# MEMORIA DEL PROYECTO


Delfín Álvaro Miguel Gómez

Rodrigo Notario Pérez

David Ángel Mata Lorenzo

# ÍNDICE

# MANUAL DE INSTALACIÓN

## REQUISITOS MÍNIMOS

Para poder disfrutar de nuestra aplicación en tu dispositivo Android deberás disponer de un nivel de API mínima 15, la cual se corresponde con la versión 4.0.3 (*Ice Cream Sandwich)* y funcionará aproximadamente en el 97,4% de los dispositivos activos en el Google Play Store.

## PREPARACIONES PREVIAS

Antes de instalar la app de *CuentaCuentas*, debes activar la opción de permitir la instalación de aplicaciones de origen desconocido en tu dispositivo. Para ello tienes que ir a:

| Ajustes > Seguridad > Administración de dispositivos, y marcar *Orígenes desconocidos*. |
|---|

Ahora ya podrás instalar el APK de nuestra aplicación sin problemas.

## PERMISOS DE LA APLICACIÓN

La aplicación hará uso de los siguientes permisos en tu teléfono:

- Ubicación aproximada (basada en red) y ubicación precisa (basada en red y GPS)
- Leer y modificar/eliminar contenido de la tarjeta SD
- Leer la configuración de los servicios de Google
- Acceso completo a red
- Recibir datos de Internet
- Ver conexiones de red
- Impedir que el teléfono entre en modo de suspensión

## CONFIGURAR EL PROYECTO EN ANDROID STUDIO + BASE DE DATOS + PARTE DE TECNOLOGÍAS WEB

Nuestro proyecto incorpora una base de datos local. En la carpeta "Proyecto" encontraréis el proyecto de Android Studio y la base de datos de la aplicación. Para que funcione debéis de:

- Arrancar *Apache* y *Mysql*.
- Importar con *phpmyadmin* el archivo *cuentacuentas.sql* de la carpeta "Base de datos".
- Colocar la carpeta "CuentaCuentas" que se encuentra dentro de la carpeta "Proyecto Android Studio" en el directorio deseado. Abrir *Android Studio* y cargar este proyecto.
- Colocar los archivos "cuentaCuentas.html" y "cuentaCuentas.css" que se encuentran dentro de la carpeta "Tecnologías web" en la ruta "…/xampp/htdocs".

**En la entrega, ya damos configurados los parámetros para conectar con la base de datos con el usuario root y que funcione en el emulador**. Para lanzarla en un dispositivo, se debe cambiar la IP por la de nuestro Wi-Fi en las clases "DAOCuentaCuentas" y "ActivityAcercaDe", y cambiar el usuario root por otro nuevo con permisos sobre nuestra BBDD. Si falla, desactiva el firewall.

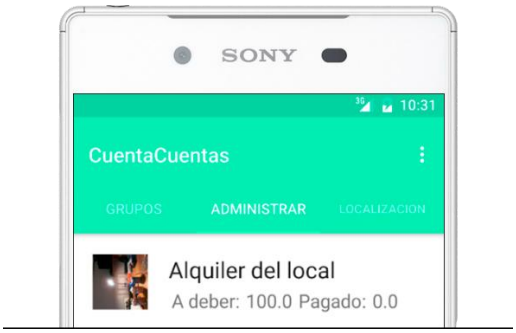
# MANUAL DE USUARIO

## ¿EN QUÉ CONSISTE LA APLICACIÓN?

*CuentaCuentas* es una aplicación que te permite llevar el control del dinero que debes a la gente y del que te deben a ti. Por ello, esta aplicación te facilitará las tareas de organizar eventos y recordar a las personas sus deudas.

## FUNCIONALIDADES PRINCIPALES

Al iniciar sesión en nuestra aplicación te encontrarás con tres pestañas en la parte superior: *Grupos, Administrar y Localización*. A continuación, explicaremos la función de cada una de ellas.




- ### PESTAÑA "GRUPOS"

  En esta pestaña encontrarás todos los grupos a los que perteneces. Pulsando sobre un elemento de la lista, accederás a la información del grupo pudiendo ver el dinero que debe cada uno y el dinero que ha sido pagado.

- ### PESTAÑA "ADMINISTRAR"

  Aquí verás los grupos que tú has creado y por tanto puedes administrar. Al pulsar en un elemento de la lista, accederás al grupo pudiendo añadir deudas y pagos sobre él.

- ### PESTAÑA "LOCALIZACIÓN"

  En ella podrás disfrutar de la funcionalidad que te ofrece el servicio de localización, pudiendo visualizar un mapa que te localiza y permite ver los puntos donde sucedieron los eventos de los grupos.

# CONSIDERACIONES TÉCNICAS FUNDAMENTALES

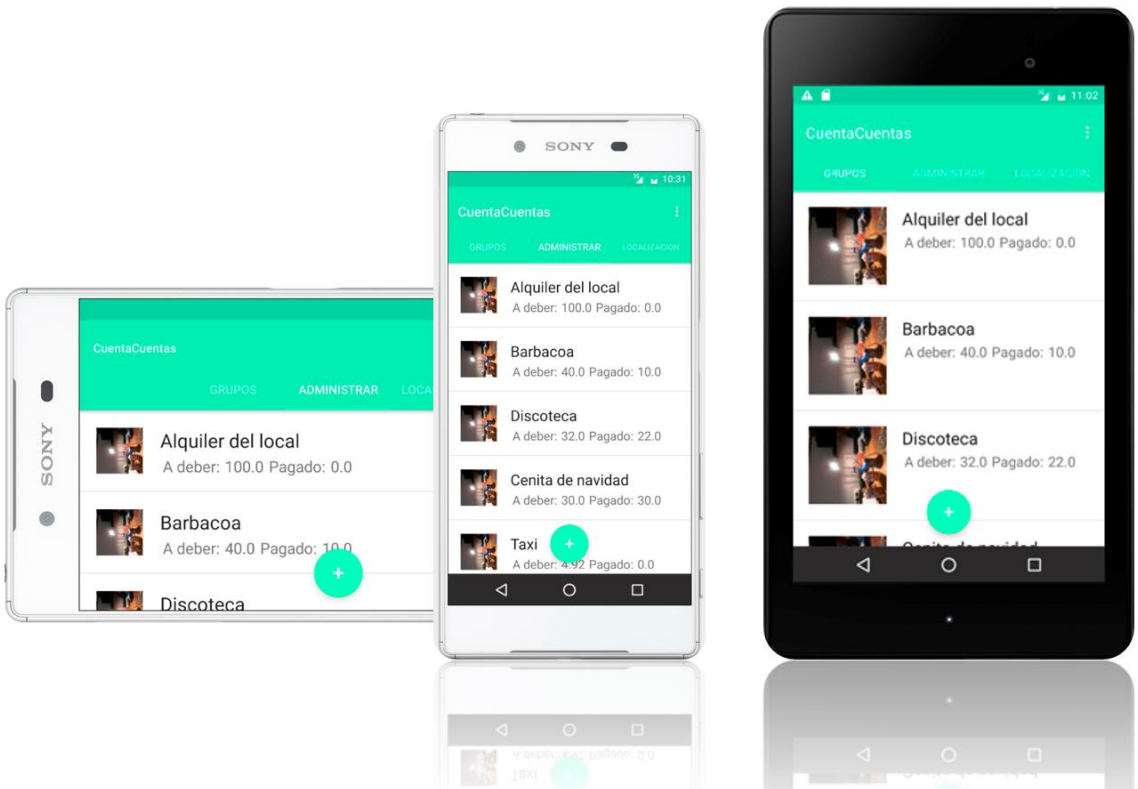
## CUMPLIMIENTO DE REQUISITOS MÍNIMOS DEL PROYECTO

Este apartado tiene como objetivo demostrar que hemos cumplido los requisitos mínimos que se especificaban en el enunciado del proyecto. Para ello, vamos a explicar para cada requisito cómo lo hemos implementado.

- Visualización apropiada en múltiples configuraciones de pantalla.
  - Distintas resoluciones y orientaciones

Hemos probado nuestra aplicación en todos los diferentes dispositivos que teníamos a mano y en diferentes emuladores que nos proporcionaba Android Studio para asegurarnos de cumplir este requisito.

En las siguientes imágenes podemos ver el aspecto que podría tomar nuestra aplicación en un *smartphone* y en una tableta:



- Almacenamiento permanente de información.
  - Debe estar disponible entre sesiones con la aplicación.

Para llevar a cabo este requisito, hemos utilizado una base de datos (local, que se adjunta en el archivo del proyecto). Para hacerla funcionar en Android Studio, hemos importado el JDBC en su última versión disponible.

