



Universidad
Carlos III de Madrid

Escuela Politécnica Superior

Departamento de Informática

PROYECTO FIN DE CARRERA

**Sistema de localización de dispositivos
portátiles a través de WiFi:**

LAPLOWIX

Autor: Francisco José Sánchez Esteban

Tutor: Julián Moreno Schneider

Directora: Jesica Rivero Espinosa

Titulación: Ingeniería Técnica en Informática de Gestión

Leganés, junio de 2012

Título: "Sistema de localización de dispositivos portátiles a través de WiFi: LAPLOWIX"

Autor: Francisco José Sánchez Esteban

Tutor: Julián Moreno Schneider

Directora: Jesica Rivero Espinosa

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto fin de Carrera en Leganés el jueves 28 de junio de 2012, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la calificación de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Agradezco el apoyo que me han brindado tanto a la hora de hacer este proyecto fin de carrera como en todas las decisiones que he tomado en mi vida, a mis padres *Paco* y *MariCarmen*, sin los que con su ayuda y constantes “¿Cómo llevas el proyecto?” y apoyo no sólo en el proyecto si no en multitud de situaciones esto no habría sido posible. También esto va dedicado a *Natalia*, mi hermana, sigue esforzándote y conseguirás tus metas. También agradecer al resto de mi familia tíos, primos y abuelos.

Me gustaría también agradecer el apoyo de mis amigos, no los nombro pero ellos saben bien quienes son, ya que también ellos han sido un pilar fundamental de este edificio.

Agradezco el apoyo mostrado a la hora de hacer este proyecto a las tres personas de parte de la universidad que me han apoyado y orientado en su elaboración. Gracias a *David*, *Jesica* y *Julián*.

¡Gracias a todos!

Resumen

En este proyecto fin de carrera se ha realizado un sistema de localización para equipos portátiles capaz de identificar la ubicación del equipo en un punto. Estos puntos pueden ser o bien lugares concretos dentro de salas o edificios o también ubicaciones físicas al aire libre siempre y cuando haya algún sistema inalámbrico mediante el cual realizar la localización

Con este objetivo, se ha realizado un estudio previo del estado actual de las tecnologías necesarias para la composición de la aplicación. A partir de los conocimientos adquiridos, se ha diseñado y desarrollado un software (programa informático) que cumple con los requisitos establecidos en el planteamiento inicial.

Finalmente, se ha implementado el sistema en un dispositivo concreto y se ha evaluado detalladamente su funcionamiento para la comprobación del resultado obtenido. El sistema que se ha creado para la evaluación de esta forma de localización se ha implementado para ordenadores portátiles con sistema operativo Linux. El desarrollo se ha llevado a cabo en Java, apoyándose en MySQL como gestor de base de datos. Para realizar la localización del dispositivo se ha optado por la utilización de las redes Wifi ampliamente extendidas hoy en día.

Los resultados obtenidos durante las pruebas han demostrado que el sistema sirve para localizar dispositivos, aunque hay que tener en cuenta que el sistema de localización mediante la señal Wifi no es todo lo preciso que debiera y se ve altamente influenciado por el entorno.

Índice de Contenidos

1. Introducción	2
1.1 <i>Motivación del Proyecto</i>	2
1.2 <i>Ámbito del Sistema</i>	3
1.3 <i>Objetivos</i>	3
1.4 <i>Estructura del Documento</i>	4
2. Estado del Arte	7
2.1 <i>Localización</i>	7
2.2 <i>Tipos de sistemas de localización</i>	8
2.2.1 Tecnología de localización basada en radiofrecuencia (RFID)	8
2.2.2 Tecnología de localización basada en GPS	9
2.2.3 Tecnología de localización basada en Bluetooth	10
2.2.4 Tecnología de localización basada en Zigbee	11
2.2.5 Tecnología de localización basada en Wimax	12
2.2.6 Tecnología de localización basada en Wifi	12
3. Gestión del Proyecto	16
3.1 <i>Descomposición en Tareas</i>	16
3.2 <i>Temporalización del Proyecto</i>	17
3.3 <i>Diagrama de Gantt</i>	18
3.4 <i>Presupuesto</i>	19
3.4.1 Costes de Personal	19
3.4.2 Costes de los equipos	20
3.4.3 Costes de subcontratación y otros costes directos	20
3.4.4 Resumen de costes	20
4. Fases de la aplicación	22
4.1 <i>Diseño</i>	23
4.2 <i>Implementación</i>	23
4.3 <i>Evaluación</i>	24
5. Diseño	26
5.1 <i>Diseño de la base de datos</i>	26
5.2 <i>Diseño de la interfaz gráfica</i>	28
5.2.1 Interfaz de “Tomar medidas”	30
5.2.2 Interfaz de “Realizar modelo”	31
5.2.3 Interfaz de “Detectar ubicación”	32
5.3 <i>Diseño de la lógica de la aplicación</i>	33
5.3.1 Lógica de obtención de la medición	34

5.3.2	Lógica del almacenamiento y recuperación de mediciones	34
5.3.3	Lógica para la realización del modelo de datos	35
5.3.4	Lógica para la detección de la posición	35
6.	Implementación	38
6.1	<i>Implementación de la base de datos</i>	38
6.2	<i>Implementación de la interfaz gráfica</i>	41
6.2.1	Ventana de bienvenida	42
6.2.2	Ventana del menú principal	43
6.2.3	Ventana de realizar experimentos	44
6.2.4	Ventana de modelado	46
6.2.5	Ventana Resultado	47
6.3	<i>Implementación de la lógica de la aplicación</i>	49
6.3.1	Lógica de la obtención de medidas	49
6.3.2	Lógica de modelado tipo media	51
6.3.3	Lógica de la localización	51
7.	Evaluación	55
7.1	<i>Evaluación de la interfaz gráfica</i>	55
7.2	<i>Evaluación de la lógica de la aplicación</i>	56
7.2.1	Prueba 1	56
7.2.2	Prueba 2	58
8.	Conclusiones	62
9.	Futuras líneas de trabajo	64
	Referencias	66
	Bibliografía	68
	ANEXO I: Diagramas de clase	70
	ANEXO II: Manual de Usuario	73
	ANEXO III: Diagrama de Gantt	78

Introducción

1. Introducción

Este capítulo ofrecerá una breve descripción del problema que se desea abordar y presentará de manera superficial la solución elegida para subsanarlo. En primer lugar, se introduce el proyecto propiamente dicho a través de su motivación, su ámbito y características y sus principales objetivos. Posteriormente, se expone la manera en que ha sido estructurado este documento enumerando cada uno de los capítulos e incluyendo una breve descripción de los mismos.

1.1 Motivación del Proyecto

A día de hoy la localización o el posicionamiento de todo tipo de dispositivos se han convertido en una especie de obsesión. Se localizan todo tipo de objetos, desde coches a barcos pasando por elementos tan diferentes a estos como son los teléfonos móviles o las cámaras de fotos.

Es posible que toda esta “obsesión” por la localización de los dispositivos electrónicos de consumo esté justificada en algunos casos, como por ejemplo en una empresa en la que todos sus empleados dispongan de un móvil y un ordenador portátil. Es bastante posible que a esta empresa le interesase saber en todo momento dónde están situados estos equipos durante el horario laboral por si por ejemplo se tiene que ir a visitar a un empleado y se quiere indicar al visitante la posición en la que se encuentra el empleado. ¿Cómo se podría llevar a cabo esta localización? ¿Es posible hacer esto?

Casi siempre que se habla de localizar algún dispositivo, la primera forma de posicionamiento en la que se piensa es el sistema de posicionamiento GPS (Global Positioning System: sistema de posicionamiento global). Este es un muy buen sistema de localización que permite identificar la posición de un objeto en el globo terráqueo con un margen de error relativamente pequeño (casi siempre menor de 6 metros).

El sistema de localización mediante GPS es un buen método de posicionamiento, pero tiene un impedimento y es que para que realmente el sistema trabaje y sea lo más efectivo posible, el objeto que se quiere posicionar debe estar preferiblemente al aire

libre, ya que incluso estando al aire libre existen puntos negros en los cuales no funciona.

Por otra parte este sistema permite obtener la posición de un objeto tomando como referencia la Tierra, pero ¿qué hay si se quiere tomar otro sistema de referencia? ¿Se podría localizar un dispositivo dentro de un edificio tomando este como referencia? ¿Se podría averiguar la posición de ese mismo dispositivo dentro de un complejo empresarial con multitud de edificios?

Se tratará de aprovechar una tecnología ya existente y ampliamente extendida en multitud de edificios, ya sean viviendas o parques empresariales, para intentar realizar un software que basándose en las infraestructuras que ya existen sirva para localizar un dispositivo portátil sin necesidad de usar la localización GPS. Por lo tanto, ¿será posible usar la tecnología Wifi o radio para este propósito?

1.2 Ámbito del Sistema

Con las limitaciones de localización que impone el sistema GPS (que actualmente se encuentra más extendido) se va a proceder a diseñar y desarrollar un software que en combinación con elementos hardware ya existentes permita realizar el posicionamiento de un dispositivo dentro de un edificio o de un complejo de edificios.

Se tratará de ofrecer al usuario un sistema de localización que se pueda adaptar a cualquier edificio y a cualquier ámbito siempre que se cumplan los requisitos preestablecidos (las características mínimas para poder localizar el dispositivo en dicho escenario).

1.3 Objetivos

Una vez vistos la motivación y el ámbito de aplicación del sistema, se establecerán los objetivos que deberán ser alcanzados para considerar el proyecto como completado. Son cuatro los objetivos que principalmente se han considerado:

1. Se debe diseñar e implementar un software capaz de tomar mediciones de las señales Wifi circundantes de modo que sirva para caracterizar la posición de un

equipo (ordenador de sobremesa, portátil, móvil o tablet) que disponga de esta tecnología. punto de la zona a mapear.

2. Se debe diseñar e implementar un sistema de almacenamiento para que puedan alojar en él las mediciones que se tomen.
3. Con las mediciones que se realicen a lo largo del tiempo, se crearán modelo de datos que posteriormente se utilizarán para tratar de inferir la posición del equipo y así localizarlo.
4. Se permitirá la inferencia de la posición de un dispositivo en función de los modelos previamente definidos y creados.

1.4 Estructura del Documento

El resto del documento está estructurado en los siguientes capítulos:

- **Estado del Arte:** Presentación de los dispositivos y las tecnologías que han sido utilizadas en el proyecto. Se analiza su estado actual, tanto sus características como sus aplicaciones.
- **Gestión del Proyecto:** En este capítulo se muestra la metodología empleada para llevar a cabo la realización del proyecto. En él se exponen las tareas en las que se ha dividido así como la duración de cada una y el consiguiente gasto.
- **Fases del Proyecto:** Breve descripción de las fases por las que ha pasado el proyecto así como del modelo de desarrollo utilizado.
- **Diseño:** Descripción de cómo se quiere que se realicen los desarrollos para la implementación de un prototipo que cumpla los requisitos expuestos anteriormente.
- **Implementación:** Descripción de cómo se ha desarrollado el prototipo diseñado en el apartado anterior.
- **Evaluación:** En este capítulo se observan las distintas pruebas y test realizadas al prototipo con la intención de comprobar su funcionalidad.
- **Conclusiones:** Presentación de las conclusiones obtenidas como consecuencia del desarrollo de este proyecto.

- **Futuras Líneas de Trabajo:** Resumen de las principales ampliaciones posibles para llevar a cabo como continuación o alternativa a este proyecto.
- **Anexos:** Como anexos se incluyen el manual de instalación de la aplicación desarrollada y un manual de usuario para facilitar su uso.

Estado del Arte

2. Estado del Arte

En este capítulo se detalla el análisis inicial realizado acerca del estado en que se encuentran en la actualidad las tecnologías utilizadas en este proyecto y se mencionan algunos trabajos realizados en la misma línea que la nuestra. Este estudio ha sido desarrollado previamente al diseño de la aplicación y ha servido de guía introductoria y referencia para las siguientes fases del proyecto.

La metodología empleada ha consistido en una primera etapa de investigación puramente teórica en la que se han identificado términos, agrupado conceptos y relacionado temas.

Posteriormente, el estudio se ha centrado en una perspectiva más práctica. Se ha sondeado el mercado para enumerar los dispositivos existentes en la actualidad. Además, se han buscado, estudiado y evaluado las aplicaciones en las que se implementan dichas tecnologías.

La documentación de esta sección ha sido dividida en base a los diferentes conceptos sobre los que se ha realizado el estudio debido a que su uso es un requisito indispensable en este proyecto.

2.1 Localización

Para intentar describir un sistema que nos permita la localización de un dispositivo mediante algún tipo de estructura ya existente, se debe primero analizar el término *localización*.

Como tal *localización* es la acción o hecho de localizar y *localizar* se define como la determinación o el emplazamiento que debe tener alguien o algo.

Una vez aclarado este concepto se puede decir como resumen de los objetivos, que con este proyecto se busca poder determinar el emplazamiento o posición de un dispositivo en una sala o edificio aprovechando tecnologías existentes, donde tecnologías actuales como el GPS no son del todo efectivas.

2.2 Tipos de sistemas de localización

Actualmente existen varios tipos de tecnologías que sirven para la localización de elementos, pero principalmente:

- Tecnología de localización basada en radiofrecuencia (RFID).
- Tecnología de localización basada en GPS.
- Tecnología de localización basada en Bluetooth.
- Tecnología de localización basada en Zigbee.
- Tecnología de localización basada en Wimax.
- Tecnología de localización basada en Wifi.

2.2.1 Tecnología de localización basada en radiofrecuencia (RFID)

El sistema de identificación por radiofrecuencia es el RFID. Este sistema se basa en adherir en los productos a localizar/identificar unas etiquetas, dotadas de una serie de datos y en un lector/emisor de ondas de radio.

El funcionamiento de este sistema es sencillo: el emisor/lector emite unas señales de radio que las antenas que se encuentran en las etiquetas se encargan de interpretar y reenvían al lector/emisor que recibe la información que está almacenada en las etiquetas.

Lo que hace que este sistema de localización/identificación esté comenzando a ganar terreno al todopoderoso código de barras, hoy por hoy el sistema más extendido, es que el sistema RFID no necesita tener visión directa como si lo necesita el sistema de códigos de barra.

Los sensores RFID se basan en la tecnología del mismo nombre [1]. Estas siglas representan el término inglés *Radio Frequency IDentification*, cuyo significado en español es identificación por radiofrecuencia. Las etiquetas RFID son unos dispositivos de reducido tamaño que se adhieren a un objeto. Contienen antenas para permitirles recibir y

responder a peticiones por radiofrecuencia desde un emisor-receptor RFID. Al emplear radiofrecuencia se evita la necesidad de visión directa entre emisor y receptor para su detección.

En la actualidad, las aplicaciones más conocidas de estos sensores son por ejemplo el seguimiento de libros en las bibliotecas, el control de acceso en edificios, la identificación de acreditaciones, el seguimiento de equipajes en aerolíneas, el seguimiento de artículos de ropa o la apertura de vehículos de alta gama.

2.2.2 Tecnología de localización basada en GPS

Las siglas **GPS** se corresponden con "*Global Positioning System*" que significa **Sistema de Posicionamiento Global** [2] (aunque sus siglas GPS han popularizado el producto en el mundo comercial).

En síntesis se puede definir el GPS como un Sistema Global de Navegación por Satélite (GNSS) que permite fijar a escala mundial la posición de un objeto, una persona, un vehículo o una nave.

La precisión del GPS puede llegar a determinar los puntos de posición con errores mínimos de centímetros (GPS diferencial), aunque en la práctica se hable de metros.

Los orígenes de este sistema hay que situarlos en el ámbito de la Defensa de los Estados Unidos de América. El Departamento de Defensa fue el que desarrolló e instaló, y opera actualmente este sistema. Para ello, una red de 24 satélites (21 operativos) en órbita a 20.200 km permite cubrir toda la superficie terrestre.

Para fijar una posición, un dispositivo equipado con GPS localiza automáticamente como mínimo 3 satélites de la red, de los que recibe unas señales indicando la posición y el reloj de cada satélite. El dispositivo GPS sincroniza su reloj y calcula el retraso de las señales (que viene dado por distancia al satélite), calculando la posición en que éste se halla.

Estimadas las distancias, se fija con facilidad la propia posición relativa del GPS respecto a los tres satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o coordenadas reales del punto de medición.

Los GPS inundan el mercado para los usuarios con fines muy diversos; senderismo; montañismo; hasta incluso se ha puesto de moda en los campos de Golf. No obstante, la utilización actual más extendida es su empleo en los vehículos que circulan por carreteras (coches, camiones, autobuses...).

