Memoria Appagar

Pablo Arranz Ropero, Juan Alberto Camino Sáez

Manual de instalación

A continuación, se detallan los requisitos mínimos con los que debe contar el dispositivo móvil para que la aplicación funcione correctamente, y las instrucciones necesarias para la instalación de la misma

Requisitos mínimos

Versión de Android mínima

Está desarrollado para la API 23, correspondiente a la versión 6.0 Marshmallow.

Permisos requeridos

Nuestra aplicación no requiere la concesión de ningún tipo de permisos.

Instalación de la APK

Para instalar la aplicación en nuestro dispositivo Android (que deberá cumplir los requisitos mínimos), se deberá permitir la instalación de aplicaciones de origen desconocido. Podremos permitir la instalación de aplicaciones de origen desconocido en los ajustes del dispositivo, más concretamente en la sección “Seguridad”. Dentro de esta sección activaremos la casilla “Orígenes desconocidos”. Después de hacer esto podremos finalmente ejecutar el archivo apk que resultará en la instalación de la aplicación en el teléfono móvil.

Manual del usuario

A continuación, se explica en qué consiste y cómo funciona la aplicación.

Resumen de la aplicación

Appagar es una aplicación que permite almacenar distintas cuentas conjuntas (como puede ser un viaje en común, una cena de negocios) y llevar a cabo una gestión fácil y sencilla de todas ellas, pudiendo registrar los gastos que realiza cada persona y cada vez que una persona pague a otra, teniendo en todo momento una visión gráfica de la cuenta, así como ver cuánto tiene que pagar cada uno, el total gastado y el importe por persona.

Funcionalidades

VENTANA INICIAL

En esta ventana inicial (que podemos observar en la imagen 1) podemos realizar las siguientes operaciones:

- Crear una nueva cuenta, introduciendo su nombre y sus participantes.
- Cargar una cuenta existente, seleccionando su nombre, con todos los datos disponibles en la misma.
- Finalizar una cuenta, cuando ya no va a haber más gastos en la misma o se considera obsoleta.

Además, hay un menú en la esquina superior derecha, que te permitirá hacer las siguientes operaciones:

- Acceder a la ayuda de la aplicación, que te explica con detalle cómo moverte por la aplicación
- Acceder a “Acerca de”, para conocer más información sobre la aplicación y sus creadores.

La aplicación cuenta con una base de datos que guarda las cuentas, por lo que se pueden recuperar, aunque hayamos salido de la aplicación.

DENTRO DE LA CUENTA

Una vez creada o cargada una cuenta, podrás realizar lo siguiente (Fondo de la imagen 3):

- Registrar un nuevo gasto, introduciendo el participante implicado y la cantidad que ha gastado.
- Registrar un nuevo pago, introduciendo los participantes involucrados en el mismo, con la cantidad pagada.
- Ver el estado de la cuenta, donde se podrá ver información del total de dinero gastado en la cuenta, el importe por persona, cuánto ha pagado cada una, y el dinero que debe recibir o pagar para que se iguale el dinero gastado por todos los participantes

Además, en una pestaña de menú arriba a la derecha de la pantalla, podrás realizar las siguientes operaciones:

- Añadir participantes nuevos a la cuenta, por si no lo habías hecho con anterioridad.
- Avisar a un participante de su estado en la cuenta, a través del servicio de mensajería que tú elijas.

El estado de las cuentas se mantiene siempre actualizado en la base de datos con la que cuenta la aplicación, modificando la tabla de la cuenta correspondiente.

Consideraciones técnicas fundamentales

Aspectos generales

Para la correcta visualización en diferentes pantallas, y que se pueda ver la aplicación tanto si la pantalla está en horizontal como en vertical, se han usado distintos archivos de layout en cada activity, uno para la orientación vertical y otro para la orientación horizontal.

Para el almacenamiento permanente de los datos, se ha usado una base de datos SQLite en local, servicio proporcionado por Android.

Módulo Cuentas

Este módulo engloba todo lo relacionado con la creación, carga y eliminado de las cuentas existentes en la aplicación. Se compone de 6 activities, cada una de ellas realizando una función específica.

