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: **Mª 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

2.4. MySQL

2.5. Apache

2.6. PHP

2.7 SQLite

3. La aplicación Mi Carrito

3.1. Iniciando la aplicación

3.2. Implementación

3.2.1. Activities y XMLs

3.2.2. El menú de la aplicación

3.2.3. Almacenamiento de datos

3.2.4. El servidor

3.2.5. JSON y HTTP

4. Resultados y conclusiones

5. Trabajos futuros

A. ANEXO: Manual de usuario

Servidor

Aplicación móvil

B.ANEXO: Base de datos Supermercado

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.

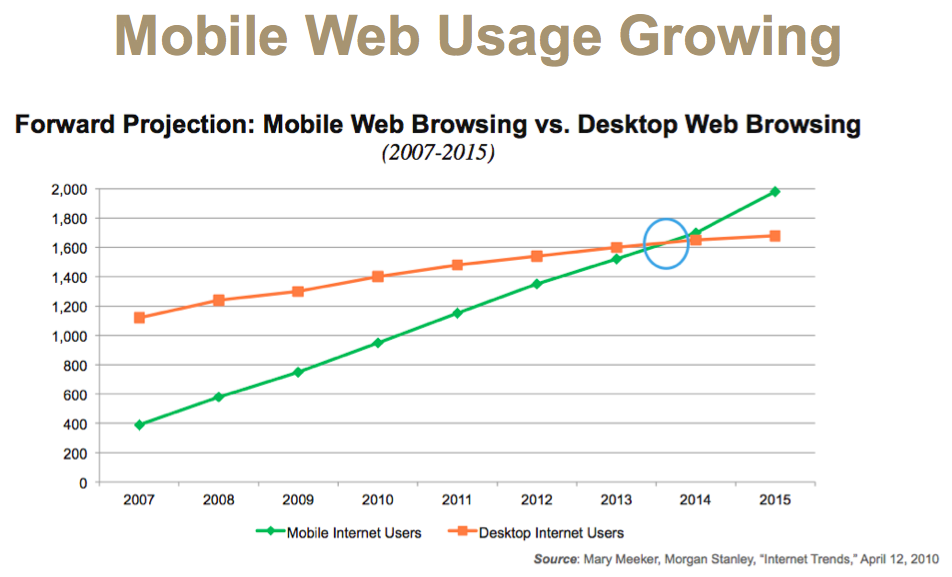
**1.1. Teléfonos Móviles**

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.



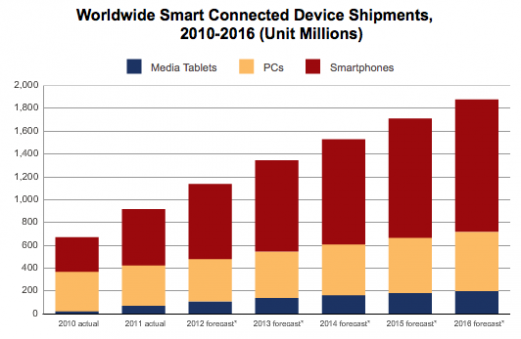
Evolución de los dispositivos móviles

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.



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.

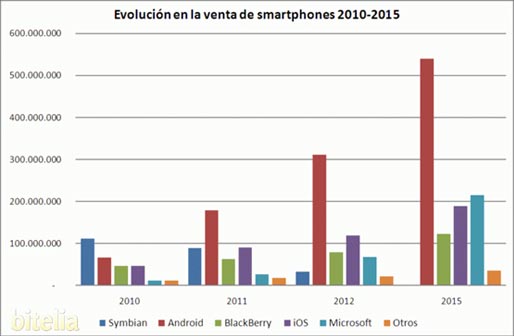


**1.2. Sistemas Operativos**

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 mas 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 esta 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 mas 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, desde ya hace un tiempo he venido utilizando y adquiriendo diversos dispositivos con la plataforma android lo que me facilita mucho la faena a la hora de probar la aplicación para ver su desempeño en los diversos artefactos.

**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](#documentacion oficial android): "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.

En este proyecto 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.

**2.4. 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.

**2.5. Apache**

El servidor HTTP [apache](#documentacion ofical apache) es un software diseñado para servir contenido HTTP de código libre. Es multiplataforma, es decir, que se puede disponer de el 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 grafica, 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.

