



Centro Profesional
Universidad Europea Madrid
LAUREATE INTERNATIONAL UNIVERSITIES

**ESCUELA
POLITÉCNICA**

**CICLO FORMATIVO DE GRADO SUPERIOR
EN
DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

Waiters Coming

Carlos Olmedo Núñez

y

Raúl Cayeiro Marín

CURSO 2013-2014

Resumen

Waiters coming es una pequeña aplicación para la gestión de bares, cafeterías o incluso restaurantes, la idea principal es la de agilizar las ventas a la hora de atender, servir y cobrar los pedidos de los clientes.

Durante todo este documento haremos tres grande bloques importantes para el proyecto, por un lado la aplicación Android que es la interfaz con el usuario, en segundo lugar hablaremos de la parte de servidor que será la encargada de la gestión de los datos y la comunicación con la aplicación cliente y por último tendremos una aplicación encargada de la facturación del bar.

En nuestro proyecto es muy importante que la aplicación Android tenga acceso a la red local de nuestro bar, no por ello tendrá que tener acceso a Internet, aunque será preferible que el propio establecimiento permita a sus clientes acceder a su salida a internet de forma gratuita dando así un servicio extra diferente.

Complementaremos la experiencia del usuario diseñando unos manteles o estampados en las mesas para que el cliente pueda descargar y aprender a usar la aplicación de forma sencilla, en las mesas estarán las instrucciones para descargar la aplicación con un QR y además un pequeño resumen para usar la aplicación.

Palabras Clave

Android, Java, comunicación, servidor, petición, protocolo, conectividad, APK, procesos, hilos, transformación, XML, XSL, xPath, Socket Stream, Layout, permiso, fichero, Wi-Fi.

Abstract

Waiters Coming is a little application to manage the bars, coffee shops or even restaurants, the main idea is to accelerate sales when it comes time to attend, serve and collect customer orders.

Throughout this document we will do three important blocks for the project, on one side the Android application which is the user interface, secondly we will discuss the sever will be in charge of data management and communication with customer application and finally we will have an application in charge of billing bar.

In our project is very important that Android application has access to the local network of our bar, therefore no need to have internet access, although it will preferable that own establishment allow customers to access their internet output, free thus giving an extra different service.

Keywords

Android, Java, communication, server, request, protocol, connectivity, APK, processes, threads, transformation, XML, XSL, xPath, Socket Stream, layout, permission, file, Wi-Fi.

Agradecimientos

En primer lugar nos gustaría dar las gracias a nuestras familias por el apoyo recibido y el ánimo que nos han dado y su ayuda en los momentos difíciles.

A la universidad Europea de Madrid y a todos nuestros profesores durante estos dos años, pues sin ello no habríamos podido alcanzar nuestras metas ni realizar nuestro proyecto.

A nuestro tutor y mentor Pedro, que durante toda nuestra formación se ha preocupado por nosotros.

A nuestros amigos que siempre han estado ahí incluso a la hora de ayudarnos a grabar un video para nuestra presentación.

Y por último a nuestros compañeros tanto de ASIR como de DAM, han sido dos años buenos con muchas experiencias, aunque nos gustaría hacer un recordatorio especial a nuestro compañero Javier González Cívicos, tus compañeros no te olvidan.

Tabla de Contenidos

PARTE I. INTRODUCCIÓN A LA INVESTIGACIÓN.....	10.
CAPÍTULO 1. INTRODUCCIÓN.....	12.
1.1. Planteamiento y justificación del trabajo.....	13.
1.2. Hipótesis y objetivos.....	14.
1.3. Metodología seguida durante la investigación.....	15.
1.4. Organización del trabajo fin de ciclo.....	16.
 PARTE II. ESTADO DEL ARTE.....	 18.
CAPÍTULO 2. APLICACIÓN CLIENTE.....	20.
2.1. Android.....	21.
 CAPÍTULO 3. APLICACIÓN SERVIDOR.....	 24.
3.1. Primeros Pasos.....	25.
3.2. Tratamiento de los mensajes.....	26.
 CAPÍTULO 4. GESTIÓN DE LA FACTURACIÓN.....	 29.
4.1. Facturación.....	30.
 PARTE III. DESARROLLO DE LA INVESTIGACIÓN.....	 32.
CAPÍTULO 5. DISEÑO.....	34.
5.1. Diseño de Protocolos.....	35.
5.2. Fusión.....	37.
5.3. Adaptación del Componente de facturas.....	38.
 CAPÍTULO 6. PRUEBAS.....	 41.
6.1. Aplicación Cliente.....	42.
6.2. Aplicación Servidor.....	44.

PARTE IV. CONCLUSIONES.....	47.
CAPÍTULO 7. CONCLUSIONES.....	49.
7.1. Verificación, contraste y evaluación de los objetivos.....	50.
7.2. Síntesis del modelo propuesto.....	51.
7.3. Aportaciones originales.....	52.
7.4. Trabajos relacionados.....	53.
7.5. Líneas de investigación futuras.....	54.
BIBLIOGRAFÍA.....	57.
ANEXOS.....	60.
A. ANEXO A.....	62.
A.1. Plantilla XML.....	63.
B. ANEXO B.....	65.
B.1. Ejemplo de Facturas.....	66.
C. ANEXO C.....	67.
C.1. Waiters Coming.....	68.
D. ANEXO D.....	69.
D.1. Diagrama de Gantt.....	70.
E. ANEXO E.....	71.
E.1. Diagrama de Gantt.....	72.

PARTE I

Introducción a la Investigación

Capítulo 1

Introducción

En este primer capítulo veremos los pasos necesarios que hemos seguido para hacer una buena implementación de la aplicación desarrollada para el proyecto de fin de ciclo.

En estos pasos se detalla la idea principal que se generó para la aplicación y como llegamos hasta ella. Gracias a dicha idea principal se alcanzan los objetivos que deseábamos cumplir con el desarrollo de esta aplicación. También se expone la razón por la que hemos elegido este proyecto y porqué era la mejor propuesta para realizar el proyecto fin de ciclo a través de ella.

En las siguientes líneas se detalla la investigación previa que ha necesitado la aplicación. También los puntos en los que hemos dividido el trabajo del proyecto, teniendo una alta flexibilidad del mismo. Gracias a esta división, cualquier componente del grupo de trabajo puede, no simplemente saber lo que tiene que hacer, sino también ser capaz de ayudar en otras partes del proyecto que no fuesen las que en un principio se le asignaron.

1.1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL TRABAJO

Elegimos realizar este proyecto porque nos parecía que reunía todas las características propias de un proyecto del CFGS DAM, que han sido adquiridas tanto durante el desarrollo de las clases como durante las FCT. Además, resultaba muy interesante juntar la gran mayoría de módulos de primero y segundo en un único proyecto.

