

CajerosCeres

Arquitecturas software para entornos
empresariales

Alejandro Rodríguez Escudero
4/4/2017

Contenido

1 INFORMACIÓN GENERAL	3
1.1 FUNCIONALIDAD PRIMARIA	3
1.2 USUARIOS OBJETIVOS	3
2 ANÁLISIS	3
2.1 REQUISITOS FUNCIONALES	3
2.2 REQUISITOS NO FUNCIONALES	4
2.3 REQUISITOS DE DATOS.....	4
2.4 PRINCIPALES CASOS DE USO	5
3 DISEÑO	6
3.1 ARQUITECTURA	6
3.2 INTERFAZ DE USUARIO	7
3.2.1 MAPA DE NAVEGACIÓN	7
3.3.2 MOCKUP	8
4 IMPLEMENTACIÓN	9
4.1 PERSISTENCIA DE DATOS.....	9
4.2 ACCESO A DATOS EXTERNOS	9

1 INFORMACIÓN GENERAL

1.1 FUNCIONALIDAD PRIMARIA

CajerosCeres es una aplicación dirigida a consultar información acerca de los cajeros automáticos de las entidades bancarias en la ciudad de Cáceres.

La aplicación permite realizar una búsqueda de cajeros en modo lista y en el mapa, ya sean todos los cajeros de Cáceres o aquellos que pertenezcan a una determinada entidad bancaria seleccionada por el usuario. Atendiendo a la visualización en modo lista, la aplicación presenta la posibilidad de mostrar los cajeros ordenarlos por menor a mayor comisión (dependiendo de la entidad bancaria a la pertenezca el usuario), atendiendo a la distancia que separa al usuario de éstos y la visualización de aquellos que hayan sido asignados como favoritos. La visualización de cajeros en el mapa mostrará la localización de dichos cajeros con una pequeña descripción y un acceso a Google Map para establecer la ruta entre el usuario y el cajero seleccionado.

1.2 USUARIOS OBJETIVOS

La aplicación está dirigida a usuarios que quieran localizar y establecer una ruta con los cajeros automáticos de la ciudad de Cáceres y atendiendo a criterios como la distancia o la comisión por la retirada de efectivo. La gama de edad comprende entre 18 y 50 años, tanto mujeres y hombres con unas nociones básicas en tecnologías.

2 ANÁLISIS

2.1 REQUISITOS FUNCIONALES

La siguiente lista representa a los requisitos funcionales del sistema.

- RF 1 - Los usuarios podrán elegir la moneda económica con la que se mostrará la comisión de los cajeros atendiendo a la entidad bancaria del usuario (Euros o Libras).
- RF 2 - Los usuarios podrán seleccionar la entidad bancaria a la que pertenezcan o deseen usarla como entidad con la que comparar comisiones.
- RF 3 - Los usuarios podrán limitar la distancia máxima de búsqueda de cajeros atendiendo a su localización.
- RF 4 - Los usuarios podrán acceder a un apartado de ayuda con información sobre la aplicación.
- RF 5 - Los usuarios podrán seleccionar una entidad bancaria que será usada como filtro para buscar los cajeros automáticos.
- RF 6 - Los usuarios podrán ver el mapa con los cajeros automáticos que haya seleccionado.
- RF 7 - Los usuarios podrán acceder a la aplicación de Google Map que establecerá una ruta entre el usuario y el cajero automático seleccionado.
- RF 8 - Los usuarios podrán acceder a la aplicación de Google Map que mostrará el cajero automático seleccionado.
- RF 9 - Los usuarios podrán acceder a un listado de cajeros automáticos ordenados por las distancias que separan al usuario de dichos cajeros.

- RF 10 - Los usuarios podrán acceder a un listado de cajeros automáticos ordenados por la comisión entre dicho cajero y la entidad bancaria del usuario.
- RF 11 - Los usuarios podrán acceder a un listado con los cajeros automáticos favoritos.
- RF 12 - Los usuarios podrán asignar como favorito un cajero automático.
- RF 13 - Los usuarios podrán desasignar como favorito un cajero automático.
- RF 14 - Los usuarios podrán ver una foto del cajero seleccionado.

*** Servicios que actualiza la base de datos de los cajeros, obteniendo el JSON y actualizando la base de datos de los cajeros al iniciar la aplicación.

