



Sistemas Informáticos

Curso 2009 – 2010

Proyecto AVANTI:

Sistema de asistencia a la evacuación de incendios

Enrique López Mañas
Francisco Javier Moreno
Javier Plá Herrero

Dirigido por:
Pablo Gervás Gómez-Navarro
Gonzalo Méndez Pozo

Facultad de Informática
Universidad Complutense de Madrid

<AVANTI>

Proyecto de Sistemas Informáticos
Facultad de Informática

Universidad Complutense de Madrid

Autores:

Enrique López Mañas
Francisco Javier Moreno
Javier Plá Herrero

Profesores directores:

Dr. Pablo Gervás Gómez-Navarro
Dr. Gonzalo Méndez Pozo

Curso 2009 / 2010

Agradecimientos

El presente trabajo es el fruto de un año de dedicación, a través de los cuales hemos sufrido, disfrutado, peleado, trasnochado y esforzado. Desde el verano de 2009 en el que empezamos a hacer recopilación de ideas para nuestro proyecto hasta su cierre definitivo ha transcurrido un año de trabajo y esfuerzo. Sin embargo, este trabajo no habría sido posible sólo con el esfuerzo de los autores.

El primer agradecimiento va para nuestras familias, por el apoyo que nos han proporcionado no sólo durante este año, sino desde el inicio de nuestros estudios universitarios. No habría sido posible sin ellos. También queremos agradecer a Patricia Mateos su desinteresada ayuda en todo lo relativo a la imagen gráfica, que ha aportado el punto “artístico” a la aplicación, así como a Marina Beltrán por el apoyo y ánimo ofrecido durante todo el tiempo que ha durado el desarrollo. También a todos aquellos amigos y compañeros anónimos que, parapetados detrás de un café y una conversación, nos animaban a proseguir con la tarea y no decaer en los momentos críticos.

También queríamos agradecer a los Drs. Gonzalo Méndez y Pablo Gervás por su apoyo y orientación. Quizás haya sido el año en el que más hemos habilidades y técnicas de trabajo hayamos aprendido.

Se autoriza a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria como el código, la documentación y/o el prototipo desarrollado.

Enrique López Mañas Francisco Javier Moreno Nacarino Francisco Javier Plá

Palabras clave para búsquedas bibliográficas

- Realidad Aumentada
- Posicionamiento WiFi
- Ubiquitous Computing
- Ambient Intelligence
- Android

Tabla de Contenidos

Introducción

I. Resumen.....	1
1. Sobre AVANTI.....	2
2. About AVANTI.....	3
3. Sobre este documento.....	4
4. About this document.....	5
II. Estado del arte.....	6
1. De la computación ubicua a los sistemas de inteligencia ambiente.....	7
2. Realidad aumentada: un paso más en la conquista de la realidad.....	9
2.1. Algunas aplicaciones de la realidad aumentada.....	11
3. Posicionamiento y localización.....	12
3.1. Posicionamiento GPS.....	13
3.2. Triangulación con estaciones base.....	13
3.3. Posicionamiento WiFi.....	14
4. Mejora en el aprendizaje.....	15
5. Motivación y objetivo.....	16
6. Descripción de AVANTI.....	18

El producto

III. Características de AVANTI.....	20
1. Introducción.....	21
2. Servidor.....	24
3. Cliente.....	25
4. Modelado del espacio.....	26
4.1. Representación 2D del escenario.....	30
4.2. Representación 3D del escenario.....	31
5. Análisis de redes.....	39
5.1. K-Closest Neighbors.....	40
5.2. Sistema de partículas.....	41
5.2.1. Predicción.....	42
5.2.2. Corrección.....	43
5.2.3. Actualización de la partícula.....	43
5.2.4. Re-muestreo.....	44
5.2.5. Estimación.....	44
6. Cálculo de rutas.....	44
6.1. Algoritmo de Dijkstra.....	47
7. Descripción de los componentes.....	50
8. Modelo de la base de datos.....	57

El proceso

IV. Gestión del proyecto.....	59
-------------------------------	----

1. Planificación del proyecto	60
2. Gestión de riesgos.....	60
2.1. Primer paso: Identificación de riesgos	61
2.2. Segundo paso: Análisis / Planificación de riesgos	61
3. Especificación de casos de uso.....	66
3.1. Casos de uso del servidor	66
3.2. Casos de uso de la aplicación móvil.....	67
3.2.1. Menú principal de la aplicación	72
3.2.2. Casos de uso relativos a la actividad Wi-Fi	83
3.2.3. Casos de uso relativos al posicionamiento en el mapa.....	88
4. Análisis de requisitos	90
4.1. Servidor	90
4.1.1. Análisis de requisitos funcionales	90
4.1.2. Análisis de requisitos no funcionales: Software	91
4.2. Aplicación móvil.....	91
4.2.1. Análisis de requisitos funcionales	91
4.2.2. Análisis de requisitos no funcionales: Software	92
4.2.3. Análisis de requisitos no funcionales: Hardware	92
V. Desarrollo del proyecto	93
1. Objetivos principales.....	94
1.1. Iteración inicial.....	94
1.2. Primera iteración	94
1.3. Segunda iteración	94
1.4. Tercera iteración.....	95
1.5. Cuarta iteración	95
2. Proceso de trabajo	96
2.1. Fase inicial.....	96
2.2. Fase de elaboración	96
2.3. Fase de construcción	97
2.4. Fase de transición	99
VI. Tests	100
1. Introducción	101
2. JUnit	101
3. Descripción de los equipos utilizados para las pruebas	102
4. Pruebas sobre bases de datos	103
4.1. Pruebas de carga.....	103
4.2. Pruebas de rendimiento	103
5.- Pruebas de precisión del posicionamiento	108
 Conclusiones	
VII. Resultados.....	110
1. Discusión sobre los resultados obtenidos y alcanzados	111
1.1. Resultados positivos.....	113
1.2. Resultados negativos.....	114
2. Descripción de alternativas a las empleadas	115

3. Análisis de la complejidad final	115
4. Trabajo futuro.....	116
VIII. Apéndices	118
1. Estadísticas del proyecto	119
IX: Bibliografía y referencias.....	122

Capítulo I

Resumen

1 Sobre AVANTI

El soporte a las técnicas de evacuación en caso de incendio es un campo que todavía puede beneficiarse mucho de los últimos avances en tecnología, y particularmente en Inteligencia Ambiente (Ambient Intelligence), una evolución natural de lo que la década pasada se conoció como Computación Ubicua (Ubiquitous Computing). Se entiende por este término la integración de la informática en el entorno de la persona, de forma que los ordenadores no se perciban como objetos diferenciados. Su objetivo es insertar dispositivos inteligentes tanto en el entorno como en apartados de uso diario, para que las personas puedan interactuar con ellos de una manera natural y desinhibida en todo tipo de situaciones y circunstancias.

En nuestro caso, los ensayos de evacuación se venían llevando a cabo sin ningún soporte informático. Un sencillo procedimiento de “levántase y vaya hacia la salida” que en el mejor de los casos tan sólo era molesto para el que lo practicaba, obligándole a dejar sus quehaceres y dirigirse durante 5 minutos a la puerta de salida, sin posibilidad de poder mejorar en sus errores o de sentirse motivados.

Utilizando como escenario la Facultad de Informática, presentamos formalmente AVANTI. El proyecto trata de aportar una plataforma intuitiva pero completa y potente para usar durante los ensayos de evacuación. Combinando posicionamiento WiFi donde el GPS no llega, predicción de movimiento gracias a los sensores que proporcionan los terminales Android y Realidad Aumentada, para poder contribuir a una experiencia más visual y estimuladora, AVANTI intenta contribuir a que los ensayos de incendio puedan ser informatizados. Estudiantes e instructores puedan analizar fácilmente fallos en los protocolos a seguir, y que llegada la situación de un incendio los ensayos puedan cumplir su objetivo: ser eficaces, y evitar que ningún miembro de la Facultad de Informática sufra daños personales. Nuestro desarrollo ha dado lugar a la publicación de una contribución en la conferencia UcamI'10

2 About AVANTI

Technical support for evacuation protocols in fire situations is an area that can still greatly benefit from the latest advances in technology, particularly in Ambient Intelligence (AmI), a natural evolution of what the last decade became known as Ubiquitous Computing (UC). This term could be understood as the integration of information technology in the person's environment, so that computers are not seen as separate objects. Its goal is to insert smart devices on both the environment and in everyday devices, in order to enable people to interact with them in a natural and uninhibited manner in all situations and circumstances.

In the case of our Computer Science School, the evacuation tests were being carried out without computer support. Just by a simple "get up and go to exit" procedure, which at best was only valid to upset the user for losing his time and stop his tasks for 5 minutes. The fire evacuation drills did not encourage the motivation on the user, and did not provide any feedback mechanism to correct his own errors or improve the last trajectories of escape.

Using the scenario of the Computer Science School from the Complutense University, we proudly introduce AVANTI. The project aims to provide an intuitive but comprehensive and powerful platform to use during evacuation drills. Combining wireless positioning where GPS has not come, motion prediction using the Android-powered sensors and Augmented Reality in order to provide a more visual and stimulating experience, AVANTI attempts to contribute to the computerization of the fire evacuation drills, to help students and instructors to easily detect failures in the designed protocols, and when the situation of fire in the building happens, the drills can accomplish their goals: to be effective, and to prevent any member of the Computer Science School to suffer personal injuries. Our work has led into the publication of a paper in the UCamI'10 conference

3 Sobre este documento

El documento se encuentra organizado en cuatro grandes secciones, que aglutinan y organizan las distintas partes de la memoria. La primera sección es la introducción, en la que se hace un resumen de la problemática que se intenta resolver, de la aplicación y del estado del arte en los distintos campos que atañen a AVANTI. La segunda sección trata más a detalle el producto. Se describe la arquitectura modular y funcional de AVANTI, entrando en detalle en cada uno de sus componentes. La tercera sección define el proceso que se ha llevado a cabo para desarrollar el producto, formalizada según las distintas propuestas de Pressman [Press05] para metodología de desarrollo de software. La cuarta sección, conclusiones, evalúa los resultados obtenidos y aporta información relativa a las pruebas, estadísticas y apéndices de la aplicación, así como la bibliografía utilizada y en la que nos hemos basado para poder analizar la problemática desde el comienzo de este proyecto. El documento en su práctica totalidad (salvo las secciones que la normativa establece) ha sido escrito en castellano por motivos prácticos.

Los contenidos abarcan prácticamente todo el ciclo de desarrollo de AVANTI; así como un análisis previo de la problemática de nuestro problema, sistemas existentes y cómo hemos intentado puentear todas las alternativas para poder solucionar nuestra problemática concreta. Sobre AVANTI hemos redactado aspectos técnicos de ingeniería del software, como manuales de usuario, descripciones, capturas de pantalla de la aplicación, etc.

4 About this document

This document is organized into four sections, which organize the different parts of memory. The first section is the introduction, which provides a summary of the problem we have been trying to solve, the application and the state of the art in different fields regarding AVANTI. The second section deals more in detail with the product. We describe the architecture and the modular functionality of AVANTI, going into detail on each of its components. The third section defines the process that has been carried out to develop the product, formalized according to the different proposals from Pressman [Press05] for software development. The fourth section, the conclusions, evaluates the results and provides information concerning the tests, statistics of the application, appendix and the bibliography we have used in order to analyze the problematic we were trying to solve since the beginning of this project. The paper is almost entirely written in Spanish, for practical reasons.

Contents cover practically the entire development cycle of AVANTI, as well as a previous analysis of the faced problems, existing systems and how we tried to bypass all the alternatives to solve our specific problems. Regarding AVANTI we have written about the technical aspects of software engineering, user manuals, descriptions, screenshots of the application, etc.

Capítulo II

Estado del arte

1. De la computación ubicua a los sistemas de inteligencia ambiental

La Computación Ubicua (Ubiquitous Computing, o ubicomp) es un modelo que propugna la integración de dispositivos alrededor de escenarios donde se encuentre localizado el ser humano, en el que éste puede interactuar de manera natural con sus dispositivos y realizar cualquier tarea diaria de manera completamente transparente con respecto a sus computadores. Durante sus actividades ordinarias, alguien que esté “usando” computación ubicua (entrecorramos “usando” porque el usuario nunca lo hará directamente) lo hace a través de diversos dispositivos y sistemas computacionales simultáneamente, y generalmente lo hará sin percibirlo. Este modelo es visto como un paso más del paradigma de uso de ordenadores de escritorio.

Mark Weiser utilizó por primera vez el término Computación Ubicua sobre 1998, mientras trabajaba como jefe de desarrollo en el Centro de Desarrollo XEROS de Palo Alto. En colaboración con el director y jefe científico de PARC, John Seely Brown, Weiser escribió algunas publicaciones sobre el tema, definiendo y dando a conocer algunos de los primeros problemas relacionados con la computación ubicua [Weis91, Weis96, Weis99]. Este paradigma también ha sido descrito como Inteligencia Ambiente (Ambient Intelligence, o sencillamente AmI) [Hans03]. Cuando lo más concerniente del ámbito de estudio son los dispositivos involucrados en la interacción, también se le conoce como “computación física”, “Internet de las Cosas”, “Computación háptica” [WHC07] y “cosas que piensan”. En lugar de proponer una sola definición para la computación ubicua y todos los términos que han ido surgiendo a lo largo de los años, se ha propuesto una taxonomía de propiedades, por el medio de la cual se puede así mismo describir una amplia gama de sistemas que pueden ser calificados de ubicuos [Posl09].

Como punto común a todos los modelos de computación ubicua podríamos destacar el hecho de que comparten la visión de ser pequeños y disimulables, robustos y con capacidad para procesamiento de red, distribuidos en todas las escalas que comprende el día a día actual, y generalmente son integrables en nuestro entorno sin ser especialmente llamativos. Por ejemplo, un dispositivo de computación ubicua doméstico podría interconectar los sistemas de iluminación y calefacción con un control ambiente, de manera que en función de la evolución del momento del día y

sus características, este sistema reaccionase y pudiese variar la temperatura y condiciones de luz en una vivienda o edificio, de manera continua e imperceptible. Otra aplicación frecuente son frigoríficos que son conscientes de su contenido cuando ha sido convenientemente etiquetado, capaces de planificar menús saludables para toda la semana en función de las necesidades de cada miembro de la familia, y advertir a los usuarios de la comida rancia o en mal estado.

No obstante, este campo presenta ciertos desafíos ingenieriles e informáticos: en el diseño de sus sistemas, en los sistemas de modelado, interfaz de usuario. Los modelos de interacción persona-ordenador actuales, sean a través de consola de comandos, a través de menús o basados en una interfaz (GUI), son inapropiados e inadecuados para la computación ubicua. Esto sugiere que el paradigma de interacción que resulte natural para el ser humano todavía ha de emerger y ser definido (a pesar de que ya hay cierto reconocimiento en el terreno de que, en cierto modo, ya hemos comenzado a vivir en un mundo “ubicomp”). Los dispositivos actuales que con más fuerza y facilidad, dada su amplia presencia, están dando algún tipo de soporte y/o inspiración son los teléfonos móviles, GPS y ultraportátiles.

Mark Weiser ha propuesto [Weis91] tres modelos básicos que puedan ser considerados para desarrollar sistemas ubícuos:

- Tabs: dispositivos de escasos centímetros, que pueden ser llevados por un usuario
- Pads: dispositivos del tamaño de una mano
- Boards: dispositivos que pueden llegar a medir metros

Estas categorías propuestas por Weiser se caracterizan generalmente por ser grandes, tener una forma plana o incorporan algún tipo de salida visual. Si se relajan estas consideraciones (permitiendo aceptar, por ejemplo, dispositivos de hasta el tamaño de nanómetros) se puede extender este rango hasta un número de dispositivos mucho mayor, y también potencialmente hasta un número de dispositivos mucho más útiles. Por lo tanto, con el tiempo se han acabado proponiendo otros tres tipos de clasificaciones:

- Dust(polvo): dispositivos miniaturizados que pueden no tener algún tipo de salida visual (por ejemplo, los sistemas Micro-electromecánicos MEMS), cuyo tamaño puede oscilar entre nanómetros hasta micrómetros o milímetros.

- Skin (piel): pueden ser fabricados con capacidades de emitir luz y con diversos materiales, como polímeros conductivos, algunos dispositivos orgánicos, etc... Se ven frecuentemente como ropa, cortinas, o diversos elementos de decoración
- Clay (arcilla): conjuntos de distintos MEMS pueden combinarse para crear formas en tres dimensiones.

En [Cast96], el autor entiende que nos estamos moviendo de un sistema casi descentralizado, compuesto por micro-ordenadores y mainframes hacia un sistema completo de computación ubicua. Castells usa internet como un ejemplo de computación pervasiva extrema. La progresión lógica de ese paradigma es un sistema donde el trabajo en red se vuelve aplicable en cualquier situación de la vida diaria, en cualquier localización y en cualquier contexto. Castells prevé para el futuro un sistema donde haya miles de millones de dispositivos miniaturizados e intercomunicados.

2. Realidad Aumentada: un paso más en la conquista de la realidad

La realidad aumentada (Augmented Reality, o AR en inglés) es un término que se emplea para describir aquellos sistemas en los que se enriquece la realidad que percibimos con nuestros propios medios con elementos o información que han sido creados virtualmente o generados por ordenador. La investigación en realidad aumentada explora posibles escenarios donde la información e imágenes generadas por ordenador pueden ser útiles para expandir el mundo real. El desarrollo más avanzado puede incluir el uso de displays acoplados sobre la cabeza o visores para los ojos, y la construcción de entornos controlados que contengan cualquier número de sensores.

Hoy día se aceptan como válidas dos definiciones de realidad aumentada. Una fue proporcionada por Ronald Azuma en 1997. La definición de Azuma [Azum97] básicamente establece tres puntos:

- Tiene que combinar lo real y lo virtual
- Tiene que ser interactivo en tiempo real
- Tiene que ser en 3 dimensiones

También Milgram y Kishino propusieron una definición [MiKi94]. Describieron un “continuum” que iba desde el entorno real hasta un entorno virtual puro. Entre medias se podían encontrar entornos de realidad aumentada (más próximo a la realidad que al mundo virtual) y entornos de virtualidad aumentada (más próximos a un mundo virtual que a un mundo real. En la Figura 1 puede verse una imagen que representa este modelo

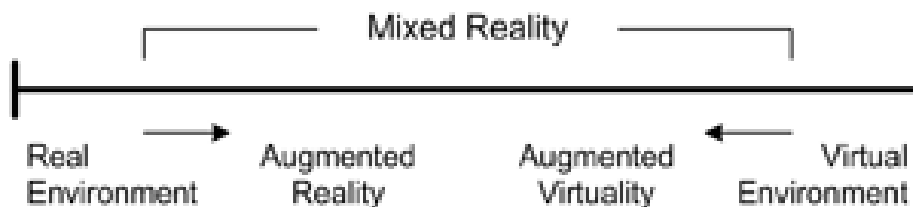


Figura 1: definición del continuum acorde a [MiKi94]

Este “continuum” se ha extendido hasta un plano de dos ejes, que representa la “virtualidad” y la “medialidad”[Maan02]. Según se desprende de la figura 2, el origen de coordenadas R representa la realidad. Según se avanza a lo largo del eje de la virtualidad, empezamos a incorporar elementos virtuales sobre la realidad (realidad aumentada). Esta taxonomía, sin embargo, ofrece también modificar elementos de la virtualidad o la realidad, o alguna combinación de estas. Si seguimos subiendo por el eje de la “medialidad”, podemos alcanzar otros sistemas como realidad medializada, o virtualidad medializada.

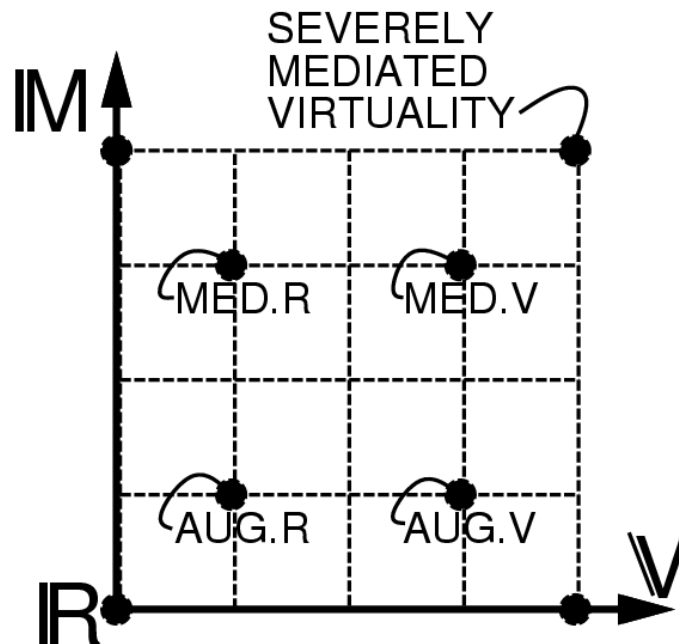


Figura 2: representación de las variaciones sobre la realidad introduciendo el concepto de medialidad.

2.1. Algunas aplicaciones de la realidad aumentada

- **Publicidad:** Algunos fabricantes han usado técnicas de realidad aumentada para publicitar sus productos. Por ejemplo, en el LA Auto Show del 2008, Nissan dió a conocer el concepto de vehículo Cubo y presento a los visitantes un folleto con un marcador, que cuando se mostraba en frente de una cámara web mostraba diferentes versiones del vehículo. En Agosto del 2009, Best Buy¹ utilizó exáctamente la misma técnica con sus clientes.
- **Medicina:** la realidad aumentada podría mostrar información sobre objetos que son a priori inaccesibles al ojo humano, lo que puede ser particularmente efectivo de cara a intervenciones quirúrgicas, diagnóstico, etc. Por ejemplo, un

¹ <http://www.engadget.com/2009/08/06/best-buy-goes-3d-even-augmented-reality-isnt-safe-from-adverti/>

doctor podría mostrar una imagen de su feto cuando está analizando a la madre en una revisión habitual. Hay un grupo de desarrollo en este campo en la universidad de North Carolina².

- **Dispositivos de navegación:** la realidad aumentada puede mejorar los dispositivos de navegación para una variedad de aplicaciones. Por ejemplo, se podría utilizar para navegar en el interior de edificios con el fin de mantener las instalaciones industriales, o al aire libre para operaciones militares o gestión de desastres.
- **Entretenimiento y educación:** se puede usar en este terreno para crear objetos virtuales en visitas guiadas a museos, o en exposiciones temáticas. Se ha usado de hecho en bastantes videojuegos a lo largo de los últimos años, como pueden serlo el ARQuake o The Eye of Judgement³.
- **Música:** el grupo de pop Duran Duran incluyó proyecciones de realidad aumentada durante su tour “Pop Trash”, en el 2000.[PWCG02] Lost Valentines lanzaron el primer vídeo interactivo que utilizaba técnicas de realidad aumentada en el 2009. Los usuarios se podían imprimir 5 marcadores que representaban a cada uno de los artistas del grupo, e interactuar con ellos a través de la webcam e internet⁴.

3. Posicionamiento y localización

Según las redes de teléfonos y otros dispositivos móviles va creciendo, el posicionamiento para estos terminales cobra una importancia cada vez mayor. De hecho, un porcentaje muy importante de las aplicaciones requieren una información certera sobre la información de localización del terminal (la anteriormente mencionada computación ubicua, aplicaciones basadas en contexto, situaciones de emergencia, etc). Mientras existen muchas soluciones para exteriores, basadas en GPS o AGPS, estas mismas soluciones no son aplicables para lugares cubiertos e interiores de edificios. La señal recibida es demasiado débil para poder obtener información

² <http://www.cs.unc.edu/Research/us/>

³ <http://www.eyeofjudgment.com/>

⁴ <http://www.gizmodo.com.au/2009/10/sydney-band-uses-augmented-reality-for-video-clip/>

exacta sobre la posición actual. No obstante, dado que cada vez hay más edificios equipados con puntos de acceso para proveer conexión a Internet inalámbrica, se abre una nueva posibilidad al utilizar estos dispositivos para determinar la localización del usuario en entornos de interiores.

3.1. Posicionamiento con GPS

GPS ⁵(del inglés Global Positioning System) es un sistema de navegación por satélite que proporciona información precisa sobre la localización de un terminal, y está preparado para otra serie de usos como información meteorológica. Resumiendo bastante la parte técnica, puede ser usado en cualquier lugar donde hay línea directa y sin obstáculos hacia por lo menos cuatro satélites de la red GPS. Está mantenido por el gobierno de los Estados Unidos, y es accesible de manera gratuita a cualquier individuo que disponga de un receptor GPS.

El GPS fue originalmente desarrollado por el Departamento Americano de Defensa (dod), e inicialmente basaba su funcionamiento en la señal recibida de 24 satélites. Se empezó a utilizar en 1973 para hacer frente a todos los problemas de navegación de la época.

Algunos países permiten el uso de repetidores GPS para permitir la recepción de señales GPS en interiores, o en localizaciones donde la cobertura no llega como se desearía. No obstante, las leyes de la Unión Europea y del Reino Unido prohíben explícitamente el uso de estos dispositivos, ya que las señales pueden causar interferencias en otros dispositivos.

Dados los obstáculos que el GPS encaraba a la hora de realizar el sistema de posicionamiento para AVANTI, fue rápidamente descartado en favor de otras tecnologías que no presentasen las mismas limitaciones.

3.2. Triangulación con estaciones base

Este tipo de triangulación utiliza las estaciones base de telefonía GSM para proporcionar una ubicación aproximada con 200-1000 metros de precisión.

⁵ <http://www.gps.gov/>

Generalmente se utiliza como un complemento a los sistemas de posicionamiento WiFi (WPS) o al GPS allí donde la cobertura de estos últimos no es lo suficientemente potente. La compañía Skyhook⁶, responsable de una API para usar posicionamiento por WPS, mantiene una base de datos mundial con información sobre estaciones base, lo que incrementa la cobertura de áreas usando XPS (sistemas híbridos de posicionamiento) y ayuda a mejorar el tiempo de adquisición de datos cuando se usa GPS. Esta triangulación también puede ayudar en la optimización de recursos energéticos, activando o desactivando los distintos sistemas que participan el XPS dependiendo de las condiciones ambientales y de los sistemas disponibles.

Dado el margen de error de este tipo de triangulación, y considerando que en las pruebas de campo realizadas el margen de error para el posicionamiento estaba más cercano a los 1000 metros que a los 200, lo descartamos y seguimos buscando alternativas.

3.3. Posicionamiento WiFi

El posicionamiento basado en WiFi (WPS, WiFi Positioning System) surge como una idea que pueda resolver los problemas anteriormente citados, aprovechando el rápido crecimiento de los puntos de acceso en áreas urbanas. SkyHook Wireless es la compañía puntera en este tipo de servicio, manteniendo una base de datos pública a la que se puede acceder a través de una API, y obtener el posicionamiento basado en los puntos de acceso accesibles desde un terminal. SkyHook proporciona su sistema a los dispositivos basados en Apple, aunque ha habido otras iniciativas similares, como la del Fraunhofer Institute [SPSH97] o Google⁷

La técnica de localización utilizada en posicionamiento con puntos de acceso inalámbricos se basa en medir la intensidad de la señal recibida (Received Signal Strength en inglés, RSS) y el método de “fingerprinting”[BaPa00][ChKo02]. La precisión depende del número de posiciones que se hayan introducido en la base de datos. Las posibles fluctuaciones de la señal que puedan ocurrir pueden llegar a introducir errores y faltas de precisión en la trayectoria del usuario. Para minimizar las fluctuaciones de la señal recibida, se aplican ciertas técnicas para filtrar ruidos. Hay nuevas leyes y regulaciones que se están imponiendo a operadores de VO-IP para que

⁶ <http://www.skyhookwireless.com>

⁷ http://www.google.com/googleblogs/pdfs/google_submission_dpas_wifi_collection.pdf

integren algún sistema en sus puntos de acceso que permita determinar la posición de los terminales en un entorno dado⁸.

4. Mejora en el aprendizaje

La utilidad final de haber incluido técnicas de simulación en nuestro proyecto de fin de carrera ha sido poder utilizar técnicas de e-learning, que mejoren la motivación y el aprendizaje de los usuarios de la aplicación. El e-learning es un viejo conocido de la informática. Una descripción bastante acertada del fenómeno de los videojuegos y simulaciones siendo aplicados a la educación se puede encontrar en el libro de Marc Prensky “Digital Game Based Learning” [Prens01] En este libro se inicia una discusión sobre cómo los estudiantes y el proceso de aprendizaje ha cambiado, y cómo funcionan los videojuegos cuando se tienen que dedicar a enseñar. Los juegos y las simulaciones no se usan únicamente con propósitos lúdicos. Muy frecuentemente se pueden usar también para entrenar y enseñar en diferentes áreas, con resultados bastante óptimos. Es fácil poder apreciar la relación y las conexiones que podemos encontrar entre objetos completos, como grupos sociales o estrategias alternativas, para resolver problemas [RaWa07]. En el caso de las simulaciones sociales, los juegos pueden proveer un entorno de experimentación seguro que puede ayudar a adultos a aprender y explorar habilidades, métodos y conceptos [Rayb05]. Es fácil entender cómo estos escenarios seguros pueden ser particularmente provechosos a la hora de enseñar técnicas de evacuación de incendios, ya que no necesitamos prender fuego a un edificio para que los estudiantes puedan realizar sus prácticas.

Dentro de los juegos educativos son especialmente conocidos los denominados “juegos serios (serious games). La palabra “serio” hace referencia a que el principal objetivo de esta clase de juegos no es el entretenimiento, sino otro tipo de objetivos tales como industrias, planificación de ciudades o ingenierías. Los juegos serios han tenido unos beneficios aproximados de 1.5 billones de dólares en el 2008 [Derr07]. Una mejor definición podría ser que los juegos serios usan el entretenimiento y la diversión para poder conseguir una mejora de las habilidades en los campos anteriormente mencionados. Recubrir un núcleo de aprendizaje con un entorno o escenario de juego o de simulación es una manera probada de conseguir motivar a los estudiantes o futuros estudiantes de ese juego. De nuevo, Marc Prensky cuenta en su

⁸ April 2005, <http://www.telecomweb.com/news/1112721769.htm>.

libro algunas historias de éxito cuando aplicó escenarios de videojuegos para ayudar a algún colectivo a obtener el éxito en un determinado proceso de aprendizaje. Particularmente, resulta interesante resaltar uno de sus primeros casos de estudio: "How to Get 3 Million Engineers to Learn". En este escenario, Marc Prensky explica cómo consiguió convencer a un grupo de ingenieros para probar una nueva aplicación CAD. La mayoría de los ingenieros se oponía fuertemente a "volver a la escuela" para aprender una aplicación nueva, cuando multitud de aplicaciones CAD estaban en el mercado disponibles desde hacía un montón de años, y consideraban que el esfuerzo de volver a aprender cómo utilizar una plataforma no les resultaba rentable. Después de construir un videojuego alrededor de la aplicación CAD para motivar a los ingenieros a probar el nuevo producto, obtuvo un éxito destacable. El juego llegó a ser popular en el circuito amateur, donde mucha gente acabó probándolo. En [Zyda05] se describe como una simulación de guerra, America's Army, fue usado para poder mejorar las habilidades de algunos reclutas del ejército americano. Cuando estos reclutas probaban la plataforma, generalmente mejoraban sus resultados en una amplia variedad de tests.

Los "juegos serios" se han utilizado en multitud de áreas y campos, con la idea de que proporcionan una experiencia inmersiva y atractiva. Considerando que todos estos estudios empíricos se realizaron para poder determinar la eficacia del enfoque, el estado del arte de los juegos y simulaciones se ha movido rápidamente de la "idea" al "hecho". No hay más discusión al respecto de si son eficientes en si, sino de si el coste de construir sistemas basados en esta alternativa es asumible (teniendo en cuenta que muchos de estos juegos se desarrollan en ámbitos académicos, donde se cuenta con un poder adquisitivo limitado. Y este es, de hecho, el principal argumento de los detractores de los juegos como herramienta educativa. Los juegos educativos tienen, además, sus propias necesidades y requisitos, que no siempre convergen con las necesidades y requisitos de los videojuegos tradicionales [KiMc04].

