

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 SISTEMA	4
2.4 REQUISITOS DE DATOS	4
2.5 PRINCIPALES CASOS DE USO	6
3 DISEÑO	7
3.1 ARQUITECTURA	7
3.2 INTERFAZ DE USUARIO	8
3.2.1 MAPA DE NAVEGACIÓN	8
3.3.2 MOCKUP	9
4 IMPLEMENTACIÓN	10
4.1 PERSISTENCIA DE DATOS	10
4.2 ACCESO A DATOS EXTERNOS	11
4.3 PATRONES DE DISEÑO	13
4.4 PRINCIPALES DECISIONES DE DISEÑO	13
4.4.1 PLANTEAMIENTO DEL DESARROLLO DE LA IDEA	13
4.4.2 PROBLEMAS ENCONTRADOS	13
5 REFERENCIAS	14
6 CONCLUSIONES	15

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. Ofrece la posibilidad de buscar aquellos cajeros que cobren menos comisión o los cajeros más cercanos de una entidad bancaria deseada. Ya sea en el mapa, pudiendo establecer una ruta con Google Maps o mostrando una imagen del cajero para su localización más sencilla.

1.2 USUARIOS OBJETIVOS

La aplicación está dirigida a usuarios de edades comprendidas entre 18 y 60 años, debido a que son los usuarios que tienen más probabilidad de tener una tarjeta de crédito y estén interesados en localizar los cajeros automáticos de la ciudad de Cáceres o establecer una ruta. Esta franja se ha restringido a un máximo de 60 debido a que el factor de aprendizaje tecnológico juega un papel importante en edades avanzadas, junto con la necesidad de tener un smartphone, conexión a internet y un mínimo conocimiento tecnológico. La restricción de edad inferior se debe a que usuarios con menos de 18 años tienen una baja probabilidad de tener una tarjeta de crédito.

2 ANÁLISIS

2.1 REQUISITOS FUNCIONALES

Definición de los servicios que el sistema debe proporcionar, cómo debe reaccionar a una entrada particular y cómo se debe a una entrada particular y cómo se debe comportar ante situaciones particulares. La siguiente lista representa a los requisitos funcionales del sistema.

- RF 1 - Los usuarios podrán seleccionar la entidad bancaria a la que pertenezcan o deseen usarla como entidad con la que comparar comisiones.
- RF 2 - Los usuarios podrán acceder a un apartado de ayuda con información sobre la aplicación.
- RF 3 - Los usuarios podrán seleccionar una entidad bancaria que será usada como filtro para buscar los cajeros automáticos.
- RF 4 - Los usuarios podrán ver el mapa con los cajeros automáticos que haya seleccionado.
- RF 5 - Los usuarios podrán acceder a la aplicación de Google Maps que establecerá una ruta entre el usuario y el cajero automático seleccionado.
- RF 6 - Los usuarios podrán acceder a la aplicación de Google Maps que mostrará el cajero automático seleccionado.
- RF 7 - Los usuarios podrán acceder a un listado de cajeros automáticos ordenados por las distancias que separan al usuario de dichos cajeros.
- RF 8 - Los usuarios podrán acceder a un listado de cajeros automáticos ordenados por la comisión entre dicho cajero y la entidad bancaria del usuario.
- RF 9 - Los usuarios podrán acceder a un listado con los cajeros automáticos favoritos.
- RF 10 - Los usuarios podrán asignar como favorito un cajero automático.
- RF 11 - Los usuarios podrán desasignar como favorito un cajero automático.
- RF 12 - Los usuarios podrán ver una foto del cajero seleccionado.
- RF 13 - Los usuarios mantendrán actualizada la base de datos de cajeros automáticos.

2.2 REQUISITOS NO FUNCIONALES

Restricciones que afectan a los servicios o funciones del sistema. 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 SISTEMA

Establece con detalle los servicios y restricciones de sistema. Es una base para el diseño e implementación detallados. Listado con los requerimientos del sistema:

- RS 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).
- RS 2 - Los usuarios podrán limitar la distancia máxima de búsqueda de cajeros atendiendo a su localización.
- RS 3 – Los usuarios podrán ver la distancia que los separa de los cajeros con diferentes colores, atendiendo a la distancia que los separa. Color rojo representará que el cajero se encuentra a más de quinientos metros, amarillo entre cien metros y quinientos metros, y verde a menos de cien metros.

2.4 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.

