

3.3.6. Manual de uso de la API

En esta sección veremos como utilizar la API para poder conectarnos remotamente a una base de datos.

En primer lugar, hay que establecer la conexión con el servidor. Para ello es necesario llamar al método *connect*. Este método recibe como parámetro el nombre de la base de datos a la que se quiere conectar, el protocolo por el que se conectará (sockets o http), el servidor y puerto por el cual se conectará y los datos de identificación para poderse conectar al SGBD, es decir, el nombre de usuario y la contraseña. Si todos los parámetros son correctos recibiremos un identificador de sesión, el cual nos permitirá hacer otras operaciones sin tener que enviarle cada vez los datos de conexión.

```
private final static J2MERemoteDBLibSockets obj= new
J2MERemoteDBLibSockets();

Runnable r=new Runnable()
{
    public void run()
    {
        obj.connect("usuarios","socket","localhost","7000",
                    "idsgbd","passsgbd");
    }
};

Thread t = new Thread(r);
```

p

Después de haber establecido la conexión con el servidor se puede pasar a hacer operaciones sobre el SGBD, tales como selecciones y actualizaciones. Para hacer una selección, se enviara como parámetro la sentencia SQL que queramos ejecutar en el método *executeQuery*. Para hacer una actualización los parámetros serán los mismos que en la selección, sentencia SQL de actualización en el método *executeUpdate*.

```
private final static J2MERemoteDBLibSockets obj= new
J2MERemoteDBLibSockets();

Runnable r=new Runnable()
{
    public void run()
    {
        obj.executeQuery("select * from usuarios;");
    }
};

Thread t = new Thread(r);
```

```

private final static J2MERemoteDBLibSockets obj= new
J2MERemoteDBLibSockets();

Runnable r=new Runnable()
{
    public void run()
    {
        obj.executeUpdate("update usuarios SET
dni=000000 where nombre= Luis;");
    }
};

Thread t = new Thread(r);

```

Por último, una vez hayamos hecho todas las operaciones necesarias sobre el SGBD, deberemos cerrar la conexión. Para ello es necesario llamar al método *closeConnection*.

```

private final static J2MERemoteDBLibSockets obj= new
J2MERemoteDBLibSockets();

Runnable r=new Runnable()
{
    public void run()
    {
        obj.closeConnection(obj.id);
    }
};

Thread t = new Thread(r);

```

3.4. Rendimiento de la API

Se ha realizado una medición del tiempo que transcurre, desde que se realiza una petición, hasta que se satisface esta. Entrando en detalle, es el tiempo que tarda el dispositivo en enviar la petición, hay que tener en cuenta que el tráfico de red es un factor externo así como el tiempo que tarde el servidor en atender nuestra petición y realizar la query. Este acto es conocido como servicio.

Dado que estas variables externas son de difícil control se ha procedido a realizar las mediciones en un entorno cerrado (un solo cliente y en local) y realizar siempre la misma acción, para que la desviación por estos factores sea mínima cuando no nula.

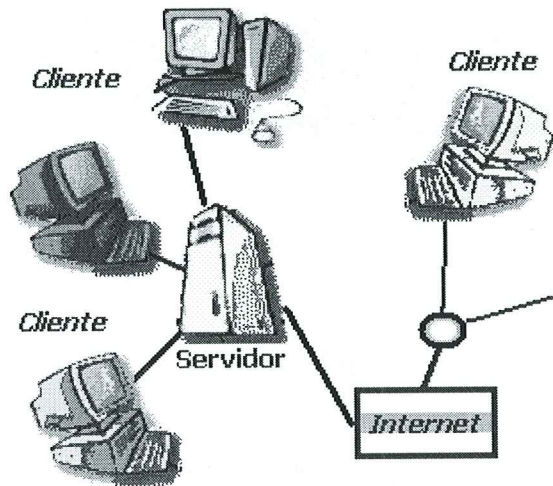


Figura 31 Conexión cliente-servidor

Mediciones realizadas:

Los resultados en milisegundo son una media del tiempo que tarda la operación en realizarse. Donde el volumen de datos es considerablemente pequeño.