GPS para el coche

Este uso permite a los conductores un apoyo muy útil a la conducción, especialmente en ciudades o rutas con las que no están familiarizados. Los navegadores con GPS [3] llevan programas con voz que le dan instrucciones al conductor sobre los movimientos que deben hacer para seguir la ruta correcta (giros, toma de salidas o entradas desde unas vías a otras, etc.); estas indicaciones de voz, permiten al conductor fijar su atención en la carretera. En el caso de existir un copiloto, este puede ver, en todo momento, en la pantalla del GPS, el movimiento continuo identificado en un mapa del coche o vehículo, indicando en nombre de las calles, vías, etc.

2.2.3 Tecnología de localización basada en Bluetooth

La tecnología Bluetooth surge tras la implementación del estándar industrial IEEE 802.15.1 por parte de compañías como Nokia, Toshiba, Ericsson, IBM e Intel. Su principal objetivo es la eliminación de cableado y la simplificación en la sincronización de datos entre dispositivos informáticos, ya sean estos fijos o portátiles.

Las ondas que emiten los elementos emisores de Bluetooth tienen un alcance de apenas 10 metros. Para localizaciones en espacios pequeños esto puede ser una ventaja ya que este factor unido al bajo coste de la tecnología en principio lo haría un buen candidato de estudio para un prototipo de localización dentro de una habitación por ejemplo.

Existen varias tecnologías que se apoyan en el Bluetooth para su funcionamiento:

- Blueps: usado para localización.
- Wibree: usado en relojes o teclados inalámbricos.

Pese a que estas tecnologías aparentemente podrían resultar útiles para intentar llevar a cabo un prototipo de localización, se descartan debido a que pese a su bajo coste, el radio de acción no es demasiado grande, unos 10 metros, y no es una tecnología tan implantada en edificios como para que no haya lagunas dentro de los mismos.

2.2.4 Tecnología de localización basada en Zigbee

Es una norma de comunicaciones inalámbricas diseñada por la *Zigbee Alliance*. Se basa en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (WPAN, *Wireless Personal Area Network*) y tiene como objetivo las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

Zigbee permite construir redes en malla que cubran diferentes superficies. Con este tipo de redes, tanto los mensajes como los componentes son sencillos, de manera tal que cada nodo con funcionalidad completa puede actuar como una estación de retransmisión.

El bajo precio de los componentes, permite que cada nodo sea instalado dentro del rango del alcance medio (unos 30 metros) de otro, y los mensajes, simplemente, podrían ir de uno a otro, hasta llegar a su destino, sin necesidad de costosos concentradores o cableado.

Como contrapartida, pese a su alcance medio, éste sigue siendo un alcance menor al Wifi ya que pertenece a redes inalámbricas de tipo WPAN.

Comentar también que esta tecnología sirvió como base para la realización de otros estudios y pruebas como por ejemplo el proyecto fin de carrera "*Estudio de diferentes sistemas de posicionamiento y de su aplicación con Zigbee*" de María González [6]

2.2.5 Tecnología de localización basada en Wimax

Wimax es una tecnología de comunicación de banda ancha, al igual que Wifi, basada en la familia de estándares IEEE 802.16

Destaca por su capacidad como tecnología portadora, lo que la hace perfectamente adecuada para entornos de grandes redes corporativas de voz y datos así como para operadores de telecomunicaciones.

La tecnología Wimax tiene una serie de ventajas que la hacen bastante atractiva como son su relativo bajo coste de implantación, el alcance ideal de hasta 50 Km y la relativa buena velocidad de transmisión (hasta 75 Mbps).

A día de hoy esta tecnología se suele implantar en zonas urbanas en las que no es posible que el par de cobre usado por las líneas telefónicas clásicas llegue, como por ejemplo en ciertas pedanías de la localidad alicantina de Elche. Pese a todo esto esta tecnología no está tan extendida como otras para poder realizar un buen trabajo de localización.

2.2.6 Tecnología de localización basada en Wifi

Es un conjunto de estándares para redes inalámbricas basados en las especificaciones IEEE 802.11. Fue creado para ser utilizado en redes locales inalámbricas, ya que es un sistema de envío de datos sobre redes computacionales que utiliza ondas de radio en lugar de cables. Su nombre comercial es *WirelessFidelity*(Fidelidad inalámbrica).

Se presentan diferentes estándares. Los estándares IEEE 802.11b e IEEE 802.11g son aceptados de forma internacional ya que utilizan la banda de frecuencia de 2,4GHz y presentan una velocidad de hasta 11Mbps y 54Mbps respectivamente. Otro de sus estándares es el IEEE 802.11a que opera en la banda de 5GHz.

Para conseguir una red inalámbrica sólo se necesita un punto de acceso (AP), que se conectaría a la red, y un dispositivo WIFI que se conectaría en nuestro aparato. Estos

APs se aconseja colocarlos en lugares altos para obtener una recepción o una transmisión más fluida y así no impedir que la velocidad sea la adecuada debido a obstáculos que se puedan encontrar en su recorrido.

Como aplicación de la tecnología Wifi al mundo de la localización tenemos el llamado Wifi-Mapping.

Wifi-Mapping

Es un sistema de localización basado en las redes Wifi. Este sistema se caracteriza por calcular la posición mediante la comparación de la señal rssi que se obtiene de cada uno de los puntos de acceso, identificados por una dirección mac[4], que el sistema que se quiere localizar compara con una base de datos que puede estar en remoto o no.

Veamos cuales son las ventajas que han llevado a que sea este el sistema de localización elegido:

1. Es un sistema de localización que sirve para interiores y exteriores (siempre que estos posean conexión Wifi), al contrario que el GPS que únicamente es válido para exteriores.
2. La otra gran ventaja que tiene este sistema de localización es que no necesita de nueva infraestructura, ya que hoy en día son muchos los lugares que están provistos de redes Wifi y sus correspondientes puntos de acceso.

Pero este sistema también tiene un par de desventajas, que aunque sean pequeñas deben ser nombradas:

1. Para que el sistema funcione en una zona determinada, antes deben haberse llevado a cabo mediciones de todos los puntos de dicha zona para que exista un modelo de referencia en la base de datos para realizar la comparación y triangulación de la posición.

2. A la hora de realizar las mediciones se ha observado que los sistemas tienen una latencia a la hora de realizar nuevas medidas de los puntos de acceso y la potencia de éstos que reciben. Esta latencia es de unos 60 segundos, dependiendo del sistema operativo del dispositivo que se quiera localizar.

También cabe mencionar que esta tecnología para la localización de dispositivos en interiores también está siendo investigada por la compañía finlandesa Nokia para incluirla en futuros terminales [5].

Gestión del Proyecto

3. Gestión del Proyecto

En este apartado se describe la gestión del proyecto presentado en este documento. Se realiza una estimación del esfuerzo del proyecto. Dicha estimación se realiza antes del comienzo de las fases que lo componen y se utiliza para valorar la viabilidad del proyecto a realizar.

3.1 Descomposición en Tareas

Para la realización de dicho proyecto se contemplan las siguientes tareas:

- Estudio preliminar:
 - ✚ Estado de la cuestión: donde se obtendrá una visión preliminar del proyecto que podría llegar a realizarse.
 - ✚ Planteamiento del problema: evaluando los sistemas de localización actuales se observa que la localización en edificios puede ser complicada con los métodos existentes en la actualidad.
 - ✚ Esbozo de la solución: se plantea una posible solución a grandes rasgos.
- Desarrollo:
 - ✚ Definición de requisitos: se definen los requisitos que se quiere que tenga el sistema de localización a crear.
 - ✚ Especificación de los requisitos: se realiza un análisis en profundidad de los requisitos definidos.
 - ✚ Diseño: se estudian las distintas maneras de abordar la solución.
 - ✚ Programación de la solución: una vez definido el diseño y la arquitectura sobre la que se va a desarrollar se procede a la implementación de la solución elegida.
 - ✚ Pruebas: en esta fase se verificará si la solución que se ha elegido cumple con los requisitos que se habían establecido o no.

- Memoria y documentación: donde se engloban todas las tareas relacionadas con la documentación del proyecto como el análisis, desarrollo, implementación y pruebas relacionadas, así como manuales de uso e instalación.

3.2 Temporalización del Proyecto

A continuación se muestran las tareas y fechas planificadas para el desarrollo del proyecto. Este plan de trabajo posee la columna real que es la que se ha utilizado para la elaboración del diagrama de Gantt que se incluye en los anexos.

Tareas	Planificado		Re-Planificado		Real		Tiempo dedicado	Observaciones
	Desde	Hasta	Desde	Hasta	Desde	Hasta		
Estudio preliminar								
Estado de la cuestión	5-sep-2011	23-sep-2011			5-sep-2011	23-sep-2011	1,5 h/día	Revisión de tecnologías métodos y trabajos existentes.
Planteamiento del problema	26-sep-2011	27-sep-2011			26-sep-2011	27-sep-2011	2 h/día	Breve descripción del problema.
Esbozo de la solución	3-oct-2011	10-oct-2011	10-oct-2011	17-oct-2011	10-oct-2011	17-oct-2011	2 h/día	Breve descripción de la solución.
Desarrollo								
Definición de requisitos	24-oct-2011	28-oct-2011			24-oct-2011	28-oct-2011	3 h/día	Definición de los requisitos del sistema.
Especificación de requisitos	31-oct-2011	4-nov-2011	2-nov-2011	7-nov-2011	2-nov-2011	7-nov-2011	3 h/día	Análisis de los requisitos y definición del modelo conceptual.
Diseño	8-nov-2011	18-nov-2011			8-nov-2011	18-nov-2011	2 h/día	Diseño del sistema: arquitectura.
Programación de la solución	21-nov-2011	31-dic-2011			21-nov-2011	31-dic-2011	4 h/día	Programación de la solución.
Pruebas	9-ene-2012	13-ene-2012			9-ene-2012	13-ene-2012	1 h/día	Fase de pruebas.
Memoria								
Introducción	21-nov-2011	25-nov-2011			21-nov-2011	25-nov-2011	1,5 h/día	Introducción al proyecto: Objetivos.
Estado del arte	28-nov-2011	12-dic-2011			28-nov-2011	12-dic-2011	1,5 h/día	Redacción del trabajo realizado en el estudio preliminar.
Gestión del proyecto	13-dic-2011	16-dic-2011			13-dic-2011	16-dic-2011	1 h/día	Organización y reparto de esfuerzo durante el proyecto.
Fases del proyecto	19-dic-2011	16-ene-2012	26-dic-2011	23-ene-2012	26-dic-2011	23-ene-2012	4 h/día	Explicación de la solución desarrollada y pruebas.
Conclusiones	24-ene-2012	24-ene-2012			24-ene-2012	24-ene-2012	2 h/día	Conclusiones personales sobre la solución desarrollada.
Futuras líneas de trabajo	25-ene-2012	31-ene-2012			25-ene-2012	31-ene-2012	1,5 h/día	Futuras líneas de investigación relacionadas con el proyecto.
Anexos	1-feb-2012	3-feb-2012			1-feb-2012	3-feb-2012	3 h/día	Documentos que constan como anexos del proyecto.
Presentación								
Preparación diapositivas	5-mar-2012	9-mar-2012	5-mar-2012	13-mar-2012	5-mar-2012	13-mar-2012	1,5 h/día	
Preparación discurso	5-mar-2012	9-mar-2012	5-mar-2012	13-mar-2012	5-mar-2012	13-mar-2012	1,5 h/día	

Figura 3.2 – Plan de trabajo

3.3 Diagrama de Gantt

El diagrama Gantt completo asociado a las tareas descritas, se incluye en el capítulo de [Anexos](#) de esta memoria. Este diagrama Gantt recoge la duración real del proyecto. Las tareas asociadas se corresponden con las descritas en la figura 3.2 y se engloban en cuatro grupos:

1. Estudio preliminar que se compone de las subtarear:
 - a. Estado de la cuestión
 - b. Planteamiento del problema
 - c. Esbozo de la solución
2. Desarrollo que se compone de las subtarear:
 - a. Definición de requisitos
 - b. Especificación de requisitos
 - c. Diseño
 - d. Programación de la solución
 - e. Pruebas
3. Memoria que se compone de las subtarear:
 - a. Introducción
 - b. Estado del arte
 - c. Gestión del proyecto
 - d. Fases del proyecto
 - e. Conclusiones
 - f. Futuras líneas de trabajo
 - g. Anexos
4. Presentación que se compone de las subtarear:
 - a. Preparación de diapositivas
 - b. Preparación de discurso

A continuación se muestra una versión simplificada del diagrama de Gantt, en el cuál se pueden observar la temporalización del proyecto y el orden de las tareas.

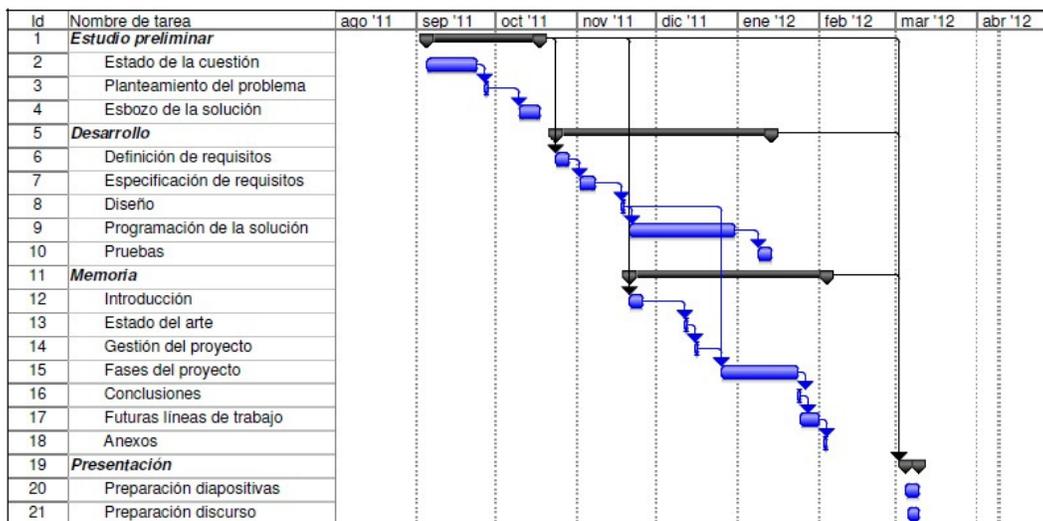


Figura 3.3 – Diagrama de Gantt

3.4 Presupuesto

3.4.1 Costes de Personal

Los costes que se describen a continuación son los derivados de las horas dedicadas por el personal que ha elaborado el proyecto en tareas de gestión, diseño, programación y evaluación.

FUNCIÓN	HORAS	PRECIO/HORA	SUBTOTAL
<i>Diseño</i>	147,25	28,71 €	4.227,55 €
<i>Gestión</i>	25,55	41,01 €	1.047,81 €
<i>Programación</i>	176,7	16,41 €	2.899,65 €
<i>Pruebas</i>	10	24,61 €	246,10 €
		TOTAL	8.421,10 €

Figura 3.4.1 - Costes de personal

Como se puede ver en la tabla anterior el mayor coste de personal se lo llevan el diseño y la implementación/programación de la solución con un 85% del presupuesto, mientras que la gestión conlleva un 12% y las pruebas un exiguo aunque no por ello despreciable 3%.

3.4.2 Costes de los equipos

A continuación se describen los costes presupuestados a raíz de los equipos hardware y software usados para la realización del proyecto.

DESCRIPCION	COSTE ANUAL	USO AL PROYECTO	DEDICACION	DEPRECIACION	COSTE IMPUTABLE
Ordenador Portátil	550 €	100	8 meses	12 meses	366,67 €
Licencia Ubuntu	0 €	100	8 meses	12 meses	- €
Licencia MySQL	0 €	100	8 meses	12 meses	- €
Licencia Java	0 €	100	8 meses	12 meses	- €
				TOTAL	366,67 €

Figura 3.4.2 – Coste de los equipos

3.4.3 Costes de subcontratación y otros costes directos

Dado que no se han subcontratado tareas a otras empresas y que no ha habido otro tipo de gastos a imputar en el proyecto como son dietas o viajes, no se ha producido ningún gasto por tanto el gasto concurrido por este apartado ha sido de 0€.

3.4.4 Resumen de costes

Teniendo en cuenta los gastos incurridos en este presupuesto para el desarrollo del proyecto y sumando todos ellos el resumen es el siguiente:

Personal	8.421,10 €
Equipos	366,67 €
Subcontratación	- €
Otros	- €
TOTAL	8.787,77 €

Figura 3.4.4 – Resumen de costes

Fases de la aplicación

4. Fases de la aplicación

Este capítulo presentará las diferentes fases en las que se ha dividido el desarrollo global del proyecto.

El hecho de que se observen tres grandes capítulos como son diseño, implementación y evaluación, no quiere decir que éstos se hayan desarrollado en ese orden específico ya que el modelo de ciclo de vida de software elegido para la construcción del prototipo ha sido un modelo de *desarrollo en espiral*.