Nuestra base de datos tiene el siguiente aspecto:


Está compuesta por 3 tablas que tienen las siguientes funciones:

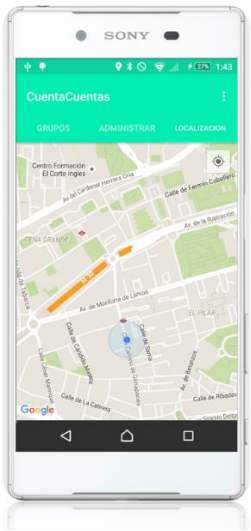
- **Usuarios**: almacena los usuarios que se registran en nuestra aplicación. Guarda sobre ellos un identificador único, su *nickname*, su *password*, su nombre, sus apellidos, su teléfono y su imagen.
- **Grupo**: almacena todos los grupos existentes en nuestra aplicación. Sobre ellos, se sabe su identificador único, su nombre de grupo, su imagen, la cantidad que deben los miembros, la cantidad que ha sido pagada y su latitud y longitud para la función de la geolocalización.
- **GrupoUsuario**: representa la relación de los grupos a los que pertenece cada usuario. Recordemos que un usuario puede pertenecer a 0, 1 o varios grupos y que un grupo puede estar formado por 0, 1 o varios usuarios.

De esta manera, nuestras tablas presentan las siguientes relaciones que relacionan la tabla "GrupoUsuario" con un id de un usuario y con un id de un grupo:

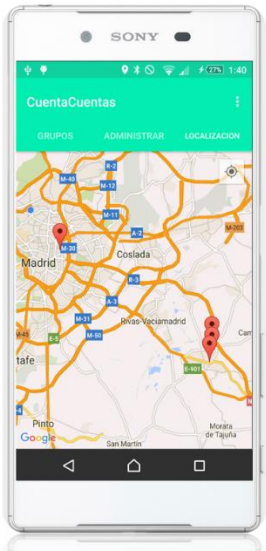
## Uso de al menos un servicio externo a la aplicación.

Como servicio externo, hemos decidido utilizar el servicio de la localización de Google, el cual nos permite añadir localizaciones en el mapa para recordar el punto en el que se ha producido el suceso de morosidad. Así, quedarán más pruebas de que otra persona te dinero ya que le puedes decir el sitio exacto en el que se lo prestaste.

Su funcionamiento es bastante sencillo, ya podemos ver en la ventana principal una pestaña de "Localización", que si pulsamos en ella nos llevará al mapa y nos situará en el sitio en el que nos encontremos:



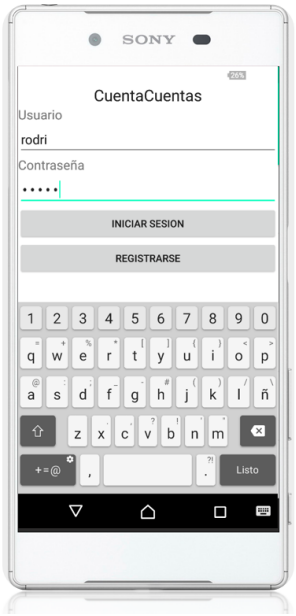

Y si queremos añadir un marcador en el mapa, simplemente debemos pulsar en la pantalla para colocarlo en el punto exacto que queramos. De esta forma, luego podríamos desplazarnos por todo el mapa y ver los marcadores:

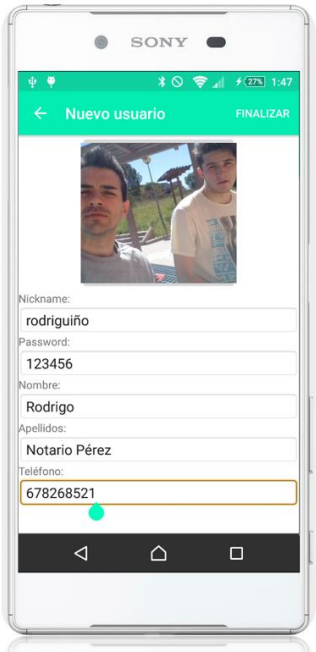
# Un mínimo de 2 formularios diferentes.

Como ya habréis visto, nosotros hemos utilizado numerosos formularios en nuestra aplicación. Hemos hecho más aparte de los dos obligatorios.

Por poner ejemplos, nada más abrir nuestra aplicación, el usuario ve un formulario para iniciar sesión:




Y, por ejemplo, tenemos otros formularios más completos para crear un nuevo usuario o para crear un nuevo grupo:

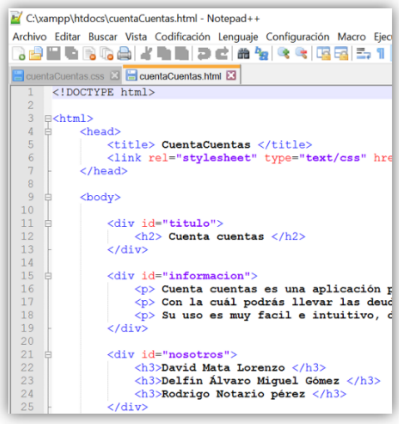


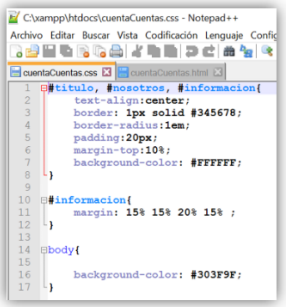

- Una parte desarrollada con tecnologías web, con interacción con la parte nativa.

Para este requisito, os preguntamos en clase qué parte sería mejor que hiciéramos con tecnologías web para que no desentonara mucho con el desarrollo nativo de nuestra aplicación y nos distéis una buena idea: hacer un “Acerca de” explicando un poco de que iba nuestra aplicación y con nos los nombres de los miembros del grupo.

Y así lo hicimos, si pinchamos en el botón superior derecho y le damos a “Acerca de” veremos la siguiente pantalla:


La cual está formada por un documento *html* y otro documento *css*, algo simples debido a que no controlábamos mucho sobre desarrollo web, pero que hemos podido desarrollar gracias a lo explicado en clase.



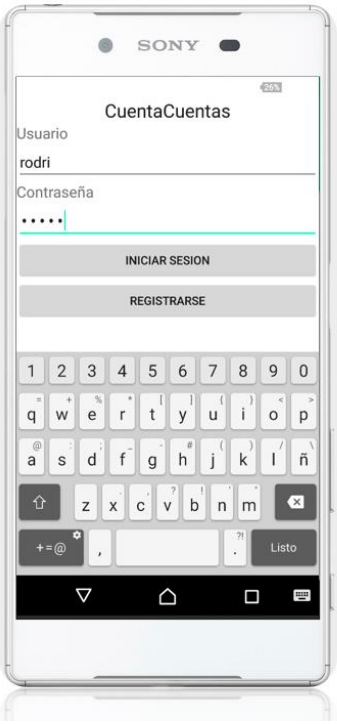

*<u>NOTA:</u> Esta parte la encontrarás dentro de la carpeta “Tecnologías web”. Recuerda copiar la carpeta de dentro “CuentaCuentas” en “…/xampp/htcdocs” para su correcta visualización.*

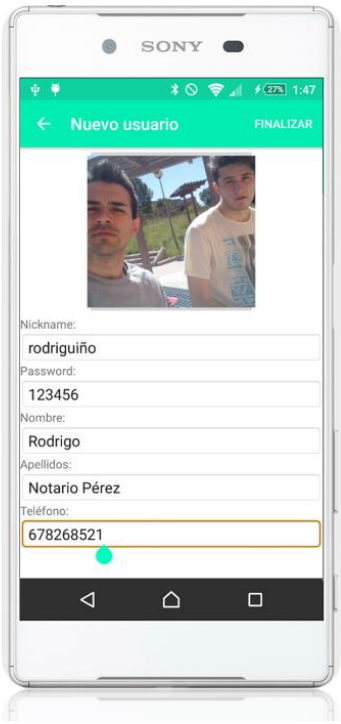
## SUBSISTEMA DE USUARIOS

Uno de los subsistemas que posee nuestra aplicación es el relacionado con los usuarios. Como veremos posteriormente en uno de los diagramas de casos de uso, aquí hemos agrupado las acciones del *login*, el *logout* y el registro de usuarios.

Al entrar en nuestra aplicación te encontrarás con la ventana de *Login*. En ella tendrás dos opciones:

- **Si ya es estás registrado**, deberás rellenar el formulario con tu nombre de usuario y contraseña, y posteriormente, darle al botón de *Iniciar sesión*. Al ejecutar esta acción, se comprobará mediante una consulta SQL que el usuario efectivamente está registrado en nuestra base de datos y que su contraseña coincide. En caso afirmativo, se le mostrará la ventana principal.
- **Si eres un nuevo usuario,** deberás darle al botón de *Registrarse*. A continuación, aparecerá una nueva ventana con un formulario para que el usuario introduzca sus datos. Cuando estos datos se hayan rellenado, deberás darle al botón superior derecho de "Finalizar". En este momento se comprobará que el *nickname* no esté ya registrado en la base de datos. Si ya se encuentra en uso, se le mostrará un mensaje al usuario informándole de ello. En caso contrario, el registro finalizará mostrando la ventana principal al usuario.
- **Para cerrar la sesión (*logout*)** simplemente deberás pulsar en la pantalla principal el botón superior derecho y ya verás esta opción.

