



# Sistemas Informáticos

## Curso 2005-2006

---

### Software de comunicación de PDAs

María Alonso Lopez  
Leticia García García  
Jose David Soria Soler

Dirigido por:  
Prof. Manuel Ortega Ortíz de Apodaca  
Dpto. Sistemas Informáticos y Programación

---

Facultad de Informática  
Universidad Complutense de Madrid

# ÍNDICE

1. Autorización
2. Resumen
3. Palabras clave
4. Introducción
5. Especificación de requisitos
  - 5.1 Descripción general
  - 5.2 Requisitos de interfaces externas
  - 5.3 Requisitos no funcionales
  - 5.4 Requisitos funcionales
  - 5.5 Coste económico
6. Casos de uso
  - 6.1 Inicio de sesión
  - 6.2 Impresión de documentos
  - 6.3 Consulta de expedientes
7. Implementación
  - 7.1 Tecnologías
  - 7.2 Diseño
8. Análisis del sistema
  - 8.1 Localización de personal
  - 8.2 Localización de dispositivos
  - 8.3 Conexión PDA-Servidor
  - 8.4 Base de datos
  - 8.5 Red
9. Manual de usuario
10. Herramientas de desarrollo
  - 10.1 Eclipse
  - 10.2 Umbrello
  - 10.3 TexMaker
11. Gestión de configuración
  - 11.1 Reuniones
  - 11.2 Comunicación interna
  - 11.3 Organización de trabajo

12. Valoración

13. Apéndices

13.1 J2ME

13.2 Algoritmo de Floyd

14. Agradecimientos

Referencias

## **1. Autorización**

Por la presente, los autores de este proyecto autorizamos a la Universidad Complutense a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

María Alonso Lopez

Leticia García García

Jose David Soria Soler

## **2. Resumen**

### **2.1. Resumen**

Nuestra aplicación está orientada a dispositivos portátiles, tales como PDAs, móviles, etc. Se trata de una herramienta capaz de localizar, a través de un algoritmo de triangulación con señales inalámbricas, cualquier dispositivo que se encuentre conectado a esa red y a partir de ahí realizar cualquier petición que el usuario requiera. Para ello hemos empleado desde la reciente variante J2ME (Java 2 Micro Edition), hasta las bases de datos más populares (MySQL), pasando por herramientas de desarrollo informático y matemático.

En nuestro caso, hemos enfocado la aplicación hacia un entorno hospitalario: un médico, a través de su PDA, solicitará la impresión de un informe o el muestreo por pantalla de un documento. Entonces el sistema detecta la posición del médico en edificio y a partir de esto, averigua dónde se halla la impresora más cercana, a la que la herramienta manda imprimir el documento solicitado. La aplicación tiene infinitos usos y mejoras, y aunque nosotros nos hemos ceñido a una serie de casos de uso, el campo de aplicación es amplio e incluye casos como el uso de otros dispositivos, por ejemplo, monitores.

### **2.2. Abstract**

Our application is mainly developed to be used with mobile devices, such as PDAs, cellulators etc. It is about a tool that is able to locate, thanks to different mathematics algorithms concerning the localization and triangulation with wireless signals, a device connected to this net structure and so, be able to attend whatever the user asks him to do. We base our job in last generation programming languages, such as J2ME (Java 2 Micro Edition), powerful and free databases (MySQL) and also, computers and mathematics tools. In our case, we decide to develop a case that takes place in a hospital. An hypothetic doctor, who uses a mobile device, wants to get and read a medical report from one of his patients. Thanks to our application, this tool will locate him inside the hospital and then, locate the closest printer and send there the desired report.

We know that our tool has infinite number of uses but, despite we have only care about one of them, we must recognize that the potential field is quite huge.

### **3. Palabras clave**

Palabras claves para su búsqueda bibliográfica (ordenadas por orden alfabético):

- Algoritmo de Floyd.
- Base de datos.
- Caminos mínimos.
- CUPS.
- J2ME.
- MySQL.
- PDA.
- Triangulación.
- UML.
- XML-RPC.

## 4. Introducción

La idea inicial del proyecto, era desarrollar una aplicación similar a la desarrollada por Miguel A. Muñoz, Marcela Rodríguez, Jesús Farela y Ana I. Martínez-García, de Universidad de México, en colaboración con Víctor M. González, de la Universidad de California, para entornos hospitalarios. Dicha aplicación tiene como finalidad la mejora de las condiciones de trabajo del personal hospitalario introduciendo en el hospital la funcionalidad del PDA. Así, el personal dotado de PDAs puede comunicarse con otros compañeros de trabajo, incluso de otros turnos, y tener acceso a datos de sus pacientes con facilidad y rapidez. La aplicación se probó en el Hospital General de Ensenada, México, con resultados satisfactorios. Nuestra aplicación pretende semejarse a ella pero es un poco más limitada.

La aplicación que hemos desarrollado también está pensada para el entorno hospitalario: el PDA debe ser una herramienta de trabajo más para el médico. Las ventajas que puede aportar el uso de PDA a un médico son varias pero destaca la rapidez, en ocasiones, crítica en el entorno que nos movemos. El médico tiene acceso a información de uso frecuente y necesario sin tener que acudir a personal auxiliar del hospital, con el consiguiente ahorro de personal que puede suponer al hospital. Además, el interfaz del PDA es muy sencillo y cualquier persona que esté familiarizada con un teléfono móvil puede entenderla.

Durante este año nos hemos centrado en un caso de uso concreto: el poder proporcionar al médico usuario del PDA la posibilidad de imprimir documentos en la impresora más cercana. Esto es posible a través del algoritmo de triangulación implementado que no sólo sería útil en este terreno, sino que sería aplicable a otros casos de uso, como mostrar información por monitores, etc.

Además de esta facilidad, el médico también tiene la posibilidad de consultar el expediente del paciente al que está atendiendo, visualizándolo en su PDA.

Las posibles ampliaciones de la aplicación son múltiples y variadas. Además, el diseño modular con que la hemos llevado a cabo hace que nuevos casos de uso sean muy fáciles de añadir. El hecho de haber escogido tecnologías libres y multiplataforma también es una ventaja.

La aplicación también tiene la ventaja de que es exportable a otros entornos como pueden ser museos, almacenes, etc.

## **5. Especificación de requisitos**

### **5.1. Descripción general**

Desde el inicio del desarrollo de la aplicación, nos hemos topado con un gran problema, la identificación de los requisitos del sistema. Esto es debido a que no es un proceso que pueda ser determinado matemáticamente. La captura de requisitos son parte esencial del proceso, evita cambios posteriores en el sistema y facilita el entendimiento con el cliente. Los requisitos que debemos contemplar deben proporcionar toda la información necesaria para llevar a cabo un eficiente desarrollo de software.

### **5.2. Requisitos de interfaces externas**

1. Interfaces y hardware destacable.
2. Impresoras de calidad notable para poder visualizar documentos médicos sin dar lugar a errores.
3. Antenas de señal inalámbrica existentes en todo el edificio.
4. La interfaz se adaptará en todo momento al estado de la aplicación.
5. La notificación de errores se hará por medio de un mensaje de texto en mitad de la pantalla.
6. La interfaz será táctil para las PDAs.

### **5.3. Requisitos no funcionales**

1. Seguridad : Todo uso requiere la autenticación de usuarios
2. Facilidad de uso : El usuario será capaz de ver la información de manera clara en la PDA. Se debe ver claramente a una distancia de 30 cm.
3. Fiabilidad : Si se produce algún tipo de fallo al usar la herramienta, el sistema deberá ser capaz de regresar a un estado estable.
4. Restricciones de implementación : La herramienta se desarrollará en lenguaje Java J2ME y accediendo a bases de datos MySQL.

5. Rendimiento : Tanto la operación de localización como la de acceso a la base de datos y su posterior búsqueda, debe poder ofrecer un rendimiento alto.
6. Cuestiones legales : El sistema debe poder ofrecer un marco legal que ofrezca confidencialidad a la hora de tratar los datos de los pacientes.
7. Robustez : La aplicación mostrará una robustez considerable.
8. Tiempo de respuesta : El tiempo de respuesta tiene que ser inferior a 5 segundos.
9. Precio : El precio de la aplicación irá siempre acorde al trabajo depositado en ella.

#### **5.4. Requisitos funcionales**

1. El usuario será capaz de acceder al sistema mediante un nombre de usuario y una contraseña.
2. Si el usuario intenta entrar al sistema con un nombre de usuario o contraseña errónea, el sistema se lo deberá de impedir.
3. El usuario podrá seleccionar entre dos tipos de operaciones a realizar, imprimir un expediente o consultarlo en pantalla.
4. Si el usuario decide consultar un expediente, éste lo verá impreso en su pantalla, de manera organizada, clara y completa.
5. El paciente consultado debe existir previamente en la base de datos.
6. Si el usuario selecciona imprimir, el sistema será capaz de localizar su posición.
7. La localización de la posición implica el empleo de algoritmos matemáticos descritos más abajo.
8. Para poder imprimir, deben existir impresoras en el edificio.
9. Las impresoras deben estar claramente numeradas para así poder indicar al usuario a cual acudir a recoger su petición.
10. La impresión se hará en papel A4, respetando los márgenes establecidos y manteniendo en todo momento una sincronización con lo existente en la base de datos.
11. Deben existir antenas de señal inalámbrica para poder realizar las tareas de localización.

12. El resultado devuelto por el sistema siempre será el más correcto posible.
13. El sistema debe detectar entradas erróneas a la hora de solicitar un servicio por parte del usuario.
14. Si algún recurso funciona mal, este debe ser revisado inmediatamente.
15. Tras realizar una acción, el sistema debe ser capaz de regresar al estado anterior a haber realizado tal acción, y así poder ofrecer de nuevo sus servicios al usuario.
16. Las consultas a la base de datos y su posterior resultado deben ser correctos en todo momento.
17. El sistema debe permitir la entrada de varios usuarios a la vez desde distintas PDAs. Debe favorecer la concurrencia.

