Guía de Puntos de Interés de la Ciudad de Madrid

Proyecto de Fin de Carrera

Daniel Cortés Fernández

Introducción

Esta presentación se corresponde con el Proyecto de Fin de Carrera de los estudios de Informática, cursados en la universidad Oberta de Cataluña.

El objetivo de este documento es mostrar, de forma global, el proceso de ejecución de este PFC. En esta presentación se ha realizado un esfuerzo de síntesis, centrando la explicación únicamente en los elementos más relevantes del proyecto.

Se comenzará la presentación con una breve descripción de los objetivos a alcanzar con la realización del proyecto, así como la definición del problema a resolver.

Una vez presentados estos apartados, se explicarán las tareas que se han llevado a cabo en cada uno de los entregables en los que se ha dividido el proyecto. Estos entregables son:

1. Plan de Trabajo
2. Análisis
3. Implementación
4. Entrega final

Objetivos

Los objetivos que se pretenden alcanzar pueden dividirse en dos ámbitos: objetivos técnicos y objetivos de gestión del proyecto. A continuación se presenta una enumeración de ambos.

Objetivos Técnicos

- Diseño un Sistema de Información Geográfica para móviles.
- Conocimiento del S.O Android
- Manejar las características más comunes de aplicaciones móviles (BBDD, GPS, uso de memoria externa)
- Conocer las funcionalidades ofrecidas por el API de Google Maps

Objetivos de Gestión

- Gestión de proyectos
- Presentación de información de forma clara y efectiva

Presentación del problema

Para alcanzar los objetivos propuestos, se creará un guía de puntos de interés. Este software, que funcionará sobre la plataforma Android, contará con las siguientes características:

Características principales:

- Ubicación de puntos de interés sobre mapa zonal
- Información sobre puntos de interés
- Calculo de la ruta desde la posición del usuario hasta un punto concreto
- Capacidad para mostrar imágenes relacionadas con el punto de interés
- Gestión de puntos definidos por el usuario

Fases

Para establecer un sistema de seguimiento, el proyecto ha sido divido en fases:

- Plan de Trabajo
- Análisis
- Implementación
- Memoria final

Estas fases servirán de hilo conductor a este documento. En páginas sucesivas se detallarán los trabajos que se llevaron a cabo en cada fase, deteniéndonos en las tareas de mayor relevancia. La última fase, «Memoria final», servirá únicamente para corregir posibles errores y repasar aspectos formales.

Plan de Trabajo

En el plan de trabajo se configuró un guion de acción preliminar para llevar a cabo el proyecto. Este guion incluía:

1. Una presentación del proyecto y de sus objetivos.
2. El establecimiento de un calendario de ejecución, identificando las tareas, actividades e hitos más importantes del proyecto.
3. Definición de roles y estimación de horas de trabajo por cada rol
4. El material necesario para la realización del proyecto.
5. Un plan de contingencia diseñado para corregir desviaciones sobre la planificación.

El plan de trabajo incluye también un diagrama te Gantt dónde se observa fácilmente el desglose de tareas, la relación entre ellas y los plazos de ejecución previstos.

Puede consultar el Plan de Trabajo completo en el documento ***«Anexo III – Plan de Trabajo»*** de este PFC.

Fase de Análisis

En la fase de análisis se realiza un estudio del problema desde el punto de vista del negocio. En primer lugar se realiza un análisis de los requisitos expresados por el cliente. Obtenidos estos requisitos se diseña y modela en UML un sistema capaz de dar respuesta a los mismos. Estos diseños servirán de input para la siguiente fase (implementación).

Las tareas que se llevan a cabo en esta fase son:

1. Presentación de la metodología SCRUM. Esta metodología es la elegida para la ejecución de este proyecto.
2. Estudio del problema y obtención de los requisitos funcionales de usuario.
3. Representación de los requisitos de usuario por medio de Historias de Usuario y diagrama de casos de uso.
4. Análisis UML del conjunto del sistema (diagrama de clases)
5. Análisis UML de las funcionalidades detectadas en las Historias de Usuario
6. Diseño de la interfaz gráfica de usuario
7. Diseño del plan de pruebas de usuario