Ventana principal

La pestaña principal se compone de una imagen, tres botones que permiten realizar distintas funcionalidades, y un menú.

Esta actividad se llama en el proyecto ActivityInicio.java

Para mostrar la imagen se ha usado un ImageView, y la imagen está guardada en la carpeta de recursos de la aplicación. Para los botones se ha usado Button y para el menú se ha definido un fichero xml con el diseño del mismo.

El paso entre actividades se realiza a través del objeto Intent, pasando datos a las distintas actividades cuando sea necesario

IMAGEN 1

Cargar cuenta y finalizar cuenta

Estas activities son bastante similares, diferenciándose tan solo en su funcionalidad.

Tiene un texto, en el que se ha usado el campo TextView, y después una pestaña desplegable con las cuentas existentes. En este caso, hemos decidido usar un Spinner, para su manejo sea más cómodo y visual. Por último se ha usado un botón, para aceptar la creación de la cuenta.

El Spinner recoge las cuentas de la base de datos, para mostrarlas al usuario.

En cuanto a las activities correspondientes a la ayuda, el acerca de, y nueva cuenta, se usan simplemente TextView para mostrar texto, EditText para escribir en el mismo, además de los botones, empleando como en el caso anterior el objeto Button.

IMAGEN 2

Módulo Gestión

Este módulo engloba todo lo relacionado con la gestión de una cuenta en concreto: registro de gastos y pagos y mostrar la tabla de información de la cuenta Se compone de 5 activities, cada una de ellas realizando una función específica.

En este caso, la ventana principal de la gestión de una cuenta usa los mismos tipos de datos y estrategias para la creación de la misma que la ventana principal en el anterior módulo

Asimismo, las activities de nuevo gasto y nuevo pago usan un Spinner, en este caso para seleccionar los participantes involucrados en el mismo, y EditText para insertar el importe. Para mostrar texto se usa de nuevo TextView.

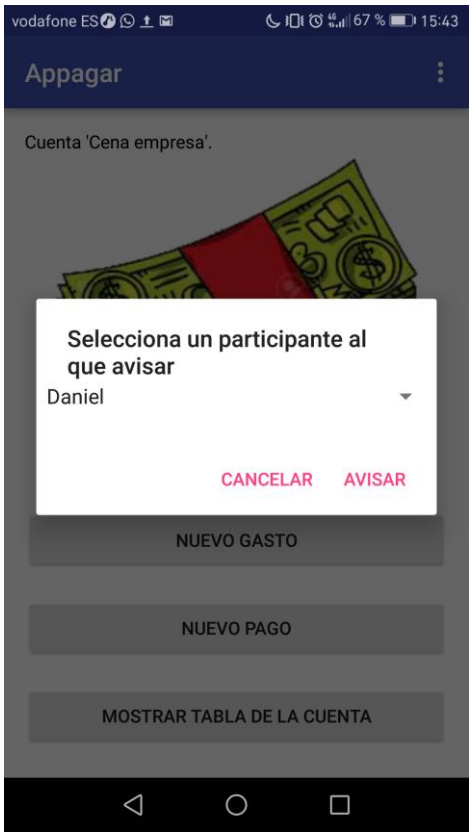

IMAGEN 3

Añadir participante

Para realizar esta funcionalidad, que forma parte de la ventana principal, se ha usado un AlertDialog, en el que se ha modificado para poder introducir un participante (con un Spinner), y de esta manera poder enviar un mensaje a un participante con su estado actual en la cuenta.

En esta activity además se usa un servicio externo a la aplicación (una aplicación de mensajería) Para ello, se ha usado un Intent, poniendo en el campo signal el atributo SEND. De esta manera, el sistema android te da a elegir una aplicación disponible en el dispositivo para enviar el mensaje personalizado de la aplicación.

Tabla de la cuenta

Esta activity muestra el estado de la cuenta. Para realizar la tabla se ha usado tecnologías web (HTML, CSS y JavaScript), que se ha insertado utilizando un WebView. Este documento web está almacenado en la carpeta de recursos de la aplicación.