## **5.5. Coste económico**

### **5.5.1. Coste de la infraestructura**

Definamos primero lo que va a ser la infraestructura:

La infraestructura es todo aquello que se debe montar y poner en funcionamiento para que la herramienta pueda ejecutarse en las mejores condiciones y sin ningún tipo de problema. Son todos aquellos aparatos, maquinas y demás utensilios que servirán para la correcta implantación de nuestro sistema, obteniendo así un emplazamiento seguro y robusto en cuanto a eficacia y posibles problemas.

Definamos por tanto que herramientas y aparatos debemos instalar para llevar a cabo nuestro objetivo, y del mismo modo establezcamos el presupuesto.

- Puntos de acceso de señal inalámbrica.
  - Descripción : Puntos de acceso inalámbrico con capacidad de transmisión de señales TCP/IP, tanto en intranet como en internet. Serán las encargadas de garantizar una correcta localización de los dispositivos.
  - Número de unidades : Según lugar de instalación. Normalmente serán 4 por planta.
  - Precio : 450 euros por punto.
- Dispositivos tipo PDA para el personal.

- Descripción : Son todos aquellos dispositivos que el personal deberá llevar consigo para poder ser localizados siempre que soliciten un servicio.
  - Número de unidades : Según plantilla.
  - Precio : 300 euros por dispositivo.
- Dispositivos electrónicos de diversa índole.
    - Descripción : Son todos aquellos dispositivos los cuales serán utilizados a la hora de resolver las peticiones del usuario. Estos pueden ser monitores, impresoras, PDAs, móviles etc.
    - Número de unidades : Según estructura del edificio
    - Precio : Entre 100 y 300 euros por dispositivo.

#### **5.5.2. Coste de desarrollo**

El coste del desarrollo cubre el desarrollar y adaptar el sistema a la nueva instalación, a las diferentes topologías existentes en el edificio, así como el mapeo de cada planta en la base de datos para su futura utilización. Del mismo modo, el coste de mantenimiento viene incluido aquí ya que obviamente, ante futuras situaciones de posible ampliación de sus estructuras, nos vemos obligados a mantener nuestro grado de satisfacción del cliente. De este modo, cualquier tipo de obra, construcción o modificación en las instalaciones debe ser comunicado de inmediato para su adaptación al código del programa.

- Precio : 3000 euros.

#### **5.5.3. Coste de personal**

Este es el coste derivado de la mano de obra proporcionada por nosotros. Es el coste de cada uno de las personas que ha estado trabajando en el proyecto.

- Precio detallado : 45 euros / hora trabajada.
- Precio : 6750 euros.

## 6. Casos de uso

### 6.1. Inicio de sesión

#### Objetivo:

Se inicia la sesión con el PDA abriéndose así la aplicación. Al iniciar la sesión será necesario que el usuario introduzca su clave y nombre de usuario, que de ser correctos le darán acceso a los menús.

#### Entradas:

- Clave de usuario.
- Nombre de usuario.

#### Precondiciones:

No hay precondiciones.

#### Salidas:

En caso de éxito (clave y nombre de usuario son correctos):

- Se muestra el menú de la aplicación en la pantalla del PDA.

En caso de fallo (clave y nombre de usuario no son correctos):

- Se informa al usuario de que los datos introducidos no son correctos.
- Se vuelve a la pantalla de inicio para que el usuario tenga la posibilidad de volver a intentar iniciar la sesión.

#### Postcondición si éxito (clave y nombre de usuario son correctos):

Se muestra el menú de la aplicación por la pantalla del PDA.

#### Postcondición si fallo (clave y nombre de usuario no son correctos):

Se informa al usuario de que los datos introducidos no son correctos y se le ofrece la posibilidad de intentar acceder a la aplicación de nuevo.

#### Actores:

- Usuario con PDA.
- Servidor del hospital.

- PDA.

*Secuencia normal:*

1. Al encender el PDA, se solicita al usuario que introduzca su clave y su nombre de usuario y éstos son enviados a la base de datos del servidor del hospital para que sean verificados.
2. El servidor busca en su base de datos la clave y el nombre de usuario que se acaban de introducir para confirmar que está iniciando sesión una persona autorizada, personal del hospital. En caso de que la búsqueda tenga éxito pasar a 3, en caso contrario pasar a E1.
3. Una vez confirmada la clave y el nombre de usuario, en el PDA tenemos disponible el menú con todas las posibles operaciones que se pueden realizar con la aplicación.

*Secuencias alternativas:*

E1.- Se informa al usuario de que los datos que ha introducido no son correctos y se le da la oportunidad de volver a introducirlos (paso 1).

Para ilustrar mejor esta secuencia de acciones, incluimos el diagrama de secuencia de este caso de uso (Figura 14)

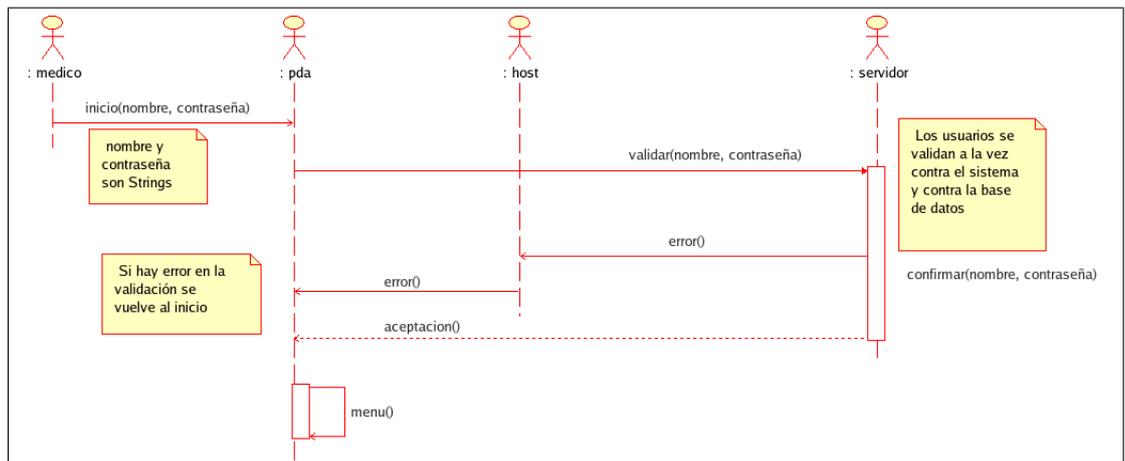


Figura 1: Diagrama de secuencia del inicio de sesión

## 6.2. Impresión de documentos

### Objetivo:

El personal sanitario, dotado de un PDA, solicita la impresión de un documento de un paciente que se encuentre disponible en la base de datos del hospital. Dicha impresión se realizará por la impresora más cercana al lugar en el que se ha solicitado ésta.

### Entradas:

- Documento a imprimir.
- Nombre del paciente del que solicitamos el documento a imprimir.

### Precondiciones:

Los documentos que se solicitan imprimir deben encontrarse en la base de datos del hospital así como el paciente, que también debe estar registrado en la base de datos.

### Salidas:

En caso de éxito:

- Confirmación de la solicitud aceptada.
- Impresora por la que se va a realizar el trabajo.
- Documento impreso en la impresora indicada.

En caso de fallo:

- Se informa al usuario de la imposibilidad de realizar la petición.

### Postcondición si éxito:

Se realiza la impresión por la impresora más cercana.

### Postcondición si fallo:

Se informa de que no se ha podido realizar la impresión y se vuelve al menú principal.

### Actores:

- Usuario con PDA.
- Servidor del hospital.

- Cola de impresión de las impresoras del centro.
- PDA.

Secuencia normal:

1. El usuario solicita al servidor la impresión de un documento concreto de un paciente del hospital desde su PDA. Si error al conectar con el servidor pasa a E1, sino pasa a 2.
2. El servidor busca en su base de datos el documento que queremos imprimir. Si lo encuentra pasa a 3, sino pasa a E1.
3. El servidor calcula la posición del PDA y en función de ella devuelve la impresora más cercana, la elegida para imprimir el documento.
4. El servidor manda el documento a la cola de la impresora más cercana.
5. El servidor envía al PDA la confirmación de impresión aceptada, informando de cual es la impresora que en la que se realiza la tarea.
6. Volvemos al menú principal.

Secuencias alternativas:

E1.- Se informa al usuario de que no se puede realizar la impresión en ese momento y se le invita a intentarlo más tarde. Volvemos al menú principal.

Para ilustrar mejor esta secuencia de acciones, incluimos el diagrama de secuencia de este caso de uso (Figura 2)

### **6.3. Consulta de expedientes**

Objetivo:

Visualización por la pantalla del PDA del expediente de un paciente del hospital.

Entradas:

- Nombre del paciente del que queremos consultar el expediente.