Las pruebas realizadas están hechas sobre MySQL y el driver JDBC MM MySQL Driver (JDBC Driver Type-4).

| | EMULADOR | PDA |
|-----------------|----------|---------|
| Connect | 5230 ms | 2300 ms |
| ExecuteQuery | 230 ms | 180 ms |
| ExecuteUpdate | 350 ms | 240 ms |
| closeConnection | 1ms | 1 ms |

Figura 32 Mediciones

Gráficas:

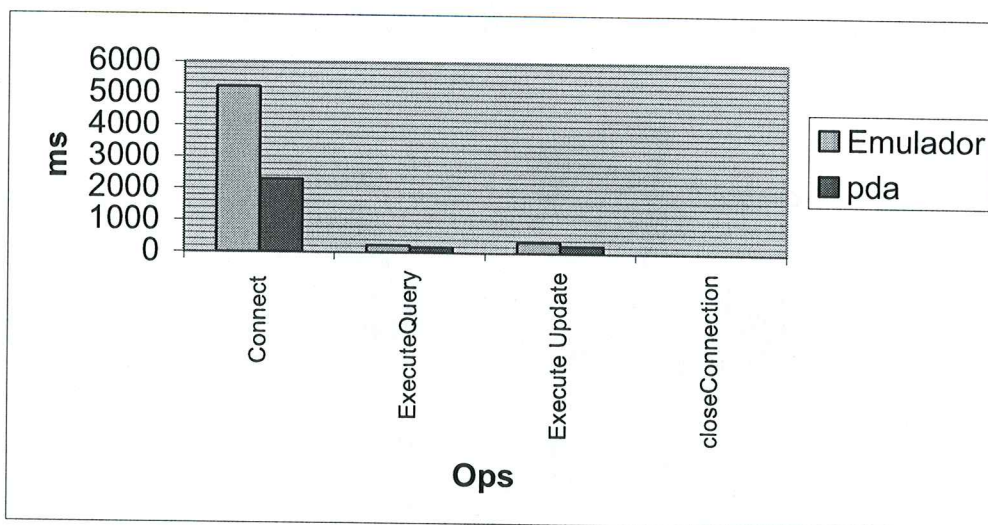


Figura 33 Gráfica tiempos

Resultados observados:

Se puede comprobar fácilmente en la gráfica que ejecutándose las operaciones desde el emulador los tiempos son algo mayores.

También se observa que el tiempo de conexión es muy superior al de las otras operaciones, eso es debido al volumen de datos con el que se han hecho los cálculos. Si el número de datos que quisiéramos consultar o bien actualizar fuera mucho mayor, el tiempo de esta operación sería incluso mayor que el tiempo de conexión.

Conclusiones:

Dado que la API espera ser de uso masivo es importante conocer sus puntos fuertes y débiles. Comprobar que el rendimiento es cercano al obtenido en un dispositivo de grandes prestaciones, sería un valor muy positivo.

4. APLICACIÓN CLIENTE

En este capítulo se comentan los criterios de usabilidad usados para la implementación de la interfaz gráfica, y un manual de esta.

4.1. Introducción

La API posibilita realizar de forma cómoda accesos a SGBD desde dispositivos móviles.

Se ha creado una interfaz grafica, para poder visualizar el comportamiento de la API en un dispositivo móvil real.

Esta interfaz intenta cumplir los criterios de usabilidad que se explica en la siguiente sección intentando ser una interfaz amigable para el usuario final.

4.2. Criterios de usabilidad

Como para toda interfaz gráfica existen ciertos patrones para diseñarlas. Hasta el momento ciertos autores han citado cuales son los puntos clave a observar, en el momento de diseñar una interfaz gráfica, como la importancia de conocer bien el usuario final, cuales son sus conocimientos, cuales sus carencias y que es lo que necesitan.

A continuación se muestra la lista de guías o aspectos a tener en cuenta en el diseño de interfaces para dispositivos móviles.

1. **Identificar la audiencia:** quiénes son los usuarios y qué necesidades tienen. Esto permitirá determinar cómo se estructura el sitio en función de los requisitos de sus usuarios.
2. **Identificar las tareas más comunes y urgentes:** qué harán más frecuentemente los usuarios de un sitio y cuál es la real importancia de ellas. Esto permitirá priorizar y definir los requisitos y condiciones en que se presentarán los contenidos.
3. **Determinar contenido y funciones clave:** no todo el contenido debe estar disponible para los móviles, hay contenidos y servicios particularmente sensibles que deben ser accesibles desde cualquier lugar, como por ejemplo, sistemas de mensajería, webmail, etc.