Se planteó una aplicación móvil sencilla con una mayor complejidad en el programa servidor y con lo aprendido durante las FCT realizaríamos el tercer bloque del proyecto.

Aunque en algunos aspectos puede resultar un proyecto complejo, sobre todo por la parte de unir comunicación entre distintas aplicaciones usando Android, nos gustó el reto y nos pareció muy atractivo desarrollar este proyecto.

1.2. HIPÓTESIS Y OBJETIVOS

Nuestra meta para este proyecto era desarrollar un sistema para agilizar y hacer más agradable la experiencia a un cliente que entra en un bar.

Los objetivos principales planteados para el proyecto serán:

- Crear una aplicación atractiva para el cliente.
- Desarrollar un programa servidor que se comuniquen con dicha aplicación.
- Implementar en el programa servidor un apartado para la gestión y el control de la facturación.

Además de estos objetivos principales algunos de los objetivos secundarios serán:

- Conseguir un automatizado para la comunicación del servidor y la aplicación.
- Generar un buen sistema de reserva de mesas.
- Un formato uniforme y simple para nuestras facturas.
- Desarrollar una aplicación sencilla, fluida y totalmente funcional.

Como punto de partida empezamos a desarrollar estos puntos.

1.3. METODOLOGÍA SEGUIDA DURANTE LA INVESTIGACIÓN

Cuando comenzamos a realizar el proyecto la primera investigación que realizamos fue un pequeño estudio de mercado y realizamos un análisis DAFO sencillo que más que darnos una idea de cara al mercado global nos permitía a nosotros mismos hacernos una idea de las cosas que estaba bien en el proyecto y que cosas teníamos que madurar para que dejaran de ser puntos débiles.

En segundo lugar comenzamos analizando los distintos permisos que nuestra aplicación Android iba a necesitar, por un lado y quizás la que más nos llamó la atención fue el acceso a la red local, le dimos prioridad máxima a crear un sencillo ejemplo y ser capaces de implementarlos en la aplicación para poder hacer todo tipo de pruebas y corroborar que era posible implementarlo tal y como nosotros queríamos que estuviese.

Durante toda la investigación de las distintas partes de la aplicación nos dimos cuenta que éramos capaces de empezar a desarrollar, la mayoría de los requisitos de la aplicación eran conceptos con los que estábamos familiarizados o no nos supusieron una gran dificultad.

Aunque en el caso de la realización de facturas en PDF tuvimos que investigar a fondo puesto que no teníamos ningún tema de referencia a algo parecido, por suerte en la misma realización de las FCT adquirimos este conocimiento con Visual Basic y solo tuvimos que adaptarlo a nuestro lenguaje de programación (Java).

El resto de pequeñas dudas o problemas encontrados durante la realización de la investigación los solucionamos con el amplio abanico de información de la web, puede verse esta referencia en el apartado - **REFERENCIAS WEB y BIBLIOGRAFÍA**.

1.4. ORGANIZACIÓN DEL TRABAJO FIN DE CICLO

Como ya hemos dicho, la memoria se dividirá en tres grandes bloques, por un lado la aplicación móvil, el segundo gran punto será el sistema servidor que también será el encargado de almacenar todos los datos que los usuarios hagan y por último el sistema de facturación, que generará e imprimirá las facturas.

Después de definir los tres grandes bloques hablaremos del diseño y la viabilidad del proyecto.

En el último capítulo se expondrán nuestras conclusiones finales y nuestra opinión sobre el proyecto que hemos realizado.

Por último agregaremos unos cuantos anexos que ayudarán a entender el proyecto y a ver su resultado final.

PARTE II

Estado del Arte

Capítulo 2

Aplicación Cliente

En este capítulo veremos todos los requerimientos previos de la aplicación y como se navegará por la misma.

En los primeros capítulos se detalla que versiones se utilizan y porque son las más indicadas dichas versiones para su implementación. En el siguiente se especifican que permisos serán necesarios para algunas funcionalidades implementadas.

En el último punto vemos como hacer un buen uso de la aplicación, donde en este se explicará cómo utilizar la aplicación y la navegación por la misma.

2.1. ANDROID

2.1.1. Versiones Android y permisos requeridos

- Versiones para la aplicación:

Las versiones escogidos son, como API máxima escogimos la 4.3 (Jelly Bean) que aunque no sea la más reciente se escogió dicha API porque la versión 4.4 (Kit-Kat) es todavía demasiado reciente. El API mínimo se escogió la 2.3.3 (Gingerbread) aunque es una versión ya un poco antigua todavía mucha gente posee móviles no tan nuevos, por ello esta versión es buena porque así también podrá utilizarla un público con móviles no tan recientes, consiguiendo así el uso de esta aplicación por un mayor número de público.

- Permisos Requeridos:

Los permisos que necesitará nuestra aplicación serán sencillos porque solo necesitará uno, que será el acceso a Internet. Al ser una aplicación con estructura cliente/servidor, dicho acceso será crucial porque sin él la aplicación no podría funcionar de ninguna de las maneras.

2.1.2. Entorno de desarrollo

El entorno de desarrollo que se utiliza para la implementación de la aplicación es el Android Studio.

La razón por la que se eligió este entorno para el desarrollo, fue principalmente porque cuando se desarrolla gráficamente la aplicación, Android Studio permite ver cómo quedaría directamente en el dispositivo móvil elegido, y no solo cuando nos situamos en la pestaña de gráfico, si no que cuando se accede al XML, también se puede observar el dispositivo con el gráfico, es decir, que el mínimo cambio que se realice en el código del XML, el desarrollador podrá verlo automáticamente en el desarrollo gráfico.

Pero no todo son ventajas, es cierto que la navegación por Eclipse comparada con la de Android Studio es más rápida, es decir, en el entorno de Eclipse uno se puede mover más rápido y trabajar con mayor rapidez, porque en Android Studio consume muchos recursos del ordenador, aún siendo más ligero que Eclipse con el ADT de Android, por lo que ralentiza el trabajo realizado.

2.1.3. Pasos en el funcionamiento de la Aplicación.

Los pasos a seguir para realizar un buen uso de la aplicación son sencillos, porque cualquiera sabría desenvolverse en la navegación de la aplicación, gracias a su sencillez.

En la primera pantalla el usuario introduce el número de mesa en el que se encuentra para que a la hora de hacer el pedido llegue correctamente a su destino.

En este paso se envía un mensaje al servidor con un código que identifica en que mesa se encontrará el cliente.

Una vez indicada la mesa el usuario accederá al menú principal de la aplicación, donde se distinguen las siguientes secciones: Pedido, Carta, Música, La Cuenta.