5. Motivación y objetivo

La necesidad de llevar a cabo ensayos para la evacuación de incendios en edificios públicos es una medida básica de seguridad, universalmente aceptada. Estos ensayos pretenden familiarizar a los ocupantes de un edificio con los protocolos de evacuación, las rutas de salida y los puntos de reunión. Generalmente estos ensayos

requieren de un esfuerzo de coordinación importante, al estar todo el personal del edificio evacuando en el mismo instante, hacia una misma localización de salida predeterminada, y con el personal de organización guiando a la gente. Ya que estos ejercicios suelen requerir un gran coste monetario para llevarlos a cabo [BMMV06] y una distracción de la rutina diaria del edificio en cuestión que afecta de hecho a la productividad de los trabajadores que allí se encuentran, estos ensayos suelen llevarse a cabo con una frecuencia inferior a la que sería deseable. Para reducir este problema, se han aplicado técnicas de “juegos serios” y “simulaciones virtuales” para poder permitir simulaciones de incendios en un entorno computerizado, reduciendo riesgos y costes [RCSZ06], [ChRa09], [John10]. Las soluciones planteadas permitían a los usuarios ejercitar el procedimiento de evacuación, y explorar las posibles rutas de salida en un entorno virtual y controlado. No obstante, Chittaro y Ranon dieron a conocer que usuarios que habían probado sus aplicaciones (aplicaciones basadas en un entorno de simulación en 3D) tenían una percepción distorsionada de la realidad en comparación con aquellos que habían realizado el ensayo en espacio real. El principal motivo de queja de los jugadores es que el sistema les permitía moverse a velocidades más lentas de la que podrían moverse en un entorno real. Chittaro y Ranon mencionaron que esto podría ser una combinación de la tendencia observada de la gente de infravalorar el tiempo general durante una evacuación de incendios (que se suele pasar de manera mayoritaria en las escaleras) y de expectativas que distan bastante de la realidad concernientes a la velocidad de desplazamiento que se adquiere debido al uso de juegos recreativos, donde la física empleada dista bastante de la conocida en el planeta Tierra.

En AVANTI hemos intentado proponer una solución que combine las ventajas de las simulaciones virtuales –la posibilidad de mantener un log o registro de la evacuación y la posibilidad de bloquear rutas de salida específicas –con las ventajas de una evacuación real –la velocidad de movimiento viene determinada en cada instante por el movimiento de cada uno de los usuarios o jugadores-, y usando el edificio real sobre el que se llevará a cabo la evacuación, en lugar de un modelo en 3D. El sistema propuesto ha utilizado técnicas Wireless para posicionamiento en interiores, y realidad aumentada para poder sobre-imprimir información en el entorno, mientras se puede observar la realidad a través de la pantalla del dispositivo móvil. El acelerómetro del teléfono se usa para recolectar información que permita estimar el movimiento del usuario, basado en estadísticas y mediciones previas, en aquellas áreas que no disponen de una señal certera, conocidas como áreas de sombra.

6. Descripción de AVANTI

AVANTI se ha desarrollado con el objetivo de poder estudiar la combinación de tecnologías pertenecientes a la Inteligencia Ambiental con otras como la Realidad Aumentada y poder determinar qué conflictos pueden surgir de combinar estas dos líneas. Usando todas estas tecnologías pretendemos que AVANTI pueda conseguir efectos de motivación sobre usuarios de la plataforma, que se tomen más en serio los ensayos para la evacuación de incendio (y por qué no, también como un juego dadas sus características de simulación) para que puedan mejorar sus habilidades sobre evacuación de edificios, y tengan presentes todas y cada una de las reglas en el fatídico momento en el que necesiten ponerlas en marcha.

La realidad aumentada nos permite mostrar fuegos virtuales en un contexto de simulación, con lo que podemos reducir todos los riesgos que tendría un simulacro de incendio real (impensable!) y beneficiarnos de las ventajas de un simulacro in-situ. El uso de plataformas móviles nos permite el movimiento y la interacción sin la necesidad de tener que estar conectados permanentemente a un ordenador de escritorio, pero necesitamos solventar un problema importante: el posicionamiento en interiores (algo para lo que el GPS, alternativa estándar para el posicionamiento, no funciona).

AVANTI sigue el paradigma de la arquitectura cliente-servidor. Después del login, el cliente establece una conexión con el servidor, y la conexión es creada. Esto significa que toda la información de perfil es recibida del servidor, así como la necesaria para el buen funcionamiento de la aplicación. El cliente se encuentra permanentemente conectado al servidor, enviando información sobre su posición (para que el servidor pueda registrar el historial de posiciones). El servidor mantiene una base de datos sobre la que se lleva a cabo el posicionamiento, siendo ésta previamente almacenada en el terminal, así como el resto de información necesaria para hacer funcionar la aplicación.

Los fuegos de la aplicación se generan utilizando modelos Wavefront pintados en el terminal mediante OpenGL ES. Se barajó utilizar sistemas de partículas así como modelos collada, pero debido al poco desarrollo para Android en estas tecnologías, y la sencillez de los objetos Wavefront, se optó por esta última.

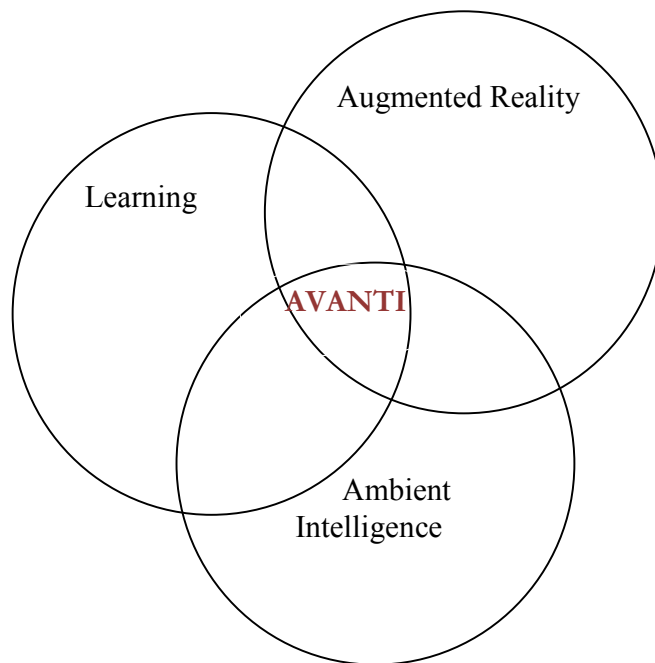


Figura 3: Diseño conceptual de AVANTI

Capítulo III

Características de AVANTI

1. Introducción

Este apartado presenta las características de AVANTI con más detalle: describe la arquitectura del sistema a nivel de módulos, explicando en qué dirección se realiza la comunicación. Desgranamos la parte del cliente y la parte del servidor, explicando qué funcionalidad es característica a cada una de las capas y qué comunicación se ejecuta durante las acciones típicas. Posteriormente, explicamos cómo hemos modelado el espacio 3D de nuestro testbed, la Facultad de Informática, para poder representarlo tanto en forma de mapa como en un espacio tridimensional que se combina con la parte de realidad aumentada para poder situar los fuegos sobre la pantalla (un escenario 2D) pero de manera que los usuarios pueden tener una percepción de distancia, profundidad y volumen. La parte en la que se explican los algoritmos de posicionamiento bajo redes WiFi es fundamental en este proyecto. Hemos usado como referencia los trabajos de [EvMa06] y [WLSB07]. El primero propone una solución basada en el algoritmo de los k-vecinos más cercanos, mientras que el segundo propone una solución basada en un filtro de partículas. Hemos realizado una implementación de ambos sistemas, obteniendo unos resultados favorables que se detallan en el capítulo de conclusiones con más detalle.

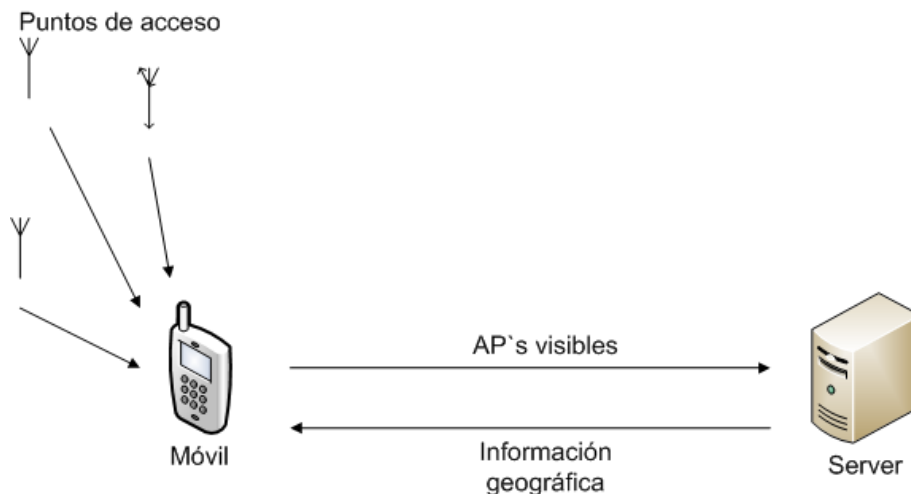


Figura 4: Esquema de funcionamiento

La Figura 4 muestra a grandes rasgos cómo fue el diseño inicial de la aplicación. El

usuario se va desplazando con el terminal móvil, que va recogiendo datos de los puntos de acceso (AP's) que escucha. Esta información (identificador del AP y potencia con la que se escucha) es enviada al servidor, que realiza el tratamiento de datos para poder averiguar la posición actual, con el algoritmo que haya sido seleccionado. Esta posición (X,Y) es enviada al terminal de nuevo, que puede entonces posicionarse sobre el mapa y realizar el pintado de los fuegos en la posición que le corresponda.

En realidad, tras una serie de pruebas sobre el terreno, pudimos observar que, al enviar el terminal móvil continuamente información hacia el servidor para poder obtener información actualizada sobre la posición actual, bloqueaba y ralentizaba de manera irremisible el mismo, provocando un efecto bola de nieve que acababa por desestabilizar el servidor. El modelo que queríamos seguir al utilizar el paradigma cliente-servidor era “cálculo de datos en el servidor, mostrarlos en el dispositivo”. Es decir, dábamos por hecho que la potencia de cálculo del dispositivo móvil iba a ser mucho menor, y queríamos mover estos datos hacia un servidor centralizado que dispusiese de mayor potencia y no se viese afectado en gran medida. No obstante, no contamos inicialmente con la frecuencia de actualización necesaria: era uno de los puntos críticos de la aplicación. Si tratamos de escapar de un incendio, no podemos permitirnos fallos en la actualización de la posición, por lo que queríamos que este dato fuese tan completo como pudiésemos. Por lo tanto, decidimos mover el cálculo de la posición hacia el propio terminal. Para ello, al iniciar sesión el terminal recibe del servidor la base de datos con toda la información sobre localizaciones y realiza el cómputo de datos en el propio terminal. Fue nuestra opinión subjetiva que el proceso era mucho más cómodo y rápido en contra de lo que inicialmente habíamos aseverado, por lo que decidimos continuar de esta manera. El terminal seguía contactando con el servidor para recibir la posición de los fuegos

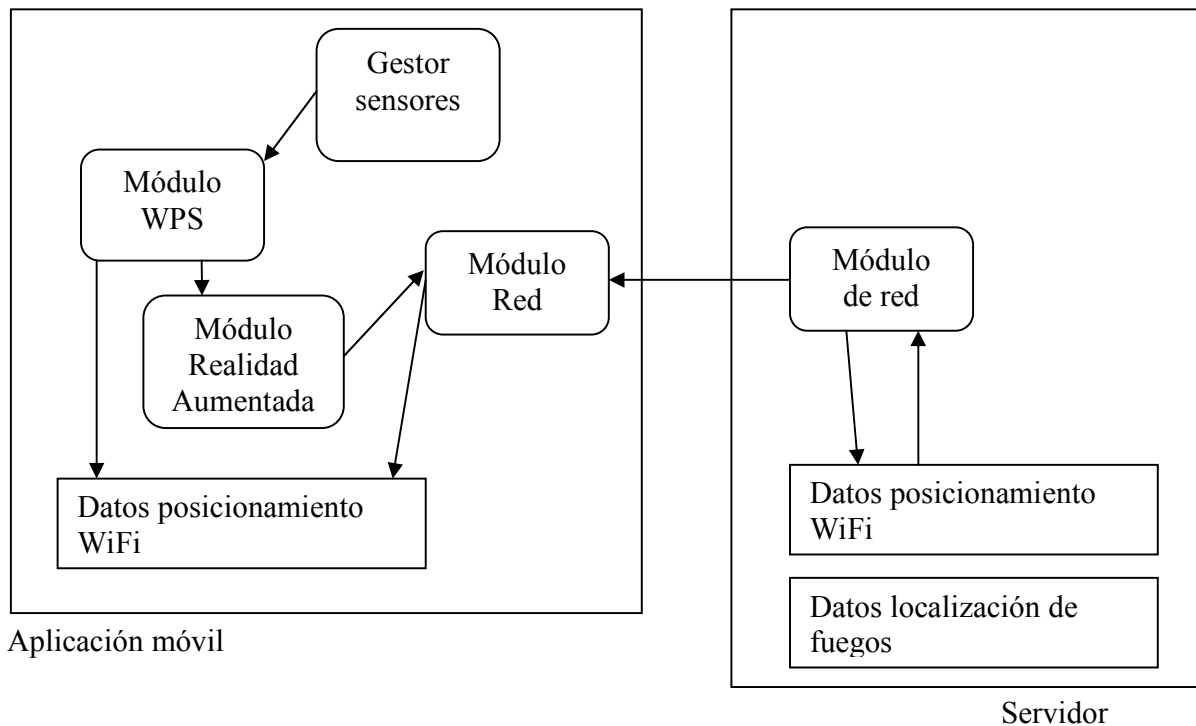


Figura 5: Esquema de funcionamiento a nivel de módulos

La Figura 5 explica cómo funciona la aplicación, a un nivel más modular. La comunicación entre ambas entidades se realiza a través del módulo de red. Inicialmente, los datos de posicionamiento WiFi se cargan en el dispositivo móvil. El módulo WPS, en base a la entrada que recibe de los AP's que están a la vista y los datos almacenados, genera unas coordenadas geográficas que se corresponden con la posición actual del usuario. Cuando el dispositivo atraviesa por una "zona oscura" o de baja cobertura, donde el posicionamiento no puede realizarse de manera correcta, solicita al módulo que gestiona los sensores información previa sobre el movimiento del usuario, para poder realizar una predicción sobre la zona en la que se encontrará el usuario mientras dure la zona oscura. El módulo de realidad aumentada recoge del módulo WPS las coordenadas espaciales donde se encuentra el dispositivo y solicita al módulo de red información sobre los fuegos. Tras realizar pruebas, encontramos que actualizando esta información cada **dos** segundos no afectaba negativamente al rendimiento del terminal y del servidor, por lo que decidimos tomarlo como valor estándar. En función de los resultados que reciba el módulo de realidad aumentada, realiza el pintado de los fuegos tal y como se explica en el apartado 4 del capítulo III.

2. Servidor

El servidor se encuentra corriendo de manera permanente en una máquina del departamento. Aparte de la función que ya se ha explicado de almacenamiento de datos, se ha diseñado una interfaz que permite insertar fuegos que el usuario irá visualizando en tiempo real, así como obtener información sobre cada uno de los usuarios de la aplicación y fuegos. Particularmente, se puede acceder a información sobre la trayectoria actual y sobre trayectorias antiguas, evacuaciones completadas con éxito, evacuaciones que no han sido exitosas y la posición actual.

Para el momento en el que el instructor o usuario de la instrucción ha de insertar fuegos se ha diseñado un sencillo sistema de control para evitar que se inserten fuegos en áreas en las que carece de lógica (interior de paredes o puertas, por ejemplo). Utilizando un mapa de bits que es una representación del mapa de bits de la facultad en blanco y negro, cada vez que se hace click se comprueba que no se ha hecho sobre un píxel negro. Si así ha sido, se muestra un mensaje de error al usuario y no se toma ninguna acción en la base de datos. En cualquier otro caso, el fuego se inserta de manera exitosa. Esta técnica se usa también para poder corregir posibles errores de posicionamiento en el caso del usuario. Si en algún momento dado los datos de posicionamiento nos indican que el usuario se encuentra atravesando alguna de estas zonas prohibidas, automáticamente recolocaremos al usuario en la zona válida más próxima a la posición actual.

El servidor ha sido desarrollado utilizando el motor de Google GWT⁹. No es el objetivo de este documento enumerar las ventajas de AJAX, pero mediante el uso de esta técnica nos podemos permitir que la página actualice en tiempo real el estado de fuegos y usuarios sin necesidad de que la página deba recargar continuamente, lo que contribuye de manera decisiva a la usabilidad del sitio. En la Figura 6 se puede observar el aspecto gráfico de la interfaz.

⁹ <http://code.google.com/webtoolkit/>

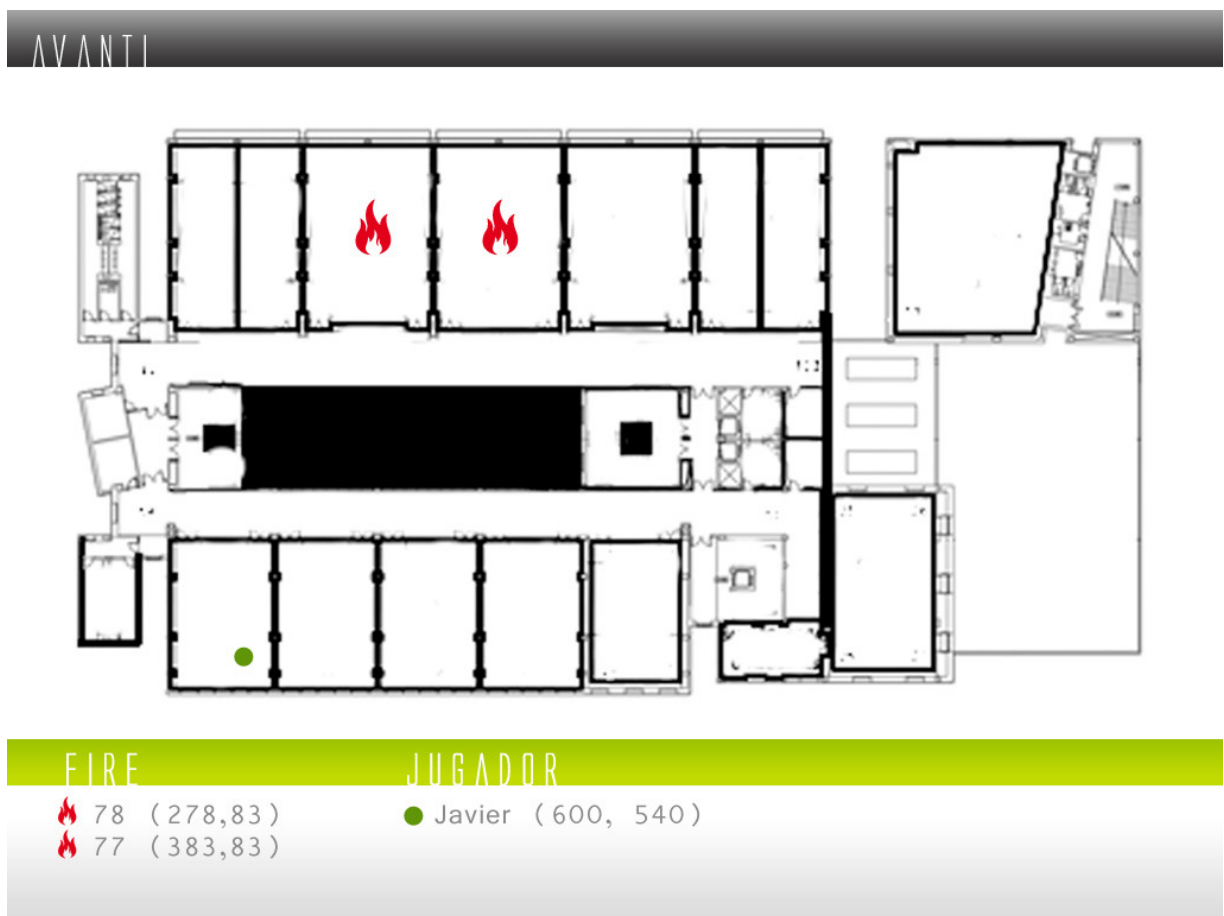


Figura 6: Captura de pantalla de la interfaz

3. Cliente

El desarrollo para el cliente se ha realizado utilizando Google Android¹⁰. Fue considerado junto a un dispositivo iPhone¹¹ de Apple. La segunda alternativa fue descartada por diversos motivos:

- No disponía de trabajos previos realizados utilizando Realidad Aumentada con la misma calidad y cantidad que Android, por lo que decidimos no correr ese riesgo.

¹⁰ <http://developer.android.com>

¹¹ <http://www.apple.com/de/iphone/>

- Los dispositivos Apple no son código abierto, por lo que algunas funciones resultan más “oscuras” y se dispone de menos recursos entre la comunidad
- Para desarrollar de manera legal con Apple hay que pagar una licencia anual, licencia que es inexistente para el caso de Android
- Los dispositivos Apple tienen un costo superior a los dispositivos Android, algo que en nuestro ámbito es relevante.

Como se explica en el capítulo de conclusiones, fue una decisión que en la etapa de finalización del proyecto consideramos acertada.

4. Modelado del espacio

Este apartado proporciona una pequeña introducción a OpenGL y las transformaciones geométricas. Dado que no es el propósito principal de este documento, se proporcionan los datos básicos para comprender el siguiente punto. Si se quiere más información al respecto se pueden consultar multitud de enlaces en Internet.

Una transformación geométrica es una operación del mismo tipo que permite generar un nuevo escenario a partir de uno previamente dado. En un nivel puramente matemático, las transformaciones pueden ser:

- Directas: el homólogo conserva el sentido del original en el plano cartesiano.
- Inversas: el sentido del homólogo y del original son contrarios.

Las transformaciones también pueden clasificarse de acuerdo con la forma del homólogo con respecto a la original en:

- Isométricas: el homólogo conserva las dimensiones y ángulos. También se pueden denominar movimientos: estos son simetría axial y puntual, rotación y traslación
- Isomórficas: el homólogo conserva la forma y los ángulos, y existe proporcionalidad entre las dimensiones del homólogo con el original.

- Anamórficas: cambia la forma de la figura original. Una de ellas es la inversión

A través de la API de OpenGL podemos manejar un escenario geométrico de dos, tres y cuatro dimensiones. Los tipos de transformaciones ofrecidos a través de la API son:

- Proyección: Transforman una escena 3d "abstracta", en una imagen plana que puede ser visualizada en una pantalla.
- Viewport: Ajustan el producto de la proyección a las dimensiones de un rectángulo contenedor (ventana).
- De vista: Que definen y afectan la posición desde la cual se visualizan las escenas tridimensionales.
- Modelado: Sirven para manipular los objetos en la escena, permitiendo trasladarlos, rotarlos y deformarlos (escalarlos).
- Modelo-Vista: Son la combinación de las dos transformaciones anteriores, que desde un punto de vista práctico son semejantes.

De las transformaciones explicadas, la más importante para AVANTI será la de Vista. Considerando nuestro terminal como una cámara, tanto real como en términos de OpenGL, podemos llegar rápidamente a la conclusión de que, generando un escenario que contiene fuegos localizados en su interior y desplazando la cámara de OpenGL conforme al movimiento del terminal móvil, tan sólo será necesario aplicar una transformación de vista para ubicar de manera realista los focos de incendio respecto al observador.

Las transformaciones de vista deben preceder las de modelado forzosamente, las demás pueden ser especificadas en cualquier momento, antes de que se haga el dibujo. Los vértices se componen de 4 coordenadas, estas son (x,y,z,w) . Si la coordenada w es diferente de cero, entonces las coordenadas corresponden al punto tridimensional $(x/w,y/w,z/w)$.

Para acomodarnos al modelo de OpenGL utilizaremos una función de especial relevancia, *gluLookAt*(*eyeX*, *eyeY*, *eyeZ*, *centerX*, *centerY*, *centerZ*, *upX*, *upY*, *upZ*). Según la página oficial de OpenGL¹², podemos describir *gluLookAt()* y cada uno de sus

¹² <http://www.opengl.org/resources/faq/technical/viewing.htm>

parámetros como sigue:

- Los tres primeros parámetros indican la posición del ojo o cámara desde el que se contempla la escena
- Los siguientes tres parámetros representan el vector hacia el que se dirige el ojo o cámara
- Los últimos tres parámetros indican el vector up, es decir, el giro de la cámara.

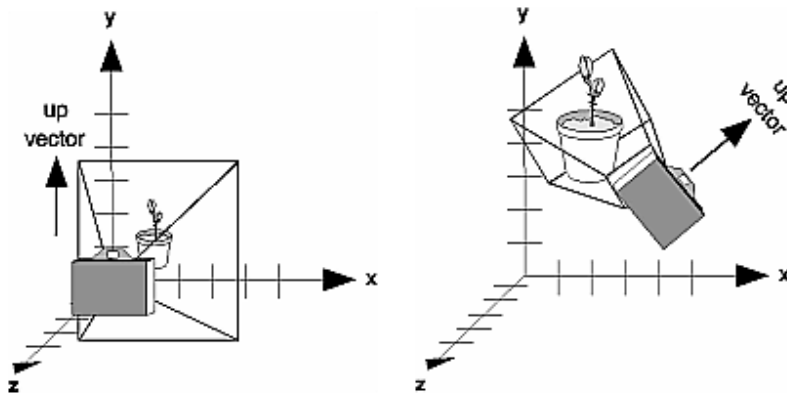


Figura 7: Ejemplo ilustrativo de espacio de coordenadas OpenGL y función gluLookAt().

Imagen obtenida de <http://www.glprogramming.com/red/chapter03.html>

En cada instante, la posición que devuelve el módulo WPS se le envía a *gluLookAt()* como el ojo, excepto el parámetro eyeY, el cual se asume como la altura media de los españoles (1,6 metros), ya que no se puede saber con precisión la altura del usuario.

Los tres siguientes parámetros, el vector hacia el que se dirige el ojo o cámara, es devuelto de la brújula del dispositivo móvil mediante la siguiente transformación:

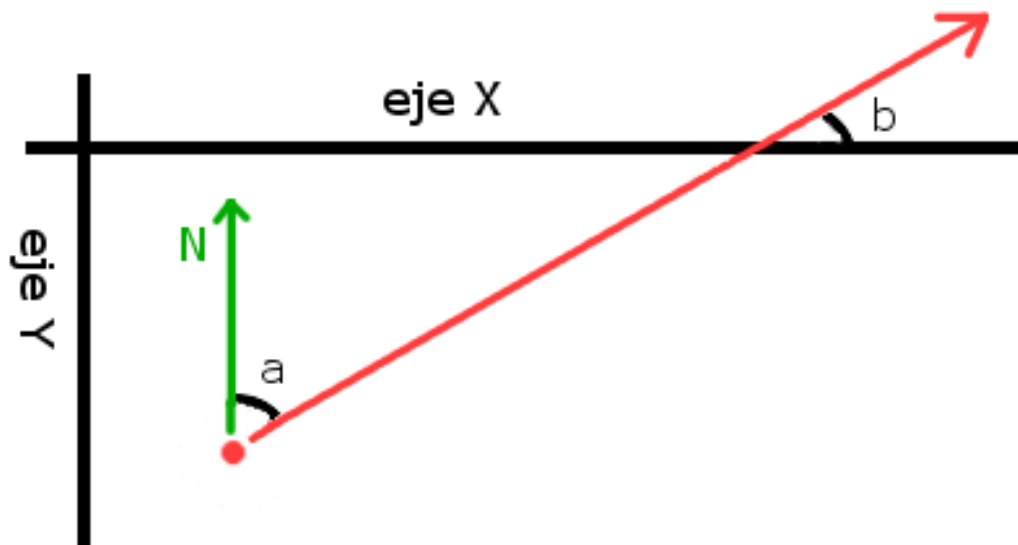


Figura 8: Representación aclaratoria para el lector del cálculo del vector de dirección.

En la Figura 8, el vector **N** representa el norte magnético, respecto al cual el terminal devuelve el ángulo **a**. Mediante una resta elemental podemos calcular el ángulo **b**, cuya tangente es la pendiente de la recta del vector de dirección. Utilizando una combinación de las ecuaciones de la recta y de la distancia entre dos puntos se calcula el punto hacia donde debe estar mirando la cámara. Suponemos que la máxima distancia de visión del usuario es de 50 metros. Los últimos tres parámetros se mantienen hacia como un vector vertical en el eje Y.

Una vez ubicado el terminal y transformada la vista, se proceden a transformar los elementos que se encuentren dentro de la visión del usuario. Esto es muy sencillo utilizando la función *glTranslatef(x,y,z)*, a la cual se realiza una llamada para cada fuego con las coordenadas de éste, pudiendo colocarlo en su posición exacta debido a que las coordenadas siempre las calculamos en metros. Además, para cada fuego dentro de la visión del usuario, utilizando los sensores de orientación se calcula un ángulo de giro del mismo para que se muestre en la posición correcta de la pantalla aunque esté girada respecto al suelo. La Figura 9 explica el comportamiento del sensor brújula en Android.



Figura 9: Funcionamiento de la brújula (Azimuth) en Android.
Imágen obtenida de <http://www.novoda.com>

4.1. Representación 2D del escenario

La representación 2D del escenario está realizada sobre el segundo piso de la Facultad de Informática de la UCM. Las medidas son tomadas y manejadas por la aplicación en metros, aunque a la hora de ser mostradas en el terminal se realiza la correspondiente escala a píxeles, siendo ésta de 10.904225532 en el mapa utilizado como representación.

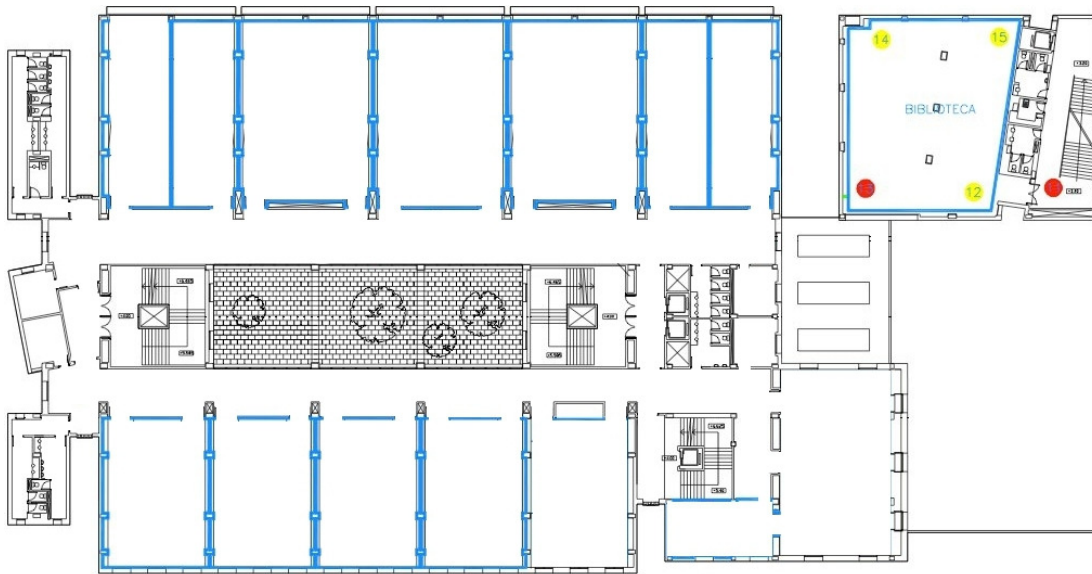


Figura 10: Imagen del mapa utilizado en la aplicación

4.2. Representación 3D del escenario

Para la representación tridimensional, usaremos la anteriormente explicada representación bidimensional en combinación con más factores:

- Orientación del terminal
- Distancia a los objetos virtuales (fuegos)
- Consideración de obstáculos en la línea de visión.

Usaremos un ejemplo. Supongamos que nos encontramos con el terminal en las coordenadas (101,63,1) (eje X = 101, eje Y = 63, primer piso). En el mapa en 2 dimensiones, la representación sería como en la Figura 11.

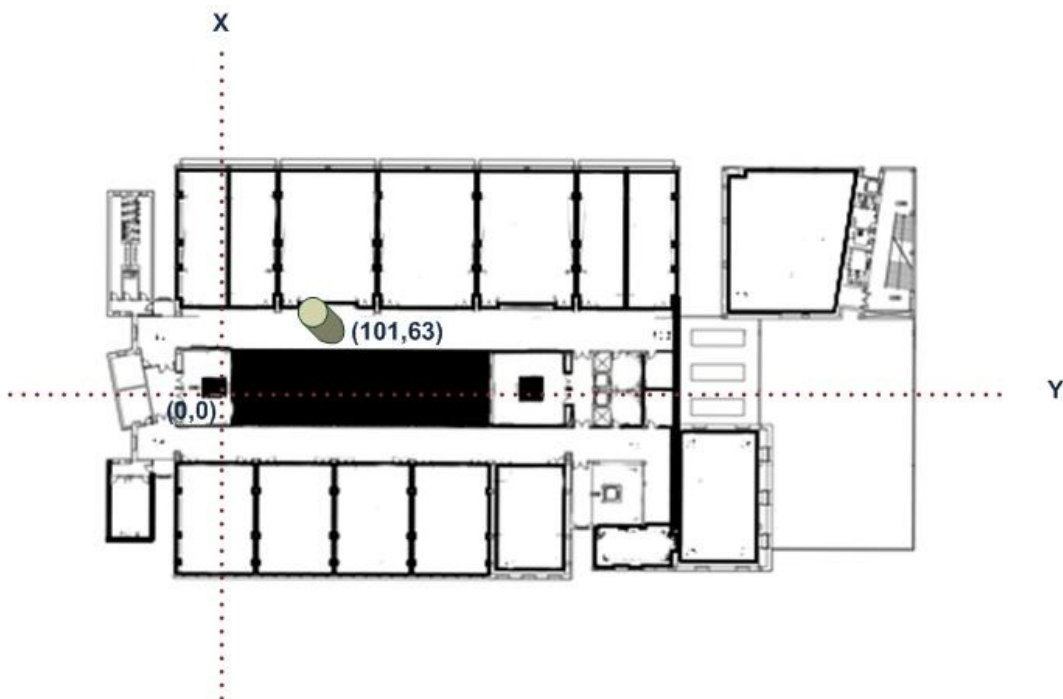


Figura 11: Representación del punto(101, 63)

El punto representado con un cilindro sería el terminal, que acorde a nuestro sistema de coordenadas se encontraría en el punto $(101,63)$, $(101,63,1)$ extendido. Supongamos ahora que se inicia un fuego (F1) en el punto correspondiente a $(304,64)$, $(304,64,1)$ extendido. Acorde a nuestra representación, este fuego se encontraría en el mismo pasillo del terminal, a una distancia en vertical de 200 pixeles sobre el eje X, como se puede ver en la Figura 12.