Análisis: Metodología

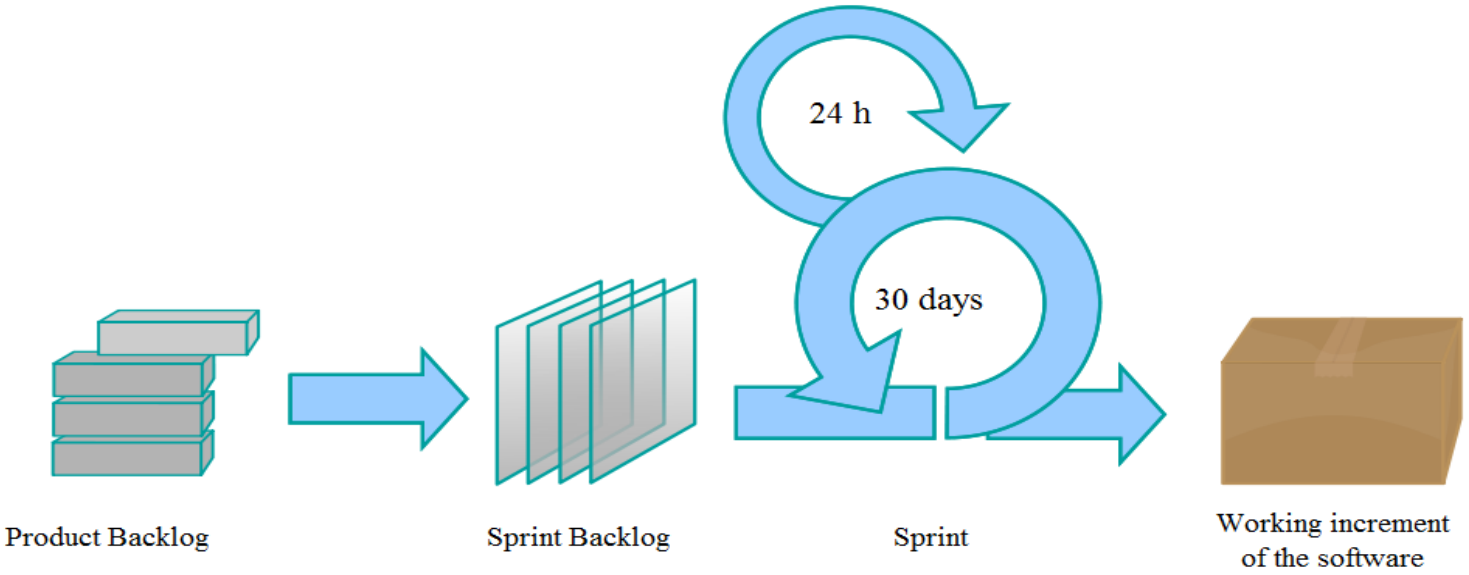
SCRUM es una metodología enfocada a la gestión de desarrollos de software. Se encuentra enmarcada en el ámbito del desarrollo rápido de software. Se ha elegido esta metodología por los siguientes motivos:

- Fácil de seguir
- Orientada a desarrollo iterativo e incremental
- Puede combinarse con técnicas de programación extrema.
- No impone un cumplimento estricto de sus reglas: puede adaptarse a las circunstancias de cada proyecto.

De acuerdo con la orientación iterativa e incremental de los desarrollos en los que se utiliza SCRUM, crearemos un prototipo como propuesta de solución al usuario. Sobre este prototipo, el usuario podrá realizar cambios, y corregir y matizar algunos aspectos que no quedaron claros en los requisitos. Este sistema permitirá detectar errores de negocio en una fase muy temprana del proyecto, siendo menos costosos de arreglar que los detectados en fases más tardías.

Análisis: Proceso SCRUM

Un proceso típico SCRUM comienza con la recogida de requisitos expresadas por medio de historias de usuario. Las historias de producto se almacenan en una pila llamada «Pila de producto». El proceso de desarrollo se basa en iteraciones, llamadas Sprints. Previo a la realización de cada Sprint, el equipo se reúne y decide cuantas historias de usuario van a ser implementadas. De esta forma se obtiene un prototipo rápidamente y se trabaja sobre él hasta completar el proyecto.



En la realización del proyecto se ha utilizado principalmente la captura de requisitos mediante historias de usuario y las ideas de desarrollo iterativo e incremental