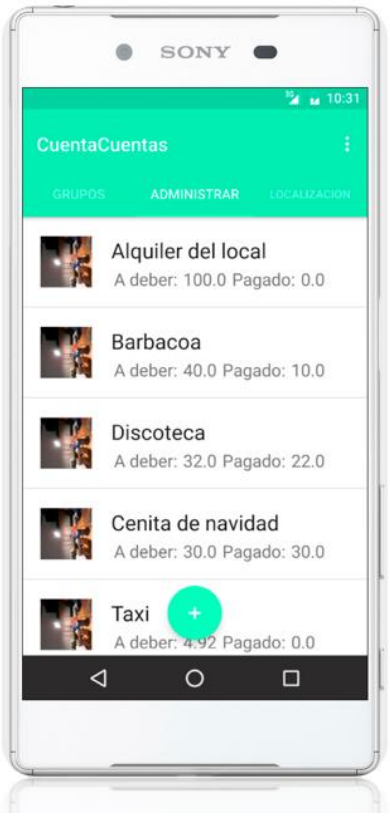



*NOTA: Al iniciar sesión o cuando te estás registrando y pulses el botón "Finalizar",* ***debes esperar unos segundos*** *(ya que hemos utilizado el JDBC importado en Android Studio). Si pulsases varias veces seguidas "Finalizar", lo que estarás haciendo será enviar varias peticiones seguidas: la primera te registrará y para el resto dirá que ya está el nick registrado y deberás pulsar "Atrás".*

## SUBSISTEMA DE GRUPOS

En este apartado veremos el subsistema de grupos. Este subsistema es el más complejo de nuestra aplicación y engloba todo lo relacionado con los "grupos de morosos" a los que pertenecen los usuarios.




En la pantalla principal veremos que se carga automáticamente una lista de los grupos a los que pertenecemos. Tenemos dos pestañas relacionadas con los grupos:

- **Grupos**: aquí veremos todos los grupos a los que pertenecemos.
- **Administrar**: aquí veremos todos los grupos que hemos creado nosotros y que, por tanto, tenemos la función de administrar.

En ambas pestañas, podemos pulsar sobre un grupo para acceder a él. Si es un grupo que administramos, podremos realizar acciones sobre él tales como:

- Añadir usuarios al grupo
- Agregar una deuda general al grupo
- Agregar un pago general al grupo
- Agregar una deuda individual a un usuario
- Agregar un pago individual de un usuario
- Reiniciar las cuentas
- Borrar completamente el grupo

Todas estas acciones tendrán acceso a la base de datos como veremos posteriormente en los diagramas de la sección de arquitectura de la aplicación.

# DISEÑO CON MODELOS Y COMENTARIOS

En este apartado trataremos de mostrar la arquitectura de la aplicación. Empezaremos explicando los diferentes casos de uso de nuestra aplicación. Posteriormente, analizaremos los aspectos de la arquitectura desde un punto de vista global. Y ya finalmente, trataremos de mostrar los diagramas más generales para entender el diseño en profundidad de cada una de las capas de la aplicación desde un modo de vista general.

## INTRODUCCIÓN: PATRONES DE DISEÑO

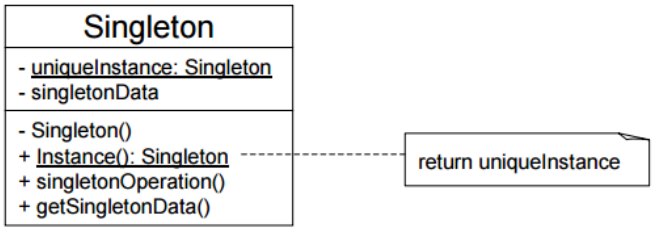
En cuanto a patrones de diseño, hemos decidido aplicar el patrón *Singleton*, el cual nos permite crear una única instancia de clase accesible a los clientes desde un punto de acceso conocido.

Las ventajas que nos brinda este patrón son las siguientes:

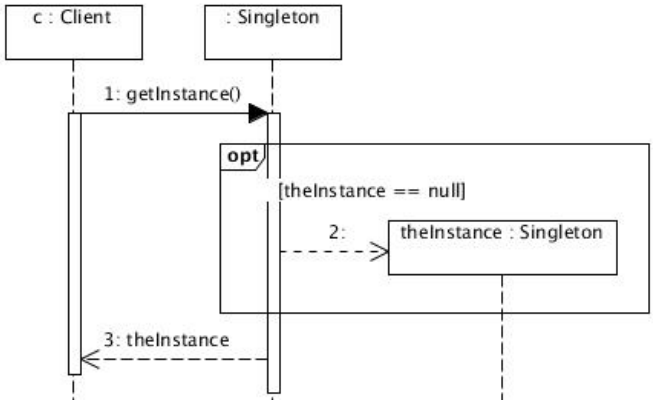
- Acceso controlado a la única instancia.
- Espacio de nombres reducido.
- Permite refinamiento de operaciones y su representación.
- Permite un número variable de instancias.
- Más flexibles que las operaciones de clase estáticas.

Además, como vimos el año pasado en la asignatura de Ingeniería del Software, este patrón no presentaba inconvenientes.

Con estas características, este patrón presenta la siguiente estructura estática:




Y sus diagramas de secuencias (comportamiento dinámico) tendrán la siguiente forma:

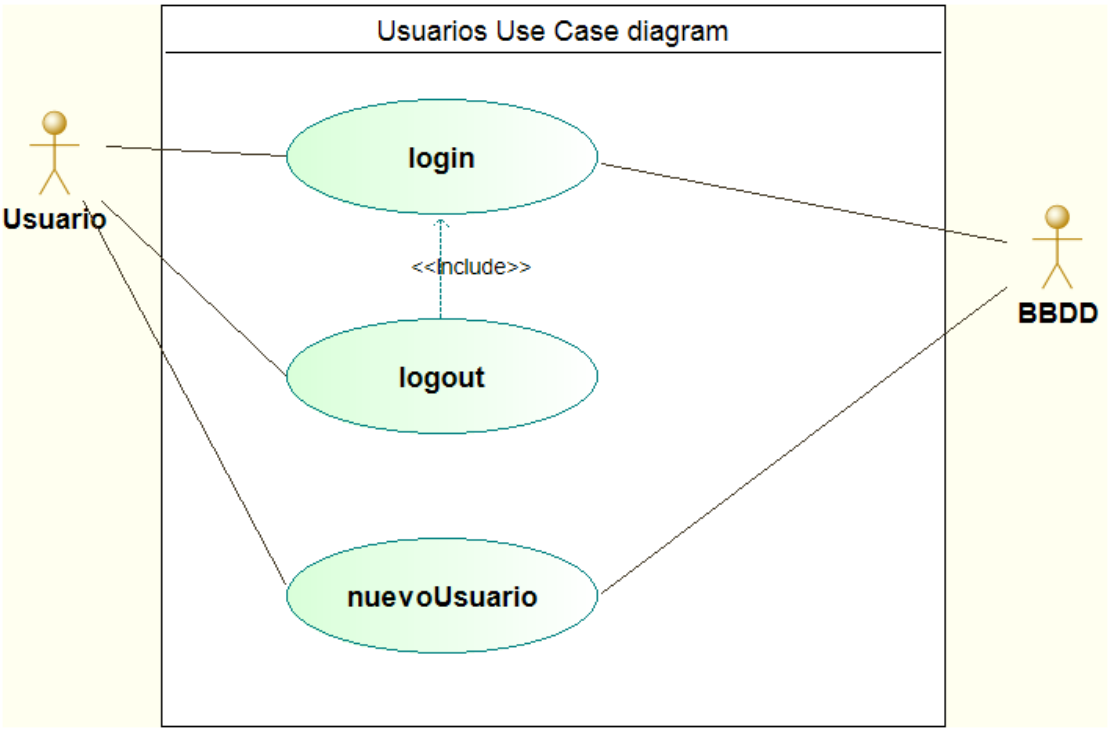


## Diagramas de casos de uso

Los casos de usos que vamos a exponer a continuación hacen posible el desempeño de las diferentes funcionalidades de *CuentaCuentas*, de forma de que los usuarios de la aplicación puedan llevar el control de sus pagos.