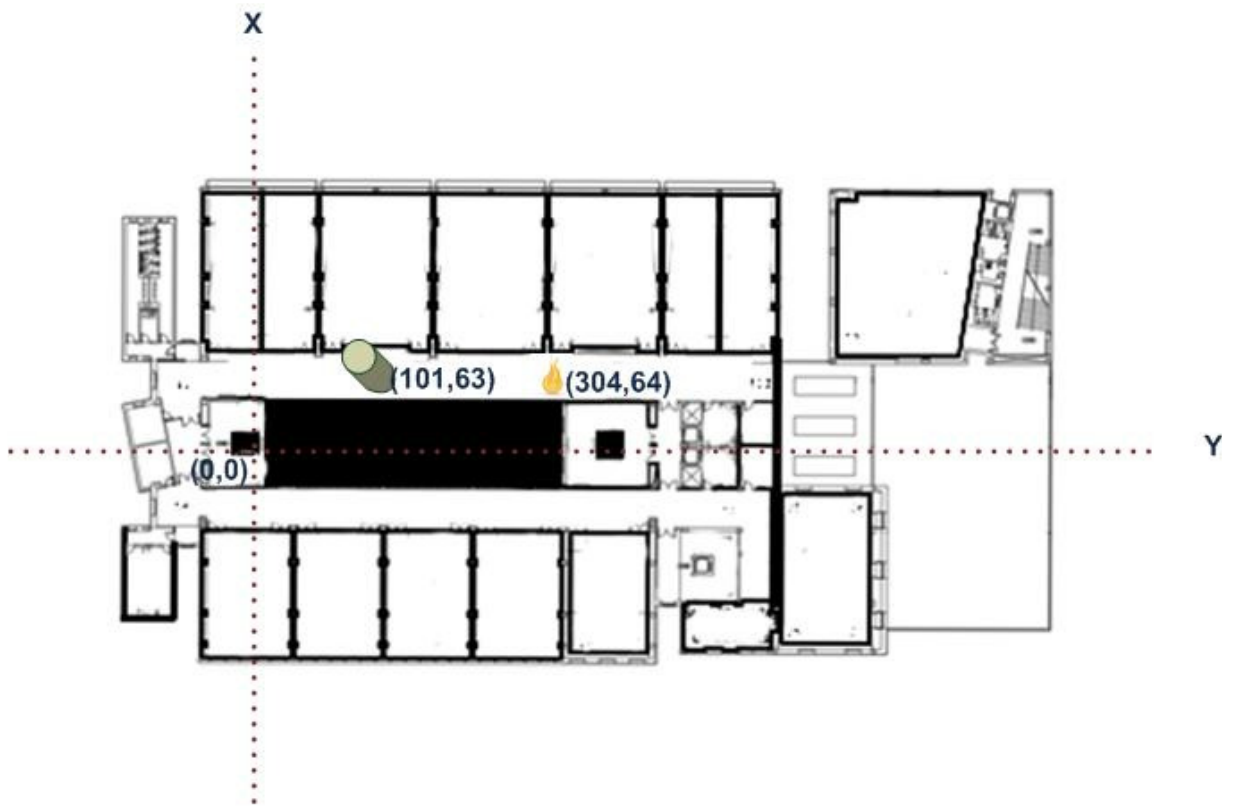


Figura 12: Inserción de un fuego en (304,64)

Se puede trazar una línea recta desde el terminal hasta F1 sin que se atravesase ninguna región en negro del mapa, de lo que se deduce que no hay ningún obstáculo que obstaculice la visión. Si desde el terminal se apuntase la cámara en la dirección del vector normalizado $(0, 1)$, es decir, apuntando al fondo del pasillo en dirección al fuego, la cámara mostraría lo siguiente:



Figura 13: Vista del fuego desde la perspectiva de Realidad Aumentada

Supongamos ahora el siguiente caso: un segundo fuego (F2) se inicia en las coordenadas (334,121), (334,121,1) extendido. Según podemos ver en la siguiente figura, el fuego se encontraría localizado dentro de una de las aulas, particularmente el aula 7. Desde la posición del terminal F2 no sería visible, ya que al trazar una línea que una ambas entidades, se corta un área negra en el mapa de bits de obstáculos (correspondiente a una pared). Por lo tanto, no sería realista pintar esta llama. La configuración del mapa sería la mostrada en la figura 14

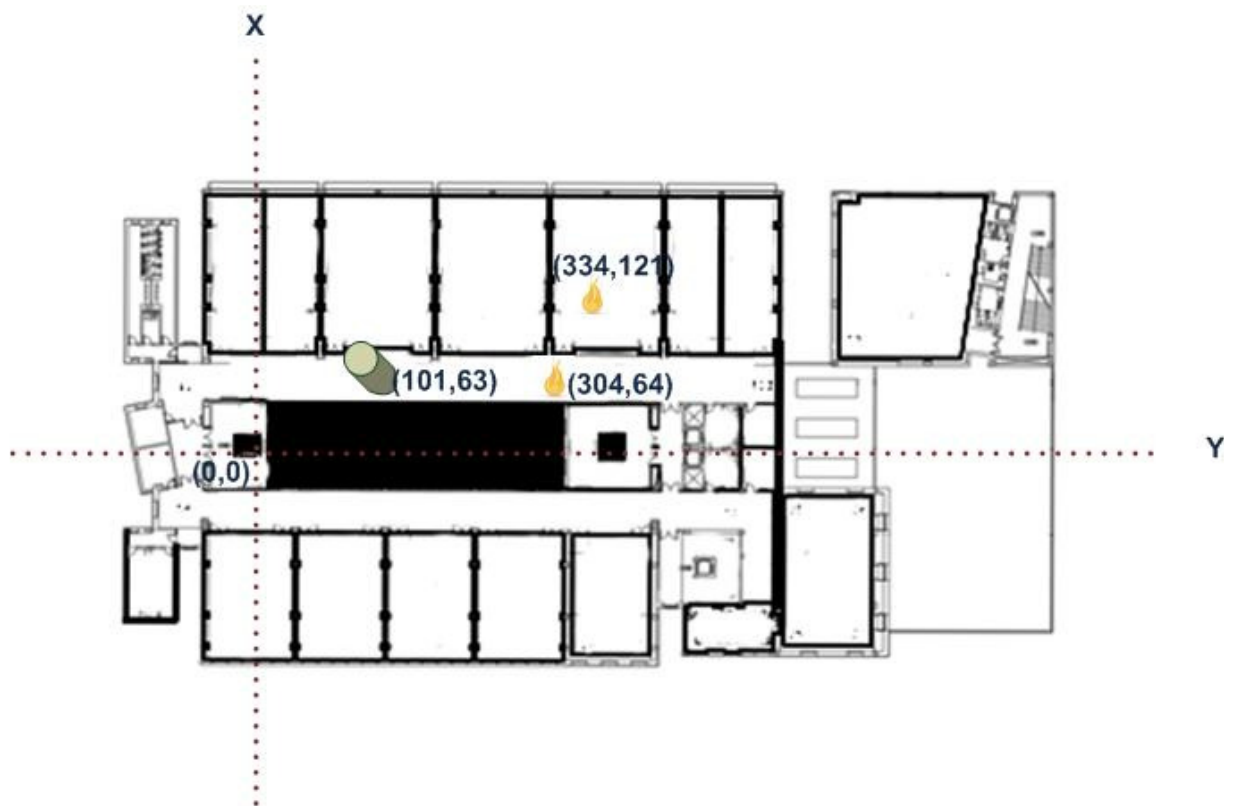


Figura 14: Fuego iniciado en (334, 121)

Si apuntásemos con el terminal a la zona donde se encuentra el fuego, veríamos lo siguiente:



Figura 15: Apuntando hacia la localización del segundo fuego

En definitiva, para la visualización 3D podemos aseverar la primera de las reglas: si al trazar una línea desde nuestra posición al foco de uno de los fuegos esta línea atraviesa en el mapa de obstáculos una región de píxeles negros, podemos afirmar que el foco se encuentra detrás de un obstáculo y por lo tanto no debería visualizarse.

Supongamos un nuevo problema. El terminal se acerca hasta la posición de F1. La lógica nos dice que, a menor distancia entre dos objetos, mayor es el tamaño con el que ese objeto se percibe. El vector que separa el terminal de F1 en la figura 1 es $(203,1)$.

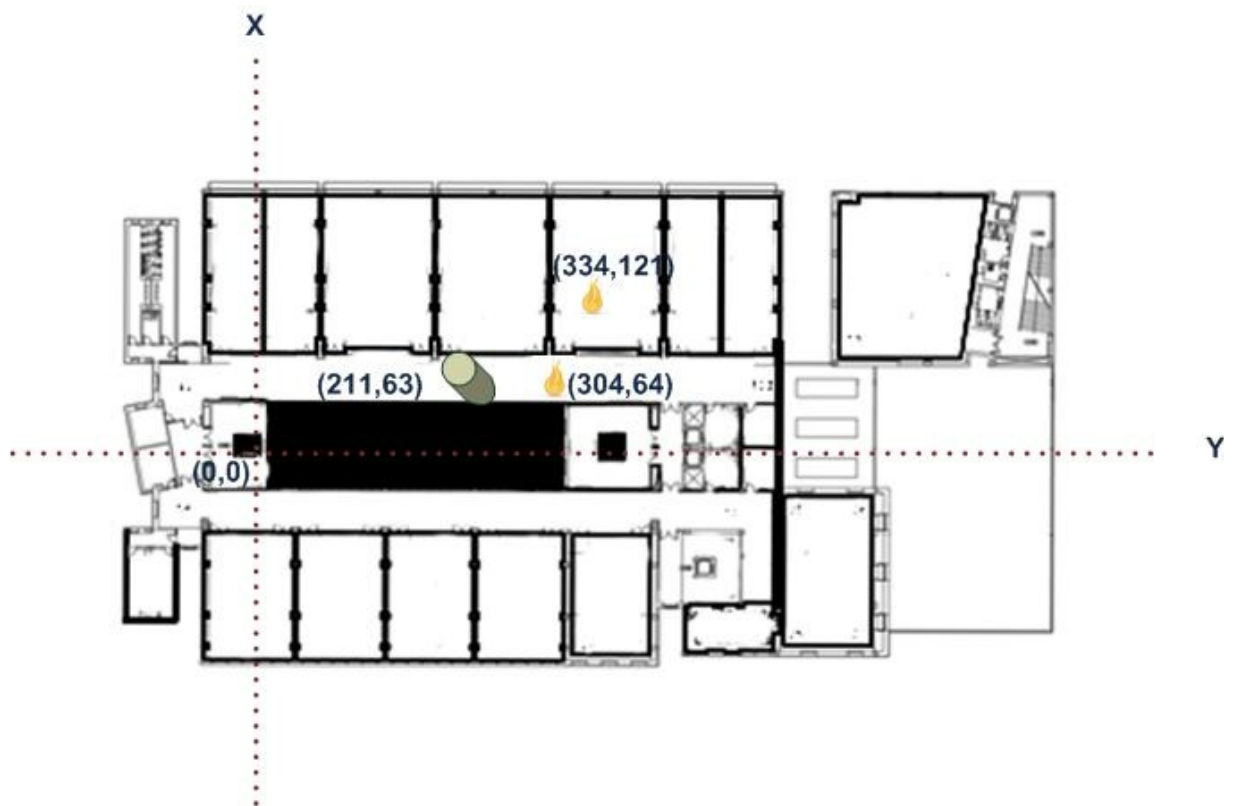


Figura 16: El usuario se desplaza hasta (211, 63)

En esta nueva configuración deberíamos ver F1 con un tamaño mayor que como lo vimos en la Figura 15, como se muestra a continuación:

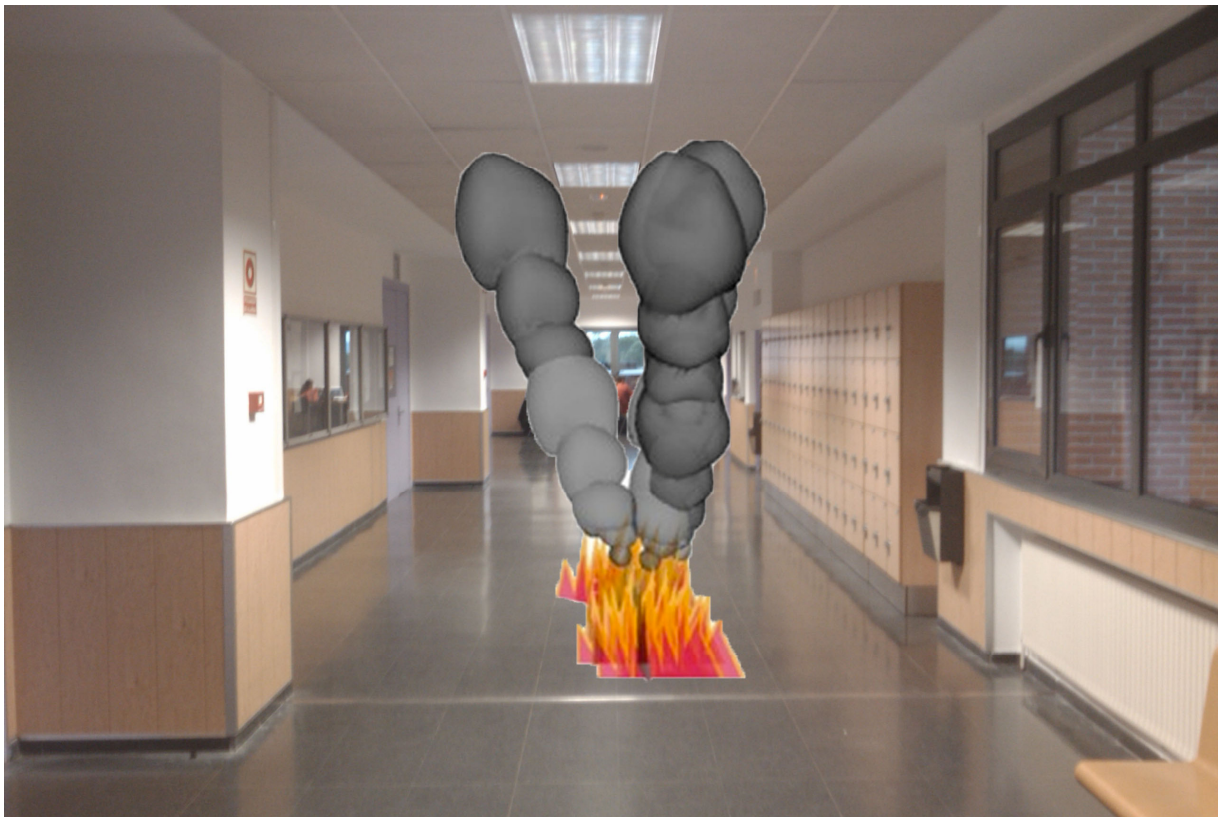


Figura 17: Visualización del fuego desde una posición más cercana

El primer dilema fue qué opción adoptar para solucionar la problemática de encontrarnos con un espacio en 3 dimensiones sobre el que aplicar transformaciones geométricas, y que tuviese resultados realistas. A priori la opción que sonaba más natural era utilizar OpenGL, que facilita en gran medida el uso de modelos tridimensionales y la aplicación de transformaciones (rotación, traslación, escalado...). El SDK de Android ofrece una versión Lite de OpenGL, con la que disponemos de estas funciones. Disponiendo de la topología del mundo real (el terminal y el escenario) y de la topología del mundo virtual (localización de los fuegos y relación con el terminal; i.e., distancia, orientación) podemos generar un escenario de realidad aumentada que muestre de manera realista los focos de incendio, y con los que se pueda interactuar de la misma manera que se haría con fuegos reales (evitando acercarnos a ellos, sortearlos, etc).

5. Análisis de redes y posicionamiento

Como se ha mencionado anteriormente, hemos usado los trabajos de [EvMa06] y [WLSB07] para implementar los sistemas de posicionamiento. Aunque todavía no se encuentran disponibles de manera muy extensa a nivel comercial, si se han usado en otros campos. El término WPS fue recogido por SkyHook Wireless, primera compañía en ofrecer una API gratuita para utilizar el posicionamiento WiFi. A pesar de que haber usado esta API habría simplificado enormemente nuestro trabajo, no fue posible debido a una restricción importante: SkyHook debe haber recogido previamente los datos de los puntos de acceso en sus bases de datos, algo que no había ocurrido con los puntos de acceso del interior de nuestra facultad. Hay otras consideraciones importantes, como las pruebas de campo que hicimos en zonas urbanas densamente pobladas (Madrid-Callao), y recogidas por SkyHook dentro de la base de datos. Como se puede apreciar en la figura 18, la precisión de SkyHook deja bastante que desear para nuestro propósito (hasta 10 metros de margen), por lo que decidimos implementar nuestro propio sistema para poder asegurar nuestros propósitos.

	XPS	GPS	A-GPS
Accuracy	10 meters	10 meters	30 meters
Availability	99.8%	80.0%	95%
Time-To-First-Fix	1 sec	65 sec	30 sec

Figura 18: Rendimiento de SkyHook. Obtenido de <http://www.skyhookwireless.com/howitworks/performance.php>

A continuación se describen formalmente los dos métodos que empleamos para la implementación del posicionamiento WiFi.

5.1. K-Closest Neighbors

Este algoritmo recupera las K posiciones de la base de datos que más concuerdan con la señal de los puntos de acceso que se pueden ver desde la posición del terminal. El criterio usado para seleccionar las mejores posiciones es la distancia euclídea (medida como la diferencia en la intensidad de la señal). Si $Z = (RSS_1, \dots, RSS_M)$ el vector de señales observadas desde el dispositivo, compuesto por M puntos de acceso en una posición desconocida $X = (x, y)$ (aquellas del terminal del cual deseamos obtenerla) y Z_i son las señales registradas en la base de datos para la posición $X_i = (x_i, y_i)$, entonces la distancia euclídea es

$$d(Z, Z_i) = \frac{1}{M} \cdot \sqrt{\sum_{j=1}^M (RSS_j(x, y) - RSS_j(x_i, y_i))^2},$$

, siendo:

1. $RSS_j(x_i, y_i)$: la fuerza de la señal recibida para el punto de acceso con la MAC j, localizado en x_i, y_i .
2. M: número de diferentes puntos de acceso que se encuentran guardados en la base de datos

El conjunto N_k de posiciones de la base de datos, que contiene el menor rango de error m en términos de RSS, se construye según el siguiente proceso iterativo:

$$N_k = \left\{ \underset{X_i \in \mathcal{L}}{\operatorname{argmin}} [d(Z, Z_i)] \mid X_i \notin N_k \right.$$

Esto recupera las K posiciones de la base de datos con el menor error, y sin ningún tipo de repetición (recupera un conjunto único). \mathcal{L} es el conjunto de posiciones registradas dentro de la base de datos. Finalmente, la posición del dispositivo se calculará como el baricentro de las k posiciones:

$$X = \frac{\sum_{j=1}^k (1/d(Z, Z_i)) \cdot X_j}{\sum_{j=1}^k (1/d(Z, Z_i))} \quad \text{with } X_j \in$$

La principal ventaja de este algoritmo es la sencillez de implementación y uso. No obstante, la precisión depende de la granularidad de la base de datos de referencia. Cuanto mayor se desee que sea la precisión, mayor debería ser la base de datos, lo que puede dar lugar a un tiempo de acceso mayor para localizar el dispositivo, y eventualmente conducir hacia una latencia inaceptable.

La selección del tamaño para el conjunto N puede variar según el número de puntos que estemos usando para calcular la posición del terminal, pudiendo modificarse para intentar alcanzar algún valor óptimo. No obstante, la experiencia nos ha demostrado que no hay un valor óptimo general, y que tan sólo las necesidades específicas de una aplicación puede determinar el valor más eficiente para cada caso.

5.2. Filtro de partículas

Hemos seguido para la implementación la descripción y metodologías explicadas en [EvMa06]. Acorde a ello, proporcionamos la definición de “filtro de partículas”. La idea principal de este algoritmo es combinar el modelo de movimiento y el mapa de información en el filtro para obtener un modelo de aproximación más realístico a la posición del dispositivo.

El filtro de partículas, basado en un conjunto de ejemplos medidos aleatorios (es decir, las partículas), representa la función de densidad de la posición del dispositivo. Cada partícula explora el entorno de acuerdo al modelo de movimiento y al mapa de información. Sus pesos son actualizados cada vez que se recibe una nueva medida. El filtro de partículas intenta estimar la distribución probabilística $\Pr [x_k | Z_{0:k}]$, donde x_k es el vector de estado del dispositivo en el instante k , y $Z_{0:k}$ es el conjunto de medidas recolectadas hasta la medida $(k+1)$. Cuando el número de partículas es lo suficientemente alto, puede ser equiparado a:

$$\Pr [x_k | Z_{0:k}] = \sum_{i=1}^{N_s} w_k^i \delta(x_k - x_k^i).$$

El filtro está compuesto de 5 pasos: predicción, corrección, actualización de la partícula, muestreo y estimación

5.2.1. Predicción

En esta etapa se predice, mediante el uso del acelerómetro, el rumbo que tomará el usuario una vez ha iniciado el desplazamiento.

Como se muestra en [RPTMS02], durante este paso las partículas se propagan a través del edificio, dada una ley de evolución que asigna una nueva posición para cada partícula con una aceleración medida desde los sensores de aceleración del móvil. La nueva partícula puede ser obtenida a través de la siguiente función de transición:

$$\begin{bmatrix} x^i(k+1) \\ y^i(k+1) \\ v_x^i(k+1) \\ v_y^i(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^i(k) \\ y^i(k) \\ v_x^i(k) \\ v_y^i(k) \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} a_x(k) \\ a_y(k) \end{bmatrix}$$

Donde $[x^i(k), y^i(k), v_x^i(k), v_y^i(k)]$ es la posición y la velocidad de una partícula en un instante temporal k , Δt es la diferencia de tiempo entre los dos momentos medidos y $[a_x(k), a_y(k)]$ es la aceleración del dispositivo en el momento k .

En este modelo, la aceleración se considera como un proceso iterativo para la simulación, pero se ha propuesto el uso de acelerómetros, así que usaremos esta última opción. En este punto incluimos el mapa de información, con el propósito de eliminar las partículas que tienen un movimiento imposible (como cruzando una pared, por ejemplo).

$$Pr[x_k | x_{k-1}] = \begin{cases} 0 & \text{if a particle crossed a wall,} \\ 1 & \text{if a particle did not cross a wall.} \end{cases}$$

5.2.2. Corrección

Esta etapa se encarga de corregir errores de cálculo que hayan podido tener lugar durante el cálculo de la posición mediante una estimación, para evitar errores con un desvío importante del valor real.

Es posible estimar $\Pr [z_k | x_k]$ con esta ecuación:

$$\Pr [Z_k | x_k^i] = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{\|X_{z_k} - X_{x_k^i}\|^2}{2 \cdot \sigma^2} \right]$$

Donde X_{z_k} es la posición del dispositivo móvil devuelta por el algoritmo k-closest neighbors $x_{x_k^i}$ la posición de la i-ésima partícula, y σ el intervalo de confianza. Cuanto más pequeño es σ , más precisa es la medida

5.2.3. Actualización de la partícula

Durante esta etapa se actualiza el peso de la partícula, según la fórmula abajo descrita:

$$w_k^i = w_{k-1}^i \cdot \Pr [x_k | x_{k-1}] \cdot \Pr [z_k | x_k].$$

Después de unas pocas iteraciones, los pesos de las partículas pueden llegar a ser muy bajos, porque muchos de ellos habrán cruzado por una pared. Para prevenir este fenómeno, necesitamos una etapa de re-muestreo

5.2.4. Re-muestreo

El remuestreo analiza de nuevo las partículas existentes. El criterio para disparar la etapa de re-muestreo (resampling, en inglés) viene dado por la siguiente fórmula:

$$\frac{1}{\sum_{i=0}^{N_s} (w_k^i)^2} \leq \text{Threshold.}$$

Si el paso de re-muestreo es necesario, el algoritmo copia la mitad de las partículas, y descarta la otra mitad que contiene las partículas con los pesos más bajos. Ahora necesitamos combinar las probabilidades calculadas para encontrar la nueva distribución:

5.2.5. Estimación

En este paso, el algoritmo usa la siguiente ecuación para estimar la nueva posición del dispositivo:

$$E[X|Z] = \sum_{x_i \in L} x_i * Pr[x_i|Z]$$

Donde L es el conjunto de posiciones almacenadas en la base de datos, y Pr [X_i|Z] es la distribución probabilística que puede ser calculada usando la ecuación que previamente introdujimos.

6. Cálculo de rutas

El principal objetivo de AVANTI es servir como simulador que pueda ayudar en el aprendizaje de protocolos de evacuación de incendios. Para ello, es necesario poder realizar un cálculo de rutas hacia las posiciones de salida, teniendo en cuenta las restricciones en caso de incendio:

- No se debe utilizar el ascensor.
- Por el peligro de bolsas de gas, no se puede abrir puertas que haya cerradas
- Se debe evitar subir de nivel (de piso) salvo que sea estrictamente necesario, procurando llegar siempre a la parte inferior del edificio (y consecuentemente a la salida)
- En caso de que todas las salidas se encuentren bloqueadas, se deberá intentar acudir a la zona al aire libre (terraza o ventana) más cercana.

Inicialmente se contemplaron dos opciones para efectuar el cálculo de rutas:

- División de la Facultad en áreas correspondientes a nodos de escape. En cada nodo se podría implantar un fuego. Algunos nodos serían salida. La heurística de los nodos sería la distancia hacia la salida más corta. La principal ventaja de este modelo es la sencillez de implementación y de cálculo de las rutas. Por otro lado, esa misma sencillez puede desembocar en falta de precisión.
- Generación de las rutas de escape a partir de los mapas de bits de obstáculos. Los fuegos pueden ser situados en cualquier coordenada continua de la Facultad. Este modelo proporciona una muy mayor precisión, pero puede que su implementación sea compleja para el ámbito de AVANTI.

Finalmente utilizamos la primera alternativa. Tras realizar pruebas, comprobamos que aplicar algoritmos para cálculo de rutas sobre un mapa de bits no arrojaba los resultados con la fluidez que la aplicación necesitaba, y descartamos la segunda alternativa.

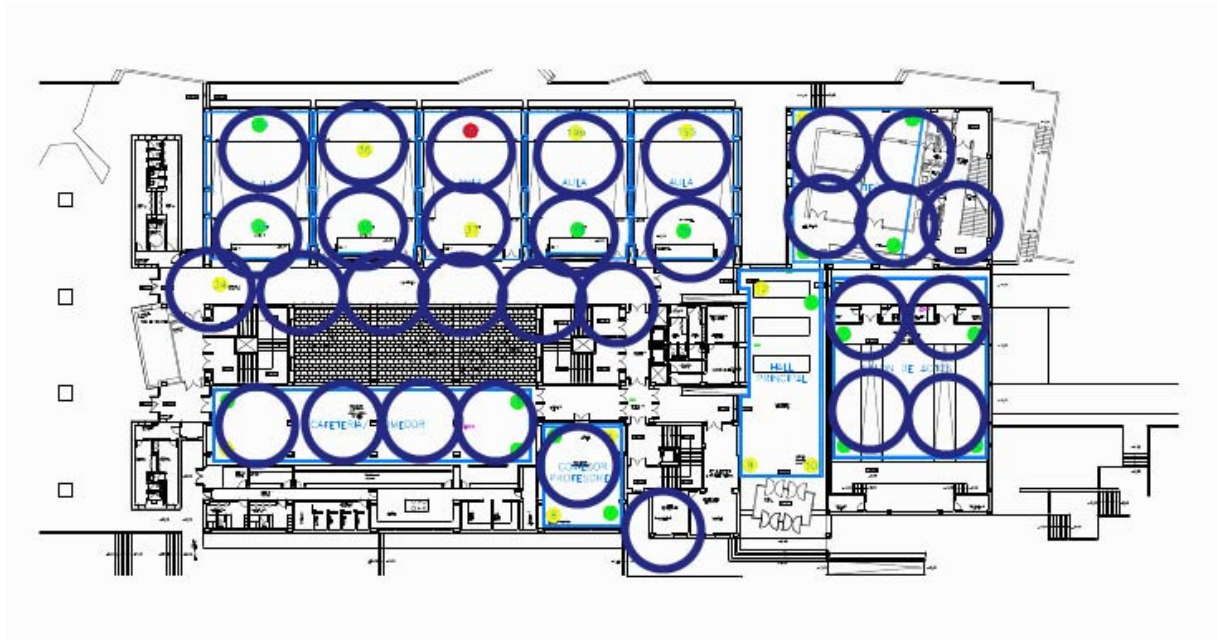


Figura 19: División de la facultad en sectores

Una vez establecido el sistema de representación, decidimos aplicar el algoritmo de caminos mínimos (algoritmo de Dijkstra) para buscar la ruta más corta hacia una salida, es decir, la ruta de menor coste hacia una de las salidas disponibles. Para aplicar este algoritmo, optamos por representar la primera planta en forma de grafo: para ello, hemos realizado una división en cuadrantes de los pasillos de la primera planta. También hemos planteado la división en zonas para la Sala de Juntas y la Sala de Grados, también situadas en la primera planta del edificio.

Internamente, cada cuadrante está definido por las posiciones superior izquierda e inferior derecha del mismo y un identificador único para cada uno de ellos, así como un indicador de si se trata de un nodo de salida o no. El conjunto de cuadrantes resultante de las divisiones de la primera planta forma el conjunto de nodos de nuestro grafo.

Para almacenar las aristas de dicho grafo, usamos una matriz de adyacencia para almacenar las conexiones entre los distintos cuadrantes: hemos acordado utilizar un coste unitario para los cuadrantes que están conectados; en caso contrario, usamos un valor muy elevado. También usamos una lista de aristas, cada una de ellas definida por su origen, destino y dirección. Cada arista tiene ocho posibles direcciones: arriba, arriba-derecha, derecha, abajo-derecha, abajo, abajo-izquierda, izquierda y arriba-izquierda. El uso de direcciones para las aristas facilita la representación gráfica de las rutas de escape en la vista de mapa de la aplicación.

El conjunto de fuegos que pueden colocarse sobre el mapa está almacenado en una tabla de nuestra base de datos. Mediante un script PHP, obtenemos todos los fuegos guardados en la base de datos y los devolvemos en un JSONArray (JavaScript Object Notation Array). Posteriormente, leemos los elementos del array y los vamos pintando sobre el mapa.

6.1. Algoritmo de Dijkstra

Dado un grafo dirigido y con aristas valoradas, el algoritmo de caminos mínimos o algoritmo de Dijkstra es un algoritmo voraz que calcula la distancia más corta desde un nodo del grafo hasta el resto de nodos de dicho grafo. El algoritmo recibe como entradas un grafo y el nodo origen a partir del cual se calcularán las distancias al resto de nodos del grafo, y devuelve como resultado un vector con las distancias mínimas a cada uno de los nodos de dicho grafo.

Las aplicaciones de este algoritmo son muy diversas: se emplea en el encaminamiento de paquetes por routers o para establecer rutas de tráfico aéreo y aviones, así como en reconocedores de lenguaje hablado para palabras que suenan de una forma similar.

