

# EXTRACCIÓN, TRATAMIENTO Y VISUALIZACIÓN DE LA INFORMACIÓN

**Trabajo Fin de Grado**

Grado en Ingeniería Informática

**Autora: Adriana Eusebio León**

**Directoras: Virginia Gutiérrez Rodríguez**

**Luz Marina Moreno de Antonio**

**Fecha: 10 de Junio de 2014**





Dña. **Virginia Gutiérrez Rodríguez** y Dña. **Luz Marina Moreno de Antonio**, Profesoras de la Escuela Técnica Superior de Ingeniería Informática, y adscrito al Departamento de Ingeniería Informática de la Universidad de La Laguna.

**CERTIFICAN:** Que la presente memoria titulada “**Extracción, tratamiento y visualización de la información**”, ha sido realizada bajo nuestra dirección por la alumna **Adriana Eusebio León**, con DNI 78641528C, y constituye su Trabajo Fin de Grado de Ingeniería Informática por la Universidad de La Laguna.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos que haya lugar, firmo el presente certificado en La Laguna, a **10 de Junio de 2014**.

Fdo: Dña. **Virginia Gutiérrez Rodríguez** y Dña. **Luz Marina Moreno de Antonio**



## Resumen

### Español

Este proyecto surge a partir de otro proyecto desarrollado anteriormente, realizado por Joaquín Aparicio Ramos y denominado "Open Supermarket, Economic Shopping List" [55]. El proyecto se basa en resolver el problema de la cesta de la compra, el cual consiste en, dada una lista de productos de la cadena de alimentación, encontrar de forma geolocalizada el supermercado más barato en el que se encuentra cada producto. Como el problema de la cesta de la compra es extenso, se acotó el proyecto a realizar una aplicación se encargue de extraer los datos de interés de los supermercados que tengan accesibles sus datos.

En este proyecto seguiremos aumentando el conjunto de información sobre productos para que en el futuro se llegue al objetivo: resolver la cesta óptima de la compra. La idea general es obtener información de un subconjunto de mercados que no se han tenido en cuenta en el anterior proyecto, por ello se recopila la información de los mercadillos del agricultor.

Se trata de extraer de diversas fuentes los precios de los productos de la cadena agroalimentaria, a nivel insular. Se generará una aplicación que permita la extracción, el tratamiento y la visualización de la información de forma oportuna para la confección de la cesta óptima de la compra a partir de la lista de productos del cliente y según los criterios que éste haya definido.

### English

This project derives from another project previously developed by Joaquín Aparicio Ramos, called "Open Supermarket, Economic Shopping List" [55]. The project is based on solving the shopping's problem, which is, given a list of products in the supply chain, find the cheapest geolocated supermarket for each product. The shopping's problem is extensive, so the project is limited to make an application that is responsible for extracting interest's data from supermarkets which have accessible data.

This project will continue to increase the set of products' information for reaching the goal: solve the optimal shopping. The general idea is to obtain information from a subset of the market that have not been taken into account in the previous project, so the information is collected from farmer's markets.

The application extract the products' prices from various sources in the food chain, islandwide. It enables the extraction, processing and display of information for making optimal shopping from the customer's products list and its criteria.



## Agradecimientos

Para empezar me gustaría dedicar este proyecto a esas personas que por desgracia ya no están y que tanta falta me hacen. Además me gustaría dedicarlo también a todas las personas que han hecho posible que haya podido llegar hasta aquí. En especial:

- ✚ A mi tutora Vicky, por confiar en mí desde el primer día y por haber aceptado realizar este proyecto conmigo. Gracias.
- ✚ A todos mis compañeros de carrera que me han ayudado durante el transcurso de ésta. Pero sobre todo a mis amigos de facultad que me han aguantado durante todos estos años: David, Iradiel, Jose, Cristo, Haridian, Guillermo, Maikel, Moisés etc...
- ✚ A Abraham, Miriam y Jose, por estar ahí siempre que los necesitaba. Por ayudarme con mis obstáculos en el desarrollo del proyecto. Gracias.
- ✚ A mi yang, Adán, por aguantar mis altibajos durante todo el desarrollo del proyecto y estar a mi lado en todo momento.
- ✚ A Judith y Bentorey, porque me enseñaron durante mi transcurso por la universidad lo importante que es tener amigos en los que poder apoyarte. Gracias por todo.
- ✚ A mi padre, porque sin él no podría ni siquiera haberme planteado realizar esta carrera. Por apoyarme en todo momento.
- ✚ A la mitad de mi vida, Laura. Por escuchar todos mis problemas y por darme los mejores consejos. Por aguantar mis lágrimas y mis gritos. Por todos esos buenos momentos que hacen que esfume todo mi estrés. Porque todo lo que me das.
- ✚ A la otra mitad de mi vida, mi madre. Porque sin ti no sería ni la sombra de la persona que soy hoy en día. Porque sin ti nada hubiese sido igual.





*<Para los que me cuidan desde el cielo>*



# Tabla de contenidos

<b>Resumen .....</b>	<b>5</b>
<b>Agradecimientos .....</b>	<b>7</b>
<b>Lista de figuras y tablas.....</b>	<b>13</b>
<b>Parte I. Introducción y fundamentos básicos</b>	<b>15</b>
<b>Capítulo 1. Introducción .....</b>	<b>17</b>
1.1 Justificación.....	17
1.2 Ideas y Objetivos.....	18
1.3 Alcance Inicial .....	18
1.4 Restricciones.....	19
<b>Capítulo 2. Estado del arte .....</b>	<b>20</b>
2.1 Antecedentes Software.....	20
<b>Parte II. Cuerpo principal. Descripción del trabajo</b>	<b>23</b>
<b>Capítulo 3. Extracción de información .....</b>	<b>25</b>
3.1 Introducción.....	25
3.2 Elección de la plataforma de desarrollo.....	25
3.3 Aplicación .....	25
3.4 Base de datos .....	29
<b>Capítulo 4. Visualización de información.....</b>	<b>32</b>
4.1 Introducción.....	32
4.2 Elección de la plataforma de desarrollo.....	32
4.3 Aplicación.....	33
4.4 Base de Datos.....	35
<b>Parte III. Conclusiones, Final</b>	<b>37</b>
<b>Capítulo 5. Conclusiones.....</b>	<b>39</b>
5.1 Acerca del uso de la librería CURL .....	39
5.2 Sobre el desarrollo del proyecto .....	39
<b>Capítulo 6. Trabajos Futuros.....</b>	<b>40</b>

---

<b>Referencias</b> .....	<b>41</b>
<b>Glosario</b> .....	<b>45</b>
<b>Anexos</b> .....	<b>46</b>
1. Definición de la base de datos relacional .....	46
2. Código de la aplicación de extracción de datos .....	50
3. Código de la aplicación móvil .....	52

## Lista de figuras y tablas

Figura 1 - Contenido del fichero correspondiente a la categoría aromáticas .....	27
Figura 2 - Interfaz web. Pantalla inicial de la extracción de datos de mercadillos .....	27
Figura 3 - Interfaz web. Pantalla de visualización de datos de mercadillos .....	28
Figura 4 - Interfaz web. Pantalla inicial de la extracción de datos de geolocalización .....	28
Figura 5 - Interfaz web. Pantalla de visualización de datos geolocalizados .....	29
Figura 6 - Diagrama relacional de la base de datos .....	29
Figura 7 - Diagrama relacional de las tablas de la base de datos que se utilizan .....	30
Figura 8 - Estudio de ventas según plataforma, realizada por Gartner .....	32
Figura 9 - Vistas de la aplicación móvil para la selección de opciones .....	33
Figura 10 - Diagrama de actividad de la selección de opción de la aplicación móvil .....	33
Figura 9 - Vistas de la aplicación móvil para la visualización de opciones .....	34
Figura 12 - Resultado JSON del servicio web REST .....	35



## **Parte I. Introducción y fundamentos básicos**





# Capítulo 1. Introducción

## Resumen:

- Justificación de la propuesta del proyecto.
- Objetivos perseguidos en el proyecto.
- Alcance del proyecto.
- Restricciones existentes para el desarrollo del proyecto.

## 1.1 Justificación

Se busca dar la mejor solución a los clientes a la hora de realizar su cesta de compra, de manera sencilla e intuitiva, y siguiendo los principios de sostenibilidad para el programa de hogares verdes del Ministerio de Agricultura, Alimentación y Medio Ambiente. Se trata de extraer de diversas fuentes los precios en la cadena agroalimentaria, generando una aplicación que permita el tratamiento y visualización de la información oportuna para la confección de la cesta óptima para la lista de productos del cliente, según los criterios que haya definido.

La Cámara de Comercio de Santa Cruz de Tenerife apoya a los mercadillos municipales, al igual que Ministerio de Economía y Competitividad, para tratar de dinamizar la actividad de un sector que en España está integrado por 1.000 recintos que dan cabida a unos 80.000 puestos de venta, mayoritariamente de alimentos frescos, que canalizan cerca del 20% de todas las frutas, verduras, carnes y pescados que se consumen, y que genera unos 280.000 puestos de trabajo.

En Canarias hay 49 mercadillos municipales: este tejido empresarial da empleo a más de 20.000 personas y en los últimos años han afrontado un profundo proceso de modernización para adaptarse a las nuevas exigencias del consumidor. La Cámara de Comercio considera además que es absolutamente necesario seguir concienciando a los canarios de la importancia de comprar en los mercadillos municipales, apoyando la producción local.

Los Mercadillos del Agricultor fueron fundados como respuesta a los agricultores para vender una parte de la producción agrícola a unos precios más ventajosos para los mismos. El Mercadillo del Agricultor se convierte en un vínculo más directo agricultor-consumidor lo que hace que la venta de los productos agrícolas sea mucho más ventajosa para el agricultor, sin llegar nunca a perjudicar al consumidor, de forma que los precios que el mismo consigue en el mercadillo son muchas veces inferiores a otros que ofrecen otros puntos de venta de la isla. Cada fin de semana el Mercadillo del Agricultor renueva su vitalidad y se reafirma claramente en el dinamismo comercial que le acompaña, un espacio vitalista que se distingue por acercar a productores y consumidores. En los últimos tiempos y bajo el influjo de la arraigada crisis se palpa una considerable necesidad de este tipo de establecimientos para relanzar la actividad económica y garantizar una cesta óptima de compra para los clientes.

