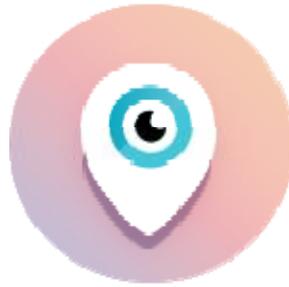




Projecte final de carrera

Programació d'aplicacions per a mobils usant HTML5





1. AGRADECIMIENTOS (página 3)
2. INTRODUCCIÓN (página 4)
3. OBJETIVOS
 - a. OBJETIVOS GENERALES (página 5)
 - b. OBJETIVOS ESPECÍFICOS (página 5)
4. PLANIFICACIÓN
 - a. DIAGRAMA DE GANTT (página 7)
 - b. RECURSOS (página 8)
 - c. FACTIBILIDAD (página 8)
5. DISEÑO CONCEPTUAL
 - a. DIAGRAMAS DE FLUJO DE DATOS. DFD (página 9)
 - i. DIAGRAMA DE CONTEXTO (nivel 0)
 - ii. DIAGRAMA DE NIVEL SUPERIOR (nivel 1)
 - iii. DIAGRAMA DE DETALLE (nivel 2)
 - b. DIAGRAMAS DE SECUENCIA
6. DISEÑO FÍSICO
 - a. BASE DE DATOS
 - i. MODELO ENTIDAD-RELACIÓN
 - ii. DISEÑO FÍSICO
 - b. DISEÑO DE INTERFACES
7. IMPLEMENTACION
8. MEJORAS FUTURAS
9. CONCLUSIONES
10. BIBLIOGRAFÍA



Agradecimientos

Me gustaría, antes de presentar este proyecto, agradecer a las personas que me han ayudado en la realización del mismo.

La primera persona que merece mi agradecimiento es mi mujer. Teniendo dos niños pequeños, uno de ellos un bebé de cuatro meses, ha hecho lo imposible para que pudiera disponer del suficiente tiempo libre para finalizar este proyecto.

Después me gustaría agradecer a mi familia que también ha puesto a mi disposición su tiempo libre para hacerse cargo de mis hijos cuando lo he necesitado.

Agradecer también a mi tutor Carlos que me haya permitido realizar las entregas casi siempre fuera de tiempo, su atención inmediata y sus palabras de ánimo constantes.

Muchas gracias.



Introducción

Ya no nos podemos sorprender de lo que puede hacer un móvil, ni tampoco, de lo que pueden hacer las aplicaciones que en él podemos instalar y aún así, a veces, tengo la sensación que es un territorio todavía por explorar y que aún podemos tener la suerte de crear una aplicación con la capacidad de sorprender o que pueda ser de utilidad en un mundo con millones de aplicaciones disponibles y listas para instalar a golpe de dedo.

Me pareció genial el día que descubrí que se podían realizar aplicaciones móviles multiplataforma con un lenguaje sencillo y universal como el HTML5 y qué podíamos también utilizar javascript y css para mejorar el estilo, usabilidad y comportamiento.

Por estos motivos he elegido realizar este proyecto tomando como base tecnológica la que proviene del mundo web y realizar una aplicación móvil que intente cubrir algún hueco libre.

Por mi experiencia laboral sé que el mundo de la pequeña y mediana empresa, en cuanto a tecnología se refiere, es un territorio por explorar y explotar. Crear una pequeña aplicación que les pueda dar a conocer o hacerlas más visibles en un mundo de grandes multinacionales, es un bonito objetivo para este proyecto.



Objetivos Generales

Mi proyecto será una aplicación móvil que permita buscar y localizar ofertas online. Estas ofertas serán publicadas por los comercios adscritos al proyecto.

La idea básica es no tener que encontrar la oferta mientras paseas un día por el centro si no que puedas buscar, a través de una aplicación móvil, dónde se encuentran esas ofertas.

El funcionamiento será sencillo. Los comercios adscritos podrán publicar las ofertas más interesantes de sus comercios, indicando tipología de oferta que publica, datos de la empresa y la localización geoespacial de la misma.

Los usuarios, por su parte, podrán buscar o filtrar por tipo de oferta según sus gustos, necesidades o por la proximidad. La aplicación mostrará información de la oferta, del comercio y también, en un mapa, la posición del usuario y del comercio.

Los comercios adscritos en este proyecto deberán darse de alta para informar de las ofertas que tienen, lo que duran y las condiciones de éstas. La intención será buscar empresas que quieran ofertar en la aplicación sus productos y/o servicios en promoción y, para conseguirlo, sólo tendrán que pagar una pequeña cuota mensual.

De la misma forma, espero poder llegar al mayor número posible de empresas que quieran potenciar la visualización de sus ofertas. Para el comercio el desembolso será bastante pequeño, sin embargo, la oferta la podrá ver mucha gente sin necesidad de pasar por allí.

Para los usuarios que quieran descargarse la aplicación, ésta será gratuita.

Objetivos específicos

Los objetivos específicos serán aquellas funcionalidades y requisitos de la aplicación que deben cumplir para que ésta sea útil, funcione correctamente y sea duradera en el tiempo.

Se va a diseñar y programar una aplicación móvil que ofrecerá al usuario buscar, localizar y ubicar los comercios con ofertas más cercanas a su posición.

La aplicación se compilará con PhoneGap Build, será multiplataforma y permitirá que se instale en dispositivos Android e IOS.

La aplicación permitirá, utilizando la tecnología de ubicación gps y móvil (a través de datos), localizar las ofertas más cercanas al usuario a través de un entorno fácil, intuitivo y agradable.

Se podrán realizar búsquedas filtradas, pudiendo buscar por provincia, población, cercanía, categoría y sub-categoría de entre todas las ofertas. Estas búsquedas serán auto-excluyentes por lo que no se producirán búsquedas vacías.



La aplicación permitirá parametrizar el radio de búsqueda de la ofertas, centrándose en aquellas que realmente están cerca del usuario.

Una vez realizado el filtrado las ofertas se mostrarán en un listado resumido, con la información básica: Nombre del comercio, descripción resumida de la oferta y foto en miniatura. Los usuarios podrán ampliar esta información al pulsar con el dedo sobre la oferta y pudiendo acceder al detalle de la misma.

El listado de ofertas será de una longitud parametrizable por el usuario que podrá elegir el número de ofertas que quiere visualizar de una sola vez. Aunque no se muestren todas las ofertas a la vez el usuario podrá cargar más ofertas al desplazarse hacia abajo.

El detalle de la oferta mostrará la información más detallada de la oferta junto con una imagen representativa. Esta información estará compuesta por una descripción larga de la oferta y las fechas que delimitarán la duración de la misma.

Junto con la información detallada de la oferta, los usuarios podrán elegir de entre varias opciones:

ver más información del comercio

llamar al comercio.

situar el comercio en el mapa

compartir la información de la oferta en las redes sociales.

La aplicación permitirá mostrar toda la información del comercio que será accesible desde el detalle de la oferta. Ésta información se compondrá de una imagen del comercio (de su fachada), su dirección, teléfono y horarios. Desde esta nueva ventana ofreceremos de nuevo la posibilidad de **llamar al comercio, situar el comercio en el mapa y compartir la información del comercio en las redes sociales.**

La aplicación permitirá situar la ubicación de la oferta/comercio en el mapa y mostrar también la posición del usuario, con el fin de que el éste pueda ver lo cerca que se encuentra de la oferta.

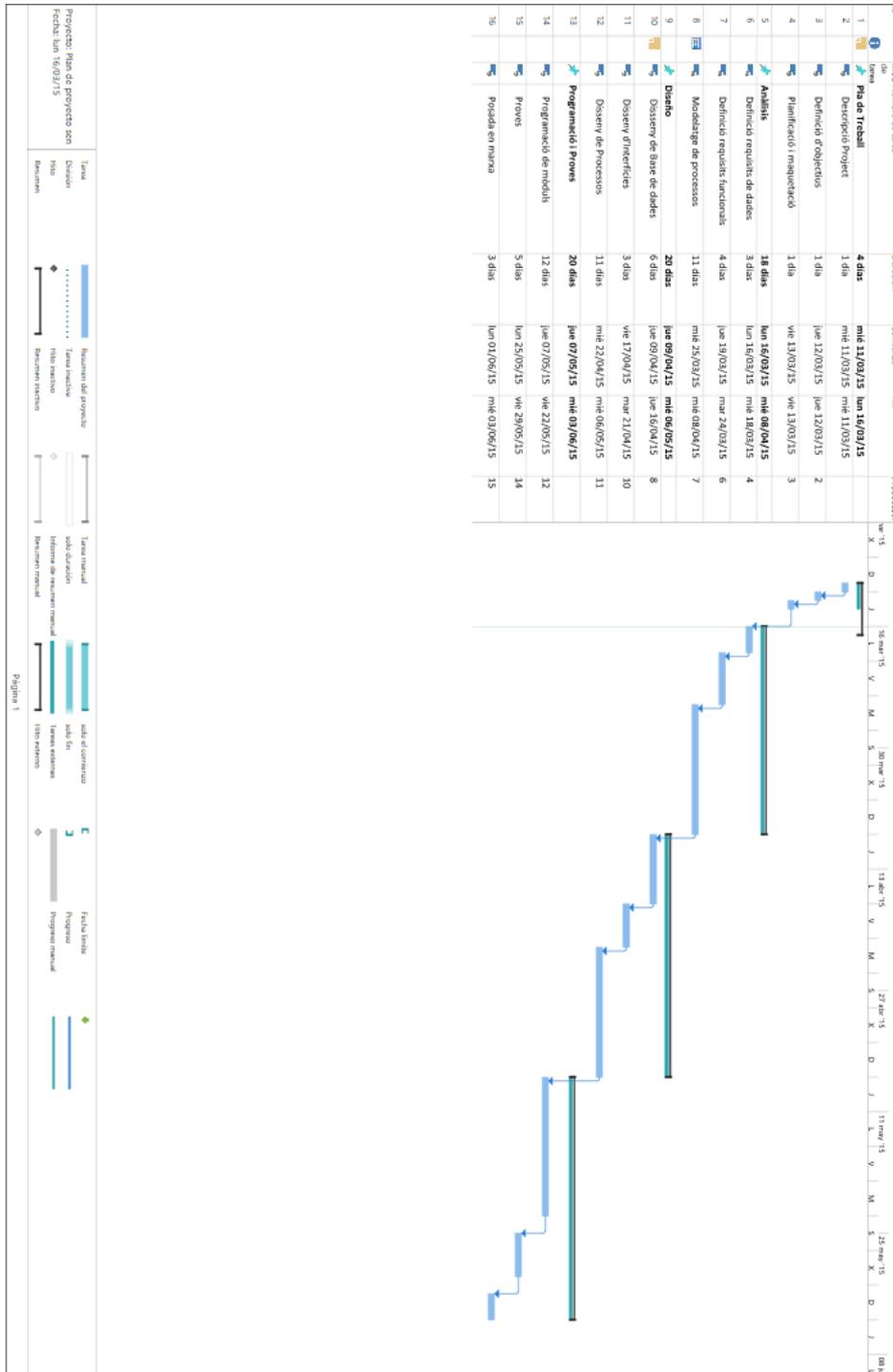
De la misma forma se podrá compartir a través de las redes sociales la información detallada de ofertas y/o comercios.

La aplicación también permitirá modificar el radio de búsqueda y el número de ofertas a mostrar dicha información quedará almacenada en un BBDD interna, para que esté disponible es posteriores búsquedas, creando así un perfil.