## Diagrama de casos de uso de usuarios

Este primer diagrama representa las tareas relacionadas con la gestión de usuarios:



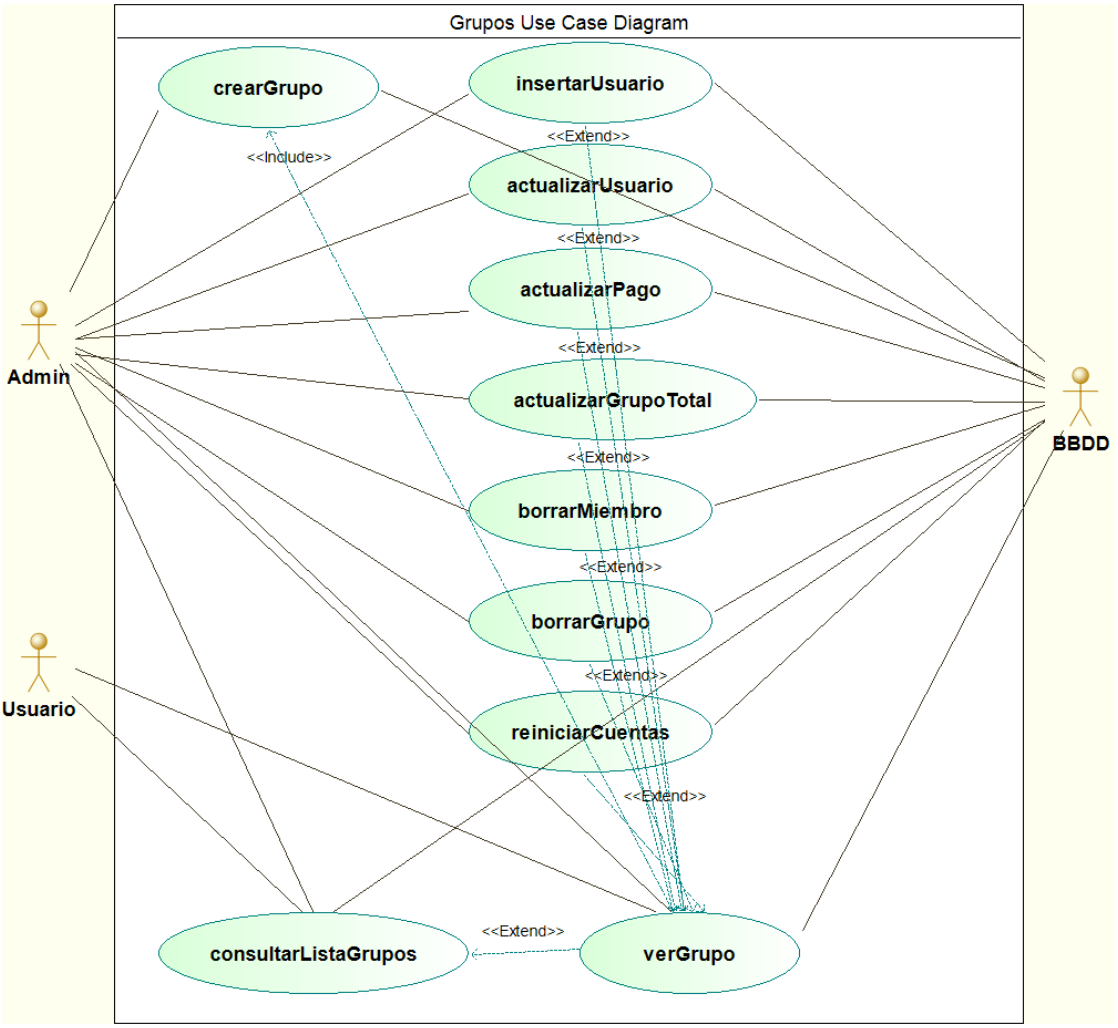

A la izquierda un actor que será el usuario que utilice nuestra aplicación de *CuentaCuentas*. Y dentro del cuadro, podemos ver las acciones que puede realizar:

- **login**: cuando el usuario inicia sesión, se consulta si coindice el *nickname* y la contraseña introducidos están contenidos en la lista de usuarios registrados.
- **logout**: cierra la sesión del usuario actual. Para ello, es necesario que previamente haya hecho "login".
- **nuevoUsuario:** cuando el usuario se registra, se comprueba en la base de datos que no existe un usuario con el mismo *nickname* y en caso afirmativo, se procede a registrar el usuario en la base de datos.

A la derecha tenemos la representación de la base de datos, la cual es accedida por la función de "login" y por la de "nuevoUsuario".

## DIAGRAMA DE CASOS DE USO DE GRUPOS

Este otro diagrama de casos de uso trata de representar las acciones que podemos realizar sobre un grupo y ya presenta una estructura más compleja:




A la izquierda, tenemos dos actores: el administrador y el usuario. El administrador es la persona que creadora de un grupo y que, por tanto, realiza tareas de administración sobre dicho grupo. Por otro lado, los usuarios son el resto de personas pertenecientes al grupo.

Las tareas que pueden realizar los usuarios son consultar la lista de grupos y ver un grupo concreto de la lista.

Las tareas que puede realizar el administrador, aparte de las mencionadas anteriormente para los usuarios, son: insertar un usuario en el grupo, actualizar un pago, actualizar la cantidad de un usuario, actualizar la cantidad total del grupo, borrar un miembro del grupo, borrar el grupo entero y reiniciar las cuentas del grupo.

Todas ellas necesitan hacer uso de la base de datos, por lo que todas ellas se relacionan con la base de datos representada en la parte derecha.

## DIAGRAMA DE CLASES

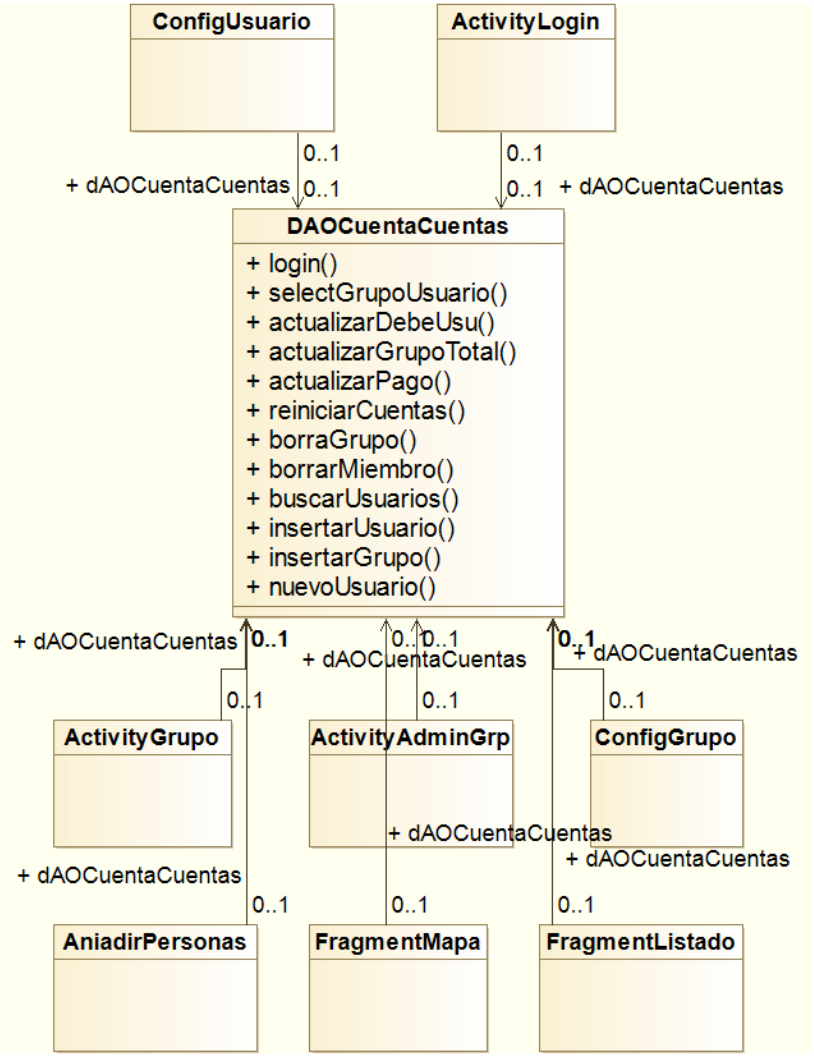
Los diagramas de clases representan la estructura estática de la aplicación.

El diagrama de clases que vamos a ver a continuación tiene como objetivo ver un esquema general de las clases que acceden al DAO.

En este diagrama, hemos combinado el subsistema “Grupos” con el subsistema “Usuarios” debido a que este último subsistema tenía apenas un par de clases que accedían al DAO, y así podemos ver una visión general.

Su estructura es muy simple debido a que como hemos dicho anteriormente, en nuestro caso, hemos utilizado el patrón *Singleton* en donde se obtiene y se devuelve una única instancia.

Los métodos incluidos en el DAO serán utilizados por las diferentes clases del diagrama para poder llevar a cabo sus objetivos.