Una implementación en pseudocódigo de este algoritmo puede ser la siguiente:

```
fun Dijkstra(Grafo G, nodo s)
//s es el nodo origen
var distancias: vector[1..n] de int;
    usados: vector[1..n] de boolean;
    n: nodo;
```

```
//inicializamos el vector distancias.
para cada n ∈ G hacer
    si (!hayArista(s,n)) entonces
        distancias[w] = +∞;
    en caso contrario
        distancias[n] = peso(s,n);
    fsi
fpara

//operaciones sobre el nodo origen.
distancias[s] = 0;
usados[s] = true;

//recorrido sobre el resto de nodos del grafo.
mientras (!usadosTodos(usados)) hacer
    v = buscarMinimo(distancias,usados);
    usados[v] = true;
    para cada n ∈ sucesores(G,v) hacer
        si distancias[n] > distancias[v] + peso(v,n)
entonces
            distancias[n] = distancias[v] + peso(v,n)
        fsi
    fpara
fmientras
ffun
```

Existen otros algoritmos de búsqueda de caminos mínimos como el algoritmo de Floyd; sin embargo, éste tiene un coste en tiempo de $O(n^3)$ frente al coste cuadrático del algoritmo de Dijkstra. Por lo tanto, la implementación escogida es muy acertada y no tiene un coste temporal muy elevado.

El cálculo de posibles rutas de salida se realiza en la vista de mapa de la aplicación. Ahí es donde se inicializa la lista de cuadrantes: para ello, se han medido sobre el plano de la Facultad la longitud y anchura de cada uno de los pasillos, la Sala de Juntas y la Sala de Grados. De esta forma, hemos determinado sobre dicho plano la posición y dimensiones de cada uno de los cuadrantes. Una vez establecida la lista de cuadrantes, los hemos conectado mediante aristas y hemos completado adecuadamente la matriz de adyacencia.

Las coordenadas de cada uno de los cuadrantes se han tomado en base al origen establecido. En la siguiente imagen, se puede apreciar la ubicación del origen de coordenadas tomado para el posicionamiento de usuarios en la primera planta del edificio.

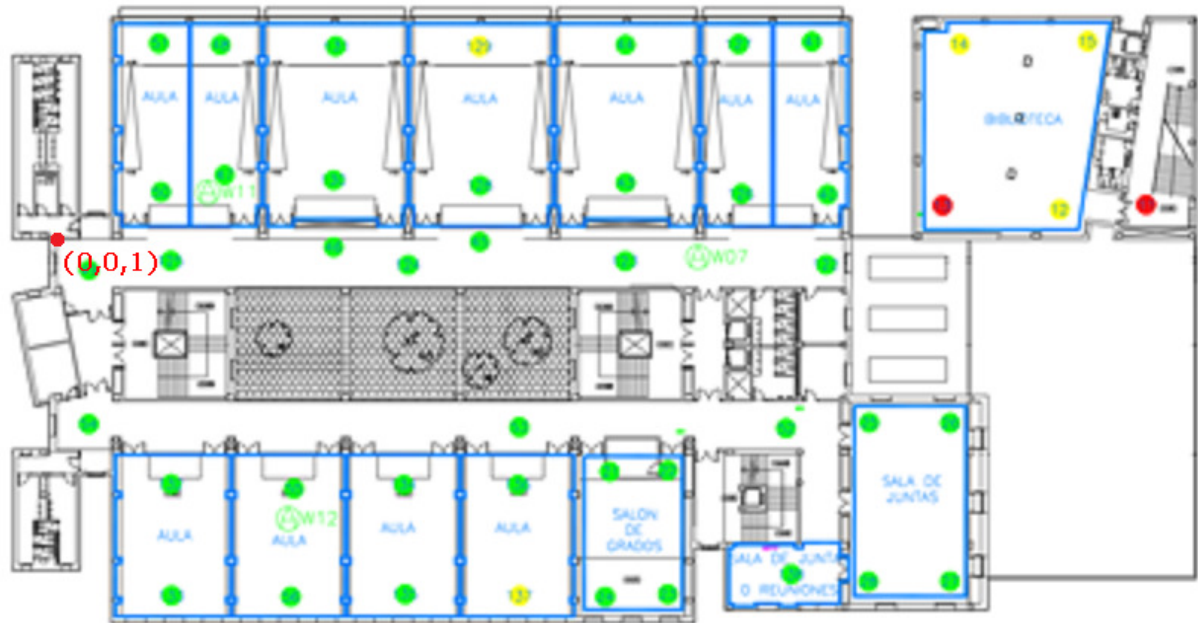


Figura 20: Localización del origen de coordenadas

Con el grafo generado, aplicamos el algoritmo de Dijkstra tomando como origen la posición actual en la que nos encontramos, hallada mediante el algoritmo de posicionamiento implementado, y obtenemos el vector de distancias mínimas desde el cuadrante origen al resto de cuadrantes. A partir del vector de distancias, elegimos el nodo de salida más próximo a nuestra posición.

La ruta de escape hacia la salida más cercana que no esté bloqueada se mostrará por pantalla mediante flechas, partiendo de la posición en la que se encuentra el usuario (calculada mediante el algoritmo de posicionamiento de los k-vecinos más cercanos).

En las pruebas realizadas, hemos dibujado sobre la vista de mapa todos los cuadrantes generados a partir de la división de los pasillos, la Sala de Grados y la Sala

de Juntas. En cada uno de los cuadrantes, se muestra su identificador y el área ocupada por cada uno de ellos. Sin embargo, en la versión final de la aplicación no representaremos gráficamente la división en cuadrantes. Se indicaría únicamente sobre la vista de mapa la ruta de salida y la posición actual, así como la ubicación de otros usuarios y los posibles fuegos que hubiera repartidos. En la parte inferior de la pantalla del terminal, se mostrarán las coordenadas de la ubicación del usuario respecto al origen tomado, y el cuadrante en el que se encuentra dicho usuario.

7. Descripción de componentes

En el siguiente apartado se describe mediante diagramas UMLs [SrPo07] los distintos componentes de la aplicación.

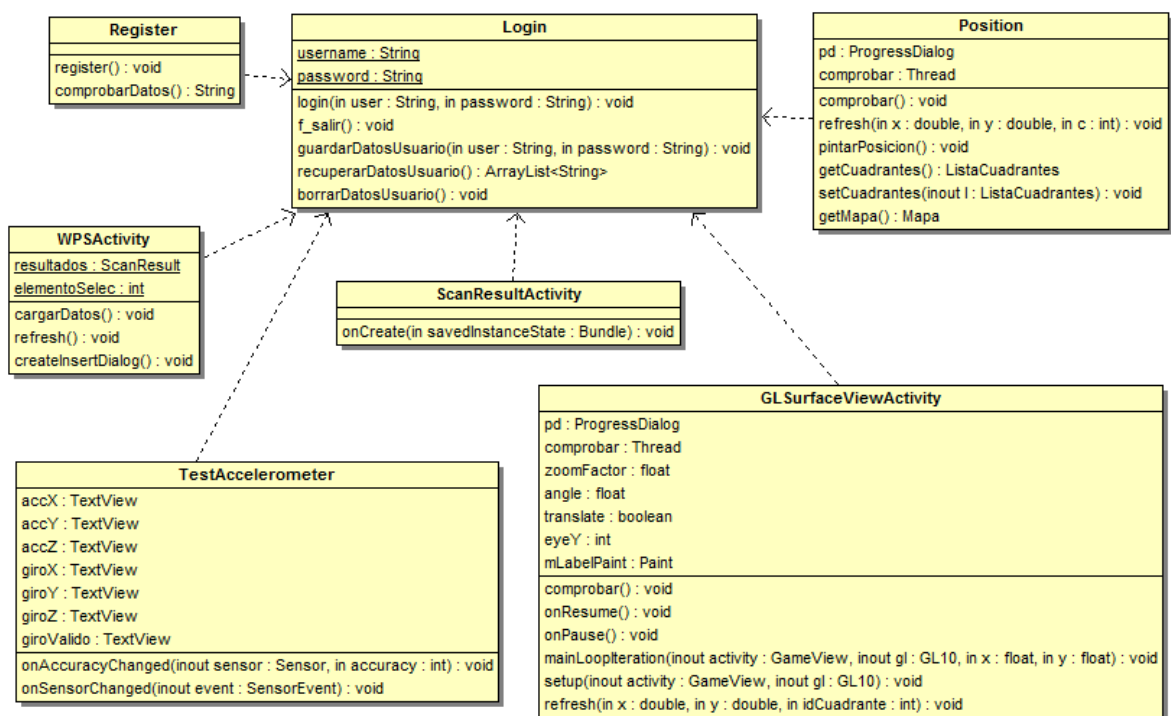


Figura 21: Diagrama de com.avanti.activities

Engloba el conjunto de clases de la aplicación que heredan de Activity.

- La clase Login se refiere a la pantalla de acceso a la aplicación. La función *login* realiza el acceso a la aplicación, comprobando si hay algún usuario en la base de datos con la contraseña introducida. Las funciones *guardarDatosUsuario* y *recuperarDatosUsuario* permiten almacenar los datos del último usuario que ha accedido a la aplicación, evitando tener que escribirlos la próxima vez que acceda a la aplicación. Por otra parte, la función *borrarDatosUsuario* elimina los datos almacenados del último usuario que accedió a AVANTI.
- La clase Register prepara la pantalla de registro de la aplicación, donde se encuentran los distintos campos a rellenar.
- TestAccelerometer engloba la prueba del acelerómetro.
- PositionTracker puede actualizar los datos de posición del usuario (coordenadas X e Y, y cuadrante en el que se encuentra) mediante la función *refresh*.
- WPSActivity comprende todo lo relativo a la actividad Wi-Fi: el método *refresh* actualiza las redes detectadas en el escaneo y el método *CreateInsertDialog* prepara una pantalla para insertar una nueva posición en la base de datos.
- ScanResultActivity prepara los datos de la red seleccionada en la pantalla de Actividad Wi-Fi, mostrándolos en una pantalla nueva.
- GLSurfaceViewActivity demuestra el uso de la vista GLSurfaceView, una vista que dibuja objetos en OpenGL sobre una superficie dedicada. Entre sus atributos está *zoomFactor*, que indica el factor de zoom aplicado al objeto 3D, y el booleano *translate*, que indica si hay que aplicar una traslación al objeto. Uno de los principales métodos de esta clase es *mainLoopIteration*, que indica las instrucciones que se deben aplicar a cada frame del objeto 3D, además de *setup*, que prepara el modelo a dibujar.

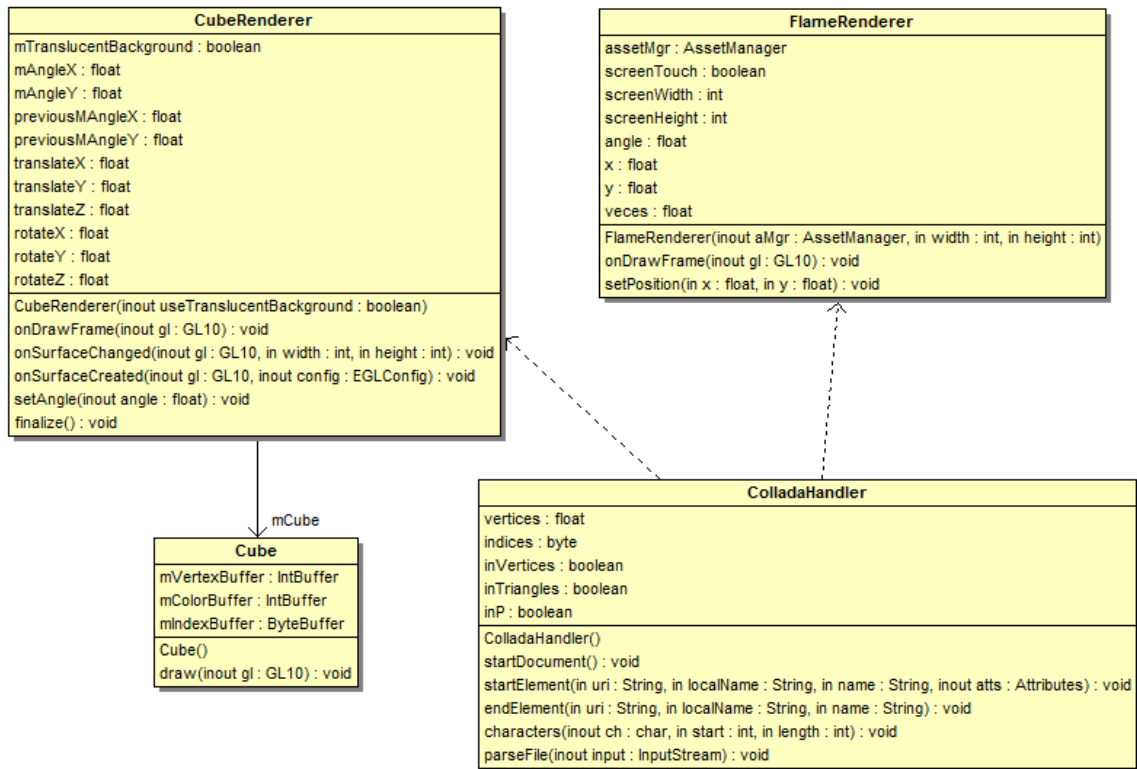


Figura 22: Diagrama de com.avanti.graphics

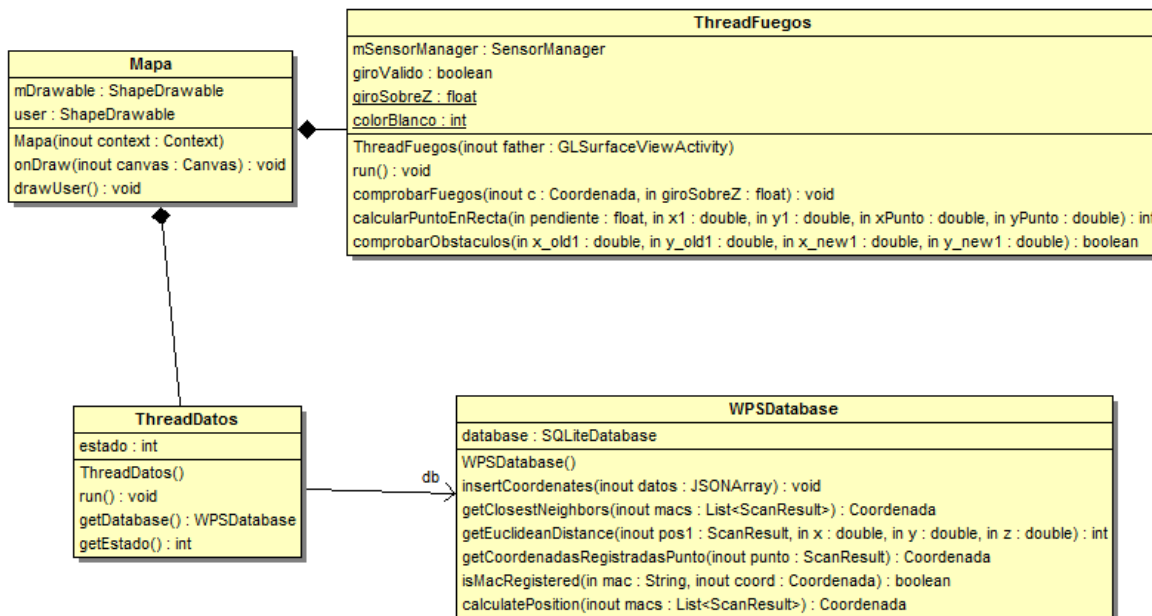


Figura 23: Diagrama de com.avanti.wps

Una de las clases más importantes de este paquete es WPSDatabase. El método *getClosestNeighbors* devuelve las coordenadas más próximas con la distancia al punto de referencia ya calculada. Por otra parte, la función *getEuclideanDistance* calcula la distancia euclídea entre el resultado del escaneo, en el primer parámetro, y la posición marcada por el resto de parámetros de la función. Finalmente, destacamos el método *calculatePosition*, que calcula una estimación de la posición exacta del terminal: para ello, se utiliza el método *getClosestNeighbors* para obtener los vecinos más próximos con sus respectivas distancias euclídeas.

Otra de las clases más importantes es ThreadUbicación. Esta clase hace uso de un hilo de ejecución para calcular la posición del terminal con el algoritmo indicado, guardando los valores para que la aplicación principal los recupere en cada iteración.

Cabe destacar que la clase ThreadFuegos es la encargada de gestionar un hilo de ejecución que recupere la posición de los fuegos del servidor. La clase ThreadDatos se utiliza para cargar al inicio de la aplicación la Base de datos del servidor en el terminal móvil, utilizando la implementación de Android de SQLite.

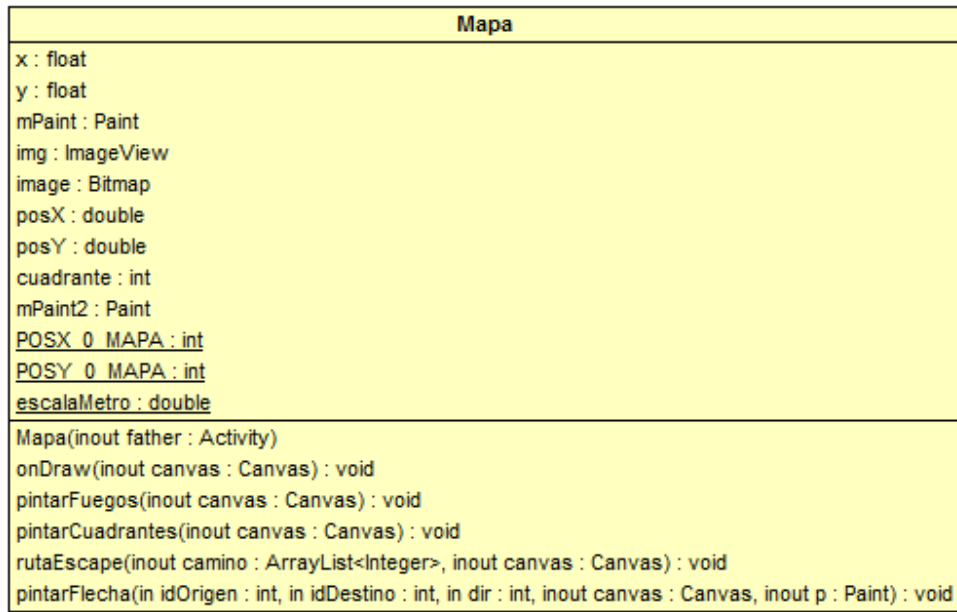


Figura 24: Diagrama de com.avanti.position

Este paquete únicamente engloba la clase Mapa. Dicha clase contiene los métodos para cargar el mapa del primer piso de la Facultad de Informática, así como para dibujar la división en cuadrantes del mapa con el método *pintarCuadrantes*. También, tenemos métodos para señalar la posición del usuario o de los fuegos cargados, además del método *rutaEscape* para representar gráficamente la ruta hacia la salida más cercana. Esta clase tiene atributos que almacenan la posición del usuario, la superficie sobre la cual se cargan las imágenes, la posición inicial que se toma para cargar la imagen del mapa y la escala para indicar cuántos píxeles equivalen a un metro en el mapa.

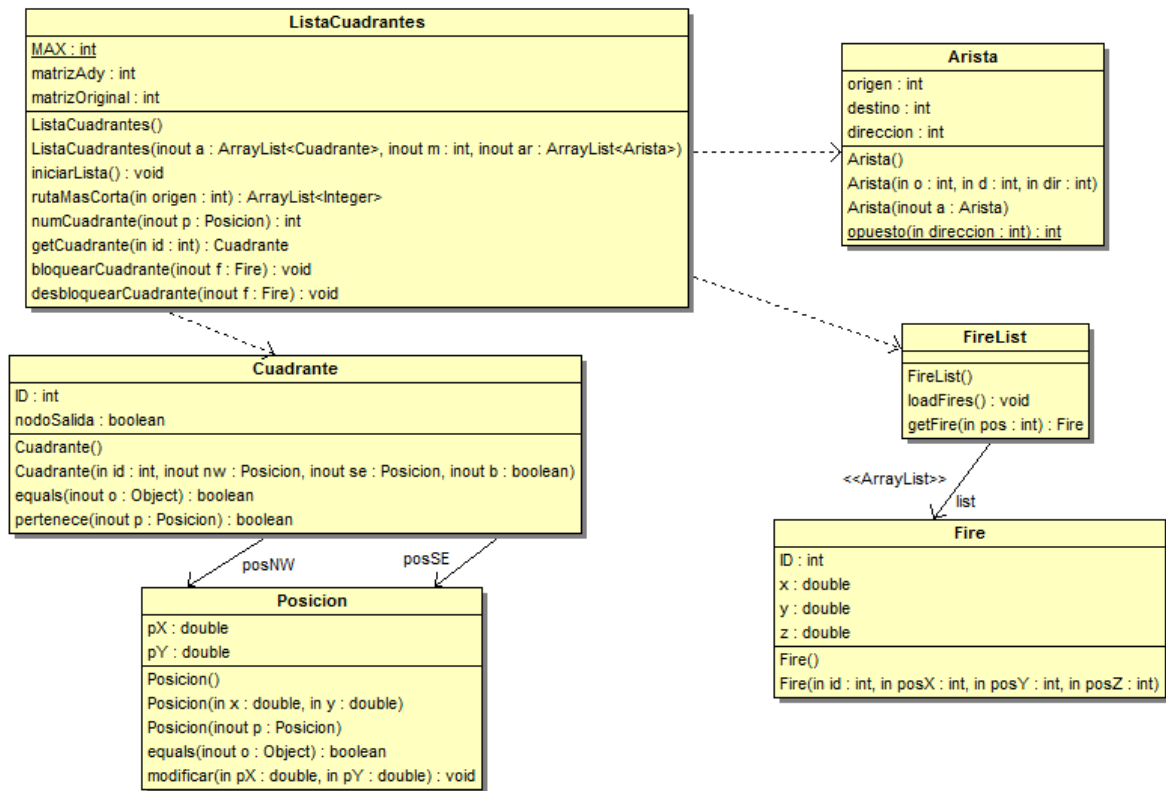


Figura 25: Diagrama de com.avanti.routes

Este paquete contiene todas las clases necesarias para el cálculo de posibles rutas de salida en caso de que haya fuegos en determinados puntos de los pasillos.

- Los objetos de la clase Cuadrante vienen identificados por las coordenadas de su esquina superior izquierda y de su esquina inferior derecha, así como un identificador único para cada uno de ellos. El atributo *nodoSalida* indica si el cuadrante está en una salida o no.
- La clase ListaCuadrantes contiene la representación del conjunto de cuadrantes en que se han dividido los pasillos de la primera planta de la Facultad de Informática. Para ello, se ha representado el conjunto de cuadrantes como un grafo mediante una matriz de adyacencia, en el atributo *matrizAdy*. En el método *iniciarLista*, se crean el conjunto de cuadrantes de los pasillos, la Sala de Grados y la Sala de Juntas, y las conexiones entre los mismos. Otro de los

métodos fundamentales de esta clase es *rutaMasCorta*, que calcula el camino que hay que seguir para alcanzar la salida más cercana.

- La clase Arista almacena el identificador del cuadrante de origen y de destino, así como la dirección de la misma. La dirección es un atributo auxiliar que hemos usado para dibujar las flechas de forma precisa sobre el mapa.
- La clase Posicion es una clase auxiliar para representar las coordenadas en los ejes X e Y.
- La clase Fire utiliza atributos similares a los de la clase Cuadrante: un identificador y sus coordenadas X, Y, Z.
- La clase FireList guarda una lista de los fuegos que hay almacenados en la base de datos. El método *loadFires* se encarga de acceder a la base de datos y guardar la lista de fuegos en un JSONArray mediante un script PHP, para finalmente almacenarlos en un ArrayList.

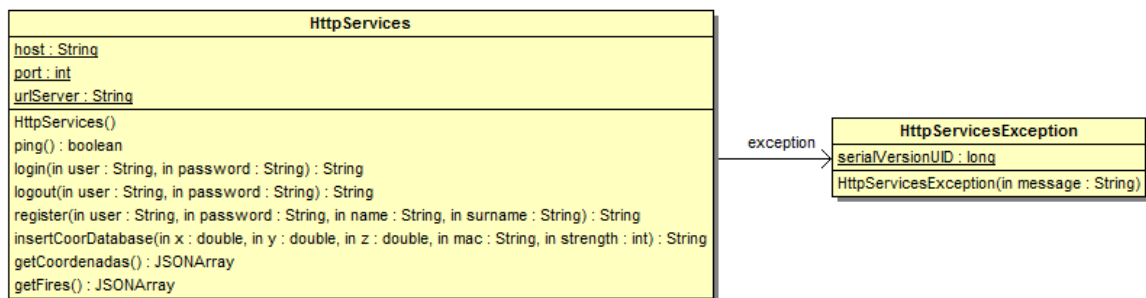


Figura 26: Diagrama de com.avanti.http

Este paquete contiene principalmente la clase `HttpServices`. Esta clase se utiliza para gestionar todas las conexiones que se realizan entre el terminal del usuario y el servidor.

8. Modelo de la base de datos

El siguiente diagrama representa un diagrama Entidad-Relación aplicado sobre la base de datos. Se ha utilizado el programa DIA¹³ para poder generar el diagrama de manera automática. El software se conecta a la base de datos, y tras seleccionar uno de los esquemas genera los diagramas correspondientes

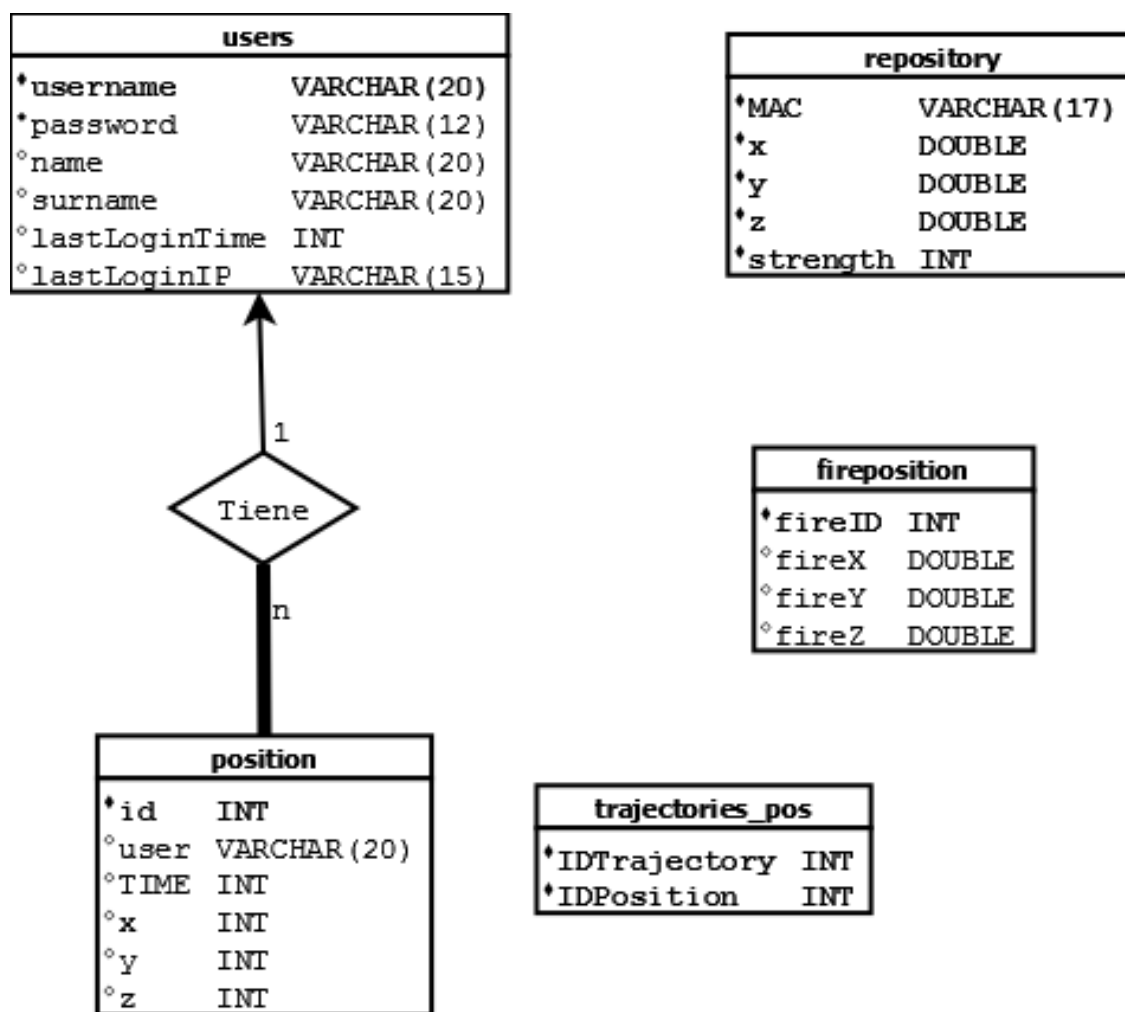


Figura 27: Modelo de la BBDD

¹³ <http://dia-installer.de>

- La tabla *users* almacena la información de los usuarios registrados en la aplicación.
- La tabla *repository* almacena las mediciones realizadas en diferentes posiciones del mapa con la información necesaria para procesarla, es decir, su posición, la MAC del punto de acceso y la potencia de la señal en dBm. Esta tabla se replica en el terminal al comienzo de la aplicación para acelerar el proceso de ubicación.
- La tabla *position* almacena la posición de un usuario en un instante de tiempo dado. Según se va actualizando la posición del usuario, el terminal actualiza el registro correspondiente a ese usuario en esta tabla.
- La tabla *trajectories_pos* almacena la trayectoria en la que se engloba el movimiento de un usuario, para poder utilizar esta información para evaluar las rutas seguidas por el usuario.
- La tabla *fireposition* almacena las posiciones de los distintos focos de incendio que existen en la simulación.

Capítulo IV

Gestión del proyecto

1. Planificación del proyecto

La planificación del proyecto se ha dividido en 5 iteraciones. En una primera iteración preliminar tratamos de determinar qué tecnologías servirían mejor a nuestros propósitos, e intentaríamos montar un prototipo para poder evaluar la velocidad con la que podríamos avanzar en el proyecto. Después de esta etapa inicial, consideramos una iteración para desarrollar el posicionamiento WiFi, otra iteración para desarrollar el servidor y otra para desarrollar la realidad aumentada. Estas iteraciones se solaparon por momentos, ya que algunas tareas eran específicas de cada módulo, y su desarrollo no afectaba al resto. Una última iteración se fue necesaria para realizar pruebas sobre la aplicación, comprobar que funcionaba sobre el terreno, mejoras sobre la eficiencia y planear futuras extensiones. Dentro de cada iteración se incluyen hitos de 2 a 4 semanas de duración, considerando esta como la duración idónea para que el manejo de las pequeñas tareas fuese flexible y fácil de adaptar en caso de problemas o incumplimiento de deadlines. Los diagramas de Gantt [Berg58] del capítulo V muestran de forma gráfica nuestra planificación y progreso.

2. Gestión de riesgos

Antes del comienzo del proyecto un análisis de riesgos se debe llevar a cabo, para poder detectar riesgos y adoptar una estrategia [Char89]. Para el plan de gestión de riesgos decidimos adoptar una estrategia pro-activa, intentando detectar los riesgos antes de caer en ellos, y proponiendo una solución que permita evitar el riesgo (así como una alternativa en caso de que el riesgo planteado sea finalmente el escenario).

Para los riesgos que no hayamos podido prever adoptaremos una estrategia reactiva (conocida como la “Escuela Indiana Jones de Gestión del Riesgo” [Thom92]). Intentaremos identificar el problema, así como sus causas. Si se da este caso, diseñaremos un plan para poder abordar el riesgo si vuelve a surgir, o algún nuevo riesgo que hayamos determinado.

2.1. Primer paso: identificación de riesgos

En esta primera etapa establecemos mediante algunas sesiones de “brainstorming” los riesgos que consideramos más relevantes para nuestro proyecto:

Definición del proyecto:

[DP-1]: Mala estimación del deadline

[DP -2]: Fechas límites para hitos incumplidas

Riesgos tecnológicos:

[RT-1]: Problema de compatibilidad con los navegadores

[RT -2]: Problemas de velocidad asociados a la realidad aumentada

[RT -3]: Problemas de velocidad asociados al posicionamiento WiFi

[RT -4]: Problemas de rendimiento con el terminal móvil

[RT -5]: Funcionalidades no implementadas en la API de OpenGL para Android

Riesgos de experiencia y habilidades personales:

[REH-1]: Falta de conocimiento en las tecnologías utilizadas

[REH -2]: Problemas de comunicación entre las capas

Riesgos personales

[RP-1]: Dedicación al proyecto en periodos como vacaciones y exámenes

2.2. Segundo paso: Análisis / Planificación de riesgos

Después de identificar los posibles, riesgos, pasamos a su análisis y planificación. Medimos la probabilidad que tiene el riesgo de ocurrir y sus posibles consecuencias. Se utiliza para medir la probabilidad de que el riesgo ocurra las siguientes categorías: baja, muy baja, moderada, alta y muy alta. Llevamos a cabo el mismo proceso con la dimensión de las posibles consecuencias, y utilizamos las siguientes categorías: catastrófica, seria, tolerable o insignificante.