Diagrama de Gantt

En el siguiente diagrama se puede observar de qué forma se ha planificado el tiempo para cada uno de los hitos y para cada una de las tareas que éste incluye.





Recursos

Los recursos necesarios para realizar el proyecto serán los siguientes.

HARDAWE: Se utilizará un portátil con un sistema operativo Windows 8, con 4 GB de RAM y un procesador de dos núcleos y 3GHz.

SOFTWARE: Para la realización del proyecto se utilizará un servidor apache y una Base de Datos mysql. El servidor php se utilizará para probar todos los servicios web en local y la mysql para crear y gestionar la Base de datos. Se instalará **WAMP**.

También utilizaré **AptanaStudio** como editor de HTML, javascript y PHP.

Para generar los instalables (apk) para android utilizaré la plataforma web **Adobe Phone Gap Build** (<https://build.phonegap.com>)

Para poder probar el proyecto sin necesidad de generar un apk también precisaré del complemento **Ripple** para Chrome, que me permitirá ver el proyecto en distintos formatos de móvil, lanzar eventos como si de un teléfono se tratara e incluso establecer los parámetros de posición.

MÓBIL: Cualquier móvil con android nos servirá.

RED: Se precisará una conexión por cable a internet

Factibilidad

A continuación se muestra un pequeño estudio de la factibilidad del proyecto.

Primero mencionar que al ser un proyecto de estudiante para la universidad no trataremos en este apartado de la factibilidad económica.

Factibilidad técnica: el proyecto es técnicamente viable ya que disponemos de todos los elementos necesarios para llevar el proyecto a término. El software utilizado está disponible y al ser de código abierto, también es gratuito.

Los elementos de hardware utilizados y necesarios son de características técnicas nada exigentes, con cualquier ordenador se podría realizar.

Factibilidad operativa: el proyecto es viable. Dispongo de los conocimientos necesarios para finalizar el proyecto. Los conocimientos recogidos durante los años de universidad y la experiencia por los años de trabajo en distintas empresas privadas y como freelance realizando proyectos web.



Diseño conceptual

En este apartado se desarrolla el modelado de procesos, se utilizan Diagramas de Flujo de Datos con el objetivo de visualizar de la mejor manera posible el procesamiento de datos y de procesos. Se generarán DFD a varios niveles de abstracción, detallando procesos concretos que aparecen como elementos simples en DFD's de nivel superior.

Diagrama de flujo de datos. DFD

Con el diagrama de flujo de datos presentado a continuación representa de una forma gráfica la interacción del sistema de información y las entidades externas del sistema y en las sucesivas “explosiones” del mismo la interacción entre los distintos subsistemas

Diagrama de contexto (nivel 0)

El primer diagrama (contexto) muestra como los usuarios de la aplicación se relacionan con el sistema y como la aplicación móvil interactúa con el servidor web y con el móvil.

Existirán dos tipos de usuarios. Los primeros serán los que utilizarán la aplicación móvil y para los que el resto del sistema será invisible. También existirá la figura del administrador, que mantendrá y cargará la Base de datos.

La comunicación entre la aplicación móvil y la Base de datos se realizará siempre desde una conexión a Internet. La información que mostrará la aplicación será totalmente dinámica y dependiente del servidor web.

El móvil, mostrado como un periférico, se encargará de suministrar a la aplicación los datos de la posición geoespacial, nos permitirá también compartir la información de las ofertas a través de las redes sociales y de realizar llamadas.

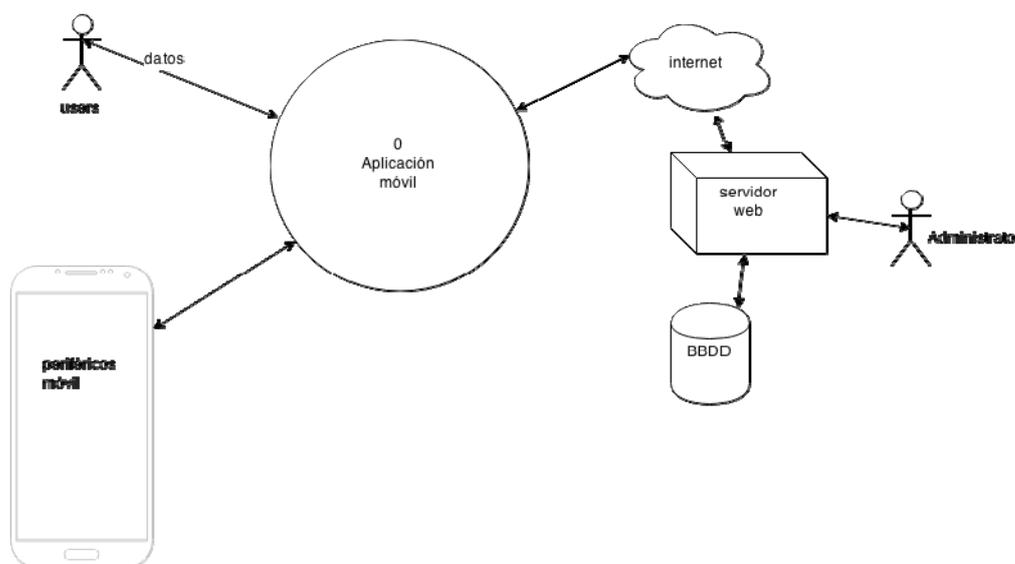




Diagrama de nivel superior (nivel 1)

En este diagrama se muestra la primera explosión del diagrama de contexto.

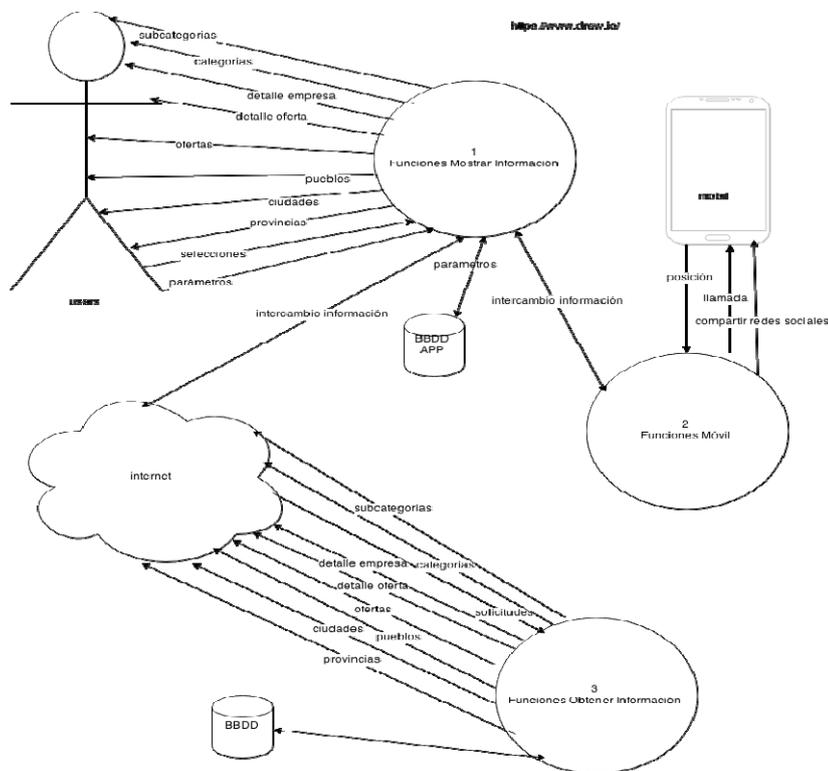
Se ha creído necesario y lógico dividirlo en tres subsistemas. El primer subsistema interactuará con los usuarios finales, el segundo con el móvil y el tercero con la información de la BBDD.

En un primer nivel (funciones Mostrar Información) se representan las funciones que permiten mostrar la información al usuario y que también se recogen las interacciones de éste con el sistema. Este primer sistema es el **interfaz** de la aplicación con el usuario. Cada vez que el usuario realice una acción saltará un evento y el sistema se encargará de solicitar la información al resto de subsistemas y también de mostrarla adecuadamente.

No se ha creído conveniente “explotar” el primer subsistema en distintas funciones y que el mecanismo de todas ellas será siempre similar.

El subsistema 2 (funciones Móvil), es el que permite interactuar con el móvil como dispositivo y que se encargará de obtener la información de la posición geoespacial, solicitar una llamada o que permitirá compartir la información con las distintas redes sociales instaladas.

El tercer subsistema (funciones para obtener información), será el encargado de proveer la información solicitada por el subsistema 1. La intención es que este sistema envíe la información ya procesada (como código HTML) y que el subsistema 1 sólo tenga que incrustarla en el código. El motivo de esta decisión es evitar cargar al primer subsistema de ésta carga de trabajo.

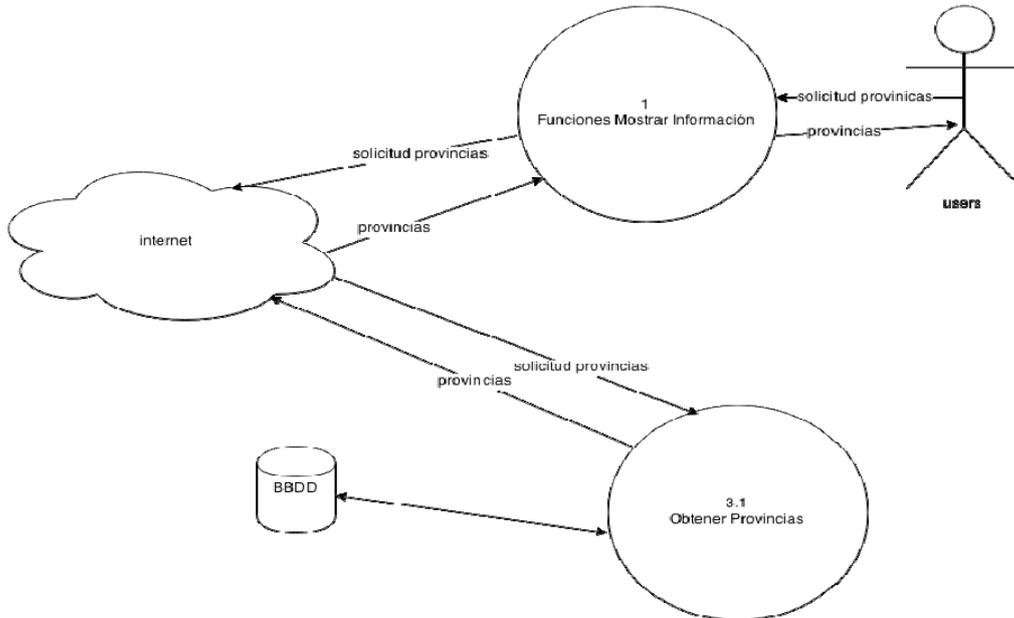




Diagramas de detalle (nivel 2)