Para los cajeros automáticos, los datos a guardar son:

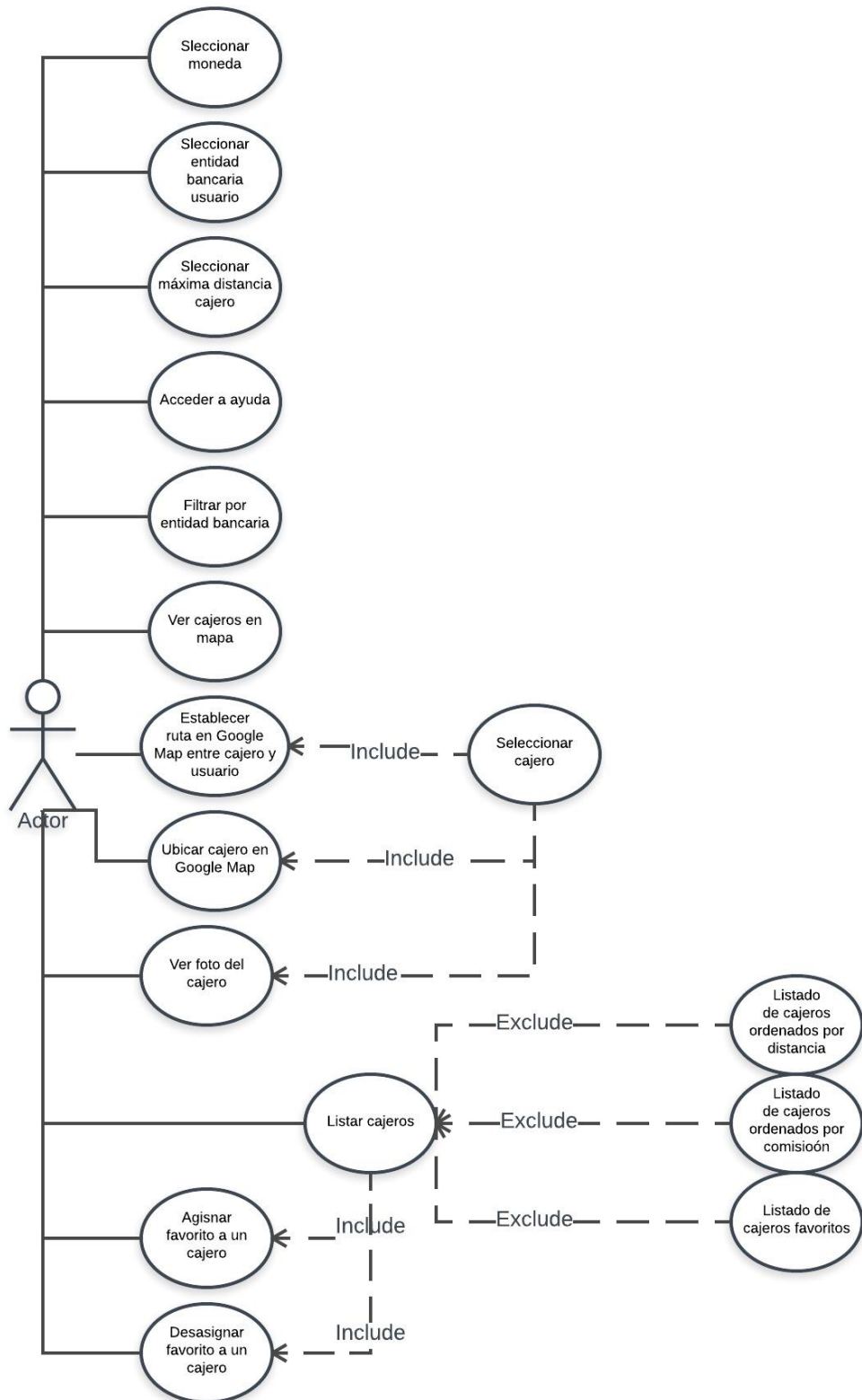
- **Id.** Identificador único para cada cajero automático.
- **EntidadBancaria.** Nombre de la entidad bancaria a la que pertenece dicho cajero automático.
- **UriFotoCajero.** Identificador de recursos uniforme que identifica a un recurso de la red donde se encuentra la foto del cajero automático.
- **Longitud.** Coordenada geográfica que referencia la longitud de un punto en la Tierra donde se encuentra un cajero automático.
- **Latitud.** Coordenada geográfica que referencia la latitud de un punto en la Tierra donde se encuentra un cajero automático.
- **Dirección.** Identificador de recursos uniforme que identifica a un recurso de la red donde se encuentra la dirección del cajero automático.
- **Fav.** Identifica al cajero automático como favorito o no favorito.