Para realizar la tabla se ha usado HTML y CSS, y para obtener los datos de la misma, el sistema Android accede a los datos de la base de datos, mandando los mismos a la parte web donde mediante JavaScript se calculan todos los datos necesarios para rellenar la tabla correctamente.

IMAGEN 4

Visualización apropiada en diferentes configuraciones de pantalla

En nuestra aplicación contamos con un diseño responsive, es decir, que se adapta a diferentes configuraciones de pantalla. Como se indicaba al comienzo de la sección, contamos con layouts “independientes” (mismos elementos, pero diferentes organizaciones) para la orientación vertical y horizontal. Un ejemplo de estas diferentes configuraciones serían las imágenes 5 y 6 que representan la misma actividad en las dos posibles orientaciones

IMAGEN 5. ORIENTACIÓN VERTICAL

IMAGEN 6. ORIENTACIÓN HORIZONTAL

Diseño con modelos

A continuación, mostraremos la arquitectura de la aplicación. Al comienzo expondremos los diagramas de casos de uso del sistema, después los aspectos arquitectónicos desde un aspecto más global y por último el diseño de cada una de las capas de aplicación (Se incluirán los diagramas más comunes como son el diagrama de clases, de secuencia...).

Casos de uso

Los casos de uso incluidos en nuestra aplicación permiten realizar las operaciones necesarias para la gestión de cuentas básicas con varios participantes.

El diagrama de la Imagen 7 muestra los casos de uso del módulo cuentas, que son los casos de uso a los que se puede acceder inicialmente.

En la actividad principal de la aplicación el usuario podrá crear, cargar o finalizar una cuenta. También podrá acceder a la ayuda o a "Acerca de".

Al pulsar sobre el botón "Crear cuenta" se ofrecerá al usuario un formulario para indicar el nombre de la cuenta y los nombres de los participantes de la misma. Si por el contrario elige la opción "Cargar cuenta" se desplegará una lista con las cuentas disponibles actualmente en la base de datos. También podrá eliminar una cuenta eligiendo "Finalizar cuenta" así como obtener ayuda más detallada en la sección "Ayuda" u obtener nuestros datos de contacto en "Acerca de".

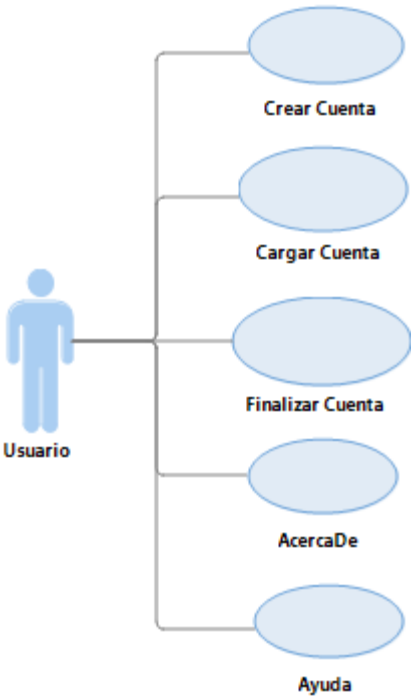

IMAGEN 7

Una vez lleguemos a la vista "Gestionar cuenta" (mediante "Cargar cuenta" o "Crear cuenta") dispondremos de los casos de uso mostrados en la imagen 8.

En esta actividad podremos indicar un pago (Transacción de dinero entre dos participantes) o un gasto (Aporte de dinero de un participante a la cuenta) así como ver la tabla de la cuenta con cierta información (Nombre de los participantes, total pagado por cada uno, importe por persona, cantidad a pagar o cobrar y cantidad total en la cuenta). También podremos realizar un aviso por cualquier medio de mensajería y añadir uno o varios participantes a la cuenta

Al pulsar sobre "Nuevo gasto" o "Nuevo pago" se mostrarán formularios con listas desplegables (para indicar los participantes) y campos para introducir la cantidad involucrada. La opción "Mostrar tabla de la cuenta" mostrará una tabla implementada mediante tecnologías web. Las opciones "Avisar participante" y "Añadir participante" mostrarán una ventana superpuesta a la principal que en el caso de "Añadir participante" pedirá el nombre del nuevo participante, mientras que en "Avisar participante" permitirá elegir el participante de una lista desplegable y luego la aplicación a usar para enviar el mensaje. Nuestra aplicación generará un mensaje personalizado para cada participante.

