



U N I V E R S I D A D
COMPLUTENSE
M A D R I D

**DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS
PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR**

D./Dña. MARLON FÉLIX CARDENAS BONETT,
estudiante en el Programa de Doctorado DE INGENIERÍA INFORMÁTICA,
de la Facultad de Informática de la Universidad Complutense de
Madrid, como autor/a de la tesis presentada para la obtención del título de Doctor y
titulada:

TÉCNICAS PARA LA EVALUACIÓN CUANTITATIVA DE SIMULACIONES 3D

y dirigida por: JORGE JESUS GÓMEZ SANZ y JUAN PAVON MESTRAS

DECLARO QUE:

La tesis es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, de acuerdo con el ordenamiento jurídico vigente, en particular, la Ley de Propiedad Intelectual (R.D. legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, modificado por la Ley 2/2019, de 1 de marzo, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), en particular, las disposiciones referidas al derecho de cita.

Del mismo modo, asumo frente a la Universidad cualquier responsabilidad que pudiera derivarse de la autoría o falta de originalidad del contenido de la tesis presentada de conformidad con el ordenamiento jurídico vigente.

En Madrid, a 23 de septiembre de 2019

Fdo.: _____

Esta DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD debe ser insertada en
la primera página de la tesis presentada para la obtención del título de Doctor.

**Técnicas para la evaluación
cuantitativa de simulaciones 3D**

**Techniques for the quantitative
evaluation of 3D simulations**



Thesis by

Marlon Félix Cárdenas Bonett

In Partial Fulfillment of the Requirements for the Degree of
Doctor por la Universidad Complutense de Madrid en el
Programa de Doctorado en Ingeniería Informática

Advisors

Jorge Jesús Gómez Sanz

Juan Pavón Mestras

Facultad de Informática
Universidad Complutense de Madrid

Madrid, Septiembre 2019

Techniques for the quantitative evaluation of 3D simulations



Thesis by

Marlon Félix Cárdenas Bonett

In Partial Fulfillment of the Requirements for the Degree of
Doctor por la Universidad Complutense de Madrid en el
Programa de Doctorado en Ingeniería Informática

Advisors

Jorge Jesús Gómez Sanz
Juan Pavón Mestras

Facultad de Informática
Universidad Complutense de Madrid

Madrid, September 2019

Técnicas para la evaluación cuantitativa de simulaciones 3D



TESIS DOCTORAL

*Memoria presentada para obtener el título de
Doctor por la Universidad Complutense de Madrid
en el Programa de Doctorado en Ingeniería Informática*

Marlon Félix Cárdenas Bonett

Directores

**Jorge Jesús Gómez Sanz
Juan Pavón Mestras**

Facultad de Informática
Universidad Complutense de Madrid

Madrid, Septiembre de 2019

*This thesis is dedicated to my family,
my mother Miryan and my brothers
Jennifer and Leonard.*

*Esta tesis está dedicada a mi Familia,
mi madre Miryan y mis hermanos
Jennifer y Leonardo.*

Acknowledgments

I give a special thanks to Juan Pavón for his support in this professional stage and to Jorge Gómez Sanz for all the experience he has given me over these years.

To my mother and my brothers for their unconditional support.

To my friends, classmates from college and work for their patience and sincere support during this time.

This doctoral thesis has been carried out within the GRASIA research group (Research Group on Social and Interdisciplinary Applications based on Agents at the UCM) and as part of the activities of the research projects “Collaborative Development of AAL Solutions”, ColosAAL (TIN2014-57028-R) and “Collaborative Design for the Promotion of Well-being in Inclusive Intelligent Cities” (TIN2017-88327-R), both funded by the Ministry of Economy and Competitiveness within the scope of the State Plan for Scientific and Technical Research and Innovation 2013-2017.

Agradecimientos

Doy un especial agradecimiento a Juan Pavón por su apoyo en esta etapa profesional y a Jorge Gómez Sanz por toda la experiencia que me aportó a lo largo de estos años.

A mi madre y mis hermanos por su apoyo incondicional.

A mis amigos, mis compañeros de universidad y de trabajo por su paciencia y apoyo sincero durante este tiempo.

Esta tesis doctoral ha sido realizada dentro del grupo de investigación GRASIA (Grupo de investigación en Aplicaciones Sociales e Interdisciplinares basadas en Agentes en la UCM) y como parte de las actividades de los proyectos de investigación “Desarrollo Colaborativo de Soluciones AAL”, ColosAAL (TIN2014-57028-R) y “Diseño Colaborativo para la Promoción del Bienestar en Ciudades Inteligentes Inclusivas” (TIN2017-88327-R), ambos financiados por el Ministerio de Economía y Competitividad dentro del marco del Plan Estatal de Investigación Científica y Técnica e Innovación 2013-2017.

Índice General

Índice de Figuras	xix
Índice de Tablas	xxiii
Lista de Acrónimos	xxvii
Abstract	xxix
Resumen	xxx
I Descripción de la investigación	1
1 Introducción	1
1.1 Motivación	1
1.1.1 La tecnología asistiva en la población mayor	2
1.1.2 Barreras en la construcción de soluciones asistivas	3
1.2 Planteamiento del trabajo y objetivos	6
1.3 Contribuciones	8
1.4 Estructura de la Tesis	11
2 Estado del arte	13
2.1 Contexto general	13
2.1.1 Técnicas de participación de los usuarios	15
2.2 Características de las soluciones de vida asistida	16
2.2.1 Elementos del dominio de la solución	18
2.3 La simulación en las soluciones de vida asistida	19
2.3.1 Validación de las soluciones	20
2.4 Uso de la simulación como especificación	21
2.4.1 Validación de la simulación	23
3 Caracterización de la solución asistiva en la simulación	25
3.1 Introducción	25
3.2 Alcance de la simulación	26
3.3 La solución asistiva	27
3.3.1 Modelado de la solución asistiva	27

3.3.2	Elementos para construir la simulación	29
3.4	Definición del caso de estudio	30
3.4.1	Usuarios	32
3.4.2	Requisitos	33
3.4.3	Elementos del dominio de la asistencia	35
3.4.4	Captura de requisitos	45
3.4.5	Organización de las tareas	46
3.4.6	Resultado de las tareas	47
3.5	Transición del caso de estudio a la simulación	52
3.5.1	Elementos de la simulación	53
3.6	Resumen	56
4	Creación de los casos de prueba de la simulación	57
4.1	Pruebas sobre la simulación	57
4.1.1	Herramientas de apoyo	58
4.1.2	Construcción del entorno de pruebas	59
4.2	Pruebas unitarias	60
4.2.1	Tratamiento de los datos	61
4.2.2	Criterios de validación de los datos	61
4.2.3	Criterios de validación del comportamiento de los datos	62
4.2.4	Criterios de validación de las actividades	63
4.2.5	Validación de la actividad física	66
4.2.6	Prueba principal y prueba secundaria o de apoyo	68
4.2.7	Pruebas unitarias predefinidas	68
4.3	Casos de prueba	69
4.3.1	Organización de los casos de prueba	70
4.3.2	Creación del caso de prueba	71
4.4	Resumen	73
5	Requisitos de diseño	75
5.1	Entorno de pruebas	75
5.2	Diseño del framework	77
5.2.1	Elementos del entorno en la simulación	79
5.2.2	Modelo de eventos	81
5.2.3	Procesamiento en capas	86
5.2.4	Modelos de reconocimiento de actividad	88
5.3	Resumen	89
6	Arquitectura del entorno de pruebas	91
6.1	Arquitectura propuesta	91
6.1.1	Motor de pruebas	93
6.1.2	Motor de reglas	94
6.2	Filtrado basado en atributos	99
6.3	Filtrado basado en umbrales e indicadores	100

6.4	Resumen	101
7	Implementación del sistema de reconocimiento	103
7.1	Sistemas de reconocimiento	103
7.2	Clasificadores	104
7.2.1	Técnicas de extracción de características	105
7.2.2	Modelos de clasificación	107
7.2.3	Configuración de los sensores	111
7.2.4	Reconocimiento de movimientos simples	111
7.3	Resumen	117
8	Experimentación con casos de prueba	119
8.1	Alcance de la experimentación	119
8.2	Casos de estudio	121
8.2.1	Usuarios	121
8.2.2	Requisitos	122
8.2.3	Soluciones asistivas propuestas	127
8.2.4	Elementos del dominio modelados	131
8.2.5	Definición de las actividades.	133
8.3	Creación de los casos de prueba	138
8.3.1	Configuración de los sensores	138
8.3.2	Definir las validaciones	140
8.3.3	Identificar los parámetros de entrada	143
8.3.4	Definir los criterios de éxito y fracaso	145
8.3.5	Crear las pruebas unitarias	145
8.4	Resultado de los casos de prueba	149
8.4.1	Evaluación del sistema de reglas	157
8.4.2	Evaluación del sistema de reconocimiento	158
8.5	Resumen	160
9	Pruebas en un entorno real	161
9.1	Alcance de las pruebas	161
9.2	Unidad principal de proceso	162
9.3	Sensores	163
9.4	Prototipos	165
9.4.1	Prueba del sistema	166
9.5	Resumen	169
10	Conclusiones y trabajo futuro	171
10.1	Contribuciones	171
10.2	Líneas de investigación abiertas	173

II	Description of the research	175
11	Techniques for the quantitative evaluation of 3D simulations	177
11.1	Introduction	177
11.1.1	Assistive technology in the elderly population	178
11.1.2	Barriers in the construction of assistive solutions	179
11.1.3	Approach of the work and objectives	181
11.2	Characterization of the assistive solution in the simulation	184
11.2.1	Model the assistive solution	184
11.2.2	Case study definition	186
11.3	Creation of simulation test cases	188
11.3.1	Building the test environment	189
11.4	Design requirements	190
11.4.1	Test environment	190
11.4.2	Event model	190
11.4.3	Activity recognition models	191
11.5	Test Environment Architecture	191
11.6	Proposed architecture	192
11.7	Implementation of the recognition system	193
11.7.1	Recognition systems	193
11.7.2	Classifiers	193
11.7.3	Modification of animations	193
11.8	Experimentation with test cases	194
11.8.1	Case study	195
11.8.2	Evaluation of the system	196
11.9	Testing in a real environment	197
11.10	Conclusions and future work	197
11.10.1	Open lines of research	199
	Bibliography	201
III	Publicaciones relacionadas con esta tesis	215
	Lista de publicaciones	217

Índice de Figuras

1.1	Tendencias del envejecimiento de la población para el año 2080.	1
1.2	Técnicas para el estudio del comportamiento de un sistema.	4
1.3	Diseño asistido por computador y manufacturación.	5
1.4	Participación de la simulación en el marco de desarrollo.	7
1.5	Descripción de los comportamientos del sistema representados por la simulación.	7
2.1	Especificaciones incluidas en la simulación.	22
3.1	Entorno de pruebas basado en el marco de desarrollo de las simulaciones.	26
3.2	Correspondencia entre las distintas iteraciones de las simulaciones.	27
3.3	Representación de los escenarios del sistema.	28
3.4	Elementos utilizados en la definición de las especificaciones del caso de estudio.	30
3.5	Diagrama de actividades utilizado en la solución global.	31
3.6	Usuarios interesados en la solución asistiva.	32
3.7	Fase de identificación de requisitos del caso de estudio.	33
3.8	Tareas recomendadas para identificar los riesgos de la tecnología invasiva.	36
3.9	Tareas para identificar posibles riesgos de aversión tecnológica.	37
3.10	Actividades para la gestión de los problemas de privacidad.	38
3.11	Diagrama de implementación del sistema de etiquetas en un entorno monitorizado con sensores.	39
3.12	Actividades propuestas para generar los planes de acción.	40
3.13	Actividades para modelar los problemas de movilidad.	41
3.14	Tareas para identificar los requisitos de la solución.	46
3.15	Transición de la simulación en el caso de estudio.	47
3.16	Resultado de la definición del caso de estudio.	48
3.17	Relación de las actividades con el caso de estudio.	49
3.18	Especificaciones del caso de estudio para generar la simulación.	52
3.19	Aspectos a validar en el elemento “ <i>usuario</i> ” de la simulación.	54
3.20	Objetivos validables en el elemento “ <i>entorno</i> ” de la simulación.	55
4.1	Esquema general del entorno de pruebas para el caso de estudio.	59
4.2	Descripción gráfica de ventana de tiempo y ventana de datos.	61

4.3	Máquina de estado para los cambios de estado y de posición en los elementos en la simulación.	62
4.4	Ocurrencia de una actividad a lo largo del tiempo.	64
4.5	Duración de la actividad en relación a una ventana de tiempo.	64
4.6	Repeticiones de la actividad en relación a una ventana de tiempo.	65
4.7	Orden en el que se ejecutan las actividades.	65
4.8	Descomposición de las actividades físicas en movimientos simples.	67
4.9	Descomposición de la actividad física en gestos simples con un único sensor.	67
4.10	Descomposición de la actividad física en gestos simples con más de un sensor.	68
4.11	Tareas necesarias para la creación del caso de prueba.	71
5.1	Principales componentes de una solución asistiva.	76
5.2	Configuración de la simulación para el uso de sensores.	79
5.3	Distribución de los sensores en el entorno simulado.	80
5.4	Organización jerárquica de eventos.	85
5.5	Jerarquías de eventos.	87
5.6	Ejemplo de animaciones en un personaje.	89
6.1	Arquitectura del sistema.	92
6.2	Diagrama de clases del sistema de pruebas.	93
6.3	Diagrama de clases empleadas en el Motor de reglas.	94
6.4	Diagrama de secuencia de la transformación de eventos realizada por el Motor de reglas.	97
6.5	Uso de filtros sobre los flujos de eventos de la simulación	98
6.6	Filtrado de eventos mediante atributos.	99
7.1	Componentes del sistema de reconocimiento de actividad.	104
7.2	Posición de los sensores sobre el cuerpo del personaje.	108
7.3	Predicción de la actividad con 3 sensores.	111
7.4	Uso de reglas para analizar los resultados de la predicción de actividades.	112
7.5	Uso de reglas para analizar los resultados de la predicción de gestos o acciones.	113
7.6	Ángulos de rotación y traslación en las articulaciones principales del personaje.	114
8.1	Herramienta web para el estudio de tecnología asistiva, aplicable en personas mayores con determinados padecimientos.	122
8.2	Estudio de síntomas específicos en enfermedades degenerativas.	123
8.3	Visualización de las simulaciones en la web.	124
8.4	Creación de comentarios sobre el vídeo de la simulación.	124
8.5	Plataforma Web para el uso de técnicas de trabajo en grupo.	125
8.6	Personajes propuestos para las simulaciones.	130
8.7	Escenario propuesto para las simulaciones.	130
8.8	Diagrama de secuencia de las actividades en el escenario típico.	134
8.9	Diagrama de secuencia de las actividades en el escenario problema.	135
8.10	Diagrama de secuencia de las actividades en el caso de estudio CS002DI.	136

8.11 Problemas añadidos en la simulación del caso de estudio CS002DI. 137

8.12 Posición de los sensores en la simulación de los casos de estudio. 139

8.13 Resultados del caso de prueba CS001ML para el escenario típico. 150

8.14 Resultados del caso de prueba CS001ML para el escenario problema. 152

8.15 Resultados del caso de prueba de CS002DI escenario típico. 152

8.16 Evolución de la predicción del etiquetado de reglas en acciones repetitivas. . 153

8.17 Resultado del caso de prueba CS002DI para el escenario problema. 154

8.18 Añadiendo pruebas secundarias en el caso de prueba CS002DI para el
escenario problema. 155

8.19 Resultados del caso de prueba CS003AR para el escenario típico. 155

8.20 Resultados del caso de prueba CS003AR para el escenario problema. 156

8.21 Matriz de confusión de la evaluación del sistema de reglas. 158

8.22 Matriz de confusión de la evaluación del sistema de reconocimiento. 159

9.1 Ordenadores de una sola placa utilizados en las pruebas. 162

9.2 Wearables utilizados en las pruebas. 164

9.3 Sensores individuales utilizados en las pruebas. 165

9.4 Primer prototipo del sistema de reconocimiento. 165

9.5 Prototipos del sistema de reconocimiento. 166

11.1 Trends in the ageing of the population for the year 2080. 177

11.2 Techniques for the study of the system’s behaviour. 179

11.3 Computer-aided design and manufacturing. 180

11.4 Participation of the simulation in the development framework. 182

11.5 Description of the behaviours of the system represented by the simulation. . 182

11.6 Correspondence between the different iterations of the simulations. 185

11.7 Representation of system scenarios. 185

11.8 Case study specifications to generate the simulation. 187

11.9 Aspects to validate in the “*user*” element of the simulation. 187

11.10 Aspects to validate in the “*environment*” element of the simulation. 188

11.11 General scheme of the test environment. 189

11.12 System architecture. 192

11.13 Rotation and translation angles in the main articulations of the character. . 194

Índice de Tablas

1.1	Lista de publicaciones que avalan la tesis.	11
3.1	Características del elemento “ <i>usuario</i> ” en la simulación.	34
3.2	Características del elemento “ <i>entorno</i> ” en la simulación.	35
3.3	Características de los “ <i>dispositivos</i> ” emulados en la simulación.	35
3.4	Criterios de aceptación utilizados en las pruebas del caso de uso.	37
3.5	Plan de acción propuesto para las respuestas no deseadas de cada usuario.	40
3.6	Caracterización de los cambios anormales en el usuario.	41
3.7	Caracterización de los cambios normales en el usuario.	43
3.8	Guía de evaluación de la movilidad orientada al desempeño.	44
3.9	Participación de los roles en la descripción de la simulación.	45
3.10	Primera aproximación: actividades que se simularían en el caso de estudio de ejemplo.	50
3.11	Segunda aproximación: actividades que se simularían en un caso de estudio más detallado.	50
3.12	Tercera aproximación: actividades que se simularían con mayor detalle en un caso de estudio.	51
3.13	Plantilla propuesta para describir el escenario de la simulación.	52
4.1	Elementos de la simulación con sus respectivas pruebas unitarias.	60
4.2	Métodos auxiliares para la comprobación del comportamiento de los elementos en la simulación.	69
4.3	Métodos auxiliares para la captura de datos previos.	69
4.4	Resumen de la información necesaria para el caso de prueba.	70
5.1	Estructura básica de los datos del sensor.	81
5.2	Descriptores para identificar reglas y actividades predefinidas.	84
5.3	Ejemplo de cada tipo de evento de la jerarquía.	86
6.1	Atributos y filtros utilizados para cada tipo de sensor.	99
6.2	Escala de colores empleada por defecto.	100
6.3	Indicadores por defecto del Motor de reglas.	100
6.4	Consultas correspondientes a los umbrales predefinidos.	101
7.1	Configuración por defecto de los clasificadores.	108

7.2	Precisión de la clasificación de los diferentes modelos con uno y dos sensores.	109
7.3	Tiempos de respuestas y latencia de los algoritmos de clasificación utilizados	110
7.4	Atributos utilizados en la modificación de la animación.	115
8.1	Descripción general del primer caso de estudio relacionado con la pérdida de memoria.	126
8.2	Descripción general del segundo caso relacionado con la desorientación. . . .	127
8.3	Descripción general del tercer caso relacionado con la detección de caídas en el hogar.	127
8.4	Características de las soluciones propuestas.	128
8.5	Restricciones en los casos de estudio propuestos.	129
8.6	Escenarios propuestos en cada solución.	129
8.7	Características a modelar en el caso de prueba de CS003AR.	131
8.8	Caracterización de los cambios anormales en el usuario modelados en el caso de prueba.	132
8.9	Actividades propuestas para los distintos escenarios del caso de estudio CS001ML.	134
8.10	Actividades propuestas para los escenarios del caso de estudio CS002DI. . .	136
8.11	Actividades propuestas para los escenarios del caso de estudio CS003AR. .	138
8.12	Configuración de los sensores en la simulación de cada caso de uso.	139
8.13	Lista de actividades y salidas esperadas en el caso de estudio CS001ML. . .	141
8.14	Lista de actividades y salidas esperadas en el caso de estudio CS002DI. . .	142
8.15	Lista de actividades y salidas esperadas en el caso de estudio CS003AR. . .	143
8.16	Parámetros necesarios para validar el caso de prueba en CS001ML.	144
8.17	Parámetros necesarios para validar el caso de prueba en CS002DI.	144
8.18	Parámetros necesarios para validar el caso de prueba en CS003AR.	144
8.19	Pruebas unitarias utilizadas en el caso de estudio CS001ML.	146
8.20	Pruebas unitarias utilizadas en el caso de estudio CS002DI.	147
8.21	Pruebas unitarias utilizadas en el caso de estudio CS003AR.	148
8.22	Reglas específicas para identificar problemas generales en elementos de la simulación.	149
8.23	Indicadores de evaluación del entorno de pruebas para CS001ML y CS002DI.	158
8.24	Total de modificaciones por cada animación.	159
8.25	Indicadores de evaluación del entorno de pruebas para CS003AR.	159
9.1	Comparativa del consumo de potencia en los dispositivos propuestos.	163
9.2	Comparativa de los dispositivos utilizados.	164
9.3	Número de datasets utilizados por cada actividad del sistema.	167
9.4	Matriz de confusión de los algoritmos utilizados en el sistema.	167
9.5	Indicadores de evaluación de los algoritmos utilizados en el sistema.	168

Lista de Acrónimos

2D	<i>2D computer graphics</i>
3D	<i>3D computer graphics</i>
AAL	<i>Ambient Assisted Living</i>
AES	<i>Advanced Encryption Standard</i>
AIDE	<i>Ambient Intelligence Development Environment</i>
AmI	<i>Ambient Inteligence</i>
CAD	<i>Computer Aided Design</i>
CAM	<i>Computer Aided Manufacturing</i>
CEP	<i>Complex Event Processing</i>
ColosAAL	Desarrollo Colaborativo de Soluciones AAL
CPU	<i>Central Processing Unit</i>
EDA	<i>Event-Driven Architecture</i>
EHA	<i>Elderly in-Home Assistance</i>
EPL	<i>Event Processing Language</i>
Eurostat	Oficina Europea de Estadística
FFT	<i>Fast Fourier Transform</i>
GLS	<i>Generic Location event Simulator</i>

HAR	<i>Human Activity Recognition</i>
HDMI	<i>High-Definition Multimedia Interface</i>
I2C	<i>Inter Integrated Circuits</i>
IoT	<i>Internet of Thing</i>
JSON	<i>JavaScript Object Notation</i>
kNN	<i>k-Nearest Neighbors</i>
MCU	<i>Master Controller Unit</i>
OMS	Organización Mundial de la Salud
PHAT	<i>Physical Human Activity Tester Simulator</i>
PLN	Procesamiento de Lenguaje Natural
RF	<i>Random Forest</i>
RRI	<i>Responsible Research & Innovation</i>
SociAAL	<i>Social Ambient Assisted Living</i>
SociAALML	<i>Social Ambient Assisted Living Modelling Language</i>
SQL	<i>Structured Query Language</i>
SVC	<i>Support-vector Clustering</i>
SWEBOK	<i>Guide to the Software Engineering Body of Knowledge</i>
TIC	Tecnologías de la Información y la Comunicación

VLL *Virtual Living Lab*

WHO *World Health Organization*

Abstract

The growing number of connected devices to Internet and many socio-cultural factors worldwide, such as the ageing of the population, has led to a considerable increase in the use of ambient intelligence for the construction of technological solutions that meet the specific needs of the people. Designing and building this type of assistive solutions is not an easy task because the cost of adapting the environment and especially because they require a great deal of adaptation to the very specific needs of each user. In addition, it is also important to take into account the different actors involved (users, caregivers, family members, experts, engineers, among others), who somehow have to facilitate their interaction in the development process.

One way to do the interaction of the different actors in the development process is through the use of 3D simulations that generate virtual scenarios capable of testing ideas and evaluating the integration of technologies. In this scenario, it is possible to do a lot of experimentation before building a final solution. However, this entails the need to solve several issues that arise with the use of simulation as a specification element. For example, its generation, its validation and its shared use among the different users. Generating the simulation suggests using a development methodology that allows covering all the requirements of the solution as well as the technical characteristics to be included in it. Validation, on the other hand, means checking that each component of the simulation fulfils its function, which is not trivial. As far as a means for communication, the use of videos generated from 3D simulations offers an added value in itself given that it can visually reach a greater number of users.

For all of the above, a development methodology is proposed that allows the simulation to be defined and the characteristics of the assistive solution to be transferred to it. The purpose is that this simulation can then be used as a specification element that collects the requirements of the solution. Subsequently, this will be validated through a testing framework for simulations that facilitates the development of solutions based on ambient intelligence. This framework promotes the virtualization of sensors and sensor networks for the construction of useful solutions in virtual assisted living environments. An event-based architecture has been proposed for the test environment, the purpose of which is to facilitate the execution of test cases that are defined with the methodology. Finally, experimentation has been made, in a virtual laboratory and in a real environment, where it is shown how the methodology and the test environment are used to determine the value that simulations provide as specification elements in the assistive solutions.

Keywords: Ambient intelligence, Artificial Intelligence, Assisted Living, Simulations, Virtual Living Labs.

Resumen

El creciente número de dispositivos conectados a Internet sumado a muchos factores socio-culturales a nivel mundial, tales como el envejecimiento de la población, han impulsado un aumento considerable del uso de la inteligencia ambiental para la construcción de soluciones tecnológicas que cubran necesidades concretas de las personas. Diseñar y construir este tipo de soluciones asistivas no es tarea fácil. El desarrollo de una solución para personas mayores u otras que lo requieran, es complicado no solo por el coste que supone la adecuación del entorno sino también porque requieren de mucha adaptación a necesidades muy específicas de cada usuario. Además, también es importante tener en cuenta los distintos actores que intervienen (usuarios, cuidadores, familiares, expertos, ingenieros, entre otros), a los que de alguna forma hay que facilitarles su interacción en el proceso de desarrollo.

Una forma de hacerlo es mediante el uso de simulaciones 3D que permitan generar escenarios virtuales capaces de poner a prueba las ideas y evaluar la integración de las tecnologías. En este escenario es posible realizar mucha experimentación antes de construir una solución final. Sin embargo, esto acarrea la necesidad de solventar varias cuestiones que surgen con el uso de simulación como elemento de especificación. Por ejemplo, su generación, su validación y su uso compartido entre los distintos usuarios. Generar la simulación sugiere usar una metodología de desarrollo que permita cubrir todos los requisitos de la solución a la vez que las características técnicas que se han de incluir en ella. La validación, por su parte, supone comprobar que cada componente de la simulación cumple su función, lo cual no es nada trivial. En lo que respecta al uso compartido, al ser un formato en 3D, esta ofrece de por sí un valor añadido dado que puede llegar de forma visual a un mayor número de usuarios.

Por todo lo anterior, se propone una metodología de desarrollo que permita definir la simulación y trasladarle las características de la solución asistiva. El propósito es que dicha simulación pueda luego ser usada como un elemento de especificación que recoja los requisitos de la solución. Posteriormente, esta será validada mediante un framework de pruebas para simulaciones que facilita el desarrollo de soluciones basadas en la inteligencia ambiental. Este framework promueve la virtualización de sensores y redes de sensores para la construcción de soluciones útiles en entornos virtuales de vida asistida. Para el entorno de pruebas se ha propuesto una arquitectura basada en eventos, cuyo propósito es facilitar la ejecución de los casos de prueba que se definan con la metodología. Finalmente, se ha hecho una experimentación en un laboratorio virtual y en un entorno real, donde se muestra cómo se aprovechan la metodología y el entorno de pruebas, para determinar el valor que aportan las simulaciones como elementos de especificación en las soluciones asistivas.

Palabras clave: Ambientes Inteligentes, Inteligencia Artificial, Simulaciones, Vida Asistida, Laboratorios Vivientes Virtuales.

Parte I

Descripción de la investigación

Capítulo 1

Introducción

1.1 Motivación

La población europea enfrenta una serie de cambios sociales críticos que alteran distintas áreas de su estructura actual. El sistema económico, los servicios de salud, el sector laboral, la estructura familiar, entre otros, se han visto afectados por la tendencia progresiva de la sociedad a envejecer, la cual se estima continúe así hasta el año 2080. Como se puede ver en la Figura 1.1, según las estadísticas de Oficina Europea de Estadística (Eurostat) [Eur18], la expectativa es que en el año 2045 la población total sea cerca de los 530 millones de personas, un 4 % más en comparación con el año 2016.

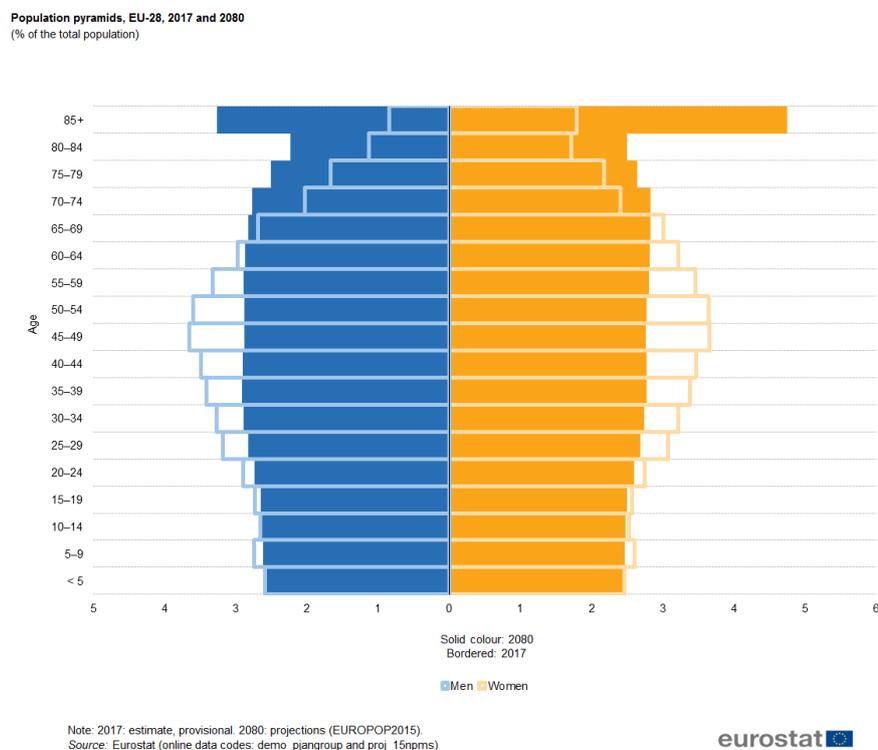


Figura 1.1: Tendencias del envejecimiento de la población para el año 2080.

Aunque es normal que la población tienda a aumentar, el problema radica en que el

crecimiento no está siendo homogéneo y equilibrado dentro de los distintos grupos de edad. Con el envejecimiento de la sociedad, aumentan otros factores de riesgos relacionados con el padecimiento de alguna enfermedad crónica y degenerativa, que impida a las personas mayores llevar una vida autónoma e independiente. Este tipo de problemas afecta no solo la calidad de vida del individuo en sí, sino que también produce cambios en el comportamiento y la estructura familiar, la cual se ve ampliamente afectada por la situación de dependencia que se genera con este tipo de enfermedades. A nivel social, si la población mayor se enferma y no es capaz de sostenerse por sí misma, al final todo esto se traduce en más gastos destinados a brindar mayor asistencia social.

Ante este panorama, el foco de atención se orienta hacia los servicios de asistencia sanitaria. El aumento de personas mayores vaticina una creciente demanda de sistemas de atención y monitorización que, a largo plazo y desde el propio domicilio, sean capaces de cubrir necesidades concretas de los pacientes, teniendo en cuenta los factores de riesgo a los que se pueda enfrentar cada persona en su propio contexto. Asimismo, también se hace necesario planificar y capacitar a más profesionales en salud con conocimientos suficientes en áreas como la atención primaria, la gerontología, la geriatría y especialmente en el uso de nuevas tecnologías que sirvan de apoyo en la atención asistencial. En relación a esto, la Organización Mundial de la Salud (OMS) recomienda que los sistemas de salud europeos [O⁺11] cuenten con un mecanismo de capacitación constante sobre el cuidado de personas mayores, especialmente en edades avanzadas donde es más crítica la atención sanitaria [HRH⁺09].

La OMS define el cuidado a largo plazo [WHO18] como “El conjunto de actividades que realizan tanto los cuidadores informales (familiares, vecinos o amigos) como los profesionales (especialistas, asistentes sociales) para garantizar que una persona que no puede cuidarse a sí misma, conserve una calidad de vida idónea en función de sus preferencias individuales, con un alto grado de independencia posible, autonomía, participación, realización personal y dignidad humana”. En esta definición, tanto los cuidadores como los profesionales de la salud tienen la misma importancia en el ámbito asistivo. Sin embargo, en los casos que el cuidador es un miembro de la familia, su papel es aun más relevante porque por lo general, esto cambia el comportamiento y los hábitos de familias enteras, las cuales se ven obligadas a adaptarse a las nuevas atenciones que demanda el usuario. La demanda de cuidadores va en aumento y esto se debe principalmente a que la inversión en infraestructura social se sigue centrando en los servicios de las instituciones y no en la creación de estrategias que mitiguen directamente la fuerte demanda de una sociedad envejecida. Ante este panorama, es necesario idear planes que hagan converger los tiempos de atención informal (cuidadores) con la inversión del gasto público, de manera que se planifiquen servicios con garantías suficientes para que los sistemas de salud cubran la atención a largo plazo de manera sostenible.

1.1.1 La tecnología asistiva en la población mayor

Con el objetivo de cubrir las actuales y futuras necesidades de la sociedad, paradigmas como *Ambient Intelligence* (AmI), surgen con gran fuerza para impulsar la creación de soluciones ambientales tecnológicas que permitan a los usuarios interactuar con el entorno

y que este les provea de algún servicio. Bajo esta misma idea, existen disciplinas basadas en la vida cotidiana asistida por el entorno *Ambient Assisted Living* (AAL) [RM13], que se centran especialmente en problemas relacionados con el envejecimiento de la población, ofreciendo alternativas que permitan a los usuarios ser asistidos por el entorno en sus hogares *Elderly in-Home Assistance* (EHA) [ZJCZ11], en función de sus preferencias, características o limitaciones [LCC⁺18].

En este escenario tecnológico, es importante disponer de un número amplio de dispositivos que permitan recoger la mayor cantidad de datos posible, no solo del contexto sino también del usuario. Con toda la información obtenida, es mucho más fácil detectar, predecir y actuar en función de las necesidades diarias del usuario, bien sea dentro o fuera del hogar. El creciente uso del Internet de las Cosas o *Internet of Thing* (IoT) [WADX15] ha contribuido al aumento del número de dispositivos que, a través de sensores, capturan información del entorno y posteriormente son capaces de publicar estos datos para que otros dispositivos consuman la información, tomen decisiones y actúen en consecuencia a las condiciones del entorno. Aprovechando esa capacidad de conexión entre dispositivos y su amplia presencia en el entorno, AAL pretende mejorar la calidad de vida de las personas [RSL⁺18], automatizando tareas cotidianas del entorno dentro y fuera del hogar, creando también servicios sensibles al contexto para ofrecer determinada asistencia justo en el momento en el que el usuario la necesita.

Es así como la AmI y AAL hacen su contribución en el campo de la asistencia, dotando al entorno de “objetos inteligentes” [PG16] cotidianos que integren sensores y procesadores para recoger información del contexto y adaptarse así a las necesidades del usuario de forma transparente y anticipada [SCSE10]. Dentro de la información del contexto también se encuentran las actividades que realiza el usuario [ABC⁺16], las cuales son analizadas e interpretadas a través de los sistemas *Human Activity Recognition* (HAR) [DOB⁺16]. En este tipo de soluciones [ABC⁺16], un sistema HAR es una pieza importante que permite identificar no solo las acciones del usuario [CFR14], sino también las distintas formas con las que este podría interactuar con la solución [ROGP⁺13] y el entorno [Har09].

1.1.2 Barreras en la construcción de soluciones asistivas

La construcción de una solución asistiva no es una tarea fácil de llevar a cabo. Para definir una solución de este tipo, es necesario identificar muchas variables que condicionan su desarrollo, en concreto aquellas cuya solución final se compone de los siguientes elementos:

- La solución o el servicio que se quiere brindar a determinado usuario.
- El software que controla la solución [HCL⁺12].
- El hardware que permitirá la interacción del usuario con la solución [BE02].

En lo que respecta al software, la primera necesidad que surge es identificar una metodología de desarrollo que permita su construcción y validación, teniendo en cuenta las implicaciones que este tendría al tratarse de un proyecto software con fines asistivos. Existen diferentes metodologías que contribuyen con determinadas pautas para crear

soluciones asistidas. Entre muchas de sus recomendaciones, se describe claramente la necesidad de utilizar modelos que representen y simplifiquen los requisitos de la solución, con el fin de poder experimentar con ella. El enfoque de AAL se basa en el uso de entornos reales controlados para la experimentación, conocidos como living labs [KOA⁺99]. El principal inconveniente de este tipo de sistemas es que requieren de una inversión económica considerable para la construcción de los laboratorios, lo cual en muchos casos puede ser equivalente a construir prototipos de la solución esperada a costes difícilmente predecibles y posteriormente poco asequibles para el usuario final.

Según la *Guide to the Software Engineering Body of Knowledge* (SWEBOK), el prototipado, el modelado y la simulación (ver Figura 1.2) son técnicas muy útiles en proyectos donde el proceso de captura de requisitos es muy importante, así como la visualización temprana del comportamiento de un sistema desde una perspectiva concreta [BF⁺14]. Sin embargo, cada una de estas técnicas tiene particularidades que se deben tener en cuenta a la hora de ser empleadas en cualquier tipo de desarrollo.

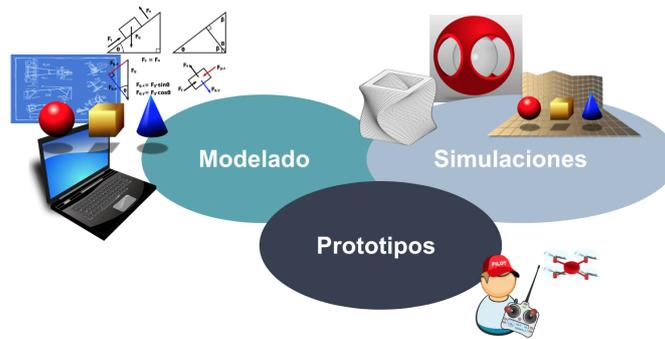


Figura 1.2: Técnicas para el estudio del comportamiento de un sistema.

En el caso de los prototipos, se busca captar una versión temprana del sistema [BF⁺14] para que los diseñadores puedan ir verificando la viabilidad del servicio de asistencia durante el diseño de la misma. El prototipo puede emplearse en muchas fases del desarrollo, pero según lo complejo que pueda ser definir la asistencia, son más útiles en el proceso de captura de requisitos, en la construcción de los elementos interactivos o en los procesos de validación de los requisitos funcionales. Pese a sus múltiples ventajas, utilizar esta abstracción requiere una inversión de tiempo y dinero para su construcción, lo cual se traduce en un aumento de los costes tanto de fabricación como implantación a nivel de hardware. El modelado por su parte, se emplea para abstraer parte de un proceso e identificar así características de la solución. Éstas son aprovechadas por las simulaciones, las cuales utilizan dichas características para facilitar la ejecución de experimentos que permitan por ejemplo, identificar el comportamiento y las relaciones del sistema, estudiar aspectos del diseño, entre otros. Mediante estas técnicas también es posible construir teorías o hipótesis sobre cómo el sistema se podría comportar y en función de ello, crear predicciones sobre la solución final.

Aunque las simulaciones hayan sido un elemento clásico en los desarrollos de sistemas por modelos, debido a su naturaleza, solo modelan determinados aspectos del sistema y es por ello que no han trascendido más allá del uso que tienen actualmente. Un ámbito en

el que han tenido cierto recorrido es en *Computer Aided Design (CAD)/Computer Aided Manufacturing (CAM)* [ZS91], las cuales, a través del trabajo asistido por computador, han facilitado el diseño CAD y la manufacturación CAM de soluciones [WZ02].

Como se observa en la Figura 1.3, la simulación está presente como una representación de la solución, en la cual partiendo de una idea, se diseñan y modelan los distintos componentes que constituyen un producto. En esta etapa de diseño y modelado, el desarrollador establece los parámetros en cada simulación y los va ajustando en función de los requisitos que vayan surgiendo. Cuando esta representación del producto permite observar y evaluar sus componentes, el paso siguiente es utilizar la simulación en los procesos de fabricación para construir los distintos elementos físicos que conforman la solución.

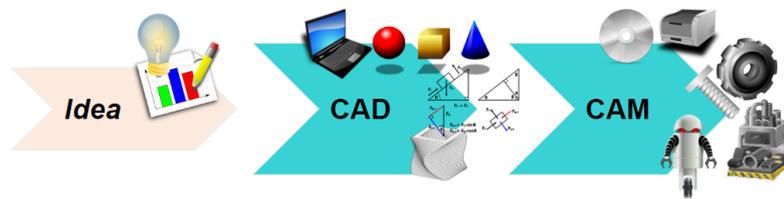


Figura 1.3: Diseño asistido por computador y manufacturación.

En una solución de asistencia, estas abstracciones se pueden utilizar para generar parte de la solución. Sin embargo, la definición general de estas representaciones del problema, puede que no ayuden a abaratar la solución debido al tiempo que se debe invertir en la construcción de los diferentes modelos. Es por eso que dicha alternativa, a largo plazo, ralentiza no solo la experimentación sino también los procesos de validación de la solución [NSK05]. Por otro lado, hay características abstractas en las soluciones asistidas que no se pueden incluir fácilmente en estos elementos, por ejemplo, las acciones y las posibles respuestas del usuario a la solución, los casos de aversión tecnológica, la adaptación de la asistencia al medio ambiente [VNKC09] y la evolución del paciente, entre otros.

La simulación ha sufrido una serie de cambios estructurales que han promovido su participación en procesos de modelado del sistema. En el caso de la Inteligencia Ambiental, herramientas como *Social Ambient Assisted Living Modelling Language (SociAALML)* [CSGS15], mediante un lenguaje gráfico y específico de dominio, permiten a usuarios no expertos en programación crear simulaciones en *3D computer graphics (3D)* que representan o bien un problema o bien una solución. SociAALML genera el código fuente necesario para ejecutar la simulación, por lo que el diseñador sólo debe preocuparse por definir la solución y no por aspectos más propios de programación. Pese a todo esto, la simulación no deja de ser una representación visual tridimensional que reproduce animaciones pre-grabadas [BBET97] en ordenador, lo que hace necesario disponer de tiempo suficiente para estudiar y evaluar tanto su contenido como el comportamiento de los distintos elementos que la componen.

En este tipo de filosofía de desarrollo conocida como *Virtual Living Lab (VLL)* [GSCBPC16], la simulación en 3D juega un papel muy importante como

herramienta de apoyo para lograr la construcción de la solución, utilizando para ello, ciclos de diseño y desarrollo convencional de software. La simulación representa un prototipo de la solución asistida, en la cual se incorpora a todos los actores: el entorno (e.g. una vivienda), los usuarios (e.g. pacientes, médicos, cuidadores, familiares), los sensores (e.g. wearables, smartwatch, electrodomésticos).

Llegados a este punto, y después de involucrar la simulación 3D en el ámbito del desarrollo de soluciones asistivas, el reto que surge es establecer de qué forma dicha simulación puede participar en el ciclo de desarrollo, al tiempo que promueve la participación de los distintos actores en la definición de la solución, especialmente cuando existan casos particulares de usuarios cuya participación esté limitada de alguna forma. Para lograr esto, se requieren de artefactos que sean entendibles por todos los participantes y que vayan más allá en aspectos colaborativos y mejora de la implantación en hardware real de lo que se desarrolla en el entorno virtual. Asimismo, se plantea la necesidad de definir una metodología que permita validar la simulación cuando ésta se incluya en el proceso de desarrollo como un elemento, que recoge especificaciones de la solución y sea utilizada para mostrar una representación temprana de lo que podría ser la solución final o una parte de ella.

Desde el punto de vista de los requisitos, así como es necesario definir técnicas que evalúan cada requisito, asimismo se debe establecer un entorno de pruebas que determine si los requisitos representados en la simulación son los esperados por el usuario final y por los expertos en la solución asistiva. De igual forma, también se debe tener en cuenta el cambio de los requisitos durante el desarrollo. Con cada nueva iteración, la simulación puede ser modificada y el entorno de pruebas debe ser capaz de adaptarse a estos cambios, garantizando así que los requisitos ya validados en iteraciones pasadas no se pierdan.

En lo que respecta al uso de la simulación como un elemento de visualización temprana, su formato visual en 3D se debe tener en cuenta al momento de ser evaluada como solución asistiva. La técnica de validación que se emplee debe minimizar el tiempo de “inspección visual” por parte de los expertos, especialmente cuando se trate de una solución que está representada por horas de simulación.

1.2 Planteamiento del trabajo y objetivos

Como se ha descrito anteriormente, la definición de soluciones asistivas involucra muchas piezas que se deben tener en cuenta para su desarrollo. De igual forma, son muchas las personas que pueden participar en la captura de requisitos de la solución final. Existen numerosos trabajos en la literatura que describen concretamente la construcción de soluciones asistivas puntuales, especialmente en el sector de la industria. Sin embargo, no hay un claro estándar que indique de qué forma es posible emplear todos los recursos existentes para diseñar, construir y validar una solución que brinde un servicio de asistencia determinado.

El uso de la simulación 3D como elemento clave en la definición de la solución asistiva, no resuelve los problemas que existen en el diseño y la validación de la asistencia. Independientemente de la metodología de desarrollo utilizada y aun con el uso de las

simulaciones propuestas por la herramienta SociAALML, es necesario establecer un mecanismo que garantice a los expertos que la simulación recoge los requisitos que la definen y puede ser usada como un indicador del comportamiento de la solución en un entorno real.

Asumiendo como hipótesis que las simulaciones hacen parte del ciclo de desarrollo de la solución, y que éstas son utilizadas como una representación temprana de su comportamiento, el planteamiento de esta Tesis es proponer un entorno de pruebas que pueda ser empleado bajo el marco conceptual de las simulaciones. En dicho marco, la simulación es usada como un elemento activo en la captura de requisitos y en la validación de la solución final. Como se observa en la Figura 1.4, en este contexto las pruebas de la solución se pueden establecer en dos fases concretas del desarrollo:

- En su definición, la cual tiene que ver directamente con los requisitos y su aceptación.
- En pruebas del sistema, donde es posible acoplar una solución tecnológica real en la simulación, gracias a la infraestructura disponible en la herramienta.

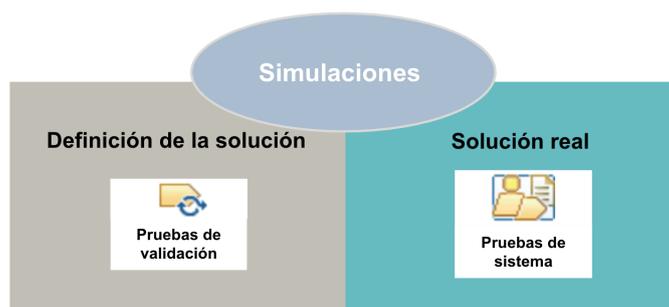


Figura 1.4: Participación de la simulación en el marco de desarrollo.

El entorno de pruebas en la fase de definición utiliza principalmente la propia simulación. Su propósito es determinar si lo que se espera de la solución se representa en la simulación, si recoge el problema concreto que se quiere resolver, si el sistema responde a esta representación y si se cumplen las expectativas esperadas por el usuario. En lo que respecta al entorno de pruebas con la solución real, el propósito en este caso es acoplar piezas de una solución real directamente con la simulación, de tal forma que se pueda comprobar si los requisitos simulados satisfacen tanto las expectativas del usuario, como las características de la solución como producto que puede ser implementado de un entorno real. Estas fases se pueden observar en la Figura 1.5.

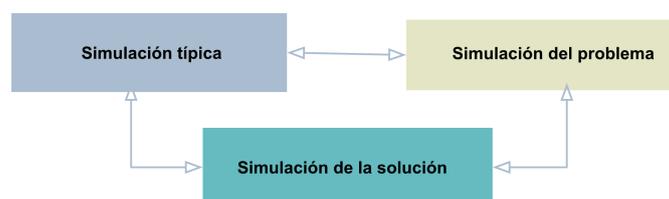


Figura 1.5: Descripción de los comportamientos del sistema representados por la simulación.

En ambos entornos, se espera que la simulación represente el comportamiento típico del sistema, el comportamiento que recoge el problema y el comportamiento esperado o correcto. Con esta representación, se describe con claridad por un lado el comportamiento de partida, el cual comprende la situación cotidiana que se ve afectada por una enfermedad o padecimiento; y por otra parte, la solución. En este punto, la asistencia es lo que diferencia la transición entre ambos comportamientos, permitiendo así que ésta actúe como la solución control que guíe a los ingenieros en la construcción de una solución final.

En condiciones normales, esta dinámica sería compleja de llevar a cabo porque habría que asumir una serie de costes relacionados con el alquiler de instalaciones y la compra de hardware u otros elementos, que permitan observar distintos comportamientos de la solución a partir de diferentes versiones de ésta. Es decir, se necesitaría construir varios prototipos de la solución para observar cómo se comporta en los distintos escenarios propuestos. Con la metodología propuesta esto no hace falta porque es posible usar directamente la simulación para definir la solución final, tanto a nivel software como hardware y para comprobar si lo simulado se parece lo suficiente a la realidad.

Para cumplir con este planteamiento, en esta tesis se proponen los siguientes objetivos específicos:

1. Hacer uso de la simulación en 3D como parte de un proceso de prototipado en el que se representa la construcción de una solución asistiva. Para ello, la simulación recogerá las características que definen la solución asistiva en un entorno real, razón por la cual dicha simulación intentará siempre representar casos de estudios lo más realista posible.
2. Definir un procedimiento que permita identificar y analizar el comportamiento de los distintos elementos que componen la simulación y que la definen como una solución simulada.
3. Facilitar la creación de escenarios de prueba que permitan validar y verificar el comportamiento de los elementos en la solución asistiva, la cual por lo general estará representada mediante el uso de dispositivos electrónicos virtualizados como sensores y actuadores que recogen toda la información del ambiente, incluida la del usuario.
4. Implementar una arquitectura que permita definir de manera programática, un entorno de pruebas que facilite la comprobación de una solución asistiva basada en simulaciones para su aceptación final. El mismo entorno de pruebas podrá garantizar a los desarrolladores que, durante la fase de desarrollo de la solución, con cada nueva simulación como prototipo no se pierden funcionalidades que ya han podido ser validadas o verificadas.

1.3 Contribuciones

Los resultados de la presente Tesis han utilizado la simulación como un producto de trabajo dentro del proceso de creación de soluciones asistivas, las cuales se basan en

tecnologías AmI aplicada a soluciones AAL. Para avalar esta investigación, los artículos científicos publicados se enfocaron en tres aspectos importantes de la investigación:

- **Definición de casos de estudio:** la simulación se utilizó como especificación de la solución asistiva en los casos de estudio con el propósito de aplicar sobre ellos la metodología propuesta en la presente Tesis. Para ello, se han estudiado técnicas de cooperación y trabajo en grupo que, mediante el uso de herramientas Tecnologías de la Información y la Comunicación (TIC), facilitan la coordinación de los distintos grupos de usuarios que participan en la creación de las simulaciones. Las técnicas utilizadas en las publicaciones han sido la *cocreación* [GSPACB17] y los *Text-Based Focus Groups* [CBCGSP18], los cuales han facilitado la tarea de automatizar los procesos de retroalimentación entre usuarios y equipo de desarrolladores de las simulaciones. Una primera aproximación de la metodología propuesta fue expuesta en el workshop [UC317] sobre sistemas socio-técnicos complejos, donde se evidenció la necesidad de usar plataformas web que contribuyan en el proceso de definición y creación de la simulación. Estos resultados hicieron posible la construcción de las plataformas que se utilizan en la presente Tesis. Un caso de estudio fue propuesto en un congreso sobre IoT [GMIJ19]. En él, los investigadores podían diseñar los escenarios, incluir las actividades diarias de los individuos, desplegar un determinado conjunto de sensores y finalmente utilizar los algoritmos de reconocimiento de actividad que más se ajustasen a la detección de patrones en el comportamiento relacionados posiblemente con la desorientación del usuario.
- **Prácticas RRI en la construcción de casos de estudio:** como parte de la filosofía de trabajo se ha tenido en cuenta incluir las prácticas *Responsible Research & Innovation* (RRI), cuyo propósito es reducir la brecha que existe entre la comunidad científica y la sociedad e incentivar a los distintos grupos de interés para que trabajen juntos en todo el proceso de investigación e innovación. En la construcción de soluciones asistivas, es fundamental involucrar a todos los actores que de una u otra forma, se ven beneficiados por la asistencia que ofrece la solución, sobre todo cuando es posible utilizar dicha asistencia para atacar problemas sociales actuales. En línea con lo anterior, una de las publicaciones [BMCB18] determina la importancia que tienen las TIC en los procesos de intercambio de conocimiento y cocreación de las partes interesadas.
- **Plataforma Hack4People¹:** es plataforma web que ha sido desarrollada para la coordinación de los distintos grupos de usuarios que participan en la creación de las simulaciones. La plataforma permite utilizar la técnica de grupos focales online para definir los casos de estudio a partir de los cuales se generan las simulaciones.
- **Validación de las simulaciones:** se usa la simulación como un elemento de especificación de la solución asistiva empleando la metodología y la arquitectura propuesta. En la publicación [CBSP18] se mostró cómo es posible abordar dicha validación dentro del ciclo de desarrollo de una solución asistiva.

¹Plataforma Web: <http://grasiagroup.fdi.ucm.es/hackwithpeople/index.php>

- **Experimentación con escenarios de prueba:** definir escenarios complejos donde la simulación se utiliza en diferentes contextos de las soluciones asistivas. Parte del framework se ha puesto a prueba en un escenario real controlado, donde se ha desplegado un grupo de sensores ambientales de distintos tipos para la monitorización [CBB18]. Las condiciones reales del entorno fueron establecidas con ayuda de la simulación, las cuales se utilizaron para diseñar distintos escenarios de prueba. El objetivo era encontrar un escenario con la mejor distribución de sensores posible según su tipo, la configuración disponible y las características técnicas. Una vez la simulación mostró la configuración adecuada, el paso siguiente fue emplear la arquitectura para validar las distintas condiciones que se esperaban del entorno para un grupo de usuarios específicos.
- **Framework de pruebas**^{2 3 4}: es la arquitectura propuesta en esta Tesis para que los desarrolladores construyan el entorno de pruebas necesario para validar y verificar las características de la solución asistiva que se han incluido en la simulación.

En otras publicaciones, se ha mostrado la importancia del uso de las simulaciones como elemento de captura de requisitos en soluciones AAL, especialmente cuando se busca representar el comportamiento de ciertos elementos dentro del contexto de la inteligencia ambiental. Algunas de estas contribuciones son:

- El uso de soluciones basadas en servicios sobre la nube para dar soporte a la demanda de usuarios [PGSOCB18].
- El uso de sensores virtualizados dentro de las simulaciones [PCBGSP17].
- El análisis del comportamiento de los usuarios en entornos concretos, el cual ha demostrado cómo las multitudes tienden a interactuar en la simulación en comparación con el mundo real [SOCBGSP18].
- El uso de técnicas de reconocimiento de actividad humana para identificar las acciones del usuario dentro de un entorno monitorizado mediante sensores [GSCBPC16].

Las publicaciones que avalan esta Tesis se presentan en la Tabla 1.1. Las publicaciones se agrupan por tipo de evento: revista, congreso, workshop y demostraciones. De cada publicación se muestra el acrónimo del evento, el título del artículo y la referencia del mismo dentro de la Tesis.

²Repositorio sistema HAR: <https://github.com/mfcardenas/phat-sim-classificactor.git>

³Repositorio framework: <https://github.com/mfcardenas/colosaal-fcepevent.git>

⁴Repositorio de ejemplos: <https://github.com/mfcardenas/phat-sim-tester-examples.git>

Publicaciones	Cita
Revistas (1)	
- CSIS 2017 (Q4, FI: 0,613), Requirement engineering activities in smart environments for large facilities	[GSPACB17]
Congresos (8)	
- ICTSS 2018 (B), Testing Ambient Assisted Living Solutions with Simulations	[CBSP18]
- DS-RT 2018 (B), A Cloud Based Simulation Service for 3D Crowd Simulations	[PGSOCB18]
- UCAMi 2018, Using Graphs of Queues and Genetic Algorithms to Fast Approximate Crowd Simulations	[SOCBGSP18]
- CAEPIA 2018, Participatory Design with On-line Focus Groups and Normative Systems	[CBCGSP18]
- CAEPIA 2018, Learning protocols through practise with Microgrids	^a
- INTERACCION 2019, InfoState: Interactive Visualisation of the Environment using Ambient Sensors	[CBB18]
- RRI-SIS 2018, Developing RRI Practices: The Role of the ICT in Stakeholders, Knowledge Exchange and Co-creation Processes	[BMCB18]
- SMART-CT 2017, Virtual Development of a Presence Sensor Network Using 3D Simulations	[PCBGSP17]
Workshop (1)	
- COMSOTEC 2017, Combining tools for enabling complex systems simulation development	[UC317]
Foro (1)	
- IEEE WF-IoT 2019, Framework-supported Mechanism of Testing Algorithms for Assessing Memory and Detecting Disorientation from IoT Sensors	[GMIJ19]
Demos (1)	
- PAAMS 2016, Building Prototypes Through 3D Simulations (1st IBM Demonstration Award)	[GSCBPC16]

Tabla 1.1: Lista de publicaciones que avalan la tesis.

^aPublicado sólo en las actas del congreso.

1.4 Estructura de la Tesis

Este trabajo de Tesis está estructurado en 10 capítulos, siendo esta introducción el primero de ellos. El resto de capítulos se describen brevemente a continuación.

- **Capítulo 2: Estado del arte.** Describe los conceptos básicos y el estado del arte de la hipótesis abordada en esta tesis, a saber, soluciones AAL representadas por simulaciones en 3D y las técnicas de validación de este tipo de elementos de especificación.
- **Capítulo 3: Caracterización de la solución asistiva en la simulación.** En este capítulo se presenta la metodología propuesta para utilizar la simulación como un elemento de especificación en el desarrollo de un caso de estudio que representa una solución asistiva. Asimismo, se detallan las principales tareas del proceso de desarrollo del caso de estudio y los artefactos generados, con los que posteriormente se construye la simulación.
- **Capítulo 4: Entorno de pruebas de la simulación.** Una vez construidas las simulaciones, este capítulo indica el procedimiento a seguir para construir el entorno de pruebas utilizado para validar los requisitos de la solución asistiva mediante la simulación. Paso a paso, se explica el proceso que identifica los elementos de la solución que son necesarios para definir, ejecutar y analizar los casos de prueba

aplicables sobre la simulación. Finalmente, con base en los resultados obtenidos se muestra cómo validar la solución o, en su defecto, cómo verificar si el caso de estudio que representa la simulación corresponde con lo simulado.

- **Capítulo 5: Diseño del Framework.** Presenta el framework propuesto para la creación del entorno de pruebas que sirve de guía a los desarrolladores en el proceso de validación de las simulaciones.
- **Capítulo 6: Arquitectura del entorno de pruebas.** Con base en el diseño del framework, se describe la arquitectura cuyas funcionalidades se aplicarán tanto en la creación de las pruebas unitarias sobre elementos concretos de la simulación, como en la definición de casos de prueba complejos.
- **Capítulo 7: Implementación del sistema HAR.** Muestra el componente específico del framework encargado de validar las actividades físicas del usuario mediante el reconocimiento de las mismas. De dicho componente se muestran los elementos que lo conforman y la forma en que éstos se integran con la arquitectura propuesta.
- **Capítulo 8: Experimentación con el entorno de pruebas.** En este capítulo se detallan los experimentos llevados a cabo para mostrar el uso de la metodología y de la arquitectura propuesta. Por un lado, se construye un entorno de pruebas para analizar el comportamiento concreto de los elementos de la simulación. Por otra parte, un segundo entorno de pruebas valida el comportamiento del usuario dentro de la simulación, aplicando para ello técnicas HAR que describan los patrones asociados a la respuesta del usuario.
- **Capítulo 9: Pruebas en un entorno real.** La experimentación con el entorno de pruebas sobre la simulación se lleva a cabo sobre un entorno real con el fin de validar determinadas características en las especificaciones definidas en la simulación, concretamente el uso del sistema HAR.
- **Capítulo 10: Conclusiones y trabajo futuro.** Finalmente, se muestran las conclusiones obtenidas con la elaboración de esta Tesis y se presentan las diferentes líneas de trabajo futuro que han surgido a lo largo de todo el trabajo de investigación.

Capítulo 2

Estado del arte

Este capítulo se centra en el estado del arte en los sistemas AAL y en el uso de las simulaciones como especificación de soluciones basadas en este tipo de sistemas. En la Sección 2.1 se describe el contexto general de los sistemas AAL, haciendo especial énfasis en las técnicas de participación que promueven la cocreación de este tipo de soluciones. Las principales características de una solución de vida asistida, así como los elementos del dominio que son transversales en estos entornos, se presenta en la Sección 2.2. Posteriormente, en la Sección 2.3 se repasa el panorama actual que tiene el uso de las simulaciones en los escenarios de vida asistida, así como las técnicas actuales de validación empleadas para validar la solución. Finalmente, en la Sección 2.4 se muestra el uso que se le da a la simulación como especificación en una solución ambiental, las limitaciones que tiene su uso en los sistemas AAL y la forma en que determinadas técnicas de monitorización utilizadas en un contexto diferente, pueden encajar dentro del proceso de especificación y validación de la simulación como artefacto de la solución asistida.

2.1 Contexto general

Una solución asistiva dentro de un entorno inteligente, como su nombre lo indica, brinda servicios de asistencia adaptados a necesidades concretas del usuario. Estos servicios se basan en la interacción entre las personas, numerosos dispositivos que se comunican y cooperan entre sí y el entorno físico real. Para que esta interacción pueda darse, los dispositivos además de comunicarse entre ellos, deben identificar e informar las condiciones del entorno físico: redes de sensores, dispositivos móviles, actividades del usuario, entre otros.

Uno de los principales objetivos de la AmI [RF05] es apoyar la autonomía y mejorar la calidad de vida en las personas [ACRV13], especialmente de aquellas con algún tipo de vulnerabilidad que demandan determinada asistencia. Los sistemas AAL [RM13] son un buen ejemplo de entornos inteligentes [RAS08] que asisten a personas, cuya vulnerabilidad está relacionada con la edad y el envejecimiento, o con alguna discapacidad. Estos entornos promueven la creación y validación de prototipos que son parte de soluciones complejas que se aplican en la vida real para ofrecer la asistencia. Su objetivo principal, es adaptar las características del entorno para promover un estilo de vida independiente, cubriendo para

ello necesidades básicas del usuario como la administración de medicamentos, la detección de caídas, el monitoreo de enfermedades crónicas o el seguimiento de actividades físicas comunes, entre otras tareas.

Aunque estas tareas de asistencia generalmente las llevan a cabo personal especializado o cuidadores, los soluciones AAL con ayuda de los *living labs*, proporcionan alternativas que permiten adaptar los servicios del entorno, en función de las necesidades del usuario y de su interacción con el contexto. Los *living labs* [HK11] precisamente facilitan la investigación de experiencias asistivas y tecnológicas y permiten medir la interacción del usuario con el entorno [ENK⁺05]. De igual forma, en ellos es posible observar las iteraciones en el proceso de desarrollo [BKHS09] y promover la participación de las personas cuya misión coincide con los objetivos de la solución a construir [MW09]. Sin embargo, aunque el papel de los sistemas AAL ha sido significativo en el ámbito asistivo, uno de los inconvenientes que tiene este tipo de entornos es la demanda de inversión suficiente que permita su construcción y explotación.

Por otra parte, la participación e interacción del usuario en los sistemas AAL, son otras de las dificultades que afrontan estas soluciones. Los usuarios en este escenario son personas mayores que, en muchos casos, presentan limitaciones físicas propias de la edad que dificultan su participación directa. En la literatura existen diversos métodos que describen cómo involucrar al usuario en los procesos creativos de la solución. Sin embargo, algunos autores [MVK08] han hecho una revisión de los métodos utilizados en estos ambientes y se han dado cuenta que existe una especial inclinación hacia los métodos tradicionales, en contraste con el uso de técnicas participativas y de cocreación [Mul12].

Los métodos tradicionales como Web 2.0 [OB09], el diseño centrado en el usuario [VH05] o la creación de contenido por el usuario [PTSS10], entre otros, han hecho historia en los procesos de creación de soluciones. En el caso de los laboratorios AAL, estos métodos por si solos podrían no ser suficiente para explotar la infraestructura que se pone a disposición del usuario y la información que se genera alrededor de dicha relación. Las acciones del usuario como hábitos y patrones de comportamiento, limitaciones físicas propias de la edad, problemas de aversión tecnológica, la interacción con el entorno, entre otras, no han sido posible abordarlas con este tipo de metodologías tradicionales.

Actualmente no hay una metodología que estudie la interacción del individuo en un ámbito tecnológico asistivo y que refleje la dinámica del mundo real. Cada sistema AAL, por lo general hace sus propias adaptaciones de las metodologías tradicionales e intenta sacar provecho de ellas [CCPFR12]. En varios estudios se muestra cómo estas metodologías, a través de un enfoque participativo [MS09] y con ayuda de herramientas, técnicas de diseño y visualización de diferentes elementos gráficos (diagramas, modelos, vídeos), fortalecen el sentido crítico de los participantes [LNC⁺11], la auto observación y la reflexión. Según los investigadores, estas técnicas tienen el poder de probar, contrastar hipótesis y obtener información aún no conocida por el grupo de trabajo. Según [MS09], la cocreación [SF07] y el codiseño [HTC⁺15], se potencian alrededor de los VLL como técnicas que involucran a los usuarios como contribuyentes activos en el proceso de creación [SSF⁺17].

2.1.1 Técnicas de participación de los usuarios

La participación de los usuarios en el diseño de la solución AAL, se realiza tradicionalmente a través de técnicas sociales como los **Focus Group** [Mor97]. Esta técnica permite que los distintos usuarios organicen sesiones de trabajo en función de su perfil, de las funcionalidades que se quieren documentar o de la atención que debe recibir el usuario o paciente [Hen14]. En cada grupo de trabajo se designan representantes, los cuales se reúnen en un espacio determinado para discutir las preguntas planteadas por el moderador. El objetivo de estas sesiones es recopilar todos los comentarios posibles sobre la solución, para luego analizarlos e intentar identificar las características claves que debe tener la solución, el sistema y todo lo relacionado con los usuarios finales (interés por la solución, características de la tecnología, etc.).

Estas técnicas por lo general implican una variedad de costos tales como gastos de reclutamiento, tarifas de participantes y moderadores, elaboración de informes más transcripciones, costos de estímulos externos (para fomentar las discusiones), viajes, preparación y ocupación de las instalaciones, para citar algunos [Gre98]. La posibilidad de considerar una versión en línea de esta técnica resulta atractiva para reducir dichos costos y proporcionar un método de trabajo mucho más ágil, así como la posibilidad de alcanzar una participación más amplia. Sin embargo, hay un problema adicional relacionado con el resultado del grupo de enfoque y en cómo aplicarlo después. Si el resultado es un documento, existe el riesgo de que el documento no se aplique correctamente o incluso se olvide.

[KRA94] establece seis requisitos para una entrevista de grupo focal: involucra a personas; se lleva a cabo en una serie; los participantes son homogéneos y desconocidos entre sí; es un método de recolección de datos; los datos son cualitativos; y la discusión está enfocada. Según [TP05], los grupos de enfoque en línea basados en texto pueden cumplir con esos seis criterios. La disposición en línea podría ser compatible con tales características, pero otras son más difíciles de cubrir. Por ejemplo, la discusión en sí misma en un grupo focal se registra y cualquier persona es libre de participar. Los métodos en línea se pueden hacer a través de video/audioconferencia para dar una sensación similar, pero esto aumentaría los costos (los costos de las transcripciones son precisos).

Por otro lado, el *focus group* en línea puede ofrecer mayor anonimato que el *telephone group* [Gre98], pero puede restringir la participación si los usuarios no están acostumbrados a Internet (lo que, de hecho, es mucho menos frecuentes a medida que pasan los años). Aunque el grupo esté en línea o no, un *focus group* requiere una preparación previa donde los participantes deben ser seleccionados y la sesión debe planificarse cuidadosamente. En el contexto de las soluciones AAL, se asume que las preguntas preliminares estarán orientadas a comprender mejor el contexto del sistema AAL que se construirá, para señalar alguna característica deseable o para obtener información sobre características concretas del sistema.

2.2 Características de las soluciones de vida asistida

Los sistemas AAL sobresalen en el campo de la inteligencia ambiental por varios principios [DxMG⁺16], entre los que se destacan los siguientes:

- **Adaptabilidad:** las soluciones se adaptan al comportamiento del usuario y a la evolución del mismo [CCPFR12], utilizando para ello, la información que describe el estado del entorno. La información del contexto es necesaria para que los servicios de asistencia se brinden al usuario de la manera más natural posible. Según [Dey01], el contexto es “cualquier información que pueda usarse para caracterizar la situación de una entidad”, donde “una entidad puede ser una persona, un espacio, un lugar o un objeto que se considera relevante para la interacción entre un usuario y una aplicación, incluidos tanto el usuario como las aplicaciones mismas”. Partiendo de esta definición, y abordando ahora el argumento de la teoría de la actividad, el contexto está definido por las actividades y como tal “la actividad comprende un sujeto (la persona o grupo que realiza la actividad), un objeto (la necesidad o el deseo que motiva la actividad) y las operaciones (la forma en que se realiza una actividad). Los artefactos y el medio ambiente son vistos como entidades que median la actividad” [Gre01]. Como se puede apreciar en estas definiciones, la actividad es la unidad básica de análisis [Nar96] empleada para observar el contexto, de ahí que su uso se considere adecuado para analizar el comportamiento de los elementos que hacen parte de la solución..
- **Interacción:** gran parte del éxito de la asistencia se mide por el grado de interacción del usuario tanto con el entorno como con la propia solución [KBR⁺07]. Por ello, muchos esfuerzos se centran en utilizar técnicas de ordenador que permiten llevar a cabo tareas como el reconocimiento de la actividad, el posicionamiento de los individuos y objetos, la detección de emergencias y los sistemas de mensajes de alertas, con el fin de describir el comportamiento del usuario de cara al uso de la solución. Dentro de estas técnicas, se destaca el reconocimiento de actividad humana o HAR [VNK15] por ser el responsable de identificar patrones en la actividad humana, que ayudan en la interacción entre el usuario y la asistencia. Los sistemas HAR no son específicos de las soluciones AAL, estos se utilizan en diferentes contextos para identificar la actividad del usuario. Estas técnicas permiten reconocer la presencia de gestos, acciones, eventos, comportamientos o interacciones realizadas por los usuarios, para posteriormente ejecutar protocolos de actuación que incluyen determinadas tareas. Algunos de los elementos usados para aplicar esta técnica son los sensores inerciales, exactamente los de aceleración y giroscopio. A diferencia de los sensores ambientales que generan datos numéricos concretos, un sensor inercial ofrece datos en forma de series temporales, los cuales facilitan la representación de las actividades físicas desde diferentes perspectivas: una actividad, un conjunto de actividades, acciones concretas o gestos específicos que componen la actividad en sí.
- **Heterogeneidad:** la tecnología y los sistemas utilizados no son específicos de los entornos AAL. Por lo general, son aportaciones independientes de otros

proveedores que cuentan con su propia tecnología, su propio conocimiento y sus propios costes [MWP⁺14]. En el caso del uso de los sistemas HAR, esta característica predomina debido a las muchas técnicas de procesamiento y aprendizaje existentes, lo cual a su vez, dificulta la rápida elección de aquella técnica que más se ajusta a las necesidades del usuario. No hay una especificación clara que indique cómo proceder para (a) seleccionar los atributos a medir (datos de sensores [PEK⁺06]), (aceleración y ritmo cardíaco [TIH⁺07]), (acelerómetro triaxial [CYL⁺08]), (actividad anormal [YYP08]), (técnicas de filtrado [HMK09]), (técnicas de transformada discreta [HJ09]), (teléfonos móviles [RMB⁺10]), (b) usar un consumo de energía bajo, no invasivo y que proporciona datos suficientes [TIH⁺07], (c) posicionar los sensores en el entorno [PCBGSP17] o en el usuario [FRN⁺10]), (d) obtener datos de manera óptima [BI04], (e) usar el contexto [DA⁺00] para ofrecer un mayor realismo y (f) entrenar el sistema para nuevas actividades y nuevos usuarios [LRE10].

- **Tecnología de apoyo:** los sistemas AAL utilizan como base distintas tecnologías de apoyo que potencian la asistencia, lo cual los hace altamente compatibles con las tecnologías existentes [MWP⁺14]. Según su tipo [KPP⁺12], es posible encontrar distintas aplicaciones útiles en determinados problemas [Ros04]: recoger datos e identificar patrones, generación de sistemas de alertas, recordatorios y recomendaciones sensibles al contexto, aplicaciones para generar diálogos con Procesamiento de Lenguaje Natural (PLN), interfaces de usuarios y personalización, sistemas de razonamiento basados en el conocimiento (sistemas recomendadores, ontologías, algoritmo de decisión, entre otros), simulaciones y sistema de agentes.
- **La población mayor como público objetivo:** al ser el principal usuario de estos sistemas, sus diferencias culturales, su estado de salud y los distintos problemas de motricidad y memoria que pueden tener debido a la edad, son condiciones que hacen aún más complejo identificar los requisitos que definan la forma adecuada de interactuar [RM13]. Una mala especificación de la asistencia no solo puede generar un mal funcionamiento del servicio [BSC⁺11], sino que puede ser la causa del rechazo de la solución por parte del usuario. En este aspecto, ya no solo es suficiente que la solución interactúe con el usuario de forma correcta, también se debe tener en cuenta problemas que son propios del dominio como la interacción. Algunos de estos problemas son la *no invasividad* [ZPR08], los posibles casos de *aversión tecnológica* comunes en personas mayores, la *privacidad* de la información sensible y las posibles reacciones o *respuestas de usuario* que son producto de la evolución positiva o negativa de sus limitaciones físicas.

Aunque los principios parecen estar claramente definidos [DxMG⁺16], no es posible encontrar un mercado maduro y estable que, además de promover y facilitar este tipo de sistemas, consolide de alguna forma estándares de desarrollo [CCS⁺17]. No existe un modelo de proceso que permitan a nuevos desarrolladores crear soluciones robustas que conozcan su futuro a largo plazo, su viabilidad económica, el impacto sobre la economía

del usuario y las necesidades de integración con otras tecnologías. Bajo este escenario, el principal reto a la hora de construir una solución, consiste en definir una metodología que permita trasladar estos principios a la solución final.

2.2.1 Elementos del dominio de la solución

En el dominio asistivo, existe una serie de elementos que son transversales o que hacen parte del dominio específico de la asistencia. Estos elementos no están asociados a la solución propiamente dicha, sino más bien a situaciones que son recurrentes y que afectan al usuario y a su entorno. En el ámbito de las soluciones asistivas, se describen las siguientes situaciones recurrentes:

- **Tecnologías invasivas:** la tecnología juega un papel importante en el ámbito de las soluciones ambientales y de asistencia [BKK⁺10]. Se puede afirmar que cuanto más tecnología haya inmersa en el contexto, mayor es la posibilidad de obtener información suficiente para optimizar los procesos de asistencia y respuesta al usuario. Sin embargo, la cantidad de dispositivos en el medio y sobre el mismo usuario, puede llegar a tener una connotación negativa que impida aprovechar los beneficios de dichas soluciones y hacer uso de ellas [ZPR08]. Parte del éxito de una solución asistiva radica en la facilidad con la que la tecnología pasa desapercibida para el usuario. Cuanta menos conciencia tenga él de los medios tecnológicos utilizados para asistirle, menor será el riesgo de que se sienta invadido. En el caso de las personas mayores [BFF⁺15], la “*invasividad*” tecnológica puede tener connotaciones mucho más negativas, debido a la poca cercanía de las personas con la tecnología actual.
- **Aversión tecnológica:** el crecimiento frenético de las nuevas tecnologías lleva implícito una serie de cambios sociales que evidencian distintos problemas humanos y sociales en la población, que afectan negativamente la implementación de soluciones asistivas [LGVBP06]. En la población mayor por ejemplo, los cambios generados por las nuevas tecnologías están induciendo a la prevención y al rechazo de las herramientas tecnológicas, sobre todo cuando los usuarios no cuentan con una cultura tecnológica previa. Dentro de estos problemas sociales se encuentran los trastornos de ansiedad o “*tecnofobia*” como suele llamarse en este caso, el cual tiene que ver con el concepto de “*tecnoestrés*” [WR97] y que está directamente relacionado con los efectos psicosociales negativos del uso de las TIC [Bro84].
- **Privacidad:** las soluciones asistivas y las tecnologías de apoyo utilizadas en cualquier contexto, requieren de datos del usuario con una calidad suficiente para que sea útil y contribuya con la eficiencia del servicio que se quiere dar al mismo usuario [TMS⁺07]. En el contexto de la salud en personas mayores [MRB06], cuando por ejemplo la asistencia se utiliza para la monitorización [CP03] del usuario, la implicación de información sensible del usuario adquiere connotaciones mayores en lo que respecta a la seguridad [LB09]. Para garantizar que la solución sea usada y adoptada por el usuario, estas implicaciones de seguridad deben ser tenidas en cuenta para evitar

confusiones que lleven al usuario a rechazar la solución [LHBC⁺11]. Por otra parte, en el caso del adulto mayor, este no siempre es quien decide usar la tecnología. Si en la vivienda otras personas conviven con él, por ejemplo, familiares, amigos o cuidadores, estos también podrán participar en la decisión de aceptar la tecnología utilizando la privacidad como un criterio.