Algunos riesgos pueden ser prevista si estos periodos se incluyen en la planificación del proyecto. Otros, como los relativos a la falta de formación en una

nueva tecnología, pueden ser minimizados formando a los miembros del equipo de desarrollo en estas tecnologías en fases previas al proyecto. No obstante, hay algunos riesgos que podían ser catastróficos para completar el proyecto en tiempo. En estos casos, se consideran algunas alternativas que permitan sortear el riesgo.

[DP-1] Mala estimación del deadline

Probabilidad: Moderada

Impacto en el proyecto: Serio

Solución: Para aquellas fases en las que los tiempos planeados no están claros o hay dudas sobre la viabilidad de conseguir los objetivos a tiempo, consideraremos un tiempo extra para poder prevenir retrasos. Se realizará una revisión de la planificación inicial cada dos semanas, para poder discutir si hay algún problema serio en alguna de las etapas, o alguna laguna de planificación. Si es necesario, se modificará la planificación.

[DP -2]: Fechas límites para hitos incumplidas

Probabilidad: Moderada

Impacto en el proyecto: Serio

Solución: Este riesgo debería evitarse utilizando la solución para el riesgo anterior. Si algún hito no se cumple, se debe verificar la planificación inmediatamente y rehacerla, para evitar reacciones en cadena.

[RT-1]: Problema de compatibilidad con los navegadores

Probabilidad: Baja

Impacto en el proyecto: Tolerable

Solución: Antes de desarrollar el prototipo, se llevará a cabo una tarea de investigación previa para asegurarnos de que existe compatibilidad con al menos un navegador en el mercado para poder desarrollar nuestra aplicación. Las herramientas originalmente planeadas para desarrollar la aplicación del servidor fueron O3D y GWT. Aunque oficialmente son compatibles¹⁴¹⁵ con la mayoría de los navegadores, mantendremos como alternativa alguna tecnología para desarrollar aplicaciones de escritorio (frameworks tales como Java3D¹⁶ o MonkeyEngine¹⁷).

¹⁴ <http://code.google.com/p/quake2-gwt-port/wiki/CompatibleBrowsers>

¹⁵ <http://code.google.com/p/o3d/wiki/FAQs>

¹⁶ <http://java.sun.com/javase/technologies/desktop/java3d/>

¹⁷ <http://www.jmonkeyengine.com/>

[RT -2]: Problemas de velocidad asociados a la realidad aumentada

Probabilidad: Moderada

Impacto en el proyecto: Serio **Solución:** Dado que la realidad aumentada es una tecnología que se puede considerar relativamente nueva para dispositivos portátiles, y que el terminal del que hemos dispuesto (HTC Hero y HTC Magic) no se caracterizan por sus grandes prestaciones, es posible que el rendimiento de la aplicación usando Realidad Aumentada no sea todo lo rápido que cabría esperar. En ese caso no nos queda otra alternativa más que seguir adelante con la tecnología propuesta, dado que no es un problema que podamos resolver. No obstante, la evolución de los dispositivos electrónicos portátiles invita a pensar que en el futuro este riesgo estará solucionado, ya que las prestaciones cada vez son superiores. Nuestra tarea más importante sería desarrollar el modelo necesario para la aplicación, aunque el desarrollo del prototipo se vea seriamente afectado.

[RT -3]: Problemas de velocidad asociados al posicionamiento WiFi

Probabilidad: Moderada

Impacto en el proyecto: Seria

Solución: En caso de que el posicionamiento WiFi no funcionase con el rendimiento requerido, o que las actualizaciones de posicionamiento no pudiesen considerarse en tiempo real, podría afectar severamente al proyecto. La tarea de evacuar en caso de incendio se podría considerar como de crítica en lo referente al tiempo de respuesta, y una simulación que no respetase esto podría considerarse como incompleta. Adicionalmente, utilizaremos los sensores del teléfono móvil para poder determinar la posición con la que el usuario se está moviendo en las zonas en las que la cobertura WiFi no sea óptima, o para poder solucionar posibles espacios “sin cobertura” o con una señal errónea

[RT -4]: Problemas de rendimiento con el terminal móvil

Probabilidad: Baja

Impacto en el proyecto: Serio

Solución: Como se mencionó en el riesgo RT-2, es un problema al que difícilmente podremos hacer frente y buscar alternativas. Nos centraremos en el desarrollo del modelo.

[RT -5]: Funcionalidades no implementadas en la API de OpenGL para Android

Probabilidad: Moderada

Impacto en el proyecto: Catastrófico

Solución: En caso de que la API de OpenGL no permita realizar determinadas funciones en el terminal, deberemos haber considerado alternativas que nos permitan sortear este riesgo, ya que el núcleo del proyecto habría sido golpeado. Para ello consideramos diferentes opciones de implementación de los fuegos en OpenGL: utilizar modelos 3D en diferentes formatos (collada, dae, maya...), sistemas de partículas, objetos compuestos con primitivas OpenGL que nos permitan representar los fuegos y como última alternativa utilizar imágenes estáticas superpuestas en la pantalla.

[REH-1]: Falta de conocimiento en las tecnologías utilizadas

Probabilidad: Moderada

Impacto en el proyecto: Tolerable

Solución: Al realizar la selección de posibles tecnologías sobre las que realizar la implementación, intentamos hacerlo con un enfoque práctica y premiar aquellas tecnologías con las que teníamos experiencia. El único punto que resultaba novedoso a los miembros del equipo de desarrollo era la programación con Android, pero que al estar basada en Java y XML nos resultaba familiar. No se considera un riesgo crítico por tal motivo. En cualquier caso, durante el verano previo al desarrollo del proyecto llevamos a cabo un periodo de formación para los miembros del grupo, durante el que realizamos diferentes tutoriales para construir aplicaciones en Android.

[REH -2]: Problemas de comunicación entre las capas

Probabilidad: Muy baja

Impacto en el proyecto: Serio

Solución: Al realizar la comunicación entre los distintos módulos y plataformas debemos tener en cuenta la posibilidad de que exista un error en la comunicación, teniendo en cuenta que en el posicionamiento WiFi operamos a nivel físico (analizando la intensidad de la señal de los puntos de acceso). Utilizamos tecnologías bien probadas y desarrolladas en otros proyectos diferentes y con las que tenemos cierta experiencia, por lo que la probabilidad de este suceso es muy baja o nula. No obstante preferimos considerarla y tener en cuenta alternativas para los distintos protocolos de comunicación.

[RP-1]: Dedicación al proyecto en periodos como vacaciones y exámenes

Probabilidad: Alta

Impacto en el proyecto: Serio

Solución: Por diferentes cuestiones, los riesgos de personal pueden surgir en cualquier momento y por diferentes motivos. Para hacerles frente intentaremos tener en cuenta todas estas situaciones en la planificación y en cada reunión que tengamos. Confiamos en el compromiso de cada uno de los desarrolladores para mantener informados al resto sobre cualquier situación personal que pudiese afectar a la buena marcha del proyecto.

A continuación mostramos visualmente dos diagramas: el primero representa la probabilidad de cada uno de los riesgos en el proyecto, y el segundo el impacto que tendrían:

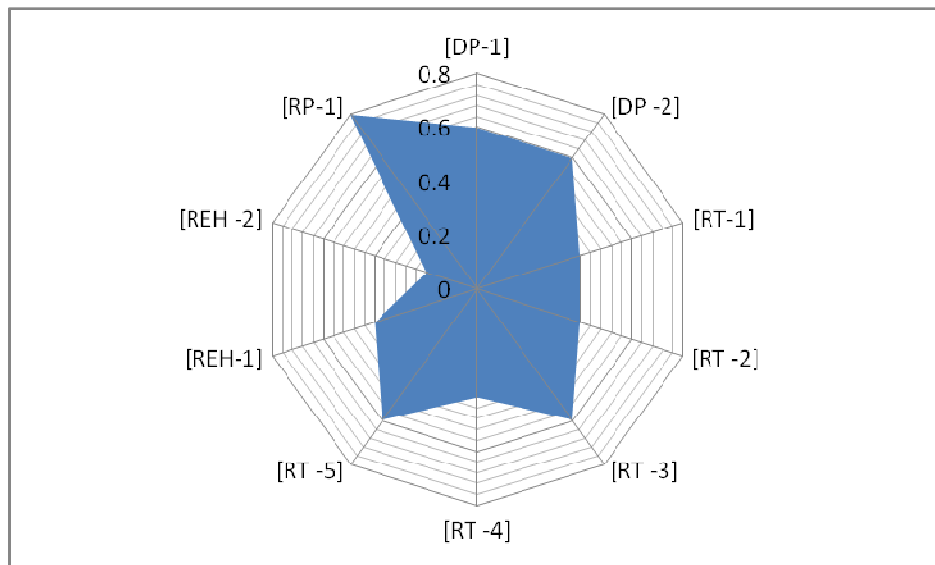


Figura 28: Gráfico de riesgos por probabilidad en el proyecto

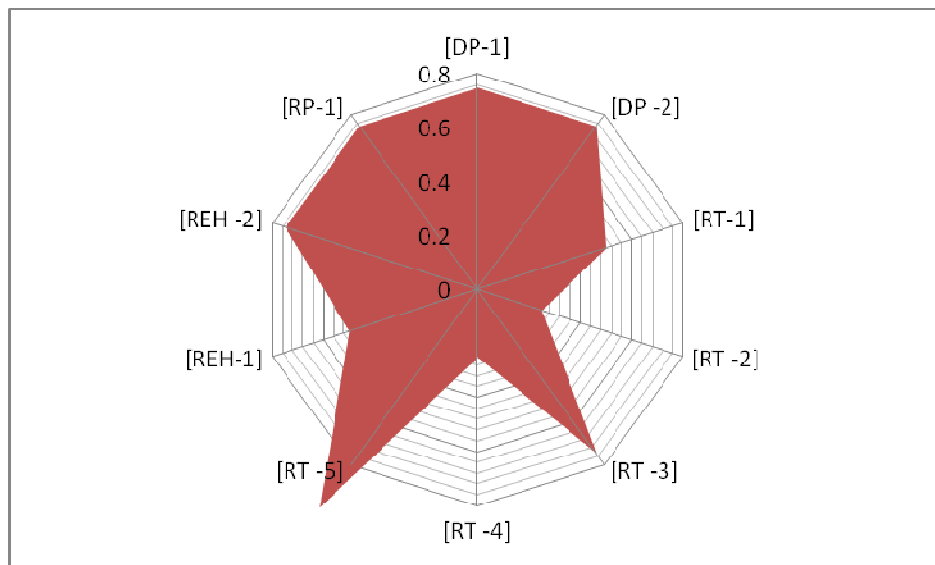


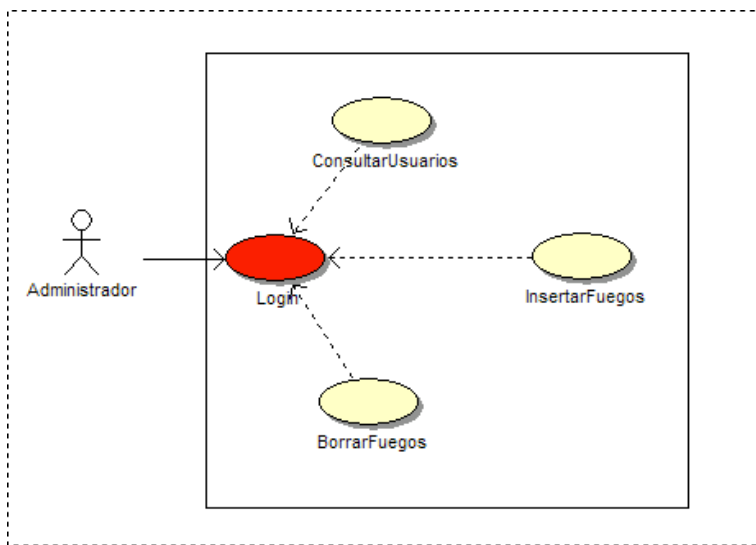
Figura 29: Gráfico de riesgos por impacto en el proyecto

3. Especificación de casos de uso

En esta sección se explican los casos de uso de la aplicación, haciendo una división entre los propios de la aplicación servidor y los de la aplicación cliente [Press05]. Utilizamos un diagrama con los actores y los casos de uso para representarlos visualmente. Se adjunta para cada diagrama una tabla, en la que se explica verbalmente cuáles son los disparadores, precondiciones, post-condiciones, el flujo normal de datos y el flujo alternativo (en el caso de que por ejemplo tenga lugar un error durante la ejecución del mismo).

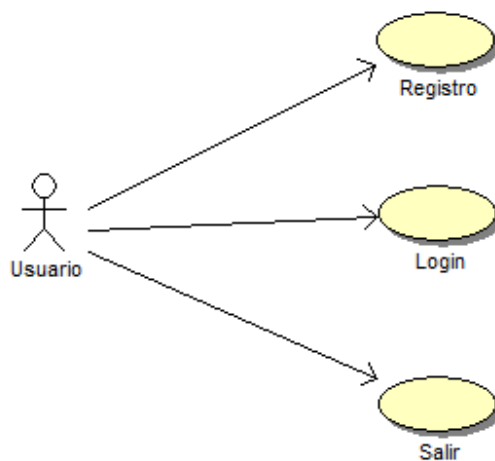
3.1. Casos de uso del servidor

La parte del servidor comprende la parte que se ejecuta exclusivamente en el ordenador y es accesible a través del navegador



Una vez que el usuario se logea en la aplicación web, puede insertar fuegos en la simulación, eliminarlos y consultar los usuarios que están participando en la simulación en ese momento.

3.2. Casos de uso de la aplicación móvil



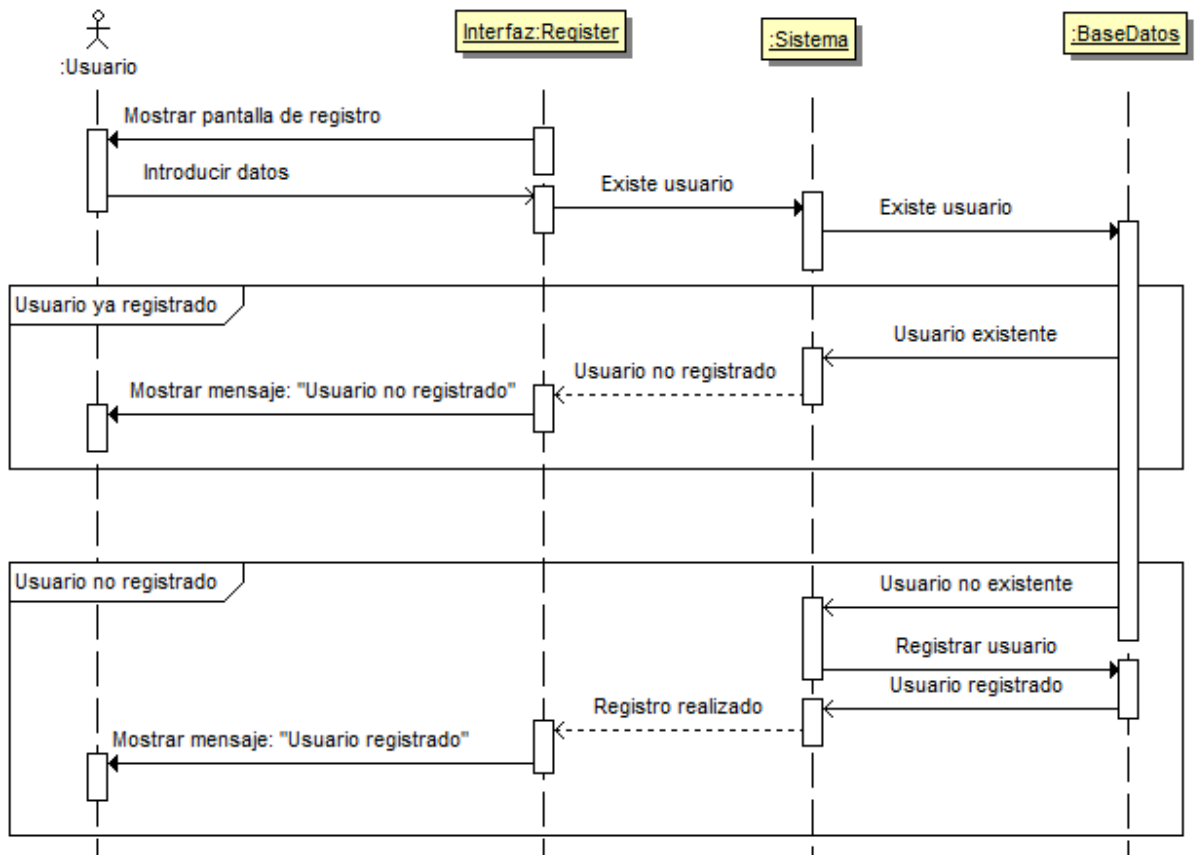
Caso de uso: Registro

ID Caso de uso:	INI-01		
Nombre Caso de uso:	Registro		
Creado por:	Francisco Javier Moreno Nacarino	Última actualización:	Javier Plá Herrero
Fecha creación:	19/05/2010	Fecha última actualización:	26/06/2010

Actores:	Usuario que no está registrado en la aplicación.
Descripción:	Este caso de uso se refiere al registro de un usuario en la aplicación. Cuando la aplicación se inicia, el usuario pulsará el botón de registro y se cargará el formulario para registrarse en la aplicación. En dicho formulario, el usuario deberá introducir su nombre y apellidos y su nombre de usuario y contraseña para poder entrar en la aplicación. El registro finaliza una vez se han introducido todos los datos y se ha pulsado el botón Aceptar.
Disparador:	El usuario debe seleccionar la opción Registrarse en la pantalla de inicio de la aplicación.
Precondiciones:	<ol style="list-style-type: none"> 1. El usuario no está registrado y quiere utilizar la aplicación. 2. La conexión a Internet debe estar disponible en la aplicación.
Postcondiciones:	<ol style="list-style-type: none"> 1. El usuario queda registrado y puede acceder a la aplicación introduciendo su nombre de usuario y contraseña en la pantalla de inicio.
Flujo normal:	<ol style="list-style-type: none"> 1. El usuario abre la aplicación 2. En la pantalla principal, se muestran dos campos para introducir el nombre de usuario y la contraseña, y los botones de Login y Registrarse. 3. El usuario elige la opción para registrarse. 4. Se muestra una nueva pantalla con varios campos

	<p>para efectuar el registro: nombre y apellidos, nombre de usuario y contraseña.</p> <p>5. El usuario rellena todos los campos y pulsa el botón de Aceptar.</p> <p>6. Se comprueba que el usuario no está registrado ya en la aplicación: si el nombre y apellidos no pertenecen ya a otro usuario, o bien si el nombre de usuario ya ha sido utilizado.</p> <p>7. Un mensaje informa de que el registro ha sido satisfactorio y se regresa a la pantalla principal (formulario de login).</p>
Flujo alternativo:	<p>5.1.a. El usuario no ha rellenado todos los campos para registrarse.</p> <p>5.1.b. Se muestra un mensaje advirtiendo de que no se han rellenado todos los campos.</p> <p>5.1.c. Vuelve a aparecer el formulario de registro.</p> <p>5.2.a. El usuario pulsa el botón Cancelar en el formulario de registro.</p> <p>5.2.b. La aplicación vuelve a la pantalla de inicio de sesión.</p> <p>6.1.a. Se informa mediante un mensaje que el usuario ya está registrado con el nombre y apellidos introducidos en el formulario de registro.</p> <p>6.1.b. Vuelve a aparecer el formulario de registro.</p> <p>6.2.a. Aparece un mensaje informando de que el nombre de usuario introducido ya ha sido utilizado por otro usuario registrado.</p> <p>6.2.b. Se carga de nuevo el formulario de registro.</p>
Excepciones:	Puede darse un error en la base de datos, por lo que se informará al usuario del problema.
Inclusiones:	-
Prioridad:	Alta
Frecuencia de uso:	Alta

Reglas de negocio:	Un usuario sólo debe registrarse una vez en la aplicación.
Requerimientos especiales:	-
Asumiciones:	-
Notas:	-

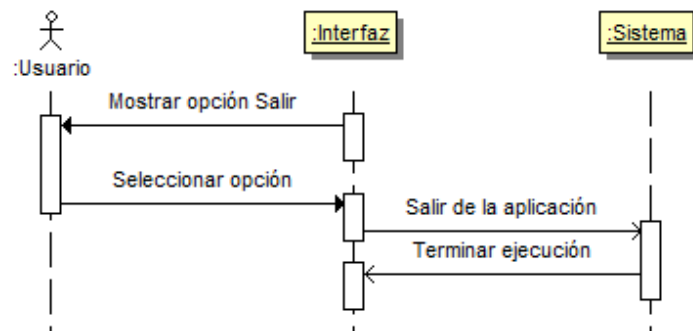


Caso de uso: Salir

ID Caso de uso:	INI-03		
Nombre Caso de uso:	Salir		
Creado por:	Francisco Javier Moreno Nacarino	Última actualización:	

Fecha creación:	19/05/2010	Fecha última actualización:	
-----------------	------------	-----------------------------	--

Actores:	Usuario.
Descripción:	Este caso de uso describe la terminación de la aplicación. Al desplegar el menú en la pantalla de inicio de sesión, aparecerá una opción para salir. La ejecución de la aplicación terminará cuando el usuario seleccione dicha opción.
Disparador:	La opción Salir aparece cuando se despliega el menú en la pantalla de login. Para ello, se debe pulsar el botón correspondiente al menú en el dispositivo móvil.
Precondiciones:	1. El usuario debe encontrarse en el formulario de inicio de sesión.
Postcondiciones:	1. La ejecución de la aplicación finaliza.
Flujo normal:	1. El usuario se encuentra en la pantalla de login. 2. Se despliega el menú y se selecciona la opción Salir del mismo. 3. La aplicación finaliza.
Flujo alternativo:	-
Excepciones:	-
Inclusiones:	-
Prioridad:	Alta
Frecuencia de uso:	Alta
Reglas de negocio:	-
Requerimientos especiales:	-
Asumiciones:	-
Notas:	-



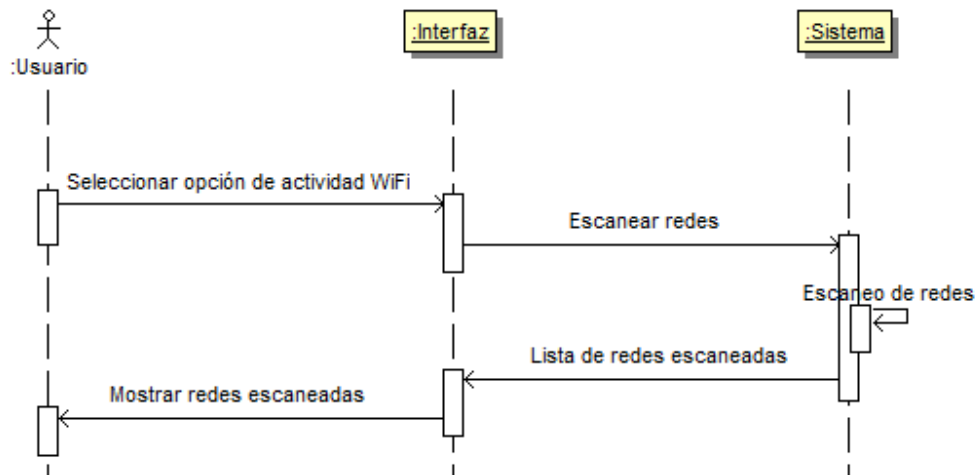
3.2.1. Menú principal de la aplicación

Caso de uso: Actividad WiFi

ID Caso de uso:	MENU-01		
Nombre Caso de uso:	Actividad WiFi		
Creado por:	Francisco Javier Moreno Nacarino	Última actualización:	
Fecha creación:	26/05/2010	Fecha última actualización:	

Actores:	Usuario.
Descripción:	Este caso de uso engloba la búsqueda de redes inalámbricas en el entorno en que se encuentra. Aparecerá en pantalla una lista con las redes escaneadas, mostrando su nombre y dirección MAC. Al seleccionar una de ellas, aparecen la velocidad y frecuencia de dicha red. Dicha pantalla tiene también un menú desplegable con opciones para volver al menú principal, insertar en la base de datos las redes escaneadas en una posición y actualizar la lista de redes escaneadas.
Disparador:	Selección de la opción Ver actividad WiFi del menú principal
Precondiciones:	1. El usuario debe haber accedido a la aplicación.

Postcondiciones:	1. Se mostrará en el dispositivo una lista con las redes inalámbricas detectadas en el entorno de la posición del usuario.
Flujo normal:	1. El usuario selecciona la primera opción del menú principal de AVANTI: Ver actividad WiFi. 2. Aparecerá en pantalla una lista con las redes encontradas durante el escaneo.
Flujo alternativo:	
Excepciones:	Error al intentar actualizar la lista de redes escaneadas. En ese caso, se mostrará un mensaje de error y la ejecución volverá al menú principal de la aplicación.
Inclusiones:	El usuario debe encontrarse en la pantalla del menú principal.
Prioridad:	Alta
Frecuencia de uso:	Media

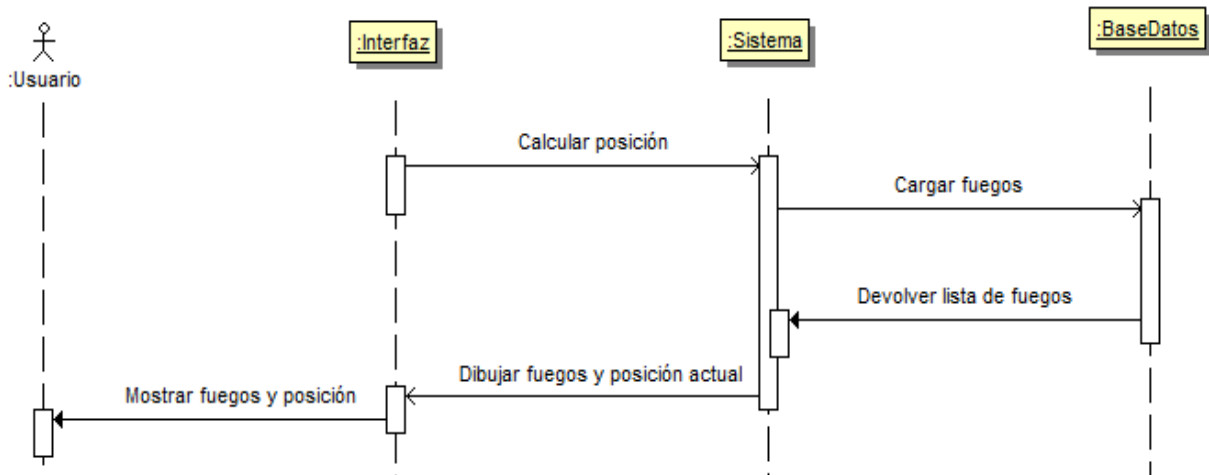


Caso de uso: Posicionar en mapa

ID Caso de uso:	MENU-02		
Nombre Caso de uso:	Posicionar en mapa		
Creado por:	Francisco Javier Moreno Nacarino	Última actualización:	Javier Plá Herrero
Fecha creación:	26/05/2010	Fecha última actualización:	26/06/2010

Actores:	Usuario
Descripción:	Este caso de uso engloba el proceso para que el usuario sea localizado en la vista de mapa. Además, se muestran en el mapa los fuegos que hay almacenados en la base de datos.
Disparador:	Se debe seleccionar la opción “Posicióname en el mapa” que se encuentra en el menú principal de AVANTI.
Precondiciones:	1. El usuario debe haber accedido correctamente a la aplicación, usando su contraseña y su nombre de usuario.
Postcondiciones:	1. El usuario puede ver en la pantalla de su dispositivo móvil un mapa de la primera planta del edificio de la Facultad de Informática. 2. Sobre ese mapa, aparece la ubicación del usuario con un punto rojo. 3. Los puntos azules representan los fuegos almacenados en la base de datos.
Flujo normal:	1. El usuario elige la opción de ubicación en el mapa del menú principal. 2. Se muestra sobre la imagen del mapa la ubicación en que se encuentra el usuario y la posición de los distintos fuegos de la base de datos. Además, se muestran las coordenadas donde está ubicado el usuario en la parte inferior izquierda de la pantalla

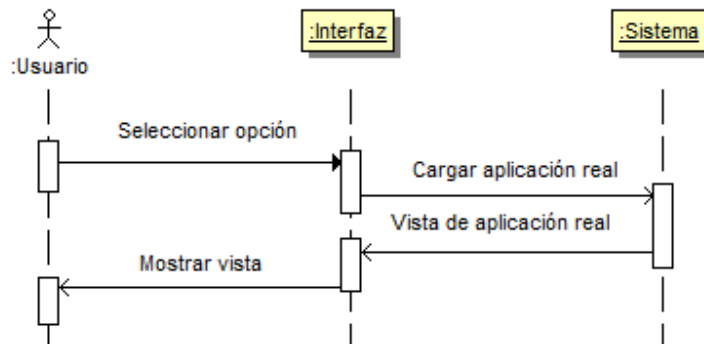
	del dispositivo.
Flujo alternativo:	-
Excepciones:	Puede darse un error al intentar cargar la lista de fuegos de la base de datos o al intentar almacenarlos en un JSONArray (JavaScript Object Notation Array) para su posterior representación sobre la imagen del mapa.
Inclusiones:	El usuario debe encontrarse en la pantalla del menú principal, tras haber iniciado sesión.
Prioridad:	Alta
Frecuencia de uso:	Alta



Caso de uso: Uso del prototipo completo

ID Caso de uso:	MENU-03		
Nombre Caso de uso:	Aplicación real		
Creado por:	Francisco Javier Moreno Nacarino	Última actualización:	Javier Plá Herrero
Fecha creación:	28/05/2010	Fecha última actualización:	26/06/2010

Actores:	Usuario
Descripción:	Explica el acceso a la función de aplicación real de AVANTI. Para ello, el usuario selecciona la opción correspondiente en el menú principal de la aplicación. Se cargará la vista de cámara del teléfono, donde aparecerá dibujado un modelo tridimensional de un fuego si hay un fuego en la base de datos situado en la posición a la que apunta el teléfono.
Disparador:	Elección de la opción Aplicación real del menú de AVANTI.
Precondiciones:	1. El usuario debe estar registrado en la aplicación.
Postcondiciones:	1. Se cargará la vista de la cámara en el terminal y, en caso de haber fuegos en la posición que apunta la cámara, aparecerán representados en un modelo tridimensional.
Flujo normal:	1. Aparece en la pantalla del teléfono móvil el menú principal de AVANTI. 2. El usuario selecciona la opción correspondiente a la aplicación real.
Flujo alternativo:	-
Excepciones:	-
Inclusiones:	Debe realizarse el proceso de acceso a la aplicación (login) para poder probar la aplicación real.
Prioridad:	Alta
Frecuencia de uso:	Alta
Reglas de negocio:	-
Requerimientos especiales:	-
Asumiciones:	-
Notas:	-

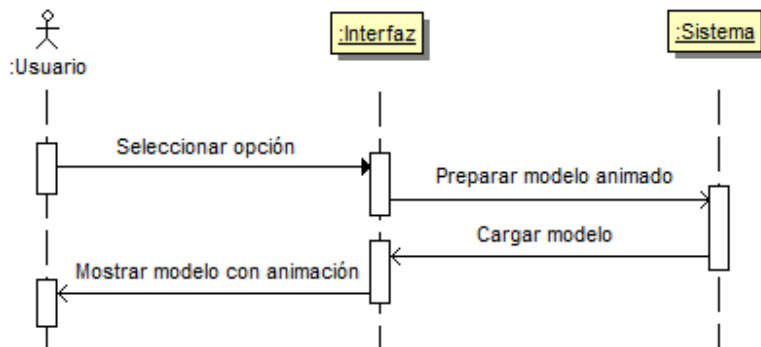


Caso de uso: Prueba de animación

ID Caso de uso:	MENU-04		
Nombre Caso de uso:	Prueba de animación		
Creado por:	Francisco Javier Moreno Nacarino	Última actualización:	
Fecha creación:	28/05/2010	Fecha última actualización:	

Actores:	Usuario
Descripción:	Explica el funcionamiento de la prueba de animación. Cuando el usuario elige esta opción en el menú principal, aparece una animación que consiste en un triángulo dibujado en OpenGL que está rotando constantemente.
Disparador:	Selección de la opción Prueba animación en el menú principal de AVANTI.
Precondiciones:	1. El usuario que lo utilice debe estar registrado en el sistema.
Postcondiciones:	1. Aparecerá en la pantalla del terminal un triángulo dibujado en OpenGL. Dicho triángulo está rotando continuamente.
Flujo normal:	1. El usuario se encuentra en la pantalla principal de AVANTI, con distintas opciones a elegir.