- **Pedido**

En el apartado pedido el cliente observará una lista de productos que ofrece el establecimiento, en los cuales indicará la cantidad de todo lo que quiera tomar y luego pulsará el botón de Hacer Pedido.

Básicamente se realiza la misma operación que cuando se indicaba la mesa, se envía un mensaje al servidor con un código, las cantidades que pide y el nombre de cada producto que pide, una vez es enviado, esta información se refleja en el ordenador del establecimiento, para que el camarero vea que mesa es y lo que ha pedido.

- **Carta**

En el apartado de la carta simplemente esto hará la función de catálogo, es decir, el usuario observa todos los productos del bar, excluyendo las bebidas, los cuales les darán más detalles de que pueden sumar a su pedido.

- **Música**

En el apartado de música se quiere conseguir un mayor disfrute de los clientes en el bar. Aquí el cliente hará una petición de que canción quiere escuchar.

Otra petición hacia el servidor con un código y el nombre de la canción serán enviadas, cada petición se ordena en una lista en la cual se dará prioridad a poner las canciones, las cuales lleven más tiempo en la lista.

- **La Cuenta**

En el apartado de cuenta, el usuario indica la forma de pago del pedido una vez es indicado, el total de la cuenta se indica en el ordenador del establecimiento y se emprime un ticket de la cuenta, el cuál llevará el camarero a la mesa del cliente.

Capítulo 3

Aplicación Servidor

En este capítulo se explicará todo lo referente al servidor, desde que el primer mensaje llega de la aplicación cliente hasta que se genera el PDF final como factura.

Explicaremos como nuestro servidor tratará los mensajes y plantearemos la necesidad del diseño de una pequeña torre de protocolos que se detallará en el capítulo 5.

También comentaremos como se crea paso a paso nuestro XML que después se convertirá en la factura.

3.1. PRIMEROS PASOS

3.1.1. Planteamiento.

Después de realizar la investigación pertinente sobre las distintas tecnologías con las que podíamos comunicar nuestra aplicación cliente con el servidor nos decantamos por socket stream, ya que nos proporcionaba más fiabilidad a la hora de recibir y mandar los mensajes que otras tecnologías.

Nuestro servidor tendrá que ser capaz de recibir conocer y tratar los distintos mensajes que la aplicación le envíe, a su vez tendrá que ser capaz de recibir y tratar las peticiones de más de una aplicación a la vez.

Recibirá las peticiones de reservar mesa, realizar pedido y efectuar pago como peticiones principales. Con esta última tendrá que iniciar el proceso de facturación automáticamente y generar el pertinente documento para entregárselo al cliente.

3.1.2. Recepción de mensajes.

Para la recepción de mensajes se utilizará una pequeña torre de protocolos que definiremos más adelante, creando una estructura parecida para cada mensaje, el servidor será capaz de identificar el tipo de petición buscando en el primer campo de cada paquete.

3.2. TRATAMIENTO DE LOS MENSAJES

3.2.1. Reservar mesa.

El primer tipo de mensaje que trataremos será el de Reservar una mesa a un cliente, cuando se identifique un mensaje de este tipo el servidor copiará el XML de plantillaWiC.xml y lo renombrará con Mesa<numero de mesa>.XML. Esto no ayudará a tener un control de que mesas están reservadas y cuáles no de forma muy rudimentaria.

Con la copia realizada rellenará los campos:

- Fecha (la fecha actual del sistema)
- Hora (la hora actual del sistema)
- Mesa (número especificado en el mensaje)
- Fact. No. (variable FactNo)

Cuando reserve la mesa incrementará la variable Fact. No.

3.2.2. Realizar pedido.

El segundo tipo de mensaje será el de realizar pedido, cuando se identifique una petición de este tipo el servidor leerá del XML la cantidad actual de un producto y sumará a esa cantidad la que el mensaje contenga y escribirá el valor final en el.

3.2.3. Efectuar pago.

Por último nuestro tercer mensaje recogerá las peticiones de pagar la cuenta, al identificar este tipo nuestra aplicación servidor escribirá en el XML que corresponda el total a pagar.

En segundo lugar mostrará un aviso del tipo:

- Mesa <numero de mesa>: paga <importe total>€, realizará el pago <efectivo/tarjeta>, y <si/no> desea factura.

Seguidamente realizará las transformaciones (si el cliente desea factura realizará los dos tipos de transformaciones y si no la desea solo realizará el de ticket), como nombre de los PDF resultantes de las transformaciones se pondrá:

- Factura: Fm2(fecha-hora).pdf
- Ticket: Tm2(fecha-hora).pdf

Gracias al formateo de los nombres podremos además conocer el periodo en el que se consumieron los ciertos producto y nos podría ayudar a la hora de hacer un estudio o la petición a nuestro proveedor.

Una vez tenemos los documentos PDF, imprimiremos estos documentos.

Lo último que la aplicación servidor llevará acabo será borrar el XML puesto que la mesa en cuestión ya ha pagado, permitiendo así que un nuevo cliente se siente en esa mesa.

Capítulo 4

Gestión de la facturación

Este capítulo contiene una breve descripción de la generación de facturas y el control de clientes que llegan a nuestro bar.

4.1. FACTURACIÓN

Gracias a el control por parte del servidor de las facturas el área de facturación solo se encargará de almacenar los distintos recibos de facturas en carpetas, cada mes crearemos una carpeta para almacenar nuestras facturas y poder hacer un estudio de nuestro mercado.

En el apartado de líneas futuras podremos ver como con una sencilla base de datos podemos tener un sistema de gestión total de nuestro establecimiento uniendo por un lado el servicio cómodo y rápido que ofrece nuestra aplicación con un control del stock y la facturación.

Además podremos llevar a cabo las cuentas anuales o un simple sonde de ganancias mensuales comprobando nuestras carpetas de facturas.

Al final de cada día, cuando apaguemos el servidor y nuestro contador de número de facturas se restablezca a 0, nuestra aplicación guardará en un archivo de texto las transacciones llevadas a cabo durante el día de hoy, una detrás de otra en este formato:

- dd – mm – aaaa : <Fact. No.>

PARTE III

Desarrollo de la Investigación

Capítulo 5

Elementos de la investigación: Diseño

<Introducción del capítulo>

5.1. DISEÑO DE LOS PROTOCOLOS

Una vez escogidas y preparadas cada una de las partes comenzamos a diseñar los distintos protocolos que la aplicación y el programa servidor deberían llevar a cabo para su correcto funcionamiento.

La aplicación debería comunicarse con el servidor por cada acción que el usuario quisiese hacer, esto nos generó la necesidad de crear y definir una torre con distintos paquetes que serían los que definirían los protocolos de comunicación.

5.1.1. Elaboración de la torre de protocolos.

Sin duda una de las partes más importantes en la creación de la aplicación era plasmar la comunicación entre la aplicación y el servidor con una serie de protocolos que nos facilitasen el tratamiento de los datos enviado.