- **Respuestas del usuario:** cuando se construye una herramienta para el usuario, es fundamental que dicha herramienta se adapte no solo a los requisitos funcionales que la definieron sino también a las características del usuario que puedan afectar su correcto funcionamiento [Ste12]. En el caso de las soluciones asistivas, existe la misma necesidad pero con varios atenuantes relacionados con las limitaciones propias de edad [FCR⁺18]. El rol que asumen los usuarios ante el uso de la tecnología, es un factor que se debe tener en cuenta al momento de definir las tecnologías de apoyo que se vincularán a la solución [Röc13]. Un usuario con un rol activo, demandará mayor interacción con la solución mientras que con un rol pasivo, este le dará potestad a la aplicación para que tome decisiones por él [OPS13]. Además de los roles, es necesario establecer protocolos de actuación para los casos en que el usuario reaccione o responda de forma inesperada ante la solución. No es posible abarcar todas las posibles respuestas del usuario, pero sí es viable determinar las respuestas más comunes que se pueden esperar en función del perfil.
- **Problemas de movilidad:** antes de considerar el uso de una solución asistiva como apoyo para un adulto mayor, es importante con ayuda de profesionales, identificar el estado físico de la persona y el nivel de independencia que tiene a la hora de llevar a cabo actividades de la vida diaria. Los expertos consideran que en un adulto mayor, acciones como caminar, sentarse, levantarse de una silla, inclinarse, incorporarse o girar el cuerpo, pueden revelar claros indicios del grado de movilidad que tiene el usuario y de su capacidad de llevar a cabo actividades diarias. La actividad de caminar puede servir de base para recoger información sobre la movilidad del usuario. La velocidad de la marcha, el tiempo que tarda en levantarse de la silla y su capacidad para pararse con los pies alineados (con un pie delante del otro, lo que permite evaluar el equilibrio), son factores independientes que pueden predecir la capacidad de desempeñar actividades instrumentales cotidianas. Los problemas de movilidad que se pueden modelar en la simulación, son los que están precisamente relacionados con la marcha, concretamente con los cambios anormales y normales de ésta. Los cambios anormales muestran aquellas variaciones físicas que podrían indicar algún tipo de trastorno, mientras que los cambios normales reflejan aquellas variaciones en la actividad física propias de la edad del individuo.

2.3 La simulación en las soluciones de vida asistida

El desarrollo de los sistemas AAL es todo un desafío. Como se mencionó en el apartado anterior, los *living labs* son altamente costosos de implementar debido a, entre otras cosas, la necesidad de realizar mucha experimentación, la alta disponibilidad de hardware, la

incertidumbre al momento de intentar acotar funcionalidades, la construcción de sistemas para la monitorización a corto y largo plazo, la necesidad de incorporar al usuario final en el desarrollo para tener una retroalimentación temprana y, no menos importante, a la complejidad de preservar la dignidad, privacidad y la integridad del usuario final. Al llevar el proceso de desarrollo de la solución AAL al ordenador [PCBGSP17], son muchos los beneficios que se logran, principalmente la reducción de costos, ya que el entorno de desarrollo no es exigente y los ordenadores actuales para videojuegos son suficientes. Otro factor que se ve beneficiado es la posibilidad de experimentación. Esta filosofía de desarrollo es lo que se conoce como laboratorio viviente virtual o VLL [GSCBPC16]. Esta transferencia existe en ambas direcciones, no solo consiste en llevar el problema al ordenador [GSCS17]. Del entorno simulado también es posible definir y construir una solución real.

La simulación aparece como una forma de desarrollar “virtualmente” las soluciones AAL. El uso de escenarios en 3D, permiten “visualizar” de forma temprana la solución, evitando así la necesidad de experimentar con entornos reales directamente. La simulación en 3D reproduce las situaciones de la vida cotidiana con suficiente realismo. Para ello, recrea a las personas y su comportamiento, a la red de dispositivos o sensores necesarios para la monitorización, los sistemas HAR y al entorno físico. Adicionalmente, permite simular la interacción entre todos estos elementos para poder analizar los posibles problemas que puedan existir en dicha solución. El objetivo es garantizar que si la solución asistida funciona en la simulación, ésta funcionará igual en escenarios similares del mundo real.

2.3.1 Validación de las soluciones

El principal motivo de fracaso en los proyectos software es no determinar los requisitos a tiempo [SLKC01]. Al igual que en cualquier otro proyecto, para comprender cómo lograr el comportamiento esperado en un ambiente AAL, es necesario que haya mucha interacción con el usuario final y que exista suficiente experimentación para garantizar la aceptación, seguridad [TJ12] y conformidad de la asistencia. Si existen ambos elementos en el proceso de definición, la ingeniería de requisitos [Pre08] y los productos de trabajo definidos por el modelo de proceso de software elegido, podrán ser suficientes para llevar a buen puerto el servicio de asistencia [ENK⁺05]. En el caso que uno de estos elementos faltase, el riesgo de que el usuario final rechace la solución aumenta considerablemente. Al hablar de sistemas AAL, no siempre hay total disposición del usuario en el proceso de desarrollo [SF07]. Cuando los usuarios son personas mayores, en muchos casos la experimentación no es una alternativa viable, además que podría ser muy costosa al requerir prototipos elaborados que cubran las características de la solución.

Como se ha descrito en la Sección 2.2, en un sistema AAL [NAA09] es posible encontrar desde piezas de software [HEN⁺09] basadas en la inteligencia ambiental hasta múltiples arquitecturas [RZ11] compuestas por dispositivos integrados en el entorno, cuyo propósito es ofrecer un servicio específico (administración de la medicación [SKR⁺11], detección de caídas [AAW⁺07], sistemas de emergencia [ESSH06] y generación de alertas [PBC⁺03], monitoreo de las actividades diaria [NAA09], sistemas de avisos a amigos, familiares y

personal médico [MRCJ01], entre otros). Aún con esta variedad de componentes, sigue siendo importante que la solución ofrezca una interacción del usuario con el sistema lo más natural posible, pudiendo incluso ser imperceptible para él [BYH⁺18]. Esta característica es la que permite luego que la solución se adapte a las necesidades del usuario. Para poder conseguirla, las especificaciones o requisitos que definen la solución, desde un principio de su concepción deben estar enfocadas en características del usuario como su condición física, mental y emocional.

Así como no existe una metodología que describa de qué forma construir la solución asistiva, tampoco existen pautas que describan cómo se debe validar dicha solución, especialmente con el uso de *living labs* tipo AAL. Algunos trabajos se centran en la evaluación de aspectos concretos del proceso de creación. [DMKJ⁺10] describen por ejemplo, cómo evaluar la experiencia de usuario desde el punto de vista técnico, utilizando métricas sobre el uso desde dispositivos móviles. Otros trabajos más amplios como el de [SMDM11], proponen una serie de principios que son producto de la experiencia previa de los autores. El objetivo de dichos principios es ofrecer una *guía* para recopilar, generar y entender las necesidades del usuario utilizando los *living labs* [Stå08], como parte del proceso de participación. Una variante de este mismo trabajo [BKHS09], se ha propuesto como herramienta que puede emplearse en todo el ciclo de vida de la solución. Esto último se acerca a lo que se propone en este trabajo, principalmente porque se enfoca en el usuario y en su participación dentro del proceso de identificación de requisitos. Sin embargo, sigue siendo necesario disponer del *living lab* para recoger toda esta información, lo cual a su vez requiere que el usuario participe directamente en las experiencias que se lleven a cabo en estos entornos.

Existen trabajos enfocados sobre los sistemas AAL [NFS⁺09] pero no distan mucho del uso convencional que se da a los laboratorios: probar la tecnología, mostrarla a los usuarios y conocer su opinión [TdRD⁺11], realizar ejercicios de movilidad [KBRSG10]. No hay duda que todos estos trabajos hacen una contribución importante al área de la vida asistida, pero todos ellos se centran en las ventajas que ofrecen los propios *living labs*. Las pautas y recomendaciones identificadas hasta el momento, podrían ser igual de útiles en entorno VLL [SSP15]. Sin embargo, en este contexto tampoco ha sido posible encontrar experimentos similares que describan algún mecanismo que valide uno de los elementos más importante del proceso de definición de la solución: *la simulación*.

2.4 Uso de la simulación como especificación

Algunas contribuciones recientes están intentando trasladar el esfuerzo de la cocreación al ordenador, abordando tanto el modelado del problema como su correspondiente solución asistencial mediante el uso de simulaciones 3D [GSCBPC16]. La simulación en 3D es más fácil de entender que las especificaciones técnicas y no requiere habilidades especiales por parte de los usuarios. Esto facilita sobre todo la comunicación en equipos multidisciplinares.

Existen ejemplos en la literatura de plataformas de simulación donde es posible representar y reproducir un espacio físico y desplegar sobre él, sensores que generan

resultados muy similares a los reales (ver Figura 2.1). Ubiwise [BV03] por ejemplo, es una herramienta en la que es posible poner a prueba conceptos sobre la computación ubicua. Esta herramienta utiliza dispositivos simulados y representaciones virtuales de los objetos físicos, pero no dispone de personajes o avatares simulados. El usuario que usa la herramienta es quien, en primera persona, recorre la escena a través de un avatar. El no poder modelar el comportamiento del usuario es una limitación de la herramienta con respecto a lo que se propone en este trabajo. La simulación propuesta, permite modelar el comportamiento esperado de los personajes y su interacción con los diferentes dispositivos. De igual forma, es posible añadir nuevos sensores que, una vez programados, se asocian a la simulación enriqueciendo su funcionalidad.

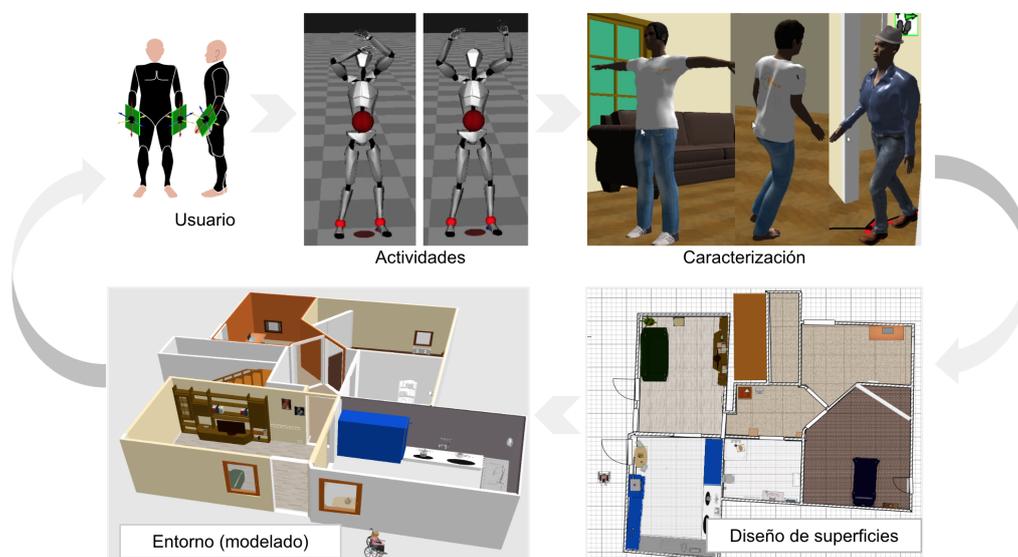


Figura 2.1: Especificaciones incluidas en la simulación.

Ubiwise cuenta con una versión más reciente llamada Ubiksim [GVCS⁺12]. Esta nueva variante ofrece mayor realismo en los escenarios, cuenta con mejores motores de representación de escenas e integración con dispositivos externos. A diferencia de su predecesor, Ubiksim si soporta la simulación de personajes cuyo comportamiento está programado con máquinas de estado jerárquico. La principal desventaja de Ubiksim es que utiliza el motor Sweet Home [Puy05] por lo que tanto el movimiento de los personajes como la manipulación de objetos está en *2D computer graphics* (2D) aunque parezca 3D. La característica 3D es importante cuando se quiere medir la interacción del usuario con objetos del entorno, los cuales podrían estar posicionados a alturas diferentes con respecto al usuario. En contraste con las simulaciones de Ubiksim, la simulación utilizada en este trabajo funciona en 3D y permite realizar operaciones típicas en tales entornos, como la distribución de objetos en la escena y selección de rayos para identificarlos.

Otra herramienta para la visualización y creación de aplicaciones controladas por eventos mediante middleware es *Generic Location event Simulator* (GLS) [SC02]. En esta plataforma, la simulación reproduce en un escenario 2D, los mensajes generados por los sensores con el fin de que otras aplicaciones puedan conectarse a ellos y hacer pruebas dentro de la simulación. En comparación con GLS, las simulaciones utilizadas

en este trabajo, ofrecen la ventaja de un mayor realismo mediante el uso de 3D, al tiempo que conservan la capacidad de emular el comportamiento de los sensores. Además, los personajes pueden agregarse con su propio comportamiento e interactuar con los dispositivos y aplicaciones.

3dSim [NSK05] es un sistema para la creación rápida de prototipos de aplicaciones de inteligencia ambiental. Esta herramienta utiliza la simulación en 3D para crear entidades en el entorno y conectar algunas de ellas con el mundo real. Fue concebido para construir salas de reuniones y dar órdenes a elementos conectados. En comparación con 3DSim, las simulaciones utilizadas en este trabajo, tiene la ventaja de permitir definir y modificar el comportamiento de los participantes.

En este trabajo, se utiliza la plataforma *Ambient Intelligence Development Environment* (AIDE) [AID14] para generar las simulaciones en 3D que interesan a esta investigación. La herramienta permite al desarrollador describir el comportamiento de los personajes simulados gráficamente y añadir al escenario cierto tipo de sensores como luminosidad, ruido, presencia, inerciales (acelerómetros), entre otros. El resultado obtenido se considera un VLL [GSCBPC16], porque tiene como objeto proporcionar al desarrollador un escenario simulado donde puede estudiar cómo funciona la solución asistiva y la forma en que los usuarios interactúan con la simulación.

2.4.1 Validación de la simulación

Aunque el uso de los VLL en el proceso de creación de soluciones asistivas representa un avance, utilizar la simulación como parte de la especificación ha generado nuevos retos y problemas: *¿cómo validar que la simulación cumple con los requisitos establecidos? ¿cubre la simulación el problema estudiado? ¿Cómo validar que la solución de asistencia en la simulación es la correcta? ¿Cómo validar que los usuarios en la simulación se comportan como se espera? ¿y el entorno?,* y no menos importante *¿Cómo garantizar que cada nueva iteración de la simulación 3D es coherente y constante con las versiones anteriores?*

Según las características de la solución asistida y de los entornos AAL, la simulación debe disponer de un conjunto considerable de sensores que permitan obtener la mayor cantidad de datos posibles. Estos datos describirán el comportamiento no solo del usuario sino de su entorno en general y para descifrar dicho comportamiento, se necesitará de algún mecanismo o técnica que permita utilizar esta información para comprobar que lo ocurrido en el entorno simulado se corresponde con lo que debería recoger la especificación propuesta en dicha simulación. Por otra parte, como en las simulaciones se ejecutan distintas acciones o sucesos sobre una misma línea de temporal, cualquier técnica empleada debe implementar algún tipo de contabilidad de tiempo mientras analiza los eventos producidos por la simulación. Evento [Luc02] es el término que se acuña a cualquier suceso que ocurra en la simulación y es por ello que la mejor forma de analizar este flujo de sucesos es mediante *Complex Event Processing* (CEP). Este paradigma es muy utilizado para monitorizar las aplicaciones basadas en eventos [CFS⁺14], especialmente aquellas que se caracterizan por ser descentralizadas [LF98], altamente escalables y muy dinámicas [CHKR14]. Cabe destacar que CEP no se utiliza en el contexto de la validación de los elementos que generan los eventos sino para la monitorización y generación de

alertas por anomalías detectadas [CM12]. En el contexto de la simulación, no existe una metodología que establezca cómo validar el comportamiento simulado, lo que sí se aplica y podría ser útil, es la monitorización de dichos entornos para identificar patrones de comportamiento no esperados.

Para definir con precisión qué tipo de comportamiento se espera en un solución asistida, se propone el uso de simulaciones y que dichas simulaciones sean validadas de alguna forma. Para este propósito, se necesita un conjunto de tareas que permitan obtener los diferentes requisitos que describan la solución. La interacción entre los ingenieros y los expertos se describe en el capítulo siguiente. Los ingenieros producen la simulación que captura el comportamiento esperado del entorno, mientras que los expertos validan las simulaciones. La especificación de la simulación es realizada por los ingenieros. Estos son quienes define el entorno, establece un conjunto de posibles comportamientos de los usuarios y enumera el total de instancias de estos comportamientos. En paralelo a esto último, asigna una marca de tiempo que indica cuándo debe ocurrir cada comportamiento y el número de instancias que se esperan del actor. Finalmente, y siempre que sea posible, estima qué tipo de tecnología de apoyo puede ser útil para el usuario en el escenario que está simulando. Todo esto permite definir escenarios donde los actores simulados utilizan el entorno asistivo en diferentes momentos y circunstancias. Los usuarios simulados serán los responsables de interactuar con el entorno, de usar la asistencia y de apoyarse en la tecnología de apoyo propuesta, como se muestra a continuación:

- **El entorno:** comprender la distribución física y los espacios disponibles de la vivienda del usuario.
- **El comportamiento de los individuos:** documenta y describe los comportamientos que se esperan del usuario. Aunque esto es algo muy específico de las ciencias sociales, los métodos clásicos, como la grabación de vídeos, la realización de encuestas y entrevistas pueden ser útiles para recopilar información. De igual forma es necesario un análisis posterior para evaluar la información y encontrar pistas sobre qué actitudes son más comunes en el usuario y qué estímulos resultan más favorables para lograr la aceptación y el comportamiento deseado.
- **La tecnología de apoyo:** a menudo, diseñar una solución asistiva implica usar tecnologías de apoyo que brinden o enriquezcan la asistencia al usuario. Cuando esta tecnología se integra en el entorno, las especificaciones de la solución se restringen aún más, porque es necesario establecer cómo se debe desplegar dicha tecnología y de qué forma el usuario tendrá que interactuar con ella.

En este trabajo se asume que la simulación representa los requisitos que definen la solución asistiva. Toda la información capturada en la etapa de especificación se representa computacionalmente mediante las simulaciones, lo cual inicia la fase de especificación de requisitos. Si estas simulaciones por computadora describen el comportamiento del usuario y el de su entorno cuando se demanda la asistencia, entonces podrían usarse como escenario de pruebas para evaluar las posibles acciones del usuario y la secuencia de estímulos que podría brindar la solución como respuesta.

Capítulo 3

Caracterización de la solución asistiva en la simulación

Este capítulo describe la metodología propuesta en la que se utiliza la simulación como un elemento de especificación en el desarrollo de soluciones asistivas. En la Sección 3.1, se hace una breve descripción del uso de la simulación en el ámbito asistivo. El alcance real que se da a la simulación dentro del ciclo de desarrollo de la solución se muestra en la Sección 3.2. La forma de modelar los elementos que componen la solución asistiva en los escenarios simulados, se presenta en la Sección 3.3. Una vez identificada la solución, en la Sección 3.4 se indican los pasos necesarios para crear los casos de estudio que representan la solución, describiendo todos los elementos que hacen parte de ella. Finalmente, en la Sección 3.5 se muestra cómo, teniendo en cuenta la relación entre los elementos de cada uno, se trasladan las características del caso de estudio a la simulación para que esta pueda ser utilizada en representación de la solución.

3.1 Introducción

Para utilizar la simulación en el proceso de desarrollo de la solución AAL, se parte de un proyecto en el que toda o parte de su ingeniería de requisitos dentro del modelo de proceso correspondiente, se quiere soportar a través de las simulaciones en 3D. Partiendo de esto, todos los esfuerzos destinados a generar las especificaciones de requisitos, podrán ser empleados de igual forma para construir la simulación que describa dichos requisitos y que represente el comportamiento de la solución asistiva. En este tipo de soluciones, el aporte que ofrecen estas simulaciones en 3D dentro del modelo de proceso se centra en dos contextos específicos:

- Como una especificación visual de los requisitos, la cual evoluciona a medida que la fase de identificación de requisitos avanza.
- Como un artefacto que permitirá realizar las pruebas de validación necesarias para garantizar que la solución asistiva se ajusta a los requisitos del cliente.

Independientemente del contexto de uso, la simulación estará presente en todo el proceso de ingeniería de requisitos. Esta característica se considera un aspecto positivo

porque no obliga al equipo de ingenieros a implementar técnicas adicionales para construir la simulación. La misma fase de especificación de requisitos es utilizada para definir los elementos que conforman la estructura de una simulación.

3.2 Alcance de la simulación

La simulación se puede emplear especialmente en dos fases concretas del ciclo de vida del software: en la fase de definición de la solución y en la fase de pruebas del sistema o implantación de la solución. Según el alcance y uso que se le quiera dar, se han establecido las pautas necesarias para trasladar las características de la solución asistiva a la simulación. En función de esto, el objetivo es usarla como una representación discreta de la solución en fases como la definición de requisitos o como un elemento de pruebas en un escenario de construcción, antes de su implantación en un entorno real.

Como se observa en la Figura 3.1, la simulación cubre los requisitos específicos de cada fase y es utilizada para definir los entornos de pruebas correspondientes. En la fase de definición, la simulación se utiliza como medio para establecer las pruebas de validación y verificación de requisitos y, al mismo tiempo, permite a los desarrolladores establecer sus pruebas unitarias.



Figura 3.1: Entorno de pruebas basado en el marco de desarrollo de las simulaciones.

La iteración de las simulaciones entre fases es otro factor que se tiene en cuenta junto con el alcance de los entornos simulados. Al momento de definir un entorno de pruebas, tanto la validación y verificación de los requisitos como las pruebas unitarias del equipo, deben garantizar que entre las distintas iteraciones de los escenarios simulados exista correspondencia, sobre todo con aquellas versiones previas que hacen parte del control de cambios funcionales.

El objetivo de esto último es garantizar que no se pierdan por el camino cambios importantes en la solución que los distintos interesados hayan podido añadir o identificar. En la fase de solución real, sobre la simulación se aplican las pruebas de sistema pertinentes para garantizar que lo simulado representa la solución asistiva. En este caso, la validación irá enfocada a evaluar el comportamiento de la solución respecto a las características que son importantes para su implementación, tales como la configuración del entorno, funcionamiento de la tecnología de apoyo, red de sensores, entre otros, (ver Figura 3.2).

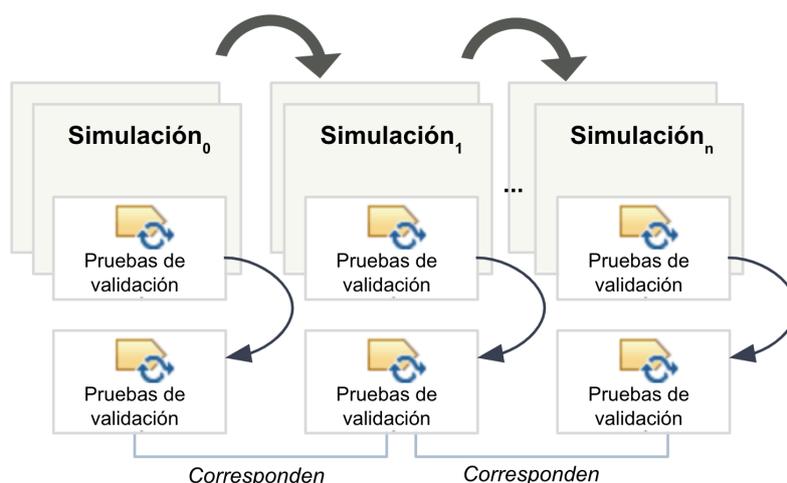


Figura 3.2: Correspondencia entre las distintas iteraciones de las simulaciones.

3.3 La solución asistiva

En el proceso de representar una solución asistiva en la simulación, es necesario identificar previamente tres elementos clave que componen la solución: el *usuario*, el *entorno* y la *tecnología de apoyo*.

- **Usuario:** es quien demanda las necesidades que debe cubrir la solución asistiva. Para ello, es necesario representar su estado, el posicionamiento dentro del entorno, su desplazamiento sobre el escenario, los posibles problemas de movilidad que puede padecer y las acciones o actividades físicas que lleva a cabo normalmente.
- **Entorno:** es el espacio físico que cubre la solución asistiva, el cual dispone de los medios necesarios que ayudan a describir las condiciones físicas alrededor del usuario. Asimismo, de él se describe su estado y los elementos que existen en él, identificando los medios necesarios para obtener dicha información, concretamente los sensores y actuadores.
- **Tecnología:** con la información contextual del usuario y del entorno, se utilizan tecnologías asistivas o de apoyo para ofrecer la asistencia que aumenta, mantiene o mejora las habilidades presentes en el usuario, compensando de esta forma las limitaciones que puedan existir en relación con la solución propuesta. Tales tecnologías pueden ser herramientas software que controlan los servicios y elementos hardware que brinden ayuda al usuario para acceder a la asistencia.

Para esta Tesis, es importante que el apoyo o la asistencia al usuario se base principalmente en el uso de **tecnologías asistivas** o de apoyo, las cuales estarán representadas por dispositivos (hardware) o programas (software) que bien son acoplados al entorno o que el propio usuario lleva consigo.

3.3.1 Modelado de la solución asistiva

Para usar la simulación como una descripción de la solución asistiva, se modela el comportamiento de los distintos elementos de la solución: “*usuario*”, “*entorno*” y

“tecnología”. Con los modelos de comportamiento definidos, se construyen tres tipos de escenarios que se utilizarán para analizar cada problema:

1. El *primer tipo de escenario* representa una situación típica o cotidiana para el usuario o el entorno, en donde no ocurre ninguna anomalía que requiera algún tipo de asistencia.
2. El *segundo tipo de escenario* identifica y describe un problema que padece el usuario como consecuencia de su condición física o por las características del entorno durante un escenario como el 1.
3. El *tercer tipo de escenario* muestra la solución asistiva que se puede dar al usuario partiendo de un escenario como el 1, que pueda desenvolverse como el escenario 2.

Por cada uno de estos escenarios se construye una o varias simulaciones que los representen. No necesariamente se debe definir una única simulación en cada caso, en ocasiones será mucho más práctico descomponer el problema en situaciones simples o concretas que sean más fáciles de analizar. Por otra parte, independiente del número de simulaciones que se construyan, en todas ellas siempre se debe mostrar toda la información posible tanto del usuario y su entorno como de la interacción entre ambos.

Como se observa en la Figura 3.3, a partir del *primer tipo de escenario* se crea la **simulación típica**, la cual describe cómo es el escenario idóneo tanto para el usuario como para el entorno sin ningún tipo de apoyo tecnológico. En esta simulación, se muestra al usuario llevando a cabo actividades cotidianas sin ninguna dificultad o impedimento y sin demandar asistencia alguna. El entorno, por su parte, describe un estado idóneo para el usuario, limitado a un espacio físico controlado lo más parecido posible a su entorno real (una habitación, la cocina, el salón o la vivienda en general).

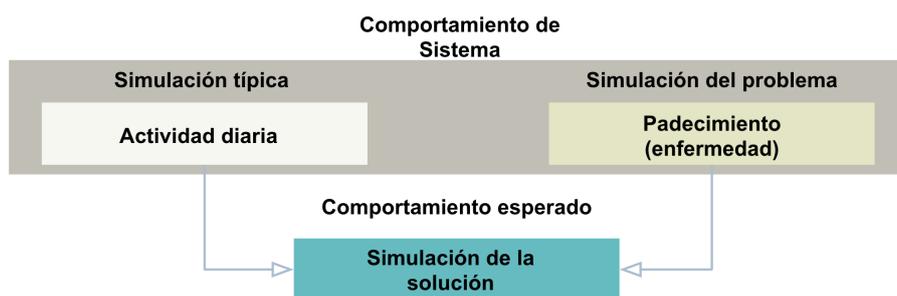


Figura 3.3: Representación de los escenarios del sistema.

Con ayuda del *segundo tipo de escenario* se crea la **simulación del problema**, describiendo así un escenario similar al anterior pero con la problemática que tiene el entorno, un usuario o ambos. Esta simulación tiene como objetivo permitir la observación y el análisis de la situación que impide al usuario realizar alguna actividad cotidiana. En este caso tampoco hay presencia de tecnologías de apoyo porque lo que se busca es analizar sólo la problemática del usuario y el impacto que ésta tiene sobre el entorno. El *tercer tipo de escenario* genera la **simulación de la solución**, la cual describe de qué forma se llevará a cabo la asistencia al usuario, bien sea mediante servicios que se

adecuen al entorno o bien a través de algún tipo de tecnología de apoyo, que mejore las capacidades de usuario o facilite su interacción con la solución.

3.3.2 Elementos para construir la simulación

A continuación se indican los componentes de los que se partirá para trasladar la solución asistiva a la simulación: el *caso de estudio* y las *pruebas de la solución*.

- **Caso de estudio:** es el elemento de partida para construir la solución. En él se lleva a cabo el proceso que captura los escenarios de interés para posteriormente crear las simulaciones correspondientes. En este proceso, se recogen todas las especificaciones que describen las características que tendrá la solución, así como la forma en que se modelará el comportamiento de los distintos elementos. Otros aspectos que abarca son los detalles estéticos y de apariencia del entorno simulado, las acciones o actividades que ocurren en la simulación, los detalles técnicos que describen la tecnología asistiva que dará asistencia, la configuración y parametrización de dicha tecnología, entre otros.
- **Pruebas de la solución:** recogen la especificación de las pruebas que se deben llevar a cabo sobre la simulación, incluyendo las pautas que describen a nivel de código, cómo se podrían implementar dichas pruebas. En función de la fase donde se ejecuten, éstas se organizarán en dos grupos:
 - *Pruebas de validación:* los requisitos del *escenario* y el resultado de su(s) *simulación(es)* son comparados para corroborar que lo simulado se corresponde con lo indicado en el caso de estudio. Si estas pruebas son correctas, se puede asumir que la simulación cumple con los requisitos establecidos y por ello afirmar que representa el problema que se está analizando en el caso de estudio. En el caso contrario, y basándose en aquellas pruebas puntuales donde han fallado las validaciones, será necesario revisar en detalle tanto el escenario como la simulación para identificar el origen de las posibles inconsistencias dentro del caso de estudio.
 - *Pruebas de la solución:* estas pruebas se utilizan para comprobar si los datos generados por la asistencia en la simulación, responden a las necesidades que se quieren cubrir. Para ello, la simulación proporciona mecanismos que permiten a programas externos acceder a los distintos elementos interactivos de la solución tales como sensores, actuadores, usuario, entre otros. A través del comportamiento de estos elementos, se valida que cumplan con los requisitos funcionales establecidos para dar la asistencia o contribuir con ella. Si el resultado de estas pruebas es correcto, la configuración general de la solución (planos, parametrización, características del entorno, red de sensores, etc.) se podrá utilizar como referencia para implementar la solución final en un entorno real.

El caso de estudio y los elementos que lo definen se describe en la Sección 3.4. El proceso

de transición del caso de estudio a la simulación se detalla en la Sección 3.5, mientras que la construcción del entorno de pruebas de la simulación es descrito en el capítulo 4.

3.4 Definición del caso de estudio

Para construir el caso de estudio, se parte de toda la información obtenida de las diferentes fuentes de requisitos disponibles en el modelo de desarrollo implementado en el proyecto. Según la SWEBOK, las fuentes de requisitos comunes en un modelo de desarrollo tradicional que aplican de igual forma a los proyectos de soluciones asistivas, son los usuarios, los requisitos iniciales, el alcance y los planes de trabajo. Lo siguiente es organizar la información en especificaciones de requisitos de software concretas que describan entre otras cosas, las necesidades del usuario, el servicio de asistencia y la definición de las simulaciones que son necesarias para representar dichas especificaciones (ver Figura 3.4).

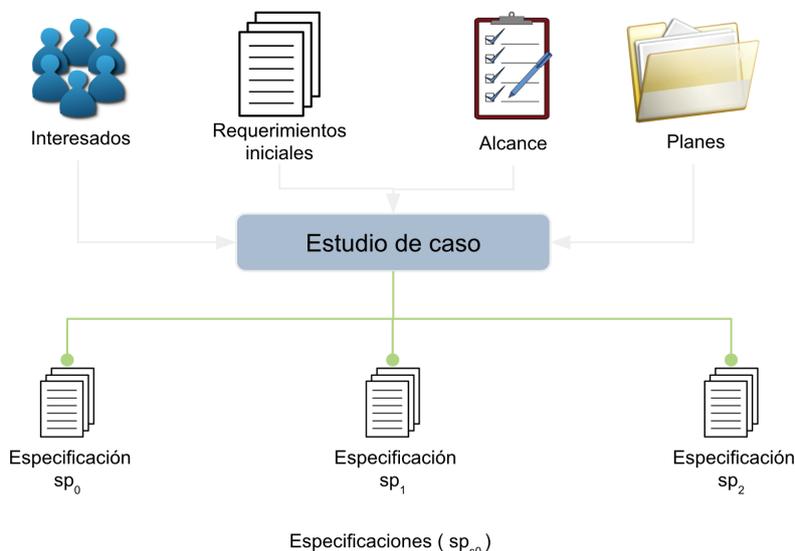


Figura 3.4: Elementos utilizados en la definición de las especificaciones del caso de estudio.

En resumen, cada especificación recoge tanto los requisitos funcionales de la solución a construir como los elementos del dominio de la solución, los cuales están enfocados en las características asistivas transversales propias del dominio de la solución y que dada su importancia, podrán ser reutilizados en todas las soluciones que puedan delimitarse dentro del marco conceptual de las simulaciones propuestas. En la Figura 3.5, se muestran los diagramas de actividades que comprenden el proceso de definición del caso de estudio.

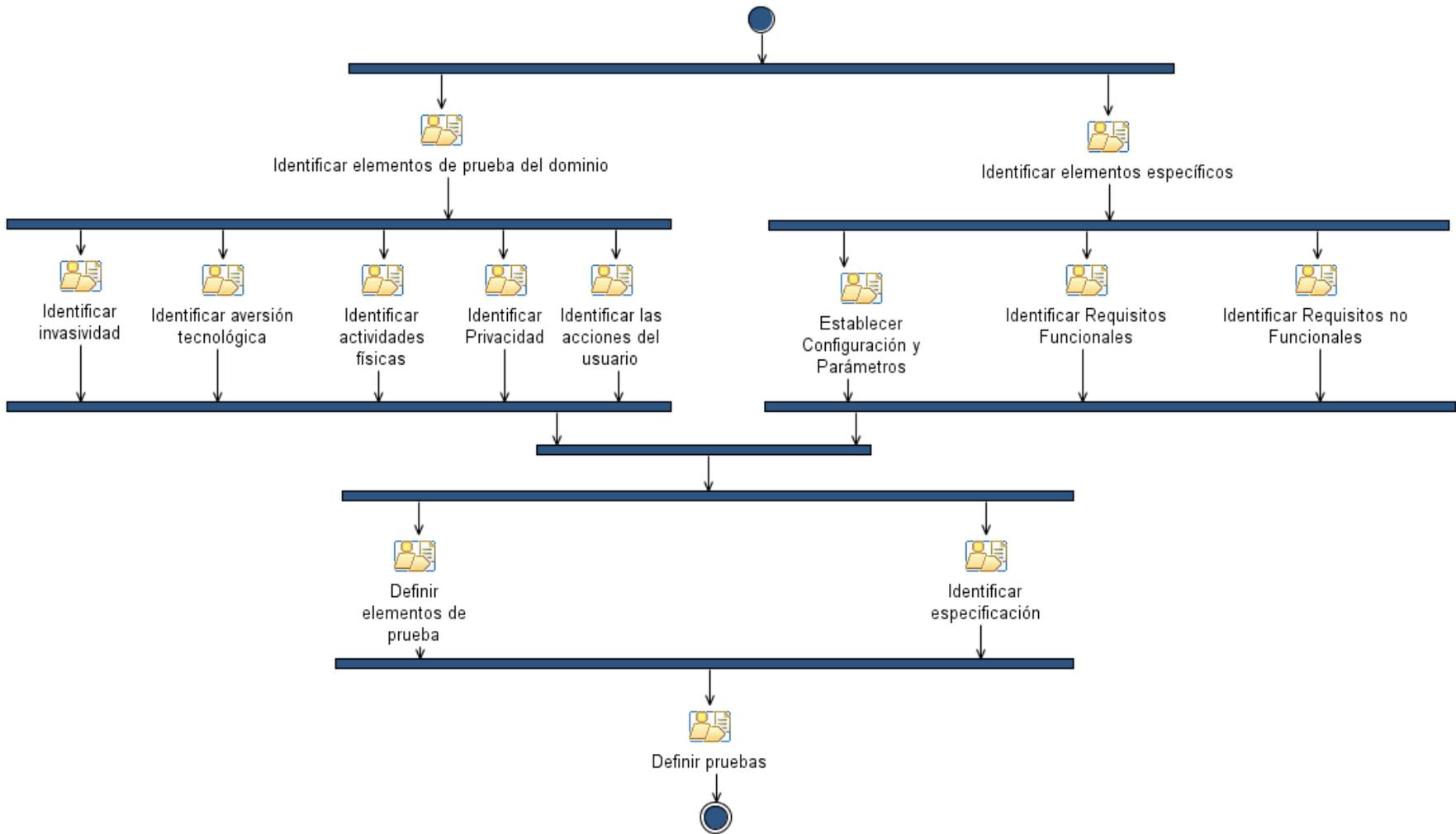


Figura 3.5: Diagrama de actividades utilizado en la solución global.

3.4.1 Usuarios

En el área de la asistencia, procesos como la identificación de requisitos se deben llevar a cabo centrándose principalmente en los usuarios. Para ello, es recomendable optar por el uso de técnicas de cocreación [PTSS10] que fomenten el trabajo en equipo y de esta forma se mitiguen problemas que puedan generar el rechazo de la solución final. En el caso concreto de la asistencia a personas mayores, es común encontrar situaciones en las que después de construida la solución, los usuarios tiendan a rechazarla por causas muy diversas. La principal tarea en la definición del caso de estudio es identificar a los usuarios que conformarán los grupos de trabajo y planificar con ellos, las actividades que se llevarán a cabo para definir las características de la solución. Aunque los grupos se organizarán en función de los perfiles asignados, es conveniente también que el trabajo permita crear equipos mixtos o multidisciplinarios (ver Figura 3.6). Lo que se busca es que la experiencia y los conocimientos de unos refuercen las opiniones de otros, generando así mayor conciencia del papel que cada uno tiene en el proceso de cocreación [PTSS10].

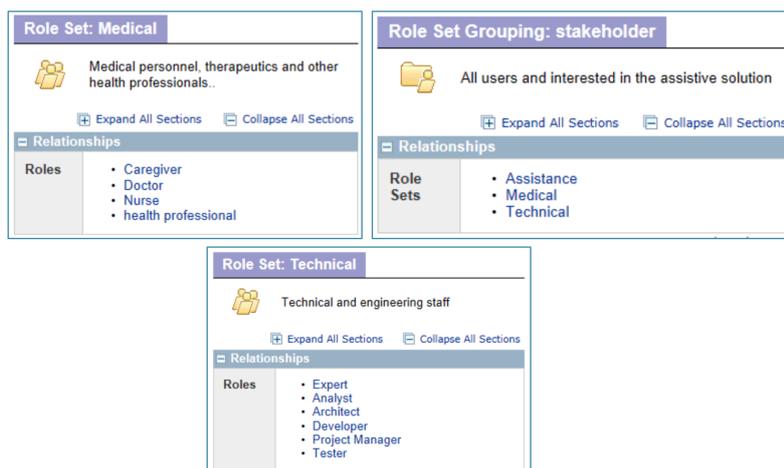


Figura 3.6: Usuarios interesados en la solución asistiva.

Los grupos de usuarios que representan a todos los interesados en la solución son los siguientes:

- **Expertos en la asistencia:** este grupo de trabajo estará compuesto por aquellos usuarios que conocen la asistencia que se quiere ofrecer (cuidadores, enfermeros, otros profesionales) y por quienes de una forma u otra estarán muy cerca de dichos servicios, una vez la solución esté definida (familiares, paciente, otros).
- **Profesionales de la salud:** el segundo grupo lo conforman expertos en temas de salud (médicos, terapeutas, psicólogos, enfermeros, etc.), los cuales dirigirán sus esfuerzos para que la solución tenga en cuenta siempre las connotaciones médicas relacionadas con la asistencia.
- **Técnicos e ingenieros:** es el grupo compuesto por aquellos expertos en soluciones tecnológicas que participarán en la construcción de una solución software y/o hardware (si la hubiese), que acompaña la solución asistiva (analista, arquitectos, desarrolladores, directores de proyecto, pruebas, entre otros).

3.4.2 Requisitos

La identificación de requisitos es otra tarea llevada a cabo para abordar la construcción del caso de estudio y es el resultado del trabajo en equipo de todos los usuarios que se han identificado previamente, responsables de definir los requisitos y las pruebas correspondientes de la solución asistiva (ver Figura 3.7).

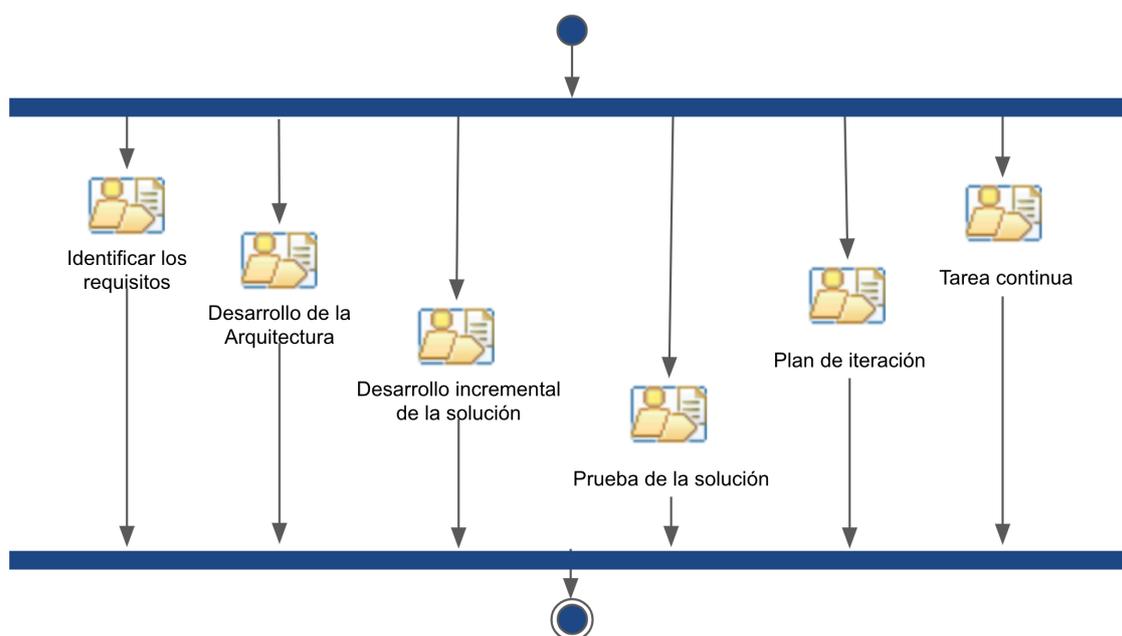


Figura 3.7: Fase de identificación de requisitos del caso de estudio.

La simulación es el artefacto que se usará en el proceso de captura de requisitos y como se ha indicado previamente en el caso de estudio, ésta recoge la especificación de los requisitos y las pruebas de validación dentro del marco conceptual de las simulaciones. En este contexto, los requisitos y las pruebas correspondientes se centran en la tarea de definir la simulación en función de acciones o actividades que ocurren en ella. La configuración y orquestación de dichas actividades van en consonancia con los requisitos de la solución asistida, la cual básicamente alude a preguntas como:

- ¿Es este el problema que se quiere resolver?
- ¿Es esta la asistencia que contribuirá a dar respuesta al problema?
- ¿El servicio de asistencia simulado se corresponde con el problema?
- ¿Es esta la forma en que se quiere resolver el problema?
- ¿Lo simulado refleja la forma en que se quiere resolver el problema?
- ¿La simulación muestra el comportamiento de la solución?

Las respuestas a este tipo de interrogantes, arrojan información útil para definir el caso de estudio y generar las simulaciones que incluyen las características tanto del problema como de la solución en el entorno simulado. Una misma simulación no representa los dos escenarios, de ahí la importancia de modelar por un lado, la situación correcta, por otra parte el contexto del problema y finalmente la solución aplicada.

3.4.2.1 Requisitos técnicos

Las simulaciones utilizadas en esta investigación, permiten una conexión directa sobre cada elemento de la simulación, teniendo en cuenta las características funcionales que se han dado a cada uno de ellos en el caso de estudio. Estas mismas características delimitarán los aspectos que podrán ser validados en la simulación.

Usuario: en la Tabla 3.1 se muestran las características de este elemento, el cual puede usarse en la simulación para representar al usuario real en un entorno concreto. Estas características recogen aspectos generales que describen el comportamiento del usuario tales como su localización dentro del escenario, su estado o posición “física” en todo momento, su interacción con el entorno y si tiene problemas de movilidad que le impidan realizar determinadas acciones. Con estas características, las validaciones pueden orientarse a determinar si el usuario se encuentra en un sitio u otro dentro de la vivienda, si interactúa con elementos del entorno, si se halla ejecutando alguna actividad física o si tiene problemas a la hora de desempeñar dicha actividad.

Usuario	Características
Posicionamiento	Ubicación del usuario dentro de la vivienda
Estado	Reposo, en movimiento, acostado, sentado, en pie
Movilidad	Limitación física o problemas de movilidad en brazos y piernas
Síntomas de enfermedades	Temblores de intensidad variable en cuello y manos.
Acciones	Tareas como: beber, caminar, correr, saludar, comer, asir

Tabla 3.1: Características del elemento “usuario” en la simulación.

Es posible utilizar síntomas comunes de determinadas enfermedades, los cuales se definen de forma programática en la simulación y se emplean como factores que limitan la condición del paciente. Los síntomas que por defecto incluye la simulación son temblores de intensidad variable en manos y cuello. Las acciones por su parte, se describen en la misma tabla y para poderlas llevar a cabo, la simulación emplea las animaciones de ordenador que ya están predefinidas en la herramienta de simulación.

Entorno: recoge las características principales que describen la delimitación del espacio físico. Entre estas características se destacan las limitaciones de la vivienda del usuario y el aspecto visual o estético del escenario en la simulación con sus correspondientes elementos decorativos. Asimismo, incluye los objetos dinámicos dotados de cierto tipo de comportamiento animado (animaciones) que no interactúan directamente con el usuario y aquellos objetos interactivos que son manipulados por el usuario y a los cuales, se les puede cambiar su estado dentro de la simulación (ver Tabla 3.2).

Dispositivos: desde el punto de vista tecnológico, la simulación cuenta con cuatro tipos de dispositivos: los *sensores* que son emulados virtualmente dentro de la simulación, los *actuadores* que permiten activar o desactivar acciones dentro de la simulación y finalmente los *Smartphone* y *Wearable*, los cuales son virtualizados o emulados con Android [CJN17] dentro de la simulación y permiten ejecutar aplicaciones móviles (ver Tabla 3.3).

Entorno	Características
Alcance	Una vivienda.
Decorativo	Aspectos estéticos de la simulación cuyo propósito es ambientar visualmente el escenario (cuadros, libros, estanterías, alfombras, entre otros).
Dinámico	Elementos que pueden animarse en la simulación (televisión, luces, grifos)
Interactivo	Elementos que pueden interactuar con el usuario (sillas, puertas, interruptores, llaves del grifo, mando a distancia)

Tabla 3.2: Características del elemento “*entorno*” en la simulación.

Dispositivos	Características
Sensores	Luminosidad, posición, aceleración, giroscopio, ruido, movimiento, micrófono
Actuadores	Simples: táctiles, de apertura/cierre, de luz
Smartphone	Emulador de un teléfono móvil (aplicación móvil)
Wearable	Emulador de un wearable (aplicaciones específicas)

Tabla 3.3: Características de los “*dispositivos*” emulados en la simulación.

3.4.3 Elementos del dominio de la asistencia

Los elementos del dominio de la asistencia elegidos para esta investigación, son indispensables para cumplir con los criterios que rige la definición de una solución asistiva. Dichos elementos se han podido modelar tanto para el caso de estudio como para la simulación, utilizando para ello determinados aspectos o situaciones que pueden ser validadas en la simulación. Para llevar a cabo dicha validación, las soluciones deben construirse bajo el framework de pruebas propuesto, estableciendo aquellas características del comportamiento que son modelables en el usuario, el entorno y en la tecnología de apoyo.

No todas las situaciones que abordan cada elemento del dominio, son cubiertas en esta investigación. Las razones que llevan a establecer esta restricción son, por un lado, que las herramientas con que se construye la simulación no dan soporte a todos los aspectos que describen cada elemento. Por otra parte, el contexto de estos elementos no siempre es totalmente ingenieril, sino que muchas veces presentan un enfoque más social o en su defecto, están muy asociados al ámbito de la salud.

La validación no diagnosticará si ocurre o no una situación o un aspecto del elemento del dominio, simplemente dará a los expertos resultados de las pruebas sobre lo “recurrente” o “frecuente” que son las situaciones en el contexto del usuario y la tecnología, alertando así de posibles casos donde se podrían estar violando las recomendaciones asociadas al dominio asistivo. El diagnóstico final de los problemas que generan los elementos del dominio quedará en manos de los profesionales correspondientes y no del equipo de ingenieros.

3.4.3.1 Identificación de situaciones donde la tecnología puede ser invasiva

Para incluir estas situaciones en la construcción del caso de estudio, se han propuesto cinco tareas concretas que deben ser abordadas durante la fase de definición. Como se observa en la Figura 3.8, estas tareas lo que buscan es identificar determinados patrones que adviertan del riesgo ante posibles escenarios donde la tecnología utilizada puede llegar a ser invasiva. Una vez identificados los factores de riesgo, la siguiente tarea es definir y establecer un plan que mitigue el problema. Este proceso debe ser iterativo y constante en toda la fase de desarrollo, por ello desde el propio framework de pruebas que se propone en este trabajo, implícitamente se comprueban determinados criterios que advierten de posibles riesgos en lo que respecta al uso de la tecnología virtual utilizada en la simulación.

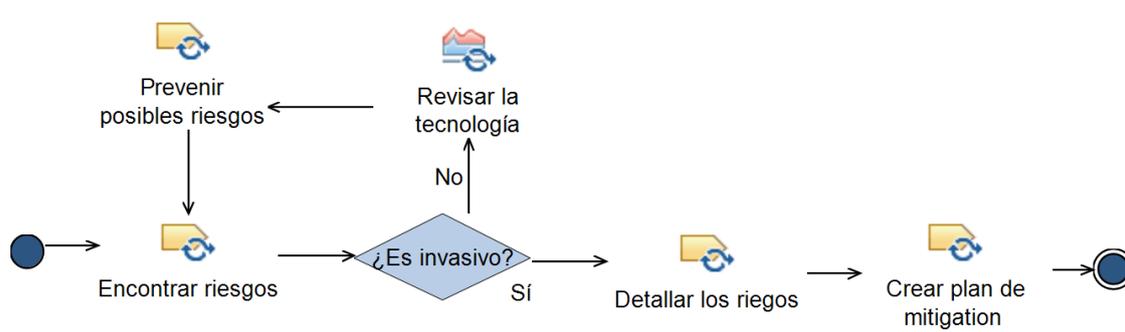


Figura 3.8: Tareas recomendadas para identificar los riesgos de la tecnología invasiva.

Las situaciones de invasividad que se han modelado para su validación, están relacionadas con la identificación del número de sensores y actuadores utilizados tanto en el entorno como en el usuario, y con la posición de dichos sensores especialmente sobre el usuario. La relación cantidad/posición de los sensores es un criterio que el ingeniero puede incluir en la validación, determinando previamente los valores que considere oportuno para ambos casos. No es una tarea fácil encontrar el número exacto de dispositivos en la solución y tampoco es posible determinar si los sensores utilizados son de por sí invasivos o no. Por ello, es necesario consensuar con todos los interesados si la red de sensores propuesta se adecuó a las necesidades del usuario.

En los casos que la validación, siguiendo el criterio de los ingenieros, determina que existe riesgo de “invasividad”, es responsabilidad del equipo encontrar la raíz del problema y establecer un plan que lo mitigue. La alternativa propuesta en estos casos, es usar la simulación como un artefacto que permita a los ingenieros optimizar el número de dispositivos [GSPACB17] y su ubicación [PBGSP17] para cumplir con el objetivo propuesto, especialmente en escenarios donde la monitorización del usuario es a largo plazo o donde deben definirse sistemas de respuestas ante emergencias (ver Tabla 3.4).

3.4.3.2 Situaciones de aversión tecnológica

Encontrar aspectos que indiquen una posible aversión tecnológica por parte del usuario en la simulación no es tarea trivial de abordar. Dada la relevancia social que tiene esta situación en la solución, es necesario determinar los aspectos generales que permiten reconocer la aversión tecnológica. Para construir la simulación se deben identificar al menos

Tipo	Ubicación	Criterioe
Sensores	Cuerpo	- Número de sensores en el cuerpo (por tipo) - Uso de sensores del mismo tipo con una configuración diferente - Uso de sensores de diferente tipo y configuración
	Entorno	- Número de sensores en el entorno (por tipo)
Actuadores	Cuerpo	- Número de actuadores en el cuerpo
	Entorno	- Número de actuadores en el entorno

Tabla 3.4: Criterios de aceptación utilizados en las pruebas del caso de uso.

tres aspectos del usuario que son fácilmente extrapolables a otros factores que provocan la aversión tecnológica: la afinidad por las nuevas tecnologías [LB97], su experiencia previa [EI04] y la formación que tiene o debe recibir sobre la solución.

En la simulación, desde un punto de vista social, estos tres aspectos se modelan junto a los requisitos que describen la tecnología de apoyo. El objetivo es acondicionar el uso de la asistencia a las preferencias del usuario, teniendo en cuenta (i) el uso de dispositivos tecnológicos afines, los cuales puede ser usados directamente o ser acoplados a otros dispositivos más comunes para él. (ii) Reducir la necesidad de que el usuario interactúe con los dispositivos o en su defecto, que pueda interactuar sin requerir experiencia alguna para ello y (iii) definir tecnologías de apoyo cuya curva de aprendizaje para su uso sea muy baja. La Figura 3.9 muestra las tareas recomendadas para identificar y actuar ante situaciones de aversión tecnológica que surjan a lo largo del ciclo de desarrollo de la solución asistiva.

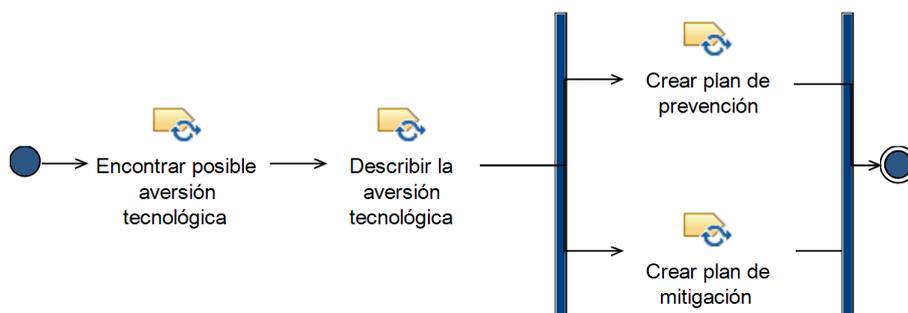


Figura 3.9: Tareas para identificar posibles riesgos de aversión tecnológica.

Como se observa en la Figura 3.9, para construir el caso de estudio teniendo en cuenta las situaciones de aversión tecnológica, se deben realizar las siguientes actividades:

- **Encontrar posible aversión tecnológica:** identificar los riesgos de que exista aversión tecnológica por parte del usuario, utilizando como base los tres aspectos citados anteriormente: poca o ninguna afinidad tecnológica, poca o ninguna experiencia previa y necesidad de formación para usar la tecnología.
- **Describir la aversión tecnológica:** describir los casos de aversión tecnológica que existen y no han podido ser mitigados. Con la ayuda de los profesionales, estos casos se describirán en las simulaciones problema del caso de estudio.

- **Crear un plan de prevención:** como parte del caso de estudio, se define un plan que ayude a prevenir los problemas de aversión. El equipo podrá ir revisando los tres aspectos descritos anteriormente, conforme va proponiendo las distintas tecnologías de apoyo en la solución.
- **Crear un plan de mitigación:** el plan de mitigación es útil en los casos donde es necesario utilizar tecnología que no cumpla alguno de los aspectos descritos. En estos casos, se pueden crear estrategias que refuercen la afinidad del usuario hacia la tecnología propuesta o que brinden algún tipo de entrenamiento para usarla, evitando así situaciones de rechazo o estrés por parte del usuario en el futuro.

3.4.3.3 Privacidad

En el marco conceptual actual de las simulaciones, no es posible abordar todas las connotaciones que implican la seguridad en los entornos asistivos. Por ello, lo que se propone es recomendar a los diseñadores del caso de estudio que se centren en uno de los aspectos más valorados a la hora de estudiar este factor dentro de la solución asistiva: la percepción del riesgo de la seguridad Vs. utilidad y facilidad de uso en la solución [VS99]. En lo que respecta a la percepción de seguridad [LHBC⁺11], además de dar valor y promover la utilidad de la asistencia, los datos y su nivel de “granularidad” al igual que conocer el destinatario de la información procesada, son aspectos que deben documentarse y especificarse claramente en el caso de estudio. En la simulación ésto se representa con la presencia de un profesional médico, del cuidador o de un familiar como posible receptor de la información del usuario. La “granularidad” de los datos se aplica sobre la información médica del usuario y sobre los datos del contexto recogidos por los dispositivos del entorno.

Para incluir estos aspectos en el caso de estudio, la tarea que se propone es básicamente encontrar los posibles riesgos relacionados con la falsificación de los datos y la extracción de información privada recogida por las redes de sensores [CP03]. Una vez se detecta el riesgo, se debe definir una estrategia que lo elimine y que revise las vulnerabilidades de la tecnología utilizada, tal y como se muestra en la Figura 3.10.

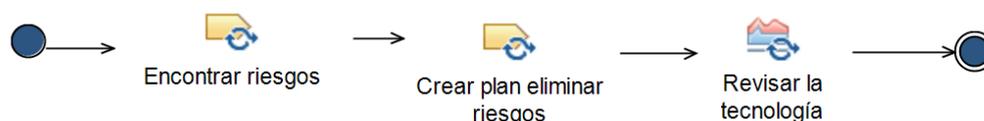


Figura 3.10: Actividades para la gestión de los problemas de privacidad.

En el framework propuesto, la implementación del entorno de pruebas utiliza una estrategia de anonimización sobre los datos que generan la red de sensores, la cual consiste en un sistema de etiquetas sobre todos los elementos de la simulación. Como se observa en la Figura 3.11, una etiqueta funciona como identificador que permite mapear y referenciar los elementos sobre la propia simulación, muy similar a los utilizados en los entornos monitorizados con sensores.

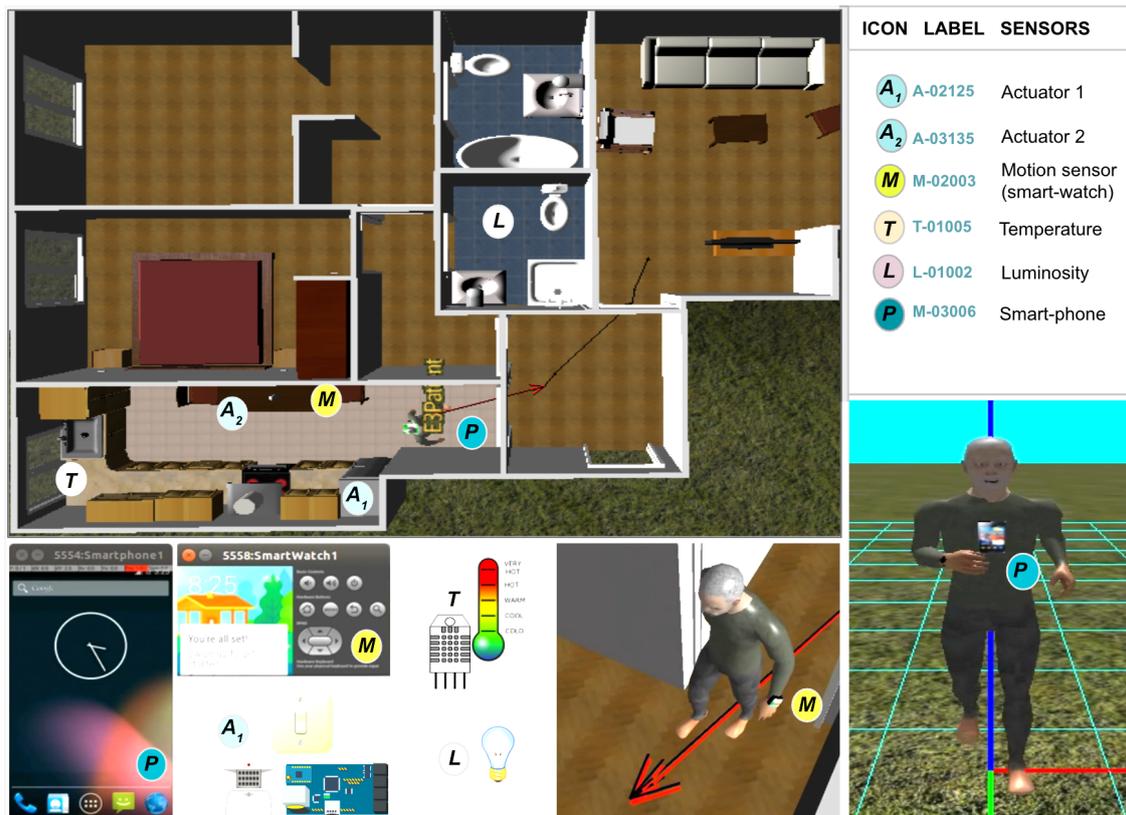


Figura 3.11: Diagrama de implementación del sistema de etiquetas en un entorno monitorizado con sensores.

3.4.3.4 Respuestas no deseadas del usuario

La primera tarea que se debe llevar a cabo para reconocer las respuestas del usuario, es “identificar el rol del usuario” con respecto al uso de la tecnología. El rol puede dar más peso a una respuesta que a otra. No es parte de esta investigación establecer dicha relación; los expertos tendrán la responsabilidad de identificarla en función de las acciones que se proponen como respuestas. Como se observa en la Figura 3.12, en esta investigación se han elegido cuatro respuestas típicas asociadas a la mayor problemática en los entornos de vida asistiva, especialmente al momento de demandar una interacción por parte del usuario. Estas respuestas son:

- **No hay interacción con la solución:** el usuario no interactúa con la herramienta ante ciertas circunstancias.
- **Ignorar la solución:** el usuario no acepta interactuar con la herramienta.
- **Respuestas no a tiempo:** el usuario no interactúa con la herramienta en el momento justo que debe hacerlo (cuando se demanda alguna acción por su parte).
- **Respuestas repetitivas:** el usuario interactúa con la herramienta más de lo necesario o lo hace de forma repetitiva.

Cada respuesta es asociada a una tarea dentro del proceso. En cada una de ellas, los expertos definirán dos aspectos importantes: los criterios que se quieren medir en la respuesta y las acciones que se desencadenarán posteriormente.

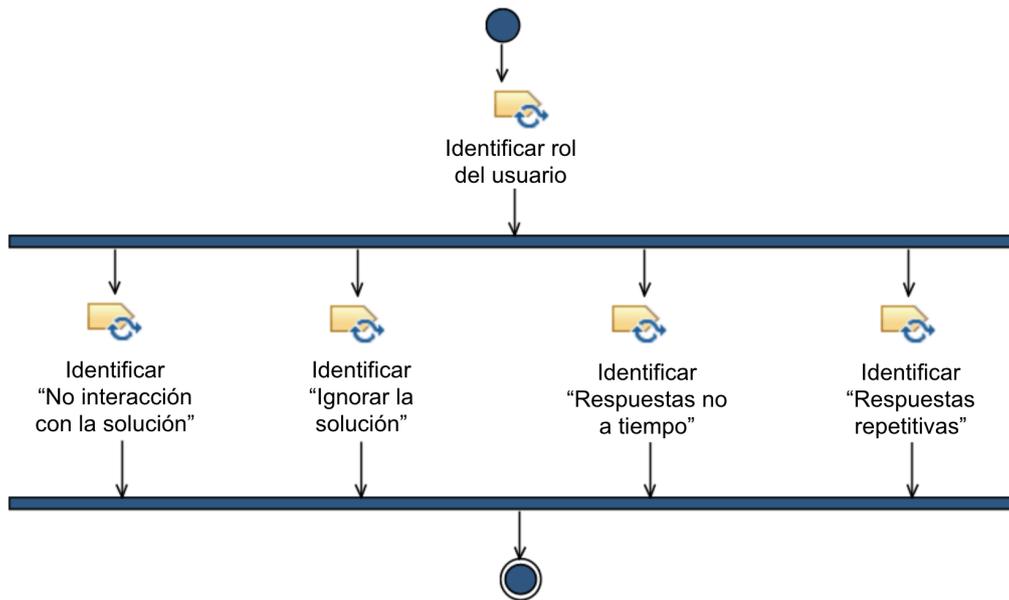


Figura 3.12: Actividades propuestas para generar los planes de acción.

En la Tabla 3.5, se puede observar los criterios que se establecen para cada respuesta, los cuales equivalen a medir el tiempo en que se producen dichas respuestas. Con base a estas métricas, se establecen acciones que se ejecutarán según las condiciones del usuario y las de cualquier acompañante en la simulación.

Respuestas	Criterios	Acciones
Ignorar la solución	- No hay respuesta	- Notificar al usuario cada x tiempo - Notificar a otro usuario - Comprobar el envío de notificaciones - Comprobar si no hay respuesta del usuario por más de x tiempo
No hay interacción con la solución	- Medir el tiempo sin respuestas - Medir los tiempos entre "no respuestas" - Detectar las tareas con respuesta - Detectar las tareas sin respuestas	- Notificar a otro usuario - Comprobar el envío de notificaciones
No hay interacción a tiempo	Medir los tiempos de respuesta: - Promedio - Mínimo - Máximo	- Notificar al usuario cada x tiempo - Notificar al usuario cada x tiempo si el tiempo sin respuesta excede x. - Comprobar el envío de notificaciones
Respuestas repetidas	Medir el tiempo de respuestas: - Promedio - Mínimo - Máximo	- ¿Con qué frecuencia se repite la respuesta? - Detectar en qué tareas repite la respuesta el usuario. - Detectar en qué tareas el usuario no repite la respuesta. - En las tareas de tipo x, las respuestas repetidas frecuentes. - En las tareas de tipo x, las no respuestas repetidas frecuentes.

Tabla 3.5: Plan de acción propuesto para las respuestas no deseadas de cada usuario.

3.4.3.5 Identificar los problemas de movilidad

El objetivo con este elemento del dominio no es reconocer ni diagnosticar el estado del paciente, lo que se busca es definir un mecanismo que permita a los expertos modelar problemas de movilidad del usuario relacionados con la marcha, los cuales pueden afectar su interacción con la solución asistiva propuesta. Las características que se pueden modelar en la simulación se muestran en la Tabla 3.6. Dichas características permiten representar los cambios anormales en la actividad diaria del usuario. Los ingenieros, las pueden asociar al comportamiento del usuario en función de los requisitos que describan la situación o el estado del mismo.

Acciones	Área	Descripción
Simetría del cuerpo	Movimientos	- Tiempo de viaje lado izquierdo y derecho. - Longitud de los pasos, cadencia, movimiento del torso y el movimiento de los tobillos, rodillas, caderas y pelvis es el mismo en ambos lados.
Retropulsión	Parte superior del cuerpo	- Inclinación hacia atrás al comenzar la marcha. - Caer hacia atrás mientras el individuo camina.
Movimientos de los pies	Caída del pie	- Elevación exagerada de la pierna para evitar tropezones. - El balanceo bajo del pie.
	Longitud de los pasos	- Disminución de la longitud de los escalones. - La pierna con el paso más corto. - La duración de la estancia de pie en la pierna derecha normal.
	Movimiento circular Circunducción	- Movimiento del pie arqueado cuando el usuario da un paso adelante. - Movimiento del pie en línea recta cuando el usuario da un paso.
Inclinación parte superior del cuerpo	Avanzar	- Inclinación - Inestabilidad
	Lateral	- Hacia el lado que permanece apoyado en el suelo.
Hombros	Balanceo de brazos	- Incrementa. - Decrementa.

Tabla 3.6: Caracterización de los cambios anormales en el usuario.

Como se observa en la Figura 3.13, la forma de incluir los problemas de movilidad en el caso de estudio y en la simulación, se reduce a tres tareas concretas: identificar el problema de movilidad que existe en el usuario, modelar el problema en el caso de estudio y definir los criterios a utilizar para identificar sus posibles ocurrencias.



Figura 3.13: Actividades para modelar los problemas de movilidad.

Identificar estas situaciones, además de ser útiles para dar información a especialistas de la actividad física del usuario, permite que la solución asistida se diseñe teniendo en cuenta estos problemas, evitando así que interfieran con el objetivo de la asistencia o el apoyo que se quiera dar al usuario. En la Tabla 3.7 se describen las características que se

modelan para representar los cambios anormales en la marcha del usuario.

Para establecer los criterios que ayuden a identificar estos problemas, los expertos pueden hacer uso de sistemas de puntuación que permitan valorar los problemas de movilidad del usuario. Un sistema de puntos que puede utilizarse en personas mayores, es el que se muestra en la Tabla 3.8 [JO17], el cual está basado en una tabla de puntos^{1,2} que permite a estimar la causa probable de diferentes problemas de movilidad. Este sistema fue adaptado a partir del estudio de Tinetti [Tin86].

En el caso de la solución asistiva, estos mismos criterios se podrían aplicar con ayuda de la tecnología de apoyo o mediante algún sistema de monitorización sobre el usuario. Posteriormente, el especialista puede hacer su valoración y un análisis posterior de la situación del usuario. En la simulación, la monitorización de las actividades físicas del usuario es lo que permitiría llevar a cabo la identificación de este tipo de situaciones.

¹La puntuación perfecta es 12.

²Una puntuación < 10 suele asociarse con limitaciones de la acción determinada, relacionada con la movilidad.

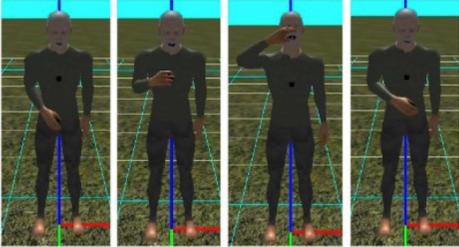
Action	Specific	Description	Example
Position (standing)	Repose	Position of the user while at rest.	
	To walk	Position of the user while walking.	
Movements (articulation)	Elbows	<ul style="list-style-type: none"> • Degree of articulation (minimum and maximum). • Degree of articulation (average). 	
	Hands		
	Knees		
	Feet		
Walk	Standing time with a single support	The time it takes the user to stand with a single support (standing).	
	Standing time with two support	The time it takes the user to stay upright with both support (both feet).	
	Cadence (steps / minute)	Represent the rhythm with which the use of energy is more efficient depending on the steps. Criteria: <ul style="list-style-type: none"> • Tall people perform longer steps with a slower rate. • Low people perform shorter steps with a faster rate. • It should not change with age. Additional data: legs height and length.	
	Speed (to walk)	<ul style="list-style-type: none"> • Minimum and maximum speed during the day. • Average speed. • Variations. 	

Tabla 3.7: Caracterización de los cambios normales en el usuario.

Componente	Hallazgos	Puntuación*	Relevancia clínica
Inicio de la marcha (inmediatamente después de la indicación de moverse)	Vacilación o múltiples intentos para empezar a caminar	0	Enfermedad de Parkinson
			Falla aislada al iniciar la marcha (accidente cerebrovascular o demencia)
			Enfermedades frontales que afectan la marcha
	Sin vacilación	1	
Longitud y altura del pie derecho (durante el balanceo del mismo pie)	El paso no supera el sitio donde se encuentra el pie izquierdo o el pie no se levanta del suelo por completo al dar el paso	0	Artritis
			Problemas en el pie
			Accidente cerebrovascular
	Supera al pie izquierdo apoyado en el suelo	1	
	Se levanta por completo del suelo	1	
Longitud y altura del pie izquierdo (durante el balanceo del mismo pie)	El paso no supera al pie derecho apoyado en el suelo o el pie no se levanta por completo del suelo	0	Artritis
			Problema en el pie
			Accidente cerebrovascular
	Supera al pie derecho apoyado en el suelo	1	
	Se levanta por completo del suelo	1	
Simetría del paso	La longitud de los pasos derecho e izquierdo no son iguales (estimado)	0	Unilateral
			Deficiencia musculoesquelética o neurológica focalizada
	Longitud equivalente del paso derecho y el izquierdo (estimada)	1	
Continuidad del paso	Detención o discontinuidad en la marcha (entre pasos)	0	Enfermedad frontal que afecta la marcha
			Miedo de caer
	Los pasos son continuos	1	
Trayectoria (estimada en relación con las baldosas del suelo de 30 cm ó 12 pulgadas de ancho; se observa excursión de un pie desviada más de 25,4 cm ó 10 pulgadas de la trayectoria)	Desviación significativa	0	Enfermedad de la marcha frontal
			Desviación entre leve y moderada o uso de dispositivo de asistencia para caminar
			Marcha recta sin dispositivo de asistencia
Tronco	Balanceo notable, uso de dispositivo de asistencia para caminar	0	Disfunción cerebelosa, subcortical y de los ganglios basales
			Marcha antiálgica (artritis de cadera o de rodilla)
	Sin balanceo pero con flexión de las rodillas, dorsalgia o separación de los brazos al caminar	1	Miedo de caer
	Sin balanceo, flexión ni uso de los brazos y sin necesidad de dispositivos de asistencia para caminar	2	
Base de sustentación (ancho del paso)	Los talones quedan muy separados durante la deambulación	0	Enfermedad de cadera
			Enfermedad cerebelosa
			Hidrocefalia con presión normal
	Los talones casi se tocan durante la marcha	1	

Tabla 3.8: Guía de evaluación de la movilidad orientada al desempeño.

3.4.4 Captura de requisitos

Debido a la cantidad de usuarios que pueden participar en los proyectos de software de asistencia, después de revisar el estado del arte de las distintas técnicas y herramientas que se pueden utilizar para promover la participación de los usuarios [MVK08] (ver Sección 2.1), se proponen concretamente cuatro técnicas que tienen un enfoque social adecuado [MS09] para la obtención de información cercana al usuario. Los grupos focales, las entrevistas, los cuestionarios y la observación son técnicas que se consideran muy útiles en ambientes cocreativos que buscan establecer los requisitos de las soluciones asistivas [CCPFR12].

Mediante estas técnicas y según el rol de los distintos interesados [NCC⁺08], éstos tienen el trabajo de identificar y describir el dominio de la aplicación y las tecnologías que darán apoyo en la solución dentro del entorno [MWP⁺14] (ver Tabla 3.9). Dicha descripción incluye como mínimo tres soportes tecnológicos importantes de la asistencia [ACRV13]: la privacidad de los datos del paciente, la seguridad del usuario mediante la monitorización o el uso de sistemas de vigilancia y la detección de emergencias con su respectivo sistema de respuesta.

Elementos de la simulación	Expertos en asistencia	Profesionales de la salud	Técnicos e ingenieros
Usuarios	✓	✓	
Entorno	✓		✓
Tecnología			✓

Tabla 3.9: Participación de los roles en la descripción de la simulación.

3.4.4.1 Técnica de trabajo en grupo propuesta

Después de revisar el estado del arte de las diferentes técnicas de grupo, se ha decidido utilizar los “*Online focus group*” basados en textos como técnica que permita la definición y construcción de los casos de estudio [CBCGSP18]. El uso de esta técnica permite que los participantes puedan expresar con una entrada de texto libre lo que piensan sin restricciones. La técnica se lleva a cabo mediante una plataforma web que reduce al mínimo los requisitos tecnológicos. Inicialmente, la plataforma ofrece una breve encuesta para filtrar a los participantes. Después de completarla, cada participante será conducido a la página apropiada del grupo de discusión.

Cada caso de estudio comienza con una breve descripción del tema de la discusión, más un vídeo introductorio y motivador sobre el tema. Hay un texto inicial que explica el objetivo de la investigación y cómo pueden contribuir los participantes. Luego sigue la participación en línea, con formularios de entrada que permiten enviar comentarios, calificar, citar o responder a otros usuarios de forma ágil. Mientras esto ocurre, los moderadores pueden intervenir para reorientar la discusión o prevenir situaciones amenazantes.

3.4.5 Organización de las tareas

Dentro del esquema general de la fase identificación de requisitos, se incluyen las tareas que se llevarán a cabo para recabar información sobre los requisitos (ver Figura 3.14).

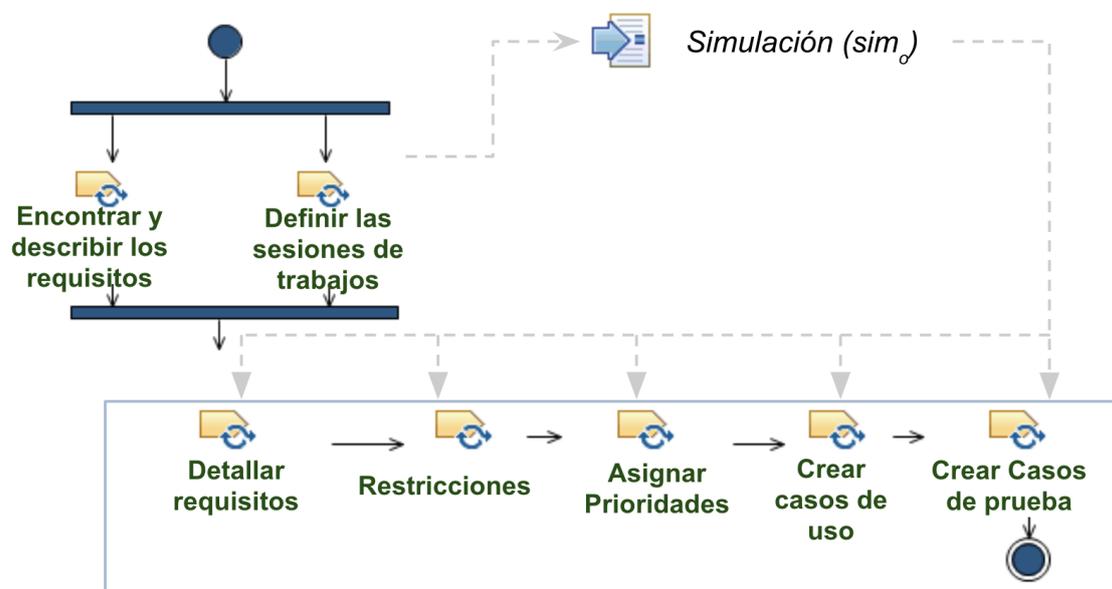


Figura 3.14: Tareas para identificar los requisitos de la solución.