**2.6. 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.

**2.7. 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 sistema 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 dichos productos mostrando su precio, la cantidad deseada así como los totales individuales por productos 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, los archivos PHP que servirán de puente entre esta base de datos y los dispositivos de los usuarios, y que pueda recoger las peticiones de dichos dispositivos mediante el uso del protocolo HTTP.

**3.1. Iniciando la aplicación**

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.

|  |  |
| --- | --- |
| D:\proyecto\carrito.jpg  Figura 3.1.1 - Icono aplicación | C:\Users\IGNACIO\Documents\pantallainicio.png  Figura 3.1.2 - Pantalla de carga |

Tras varios segundos de carga la aplicación nos llevará directamente a la activity principal *(Figura 3.1.3)*. Esta activity 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)*.

|  |  |
| --- | --- |
| C:\Users\IGNACIO\Documents\listavacia.png  Figura 3.1.3 - Vista principal | C:\Users\IGNACIO\Documents\vistamenu.png  Figura 3.1.4 - Vista menú |

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 activity *(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 activity en la cual se nos mostrará información sobre la aplicación.

|  |  |
| --- | --- |
| C:\Users\IGNACIO\Documents\atuda.png  Figura 3.1.5 - Activity Ayuda | C:\Users\IGNACIO\Documents\acercade.png  Figura 3.1.6 - Activity Acerca de |

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

Volviendo a esta activity carrito, si pulsamos el botón Vaciar lista limpiamos todos los productos que en ese momento se encuentran en la lista, mientras que si pulsamos el botón Escanear accedemos a la activity Lector (no tiene pantalla propia) en la cual la aplicación hace una llamada a otra aplicación externa llamada Barcode Scanner *(Figura 3.1.7)* la cual debe estar instalada en el dispositivo, esta aplicación nos permite escanear todo tipo de códigos de barras *(Figura 3.1.8)* devolviendo el número que va situado debajo de este, si Barcode Scanner no estuviera instalada, nos llevará directamente a la Google Play Store para que la instalemos. Una vez iniciada la aplicación Barcode Scanner se nos presenta la pantalla para realizar el escaneo de los códigos de barras de los productos.

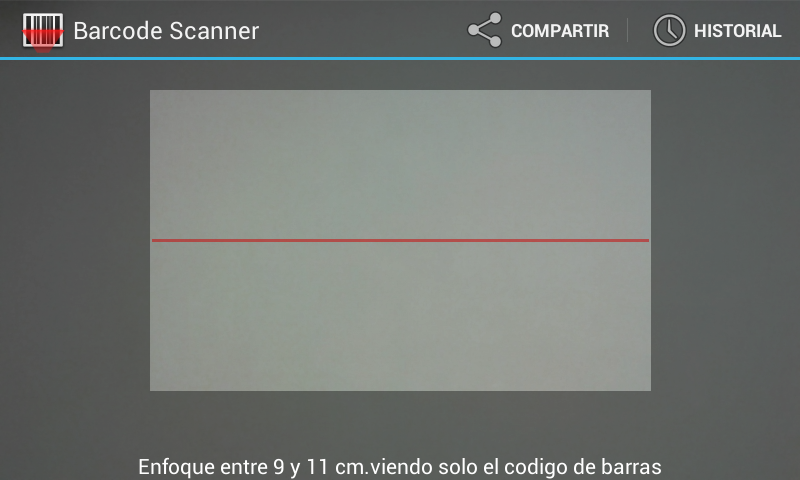


Figura 3.1.7 - Barcode Scanner

|  |  |
| --- | --- |
| D:\proyecto\codigos de barras\bifrutas pascual 6 uds 2.56.gif | 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 activity Carrito *(Figura 3.1.9)* en la cual se visualizará el producto escaneado y se sumará su precio al cálculo total.