## 1.2 Ideas y Objetivos

En primer lugar, se obtendrán los datos de los productos que ofertan los diferentes mercadillos, a nivel insular, de sus correspondientes páginas web. Se dispondrá de una interfaz web para la actualización y visualización de dichos datos de forma sencilla e intuitiva.

Por otro lado, se obtendrán de manera automatizada las posiciones geográficas de las Islas Canarias, de las provincias de España y de los mercadillos de los que se posee su dirección física. Estos datos también serán administrados desde la interfaz web nombrada anteriormente,

Por último, se desarrollará una aplicación móvil de visualización de datos recogidos en la parte de extracción de datos de este proyecto, junto con otros datos relativos a supermercados canarios, recogidos por el proyecto Open Supermarket [55]. Añadir que la API tendrá integrado un servicio web para el acceso a la base de datos, en la cual se encuentra toda la información.

## 1.3 Alcance

Para comenzar se creará una aplicación de extracción de datos la cual contará con dos módulos: el primero será el encargado de recoger la información de los productos de los mercadillos y el segundo se dedicará a obtener el posicionamiento de los lugares de interés.

El primer módulo recogerá el código HTML de la página web del mercadillo y lo guardará en un fichero de texto plano. Una vez tengamos el código HTML en el fichero, le realizaremos un parseo al mismo con el fin de extraer los datos que nos interesan, que serán aquellos relacionados con los productos que se ofertan, como son la categoría a la que pertenece un producto, el precio mínimo, el precio máximo, la fecha en la que tienen validez dichos valores, además de la dirección física del mercadillo.

Todos estos valores se cargarán en la base de datos, junto con datos referentes a los supermercados que han sido añadidos a esta con anterioridad. Hay que tener en cuenta que, si los productos que vamos a insertar en la base de datos ya han sido añadidos anteriormente, se deberá realizar una modificación de los datos correspondientes al producto, como son el precio y la fecha en la que tiene validez dicho precio.

El segundo módulo de la aplicación se encargará de calcular la posición geográfica del mercadillo (longitud, latitud), a partir de su dirección. Este módulo se utilizará, por tanto, cada vez que añadamos un nuevo mercado a la base de datos. Además aprovecharemos este módulo para calcular la posición geográfica de las provincias de España y de cada una de las Islas Canarias.

Por último se creará una interfaz web para la administración de los datos obtenidos. En ella se podrá visualizar toda la información que se ha extraído de las páginas webs de los mercadillos, así como la información de geolocalización. También contará con una opción para la carga de todos estos datos en una base de datos relacional.

La segunda parte de este proyecto será la realización de una aplicación móvil que tendrá como objetivo principal la visualización de los datos recogidos anteriormente. Por un lado, se visualizarán, en un mapa, los mercados que se tengan almacenados en la base de

datos, resaltando el mercado más caro y el más barato, según el precio de todos los productos que ofertan. En este caso los datos se agrupan por provincias.

Por otro lado, se visualizarán en un mapa los mercados donde se vende un determinado producto, ya sean vendidos en supermercados o en mercadillos, resaltando donde se encuentra el producto más barato y el producto más caro. Los datos se agrupan por isla canaria.

## 1.4 Restricciones

La única restricción encontrada durante el desarrollo del proyecto ocurre a la hora de obtener la geolocalización de los mercados a través de su dirección, puesto que no todas las direcciones que están almacenadas en la base de datos son reconocidas por Google Maps. Esto hará que, para aquellas direcciones no reconocidas, se tenga que insertar la posición geográfica manualmente por el administrador de ésta.

## Capítulo 2. Estado del Arte

### Resumen:

- Antecedentes software.

### 2.1 Antecedentes Software

Primeramente, se han buscado aplicaciones móviles y páginas webs que tengan unas directrices similares al objetivo general de nuestra primera parte del proyecto, que es crear una app móvil que compare los precios de los supermercados, o de un determinado producto, y nos diga cuál es el más barato, según las especificaciones del usuario.

#### 2.2.1 Aplicaciones Web

**SUPERTRUPER** [33]: Compara los precios de los productos de la cesta de la compra en distintos supermercados escaneando su código de barras. También se pueden agregar los artículos tecleando su nombre en la aplicación. Una vez confeccionada la lista de la compra, se puede comparar entre distintos supermercados. De un vistazo puedes ver cuántos productos de tu lista de la compra tiene cada supermercado y cuál es el precio global en cada uno de ellos.

**TIENDEO** [34]: Aplicación desde la que se puede consultar los catálogos, ofertas y cupones de los principales supermercados.

**CARRITUS**[35]: Permite hacer la lista de la compra desde cualquier sitio, aunque para finalizar la compra te redirige a la página web. El usuario solamente tiene que elegir supermercado y empezar a llenar su carrito. En cualquier momento puede comparar el precio de su lista en los otros supermercados y cambiar de súper tantas veces como desee.

**REDLASE** [36]: La aplicación muestra el precio de los artículos y permite a los consumidores comparar precios y, por lo tanto, encontrar el mejor. Los usuarios también pueden ver una selección de los minoristas locales que tienen el producto en stock. Aunque está la versión en español, lo cierto es que su implantación en España no es comparable con las versiones americana y británica.

**AMAZON MOVIL** [37]: La aplicación permite a los usuarios analizar contenidos y comprar por departamentos, buscar contenidos y comparar precios. La aplicación Amazon Móvil también permite a los usuarios escanear un código de barras para un acceso inmediato a la información y precio del producto.

#### 2.2.2 Aplicaciones Móviles

**SOYSUPER** [38]: Permite comparar la lista de la compra en seis supermercados online y ver así el más económico.

**TIENDEO** [39]: Web desde la que se puede consultar los catálogos, ofertas y cupones de los principales supermercados.

**CARRITUS** [40]: Tienda online que compara precios entre los supermercados que distribuyen a tu zona. En esta página encontramos un servicio que resuelve a priori el problema de la cesta de la compra.

**COMPARASUPER** [41]: Este site, que se presenta asimismo como “portal de investigación”, está en una fase de desarrollo beta. Ésta es la razón, por la que tus búsquedas dentro del portal se encuentran restringidas a las ofertas online de tres grandes superficies: Eroski, Mercadona y Carrefour y sólo se pueda consultar para cuatro ciudades: Madrid, Barcelona, Valencia y Palma de Mallorca.

**MUCHA OFERTA** [42]: Aplicación web donde encontrarás todos los catálogos, ofertas, descuentos, promociones de las tiendas de tu entorno, puesto que se busca por localidad, y sólo de aquello que te interesa, como puede ser los alimentos.

**AHORRARTE** [43]: Pretende responder a la necesidad de favorecer la transparencia y objetividad en el proceso de formación de los precios de los alimentos y otros productos, en beneficio de los consumidores y usuarios de los mismos, constituyéndose como un sitio web de consulta gratuita. Sólo supermercados de las Islas Canarias.

**SURFPRICER** [44]: Aplicación para buscar ofertas y comparar precios de todo tipo de productos entre miles de tiendas, incluyendo alimentos. Puedes buscar por nombre de producto o utilizar el escáner para leer códigos de barras y buscar más rápido. Además podrás crear alertas y estar informado cuando el precio de un producto baje hasta donde estés dispuesto a pagar.

**OCU** [45]: Aplicación web de la Organización de Consumidores y Usuarios para saber qué supermercado es el más barato de tu zona.

Por lo tanto, se pueden encontrar aplicaciones móviles y páginas webs que tengan unas directrices similares al objetivo general del proyecto, comparar los precios de un determinado producto en distintos supermercados, mostrando donde se encuentra más barato, según las especificaciones del usuario. Sin embargo, ninguna está orientada a productos de la cadena agroalimentaria canaria.

Las aplicaciones móviles que más se asemejan son Tiendeo [34], Carritus [35] y Redlase [36], y las aplicaciones web son SoySuper [38], Carritus [40], ComparaSuper [41], MuchaOferta [42] y Ahorrarte [43], resaltando ésta última ya que se centra en mercados a nivel insular.



## **Parte II. Cuerpo principal. Descripción del trabajo**





## Capítulo 3. Extracción de información

### Resumen:

- Explicación de la plataforma de desarrollo elegida.
- Base de datos utilizada en el aplicativo.

### 3.1 Introducción

Como ya hemos ido introduciendo en capítulos anteriores el desarrollo del proyecto se puede dividir en dos partes: La extracción y el tratamiento de la información por un lado, y la visualización de dicha información por otro. A continuación se profundizará en la realización del aplicativo de extracción: la plataforma de desarrollo elegida, los pasos seguidos para llevar a cabo la aplicación y la base de datos que se ha utilizado en ésta.

### 3.2 Elección de la plataforma de desarrollo

Para escoger la plataforma de desarrollo primero se valoró la opción de desarrollar en C# como Backend, y ASP.NET como Frontend, pero al final se optó por utilizar WAMP, un entorno de desarrollo web para Windows, ya que se tenía conocimiento previo con el lenguaje de programación PHP y la plataforma en general, por lo tanto se reducía considerablemente el tiempo utilizado para aprendizaje.

También se decidió utilizar Bootstrap para modificar el aspecto de la página web y la librería CURL [13], que nos aporta PHP, para extraer el código fuente de la página web de la que se extraerán la información de interés.

Aunque WAMP integra el gestor de base de datos MySQL, se optó por SQL Server 2008 R2, puesto que ya existía parte de ésta creada sobre dicha plataforma la cual fue desarrollada en el proyecto Open Supermarket [55]. Además ha tenido que configurar un puente ODBC para conectar dicha base de datos con la aplicación.










PHP no tiene un librería para conectarse directamente con el sistema gestor de base de datos SQL Server. Por ello se tendrá que crear un puente para la conexión de la aplicación de extracción de datos y la base de datos relacional donde se guardarán los datos. Se hará uso de una de las herramientas administrativas que nos ofrece Microsoft, denominada Orígenes de Datos ODBC [10].

Además se utilizará la API de Google Maps desarrollada para PHP [22] para mostrar los datos de localización en la interfaz web de la aplicación de extracción.