Esta tarea permitirá desde el inicio del proyecto organizar y planificar las sesiones de trabajo con todos los interesados para definir los primeros hitos de la solución y, al mismo tiempo, las características iniciales del caso de estudio. Las primeras especificaciones describirán las versiones iniciales de la simulación sim_0 , las cuales mostrarán el escenario típico (*situación cotidiana*) que rodea al usuario en su día a día. La evolución de estas primeras versiones de la simulación, se utilizarán para mostrar variaciones en las situaciones cotidianas que vive el paciente sin describir en ningún momento cuestiones relacionadas con la problemática que este presenta. En las siguientes simulaciones $sim_1, sim_2, \dots, sim_n$ si se irá profundizando gradualmente en la descripción del problema hasta que finalmente, se describa la solución que se usará para dar la asistencia correspondiente.

Como se muestra en la Figura 3.15, una transición gradual entre simulaciones se produce con la participación de los distintos interesados. Las simulaciones no deben ser parciales ni deben limitarse a la perspectiva de un único rol. La cocreación justamente permitirá eliminar esta barrera y se buscará promover con ella que el caso de estudio recoja la visión global de todo el equipo de trabajo. Las simulaciones que se construyan serán utilizadas en las siguientes sesiones de trabajo como un artefacto que recoge la especificación de uno o varios requisitos. Al tratarse de una representación de éstos, la simulación abarca tanto el detalle del requisito así como las restricciones que se le hayan asignado. De la misma forma, muestra las prioridades asignadas y el caso de uso al que corresponde con las respectivas pruebas que se aplicarán sobre él.

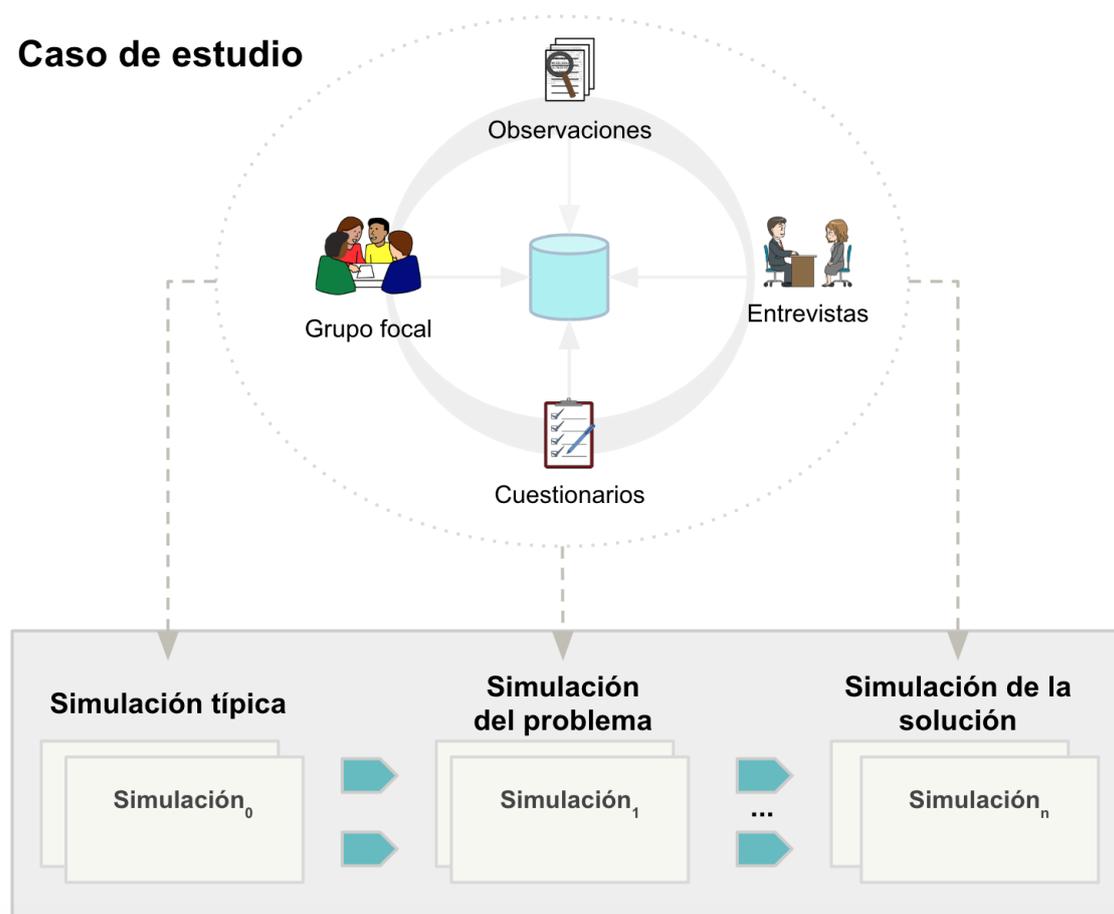


Figura 3.15: Transición de la simulación en el caso de estudio.

3.4.6 Resultado de las tareas

Como se observa en la Figura 3.16, los resultados del proceso de definición del caso de estudio son principalmente aquellos artefactos que permitirán no solo organizar el trabajo del equipo de desarrollo sino también los que se utilizarán para construir y validar la simulación.

- **Productos del trabajo:** desde el ámbito ingenieril, los productos del trabajo son aquellos modelos, registros, formatos, listas, documentos, programas y datos generados por las actividades y las tareas establecidas en el proceso de desarrollo. De cara a la simulación, con este artefacto se podrán contrastar las especificaciones a medida que se construye la simulación, especialmente en las primeras versiones de los distintos escenarios.
- **Glosario de términos:** recoge el vocabulario propio del dominio en el área asistiva y del área médica en relación al estado del paciente. Para el trabajo con equipos interdisciplinarios, es importante que este artefacto esté al alcance de todos, especialmente cuando se estén trasladando las especificaciones a la simulación. Es muy probable que en este proceso participen todos los interesados de la solución por lo que la puesta en común de la terminología, estimulará notoriamente la

participación.

- **Plan de iteración:** el plan de iteración describe con mayor o menor detalle la planificación temporal (cortos lapsos de tiempo o iteraciones) del proyecto. Además, se podrán encontrar las tareas o actividades detalladas (ver Sección 3.4.5), los riesgos identificados (ver Sección 3.4.3), los artefactos que cambien o que se creen nuevos durante el proceso y no menos importante, los hitos esperados en cada iteración. La simulación como elemento de especificación se verá afectada por esta planificación, sobre todo cuando en cada iteración se proponga una simulación diferente o mejorada.
- **Especificaciones:** describen los elementos que se incluirán en la simulación (usuario, entorno y tecnología), así como la lista de acciones o actividades que se ejecutarán en la simulación para analizar la interacción entre los distintos elementos simulados.

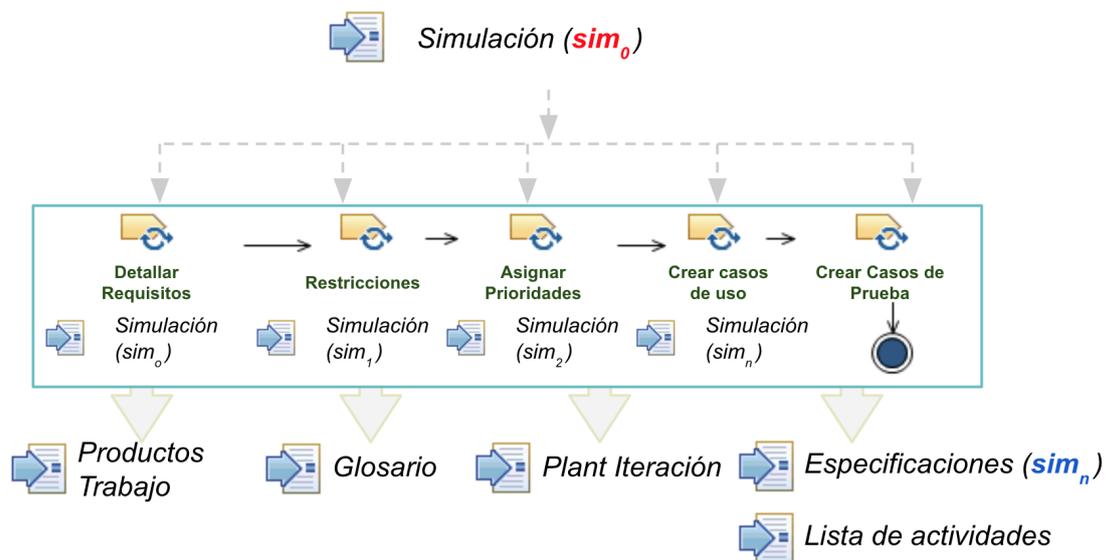


Figura 3.16: Resultado de la definición del caso de estudio.

3.4.6.1 Lista de actividades en la especificación

La especificación de la simulación abarca los requisitos que definen y le dan una finalidad concreta a la solución. La forma de mostrar dicha finalidad es mediante la representación de “actividades” o “acciones” que ejecutarán los elementos de la simulación en un orden determinado. Este conjunto de actividades es lo que se conoce como la *lista de acciones o actividades* a modelar y en ella se recoge, con un estricto detalle, todo lo que ocurrirá durante el tiempo que dure la simulación.

Cada “actividad” definida, tiene asociada una serie de características que permiten a los ingenieros interpretarlas y modelarlas posteriormente en la simulación. Cada una de ellas va acompañada de información que muestra su relación o dependencia con los diferentes elementos de la simulación, como se describe en la Sección 3.5.1. Como se observa en la Figura 3.17, una acción o actividad dependerá del objeto que la lleve a cabo en la

simulación, por ejemplo: la acción de *encender la luz* dependerá o bien del usuario que ejecuta la acción física o del dispositivo que esté programado para encenderse (ejecutar el efecto luz encendida o apaga).

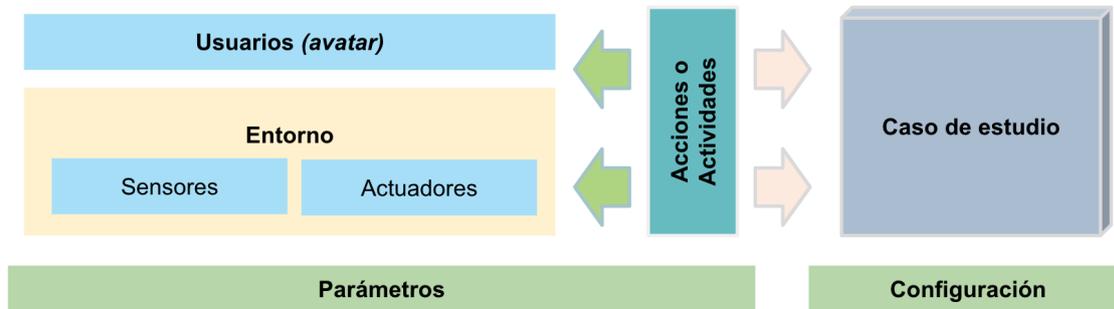


Figura 3.17: Relación de las actividades con el caso de estudio.

3.4.6.2 Descripción de las actividades

Las actividades que se modelan en la simulación, deben describirse con la siguiente información:

- **Nombre** n_i : descripción breve y clara de la actividad.
- **Orden** or_i : el orden en el que se ejecutará la actividad a lo largo de la línea temporal de la simulación.
- **Prioridad** pr_i : es un valor entre 1 y 10 que establece la prioridad de una actividad para que sea llevada a cabo en la simulación. Esta escala se traduce en un valor PR_i a través de la ecuación $PR_i = (10 - pr_i)$. Cuanto mayor sea la prioridad, mayor será la probabilidad de que la actividad ocurra en el orden establecido durante la simulación. Una prioridad baja le quita importancia a la actividad y aumenta la probabilidad de que ésta no ocurra.
- **Peso** we_i : es el grado de importancia que tiene una actividad en relación a otra. Este atributo es especialmente útil cuando hay más de una actividad en el mismo período de tiempo con la misma prioridad.
- **Agente en la simulación** a_i : indica qué elemento en la simulación ejecuta la actividad en un momento dado, por ejemplo: el usuario, un objeto del entorno, un dispositivo, etc.
- **Salida esperada** ou_i : indica concretamente qué ocurre en la simulación si una actividad se lleva a cabo o no. Por ejemplo, para la actividad “*el usuario camina hacia la puerta*”, se puede establecer como salida esperada la ejecución de una acción como “*caminar*”, “*caminar hacia un destino de la casa*” o “*ubicarse cerca de una zona dentro de la vivienda*”. La *salida esperada* es la característica que se utilizará para determinar si la actividad ocurre o no, de ahí su importancia en todo este proceso de pruebas.

3.4.6.3 Definición de las acciones o actividades

Definir las acciones o actividades implica describir el comportamiento de los elementos del caso de estudio. Cuando se quiere describir el comportamiento del entorno y la tecnología, se definen concretamente acciones que ocurren sobre dichos elementos. En el caso del entorno pueden definirse las acciones que recaen sobre los objetos que la conforman: “*se enciende/apaga la luz*”, “*el objeto x cambia de estado o posición*”, *entre otros*. En el caso de la tecnología se podrán definir acciones como: “*se activa/desactiva el sensor*”, “*el dispositivo captura datos*”, “*el dispositivo se activa/desactiva a las n horas*”, *etc.*. Las acciones del usuario, por su parte, están representadas por las actividades físicas que este realiza para interactuar con el entorno o con la solución. Los pasos que se deben seguir para describir las actividades son los siguientes:

Paso 1: *Crear una descripción general:* como se puede ver en la Tabla 3.10, a partir de una actividad rutinaria del usuario (*escenario típico*) como “*se levanta por las mañanas a las 5:00am y luego va al baño*”, se pueden obtener los distintos atributos que la describen. El primero de ellos es el nombre de la actividad, el cual describe de forma breve y concisa en qué consiste ésta. Le sigue el orden en que se ejecuta, que en este caso al ser única, indica que será la primera y última actividad en llevarse a cabo. La prioridad asignada (*siendo 1 el valor más alto*), en este caso interesa que la actividad se ejecute en la simulación asignándole así la prioridad más alta. El peso que tiene la actividad (*en este ejemplo concreto esta información no es relevante porque solo hay una actividad*). La ubicación dónde se lleva a cabo en el escenario y finalmente la salida que se espera de ella. El indicador usado en este caso es que el usuario realiza algún movimiento físico para saber que no está aún dormido después de una hora concreta, por ejemplo, “*después de 5:00 a.m.*”.

Actividad	Orden	Prioridad	Peso %	Ubicación	Salida
el usuario se despierta a las 5:00 a.m. y luego va al baño	1	1	100	habitación	el usuario se mueve a las 5:00am

Tabla 3.10: Primera aproximación: actividades que se simularían en el caso de estudio de ejemplo.

Paso 2: *Crear una descripción más detallada:* en este paso se obtiene un mayor nivel de detalle en la descripción de la rutina del usuario mediante la identificación de actividades más simples. En este punto, podría ser útil conocer, por ejemplo, (i) *si el usuario se despierta a las 5:00am*, (ii) *si después de despertarse permanece aún en la cama* o saber (iii) *cuánto tiempo ha pasado el usuario despierto antes de ir al baño*. Ante estas nuevas actividades, y con el objetivo de poder comprobar que cada actividad se ejecuta de forma separada, se obtiene un nuevo conjunto de actividades como el que se muestra en la Tabla 3.11.

Actividad	Orden	Prioridad	Peso %	Ubicación	Salida
el usuario se despierta a las 5:00 a.m.	1	1	100	habitación	el usuario se mueve a las 5:00am
el usuario va al baño	2	1	100	habitación	el usuario camina, destino: el baño

Tabla 3.11: Segunda aproximación: actividades que se simularían en un caso de estudio más detallado.

Paso 3: *Diferenciar las actividades entre sí:* la descripción de cada actividad por separado facilita tanto el seguimiento de tareas más simples como la especificación de salidas esperadas más concretas y fáciles de validar. Cuanto más simple la tarea, más factible es poder encontrar un criterio de validación que permita determinar el grado de satisfacción con dicha actividad. Por otra parte, el contexto del usuario puede ser determinante para crear el caso de estudio. Si el caso propuesto girase en torno a un paciente mayor a quien se le dificulta la tarea de levantarse de la cama, esta nueva condición obligaría al grupo de estudio que analiza el caso a determinar actividades mucho más simples que las hasta ahora vistas.

Como se observa en la Tabla 3.12, en un caso como el anterior el cuidador o los profesionales médicos podrían considerar oportuno saber si el paciente antes de levantarse pasa un tiempo t_i sentado en la cama, o medir el tiempo que le cuesta al usuario ponerse en pie desde que está sentado en la cama.

Actividad	Orden	Prioridad	Peso %	Ubicación	Salida
el usuario se despierta a las 5:00 a.m.	1*	1	100	habitación	el usuario se mueve a las 5:00am
el usuario se sienta en la cama	2	2	50	habitación	posición: sentado
el usuario se levanta	3	2	50	habitación	posición = de pie
el usuario camina al baño	4*	1	100	baño	acción caminar, destino: baño

Tabla 3.12: Tercera aproximación: actividades que se simularían con mayor detalle en un caso de estudio.

Note el énfasis que se hace en las actividades con orden 1* y 4* en la Tabla 3.12. Una actividad más general como la planteada en la Tabla 3.10, podría resumir las actividades del 1 al 4, pero su uso no permitiría dar mayor detalle a acciones más simples que realiza el usuario. Si solo se utilizasen las dos actividades de la Tabla 3.11, solo se podrían evaluar acciones con un bajo nivel de detalle perdiendo así la oportunidad de estudiar casos más puntuales como los vistos en este último ejemplo (ver Tabla 3.12). En este caso se pueden evaluar aspectos como: *si el usuario despierta a las 5:00am, si se sienta en la cama y posteriormente se pone en pie, cuánto tiempo dura sentado, cuánto tiempo tarda en ponerse en pie* y finalmente *si camina hacia el baño y cuánto tiempo le toma moverse de un sitio a otro*.

Paso 4: *Establecer la prioridad y el peso de cada actividad:* al observar las actividades que se muestran en la Tabla 3.12, la prioridad es la misma para las actividades 1 y 4, mientras que para las actividades 2 y 3, existe una menor relevancia debido a que alguna de ellas podría no ocurrir o llevarse a cabo (el usuario no está obligado a sentarse en la cama antes de ponerse en pie). Como estas dos actividades no se espera que ocurran en el orden propuesto, tampoco es de esperar que una de ella tenga más importancia sobre la otra, de ahí la razón que el peso de ambas sea exactamente el mismo y las dos en conjunto mantengan una probabilidad total del 100%. El peso por su parte, es un atributo que puede usarse en función de los problemas de movilidad del usuario. Retomando el ejemplo anterior, en el caso que el paciente tuviese dificultades para levantarse de la cama, el peso de la actividad número 2 tendría que ser mayor que el peso de la actividad número 3, por ejemplo, un 70% – 30% respectivamente. Esta asignación garantizaría que la actividad

con mayor peso aparecería en el escenario simulado. Si la dificultad del usuario está en la acción de ponerse en pie, un peso mayor tendría que asignarse a la actividad número 3, por ejemplo, un 30 % – 70 %.

Una vez recogida toda la información de las actividades, ésta se incluye dentro de la descripción del escenario, utilizando para ello la plantilla que se muestra en la Tabla 3.13. En dicha plantilla, se indican los aspectos más relevantes del escenario, las características del comportamiento del usuario y la lista de actividades que se ha definido previamente.

Nombre del escenario			
Descripción	...		
Acciones o Actividades	...	Usuarios (avatares)	...
Duración	...	Ubicación	...
Monitorización	Si/No	Áreas	...
Problemas de movilidad	Si/No	Problemas específicos	...

Tabla 3.13: Plantilla propuesta para describir el escenario de la simulación.

3.5 Transición del caso de estudio a la simulación

Para hacer la transición entre el caso de estudio y la simulación, se parte de las especificaciones definidas en el caso de estudio, las cuales describen cada escenario creado: una situación típica o cotidiana del usuario, una situación en la que el usuario se enfrenta al problema y una última situación en la que la solución propuesta, brinda la asistencia (ver Figura 3.18). No es un requisito indispensable definir exactamente tres simulaciones, el número de escenarios simulados dependerá de lo compleja que sea la solución o de las necesidades del equipo al valorar cada escenario por separado.

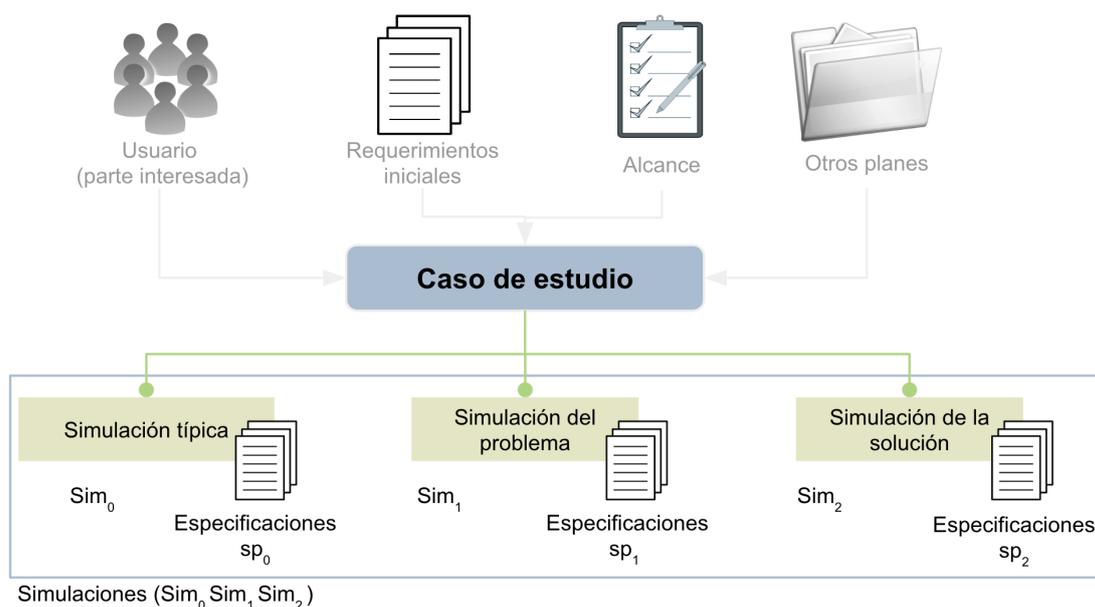


Figura 3.18: Especificaciones del caso de estudio para generar la simulación.

En las especificaciones de cada escenario, se encuentran todos los aspectos que se deben añadir en cada simulación, incluyendo los modelos de comportamientos que se trasladarán a cada elemento simulado. La relación entre los elementos de la simulación y cada especificación definida, también se incluye como parte de la descripción en los escenarios. Con esta información en mano, el desarrollo de la simulación se lleva a cabo mediante el marco de herramientas software AIDE [AID14], basándose en el uso de un lenguaje de modelado para escenarios de Inteligencia Ambiental, SociAALML [SSP15].

3.5.1 Elementos de la simulación

Al igual que en el caso de estudio, para la simulación es necesario identificar los componentes principales que la definen, los cuales se pueden representar con la ecuación $SIM \equiv AG_i + EN + OB_i$, donde AG_i agrupa los usuarios o avatares, los sensores y los actuadores. La simulación se compone de n agentes AG_i , de un entorno simulado EN y de objetos OB_i que hacen parte del entorno y/o de los agentes. Un agente AG_i es aquel elemento que participa o tiene un papel relevante en la simulación (posee un comportamiento definido) y que puede dar información sobre su propio estado.

El entorno EN por su parte, describe los elementos ambientales del escenario mientras que los objetos OB_i , solo ofrecen una utilidad muy limitada o específica en el escenario (objetos de decoración, de uso simple o puntual como una silla, la cama, una mesa). Los objetos no aportan datos relevantes sobre su estado o su contexto. Dentro de la categoría de agente, también se incluyen aquellos elementos de la simulación que, además de hacer parte del decorado en el escenario, se les asocian comportamientos predefinidos en el caso de estudio para representar determinados elementos de una solución asistiva, los cuales podrán interactuar de alguna forma con el usuario. Los elementos que modelan el comportamiento en la simulación son los agentes que se describen a continuación:

- **Personajes (Avatar):** son las entidades que simulan a las personas o usuarios y los que llevan a cabo las actividades humanas necesarias en el caso de estudio simulado para representar la interacción con OB_i y con otros AG_i . La representación gráfica de las actividades que ejecutan los personajes o aquellos objetos del entorno con algún tipo de movimiento en la simulación, se llama animación. Una o varias animaciones pueden representar una o más actividades del personaje, por ejemplo: *correr, caminar, saltar, etc.* En el caso de los objetos: *un grifo abierto, luz que se enciende o apaga, la televisión que muestra imágenes, la puerta que se cierra o se abre, entre otros.*
- **Entorno:** es el escenario que se modela a partir de un espacio físico real relacionado con la problemática del usuario (su vivienda normalmente). En este escenario se llevan a cabo todas las acciones o actividades del personaje en representación del usuario y se sitúan todos los dispositivos tecnológicos de interés para el caso de estudio, concretamente los actuadores y sensores. Los *Actuadores* permiten a los personajes de la simulación interactuar con el entorno, por ejemplo: *interruptor de luz, control remoto de televisión, grifos para abrir/cerrar, puertas, etc.*

- **Sensores:** permiten obtener datos del contexto de un escenario simulado. Esta información podrán generarla los personajes, el entorno o la interacción de ambos. En el caso de la monitorización a largo plazo, el objetivo es detectar las actividades cotidianas que realizan las personas mayores en el entorno y para ello, se hace necesario utilizar sensores que puedan reconocer los cambios de actividad y que diferencien las distintas actividades que puedan realizar los usuarios. En una solución asistiva, los sensores que se utilicen deben caracterizarse por ser pequeños, no intrusivos, de bajo coste y de baja potencia para que puedan ser usados de forma continuada dentro de la solución asistiva.

3.5.1.1 Características del usuario a modelar

Las características del usuario que son modeladas, tienen relación con información relacionada a su perfil tales como la edad, el género, el comportamiento, las características físicas y su localización dentro del entorno. El comportamiento se modela mediante la actividad física que este realiza, la cual puede representarse como movimientos o gestos simples: *subir o bajar las manos, extender el brazo, flexionar las piernas, episodios de temblores en manos y cuello*. De igual forma, se asocian al usuario acciones más completas que han sido predefinidas programáticamente, tales como “beber”, “comer”, “correr”, “saltar”, “caminar”, “llamar la atención”, “caerse”. En adelante, se hará referencia a cada acción mediante la etiqueta que identifica la animación correspondiente, estas son: “*drink*”, “*eat*”, “*run*”, “*jump*”, “*walk*”, “*wave attention*” y “*fall down*”. Un resumen de las características del usuario a modelar se muestran en la Figura 3.19.

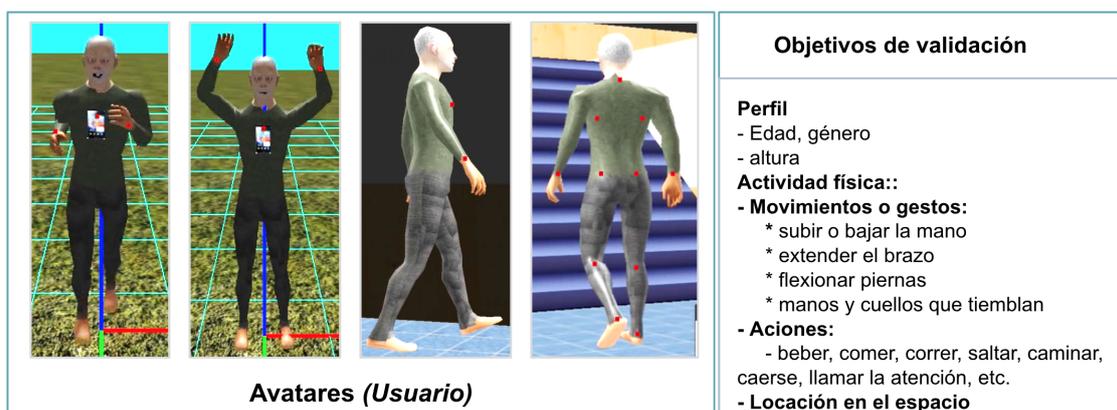


Figura 3.19: Aspectos a validar en el elemento “usuario” de la simulación.

3.5.1.2 Características a modelar del entorno

Los elementos del entorno están organizados en dos grupos: por un lado están *los elementos decorativos* o *estáticos* cuyo único papel en la simulación es el de adornar y decorar el entorno, por ejemplo: *cuadros, flores, alfombras*, etc. Por otra parte, están los *elementos dinámicos*, los cuales si disponen de características interactivas con las que el usuario y los demás objetos pueden interactuar, por ejemplo: *televisión, sillas, puertas*,

grifo, cama, entre otros. Los *elementos dinámicos* por su parte, deben disponer de sensores que permitan su monitorización, independientemente de que los lleve consigo el usuario en la simulación o los elementos del entorno (ver Figura 3.20).



Figura 3.20: Objetivos validables en el elemento “entorno” de la simulación.

3.5.1.3 Configuración y parametrización

Además de relacionar los elementos de la simulación con cada especificación, también puede ser necesario añadir información que permita parametrizar y configurar a los agentes. Esto último es especialmente relevante en el caso de los sensores utilizados en la simulación, los cuales son virtualizados y desplegados en la solución asistiva, para simular la monitorización y seguir el comportamiento de los objetos en el entorno. Es imprescindible que cada objeto animado dotado de interactividad en la simulación, disponga de algún sensor que recoja y exponga información de su actividad y estado. Cuando se definan todos los sensores que existirán en la simulación, se deben especificar los siguientes atributos:

- **Flujo de eventos de cada sensor (f_{si}):** cada sensor independientemente de su tipo, genera un flujo “constante” de eventos f . Estos flujos se podrán utilizar o bien por separado o bien combinándolos pero en ambos casos es necesario que cada uno

disponga de su propia identificación y formalización, por ejemplo: el sensor s_1 tiene un flujo de eventos f_{s_1} , s_2 tiene asociado el flujo f_{s_2} y s_3 el flujo f_{s_3} . Para representar los flujos combinados (s_1, s_2, s_3) se usará $f_{\{s_1, s_2, s_3\}}$.

- **Muestreo del sensor (Hz):** es la frecuencia con que el sensor genera valores de lectura del entorno. Cuántas más medidas por segundo genere el sensor mayor será su frecuencia. Para el desarrollador, este valor es importante para ajustar por ejemplo la segmentación del procesamiento de los datos.

3.6 Resumen

Con respecto a las soluciones asistivas, en este capítulo se han descrito las tareas necesarias para construir la simulación a partir del caso de estudio que representa la solución asistiva. La simulación se ha propuesto como un elemento que recoge la especificación de requisitos cuyo propósito es conseguir que distintos actores participen en el ciclo de desarrollo. Para que la participación sea efectiva, se propone el uso de artefactos que sean entendibles por todos los participantes y que contribuyan a construir la solución. Esto viene motivado porque la mayoría de las soluciones asistivas que se suelen implantar no son apropiadas para los casos particulares de las personas, y deben adaptarse a cada uno, haciendo necesario que éstas se involucren en el proceso de desarrollo. En este trabajo se parte de la idea de los VLL desarrollada en el proyecto *Social Ambient Assisted Living* (SociAAL) y busca ir más allá en lo que respecta a los aspectos colaborativos y en la mejora de técnicas que faciliten la implantación en hardware real de lo que se ha desarrollado en el entorno virtual o simulado.

De igual manera, se indican las características tanto tecnológicas como del dominio de la asistencia que se deben tener en cuenta a la hora de diseñar y construir un servicio asistivo. Estas características, se definen a partir de los requisitos que establecen el tipo de servicio que se quiere ofrecer al usuario y el apoyo tecnológico que este usará. Cuando se crea la simulación, estas mismas características se trasladan al entorno simulado, provocando que la simulación pase a ser un elemento de especificación que represente la solución.

Para que la metodología propuesta sea útil, se parte de dos elementos claves en el proceso de creación y validación de la solución asistiva, estos son:

- El uso de casos de estudio que describan las características del usuario, de su entorno y de la tecnología de apoyo que usa la solución asistiva. Para construir el caso de estudio, se proponen tareas y procedimientos que ayudan en identificar los aspectos más relevantes de la solución, a partir de los requisitos obtenidos. Estas tareas y procedimientos pueden emplearse junto a otra metodología de proyectos para desarrollar la solución final.
- La simulación como elemento de especificación que recoge los requisitos del caso de estudio y a su vez, ofrece los medios necesarios para que se use como un elemento de validación de dichos requisitos.

Capítulo 4

Creación de los casos de prueba de la simulación

En el Capítulo 3 se ha indicado cómo se definen los casos de estudio de las soluciones asistivas y cómo a partir de ellos, se generan las simulaciones que se usarán como especificación de la solución. En este capítulo, el objetivo es construir los casos de prueba necesarios para validar que la simulación cumple con las características de la solución que le han sido asignadas. En la Sección 4.1, se describe el entorno de prueba que se aplica sobre las simulaciones para su validación. Las pruebas unitarias y los criterios de validación existentes para evaluar las características de la simulación, se presentan en la Sección 4.2. Por último, en la Sección 4.3 se muestran los pasos necesarios para construir el caso de prueba a partir de las pruebas unitarias identificadas y de los criterios de validación que se aplicarán sobre los escenarios simulados del caso de estudio.

4.1 Pruebas sobre la simulación

El uso de la simulación 3D como especificación, genera nuevas cuestiones en el proceso de desarrollo que deben ser resueltos de alguna manera, por ejemplo:

- *¿Cómo se puede comprobar que la simulación 3D es una especificación válida del caso de estudio?*
- *¿Cómo se puede verificar que la solución asistiva se ve reflejada en la simulación?*
- *¿Los elementos de la simulación se están comportando como deberían?*
- y no menos importante *¿Cómo garantizar que cada nueva iteración de la simulación es coherente con las versiones anteriores?*

La contribución de este apartado busca precisamente resolver estos interrogantes, describiendo para ello cómo se podría aplicar una formulación informal de pruebas unitarias sobre los distintos elementos de la simulación, para validar las simulaciones 3D como especificación global o parcial de una solución AAL. La manera de utilizar las pruebas unitarias, es mediante la definición de *casos de prueba* a partir del *caso de estudio* que describe la solución AAL.

Para validar la simulación, los casos de prueba se aplican según el alcance establecido de la simulación sobre la solución asistiva, descrito en la Sección 3.2. En la fase definición de la solución se define un **entorno de pruebas de la simulación**, donde se valida si se cumplen con los requisitos establecidos en el caso de estudio. En la fase de pruebas del sistema y para un **entorno de pruebas en real**, se realizan validaciones cuyo objetivo es comprobar si la simulación puede ser empleada como especificación final que sirva de guía para implementar la solución en un entorno real. Este segundo entorno de pruebas sólo será posible si la simulación y el caso de estudio han pasado previamente por la validación en la fase de definición.

4.1.1 Herramientas de apoyo

Para la construcción del entorno de pruebas propuesto se utilizan un conjunto de herramientas que sirven de apoyo en todo el proceso de validación de las simulaciones. Algunas de estas herramientas son el fruto de otros proyectos de investigación que hacen parte del trabajo previo de esta Tesis. El propósito de ellas es permitir la definición de los modelos de comportamiento y las secuencias de actividades en los escenarios simulados, generar la simulación y añadir las características asistivas a dichos escenarios. A continuación se describe este grupo de herramientas:

- **SociAALML** [CSGS15]: es la herramienta de edición con que se generan los modelos de comportamiento que se han definido en el caso de estudio. Permite definir la secuencia de actividades de los usuarios, del entorno y las interacciones.
- **PHAT Simulator** [SSP15]: es simulador 3D necesario para construir la simulación a partir de los modelos de comportamiento generados previamente. La simulación es el resultado de procesar los modelos creados con SociAALML.
- **AIDE** [SP19]: es el framework de la herramienta *Physical Human Activity Tester Simulator* (PHAT) y se basa en el desarrollo temprano y en la creación rápida de prototipos de sistemas AmI.

Como parte de la propuesta de esta Tesis, las herramientas que se describen a continuación fueron construidas para dar soporte a lo que es la definición de los casos de estudio que definen la simulación y para construir el entorno donde se ejecutarán los diferentes casos de prueba para validar la simulación. Las herramientas son las siguientes:

- **Hack4People**¹: plataforma web en la que se pueden llevar a cabo la técnica de grupos focales online para definir el caso de estudio a simular.
- **ColosAAL Framework**^{2 3}: es la arquitectura propuesta en este trabajo para que los desarrolladores construyan el entorno de pruebas necesario que valide y verifique los modelos de comportamiento que se han incluido en la simulación. Los ejemplos de prueba de las simulaciones están disponible en la plataforma Github⁴.

¹Plataforma web: <http://grasiagroup.fdi.ucm.es/hackwithpeople/index.php>

²Repositorio sistema HAR: <https://github.com/mfcardenas/phant-sim-classificator.git>

³Repositorio framework: <https://github.com/mfcardenas/colosaal-fcepevent.git>

⁴Repositorio de ejemplos: <https://github.com/mfcardenas/phant-sim-tester-examples.git>

4.1.2 Construcción del entorno de pruebas

Como se observa en la Figura 4.1, el entorno de pruebas se compone de tres elementos principales: el caso de estudio y la simulación asociada, las pruebas unitarias que validan cada requisito y los casos de prueba que agrupan y organizan dicha pruebas unitarias.

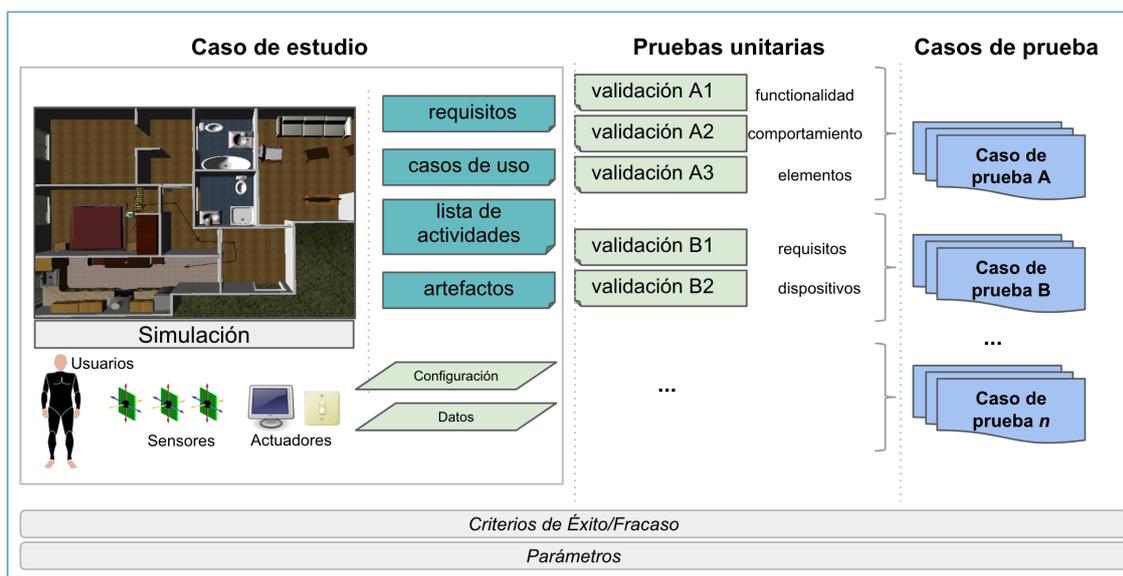


Figura 4.1: Esquema general del entorno de pruebas para el caso de estudio.

Para construir el entorno de pruebas, los ingenieros parten del caso de estudio y de la simulación definida según lo descrito en la Sección 3.4. De ambos elementos se tienen en cuenta los usuarios que participan directamente en la solución asistiva (ver Sección 3.4.1), los requisitos funcionales que describen las actividades que se esperan en la simulación y las especificaciones técnicas de cada elemento simulado (ver Sección 3.4.2). Asimismo, se identifican los elementos del dominio de la asistencia que enmarcan las situaciones recurrentes que deben ser validadas en la simulación, como las descritas en la Sección 3.4.3.

Las pruebas unitarias, por su parte, van asociadas directamente a los casos de uso que se han establecido en la fase de identificación de requisitos, según las tareas mencionadas en la Sección 3.4.5. Cada caso de uso se centra en un aspecto de la solución que interesa ser validado en la simulación. Dentro de estos aspectos están las acciones que se describen en la *lista de actividades*, la cual se ha obtenido como resultado de estas tareas (ver Sección 3.4.6).

Después de definir las pruebas unitarias el siguiente paso es crear los casos de pruebas que agruparán dichas pruebas, bajo algún criterio específico como la validación de un requisito puntual, validar una situación recurrente, la ejecución de una actividad del usuario, un elemento de la simulación, entre otros. El conjunto de validaciones dan cuerpo a lo que es el *caso de prueba* y los resultados de todos los casos que se definan, siguiendo los criterios de éxito/fracaso establecidos por los expertos, es lo que finalmente determinará si el caso de estudio es coherente con la simulación y si cumple con las expectativas del usuario como un escenario adecuado para la solución final.

4.2 Pruebas unitarias

Las pruebas unitarias se aplican sobre cada “*elemento*” de la simulación que ha sido identificado previamente, según las indicaciones de la Sección 3.3. Cada prueba es responsable en la simulación de validar una *actividad*, utilizando los criterios de validación que se han especificado para cada uno de ellos. La forma de definir la prueba unitaria se basa en las características del elemento en la simulación, las cuales a su vez, se asocian a distintos criterios de validación. En la Tabla 4.1 se puede observar qué características se necesitan para establecer los criterios de validación sobre dichos elementos.

Simulación				Criterios de prueba	
Elemento	Nombre		Salida esperada	Estado del elemento	
	Orden (1)			Localización	
	Orden (2)			Actividad	
Validación de actividad	Prioridad			Criterio de ejecución: ocurrencia, duración, repetición.	
	Peso			Según la Prioridad y el Peso	
Tipo de sensor (datos)	Sensores ambientales (<i>simple</i>)			Estado del entorno	
	Sensor inercial (<i>complejo</i>)			- Aceleración del usuario (actividad física) o de objetos	
Datos del sensor/actuador	Estado	Posición	Cambio	Comportamiento simple de los datos	
	Actividad física			comportamiento complejo de los datos	
Especificación técnica (características de la asistencia)	Número de sensores			- número de sensores, posición y cambios de estado	
	Flujo de datos			datos prueba	
	Frecuencia			pruebas de frecuencia	
	Ventana de tiempo			tamaño de la ventana de tiempo (<i>precisión, exactitud</i>)	
	Ventana de datos			tamaño de la ventana de datos (<i>precisión, exactitud</i>)	

Tabla 4.1: Elementos de la simulación con sus respectivas pruebas unitarias.

En el caso del elemento propiamente dicho, a partir del nombre, del orden de ejecución y de la salida esperada se pueden establecer los criterios de validación asociados al estado de dicho elemento, su posición en el entorno y la actividad que este ejecuta. Para la validación de las actividades, la información de partida es la prioridad, el peso y la salida esperada de cada actividad descrita en el *listado de actividades*. Los criterios de validación que se aplican en este caso, se asocian al orden y a diferentes criterios de ejecución de la actividad tales como su ocurrencia, la duración, las repeticiones en el tiempo y el fracaso de la misma. En lo que respecta a los sensores, los criterios de validación se aplican sobre los datos que éstos generan en función de su tipo.

En el caso del sensor ambiental, la prueba unitaria solo puede validar la lectura que este arroja, mientras que en el caso de los sensores inerciales, la validación se aplica sobre la aceleración del movimiento del objeto que porta dicho sensor. Cuando se trata de validar el comportamiento de los datos del sensor, en función de su tipo, se podrán definir pruebas unitarias que validen no solo el estado, la posición y los cambios de las lecturas del sensor sino también comportamientos más complejos como la actividad física del usuario. En este último caso, el “*usuario*” es el elemento de la simulación que lleva consigo el sensor inercial.

Las pruebas unitarias asociadas a las especificaciones técnicas y a las características de la solución, se asocian a criterios muy concretos de la red de sensores como la comprobación del número de dispositivos en la solución, la posición de éstos en el entorno, los cambios de estado, su actividad y la frecuencia de muestreo. Pruebas unitarias más específicas se pueden aplicar sobre el conjunto de datos generados y sobre distintos parámetros que pueden ser utilizados para analizar esta información, entre ellos la ventana de tiempo y la ventana de datos.

4.2.1 Tratamiento de los datos

Una vez puesta en marcha la simulación, de inmediato la red de sensores genera datos de forma continua. Como se observa en la Figura 4.2, estos flujos de datos se caracterizan por ser continuos en el tiempo mientras dura la simulación. En este punto, es necesario introducir tres conceptos que son claves para definir las pruebas unitarias: **instante de tiempo** t_i , **ventana de tiempo** w_t y **ventana de datos** w_d .

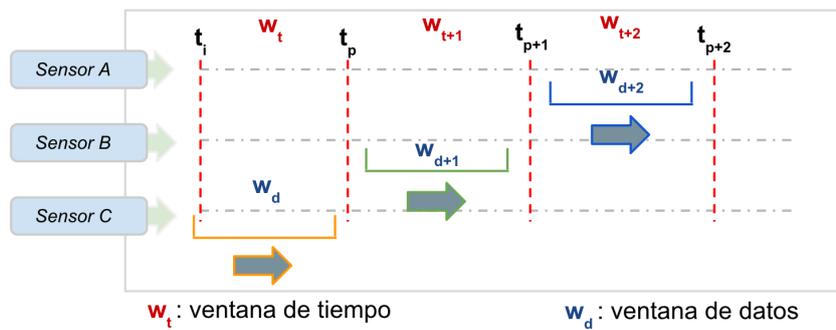


Figura 4.2: Descripción gráfica de ventana de tiempo y ventana de datos.

Un instante t se define como un momento puntual en la línea de tiempo de la simulación, cuyo valor lo conforma un número natural $t \rightarrow instant \in \mathbb{N}$. Una ventana temporal, por su parte, es un intervalo de tiempo $w_t = (t_i, t_p)$, donde $t_p = t_i + s, s \in \mathbb{N}$, donde s determina el tamaño de la ventana. Partiendo de estas ecuaciones, la *ventana de tiempo* w_t hace referencia a un intervalo fijo de tiempo entre t_i y t_p , utilizado para procesar flujos de datos de diferentes dispositivos. Cada dispositivo puede utilizar una ventana temporal diferente según su velocidad de muestreo [GGR16] o la granularidad de sus datos. La *ventana de datos* w_d por su parte, agrupa los eventos producidos dentro de la ventana de tiempo hasta que éstos sean utilizados por el proceso correspondiente [ABB⁺16].

En el momento que se ejecuten las pruebas unitarias, el modelo propuesto ejecutará las reglas o consultas *Structured Query Language* (SQL) correspondientes sobre las *ventanas de datos*, utilizando los criterios que se hayan establecidos en la misma prueba [DFOO11].

4.2.2 Criterios de validación de los datos

Las pruebas unitarias asociadas a los datos, se llevan a cabo a partir del tipo y la frecuencia de datos que genera cada actividad. Los datos son el medio utilizado para analizar las actividades de la “*lista de actividades*” y la forma de obtener dichos datos es

mediante la red de *sensores o actuadores* ubicados en toda la simulación. Estos monitorizan constantemente el estado del elemento que realiza la actividad y su participación en la simulación. Con ayuda de las herramientas de apoyo utilizadas en este proyecto (ver Sección 4.1.1), los sensores distribuidos en el entorno facilitan la información mediante dos tipos:

- **Tipo básico** o simple, equivalente a un único valor con una frecuencia determinada, por ejemplo: la lectura de temperatura, la luminosidad del entorno, el estado de un actuador o cualquier otro que se pueda medir.
- **Tipo registro** o complejo, con más de un valor como lectura, por ejemplo: la posición en el espacio del objeto (x, y, z) donde $(x, y, z) \in \mathbb{R}^3$, la aceleración del objeto (a_x, a_y, a_z) donde $(a_x, a_y, a_z) \in \mathbb{R}^3$.

4.2.3 Criterios de validación del comportamiento de los datos

Las pruebas unitarias asociadas al comportamiento de la actividad se basan en la validación que identifica el estado final del objeto que realiza la acción. Es decir, una vez se ejecuta la actividad, las pruebas unitarias podrán comparar qué ha ocurrido con el objeto cuando este haya finalizado la ejecución de dicha actividad. En este caso se han identificado dos tipos de validaciones concretas:

- *Un comportamiento simple* que describe el estado, la posición y los cambios de estado que sufre el objeto dentro del entorno.
- *Un comportamiento complejos* que describe las acciones físicas o de movimiento que llevan a cabo los objetos, entre ellos los personajes.

En el caso del comportamiento simple, ejecutar una actividad puede generar un cambio de estado o posición del objeto en la simulación que se rige con la máquina de estados finita simple que se muestra en la Figura 4.3.

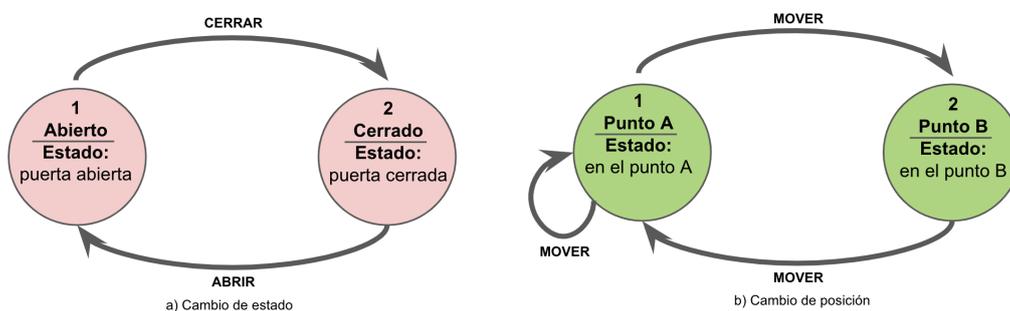


Figura 4.3: Máquina de estado para los cambios de estado y de posición en los elementos en la simulación.

Después de ejecutar la actividad, solo es relevante conocer el estado final del objeto para interactuar con él. Por ello, en este tipo de validaciones los sensores o actuadores solo se limitan a informar los estados finales del objeto (la luz está encendida o apagada, la puerta está abierta o cerrada, el control remoto está en un sitio u otro). En el caso del comportamiento complejo, la actividad está compuesta por datos con un

comportamiento *complejo*, cuyas lecturas son *series temporales* representadas por valores que, cronológicamente, describen la ejecución de movimientos o actividades físicas en un período t_0 y t_1 .

Estas series de tiempo están basadas específicamente en lecturas de movimiento y aceleración del objeto en un plano de dos y tres dimensiones. Es decir, la señal biaxial o triaxial de aceleración se compone de una serie temporal en cada eje: una serie de tiempo específica a_x en el eje X , una para a_y en el eje Y y una más para a_z en el eje Z . Para las señales biaxiales solo se tienen en cuenta (a_x, a_y) del plano XY y para los triaxiales (a_x, a_y, a_z) del plano XYZ . Cuando el movimiento físico proviene de un objeto animado, la actividad puede ser descrita por un único conjunto de series temporales de la forma (a_x, a_y, a_z) (*un único sensor inercial*) en un período t_0 y t_1 . En el caso de las actividades físicas de las personas, es recomendable utilizar más de un conjunto de series temporales para enriquecer la información que describa la acción física (*varios sensores inerciales*).

4.2.4 Criterios de validación de las actividades

Para comprender las pruebas unitarias asociadas a los criterios de validación de las actividades, es necesario hacer referencia al concepto *ventana de tiempo* w_i (ver Sección 4.2.1). Una *ventana de tiempo* hace referencia al intervalo de tiempo fijo t necesario para procesar una secuencia de actividades. El tamaño de t y por ende el de w debe ser determinado en función de lo simulado en el entorno y de las características en los elementos que existan en él. Los criterios que se pueden aplicar para la comprobación de las actividades se describen a continuación.

4.2.4.1 Por ocurrencia

Con este criterio se busca identificar si una actividad concreta ocurre o no en determinados instantes de tiempo, como se muestra en la Figura 4.4. Los criterios en este caso son:

- Una actividad concreta ocurre antes de un instante t_i .
- Una actividad ocurre en un intervalo de tiempo específico w_i .
- Una actividad concreta ocurre después de un instante t_{i+1} .
- Una actividad concreta ocurre en distintas ventanas w_i, w_j, w_k .

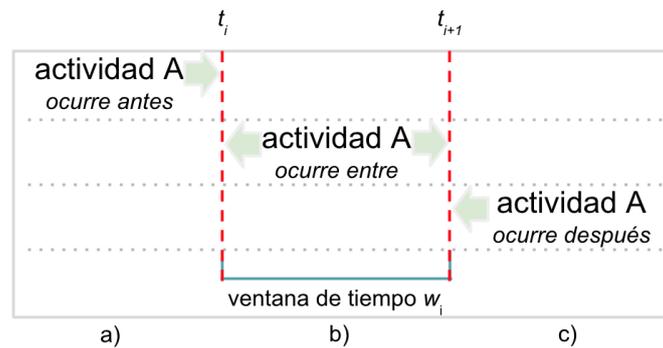


Figura 4.4: Ocurrencia de una actividad a lo largo del tiempo.

4.2.4.2 Por duración

En este criterio, lo que interesa reconocer es el tiempo que le toma a una actividad en llevarse a cabo, usando como referencia la *ventana de tiempo* w_i definida (ver Figura 4.5). Estos criterios son útiles para medir si determinadas acciones en el usuario están tomando más tiempo del que deberían (cambiar su ubicación dentro de la casa o caminar entre las habitaciones, permanecer más tiempo acostado, realizar un gesto con la mano, etc.). Los criterios para este otro caso son:

- La duración de una actividad cubre toda la ventana de tiempo w_i .
- La duración de la actividad supera el tamaño de la ventana w_i .
- La duración de una actividad cubre una fracción de la ventana de tiempo w_i .

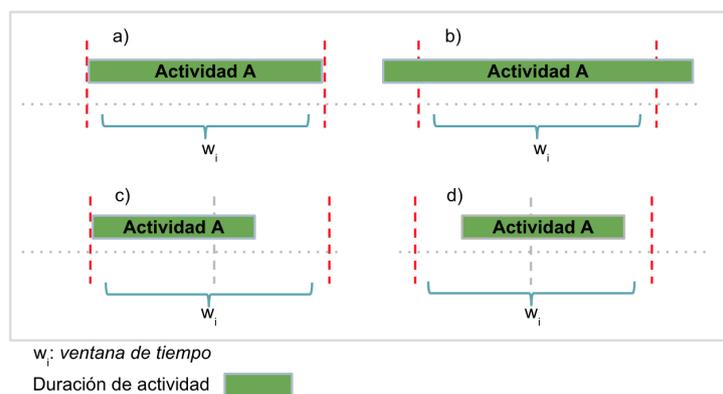


Figura 4.5: Duración de la actividad en relación a una ventana de tiempo.

4.2.4.3 Por repeticiones

Este criterio hace referencia al hecho de que las actividades se pueden repetir en el tiempo. En la Figura 4.6 se muestran los criterios identificados para reconocer las repeticiones. Las pruebas unitarias en este caso reconocerán cuándo dichas repeticiones se producen y con qué frecuencia lo hacen. Los criterios a utilizar en este caso son los siguientes:

- La actividad se repite n veces al principio de la ventana w_i .
- La actividad se repite n veces al final de la ventana w_i .
- La actividad se repite n veces dentro de la ventana w_i .
- La actividad se repite n veces fuera de la ventana w_i .
- La actividad se repite en distintas ventanas w_i, w_j, w_k .

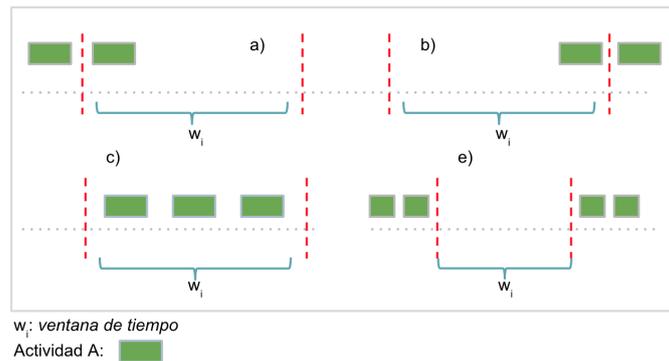


Figura 4.6: Repeticiones de la actividad en relación a una ventana de tiempo.

4.2.4.4 Por ejecución

Como se indica en la Sección 3.4.6.1, una simulación se basa en una secuencia de acciones o actividades (“*lista de actividades*”) que ocurren en un orden establecido. Es así como se define el guión que describe la simulación, el cuál recoge todo lo que acontece en lo simulado como representación de los escenarios de la solución. En la Figura 4.7 se muestra el criterio de ejecución aplicado a cada elemento de la lista de actividades.

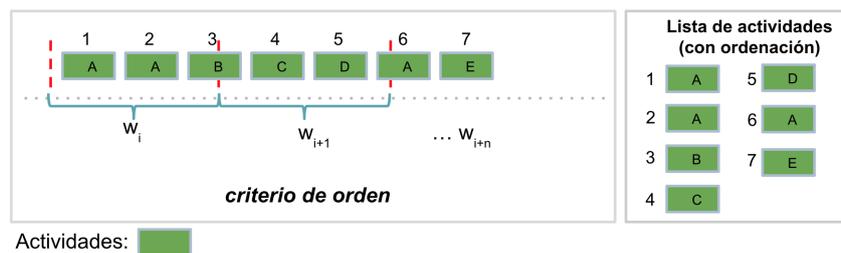


Figura 4.7: Orden en el que se ejecutan las actividades.

Las pruebas unitarias utilizan el orden de ejecución para determinar si las actividades se ejecutan en la secuencia esperada. La *ventana de tiempo* w_i es útil en este caso, para identificar las actividades próximas entre sí o las que deben ejecutarse en un estricto orden dentro de un margen de tiempo concreto.

4.2.4.5 Por reconocimiento

Un elemento importante en una solución AAL es el reconocimiento de la actividad humana, el cual permite identificar acciones concretas del usuario y adaptar el entorno

para que reaccione de forma adecuada a dichas acciones. Las técnicas de reconocimiento no son propias de los sistemas AAL por lo que se requieren mecanismos que faciliten su integración tanto con el entorno como con los usuarios. Las tecnologías que utiliza AAL incorporan una gran variedad de sensores que pueden ser usados para reconocer la actividad física de las personas.

Según el objetivo al que se quiere llegar con dicho reconocimiento, puede ser necesario implementar técnicas complejas que procesen la información de dichos sensores para identificar *patrones* en la actividad y asociarlos al contexto de la persona [DA⁺00] en un momento determinado. Uno de los criterios propuestos para validar la actividad física del usuario, es utilizar estas técnicas para reconocer la actividad en el momento que se está llevando a cabo en la simulación. Dicho criterio puede aplicarse en dos contextos posibles dentro de la simulación, bien para validar que ocurre una actividad física concreta como “caminar” o “correr”, o para validar que ocurren movimientos simples que ejecuta el usuario tales como “subir o bajar las manos”, “temblores” o “saltos”.

4.2.5 Validación de la actividad física

Estas acciones físicas del mundo real se representan mediante animaciones de ordenador en la simulación, las cuales son añadidas al personaje de forma programática y ejecutadas conforme las necesita. La principal característica que debe tener una animación es su similitud con la actividad física del mundo real, cuanto más realista sea la secuencia de gestos animados, más útil será la animación para analizar la actividad humana de los personajes en el entorno simulado.

El criterio de validación de la actividad mediante su reconocimiento, implica el uso de sistemas HAR que utilicen información de sensores inerciales posicionados sobre el usuario para identificar la actividad. La arquitectura propuesta en este trabajo soporta el uso de algoritmos de clasificación que, con una configuración determinada en los sensores, puede predecir la actividad que realiza el usuario en el escenario simulado (en el capítulo 7 se describe el sistema HAR implementado).

4.2.5.1 Ocurrencia de movimientos simples

El criterio de validar la actividad física mediante movimientos simples hace referencia a la posibilidad de descomponer las actividades en movimientos o gestos muchos más simples y fácilmente identificables mediante el uso de patrones de movimiento. La acción física de una persona se compone de un conjunto de movimientos más simples que realizan determinadas partes del cuerpo. La actividad de “caminar”, por ejemplo, se compone de gestos simples que el torso y cada extremidad realiza en un orden determinado para conseguir el desplazamiento del individuo. Como se observa en la Figura 4.8, partiendo de una animación lo suficientemente realista, la identificación de gestos simples se hace a través del cambio de posición del objeto animado sobre uno de sus ejes de rotación o mediante los desplazamientos en un único sentido. Si los gestos que se quieren analizar se llevan a cabo en un único eje del plano cartesiano, estos se podrán identificar utilizando *patrones* simples mediante el sistema propuesto.

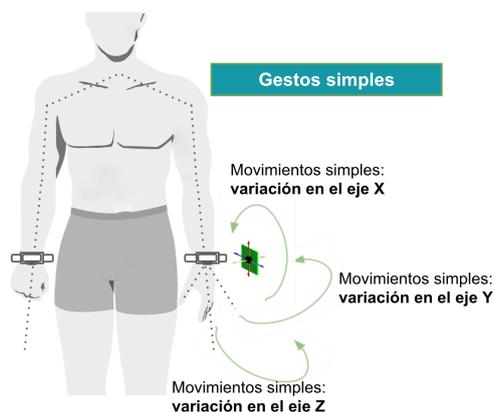


Figura 4.8: Descomposición de las actividades físicas en movimientos simples.

En la Figura 4.9 se muestra la descomposición de la actividad física “beber” en movimientos más simples, la cual se supone tiene un orden específico dentro de la *lista de actividades* del escenario. Un segundo orden en este caso, describe los gestos que conformarán la acción en sí. La descomposición natural de esta actividad, sugiere que se parte de (1) un primer gesto por parte del usuario que consiste en llevar la mano hasta el vaso o “*tomar el vaso con la mano*”. (2) El segundo gesto consiste en “*levantar la mano*” para llevar el vaso hasta la boca mientras que (3) el tercer gesto es la acción concreta de “*beber*”. (4) El último gesto simplemente es “*dejar el vaso en su sitio*” o en su defecto “*bajar la mano*”. Siguiendo esta misma filosofía, uno de estos gestos puede ser la suma de otros movimientos más simples, los cuales se ejecutarán siguiendo un “*nuevo orden*” que se usará para conocer la secuencia correcta.

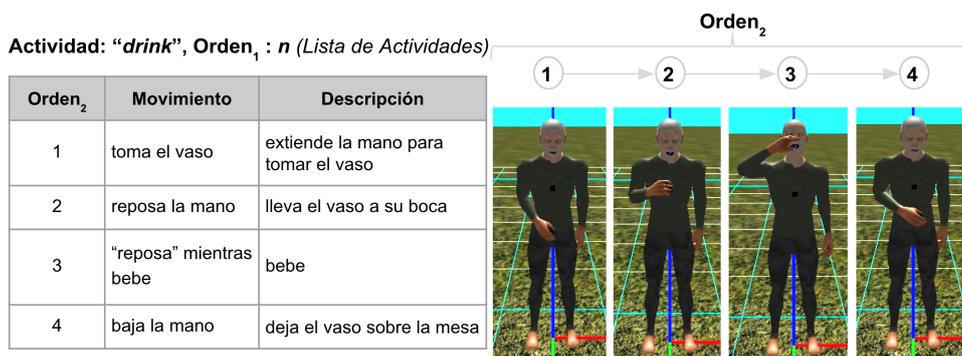


Figura 4.9: Descomposición de la actividad física en gestos simples con un único sensor.

El criterio de validación de la actividad mediante el reconocimiento de gestos simples, también es aplicable sobre actividades que demandan ser monitorizadas por varios sensores a la vez, como se observa en la Figura 4.10. En estos casos, como cada sensor está posicionado sobre una parte del cuerpo concreta, cada uno de ellos responde a una serie de movimientos que pueden ser analizados por separado. Retomando el ejemplo de la actividad “*beber*”, si se usasen distintos sensores para monitorizar al usuario, cada sensor se podría utilizar para reconocer movimientos simples que indiquen alguna característica del comportamiento que pueda ser empleada para analizar la actividad física del usuario.

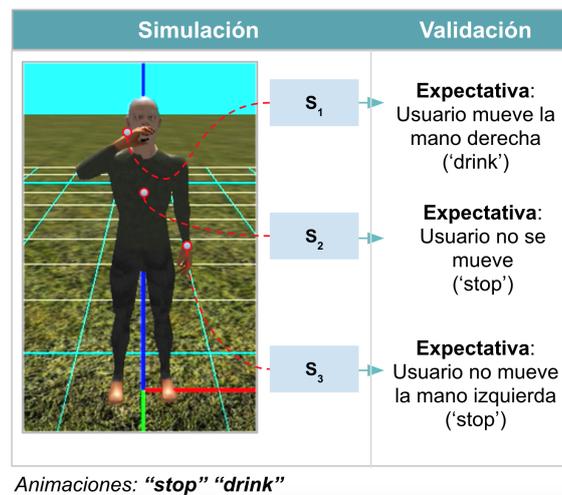


Figura 4.10: Descomposición de la actividad física en gestos simples con más de un sensor.

4.2.6 Prueba principal y prueba secundaria o de apoyo

Debido al amplio uso que se le pueden dar a las pruebas unitarias, éstas han sido clasificadas en tres tipos concretos:

- **Prueba principal:** se utiliza para validar específicamente la salida esperada de la actividad, aplicando los criterios de éxito o fracaso establecidos.
- **Prueba secundaria:** este tipo de pruebas se usan para apoyar la decisión de aceptar o no una salida esperada como un caso exitoso. Estas pruebas son necesarias porque no siempre es posible validar la salida esperada con una única prueba unitaria, lo que hace necesario usar pruebas adicionales o de apoyo que refuercen la decisión o el criterio utilizado para validar la actividad.
- **Prueba de dispositivo:** son pruebas utilizadas para verificar que los dispositivos en la simulación están activos y su comportamiento es el esperado. Cuando se aplican los criterios de validación de las actividades, las pruebas de dispositivo son útiles para comprobar con anterioridad, que los sensores o actuadores funcionan correctamente siguiendo los parámetros de configuración establecidos en el diseño de la solución.

4.2.7 Pruebas unitarias predefinidas

En la Tabla 4.1 se muestra el conjunto de pruebas unitarias predefinidas que utilizan los distintos criterios de validaciones descritos anteriormente. Cada prueba permite obtener la información correspondiente de la simulación para que los ingenieros utilicen dicha información para definir las pruebas unitarias que considere. La forma de acceder a los elementos de la simulación es a través de identificadores propios que tienen todos los objetos vinculados a la simulación.

De la misma forma, se puede acceder al comportamiento de los objetos, con el fin de determinar si la tarea principal del elemento se está llevando a cabo. Adicional a esto, también es posible ejecutar instrucciones propias del usuario bien mediante la creación de nuevas consultas (reglas) o a través de los descriptores que reutilizan consultas existentes

en el *Catálogo*. La implementación de estas pruebas, se explica con más detalle en el siguiente capítulo.

Método auxiliar	Entrada	Salida	Criterio de prueba
checkPosition	idObj,idDevice,w,w_data	true/false	element status, position, behavior simple data
checkStop	idObj,idDevice,w,w_data	true/false	all
checkValue	idObj,idDevice,w,w_data	true/false	all
checkMovement	idObj,bodyPart,#sensor,w,w_data	true/false	behavior simple/complex data
checkActivity	idObj,bodyPart,#sensor,w,w_data	(string) name activity	behavior complex data
checkActivities	idObj,bodyPart,#sensor,w,w_data	(array) name activities	behavior complex data
checkElementEnv	idSim,name,position,action	true/false	element status, position
checkAvatarEnv	idSim,name,position,action	true/false	element status, position
checkDevice	idDevice,name,position,action	true/false	all
executeRuler	idSim,idElement,rule	object	all
executeDescriptor	idSim,idElement,descriptor	object	all

Tabla 4.2: Métodos auxiliares para la comprobación del comportamiento de los elementos en la simulación.

La tarea de los ingenieros, es utilizar dichas pruebas predefinidas o definir sus propias pruebas unitarias, especificando los valores de entrada que estime oportunos para luego operar con los valores de salida en cada prueba. En la Tabla 4.3, se observan los métodos que permiten obtener datos de los elementos del entorno. Cada método dispone de los parámetros necesarios para conectar con la simulación y la salida que se espera en cada uno de ellos.

Método auxiliar	Entrada	Salida	Criterio de prueba
getSimpleData	idObj,bodyPart,#sensor,w,w_data	(object) data	behavior simple data
getComplexData	idObj,bodyPart,#sensor,w,w_data	array(object) data	behavior simple data
getElementEnv	idSim,name,position,action	array(object) element	element status, position
getAvatarEnv	idSim,name,position,action	array(object) element	element status, position

Tabla 4.3: Métodos auxiliares para la captura de datos previos.

4.3 Casos de prueba

El caso de prueba es la forma de confrontar determinadas especificaciones del caso de estudio con la simulación. La definición de los casos de prueba requiere identificar todos los elementos tanto del caso de estudio como de la simulación. Estos elementos se relacionan entre sí y, desde el punto de vista de los revisores expertos, serán lo suficientemente útiles para determinar si lo simulado se corresponde con el caso en cuestión y además cubre la asistencia que se quiere brindar con la solución final.

En la Tabla 4.4, el caso de prueba busca cubrir determinados aspectos del caso de estudio, tales como las actividades que describen la simulación, los datos generados, el comportamiento de dichos datos y las características de asistencia de la solución. Cada uno de estos aspectos, se asocia directamente a la simulación, lo que conlleva a que el caso de prueba agrupe las pruebas unitarias en función de ellos. Es decir, siguiendo esta premisa, los

casos de pruebas se basarán en estos aspectos generales para evaluar el comportamiento de la simulación, mientras que las pruebas unitarias utilizarán características más específicas de dichos aspectos. En lo que respecta a la simulación, los aspectos equivalentes al caso de estudio son las validaciones de las actividades, los sensores y sus características, los datos generados por los sensores y su comportamiento, y finalmente, las especificaciones técnicas de la simulación.

Caso de estudio				Simulación			
Actividades	Elementos en la simulación	Nombre	Salida esperada	Validación	Elemento en la simulación	Nombre	Salida esperada
		<i>Orden₁</i>				<i>Orden₁</i>	
		<i>Orden₂</i>				<i>Orden₂</i>	
		Prioridad				Prioridad	
		Peso				Peso	
Datos	Simple Complejos		Tipo de sensor	Sensor ambiental Sensor inercial			
Comportamiento (datos)	Estado	posición	cambios	Comportamiento (datos)	Datos simples		
	Actividad física				Datos complejos		
Característica de la asistencia	No invasivos Privacidad Respuesta del usuario Problemas de movilidad Aversión tecnológica			Especificación técnica	Número de sensores Flujos de datos Frecuencia Ventana de tiempo Ventana de datos		

Tabla 4.4: Resumen de la información necesaria para el caso de prueba.

Antes de construir el caso de prueba, la información del caso de estudio necesaria para dicha tarea es la siguiente:

- **La lista de actividades:** o acciones definidas en el caso de estudio con sus respectivas salidas esperadas, asociadas a los diferentes elementos de la simulación.
- **Las salidas esperadas:** de cada actividad o acción, las cuales determinan los criterios de éxito o fracaso en cada una de ellas.
- **Los criterios de aceptación:** que se deben aplicar sobre la solución asistiva, los cuales determinarán el criterio de éxito o fracaso del caso de prueba.

4.3.1 Organización de los casos de prueba

Como se ha descrito en la Sección 4.2, las pruebas unitarias validan cada actividad por separado y luego sus resultados, son utilizados en el caso de prueba para validar la solución de forma global, es decir, como un conjunto de validaciones que, al igual que las actividades, deben ejecutarse en un orden determinado. Como el número de pruebas unitarias puede llegar a ser muy grande debido a la cantidad de características que se pueden validar en la simulación, los casos de prueba deben especializarse en aspectos concretos de la simulación o en objetivos específicos de validación. Esta tarea, además de organizar las validaciones, permite tomar los criterios de aceptación de la solución y aplicarlos sobre los resultados del caso prueba. Cuando el número de casos de prueba sea

elevado, siguiendo la metodología JUnit [Bec18], se pueden utilizar suites de prueba que los agrupen para una mejor organización del entorno.

4.3.2 Creación del caso de prueba

Después de definir y diseñar las pruebas unitarias, la siguiente tarea es construir los casos de prueba con ayuda de todas las especificaciones obtenidas en el proceso de desarrollo de la solución, descritas en la Sección 3.4.6. Como se puede observar en la Figura 4.11, crear el caso de prueba lleva consigo una serie de tareas que buscan por un lado recoger toda la información necesaria para definir las pruebas unitarias sobre el entorno simulado y, por otra parte, acceder a la simulación para analizar su comportamiento en función de los parámetros de configuración establecidos.

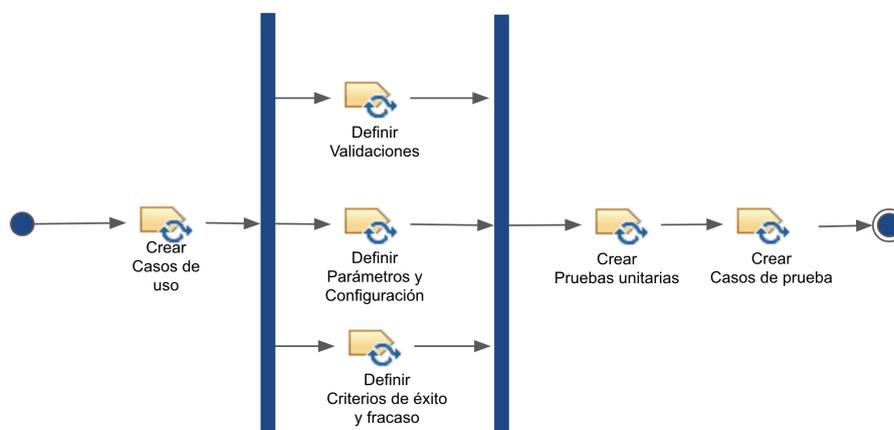


Figura 4.11: Tareas necesarias para la creación del caso de prueba.

- **Crear el caso de uso:** es la primera tarea que se propone y consiste en identificar el aspecto de la simulación que será validado, basándose para ello en las especificaciones que describen los requisitos de la solución asistida (ver Tabla 4.4).
- **Definir las validaciones:** con esta tarea se buscan establecer las validaciones que se usarán en el caso de uso para validar la simulación o la solución en sí misma. Para ello, se tendrán en cuenta la *lista de actividades*, los elementos de la simulación y los criterios de validación correspondientes (ver Tabla 4.1). De cada validación que se defina, será necesario identificar los siguientes atributos:
 - El orden de ejecución de cada validación dentro del conjunto.
 - El peso de cada una de ellas.
 - La prioridad de cada validación con respecto a las demás.
- **Identificar parámetros y configuración:** la parametrización y configuración de la simulación (ver Sección 3.5.1.3) es información necesaria para poder hacer seguimiento de ella y aplicar las validaciones correspondientes. Es por ello que una de las tareas propuestas es identificar todos los parámetros que podrían ser usados como condiciones adicionales en el caso de prueba.

- **Definir los criterios de éxito y fracaso:** dentro de la *lista de actividades* que definen la simulación, es posible encontrar las salidas esperadas de cada actividad. Dichas salidas pueden ser usadas como criterio de éxito o fracaso de la actividad. Es decir, si la salida esperada se produce, se puede considerar que la actividad ha sido exitosa, en el caso contrario se afirmaría que la actividad ha fracasado.
- **Crear las pruebas unitarias:** por cada validación definida se crea una o varias pruebas unitarias que, con ayuda de los parámetros de configuración y los criterios de éxito o fracaso identificados previamente, accedan a la simulación para ejecutar las validaciones correspondientes. Las siguientes, son algunas consideraciones que se deben tener en cuenta al momento de crear las pruebas unitarias:
 - Se utiliza una única “*prueba principal*” para validar la salida esperada de una actividad.
 - En el caso de que sean necesarias “*pruebas secundarias*”, éstas deben organizarse en un orden concreto, de tal forma que sus resultados junto con el resultado de la prueba principal, puedan interpretarse rápidamente. Cuando sea posible, una prueba secundaria adicional podrá validar los resultados de todas las pruebas secundarias que acompañan la prueba principal.
 - Es importante considerar el uso de funciones *pre* y *post* prueba, especialmente en aquellos casos donde es importante realizar pruebas de dispositivo.
 - Las “*pruebas secundarias*” se usarán especialmente cuando se lleve a cabo la descomposición de las actividades físicas del usuario en movimientos o gestos simples. El desarrollador podrá emplear tantas pruebas secundarias como gestos simples deriven de la actividad física.
 - Debido al número considerable de pruebas que podrían generarse en este caso, a cada prueba secundaria se le asigna un orden secundario de ejecución *Order₂*, el cual irá asociado al orden en que se realizan los gestos simples para formar el movimiento complejo (ver la Figura 4.9).

Es recomendable que los ingenieros sean quienes asuman la responsabilidad de analizar el caso de estudio para darle cuerpo y forma a lo que será el caso de prueba. Del caso de estudio se extrae la información necesaria para indicar la visión general de cada caso, las acciones necesarias para ejecutar la prueba, una descripción de los datos de entrada, la configuración del ambiente de la prueba y finalmente los resultados obtenidos, diferenciando claramente las salidas obtenidas con respecto a las salidas esperadas. Por otra parte, los expertos en la asistencia juegan un papel fundamental al momento de representar las características de la solución asistiva en la simulación. Para ello, tanto ingenieros como expertos, utilizan la especificación técnica del despliegue tecnológico virtual hecho en el entorno simulado.