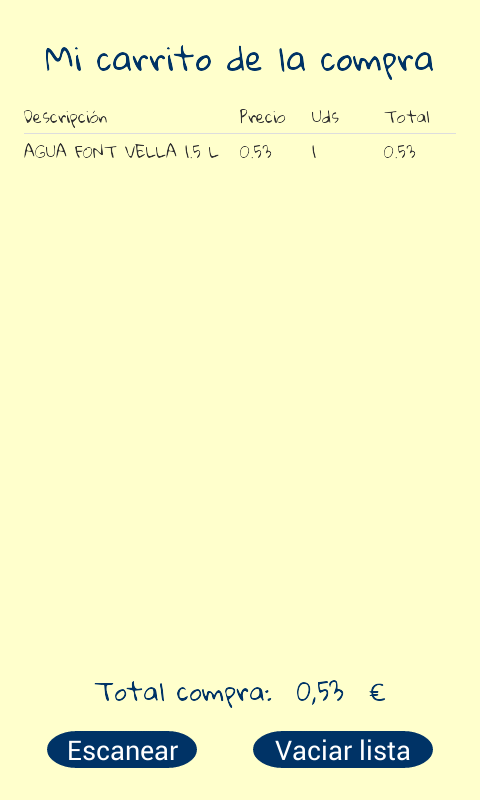


Figura 3.1.9 - Activity Carrito

Ahora en esta activity podemos seleccionar, pulsando sobre el, cualquiera de los productos que se encuentran en la lista lo cual nos llevará a una nueva activity Mostrar Producto *(Figura 3.1.10)* en la cual se nos muestra 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 activity Carrito, si pulsamos el botón Volver, guardaremos los cambios en la cantidad del producto y volveremos a la activity Carrito la cual quedará actualizada.

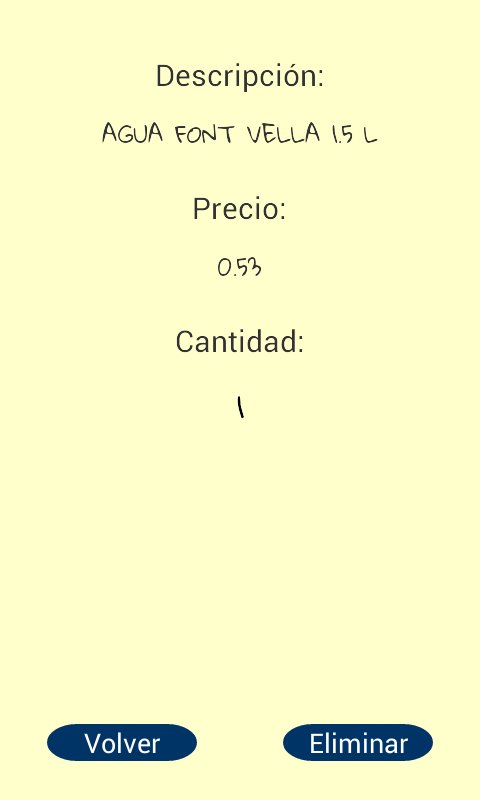


Figura 3.1.10 - Mostrar Producto

**3.2. Implementación**

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 y explicaré de maneras justificadas aquellas que son herramientas externas o que he creído más interesantes profundizar un poco más.

**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. Activities y XMLs**

Para empezar a desarrollar una aplicación debemos entender que es un *Activity*. Un *Activity* es una clase pública en la que el usuario puede desarrollar el código que desee. Este tipo de clases vienen heredadas de la clase base “*android.app.Activity*”.

Habitualmente las aplicaciones se componen de activities. Cuando se solicita un *Activity*, esta pasa al primer plano colocándose encima de otras, formado así una pila de activities. El botón del dispositivo “Back” cierra el *Activity* actual y recupera el primero de la pila.

En Android una aplicación no tiene control de ciclo de vida por lo que debemos poder finalizarlas en cualquier momento. Cada programa ejecuta su propio proceso.

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

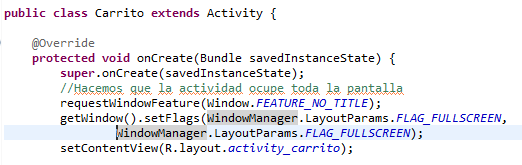
El *runtime7* de Android se encarga de gestionar una aplicación, y por tanto las activities que la forman.