Al ser una de las partes más delicadas y que exigía máxima compenetración por parte de los miembros del grupo la creación de esta torre fue minuciosa. Se siguieron estas fases:

- Se inició el proceso de confección del protocolo basándonos en una idea inicial a la cual se le añadirían después una serie de elementos externos pero nos permitió crear un esquema que sirviese de plantilla.
- Con este esquema-plantilla se organizo una estructura general que compartiesen todos los paquetes y se definió un primer lote de paquete:

Que fueron definidos así:

- Un primer número que define el tipo de mensaje que es, separado por una / que nos ayudará a delimitar los distintos campos, el segundo campo también será genérico para todos los mensajes y contendrá el número de identificación de mesa, el tercer campo del paquete contendrá la información general del mensaje, y por ultimo en el caso que se precise encontraremos un tercer campo con información adicional.

1. Reserva de la mesa.

1 / <numero de mesa 0>

2. Realización de pedido.

2 / <numero de mesa 0> / <código de producto 0000> - <cantidad producto 00>

3. Pago de la factura.

3 / <numero de mesa 0> / <cantidad total a pagar 00000,00> / <forma de pago(E / T)> / <factura (S / N)>

- Estos fueron los tres protocolos principales que definimos, se los denominó internamente protocolos de identificación ya que necesitaban obligatoria mente de un código que asegurase cual es la persona que realiza la petición.

- También se definieron en este punto el paquete de música que se implementaría después.

4. Petición de música.

4 / <id de la canción seleccionada en la lista>

5.1.2. Justificación.

Creemos que crear esta estructura de paquetes es importante y definirlos correctamente también, la idea de Waiters Coming es crear una aplicación fácil de adaptar a las necesidades de cada establecimiento.

Tener un apartado con estos protocolos bien definidos y con una estructura parecida puede facilitar una futura remo-delación de estos.

5.2. FUSIÓN

Una vez definidos los distintos protocolos comenzamos a unir y a generar la funcionalidad real en las dos aplicaciones a la vez.

5.2.1. Método de trabajo

En la primera sesión de la fusión se estableció la creación de un repositorio para el trabajo paralelo de los miembros del equipo, en este repositorio se crearon dos proyectos, por un lado prW1A que hace referencia a la aplicación Android y por otro lado prWserver en el que se encontraba la aplicación servidor.

El repositorio elegido fue subversion ya que es el que se usó durante todas las FCT y teníamos acceso a manuales propios de la empresa.

5.2.2. Reparto de las tareas de fusión

Se estableció un reparto equitativo entre los miembros del grupo, por un lado uno realizó los ajustes y la carga en la aplicación móvil, por otro lado se realizó la recogida y tratamiento de los distintos paquetes.

5.3. ADAPTACIÓN DEL COMPONENTE DE FACTURAS

Una vez solucionados los problemas de comunicación entre cliente y servidor se unió el plug-in de facturas a la aplicación servidor. La prioridad principal de este apartado era que la impresión de facturas fuese automática y dinámica tanto sus dimensiones como su impresión.

5.3.1. Automatización.

En primer lugar se le dio prioridad a la automatización de la creación de factura, se definió así que cuando el servidor recibiese un paquete de tipo 2 / comenzase a crear un XML, dicho XML sería modificada cada vez que otra petición 2 / llegase aumentando a sus registros los que contuviese la petición. En cuanto el servidor reciba un mensaje de tipo 3 / realizará la oportuna transformación e imprimirá lo que el usuario indique, si prefiere factura, un ticket, o ambas.

5.3.1.1. XML.

El XML contendrá una sencilla estructura con:

Figura 1: Estructura XML



- **fecha:** en este campo se instará la fecha de creación final de la factura.
- **hora:** en este campo se instará la hora de creación final de la factura.
- **mesa:** contendrá el número de la mesa que realiza los pedidos.
- **facNo:** en este campo estará el número interno que recibe la factura en el día de hoy.
- **pedido:** en este campo se insertará todos los pedidos agregando:
 - nombre
 - cantidad
 - precio unitario
- **TOTAL:** el servidor calculará el total del último pedido y lo sumará al TOTAL actual del XML para que sea más fácil imprimirlo en la transformación XSL.

5.3.1.2. Transformaciones XSLT.

La aplicación servidor podrá generar dos tipos de transformaciones, por un lado generará tickets que servirán para notificar al cliente de su consumo final desglosado y para que los camareros sepan a qué mesa devén cobrar, por otro lado también generará facturas, estas no se crearan automáticamente después de que el servidor reciba un mensaje de tipo 3 / la factura solo se generará si el mensaje contiene una “S” en la opción de <factura>.

Puede consultarse en el apartado de anexo: Z.Z.1 Facturas y tickets las páginas 3 y 4 para visualizar un ejemplo de cada uno.

Como apunte interesante de esta sección podemos destacar la pequeña función que tuvimos que crear para que el tamaño del ticket aumentase:

```
<fo:simple-page-master master-name="default-page" page-height="{75 +  
count(/fact/pedido/prod) * 6}mm" page-width="80mm" margin-left="0" margin-  
right="0" margin-top="0" margin-bottom="0">
```

En la que tuvimos que indicar por cada línea aumentase 6 mm el alto de la hoja más 75 de base que es aproximadamente lo que ocupaba nuestro encabezado y el tamaño para que se mostrase el “TOTAL”. Decidimos además crearlo de 80 mm de ancho para que tuviese un formato más apropiado de ticket y le quitamos todos los márgenes.

Capítulo 6

Elementos de la investigación: Pruebas

En el siguiente capítulo se muestra una serie de pruebas realizadas tanto para la aplicación, como para el servidor.

Las pruebas realizadas se detallan para cada caso, es decir, que para las diferentes partes del proyecto se realizan una serie de pruebas, para la comprobación tanto de la solidez como para la verificación de las funcionalidades que se esperaban para ambos.

6.1. Aplicación cliente

Las pruebas realizadas en este apartado serán las que realizamos en la misma aplicación, principalmente aquí distinguiremos tres tipos de pruebas:

- Pruebas en la que los campos estén vacíos
- Pruebas en la que se introduzcan datos no válidos.
- Pruebas en la que la longitud requerida de los datos sea mayor.

6.1.1. Campos Vacíos

Este tipo de pruebas se realizarán en los campos en los que la aplicación requiera enviar datos o simplemente extraer un dato para realizar una comprobación.

En todos los campos EditText de las pantallas de la aplicación no insertaremos ningún dato y pulsaremos el botón propio de cada pantalla. En estos casos lo normal es que cada botón tenga asociada una función para tratar los datos indicados.