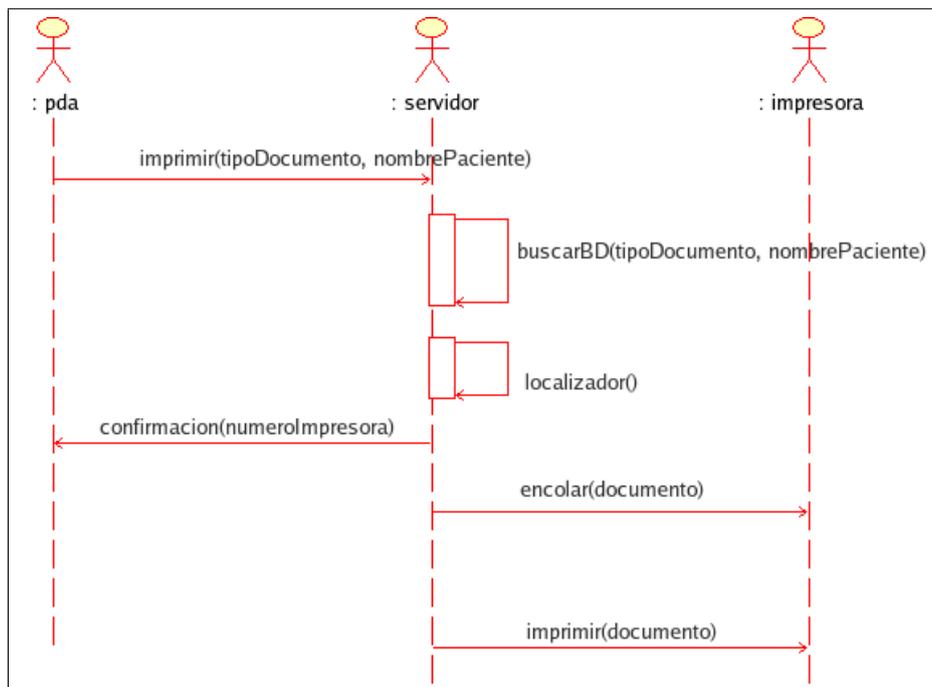


Figura 2: Diagrama de secuencia de impresión de documentos

Precondiciones:

El propietario del expediente que queremos visualizar debe ser un paciente registrado en la base de datos del hospital.

Salidas:

En caso de éxito (el nombre paciente se encuentra en la base de datos del hospital):

- Se muestra en la pantalla del PDA el expediente del paciente solicitado.

En caso de fallo (el nombre paciente no se encuentra en la base de datos del hospital):

- Se informa al usuario de la imposibilidad de visualizar la información solicitada.
- Se vuelve al menú principal.

Postcondición si éxito (el nombre paciente se encuentra en la base de datos del hospital):

Se muestra el expediente del paciente solicitado en la pantalla del

PDA.

Postcondición si fallo (el nombre paciente no se encuentra en la base de datos del hospital):

Se informa al usuario de la imposibilidad de visualizar los datos solicitados y se vuelve al menú principal.

Actores:

- Usuario con PDA.
- Servidor del hospital.
- PDA.

Secuencia normal:

1. El usuario solicita al servidor la consulta del expediente de un paciente del hospital desde su PDA.
2. El servidor busca en su base de datos el expediente del paciente solicitado. En caso de que la búsqueda tenga éxito pasar a 3, en caso contrario pasar a E1.
3. Se hace una lectura del expediente solicitado y se envía al PDA, mostrándose por pantalla.
4. Volvemos al menú principal.

Secuencias alternativas:

E1.- Se informa al usuario del error en la consulta y se vuelve al menú principal.

Para ilustrar mejor esta secuencia de acciones, incluimos el diagrama de secuencia de este caso de uso (Figura 3)

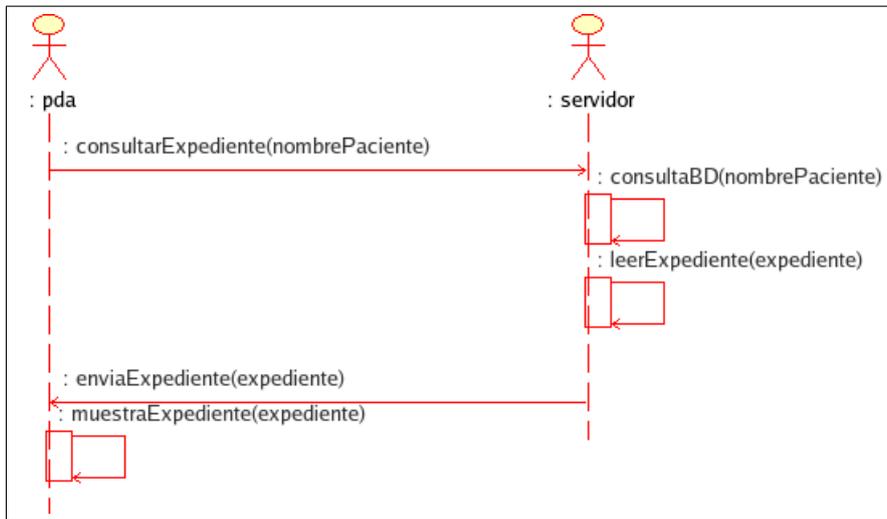


Figura 3: Diagrama de secuencia de consultar expediente

## 7. Implementación

### 7.1. Tecnologías

- J2ME:

Java 2 Micro Edition (J2ME), es un subconjunto de J2SE orientado al desarrollo de aplicaciones Java destinadas a dispositivos con capacidades restringidas, tanto con respecto a la capacidad de memoria disponible, limitaciones de memoria gráfica como con respecto a la capacidad de procesamiento, en nuestro caso, un PDA.

Hay varias formas de implementar aplicaciones del tipo que estamos presentando.

Por un lado, tenemos las aplicaciones ligeras, basadas en el mundo web. Utilizan lenguajes como WML o HTML para el desarrollo de contenidos y emplean el protocolo HTTP para comunicarse con un servidor que proporciona el contenido de el cliente debe presentar al usuario. El principal inconveniente de estas aplicaciones es que el usuario tiene que estar continuamente conectado.

Por otro lado, están las aplicaciones nativas, que son aquellas desarrolladas para un sistema operativo determinado como PalmOS, Windows CE o EPOC y están escritas en lenguaje C/C++ o incluso BASIC. Estas aplicaciones se pueden ejecutar de forma autónoma, sin necesidad de conexión, pero son totalmente dependientes del sistema operativo para el que han sido creadas.

Finalmente, J2ME auna lo mejor de los dos mundos y salva sus principales inconvenientes al ser multiplataforma, importante inconveniente de las aplicaciones nativas, y al proporcionar todos los elementos de Java, como control de la persistencia, conexión a la red o interfaz de usuario, que no están disponibles en las aplicaciones ligeras.

- XML-RPC:

Dada la naturaleza de la aplicación, se hacía necesario el empleo de un protocolo de intercambio de información entre el PDA y el servidor del hospital. El protocolo XML-RPC es un protocolo extremadamente ligero para la invocación de procesos

remotos sobre una red enviando mensajes XML formateados sobre protocolo HTTP.

Aunque la especificación de J2ME no proporciona soporte nativo para XML-RPC, el proyecto de código abierto kXML-RPC es una implementación de XML-RPC destinada a dispositivos compatibles MIDP, basada en Kxml , con lo que empleando estas librerías ya teníamos solucionado el intercambio de información.

Otra posibilidad era emplear el protocolo Simple Object Access Protocol (SOAP). SOAP es un protocolo ligero de intercambio de información estructurada en un entorno descentralizado y distribuido. La idea subyace en la creación de este protocolo es proporcionar un mecanismo uniforme para realizar llamadas a procedimientos remotos utilizando HTTP como protocolo de comunicación y XML como mecanismo de serialización de datos.

XML-RPC y SOAP son protocolos muy similares pero SOAP ofrece una mayor versatilidad a cambio de una mayor sobrecarga. A la hora de la elección teníamos que tener presente la escasez de memoria de los dispositivos compatibles J2ME, que el poder de procesamiento no es abundante y que el ancho de banda disponible en los dispositivos móviles no es demasiado grande y dado que SOAP proporciona propiedades extra que no eran necesarias para la aplicación, la elección automática fue XML-RPC.

- MySQL:

En la actualidad, hay muchos sistemas gestores de bases de datos, por ejemplo, Oracle o Microsoft Access. MySQL, es el sistema que hemos escogido para gestionar la base de datos del hospital. El motivo que nos ha impulsado a ello es que está desarrollado bajo la filosofía de código abierto y que es multiplataforma. Además, J2SE cuenta con el paquete java.sql que proporciona numerosas utilidades para hacer consultas a bases de datos MySQL desde aplicaciones Java, lo que hace muy fácil su integración con el programa servidor del hospital.

- Ubuntu Server:

En el servidor hemos instalado Ubuntu server. Una variedad de

Ubuntu para servidores. Ubuntu es una distribución de Linux, basada en Debian. Sus principales características son:

1. Se en la distribución Debian.
2. Est disponible para Intel x86, AMD64, PowerPC.
3. El sistema incluye funciones avanzadas de seguridad y entre sus políticas se encuentra el no activar procesos latentes por omisión al momento de instalarse. Por eso mismo, no hay un firewall predeterminado, ya que no existen servicios que puedan atentar a la seguridad del sistema.

Ubuntu divide todo el software en cuatro secciones:

1. main
2. estriected
3. universe
4. multiverse

■ CUPS:

Como servidor de impresión hemos usado Common Unix Printing System (CUPS). Es un sistema de impresión modular para sistemas de operativos unix. Permite que un computador actúe como servidor de impresión. Un computador que ejecuta CUPS actúa como un servidor que puede aceptar tareas de impresión desde otros computadores clientes, los procesa y los envía al servidor de impresión apropiado.

CUPS está compuesto por una cola de impresión con su programación, un sistema de filtros que convierte datos para imprimir hacia formatos que la impresora conozca, y un sistema de soporte que envía los datos al dispositivo de impresión. CUPS utiliza el protocolo IPP como base para el manejo de tareas de impresión y de colas de impresión. También provee los comandos tradicionales de impresión de los sistemas Unix y un soporte limitado de operaciones bajo el protocolo SMP. Los programas de manejo de dispositivo de impresión que CUPS provee están basados en la Descripción de impresoras PostScript (PPD). Existen varias interfaces de usuario para diferentes plataformas para configurar CUPS, incluso una interfaz Web. CUPS se distribuye bajo licencia GNU General Public License y GNU Lesser General Public License, Versin 2.