4. **Identificar las plataformas más utilizadas por los usuarios:** conocer las condiciones de navegación de modo tradicional (PC, notebook) y las plataformas móviles preferidas por los usuarios permite establecer una estrategia más precisa.

Por lo que a nuestra interfaz representa, el apartado uno, sabemos claramente quienes son los usuarios, es decir, programadores o personal técnico entendido en bases de datos. En el apartado número dos, las tareas más comunes o urgente son las equivalentes en los menús: Conectar, Selección, Actualización y Cerrar. Al igual que el punto numero 1 y 2, el 3 y el 4 también se han tenido en cuenta.

Además de estas guías, hay estudios específicos de cómo incrementar la usabilidad en estos pequeños dispositivos. Las propuestas más usuales son las siguientes:

1. La interfaz debe ser simple, eso implica que el número de botones de la interfaz tiene que ser mínimo.
2. Todos los elementos y posibles combinaciones del dispositivo tienen que ser intuitivas y comprensibles para usuarios con un mínimo conocimiento del terminal, éste no tiene que ser un experto.
3. La profundidad de los menús debe ser la correcta, no muy profundos, a ser posible no mayores de 3 ya que podría desorientar al usuario.
4. El número idóneo de elementos que aparecen en pantalla no debe ser superior a 8 para evitar así una sensación de desorden.
5. Los diferentes tipos de letras, así como los colores de éstas, serán reducidos también, para evitar que el usuario se distraiga.
6. Cada botón será utilizado para una única acción, a excepción del botón de encendido y apagado (*on/off*) el cual permite realizar dos tareas como máximo, tres ya sería un error y podría causar equivocaciones.
7. En los botones principales, se deberían asociarse las tareas más frecuentes del usuario, nunca pondríamos la opción eliminar un contacto en un botón principal.

Como se puede ver en las Figuras de la 34 a la 43 se cumplen todas estas propuestas: en la mayoría de pantallas hay uno a dos botones, los elementos son intuitivos (teniendo en cuenta que el usuario final conoce el lenguaje de las bases de datos), los menús tienen una profundidad máxima de tres niveles, no se usan demasiados colores, cada botón solamente es usado para un fin, etc.

Habiendo comprobado estas guías y aspectos podemos clasificar nuestro diseño de la interfaz gráfica como Usable.

4.3. Manual de usuario

En esta sección queremos proporcionar un manual de usuario de la aplicación de demostración encargada de conectarse a un SGBD remoto y consultar tablas guardadas en el dispositivo móvil. Esta aplicación es sólo una aplicación pensada para ilustrar el funcionamiento de la librería.

Para ejecutar la aplicación hay que hacer doble clic sobre el icono APIBDREM.jad. A continuación aparece la pantalla del emulador con la lista de midlets a ejecutar. En pantalla aparece la midlet que tiene por nombre Demo. Seleccionarla y pulsar el botón de "launch". Entonces aparece en pantalla las dos primeras opciones, o bien conectarse al servidor para trabajar con bases de datos remotas, o trabajar con las tablas que tenemos guardadas en la PDA o Smart Phone (Figura 34).



Figura 34 Pantalla Acceso BD Remota



Figura 35 Pantalla de Conexión

Para conectarnos basta con seleccionar la opción *Conectar* del menú y seleccionar *OK*. Entonces aparecerá la nueva ventana en la que se deberán introducir los datos necesarios para establecer la conexión (*Figura 35*). Los datos son, el nombre de la base de datos a la que nos queremos conectar, el protocolo que puede ser *socket* o *http* solamente, el nombre de usuario y la contraseña del SGBD.

Una vez introducidos los datos, si alguno de ellos no es correcto se muestra una ventana en la que se nos informa de ello (*Figura 36*). En caso de que todo haya ido bien se pasa a la siguiente ventana (*Figura 37*).



Figura 36 Pantalla de error en los datos



Figura 37 Pantalla Operaciones

En esta ventana se elige la operación a ejecutar contra el SGBD, o bien una sentencia de consulta, o una sentencia de actualización, o el cierre de la conexión. Tanto en la consulta como actualización, una vez se ha seleccionado el *OK* aparece la ventana donde se introduce la sentencia SQL (*Figuras 38 y 39*).



