

INSTITUTO DE EDUCACIÓN SECUNDARIA SERPIS

CICLO SUPERIOR DE DESARROLLO DE APLICACIONES MULTIPLATAFORMA



"Creación de una aplicación programada en Java, para móviles basados en el sistema operativo Android, para facilitar y agilizar la compra a clientes de supermercado"

Autor: **Ignacio Vives Menor**

Director: **M^a Ángeles Lorenzo**

Fecha de entrega: **29 de Mayo de 2013**

Índice:

Prólogo

1. Introducción

1.1. Teléfonos móviles

1.2. Sistemas operativos

1.2.1. Android

1.2.2. iOS

1.2.3. BlackBerry OS

1.3. Android

1.4. Google Play Store

2. Tecnologías utilizadas

2.1. Android

2.2. JSON

2.3. XAMPP

2.3.1. MySQL

2.3.2. Apache

2.3.3. PHP

2.4. SQLite

3. La aplicación Mi Carrito de la compra

3.1. Esquemas de la aplicación

3.1.1. MainActivity

3.1.2. Carrito

3.1.3. MostrarProducto

3.1.4. Lector

3.1.5. Ayuda y Acerca de

3.2. Diseño cliente Android

3.2.1. Implementación cliente Android

3.2.1.1. Activities y XMLs

3.2.1.2. El menú de la aplicación

3.2.1.3. Almacenamiento de datos

3.2.1.4. JSON y HTTP

3.3. El servidor

4. Resultados y conclusiones

5. Trabajos futuros

A. ANEXO: Manual de usuario

Servidor

Aplicación móvil

B. ANEXO: Base de datos Supermercado

C. ANEXO: Códigos de barras

Bibliografía

Prólogo:

La rutina es parte de nuestra vida diaria, son acciones que realizamos cada día por igual, tal vez con alguna pequeña variación para llegar siempre al mismo objetivo.

Un ejemplo de los muchos que nos podemos encontrar a diario sería el de realizar la compra, llenamos nuestros carritos de la compra con una gran cantidad de artículos y luego cuando llegamos a la caja casi siempre nos llevamos una sorpresa, si queremos saber cuánto llevamos gastado debemos ir apuntando lo que compramos e ir realizando la suma de lo que nos costará. La vida en si está basada en rutinas, eso no podemos evitarlo, pero lo que sí podemos hacer facilitar y en muchos casos disminuir la cantidad de acciones a realizar para conseguir el objetivo.

Por lo tanto lo que aquí les presento no es más que una idea para hacer que su rutina diaria, para una tarea en particular, sea mucho más sencilla.

Para facilitar las cosas he decidido desarrollar una aplicación para móviles con SO Android, que se base en la simplicidad, con el fin tener nuestra lista de la compra en nuestros dispositivos a medida que vamos introduciendo los productos en el carro o cesta.

1. Introducción:

Es un hecho que la tecnología es una parte importante de nuestras vidas. Desde que empieza el día y hasta que este termina, nos habremos cruzado y seguramente utilizado diversos artefactos tecnológicos y aplicaciones que nos facilitan la rutina diaria. Sin embargo existe un pequeño artefacto electrónico que nos acompaña a diario y que cada día nos ofrece mas y mas soluciones, el teléfono móvil. Actualmente los podemos encontrar de muchos tamaños y formas y a su vez con una gran variedad de sistemas operativos. Aparte de ofrecernos su función básica, la de llamar, hoy en día podemos instalarle un sin fin de aplicaciones de casi cualquier tipo, desde aplicaciones de ocio hasta aplicaciones de gran utilidad que nos ayudan a resolver o por lo menos organizar gran parte de nuestros días.

1.1. Teléfonos Móviles

Los teléfonos móviles son dispositivos inalámbricos que utilizan y acceden a servicios de red de telefonía celular o móvil. Como bien se menciona anteriormente, actualmente han adquirido funcionalidades que van mucho más allá de su objetivo inicial, el de realizar llamadas telefónicas. Así que hoy en día un teléfono móvil es más que un dispositivo, es un conjunto de dispositivos fusionados en uno solo, por ejemplo: cumple la función de cámara, reloj despertador, dispositivo GPS, calculadora, reproductor multimedia, etc. Este auge ha hecho que las compañías dedicadas a la telefonía móvil, potencien mas sus aparatos electrónicos ofreciéndole un software con el cual podamos descargar e instalar o desarrollar nosotros mismos, aplicaciones con el fin de satisfacer una necesidad. Si seguimos la evolución de los dispositivos móviles a lo largo de la historia podríamos darnos cuenta que con el paso del tiempo se tiende a las cosas más compactas y pequeñas con el fin de facilitar su portabilidad, sin embargo parece que hemos llegado a un momento donde los usuarios establecen por defecto un tamaño estándar entre 3,5 y 5 pulgadas, esto obviamente se ve influenciado por las tendencias de los dispositivos móviles, ya que cada vez mas necesitamos pantallas estándares donde podamos visualizar de manera agradable la información a la hora de leer

correos, navegar por internet, leer documentos, tomar fotografías, visualizarlas y/o modificarlas y por ultimo aunque no menos importantes cuando usamos las aplicaciones de nuestro móvil.

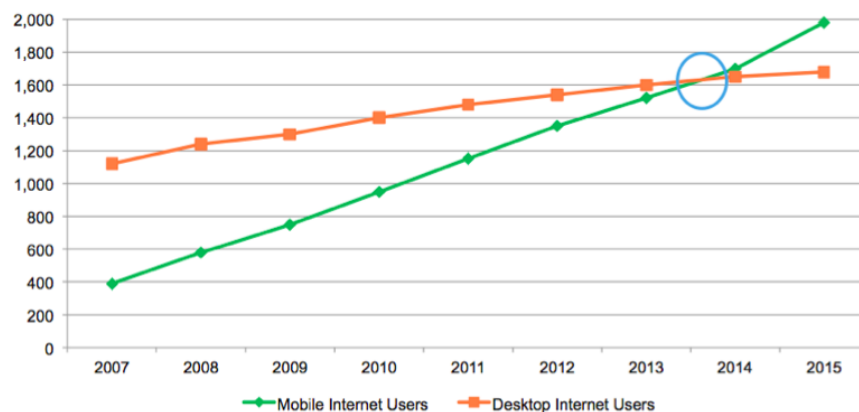


Evolución de los dispositivos móviles

Otro de los factores que hace que los dispositivos móviles sigan en aumento es el uso masivo que se le está dando al uso de las conexiones inalámbricas, en particular a internet. Como podemos comprobar, según las proyecciones para el presente año, existirán el mismo números de usuarios conectados a través de dispositivos móviles que de ordenadores de casa, y con el tiempo la tendencia seguirá aumentando de mayor manera que los ordenadores.

Mobile Web Usage Growing

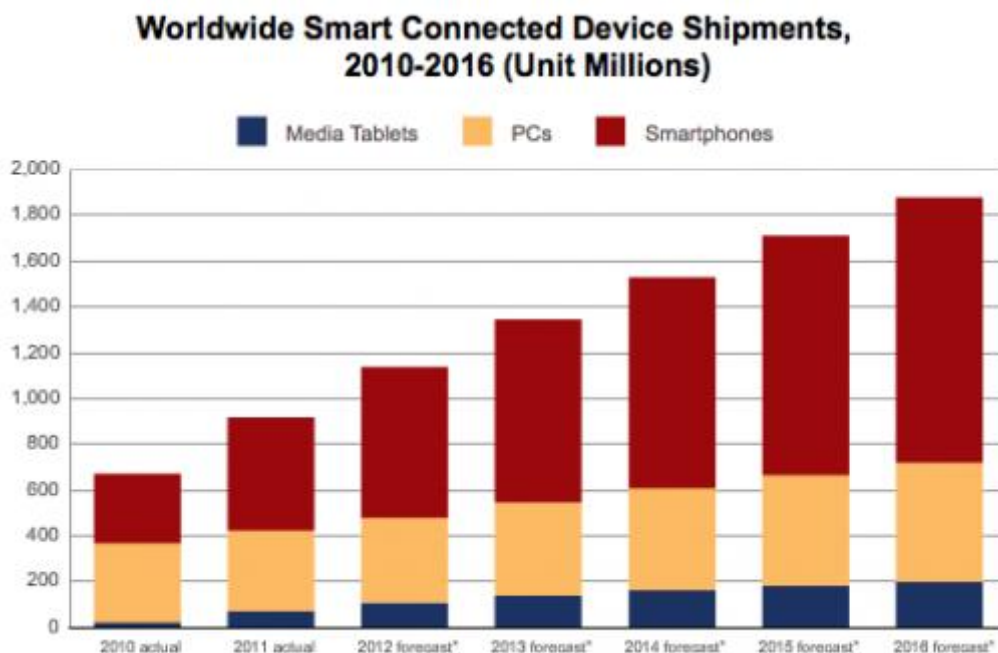
Forward Projection: Mobile Web Browsing vs. Desktop Web Browsing
(2007-2015)



Source: Mary Meeker, Morgan Stanley, "Internet Trends," April 12, 2010

Móvil vs Escritorio

Por otro lado podemos comprobar de manera similar la evolución de los dispositivos móviles con respecto al resto de aparatos electrónicos en cuanto a su uso, y nuevamente es sorprendente como cada vez es mayor la tendencia del uso de dispositivos personales y compactos, hasta tal punto que parece que pudiera comerse casi por completo la cuota de mercado que tenían sus antiguos predecesores; lo cual era de esperarse ya que hoy existen móviles que son más poderosos que muchos de nuestros ordenadores y portátiles que tenemos en casa.



1.2. Sistemas Operativos

Los sistemas operativos son un conjunto de programas que se encargan de gestionar en un sistema informático los recursos de hardware de los diversos dispositivos, y a su vez provee los servicios necesarios a los programas de aplicación. Sus funciones básicas consisten en la gestión de los recursos del dispositivo y la protección de los accesos al hardware.