En estos casos se esperará que la aplicación se detenga, pero sabiendo lo que ocurre y tratando con el tratamiento de errores adecuado, gestionaremos de forma adecuada los errores para que nuestra aplicación no se cierre y podamos continuar trabajando/navegando con ella.

Normalmente este tipo de errores se mostrarán en la aplicación con un mensaje informativo que explicará al usuario/cliente, que no puede dejar los campos vacíos.

6.1.2. Datos no Válidos

Este tipo de pruebas se realizarán en los campos en los que la aplicación requiera enviar datos o simplemente extraer un dato para realizar una comprobación.

En los campos EditText de la aplicación que nos encontremos introduciremos un dato cualquiera que no sea válida, es decir, si en la aplicación se nos requiere el número de mesa y nosotros indicamos en esos campos un 'eeee', entonces la aplicación procederá a cerrarse.

Utilizando de nuevo un buen tratamiento de errores para evitar este tipo de situaciones, en dichos campos procederemos a establecer una serie de condiciones o simplemente eligiendo el tipo de EditText de forma adecuada.

En el diseño se nos permite elegir de que tipo será el campo a simplemente lo indicamos luego en el XML, así condicionaremos a que los datos siempre sean los correctos para cada campo.

6.1.3. Longitud de campos KO

Este tipo de pruebas se realizarán en los campos en los que la aplicación requiera enviar datos o simplemente extraer un dato para realizar una comprobación.

En los campos EditText de la aplicación que nos encontremos introduciremos un dato de una longitud de caracteres que se desee, una vez se envíe este, el tratamiento de datos por parte del servidor fallará debido a que la estructura de todos los mensajes fue definida de manera que la longitud de cada valor tendría que ser constante en cada campo.

Dicho tratamiento de errores se tratará directamente en la aplicación, para que cuando el cliente introduzca datos, dichos datos se condicionarán a que sean de una longitud constante, a través de dicha condición se informará al cliente desde la aplicación de que los datos no son válidos y se requiere un formato de envío específico.

De todas formas cuando se indique un campo de una longitud menor de la requerida dicho campo se rellenará con datos que no aporten información de ningún tipo para que el tratamiento de datos se realice de forma adecuada.

La navegación de la aplicación de esa pantalla se bloqueará hasta que los datos que se desean enviar estén con el formato correcto.

6.2. Pruebas Aplicación servidor

Las pruebas realizadas en este apartado serán las que realizamos para el servidor, principalmente aquí distinguiremos tres tipos de pruebas:

- Pruebas en la que lleguen dos o más mensajes a la misma vez.
- Pruebas de llegada de orden de los mensajes.
- Pruebas de generación de PDF con XML inexistente.

6.2.1. Recepción de mensajes simultáneos

Las pruebas contra el servidor se dispusieron en una nueva clase Java de prueba en ella creamos el apartado para cliente, en el cual enviaremos los mensajes al servidor, en estas pruebas ejecutamos como siempre primeramente la clase servidor y después ejecutamos la clase del cliente varias veces en la que los mensajes se enviaban sin orden alguno.

En este caso el resultado esperado de estas pruebas es que el servidor recibiría los mensajes en cualquier orden. Estas pruebas se diseñan para observar el fallo en el envío de mensajes.

Una vez observados todos los resultados esperados y comprobando que eran idénticos a los resultados imprimidos por el servidor, se dispuso un orden para resolver las incidencias de estas pruebas, las cuales conllevaron a la implementación de un sistema de orden de peticiones de los clientes, es decir, para tratar dichos errores se dispuso que el servicio de atención fuese el de un proceso FIFO.

6.2.2. Orden de mensajes

Debido a la estructura de los mensajes que se enviaran al servidor, estos requieren un orden de envío, por lo que en estas pruebas se observará los resultados que se esperan al enviar un mensaje con una petición de tipo 2 antes que de tipo 1 o de tipo 3 antes que de tipo 2, etc.

Como es normal todos los resultados esperados en el servidor conllevan a la detención del mismo, el cual no espera un mensaje tipo 2, sin antes la recepción del mensaje tipo 1.

El manejo de este orden lo condiciona la aplicación en la cual sin antes realizar los pasos de anteriores, es decir, que sin enviar antes el mensaje tipo 1(reserva de mesa) no se podrá acceder al envío del mensaje tipo 2(realizar pedido), porque a través de la navegación de la aplicación condiciona todo a que todos los mensajes que se envíen al servidor obtengan el orden necesario de las peticiones.

6.2.3. Ausencia de documento XML

Se comprobará que la generación del documento de factura (archivo PDF), será necesario obtener la información del documento XML creado por el servidor con los datos del pedido del cliente.

En las pruebas se ejecutara la parte del servidor que genera automáticamente las facturas en las cuales se esperará que no genere ningún documento PDF aunque el mismo no contenga datos.

Para ello condicionamos que generación de documentos requiera el documento XML necesario para cada pedido, sino a través de la gestión de errores necesaria, cuando se el caso de que el documento XML no exista se comunicará debidamente del error mediante un mensaje, consiguiendo en las pruebas que la detención del servidor ocurra, pero que no se genere ningún tipo de documento de facturación hasta que no se recojan los datos debidamente en el documento XML necesario.

PARTE IV

Conclusiones

Capítulo 7

Conclusiones

En este capítulo principalmente se explicará en detalle el cumplimiento de todos los objetivos que se propusieron en un principio para el proyecto final y como se abordaron a través de la metodología implementada para la aplicación y para el servidor.

También se realizará un análisis de que ventajas y aportaciones se dan para el negocio de los bares a través del uso de la aplicación.

Haremos referencia algunos de los proyectos realizados durante el curso que nos sirvieron de apoyo para la realización del proyecto y también en la resolución de problemas hallados en el mismo.

Como último punto se desarrollara un apartado de futuras líneas de investigación.

7.1. VERIFICACIÓN, CONTRASTE Y EVALUACIÓN DE LOS OBJETIVOS

Nuestra meta para este proyecto era desarrollar un sistema para agilizar y hacer más agradable la experiencia a un cliente que entra en un bar.

Objetivos principales del proyecto:

- Crear una aplicación atractiva para el cliente.
- Desarrollar un programa servidor que se comuniquen con dicha aplicación.
- Implementar en el programa servidor un apartado para la gestión y el control de la facturación.

La aplicación desarrollada se realizó concretamente para que fuese sencilla a la vez que atractiva. Nuestra aplicación posee un sistema de navegación muy sencillo en el que todo el mundo podrá desenvolverse adecuadamente sin hallar mayor problema en ello.

En este caso se hicieron comprobaciones dejando a otros usuarios manejar la aplicación por ellos, sin explicarles la correcta navegación por supuesto, las impresiones del público fueron buenas y no tuvieron problemas en manejarse con ella.

