



Índice Principal

1. Introducción al “fast work”	4
2. Definición general del proyecto	4
2.1 Objetivos.....	4
2.2. Funcionalidades	5
3. Calendario del Proyecto	6
3.1 Entregas establecidas.....	6
3.2 Calendario	7
3.3 Detalle de entrega para la PAC1	8
3.4 Detalle de entrega para la PAC2	9
3.5 Detalle de entrega para la PAC3	10
3.6 Detalle de entrega para la entregar final	10
4. Recursos y infraestructura	11
4.1 Recursos Hardware.....	12
4.2 Recursos de Software desarrollo.....	13
4.3 Recursos Software al Servidor	13
5. Tecnología a aplicar	14
5.1 Tecnología móvil (Android).....	14
5.2 Tecnología de sincronización de información(Webservice)	15
5.3 Tecnología de web.....	15
6.Riesgos del proyecto	16
7. Análisis funcional	17
7.1 Requerimientos funcionales	17
7.1.1 Descripción básica del funcionamiento	17
7.1.2 Particularidades de la versión móvil i de la web	18
7.1.3 Seguridad	18
7.2 Requerimientos no funcionales	19
7.2.1 Interfaz	19
7.2.2 uso de característica propias del dispositivo móvil.....	20
7.3 Funcionalidad del sistema	20
7.3.1 Plataforma móvil.....	20
7.3.2 Plataforma web.....	23
7.3.3 Plataforma Webservice.....	25
7.4 Usuario del sistema.....	26
7.5 Caso de uso.....	27
7.5.1 Descripción del caso de uso relacionado con Tareas	29
7.5.2 Descripción del caso de uso relacionado con Categoría	30
7.5.3 Descripción del caso de uso relacionado con Clientes	31
7.5.4 Descripción del caso de uso relacionado con Facturas	32
8. Diseño Técnico	33
8.1 Arquitectura Móvil	33
8.1.1 Característica de la plataforma móvil.....	34
8.2 Arquitectura Web y WebService (parte servidor)	35
8.2.1 Característica de la plataforma web y Webservice	36
8.2 Arquitectura Física	36
8.3 Arquitectura Lógica	37
8.3.1 Arquitectura lógica de la plataforma móvil.....	38

8.3.2) Arquitectura lógica de la plataforma web.....	39
8.3.3) Arquitectura lógica de intercambio de información entre plataformas	41
8.4 Arquitectura de Base de datos	43
8.4.1 Modelo relacional de la base de datos	43
8.4.1 Tareas	44
8.4.2 Categoría.....	45
8.4.3 Cliente	45
8.4.4 Factura.....	45
8.5 Diagrama de Base de datos	46
8.5.1 Diagrama de base de datos de la plataforma móvil i web.....	46
8.5.2) Diagrama de clases de la plataforma WebService	47
9. Prototipo y funcionamiento	49
9.1 Prototipo móvil	49
9.1.1 Pantalla Menú.....	49
9.1.2 Pantalla de Tarea	50
9.1.3 Pantalla de Categoría	51
9.1.4 Pantalla de Cliente.....	52
9.1.5 Pantalla de configuración.....	52
9.1.6 Sincronización de datos	53
9.1.6 Buscar datos.....	54
10 . Prototipo web	55
10.1 Pantalla de identificación.....	55
10.2 Crear Usuario	56
10.3 Pantalla Principal de Fast-Work.....	57
10.4 Listas de tarea	57
11. Implementación	58
11.1 Premisas de la implementación	58
11.1.1 Minimizar el consumo de batería.....	58
11.1.2 Facilidades en la navegación.....	59
11.1.3 Teclado en función del tipo de campo	60
11.1.4 Aplicación de multi-idioma.....	61
11.1.5 Código fuente comentado.....	61
11.2 Implementación de la base de datos	62
11.3 Implementación de la plataforma móvil.....	63
11.3.1 Librerías de diseño GreenDroid	63
11.3.2 Organización del proyecto móvil	66
11.3.3 Herencia de la clase <i>dialog</i>	70
11.3.4 Comunicación con la plataforma web	71
11.4 Implementación de la plataforma web.....	72
12. Conclusión.....	74
12.1 Objetivos cumplidos.....	74
12.2 Valoraciones personal.....	75
12.3 Futuras mejoras.....	76
13. Fuente de información.....	77
13.1 Biografía.....	77
13.2 Consultas Android	77
13.3 Consultas PHP	77
13.4 Consultas SQL	78
13.5 Otras Consultas	78

1. Introducción al “fast work”

El Fast-Work es una iniciativa de dar un mayor control a las horas que pasamos trabajando en las diferentes tareas del día. Algunas veces hacemos trabajos para alguien como reparar un ordenador, limpiar una oficina, dar clase de idioma a alguien, hacer una conferencia ,etc. y no controlamos el tiempo que le hemos dedicado. El Fast-Work nos permite tener un control del tiempo realizado en estas tareas y además nos dará la opción de poder facturarlos.

2 . Definición general del proyecto

Gracias a las tecnologías móviles y su gran ventaja de hardware, podemos aplicar este proceso de control de horas. La gran ventaja de los móviles es que son fáciles de llevar consigo mismo y permitimos utilizar esta aplicación en cualquier lugar.

2.1 Objetivos

El proyecto que pretendo presentar es el desarrollo de una aplicación para dispositivos Android que permite controlar (en registrar, consultar y eliminar) las tareas y el tiempo que se ha podido pasar en los diferentes tipos de tarea para los diferentes clientes disponibles. De esta manera facilitar al usuario un mayor control de los tiempos que pasa en una tarea. Para que la aplicación sea operativa de diferentes maneras, tendrá un portal web donde podrá consultar todos los datos del usuario . Es decir todos los datos que estarán disponibles en el móvil podrán ser trasladados al portal web del usuario. Este portal web a parte realizar consultas , podrán sacar las estadísticas de sus tareas para los diferentes clientes.

La aplicación móvil estará programada en el lenguaje Java (Android SDK) y el portal web estará en el lenguaje de PHP. Dos lenguajes diferentes , donde cada uno tendrá un propósito diferente, para conseguir una calidad i potencia del producto.

2.2. Funcionalidades

Para poder conseguir un producto como el “Fast-Work” es necesario que cumpla las siguientes funcionalidades :

- Identificación del usuario : permitirá el acceso a los usuarios registrados al sistema.
- Control de clientes : permitirá registrar, modificar y eliminar cualquier cliente y asignar las diferentes tarea que tienen.
- Control de los tipos de tarea: permitirá registrar, modificar y eliminar cualquier tipo de tareas o actividades.
- Mostrar la información de las tareas realizadas : permitirá mostrar toda la información en tablas i gráficos , para una mejor comprensión de los datos.
- Facturar las horas de las tareas realizadas: permitirá la posibilidad de facturar las horas de tarea que se han realizado a un cliente.
- Exportación de datos del sistema : permitirá exportar los datos del sistema de diferentes formatos de: Formato Excel y Formato CSV.

3. Calendario del Proyecto

Para conseguir con éxito los objetivos del proyecto, se seguirá la evaluación continuada propuesta en la asignatura de PFC(Proyecto Final de Carrera).

La planificación seguirá el ciclo de entregas de las PAC, ofrecido en la asignatura por nuestro consultor.

3.1 Entregas establecidas

Según la programación indicada por nuestro consultor, los diferentes hitos a tener en cuenta son :

Fecha de entregar	Hito evaluación continuada	Definición
11/03/2013	PAC1	Plan de trabajo
08/04/2013	PAC2	Análisis funcional, diseño técnico y prototipo
20/05/2013	PAC3	Implementación
10/06/2013	Entrega Final	Memoria y video con la presentación del proyecto

3.2 Calendario

Id	Nombre de Tarea	Duració (días)	Comienzo	Fin
1	PFC-FastWork	110	20/02/13	10/06/13
2	PAC1 - Plan de Trabajo	19	20/02/13	11/03/13
3	Análisi de Información sobre el PFC	6	20/02/13	26/02/13
4	Preparacion del informe previo	2	26/02/13	28/02/13
5	Preparacion del plan de trabajo	4	28/02/13	04/03/13
6	Encuentro virtual	1	04/03/13	05/03/13
7	Editar el Plan de trabajo	4	05/03/13	09/03/13
8	Revisión del Plan de trabajo	2	09/03/13	11/03/13
9	Entrega PAC1-Plan de trabajo	0	11/03/13	11/03/13
10	PAC2-Análisi funcional, diseño técnico y prototipo	28	11/03/13	08/04/13
11	Preparacion de l'entorno de desarrollo	4	11/03/13	15/03/13
12	Descargar los software	1	11/03/13	12/03/13
13	Instalacion i configuracion de los software	2	12/03/13	14/03/13
14	Pruebas d'integracion de los software	1	14/03/13	15/03/13
15	Análisi funcional plataforma movil	6	15/03/13	21/03/13
16	Estudio de requerimientos i funcionalidades	2	15/03/13	17/03/13
17	Especificacion de funcionalidades	2	17/03/13	19/03/13
18	Elaborar los diagramas (UML)	2	19/03/13	21/03/13
19	Análisi funcional plataforma web	5	21/03/13	26/03/13
20	Estudio de requerimientos i funcionalidades	1	21/03/13	22/03/13
21	Especificacion de funcionalidades	2	22/03/13	24/03/13
22	Elaborar los diagramas (UML)	2	24/03/13	26/03/13
23	Diseño funcional	7	26/03/13	02/04/13
24	Definicion de l'arquitectura del sistema	1	26/03/13	27/03/13
25	Preparacion de diseño funcional	3	27/03/13	30/03/13
26	Elaboracion de diagramas de base de datos	1	30/03/13	31/03/13
27	Elaboracion de diagramas classes	2	31/03/13	02/04/13
28	Prototipo	4	02/04/13	06/04/13
29	Preparacion d'interfaz graficas	3	02/04/13	05/04/13
30	Preparacion de los documentos de la interfaz grafica	1	05/04/13	06/04/13
31	Recopilacion del documento de analisis funcional,diseño tecnico y prototipo	1	06/04/13	07/04/13
32	Revisión del documento d'análisi funcional,diseño tecnico y prototipo	1	07/04/13	08/04/13
33	Entregar PAC2 - Analisis funcional,diseño tecnico y prototipo	0	08/04/13	08/04/13
34	PAC3-Implementacion	42	08/04/13	20/05/13
35	Implementacion plataforma movil	21	08/04/13	29/04/13
36	Desarrollo de la aplicación Android	18	08/04/13	26/04/13
37	Control de errores de la aplicacion movil	2	26/04/13	28/04/13
38	Pruebas en la aplicacion movil	1	28/04/13	29/04/13
39	Implementacion plataforma web	12	29/04/13	11/05/13
40	Creacion de la base de datos	3	29/04/13	02/05/13
41	Desarrollo del Webservice(info) + web	6	02/05/13	08/05/13
42	Control de errores	2	08/05/13	10/05/13
43	Pruebas en la aplicación Web	1	10/05/13	11/05/13
44	Pruebas de Integracion	3	11/05/13	14/05/13
45	Preparacion de las pruebas de integracion	1	11/05/13	12/05/13
46	Pruebas d'integracion en la plataforma web/movil	2	12/05/13	14/05/13
47	Despliegamiento i pruebas	4	14/05/13	18/05/13
48	Despliegamiento	1	14/05/13	15/05/13
49	Pruebas	3	15/05/13	18/05/13
50	Revisión y correcciones de la implementacion	2	18/05/13	20/05/13
51	Entregar PAC3 - Implementacion	0	20/05/13	20/05/13
52	Entregar Final	21	20/05/13	10/06/13
53	Preparacion de la memoria	12	20/05/13	01/06/13
54	Preparacion del documento final	12	20/05/13	01/06/13
55	Preparacion de la presentacion	8	01/06/13	09/06/13
56	Preparacion de la presentacion (PowerPoint)	4	01/06/13	05/06/13
57	Preparacion del video	4	05/06/13	09/06/13
58	Revisión de la Entregar Final	1	09/06/13	10/06/13
59	Entregar de la Entregar Final	0	10/06/13	10/06/13

Aplicamos el día 20/02/2013 como inicio del proyecto y con fecha final el día 10/06/2013.

En el calendario anterior nos muestra todos los detalles y fechas de cada punto del proyecto Fast-Work.

Como podemos ver en el calendario anterior disponemos de 110 días para cumplir nuestro objetivo de este proyecto.

3.3 Detalle de entrega para la PAC1

Id	Nombre de Tarea	Duració (días)	Comienzo	Fin
2	PAC1 - Plan de Trabajo	19	20/02/13	11/03/13
3	Análisi de Información sobre el PFC	6	20/02/13	26/02/13
4	Preparacion del informe previo	2	26/02/13	28/02/13
5	Preparacion del plan de trabajo	4	28/02/13	04/03/13
6	Encuentro virtual	1	04/03/13	05/03/13
7	Editar el Plan de trabajo	4	05/03/13	09/03/13
8	Revisión del Plan de trabajo	2	09/03/13	11/03/13
9	Entrega PAC1-Plan de trabajo	0	11/03/13	11/03/13

La PAC1 tiene como objetivo de analizar y planificar el proyecto

3.4 Detalle de entrega para la PAC2

Id	Nombre de Tarea	Duració (días)	Comienzo	Fin
10	PAC2-Análisis funcional, diseño técnico y prototipo	28	11/03/13	08/04/13
11	Preparacion de l'entorno de desarrollo	4	11/03/13	15/03/13
12	Descargar los software	1	11/03/13	12/03/13
13	Instalacion i configuracion de los software	2	12/03/13	14/03/13
14	Pruebas d'integracion de los software	1	14/03/13	15/03/13
15	Analisi funcional plataforma mobil	6	15/03/13	21/03/13
16	Estudio de requerimientos i funcionalidades	2	15/03/13	17/03/13
17	Especificacion de funcionalidades	2	17/03/13	19/03/13
18	Elaborar los diagramas (UML)	2	19/03/13	21/03/13
19	Analisi funcional plataforma web	5	21/03/13	26/03/13
20	Estudio de requerimientos i funcionalidades	1	21/03/13	22/03/13
21	Especificacion de funcionalidades	2	22/03/13	24/03/13
22	Elaborar los diagramas (UML)	2	24/03/13	26/03/13
23	Diseño funcional	7	26/03/13	02/04/13
24	Definicion de l'arquitectura del sistema	1	26/03/13	27/03/13
25	Preparacion de diseño funcional	3	27/03/13	30/03/13
26	Elaboracion de diagramas de base de datos	1	30/03/13	31/03/13
27	Elaboracion de diagramas classes	2	31/03/13	02/04/13
28	Prototipo	4	02/04/13	06/04/13
29	Preparacion d'interfaz graficas	3	02/04/13	05/04/13
30	Preparacion de los documentos de la interfaz grafica	1	05/04/13	06/04/13
31	Recopilacion del documento de analisis funcional,diseño tecnico y prototipo	1	06/04/13	07/04/13
32	Revision del documento d'analisi funcional,diseño tecnico y prototipo	1	07/04/13	08/04/13
33	Entregar PAC2 - Analisis funcional,diseño tecnico y prototipo	0	08/04/13	08/04/13

La PAC2 tiene como objetivo preparar el entorno de desarrollo, analizar la funcionalidad para la plataforma móvil y web, el diseño funcional y para acabar el prototipo.