### 3.3 Aplicación

Para la elección de las páginas webs a las que se le realizará la extracción de datos se ha realizado un estudio de todos los mercadillos del agricultor que se encuentran en las Islas Canarias. Primeramente se han descartado los mercadillos que no tienen actualmente una

página web propia, lo que nos ha dejado con 9 mercadillos, los cuales pertenecen a las islas de Tenerife y Gran Canaria, que son:

-  Mercado del Agricultor de Güímar. [46]
-  Mercadillo del Agricultor de Tacoronte. [47]
-  Mercadillo del Agricultor de Tegueste. [48]
-  Asociación Mercado del Agricultor Arico de Abona. [49]
-  Mercadillo del Agricultor de La Orotava. [50]
-  Mercado de Nuestra Señora de Africa. [51]
-  Mercado de Vegueta. [52]
-  Mercado de Altavista. [53]
-  Mercado del Puerto.[54]





De este grupo de mercadillos, nos hemos visto limitados a elegir la página web correspondiente al Mercadillo del Agricultor de Tacoronte, puesto que es la única en la que podemos realizar la extracción de los datos.

Son varias las razones las que nos han hecho eliminar como páginas viables a las demás opciones. Existen páginas web (mercados de Güímar y Arico) que están inactivas, en los mercados de Tegueste y La Orotava los precios de los productos sólo son accesibles descargando archivos en formato PDF, y en los restantes mercados no tienen ningún listado de precios de sus productos.

La idea inicial para el módulo de extracción de datos era continuar con la metodología utilizada en el anterior proyecto a éste, Open Supermarket [55], donde se hacía uso de Web Scraping, técnica de extracción de información de webs mediante agentes o robots. Después de analizar la página web del mercadillo de Tacoronte, se descartó esta idea ya ésta no requiere interacción con el usuario, como rellenar formularios o iniciar eventos con el ratón, lo que conlleva a que dicha página sean más bien estáticas.

Como idea final, se optó por realizar un procedimiento encargado de recoger el código HTML de la página web del mercadillo para almacenarlo en un fichero de texto plano. Como el mercadillo del agricultor de Tacoronte tiene la información de los productos separados por categorías, tendremos tantos ficheros como categorías tenga el mercadillo. Esta opción es viable debido a la estructura de la web del Mercadillo del Agricultor de Tacoronte, puesto que no tiene un número elevado de categorías, al contrario que pasa con los supermercados, por lo que la cantidad de ficheros que se crea es mínima.

El último paso para obtener los datos de interés será la realización de un parseo HTML sobre los ficheros creados. Los datos que extraeremos de estos ficheros serán aquellos relacionados con los productos que se ofertan en el mercadillo, que son los siguientes:

-  Nombre del producto
-  Precio mínimo del producto
-  Precio máximo del producto
-  Fecha en la que tienen validez los valores de los precios