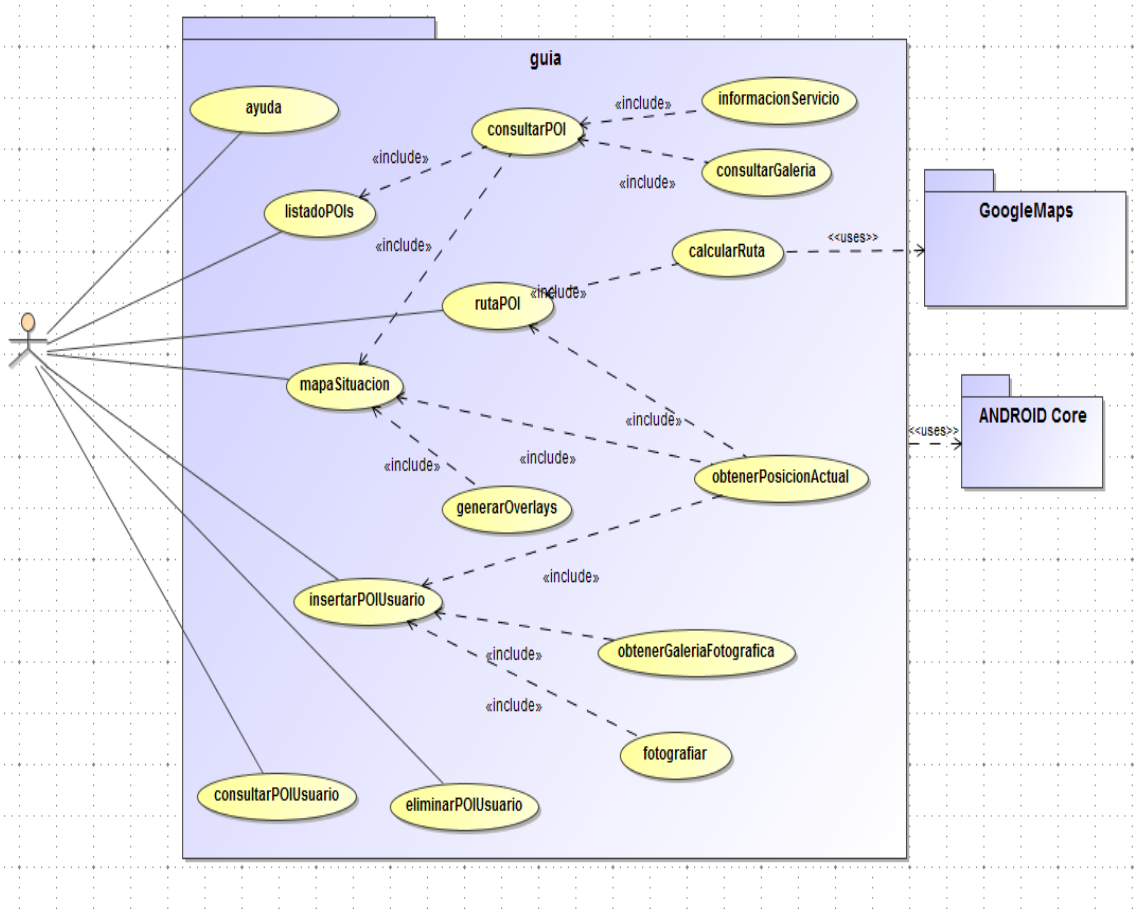
Análisis: Historias de Usuario

Una historia de usuario es la expresión, simple y coloquial, de una funcionalidad deseada por el usuario. En este proyecto se han registrado las siguientes historias de usuario:

1. **Mapa de Situación**: Un usuario que está en Madrid desea saber qué puntos de interés están cercanos al lugar dónde se encuentra en este momento y, de esta forma, poder planificar el recorrido que quiere realizar.
2. **Consulta de punto de interés**: Un usuario podrá obtener información sobre un punto de interés. Esta información puede variar entre información del contexto histórico, de la arquitectura del lugar o información de servicio (horarios, precios, etc…). La información podrá estar completada por fotografías de interés: pictórico, arquitectónico o de situación.
3. **Cómo llegar**: Un usuario desea que la guía le indique como llegar a un punto de interés de su elección desde el lugar en el que se encuentra actualmente.
4. **Mis favoritos**: Un usuario debe poder guardar en su terminal puntos de la ciudad que le han gustado, independientemente de si están catalogados como de interés o no. De esta forma podría tenerlos para visitas posteriores o compartirlos con personas de su entorno.

Análisis: Identificación de casos de uso

De las historias de usuario se han extraído los siguientes casos de uso:



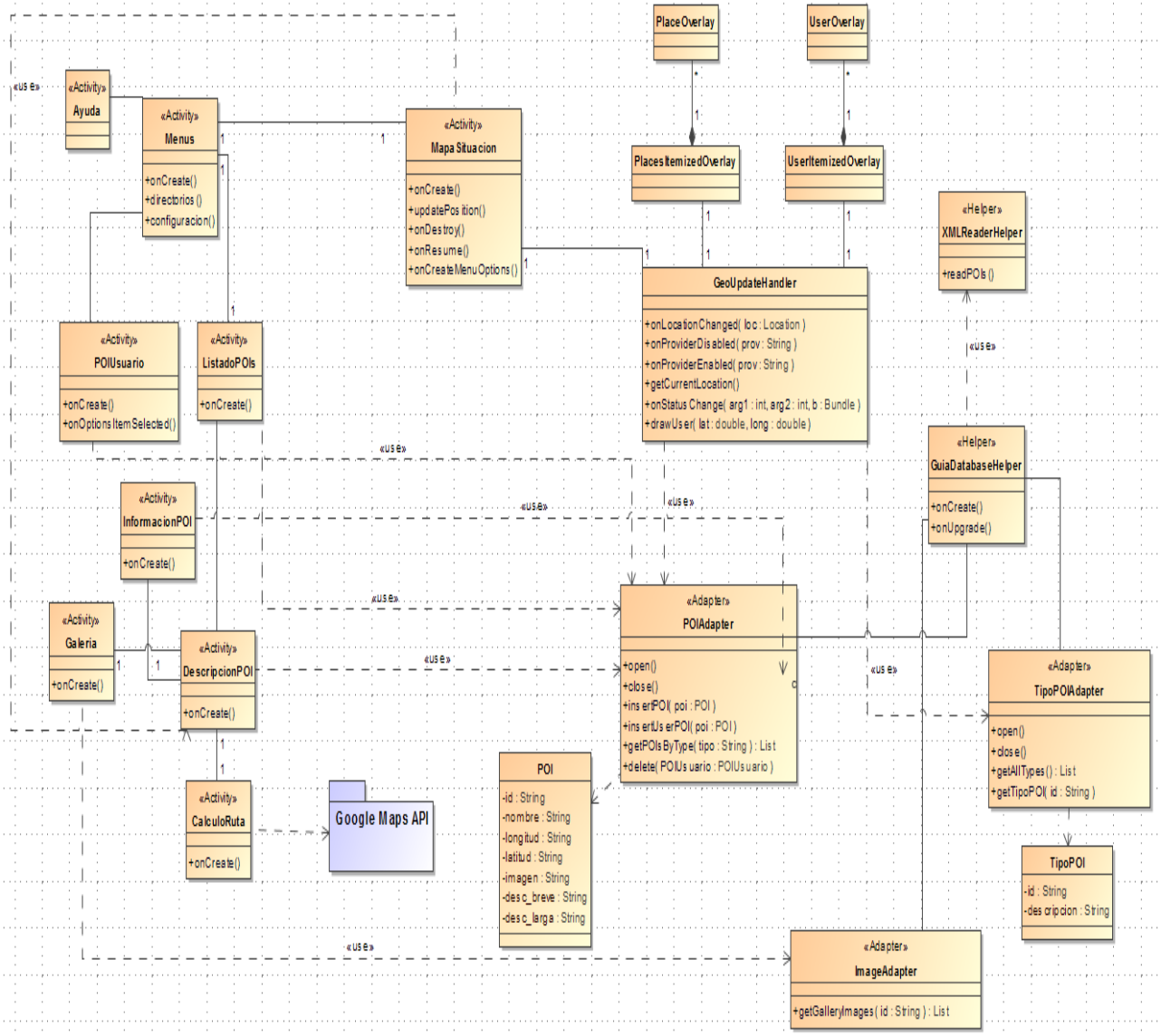
Algunas de las historias de usuario han dado como resultado la creación de varios casos de uso.