En el programa del servidor se implementó una sencilla estructura de paquetes (que se puede ver en detalle en el capítulo 5) para que cada petición del cliente fuese atendida según la prioridad de los pedidos. Asignando un protocolo a cada uno de los mensajes, se tratan mejor los datos en el servidor y esto da una mejor respuesta por parte del sistema implementado, lo que en la vida real esto desembocará en una mejor atención por parte de los camareros y una mayor satisfacción por parte de los clientes.

El control para la gestión de la facturación de la aplicación se automatizó, es decir, cuando el cliente ya termine y pida la cuenta, el ticket se generará de manera automática, de todas formas antes de realizar el pago se consulta al cliente si desea recibir una factura de todo lo consumido en el bar, una vez se confirme también se imprimirá dicha factura. El sistema de impresión no se realiza de manera automática si no que a través de la confirmación en el mensaje que es enviado al servidor para su tratamiento se especifica una parte son una 'S' o 'N' para que se reciba la factura o no.

7.2. SÍNTESIS DEL MODELO PROPUESTO

Al estar el proyecto dividido en tres grandes bloques, este punto se dividirá en los tres bloques, para hablar de la tecnología necesaria utilizada y con qué motivo para cada bloque.

7.2.1. Aplicación cliente.

Para el envío de mensajes entre cliente/servidor se implemento dicho envío con Socket Stream, estableciendo la aplicación con el papel del cliente en la que simplemente se conectará a la IP del servidor y enviará todos los mensajes tal y como la estructura de paquetes descrita en el apartado 5 de la memoria.

La aplicación de cliente no deberá tener una implementación para la recepción de mensajes originados por el servidor, porque la comunicación que se establece entre los dos es de una sola dirección, debido a que el objetivo de la aplicación es la de enviar pedidos para que los reciban los camareros.

7.2.2. Aplicación servidor.

Como se expone en el punto anterior también se dispone el servidor con la implementación de sockets para la recepción de mensajes del cliente.

En la generación de los XML se utilizaron DOM y SAX, con los cuales generamos la estructura y se insertaran los datos que se necesitarán más tarde, a partir de la lectura gracias a la metodología utilizada obtenemos toda la información introducida en cada nodo del documento XML, y con dicha información se generará la factura de cada cliente del establecimiento.

7.2.3. Sistema de Facturación.

Para el sistema de facturación se implemento una metodología de transformación para el tratamiento de la información insertada en el documento XML generado por el servidor, llamado XSL.

Con esta metodología lograremos tratar toda la información del documento para después generar las facturas de cada cliente con todos los productos consumidos del establecimiento y el total que se deberá pagar. También si el cliente indica previamente que desea recibir un ticket además de la factura correspondiente, la información del XML se utilizará y se realizará la transformación correspondiente para generar los tickets.

Por último todo el tratamiento de información se une a través de la metodología de FOP, implementada también en el servidor, con ella uniremos las transformaciones XSL y los documentos XML y generaremos gracias a ello los documentos de las facturas y los tickets, creándolos en documentos PDF.

7.3. APORTACIONES ORIGINALES

Ventajas que se obtienen al utilizar la aplicación de ‘Waiters Coming’:

Alta disponibilidad en la que no se perderá tiempo en llamar la atención a un camarero del establecimiento gracias a la realización de pedidos desde la aplicación.

Con la estructura de los protocolos todos los datos enviados desde la aplicación serán tratados más fácilmente.

A través de la comunicación entre la aplicación y el servidor los pedidos serán atendidos en orden, que cada pedido tendrá prioridad del que primero que llegue será el primero en ser atendido (FIFO).

Todo el sistema de facturación se automatiza. Aquí el cliente simplemente con la elección de la forma de pago de manera automática a través del servidor se imprimirá el ticket de la factura del cliente.

Sobre todo facilitará el trabajo de los camareros tanto a la hora de llevar el pedido, como a la hora de atender a los clientes.

Todo lo que aporta el proyecto es como una adaptación del negocio de los bares a las nuevas tecnología, facilitando y ahorrando tiempo gracias al uso de una simple aplicación en un establecimiento cualquiera.

7.4. TRABAJOS RELACIONADOS

Para la realización del proyecto se consultaron proyectos realizados durante el curso de DAM, en los que de ellos se consultaron o recogieron partes para el apoyo en el proyecto.

- Actividad Facturas.
- Lenguaje de Marcas.
- Carlos Olmedo y Raul Cayeiro.

- Tarea 2 - Programación de Sockets I.
- Programación de Servicios y Procesos.
- Carlos Olmedo y Raul Cayeiro.

- Aplicación proyecto integrados.
- Programación de dispositivos Móviles y Desarrollo de Interfaces.
- Carlos Olmedo y Raul Cayeiro.

- Actividad 1 Pruebas Unitarias.
- Desarrollo de Interfaces.
- Carlos Olmedo y Raul Cayeiro.

- Actividad 9 y 10 (DOM y SAX).
- Acceso a Datos.
- Carlos Olmedo y Raul Cayeiro.

7.5. LÍNEAS DE INVESTIGACIÓN FUTURAS

7.5.1. Creación del servidor de descarga.

Aunque ya hemos hablado en alguna ocasión de esta posible mejora en el sistema sería un gran complemento tener un servidor de descargas privado en el que con tan solo leer un QR, impreso en una de las mesas, con nuestro Smartphone podamos iniciar la descarga de nuestra aplicación.

7.5.2. Desarrollo del sistema de reserva de mesa con QR.

El sistema de reserva de mesas actual consiste en agregar el número de la mesa en la que el usuario se sienta y enviarlos al servidor, pensamos que será más atractivo y rápido usar unos QR en la esquina opuesta al de descarga indicando correctamente en las instrucciones como poder usar correctamente nuestra aplicación.

7.5.3. Creación de una Base de Datos para almacenar la gestión interna del local.

Creando una pequeña base de datos que actualizase sus tablas con cada pedido podríamos conseguir un sistema de gestión muy amplio, nos permitiría tanto conocer nuestra facturación total como llevar un control de nuestros productos, sabiendo cual es el más consumido e incluso cual es la hora más habitual en la que tenemos más clientes, pudiendo así anticipar, por ejemplo un pedido mayor o contratar un nuevo camarero para atender esta mayor demanda.

7.5.4. Mejora en el sistema de petición de música.

Nuestro sistema de petición de canciones actualmente no funciona correctamente ya que creemos que no es buena idea que, por ejemplo un cliente pueda agregar más de una canción consecutiva o que varios clientes pidan la misma canción y suene en el local una y otra vez. Sería recomendable crear un sistema que nos permitiese gestionar este tipo de peticiones.