Los datos de las entidades bancarias son los siguientes:

- **Id.** Identificador único para cada entidad bancaria.
- **EntidadBancaria.** Nombre de la entidad bancaria.
- **comisionBancoPopular.** Comisión existente entre la entidad bancaria perteneciente al Id y el Banco Popular, con respecto a un 1€.
- **comisionBancaPueyo.** Comisión existente entre la entidad bancaria perteneciente al Id y la Banca Pueyo, con respecto a un 1€.
- **comisionBankinter.** Comisión existente entre la entidad bancaria perteneciente al Id y Bankinter, con respecto a un 1€.
- **comisionBBVA.** Comisión existente entre la entidad bancaria perteneciente al Id y BBVA, con respecto a un 1€.
- **comisionCaixa.** Comisión existente entre la entidad bancaria perteneciente al Id y la Caixa, con respecto a un 1€.
- **comisionCaixaGeral.** Comisión existente entre la entidad bancaria perteneciente al Id y la Caixa Geral, con respecto a un 1€.
- **comisionCajaAlmendalejo.** Comisión existente entre la entidad bancaria perteneciente al Id y Caja Almendralejo, con respecto a un 1€.
- **comisionCajaBadajoz.** Comisión existente entre la entidad bancaria perteneciente al Id y Caja Badajoz, con respecto a un 1€.
- **comisionCajaDuero.** Comisión existente entre la entidad bancaria perteneciente al Id y Caja Duero, con respecto a un 1€.
- **comisionCajaExtremadura.** Comisión existente entre la entidad bancaria perteneciente al Id y Caja Extremadura, con respecto a un 1€.
- **comisionCajaRural.** Comisión existente entre la entidad bancaria perteneciente al Id y Caja Rural, con respecto a un 1€.
- **comisionDeutscheBank.** Comisión existente entre la entidad bancaria perteneciente al Id y DeutschBank, con respecto a un 1€.
- **comisionLiberbank.** Comisión existente entre la entidad bancaria perteneciente al Id y Liberbank, con respecto a un 1€.
- **comisionPopular.** Comisión existente entre la entidad bancaria perteneciente al Id y Popular, con respecto a un 1€.
- **comisionSabadell.** Comisión existente entre la entidad bancaria perteneciente al Id y Sabadell, con respecto a un 1€.
- **comisionSantander.** Comisión existente entre la entidad bancaria perteneciente al Id y Santander, con respecto a un 1€.

2.5 PRINCIPALES CASOS DE USO

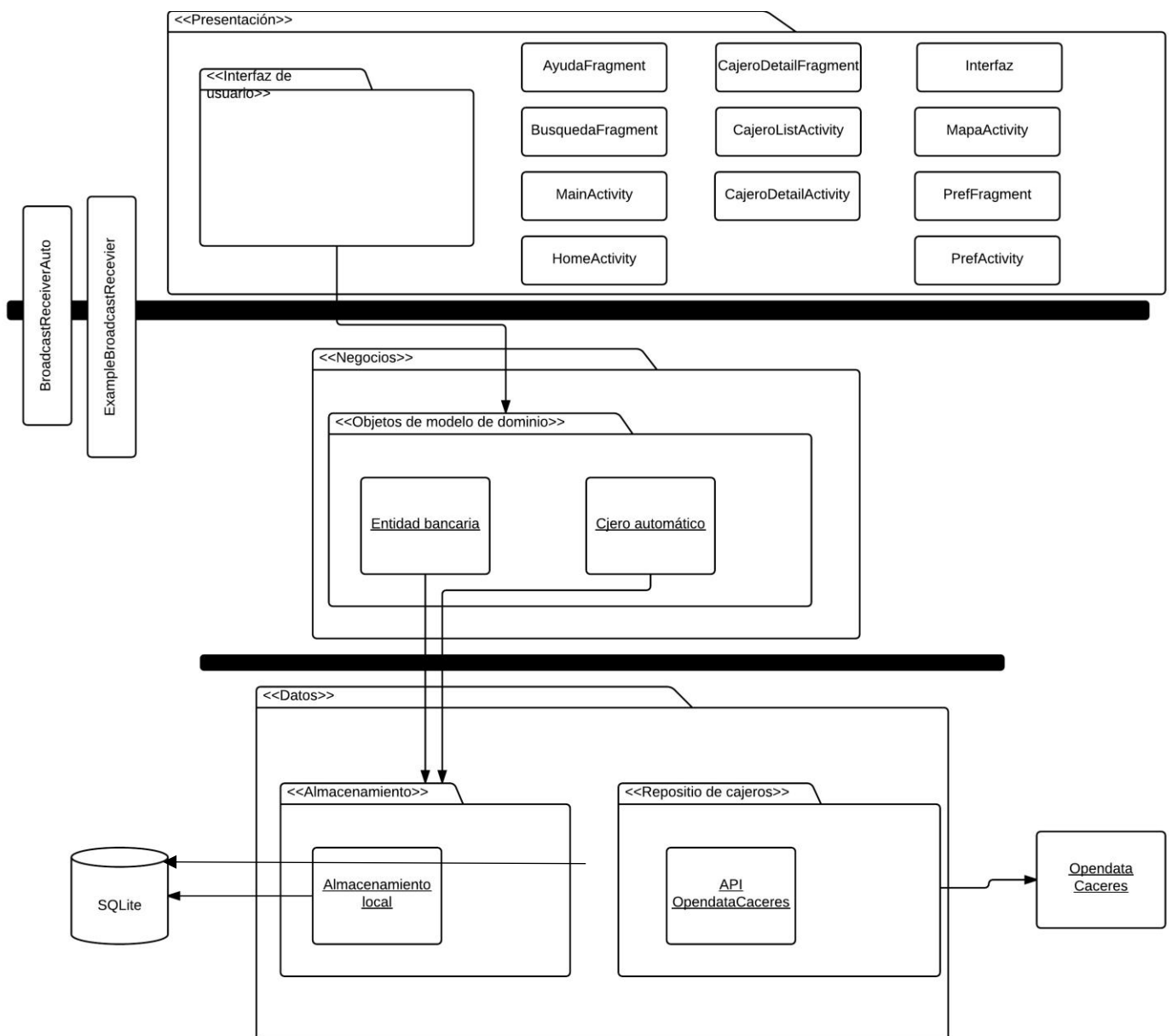
La siguiente figura representa los casos de uso que puede realizar un actor (usuario), mencionados anteriormente en el apartado 2.1