Según la característica que se quiera modelar, entre ambos determinan qué aspecto técnico de la solución se ve influenciado por dicha característica y a partir de ahí, establecen la forma de potenciarlo positivamente. Si se quisiese evitar la invasividad tecnológica de la solución propuesta, ambos podrían plantearse varias alternativas tecnológicas en el escenario que ayudasen a evitar o minimizar este problema.

4.3.2.1 Resultado del caso de prueba

El resultado del caso de pruebas consiste en un resumen de ejecución que muestra las salidas de todas las pruebas unitarias que lo conforman. Este informe muestra el orden de la actividad que se está validando, la descripción asignada, la prueba unitaria que se ha ejecutado, el elemento de la simulación sobre el que ejecuta la prueba, la hora de inicio y de fin en que se ejecutó la actividad, la descripción de la salida esperada en la actividad, el descriptor de la regla que se ha ejecutado cuyo nombre coincide con la acción o actividad identificada, un semáforo de colores donde “rojo” indica que hay un caso de fracaso en la prueba, “verde” cuando el caso es de acierto y “precaución” cuando se requiere que el experto observe la salida obtenida. Finalmente, se muestra la marca de tiempo del momento en que se detecta el patrón asociado al descriptor.

4.4 Resumen

El uso de la simulación en el ciclo de desarrollo de las soluciones asistivas ha hecho necesario establecer un mecanismo que permita trasladar las especificaciones de requisitos de la solución a la simulación. Esto a su vez ha supuesto nuevos retos relacionados con la forma de validar que dicha simulación realmente representa la solución asistiva y que ésta incluye las funcionalidades y las características que han sido recogidas a lo largo del proceso de ingeniería de requisitos.

Para suplir estas nuevas necesidades, en este capítulo se propone una formulación informal de pruebas unitarias sobre los distintos elementos que hacen parte de la simulación, con el fin de validarlos como parte de la especificación total o parcial de la solución asistiva. Las pruebas unitarias se consolidan en casos de prueba que se asocian directamente al caso de estudio que ha definido la solución. El escenario que se ha propuesto para llevar a cabo dichas validaciones incluye la construcción de un entorno de pruebas para la simulación, cuyo objetivo es validar si los requisitos establecidos en el caso de estudio han sido trasladados correctamente a la simulación. De esta forma se asegura que lo simulado puede ser utilizado como especificación final que contribuya en la implementación de la solución en un entorno real. Las validaciones en este caso se aplican directamente sobre el entorno simulado, el cual dispone del formato adecuado que permite comprobar el comportamiento de todos los elementos que hacen parte del entorno.

Capítulo 5

Requisitos de diseño

En el Capítulo 4 se definió el entorno de pruebas que se requiere para validar la simulación, así como los distintos elementos que hacen parte de él. En este Capítulo se describen los requisitos de diseño del framework que soporta la creación de dicho entorno, en el que además se llevarán a cabo las pruebas unitarias asociadas a los casos de prueba correspondientes. En la Sección 5.1 se describe de forma general el entorno de pruebas necesario para validar la simulación como parte de una solución asistiva.

En base a las características del entorno y a sus necesidades técnicas, en la Sección 5.2 se muestran los requisitos de diseño que se tuvieron en cuenta para diseñar el framework. Asimismo, se describen otros elementos importantes para su funcionamiento, tales como la interacción con los elementos de la simulación, el modelo de eventos empleado como base del diseño, las técnicas de procesamiento aplicadas y los modelos de reconocimiento de actividad que complementan la funcionalidad del framework para abordar la validación de actividades.

5.1 Entorno de pruebas

Una solución asistiva debe ser capaz de capturar datos, procesarlos y, con los resultados obtenidos, analizar las acciones del entorno para mediar o facilitar la interacción del usuario con la propia solución. Una forma de llevar a cabo la captura de datos es a través de *sensores*, los cuales detectan y recogen las condiciones del entorno y del usuario, así como las actividades que este realiza. Las interacciones del usuario con el entorno también se pueden obtener por medio de *actuadores*, los cuales permiten reconocer las respuestas a determinadas acciones ocurridas. En una simulación 3D que representa una solución asistiva, se construyen escenarios que cumplen estos mismos requisitos técnicos. Como se describe en la Sección 4.1.2, la simulación está dotada de dispositivos que reconocen el entorno simulado e identifican la actividad del usuario a través de las técnicas que se hayan propuesto para tal fin.

Cuando dichos dispositivos y técnicas se modelan en la simulación, se trasladan al escenario simulado muchos de los problemas que estos afrontan en el mundo real. Por un lado, la solución asistiva necesita de muchos sensores desplegados en el entorno, los cuales tendrán que ser coordinados tal y como ocurre en una solución real. Por otra parte,

tanto sensores como actuadores están activos y funcionando todo el tiempo que dura la simulación, por lo que es necesario procesar e interpretar todos los datos que éstos generen para obtener información relevante del contexto.

La Figura 5.1 muestra los principales elementos que interactúan y hacen parte de una solución asistiva. Inicialmente se parte de los requisitos que definen la solución, los cuales describen tanto las características de los usuarios como las del entorno. La participación de los expertos y profesionales médicos que han contribuido en la definición de la solución asistiva, es necesaria en todo el proceso de pruebas, especialmente cuando se evalúen aspectos relacionados con su campo de estudio. Finalmente, por medio de los sensores y actuadores se podrá determinar el comportamiento de la solución asistiva, empleado para ellos técnicas que permitan analizar la información generada.

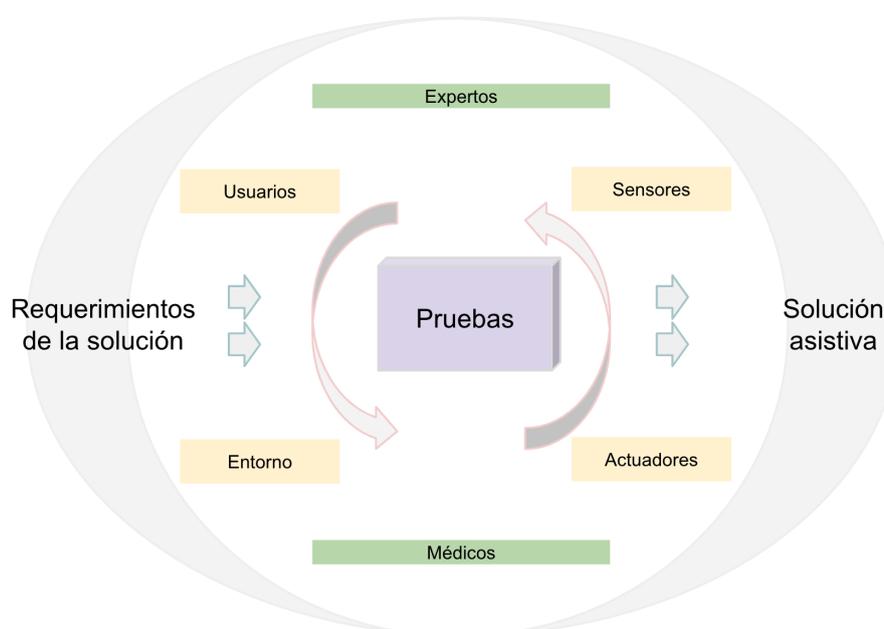


Figura 5.1: Principales componentes de una solución asistiva.

La interacción del usuario con el entorno es otro elemento importante de la simulación y, para identificarlo, el personaje también debe ser monitorizado de alguna forma para reconocer, por ejemplo, su ubicación, su estado dentro del escenario y las actividades o acciones que realiza en todo momento. Este último requerimiento describe a grandes rasgos la necesidad de monitorizar no solo al usuario sino a cada elemento de la simulación. Al momento de definir el caso de estudio, este aspecto se debe tener en cuenta y será el ingeniero quien se responsabilice de ello. Los ingenieros son quienes establecen el mejor mecanismo de seguimiento, bien sea mediante los sensores o actuadores que incluye la solución asistiva, a través de aplicaciones específicas ad hoc emuladas en el entorno con un fin determinado o con la ayuda de sensores utilizados por el desarrollador para instrumentar la simulación y realizar sus propias comprobaciones.

El mecanismo de seguimiento que se emplee en la simulación, debe considerar el hecho de que se utilicen diferentes escenarios simulados como los descritos en la Sección 3.3.1. Varias versiones de la simulación obligan al ingeniero a pensar en pruebas que garanticen

el funcionamiento de la solución en cada nueva versión de los escenarios. Es decir, que los aspectos validados que funcionan correctamente continúen así a lo largo de las distintas simulaciones que se puedan generar. Ante estas consideraciones en las pruebas de la simulación, el objetivo principal es construir un entorno de pruebas que, utilizando la simulación como especificación, facilite la validación de los elementos que la componen. Con la participación de expertos y profesionales de la salud en este proceso, se crearán casos de prueba que validen no solo cada elemento de la simulación sino también la solución asistiva que ésta representa. Asimismo, el entorno de pruebas proporcionará las herramientas necesarias para que se configuren los componentes de la simulación, de forma que cumplan con las especificaciones del caso de estudio real. El framework presentado en este Capítulo ha sido publicado en los artículos [GSCBPC16, PBGSP17] orientados al uso de sensores en la simulación y [CBSP18] donde se presentó el framework de pruebas.

5.2 Diseño del framework

Las pruebas necesarias para validar la simulación, requieren de un sistema adicional que permita construir el entorno de pruebas. La arquitectura que se propone en este trabajo, busca definir este sistema de forma que sea posible para el desarrollador construir los casos de prueba. Los requisitos que se han tenido en cuenta para definir la arquitectura son los siguientes:

1. La simulación, al igual que una solución AAL real, son sistemas basados en sensores y actuadores que generan flujos de eventos a lo largo del tiempo.
2. Debido al alto volumen de eventos generados por los distintos dispositivos, se debe especificar uno o varios flujos de procesos predefinidos que faciliten el procesamiento de grandes volúmenes de eventos en tiempo real. Con el procesamiento de los eventos lo que se busca es poder interpretar y reconocer lo que ocurre en el entorno (sensores ambientales) y con el usuario (sensores inerciales) dentro de la simulación.
3. La interacción del usuario con el entorno es un aspecto que se debe identificar para que la solución asistiva cumpla con su objetivo. Para ello, se deben definir procesos especializados en analizar y describir las dependencias temporales entre los eventos [Lam78]. Estos procesos son importantes porque ayudan a determinar la forma de relacionar ciertas situaciones dentro de la simulación, por ejemplo: la relación entre los eventos de sensores inerciales (usuario) y los ambientales (entorno).
4. Las condiciones físicas de un entorno simulado pueden variar de un contexto a otro y el usuario las puede percibir de forma muy distinta. Esta razón conlleva a la necesidad de utilizar umbrales estándares y umbrales que puedan ser configurados por el desarrollador, por ejemplo: niveles de ruido aceptables, nivel de temperatura, etc. Los umbrales en cada una de estas condiciones se pueden adaptar en tiempo real, permitiendo volver a ejecutar las consultas, pero con los nuevos parámetros establecidos.

5. El entorno se compone de diferentes elementos que con su comportamiento, describen su estado (*una puerta está abierta o cerrada, la luz se enciende o se apaga, hay personas en una zona de la casa*). De igual forma, el usuario en la simulación puede realizar diferentes actividades o acciones que describen su comportamiento (*el personaje está hablando, camina o está levantando una mano*). Para reconocer el comportamiento tanto del usuario como del entorno, se requieren de mecanismos que monitoricen a ambos durante todo el tiempo que esté funcionando la simulación. En la Sección 3.4.2.1 se describen los dispositivos disponibles en la simulación para la monitorización.

Para construir los entornos de prueba, se propone una arquitectura de software basada en eventos *Event-Driven Architecture* (EDA), como paradigma arquitectónico [Luc02] que ayude en la definición de los casos de pruebas (requisito 1). El objetivo es priorizar el procesamiento de eventos y utilizarlo como el elemento central de la arquitectura y del caso de prueba [LF98]. Para poder gestionar el volumen de datos y trabajar con ellos en tiempo real mientras dure la simulación, se ha propuesto utilizar el procesamiento de eventos complejos CEP [Luc02] como el modelo base para la toma de decisiones basado en eventos (requisito 2). Los flujos de eventos generados por la simulación, pueden contener un gran volumen de eventos diferentes [CFS⁺14]. Todos ellos se deben transformar, filtrar, etiquetar, almacenar, agregar y evaluar para disparar las acciones que se estimen conveniente en la simulación [CHKR14]. El paradigma de procesamiento propuesto, permite hacer un análisis de los flujos de eventos complejos en tiempo real, tanto individual como agrupados. Por esa característica, ésta técnica se considera como una solución idónea para analizar el comportamiento de los diferentes elementos que hacen parte de la solución asistiva, las cuales se basan en los sensores y actuadores dentro de la simulación.

Debido a que las soluciones ambientales basadas en sensores emiten continuamente datos tanto del entorno como del usuario, son particularmente adecuadas para CEP. El procesamiento de eventos complejos sigue un principio que resulta muy útil en soluciones de tipo ambiental. Este principio considera que los eventos no son independientes entre sí, sino que están relacionados y tienden a estar fuertemente correlacionados tanto en el tiempo como en el espacio [BK09]. De la misma forma, los bloques de eventos en cada *ventana temporal* [KC14] son igualmente relevantes para poder analizar el flujo en su totalidad (requisito 3). El interés no es procesar lecturas individuales en el tiempo o de distintos dispositivos dentro del ambiente, sino adecuar servicios concretos temporales y espaciales [JAF⁺06] que se enfoquen en el usuario.

Una simulación puede implementar diferentes tipos de sensores y por cada uno de ellos generar un flujo de eventos que lleva asociado toda la información propia de cada sensor. Cuando se construye el caso de prueba, muchas veces es necesario filtrar los flujos de eventos para poder procesarlos, bien sea por sus características y atributos, por el tipo de sensor del que proviene o por las condiciones del entorno que obedezcan a preferencias del usuario (requisito 4).

En lo que respecta al seguimiento del usuario, una forma de llevar a cabo esta tarea es mediante la monitorización a largo plazo del usuario, identificando en él aspectos

relevantes como su posición o ubicación dentro de la vivienda o las actividades físicas que realiza en un momento dado [SCSE10]. La forma tradicional de conseguir esto es por medio del reconocimiento de actividad HAR, cuyo objetivo no es otro que el de identificar las actividades conforme ocurren, en función de los datos recopilados por los sensores que se usen [ACRV13]. Debido a los múltiples enfoques que se pueden aplicar con ésta técnica, en este trabajo se ha hecho uso de los sensores de aceleración de 3 ejes [MHS01] que se acoplan al cuerpo del usuario [YJSB09], los cuales son los más usados para reconocer gestos [JALT08], la posición del cuerpo [LM02], patrones específicos del movimiento [AS95] y actividades cotidianas como correr, caminar, sentarse, levantarse, caer, entre otros [MSSD06]. La monitorización a largo plazo se delega en técnicas de clasificación especializadas en el reconocimiento de la actividad humana [PGK⁺09], las cuales se han integrado en la arquitectura para aprovechar la *gestión de flujos de eventos*. De esta forma, es posible generar acciones sobre la simulación en función de las características de la actividad física y del comportamiento del usuario (requisito 5).

5.2.1 Elementos del entorno en la simulación

Como se observa en la Figura 5.2, al momento de crear la simulación se establecen todos los elementos que intervendrán en ella: *el escenario, los usuarios, los sensores y actuadores, las actividades a realizar, entre otros*. Cada uno de estos elementos, como se ha descrito en la Sección 4.2, lleva implícito un conjunto de características que podrán ser validadas individualmente en el caso de prueba. Para que la validación sea posible, estos elementos cuentan con un conjunto de parámetros que deben ser configurados para que su comportamiento sea muy similar al del entorno real.

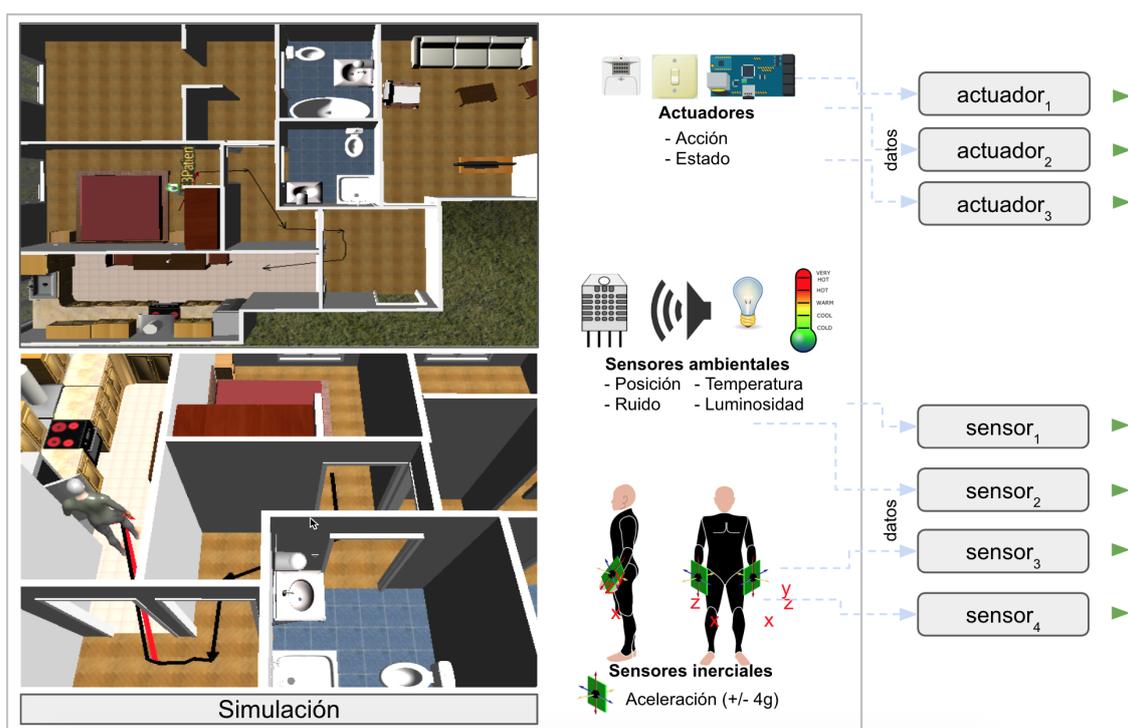


Figura 5.2: Configuración de la simulación para el uso de sensores.

En la simulación, es posible definir los diferentes componentes y posicionarlos en cualquier parte del entorno. Así como se definen y posicionan los elementos decorativos, de la misma forma se crean y se ubican los sensores que se usarán bien como parte de la solución asistiva o para instrumentar la simulación y medir determinadas características en ella. Aunque técnicamente los sensores de la solución y los utilizados para instrumentar la simulación son iguales, su principal diferencia radica en el uso que se les da. Los sensores de la solución forman parte de ella y contribuyen con un propósito en el entorno, mientras que los sensores utilizados para instrumentar, son útiles para analizar características concretas de la simulación sin intervenir o causar algún efecto en la solución asistiva.

Para el uso de la arquitectura propuesta, solo se tienen en cuenta la especificación de la simulación y el caso de estudio que define la solución asistiva, por lo que creación de la simulación no será un aspecto que se detalle en esta tesis. En el trabajo de investigación de Campillo-Sánchez [CS⁺17], se puede encontrar más información sobre el funcionamiento de la herramienta *SociAAL*, al igual que una descripción técnica de cómo se crean las simulaciones y la forma en que se definen las actividades sobre el usuario.

5.2.1.1 Acceso a datos

Después de ubicar los sensores en el escenario, éstos disponen de sus propios mecanismos de publicación de datos y de parámetros de configuración que podrán modificarse desde la simulación antes de ponerlos en marcha, por ejemplo: *posición*, *unidad de medida*, *frecuencia de muestreo* y *estado*. Como se observa en la Figura 5.3, el conjunto de sensores y actuadores sobre el escenario, son los elementos que brindan toda la información de lo que ocurre en la simulación y es por ello que su configuración y parametrización, es importante a la hora de ser puestos en escena.

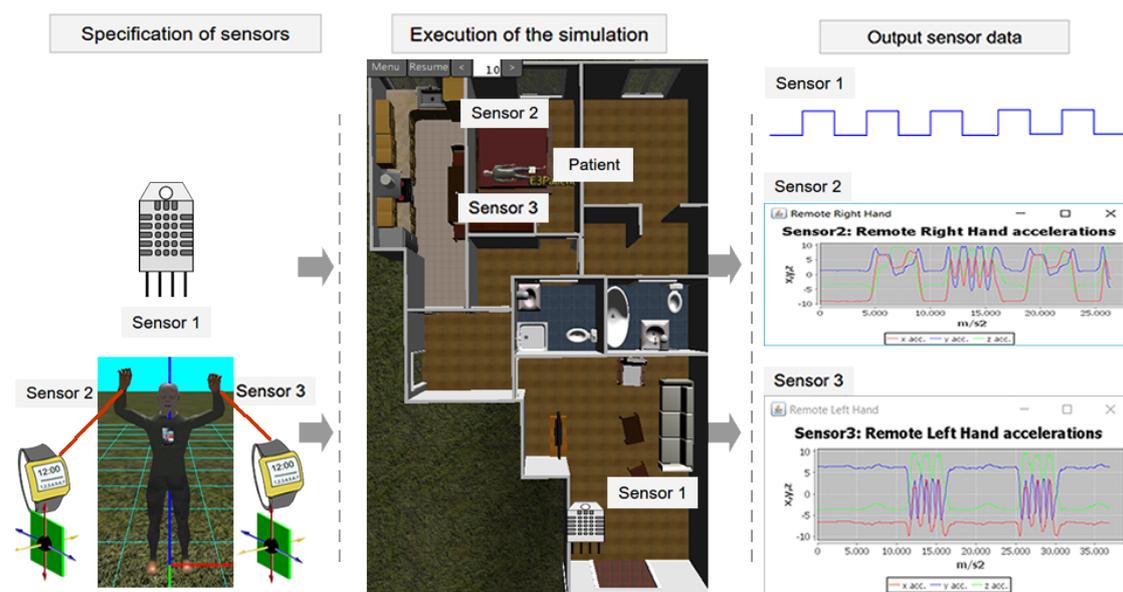


Figura 5.3: Distribución de los sensores en el entorno simulado.

Cuando se crea la simulación, los sensores de la solución ambiental se posicionan en los puntos que interesan para el problema estudiado. En base a la configuración

de cada sensor, de ellos se obtiene la información tanto del entorno (*ambiental*) como del usuario (*comportamiento*). Cuando la simulación se pone en marcha, los sensores inmediatamente se activan y es posible acceder a ellos teniendo en cuenta los atributos que estos tienen definidos, como se observa en la Tabla 5.1. Los sensores que se pueden utilizar en la simulación son: *temperatura*, *posición*, *detector de presencia*, *inercial (aceleración)* y *sonido*. Para acceder a ellos, una vez situados en el escenario, la simulación implementa *websocket* por su eficiencia [PN12] y ofrece sus datos mediante streaming [GGR16].

Atributos	Valor de ejemplo	Descripción
id	"11b4de33"	Identificador único del sensor
category	"sensor"	Categoría del dispositivo: "sensor", "actuador"
messageFormat	[*]	Formato del mensaje que describe la lectura: unidad y nombre.
type	"Temperature"	Tipo de sensor
control	[*]	Elementos que tienen control de sensor asociado en la simulación.
contextTopic	"sensor\11b4de33"	Breve descripción del sensor en el contexto: ambiental, actividad, entre otros.
serviceTopic	"ambient\11b4de33"	Breve descripción del sensor como servicio de la solución.
URI	ip:port:topic	Uniform resource identifier
location	"room"	Localización del sensor en el entorno.

Tabla 5.1: Estructura básica de los datos del sensor.

Una vez el framework dispone de los datos, el flujo de eventos se formaliza a través de tuplas, las cuales se componen de un identificador del evento y por el evento mismo (ver listado 5.1). El tipo de evento se infiere a partir de la estructura de los datos del propio sensor, la cual se formaliza una vez el evento pasa por la primera etapa de procesamiento.

```

{
  /**
   * Formalization of event flows using tuples.
   */
  Stream<SimulationEvent> input = streamSource
    .flatMap(new SimulationFlatMap.SimpleFlatMap())
    .assignTimestamps(new AssignerWithPunctuated<SimulationEvent>() {
      @Nullable
      @Override
      public Watermark checkAndGetNextEvent(SimulationEvent simulationEvent, long l) {
        // Checking for new events and processing timestamps.
        return new Event(simulationEvent.getTimestamp().longValue() - Sim.delayed);
      }
    })
    .flatMap(new SimulationFlatMap.SimpleFlatMap() {
      @Override
      public long extractTimestamp(SimulationEvent simulationEvent, long l) {
        // Recovery of capture or processing times.
        return simulationEvent.getTimestamp().longValue();
      }
    })
    .map((MapFunction<SimulationEvent, SimulationEvent>) simulationEvent -> simulationEvent);
}

```

Listing 5.1: Formalización de tuplas en los datos de los flujos de eventos.

5.2.2 Modelo de eventos

En la arquitectura se propone un modelo que procesa los flujos de eventos mediante el paradigma de procesamiento de eventos complejos CEP. Este se basa principalmente en que los eventos independientemente de su fuente, están relacionados entre sí [CFS⁺14]. Este mismo principio se aplica en los entornos monitorizados mediante sensores. En ellos, las

lecturas que se obtienen en un determinado instante serán útiles para analizar las lecturas de los instantes siguientes. En la simulación ocurre exactamente igual, este principio se aplica de la misma manera que ocurre en un entorno real. Los sensores no operan de forma individual, en su conjunto brindan información que es relevante para responder a un servicio adecuado para el usuario. Para encontrar la relación y correlación entre toda esta información, el modelo se apoya en CEP para identificar la coincidencia de patrones entre los eventos que sean significativos para el caso de prueba que se quiere validar. Si el caso en cuestión se centra por ejemplo, en la posición del usuario dentro de la simulación, los patrones en los eventos tienen que ir en función de aquellos sensores que permitan relacionar al usuario con su posición actual: *luminosidad* (la luz encendida en la habitación donde está el usuario), *actuador* (indica que hay un electrodoméstico en uso), *sensor de presencia*, etc.

Por otra parte, el procesamiento de todos los flujos de eventos que se generan, requiere de un formalismo que unifique o estandarice la información de los sensores. Esta formalización es necesaria porque los eventos provienen de fuentes distintas que pueden contar con una configuración diferente entre sí. El modelo de eventos permite usar *metadatos* sobre la información del sensor, los cuales brindan una mayor descripción de la relación “*dato-sensor-entorno*” y las posibles dependencias con la red de sensores. Los *metadatos* también se utilizan para enriquecer la información a procesar, añadiendo atributos como un identificador del dispositivo, su ubicación, la tipología del sensor, la frecuencia de muestreo empleada, el posicionamiento con respecto a otros sensores, la segmentación de los eventos y las marcas de tiempo. El propósito de los atributos es ofrecer más información que permita relacionar la lectura con las características y condiciones del sensor en el momento de hacer dicha lectura. A continuación se muestran la primera definición formal que se usará para describir el concepto de *evento*.

Definición 5.2.1 *Un evento de tipo E se puede definir como $E = (id, a, t_i)$, donde cada tipo de evento está identificado de forma única con un id ; cuenta además con una serie de atributos $a = \{attr_1, attr_2, \dots, attr_n\}$, donde $n \geq 0$, que lo caracterizan y que describen tanto la zona donde está ubicado el sensor como la actividad que este representa, así como un tiempo t_i en que se produjo el evento.*

Siguiendo la Definición 5.2.1, el uso de atributos en los eventos permite que se pueda emplear un lenguaje de procesamiento de eventos como *Event Processing Language* (EPL) [dCAT08]. EPL es un lenguaje estándar basado en el álgebra de eventos [WC96], el cual utiliza cláusulas SQL como si se tratase de una base de datos relacional. Esta característica permite al desarrollador utilizar consultas SQL o reglas [PK09] convencionales sobre un modelo de datos no relacional (*select, from, where, group by, having y orden by*), en el cual persisten todos los eventos extraídos de los distintos flujos generados por cada sensor en la simulación. En este punto, se hace necesario añadir una definición más que permita describir formalmente el concepto de flujo de eventos.

Definición 5.2.2 *Un flujo de eventos se puede definir como $f_i = (e_1, e_2, \dots, e_i)$, donde cada elemento e_i es una instancia de un evento de tipo E que proviene de un mismo sensor de la simulación.*

En aquellas pruebas que impliquen el uso de eventos de diferentes fuentes, el modelo propuesto lo que hará es generar nuevos flujos compuestos por eventos que provienen de distintos dispositivos, con el propósito de analizar características en el comportamiento del entorno que no pueden deducirse con información de un mismo tipo de sensor. La Definición 5.2.3 es la utilizada en este caso para formalizar estos flujos de eventos.

Definition 5.2.3 *Un flujo de diferentes tipos de eventos E_1 y E_2 , se define como $f_{s_i} = (e_1^{s_1^1}, e_1^{s_1^2}, e_2^{s_2^1}, e_2^{s_2^2}, \dots, e_n^{s_i^i}, e_n^{s_i^j})$, donde cada elemento e^{s_i} es una instancia de un evento de tipo E_i que proviene de un sensor concreto s_i en la simulación.*

Los sensores en la simulación generan los flujos de lecturas una vez están en funcionamiento, pero es el modelo quien se responsabiliza de formalizar la información en eventos, añadiendo los distintos atributos descritos en las definiciones 5.2.2 y 5.2.3). Una vez formalizados, estos se comportan como fuentes de datos constantes que son representados como tablas de datos relacionales. Con base a esto, el modelo propuesto mediante EPL permitirá en las pruebas utilizar agregaciones [MSHR02], combinaciones [ABW03] o subconsultas [ABB⁺16] a través de la agrupación de eventos.

5.2.2.1 Reglas

Una regla o consulta, es un patrón específico que identifica variaciones en las lecturas de los sensores. En el framework, los patrones se identifican con etiquetas al igual que las animaciones, y algunos de ellos están predefinidos en la herramienta. Los patrones utilizados, describen dos tipos de variaciones en la lectura: cambio (“change”, “gyro”) y tendencia (“up”, “down”, “left”, “right”). También se han escrito funciones (reglas) que calculan valores estadísticos como *sum*, *max*, *min*, *avg*, *var*, *mean*, aplicables a lecturas de sensores inerciales. En el contexto de CEP, la definición de regla es más formal y exige una sintaxis genérica que permita estandarizar su estructura dentro del framework.

Definition 5.2.4 *A CEP ruler has the following syntax:*

$et = a* : e_1 \dots e_n \Leftrightarrow \text{predicate}$; *predicate = condition to be met to generate the event along the time windows*; $et \equiv \text{event to be generated}$
 $e_1, \dots, e_n \equiv \text{event from a previous CEP run}$.

Siguiendo la Definición 5.2.4, una regla debe tener un descriptor único que la diferencie y debe cumplir con la especificación antes descrita y conservar la estructura de parámetros que se establezca al momento de crearla. Cada regla se almacena por separado, luego el framework se encarga de unir las en función de las pruebas unitarias establecidas por el desarrollador y de las especificaciones que se incluyan en el caso de prueba en general. Después de ejecutar las reglas que componen el caso de prueba, el resultado final es un listado que muestra las distintas salidas obtenidas que describen los patrones encontrados, los cuales se usarán en la prueba unitaria para determinar si se produce un acierto o un error según lo especificado en dicha prueba.

5.2.2.2 Reglas simples

Una regla simple identifica un patrón sobre eventos que poseen un único valor en su lectura, sin necesidad de emplear reglas anidadas o subconsultas para encontrar dicho patrón. Es así como el comportamiento simple de los datos se describe mediante reglas simples que muestran cómo varían los datos a lo largo del tiempo. Los sensores ambientales son los que mejor se adaptan al uso de reglas simples para monitorizar su comportamiento.

5.2.2.3 Reglas complejas

Una regla compleja identifica un patrón sobre eventos que poseen más de un valor en su lectura, como ocurre con los sensores inerciales. Este tipo de reglas, para encontrar dicho patrón, recurre a reglas anidadas o subconsultas que permiten realizar cálculos adicionales necesarios para analizar el comportamiento de los eventos. Una regla compleja también se usa para procesar múltiples fuentes de datos o distintas reglas simples que procesen eventos simultáneamente. En el caso de los sensores inerciales, una regla compleja actuaría sobre la lectura de aceleración (tres ejes). Una regla simple por su parte, sólo podría actuar sobre las lecturas de uno de los ejes x , y o z del sensor. En la Tabla 5.2 se pueden observar algunos ejemplos de reglas simples y reglas complejas, las cuales describen su tipo en función del cálculo que realizan.

Descriptor	Regla	Descripción
@up	regla simple	sube la medida del sensor
@down	regla simple	baja la medida del sensor
@left	regla compleja	se incrementa la medida del sensor en los ejes (x,y)
@right	regla compleja	se decrementa la medida del sensor en los ejes (x,y)
@drink, @walk, @clap, @run, @sitdown @getup, @waveattention	regla compleja actividades	Actividades que soporta el sistema HAR.
- @raisehandvf - @lowerhandvf - @raisehandvs - @lowerhandvs	regla compleja	- levantar la mano (de frente en vertical) - baja la mano (de frente vertical) - levanta la mano (de lado vertical) - baja la mano (de lado vertical)

Tabla 5.2: Descriptores para identificar reglas y actividades predefinidas.

5.2.2.4 Jerarquía de eventos

El uso de distintos sensores hace necesario operar con diferentes tipos de eventos. En el entorno existirán sensores muy distintos que generarán flujos de eventos característicos (ver Definición 5.2.2 y 5.2.3), que tendrán que ser procesados simultáneamente para encontrar información coherente que describa el contexto. En un escenario como este, los eventos

deben seguir alguna jerarquía que permita organizar y estructurar los distintos tipos de eventos que son significativos para el dominio de las pruebas [DFOO11]. Como se observa en la Figura 5.4, todos los eventos se pueden organizar de forma jerárquica, siguiendo un sistema de capas que describe el orden en el que irán siendo procesados conforme entran al motor de reglas.

Esta jerarquía refleja la secuencia de pasos que son necesarios para el procesamiento de eventos, transformando los eventos de sensores sin procesar en eventos de dominio sofisticados y más abstractos [Dun09]. A continuación se describen los tipos de eventos genéricos que se esperan en el modelo propuesto.

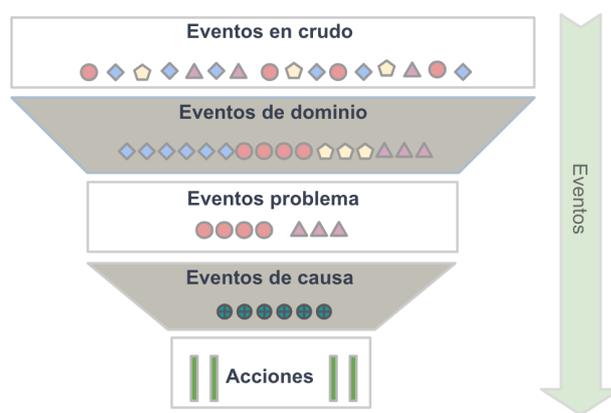


Figura 5.4: Organización jerárquica de eventos.

- Eventos en crudo:** son los eventos generados por los sensores sin ningún tipo de tratamiento. Este tipo de eventos por lo general, contienen muchas anomalías que debe ser revisadas, por ejemplo: ruido o valores atípicos en las lecturas, pérdida de lecturas o duplicidad de la misma [WP16], entre otras. Como sugiere [JAF⁺06] en su obra, los eventos crudos se deben procesar, aplicando técnicas que permitan corregir el mayor número de anomalías posible.
- Eventos de dominio:** después de limpiar y organizar los eventos, se aplican otras técnicas que permitan relacionar los eventos entre sí para dar más claridad al flujo de información [BCK⁺07]. Definir un dominio para los flujos de eventos, permite identificar la naturaleza del evento y a todos aquellos eventos que puedan estar relacionados con él. Si por ejemplo, se han posicionado sensores para monitorizar una habitación, estos podrán tener relación con los sensores inerciales que estén monitorizando la actividad del usuario durante el tiempo que esté dentro de dicha habitación.
- Eventos de problema:** son todos aquellos eventos que pertenecen a un dominio y que de alguna forma provocan una inconsistencia en el funcionamiento general del sistema. De igual forma, pueden ser parte de una situación específica anómala que se quiere identificar dentro de la simulación.
- Eventos de causas:** la forma de dar una explicación a un evento problema es mediante eventos que representen la causa de dicho problema. En este caso,

se generan nuevos eventos que representan las posibles razones por las que se originó el problema. El objetivo de estos eventos es ejecutar acciones posteriores que modifiquen alguna característica de la prueba unitaria. Un ejemplo práctico sería que ante una acción como *action* = “*user turns on the light*”, se pueden determinar dos o más acciones que sean la causa de que dicha acción ocurriese $cause_1 = \text{“turn on the light”}$ y $cause_2 = \text{“user moves the hand”}$.

- **Eventos de acción:** son aquellos eventos que se generan después de analizar los eventos de causa. Estos podrán ser eventos puntuales o flujos de eventos que se utilizarán para corregir la causa del problema o reparar los daños que haya generado. En el caso de las pruebas unitarias, se podrán utilizar para determinar si, por ejemplo, el entorno debería actuar de una forma ante un evento que explique la causa de un problema y este responda de otra forma.

Tipo de evento	Ejemplos
Acción	a) - Notificación es enviada al cuidador
Causa	a) - el usuario camina entre las 02:00 a.m. y 04:00 a.m. b) - el usuario mueve sus manos c) - la luz se enciende d) - el actuador de la luz está activado
Problema	a) - el usuario va al baño a las t horas b) - el usuario va al baño a las t+1 horas c) - la hora actual está entre las 02:00 a.m. y 04:00 a.m. d) - $t + 1 - t \leq 10$ minutos
Dominio	a) - vivienda, actividad física b) - location=“habitació” y type=“inertial”

Tabla 5.3: Ejemplo de cada tipo de evento de la jerarquía.

5.2.3 Procesamiento en capas

Para procesar los eventos, la literatura recomienda un procesamiento de los eventos siguiendo una arquitectura de capas como la que propone CEP. Como se observa en la Figura 5.5, CEP utiliza la jerarquía de eventos [LF98] descrita en la sección anterior y la abstrae mediante el uso de capas [BDB⁺14] que procesan los eventos de forma más inmediata y sin pasar por tantos estados. Como requisito de diseño, se ha propuesto seguir este planteamiento donde la responsabilidad de cada capa es la siguiente:

- **Primera Capa:** recibe y procesa los eventos crudos o reales que provienen directamente de la simulación.
- **Segunda Capa:** toma los eventos de salida de la primera capa y genera nuevos eventos virtuales o eventos de dominio a partir de un proceso de filtrado de flujos.
- **Tercera Capa:** aplica las reglas para detectar patrones específicos, que permitan identificar eventos de problemas que interesen en los flujos. Luego a partir de ellos,

genera eventos intermedios o de causa que describan el problema para finalmente lanzar los eventos de acción que ayuden a describir el comportamiento de los componentes de la simulación.

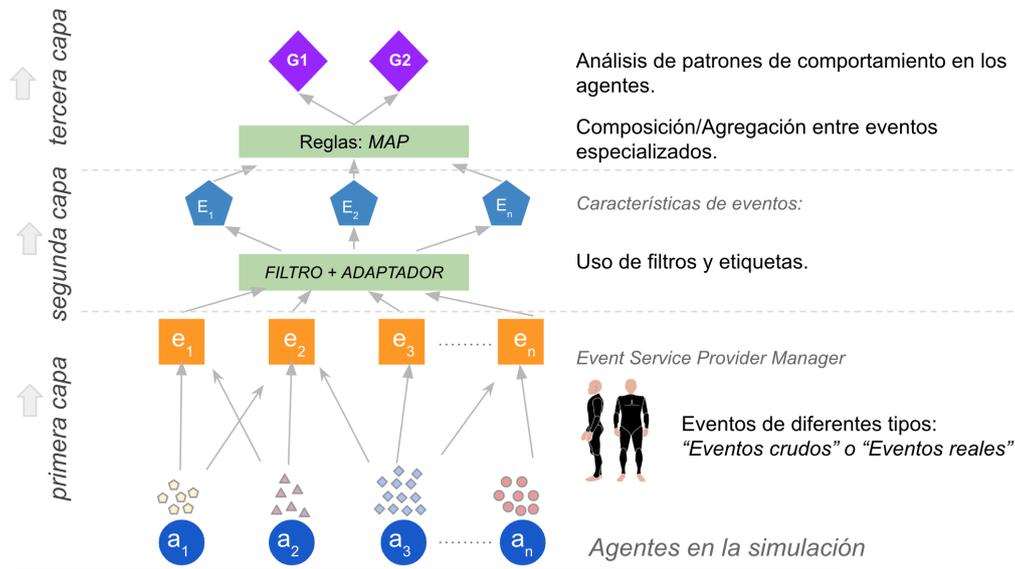


Figura 5.5: Jerarquías de eventos.

5.2.3.1 Primera Capa

Los datos que se obtienen de la simulación son los denominados eventos en crudo, simples o reales [Mic06]. Cada sensor dentro de la simulación genera un evento con la información que él mismo mide del entorno en tiempo de ejecución. En esta capa, los datos del sensor se homogeneizan teniendo en cuenta la Definición 5.2.1 y asumiendo las siguientes premisas:

- Uno o más agentes *agents* $\{a_1, a_2, a_3, \dots, a_n\}$ pueden ser introducidos en la simulación en un momento determinado.
- Si el personaje o el entorno se están monitorizando con n sensores, cada sensor debe ser identificado de forma única $\{s_1, s_2, s_3, \dots, s_n\}$.
- Los agentes se mueven dentro de la simulación. Cada vez que el agente cambia de posición, la simulación indica su nueva posición creando un nuevo evento.
- Todos los agentes generan flujos de eventos en forma constante, mientras la simulación está en ejecución (ver Definición 5.2.2).

La entrada de esta capa son los eventos de la simulación que cumple la Definición 5.2.1 y se pueden describir de la forma $e_i = (id_a, s_a^t, t_i, w_i)$, donde cada evento e_i cuenta con un identificador único id_a y un conjunto de datos del sensor s en el instante de tiempo t_i y dentro de la ventana temporal w_i .

La **salida** es un nuevo evento de la forma $e_{1L} = (id_a, s_a^t, status_a^t, t_{i+1})$, donde cada nuevo evento e_{1L} conserva su identificación unívoca id_a , contiene una serie de atributos s_a^t que le caracterizan (“*position*”, “*type*”, “*frecuency*”, “*size*”, “*coordinate*”), un estado $status_a^t$ y una marca de tiempo t_{i+1} de proceso. En el caso de los agentes, s_a contiene el atributo *coordinate* que indica un punto p_a de la simulación, donde p_a se representa como un vector $\vec{p}_a = (x, y, z)$ donde $x, y, z \in \mathbb{R}^3$.

5.2.3.2 Segunda Capa

Esta capa parte de los eventos e_{1L} generados en la capa anterior, y son utilizados como **entrada** en esta parte del proceso. Sobre éstos, se aplican los filtros necesarios para generar nuevos eventos de tipo e_{2L} , que permitan diferenciar los sensores por su etiquetado o por el conjunto de características que los describan lo mejor posible.

La **salida** de esta capa serán nuevos eventos de la forma $e_{2L} = (id_a, s_a, t_i, g_a)$, donde g_a es la información tanto de etiquetado añadida al nuevo evento como de dominio, necesaria para contextualizar el evento dentro de la simulación (área del sensor, ubicación, usuario más próximo y actividad en ejecución). Estos atributos se definen tal y como se muestra en $g_a = \{w_t^{default}, w_t^{test}, w_d, c_a, context\}$, donde $w_t^{default}$ es la ventana de tiempo por defecto, w_t^{test} la ventana de tiempo especificada en el caso de prueba, w_d es la ventana de datos, c_a es un indicador textual utilizado para asociar el flujo a los nodos clasificadores y finalmente información sobre el contexto *context*, cuyos atributos son *area*, *ubication*, *user* y *activity*.

5.2.3.3 Tercera Capa

Partiendo de los eventos e_{2L} de la capa anterior y según los criterios de éxito que se definan en el caso de prueba, esta capa aplica las reglas que detectan patrones específicos que permitan identificar “eventos de problemas” que interesen en los flujos. A partir de ellos, se generan eventos intermedios o “eventos de causa” que describen el problema y finalmente arrojan “eventos de acción” que determinan realmente el estado y el comportamiento de los componentes en la simulación.

La **salida** de esta capa es un evento con la forma $e_{3L} = (w_d, t_i, g_a, idRule)$, donde cada nuevo evento e_{3L} se representa en una ventana de datos w_d^{test} , g_a mantiene la información de etiquetado y *idRule* contiene el identificador de regla utilizada para la predicción, la cual es sólo útil cuando el evento proviene de la actividad física de los agentes.

5.2.4 Modelos de reconocimiento de actividad

Como se menciona en la Sección 5.2 (requisito 4), en determinados casos la motorización del ambiente requiere no solo comprobar el estado del entorno, sino que además es importante verificar o identificar qué actividades puede estar realizando el personaje dentro de la simulación.

Una actividad física del personaje puede ser la respuesta, o parte de ella, que espera el entorno por parte del usuario. Es por ello que identificar si dicha actividad física se ha llevado a cabo o no, será primordial en los casos de prueba que verifiquen la interacción

del usuario con el sistema. Como se muestra en la Figura 5.6, el framework hace uso de las **animaciones** del personaje como representación de esas actividades físicas. Para que sea posible la monitorización, el personaje en la simulación debe portar sensores inerciales que permitan capturar la *aceleración* de sus movimientos.



Figura 5.6: Ejemplo de animaciones en un personaje.

El sensor debe ser configurado y luego ubicado en las partes del cuerpo que se consideren relevantes para analizar el movimiento físico. La configuración del sensor y su posición sobre el personaje es información relevante tanto para el caso de estudio como para la construcción de los clasificadores de actividad.

Los sistemas clasificadores son componentes del modelo propuesto, encargados de procesar los flujos de eventos que provienen de sensores inerciales, los cuales como se ha mencionado anteriormente, son utilizados en la simulación para monitorizar la actividad física del personaje. Estos flujos previamente atraviesan toda la estructura de capas propuesta en el apartado 5.2.3. En la *tercera capa* de la jerarquía, se aplican reglas que separan los flujos de eventos e_{3L} que corresponden a sensores inerciales para que sean enviados directamente a los sistemas de reconocimiento de actividad. Este sistema lleva a cabo su propio procesamiento de eventos con el que extrae las características predefinidas que usará con los modelos de clasificación correspondiente. El resultado del modelo, será la predicción que determine la actividad física que está realizando el usuario en la simulación.

5.3 Resumen

El presente capítulo ha descrito los principales requisitos de diseño que se han tenido en cuenta para construir el framework de pruebas propuesto. El propósito de dicho framework es facilitar la creación del entorno de prueba donde se validará la simulación, teniendo en cuenta los casos de prueba y las pruebas unitarias correspondientes descritas en el Capítulo 4, que actuarán directamente sobre los escenarios simulados.

Dentro de las características más relevantes que se han abordado en el diseño, se encuentran el uso de sensores para monitorizar y describir el entorno y el comportamiento

del usuario, la necesidad de procesar los altos volúmenes de datos que se genera en este tipo de soluciones, la definición de interfaces que faciliten la interacción natural del usuario con el entorno, la creación de mecanismos que reconozcan las condiciones físicas de los elementos simulados (entorno y usuario) y por último, y no menos importante, el reconocimiento de las actividades o acciones físicas del usuario que se espera ocurran en la simulación para que la solución sepa responder a ellas.

Capítulo 6

Arquitectura del entorno de pruebas

En el Capítulo 3 se definió la metodología propuesta para construir las simulaciones a partir de los casos de estudio que representan soluciones asistivas. El fin en este caso es usar la simulación como especificación de dicha solución. En el Capítulo 4 se planteó la forma de definir un entorno donde es posible validar que la simulación se corresponde con dicha solución, empleando para ello los casos de prueba y las pruebas unitarias para validar aspectos concretos en la simulación. El framework que soporta la creación de este entorno se presenta en el Capítulo 5, enfatizando en los requisitos de diseño que se han tenido en cuenta para su construcción.

En este Capítulo se muestra la arquitectura desarrollada para construir los casos de prueba que abordarán la simulación para que estos sean ejecutados sobre ella. Asimismo, se muestran las diferentes funcionalidades que se aplican tanto para la creación de las pruebas unitarias como para la definición de casos de prueba más complejos. En la Sección 6.1, se describen los principales componentes de la arquitectura, entre ellos el motor de reglas que se encarga de procesar los flujos de eventos de la simulación y los clasificadores, los cuales intervienen cuando se requiere reconocer la actividad del personaje en el escenario. Finalmente, en la Sección 6.2 se muestra el filtrado basado en atributos de los eventos y en la Sección 6.3, el filtrado mediante el uso de umbrales e indicadores en la simulación.

6.1 Arquitectura propuesta

Teniendo en cuenta los requisitos de diseño descritos en la Sección 5.2, se propone la arquitectura que se muestra en la Figura 6.1. En ella se puede observar el diagrama de componentes que conforman el sistema. En particular, los dos elementos principales: La *Simulación* y el *Motor de pruebas (Tester)*.

Simulación es el componente que representa el modelado del problema y describe las especificaciones del caso de estudio que modela la solución asistiva. Los casos de prueba se aplican sobre la simulación por lo que para su construcción se tienen las características del formato 3D de ésta. La simulación en 3D se encarga de recrear, de la forma más exacta posible, el escenario donde se quiere implementar la solución asistiva, proporcionando

información del entorno mediante la red de sensores, los actuadores y el usuario.

Motor de pruebas es el componente que proporciona las herramientas necesarias para construir los casos de prueba que se aplicarán sobre la información que genera la simulación. Básicamente permite al desarrollador crear conjuntos de *pruebas unitarias* que podrá usar sobre cada componente de la simulación. El desarrollador establece las características que quiere validar y para ello, enfoca cada prueba unitaria al elemento de la simulación responsable de dichas características, como se ha descrito en la Sección 4.2.

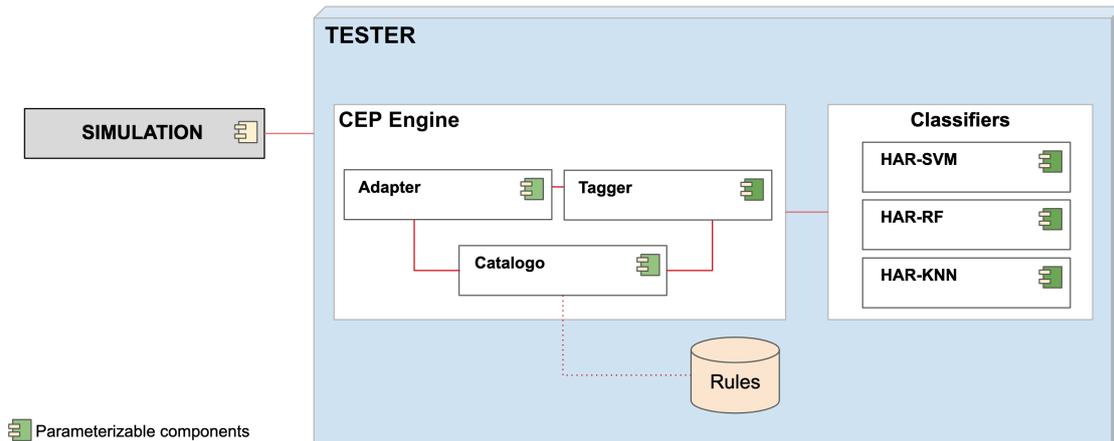


Figura 6.1: Arquitectura del sistema.

Para su funcionamiento, este componente se apoya en dos elementos claves: *Motor de reglas* y los *Clasificadores*.

- **Motor de reglas:** es el nodo responsable de ejecutar todas las validaciones del caso de prueba sobre los datos generados por la simulación, teniendo en cuenta los distintos parámetros de configuración que se hayan establecido en la solución asistiva y que estén definidos también en el mismo caso de prueba. Este nodo inicializa y lanza las instancias de procesamiento de eventos complejos que están detrás de cada prueba unitaria. La instancia se conecta al flujo de eventos correspondiente y ejecuta las reglas o consultas que determinarán si los criterios de éxito o fracaso de la prueba, se cumplen dentro de la ventana de tiempo establecida.
- **Clasificadores:** este componente agrupa servicios especializados en el reconocimiento de la actividad humana HAR. Estos servicios son ofrecidos por nodos clasificadores, los cuales han sido construidos utilizando técnicas de aprendizaje automático supervisado. El reconocimiento de la actividad es necesario cuando en la simulación se quiere identificar movimientos o gestos concretos de personajes humanos, los cuales hacen parte de animaciones complejas por ordenador. Este componente se describe con mayor detalle en el Capítulo 7.

Los detalles sobre las interfaces del framework y la funcionalidad de cada módulo se describen más detalle en los siguientes apartados.

6.1.1 Motor de pruebas

Este componente ofrece todos los elementos necesarios para que el desarrollador y los expertos diseñen, definan y construyan los casos de prueba que interesen a ambos. En la construcción de un caso de prueba se crean las pruebas unitarias que validan las características que se consideren relevantes en la simulación, utilizando para ello, toda la información que genera la red de sensores distribuida en el entorno simulado. Los componentes que conforman el *Motor de pruebas* se muestran en la Figura 6.2. La descripción de cada componente se indica a continuación.

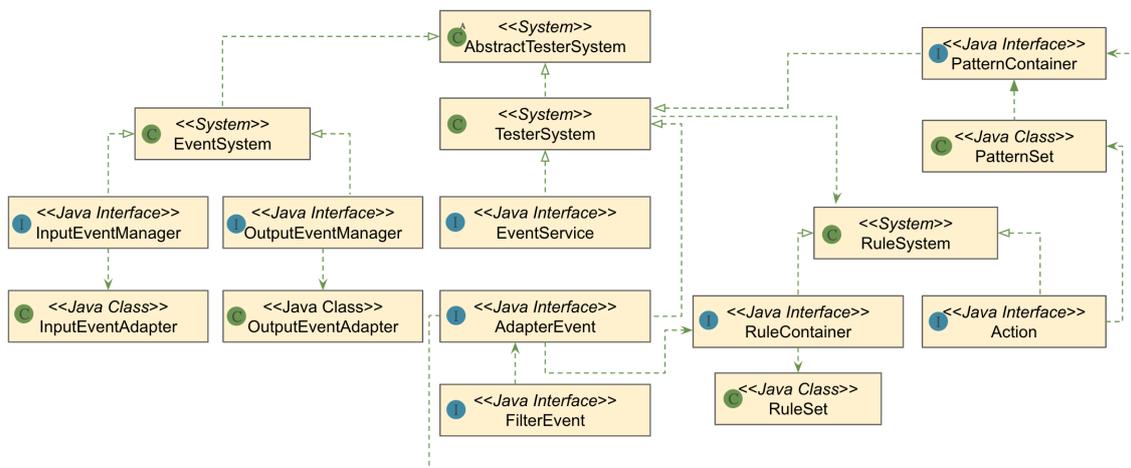


Figura 6.2: Diagrama de clases del sistema de pruebas.

- **AbstractTesterSystem:** define el comportamiento general de los principales elementos sistema en la arquitectura, los cuales se relacionan con el modelo de eventos propuesto y con la estructura de los casos de prueba.
- **EventSystem:** proporciona el acceso a los eventos de la simulación, diferenciando los eventos que provienen directamente de la solución de aquellos que utiliza el desarrollador para instrumentar la simulación.
- **InputEventManager:** implementa las funciones disponibles para operar sobre los eventos de entrada al sistema, tanto los que provienen de la simulación como de otras fuentes que se definan en el caso de prueba.
- **OutputEventManager:** implementa las funciones que conducen los eventos que han sido procesados y que deben pasar a otras etapas del proceso.
- **TesterSystem:** proporciona la estructura necesaria para construir las pruebas basándose en la biblioteca de JUnit [Bec18]. El desarrollador de los casos de prueba usará la filosofía que sigue JUnit para hacer pruebas unitarias en aplicaciones Java, por lo que podrá definir tantos casos de prueba como considere y agruparlos en suites para su organización.

- **AdapterSystem:** procesa todo flujo de eventos que entren al motor para que puedan ser accedidos por las pruebas unitarias que se definan en el caso de prueba.
- **RulerSystem:** recoge la estructura que deben seguir las reglas que se aplicarán sobre los flujos de eventos. Para ello, implementa en *RuleContainer* funciones que permiten al desarrollador definir reglas en la prueba unitaria y lanzar acciones con *Action*, como resultado de dichas reglas.
- **PatternContainer:** define la estructura de patrones predefinidos que se utilizan dentro del sistema para el tratamiento de datos, aplicar criterios de validación en los datos y en las actividades (ver Sección 4.2). De igual forma, crea eventos que etiqueta y combina con otras reglas en *RulerSystem*, añadiendo nuevos atributos de información, que son útiles a la hora de implementar los criterios de éxito establecidos en el caso de prueba.

6.1.2 Motor de reglas

Como se ha indicado previamente, este es el nodo de proceso principal de la arquitectura. Su función es inicializar y lanzar las instancias de procesamiento de eventos que hay detrás de las pruebas unitarias que se definan dentro del caso de prueba. En la Figura 6.3 se muestra el diagrama de clases empleadas para el funcionamiento del Motor de reglas.

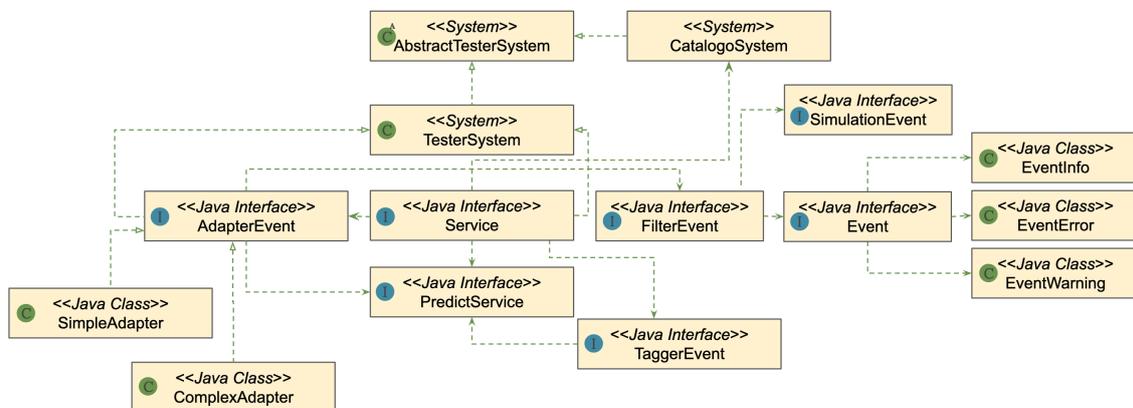


Figura 6.3: Diagrama de clases empleadas en el Motor de reglas.

El motor utiliza el procesamiento en capas descrito en la Sección 5.2.3, teniendo en cuenta los parámetros que establezcan en el caso de pruebas. Las instancias de procesamiento se basan en el uso de reglas encargadas de analizar y validar características concretas sobre los flujos de eventos. Las reglas se almacenan en el *Catálogo* y el motor accede a ellas siempre que las necesita para ejecutar una prueba. Las validaciones que se aplican, tienen en cuenta los criterios de éxito o fracaso que se han definido en la prueba unitaria, así como los parámetros de configuración de los elementos de la simulación útiles para la segmentación y el procesamiento de los flujos (ventanas de tiempo, ventana de datos, frecuencia de muestreo, entre otros). Como se indicó en la Sección 4.2.4, algunos criterios se deducen a partir de la identificación de patrones esperados, los cuales se

aplicarán según los criterios de validación establecidos por el usuario. Las clases que componen el Motor de reglas se describen a continuación:

- **AdapterEvent**: se encarga de aplicar filtros sobre los eventos, los cuales son elegidos por el usuario al momento de crear el caso de prueba. Su objetivo, es reducir el número de eventos que pasan de un flujo a otro entre capas. No todos los eventos son relevantes en el entorno y es por ello que desde el caso de pruebas, se hace necesario ofrecer un mecanismo al desarrollador que le permita seleccionar aquellos eventos que sí son relevantes. El framework ofrece tres tipos de filtros para trabajar con los flujos:
 - **Filtros predefinidos**: aplican varios criterios y umbrales de filtrado preestablecidos dentro del sistema. El primer criterio de filtro está relacionado con la tipología del dispositivo, el cual diferencia un sensor de otro. El segundo criterio tiene que ver con la naturaleza de los datos del sensor, el cual aplica reglas concretas cuando el sensor genera un único valor como lectura del entorno (filtrado simple). Este tipo de filtros se usa especialmente con los sensores ambientales.
 - **Filtros complejos**: están pensados para aquellos dispositivos cuyas lecturas se componen de series temporales. En este tipo de sensores, se utilizan reglas más elaboradas que se basan en el uso de ventanas temporales, en el tratamiento de la señal y en patrones sobre datos de un eje concreto.
 - **Filtros personalizados**: son los que el usuario puede crear programáticamente, añadiendo nuevas reglas que luego podrán usarse de forma genérica en la herramienta.
- **TaggerEvent**: se encarga de la segmentación del flujo de eventos mediante el uso de ventanas temporales w_t , cada cual establecida en el caso de pruebas correspondiente. De la misma forma, la segmentación se ve afectada por los criterios de validación definidos en el caso de prueba, siendo estos los que en último caso, establecen qué ventanas de datos se usarán en las siguientes etapas de proceso. Este componente, también se encarga de añadir información a los eventos relacionada con las características del sensor y su configuración en la simulación. El objetivo es que, desde las pruebas unitarias, sea más flexible el procesamiento entre capas, permitiendo al desarrollador manipular la estructura de los eventos dentro del sistema. Aunque esta información se puede añadir al sensor antes de iniciar la simulación, es más práctico que el desarrollador pueda configurarla en el caso de prueba sin necesidad de modificar la simulación.
- **Catalogo**: administra el acceso a la base de datos *Rules* que almacena las consultas SQL que describen los filtros predefinidos, las plantillas de etiquetas, la estructura de metadatos, las reglas en EPL y los identificadores de las actividades del usuario reconocidas por el sistema HAR. Las consultas se almacenan individualmente en formato texto, con un descriptor o identificador que permite al motor acceder a

ellas en el momento que las necesite. Esta forma de almacenarlas permite que las consultas sean reutilizadas en las distintas capas de procesamiento de eventos, descrita en la Sección 5.2.3. Para ejecutarlas, se emplea un sistema que hace las veces de esquemas, creado para armar la consulta principal antes de ser instanciadas por el motor. La arquitectura dispone de un conjunto de consultas predefinidas que se utilizan en distintas funcionalidades, tales como filtros predefinidos, umbrales, semáforos, etiquetas, criterios de validación, entre otros. Para los casos de pruebas, el desarrollador podrá definir nuevas consultas que actúen sobre los flujos de eventos que estime pertinente, siempre que éstas cumplan con el estándar que se ha definido en la arquitectura para el uso del sistemas de reglas.

- **PredictService:** se encarga de realizar la predicción del motor en función del catálogo de reglas y de los sistemas de reconocimiento de actividad implementados.
 - **Catálogo:** se usa cuando el objetivo del caso de prueba es detectar variaciones simples o comunes en la medida del sensor, el orden de ocurrencia de dichas variaciones, el cambio de estado y posición de los sensores y actuadores y patrones simples en las variaciones de la medida del sensor como: “up”, “down”, “left”, “right”, “change”, “gyro”.
 - **Sistema de reconocimiento de actividad:** se usa cuando en el caso de prueba es necesario identificar la actividad que realiza el usuario en la simulación. Su objetivo es aplicar modelos de clasificación pre-entrenados en la propia simulación, para reconocer movimientos complejos, actividades o animaciones que realiza el personaje dentro de la simulación o acciones específicas del usuario. El framework dispone de tres clasificadores que reconocen las actividades que incluyen por defecto las animaciones de los avatares, estas son “running”, “walking”, “drink”, “clap”, “get up”, “get down”.

TesterSystem define el comportamiento de los adaptadores que se aplican sobre los eventos y establece los servicios a los que se acceden desde las pruebas unitarias, tales como publicar, filtrar y etiquetar los flujos de eventos. Para utilizar los adaptadores, este se apoya en *AdapterEvent* para diferenciar los eventos simples y los eventos complejos usados para validar el comportamiento de los datos en la simulación. Desde *Service*, se hace uso de *AdapterEvent* y *TaggerEvent* para filtrar y etiquetar los eventos respectivamente.

Para trabajar con distintos tipos de eventos, se han definido *Event* como una implementación global de eventos genéricos y *SimulationEvent* como una implementación particular de eventos que provienen directamente de la simulación. La diferencia entre estas dos implementaciones, es que *Event* se creó para ser usado también por el sistema de reconocimiento, el cual demanda una serie de funcionalidades útiles en el proceso de extracción de características.

Dentro de la implementación global se hace uso de tres tipos de eventos concretos: *EventInfo*, *EventError* y *EventWarning*, los cuales se utilizan para determinar estados entre las fases de procesamiento del sistema de reconocimiento.

La secuencia de pasos para la transformación de eventos se presenta en el diagrama de la Figura 6.4. Inicialmente, mediante *InputEventAdapter* se hace una petición de publicar eventos de entrada que ha recibido el motor desde la simulación. *EventSystem* organiza el flujo como instancias de eventos de simulación, haciendo uso de *SimulationEvent*. Cuando participa el sistema de reconocimiento, se hace una segunda organización de los eventos pero esta vez haciendo uso de *Event* para crear las instancias. El motor dispone de los flujos obtenidos desde *Event* para hacer uso de ellos en las otras etapas de procesamiento del sistema de reconocimiento.

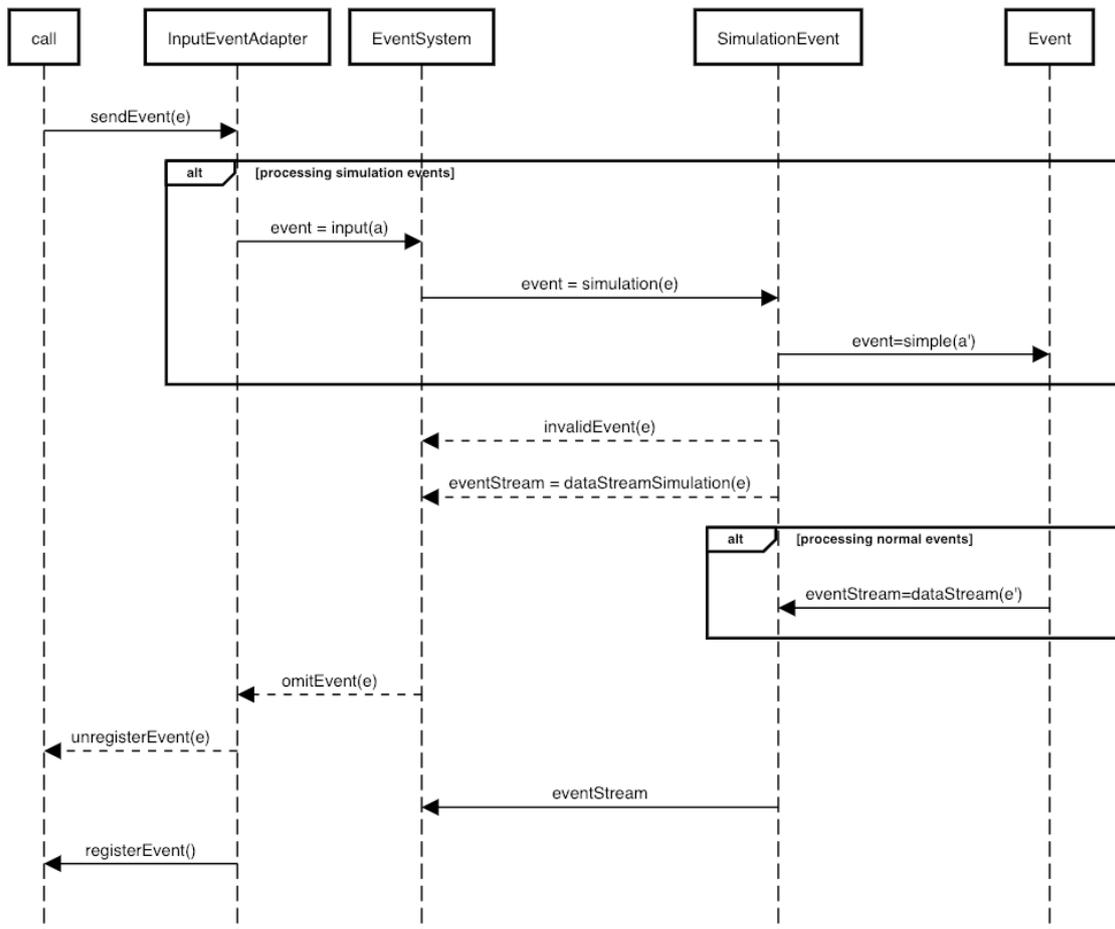


Figura 6.4: Diagrama de secuencia de la transformación de eventos realizada por el Motor de reglas.

La interacción de los componentes de filtrado y etiquetado en los flujos, se muestra en la Figura 6.5. *EventSystem* es responsable del streaming de datos y delega en *Service* la responsabilidad de recoger los eventos de entrada y procesarlos con ayuda de *AdapterEvent* y *FilterEvent*. Una vez son depurados, los flujos (ver Definiciones 5.2.2 y 5.2.3), se aplica sobre ellos el proceso de etiquetado con *TaggerEvent* para añadir metadatos que serán usados por el motor de reglas cuando pasen a otras capas de procesamiento.

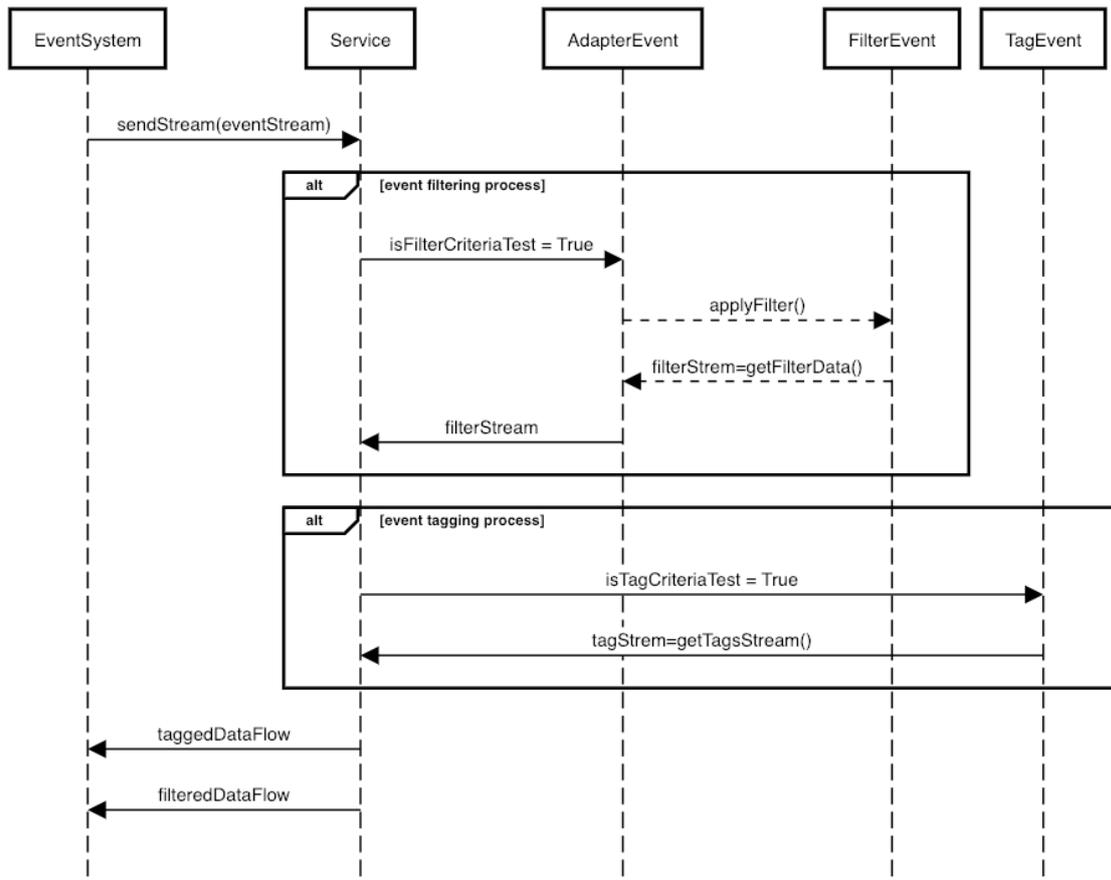


Figura 6.5: Uso de filtros sobre los flujos de eventos de la simulación

6.2 Filtrado basado en atributos

En la Figura 6.6, se muestra un ejemplo en el que se aplican filtros sobre la simulación. El filtrado en este caso, se lleva a cabo mediante el atributo “*type*”, el cual determina el tipo de sensor que se quiere tratar en el ejemplo.

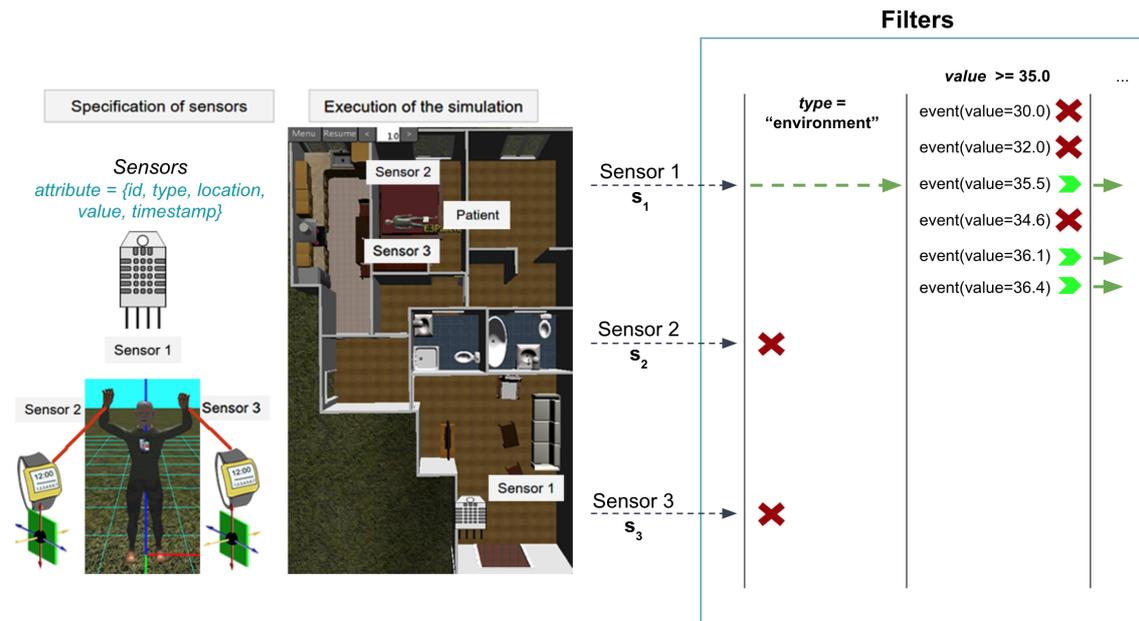


Figura 6.6: Filtrado de eventos mediante atributos.

En el escenario propuesto como ejemplo, existen tres sensores s_1 , s_2 y s_3 , de los cuales solo s_1 cumple el criterio del filtro. Uno de los atributos genéricos “*type*” indica esa característica en el sensor y mediante la condición ($type = “environment”$) es posible extraer solo aquellos eventos que la cumplan. Es así como los flujos se separan y son puestos a disposición de otros procesos u otras filtros. Los sensores s_2 y s_3 no cumplen esta condición por lo que *AdapterEvent* no los tiene en cuenta para aplicar las siguientes tareas. Cada sensor contiene una serie de atributos que son utilizados para establecer los filtros. En la Tabla 6.1, se muestran los dos tipos de sensores utilizados en la simulación con sus atributos y un ejemplo de filtro aplicado con cada uno de ellos.

Sensor	Atributo	Descripción	Filtro de ejemplo
- Entorno	id	identificador del sensor	r1:id = '01'
	position	posición del sensor en la escena	r1:position = 'door'
- Inercial	value	valor de la lectura	r1:value \geq 12 and r2:value.x \geq 8 r3:value.y \leq 7 and r4:value.z \geq 7
	type	tipo de sensor	r1:type = 'environment'
	frequency	frecuencia o muestreo	r1:freq = 2s

Tabla 6.1: Atributos y filtros utilizados para cada tipo de sensor.

6.3 Filtrado basado en umbrales e indicadores

Un umbral es un valor mínimo de una magnitud a partir del cual se produce un efecto determinado en la simulación. Un indicador por su parte, es una característica observable y medible usada para mostrar los cambios o los progresos que están ocurriendo en la simulación con respecto a una condición del entorno o del usuario. El filtrado basado en umbrales e indicadores, utiliza esta información precisamente para seleccionar subconjunto de eventos que cumpla con dichos umbrales o indicadores. Existen tres tipos de umbrales en la arquitectura de pruebas:

- **Umbrales estándares:** son umbrales que se han definido previamente para cada tipo de sensor, siguiendo las especificaciones de los fabricantes y el estándar común. Éstos le dan al desarrollador valores para etiquetar lo que ocurre en el ambiente dentro de la simulación.
- **Umbrales personalizados:** son umbrales que pueden ser modificados y adaptados a las necesidades de cada caso de prueba, utilizando para ello un identificador único por cada usuario y/o por cada simulación. Esta característica permite generar informes de validación de actividades que se adapten a los resultados de las pruebas unitarias.
- **Semáforos o escala de colores:** son umbrales que utilizan una codificación de colores para mostrar en los resultados de las pruebas. Esta escala es útil para representar visualmente, qué condiciones hay en una zona concreta de la simulación, en función de la actividad de los sensores ambientales. La escala de colores utilizada por defecto en la arquitectura y su correspondiente significado se muestra en la Tabla 6.2.

Verde	Verde: Área segura o resultado correcto
Amarillo	Amarillo: Área de precaución, resultado impreciso
Rojo	Rojo: Área de peligro o resultado incorrecta.

