



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Appccounting: gestiona tu microempresa desde tu Smartphone

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Fabián Martínez González

**Tutor:** David De Andrés Martínez

2016



# Resumen

---

El presente Trabajo de Fin de Grado consiste en la creación de una aplicación para sistemas operativos Android. El trabajo se plantea como el desarrollo de la aplicación en sí y como un proyecto de negocio. Dada la situación actual económica de España y el elevado número de micro-empresas existentes, hay una buena oportunidad de negocio a largo plazo proporcionando un pequeño sistema de información que una micro-empresa se pueda permitir y le permita crecer con el uso de las Tecnologías de la Información. Por lo que mi modelo de negocio se basa en crear un producto inicial con el que empezar a conseguir usuarios además de mejorar y añadir nuevas funcionalidades a la aplicación con el paso del tiempo.

La aplicación se plantea como un pequeño sistema de información por lo que sus funcionalidades se extraen de sistemas de información existentes. Contiene funcionalidades de gestión de las relaciones con los clientes y de contabilidad. Es posible llevar la agenda de clientes y contactos, llevar una agenda de eventos además de controlar y visualizar los ingresos y gastos. Toda la información de eventos e ingresos y gastos es posible asociarla a clientes o contactos para facilitar el control del negocio, también es posible asociar notas con comentarios a los contactos o clientes.

**Palabras clave:** Android, micro-empresa, sistemas de información

# Abstract

---

This Project consists in the creation of an application for Android operating systems. The project deals both with the application development and the business model. Given the current economic situation in Spain and the large number of existing micro-enterprises, there is a good long term business opportunity by providing a small information system that a micro-enterprise can afford and let it grow using information technology. So that my business model is based on developing an initial product with which begin to get users besides improving and adding new functionality to the app over time.

The application is defined as a small information system so that its functionality is extracted from existing information systems. The application provides functionalities for customer relationship management and accounting. It is possible to manage clients and contacts, events, and incomes and expenses. It is also possible to associate all the information of events, incomes, and expenses with clients to ease business management. It is also possible to associate notes with comments to the clients.

**Keywords:** Android, micro-enterprise, Information system

# Tabla de contenidos

1	Introducción.....	8
1.1	Motivación .....	8
1.2	Objetivos.....	11
1.3	Estructura del documento.....	11
2	Visión empresarial del proyecto .....	12
3	Aplicaciones de gestión para pequeñas empresas.....	15
3.1	Xero Accounting Software.....	16
3.2	Base CRM.....	17
3.3	Conclusiones del estado del arte.....	18
4	Diseño y Especificación .....	20
4.1	Requisitos funcionales .....	20
4.2	Arquitectura de la aplicación .....	21
4.2.1	Capa de presentación (Interfaz de usuario) .....	21
4.2.3	Capa de negocio o lógica de la aplicación .....	36
4.2.4	Capa de persistencia.....	43
5	Implementación.....	45
5.1	Contexto tecnológico .....	45
5.1.1	Entorno de desarrollo .....	46
5.1.2	Java.....	47
5.1.3	Android .....	47
5.1.4	XML .....	47
5.1.5	SQLite .....	48
5.1.6	App Engine .....	48
5.1.7	Cloud Endpoints .....	49
5.1.8	Objectify.....	50
5.2	Desarrollo Android .....	51
5.2.1	Activities .....	51
5.2.2	Fragments.....	51
5.2.3	Layouts .....	52
5.2.4	Navigation drawer.....	52
5.2.5	Intents .....	52
5.2.6	Adapter.....	53
5.2.7	Recyclerview y Cardview .....	54
5.2.8	AsyncTask .....	54

5.3 Desarrollo e implementación del servidor.....	55
6 Resultados obtenidos .....	57
6.1 Pantalla principal .....	57
6.2 Nuevo cliente o contacto.....	59
6.3 Nuevo evento .....	60
6.4 Nuevo movimiento .....	61
6.5 Listar clientes y contactos.....	62
6.6 Listar eventos.....	63
6.7 Listar movimientos .....	64
6.8 Informes y gráficos .....	67
6.9 Ajustes.....	68
6.10 Detalles contacto y detalles cliente .....	69
6.11 Añadir nota .....	71
6.12 Editar evento .....	72
6.13 Editar contacto y editar cliente .....	73
6.14 Editar movimiento .....	74
6.15 Test de la aplicación .....	75
7 Conclusiones.....	76
7.2 Trabajos futuros.....	77
Bibliografía .....	78

# Índice de figuras

Figura 1 Comparación de tamaño del mercado de teléfonos en España	10
Figura 2 Evolución de la cuota de penetración de los sistemas operativos en smartphones en España	10
Figura 3 Lienzo para el modelo de negocio propuesto	12
Figura 4 Xero Accounting Software	16
Figura 5 Base CRM	17
Figura 6 Pantalla resumen	22
Figura 7 Menú desplegable	22
Figura 8 Ejemplo botón para añadir información	23
Figura 9 Nuevo cliente/contacto	23
Figura 10 Nuevo evento	24
Figura 11 Nuevo movimiento	25
Figura 12 Listar clientes	26
Figura 13 Listar eventos	27
Figura 14 Listar movimientos	28
Figura 15 Dialogo buscar movimientos	29
Figura 16 Dialogo clasificar movimientos	29
Figura 17 Pantalla clasificar movimientos	30
Figura 18 Informes y gráficos	31
Figura 19 Pantalla de ajustes	32
Figura 20 Dialogo crear usuario	33
Figura 21 Pantalla copia de seguridad	33
Figura 22 Detalles contacto/cliente	34
Figura 23 Modelo de datos empleado en la aplicación	43
Figura 24 Modelo de datos empleado en el servidor	44
Figura 25 Android Studio	46
Figura 26 Ejemplo Gradle	46
Figura 27 Ejemplo de uso MySQLiteOpenHelper	48
Figura 28 Estructura de la plataforma en la nube de Google	49
Figura 29 Ejemplo de código de Objectify	50
Figura 30 Ejemplo Intent implícito	53
Figura 31 Ejemplo código entidad User	55
Figura 32 Ejemplo código obtener copia de seguridad	56
Figura 33 Ejemplo código llamada a la API desde la aplicación	56
Figura 34 Aplicación final, Pantalla resumen	57
Figura 35 Aplicación final, Dialogo añadir información	57
Figura 36 Aplicación final, Panel lateral desplegable	58
Figura 37 Aplicación final, Nuevo cliente/contacto	59
Figura 38 Aplicación final, Añadir dirección	59
Figura 39 Aplicación final, Nuevo evento	60
Figura 40 Aplicación final, Introducir fecha	60
Figura 41 Aplicación final, Introducir hora	61
Figura 42 Aplicación final, Seleccionar cliente o contacto	61
Figura 43 Aplicación final, Nuevo movimiento	61
Figura 44 Aplicación final, Listar clientes	62

Figura 45 Aplicación final, Listar contactos	62
Figura 46 Aplicación final, Listar eventos	63
Figura 47 Aplicación final, Listar movimientos	64
Figura 48 Aplicación final, Pantalla buscar movimientos por fecha	65
Figura 49 Aplicación final, Dialogo buscar movimientos por fecha	65
Figura 50 Aplicación final, Dialogo agrupar movimientos	65
Figura 51 Aplicación final, Dialogo agrupar movimientos desplegable	65
Figura 52 Aplicación final, Pantalla agrupar movimientos	66
Figura 53 Aplicación final, Detalle pantalla agrupar movimientos	66
Figura 54 Aplicación final, Pantalla Gráficos	67
Figura 55 Aplicación final, Pantalla Ajustes	68
Figura 57 Aplicación final, Pantalla copia de seguridad	68
Figura 56 Aplicación final, Notificación Guardar copia de seguridad	68
Figura 58 Aplicación final, Notificación Restaurar copia de seguridad	68
Figura 59 Aplicación final, Pantalla detalles cliente	70
Figura 60 Aplicación final, Pantalla notas cliente	70
Figura 61 Aplicación final, Pantalla Eventos cliente	70
Figura 62 Aplicación final, Pantalla movimientos cliente	70
Figura 63 Aplicación final, Pantalla añadir nota	71
Figura 64 Aplicación final, Pantalla Editar evento	72
Figura 65 Aplicación final, Dialogo eliminar evento	72
Figura 66 Aplicación final, Pantalla editar contacto o cliente	73
Figura 67 Aplicación final, Pantalla Editar movimiento	74

## Índice de Tablas

Tabla 1 Resumen de funcionalidades de aplicaciones de la competencia	19
Tabla 2 Definición formal del caso de uso "Listar clientes"	37
Tabla 3 Definición formal del caso de uso "Listar eventos"	37
Tabla 4 Definición formal del caso de uso "Listar movimientos"	38
Tabla 5 Definición formal del caso de uso "Listar contactos"	38
Tabla 6 Definición formal del caso de uso "Añadir cliente o contacto"	38
Tabla 7 Definición formal del caso de uso "Añadir evento"	39
Tabla 8 Definición formal del caso de uso "Añadir movimiento"	39
Tabla 9 Definición formal del caso de uso "Añadir nota"	39
Tabla 10 Definición formal del caso de uso "Ver cliente o contacto"	40
Tabla 11 Definición formal del caso de uso "Editar cliente o contacto"	40
Tabla 12 Definición formal del caso de uso "Editar evento"	40
Tabla 13 Definición formal del caso de uso "Editar movimiento"	41
Tabla 14 Definición formal del caso de uso "Buscar movimientos por fecha"	41
Tabla 15 Definición formal del caso de uso "Listar movimientos clasificados"	41
Tabla 16 Definición formal del caso de uso "Ver gráficos"	42
Tabla 17 Definición formal del caso de uso "Hacer copia de seguridad"	42
Tabla 18 Definición formal del caso de uso "Restaurar copia de seguridad"	42
Tabla 19 Definición formal del caso de uso "Crear usuario"	43
Tabla 20 Definición formal del caso de uso "Iniciar sesión"	43

# 1 Introducción

## 1.1 Motivación

En la actualidad empresas de todo el mundo alcanzan el éxito gracias al uso de sistemas de información. En 2001 Laudon definió un sistema de información como “*un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control de una organización*” [1]. En otras palabras, un sistema de información es un conjunto de componentes software y hardware que colaboran para apoyar en la gestión de una empresa.

El origen de los sistemas de información comienza con las distintas soluciones software que fueron surgiendo para los departamentos de las empresas. La estructura organizativa tradicional de una empresa se basa en la separación en departamentos de la fuerza de trabajo disponible, para que cada departamento se centre en resolver sus tareas asignadas de la forma más eficaz y eficiente. Con la aparición de la informática se pudo, poco a poco, dar soporte a cada uno de los distintos departamentos de forma aislada.

Los potentes sistemas de información actuales se asemejan a sus raíces porque están formados por múltiples módulos de funcionalidad que cubren todas las áreas funcionales de una empresa de forma integrada.

Hay distintos tipos de sistemas de información, según el área funcional a la que apoyen, pero en la actualidad se utiliza el término ERP, siglas en inglés de Enterprise Resource Planning [2] que significa planificación de los recursos empresariales, para referirse a un sistema integrado de gestión que cubre todas las áreas funcionales de una organización.

El término ERP surge en los años noventa y procede por extensión de los sistemas existentes hasta la fecha [2]: sistema de gestión de inventarios MRP (Material Requirements Planning) de los años setenta y sistema de planificación de la producción MRP II (Manufacturing Resources Planning) de los años ochenta.

Como sistemas de información que se pueden diferenciar puedo mencionar los sistemas de gestión de la cadena de suministro SCM (Supply Chain Management), que se encargan de gestionar el flujo entre proveedores, fabricantes, distribuidores, minoristas y clientes de la empresa. También existen los sistemas de gestión de las relaciones con los clientes CRM (Customer Relationship Management), que ayudan a la fuerza de ventas, en el seguimiento de oportunidades de negocio, el control de la agenda y los contactos, etc [3]. Estos y otros sistemas están incluidos dentro de la concepción actual que hay de los ERP. También se puede mencionar que estos sistemas brindan a las empresas herramientas para obtener informes de la situación de la empresa, cuadros de mando e indicadores para ayudar a la toma de decisiones, gráficos con los datos del sistema, etc.

Sin embargo, y a pesar de todas las posibilidades que ofrecen los sistemas de información, la situación en España no es favorable para que la mayoría de empresas se beneficien con su uso. Según el retrato de las pyme publicado por el Ministerio de Industria, Energía y Turismo en enero de 2015 el 99.9% de las empresas españolas son pymes, en concreto un 55,1% son micro empresas sin asalariados, y un 40,8% son micro empresas entre 1 y 9 asalariados [4]. Este panorama implica que el acceso por parte del 55.1% de las empresas españolas a sistemas de información para facilitar la ejecución de sus procesos de negocio no sea viable, o por lo menos en la actualidad no hay las suficientes facilidades para que todas estas empresas puedan acceder a un sistema de información.

Los sistemas de información actualmente no son viables de implantar en esas empresas simplemente por motivos económicos: una empresa con tan pocos ingresos y con tan poco volumen de información no podría afrontar los costes que lleva implantar uno de estos sistemas, además de la gran infrautilización de recursos que tendría el sistema, debido a que actualmente estos sistemas están pensados para manejar grandes volúmenes de datos.

Teniendo en cuenta lo anteriormente comentado, me planteo la pregunta de si una microempresa podría beneficiarse del uso de sistemas de información u otro tipo de herramientas informáticas, y en caso de respuesta afirmativa, me planteo cómo podría ser la solución que se adaptase al tamaño de su empresa y qué funcionalidades serían de utilidad.

Suponiendo que una microempresa se podría beneficiar utilizando algún software para gestionar su información y ayudar en los procesos de negocio, me planteo desarrollar un software que sería de acceso gratuito, para que una microempresa que comienza con recursos mínimos tenga la posibilidad de gestionar la información de su negocio en un sistema de información sin realizar ningún desembolso de dinero y poder comprobar sin riesgos si necesita una gestión informatizada.

Además de este carácter de solución gratuita que sería necesario para un software de gestión de una microempresa, teniendo en cuenta ese 55.1% de microempresas unipersonales, sería buena idea que la solución ofreciese todas sus posibilidades en una aplicación para dispositivos móviles. Al ser una sola persona la que representa la empresa sería recomendable que pudiera disponer de toda su información en todo momento, y esto se consigue mejor con una aplicación para dispositivos móviles que con una aplicación convencional para ser usada en un ordenador.

Respecto a los dispositivos móviles, es evidente que han revolucionado la sociedad y han incrementado las posibilidades de aplicación de la informática en la vida de las personas. Hoy en día llevamos en el bolsillo teléfonos con la misma o más capacidad de cómputo que los ordenadores de hace unos años. Estos teléfonos son comúnmente conocidos como *smartphones* o teléfonos inteligentes, son teléfonos móviles que ofrecen más funcionalidades que un teléfono móvil común. La característica principal que los diferencia es la conectividad a Internet y posibilidad de instalar aplicaciones que incrementan las posibilidades de uso del dispositivo.

El incremento en el uso de los *smartphones* respecto a los teléfonos convencionales en España se puede apreciar en la Figura 1.

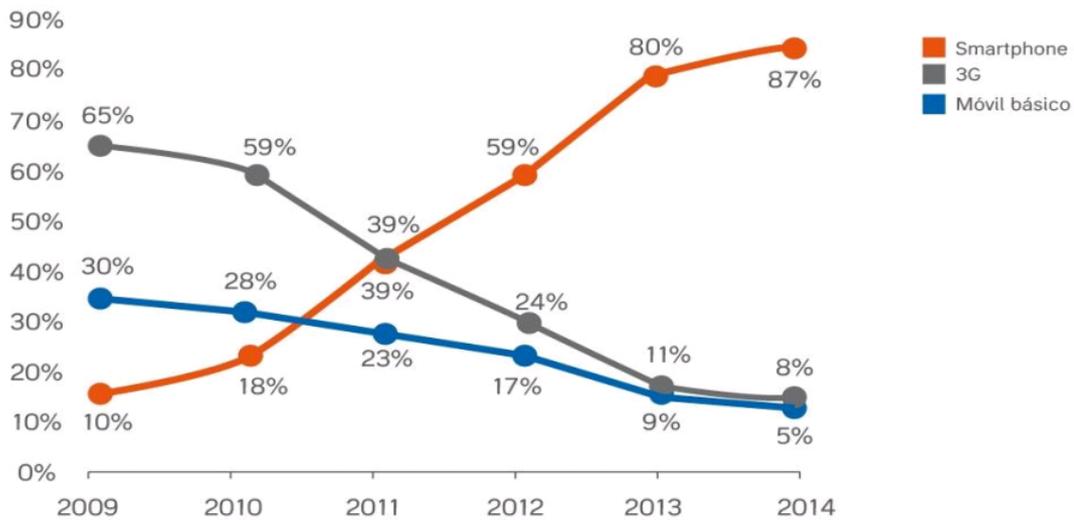


Figura 1 Comparación de tamaño del mercado de teléfonos en España

Pero para los *smartphones* que hay actualmente en el mercado hay distintos sistemas operativos, de los cuales hay uno que domina el mercado en España. Este es el sistema operativo Android. Esta información se puede apreciar en la Figura 2.

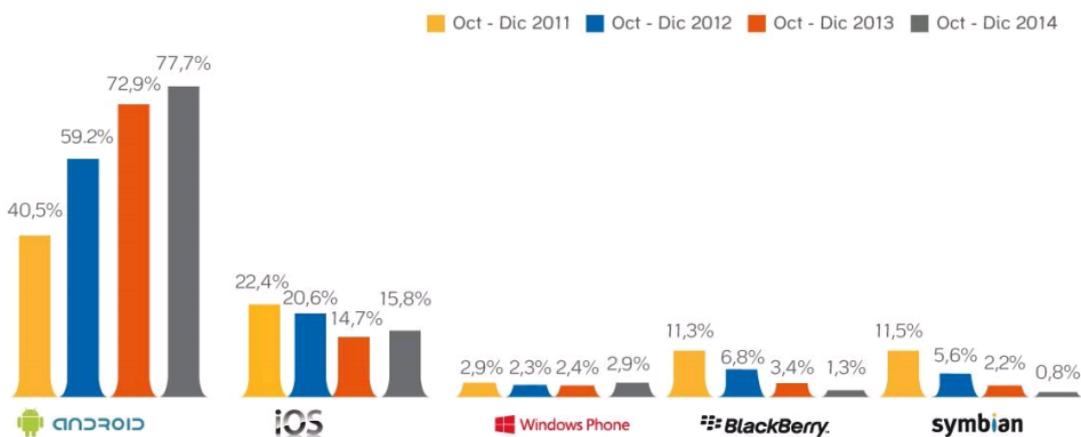


Figura 2 Evolución de la cuota de penetración de los sistemas operativos en smartphones en España

Por lo tanto la aplicación para empresas que pretendo desarrollar la desarrollaré inicialmente para el sistema operativo Android, por varios motivos, por la penetración que tiene en el mercado actual y para aprovechar los conocimientos que he adquirido este último año en la asignatura Soluciones Informáticas para Dispositivos Móviles.