## 7.2. Diseño

El diseño de la aplicación es extremadamente modular. Hay dos módulos claramente diferenciados: por un lado tenemos la aplicación residente en el PDA y por otro la aplicación que ejerce de servidor, alojada en el ordenador central del hospital, donde también se aloja la base de datos.

En la figura 4, mostramos el diagrama de clases del MIDlet que implementa la aplicación del PDA. Consta de un único MIDlet, PdaMIDlet, y tiene varias clases auxiliares, PantallaInicio, MenuPrincipal, MenuImprimir y ConsultaExpediente, con las que se implementan las distintas pantallas que se muestran en la aplicación. Desde PdaMIDlet se llama a la primera pantalla (PantallaInicio) y ellas se comunican entre sí a través de métodos de PdaMIDlet.

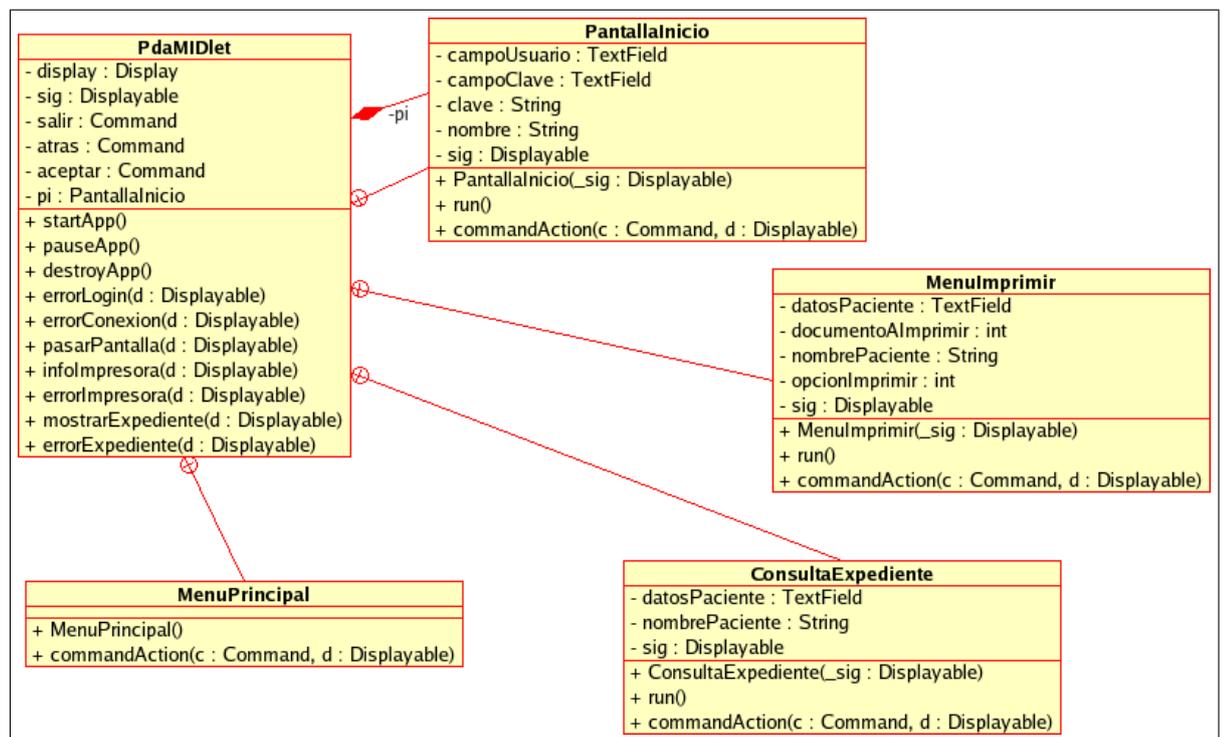


Figura 4: Diagrama de clases de la aplicación residente en el PDA

En el servidor tenemos varias partes:

- Base de datos
- Localizador de personal: triangulador

- Localizador de dispositivos: mapa

Tenemos clases que aportan tipos de datos como:

- Antena: modela los puntos de acceso
- Punto: modela las posiciones del personal en el hospital
- Nodo: modela los nodos del grafo de salas

Además está la clase `JavaServer` que contiene la parte de la aplicación que recibe las llamadas de la PDA. En las figuras 5 y 6 se muestra el diagrama de clases de la aplicación residente en el servidor.

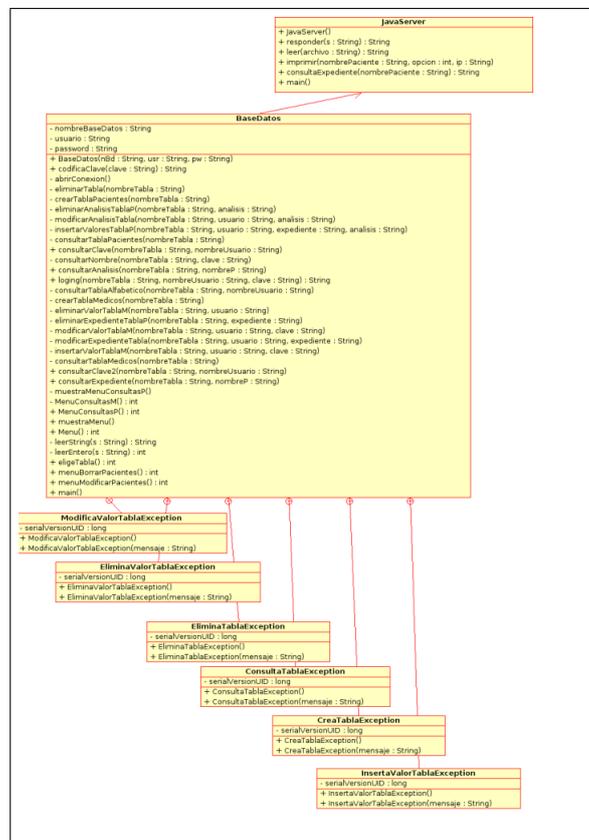


Figura 5: Diagrama de clases de la aplicación residente en el Servidor: la base de datos

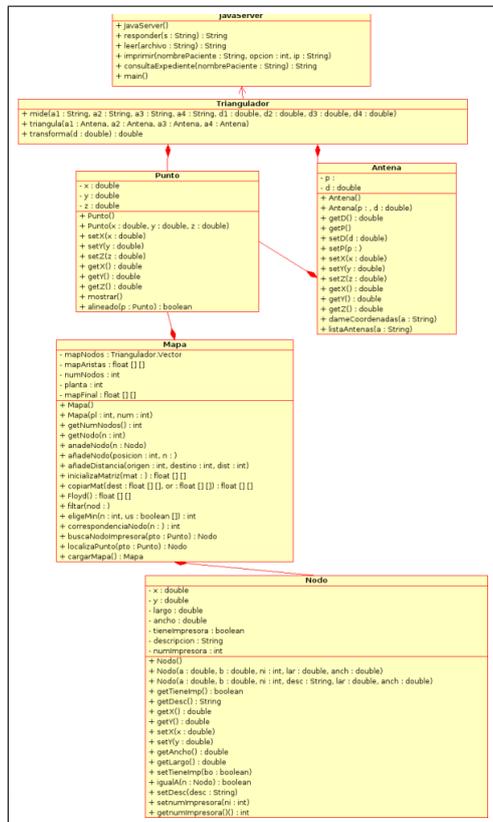


Figura 6: Diagrama de clases de la aplicación residente en el Servidor (sin base de datos)

## 8. Análisis del sistema

### 8.1. Localización de personal

Este es es diagrama de secuencia de la llamada a la localización (figura 7).

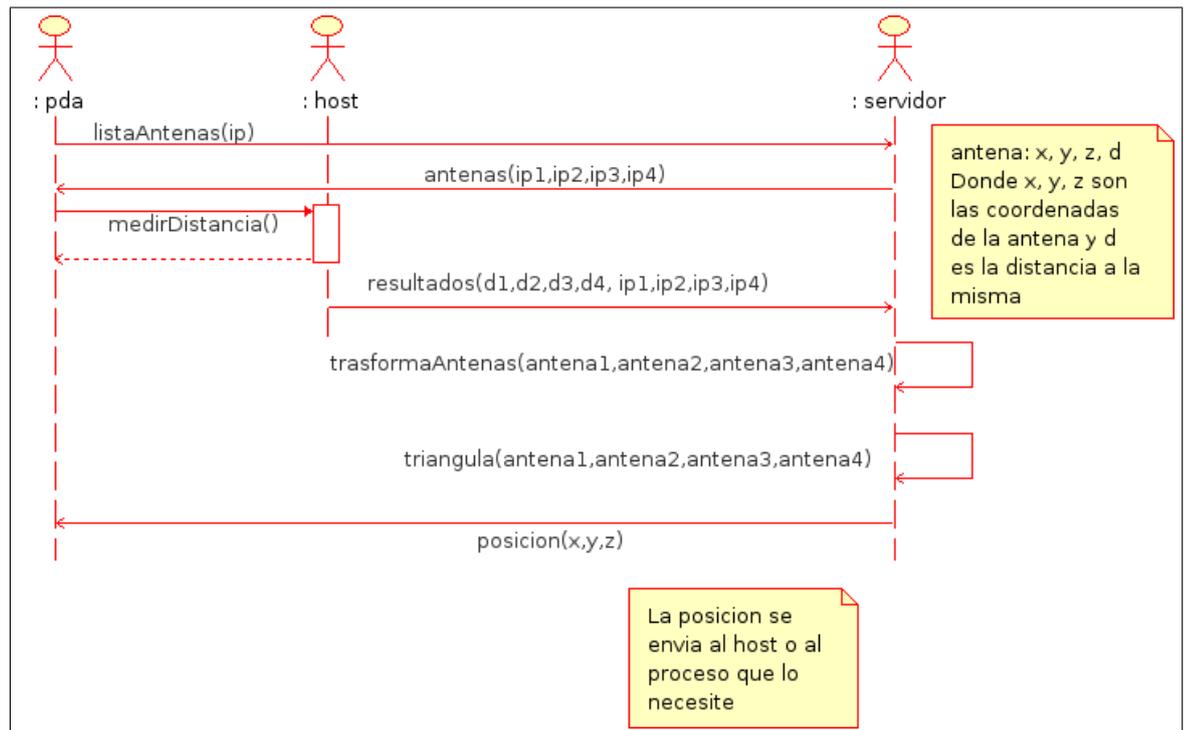


Figura 7: Secuencia de localización

El proceso de localización consta de dos partes:

- Medir las distancias a los puntos de acceso.
- aplicar el algoritmo de triangulación.