Análisis: Diagrama de Clases

Identificada la funcionalidad, se propone un sistema que pueda dar respuesta a todos los problemas planteados.

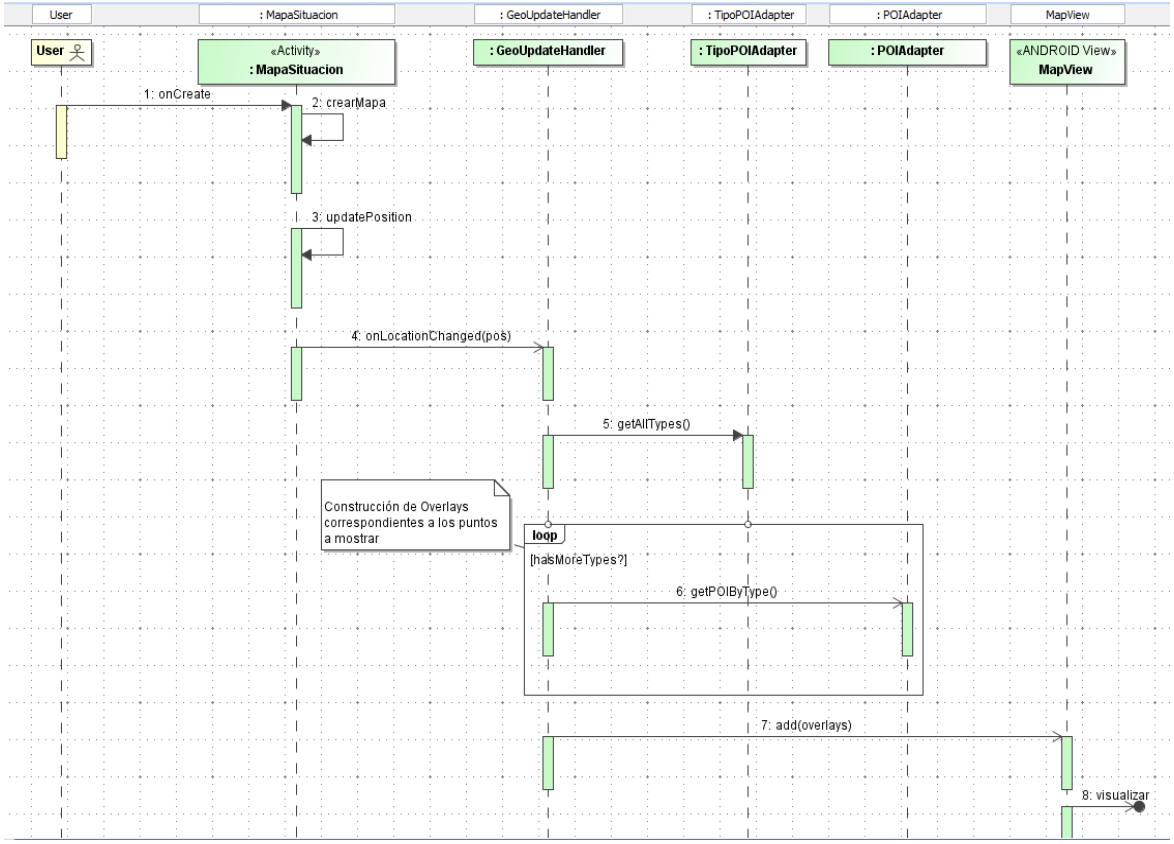
Para modelizar el sistema, utilizaremos distintos tipos de diagramas UML que representan diferentes aspectos del sistema.

El diagrama de clases permite representar las clases que componen el sistema y la relación existente entre ellas.