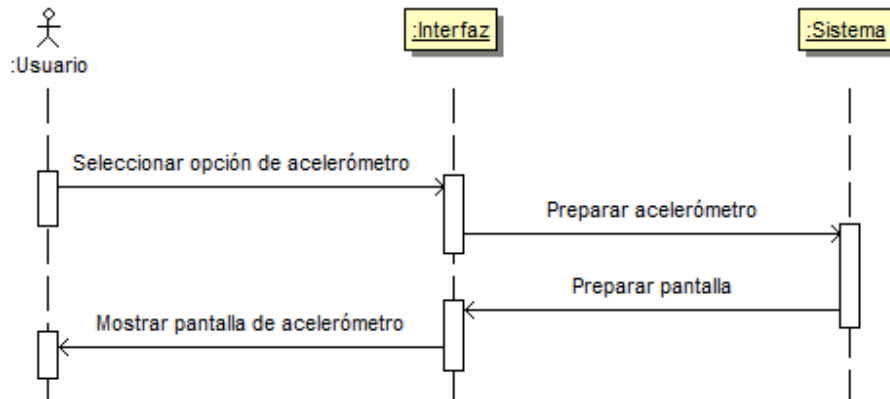
	2. El usuario selecciona la opción de prueba de animación. 3. Se carga la animación sobre la pantalla del terminal.
Flujo alternativo:	-
Excepciones:	-
Inclusiones:	El usuario debe haber accedido con su nombre y contraseña a la aplicación.
Prioridad:	Media
Frecuencia de uso:	Media
Reglas de negocio:	-
Requerimientos especiales:	-
Asumiciones:	-
Notas:	-



Caso de uso: Prueba del acelerómetro

ID Caso de uso:	MENU-05		
Nombre Caso de uso:	Prueba del acelerómetro		
Creado por:	Francisco Javier Moreno Nacarino	Última actualización:	Javier Plá Herrero
Fecha creación:	28/05/2010	Fecha última actualización:	26/06/2010

Actores:	Usuario
Descripción:	Este caso de uso describe el funcionamiento del acelerómetro mediante un test.
Disparador:	El usuario debe seleccionar la opción Prueba del acelerómetro en el menú principal de nuestra aplicación. Se muestra una pantalla con los valores de aceleración en los ejes X, Y, Z, además de los giros sobre los mismos ejes. Además, se muestra por pantalla si el giro realizado en el teléfono es válido o no.
Precondiciones:	1. El usuario debe estar registrado en AVANTI para poder acceder a la prueba del acelerómetro.
Postcondiciones:	1. La pantalla del terminal muestra los valores de aceleración en los ejes X, Y, Z cuando se mueve el teléfono. También aparecen los valores de giro sobre dichos ejes, además de indicar si el giro ha sido válido o no.
Flujo normal:	<ol style="list-style-type: none"> 1. El usuario está en el menú principal de la aplicación, donde aparecen las distintas opciones. 2. El usuario selecciona la opción del acelerómetro. 3. Se carga la pantalla donde se muestran los distintos valores que se miden con el acelerómetro.
Flujo alternativo:	-
Excepciones:	Puede darse un error en la ejecución del acelerómetro.
Inclusiones:	El usuario debe haber iniciado sesión en la aplicación.
Prioridad:	Media
Frecuencia de uso:	Media
Reglas de negocio:	-
Requerimientos especiales:	-
Asumiciones:	-
Notas:	-

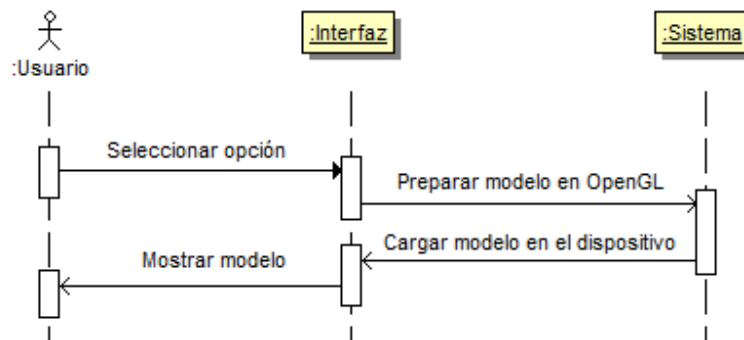


Caso de uso: Prueba de modelos en OpenGL

ID Caso de uso:	MENU-06		
Nombre Caso de uso:	Prueba de modelos en OpenGL		
Creado por:	Francisco Javier Moreno Nacarino	Última actualización:	
Fecha creación:	28/05/2010	Fecha última actualización:	

Actores:	Usuario
Descripción:	Este caso de uso describe una prueba de cómo trabaja la aplicación con los modelos en OpenGL.
Disparador:	Elección de la opción correspondiente (Prueba de modelos en OpenGL) en el menú de la aplicación.
Precondiciones:	1. El usuario debe haberse registrado previamente en la aplicación.
Postcondiciones:	
Flujo normal:	<ol style="list-style-type: none"> 1. En el menú principal de la aplicación, aparecen distintas opciones a seleccionar. 2. El usuario selecciona la penúltima opción del mismo, correspondiente a las pruebas de modelos hechos en OpenGL. 3. Se carga un modelo en OpenGL sobre el que se

	podrán probar las distintas transformaciones que se le pueden aplicar al mismo.
Flujo alternativo:	-
Excepciones:	-
Inclusiones:	Es necesario que el usuario haya accedido a la aplicación mediante la pantalla de login.
Prioridad:	Media
Frecuencia de uso:	Media
Reglas de negocio:	-
Requerimientos especiales:	-
Asumiciones:	-
Notas:	-

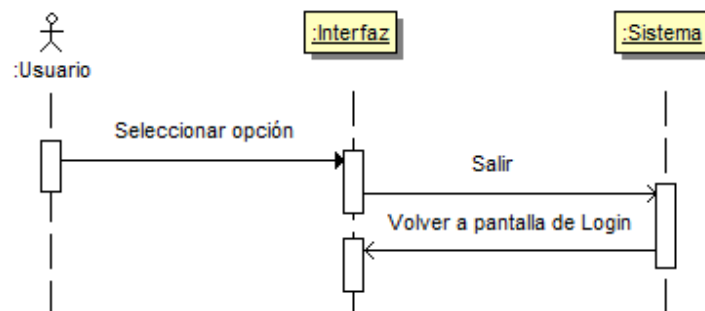


Caso de uso: Logout

ID Caso de uso:	MENU-07		
Nombre Caso de uso:	Logout		
Creado por:	Francisco Javier Moreno Nacarino	Última actualización:	
Fecha creación:	25/05/2010	Fecha última actualización:	

Actores:	Usuario
Descripción:	Este caso de uso describe la desconexión del usuario

	de la aplicación. Al elegir esta opción, se carga la pantalla de login de AVANTI.
Disparador:	El usuario debe seleccionar la opción Logout en el menú principal de la aplicación.
Precondiciones:	1. El usuario debe haber accedido con su nombre y contraseña a la aplicación.
Postcondiciones:	1. El usuario habrá salido correctamente de la aplicación, regresando a la pantalla de inicio de sesión.
Flujo normal:	<ol style="list-style-type: none"> 1. El usuario se encuentra en el menú principal de la aplicación. 2. El usuario elige la opción Logout, situada en dicho menú. 3. Aparece la pantalla de login, habiendo cerrado el usuario su sesión.
Flujo alternativo:	-
Excepciones:	Error en la base de datos a la hora de actualizar el estado del usuario (logueado o no).
Inclusiones:	El usuario debe haber iniciado sesión.
Prioridad:	Alta
Frecuencia de uso:	Alta
Reglas de negocio:	-
Requerimientos especiales:	-
Asumiciones:	-
Notas:	-



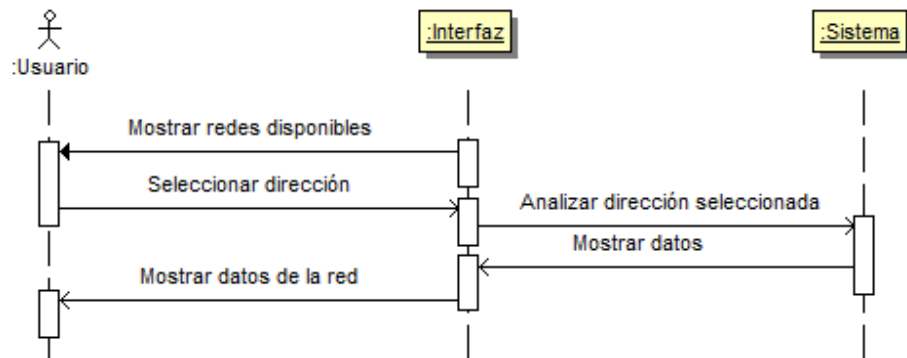
3.2.2. Casos de uso relativos a la actividad Wi-Fi

Caso de uso: Datos de red

ID Caso de uso:	RED-01		
Nombre Caso de uso:	Datos de red		
Creado por:	Francisco Javier Moreno Nacarino	Última actualización:	
Fecha creación:	19/05/2010	Fecha última actualización:	

Actores:	Usuario que está usando la aplicación.
Descripción:	Se muestran los datos de la señal que el usuario ha seleccionado. En concreto, aparecen los siguientes campos: el nombre de la red (SSID), la dirección MAC (BSSID), la frecuencia y la potencia de dicha red.
Disparador:	Seleccionar una de las redes detectadas en el escaneo de actividad WiFi.
Precondiciones:	1. El usuario debe haber accedido a la aplicación.
Postcondiciones:	1. Se muestra un formulario con los datos de la red: nombre, dirección MAC, frecuencia y potencia; además de una opción para volver a la pantalla anterior.
Flujo normal:	<ol style="list-style-type: none"> 1. Aparece un listado con las redes encontradas en el escaneo. 2. El usuario selecciona una de las redes. 3. Se muestra un nuevo formulario con los datos de la red y un botón para volver a la pantalla anterior. 4. Al pulsar el botón Volver, vuelve a aparecer la pantalla con el resultado del escaneo.
Flujo alternativo:	-
Excepciones:	-

Inclusiones:	-
Prioridad:	Alta
Frecuencia de uso:	Media
Reglas de negocio:	-
Requerimientos especiales:	-
Asumiciones:	-
Notas:	-



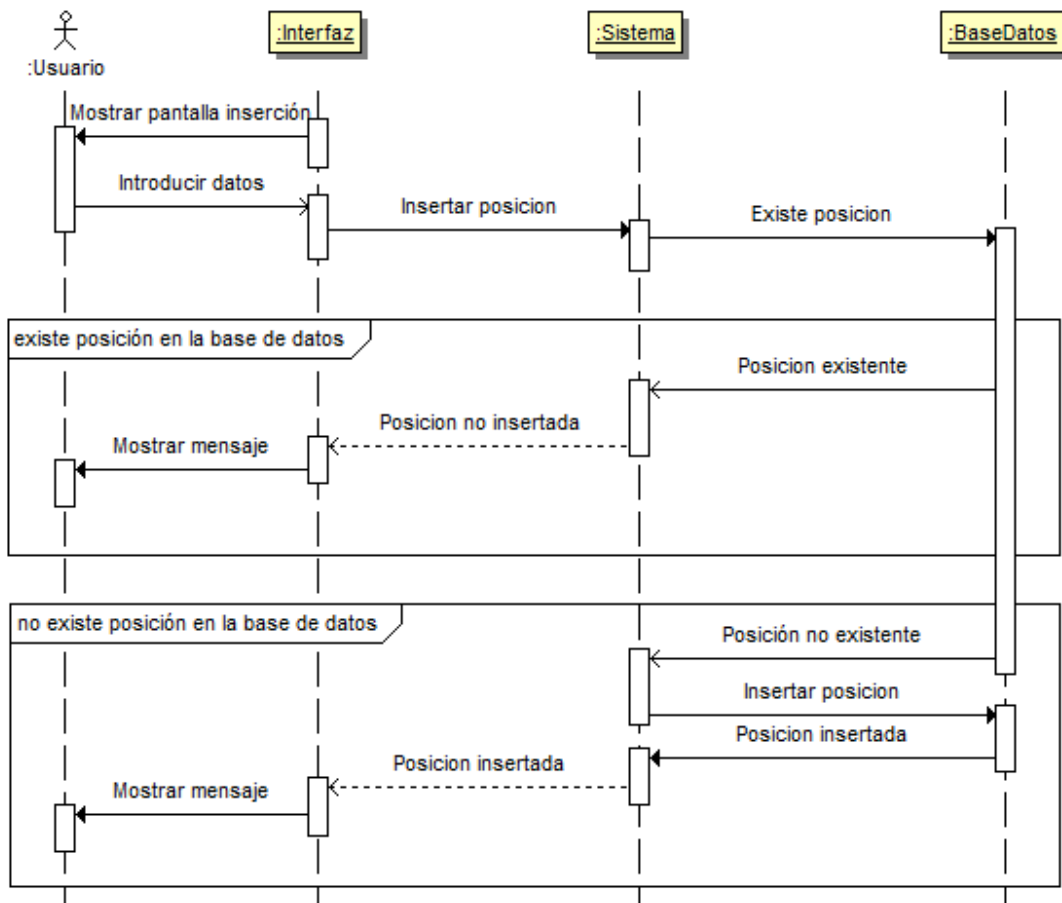
Caso de uso: Insertar posición

ID Caso de uso:	RED-02		
Nombre Caso de uso:	Insertar posición		
Creado por:	Francisco Javier Moreno Nacarino	Última actualización:	
Fecha creación:	20/05/2010	Fecha última actualización:	

Actores:	Usuario que está utilizando la aplicación.
Descripción:	Este caso de uso explica el almacenamiento en la base de datos de las redes escaneadas en una posición determinada. En el menú de la pantalla de Actividad

	WiFi, aparece la opción de Insertar. Cuando el usuario selecciona esta opción, aparece una nueva pantalla donde deben insertarse las coordenadas de la posición actual del usuario. Al elegir la opción Aceptar, se procede a la inserción de las redes escaneadas en dicha posición.
Disparador:	Se elige la opción Insertar en el menú desplegable de la pantalla con los resultados del escaneo.
Precondiciones:	1. El usuario debe estar logueado.
Postcondiciones:	1. Quedarán insertadas en la base de datos las redes escaneadas para la posición que el usuario ha introducido.
Flujo normal:	<ol style="list-style-type: none"> 1. El usuario despliega el menú en la pantalla con los resultados del escaneo. 2. Se selecciona la opción de inserción de dicho menú. 3. Aparece un formulario de petición de datos para insertar las redes en el servidor. El usuario debe introducir las coordenadas (X,Y,Z) 4. El usuario rellena todos los campos y elige la opción de Aceptar. 5. Se comprueba que se han introducido las tres componentes de la posición, y que los datos para esa posición no se encuentran ya en la base de datos. 6. Se informa mediante un mensaje que los datos se han insertado correctamente.
Flujo alternativo:	<ol style="list-style-type: none"> 5.1.a. No se han introducido los datos en todos los campos del formulario. 5.1.b. Un mensaje avisa del error. 5.1.c. La ejecución vuelve a la pantalla de inserción de coordenadas. 5.2.a. Para las coordenadas introducidas, ya existe alguna de las redes escaneadas en la base de datos. 5.2.b. Se informa del error al usuario. 5.2.c. Se regresa a la pantalla de petición de datos

	para insertar la posición.
Excepciones:	Error en la base de datos; en este caso, el usuario recibe un mensaje de error.
Inclusiones:	-
Prioridad:	Alta
Frecuencia de uso:	Media
Reglas de negocio:	-
Requerimientos especiales:	-
Asumiciones:	-
Notas:	-

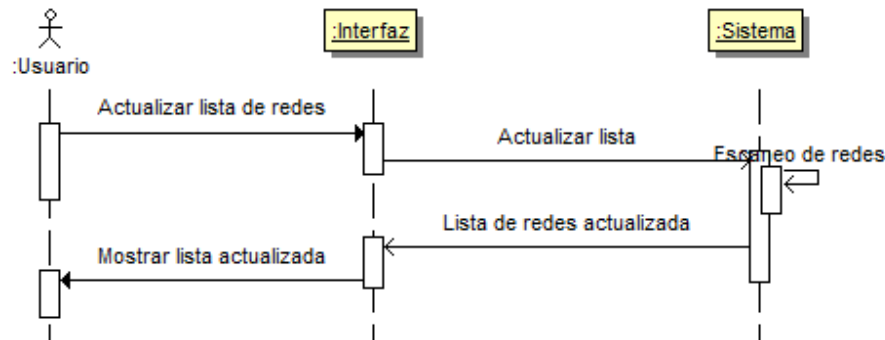


Caso de uso: Actualizar lista

ID Caso de uso:	RED-03		
Nombre Caso de uso:	Actualizar lista		
Creado por:	Francisco Javier Moreno Nacarino	Última actualización:	
Fecha creación:	20/05/2010	Fecha última actualización:	

Actores:	Usuario.
Descripción:	Este caso de uso describe la operación de realizar un nuevo escaneo de redes para actualizar la lista con los resultados escaneados. Es conveniente usar esta opción cuando el usuario ha variado su posición y quiere hacer una nueva comprobación de las redes disponibles
Disparador:	El usuario debe elegir la opción Refrescar lista en el menú de la pantalla de Actividad WiFi.
Precondiciones:	1. El usuario debe estar logueado en la aplicación.
Postcondiciones:	1. La lista de redes escaneadas queda actualizada
Flujo normal:	1. El usuario despliega el menú en la pantalla de Actividad WiFi. 2. El usuario selecciona la opción Refrescar lista. 3. La lista de redes escaneadas queda actualizada.
Flujo alternativo:	-
Excepciones:	Puede darse una excepción en el escaneo de las redes. En ese caso, la aplicación informará del error y volverá al menú principal.
Inclusiones:	Este caso de uso incluye la ejecución normal del caso de uso MENU-01.
Prioridad:	Alta
Frecuencia de uso:	Alta
Reglas de negocio:	-

Requerimientos especiales:	-
Asumiciones:	-
Notas:	-



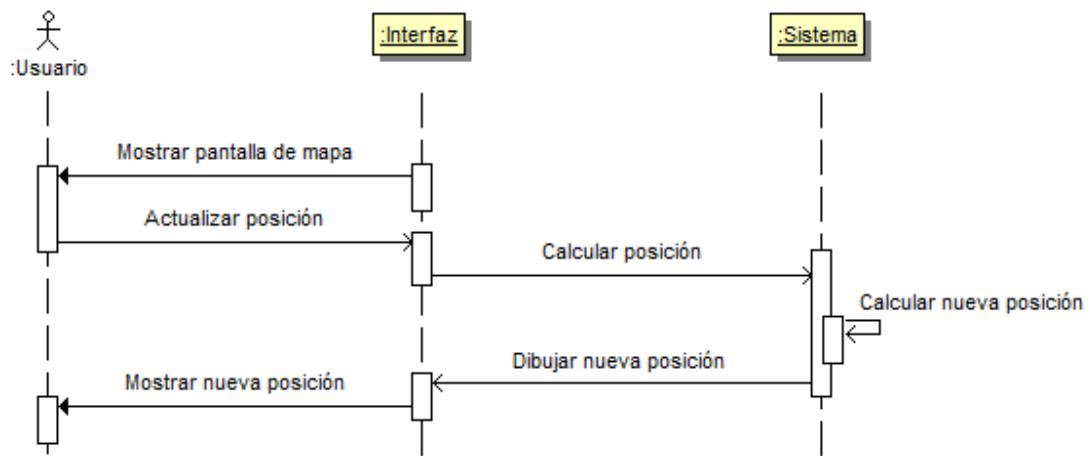
3.2.3. Casos de uso relativos al posicionamiento en el mapa

Caso de uso: Actualizar posición

ID Caso de uso:	POS-01		
Nombre Caso de uso:	Actualizar posición en el mapa		
Creado por:	Francisco Javier Moreno Nacarino	Última actualización:	Javier Plá Herrero
Fecha creación:	25/05/2010	Fecha última actualización:	26/06/2010

Actores:	Usuario
Descripción:	Este caso de uso explica el posicionamiento en el mapa del usuario que está usando la aplicación, cuando se quiere actualizar la posición desde la propia vista de mapa. Del mismo modo, se dibujan sobre el mapa los fuegos que hay almacenados en la base de datos.
Disparador:	Ocurre cuando se desliza el dedo por la pantalla táctil

	en la vista de mapa.
Precondiciones:	1. El usuario debe haber iniciado sesión en la aplicación.
Postcondiciones:	1. La posición del usuario queda actualizada, mostrándola en el mapa.
Flujo normal:	<ol style="list-style-type: none"> 1. En la pantalla de mapa, aparece representado el usuario mediante un punto rojo. También se muestran las coordenadas en las que está situado el propio usuario. 2. El usuario se desplaza con el dedo por la pantalla táctil para moverse por el mapa. 3. La posición del usuario queda actualizada en el mapa. Se muestra la nueva ubicación del usuario en el mapa, y su posición en la parte inferior izquierda de la pantalla del terminal.
Flujo alternativo:	-
Excepciones:	Error a la hora de calcular la nueva posición del usuario. Se informa del error y la ejecución vuelve al menú principal de la aplicación.
Inclusiones:	El usuario debe encontrarse en la pantalla de mapa.
Prioridad:	Alta
Frecuencia de uso:	Alta
Reglas de negocio:	-
Requerimientos especiales:	-
Asumiciones:	-
Notas:	-



4. Análisis de requisitos

En esta sección explicaremos los requisitos del proyecto. Vamos a distinguir entre la aplicación cliente (la que se usa desde Android) y la aplicación del servidor (aquella que sirve para gestionar los clientes conectados y los fuegos). Diferenciamos también entre requisitos funcionales y requisitos no funcionales. Los requisitos funcionales definen qué es lo que debe hacer un sistema: en nuestro caso, estos requisitos definen principalmente la funcionalidad del producto final. Estos requisitos fueron definidos inicialmente por nuestros tutores de proyectos al inicio del curso académico, aunque en función de la evolución del desarrollo fueron modificándose hasta llegar a ser lo que presentamos en este capítulo, siempre previa consulta con nuestros tutores. Los requisitos no funcionales se podrían definir como aquellos que definen en qué condiciones debe funcionar el proyecto. Generalmente se dividen en dos categorías, y así lo hemos hecho nosotros: hardware y software

4.1. Servidor

4.1.1 Análisis de requisitos funcionales

- **Debe funcionar en cualquier plataforma:** para cumplir este requisito funcional, decidimos realizar una aplicación web. Usamos GWT para desarrollarla, y al hacer pruebas de funcionalidad sobre distintos navegadores obtuvimos resultados satisfactorios. La aplicación es

relativamente sencilla de instalar. Para facilitarlo aún más, proveemos una máquina virtual ya configurada con la aplicación y un servidor web para servirla, de manera que es tan sencillo como instalar un gestor de máquinas virtuales (VMWare, por ejemplo) y abrir el fichero proporcionado.

- **Debe hacer sencillo que el proyecto pueda ser expandido:** de cara a una posible ampliación del proyecto o para añadir nuevas características, se debe proporcionar una arquitectura modular en la medida que sea posible. Para ello deberemos esforzarnos en definir los módulos lo mejor posible, y separar lógicamente cada capa de acceso para facilitar la tarea a un futuro programador
- **Debe proporcionar conexión en tiempo real con el terminal móvil:** para que el servidor pueda ser usado en tiempo real, todas las actualizaciones que se lleven a cabo en este deben poder visualizarse en tiempo real en el terminal móvil.
- **Debe almacenar información histórica sobre rutas de incendio, escenarios, etc:** para poder efectuar estudios estadísticos, análisis sobre los errores más comunes o cualquier investigación al respecto, se debe almacenar información histórica sobre todos los escenarios relativos a la ejecución de evacuaciones de incendios

4.1.2 Análisis de requisitos no funcionales: Software

- **Versión de Java 1.6.0_02:** esta es la versión de Java que utilizaremos para desarrollar la aplicación de escritorio.
- **Versión 2.0 de GWT.** Versión que usaremos para desarrollar.

4.2. Aplicación móvil

4.2.1 Análisis de requisitos funcionales

- **Debe ser capaz de acceder a bajo nivel a la capa de comunicación:** con el fin de llevar a cabo el posicionamiento WiFi, la aplicación

necesitará acceder a datos de bajo nivel sobre los puntos de acceso con el fin de llevar a cabo la triangulación necesaria.

- **Debe ser capaz de sobreponer una capa en OpenGL sobre una vista de cámara:** un requisito que puede parecer trivial, pero que tiene que estar soportado por la API con la que trabajemos. La capa de modelado en OpenGL tiene que proveer acceso a los valores de transparencia para poder conseguir llevar a cabo técnicas de Realidad Aumentada

4.2.2 Requisitos no funcionales: software

- **Versión de Android: 1.6.** Esta es la versión con la que empezamos a desarrollar para Android, aunque a mitad de desarrollo surgieron más versiones. Hubo algunos problemas serios que nos hicieron replantearnos el cambiar de versión (la vista de cámara tiene reportado un bug serio en la versión 1.6), pero finalmente conseguimos solucionarlos.

4.2.3 Requisitos no funcionales: hardware

- **HTC Hero – Magic:** estos dos dispositivos son de los que dispondremos para desarrollar la aplicación. El funcionamiento debe ser asegurado para ambos terminales, ya que las pruebas y demostraciones se harán utilizando los mismos (sin perjuicio de que se consiga hacer correr la aplicación en otros terminales)

Capítulo V

Desarrollo del proyecto

1. Objetivos principales

En esta sección explicaremos los objetivos principales y las expectativas que teníamos depositadas en el proyecto, divididas en iteraciones, y los objetivos finales que hemos conseguido.

1.1 Iteración preliminar

Esta etapa tenía como objetivo definir un producto concreto y analizar las posibilidades que teníamos de convertirlo en realidad. Tras diversas reuniones con nuestros tutores fuimos definiendo el producto, e internamente categorizando los componentes como “must” y “should”, esto es: módulos que era obligatorios desarrollar para que el objetivo pudiese considerarse como cumplido, y módulos a los que podríamos dedicarnos si los objetivos previos habían sido alcanzados, o si por solapamiento de tareas alguno de los miembros se encontraba sin hitos o tareas a conseguir.

1.2. Primera iteración

Después de haber definido los objetivos principales de la aplicación, la primera iteración consistió en prototipar la aplicación para Android, de manera que nos sirviese para aprender algo más de la tecnología con la que íbamos a trabajar mientras hacíamos pequeñas aplicaciones y nos sirviese para establecer expectativas realistas en cuanto a los tiempos. En esta etapa desarrollamos la interfaz básica de nuestro programa (que posteriormente fuimos incrementando conforme surgían nuevas necesidades) y trabajamos el aspecto gráfico inicial de la aplicación.

1.3. Segunda iteración

Con un prototipo desarrollado era hora de aplicarle algo de funcionalidad. Nuestra aplicación tenía dos módulos que iban a resultar básicos: el módulo de realidad aumentada y el módulo de posicionamiento wifi en interiores. Era necesario comenzar a desarrollar unos modelos de representación para la facultad en ambas tecnologías, y aunque la teoría la teníamos bien conocida, necesitábamos un punto de apoyo práctico para poder contrastar nuestras propuestas. En esta iteración se escaneó la

información de los puntos de acceso de la facultad acorde a las técnicas propuestas en [EvMa06] y [WLSB07] para poder desarrollar un posterior modelo de datos acorde a los resultados que recibiésemos.

1.4 Tercera iteración

La tercera iteración supuso dos partes de trabajo paralelas: por una parte, se desarrolló el modelo de datos para la representación de posiciones en la facultad basada en la información que la anterior iteración había proporcionado. Esto permitió trabajar en la implementación de los módulos de cálculo de rutas de escape en la parte del cliente. Por otro lado, se desarrolló la aplicación servidor que permitía extraer información contextual de los usuarios que se encontraban usando la aplicación, e interactuar a un instructor e insertar fuegos a tiempo real en la aplicación, lo que añadía dinamismo a la aplicación

1.5. Cuarta iteración

En esta iteración se realizó la integración del trabajo anterior con la parte de realidad aumentada: es decir, las transformaciones desde un espacio 2D que permitiesen visualizar en 3 dimensiones el escenario e interactuar con él. Fue quizás la más compleja, ya que supuso definir errores que existían en iteraciones previas y que detectamos en esta fase, al tener que realizar las transformaciones que eran necesarias para poder usar la realidad aumentada. Esta iteración incluyó pruebas generales sobre la aplicación que iban resultando necesarios, así como el cierre del código, de la documentación y de la memoria (un aspecto que también consumo un tiempo importante en el proceso de desarrollo).

2. Proceso de trabajo

Esta sección contiene una información más detallada sobre el proceso de desarrollo durante el proyecto, según la metodología propuesta en [Press05]

2.1. Fase inicial

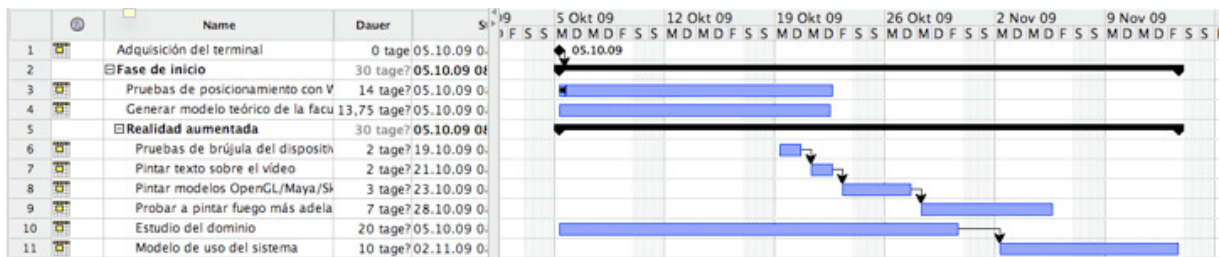


Diagrama de proyecto de la fase inicial

En esta primera etapa realizamos el prototipado de la aplicación y realizamos distintos programas pequeños y “prueba de concepto” que nos sirviesen para poder adquirir conocimiento y emitir predicciones de tiempo y fases más certeras. Cabe destacar que experimentamos bastantes problemas con las pruebas sobre la pantalla, pruebas que realmente no terminamos de cerrar en esta etapa y fuimos arrastrando durante todo el proyecto, hasta que conseguimos cerrarlo casi al final.

2.2. Fase de elaboración

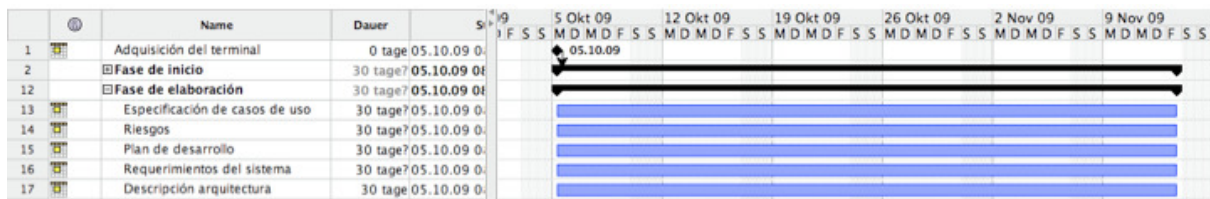
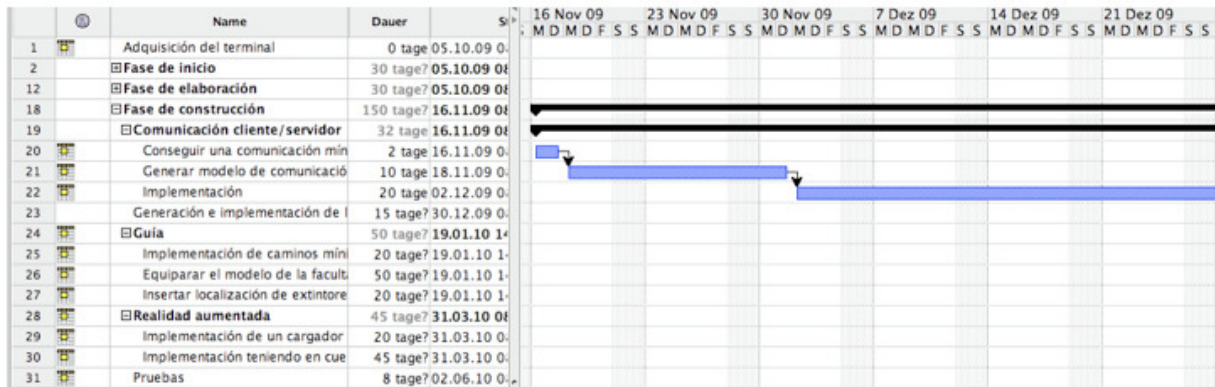


Diagrama de proyecto de la fase de elaboración

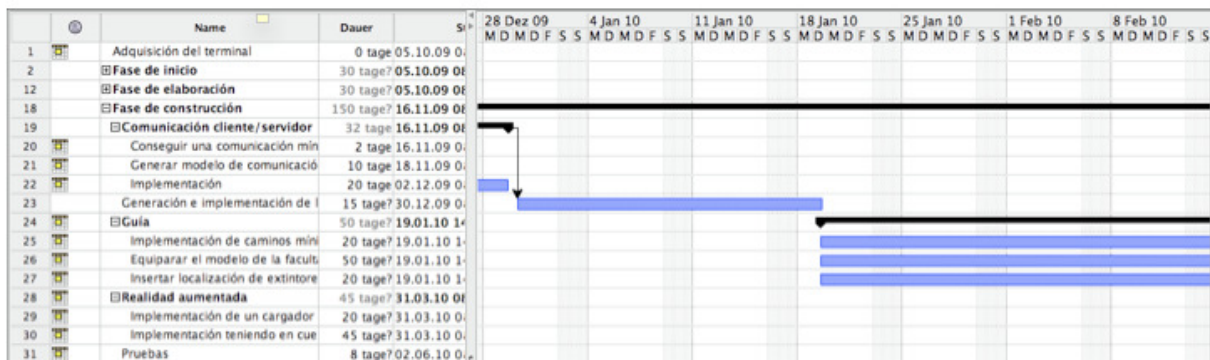
La fase de elaboración supone la redacción en términos formales del proceso que se va a seguir para desarrollar el proyecto, incluyendo explicaciones detalladas de los casos de uso, riesgos que se van a afrontar, requisitos funcionales y no funcionales y descripciones a alto y bajo nivel del desarrollo. Aunque esta etapa nos proporciona una buena base que permaneció inalterable en su mayor parte, se fueron realizando

modificaciones conforme avanzaba el proyecto y detectábamos debilidades de diseño, o nuevas necesidades iban surgiendo.