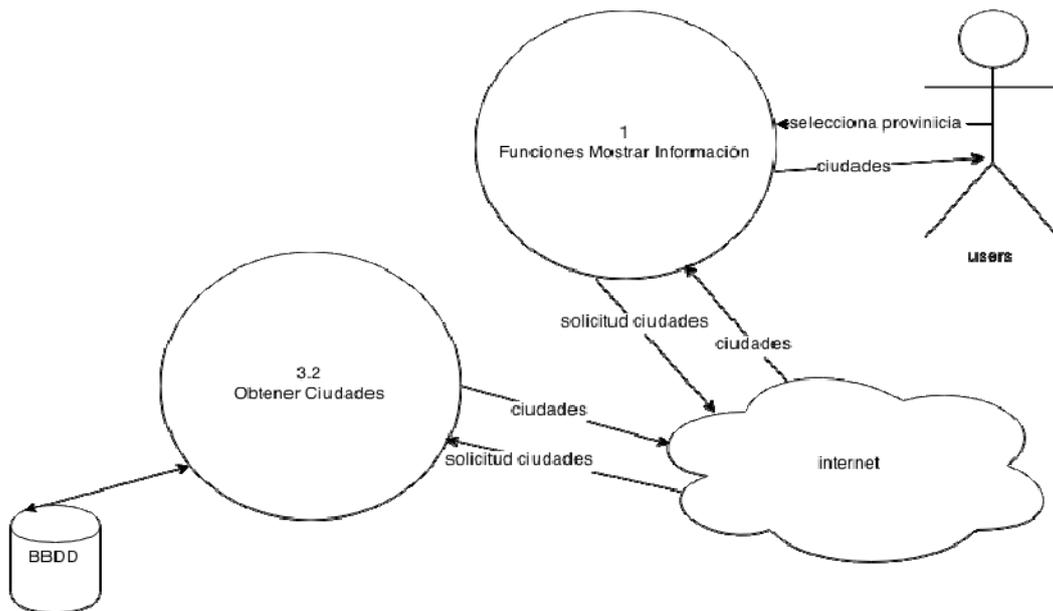
A continuación muestro los diagramas que surgirían de la explosión del subsistema 3

Adjunto a cada diagrama una muestra en pseudocódigo lo que podría ser la consulta que obtuviera los datos de la BBDD y los presentara al subsistema 1 como código HTML.



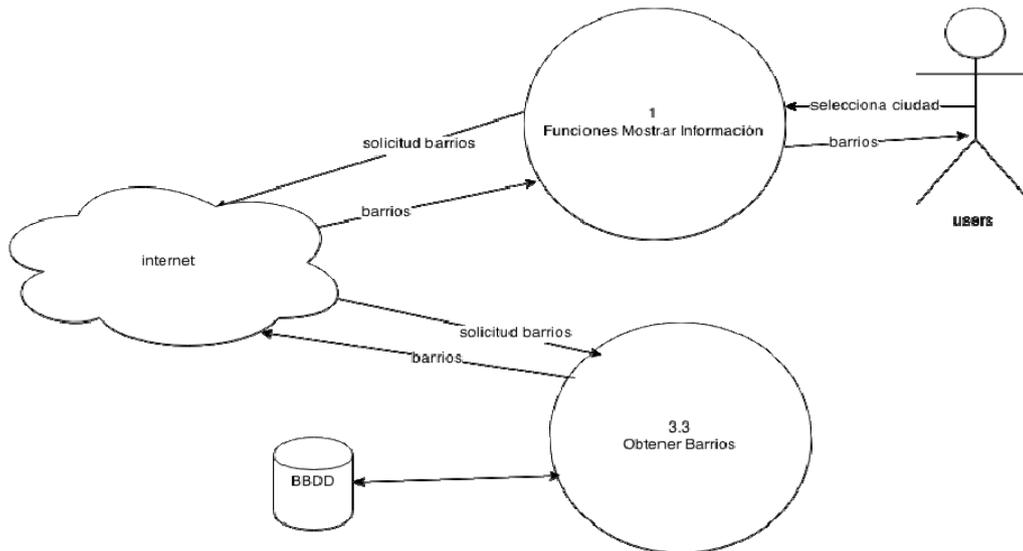
Ejemplo de Función en servidor web, con código php.
La función sin parámetros devuelve un select con todas las provincias.
Se devuelve un select en código HTML utilizando json que será recogido e incrustado en el código HTML de la app.

```
$select = "<select name='provincias'>"
$query = "SELECT provincia FROM provincia."
$res = mysql_query($query);
if ($res)
{
    $select = $select . "<option value='>'>SIN PROVINCIAS</option>";
}
else
{
    $select = $select . "<option value='ALL'> - TODAS LAS PROVINCIAS - </option>";
    while($fila = mysql_fetch_row($res))
    {
        $select = $select . "<option value='".$fila[0]."'>".$fila[0]."</option>";
    }
    mysql_free_result($res);
}
$select = $select . "</select>";
```



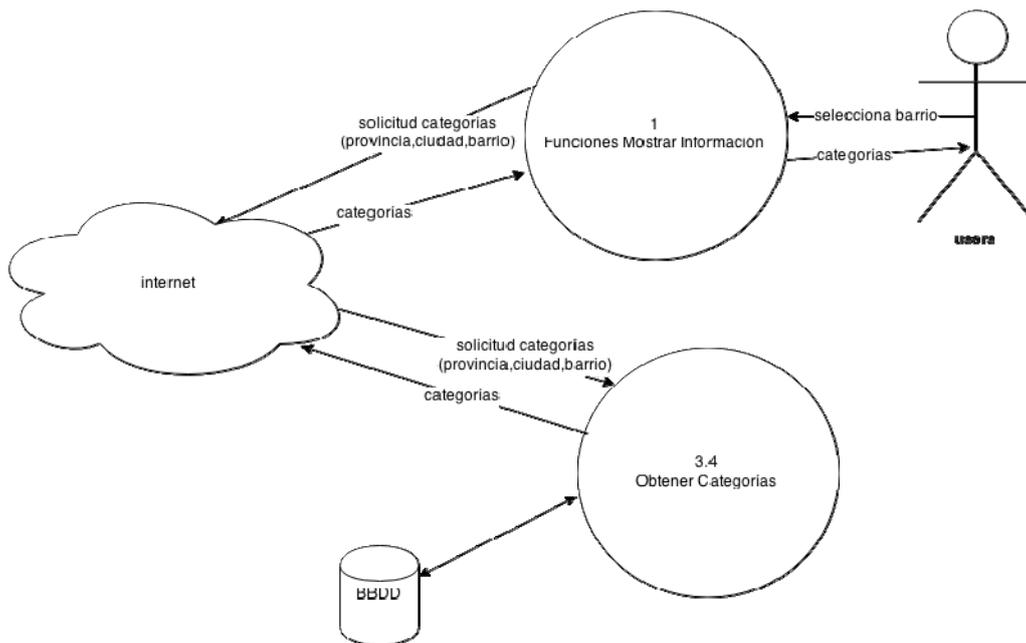
Ejemplo de Función en servidor web, con código php.
La función con un parámetro(provincia) devuelve un select con todas las ciudades.
Se devuelve un select en código HTML utilizando json que será recogido e incrustado en el código HTML de la app.

```
$select=<select name='ciudades'>.  
  
$query = "SELECT ciudad FROM ciudad where provincia='".$provincia."'".  
$res = mysql_query($query).  
if (!$res)  
{  
    $select=$select.<option value='>--SIN CIUDADES--</option>".  
}  
else  
{  
    $select=$select.<option value='ALL'> - TODAS LAS CIUDADES - </option>".  
    while($fila = mysql_fetch_row($res))  
    {  
        $select=$select.<option value='".$fila[0].">".$fila[0].</option>".  
    }  
    mysql_free_result($res).  
}  
$select=$select.</select>".
```



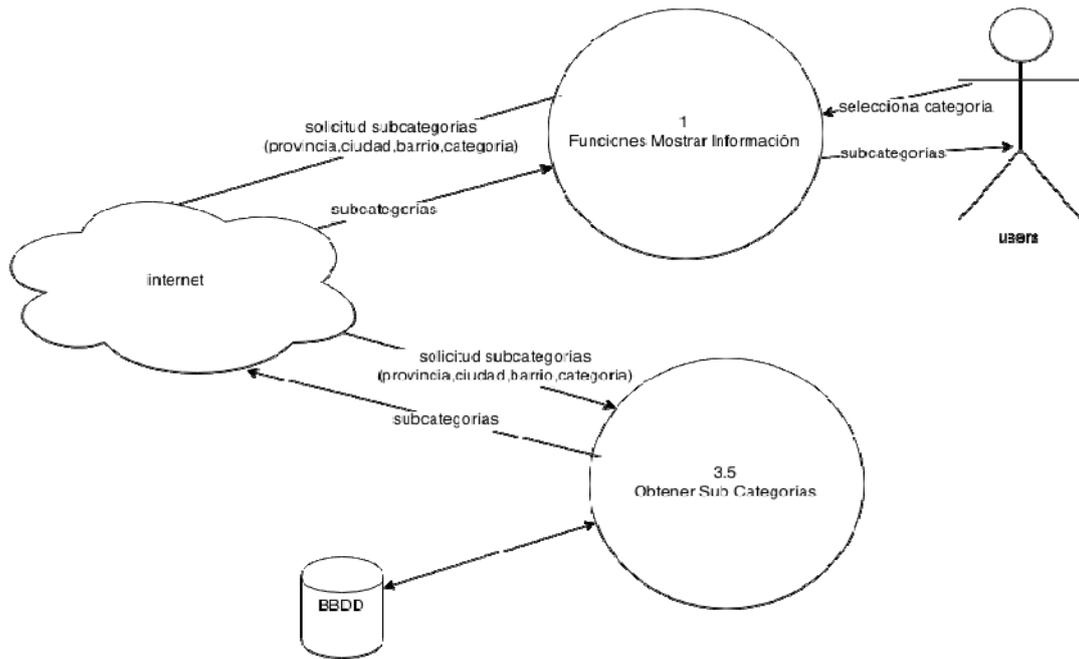
Ejemplo de función en servidor web, con código php.
La función con dos parámetros(provincia,ciudad) devuelve un select con todas los barrios.
Se devuelve un select en código HTML utilizando json que será recogido e incrustado en el código HTML de la app.

```
$query = "SELECT barrio FROM barrios where provincia=".$provincia." and ciudad=".$ciudad."";
$res = mysql_query($query);
if ($res)
{
    $select=$select."<option value='> TODOS</option>";
}
else
{
    $select=$select."<option value='ALL'> - TODOS LOS BARRIOS - </option>";
    while($fila = mysql_fetch_row($res))
    {
        $select=$select."<option value='".$fila[0].">".$fila[0]."</option>";
    }
    mysql_free_result($res);
}
$select=$select."</select>";
```