El modelo de desarrollo en espiral fue definido en el año 1986 por Barry Boehm. En este modelo de desarrollo de software las tareas se representan en una espiral en la que cada iteración representa un conjunto de todas ellas.

Básicamente consiste en una serie de ciclos que se repiten en forma de espiral, comenzando desde el centro. Se suele interpretar como que dentro de cada ciclo de la espiral se sigue un Modelo Cascada, pero no necesariamente debe ser así. El Espiral puede verse como un modelo evolutivo que conjuga la naturaleza iterativa del Modelo de Prototipos con los aspectos controlados y sistemáticos del Modelo Cascada, con el agregado de gestión de riesgos.

En cada iteración las tareas que se realizarán serán de cuatro tipos:

- Análisis de requisitos.
- Diseño del sistema.
- Etapas de construcción.
- Test y evaluación.

Se puede ver de forma gráfica cómo se representaría el modelo de desarrollo en espiral en la siguiente figura:

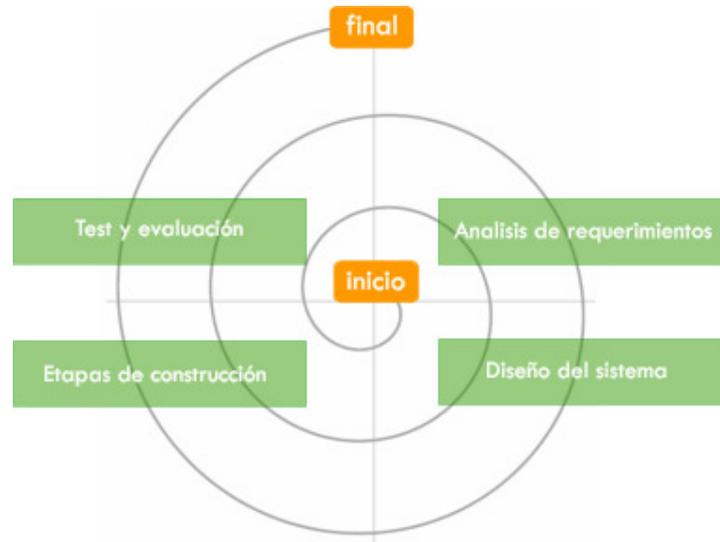


Figura 4.1 – Modelo de Desarrollo en Espiral

4.1 Diseño

En el apartado de Diseño del sistema se van a establecer qué es lo que se quiere que se lleve a cabo en la implementación y cómo en teoría debería de ser ese desarrollo.

También se dividirá el trabajo de desarrollo intentando diseñar módulos que sean independientes entre sí y que puedan posteriormente acoplarse sin problemas entre ellos o con otros que se desarrollen en un futuro.

4.2 Implementación

Si en la fase de diseño se trazan las líneas sobre qué hay que desarrollar, en esta fase se explicará cómo se han desarrollado realmente las ideas diseñadas y por qué se han desarrollado así y no de otra forma, viendo las alternativas que hay y las opciones escogidas.

Esta fase sin duda será de carácter mucho más técnico que la fase de diseño pues en ella se tendrán en cuenta detalles sobre qué lenguaje de programación escoger o qué gestor de base de datos se usará.

4.3 Evaluación

La última fase de cada una de las iteraciones del ciclo de vida/desarrollo en espiral es la evaluación.

En esta fase se validará que cada una de las ideas o módulos que se diseñaron y posteriormente se implementaron. Se evaluará si cumplen con la función para la que fueron creados y satisfacen así de forma correcta alguno de los requisitos iniciales que se establecieron.

Diseño

5. Diseño

En este apartado se va a definir el *qué* hacer para obtener un prototipo más o menos funcional y que cumpla con los requisitos definidos en el [capítulo 1](#).

El prototipo de la aplicación estará compuesto por 3 grandes elementos que se pueden diseñar por separado y posteriormente acoplar de forma que la aplicación funcione como un todo.

Este prototipo recibirá como nombre *Laplowix*, ya que es el acrónimo de *LA*ptop*LO*ca-tion*W*ifi al cual se le añade una *X* final debido a que la implementación será llevada a cabo para un sistema Unix

Los apartados en los que se va a dividir el diseño de la aplicación son:

- Base de datos.
- Interfaz gráfica.
- Lógica de la aplicación (localización).

5.1 *Diseño de la base de datos*

Con la realización de este prototipo se quiere poder identificar cualquier punto dentro de un edificio, por tanto la primera entidad que se debe definir es un *Punto* como lugar de partida del diseño de la base de datos. El punto llevará asociado como atributos las coordenadas espaciales “x y z”, así como una descripción que indique con palabras más coloquiales cuál es el punto.

Una vez ya se tiene claro que la primera entidad será Punto, estos Puntos se pueden agrupar en conjuntos dentro de un *Escenario*. El Escenario se puede identificar con una habitación en concreto o también con un conjunto de puntos que en un momento dado tenían unas condiciones concretas comunes como puede ser por ejemplo que estén en la misma planta de un edificio o que tengan una ventana abierta. Con esto se daría la posibilidad de poder tener definido un mismo punto pero con varias condiciones “ambientales” distintas lo que a priori daría una mejor representación del mundo real

en la base de datos. Los atributos que tendrán los Escenarios serán un id y una descripción.

Por otra parte, una vez están definidos los Puntos y los Escenarios habrá que identificar los *Puntos de Acceso (Access Point)* a los cuales se conectará la aplicación usando la tecnología WiFi. Los Puntos de Acceso se identificarán mediante su dirección mac que es única.

Con las 3 entidades definidas anteriormente ya se puede definir la que en principio será una de las entidades básicas del esquema de base de datos y que es la *Medición*. Una Medición vendrá identificada por un Punto, un Punto de Acceso y la fecha en la que fue tomada, junto con la potencia rssi que se obtiene de la medición realizada.

Para poder dar sentido a la fecha de la medición se creará la entidad *Experimento* que realmente será un conjunto de mediciones realizadas con unos puntos de accesos determinados en un punto concreto de un escenario y en una hora concreta.

Con estas premisas el esquema diseñado para la base de datos sería:

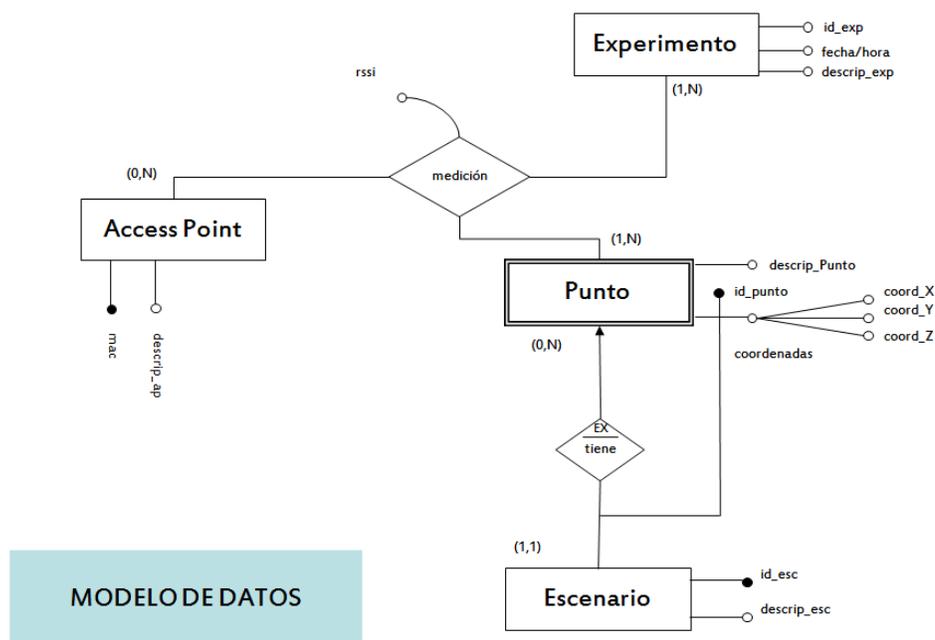


Figura 5.1.1 – Esquema E/R de la base de datos

5.2 Diseño de la interfaz gráfica

A la hora de diseñar la interfaz gráfica de la aplicación lo primero que se tiene que tener en cuenta son los posibles casos de uso que se van a desarrollar posteriormente. En este aspecto este proyecto va a desarrollar tres casos de uso para un usuario final y teniendo en cuenta éstos así se desarrollará la interfaz.

Los casos de uso serían “Tomar medidas”, “Realizar modelo” y “Detectar ubicación”. Estos casos se pueden observar representados en siguiente diagrama:

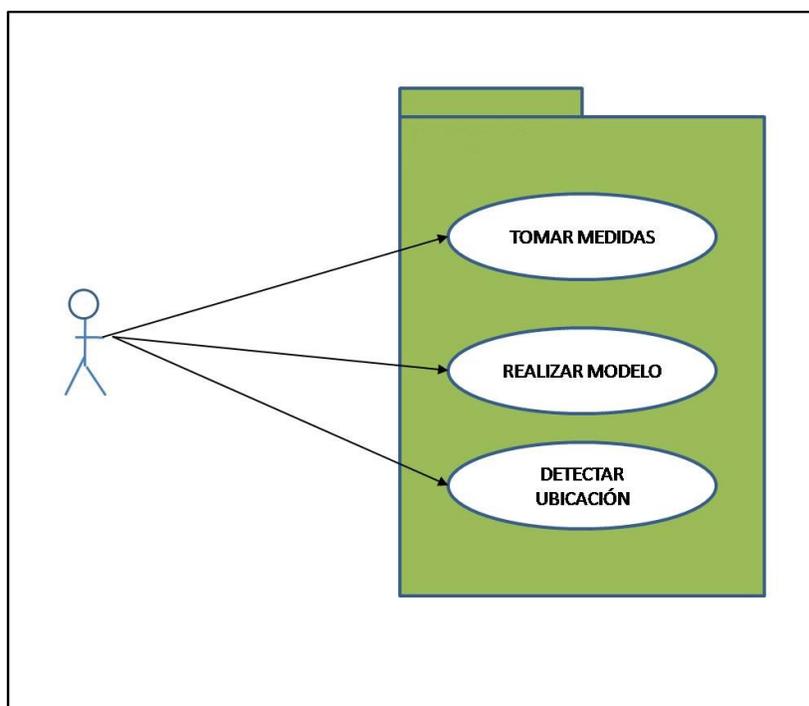


Figura 5.2.1 – Diagrama de casos de uso

Antes de pasar a diseñar y especificar cómo será el diseño de la interfaz para cada uno de los casos de uso que se ven en la figura 5.2.1 se deben desarrollar dos ventanas iniciales. Estas ventanas iniciales serán una ventana de bienvenida a la aplicación que deberá tener un diseño como el que se puede ver en la figura 5.2.2 y una ventana de

menú principal, figura 5.2.3, desde la cual se puedan iniciar los 3 casos de uso de los que se compondrá el prototipo Laplowix.

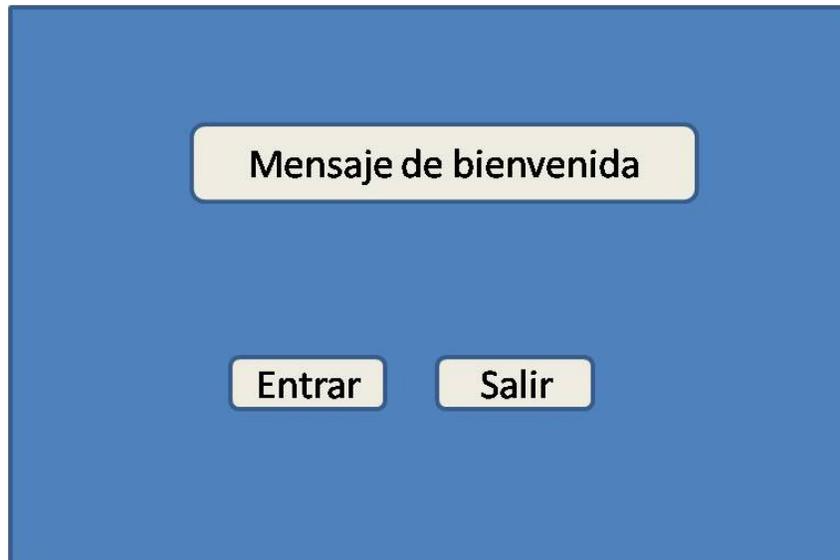


Figura 5.2.2 – Ventana de bienvenida a la aplicación



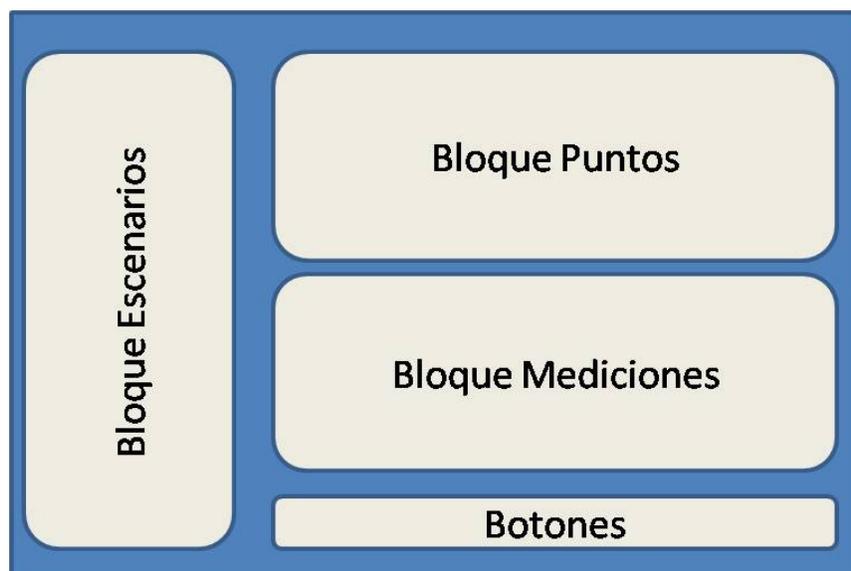
Figura 5.2.3 – Ventana de menú principal

5.2.1 Interfaz de “Tomar medidas”

La interfaz que desarrolla este caso de uso se va a implementar siguiendo el diseño de una única pantalla. Posiblemente sea la pantalla con más complejidad de la aplicación pero tendrá un uso sencillo.

A esta pantalla se accederá desde la pantalla del menú principal pulsando el correspondiente botón de acceso.

La pantalla deberá tener un aspecto similar al que se muestra en la figura 5.2.1.1



5.2.1.1 – Ventana Realiza experimento

En la imagen se aprecia que se deberá dividir verticalmente la pantalla en 2 para albergar así en su margen derecho y en una estructura de datos con los escenarios. Esta estructura de poder desplegar algún tipo de menú que permita la modificación de los escenarios existentes y la inserción de nuevos.

En la parte derecha de la pantalla se debe dividir de nuevo el espacio, pero esta vez de forma horizontal y en 3 secciones. En la parte superior se debe crear una estructura de datos para albergar los puntos relacionados con el escenario seleccionado en el margen izquierdo así como botones que permitan editar los puntos existentes, borrarlos o crear nuevos.

En la parte central se debe de crear una estructura que albergue los datos de las mediciones que se tomen cuando se pulse el botón de esta zona etiquetado con el texto Realizar mediciones. Cada medición debe estar identificada por la mac del punto de acceso y la potencia rssi obtenida.

Por último en la parte inferior de esta división habrá 3 botones:

- Limpiar Mediciones: este botón dejará vacía la tabla de las mediciones y no las almacenará en ninguna parte.
- Guardar experimento: limpiará la tabla de las mediciones y las guardará junto con el punto al que pertenecen y la fecha en la que se ha pulsado el botón pasando a formar así parte de un nuevo experimento.
- Volver: al pulsar este botón se volverá a la pantalla del menú principal de la aplicación.

5.2.2 Interfaz de “Realizar modelo”

Al igual que el caso de uso “Tomar medidas”, la interfaz de este caso de uso será también una única pantalla. Será una pantalla muy sencilla y a ella se accederá a través de la correspondiente opción del menú principal.