Para medir las distancias hay que tener en cuenta la distribución de los puntos de acceso formando cuadrados. Pedimos al servidor los puntos que estén más próximos. Medimos la distancia al punto al que nos conectamos (punto A) con una llamada a ping. A continuación hacemos ping al segundo punto de la lista (punto B). Como sabemos la distancia entre los dos puntos y la distancia de la PDA (punto C) al punto A, solo tenemos que aplicar reglas geométricas para conocer la distancia del punto B al C, ya que forman un triángulo. Repetimos este proceso para los otros dos puntos de la

lista.

La llamada a la función triangula se encarga de la localización de la PA una vez que se tienen medidas las distancias a los puntos de acceso.

La localización corresponde a la resolución del sistema de ecuaciones que se define por el punto de intersección de tres esferas con centro en las coordenadas del punto de acceso y radio la distancia al dispositivo inalámbrico a localizar. Si los puntos de acceso son:

1. Punto 1:

a) Coordenadas:  $(x_1, y_1, z_1)$

b) Distancia:  $d_1$

2. Punto 2:

a) Coordenadas:  $(x_2, y_2, z_2)$

b) Distancia:  $d_2$

3. Punto 3:

a) Coordenadas:  $(x_3, y_3, z_3)$

b) Distancia:  $d_3$

4. Punto 4:

a) Coordenadas:  $(x_4, y_4, z_4)$

b) Distancia:  $d_4$

Para localizar el punto  $(x, y, z)$  tenemos que resolver:

1. ecuación 1:  $(x - x_1) + (y - y_1) + (z - z_1) = d_1$

2. ecuación 2:  $(x - x_2) + (y - y_2) + (z - z_2) = d_2$

3. ecuación 1:  $(x - x_3) + (y - y_3) + (z - z_3) = d_3$

4. ecuación 1:  $(x - x_4) + (y - y_4) + (z - z_4) = d_4$

El cuarto punto lo usaremos para comprobar la corrección de la resolución o para reemplazar a uno de los anteriores si están alineados. En otro caso no es necesario porque sabemos que el punto a localizar está dentro del edificio.

La resolución analítica directa genera denominadores del tipo  $x_1 - x_2$ , esos denominadores se anulan cuando las antenas están

en las esquinas de un cuadrado. Esa situación es frecuente así que es necesario realizar una serie de translaciones y rotaciones para garantizar que los denominadores no se anulan. Hacemos que el origen del sistema de coordenadas esté en el punto 1, el eje OX pase por el punto 2 y el plano XY pase por el punto 3. Ver figura 8.

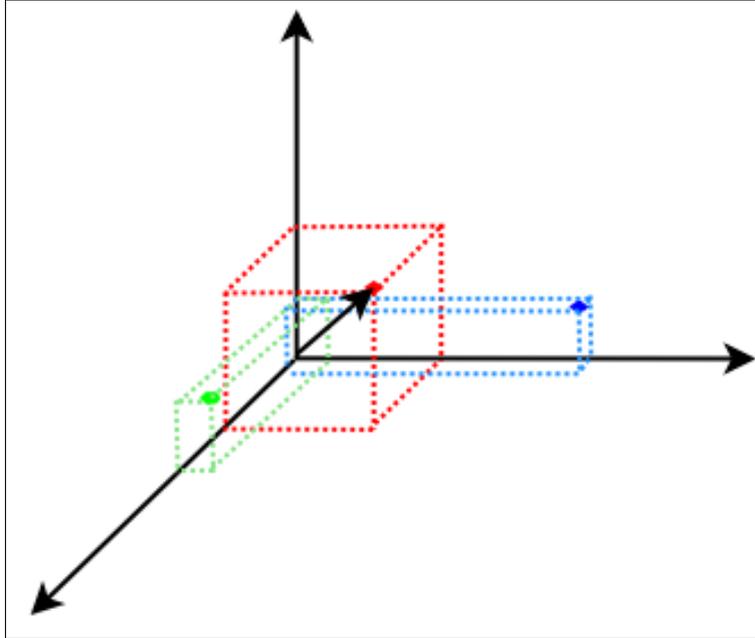


Figura 8: Distribución inicial de los puntos

El primer paso es trasladar el origen al punto 1  $(x_1, y_1, z_1)$  desde el origen  $(0, 0, 0)$ . El vector de translación es  $V = (v_1, v_2, v_3) = (x_1, y_1, z_1) - (0, 0, 0)$ . Aplicamos la rotación a los tres puntos. Ver figura 9.

Rotamos los ejes para trasladar el eje OX y que pase por el punto 2. Ver figura 10.

Es necesario realizar dos rotaciones. La primera rotación se corresponde al ángulo  $a$ :  $a = \arccos \frac{x_2}{\sqrt{x_2^2 + z_2^2}}$ . El eje OX se alinea con la proyección del punto 2 sobre el plano XZ. La nuevas coordendas del punto 2 son:

1.  $x'_2 = x_2 \sen a + z_2 \cos a$
2.  $z'_2 = -x_2 \sen a + z_2 \cos a$

La segunda rotación se corresponde al ángulo  $b$ :  $b = \arccos \frac{x_2}{\sqrt{x_2^2 + y_2^2}}$ . El eje OX se alinea con el punto 2. La nuevas coordendas del punto 2 son:

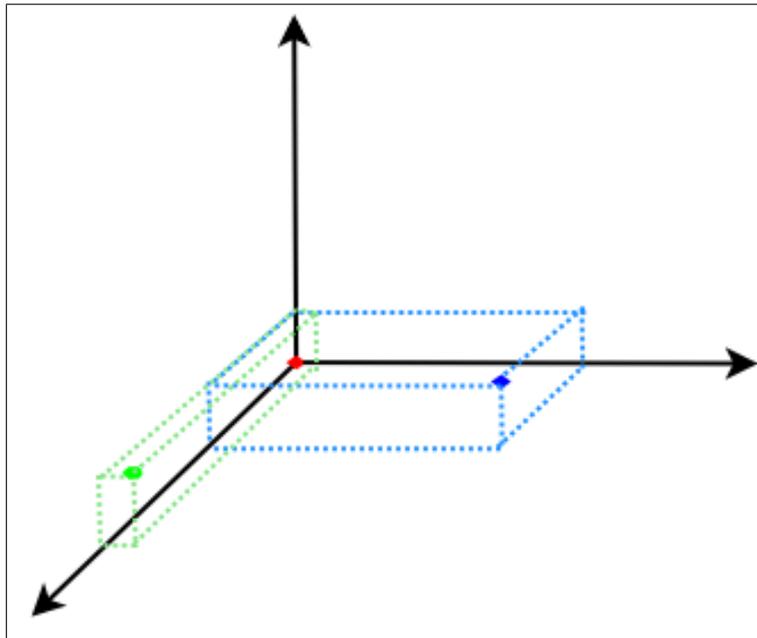


Figura 9: Distribución de los puntos tras la traslación

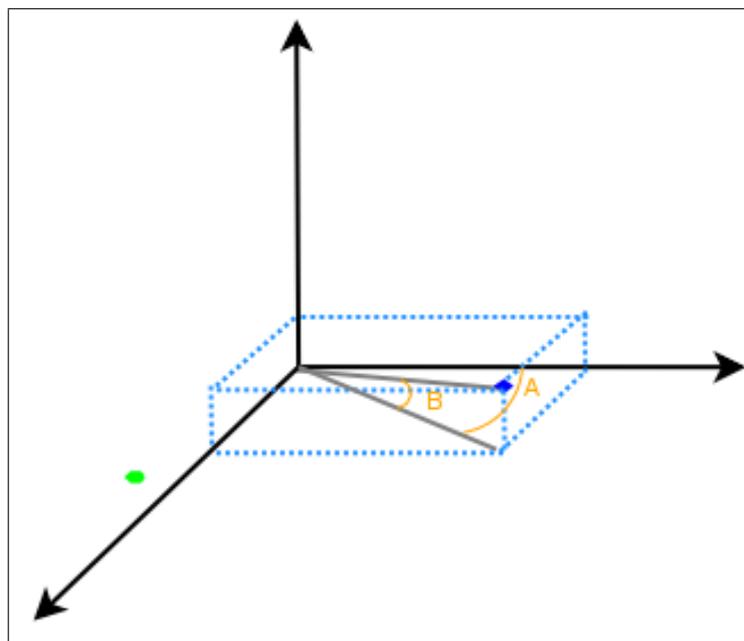


Figura 10: Ángulos de rotación

1.  $x'_2 = x_2 \text{ sen } b + y_2 \text{ cos } b$

$$2. y'_2 = -x_2 \operatorname{sen} b + y_2 \operatorname{cos} b$$

Las rotaciones también afectan a los puntos 1 y 2. En el caso del punto 1 no es necesario realizar ninguna operación pues se corresponde al origen. Los resultados para el punto 3 son análogos.

Rotamos los ejes para que el plano XY y que pase por el punto 3.  
3. Ver figura 11.