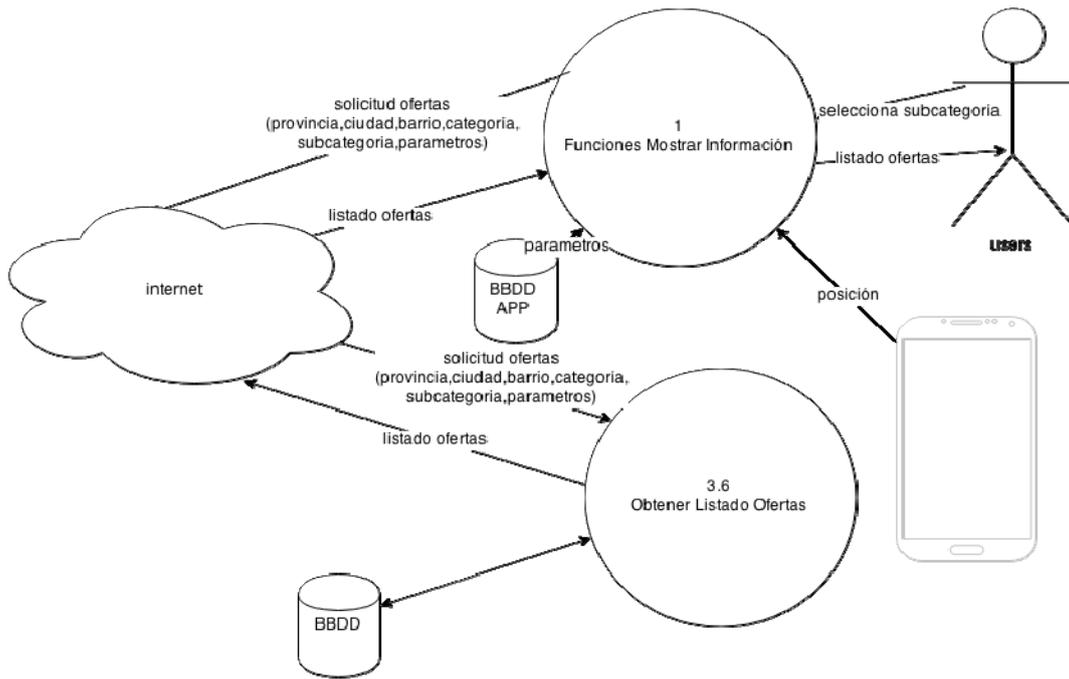
Ejemplo de función en servidor web, con código php.
La función con tres parámetros(provincia,ciudad,barrio) devuelve un table con todas categorías.
Se devuelve un table en código HTML, utilizando json que será recogido e incrustado en el código HTML de la app.

```
$table = <select name='categorias'>";
$query = "SELECT barrio FROM categorias where provincia='.$provincia.'" and ciudad='.$ciudad.'" and barrio='.$barrio.'"";
$result = mysql_query($query);
if (!$result)
{
    $table = $table . <option value="">--NINGUNA CATEGORIA--</option>";
}
else
{
    $table = $table . <option value="ALL">-- TODOS LAS CATEGORIAS--</option>";
    while($row = mysql_fetch_row($result))
    {
        $table = $table . <option value="$row[0]">.$row[0].</option>";
    }
    mysql_free_result($result);
}
$table = $table . </select>";
```



Ejemplo de función en servidor web con código php.
La función con cuatro parámetros (provincia, ciudad, barrio, categoría) devuelve un table con todas categorías.
Se devuelve un table en código HTML utilizando json que será recogido e incrustado en el código HTML de la app.

```
$table = "<select name='subcategorías'>";  
$query = "SELECT barrio FROM subcategorías where provincia='". $provincia ."' and ciudad='". $ciudad ."'  
and barrio='". $barrio ."' and categoría='". $categoría ."'";  
$res = mysql_query($query);  
if ($res)  
{  
    $table = $table . "<option value='> NINGUNA SUBCATEGORÍA </option>";  
}  
else  
{  
    $table = $table . "<option value='ALL'> - TODOS LAS SUBCATEGORÍAS - </option>";  
    while ($fila = mysql_fetch_row($res))  
    {  
        $table = $table . "<option value='". $fila[0] ."'> " . $fila[0] . "</option>";  
    }  
    mysql_free_result($res);  
}  
$table = $table . "</select>";
```

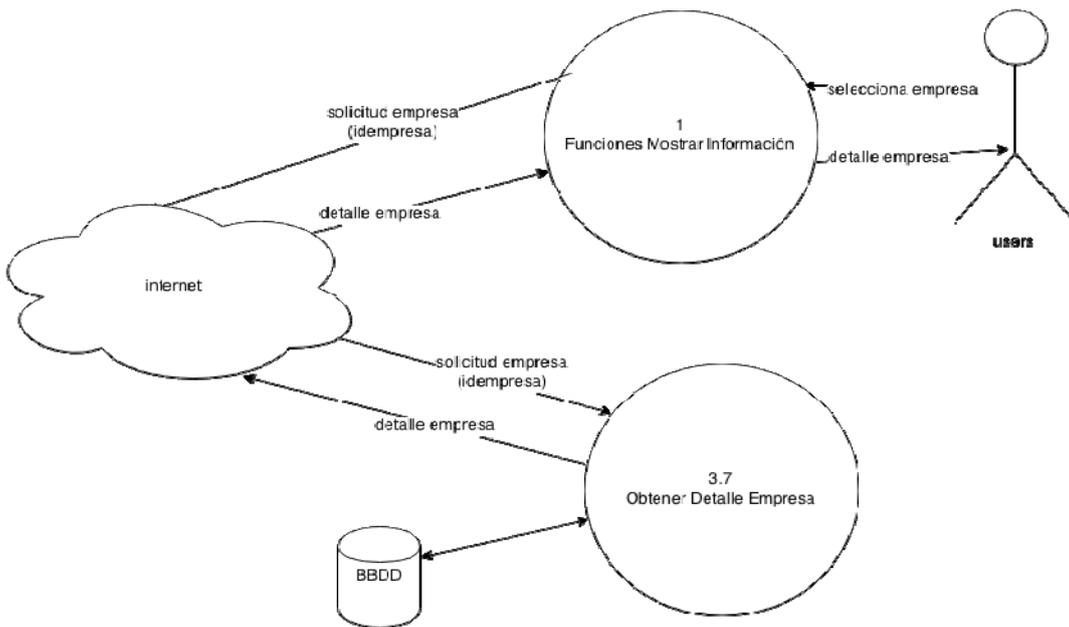
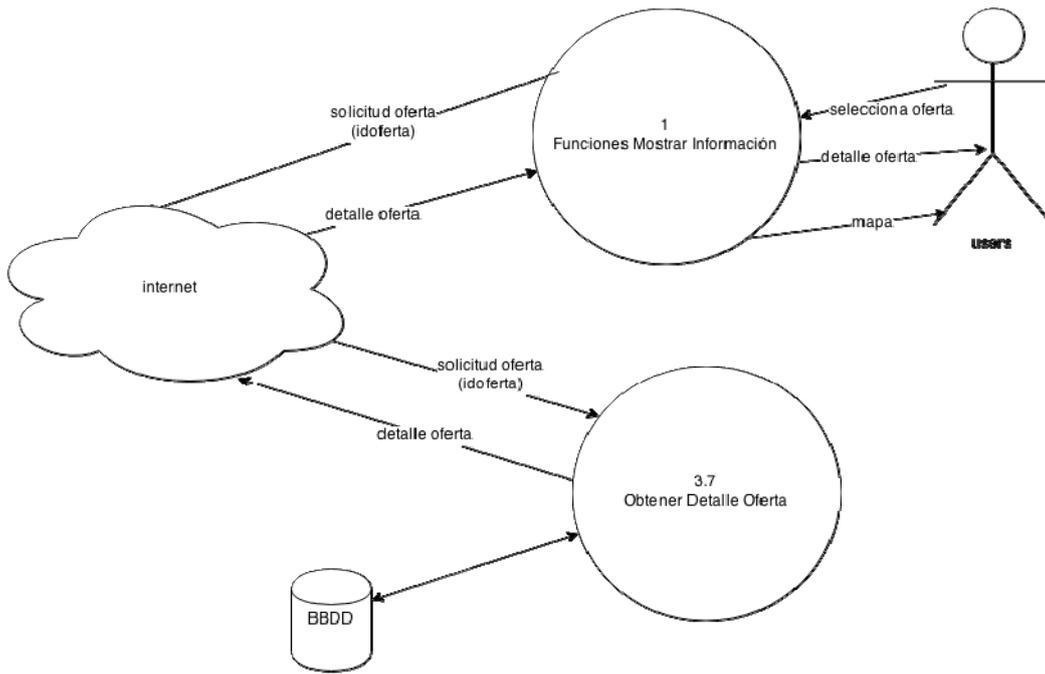


Ejemplo de función en servidor web, con código php.

La función con 8 parámetros (provincia, ciudad, barrio, categoría, subcategoría, radio búsqueda, max ofertas, posición) devuelve un table con todas categorías.

Se devuelve un table en código HTML utilizando |son que será recogido e incrustado en el código HTML de la app.

```
$table="<select name='ofertas'>";
$query = "SELECT * FROM ofertas where provincia='". $provincia. "' and ciudad='". $ciudad. "'
and barrio='". $barrio. "' and categoría='". $categoría. "' and subcategoría='". $subcategoría. "' and distancia<". $radio.
" limit 0, ". $maxofertas.
";
$res = mysql_query($query);
if ($res)
{
    $table=$table."<option value='>--NINGUNA OFERTA--</option>";
}
else
{
    while($fila = mysql_fetch_row($res))
    {
        $table=$table."<option value='". $fila[0]. "'>". $fila[0]. "</option>";
    }
    mysql_free_result($res);
}
$table=$table."</select>";
```