La pantalla será algo similar al diseño presentado en la figura 5.2.2.1:

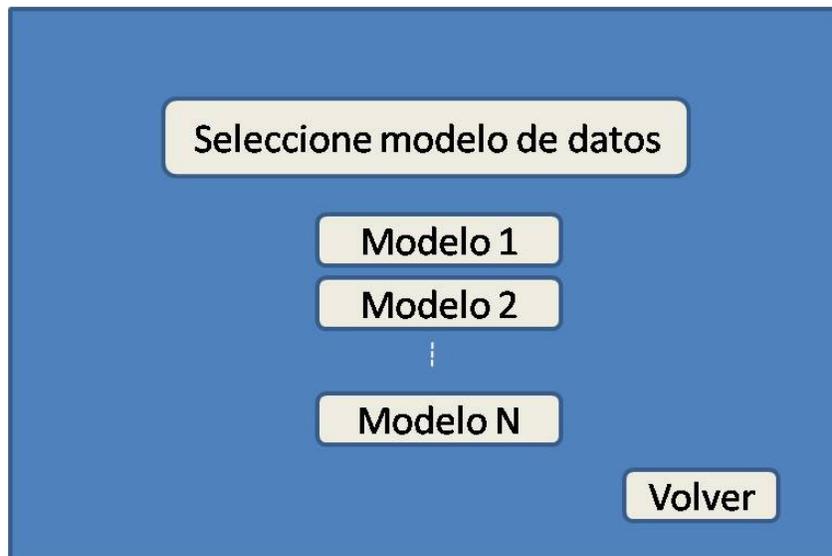


Figura 5.2.2.1 – Ventana de modelado

Como se puede observar en un estado inicial la ventana constará de 3 elementos básicos:

- Etiqueta de texto: donde se le invita al usuario que seleccione una opción para realizar el modelo de datos con las medidas de las que se dispone en la base de datos.
- Lista que enumera los tipos de modelos de datos que se pueden realizar con los datos de los que se dispone en la base de datos, en la que de algún modo debe poder seleccionarse un modelo para así usarlo como modelo de referencia.
- Botón volver: oculta la pantalla de modelado y devuelve al usuario al menú principal.

5.2.3 Interfaz de “Detectar ubicación”

La interfaz que se diseñada para este caso de uso es una única ventana muy sencilla que será de un tamaño ligeramente inferior al de las otras ventanas y en el cuál se

mostrará un texto con la ubicación si se ha localizado el equipo y si no se ha conseguido localizar, el texto indicará que no ha sido posible la localización.

Como añadido esta ventana, al igual que las anteriores, dispondrá de un botón que permitirá al usuario retornar al menú principal.

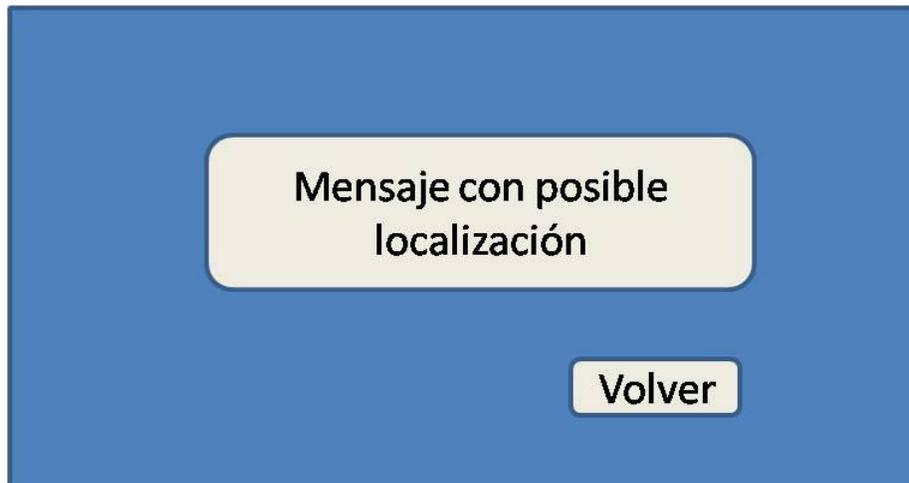


Figura 5.3.2.1 – Ventana de Detectar ubicación

5.3 Diseño de la lógica de la aplicación

Por lógica de la aplicación se entienden todas las operaciones que corren por debajo de la interfaz y hacen que ésta funcione y realice lo que se espera de la aplicación.

Los la lógica de los procesos que se van a desarrollar se pueden dividir en 4 grupos en función de lo que se realizará:

- Obtención de la medición.
- Almacenamiento y recuperación de mediciones.
- Realización de modelo de datos.
- Detectar ubicación actual.

5.3.1 Lógica de obtención de la medición

La lógica para la obtención de las mediciones es el pilar básico sobre el cual se fundamenta el prototipo de ahí que su obtención sea una de las lógicas más importantes del desarrollo, ya que una vez se obtengan las mediciones todo girará alrededor de ellas.

Este proceso lógico de obtención de las medidas se puede dividir en 2 procesos. El primero de estos procesos es la realización de una llamada al sistema para así poder obtener los datos de la tarjeta de red Wifi. Con esta llamada obtendremos los datos que se necesitan junto con otros muchos y en un formato que no interesa por lo que habrá que tratarlos. Ya que el desarrollo se va a llevar a cabo en un sistema Linux (Ubuntu), la llamada al sistema que existe y mejor se ajusta a lo que se necesita es el comando *iwlist*.

El segundo proceso de lógica que hay que desarrollar para obtener los datos que interesan de la medición es un tratamiento de la respuesta a la llamada al sistema realizada. Este proceso variará en función del formato de la respuesta de la llamada al sistema.

5.3.2 Lógica del almacenamiento y recuperación de mediciones

Para poder llevar a cabo toda integración de la interfaz gráfica del prototipo con el soporte de almacenamiento permanente de datos se deberá de crear una serie de clases u objetos que contengan todas las operaciones, métodos o funciones para poder manipular (almacenamiento, modificación y borrado) los distintos elementos que conforman el esquema descrito en el apartado 5.1

También habrá que proveer de algún método o función mediante el cual el apartado gráfico pueda cargar en la interfaz de una sola vez los datos necesarios bien sea para mostrarlos a la hora de hacer nuevas mediciones o por que se necesiten a la hora de realizar un modelo de datos de referencia.

Por último hacer notar que para que todas estas operaciones se puedan llevar a cabo habrá que proveer al desarrollo de una conexión con el sistema gestor de base de datos con el cual se lleve a cabo la integración.

5.3.3 Lógica para la realización del modelo de datos

Para poder llevar a cabo posteriormente la localización de un dispositivo se debe de contar con un modelo de datos de referencia para poder comparar valores y así tratar de establecer la localización del dispositivo.

Debido a que de un mismo punto se puede tener un infinito número de medidas, que se hayan tomado en diferentes días o que se hayan tomado en situaciones ambientales distintas se debe de conseguir una medida “estándar” para cada punto que se tenga registrado de forma que esa nueva medida sea la que defina la localización en dicho punto.

La opción que se tomará a la hora de realizar el modelo de datos para cada uno de los puntos será la de calcular la media de las señales rssi asociadas a una misma mac y para un mismo punto.

5.3.4 Lógica para la detección de la posición

La lógica que se va implementará para detectar la ubicación del dispositivo es bastante sencilla. Se realizará una medida en el momento de querer establecer la posición, esta medida será la medida de referencia.

Con la medida de referencia y el modelo de datos, se calculará la distancia euclídea desde el punto en el que se ha tomado la medida de referencia con respecto a cada uno de los puntos del modelo de datos.

Para determinar en qué punto se encuentra el dispositivo bastará con ver cuál es el punto en el que la distancia es menor y si coincide que en ese mismo punto se han detectado los mismos AP's que en el punto donde se ha tomado la medida de referencia, se podrá indicar que se cree que dicho punto es en el que está situado el dispositi-

vo. Si por el contrario no coincidieran todos los puntos de acceso, lo que se podrá afirmar es que es posible que se encuentre cerca de dicho punto pero nada más.

Implementación

6. Implementación

En este capítulo se va a describir cómo se ha llevado a cabo la implementación y desarrollo de los diseños expuestos en el capítulo 5. Se indicará la tecnología que se ha usado para llevar a cabo las implementaciones así como el resultado final de las mismas.

El desarrollo se realizará usando el lenguaje de programación Java, ya que el usar este lenguaje hará que simplemente retocando una pequeña parte del código este prototipo sea fácilmente portable a sistemas operativos y otro tipo de dispositivos como teléfonos móviles, pdas o tablets.

6.1 Implementación de la base de datos

Para realizar la implementación de la base de datos se ha elegido a MySQL como sistema gestor para este prototipo. La elección de este gestor ha venido determinada por la gratuidad de la herramienta y porque utilizado un lenguaje SQL sencillo y bastante “limpio”.

Basado en el esquema E/R de la figura 5.1.1 se ha implementado el script de creación para 5 tablas. Estas tablas son las 4 entidades que se pueden observar en el esquema E/R (Punto de Acceso, Experimento, Escenario y Punto) y la tabla Medición que se origina a partir de la relación existente entre un punto, un punto de acceso y un experimento.

