

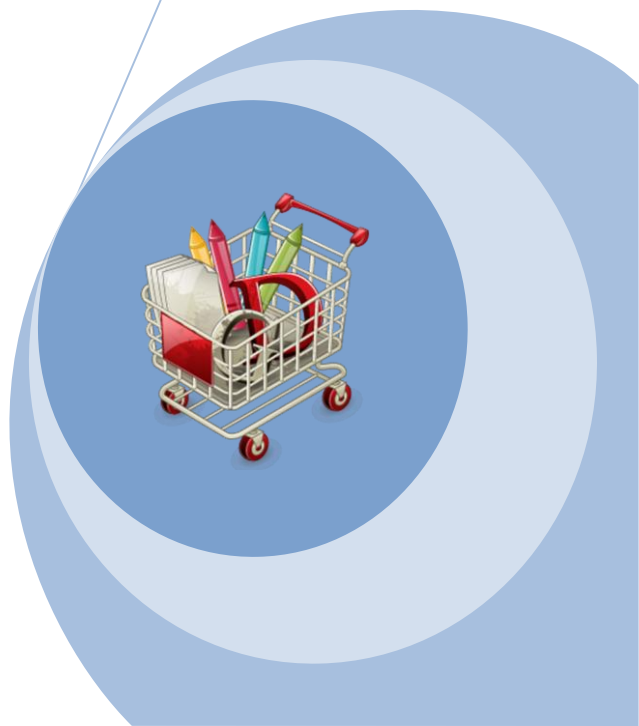


# Aplicación Android: Lista de compra de los productos más baratos.

TFC – Entrega Final.

Listas de compra personales con los precios más baratos  
de los supermercados.

**Autor: Fernando Loza Escudero**  
**Consultor: Roberto Ramírez Vique**  
**07/01/2014**





## Contenido:

<b>Definición del proyecto.</b> .....	3
<b>Descripción del proyecto.</b> .....	3
<b>Justificación del proyecto.</b> .....	3
<b>Recursos hardware.</b> .....	3
<b>Recursos software.</b> .....	4
<b>Versiones y compatibilidades.</b> .....	4
<b>Objetivo del proyecto.</b> .....	4
<b>Listado de funcionalidades.</b> .....	5
<b>Calendario del proyecto.</b> .....	5
<b>Tecnología escogida.</b> .....	6
<b>Riesgos del proyecto.</b> .....	6
<b>Análisis de requisitos.</b> .....	7
<b>Funciones del producto.</b> .....	7
<b>Tipos de usuarios.</b> .....	7
<b>Restricciones.</b> .....	7
<b>Requisitos específicos.</b> .....	8
<b>Caso de usos.</b> .....	8
<b>Diseño de la interfaz de usuario: prototipo.</b> .....	15
<b>Pantalla inicial.</b> .....	15
<b>Pantalla de lista creada.</b> .....	16
<b>Pantalla de lista de productos de una lista de compra.</b> .....	16
<b>Pantalla de añadir productos.</b> .....	17
<b>Pantalla de MAPA (producto seleccionado).</b> .....	17
<b>Diseño del proyecto</b> .....	18
<b>Modelo de datos.</b> .....	18
<b>Análisis de los tipos de entidad.</b> .....	20
<b>Análisis de los tipos de interrelación.</b> .....	21
<b>Esquemas entidad – relación.</b> .....	22
<b>Arquitectura de la aplicación</b> .....	23
<b>Diseño de la interfaz</b> .....	24
<b>Inicio de la aplicación.</b> .....	25
<b>Configurar el acceso a la Base de Datos externa.</b> .....	26
<b>Inserción de una nueva lista.</b> .....	26



<b>Consulta de una lista. ....</b>	<b>27</b>
<b>Borrar una lista.....</b>	<b>28</b>
<b>Agregar un producto nuevo a una lista.....</b>	<b>28</b>
<b>Consultar un producto de la lista.....</b>	<b>29</b>
<b>Enviar precio a la Base de Datos externa. ....</b>	<b>30</b>
<b>Localizar en el Mapa los dos productos más baratos con respecto a la ubicación del usuario. ....</b>	<b>30</b>
<b>Salir de la aplicación. ....</b>	<b>31</b>
<b>Instrucciones de instalación ....</b>	<b>31</b>
<b>Aplicación interna. ....</b>	<b>31</b>
<b>Aplicación externa.....</b>	<b>35</b>
<b>Pruebas realizadas ....</b>	<b>37</b>
<b>Conclusiones y Mejoras.....</b>	<b>37</b>
<b>Imágenes utilizadas.....</b>	<b>38</b>
<b>Bibliografía.....</b>	<b>39</b>
<b>Consultas bibliográficas.....</b>	<b>39</b>
<b>Consultas Web.....</b>	<b>39</b>



## Definición del proyecto.

### Descripción del proyecto.

El siguiente proyecto se denominará <<Cheaplist (ChLs)>>.

En los tiempos que corren, actualmente, la crisis económica está azotando a miles de familias que se ven obligadas a escatimar, en precios, los productos de la lista de la compra.

Esta aplicación es personal, en el sentido de que cada usuario puede crear y consultar su propia lista de la compra.

A su vez, mediante geolocalización, el usuario puede contrastar, a partir de su ubicación, los dos supermercados donde se encuentran los productos más baratos que desea introducir en su lista.

### Justificación del proyecto.

La justificación de realizar el proyecto, en cliente, mediante Android, viene justificada por una serie de ventajas:

- Gratuidad. El desarrollo en esta plataforma es completamente gratuito, basta con instalar el ADT SDK de eclipse, con emulador de dispositivo incluido y empezar a desarrollar.
- Alta usabilidad de reutilización de código, característica propia de aplicaciones de código abierto. Esto conlleva una valiosa eficacia y un alto ahorro de tiempo, dando lugar a proyectos más creativos y eficientes.
- La calidad de los gráficos, en Android, tanto 2D como 3D, es muy alta y atractiva, lo que permite la creación de aplicaciones y juegos muy atractivos, tanto para usuarios, como para empresas que desean explotar este sistema.
- La posibilidad de publicar una aplicación en “Play Store” (<https://play.google.com/store?hl=es>) es bastante sencilla, lo que permite a un abanico muy variado de usuarios, tanto la publicación como la descarga y uso de las mismas, así como la valoración de la comunidad sobre la aplicación y un rápido acceso a las actualizaciones de las mismas.
- El usos del API de Google MAPS, añade, si cabe, todavía más ventajas:
  - Fácil y rápida integración de mapas y geolocalización en nuestras aplicaciones.
  - Fiabilidad y rápida respuesta en cualquier caso.
  - Márquetin asegurado, por asociar nuestra aplicación a una empresa, Google, mundialmente reconocida, en cuanto a calidad y fama.
  - Constante actualización de esta tecnología, de forma transparente a la aplicación.

### Recursos hardware.

El material hardware del que se dispone es el siguiente:

- Portátil TOSHIBA, INTEL® Core™2 Duo CPU T8300 @ 2.40GHz 2.40 GHz con SO Windows Vista.
- PC Sobremesa Pentium® Dual-Core CPU E5700 @ 3.00GHZ 3.00 GHz con SO Windows 7



- Smartphone Sony Ericsson XPERIA-ARC.

## Recursos software.

El material software que se utilizará para el desarrollo de la aplicación es el siguiente:

- `product com.android.ide.eclipse.adt.package.product`
- Android Developer Tools Build: v21.0.1-543035
- IIS
- Visual Web developer:  
ASP.NET 3.5 + Visual Basic. Net → Servidors con web services.
- SQLite
- Paquete con la APIs de Google Maps correspondiente al anterior SDK.
- Gantt Project 2.5.5. **GanttProject 2.6.1 Brno (build 1499)** (C) Equipo de GanttProject 2012. Licenciado bajo General Public License (GPL) versión 3
- Dia Diagram Editor V 0.9.72
- MagicDraw 17.0.4 UML
- Prototyper Free 5.5.0
- Microsoft Office 2007 y 2013: Word, Access.
- Camtasia Studio 8 (version de evaluación)

## Versiones y compatibilidades.

La aplicación será compatible con la versión de Android 4.3 y anteriores. Por otro lado. Al tratarse de una aplicación que utiliza geolocalización hay que tener en cuenta la funcionalidad de GPS del terminal. Si bien no es imprescindible, pues en caso de no disponer de ella podrían usarse otros proveedores como localización mediante la red de telefonía.

## Objetivo del proyecto.

Aplicación Android que permite al usuario crear una lista de la compra con la originalidad de que los productos elegidos son los más baratos de entre los existentes en los supermercados cercanos a la ubicación del usuario.

Así pues el usuario, desde el móvil puede: consultar una lista de la compra de entre un listado de listas de la compra que tiene guardada en local, en su terminal. Dichas listas las crea el usuario con los productos que desea añadir a los mismos. Cuando consulta una de sus listas y selecciona de ella un producto tiene la facilidad de solicitar que se muestre en un mapa los dos supermercados que tienen dicho producto más barato.

Una segunda función de la aplicación en el dispositivo móvil sería la posibilidad de escanear el precio de un producto, enviando a un servidor externo precio y localización del supermercado, incorporándose a la base de datos por orden de precio de menor a mayor. Esta funcionalidad podría establecerse solo para usuarios administradores de la empresa



que explote la aplicación o bien por cualquier usuario, que introduzca para beneficio del resto de la comunidad precios mediante escaneo. En cualquier caso, el proyecto, inicialmente y por motivos de tiempo, solo tendrá la funcionalidad móvil de la geolocalización y la posibilidad de enviar a la BD externa, mediante formulario un producto y precio y supermercado geolocalizado.

## Listado de funcionalidades.

Listado de listas de la compra del usuario, en local.

Seleccionar lista de la compra para consultar.

Crear lista de la compra.

Borrar lista de la compra.

Añadir productos a la lista.

Borrar productos de la lista.

Seleccionar ubicación del usuario

Seleccionar y localizar centro de menor precio.