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 ></TD><TD WIDTH=662 ALIGN=center ><U><FONT style=FONT-SIZE:24pt FACE="Times New Roman" COLOR=#000000>LISTA DE
PRECIOS</FONT></U>
</TABLE><TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 > <TD WIDTH=662 ALIGN=CENTER ><B><FONT style=FONT-SIZE:10pt FACE="Times New Roman" COLOR=#000000>Fecha de los
precios</FONT></B> <FONT style=FONT-SIZE:16pt FACE="Times New Roman" COLOR=#000000>03/05/2014</FONT></B>
<br>
<font color="purple"size=+1><a href="PreciosA.htm">Aromáticas<a </font>
<font color="purple"size=+1><a href="PreciosB.htm">Fruta<a </font>
<font color="purple"size=+1><a href="PreciosD.htm">Verduras<a </font>
<font color="purple"size=+1><a href="PreciosF.htm">Papas<a </font>
<br>
<TABLE BORDER=1 CELLSPACING=1 CELLPADDING=1 ></TD><TD WIDTH=540 ALIGN=center ><FONT style=FONT-SIZE:24pt FACE="Times New Roman" COLOR=#000000>Aromáticas</FONT>
</TR></TABLE><TABLE bgcolor=#88F9B3 BORDER=1 CELLSPACING=0 CELLPADDING=0 ><TR HEIGHT=30 ><TD WIDTH=340 ALIGN=LEFT ><B><FONT style=FONT-SIZE:14pt FACE="Times New
Roman" COLOR=#000000 title="Entre otros los puede encontrar en los puestos de:
Mª Remedios Dorta Rodríguez en puesto '33-34'
"style="cursor:hand"><a href="javascript:INFORME('paginaSOCIOS/ProductoA_02.htm','A02')">Berros Kg.</a></FONT></B></TD><TD WIDTH=100 ALIGN=RIGHT ><B><FONT
style=FONT-SIZE:14pt FACE="Times New Roman" COLOR=#000000>1,00 €</FONT></B></TD><TD WIDTH=100 ALIGN=RIGHT ><B><FONT style=FONT-SIZE:14pt FACE="Times New Roman"
COLOR=#000000>2,00 €</FONT></B></TD></TR>
</TR></TABLE><TABLE bgcolor=#88F9B3 BORDER=1 CELLSPACING=0 CELLPADDING=0 ><TR HEIGHT=30 ><TD WIDTH=340 ALIGN=LEFT ><B><FONT style=FONT-SIZE:14pt FACE="Times New
Roman" COLOR=#000000 title="Entre otros los puede encontrar en los puestos de:
José González Dorta en puesto '45-46'
Mª Remedios Dorta Rodríguez en puesto '33-34'
"style="cursor:hand"><a href="javascript:INFORME('paginaSOCIOS/ProductoA_03.htm','A03')">Cilantro</FONT></B></TD><TD WIDTH=100 ALIGN=RIGHT ><B><FONT style=FONT-
SIZE:14pt FACE="Times New Roman" COLOR=#000000>0,15 €</FONT></B></TD><TD WIDTH=100 ALIGN=RIGHT ><B><FONT style=FONT-SIZE:14pt FACE="Times New Roman"
COLOR=#000000>0,30 €</FONT></B></TD></TR>
</TR></TABLE><TABLE bgcolor=#88F9B3 BORDER=1 CELLSPACING=0 CELLPADDING=0 ><TR HEIGHT=30 ><TD WIDTH=340 ALIGN=LEFT ><B><FONT style=FONT-SIZE:14pt FACE="Times New
Roman" COLOR=#000000 title="Entre otros los puede encontrar en los puestos de:
Mª Remedios Dorta Rodríguez en puesto '33-34'
"style="cursor:hand"><a href="javascript:INFORME('paginaSOCIOS/ProductoA_01.htm','A01')">Perejil manojo</a></FONT></B></TD><TD WIDTH=100 ALIGN=RIGHT ><B><FONT
style=FONT-SIZE:14pt FACE="Times New Roman" COLOR=#000000>0,15 €</FONT></B></TD><TD WIDTH=100 ALIGN=RIGHT ><B><FONT style=FONT-SIZE:14pt FACE="Times New Roman"
COLOR=#000000>0,40 €</FONT></B></TD></TR>
</TR></TABLE><TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 ><TR HEIGHT=30 ><TD WIDTH=440 ALIGN=LEFT ></TR>
<font color="purple"size=+1><a href="PreciosA.htm">Aromáticas<a </font>
<font color="purple"size=+1><a href="PreciosB.htm">Fruta<a </font>
<font color="purple"size=+1><a href="PreciosD.htm">Verduras<a </font>
<font color="purple"size=+1><a href="PreciosF.htm">Papas<a </font>
</body>
</html>
```

Figura 1 - Contenido del fichero correspondiente a la categoría aromáticas

Estos valores los insertaremos en la base de datos relacional correspondiente. Para ello hemos creado una interfaz web para la administración de los datos, en la que tendremos tanto la opción de insertar los valores obtenidos en la extracción en la base de datos como la opción de visualización de dichos datos. Veámosla a continuación:



Figura 2 - Interfaz web. Pantalla inicial de la extracción de datos de mercadillos

INSERTAR DATOS  
**Pulsa aquí...**  
 VISUALIZAR DATOS  
 Pulsa aquí...

### DATOS DE PRODUCTOS CANARIOS

ID	FECHA	PRECIO MINIMO	PRECIO MAXIMO
MTAromaticasBerros Kg.	2014-04-19	1	2
MTAromaticasBerros Kg.	2014-04-26	1	2
MTAromaticasBerros Kg.	2014-05-03	1	2
MTAromaticasCilantro	2014-04-19	0.15	0.3
MTAromaticasCilantro	2014-04-26	0.15	0.3
MTAromaticasCilantro	2014-05-03	0.15	0.3
MTAromaticasPerejil manojo	2014-04-19	0.15	0.4
MTAromaticasPerejil manojo	2014-04-26	0.15	0.4
MTAromaticasPerejil manojo	2014-05-03	0.15	0.4
MTFrutaAguacate Fuerte	2014-04-19	0.8	1.9
MTFrutaAguacate Fuerte	2014-04-26	0.8	1.9

Figura 3 - Interfaz web. Pantalla de visualización de datos de mercadillos

Por otra parte crearemos un módulo para calcular el geoposicionamiento de los mercados de los que hemos recogido los datos. Además utilizaremos este módulo para calcular las posiciones geográficas de las provincias de España y cada una de las Islas Canarias, datos que utilizaremos posteriormente en los aplicativos de visualización.

Para el cálculo de las coordenadas geográficas se ha hecho uso de la API de codificación geográfica que nos ofrece la API de Google Maps [17], que el proceso de transformar direcciones en coordenadas geográficas, y solo se pueden utilizar para colocar marcadores o situar el mapa. Las coordenadas obtenidas se añadirán a la base de datos en el momento que se inserte el mercado al que corresponden, pudiendo ser recalculadas posteriormente.

La interfaz web contará con una parte dedicada a éste módulo, e igual que para el módulo anterior contará con una opción para insertar estos datos en la base de datos correspondiente y otra opción para visualizar los datos, y tendría el siguiente aspecto:

PROYECTO FIN DE GRADO Inicio

INSERTAR DATOS  
**Pulsa aquí...**  
 VISUALIZAR DATOS  
 Pulsa aquí...

## GEOLOCALIZACION

El modulo de *Geolocalizacion* se encarga de calcular las posiciones geograficas de los diferentes mercados a traves de su direccion. Tambien se calculan las posiciones geograficas de las Islas Canarias, y las Provincias de España.

Se podra tanto insertar como visualizar dichos datos.

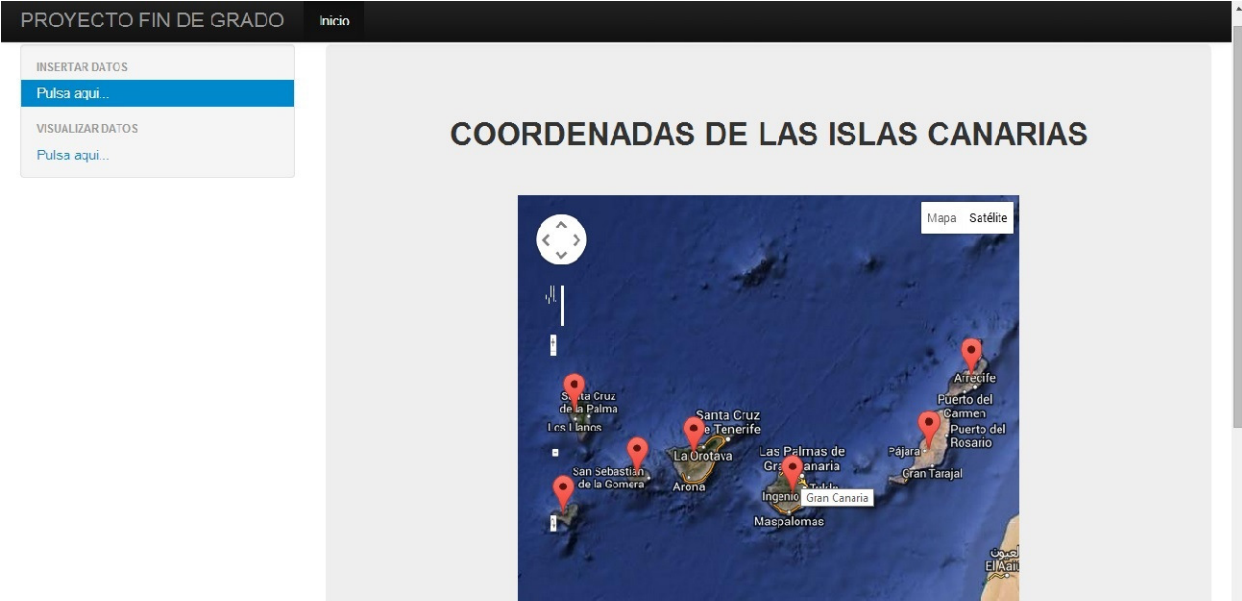


Figura 5 - Interfaz web. Pantalla de visualización de datos geolocalizados

### 3.4 Base de Datos

La base de datos realizada para este proyecto, es una adaptación de la base de datos utilizada en el proyecto OpenSupermarket [55] Se han añadido nuevas tablas y modificado algunas de las ya existentes [ANEXO “Definición de la base de datos relacional”]. Veamos a continuación la base de datos final:

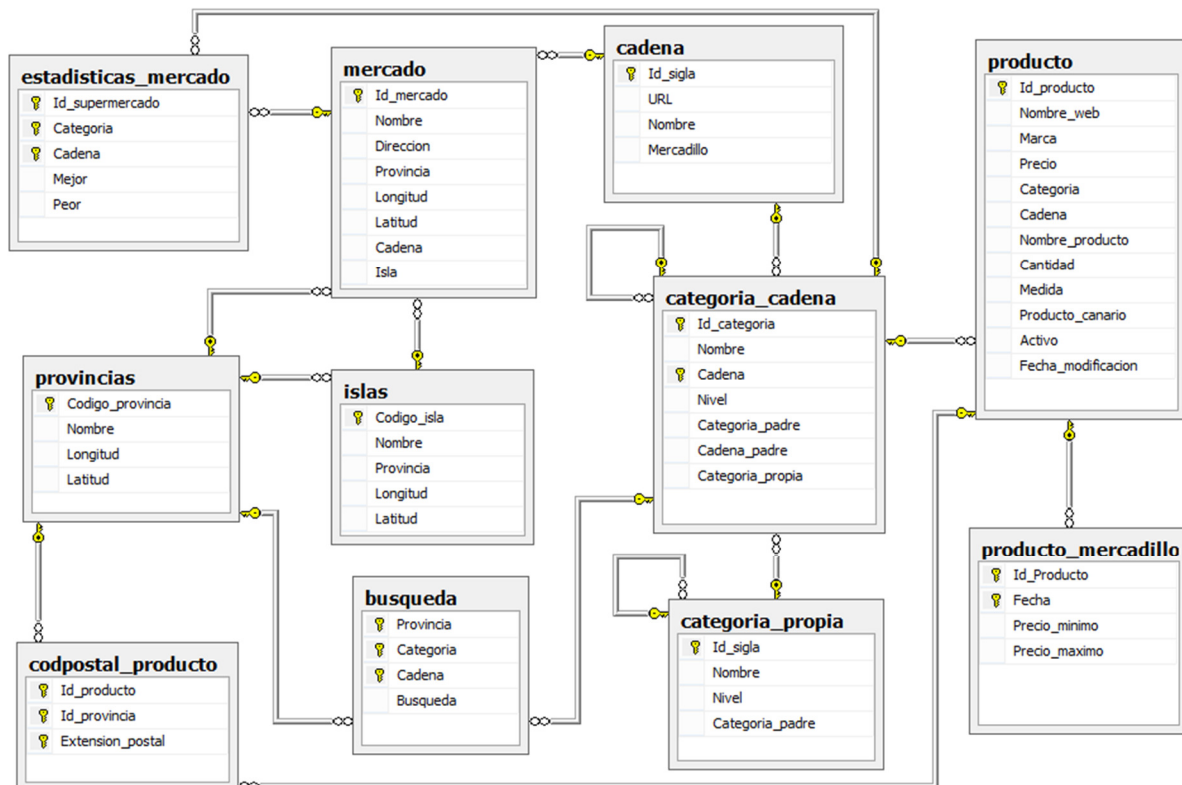


Figura 6 - Diagrama relacional de la base de datos

Las tablas que se encontraban ya creadas son 'mercado', 'cadena', 'producto', 'categoria\_cadena', 'estadisticas\_mercado', 'codpostal\_producto' y 'busqueda'. Las tres últimas tablas no son necesarias para este proyecto, puesto que no extraemos información relativa a éstas.

Además se han realizado varios cambios en la estructura de las tablas 'mercado', 'cadena' y 'producto'. A la tabla 'mercado', en la que se guardan los mercados físicos, se han añadido los atributos longitud y latitud, para mostrar los mercados en mapas, y la isla y provincia a la que pertenecen. También se ha añadido a la tabla donde se guarda la información de las cadenas de mercados, definida como 'cadena', un nuevo atributo denominado mercadillo, para diferenciar los supermercados de los mercadillos. También se añaden varios atributos a la tabla 'producto', los cuales son:

- ✚ 'Producto\_canario', para marcar los productos de origen insular.
- ✚ 'Activo', para marcar si un producto está en venta en un determinada fecha o no. Este atributo se utiliza para que la información referente al producto no quede obsoleta en las épocas que un producto no se encuentra en venta.

La parte de la base de datos que utilizamos en este proyecto es el siguiente:

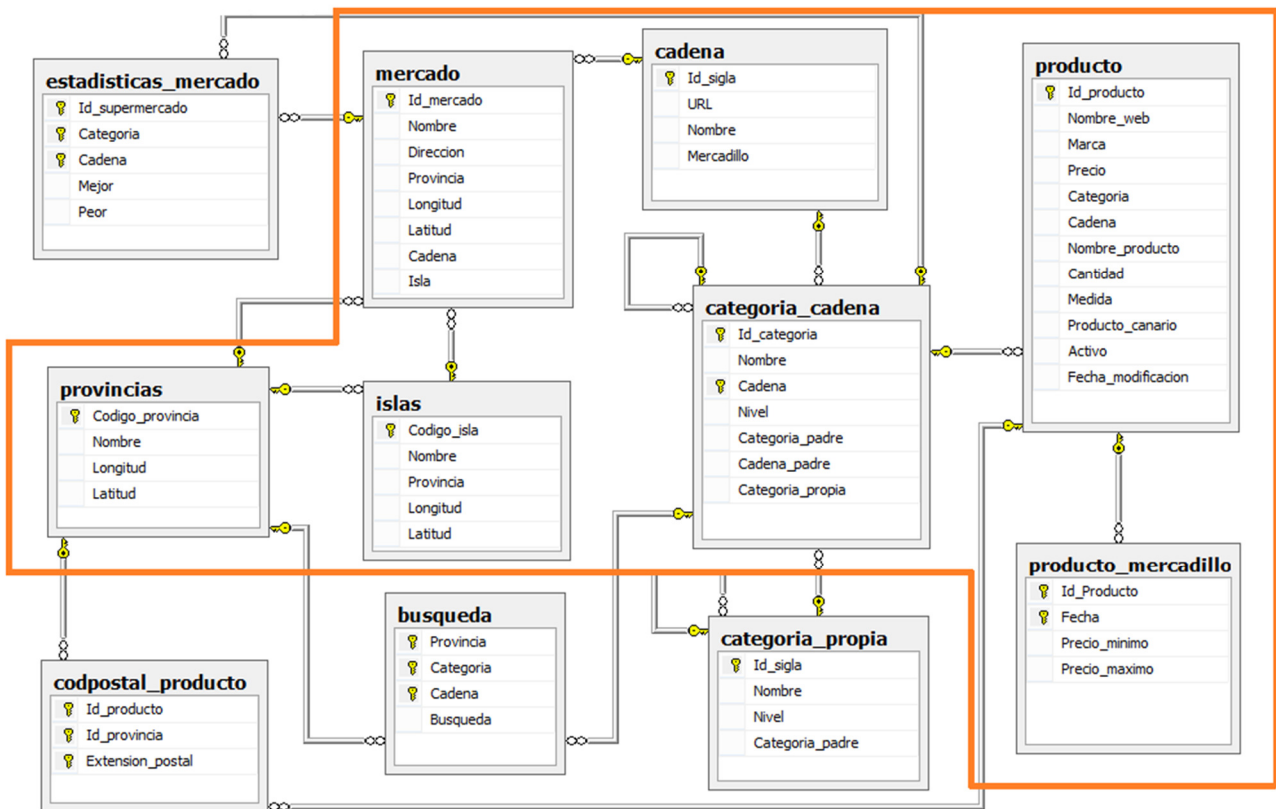


Figura 7 - Diagrama relacional de las tablas de la base de datos que se utilizan

Como podemos apreciar en la imagen anterior, se trabaja con un total de siete tablas:

- ✚ ISLA: Tabla donde se guarda la geolocalización de las Islas Canarias.
- ✚ PROVINCIA: Tabla donde se guarda la geolocalización de las Provincias de España.

- ✚ CADENA: Tabla donde se guarda la información referente a las cadenas de mercados.
- ✚ MERCADO: Tabla donde se guarda la información referente a los mercados físicos que pertenecen a una cadena de mercados, tanto de mercadillos del agricultor como de supermercados.
- ✚ CATEGORIA\_GENERAL: Tabla donde se guarda la información referente a las categorías que existen para una cadena de mercados.
- ✚ PRODUCTO: Tabla donde se guarda la información referente a los productos que se venden en los mercados atendiendo a la categoría.
- ✚ PRODUCTO\_MERCADILLO: Tabla donde se guarda la información referente a los productos que se venden en los mercadillos. En esta tabla se almacena el histórico de los precios de los productos que se ofertan en los mercadillos.

## Capítulo 4. Visualización de información

### Resumen:

- Explicación de la plataforma de desarrollo elegida.
- Base de datos utilizada en el aplicativo.

### 4.1 Introducción

Una vez lista la aplicación de extracción de datos se comenzó con la aplicación móvil de visualización de la información requerida por el usuario o como sea. A continuación se ahondará en la realización de éste aplicativo: la plataforma de desarrollo elegida, los pasos seguidos para llevar a cabo la aplicación y la base de datos que se ha utilizado en ésta.

### 4.2 Elección de la plataforma de desarrollo

Desde un principio se tenía clara la decisión de desarrollar la aplicación para la plataforma Android, puesto que actualmente es la más usada por los usuarios con una predicción a su favor bastante favorable. La justificación se encuentra en un estudio realizado por Garnet, que da como resultado un pronóstico para el año 2105 del mercado dominante, donde Android rondará sobre el 50% de usuarios, como la plataforma más elegida.

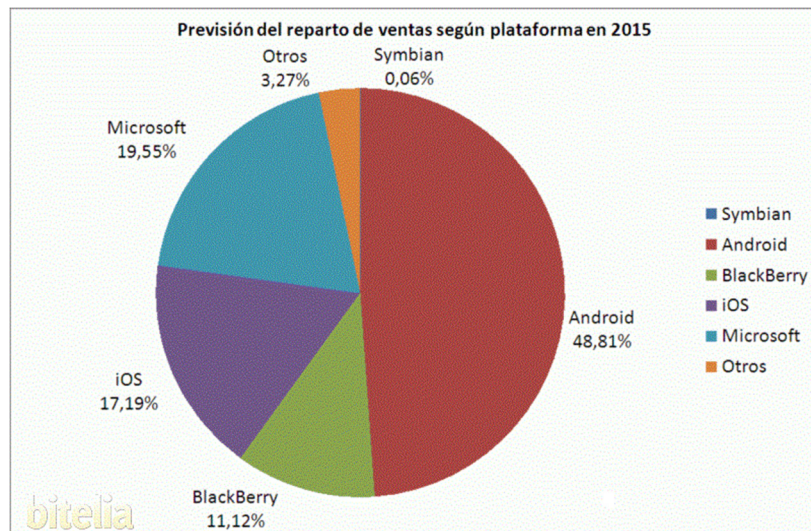


Figura 8 - Estudio de ventas según plataforma realizada por Gartner

Además se desarrollará en Java como Frontend y en XML como Backend, lenguajes de los cuales ya se tenía conocimiento previo lo que redujo considerablemente el tiempo utilizado para su aprendizaje.

En cuanto a la base de datos, se optó por SQL Server 2008 R2, puesto que ya existía parte de ésta creada sobre dicha plataforma. Además se desarrollará un Web Service de tipo REST [31] en C# para conectar dicha base de datos con la aplicación móvil, consiguiendo un mínimo acoplamiento entre ambas.



Se hará uso de la API de Google Maps [23] para mostrar los datos de localización en la app. Además se utilizará el servicio de codificación geográfica de Google [17], para calcular las coordenadas geográficas de los mercados.

### 4.3 Aplicación

Antes de comenzar a desarrollar la aplicación se realizó un prototipado de todas las pantallas con conformaría dicha aplicación. Se ha utilizado el IDE Eclipse para realizar el prototipado ya que cuenta con una interfaz gráfica para la creación de las vistas. A la hora de desarrollar, y aunque el IDE Eclipse cuente la utilidad de crear las vistas gráficamente, se han programado cada una de ellas utilizando XML.