*** Carga de los valores de las comisiones estáticos en la base de datos de entidades bancarias al iniciar la aplicación.

*** Obtención de la localización del usuario al acceder a la ventana de búsqueda.

*** Colores (rojo, amarillo o verde) de la distancia atendiendo a si se encuentra a menos de 100m, entre 100m y 500m o a más de 500m

*** Color verde en caso de que la comisión sea 0

2.2 REQUISITOS NO FUNCIONALES

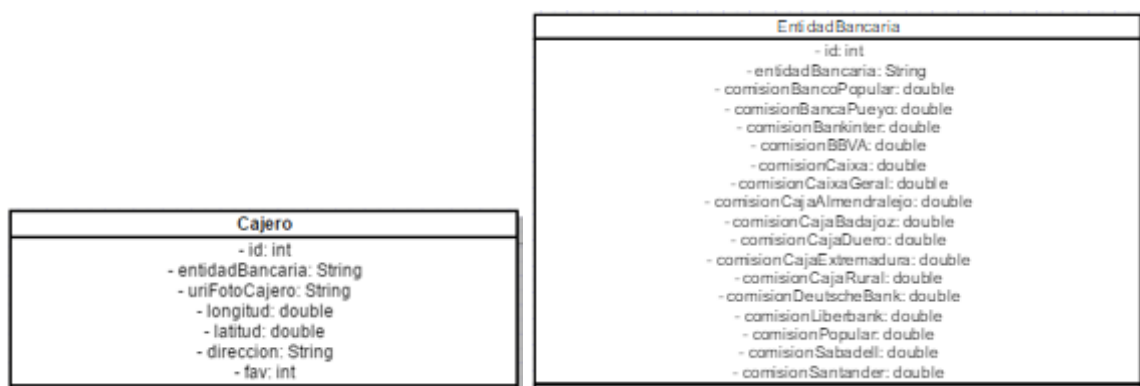
Listado de los requisitos no funcionales del sistema.

- RNF 1 - Para que la aplicación funcione correctamente, el usuario debe permitir el acceso a internet y localización.
- RNF 2 - La interfaz de la aplicación es amigable.
- RNF 3 - La aplicación debe de tener 3MB libres para soportar las bases de datos locales.
- RNF 4 - La carga de imagen del cajero automático no supera los 5 segundos.

2.3 REQUISITOS DE DATOS

La información que necesita ser almacenada será la referente a la información de cada cajero y las entidades de dichos cajeros con las comisiones respecto a los otros.

La información se guardará en dos bases de datos, que contendrán las operaciones POJO.