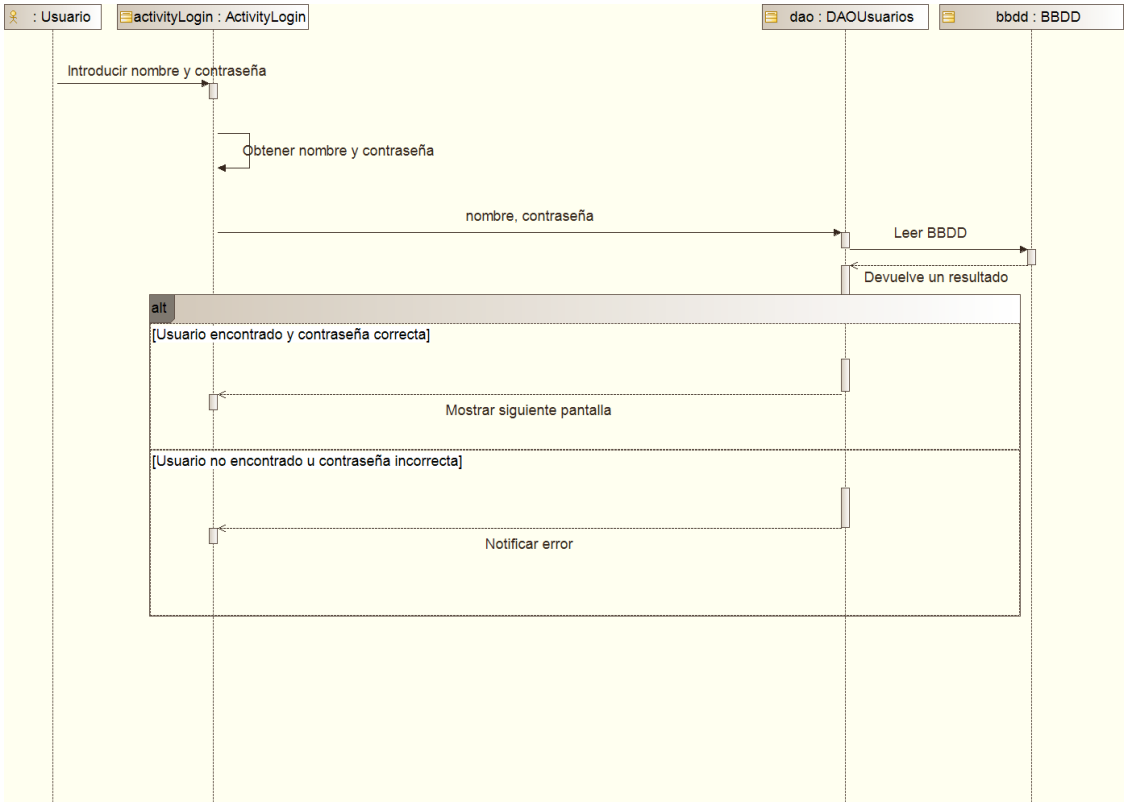


## DIAGRAMAS DE SECUENCIA

A continuación, vamos a exponer los diagramas de secuencia realizados para entender nuestra aplicación. Lo diagramas de secuencia, tienen la función de mostrar la interacción de los objetos en nuestra aplicación a lo largo del tiempo y serán modelados para cada caso de uso.

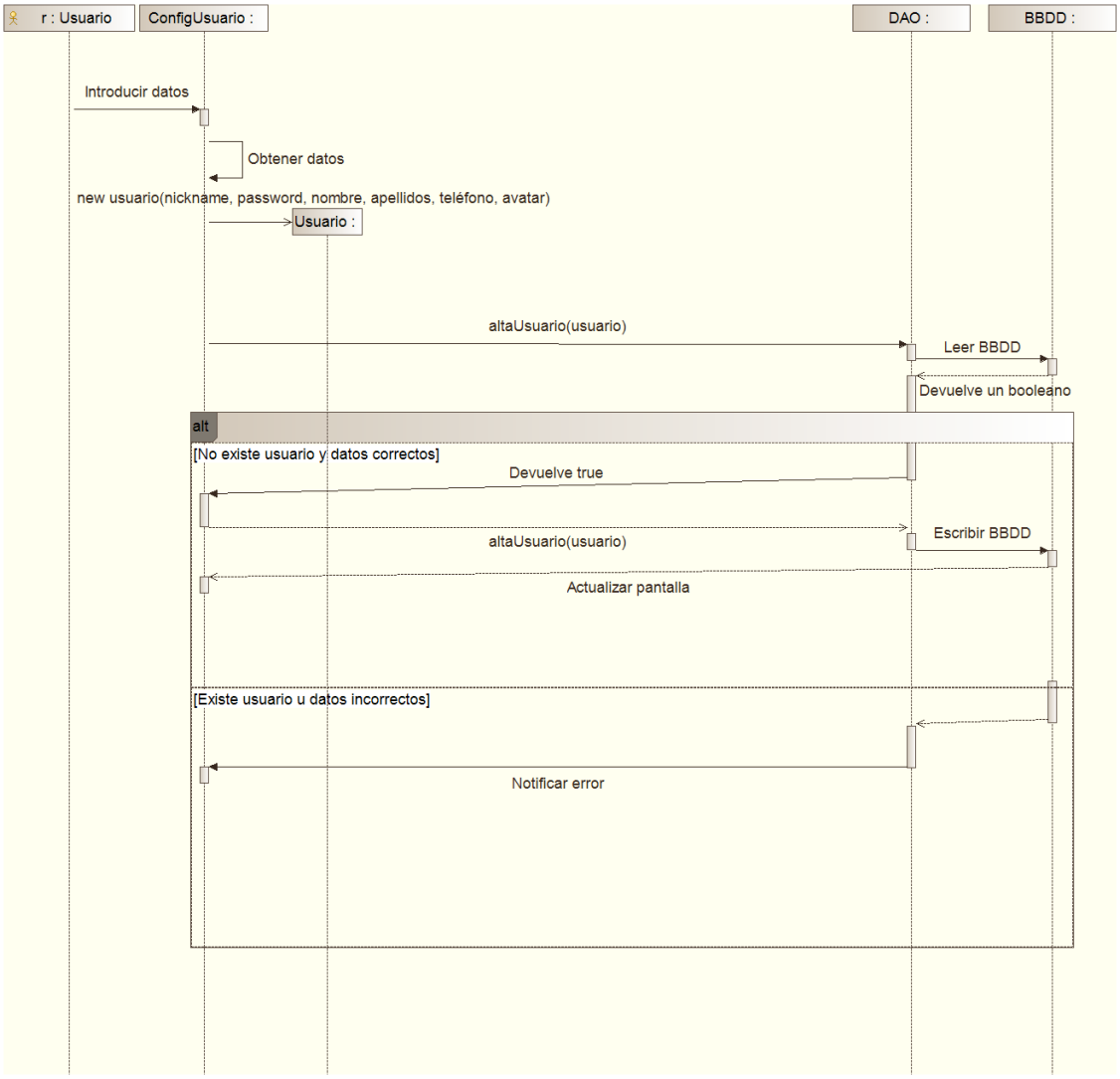
## DIAGRAMA DE SECUENCIA DEL LOGIN

El primer diagrama de secuencia trata de representar la secuencia que se produce cuando un usuario inicia sesión, y los diferentes desenlaces que se producen, en el caso de que el usuario haya introducido los datos correctamente o no.