3 DISEÑO

3.1 ARQUITECTURA

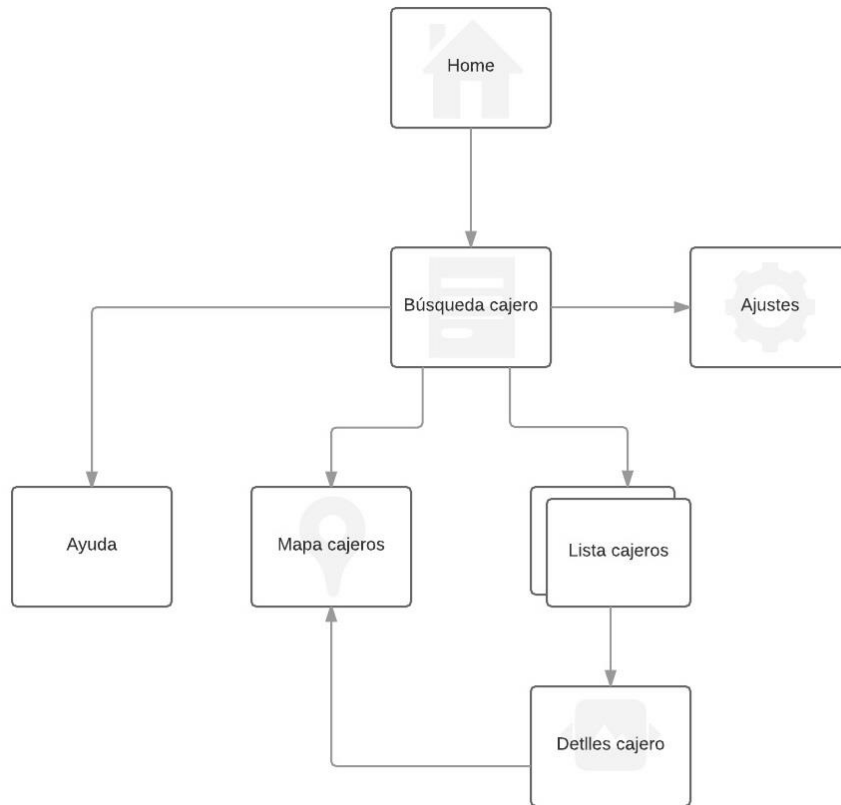
La aplicación consta de dos tipos de modelos, entidad bancaria y cajero automático, administrados por una base de datos local (SQLite). Por otra parte, la información que presentará en la búsqueda cajeros, se obtiene de API's externas que ofrecen sus servicios, como es el caso de la API de Opendata Cáceres. La base de datos local tendrá las operaciones básicas como añadir o borrar. Los servicios *BroadcastReceiverAuto* y *ExampleBroadcastReceiver* encargados de actualizar la base de datos de cajeros automáticos cada 24h y al inicio de la app.