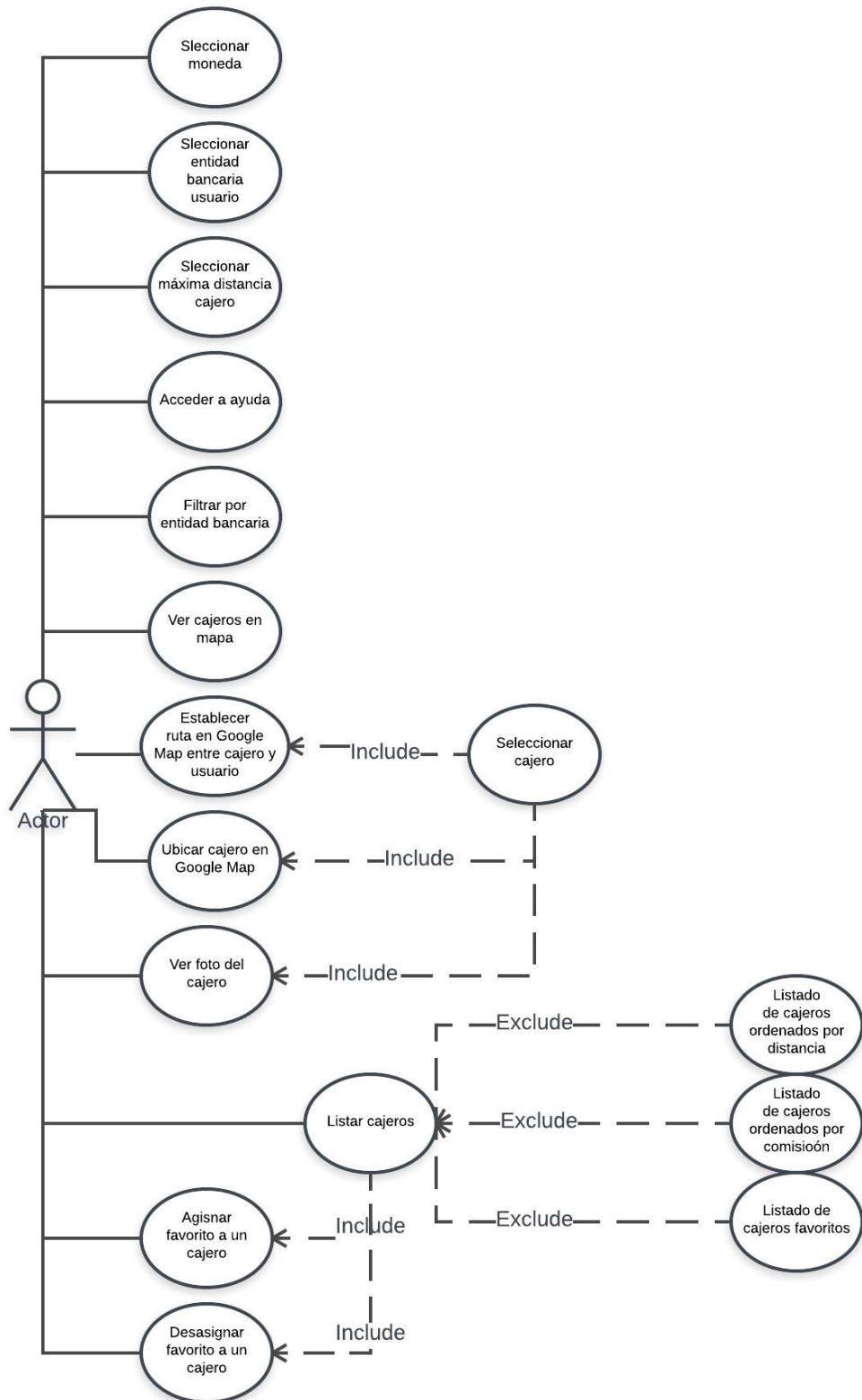


Base de datos Cajeros

Base de datos Entidades bancarias

2.4 PRINCIPALES CASOS DE USO

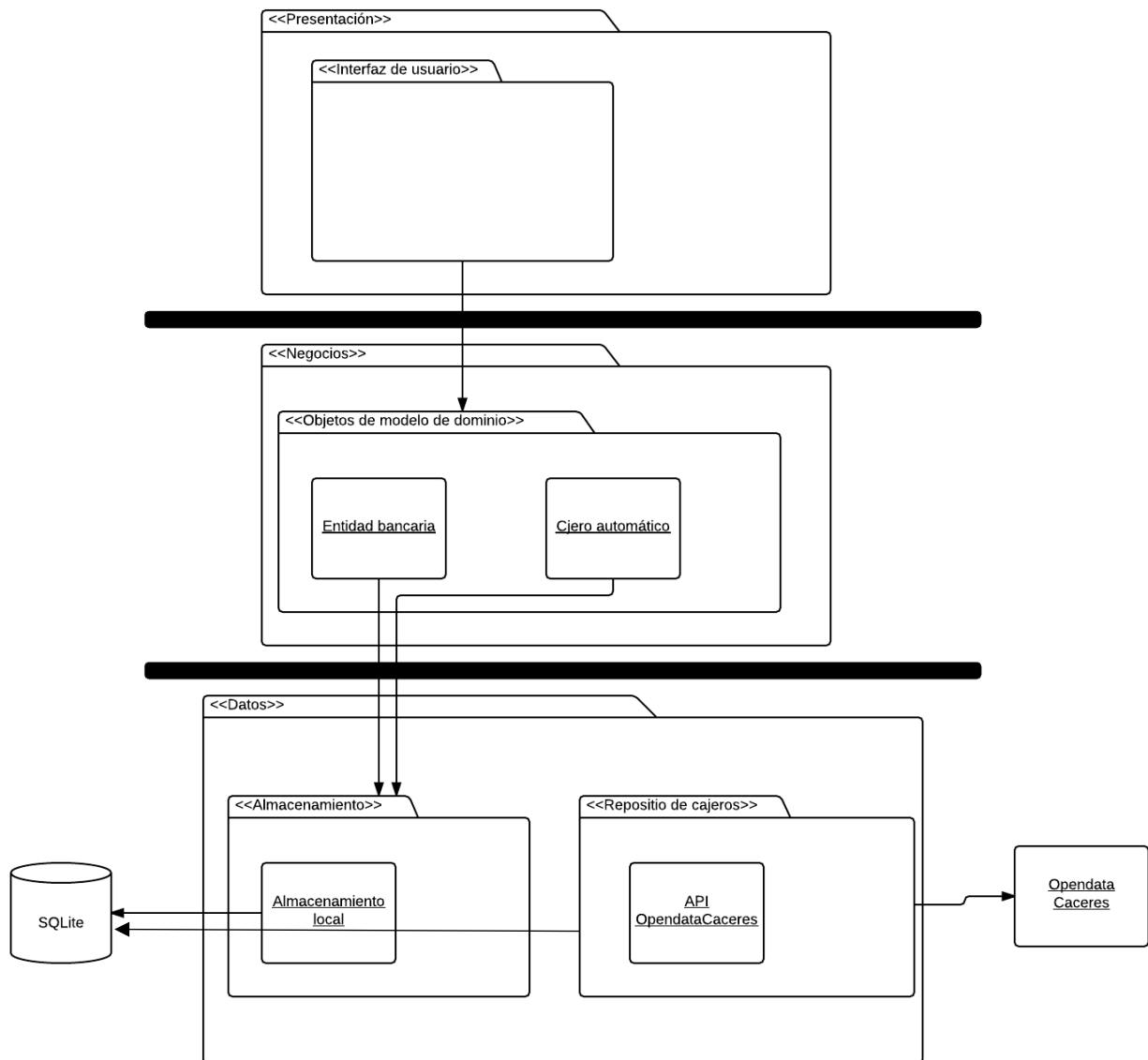
La siguiente figura representa los casos de uso que puede realizar un actor (usuario), mencionados anteriormente en el apartado 2.1



3 DISEÑO

3.1 ARQUITECTURA

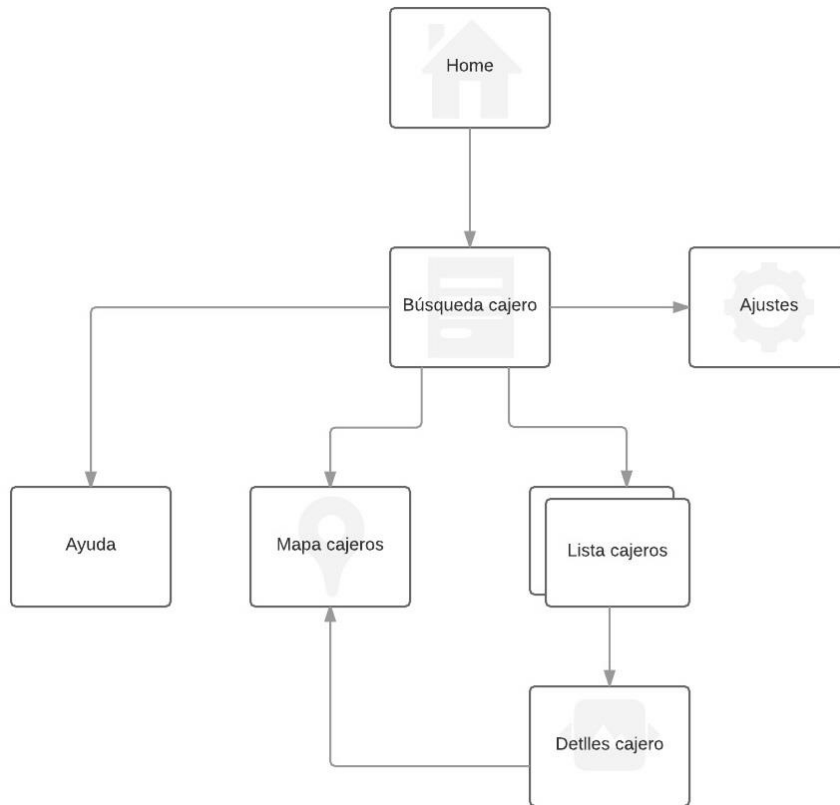
La aplicación consta de dos tipos de modelos, entidad bancaria y cajero automático, administrados por una base de datos local (SQLite). Por otra parte, la información que presentará en la búsqueda cajeros, se obtiene de API's externas que ofrecen sus servicios, como es el caso de la API de Opendata Cáceres. La base de datos local tendrá las operaciones básicas como añadir o borrar.