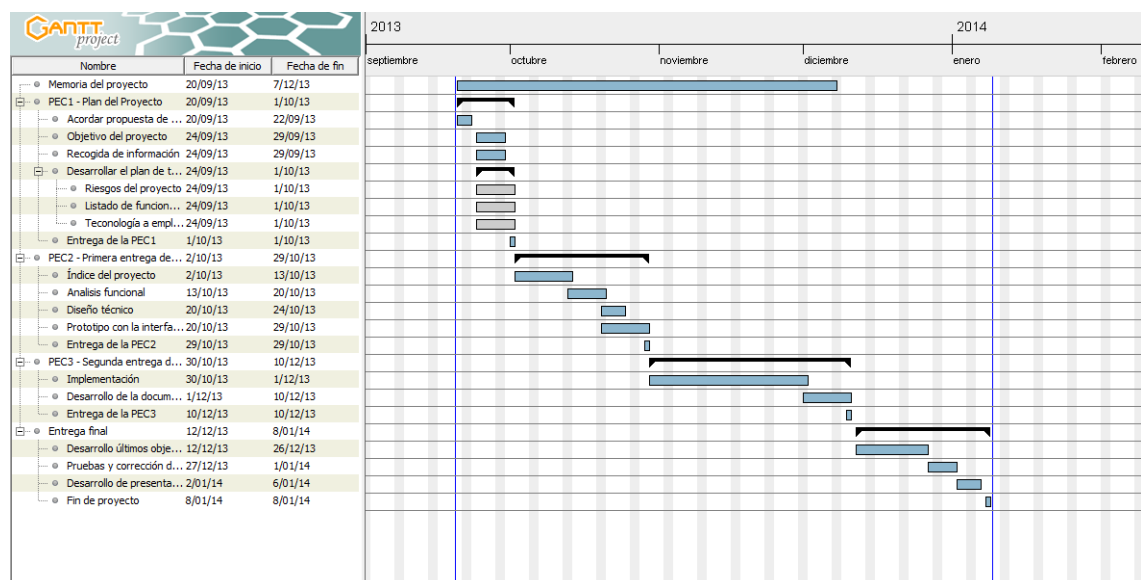
Escanear precio de producto en supermercado. *(En principio no se implementará)*

Enviar precio y localización a BD centralizada.

## Calendario del proyecto.

El calendario inicial viene dado por la imagen siguiente. Evidentemente, será el avance del proyecto el que pueda variar tanto la distribución en el tiempo como los apartados recogidos en el mismo.

(modificada planificación con respecto a la PEC 1)





## Tecnología escogida.

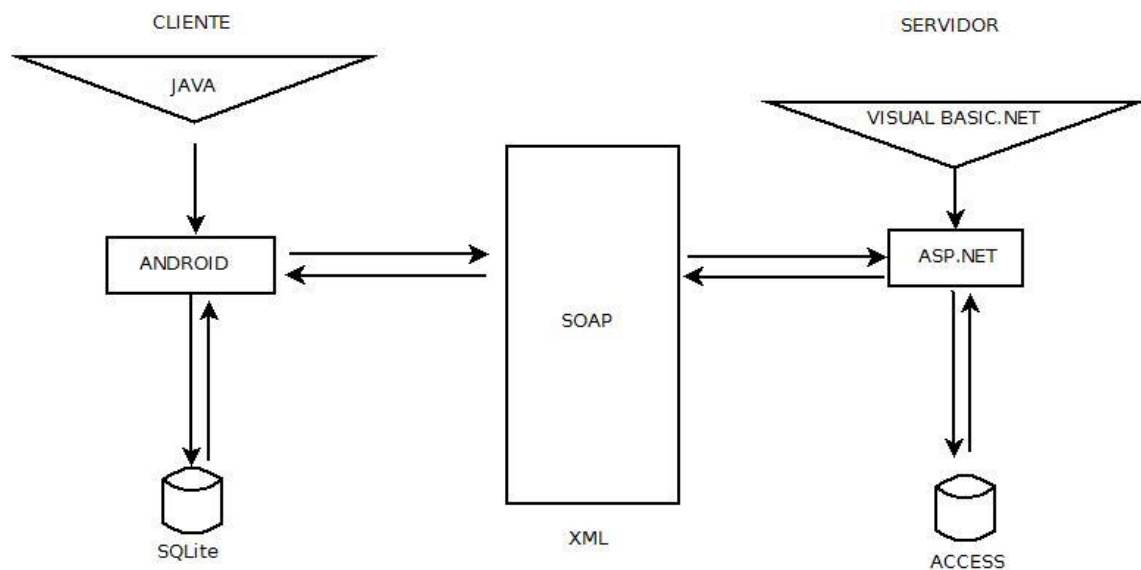
Para la realización del proyecto se opta por un esquema cliente servidor externo, de tal forma que en el cliente tendríamos lo que sería la aplicación Android, pensada, por su carácter de movilidad, para dispositivos Smartphone, aunque igualmente válido para Tablet. Y en el servidor tendremos los datos, la información.

En el servidor se ofrecen web services, que serán consumidos por la aplicación Android. Para ello se usará ASP.NET + Visual Basic. En mi propio pc de sobremesa y SOAP como protocolo de intercambio de datos XML con el cliente.

Tanto en el cliente como en el servidor dispondremos de una base de datos.

En el caso del servidor, la base de datos, será Access, por motivos de facilidad, en mi caso en particular, para el desarrollo e integración con ASP.NET, aunque podríamos haber usado cualquier otra más potente y conocida: SQL Server, MySQL, PostgreSQL,...

En cuanto al cliente, dispondrá, asociada a la aplicación, por su poco peso y su facilidad de integración SQLite. Ésta servirá para guardar las listas personales de la compra.



## Riesgos del proyecto.

- Los riesgos del proyecto vienen definidos por la forma en que se mantengan los datos del servidor. Si hablamos de una comunidad de usuarios que van actualizando los datos mediante escaneo de códigos de barras y coordenadas del supermercado, aunque la aplicación será más dinámica, se corre el riesgo de que los datos introducidos por el usuario no sean del todo fiable.



Pero si hablamos de una empresa que se dedique al escaneo de precios y mantengan el servidor, en ese sentido se ganará en fiabilidad en las consultas de las listas de compra de los usuarios.

- Por otro lado, en cuanto a lo que el desarrollo del proyecto y la consecución de los objetivos marcados se refiere son diversos e importantes riesgos los que se ponen en juego:
  - Mi conocimiento de la tecnología no tiene la experiencia ni la profundidad necesarias para poder dar por conseguidos, a priori, los objetivos planteados.
  - El tiempo de dedicación que requiere el proyecto es bastante incompatible con el tiempo del que dispongo, es por ello que el objetivo básico que me marco es implementar, al menos, las funciones más básicas que hagan funcionar la aplicación como una demo básica del proyecto real.
  - En cuanto al consumo de servicios webs, estoy barajando la posibilidad entre tres tipos: SOAP, Rest o Json. Finalmente, el tiempo y la sencillez harán que me decante por uno u otro método, en este caso SOAP y xml.

## Análisis de requisitos.

### Funciones del producto.

- Crear Lista de la compra nueva
- Seleccionar lista de la compra ya creada.
- Crear nuevo producto en lista de la compra
- Seleccionar producto
- Localizar en MAPA Supermercados con productos más baratos.
- Crear supermercado – coordenadas – producto en servidor externo.

### Tipos de usuarios.

El usuario de esta aplicación es cualquiera que disponga de un móvil. Puede tratarse de un adulto, una persona mayor o un niño, por lo tanto la aplicación se presenta como breve, visual y sencilla de usar, sea cual sea la edad del usuario.

El único requisito a tener en cuenta es que el usuario debe de disponer de un dispositivo Android: ya sea Smartphone o Tablet.

### Restricciones.

La principal restricción de la aplicación viene determinada por el tipo de información disponible en la base de datos externa.

Vamos a suponer (aunque la aplicación pueda tener una segunda versión en la que los propios usuarios mantengan la información externa mediante escaneo del producto in situ) que hay una







<b>Caso de uso</b>	<b>Crear nueva lista</b>
Actor	Usuario final de la aplicación
Descripción	El usuario final al iniciar se encuentra con un listado de listas, si las hubiere. En esta pantalla inicial podrá: <ul style="list-style-type: none"><li>- Seleccionar una de las listas de compra.</li><li>- <b>Crear una lista nueva.</b></li></ul>
Precondiciones	<ul style="list-style-type: none"><li>- La aplicación debe de funcionar correctamente.</li><li>- La aplicación debe de mostrar la opción de crear una lista nueva.</li></ul>
Escenario	<ul style="list-style-type: none"><li>- La aplicación se encuentra en funcionamiento.</li><li>- El usuario decide crear una lista nueva.</li></ul>
Postcondiciones	Lista creada

<b>Caso de uso</b>	<b>Seleccionar lista</b>
Actor	Usuario final de la aplicación
Descripción	El usuario final al iniciar se encuentra con un listado de listas, si las hubiere. En esta pantalla inicial podrá: <ul style="list-style-type: none"><li>- <b>Seleccionar una de las listas de compra.</b></li><li>- Crear una lista nueva.</li></ul>
Precondiciones	<ul style="list-style-type: none"><li>- La aplicación se encuentra en funcionamiento.</li><li>- La aplicación ofrece un listado de listas de compras y son seleccionables. Debe de haber alguna lista creada.</li></ul>
Escenario	<ul style="list-style-type: none"><li>- La aplicación se encuentra en funcionamiento.</li><li>- El usuario ha iniciado la aplicación y ésta ofrece el listado de lista de compras porque existen registro de listas. En caso contrario le muestra que no hay listas creadas.</li></ul>
Postcondiciones	Lista seleccionada



Caso de uso	<b>Consultar lista de compra.</b>
Actor	Usuario final de la aplicación
Descripción	El usuario final ha seleccionado una de las listas de compras del listado, la cual desea consultar.
Precondiciones	<ul style="list-style-type: none"><li>- La aplicación se encuentra en funcionamiento.</li><li>- El usuario ha decidido seleccionar una lista, la cual quiere consultar. Debe de haber algún registro (lista) creado.</li></ul>
Escenario	<ul style="list-style-type: none"><li>- La aplicación se encuentra en funcionamiento.</li><li>- El usuario ha decidido seleccionar una lista, la cual quiere consultar.</li></ul>
Postcondiciones	Listado de listas de la compra

Caso de uso	<b>Consultar lista de productos.</b>
Actor	Usuario final de la aplicación
Descripción	El usuario final se encuentra con un listado de productos, si los hubiere. En esta pantalla inicial podrá: <ul style="list-style-type: none"><li>- Seleccionar una de los productos.</li><li>- Añadir un producto nuevo a la lista.</li></ul>
Precondiciones	<ul style="list-style-type: none"><li>- La aplicación se encuentra en funcionamiento.</li><li>- La aplicación ofrece un listado de productos y son seleccionables. Debe de haber algún producto introducido en la lista de compra.</li></ul>
Escenario	<ul style="list-style-type: none"><li>- La aplicación se encuentra en funcionamiento.</li><li>- El usuario ha seleccionado una lista de comprar y ésta ofrece el listado de lista de productos de dicha lista porque existen registros. En caso contrario le muestra que no hay productos.</li></ul>
Postcondiciones	Listado de productos de la lista



Caso de uso	<b>Crear nuevo producto.</b>
Actor	Usuario final de la aplicación
Descripción	El usuario final al iniciar se encuentra con un listado de productos, si los hubiere. En esta pantalla inicial podrá: <ul style="list-style-type: none"><li>- Seleccionar una de los productos.</li><li>- <b>Añadir un producto nuevo a la lista.</b></li></ul>
Precondiciones	<ul style="list-style-type: none"><li>- La aplicación se encuentra en funcionamiento.</li><li>- La aplicación ofrece la posibilidad de añadir un nuevo producto a la lista.</li></ul>
Escenario	<ul style="list-style-type: none"><li>- La aplicación se encuentra en funcionamiento.</li><li>- El usuario ha seleccionado añadir un nuevo producto.</li></ul>
Postcondiciones	Producto creado