3.5 Detalle de entrega para la PAC3

Id	Nombre de Tarea	Duració (días)	Comienzo	Fin
34	PAC3-Implementacion	42	08/04/13	20/05/13
35	Implementacion plataforma movil	21	08/04/13	29/04/13
36	Desarrollo de la aplicación Android	18	08/04/13	26/04/13
37	Control de errores de la aplicacacion movil	2	26/04/13	28/04/13
38	Pruebas en la aplicacion movil	1	28/04/13	29/04/13
39	Implementacion plataforma web	12	29/04/13	11/05/13
40	Creacion de la base de datos	3	29/04/13	02/05/13
41	Desarrollo del Webservice(info) + web	6	02/05/13	08/05/13
42	Control de errores	2	08/05/13	10/05/13
43	Pruebas en la aplicación Web	1	10/05/13	11/05/13
44	Pruebas de integracion	3	11/05/13	14/05/13
45	Prepacion de las pruebas de integracion	1	11/05/13	12/05/13
46	Pruebas d'integracion en la plataforma web/movil	2	12/05/13	14/05/13
47	Despliegamiento i pruebas	4	14/05/13	18/05/13
48	Despliegamiento	1	14/05/13	15/05/13
49	Pruebas	3	15/05/13	18/05/13
50	Revison y correcciones de la implementacion	2	18/05/13	20/05/13
51	Entregar PAC3 - Implementacion	0	20/05/13	20/05/13

La PAC3 tiene como objetivo principal la implementación de la plataforma móvil y plataforma web. Es el punto mas critico , es importante seguir muy rigurosamente la planificación de cada tarea.

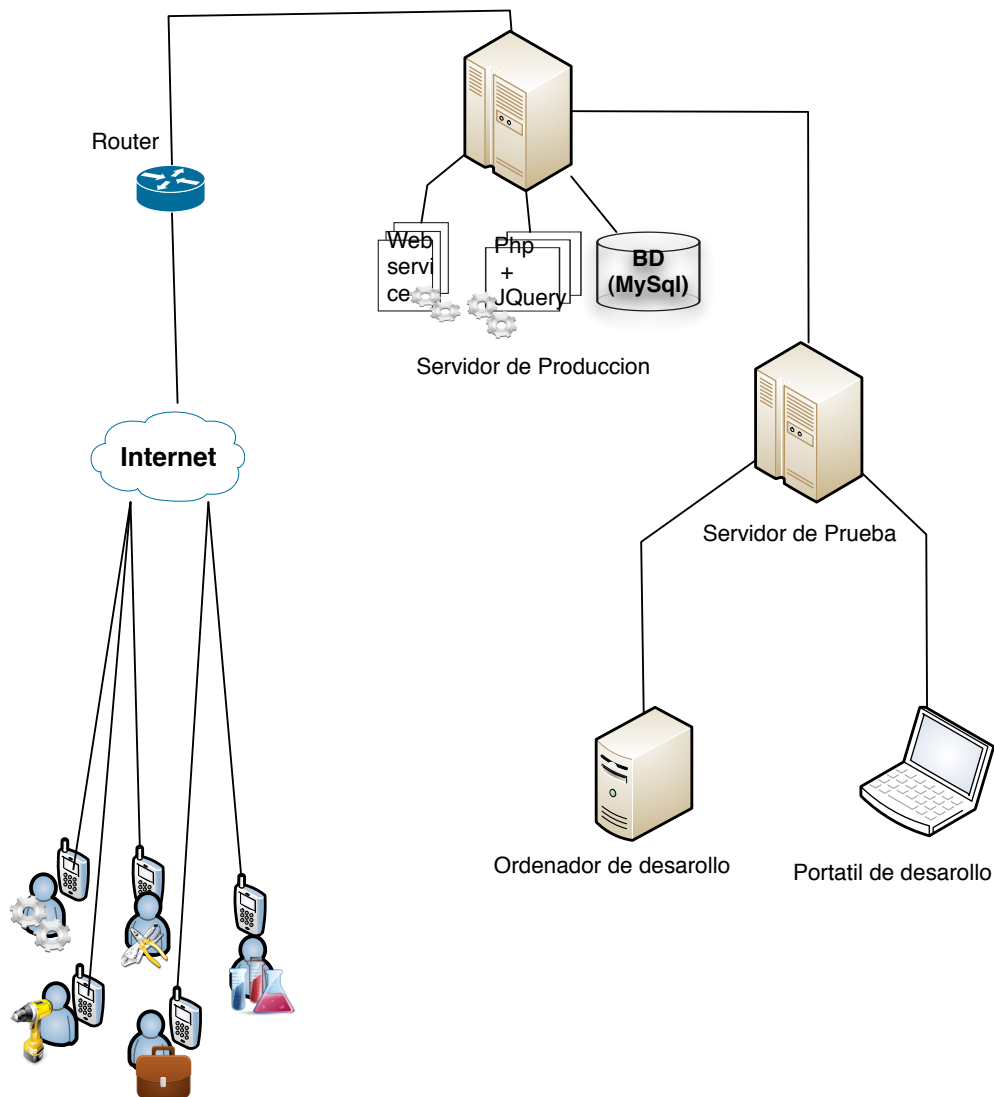
3.6 Detalle de entrega para la entregar final

52	Entregar Final	21	20/05/13	10/06/13
53	Preparacion de la memoria	12	20/05/13	01/06/13
54	Preparacion del documento final	12	20/05/13	01/06/13
55	Preparacion de la presentacion	8	01/06/13	09/06/13
56	Preparacion de la presentacion (PowerPoint)	4	01/06/13	05/06/13
57	Preparacion del video	4	05/06/13	09/06/13
58	Revison de la Entregar Final	1	09/06/13	10/06/13
59	Entregar de la Entregar Final	0	10/06/13	10/06/13

La Entregar Final del Proyecto es la recta final , tiene como objetivo preparar la memoria del proyecto, la presentación y el video. Es muy importante ser claro y conciso con las ideas, sobre todo en la presentación.

4. Recursos y infraestructura

Para el desarrollo y la implementación de este proyecto es necesario preparar un entorno de trabajo compuesto de los siguientes elementos mostrados en el siguiente esquema :



El servidor web , tendrá tres funcionalidades :

- **la funcionalidad de web** (vía Web Apache): para poder dar servicio a la aplicación web.
- **la funcionalidad de Base de datos (MySQL)** : para poder responder a todas las consultas tanto de la paginas webs como la del webservice.

- **funcionalidad de servicio web** (webservice) : para poder responder a todas las consultas i peticiones que puedan realizarse des de el móvil.

4.1 Recursos Hardware

La siguiente tabla especificar las características técnicas de los elementos que interviene en el sistema :

Dispositivo	Definición técnica	Función
Ordenador de desarrollo	- CPU : Intel Pentium 4 - Memoria RAM : 4GB - Sistema Operativo : Windows Xp Professional	Este ordenador servirá para la preparación de diseño i de desarrollo de la aplicación web
Portátil de desarrollo	- CPU : Intel Core i7 - Memoria RAM : 4GB - Sistema Operativo : Mac OS X 10.6.8	Este ordenador servirá para el desarrollo de la aplicación móvil
Servidor de Prueba	- CPU : Intel Pentium 4 - Memoria RAM : 4GB - Sistema Operativo : Windows Xp Professional	Este servidor servirá de prueba para las nuevas versiones las aplicaciones web i móvil
Servidor de Producción	- CPU : Intel Pentium 4 - Memoria RAM : 4GB - Sistema Operativo : Windows Xp Professional	Este es el servidor de prueba para las aplicación de producción : web i móvil.
Móvil Android	- HTC Hero - Android 2.1	Este móvil nos servirá para hacer pruebas i ver los resultados de los desarrollos.

4.2 Recursos de Software desarrollo

A continuación mostraré los recursos de software de desarrollo para implementar la aplicación móvil i web :

Software de desarrollo	Función
Eclipse Hero	Entorno de desarrollo para la aplicación móvil de Fast-Work
Android SDK	Librerías de acceso a las características del móvil
Microsoft Office 2010	Herramientas para elaborar los documentos del proyecto, para la presentación i esquemas.
- Microsoft Word	
- Microsoft Excel	
- Microsoft Power Point	
FireWorks 8	Software para generar i retocar las imágenes de las aplicaciones
Xamp	Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los interpretes para lenguajes de script PHP.

4.3 Recursos Software al Servidor

A continuación mostraré los recursos de Software al Servidor web:

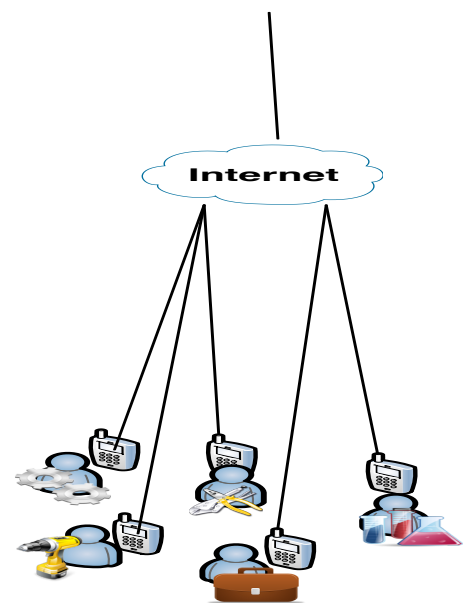
Servidor	Función
Apache Web Server	Servidor Web que permite el acceso a la aplicación web y al webService
PHP	Interprete que habilita ejecución de código escrito en PHP en el servidor Web
MySQL	Sistema gestor de base de datos que guardara la información referente a los datos del usuario
JQuery	Jquery es una biblioteca de Javascript que permite simplificar la manera de interactuar con los documentos HTML

5. Tecnología a aplicar

Este Proyecto está formado por tres partes : **la primera parte**, trata de la aplicación móvil : donde los usuario anotan los tiempo de las diferentes tareas que han realizado. **La segunda parte** trata de la sincronización de datos entre la aplicación móvil y la aplicación web: esta parte el móvil envía todos los datos a la aplicación Web vía WebService (tareas realizadas, tareas en curso, clientes creados, facturas creadas, etc..). Y **la tercera parte**, trata de un portal web : donde los usuarios pueden consultar todas las tareas realizadas por todos los móviles sincronizados.

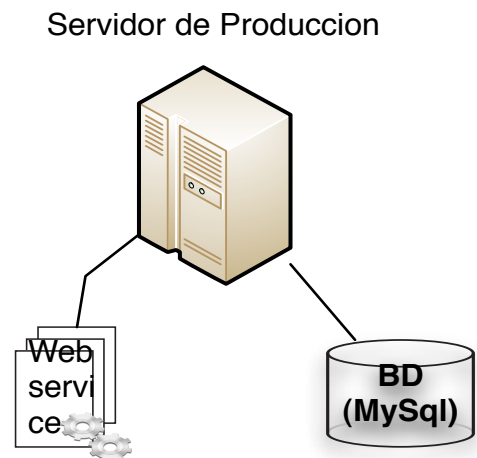
5.1 Tecnología móvil (Android)

La tecnología móvil que pondré en marcha en la aplicación será dada por el SDK de Android. Donde nos permitirá desarrollar la interfaz entre usuario y dispositivo móvil. Gracias a las librería que nos proporciona el SDK , podremos recuperar los tiempos de las diferentes tareas realizadas por el usuario, las creaciones de nuevos clientes y las nuevas facturas generadas y otras funcionalidades de Fast-Work.



5.2 Tecnología de sincronización de información(Webservice)

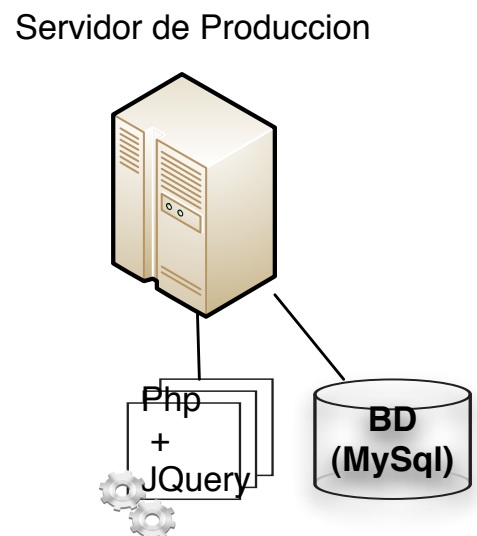
La Tecnología de sincronización será mediante un Webservice. Donde el móvil será el responsable de enviar el flujo de información hacia el servidor vía Webservice. El protocolo que utilizaremos para el Webservice será SOAP (*Simple Object Access Protocol*). Es un protocolo estándar que define como dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.



5.3 Tecnología de web

La tecnología web será basado en un servidor web habilitado para el acceso externo (vía un Apache Web Server) y modulo de PHP + JQuery. Gracias a la tecnología de PHP (Personal Home Page Tools) nos permitirá definir y ejecutar las funcionalidades del portal web de Fast-Work.

Utilizaremos la tecnología de JQuery para aportar un estilo agraciado y profesional. JQuery es una biblioteca de Javascript que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos y aportar un estilo RIA (*Rich Internet Applications*) a las páginas webs.



6.Riesgos del proyecto

Durante el desarrollo del proyecto pueden aparece ciertos eventos que pueden causar problemas para la planificación y el logro de los objetivos del proyecto. A continuación muestro los diferentes riesgos que pueden existir y el impacto que pueden causar en el proyecto :

Riesgo	Descripción	Probabilidades de aparición	Impacto	Acciones a realizar para evitar riesgo
Deterioro de equipamiento técnico	El deterioro de los equipamiento informáticos puede tener un impacto importante en los objetivos del proyecto.	Medio	critico	Realizar copias de seguridad periódicas (por cada días)
Perdida de datos en los equipamientos	A veces algunos software de desarrollo suelen colgarse y la gran consecuencia es que perdemos todo el trabajo realizado.	Medio	Muy Critico	Realizar copias de seguridad periódicas (por cada días o horas)
Planificación incorrecta del proyecto o imprevisto	Durante el desarrollo pueden aparecer errores o situación (que no se analizaron correctamente) donde existe se paso mas tiempo de lo previsto en la planificación.	Medio	Critico	Seguimiento estrictico en el calendario y realizar modificaciones continuas en la planificación de proyecto.

7 . Análisis funcional

En esta parte nos dedicaremos a definir los elementos de la fase de análisis del sistema. También definiremos los requisitos funcionales y no funcionales a partir de la especificación del caso de uso.

La aplicación de Fast Works tiene como objetivo gestionar el tiempo de cada tarea realizada por un usuario. A través de la aplicación móvil se podrá realizar todas las operaciones básicas con las tareas. A continuación detallamos estas funcionalidades :

7.1 Requerimientos funcionales

7.1.1 Descripción básica del funcionamiento

Como comentamos anteriormente , el Fast Work sigue el objetivo de gestionar las diferentes tareas, facturarlas y compartirlas. Para poder conseguir un funcionamiento completo de esta iniciativa, es necesario que la aplicación simule los tres conceptos en el que se basa :

- a) Crear la tareas y asignar su tipo de categoría: se crear la tarea de trabajo que se va a realizar y se le asigna la categoría a realizar.
- b) Cambiar el estado de la tarea : Esta tarea se podrá comenzar, pausar y finalizar a gusto del usuario.
- c) Facturar las tareas : Una vez finalizadas todas las tareas , pasan a ser facturadas. También con posibilidad de enviar por mail a sus clientes.

7.1.2 Particularidades de la versión móvil i de la web

La implementación de esta aplicación para un dispositivo móvil tiene que tener la capacidad de gestionar correctamente las tareas que se van a realizar. Cogemos como punto de inicio del usuario, crear la tarea y decidir el tipo de tareas (categoría) que va realizar. Esta que la tarea no finalizar el usuario no podrá facturar la tareas.