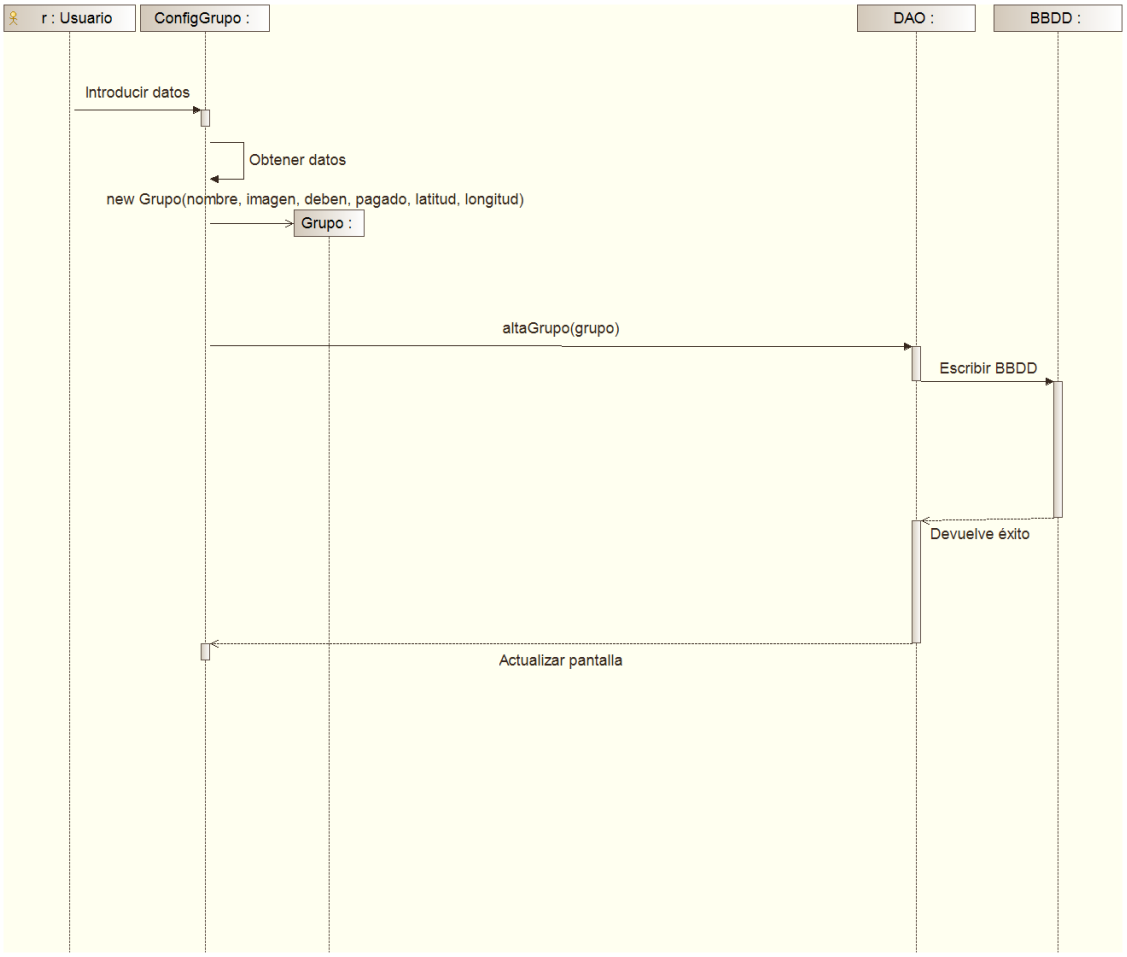
## DIAGRAMA DE SECUENCIA DEL REGISTRO DE USUARIOS

Este diagrama de secuencia, representa la transición que se produce cuando un usuario quiere registrarse en nuestra aplicación. En primer lugar, el usuario introduce los datos, los cuales son recogidos y posteriormente se crea un objeto de tipo *Usuario* que será pasado para poder darlo de alta en nuestra base de datos. También posee dos posibles desenlaces dependiendo de la comprobación que se hace para ver si el *nickname* ya está registrado o no.



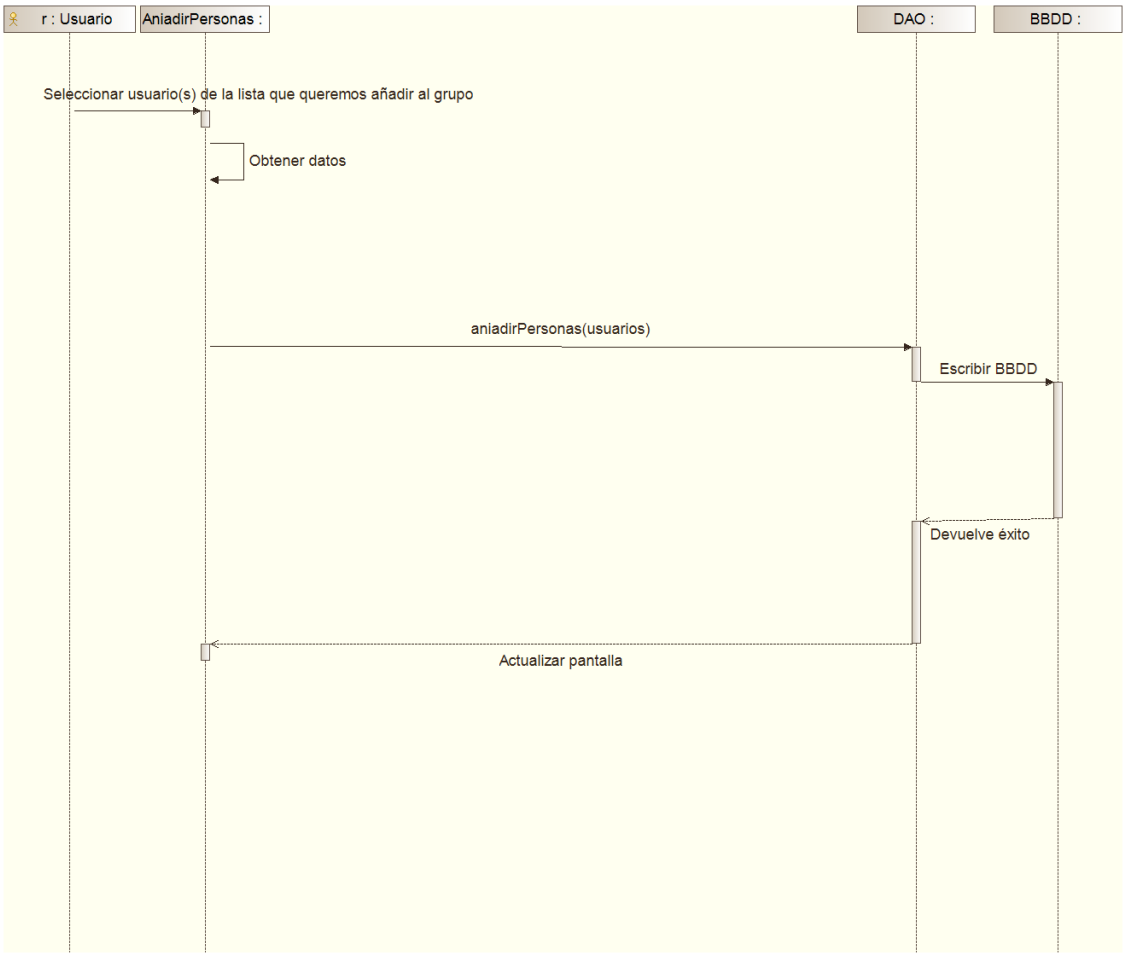
## DIAGRAMA DE SECUENCIA DE LA CREACIÓN DE UN GRUPO

Este otro diagrama hace visible la secuencia que se produce al añadir un nuevo grupo. Es muy parecido al anterior que creaba un nuevo usuario, pero en este caso no hay dos posibles desenlaces, ya que la creación de un grupo siempre será exitosa.