Caso de uso	<b>Consultar ubicación productos más baratos.</b>
Actor	Usuario final de la aplicación
Descripción	El usuario final se encuentra con un listado de productos, si los hubiere. En esta pantalla inicial podrá: <ul style="list-style-type: none"><li>- <b>Seleccionar una de los productos.</b></li><li>- Añadir un producto nuevo a la lista.</li></ul>
Precondiciones	<ul style="list-style-type: none"><li>- La aplicación se encuentra en funcionamiento.</li><li>- La aplicación ofrece un listado de productos y son seleccionables. Debe de haber algún producto introducido en la lista de compra.</li></ul>
Escenario	<ul style="list-style-type: none"><li>- La aplicación se encuentra en funcionamiento.</li><li>- El usuario ha seleccionado un producto de la lista de compra y esta acción le lleva a un mapa donde se geolocaliza establecimientos cercanos a la ubicación del usuario con los dos precios más baratos para ese producto</li></ul>
Postcondiciones	Listado de los dos productos mas baratos



Caso de uso	<b>Enviar datos base de datos externa</b>
Actor	Usuario final de la aplicación
Descripción	El usuario final desea enviar a la base de datos externa un producto con un precio que considera más barato y el supermercado y localización del mismo donde se haya
Precondiciones	<ul style="list-style-type: none"> <li>- El usuario ha seleccionado un producto y desea enviar sus datos a la BD</li> <li>- La aplicación ha geolocalizado al usuario y dispone del nombre del producto</li> <li>- El usuario debe rellenar el nombre del supermercado y su dirección si la conoce</li> <li>- La aplicación le da la opción enviar</li> </ul>
Escenario	<ul style="list-style-type: none"> <li>- La aplicación se encuentra en funcionamiento.</li> <li>- El usuario ha seleccionado un producto de la lista de compra y esta acción le lleva a una pantalla donde puede solicitar su envío a la BD externa</li> </ul>
Postcondiciones	Producto enviado, confirmado por el servidor

Caso de uso	<b>Producto</b>
Actor	Usuario final de la aplicación
Descripción	<ul style="list-style-type: none"> <li>- El usuario selecciona un producto y opta por enviar producto a la BD y la aplicación rellena el producto seleccionado</li> </ul>
Precondiciones	<ul style="list-style-type: none"> <li>- Se ha seleccionado un producto, previamente</li> </ul>
Escenario	<ul style="list-style-type: none"> <li>- Seleccionado el producto se pulsa enviar a BD</li> </ul>
Postcondiciones	Producto seleccionado



Caso de uso	<b>Precio</b>
Actor	Usuario final de la aplicación
Descripción	<ul style="list-style-type: none"><li>- El usuario selecciona un producto y opta por enviar producto a la BD y rellena su precio en el formulario</li></ul>
Precondiciones	<ul style="list-style-type: none"><li>- Se ha seleccionado un producto, previamente</li></ul>
Escenario	<ul style="list-style-type: none"><li>- El usuario selecciona un producto y opta por enviar producto a la BD y la aplicación rellena el producto seleccionado. Rellena su precio en el formulario</li></ul>
Postcondiciones	Campo de precio relleno

Caso de uso	<b>Coordenadas Supermercado</b>
Actor	Usuario final de la aplicación
Descripción	<ul style="list-style-type: none"><li>- El usuario selecciona un producto y opta por enviar producto a la BD y la aplicación rellena las coordenadas geolocalizadas del dispositivo del usuario.</li></ul>
Precondiciones	<ul style="list-style-type: none"><li>- Se ha seleccionado un producto, previamente</li></ul>
Escenario	<ul style="list-style-type: none"><li>- El usuario selecciona un producto y opta por enviar producto a la BD y la aplicación rellena las coordenadas geolocalizadas del dispositivo del usuario.</li></ul>
Postcondiciones	Campo de Latitud y longitud relleno

Caso de uso	<b>Nombre Supermercado</b>
Actor	Usuario final de la aplicación
Descripción	<ul style="list-style-type: none"><li>- El usuario selecciona un producto y opta por enviar producto a la BD. Rellena el nombre del supermercado.</li></ul>
Precondiciones	<ul style="list-style-type: none"><li>- Se ha seleccionado un producto, previamente</li></ul>
Escenario	<ul style="list-style-type: none"><li>- El usuario selecciona un producto y opta por enviar producto a la BD. Rellena el nombre del supermercado.</li></ul>
Postcondiciones	Campo nombre del supermercado relleno



Caso de uso	<b>Dirección Supermercado</b>
Actor	Usuario final de la aplicación
Descripción	<ul style="list-style-type: none"><li>- El usuario selecciona un producto y opta por enviar producto a la BD. Rellena la dirección del supermercado.</li></ul>
Precondiciones	<ul style="list-style-type: none"><li>- Se ha seleccionado un producto, previamente</li></ul>
Escenario	<ul style="list-style-type: none"><li>- El usuario selecciona un producto y opta por enviar producto a la BD. Rellena la dirección del supermercado.</li></ul>
Postcondiciones	Campo de la dirección del supermercado rellena

Caso de uso	<b>Crear nuevo producto</b>
Actor	Administrador Servidor Externo
Descripción	<ul style="list-style-type: none"><li>- El administrador registra productos y el establecimiento en el que se localizan y precios.</li></ul>
Precondiciones	<ul style="list-style-type: none"><li>- Debe de recorrer establecimiento y escanear precios.</li></ul>
Escenario	<ul style="list-style-type: none"><li>- El administrador dispone de nuevos registros y los inserta en la Base de Datos del Servidor Externo.</li></ul>
Postcondiciones	Producto nuevo creado

Caso de uso	<b>Crear nuevo supermercado</b>
Actor	Administrador Servidor Externo
Descripción	<ul style="list-style-type: none"><li>- El administrador registra un Supermercado nuevo.</li></ul>
Precondiciones	<ul style="list-style-type: none"><li>- Debe de darse un nuevo establecimiento que registrar.</li></ul>
Escenario	<ul style="list-style-type: none"><li>- El administrador detecta un nuevo establecimiento y lo registra.</li></ul>
Postcondiciones	Supermercado nuevo creado con sus coordenadas



Caso de uso	<b>Crear coordenadas supermercado</b>
Actor	Administrador Servidor Externo
Descripción	<ul style="list-style-type: none"><li>• El administrador cuando registra un Supermercado registra sus coordenadas.</li></ul>
Precondiciones	<ul style="list-style-type: none"><li>• Debe de darse un nuevo establecimiento que registrar.</li></ul>
Escenario	<ul style="list-style-type: none"><li>• El administrador detecta un nuevo establecimiento y lo registra, así como sus coordenadas.</li></ul>
Postcondiciones	Coordenadas de supermercado insertadas

## Diseño de la interfaz de usuario: prototipo.

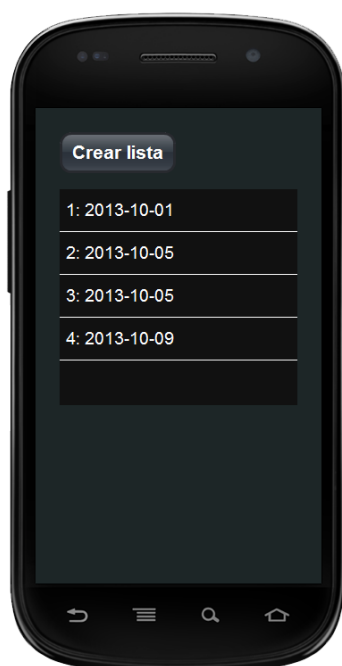
Para diseñar un prototipo, inicial, de la aplicación, se ha empleado el software: Prototyper Free 5.5.0. Se ha simulado un Smartphone de 320 X 480. Aunque la aplicación podría ser, igualmente válida, para otro dispositivo Android, como puede ser, por ejemplo una Tablet Pc.

Este prototipo es sobre el que se ha comenzado el proyecto, si bien, más adelante, se presenta el diseño final de la aplicación.

### **Pantalla inicial.**

Esta es la pantalla con la que arranca el usuario. Le muestra las listas de compra que tenga almacenadas en su dispositivo. Si no tuviera ninguna mostraría un mensaje indicándolo. Cada lista está indexada por su Id y por la fecha de creación de la misma. Son seleccionables.

Igualmente podremos crear una lista nueva, accediendo, simplemente, al botón <<Crear lista>>.

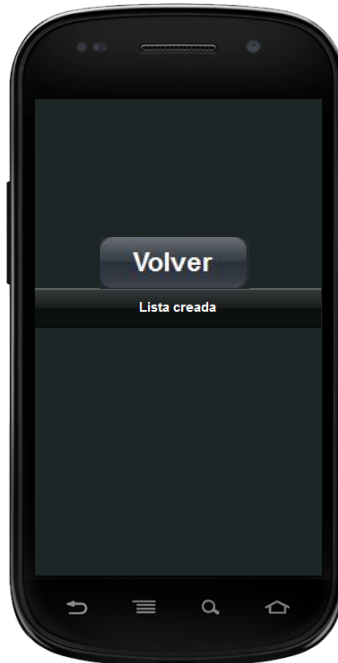






### Pantalla de lista creada.

La siguiente pantalla es la que se muestra cuando se selecciona crear lista. Una vez creada, podemos pulsar <<Volver>> para regresar a la pantalla donde se muestran todas las listas, entre ellas la nueva lista creada.



### Pantalla de lista de productos de una lista de compra.

Una vez que el usuario haya seleccionado una lista, se mostrará una pantalla donde le aparezcan los productos que contiene dicha lista y a los que puede ir agregando más productos.

Estos productos son seleccionables.





### Pantalla de añadir productos.

La siguiente pantalla es la que se muestra cuando hemos seleccionado <<Añadir producto>>. Podremos añadir un producto de la lista. Dicha lista debería ser la que nos envíe el servidor externo.



### Pantalla de MAPA (producto seleccionado).

La siguiente pantalla es la que se muestra al seleccionar un producto de una lista. Nos lleva a un Activity que localiza nuestra ubicación (punto azul) y nos muestra los establecimientos con los productos más baratos cercanos a nuestra ubicación.



## Diseño del proyecto

El objetivo de esta fase es realizar el diseño del sistema mediante técnicas de modelado que nos definirán el sistema a implementar.

### Modelo de datos.

Partimos del siguiente dominio de datos:

1. Por un lado tenemos los datos del servidor externo.
  - ✓ Provincias: Las provincias para el desarrollo de esta aplicación son las del Estado Español. Cada provincia es única e incluye el código territorial de la provincia.
  - ✓ Supermercados: Esta tabla está pensada para contener supermercados y sus coordenadas geográficas. Cada entrada es única. Para este proyecto y con el objeto de demostrar su funcionamiento solo se han tenido en cuenta dos marcas de supermercados: CARREFOUR, HIPERCOR y MERCADONA, 5 supermercados por provincia y solo se han tenido en cuenta dos provincias Sevilla y Barcelona.
  - ✓ Productos: Igualmente para los productos solo se ha tenido en cuenta una pequeña lista de ellos para demostrar el funcionamiento:
    - Aceite de oliva
    - Ajo
    - Bacalao
    - Helados



- Nueces
- Remolachas
- Tomates

Cada registro de producto lleva asociado el Id del supermercado en el que se encuentra y el precio de ese producto en ese supermercado.