La actividad o *Activity* representa una acción que puede realizar el usuario. Se corresponden en muchos casos con al UI (Interfaz de usuario). Muestran los *widgets8* dispuestos y gestiona las interacciones del usuario con estos. Toda actividad debe estar declarada en el AndroidManifest.xml (es un fichero el cual se deben declarar todos activities y todos aquellos permisos adicionales (Internet, Cámara, etc.) que el sistema necesite para que este lo lea antes de ser ejecutada).

Para comunicar activities entre ellas Android nos proporciona ciertos métodos. Para lanzarlos primero debemos declararlos dentro de un *Intent* que es una clase en la que permite especificarlo.

* **startActivity(intent):** lanza el *Activity* sin esperar resultados.
* **startActivityForResult(intent, requestCode):** lanza el *Activity* de la cual esperamos un resultado. Cuando el *Activity* llamado finaliza, se invoca el método onActivityResult() pasándole el *requestCode* con el que se lanzó el *Activity*.

Para crear un *Activity* seria de la siguiente manera:



*7 Intervalo de tiempo en el que un programa se ejecuta en un sistema operativo.*

*8 pequeñas aplicaciones informáticas en forma de iconos gráficos que se utilizan en páginas web para permitir la interactuación con el usuario o visitante*.

* **onCreate:** llamamos a ese método al crear el *Activity.*
* **setContentView:** asigna a la vista el contenido del layout9.
* **R.layout.screen:** definición de la vista de la aplicación.
* **Bundle savedInstanceState:** contiene el estado de la actividad anterior que ha sido suspendida.

Durante la vida de un *Activity* esta pasa por una serie de estados. La clase Activity nos permite redefinir ciertos métodos mediante *@Override*, en sus clases derivadas que incluyen el código a ejecutar en las transiciones de estados.



Figura 3.2.1 – Estados de un Activity

*9 Pagina de diseño.*

Estados de un *Activity*:

* **Activo (Running):** el *Activity* está en la cima de la pila, es visible y tiene el foco.
* **Pausado (Paused):** el *Activity* es visible pero no tiene el foco. Llegamos a este estado lanzando activities transparentes o que no ocupan toda la pantalla.
* **Parado (Stopped):** cuando el *Activity* no es visible. Se recomienda guardar el estado de la UI.
* **Destruido (Destroyed):** cuando el *Activity* termina, o es finalizada por Android. Sale de la pila.

Métodos de transición de estados:

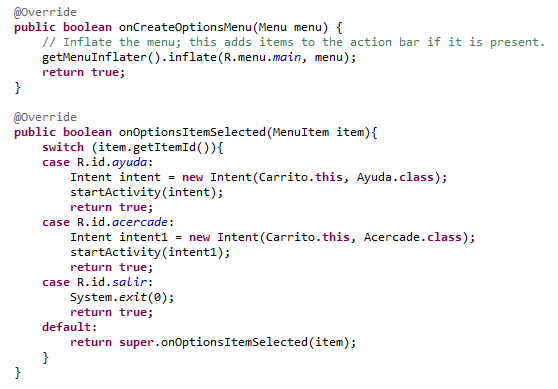
* **onCreate(Bundle):** se invoca cuando el *Activity* arranca por primera vez. Se utiliza para tareas de inicialización. Su parámetro es “*null*” o bien un estado guardado con anterioridad.
* **onStart():** se invoca cuando un *Activity* va a ser mostrado al usuario.
* **onResume():** se invoca cuando 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. Si hay escasez de recursos en el sistema, este método podría llegar a ser ignorado destruyendo el *Activity* directamente.
* **onRestart():** se invoca cuando un *Activity* parado pasa a estar activo.
* **onDestroy():** se invoca cuando un *Activity* va a ser destruido. De igual forma que con onStop(), en caso de escasez de recursos podría llegar a ser ignorado, destruyendo el *Activity* directamente.
* **onSaveInstanceState(Bundle):** se invoca para permitir al *Activity* guardar su estado. Normalmente no necesita ser redefinido.
* **onRestoreInstaceState(Bundle):** Se invoca para recuperar el estado guardado por onSaveInstaceState(). Normalmente no necesita ser redefinido.