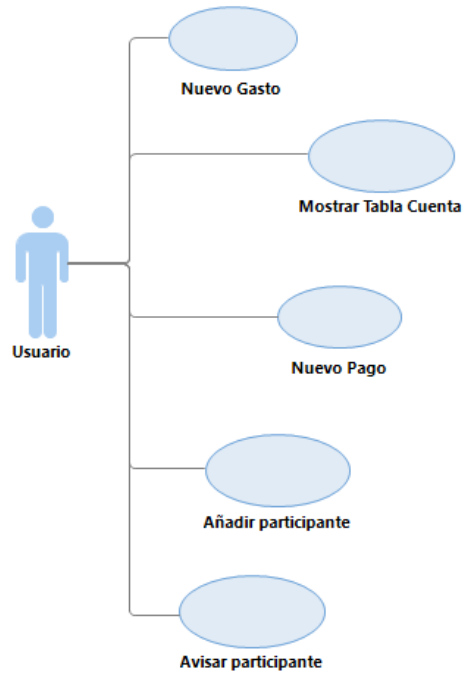

IMAGEN 8

Arquitectura

Para desarrollar nuestra aplicación hemos utilizado técnicas básicas de Ingeniería de Software.

Nuestra aplicación sigue una arquitectura multicapa, hecha explícitamente separando cada capa de la aplicación en paquetes con el nombre de su capa y, además, hemos seguido el patrón modelo-vista-controlador (MVC). Nuestras vistas están formadas por actividades que se componen de un fichero xml en el que se define la estructura de la vista y un fichero java en el que se implementa el comportamiento de la misma. Para mostrar los datos al usuario la vista dispondrá de adaptadores de modo que podremos incluirlos en elementos como Spinners. Además, las vistas intercambiarán información entre sí mediante objetos Bundle, que se asemejan a las clases utilizadas en el patrón "Transfer Object" pero que envían la mínima información entre las actividades.

El controlador está formado por métodos que accederán a la funcionalidad del modelo, que estará compuesto de las capas de negocio e integración.

La capa de negocio está formada por una clase que contiene todas las operaciones y se asemeja a los "Application Service" (Pero sin estar separadas en los dos módulos que tiene la aplicación, ya que es una aplicación bastante pequeña) y por las dos clases que representan los dos módulos de la aplicación (que serían nuestros objetos "Transfer") y accederá al DAO de la capa de integración para poder acceder a la información de la capa de datos.

Por último, la capa de integración está formada por objetos DAO que realizarán operaciones de búsqueda, inserción, eliminación, etc... Es decir, todas las operaciones relativas al acceso a la base de datos, la cual será una base de datos local SQLite que Android proporciona y facilita el acceso a la misma. La base de datos de imágenes la encontramos en la carpeta "res" de la propia aplicación.

En el siguiente apartado se expondrá el diseño generalizado de las capas de la aplicación.

Diseño de la arquitectura

En la imagen 9 mostramos, mediante diagramas de clases, todas las clases necesarias para hacer funcionar nuestra aplicación (No está en un tamaño visible porque más adelante habrá imágenes ampliadas de cada paquete). A la izquierda podemos observar el paquete presentación, arriba a la derecha el paquete negocio y abajo a la derecha el paquete integración. Como podemos observar hemos intentado que el acoplamiento sea el mínimo (Presentación solo se comunica con negocio a través del controlador, y Negocio solo se comunica con integración desde la clase Operaciones al DAO)