Para limitar el número de sitios con los productos más baratos se ha limitado el número de ellos a dos. Por lo tanto la tabla de productos deberá incluir, al menos, dos productos con su establecimiento correspondiente.

Cualquiera de las tablas es ampliable a más datos. Por motivo de tiempo y con el objeto de demostrar el funcionamiento de la aplicación solo se han incluidos los datos mencionados.

2. Por otro lado tenemos los datos del servidor interno (de la aplicación en el dispositivo del usuario).

- ✓ Tenemos la tabla lists: en esta tabla es donde el cliente va introduciendo sus listas de la comprar. Cada vez que el cliente crea una nueva lista esta se guardará con un Id único y la fecha de introducción. De esta manera siempre sabrá el cliente sobre que lista de la compra opera.
- ✓ Products: esta tabla guardará los productos que el cliente introduzca en la lista escogida, de tal forma que cada producto insertado llevará asociado el Id de la lista de compra. De esta manera siempre que el cliente consulte los productos de la lista los hará de los añadidos a la lista seleccionada.
- ✓ La lista de productos posibles a añadir se ha incluido directamente en el fichero string.xml:

```
<string-array name="Istprod">  
  <item>Aceite de oliva</item>  
  <item>Ajo</item>  
  <item>Bacalao</item>  
  <item>Helados</item>  
  <item>Nueces</item>  
  <item>Remolachas</item>  
  <item>Tomates</item>  
  <item></item>  
</string-array>
```

- ✓ La lista de provincias posibles se ha incluido tambien directamente en el fichero provincia.xml:

Una posible mejora sería cargar la lista desde la base de datos externas, pero eso llevaría a una descarga pesada cada vez que se inicie la aplicación, cuando, de esta manera, una vez instalado



el programa ya tendría instalada la lista, sujeta, no obstante a futuras actualizaciones de la aplicación.

### Análisis de los tipos de entidad.

- Base de Datos externa.

- Tipo de entidad TBLProvincias.

Provincias con las que podemos trabajar en nuestro sistema, España. En el caso del código que se adjunta: Sevilla y Barcelona.

#### **Atributos:**

- ✓ CódigoProvincia: Código de la provincia.
- ✓ NombreProvincia: Nombre de la provincia.

- Tipo de entidad TBLSupermercados.

Supermercados con sus coordenadas geográficas y código de provincias, así como la dirección local, que vayamos a considerar en la aplicación. Por efectos de falta de tiempo solo voy a considerar dos marcas CARREFOUR y MERCADONA, dos provincias: Sevilla y Barcelona y 5 supermercados de cada marca.

#### **Atributos:**

- ✓ Id: Entero autonumérico.
- ✓ NombreSupermercado: Nombre del supermercado.
- ✓ Latitud: Coordenadas de la latitud del supermercado.
- ✓ Longitud: Coordenadas de la longitud del supermercado.
- ✓ CódigoProvincia: Código de la provincia.
- ✓ Dirección: Dirección local del supermercado.

- Tipo de entidad TBLProductos.

Productos y precios de los mismos, relacionado cada uno de ellos con el id del supermercado en el que se encuentra.

#### **Atributos:**

- ✓ IdProducto: entero autonumérico.
- ✓ Producto: Nombre del producto.
- ✓ Precio: precio del producto en el supermercado idSupermercado.
- ✓ IdSupermercado: id del supermercado para ese producto.



- Tipo de entidad TBLDominioProductos.

**Atributos:**

- ✓ Producto: producto a escoger para insertar en la tabla de productos y precios.

- Base de Datos interna.

- Tipo de entidad lists.

Contiene las listas de comprar: id y fecha de creación, creadas por el usuario.

**Atributos:**

- ✓ Id: Entero autonumérico.
- ✓ fecha: fecha (aaaa/mm/dd) de inserción de la lista de número id.
- ✓ nombre: nombre de la lista

- Tipo de entidad products.

Contiene los productos de cada lista de la compra creada por el usuario. Cada producto de la lista hace referencia al id de la lista a la que pertenece.

**Atributos:**

- ✓ idLista: id de la lista en la que insertamos ese producto.
- ✓ productos: productos para la lista idLista.

### Análisis de los tipos de interrelación.

- Base de datos externa.

- TBLProvincias -- (1,n) -- TBLSupermercados.

Relaciona el tipo de entidad TBLProvincias con el tipo de entidad TBLSupermercados. A una provincia (código territorial) le pueden pertenecer varios supermercados y a un supermercado, localizado geográficamente por sus coordenadas, le corresponde una sola provincia.



- TBLSupermercados -- (1,n) -- TBLProductos.

Relaciona el tipo de entidad TBLSupermercados con el tipo de entidad TBLProductos de forma que a un supermercado, localizado geográficamente por sus coordenadas, le pueden corresponder varios productos y a un producto/precio en concreto le corresponde un solo supermercado.

- TBLDominioProductos -- (1,n) -- TBLProductos.

Relaciona el tipo de entidad TBLDominioProductos con el tipo de entidad TBLProductos de forma que los productos que puede contener la TBLProductos deben de ser uno de los contenidos en la TBLDominioProductos (que igualmente coincide con la tabla que se carga en string.xml)

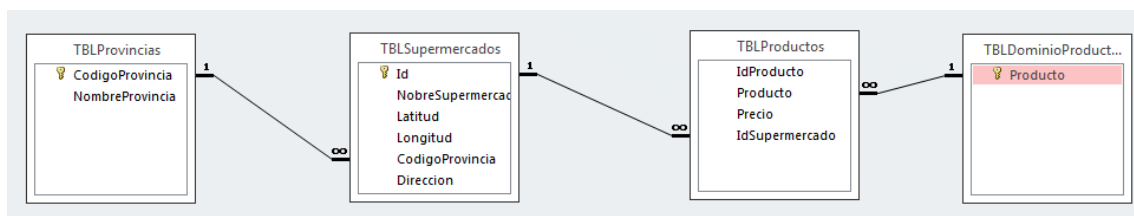
- Base de datos interna.

- lists – (1,n) – prodcts

Relaciona el tipo de entidad lists con el tipo de entidad products. A una lista de la compra le pueden pertenecer diversos productos y a un producto en particular le corresponde un sola lista de la compra

### Esquemas entidad – relación.

- Base de datos externa: registro de productos, precios, supermercados y localización. El diagrama Entidad-Interrelación que se ha obtenido es el siguiente:



La vista que relaciona las tres tablas de datos y que entrega los resultados a la aplicación del usuario, a través del webservice presentado por la aplicación externa (ASP.NET + Visual Basic.NET) es la siguiente:

```
SELECT TOP 2 TBLProductos.Producto, TBLProductos.Precio,  
TBLSupermercados.NobreSupermercado, TBLSupermercados.Latitud,  
TBLSupermercados.Longitud, TBLProvincias.NombreProvincia, TBLSupermercados.Direccion
```

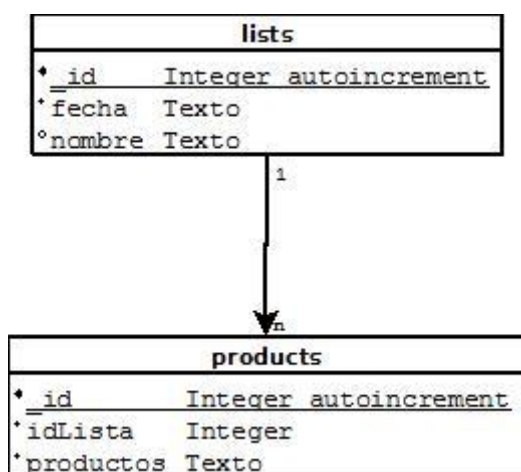


```
FROM (TBLProvincias INNER JOIN TBLSupermercados ON TBLProvincias.CodigoProvincia =
TBLSupermercados.CodigoProvincia) INNER JOIN TBLProductos ON TBLSupermercados.Id =
TBLProductos.IdSupermercado ORDER BY TBLProductos.Producto, TBLProductos.Precio;
```

- Base de datos interna al dispositivo del usuario:

```
CREATE TABLE lists (_id INTEGER PRIMARY KEY AUTOINCREMENT, fecha TEXTO, nombre
TEXTO)
```

```
CREATE TABLE products (_id INTEGER PRIMARY KEY AUTOINCREMENT, idLista INTEGER,
producto TEXTO)
```



## Arquitectura de la aplicación

Para la realización de la arquitectura de la aplicación se muestra, a continuación, el diagrama de paquetes UML. Con dicho diagrama se muestran las agrupaciones lógicas de tal forma que un paquete corresponde a un directorio y cada paquete contiene/agrupa las clases comunes al mismo.

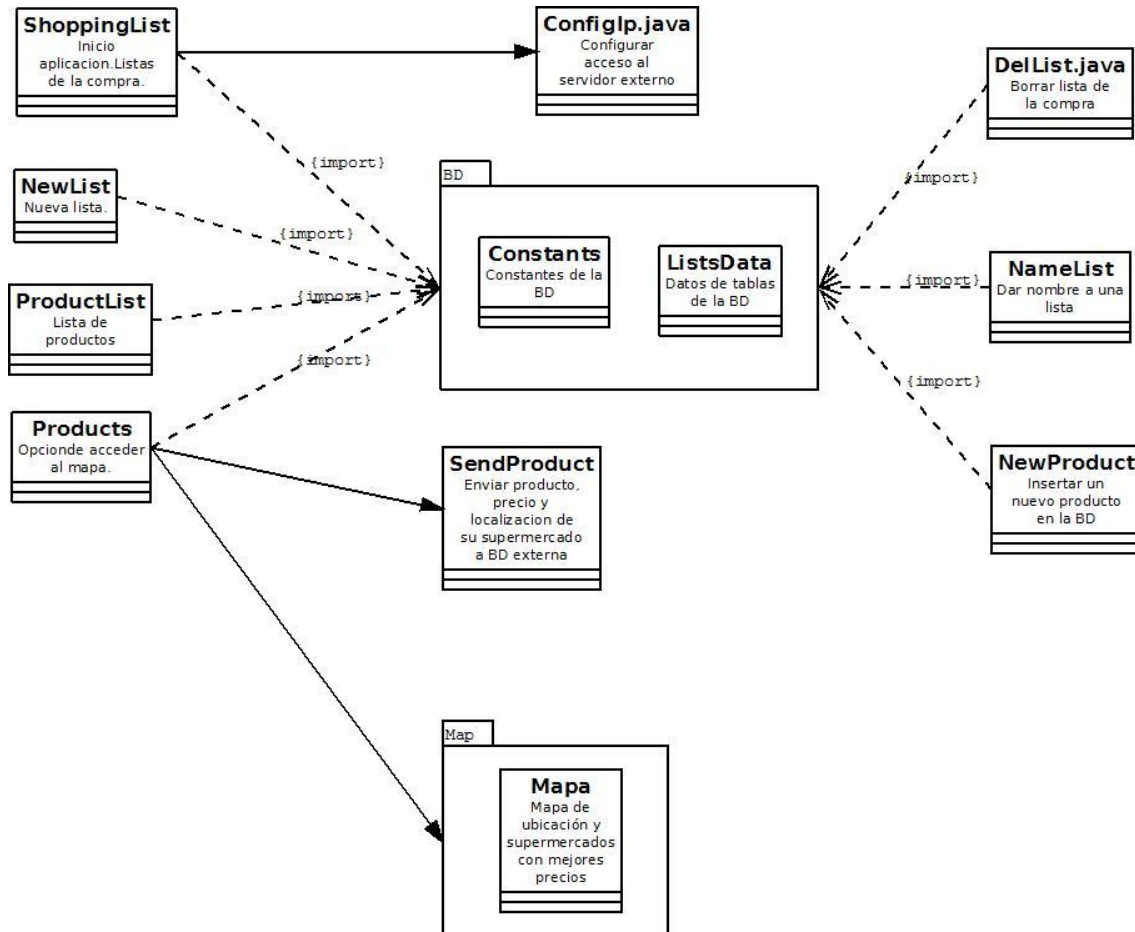
Así pues tenemos los siguientes directorios a paquetes:

1. Paquete raíz, que contiene las actividades: *ShoppingList*, *NewList*, *ProductList*, *Products*, *NewProduct*.
2. Paquete BD, que contiene elementos comunes de la base de datos y que suministran a las clases del paquete raíz los elementos necesarios para las operaciones de inserción y consulta a la BD: *Constans*, *ListData*.
3. Paquete Mapa: contiene solo una clase, la actividad mapa, que recoge el parámetro producto pasado desde la clase *Products* y consume el webservice:





<http://192.168.1.33/TFC/WebService.asmx> representando en un mapa la ubicación del usuario así como los dos supermercados con el producto pasado, más barato.

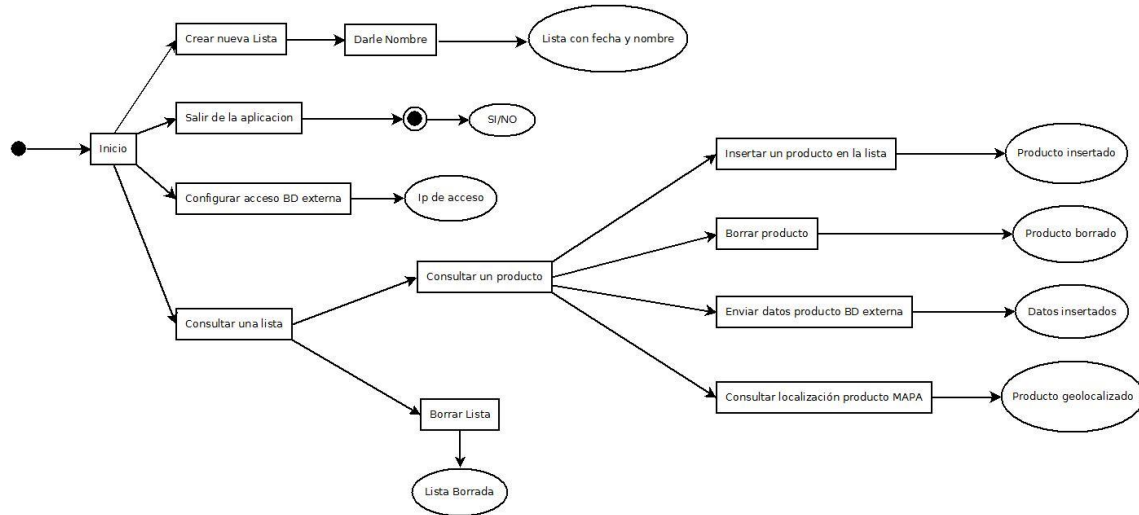


## Diseño de la interfaz

A continuación se muestra el diseño de la interfaz. Como se puede comprobar no se ha diseñado nada de forma especial, todo ello con el objeto de dedicar el escaso tiempo al objetivo del proyecto de realizar una aplicación funcional y operativa que puede servir muy bien como base sólida de una aplicación mejorada.

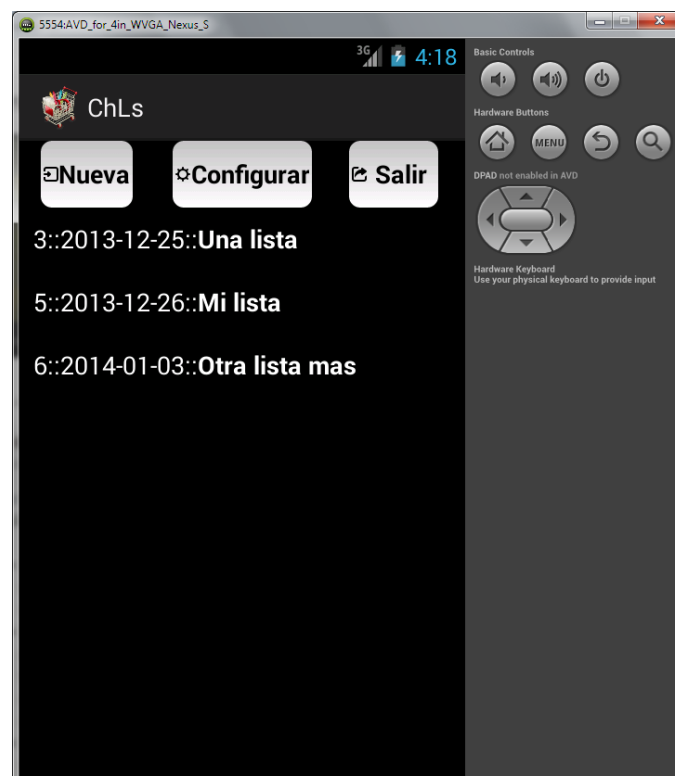


El diseño de la aplicación se ajusta al funcionamiento de la misma, según el esquema siguiente:



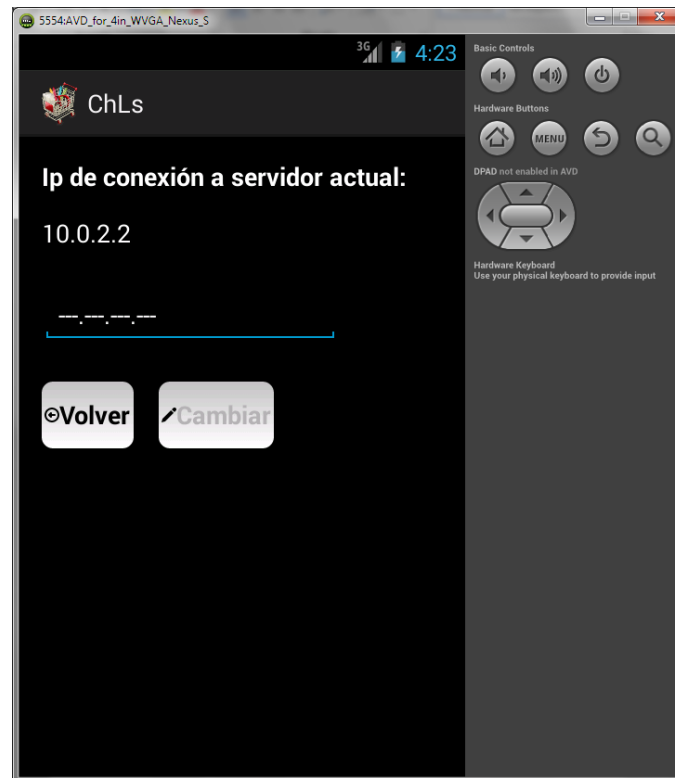
Para mostrar el diseño de la interfaz y con él todas las pantallas del funcionamiento de la aplicación he utilizado impresiones de pantalla del emulador del ADT.

Inicio de la aplicación.

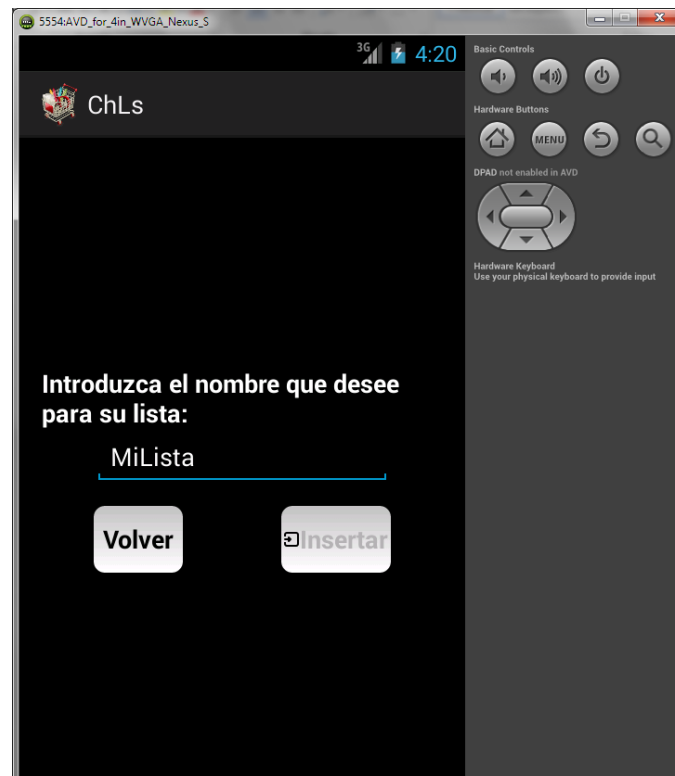


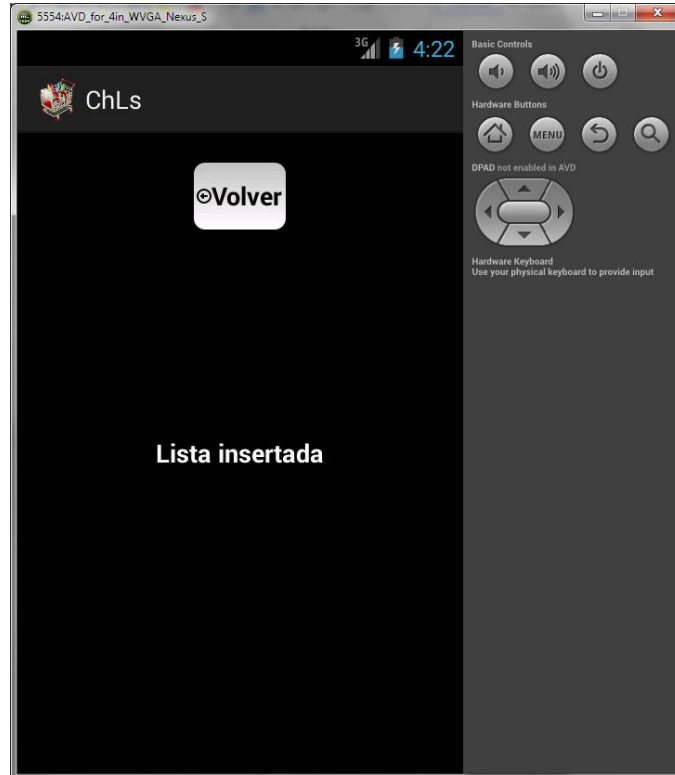


Configurar el acceso a la Base de Datos externa.