A continuación se muestran las vistas que conformarán la parte de selección de opciones de la aplicación.

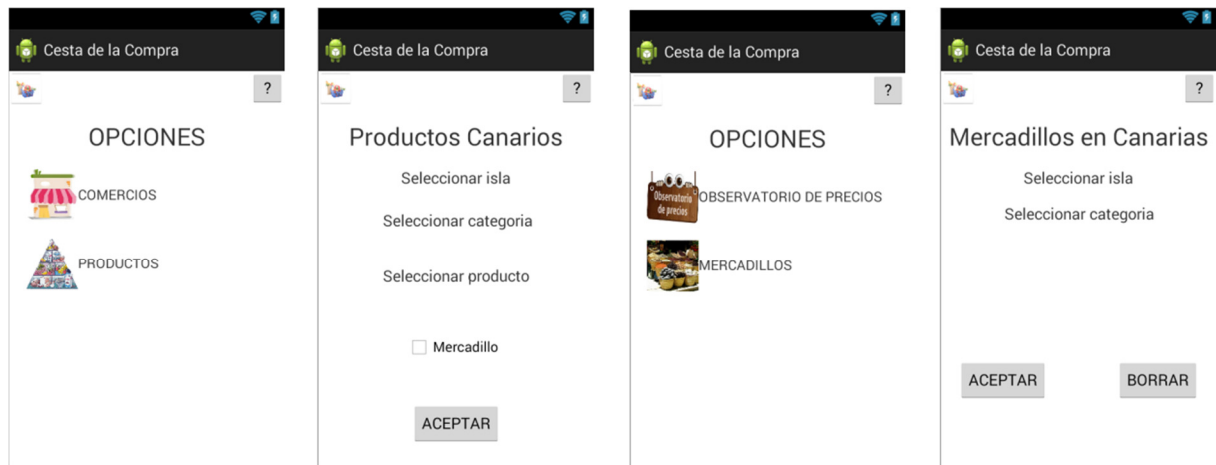


Figura 9 - Vistas de la aplicación móvil para la selección de opciones

Para explicar cómo se realiza el proceso de elección de opciones por parte del usuario se ha creado el siguiente diagrama de actividades:

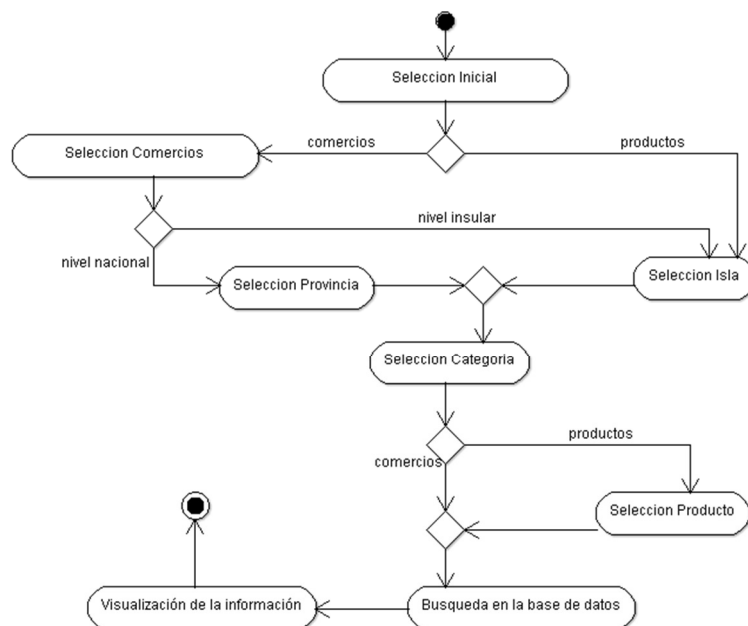


Figura 10 - Diagrama de actividad de la selección de opción de la aplicación móvil

Como se puede observar, una vez seleccionados todos los parámetros, se realizará una búsqueda en la base de datos de la información que coincida con estos, para mostrarla en una nueva pantalla.

Una consulta a la base de datos se define como una operación larga o costosa, y si la realizamos sobre el hilo principal de la aplicación bloqueara la ejecución del resto de componentes y la interfaz, produciendo al usuario un efecto evidente de lentitud, bloqueo, o mal funcionamiento en general. Por ello, estas consultas se realizarán sobre un hilo secundario.

Para trabajar con hilos secundarios en Android hacemos uso de la clase auxiliar AsyncTask [26]. La forma básica de utilizar esta clase consiste en crear una nueva clase que extienda de ella y sobrescribir varios de sus métodos entre los que repartiremos la funcionalidad de nuestra tarea. Estos métodos son los siguientes:

- ✚ onPreExecute(). Se ejecutará antes del código principal de nuestra tarea. Se suele utilizar para preparar la ejecución de la tarea, inicializar la interfaz....
- ✚ doInBackground(). Contendrá el código principal de nuestra tarea.
- ✚ onPostExecute(). Se ejecutará cuando finalice la tarea, o dicho de otra forma, tras la finalización del método doInBackground().
- ✚ onCancelled(). Se ejecutará cuando se cancele la ejecución de la tarea antes de su finalización normal.

A continuación se muestran las vistas que conforman la visualización de los datos.

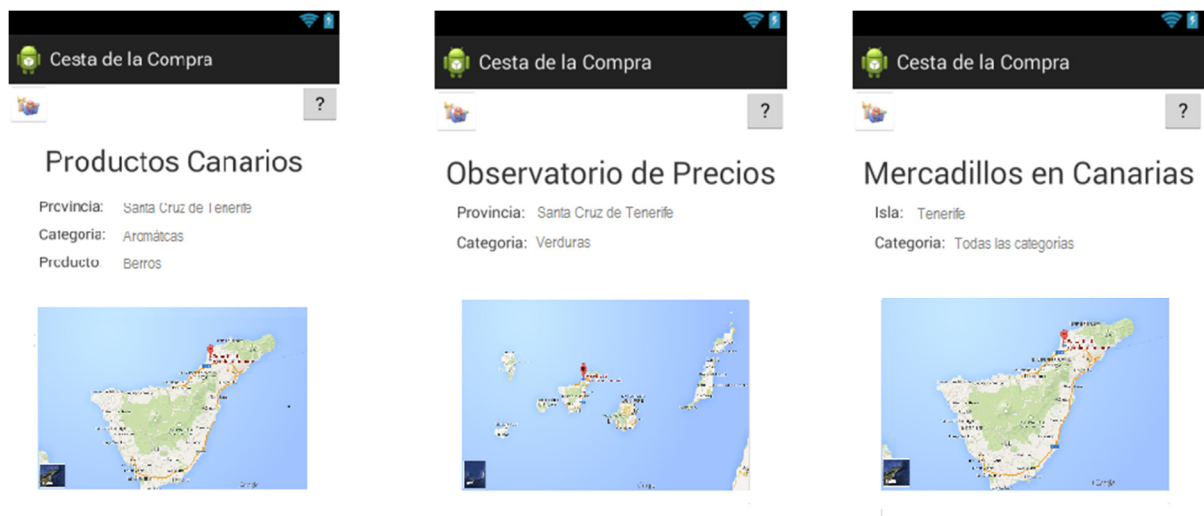


Figura 11 - Vistas de la aplicación móvil para la visualización de opciones

En todas estas vistas finales se mostrarán las opciones elegidas por el usuario y, a continuación, un mapa con los mercados que coincidan con dichas opciones. La información referente a cada mercado se podrá visualizar si se posiciona el puntero del ratón sobre el marcador correspondiente. Para poder añadir mapas a la aplicación utilizaremos la API de Google Maps [19].

Por último, se han creado una vista de ayuda para cada una de las pantallas que hemos definido anteriormente. Con esto se minimizan las dudas que puedan llegar a tener los usuarios finales.

Además se ha programado un servicio web, usando el protocolo de comunicación REST, para la conexión entre la aplicación móvil y la base de datos. Desde la aplicación móvil enviaremos al web service la consulta que deseemos realizar sobre la base de datos y dicho web service será el encargado de hacer las llamadas a ésta. El resultado obtenido será devuelto con formato JSON, como vemos a continuación:

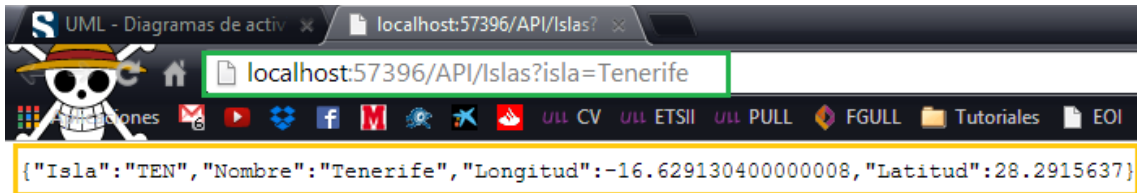


Figura 12 - Resultado en JSON del servicio web REST

El resultado será recogido desde la aplicación móvil. Para ello tendremos que añadir a nuestro proyecto Android la librería ksoap2, la cual nos dará la funcionalidad necesaria para parsear el resultado de manera que se pueda manipular en el código fuente de la aplicación. Una vez hecho esto solo nos quedará utilizar los datos obtenidos para, por un lado, crear el mapa y, por otro, añadir a este mapa los datos de interés para el usuario final.

## 4.4 Base de Datos

La base de datos de la cual se extraen los datos que mostraremos en la aplicación móvil es aquella que hemos visto en el apartado 3.4.

Primeramente se extraerán los datos de geolocalización para ajustar las dimensiones del mapa a la zona elegida por el cliente. Si la opción elegida es una isla, entonces los datos se sacarán de la tabla “islas” y si la opción es una provincia, se extraerán los datos de la tabla “provincias”.

A continuación se añadirán al mapa los marcadores de los mercados de la zona elegida. Para ello se extraen los datos de los mercados que coinciden con la provincia o isla seleccionada y con la categoría elegida por el usuario. Estos datos se extraerán de la tabla “mercados”. [ANEXO “Definición de la base de datos relacional”]

Por último se añadirán los datos que puedan ser de utilidad al cliente al cuadro de texto correspondiente a cada marcador. Estos datos dependerán de nuevo de las opciones elegidas por el usuario. Si el usuario elige cualquiera de las dos opciones correspondientes a comercios entonces en el cuadro de texto aparecerá únicamente los datos relativos a comercios como son la cadena de supermercados a la que pertenece y la dirección donde se encuentra, por lo tanto los datos solo se extraen de la tabla “mercados”. Por el contrario, si el usuario elige la opción de productos, aparecerá en el cuadro de texto la información de los mercados y, en el caso de que la búsqueda sea de un único producto, se mostrarán también los datos correspondientes a ese producto, como puede ser el precio del producto y la fecha de validez de ese precio. Estos datos se extraerán de la tabla “producto” y/o la tabla “producto\_mercadillos”.



## **Parte III. Conclusiones, Final**



## Capítulo 5. Conclusiones

### 5.1 Acerca del uso de la librería CURL

Español

Durante el desarrollo de este proyecto nos hemos planteado hasta qué punto es ético el uso de herramientas de extracción en Internet. Creemos que los datos obtenidos deberían ser públicos y de fácil acceso, de manera que no se tengan que utilizar herramientas de extracción de datos web, puesto que éstas se pueden utilizar para fines impropios. Sin embargo, con herramientas de este tipo podemos contribuir a que la información que en teoría debe ser accesible para los usuarios de verdad lo sea.

English

During the project development we have considered if is ethical to use extraction tools on the Internet. We believe that the data obtained should be public and easily accessible, so no extraction tools will be needed since they can be used for improper purposes. However, such tools can contribute to the information that, in theory, should be accessible to users be really available.

### 5.2 Sobre el proyecto

Español

Con la realización de este proyecto, junto con el proyecto Open Supermarket [55], hemos logrado un gran avance sobre la idea inicial que se tenía, que era la realización de una aplicación que ayude a los usuarios finales a realizar la cesta de la compra óptima.

Además este proyecto puede ayudar a potenciar el consumo de productos canarios. Bajo la influencia de la arraigada crisis por la que estamos pasando, existe una considerable necesidad de crecimiento de los mercadillos del agricultor, con el fin de relanzar la actividad económica y garantizar una cesta óptima de compra para los clientes.

English

With the project completion, along with the Open Supermarket [55] project, we have made great progress on the initial idea, which was the realization of an application that helps end users to make the optimal shopping.

Furthermore, this project can help boost Canarian products consumption. Under the crisis' influence, there is a considerable need for farmer's markets existence, to boost economic activity and ensure optimal shopping cart for customers.





## Capítulo 6. Trabajos Futuros

A continuación se listan los trabajos futuros que se podrían realizar a partir del proyecto “Extracción, Tratamiento y Visualización de la Información”:

- ✚ Incorporación de un sistema de revisión continua que detecten cambios en las páginas de los mercadillos, o de otras páginas que recopilen esos datos.
- ✚ Incorporación de sistema de autenticación en ambos aplicativos.
- ✚ Creación de una API web que tenga la misma funcionalidad que la aplicación móvil, para que los usuarios finales puedan elegir el entorno donde crear su lista de la compra óptima.
- ✚ Ampliando el alcance del proyecto, se podría crear una aplicación que establezca rutas mínimas entre mercados.

## Referencias

Todos los links de las referencias han sido comprobados el día XX-06-2014

### Páginas Web:

#### Cursos de HTML, JavaScript, JQuery

1. <http://es.tldp.org/Manuales-LuCAS/doc-curso-html/doc-curso-html.pdf>  
Curso completo de HTML.(libro)
2. <http://html5tutoriales.com/tutoriales/tutorial-basico/iniciando-con-html5.html>  
Tutoriales básicos de HTML5 y JQuery.
3. <http://aprendeonline.udea.edu.co/lms/moodle/course/view.php?id=414>  
Curso básico de JavaScript.
4. <http://try.jquery.com/>  
Curso en inglés de JQuery muy completo, con videos y ejercicios.
5. <http://www.imaginanet.com/blog/usando-ajax-un-simple-ejemplo-con-php-y-jquery.html>  
Paso de parámetros desde Cliente al Servidor con AJAX
6. <http://www.anerbarrena.com/jquery-load-html-div-460/>  
Carga de un HTML externo dentro de un div de HTML.
7. <http://librojquery.com/>  
Curso completo de JQuery (libro)
8. [http://mundosica.github.io/tutorial\\_hispano\\_jQuery/todo/manual\\_jquery\\_basico.pdf](http://mundosica.github.io/tutorial_hispano_jQuery/todo/manual_jquery_basico.pdf)  
Manual básico de JQuery
9. <http://www.jose-aguiar.com/blog/efecto-cargando-con-jquery-ajax/>  
Mensajes de carga con JQuery

#### ODBC

10. <http://www.php.net/manual/es/book.uodbc.php>  
Manual muy completo de PHP. Sección dedicada a ODBC.
11. <http://www.maestrosdelweb.com/editorial/odbc-y-php-en-7-pasos/>  
Manual de configuración de ODBC con PHP

#### CURL

12. <http://web.ontuts.com/tutoriales/aprendiendo-a-utilizar-la-libreria-curl-en-php/>  
Tutorial de CURL con varios ejemplos de uso.

13. <http://www.php.net/manual/es/book.curl.php>

Manual muy completo de PHP. Sección dedicada a CURL.

14. <http://www.phperos.net/foro/index.php?topic=5672.0>

Foro donde se explican las funciones para trabajar con CURL, con ejemplos.

15. <http://www.sitepoint.com/using-curl-for-remote-requests/>

Tutorial de CURL en inglés con ejemplos de uso.

16. <http://massyflores.wordpress.com/2011/03/31/activando-y-desactivando-las-curls-en-wamp/>

Como habilitar la librería CURL si se utiliza WAMP

### API Google Maps

17. <http://www.usosweb.com/content/tutorial-geocodificaci%C3%B3n-en-google-maps>

Tutorial de Codificación Geográfica, con ejemplos.

18. <http://www.openalfa.com/index.php/informacion-geografica/google-maps-api>

Tutorial de Google Maps, en el que se incluye la codificación geográfica

19. <http://norfipc.com/web/como-insertar-mostrar-mapas-google-paginas-web.php>

Manual de utilización de Google Maps V2

20. <http://www.maestrosdelweb.com/editorial/google-maps-api-v3-introduccion-y-primeros-pasos/>

Manual de utilización de Google Maps V3

21. <https://developers.google.com/maps/faq?hl=es>

Manual oficial de Google Maps, preguntas frecuentes.

22. <http://bradwedell.com/php-google-maps-api>

API de Google Maps desarrollada para PHP.

23. <http://www.elandroidelibre.com/2013/10/desarrollando-en-android-2-google-maps-api.html>