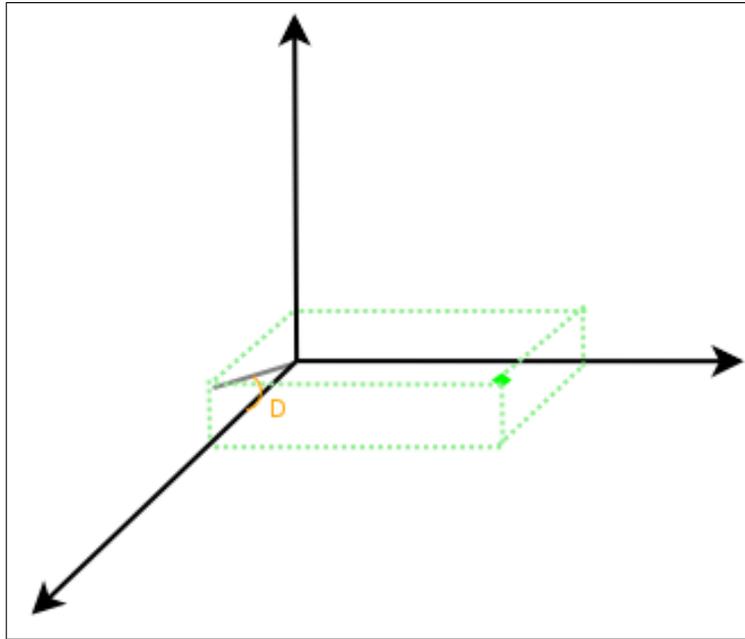


Figura 11: Ángulos de rotación

La rotación se corresponde al ángulo  $d$ :  $d = \operatorname{arc} \cos \frac{y_3}{\sqrt{z_3^2 + y_3^2}}$ . El plano XY se alinea con la proyección del punto 3 sobre el plano YZ.

Tras la resolución analítica es necesario deshacer estas modificaciones.

1. Dehacemos la tercera rotación  $-d$  grados:

$$d = \operatorname{arc} \cos \frac{y_3}{\sqrt{z_3^2 + y_3^2}}$$

2. Dehacemos la segunda rotación  $-b$  grados:

$$b = \operatorname{arc} \cos \frac{x_2}{\sqrt{x_2^2 + y_2^2}}$$

3. Dehacemos la primera rotación -a grados.

$$a = \arccos \frac{x_2}{\sqrt{x_2^2 + z_2^2}}$$

4. Transladamos -V.

$$V = (x_1, y_1, z_1)$$

## 8.2. Localización de dispositivos

### 8.2.1. Algoritmo de Floyd

Para llevar a cabo la tarea de localizar la impresora más cercana al la posición del usuario-cliente de la PDA, debmos decantarnos por un algoritmo informático que nos facilitara la búsqueda y localización del dispositivo más próximo. Se trata de un algoritmo de coste muy eficiente, ya que itera de manera sucesiva por todos y cada uno de los nodos presentes en el mapa, y va actualizando las distancias entre ellos, a pesar de que en un principio no hubiese un camino directo entre ellos. De esta manera, siempre que la distancia encontrada entre dos nodos, utilizando otro del mapa como nodo auxiliar, sea mas pequeño que la hallada hasta ese momento, el algoritmo se encarga de asignar ese resultado a la distancia actual, minimizándose de esta manera la distancia y el camino a ese nodo. No podemos asegurar que todos los nodos sean accesibles desde todos los demás pero si que podemos asegurar que si hay un camino, este será mínimo.

### 8.2.2. Filtrado de dispositivos

Una vez conseguido todo el mapa de la planta, con caminos mínimos entre todos los nodos, lo que resta por hacer será encontrar aquél nodo, que siendo el más cercano a la posición actual del usuario-cliente de la PDA, este contenga una impresora o dispositivo capaz de satisfacer las necesidades de la persona. Así lo que estamos haciendo es iterando sobre cada nodo, de manera ordenada, comenzando por el nodo más cercano y comprobando si este contiene o no ese dispositivo. Si no lo tiene, seguimos iterando pero si lo posee, lo guardamos y pasamos a resolver la correspondencia entre las coordenadas devueltas y la posición real dentro del edificio.

### 8.2.3. Correspondencia coordenadas-posición real

Llegado hasta aquí solo nos queda resolver de manera analítica, cual es la posición dentro del edificio, siguiendo el mapa de habita-

ciones y pasillos, de las coordenadas devueltas por el anterior paso. Esto se resuelve de manera sencilla, ya que tenemos almacenado en la base de datos las dimensiones de cada una de las habitaciones, según la planta, y de esta manera podemos ver "dentro" de que habitación ha caído la posición devuelta como resultado.

Acto seguido, lo único que queda es comunicarle al usuario-cliente a que habitación del edificio (o igualmente a que dispositivo) debe acudir para recoger su solicitud.

### 8.3. Conexión PDA-Servidor

Las llamadas remotas a procedimientos del servidor del hospital (login, consultas a bases de datos...) que se hacen desde el PDA se llevan a cabo a través del protocolo XML-RPC.

Un mensaje XML-RPC se incrusta en una petición POST enviada por el PDA. El servidor del hospital recibe la petición, analiza el documento XML, ejecuta el procedimiento con los parámetros adecuados y devuelve el resultado como un documento XML formateado al PDA mediante respuesta HTTP.

Para ejemplificar este proceso, transcribimos los respectivos mensajes que se pasan servidor y PDA en el procedimiento de login, cuando queremos iniciar sesión con el PDA.

Primero el PDA envía el nombre de usuario y la clave personal que ha introducido el médico al servidor del hospital para que contraste estos datos con la base de datos.

```
<methodCall>
  <methodName>bd.login</methodName>
  <params>
    <param>
      <value>
        <string>tablaMedicos</string>
      </value>
    </param>
    <param>
      <value>
        <string>HOUSE</string>
      </value>
    </param>
    <param>
```

```

    <value>
      <string>STACY</string>
    </value>
  </param>
</params>
</methodCall>

```

Una vez que éste ha ejecutado el procedimiento solicitado, el servidor envía la respuesta con el resultado al PDA:

```

<methodResponse>
  <methodName>bd.login</methodName>
  <params>
    <param>
      <value>
        <string>si</string>
      </value>
    </param>
  </params>
</methodResponse>

```

#### 8.4. Base de datos

Dado los escasos casos de uso que se han implementado a lo largo del proyecto, no se consideró necesario una base de datos muy sofisticada. Así creamos una base de datos "Hospital", con dos tablas, una para datos del personal del hospital, "tablaMedicos", y otra para datos de pacientes, "tablaPacientes".

```

+-----+
| Tables_in_Hospital |
+-----+
| tablaMedicos       |
| tablaPacientes     |
+-----+

```

La tabla "tablaMedicos" tiene dos entradas, usuario y clave, correspondientes con los datos que tienen que introducir los médicos cada vez que quieren iniciar sesión en el PDA:

```

+-----+-----+
| USUARIO | CLAVE |
+-----+-----+
| HOUSE   | STACY |
| CAMERON | ISLA  |
| FOREMAN | CARCEL|
+-----+-----+

```

La tabla "tablaPacientes" tiene tres entradas correspondientes con el nombre del paciente, la ruta en la que se encuentran almacenados en el servidor del hospital sus últimos análisis y su expediente:

```

+-----+-----+-----+
| NOMBRE | ANALISIS | EXPEDIENTE |
+-----+-----+-----+
| LUISITO | /home/maria/Analisis/analisisLuisito.txt | /home/maria/Expedientes/expedienteLuisito.txt |
| MANOLIN | /home/maria/Analisis/analisisManolin.txt | /home/maria/Expedientes/expedienteManolin.txt |
| PAQUITO | /home/maria/Analisis/analisisPaquito.txt | /home/maria/Expedientes/expedientePaquito.txt |
+-----+-----+-----+

```

El servidor MySQL está continuamente funcionando en el servidor del hospital de manera que en cualquier momento son posibles los accesos a la base de datos tanto para login de los médicos con su PDA como para consulta de documentos de los pacientes. Estas consultas a la base de datos se hacen a través de Java, gracias al paquete java.sql de J2SE.

## 8.5. Red

La red tiene se compone de dos partes:

- Red física: compuesta por el servidor, los puntos de acceso y los dispositivos periféricos.
- Red inalámbrica: compuesta por los dispositivos inalámbricos del personal, las PDA.

La distribución de los puntos de acceso se debe realizar ubicándolos en las esquinas de unos cuadrados imaginarios. El número de puntos y dispositivos periféricos depende del tamaño de la planta. Un ejemplo de planta sería el de la figura 13.