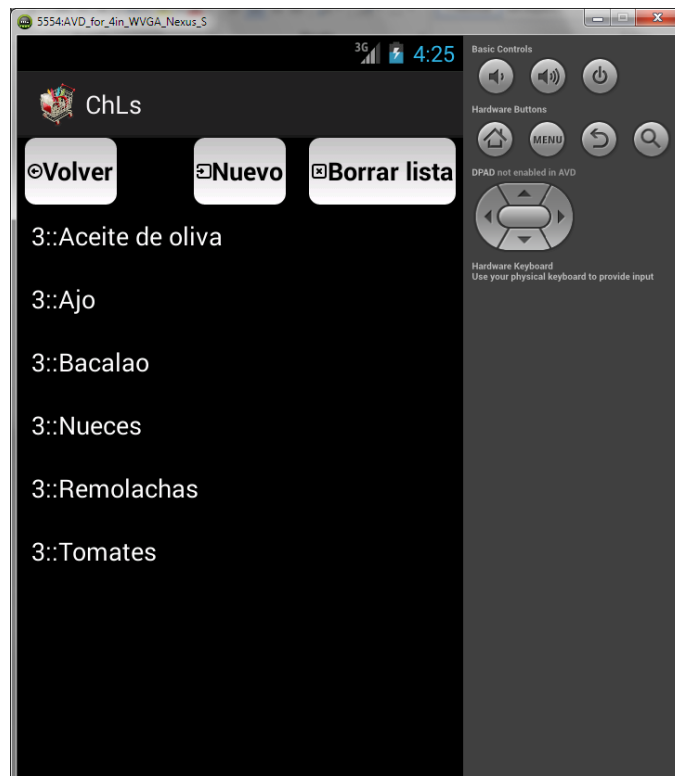


Inserción de una nueva lista.



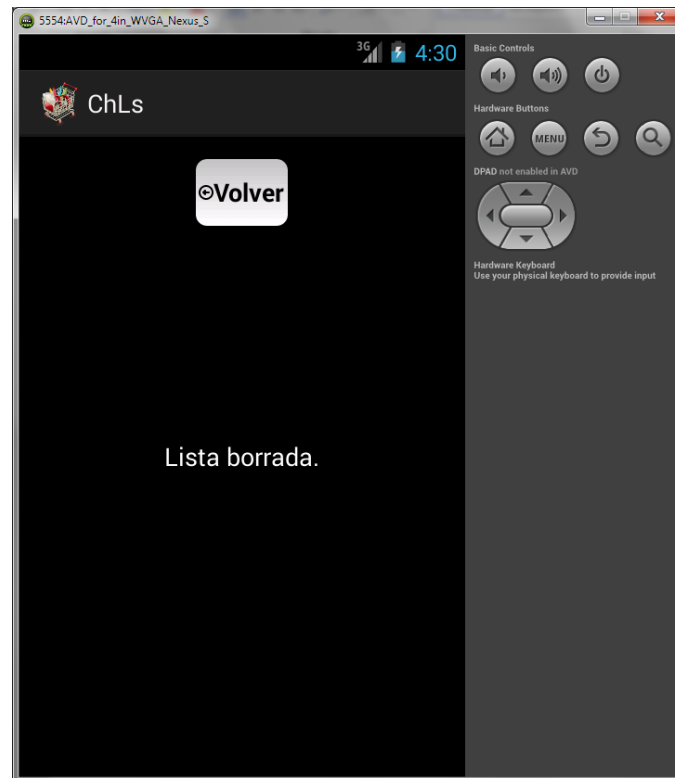


Consulta de una lista.

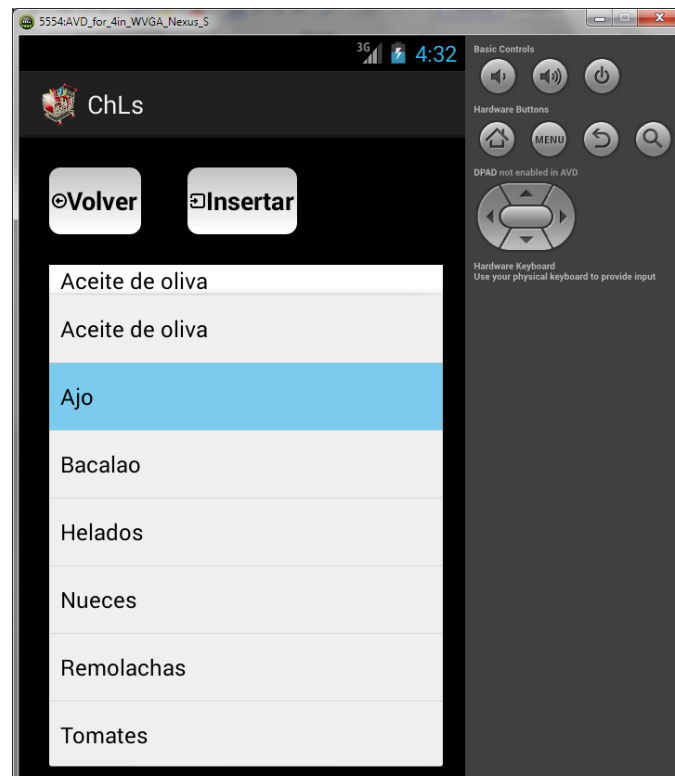


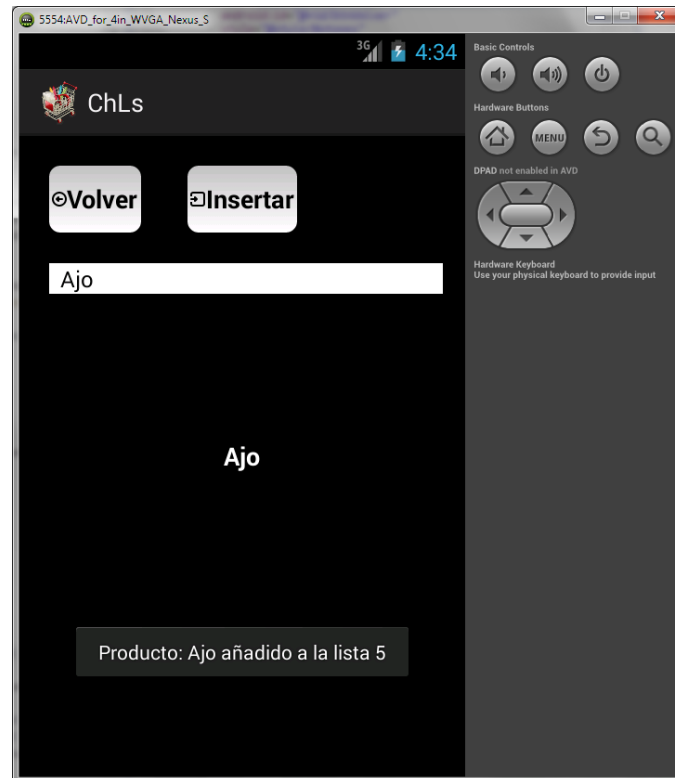


Borrar una lista.



Agregar un producto nuevo a una lista.





Consultar un producto de la lista.





Enviar precio a la Base de Datos externa.

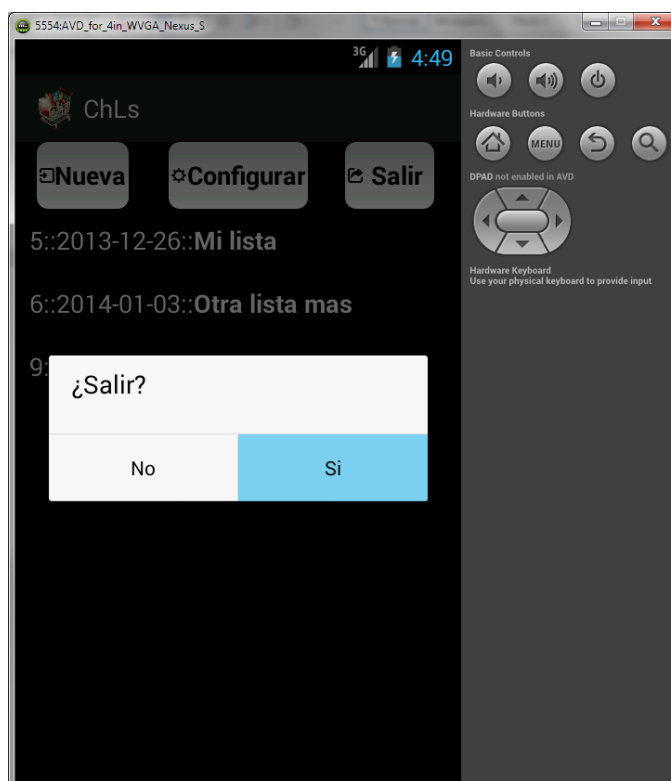


Localizar en el Mapa los dos productos más baratos con respecto a la ubicación del usuario.





Salir de la aplicación.



## Instrucciones de instalación

Como ya se ha indicado anteriormente, la aplicación del proyecto está constituida por dos partes diferenciadas: una parte interna que corresponde a la aplicación que corre en cada terminal del usuario y una parte externa, común a todos los usuarios que no es más que los datos que serán consumidos mediante un webservice por la aplicación interna de los usuarios.