Todos los dispositivos móviles utilizan un SO (Sistema Operativo), encargado de controlar el funcionamiento del aparato electrónico de la misma manera que los SO controlan nuestros PC's. Cabe acotar que su dichos sistemas no son tan complejo como el de un ordenador (aunque cada día más tiende a ello) sino que están más orientados a las conexiones inalámbricas.

Actualmente existe una gran variedad de SO, sin embargo podríamos hablar de 3 que son los que se llevan la cuota actual de mercado.



1.2.1. Android

Aparece en el año 2005 desarrollado por Android Inc., dicha firma fue comprada por Google en el mismo año. “Android es una plataforma móvil, que incluye un sistema operativo móvil basado en Linux, junto con un conjunto de aplicaciones middleware⁸ y aplicaciones nativas. Está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos”.¹

Desde sus inicios, es un SO que ha venido en gran ascenso, hasta tal punto de ocupar hoy en día una cuota de mercado mundial del 75% desplazando a al sistema iOS de Apple. Una de las virtudes de Android es que su código es libre y gratuito, lo que permite a cualquier persona tener acceso al kit de desarrollo para crear casi cualquier tipo de aplicaciones para esta plataforma. Dichas aplicaciones son distribuidas a través de una plataforma de google llamada “Google Play”, donde podemos encontrar más de 650 mil aplicaciones distintas tanto de pago como gratuitas.

¹Fuente: <http://es.wikipedia.org/wiki/Android>

1.2.2. iOS

El iOS es el sistema operativo de Apple desarrollado para sus teléfonos móviles el iPhone en el 2007. Este deriva de del Mac OS X (SO de los ordenadores) que está basado en Darwin BSD (Unix). Era uno de los SO más vendidos para el año 2009 y comienzos del 2010 hasta que le fue arrebatada por el SO Android.

Aunque su sistema fue pensado inicialmente para el iPhone, con el paso del tiempo se empezó a utilizar en el resto de sus dispositivos: iPod Touch, iPad y Apple TV. Actualmente tiene una cuota de mercado mundial de casi el 15% lo cual lo coloca en el segundo SO más vendido.

Inaugurada en julio del 2008 Apple posee uno de sus grandes castillos que incluso a Android le cuesta penetrar, la App Store. Con más de 800.000 aplicaciones y más de 25 mil millones de descargas, Apple demuestra que tiene el mayor de los pilares en aplicaciones para móviles.

A pesar de sus grandes logros, el hecho de que sus móviles no estén al alcance de cualquiera y que la libertad que le ofrece al usuario es escasa, es lo que los hace vulnerables a los mercados emergentes.

1.2.3. BlackBerry OS

Es el SO desarrollado por Research In Motion para los dispositivos BlackBerry. Este sistema operativo y sus móviles, estaban y están orientados a un uso profesional del gestor de correo y agendas. Su código esta basado en una combinación de Java y C++ y su desarrollo se remonta desde el año 1999 con la aparición de las primeras PDA (Personal Digital Assistant). Si bien su uso estaba dirigido hacia profesionales, RIM introduce uno de los primeros sistemas de comunicación de mensajería instantánea entre sus dispositivos 16 que se comunicaban entre ellos mediante un PIN (código de identificación de los aparatos), esto fue un gran acierto ya que cada vez conseguía mas y mas seguidores.

Al igual que sus competidores, posee el BlackBerry App World y es el mayor proveedor que recibe más ingresos por la venta de aplicaciones con \$9,166.67 seguido por los \$6,480.00 de App Store y los \$1,200.00 de Google Play.

Actualmente ocupa el tercer puesto del mercado mundial con una cuota del 4,3%.

1.3. Android

Anteriormente dimos una breve introducción de la plataforma Android ahora explicare el por que tome la decisión de usar esta plataforma.

La primera y la que más pesa es que debido a mis estudios, la programación en el lenguaje JAVA es donde me desenvuelvo mejor y Android utiliza la sintaxis y la semántica de JAVA.

En segundo lugar Android es una plataforma que está en constante crecimiento y actualmente se activan más de 700.000 móviles Smartphone al día con el este sistema operativo², por lo que cada vez llega a más personas y su mercado va creciendo.

Por último aunque no menos importante, he optado por dispositivos con una proyección amplia de mercado.

1.4. Google Play Store

La Google Play Store es la tienda oficial de software para dispositivos Android desarrollada por Google. Desde allí los usuarios pueden buscar, obtener información y descargar aplicaciones publicadas por desarrolladores de terceros.

Fue puesta a disposición de los usuarios en octubre del 2008 bajo el nombre del Android Market. Para el año de 2012 dicho servicio fue relanzado con una nueva apariencia y nuevo nombre y además contaba con mas de 450 mil aplicaciones. Actualmente la tienda tiene mas de 700 mil aplicaciones.

² Fuente: <http://www.ticbeat.com/sim/activan-700000-android-dia/>

2. Tecnologías utilizadas

2.1. Android

Es un sistema operativo basado en el kernel de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, inicialmente desarrollado por Android.

Según la documentación oficial: "Android es un conjunto de software para dispositivos móviles que incluye un sistema operativo, un middleware, y aplicaciones clave. La SDK de Android proporciona las herramientas y las API's necesarias para empezar a desarrollar aplicaciones para la plataforma Android, mediante el uso del lenguaje de programación Java."

2.2. JSON

Acrónimo de *JavaScript Object Notation*, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON. En JavaScript, un texto JSON se puede analizar fácilmente usando la función `eval()`, lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador web.

En la práctica, los argumentos a favor de la facilidad de desarrollo de analizadores o del rendimiento de los mismos son poco relevantes, debido a las cuestiones de seguridad que plantea el uso de `eval()` y el auge del procesamiento nativo de XML incorporado en los navegadores modernos. Por esa razón, JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia cuando la fuente de datos es explícitamente de fiar y donde no es importante el no disponer de procesamiento XSLT para manipular los datos en el cliente.

2.3. XAMPP

Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.

Aquí se ha utilizado para albergar la base de datos MySQL de productos del supermercado y el archivo PHP a través del cual conectamos con ella.

2.3.1. MySQL

Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet.

2.3.2. Apache

El servidor HTTP apache es un software diseñado para servir contenido HTTP de código libre. Es multiplataforma, es decir, que se puede disponer de él en sistemas Unix, Windows y Mac, entre otros.

Apache es un servidor muy bueno a la hora de configurar los múltiples parámetros que se pueden establecer, pero no dispone de una interfaz gráfica, por lo que toda la configuración debe realizarse mediante la modificación de

archivos de configuración y comandos de Shell.

Este servidor tiene vulnerabilidades de seguridad detectadas, pero solo pueden ser utilizadas por usuarios malintencionados que se encuentren en red local. Sin embargo, si se utiliza el modulo de PHP para Apache, algunas de estas vulnerabilidades pueden ser accedidas remotamente. En cualquier caso, este software presenta muchas ventajas, como son, entre otras, que se trata de software de código abierto, modular, multiplataforma, y muy popular, lo que hace que sea extremadamente sencillo encontrar ayuda o soporte.

2.3.3. PHP

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

2.4. SQLite

Es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña biblioteca escrita en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos.

El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

3. La aplicación Mi Carrito de la Compra

Se pretende realizar una aplicación que se ejecute en un dispositivo móvil que tiene como sistema operativo Android.

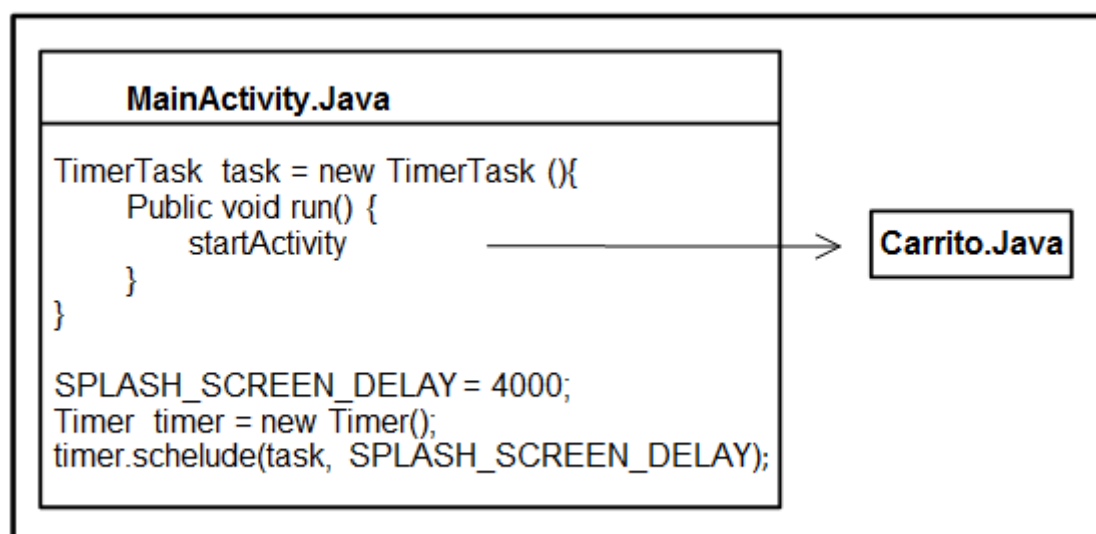
Esta aplicación se instalará en el dispositivo móvil del cliente permitiéndole, mediante el escaneado de los códigos de barras de los productos que pretende comprar, visualizar la lista de los productos mostrando su precio, la cantidad deseada así como los totales individuales y la suma total de estos. Dichas cantidades podrán ser modificadas por el cliente permitiendo incluso la eliminación de algún elemento de la lista o el borrado total de esta.

Por otra parte, será necesario disponer de un servidor para alojar la base de datos que contenga el listado de productos y los archivos PHP que servirán de puente entre esta base de datos y los dispositivos de los usuarios, que pueda recoger sus peticiones mediante el uso del protocolo HTTP.