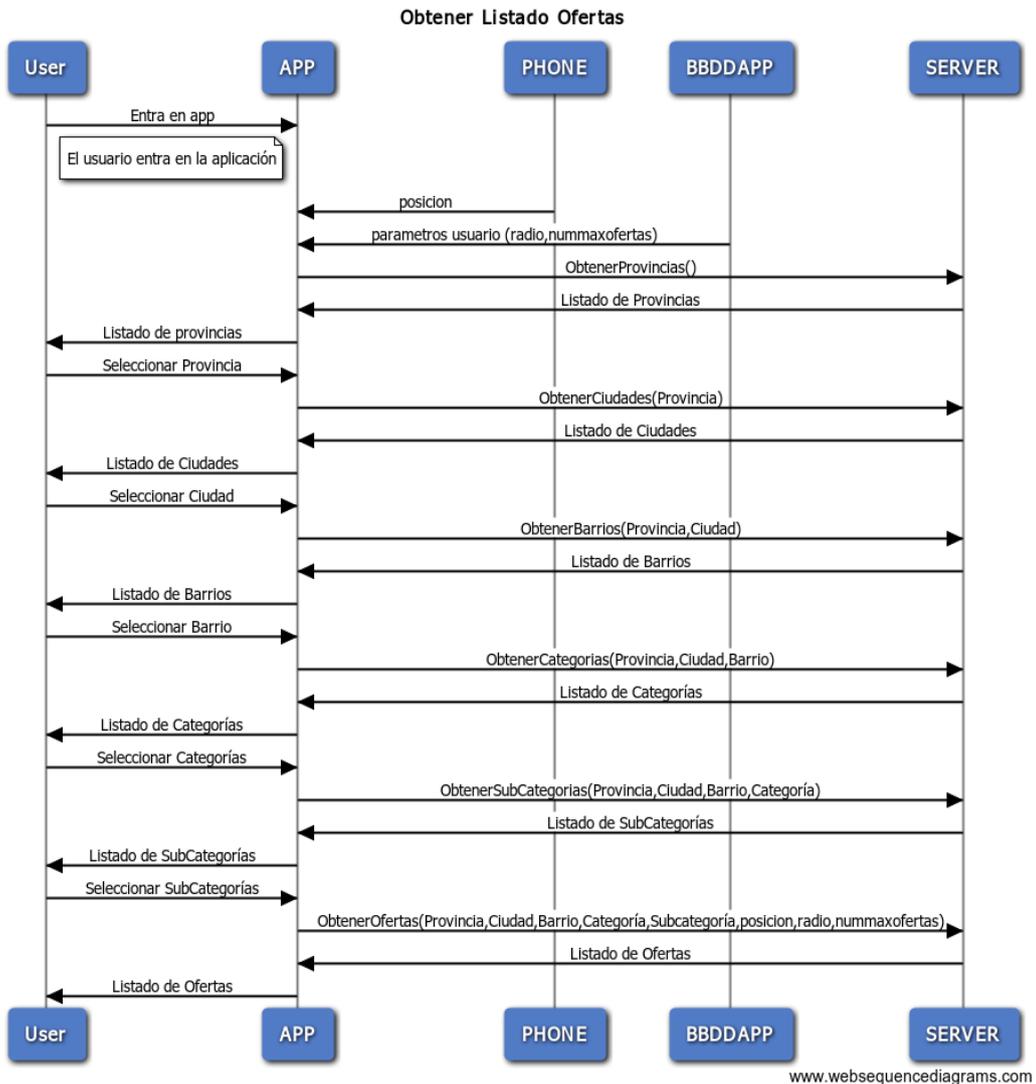




Diagramas de Secuencia

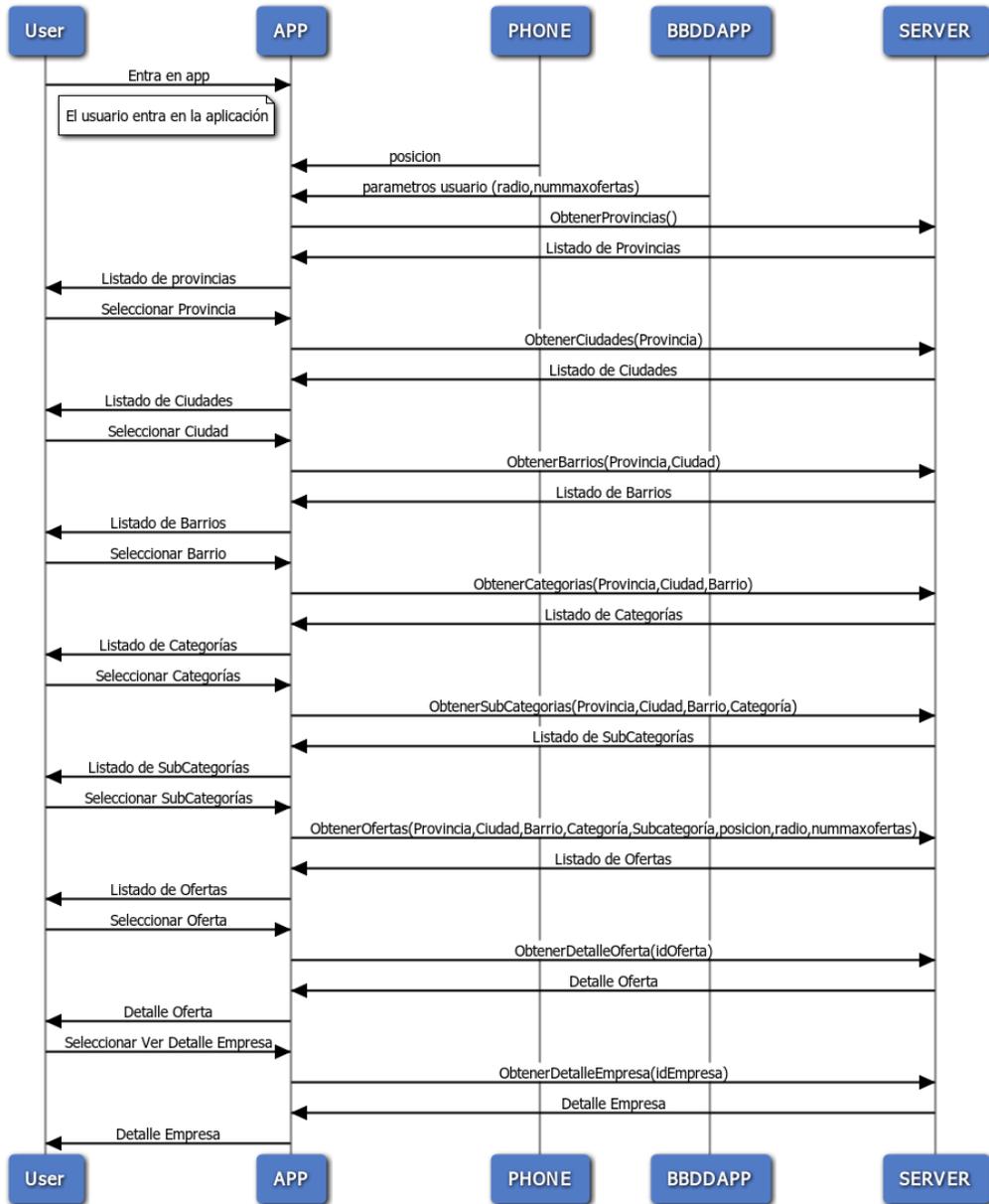
El **diagrama de secuencia** muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.

A continuación se muestran los diagramas de secuencia de los principales procesos del sistema, la intención es mostrar, a través de una línea de tiempo, el orden en el que se suceden las interacciones de los usuarios con el sistema y cómo éste obtiene la información para mostrarla.