Figura 38 Pantalla selección



Figura 39 Pantalla actualización

Si la sentencia está bien escrita, en el caso de la selección se mostrará el resultado en una nueva pantalla (Figura 40). Y en el caso de la actualización se mostrará un mensaje conforme la actualización se ha hecho correctamente (Figura 49).



Figura 40 Pantalla resultado selección



Figura 41 Pantalla resultad actualización

La ultima posible operación es la de *Cerrar*, que una vez marcada y seleccionado el *OK*, se cerrará la conexión con el SGBD, volviendo así a la pantalla principal.

Desde la pantalla Operaciones (Figura 37), si se selecciona Cancelar, equivaldrá a la operación de cerrar

En caso que en la primera pantalla se quiera consultar las tablas que tenemos en el propio móvil, hay que seleccionar la opción "Listar Tablas". Y se mostrará la ventana correspondiente con todas las tablas existentes (Figura42).

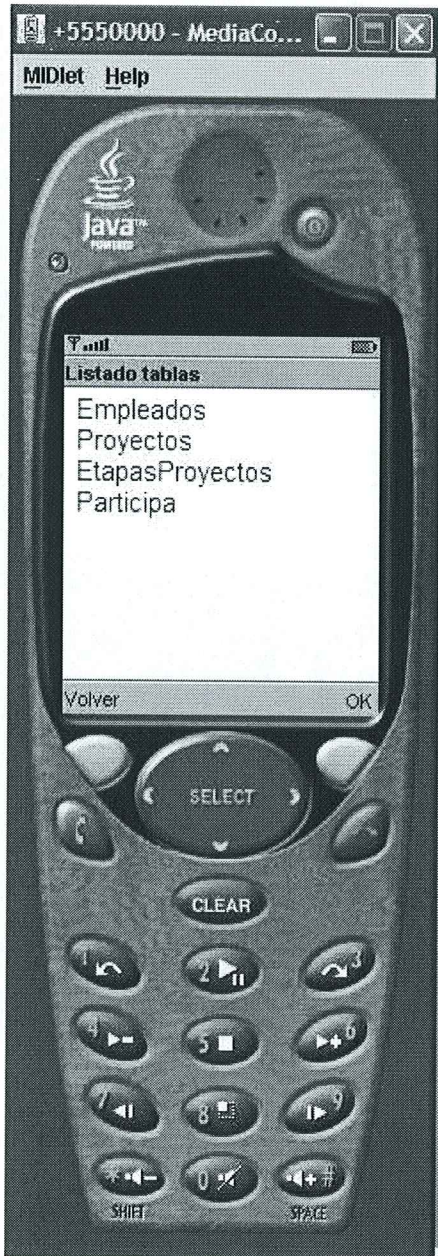


Figura 42 Pantalla Listado Tablas



Figura 43 Pantalla Opciones

Para hacer operaciones sobre estas tablas, habrá que seleccionar la tabla con la que se quiera trabajar y seleccionar *OK*. Entonces aparecerá la siguiente ventana con las operaciones disponibles (Figura 41).

En esta ventana, si marcamos la opción de *Ver contenido* y seleccionamos *OK*, se mostrará el contenido de la tabla.

Si marcamos la opción *Borrar* y seleccionamos *OK*, se mostrara un mensaje conforme aseguramos o no, que queremos borrar la tabla (Figura 44) y en caso de confirmarlo, aparecerá una alerta conforme si la tabla ha sido borrada correctamente o no(Figura 45).



Figura 44 Pantalla aviso borrado



Figura 45 Pantalla estado borrado

Como última opción, hay la opción de *Ver Estructura* que una vez marcada y seleccionado el *OK*, se mostrará los campos y el tipo de dato de cada campo para la tabla anteriormente seleccionada.

Todos los botones que su etiqueta sea *Volver*, al seleccionarla nos llevarán a la pantalla anterior.

5. GUIA DE INSTALACIÓN DEL PROXY

En este capítulo se explicará el procedimiento a seguir para instalar lo necesario para poder empezar a desarrollar con la API.

GUIA INSTALACIÓN

Manual de usuario

V 1.0

Septiembre 2007

1. Introducción
2. Requisitos del sistema
3. Instalación y configuración
4. Uso

5.1. Introducción

Bienvenido al manual de instalación tanto del servidor proxy como de la aplicación cliente. Este manual pretende que consiga instalar un servidor para que atienda peticiones y un cliente para poder realizarlas.

Se deben realizar tres pasos:

Instalar un servidor, Instalar un SGBD e Instalar el cliente.