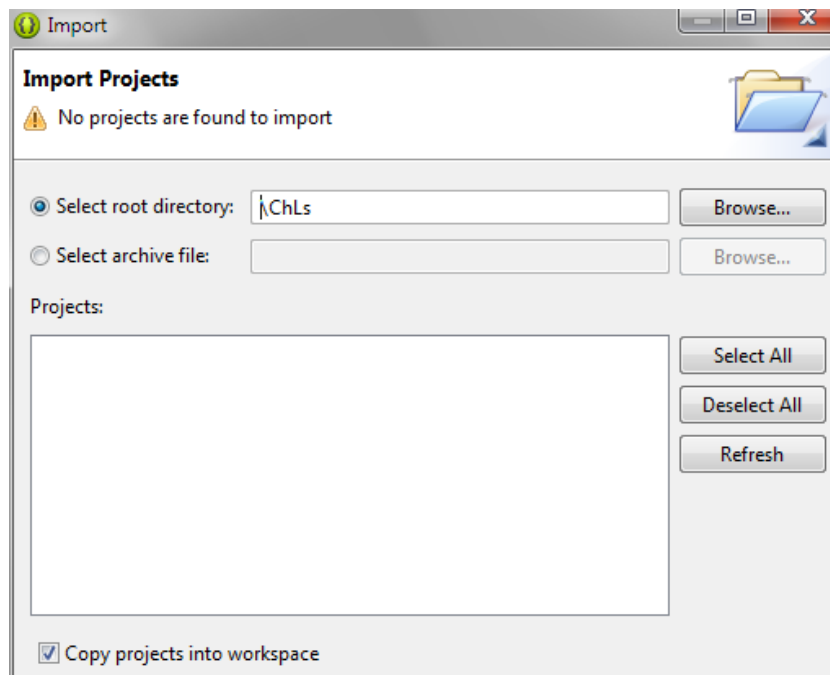
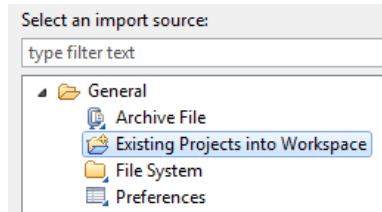
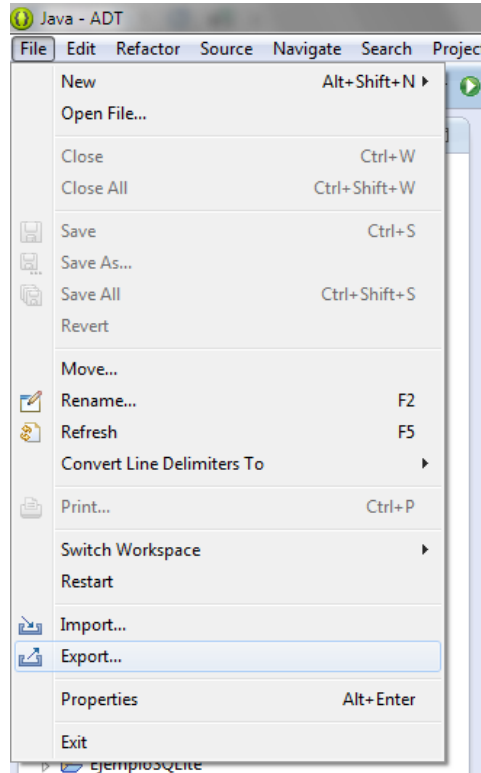
### Aplicación interna.

Corresponde al fichero del envío de la PEC3: */ChLs*

Para su desarrollo se ha utilizado el ADT de Eclipse Build: v22.0.5-757759, siendo SQLITE el motor de base de datos empleado, cuya base de datos se crea al iniciar por primera vez la aplicación.

Para instalar la aplicación interna tenemos dos posibilidades. Una es descargar en la memoria del terminal Android el fichero `ChLs\bin\ChLs.apk` y ejecutarlo directamente, con lo que se instalará la aplicación en el terminal. Otra posibilidad es importar la aplicación con el IDE de Android Eclipse:

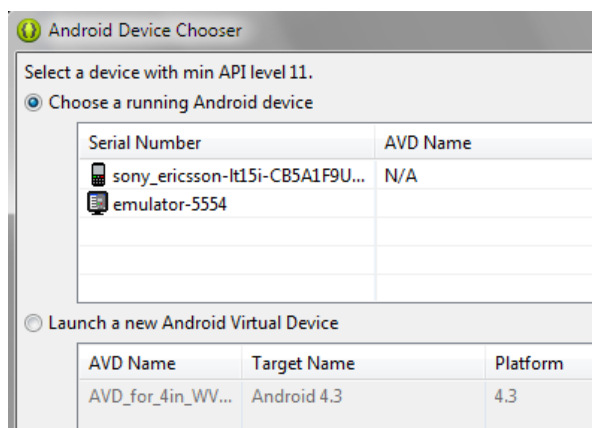
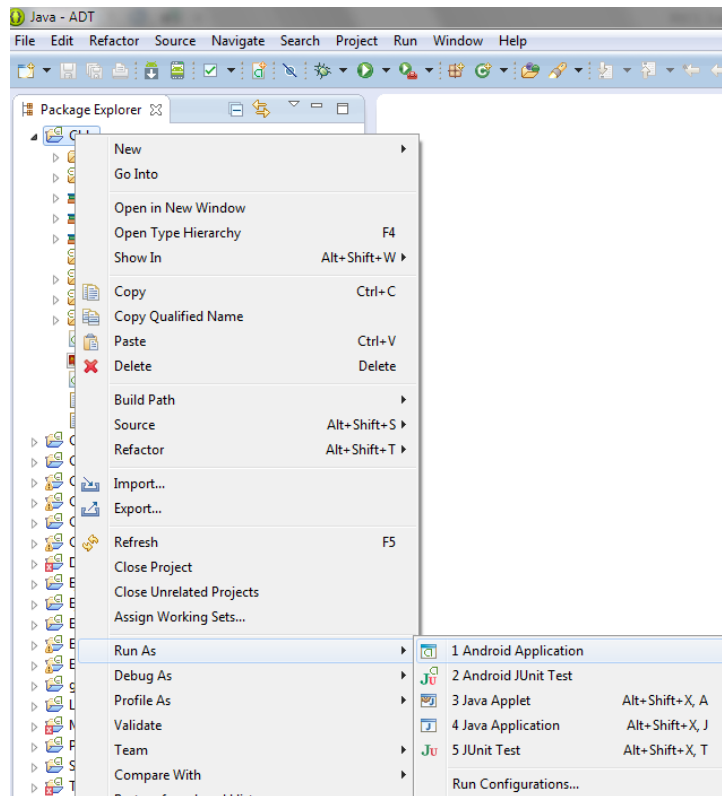






Una vez importado deberemos incluir la API de Google si no la tenemos ya. Para ello podemos guiarnos por el siguiente artículo: <<Usando Google Maps con Android>> (Escrito por Condesa) – url: <http://androideity.com/2011/10/07/usando-google-maps-con-android/>

A partir de aquí, para ejecutar la aplicación (suponemos que se ha instalado la segunda parte que explicaré a continuación) solo tenemos que ejecutarla como un proyecto de Android y elegir que la ejecución se cargue sobre uno de los emuladores configurados del propio IDE o bien sobre el terminal si éste lo tenemos conectados al PC, por ejemplo al puerto USB:



No debemos de olvidar que en caso de elegir el emulador de terminal proporcionado por el IDE, este debe dar soporte de acceso a los servicios de Google Play para emular Google maps y



funcionar correctamente, si no es así procederemos a elegir el terminal físico que tengamos conectados en el puerto USB.

En lo a que la instalación en el dispositivo físico, Smartphone o Tablet hay que tener habilitado, si la instalación se hace a través del puerto USB los siguientes parámetros, seleccionando el check correspondiente:

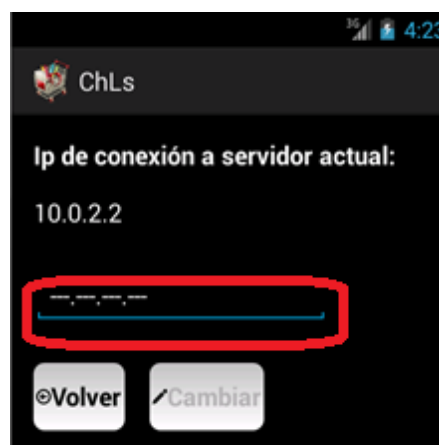
- Ajustes > Seguridad > Origenes desconocidos
- Opciones de desarrollo > Depuración USB

Hay que tener en cuenta que para la ejecución del proyecto, la url a la que llama la aplicación Android para consumir el webservice de la aplicación externa dependerá de dónde tengamos instalada la aplicación externa y cómo conectemos el terminal con la aplicación externa.

Para ello se ha añadido a la aplicación la funcionalidad configurar:



Dicha funcionalidad nos lleva a una pantalla en la que configuramos la ip de acceso al servidor manualmente:





Una vez cambiada la ip de conexión, se guardan los datos en un fichero XML en la ruta, donde se actualiza: `/data/data/com.example.chls.java/shared_prefs/ConfigIp.xml`, modificando el valor del atributo `connection` por el de la Ip especificada.

Podemos distinguir varios casos, cada uno de los cuales requiere, según sea uno u otro que configuremos, adecuadamente, el destino de la llamada:

1. Caso en que la aplicación externa resida en un servidor externo con ip 80.xx.xx.xxx (servicio web corriendo sobre la misma ip) y el terminal se conecte vía WIFI o vía 3G:
  - Ip = **80.xx.xx.xxx**
  
2. Caso en que la aplicación externa resida en un servidor interno (el mismo que el pc de desarrollo). Aquí podemos distinguir varios caso:
  - a. La aplicación llamada externa corra sobre el localhost de la máquina, en este caso el emulador o el dispositivo conectado en el puerto USB debe de apuntar a la ip 10.0.2.2 que es la que se traduce al localhost:
    - Ip = **10.0.2.2**
  
  - b. El pc forma parte de una red interna con salida a internet a través del modem/router ADSL, con ip, por ejemplo, 192.168.1.33. Este es mi caso y para ello he configurado el servidor web en 192.168.1.33 y he conectado desde el terminal Android por WIFI a la red:
    - Ip = **192.168.1.33**

No debemos de olvidar modificar, si fuese necesario, las reglas del firewalls si es que tenemos alguno instalado en el PC para que permita la comunicación del terminal al servidor.

## Aplicación externa.

Corresponde al fichero del envío de la PEC3: `/TFC`

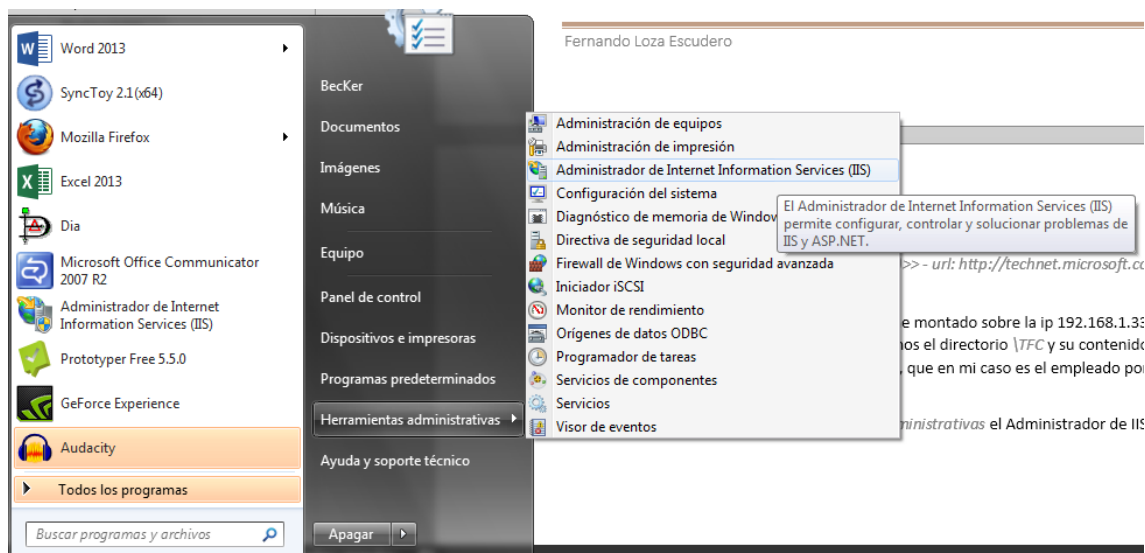
Para la aplicación externa he utilizado (por facilidad y aprovechamiento de los recursos que disponía en mi PC) un servidor de Microsoft sobre un SO W7: Internet Information Services (IIS), cuya instalación y configuración podemos realizar siguiendo las indicaciones de la siguiente página: <<Como instalar, configurar y usar el servidor IIS en Windows>>. (norfiPC) – url: <http://norfipc.com/internet/instalar-usar-servidor-web-iis-windows.html> o bien desde la propia página de Microsoft: << Servidor web (IIS)>> - url: [http://technet.microsoft.com/es-es/library/cc753433\(v=ws.10\).aspx](http://technet.microsoft.com/es-es/library/cc753433(v=ws.10).aspx)