Tabla 6.2: Escala de colores empleada por defecto.

Los indicadores definidos en la arquitectura llevan consigo asociada una o varias reglas que identifican su presencia a través de umbrales en los sensores ambientales de la simulación. Estos indicadores describen condiciones concretas en áreas del entorno simulado que pueden ser utilizadas en las pruebas unitarias. En la Tabla 6.3, se muestran ejemplos de reglas que utilizan los indicadores en sus correspondientes consultas.

Indicator	Descriptor	Umbral (etiqueta)
Área con cierta luminosidad	@luminosity	Alto (@High), Medio (@Medium), Bajo (@Low)
Área con cierto nivel de ruido	@noise	Alto (@High), Medio (@Medium), Bajo (@Low)
Área con cierta temperatura	@temp	Alto (@High), Medio (@Medium), Bajo (@Low)
Área con movimiento	@moves	Fuerte (@Strong), Débil (@Weak), Poco (@Little)

Tabla 6.3: Indicadores por defecto del Motor de reglas.

Desde una regla, se puede hacer uso de otros elementos como filtros, umbrales e indicadores que se hayan definidos previamente. Todos los elementos del Catálogo pueden ser utilizados como parte de otras reglas para una determinada simulación. En estos casos, cuando se crea un caso de prueba, la arquitectura ofrece la posibilidad de definir instancias únicas dentro del Catálogo que hereden todos estos elementos y que a su vez, las diferencie del resto de instancias para evitar que desde otros casos de prueba, éstas sean modificadas. Como se observa en la Tabla 6.4, los indicadores son referencias a funciones concretas que permiten reconocer el estado del entorno, como por ejemplo “*poco iluminado*”, “*ruidoso*” o “*caluroso*”. Para poder ejecutar las reglas en función de los indicadores, el motor asocia a cada criterio otro conjunto de reglas que definen ese indicador y que luego se aplican sobre el flujo.

Regla	Sensor	Consulta
rule:threshold:low	*	select Low from EventFlowName where[$low_{min} \leq value \leq low_{max}$]
rule:threshold:medium	*	select Medium from EventFlowName where[$medium_{min} \leq value \leq medium_{max}$]
rule:threshold:high	*	select High from EventFlowName where[$high_{min} < = value < = high_{max}$]

Tabla 6.4: Consultas correspondientes a los umbrales predefinidos.

6.4 Resumen

En este apartado se ha descrito la arquitectura propuesta para construir los entornos de prueba para la simulación. Los dos componentes principales que la componen son, por un lado, un motor de reglas que utiliza el paradigma CEP para procesar todos los eventos generados en la simulación y, por otra parte, un sistema HAR que se emplea para describir el comportamiento del usuario dentro del entorno simulado. Estos dos componentes son esenciales para identificar los requisitos de la solución asistiva mediante el reconocimiento de las actividades o acciones que se han establecido en la simulación.

Para describir el comportamiento del entorno, el motor de reglas a grandes rasgos, aplica sobre los datos generados por los sensores procesos encargados de adaptar, filtrar, etiquetar y reconocer patrones dentro de los flujos de eventos establecidos. En el reconocimiento de la actividad física de los usuarios simulados, los sistemas HAR emplean modelos de clasificación de actividades para identificar acciones concretas. Estas acciones son animaciones por ordenador añadidas a los personajes, las cuales son modificadas para evitar que sean repetitivas y que los algoritmos de aprendizaje se sobreajusten con ellas. Todas estas funcionalidades se ponen a disposición del desarrollador para que cuando construya los entornos de prueba, en las pruebas unitarias pueda validar las actividades que definen el comportamiento del entorno simulado.

Capítulo 7

Implementación del sistema de reconocimiento

En este capítulo se describe el sistema de reconocimiento de actividad que ha sido implementado como parte de la arquitectura propuesta, para identificar las acciones del personaje en el escenario. En la Sección 7.1, se indica la necesidad de utilizar el sistema de reconocimiento en el entorno de pruebas y en la Sección 7.2, se presenta el sistema de reconocimiento que ha sido implementado. Asimismo se describen las técnicas de extracción de características utilizadas, los modelos de clasificación del sistema y la técnica empleada para el reconocimiento de patrones en movimientos simples que no describen una actividad como tal.

7.1 Sistemas de reconocimiento

Los sistemas de reconocimiento son componentes de la arquitectura que se encargan de reconocer las actividades o movimientos que realizan los personajes dentro de la simulación. Cuando los eventos en la simulación son originados por sensores que monitorizan actividades físicas como “*caminar*”, “*correr*”, “*beber*”, “*aplaudir*” o “*abrir o cerrar puertas*”, no es suficiente operar con reglas o patrones de eventos complejos para identificar qué actividad realiza el personaje. La razón de ello es que las personas no realizan siempre una actividad física de la misma forma, por lo que no es posible identificar patrones usando simplemente la aceleración del cuerpo sin más. Ante esta necesidad, los sistemas HAR llevan a cabo un mejor papel en la identificación de dichas acciones. La arquitectura para estos casos utiliza modelos de clasificación específicos que reconocen la actividad que está ejecutando el usuario en un momento dado. El principal uso que tiene el sistema HAR en los entornos de prueba es que permiten al desarrollador definir pruebas donde se compruebe si el usuario en la simulación realiza actividades que se espera lleve a cabo en determinados momentos de la simulación. Estas acciones incluso podrán ser parte de la respuesta que espera la solución asistiva para seguir funcionando o ejecutar otra tarea.

7.2 Clasificadores

Para construir el sistema HAR, sobre la arquitectura propuesta se han definido varios componentes estructurados en capas, los cuales son necesarios para llevar a cabo determinadas tareas dentro del proceso de reconocimiento de la actividad. Como se observa en la Figura 7.1, el componente *PredictService* de la arquitectura se acopla al sistema de reconocimiento aprovechando la estructura propuesta, permitiendo que desde la arquitectura principal se accedan a servicios específicos del sistema como el procesamiento de los eventos, el proceso de extracción de características y los algoritmos de clasificación empleados.

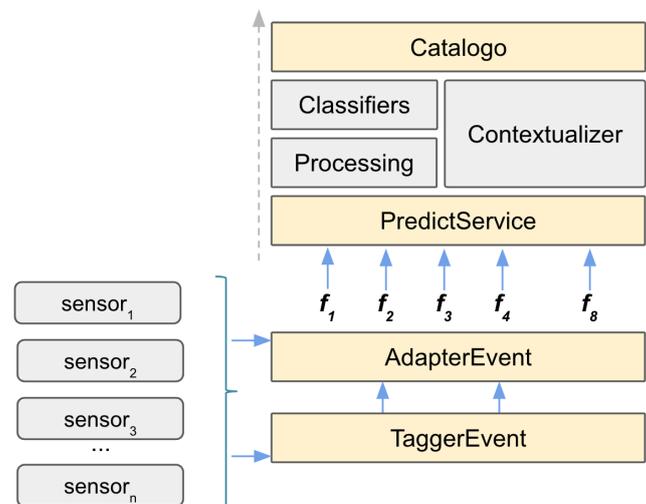


Figura 7.1: Componentes del sistema de reconocimiento de actividad.

Los elementos que componen el sistema HAR se describen a continuación:

1. **Flujos de eventos f_i :** están en el nivel inferior del sistema y representan el flujo de datos que proviene del motor principal. Estos eventos ya han sido tratados por *AdapterEvent* y *TaggerEvent* siguiendo la Definición 5.2.1, por lo que al sistema llegan con los atributos y metadatos necesarios para el procesamiento previo, antes de ser utilizados por los algoritmos de reconocimiento.
 2. **PredictService:** para esta parte del sistema, se encarga de orquestar la entrada de los diferentes flujos de eventos en función de su tipología y sus características técnicas. Orquestar los flujos es posible gracias a la información de filtrado y etiquetado del propio evento, el cual es procesado previamente por el Motor de reglas. Este componente también ofrece servicios concretos que permiten acceder a los distintos componentes del sistema HAR.
- Esta capa intermedia del sistema abstrae detalles de bajo nivel proporcionando independencia entre la simulación y el propio clasificador, permitiendo incluso la posibilidad de ampliar el número de sensores independientemente de su tipo.
3. **Processing:** este componente aplica sobre los flujos de eventos las técnicas de extracción de características que se consideren relevantes antes de emplear

el algoritmo de reconocimiento correspondiente. Las técnicas de extracción de características empleadas por defecto en el sistema, están asociadas con el cálculo de métricas estadísticas convencionales y con el tratamiento de los datos en el dominio del tiempo y la frecuencia.

4. **Clasificadores:** contienen los modelos de clasificación pre-entrenados que están disponibles en el entorno de pruebas. Actualmente, la arquitectura proporciona tres modelos de clasificación preconfigurados: *k-Nearest Neighbors* (kNN), *Support-vector Clustering* (SVC) y *Random Forest* (RF).
5. **Contextualizador:** este componente cumple dos funciones específicas dentro del sistema. Por un lado, asocia a la actividad física reconocida, información del contexto que se obtiene a partir de los metadatos en los sensores. Por otra parte, almacena en el *Catalogo* tanto las predicciones como la información que le ha sido asociada. El objetivo de esto es poder describir la actividad física reconocida junto con hechos que hayan ocurrido en el entorno en ese mismo momento, por ejemplo, “*el usuario camina y se enciende la luz de la habitación*” o “*el usuario camina, se encuentra en el baño y la puerta del refrigerador está abierta*”. Una vez se almacena en el *Catalogo*, esta información se podrá consultar a modo de histórico por cada caso de prueba, lo cual permite generar los informes correspondientes con los resultados de las pruebas unitarias ejecutadas.

7.2.1 Técnicas de extracción de características

En el framework propuesto, la actividad física se monitoriza mediante el uso de sensores inerciales de aceleración. Debido a la naturaleza de sus datos, estos requieren de una etapa de procesamiento relativamente compleja, en la cual se busca generar bloques de características distintivas que representen con precisión los datos originales [KJCP09]. El objetivo es transformar los eventos de entrada en un subconjunto reducido de datos que incluya aspectos característicos de las distintas actividades [HS05, WLT05, YWC08], para luego utilizarlas como entrada en los diferentes algoritmos de clasificación [PGK⁺09] empleados por el sistema.

Extraer las características claves de los datos permite a los algoritmos de aprendizaje encontrar patrones relevantes a la hora de diferenciar una actividad de otra. El número de técnicas utilizadas es muy amplio y la literatura sugiere el uso de una técnica antes que otra según la actividad. Para esta Tesis, los datos son representados mediante técnicas basadas en el dominio del tiempo y la frecuencia, utilizando en el primer caso, métricas estadísticas [WPP⁺07] comunes de bajo coste computacional, y en el segundo, los coeficientes derivados de las transformadas de Fourier [YY06] con el fin de discriminar señales periódicas o repetitivas.

7.2.1.1 Dominio del tiempo

- **Distancia Euclidiana:** cuando la actividad se monitoriza utilizando parejas de sensores inerciales sobre flujos de eventos f_{s1} y f_{s2} , se calcula la distancia euclidiana

como un valor característico. Este valor se define como la distancia numérica correspondiente entre los valores de las señales. Otra métrica que se ha incluido en este caso es el límite inferior de la distancia euclidiana o distancia mínima euclidiana [LKLC03]. La fórmula que calcula cada métrica se muestra en la Ecuaciones (7.1). Un elemento que se utiliza en la función es el valor de n como la longitud de las cadenas, en este caso $n = size(w_t)$, cuyos valores han sido cuantificadas mediante una distribución *gaussiana*. La función *dist* está representada por una matriz de distancia simétrica basada en los puntos de corte de la distribución anterior. Para la transformación se utiliza una fracción de normalización de la relación de compresión de la señal.

$$d_E(f_1, f_2) = \sqrt{\sum_{i=1}^n (|e_2^{t_i} - e_1^{t_i}|)^2} \quad (7.1)$$

$$dm_E(f_1, f_2) = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^n dist(e_2^{t_i} - e_1^{t_i})^2}$$

- **Media:** La media es una métrica muy utilizada en el tratamiento de la señal de cualquier tipo de sensor. Su bajo coste computacional y su poco uso de memoria la hace ideal para incluirla en el procesamiento de datos de sensores en streaming. En el caso de los sensores de aceleración, su uso es conveniente para minimizar el impacto del ruido y de los picos que puede generar este tipo de señal. Con respecto a la simulación, es útil para suavizar los defectos que pueden presentarse en la animación, los cuales por lo general se derivan en picos muy elevados que indican una mala interpolación de los movimientos del personaje. En la actividad humana real pueden presentarse estos mismos picos pero su valor suele ser bajo. En actividades donde la posición del cuerpo es relevante, la media resulta ser una métrica muy útil para diferenciar la postura [JKC07] del personaje (acostado, inclinado o sentado). Por otra parte, la media también es útil para cubrir las imperfecciones asociadas a la posición del sensor, la cual puede variar haciendo que las orientaciones de los ejes sea diferente.
- **Varianza r^2 y Desviación Estándar r :** la varianza es el promedio de las diferencias al cuadrado de la media y la desviación estándar r es la raíz cuadrada de la varianza, la cual representa tanto la variabilidad de un conjunto de datos como la distribución de probabilidad. En actividades pausadas y larga duración, la varianza y la desviación estándar son útiles para inferir el movimiento [SKK⁺05] del usuario en función de la estabilidad de la señal del sensor. En la realización de animaciones muy complejas, ambas métricas son menos útiles porque se sabe de antemano que la señal del sensor puede incluir valores espurios.
- **RMS** o Cuadrado Medio de la Raíz de una señal se compone de una secuencia de valores discretos z_1, z_2, \dots, z_n , que son obtenidos mediante la función $RMS = \sqrt{z_1^2 + z_2^2 + \dots + z_n^2}$. Esta métrica es utilizada para procesar datos de actividades

uniformes y repetitivas como caminar o correr. En el caso de la simulación, las actividades caminar y correr se llevan a cabo mediante animaciones que pueden ser cíclicas.

- **Correlación cruzada:** es una medida de similitud entre dos onda que suele utilizarse para buscar un patrón conocido en señales largas. Para calcular los coeficientes de correlación cruzada, se halla el producto escalar entre las señales, normalizando sobre el tamaño de ventana de muestreo n . En el caso de la aceleración, esta métrica calcula los coeficientes de correlación cruzada para los pares de señales correspondientes a los tres ejes (x, y) , (x, z) y (y, z) . Posteriormente, se selecciona el par de señales que exhibe los coeficientes más grandes [VBDV⁺96] para diferenciar las actividades.
- **Valor Máximo y Mínimo:** de una muestra de la señal de tamaño n , se toman los valores máximos o mínimos de ella en dicha muestra (n coincide con el tamaño de la ventana de tiempo w_t).

7.2.1.2 Dominio de la frecuencia

En actividades repetitivas o periódicas como “caminar”, “correr” o “aplaudir”, para extraer el conjunto de características, se ha utilizado la técnica de dominio de frecuencia F mediante el uso de la *Fast Fourier Transform* (FFT), la cual es considerada una de las técnicas más relevantes [PGKH09] entre todas las funciones de dominio de tiempo, frecuencia y wavelet sobre este tipo de señales. FFT permite extraer características basadas en el tiempo sobre señales periódicas o repetitivas. En las señales de aceleración se generan valores o coeficientes que van en función de la frecuencia seleccionada, de ahí la importancia de respetar la configuración recomendada de los sensores en los distintos escenarios simulados (ver Sección 3.5.1.3).

7.2.2 Modelos de clasificación

Los algoritmos que se han seleccionado para generar los modelos de clasificación de las actividades son tres: kNN, SVC y RF. Cada modelo está entrenado para reconocer las actividades “caminar”, “correr”, “beber”, “levantarse”, “sentarse” y “aplaudir”. El reconocimiento se lleva a cabo mediante uno o dos sensores de aceleración, ubicados en las muñecas del usuario. Las razones por las que han sido elegidas estas tres técnicas, se indican a continuación:

- Por su *precisión* y *exactitud* al reconocer la actividad del usuario.
- Por la *sensibilidad* a la hora de utilizar el menor número de sensores.
- Por el *tiempo de respuesta* en la clasificación y
- Por la baja *latencia* del reconocimiento entre las transiciones de actividades.

El uso de un clasificador está sujeto a las condiciones de lo que se quiere monitorizar en el personaje. En el momento que se define el caso de estudio, se debe valorar si las

especificaciones que tienen los clasificadores incluidos en el framework, son suficientes para cubrir las necesidades del caso de prueba o si por el contrario, es necesario ampliar la funcionalidad de los mismos. También es importante determinar el número de sensores que se utilizarán en el entorno, la posición del sensor y su ubicación sobre el personaje, el tamaño de la ventana temporal y la frecuencia de muestreo en cada dispositivo. Por defecto, el sistema contempla usar uno o dos sensores de aceleración según el algoritmo y de acuerdo a la actividad, como se muestra en la Tabla 7.1.

Clasificador	Algoritmo	Posición	Actividad (animación)
C1	RF	mano derecha, mano izquierda, ambas manos	clap drink
C2	kNN	mano derecha, mano izquierda, ambas manos	drink, clap, run, walking
C3	SVM	mano derecha, mano izquierda, ambas manos	levantar la mano (up), bajar la mano (down), drink

Tabla 7.1: Configuración por defecto de los clasificadores.

Una forma de evitar ruido a la hora de capturar los datos de sensores inerciales reales, es ajustar el dispositivo a la muñeca del usuario sin que haya holgura alguna que permita que este se mueva libremente. En el caso de la simulación, el sensor se posiciona como si fuese un reloj sobre el personaje, con la orientación y la posición de los ejes esperada por los modelos de clasificación implementados. En los casos de prueba, no es una preocupación posicionar el sensor sobre el usuario puesto que es la simulación la que se encarga de situarlos de manera correcta, como se observa en la Figura 7.2.

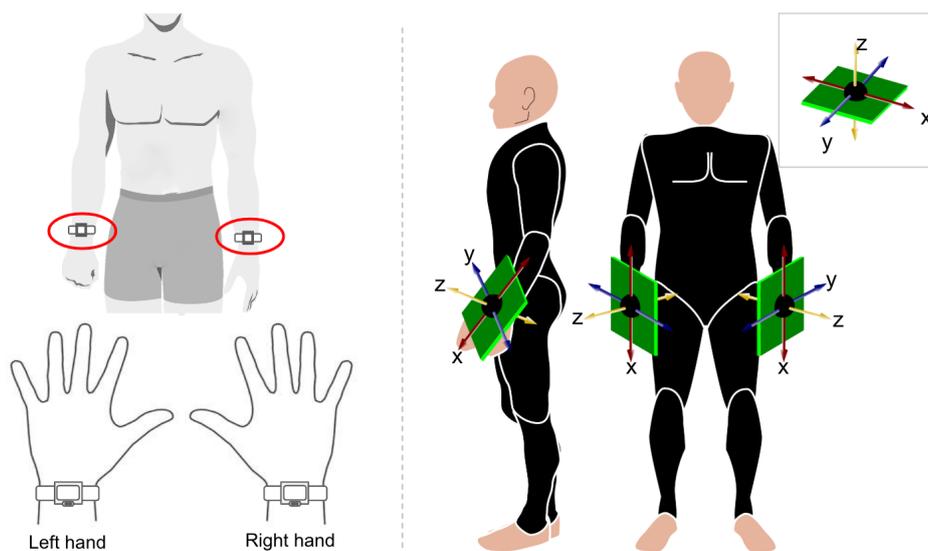


Figura 7.2: Posición de los sensores sobre el cuerpo del personaje.

7.2.2.1 Criterios de aceptación de los modelos

La precisión pr , la exactitud ac y la sensibilidad se se utilizaron para evaluar la confiabilidad de la técnica de clasificación y se calcularon utilizando las siguientes fórmulas:

$$pr = vp/(vp + fp) \quad (7.2)$$

$$ac = (vp + vn)/(vp + fn + vn + fp) \quad (7.3)$$

$$ac = vp/(vp + fn) \quad (7.4)$$

Donde vp es “verdadero positivo” y corresponde con las actividades clasificadas correctamente; fn o “falsos negativos” son las actividades clasificadas como una actividad diferente a la que realizó el usuario en ese instante de tiempo; tn o “verdadero negativo” son las actividades clasificadas correctamente como una actividad distinta del usuario y fp o “falsos positivos” son las actividades clasificadas correctamente a pesar de que no se realizaron.

Además de clasificar la actividad con una precisión alta, es importante que dicha clasificación se lleve a cabo en el menor tiempo posible, sobre todo en momentos claves de la simulación como es la transición de actividades. Las actividades no siempre tienen una duración constante, algunas incluso podrán tener una duración muy reducida que dificulta apreciarlas a simple vista. En la transición de este tipo de actividades es donde se espera una mayor atención del sistema, porque se espera que las técnicas aplicadas sean capaces de reconocer las transiciones incluso cuando no sea posible identificar la actividad como tal. En lo que respecta a los casos de prueba, las pruebas unitarias podrían usarse para identificar este tipo de acciones de corta duración, al tiempo que se comprueban otras situaciones del entorno. Si la técnica no cumple el reconocimiento de la actividad, se espera que por lo menos sea capaz de identificar que una situación distinta a la anterior, se ha llevado a cabo por el usuario.

Las pruebas realizadas con cada algoritmo propuesto se muestra en la Tabla 7.2. La elección de cada algoritmo ha sido gracias a su grado de precisión y exactitud a la hora de clasificar las actividades que el motor incorpora por defecto, según la configuración de los sensores de aceleración predefinida.

Actividad	Uso de un sensor			Uso de dos sensores		
	SVM	RF	kNN	SVM	RF	k-NN
walk	71.62 %	79.11 %	81.25 %	89.71 %	81.20 %	92.10 %
run	82.22 %	61.01 %	91.33 %	93.11 %	73.21 %	93.15 %
drink	63.87 %	72.57 %	89.47 %	75.15 %	61.23 %	94.61 %
get up	77.99 %	79.09 %	78.33 %	85.46 %	87.59 %	86.74 %
sit down	70.28 %	88.89 %	78.66 %	61.19 %	63.42 %	68.48 %
clap	79.78 %	75.77 %	85.29 %	75.23 %	79.20 %	88.13 %

Tabla 7.2: Precisión de la clasificación de los diferentes modelos con uno y dos sensores.

En la Tabla 7.3, se observa cómo varía el tiempo de respuesta en función del número de sensores utilizados por cada algoritmo. Entre los clasificadores actuales, el kNN en combinación con 1 y/o 2 sensores es el que mejor responde en tiempos a la hora de clasificar las actividades predefinidas. Este tiempo de respuesta hace que su latencia sea baja en las transiciones, con lo cual es útil al momento de diferenciar los cambios de actividad en la simulación empleando el menor número de sensores. En el caso del RF y el SVC, el tiempo de respuesta es bueno para el uso de 1 y/o 2 sensores, pero su latencia es muy alta en la identificación de los cambios de actividad.

Algoritmo	Sensores	Tiempo de respuesta	Latencia entre transiciones
kNN	1,2	0.0011 s	Baja
	3	0.0027 s	Baja
	4	0.0032 s	Baja
RF	1,2	0.0128 s	Media
	3	0.0161 s	Alta
	4	0.0181 s	Alta
SVC	1	0.1505 s	Alta
	2	0.1809 s	Alta
	3	0.2012 s	Alta
	4	0.2028 s	Alta

Tabla 7.3: Tiempos de respuestas y latencia de los algoritmos de clasificación utilizados

Aunque en la arquitectura se han implementado los tres modelos de clasificación, en cada prueba unitaria se podrá usar el sistema de reconocimiento que mejor se adapte a las necesidades del caso de prueba. En situaciones donde, por ejemplo, las actividades tienen una duración superior a 15 segundos, cualquier modelo de clasificación podría encajar perfectamente para validar el resultado de la predicción. Como se observa en la Tabla 7.2, algunos algoritmos responden mejor que otros en lo que respecta a precisión y para aprovechar esta característica, el desarrollador podría considerar emplear un algoritmo más especializado en función de esta métrica.

El dataset ¹ utilizado para esta valoración, cuenta con un total de 250 ficheros con datos de 6 actividades diferentes, las cuales han sido realizadas por 15 usuarios.

7.2.2.2 Modificación de los clasificadores

Como se ha mencionado previamente, los modelos de clasificación que se han implementado en la arquitectura cumplen con una serie de características que podrán ser útiles ante determinados casos de prueba. Ante simulaciones que aborden actividades diferentes a las aquí propuestas, es posible añadir o modificar los modelos de reconocimiento que han sido implementados por defecto y aplicarlos luego sobre los casos de prueba que se definan. Para ello, la estructura de los componentes en capas del sistema, posibilita no solo la modificación y el reemplazo de los modelos sino que también permite añadir nuevos modelos de clasificación que cumplan con unas condiciones concretas.

¹Dataset: <https://drive.google.com/drive/folders/1f1G0lv6BImSH27Mlj9f7BAyMgUvLp0a5?usp=sharing>

7.2.3 Configuración de los sensores

La configuración utilizada en el sistema HAR, corresponde a un acelerómetro que mide la aceleración en 3 dimensiones (triaxial), en un rango de hasta $\pm 4G$ ($G = 9.81m/s^2$), y con una tasa de muestreo de $35Hz$, equivalente a una frecuencia real de $100Hz$. Sin embargo, para reducir el consumo de energía y para favorecer el reconocimiento de las actividades por defecto en la simulación, con las técnicas de extracción de características propuestas, el sensor transmite a una frecuencia de $20Hz$ y $40Hz$ en función de la ventana temporal que se emplee en la prueba unitaria. Una ventana temporal w_t de $1seg.$ es conveniente que trabaje con una frecuencia de $20Hz$ en el sensor. En el caso de un $w_t = 2seg.$, la frecuencia recomendada en este caso es de $40Hz$, ajustándose así al proceso de aprendizaje empleado en los modelos de clasificación actuales. Cualquier otro valor en w_t , implicará usar una frecuencia de muestreo mínima de $Hz = 2 * w_t$ y máxima de hasta $100Hz$, que es lo recomendado después de realizar las diferentes pruebas con el sistema HAR.

7.2.4 Reconocimiento de movimientos simples

Como se ha definido en la Sección 4.2.5.1, un movimiento simple es un gesto concreto que realiza el usuario con una parte determinada de su cuerpo. El conjunto de movimientos simples puede suponer lo que se considera una actividad física cotidiana como “*correr*” o “*beber*”. Identificar una de estas actividades supone posicionar en el usuario uno o más sensores sobre su cuerpo de forma que se mida el movimiento sobre las partes de su cuerpo que resultan importantes para el desarrollo de dicha acción.

En la Figura 7.3 se muestra un ejemplo donde se utilizan tres sensores inerciales de aceleración para identificar la actividad “*beber*”. En dicho ejemplo, los sensores han sido posicionados sobre la mano derecha s_1 , la mano izquierda s_2 y sobre el pecho s_3 del personaje mientras este realiza la actividad.

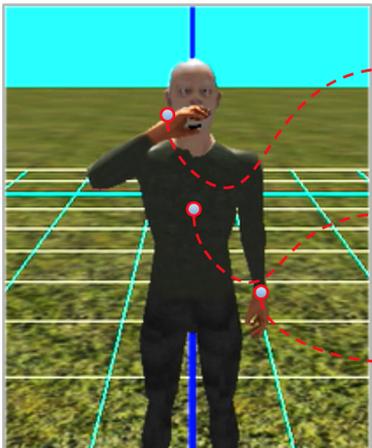
Simulación	Criterio	HAR
	expectativa: usuario mueve la mano derecha ('drink')	predicción: activity = 'drink'
	expectativa: usuario no se mueve ('stop')	predicción: activity = 'stop'
	expectativa: usuario no mueve la mano izquierda ('stop')	predicción: activity = 'stop'

Figura 7.3: Predicción de la actividad con 3 sensores.

De cara a realizar la comprobación que se quiere demostrar en esta Sección, sobre cada

uno de los sensores se define una prueba unitaria cuya salida esperada indicaría lo que debería hacer cada parte del cuerpo. En s_1 , al tratarse de una persona diestra, se espera que el sensor detecte que produce un movimiento significativo que indique que el usuario ha levantado la mano. En s_2 y en s_3 , al ser una actividad que solo implica la mano derecha del usuario, no se esperara que haya movimiento alguno.

Las salidas de las pruebas sobre los sensores, arrojarán un resultado que luego podrá ser interpretado y contrastado con la salida esperada. Cuando interese, la actividad “beber” podría ser interpretar como la combinación de gestos simples como “mueve mano derecha”, “no mueve mano izquierda” y “no mueve el pecho”. Con el uso de los sistemas HAR, bastaría con pasar los datos de los tres sensores y que sea este quien decida si la actividad que más se parece es la esperada o no. Según el interés de la prueba, podría resultar más interesante reconocer la actividad mediante gestos simples o como una actividad específica.

7.2.4.1 Utilizar eventos complejos en la predicción

Cada una de las predicciones obtenidas de los sensores en la simulación, es útil para validar algún aspecto concreto en el comportamiento de los usuarios. Sin embargo, cuando el número de predicciones es muy grande, puede que la comprobación de las distintas salidas requiera tiempo adicional por parte de los expertos. En estos casos, se pueden aplicar reglas que permitan validar que determinadas salidas se cumplen y lo hacen en un orden establecido dentro de los criterios del caso de prueba. Como se observa en la Figura 7.4, esta combinación de reglas permite analizar la interacción del usuario con cada dispositivo que monitorice el entorno.

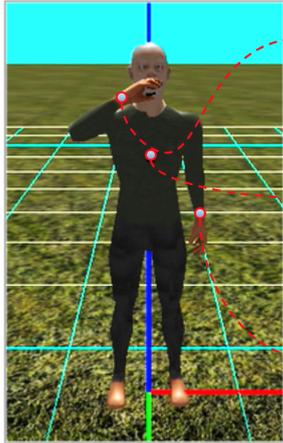
Simulación	Criterio	HAR	CEP
	S_1	predicción: activity = 'drink'	<i>rule:a:</i> s1:activity = 'drink'
	S_2	predicción: activity = 'stop'	<i>rule:b:</i> s2:activity = 'stop' <i>rule:c:</i> s3:activity = 'stop'
	S_3	predicción: activity = 'stop'	<i>rule:d:</i> s1:activity = 'drink' and s2:activity = 'stop' and s3:activity = 'stop'

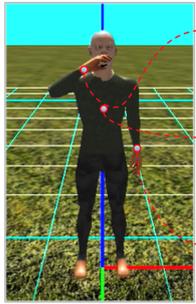
Figura 7.4: Uso de reglas para analizar los resultados de la predicción de actividades.

Desde esta perspectiva, tanto los gestos o acciones individuales del usuario como los cambios de estado del entorno, generan mucha más información que disgregada, alerta sobre lo que está ocurriendo dentro en la simulación en cortos intervalos de tiempo. En el caso del usuario, una situación ideal es la que se muestra en la Figura 7.4. Sin embargo, pueden ocurrir variaciones en el comportamiento del usuario que revelen otros problemas que se deben comunicar a los expertos. La acción de “beber” la lleva a cabo con la mano

derecha pero podría ocurrir que con la mano izquierda padezca un episodio de temblores que afecte su estado general. Otra variación podría ser que el usuario “beba” mientras su cuerpo se balancea, lo cual es un claro síntoma de problemas de postura.

Estos mismos criterios se podrían aplicar sobre casos de pruebas que validen gestos más específicos en el tiempo de simulación. Las pruebas unitarias en este caso, son más concretas y no buscan identificar una actividad como tal, sino un conjunto de movimientos o gestos más simples a lo largo del tiempo. Como se describe en la Sección 4.3.1, los casos de prueba para validar la simulación se diseñan de forma ordenada, al igual que la lista de actividades, gestos o acciones que se quieren identificar.

En el ejemplo que se muestra en la Figura 7.5, cada sensor dispone de una prueba unitaria que validará una condición concreta. La regla *rule:a* valida que el personaje realice un conjunto de gestos en instantes de tiempos t_i consecutivos: *orden 1*: s_1 sobre la muñeca en t_{i+1} debe detectar que el usuario realice el gesto “up”, *orden 2*: s_1 sobre la muñeca en t_{i+2} debe detectar que el usuario esté en reposo y *orden 3*: s_1 sobre la muñeca del usuario en t_{i+3} debe detectar que el usuario realice el gesto “down”). Una segunda regla *rule:b* valida que el sensor cambie de posición al mismo tiempo. En este ejemplo, la segunda regla es opcional y sólo se utiliza como *testigo* para confirmar que no haya un desplazamiento del propio sensor. De este mismo modo, se aplican otro par de reglas similares (s_2 : *rule:c*, *rule:d*; s_3 : *rule:e* y *rule:f*) en los sensores s_2 y s_3 .

Simulación	Criterio	HAR	CEP		
	s_1 expectativa: usuario mueve la mano derecha ('drink')	predicción: activity = 'drink'	<i>rule:a:</i> s1:move = 'up' <i>rule:b:</i> s1:position = 'change'	<i>rule:a:</i> s1:move = 'none' <i>rule:b:</i> s1:location = 'none'	<i>rule:a:</i> s1:move = 'down' <i>rule:b:</i> s1:location = 'change'
	s_2 expectativa: usuario no se mueve ('stop')	predicción: activity = 'stop'	<i>rule:c:</i> s2:move = 'none' <i>rule:d:</i> s2:position = 'none'	<i>rule:c:</i> s2:move = 'none' <i>rule:d:</i> s2:location = 'none'	<i>rule:c:</i> s2:move = 'none' <i>rule:d:</i> s2:location = 'none'
	s_3 expectativa: usuario no mueve la mano izquierda ('stop')	predicción: activity = 'stop'	<i>rule:e:</i> s2:move = 'none' <i>rule:f:</i> s2:position = 'none'	<i>rule:e:</i> s2:move = 'none' <i>rule:f:</i> s2:location = 'none'	<i>rule:e:</i> s2:move = 'none' <i>rule:f:</i> s2:location = 'none'

w_t ventana de tiempo w_t w_{t+1} w_{t+2}

Figura 7.5: Uso de reglas para analizar los resultados de la predicción de gestos o acciones.

Como resultado del caso de prueba, se ha validado la realización de una actividad o animación durante la ventana temporal w_t (ver Figura 7.3). En el último ejemplo, la validación de la misma actividad se ha basado en los gestos que la componen, los cuales se han ejecutado en un orden propuesto y en un tiempo t_i durante distintas ventanas temporales w_t . En este caso, el experto ha definido la secuencia de gestos “up”, “stop” y “down” como el criterio de éxito que se espera del caso de prueba. Sin embargo, se podría definir otro criterio que utilice no solo el sensor principal sino también los sensores secundarios. Es decir, los sensores s_2 y s_3 podrían ser útiles para comprobar si el personaje realiza simultáneamente otro gesto mientras ejecuta la acción principal: *inclina el pecho hacia adelante mientras bebe, utiliza ambas manos para beber*, entre otras.

7.2.4.2 Modificación de las animaciones

Las actividades del usuario se modelan en la simulación con ayuda de las animaciones por ordenador. Cada animación se compone de una secuencia de acciones en el personaje que se rigen por alguna técnica que modela el movimiento humano. En un motor de videojuegos, una de las técnicas utilizadas es la animación por esqueleto, la cual consiste en una jerarquía compuesta por un conjunto de huesos y articulaciones. Cuantos más huesos hagan parte de la estructura, más realismo tendrá el movimiento [KMTM⁺98, AMHH18]. Como se observa en la Figura 7.6, solo se han utilizado articulaciones principales que rigen acciones muy general.

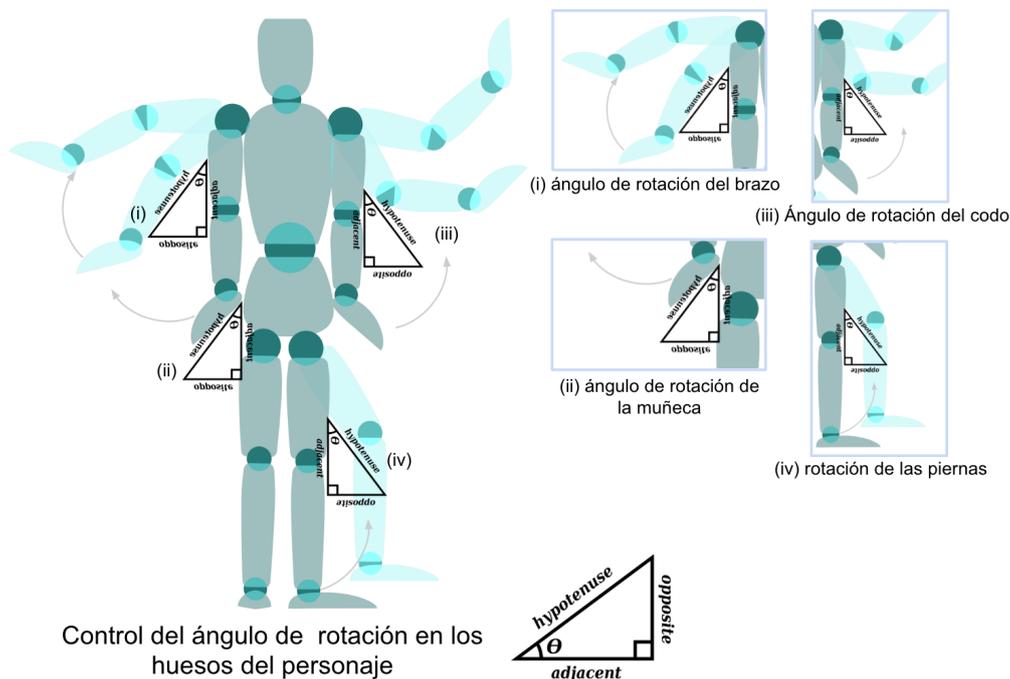


Figura 7.6: Ángulos de rotación y traslación en las articulaciones principales del personaje.

Las acciones están asociadas al movimiento de una parte del cuerpo y se caracteriza principalmente por los cambios de parámetros que describen el movimiento de esa zona del cuerpo. Estos cambios afectan la rotación y traslación de la parte del cuerpo en cuestión y varían según el tipo de acción. Las acciones son de dos tipos: periódicas y no periódicas. En una acción no periódica como “levantar el brazo”, los parámetros modifican la duración de la acción. En las acciones periódicas, los parámetros fijan el número de periodos que define la acción en si. La duración en este caso, se define multiplicando el número de periodos por el período de acción.

En la simulación, para facilitar la modificación de las acciones, es posible alterar estos parámetros en la estructura de huesos. Cada parámetro rige la movilidad del personaje y pueden variar entre dos valores límites, basados en los grados de libertad de la rotación permitida. Actualmente, los motores de videojuegos tienen la capacidad de incorporar modelos de física que facilitan la construcción de las animaciones. Con el uso de estos parámetros, los controles aplicados sobre los huesos pueden ser usados de forma externa sobre el personaje para modificar el comportamiento de sus articulaciones.

El framework propuesto, aprovechando las funcionalidades que ofrece la herramienta *AIDE* [Gra15], cuenta con la capacidad de modificar el comportamiento de las articulaciones en el personaje. Como se observa en Tabla 7.4, mediante una serie de comando programados en formato *JavaScript Object Notation* (JSON), se aplican nuevos valores de rotación y traslación a cada articulación principal del cuerpo: “*arms or shoulder*”, “*elbow*”, “*wrist*”, “*trunk*” y “*legs or knees*”. Cada articulación del personaje, dispone de un conjunto de modificaciones que están predefinidas en el Catalogo. La modificación se aplica con comandos en la simulación antes de que esta inicie, provocando que el desplazamiento y el comportamiento del personaje varíen. En total hay 30 modificaciones sobre el conjunto de huesos y cada modificación hace diferente el comportamiento físico del personaje, como si se tratase de 30 usuarios con características diferentes.

Atributos	Descripción	Ejemplo
id	Identificación del comando	movRightArmCommand012
class	Clase Java que ejecuta la modificación del movimiento	MovRightArmCommand
bone	Hueso a rotal	“ <i>arms</i> ”, “ <i>elbow</i> ”, “ <i>wrist</i> ”...
angular	Momento angular de la rotación	3.14159265
minangle	Mínima amplitud de la rotación del ángulo	-1.51707964
maxangle	Máxima amplitud de la rotación	0
side	Posición del hueso en el esqueleto	R
active	Activar o desactivar la rotación	False/True

Tabla 7.4: Atributos utilizados en la modificación de la animación.

La estructura ósea cuenta con un conjunto de etiquetas que permiten su manipulación dentro de la simulación. Por ello, es posible aplicar estos parámetros directamente sobre una articulación concreta (ver Listado 7.1). En la práctica, es posible mover todos los huesos que tiene el personaje, pero para que estas modificaciones sean coherentes, hay que procurar que los ángulos de rotación y de traslación sean coherente con la estructura osea del personaje y con las distintas animaciones que se quieren alterar.

```

//Identification of bones and spatial positioning.
@Override
public void setSpatial(Spatial spatial) {
    super.setSpatial(spatial);
    if (spatial != null) {
        skeletonControl = spatial.getControl(SkeletonControl.class);
        hand = skeletonControl.getSkeleton().getBone("RightForeArm");
        rotation = new Quaternion();
    } else {
        resetHand();
        skeletonControl = null;
        rotation = null;
    }
}

```

Listing 7.1: Identificación de los huesos en tiempo de ejecución.

Al igual que cualquier otro elemento en el escenario 3D, tanto el posicionamiento espacial como el pintado de los huesos, se lleva a cabo dentro del bucle de la renderización general que se muestra en el Listado 7.2. Esto hace posible que, al modificar los valores límites de las articulaciones, con cada nueva ejecución de la simulación se puedan observar

los cambios sobre el personaje. Estos nuevos parámetros afectan no solo al personaje en si, sino también a las distintas animaciones que este ejecute después de aplicada la modificación.

```
//Rendering of bone movement.
@Override
protected void controlUpdate(float fps) {
    if (hand != null) {
        setUserControlFrom(hand, true);
        hand.getCombinedTransform(position, rotation);
        updateRotation(rotation, fps);
        hand.setUserTransforms(position, rotation, Vector3f.UNIT_XYZ);
        updateBonePositions(hand);
    }
}
```

Listing 7.2: Renderizado del movimiento de los hueso.

```
//Update of bone mobility parameters.
@Override
private void updateBonePositions(Bone bone) {
    Transform t = new Transform();
    for (Bone b : bone.getChildren()) {
        t = b.getCombinedTransform(bone.getModelSpacePosition(), bone.getModelSpaceRotation());
        b.setUserTransformsWorld(t.getTranslation(), b.getModelSpaceRotation());
        updateBonePositions(b);
    }
}
```

Listing 7.3: Actualización de los parámetros de movilidad del hueso.

Las modificaciones predefinidas, básicamente realizan dos cambios concretos en las articulaciones. Por un lado, establecen distintos límites de amplitud del ángulo de giro del hueso principal y por otra parte, cambian la velocidad del movimiento [BMvD⁺08]. Cuando se modifica el ángulo de rotación esperado en un hueso, respetando sus límites naturales [HP97, SHP04], el motor de la física puede realizar los cálculos correspondientes para generar los movimientos derivados. Este cálculo utiliza estas dos modificaciones para generar los valores deseados en cada articulación [HP08]. En el caso de las actividades periódicas, la ecuación de control que suele utilizar la física para calcular estos nuevos valores $\tau = \kappa(\theta_d - \theta) + \kappa_v(\bar{\theta}_d - \bar{\theta})$, donde θ es el ángulo de la articulación, θ_d es el ángulo deseado, $\bar{\theta}$ es la velocidad de la articulación, $\bar{\theta}_d$ es la velocidad deseada y κ y κ_d son la ganancia proporcional y derivada. Mediante esta ecuación, es posible que desde la observación de los movimientos del personaje, se pueda obtener los límites en este tipo de acciones.

Cuando no es posible determinar con facilidad los límites del movimiento, lo cual puede ocurrir cuando el usuario tiene algún problema que altera totalmente su movilidad, lo ideal en estos casos es utilizar algún sistema que permita capturar movimientos concretos. En estos casos, en esta Tesis se ha utilizado el sistema *Perception Neuron* [Neu17], el cual cuenta con 32 sensores que se distribuyen sobre las articulaciones principales. En los casos donde no es posible determinar los ángulos de rotación, este dispositivo cuando captura las posturas del usuario, ofrece información detallada del movimiento, incluido los ángulos de giro en cada extremidad.

7.3 Resumen

El reconocimiento de la actividad humana es un elemento importante en soluciones donde la interacción entre las personas, el entorno y la tecnología es constante. Las soluciones asistivas se caracterizan por tener esta necesidad, de ahí que sea importante establecer algún mecanismo que permita validar su funcionamiento en la simulación. En este apartado, se ha descrito el componente de la arquitectura encargado de hacer el reconocimiento de las actividades físicas de los usuarios simulados. Para ello, se han utilizados varias técnicas de clasificación como RF, kNN y SVC, cuyo propósito ha sido identificar las actividades incluidas en la simulación. Las actividades representaron acciones físicas que ejecutaba el personaje como parte de las tareas en la simulación. Las animaciones usadas en la experimentación fueron *“caminar”*, *“correr”*, *“sentarse”*, *“levantarse”*, *“beber”*, *“aplaudir”*, *“saludar”* y *“abrir o cerrar puertas”*.

El sistema de reconocimiento propuesto tiene la tarea de identificar estas acciones y validar que se cumplen en la simulación cuando éstas son esperadas en el contexto de la solución asistiva. Es decir, la solución espera que el usuario interactúe con el entorno empleando alguna animación predefinida y una vez ésta es identificada, la solución podrá actuar en consecuencia a dicha acción y el entorno de prueba podrá validar que ésta se ha llevado a cabo en lo simulado.

Capítulo 8

Experimentación con casos de prueba

En este capítulo se describe la experimentación llevada a cabo para validar que la simulación, como representación de la solución asistiva, cumple con las especificaciones de asistencia necesarias, aplicables en un entorno real. En la Sección 8.1, se muestra el alcance de la experimentación propuesta, basándose en el alcance de la simulación propuesto en la Sección 3.2 y en las características de las soluciones asistivas descritas en la Sección 3.3.

Los casos de estudio que han sido utilizados en el experimento se presentan en la Sección 8.2, estos han sido definidos siguiendo las pautas contenidas en la Sección 3.4. Las características de la solución asistiva que han sido trasladadas a la simulación, según lo propuesto en la Sección 3.5, se detallan en la Sección 8.2.3 de este Capítulo.

El entorno de prueba que se ha propuesto para la experimentación se muestra en la Sección 8.3. Tanto los casos de prueba como las pruebas unitarias se definieron a partir de las condiciones del framework propuesto en el Capítulo 5. Para ejecutar las pruebas unitarias sobre la simulación, la arquitectura desarrollada en el Capítulo 6 se ha utilizado para ejecutar las diferentes validaciones contenidas en los casos de prueba. Finalmente, los resultados obtenidos se resumen en la Sección 8.4.

8.1 Alcance de la experimentación

La experimentación realizada en este trabajo se ha llevado a cabo en dos fases. En una primera fase se usó la metodología descrita en el capítulo 3 para definir los casos de estudio y sobre ellos aplicar tres soluciones asistivas. En todos los casos de estudio la simulación se utilizó como elemento principal de especificación. Las soluciones asistivas propuestas se han basado en casos de estudio cuyo objetivo era observar diferentes problemas que pueden afrontar personas mayores en su día a día. El propósito final de las soluciones es diseñar y desarrollar servicios que cubriesen dos necesidades en el entorno del usuario, estas son:

1. Realizar la monitorización a largo plazo del usuario mientras realiza sus actividades cotidianas dentro del hogar.

2. Hacer el reconocimiento de las actividades físicas que lleva a cabo el usuario mientras es monitorizado.

En la segunda fase de la experimentación, el trabajo se centró en definir los casos de pruebas necesarios para validar las simulaciones propuestas en la fase anterior, empleando para ello el framework propuesto en el capítulo 5. Los casos de prueba permitieron ejecutar las validaciones necesarias para garantizar que la simulación, en todos los escenarios, cumplía con las especificaciones descritas en el caso de estudio propuesto. Para uno de los casos propuestos, se han ejecutado validaciones adicionales una vez ha sido implementado en un entorno real. Esta implementación ha sido posible gracias al desarrollo de un prototipo en tecnología asistiva, diseñado a partir de la configuración en lo simulado y de las necesidades de asistencia observadas.

Los pasos que se han seguido para desarrollar la primera fase se muestran a continuación.

Fase 1: pasos para definir y probar la solución asistiva.

1. Identificar el **Caso de estudio**:
 - a. Usuarios.
 - b. Requisitos.
2. Definir la **solución asistiva**:
 - a. Alcance y restricciones.
 - b. Escenarios simulados.
 - c. Definir la lista de actividades.
 - d. Elementos en la simulación.
 - e. Elementos del dominio de la solución a tener en cuenta.
3. **Probar las simulaciones**: *escenario típico, escenario problema y escenario de la solución.*

Los pasos seguidos para llevar a cabo la segunda fase se indican a continuación.

Fase 2: pasos para validar la simulación.

1. Comprobar la configuración de los sensores utilizados.
2. **Definir las validaciones** que:
 - Necesitan de sensores ambientales e inerciales.
 - Demandan el uso de reglas o del reconocimiento de actividad.
3. **Identificar los parámetros** necesarios para cada validación.
4. Definir los **criterios de éxito y fracaso**.
5. **Crear las pruebas unitarias** necesarias para las validaciones, utilizando:
 - **Los parámetros de entrada** identificados en el punto 3.
 - **Los criterios de éxito o fracaso** identificados en el punto 4.
6. **Establecer los casos de prueba** a partir del conjunto de pruebas unitarias definidas en el punto anterior. Considerar el uso de bancos de prueba (*Test Suite*) para organizar los casos creados.

8.2 Casos de estudio

Para definir los casos de estudio utilizados en esta experimentación, se ha hecho uso del catálogo de entrevistas del proyecto Desarrollo Colaborativo de Soluciones AAL (ColosAAL). Este repositorio incluye un total de 20 entrevistas correctamente documentadas y publicados en la web¹ El catálogo se ha creado a partir de la información obtenida de entrevistas previas realizadas a personas mayores. Estos usuarios, al momento de la entrevista, padecían sintomatologías comunes relacionadas con trastornos conductuales y deterioro cognitivo que afectaban su independencia dentro del hogar. En las entrevistas también participaron sus familiares y cuidadores, quienes indicaron cómo estos problemas de salud afectaban al resto de la familia. Las entrevistas fueron diseñadas por profesionales en el área de la sociología, psicología y medicina, quienes también participaron en la tabulación y el análisis de los resultados. Asimismo, profesionales en ciencias de la computación han contribuido con herramientas y plataformas de apoyo para el análisis en grupo de los distintos casos.

El objetivo para esta experimentación es elegir los casos del catálogo que, siguiendo la metodología propuesta en la Sección 3.4, sean útiles para identificar necesidades alrededor del usuario que puedan favorecerse por algún tipo de solución asistiva o parte de ella. Básicamente se buscaba que las soluciones que se propusieran les apoyaran en sus actividades diarias o simplemente monitorizaran sus acciones a largo plazo.

8.2.1 Usuarios

La primera consideración ha sido definir casos de estudios en los que hayan participado usuarios con los roles descritos Sección 3.4.1. El propósito de esto era contar con perfiles de usuarios que hayan hecho algún aporte al momento de definir requisitos asociados con las necesidades del paciente, las condiciones del entorno y las posibles recomendaciones de mejora. En este sentido, los grupos de usuarios que hicieron algún aporte son:

- Los usuarios entrevistados, entre los cuales estaba el paciente y sus familiares. Este grupo según la metodología propuesta, representa a los *Expertos en la asistencia* o la solución. El hecho de que sobrelleven diariamente los distintos padecimientos, los hace expertos para determinar lo que debería aportar la solución en su día a día para que la mejora en calidad de vida sea efectiva.
- Los profesionales en áreas de sociología, psicología y personal médico que colaboraron en la construcción y el análisis de las entrevistas realizadas. Este grupo representa a los *Técnicos e ingenieros* de la solución y son quienes han aportado los aspectos sociales y médicos a tener en cuenta en la definición de los casos de estudio.
- Los ingenieros o informáticos que contribuyeron con la creación de herramientas que permitiesen recoger y analizar toda la información obtenida, así como en la definición de las distintas simulaciones empleadas para representar los casos de estudio. Finalmente, este grupo representa a los *Técnicos e ingenieros* de la solución.

¹Acceso público: <http://grasiagroup.fdi.ucm.es/aidendd/cases-list-1-10/>

8.2.2 Requisitos

Como se ha indicado anteriormente, en cada caso de estudio propuesto el objetivo era usar la simulación como un artefacto para representar las características de este. La simulación es propiamente la especificación de los requisitos y además, incluye las consideraciones necesarias para construir los casos de prueba. Para poder emplear la simulación en todo el proceso, en la definición de los casos de estudio se tuvo en cuenta que fuese posible determinar los requisitos que describían el entorno, al usuario y a los dispositivos empleados en la simulación (ver Sección 3.4.2). Todos estos requisitos se han condensado en el caso de estudio y por ende en la simulación.

8.2.2.1 Captura de requisitos

Parte del trabajo de esta Tesis, ha sido la construcción de herramientas que facilitasen la recolección de información de todos los usuarios implicados. Como se observa en la Figura 8.1, una de las herramientas Web que se construyó es **HackWithPeople**², la cual presenta distintos casos de estudio relacionados con síntomas muy específicos de enfermedades como Alzheimer y Párkinson.

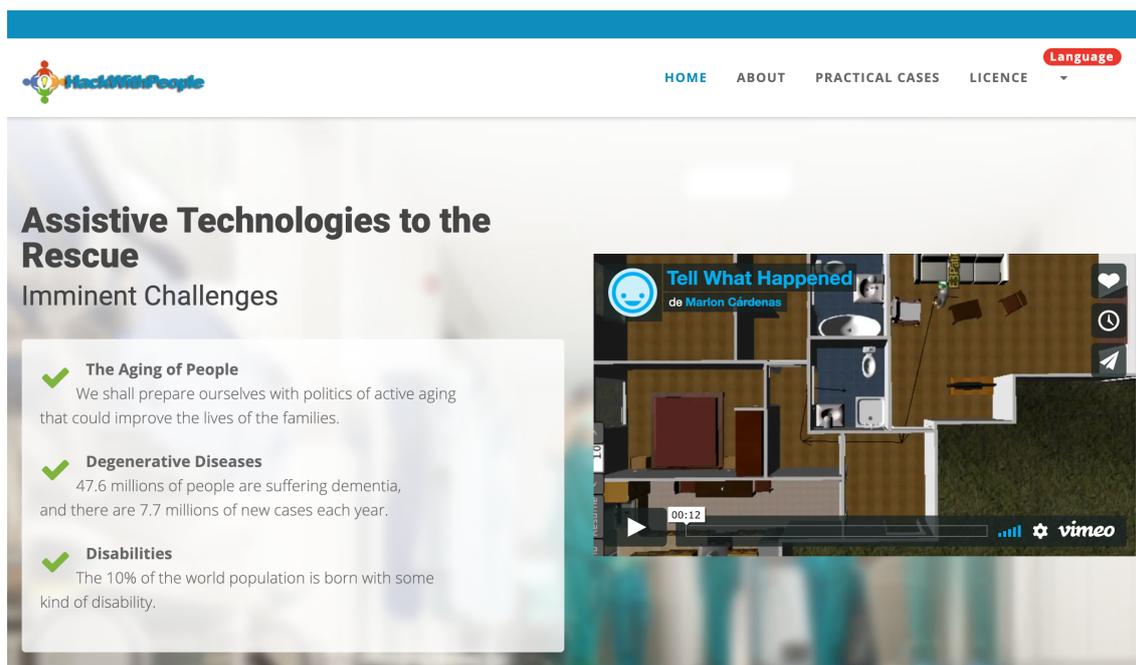


Figura 8.1: Herramienta web para el estudio de tecnología asistiva, aplicable en personas mayores con determinados padecimientos.

Como se observa en la Figura 8.2, en la herramienta se muestran distintas simulaciones que representan los casos de estudio, así como una descripción general que introduce al participante en el contexto de la situación analizada. En cada caso de estudio, los usuarios interesados pueden observar y manipular la simulación, la cual se presenta en formato vídeo para facilitar la interacción con su contenido. La herramienta ofrece una serie de controles que le permiten iniciar, pausar, adelantar o atrasar, entre otras funciones,

²HackWithPeople, acceso público: <http://grasia.fdi.ucm.es/hack4people/index.php>

la reproducción del vídeo (ver Figura 8.3). La participación de los usuarios consiste en participar en sesiones de grupo mediante grupos focales, técnica de trabajo en grupo propuesta en la Sección 3.4.4. En el grupo focal los usuarios participan principalmente, generando comentarios o sugerencias cortas que, a su criterio, señalan aspectos positivos y negativos de la solución presentada en la simulación.

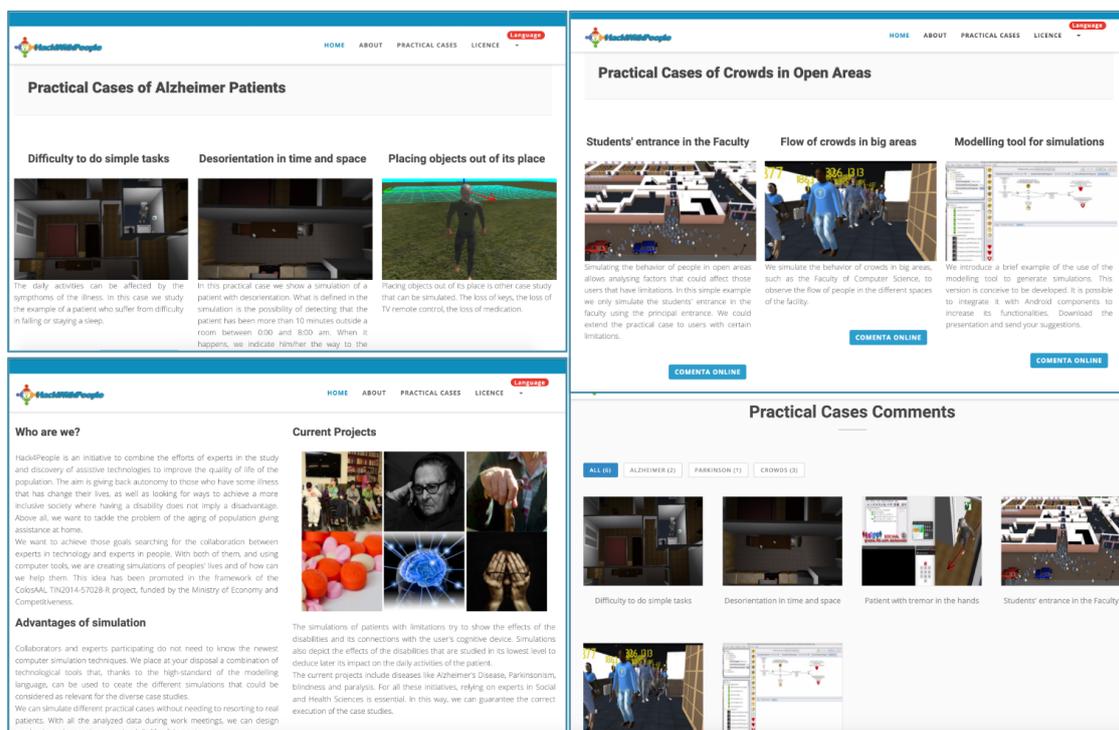


Figura 8.2: Estudio de síntomas específicos en enfermedades degenerativas.

En la Figura 8.4 se muestran los controles que ofrece la herramienta al usuario, para facilitar la tarea de añadir dichos comentarios sobre el vídeo en los instantes de tiempo que interesen al participante. A partir de las entrevistas, el equipo de ingenieros ha realizado un análisis sobre la información recogida y de ella ha obtenido ideas que ha considerado útiles para dirigir la construcción de la simulación.

Con este análisis en mano, el paso siguiente ha sido plasmar en la herramienta Hack4People³ diferentes aspectos de la solución, tales como:

- Si es pertinente el caso de estudio propuesto.
- El alcance y las restricciones propuestas inicialmente.
- Una primera versión de los escenarios, en la cual se tiene en cuenta el aspecto visual del entorno.
- Los elementos que se han incluido en la simulación: entorno, usuarios y tecnología.
- Una primera aproximación de lo que sería la lista de actividades, la cual con cada revisión de los usuarios, se irá modificando hasta definir la lista de actividades que se utilizará en el caso de estudio.

³Hack4People, acceso público: <http://grasiagroup.fdi.ucm.es:3128/index.php>

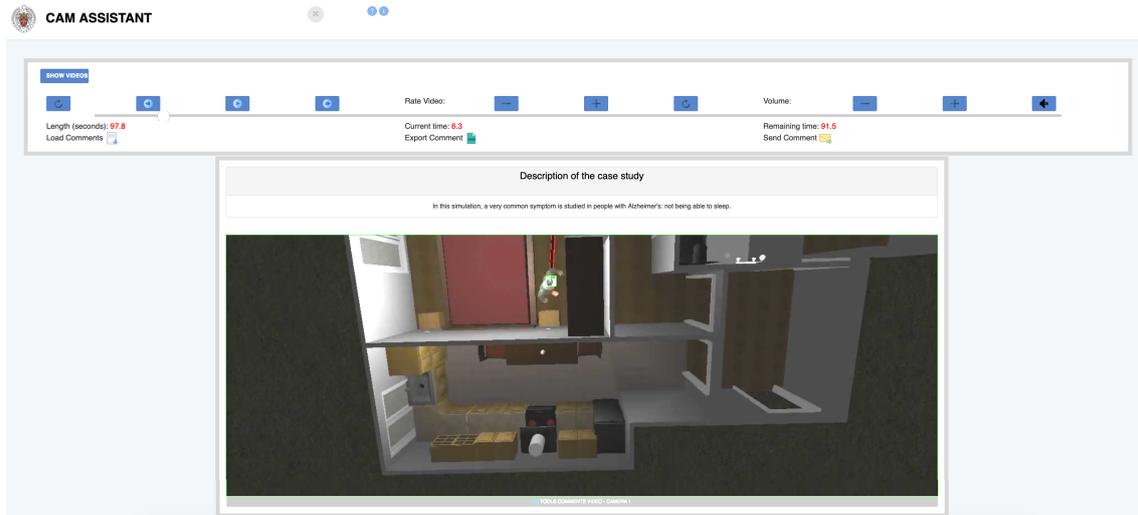


Figura 8.3: Visualización de las simulaciones en la web.

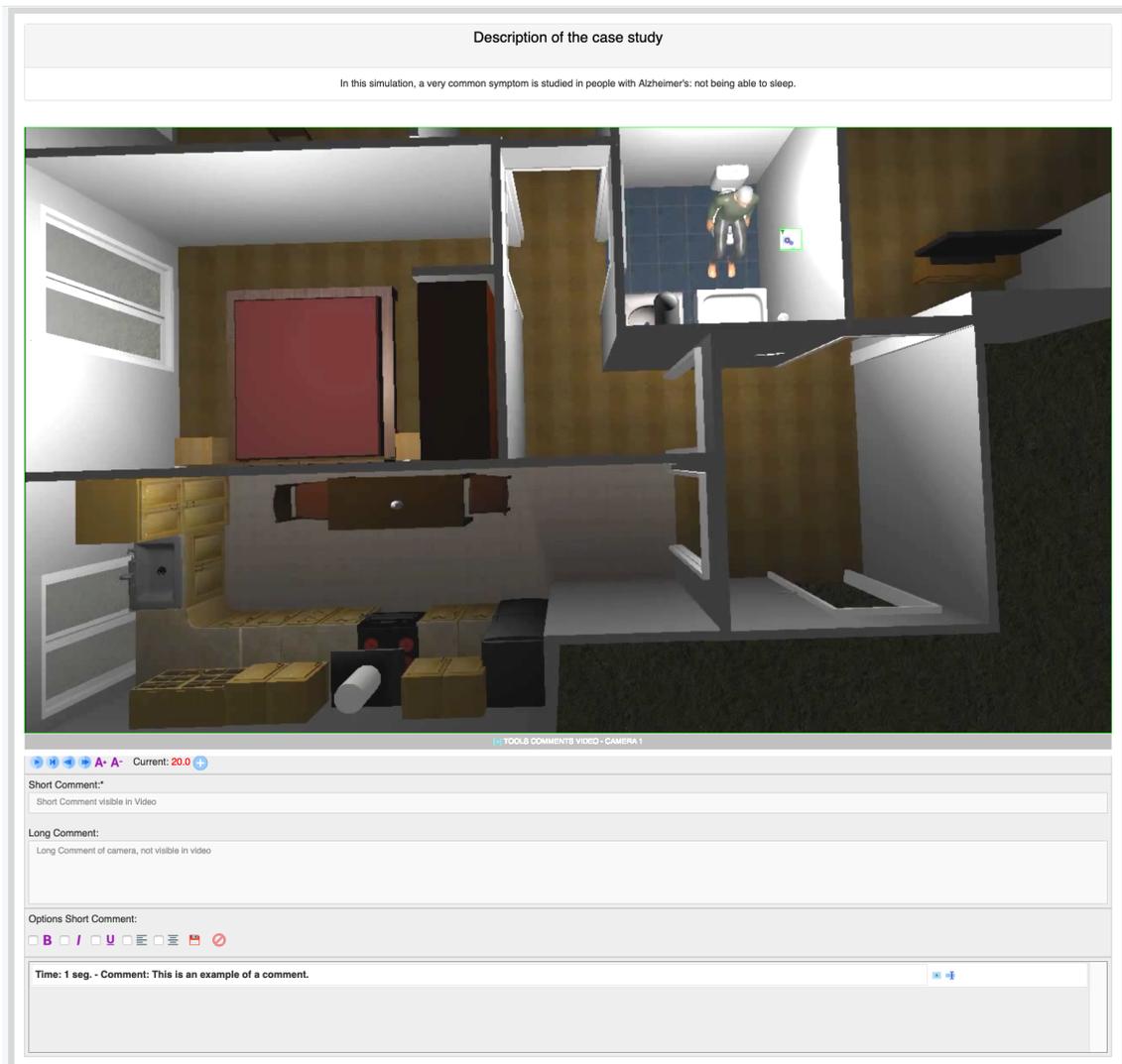


Figura 8.4: Creación de comentarios sobre el vídeo de la simulación.

A la herramienta⁴ se le añadieron algunas funcionalidades como la puntuación de casos de estudio y de la simulación, la gestión de usuarios y un sistema de puntos para la participación, como se observa Figura 8.5. Con estas modificaciones se utilizó en una de las publicaciones que avalan el presente trabajo [CBCGSP18]. En la Sección 2.1.1, se describe la técnica de estudio que ha sido empleada con ayuda de esta plataforma para analizar los casos propuestos.

Figura 8.5: Plataforma Web para el uso de técnicas de trabajo en grupo.

8.2.2.2 Caso de estudio CS001ML

El primer caso de estudio propuesto consiste en simular un síntoma común en personas que padecen deterioro cognitivo: *pérdida de memoria*. En el escenario simulado, se planifica un conjunto de actividades que lleva a cabo el usuario en momentos concretos del día. Dentro de estas actividades, se añaden acciones concretas que el usuario manifiesta ha llevado a cabo cuando padece algún episodio de pérdida de memoria. Estos episodios son de especial interés para el caso de estudio porque podrían significar un problema que limite

⁴Hack4People, acceso público: <http://grasiagroup.fdi.ucm.es:3128/index.php>

la independencia del usuario, sobre todo en momentos cuando debe realizar actividades que requieren mayor concentración como cocinar, tomar la medicación o seguir algún tratamiento o rehabilitación. No se profundiza en aspectos médicos del padecimiento, solo se muestran actividades cotidianas que realiza el usuario y que podrían verse afectadas por episodios esporádicos de pérdida de memoria. Partiendo de estas actividades, se construye la simulación inicial que irá orientada a mostrar un escenario de la vivienda y al usuario como personaje en la simulación. En la Tabla 8.1, se muestra una descripción general del caso de estudio propuesto.

CS001ML: Escenario			
Descripción	Una persona de 82 años que vive sola y no tiene problemas de movilidad, excepto cuando camina porque muestra un ritmo muy suave. A lo largo del día realiza actividades diarias. Estas actividades en momentos específicos tienden a ser repetitivas y están acompañadas por episodios en los que el usuario olvida realizar acciones concretas en la casa.		
Actividades	13	Usuarios (avatares)	1
Duración	10 min.	Ubicación	Vivienda
Problemas de movilidad	No	Áreas	habitación, baño, salón
Cadencia al caminar	Sí	Monitorización	Sí

Tabla 8.1: Descripción general del primer caso de estudio relacionado con la pérdida de memoria.

8.2.2.3 Caso de estudio CS002DI

El segundo caso de estudio propone una simulación con un síntoma común en personas que padecen trastornos conductuales: *la desorientación*. Al igual que en el primer caso, se usan algunas características de este síntoma y se construyen varios escenarios en los que una persona mayor dentro de su vivienda presenta una serie de episodios que podrían indicar que existe desorientación. En la Tabla 8.2, se muestra una descripción general del caso de estudio propuesto.

8.2.2.4 Caso de estudio CS003AR

El tercer caso de estudio que se propone, es modelar en la simulación, escenarios donde la persona mayor realiza actividades físicas concretas dentro de la vivienda. El escenario permite modelar este tipo de actividades en distintas áreas de la vivienda y con diferentes posiciones del cuerpo a la hora de llevarlas a cabo. El objetivo final de este caso, era identificar acciones concretas del usuario que permitan describir su actividad diaria. Este reconocimiento de las acciones era importante que se hiciese con información del entorno y del propio usuario de forma no invasiva y llevando a cabo toda la experimentación necesaria. En la Tabla 8.3, se muestra una descripción general del caso de estudio. Para conseguir la monitorización constante, además de definir los tres escenarios de simulación (normal, problema y solución), se hicieron los experimentos necesarios para diseñar el

sistema con el menor número de sensores posibles y sin demandar la participación del usuario en todo el proceso de desarrollo.

CS002DI: Escenario			
Descripción	La persona de 57 años que vive sola y no tiene problemas de movilidad. El principal problema que muestra este usuario es que tiende a desorientarse en el hogar y no saber qué hacer en los momentos específicos de su día de hoy. Aunque pueda parecer un problema de pérdida de memoria, en estos casos la recuperación de la memoria ocurre después de un tiempo, por lo que el problema, en este caso, es que este episodio ocurre a altas horas de la noche.		
Actividades	14	Usuarios (avatares)	1
Duración	12 min.	Ubicación	Vivienda
Problemas de movilidad	No	Áreas	habitación, baño, salón
Cadencia al caminar	Sí	Monitorización	Sí

Tabla 8.2: Descripción general del segundo caso relacionado con la desorientación.

CS003AR: Escenario típico			
Descripción	Una persona de 61 años que vive sola y no tiene problemas de movilidad, excepto cuando camina que muestra un ritmo suave. Su mayor temor es sufrir caídas dentro del hogar, por lo que quiere algún mecanismo que le permita monitorearlo durante el día y validar con él la posibilidad de detectar y reconocer las actividades. La monitorización se utilizará en un escenario simulado donde el usuario realiza actividades diarias.		
Actividades	10	Usuarios (avatares)	1
Duración	10 min.	Ubicación	Vivienda
Problemas de movilidad	No	Áreas	habitación, baño, salón
Cadencia al caminar	Sí	Monitorización	Sí

Tabla 8.3: Descripción general del tercer caso relacionado con la detección de caídas en el hogar.

8.2.3 Soluciones asistivas propuestas

Se propuso aplicar sobre los casos de estudio definidos CS001ML, CS002DI y CS003AR, una solución asistiva que brindase un servicio específico a personas mayores que padecen algún tipo de problema. Los problemas propuestos, giran en torno a sintomatologías comunes que podrían indicar algún tipo de trastorno conductual y de deterioro cognitivo, los cuales afectan principalmente a personas mayores y son un problema serio cuando estos viven solos en sus hogares. La asistencia que se proponen en estos escenarios, comienza con la definición de dos servicios de monitorización a largo plazo que ayudan a hacer un seguimiento de los problemas propuestos, no en su diagnóstico como tal, sino en la identificación de situaciones que puedan ser un problema para el usuario que vive solo. La descripción de cada escenario se indica a continuación:

- **Solución 1 y 2:** en adelante casos de estudio CS001ML y CS002DI, corresponden a soluciones donde se propone un escenario que represente la vivienda del usuario, en el que él pueda desempeñar distintas actividades a lo largo del día e interactúe con diferentes elementos del entorno. Durante el tiempo que dura la simulación, se propone utilizar un sistema de monitorización a largo plazo que permita reconocer las actividades diarias del usuario e identifique características que sugieran la presencia de alguna anomalía en la rutina diaria. En el caso de la *Solución 1*, se hace uso de la información recogida del entorno y del usuario para describir las actividades diarias, mientras que en la *Solución 2*, sólo se tiene en cuenta la información del usuario y no la del entorno.
- **Solución 3:** en adelante caso de estudio CS003AR, corresponde a una solución que, además de monitorizar al usuario a largo plazo para describir las actividades que realiza, hace un reconocimiento de determinadas actividades físicas conforme la monitorización se lleva a cabo. En esta solución, se hace uso tanto de la información del entorno como la del usuario, especialmente la que se relaciona con su actividad física dentro de la vivienda.

Como se muestra en la Tabla 8.4, dentro de las características de los escenarios, se utilizó una red de sensores que se desplegó virtualmente en la vivienda simulada. Posteriormente, cada sistema utilizó dicha red para llevar a cabo la monitorización del usuario y de su entorno. Las dos primeras soluciones de monitorización solo fueron desplegadas de forma virtual sobre la simulación. En el caso de la tercera solución asistiva, además de desplegarse en la simulación, se utilizó su configuración para construir un prototipo de la solución que fuese igual de útil en un entorno real.

Características de la solución	CS001ML	CS002DI	CS003AR
Monitorización a largo plazo	✓	✓	✓
Uso de sensores ambientales	✓	X	✓
Uso de sensores inerciales	✓	✓	✓
Uso del reconocimiento de actividad	✓	✓	✓
Despliegue de la simulación	✓	✓	✓
Despliegue en un entorno real	X	X	✓

Tabla 8.4: Características de las soluciones propuestas.

8.2.3.1 Alcance y restricciones

Las simulaciones han sido empleadas en la fase de definición del caso de estudio y en la fase de pruebas de la solución. En los dos primeros casos, no se ha llevado a cabo un despliegue de la solución en un entorno real, mientras que en el tercero de ellos, sí se ha definido una fase de pruebas adicional sobre la simulación para validarla como solución real. Estas pruebas han sido previas al despliegue en el entorno real. Como un requisito del framework, es necesario que todas las pruebas ejecutadas persistan a lo largo del

proceso, garantizando de esta forma que por cada simulación validada, los resultados se han almacenado y están siempre disponibles para consultas posteriores. Un resumen de las restricciones en la definición de la solución asistiva en la simulación, se muestra en la Tabla 8.5.

Restricciones				
Condición	Tarea/Actividad	Solución 1	Solución 2	Solución 3
¿Es asistiva la solución?	Definir los escenarios de la solución	✓	✓	✓
¿Se usará la simulación?	Identificar los requisitos	✓	✓	✓
	Modelos de los elementos	✓	✓	✓
¿Hay interacción del usuario?	Pruebas de la simulación	✓	✓	✓
	Pruebas de sistema	X	X	✓

Tabla 8.5: Restricciones en los casos de estudio propuestos.

8.2.3.2 Escenarios simulados

Las soluciones 1 y 2, han sido representadas sólo con dos de los escenarios descritos en la Sección 3.3.1. Un primer “escenario típico” que describió un lapso de tiempo durante el día, en el que un usuario realizó una serie de actividades cotidianas sin ningún tipo de incidente o comportamiento diferente al habitual. En el escenario del problema, se mostró el mismo contexto descrito en el escenario anterior, pero se añadieron acciones alrededor del usuario que indicasen la presencia de situaciones que fuesen un problema para día a día del usuario. El escenario solución no se abordó en estas soluciones porque no interesaba validar el comportamiento de la solución como tal, sino comprobar que la simulación en ambos escenarios, respondía a las especificaciones definidas previamente.

La solución 3 por su parte, además de utilizar los dos escenarios anteriores “típico” y “de problema”, empleó un escenario solución donde se utilizó como propuesta de la solución asistiva, los servicios de monitorización a largo plazo y de reconocimiento de actividad para llevar a cabo el seguimiento del usuario. En la Tabla 8.6 se muestra la participación de los diferentes escenarios en las soluciones propuestas.

Solución	Escenario		
	Típico	Problema	Solución
1	✓	✓	X
2	✓	✓	X
3	✓	✓	✓

Tabla 8.6: Escenarios propuestos en cada solución.

8.2.3.3 Elementos en la simulación

La simulación fue el artefacto clave para recoger la especificación de las soluciones. En cada simulación se incluyen los elementos descritos en la Sección 3.5.1. Uno de los elementos es el personaje animado que representa al usuario o el paciente, y cuya caracterización se basó principalmente en los problemas de movilidad que tenía el usuario. Como se aprecia en la Figura 8.6, la apariencia física del personaje y su parecido con el usuario real, no fueron un objetivo en este trabajo.



Figura 8.6: Personajes propuestos para las simulaciones.

Otro elemento es el escenario modelado, representado en los tres casos por una vivienda con un mobiliario común, establecido mediante la herramienta de desarrollo PHAT. Una característica de los entornos simulados es que se pueden adecuar y hacer accesibles, permitiendo así estudiar casos en los que el usuario demanda modificaciones en el entorno que mejoren su día a día. Los escenarios utilizados se muestran en la Figura 8.7.



Figura 8.7: Escenario propuesto para las simulaciones.

8.2.4 Elementos del dominio modelados

Los elementos de dominio descritos en la Sección 3.4.3, se han tenido en cuenta solo para el caso de estudio CS003AR, puesto que es él único escenario que se quiere usar en un entorno real. Con el ánimo de analizar la participación de cada elemento en la experimentación, se han incluido aquellas características que son indispensables para cumplir con los criterios que rigen la definición de una solución asistiva y que pueden modelarse tanto en el caso de estudio como en la simulación. Al afirmar que se pueden modelar, se da por sentado que dentro de las validaciones, estos criterios también estarán presentes.