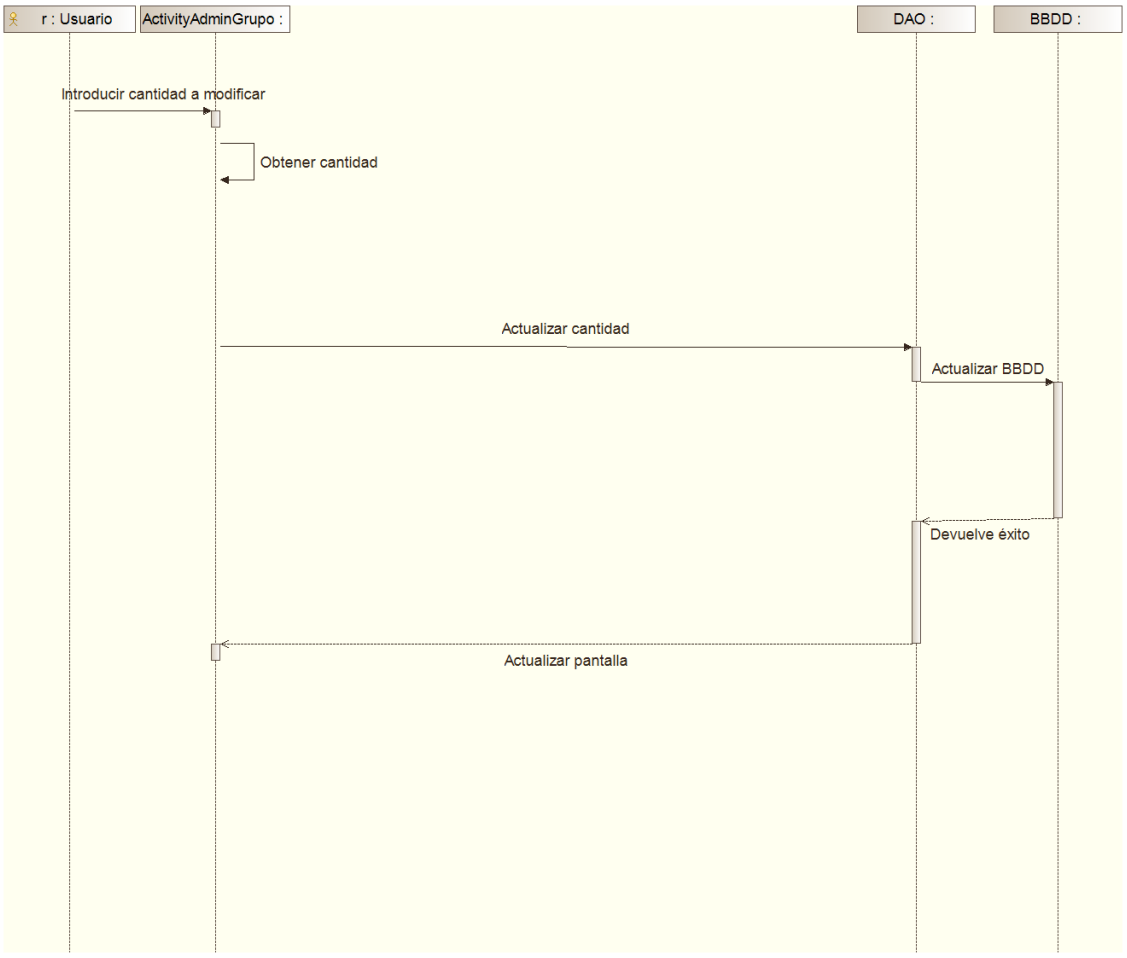
## DIAGRAMA DE SECUENCIA DE AÑADIR USUARIO(S) A UN GRUPO

Este diagrama de secuencia, representa el añadir uno o varios usuarios a un grupo. Recordemos que los usuarios que añadíamos se seleccionaban de una lista, por lo que en este caso no habría que realizar comprobaciones acerca de si ese usuario existe o no, sino que siempre será exitosa su integración en el grupo.



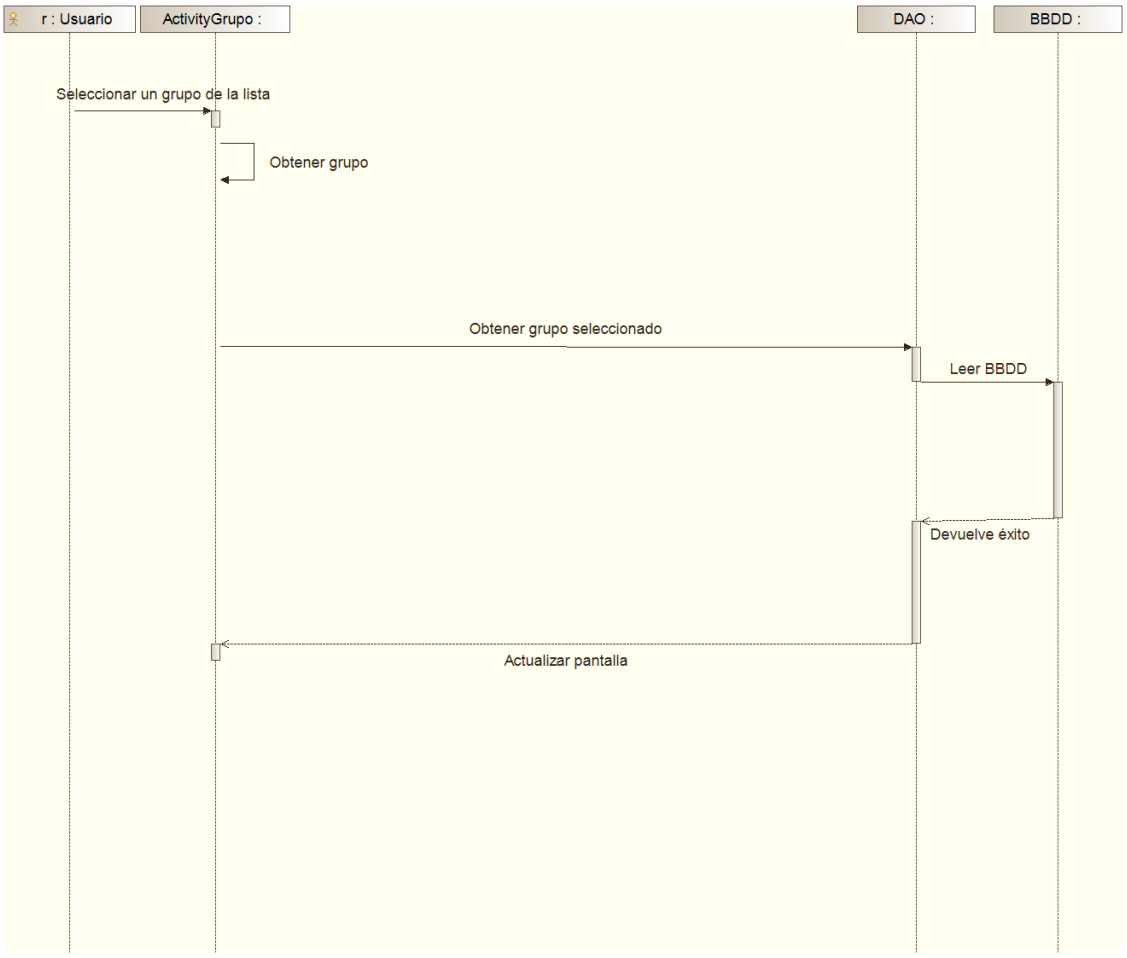
## DIAGRAMA DE SECUENCIA DE ACTUALIZAR UNA CANTIDAD EN UN GRUPO

En este otro diagrama, podemos ver la secuencia que se produce cuando actualizamos una cantidad en un grupo. Esta cantidad puede ser el dinero que se debe total, el dinero que ha pagado un miembro del grupo, el dinero que debe un miembro… Pero todo ello sigue el mismo modelo que vemos a continuación.



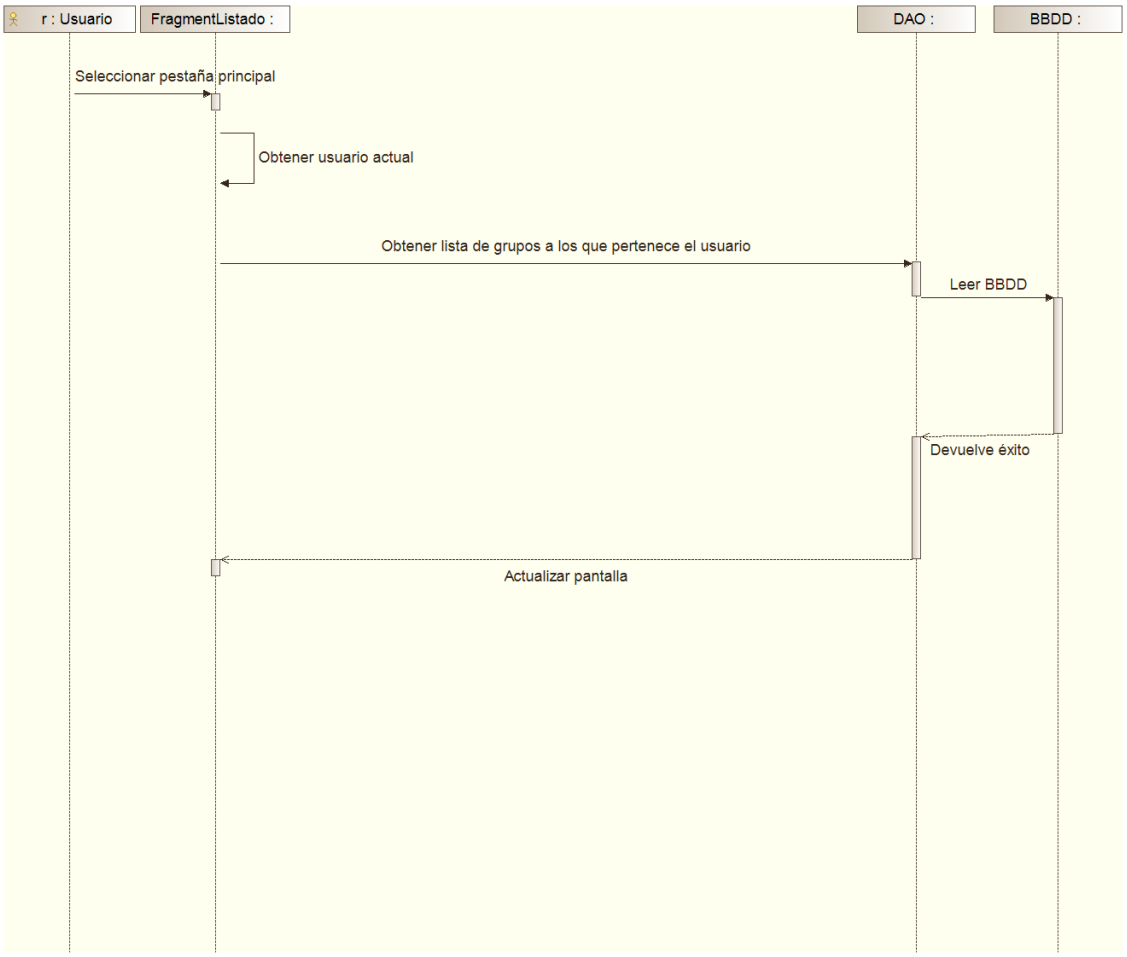
## DIAGRAMA DE SECUENCIA DE VISUALIZAR UN GRUPO

Aquí podemos ver la secuencia qué se produce cuando un usuario visualiza un grupo. Hemos de recordar que los grupos se seleccionan de una lista de grupos a los que pertenece cada usuario que veremos en el diagrama siguiente a este cómo se obtienen. Pero al seleccionarse de una lista no hay ningún error que controlar.