Manual de configuración y utilización de Google Maps para aplicaciones Android

24. <http://www.sgoliver.net/blog/?p=3244>

Tutorial completo de desarrollo en Android. Parte dedicada a Google Maps.

### Tareas asíncronas con Android

25. <http://danielme.com/tip-android-2-realizar-tareas-asincronas-con-async-task/>

Tutorial básico de tareas asíncronas, con ejemplo.

26. <http://www.sgoliver.net/blog/?p=3099>

Tutorial completo de desarrollo en Android. Parte dedicada a tareas asíncronas.

27. <http://www.nosinmiubuntu.com/2013/05/como-manejar-tareas-en-segundo-plano.html>

Tutorial donde se explica muy bien las funciones de una tarea asíncrona.

### Web Service

28. <http://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/>

Tutorial donde se explica que son los servicios web.

29. <https://www.youtube.com/watch?v=iXyt-BwOdcE>

Video-tutorial extenso de conexión entre aplicación Android y un servicio web.

30. <http://www.sgoliver.net/blog/?p=2571>

Tutorial completo de desarrollo en Android. Parte dedicada a servicios web SOAP.

31. <http://www.sgoliver.net/blog/?p=2610>

Tutorial completo de desarrollo en Android. Parte dedicada a servicios web REST.

32. <http://androcode.es/2012/05/consumiento-web-services-soap-en-android-con-ksoap2/>

Tutorial de servicios web SOAP utilizando la librería ksoap2 para Android

### Aplicaciones web y móviles similares a la idea del proyecto:

33. Super Truper: <http://www.supertruper.com/>

34. Tiendeo: <http://www.tiendeo.com/info/apps-movil/>

35. Carritus: <http://itunes.apple.com/es/app/carritus.com/id497930135?mt=8>

36. Red Lase: <http://redlaser.com/>

37. Amazon Movil: <https://play.google.com/store/apps/details?id=uk.amazon.mShop.android>

38. Soy Super: <https://soysuper.com/>

39. Tiendeo: <http://www.tiendeo.com/>

40. Carritus: <http://www.carritus.com/>

41. Compara Super: <http://comparasuper.com/>

42. Mucha Oferta: <http://www.muchooferta.com/>

43. Ahorrarte: <http://www.ahorrarte.com/>

44. Surf Pricer: <http://www.surfpricer.es/>

45. OCU: <http://www.ocu.org/>

### Páginas web de mercadillos:

46. Mercado del Agricultor de Güímar, <http://www.mercadoguimar.com/>.

47. Mercadillo del Agricultor de Tacoronte, <http://www.mercadillodelagricultor.com/>.

48. Mercadillo del Agricultor de Tegueste, <http://www.mercadillodetegueste.es/>.
49. Asociación Mercado del Agricultor Arico de Abona, <http://www.mercaarico.es/>.
50. Mercadillo del Agricultor de La Orotava, <http://www.villadelaorotava.org/mercadillo.php>.
51. Mercado de Nuestra Señora de Africa, <http://www.la-recova.com/>.
52. Mercado de Vegueta, <http://www.mercadovegueta.com/>.
53. Mercado de Altavista, <http://www.mercadoaltavista.com/>.
54. Mercado del Puerto, <http://www.mercadopuerto.com/>.

### **Proyecto de Fin de Carrera de Joaquín Aparicio Ramos:**

55. Open Supermarket, Economic Shopping List, ETL Web

## Glosario

**API:** Application Programming Interface. Conjunto de clases que ofrecen un servicio.

**CURL:** Librería de funciones para realizar acciones sobre archivos que hay en URLs de Internet. Soporta los protocolos más comunes, como http, ftp, https, etc

**Codificación geográfica:** Proceso de convertir direcciones físicas en coordenadas geográficas (longitud, latitud). Herramienta de la API de Google Maps.

**DSN:** Domain Name System es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada.

**Garnet:** Una de las principales compañías de investigación y asesoramiento de las tecnologías de la información.

**ODBC:** Open DataBase Connectivity (ODBC) es un estándar de acceso a las bases de datos desarrollado por SQL Access Group en 1992.

**SOAP:** Simple Object Access Protocol. Protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

**WAMP:** Entorno de desarrollo web para Windows con el que se podrán crear aplicaciones web con Apache, PHP y base de datos MySQL.

**Web Scraping:** Técnica de extracción de información de webs mediante agentes o robots.

**Web Service:** Tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

# Anexos

## 1. Definición de la base de datos relacional

Se procede a definir las tablas que conforman la base de datos relacional:

- ✚ PROVINCIA: Tabla donde se guarda la geolocalización de las Provincias de España. Sus atributos son:
  - ✚ Código\_provincia (varchar(3)): Clave primaria. Código significativo para cada una de las provincias.
  - ✚ Nombre (varchar(20)): Clave secundaria. Nombre de cada una de las provincias.
  - ✚ Longitud (double): Coordenada geográfica longitud referente a una provincia.
  - ✚ Latitud (double): Coordenada geográfica latitud referente a una provincia.
  
- ✚ ISLA: Tabla donde se guarda la geolocalización de las Islas Canarias. Sus atributos son:
  - ✚ Código\_isla (varchar(3)): Clave primaria. Código significativo para cada una de las islas.
  - ✚ Nombre (varchar(15)): Clave secundaria. Nombre de cada una de las islas.
  - ✚ Provincia (varchar(3)): Provincia a la que pertenece una determinada isla. Clave ajena de PROVINCIA.Codigo\_provincia.
  - ✚ Longitud (double): Coordenada geográfica de longitud de una isla.
  - ✚ Latitud (double): Coordenada geográfica de latitud de una isla.
  
- ✚ CADENA: Tabla donde se guarda la información referente a las cadenas de mercados. Sus atributos son:
  - ✚ Id\_cadena (varchar(3)): Clave primaria. Código significativo de una cadena de mercados.
  - ✚ URL (varchar(50)): URL principal de la cadena de mercados.
  - ✚ Nombre (varchar(20)): Nombre de la cadena de mercados.
  - ✚ Mercadillo (boolean): Atributo con el que sabemos si la cadena de mercados es de un mercadillo o de un supermercado. Si está con valor true entonces es un mercadillo. Por defecto toma el valor false.
  
- ✚ MERCADO: Tabla donde se guarda la información referente a los mercados físicos que pertenecen a una cadena de mercados. , tanto de mercadillos del agricultor como de supermercados. Sus atributos son:

- ✚ Id\_mercado (varchar(5)): Clave primaria. Código significativo de un mercado.
  - ✚ Nombre (varchar(30)): Nombre que recibe un determinado mercado.
  - ✚ Dirección(vARCHAR(100)): Dirección física donde se encuentra situado un determinado mercado.
  - ✚ Cadena(vARCHAR(5)): Cadena de mercados a la que pertenece. Clave ajena de CADENA.Id\_cadena.
  - ✚ Provincia(vARCHAR(5)): Provincia en la que está situado un mercado. Clave ajena de PROVINCIA.Codigo\_provincia.
  - ✚ Isla(vARCHAR(5)): Isla en la que está situado un mercado. Puede tomar valores nulos. Clave ajena de ISLA.Codigo\_isla.
  - ✚ Longitud(float): Coordenada geográfica de longitud de un mercado.
  - ✚ Latitud(float): Coordenada geográfica de latitud de un mercado.
- ✚ CATEGORIA\_CADENA: Tabla donde se guarda la información referente a las categorías que existen para una cadena de mercados. Sus atributos son:
- ✚ Id\_categoria (varchar(3)): Clave primaria. Código significativo de una categoría.
  - ✚ Nombre (varchar(20)): Nombre de la categoría a la que pertenecen los productos.
  - ✚ Cadena (varchar(3)): Clave primaria. Cadena de mercados a la que pertenece la categoría. Clave ajena de CADENA.Id\_cadena.
  - ✚ Nivel (int): Nivel, en la página web, donde se encuentra la categoría. El nivel más alto tomará el valor 0.
  - ✚ Categoria\_padre (varchar(3)): Código significativo de la categoría padre.
  - ✚ Cadena\_padre (varchar(3)): Cadena de mercados donde se encuentra la categoría padre.
  - ✚ Categoria\_propia (varchar(3) ): Nombre identificativo que recibe una categoría. Clave ajena de CATEGORIA\_PROPIA.Id\_sigla.
- ✚ CATEGORIA\_PROPIA: Tabla donde se guarda la información referente a las categorías definidas a partir de las categorías que existen para una cadena de mercados. Sus atributos son:
- ✚ Id\_sigla (varchar(3)): Clave primaria. Código significativo de una categoría propia.
  - ✚ Nombre (varchar(20)): Nombre de la categoría propia a la que pertenecen los productos.
  - ✚ Nivel (int): Nivel, en la página web, donde se encuentra la categoría propia. El nivel más alto tomará el valor 0.



- ✚ Categoría\_padre (varchar(3)): Código significativo de la categoría propia padre.
  
- ✚ PRODUCTO: Tabla donde se guarda la información referente a los productos que se venden en los mercados atendiendo a la categoría. Sus atributos son:
  - ✚ Id\_producto (varchar(50)): Clave primaria. Código significativo de un producto. Se conformará de una parte del nombre de la cadena a la que pertenece, otra parte del supermercado de dicha cadena y por último el nombre del producto.
  - ✚ Nombre\_web (varchar(50)): Nombre que recibe el producto en la página web desde la que se extrae.
  - ✚ Marca (varchar(30)): Marca del producto. Los productos que se encuentran en los mercadillos no tienen marca, por lo tanto, este atributo puede tomar valores nulos.
  - ✚ Precio (double): Precio correspondiente a una determinado producto. Para los productos de los mercadillos será el precio medio entre el precio mínimo y el precio máximo de dicho producto, valores que se extraen de la tabla PRODUCTO\_MERCADILLO.
  - ✚ Categoría (varchar(3)): Categoría en la cadena a la que pertenece el producto.
  - ✚ Cadena (varchar(3)): Cadena de mercados a la que pertenece el producto. La clave ajena será la concatenación de los atributos CADENA.Id\_cadena y CATEGORIA\_CADENA.Id\_categoria.
  - ✚ Nombre\_producto (varchar(50)): Nombre asignado a un determinado producto. Será el nombre que se mostrará en la app móvil. Este atributo concatenado con el atributo 'Cadena' conforman la clave secundaria de la tabla.
  - ✚ Cantidad(double): Cantidad del producto que se vende al precio correspondiente. Puede tomar valores nulos.
  - ✚ Medida varchar(10): Medida en la que se oferta el producto (litros, gramos, kilos, ...) Puede tomar valores nulos si se desconoce.
  - ✚ Producto\_canario (boolean): Atributo con el que sabemos si un producto pertenece a las islas canarias o no. Por defecto toma el valor "no".
  - ✚ Activo (boolean): Atributo con el que sabemos si un producto está en el mercado en una determinada fecha o no. Por defecto toma el valor "si".
  - ✚ Fecha\_modificación (date): Fecha en la que se actualiza la información de un determinado producto.
  
- ✚ PRODUCTO\_MERCADILLO: Tabla donde se guarda la información referente a los productos que se venden en los mercadillos. En esta tabla se almacena el histórico de los precios de los productos que se ofertan en los mercadillos. Sus atributos son:
  - ✚ Id\_producto (varchar(50)): Clave primaria. Código significativo de un producto. Clave ajena de PRODUCTO.Id\_producto.

- ✚ Fecha (date): Clave primaria. Fecha en la que comienza una determinada tanda de precios para los productos del mercadillo.
- ✚ Precio\_minimo (double): Precio mínimo del rango de precios que disponen los puestos que ofertan el producto dentro de un determinado mercadillo.
- ✚ Precio\_maximo (double): Precio máximo del rango de precios que disponen los puestos que ofertan el producto dentro de un determinado mercadillo.

## 2. Código de la aplicación de extracción de datos

🚧 Código para la extracción del código HTML de las páginas webs, usando CURL