5.2. Requisitos del Sistema

Proxy

- Java Virtual Machina correctamente instalado.
- Driver JDBC.
- Sistema operativo Windows o Linux
- PC multimedia.
- SGBD que interprete sentencias SQL

Cliente:

- Teléfono multimedia (3 Mb libres de espacio)

5.3. Instalación y Configuración

Con la API se suministra una interfaz que posibilita su acceso. Debido a que la dirección del servidor se a de configurar manualmente necesitará recompilar el cliente.

Una vez tenga abierto el proyecto (se explica en el punto "Diseño de la API") solo ha de poner la dirección URL de su servidor SGBD favorito y compilar el proyecto (generar el JAR) una vez realizada esta operación puede pasar el .jar generado directamente a su dispositivo (copiando el .jar a su tarjeta de memoria, mediante bluetooth o infrarrojos)

Instalación del JDK (J2SE)

La base para operar cualquier producto que utiliza "Java" es el "JDK" de la plataforma correspondiente, puede encontrar instrucciones para plataformas Linux así como Windows, en las siguientes direcciones:

- JDK para Linux : http://javabasico.osmosislatina.com/java_linux.htm
- JDK para Windows : http://javabasico.osmosislatina.com/java_windows.htm

Instalación del driver JDBC para MySQL

Existen varios drivers JDBC para MySQL. Nos hemos decidido a instalar el *MM MySQL Driver* (JDBC Driver Type-4) ya que es uno de los recomendados por los propios autores de MySQL, es bastante completo y se distribuye bajo la licencia LGPL (igual que la GPL, pero permite el uso del software cubierto bajo la licencia en programas propietarios y no open-source).

El paquete con la última versión (2.0.2) se puede obtener de la página web del MM MySQL (binarios y código fuente y documentación). Para instalar el driver únicamente necesitamos el paquete con los binarios (bytecode Java ya compilado), aunque no está de más obtener también el otro archivo para tener la documentación.

Para instalar el MM MySQL hay que copiar el archivo con los binarios (mm.mysql-2.0.2-bin.jar) en algún directorio especificado en la variable de entorno CLASSPATH, o bien copiarlo en algún otro sitio y añadir *el path* completo del fichero, incluyendo su nombre, a dicha variable. Nosotros lo hemos copiado en el directorio /usr/local/mm.mysql y por tanto hemos tenido que añadir el path del fichero a la variable CLASSPAHT (en /etc/environment):

```
export CLASSPATH=/usr/local/mm.mysql/mm.mysql-2.0.2-bin.jar:$CLASSPATH
```

Además, tendremos que añadir el path del fichero en la variable classpath interna del servidor de Servlets JServ, en el fichero `/etc/jserv/jserv.properties`:

wrapper.classpath=/usr/local/mm.mysql/mm.mysql-2.0.2-bin.jar

Una vez hecho esto, ya podemos acceder a MySQL utilizando el API JDBC.

Instalación del proxy

Por último, habrá que proceder a instalar el proxy. Para poder acceder desde el móvil al SGBD habrá que tener encendido el proxy. Para hacerlo solamente se tiene que copiar el ejecutable en una carpeta dónde tengamos los ficheros compilados del proxy, es decir, los `.class`.

Es importante saber que:

- 1- Cuando se enciende el proxy se genera un bucle infinito dado que este, está en todo momento escuchando peticiones de los clientes.
- 2- El proxy contiene un fichero donde se guardan las bases de datos a las que podemos acceder.

5.4. Uso

El uso del cliente, está definido en la sección: manual de usuario.

6. COSTES Y TIEMPO EMPLEADO

En este capítulo, se planteará un informe temporal y económico del coste del proyecto.

6.1. Introducción

En este apartado se informará del tiempo empleado en cada parte del proyecto así como del coste que hubiera tenido.

Para entender la información mostrada a continuación habrá que tener en cuenta que la jornada laboral es de lunes a viernes, es decir sin incluir ni sábados ni festivos. El horario laboral es de entre 4 y 5 horas diarias.