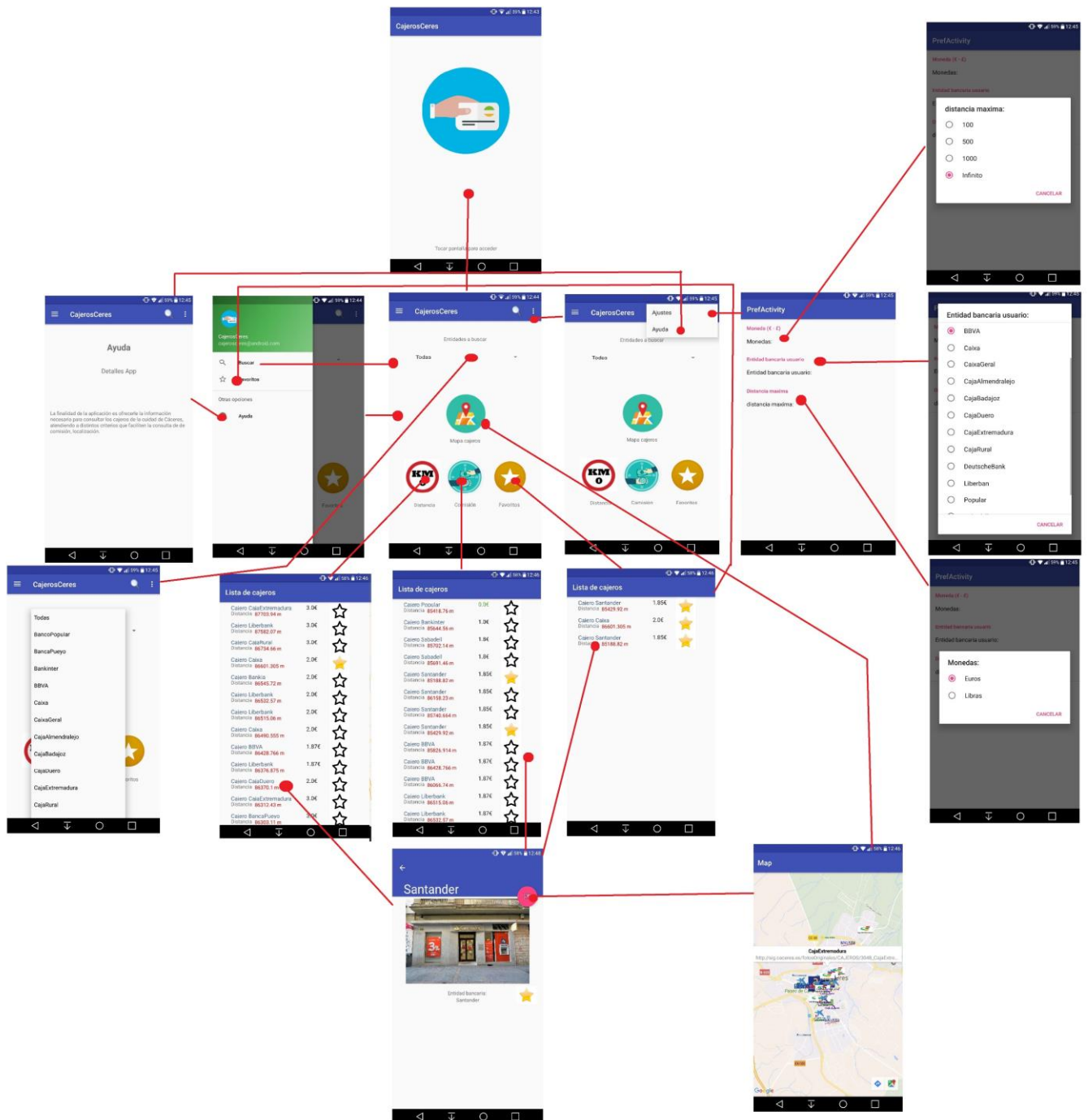
3.2 INTERFAZ DE USUARIO

3.2.1 MAPA DE NAVEGACIÓN

El presente diagrama muestra de forma visual la estructura de los módulos que componen la aplicación, y una breve indicación de que acciones llevan de uno a otro.