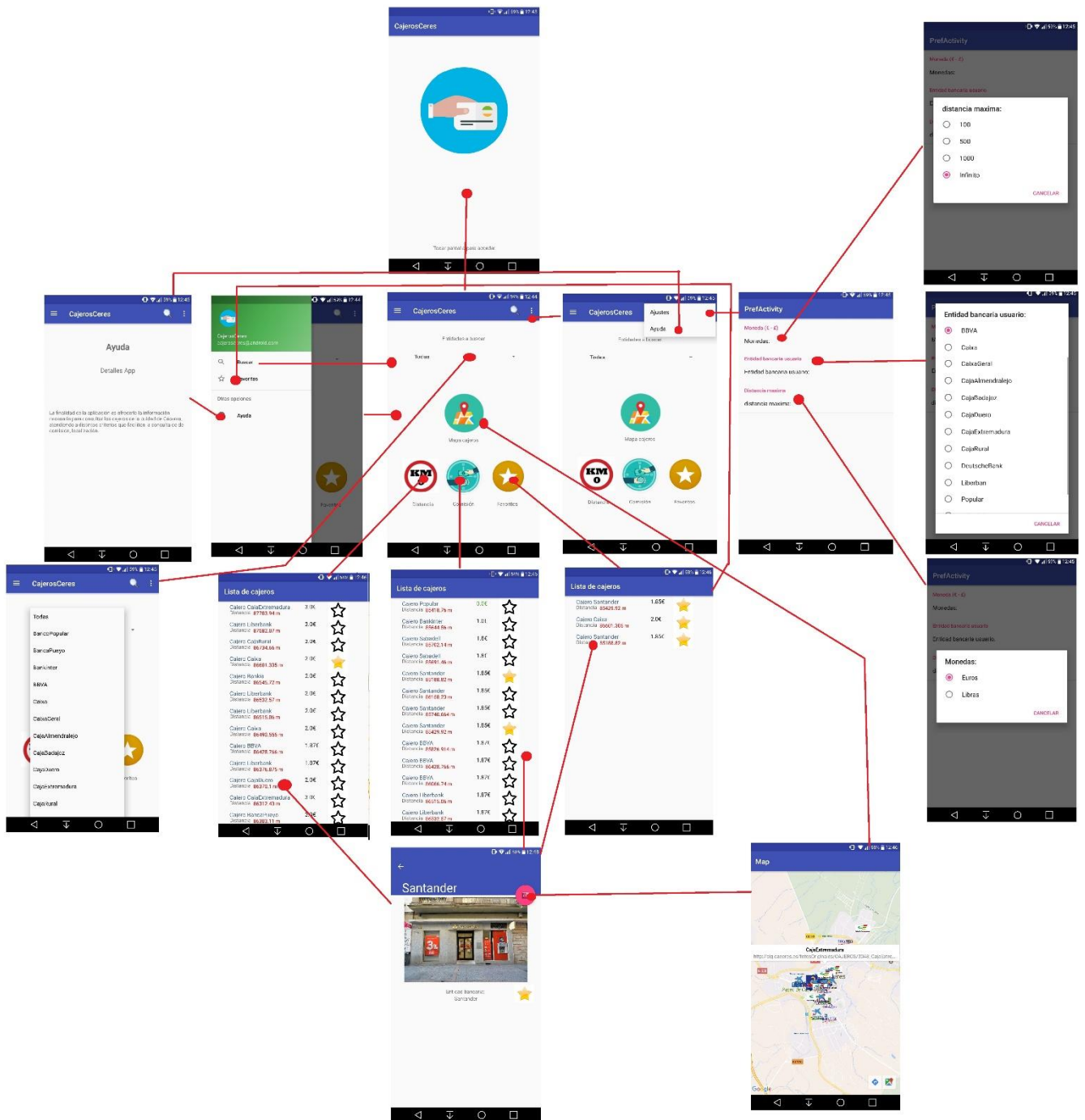
3.2 INTERFAZ DE USUARIO

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.