También es importante destacar que la implementación de esta aplicación no tiene sentido sin su interacción con la plataforma web, donde proporciona al usuario un mejor control y seguridad de sus tareas. Esta plataforma , aunque no tiene la interfaz habilitada por los usuarios, ofrece un Webservice que interactúa con la Base de datos del sistema i que pueden ser consumido por la aplicación móvil, permitiendo realizar las operaciones de Fast Work.

7.1.3 Seguridad

Para poder utilizar la aplicación móvil , no hace falta registrarse. Solo hace falta registrarse en el momento de sincronizar los datos de las tareas, las categorías , los clientes y las facturas creadas. Pero para el acceso a la aplicación web si hace falta un usuario y una contraseña. De esta manera toda la información guardada en el móvil será pasada a la base de datos del servidor web, en un lugar mas seguro y quedando la información protegida. En la aplicación web contempla la funcionalidad de alta de usuario al sistema, como el inicio y fin de sesión.

Por otra parte, el hecho de que el sistema requiere de una cuenta de usuario hace que la encriptación de datos sea necesario durante la comunicación (sincronización de datos) entre las dos plataformas, de esta manera podremos garantizar que estos datos sea inaccesible para evitar cualquier acceso mal intencionado.

7.2 Requerimientos no funcionales

A continuación trataremos ciertos aspectos o recomendaciones que el propio sistema pide, que no son estrictamente relacionadas con el proyecto.

7.2.1 Interfaz

Como toda aplicación destinada a un dispositivo móvil, es necesario que su interfaz sea la más atractiva posible, a la vez que permite un uso más ergonómico. Es necesario, dar una prioridad a la lista desplegable, a los botones a mida considerables, facilitar una pulsación correcta con los dedos en la pantalla móvil donde suele ser de un tamaño pequeño.

A demás es necesario que la interfaz sea la más intuitiva posible para reducir la dudas de los usuarios finales. Se tiene que buscar una armonía en la interfaz (componentes y colores) que faciliten su comprensión en los momentos que se esta navegando por la aplicación, tiene que ser lo más claro i rápido posible.

Finalmente, es imprescindible que dada la gran variedad de dispositivos Android que existen en el mercado (y con varios formatos de pantalla que puede tener), es importante poder garantizar la correcta adaptación de la interface de la aplicación en todos los dispositivos. Para ello tendríamos que probar la aplicación en todos estos dispositivos. En esta primera versión de nuestro proyecto no es viable. Nosotros durante el desarrollo lo realizaremos con el móvil de HTC Hero.

7.2.2 uso de característica propias del dispositivo móvil

La aplicación móvil es la responsable de sincronizar la información hacia el servidor de base de datos, vía el Webservice. Por cada tarea finalizada la aplicación en registrara la posición actual del móvil, de esta manera se podrá visualizar el lugar donde se a realizado la tarea en la aplicación web.

Es importante tener en cuenta el consumo de la batería , ya que la aplicación móvil estará despierta durante el tiempo que se este realizando las tareas. Por lo cual tendremos que tener en cuenta en todo momento este consumo de batería.

7.3 Funcionalidad del sistema

Dadas las funcionalidades descritas anteriormente, es necesario definir las funcionalidades exigidas y implementarlas al sistema , donde van ha resolver las necesidades descritas anteriormente. Tal como se a describió en el punto 1.1.2 , las funcionalidades a implementar tiene que interactuar a través de las dos plataformas del sistema y cooperar para ofrecer las ventajas que proporciona cada una de ellas. A continuación describo estas funcionalidades y su interacción :

7.3.1 Plataforma móvil

Des de la aplicación móvil es necesario realizar las siguientes operaciones:

- Asignar usuario:
Esta función proporciona al usuario la creación de un perfil del sistema para poder acceder al servicio de Fast Work.

- Crear tareas :
Esta función permite que el usuario cree una tarea, donde estará su titulo, descripción, el cliente, la fecha inicio, fecha final y el precio a considerar.

- Mostrar tareas :
Esta función permite visualizar toda la información que podría tener la tarea y su estado que tiene actualmente (Pendiente, En curso, Pausada o Finalizada)

- Modificar tareas:
Esta función permite modificar toda la información de una tarea como su título, descripción, el cliente, la fecha inicio, fecha final y el precio.

- Eliminar tareas:
Esta función permite eliminar una tarea existen en la base de datos.

- Crear categorías :
Esta función permite crear una categorías (es decir un tipo de tarea) donde estará su nombre y su precio por defecto de esta categoría.

- Mostrar la categorías:
Esta función permite mostrar toda la información de la categoría como el nombre de la categoría y sur precio por defecto.

- Modificar la categoría :
Esta función permite modificar la categoría , es decir el nombre y el precio por defecto.

- Eliminar categorías:
Esta función permite eliminar la categoría.

- Crear cliente :
Esta función permite crear al cliente, que estará constituido del nombre cliente, dirección y teléfono.

- Mostrar cliente :
Esta función permite mostrar la información del nombre del cliente , dirección y teléfono.

- Eliminar cliente :
Esta función permite eliminar un cliente.

- Crear factura :
Esta función te permite crear una factura a partir de las tareas realizadas.

- Mostrar factura:
Esta función te permite mostrar el valor de la factura y las tareas realizadas.

- Modificar factura:
Esta función te permite modificar la facturar, eliminando o añadiendo las tareas que tiene que corresponder con la factura.

- Eliminar facturar:
Esta función te permite eliminar la factura y poner a estado realizado las tareas.

7.3.2 Plataforma web

Des de la aplicació web es necessari realitzar les següents operacions :

➤ Registrar usuario :

Esta función te permite crear un usuario con sus datos como el nombre , apellido , email y contraseña.

➤ Iniciar sesión :

Esta función facilitar al usuario el poder de acceder al servicio de Fast Work y de esta manera utilizar las otras funcionalidades. Esta identificación se realiza introduciendo el nombre del usuario y la contraseña.

➤ Finalizar sesión :

Esta función permite a un usuario salir del sistema.

➤ Consultar las tareas :

Esta función permite mostrar todas las tareas existen en la base de datos del servidor web.

➤ Consultar las categorías de las tareas :

Esta función permite mostrar todas las categorías de las tareas existentes en la base de datos del servidor web.

➤ Consultar los clientes:

Esta función permite mostrar todas los clientes existentes en la base de datos del servidor web.

➤ Consultar las facturas :

Esta función permite mostrar todas las facturas existentes en la base de datos del servidor web.

- Consultar y modificar la configuración del usuario web:
Esta función permite consultar y modificar la configuración del usuario existentes en la base de datos del servidor web, como por ejemplo la contraseña del usuario.

7.3.3 Plataforma Webservice

La plataforma del webservice tan solo servirá para la sincronización de datos entre la plataforma móvil y la plataforma web.

Des de la interfaz Webservice es necesario realizar las siguientes operaciones :

➤ Verificación de la identificación del usuario :

Esta función verificar la identificación del usuario, si esta activo y su contraseña es la correcta.

➤ Sincronizar tareas :

Esta función permite la sincronización de las tareas des de la plataforma móvil hacia la plataforma web.

➤ Sincronizar categorías :

Esta función permite la sincronización de las categorías des de la plataforma móvil hacia la plataforma web.

➤ Sincronizar clientes :

Esta función permite la sincronización de los clientes des de la plataforma móvil hacia la plataforma web.

➤ Sincronizar facturas:

Esta función permite la sincronización de las facturas des de la plataforma móvil hacia la plataforma web.

7.4 Usuario del sistema

Los usuarios que intervendrá en el sistema hace falta destacar que el sistema es híbrido , donde intervienen por una parte web y otra parte móvil.

Los diferentes perfiles del usuario son los siguientes:

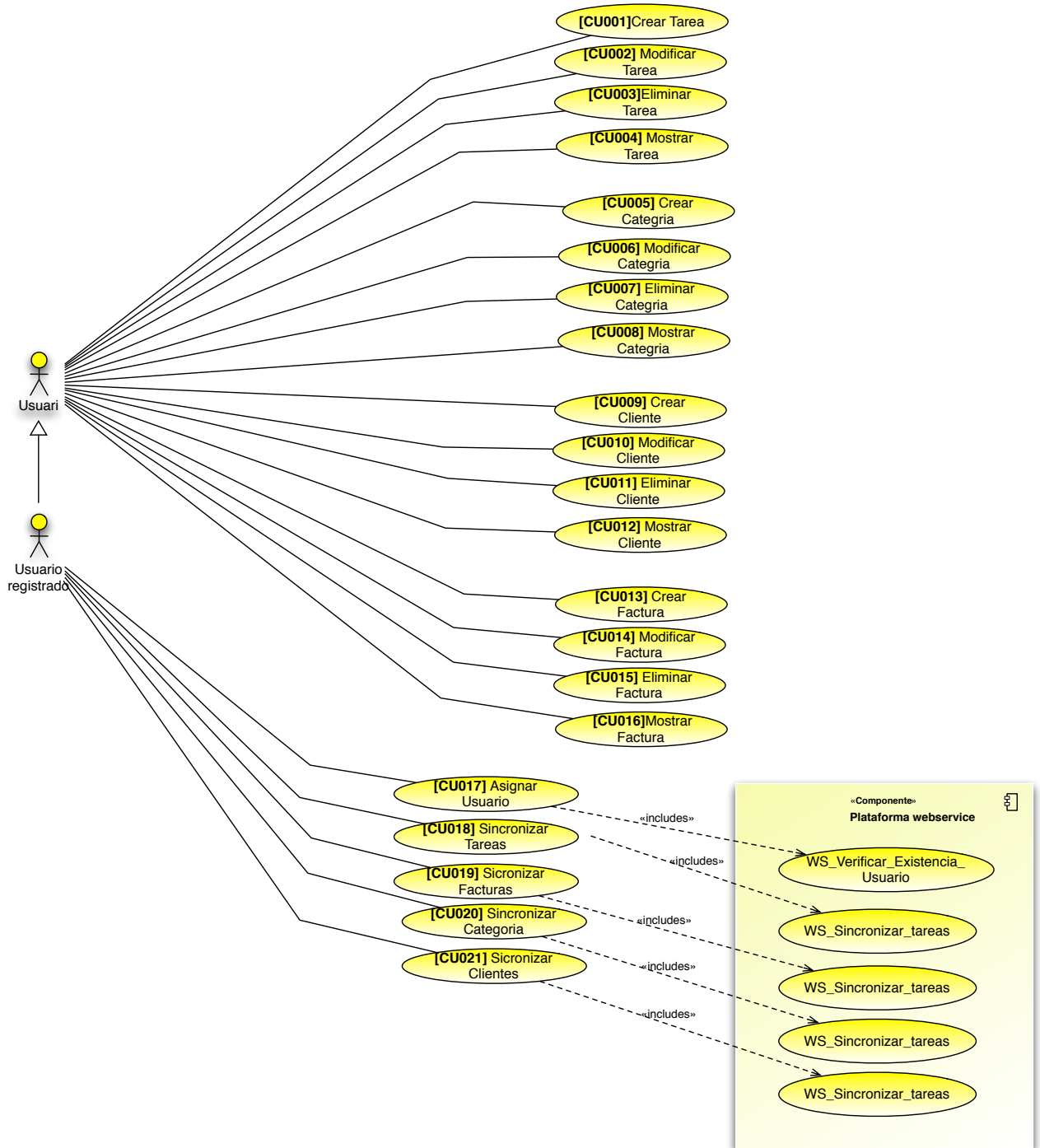
- Usuario móvil : representar cualquier usuario que crea tareas en el móvil , sin tener que identificarse.

- Usuario web : representar un usuario Web , donde tiene que identificarse bajo un usuario y contraseña, para acceder a toda la información (tareas, categoría, facturas y clientes).

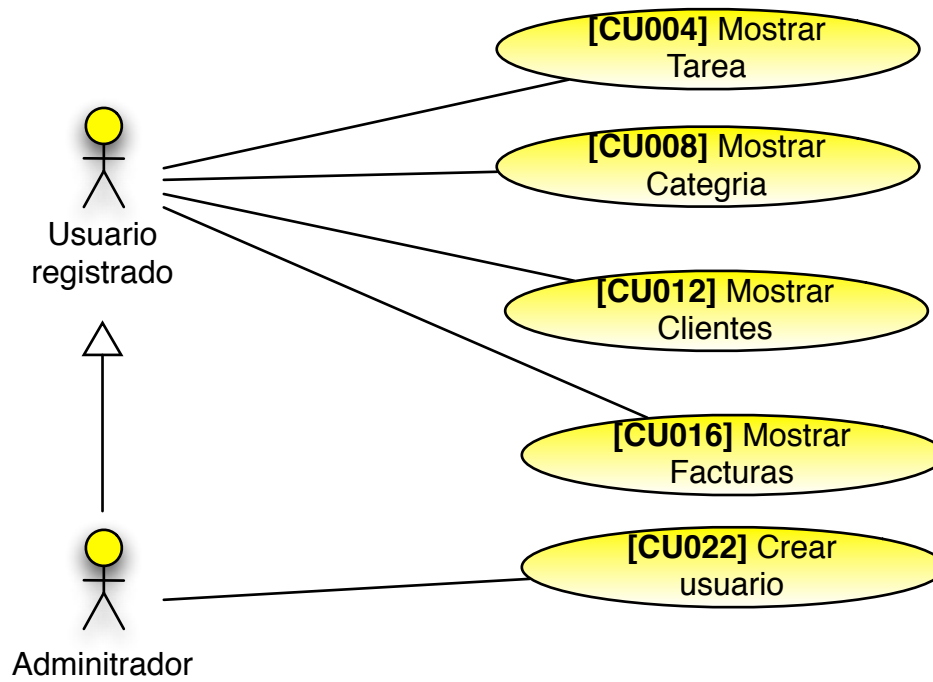
- Administrador : Tiene el acceso a todas la información del sistema y también es el responsable de crear los usuarios.

7.5 Caso de uso

A partir de la identificación de los actores que interviene en la aplicación, pueden recoger en una vista global el caso de uso donde describen todas las funcionalidades de la aplicación móvil y interfaz Webservice:



El caso de uso de las funcionalidades de la aplicación web :



7.5.1 Descripción del caso de uso relacionado con Tareas

Identificador:	CU001, CU002, CU003, CU004
Nombre:	Crear, Modificar , eliminar , Mostrar Tarea
Resumen:	Estas 4 funcionalidades realizan toda la gestión de una tarea en el momento de crear, modificar , eliminar y mostrar Tarea
Actores:	Usuario registrado
Precondiciones:	El usuario registrado eliminar, consultar , crea y modificar la tarea
Flujo normal:	<p>CU001(crear) :</p> <p>1 ->El usuario seleccionara la acción de crear una tarea</p> <p>2 -> El sistema le mostrara un formulario donde podrá introducir toda la información</p> <p>3-> El sistema guardara la información</p> <p>CU002(Modificar) :</p> <p>1 ->El usuario seleccionara la tarea que desea modificar</p> <p>2 -> El sistema le mostrara un formulario donde podrá modificar la información.</p> <p>3-> El sistema guardara dicha modificación de la tarea</p> <p>CU003(eliminar):</p> <p>1 ->El usuario seleccionara la tarea que desea eliminar</p> <p>2 -> El sistema le notificara una confirmación si se a realizado correctamente la eliminación.</p> <p>CU004(Mostar):</p> <p>1 ->El usuario seleccionara la tarea que desea visualizar</p> <p>2 -> Le mostrar un formulario donde podrá visualizar la información.</p>
Flujo alternativo:	En caso de algún error o falta de datos el sistema informa vía una notificación el detalle del fallo o error.
Inclusiones:	ninguno

Extensiones:	Mostrar la lista de Tareas.
--------------	-----------------------------

7.5.2 Descripción del caso de uso relacionado con Categoría

Identificador:	CU005, CU006, CU007, CU008
Nombre:	Crear, Modificar , eliminar , Mostrar Categoría
Resumen:	Estas 4 funcionalidades realizan toda la gestión de una Categoría en el momento de crear, modificar , eliminar y mostrar Categoría
Actores:	Usuario registrado
Precondiciones:	El usuario registrado eliminar, consultar , crea y modificar la Categoría
Flujo normal:	<p>CU005(crear) :</p> <ol style="list-style-type: none"> 1 ->El usuario seleccionara la acción de crear usuario 2 -> El sistema le mostrara un formulario donde podrá introducir la información. 3-> El sistema guardara la información <p>CU006(Modificar) :</p> <ol style="list-style-type: none"> 1 ->El usuario seleccionara la Categoría que desea modificar 2 -> El sistema le mostrara un formulario donde podrá modificar la información. 3-> El sistema guardara dicha modificación de la Categoría. <p>CU007(eliminar):</p> <ol style="list-style-type: none"> 1 ->El usuario seleccionara la Categoría que desea eliminar 2 -> El sistema le notificara una confirmación si se a realizado correctamente la eliminación. <p>CU008(Mostar):</p> <ol style="list-style-type: none"> 1 ->El usuario seleccionara la Categoría que desea visualizar 2 -> Le mostrar un formulario donde podrá visualizar la

	información.
Flujo alternativo:	En caso de algún error o falta de datos el sistema informa vía una notificación el detalle del fallo o error.
Inclusiones:	ninguno
Extensiones:	Mostrar la lista de Categorías.