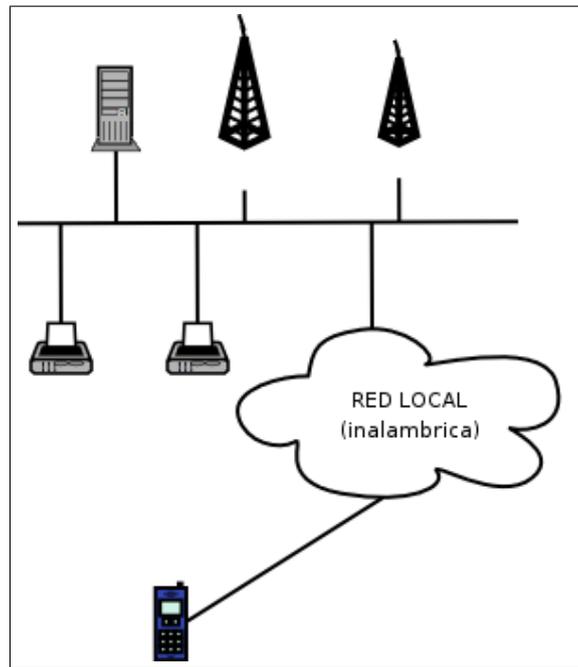


Figura 12: Diagrama de red

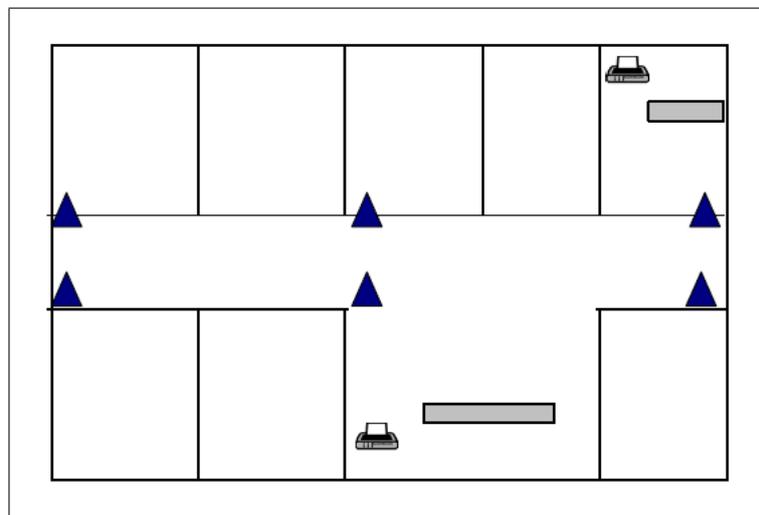


Figura 13: Mapa de una planta

## 9. Manual de usuario

### 1. Inicio de sesión:

Cuando un médico enciende su PDA para empezar a trabajar, lo primero que se encuentra es con la pantalla Inicio de sesión. El médico deberá introducir su nombre de usuario y su clave personal en los cuadros de texto correspondientes valiéndose del teclado del PDA y pulsar el botón Aceptar. Este es un procedimiento rutinario que impide que personal no autorizado pueda acceder a información confidencial del hospital.

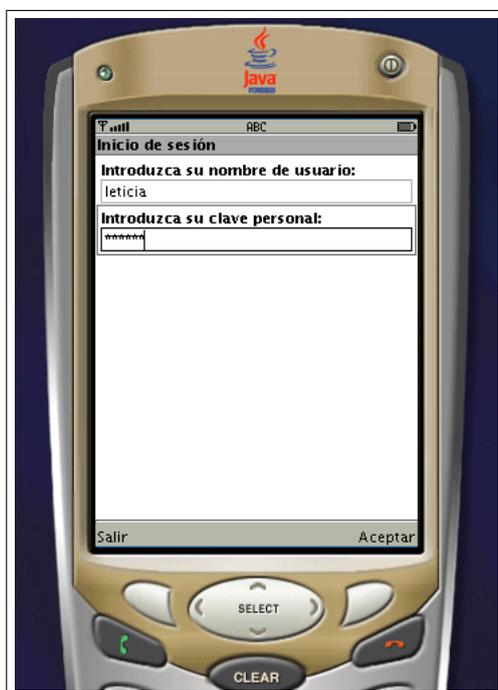


Figura 14: Pantalla de inicio de sesión

En caso de que el médico no haya introducido correctamente su nombre de usuario o su clave personal, no podrá acceder a la aplicación. Se le informará de ello y se le dará la oportunidad de introducir sus datos de nuevo.

En caso de que pulsáramos el botón Salir, en el margen inferior izquierdo de la pantalla, saldríamos de la aplicación y se apagaría el PDA.

### 2. Menú principal:

Una vez iniciada la sesión, el médico tiene acceso al menú principal donde se le ofertan todas las utilidades de la aplicación.



Figura 15: Pantalla de error en el inicio de sesión



Figura 16: Pantalla del menú principal

Para elegir una opción del menú debe desplazarse por el menú con las flechas centrales hacia arriba y hacia abajo y pulsar Select cuando tenga seleccionada la tarea que desea realizar.

Ahora hablaremos de las diferentes opciones del menú principal:

- Menú de impresión:  
A través de esta opción podemos imprimir un documento disponible del paciente del hospital deseado: sólo tenemos que elegir el tipo de documento a imprimir (Expediente o Últimos análisis) e introducir a través del teclado el nombre del paciente del que precisamos los documentos. Una vez introducidos estos datos, solo hay que pulsar el botón Aceptar, en el margen inferior derecho de la pantalla.



Figura 17: Pantalla del menú de impresión

El botón Atrás nos devuelve al menú principal.

En caso de que introduzcamos mal el nombre del paciente, no estén disponibles los documentos o no sea posible el uso de las impresoras, se informará al usuario y se le instará a intentarlo más tarde.

En caso de que no haya ningún problema en la realización de la tarea, se informará al usuario de la impresora por la que se llevará a cabo la impresión solicitada.

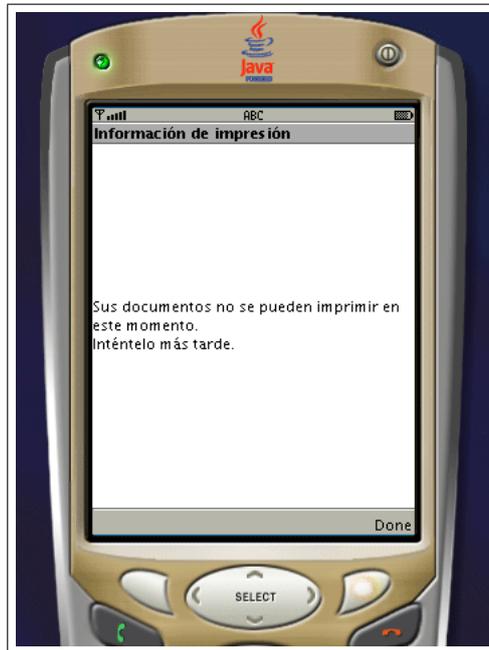


Figura 18: Pantalla de error en el procedimiento de imprimir



Figura 19: Pantalla de información de impresión

- Consultar expediente:  
Con esta opción podemos visualizar en la pantalla del PDA el expediente del paciente deseado. Al seleccionarla llegamos a una pantalla en la que solo tenemos que introducir el nombre del paciente y pulsar el botón Aceptar para acceder al expediente deseado.



Figura 20: Pantalla de petición de consulta de expediente

El botón Atrás nos devuelve al menú principal.

En caso de que introduzcamos mal el nombre del paciente o no estén disponibles los documentos, se informará al usuario.

En caso de que los datos sean correctos se mostrará por pantalla el expediente del paciente solicitado, como se muestra en la figura 22



Figura 21: Pantalla de error en la consulta de expediente

- Salir:  
Esta opción nos permite salir de la aplicación y apagar el PDA.

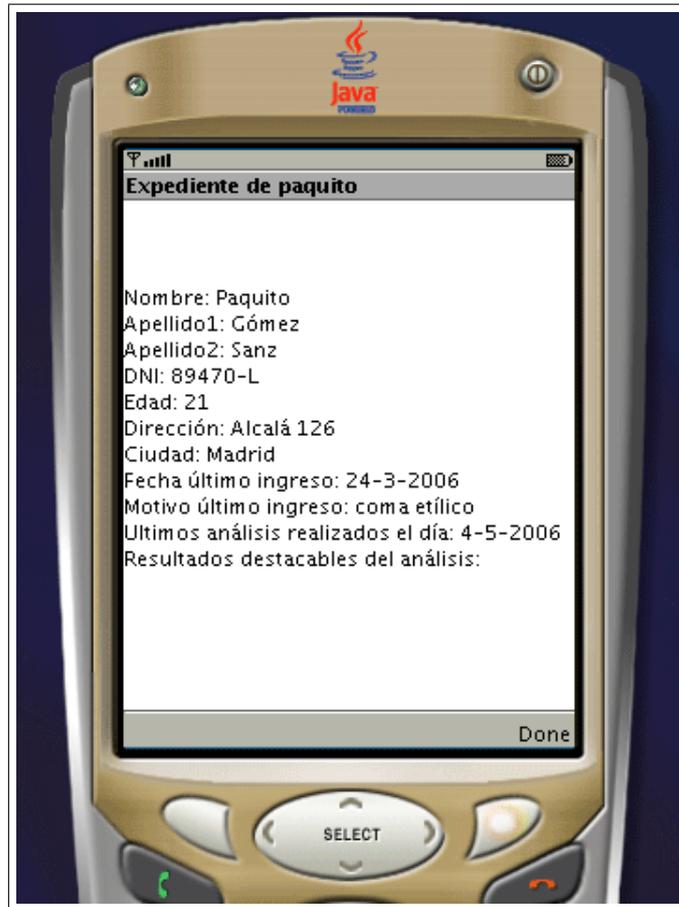


Figura 22:

## **10. Herramientas de desarrollo**

### **10.1. Eclipse**

Eclipse es un IDE multiplataforma libre para crear aplicaciones clientes de cualquier tipo. La primera y ms importante aplicacin que ha sido realizada con este entorno es el afamado IDE Java llamado Java Development Toolkit (JDT) y el compilador incluido en Eclipse, que se usaron para desarrollar el propio Eclipse.

El entorno integrado de desarrollo (IDE) de Eclipse emplea mdulos (en ingls plug-in) para proporcionar toda su funcionalidad. El mecanismo de mdulos permite que el entorno de desarrollo soporte otros lenguajes adems de Java. Por ejemplo, existe un mdulo para dar soporte a C/C++. Existen mdulos para aadir un poco de todo, desde Telnet hasta soporte a bases de datos. Esto nos ha permitido usar este entorno tanto para el desarrollo de la aplicacin residente en el PDA (plug-in para J2ME) como la parte del servidor (el antes comentado JDT).

### **10.2. Umbrello**

Umbrello UML Modeller es una herramienta software para el proceso de desarrollo de aplicaciones que usa el estandar Unified Modelling Language (UML) permitiendo crear diagrama para el diseo y la documentacin de los sistemas. Hemos empleado este programa para generar los diagramas de secuencia y los diagramas de clases de la aplicacin.