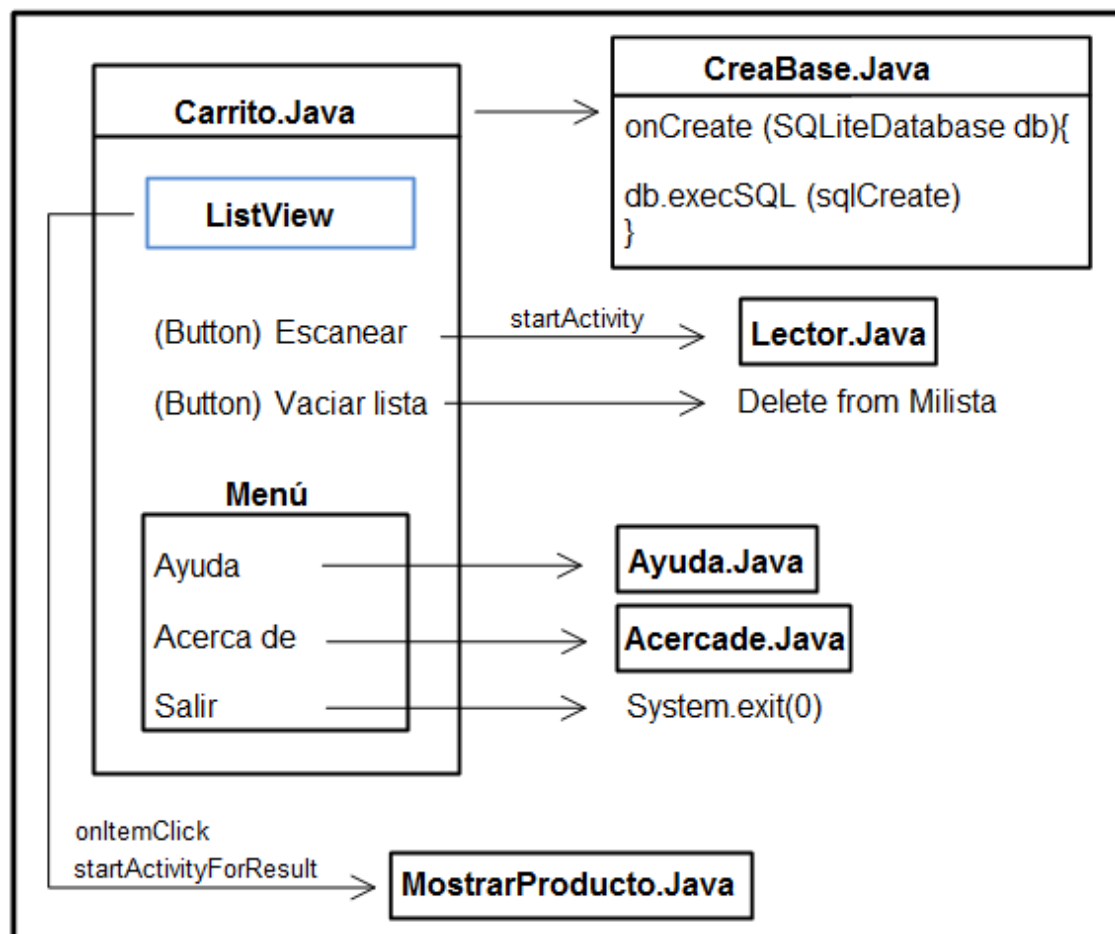
3.1. Esquemas de la aplicación

NOTA: Esta esquematización **no responde a ningún estándar**, pero es la forma gráfica de entender todo el control de flujo de la aplicación.

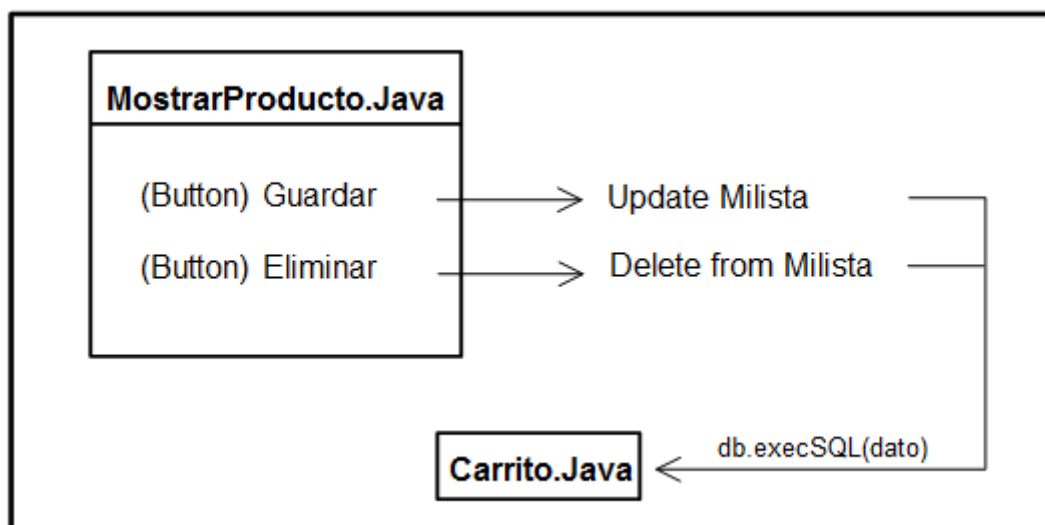
3.1.1. MainActivity



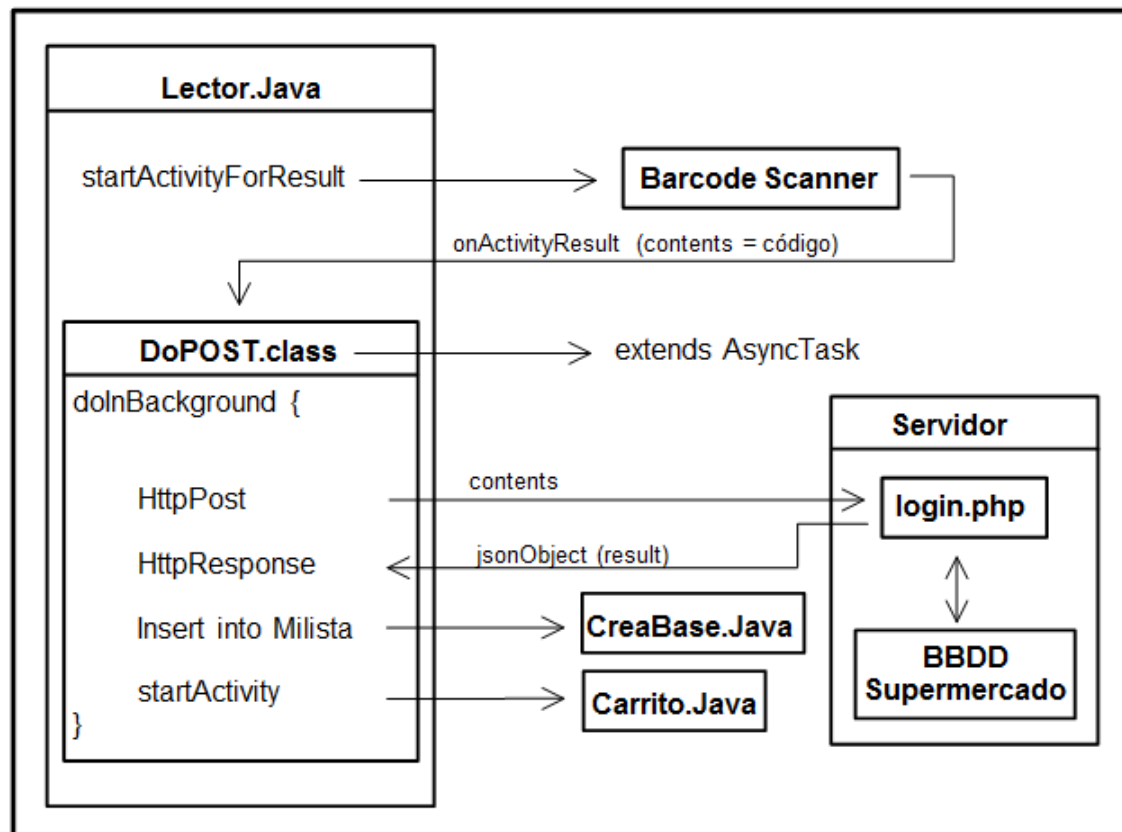
3.1.2. Carrito



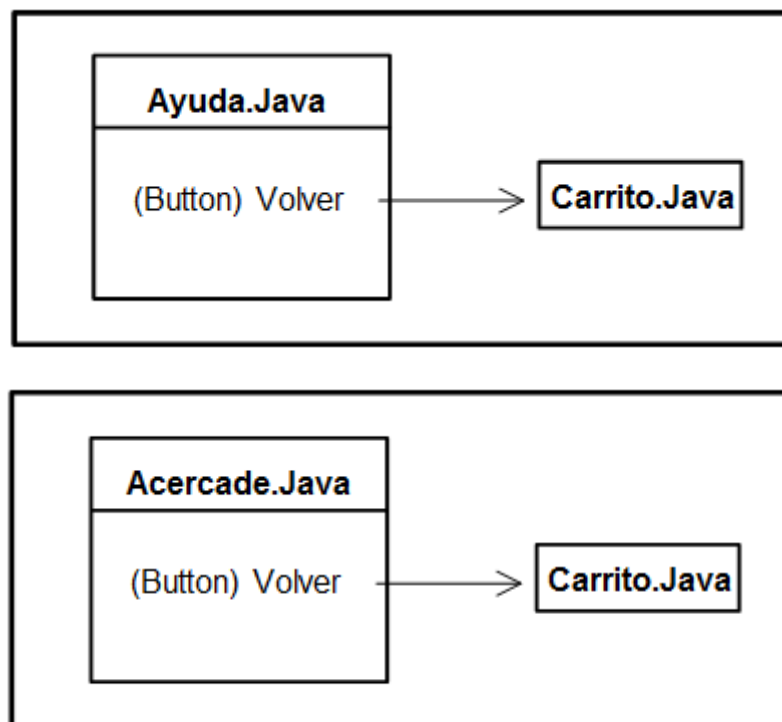
3.1.3. MostrarProducto



3.1.4. Lector



3.1.5. Ayuda y Acercade



3.2. Diseño cliente Android

La aplicación móvil es la parte principal del sistema junto con el servicio web con el que se conecta con la base de datos del supermercado. Esta aplicación será la que se ejecute en los dispositivos de los clientes, de tal manera que estos podrán interactuar con el servidor.