7.5.3 Descripción del caso de uso relacionado con Clientes

Identificador:	CU009, CU010, CU011, CU012
Nombre:	Crear, Modificar , eliminar , Mostrar Clientes
Resumen:	Estas 4 funcionalidades realizan toda la gestión de clientes en el momento de crear, modificar , eliminar y mostrar cliente
Actores:	Usuario registrado
Precondiciones:	El usuario registrado eliminar, consultar , crea y modificar el Cliente
Flujo normal:	<p>CU009(crear) :</p> <ol style="list-style-type: none"> 1 ->El usuario seleccionara la acción de crear un cliente 2 -> El sistema le mostrara un formulario donde podrá introducir toda la información 3-> El sistema guardara la información <p>CU010(Modificar) :</p> <ol style="list-style-type: none"> 1 ->El usuario seleccionara el cliente que desea modificar 2 -> El sistema le mostrara un formulario donde podrá modificar la información. 3-> El sistema guardara dicha modificación del cliente <p>CU011(eliminar):</p> <ol style="list-style-type: none"> 1 ->El usuario seleccionara el cliente que desea eliminar 2 -> El sistema le notificara una confirmación si se a realizado

	correctamente la eliminación. CU012(Mostar): 1 ->El usuario seleccionara el cliente que desea visualizar 2 -> Le mostrar un formulario donde podrá visualizar la información.
Flujo alternativo:	En caso de algún error o falta de datos el sistema informa vía una notificación el detalle del fallo o error.
Inclusiones:	ninguno
Extensiones:	Mostrar la lista de Clientes.

7.5.4 Descripción del caso de uso relacionado con Facturas

Identificador:	CU013, CU014, CU015, CU016
Nombre:	Crear, Modificar , eliminar , Mostrar Facturas
Resumen:	Estas 4 funcionalidades realizan toda la gestión de facturas en el momento de crear, modificar , eliminar y mostrar la Factura
Actores:	Usuario registrado
Precondiciones:	El usuario registrado eliminar, consultar , crea y modificar la Factura
Flujo normal:	<p>CU013(crear) :</p> <p>1 ->El usuario seleccionara la acción de crear Factura 2 -> El sistema le mostrara un formulario donde podrá introducir toda la información 3-> El sistema guardara la información</p> <p>CU014(Modificar) :</p> <p>1 ->El usuario seleccionara la factura que desea modificar 2 -> El sistema le mostrara un formulario donde podrá modificar la información. 3-> El sistema guardara dicha modificación de la factura</p>

	<p>CU015(eliminar):</p> <p>1 ->El usuario seleccionara la factura que desea eliminar</p> <p>2 -> El sistema le notificara una confirmación si se a realizado correctamente la eliminación.</p> <p>CU016(Mostar):</p> <p>1 ->El usuario seleccionara la factura que desea visualizar</p> <p>2 -> Le mostrar un formulario donde podrá visualizar la información.</p>
Flujo alternativo:	En caso de algún error o falta de datos el sistema informa vía una notificación el detalle del fallo o error.
Inclusiones:	ninguno
Extensiones:	Mostrar la lista de facturas.

8. Diseño Técnico

La arquitectura general del proyecto de Fast Work se basa en un modelo Cliente-Servidor, de forma que ambas plataformas tienen un funcionamiento diferente.

8.1 Arquitectura Móvil

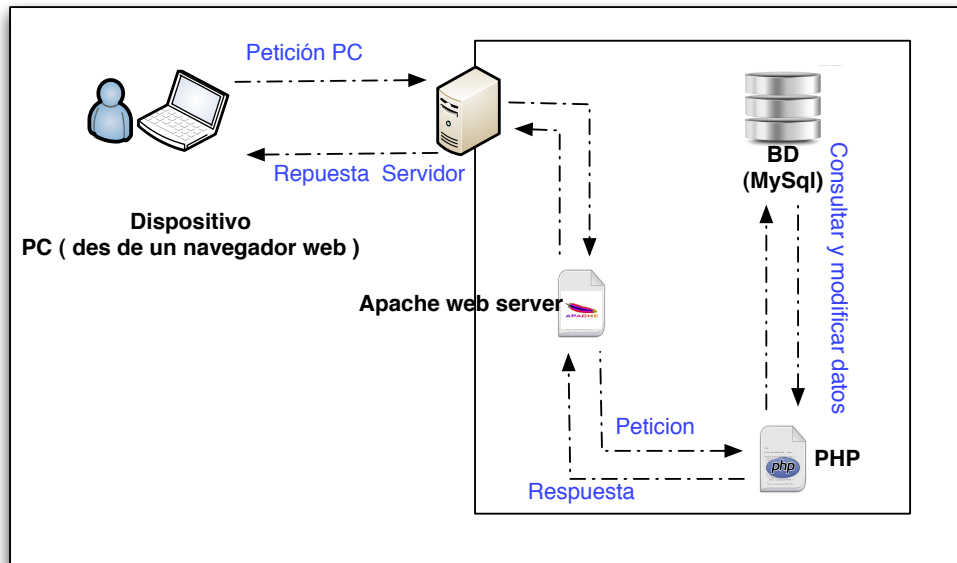
Esta plataforma móvil esta formada tan solo por la aplicación Android, que se puede instalar a varios dispositivos. En el funcionamiento de la aplicación Android no le hace falta tener conexión con la red para poder funcionar. Solamente en el momento de sincronizar los datos hacia al servidor, vía un Webservice.

8.1.1 Característica de la plataforma móvil

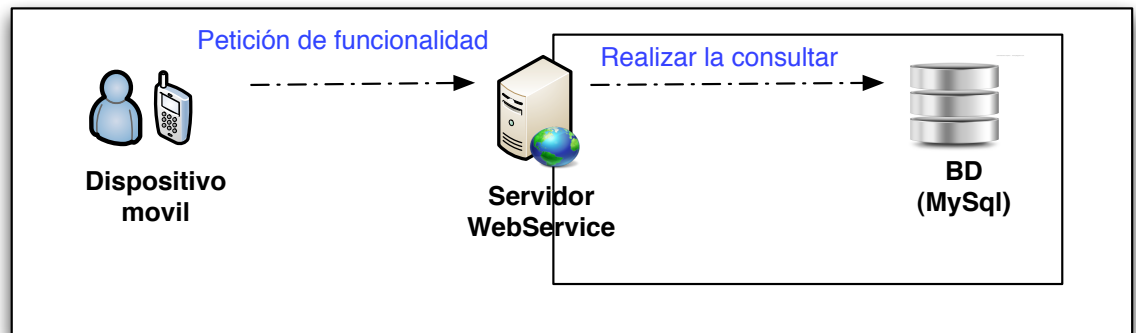
- Los clientes son la parte del sistema que inician el proceso, y envían la información hacia el servidor web.
- Una vez realizada las tareas, se mantienen en espera hasta ser sincronizadas por la aplicación móvil.
- La acción de sincronizar es responsabilidad del usuario.

8.2 Arquitectura Web y Webservice (parte servidor)

La plataforma web esta formada por la aplicación web, donde brinda diversas funcionalidades. La plataforma trata de las peticiones de todos los clientes a través de una API desarrollada en PHP que se ejecuta en el servidor web Apache. Todas las peticiones de la aplicación web realizan este siguiente recorrido :



Por otra parte , el mismo servidor operara de Webservice donde interactúa entre las peticiones del móvil y la Base de datos. Todas las peticiones del móvil las realizara el servidor de Webservice donde interactúa con la Base de datos para dar una respuesta a la petición del móvil. Por ejemplo las operaciones de sincronización de tareas, facturas, etc. Pasarían por este siguiente recorrido :

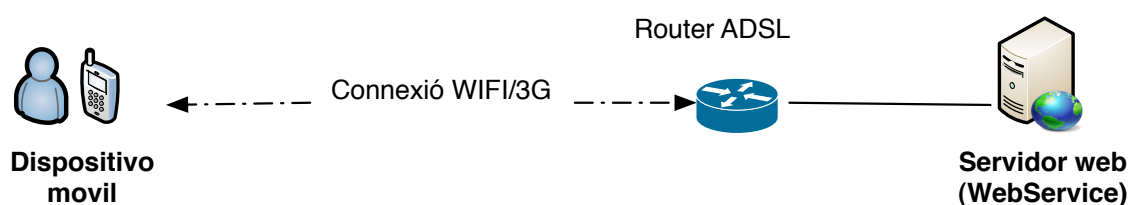


8.2.1 Característica de la plataforma web y Webservice

- Esta constantemente en ejecución, esperando las solicitudes de los clientes y con un rol completamente pasivo.
- Se encarga del procesamiento de las peticiones llegadas des del cliente y envía las respuestas correspondientes.
- Puede atender simultáneamente múltiples peticiones de los clientes.

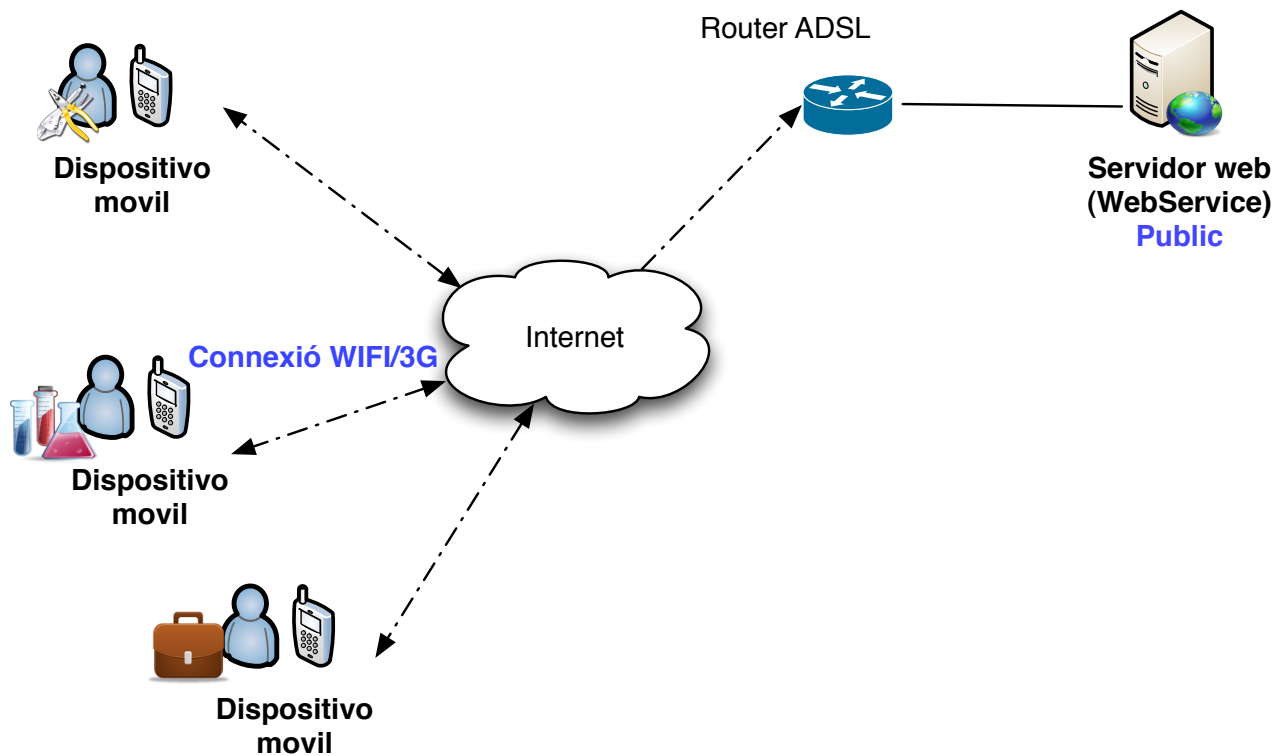
8.2 Arquitectura Física

Como se comento anteriormente, el sistema se basa en una arquitectura Cliente-Servidor, donde los clientes se pueden encontrar distribuidos geográficamente y en una movilidad pueden sincronizar sus tareas pendientes o otras peticiones. Durante la fase de desarrollo , la arquitectura del sistema va ser reducida para evitar este flujo de información innecesario, como se puede ver en el siguiente esquema :



Durante las pruebas para la interfaz móvil y web se realizarán en una red interna, donde tendremos en un único servidor la base de datos, la aplicación web y el Webservice que gestiona las peticiones de la aplicación móvil.

Una vez que todas las pruebas allí salieron exitosamente, pondremos a disposición una arquitectura pública a la red de internet. Será una estructura de producción como se detalla en el siguiente esquema:



8.3 Arquitectura Lógica

La arquitectura lógica sigue el siguiente paradigma del Modelo Vista Controlador. Donde estará distribuida en tres fases :

- La parte de la vista (Interfaz)
- La parte de negocio (controlador)
- La parte de los datos (modelo)

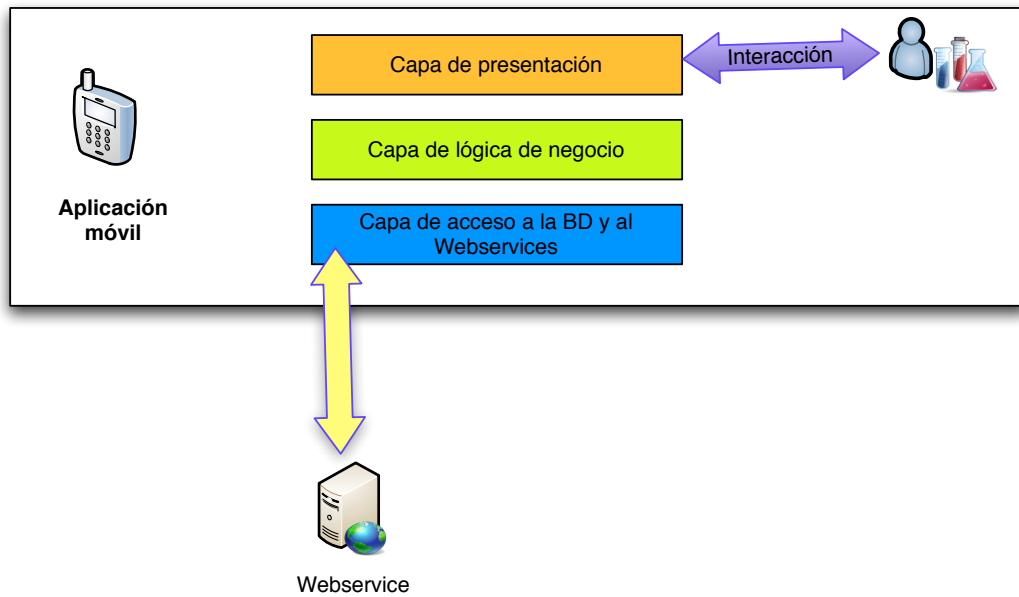
De esta manera pienso que se conseguirá un bajo nivel de acoplamiento y una alta cohesión entre los diferentes elementos de la aplicación.