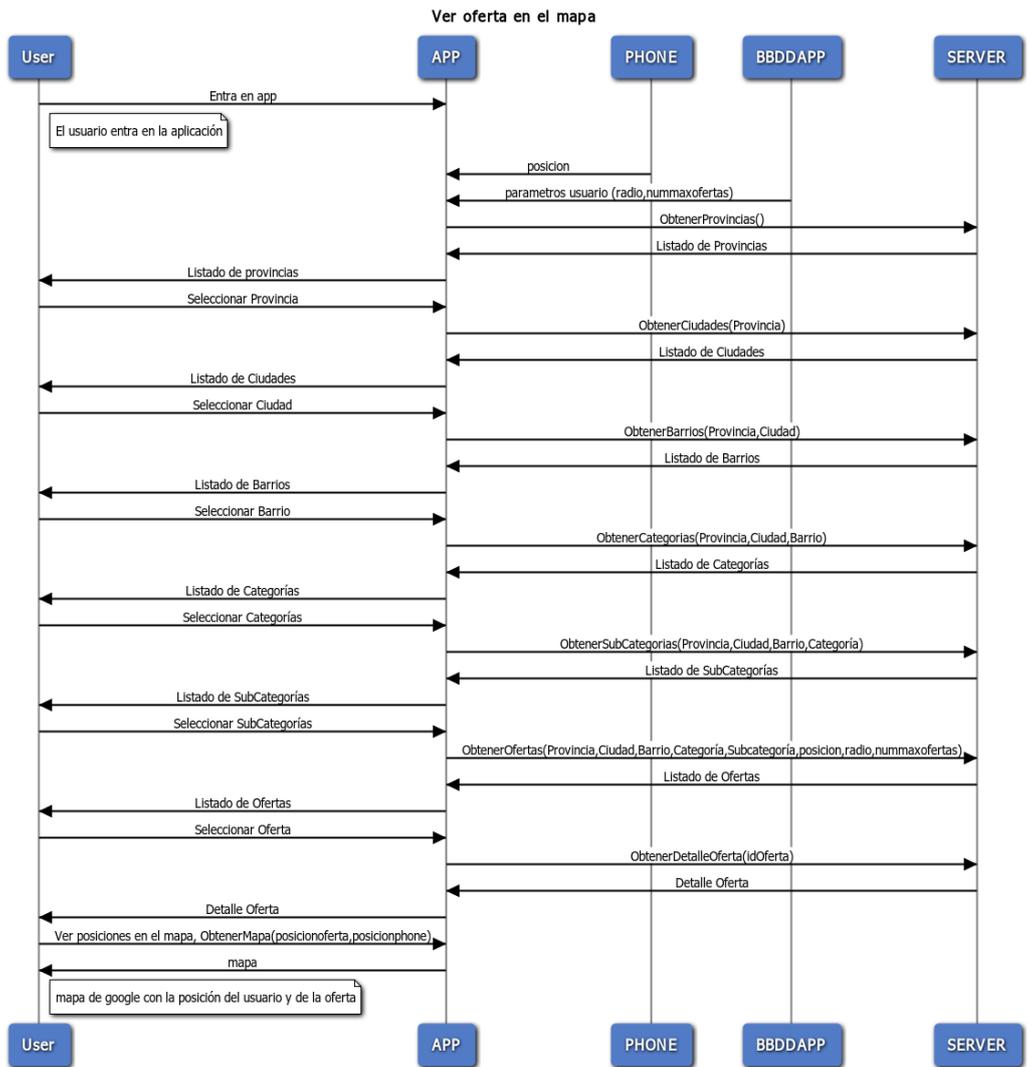


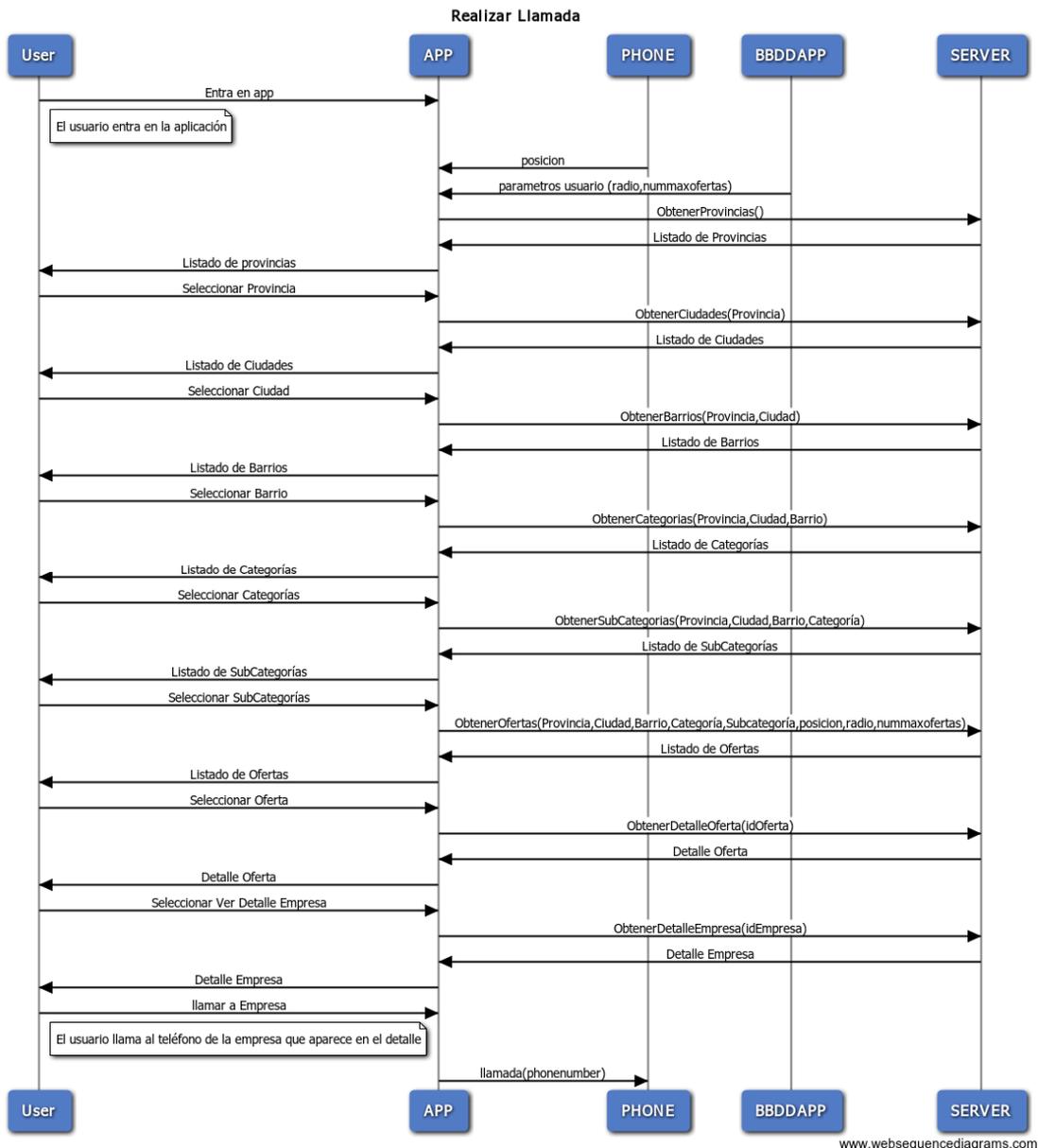


Ver Detalle Empresa



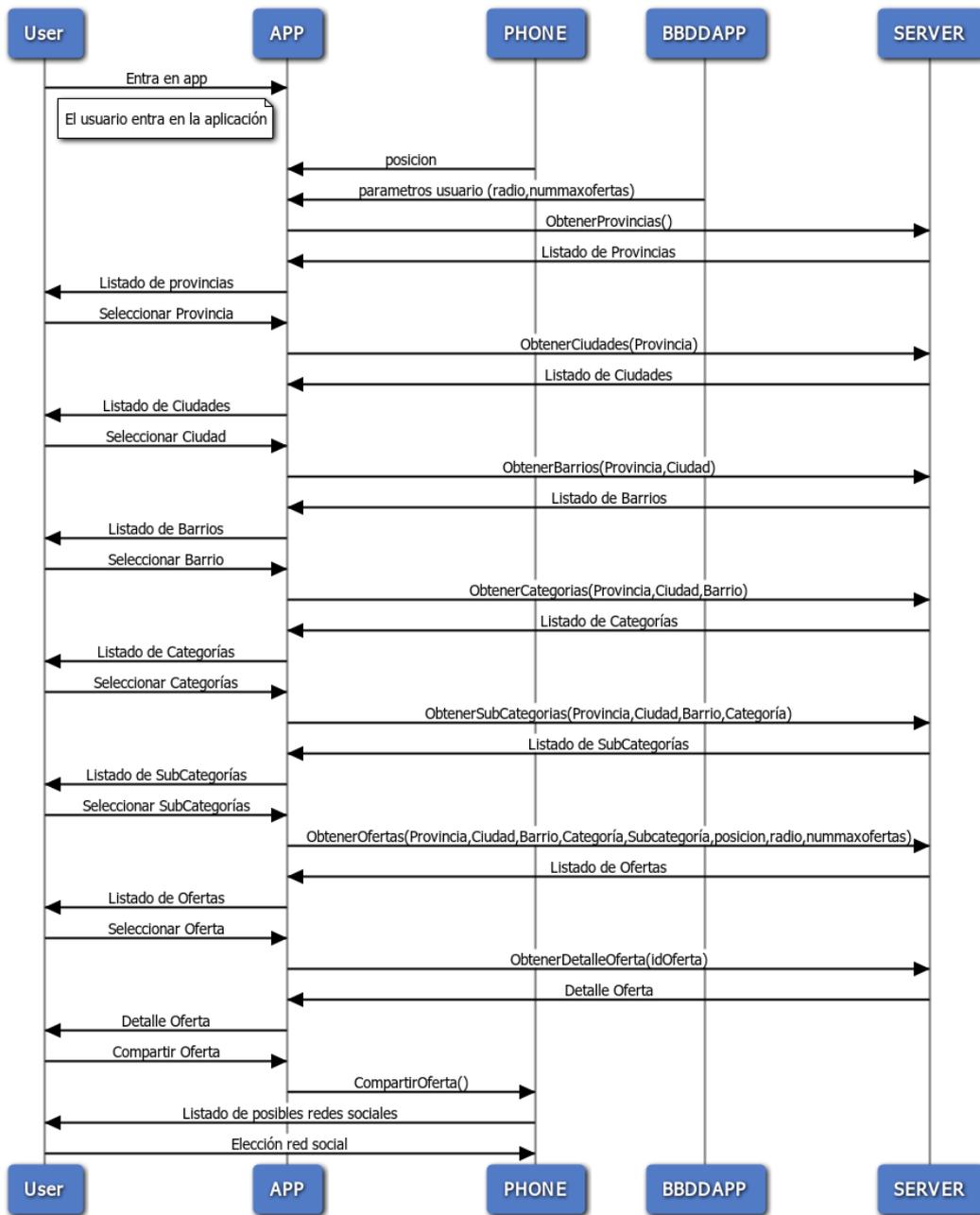
www.websequencediagrams.com







Compartir Oferta Redes Sociales



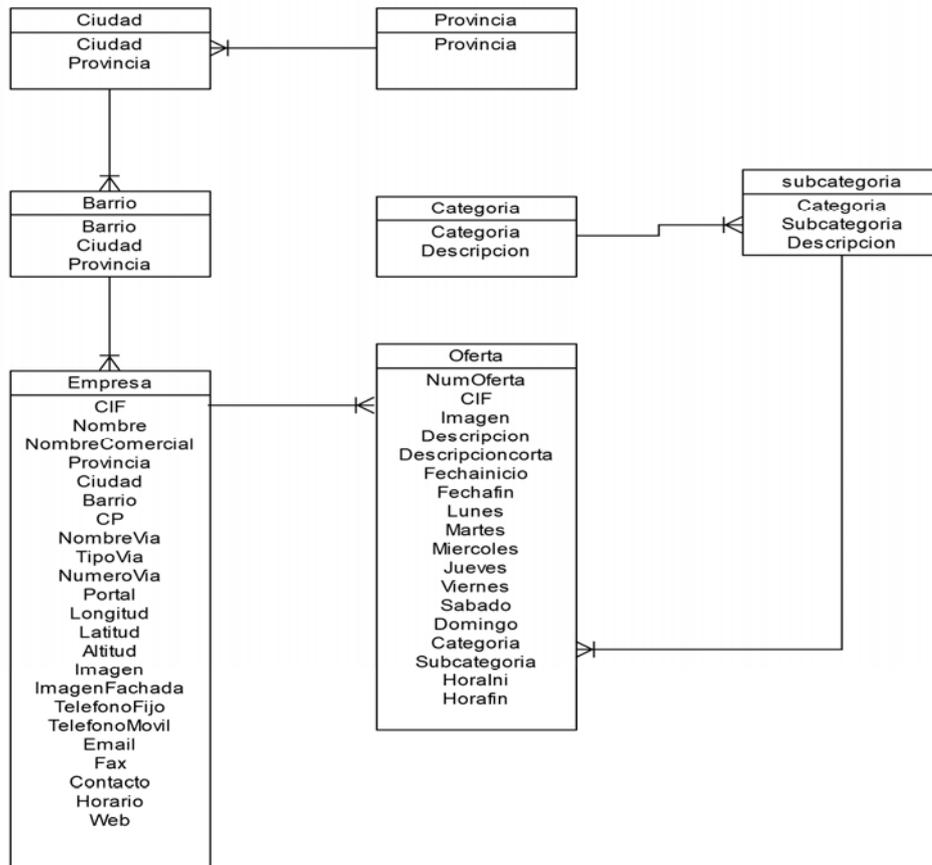
www.websequencediagrams.com



Modelo Entidad-relación

El **diagrama entidad-relación** es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.

A continuación se muestra el modelo entidad-relación que representa las entidades de la BBDD de ofertas.



Diseño físico(mysql)

En el diseño físico nos centramos en determinar las estructuras de almacenamiento y de acceso para que cuando la aplicación acceda a la base de datos tenga un buen rendimiento. A partir de aquí nos centramos en el diseño físico de la base de datos. Todas las sentencias necesarias para crear base de datos, tablas, modificaciones, etc...en un Sistema Gestor de Base de Datos como es MYSQL.