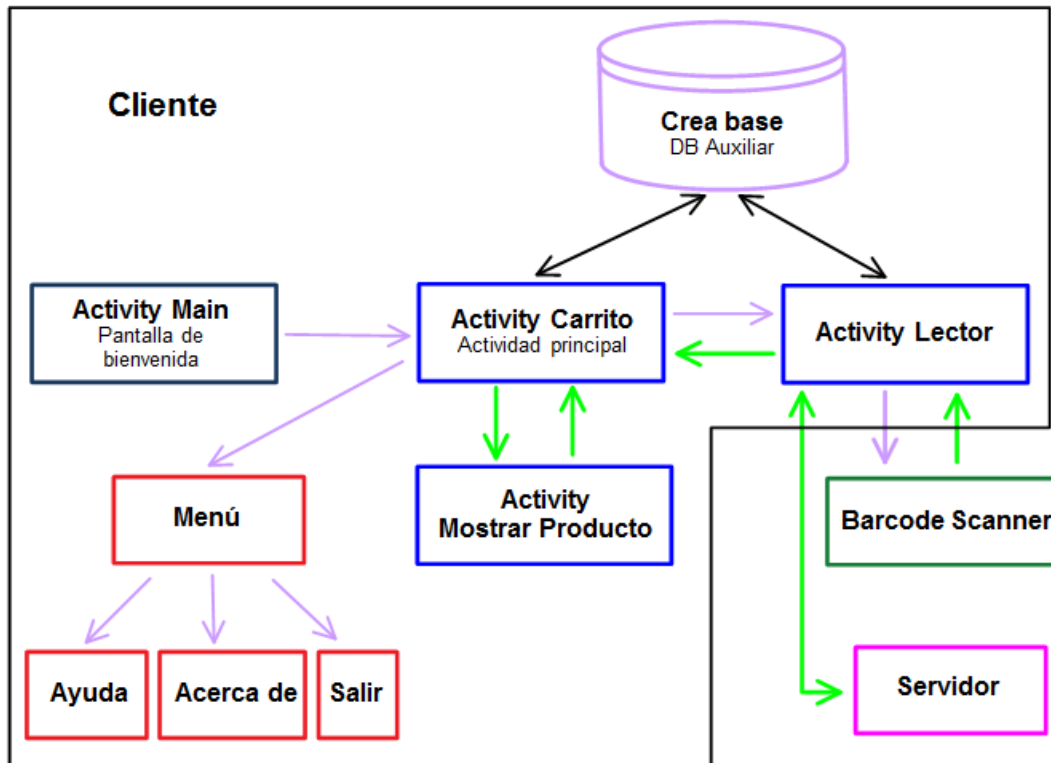


Figura 3.1 - Gráfico diseño cliente Android

A continuación explico la funcionalidad de la aplicación haciendo un breve repaso de todas las pantallas por las cuales podrá pasar el usuario.

La aplicación al ser ejecutada desde el icono que mostramos en la *Figura 3.1.1* nos lleva a una pantalla de carga (*Figura 3.1.2*) la cual podemos ver como empieza su ejecución. He escogido este título porque he creído que era bastante intuitivo a la hora de explicar con pocas palabras el concepto del programa. La aplicación está disponible en dos idiomas Español e Inglés los cuales se seleccionan dependiendo del idioma que este seleccionado en el dispositivo.



Tras varios segundos de carga la aplicación nos llevará directamente a la actividad principal (*Figura 3.1.3*). Esta actividad simula una hoja de papel en la cual anotaremos los productos. En ella nos encontramos los botones escanear y vaciar lista y podremos desplegar el menú de la aplicación (*Figura 3.1.4*).



El menú se compone de tres ítems, Ayuda, Acerca de y Salir, al pulsar este último abandonaremos la aplicación. Al pulsar el ítem Ayuda nos llevará a otra actividad (*Figura 3.1.5*) en la cual se nos muestra una breve descripción del funcionamiento de la aplicación. Al pulsar el ítem Acerca de (*Figura 3.1.6*) nos llevará a otra actividad en la cual se nos mostrará información sobre la aplicación.

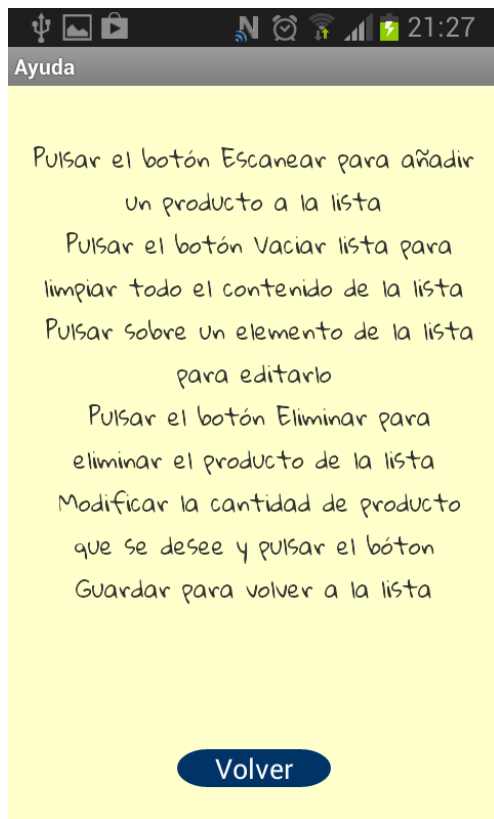


Figura 3.1.5 - Actividad Ayuda

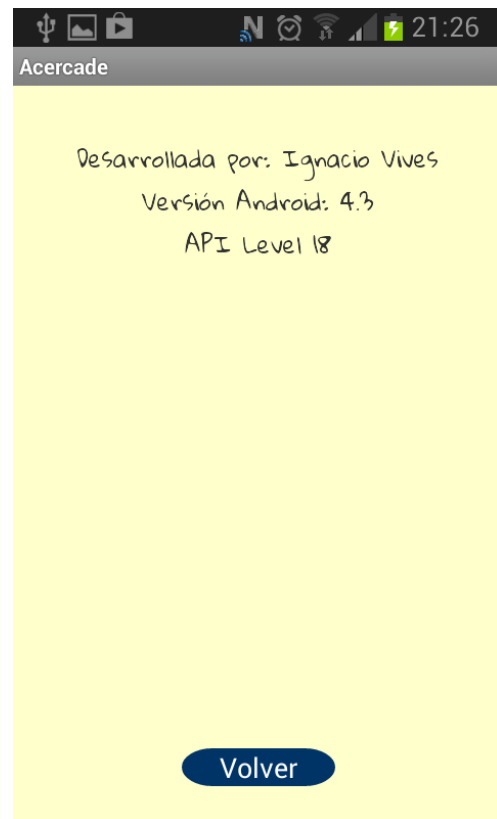


Figura 3.1.6 - Actividad Acerca de

Si pulsamos el botón volver en ambas pantallas volveremos a la actividad carrito.

En la actividad carrito, hay dos botones: Vaciar que permite limpiar la lista de productos y Escanear que nos lleva a la aplicación que gestiona la lectura de los códigos de barras de los artículos de la compra. Con el botón Escanear accedemos a la actividad Lector (no tiene pantalla propia) que hace una llamada a la aplicación externa "Barcode Scanner" (*Figura 3.1.7*) que nos permite escanear todo tipo de códigos de barras (*Figura 3.1.8*), devolviendo el número que va situado debajo de este. La aplicación Barcode Scanner debe estar instalada en el dispositivo, en caso contrario, el

botón Escanear, nos da acceso a Google Play Store para que la instalemos. Barcode Scanner presenta la pantalla para realizar el escaneo de los códigos de barras de los productos.

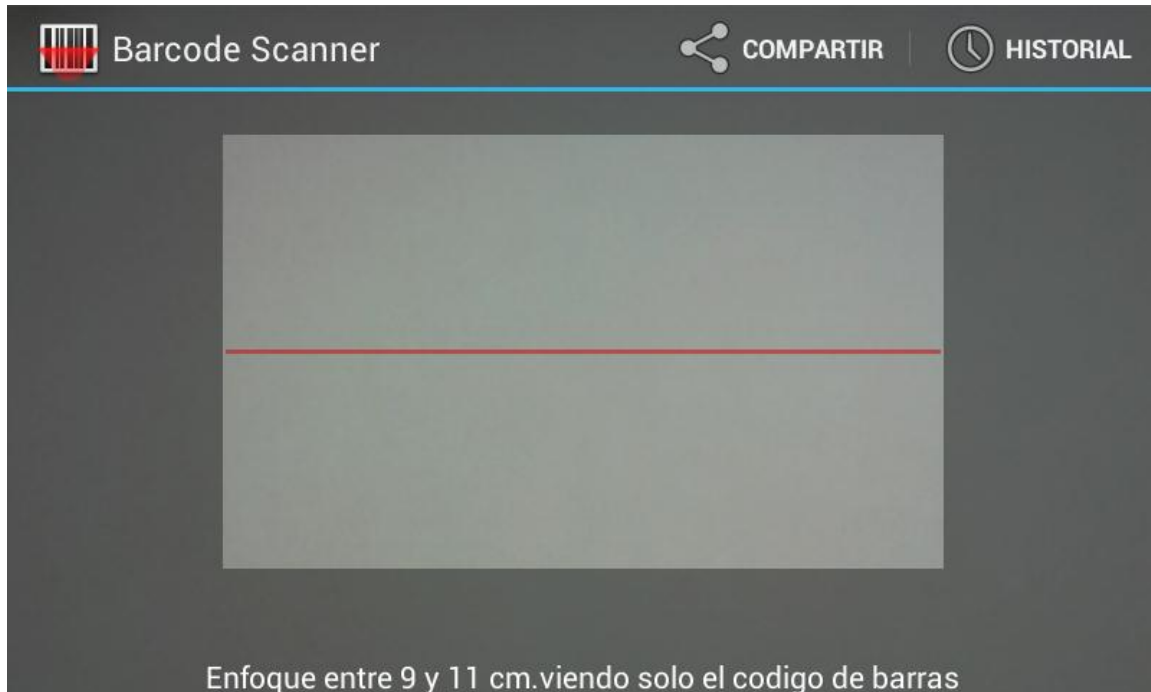


Figura 3.1.7 - Barcode Scanner



Figura 3.1.8 - Código de barras

Una vez escaneado el código de barras, la aplicación guardará los datos del producto en una base de datos auxiliar que creará en el dispositivo, y nos devolverá a la actividad Carrito (*Figura 3.1.9*) en la cual se visualizará el producto escaneado y se sumará su precio al cálculo total.