Si este proyecto tiene éxito me plantearé en un futuro el desarrollo de la aplicación para otras plataformas, como iOS o Windows Phone.

## **1.2 Objetivos**

Así pues respecto a lo anteriormente mencionado, el objetivo del proyecto es realizar el desarrollo de una aplicación para sistemas operativos Android con la que se podrá llevar la gestión de una microempresa, principalmente enfocada a empresas formadas por una sola persona. La aplicación, que combinará funcionalidades de diferentes sistemas de información y las herramientas para el desarrollo de aplicaciones de Google, servirá para llevar en el Smartphone toda la información de una empresa. Así, pequeñas empresa con pocos recursos podrán tener en su bolsillo un software con toda su información de clientes, ingresos, gastos,... y que les ayudará a manejar su negocio y mantenerlo rentable.

Habrá que tener en cuenta que al ser una aplicación para dispositivos móviles la capacidad de cómputo de la que se va a disponer va a ser mucho menor de la que se dispondría en un sistema de información convencional, por lo que habrá que seleccionar qué funcionalidades de un sistema de información convencional se van a implementar y cuáles no. Será criterio para la selección de funcionalidades, aquellas que no hagan muy compleja la implementación de la aplicación, debido a la necesidad de ofrecer una aplicación de uso gratuito, o aquellas funcionalidades que no estén orientadas principalmente a una gran empresa.

## **1.3. Estructura del documento**

A continuación hago un pequeño resumen del contenido de cada capítulo del presente documento.

En el capítulo 2 haré un estudio de la viabilidad que tiene como negocio la realización de los objetivos de este trabajo, planteándome como sería el modelo de negocio a llevar a cabo.

En el capítulo 3 analizaré aplicaciones relacionadas con la que se desea realizar con el fin de sacar ideas y comenzar a plantearme cómo será mi aplicación.

En el capítulo 4 tomaré las decisiones de diseño y especificación de la aplicación, como la arquitectura o la lista de requisitos funcionales de la aplicación. También mostraré los bocetos realizados de la interfaz de usuario, así como el diagrama de flujo para comprender con facilidad cómo se relacionan los bocetos y la definición formal de los casos de uso. Por último se especificará la estructura de los datos que manejará mi aplicación.

En el capítulo 5 hablaré sobre las tecnologías usadas en este proyecto y explicaré los aspectos relevantes del proceso de implementación.

En el capítulo 6 mostraré la aplicación desarrollada con algunas explicaciones para facilitar la comprensión de las decisiones tomadas.

Y finalmente en el capítulo 7 valoro la aplicación obtenida con los objetivos propuestos, además de plantear los trabajos futuros relacionados con este trabajo.

## 2 Visión empresarial del proyecto

Como comento en la introducción, el objetivo de este trabajo es crear una aplicación que me ofrezca un plan de futuro, para poder seguir desarrollándola y si fuese posible ganar dinero haciéndolo. Este proyecto podría perfectamente convertirse en una StartUp, ya que es un proyecto escalable que puede ir creciendo con el tiempo y el aumento de usuarios. Y es por eso por lo que tengo que plantearme el modelo de negocio a adoptar.

Para formular mi modelo de negocio me apoyaré en el modelo del lienzo para modelos de negocio que Alexander Osterwalder desarrolló junto a Yves Pigneur en 2008 [5]. Es una herramienta estratégica para desarrollar nuevos modelos de empresa, o documentar modelos de negocio existentes, consistente en un gráfico visual con elementos que describen una empresa y sus productos (propuestas de valor), infraestructura, clientes y finanzas.

A continuación comentaré el lienzo para modelos de negocio de mi propuesta, que puede verse en la Figura 3.

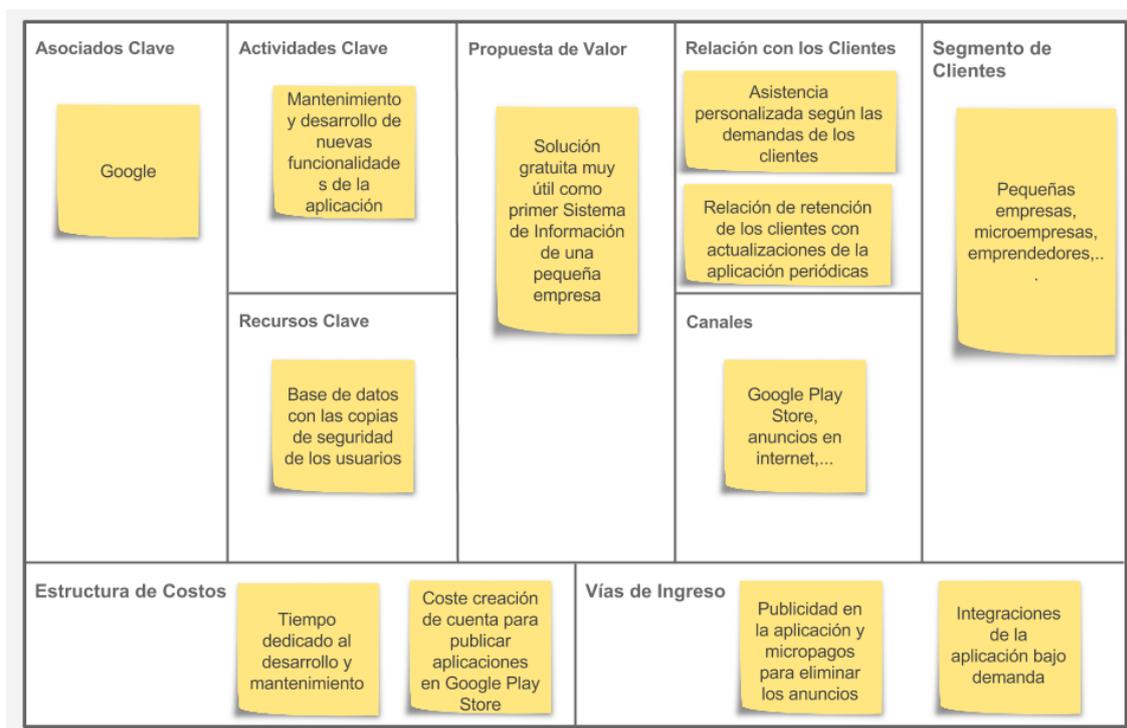


Figura 3 Lienzo para el modelo de negocio propuesto

El segmento de mercado al que está orientada esta aplicación es un nicho de mercado formado por microempresas, pequeñas empresas y emprendedores.

Y la propuesta de valor que ofrezco a este segmento de mercado, es poder tener una primera aproximación a lo que es la gestión de una empresa utilizando sistemas de

información. Es importante que esta inmersión en el uso de sistemas de información no produzca pérdidas económicas en el emprendedor o empresario que quiere comprobar si este tipo de gestión es necesario para él, por lo que el acceso a esta aplicación será totalmente gratuito.

Debido a esta propuesta de valor las vías de ingresos del negocio serán por publicidad en la aplicación, y en un futuro con la realización de trabajos de integración o mejoras de la aplicación por encargo.

Por lo tanto las relaciones con los clientes serán para obtener realimentación acerca del funcionamiento de la aplicación y poder realizar mantenimientos y actualizaciones a la aplicación. Además esto permitirá llevar a cabo futuras posibles integraciones con sistemas que utilicen en la empresa del usuario que solicite dicha integración.

Los sistemas operativos para dispositivos móviles disponen de aplicaciones como plataforma para la descarga de contenido digital, contenido que puede ser desde aplicaciones, libros o música. Google Play Store es la plataforma para descargar contenido utilizada en Android, es posible descargar aplicaciones, gratis o pagando e incluso hacer pagos dentro de las aplicaciones descargadas, y comprar contenido como música o libros. El canal de acceso al producto que ofrezco es el mismo que cualquier otra aplicación para Android.

Las actividades clave de las que constará este negocio son simplemente mi propia fuerza de trabajo y horas invertidas para el mantenimiento y actualización de la aplicación.

Como recurso clave de esta actividad de negocio está la base de datos donde se guardan las copias de seguridad con todos los datos que mis usuarios tendrán en la aplicación, estos datos pueden ser fuente de negocio, por ejemplo, como propósito de las integraciones anteriormente comentadas.

Como asociado clave al negocio está Google, que me proporciona tanto la infraestructura para que mi aplicación pueda ser accedida por millones de usuarios (Android y Google Play Store) como las herramientas para su desarrollo, parte de la que hablaremos más en detalle en apartados posteriores.

Y por último respecto a mi modelo de negocio tenemos la estructura de costes. Esta estructura a priori es muy simple, y se compone del desembolso inicial que hay que hacer para poder publicar contenido en Google Play Store, unos 25€ para toda la vida, y del coste de mi tiempo dedicado a desarrollar, mantener y actualizar la aplicación. El coste de desarrollarla inicialmente no habría que tenerlo en cuenta porque es un trabajo que tendría que realizar igualmente debido a que es el objetivo de este trabajo.

Una vez descrito mi modelo de negocio a través de los nueve bloques que componen el lienzo, tengo cubiertas las cuatro áreas principales de un negocio: clientes, oferta, infraestructura y viabilidad financiera.

Seguidamente tengo que destacar el reducido tiempo de comercialización que tengo para publicar la aplicación. Por varios motivos, porque quiero publicar la aplicación cuanto antes sea posible y poder conseguir la mayor cantidad de usuarios posible antes de la finalización de este trabajo, y porque hay unos plazos para entregar el TFG. Por lo

tanto al igual que hacen en las StartUps tengo que elaborar un producto viable mínimo (MVP) [6] con las funcionalidades mínimas que un usuario pueda aceptar e ir incrementando las funcionalidades de la aplicación con el tiempo.

### **3 Aplicaciones de gestión para pequeñas empresas**

Existen en el mercado diferentes aplicaciones móviles que ofrecen funcionalidades para la gestión de un negocio, aunque no todas tienen los mismos objetivos. Algunas están enfocadas a la gestión de stocks e inventario dando funcionalidades como las de un ERP, aunque este tipo de aplicaciones por norma general son de pago, debido a su complejidad y el coste de mantenimiento. Otras aplicaciones que se encuentran en el mercado son aplicaciones que se utilizan como complemento a un sistema de información previamente instalado en una empresa. Aunque hay aplicaciones para ser el complemento de sistemas muy populares, como podría ser SAP [7], también hay aplicaciones que se complementan con una web (a la que se accede pagando) para poder acceder a las mismas funcionalidades desde un ordenador que desde el dispositivo móvil, web.

Entre todas estas aplicaciones se han elegido dos aplicaciones, una cuyo enfoque se asemeja más al de los objetivos de este proyecto, es decir, aplicaciones enfocadas a pequeños negocios, y la otra porque implementa funcionalidades de un sistema CRM y ofrece una usabilidad digna de ser ejemplo.

### 3.1 Xero Accounting Software

Xero (ver Figura 4) es un software de contabilidad que cuenta en la actualidad con más de 700.000 usuarios suscritos a sus servicios (Fuente: [www.xero.com](http://www.xero.com)). Dispone de aplicación web y aplicación para dispositivos móviles Android y iOS, y aunque se puede descargar y utilizar de forma gratuita durante 30 días, se necesita registrarse y pagar para poder acceder de forma completa a sus funcionalidades. El objetivo principal es ofrecer un sistema de contabilidad bonito y sencillo de usar.



Figura 4 Xero Accounting Software

Entre las funcionalidades más interesantes que dispone, puedo destacar:

- Es posible crear, aprobar y enviar facturas directamente desde el dispositivo móvil
- Gestión de clientes, pudiendo guardar notas sobre los clientes, o visualizar los clientes en el mapa
- Permite gestionar cuentas bancarias, facturas y gastos.
- Introducir gastos desde la fotografía de un recibo.
- Es posible ver informes y balances con la información de cuentas bancarias, facturas y gastos introducidos.

### 3.2 Base CRM

Ofrece un sistema CRM utilizado en más de 7000 empresas, esta solución consta de una aplicación web todo en uno y una aplicación para iOS y Android. Se puede comenzar una prueba gratuita de 14 días para la aplicación web, aunque si solo se quiere utilizar la aplicación para dispositivos móviles es totalmente gratuito (ver Figura 5), pero para implantarlo en el negocio tiene unos costes mensuales. Con Base CRM la información está sincronizada en todo momento con la aplicación para dispositivos móviles, permitiendo movilidad total a los agentes de ventas (Fuente: [www.getbase.com](http://www.getbase.com)).

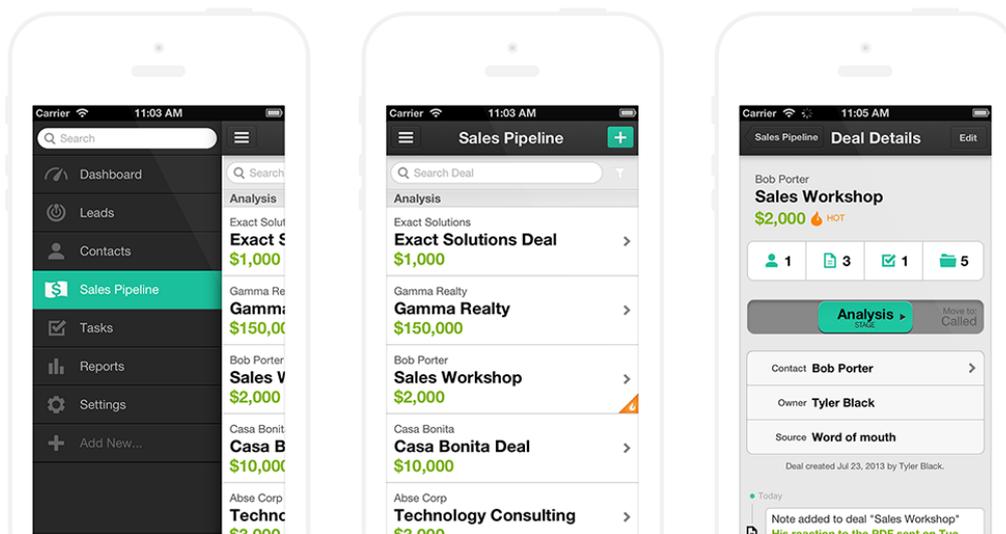


Figura 5 Base CRM

A continuación muestro algunas funcionalidades que ofrece la aplicación para dispositivos móviles Base CRM:

- Gestión de contactos y clientes, pudiendo asociar ventas, notas y eventos.
- Gestión del calendario y recordatorios.
- Seguimiento de las ventas.
- Ver localizados en un mapa los clientes y las oportunidades de venta.
- Creación de notas o recordatorios tras realizar una llamada.
- Información en tiempo real del rendimiento económico con informes, gráficos, etc.

### 3.3 Conclusiones del estado del arte

El motivo de este análisis de diferentes aplicaciones con objetivos similares o enfocadas al mismo sector, es extraer información para que el desarrollo de mi aplicación cumpla las expectativas de los usuarios que busco.

Es interesante para la decisión concreta de qué funcionalidades implementar en mi aplicación, ver y comparar que funcionalidades ofrecen aplicaciones exitosas en el mercado actual. También me ha sido de utilidad este análisis de la competencia para aclarar mis ideas en cuanto al diseño de las interfaces de la aplicación.

A continuación la Tabla 1 muestra un listado de funcionalidades y si están implementadas o no en las dos aplicaciones analizadas.

Tabla 1 Resumen de funcionalidades de aplicaciones de la competencia

<b>Funcionalidad</b>	<b>Xero Accounting Software</b>	<b>Base CRM</b>
<i>Gestión de clientes</i>	Si	Si
<i>Enviar facturas desde la aplicación</i>	Si	No
<i>Gestionar facturas y cuentas bancarias</i>	Si	No
<i>Informes y balance económico</i>	Si	Si
<i>Gestión del calendario</i>	No	Si
<i>Localización de clientes en un mapa</i>	Si	Si
<i>Asociar notas o recordatorios a clientes</i>	Si	Si
<i>Añadir notas o recordatorios al terminar una llamada</i>	No	Si
<i>Introducir un gasto desde una fotografía</i>	Si	No

Como se puede observar en la tabla 1 tras el análisis de las dos aplicaciones elegidas, puedo diferenciar qué funcionalidades son fundamentales de implementar, porque las aplicaciones analizadas las contemplan y son lo mínimo que un usuario de este tipo de aplicaciones podría esperar de un software de gestión para su empresa, o qué funcionalidades son interesantes para tenerlas en cuenta en el futuro para posibles actualizaciones, porque no todas las aplicaciones analizadas las implementan.

Para mostrar los resultados de los datos analizados utilizare el método MoSCoW [8]. Esta es una técnica muy utilizada en análisis de negocios y en el desarrollo de software para priorizar requerimientos y conseguir el consenso de las partes involucradas en un proyecto respecto a las decisiones que tomar para implementar o no funcionalidades. Según lo analizado tendríamos que:

La aplicación debe tener (funcionalidades que deben implementarse como parte del producto mínimo viable):

- Gestión de clientes y toda la información que se pueda relacionar con ellos, ya sea notas, recordatorios, eventos del calendario, facturas,...
- Mostrar informes y gráficos sobre la economía de la empresa con los datos introducidos en la aplicación. También se incluye mostrar un balance de ingresos y gastos.
- Poder ver localizados todos los clientes en un mapa, para conocer la expansión del negocio.

La aplicación debería tener, (funcionalidades importantes pero no críticas para el primer producto mínimo viable):

- Gestión del calendario con eventos y recordatorios. Esta funcionalidad debería estar para poder complementarse con la información que se puede asociar a los clientes.

La aplicación podría tener (funcionalidades deseables pero no necesarias):

- Creación y gestión de facturas además de poder enviarlas por correo electrónico a los clientes.
- Introducir gastos para el balance económico a partir de una fotografía, para agilizar la toma de datos.
- Poder añadir notas o recordatorios a un cliente al terminar una llamada.

La aplicación no tendrá (funcionalidades no apropiadas actualmente):

- Integración con redes sociales, pues es una funcionalidad que no se ha encontrado en las aplicaciones analizadas. Aunque si en un futuro se descubre que los usuarios de mi aplicación necesitan esa integración se podría añadir como actualización.

Antes de finalizar este apartado, cabe destacar que estas aplicaciones que he analizado, aplicaciones que se complementan un sistema en la nube, son buenos ejemplos del objetivo que pretendo alcanzar a largo plazo con mi proyecto. En un comienzo pequeñas empresas pueden empezar a gestionarse utilizando esta aplicación, pero si estas empresas crecen, probablemente necesitaran más formas de acceso al sistema de gestión que ofrece la aplicación, u otros servicios relacionados con dar mayor accesibilidad a los datos guardados en la aplicación. Y mi planteamiento respecto a esta situación es el propuesto en mi modelo de negocio, ofrecer un servicio de mantenimiento y personalización de la aplicación para mis clientes.