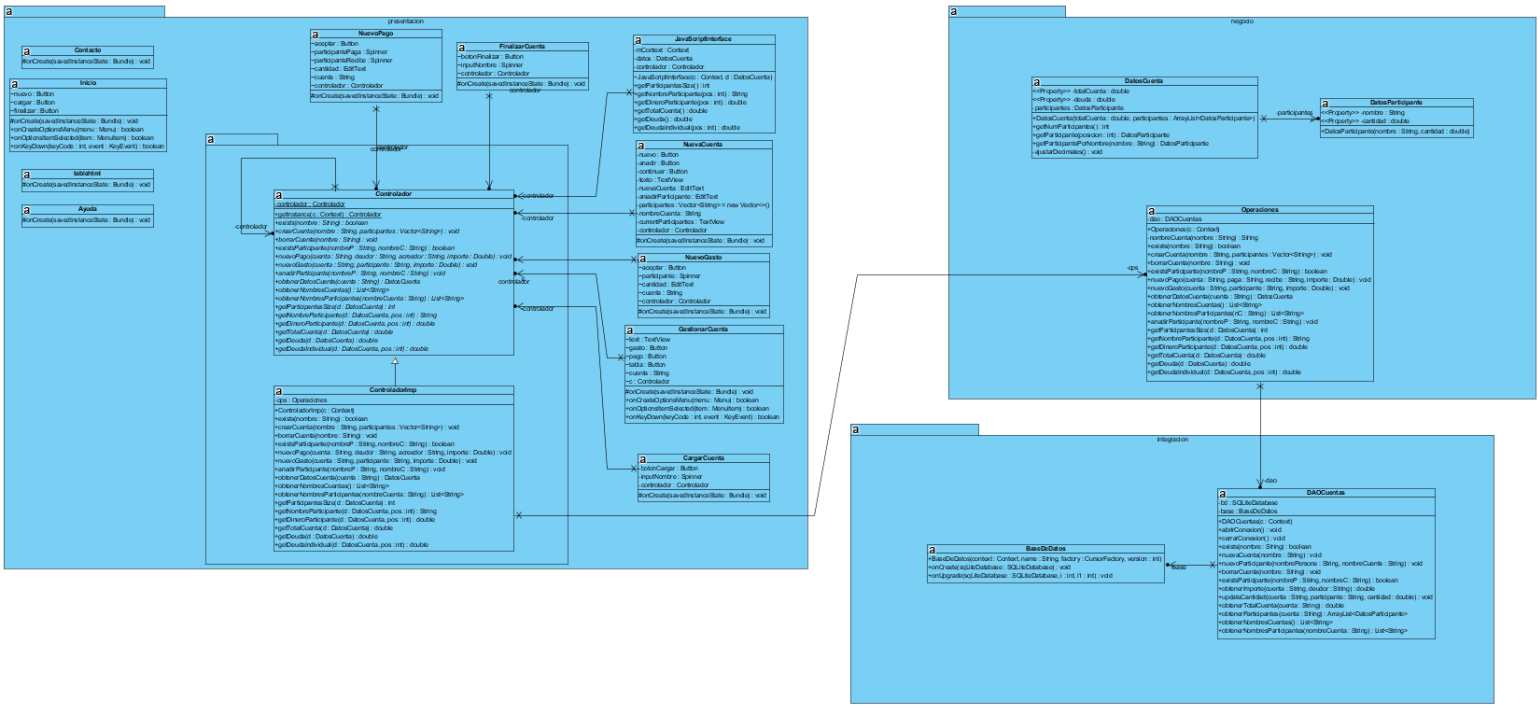

IMAGEN 9. DIAGRAMA DE CLASES DE LA APLICACIÓN.

En las siguientes imágenes se muestran los diagramas individuales de cada paquete. En la imagen 10 se muestra el paquete presentación, en la 11 el paquete negocio y en la 12 el paquete integración.