Análisis: Diagramas de secuencia

Se han utilizado diagramas de secuencia para mostrar la interacción de los objetos de cada caso de uso a través del tiempo. Los diagramas muestran las llamadas, las respuestas y los parámetros que se intercambian los objetos. Todos los diagramas de secuencia realizados se encuentran disponibles en el documento ***«Memoria de Proyecto».***



Prototipo: Mapa de Situación

La propuesta de funcionalidad para esta historia de usuario es la siguiente:

1. Seleccionamos «Mi posición» desde el menú contextual.

2. La aplicación extrae las coordenadas del GPS y muestra el mapa zonal.

3. Los puntos de interés responden a pulsaciones de usuario mostrando una breve descripción y opciones.

4. El icono que representa la posición actual de usuario muestra la dirección postal en la que este se encuentra al ser pulsado.

Prototipo: Consulta de puntos de interés

La propuesta de funcionalidad para esta historia de usuario es la siguiente:

1. Seleccionamos «Puntos de interés» desde el menú contextual.

2. Se muestra una lista de puntos de usuario.

3. Seleccionando un punto de interés accedemos a su descripción completa y a otras opciones sobre el punto (Cómo llegar, galería, información)

4. Información de servicio sobre el punto de interés.

5. Se visualizan imágenes asociadas al punto de interés.

Prototipo: Cómo llegar

Esta funcionalidad, por comodidad para el usuario, puede ejecutarse de dos formas distintas:

1

2

3

Pulsando «Como Llegar» o «Llegar» se calcula y muestra la ruta a pie al punto indicado.

1

2

Prototipo: Mis favoritos

1 Seleccionamos «Mis Favoritos» desde el menú contextual.

2 Es necesario rellenar el nombre del punto. El sistema lo autocompletará si ya está en BBDD.

3 Se puede capturar una imagen desde la cámara o la galería de imágenes.

3 Pulsando «Borrar punto» se elimina el punto de usuario de la BBDD.

4 Pulsando «Guardar punto» se inserta el punto de usuario de la BBDD.

Fase de Implementación

En la fase de implementación se analizará, a nivel técnico, la forma de llevar a cabo las funcionalidades localizadas en la fase anterior. Esta fase incluye también la investigación sobre todas las cuestiones técnicas que comprende este proyecto, entre las que cabe destacar:

1. Creación de interfaces gráficas de usuario
2. Manejo del GPS del terminal
3. Google Maps API
4. Manejo de mapas desde Android
5. Manejo de SQLite (BBDD embebida en el sistema Android)
6. Gestión de ficheros XML
7. Gestión de imágenes
8. Manejo de la tarjeta de memoria externa del terminal

La fase de implementación da comienzo con la configuración y puesta a punto del entorno de desarrollo a utilizar.

Implementación: Entorno de desarrollo

Las herramientas utilizadas para el desarrollo son las siguientes:

1. Eclipse
2. Android SDK

Eclipse es un entorno de desarrollo, de propósito general, que admite varios lenguajes y puede ampliar sus prestaciones por medio de plug-ins.

Android SDK es el kit de desarrollo estándar para Android. Este kit contiene:

a. Sistema Operativo Android
b. SQLite : SGBD especialmente diseñado para disminuir el consumo de recursos.
c. AVD: Dispositivo virtual Android, incorporado al kit con fines de testing.

Ambas herramientas se integran mediante el **plugin ADT**, que permite utilizar las herramientas incorporadas en Android SDK desde Eclipse. La explicación detallada de la configuración del entorno puede consultarse en el documento ***«Anexo IV – Configuración del entorno de desarrollo».***

Implementación: Interfaces de Usuario

El desarrollo de interfaces de usuario se basa en la creación de ficheros XML que describen la configuración que se visualizará en pantalla.

Cada actividad Android tiene asociado un fichero que describe su aspecto y que está compuesto por gestores de posicionamiento (**Layouts**) y por componentes a mostrar (textos, scrolls, imágenes)

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:id="@+id/layout_root"
              android:orientation="horizontal"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
              android:padding="10dp"
              >
    <ImageView android:id="@+id/image"
               android:layout_width="wrap_content"
               android:layout_height="wrap_content"
               android:layout_marginRight="10dp"
               />
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:textColor="#FFF"
              />
</LinearLayout>
```

```java
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.detalle_poi);

    String id=(String)(getIntent().getExtras()).get("id");

    POIAdapter poiAdapter=new POIAdapter(getApplication());
    poiAdapter.open();

    poi=poiAdapter.getPOI(id);

    imgCabecera=(ImageView)findViewById(R.id.icon);
    imgCabecera.setImageResource(Menus.hImagenes.get(poi.getImagen()));

    TextView titulo=(TextView)findViewById(R.id.titulo);
    titulo.setText(poi.getNombre());