Estas son las sentencias :



```
CREATE DATABASE IF NOT EXISTS `proyectoapp` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;  
USE `proyectoapp`;
```

```
CREATE TABLE IF NOT EXISTS `provincia` (  
  `PROVINCIA` varchar(50) NOT NULL,  
  PRIMARY KEY (`PROVINCIA`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE IF NOT EXISTS `ciudad` (  
  `PROVINCIA` varchar(50) NOT NULL,  
  `CIUDAD` varchar(50) NOT NULL,  
  PRIMARY KEY (`PROVINCIA`,`CIUDAD`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
ALTER TABLE `ciudad`  
  ADD CONSTRAINT `ciudad_ibfk_1` FOREIGN KEY (`PROVINCIA`) REFERENCES `provincia` (`PROVINCIA`) ON UPDATE CASCADE;
```

```
CREATE TABLE IF NOT EXISTS `barrio` (  
  `PROVINCIA` varchar(50) NOT NULL,  
  `CIUDAD` varchar(50) NOT NULL,  
  `BARRIO` varchar(50) NOT NULL,  
  PRIMARY KEY (`PROVINCIA`,`CIUDAD`,`BARRIO`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
ALTER TABLE `barrio`  
  ADD CONSTRAINT `barrio_ibfk_1` FOREIGN KEY (`CIUDAD`,`PROVINCIA`) REFERENCES `ciudad` (`CIUDAD`,`PROVINCIA`) ON  
UPDATE CASCADE;
```

```
CREATE TABLE IF NOT EXISTS `categoria` (  
  `CATEGORIA` varchar(50) NOT NULL,  
  `DESCRIPCION` varchar(150) NOT NULL,  
  PRIMARY KEY (`CATEGORIA`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE IF NOT EXISTS `subcategoria` (  
  `CATEGORIA` varchar(50) NOT NULL,  
  `SUBCATEGORIA` varchar(50) NOT NULL,  
  `DESCRIPCION` varchar(150) NOT NULL,  
  PRIMARY KEY (`CATEGORIA`,`SUBCATEGORIA`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
ALTER TABLE `subcategoria`  
  ADD CONSTRAINT `subcategoria_ibfk_1` FOREIGN KEY (`CATEGORIA`) REFERENCES `categoria` (`CATEGORIA`) ON UPDATE  
CASCADE;
```

```
CREATE TABLE IF NOT EXISTS `empresa` (  
  `CIF` varchar(9) NOT NULL,  
  `NOMBRE` varchar(100) NOT NULL,  
  `NOMBRECOMERCIAL` varchar(100) NOT NULL,  
  `PROVINCIA` varchar(50) NOT NULL,  
  `CIUDAD` varchar(50) NOT NULL,  
  `BARRIO` varchar(100) NOT NULL,  
  `CP` varchar(5) NOT NULL,  
  `NOMBREVIA` varchar(100) NOT NULL,  
  `TIPOVIA` varchar(5) NOT NULL,  
  `NUMEROVIA` varchar(5) DEFAULT NULL,  
  `PORTAL` varchar(5) DEFAULT NULL,  
  `LONGITUD` double NOT NULL,  
  `LATITUD` double NOT NULL,  
  `ALTITUD` double NOT NULL,  
  `IMAGEN` varchar(150) NOT NULL,  
  `IMAGENFACHADA` varchar(200) DEFAULT NULL,
```



```
`TELEFONOFIJO` varchar(9) DEFAULT NULL,  
`TELEFONOMOVIL` varchar(9) DEFAULT NULL,  
`EMAIL` varchar(100) NOT NULL,  
`FAX` varchar(9) DEFAULT NULL,  
`CONTACTO` varchar(100) DEFAULT NULL,  
`HORARIO1` varchar(50) DEFAULT NULL,  
`WEB` varchar(100) DEFAULT NULL,  
PRIMARY KEY (`CIF`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
ALTER TABLE `empresa`  
ADD CONSTRAINT `empresa_ibfk_2` FOREIGN KEY (`PROVINCIA`, `CIUDAD`, `BARRIO`) REFERENCES `barrio` (`PROVINCIA`,  
`CIUDAD`, `BARRIO`) ON UPDATE CASCADE;
```

```
CREATE TABLE IF NOT EXISTS `oferta` (  
`NUMOFERTA` bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
`CIF` varchar(9) NOT NULL,  
`IMAGEN` varchar(200) DEFAULT NULL,  
`DESCRIPCION` varchar(500) NOT NULL,  
`DESCRIPCIONCORTA` varchar(50) NOT NULL,  
`FECHAINICIO` date NOT NULL,  
`FECHAFIN` date NOT NULL,  
`LUNES` tinyint(1) NOT NULL,  
`MARTES` tinyint(1) NOT NULL,  
`MIERCOLES` tinyint(1) NOT NULL,  
`JUEVES` tinyint(1) NOT NULL,  
`VIERNES` tinyint(1) NOT NULL,  
`SABADO` tinyint(1) NOT NULL,  
`DOMINGO` tinyint(1) NOT NULL,  
`CATEGORIA` varchar(50) NOT NULL,  
`SUBCATEGORIA` varchar(50) NOT NULL,  
`HORAINI1` int(11) NOT NULL DEFAULT '0',  
`HORAFIN1` int(11) NOT NULL DEFAULT '0',  
`HORAINI2` int(11) NOT NULL DEFAULT '0',  
`HORAFIN2` int(11) NOT NULL DEFAULT '0',  
`HORAINI3` int(11) NOT NULL DEFAULT '0',  
`HORAFIN3` int(11) NOT NULL DEFAULT '0',  
`HORAINI4` int(11) NOT NULL DEFAULT '0',  
`HORAFIN4` int(11) NOT NULL DEFAULT '0',  
`HORAINI5` int(11) NOT NULL DEFAULT '0',  
`HORAFIN5` int(11) NOT NULL DEFAULT '0',  
`HORAINI6` int(11) NOT NULL DEFAULT '0',  
`HORAFIN6` int(11) NOT NULL DEFAULT '0',  
`HORAINI7` int(11) NOT NULL DEFAULT '0',  
`HORAFIN7` int(11) NOT NULL DEFAULT '0',  
PRIMARY KEY (`NUMOFERTA`),  
UNIQUE KEY `NUM_OFERTA` (`NUMOFERTA`),  
KEY `FK_OFERTA_EMPRESA` (`CIF`),  
KEY `FK_OFERTA_SUBCATEGORIA` (`CATEGORIA`, `SUBCATEGORIA`),  
KEY `PREMIUM` (`PREMIUM`),  
KEY `CATEGORIA` (`CATEGORIA`),  
KEY `SUBCATEGORIA` (`SUBCATEGORIA`),  
KEY `FECHAFIN` (`FECHAFIN`),  
KEY `ANUNCIO` (`ANUNCIO`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=125 ;
```

```
ALTER TABLE `oferta`  
ADD CONSTRAINT `oferta_ibfk_1` FOREIGN KEY (`CIF`) REFERENCES `empresa` (`CIF`) ON UPDATE CASCADE,  
ADD CONSTRAINT `oferta_ibfk_2` FOREIGN KEY (`CATEGORIA`) REFERENCES `categoria` (`CATEGORIA`) ON UPDATE CASCADE,  
ADD CONSTRAINT `oferta_ibfk_3` FOREIGN KEY (`CATEGORIA`, `SUBCATEGORIA`) REFERENCES `subcategoria` (`CATEGORIA`,  
`SUBCATEGORIA`) ON UPDATE CASCADE;
```



Diseño de interfaces

A continuación se presenta todo el diseño de las pantallas de la aplicación, de cada una de las pantallas para cada una de las funcionalidades de la aplicación móvil.

Cada pantalla se mostrará en el orden en el que (lógicamente) se usaría la aplicación, configuración de búsqueda(provincia, categorías, sub-categorías, etc...), búsqueda de ofertas, listado de ofertas e información de la oferta con detalle. Ver en el mapa, llamar, compartir en redes sociales.



Imagen 1. Pantalla de inicio.

Es la primera pantalla que se mostrará cuando entremos en la aplicación. Muestra un buscador que permite filtrar por palabra clave, provincia, ciudad y barrio



Imagen 2. Pantalla de inicio, ejemplo de desplegado de opciones de los menús provincia, ciudad y barrio.



Imagen 3. Pantalla de categorías.

Las distintas opciones de categorías se mostrarán como imágenes con el fin de facilitar la tarea de selección.

Las categorías disponibles se mostrarán habilitadas con un color blanco fuerte, las no disponibles se mostrarán más apagadas y no estará habilitadas (no se podrán seleccionar)

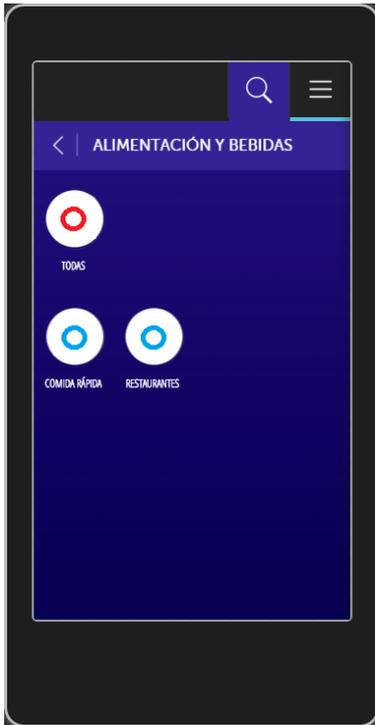


Imagen 4. Pantalla de sub-categorías.

Esta pantalla muestra un listado de las distintas sub-categorías disponibles una vez elegida la categoría principal. Si sólo existe una sub-categoría esta pantalla NO se mostrará y pasaremos directamente al listado de ofertas

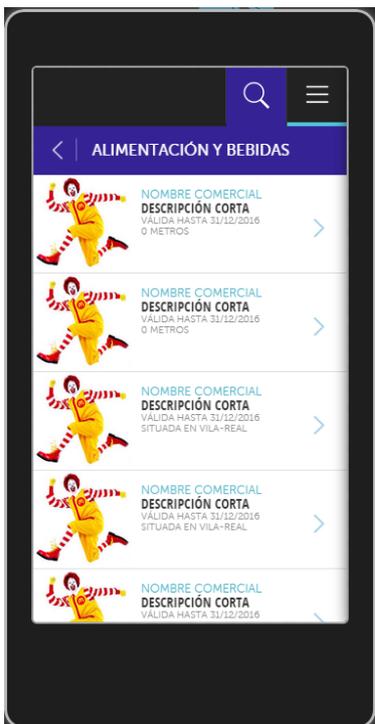


Imagen 5. Pantalla de listado de ofertas.

Esta pantalla muestra el listado de ofertas encontradas según filtros elegidos hasta el momento. Cada oferta llevará asociada una imagen distintiva el nombre comercial de la empresa, una descripción corta de la oferta y su ubicación.



Imagen 6 y 7. Pantalla de detalle de oferta.

Esta pantalla muestra el detalle de la oferta que se mostrará al seleccionar una oferta del listado de ofertas.

También muestra las distintas opciones disponibles desde esta pantalla:

.Ver más información del comercio

.Llamar al comercio

.Ver en el mapa

.Compartir con redes sociales



Imagen 7 y 8. Pantalla de detalle de comercio.

Esta pantalla aparece cuando seleccionamos (desde la oferta) ver más información de comercio.

Se muestra la información detallada del comercio (dirección, horario...)

También muestra las distintas opciones de navegación:

.Llamar al comercio

.Ver en el mapa

.Compartir con redes sociales

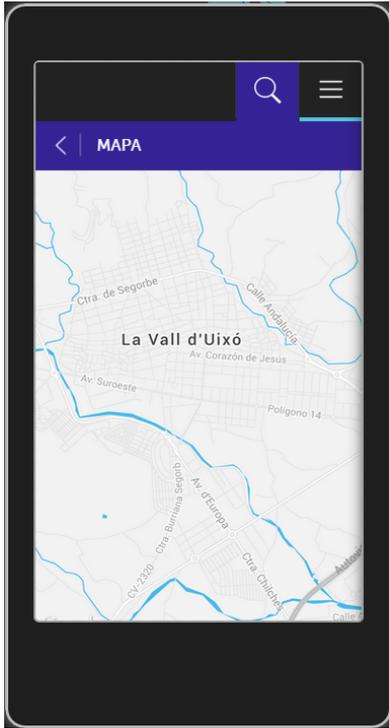


Imagen 9. Pantalla de visualizaci n de posici n del comercio/oferta.

Esta pantalla muestra la localizaci n exacta de la oferta en el mapa as  como la localizaci n del usuario



Imagen 10. Pantalla de men .

Muestra las distintas opciones de configuraci n de la aplicaci n, FAQ's y Condiciones de uso.



Imagen 11. Pantalla de menú, distancia de búsqueda.

Esta pantalla muestra a través de una barra slider la distancia máxima de búsqueda de ofertas que el usuario quiere aplicar



Imagen 12. Pantalla de menú, número de ofertas.

Esta pantalla muestra las distintas opciones que tiene el usuario para seleccionar el número máximo de ofertas que quiere que se le muestren en los listados de ofertas.



Implementación

En este apartado explicaré todos los pasos que se han seguido para la implementación del proyecto, qué complicaciones surgieron, cómo se solucionaron y qué herramientas se utilizaron.

Lo primero que creo necesario es exponer que metodología de programación web se utilizó para obtener y mostrar los datos en pantalla.

La obtención de información de la BBDD se realizará a través de llamadas AJAX a páginas php alojadas en un servidor externo. Estas páginas son las que acceden a la Base de datos alojada en el mismo servidor y que devolverán el código HTML necesario a través de respuestas codificadas con Json.

A continuación se muestra un gráfico que ilustra un ejemplo de llamada y respuesta.



Por lo tanto, siempre que se requiere obtener información desde la aplicación se realizará una llamada AJAX a un página php de un servidor externo y este nos devolverá el código HTML codificado en Json que mostraremos en nuestra app.