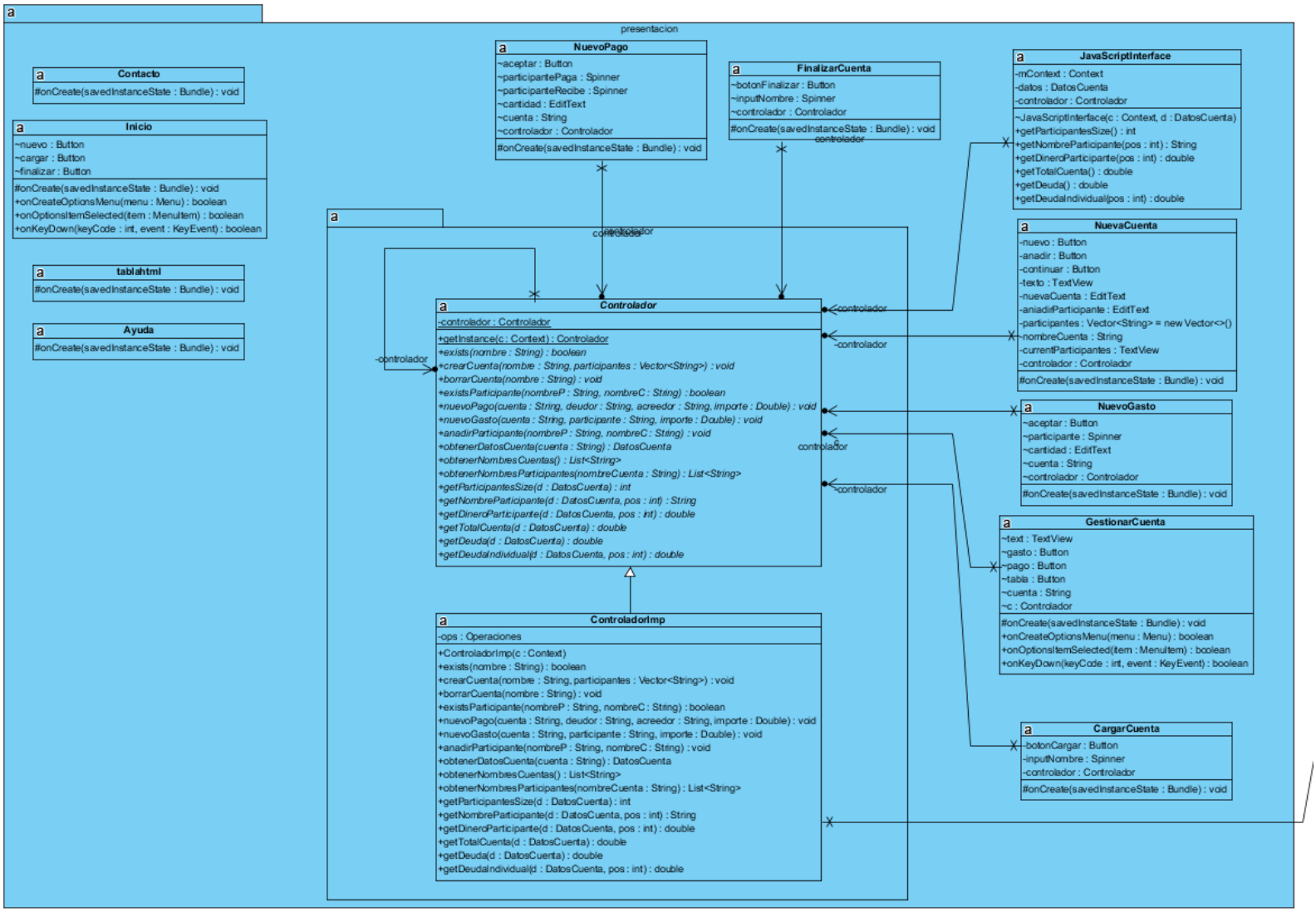

IMAGEN 10. DIAGRAMA DE CLASES DE PRESENTACIÓN.