## 4 Diseño y Especificación

En este apartado se van a describir las decisiones que se han tomado en cuanto al diseño de todos los componentes de la aplicación, decisiones que afectan a cómo se van a desarrollar los componentes para poder alcanzar los objetivos de este proyecto.

Además, se detallará la lista de requisitos funcionales de la solución que se pretende dar, lista que me permitirá realizar la especificación formal de los componentes de la arquitectura que se detallará más adelante. Cabe destacar que prestaré atención en las decisiones que tengo que tomar para crear el mejor software posible en los plazos de tiempo más reducidos posibles

### 4.1 Requisitos funcionales

Una de las primeras cosas que se ha hecho al comenzar este proyecto ha sido decidir una lista cerrada de funcionalidades que la aplicación debe ofrecer. A continuación se muestra la lista que he decidido que compondrá el total de funcionalidades que la aplicación ofrecerá a los usuarios.

1. REQ001 – Gestión de contactos: se podrán crear dos tipos de contactos para diferenciar dos tipos de relaciones diferentes de una persona con la empresa (por ejemplo clientes y proveedores), se podrá actualizarlos y eliminarlos. Se podrán añadir notas a los contactos. Se podrán mostrar listas con los contactos.
2. REQ002 – Gestión del calendario: se podrán crear eventos en el calendario, actualizarlos y eliminarlos. Se listarán los eventos por días. Podrán estar asignados a un contacto y a una ubicación.
3. REQ003 – Gestión de movimientos de dinero: se podrán añadir ingresos y gastos, actualizarlos y eliminarlos. Se podrán listar todos juntos, solo ingresos o solo gastos. Se podrán buscar por fechas y se podrán clasificar para obtener información agregada de ingresos y gastos. Podrán estar asignados a un contacto.
4. REQ004 – Informes y gráficos: se podrá ver el histórico de ingresos y gastos de diferentes formas, con varios gráficos diferentes.
5. REQ005 – Mostrar la expansión del negocio: se podrán ver ubicados en un mapa todos los contactos.
6. REQ006 – Copias de seguridad y restauración de los datos.

## **4.2 Arquitectura de la aplicación**

De acuerdo a lo comentado en capítulos anteriores respecto a las funcionalidades a priorizar y respecto al reducido tiempo de comercialización que me he propuesto para publicar la aplicación, se ha decidido que la arquitectura utilizada en el desarrollo de esta aplicación se basará en el clásico modelo a tres capas [9], que separa el desarrollo del software en las capas de interfaz de usuario o capa de presentación, de la capa de negocio o lógica de la aplicación y de la capa de persistencia o datos.

Las ventajas de esta decisión es que podré aislar la lógica de la aplicación en componentes separados para poder reutilizarlos y reducir el tiempo de desarrollo y mantenimiento en la medida de lo posible.

Por otra parte y con el fin de desarrollar una capa de persistencia que encaje con los plazos para el desarrollo y a la vez ofrezca un servicio de calidad, los datos generados en la aplicación se almacenarán en una base de datos en el dispositivo móvil. Adicionalmente, si el usuario final lo desea, podrá crear un usuario en un servidor a desarrollar y realizar copias de seguridad o restaurarlas de forma manual, utilizando la conexión a internet del dispositivo para el intercambio de datos entre la aplicación y el servidor. Esta decisión tomada en cuanto a la persistencia ofrece seguridad y es aceptable en un comienzo, pero para una seguridad mayor y un uso más sencillo para el usuario final del sistema de copias de seguridad, refiriéndome al servidor en este caso, tendré en cuenta en el capítulo de trabajos futuros añadir opciones para realizar copias de seguridad periódicas, o automáticas.

A continuación pasaré a describir la especificación de las diferentes capas de la arquitectura de la aplicación.

### **4.2.1 Capa de presentación (Interfaz de usuario)**

En las fases iniciales del proyecto se han creado los diferentes bocetos que componen la interfaz de la aplicación de las funcionalidades que se quieren implementar para tomar una idea inicial de cómo sería la aplicación final y tener unos modelos en los que fijarse para la implementación de la interfaz. Para la realización de los bocetos se ha utilizado la aplicación web ninjamock [10], una herramienta gratuita para para realizar prototipos de aplicaciones para diferentes plataformas, y que es sencilla y rápida de usar.

En este apartado se describirán los diferentes prototipos creados y se indicará su relación con la lista de requerimientos funcionales anteriormente especificada. Y esto me permitirá definir en apartados posteriores los casos de uso de la aplicación.

#### 4.2.1.1 Pantalla principal o resumen

La pantalla de resumen (ver Figura 6) está pensada como pantalla inicial de la aplicación, ofrecerá información resumida y permitirá acceder al resto de funcionalidades de la aplicación. Dispondrá de un menú desplegable desde el lateral izquierdo (ver Figura 7) para acceder a las pantallas de clientes, contactos, eventos, movimientos y ajustes. También desde el elemento de resumen Balance y Gráficos podrán acceder a una pantalla con más gráficos. Esta pantalla además de ser el enlace para acceder a toda la funcionalidad de la aplicación, también cumple con el requerimiento funcional REQ005, que muestra la expansión del negocio. También se podrá acceder, en el recuadro de balance y gráficos, a la funcionalidad del requerimiento REQ004. Por último, en esta pantalla se podrá utilizar el botón circular para añadir nueva información a la aplicación, situado en la parte inferior derecha de la pantalla.

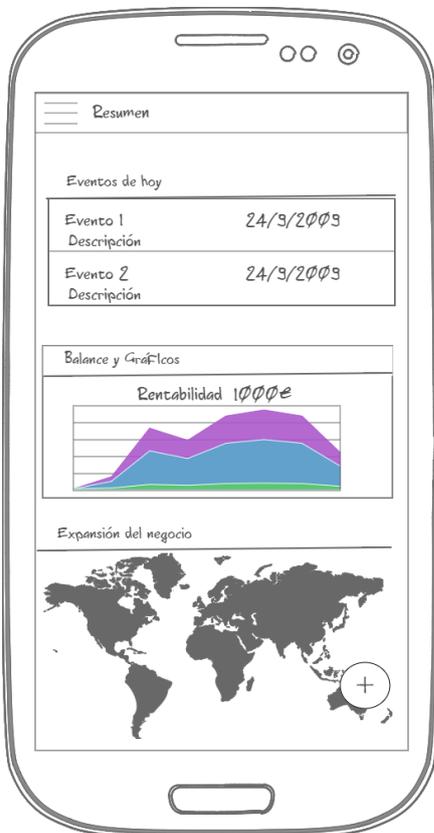


Figura 6 Pantalla resumen

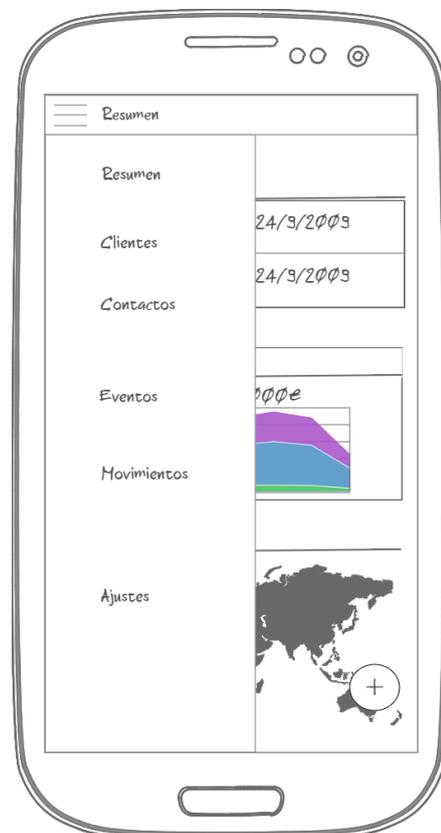


Figura 7 Menú desplegable

### 4.2.2 Añadir información

Será posible introducir información nueva a la aplicación a través del botón circular que se encuentra en la parte inferior derecha de algunas pantallas (ver Figura 8). La funcionalidad de este botón consistirá en un dialogo que mostrará las diferentes opciones de introducción de información que hay, el dialogo mostrará las opciones de añadir un contacto/cliente, un evento, un movimiento de dinero y una nota. Y según la opción escogida se abrirá el correspondiente formulario o pantalla.



Figura 8 Ejemplo botón para añadir información

### 4.2.1.3 Nuevo cliente o contacto

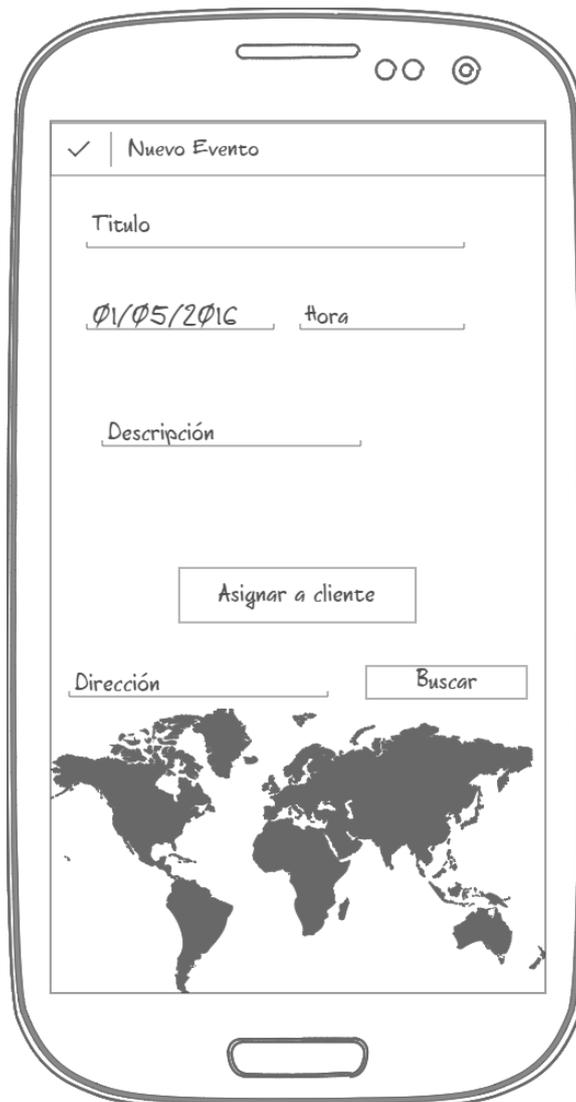
Esta pantalla (ver Figura 9) es el formulario para añadir un nuevo contacto o cliente, se podrá elegir de qué tipo se desea crear, contacto o cliente, y se podrá añadir información básica como nombre de la empresa, teléfono, email, web y dirección, se mostrará la dirección y la ubicación en el mapa. Esta pantalla será parte de la funcionalidad para el requerimiento funcional REQ001.



Figura 9 Nuevo cliente/contacto

#### 4.2.1.4 Nuevo evento

Este es el formulario (ver Figura 10) para añadir un nuevo evento. Será obligatorio que el evento tenga un título y fecha, la hora será opcional. Además un evento podrá estar relacionado a un contacto o cliente y a una ubicación. Esta pantalla será parte de la funcionalidad para el requerimiento funcional REQ002.



El diagrama muestra un formulario en un dispositivo móvil con el título "Nuevo Evento". El formulario contiene los siguientes campos y elementos:

- Un campo de texto etiquetado "Titulo".
- Un campo de texto etiquetado "Fecha" con el valor "01/05/2016" y un campo de texto etiquetado "hora".
- Un campo de texto etiquetado "Descripción".
- Un botón etiquetado "Asignar a cliente".
- Un campo de texto etiquetado "Dirección" y un botón etiquetado "Buscar".
- Un mapa del mundo que ocupa la parte inferior de la pantalla.

Figura 10 Nuevo evento

#### 4.2.1.5 Nuevo movimiento

Con este formulario (ver Figura 11) se podrán introducir gastos o ingresos que tendrán como mínimo una fecha y una cantidad de dinero. También se podrá añadir una descripción y asociarlo a un cliente o contacto. Esta pantalla será parte de la funcionalidad para el requerimiento funcional REQ003.

The image shows a hand-drawn sketch of a mobile application screen. At the top, there is a title bar with a checkmark icon and the text 'Nuevo movimiento'. Below the title bar, there are several input fields and a button. The first field is labeled 'Tipo:' and has a dropdown menu with 'Ingreso' selected. Below this, there are two fields: one for a date, containing '01/05/2016', and another for 'Cantidad'. Below these is a larger field for 'Descripción'. At the bottom of the form is a button labeled 'Asignar a cliente'. The entire form is enclosed in a rounded rectangular border, representing the mobile device screen.

Figura 11 Nuevo movimiento

#### 4.2.1.6 Listar clientes

Esta será la pantalla que se abrirá al acceder al elemento de menú de clientes desde el menú desplegable de la pantalla principal (ver Figura 12). Listará los clientes en dos órdenes diferentes, por clientes usados más recientemente, es decir consultados, asignados o introducidos y por orden alfabético. Esta pantalla será parte de la funcionalidad para el requerimiento funcional REQ001. Por último, en esta pantalla se podrá utilizar el botón circular para añadir nueva información a la aplicación, situado en la parte inferior derecha de la pantalla.

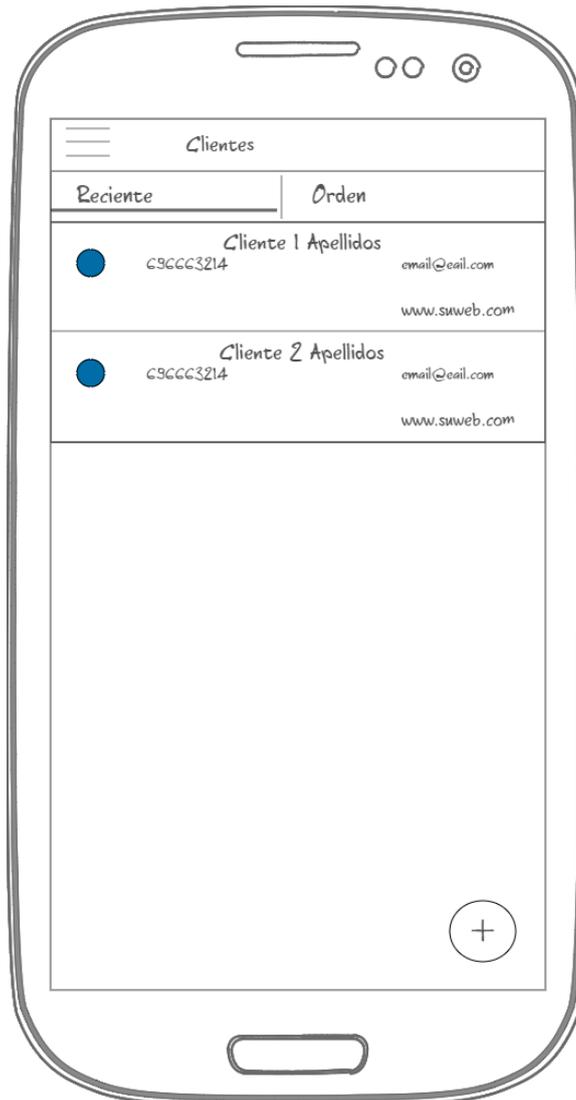


Figura 12 Listar clientes

#### 4.2.1.7 Listar contactos

La pantalla para listar contactos será similar a la de listar clientes, el único cambio será el contenido de las listas en sí y el título la pantalla. Debido a esta similitud con la pantalla de listar clientes, y para no cargar con excesivos bocetos, no se muestra el prototipo de esta pantalla. Esta pantalla será parte de la funcionalidad para el requerimiento funcional REQ001.

#### 4.2.1.8 Listar eventos

Esta pantalla (ver Figura 13) es la que se abrirá al acceder al elemento de menú de eventos desde el menú desplegable de la pantalla principal. Se abrirá con la lista de eventos del día actual, y se podrá ir navegando por el calendario para ver los eventos del día que el usuario desee. En esta pantalla se podrá utilizar el botón circular para añadir nueva información a la aplicación, situado en la parte inferior derecha de la pantalla. Esta pantalla cubrirá parte de la funcionalidad del requerimiento funcional REQ002.



Figura 13 Listar eventos

#### 4.2.1.9 Listar movimientos

A esta pantalla (ver Figura 14) se accederá al utilizar el elemento de menú de movimientos desde el menú desplegable de la pantalla principal. Se podrán visualizar o todos los movimientos, o solo los ingresos, o solo los gastos, en orden cronológico. Habrá dos opciones de menú en la parte superior derecha de la pantalla. Se podrá utilizar el botón circular para añadir nueva información a la aplicación, situado en la parte inferior derecha de la pantalla. El conjunto de prototipos que se mostrarán en este apartado cubren parte de la funcionalidad del requerimiento funcional REQ003.

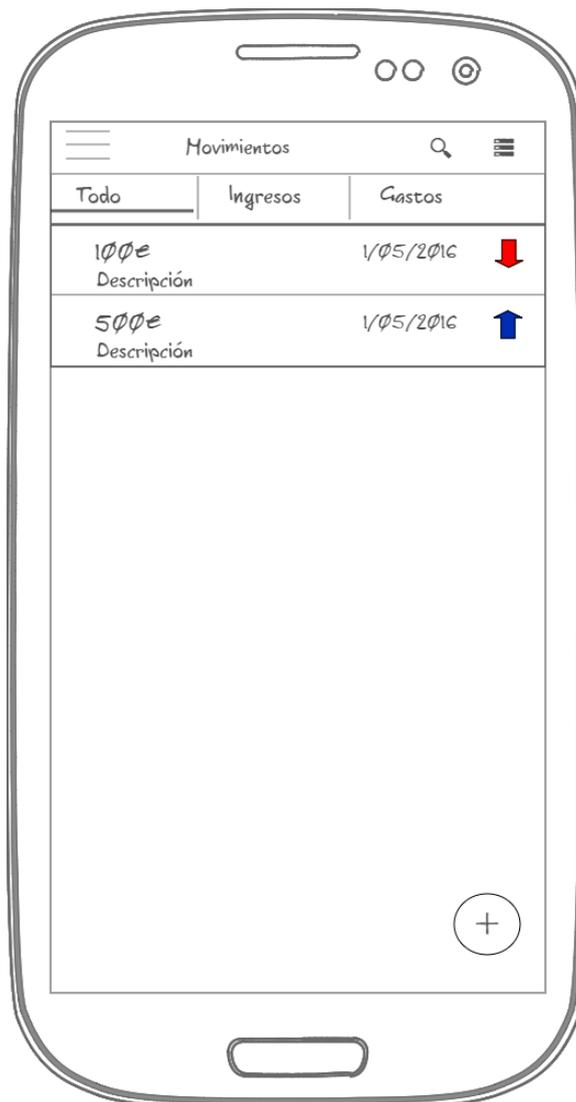


Figura 14 Listar movimientos

Los elementos de menú situados en la parte superior derecha de la pantalla servirán, respectivamente, para buscar movimientos por fecha (ver Figura 15) y para clasificarlos (ver Figura 16) y que se muestre información agregada de los movimientos.