```
function extraer_html($categoria){
    $curl =
curl_init("http://www.mercadillodelagricultor.com/precios/Precios".$categoria
.".htm");
    switch($categoria){
        case "A":
            $fichero = fopen("../Aromaticas.txt", "w");
            break;
        case "B":
            $fichero = fopen("../Fruta.txt", "w");
            break;
        case "D":
            $fichero = fopen("../Verduras.txt", "w");
            break;
        case "P":
            $fichero = fopen("../Papas.txt", "w");
            break;
    }
    curl_setopt($curl, CURLOPT_FILE, $fichero);
    curl_setopt($curl, CURLOPT_HEADER, 0); // opciones para transf de datos
    curl_exec($curl);
    curl_close($curl);
    fclose($fichero);
}
```

🚧 Código donde se utiliza la codificación geográfica de Google Maps.

```
function buscar_coordenadas(direccion_resolver, tipo_punto){
    geocoder = new google.maps.Geocoder();
    var direccion = direccion_resolver;
    geocoder.geocode({'address': direccion}, function(results, status) {
        var latitud;
        var longitud;
        if (status == google.maps.GeocoderStatus.OK) {
            latitud = results[0].geometry.location.lat();
            longitud = results[0].geometry.location.lng();
            $.post('../bbdd/bbdd2.php', {lat: latitud, lng: longitud,
            dir:direccion, accion:"insertar", tipo:tipo_punto},
            function(datos){
                $(".hero-unit").append(''); }
            ).done(function(){
                llamadas++;
                if (llamadas == puntos)
                    actualizar_vista_2();
            }
        }
    });
}
```