Las *layouts* en XML’s son aquellos ficheros llamados después de la creación de la clase para mostrar la interfaz de usuario creada. Cada pantalla se implemente como un *Activity* diferente. Estos diseños se guardan en la carpeta “*res/layout*” del programa y para llamarlos se utiliza la clase R (es una clase autogenerada al compilar el código que reúne todos los identificadores de los widgets declarados). La función que se ejecuta con *setContentView(R.layout.pantalla).*

**3.2.2. El menú de la aplicación**

La actividad principal de la aplicación tiene definido un menú, para ello lo primero que hay que hacer es implementar el evento onCreateOptionsMenu(). En este evento deberemos “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. 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() de la actividad que muestra el menú. Este evento recibe como parámetro el ítem de menú que ha sido pulsado por el usuario, cuyo ID podemos recuperar con el método getItemId(). Según este ID podremos saber qué opción ha sido pulsada y ejecutar unas acciones u otras.



**3.2.3. 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. A pesar de que este SO provee de varios métodos para guardar datos persistentes de tu aplicación, he decidido utilizar el soporte que ofrece esta plataforma sobre una base de datos llamada *SQLite*. Esta librería es *OpenSource10* y proporciona capacidades de una base de datos relacional.

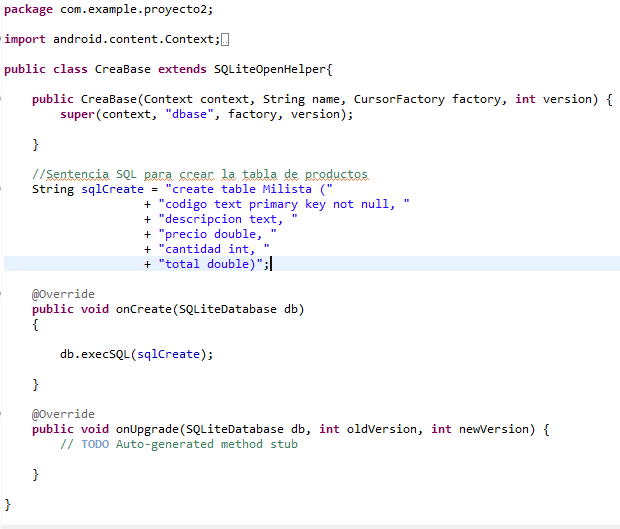
Me he decidido a usar *SQLite* y no otros métodos que ofrece Android porque es una base de datos muy extendida en el mundo y otras plataformas, como por ejemplo OSX, también la utilizan en sus dispositivos móviles. Además he creído conveniente compartir las mismas estructuras de datos tanto en el servidor como en el cliente.

10 Programación Libre (gratuita)

Esta BD (Base de Datos) se almacena en la carpeta del proyecto “*/data/data/<package-name>/databases*”. Por defecto son privadas a la aplicación pero por sentencias SQL se puede acceder a ella. Para compartir datos entre aplicaciones con este otras aplicaciones se usan los *Content Providers* (clases que permiten comunicar datos).

Para acceder con estas sentencias SQL a la base de datos trabajamos con cursores. Estos cursores es muy recomendables cerrarlos (con la sentencia *close()*) una vez realizada la consulta para liberar los recursos.

En la siguiente imagen veremos la acción para crear la base de datos SQLite utilizada en la aplicación.



Para insertar o actualizar una fila de la tabla he usado las funciones estándar *insert* o *update* que tiene la clase *Database* de Android. Esta clase también contiene la función *delete* para eliminar filas.

Para crear la base de datos (Anexo B) en mi servidor he necesitado usar la herramienta de *phpmyadmin* incluida dentro del paquete XAMPP(explicado anteriormente).

**3.2.4. El servidor**

El servidor de este proyecto está compuesto por el 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();

}

?>

Este PHP 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.