Como ya he dicho anteriormente en mi caso lo he montado sobre la ip 192.168.1.33. Una vez instalado y configurado el servidor IIS descargamos el directorio `\TFC` y su contenido en el

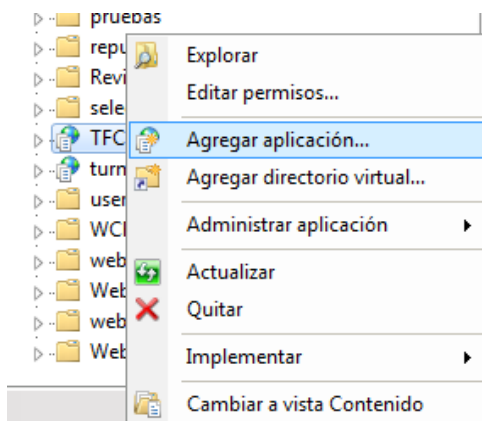


directorio que hayamos elegido raíz del servidor, que en mi caso es el empleado por defecto: `C:\inetpub\wwwroot`.

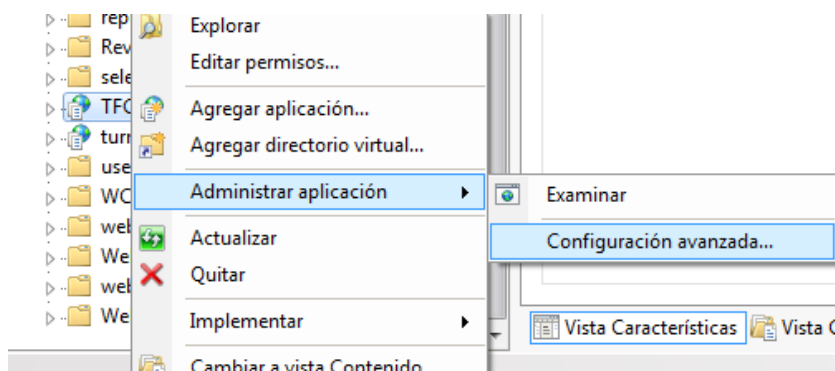
A continuación abrimos desde *Herramientas Administrativas* el Administrador de IIS:

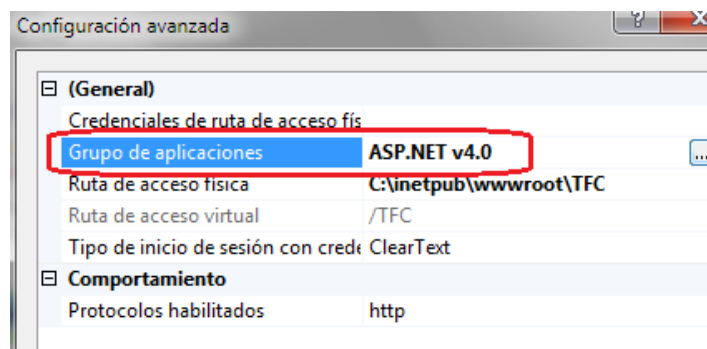


Y situándonos sobre el directorio agregado a *Sitios* lo agregamos como aplicación:



Y en configuración avanzada nos aseguramos de configurar la aplicación con la versión .NET de Framework 4.0:





En cuanto a la BD externa, realizada en mi caso en Access 2013 compatible con versiones anteriores con extensión *.mdb* no hay nada que realizar de especial, pues esta viene incluida en el propio directorio de la aplicación, por motivos de seguridad y correcto funcionamiento:  
*C:\inetpub\wwwroot\TFC\App\_Data\tfc.mdb*

## Pruebas realizadas

Como se indica en el vídeo demostración que se acompaña y se detalla en el apartado de diseño se han realizado pruebas tanto en un Smartphone SonyEricsson con pantalla de 4,1 “ como en una Tablet Pc BQ Edison pro 2 3G, con pantalla de 10,1”, en ambos satisfactorias.

## Conclusiones y Mejoras

En mi caso es la primera vez que me he enfrentado a la programación de estos dispositivos en sistemas Android. Conforme he ido avanzando en el proyecto y he ido buscando documentación he ido comprobando la infinidad de valores añadidos que se podrían aplicar al proyecto: lector de códigos de barras mediante scanner, fotografía de los productos, un chat de opiniones en tiempo real, etc.

Lamentablemente, el tiempo que le he podido dedicar al proyecto, por causas ajenas a mi voluntad, ha sido muy inferior al que hubiese deseado/necesitado, eso me ha causado una gran ansiedad a pesar de lo que he disfrutado con el desarrollo del mismo. Como consecuencia de ello me decanté por la funcionalidad básica del mismo antes que por la espectacularidad de los diseños y valores añadidos, dejando así abierta una ventana al desarrollo de posteriores versiones mejoradas en tres aspectos, que creo fundamentales: seguridad de la comunicación, mejor gestión de la BD, inclusión de valores añadidos que hagan más simple y eficaz la utilización del programa.



## Imágenes utilizadas.

- El icono de la aplicación que se ha utilizado se han descargado de la página **Getty Icons**: <http://www.gettyicons.com/free-icon/101/ecommerce-icon-set/free-full-shopping-cart-icon-png/> en la que se indican las siguientes características:
  - **Descarga:** <http://www.gettyicons.com/download/?id=2428&t=png&s=256>
  - **Autor:** Icon Shock, <http://www.gettyicons.com/author/101>
  - **Website:** <http://www.icons shock.com>
  - **Licencia:** Free For Commercial Use
- En cuanto a los iconos de los botones que se han utilizado, se han descargado de la página **Batch**: <http://adamwhitcroft.com/batch/> en la que se indica:
  - **Autor:** Adam Whitcroft, <http://adamwhitcroft.com/batch/>, <https://twitter.com/adamwhitcroft>
  - **Descarga:** <https://github.com/AdamWhitcroft/Batch/archive/master.zip>
  - **Licencia de uso:**

<< You are free to use Batch (the "icon set") or any part thereof (the "icons") in any personal, open-source or commercial work without obligation of payment (monetary or otherwise) or attribution.

Do not sell the icon set, host the icon set or rent the icon set (either in existing or modified form).

While attribution is optional, it is always appreciated.

Intellectual property rights are not transferred with the download of the icons.

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL ADAM WHITCROFT BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THE USE OF THE ICONS, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

---

I hope you enjoy Batch. I put a lot of time and effort into their design!

Adam. >>



## Bibliografía

### Consultas bibliográficas.

- **José Manuel Alarcón Aguíñ.** Programación Web con Visual Studio y ASP.NET 2.0. KRASIS Press (ISBN: 84-935489-0-2)
- **Bill Evjen; Scott Hanselman; farhan Muhammad; Sriniviasa Sivakumar; Devin Rader.** ASP.NET 2.0. Ediciones Anaya Multimedia (Grupo Anaya, S.A.), 2006 (ISBN: 84-415-2100-X)
- **Ed Burnette.** Android. Ediciones Anaya Multimedia (Grupo Anaya, S.A.), 2012 (ISBN: 978-84-415-2876-5)
- **Michael R. Groh; Joseph C. Stockman; Gavin Powell y otros.** Access 2007 (La Biblia). Ediciones Anaya Multimedia (Grupo Anaya, S.A.), 2007 (ISBN: 978-84-415-2227-5)

### Consultas Web.

- **Usando Google Maps con Android.** Escrito por Condesa.  
URL: <http://androideity.com/2011/10/07/usando-google-maps-con-android/>
- **Como instalar, configurar y usar el servidor IIS en Windows.** norfiPC  
URL: <http://norfipc.com/internet/instalar-usar-servidor-web-iis-windows.html>
- **Servidor web (IIS)**  
URL: [http://technet.microsoft.com/es-es/library/cc753433\(v=ws.10\).aspx](http://technet.microsoft.com/es-es/library/cc753433(v=ws.10).aspx)
- **Mapas en Android (Google Maps Android API v2).** Por Sgoliver el 05/12/2012  
URL: <http://www.sgoliver.net/blog/?p=3244>
- **Android Manifest, una introducción.** Publicado el 25 noviembre, 2010 por Funnybit  
URL: <http://piedralibre.wordpress.com/2010/11/25/android-manifest-una-introduccion/>
- **Consumiento Web Services SOAP en Android con ksoap2.** Por JMPergar  
URL: <http://androcode.es/2012/05/consumiento-web-services-soap-en-android-con-ksoap2/>
- **Android :: Consumir un Servicio Web SOAP.** Blog :: UPC  
URL: [http://www.innova-cloud.com/blog\\_u712770/?p=33](http://www.innova-cloud.com/blog_u712770/?p=33)
- **Programación Android: Interfaz gráfica – Diálogos y notificaciones.** 5 septiembre, 2013  
Alejandro Alcalde  
URL: [http://elbauldelprogramador.com/opensource/programacion-android-interfaz-grafica\\_11/](http://elbauldelprogramador.com/opensource/programacion-android-interfaz-grafica_11/)





- Android XML Processing with the XmlPullParser – Tutorial. Lars Vogel, Version 1.3, Copyright © 2011, 2012, 2013 Lars Vogel, 19.05.2013

URL: <http://www.vogella.com/articles/AndroidXML/article.html>

- **Introducción a los Layouts para Android.** Publicado por Ángel Carrero para Programación en Castellano.

URL: [http://www.programacion.com/articulo/introduccion\\_a\\_los\\_layouts\\_para\\_android\\_400](http://www.programacion.com/articulo/introduccion_a_los_layouts_para_android_400)

- **Programación con estilo. Pool de conexiones en VisualBasic.net y Mysql.** Posted by Slatancrouch in bases de datos, mysql, pool de conexiones, Visualbasic.net on lunes, 18 de mayo de 2009

URL: <http://programacion-con-estilo.blogspot.com.es/2009/05/pool-de-conexiones-en-visualbasicnet-y.html>

- **Preferencias en Android I: Shared Preferences.** Por sgoliver el 14/03/2011 en Android, Programación.

URL: <http://www.sgoliver.net/blog/?p=1731>