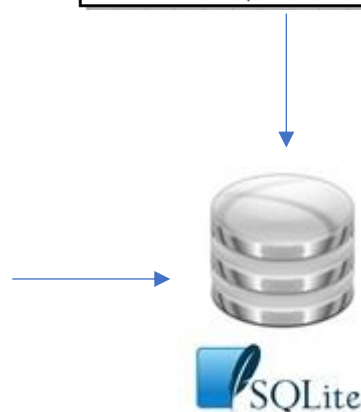
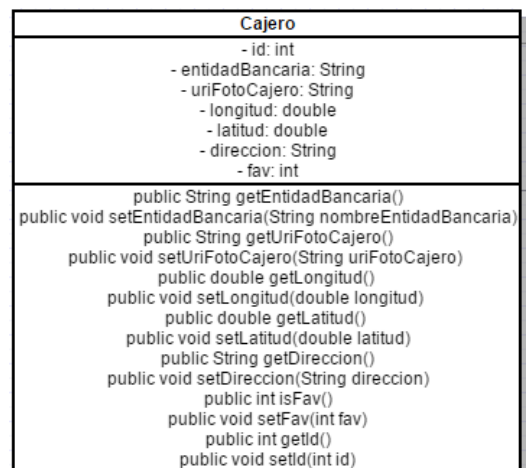
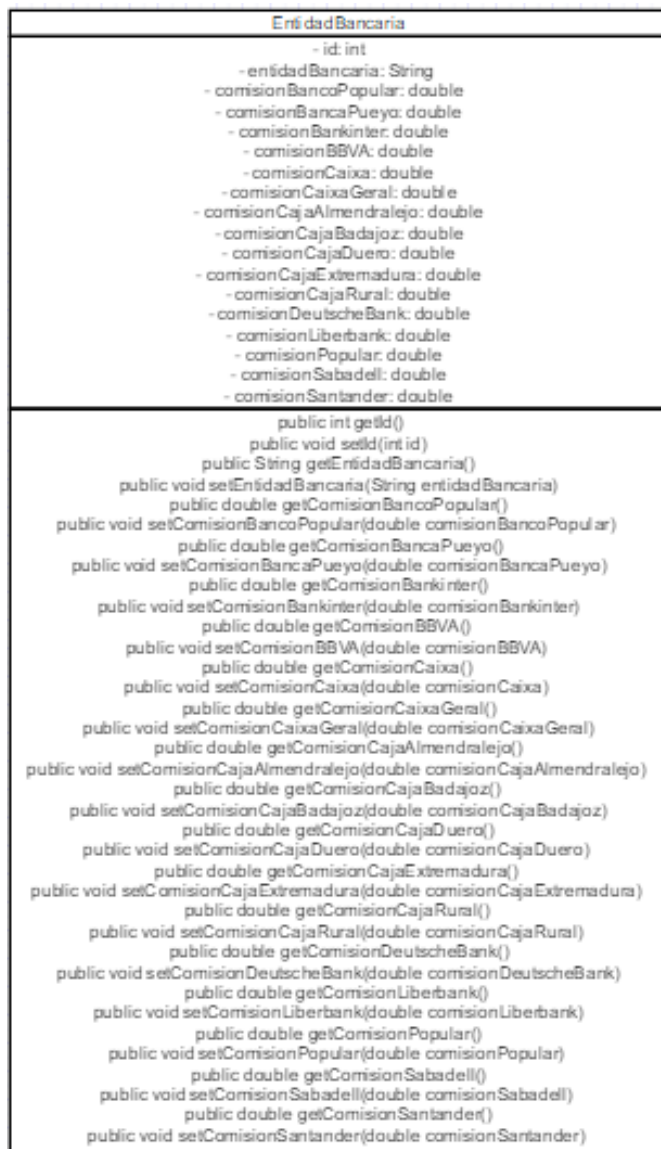
La base de datos de entidades bancarias se usa para volcar los datos de las entidades bancarias y las comisiones que tienen respecto a la retirada de 1€ en el cajero del resto de las entidades bancarias. Se ha optado por crear esta base de datos para en una posterior actualización, el volcado de datos venga de un servicio API REST y no de unos datos locales introducidos en la aplicación. Dicha ampliación consistirá en un servicio que descargue los datos de un servidor al iniciar la aplicación y los guarde localmente para no demorar el tiempo de consulta de estos datos con una petición al servidor.

La otra base de datos, la perteneciente a las entidades bancarias, es usada para volcar los datos de los cajeros automáticos de la ciudad de Cáceres tras una petición REST al cliente API de Opendata. Dichos datos se guardan en local al iniciar la aplicación para optimizar el tiempo de las consultar y se actualizará cierto tiempo.

Las clases donde se localiza el código de acceso las clases de datos son:

- **ExampleBroadcastReceiver** en la creación de la tabla de comisiones
- **ExampleBroadcastReceiver** en la creación y actualización de la tabla de cajeros automáticos.
- **CajeroDetailFragment** para el acceso a los datos a mostrar sobre un cajero automático y la modificación del atributo favorito.
- **CajeroListActivity** para el acceso a los datos a mostrar en la lista de cajeros automáticos y la modificación del atributo favorito.
- **CajeroListActivity** para el acceso a los atributos de comisiones a mostrar en la lista de cajeros automáticos.
- **MapaActivity** para el acceso a los datos de los cajeros automáticos para representar dichos cajeros en el mapa.
- **MapaActivity** para el acceso a los datos de las comisiones de las entidades bancarias para mostrarlas en los cajeros del mapa.

Cada 24 horas y al inicio de la App, la clase BroadcastRecevierAuto se encargará de actualizar la base de datos de cajeros automáticos, manteniendo así los cajeros automáticos actualizados.