**3.2.5. 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.



**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 asi 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).*

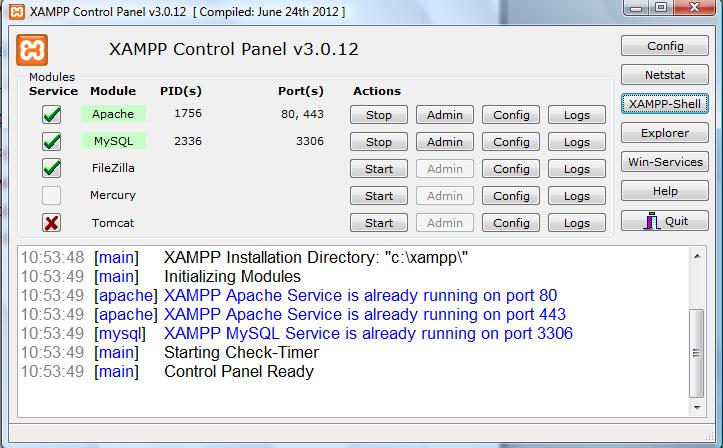


Figura 6.1 – XAMPP Control Panel

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

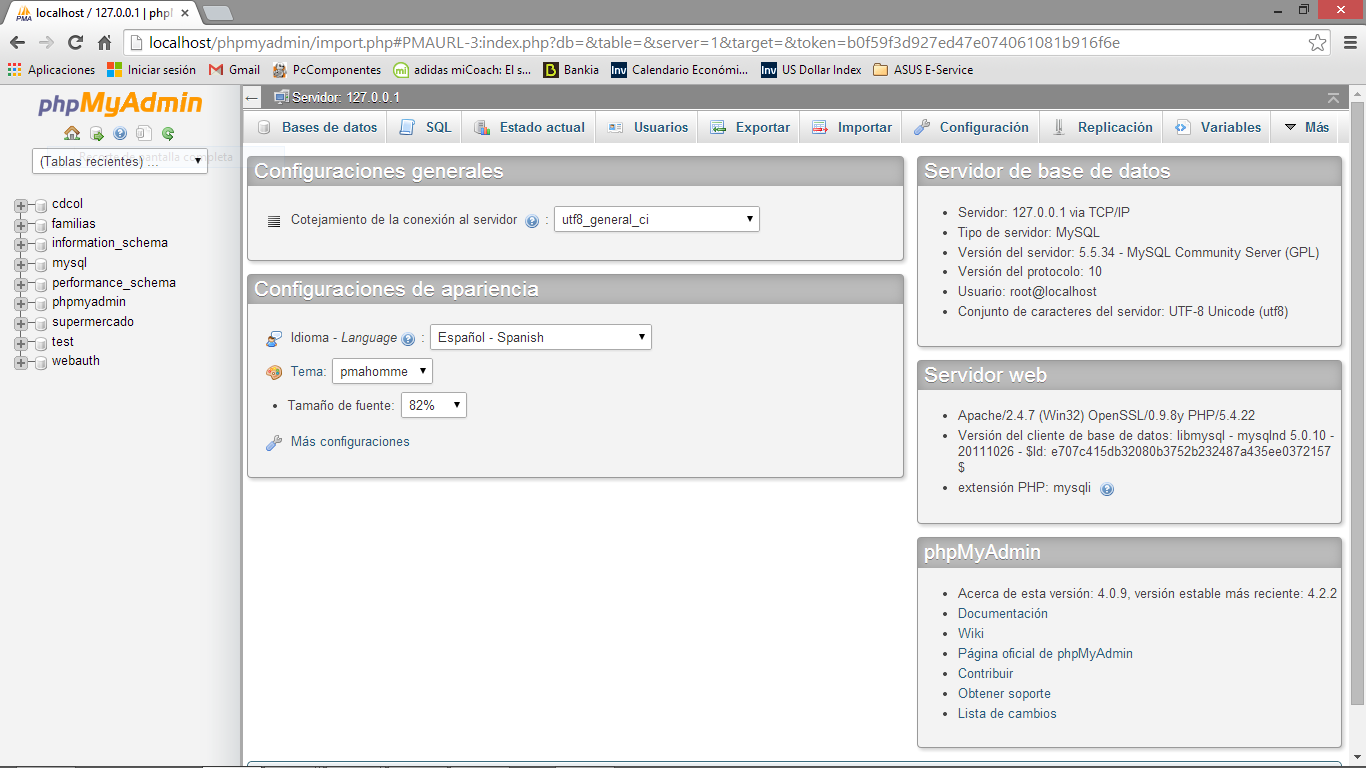


Figura 6.2 – PHPMyAdmin

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 activity *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 activity 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);

**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/ programacion/*

**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/*