Las partes en las que está formado el proyecto son :

- Formación
- Desarrollo del proyecto :
 - Desarrollo del servidor proxy
 - Desarrollo de la API
 - Desarrollo de la interfaz gráfica
- Documentación
- Pruebas y correcciones
- Presentación

6.2. Especificación del tiempo empleado

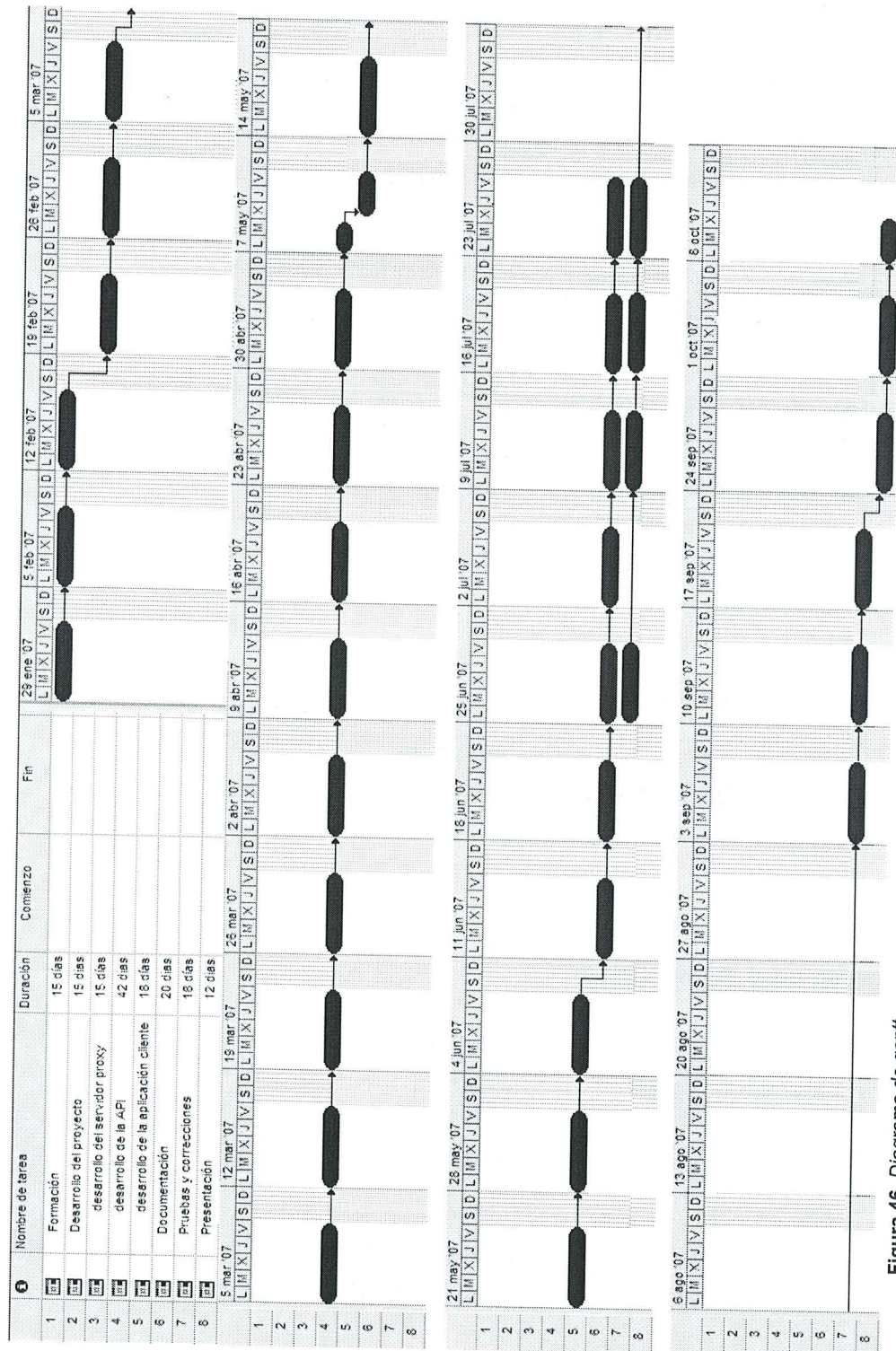


Figura 46 Diagrama de gantt

6.3. Costes del Proyecto

Coste Software:

Se ha optado claramente por el open source, con lo que se ha eliminado el coste del software utilizado en el proyecto. Exceptuando el sistema operativo (Windows XP).