Es un caso particular , la parte vista esta ligada al dispositivo móvil (donde se encargan de interactuar con el usuario), el modelo esta situado a la parte del servidor (donde gestiona los datos) y el controlador se sitúa en ambas partes, ya que comparten la responsabilidad entre las plataformas durante la ejecución de las funcionalidades.

En los siguientes puntos definiremos la arquitectura lógica para cada plataforma:

8.3.1 Arquitectura lógica de la plataforma móvil

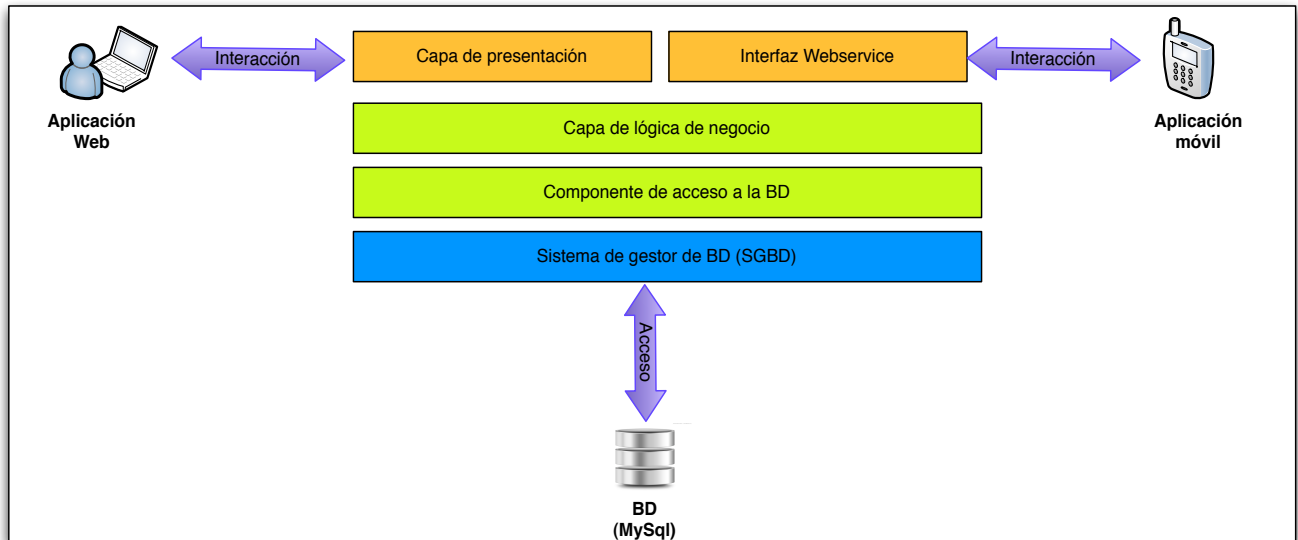
La aplicación del Modelo-Vista-Controlador en la plataforma móvil es basada en un desarrollo en 3 capas diferentes :



- Capa de presentación : se encarga de gestionar la interfaz con el usuario. Esta capa contiene las interfaces visuales i los métodos que solicitan y comunica las acciones de los usuarios a la capa inferior(capa de lógica de negocio), mostrando los resultados adaptados al dispositivo. Es necesario que esta capa tenga un buen diseño ya que es lo que ve el usuario durante el tiempo que navega en la aplicación.
- Capa de negocio : esta capa representa el núcleo de la aplicación. Todas las funcionalidades del sistema de la aplicación se centran en esta capa. Se encarga de implementación de las funcionalidades principales de la aplicación, estableciendo el canal de comunicación entre la interface de usuario y la capa de acceso a la Base de datos (o al Webservice, dependiendo de la función solicitada).
- Capa de acceso a la Base de datos y al Webservice : Es la responsable de gestionar el acceso a la Base de datos interna del móvil y el acceso al Webservice. En el caso de acceder a la Base de datos interna no seria necesario la conexión WiFi, pero para el caso del acceso al WebService si que lo es.

8.3.2) Arquitectura lógica de la plataforma web

Como se comento anteriormente, la plataforma web aplica también el esquema Modelo-Vista-Controlador, estableciendo una estructura del sistema por capas que facilitar un bajo acoplamiento y una alta cohesión :



- Capa de presentación : Se encargar de gestionar la interfaz con el usuario , pero esta vez en la aplicación web. Esta capa contiene las interfaces web visuales y los métodos que solicitar el usuario. Donde cada acción del usuario comunicar hacia la capa inferior (capa de lógica de negocio), mostrando los resultados adaptados al PC.
- Capa de Interfaz Webservice: en esta capa el usuario no interactúa con una interfaz visual directamente, sino que es la aplicación móvil en el momento de sincronizar los datos. La capa interfaz Webservice es la que utilizar el web XML, actuando como una interfaz externa de la lógica de negocio. De esta manera la aplicación móvil recibirá todos los resultados y enviara todas las peticiones pertinentes.
- Capa de lógica de negocio : como en el caso de la plataforma móvil, contiene la implementación de las funcionalidades principales del Webservice. Esta lógica es

la que requiere el acceso a la base de datos y es la única capa de todo el sistema que puede solicitarlo.

- Capa de Componentes de acceso a Base de datos : es la capa que facilita el acceso a los datos , donde se oculta la implementación de la base de datos al resto de capas. De esta manera se puede conseguir la independencia de los datos y la aplicación.

- Capa de gestor de base de datos : es la capa que gestiona la persistencia de los datos y ofrece los mecanismos necesarios para la recuperación y almacenamiento de los datos.

8.3.3) Arquitectura lógica de intercambio de información entre plataformas

Para una buena comunicación entre la dos plataforma (móvil y web) donde existirá un enlace de intercambio de datos será respaldado por la tecnología Webservice (ubicada en el servidor web). Un Webservice es un conjunto de protocolos y estándares que sirven de intercambio de datos entre dos aplicaciones. De esta manera, las diferentes aplicaciones implicadas pueden ser desarrollables en lenguajes diferentes y ejecutarse en plataformas diferentes y intercambiar datos de forma transparente. En nuestro caso la plataforma web es en lenguaje PHP y ejecutada en el servidor , y la plataforma móvil es en lenguaje JAVA y ejecutada en un móvil.

Este intercambio es posible a través del estándar de código abierto SOAP (Simple Object Acces Protocol) y XML-RPC (XML Remote Procedure Call).

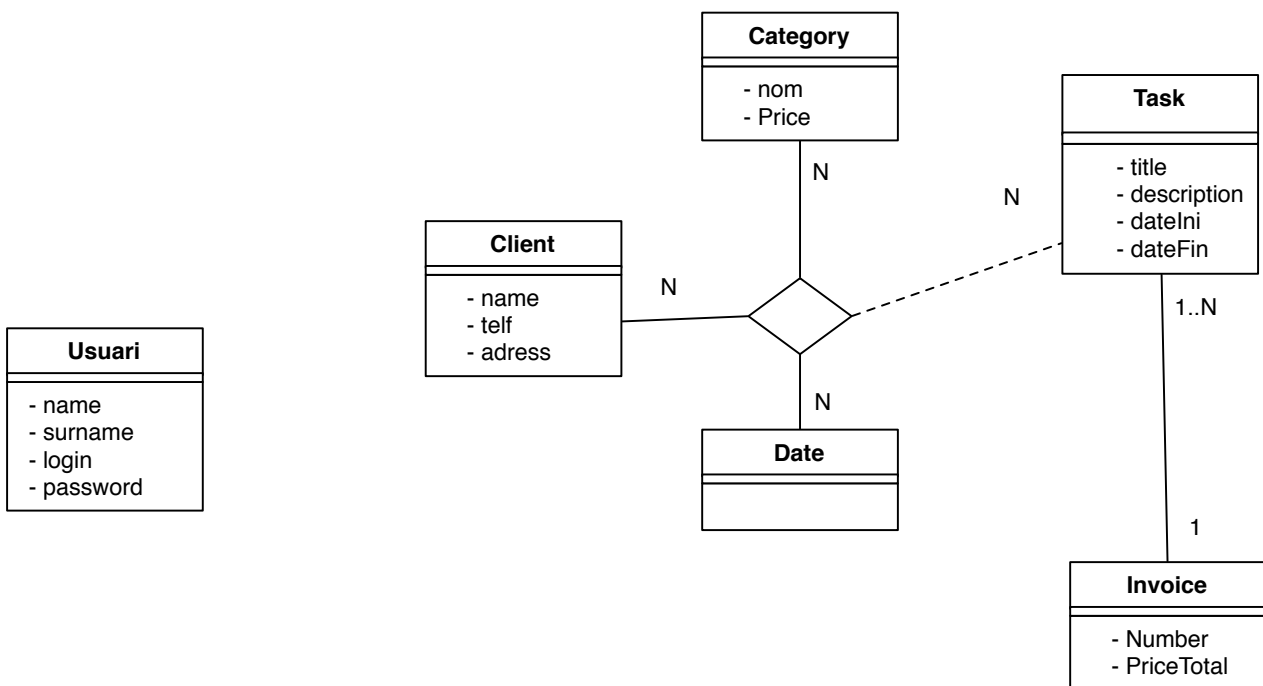
El Webservice facilita un conjunto de métodos que pueden ser invocados desde la aplicación externa y que son definidas mediante WSDL(Web Services Description Language). Gracias al protocolo SOAP las dos interfaces se comunican mediante ficheros estándares de XML.

8.4 Arquitectura de Base de datos

La Base de datos del sistema tiene que mantener la persistencia de los datos referente a los usuario, las tareas realizadas, las facturas, los clientes y las categorías. A continuación definiremos el modelo de relación de las tablas :

8.4.1 Modelo relacional de la base de datos

La Base de datos del proyecto, que es accesible des la aplicación móvil y des de el servidor web, tienen que mantener la información de los usuarios, tareas , categorías, facturas y clientes. A continuación mostramos el diseño de la base de datos del móvil :



Entidad	Descripción	cardinalidad	Entidad relacionada
Lógica			

Task	Mantiene la persistencia de los datos relacionado con las Tareas	1:N	Category
		1:N	Client
		1:01	Factura
Categoria	Mantiene la persistencia de los datos relacionados con las Categorías	1:N	Task
Client	Mantiene la persistencia de los datos relacionados con el Client	1:N	Task
Factura	Mantiene los datos relacionados con la Factura	1:N	Task

8.4.1 Tareas

La tabla de task (tarea) mantendrá el siguiente resultado:

Atributo	Descripción	Tipo	Longitud	Clave
id	Identificador la la tarea	In		Si
title	Guarda el titulo de la tarea	Texto	40	no
descripcion	Guardar la descripción de la tarea	Texto	40	No
dateIni	Guarda la fecha de inicio de la tarea con el formato dd/mm/yyyy hh:mm:ss	Fecha	20	No
dateFin	Guarda la fecha de fin de la tarea con el formato dd/mm/yyyy hh:mm:ss	Fecha	20	No

8.4.2 Categoría

La tabla de Category (categoría) mantendrá el siguiente resultado:

Atributo	Descripción	Tipo	Longitud	Clave
Name	Guarda el nombre de la categoría	Texto	40	Si
Price	Guarda el precio por defecto en la categoría	Número	40	No

8.4.3 Cliente

La tabla de Client (Cliente) mantendrá el siguiente resultado:

Atributo	Descripción	Tipo	Longitud	Clave
Name	Guarda el nombre del Client	Texto	40	Si
Telf	Guarda el número de teléfono del Client	Texto	40	No
address	Guarda la dirección del client	Texto	60	No

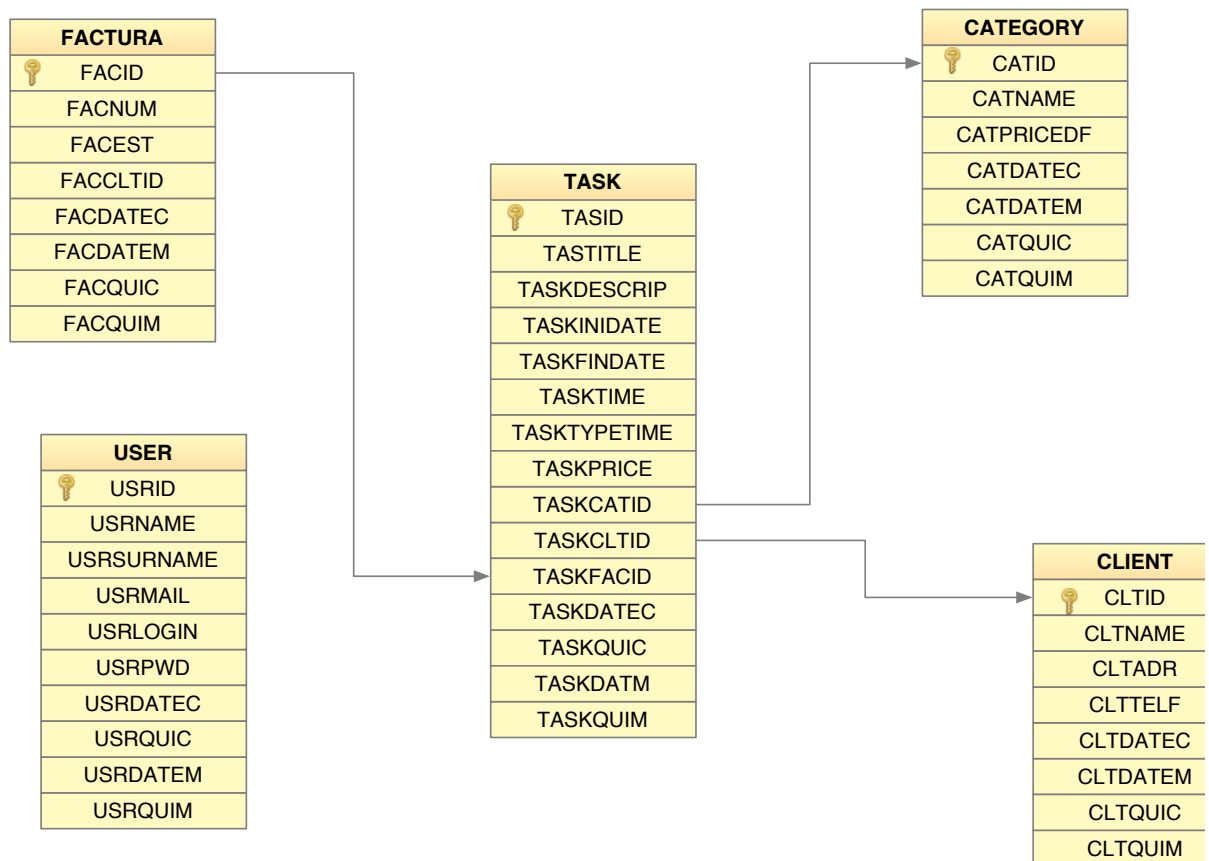
8.4.4 Factura


La tabla de Invoice (Factura) mantendrá el siguiente resultado:

Atributo	Descripción	Tipo	Longitud	Clave
number	Guarda el numero de la factura	Texto	40	Si
PriceTotal	Guarda el precio total de la factura	Número	50	No

8.5 Diagrama de Base de datos

8.5.1 Diagrama de base de datos de la plataforma móvil i web



En el anterior diagrama mostramos la estructura de la Base de datos más detalladamente. Como podemos ver marcamos con el icono  para mostrar la clave de cada tabla como TASID (Tarea),CATID (Categoría),CLTID (Cliente) ,USRID (usuario) y

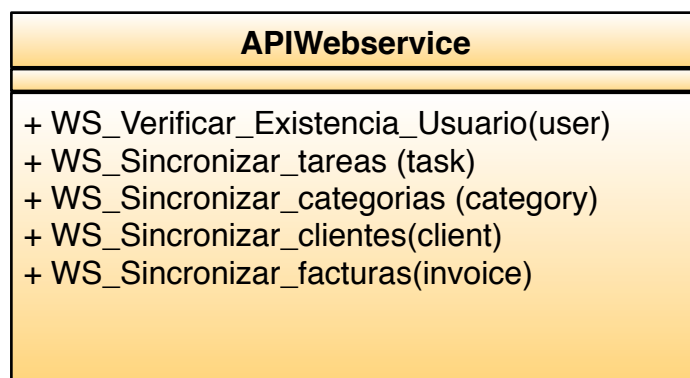
FACID (Factura).