Ahora en esta actividad podemos seleccionar, pulsando sobre él, cualquiera de los productos que se encuentran en la lista, lo cual nos llevará a la actividad Mostrar Producto (*Figura 3.1.10*) en la cual se nos presenta la descripción del

producto, su precio y la cantidad que vamos comprar, pudiendo modificar esta última si queremos adquirir más unidades.

También encontramos dos botones, Eliminar que al pulsarlo eliminamos el producto de la lista y por tanto de la base de datos auxiliar volviendo a la actividad Carrito, y el botón Volver que al pulsarlo, guardaremos los cambios en la cantidad del producto y volveremos a la actividad Carrito, la cual quedará actualizada.



Figura 3.1.9 - Activity Carrito

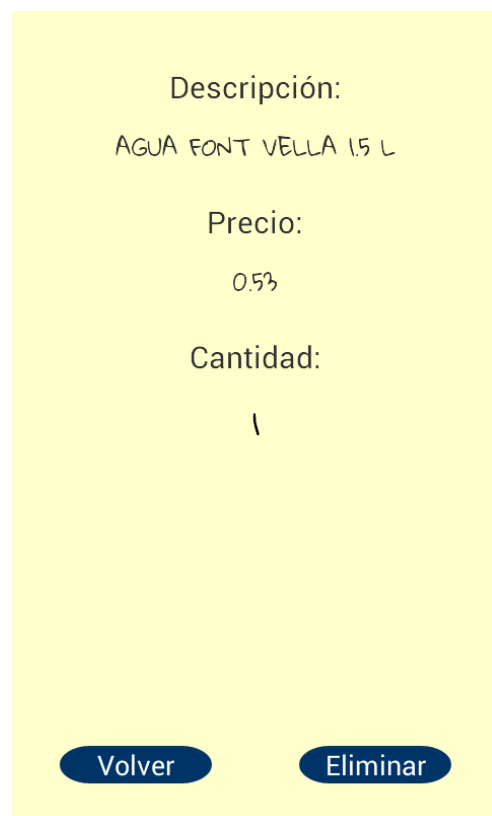


Figura 3.1.10 - Mostrar Producto

3.2.1. Implementación cliente Android

Para llevar a cabo el desarrollo de *Mi carrito de la compra* he tenido que utilizar ciertas tecnologías Android. Comentaré brevemente las más básicas para crear una aplicación. A continuación explicaré algunos trozos de código de la aplicación Android para que se pueda entender mejor su funcionamiento

Tecnologías básicas:

- **Eclipse³**: Es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar aplicaciones.

- **Java Development Kit⁴:** Se puede definir como un conjunto de herramientas, utilidades, documentación y ejemplos para desarrollar todo tipo de aplicaciones Java.
- **ADT Plugin⁵:** Es un *plugin* para el IDE (Entorno de Desarrollo Integrado) de Eclipse que está diseñado para crear un entorno potente e integrado con el fin de crear aplicaciones Android.
- **SDK de Android⁶:** Es un kit de desarrollo necesario para crear herramientas Android. En él podemos encontrar entornos de desarrollo, librerías, emuladores de móviles, documentación, tutoriales, códigos de ejemplo, etc.

Una vez instalado y configurado este entorno ya podemos pasar a desarrollar aplicaciones Android en nuestro ordenador. La plataforma para instalarlo puede ser GNU/Linux, OSX o Windows.

3.2.1.1. Actividades y XMLs

Habitualmente las aplicaciones se componen de Activities o actividades. Cuando se solicita una *Activity*, esta pasa al primer plano colocándose encima de otras, formado así una pila de Activities, cuando se cierra la Activity actual se recupera la primera de la fila. El código para crear la Actividad Carrito será el siguiente.

```
public class Carrito extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //Hacemos que la actividad ocupe toda la pantalla
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_carrito);
    }
}
```

³ Se puede obtener desde <http://www.eclipse.org>

⁴ Se puede obtener desde <http://www.oracle.com/technetwork/java/javase/downloads/jdk6-jsp-136632.html>

⁵ Se puede obtener desde <http://eddfox.blogspot.com.es/2011/09/como-instalar-adt-plugin-para-eclipse.html>

⁶ Se puede obtener desde <http://developer.android.com/sdk/index.html>

Durante la vida de un *Activity* esta pasa por una serie de estados. El siguiente esquema muestra el ciclo de vida de las actividades de la aplicación.

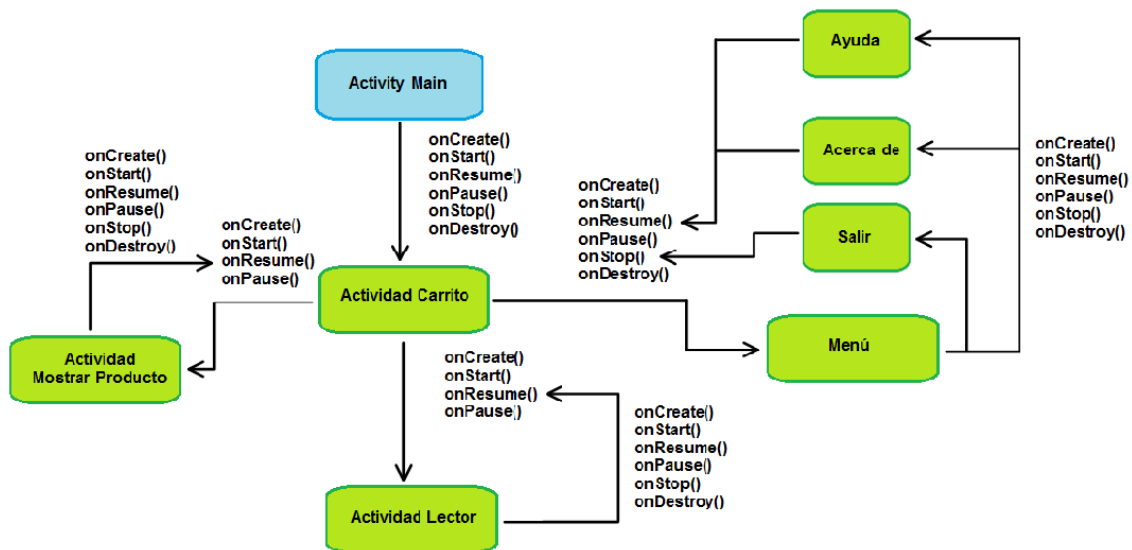


Figura 3.2.1 – Ciclo de vida de la aplicación

- **onCreate():** se invoca cuando el *Activity* arranca por primera vez. Se utiliza para tareas de inicialización.
- **onStart():** se invoca cuando un *Activity* va a ser mostrado al usuario.
- **onResume():** se invoca cuando un *Activity* pausado va a empezar a interactuar con el usuario.
- **onPause():** se invoca cuando el *Activity* va a pasar al fondo porque otro *Activity* ha sido lanzado para ponerse en primer plano. Se utiliza para conservar el estado del *Activity*.
- **onStop():** se invoca cuando el *Activity* va a dejar de ser visible y no va a ser usada durante un periodo de tiempo largo.
- **onDestroy():** se invoca cuando un *Activity* va a ser destruido.

Los **layouts** en **XML's** son aquellos ficheros llamados después de la creación de la clase para mostrar la interfaz de usuario creada y están formados por diversos componentes que configuran la vista de la actividad.

Estos son algunos ejemplos de componentes de la aplicación:

```

<Button
    android:id="@+id/vaciar"
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_marginRight="15dp"
    android:layout_alignParentRight="true"
    android:text="@string/vaciar"
    android:textSize="18dp"
    android:layout_marginBottom="5dp"
    android:background="@drawable/redondeo"
    android:textColor="@android:color/white"
/>

```

```

<ListView
    android:id="@+id/LstOpciones"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:layout_below="@+id/titulo"
    android:scrollbars="vertical"
    android:layout_marginBottom="80dp"
/>

```

```

<TextView
    android:id="@+id/titulo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/cabecera"
    android:layout_alignParentTop="true"
    android:layout_centerInParent="true"
    android:textSize="20dp"
    android:typeface="serif"
    android:textColor="#003366"
/>

```

También he usado estos XML para dotar a la aplicación de dos idiomas posibles (Español e Inglés), esto le he conseguido creando dos carpetas values (values y values-en) en cada una de ellas encontramos el archivo strings.xml con los textos en el idioma correspondiente.

```

<string name="app_name">My Cart</string>
<string name="action_settings">Settings</string>
<string name="titulo">My shopping cart</string>
<string name="escanear">Scan</string>
<string name="compra">Start</string>

```

Ejemplo de textos en inglés

```

<string name="app_name">Mi carrito</string>
<string name="action_settings">Settings</string>
<string name="titulo">Mi carrito de la\nCompra</string>
<string name="escanear">Escanear</string>
<string name="compra">Comenzar</string>

```

Ejemplo de textos en español

3.2.1.2. El menú de la aplicación

En la actividad carrito he definido un menú, para ello he implementado el evento `onCreateOptionsMenu()` (Figura 3.1.2.1), en este evento se debe “inflar” el menú, obteniendo una referencia al *inflater* mediante el método `getMenuInflater()` y posteriormente se genera la estructura del menú llamando a su método `inflate()` pasándole como parámetro el ID del menú definido en XML, que en mi caso será `R.menu.main` (Figura 3.1.2.2). Por último devolveremos el valor `true` para confirmar que debe mostrarse el menú.