### **10.3. TextMaker**

TextMaker es un programa para editar LaTeX, el programa con el que hemos escrito esta memoria. TextMaker no s3lo facilita la escritura del c3digo LaTeX, sino que adem3s permite compilar desde el mismo editor dicho c3digo y visualizar posibles errores y vistas preliminares en diferentes formatos.

## 11. Gestión de configuración

Debido a la cantidad de código que se iba a generar, y a los problemas de integración que pudieran surgir, se decidió usar como sistema de mantenimiento de código el CVS. El CVS nos permitía trabajar desde nuestras casas y mantener actualizadas las últimas versiones de los ficheros fuente. Debido a que el CVS nos permite almacenar no sólo código, lo hemos empleado también para guardar cualquier otro dato de interés para el proyecto: documentos, actas de reuniones, etc

La estructura de directorios en el CVS es la siguiente:

- Código: los ficheros fuente de los distintos lenguajes.
- Documentos: documentos de interés, tales como actas, manuales, etc
- Memoria: los tex que documentan el proyecto y forman parte de la memoria.

El servidor CVS nos ha sido proporcionado por Berlios, <http://developer.berlios.de>

### 11.1. Reuniones

Con el fin de poner en común los avances, así como evaluar el estado del proyecto, se fijaron reuniones cada dos semanas. De cada una de estas reuniones hay un acta, con el nombre de la fecha en la que tuvo lugar la reunión. Este acta era escrita y pasada a limpio por un secretario, puesto que se cambiaba semanalmente entre los miembros del equipo.

También se realizaron diversas reuniones con el director del proyecto, se intentaban realizar cada 2 semanas, pero en algunas ocasiones, debido a que los avances no eran los esperados o a que se tenían que tomar decisiones importantes, estas reuniones se retrasaban o adelantaban.

### 11.2. Comunicación interna

Para mantener el contacto entre los distintos miembros del grupo, se creo una lista de correo electrónico para poder difundir con la mayor brevedad posible una noticia a todos los integrantes. Esta lista nos la ofrece Berlios, tras registrar allí nuestro proyecto.

En ocasiones extraordinarias la comunicación se realizaba mediante llamadas telefónicas.

### **11.3. Organización de trabajo**

Se ha intentado asignar el trabajo individual de la manera más eficiente posible, para que cada miembro se distribuya su tiempo como mejor considere. Esta asignación se realizaba en las reuniones semanales. Cuando el trabajo tenía que ser realizado de manera colectiva, en casos como integrar las distintas partes del proyecto nos reuníamos en un mismo local los miembros involucrados en las partes a unir.

## 12. Valoración

La valoración del proyecto la vamos a realizar en dos áreas: uso de tecnologías e impacto de la aplicación.

### ▪ Uso de tecnologías:

En el análisis de las tecnologías aplicadas en este proyecto cabe reseñar el uso de tecnologías punteras como J2ME y redes inalámbricas. Además es el uso de múltiples tecnologías y su coordinación.

Integrar bases de datos, servidores de impresión, dispositivos inalámbricos, periféricos en red, etc habría sido una tarea imposible si no hubiéramos llevado a cabo una documentación en inicial no solo sobre las características de las tecnologías y aplicaciones usadas para el desarrollo sino además sobre la compatibilidad de las distintas tecnologías antes de llevar a cabo el diseño de este proyecto.

### ▪ Impacto de la aplicación

Esta aplicación tiene un gran calado no solo por su utilidad actual sino por su adaptabilidad a distintas áreas.

En la era de las nuevas tecnologías y los móviles 3G una aplicación que facilite las operaciones con archivos de un hospital etc es básica. Sus utilidades son muchas, debido a su amplitud, podemos tener una aplicación que no solo envíe los informes a la PDA o al dispositivo más próximo al usuario, sino que permita ver gráficos de evolución de datos del paciente, localizar a otro miembro del personal, reservar el uso de determinadas dependencias del centro, consultar el personal que hay de guardia, etc. Además es un diseño reutilizable para otro tipo de actividades:

#### • Museos

En un museo se podría indicar al visitante que camino a de recorrer para llegar a determinada obra, cual es el mejor camino para visitar una serie de obras, información sobre la obra que tenga más próxima, localizar al personal del museo para consultas, etc

- Bibliotecas

En este caso las utilidades son parecidas. Además se podría, por ejemplo, ubicar los libros en un mapa virtual o consultar sus descripciones antes de desplazarnos a buscarlos.

- Almacenes

Facilitaría la tarea a los empleados, que tendrían localizados a sus compañeros y podrían consultar la ubicación de un paquete o como llegar a él sin encontrarse con los puntos más congestionados de gente. Para el encargado le permite conocer el nivel de ocupación del almacén, el estado de las tareas actuales,...

## 13. Apéndices

### 13.1. J2ME

Como ya hemos comentado anteriormente, Java 2 Micro Edition (J2ME), es un subconjunto de J2SE orientado al desarrollo de aplicaciones Java destinadas a dispositivos con capacidades restringidas.

La arquitectura J2ME está diseñada con la filosofía de ser escalable y modular, ya que no se conoce como serán los dispositivos en el futuro y debe estar preparada para adaptarse a ellos. Sus características están definidas en un entorno global constituido por varias capas, de abajo a arriba:

1. Capa correspondiente a la Máquina Virtual de Java: una versión reducida para dispositivos reducidos.
2. Capa de Configuración: está orientada al dispositivo y define el mínimo conjunto de características de la máquina virtual Java y de las librerías de clases Java que están disponibles para un conjunto de dispositivos.
3. Capa de Perfil: está orientada a la aplicación y define el mínimo conjunto de APIs disponibles para una determinada familia de dispositivos. Entre los perfiles que se han desarrollado hasta ahora, destaca el perfil PDA, que extiende el perfil CLDC para adecuarse a las ventajas que ofrecen los dispositivos PDA.
4. Capa del *Perfil para Dispositivos de Información Móvil* (MIDP) consiste en un conjunto de APIs Java que permiten la creación de interfaces de usuario, conexiones de red, manipulación de datos, sonido, seguridad...

La combinación de las tres primeras capas constituye la configuración CLDC, que junto con la capa MIDP forman el entorno de ejecución estándar para las aplicaciones y servicios que se pueden descargar dinámicamente sobre los dispositivos de los usuarios finales, como pueden ser los PDA.

El corazón del perfil MIDP es un **midlet**. Una aplicación midlet extiende la clase MIDlet, que proporciona a esa aplicación la posibilidad de recuperar propiedades, controlar cambios de estado y constituye la interfaz entre el entorno de ejecución del dispositivo y el código de la aplicación midlet.

El hecho de que una aplicación midlet extienda la clase MIDlet hace que el programador se vea obligado a implementar los siguientes métodos abstractos:

- `startApp()`: en este método la aplicación hace acopio de los recursos que va a necesitar el midlet y donde se preparan los controladores de eventos.
- `pauseApp()`: este método es invocado cuando se necesita detener la ejecución del midlet temporalmente.
- `destroyApp()`: este método es invocado cuando el midlet debe ser destruido o también puede ser invocado por el propio midlet antes de finalizar su ejecución.

La implementación de estos métodos asegura que el midlet pueda pasar por todos los estados de su ciclo de vida: DETENIDO, ACTIVO, DESTRUIDO.

### 13.2. Algoritmo de Floyd

Pseudocódigo del algoritmo de Floyd en el que nos hemos basado para la implementación:

```
function fw(int[1..n,1..n] graph) {
    var int[1..n,1..n] dist := graph
    var int[1..n,1..n] pred
    for i from 1 to n
        for j from 1 to n
            if dist[i,j] < Infinity
                pred[i,j] := i
    for k from 1 to n
        for i from 1 to n
            for j from 1 to n
                if dist[i,j] > dist[i,k] + dist[k,j]
                    dist[i,j] = dist[i,k] + dist[k,j]
                    pred[i,j] = pred[k,j]
    return dist
}
```

## 14. Agradecimientos

Queremos mencionar a las siguientes personas por la ayuda que nos han proporcionado durante el desarrollo de este proyecto:

- Manuel Ortega Ortíz de Apodaca
- María Teresa Hortalá González
- Juan Antonio Recio García
- Juan Rodríguez Hortalá
- Luis Vázquez Martínez
- Alberto Velázquez Alonso
- Familiares y amigos

## Referencias

- [1] J2ME Java 2 Micro Edition. Manual de usuario y tutorial. Agustín Froute Quintas, Patricia Jorge Cárdenes. Ed. Ra-Ma.
- [2] <http://java.sun.com>
- [3] <http://kxmlrpc.objectweb.org>
- [4] <http://eclipseme.sourceforge.net>
- [5] <http://es.wikipedia.org>
- [6] [http://www.programacion.com/bbdd/tutorial/mysql\\_basico](http://www.programacion.com/bbdd/tutorial/mysql_basico)
- [7] [http://www.francispisani.net/2004/04/wifi\\_y\\_la\\_local.html](http://www.francispisani.net/2004/04/wifi_y_la_local.html)
- [8] <http://standards.ieee.org/getieee802/802.11.html>
- [9] <http://www.cups.org>
- [10] <http://www.ubuntu-es.org>
- [11] <http://www.wifi-shootout.com>
- [12] <http://innovexpo.itee.uq.edu.au/2003/exhibits/s358272/thesis.pdf>
- [13] <http://www.herecast.com/Herecast.pdf>
- [14] <http://www.stumbler.net>
- [15] <http://es.tldp.org/Tutoriales/doc-openldap-samba-cups-python/htmls/cups-instalacion.html>
- [16] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=linuxprint>
- [17] <http://ditec.um.es/aso/teoria/tema7.pdf>
- [18] <http://forums.wi-fiplanet.com>