Esta estructura de base de datos será aplicada para las dos aplicaciones : móvil y web. La razón de esta decisión es para agilizar el procesos de sincronizar los datos, donde la pequeña Base de datos del móvil alimenta la gran Base de datos del servidor Web.

8.5.2) Diagrama de clases de la plataforma Webservice

El Webservice esta ubicado en el servidor Web , su función principal es alimentar de datos actuales a la Base de datos del Servidor Web.

A continuación mostramos los diferentes métodos de esta interfaz :



Clase	Método	Descripción
APIWebservice	WS_Verificar_Existencia_Usuario	Hace una petición al Webservice para verificar si el login y password del usuario son correctos. Sus posible resulta de la petición es : True (usuario correcto) o False (Usuario incorrecto)
APIWebservice	WS_Sincronizar_Tareas	Hace una petición al Webservice para insertar la

tarea en la Base de datos del servidor web.

Su posible respuesta es :
True (tarea sincronizada correctamente) o False (tarea no sincronizada)

APIWebservice WS_Sincronizar_Categoria

Hace una petición al Webservice para insertar la categoría en la Base de datos del servidor web.

Su posible respuesta es :
True (categoría sincronizada correctamente) o False (categoría no sincronizada)

APIWebservice WS_Sincronizar_Clientes

Hace una petición al Webservice para insertar la Cliente en la Base de datos del servidor web.

Su posible respuesta es :
True (cliente sincronizada correctamente) o False (cliente no sincronizada)

APIWebservice WS_Sincronizar_Facturas

Hace una petición al Webservice para insertar la factura en la Base de datos del servidor web.

Su posible respuesta es :

True (factura sincronizada correctamente) o False (factura no sincronizada)

9. Prototipo y funcionamiento

Como se comento anteriormente, la plataforma móvil y web serán las únicas responsables de implementar la interfaz visual de interacción con el usuario. A continuación presentare los prototipos de algunas pantallas que pertenecen a la interfaz móvil y web. Estas pantallas sirven de referencia al usuario, durante la fase de desarrollo, para aportar sus comentarios y facilitar los ajustes pertinentes al desarrollador.

9.1 Prototipo móvil

9.1.1 Pantalla Menú



La pantalla del menú estará dividida por 4 apartados :



Tareas, Categorías , Clientes y configuración.



En el apartado de tareas estarán todas las acciones asociadas con tareas como crear, eliminar y modificar tareas.



En el apartado de categorías estarán todas las acciones asociadas con categorías como crear, eliminar y modificar categoría.

En el apartado de cliente estarán todas las acciones asociadas con cliente como crear,

eliminar y modificar cliente.

En el apartado de configuración estarán todas las acciones asociadas a la configuración de la aplicación como el idioma, las opciones de sincronización y los filtros de las tareas a mostrar.

9.1.2 Pantalla de Tarea

Modificar Tarea

Gestion de la tarea

Categoria: TODAS LAS CATEGORIAS

Titulo: tarea 1

Descripcion: tarea 2

Gestion de tiempo

Inicio: 25/05/2013 09:31:00

Fin: 26/05/2013 09:41:00

Tiempo: 24 Horas

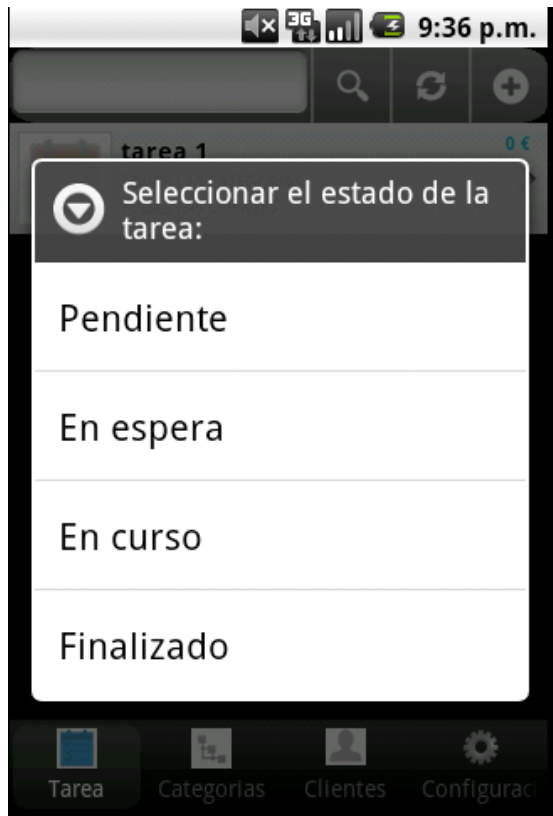
Gestion de cliente

Sincronizar:

Cliente: TODAS LAS CLIENTES

Precio: 0 € / hora

En la pantalla de tareas se podrá gestionar toda la información como la categoría, el título, la descripción, las fechas de inicio y fin, el tiempo consumido en la tarea, el cliente y la tarifa aplicada a esta tarea.



También el sistema podrá notificar el estado de la tarea. Los diferentes estados son :

- 0) Pendiente
- 1) En espera
- 2) En curso
- 3) Finalizado

9.1.3 Pantalla de Categoría



En la pantalla de categoría se podrá gestionar toda la información relacionada con categoría como su nombre, el precio por defecto y si tiene que ser sincronizado hacia el servidor Web.

9.1.4 Pantalla de Cliente



En la pantalla de cliente se podrá gestionar toda la información relacionada con el cliente como su nombre, teléfono, su dirección y si tiene que ser sincronizado hacia el servidor Web.

9.1.5 Pantalla de configuración



Esta pantalla es la de configuración. Aquí el usuario podrá elegir el idioma en que quiere navegar. Tan solo tiene dos opciones por el momento : el castellano y el catalán.

También tiene un apartado que corresponde al usuario de sincronización. Este usuario corresponde al usuario web. De esta forma podrá sincronizar todas las tareas, clientes y categorías.




9.1.6 Sincronización de datos

La sincronización de datos es importante, ya que es la responsable de traspasar la información desde el móvil hacia la base de datos de la web.






En la sincronización tenemos tres estados posibles :




1) Dato no sincronizado, a continuación mostramos la leyenda:

-  Tarea aun no sincronizada
-  Cliente aun no sincronizado
-  Categoría aun no sincronizada

2) Error al enviar los datos, a continuación mostramos la leyenda:

-  Error de sincronizar esta tarea
-  Error de sincronizar esta cliente
-  Error de sincronizar esta categoría

3) Sincronización realizada correctamente, a continuación mostramos la leyenda:


-  Tarea sincronizada correctamente
-  Cliente sincronizada correctamente
-  Categoría sincronizada correctamente

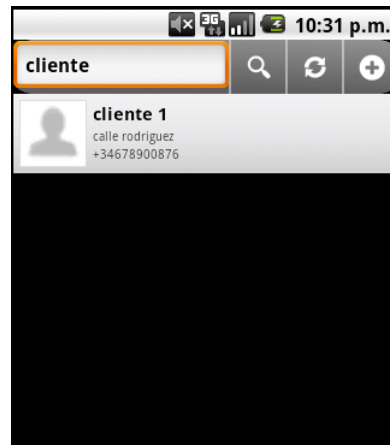
Existe la posibilidad de poder elegir si sincronizar un dato o no. Gracias a la casilla sincronizar que esta en los tres formularios.



Si esta casilla esta activa el sistema , enviara el dato.

9.1.6 Buscar datos

Para poder buscar el dato hace falta darle poner el elemento en la recuadro de texto y pulsar en el icono  .



La búsqueda la va realizar sobre todos los campos de los datos. Es decir para la búsqueda del cliente, por ejemplo , será por el nombre, dirección, numero de teléfono.

10 . Prototipo web

La aplicación web tendrá la responsabilidad de mostrar todas la información que viene des del móvil. La aplicación móvil es la que alimenta los datos de la base de datos del servidor web.

10.1 Pantalla de identificación

En esta pantalla se solicitar el Login (el mail del usuario) y el password de la aplicación web. Pulsando en el botón “Iniciar Sesión” se realizar la verificación del usuario y la contraseña introducida.



En caso de que el usuario sea incorrecto se le mostrar un error de usuario.

Crear Usuario'." data-bbox="151 81 783 232"/>

10.2 Crear Usuario

Para crear usuario en Fast-Work , el usuario tendrá que rellenar el siguiente formulario:

Iniciar session'." data-bbox="189 374 849 606"/>

En este formulario el usuario tendrá que registrar :

- El nombre
- El apellido
- Dirección de correo
- La contraseña
- La confirmación de la contraseña

10.3 Pantalla Principal de Fast-Work

En la pantalla principal de la aplicación podemos ver todas las tareas, clientes, categorías y facturas que hay en el sistema.

En un primer momento tenemos este menú:



A partir de este menú podemos acceder a las listas de categoría, clientes, tareas y facturas.

10.4 Listas de tarea

En esta lista de tareas nos muestra la información sincronizada des de la aplicación del móvil , aquí mostramos la pantalla:

Tarea	Fecha de Inicio	Fecha de fin	Cliente	precio total	realizado por
instalar calefaccion	2013-05-02 08:00:00	2013-05-28 06:00:00	TODAS LOS CLIENTES	3.7	Usuario_por_defecto
transportar alimentos	2013-06-10 08:19:00	2013-06-10 11:19:00	cliente 1	37.5	Usuario_por_defecto

Muestra de 1 a 2 de 2 registros

Primero Antes 1 Siguiete Ultimo

11. Implementación

A continuación vamos a comentar las decisiones tomadas durante la fase de implementación y detallar algunas de las características más importantes.

11.1 Premisas de la implementación

Durante la fase de desarrollo se ha intentado respetar ciertos criterios para preservar tanto el buen funcionamiento del sistema (garantizando agilidad y coherencia en los datos) como la utilización coherente de los recursos del dispositivo móvil (que es la parte con más carencias del sistema). A continuación se comentan los rasgos más significativos:

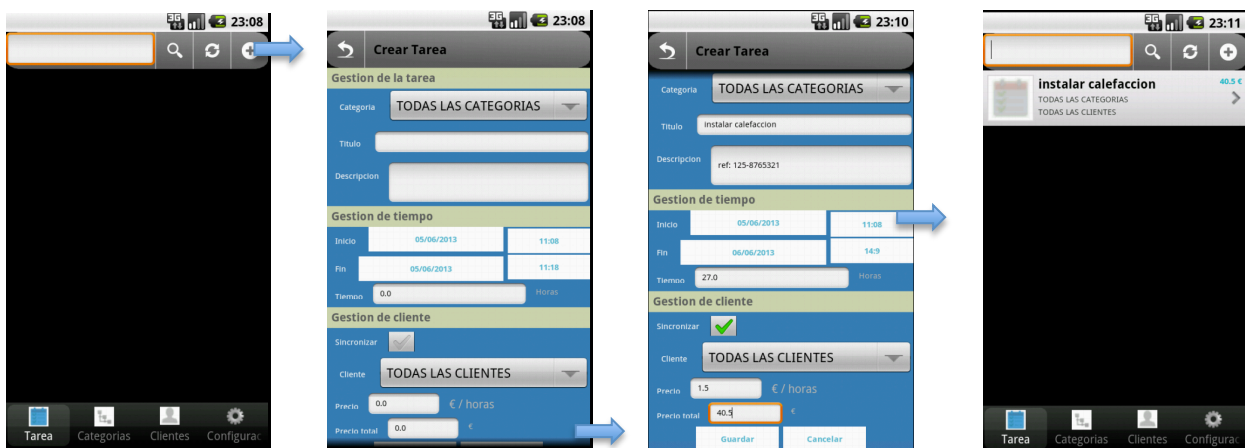
11.1.1 Minimizar el consumo de batería

Se ha tratado de minimizar el acceso a Internet en los dispositivos móviles, para que el consumo de batería sea el mínimo posible. Aunque el sistema se basa en el paradigma Cliente-Servidor en el momento de la sincronización y es inevitable que las operaciones necesiten realizarse en la parte servidora. Existen ciertas comprobaciones que se pueden hacer a nivel local en el propio dispositivo. De esta manera ahorramos comunicaciones que quizás pueden quedarse anuladas por la incorrectas de los datos introducidos y otros motivos.

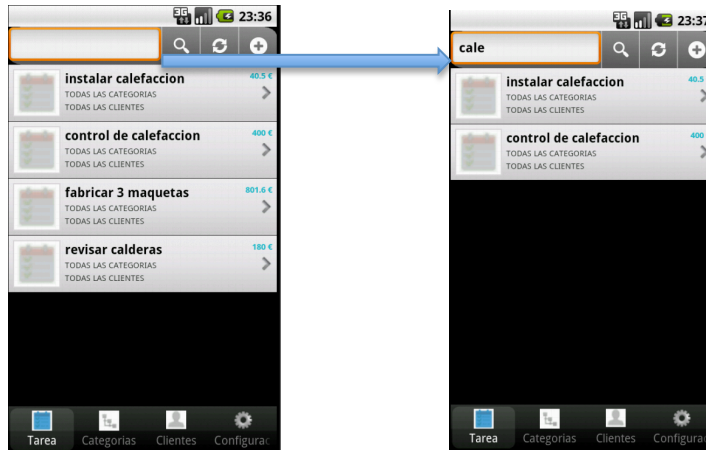
Estas comprobaciones se han realizado tanto verificando que los datos de los objetos de las operaciones son correctas (por ejemplo, se comprueba si el usuario y contraseña introducidas son correctas o no) como creando estructuras en memoria temporales que nos permitan verificar la corrección de los datos (por ejemplo, en el momento de crear una tarea, pasa por una verificación de datos como el título, la fecha de inicio, fecha de fin, etc.).

11.1.2 Facilidades en la navegación

Dado que el usuario debe operar con un dispositivo donde la introducción por teclado no es cómodo, se ha intentado facilitar al máximo la operativa con el teclado. Los grandes bloques de introducción de datos de la aplicación se encuentran en el registro de tarea. Las demás como la categoría y los clientes es un proceso fácil y rápido a crear. Tan solo requiere completar dos campos para poder crear el registro de ambos:



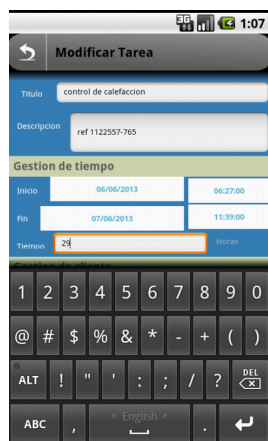
Aparte de este mecanismo se ha hecho un esfuerzo para integrar una funcionalidad que permitiera buscar fácilmente los elementos en las listas de tarea, categoría y clientes. Esta brusquedad busca en todos los campos del registro, sea un campo de fecha, de cadena o de número. El usuario tan solo tendrá que poner el valor a buscar en el campo de búsqueda y pulsar el botón de buscar.