Buscar	
Ingresos/Gastos	
De	<input type="text"/>
Hasta	<input type="text"/>
Cancel	OK

Figura 15 Dialogo buscar movimientos

El resultado de la búsqueda de movimientos es una pantalla con la lista de los movimientos encontrados, y por la simpleza de esa pantalla no se adjunta prototipo.

Clasificar	
Ingresos/Gastos	
Mes	
Cancel	OK

Figura 16 Dialogo clasificar movimientos

El resultado de la clasificación (ver Figura 16) mostrará una lista con el balance, ingresos menos gastos, ingresos totales y gastos totales en el periodo de agrupación seleccionado. A continuación muestro el prototipo de la pantalla que se abrirá al aceptar el dialogo de clasificar movimientos (ver Figura 17).

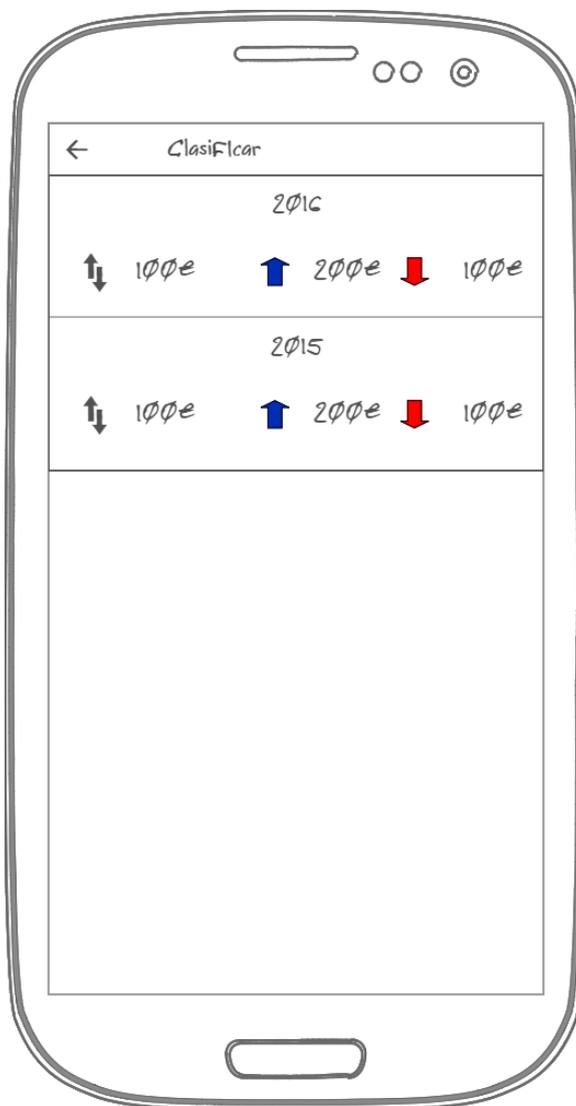


Figura 17 Pantalla clasificar movimientos

#### 4.2.1.10 Informes y gráficos

Esta pantalla (ver Figura 18) cubre las funcionalidades del requerimiento funcional REQ004. Se podrá acceder desde la pantalla principal de la aplicación y dispondrá de tres gráficos, uno mostrando el histórico de ingresos menos gastos, y dos gráficos de barras para mostrar los ingresos por mes, y los gastos por mes.

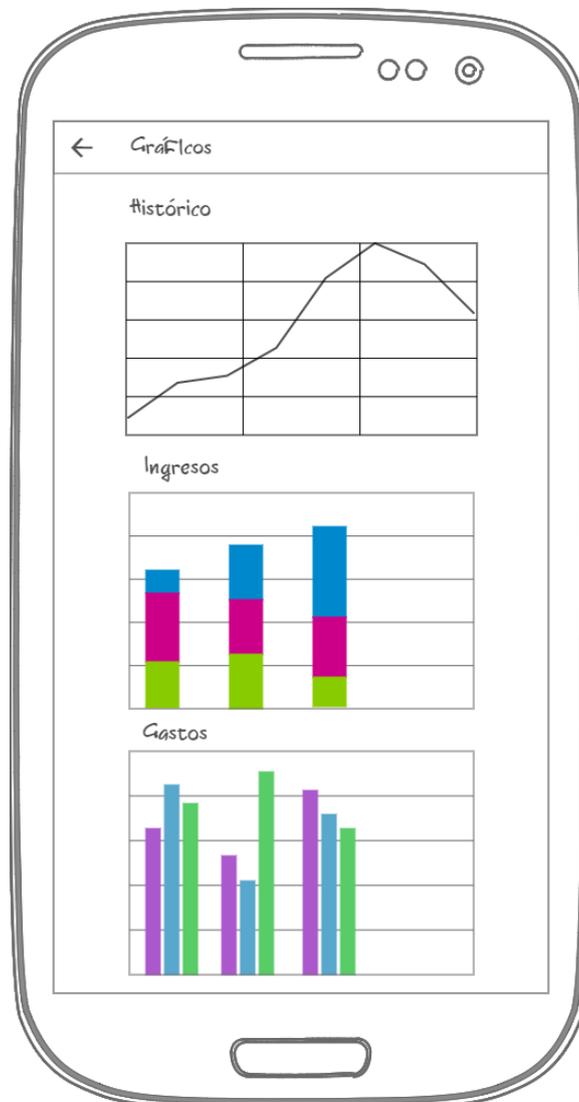


Figura 18 Informes y gráficos

#### 4.2.1.11 Pantalla de ajustes

Este es el prototipo para la pantalla de ajustes (ver Figura 19), donde se podrá acceder a la funcionalidad del requerimiento funcional REQ006. Será posible crearse un usuario gratuitamente, que permitirá acceder a una nueva pantalla para gestionar las copias de seguridad. Para la primera versión de la aplicación se espera que las operaciones que se puedan realizar con las copias de seguridad sean guardar copia y restaurarla.

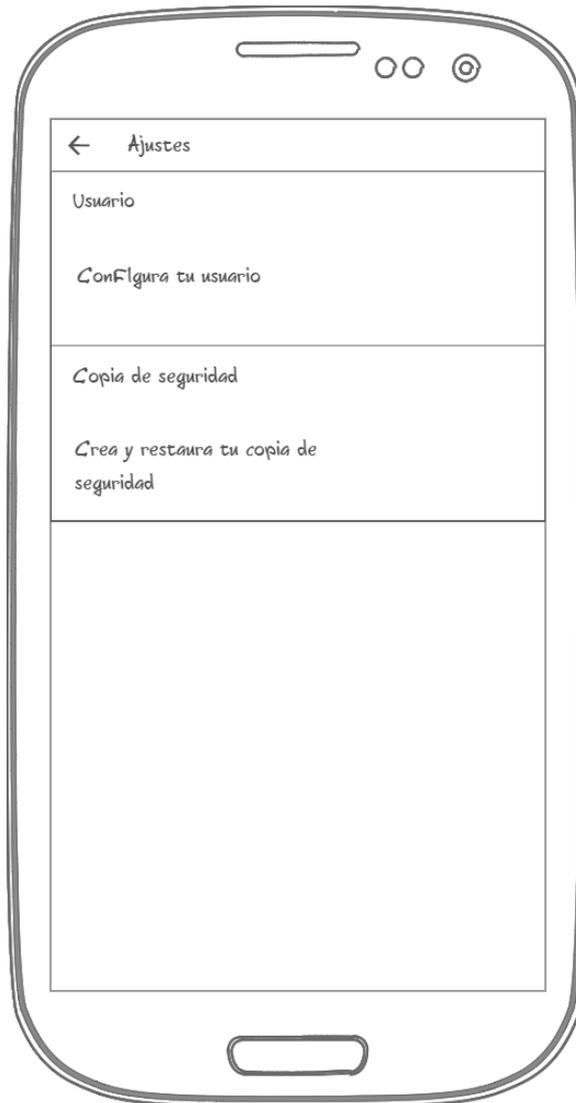


Figura 19 Pantalla de ajustes

Si el usuario no se ha registrado antes se le dará la opción de iniciar sesión con una cuenta creada anteriormente o registrarse.

Este será el dialogo (ver Figura 20) que se utilizará para la creación de un usuario.

Crear usuario	
Nombre	<input type="text"/>
Empresa	<input type="text"/>
Usuario	<input type="text"/>
Contraseña	<input type="password"/>
Hecho	Cancelar

Figura 20 Dialogo crear usuario

Una vez iniciada sesión se podrá acceder a la pantalla de copias de seguridad, que tendrá las siguientes opciones (ver Figura 21). Una vez seleccionada una acción, se mostrará la finalización con una notificación. Se mostrará como ha quedado la implementación de esas notificaciones.

← Copia de seguridad
Guardar copia de seguridad
Restaurar copia de seguridad

Figura 21 Pantalla copia de seguridad

#### 4.2.1.12 Ver detalles de contacto o cliente

Este es el prototipo (ver Figura 22) para poder visualizar toda la información relacionada con los contactos o clientes. Se podrá visualizar la información básica como sus datos o dirección, así como las notas, eventos y movimientos relacionados con ese contacto o cliente. Se podrá acceder a ver los detalles de un contacto o de un cliente al seleccionar un elemento de la lista de contactos o clientes respectivamente.

Para no cargar con excesivos prototipos solo se adjunta el prototipo para ver los detalles, las partes de notas, eventos y movimientos se compondrán cada una con su respectiva lista. Por último, estas funcionalidades forman parte del requerimiento funcional REQ001.



Figura 22 Detalles contacto/cliente

#### **4.2.1.13 Añadir nota**

Esta funcionalidad consistirá en poder añadir notas de texto a contactos o clientes, una nota tendrá que estar obligatoriamente relacionada a un solo contacto o cliente. El diseño de este prototipo es muy simple y no se adjunta imagen porque simplemente constaría de una lista y una entrada en la parte inferior de la pantalla para introducir el texto de la nota. Se mostrará en apartados posteriores cómo se ha implementado esta funcionalidad en la aplicación. Esta funcionalidad forma parte del requerimiento funcional REQ001.

#### **4.2.1.14 Editar contacto o cliente**

No se adjunta prototipo para esta pantalla porque se utilizará el mismo diseño que para introducir un nuevo contacto/cliente. Al editarlo se podrá cambiar el tipo, por ejemplo de contacto pasar a ser cliente. Se podrá acceder a editar un contacto/cliente desde un elemento de menú al visualizar los detalles del cliente. Esta funcionalidad es parte del requerimiento funcional REQ001.

#### **4.2.1.15 Editar evento**

No se adjunta prototipo para esta pantalla porque se utilizará el mismo diseño que para introducir evento nuevo. Se podrá acceder a editar un evento desde el listado de eventos. Esta funcionalidad es parte del requerimiento funcional REQ002.

#### **4.2.1.16 Editar movimiento**

No se adjunta prototipo para esta pantalla porque se utilizará el mismo diseño que para introducir un nuevo movimiento. Se podrá acceder a editar un movimiento desde la lista de movimientos. Esta funcionalidad es parte del requerimiento funcional REQ003.

#### **4.2.1.17 Eliminar contactos, eventos y movimientos**

En los prototipos anteriormente mostrados no se ha contemplado la funcionalidad para eliminar los objetos de la aplicación. Debido a su simpleza y para no cargar la memoria con excesivos prototipos no se adjuntan los prototipos de los casos de eliminar contactos o clientes, eventos y movimientos. Esta funcionalidad consistirá en un elemento de menú, al que se podrá acceder cuando se edita el objeto correspondiente, y que mostrará un dialogo de confirmación para eliminarlo. Se mostrará en apartados posteriores cómo se ha implementado en la aplicación alguno de estos casos.

### 4.2.3 Capa de negocio o lógica de la aplicación

Una vez clara la lista de requisitos que debe cumplir el total de funcionalidades de la aplicación, se puede realizar la especificación formal de los casos de uso. Con los casos de uso es posible especificar el funcionamiento de un sistema, y definir las acciones que el usuario puede realizar para interactuar con la aplicación. A continuación se realiza la definición formal de los casos de uso de la aplicación.

Tabla 2 Definición formal del caso de uso "Listar clientes"

<b>UC001</b>	<b>Nombre:</b> Listar clientes
<b>Descripción:</b> El usuario deberá acceder al menú desplegable desde el lateral izquierdo de la pantalla principal y elegir la opción Clientes. Se abrirá una pantalla con la lista de todos los clientes que se podrá ver en dos órdenes diferentes.	
<b>Entradas:</b> El usuario debe encontrarse en la pantalla principal o en alguna pantalla que permita abrir el menú desplegable (Listar clientes, contactos, listar eventos,...).	<b>Salidas:</b> Se abrirá la pantalla de listar clientes

Tabla 3 Definición formal del caso de uso "Listar eventos"

<b>UC002</b>	<b>Nombre:</b> Listar eventos
<b>Descripción:</b> Se debe acceder por el menú desplegable desde el lateral izquierdo desde las pantallas que lo permitan y elegir la opción eventos. Se abrirá una pantalla con un calendario, con el día de hoy seleccionado, y una lista que contendrá los eventos del día seleccionado en el calendario. Se podrá ir navegando por el calendario para ver los diferentes eventos de cada día.	
<b>Entradas:</b> El usuario debe de encontrarse en la pantalla principal o en alguna pantalla que permita abrir el menú desplegable.	<b>Salidas:</b> Se abrirá la pantalla de listar eventos.

Tabla 4 Definición formal del caso de uso "Listar movimientos"

<b>UC003</b>	<b>Nombre:</b> Listar movimientos	
<b>Descripción:</b> Se debe acceder por el menú desplegable desde el lateral izquierdo y elegir la opción movimientos. Se abrirá una pantalla con tres pestañas que mostrarán tres listas diferentes.		
<b>Entradas:</b> El usuario debe de encontrarse en la pantalla principal o en alguna pantalla que permita abrir el menú desplegable.		<b>Salidas:</b> Se abrirá la pantalla de listar movimientos

Tabla 5 Definición formal del caso de uso "Listar contactos"

<b>UC004</b>	<b>Nombre:</b> Listar contactos	
<b>Descripción:</b> El usuario deberá acceder al menú desplegable desde el lateral izquierdo de la pantalla principal y elegir la opción Contactos, se abrirá una pantalla con la lista de todos los contactos que se podrá ver en dos órdenes diferentes.		
<b>Entradas:</b> El usuario debe encontrarse en la pantalla principal o en alguna pantalla que permita abrir el menú desplegable.		<b>Salidas:</b> Se abrirá la pantalla listar contactos

Tabla 6 Definición formal del caso de uso "Añadir cliente o contacto"

<b>UC005</b>	<b>Nombre:</b> Añadir cliente o contacto	
<b>Descripción:</b> Se deberá acceder al botón de añadir información y elegir la opción de Añadir Cliente/Contacto. Se deberá rellenar al menos los datos mínimos del formulario y elegir si es cliente o contacto (por defecto estará seleccionado Contacto), y por ultimo pulsar el botón de guardar.		
<b>Entradas:</b> Datos del cliente.		<b>Salidas:</b> Si se han introducido los datos mínimos, se cierra la pantalla y se crea el nuevo cliente o contacto. Si no, un mensaje notifica al usuario que introduzca los datos mínimos.

Tabla 7 Definición formal del caso de uso "Añadir evento"

<b>UCoo6</b>	<b>Nombre:</b> Añadir evento	
<b>Descripción:</b> Se deberá acceder al botón de añadir información y elegir la opción de Añadir evento. Se deberá rellenar al menos los datos mínimos del formulario y pulsar el botón de guardar.		
<b>Entradas:</b> Datos del evento.		<b>Salidas:</b> Si se han introducido los datos mínimos, se cierra la pantalla y se crea el nuevo evento. Si no, un mensaje notifica al usuario que introduzca los datos mínimos.

Tabla 8 Definición formal del caso de uso "Añadir movimiento"

<b>UCoo7</b>	<b>Nombre:</b> Añadir movimiento	
<b>Descripción:</b> Se deberá acceder al botón de añadir información y elegir la opción de Añadir movimiento. Se deberá introducir como mínimo la cantidad, la fecha se pondrá automáticamente, y se deberá elegir de que tipo es Ingreso o Gasto (por defecto es Ingreso), y por último se pulsará el botón de guardar.		
<b>Entradas:</b> Datos del movimiento.		<b>Salidas:</b> Si se ha introducido la cantidad, se creará el nuevo movimiento. Si no, se mostrará un mensaje al usuario indicando que rellene ese campo.

Tabla 9 Definición formal del caso de uso "Añadir nota"

<b>UCoo8</b>	<b>Nombre:</b> Añadir nota	
<b>Descripción:</b> Se deberá acceder al botón de añadir información y elegir la opción de Añadir nota. Aparecerá una lista para elegir a quién se quiere añadir la nota y finalmente habrá una lista con todas las notas de ese contacto o cliente y se podrá añadir una nota al final de la lista.		
<b>Entradas:</b> Contacto o cliente a quien asignarle la nota y el texto de la nota.		<b>Salidas:</b> Lista de notas con la nueva nota al final.

Tabla 10 Definición formal del caso de uso "Ver cliente o contacto"

<b>UCo09</b>	<b>Nombre:</b> Ver cliente o contacto	
<b>Descripción:</b> Se deberá seleccionar un elemento en la lista de clientes o contactos, y se abrirá una nueva pantalla donde se podrá ver la información personal de ese contacto o cliente, además de sus notas, eventos y movimientos.		
<b>Entradas:</b> Elemento seleccionado en la lista de clientes o contactos.		<b>Salidas:</b> Nueva pantalla con la información del cliente o contacto seleccionado.

Tabla 11 Definición formal del caso de uso "Editar cliente o contacto"

<b>UCo10</b>	<b>Nombre:</b> Editar cliente o contacto	
<b>Descripción:</b> Una vez dentro de los detalles de un contacto o cliente se deberá utilizar la opción de menú para entrar en la pantalla de edición. Una vez ahí se podrá modificar la información. Si se dejan sin informar los campos mínimos no se podrá guardar el cambio.		
<b>Entradas:</b> Cliente seleccionado.		<b>Salidas:</b> Si se han informado los datos mínimos se guardará la modificación. Si no, se notificará al usuario para que informe los datos mínimos.

Tabla 12 Definición formal del caso de uso "Editar evento"

<b>UCo11</b>	<b>Nombre:</b> Editar evento	
<b>Descripción:</b> Se deberá seleccionar un evento de la pantalla lista de eventos, y se abrirá una pantalla nueva donde se podrá ver la información completa del evento, editar la información del evento y guardar los cambios pulsando el botón de guardar. Si los datos mínimos no están informados no se podrán guardar los cambios.		
<b>Entradas:</b> Evento seleccionado.		<b>Salidas:</b> Si se han informado los datos mínimos se guardará la modificación. Si no, se notificará al usuario para que informe los datos mínimos.