El esquema de creación de las tablas con sus relaciones y dependencias es el siguiente:

```
DROP TABLE Medicion;
DROP TABLE Experimento;
DROP TABLE AccessPoint;
DROP TABLE Punto;
DROP TABLE Escenario;

CREATE TABLE Escenario(
    id_esc varchar(50),
    descrip_esc varchar(100),
    CONSTRAINT EscenarioPK PRIMARY KEY (id_esc)
```

```
);
```

```
CREATE TABLE Punto(  
    id_esc varchar(50) NOT NULL,  
    coord_x int(6),  
    coord_y int(6),  
    coord_z int(6),  
    descrip_punto varchar(100),  
    CONSTRAINT PuntoPK PRIMARY KEY (id_esc, coord_x, coord_y, coord_z),  
    CONSTRAINT PuntoFKEscenario FOREIGN KEY (id_esc) references  
Escenario(id_esc) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE AccessPoint(  
    MAC varchar(17),  
    CONSTRAINT AccessPointPK PRIMARY KEY(MAC)  
);
```

```
CREATE TABLE Experimento(  
    id_Experimento int(9) AUTO_INCREMENT,  
    fecha_hora varchar(50),  
    descrip_exp varchar(100),  
    CONSTRAINT ExperimentoPK PRIMARY KEY (id_Experimento)  
);
```

```
CREATE TABLE Medicion(  
    experimento int(9),  
    escenario varchar(50),  
    coord_x int(6),  
    coord_y int(6),  
    coord_z int(6),  
    AP varchar(17),  
    rssi varchar(50),  
    CONSTRAINT MedicionPK PRIMARY KEY(experimento, escenario, coord_x,  
coord_y, coord_z, AP),  
    CONSTRAINT MedicionFKExperimento FOREIGN KEY (experimento) references  
Experimento(id_Experimento) ON DELETE CASCADE ON UPDATE CASCADE,  
    CONSTRAINT MedicionFKPunto FOREIGN KEY (escenario, coord_x, coord_y,  
coord_z) references Punto(id_esc, coord_x, coord_y, coord_z) ON DELETE CASCADE  
ON UPDATE CASCADE,  
    CONSTRAINT MedicionFKAccessPoint FOREIGN KEY (AP) references  
AccessPoint(MAC) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Para que el prototipo tenga acceso a esta base de datos y a las operaciones que debe de realizar con ella, se ha creado, dentro del proyecto Java de Laplowix, un paquete llamado accesoBBDD mediante el cual se permitirá integrar la base de datos con el resto del desarrollo.

El paquete está compuesto por una interfaz y 3 clases java como se puede ver en la siguiente figura:

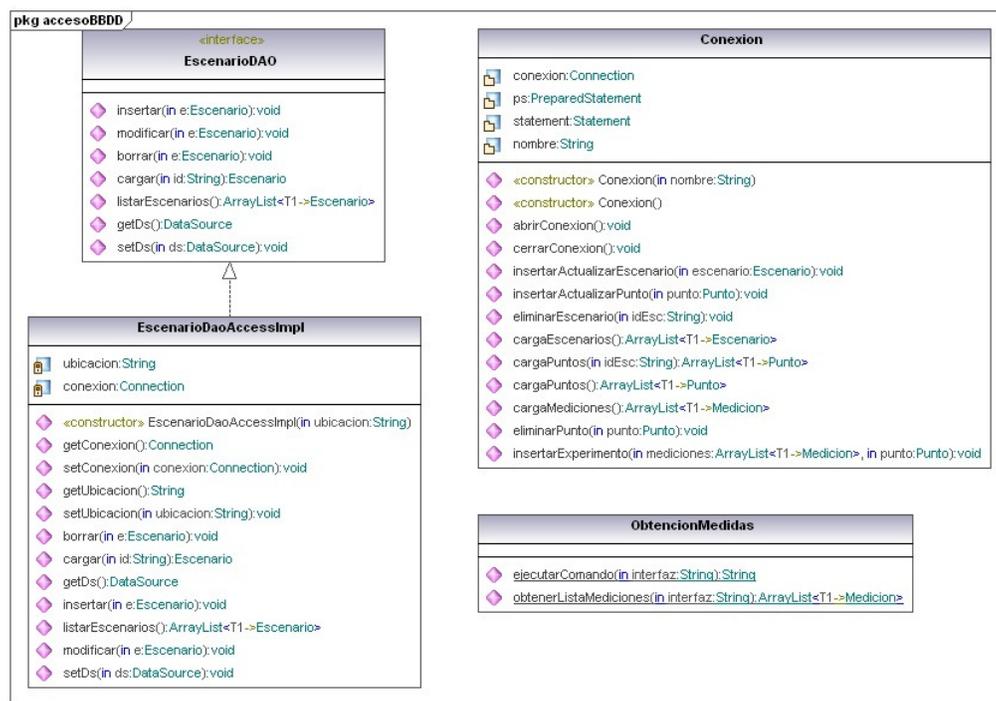


Figura 6.1.1 – Paquete accesoBBDD

La interfaz EscenarioDAO, se ha creado con la intención de enumerar las operaciones básicas que toda implementación de acceso a los objetos de tipo Escenario debieran tener. La implementación de dicha interfaz se puede observar en la clase EscenarioDAOAccessImpl, en la cual se han implementado para MySQL las operaciones requeridas.

Luego se puede observar la clase Conexion, en la cual se han implementado las operaciones básicas que se van a realizar con los objetos del modelo de datos, como es la

inserción o modificación de Puntos y Escenarios, la carga de Puntos, Mediciones o escenarios y quizás la operación más importante de esta clase que es la operación que permite la inserción en el sistema de base de datos de un experimento. Se decidió que estas lógicas operacionales estuviesen juntas en una sola clase ya que no son demasiadas operaciones y así se tendría acceso a todas las operaciones que se usan de una forma más rápida.

La última clase que queda en este paquete es ObtencionDeMedidas, esta clase es por así decirlo la piedra angular del proyecto y sobre la que se asienta gran parte de la lógica. Si bien no accede directamente a la base de datos, la clase Conexion se nutre de su información a la hora de insertar un nuevo Experimento. Esta clase tiene dos métodos, en ejecutarComando se realiza la ejecución de la llamada al sistema mediante la cual se obtienen los datos necesarios para el funcionamiento del sistema y en obtenerListaMediciones se parsea la respuesta del sistema para poder así traducirlo al modelo de datos con el que se está trabajando.

6.2 Implementación de la interfaz gráfica

La implementación de la interfaz gráfica de Laplowix se ha desarrollado en el paquete java llamado igu (interfaz gráfica de usuario).

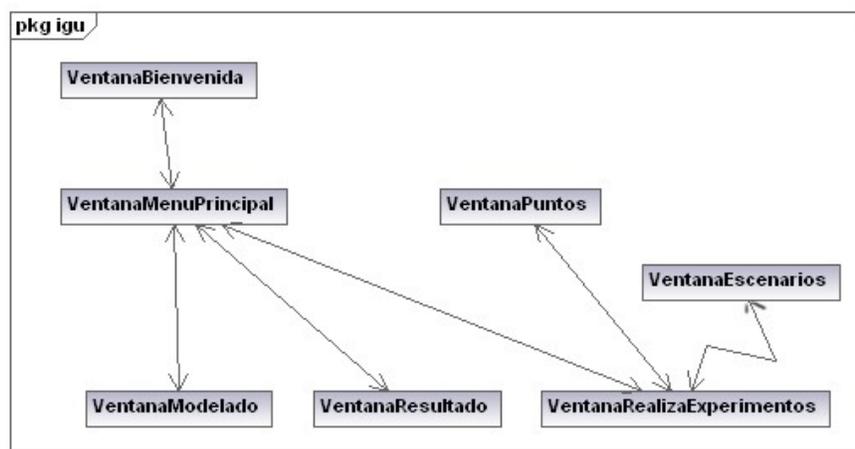


Figura 6.2.1 – Paquete igu

Se puede observar cómo en el paquete hay 7 clases desarrolladas, todas salvo `VentanaEscenarios` y `VentanaPuntos` son las que se establecieron para realizarse en el análisis del prototipo. Las dos ventanas nuevas se han creado para facilitar la tarea de crear o modificar nuevos escenarios y puntos. En los anexos se podrá ver una imagen con las clases ya desarrolladas con todos sus atributos y métodos.

6.2.1 Ventana de bienvenida

Esta ventana se ha implementado en la clase `VentajaBienvenida.java`, es una ventana bastante sencilla y tal y como se pedía en el diseño tiene un mensaje de bienvenida, un botón para pasar al menú principal y un botón por si se quiere salir de la aplicación.

Se puede observar cómo ha quedado en la figura 6.2.1.1:



Figura 6.2.1.1 – Ventana de bienvenida

6.2.2 Ventana del menú principal

La ventana del menú principal es una ventana sencilla, como se especifica en la fase de diseño tiene una etiqueta que indica que se debe seleccionar una opción y también posee 4 botones, los 3 primeros son para entrar en las opciones de la aplicación y el último es para salir de la misma.

La ventana se ha desarrollado implementando la clase `VentanaMenuPrincipal.java`:



Figura 6.2.2.1 – Ventana menú principal

Dentro de la implementación de esta ventana se ha implementado un pop-up de advertencia por si se accede a la aplicación y se trata directamente de realizar una localización sin antes haber realizado un modelo de referencia con los datos de los que ya se disponía. Dicho mensaje de aviso se muestra en la figura 6.2.5.1 dentro del apartado de la ventana de resultado

6.2.3 Ventana de realizar experimentos

Al igual que las otras dos ventanas esta clase se ha desarrollado implementando la clase `VentanaRealizarExperimentos.java` y siguiendo el diseño indicado.

La ventana se ha dividido en dos partes. En la parte izquierda se ha creado un espacio en el que se pueden ver los escenarios que hay en una estructura de tipo árbol.

Por otra parte el lado derecho se ha subdividido horizontalmente en 3 partes. En la primera se ha insertado una fila de botones y una tabla, con los botones se podrán manipular los puntos y en la tabla aparecerán los puntos asociados al escenario seleccionado en cada momento.

En la parte central de la zona derecha se puede ver una estructura de tabla en la que se pueden ver las medidas que se acaben de realizar.

Por último en la parte de abajo existen 3 botones que permiten guardar el experimento realizado, limpiar las medidas tomadas y volver a la ventana del menú principal.

En la siguiente figura se puede ver cómo ha quedado la ventana:



Figura 6.2.3.1 – Ventana de realizar experimentos

Adicionalmente se han creado dentro de esta clase dos subclases para representar dos pequeñas ventanas en las que se insertan los datos de modificación y/o creación de escenarios y puntos. Estas ventanas son bastantes sencillas y a parte de los botones de aceptar y cancelar, sólo tienen los cuadros de texto necesarios en cada caso con las etiquetas que identifican a qué campo del objeto en cuestión se corresponde.

La ventana en la que se insertan o modifican los escenarios es la siguiente:

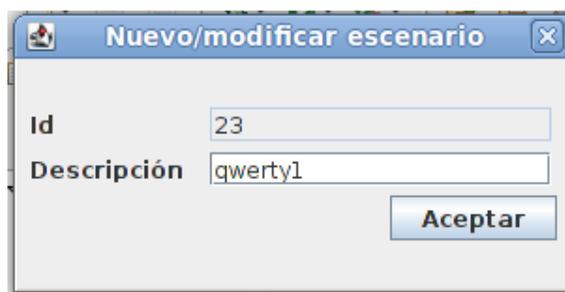


Figura 6.2.3.2 – Ventana escenarios

La ventana en la que se modifican o insertan nuevos puntos es:

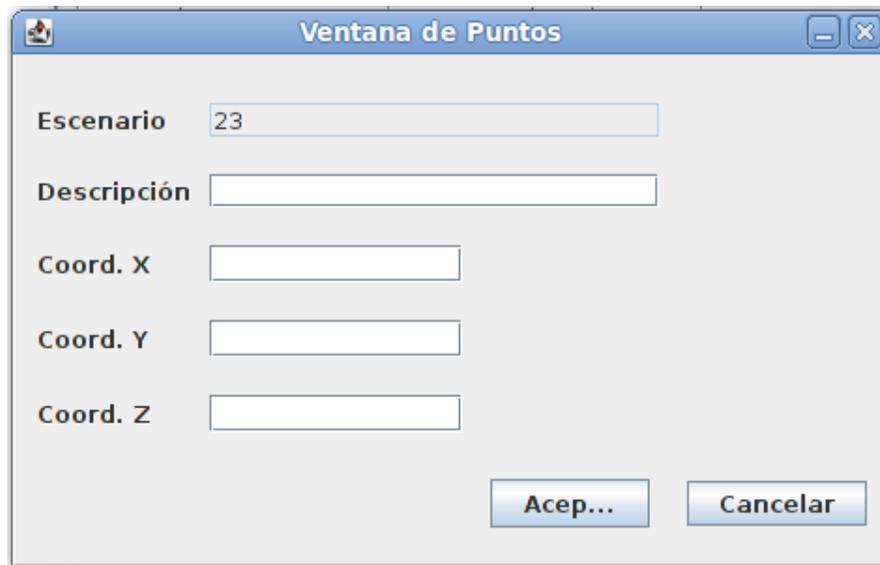


Figura 6.2.3.3 – Ventana puntos

6.2.4 Ventana de modelado

La clase de modelado se ha desarrollado en `VentanaModelado.java`. La ventana tiene 3 elementos básicos.

En la parte superior tiene una etiqueta de texto grande en la que indica que se elija una opción. En la parte central tiene un checkbox con dos cheks que están puesto para que se implemente un modelo de datos basado en la media u otros. Realmente sólo se ha implementado el modelo de la media y si se pulsa el check "otros" saldrá un mensaje indicando que sólo se puede realizar la media actualmente. Por último dispone de un botón para volver al menú principal en caso de que se quiera salir sin realizar el modelo.

La ventana de modelado tiene el siguiente aspecto:



Figura 6.2.4.1 – Ventana de modelado

6.2.5 Ventana Resultado

Al igual que las otras ventanas principales, esta ha sido desarrollada en una clase llamada `VentanaResultado.java`

Se accede a ella pulsando “Localizar equipo” desde el menú principal y si se ha realizado un modelo de datos ya que si no se ha hecho aparecerá el siguiente mensaje:

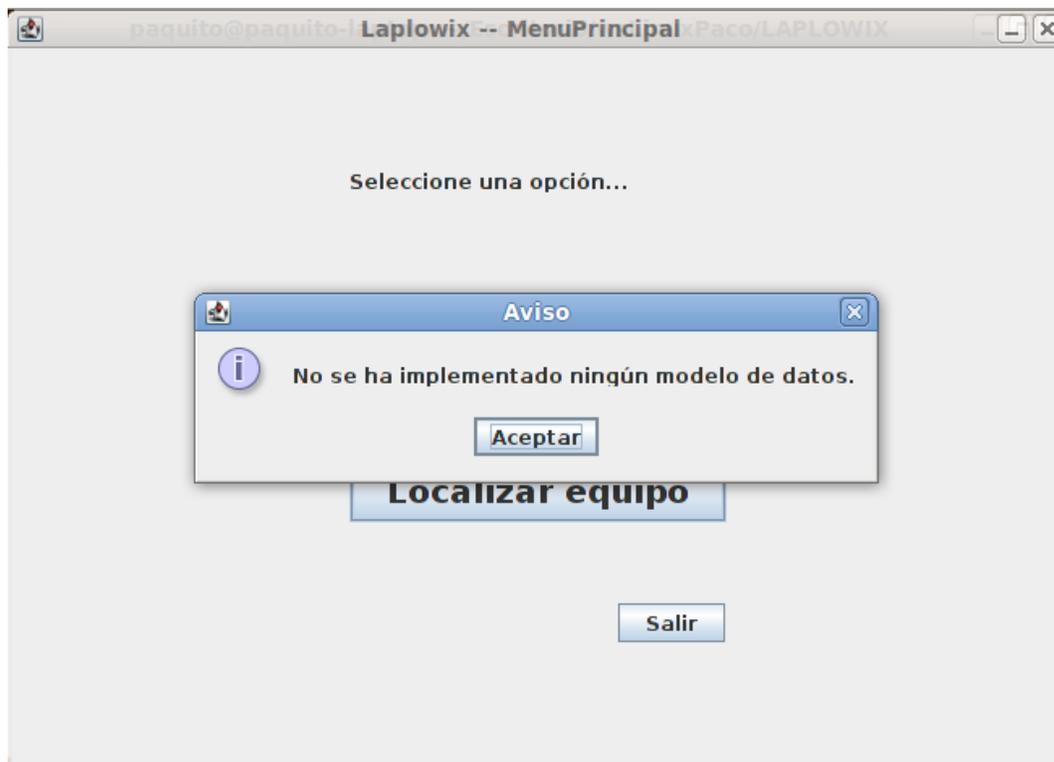


Figura 6.2.5.1 – Mensaje de aviso

Esta ventana es ligeramente más pequeña que las anteriores. Posee la misma anchura pero es ligeramente menos alta. En esta ventana tal y como se indicó en el diseño sólo hay un elemento con el cuál el usuario puede interactuar y es el botón que indica *volver* al menú principal. A parte de este botón hay un cartel que indica si se ha producido la localización correcta o no y en caso de ser correcta indica en punto en el que se cree que está.

En la siguiente figura se puede observar cómo se ha implementado dicha ventana:

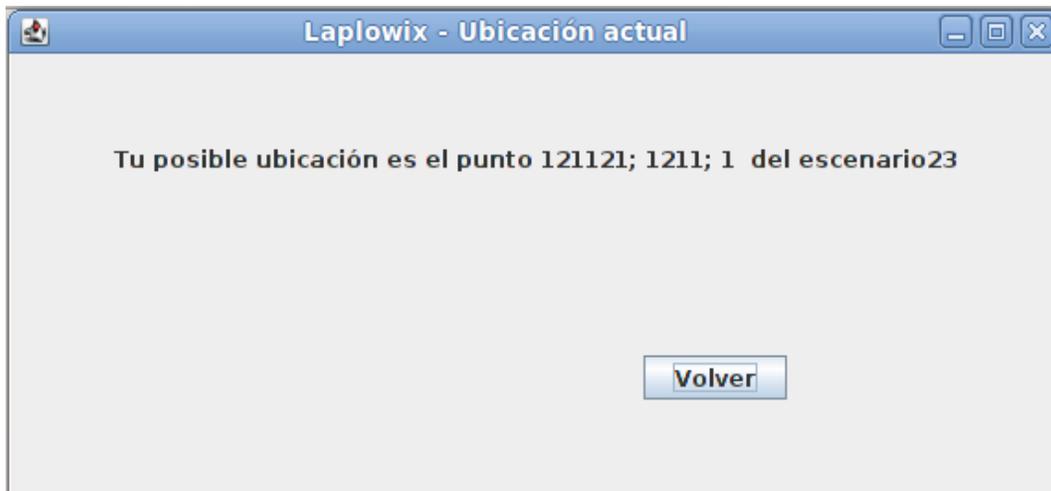


Figura 6.2.5.2 – Ventana de resultado

6.3 Implementación de la lógica de la aplicación

La lógica de la aplicación se divide en 3 partes. Por un lado se hablará de la obtención de las macs de los puntos de acceso y la potencia que llega a éstas, por otra parte habrá que realizar el modelo de datos para poder realizar la comparación de la localización y por otro lado se hablará de cómo se detecta la localización en un punto concreto del equipo en el que se ejecuta Laplowix.

6.3.1 Lógica de la obtención de medidas

Todo este proceso de la obtención de medidas se ha implementado en la clase `ObtencionMedidas.java` del paquete `accesoBBDD`.

Tal y como se indica en el diseño la obtención de las macs y de la potencia rssi se hará mediante una llamada al sistema. La llamada al sistema que se emplea es:

```
iwlist nombreDelDispositivoRed scan
```

Y la respuesta que se obtiene de una llamada es la siguiente.

```
paquito@paquito-laptop:~$ iwlist eth1 scan  
eth1 Scan completed :  
Cell 01 - Address: 40:4A:03:CD:CD:60  
ESSID:"WLAN_1D"
```