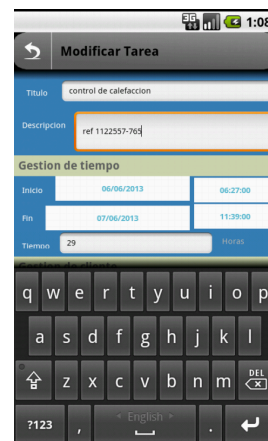
Como se muestra en estas imagines vemos que tan solo poniendo los caracteres “cale” , el sistema buscar todos los caracteres de todos los campos de tarea que estén asociados a “cale”. De esta manera los muestra en la lista de tareas.

11.1.3 Teclado en función del tipo de campo

Durante el desarrollo se ha definido el tipo de teclado emergente adecuado que tenia que aparecer al recibir el campo el foco de edición. De esta manera se facilita la introducción de las datos al usuario. En concreto, los campos numéricos se muestra un tipo de teclado donde sólo aparecen dígitos y símbolos comunes (como los guiones), mientras que los campos alfanuméricos se muestra el teclado habitual.



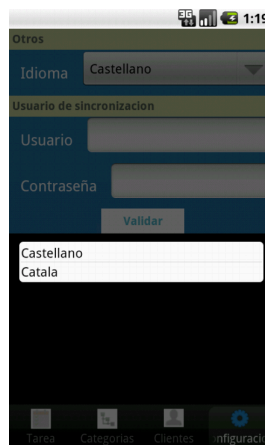
Teclado numerico



Teclado alfanumerico

11.1.4 Aplicación de multi-idioma

Gracias a la propia filosofía de desarrollo del lenguaje Android se ha conseguido que la aplicación móvil esté preparada para trabajar con múltiples idiomas. se han invertido esfuerzos en mantener los textos de la aplicación siguiendo esta filosofía, de modo que en un futuro se pueda traducir (principalmente el inglés, el mercado más amplio que puede acoger). Por ahora esta disponible en catalán y castellano.



11.1.5 Código fuente comentado

El código fuente de todo el desarrollo (tanto de la plataforma web como la plataforma de móvil) ha sido exhaustivamente comentado para favorecer el mantenimiento posterior y la producción de nuevas versiones con mejoras. De esta manera me permitirá realizar el seguimiento de la aplicación y sus funcionalidades con más facilidad en un futuro.

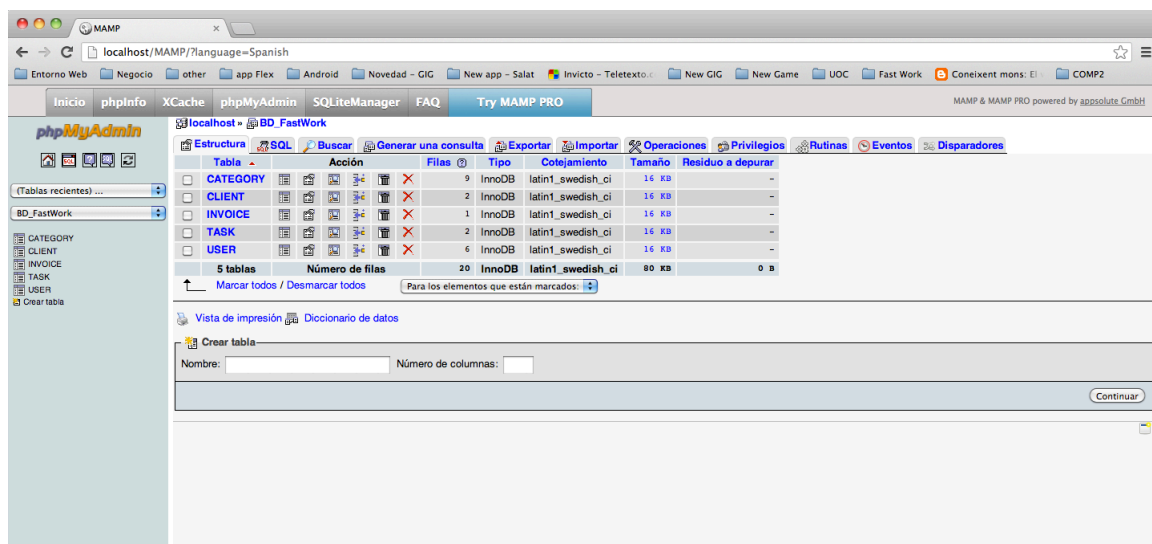
11.2 Implementación de la base de datos

La generación del script de creación de la base de datos se ha realizado con la herramienta de diseño utilizada (Navicat Data Modeler). La importación se ha realizado utilizando el asistente que proporciona PhpMyAdmin, empleando una ordenación `utf8_spanish_ci` para reducir el problema con los caracteres especiales. He utilizado la herramienta de MAMP.

El MAMP sigla se refiere a un conjunto de programas de software libre de uso común en conjunto para ejecutar sitios web dinámicos en los servidores que ejecutan el sistema operativo Macintosh de Apple, Mac OS X:

- **M**ac OS X, el sistema operativo;
- **A**pache, el servidor Web;
- **M**ySQL, el sistema de gestión de base de datos (o el servidor de base de datos);
- **P** para PHP, Perl o Python, todos los lenguajes de programación utilizados para el desarrollo web.

Cualquier plataforma web de código abierto formado por estos programas de software y construido sobre Mac OS X es un MAMP.



11.3 Implementación de la plataforma móvil

A continuación mostraré algunas de las decisiones o mecanismos que se han implementado durante el desarrollo de la plataforma web.

11.3.1 Librerías de diseño GreenDroid

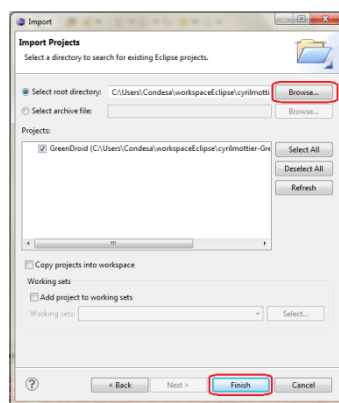


La carpeta Grendroid contienen las librerías del diseño del sistema.

GreenDroid es una librería para Android que nos permite trabajar con widgets muy llamativos para integrarlos en nuestras interfaces de usuario.

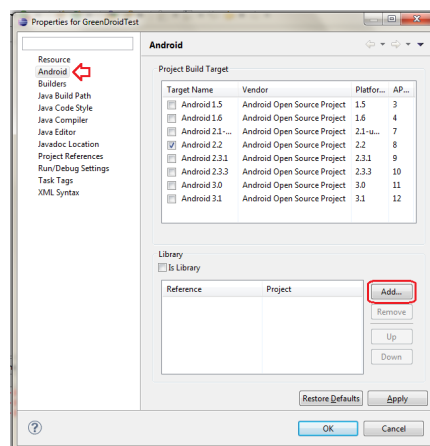
Para poder trabajar en estas librerías de Grendroid hace falta seguir estos pasos :

1) Importar la librería en Eclipse haciendo clic en **File > Import > General > Existing Projects into Workspace**. En la ventana que nos aparecerá, damos clic sobre el botón **Browse...** para buscar la carpeta *cyrilmottier-GreenDroid-2fc5128/GreenDroid*.



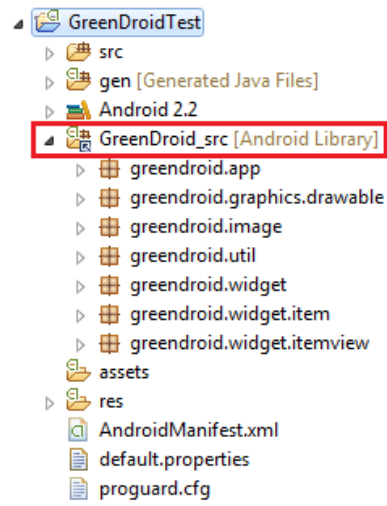
Una vez hecho esto hacemos clic en el botón **Finish**.

2) Ahora creamos el proyecto Android para GreenDroid. En mi caso al proyecto le llamando *GreenDroid* utilizando la versión 2.2. Una vez creado, damos clic derecho sobre el nombre del proyecto y elegimos la opción **Properties**. Dentro de la nueva ventana elegimos *Android* en el menú del lado izquierdo y pulsamos el botón **Add...** de la parte inferior como te muestro a continuación:



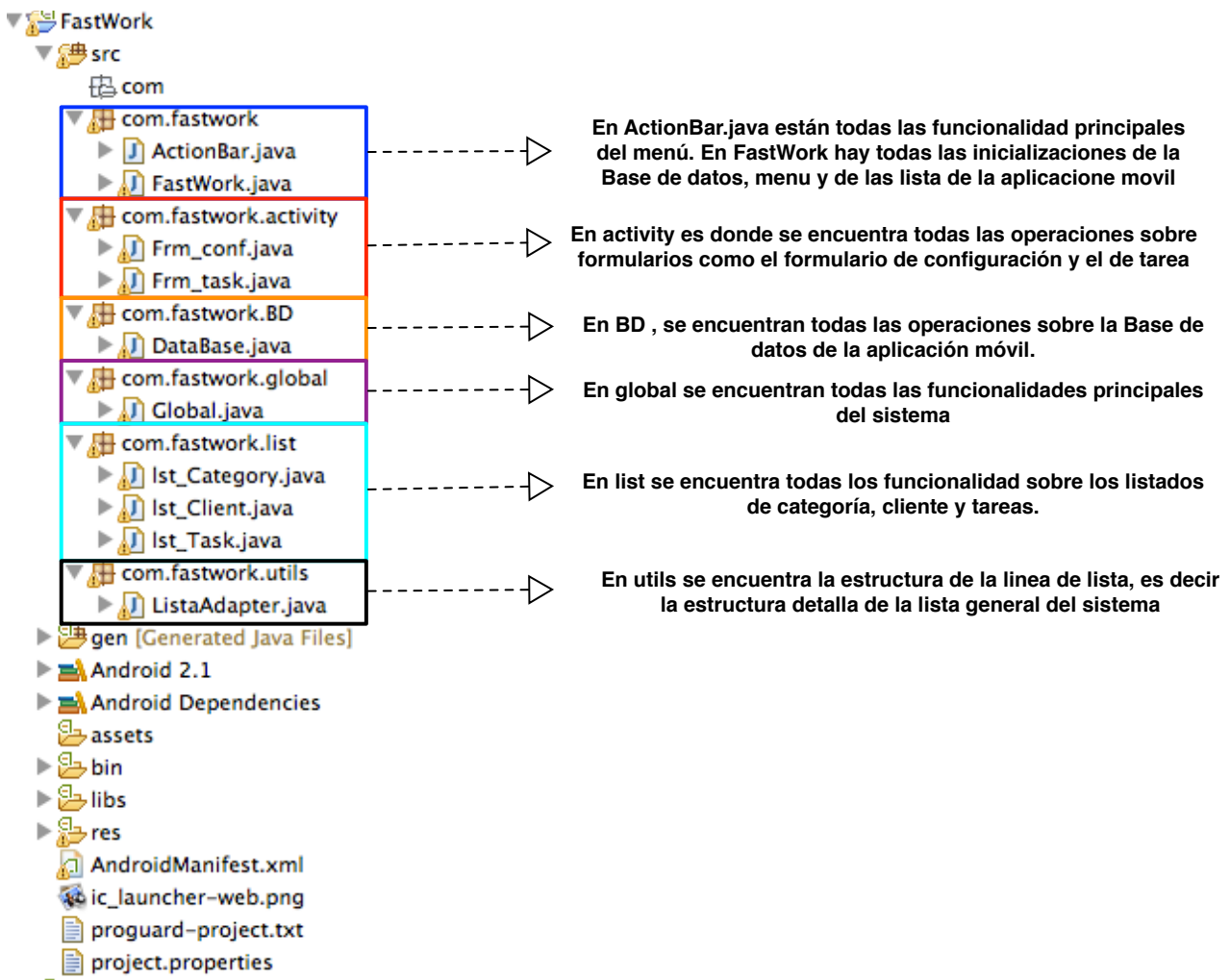
En el nuevo cuadro de diálogo nos debe aparecer la opción **GreenDroid** que elegimos y damos clic en el botón **Ok**.

3) Si hemos hecho todo bien hasta ahora, deberás poder ver el directorio **GreenDroid_src** incluido en tu proyecto:

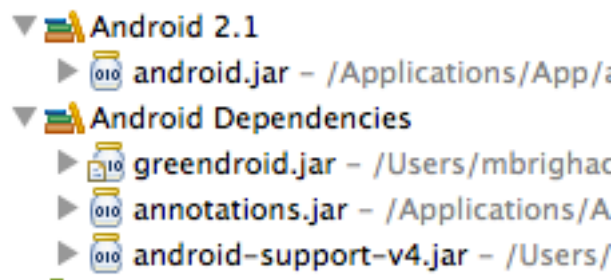


11.3.2 Organización del proyecto móvil

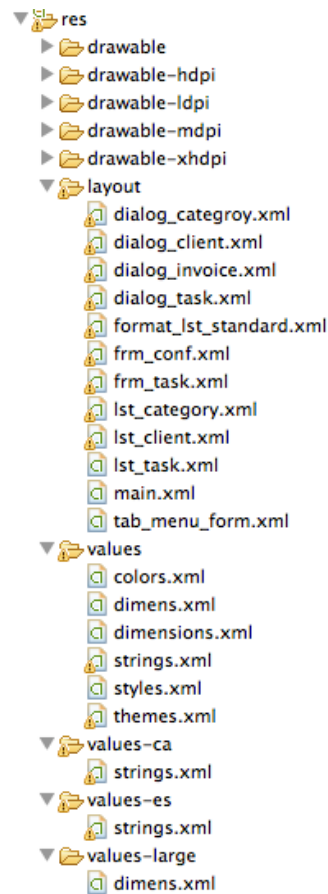
El desarrollo del proyecto implicado en la plataforma móvil está dividido de forma a mantener el paradigma Modelo - Vista - Controlador. De esta manera, podemos encontrar los archivos distribuidos de la siguiente forma:



Las carpetas Google APIs y Android Dependencias contienen las librerías del sistema y aquellas con externas con las que el dispositivo debe operar. La librería esta estructuras de la siguientes manera :

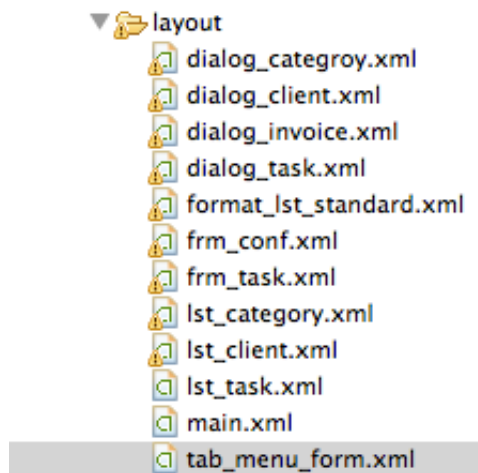


La carpeta **res** contiene los recursos que la aplicación utilizará. Las carpetas **drawable-hdpi**, **drawable-ldpi**, **drawable-mdpi** y **drawable-xdpi** contienen los elementos gráficos de la aplicación en varias resoluciones para adaptarse al dispositivo donde se ejecute. La carpeta **layout** contiene los diseños de pantalla de todas las interfaces que aparecen en la aplicación. Cada una de ellas tiene asociada una clase controladora en la capa de presentación.