Tabla 13 Definición formal del caso de uso "Editar movimiento"

<b>UC012</b>	<b>Nombre:</b> Editar movimiento	
<b>Descripción:</b> Se deberá seleccionar un movimiento de la pantalla de lista de movimientos, y se abrirá una nueva pantalla en la que se podrá ver la información completa del movimiento y modificarla, pudiendo guardarla pulsando el botón de guardar. Si la cantidad no está informada no se podrán guardar los cambios.		
<b>Entradas:</b> Movimiento seleccionado.		<b>Salidas:</b> Si se han informado los datos mínimos se guardará la modificación. Si no, se notificará al usuario para que introduzca como mínimo la cantidad.

Tabla 14 Definición formal del caso de uso "Buscar movimientos por fecha"

<b>UC013</b>	<b>Nombre:</b> Buscar movimientos por fecha	
<b>Descripción:</b> Desde la pantalla de listar movimientos utilizando una opción de menú se podrán buscar movimientos, ya sean todos o solo ingresos o solo gastos, de una fecha determinada o en un rango de fechas. Una vez seleccionados los criterios de búsqueda se abrirá una pantalla que tendrá una lista con los movimientos encontrados.		
<b>Entradas:</b> El usuario debe encontrarse en la pantalla de listar movimientos y debe seleccionar los criterios de búsqueda además de seleccionar una o dos fechas.		<b>Salidas:</b> Se abrirá una nueva pantalla con una lista con los movimientos en orden cronológico.

Tabla 15 Definición formal del caso de uso "Listar movimientos clasificados"

<b>UC014</b>	<b>Nombre:</b> Listar movimientos clasificados	
<b>Descripción:</b> Desde la pantalla de listar movimientos, utilizando una opción de menú, se podrán clasificar los movimientos, ya sean todos o solo ingresos o solo gastos, por mes o por año. Una vez elegidos los criterios de clasificación se abrirá una nueva pantalla con información del balance, ingresos y gastos agregada según el periodo escogido.		
<b>Entradas:</b> El usuario debe encontrarse en la pantalla de listar movimientos y debe seleccionar los criterios de clasificación.		<b>Salidas:</b> Se abrirá una nueva pantalla con una lista con los movimientos agrupados según el periodo escogido. Si se selecciona un elemento de esta lista se abrirá una nueva pantalla con una lista de los movimientos que se encuentren en el periodo seleccionado.

Tabla 16 Definición formal del caso de uso "Ver gráficos"

<b>UC015</b>	<b>Nombre:</b> Ver gráficos	
<b>Descripción:</b> En la pantalla principal se podrá ver un gráfico que reflejará el estado de ingresos y gastos del mes actual. Si se seleccionan esos gráficos se abrirá una nueva pantalla que contendrá tres gráficos para mostrar la situación económica a lo largo del tiempo de uso de la aplicación.		
<b>Entradas:</b> El usuario debe encontrarse en la pantalla principal de la aplicación.		<b>Salidas:</b> Si el usuario ha seleccionado la caja de los gráficos en la pantalla principal se abrirá una nueva pantalla con el resto de gráficos.

Tabla 17 Definición formal del caso de uso "Hacer copia de seguridad"

<b>UC016</b>	<b>Nombre:</b> Hacer copia de seguridad	
<b>Descripción:</b> Si el usuario se ha registrado o ha iniciado sesión anteriormente en el servidor y entra en Ajustes, selecciona la opción de copias de seguridad y selecciona la opción de Guardar copia de seguridad, comenzará un proceso que envía al servidor toda la información de clientes, contactos, eventos, movimientos y notas y la guarda asociada a la cuenta de usuario con la que se ha iniciado sesión anteriormente.		
<b>Entradas:</b> Usuario registrado en el servidor.		<b>Salidas:</b> Aparecerá una notificación en el teléfono cuando termine el proceso de envío de los datos.

Tabla 18 Definición formal del caso de uso "Restaurar copia de seguridad"

<b>UC017</b>	<b>Nombre:</b> Restaurar copia de seguridad	
<b>Descripción:</b> Si el usuario se ha registrado o ha iniciado sesión anteriormente en el servidor y entra en Ajustes, selecciona la opción de copias de seguridad y selecciona la opción de Restaurar copia de seguridad, comenzará un proceso que recibirá del servidor y almacenará en el teléfono toda la información de clientes, contactos, eventos, movimientos y notas asociada a la cuenta de usuario con la que se ha iniciado sesión anteriormente.		
<b>Entradas:</b> Usuario registrado en el servidor.		<b>Salidas:</b> Aparecerá una notificación en el teléfono cuando termine el proceso de descarga de los datos.

Tabla 19 Definición formal del caso de uso "Crear usuario"

<b>UC018</b>	<b>Nombre:</b> Crear usuario	
<b>Descripción:</b> Si el usuario no estaba registrado antes y entra en Ajustes, selecciona la opción de Usuario y elige en el diálogo que le aparecerá la opción de Crear usuario. Aparecerá un nuevo dialogo para rellenar, como mínimo deberá introducir su nombre, un nombre de usuario y su contraseña. Una vez registrado podrá acceder a la opción copias de seguridad en la pantalla de Ajustes.		
<b>Entradas:</b> Encontrarse en la pantalla de Ajustes y no estar registrado ni autenticado anteriormente.		<b>Salidas:</b> Si el nombre de usuario introducido no está usado se creará el nuevo usuario, en caso contrario se indicará al usuario de que escoja otro nombre de usuario.

Tabla 20 Definición formal del caso de uso "Iniciar sesión"

<b>UC019</b>	<b>Nombre:</b> Iniciar sesión	
<b>Descripción:</b> Si el usuario no estaba autenticado anteriormente, y accede a la opción de Usuario en la pantalla de Ajustes, y elige la opción de iniciar sesión del diálogo que aparecerá. Se abrirá un nuevo diálogo para introducir el nombre de usuario y la contraseña.		
<b>Entradas:</b> No estar autenticado anteriormente y encontrarse en la pantalla de Ajustes.		<b>Salidas:</b> Si los datos introducidos son correctos se podrá acceder a la opción copias de seguridad en la pantalla de Ajustes, en caso contrario se informará al usuario del error.

Respecto a los casos de uso de eliminar información que se puede crear en la aplicación, ya sea clientes, contactos, eventos y movimientos. Son unos casos de uso muy similares y por no extender con demasiadas definiciones formales la memoria no se incluyen los casos de uso de "Eliminar cliente o contacto", "Eliminar evento", "Eliminar movimiento".

El funcionamiento de estos tres casos de uso es muy similar, será requisito encontrarse editando un cliente o contacto, un evento o un movimiento y pulsar el botón para eliminar situado en la parte superior derecha de la pantalla. Seguidamente se abrirá un dialogo de confirmación para eliminarlo.

#### 4.2.4 Capa de persistencia

En este apartado se presenta el modelo de datos utilizado, modelo que se utilizará para almacenar toda la información generada por la aplicación. La forma más común de presentar este modelo es con un esquema entidad relación, que permite definir los objetos del mundo real como entidades que tienen atributos y relaciones entre las entidades.

A continuación, en la Figura 23, se muestra el esquema entidad-relación en cuestión.

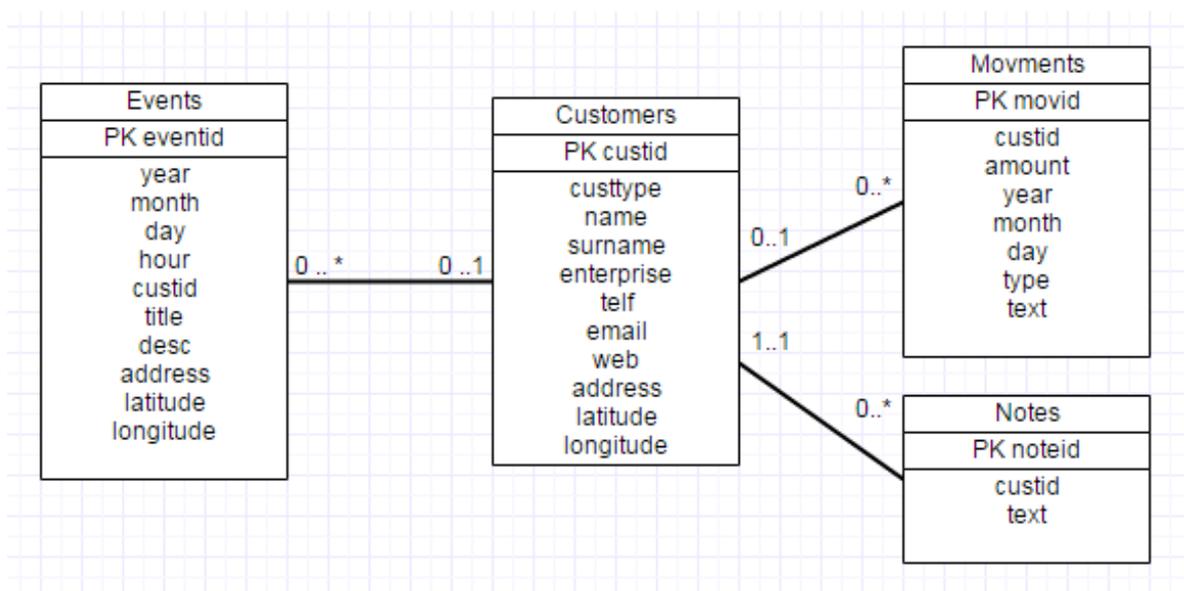


Figura 23 Modelo de datos empleado en la aplicación

Del esquema entidad relación se puede extraer que las relaciones entre las distintas entidades serán las siguientes: un cliente puede tener cero o muchos eventos, movimientos y notas, pero una nota solo puede estar asociado a uno y solo a un cliente, un evento puede estar asociado a ninguno o a un cliente, e igual para los movimientos.

Respecto a la especificación del modelo de datos cabe destacar que en el servidor que se utilizará para hacer las copias de seguridad se usará otro modelo de datos (ver Figura 24), muy similar al anteriormente mostrado pero con algunas diferencias.

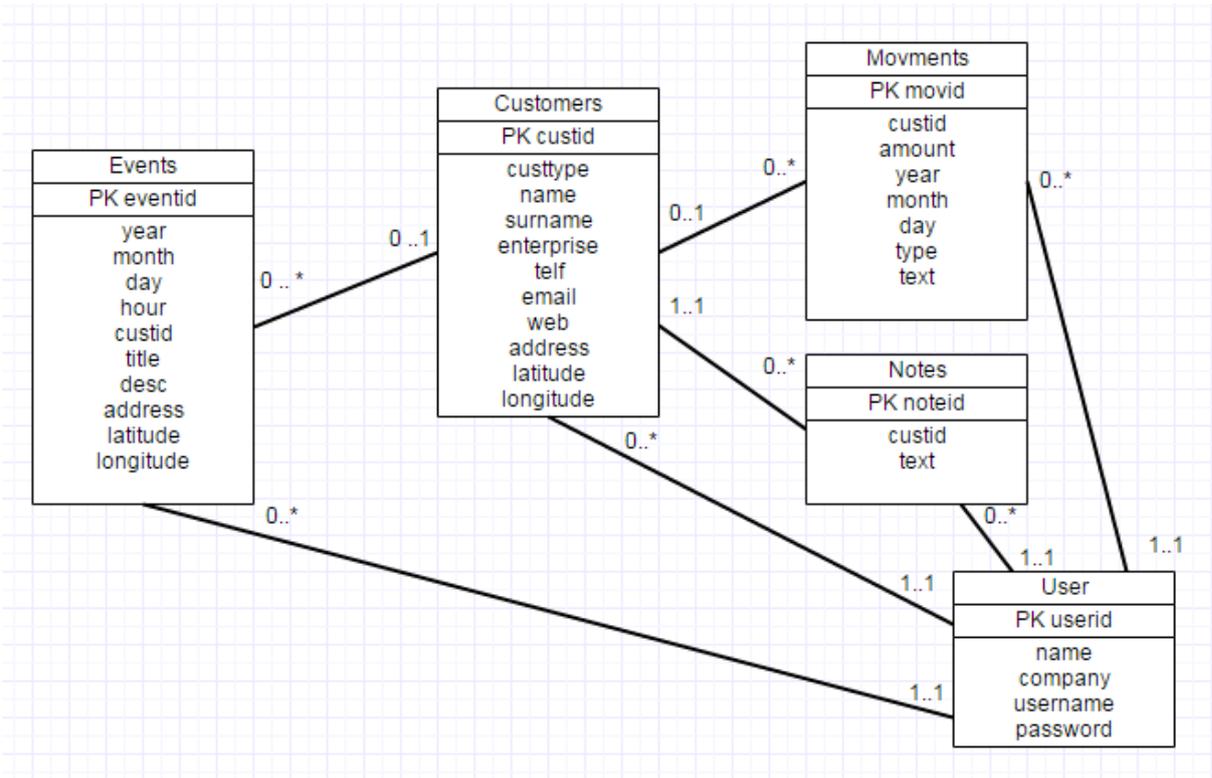


Figura 24 Modelo de datos empleado en el servidor

El modelo de datos empleado en el servidor se diferencia en que se añade la entidad User, que simboliza el usuario de la aplicación dueño de los datos almacenados que se relacionan con él. Este modelo permitirá poder almacenar todas las copias de seguridad de los usuarios en la misma base de datos y poder recuperar los datos con la certeza de que se recuperan los datos del usuario que los solicite, y no los de otros usuarios.

## **5 Implementación**

En este capítulo se va a hablar de los aspectos relevantes del proceso de implementación de la aplicación y del servidor. Primero se comentarán todas las tecnologías necesarias para la ejecución de este proyecto, para después hacer hincapié en los aspectos más interesantes que han ido surgiendo durante el proceso de implementación.

### **5.1 Contexto tecnológico**

A continuación describiré las distintas tecnologías, entornos de desarrollo y lenguajes que han sido necesarios para el desarrollo de la aplicación y del servidor que componen los objetivos de este proyecto.

## 5.1.1 Entorno de desarrollo

Android Studio [11] es el entorno de desarrollo o IDE (Integrated Development Environment) que se usa actualmente para desarrollar para el sistema operativo Android. Está basado en el IDE IntelliJ [12]. El desarrollo se ha realizado con la versión estable 1.5.1.

Este IDE ha sido utilizado tanto para el desarrollo de la aplicación como el desarrollo de los servicios del servidor.

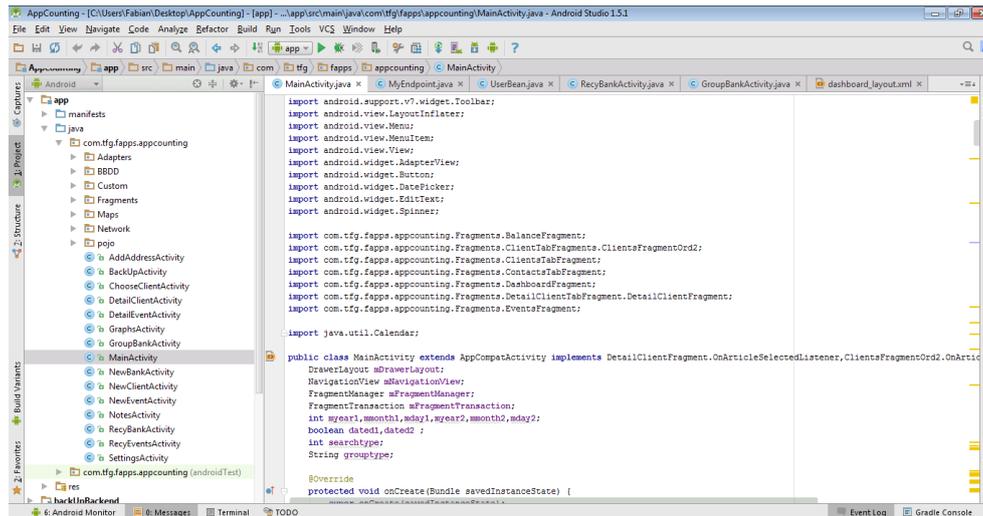


Figura 25 Android Studio

Hay que destacar de este entorno de desarrollo que soporta la construcción de aplicaciones basada en gradle. Gradle [13] una herramienta de código que permite generar las dependencias del proyecto con un lenguaje basado en *Groovy* [14] en lugar de con XML. Está basado en Apache ANT [15] y Apache Maven [16].

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.3"

    defaultConfig {
        applicationId "com.tfg.fapps.appcounting"
        minSdkVersion 17
        targetSdkVersion 23
        versionCode 2
        versionName "1.1"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile project(path: ':backUpBackend', configuration: 'android-endpoints')
    compile 'com.android.support:appcompat-v7:23.2.1'
    compile 'com.android.support:design:23.2.1'
    compile 'com.android.support:support-v4:23.2.1'
    compile 'com.android.support:cardview-v7:23.2.1'
    compile 'com.android.support:recyclerview-v7:23.2.1'
    compile 'com.google.code.gson:gson:2.4'
    compile 'com.google.maps.android:android-maps-utils:0.4'
    compile 'com.google.android.gms:play-services-maps:8.4.0'
    compile 'com.google.android.gms:play-services-location:9.8.0'
```

Figura 26 Ejemplo Gradle

### **5.1.2 Java**

Lenguaje de programación de propósito general, orientado a objetos y concurrente. Las aplicaciones de Java son compiladas a bytecode, que puede ser ejecutado desde cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora en la que se ejecute. En la actualidad, Java [17] es propiedad de Oracle [18].

### **5.1.3 Android**

Sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para *smartphones* y *tablets* y actualmente se está impulsando su desarrollo para relojes inteligentes, televisores y coches. Android es propiedad de la empresa Google desde 2005 [19].

Android se puede describir como una plataforma Software que abstrae el Hardware y facilita el desarrollo de aplicaciones para dispositivos con recursos limitados.

Por otra parte, en lo que se refiere a la estructura interna de Android, la ejecución de los procesos en Android se realiza utilizando una máquina virtual propia. Hasta la versión 5.0 de Android se utilizaba la máquina virtual Dalvik [20], desarrollada por Google y similar a la máquina virtual java, pero optimizada para reducir el consumo de energía. A partir de la versión 5.0 de Android se utiliza el entorno de ejecución Android (ART) [21]. La diferencia principal con la anterior máquina virtual utilizada para ejecutar los procesos en Android es la introducción de la compilación Ahead-of-Time (AOT). Esta compilación crea un archivo de compilación posterior a la instalación de la aplicación, que se utiliza al abrir la aplicación, evitando que la aplicación se compile una y otra vez, esto reduce el uso del procesador y aumenta la vida de la batería.

### **5.1.4 XML**

Es un lenguaje de marcas extensible (eXtensible Markup Language) utilizado para almacenar datos de forma legible por las personas. En Android se utiliza para definir las interfaces gráficas de usuario. La creación de los archivos XML [22] con las interfaces se puede hacer de forma automática gracias a un editor gráfico en Android Studio.

## 5.1.5 SQLite

SQLite [23] es un Sistema Gestor de Bases de Datos relacional muy popular con soporte para transacciones con Atomicidad, Consistencia, Aislamiento y Durabilidad, esta característica es más conocida por sus siglas en inglés ACID. Destaca por su pequeño tamaño y que no se basa en la arquitectura cliente-servidor, además de ser de código libre.

Es el motor de bases de datos que incorpora Android, y para su utilización es necesario sobrescribir una clase Java que herede de SQLiteOpenHelper.