```
Mode:Master
Channel:9
Frequency:2.452 GHz (Channel 9)
Signal level:-55 dBm Noise level:-106 dBm
Encryption key:on
Extra:bcn_int=100
Extra:capab=0x0471
Extra: Last beacon: 2748ms ago
Cell 02 - Address: 50:67:F0:A9:E1:3B
ESSID:"WLAN_B9"
Mode:Master
Channel:1
Frequency:2.412 GHz (Channel 1)
Signal level:-87 dBm Noise level:-100 dBm
Encryption key:on
Extra:bcn_int=100
Extra:capab=0x0471
Extra: Last beacon: 2748ms ago
```

Como se puede ver en la respuesta puesta como ejemplo se devuelve con dos puntos de acceso ya que viene cada punto de acceso separado por la palabra *Cell*. Entre los parámetros que se pueden obtener se encuentran los que se necesitarán para identificar el punto de acceso *Address* y también la señal para ver la potencia con el campo *Signal level*.

Este proceso se lleva a cabo en el método `ejecutarComando` dentro de la clase `ObtencionMedidas` del paquete `accesoBBDD`.

Una vez realizada la llamada al sistema también se ha creado un método que permita traducir la respuesta de la llamada al sistema al modelo de datos creado para representar el dominio conocido. El método que se encarga de realizar esta conversión es `obtenerListaMediciones` de la misma clase `ObtencionMedidas`. Este método devolverá un `ArrayList` de datos tipo `Medición` de forma que así el resto del sistema pueda entender el tipo de datos devuelto por la llamada al sistema.

6.3.2 Lógica de modelado tipo media

El proceso de creación del modelo de datos para realizar las comparaciones que luego servirá para realizar la localización se encuentra en la clase de la interfaz `VentanaModelado`. El método se llama `realizarModeloMedia`.

En este método se cargan en una estructura de datos todos los puntos y todas las medidas asociadas a éstos.

El proceso irá pasando punto a punto y dentro de cada uno de los puntos recorrerá todas las mediciones asociadas a ese punto. Y agrupando las mediciones por la `mac` a las que pertenecen va realizando la media de la potencia rssi asociada a cada `mac`.

Recorriendo así todas las mediciones de cada punto se va obteniendo poco a poco las mediciones medias para cada uno de los puntos.

Una vez se hayan realizado las medias de todos los puntos ya tendremos un modelo de datos contra el cuál poder contrastar para intentar realizar de forma posterior la localización.

Cabe mencionar que cuantos más experimentos y medidas se tomen sobre los puntos en los que posteriormente se intentará la localización, mejor será el modelo de datos que se pueda tomar como modelo.

6.3.3 Lógica de la localización

La lógica para llevar a cabo la localización del equipo no es demasiado complicada aunque quizás sí algo liosa. Está desarrollada en la clase `VentanaResultado.java` dentro del método `calcularUbicacion`.

El algoritmo que se ha implementado para determinar la ubicación comienza con la obtención de una medida con la que poder comparar la ubicación actual.

Una vez se obtiene la medida se puede proceder al siguiente paso que es la creación de una matriz de tipo `Double` de tamaño igual al número de puntos que se tiene en el modelo que se usará como referencia, esta matriz *Distancias* tendrá todas sus posiciones inicializadas a cero. Una vez se dispone del modelo, de la medición hecha y de la

estructura auxiliar, se comienza el proceso de comparar la medición hecha con los puntos que hay en el modelo. En esta comparación lo que se va a hacer es calcular las distancias euclídeas punto a punto, esto se hará comparando las macs de cada uno de los puntos del modelo con la medición inicial que se ha hecho. Estas distancias se van acumulando en la posición correspondiente al punto dentro de la matriz de *Distancias*. Si el punto la mac de la medida tomada que se está intentando comparar no se encuentra en el punto en cuestión, se agregará una distancia “máxima”.

Cuando se hayan calculado todas las distancias y las estructura *Distancias* ya esté rellena se recorrerá la matriz dicha matriz. Una vez recorrida esta matriz nos quedaremos con la posición que tenga un valor menor, que será aquel punto que esté más cerca o incluso el mismo punto. Una vez se tenga la posición de *Distancias*, -entonces se va con la posición al array en el que se tiene el modelo de datos y ya se tiene el punto en el que se encuentra el dispositivo.

Tras esta versión y habiendo analizando diversas pruebas se llegó a la conclusión de que a este sistema para inferir la localización fallaba en algo. Ese algo es que podía pseudo-localizar el dispositivo incluso si ninguna de las macs de la medida auxiliar coincidía con las macs del punto con el cuál se estaba comparando y calculando las distancias. Pero una vez detectado el fallo, también se encontró la solución.

La solución pasa por la creación de una matriz de tipo de dato booleano y de tamaño el mismo que la matriz de distancias y que se llama *distanciaExacta*. Dicha matriz tendrá todas sus posiciones inicializadas a “verdadero”. Ahora con esta matriz cuando se esté comparando una mac de la medida de referencia con las macs del punto que se esté comparando si no se encuentra una mac con la que comparar se rellenará la posición de este array correspondiente al punto con el valor “falso”, así si la distancia de ese punto fuera la mínima se sabría que no es certera ya que al menos una de las direcciones mac no se han podido cotejar.

Por tanto ahora a la hora de decidir en qué punto se encuentra el dispositivo y se tenga la posición de *Distancias* donde la distancia es la mínima, se va a comprobar en el array *distanciaExacta* para ver si la distancia y la comparación ha sido una comparación

buena o no. Si el valor en esta matriz es “verdadero”, entonces se va con la posición al array en el que se tiene el modelo de datos y ya se tiene el punto en el que se encuentra el dispositivo. Si el valor en la tabla *distanciaExacta* es false, se concluye que no se ha podido realizar la localización del dispositivo.

Evaluación

7. Evaluación

En este apartado tendrá lugar la documentación de las pruebas que se han realizado para verificar el correcto funcionamiento del prototipo Laplowix.

Con estas pruebas se podrá posteriormente concluir si el sistema es válido o no para usarse como inicialmente se pensó para poder usarlo como sistema de localización de dispositivos portátiles para interiores.

7.1 *Evaluación de la interfaz gráfica*

No se ha llevado a cabo una evaluación de la interfaz gráfica debido a la falta de usuarios finales y a que esta evaluación no formaba parte de los objetivos iniciales del proyecto.

Esta evaluación podría haberse realizado con unos pocos usuarios pero no habría sido una muestra que tuviese significancia estadística. Si se hubiera podido disponer de un presupuesto amplio se habría llevado a cabo dicha evaluación con el número suficiente de usuarios de forma que la evaluación fuera lo suficientemente objetiva.

Pese a no haber podido realizar esta evaluación objetiva sí se ha realizado una pequeña evaluación personal y subjetiva llegando a la conclusión personal de que la interfaz realizada ha facilitado en gran medida la creación de puntos y escenarios, la toma de medidas y creación de experimentos y la creación de un modelo de datos, así como la localización. Esto ha sido posible porque es una interfaz sencilla y funcional que cumple con los requisitos que se le piden en el diseño sin agregar otras funciones que puedan hacerla más engorrosa y menos sencilla al uso.

7.2 Evaluación de la lógica de la aplicación

A continuación se van a describir dos pruebas que se han realizado para comprobar el funcionamiento del prototipo desarrollado para tratar de realizar la localización de dispositivos mediante la red WiFi.

Se van a realizar dos pruebas en dos habitaciones (escenarios) distintos, serán de distinto tamaño, se detectará distinto número de redes WiFi y tendrán distinta forma.

7.2.1 Prueba 1

Esta prueba se ha llevado a cabo en una habitación de aproximadamente 22 m^2 con la siguiente silueta:

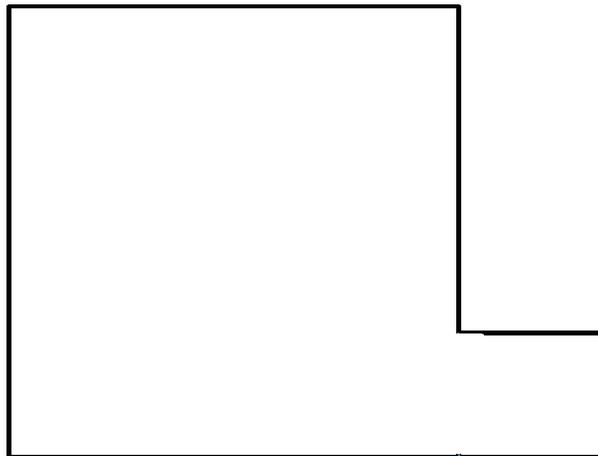


Figura 7.2.1.1 – Silueta habitación 22 m^2

La parte cuadrada de la habitación tiene 20 m^2 y el apéndice de la esquina inferior derecha tiene 2 m^2 .

Antes de realizar la prueba propiamente dicha se han realizado varias mediciones al azar en unos cuantos puntos aleatorios y se ha comprobado que en toda la habitación hay acceso al mismo número de redes WiFi, en concreto se detectan cuatro redes.

Para llevar a cabo la prueba se identificaron dos puntos en la habitación sobre los cuáles se iban a realizar mediciones para posteriormente poder realizar un modelo de datos sobre ellos. Los puntos están situados donde se indica en la siguiente figura:

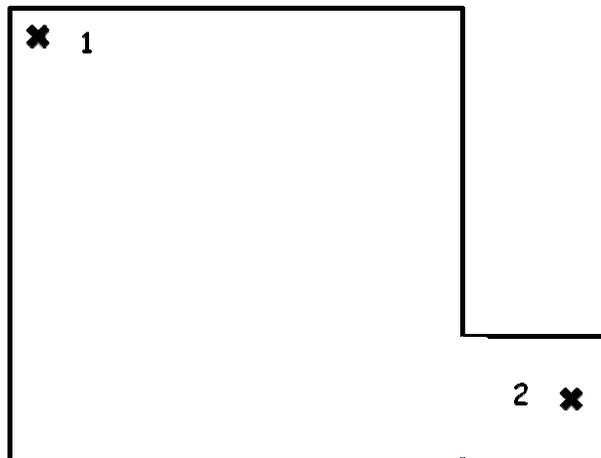


Figura 7.2.1.2 – Puntos de referencia en habitación 22m²

Para realizar la toma de mediciones de forma que se recojan bien todas las medidas posibles se han realizado sobre cada uno de los puntos 3 experimentos o toma de mediciones con un intervalo de tiempo entre cada uno de 10 minutos.

Una vez ya obtenidas las mediciones necesarias y realizado el modelo de datos de referencia se han comenzado a realizar localizaciones aleatorias, en torno a las 100 localizaciones, en casi cualquier parte de la habitación y analizando estas mediciones se han podido identificar 3 zonas en la habitación.

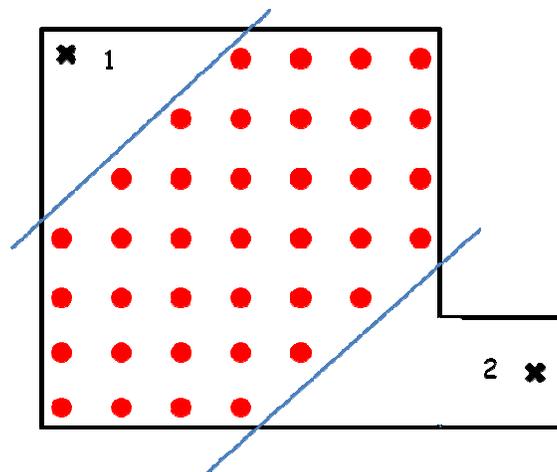


Figura 7.2.1.3 – Zonas en habitación de 22 m²

En las zonas próximas a los puntos 1 y 2 más del 90% de las localizaciones realizadas daban como resultado que el dispositivo se encontraba en estos puntos. Sin embargo en las mediciones aparece una zona de incertidumbre, la marcada con puntos en la imagen. En esta zona de incertidumbre los resultados han sido contradictorios ya que aproximadamente un 50% de las mediciones han dado como resultado que la ubicación era el punto 1, un 30% que la ubicación era el punto 2 y el 20% restante ha dado como resultado que no se ha podido establecer la localización.

Si analizamos estos resultados, el que en la zona de incertidumbre se pueda dar la localización en el punto 1 y 2 es algo predecible, pero no lo es el que aparezca la no localización por tanto habrá que analizar posteriormente este resultado.

7.2.2 Prueba 2

Esta segunda prueba se ha llevado a cabo en una habitación de unos 40 m² que presenta la siguiente silueta:

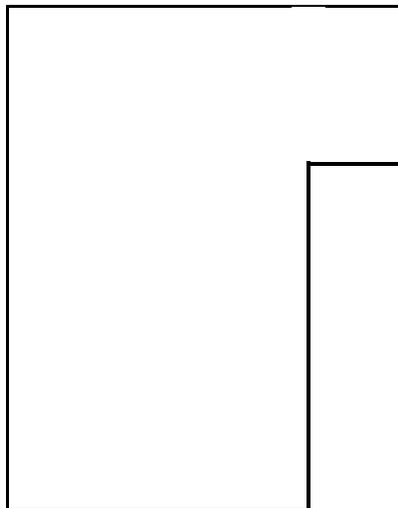


Figura 7.2.2.1 – Silueta habitación de 40 m²

La parte superior izquierda de esta habitación ocupa una superficie de unos 3'5 m² dejando al resto de la estancia en 36'5 m².

Al igual que con la anterior habitación se han realizado unas cuantas mediciones de forma aleatoria por la habitación para ver que en la mayor parte de la misma se recibía la señal del mismo número de redes WiFi y en concreto se recibe señal de 3 redes.

Debido a que esta habitación es más grande que la anterior, para realizar el mapeado del escenario y proceder así a obtener un modelo más o menos preciso de la habitación se van a establecer tres puntos desde los cuales realizar las mediciones para este modelo.

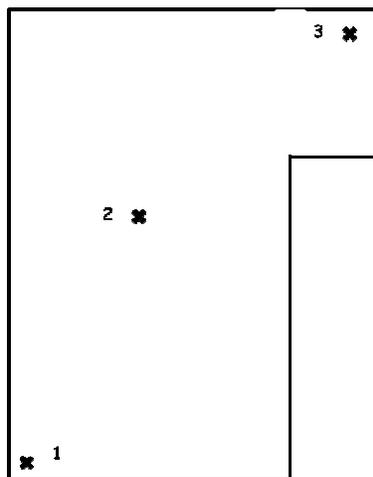


Figura 7.2.2.2 – Puntos de referencia en habitación 40m²

Al igual que con la otra habitación, para realizar la toma de mediciones de forma que se recojan bien todas las medidas posibles se han realizado sobre cada uno de los puntos 3 experimentos o toma de mediciones con un intervalo de tiempo entre cada uno de 10 minutos.

Una vez ya obtenidas las mediciones necesarias y realizado el modelo de datos de referencia se han comenzado a realizar localizaciones aleatorias, en torno a las 150 localizaciones, en casi cualquier parte de la habitación y analizando estas mediciones se han podido identificar 4 zonas en la habitación.

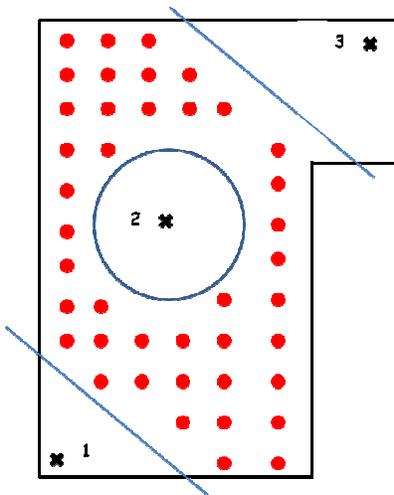


Figura 7.2.2.3 – Zonas en habitación de 40 m²

En las zonas próximas a los puntos 1, 2 y 3 más del 90% de las localizaciones realizadas daban como resultado que el dispositivo se encontraba en estos puntos. Sin embargo en las mediciones aparece una zona de incertidumbre, la marcada con puntos en la imagen, en esta zona de incertidumbre los resultados han sido contradictorios ya que aproximadamente un 50% de las mediciones han dado como resultado que la ubicación era el punto 2, que se encuentra en el centro de la habitación, un 35% que la ubicación era el punto 1 y el 15% restante ha dado como resultado que la ubicación era el punto 3.

Dentro de la zona de incertidumbre todos los casos que dieron como localización el punto 1 estaban situados de la mitad para abajo del plano, así como los que dieron como ubicación el punto 3 estaban situados en la parte superior. Pudiendo así dividir la zona de incertidumbre en 2 subzonas, aquella en la que la incertidumbre está entre el 1 y el 2 y aquella en la que la incertidumbre es entre el 2 y el 3, no habiéndose observado zonas en las que la incertidumbre sea entre los puntos 1 y 3.

Conclusiones

8. Conclusiones

A la vista de los resultados obtenidos en las pruebas 1 y 2 reflejadas en el apartado 7 de este documento se puede concluir que el prototipo Laplowix funciona ya que es capaz de localizar el equipo y también de no inferir localización alguna si se intenta realizar una localización en una zona que no es la esperada o si por el contrario se han introducido otras redes WiFi entre el mapeo de la zona y el intento de localización.

Pese a que el prototipo desarrollado funciona correctamente el resultado de las pruebas muestra que la aplicación está condicionada a la aparición de zonas de incertidumbre en las cuáles no es capaz de identificar correctamente en qué punto se encuentra pudiendo realizar en estas zonas dos operaciones de localización y obtener dos resultados distintos. Por tanto para mejorar el prototipo habría que tratar de eliminar estas zonas de incertidumbre o habría que tratar de minimizarlas. Dado que esta eliminación es prácticamente imposible y va ligada a la propia tecnología WiFi habría que tratar de minimizarla.

Es posible que debido a la aparición de las zonas de incertidumbre el prototipo no sea válido para tratar de localizar dispositivos dentro de una habitación. Por tanto es posible que este desarrollo, con las modificaciones que requiera, pudiera ser útil para por ejemplo identificar que el dispositivo se encuentra en una planta de un edificio si no es muy grande o para identificar si por ejemplo el dispositivo se encuentra en una zona amplia de una gran planta de edificio, como por ejemplo, se podría tratar de localizar en una planta de hospital si un dispositivo está en la zona de cirugía coronaria o en la de medicina interna, ambas amplias zonas dentro de una misma planta física.

Como conclusión final se puede decir que el prototipo desarrollado funciona y es capaz de localizar el dispositivo, pero debido a las variaciones que se producen en las potencias recibidas de parte de las redes WiFi debido a factores como son los muebles que se colocan en habitaciones, ventanas y puertas abiertas o cerradas y sobre todo a la aparición de las zonas de incertidumbre en las que el dispositivo no es capaz de discernir entre varios puntos en cuál se encuentra este prototipo será más útil y efectivo en zonas más amplias y no en habitaciones y salas no demasiado grandes.

Futuras líneas de trabajo

9. Futuras líneas de trabajo

Dentro de las futuras líneas de trabajo destinadas a mejorar el prototipo implementado en este proyecto fin de carrera se pueden distinguir cuatro líneas de trabajo a seguir:

- Realización de estudios destinados a minimizar la existencia de las zonas de incertidumbre descubiertas en las pruebas realizadas al prototipo Laplowix.
- Revisión y mejora de la interfaz gráfica, ya que pese a que cumple las funciones para las que ha sido creada es una interfaz sencilla y simple por lo que es uno de los puntos que se podría mejorar tratando así de dar otro aspecto al desarrollo realizado o incluso haciendo que deje de ser una aplicación de escritorio para pasar a ser una aplicación web.
- Otro de los puntos a mejorar es el de las comunicaciones de la interfaz con la base de datos. Se podría trabajar con una base de datos centralizada en algún servidor y que las distintas aplicaciones se conecten a ella a través de internet o de la intranet correspondiente.
- Por último otra futura línea de trabajo es la de migrar el sistema para que sea compatible con otros sistemas operativos. La interfaz actual es ampliamente compatible con otros sistemas operativos ya que está desarrollada en Java. Habría que tratar de adaptar las conexiones de la base de datos y habría que trabajar en realizar otra llamada al sistema, ya que al cambiar de sistema operativo lo normal es que no sea la misma.

Durante el proyecto se ha trabajado sobre la llamada al sistema que debería realizarse si el sistema se portara a OS X 10.7 Lion pero es un trabajo que no ha sido evaluado en este proyecto. La llamada sobre la que trabajar sería la llamada al sistema "*airport -s*", comando que se encuentra en dicho sistema operativo en la ruta:

```
/System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/airport/usr/sbin
```

Referencias

Referencias

[1] Definición sistema RFID

Definición de los fundamentos y características de los RFID

<http://es.wikipedia.org/wiki/RFID>

[2] Definición sistema GPS

Definición básica del funcionamiento del sistema GPS

<http://www.euroresidentes.com/gps/que-es-el-gps.htm>

[3] Definición navegador GPS

Definición básica del funcionamiento de un sistema de navegación GPS

<http://www.navegadorgps.org/>

[4] Definición de dirección MAC

Definición básica de qué es la dirección mac en los dispositivos de red

http://es.wikipedia.org/wiki/Direcci%C3%B3n_MAC

[5] Investigación Nokia Wifi

Artículo donde se habla de las investigaciones de Nokia acerca de la localización Wifi

<http://www.xataka.com/gps/nokia-trabaja-para-localizarnos-tambien-en-interiores>

[6] PFC María González García

Enlace al PFC de María González García de la biblioteca de la UC3M

<http://biblioteca.uc3m.es/uhtbin/cgiirsi/cTLkd1SMjc/CCSSJJ/286330135/9>

Bibliografía

Bibliografía

Java: Fundamentos de programación

Autor: Judy Bishop

Editorial: ADDISON-WESLEY

Año: 1999

Java 2: Interfaces graficas y aplicaciones para internet

Autor: Francisco Javier Ceballos Sierra

Editorial: Ra-Ma

Año: 2008

Java 2: Curso de programación

Autor: Francisco Javier Ceballos Sierra

Editorial: Ra-Ma

Año: 2010

Estudio de diferentes sistemas de posicionamiento y de su aplicación con Zigbee

Autor: María González García

Editorial: PFC – Biblioteca Escuela Politécnica UC3M

Año: 2009

Anexos

ANEXO I: Diagramas de clase

A continuación se muestran los diagramas de clase de las clases implementadas en el desarrollo del prototipo Laplowix.

El primer diagrama que se muestra a continuación es el diagrama general en el que se puede ver la clase principal (Laplowix) y los paquetes de los que se compone la aplicación:

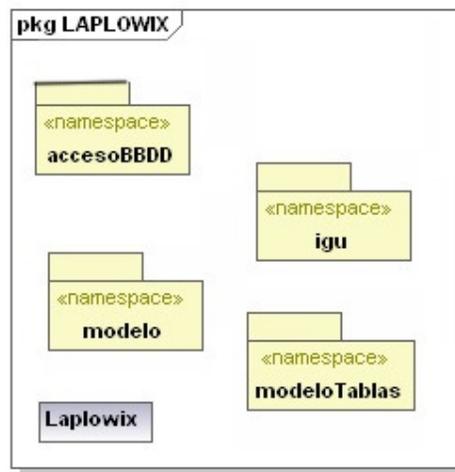


Figura Anexo I 1 – Paquete general de la aplicación

A continuación se muestra el paquete modeloTablas, que son las clases en las que se apoyan las tablas que aparecen en la ventana en la que se realizan los experimentos:

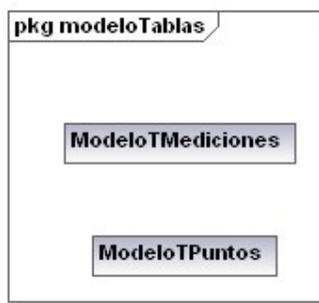


Figura Anexo I 2 – Paquete modeloTablas

Ahora se muestra el paquete modelo, en el cuál se han tratado de representar los elementos que se almacenan en la bases de datos y que sirven como referencia para modelar el universo de la aplicación:

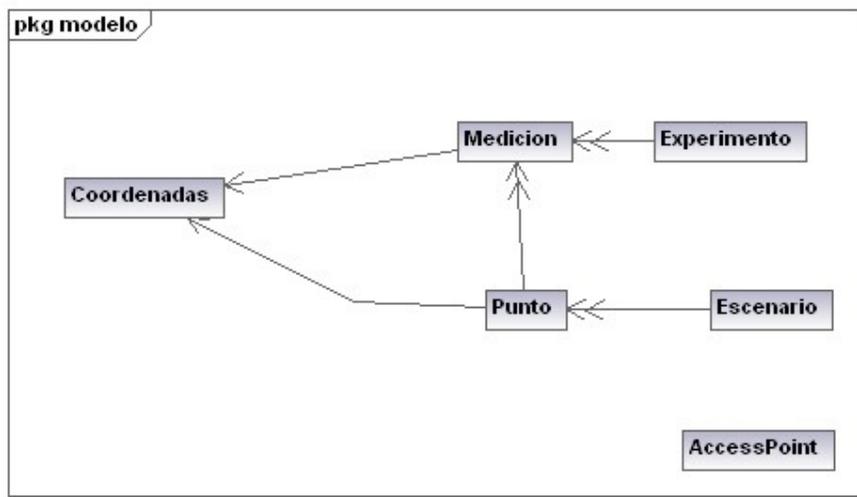


Figura Anexo I 3 – Paquete modelo

Tenemos a continuación el paquete que permite el acceso a la base de datos y la obtención de las medidas mediante la llamada al sistema. El paquete accesoBBDD:

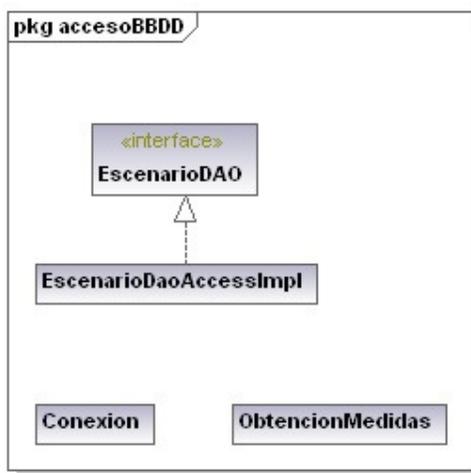


Figura Anexo I 4 – Paquete accesoBBDD

Por último se muestra el paquete `igu`, en el cual se ven las clases que se han desarrollado para la implementación de la interfaz gráfica de usuario, que pese a ser una interfaz sencilla ha resultado tremendamente útil para el desarrollo de las pruebas:

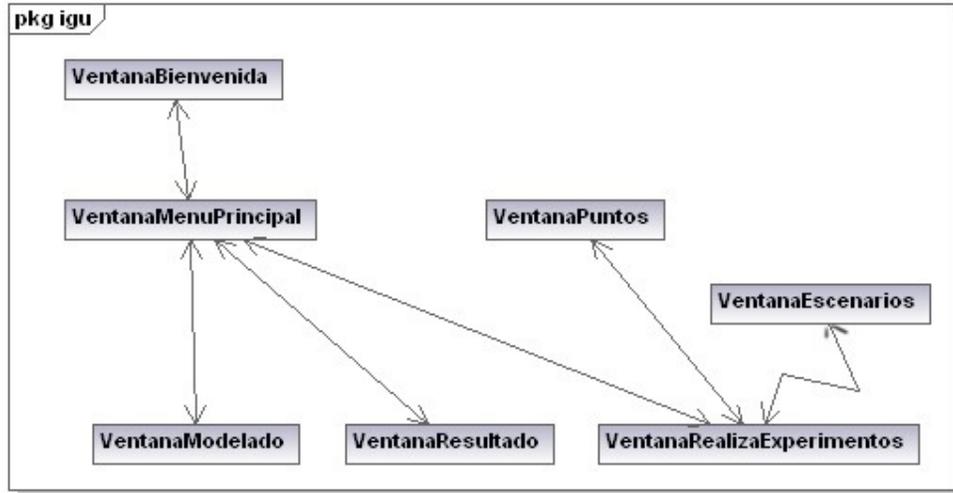


Figura Anexo I 5 – Paquete igu

ANEXO II: Manual de Usuario

El objetivo de este manual de usuario es facilitar la utilización de la aplicación para que de esta forma pueda ser usada de una manera óptima.

Lo primero de todo que se debe de hacer cuando un usuario inicia la aplicación es acceder al menú principal pulsando el botón correspondiente en la ventana de bienvenida.



Figura Anexo II 1 – Bienvenida y botón de entrada

Ahora el usuario ya estará en el menú principal de la aplicación y lo que el usuario querrá hacer si es la primera vez que arranca la aplicación será realizar mediciones y poder agregar escenarios y puntos para posteriormente poder realizar mapeos de datos y localizaciones.