    TextView cuerpo=(TextView)findViewById(R.id.cuerpo);
    cuerpo.setText(poi.getDesc_larga());
```

Desde la clase Java de la actividad pueden recuperarse estos elementos y realizar acciones sobre ellos. Además, pueden asignarse **Listeners** para reaccionar ante eventos originados por el usuario.

Implementación: Manejo del GPS

Android proporciona una interfaz que permite interactuar con el dispositivo GPS del terminal. Esta interfaz permite conocer:

1. Los cambios de localización del dispositivo
2. La activación del GPS
3. La desactivación del GPS
4. Los cambios de estado del GPS

Se implementa este interfaz mediante la clase **GeoUpdateHandler**, encargada de gestionar la comunicación con el GPS. La aplicación actualiza la posición cada 5s o cada cambio de 5 metros ya que es una guía pensada para ir a pie y requiere precisión en la localización.

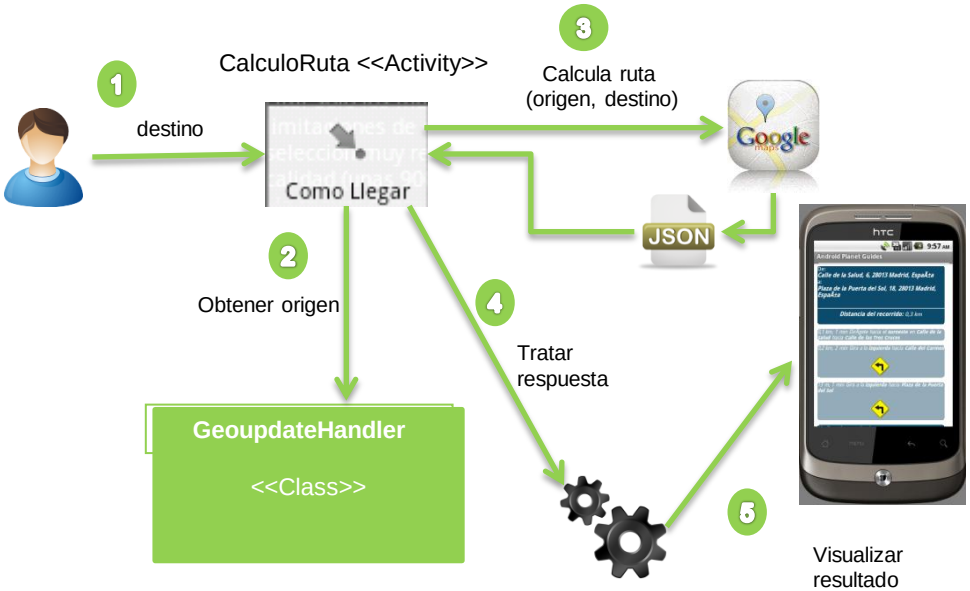
```
miGPS= new GeoUpdateHandler(this,mapController,mapView,mapOverlays,this);
locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);

if(locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)){
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 5000, 5, miGPS);
}else{
    Toast.makeText(getBaseContext(), "Proveedor de servicio GPS no encontrado", Toast.LENGTH_SHORT).show();
}
```

Implementación: Web Services de Google Maps API

Los **servicios web de Google Maps** son un conjunto de interfaces HTTP que proporcionan datos geográficos para aplicaciones de mapas. Para utilizarlos es preciso enviar solicitudes a URLs específicas, proporcionando una serie de datos. El servicio devuelve la respuesta en notación **JSON**, esta notación es tratada posteriormente para mostrar la información recogida.

En este proyecto se ha utilizado el web service directions para calcular la ruta a seguir entre dos puntos. El esquema de utilización es el siguiente:




```
Log.d("INFORMACION","Preparando conexion ...");
URL url= new URL("http://maps.google.com/maps/api/directions/json?origin="
        +loc.getLatitude()+","+loc.getLongitude()
        +"&destination="+latitudDestino+","+longitudDestino
        +"&sensor=true&mode=walking&language=es");

URLConnection urlCon=url.openConnection();
Log.d("INFORMACION","Conexion abierta ...");
InputStream in=urlCon.getInputStream();

StringBuffer store=new StringBuffer();

byte [] b=new byte[1024];
int n=0;
while ((n=in.read(b))>0){
    store.append(new String(b, 0, n, "UTF-8"));
}