La implementación de cada una de las opciones se incluye en el evento `onOptionsItemSelected()` (Figura 3.1.2.3) de la actividad que muestra el menú.

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
```

Figura 3.1.2.1 - Evento `onCreateOptionsMenu`

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <item
        android:id="@+id/ayuda"
        android:title="@string/ayuda"
        android:icon="@android:drawable/ic_menu_help" />

    <item
        android:id="@+id/acercade"
        android:title="@string/acercade"
        android:icon="@android:drawable/sym_contact_card" />

    <item
        android:id="@+id/salir"
        android:title="@string/salir"
        android:icon="@android:drawable/ic_lock_power_off"
        />

</menu>
```

Figura 3.1.2.2 - Definición XML del menú

```

@Override
public boolean onOptionsItemSelected(MenuItem item){
    switch (item.getItemId()){
        case R.id.ayuda:
            Intent intent = new Intent(Carrito.this, Ayuda.class);
            startActivity(intent);
            return true;
        case R.id.acercade:
            Intent intent1 = new Intent(Carrito.this, Acercade.class);
            startActivity(intent1);
            return true;
        case R.id.salir:
            System.exit(0);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

```

Figura 3.1.2.3 - Evento onOptionsItemSelected

3.2.1.3. Almacenamiento de datos

La aplicación Mi carrito de la compra contiene una gran capacidad de datos ya que debe gestionar muchos tipos de productos. Para ello he necesitado tener una base de datos en el servidor (MySQL) y otra en la aplicación (SQLite). Para empezar vamos a explicar de qué manera he gestionado el almacenamiento de datos en Android, esta BD (Base de Datos) se almacena en la carpeta del proyecto “/data/data/<com.example.proyecto2>/dbase”.

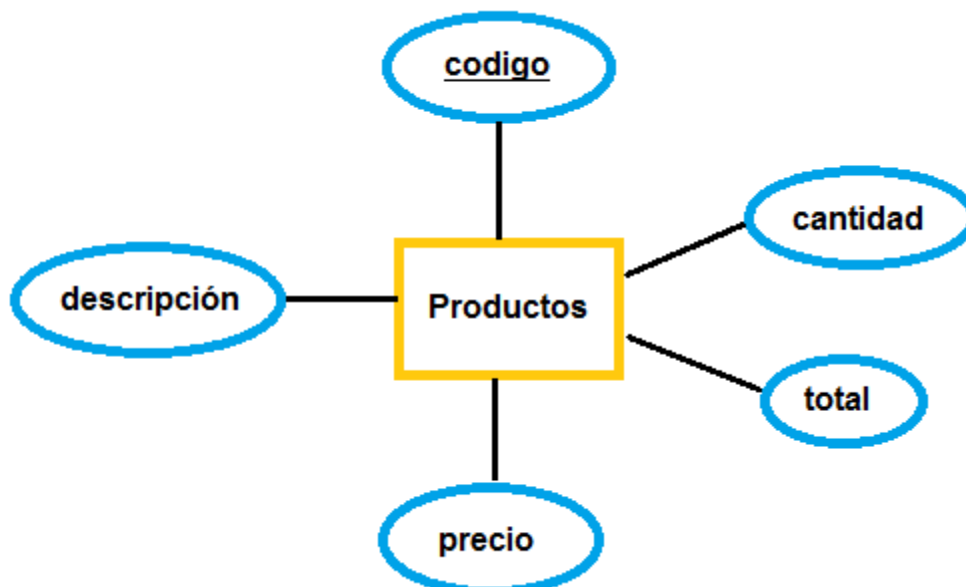


Figura 3.1.3.1 - Esquema entidad-relación

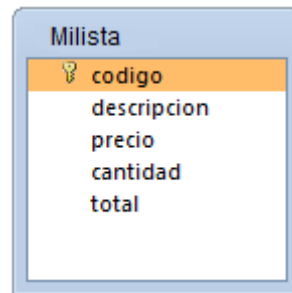


Figura 3.1.3.2 - Tabla Milista

Para acceder a la base de datos trabajamos con cursores, a continuación muestro el cursor que he empleado para recorrer la base de datos y obtener la suma de los totales.

```
//Obtengo de la db la suma de los totales individuales y la muestro por pantalla
double columntotal = 0;
SQLiteDatabase db=base.getReadableDatabase();
Cursor cursor1 = db.rawQuery("SELECT SUM(total) FROM Milista", null);
if(cursor1.moveToFirst()) {
    columntotal = cursor1.getDouble(0);
}
```

En la siguiente imagen veremos la acción para crear la base de datos SQLite y la tabla Milista que se alojarán en el dispositivo móvil del cliente.

```
public class CreaBase extends SQLiteOpenHelper{

    public CreaBase(Context context, String name, CursorFactory factory, int version) {
        super(context, "dbase", factory, version);
    }

    //Sentencia SQL para crear la tabla de productos
    String sqlCreate = "create table Milista ("
        + "codigo text primary key not null, "
        + "descripcion text, "
        + "precio double, "
        + "cantidad int, "
        + "total double);|

    @Override
    public void onCreate(SQLiteDatabase db)
    {

        db.execSQL(sqlCreate);
    }
}
```

Para insertar o actualizar una fila de la tabla he usado las funciones estándar *insert* o *update* y *delete* para eliminar filas.

Sentencia para insertar un producto en la base de datos

```
String valor = "INSERT INTO Milista VALUES ("
              + ""+strCodigo+", "
              + ""+strDescripcion+", "
              + ""+douPrecio+", "
              + ""+cantidad+", "
              + ""+total+");"
```

Sentencia para actualizar la cantidad de producto

```
String sentencia = "UPDATE Milista SET cantidad = ''+can+', total = ''+total+' WHERE descripcion = ''+des+'";"
```

Sentencia para eliminar un producto de la lista

```
String sentencia = "DELETE FROM Milista WHERE descripcion = ''+des+'";"
```

3.2.1.4. JSON y HTTP

Para realizar la consulta a la base de datos del supermercado, he usado las clases HTTP que proporciona Java y JSON como encapsulamiento de datos.

Esta consulta se encuentra dentro de la clase *DoPOST* la cual he hecho que extienda *AsyncTask* (con esto he creado una tarea asincrónica y así realizar la consulta en segundo plano).

Preparamos los parámetros en una variable de tipo *ArrayList* que serán los datos a enviar, y una URL. Lo que hace es simplemente realizar la petición POST a la URL enviándole los datos, obtener la respuesta (Que estará en formato JSON) y guardarla.

Después cogemos el String *result* y lo transformamos de nuevo en un objeto JSON para después poder extraer cada una de sus partes.

```

//Configuración de los parámetros
ArrayList<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();
nameValuePairs.add(new BasicNameValuePair("CodeToSearch", strCodeToSearch));

//Creación de la petición HTTP
HttpParams httpParameters = new BasicHttpParams();

//Tiempos de espera para la petición
HttpConnectionParams.setConnectionTimeout(httpParameters, 15000);
HttpConnectionParams.setSoTimeout(httpParameters, 15000);

//Hacemos la petición a la base de datos
HttpClient httpClient = new DefaultHttpClient(httpParameters);
HttpPost httppost = new HttpPost("http://192.168.1.103/clienteservidor/login.php");//local
httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
HttpResponse response = httpClient.execute(httppost);
HttpEntity entity = response.getEntity();

String result = EntityUtils.toString(entity);

// Creamos un objeto JSON con la respuesta de la petición
JSONObject jsonObject = new JSONObject(result);

//Recuperamos los datos del objeto JSON
strCodigo = jsonObject.getString("codigo");
strDescripcion = jsonObject.getString("descripcion");
douPrecio = jsonObject.getDouble("precio");

```

3.3. El servidor

El servidor de este proyecto está compuesto por una base de datos "Supermercado" (Anexo B) con una tabla "Productos".

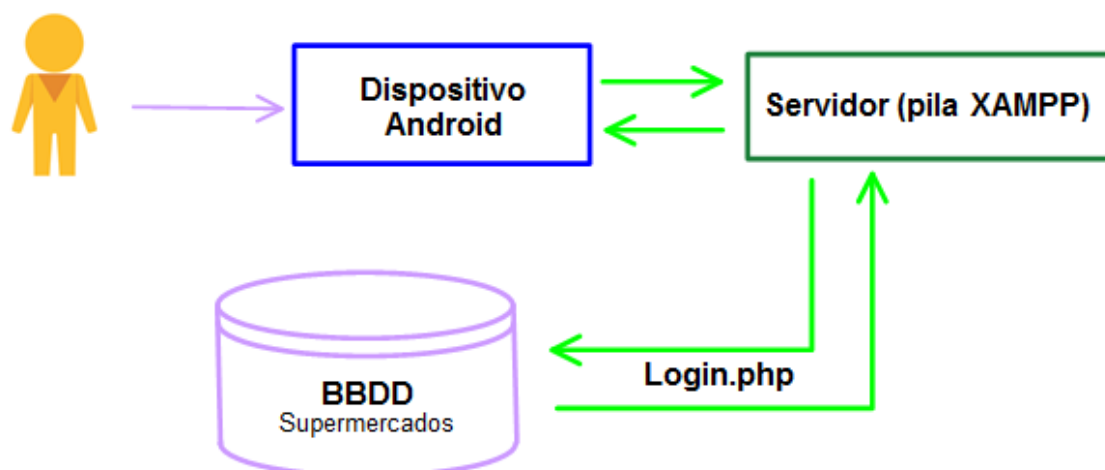


Figura 3.2 - Gráfico diseño servidor

Para crear la base de datos he utilizado la herramienta de *phpmyadmin* incluida dentro del paquete XAMPP (explicado anteriormente) y un archivo `login.php`, este recoge el código de barras enviado por la petición POST del dispositivo

móvil, se conecta con la base de datos del Supermercado, realiza la consulta "SELECT * FROM productos WHERE código = '\$code'" y devuelve en formato JSON la descripción y el precio del producto escaneado.