4.2 ACCESO A DATOS EXTERNOS

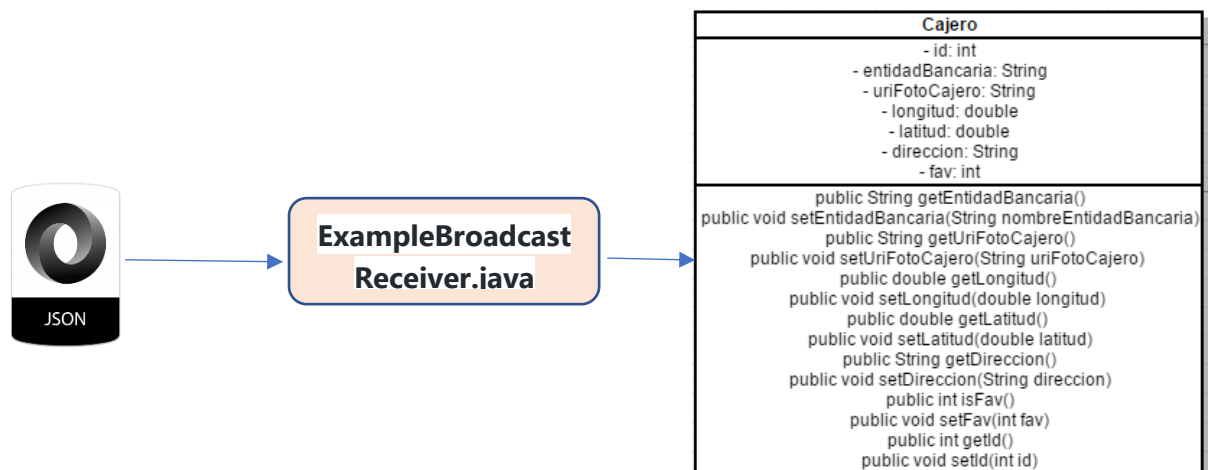
Para el acceso a datos externos para obtener los cajeros automáticos que hay en Cáceres, utilizamos un cliente API publicado por el OPENDATA de Cáceres <http://opendata.caceres.es/dataset/cajeros-automaticos-caceres>. Realizando una petición a la URL <http://opendata.caceres.es/GetData/GetData?dataset=om:CajeroAutomatico&format=json>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



4.3 PATRONES DE DISEÑO

- **Navigation drawer:** Disponemos de este menú desplegable hacia la derecha, para que el usuario pueda ir a su cuenta personal, cambiar los ajustes y ver la información de ayuda de la aplicación.
- **Patrón de navegación ancestral:** Una funcionalidad muy demandada en las aplicaciones por los usuarios, un botón para ir a la anterior pantalla, el botón back.

4.4 PRINCIPALES DECISIONES DE DISEÑO

4.4.1 PLANTEAMIENTO DEL DESARROLLO DE LA IDEA

En primer lugar, la aplicación se ha dividido en diferentes fases en función de la importancia que se le ha dado. Dando prioridad a la funcionalidad sobre la interfaz. Las partes más importantes que se ha estimado son las siguientes:

- Proporcionar al usuario la facilidad de moverse por la aplicación, para ello, se ha desarrollado un menú lateral con la mayoría de acciones que se pueden realizar, implementadas cada una en un fragment diferente, estos fragment se conectarán entre sí por medio de una interfaz que los enlazará.
- Para la forma de almacenar los diversos cajeros automáticos que tendrá el usuario en función de su búsqueda, se ha implementado un activity maestro/detalle que permita tener una lista de ellos y que, sobre cada uno, tenga sus atributos de la forma más detallada. Dicha lista, tendrá incluida una recyclerview, para los diversos resultados de la búsqueda en la lista.
- Dotar al usuario de la posibilidad de visualizar los cajeros en el mapa y poder establecer una ruta entre el cajero automático y el usuario mediante el uso de Google Maps.
- Para las bases de datos, se ha desarrollado una base de datos local SQLITE, para almacenar todo lo relacionado con los cajeros automáticos y entidades bancarias, tanto la lista de resultados de la búsqueda, como la de cajeros favoritos.
- Descarga de datos en formato JSON en segundo plano, para liberar de procesamiento al thread principal.
- Establecimiento de los precios de comisiones atendiendo a lo que dicta la Comisión Nacional de los Mercados y la Competencia.

4.4.2 PROBLEMAS ENCONTRADOS

Durante el desarrollo de la aplicación se han encontrado con errores que se ha podido solucionar. Algunos son los siguientes:

- Al pulsar el botón Back desde los detalles de un cajero automático, la aplicación fallaba debido a que la lista de cajeros automáticos necesitaba un conjunto de valores para poder aplicar el filtro en la lista de cajeros automáticos que tenía la lista por la cual se accedía al detalle del cajero

automático. La solución ha consistido en guardar dichos valores en un archivo de preferencias mediante el uso de SharedPreferences.