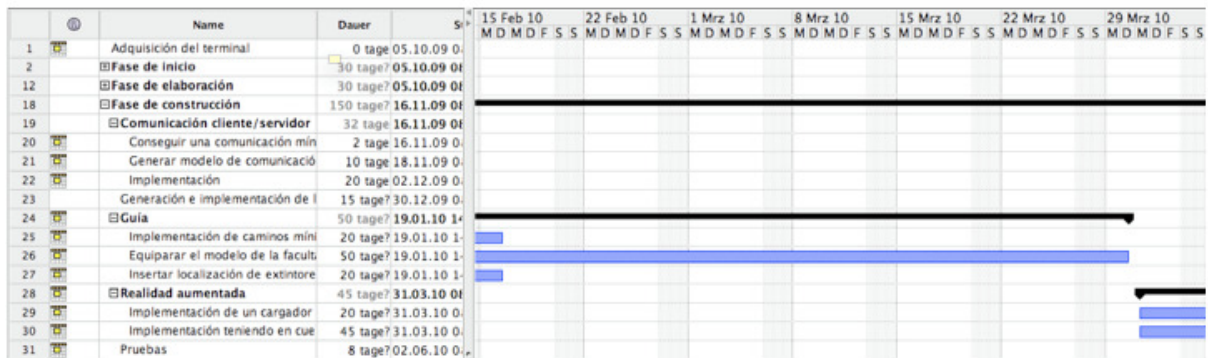
2.3. Fase de construcción



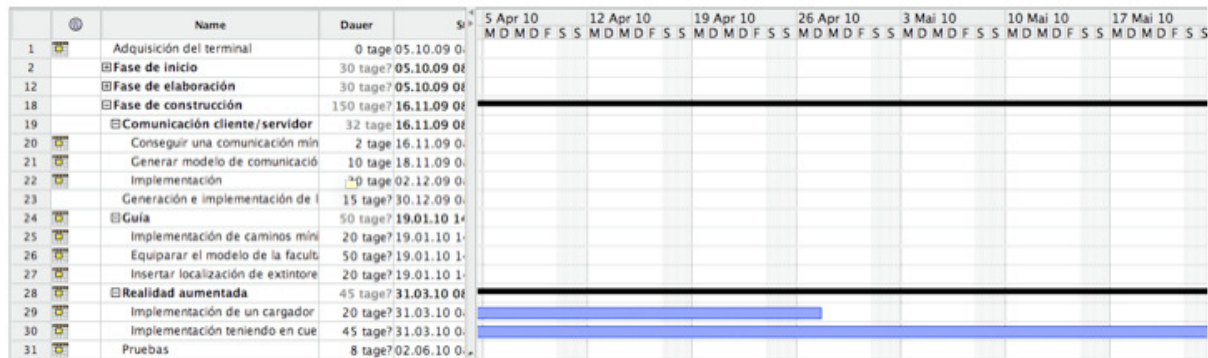
Primera etapa del diagrama de proyecto de la fase de construcción



Segunda etapa del diagrama de proyecto de la fase de construcción



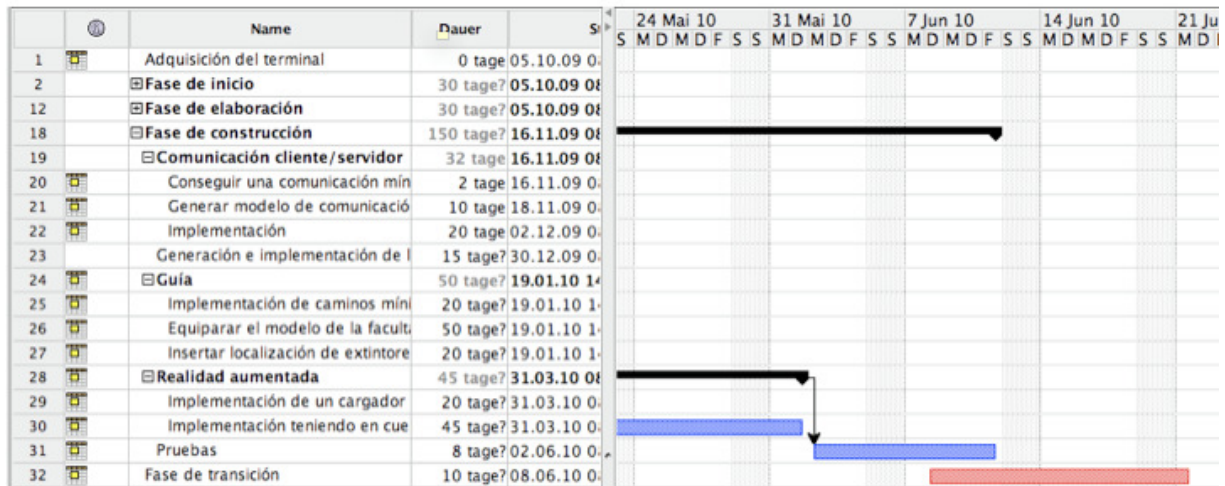
Tercera etapa del diagrama de proyecto de la fase de construcción



Cuarta etapa del diagrama de proyecto de la fase de construcción

Se ha dividido en 4 diagramas para poder facilitar la visualización de los mismos, a lo largo de una línea visual extensa. En esta etapa subestimamos principalmente el módulo de realidad aumentada. Dada nuestra inexperiencia con programación en 3D y OpenGL, y las limitaciones propias de un terminal móvil, extendimos más de lo previsto (tanto hacia el pasado como hacia el futuro) nuestra planificación para desarrollar la realidad aumentada. Los principales afectados fueron el hito de pruebas, que finalmente fue incorporado al resto de fases para poder llevar un control actualizado de la validez de nuestros desarrollos, y el hito de guía, del que algunos módulos calificados como “should” no fueron realmente implementados, como la interacción con extintores.

2.4. Fase de transición



La fase de transición fue desarrollada sin mayor percance. Se dispuso todo el software en una máquina propia de NIL. Más concretamente: la aplicación web y las bases de datos que conforman la aplicación fueron desplegadas en esta máquina, con lo que nos asegurábamos cierta estabilidad y posibilidad de acceder en cualquier momento. Además, el hecho de que estuviese en red local con la red de la facultad nos aseguraba de cara a la exposición pública del proyecto y una posible demo una velocidad que habría sido difícil de conseguirlo con los recursos disponibles anteriormente. En el pasado, habíamos ubicado nuestra aplicación en una antigua máquina de la que disponíamos y a la que no dábamos uso, pero con un tiempo bastante superior a los 10 años.

Capítulo VI

Tests

1. Introducción

En este capítulo se proporcionan los tests que se han ejecutado sobre la aplicación, para validar y comprobar errores de funcionamiento (y depurar, si procediese). Se han realizado pruebas unitarias JUnit sobre la aplicación, pruebas de carga y rendimiento sobre la base de datos y tests sobre la precisión del posicionamiento WiFi.

2. JUnit

JUnit es un conjunto de librerías de Java, que se emplean en programación para hacer pruebas unitarias de aplicaciones implementadas en este lenguaje. Permite ejecutar clases Java de forma controlada, evaluando el funcionamiento de cada uno de los métodos de la clase y comprobando si el resultado es el correcto. En otras palabras, se evalúa el valor que debe devolver la función en base de un determinado valor de entrada; en este caso, JUnit devolverá que la prueba ha sido satisfactoria. En cambio, si el valor esperado no coincide con aquél que un método devuelve, JUnit devolverá un fallo en dicho método.

Otra de las funcionalidades de JUnit es la de controlar las pruebas de regresión; este tipo de pruebas deben realizarse en el caso de que una parte de código haya sido modificada y se quiera comprobar que el nuevo código funcione correctamente, sin haber alterado su funcionalidad original.

Pruebas sobre el paquete com.avanti.routes: la clase *ListaCuadrantes* es la que contiene toda la funcionalidad relativa a la división en cuadrantes de la primera planta del edificio de la Facultad de Informática, así como al cálculo del camino más corto entre la situación actual del usuario y la salida más cercana.

Por lo tanto, se han realizado pruebas sobre los métodos *caminoMasCorto*, *listaAdyacentes*, *numCuadrante*, *dirArista*, *bloquearCuadrante* y *desbloquearCuadrante*. Estos métodos son fundamentales para el cálculo de la ruta más corta hacia una salida aplicando el algoritmo de caminos mínimos de Dijkstra, salvo el método *dirArista*, usado para la representación gráfica del camino a seguir en la vista de mapa de la aplicación.

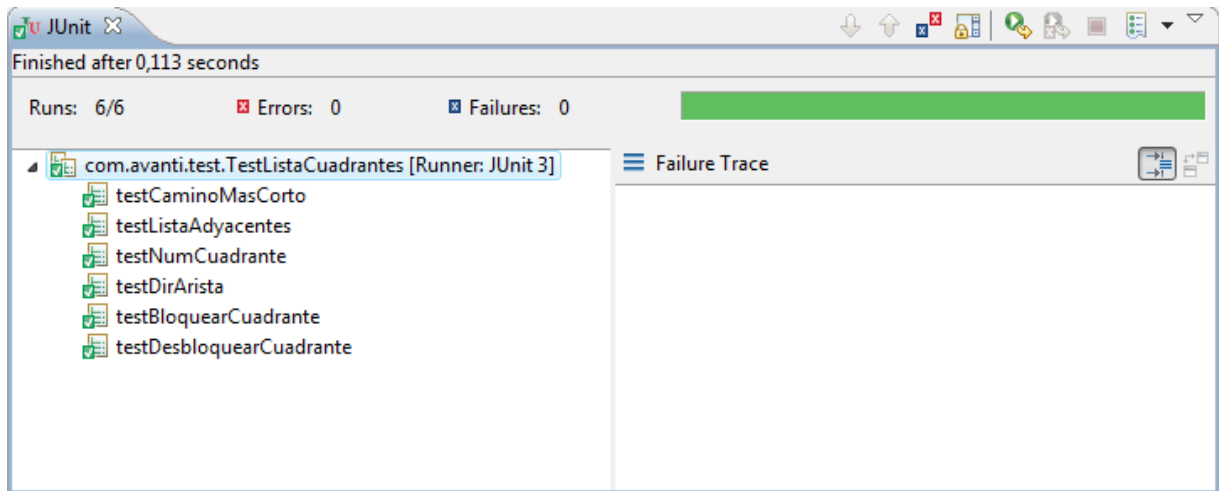


Figura 30: Pruebas con JUnit

En las pruebas con JUnit sobre este paquete, no se han detectado errores importantes debido a que se realizaron muchas pruebas manualmente sobre estos métodos, todas ellas previas a la creación del TestCase empleado en este caso. Como la división en cuadrantes se fue realizando de forma gradual, todos estos métodos se probaron varias veces en cada ampliación del número de cuadrantes para verificar el correcto funcionamiento de dichos métodos.

3. Descripción de los equipos utilizados para las pruebas

Para la realización de las pruebas se utilizó el equipo listado a continuación. No se accedió a través de red local a la base de datos, sino a través de Internet, lo que afectó negativamente a la velocidad de acceso y al rendimiento de la base de datos

Procesador Intel T2130 1,86 GHz

Memoria RAM: 1 GB

Sistema operativo Windows Vista de 32 bits

Conexión a Internet: 8192 Kbps de download y 900 Kbps de subida

4. Pruebas sobre bases de datos

En este apartado se listan las pruebas que se ejecutaron sobre la base de datos así como la metodología aplicada y los resultados obtenidos

4.1. Pruebas de carga

Tanto en la base de datos local como en la remota, se han empleado las mismas tablas con el mismo número de registros cada una.

<i>Tabla</i>	<i>Número de tuplas</i>
FirePosition	2
Position	1
Repository	1050
Users	2

Número de entradas en cada tabla de la base de datos

4.2. Pruebas de rendimiento

En este tipo de pruebas, hemos realizado las operaciones que hemos determinado como funciones críticas sobre una base de datos remota. No hemos contemplado comparar los tiempos con una base de datos local; pero sería posible crear una base de datos local en el terminal mediante SQLite3, un gestor de datos que el sistema operativo Android tiene internamente.

Para todas las funciones, se han tomado 5 mediciones diferentes y se ha elegido el tiempo correspondiente a la media aritmética de dichas mediciones.

<u>Caso de uso/Función</u>	<u>Remoto (s)</u>
Registro en la aplicación	6,535 s.
Login	6,398 s.
Logout	6,040 s.
Insertar nueva posición	12,110 s.
Posicionar en mapa	10,485 s.

Tiempo de ejecución de algunos casos de uso

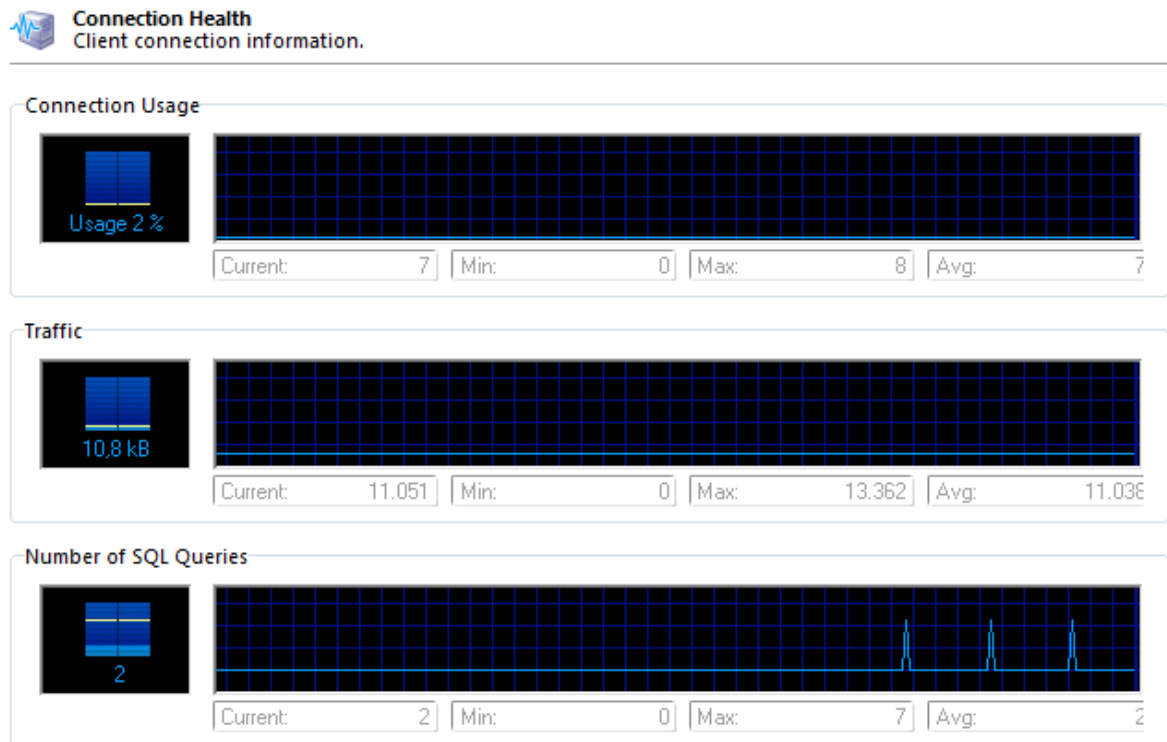


Figura 31: Consumo de las operaciones de registro, login y logout

En primer lugar, analizamos el registro de un usuario nuevo en AVANTI, que se corresponde con el primer pico de la gráfica. Esta operación implica la inserción en la base de datos de los campos introducidos en el formulario de registro (nombre, apellidos, nombre de usuario y contraseña), además de otros campos que indican la última dirección IP desde la que dicho usuario ha accedido a la aplicación o la fecha del último acceso (login). Para esta operación, se realizan dos accesos a la base de datos. En la primera consulta, se comprueba si el usuario ya existe en la tabla Users de la base de datos; en caso de no encontrarlo, se realiza una segunda consulta para insertar el usuario en la tabla correspondiente con los valores introducidos en el formulario. Por defecto, se establecen a null los valores de *lastLoginTime* y *lastLoginIP*, y los campos *isAdmin* y *login* tendrán valor 0 (equivalente a falso).

Vemos en la tabla anterior que la operación tarda aproximadamente unos 6,5 s. Esto se debe principalmente a que la base de datos es remota, además de que el

acceso a la misma se hace mediante scripts PHP.

El segundo pico corresponde a la operación de acceso a la aplicación (login). En este caso, también se realizan dos accesos a la base de datos: una consulta para comprobar si existe un usuario en la base de datos con el nombre y contraseña introducidos en la pantalla de acceso, y otra para actualizar el campo *login* de la tabla de usuarios a 1, así como el campo relativo a la dirección IP desde la cual se ha hecho la operación y al momento en el que se ha realizado el último acceso a la aplicación.

Por último, el tercer pico se refiere a la operación de salida de la aplicación (logout). Al realizarse esta operación, el campo *login* se pone a 0, indicando que el usuario no está conectado a la aplicación. En este caso, también se realizan dos operaciones sobre la base de datos como en el caso del acceso a AVANTI: una consulta y una operación para actualizar el campo *login* a 0.

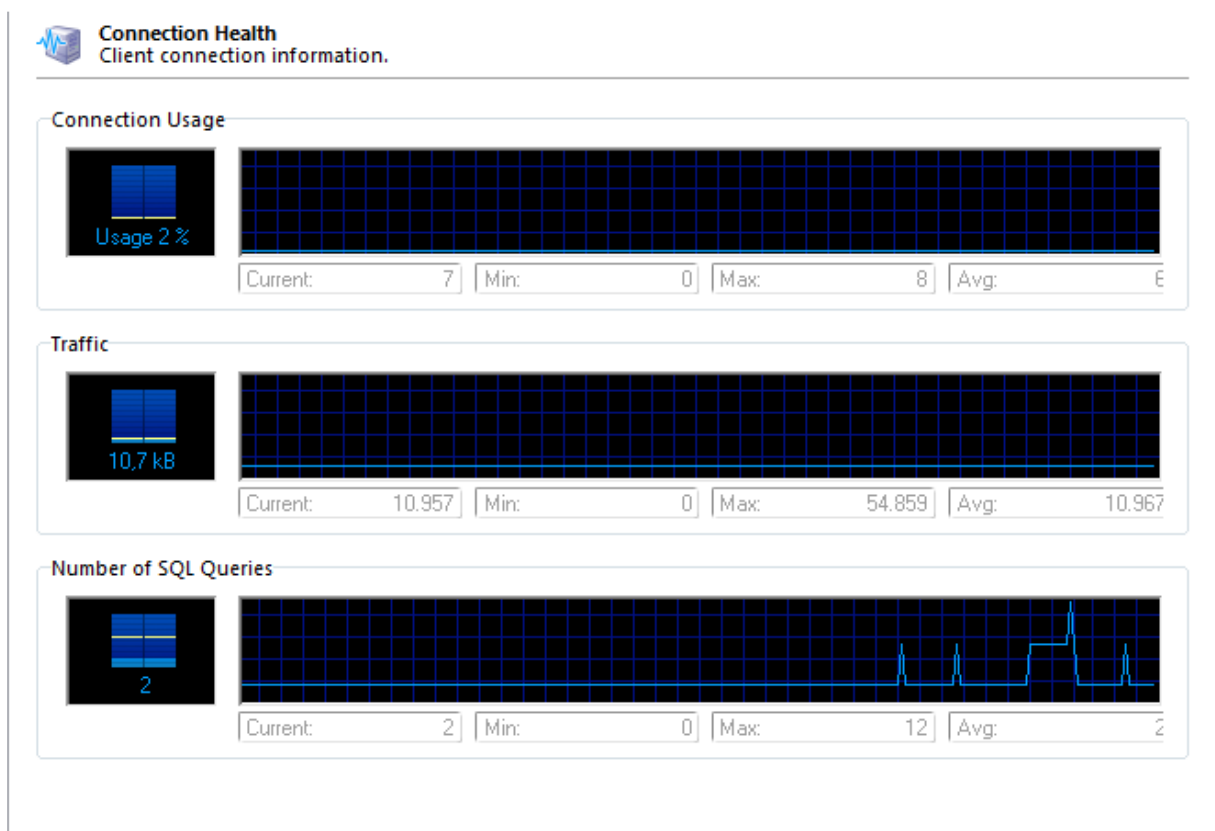


Figura 31: Consumo de la operación “insertar posición”

La siguiente gráfica muestra la acción de insertar una nueva posición en la base de datos, y se corresponde con el tercer pico. En este caso, el usuario introduce las coordenadas X, Y, Z del punto en el que se encuentra. La base de datos almacena para dicha posición la dirección MAC y la intensidad de cada una de las redes detectadas en dicha posición, además de las propias coordenadas introducidas. En este caso, se ejecutan también dos operaciones sobre la base de datos, similares a las del caso de registro: una operación de consulta y otra de inserción.

En este caso, vemos que el pico correspondiente a la inserción de posiciones es más amplio que los anteriores. Esto se debe a que en este caso, se realizan tantas inserciones como redes se han detectado en el escaneo. Para el caso en particular de la gráfica anterior, se habían encontrado 9 redes diferentes en el escaneo. Debemos destacar que el tiempo que se tarda en ejecutar este caso de uso es algo superior que en el resto de casos de uso, pero hay que tener en cuenta que hay un mayor número de accesos a la base de datos que en los otros casos.

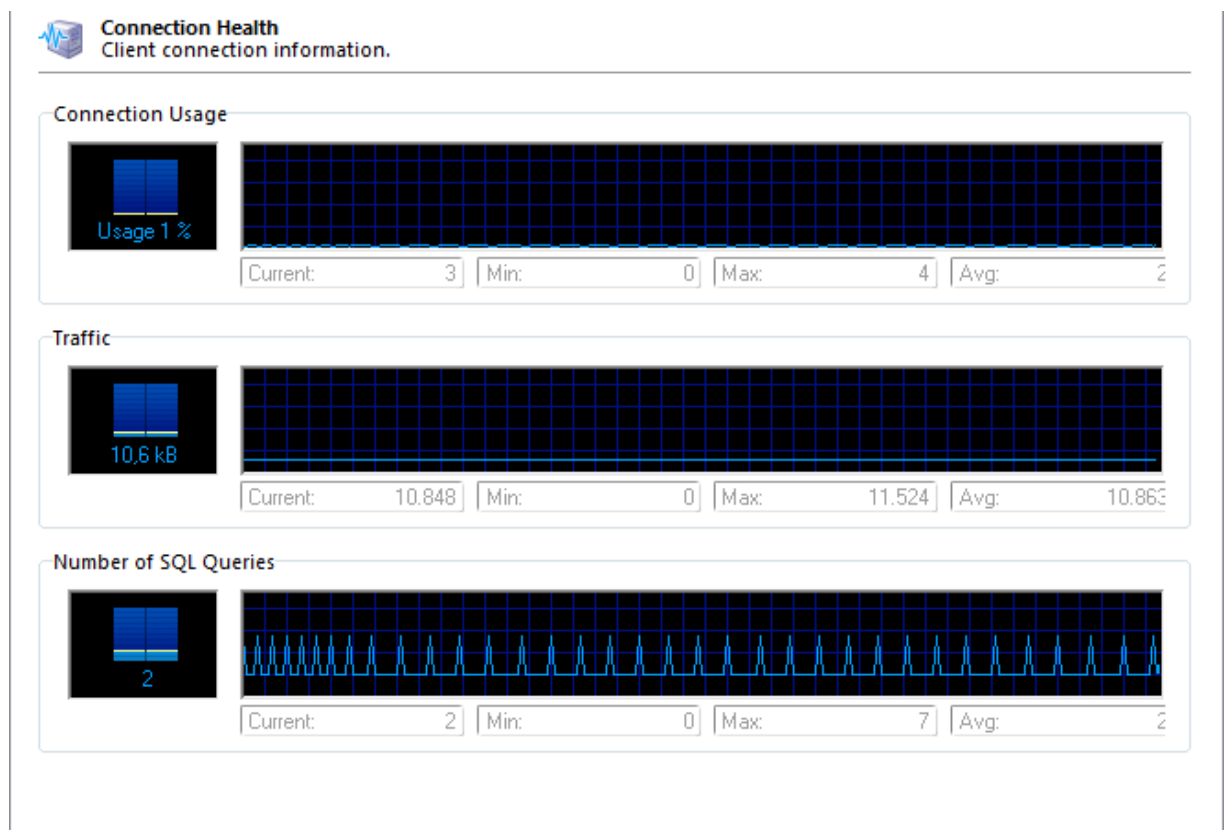


Figura 32: Posicionar mapa y actualizar posición

En último lugar, podemos apreciar en la gráfica los picos correspondientes a las operaciones de posicionamiento en el mapa y de actualizar la posición. Aparecen varios picos pues la lista de fuegos se está cargando cada pocos segundos, y para ello, se guardan todos los datos de la tabla Fire de la base de datos. Independientemente del número de fuegos que hay en la base de datos, sólo se realiza una única consulta para cargar todos. Viendo el tiempo obtenido para el posicionamiento en la tabla anterior, se debe tener en cuenta que esta operación requiere cargar la imagen completa en el terminal, lo que repercute en el tiempo de la operación.

En el caso de la actualización de la posición del usuario sobre el mapa, se produce un pequeño retardo cada vez que se quiere recalculiar la ubicación del usuario, pues además se tiene que calcular en cada momento la ruta más corta desde la posición del usuario hasta la salida más cercana. Dicho retardo se puede apreciar sobre todo en el caso de deslizar el dedo sobre la pantalla táctil para moverse por la imagen del mapa.

5. Pruebas de precisión del posicionamiento

A continuación se incluyen pruebas de precisión de posicionamiento que realizamos en la etapa final del proyecto. A pesar de que pueda parecer que el margen de error para las pruebas es alto (una media de 3 metros, con picos de hasta 10) la hemos calificado de resultado positivo. En las pruebas realizadas en [EvMa06] se obtienen 3 metros como el margen de error medio al realizar las pruebas. Nuestro escenario disponía de un número de puntos de acceso elevado para proporcionar acceso WiFi a todos los estudiantes, lo que nos facilitó la recolección de datos. Por otro lado, la lenta recolección de datos y la necesidad de programar la aplicación de posicionamiento desde 0 no nos permitió dedicar la cantidad de tiempo suficiente a un módulo que, por si sólo, sería de envergadura suficiente para considerar un ciclo de desarrollo de un año.

Medida	x	y	Tasa de error absoluta en X	Tasa de error absoluta en Y
Real	2,665	2,095	3,84	0,1
Aplicación	6,500	2,000		
Real	2,665	2,095	1,84	0,41
Aplicación	4,500	2,500		

<AVANTI> - SI 2009/2010

Real	2,665	2,095	0,17	0,1
Aplicación	2,500	2,000		
Real	3,558	5,106	0,56	2,11
Aplicación	3,000	3,000		
Real	3,558	5,106	0,94	1,61
Aplicación	4,500	3,500		
Real	3,558	5,106	0,56	1,11
Aplicación	3,000	4,000		
Real	6,711	2,638	0,21	0,14
Aplicación	6,500	2,500		
Real	6,711	2,638	4,29	0,64
Aplicación	11,000	2,000		
Real	6,711	2,638	1,79	0,64
Aplicación	8,500	2,000		
Real	10,866	0,840	1,87	1,16
Aplicación	9,000	2,000		
Real	10,866	0,840	0,87	1,16
Aplicación	10,000	2,000		
Real	10,866	0,840	1,63	1,16
Aplicación	12,500	2,000		
Real	16,843	3,030	2,16	1,03
Aplicación	19,000	2,000		
Real	16,843	3,030	4,16	1,03
Aplicación	21,000	2,000		
Real	16,843	3,030	2,84	1,53
Aplicación	14,000	1,500		
Real	28,070	1,651	1,57	0,15
Aplicación	26,500	1,500		
Real	28,070	1,651	10,07	0,35
Aplicación	18,000	2,000		
Real	28,070	1,651	7,07	0,15
Aplicación	21,000	1,500		
Real	3,558	9,929	1,06	3,93
Aplicación	2,500	6,000		
Real	3,558	9,929	1,06	3,43
Aplicación	2,500	6,500		
Real	3,558	9,929	1,06	2,93
Aplicación	2,500	7,000		
Real	30,372	1,911	1,37	0,41
Aplicación	29,000	1,500		
Real	30,372	1,911	3,37	0,41
Aplicación	27,000	1,500		

<AVANTI> - SI 2009/2010

Real	30,372	1,911	5,87	0,41
Aplicación	24,500	1,500		
Real	6,802	1,813	0,7	0,69
Aplicación	7,500	2,500		
Real	6,802	1,813	1,8	0,69
Aplicación	5,000	2,500		
Real	6,802	1,813	2,3	0,19
Aplicación	4,500	2,000		
Real	15,000	1,792	1,5	0,21
Aplicación	16,500	2,000		
Real	15,000	1,792	1	0,21
Aplicación	14,000	2,000		
Real	15,000	1,792	3,5	0,21
Aplicación	11,500	2,000		
Real	15,000	1,792	0	0,29
Aplicación	15,000	1,500		
Real	19,456	1,800	3,54	0,2
Aplicación	23,000	2,000		

Media de tasa de error en X 2,33
 Media de tasa de error en Y 0,9

Mediana de error en X 1,71
 Media de error en Y 0,52

Capítulo VII

Resultados

1. Discusión sobre los resultados obtenidos y alcanzados

La solución propuesta utilizando teléfonos móviles, coordinándose con un servidor central para poder llegar a cabo simulaciones sobre evacuaciones de incendios, traza un puente entre las ventajas de utilizar una solución virtual y llevar a cabo una evacuación real. El uso de posicionamiento WiFi permite al sistema localizar el movimiento del usuario a través de un edificio, así que se puede generar un fichero de log con todos los movimientos del usuario, automatizando llegado el caso el análisis de estos datos. Es posible también detectar cuando los usuarios intentan acceder a zonas de peligro, por lo que se les puede enviar un aviso. Además, el hecho de que el usuario se está realmente moviendo a través del edificio asegura que la velocidad de movimiento es realista, y que no habrá problemas entre la representación virtual y la posición real. Usando la realidad aumentada podemos situar una capa adicional sobre la vista de cámara incluyendo modelos virtuales de fuegos sobre la localización deseada, generando un escenario acorde a los deseos del instructor.

Después de finalizar el primer prototipo de AVANTI, la principal conclusión entre los desarrolladores es que la parte referente al WPS, y más particularmente la recolección de datos para poder montar las bases de datos necesarias para ejecutar el posicionamiento, necesita ser refinada y mejorada para poder extender la disponibilidad de este sistema al proceso comercial. Sistemas comerciales como SkyHook, como es obvio, no proporcionan el posicionamiento detallado en el interior de edificios (ya que no disponen de privilegios para acceder a todas las localizaciones donde pudiesen ejecutar la recolección de datos), y el proceso de colección manual es extremadamente tedioso. La automatización de este proceso está todavía lejos de ser una realidad, y se necesita una continua supervisión humana mientras se recolectan los datos. El crecimiento de SkyHook y su alianza con Apple para proporcionar posicionamiento WPS a sus dispositivos invita a pensar que en el futuro mejorarán su API pública y proporcionarán algún sistema más amistoso para que los usuarios puedan colaborar en expandir su base de datos (siguiendo la definición de Web 2.0 “los usuarios generan el contenido, la compañía obtiene el beneficio”¹⁸)

El uso de Android como plataforma de desarrollo fue una de las elecciones adecuadas desde el principio del desarrollo. Una enorme comunidad de desarrolladores y recursos libres en Internet hace que desarrollar aplicaciones sea un

¹⁸ <http://bash.org/?779320>. Last access 18 October 2009

proceso mucho más sencillo. En nuestra opinión, Android ha alcanzado la madurez tan sólo unos años después de haber sido publicado, y está listo para ser considerado en cualquier desarrollo profesional.