Para entrar a la opción de realizar medidas y poder así agregar escenarios y puntos se deberá seleccionar la primera opción del menú principal.



Figura Anexo II 2 – Menú principal y selección de tomar medidas

Ahora si es la primera vez que se accede a la aplicación se verá que no hay ningún escenario cargado en la barra de la izquierda que es donde se muestran los escenarios. Lo primero que tiene que hacer el usuario es insertar un escenario o seleccionar uno existente. Para agregar un nuevo escenario basta con pulsar el botón derecho en cualquier zona de la barra de la izquierda y se pulse la opción *nuevo*, se rellenan los datos que se solicitan y ya se tendrá el escenario creado.

Una vez que se ha creado o seleccionado un escenario se podrán observar en la parte superior derecha de la aplicación una tabla con los puntos correspondientes a cada escenario. Si no aparece ningún punto se podrán insertar puntos usando el menú de tres botones que aparece en la parte superior de la tabla.

Una vez que se tenga el punto sobre el cuál se quieren tomar mediciones se debe seleccionar el punto en la tabla. Ahora simplemente bastará con pulsar el botón *realizar mediciones* y éstas aparecerán en la tabla que se encuentra justo debajo del botón. Ahora que ya se tienen las mediciones para guardar estas mediciones se debe pulsar el botón *guardar experimento* si se quieren guardar o *limpiar mediciones* si no se quieren guardar.

Ahora bastará con repetir estas operaciones con los escenarios y los puntos que se quiera hasta haber realizado mediciones en todos los sitios que posteriormente se quieran localizar.

A continuación una imagen de la ventana de tomar medidas y las zonas explicadas:

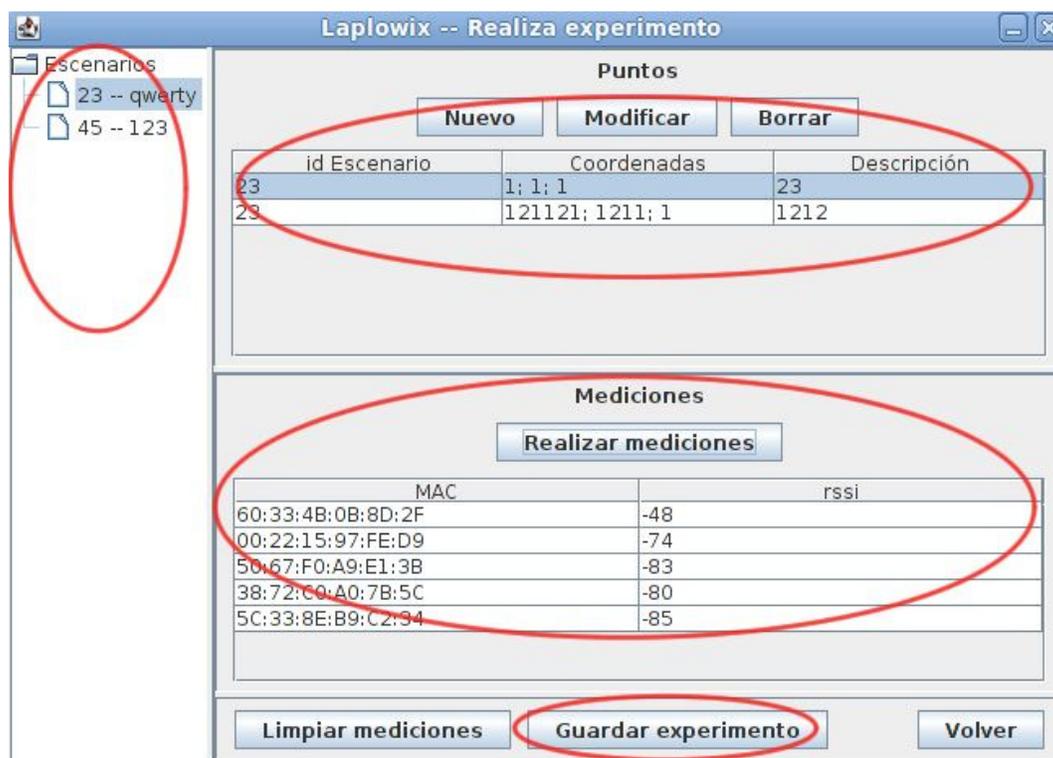


Figura Anexo II 3 – Toma de medidas

Una vez ya se han tomado las medidas o si no es la primera vez que se arranca la aplicación y ya hay medidas, lo que se puede hacer desde el menú principal es acceder a la

parte de la aplicación desde donde se indica que se cree el modelo de datos de referencia que se usará para comparar y tratar de establecer la ubicación del dispositivo. La opción que hay que pulsar en el menú principal es *realizar modelo*, que es la segunda y nos encontraremos con una ventana como la siguiente:



Figura Anexo II 3 – Ventana modelado

Una vez en esta ventana se seleccione el botón que se seleccione el modelo que aplicará será el de la media que es el único implementado, pero si se selecciona *otros* se indicará mediante un popup que es el de media el que se implementará.

Una vez de vuelta en la pantalla de inicio y habiendo realizado ya las tomas de medidas necesarias y el modelo de datos oportuno ya se está en disposición de pulsar sobre la opción *localizar equipo* del menú principal.

Una vez se pulse esta opción aparecerá una ventana que en caso de haber sido capaz de realizar la localización indicará el punto (sus coordenadas) y el escenario (su id) en el que se ha establecido que puede estar el dispositivo. En caso de que no se haya po-

dido localizar, también aparecerá un mensaje indicando que la localización no se ha producido.

Aquí un ejemplo de localización exitosa:

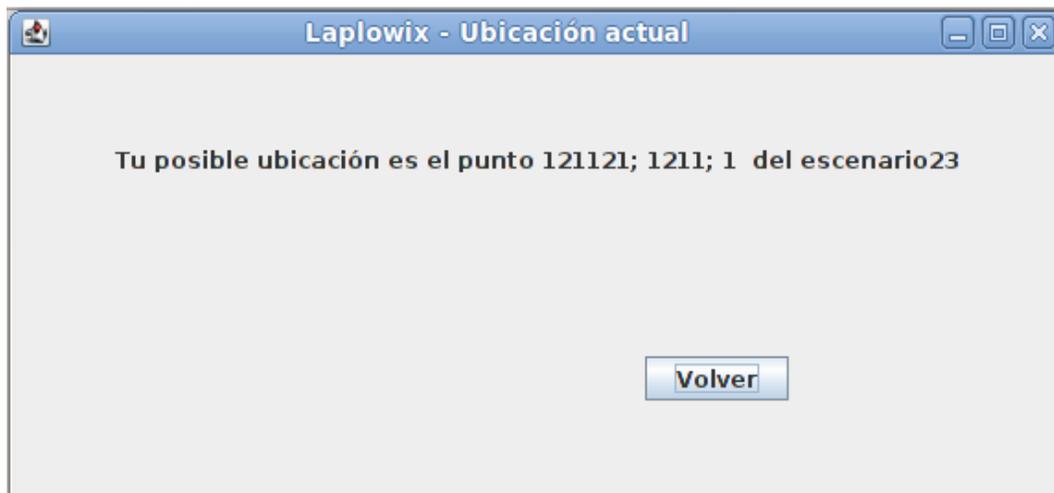


Figura Anexo II 4 – Localización exitosa

ANEXO III: Diagrama de Gantt

A continuación se muestran en 5 imágenes el diagrama de Gantt ampliado:

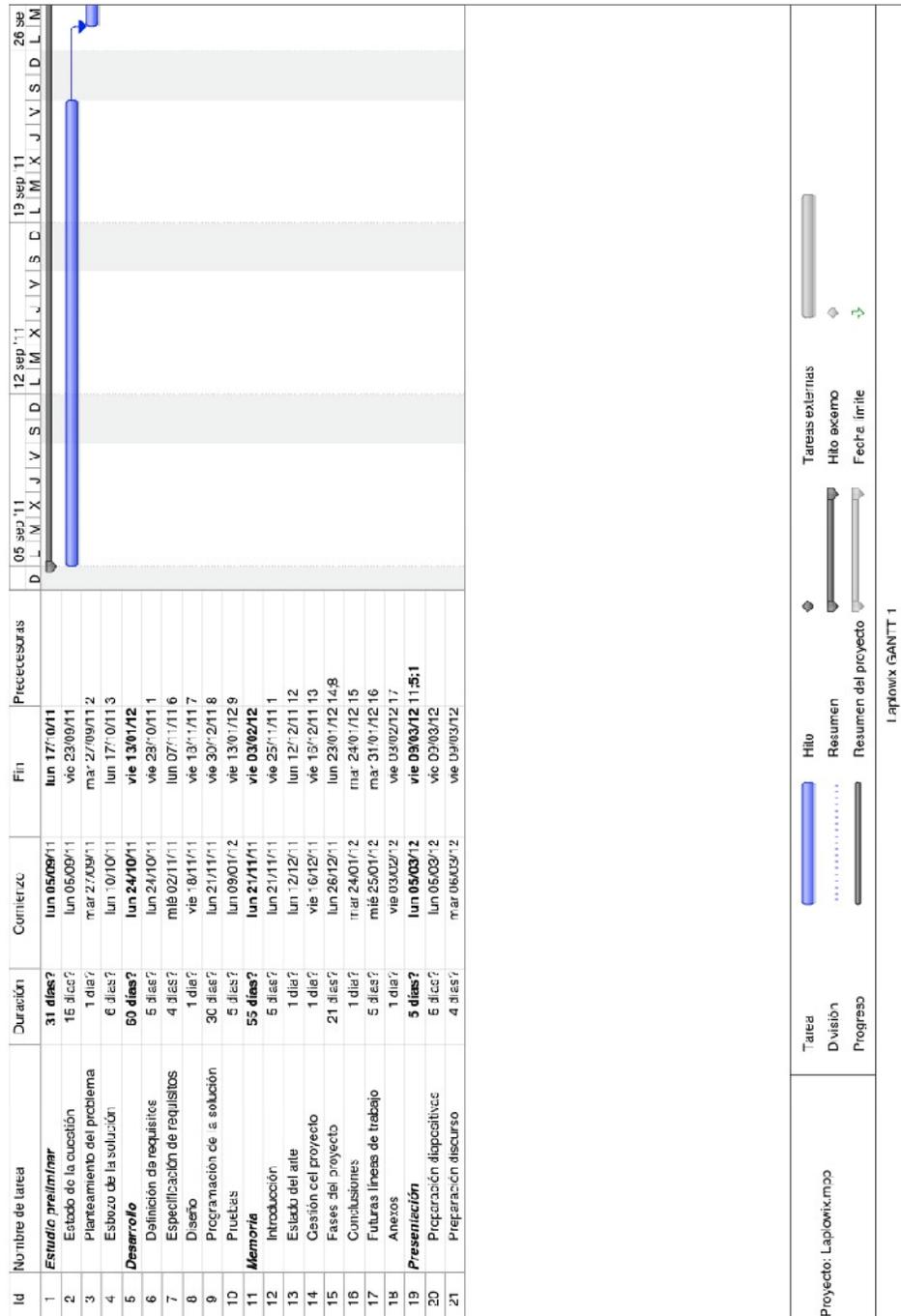


Figura Anexo III 1 – Gantt 1

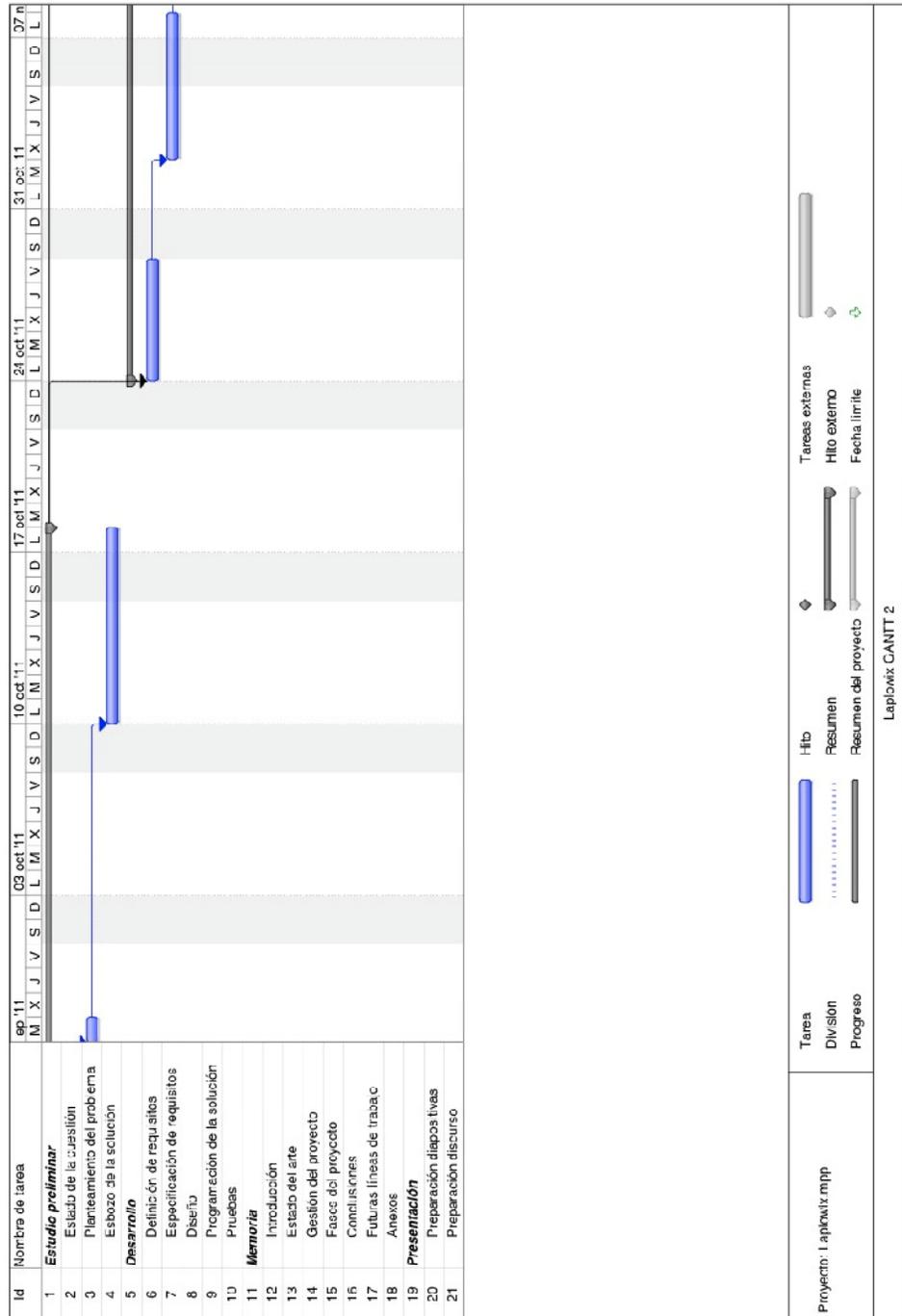


Figura Anexo III 2 – Gantt 2

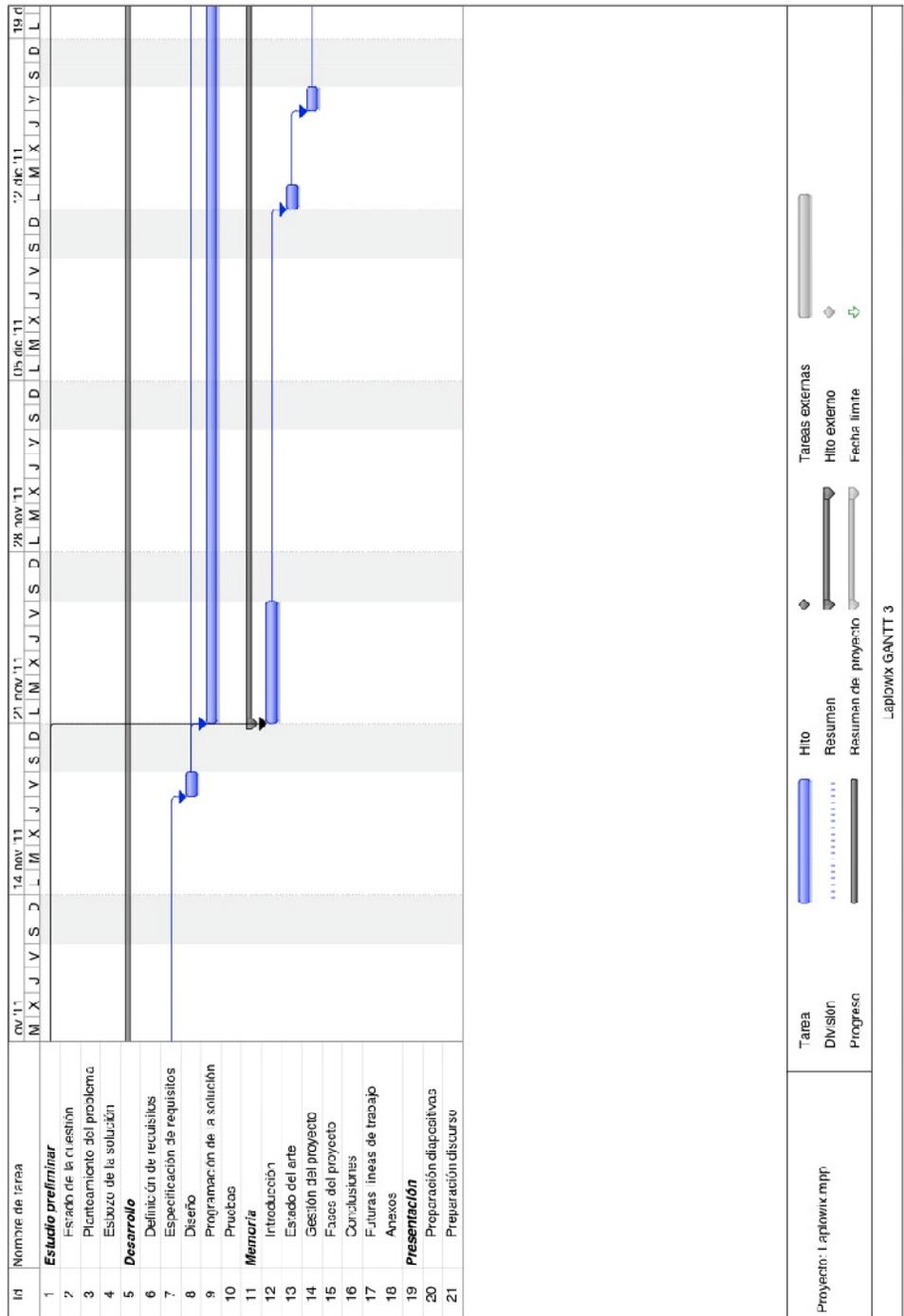


Figura Anexo III 3 – Gantt 3

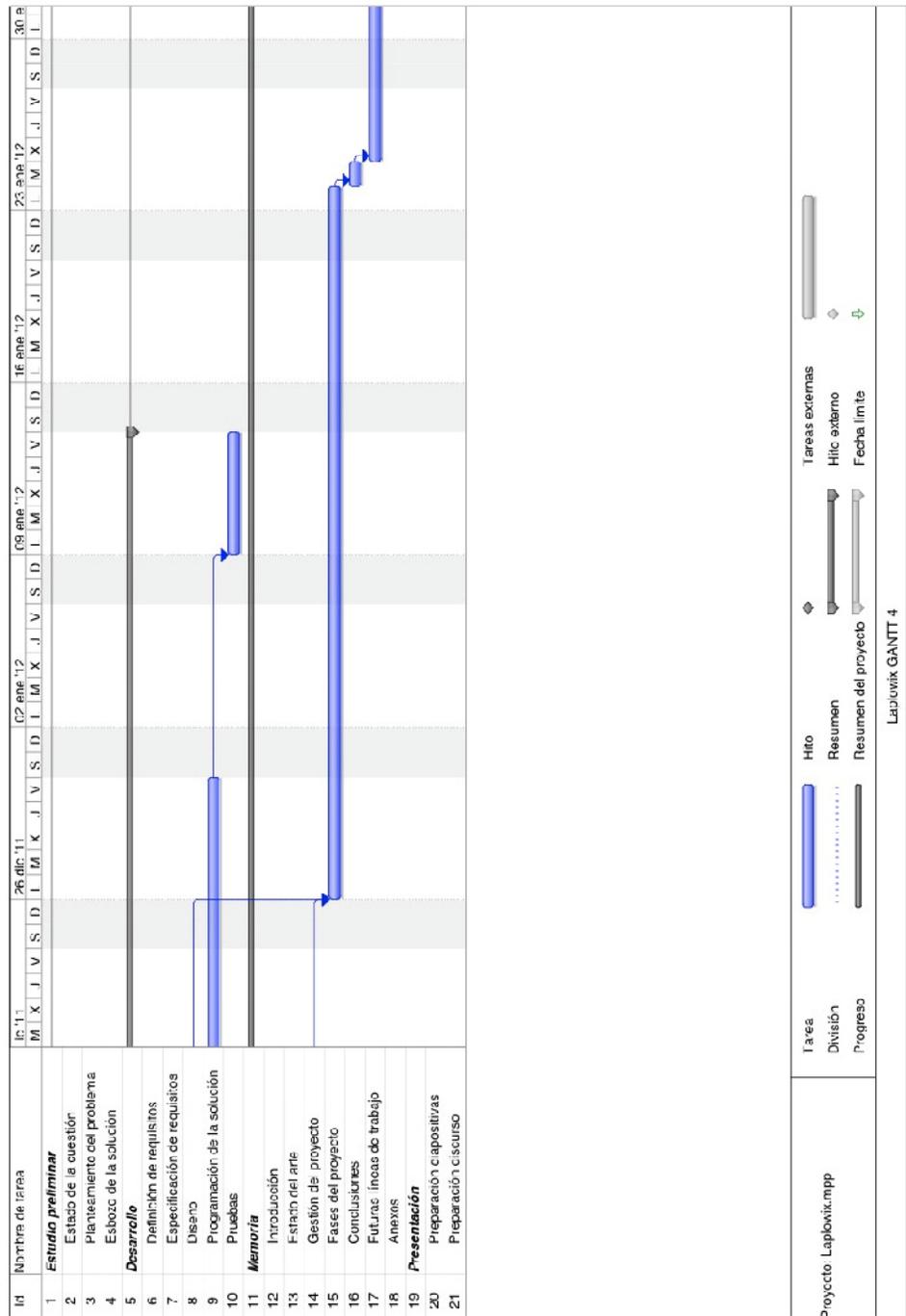


Figura Anexo III 4 – Gantt 4

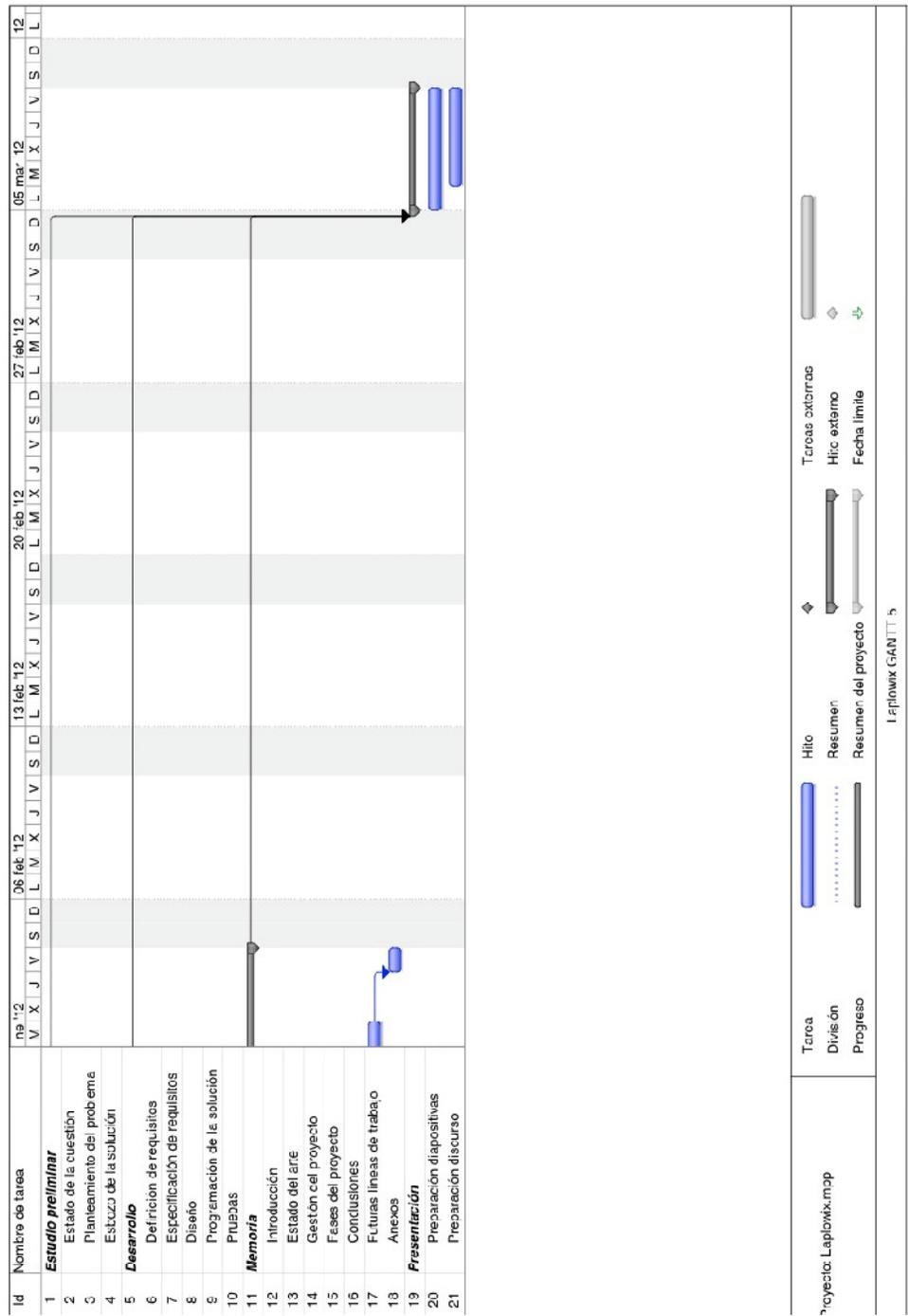


Figura Anexo III 5 – Gantt 5