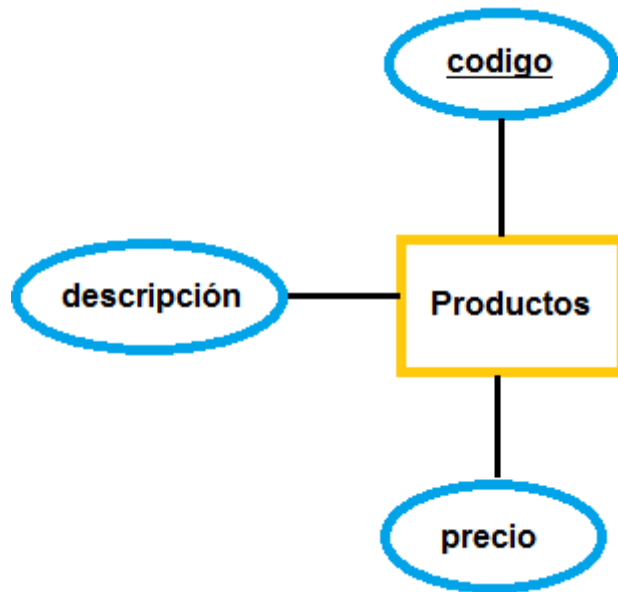


Figura 3.2.1 - Esquema entidad-relación

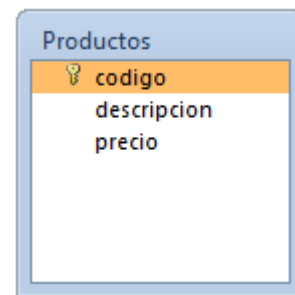


Figura 3.2.2 - Tabla productos

Código del archivo login.php

```
<?php
#Nos aseguramos de que el cliente ha proporcionado un valor para "CodeToSearch"
if (isset($_POST["CodeToSearch"]) && $_POST["CodeToSearch"] != ""){
#Variable de configuracion
$code = $_POST["CodeToSearch"];
#Conexión a la base de datos
$DB_SERVER="localhost";
$DB_USER="root";
$DB_PASS=""; #Modificamos esta línea si se ha establecido una contraseña para el usuario
root en MySQL
$DB_DATABASE="supermercado";
$con = mysqli_connect($DB_SERVER, $DB_USER, $DB_PASS, $DB_DATABASE);
#Comprobamos la conexión
if (mysqli_connect_errno()) {
    echo 'Error de conexion a la base de datos<br>';
    echo 'Database connection error: ';
    exit();
}
#Escapamos caracteres especiales para evitar ataques de inyección SQL
```

```

$code = mysqli_real_escape_string($con, $code);
#Consultamos la base de datos para obtener los datos de los productos
$productdetails = mysqli_query($con, "SELECT * FROM productos WHERE código = '$code'");
#Si no se devuelven datos, comprobamos que no existen errores de SQL
if (!$productdetails) {
    echo 'No se pudo ejecutar la consulta<br>';
    echo 'Could not run query';
    exit();
}
#Obtenemos la primera fila de los resultados
$row = mysqli_fetch_row($productdetails);
#Construimos el array resultante (Asignamos claves a los valores)
$result_data = array(
    'código' => $row[0],
    'descripción' => $row[1],
    'precio' => $row[2],
);
#Salida de los datos JSON
echo json_encode($result_data);
}else{
    echo "No se pudo completar la consulta. Falta un parámetro<br>";
    echo "Could not complete query. Missing parameter";
    exit();
}
?>

```

4. Resultados y conclusiones

Durante el proyecto he reforzado mis conocimientos del lenguaje de programación Android. El trabajo para llevar a cabo todos los objetivos ha sido muy exhaustivo ya que conforme avanzaba me he encontrado con dificultades las cuales me han obligado a estudiar más a fondo el funcionamiento de Android.

También he tenido la oportunidad de ampliar mis conocimientos y la experiencia que tenía en el proceso de desarrollo de un proyecto. Además he podido aplicar gran cantidad de conocimientos que he adquirido a lo largo de la ciclo como son el diseño de las estructuras de datos, la orientación a objetos, el

uso de las bases de datos, uso de herramientas de programación, conexiones servidores – cliente, desarrollo de documentación, etc.

He intentado que el programa posea ciertas características:

Usable: Lo que pretendía es que Mi Carrito de la Compra fuese muy cómoda de usar y el usuario se sintiera cómodo usándola.

Eficiente: La rapidez de respuesta en una aplicación es muy importante tanto en las aplicaciones móviles como el software de ordenadores.

Sencillo: La intención de la aplicación es que sea fácil de usar y con pocas funcionalidades pero que sea efectivas y sin problemas. Un programa muy complejo para una aplicación móvil puede resultar muy pesado e incomodo a la hora de utilizarlo.

Integrado con Internet (o la nube): Actualmente la gran mayoría de Smartphones están conectados a internet y el uso y compartición de información es muy común.

Finalmente podemos decir que, con estos objetivos cumplidos, hemos alcanzado la meta de la aplicación con éxito.

5. Trabajos futuros

Los principales aspectos que se pueden modificar serán los siguientes:

- Mejorar el aspecto de la aplicación.
- Conectar la aplicación con alguna base de datos real (existen en el mercado aplicaciones que comparan precios de distintos supermercados), esto conllevaría un estudio profundo de su estructura y funcionamiento interno.
- Mejorar la funcionalidad de la aplicación añadiendo por ejemplo que los productos puedan ser borrados de la lista si se realiza una pulsación larga sobre ellos.
- Añadir a la aplicación una plataforma de pago con la cual si un comercio distribuye a sus clientes la aplicación, estos puedan pagar antes de llegar a la caja y en esta sólo habría que revisar la compra.
- Se podría hacer un registro de clientes y almacenar en el sus historiales de compra.

- Se podría permitir a los clientes hacer pedidos y que les fueran enviados donde quisieran.
- Se podría añadir una interfaz web desde la cual se podría gestionar la base de datos de productos y gestionar los datos de los clientes así como sus pedidos.

A. ANEXO: Manual del usuario

Servidor

Lo primero que tendremos que hacer es instalar el servidor. Para este proyecto se ha usado XAMPP. (<https://www.apachefriends.org/es/download.html>)

Una vez que tenemos XAMPP instalado en nuestro equipo tendremos que copiar la carpeta "clienteservidor" del proyecto y pegarla en el directorio htdocs de XAMPP. Después iniciaremos XAMPP y comprobaremos que los módulos Apache y MySQL están funcionando (*Figura 6.1*).

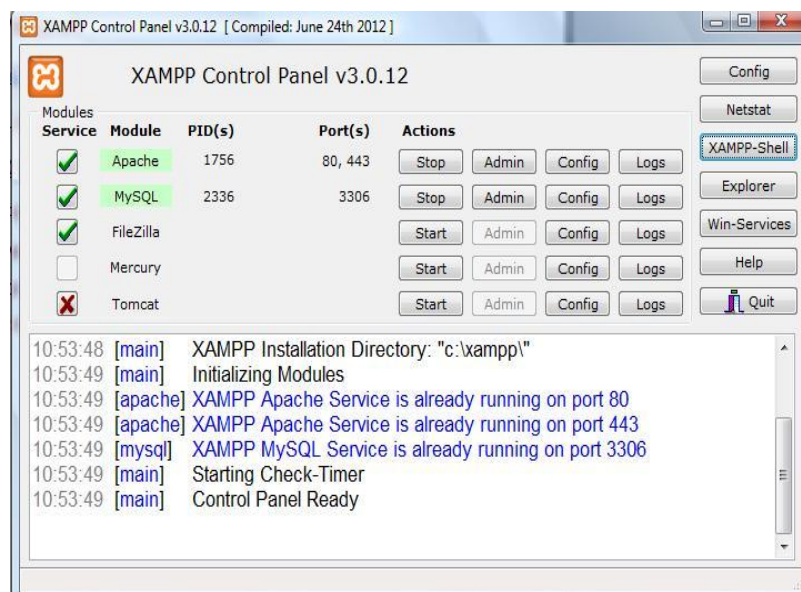


Figura 6.1 – XAMPP Control Panel

Para cargar la base de datos en el navegador iremos a <http://localhost/phpmyadmin> (*Figura 6.2*).