7.5.5. Maquetas de mesas explicativas para los clientes.

Aunque también hemos hablado de ellas la maqueta de las mesas debe ser muy atractiva e interesante, debe llamar la atención, cuando una persona que no conozca la aplicación pase alado de una terraza en la que esté implementado nuestro sistema debe llamarle la atención ya sea por sus colores o algo en el diseño que le anime a interesarse por esta nueva tecnología.

7.5.6. Creación de un sistema de alerta que avise al cliente de posibles ofertas y descuentos.

Crear un sistema de alertas nos podría proporcionar un mayor beneficio, puesto que en cuanto la aplicación se conectase a Internet podrían llegar nuevas ofertas o promociones, siendo este un sistema de publicidad totalmente gratuito y muy directo.

En cualquier momento el usuario podría desactivar este sistema de alertas, ya que tampoco queremos perder clientes a cambio de fidelizar a unos pocos.

7.5.7. Actualización de la aplicación cliente para que recoja los datos del XML almacenado en el servidor en lugar de ser estático.

Actualmente la aplicación es estática y algo rígida, para agregar un nuevo producto hay que retocar el código y regenerar la plantilla XML, podría ser muy interesante que la aplicación recogiese los datos de un único XML y mostrase su contenido formateado.

Así solo tendríamos que regenerar el XML cada vez que quisiésemos agregar una oferta o producto.

7.5.8. Creación de un interfaz para el propio dueño del local en la que pueda añadir, modificar o eliminar productos, tanto en la aplicación cliente como en el XML que hace de plantilla sin necesidad de conocer el código.

Esto sería interesante a la hora de vender el producto, puesto que darle independencia al personal del local seguramente haga más atractiva la compra de la aplicación por su parte.

7.5.9. Desarrollo de una interfaz para el servidor fácil de usar.

En este momento el servidor muestra la información por comando y no es nada amigable con la persona que necesite trabajar con esta información. Sería altamente recomendable diseñar y crear una interfaz fácil de usar y descriptiva.

7.5.10. Agregar a la aplicación cliente un sistema de mapas en el que se puedan ver todos los locales que disponen del servicio “Waiters Coming”.

Además del sistema de alerta de oferta se podría agregar un apartado que mostrase en un área determinada todos los locales con nuestro sistema.

7.5.11. Calendario personalizado que nos avisa de posibles eventos.

Podríamos crear un apartado de calendario en el que podamos anotar eventos importantes, como partidos de fútbol o eventos deportivos en los que haya un gran número de clientes.

Unos días antes nuestro sistema podría realizar una consulta con la base de datos que propusimos crear en el punto 2 de este apartado y recibir una estimación de cuanto deberíamos pedir a nuestro proveedor.

BIBLIOGRAFÍA

Referencias web de interés utilizadas en el desarrollo del proyecto.

- Autor/es: Alejandro Alcalde.
- Título: Programa que envía mensajes desde Android a PC.
- Dirección web: http://elbauldelprogramador.com/programa-que-envia-mensajes-desde_10/
- Autor/es: Alex Guerra.
- Título: Java y XML: JDOM.
- Dirección web: <http://www.latascadexela.es/2008/08/java-y-xml-jdom.html>
- Autor/es: Ap0C552.
- Título: Android Client connecting to Java server.
- Dirección web: <http://www.dreamincode.net/forums/topic/338653-android-client-connecting-to-java-server/>
- Autor/es: Bartolomé Sintés Marco.
- Título: XSLT: Transformaciones XSL.
- Dirección web: http://www.mclibre.org/consultar/xml/lecciones/xml_xslt.html
- Autor/es: Eduard Puigdemunt.
- Título: Convertir XML en PDF utilizando XSL-FO y FOP.
- Dirección web: http://www.programacion.com/articulo/convertir_xml_en_pdf_utilizando_xsl-fo_y_fop_73
- Autor/es: Jason Hunter, Brett McLaughlin.
- Título: JDOM v2.0.5 API Specification.
- Dirección web: <http://www.jdom.org/docs/apidocs/>

- Autor/es: Oracle.
- Título: Class Socket.
- Dirección web: <http://docs.oracle.com/javase/7/docs/api/java/net/Socket.html>

- Autor/es: Ricardo Borillo Domenech.
- Título: Primeros pasos con XML y XSL.
- Dirección web: <http://xml-utils.com/conferencia-xsl.html>

- Autor/es: Ricardo Moya y Ramón Invarato.
- Título: ListView o listado en Android.
- Dirección web: <http://jarroba.com/listview-o-listado-en-android/>

- Autor/es: Tunelko. A technology point of blog.
- Título: Android SDK: ¿Cómo detecto la conexión a Internet?
- Dirección web: <http://blogs.tunelko.com/2011/12/26/android-sdk-como-detecto-la-conexion-a-internet/>

Otras referencias utilizadas para el proyecto final.

- Autor/es: Pedro Jesus Camacho
- Título: Programación de Sockets
- Módulo: Programación de Servicios y Procesos
- Curso: 2º DAM
- Universidad: Universidad Europea de Madrid

- Autor/es: David Perez
- Título: Documentación XML y XSL
- Módulo: Lenguaje de Marcas
- Curso: 1º DAM
- Universidad: Universidad Europea de Madrid

PARTE V

Anexos

Anexo A

Plantilla XML

En este anexo se encuentra la plantilla usada para crear los XML de los clientes totalmente bacía.

A.1. PLANTILLA

```

<?xml version='1.0' encoding='ISO-8859-1'?>
<fact>
  <fecha></fecha>
  <hora></hora>
  <mesa></mesa>
  <factNo></factNo>
  <pedido>
    <prod id="3241">
      <cantidad>0</cantidad>
      <nombre>Coca-cola</nombre>
      <precio>2</precio>
    </prod>
    <prod id="5673">
      <cantidad>0</cantidad>
      <nombre>Fanta Limon</nombre>
      <precio>2</precio>
    </prod>
    <prod id="1231">
      <cantidad>0</cantidad>
      <nombre>Cerveza</nombre>
      <precio>2</precio>
    </prod>
    <prod id="7812">
      <cantidad>0</cantidad>
      <nombre>Cafe</nombre>
      <precio>1.2</precio>
    </prod>
    <prod id="9818">
      <cantidad>0</cantidad>
      <nombre>Bocadillo Tortilla</nombre>
      <precio>3</precio>
    </prod>
    <prod id="1611">
      <cantidad>0</cantidad>

      <nombre>Bocadillo Lomo</nombre>
      <precio>4</precio>
    </prod>
  </pedido>
  <TOTAL></TOTAL>
</fact>

```

Anexo B


Ejemplos de Facturas

En este anexo se pueden ver las imágenes de los dos tipos de factura que el programa genera, tanto los tickets como las facturas.