8.2.4.1 Tecnología no invasiva

En la Tabla 8.7 se muestran las características relacionadas con la invasividad que han sido utilizadas en el modelado de la simulación y en el desarrollo de la solución. Estas características se enfocan principalmente en la configuración técnica de la solución y en los dispositivos utilizados para su construcción. El sistema construido en el caso de estudio CS003AR, una vez finalizado, mostrará que cumple con las características que se indican en la tabla, siendo la más idónea para llevar a cabo la funcionalidad para la que ha sido desarrollada.

Tipo	Ubicación	Criterio
Sensores	Cuerpo	- Número de sensores en el cuerpo - Combinación de sensores del mismo tipo con diferente configuración - Combinación de sensores de diferente tipo y configuración

Tabla 8.7: Características a modelar en el caso de prueba de CS003AR.

8.2.4.2 Aversión tecnológica

El sistema de monitorización propuesto no necesita de la interacción directa del usuario. Para evitar que este dedique tiempo a aprender el uso del sistema, los dispositivos utilizados funcionan de forma independiente y no cuentan con mecanismos de interacción que obliguen al usuario a manipular la solución. La característica que sí se tiene en cuenta en el desarrollo de la solución, es el tiempo de funcionamiento del sistema con la carga de energía del dispositivo. Este factor, se usa para diseñar la solución final, de tal forma que los elementos utilizados requieran la mínima potencia posible, minimizando así el consumo de energía del sistema. Cuanta menos energía requiera el dispositivo, menor será la intervención del usuario para el funcionamiento de la solución.

8.2.4.3 Privacidad

En este apartado es importante diferenciar el alcance de los dos servicios que se están utilizando para la experimentación. Por un lado se encuentra el sistema de monitorización empleado en la simulación, el cual no utiliza ni almacena información sensible del usuario.

Por otra parte, está el framework que, para ejecutar los distintos de casos prueba sí recoge y almacena información que identifica de forma unívoca a todos los elementos de la simulación, incluido el usuario. Sin embargo, esta información de identificación no está compuesta por datos que pertenezcan al usuario, simplemente se heredan los identificadores de cada elemento incluido en los escenarios de la simulación. Pese a que no es estrictamente necesario proteger esta información, antes de almacenar los identificadores el *Catalogo* aplica una técnica simple de anonimización llamada *Pseudonimización* [TMO09], que consiste en sustituir los identificadores por un código de referencia o “pseudónimo”, de forma que solo sea posible identificar los elementos de la simulación por este nuevo código de referencia. En este caso, el identificador se reemplaza por su valor encriptado, utilizando un algoritmo *Advanced Encryption Standard* (AES) [Wie18]. La clave de cifrado se haya en un fichero plano anexo al que solo tiene acceso el propio Catálogo y el administrados de la instalación.

8.2.4.4 Respuestas del usuario

Estas características no se tienen en cuenta en los casos de prueba propuestos porque la solución en las simulaciones no requiere de la interacción del usuario. La única consideración que se ha tenido en cuenta con respecto a lo declarado en el apartado 8.2.4.2, es que el sistema proporcionará una notificación al usuario indicando la disminución de energía en la batería del dispositivo para que este proceda a alimentarlo. En los casos que la solución no responda, también existen reglas definidas en el motor que identifican la actividad nula de los sensores en la simulación para notificarlo. El mismo sistema de reglas se emplea para identificar valores muy bajos y continuos en la aceleración del usuario (el usuario está muy quieto) durante la simulación.

8.2.4.5 Problemas de movilidad

Los casos de estudio propuestos hacen necesario modelar una serie de situaciones alrededor del paciente que son importantes para observar su comportamiento. En lo que respecta a los problemas de movilidad, se ha incluido pruebas que identifican y advierten de la ocurrencia de patrones muy similares a los que se describen la Tabla 8.8.

Acción	Área específica	Descripción
Simetría del cuerpo	Movimientos	Longitud de los pasos, cadencia, movimiento del torso y el movimiento de los tobillos, las rodillas, las caderas y la pelvis son iguales en ambos lados.
Retropulsión	Parte superior del cuerpo	- Inclinación hacia atrás al comenzar la marcha. - Caer hacia atrás mientras el individuo camina.
Movimientos de los pies	Tamaño de los pasos	- Disminución de la longitud de los escalones.
Parte superior del cuerpo	Avanzar	- Inestabilidad
Brazos	balanceo de brazos	- Incremento. - Decremento.

Tabla 8.8: Caracterización de los cambios anormales en el usuario modelados en el caso de prueba.

La cadencia al caminar por ejemplo, se mide con los sensores posicionados en el torso del usuario y en las manos. En cada caso, hay reglas que el desarrollador en el entorno de pruebas puede parametrizar para intentar identificar alguna particularidad en este tipo de movimiento. Los balanceos del cuerpo hacia adelante o hacia atrás, son otro aspecto que también se pueden incluir en la prueba. El sensor del torso en este caso registra dichos balanceos cuando el usuario cambia de posición o de actividad en la simulación. Por otra parte, el caminar más rápido o más lento es medido con el balanceo de las manos para determinar la aceleración en la marcha. Añadiendo más sensores sobre el usuario, se pueden definir otros parámetros para medir por ejemplo, el número de pasos a lo largo del tiempo. La inestabilidad es más complicada de identificar porque es necesario puntualizar en qué parte del cuerpo se quiere reconocer.

En el framework se hace uso de los las manos para determinar situaciones de temblores como casos de inestabilidad física. De igual forma, el ritmo del balanceo de las manos también puede determinar si hay un aumento o disminución de la misma a lo largo de la simulación. No se busca crear o definir una acción correctiva ante la identificación de estas situaciones en el usuario, simplemente se busca identificar para garantiza que, con ayuda de expertos en el tema, futuros casos de estudio puedan abordar dichas problemáticas para diseñar posibles acciones que minimicen o contrarresten los problemas que éstas generen.

8.2.5 Definición de las actividades.

Los resultados del proceso de definición de requisitos del caso de estudio descrita en la Sección 3.4.6, se utilizaron para definir la lista de actividades que se llevaron a cabo en cada caso de estudio propuesto.

CS001ML: las actividades establecidas para este caso, se muestran en la Tabla 8.9. En total hay 13 actividades descritas que se han tomado de la información obtenida de las entrevistas, después de observar su conveniencia en *Hack4People*. En el *escenario típico* las actividades el usuario las ejecuta sin ningún inconveniente, mientras que en el *escenario problema* se han diferenciado algunas actividad entre los dos escenarios, especialmente la número 12*. Estas variaciones añadidas a la actividad física hace que la simulación de un escenario y del otro sean totalmente diferentes pese a que las actividades en sí sean las mismas.

En el escenario típico, la acción principal del paciente es “*beber agua en la cocina*” y dicha actividad la lleva a cabo sin ningún tipo de inconveniente. En el escenario problema, esta misma actividad varía en una acción específica que no realiza el paciente después de beber el agua: “*cerrar la puerta del refrigerador*”. Como se indicó en la descripción del caso de estudio, en ambos escenarios el usuario realiza las actividad sin ningún tipo de problema físico. La una consideración física que se tiene en cuenta de él, es una cadencia moderada uniforme a la hora de ejecutar las acciones.

En la Figura 8.8, se muestra el modelo de la secuencia de actividad usada en la herramienta SociAALML para generar la simulación del escenario típico. En el modelo se especifican las actividades que ha ejecutado el usuario, los tiempos de espera entre actividad y las condiciones físicas de usuario. También es posible observar en el recuadro rojo, la acción diferenciadora que se lleva a cabo en este escenario.

CS001ML Actividades

Orden	Escenario típico	Escenario problema
-	inicio simulación	inicio simulación
1	el paciente está en el salón	el paciente está en el salón
2	el paciente enciende la TV	el paciente enciende la TV
3	el paciente se sienta en el sofá	el paciente se sienta en el sofá
4	espera 30 segundos en pie	espera 30 segundos en pie
5	el paciente se levanta	el paciente se levanta
6	el paciente va hacia la habitación	el paciente va hacia la habitación
7	se detiene y espera 5 segundos	se detiene y espera 5 segundos
8	el paciente va hacia la cocina	el paciente va hacia la cocina
9	el paciente camina hacia el refrigerador	el paciente camina hacia el refrigerador
10	el paciente abre el refrigerador	el paciente abre el refrigerador
11	el paciente bebe agua durante 30 segundos	el paciente bebe agua durante 30 segundos
12*	el paciente cierra la puerta del refrigerador	el paciente regresa al sofá y olvida cerrar la puerta del refrigerador
13	el paciente espera 5 segundos y va hacia el salón	el paciente espera 5 segundos y va hacia el salón
-	fin simulación	fin simulación

Tabla 8.9: Actividades propuestas para los distintos escenarios del caso de estudio CS001ML.

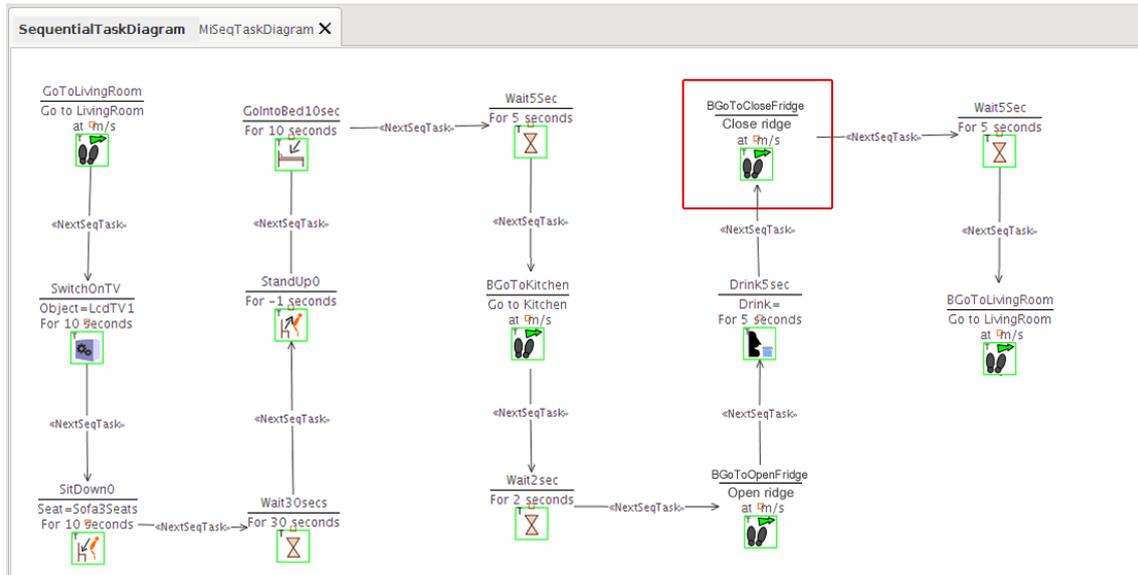


Figura 8.8: Diagrama de secuencia de las actividades en el escenario típico.

La secuencia de actividades del *escenario problema* se muestra en la Figura 8.9. En este modelo se puede observar que la actividad diferenciadora (la número 12* en el primer escenario) no aparece en la secuencia de actividades, por lo cual la misión del sistema es no encontrarla en la simulación generada.

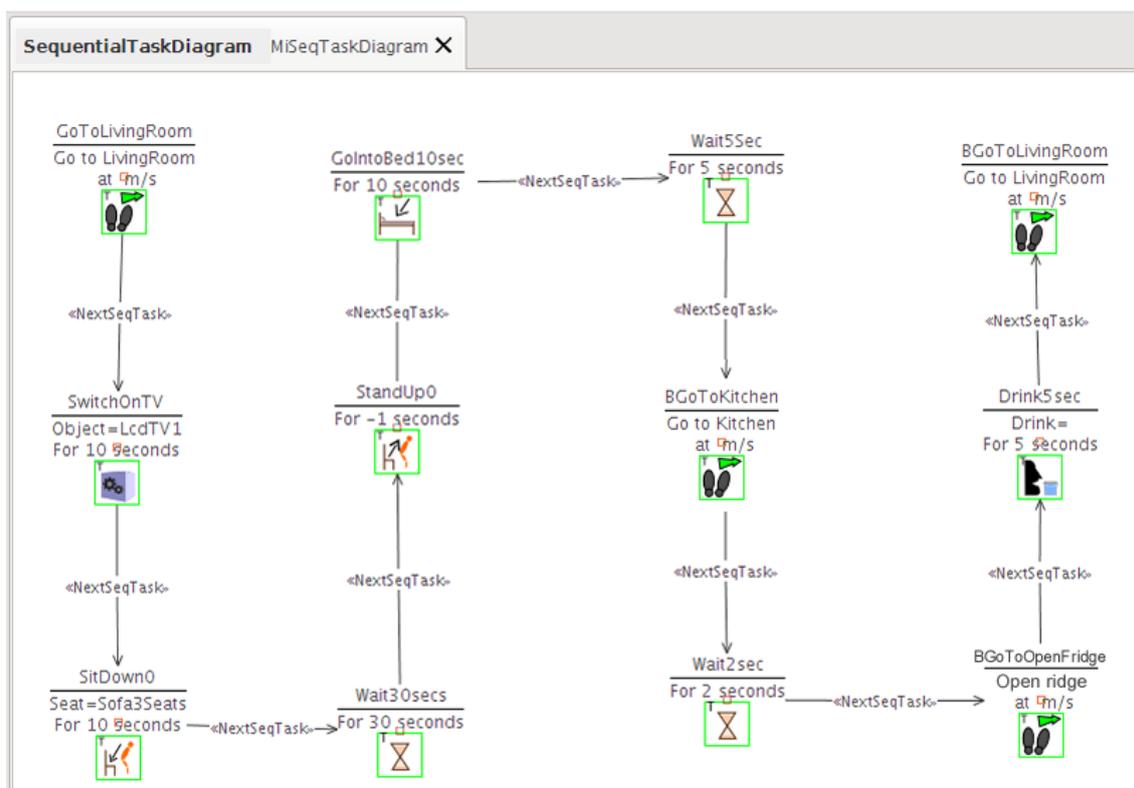


Figura 8.9: Diagrama de secuencia de las actividades en el escenario problema.

CS002DI: las actividades establecidas para este otro caso de estudio, se muestran en la Tabla 8.10. En total hay 14 actividades descritas, de las cuales se han recreado 9 en ambos escenarios. Para este problema, se han diferenciado 5 actividades entre los dos escenarios, concretamente las 2*, 8*, 10*, 12* y 13*. En el escenario típico no hay una acción principal a seguir, la simulación muestra al usuario realizando varias actividades de forma cotidiana sin ningún tipo de incidencia. En el escenario problema se proponen una serie de variaciones en las actividades físicas del usuario con respecto al primero escenario, las cuales podrían ser indicios de síntomas relacionados con algún problema médico que interesarían que el sistema reconociera. Estas alteraciones en las actividades tienen que ver con los siguientes criterios de validación:

- Variación en la intensidad de movimientos periódicos como “temblores”.
- Variación en la duración de las actividades.
- Repetición de las actividad.
- Variación en los tiempos de espera del usuario (tiempos de no actividad).

En la actividad número 2* por ejemplo, se ha añadido en la escena un episodio de leves temblores en la mano izquierda del personaje. En la actividad número 8* los temblores se agudizan y a partir de la actividad 10*, los tiempos de espera del usuario disminuyen. El tiempo de espera se caracterizan porque el usuario permanece de pie y en un estado de reposo. Finalmente, en la actividad número 13*, los temblores ocurren en ambas manos del personaje con una cadencia mayor a la que tenía originalmente en la actividad 2*.

CS002DI Actividades

Orden	Escenario típico	Escenario problema	
		Actividad	Variación
-	inicio simulación	inicio simulación	-
1	el paciente se despierta	el paciente se despierta	-
2*	el paciente tiene episodios de temblores	el paciente tiene episodios de temblores	cadencia moderada
3	el paciente camina hacia el salón	el paciente camina hacia el salón	-
4	enciende la TV	enciende la TV	-
5	camina hacia la cocina	camina hacia la cocina	-
6	el paciente bebe agua	el paciente bebe agua	-
7	regresa al salón	regresa al salón	-
8*	espera 60 segundos en pie	espera 60 segundos en pie	incrementan los temblores
9	camina hacia la cocina	camina hacia la cocina	-
10*	espera 30 segundos en pie	espera 30 segundos en pie	reduce los tiempos de espera
11	camina hacia el salón	camina hacia el salón	-
12*	espera 20 segundos en pie	espera 20 segundos en pie	reduce los tiempos de espera
13*	camina hacia la habitación	camina hacia la habitación	aumentan los temblores
14	el paciente regresa a la habitación	el paciente regresa a la habitación	-
-	fin simulación	fin simulación	-

Tabla 8.10: Actividades propuestas para los escenarios del caso de estudio CS002DI.

La secuencia de actividades utilizada para generar la simulación de este caso se muestra en la Figura 8.10.

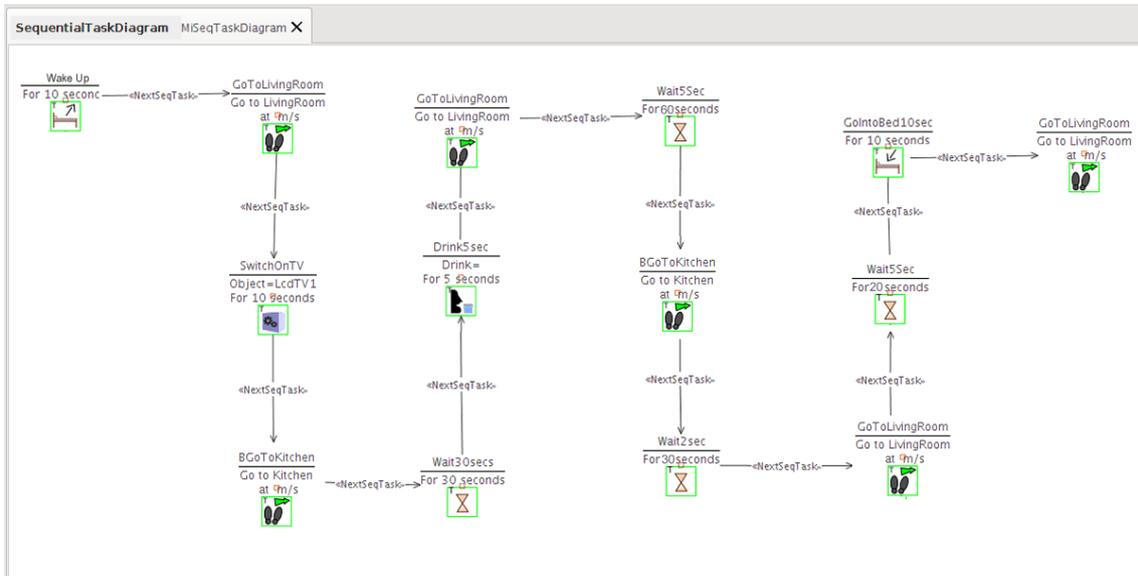


Figura 8.10: Diagrama de secuencia de las actividades en el caso de estudio CS002DI.

Los problemas que se han incluido en este caso de estudio, se han modelado con la herramienta de generación de la simulación. Como se aprecia en la Figura 8.11, dichos problemas se incluyen en el escenario simulación desde tres perspectivas para el usuario:

alteración de los tiempos para ejecutar acciones físicas, patrones de comportamiento que podrían indicar pérdida de memoria (ir y venir sin un fin concreto dentro de la vivienda) y episodios de temblores que van agudizando conforme todos los problemas coinciden en el tiempo.

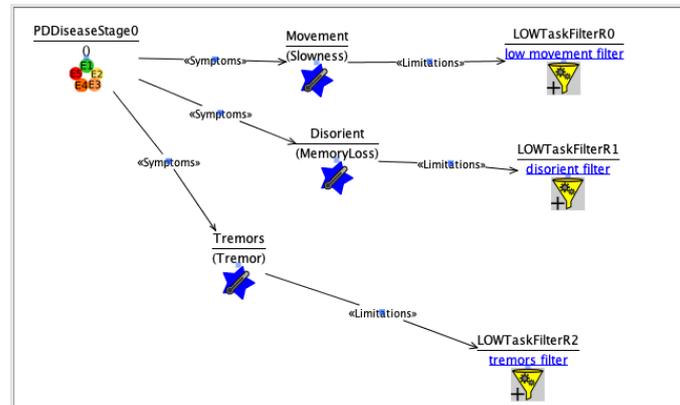


Figura 8.11: Problemas añadidos en la simulación del caso de estudio CS002DI.

CS003AR: las actividades establecidas para este caso, se muestran en la Tabla 8.11. En total son 10 las actividades, de las cuales siete de ellas han sido recreadas en ambos escenarios, mientras que las otras tres varían entre un escenario y otro. Al igual que en el caso de estudio anterior, en el *escenario problema* se proponen variaciones en las actividades físicas del usuario con respecto al primero escenario, las cuales podrían tener relación con síntomas de algún problema médico. Se espera que el sistema en este caso identifique estas variaciones y sea capaz además de reconocer lo más preciso posible ciertas actividades. Para ello, las alteraciones tienen que ver con los siguientes criterios de validación:

- Variación en la intensidad de movimientos periódicos como “temblores”.
- Variación en la duración de las actividades.
- Repetición de las actividad.
- Variación en los tiempos de espera del usuario (tiempos de no actividad).
- Reconocimiento de la actividad.

En la actividad número 2*, el episodio de temblores dentro del escenario normal, se presenta en la mano izquierda del usuario, mientras que en el escenario problema, el temblor se produce en la mano derecha. En la actividad 7* se incluye una variación en el tiempo de espera del usuario y en la actividad 8*, se cambia específicamente la acción que realiza el usuario.

		CS003AR Actividades		
		Escenario problema		
Orden	Escenario típico	Actividad	Variación	Acción a reconocer
0	inicio simulación	inicio simulación	-	-
1	el paciente se despierta	el paciente se despierta	-	cualquier movimiento
2*	el paciente tiene episodios de temblores en la mano derecha	el paciente tiene episodios de temblores en la mano izquierda	cambio de mano	temblor en cada mano
3	el paciente camina hacia el salón	el paciente camina hacia el salón	-	walk
4	enciende la TV	enciende la TV	-	gesto con las manos
5	camina hacia la cocina	camina hacia la cocina	-	walk
6	el paciente bebe agua	el paciente bebe agua	-	drink
7	espera 30 segundos en pie	espera 30 segundos en pie	tiempo de espera	estado de reposo: inmóvil
8	el paciente aplaude	el paciente llama la atención	cambio de actividad	actividad diferente
9*	camina hacia el salón	camina hacia el salón	-	-
-	fin simulación	fin simulación	-	-

Tabla 8.11: Actividades propuestas para los escenarios del caso de estudio CS003AR.

8.3 Creación de los casos de prueba

La primera fase de la experimentación propuesta permitió definir los casos de estudio y generar a partir de ellos las simulaciones. La metodología propuesta en el Capítulo 3 fue necesaria para identificar los diferentes elementos que se requerían para construir dichos casos. En la segunda fase de la experimentación, lo que se hizo inicialmente fue identificar los aspectos del caso de estudio que se trasladaron a la simulación, siendo el primero de ellos la configuración de los sensores virtualizados (ver Sección 3.5.1). Posteriormente, se definieron las validaciones a aplicar sobre la simulación en cada caso de estudio, usando como referencia los criterios de validación que se necesitan para crear las pruebas unitarias, descritos en la Sección 4.2.

8.3.1 Configuración de los sensores

En la Tabla 8.12, se puede observar las características de los sensores utilizados en cada caso de estudio. Asimismo, se muestra la tipología de los sensores, su posicionamiento en el entorno, la frecuencia de muestreo que emplean y las ventanas de tiempo y de datos con las que son procesados en la simulación.

Inicialmente, el número de sensores fue relevante para evitar invasividad tecnológica. Teniendo en cuenta este aspecto, sobre el entorno se han posicionado 2 sensores ambientales y 4 actuadores para interactuar con el usuario. Sobre el usuario los sensores han sido 3 y se han ubicado en el torso y en las muñecas. La posición ha sido importante para realizar las pruebas sobre aquellas áreas de la vivienda donde se llevaron a cabo las actividades. En el caso del usuario, la posición ha sido por defecto asumida para reconocer actividades relacionadas con la marcha del usuario y la posición del cuerpo. El siguiente aspecto que queda por determinar es la distribución que tienen dichos sensores en el escenario simulado.

Caso de estudio	Número de sensores Identificadores	Tipo de sensor	Posición	Frecuencia	Ventana de tiempo w_t	Ventana de datos w_d
CS001ML	(7) $s_1, s_2, s_3; a_1, a_2$ sa_1, sa_2	- inercial, actuador - ambiente	usuario y entorno (salón, cocina)	10Hz - 40Hz	- 2s - 1s-2s	- $w_t/3$ - w_t
CS002DI	(7) $s_1, s_2, s_3; a_1, a_2$ sa_1, sa_2	- inercial, actuador - ambiente	usuario y entorno (salón y cocina)	10Hz - 40Hz	- 2s - 1s-2s	- $w_t/3$ - w_t
CS003AR	(1-3) s_1, s_2, s_3	- inercial	usuario	20Hz - 40Hz	- 2s	- w_t

Tabla 8.12: Configuración de los sensores en la simulación de cada caso de uso.

En la Figura 8.12, se puede observa cómo han sido distribuidos los sensores sobre el usuario y en el entorno. En cada muñeca y sobre el pecho del usuario se fijan tres sensores inerciales de aceleración. En el entorno, se posicionan estratégicamente, dos sensores de luz y dos de temperatura, así como dos actuadores que por defecto irán instalados en la puerta del refrigerador y en la televisión.



Figura 8.12: Posición de los sensores en la simulación de los casos de estudio.

Para el reconocimiento de otras actividades diferentes a las propuestas, será necesario que la posición vuelva a ser revisada, esto dependerá del caso de prueba que se quiera aplicar. En estos casos de prueba concretamente, la posición de los sensores elegida en el usuario ha permitido reconocer las actividades propuestas. En el caso de los sensores del entorno, el posicionamiento indicado anteriormente, es el mismo en los tres escenarios.

8.3.2 Definir las validaciones

La definición del caso de estudio y la lista de actividades fueron utilizadas para definir los casos de prueba y las validaciones correspondientes en cada escenario. Según los escenarios simulados, los casos de pruebas propuestos se han enfocado en tres aspectos del caso de estudio:

- Validar que las actividades definidas se llevan a cabo en cada escenario simulado. En este caso importaba comprobar que el número de actividades era el correcto y que las pruebas unitarias utilizadas eran capaces de diferenciar una actividad de la otra aunque no identificaran concretamente dicha actividad.
- Validar que cada actividad definida cumple el criterio de éxito o fracaso establecido en la lista de actividades. En este otro caso, se comprueba que lo que ocurre en la simulación es lo que se espera de ella en cada escenario. De igual forma, se comprueba que los resultados entre los distintos escenarios realmente los diferencia uno del otro.
- Validar que la simulación, en lo que respecta a configuración y parametrización de los sensores, cumplía con los valores establecidos necesarios para validar las actividades. Esto fue particularmente importante cuando las pruebas unitarias necesitaban acceder a datos de sensores inerciales para hacer reconocimiento de actividad.

En cada actividad se definieron pruebas principales y secundarias en función de su complejidad. En las actividades complejas propuestas, se ha utilizado un número de pruebas suficientes (una prueba de configuración, una prueba principal y una prueba secundaria) para identificar la actividad, utilizando como criterio de validación la salida esperada. En las actividades simples, no ha hecho falta utilizar pruebas secundarias por lo que las pruebas unitarias propuestas junto con la prueba de configuración, han sido suficientes para determinar el éxito o fracaso de la actividad. En los casos donde las pruebas unitarias no han sido capaces de identificar concretamente la actividad, sí ha sido posible observar la transición de una actividad anterior conocida a una actividad nueva no conocida. Es decir, la prueba ha comprobado que se ha llevado a cabo una actividad nueva diferente a la anterior pero no ha sido capaz de indicar de cuál se trataba.

8.3.2.1 Validaciones del caso CS001ML

En cada escenario hay un número de actividades con sus respectivas salidas esperadas. Como se puede ver en la Tabla 8.13, todas las salidas necesitan que el sistema de monitorización implementado en la simulación, suministre la información correspondiente tanto del entorno como del usuario para poder identificarlas. En este caso concreto, no se propone que haya un reconocimiento específico de la actividad del usuario, simplemente se quiere identificar algún patrón “predecible” (“*up*”, “*down*”, “*left*”, “*right*”) mediante el motor de reglas. De igual forma, se espera que se valide el número de actividades ejecutadas y posteriormente que se puedan comparar los resultados de ambos escenarios

para identificar las diferencias establecidas en el diseño del caso de estudio (ver actividades 2, 3, 5, 10 y 12).

CS001ML Actividades

Orden	Escenario típico	Salida esperada	Escenario problema	Salida esperada
0	inicio simulación	-	inicio simulación	-
1	el paciente está en el salón	-	el paciente está en el salón	-
2*	enciende la TV	cualquier gesto "up", "down", "left", "right"	enciende la TV	cualquier gesto "up", "down", "left", "right"
3*	el paciente se sienta en el sofá	cualquier gesto "up", "down", "left", "right"	el paciente se sienta en el sofá	cualquier gesto "up", "down", "left", "right"
4	espera 30 segundos	-	espera 30 segundos	-
5*	el paciente se levanta	cualquier gesto "up", "down", "left", "right"	el paciente se levanta	cualquier gesto "up", "down", "left", "right"
6	el paciente va hacia el salón	-	el paciente va hacia el salón	-
7	se detiene y espera 5 segundos	-	se detiene y espera 5 segundos	-
8	el paciente va hacia la cocina	-	el paciente va hacia la cocina	-
9	camina hacia el refrigerador	-	camina hacia el refrigerador	-
10*	el paciente abre el refrigerador	cualquier gesto "up", "down", "left", "right"	el paciente abre el refrigerador	cualquier gesto "up", "down", "left", "right"
11	bebe agua durante 30 segundos	-	bebe agua durante 30 segundos	-
12*	cierra la puerta del refrigerador	cualquier gesto "up", "down", "left", "right"	el paciente regresa al sofá y olvida cerrar la puerta del refrigerador	cualquier gesto "up", "down", "left", "right"
13	espera 5 segundos y va hacia el salón	-	espera 5 segundos y va hacia el salón	-
14	fin simulación	-	fin simulación	-

Tabla 8.13: Lista de actividades y salidas esperadas en el caso de estudio CS001ML.

En el escenario típico, las actividades críticas son la 10 y 12 porque indican el "inicio" y el "fin" de lo que sería una actividad del usuario, es el escenario equivale a abrir y cerrar una puerta del refrigerador. En ambas actividades, se debería poder reconocer patrones contrapuestos y relacionados entre sí, por ejemplo: "up" y "down" o "left" y "right". Alguna otra combinación también podría ser posible, pero en ese caso, una prueba unitaria adicional debería validar que existe o no dicha relación contrapuesta. Con respecto al escenario problema, la prueba consiste en detectar que falta una de estas actividades, concretamente la del "fin" (actividad 12). Mientras que en el escenario normal se detectan los dos patrones, en este segundo escenario el motor debe arrojar que hace falta uno de ellos.

Las otras salidas esperadas solo demandan que el motor reconozca que se ha producido una actividad física y no precisa determinar cuál. En todo caso, en los resultados de las pruebas se podrá observar si el motor ha sido capaz de asociar cada actividad física a las que tiene predefinida en el sistema de reglas.

8.3.2.2 Validaciones del caso CS002DI

En el segundo caso, las validaciones van orientadas a usar el tiempo y la intensidad como dos factores en la evaluación de la actividad. Como se puede ver en la Tabla 8.14, Las actividades 2, 8, 10, 12 y 13 son consideradas críticas porque están relacionadas con

un problema que tiene el usuario en esos momentos concretos que ocurren en el escenario. Aunque la causa no se especifique con claridad, lo que debe ser capaz la solución es de reconocer las variaciones de las alteraciones físicas que padezca el usuario a lo largo del día. Para ello, los criterios de validación de las actividades son los indicadores a emplear en cada validación que lo requiera. En el caso de los temblores, a determinada edad suele ser considerados comunes según el padecimiento del paciente, pero una alteración en el ritmo de estos podría ser evidencia de algún problema más.

CS002DI Actividades					
		Escenario problema			
Orden	Escenario típico	Actividad	Variación	Elemento	Criterio
-	inicio simulación	inicio simulación	-	-	-
1	el paciente se despierta	el paciente se despierta	-	-	-
2*	el paciente tiene episodios de temblores	el paciente tiene episodios de temblores	cadencia moderada	sensor: usuario	datos simples
3	el paciente camina hacia el salón	el paciente camina hacia el salón	-	-	-
4	enciende la TV	enciende la TV	-	-	-
5	camina hacia la cocina	camina hacia la cocina	-	-	-
6	el paciente bebe agua	el paciente bebe agua	-	-	-
7	regresa al salón	regresa al salón	-	-	-
8*	espera 60 segundos en pie	espera 60 segundos en pie	aumentan los temblores	sensor: usuario	datos simples
9	camina hacia la cocina	camina hacia la cocina	-	-	-
10*	espera 30 segundos en pie	espera 30 segundos en pie	reduce tiempo de espera	sensor: usuario	datos simples
11	camina hacia el salón	camina hacia el salón	-	-	-
12*	espera 20 segundos en pie	espera 20 segundos en pie	reduce tiempo de espera	sensor: usuario	datos simples
13*	camina hacia la habitación	camina hacia la habitación	aumentan los temblores	sensor: usuario	datos simples
14	regresa al salón	regresa al salón	-	-	-
-	fin simulación	fin simulación	-	-	-

Tabla 8.14: Lista de actividades y salidas esperadas en el caso de estudio CS002DI.

En este caso se han combinado algunos criterios de validación de las actividades con distintas situaciones que afectan al comportamiento del usuario, estas son:

- El usuario va de un lugar a otro en instantes cortos de tiempo (repetir actividades con cierta frecuencia).
- Los tiempos de espera entre estas actividades es inferior a una ventana de tiempo determinada, en este caso se ha usado 30 segundos (variación en la duración de la actividad).
- Hay episodios de temblores en el usuario que aumentan conforme los tiempos de espera disminuyen (repetir actividades con una frecuencia alta).

Todos estos factores están inmersos en momentos específicos de la simulación, por lo que los casos de prueba deben ser capaces de detectarlos en el mismo orden en que ocurren.

La validación del caso de estudio consiste precisamente en utilizar la simulación y validar que estas acciones ocurren tal y como se esperan (la simulación como especificación), de forma que dicha simulación muestre correctamente las irregularidades que ocurren con el comportamiento del usuario (la simulación como solución), incluidos los aspectos físicos que son medidos por los sensores.

8.3.2.3 Validación del caso CS003AR

En este caso de estudio, además de validar los aspectos mencionados en la Sección 8.3.2, se añade como una validación más específica el reconocimiento de la actividad humana dentro de las simulaciones. Como se observa en la Tabla 8.15, el sistema de reconocimiento empleado en el framework, debe ser capaz de reconocer e identificar cada una de las acciones esperadas en las actividades 3, 5, 6 y 8, concretamente “walk”, “drink”, “clap” y “wave attention”. En cuanto al reconocimiento de patrones comunes, el motor deben reconocer la presencia de temblores (actividad número 2) y gestos simples (actividad número 4).

CS003AR Actividades				
Orden	Escenario típico	Escenario problema		
		Actividad	Criterio	Salida esperada
0	inicio simulación	inicio simulación	-	-
1	el paciente se despierta	el paciente se despierta	-	cualquier movimiento
2*	el paciente tiene episodios de temblores en la mano izquierda	el paciente tiene episodios de temblores en la mano derecha	cambio de mano	temblor en cada mano
3	el paciente camina hacia el salón	el paciente camina hacia el salón	-	walk
4	enciende la TV	enciende la TV	-	algún gesto: <i>up, down, left, right</i>
5	camina hacia la cocina	camina hacia la cocina	-	walk
6	el paciente bebe agua	el paciente bebe agua	-	drink
7	espera 30 segundos en pie	espera 30 segundos en pie	tiempo de espera	estado de reposo: inmóvil
8	el paciente aplaude	el paciente llama la atención	cambio actividad	actividades diferentes
9*	camina hacia el salón	camina hacia el salón	-	-
-	fin simulación	fin simulación	-	-

Tabla 8.15: Lista de actividades y salidas esperadas en el caso de estudio CS003AR.

8.3.3 Identificar los parámetros de entrada

A lo largo de los diferentes pasos descritos en esta sección, se han ido definiendo las características que definen tanto al caso de estudio como los elementos que componen la simulación. Estas mismas características son las usadas en la especificación que define la simulación, por ello se deben tener en cuenta a la hora de establecer cualquier plan de pruebas. El framework necesita de esta información para poder tratar los datos de forma interna y aplicar sobre ellos las reglas correspondientes que validen su

comportamiento. Como resumen de todas estas consideraciones, en las Tablas 8.16, 8.17 y 8.18, se muestran aquellas características que han sido usadas al definir las validaciones. Estos mismos valores, son usados como elementos de entrada en los diferentes métodos predefinidos que existen en el framework y que soportan la creación de las pruebas unitarias correspondientes.

Caso de estudio CS001ML			
Actividades	13	Monitorización a largo plazo	✓
Uso de reglas	✓ 5 actividades	HAR	x -
Duración	11 min.	Problemas de movilidad:	x
Tratamiento de datos inerciales			
Ventana de tiempo w_t :	2 seg.	Flujos de eventos:	3 máx. s_1, s_2, s_3
Ventana de datos w_d :	$w_t/3$	Frecuencia (Hz):	20 - 40
Tratamiento de datos ambientales.			
Ventana de tiempo w_t :	1 seg.	Flujos de eventos:	3, $f_{s_1}, f_{s_2}; f_{a_1}$
Ventana de datos w_d :	w_t	Frecuencia (Hz):	10 - 20

Tabla 8.16: Parámetros necesarios para validar el caso de prueba en CS001ML.

Caso de estudio CS002DI			
Actividades	14	Monitorización a largo plazo	✓
Uso de Reglas	✓ 5 actividades	HAR	x -
Duración	12 min.	Problemas de movilidad:	✓
Tratamiento de datos inerciales			
Ventana de tiempo w_t :	2 seg.	Flujos de eventos:	3 máx. $s_1 - s_3$
Ventana de datos w_d :	$w_t/3$	Frecuencia (Hz):	20 - 40
Tratamiento de datos ambientales			
Ventana de tiempo w_t :	1 seg.	Flujos de eventos:	6, $f_{s_1} - f_{s_4}; f_{a_1}, f_{a_2}$
Ventana de datos w_d :	w_t	Frecuencia (Hz):	10 - 20

Tabla 8.17: Parámetros necesarios para validar el caso de prueba en CS002DI.

Caso de estudio CS003AR			
Actividades	10	Monitorización a largo plazo	✓
Uso de reglas	✓ 3 actividades	HAR	✓ 5 actividades
Duración	25 min.	Problemas de movilidad:	✓
Tratamiento de datos inerciales			
Ventana de tiempo w_t :	2 seg.	Flujos de eventos:	3 máx. $s_1 - s_3$
Ventana de datos w_d :	$w_t/3$	Frecuencia (Hz):	20 - 40
Tratamiento de datos ambientales			
Ventana de tiempo w_t :	1 seg.	Flujos de eventos:	6, $f_{s_1} - f_{s_4}; f_{a_1}, f_{a_2}$
Ventana de datos w_d :	w_t	Frecuencia (Hz):	10 - 20

Tabla 8.18: Parámetros necesarios para validar el caso de prueba en CS003AR.

8.3.4 Definir los criterios de éxito y fracaso

Para complementar la tarea “Identificar los requisitos” descrita en la Sección 3.4.2, se llevaron a cabo otras tareas que permitieron establecer de qué forma dichos requisitos serían evaluados posteriormente. Los primeros criterios de validación fueron asociados a la identificación de las situaciones recurrentes vistas en la Sección 3.4.3. En aspectos invasivos de la tecnología, las tareas “encontrar riesgos” y “crear plan de mitigación” han sido útiles para incluir estas consideraciones en los casos de uso. En relación con la aversión tecnológica, “encontrar posibles riesgos” y luego “describir la aversión tecnológica” en dichos riesgos, se han usado para añadir aspectos validables que puedan estar relacionados con la vivienda del paciente, en función de su perfil (edad, cercanía a la tecnología, entre otros).

La privacidad fue otro factor a tener en cuenta a la hora de incluir criterios de validación tanto en la solución como en el framework propuesto. La tarea “revisar la tecnología” se aplicó junto con los criterios que se establecen en las pruebas unitarias, como se muestra en la Tabla 4.1. En lo que respecta a las respuestas del usuario, todas las tareas propuestas resumidas en la Tabla 3.5, se añadieron como reglas que pueden incluirse en los casos de prueba para aplicarlas como criterios de éxito o fracaso en la simulación. Un ejemplo práctico en este caso es la interacción del usuario con un actuador: puertas, grifos, entre otros. Si se espera que el usuario responda ante su uso, el escenario propuesto permite reconocer la presencia de dicha respuesta. Lo que no se ha establecido es el plan de actuación en el caso de que ocurran. Por otra parte, en relación a los problemas de movilidad, se ha aplicado la misma metodología. La identificación de estos problemas en el escenario es posible y puede usarse como otro criterio más de validación, por ejemplo, que ocurra un balanceo del tronco al caminar es un fracaso y que no ocurran es un éxito, o viceversa.

Conforme se han identificado los requisitos del caso de estudio, en las listas de actividades de cada caso de estudio, de la misma forma a cada actividad se la ha ido asociando la salida esperada correspondiente. Estas salidas esperadas son parte del criterio de éxito o fracaso de cada actividad en el caso de estudio. Otro criterio que se añade al caso de pruebas es la validación de que el escenario cumpla con todas las salidas esperadas o al menos, un porcentaje de ellas. Un criterio general que se aplica en esta experimentación es que al menos el 50 % de las salidas esperadas se deben cumplir en el caso de prueba o sobre el escenario simulado, para considerar que el caso de prueba es exitoso. El criterio elegido en este caso parte de la relación CA/TA , donde CA son las actividades identificadas correctamente y TA el total de actividades validadas. En el caso de la experimentación, este mismo será el criterio utilizado para validar globalmente el caso de estudio.

8.3.5 Crear las pruebas unitarias

Una vez se identificaron los elementos de la simulación que son validables de cara a las pruebas unitarias, resumidos en la Tabla 4.1, y se establecieron los criterios de validación que se aplicarían tanto al entorno como al usuario, sobre la lista de actividades de cada caso de estudio se especificaron las pruebas unitarias que serían necesarias para validar

dicha actividades. En cada caso de prueba se han establecido las pruebas unitarias que se aplicaron sobre las actividades ejecutadas en la simulación. Inicialmente se identifican los elementos del entorno simulado sobre el que aplica la prueba y posteriormente, las pruebas unitarias utilizadas.

8.3.5.1 Pruebas unitarias de CS001ML

Una vez se tiene claro qué elemento de la simulación lleva a cabo la actividad, lo siguiente es definir el criterio que se quiere aplicar para validar. En la Tabla 8.19, se observa el elemento de la simulación al que pertenece la actividad y el criterio que se usará para validarlo. En el caso de las actividades que tienen criterio de validación, aunque el usuario es quien realiza la actividad, realmente la forma de llegar a su descripción es mediante el dispositivo que está monitorizando al usuario, en este caso, los sensores.

Orden	Escenario típico	Elemento	Criterio	Prueba
2	enciende la TV	sensor: usuario, entorno	entorno: datos simples	getSimpleData checkMovement
3	el paciente se sienta en el sofá	sensor: usuario	datos simples	getSimpleData checkPosition checkMovement
5	el paciente se levanta	sensor: usuario	datos simples	getSimpleData checkPosition checkMovement
10	el paciente abre el refrigerador	sensor: usuario	datos simples	getSimpleData checkPosition checkMovement
12	el paciente cierra la puerta del refrigerador	sensor: usuario	datos simples	getSimpleData checkPosition checkMovement

Tabla 8.19: Pruebas unitarias utilizadas en el caso de estudio CS001ML.

Una vez se ha identificado el criterio a aplicar en la prueba, lo siguiente es seleccionar aquellos componentes del framework que permiten aplicar dicho criterio sobre un elemento concreto de la simulación. Los componentes disponen de interfaces que han sido implementadas previamente, definiendo métodos específicos que se usan en las pruebas unitarias por el desarrollador. No hay una relación uno a uno entre criterio y método, el desarrollador tiene la libertad de hacer combinaciones y/o adaptaciones de los métodos para elaborar un caso de prueba más acordes a las necesidades de la simulación.

Siguiendo esa libertad entre método criterio, en cada actividad a validar se ha indicado qué método podría utilizarse para construir la prueba correspondiente. Las actividades en el primer caso de estudio prácticamente pueden validarse obteniendo el flujo de eventos “*getSimpleData*” del sensor específico e identificando un patrón sobre él “*checkMovement*”. De la misma forma, comprobar la posición del sensor “*checkPosition*” es igualmente útil cuando hay un movimiento por parte del usuario y esto influye en la actividad. Al obtener el flujo de eventos con “*getSimpleData*”, se hacen tres pruebas implícitas de simulación: conexión al flujo, comprobación de parámetros de configuración y tratamiento de los datos.

8.3.5.2 Caso de prueba CS002DI

Como se puede ver en la Tabla 8.20, todos los criterios tanto en primer caso de estudio como en el segundo, usan como base el tipo de sensor y el comportamiento que tienen los datos de este. Esto último es lo que permite “limitar” el uso de reglas en el motor. Es decir, si se analiza el comportamiento simple de los datos, los criterios disponibles para una validación son el estado del sensor, su posición y un cambio o variación en la lectura del sensor. Si el caso fuese analizar su comportamiento complejo, se podría aplicar directamente el sistema de reconocimiento de actividad tal y como ocurre con el tercer caso de estudio.

Orden	Escenario típico	Elemento	Criterio	Prueba
2*	el paciente tiene episodios de temblores	sensor: usuario	datos simples	getSimpleData checkMovement
8*	espera 60 segundos en pie	sensor: usuario	datos simples	getSimpleData checkMovement checkStop
10*	espera 30 segundos en pie	sensor: usuario	datos simples	getSimpleData checkMovement checkStop
12*	espera 20 segundos en pie	sensor: usuario	datos simples	getSimpleData checkMovement checkStop
13*	camina hacia la habitación	sensor: usuario	datos simples	getSimpleData checkMovement checkStop

Tabla 8.20: Pruebas unitarias utilizadas en el caso de estudio CS002DI.

Determinar el número de pruebas unitarias va en función de lo compleja que sea la actividad y del número de sensores utilizados para monitorizar. Cuantas más pruebas se definan para una actividad, mayor será la probabilidad de precisión en la prueba. Esta es la razón por la que en algunos casos, es posible usar más de una prueba unitaria que valide una acción concreta. Este criterio se ha usado tanto en el primer caso de estudio como en el segundo.

8.3.5.3 Caso de prueba CS003AR

En el tercer caso de estudio como el objetivo es reconocer actividades, necesariamente se debe definir como criterio de validación el comportamiento complejo de los datos. En este caso, el motor sabe que debe enviar este tipo de flujos de eventos al sistema de reconocimiento de actividad implementado en el framework. Como se observa en la Tabla 8.21, algunas actividades podrían parecer simples por el hecho de asumir determinados comportamientos en el usuario.

La actividad 7, por ejemplo, describe un comportamiento tan simple en el usuario como es el estado en reposo o quieto. Se podría pensar que con no detectar actividad en los sensores sería suficiente pero es necesario pensar que aunque no haya movimiento

Orden	Escenario típico	Elemento	Criterio	Prueba
1	el paciente se despierta	sensor: usuario, entorno	datos simples	getSimpleData checkPosition checkValue
2*	el paciente tiene episodios de temblores en la mano izquierda	sensor: usuario	datos simples	getSimpleData getMovement
3	el paciente camina hacia el salón	sensors: usuario	datos complejos	checkActivity
4	enciende la TV	sensor: usuario, entorno	datos simples	checkMovement
5	camina hacia la cocina	sensor: usuario	datos complejos	checkActivity
6	el paciente bebe agua	sensor: usuario	datos complejos	checkActivity
7	espera 60 segundos en pie	sensor: usuario	datos complejos	checkStop checkMovement
8	el paciente aplaude	sensors: usuario	datos complejos	checkActivity

Tabla 8.21: Pruebas unitarias utilizadas en el caso de estudio CS003AR.

o cambio de posición del cuerpo, las manos sí podrían estar en movimiento. Lo mismo ocurre con los problemas de movilidad. Un balanceo del cuerpo o una mala postura se detectará con reglas especiales que busquen ese patrón, sin que esto signifique que el usuario no podrá hacer otra actividad distinta en ese momento.

El reconocimiento de la actividad humana es una tarea compleja dentro de una solución ambiental, de ahí la necesidad de emplear técnicas especializadas ante este tipo de necesidades. El sistema en estos casos ofrece una serie de métodos que ayudan a canalizar los flujos de eventos hacia los sistemas de reconocimientos implementados. En el tercer caso de estudio, métodos como “*checkActivity*” y “*checkActivities*”, mediante el nombre de una regla concreta, son capaces de poner en marcha el motor para que reconozca una acción a partir de un flujo de datos. En el caso de estudio se utiliza este método siempre que sea necesario emplear el reconocimiento de la actividad. Lo importante en este caso es saber qué identificador tiene la actividad que se quiere reconocer.

8.3.5.4 Utilizar pruebas unitarias secundarias

Las pruebas secundarias o de apoyo, permiten corroborar la información o el resultado de las pruebas principales. En algunos casos, no es suficiente con crear una prueba que verifique si un gesto se lleva a cabo o no. Debido a los diferentes matices que puede tener el movimiento o incluso el mismo sensor, las pruebas secundarias le darán al desarrollador del caso de prueba, la oportunidad de validar que no hayan por ejemplo, anomalía en el comportamiento de los eventos o que la predicción de una gesto se contradiga con la naturaleza del gesto.

Para los casos propuestos, las pruebas secundarias se han utilizado para identificar factores relacionados con la invasividad tecnológica y con problemas de movilidad del usuario, tal y como se describe en las secciones 8.2.4.1 y 8.2.4.5 respectivamente. Para identificar estos factores, se han definido una serie de reglas específicas que buscan detectar patrones relacionados con estas situaciones, sugiriendo de esa forma la presencia de una posible anomalía en el comportamiento del entorno. En la Tabla 8.22, se muestran las

principales reglas que existen para identificar patrones relacionados con los factores antes mencionados.

Specific area	Descriptor de regla	Tipo de regla	Descripción
Simetría del cuerpo (movimientos)	@lengthStep	regla compleja	tamaño de los pasos
	@frontalRollingTorso	regla simple	balanceo frontal del torso
	@lateralSwingTorso	regla simple	balanceo lateral del torso
Retropulsión (parte superior del cuerpo)	@backwardWalkRetro	regla compleja	inclinación hacia atrás al comenzar la marcha
	@forwardWalkRetro	regla compleja	inclinación hacia adelante al comenzar la marcha
Movimientos de los pies	@footMovements	regla compleja	Disminución de la longitud de los escalones.
Inestabilidad (balanceo de brazos)	@increaseSwingingArms	regla compleja	Incremento del balanceo de los brazos.
	@decreaseSwingingArms	regla compleja	decremento en el balanceo de brazos
Sensor	@numberSensors	regla simple	número de sensores activos del mismo tipo
	@numberDiffSensors	regla simple	número de sensores activos de diferente tipo

Tabla 8.22: Reglas específicas para identificar problemas generales en elementos de la simulación.

Cuando se presenten situaciones en las que la prueba principal y la secundaria se contradicen entre sí, se ha establecido una prueba *@afterTest* que muestre la anomalía en la simulación, siempre que provenga de la animación del personaje (reconocimiento de actividad) o bien del propio sensor (inconsistencia del método *“getSimpleData”*). Como no es posible determinar con certeza qué ocurre en estos casos, esta prueba mostrará qué intervalo de tiempo exacto se debe revisar en la simulación para comprobar lo que ocurrió.

8.4 Resultado de los casos de prueba

La ejecución de los casos de prueba sobre el entorno de la simulación, ha supuesto que todas las pruebas unitarias establecidas se llevaran a cabo siguiendo el orden de las actividades definidas en cada escenario. Por cada actividad ha sido posible observar el resultado de las distintas validaciones realizadas tanto a la simulación como especificación de la solución asistiva, como a las características que se aplicaron en ella para definir los escenarios. En el tercer caso de prueba la experimentación ha ido un paso más allá al validar no solo la simulación y sus características, sino también a la herramienta que se desarrolló en función de las especificaciones de la simulación.

El proceso de validación se ha llevado a cabo en parte gracias al sistema de monitorización empleado en la simulación. Dicho sistema ha contado con la especificación técnica que se ha establecido gracias a la metodología propuesta en este trabajo. Adicionalmente, algunas pruebas instrumentales han sido necesarias para validar aspectos de configuración y parametrización de la simulación. A continuación se describe con más detalle los resultados obtenidos al ejecutar los casos de prueba definidos en las secciones anteriores.

El informe del resultado de las pruebas, muestra por cada actividad el número de orden de ejecución, su descripción, la prueba unitaria que se ejecutó, el elemento de la simulación sobre el cual se aplicó la prueba, la hora de inicio y de fin de ejecución de la actividad,

la salida esperada del caso de prueba, la salida real que arroja el motor y finalmente una marca de tiempo del momento en que se ha validado la actividad. El filtrado mediante indicadores descrito en la Sección 6.3 se utilizó para mostrar las diferencias entre la salida real y la salida esperada.. Para evitar confusiones en el texto siguiente con las referencias a las actividades, se utilizará la abreviatura $nCS(N)(X)$, donde n es el número de la actividad, N el número del caso de estudio y X es el tipo de escenario (T escenario típico, P escenario problema y S escenario solución).

En la “salida real” se ha utilizado el sistema de reglas etiquetadas cuyo nombre coincide con el patrón que se detecta. El los sensores inerciales estos son: @left para el desplazamiento horizontal a la izquierda, @right para el desplazamiento horizontal a la derecha, @up para el desplazamiento vertical hacia arriba y @down para el desplazamiento vertical hacia abajo. En los sensores ambientales @up y @down indican que sube y baja el valor de la lectura en comparación con los valores anteriores y @change que cambia el valor de la lectura con respecto al anterior.

8.4.0.1 Caso de prueba de CS001ML

Después de definir todos las pruebas unitarias correspondientes al primer caso de prueba, la ejecución de dichas pruebas ha arrojado el informe que se muestra en la Figura 8.13. En el resumen de ejecución, solo se muestran actividades que interesan observar y sobre las que se ha definido una o varias pruebas unitarias concretas.

Simulación: CS001ML (Escenario típico)

::inicio de simulación::

#	Actividad	Prueba	Elemento	Inicio	Fin	Salida esperada	Salida real	Hora
2	El paciente enciende la televisión	getSimpleData	sensor: usuario	12:01:40	12:02:10	🟡	● @data	12:01:45
		checkMovement	sensor: entorno	12:01:40	12:02:10	cualquier gesto	⚠️ @up @left @right @down	12:01:58
3	el paciente se sienta en el sofá	getSimpleData	sensor: usuario	12:02:10	12:02:30	🟡	● @data	12:02:11
		checkPosition	sensor: usuario	12:02:10	12:02:30	cambio de posición	● @changePosition	12:02:18
		checkMovement	sensor: usuario	12:02:10	12:02:30	cualquier gesto	⚠️ @left @down	12:02:19
5	el paciente se levanta	getSimpleData	sensor: usuario	12:03:00	12:03:20	🟡	● @data	12:03:03
		checkPosition	sensor: usuario	12:03:00	12:03:20	cambio de posición	● @changePosition	12:03:09
		checkMovement	sensor: usuario	12:03:00	12:03:20	cualquier gesto	⚠️ @up @left @right	12:03:11
10	el paciente abre el refrigerador	getSimpleData	sensor: usuario	12:05:40	12:06:00	🟡	● @data	12:05:43
		checkPosition	sensor: usuario	12:05:40	12:06:00	cambio de posición	■ @changePosition	12:05:51
		checkMovement	sensor: usuario	12:05:40	12:06:00	cualquier gesto	⚠️ @up @left @left	12:05:51
12	el paciente cierra la puerta del refrigerador	getSimpleData	sensor: usuario	12:06:30	12:06:52	🟡	● @data	12:06:31
		checkPosition	sensor: usuario	12:06:30	12:06:52	cambio de posición	■ @changePosition	12:06:33
		checkMovement	sensor: usuario	12:06:30	12:06:52	cualquier gesto	⚠️ @left @up @left @down	12:06:41

::fin simulación::

🟡 Se esperan datos para la prueba.

⚠️ La salida esperada y la salida real concuerdan pero no son exactas.

● La salida esperada y la salida real Sí concuerdan.

■ La salida esperada y la salida real No concuerdan.

Figura 8.13: Resultados del caso de prueba CS001ML para el escenario típico.

El resultado de la ejecución de este caso de prueba se muestra favorable según las expectativas que se tenían con cada actividad. En la actividad 2CS1T por ejemplo, además de poder verificar que había un flujo de datos asociado a los sensores ambientales en ese intervalo de tiempo (“getSimpleData”), se han identificados varios patrones que

denotan movimientos uniformes tales como *@up,@left,@right,@down*, al tiempo que el personaje realizaba la acción correspondiente. Esto no quiere decir que las acciones se hayan ejecutado en este orden. El resultado lo que muestra son las ocurrencias predominantemente de estos patrones a lo largo del tiempo que duró la actividad validada.

En el caso de la actividad 3CS1T y 5CS1T, sucede un algo muy similar con la acción de “sentarse” y “levantarse”. En la actividad 3CS1T, el último descriptor muestra un tiempo razonable de 9 segundos para encontrar el patrón *@down* dentro de los 20 segundos que dura la actividad. Esta información se complementa con la comprobación del cambio de posición, el cual indica que a los 8 segundos de la actividad el motor ha reconocido que el cuerpo ha hecho un movimiento de traslación en el espacio (desplazamiento vertical), según los sensores inerciales utilizados.

En la actividad 5CS1T, ocurre el efecto contrario con el descriptor, uno de los patrones predominantes en el patrón *@up* y luego le siguen dos patrones diferentes *@left@right*. En este caso lo que ocurre es que la acción de levantarse puede estar acompañada de una acción inmediata posterior que no se ha reconocido, hecho que hace que tengamos más de un descriptor al finalizar la actividad.

Aunque no es fácil distinguir un patrón único en las actividades 10CS1T y 12CS1T, bien es cierto que dentro de las acciones identificadas, se describe la acción en curso en un tiempo medio de duración de la actividad. *@up* y *@down* son característicos de levantar y bajar la mano de forma frontal-vertical y esto, como se puede apreciar en el resumen, ocurre en ambas acciones. Por otra parte, el que no haya cambio de posición en la actividad 10 (*@changePosition=red*) nos confirma que dichas acciones han ocurrido mientras el usuario está quieto.

Si se observa el resultado de la ejecución de las pruebas sobre el escenario problema en la Figura 8.14, en la actividad 3CS1P se puede apreciar que ciertos patrones se mantienen y en otros el patrón optimiza su identificación en el mismo tiempo promedio.

Lo mismo ocurre con las actividades 5CS1P y 10CS1P, las cuales mantiene el descriptor principal y reducen el ruido en los descriptores finales. En la actividad 12CS1P, la descripción no termina de ser clara pero es evidente que un gesto *@down* es reconocido en el escenario típico, en esta segunda ejecución no se ha llevado a cabo por el usuario.

8.4.0.2 Caso de prueba de CS002DI

El informe de este caso de prueba muestra cómo las salidas reales de las validaciones se ajustan más a las salidas esperadas. Dichas salidas se acotan más al patrón específico esperado y permiten valorar fácilmente la situación que ocurre en esos instantes de la simulación. Como se observa en la Figura 8.15, en la actividad 2CS2T, el episodio del temblor es la única acción del usuario que cuesta más reconocer, a diferencia de las otras actividades donde los descriptores son precisos. El criterio en estos casos es que el descriptor buscado debe aparecer en el resultado, independientemente de dónde se encuentre en la salida de toda la prueba.

Cuanto más a la izquierda esté el descriptor esperado, para el motor esto supone un mayor esfuerzo para reconocer el patrón porque le obliga a repetir las consultas sobre nuevas ventanas de datos y a comparar los resultados para quedarse con el patrón más

Simulación: CS001ML (Escenario problema)

::inicio simulación::

#	Actividad	Prueba	Elemento	Inicio	Fin	Salida esperada	Salida real	Hora
2	el paciente enciende la televisión	getSimpleData	sensor: usuario	12:01:40	12:02:10	🟡	● @data	12:01:45
		checkMovement	sensor: entorno	12:01:40	12:02:10	cualquier gesto	⚠️ @up @right @right @down	12:01:58
3	el paciente se sienta en el sofá	getSimpleData	sensor: usuario	12:02:10	12:02:30	🟡	● @data	12:02:11
		checkPosition	sensor: usuario	12:02:10	12:02:30	cambio de posición	● @changePosition	12:02:18
		checkMovement	sensor: usuario	12:02:10	12:02:30	cualquier gesto	● @down	12:02:19
5	el paciente se levanta	getSimpleData	sensor: usuario	12:03:00	12:03:20	🟡	● @data	12:03:03
		checkPosition	sensor: usuario	12:03:00	12:03:20	cambio de posición	● @changePosition	12:03:09
		checkMovement	sensor: usuario	12:03:00	12:03:20	cualquier gesto	⚠️ @left @up	12:03:11
10	el paciente abre el refrigerador	getSimpleData	sensor: usuario	12:05:40	12:06:00	🟡	● @data	12:05:43
		checkPosition	sensor: usuario	12:05:40	12:06:00	cambio de posición	■ @changePosition	12:05:51
		checkMovement	sensor: usuario	12:05:40	12:06:00	cualquier gesto	⚠️ @left @up	12:05:51
12	El paciente vuelve al sofá y se olvida de cerrar la puerta del refrigerador.	getSimpleData	sensor: usuario	12:06:30	12:06:52	🟡	● @data	12:06:31
		checkPosition	sensor: usuario	12:06:30	12:06:52	cambio de posición	● @changePosition	12:06:33
		checkMovement	sensor: usuario	12:06:30	12:06:52	cualquier gesto	● @left	12:06:41

::fin simulación::

- 🟡 Se esperan datos para la prueba.
- ⚠️ La salida esperada y la salida real concuerdan pero no son exactas.
- La salida esperada y la salida real Sí concuerdan.
- La salida esperada y la salida real No concuerdan.

Figura 8.14: Resultados del caso de prueba CS001ML para el escenario problema.

predominante. Cuando el caso es que no hay patrones predominantes, esto podría ser un indicio de que la actividad ejecutada dura poco tiempo o que el personaje realiza demasiadas acciones que no se pueden analizar con el número de sensores utilizados.

Simulación: CS002DI (Escenario típico)

::inicio simulación::

#	Actividad	Prueba	Elemento	Inicio	Fin	Salida esperada	Salida real	Hora
2*	el paciente tiene episodios de temblores	getSimpleData	sensor: usuario	13:00:00	13:00:35	🟡	● @data	13:00:15
		checkMovement	sensor: usuario	13:00:00	13:00:35	temblores	⚠️ @left @left @left @tremor	13:00:18
8*	espera 60 segundos en pie	getSimpleData	sensor: usuario	13:04:30	13:05:31	🟡	● @data	13:04:39
		checkMovement	sensor: usuario	13:04:30	13:05:31	inmóvil temblores	● @right ⚠️ @left @left	13:04:41
		checkStop	sensor: usuario	13:04:30	13:05:31	cambio de posición	■ @stop	13:04:48
10*	espera 30 segundos en pie	getSimpleData	sensor: usuario	13:06:21	13:06:51	🟡	● @data	13:06:24
		checkMovement	sensor: usuario	13:06:21	13:06:51	inmóvil más temblores menos descanso	● @up ⚠️ @left @left ● @up	13:06:25
		checkStop	sensor: usuario	13:06:21	13:06:51	cambio de posición	■ @stop	13:06:26
12*	espera 20 segundos en pie	getSimpleData	sensor: usuario	13:07:42	13:08:31	🟡	● @data	13:07:44
		checkMovement	sensor: usuario	13:07:42	13:08:31	inmóvil más temblores menos descanso	● @left ⚠️ @right @right	13:07:47
		checkStop	sensor: usuario	13:07:42	13:08:31	cambio de posición	■ @stop	13:07:55
13	el paciente camina hacia la habitación	getSimpleData	sensor: usuario	13:08:01	13:08:52	🟡	● @data	13:08:03
		checkMovement	sensor: usuario	13:08:01	13:08:52	más temblores	⚠️ @left	13:08:05
		checkStop	sensor: usuario	13:08:01	13:08:52	cambio de posición	● @stop	13:08:08

::fin simulación::

- 🟡 Se esperan datos para la prueba.
- ⚠️ La salida esperada y la salida real concuerdan pero no son exactas.
- La salida esperada y la salida real Sí concuerdan.
- La salida esperada y la salida real No concuerdan.

Figura 8.15: Resultados del caso de prueba de CS002DI escenario típico.

En casos donde no hay patrones predominantes, los descriptores no son muy precisos

a la hora de identificar la salida esperada y se debe valorar la posibilidad de añadir nuevas pruebas unitarias que permitan aclarar la situación, como se observa Figura 8.16.

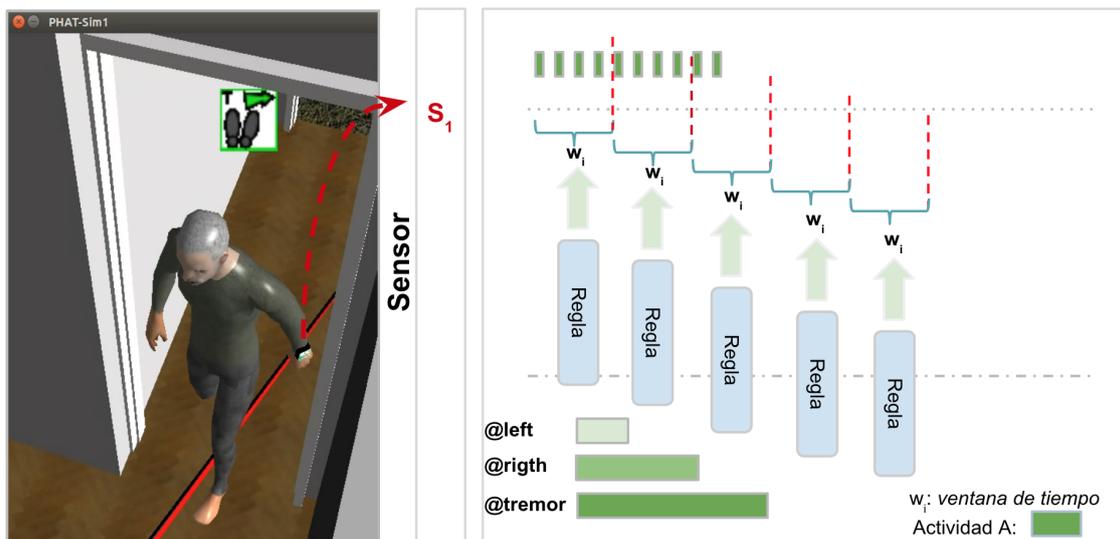


Figura 8.16: Evolución de la predicción del etiquetado de reglas en acciones repetitivas.

Un ejemplo de esto ocurre con la actividad 8CS2T, en la que al intentar identificar patrones de temblores, el motor arroja un patrón diferente *@left@left*. En cuanto a los cambios de posición, los descriptores son precisos y dejan claro que el usuario no realiza ningún cambio de posición durante las actividades 8CS2T, 10CS2T y 12CS2T, mientras que en la actividad 13CS2T sí lo hace. Se debe tener presente que en este escenario, los temblores son puntuales de la actividad 2CS2T, lo cual quiere decir que no se intensifican a lo largo del tiempo.

En el escenario problema, esta situación sí se propone como un reto que la solución debe reconocer. En los resultados que se muestran en la Figura 8.17, además de reducir los tiempos de espera, los temblores se agudizan como un posible indicio de la desorientación que se ha querido modelar en este escenario.

8.4.0.3 Pruebas secundarias para reconocer problemas de movilidad

Al introducir características en el comportamiento del usuario que involucren su actividad física, se han establecido nuevas pruebas secundarias que permitieron evidenciar si hay algún otro problema que se ha incluido en la especificación de la simulación relacionada con el usuario. Tal y como se describe en la Sección 4.2.6, pruebas destinadas a identificar si hay balanceo del tronco en el usuario o si este mueve mucho los brazos al caminar, pueden ser útiles para describir con más detalle a los expertos lo que se quiso plasmar en la simulación con respecto al comportamiento físico del usuario.

En el escenario problema del caso de estudio CS002DI, sobre la actividad 2CS2P se han añadido dos pruebas *executeDescriptor* que buscan identificar si además de temblores, el usuario presenta episodios donde balancea el tronco de frente *@frontalRollingTorso* o lateralmente *@lateralSwingTorso*. En ambos casos, las pruebas no encuentran dichos problemas en la movilidad del usuario. En la actividad 8CS2P el objetivo era intentar

Simulación: CS002DI (Escenario problema)

::inicio simulación::

#	Actividad	Prueba	Elemento	Inicio	Fin	Salida esperada	Salida real	Hora
2*	el paciente tiene episodios de temblores	getSimpleData	sensor: usuario	13:00:00	13:00:35	🟡	● @data	13:00:15
		checkMovement	sensor: usuario	13:00:00	13:00:35	temblores	▲ @up @left @left	13:00:18
8*	espera 60 segundos en pie	getSimpleData	sensor: usuario	13:04:30	13:05:31	🟡	● @data	13:04:39
		checkMovement	sensor: usuario	13:04:30	13:05:31	inmóvil	● @up	13:04:41
		checkMovement	sensor: usuario	13:04:30	13:05:31	más temblores	▲ @left @right @down @tremor	
		checkStop	sensor: usuario	13:04:30	13:05:31	cambio de posición	■ @stop	13:04:48
10*	espera 30 segundos en pie	getSimpleData	sensor: usuario	13:06:21	13:06:51	🟡	● @data	13:06:24
		checkMovement	sensor: usuario	13:06:21	13:06:51	inmóvil	● @up	13:06:25
		checkMovement	sensor: usuario	13:06:21	13:06:51	más temblores	▲ @left @tremor @tremor	
		checkMovement	sensor: usuario	13:06:21	13:06:51	menos descanso	● @up	
		checkStop	sensor: usuario	13:06:21	13:06:51	cambio de posición	■ @stop	13:06:26
12*	espera 20 segundos en pie	getSimpleData	sensor: usuario	13:07:42	13:08:31	🟡	● @data	13:07:44
		checkMovement	sensor: usuario	13:07:42	13:08:31	inmóvil	● @up	13:07:47
		checkMovement	sensor: usuario	13:07:42	13:08:31	más temblores	● @tremor	
		checkMovement	sensor: usuario	13:07:42	13:08:31	menos descanso		
		checkStop	sensor: usuario	13:07:42	13:08:31	cambio de posición	■ @stop	13:07:55
13	camina hacia la habitación	getSimpleData	sensor: usuario	13:08:01	13:08:52	🟡	● @data	13:08:03
		checkMovement	sensor: usuario	13:08:01	13:08:52	más temblores	▲ @tremor @tremor	13:08:05
		checkStop	sensor: usuario	13:08:01	13:08:52	cambio de posición	● @stop	13:08:08

::fin simulación::

🟡 Se esperan datos para la prueba.

▲ La salida esperada y la salida real concuerdan pero no son exactas.

● La salida esperada y la salida real Si concuerdan.

■ La salida esperada y la salida real No concuerdan.

Figura 8.17: Resultado del caso de prueba CS002DI para el escenario problema.

reconocer si los temblores van acompañados de un incremento *@increaseSwingingArms* o decremento *@decreaseSwingingArms* en el balanceo de los brazos. En este caso concreto, el segundo descriptor señala que la movilidad de los brazos se ha visto reducida desde que empezó la simulación, situación que podría notificarse al experto para que revise la simulación en estos instante de tiempo. Al observar que el resultado de las pruebas secundarias y la salida esperada coinciden, se ha aplicado el mismo criterio de las pruebas principales, las cuales superando el 50,00% de las coincidencias determinan el acierto del caso de prueba que describe el comportamiento del usuario en la simulación. El resultado de las pruebas secundarias se muestran en la Figura 8.18.

8.4.0.4 Caso de prueba de CS003AR

Como se indicó en la definición de los casos de estudio, en este caso de prueba existen varios aspectos de la simulación que se han evaluado en el entorno de pruebas. Por un lado se han validado las actividades establecidas, las cuales describen el comportamiento esperado de lo simulado de cara a la solución asistiva. Por otra parte, está el sistema de monitorización del usuario que se ha incluido en la simulación como parte de la solución asistiva y, finalmente, se encuentra el sistema HAR que se ha empleado para reconocer las actividades físicas del usuario. En la Figura 8.19 se puede ver cómo abordando estos tres aspectos en la evaluación, los resultados son más acertados cuando se incluye el sistema HAR para el reconocimiento de actividades concretas como “walk”, “drink”, “clap”, “wave attention”, “stop” o “motionless”).

Simulación: CS002DI (Escenario problema)

:::inicio simulación:::

#	Actividad	Prueba	Elemento	Inicio	Fin	Salida esperada	Salida real	Hora
2*	el paciente tiene episodios de temblores	getSimpleData	sensor: usuario	13:00:00	13:00:35	■	● @data	13:00:15
		checkMovement	sensor: usuario	13:00:00	13:00:35	temblores	▲ @up @left @left	13:00:18
		executeDescriptor	sensor: usuario	13:00:00	13:00:35	balanceo frontal torso @frontalRollingTorso	■	13:00:18
		executeDescriptor	sensor: usuario	13:00:00	13:00:35	balanceo lateral torso @lateralSwingTorso	▲ @left @left	13:00:19
8*	espera 60 segundos en pie	getSimpleData	sensor: usuario	13:04:30	13:05:31	■	● @data	13:04:39
		checkMovement	sensor: usuario	13:04:30	13:05:31	inmóvil	● @up	13:04:41
		checkMovement	sensor: usuario	13:04:30	13:05:31	más temblores	▲ @left @right @down @tremor	13:04:41
		checkStop	sensor: usuario	13:04:30	13:05:31	cambio de posición	■ @stop	13:04:48
		executeDescriptor	sensor: usuario	13:04:30	13:05:31	aumentar balanceo de brazos @increaseSwingingArms	■ @stop	13:04:50
executeDescriptor	sensor: usuario	13:04:30	13:05:31	disminuir balanceo de brazos @decreaseSwingingArms	● @decreaseSwingingArms	13:04:50		
10*	espera 30 segundos en pie	getSimpleData	sensor: usuario	13:06:21	13:06:51	■	● @data	13:06:24
		checkMovement	sensor: usuario	13:06:21	13:06:51	inmóvil	● @up	13:06:25
		checkMovement	sensor: usuario	13:06:21	13:06:51	más temblores	▲ @left @tremor @tremor	13:06:25
		checkMovement	sensor: usuario	13:06:21	13:06:51	menos descanso	● @up	13:06:25
checkStop	sensor: usuario	13:06:21	13:06:51	cambio de posición	■ @stop	13:06:26		
12*	espera 20 segundos en pie	getSimpleData	sensor: usuario	13:07:42	13:08:31	■	● @data	13:07:44
		checkMovement	sensor: usuario	13:07:42	13:08:31	inmóvil	● @up	13:07:47
		checkMovement	sensor: usuario	13:07:42	13:08:31	más temblores	● @tremor	13:07:47
checkMovement	sensor: usuario	13:07:42	13:08:31	menos descanso	● @tremor	13:07:47		
checkStop	sensor: usuario	13:07:42	13:08:31	cambio de posición	■ @stop	13:07:55		
13	el paciente camina hacia la habitación	getSimpleData	sensor: usuario	13:08:01	13:08:52	■	● @data	13:08:03
		checkMovement	sensor: usuario	13:08:01	13:08:52	más temblores	▲ @tremor @tremor	13:08:05
		checkStop	sensor: usuario	13:08:01	13:08:52	cambio de posición	● @stop	13:08:08

:::fin simulación:::

■ Se esperan datos para la prueba..

▲ La salida esperada y la salida real concuerdan pero no son exactas.

● La salida esperada y la salida real Sí concuerdan.

■ La salida esperada y la salida real No concuerdan.

Figura 8.18: Añadiendo pruebas secundarias en el caso de prueba CS002DI para el escenario problema.

Simulación: CS003AR (Escenario típico)

:::inicio simulación:::

#	Actividad	Prueba	Elemento	Inicio	Fin	Salida esperada	Salida real	Hora
1*	el paciente se despierta	getSimpleData	sensor: usuario, entorno	14:00:00	14:01:35	■	● @data	14:00:04
		checkPosition	sensor: usuario	14:00:00	14:01:35	wake up (@up: lift the torso)	▲ @up @left @left @stop	14:00:18
		checkValue	sensor: usuario	14:00:00	14:01:35	@lightOn in the room	● @lightOn	14:00:22
2*	el paciente tiene episodios de temblores en la mano izquierda	getSimpleData	sensor: usuario	14:01:35	14:02:20	■	● @data	14:01:38
		getMovement	sensor: usuario	14:01:35	14:02:20	@tremor (temblor en la mano izquierda)	● @activity='tremor'	14:01:52
3	el paciente camina hacia el salón	checkActivity	sensor: usuario	14:02:20	14:03:15	action='walk'	● @activity='walk'	14:02:31
4	enciende la TV	checkMovement	sensor: usuario, entorno	14:03:15	14:04:00	cualquier gesto	▲ @left @left @up	14:03:33
5	camina a la cocina	checkActivity	sensor: usuario	14:04:00	14:04:55	action='walk'	● @activity='walk'	14:04:13
6	el paciente bebe agua	checkActivity	sensor: usuario	14:04:55	14:05:40	action='drink'	● @activity='drink'	14:05:06
7	espera 60 segundos en pie	checkStop	sensor: usuario	14:05:40	14:06:40	action='stop'	● @stop	14:05:57
		checkMovement	sensors: user	14:05:40	14:06:40	action='stop'	▲ @activity='stop' @stop	14:06:11
8	el paciente aplaude	checkActivity	sensor: usuario	14:06:40	14:07:35	action='clap'	● @activity='clap'	14:06:52

:::fin simulación:::

■ Se esperan datos para la prueba..