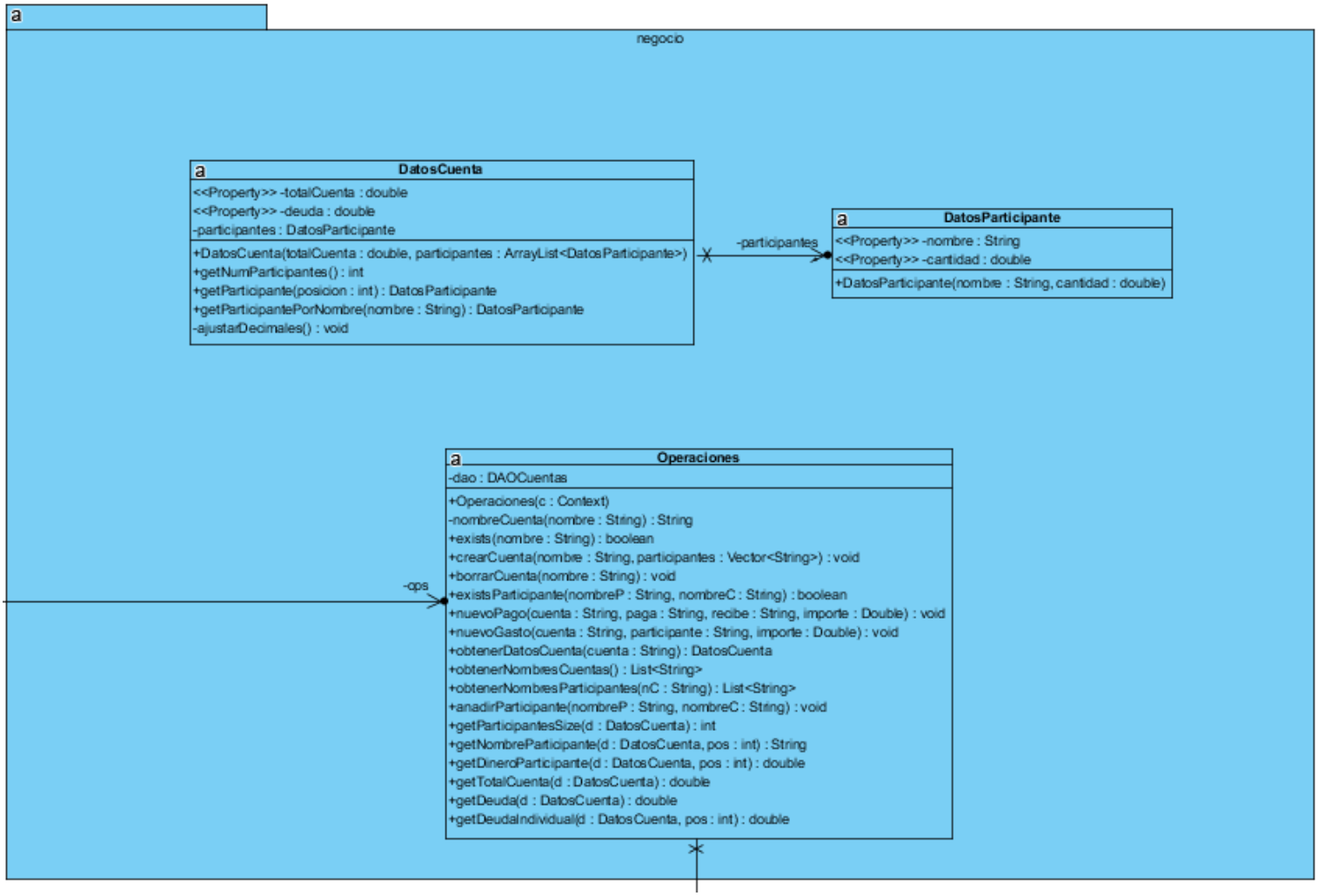

IMAGEN 11. DIAGRAMA DE CLASES DE NEGOCIO.