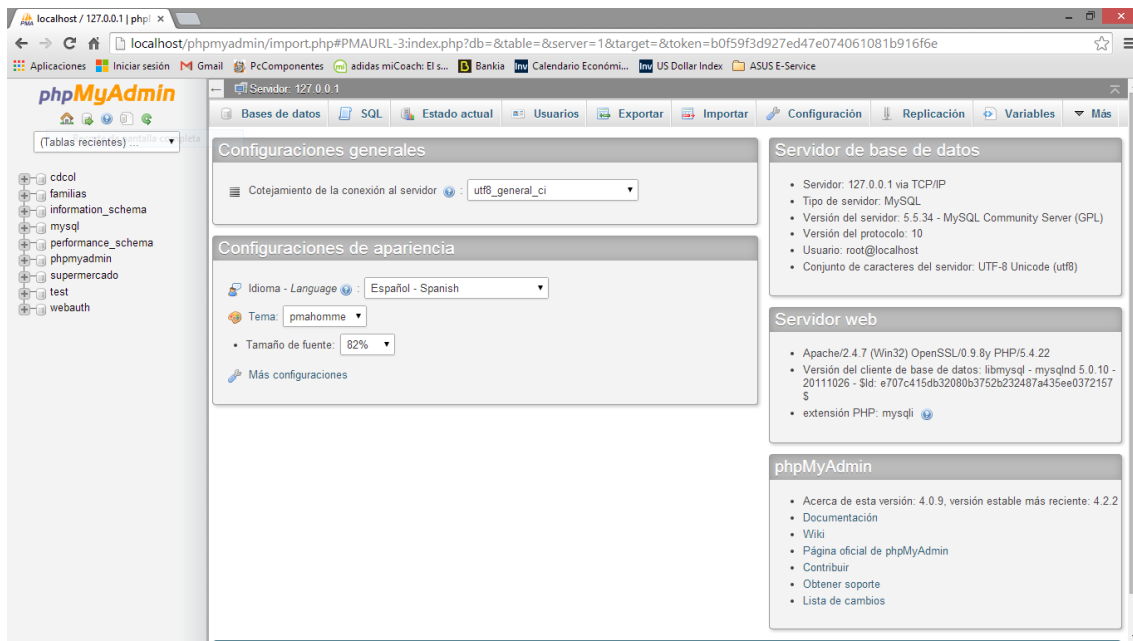


Figura 6.2 – PHPMyAdmin

Seleccionamos la pestaña *Bases de datos*, introducimos el nombre que queramos darle a la base de datos y pulsamos *Crear*. Ahora debemos seleccionar la base de datos creada en el panel de la izquierda, seleccionamos la pestaña *Importar*, en esta, seleccionamos el archivo que vamos a importar, en nuestro caso *productos.sql*, pulsamos *continuar*. Con esto ya la tenemos creada.

Aplicación móvil

Antes de instalar la aplicación en nuestro dispositivo deberemos modificar el código de la actividad *Lector* en la línea 164 donde se encuentra la IP del servidor y sustituirla por la del servidor propio. Una vez instalada, pulsar sobre el icono de la aplicación, lo primero que nos encontramos es una pantalla de bienvenida la cual transcurridos 4 segundos nos lleva a la actividad principal (*Carrito*).

Una vez aquí podemos, desplegar el menú d la aplicación para obtener más información y abandonar la aplicación, o bien comenzar a llenar nuestro carrito de la compra.

Si pulsamos el botón *Escanear* accedemos a otra pantalla en la cual podemos leer los códigos de barras de los productos y en la cual los almacenaremos en la base de datos auxiliar creada en el dispositivo.

Una vez escaneado el código nos devolverá a la activity principal en la cual aparecerá el producto anotado en la lista y la suma de los totales de los productos que hay en ella. En esta activity también encontramos el botón *Vaciar lista* el cual al ser pulsado vacía totalmente la lista y la base de datos auxiliar.

Si pulsamos sobre un producto de la lista, se nos mostrará una nueva activity en la cual veremos la descripción del producto, su precio y la cantidad de este que deseamos comprar (por defecto esta cantidad es 1) y si pulsamos sobre la cantidad podemos modificarla a nuestro antojo.

Esta activity tiene dos botones, *Eliminar* el cual al pulsarlo elimina el producto de la lista y de la base de datos, devolviéndonos a la activity principal, y el botón *Guardar* que al ser pulsado modifica la cantidad de producto y nos devuelve a la activity principal.

B. ANEXO: Base de datos Supermercado

Base de datos: `supermercado`

Estructura de tabla para la tabla `productos`

```
CREATE TABLE IF NOT EXISTS `productos` (  
  `codigo` bigint(13) NOT NULL,  
  `descripcion` varchar(50) CHARACTER NOT NULL,  
  `precio` double NOT NULL,  
  PRIMARY KEY (`codigo`));
```

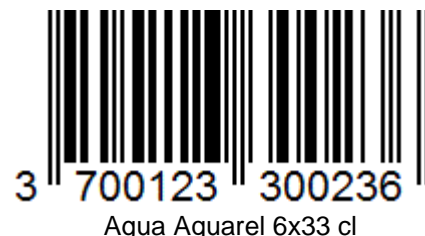
Volcado de datos para la tabla `productos`

```
INSERT INTO `productos` (`codigo`, `descripcion`, `precio`) VALUES  
(3600541238961, 'MASCARILLA FRUCTIS GARNIER', 4.5),  
(3700123300236, 'AGUA AQUAREL 6x33 CL', 1.5),
```

(4015600715755, 'ARIEL ACTILIFT 30 LAV', 6.99),
(5029053038001, 'SCOTTEX 24 UDS', 6.95),
(5449000000996, 'COCACOLA LATA', 0.56),
(5449000006004, 'FANTA LIMON LATA', 0.43),
(5449000011527, 'FANTA NARANJA LATA', 0.43),
(5449000131805, 'COCACOLA ZERO LATA', 0.56),
(5906747316013, 'OSITO LULU FONTANEDA', 1.7),
(7622210055828, 'QUESO PORCIONES 24 UDS', 2.3),
(8410014232335, 'COLACAO 3 KG', 14.95),
(8410014308610, 'NOCILLA 400 GR', 2.5),
(8410055150018, 'AGUA FONT VELLA 1.5 L', 0.53),
(8410069006585, 'GALLO HELICES CON VEGETALES', 1.05),
(8410069010803, 'GALLO LAZOS CON VEGETALES', 1.05),
(8410084041110, 'ARROZ LA FALLERA', 1.85),
(8410090104311, 'ATUN CLARO CALVO 6 UDS', 4.45),
(8410100065144, 'PURE PATATAS MAGGI', 2.2),
(8410100180045, 'PAPILLA 8 CEREALES 1200 GR', 6.2),
(8410128276003, 'BIFRUTAS PASCUAL 6 UDS', 2.56),
(8410180225087, 'GALLETAS DINO', 2.09),
(8410300349006, 'CALDO COCIDO GALLINA BLANCA', 1.95),
(8410300349044, 'CALDO CARNE GALLINA BLANCA', 1.95),
(8410300349051, 'CALDO POLLO GALLINA BLANCA', 1.95),
(8410500006266, 'VITALINEA NATURAL EDUL 4 UDS', 1),
(8410500006273, 'VITALINEA LIMON 4 UDS', 1),
(8410714102129, 'AZUCAR 1 KG', 0.93),
(8411384005062, 'HUEVOS FRESCOS 12 UDS', 1.49),
(8411700005837, 'BATIDO CACAO PULEVA 6 UDS', 1.72),
(8411831500164, 'VINAGRE MODENA 250 ML', 1.59),
(8412500927992, 'PATE LA PIARA 3 LATAS', 2.85),
(8414100314257, 'SUNNY FLORIDA 4 UDS', 1.34),
(8480000180025, 'ATUN ACEITE OLIVA 6 LATAS', 3.95),
(8480000260390, 'GARBANZOS BOTE 300 GR', 0.43),
(8480000450814, 'JABON DE MANOS', 0.75),
(8480000478160, 'TOALLITAS HUMEDAS WC', 1.95),

(8480000497505, 'BOLSAS BASURA 40 UDS', 1.6),
 (8480000771155, 'BASTONCILLOS ALGODON', 0.55),
 (8712000010294, 'HEINEKEN 8 LATAS', 4.4),
 (8722700090366, 'CREMA VERDURAS KNORR', 1.45),
 (8722700090571, 'CREMA CALABAZA KNORR', 1.45),
 (8722700179337, 'CREMA ZANAHORIA KNORR', 1.45),
 (8722700781523, 'CREMA SETAS DEL BOSQUE KNORR', 1.45),
 (8722700781646, 'CREMA CALABACIN KNORR', 1.45);

C. ANEXO: Códigos de barras





Vinagre de Modena 250 ml



Atún aceite oliva 6 latas



Azucar 1 Kg



Bolsas basura 40 uds.



Caldo cocido Gallina Blanca



Cola Cao 3 Kg



Crema calabaza Knorr



Crema verduras Knorr



Ariel actilift 30 lavados



Atún claro Calvo 6 latas



Bifrutas Pascual 6 uds.



Caldo carne Gallina Blanca



Caldo pollo Gallina Blanca



Crema calabacín Knorr



Crema setas del bosque Knorr



Crema zanahoria Knorr



Agua Font Vella 1.5 l



Garbanzos 300 gr



Lata CocaCola zero



Lata Fanta Limón



Mascarilla Fructis Garnier



Nocilla 400 gr



Pate La Piara 3 latas



Papel higiénico Scottex 24 uds.



Galletas Dino



Jabón de manos



Lata CocaCola regular



Lata Fanta Naranja



Papilla 8 cereales Nestle



Osito Lulu Fontaneda



Queso porciones El Caserio 24 uds.



Toallitas húmedas wc



Vitalínea limón 4 uds.



Vitalínea nat. Edulcorado 4 uds.

Bibliografía:

En la parte de bibliografía mostraré todas fuentes de información a través de las cuales he extraído o consultado algún tipo de información para desarrollar mi proyecto.

Manual para programar Android - <http://www.sgoliver.net/blog/>

Manual para programar Android - <http://www.elbauldelprogramador.com/programación/>

Lectura de códigos de barras - <http://expocodetech.com/lectura-de-codigo-de-barras-en-android/>

Generador de códigos de barras - <http://barcode.tec-it.com/barcode-generator.aspx?LANG=es>

Conexión de Android y MySQL con web services - <http://dbarrerap.wordpress.com/2013/07/23/android-y-web-services/>

ListAdapter de Android - <http://es.scribd.com/doc/61548244/7/ListAdapter-ArrayAdapter-SpinnerAdapter-SimpleCursorAdapter>

Tutorial configuración Android - <http://yosoyandroid.com/2011/05/mis-primeros-pasos-en-programacion-android-mi-primer-hello-world/>

Guía Android - <http://developer.android.com/guide>

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