No obstante, el hardware sobre el que Android corre nos ha dejado una sensación agrídulce. La aplicación fue testada en una HTC Hero y una HTC Magic, que en ocasiones no tienen una compatibilidad 100% con el sistema operativo Android. Tuvimos bastantes problemas intentando combinar la vista de cámara y la vista de OpenGL, problema que no ocurría en otros teléfonos con Android y que está registrado como un bug oficial en la comunidad de desarrolladores de Android. La última versión de Android, que solucionaba algunos bugs, no estuvo disponible para la HTC Hero hasta pocos meses antes de la finalización del proyecto, por lo que no tuvimos tiempo de considerar el efectuar la portabilidad de versión dado el poco tiempo de desarrollo con el que contábamos. No obstante, el Nexus One de Google fue lanzado algún tiempo antes de finalizar el proyecto y las valoraciones en general son muy positivas, por lo que esperamos que el hardware disponible para Android sea realmente competitivo, y la experiencia de uso pueda ayudar a impulsar Android a pelear en el mismo terreno que otros “teléfonos inteligentes”, como el iPhone de Apple.

También es importante señalar que el hardware acelerómetros, en nuestra opinión, está bastante inmaduro en lo que respecta a dispositivos móviles de uso general. Debido al alto coste de sensores de aceleración con alta precisión, los sensores que montan los dispositivos HTC Hero y Magic tienen bastante margen de error. Este error se ve incluso aumentado cuando el terminal se acerca a algún campo magnético, cómo pueden ser ordenadores o cuadros de luz. Estas fluctuaciones se reflejan en el prototipo generando constantes temblores en la posición de los elementos dibujados como realidad aumentada, y dando una sensación de movimiento bastante molesta. Tal y como dijimos en el párrafo anterior, esperamos que se abaraten los costes de estos sensores y en medio plazo se disponga de una precisión mayor en los dispositivos de propósito general.

Respecto a la realidad aumentada, hay algunos hechos que todavía deberían tenerse en cuenta a la hora de desarrollar aplicaciones profesionales, como se explica en [Azum97]: todavía no está disponible para la ejecución en tiempo real de aplicaciones, y su uso en dispositivos móviles esta todavía en fase de desarrollo. Bajo estas circunstancias, nuestro proyecto trata de proponer una prueba de concepto, que

pueda desarrollar y proponer un modelo para resolver una problemática determinada, y no actuar como un producto con un ciclo de desarrollo completo. En cualquier caso, es de esperar mejoras sobre el terreno conforme la tecnología en este campo vaya avanzando.

Otra cuestión importante es el efecto de la realidad aumentada incluida en el prototipo como una ayuda a la motivación de los usuarios de la aplicación. El uso de simulaciones para mejorar la motivación, como se ha explicado en capítulos anteriores, es un procedimiento aceptado en contextos educativos [Dill00]. Marc Prensky, en su libro “Digital Game-Based Learning” [Prens01] nos cuenta algunas historias interesantes donde aplicó juegos y simulaciones para ayudar a ciertos colectivos a involucrarse dentro de un proceso de aprendizaje. La Realidad Aumentada se ha usado también con propósitos similares [Kauf03].

1.1 Resultados positivos

Listamos a continuación los resultados de esta memoria que interpretamos como positivos, dando razones para justificarlo y nuestras interpretaciones.

- Las técnicas de posicionamiento han probado ser utilizables en este contexto. [EvMa06] y [WLSB07] indican que obtuvieron en sus pruebas un margen de error de hasta 3 metros. En el capítulo de “Tests”, apartado 7, proporcionamos nuestras pruebas de campo, donde el margen de error para el posicionamiento queda por debajo de 3 (aunque con una desviación típica superior a la esperada). No obstante, perfeccionar o definir nuevas técnicas de posicionamiento WiFi no era el objetivo del presente trabajo, sino mostrar que era factible su implementación y uso en terceras aplicaciones
- La realidad aumentada es viable con la versión de Android con la que desarrollamos la aplicación (1.5). No obstante, versiones posteriores de Android han incluido funciones en el API que sistematizan el desarrollo de este tipo de aplicaciones, y lo facilitan en gran medida. Entendemos que podría considerarse incluso más sencillo desarrollar aplicaciones de realidad aumentada en la actualidad que cuando nosotros trabajamos con ello.
- Desarrollar un modelo que combinase la información de la brújula, del

acelerómetro, del posicionamiento WiFi y fuese capaz de representarlo utilizando realidad aumentada, aunque una tarea complicada, es posible. Es necesario definir con precisión extrema la aportación de cada uno de los elementos al global del proyecto, y aunque la calidad de los sensores no sea todo lo buena que cabría esperar, proporcionan un buen punto de entrada.

1.2. Resultados negativos

A continuación se muestran los puntos que consideramos no han resultado positivos o han limitado en gran medida el desarrollo:

- Aunque el posicionamiento WiFi funciona, como se ha indicado en el punto previo, no sería el indicado para una aplicación con un alto grado crítico. El límite de 3 metros al posicionamiento puede resultar un inconveniente en determinados escenarios.
- El pintado de objetos OpenGL sobre Android es extremadamente limitado en cuanto a tamaño, texturas y vértices. Es de esperar que conforme el hardware vaya mejorando y abaratando mayores opciones estén disponibles para la plataforma, pero en el presente Android no debería ser considerado para resultados más complejos (i.e, un juego)
- Para realizar el posicionamiento de manera certera es necesario analizar continuamente información ambiental (puntos de acceso en el área de visión del dispositivo) o bien realizar el cálculo sobre el propio terminal, con el consiguiente gasto en recursos. Finalmente la opción que implementamos fue la segunda, ya que la primera resultó inviable: el envío continuo de información desde el terminal hacia el servidor con la frecuencia requerida para que el cálculo de la posición fuese efectivo actuaba como un ataque de “ping” sobre la máquina servidor, que experimentaba cuelgues y lentitud extrema en la transmisión de datos.

2. Descripción de alternativas a las empleadas

Realmente, desde el principio tuvimos pocas alternativas entre las que escoger. La cuestión principal fue si usar como terminal un dispositivo Apple o Android. Como se argumenta en las conclusiones, la elección del segundo fue un acierto por la convergencia de multitud de factores. No obstante, entre las alternativas podemos citar teléfonos de la marca Sony Ericsson o Nokia. Los primeros fueron descartados porque no veíamos que la documentación disponible o el apoyo de la comunidad de desarrolladores fuese lo suficientemente extenso. En cuanto a los segundos, tuvimos la sensación de que el SDK no era tan maduro como el que ofrecían Apple o Android, a pesar del tiempo que lleva en el mercado este producto.

Para la parte del servidor existían diversas alternativas de desarrollo dinámico, pero la elección de uno o de otro no habría afectado en lo esencial al curso del proyecto, tratándose más de un aspecto puramente técnico. GWT fue elegido por disponer de más experiencia con él. Sabíamos de la existencia de otros SDK's, tales como ASP.net AJAX¹⁹ y jQuery²⁰, con el que también se tenía experiencia.

Como motor de bases de datos usamos MySQL, que viene siendo el gestor de bases de datos más utilizado para desarrollos de código libre. Existían otras alternativas, como Oracle²¹, que no utilizamos principalmente por ser un producto de pago y no poder permitírnos la licencia. En cualquier caso, dada la naturaleza de nuestro proyecto y el controlado número de transacciones que se iban a efectuar, no era necesaria una alternativa más potente que MySQL.

3. Análisis de la complejidad final

Desde el inicio determinamos que la complejidad de nuestro proyecto sería media-alta. El desarrollo de aplicaciones con gráficos 3D siempre es costoso en términos de esfuerzo y tiempo. De manera adicional, insertar un componente tal como el posicionamiento WiFi que requería un trabajo tan minucioso al no poder reutilizar

¹⁹ <http://ajax.asp.net/>

²⁰ <http://jquery.com/>

²¹ <http://www.oracle.com>

ninguna librería y tener que generar la base de datos geográfica a mano era un factor a tener en cuenta, a la que había que sumar nuestra inexperiencia en el mismo campo.

La siguiente tabla muestra información cuantitativa sobre el tamaño que ha alcanzado el desarrollo. Tanto para la aplicación del servidor como para la del cliente se muestra información sobre el número de clases que la componen, así como el número de líneas de código. No es la técnica métrica que proporciona información más fidedigna, pero representa una aproximación. Hay que tener en cuenta que es más difícil cuantificar otro tipo de esfuerzos, como el esfuerzo de diseño para la base de datos o para generar una apariencia gráfica para AVANTI.

Unit	Nominal Lines	Source Code Lines	Comment Lines	Blank Lines	Mixed Lines	Classes
Aplicación cliente	11136	8844	462	1802	29	80
Aplicación servidor	2237	1640	70	521	6	30
Total	13373	10484	532	2323	27	110

4. Trabajo futuro

La ejecución de evacuaciones en protocolos de incendio, coordinadas con un servidor central y haciendo uso de técnicas de WPS y Realidad Aumentada es una posibilidad viable. Este modelo combina las ventajas que proporciona las simulaciones virtuales con aquellas provenientes de escenarios de fuego reales, pero evitamos los peligrosos inconvenientes de tener que huir de un fuego real. El próximo paso en AVANTI sería poder extender la aplicación para poder reflejar que existen distintos tipos de fuegos, con diferentes restricciones, y que cada uno de ellos ha de ser atajado de diferente manera. Los fuegos se podrían representar adaptando un sistema de partículas para reflejar las distintas características de los mismos (virulencia, altura, etc.). Otra idea que surgió durante el desarrollo de la aplicación era añadir un sistema donde los extintores de incendio fueran relevantes, y que se pudieran usar para poder extinguir los fuegos. Esto requeriría una etapa previa de investigación para analizar exactamente las características de cada tipo de fuego distinto, y cómo se combaten (por ejemplo, el agua no debería utilizarse contra un incendio que ha sido provocado por un fallo eléctrico). Esto se podría extender todavía más colocando sensores en

los extintores. Si somos capaces de localizar estos elementos durante la simulación, y saber si han sido usados o no, podríamos proporcionar un nuevo y atractivo nivel de realismo a la aplicación, consiguiendo mayor realismo y mejorando la experiencia de aprendizaje (al hacer que aquel que está practicando la situación de escape sea parte activa en la simulación, y no tenga que limitarse a escapar del edificio).

Debido al coste de generar el mapa completo de señales para todo el edificio, utilizamos la representación tan sólo para la segunda planta. Una futura versión podría incluir un modelado de todo el edificio. Esto podría requerir que los algoritmos sean mejorados y adaptados para poder incluir sectores complejos y problemáticos, tales como las escaleras.

Una idea inicial fue modelar el servidor utilizando O3D, aunque debido al coste de aprendizaje de la plataforma y de desarrollo con la misma fue descartado. No obstante, es una buena idea utilizar O3D para resolver algunos de los problemas que se han indicado en el punto 2.2.1 sobre el tema de costes en las aplicaciones educativas. Al mismo tiempo que obtenemos una simulación en 3D que puede mejorar la motivación de los usuarios, desarrollaríamos una aplicación web, con todas las ventajas inherentes que esto proporciona relativo al coste. En caso de que no fuese posible, utilizar algún motor de escritorio como jMonkey o Java3D sería una opción a priori también atractiva.

Capítulo VIII

Apéndice

1. Estadísticas del proyecto

La siguiente tabla muestra el número de líneas de código por fichero. Previamente, en el apartado de complejidad final (apartado 5 del capítulo VI) se mostraban organizados por paquetes). Se ha usado Practiline Source Code Counter²² para generar la tabla. La primera tabla muestra el resultado para la aplicación móvil. La segunda muestra el resultado para la aplicación del servidor.

File Name	Nominal Lines	Source Code Lines	Source Code Lines (%)	Comment Lines	Comment Lines (%)	Blank Lines	Blank Lines (%)	Mixed Lines	Mixed Lines (%)	Total Lines
com.avanti.activities.About	345	249	72.17 %	49	14.20 %	47	13.62 %	0	0.00 %	345
com.avanti.activities.EvacuationProtocol	33	21	63.64 %	2	6.06 %	10	30.30 %	0	0.00 %	33
com.avanti.activities.GLSurfaceViewActivity	262	179	68.32 %	20	7.63 %	62	23.66 %	1	0.38 %	262
com.avanti.activities.Login	252	210	83.33 %	5	1.98 %	37	14.68 %	0	0.00 %	252
com.avanti.activities.Main	73	56	76.71 %	1	1.37 %	15	20.55 %	1	1.37 %	73
com.avanti.activities.MainMenu	239	189	79.08 %	4	1.67 %	45	18.83 %	1	0.42 %	239
com.avanti.activities.MovementPrediction	5	3	60.00 %	0	0.00 %	2	40.00 %	0	0.00 %	5
com.avanti.activities.Position	102	76	74.51 %	4	3.92 %	22	21.57 %	0	0.00 %	102
com.avanti.activities.PositionActivity	9	5	55.56 %	0	0.00 %	4	44.44 %	0	0.00 %	9
com.avanti.activities.Register	102	86	84.31 %	0	0.00 %	16	15.69 %	0	0.00 %	102
com.avanti.activities.ScanResultActivity	42	29	69.05 %	0	0.00 %	13	30.95 %	0	0.00 %	42
com.avanti.activities.TestAccelerometer	272	198	72.79 %	17	6.25 %	53	19.49 %	4	1.47 %	272
com.avanti.activities.WPSActivity	221	192	86.88 %	0	0.00 %	29	13.12 %	0	0.00 %	221
com.avanti.camera.CustomCameraView	52	38	73.08 %	1	1.92 %	13	25.00 %	0	0.00 %	52
com.avanti.camera.DrawOnTop	79	49	62.03 %	8	10.13 %	22	27.85 %	0	0.00 %	79
com.avanti.camera.Preview	77	52	67.53 %	9	11.69 %	16	20.78 %	0	0.00 %	77
com.avanti.dialogs.DialogController	78	65	83.33 %	0	0.00 %	13	16.67 %	0	0.00 %	78
com.avanti.GEO.Compass	194	155	79.90 %	18	9.28 %	21	10.82 %	0	0.00 %	194
com.avanti.GEO.CompassRenderer	102	75	73.53 %	4	3.92 %	23	22.55 %	0	0.00 %	102
com.avanti.graphics.ColladaHandler	95	73	76.84 %	0	0.00 %	22	23.16 %	0	0.00 %	95
com.avanti.graphics.ColladaLoader	90	68	75.56 %	1	1.11 %	21	23.33 %	0	0.00 %	90
com.avanti.graphics.ColladaObject	65	46	70.77 %	5	7.69 %	14	21.54 %	0	0.00 %	65
com.avanti.graphics.Cube	84	66	78.57 %	7	8.33 %	11	13.10 %	0	0.00 %	84
com.avanti.graphics.CubeRenderer	135	98	72.59 %	7	5.19 %	30	22.22 %	0	0.00 %	135
com.avanti.graphics.CVertex	24	5	20.83 %	0	0.00 %	12	50.00 %	7	29.17 %	24
com.avanti.graphics.Eye	108	83	76.85 %	0	0.00 %	25	23.15 %	0	0.00 %	108
com.avanti.graphics.FlameRenderer	123	82	66.67 %	16	13.01 %	25	20.33 %	0	0.00 %	123
com.avanti.graphics.GameView	430	265	61.63 %	109	25.35 %	56	13.02 %	0	0.00 %	430
com.avanti.graphics.Mesh	416	367	88.22 %	1	0.24 %	48	11.54 %	0	0.00 %	416
com.avanti.graphics.MeshLoader	169	146	86.39 %	0	0.00 %	23	13.61 %	0	0.00 %	169
com.avanti.graphics.ParticleAnimation	50	35	70.00 %	0	0.00 %	15	30.00 %	0	0.00 %	50

²² <http://sourcecount.com/>

<AVANTI> - SI 2009/2010

com.avanti.graphics.ParticleRokon	56	40	71.43 %	0	0.00 %	16	28.57 %	0	0.00 %	56
com.avanti.graphics.Texture	204	172	84.31 %	0	0.00 %	32	15.69 %	0	0.00 %	204
com.avanti.graphics.ViewListener	31	28	90.32 %	0	0.00 %	3	9.68 %	0	0.00 %	31
com.avanti.http.HttpServices	209	191	91.39 %	0	0.00 %	17	8.13 %	1	0.48 %	209
com.avanti.http.HttpServicesException	14	10	71.43 %	0	0.00 %	4	28.57 %	0	0.00 %	14
com.avanti.menu.Menu	12	7	58.33 %	0	0.00 %	5	41.67 %	0	0.00 %	12
com.avanti.position.Mapa	417	335	80.34 %	13	3.12 %	69	16.55 %	0	0.00 %	417
com.avanti.routes.Arista	91	77	84.62 %	0	0.00 %	14	15.38 %	0	0.00 %	91
com.avanti.routes.Cuadrante	102	84	82.35 %	0	0.00 %	18	17.65 %	0	0.00 %	102
com.avanti.routes.Fire	103	78	75.73 %	1	0.97 %	24	23.30 %	0	0.00 %	103
com.avanti.routes.FireList	94	79	84.04 %	1	1.06 %	14	14.89 %	0	0.00 %	94
com.avanti.routes.ListaCuadrantes	507	446	87.97 %	11	2.17 %	50	9.86 %	0	0.00 %	507
com.avanti.routes.Posicion	56	45	80.36 %	0	0.00 %	11	19.64 %	0	0.00 %	56
com.avanti.sensor.AccelerometerReader	209	187	89.47 %	2	0.96 %	20	9.57 %	0	0.00 %	209
com.avanti.sensor.SensorManagerAVANTI	58	41	70.69 %	1	1.72 %	16	27.59 %	0	0.00 %	58
com.avanti.service.PositionTracker	70	44	62.86 %	7	10.00 %	19	27.14 %	0	0.00 %	70
com.avanti.sockets.Client	37	30	81.08 %	1	2.70 %	6	16.22 %	0	0.00 %	37
com.avanti.sockets.Server	36	30	83.33 %	1	2.78 %	5	13.89 %	0	0.00 %	36
com.avanti.test.TestHttp	50	36	72.00 %	0	0.00 %	14	28.00 %	0	0.00 %	50
com.avanti.test.TestListaCuadrantes	51	40	78.43 %	1	1.96 %	10	19.61 %	0	0.00 %	51
com.avanti.tests.ColladaHandler	101	82	81.19 %	1	0.99 %	18	17.82 %	0	0.00 %	101
com.avanti.tests.ColladaLoader	92	70	76.09 %	0	0.00 %	22	23.91 %	0	0.00 %	92
com.avanti.tests.ColladaObject	40	29	72.50 %	0	0.00 %	10	25.00 %	1	2.50 %	40
com.avanti.tests.Cube	40	26	65.00 %	7	17.50 %	7	17.50 %	0	0.00 %	40
com.avanti.tests.GLColor	35	30	85.71 %	0	0.00 %	5	14.29 %	0	0.00 %	35
com.avanti.tests.GLFace	78	56	71.79 %	5	6.41 %	17	21.79 %	0	0.00 %	78
com.avanti.tests.GLShape	86	67	77.91 %	2	2.33 %	16	18.60 %	1	1.16 %	86
com.avanti.tests.GLVertex	73	61	83.56 %	1	1.37 %	10	13.70 %	1	1.37 %	73
com.avanti.tests.GLWorld	82	65	79.27 %	0	0.00 %	17	20.73 %	0	0.00 %	82
com.avanti.tests.KubeRenderer	99	77	77.78 %	1	1.01 %	21	21.21 %	0	0.00 %	99
com.avanti.tests.Layer	85	69	81.18 %	6	7.06 %	10	11.76 %	0	0.00 %	85
com.avanti.tests.M4	64	55	85.94 %	0	0.00 %	9	14.06 %	0	0.00 %	64
com.avanti.tests.MD2Loader	406	358	88.18 %	13	3.20 %	35	8.62 %	0	0.00 %	406
com.avanti.tests.ModelView	252	185	73.41 %	13	5.16 %	54	21.43 %	0	0.00 %	252
com.avanti.tests.ModelViewer	51	38	74.51 %	5	9.80 %	8	15.69 %	0	0.00 %	51
com.avanti.tests.ModelViewInterpolated	297	224	75.42 %	12	4.04 %	60	20.20 %	1	0.34 %	297
com.avanti.tests.ViewAnimator	51	43	84.31 %	0	0.00 %	8	15.69 %	0	0.00 %	51
com.avanti.wps.Coordenada	106	87	82.08 %	1	0.94 %	18	16.98 %	0	0.00 %	106
com.avanti.wps.Mapa	37	25	67.57 %	0	0.00 %	12	32.43 %	0	0.00 %	37
com.avanti.wps.Particle	111	89	80.18 %	0	0.00 %	22	19.82 %	0	0.00 %	111
com.avanti.wps.ParticleFilter	335	293	87.46 %	3	0.90 %	39	11.64 %	0	0.00 %	335
com.avanti.wps.ThreadDatos	76	60	78.95 %	0	0.00 %	16	21.05 %	0	0.00 %	76
com.avanti.wps.ThreadFuegos	445	370	83.15 %	12	2.70 %	54	12.13 %	9	2.02 %	445
com.avanti.wps.ThreadUbicacion	319	258	80.88 %	14	4.39 %	47	14.73 %	0	0.00 %	319
com.avanti.wps.WPS	31	21	67.74 %	0	0.00 %	10	32.26 %	0	0.00 %	31
com.avanti.wps.WPSApiTest	530	462	87.17 %	25	4.72 %	43	8.11 %	0	0.00 %	530
com.avanti.wps.WPSDatabase	315	273	86.67 %	8	2.54 %	33	10.48 %	1	0.32 %	315
com.avanti.wps.WPSException	14	10	71.43 %	0	0.00 %	4	28.57 %	0	0.00 %	14
com.example.helloandroid.HelloAndroid	150	100	66.67 %	6	4.00 %	44	29.33 %	0	0.00 %	150
Total:	11136	8844	79.42 %	462	4.15 %	1801	16.17 %	29	0.26 %	11136

File Name	Nominal Lines	Source Code Lines	Source Code Lines (%)	Comment Lines	Comment Lines (%)	Blank Lines	Blank Lines (%)	Mixed Lines	Mixed Lines (%)	Total Lines
com.avanti.web.AVANTI_web.gwt.xml	18	7	38.89 %	8	44.44 %	3	16.67 %	0	0.00 %	18
com.avanti.web.client.AVANTI_web	207	149	71.98 %	11	5.31 %	47	22.71 %	0	0.00 %	207
com.avanti.web.client.fire.FireImage	169	124	73.37 %	5	2.96 %	40	23.67 %	0	0.00 %	169
com.avanti.web.client.fire.FireInformationPanel	140	100	71.43 %	4	2.86 %	36	25.71 %	0	0.00 %	140
com.avanti.web.client.fire.FireObject	57	42	73.68 %	0	0.00 %	15	26.32 %	0	0.00 %	57
com.avanti.web.client.fire.FireService	13	10	76.92 %	0	0.00 %	3	23.08 %	0	0.00 %	13
com.avanti.web.client.fire.FireServiceAsync	12	8	66.67 %	0	0.00 %	4	33.33 %	0	0.00 %	12
com.avanti.web.client.GreetingService	12	10	83.33 %	0	0.00 %	2	16.67 %	0	0.00 %	12
com.avanti.web.client.GreetingServiceAsync	10	8	80.00 %	0	0.00 %	2	20.00 %	0	0.00 %	10
com.avanti.web.client.login.LoginService	9	7	77.78 %	0	0.00 %	2	22.22 %	0	0.00 %	9
com.avanti.web.client.login.LoginServiceAsync	8	5	62.50 %	0	0.00 %	3	37.50 %	0	0.00 %	8
com.avanti.web.client.map.MapaService	13	10	76.92 %	0	0.00 %	3	23.08 %	0	0.00 %	13
com.avanti.web.client.map.MapaServiceAsync	10	8	80.00 %	0	0.00 %	2	20.00 %	0	0.00 %	10
com.avanti.web.client.MapWindow	182	113	62.09 %	8	4.40 %	61	33.52 %	0	0.00 %	182
com.avanti.web.client.Player.PlayerImage	309	223	72.17 %	10	3.24 %	76	24.60 %	0	0.00 %	309
com.avanti.web.client.Player.PlayerInformationPanel	61	44	72.13 %	0	0.00 %	17	27.87 %	0	0.00 %	61
com.avanti.web.client.Player.PlayerObject	115	90	78.26 %	0	0.00 %	25	21.74 %	0	0.00 %	115
com.avanti.web.client.Player.PlayerService	11	8	72.73 %	0	0.00 %	3	27.27 %	0	0.00 %	11
com.avanti.web.client.Player.PlayerServiceAsync	11	6	54.55 %	0	0.00 %	5	45.45 %	0	0.00 %	11
com.avanti.web.daos.DAOFire	132	99	75.00 %	5	3.79 %	28	21.21 %	0	0.00 %	132
com.avanti.web.daos.DAOManager	84	57	67.86 %	3	3.57 %	18	21.43 %	6	7.14 %	84
com.avanti.web.daos.DAOPlayer	54	42	77.78 %	1	1.85 %	11	20.37 %	0	0.00 %	54
com.avanti.web.server.FireServiceImpl	48	31	64.58 %	0	0.00 %	17	35.42 %	0	0.00 %	48
com.avanti.web.server.GreetingServiceImpl	19	16	84.21 %	0	0.00 %	3	15.79 %	0	0.00 %	19
com.avanti.web.server.LoginServiceImpl	14	10	71.43 %	1	7.14 %	3	21.43 %	0	0.00 %	14
com.avanti.web.server.MapaServiceImpl	46	28	60.87 %	1	2.17 %	17	36.96 %	0	0.00 %	46
com.avanti.web.server.PlayerServiceImpl	22	17	77.27 %	0	0.00 %	5	22.73 %	0	0.00 %	22
com.avanti.web.test.JOGLTest	62	52	83.87 %	0	0.00 %	10	16.13 %	0	0.00 %	62
com.avanti.web.test.SQLTest	29	18	62.07 %	1	3.45 %	10	34.48 %	0	0.00 %	29
com.avanti.web.test.TestBits	77	64	83.12 %	3	3.90 %	10	12.99 %	0	0.00 %	77
com.avanti.web.util.TextConstants	283	234	82.69 %	9	3.18 %	40	14.13 %	0	0.00 %	283
Total:	2237	1640	73.31 %	70	3.13 %	521	23.29 %	6	0.27 %	2237

Capítulo IX

Bibliografía y referencias

1 Bibliografía y referencias

- [Azum97] R. Azuma, A Survey of Augmented Reality Presence: Teleoperators and Virtual Environments, pp. 355–385, August 1997.
- [BaPa00] P. Bahl and V. N. Padmanabhan, “RADAR: an in-building RF-based user location and tracking system,” in Proceedings of 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00), vol. 2, pp. 775–784, Tel Aviv, Israel, March 2000.
- [Berg58] Berge, C. Theorie des Graphes et ses Applications. Dunod, Paris (1958).
- [BMMV06] Balasubramanian, V., Massaguer, D., Mehrotra, S., Venkatasubramanian, N.: Drillsim: A simulation framework for emergency response drills. In: Intelligence and Security Informatics (ISI 2006). (Mayo 2006)
- [Cast96] Castells, Manuel (1996, second edition, 2000). The Rise of the Network Society, The Information Age: Economy, Society and Culture Vol. I. Cambridge, MA; Oxford, UK: Blackwell. ISBN 978-0631221401.
- [Char89] N. Charette. Software Engineering Risks Analysis and Management. McGraw Hill. Inc. Year of Publication: (1989).
- [ChKo02] Y. Chen and H. Kobayashi, “Signal strength based indoor geolocation,” in Proceedings of the IEEE International Conference on Communications (ICC '02), vol. 1, pp. 436–439, New York, NY, USA, April-May 2002.
- [ChRa09] Chittaro, L. Ranon, R. Serious games for training occupants of a building in personal fire safety skills. 2009 Conference in Games and Virtual Worlds for Serious Applications, IEEE
- [Derry07] A. Derryberry. Serious Games: online games for learning. I'm Serious.net. (2009)
- [Dill00] Dillenbourg, P.,” Virtual Learning Environments”. Proc. EUN Conference. 2000, Workshop on Virtual Environments, 2000.
- [EvMa06] Frederic Evennou and Francois Marx, “Advanced Integration of WiFi and Inertial Navigation Systems for Indoor Mobile Positioning”. Hindawi Publishing Corporation, EURASIP Journal on Applied Signal Processing, Volume 2006, Article ID 86706, Pages 1–11
- [Hans03] Hansmann, Uwe (2003). Pervasive Computing: The Mobile World. Springer. ISBN 3540002189.
- [John10] Johnson C W. Using computer simulations to support a risk-based approach for hospital evacuation. Technical Report. Glasgow Accident Analysis Group, University of Glasgow, 2006.
- [Kauf03] Kaufmann, H., Collaborative Augmented Reality in education. Proc. Imagina 2003 Conf (Imagina03) Monaco, 2003.
- [KiMc04] J. Kirriemur, A. McFarlane.: Literature review in games and learning: NESTA Futurelab. Series. NESTA Futurelab., Bristol (2004)

- [Mann02] Mann., Steve. Mediated Reality with implementations for everyday life, 2002 August 6th, Presence Connect, the on line companion to the MIT Press journal PRESENCE: Teleoperators and Virtual Environments, MIT Press
- [MiKi94] P. Milgram and A. F. Kishino, Taxonomy of Mixed Reality Visual Displays IEICE Transactions on Information and Systems, E77-D(12), pp. 1321-1329, 1994.
- [Posl09] Poslad, Stefan (2009). Ubiquitous Computing Smart Devices, Smart Environments and Smart Interaction. Wiley. ISBN 978-0-470-03560-3.
<http://www.elec.qmul.ac.uk/people/stefan/ubicom/index.html>.
- [Press05] Pressman, R.S. (2005). Ingeniería del Software. McGraw-Hill Interamericana
- [PWCG02] Pair, J., Wilson, J., Chastine, J., Gandy, M. "The Duran Duran Project: The Augmented Reality Toolkit in Live Performance". The First IEEE International Augmented Reality Toolkit Workshop, 2002.
- [Rayb05] E.M. Raybourn: Adaptive thinking and leadership training for cultural awareness and communication competence. Esmerald Group Publishing Limited (2005)
- [RCSZ06] Aizhu Ren, Chi Chen, Jianyong Shi, Liang Zou: Application Of Virtual Reality Technology To Evacuation Simulation In Fire Disaster. Proceedings of the 2006 International Conference on Computer Graphics & Virtual Reality, CGVR 2006, Las Vegas, Nevada, USA
- [Reev83] Reeves W. T., Particle Systems – A Technique for Modeling a Class of Fuzzy Objects, Computer Graphics 17:3 pp, 359-376, 1983
- [RPTMS02] Teemu Roos, Petri Myllymäki, Henry Tirri, Pauli Misikangas, and Juha Sievänen. "A Probabilistic Approach to WLAN User Location Estimation". International Journal of Wireless Information Networks, Vol. 9, No. 3, July 2002
- [SPSH97] J.Seitz, L. Patiño-Studencka, B. Schindler, S. Haimerl, J. Gutiérrez Boronat, S. Meyer, J. Thielecke. "Sensor Data Fusion for Pedestrian Navigation Using WLAN and INS"
- [SrPo07] Srevens, P., Pooley, R. (2007). Utilización de UML en ingeniería del Software con objetos y componentes. Addison Wesley
- [Thom92] R. Thomsett: "The Indiana Jones School of Risk Management," American Programmer, vol. 5, no. 7, September, pp. 10–18 (1992).
- [Weis91] Weiser, Mark (1991). "The Computer for the 21st Century".
<http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>. Última visualización: 10 - Mayo – 2010.
- [Weis96] Weiser, Mark (1996-03-17). "Ubiquitous computing".
<http://www.ubiq.com/hypertext/weiser/UbiHome.html>. Retrieved 2007-11-03.
- [Weis99] Weiser; Gold; Brown (1999-05-11). "Ubiquitous computing".
<http://www.research.ibm.com/journal/sj/384/weiser.html>. Retrieved 2008-05-07.
- [WHC07] "World Haptics Conferences". Haptics Technical Committee.
<http://www.worldhaptics.org/hapticConferences.htm>. Última visualización: 10 - Mayo – 2010.
- [WLSB07] Hui Wang, Henning Lenz, Andrei Szabo, Joachim Bamberger, Uwe D. Hanebeck,

- ”WLAN-Based Pedestrian Tracking Using Particle Filters and Low-Cost MEMS Sensors”. 4th Workshop on positioning, navigation and communication. 2007 (WPNC’07), Hannover, Germany.
- [Zyda05] M. Zyda. From Visual Simulation to Virtual Reality Games. IEEE Computer Society. (2005)

“Quien no puede vencerse a sí mismo ha nacido para ser esclavo”