A continuación se muestra un ejemplo del código JavaScript y JQuery necesario para realizar una solicitud y después el código php que genera y devuelve el HTML



```
function ObtenerProvincias(latitud, longitud)
{
    url=ruta+'/php/WSprovincias.php';
    url=url+ '?format=json&longitud='+longitud+'&latitud='+latitud+'&radio='+radio;
    jQuery.ajax(
        {
            "url": url,
            "dataType": "jsonp",
            beforeSend: function(data2)
            {
                if(httpR5)
                {
                    httpR5.abort();
                }
                httpR5 = data2;
            },
            success: function(data) {

                document.getElementById("scrollercontenedororbuscar").innerHTML=data;
            },
            error: function(e, xhr){
            }

        });
}
```

```
<?php
error_reporting(E_ERROR);
header('Content-Type: text/html; charset=UTF-8');
require_once 'privado/bdusuario.php';

if (isset($_REQUEST["format"]))
{
    $format=$_REQUEST["format"];
}
else
{
    $format="json";
}

$bok=false;
if (isset($_REQUEST["longitud"]))
{
    if (isset($_REQUEST["latitud"]))
    {
        $latitud=$_REQUEST["latitud"];
        $longitud=$_REQUEST["longitud"];
        $bok=true;
    }
}

if (isset($_REQUEST["radio"]))
{
    $radio=$_REQUEST["radio"]*1000;
}
else
{
    $radio="1000000";
}

$bprimera=true;
$provincia="";
$textoprovincia="";
$hoy= date("Y")."-".date("m")."-". date("d");
```



```
$textoprovincia="                <div id='wrap'>";
$textoprovincia=$textoprovincia."                <div id='buscador'>";
$textoprovincia=$textoprovincia."                <div id='blog'></div>";
$textoprovincia=$textoprovincia."                <div class='searchbox'>";
$textoprovincia=$textoprovincia."                <input id='txtScriptAddress'
placeholder='Introduce Palabra Clave' type='text' />";
$textoprovincia=$textoprovincia."                </div>";
$textoprovincia=$textoprovincia."                <div class='searchbox' id='provincia'>";
$textoprovincia=$textoprovincia."                <label>";
$textoprovincia=$textoprovincia."                <select id='genderprovincia' data-
native-menu='true' onChange='ObtenerCiudades();>";

$where=' where (acos( sin( radians( latitud ) ) * sin( radians( '.$latitud.' ) ) + cos( radians( latitud ) ) * cos(
radians( '.$latitud.' ) ) * cos( radians( longitud ) - radians( '.$longitud.' ) ) ) *6378000
) <'.$radio;
$hoy= date("Y")."."date("m")."." date("d");
$where=$where." and fechafin>=".$hoy."";
$order=' order by empresa.provincia ASC';
$query = 'SELECT distinct empresa.provincia
FROM oferta
INNER JOIN empresa on empresa.cif=oferta.cif '.$where.$order;

$res = mysql_query($query);
if (!$res)
{
    $textoprovincia="<option value="">NADA</option>";
}
else
{
    $textoprovincia=$textoprovincia."<option value='ALL'>- TODAS LAS PROVINCIAS - -</option>";
    while($fila = mysql_fetch_row($res))
    {
        if ($bprimera)
        {
            $provincia=$fila[0];
            $bprimera=false;
        }
        $textoprovincia=$textoprovincia."<option value='".$fila[0]."'>".$fila[0]."</option>";
    }
    mysql_free_result($res);
}

$textoprovincia=$textoprovincia."                </select>";
$textoprovincia=$textoprovincia."                </label>";
$textoprovincia=$textoprovincia."                </div>";
$textoprovincia=$textoprovincia."                <div class='searchbox' id='ciudad'>";
$textoprovincia=$textoprovincia."                <select name='gender' id='genderciudad' data-
theme='b' data-native-menu='false' >";
$textoprovincia=$textoprovincia."                <option value='ALL'>- TODAS LAS
CIUDADES - -</option>";
$textoprovincia=$textoprovincia."                </select>";
$textoprovincia=$textoprovincia."                </div>";
$textoprovincia=$textoprovincia."                <div class='searchbox' id='barrio'>";
$textoprovincia=$textoprovincia."                <select name='gender' id='genderbarrio' data-
theme='b' data-native-menu='false' >";
$textoprovincia=$textoprovincia."                <option value='ALL'>- TODOS LAS BARRIOS
- -</option>";
$textoprovincia=$textoprovincia."                </select>";
$textoprovincia=$textoprovincia."                </div>";
$textoprovincia=$textoprovincia."                <div class='searchboton'>";
$textoprovincia=$textoprovincia."                <button type='submit' id='searching'
onClick='ObtenerCategorias();>BUSCAR</button>";
$textoprovincia=$textoprovincia."                </div>";
```



```
$textoprovincia=$textoprovincia." </div>";
$textoprovincia=$textoprovincia."</div>";

if($format == "json")
{
    if(isset($_REQUEST['callback']))
    {
        $callback = $_REQUEST['callback'];
        echo "$callback(" . json_encode($textoprovincia) . ")";
    }
    else
    {
        echo "callbackFunction(" . json_encode($textoprovincia) . ")";
    }
}
?>
```

Todo el desarrollo del app se realizará utilizando llamadas y respuestas del tipo expuesto.

Software para desarrollo

Para empezar comentar que todo el software utilizado se puede descargar de internet de forma gratuita y siempre como open-source.

Lo primero que tuve que descargarme fue el **WAMP (WINDOWS, APACHE, MYSQL y PHP)** como herramienta para gestionar la Base de Datos y como servidor web y php dónde realizar las pruebas de la aplicación. La descarga la hice desde la página oficial <http://www.wampserver.com>

La instalación fue muy sencilla y el acceso una vez instalado también, accediendo directamente desde el navegador a localhost.

Como ya podía gestionar la web y la BBDD ahora necesitaba una herramienta de desarrollo integrado que me permitiera trabajar con los lenguajes HTML, PHP, CSS, JavaScript y JQuery de una manera intuitiva y con un entorno sencillo y parecido al que estaba acostumbrado (DreamWeaver). Por eso después de mucho buscar encontré **AptanaStudio**.

La instalación también fue muy sencilla, siguiendo los pasos, y descubrí una gran herramienta para desarrollo web y muy similar a las herramientas de pago a las que estaba acostumbrado.

Con estas dos herramientas sabía que podía construir y probar todo el código pero aún faltaba algo... sabía que no sería una web y que precisaría siempre de un móvil para realizar las pruebas finales así que me dispuse a buscar alguna solución web, como un



emulador que me permitiera realizar pruebas sin tener que generar un instalable cada vez. Y lo encontré. El complemento que buscaba se llamada **Ripple**.

Con este plugin y el soporte HTML5 de Google Chrome es posible simular, con bastante aproximación a la realidad, prácticamente cualquier aplicación basada en tecnologías web, incluso si utiliza eventos o sensores exclusivos del móvil como uso de botones o movimientos del acelerómetro.

Problemas y soluciones

Ya tenía experiencia en la programación de webs dinámicas, que obtenían de forma asíncrona, utilizando AJAX, la información de un servidor php pero no sabía cómo podría obtener la información que sólo el móvil podía ofrecerme.

A continuación se detallan las problemas que tuve y las soluciones que se aportaron:

¿Cómo obtener la posición geo-espacial?

Esta pregunta siempre fue mi mayor temor. La aplicación debía obtener los datos de latitud y longitud a través del móvil, utilizando los datos móviles o el gps.

Sabía que se podía hacer, existen cientos o miles de aplicaciones móviles compiladas con phonegap que permitían obtener los datos de posición geoespacial.

La solución al problema que tanto me atormentaba la encontré enseguida, a través de la documentación disponible en la web de phonegap.

Aquí se expone la solución que consta de dos partes:

1. Primero debemos vincular el plugin a nuestro proyecto a través del fichero **config.xml**

```
<gap:plugin name="org.apache.cordova.geolocation" />
```

2. Desde el código JavaScript

```
navigator.geolocation.getCurrentPosition(localizacion,errores2,{ timeout: 6000,  
maximumAge: 3000});
```

¿Cómo compartir con las redes sociales?

Mi próxima duda y preocupación también estaba relacionada con la interacción aplicación-móvil y era cómo podría compartir la información que obtenía de la aplicación con las distintas redes sociales.



La solución también la encontré en la documentación facilitada por la web de phonegap.

Primero debía vincular la aplicación con el plugin correspondiente y después realizar la llamada adecuada desde el código en JavaScript

1. Primero debemos vincular el plugin a nuestro proyecto a través del fichero **config.xml**

```
<gap:plugin name="nl.x-services.plugins.socialsharing" />
```

2. Desde el código JavaScript

```
window.plugins.socialsharing.available(function(isAvailable)  
{  
    if (isAvailable)  
    {  
        url=ruta+'/php/VerEmpresa.php?cif='+c;  
        window.plugins.socialsharing.share(null, null, null,url);  
    }  
});
```



Mejoras Futuras

En esta sección se presentan lo que podrían ser, bajo mi punto de vista, posibles mejoras que se podrían incluir en la aplicación móvil.

Favoritos. Añadir la opción de poder marcar y almacenar las ofertas más interesantes para el usuario y ofrecer la posibilidad de que estas fueran accesibles desde una sección aparte.

Login. Ofrecer la posibilidad de darse de alta y hacer login en el app, indicando dirección de correo y password. La intención sería doble, por un lado, al disponer del correo electrónico del usuario, le podríamos enviar información acerca de ofertas y por otro lo podríamos utilizar esta clave única para vincular la información que se estaba almacenando en el móvil y que ahora se almacenara en la red y así poder acceder a ella desde cualquier otro dispositivo.

Web Comercios. Aunque no era el propósito de este proyecto, estaría bien ofrecer la posibilidad a los comercios de gestionar ellos mismos sus propias ofertas a través de un portal web. En esta portal web podrían añadir, modificar y borrar ofertas vinculadas con su comercio.

Ofertas Especiales. Crear un sistema para poder marcar de alguna manera aquellas ofertas que por motivos económicos o simplemente por ser especialmente interesantes pudieran mostrarse las primeras en los listados de búsquedas, saltándose el orden por cercanía actualmente contemplado.

Sistema de puntos. Crear, junto con los comercios, algún tipo de código vinculado al comercio y que los usuario supieran que al ir al comercio y comprar alguno de los productos mostrados por la aplicación, el comercio pudiera introducir su código y así poder saber que usuario hacen más uso de la aplicación. El propósito sería el de poder premiar a aquellos usuarios con más puntos con algún tipo de regalo o cheque descuento.



Conclusiones

Supongo que el objetivo general del proyecto final de carrera es poner en práctica parte de lo aprendido durante los años de estudio. Y no hablo sólo de los conocimientos en programación, en Bases de datos, redes o tecnologías web, sino también de la metodología seguida para conseguir que todo encaje, ver cómo pasando por las distintas fases de análisis, diseño e implementación se puede conseguir un producto de calidad.

Creo que los objetivos propios del proyecto presentado hoy quedan cumplidos y también queda cumplido el objetivo general del proyecto fin de carrera.

Por supuesto que se podría mejorar, que se podría hacer más eficiente, más bonito o con más funcionalidades pero eso queda para el futuro.

Por mi parte sólo decir que he invertido muchas horas en este proyecto, seguramente más de las que creía, que he podido poner en práctica muchos de los conocimientos adquiridos durante estos años. Que también he tenido que realizar labores de búsqueda e investigación para solucionar problemas a los que nunca me había enfrentado y que estoy encantado con elección de la rama de mi proyecto. Poder desarrollar una aplicación móvil multiplataforma utilizando únicamente tecnología web es increíble.

Por último, agradecer a la UOC la forma que tiene de ofrecer a miles de estudiantes como yo, que no pueden asistir a clases presenciales, la posibilidad de seguir estudiando, ampliando conocimientos y competencias.



Bibliografía

Programación HTML5:

<http://www.w3schools.com/html/>

Programación JavaScript:

<http://www.w3schools.com/js/>

Programación JQuery:

<http://api.jquery.com/>

Programación CSS:

<http://www.w3schools.com/css/>

Programación PhoneGap:

http://docs.phonegap.com/en/edge/guide_platforms_index.md.html

<http://www.phonegapspain.com/>

Compilar y generar app:

http://docs.build.phonegap.com/en_US/support_getting_help.md.html#googtrans%28en%29