in.close();
```

Llamada al servicio

Implementación: Mapas

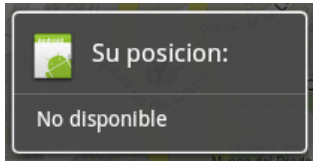
La gestión de mapas se realiza mediante las vistas de mapa (**MapView**). Los MapView muestran, dadas unas coordenadas, el plano zonal correspondiente. Además, los MapView pueden actuar a modo de contenedor, permitiendo añadir elementos superpuestos (**overlays**) sobre unas coordenadas dadas.

Para representar los puntos de interés a visitar se han utilizado diferentes overlays en función de la tipología del punto, de esta forma, el usuario puede identificar la temática fácilmente. La posición de usuario se representa mediante un icono humanoide.

Puede modificarse el comportamiento de los overlays para que el usuario pueda Interactuar con ellos. En este proyecto se ha modificado el comportamiento de los Overlays en función de su tipología:

a. Punto de interés: Se muestra una ventana emergente con información previa y opciones
b. Posición usuario: Muestra una ventana emergente con la dirección postal en la que se encuentra el usuario.

Comportamiento de punto de interés

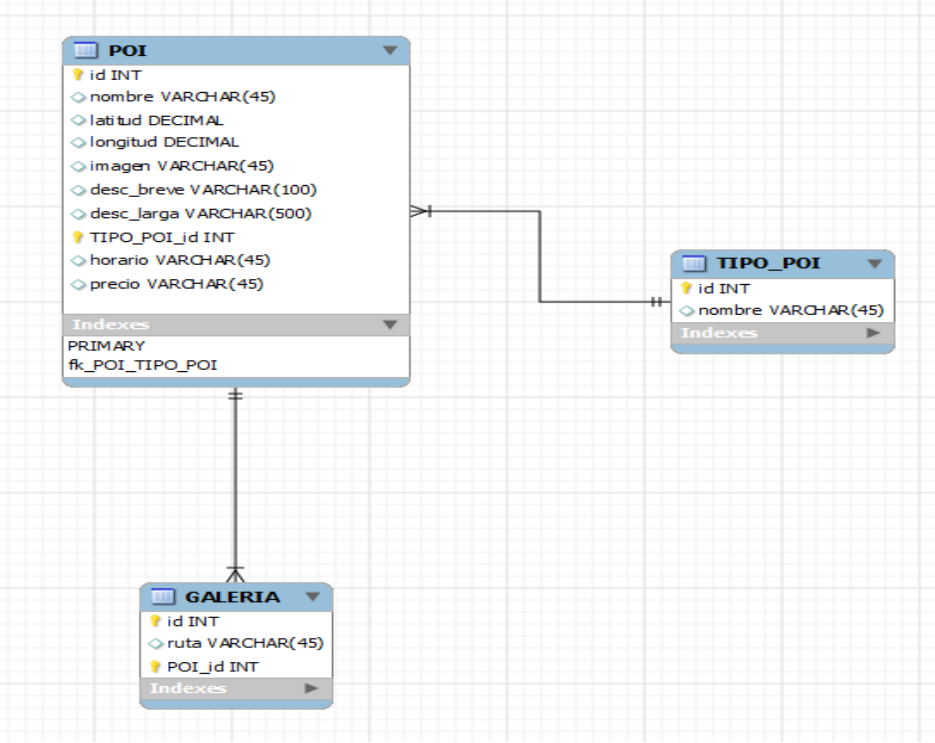


Comportamiento de usuario

Implementación: SQLite

SQLite es el **SGBD** incorporado en Android por defecto. En este proyecto se ha utilizado para crear una BBDD en la que almacenar los puntos de interés definidos por la aplicación y los puntos de interés definidos por el usuario. El modelo de datos que se ha utilizado se muestra en la figura.

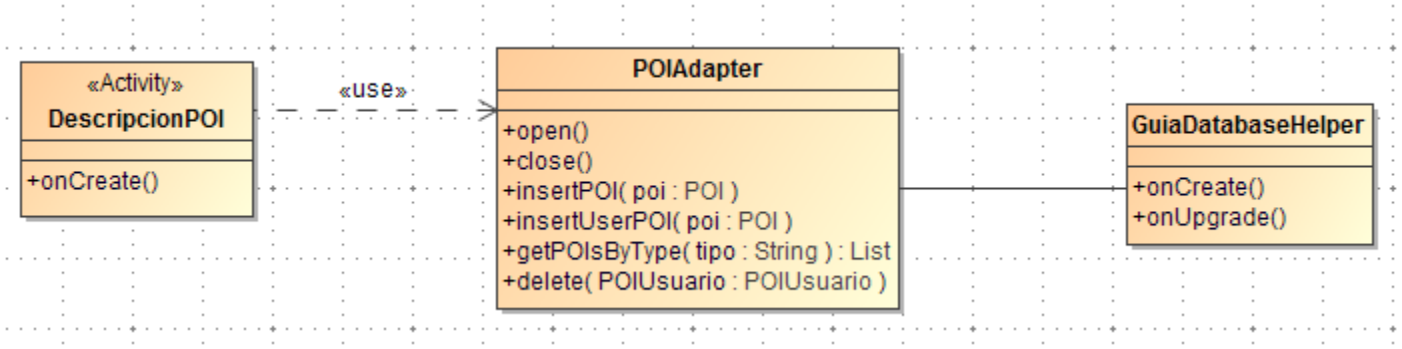
Para operar con BBDD, Android provee la clase **SQLiteOpenHelper**, debemos extender la funcionalidad de esta clase con el fin de adaptarla a las particularidades de nuestro negocio.



Implementación: Patrón de uso BBDD

Con el fin de estructurar las operaciones con BBDD, se ha creado un sistema basado ***Helpers*** y ***Adapters***. Cada entidad de BBDD cuenta con una clase ***Adapter*** que implementa las funcionalidades ***CRUD*** asociadas a esa entidad. Los ***Adapters*** se aíslan de los detalles de la conexión con BBDD mediante una clase ***Helper***, que se encarga de proveer esta conexión.

Siguiendo este patrón, cualquier actividad puede operar con BBDD aislándose de los detalles técnicos de la operación. La actividad únicamente deberá centrarse en el negocio.

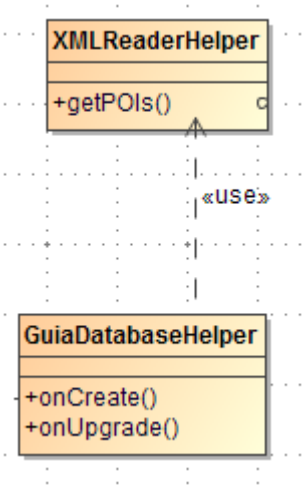


Esquema de operación de BBDD

Implementación: Gestión de XML

La carga inicial de los puntos de interés se realiza mediante un fichero XML. Mediante este diseño es fácil añadir nuevos puntos a la guía sin tener que modificar el código de la aplicación. Cuando se arranca la aplicación y se detecta que la versión de BBDD ha cambiado, automáticamente se lee el fichero XML de puntos de interés y se realiza una nueva carga que actualiza estos puntos.

Se ha creado la clase *XMLReaderHelper* para aislar a otras clase de los detalles de la lectura y tratamiento del fichero XML.

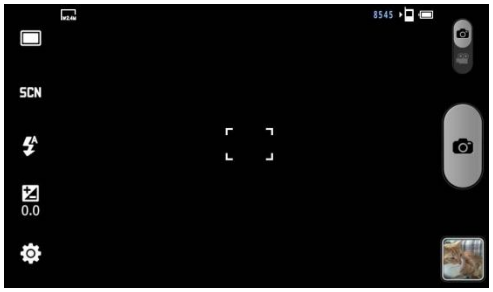


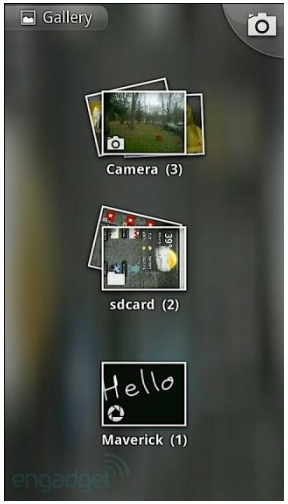