```
        });
    }
    else if(status == google.maps.GeocoderStatus.OVER_QUERY_LIMIT) {
        setTimeout(function() {
            buscar_coordenadas(direccion_resolver, tipo_punto);
        }, 400);
    }
    else
        llamadas++;
});
}
```

🚧 Código donde se muestran los mapas, utilizando la API de Google Maps para PHP

```
$gmap = new GoogleMapAPI();
$array_resultado = array(); $i = 0;
if(isset($_GET['lugar']) && !empty($_GET['lugar'])){
    $lugar = $_GET['lugar'];
    if ($lugar == 'canarias')
        $array_resultado = mostrar_puntos_islas();
    else if ($lugar == 'espana')
        $array_resultado = mostrar_puntos_provincias();
    else if ($lugar == 'mercado')
        $array_resultado = mostrar_puntos_mercados();
    while ($i < count($array_resultado)) {
        $punto =$array_resultado[$i];
        $lng =$array_resultado[$i+1];
        $lat =$array_resultado[$i+2];
        $gmap->addMarkerByCoords($lng, $lat, $punto);
        $i =$i + 3;
    }
}
echo $gmap->getHeaderJS();
echo $gmap->getMapJS();
echo $gmap->printOnLoad();
echo $gmap->PrintMap();
```

### 3. Código de la aplicación móvil

🚧 Código para hacer consultas a la base de datos, utilizando AsyncTask

```
public class ConsultaMercados extends AsyncTask<Object,Void,JSONArray> {

@Override
protected JSONArray doInBackground(Object... params) {
    JSONArray Mercado_Seleccionado;
    String lugar = params[0].toString();
    String categoria = params[1].toString();
    Boolean provincia = (Boolean)params[2];

    HttpClient httpClient = new DefaultHttpClient();

    HttpGet del = new
        HttpGet("http://192.168.1.37:2731/API/Mercados?lugar="+lu
            gar+"&categoria="+categoria+"&provincia="+provincia);
    del.setHeader("content-type", "application/json");
    try
    {
        HttpResponse resp = httpClient.execute(del);
        String respStr = EntityUtils.toString(resp.getEntity());
        Mercado_Seleccionado = new JSONArray(respStr);
    }
    catch(Exception ex)
    {
        Log.e("ServicioRest","Error!", ex);
        Mercado_Seleccionado = new JSONArray();
    }

    return Mercado_Seleccionado;
}

ConsultaMercados consulta2 = new ConsultaMercados(){
    @Override
    protected void onPostExecute(JSONArray result){
        IncluirMarcadores(result);
    }
};
consulta2.execute(IslaSeleccionada, CategoriaSeleccionada, false);
```

🚧 Código correspondiente a la creación de mapas, con la API de Google.

```
LatLng ProvinciaSeleccionadaMapa = new LatLng(lat,lng);
GoogleMap mapa
    =((SupportMapFragment)getSupportFragmentManager().findFragmentBy
        Id(R.id.MapaSupermercados)).getMap();
CameraPosition centrarmapa = new CameraPosition.Builder().
    target(ProvinciaSeleccionadaMapa).zoom(7).build();
CameraUpdate inicializar = CameraUpdateFactory.
    newCameraPosition(centrarmapa);
mapa.animateCamera(inicializar);
```

🚧 Código correspondiente al Servicio Web tipo REST.

```
private static string cadenaConexion = @"Data Source=ADRIANA-PC;Initial
Catalog=Cesta_Compra;Integrated Security=True";
```

```

public List<Mercados> ObtenerMercados(string lugar, string categoria, bool
provincia){
    List<Mercados> lista = new List<Mercados>();
    SqlConnection con = new SqlConnection(cadenaConexion);
    con.Open();
    string sql = "";
    if (!provincia) {
        if (categoria == "Todas las categorias")
            sql = "SELECT
m.Id_Mercado,m.Nombre,m.Direccion,m.Longitud,m.Latitud, sum(p.Precio) as Suma
FROM mercado m INNER JOIN producto p ON (m.cadena = p.cadena) INNER JOIN islas i
ON (m.Isla = i.Codigo_isla) WHERE (i.Nombre = @lugar) GROUP BY m.Id_Mercado,
m.Nombre,m.Direccion,m.Longitud,m.Latitud ORDER BY Suma";
        else
            sql = "SELECT m.Id_Mercado, m.Nombre, m.Direccion,
m.Longitud, m.Latitud, sum(p.Precio) as Suma FROM mercado m INNER JOIN producto p
ON (m.cadena = p.cadena) INNER JOIN categoria_cadena c ON (m.Cadena = c.Cadena)
INNER JOIN islas i ON (m.Isla = i.Codigo_isla) WHERE ((i.Nombre = @lugar) AND
(c.Nombre = @categoria)) GROUP BY m.Id_Mercado,
m.Nombre,m.Direccion,m.Longitud,m.Latitud ORDER BY Suma";
    }
    else{
        if (categoria == "Todas las categorias")
            sql = "SELECT
m.Id_Mercado,m.Nombre,m.Direccion,m.Longitud,m.Latitud, sum(p.Precio) as Suma
FROM mercado m INNER JOIN producto p ON (m.cadena = p.cadena) INNER JOIN
provincias i ON (m.Provincia = i.Codigo_provincia) WHERE (i.Nombre = @lugar)
GROUP BY m.Id_Mercado, m.Nombre,m.Direccion,m.Longitud,m.Latitud ORDER BY Suma";
        else
            sql = "SELECT m.Id_Mercado, m.Nombre, m.Direccion,
m.Longitud, m.Latitud, sum(p.Precio) as Suma FROM mercado m INNER JOIN producto p
ON (m.cadena = p.cadena) INNER JOIN categoria_cadena c ON (m.Cadena = c.Cadena)
INNER JOIN provincias i ON (m.Provincia = i.Codigo_provincia) WHERE ((i.Nombre =
@lugar) AND (c.Nombre = @categoria)) GROUP BY m.Id_Mercado,
m.Nombre,m.Direccion,m.Longitud,m.Latitud ORDER BY Suma";
    }
    SqlCommand cmd = new SqlCommand(sql, con);
    cmd.Parameters.Add("@lugar", System.Data.SqlDbType.NVarChar).Value= lugar;
    cmd.Parameters.Add("@categoria", System.Data.SqlDbType.NVarChar).Value =
        categoria;

    SqlDataReader reader =
        cmd.ExecuteReader(System.Data.CommandBehavior.CloseConnection);
    while (reader.Read()){
        Mercados mer = new Mercados();
        mer.ID = reader.GetString(0);
        mer.Nombre = reader.GetString(1);
        mer.Direccion = reader.GetString(2);
        mer.Longitud = reader.GetDouble(3);
        mer.Latitud = reader.GetDouble(4);
        mer.Precio = reader.GetDouble(5);
        lista.Add(mer);
    }
    reader.Close();
    return lista
}

[HttpGet] //...API/Mercados
public JsonResult Mercados(string lugar, string categoria, bool provincia){
    return Json(ObjectManager.ObtenerMercados(lugar, categoria, provincia),
        JsonRequestBehavior.AllowGet);
}

```