La carpeta layout/tab_menu_form.xml contiene la definición de los menús contextuales que aparecen en la aplicación.

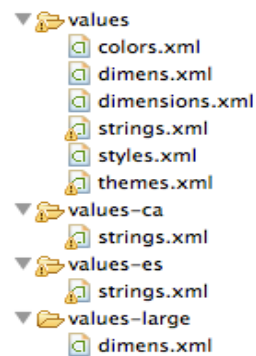
En este caso aparecen tanto en la pantalla principal (donde se permite visualizar la lista de tareas , categoría y de los clientes) como la de configuración (donde se puede elegir el idioma y el usuario).



```
private void setTabs()
{
    addTab(getResources().getString(R.string.menu_tab_task), R.drawable.tab_task, lst_Task.class);
    addTab(getResources().getString(R.string.menu_tab_category), R.drawable.tab_category, lst_Category.class);
    addTab(getResources().getString(R.string.menu_tab_client), R.drawable.tab_client, lst_Client.class);
    addTab(getResources().getString(R.string.menu_tab_conf), R.drawable.tab_conf, Frm_conf.class);
}
```

Como podéis ver la función de setTabs() es la que inicializar y crea los menús en el inicio de la aplicación. Asignar a cada tabulación del menú la clase que corresponde.

La carpeta **values** contiene las definiciones de constantes y estilos visuales de la aplicación. Durante el desarrollo se han definido constantes por los colores, se han configurado estilos de fuentes para ser reaprovechados en toda la aplicación y se han definido los **strings** de literales asociados a los componentes en pantalla. La definición de estas constantes facilita los rápido cambio visual en caso de que sea necesario, y además, permite la traducción de la aplicación con muy pocos cambios. Las carpetas **values-ca** y **values-es** contienen el contenido de los idiomas en catalán y castellano en forma de xml.



El fichero AndroidManifest.xml define los permisos que debe conceder el sistema operativo en la aplicación y de todas las Activities que forman la aplicación. En cuanto a los permisos definidos, ha sido necesario activar los siguientes, para acceder a la conexión a Internet :

```
<uses-sdk
    android:minSdkVersion="7"
    android:targetSdkVersion="7" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

11.3.3 Herencia de la clase *dialog*

Se ha intentado reutilizar el máximo de código posible, aunque en muchas ocasiones (sobre todo en la capa de presentación) parametrizar los métodos era inviable. En este sentido, una de las mejoras ha sido que todas las Activities extienden la clase Dialog, para beneficiarse de uno de los métodos que incorpora y que facilita la aparición de diálogos de alerta.

Los formularios de Categoría y cliente son *dialog*. La razón por haber decido de hacer dialog es para facilitar la navegación del usuario y la rapidez de crear los registros.



11.3.4 Comunicación con la plataforma web

En la comunicación entre el dispositivo móvil y la plataforma web se realiza vía Webservice, donde el dispositivo móvil es el responsable de atacar las peticiones a la plataforma web a través de la clase APIWebservice para hacer la sincronización de datos. Esta clase se encarga de enviar los datos de forma en XML. Aquí tenemos la función de sincronizar los datos de tarea :

```
public String SyncTask(String login,String password, String title, String description,String name_category,String data_ini,String data_fin,String name_client,String price,String price_total,
String url = "http://localhost/webservice/SyncTask.php?";
String respuesta="";

url += "login=" + login;
url += "&password=" + password;
url += "&title=" + title;
url += "&description=" + description;
url += "&name_category=" + name_category;
url += "&data_ini=" + data_ini;
url += "&data_fin=" + data_fin;
url += "&name_client=" + name_client;
url += "&price=" + price;
url += "&price_total=" + price_total;
url += "&data_create=" + data_create;
url += "&user_create=" + user_create;

url= verifURL(url);

try{

    HttpClient httpClient = new DefaultHttpClient();
    HttpContext localcontext = new BasicHttpContext();
    HttpGet httpGet = new HttpGet(url);
    HttpResponse response = httpClient.execute(httpGet,localcontext);
    respuesta = EntityUtils.toString(response.getEntity());

}catch(Exception e){
    respuesta="ko"+e.getMessage();
}

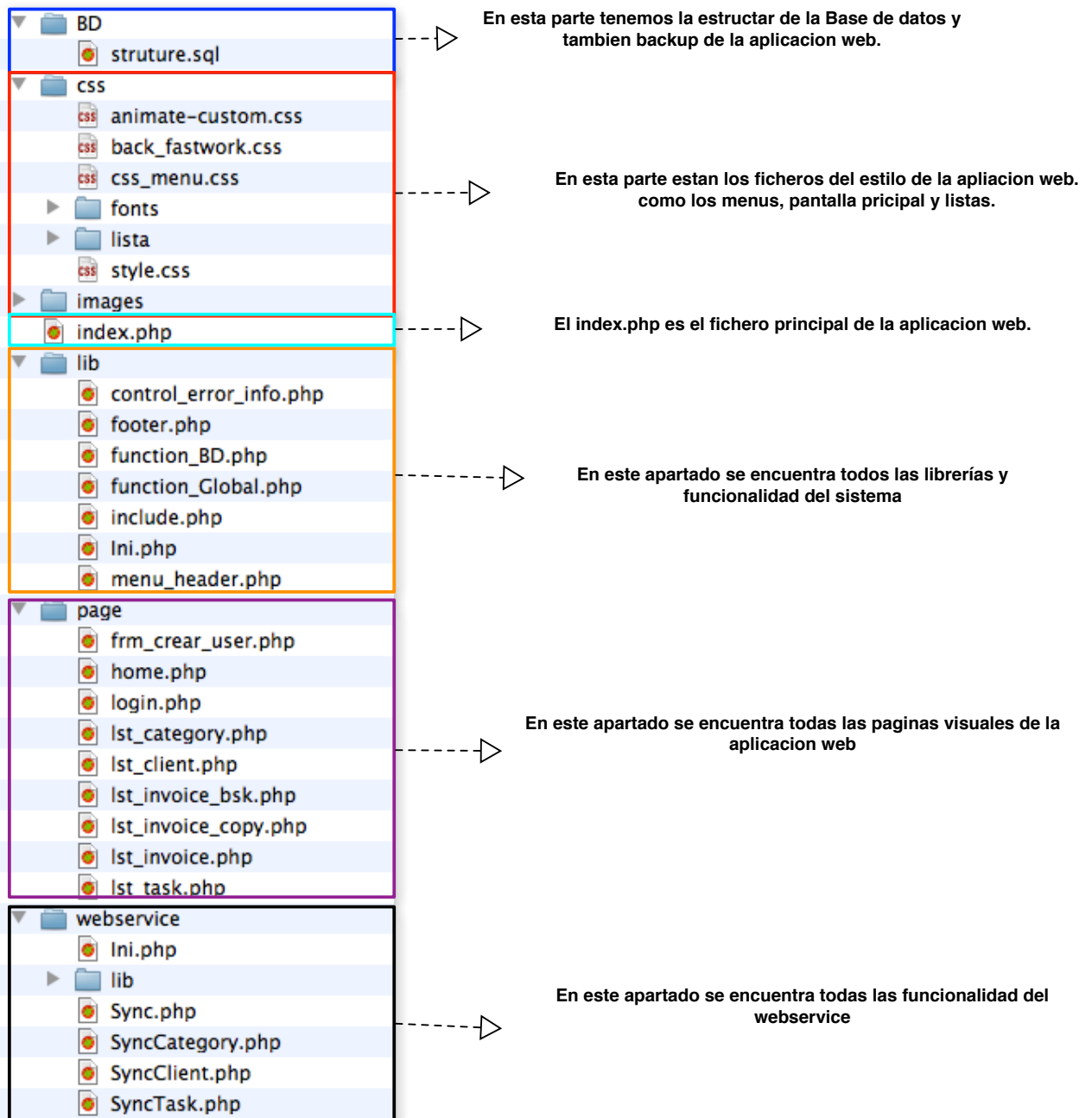
return respuesta;
}
```

Como podéis comprobar es necesario que el login y password sea enviado a cada petición, sin estos requisitos la sincronización no será realizada. Es un mecanismo de seguridad para poder evitar toda intrusión de datos malignos.

La comunicación se realiza a partir del objeto HttpClient, utilizando el protocolo SOAP1.1.

11.4 Implementación de la plataforma web

En esta parte del proyecto de la plataforma web está dividido en el paradigma Modelo-Vista-Controlador. De esta manera, podemos encontrar los archivos distribuidos de la siguiente forma:



Las peticiones se han distribuido en varios archivos php para agilizar la implementación. Cada carpeta tiene un funcionalidad diferente : En primer lugar en la carpeta BD es donde hay ficheros relacionados con la Base de datos como la estructura de la base de datos y las copias de seguridad. En la carpeta css es donde hay todos los ficheros relacionados con el estilo de las paginas como el color, el fondo de pagina y la fuente. En la carpeta lib, es donde hay todas las funcionalidades del sistema de la aplicación, tanto las funcionalidades generales como las individuales. En la carpeta Page es donde hay todas las paginas de la aplicación web. En la carpeta Webservice , se encuentra todas las operaciones relacionadas con el servicio web, como la sincronización de datos entre la aplicación móvil y web.

12. Conclusión

El desarrollo e implementación de este proyecto ha sido una tarea dura y con una inversión de tiempo muy importante. Las previsiones indicadas en el cronograma inicial se cumplido en su totalidad, hasta el punto de que los cambios provocados en algunas de las tareas han permitido añadir funcionalidades no previstas en la aplicación. Dada la experiencia previa con la que contaba en el ámbito de la programación móvil y web el desarrollo de este proyecto ha sido complejo, pero a la vez, ver el resultado final ejecutándose en un móvil ha sido de lo más exitoso.

Así pues, una vez finalizada la exposición del proyecto, paso a comentar las conclusiones estrechas de esta experiencia :

12.1 Objetivos cumplidos

Los objetivos y requerimientos de la aplicación se han completado en su totalidad. He obtenido un producto que suple la aplicación Fast-Work existente para web potenciando todas mejoras que un dispositivo móvil aporta (movilidad, conectividad permanente a Internet, etc.). La aplicación permite dar toda la gestión de sus tareas de su empresa y poder facturar a sus clientes.

Se ha conseguido una aplicación ergonómicamente cómodo, con una curva de aprendizaje reducida y con una reacción al tacto rápida. Las comunicaciones se han optimizado para que el producto realice las mínimas conexiones posibles al servidor y por tanto el tiempo de espera sea el mínimo posible. En pruebas realizadas con algunos usuarios, su reacción ha sido muy positiva.

Por tanto, podemos afirmar que el proyecto cumple con los requerimientos definidos en el inicio del proyecto resultante en un producto de calidad y prácticamente preparado para ser puesto a disposición de los usuarios. Por falta de tiempo, han quedado algunos detalles o funcionalidades adicionales que hubiese querido incluir (no definidas inicialmente) pero que trataré realizar en un futuro cercano.

12.2 Valoraciones personal

La experiencia estrecha de la implementación del proyecto ha sido del todo satisfactoria y enriquecedora. Aparte de la incertidumbre inicial por el desconocimiento de las temáticas y cuestiones técnicas relacionadas (ya que yo provengo de la rama de la Informática que trabaja con programas de gestión empresarial, todas diseñadas para equipos de escritorio y sin interacción con servicios en Internet), poco a poco he ido haciendo pequeñas pruebas en proyectos aislados y obteniendo resultados que me han ido animando y animando a continuar el duro camino que ha supuesto. pese embargo, ha sido difícil cambiar el enfoque tradicional de la programación de aplicaciones en un enfoque donde muchas de las técnicas y recursos empleados en aplicaciones de escritorio no existen o no se pueden aplicar, y ha sido necesario adaptarse a estas circunstancias.

La parte que me ha sido muy difícil ha sido la parte de la aplicación Android , ya que no tenia mucha experiencia en ello y tenia que pasar mucho tiempo a documentarme a partir de muchos tutoriales. Es una de las razones que me ha retrasado este proyecto.

En cambio, la aplicación web ha sido satisfactoria, pero espero mejorar muchas funcionalidades más en futuro cercano.

12.3 Futuras mejoras

Como todo proyecto con una fecha límite, y en un campo donde las posibilidades son tan grandes, siempre hay ciertos detalles o funcionalidades que quedan pendientes por falta de tiempo. En mi caso, ya he conseguido incluir algunas funcionalidades que me ocurrieron mientras realizaba la implementación del proyecto y que se han resuelto positivamente, pero hay otros que no era posible abordar y por tanto, quisiera implementar en una futura versión, antes de ponerla a disposición de los usuarios :

- 1) Posibilidad de generar diferentes formatos de facturas en PDF o Excel : tener varias plantillas para genera facturas en PDF, y también que el cliente decida que formato de factura quiere enviar a sus clientes.
- 2) Gestión de Tareas en Web : Actualmente solo el usuario de la aplicación móvil puede generar las tareas y en la aplicación web tan solo puede visualizarlas y generar las facturas asociada. Una posible mejora en la aplicación móvil seria que tanto el usuario de la aplicación móvil y web puedan crear , eliminar y modificar tareas.
- 3) Gestión de estadística: Una posible mejora seria mostrar gráficos de estadísticas de todas las tareas realizadas, pendientes y facturas. De esta manera el usuario podrá manejar mejor sus tareas, de una formar visual y fácil. También ver el crecimiento de sus negocios a través de estos gráficos.
- 4) Multi-idioma en la Web: Actualmente solo en la aplicación móvil se puede visualizar diferentes idiomas. Una posible mejor en la aplicación seria incluir el multi-idioma , de esta manera los usuarios podrían trabajar en el idioma que mas les convenga.
- 5) Control de alertas : Otra posible mejora en esta aplicación seria un control de alertas de las tareas no facturadas, pendientes o pasadas de fecha a la que han sido previstas. De esta manera el cliente tendrá un control de las tareas y no olvidarse de ninguna.

13. Fuente de información

13.1 Biografía

Libros

Reto Meier (2012). *Android Application Development*. London : wrox

Marko Gargenta (2011). *Learning Android*. Estados-Unidos : O'REILLY

Cristian Darie and Emilian Balanescu (2008). *Php and MySql E-commerce*. Estats Units: Apress

13.2 Consultas Android

Recursos web

<http://developer.android.com/guide/topics/ui/controls/spinner.html>

<http://www.htcmania.com/showthread.php?t=218891>

<http://jonsegador.com/2010/02/tutorial-android-paso-a-paso-desarrollo-aplicacion-notepad/>

<http://www.tutorialandroid.com/>

<http://www.android-spa.com/>

<http://greendroid.cyrilmottier.com/>

<http://androideity.com/2011/09/30/primeros-pasos-con-greendroid/>

<http://tech.chitgoks.com/2011/04/02/hide-greendroid-actionbar-in-android-app/>

<http://www.android.com/>

<http://www.androidcurso.com/index.php/recursos-didacticos/tutoriales-android/32-unidad-2-diseno-de-la-interfaz-de-usuario-vistas-y-layouts/115-una-aplicacion-de-ejemplo-asteroides>

13.3 Consultas PHP

Recursos web

<http://www.w3schools.com/php/>

<http://php.net/>

13.4 Consultas SQL

Recursos web

<http://www.w3schools.com/sql/>

<http://www.sqlcourse.com/index.html>

13.5 Otras Consultas

Recursos web

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

<http://www.mamp.info/en/index.html>

<http://www.oscommerce.com/>

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