▲ La salida esperada y la salida real concuerdan pero no son exactas.

● La salida esperada y la salida real Sí concuerdan.

■ La salida esperada y la salida real No concuerdan.

Figura 8.19: Resultados del caso de prueba CS003AR para el escenario típico.

En las pruebas donde se accede al uso de reglas, no hay predicción concreta por parte del framework, por lo que es necesario añadir pruebas secundarias que recojan información del

ambiente para corroborar determinada situación o salida esperada. En la actividad 1CS3T por ejemplo, se usó la información de un sensor ambiental para determinar si había luz o no en la habitación, dado que asegurar que el usuario se ha “despertado” por el solo hecho de reconocer algún tipo de movimiento puede no ser correcto (el moverse mientras se duerme es común).

Reconocer temblores es una tarea más sencilla de llevar a cabo con un sistema HAR que con un conjunto de reglas. Los sistemas HAR aprenden más fácilmente de señales periódicas y los temblores suelen estar asociados a este tipo de señal, siempre que no haya un trastorno motriz mayor. La predicción en este caso (ver actividad 2CS3T) es acertada, como ocurre también con actividades periódicas similares “walk”, “drink”, “clap”, o “wave attention” en las actividades 3CS3T, 5CS3T, 6CS3T y 8CS3T.

En la actividad 7CS3T se ha combinado el uso de reglas con el sistema HAR y el resultado ha mostrado coincidencias en ambas pruebas. Es importante destacar que en acciones como “estar en reposo” la predicción mediante el sistema HAR es mucho más complicada de obtener debido a que el usuario nunca estará en absoluto reposo. Las reglas en este caso son más fiables porque al actuar sobre el comportamiento de los sensores por separado, es mucho más factible considerar que ocurra el estado de reposo en función de un criterio puntual como “el cuerpo no se mueve de sitio por ejemplo”, independientemente de que el usuario esté moviendo las manos.

En la Figura 8.20, se muestra cómo los resultados del reconocimiento siguen siendo precisos excepto en casos donde la acción física describe patrones muy similares a otras acciones, como ocurre con las actividades 2CS3P y 7CS3P. En este caso, se puede asegurar que la información de los sensores utilizados, no ha sido insuficiente para obtener una precisión mayor en la predicción de dichas acciones.

Simulación: CS003AR (Escenario problema)

:::inicio simulación:::

#	Actividad	Prueba	Elemento	Inicio	Fin	Salida esperada	Salida real	Hora
1*	el paciente se despierta	getSimpleData	sensor: usuario, entorno	16:00:00	16:01:35	🟡	● @data	16:00:04
		checkPosition	sensor: usuario	16:00:00	16:01:35	wake up (@up: lift the torso)	⚠ @up @left @left @stop	16:00:18
		checkValue	sensor: usuario	16:00:00	16:01:35	@lightOn in the room	● @lightOn	16:00:22
2*	el paciente tiene episodios de temblores en la mano izquierda	getSimpleData	sensor: usuario	16:01:35	16:02:20	🟡	● @data	16:01:38
		getMovement	sensor: usuario	16:01:35	16:02:20	@tremor (temblor en la mano izquierda) (mayor intensidad)	⚠ @activity='tremor' @activity='tremor' @activity='run'	16:01:52
3	el paciente camina hacia el salón	checkActivity	sensor: usuario	16:02:20	16:03:15	action='walk'	● @activity='walk'	16:02:31
4	turn on the TV	checkMovement	sensor: usuario, entorno	16:03:15	16:04:00	cualquier gesto	⚠ @left @left @up	16:03:33
5	el paciente camina hacia la cocina	checkActivity	sensors: user	16:04:00	16:04:55	action='walk'	● @activity='walk'	16:04:13
6	el paciente bebe agua	checkActivity	sensor: usuario	16:04:55	16:05:40	action='drink'	● @activity='drink'	16:05:06
7	espera 60 segundos en pie	checkStop	sensor: usuario	16:05:40	16:06:40	action='stop'	● @stop	16:05:57
		checkMovement	sensor: usuario	16:05:40	16:06:40	action='stop'	⚠ @activity='stop' @stop	16:06:11
8	el paciente llama la atención	checkActivity	sensors: user	16:06:40	16:07:35	action='waveAttention'	⚠ @activity='waveAttention' @activity='drink' @activity='waveAttention'	16:06:52

:::fin simulación:::

🟡 Se esperan datos para la prueba..

⚠ La salida esperada y la salida real concuerdan pero no son exactas.

● La salida esperada y la salida real Sí concuerdan.

■ La salida esperada y la salida real No concuerdan.

Figura 8.20: Resultados del caso de prueba CS003AR para el escenario problema.

En el escenario problema parte de la situación que se ha añadido es mantener los temblores durante toda la simulación y aumentar su intensidad en determinados momentos. De igual forma, se ha usado animaciones diferentes en las actividades del personaje y se ha cambiado la acción física en la actividad 8CS3P. Todo el proceso de entrenamiento con la simulación descrito en el Capítulo 7, ha permitido generar nuevos modelos de clasificación capaces de reconocer todas las animaciones de las actividades utilizadas en el caso de prueba.

8.4.1 Evaluación del sistema de reglas

Para evaluar el sistema de reglas, lo que se propone es realizar modificaciones sobre las animaciones de cada actividad física del personaje. Lo que se busca es variar la forma de ejecutar determinadas animaciones en la simulación para evitar que éstas sean repetitivas (no usar siempre la misma animación). Cómo se indica en la Sección 7.2.4.2, en el framework la actividad física o animación, puede alterarse de forma programática para cambiar tanto el ángulo de desplazamiento en los huesos del personaje como la cadencia con que realiza dichas acciones. De esta forma, es posible asumir que para validar la ejecución del mismo caso de uso contra el sistema HAR, realmente se estarían ejecutando un número considerable de simulaciones cuyas actividades se describen igual pero físicamente son realizadas por el personaje de forma diferente (animaciones diferentes de una misma actividad).

Lo que sigue a continuación, es medir la coincidencia entre las diferencias del comportamiento en cada actividad y los resultados obtenidos con la ejecución del entorno de pruebas. Para este propósito, las salidas del sistema se han asociado a dos escenarios concretos: Problema, (P, es decir, simulaciones que señalan un escenario problema en el entorno) y Normales, (N, simulaciones de un escenario típico que no especifican problemas en el entorno). Teniendo en cuenta que un caso de estudio puede especificar escenarios diferentes (Normal y Problema), el número total de simulaciones disponibles son 50, de las cuales 25 corresponden al caso de estudio CS001ML y las otras 25 al caso CS002DI.

Para cada caso, 15 simulaciones (un 60 %) son de escenario Problema y 10 (un 40 %) son de escenarios Normales. El comportamiento de los usuarios no es posible modelar uniformemente, por ello no sería correcto definir igual número de simulaciones para ambos escenarios. En el caso del escenario problema, se espera que el sistema reconozca que no se ha encontrado una acción concreta esperada o que hay un resultado diferente al esperado, mientras que en las simulaciones normales es totalmente contrario, se espera que no haya ninguna salida que genere alarmas. Bajo estas premisas, para que una simulación sea considerada como exitosa, al menos el 80 % de sus validaciones han sido correctas. Es decir, que el resultado de las validaciones ha coincidido con la salida esperada de cada actividad.

En la Figura 8.21, se muestra la matriz de confusión generada a partir de los resultados del entorno de pruebas. Las columnas P_v y N_v son las instancias de simulaciones validadas, mientras que P y N , en cada fila, representan el total de instancias de simulaciones. La diagonal sombreada indica el número de simulaciones validadas como correctas y las celdas blancas el número de validaciones que no han superado la validación. A los

primeros valores se les conoce como verdaderos positivos (TP) y verdaderos negativos (TN) respectivamente, mientras que los últimos falsos positivos (FP) y falsos negativos (FN), considerando las validaciones correctas como verdaderas y las validaciones erróneas como falsas.

		Pruebas framework				Pruebas framework			
		P_v	N_v			P_v	N_v		
simulaciones	CS001ML	12	3	simulaciones	CS002DI	10	5	TP	FN
	P	1	9		N	3	7	FP	TN

P: Escenario problema, N: Escenario típico, P_v : validaciones del escenario problema, N_v : validaciones del escenario típico

Figura 8.21: Matriz de confusión de la evaluación del sistema de reglas.

La Tabla 8.23, se muestran los indicadores de evaluación del sistema. *Accuracy* es la tasa de acierto, es decir, el número de simulaciones problema validadas correctamente con respecto al número total de ellas. Los otros indicadores son *Precision* y *Recall*, los cuales describen el número de validaciones problema validadas correctamente, con respecto al total de simulaciones problema y al total de normales, respectivamente. Otro indicador que resulta útil en estos caso es *F-Measurem*, el cual se calcula a partir de estos dos últimos indicadores.

	CS001ML	CS002DI
Accuracy	0.840	0.680
Precision	0.923	0.769
Recall	0.800	0.667
F-Measurem	0.857	0.714

Tabla 8.23: Indicadores de evaluación del entorno de pruebas para CS001ML y CS002DI.

Con estos indicadores, se puede concluir que en el caso CS001ML cumple con 84 % las expectativas con respecto a la ejecución de la actividades propuestas. La precisión es favorable al encontrarse con un 92,3 % a la hora de identificar dichas actividades en este escenario. En el caso CS002DI la precisión es un tanto menor, con un 68 % se puede considerar que las pruebas no han sido suficientes y que convendría aumentar el número de ellas para mejorar estos resultados.

8.4.2 Evaluación del sistema de reconocimiento

Para evaluar el sistema de reconocimiento, se han usado las modificaciones de las actividades físicas del personaje, las cuales variaron la forma de realizar las acciones en la animación. Gracias a este procedimiento, se han generado un total de 109 acciones, distribuidas entre las distintas actividades a reconocer, como se observa en la Tabla 8.24. Esta tarea permitió que el personaje ejecutara la misma acción pero de forma diferente, como si se tratasen de usuarios distintos. Las animaciones modificadas fueron las que

se querían identificar en el caso de estudio: “walk”, “drink”, “clap”, “wave-attention” o “stop” (*motionless*). Este proceso se llevó a cabo de forma programática, cambiando tanto el ángulo de desplazamiento en los huesos del personaje como la cadencia con que realizaba dichas acciones.

walk	drink	clap	wave-attention	stop
28	31	29	12	9

Tabla 8.24: Total de modificaciones por cada animación.

La matriz de confusión generada en este caso, se muestra en la Figura 8.22. Siguiendo el mismo criterio empleado en la evaluación del sistema de reglas, los escenarios de simulación se han asociado a dos tipos de escenarios concretos: Problema, (P, es decir, simulaciones que señalan un problema en el entorno) y Normales, (N, simulaciones que no especifican un problema en el entorno).

		Pruebas framework	
		P_v	N_v
simulaciones	P	11	4
	N	2	8

TP	FN
FP	TN

P: Escenario problema, N: Escenario típico
 P_v : validaciones del escenario problema, N_v : validaciones del escenario típico

Figura 8.22: Matriz de confusión de la evaluación del sistema de reconocimiento.

Para este proceso, el número total de simulaciones disponibles son 25, de las cuales 15 corresponden a escenarios problema y las otras 10 a escenarios normales. Para la matriz de confusión, las columnas P_v y N_v son las instancias de simulaciones validadas, mientras que P y N , en cada fila, representan las instancias de simulaciones generadas. La diagonal sombreada indica el número de simulaciones validadas como correctas y las celdas blancas aquellas que no han superado el proceso de validación. Los indicadores de evaluación obtenidos de ambos escenarios (típico y problema) se muestran en la Tabla 8.25.

	CS003AR
Precision	0,760
Accuracy	0,846
Recall	0,733
F-Measurem	0,786

Tabla 8.25: Indicadores de evaluación del entorno de pruebas para CS003AR.

En el caso CS003AR se han añadido pruebas enfocadas al reconocimiento de la actividad del personaje. Con una precisión del 76 % se han podido reconocer gran parte de las actividades propuestas, con una exactitud importante del 84,6 % mostrando que

el sistema HAR funciona ante la variedad de modificaciones hechas sobre las actividades. Con el favorable resultado de los tres escenarios, especialmente en el caso CS003AR, el paso siguiente será poner a prueba el sistema HAR en un entorno real, respetando los parámetros establecidos en este caso de estudio.

8.5 Resumen

En este capítulo se ha descrito la experimentación llevada a cabo para validar distintas características de soluciones asistivas que han sido trasladadas a las simulaciones propuestas. En la experimentación, se ha hecho uso de la metodología descrita en los Capítulos 3 y 4 para definir los casos de prueba. Los casos de estudio han sido generados a partir de entrevistas previas que se obtuvieron para el proyecto ColosAAL. Con la información aportada por los pacientes, se establecieron las actividades que serían relevantes para construir los escenarios simulados. A las actividades elegidas se le asignaron diferentes salidas esperadas que luego fueron utilizadas para comparar los resultados en el entorno de pruebas sobre las simulaciones.

La experimentación propuesta, ha supuesto la construcción de tres casos de prueba relacionados con los casos de estudios correspondientes. Cada caso de prueba se asoció a los escenarios definidos en cada caso de estudio. Posteriormente, en los casos de prueba se incluyeron las pruebas unitarias necesarias para validar la lista de actividades que componían las simulaciones. Los resultados de dichas pruebas, se han podido observar en informes que ha generado el entorno de pruebas con ayuda del framework.

Con el resultado de las pruebas unitarias, se ha podido comprobar cómo cada simulación cumplía con las especificaciones de requisitos de la solución, es decir, con las actividades que tendrían que ejecutarse en la simulación para describir el escenario propuesto. Con dichos escenarios, se podía observar de forma visual y con ayuda del framework, cómo cada requerimiento era ejecutado en la simulación cumpliendo o no con las expectativas (salidas esperadas) que se habían establecido en el caso de estudio. El uso de reglas permitió reconocer en la simulación tanto características del entorno como del usuario, las cuales luego fueron usadas para contrastarlas con las salidas esperadas de cada actividad. El uso del sistema de recomendación ha sido útil en los casos que se quería reconocer la actividad física del persona, la cual había sido modelada previamente a partir de comportamientos de usuarios reales.

Los resultados de la evaluación han mostrado que el uso de las pruebas unitarias sobre la simulación, permiten reconocer aquellas características que son claramente identificables y que pueden ser medidas en la simulación. Identificar dichas características requiere de la participación de los expertos que conocen de la asistencia, especialmente cuando lo que se busca es utilizarlas para generar soluciones en este ámbito.

Capítulo 9

Pruebas en un entorno real

En el Capítulo 8 se construyó el sistema HAR para reconocer y validar las actividades propuestas en uno de los casos de estudio. Este sistema fue entrenado con ayuda de la simulación, utilizando las actividades físicas simuladas con las modificaciones aplicadas por el framework. Según los resultados obtenidos en la evaluación del sistema de reconocimiento, descrito en la Sección 8.4.2, el sistema parece comportarse bien a la hora de hacer la identificación de la actividad, funcionando claro está en un entorno por ordenador aplicado sobre la simulación.

En este capítulo, lo que se propone es validar dicho sistema usándolo directamente en un entorno real donde sea posible implementar y desplegar dicha solución y utilizarla contra datos reales para reconocer las mismas actividades propuestas en el caso de estudio para el que fue construido. El alcance de las pruebas se explica en la Sección 9.1. En la Sección 9.2, se muestra la unidad central de proceso utilizada como nodo principal de la solución. Los sensores utilizados se describen la Sección 9.3, mientras que los diferentes prototipos que han sido construido se presentan en la Sección 9.4. En esta misma Sección se incluyen las pruebas y los resultados de la evaluación del prototipo que finalmente se ha probado.

9.1 Alcance de las pruebas

En el capítulo anterior se ha podido comprobar que los indicadores de evaluación obtenidos de la simulación, son lo suficientemente buenos como para dar por sentado que los escenarios simulados se corresponden con la especificación definida para la solución. De igual forma, los sistemas que se han incluido en dicha simulación como parte de la solución, han sido utilizados en este proceso de evaluación para comprobar que su funcionamiento es el esperado en un entorno real que cumpla con las mismas especificaciones del simulado. En el caso del sistema HAR, su funcionamiento se puso a prueba evaluando las distintas actividades físicas que realizaba el personaje en cada escenario. Esto permitió comprobar que los movimientos físicos del personaje, se correspondían con acciones físicas muy similares a las reales.

Ante estos resultados, el reto siguiente ha sido implementar el sistema de reconocimiento, ya entrenado y configurado, en un entorno real que cumpliera con todas

las especificaciones que se han descrito en los casos de estudios correspondientes, tanto en la Sección 8.3.1 como en la Sección 8.3.3. Para poder llevar a cabo la implementación, fue necesario utilizar un dispositivo capaz de cumplir con las mismas funciones que se llevaban a cabo en las pruebas de la simulación. Es decir, un dispositivo donde fuese posible desplegar y poner en funcionamiento el sistema diferente al ordenador. Después de esto, el siguiente paso fue realizar las mismas validaciones de las actividades físicas pero en esta ocasión utilizando datos de usuarios reales.

Cumplir con el despliegue y la implementación del sistema de monitorización propuesto requería definir dos elementos importantes para el funcionamiento del dispositivo a utilizar: construir una unidad de proceso principal donde se ejecutase el sistema y establecer los sensores inerciales para la monitorización de la actividad física sobre el propio usuario.

- **Unidad principal de proceso:** en este elemento se desplegarían los componentes de la arquitectura necesarios para poner en marcha el framework.
- **Sensores inerciales:** serían los encargados monitorizar al usuario siguiendo las especificaciones utilizadas en el caso de prueba CS003AR.

9.2 Unidad principal de proceso

En el sistema a utilizar para las pruebas, era necesario contar con un dispositivo que cumpliera con una serie de requisitos necesarios para el funcionamiento del sistema. Entre estos se destacan principalmente: (i) que haya compatibilidad con el lenguaje Python y Java, (ii) que tenga la potencia suficiente para ejecutar los algoritmos de clasificación utilizados por el framework, (iii) que use un bajo consumo de energía que permita al sistema estar en marcha al menos durante 12 horas ininterrumpidas y por último (iv) que ofrezca una fácil comunicación mediante puertos *Inter Integrated Circuits* (I2C) [i2c14], wireless [Sek10] y *Bluetooth* [Lu12]. Estas premisas llevaron a realizar pruebas con los modelos de hardware que se observan en la Figura 9.1.

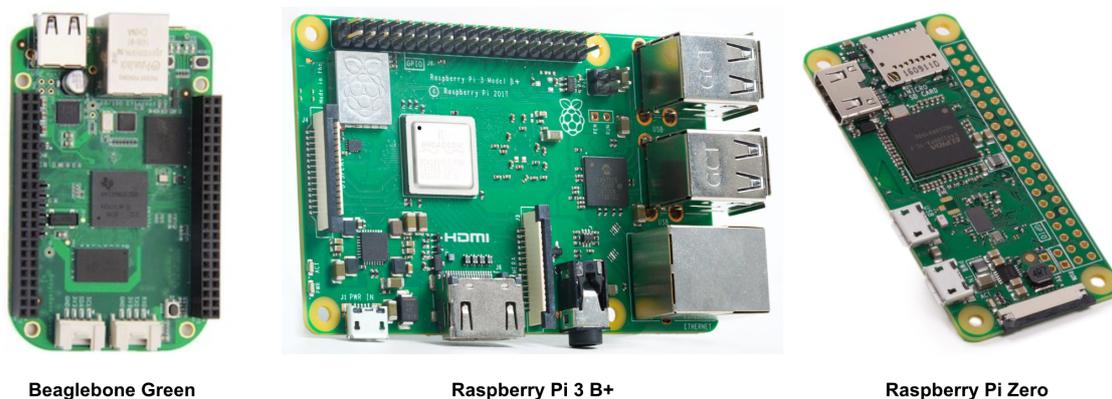


Figura 9.1: Ordenadores de una sola placa utilizados en las pruebas.

Otros modelos como *Beaglebone black* [Bea15], *Arduino* [Ard12] o *pyarduino* [PyP14], no se usaron en este desarrollo por no ser del todo compatible con ciertas librerías de software que eran necesarias para el despliegue del framework.

Una vez desplegado el framework en los dispositivos, se puso a prueba el rendimiento de cada uno de ellos ejecutando las pruebas descritas en la Sección 7.2.2, empleando sólo el algoritmos de clasificación SVC que más tiempo de proceso consumió en dichas pruebas. Aunque las pruebas de potencia y *Central Processing Unit* (CPU) fueron adecuadas en los tres modelos propuestos, el dispositivo seleccionado para hacer el despliegue final fue *Beaglebone Green* [Bea15], debido principalmente a su bajo consumo de energía a la hora de poner en marcha el sistema.

En comparación con este modelo, *Raspberry Pi* [Ras16a] cuenta con interfaz gráfica *High-Definition Multimedia Interface* (HDMI), lo cual le lleva a demandar mayor consumo de energía salvo que dicha interface sea desactivada. Con esto último se tenía otro problema y es que se hacía necesario eliminar funcionalidades en el dispositivo para lograr el consumo de energía adecuado. Con el modelo Zero [Ras16b] ocurría el mismo fenómeno, aunque en este caso había otro problema añadido y era que el tratamiento de datos con clasificadores como SVC, se tomaba tiempos de procesos superiores a los observados en la simulación, con lo cual se asumió que podría no haber un tratamiento oportuno de los datos en tiempo de ejecución. Estos resultados de las pruebas de potencia se muestran en la Tabla 9.1.

Dispositivo	Reposo mA	Trabajando mA	Capturando datos mA	Ejecutando SVC mA
Beaglebone green	89	98	101	176
Raspberry Pi 3 B+	144	310	377	778
Zero	105	230	293	898

Tabla 9.1: Comparativa del consumo de potencia en los dispositivos propuestos.

Esta comparativa no tuvo en cuenta la conexión con los sensores, simplemente se basó en medir el consumo de energía mientras el sistema se ejecutaba en el propio dispositivo.

9.3 Sensores

Las pruebas relacionadas con los sensores implicaron trabajar con dos tipos concretos de dispositivos. Por un lado están los (i) *wearables*, cuyos modelos se muestran en la Figura 9.2. Sus características principales son, entre otras, soportar varios sensores en un mismo dispositivo, ofrecer una interface propia para acceder a los datos del sensor, incluir su propia fuente de alimentación y mantener una conexión constante (preferiblemente *Bluetooth*) con la unidad principal de proceso.

Los modelos utilizados en este grupo son (a) *SimpleLink SensorTag*¹, (b) *Hexiwear*² y (c) *Steval WeSU*³. Debido a que estos dispositivos tienen su propia fuente de alimentación, una de las pruebas realizadas fue determinar el tiempo de autonomía con *Trabajando* y sin *Reposo* la conexión *Bluetooth* activada, después de una carga completa. Mientras estuvo activado el *Bluetooth* del dispositivo, este enviaba datos constantes de tres sensores internos

¹Propietario: <http://www.ti.com/tools-software/sensortag.html>

²Propietario: <https://www.hexiwear.com>

³Propietario: <http://www.st.com/en/evaluation-tools/steval-wesu1.html>



Figura 9.2: Wearables utilizados en las pruebas.

a una *Raspberry Pi*.

Como se observa en la Tabla 9.2, bajo este criterio, el sensor *Steval WeSU* resulta ser el más idóneo pero tras varias de pruebas de comunicación, se pudo determinar que la conexión *Bluetooth* no era muy estable. Los otros dos modelos, aunque el tiempo de autonomía era inferior, la comunicación con los dispositivos mostraba mayor estabilidad. Por otro lado, en lo que respecta a la configuración del sensor, solo en el modelo (a) permite configurar la frecuencia de muestreo de forma directa, mientras que los otros dos modelos, el cambio de este parámetro exige tener conocimientos específicos sobre cada dispositivo.

	Autonomía (horas)		Comunicación (tiempo de conexión constante)	
	Reposo	Trabajando	Conexión activa	Reconectando
SimleLink SensorTag	2.1	1.5	1h ~ 1.8h	±33sec.
Hexiwear	4.6	3.1	3.1 ~ 3.5h	±46sec.
Steval WeSU	6.2	4.3	0.1h ~ 0.5h	±83sec.

Tabla 9.2: Comparativa de los dispositivos utilizados.

Por otra parte, están los (ii) sensores individuales, los cuales necesitan un sistema de alimentación externo que los haga funcionar y de un nodo receptor que se conecte a ellos y extraiga los datos para que sean procesados luego por la unidad principal de proceso. En la Figura 9.3, se muestran los modelos utilizados en la prueba de selección del dispositivo final, esto son: (a) *Acelerómetro ADXL345*⁴, (b) *Acelerómetro +-400g*⁵ y (c) *MPU6051*⁶.

Aunque las características de estos sensores no favorecen el despliegue de un sistema poco invasivo, la principal ventana de este tipo de sensores es su bajo consumo de energía y su gran precisión a la hora de realizar las lecturas del entorno. Al ser un único sensor, han sido fabricados especialmente para obtener lecturas con un nivel de calidad bastante alto, tal y como ocurre con el modelo *Acelerómetro +-400g*.

⁴Grove: <https://www.seeedstudio.com/category/Sensor-for-Grove-c-24.html>

⁵Grove: <https://www.seeedstudio.com/category/Sensor-for-Grove-c-24.html>

⁶Chip principal: <https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>

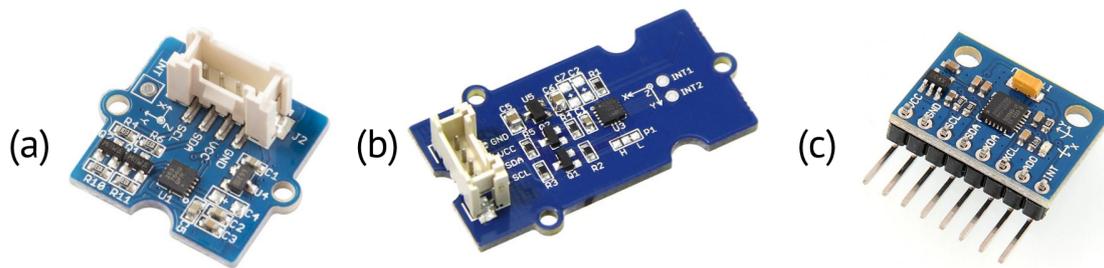


Figura 9.3: Sensores individuales utilizados en las pruebas.

9.4 Prototipos

Después de revisar las distintas opciones en dispositivos, el paso siguiente fue desarrollar una primera aproximación de lo que sería el sistema. Como se observa en la Figura 9.4, en esta versión inicial se utilizó el sensor *ADXL345* y como unidad de proceso, la *Beaglebone Gren*.

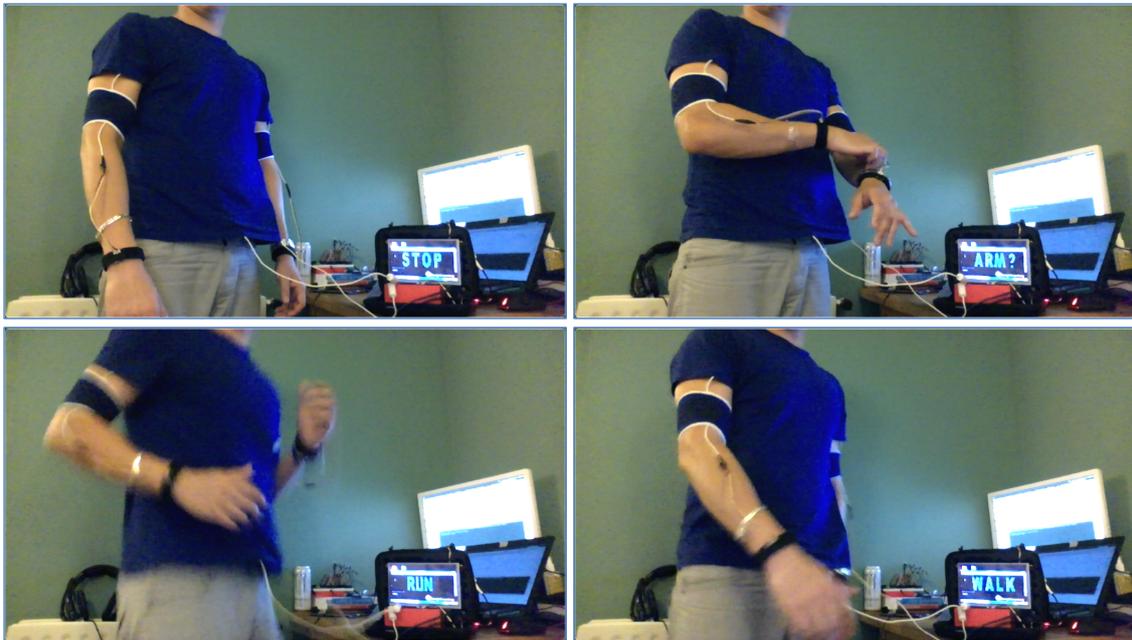


Figura 9.4: Primer prototipo del sistema de reconocimiento.

Esta aproximación buscaba probar el sistema de reconocimiento en personas, utilizando directamente un despliegue del framework sobre la unidad de proceso, sin tener en cuenta aspectos como la invasividad de la solución. Al ver que el sistema parecía responder adecuadamente al reconocimiento de las actividades predefinidas en algunos usuarios, lo siguiente fue mejorar la apariencia, la ergonomía y la usabilidad del prototipo, utilizando para ellos los dispositivos antes mencionados.

Como se observa en la Figura 9.5, en el proceso de desarrollo se pasó por distintas configuraciones de la solución. En un primero intento de desarrollo del prototipo (1) se utilizó un sensor individual de tipo *ADXL345* junto con una *Beaglebone Gren* como

unidad principal de proceso. Posteriormente, (2) el sensor fue reemplazado por el modelo *MPU6051* que demostró ser más preciso y permitía modificar su frecuencia de muestreo. El siguiente prototipo propuesto (3) empleó el wearable *Steval WeSU* cuya principal característica era que incluía una batería Li-Ion de 100mAh, un sensor de aceleración de altas prestaciones *LSM6DS3* y una *Master Controller Unit* (MCU) de 32-bit de baja potencia. Finalmente, (4) el prototipo final empleó el wearable *Hexiwear* que hacía las veces de smartwatch.

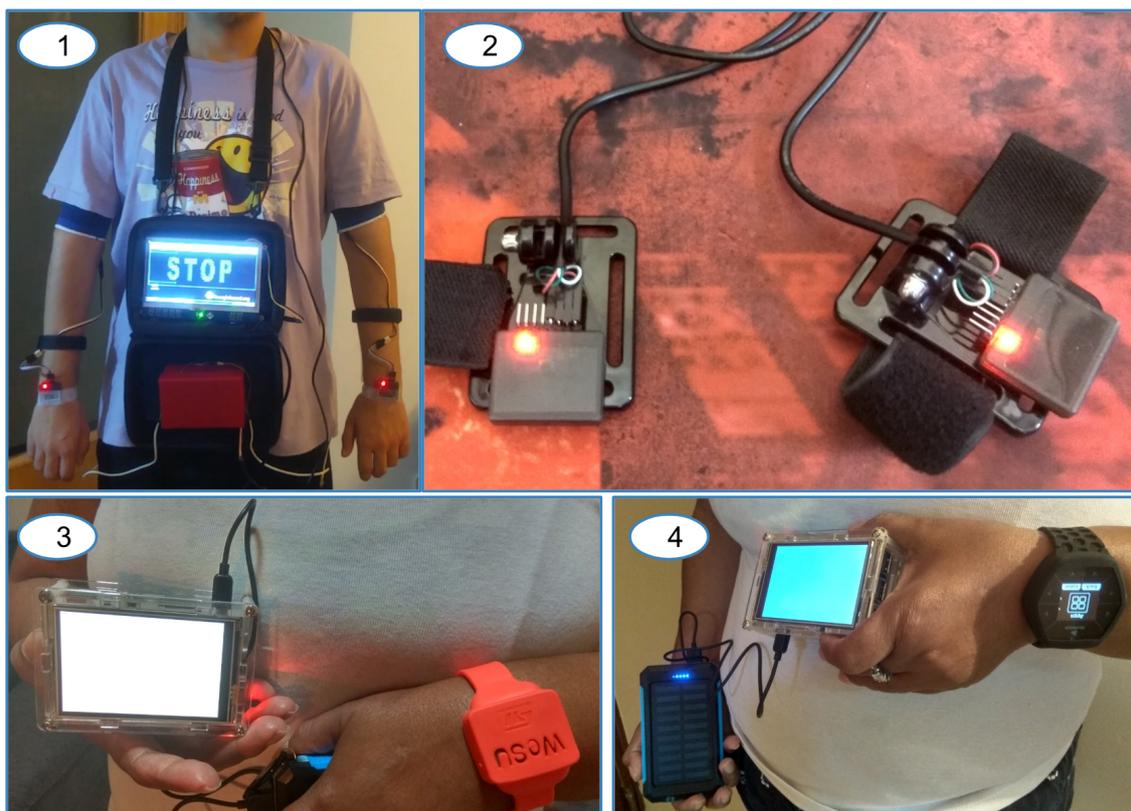


Figura 9.5: Prototipos del sistema de reconocimiento.

9.4.1 Prueba del sistema

Para evaluar el prototipo desarrollado, se ha propuesto validar el sistema contra un dataset⁷ público que contiene datos de aceleración de actividades físicas llevadas por usuarios reales, las cuales coincidían con las actividades simuladas en el caso de prueba. Se utilizó un conjunto de 323 ficheros con datos de 6 actividades, realizadas por 15 usuarios diferentes.

Cada fichero contiene 15 minutos del usuario realizando la actividad correspondiente, la relación fichero actividad se puede ver en la Tabla 9.3. Para las pruebas, las actividades se fueron suministrando al sistema de forma continua y aleatoria. Es decir, procurando no repetir las actividades entre una y otra. Adicionalmente, se han añadido dos actividades

⁷Dataset: <https://drive.google.com/drive/folders/1f1Golv6BImSH27Mlj9f7BAyMgUvLp0a5?usp=sharing>

aleatorias que están animadas en el personaje de la simulación, y que han sido capturadas de dos usuarios en un entorno real . Las actividades son “rascarse la mano” “*arm?*” y “rascarse la barriga” “*belly?*”, por cada una de ellas se han generado 10 ficheros con datos de aceleración siguiendo las especificaciones del caso de prueba.

stop	running	walk	wave-attention	drink	clap	arm	belly
72	16	48	90	86	10	10	11

Tabla 9.3: Número de datasets utilizados por cada actividad del sistema.

Al validar cada algoritmos del sistema con el dataset propuesto, se obtienen las matrices de confusión que se muestran en la Tabla 9.4. El criterio de evaluación utilizado en este caso, se ha enfocado en validar la precisión del reconocimiento de la actividad por parte del sistema. Para ello, se han calculado los indicadores *Precisión*, *Recall* y *F1-Score* para observar la efectividad del sistema en este otro entorno.

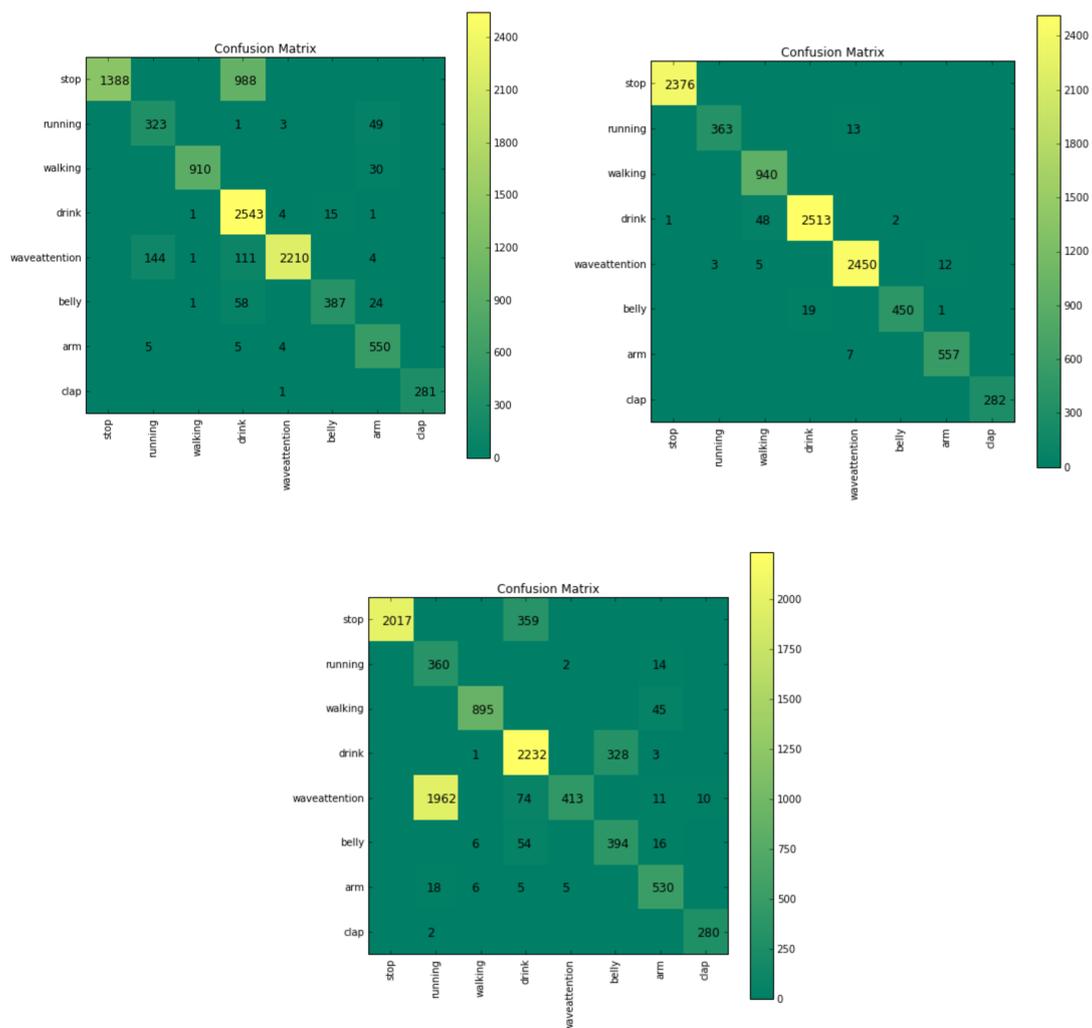


Tabla 9.4: Matriz de confusión de los algoritmos utilizados en el sistema.

Como se está utilizando más de un algoritmo para la misma tarea, este otro indicador *F1-Score* mostrará la medida de precisión que tiene la prueba para tenerla en cuenta entre los distintos clasificadores utilizados. En la Tabla 9.4 se observan los resultados obtenidos de la evaluación del sistema. Aunque la clasificación mediante RF no es la más efectiva en comparación a las otras dos, se deben indicar que como algoritmo, el uso de CPU de este algoritmo y el consumo de energía es bastante bajo por lo que su uso en el sistema se ajusta mucho a las especificaciones que se han descrito en el caso de estudio.

	RF			KNN			SVC		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
stop	0,85	1,00	0,92	1,00	1,00	1,00	0,85	1,00	0,92
running	0,96	0,15	0,26	0,97	0,99	0,98	0,96	0,15	0,26
walk	0,95	0,99	0,97	1,00	0,95	0,97	0,95	0,99	0,97
drink	0,87	0,82	0,84	0,98	0,99	0,99	0,87	0,82	0,84
waveattention	0,17	0,98	0,29	0,99	0,99	0,99	0,17	0,98	0,29
clap	0,99	0,97	0,98	1,00	1,00	1,00	0,99	0,97	0,98

Tabla 9.5: Indicadores de evaluación de los algoritmos utilizados en el sistema.

Ante los resultados que se muestran en la Tabla 9.5, un entrenamiento previo del sistema con la simulación, supone una precisión bastante alta a la hora de reconocer actividades como las propuestas. La simulación definió un escenario donde la actividad era modificada para asemejarla a la real y el resultado ha sido que, dichas variaciones se han podido utilizar para entrenar y enseñar a un sistema a reconocerlas. La variación de los resultados entre los distintos algoritmos refleja exactamente la situación que ocurre cuando los sistema de reconocimiento se entrenan en escenarios reales. Se sabe de antemano que algunos algoritmos se adaptan mejor que otros a determinados patrones de actividad, tal es el caso del kNN que se ajusta más a actividades periódicas cuando incluye la FFT como uno de sus características distintivas en los datos. SVC es igual de bueno pero aún en la simulación es costoso en tiempo y consumo de energía en dispositivos ligeros como los utilizados en este experimento. Actividades más complejas podrían ser simuladas y modificadas para construir sistemas que sean capaces de reconocerlas posteriormente en real. En una solución asistiva, el reconocimiento de la actividad juega un papel importante porque permite reconocer acciones del usuario al momento de interactuar con él. Modelar esta interacción es posible gracias a la simulación utilizada en la observación de escenarios de es tipo de soluciones.

En este punto, es posible darse cuenta que las simulaciones, como especificación que describe las situaciones de la vida cotidiana con un realismo suficiente, pueden ser usadas como un elemento que recoja requisitos concretos en un escenario de ayuda al usuario. Si ha sido posible validar la simulación con todos los elementos que la definen como una solución asistiva, es posible determinar que estas mismas condiciones del entorno simulado pueden ser trasladadas a un ambiente real y con ello conseguir que la solución propuesta en dicha simulación, funcione de igual forma en el mundo real. En algunos puntos de este proceso de evaluación, determinadas validaciones no mostraban con precisión lo que ocurría en la simulación. En realidad, esta situación no es un problema a la hora de usar

la simulación como especificación, porque realmente lo que se está consiguiendo con este formato de requisitos, es comunicar a los distintos usuarios aquellas características en las que se debe seguir trabajando para lograr que la solución sea efectiva.

El problema principal en los ambientes asistivos es precisamente identificar aquellas situaciones en las que, a largo plazo, pueden producirse problemas de interacción. Detectar estas inconsistencias a tiempo, difícilmente es factible utilizando solo los ambientes asistivos, porque no siempre es pueden modelar todos los comportamiento del usuario. Esto ocurre también con la simulación, pero el panorama para ir añadiendo nuevos modelos de comportamiento sobre lo simulado, es mucho más factible con el tiempo que, haciendo uso de usuarios reales para la experimentación.

9.5 Resumen

En este capítulo se llevó a cabo la ejecución de los casos de pruebas propuestos en el Capítulo 8, pero sobre un contexto real para el que se construyeron diferentes prototipos de tecnologías de apoyo. Estos dispositivos, se diseñaron a partir de las especificaciones de la simulaciones generadas con los casos de estudio propuestos. La experimentación implicó desplegar el framework sobre los prototipos creados, definiendo así pruebas unitarias que identificaran aspectos concretos del servicio que estos prestaban. El resultado de estas validaciones, permitió evaluar el uso de la simulación como especificación de una solución real, a partir de cuantificar los criterios de éxito cumplidos en cada escenario simulado, que posteriormente se corresponderían con los resultados de las pruebas en real.

Capítulo 10

Conclusiones y trabajo futuro

10.1 Contribuciones

Los aportes principales de esta Tesis se han orientado a demostrar que la simulación 3D puede ser usada como un elemento de validación temprana, que permite decidir si el desarrollo de una solución asistiva funciona como se espera o no. Para ello, se ha propuesto que el esfuerzo aplicado en el proceso de definición y construcción de la simulación, sea aprovechado a lo largo del ciclo de desarrollo de la solución para conseguir una mayor calidad de la asistencia a un coste menor. Siguiendo este planteamiento, las principales contribuciones han sido definir una metodología que permita, por una parte, caracterizar el comportamiento de la solución asistiva en la simulación y, por otro lado, construir los entornos de prueba que faciliten la validación bajo el marco conceptual de las simulaciones.

Para el entorno de pruebas se ha propuesto una arquitectura basada en eventos cuyo propósito es facilitar la ejecución de los casos de prueba que se definan con la metodología. Finalmente, se ha hecho una experimentación donde se muestra cómo se aprovechan la metodología y el entorno de pruebas, para determinar el valor que aportan las simulaciones como elementos de especificación en las soluciones asistivas.

En resumen, los principales aportes de esta Tesis son los siguientes:

- La metodología que permite caracterizar la solución asistiva en la simulación, descrita en el Capítulo 3, que se basa en la definición de casos de estudio que utilizan los requisitos de la solución para modelarlos en el comportamiento de los elementos simulados: “*usuario*”, “*entorno*” y “*tecnología*”. Para construir los casos de estudio, los diferentes usuarios de la solución participan en sesiones de trabajo cocreativo, donde identifican los requisitos y establecen los criterios de validación que aplicarán para su verificación. Asimismo, se fijan distintos escenarios simulados donde dichos requisitos son observados y comparados mientras son visualizados en la ejecución de la simulación. Como resultado de aplicar la metodología, se obtienen las simulaciones que representan el caso de estudio y que especifican los criterios necesarios para validar cada simulación. Las técnicas utilizadas para desarrollar las sesiones de trabajo han sido publicadas en [BMCB18] (técnicas de cocreación) y en [CBCGSP18] (técnicas de participación). El proceso de identificación de requisitos de actividades

en entornos inteligentes se publicó en [GSPACB17].

- La plataforma web *Hack4People* utilizada para desarrollar las sesiones de trabajo con los usuarios en la definición del caso de estudio. Aunque esta herramienta inicialmente no hacía parte de los logros propuestos en esta investigación, es un aporte importante que ha contribuido en el proceso de identificación de requisitos.
- Los casos de prueba necesarios para validar las simulaciones, los cuales se describen en el Capítulo 4 y hacen parte de la metodología propuesta. La validación depende de las pruebas unitarias que recoge y organiza el caso de prueba. Las pruebas unitarias se aplican utilizando criterios preestablecidos sobre la simulación, relacionados con los datos, las actividades del usuario, el estado del entorno y la participación de la tecnología de apoyo. Los elementos de la metodología relacionados con los casos de prueba han sido publicados en [CBSP18].
- El framework propuesto que brinda la infraestructura necesaria para crear el entorno donde se ejecutan los casos de prueba sobre la simulación. Para construir estos entornos se propuso una arquitectura de software basada en eventos EDA que ayuda en la definición, creación y ejecución de los casos de prueba. Para ello, se prioriza el procesamiento de eventos y se utiliza como el elemento central de la arquitectura y del caso de prueba. La gestión del volumen de datos y su procesamiento se lleva a cabo con un motor de reglas CEP utilizado como modelo base para la toma de decisiones basada en eventos. Esta arquitectura ha sido publicado en [CBSP18] y algunos de sus componentes en [GMIJ19] (mecanismos de pruebas con sensores), [PCBGSP17] (virtualización del sensor), [PGSOCB18] (servicios en la simulación) y [SOCBGSP18] (configuración y parametrización de escenarios).
- Como complemento del entorno de pruebas, se construyó un sistema especializado en el reconocimiento de las actividades humanas que se integra con el entorno de las pruebas de la simulación, descrito en el Capítulo 7. El sistema se centra en el reconocimiento de las actividades o “animaciones” que realiza el “usuario” dentro del “entorno” simulado. Su propósito fue incluir el reconocimiento de actividad como una técnica que permita identificar la interacción del usuario con el entorno mediante las acciones físicas. Para que el uso de la simulación fuese posible, cada animación es modificada por el sistema antes de ser usada en el escenario, con el fin de evitar que esta sea poco realista y muy repetitiva. El objetivo de la modificación, es que el algoritmo de aprendizaje no sobreajuste al momento de entrenar con las animaciones. El primer prototipo del sistema de reconocimiento ha sido publicado en [GSCBPC16].
- La experimentación con casos de prueba planteada en el Capítulo 8, la cual se ha llevado a cabo con el propósito de mostrar la utilidad que tiene la simulación en la validación de soluciones asistivas. Se ha construido un entorno para ejecutar los casos de prueba obtenidos a partir de tres casos de estudio concretos, donde se observaron diferentes necesidades del usuario que fueron cubiertas total o parcialmente por una solución asistiva. El resultado de las pruebas, se ha utilizado como criterio para validar la simulación.

- La experimentación en un entorno real descrita en el Capítulo 9, siendo esta un reto que ha supuesto un mayor esfuerzo debido a las múltiples pruebas que se hicieron con diferentes dispositivos del tipo ordenadores de placa reducida y sensores. En dichas pruebas no solo se buscaba la compatibilidad del framework con el hardware, sino que además se incluyeron características en la configuración que permitiesen optimizar la autonomía del dispositivo, la portabilidad del mismo y el rendimiento de la solución una vez puesta en marcha.

10.2 Líneas de investigación abiertas

Durante el desarrollo de esta Tesis se han ido identificando distintos aspectos que no han sido abordados debido a su complejidad, los cuales, dada su relevancia, se han dejado como posibles líneas de investigación abiertas. En lo que respecta a la metodología para caracterizar la simulación, los aspectos que se han identificado son los siguientes:

- Es costoso modelar características del usuario y del entorno que describan aspectos concretos de su comportamiento, como por ejemplo los problemas de movilidad que evolucionan o se deterioran con el tiempo, la interacción con objetos dotados de mayor funcionalidad en el entorno, entre otros. Investigar y aplicar mecanismos para incluir estos modelos en la simulación podría ser útil para representar escenarios más realistas y complejos de soluciones asistivas.
- En el ámbito asistivo, pese a la importancia que tienen situaciones como la aversión tecnológica y los problemas de respuestas no deseadas del usuario, no existen estudios específicos que describan de qué manera abordar este tipo de situaciones. Sería interesante definir una metodología que guíe a los distintos interesados en la construcción de las soluciones teniendo en cuenta este tipo de problemáticas.

En lo que respecta a la construcción de escenarios de simulación, la creación de objetos visuales que, además de enriquecer al entorno y a los personajes, permiten validar características del contexto asistivo, implica muchos costes que dificultan la generación de ambientes más realistas. En la presente Tesis, se crearon nuevos personajes y escenarios relacionados con la problemática tratada en los casos de estudio. Sin embargo, sería conveniente desarrollar herramientas alrededor de los sistemas AAL que permitan usar la simulación para observar características concretas que puedan ser validadas desde la misma simulación. Para la creación de estos objetos, se han identificado las siguientes necesidades:

- Los casos de estudio contemplan el uso de elementos en la simulación que deben tener una representación fácilmente modelable a partir del entorno asistivo. Facilitar la generación de estos modelos con ayuda de los profesionales correspondientes, es una tarea que contribuirá con la escenificación de situaciones concretas que permitirán validar tanto el alcance la solución final como su aplicación en un entorno real.

- Los procesos de modelado de las actividades físicas de usuarios mayores debería contar con galerías de actividades comunes llevadas a cabo por usuarios con diferentes limitaciones físicas. Incluir en el proceso de modelado a este tipo de usuarios no es conveniente y en muchos casos, es imposible. Para ello, sería oportuno definir técnicas que permitan recoger estos comportamientos de la manera menos invasiva posible y con un nivel de realismo suficiente que permita la experimentación posterior.

Con base a las líneas de investigación descritas en el apartado anterior, esta Tesis deja abierta las líneas de trabajo cuyo propósito general será mejorar la construcción de entornos de prueba de simulaciones más realistas. Los trabajos que se han identificado para tal fin, son los siguientes:

- Añadir nuevas características en el comportamiento del usuario y del entorno para que estos sean modelados de forma más realista en el caso de estudio.
- Incluir especificaciones que describan la forma de modelar tecnologías de apoyo en entornos simulados, con el fin de observar su interacción con otros elementos de la simulación.
- Añadir nuevos sensores y actuadores que amplíen la descripción del entorno. Con respecto al usuario, se necesitan nuevos sensores inerciales que cuenten con diferentes especificaciones técnicas que permitan experimentar con el reconocimiento de la actividad física afectada por los problemas de movilidad del usuario.
- Ampliar la funcionalidad del sistema HAR para que pueda utilizarse ante escenarios con actividades muy diversas.

Part II

Description of the research

Chapter 11

Techniques for the quantitative evaluation of 3D simulations

11.1 Introduction

The European population faces a series of critical social changes which alter different areas of its current structure. The economic system, the health services, the labour sector, the family structure, among others, have been affected by the progressive tendency of society to age, which is estimated to continue until 2080. As shown in Figure 11.1, according to Eurostat [Eur18] statistics, the expectation is that in 2045, the total population will be close to 530 million people, 4% more compared to 2016. Although the population's tendency to increase is normal, the problem lies in the fact that the growth is not being homogeneous and balanced within the different age groups.

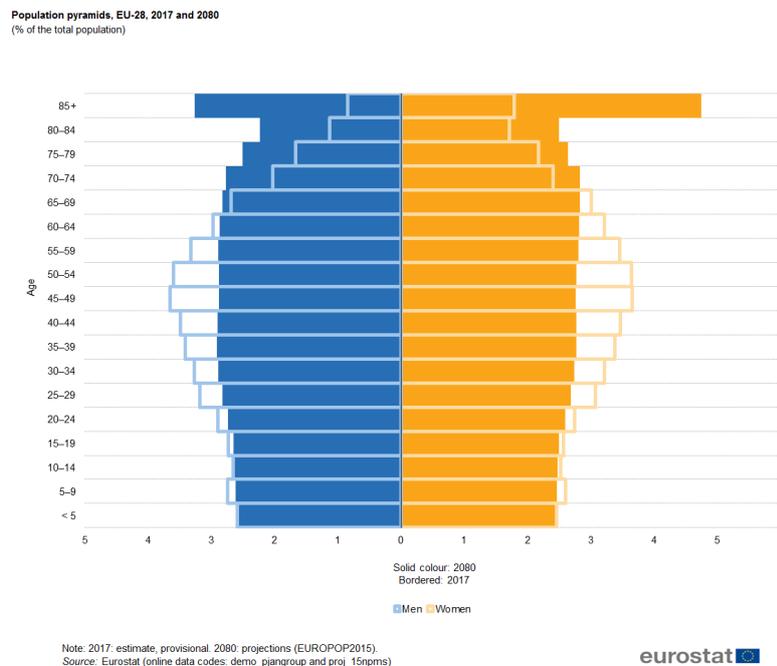


Figure 11.1: Trends in the ageing of the population for the year 2080.

With the ageing of society, other risk factors increase related to the suffering of a chronic and degenerative disease, which prevents older people from living independently.

This type of problem affects not only the quality of life of the individual itself but also produces changes in behaviour and family structure, which is widely affected by the situation of dependence that is generated with this type of disease. At the social level, if the elderly population becomes ill and is not able to sustain itself, in the end, all this translates into more expenses destined to provide more significant social assistance.

Against this background, the attention is focused on health care services. The growing segment of elder population predicts an increasing demand for care and monitoring systems. Such systems should be able, in the long term, to be accessed from home and to meet the specific needs of patients, while still taking into account the risk factors each person might be facing in their own context. Similarly, it is also necessary to assign and train more health professionals with sufficient knowledge in areas such as primary care, gerontology, geriatrics and especially in the use of new technologies that support health care. In this regard, the *World Health Organization* (WHO) recommends that European health systems [O⁺11] have a constant training mechanism in the field of health care for the elderly, especially in advanced ages where health care is most critical [HRH⁺09].

The WHO defines long-term care [WHO18] as “The set of activities carried out by both informal caregivers (relatives, neighbors or friends) and professionals (specialists, social workers) to ensure that a person who can not take care of themselves, maintain a quality of life suited to their individual preferences, with a high degree of possible independence, autonomy, participation, personal fulfillment and human dignity.” In this definition, both caregivers and health professionals have the same importance in the assistive field. However, in cases where the caregiver is a member of the family, their role is even more relevant because, in general, this changes the behaviour and habits of entire families, which are forced to adapt to the new attention the user demands. The demand for caregivers is increasing and this is mainly due to the fact that investment in social infrastructure continues to focus on the services of institutions and not on the creation of strategies that directly mitigate the strong demand of an ageing society. In view of this situation, it is necessary to devise plans that make informal care times converge (caregivers) with the investment of public spending so that services are planned with sufficient guarantees for health systems to cover long-term care in a sustainable manner.

11.1.1 Assistive technology in the elderly population

To cover the current and future needs of society, paradigms such as AmI, there is the need to promote the creation of technological, environmental solutions that allow users to interact with the environment and that it provides them with a particular service. Under this same idea, there are disciplines based on daily life assisted by the AAL [RM13] environment that focus especially on problems related to the ageing of the population offering alternatives that allow users to be assisted by the environment according to their preferences, characteristics or limitations.

In this technological scenario, it is important to have a large number of devices that allow collecting as much data as possible, not only from the context but also from the user. With all the information obtained, it is much easier to detect, predict and act according to the daily needs of the user, whether inside or outside the home. The growing use of the Internet of Things or IoT [WADX15] has contributed to the increase in the number of devices that, through sensors, capture information from the environment and subsequently are able to publish this data so that other devices consume information, make decisions and act on according to the environmental conditions. Taking advantage of this capacity of connection between devices and its wide presence in the environment, AAL aims to improve the quality of life in people [RSL⁺18], automating everyday tasks

of the environment inside and outside the home, also creating context-sensitive services to offer certain assistance right at the moment in which the user needs it.

This is how AmI and AAL make their contribution in the field of assistance, providing the environment with intelligence everyday objects that integrate sensors and processors to gather information from the context and thus adapt to the needs of the user in a transparent manner [SCSE10]. The context information also includes the activities performed by the user [ABC⁺16], which are analyzed and interpreted through the HAR systems [DOB⁺16]. In this type of solutions, a HAR system is an important piece that allows identifying not only the actions of the user [CFR14] but also the different ways in which this could interact with the solution [Har09] and the environment [ROGP⁺13].

11.1.2 Barriers in the construction of assistive solutions

The construction of an assistive solution is not an easy task to carry out. To define a solution of this type, it is necessary to identify any variables that condition its development, especially those whose final solution consists of the following elements:

- The solution or service that you want to provide to a certain user.
- The software that controls the solution [HCL⁺12].
- The hardware that will allow the user to interact with the final solution. [BE02].

Regarding the software, the first need that arises is to identify the development methodology which allows its construction and validation. This is of utmost importance, given the fact that the software project is created for support purposes. Different methods contribute to specific guidelines to create assisted solutions. Among many of its recommendations, it is clearly described the need to use models that represent and simplify the requirements of the solution, in order to be able to experiment with it. The main focus of AAL is based on the use of real controlled environments for experimentation, known as living labs [KOA⁺99]. The main drawback in these type of systems is that they require considerable economic investment for the construction of laboratories, which in many cases can be equivalent to building prototypes of the expected solution at costs that are difficult to predict and subsequently not affordable for the end user.

According to the SWEBOK, prototyping, modelling and simulation (see Figure 11.2) are very useful techniques in projects where the process of capturing requirements is very important, as well as the early visualization of the behaviour of a system from a specific perspective [BF⁺14]. However, each of these techniques have particularities that must be taken into account at the time of being used in any type of development.

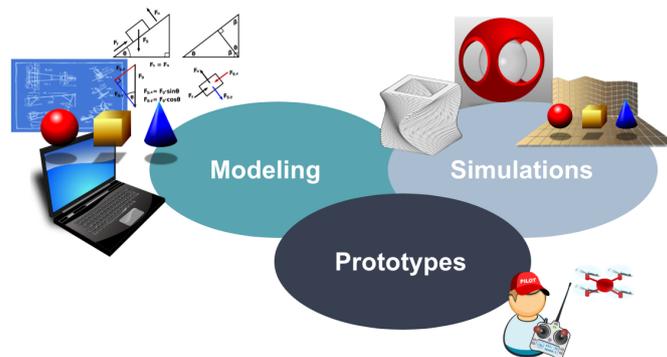


Figure 11.2: Techniques for the study of the system's behaviour.

In the case of prototypes, the aim is to capture an early version of the system [BF⁺14] so that designers can verify the viability of the assistance service during the design of the system. The prototype can be used in many phases of development, but depending on how complex it can be to define assistance, they are more useful in the process of capturing requirements, in the construction of interactive elements or in the validation processes of functional requirements. Despite its many advantages, using this abstraction requires an investment of time and money for its construction, which translates into an increase in both manufacturing and implementation costs at the hardware level. Modelling, on the other hand, is used to abstract part of a process and thus identify the characteristics of the solution. These are taken advantage of by the simulations, which use these characteristics to facilitate the execution of experiments that allow, among others, to identify the behaviour and relationships of the system, to study aspects of the design. Through these techniques, it is also possible to build theories or hypotheses about how the system could behave and based on that, create predictions about the final solution.

Although simulations have been a classic element in the development of systems by models, due to their nature, they only model certain aspects of the system and that is why they have not transcended beyond the use they currently have. A field in which they have had success is in CAD/CAM [ZS91], which through computer-assisted work, have facilitated CAD design and CAM manufacturing solutions [WZ02].

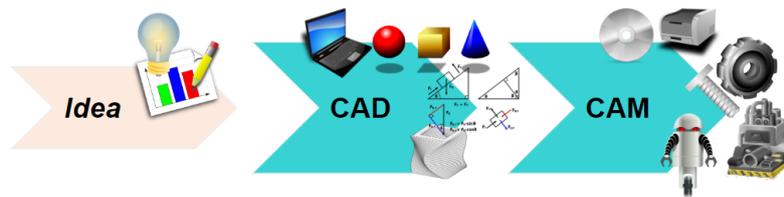


Figure 11.3: Computer-aided design and manufacturing.

As shown in Figure 11.3, the simulation is present as in a representation of the solution [NSK05], in which starting from an idea, the different components that constitute a product are designed and modelled. In this stage of design and modeling, the developer establishes the parameters in each simulation and adjusts them according to the requirements that are defined. When this representation of the product allows to observe and evaluate its components, the next step is to use the simulation in the manufacturing processes to build the different physical elements that make up the solution.

In an assistance solution, these abstractions can be used to generate part of the solution. However, the general definition of these representations of the problem [VNKC09], may not help to make the solution cheaper due to the time that must be spent on the construction of the different models. That is why this alternative, in the long term, slows down both the experimentation and the validation processes of the solution. On the other hand, there are abstract features in assisted solutions that can not be easily included in these elements, for example, actions and possible user responses to the solution, cases of technological aversion, adaptation of assistance to the environment and the evolution of the patient, among others.

The simulation has undergone a series of structural changes that have promoted its participation in system modelling processes. In the case of the Environmental Intelligence, tools such as SociAALML [CSGS15], through a graphics and domain-specific language, allows users who are not experts in programming to create 3D simulations that represent either a problem or a solution. SociAALML generates the necessary source code to execute

the simulation, so the designer should only worry about defining the solution and not about aspects more specific to programming. Despite all this, the simulation does not stop being a three-dimensional visual representation that reproduces pre-recorded animations in a computer [BBET97], which makes it necessary to have enough time to study and evaluate both its content and the behaviour of the different elements that compose it.

In this type of development philosophy known as VLL [GSCBPC16], 3D simulation plays a very important role as a support tool to achieve the construction of the solution, using for it, design cycles and conventional software development. The simulation represents a prototype of the assisted solution, in which all the actors are incorporated: the environment (e.g. a dwelling), the users (e.g. patients, doctors, caregivers, relatives), the sensors (e.g. wearables, smartwatch, appliances).

At this point and after involving 3D simulation in the field of development of assistive solutions, the challenge that arises is to establish how this simulation can promote the participation of the different actors in the definition of the solution, especially in those cases where the user's participation is limited in some way. To achieve this, artifacts that are understandable by all participants and that go beyond collaborative aspects and improvement of a real hardware implementation of what is developed in the virtual environment are required. Likewise, there is a need to define a methodology that allows validating the simulation when it is included in the development process as an element that collects specifications of the solution and is used to show an early representation of what could be the final solution or a part of it. From the point of view of the requirements, as well as it is necessary to define techniques that evaluate each requirement, likewise a test environment must be defined to determine if the requirements represented in the simulation are those expected by the end user and by the experts in the assistive solution. Similarly, the change in requirements during development should also be taken into account. With each new iteration, the simulation can be modified and the testing environment must be able to adapt to these changes, thus ensuring that requirements already validated in past iterations are not lost.

Regarding the use of simulation as an early visualization element, its 3D visual format must be taken into account when being evaluated as an assistive solution. The validation technique used should minimize the time of "visual inspection" by the experts, especially when dealing with a solution that is represented by simulation hours.

11.1.3 Approach of the work and objectives

As described above, the definition of assistive solutions involves many pieces that must be taken into account for their development. Similarly, many people can participate in the capture of requirements of the final settlement. Many works in the literature specifically describe the construction of assistive solutions, especially in the industry sector. However, there is no clear standard for how all existing resources can be used to design, build, and validate a solution that provides a particular support service.

The use of 3D simulation as a critical element in the definition of the assistive solution does not solve the problems that exist in the design and validation of the assistance. Regardless of the development methodology used and even with the use of simulations proposed by the tool SociAALML, it is necessary to establish a mechanism that ensures experts that the simulation meets the requirements that define it and can be used as an indicator of the behaviour of the solution in a real environment.

Assuming as a hypothesis that simulations are part of the development cycle of the solution and that they are used as an early representation of its behaviour, the approach of this Thesis is to propose a testing environment that can be used under the conceptual

framework of simulations. In this framework, the simulation is used as an active element in the capture of requirements and the validation of the final solution. As can be seen in Figure 11.4, in this context, the solution tests can be established in two specific phases of development:

- In its definition, which has to do directly with the requirements and their acceptance.
- In system tests, where it is possible to couple a real technological solution in the simulation, thanks to the infrastructure available in the tool.

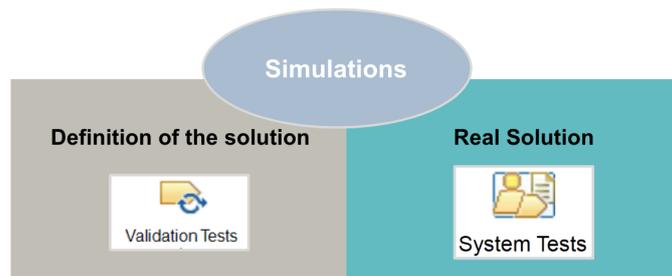


Figure 11.4: Participation of the simulation in the development framework.

The test environment in the definition phase mainly uses the simulation itself. Its purpose is to determine if what is expected from the solution is represented in the simulation, if it includes the specific problem to be solved, if the system responds to this representation and if the expectations expected by the user are fulfilled. Regarding the test environment with the real solution, the purpose, in this case, is to couple parts of a real solution directly with the simulation. The purpose is to be able to check if the simulated requirements satisfy both the user's expectations and the characteristics of the solution as a product that can be implemented from a real environment. These phases can be seen in Figure 11.5.

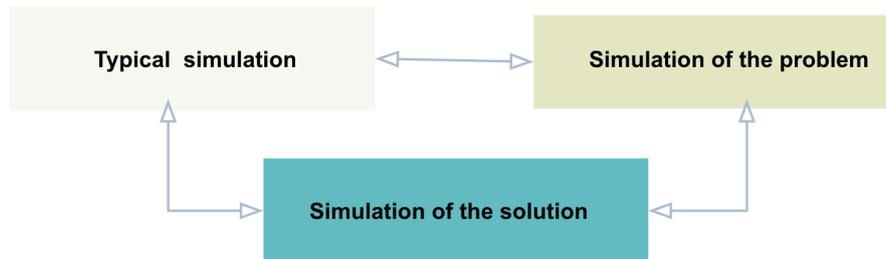


Figure 11.5: Description of the behaviours of the system represented by the simulation.

In both environments, the simulation is expected to represent the typical behaviour of the system; the behaviour picked up by the problem and the expected or correct behaviour. With this representation, the starting behaviour is clearly described on the one hand, which includes the daily situation that is affected by an illness or ailment; and on the other hand, the solution. At this point, assistance is what differentiates the transition between the two behaviours, allowing it to act as the control solution that guides engineers in building a final solution.

Under normal conditions, this dynamic would be complex to carry out because it would be necessary to assume a series of costs related to the rental of facilities and the purchase of hardware or other elements, which allow observing different behaviours of the

solution from different versions of it. In other words, it would be necessary to build several prototypes of the solution to observe how it behaves in the different scenarios proposed. With the proposed methodology, this is not necessary because it is possible to use the simulation directly to define the final solution, both at the software and hardware level, and to check whether the simulated solution is close enough to reality.

To comply with this approach, the following specific objectives are proposed in this Thesis:

1. Establish the use of simulation in 3D as a prototyping process to represent the construction of an assistive solution. To do this, the simulation collects the characteristics that define the assistive solution in a real environment, which is why the simulation always try to represent case studies as realistic as possible.
2. Define a procedure to identify and analyze the behaviour of the different elements that make up the simulation and define it as a simulated solution.
3. Facilitate the creation of test scenarios to validate and verify the behaviour of the elements in the assistive solution, which generally be represented through the use of virtualized electronic devices such as sensors and actuators that collect all information from the environment, including the user.
4. To implement an architecture that allows defining programmatically, a test environment that facilitates the verification of an assistive solution based on simulations for its final acceptance. The same test environment can guarantee to the developers that, during the development phase of the solution, with each new simulation as a prototype, functionalities that have already been validated or verified are not lost.

11.1.3.1 Thesis outline

This Thesis work has a total of 10 chapters, and this introduction is the first of them. The remaining chapters are briefly presented below.

- **Chapter 2: State of the art.** Describes the basic concepts and state of the art of the hypothesis addressed in this thesis, namely, solutions represented by simulations in 3D and techniques for validating this type of specification elements.
- **Chapter 3: Characterization of the assistive solution in the simulation.** This chapter presents the methodology proposed for using simulation as a specification element in the development of a case study that includes an assistive solution. It also details the main tasks of the case study development process and the artifacts generated, necessary to build the simulation.
- **Chapter 4: Simulation Test Environment.** This chapter indicates the procedure for building the test environment that validates the requirements of the assistive solution with the help of simulation. Step by step, the process is explained that identifies the elements of the solution that are necessary to define, execute and analyze the applicable test cases on the simulation. Finally, based on the results obtained, the objective is to show how to validate the solution or, in its absence, how to verify if the case study represented by the simulation corresponds to the simulation.
- **Chapter 5: Framework Design.** It presents the proposed framework for the creation of the test environment that serves as a guide for developers in the process of validation of simulations.

- **Chapter 6: Testing Environment Architecture.** Based on the design of the framework, it describes the architecture whose functionalities will be applied both in the creation of unit tests on simulation elements and in the definition of complex test cases.
- **Chapter 7: Implementing the system HAR.** Shows the specific component of the framework in charge of validating the user's physical activities by recognising them. The system shows the elements that make it up and how they fit with the proposed architecture.
- **Chapter 8: Experimentation with the test environment.** This chapter details the experiments carried out to show the use of the methodology and the proposed architecture. On the one hand, the experimentation has a test environment to analyse the specific behaviour of the simulation elements. On the other hand, a second test environment validates the user behaviour within the simulation, with the help of HAR techniques that recognise the patterns associated with the user responses.
- **Chapter 9: Testing in a real environment.** The experimentation uses the real environment to validate specific characteristics that exist in the simulation, mainly related to the implemented system HAR.
- **Chapter 10: Conclusions and future work.** Finally, this chapter shows the conclusions obtained with the elaboration of this Thesis and presents the different lines of future work that arose throughout the research work.

11.2 Characterization of the assistive solution in the simulation

Simulation is useful in an AAL solution project when it can be part of the requirements engineering process within the corresponding process model. All efforts to generate the requirements specifications are used by the engineers to build the simulation and include in it the requirements that represent the behaviour of the assistive solution. In this type of solutions, the contribution offered by simulations in 3D within the process model focuses on two specific contexts:

- A visual specification of requirements that evolves as the requirements identification phase progresses.
- A device that helps create the validation that will ensure that the assistive solution meets the customer's requirements.

Regardless of the context, simulation is present throughout the requirements engineering process. This feature is a positive aspect because it prevents engineers from implementing additional techniques to build the simulation. As can be seen in Figure 11.6, simulation appears primarily in two specific phases of the software life cycle: in the solution definition phase and the system testing or solution implementation phase.

11.2.1 Model the assistive solution

The simulation models the behavior of the solution elements: *“user”*, *“environment”* and *“technology”*. It is necessary to build at least three scenarios to analyze the different problems of the solution, where the previous behaviour models are observed from different points of view. The three scenarios are as follows:

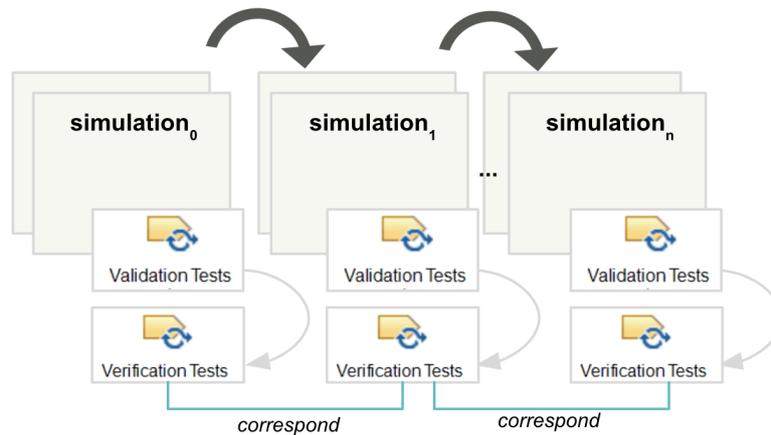


Figure 11.6: Correspondence between the different iterations of the simulations.

1. The *first type of scenario* represents a typical or everyday situation for the user or the environment, where no anomaly occurs that requires some assistance.
2. The *second type of scenario* identifies and describes a problem that the user suffers as a consequence of his physical condition or the characteristics of the environment during a scenario such as the 1.
3. The *third type of scenario* shows the assistive solution that can be given to the user starting from a scenario such as the 1, which can unfold as the scenario 2.

As can be seen in Figure 3.3, from the *first type of scenario* the *typical or normal simulation* is created, which describes how the ideal scenario is for both the user and the environment without any technological support.

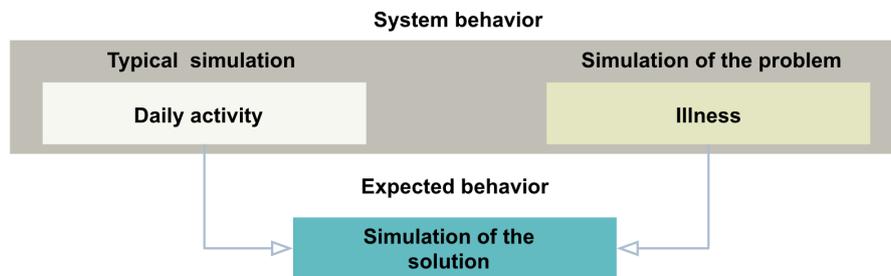


Figure 11.7: Representation of system scenarios.

With the help of the *second type of scenario* the *simulation of the problem* is created, thus describing a scenario similar to the previous one but with the problem that has the environment, a user or both. The objective of this simulation is to allow the observation and analysis of the situation that prevents the user from carrying out any daily activity. The *third type of scenario* generates the *solution simulation*, which describes how the user assistance will be carried out, either through services that are adapted to the environment or using some assistive technology, which improves the user's capacities or facilitates their interaction with the solution.

11.2.1.1 Elements to build the simulation

The necessary elements to move the assistive solution to simulation are the *study case* and the *solution tests*.

- **Case study:** this is the starting point for building the solution. In it, the process is carried out that captures the scenarios of interest to create the corresponding simulations later. In this process, all the specifications describing the characteristics of the solution are collected, as well as how the behaviour of the different elements will be modelled. Other aspects covered are the aesthetic and appearance details of the simulated environment, the actions or activities that occur in the simulation, the technical details that describe the assistive technology that will provide assistance, the configuration and parameterization of that technology, among others.
- **Solution Tests:** collect the specification of the simulation tests, including guidelines describing how these tests are implemented at the code level. Depending on the phase where the tests are executed, there will be two groups:
 - *Validation tests:* the requirements of the scenario and the result of the simulations are compared to corroborate that the simulated corresponds to what is indicated in the case study. If these tests are correct, it is assumed that the simulation meets the established requirements and therefore represents the problem being analyzed in the case study. In the opposite case, and depending on the validations that failed, it is necessary to review in greater detail the scenario and the simulation to identify the origin of the inconsistencies within the case study.
 - *Solution tests:* these tests are to check if the data generated by the assistance in the simulation meet the needs covered by it. The simulation provides mechanisms that allow external programs to access the different interactive elements of the solution, such as sensors, actuators, user, among others. Through the behaviour of these elements, the tests validate that they comply with the functional requirements established to provide assistance or contribute to it. If the result of these tests is correct, the general configuration of the solution, e.g. plans, parameterization, characteristics of the environment, sensor network, among others, can be used as a reference to implement the final solution in a real environment.

11.2.2 Case study definition

The construction of the case study needs all the information obtained from the different sources of requirements available in the development model implemented in the project. According to SWEBOK, the sources of requirements common to a traditional development model that apply equally to assistive solution projects are users, initial requirements, scope and work plans. The following is to organize the information into specific software requirements specifications that describe, among other things, the user needs, the helpdesk and the definition of the simulations that are needed to represent those specifications, as shown in Figure 11.8.

11.2.2.1 Transition from case study to simulation

The transition between case study and simulation requires specifications, defined in the case study, that describe the scenarios previously: a scenario with a typical or daily situation of the user, a scenario with a situation in which the user faces the problem and the last scenario with a situation in which the proposed solution provides assistance (see Figure 11.8). It is not an indispensable requirement to define precisely three simulations;

the number of simulated scenarios depends on how complex the case study is or on the team’s need to assess each scenario separately.