```
public class MySQLiteOpenHelper extends SQLiteOpenHelper {

    private static MySQLiteOpenHelper instance;

    public synchronized static MySQLiteOpenHelper getInstance(Context context) {
        if (instance == null) {
            instance = new MySQLiteOpenHelper(context.getApplicationContext());
        }
        return instance;
    }

    public MySQLiteOpenHelper(Context context) { super(context, "database_file", null, 1); }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE Clients " +
            "(custid INTEGER PRIMARY KEY AUTOINCREMENT,iscust INTEGER NOT NULL ,name TEXT NOT NULL," +
            " surname TEXT , enterprise TEXT , address TEXT , latitude REAL, longitude REAL, telf INTEGER, email TEXT, web TEXT, las

        db.execSQL("CREATE TABLE Notes " +
            "(noteid INTEGER PRIMARY KEY AUTOINCREMENT,custid INTEGER NOT NULL ,texto TEXT NOT NULL," +
            " FOREIGN KEY (custid) REFERENCES Clients (custid) );");

        db.execSQL("CREATE TABLE ProfitLoss " +
            "(id INTEGER PRIMARY KEY AUTOINCREMENT,custid INTEGER ,amount REAL NOT NULL, year INTEGER NOT NULL,month INTEGER NOT NUL
```

Figura 27 Ejemplo de uso MySQLiteOpenHelper

## 5.1.6 App Engine

Google App Engine [24] es un *Platform as a service* (PaaS) que ofrece Google, una plataforma que permite a los desarrolladores gestionar, ejecutar y desarrollar aplicaciones sin la complejidad de construir y mantener la infraestructura sobre la que ejecutar sus aplicaciones.

Es un servicio escalable y móvil, muy atractivo para los desarrolladores porque es gratuito de usar hasta ciertos límites relacionados con la cantidad de peticiones que soportan los servidores, aunque si alcanzas esos máximos no supone un problema pagar, porque es probable que la aplicación que use esos servidores este proporcionando ingresos.

Actualmente las aplicaciones Google App Engine se pueden implementan mediante los lenguajes de programación Python, Java, Go y php.

### 5.1.7 Cloud Endpoints

*Google Cloud Endpoints* [25] consiste de herramientas, bibliotecas y capacidades que le permite generar APIs y librerías de clientes de una aplicación desplegada en *Google App Engine*. Los *Endpoints* hacen más fácil crear servidores para clientes web y clientes móviles como Android o iOS de Apple.

Es imposible crear bibliotecas para clientes móviles de un servidor Google App Engine. Sin embargo, usar los Endpoints hace el proceso más fácil porque te libera de tener que escribir código para manejar las comunicaciones con App Engine. Las bibliotecas de clientes generadas por los Endpoints te permiten tener solo que preocuparte de escribir las llamadas a la API.

Como se puede observar en la Figura 28 el servidor con App Engine ofrece sus funciones a clientes web Java Script, Android e iOS. Estas funciones están disponibles para los clientes a través de los Endpoints, que están expuestos con una API que los clientes pueden llamar.

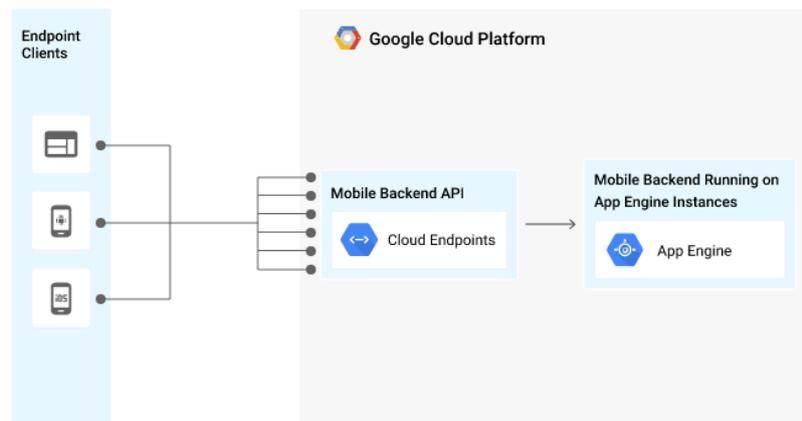


Figura 28 Estructura de la plataforma en la nube de Google

### 5.1.8 Objectify

Objectify [26] es una API (Application Programming Interface) en Java diseñada específicamente para acceder al almacén de datos de Google App Engine. Está diseñado para ser fácil de usar y rápido de aprender.

Como se puede observar en la Figura 29, el modelo de datos que se almacena en el almacén de datos NoSQL se puede gestionar con simples anotaciones en las clases Java. Con una notación simple es posible crear nuevas entidades, recuperarlas, modificarlas y eliminarlas.

```
@Entity
class Car {
    @Id String vin; // Can be Long, long, or String
    String color;
}

ofy().save().entity(new Car("123123", "red")).now();
Car c = ofy().load().type(Car.class).id("123123").now();
ofy().delete().entity(c);
```

*Figura 29 Ejemplo de código de Objectify*

## 5.2 Desarrollo Android

La parte de implementación de la aplicación Android es la parte que más carga de trabajo ha requerido a este trabajo de fin de grado. Ya contaba con conocimientos previos adquiridos durante el segundo cuatrimestre de este año. Pero esta ha sido la aplicación de mayor envergadura que he desarrollado. Aun así gracias al esfuerzo y dedicación realizado se culminó el desarrollo en un mes.

Además del esfuerzo y dedicación también fue crucial para conseguir ese tiempo de desarrollo el correcto uso de la API de Android. Me refiero a las estructuras y servicios que tienes a tu disposición cuando desarrollas para este sistema operativo.

A continuación comentaré los aspectos relevantes de los distintos elementos que forman parte del *framework* de Android y que han sido usados en el proyecto. Cabe destacar que gracias a algunos de estos elementos he podido reducir el código a implementar y llevar a cabo reutilización de código.

### 5.2.1 Activities

Una *Activity* [27] es un componente de las aplicaciones Android que proporciona una pantalla con la que el usuario puede interactuar para hacer algo, como puede ser marcar un número, hacer una foto, enviar un email o ver un mapa. La interfaz que proporciona una *Activity* se debe implementar en un fichero XML.

Las *Activities* tienen un ciclo de vida, que consiste de una serie de métodos, que el desarrollador puede sobrescribir, que representan los diferentes estados por los que pasa la *Activity* cuando cambia de visible a invisible para el usuario o viceversa.

La implementación más básica de una *Activity* consiste en sobrescribir el método *onCreate* donde se inicializan y añaden funcionalidad a todos los componentes de la interfaz de la correspondiente pantalla.

En lo que se refiere a las *Activities* se ha reducido el tiempo de desarrollo en la medida de lo posible utilizando *Intents*, otro componente de Android explicado posteriormente.

### 5.2.2 Fragments

Un *Fragment* [28] representa el comportamiento de una porción de la interfaz de usuario de una *Activity*. Se pueden combinar múltiples *Fragments* en una sola *Activity* para hacer una interfaz de usuario multi-panel o reutilizar un *Fragment* en muchas *Activities*.

Es importante mencionar que un *Fragment* se debe incrustar en una *Activity*, por lo que el ciclo de vida del *Fragment* se ve afectado por el ciclo de vida de su *Activity* padre. Las opciones principales de navegación que tiene la aplicación se han implementado utilizando esta técnica.

Hay una clase *Activity* principal que contiene los *Fragments* de pantallas como la de listar contactos, o listar movimientos, o la pantalla principal o resumen.

### 5.2.3 Layouts

Un *Layout* [29] define la estructura visual de una interfaz de usuario. Es posible declarar las interfaces de dos formas distintas, declarar la interfaz en un fichero XML o crear los elementos de la interfaz en tiempo de ejecución.

Para el desarrollo de la aplicación se han utilizado los ficheros XML, porque es una forma de tener separada la capa de presentación de la lógica de la aplicación. Para enlazar cada fichero a su clase Java con su respectiva lógica de negocio se utiliza el identificador del fichero que se le pasa a la *Activity* o *Fragment* en el método *onCreate* y *onCreateView*, respectivamente.

### 5.2.4 Navigation drawer

*Navigation drawer* [30] es un concepto introducido en Android con Material Design. Material Design [31] son una serie de elementos de interfaz de usuario y guías de diseño para intentar estandarizar el proceso de desarrollo de interfaces de usuario en las versiones más recientes de Android.

Un *Navigation drawer* consiste de un panel desplegable en un margen de la pantalla donde se tiene acceso a las opciones principales de navegación de una aplicación.

La estructura de las opciones principales de navegación de la aplicación se ha implementado con *Fragments* por imposibilidad de otras opciones más usables.

En los tutoriales que proporciona Google para implementar un *Navigation drawer* se dispone de una vista principal sobre la que se despliega el panel lateral, y si se pulsa un elemento del panel desplegable se inserta un *Fragment* sobre la vista principal de la aplicación.

### 5.2.5 Intents

Los *Intents* [32] son mensajes asíncronos que se utilizan para solicitar acciones de otros componentes en Android. Estas acciones solicitadas pueden ser por ejemplo lanzar una actividad o lanzar una notificación. En el caso de lanzar una actividad estos mensajes se pueden utilizar para enviar y recoger entre *Activities*, añadiendo

información extra en el *Intent* que puede ser obtenida en la actividad que recibe el *Intent*. Existen dos tipos de *Intents*, explícitos e implícitos. En mi aplicación he usado ambos.

Los *Intents* explícitos llevan indicado exactamente la clase que debe ser lanzada, y han sido usados por ejemplo, para abrir un formulario con los datos de un cliente determinado, indicando el identificador del cliente que se deseaba abrir en el *Intent*. Esta técnica me ha permitido reutilizar *Activities* como las de los formularios de nuevo contacto o cliente, nuevo evento y nuevo movimiento. Si se desea crear un nuevo evento simplemente se lanza una nueva *Activity* con un *Intent* explícito en el que no se añade información extra. Y si se desea modificar o visualizar los detalles de un evento el procedimiento es el mismo, pero en el *Intent* se añade como información extra el identificador del evento en cuestión.

Los *Intents* implícitos no indican la clase o el componente que debe ser lanzado, por lo que incluyen información para que el sistema determine el componente más adecuado. Estos *Intents* han sido utilizados en el caso de que, entre otros, el usuario desee enviar un correo electrónico al email de un determinado cliente. Cuando se pulsa sobre el correo electrónico el sistema le solicita al usuario que elija una aplicación con la que realizar la acción.

```
if(data.get(position).getEmail()!=null && !data.get(position).getEmail().equals("")){
    holder.tv_email.setText(data.get(position).getEmail());
    holder.tv_email.setOnClickListener((v) -> {
        Intent emailIntent = new Intent(Intent.ACTION_SENDTO, Uri.fromParts(
            "mailto", data.get(position).getEmail(), null));
        context.startActivity(emailIntent);
    });
}
```

Figura 30 Ejemplo Intent implícito

### 5.2.6 Adapter

Un *Adapter* [33] es un objeto del *framework* de Android que actúa como un puente entre un elemento de la interfaz de usuario y un conjunto de datos. Son de gran utilidad para crear listas, y para eso han sido utilizados en todas las listas que tiene mi aplicación. Android proporciona una serie de *Adapters* por defecto, pero según el contenido que desees que forme cada elemento de la lista puedes crear tus *Adapters* personalizados. Se han creado variedad de *Adapters* personalizados para todos los tipos de lista diferentes que se pueden encontrar en la aplicación.

La implementación de *Adapters* ha supuesto la mayor reutilización de código en todo el desarrollo de la aplicación.

### 5.2.7 Recyclerview y Cardview

Estos son dos objetos para la interfaz de usuario introducidos junto con Material Design, anteriormente mencionado. Combinar el uso de *Recyclerview* con *Cardview* y *Adapter* permite crear listas de una forma relativamente sencilla, con unos resultados muy eficientes, en lo que a carga del dispositivo móvil se refiere, y con una apariencia que sigue las pautas mencionadas en la especificación de Material Design.

Un *Recyclerview* [34] es un objeto de interfaz de usuario que introduce una capa de abstracción por debajo de un *Adapter*. De forma resumida su funcionamiento se basa en llevar un control entre los elementos de la lista que caben en la pantalla y las posiciones de los elementos mostrados en el conjunto de datos que forma dicha lista. La principal novedad que ofrece es que fuerza al desarrollador a utilizar el patrón de diseño ViewHolder [35], para mantener una referencia a los elementos gráficos que componen la lista, con lo que se aligera el proceso de obtener dichas referencias.

*Cardview* [36] es otro elemento de interfaz de usuario introducido con Material Design que puede ser personalizado con facilidad y muy útil para utilizar como elemento de las listas, que ofrece una apariencia más elegante con menos esfuerzo.

Al igual que los *Adapters*, los *Recyclerview* y *Cardviews* han sido utilizados en todas las listas implementadas en la aplicación.

### 5.2.8 AsyncTask

Esta clase predefinida en Android permite realizar operaciones en segundo plano o en otro hilo de ejecución para evitar ralentizar la experiencia de usuario en el hilo de ejecución principal conocido como *UI thread*. Es la implementación recomendada para realizar operaciones de red, como puede ser acceder a un servicio web.

La clase *AsyncTask* [37] dispone de diferentes métodos para sobrescribir, dicho de forma resumida: dispone de un método (*onPreExecute*) que se ejecuta en el hilo principal antes de comenzar la ejecución en segundo plano, dispone de otro método (*doInBackground*) que corresponde al código que se ejecuta en otro hilo de ejecución debido a que las operaciones que se realicen aquí son las que pueden ralentizar la experiencia de usuario, y dispone de otro método (*onPostExecute*) que se ejecuta en el hilo principal al terminar la ejecución en segundo plano. En este último método normalmente se realizan operaciones de procesado de los datos obtenidos tras la ejecución del hilo en segundo plano.

Su utilización es relativamente sencilla, es necesario que la clase Java extienda de la clase *AsyncTask* y sobrescribir como mínimo el método *doInBackground*.

Esta técnica ha sido utilizada en mi aplicación para las operaciones de red de las copias de seguridad.

### 5.3 Desarrollo e implementación del servidor

Cuando comencé a realizar este trabajo no tenía conocimientos anteriores de desarrollo de servicios web y tampoco de servicios en la nube. Pero mi tutor me recomendó usar los servicios en la nube de Google, me decía que era una forma muy sencilla y rápida de crear y desplegar mis servicios web para mis aplicaciones Android. Una decisión acertada debido al potencial de los conocimientos adquiridos para mis próximos proyectos y mi currículum, y a que realmente me resultó muy sencillo y gratificante escribir el código del servidor.

Simplemente hubo unas complicaciones al comienzo para configurar el entorno de desarrollo, debido seguramente a mi inexperiencia, aun así hacer este trabajo también supone aprender cosas nuevas.

El desarrollo se realizó sin complicaciones y en unos plazos muy cortos. Consta de la definición de las clases Java con las distintas entidades que componen mi modelo de datos y del desarrollo de la lógica de las diferentes llamadas que tiene la API generada.

Me resultó sencillo gracias a cosas como la que se puede apreciar en la Figura 31 y Figura 32.

En la Figura 31 se puede apreciar que los usuarios de mi aplicación se identifican con el *uid* generado automáticamente. Para el registro de usuarios, por ejemplo, gracias a la etiqueta *@Index* puedo comprobar con facilidad en la lógica de las llamadas a la API si el nombre de usuario introducido existe en la base de datos o puedo comprobar con facilidad si la contraseña introducida es la del usuario introducido.

```
@Entity
public class UserBean {

    @Id
    Long uid;
    @Index
    String username;
    @Index
    String pwd;
    String name, company;

    List<ClientBean> clients;
    List<EventBean> events;
    List<MovBean> movs;
    List<NoteBean> notes;

    public UserBean() {
        clients = new ArrayList<ClientBean>();
        events = new ArrayList<EventBean>();
        movs = new ArrayList<MovBean>();
        notes = new ArrayList<NoteBean>();
    }
}
```

Figura 31 Ejemplo código entidad User

En la Figura 32 se destaca el poco código que requiere obtener todo el contenido almacenado por un usuario gracias a la estructura dada a la clase *UserBean*.

```
@ApiMethod(name = "userBackUp", httpMethod = "post")
public UserBean userBackUp(@Named("uid") Long userId) {
    return ofy().load().type(UserBean.class).id(userId).now();
}
```

Figura 32 Ejemplo código obtener copia de seguridad

Y para finalizar, las llamadas desde la aplicación a la API se implementan de una forma muy sencilla también (ver Figura 33). Gracias al objeto *myApiService* hacer la llamada al servicio es tan sencillo como hacer una llamada a un método de clase.