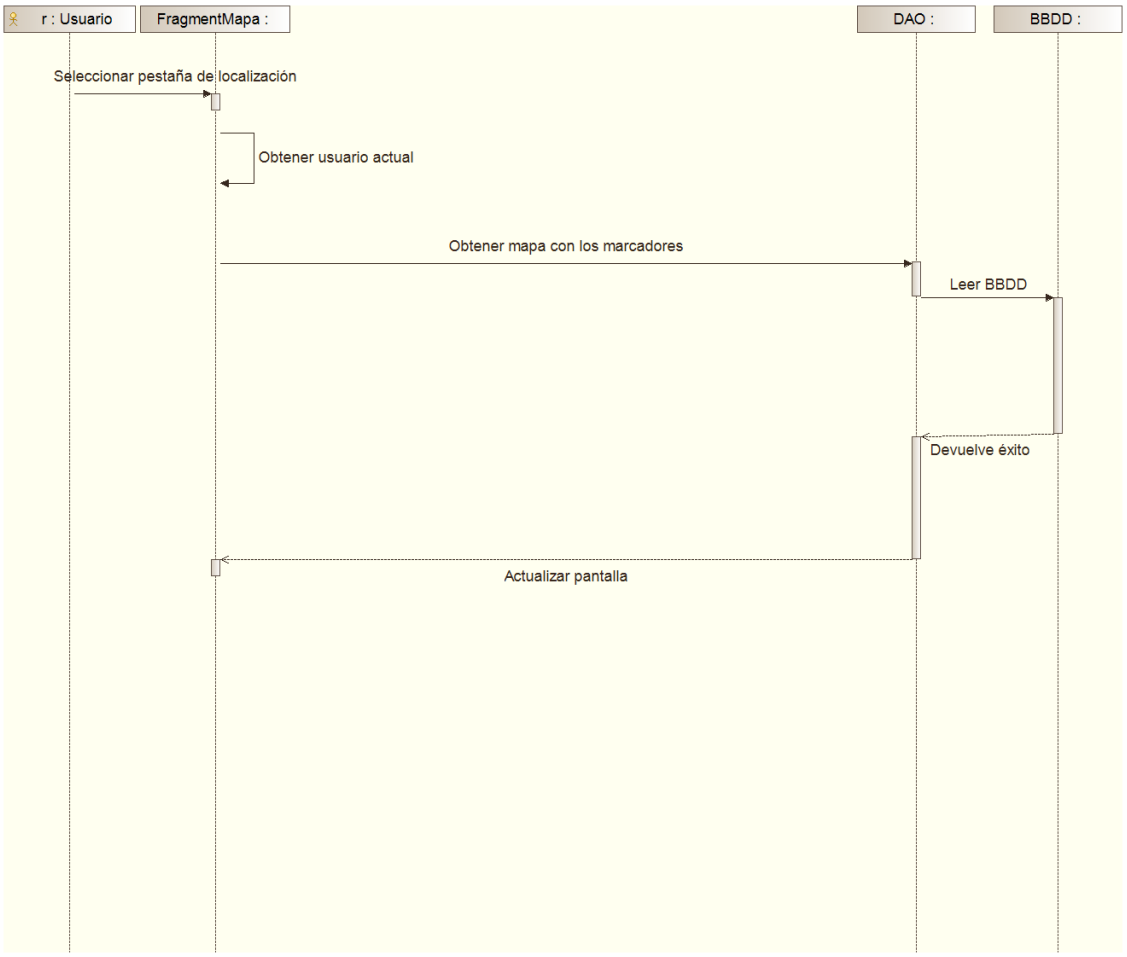
## DIAGRAMA DE SECUENCIA DE OBTENER LA LISTA DE GRUPOS

Este penúltimo diagrama tiene como objetivo hacernos ver cómo se obtiene la lista de grupos a los que pertenece un usuario. Esta lista, se muestra en la pantalla principal y por tanto debe cargarse automáticamente cuando el usuario acceda a ella. Para ello, debemos de pasarle el usuario actual y leer de la base de datos todos los grupos a los que pertenece para que sean devueltos.



## DIAGRAMA DE SECUENCIA DE OBTENER MAPA CON MARCADORES

Y ya para terminar, tenemos este otro diagrama de secuencias muy sencillo que nos permite ver la secuencia que se produce cuando tratamos de obtener un mapa (con marcadores representados en él). Aquí simplemente al usuario pulsar en la pestaña de localización correspondiente, se accedería a la base de datos para devolver el resultado que sería mostrado por pantalla:



# APORTACIONES DE LOS INTEGRANTES DEL GRUPO, METODOLOGÍA Y CONCLUSIONES

Por último, para terminar la memoria vamos a hablar acerca del trabajo general que nos ha supuesto este proyecto y sobre las principales decisiones de diseño que hemos tomado.

Al comienzo del proyecto, lo primero que hicimos fue pensar sobre qué podíamos hacer nuestra aplicación. Uno de los miembros del grupo, necesitaba una aplicación que le permitiera controlar el dinero que debía cada persona en el alquiler de locales. Así que, ante este problema real, decimos que haríamos una aplicación sobre ello, pero que además permitiera controlar todo tipo de deudas: desde el dinero que le prestas a alguien para pagar una cerveza hasta la gestión del dinero de un viaje.

Decidimos utilizar Android Studio porque sentíamos la necesidad de conocer este entorno de desarrollo. Ya que ya estábamos aprendiendo tecnologías web en la asignatura de "Aplicaciones Web" del segundo cuatrimestre, así que decidimos por aprender más y tirar por lo nuevo. Además, esta aplicación en principio la utilizarían personas con teléfonos Android.

Posteriormente, realizamos los diagramas de diseño a papel. Con ello, conseguimos diseñar cómo queríamos que fuese la arquitectura de nuestra aplicación.

Durante la fase de realización del proyecto, nos empezó a parecer complicada la idea, ya que mientras algunos compañeros estaban haciendo aplicaciones con dos únicos formularios, nosotros estábamos haciendo una aplicación más compleja con numerosos formularios que nos estaba dando mucho trabajo.

Para organizarnos mejor, utilizamos Google Drive que nos brinda la UCM con nuestras cuentas de correo, como herramienta para intercambiar nuestros archivos y llevar a cabo un control de versiones. Ya que a cada archivo me subíamos le poníamos un número de versión (v1, v2, v3, v3.1 si no había muchos cambios con la anterior, etc.).

Además, cada semana veníamos al laboratorio para contarnos lo que habíamos avanzado cada uno y proponernos nuevas tareas. Así, una persona contaba lo que había hecho y cómo lo había hecho para ayudar al resto por si no entendíamos algo y que nos sirviera de cara a avanzar en nuestra tarea y, además, que nos sirviese de cara al examen teórico.

Finalmente, en la última fase combinamos tres tareas principales: la realización de pruebas de caja negra, la corrección de errores y completar la memoria de la aplicación. De esta manera, los turnábamos para que mientras una persona corregía errores, otra iba probando y comentándole a la anterior todo lo que veía, y otro miembro se encargaba de completar la memoria (ya que, por ejemplo, los diagramas los teníamos que pasar de papel a ordenador, teníamos que realizar capturas de pantalla, etc.).

Así que podemos decir que verdaderamente hemos trabajado todos los miembros del grupo adquiriendo conocimientos sobre esta asignatura.

# BIBLIOGRAFÍA

## LIBROS


- **Desarrollo de aplicaciones para Android (Edición 2016)** del autor Joan Ribas Lequerica y de la editorial Anaya.

- **El gran libro de Android (5ª Edición)** del autor Jesús Tomás y de la editorial Marcombo.


## PÁGINAS WEBS


- **Google developers**: https://developers.google.com/

- **Blog "Curso Android Studio"**: http://cursoandroidstudio.blogspot.com.es/2015/01/base-de-datos-remotas-login.html

- **Udacity**: https://www.udacity.com/course/viewer#!/c-ud853/l-1395568821/m-1643858568


## CANALES DE YOUTUBE


- **Curso para Android de Jesús Conde:** https://www.youtube.com/playlist?list=PL7EA29F3B739286CA

- **Curso Android de *pildorasinformaticas***: https://www.youtube.com/playlist?list=PLU8oAlHdN5Bkn-KS1sRFlSEnXXcAtAJ9P


## DIAPOSITIVAS


- **PDF's de la asignatura**, tanto de la parte de Android para el desarrollo de la aplicación nativa como de la parte de tecnologías web para hacer el "Acerca de".