- Obtener la localización geográfica del usuario y el uso de permisos, los cuales son necesarios a partir de las versiones 6.0. Se ha solucionado controlando que se aceptan los permisos y usando la clase Criteria para obtener el mejor proveedor.
- Integración de imágenes desde una URL, solucionado mediante el uso de la librería Picasso.
- Ejecución automática al iniciar la App de los servicios de actualización de la base de datos de cajeros automáticos. Se ha decidido ejecutar dichos servicios manualmente al iniciar la app.
- Localización exacta de la ubicación del usuario al utilizar la clase Criteria en vez de la clase de Google. Se ha decidido utilizar esta clase por la posibilidad de seleccionar el mejor proveedor, ya sea por GPS o por redes.

5 REFERENCIAS

<http://square.github.io/picasso/> , integración de imágenes desde una URL.

<https://www.fluidui.com/editor/live/> , creación del Mockup.

<https://www.lucidchart.com/> , creación de diagramas de caso de uso, arquitectura y mapa de navegación.

<http://www.hermosaprogramacion.com/2015/01/android-json-parsing/> , parsear datos Json en Android con JsonReader y Gson.

<http://opendata.caceres.es/dataset/cajeros-automaticos-caceres> , Opendata de Cáceres sobre los cajeros automáticos.

<http://opendata.caceres.es/dataset/cajeros-automaticos-caceres/resource/757686f9-8d4b-4a8b-b2b4-6839957dcc19> , información sobre el Json de cajeros automáticos del Opendata de Cáceres.

https://docs.google.com/presentation/d/1O6V1b4LwmJF4R_6RMUYUCPinGBcUHVcdfwkeH8jDwIq4/edit , Vistas de Módulo y estilo de capas(multipapa), ASEE.

<https://docs.google.com/presentation/d/18gfMjotlbz19AlqyDiEGPjFSGLPJC6HEFZHpZRZ-9kM/edit#slide=id.p4> , Diseño de interfaz de usuario e interacción, ASEE.

https://docs.google.com/presentation/d/1vbTNDtpB03B8_t18fNCEA63jpFTDA_NvMMkfpKWMCSY/edit#slide=id.p4 , Modelo de tres capas (Capa de negocios), ASEE.

<https://docs.google.com/presentation/d/1NzzaAI3g2k7Qq4BOLPWjLLKmk7LZyb0cPZdIY4JASoc/edit#slide=id.p4> , Mapeo a BD relacionales (Fuentes de datos), ASEE.

<https://docs.google.com/presentation/d/1NzzaAI3g2k7Qq4BOLPWjLLKmk7LZyb0cPZdIY4JASoc/edit#slide=id.p80> , Capa de fuente de datos, ASEE.

<http://www.elmundo.es/economia/2016/07/20/578e87d8e2704e9c3e8b46ec.html> , Mapa de comisiones en los bancos.

<https://developer.android.com/training/permissions/requesting.html> , Solicitar permisos en tiempo de ejecución.

<http://stackoverflow.com/questions/32184156/how-to-add-progress-bar-to-picasso-library> , Añadir barra de progreso a Picasso.

<https://github.com/AdguardTeam/AdguardForAndroid/issues/183> , BroadcastReceiver Boot_completed.

<http://stackoverflow.com/questions/39360477/broadcastreceiver-receiving-boot-completed-on-samsung-lg-devices-but-not-on-len> , BroadcastReceiver Boot_completed en LG.

6 CONCLUSIONES

Me ha resultado interesante profundizar en un campo tan actual como son las plataformas móviles, precisamente por el hecho de trabajar en una tecnología en constante desarrollo y cuyo resultado se puede comprobar. La plataforma Android es bastante compleja y su curva de aprendizaje puede ser lenta al principio, no obstante, considero alcanzado un mínimo para poder desempeñar la función de desarrollador Android en un futuro trabajo.

La aplicación que he desarrollado me sirve de incentivo para seguir desarrollando otras aplicaciones Android que me harán seguir profundizando, formándome y resolviendo problemas que no se me hayan planteado en este proyecto.

De la aplicación CajerosCeres, me quedo con el uso del cliente Api de Opendata Cáceres al ver el potencial que tiene y el uso de la base de datos SQLite.

Quedan pendientes algunas futuras ampliaciones para profundizar más en el mundo de Android. La posibilidad de mantener actualizadas las comisiones de las entidades bancarias, las cuales hasta ahora son estáticas y existe la posibilidad de crear un servicio y acceder alguna API que pueda crear y tener obtener los datos de manera dinámica