```
private static MyApi myApiService = null;
private Context context;
@Override
protected UserBean doInBackground(Pair<Context,UserBean>... params) {

    if(myApiService == null) {
        MyApi.Builder builder = new MyApi.Builder(AndroidHttp.newCompatibleTransport(),
            new AndroidJsonFactory(), null)
            .setRootUrl("https://appcounting.appspot.com/_ah/api/");

        myApiService = builder.build();
    }

    context = params[0].first;
    UserBean user = params[0].second;

    try {
        return myApiService.userBackUp(user.getId()).execute();
    } catch (IOException e) {
        Log.d("BCK:", e.getMessage());
        return null;
    }
}
```

Figura 33 Ejemplo código llamada a la API desde la aplicación

## 6 Resultados obtenidos

En este capítulo se va a mostrar la aplicación final desarrollada. Se ha intentado conseguir una interfaz lo más similar posible a la de los prototipos mostrados en capítulos anteriores. Hay subapartados en los que no se muestran todas las capturas relacionadas porque esas pantallas se reutilizan y han sido ya mostradas en subapartados anteriores.

### 6.1 Pantalla principal

En la Figura 34 se muestra el resultado final de la pantalla de resumen, que contiene una lista con los eventos del día actual, un mapa con la expansión del negocio, según las direcciones que haya introducidas para los clientes y contactos, y un balance de la economía del mes actual. La barra azul que se puede observar muestra como son los ingresos del mes actual según la media de ingresos que haya introducidos en la aplicación a lo largo del tiempo y la segunda barra indica en porcentaje lo que suponen los gastos del mes actual respecto a los ingresos del mes actual.

En la Figura 35 se muestra el dialogo para añadir información del que se habló en el capítulo 4. Este dialogo ha sido reutilizado en las pantallas de listar clientes, listar contactos, listar eventos y listar movimientos.



Figura 34 Aplicación final, Pantalla resumen

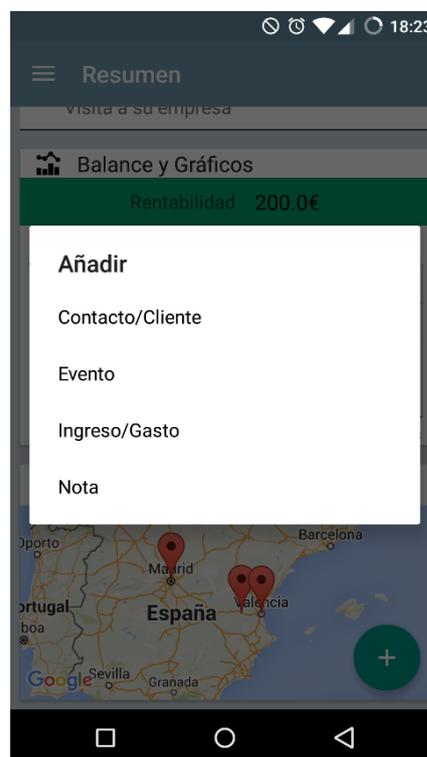
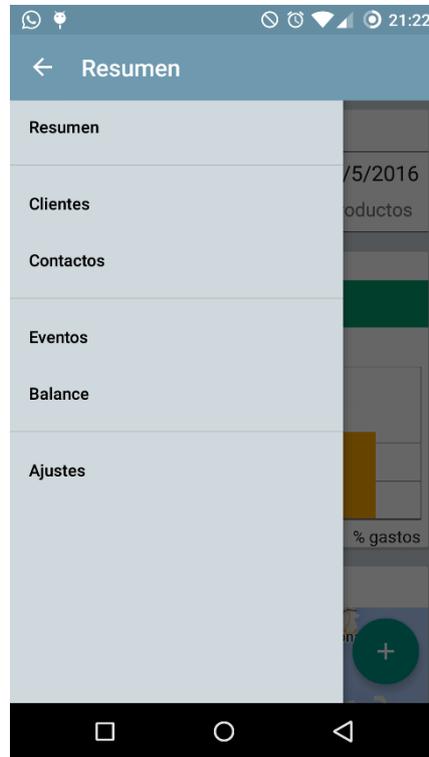


Figura 35 Aplicación final, Dialogo añadir información

En la Figura 36 se puede ver el panel desplegable desde el lateral izquierdo de la pantalla del que se habló en el capítulo 4. Es posible desplegar este panel desde las pantallas de resumen, listar clientes, listar contactos, listar eventos y listar movimientos.



*Figura 36 Aplicación final, Panel lateral desplegable*

## 6.2 Nuevo cliente o contacto

En la Figura 37 se muestra el formulario para crear un nuevo cliente o contacto. En el interruptor que hay en la parte superior derecha se puede cambiar de contacto a cliente y viceversa. Si se pulsa el botón dirección se abrirá la pantalla que se muestra en la Figura 38. La información introducida se guarda pulsando el icono del *diskette* de la parte superior derecha de la **¡Error! No se encuentra el origen de la referencia..**



Figura 37 Aplicación final, Nuevo cliente/contacto



Figura 38 Aplicación final, Añadir dirección

### 6.3 Nuevo evento

En la Figura 39 se muestra la pantalla para introducir un nuevo evento. El procedimiento para guardar información es igual que para el resto de formularios. Si se pulsa sobre la fecha que está subrayada se abre el dialogo de la Figura 40. Y si primero se pulsa sobre el interruptor de la hora aparecerá un texto con la hora subrayada. Si se pulsa sobre él se abrirá el dialogo de la Figura 41.

Si se pulsa el botón de “Asignar a” se abrirá la pantalla de la Figura 42. La lista que muestra esta pantalla contiene tanto clientes como contactos y están ordenados por más recientemente consultados, asignados o introducidos. También es posible en esta pantalla (ver Figura 42) buscar por nombre y apellidos en la lista.

Y por último si se pulsa sobre el botón “dirección” se reutiliza la pantalla de la Figura 38.



Figura 39 Aplicación final, Nuevo evento



Figura 40 Aplicación final, Introducir fecha

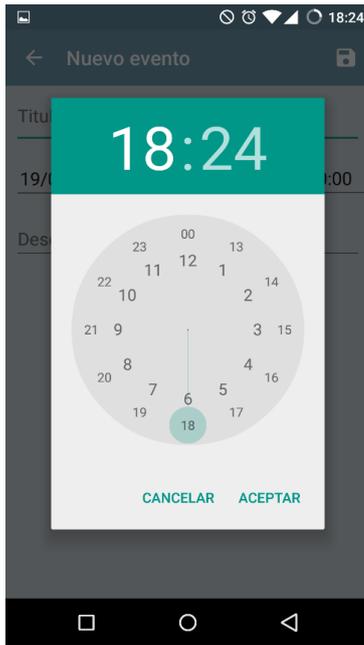


Figura 41 Aplicación final, Introducir hora

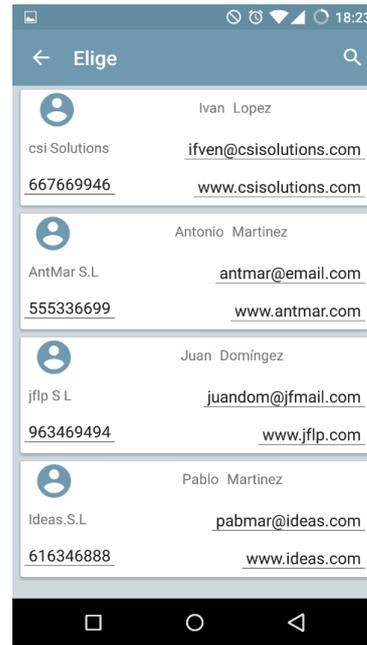


Figura 42 Aplicación final, Seleccionar cliente o contacto

## 6.4 Nuevo movimiento

En la Figura 43 se muestra la pantalla para introducir un nuevo movimiento, ya sea gasto o dinero. El procedimiento para guardar la información introducida es el mismo que en el resto de formularios, y el procedimiento para elegir entre ingreso y gasto es similar al de elegir entre cliente o contacto. Si se pulsa el botón de “Asignar a” se reutiliza la pantalla de la Figura 42.

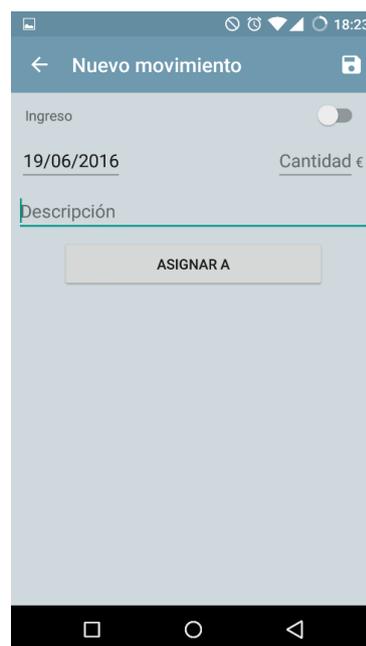


Figura 43 Aplicación final, Nuevo movimiento

## 6.5 Listar clientes y contactos

En este subapartado se muestran las pantallas de listar clientes y listar contactos (ver Figura 44 y Figura 45, respectivamente). Como ya se comentó en la especificación, son dos pantallas muy similares, siendo el contenido de las listas lo único que las diferencia. Si se selecciona un elemento de dichas listas, se abrirá la pantalla de editar cliente o contacto con la información del cliente o contacto seleccionado.



Figura 44 Aplicación final, Listar clientes

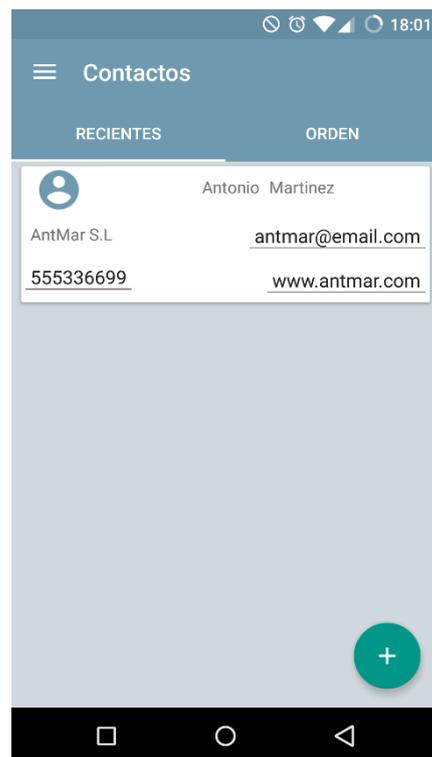


Figura 45 Aplicación final, Listar contactos

## 6.6 Listar eventos

En la Figura 46 se muestra la pantalla para explorar el calendario. La lista que hay en la parte inferior de la pantalla muestra los eventos correspondientes al día seleccionado en el calendario. Si se selecciona un evento de dicha lista se abrirá la pantalla de editar evento con la información del evento seleccionado.



Figura 46 Aplicación final, Listar eventos

## 6.7 Listar movimientos

En la Figura 47 se muestra la pantalla de listar movimientos, en la que se disponen de tres pestañas para visualizar todos los movimientos, solo los ingresos o solo los gastos.

En esta pantalla se disponen de dos opciones de menú, buscar por fecha (ver Figura 49) y agrupar movimientos (ver Figura 50) **Error! No se encuentra el origen de la referencia.** y Figura 51).

Para buscar por fecha se puede rellenar solo el primer campo de fecha y se busca solo para ese día, o se pueden rellenar los dos y la búsqueda se realiza en un rango de fechas. Cuando se pulsa sobre buscar en el dialogo de buscar por fecha se abre la pantalla de la Figura 48, que muestra una lista con los movimientos ordenados cronológicamente.

La opción de agrupar movimientos permite ver de forma resumida la economía de la empresa a lo largo del tiempo, organizado en franjas de tiempo que pueden ser meses o años. Cuando se pulsa uno de los dos botones disponibles en el diálogo de agrupar movimientos se abre la pantalla de la Figura 52, que muestra el resumen de la economía. Cada elemento de la lista muestra el periodo de tiempo al que hace referencia, el balance (ingresos menos gastos) de ese periodo, los ingresos de ese periodo y los gastos de ese periodo. Si se pulsa sobre un elemento de esta lista se abre la pantalla de la Figura 53, que simplemente es una lista con los movimientos que componen el periodo seleccionado.



Figura 47 Aplicación final, Listar movimientos



Figura 48 Aplicación final, Pantalla buscar movimientos por fecha



Figura 49 Aplicación final, Dialogo buscar movimientos por fecha

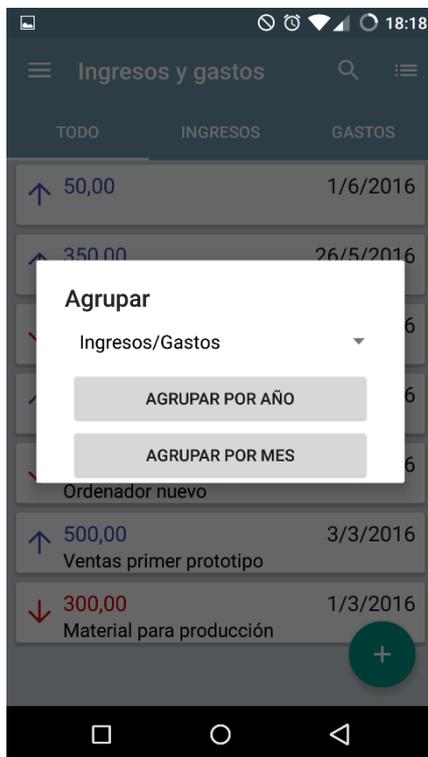


Figura 50 Aplicación final, Dialogo agrupar movimientos

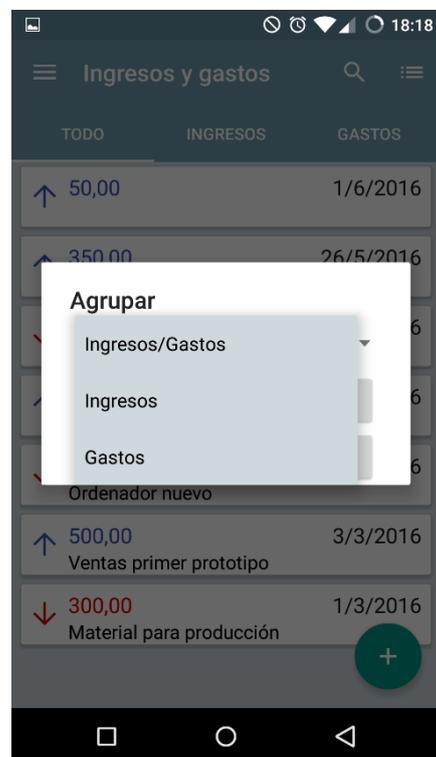


Figura 51 Aplicación final, Dialogo agrupar movimientos desplegable



Figura 52 Aplicación final, Pantalla agrupar movimientos



Figura 53 Aplicación final, Detalle pantalla agrupar movimientos

## 6.8 Informes y gráficos

En la Figura 54 se muestra la pantalla de gráficos, a la que se puede acceder pulsando sobre los gráficos en la pantalla resumen. Esta pantalla de gráficos contiene solo tres gráficos.

El primero es un gráfico lineal que muestra el histórico de los movimientos, con el que se puede apreciar cuándo ha habido ingresos y gastos.

El segundo y tercer gráficos son gráficos de barras para mostrar los ingresos y los gastos por meses.



Figura 54 Aplicación final, Pantalla Gráficos

## 6.9 Ajustes

En la Figura 55 se muestra la pantalla de ajustes. Esta pantalla es utilizada para que el usuario se registre en el servidor si desea realizar copias de seguridad y para acceder a las opciones de copia de seguridad. Las opciones de copia de seguridad se pueden observar en la Figura 56. Cuando se selecciona una de estas opciones comienza un proceso para enviar la información de la aplicación al servidor o para recibirla, según la opción seleccionada. Cuando el proceso termina se notifica al usuario (ver Figura 57 y Figura 58).

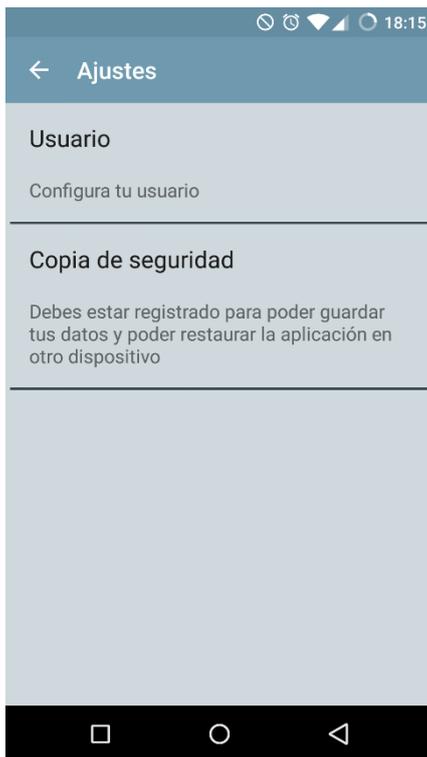


Figura 55 Aplicación final, Pantalla Ajustes

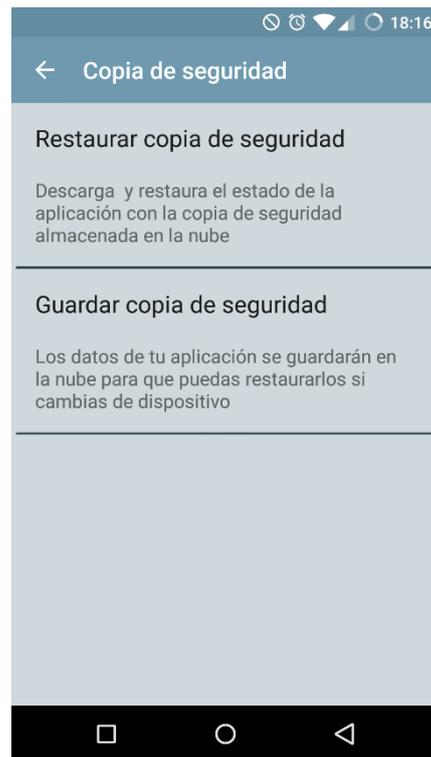


Figura 56 Aplicación final, Pantalla copia de seguridad



Figura 57 Aplicación final, Notificación Guardar copia de seguridad

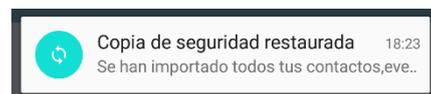


Figura 58 Aplicación final, Notificación Restaurar copia de seguridad

## 6.10 Detalles contacto y detalles cliente

En este subapartado se muestra la pantalla desarrollada para mostrar los detalles de un contacto o un cliente. Se puede acceder a esta pantalla seleccionando un elemento en la lista de clientes o lista de contactos. Esta pantalla se compone de cuatro pestañas.

En la Figura 59 se pueden ver los detalles del cliente o contacto seleccionado, como su nombre, email, teléfono, etc. Esta pestaña dispone de una opción de menú que permite editar los detalles de este cliente o contacto.

En la Figura 60 se pueden ver las notas relacionadas con el contacto o cliente, además de poder añadir notas nuevas.

En la Figura 61 se pueden ver los eventos relacionados con el cliente o contacto, además de poder añadir un evento a este cliente o contacto pulsando el botón circular situado en la parte inferior derecha de la pantalla.

En la Figura 62 se pueden ver los movimientos de dinero relacionados con el cliente, tanto ingresos como gastos, además de poder añadir un nuevo movimiento a este cliente o contacto pulsando el botón circular situado en la parte inferior derecha de la pantalla.

En las pestañas de eventos y movimientos hay una opción de menú para buscar por fecha tanto eventos como movimientos. En la búsqueda de movimientos, tanto el diálogo como la pantalla que componen esta funcionalidad han sido reutilizados (ver Figura 48 y Figura 49). Para la búsqueda de eventos se ha creado un diálogo muy similar al de la Figura 49 y se ha reutilizado la pantalla de la Figura 48.

Es importante mencionar que en la especificación de este proyecto no se contemplaba el uso del botón de añadir información en los detalles del cliente, pero finalmente se ha utilizado porque permite de forma muy sencilla añadir nueva información relacionada con un cliente o contacto.



Figura 59 Aplicación final, Pantalla detalles cliente



Figura 60 Aplicación final, Pantalla notas cliente

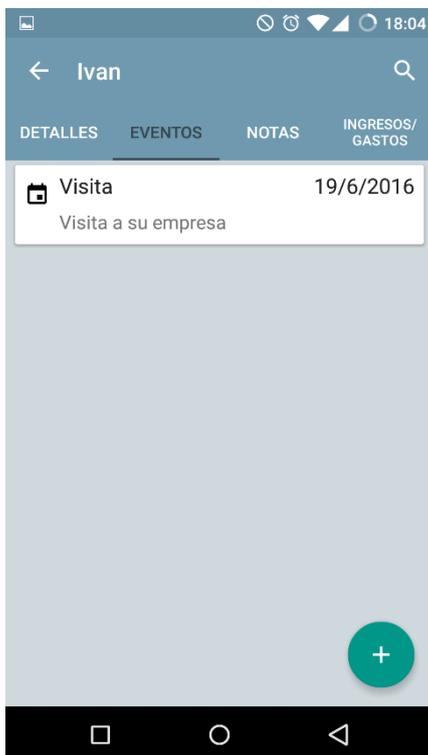


Figura 61 Aplicación final, Pantalla Eventos cliente

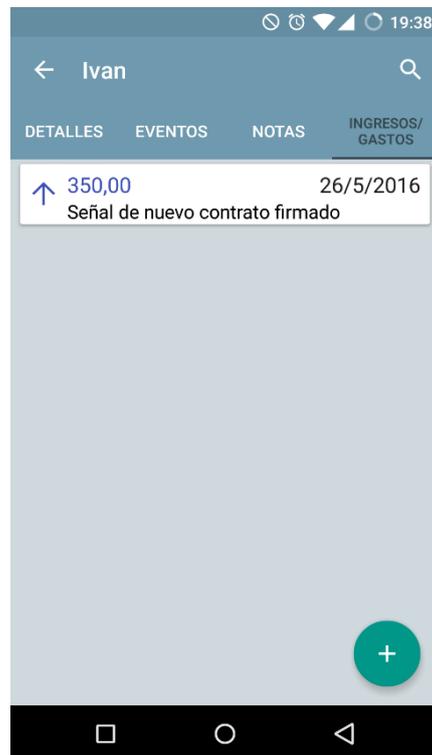


Figura 62 Aplicación final, Pantalla movimientos cliente

## 6.11 Añadir nota

En la Figura 63 se puede ver la pantalla para añadir notas a un cliente o contacto. A esta pantalla se puede acceder a través de la opción de añadir nota que se encuentra en el diálogo que abre el botón de añadir información (ver Figura 35). Cuando se selecciona la opción de añadir nota se abre la pantalla de la Figura 42 y cuando se selecciona un elemento de esta lista se abre la pantalla de la Figura 63.

Es importante mencionar que en la aplicación es posible añadir notas a un cliente desde dos pantallas diferentes, desde la de la Figura 63 y desde la pantalla de la Figura 60.

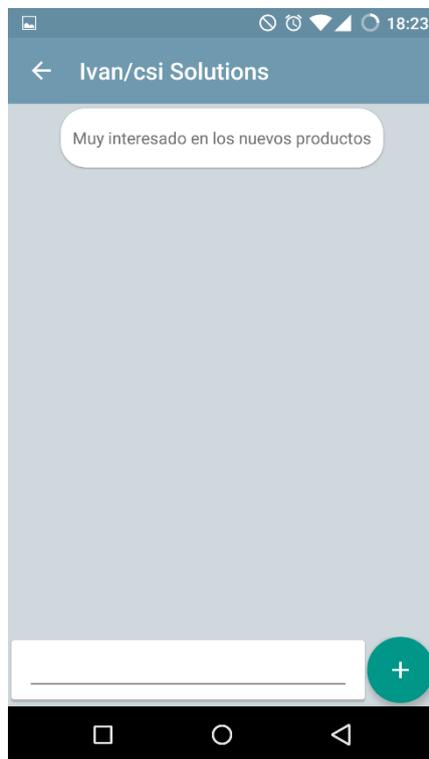


Figura 63 Aplicación final, Pantalla añadir nota

## 6.12 Editar evento

En la Figura 64 se muestra la pantalla para editar un evento, solo son modificables el título, la fecha, la hora y la descripción.

Esta pantalla dispone de dos opciones de menú, el icono del *diskette* para guardar los cambios, y el icono de la papelera para eliminar el evento. Al pulsar el icono de eliminar evento se abre un diálogo como el de la Figura 65 para confirmar si el usuario desea eliminarlo.



Figura 64 Aplicación final, Pantalla Editar evento

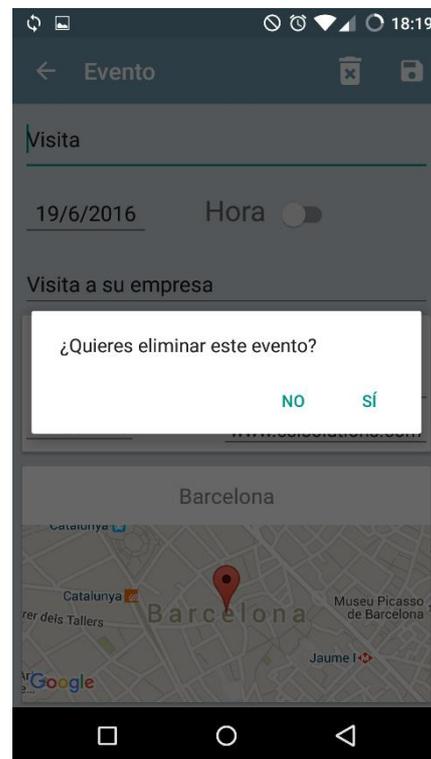


Figura 65 Aplicación final, Dialogo eliminar evento

## 6.13 Editar contacto y editar cliente

En la Figura 66 se muestra la pantalla para editar los detalles de un contacto o cliente, pantalla a la que se puede acceder desde una opción de menú en los detalles del contacto o cliente. En esta pantalla sí puede modificar toda la información.