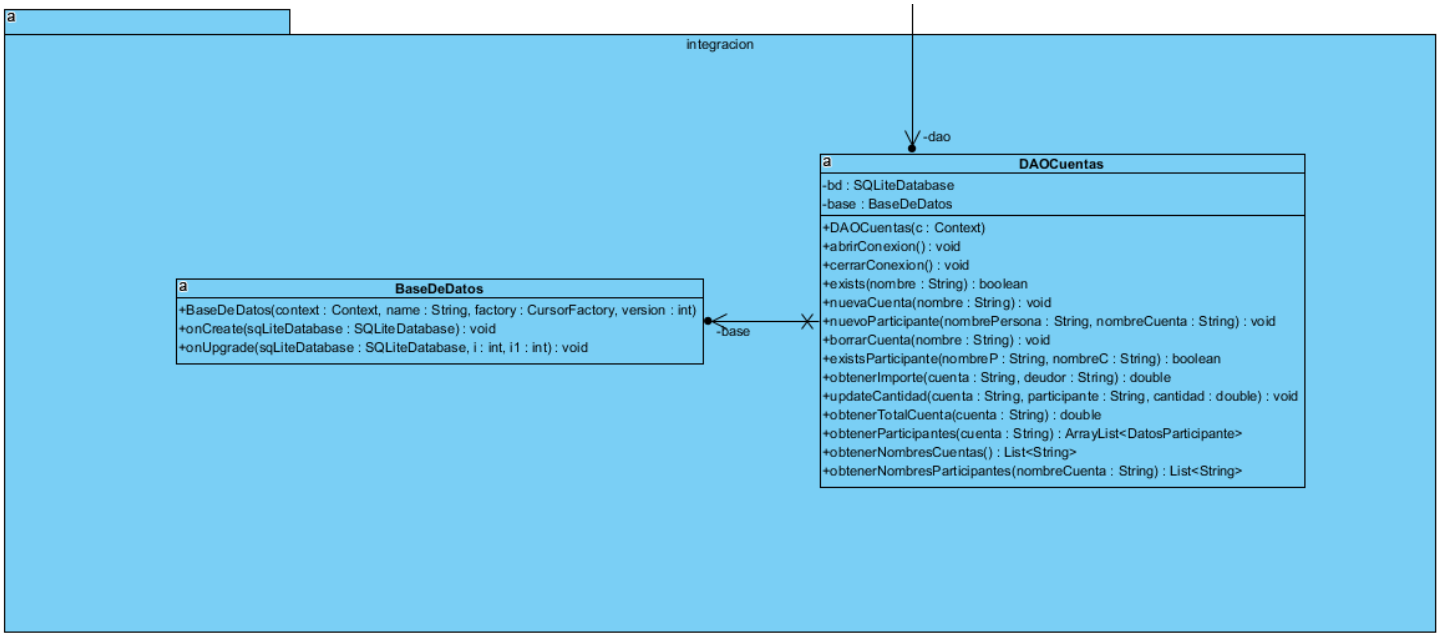

IMAGEN 12. DIAGRAMA DE CLASES DE INTEGRACIÓN.

Aportaciones de los integrantes, metodología de trabajo y conclusiones.

Para el desarrollo de la aplicación hemos usado una metodología que se asemejaba a cualquier metodología ágil, como Scrum o Kanban. El desarrollo se hacía en iteraciones de 1 semana tras la cual, cada lunes y/o martes nos reuníamos en el laboratorio para consultar nuestras dudas entre nosotros y con el profesor.

También disponíamos de una tabla de seguimiento muy sencilla (alojada en Google Keep, una aplicación para tomar notas en la que se puede incluir casillas de verificación al lado de cada párrafo) en la cual apuntábamos las tareas pendientes a modo de Backlog y al terminar la tarea se marcaba la casilla de verificación para que el otro compañero supiera que estaba completada, pero pudiera leerla y realizar pruebas si lo consideraba oportuno.

En cuanto al repositorio, hacíamos uso de GitHub para llevar un control de las versiones y Dropbox para compartir entre nosotros la versión más actualizada.

En lo referente a las fases de desarrollo, en la primera realizamos unos bocetos de las vistas en pareja para proponer ideas y dar el visto bueno a las mismas. Después llevamos a cabo el diseño de los layout mediante Android Studio y una vez terminados los diseños pasamos a implementar el comportamiento de estos.

A continuación implementamos la capa de negocio y al finalizar la misma la capa de integración, tras discutir que diseño sería más óptimo para la base de datos.