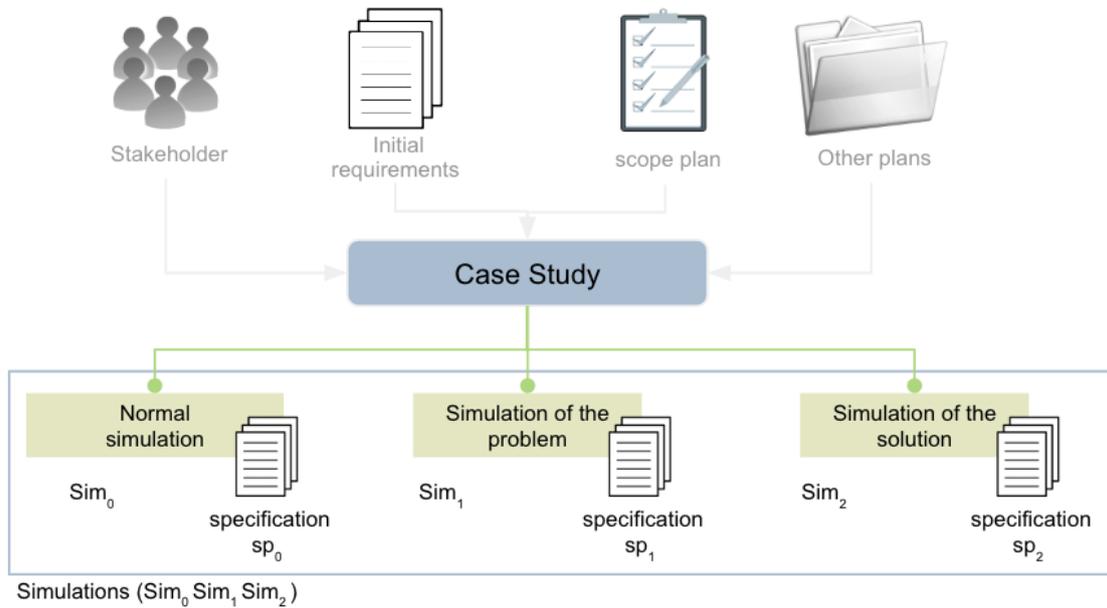


Figure 11.8: Case study specifications to generate the simulation.

11.2.2.2 Characteristics of the user to be modelled

The characteristics of the user who models the case study are related to profile information, for example, age, gender, behaviour, physical characteristics and location within the environment. Behaviour is modelled by the physical activity that the user performs, which may represent simple movements or gestures such as *raising or lowering hands, extending the arm, flexing legs, episodes of hand and neck tremors*. Likewise, the user has associated more complete actions that are predefined programmatically, such as “*drinking*”, “*eat*”, “*run*”, “*jump*”, “*walk*”, “*call for attention*”, “*fall*”. A summary of the characteristics of the user to be modelled is in Figure 11.9.

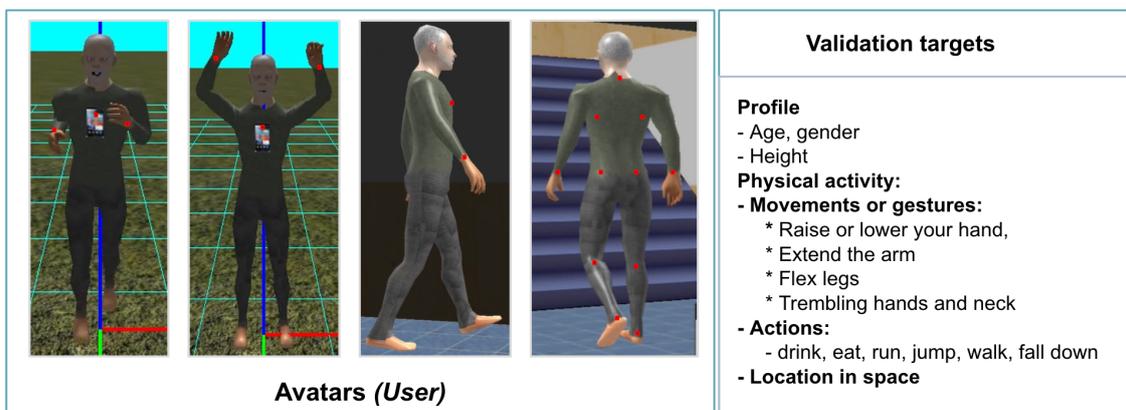


Figure 11.9: Aspects to validate in the “user” element of the simulation.

11.2.2.3 Characteristics of the environment to be modelled

The elements of the environment are organised in two groups: on the one hand, there are the *decorative* or *static* elements whose only role in the simulation is to adorn and decorate the environment, e.g. *paintings, flowers, carpets*, among others. On the other hand, there are the *dynamic elements*, which do have interactive features with which the user and other objects can interact, e.g. *television, chairs, doors, tap, bed*, among others. The dynamic elements, on the other hand, have sensors that allow their monitoring, regardless of whether the user carries them in the simulation or the elements of the environment. In the Figure 11.10 the elements of the environment are observed.



Figure 11.10: Aspects to validate in the “environment” element of the simulation.

11.3 Creation of simulation test cases

Chapter 3 shows how case studies of assistive solutions are defined and how simulations are generated from them that will be used to specify the solution. In this chapter, the goal is to build the necessary test cases to validate that the simulation meets the characteristics of the solution assigned to it. In the validation of the simulation, the test cases are applied according to the scope of the assistive solution, described in Section 3.2. In the solution definition phase, the simulation **test environment** is defined, where the test cases that

validate the simulation are executed. The test phase of the system acts on a **real test environment** and validates that the simulation can be used as a final specification to guide the implementation of the solution in a real environment. This second test environment is only possible if the simulation and the case study have gone through the validation of the definition phase before.

11.3.1 Building the test environment

As shown in Figure 11.11, the test environment is composed of three main elements: the case study and the associated simulation, the unit tests that validate each requirement, and the test cases that group and organize the unit tests.

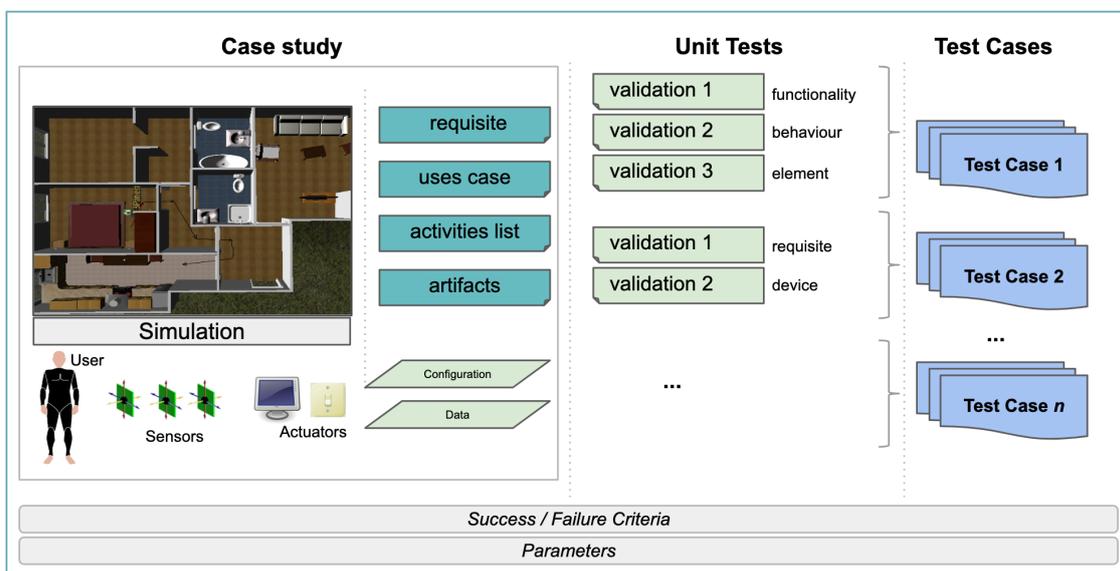


Figure 11.11: General scheme of the test environment.

Engineers use the case study and simulation described in Section 3.4 to build the test environment. The process takes into account the users directly involved in the assistive solution (see Section 3.4.1), the functional requirements describing the activities transferred to the simulation and the technical specifications of each simulated element (see Section 3.4.2). Also, the elements of the assistance domain described in Section 3.4.3 are identified, which frame the recurring situations that must be validated in the simulated scenarios. The unit tests, on the other hand, are directly associated with the use cases established in the requirements identification phase, following the tasks mentioned in Section 3.4.5. Each use case focuses on an aspect of the solution that is interested in being validated in the simulation. Within these aspects are the actions described in the list of activities, which has been obtained as a result of these tasks (see Section 3.4.6).

After defining the unit tests, the next step is to create the test cases that will group these tests, following specific criteria such as validating a specific requirement, validating a recurrent situation, the execution of user activity, a simulation element, among others. The set of validations give body to what is the test case and the results of all the cases created, according to the success/failure criteria established by the experts, is what finally determines if the case study is coherent with the simulation and if it meets the expectations of the user as a suitable scenario for the final solution.

11.4 Design requirements

Chapter 4 defined the test environment required to validate the simulation, as well as the different elements that are part of it. This Chapter describes the design requirements of the framework that supports the creation of such an environment, in which unit tests associated with the corresponding test cases will also be carried out.

11.4.1 Test environment

The main objective is to build a test environment that, using simulation as a specification, facilitates the validation of the elements that compose it. With the participation of experts and health professionals in this process, test cases are created that validates not only each element of the simulation but also the assistive solution it represents. Likewise, the test environment will provide the necessary tools for the simulation components to be configured so that they meet the specifications of the real case study. It proposes an event-based software architecture EDA to build test environments, used as an architectural paradigm to help in the definition of test cases.

The goal is to prioritize event processing and use it as the central element of the architecture and test case. In order to be able to manage the volume of data and work with it in real time during the simulation, it has been proposed to use the processing of complex events CEP as the base model for event-based decision making. The event flows generated by the simulation may contain a large volume of different events. All of them must be transformed, filtered, tagged, stored, aggregated and evaluated to trigger the actions that are deemed appropriate in the simulation. The proposed processing paradigm allows an analysis of the flows of compound events in real time, both individual and group. Due to this characteristic, this technique is considered an ideal solution to analyze the behaviour of the different elements that are part of the assistive solution, which is based on the sensors and actuators within the simulation.

11.4.2 Event model

The architecture proposes a model that processes event flows using the complex event processing paradigm CEP. The model is based on the fact that events, regardless of their source, are related to each other. Monitored environments apply this same principle through sensors. In these environments, the readings obtained in a given instant are useful to analyze the readings of the successive instants. In the simulation, it happens the same; this principle works in the same way that it happens in a real environment. The sensors do not operate individually, as a whole provide information that is relevant to respond to a service suitable for the user.

11.4.2.1 Rules

A rule or query is a specific pattern that identifies variations in sensor readings. In the framework, patterns are identified with labels just like animations, and some of them are predefined in the tool. The patterns used describe two types of variations in the reading: “change”, “gyro” and trend (“up”, “down”, “left”, “right”). Functions (rules) have also been written that calculate statistical values such as *sum*, *max*, *min*, *avg*, *var*, *mean*, applicable to inertial sensor readings. In the context of CEP, the rule definition is more formal and requires a generic syntax to standardize its structure within the framework. There are two types of rules: simple and complex. Simple rules identify patterns on events that have a unique value in their reading, without the need to use nested rules or

sub-consultations to find this pattern. Complex rules identify a pattern on events that have more than one value in their reading, as happens with inertial sensors.

11.4.2.2 Layer processing

The literature recommends event processing following a layered architecture such as that proposed by CEP. CEP uses the event hierarchy and abstracts it through the use of layers that process events immediately and without passing through intermediate states. As a design requirement, it has been proposed to follow this approach, where the responsibility of each layer is as follows:

- **First layer:** receives and processes raw or real events that come directly from the simulation.
- **The second layer:** takes output events from the first layer and generates new virtual events or domain events from a flow filtering process.
- **The third layer:** applies the rules to detect specific patterns, which allow identifying events of problems that interest in the flows. Then, from them, it generates intermediate or causes events that describe the problem to finally launch the action events that help to describe the behaviour of the simulation components.

11.4.3 Activity recognition models

In some instances, the motorization of the environment requires not only checking the state of the environment, but it is also important to verify or identify what activities the character performs within the simulation. Physical activity of the character can be the response, or part of it, that the environment expects from the user. This is why identifying whether or not the physical activity has been carried out is essential in test cases that verify the user's interaction with the system. The framework makes use of the character's *animations* as a representation of those physical activities. For monitoring to be possible, the character in the simulation must carry inertial sensors that allow capturing the acceleration of its movements. Classifying systems are components of the proposed model, responsible for processing the flow of events from inertial sensors, which are used in the simulation to monitor the physical activity of the character. These flows previously cross the entire layer structure proposed in Section 5.2.3.

11.5 Test Environment Architecture

Chapter 3 defined the proposed methodology for building simulations from case studies that represent assistive solutions. The purpose, in this case, is to use the simulation as a specification of the solution. In Chapter 4 it was proposed the way to define an environment where it is possible to validate that the simulation corresponds to this solution, using for it the test cases and the unit tests to validate specific aspects in the simulation. The framework that supports the creation of this environment is presented in Chapter 5, emphasizing the design requirements that have been taken into account for its construction. This Chapter shows the architecture developed to build the test cases that will address the simulation to be executed on it. It also shows the different functionalities that are applied both for the creation of unit tests and for the definition of more complex test cases.

11.6 Proposed architecture

The architecture shown in Figure 11.12 is proposed taking into account the design requirements described in Section 5.2. The architecture describes the diagram of components that make up the system. In particular, the two main elements: The *Simulation* and the *Tester*.

Simulation is the component that represents the modelling of the problem and describes the specifications of the case study that models the assistive solution. The *test cases* are applied to the simulation so that for its construction, we have the characteristics of the 3D format of it. The simulation in 3D is responsible for recreating, as accurately as possible, the scenario where the assistive solution is implemented, providing information about the environment through the network of sensors, actuators and the user.

Tester is the component that provides the necessary tools to build the test cases that will be applied to the information generated by the simulation. It allows the developer to create sets of *unit tests* that can be used on each component of the simulation. The developer establishes the characteristics that he wants to validate and, to do so, focuses each unit test on the simulation element responsible for those characteristics, as described in Section 4.2.

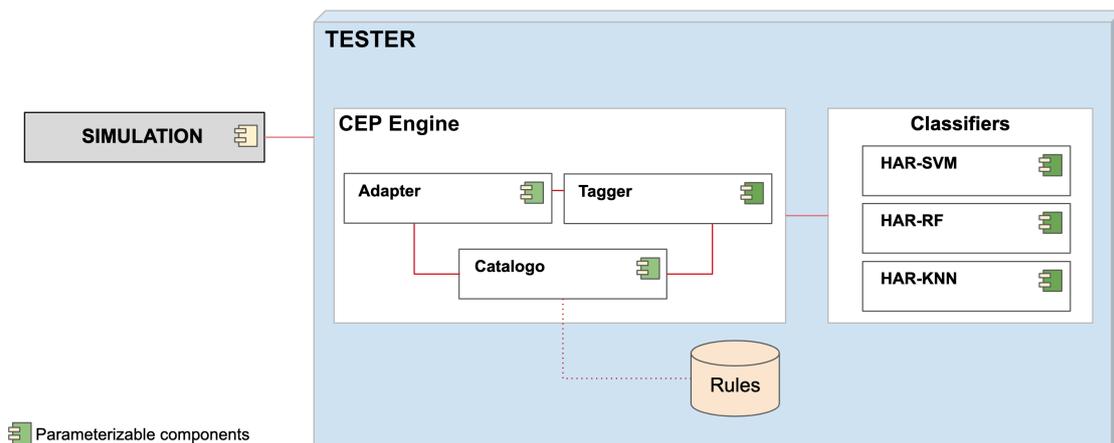


Figure 11.12: System architecture.

The Test Engine relies on two crucial elements of the system: the *CEP Engine* and the *Classifiers*.

- **CEP engine:** it is the node responsible for executing all the validations of the test case on the data generated by the simulation, taking into account the different configuration parameters established in the assistive solution and in the test case itself. This node initializes and launches the instances of processing of complex events that are behind each unit test. The instance connects to the corresponding event flow and executes the rules or queries that will determine, provided that the criteria for success or failure of the test are met within the established time window.
- **Classifiers:** this component groups specialized services in recognition of human activity HAR. These services are offered by classifier nodes, which are built using supervised automatic learning techniques. Recognition of the activity is necessary when it is necessary to identify specific movements or gestures of human characters, which are part of complex computer animations. Chapter 6 describes this component in more detail.

11.7 Implementation of the recognition system

This chapter describes the activity recognition system implemented as part of the proposed architecture to identify the actions of the character within the scenario.

11.7.1 Recognition systems

Recognition systems are components of the architecture in charge of recognizing the activities or movements carried out by the characters within the simulation. When sensors monitor physical activity in the simulation, for example, “*walk*”, “*run*”, “*drink*”, “*clap*” or “*open or close doors*”, it is not enough to use complex rules or event patterns to identify the activity the character performs. The reason for this is that people do not always perform physical activity in the same way, so it is not possible to identify constant patterns only by using the acceleration of the body. Systems HAR work better at identifying such actions. The proposed architecture uses specific classification models that recognize the activity executed by the user at a given moment. The primary use of the HAR system in test environments is that they allow the developer to define unit tests to check whether the user in the simulation performs normal activities and does so at a specific moment in the simulation. The actions can be part of the answer that the assistive solution expects to interact with the user and decide whether to execute other tasks.

11.7.2 Classifiers

There are three algorithms selected to generate the activity classification models: kNN, SVC and RF. Each model is trained to recognize the activities “*walking*”, “*running*”, “*drinking*”, “*get up*”, “*sit*” and “*clap*”. Recognition is carried out with one and two acceleration sensors, located on the user’s wrists. The reasons for choosing these three techniques are as follows:

- The *accuracy* and *precision* in recognizing user activity are high.
- The algorithms are *sensitive* when using the least number of sensors.
- The *response time* in the classification is low and
- The *latency* of recognition between transitions of activities is low.

The use of a classifier is subject to the conditions of the activities in character. When the engineers define the case study, they assess whether the specifications of the classifiers supported by the framework are sufficient to meet the needs of the test case or whether, on the contrary, they need to extend their functionality. It is also essential to determine how many sensors there are in the environment, the position on the character, the size of the time window and the sampling frequency in each device, described in more detail in the main chapter.

11.7.3 Modification of animations

The user’s activities are modelled in the simulation using computer animations. Each animation is a sequence of actions in character governed by some technique that models human movement. In a video game engine, one of the techniques used is skeleton animation, which consists of a hierarchy composed of a set of bones and joints. The more bones are part of the structure, the more realistic the movement. As can be seen in Figure 7.6, the proposed animations only use the primary joints that govern actions in a general way.

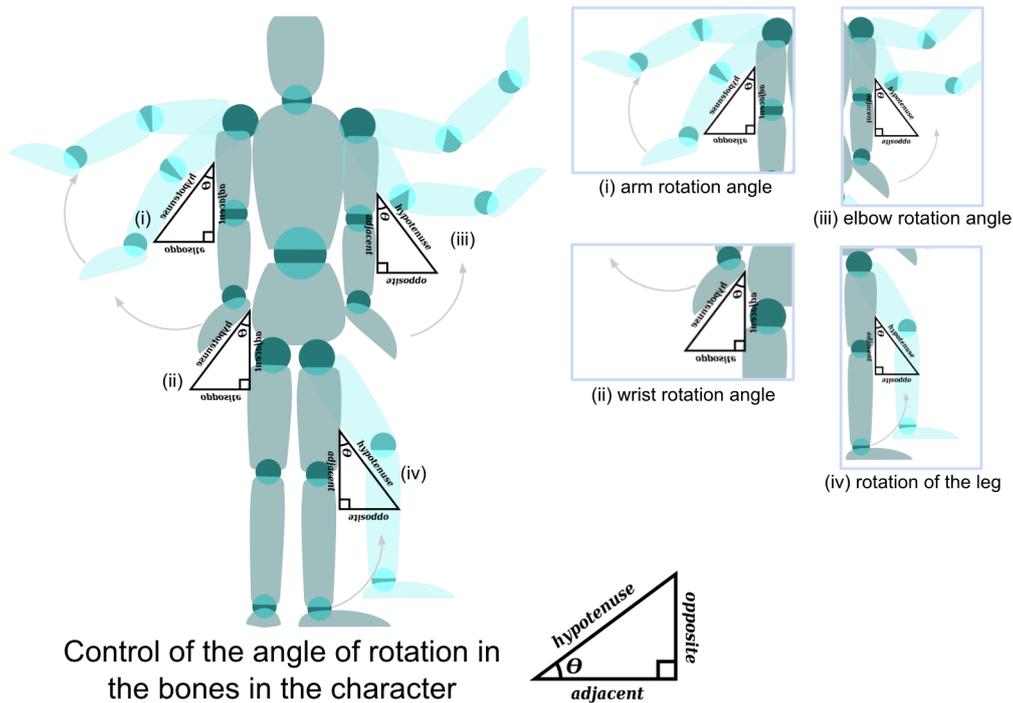


Figure 11.13: Rotation and translation angles in the main articulations of the character.

Actions are associated with the movement of a part of the body, and their main attributes are the changes of parameters used to describe the movement of that part of the body. The changes affect the rotation and translation of the body part in a question and vary according to the type of action. Actions are of two types: periodic and non-periodic. In a non-periodic action such as “raising the arm”, the parameters modify the duration of the action. In periodic actions, the parameters set the number of periods that define the action itself. The duration, in this case, is the result of the multiplication of the number of periods and the period of action.

11.8 Experimentation with test cases

The experimentation of this Thesis was carried out in two phases. The first phase consisted of using the methodology described in Chapter 3 to define the case studies and to apply three assistive solutions to them. Each case study used simulation as the primary specification element. The proposed assistive solutions are case studies whose objective is to observe different problems faced by older people in their daily life inside the home. The ultimate purpose of the solutions was to design and develop services that meet two needs in the user’s environment; these are:

1. To carry out long-term monitoring of the user while carrying out their daily activities inside the home.
2. To make the recognition of the physical activities that the user carries out while being monitored.

In the second phase of the experimentation, the work was to define the test cases necessary to validate the simulations proposed in the previous phase, using the framework

proposed in Chapter 5. The test cases contained the validations that showed how the simulation, in all scenarios, met the specifications described in the proposed case study.

11.8.1 Case study

The initial proposal of the case studies used in this experimentation arises from the interview catalogue of the project ColosAAL. This repository includes a total of 20 interviews correctly documented and published on the web¹. The catalogue was constructed with information obtained from previous interviews with older people. These users, at the time of the interview, suffered from common symptoms related to behavioural disorders and cognitive deterioration that affected their independence within the home. Family members and caregivers also participated in the interviews, indicating how these health problems affected the rest of the family. The interviews were designed by professionals in the areas of sociology, psychology and medicine, who also participated in the tabulation and analysis of the results. Also, computer science professionals have contributed tools and support platforms for group analysis of different cases.

Below are the different elements of the methodology used to define the case studies.

- **Users:** the first consideration was to define case studies in which users have participated with the roles described in Section 3.4.1. The purpose of this was to have profiles of users who have made some contribution when defining requirements associated with patient needs, environmental conditions and possible recommendations for improvement.
- **Requirements:** in each proposed case study, the objective was to use simulation as an artifact to represent its characteristics. The simulation is the specification of the requirements and also includes the necessary considerations to build the test cases. In order to capture the requirements, the engineers built tools that would facilitate the collection of information from all the users involved. One of these tools is **HackWithPeople**², which presents different case studies related to particular symptoms of diseases such as Alzheimer's and Parkinson's disease.
- **Case studies:** the case studies proposed for experimentation were as follows:
 - CS001ML: consists of simulating a common symptom in people suffering from cognitive impairment: memory loss. A set of activities carried out by the user were planned in the simulated scenario. Within these activities, there were concrete actions that the user executed when he suffered some episode of memory loss.
 - CS002DI: consists of simulating a common symptom in people suffering from behavioural disorders: disorientation. As in the first case, the scenario used some characteristics of this symptom and reproduced them in older people inside their homes, to show a series of episodes that would possibly indicate the existence of disorientation.
 - CS003AR: proposes to model scenarios in the simulation where the older adult carries out specific physical activities within the dwelling. The scenario allows modelling this type of activities in different areas of the house and with different positions of the body at the time of carrying them out.

¹Public access: <http://grasiagroup.fdi.ucm.es/aidendd/cases-list-1-10/>

²HackWithPeople, public access: <http://grasia.fdi.ucm.es/hack4people/index.php>

- **Assistive solutions:** the objective was to apply on the defined case studies an assistive solution with a specific service to older adults with the proposed ailments.
 - *Solution 1 and 2:* The objective was to use a long-term monitoring system that would recognize the user’s daily activities and identify typical anomalies in their daily routine. *Solution 1* used the information collected from the environment and the user to describe the daily activities, while *Solution 2* only took into account the information from the user and not from the environment.
 - *Solution 3:* corresponds to a solution for long-term monitoring of the user, describing the activities performed by the user and making recognition of specific physical activities as the simulation progresses.

Solutions 1 and 2 were represented with only two of the scenarios described in Section 3.3.1, a “typical scenario” and a “problem scenario”. In the first scenario, the user performed daily activities without any incident or behaviour different from the usual. The problem scenario showed the same context described in the previous scenario but added actions around the user to indicate the presence of anomalies in his day today. On the other hand, solution 3, besides using the two previous scenarios “typical” and “problem”, used a solution scenario where the long-term monitoring and activity recognition services were used as a proposal for the assistive solution to carry out the user follow-up. The domain elements described in Section 3.4.3 are present only in the CS003AR case study because it was the only scenario where it was necessary to use a real environment.

11.8.2 Evaluation of the system

To evaluate the system, the rules and the recognition system acted on the simulation to validate it. The proposal was to make modifications on the animations of each physical activity of the character to vary the way to execute individual animations in the simulation and to avoid them being repetitive. As indicated in Section 7.2.4.2, in the framework the physical activity or animation is programmatically altered to change both the angle of displacement in the bones of the character and the cadence with which it performs these actions. The modifications of the physical activities of the character varied the way of carrying out the actions in the animation. Thanks to this procedure, the framework generated a total of 109 actions, distributed among the different activities to be recognized. This task allowed the character to execute the same action differently as if they were different users. The modified animations were those included in the case study: “walk”, “drink”, “clap”, “wave-attention” or “stop” (*motionless*).

The proposed experimentation involved the construction of three test cases related to the corresponding case studies. Each test case validated the scenarios defined in each case study. Subsequently, the test cases included the unit tests necessary to validate the list of activities that made up the simulations. The results of these tests yielded reports that showed what happened with each unit test and what result is obtained compared to the expectations of the test developer. The result of the proposed unit tests showed how each simulation met the solution requirements specifications. That is, with the activities that were run in the simulation to describe the proposed scenario. The use of the recommendation system was useful in cases where there was a physical activity of the user, previously modelled from the behaviours of real users.

The results of the evaluation showed that the use of unit tests on the simulation, allow recognizing those characteristics that are identifiable and that when measured give a value as an indicator of the success of the simulation as an element of specification. Identifying

these characteristics requires the participation of experts who know about the assistance, especially when the aim is to use them to generate solutions in this field.

11.9 Testing in a real environment

In Chapter 8, the HAR system was built to recognize and validate the activities proposed in one of the case studies. This system was trained with the help of simulation, using the physical activities simulated with the modifications applied by the framework. According to the results obtained in the evaluation of the recognition system, described in Section 8.4.2, the system seems to behave well when it comes to identifying the activity, working of course, in a computer environment applied over the simulation. In this chapter, what is proposed is to validate this system using it directly in a real environment where it is possible to implement and deploy this solution and use it against real data to recognize the same activities proposed in the case study for which it was built.

The challenge is to implement the recognition system, trained and configured with the simulation, in a real environment that complies with all the specifications described in the corresponding case studies. The implementation required a device capable of fulfilling the same functions carried out with the simulation tests. That is to say, a device capable of housing and putting into operation the system that was used in the computer. The next step was to perform the same validations of the physical activities but using real user data. Complying with the deployment and implementation of the proposed monitoring system required defining two essential elements for the operation of the device: building a central processing unit were to run the system and choosing inertial sensors for monitoring the physical activity of the user.

The evaluation of the prototype included public datasets³ containing data on the acceleration of physical activities carried out by real users, which coincided with the simulated activities in the test case. A set of 323 files with data from 6 activities, performed by 15 different users, was used. The evaluation criterion used was to identify the accuracy of the recognition of the activity by the system. The indicators applied were *Precision*, *Recall* and *F1-Score* to observe the effectiveness of the system in this other environment.

In an assistive solution, the recognition of the activity plays an essential role because it allows recognizing actions of the user at the moment of interacting with him. Modelling this interaction is possible thanks to the simulation used in the observation of scenarios of this type of solutions. The main problem in assistive environments is precisely to identify those situations in which, in the long term, they produce interaction problems. Detecting these inconsistencies in time is hardly feasible using only assistive environments because it is not always possible to model all the user's behaviour. This inconsistencies also happens with simulation, but the panorama to add new models of behaviour over what is simulated is much more feasible with time than using real users for experimentation.

11.10 Conclusions and future work

The main contributions of this Thesis have been oriented to demonstrate that 3D simulation can be used as an element of initial validation, which allows deciding if the development of an assistive solution works as expected or not. To this end, it has been proposed that the effort applied in the process of defining and building the simulation,

³Dataset: <https://drive.google.com/drive/folders/1f1Golv6BImSH27M1j9f7BAyMgUvLp0a5?usp=sharing>

be used throughout the development cycle of the solution to achieve a higher quality of assistance at a lower cost. Following this approach, the main contributions have been to define a methodology that allows, on the one hand, to characterize the behaviour of the assistive solution in the simulation and, on the other hand, to build the test environments that facilitate the validation under the conceptual framework of the simulations.

An event-based architecture has been proposed for the test environment, the purpose of which is to facilitate the execution of test cases that are defined with the methodology. Finally, an experiment has been carried out showing how the methodology and the test environment are used to determine the value provided by simulations as specification elements in the assistive solutions.

In summary, the main contributions of this Thesis are the following:

- The methodology for characterizing the assistive solution in the simulation, described in Chapter 3, which is based on the definition of case studies that use the requirements of the solution to model them in the behavior of the simulated elements: “*user*”, “*environment*” and “*technology*”. To build the case studies, the different users of the solution participate in co-creative work sessions, where they identify the requirements and establish the validation criteria that they will apply for verification. Likewise, different simulated scenarios are set where these requirements are observed and compared while they are visualized in the execution of the simulation. As a result of applying the methodology, simulations are obtained that represent the case study and specify the criteria necessary to validate each simulation. The techniques used to develop the working sessions have been published in [BMCB18] (co-creation techniques) and in [CBCGSP18]. (participation techniques). The process of identifying requirements for activities in intelligent environments was published in citegomez2017requirement.
- The web platform *Hack4People* used to develop working sessions with users in the definition of the case study. Although this tool was not initially part of the proposed achievements in this research, it is an important contribution that has contributed to the process of identifying requirements.
- The test cases necessary to validate the simulations, which are described in Chapter 4 and are part of the proposed methodology. Validation depends on the unit tests that the test case collects and organizes. The unit tests are applied using pre-established criteria on the simulation, related to the data, user activities, the state of the environment and the involvement of the assistive technology. Elements of the methodology related to testing cases have been published in [CBSP18].
- The proposed framework provides the necessary infrastructure to create an environment where the simulation test cases are executed. To build these environments, a software architecture based on events EDA was proposed to help in the definition, creation and execution of test cases. To do this, event processing is prioritized and used as the central element of the architecture and test case. Data volume management and processing are carried out with a rule engine CEP used as the base model for event-based decision making. This architecture has been published in [CBSP18] and some of its components in [GMIJ19] (sensor testing mechanisms), [PCBGSP17] (sensor virtualization), [PGSOCB18] (simulation services) and [SOCBGSP18]. (configuration and parameterization of scenarios).
- As a complement to the test environment, a specialized system was built for the recognition of human activities that integrates with the simulation test environment,

described in Chapter 7. The system focuses on the recognition of the activities or “*animations*” performed by the “*user*” within the simulated “*environment*”. Its purpose was to include activity recognition as a technique to identify the user’s interaction with the environment through physical actions. In order to make the use of simulation possible, each animation is modified by the system before being used in the scenario, in order to avoid it being unrealistic and very repetitive. The objective of the modification is that the learning algorithm does not over-adjust when training with the animations. The prototype of the recognition system has been published in [GSCBPC16].

- The experimentation with test cases presented in Chapter 8, which has been carried out with the purpose of showing the usefulness of simulation in the validation of assistive solutions. An environment has been built to execute the test cases obtained from three specific case studies, where different user needs were observed that were covered totally or partially by an assistive solution. The result of the tests was used as a criterion to validate the simulation.
- The experimentation in a real environment described in Chapter 9, is this a challenge that has meant a greater effort due to the multiple tests that were made with different devices such as computers with reduced boards and sensors. These tests not only sought the compatibility of the framework with the hardware but also included features in the configuration to optimize the autonomy of the device, the portability of it and the performance of the solution once implemented.

11.10.1 Open lines of research

During the development of this Thesis, different aspects have been identified that have not been addressed due to their complexity, which, given their relevance, are proposed as possible open lines of research. Regarding the methodology to characterize the simulation, the aspects found are the following:

- It is costly to model user and environment characteristics that describe specific aspects of their behaviour, such as mobility problems that evolve or deteriorate over time, interaction with more functional objects in the environment, among others. Researching and applying mechanisms to include these models in the simulation could be useful to represent more realistic and complex scenarios of assistive solutions.
- In the area of assistance, despite the importance of situations such as technological aversion and problems of unwanted user responses, no specific studies are describing how to deal with this type of situation. It would be interesting to define a methodology to guide the different stakeholders in the construction of solutions taking into account this type of problem.

Concerning the construction of simulation scenarios, the creation of visual objects that, in addition to enriching the environment and the characters, allow the validation of characteristics of the assistive context, implies many costs that make it difficult to generate more realistic environments. In the present Thesis, new characters and scenarios related to the problems dealt with in the case studies were created. However, it would be convenient to develop tools around the systems AAL that allow using the simulation to observe particular characteristics that are validated from it. For the creation of these objects, the following needs have been identified:

- The case studies contemplate the use of elements in the simulation that must have an easily modelable representation from the assistive environment. Facilitating the generation of these models with the help of the corresponding professionals is a task that contributes to the staging of specific situations that validate both the scope of the final solution and its application in a real environment.
- The modelling processes of the physical activities of older users should have galleries of everyday activities carried out by users with different physical constraints. Including this type of user in the modelling process is not convenient, and in many cases, it is impossible. To this end, it would be appropriate to define techniques that allow these behaviours to be collected in the least invasive way possible and with a sufficient level of realism to allow subsequent experimentation.

Based on the lines of the research described in the previous section, this Thesis leaves open the lines of work whose general purpose be to improve the construction of more realistic simulation test environments. The works that have been identified for this purpose are the following:

- Add new features to user behaviour and the environment so that they are more realistically modelled in the case study.
- Include specifications describing how to model assistive technologies in simulated environments, to observe their interaction with other elements of the simulation.
- Add new sensors and actuators to expand the description of the environment. Concerning the user, new inertial sensors with different technical specifications are needed to experiment with the recognition of physical activity affected by the user's mobility problems.
- To extend the functionality of the system HAR so that it can be used in scenarios with very diverse activities.

Bibliography

- [AAW⁺07] Hamid Aghajan, Juan Carlos Augusto, Chen Wu, Paul McCullagh, and Julie-Ann Walkden. Distributed vision-based accident management for assisted living. In *International conference on Smart homes and health telematics*, pages 196–205. Springer, 2007.
- [ABB⁺16] Arvind Arasu, Brian Babcock, Shivnath Babu, John Cieslewicz, Mayur Datar, Keith Ito, Rajeev Motwani, Utkarsh Srivastava, and Jennifer Widom. Stream: The stanford data stream management system. In *Data Stream Management*, pages 317–336. Springer, 2016.
- [ABC⁺16] Giuseppe Amato, Davide Bacciu, Stefano Chessa, Mauro Dragone, Claudio Gallicchio, Claudio Gennaro, Hector Lozano, Alessio Micheli, Gregory MP O’Hare, Arantxa Renteria, et al. A benchmark dataset for human activity recognition and ambient assisted living. In *International Symposium on Ambient Intelligence*, pages 1–9. Springer, 2016.
- [ABW03] Arvind Arasu, Shivnath Babu, and Jennifer Widom. Cql: A language for continuous queries over streams and relations. In *International Workshop on Database Programming Languages*, pages 1–19. Springer, 2003.
- [ACRV13] Giovanni Acampora, Diane J Cook, Parisa Rashidi, and Athanasios V Vasilakos. A survey on ambient intelligence in healthcare. *Proceedings of the IEEE*, 101(12):2470–2494, 2013.
- [AID14] Grasia UCM AIDE. Aide framework. <http://grasia.fdi.ucm.es/aide/>, 01 2014. (Accessed on 12/02/2019).
- [AMHH18] Tomas Akenine-Moller, Eric Haines, and Naty Hoffman. *Real-time rendering*. AK Peters/CRC Press, 2018.
- [Ard12] Arduino. Arduino - home. <https://www.arduino.cc/>, 2012. (Accessed on 03/14/2019).
- [AS95] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *icde*, page 3. IEEE, 1995.
- [BBET97] Ronan Boulic, Pascal Bécheiraz, Luc Emering, and Daniel Thalmann. Integration of motion control techniques for virtual human and avatar real-time animation. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 111–118. ACM, 1997.
- [BCK⁺07] Peter J Bickel, Chao Chen, Jaimyoung Kwon, John Rice, Erik Van Zwet, and Pravin Varaiya. Measuring traffic. *Statistical Science*, pages 581–597, 2007.
- [BDB⁺14] Ralf Bruns, Jürgen Dunkel, Holger Billhardt, Marin Lujak, and Sascha Ossowski. Using complex event processing to support data fusion for ambulance coordination. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–7. IEEE, 2014.
- [BE02] Markus Bylund and Fredrik Espinoza. Testing and demonstrating context-aware services with quake iii arena. *Communications of the ACM*, 45(1):46–48, 2002.

- [Bea15] BeagleBoard.org. Beagleboard.org - black. <https://beagleboard.org/black>, 2015. (Accessed on 03/14/2019).
- [Bec18] Erich Gamma; Kent Beck. Testing framework for java and platform for the jvm. <https://junit.org/junit5/>, 2018.
- [BF⁺14] Pierre Bourque, Richard E Fairley, et al. *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*. IEEE Computer Society Press, 2014.
- [BFF⁺15] Flavia Benetazzo, Francesco Ferracuti, Alessandro Freddi, Andrea Giantomassi, Sabrina Iarlori, Sauro Longhi, Andrea Monteriù, and Davide Ortenzi. Aal technologies for independent life of elderly people. In *Ambient Assisted Living*, pages 329–343. Springer, 2015.
- [BI04] Ling Bao and Stephen S Intille. Activity recognition from user-annotated acceleration data. In *International Conference on Pervasive Computing*, pages 1–17. Springer, 2004.
- [BK09] Alejandro Buchmann and Boris Koldehofe. Complex event processing. *IT-Information Technology Methoden und innovative Anwendungen der Informatik und Informationstechnik*, 51(5):241–242, 2009.
- [BKHS09] BHMSA Bergvall-Kareborn, M Hoist, and Anna Stahlbrost. Concept design with a living lab approach. In *2009 42nd Hawaii International Conference on System Sciences*, pages 1–10. IEEE, 2009.
- [BKK⁺10] Shirley Beul, Lars Klack, Kai Kasugai, Christian Moellering, Carsten Roecker, Wiktorija Wilkowska, and Martina Ziefle. Between innovation and daily practice in the development of aal systems: Learning from the experience with today’s systems. In *International Conference on Electronic Healthcare*, pages 111–118. Springer, 2010.
- [BMCB18] Susana Bautista, Jelena Mazaj, and Marlon Cárdenas-Bonett. *Developing RRI Practices: The Role of the ICT in Stakeholders’ Knowledge Exchange and Co-creation Processes*, pages 21–26. Springer International Publishing, Cham, 2018.
- [BMvD⁺08] Sjoerd M. Bruijn, Onno G. Meijer, Jaap H. van Dieen, Idsart Kingma, and Claudine J.C. Lamoth. Coordination of leg swing, thorax rotations, and pelvis rotations during gait: The organisation of total body angular momentum. *Gait & Posture*, 27(3):455 – 462, 2008.
- [Bro84] Craig Brod. *Technostress: The human cost of the computer revolution*. Addison Wesley Publishing Company, 1984.
- [BSC⁺11] Robert Beringer, Andrew Sixsmith, Michael Campo, Julie Brown, and Rose McCloskey. The acceptance of ambient assisted living: Developing an alternate methodology to this limited research lens. In *International Conference on Smart Homes and Health Telematics*, pages 161–167. Springer, 2011.
- [BV03] John J Barton and Vikram Vijayaraghavan. Ubiwise, a simulator for ubiquitous computing systems design. *Hewlett-Packard Laboratories Palo Alto, à AI HPL-2003-93*, 2003.
- [BYH⁺18] Jennifer Boger, Victoria Young, Jesse Hoey, Tizneem Jiancaro, and Alex Mihailidis. Zero-effort technologies: Considerations, challenges, and use in health, wellness, and rehabilitation. *Synthesis Lectures on Assistive, Rehabilitative, and Health-Preserving Technologies*, 8(1):i–118, 2018.
- [CBB18] Marlon Cárdenas-Bonnet and Susana Bautista. Infostate: Interactive visualisation of the environment using ambient sensors. In *Proceedings of the XIX International Conference on Human Computer Interaction*, Interacción 2018, pages 23:1–23:2, New York, NY, USA, 2018. ACM.

- [CBCGSP18] Marlon Cárdenas-Bonett, Noelia García Castillo, Jorge Gómez-Sanz, and Juan Pavón. Participatory design with on-line focus groups and normative systems. In *Conference of the Spanish Association for Artificial Intelligence*, pages 66–75. Springer, 2018.
- [CBSP18] Marlon Cárdenas-Bonett, Jorge Gómez Sanz, and Juan Pavón. Testing ambient assisted living solutions with simulations. In *IFIP International Conference on Testing Software and Systems*, pages 56–61. Springer, 2018.
- [CCPFR12] Alexandros André Chaaaraoui, Pau Climent-Pérez, and Francisco Flórez-Revuelta. A review on vision techniques applied to human behaviour analysis for ambient-assisted living. *Expert Systems with Applications*, 39(12):10873–10888, 2012.
- [CCS+17] Davide Calvaresi, Daniel Cesarini, Paolo Sernani, Mauro Marinoni, Aldo Franco Dragoni, and Arnon Sturm. Exploring the ambient assisted living domain: a systematic review. *Journal of Ambient Intelligence and Humanized Computing*, 8(2):239–257, 2017.
- [CFR14] Alexandros Andre Chaaaraoui and Francisco Flórez-Revuelta. Continuous human action recognition in ambient assisted living scenarios. In *International Conference on Mobile Networks and Management*, pages 344–357. Springer, 2014.
- [CFS+14] Ching Yu Chen, Jui Hsi Fu, Today Sung, Ping-Feng Wang, Emery Jou, and Ming-Whei Feng. Complex event processing for the internet of things and its applications. In *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, pages 1144–1149. IEEE, 2014.
- [CHKR14] Michael Cammert, Christoph Heinz, Jürgen Krämer, and Tobias Riemenschneider. Systems and/or methods for user feedback driven dynamic query rewriting in complex event processing environments, July 22 2014. US Patent 8,788,484.
- [CJN17] Angelo Costa, Vicente Juliá'n, and Paulo Novais. Advances and trends for the development of ambient-assisted living platforms. *Expert Systems*, 34(2):e12163, 2017. e12163 EXSY-Mar-15-075.R1.
- [CM12] Gianpaolo Cugola and Alessandro Margara. Processing flows of information: From data stream to complex event processing. *ACM Comput. Surv.*, 44(3):15:1–15:62, June 2012.
- [CP03] Haowen Chan and Adrian Perrig. Security and privacy in sensor networks. *computer*, 36(10):103–105, 2003.
- [CS+17] Pablo Campillo Sánchez et al. *Desarrollo dirigido por modelos en simulaciones de inteligencia ambiental*. PhD thesis, Universidad de Murcia, 2017.
- [CSGS15] Pablo Campillo-Sanchez and Jorge J Gomez-Sanz. A framework for developing multi-agent systems in ambient intelligence scenarios. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1949–1950. International Foundation for Autonomous Agents and Multiagent Systems, 2015.
- [CYL+08] Yen-Ping Chen, Jhun-Ying Yang, Shun-Nan Liou, Gwo-Yun Lee, and Jeen-Shing Wang. Online classifier construction algorithm for human activity detection using a tri-axial accelerometer. *Applied Mathematics and Computation*, 205(2):849–860, 2008.
- [DA+00] Anind K Dey, Gregory D Abowd, et al. The context toolkit: Aiding the development of context-aware applications. In *Workshop on Software Engineering for wearable and pervasive computing*, pages 431–441, 2000.
- [dCAT08] Alexandre de Castro Alves and James Taylor. Event processing query language using pattern matching, December 4 2008. US Patent App. 12/043,552.

- [Dey01] Anind K Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.
- [DFOO11] Jürgen Dunkel, Alberto Fernández, Rubén Ortiz, and Sascha Ossowski. Event-driven architecture for decision support in traffic management systems. *Expert Systems with Applications*, 38(6):6530–6539, 2011.
- [DMKJ⁺10] Katrien De Moor, Istvan Ketyko, Wout Joseph, Tom Deryckere, Lieven De Marez, Luc Martens, and Gino Verleye. Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting. *Mobile Networks and Applications*, 15(3):378–391, 2010.
- [DOB⁺16] Kadian Davis, Evans Owusu, Vahid Bastani, Lucio Marcenaro, Jun Hu, Carlo Regazzoni, and Loe Feijs. Activity recognition based on inertial sensors for ambient assisted living. In *2016 19th international conference on information fusion (fusion)*, pages 371–378. IEEE, 2016.
- [Dun09] Jürgen Dunkel. On complex event processing for sensor networks. In *2009 International Symposium on Autonomous Decentralized Systems*, pages 1–6. IEEE, 2009.
- [DxMG⁺16] Ciprian Dobre, Constandinos x Mavromoustakis, Nuno Garcia, Rossitza Ivanova Goleva, and George Mastorakis. *Ambient Assisted Living and Enhanced Living Environments: Principles, Technologies and Control*. Butterworth-Heinemann, 2016.
- [EI04] Jacqueline K Eastman and Rajesh Iyer. The elderly’s uses and attitudes towards the internet. *Journal of Consumer Marketing*, 21(3):208–220, 2004.
- [ENK⁺05] Mats Eriksson, Veli-Pekka Niitamo, Seija Kulkki, et al. State-of-the-art in utilizing living labs approach to user-centric ict innovation-a european approach. *Lulea: Center for Distance-spanning Technology. Lulea University of Technology Sweden: Lulea*, 2005.
- [ESSH06] J Mikael Eklund, J Sprinkle, Shankar Sastry, and TR Hansen. Information technology for assisted living at home: building a wireless infrastructure for assisted living. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 3931–3934. IEEE, 2006.
- [Eur18] Eurostat. People in the eu - population projections - statistics explained. https://ec.europa.eu/eurostat/statistics-explained/index.php/People_in_the_EU_-_population_projections#An_ageing_society, 11 2018. (Accessed on 12/03/2018).
- [FCR⁺18] Arthur D Fisk, Sara J Czaja, Wendy A Rogers, Neil Charness, and Joseph Sharit. *Designing for older adults: Principles and creative human factors approaches*. CRC press, 2018.
- [FRN⁺10] K. Frank, M. Röckl, M. J. Vera Nadales, P. Robertson, and T. Pfeifer. Comparison of exact static and dynamic bayesian context inference methods for activity recognition. In *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 189–195, March 2010.
- [GGR16] Minos Garofalakis, Johannes Gehrke, and Rajeev Rastogi. *Data Stream Management: Processing High-Speed Data Streams*. Springer, 2016.
- [GMIJ19] Gómez-Sanz Jorge, García-Magariño Iván., Cárdenas-Bonett Marlon and Pérez Díez Juan. Framework-supported mechanism of testing algorithms for assessing memory and detecting disorientation from iot sensors. In *5th IEEE World Forum on Internet of Things*, pages 1–6. IEEE, 2019.
- [Gra15] Grasia UCM. Ambient Intelligence Development Environment (AIDE), 2015.

- [Gre98] Thomas L Greenbaum. *The handbook for focus group research*. Sage, 1998.
- [Gre01] Saul Greenberg. Context as a dynamic construct. *Human-Computer Interaction*, 16(2):257–268, 2001.
- [GSCBPC16] Jorge J Gómez-Sanz, Marlon Cárdenas-Bonett, Rafael Pax, and Pablo Campillo. Building prototypes through 3d simulations. *PAAMS*, pages 299–301, 2016.
- [GSCS17] Jorge J Gomez-Sanz and Pablo Campillo-Sánchez. Domain independent regulative norms for evaluating performance of assistive solutions. *Pervasive and Mobile Computing*, 34:79–90, 2017.
- [GSPACB17] Jorge J Gomez-Sanz, Rafael Pax, Millán Arroyo, and Marlon Cárdenas-Bonett. Requirement engineering activities in smart environments for large facilities. *Comput. Sci. Inf. Syst.*, 14(1):239–255, 2017.
- [GVCS⁺12] Teresa Garcia-Valverde, Francisco Campuzano, Emilio Serrano, Ana Villa, and Juan A Botia. Simulation of human behaviours for the validation of ambient intelligence services: A methodological approach. *Journal of Ambient Intelligence and Smart Environments*, 4(3):163–181, 2012.
- [Har09] Mossaab Hariz. *Mechanism for handling user interface plasticity in ambient assisted living*. PhD thesis, Evry, Institut national des télécommunications, 2009.
- [HCL⁺12] Abdelsalam Helal, Kyungeun Cho, Wonsik Lee, Yunsick Sung, JW Lee, and Eunju Kim. 3d modeling and simulation of human activities in smart spaces. In *Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on*, pages 112–119. IEEE, 2012.
- [HEN⁺09] Andreas Hein, Marco Eichelberg, Oliver Nee, Arne Schulz, Axel Helmer, and Myriam Lipprandt. A service oriented platform for health services and ambient assisted living. In *2009 International Conference on Advanced Information Networking and Applications Workshops*, pages 531–537. IEEE, 2009.
- [Hen14] Monique M Hennink. *Focus group discussions*. Series in understanding qualitative research. Oxford University Press, 1 edition, 2014.
- [HJ09] Z. He and L. Jin. Activity recognition from acceleration data based on discrete cosine transform and svm. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 5041–5044, Oct 2009.
- [HK11] Allen Higgins and Stefan Klein. Introduction to the living lab approach. In *Accelerating global supply chains with IT-innovation*, pages 31–36. Springer, 2011.
- [HNK09] Y. Hanai, J. Nishimura, and T. Kuroda. Haar-like filtering for human activity recognition using 3d accelerometer. In *2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, pages 675–678, Jan 2009.
- [HP97] Jessica K Hodgins and Nancy S Pollard. Adapting simulated behaviors for new characters. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 153–162. ACM Press/Addison-Wesley Publishing Co., 1997.
- [HP08] Hugh Herr and Marko Popovic. Angular momentum in human walking. *Journal of experimental biology*, 211(4):467–481, 2008.
- [HRH⁺09] Manfred Huber, Ricardo Rodrigues, Frédérique Hoffmann, Katrin Gasior, and Bernd Marin. Facts and figures on long-term care. *Europe and North America. Vienna: European Centre for Social Welfare Policy and Research*, 2009.

- [HS05] Tãm Huynh and Bernt Schiele. Analyzing features for activity recognition. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, pages 159–163. ACM, 2005.
- [HTC⁺15] Amy S Hwang, Khai N Truong, Jill I Cameron, Eva Lindqvist, Louise Nygård, and Alex Mihailidis. Co-designing ambient assisted living (aal) environments: unravelling the situated context of informal dementia care. *BioMed research international*, 2015, 2015.
- [i2c14] i2c.org. I2c interface - i2c bus. <https://www.i2c-bus.org/i2c-interface/>, 2014. (Accessed on 03/14/2019).
- [JAF⁺06] Shawn R Jeffery, Gustavo Alonso, Michael J Franklin, Wei Hong, and Jennifer Widom. A pipelined framework for online cleaning of sensor data streams. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 140–140. IEEE, 2006.
- [JALT08] Holger Junker, Oliver Amft, Paul Lukowicz, and Gerhard Tröster. Gesture spotting with body-worn inertial sensors to detect user activities. *Pattern Recognition*, 41(6):2010–2024, 2008.
- [JKC07] Do-Un Jeong, Se-Jin Kim, and Wan-Young Chung. Classification of posture and movement using a 3-axis accelerometer. In *Convergence Information Technology, 2007. International Conference on*, pages 837–844. IEEE, 2007.
- [JO17] Judge James O. Gait disorders in the elderly - geriatrics - msd manual professional edition. <https://www.msmanuals.com/professional/geriatrics/gait-disorders-in-the-elderly/gait-disorders-in-the-elderly>, 6 2017. (Accessed on 02/07/2019).
- [KBR⁺07] Thomas Kleinberger, Martin Becker, Eric Ras, Andreas Holzinger, and Paul Müller. Ambient intelligence in assisted living: enable elderly people to handle future interfaces. In *International conference on universal access in human-computer interaction*, pages 103–112. Springer, 2007.
- [KBRSG10] Bernd Krieg-Brückner, Thomas Röfer, Hui Shi, and Bernd Gersdorf. Mobility assistance in the bremen ambient assisted living lab. *GeroPsych*, 2010.
- [KC14] Narayanan C Krishnan and Diane J Cook. Activity recognition on streaming sensor data. *Pervasive and mobile computing*, 10:138–154, 2014.
- [KJCP09] Narayanan C Krishnan, Colin Juillard, Dirk Colbry, and Sethuraman Panchanathan. Recognition of hand movements using wearable accelerometers. *Journal of Ambient Intelligence and Smart Environments*, 1(2):143–155, 2009.
- [KMTM⁺98] Prem Kalra, Nadia Magnenat-Thalmann, Laurent Moccozet, Gael Sannier, Amaury Aubel, and Daniel Thalmann. Real-time animation of realistic virtual humans. *IEEE Computer Graphics and Applications*, 18(5):42–56, 1998.
- [KOA⁺99] Cory D Kidd, Robert Orr, Gregory D Abowd, Christopher G Atkeson, Irfan A Essa, Blair MacIntyre, Elizabeth Mynatt, Thad E Starner, and Wendy Newstetter. The aware home: A living laboratory for ubiquitous computing research. In *International Workshop on Cooperative Buildings*, pages 191–198. Springer, 1999.
- [KPP⁺12] Catriona M Kennedy, John Powell, Thomas H Payne, John Ainsworth, Alan Boyd, and Iain Buchan. Active assistance technology for health-related behavior change: an interdisciplinary review. *Journal of medical Internet research*, 14(3):e80, 2012.
- [KRA94] A Krueger Richard and Casey Mary Anne. Focus groups. a practical guide for applied research, 1994.
- [Lam78] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.

- [LB97] Kerrie Laguna and Renée L Babcock. Computer anxiety in young and older adults: Implications for human-computer interactions in older populations. *Computers in human behavior*, 13(3):317–326, 1997.
- [LB09] Linda Little and Pam Briggs. Privacy factors for successful ubiquitous computing. *International Journal of E-Business Research (IJEER)*, 5(2):1–20, 2009.
- [LCC⁺18] Giuseppe Liotta, Helena Canhao, Fabian Cenko, Rita Cutini, Ercole Vellone, Maddalena Illario, Przemyslaw Kardas, Andrea Poscia, Rute Dinis Sousa, Leonardo Palombi, and Maria Cristina Marazzi. Active ageing in europe: Adding healthy life to years. *Frontiers in Medicine*, 5:123, 2018.
- [LF98] David C Luckham and Brian Frasca. Complex event processing in distributed systems. *Computer Systems Laboratory Technical Report CSL-TR-98-754*. Stanford University, Stanford, 28, 1998.
- [LGVBP06] Leticia Elizabeth Luque, María Cristina González Verheust, and María Cristina Burba Pons. Estudio sobre el miedo a la tecnología en adultos mayores. In *XIII Jornadas de Investigación y Segundo Encuentro de Investigadores en Psicología del Mercosur*. Facultad de Psicología-Universidad de Buenos Aires, 2006.
- [LHBC⁺11] Lesa Lorenzen-Huber, Mary Boutain, L Jean Camp, Kalpana Shankar, and Kay H Connelly. Privacy, technology, and aging: a proposed framework. *Ageing International*, 36(2):232–252, 2011.
- [LKLC03] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11. ACM, 2003.
- [LM02] Seon-Woo Lee and Kenji Mase. Activity and location recognition using wearable sensors. *IEEE pervasive computing*, 1(3):24–32, 2002.
- [LNC⁺11] Luís Lima, Paulo Novais, Ricardo Costa, José Bulas Cruz, and José Neves. Group decision making and quality-of-information in e-health systems. *Logic Journal of the IGPL*, 19(2):315–332, 2011.
- [LRE10] B. Longstaff, S. Reddy, and D. Estrin. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *2010 4th International Conference on Pervasive Computing Technologies for Healthcare*, pages 1–7, March 2010.
- [Lu12] Younan Lu. Systems and methods of performing speech recognition with barge-in for use in a bluetooth system, February 7 2012. US Patent 8,112,280.
- [Luc02] David Luckham. *The power of events*, volume 204. Addison-Wesley Reading, 2002.
- [MHS01] Jani Mantyjarvi, Johan Himberg, and Tapio Seppanen. Recognizing human motion with multiple acceleration sensors. In *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236)*, volume 2, pages 747–752. IEEE, 2001.
- [Mic06] Brenda M Michelson. Event-driven architecture overview. *Patricia Seybold Group*, 2, 2006.
- [Mor97] David L Morgan. *The focus group guidebook*, volume 1. Sage publications, 1997.
- [MRB06] Anne-Sophie Melenhorst, Wendy A Rogers, and Don G Bouwhuis. Older adults’ motivated choice for technological innovation: Evidence for benefit-driven selectivity. *Psychology and aging*, 21(1):190, 2006.
- [MRCJ01] Elizabeth D Mynatt, Jim Rowan, Sarah Craighill, and Annie Jacobs. Digital family portraits: supporting peace of mind for extended family members. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 333–340. ACM, 2001.

- [MS09] Ingrid Mulder and Pieter Jan Stappers. Co-creating in practice: results and challenges. In *2009 IEEE International Technology Management Conference (ICE)*, pages 1–8. IEEE, 2009.
- [MSHR02] Samuel Madden, Mehul Shah, Joseph M Hellerstein, and Vijayshankar Raman. Continuously adaptive continuous queries over streams. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 49–60. ACM, 2002.
- [MSSD06] Uwe Maurer, Asim Smailagic, Daniel P Siewiorek, and Michael Deisher. Activity recognition and monitoring using multiple sensors on different body positions. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2006.
- [Mul12] Ingrid Mulder. Living labbing the rotterdam way: Co-creation as an enabler for urban innovation. *Technology Innovation Management Review*, 2(9), 2012.
- [MVK08] Ingrid Mulder, Daan Velthausz, and Martijn Kriens. The living labs harmonization cube: Communicating living lab’s essentials. *The Electronic Journal for Virtual Organizations and Networks*, 10:1–14, 2008.
- [MW09] Vasiliki Mouttzi and Christopher Wills. Utilizing living labs approach for the validation of services for the assisting living of elderly people. In *2009 3rd IEEE International Conference on Digital Ecosystems and Technologies*, pages 552–557. IEEE, 2009.
- [MWP⁺14] Mukhtiar Memon, Stefan Wagner, Christian Pedersen, Femina Beevi, and Finn Hansen. Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes. *Sensors*, 14(3):4312–4341, 2014.
- [NAA09] Hideyuki Nakashima, Hamid Aghajan, and Juan Carlos Augusto. *Handbook of ambient intelligence and smart environments*. Springer Science & Business Media, 2009.
- [Nar96] Bonnie A Nardi. Studying context: A comparison of activity theory, situated action models, and distributed cognition. *Context and consciousness: Activity theory and human-computer interaction*, 69102, 1996.
- [NCC⁺08] Paulo Novais, Ricardo Costa, Davide Carneiro, José Machado, Luís Lima, and José Neves. Group support in collaborative networks organizations for ambient assisted living. In *Towards Sustainable Society on Ubiquitous Networks*, pages 353–362. Springer, 2008.
- [Neu17] Neuronmocap. Perception neuron by noitom — perception neuron motion capture for virtual reality, animation, sports, gaming and film. <https://neuronmocap.com/>, 01 2017. (Accessed on 03/19/2019).
- [NFS⁺09] Juan-Carlos Naranjo, Carlos Fernandez, Pilar Sala, Michael Hellenschmidt, and Franco Mercalli. A modelling framework for ambient assisted living validation. In *International Conference on Universal Access in Human-Computer Interaction*, pages 228–237. Springer, 2009.
- [NSK05] Ali A Nazari Shirehjini and Felix Klar. 3dsim: rapid prototyping ambient intelligence. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, pages 303–307. ACM, 2005.
- [O⁺11] World Health Organization et al. Palliative care for older people: better practices. *World Health Organization. Regional Office for Europe*, 2011.
- [OB09] Tim O’Reilly and John Battelle. *Web squared: Web 2.0 five years on.* ” O’Reilly Media, Inc.”, 2009.

- [OPS13] Andrea Orlandinia, Filippo Palumbob, and Ales Stimecc. Steps toward end-to-end personalized aal services. In *Workshop Proceedings of the 9th International Conference on Intelligent Environments*, volume 17, page 78. IOS Press, 2013.
- [PBC⁺03] Martha E Pollack, Laura Brown, Dirk Colbry, Colleen E McCarthy, Cheryl Orosz, Bart Peintner, Sailesh Ramakrishnan, and Ioannis Tsamardinis. Autominder: An intelligent cognitive orthotic system for people with memory impairment. *Robotics and Autonomous Systems*, 44(3-4):273–282, 2003.
- [PBGSP17] Rafael Pax, Marlon Cárdenas-Bonett Bonett, Jorge J Gómez-Sanz, and Juan Pavón. Virtual development of a presence sensor network using 3d simulations. In *International Conference on Smart Cities*, pages 154–163. Springer, 2017.
- [PCBGSP17] Rafael Pax, Marlon Cárdenas-Bonett, Jorge J. Gómez-Sanz, and Juan Pavón. Virtual development of a presence sensor network using 3d simulations. In Enrique Alba, Francisco Chicano, and Gabriel Luque, editors, *Smart Cities*, pages 154–163, Cham, 2017. Springer International Publishing.
- [PEK⁺06] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen. Activity classification using realistic data from wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):119–128, Jan 2006.
- [PG16] A. B. Pawar and S. Ghumbre. A survey on iot applications, security challenges and counter measures. In *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, pages 294–299, Dec 2016.
- [PGK⁺09] Stephen J Preece, John Y Goulermas, Laurence PJ Kenney, Dave Howard, Kenneth Meijer, and Robin Crompton. Activity identification using body-mounted sensors a review of classification techniques. *Physiological measurement*, 30(4):R1, 2009.
- [PGKH09] Stephen J Preece, John Yannis Goulermas, Laurence PJ Kenney, and David Howard. A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Transactions on Biomedical Engineering*, 56(3):871–879, 2009.
- [PGSOCB18] Rafael Pax, Jorge Gomez-Sanz, Ismael Sagredo Olivenza, and Marlon Cárdenas-Bonett. A cloud based simulation service for 3d crowd simulations. In *2018 IEEE/ACM 22nd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, pages 1–5, Oct 2018.
- [PK09] Adrian Paschke and Alexander Kozlenkov. Rule-based event processing and reaction rules. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, pages 53–66. Springer, 2009.
- [PN12] Victoria Pimentel and Bradford G Nickerson. Communicating and displaying real-time data with websocket. *IEEE Internet Computing*, 16(4):45–53, 2012.
- [Pre08] Roger S Pressman. Ingeniería del software un enfoque práctico. 6^a. Edición México: MC GRAW HILL, 2008.
- [PTSS10] Marc Pallot, Brigitte Trousse, Bernard Senach, and Dominique Scapin. Living lab research landscape: From user centred design and user experience towards user cocreation. In *First European Summer School” Living Labs”*, 2010.
- [Puy05] Emmanuel Puyubaret. Sweet home 3d. <http://www.sweethome3d.com/es/>, 5 2005. (Accessed on 02/23/2019).
- [PyP14] PyPI. pyarduino · pypi. <https://pypi.org/project/pyarduino/>, 2014. (Accessed on 03/14/2019).
- [RAS08] C. Ramos, J. C. Augusto, and D. Shapiro. Ambient intelligence the next step for artificial intelligence. *IEEE Intelligent Systems*, 23(2):15–18, March 2008.
- [Ras16a] Raspberry. Raspberry pi 3 model b - raspberry pi. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>, 01 2016. (Accessed on 03/14/2019).

- [Ras16b] Raspberry. Raspberry pi zero - raspberry pi. <https://www.raspberrypi.org/products/raspberry-pi-zero/>, 2016. (Accessed on 03/14/2019).
- [RF05] P. Remagnino and G. L. Foresti. Ambient intelligence: A new multidisciplinary paradigm. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 35(1):1–6, Jan 2005.
- [RM13] Parisa Rashidi and Alex Mihailidis. A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, 17(3):579–590, 2013.
- [RMB⁺10] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. Using mobile phones to determine transportation modes. *ACM Trans. Sen. Netw.*, 6(2):13:1–13:27, March 2010.
- [Röc13] Carsten Röcker. User-centered design of intelligent environments: requirements for designing successful ambient assisted living systems. In *Central European Conference on Information and Intelligent Systems*, page 4. Faculty of Organization and Informatics Varazdin, 2013.
- [ROGP⁺13] Jorge Luis Reyes-Ortiz, Alessandro Ghio, Xavier Parra, Davide Anguita, Joan Cabestany, and Andreu Catala. Human activity and motion disorder recognition: towards smarter interactive cognitive environments. In *ESANN*. Citeseer, 2013.
- [Ros04] Mark Ross. Hearing assistance technology: Making a world of difference. *The Hearing Journal*, 57(11):12–17, 2004.
- [RSL⁺18] Michele Ruta, Floriano Scioscia, Giuseppe Loseto, Filippo Gramegna, Saverio Ieva, Agnese Pinto, and Eugenio Di Sciascio. Social internet of things for domotics: a knowledge-based approach over ldp-coap. *Semantic Web*, 2018.
- [RZ11] Arie Reichman and Moti Zwiling. The architecture of ambient assisted living system. In *2011 IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS 2011)*, pages 1–4. IEEE, 2011.
- [SC02] Kumaresan Sanmugalingam and George Coulouris. A generic location event simulator. In *International Conference on Ubiquitous Computing*, pages 308–315. Springer, 2002.
- [SCSE10] Geetika Singla, Diane J Cook, and Maureen Schmitter-Edgecombe. Recognizing independent and joint activities among multiple residents in smart environments. *Journal of ambient intelligence and humanized computing*, 1(1):57–63, 2010.
- [Sek10] Ramesh Sekhar. Wireless network system with wireless access ports, June 1 2010. US Patent 7,729,326.
- [SF07] Jens Schumacher and Karin Feurstein. Living labs-the user as co-creator. In *2007 IEEE International Technology Management Conference (ICE)*, pages 1–6. IEEE, 2007.
- [SHP04] Alla Safonova, Jessica K Hodgins, and Nancy S Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 514–521. ACM, 2004.
- [SKK⁺05] Hua Si, Yoshihiro Kawahara, Hisashi Kurasawa, Hiroyuki Morikawa, and Tomonory Aoyama. A context-aware collaborative filtering algorithm for real world oriented content delivery service. *Proc. of ubiPCMM*, 2005.
- [SKR⁺11] Katie A Siek, Danish U Khan, Stephen E Ross, Leah M Haverhals, Jane Meyers, and Steven R Cali. Designing a personal health application for older adults to manage medications: a comprehensive case study. *Journal of medical systems*, 35(5):1099–1121, 2011.

- [SLKC01] Roy Schmidt, Kalle Lyytinen, Mark Keil, and Paul Cule. Identifying software project risks: An international delphi study. *Journal of management information systems*, 17(4):5–36, 2001.
- [SMDM11] Dimitri Schuurman, Dominik Mahr, and Lieven De Marez. User characteristics for customer involvement in innovation processes: Deconstructing the lead user-concept. In *ISPIM 22nd conference: Sustainability in innovation: innovation management challenges*. International Society for Professional Innovation Management (ISPIM), 2011.
- [SOCBGSP18] Ismael Sagredo-Olivenza, Marlon Cárdenas-Bonett, Jorge Gómez-Sanz, and Juan Pavón. Using graphs of queues and genetic algorithms to fast approximate crowd simulations. In *Multidisciplinary Digital Publishing Institute Proceedings*, volume 2, page 1216, 2018.
- [SP19] UCM SociAAL Project. Aide web - framework. <http://grasia.fdi.ucm.es/aide/>, 01 2019. (Accessed on 01/12/2019).
- [SSF⁺17] Susanna Spinsante, Vera Stara, Elisa Felici, Laura Montanini, Laura Raffaelli, Lorena Rossi, and Ennio Gambi. The human factor in the design of successful ambient assisted living technologies. In *Ambient Assisted Living and Enhanced Living Environments*, pages 61–89. Elsevier, 2017.
- [SSP15] Jorge Gómez Sanz, Pablo Campillo Sánchez, and Juan Pavón. Social: Un entorno 3d para la simulación de soluciones aal. In *ACTAS V Congreso Internacional de Turismo para Todos: VI Congreso Internacional de Diseño, Redes de Investigación y Tecnología para todos DRT4ALL, 2015*, pages 245–270. Universidad Internacional de Andalucía, 2015.
- [Stå08] A Ståhlbröst. *The living lab way of user involvement*. PhD thesis, Ph. D thesis, Lulea University of Technology, Lulea, 2008.
- [Ste12] Constantine Stephanidis. The encyclopedia of human-computer interaction. *The encyclopedia of human-computer interaction*, 2012.
- [TdRD⁺11] António JS Teixeira, Nelson Pacheco da Rocha, Miguel Sales Dias, Daniela Braga, Alexandra Queirós, Osvaldo Pacheco, José Alberto Fonseca, Joaquim Sousa Pinto, Hugo Gamboa, Luís Corte-Real, et al. A new living lab for usability evaluation of ict and next generation networks for elderly@ home. In *AAL*, pages 85–97, 2011.
- [TIH⁺07] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *2007 11th IEEE International Symposium on Wearable Computers*, pages 37–40, Oct 2007.
- [Tin86] Mary E Tinetti. Performance-oriented assessment of mobility problems in elderly patients. *Journal of the American Geriatrics Society*, 34(2):119–126, 1986.
- [TJ12] Adrien Mouaffo Tiadjio and Kavyashree Jamboti. Requirements and evaluation of safety analysis techniques for ambient assisted living systems. In *2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops*, pages 319–324. IEEE, 2012.
- [TMO09] Rose Tinabo, Fred Mtenzi, and Brendan O’Shea. Anonymisation vs. pseudonymisation: Which one is most useful for both privacy protection and usefulness of e-healthcare data. In *2009 International Conference for Internet Technology and Secured Transactions, (ICITST)*, pages 1–6. IEEE, 2009.
- [TMS⁺07] Machiko R Tomita, William C Mann, Kathleen Stanton, Akihiko D Tomita, and Vidyalakshmi Sundar. Use of currently available smart home technology by frail elders: process and outcomes. *Topics in geriatric rehabilitation*, 23(1):24–34, 2007.
- [TP05] Lyn Turney and Catherine Pocknee. Virtual focus groups: New frontiers in research. *International Journal of Qualitative Methods*, 4(2):32–43, 2005.

- [UC317] UC3M. Comsotec. <http://www.comsotec.org/index.html>, 05 2017. (Accessed on 24/01/2019).
- [VBDV⁺96] Peter H Veltink, HansB J Bussmann, Wiebe De Vries, WimL J Martens, and Rob C Van Lummel. Detection of static and dynamic activities using uniaxial accelerometers. *IEEE Transactions on Rehabilitation Engineering*, 4(4):375–385, 1996.
- [VH05] Eric Von Hippel. Democratizing innovation: The evolving phenomenon of user innovation. *Journal für Betriebswirtschaft*, 55(1):63–78, 2005.
- [VNK15] Michalis Vrigkas, Christophoros Nikou, and Ioannis A Kakadiaris. A review of human activity recognition methods. *Frontiers in Robotics and AI*, 2:28, 2015.
- [VNKC09] Tam Van Nguyen, Jin Gook Kim, and Deokjai Choi. Iss: the interactive smart home simulator. In *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*, volume 3, pages 1828–1833. IEEE, 2009.
- [VS99] Viswanath Venkatesh and Cheri Speier. Computer technology training in the workplace: A longitudinal investigation of the effect of mood. *Organizational behavior and human decision processes*, 79(1):1–28, 1999.
- [WADX15] Andrew Whitmore, Anurag Agarwal, and Li Da Xu. The internet of things—a survey of topics and trends. *Information Systems Frontiers*, 17(2):261–274, Apr 2015.
- [WC96] Jennifer Widom and Stefano Ceri. *Active database systems: Triggers and rules for advanced database processing*. Morgan Kaufmann, 1996.
- [WHO18] WHO. Who/europe — healthy ageing - health and social care systems. <http://www.euro.who.int/en/health-topics/Life-stages/healthy-ageing/data-and-statistics/health-and-social-care-systems>, 11 2018. (Accessed on 12/11/2018).
- [Wie18] Bartosz Wiczorek. Light blockchain communication protocol for secure data transfer integrity. In *Cyberspace Safety and Security: 10th International Symposium, CSS 2018, Amalfi, Italy, October 29–31, 2018, Proceedings*, volume 11161, page 194. Springer, 2018.
- [WLT05] Jamie A Ward, Paul Lukowicz, and Gerhard Tröster. Gesture spotting using wrist worn microphone and 3-axis accelerometer. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, pages 99–104. ACM, 2005.
- [WP16] Brian Wolshon and Anurag Pande. *Traffic engineering handbook*. John Wiley & Sons, 2016.
- [WPP⁺07] Shiaokai Wang, William Pentney, Ana-Maria Popescu, Tanzeem Choudhury, and Matthai Philipose. Common sense based joint training of human activity recognizers. In *IJCAI*, volume 7, pages 2237–2242, 2007.
- [WR97] Michelle M Weil and Larry D Rosen. *Technostress: Coping with technology@ work@ home@ play*. Wiley New York, 1997.
- [WZ02] Hui-Fen Wang and You-Liang Zhang. Cad/cam integrated system in collaborative development environment. *Robotics and Computer-Integrated Manufacturing*, 18(2):135–145, 2002.
- [YJSB09] Allen Y Yang, Roozbeh Jafari, S Shankar Sastry, and Ruzena Bajcsy. Distributed recognition of human actions using wearable motion sensor networks. *Journal of Ambient Intelligence and Smart Environments*, 1(2):103–115, 2009.
- [YWC08] Jhun-Ying Yang, Jeen-Shing Wang, and Yen-Ping Chen. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern recognition letters*, 29(16):2213–2220, 2008.

- [YY06] Guang-Zhong Yang and Guangzhong Yang. *Body sensor networks*, volume 1. Springer, 2006.
- [YYP08] J. Yin, Q. Yang, and J. J. Pan. Sensor-based abnormal human-activity detection. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1082–1090, Aug 2008.
- [ZJCZ11] F. Zhou, J. R. Jiao, S. Chen, and D. Zhang. A case-driven ambient intelligence system for elderly in-home assistance applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(2):179–189, March 2011.
- [ZPR08] Wolfgang L Zagler, Paul Panek, and Marjo Rauhala. Ambient assisted living systems-the conflicts between technology, acceptance, ethics and privacy. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum fr Informatik, 2008.
- [ZS91] Ibrahim Zeid and R Sivasubramanian. *CAD/CAM theory and practice*, volume 6. McGraw-Hill New York, 1991.

Parte III

Publicaciones relacionadas con esta tesis

Lista de publicaciones

1. García-Magariño, Iván, Cárdenas-Bonett, Marlon, Gómez-Sanz, Jorge and Pérez Díez, Juan: Framework-supported mechanism of testing algorithms for assessing memory and detecting disorientation from IoT sensors. In 5th IEEE World Forum on Internet of Things (WF-IoT). Limerick, Ireland, April 15-18 2019. IEEE.
2. Rafael Pax, Jorge J. Gomez-Sanz, Ismael Sagredo Olivenza and Marlon Cárdenas-Bonett: A cloud based simulation service for 3d crowd simulations. In 2018 IEEE/ACM 22nd International Symposium on Distributed Simulation and Real Time Applications (DS-RT), pages 1 - 5, Oct 2018.
3. Marlon Cárdenas-Bonett, Jorge Gómez Sanz, and Juan Pavón: Testing ambient assisted living solutions with simulations. In IFIP International Conference on Testing Software and Systems (ICTSS), pages 56 - 61. 2018 Springer.
4. Ismael Sagredo-Olivenza, Marlon Cárdenas-Bonett, Jorge Gómez-Sanz, and Juan Pavón: Using graphs of queues and genetic algorithms to fast approximate crowd simulations. In Multidisciplinary Digital Publishing Institute Proceedings, volume 2, page 1216, 2018.
5. Marlon Cárdenas-Bonett, Noelia García Castillo, Jorge Gómez-Sanz, and Juan Pavón: Participatory design with on-line focus groups and normative systems. In Conference of the Spanish Association for Artificial Intelligence, pages 66 - 75. 2018 Springer.
6. Marlon Cárdenas-Bonnet and Susana Bautista. Infostate: Interactive visualisation of the environment using ambient sensors. In Proceedings of the XIX International Conference on Human Computer Interaction, Interacción 2018, pages 23:1 - 23:2, Palma de Mayorca, Spain. 2018, ACM.
7. Susana Bautista, Jelena Mazaj, and Marlon Cárdenas-Bonett: Developing RRI practices: The role of the ICT in stakeholders' knowledge exchange and co-creation processes. In Responsible Research and Innovation Actions in Science Education, Gender and Ethics, pages 21 - 26. 2018, Springer.
8. Rafael Pax, Marlon Cárdenas-Bonett, Jorge J. Gómez-Sanz, and Juan Pavón: Virtual development of a presence sensor network using 3d simulations. Smart Cities, pages 154 - 163, Cham. 2017, Springer.
9. Jorge J. Gomez-Sanz, Rafael Pax, Millán Arroyo, and Marlon Cárdenas-Bonett: Requirement engineering activities in smart environments for large facilities. Comput. Sci. Inf. Syst., 14(1): 239 - 255, 2017.

10. Jorge J. Gómez-Sanz, Marlon Cárdenas-Bonett, Rafael Pax, and Pablo Campillo:
Building prototypes through 3d simulations. PAAMS, pages 299 - 301, 2016.