Dispone de dos opciones de menú semejantes a las mostradas en editar un evento. Si se pulsa en la opción de menú de eliminar se mostrará un diálogo similar al de la Figura 65.

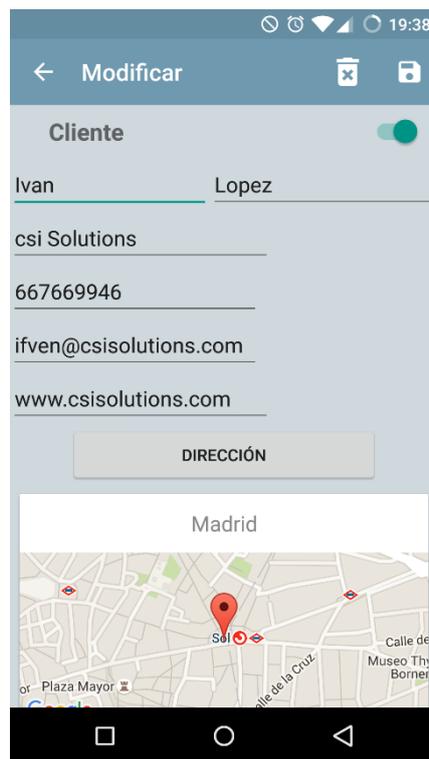


Figura 66 Aplicación final, Pantalla editar contacto o cliente

## 6.14 Editar movimiento

En la Figura 67 se muestra la pantalla de editar un movimiento de dinero, pantalla a la que se puede acceder seleccionando un elemento de la pantalla listar movimientos.



Figura 67 Aplicación final, Pantalla Editar movimiento

## 6.15 Test de la aplicación

Para el testeo de la aplicación se ha realizado un test de usabilidad realizado por mí y por unos voluntarios, a quienes se les ha proporcionado la aplicación para que la utilicen en su día a día.

Estos voluntarios son dos personas poseedoras de un pequeño negocio, negocios distintos, pero ambos gestionados por una sola persona.

La información recogida tras el uso de la aplicación es la siguiente:

- En la pantalla de la Figura 42 hay que revisar la funcionalidad de buscar por nombre y apellidos en la lista porque no se ha conseguido que funcione perfectamente.
- En la pantalla de la Figura 46 hay un problema de usabilidad que no se ha solucionado en el desarrollo. No se colorean los días que contienen eventos, lo que impide saber qué días del mes hay eventos de forma visual y rápida.
- Respecto a las notificaciones de los procesos de guardado y restauración de la copia de seguridad, estos procesos simplemente notifican al usuario cuando el proceso termina, y sería recomendable que mostrase el avance del intercambio de datos.
- En la pantalla de la Figura 64 el cliente relacionado y la dirección no se pueden cambiar, esto es un fallo del desarrollo que debe ser arreglado en el futuro.
- En la pantalla de la Figura 67 se puede modificar toda la información del movimiento excepto el cliente relacionado. Esto es un fallo del desarrollo que será solucionado en el futuro.

De la información recogida en los tests se toma la decisión de eliminar la funcionalidad de buscar de la Figura 42 para poder cumplir los plazos de publicación. Quedando pendiente para las siguientes versiones añadir esta funcionalidad corregida además de la corrección del resto errores encontrados

## 7 Conclusiones

Este trabajo ha relatado el desarrollo de un producto software orientado al mundo empresarial, más en concreto de las pequeñas empresas, con el que se busca ofrecer una solución innovadora, sencilla y efectiva sobre la variedad de soluciones software que ya existen para el mundo empresarial.

Este proyecto ha abarcado el desarrollo del software comenzando por el análisis de la viabilidad como negocio del desarrollo y mantenimiento del software, para después analizar el entorno de aplicaciones relacionadas con las empresas. Una vez realizadas estas tareas se realizó la especificación formal de los componentes de la aplicación además de tomar diversas decisiones de diseño. Con las decisiones de diseño y la especificación formal se realizó la implementación de la aplicación. Tras la implementación se probó el correcto funcionamiento de la aplicación

Finalmente la aplicación fue publicada en Google Play el 28 de mayo de 2016, pasado un mes no hay resultados significativos. La aplicación no tiene una cantidad significativa de descargas, por lo que hay que seguir esperando y realizar mejoras.

Respecto a mi plan de negocio, esta es una situación que se contemplaba, dado que no se ha realizado ninguna campaña promocional, ni posicionamiento SEO, por lo que es muy poco probable que la aplicación tuviese éxito inicialmente, y la solución que estaba planeada para este caso es lo comentado anteriormente, esperar y mejorar la aplicación. Los objetivos de negocio son más a largo plazo.

Respecto a los resultados obtenidos en este trabajo, en mi opinión los resultados obtenidos son positivos y beneficiosos para mí, he aprendido una forma de desarrollar aplicaciones web con mucha proyección de futuro y he afianzado mis conocimientos de Android, además de poder añadir la aplicación desarrollada a mi currículum.

Comparando la aplicación obtenida con los objetivos mínimos planteados puedo destacar que:

- El objetivo de permitir gestionar la información de clientes se ha cumplido de forma completa.
- El objetivo de poder mostrar gráficos e informes se ha cumplido, pero quizá no de la mejor forma. Puede que a los usuarios les parezca muy poco con las tres gráficas disponibles.
- El objetivo de poder mostrar los clientes localizados en el mapa se ha cumplido correctamente.

Respecto a los objetivos planteados de funcionalidades que la aplicación debería tener o podría tener se puede destacar que:

- La gestión del calendario con eventos y recordatorios es un objetivo que se ha cumplido, pero no completamente, no es posible programar recordatorios desde la aplicación.
- La creación y gestión de facturas es un objetivo que no se ha cumplido. En su lugar se ha desarrollado la gestión de movimientos de dinero, que permite tener

controlados los flujos de dinero en el negocio. Pero simplemente a título informativo para uno mismo.

- No se ha desarrollado la opción de poder introducir gastos a la aplicación a través de fotografías (Ejemplo: fotografía a un ticket).
- Como ya se planteaba al comienzo, la integración con redes sociales para expandir el negocio es un objetivo que no estaba en los planes para este proyecto.

Hay que reconocer que la interfaz de usuario aunque no tiene flujos demasiado complejos no se ha desarrollado teniendo en cuenta la heurística ni el fácil aprendizaje de uso de la aplicación. Pero son aspectos que se pueden ir mejorando con el tiempo.

Es destacable que las funcionalidades conseguidas se podrían contemplar como un pequeño módulo de CRM y un módulo de contabilidad. Lo que forman una buena base para construir poco a poco un sistema de información completo para dispositivos móviles. El resultado que se puede conseguir a largo plazo es muy interesante para llegar a otro nivel en lo que se refiere a los comienzos de una empresa.

## 7.2 Trabajos futuros

Para culminar este trabajo, teniendo en cuenta el estado actual de la aplicación me planteo los siguientes pasos que se deberían dar para continuar el trabajo realizado hasta ahora. A continuación los siguientes puntos que indico son tareas a tener en cuenta para continuar el desarrollo.

- Cambiar o añadir a la gestión de movimientos de dinero la gestión de facturas, con una interfaz amigable que haga rápido y cómodo la creación de facturas. Actualmente con la aplicación es posible llevar un balance propio para estar informado de la situación económica del negocio, pero es probable que los usuarios demanden una gestión de sus movimientos de dinero que controle facturas, pagos, tiempos de cobro, etc.
- Realizar copias de seguridad automáticas. Esta es una funcionalidad que sería posible de desarrollar con facilidad debido a que la API generada en el servidor ya tiene desarrolladas llamadas que permitirían, por ejemplo guardar un cliente en el servidor automáticamente tras guardarse en la aplicación. Solo faltaría hacer esas llamadas a la API desde la aplicación tras guardar dicho cliente en la aplicación.
- Crear una aplicación web con la que poder consultar la información de la aplicación y más funcionalidades. Esta es una idea tomada de las aplicaciones de la competencia analizadas al comienzo de este trabajo.
- Integración de la aplicación con Google Analytics [38] para monitorizar el uso de la aplicación por parte de los usuarios, conocer su perfil y así adaptar la aplicación mejor a las necesidades de los usuarios.

## Bibliografía

- [1] L. K. C y L. J. P, Sistemas de información, Pearson Custom Publishing, 2000.
- [2] Á. Gómez Vieites y A. Suárez Rey, Sistemas de Información Herramientas prácticas para la gestión empresarial.
- [3] I. G. Valcárcel, CRM: gestión de la relación con los clientes., FC Editorial, 2001.
- [4] E. y. T. Ministerio de Industria, «Retrato de las Pyme,» 2015. [En línea]. Available:  
[www.ipyme.org/Publicaciones/Retrato\\_PYME\\_DIRCE\\_1\\_enero\\_2015.pdf](http://www.ipyme.org/Publicaciones/Retrato_PYME_DIRCE_1_enero_2015.pdf).  
[Último acceso: 26 05 2016].
- [5] A. Osterwalder y Y. Pigneur, Generación de modelos de negocios, Deusto, 2011.
- [6] «Minimo producto viable,» [En línea]. Available:  
[https://en.wikipedia.org/wiki/Minimum\\_viable\\_product](https://en.wikipedia.org/wiki/Minimum_viable_product). [Último acceso: 20 06 2016].
- [7] «Google Play Store, SAP - SAP Business One,» [En línea]. Available:  
<https://play.google.com/store/apps/details?id=b1.mobile.android&hl=es>.
- [8] «Metodo MOSCOW,» [En línea]. Available:  
[https://en.wikipedia.org/wiki/MoSCoW\\_method](https://en.wikipedia.org/wiki/MoSCoW_method). [Último acceso: 28 05 2016].
- [9] C. Reynoso, «Schoolar google,» [En línea]. Available:  
<http://carlosreynoso.com.ar/archivos/arquitectura/Arquitectura-software.pdf>.  
[Último acceso: 20 06 2016].
- [10] «Ninjamock,» [En línea]. Available: [ninjamock.com](http://ninjamock.com).
- [11] «Android Studio,» [En línea]. Available:  
[https://es.wikipedia.org/wiki/Android\\_Studio](https://es.wikipedia.org/wiki/Android_Studio). [Último acceso: 18 06 2016].
- [12] IntelliJ, [En línea]. Available: [https://es.wikipedia.org/wiki/IntelliJ\\_IDEA](https://es.wikipedia.org/wiki/IntelliJ_IDEA).
- [13] «Gradle,» [En línea]. Available: <https://en.wikipedia.org/wiki/Gradle>.
- [14] «Groovy,» [En línea]. Available: <http://www.groovy-lang.org/>.
- [15] «Apache Ant,» [En línea]. Available: <http://ant.apache.org/>.
- [16] «Apache Maven,» [En línea]. Available: <https://maven.apache.org/>.
- [17] «Java,» [En línea]. Available:  
[https://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)).
- [18] «Oracle,» [En línea]. Available:  
[https://es.wikipedia.org/wiki/Oracle\\_Corporation](https://es.wikipedia.org/wiki/Oracle_Corporation).
- [19] «Android,» [En línea]. Available: <https://es.wikipedia.org/wiki/Android>.

- [20] «Dalvik,» [En línea]. Available: [es.wikipedia.org/wiki/Dalvik](https://es.wikipedia.org/wiki/Dalvik).
- [21] «ART,» [En línea]. Available: [https://es.wikipedia.org/wiki/Android\\_Runtime\\_\(ART\)](https://es.wikipedia.org/wiki/Android_Runtime_(ART)).
- [22] «XML,» [En línea]. Available: [es.wikipedia.org/wiki/Extensible\\_Markup\\_Language](https://es.wikipedia.org/wiki/Extensible_Markup_Language).
- [23] «SQLite,» [En línea]. Available: <https://es.wikipedia.org/wiki/SQLite>.
- [24] «App Engine,» [En línea]. Available: [https://es.wikipedia.org/wiki/Google\\_App\\_Engine](https://es.wikipedia.org/wiki/Google_App_Engine).
- [25] «Endpoints,» [En línea]. Available: <https://cloud.google.com/appengine/docs/java/endpoints/>.
- [26] «Objectify,» [En línea]. Available: <https://github.com/objectify/objectify>.
- [27] «Activity,» [En línea]. Available: <https://developer.android.com/reference/android/app/Activity.html>.
- [28] «Fragment,» [En línea]. Available: <https://developer.android.com/reference/android/app/Fragment.html>.
- [29] «Layout,» [En línea]. Available: <https://developer.android.com/guide/topics/ui/declaring-layout.html>.
- [30] «Navigation drawer,» [En línea]. Available: <https://developer.android.com/training/implementing-navigation/nav-drawer.html>.
- [31] «Material Design,» [En línea]. Available: <https://developer.android.com/design/material/index.html>.
- [32] «Intent,» [En línea]. Available: <https://developer.android.com/guide/components/intents-filters.html>.
- [33] «Adapter,» [En línea]. Available: <https://developer.android.com/reference/android/widget/Adapter.html>.
- [34] «RecyclerView,» [En línea]. Available: <https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html>.
- [35] «Patron de diseño View Holder,» [En línea]. Available: <https://developer.android.com/reference/android/support/v7/widget/RecyclerView.ViewHolder.html>.
- [36] «CardView,» [En línea]. Available: <https://developer.android.com/reference/android/support/v7/widget/CardView.html>.
- [37] «AsyncTask,» [En línea]. Available: <https://developer.android.com/reference/android/os/AsyncTask.html>.

- [38] «Google Analytics para móviles,» [En línea]. Available:  
<https://www.google.es/intl/es/analytics/features/mobile.html>.
- [39] ditrendia, «www.ditrendia.es,» [En línea]. Available:  
<http://www.ditrendia.es/wp-content/uploads/2015/07/Ditrendia-Informe-Mobile-en-Espa%C3%B1a-y-en-el-Mundo-2015.pdf>. [Último acceso: 27 05 2016].