| | |
|---------------------|--------------|
| Windows XP | 300 € |
| Eclipse | 0 € |
| Jakarta-Tomcat | 0 € |
| JDK 1.5.0 | 0 € |
| Mysql Server | 0 € |
| Wireless toolkit | 0 € |
| Visio (licencia) | 0 € |
| Project (licencia) | 0 € |
| | |
| Total: | 300 € |

Horas de desarrollo

| | | |
|-----------------------|-----------|----------------|
| 70 horas analista | 40 €/hora | 2.800 € |
| 190 horas programador | 10 €/hora | 1.900 € |
| | | |
| Total: | | 4.700 € |

Coste total del proyecto:

| | |
|------------------|----------------|
| Coste Software | 300 € |
| Coste Desarrollo | 4.700€ |
| | |
| Total: | 5.000 € |

No se ha contado en este importe el gasto derivado de la redacción de manuales ni de la memoria. Para hacerlo, habría que sumarle 3.600 €.

7. CONCLUSIONES

En este capítulo se comentarán los objetivos conseguidos, los trabajos futuros y finalmente una valoración personal.

7.1. Objetivos conseguidos

Los principales objetivos del proyecto eran implementar una librería para poder conectar dispositivos móviles a base de datos remotas y la implementación de un cliente móvil para poder hacer servir la API.

Los dos objetivos se han cumplido.

7.2. Posibles mejoras y faena futura

Encapsular el método `executeQuery` y `executeUpdate` en uno solo, ya que quien hace la distinción es el propio SGBD al catalogar la petición para ser ejecutada.

Hacer que el proxy permitiese configurar a través de una aplicación gráfica las bases de datos a las que se podrá acceder desde el móvil en lugar de tener que leer los datos desde un fichero.

Definir otra librería que accediera a esta librería para acceder a BD remotas que actuase como si fuera un driver JDBC.

7.3. Valoración personal

¿Qué me ha aportado este proyecto en el terreno profesional y personal? La realización de este proyecto me ha aportado conocimientos técnicos y ha mejorado mi capacidad de trabajo individual, que requiere una gran dosis de disciplina. En un equipo, tienes un compromiso con los componentes de este además del compromiso con el propio trabajo. Cuando trabajas solo, no puedes eliminar la motivación del grupo y tienes que ser tu propio gestor, tanto de tiempo como de recursos.

Respecto a los conocimientos técnicos, he podido acercarme a una tecnología desconocida, pero que tiene bastantes recursos disponibles y sobre la cual se puede encontrar información aceptable, todo y que no se puede comparar ni en cantidad ni diversidad con otras tecnologías. Todavía es una tecnología que tiene que acabar de desarrollarse y crecer, porque en determinados ámbitos y a causa de la gran diversidad de dispositivos móviles, no se garantiza que una misma aplicación funcione en todos los dispositivos. Incluso dentro de la misma familia, cada uno tiene sus requerimientos.

Referencias y bibliografía

Generales:

- <http://es.wikipedia.org/>

Java:

- <http://java.sun.com>
- Javadoc: <http://java.sun.com/j2se/javadoc/>

J2ME:

- J2ME: The Complete Reference
 - Autor: James Keogh
 - Editorial: McGraw-Hill/Osborne
- J2ME (Java 2 Micro Edition)
 - Autores: Sergio Gálvez Rojas i Lucas Ortega Díaz
 - Web descarga:
<http://www.lcc.uma.es/~galvez/ftp/libros/J2ME.pdf>

Eclipse i plug-ins:

- Página principal d'*Eclipse*, donde se puede encontrar la documentación y descargarse el IDE:
 - <http://www.eclipse.org/>
- Plug-in *EclipseME*, para el desarrollo integrado de MIDlets. Básica la documentación sobre instalación y configuración que se puede encontrar:
 - <http://eclipseme.org/>

Emulador J2ME Wireless Toolkit

- <http://java.sun.com/products/sjwtoolkit/>

Otros links consultados:

http://www.latinoamerica.nokia.com/lta_es/phones/technologies/java/frequently_asked_questions/index.htm

<http://www.devarticles.com/c/a/Java/Listeners-in-Java/1/>

<http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=137>

<http://www.onjava.com/pub/a/onjava/2005/05/11/sqlexplorer.html>

http://www.it.uc3m.es/celeste/docencia/j2me/tutoriales/midp2_0/PracticalO/

<http://www.java2s.com/Code/Java/J2ME/SocketMIDlet.htm>

<http://java.sun.com/javame/index.jsp>