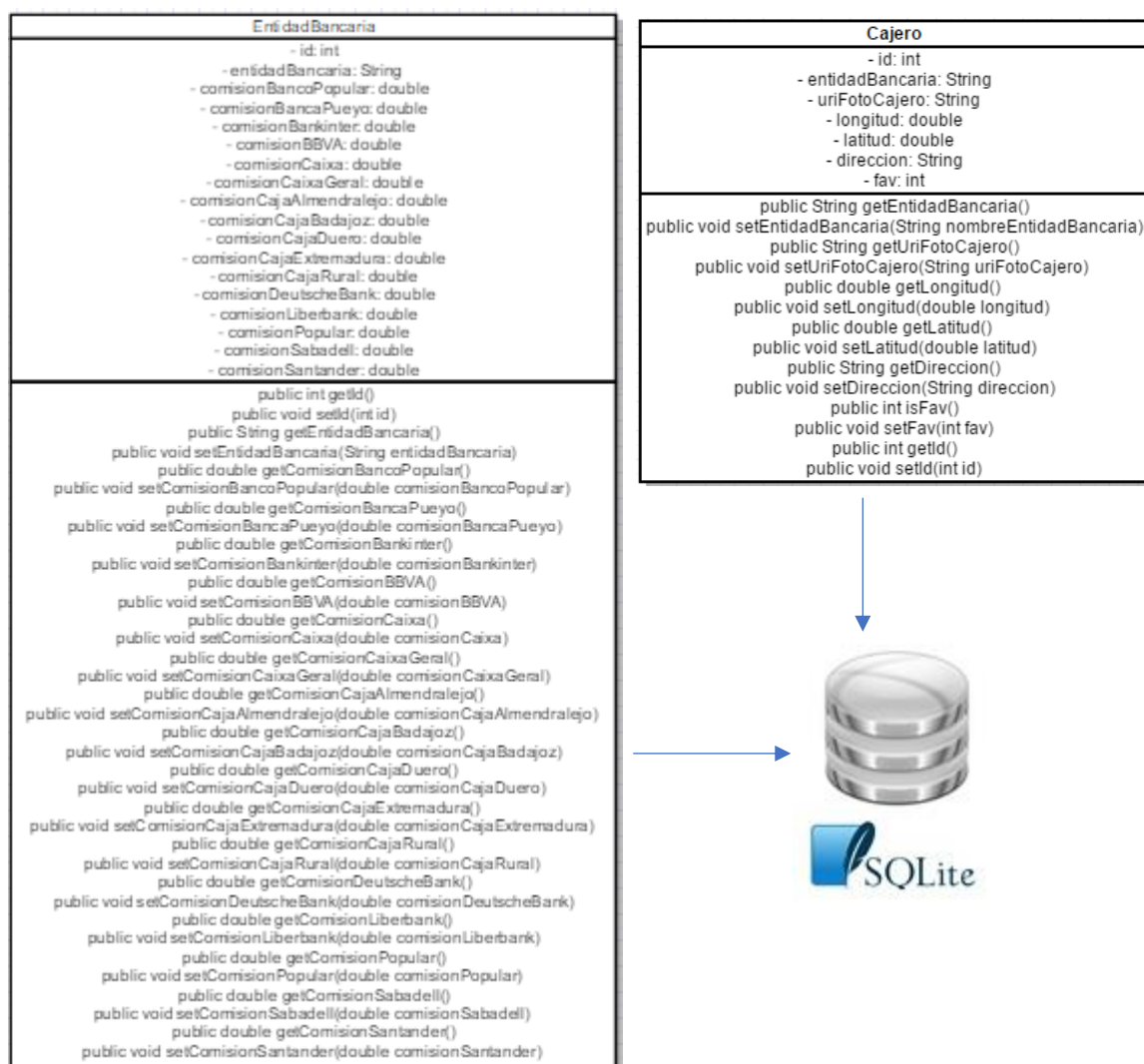
3.3.2 MOCKUP



4 IMPLEMENTACIÓN

4.1 PERSISTENCIA DE DATOS

Para la persistencia de los datos en la aplicación, se ha usado un base de datos interna, SQLite, debido al escaso tamaño que necesita y cumple las necesidades que la aplicación necesita para guardas las entidades bancarias y los cajeros automáticos.