B.1. FACTURACIÓN

C/ falsa nº 123 Illescas (Toledo) Tlf. 925-51-00-00		Fecha 21-05-2014 - 15:05	
		Nº de Factura 002	
Mesa:1			
Concepto	Cantidad	Precio Unitario	Precio Total
Patatas estofadas	X 4	3.5€	14€
Botella de agua	X 4	1€	4€
Coca - Cola	X 1	2€	2€
Carne	X 3	3.5€	10.5€
Tortilla	X 1	3.5€	3.5€
Flan	X 2	1€	2€
Tarta	X 1	1€	1€
Pan	X 4	1€	4€
			TOTAL:38€

B.2. TICKET

Mesa Nº: 1			
21-05-2014 - 15:05			
Nº 002			
C/ falsa nº 123			
Illescas (Toledo)			
Tlf: 925-51-00-00			
Patatas	X 4	3.5€	14€
estofadas			
Botella de	X 4	1€	4€
agua			
Coca - Cola	X 1	2€	2€
Carne	X 3	3.5€	10.5€
Tortilla	X 1	3.5€	3.5€
Flan	X 2	1€	2€
Tarta	X 1	1€	1€
Pan	X 4	1€	4€
.....			TOTAL: 38€
 XML to PDF by Rendition			

Anexo C

Navegación de la Aplicación

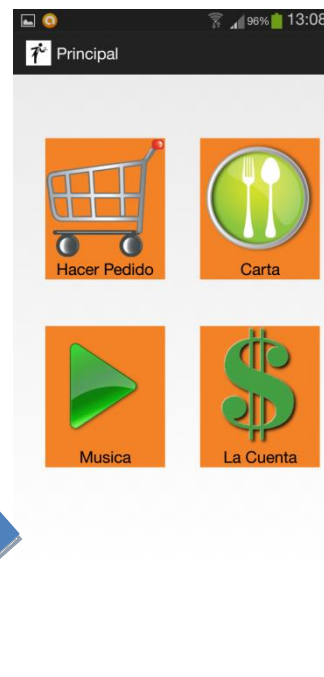
En este anexo se pueden ver las imágenes de la navegación realizada por la aplicación de Waiters Coming.

C.1 WAITERS COMING

Inicio Aplicación



Menú Principal



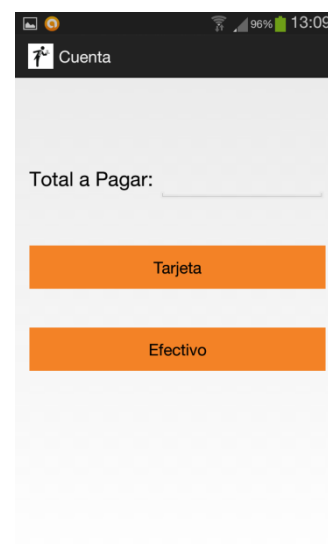
Hacer Pedidos



Carta



La Cuenta



Anexo D

Diagrama de Gantt

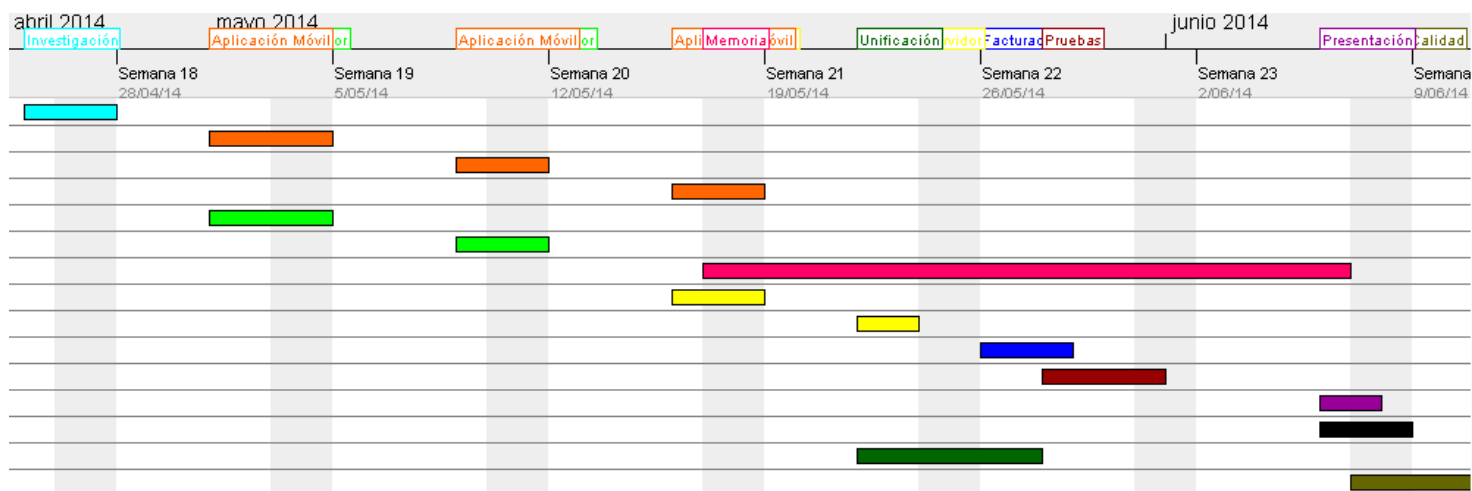
En este anexo se pueden ver las imágenes de la planificación del proyecto realizada en un diagrama de Gantt.

D.1. DIAGRAMA DE GANTT

Estimación de horas.

Nombre	Fecha d...	Fecha d...	Tiempo
Investigación	25/04/14	27/04/14	8 Horas
Aplicación Móvil	1/05/14	4/05/14	3 Horas
Aplicación Móvil	9/05/14	11/05/14	3 Horas
Aplicación Móvil	16/05/14	18/05/14	2 Horas
Protocolos Servidor	1/05/14	4/05/14	3 Horas
Protocolos Servidor	9/05/14	11/05/14	2 Horas
Memoria	17/05/14	6/06/14	10 Horas
Mensajes servidor	16/05/14	18/05/14	2 Horas
Mensajes servidor	22/05/14	23/05/14	2 Horas
Facturación	26/05/14	28/05/14	4 Horas
Pruebas	28/05/14	31/05/14	5 Horas
Presentación	6/06/14	7/06/14	3 Horas
Video Presentación	6/06/14	8/06/14	3 Horas
Unificación	22/05/14	27/05/14	6 Horas
Análisis Calidad	7/06/14	10/06/14	4 Horas

Diagrama



Anexo E

Diseño de las Mesas

En este anexo presentaremos el diseño que tendrán las mesas del establecimiento que use nuestra aplicación.

Diseño principal de la mesa.



Diseño en establecimiento.