```
<listaPOI>
    <poi nombre="El Prado" latitud="40.413858" longitud="-3.692386" tipo="1" precio="7" horario="9:00 h - 20:00 h">
        <imagen>pre_elprado</imagen>
        <descBreve>
        <![CDATA[
        Considerado el noveno museo del mundo por importancia. Reune una amplia colección del realismo español con V
        ]]>
        </descBreve>
        <descLarga>
        <![CDATA[
        El Museo Nacional del Prado, ubicado en Madrid, España, es uno de los más importantes y más visitados del mu
        que posee las mejores colecciones que existen a nivel mundial, a lo que hay que sumar destacados conjuntos d
        Al igual que otros grandes museos europeos, como el Louvre de París y los Uffizi de Florencia, el Prado debe
        El Prado no es un museo enciclopédico al estilo del Museo del Louvre, la National Gallery de Londres, o incl
        Las escuelas pictóricas de España, Flandes e Italia (sobre todo Venecia) ostentan el protagonismo en el Prad
        Aunque sean aspectos menos conocidos, cuenta también con una importante sección de Artes decorativas (Tesoro
        ]]>
        </descLarga>
        <listadoImagenesGaleria>
            <imagenGaleria>fraguavulcano_velazquez_b</imagenGaleria>
            <imagenGaleria>hilanderas_velazquez_b</imagenGaleria>
            <imagenGaleria>meninas_velazquez_b</imagenGaleria>
            <imagenGaleria>rendicionbreda_velazquez_b</imagenGaleria>
            <imagenGaleria>venusespejo_velazquez_b</imagenGaleria>
            <imagenGaleria>esopo_velazquez_b</imagenGaleria>
        </listadoImagenesGaleria>
    </poi>
```

Implementación: Gestión de imágenes

La guía facilita la adición de puntos definidos por el usuario. Estos puntos de usuario pueden llevar asociadas imágenes cuya procedencia puede ser:

1. Cámara fotográfica
2. Galería de imágenes





Utilizando la clase MediaStore, se obtiene el acceso a la cámara o galería de imágenes del terminal.

Implementación: SD Card

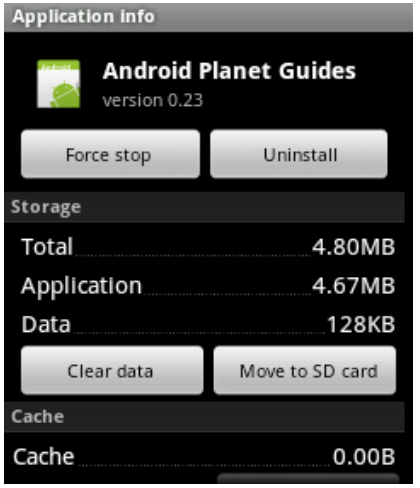
Se ha utilizado la tarjeta externa con dos fines:

1. Almacenar las fotografías referentes a los puntos de interés tomadas por el usuario.
2. Almacenar la propia aplicación para dejar libre memoria interna en el terminal

Es necesario solicitar permisos de escritura en memoria externa al usuario. Este proceso se realiza durante la instalación de la aplicación y el permiso solicitado viene registrado en el fichero AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="es.guia.interfaz"
      android:versionCode="1"
      android:versionName="0.29"
      android:installLocation="preferExternal">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Permisos de escritura y ubicación



Administración de almacenamiento