4.2 ACCESO A DATOS EXTERNOS

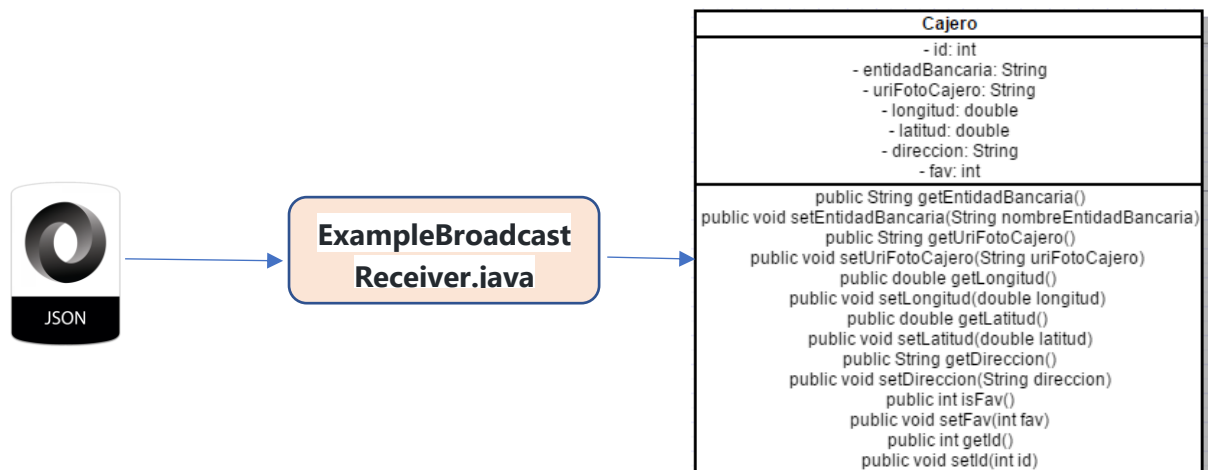
Para el acceso a datos externos para obtener los cajeros automáticos que hay en Cáceres, utilizamos un cliente API publicado por el OPENDATA de Cáceres <http://opendata.caceres.es/dataset/cajeros-automaticos-caceres>. Realizando una petición a la URL <http://opendata.caceres.es/GetData/GetData?dataset=om:CajeroAutomatico&format=json> que devuelve un JSON con un conjunto de datos de cajeros.

El tratamiento del JSON fue un quebradero de cabeza, que se solucionó realizando un parseador JSON a clases Java con una ayuda de visualización aportada por <http://jsonviewer.stack.hu/>

Ejemplo de cajero en JSON:

```
"bindings": [
{
  "uri": {
    "type": "uri",
    "value": "http://opendata.caceres.es/recurso/economia/banca/CajeroAutomatico/0"
  },
  "om_entidadBancaria": {
    "type": "literal",
    "value": "CajaRural"
  },
  "om_tieneEnlaceSIG": {
    "type": "uri",
    "value": "http://sig.caceres.es/fotosOriginales/CAJEROS/3639_CajaRural_1.jpg"
  },
  "geo_long": {
    "datatype": "http://www.w3.org/2001/XMLSchema#decimal",
    "type": "typed-literal",
    "value": "-6.381278"
  },
  "geo_lat": {
    "datatype": "http://www.w3.org/2001/XMLSchema#decimal",
    "type": "typed-literal",
    "value": "39.456235"
  },
  "om_situadoEnVia": {
    "type": "uri",
    "value": "http://opendata.caceres.es/recurso/urbanismo-infraestructuras/vias/Via/calle_3639"
  },
  "rdfs_label": {
    "type": "literal",
    "value": "Cajero CajaRural"
  }
}
],
```

Los datos de los cajeros son introducidos en la clase java correspondiente y añadidos a la base de datos SQLite mediante la clase ExampleBroadcastReceiver.java, que contiene un manejador del cual obtiene los datos del JSON y los trata para introducirlos en la clase Cajero.java



*** Meter código sobre el manejador?