

El Amigo Invisible en móviles conectados

Proyecto de Fin de Carrera

Ingeniería Técnica en Informática de Sistemas



**VNiVERSiDAD
D SALAMANCA**

Junio 2014

Autor

Quintín Sanabria Sánchez

Tutora

Ángeles M^a. Moreno Montero

D^a Ángeles M^a. Moreno Montero, profesora del Departamento de Informática y Automática, Facultad de Ciencias, de la Universidad de Salamanca

CERTIFICA:

Que el trabajo titulado “El Amigo Invisible en móviles conectados” ha sido realizado por D. Quintín Sanabria Sánchez, con DNI 41.519.550-L y constituye la memoria del trabajo realizado para la superación de la asignatura Proyecto de Fin de Carrera de la Titulación Ingeniería Técnica en Informática de Sistemas de esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a _____ de _____ de 20____

D^a Ángeles M^a: Moreno Montero
Dpto. Informática y Automática
Universidad de Salamanca

D^o Quintín Sanabria Sánchez
Universidad de Salamanca

CONTENIDO

Lista de figuras.....	6
1. INTRODUCCIÓN.....	8
2. OBJETIVOS DEL PROYECTO.....	10
3. CONCEPTOS TEÓRICOS.....	11
4. LOS DISPOSITIVOS MÓVILES CONECTADOS Y ANDROID.....	13
4.1. Los dispositivos móviles conectados.....	13
4.2. ¿Qué es Android?.....	14
4.3. Arquitectura de Android.....	15
4.3.1. Kernel de Linux.....	16
4.3.2. Bibliotecas nativas.....	16
4.3.3. Android Runtime.....	16
4.3.4. Framework de aplicación.....	16
4.3.5. Aplicaciones.....	17
4.4. Arquitectura de una aplicación.....	17
5. TÉCNICAS Y HERRAMIENTAS.....	20
5.1. Eclipse.....	20
5.2. Java.....	21
5.3. SDK Android.....	21
5.4. REM.....	22
5.5. Visual Paradigm.....	22
5.6. UML.....	23
5.7. Gantt Project.....	23
5.8. SQLite.....	24
5.9. SQLite Manager.....	24
5.10. Shared Preferences.....	25
5.11. Microsoft Word.....	25

5.12. Paint.....	25
5.13. Balsamiq Mockups.....	26
5.14. Deknop.....	26
6. ASPECTOS RELEVANTES DEL DESARROLLO.....	27
6.1. El ciclo de vida.....	27
6.2. Aspectos importantes de las fases.....	30
6.2.1. Requisitos.....	30
6.2.2. Análisis.....	30
6.2.3. Diseño.....	33
6.2.4. Implementación.....	39
6.2.5. Pruebas.....	39
6.3. Versión Android.....	40
6.4. Evaluación del evento.....	40
6.5. Modelo de datos.....	41
6.6. Restricciones del sorteo.....	42
6.7. Realización del sorteo.....	43
6.8. Notificación de los resultados del sorteo.....	46
6.9. Descripción funcional de la aplicación.....	47
7. TRABAJOS RELACIONADOS.....	55
8. CONCLUSIONES Y LÍNEAS FUTURAS.....	59
9. BIBLIOGRAFÍA.....	61

LISTA DE FIGURAS

Figura 1. Arquitectura de Android.....	15
Figura 2. Incrementos de la aplicación.....	29
Figura 3. Diagrama de casos de uso.....	32
Figura 4. Diagrama de paquetes.....	33
Figura 5. Diagrama de clases.....	36
Figura 6. Diseño interfaz.....	38
Figura 7. Diagrama de flujo sin exclusiones.....	44
Figura 8. Diagrama de flujo con exclusiones.....	45
Figura 9. Pantalla principal.....	47
Figura 10. Pantalla introducción evento.....	48
Figura 11. Pantalla selección grupos.....	49
Figura 12. Pantalla grupo.....	50
Figura 13. Pantalla nuevo participante.....	51
Figura 14. Pantalla exclusiones.....	52
Figura 15. Pantalla datos finales.....	53
Figura 16. Pantalla evento.....	55
Figura 17. Pantalla incluir grupo.....	56
Figura 18. Pantalla resumen.....	57
Figura 19. Tabla comparativa.....	58

1. INTRODUCCIÓN

Durante mucho años, los juegos populares han sido un pasatiempo en nuestra sociedad.

El denominado amigo invisible es uno de los más famosos en este ámbito. Proveniente de Sudamérica, y actualmente establecido en todo el mundo, consiste en una reunión de un grupo reducido de personas en la que se repartirán regalos según un sorteo ya desarrollado anteriormente. La lógica del juego es simple, pero cuenta con un problema, el sorteo, durante su realización todos los participantes deberían estar presentes.

Los avances tecnológicos ayudan a la conectividad entre personas y a la solución de problemas como este, haciendo que puedan realizarse con una mayor facilidad, rapidez y en cualquier lugar, únicamente con un dispositivo móvil.

El desarrollo de esta aplicación surge como idea para solucionar este problema, dando así la facilidad al grupo de usuarios del evento de realizar la asignación de los regalos e informar a cada participante de una manera individual de ese resultado, incluir los datos del evento (lugar, hora, temática, etc.), etc.

La memoria estará constituida por dos grandes bloques diferenciados, la descripción del proyecto (parte teórica en la que se expresará todo acerca de lo que trata el proyecto y lo relacionado con su consecución), y la parte de documentación técnica (dependerá de la metodología usada y será la parte donde recogeremos todo tipo de documentos generados por ella).

La parte de descripción estará formada por:

- **Objetivos del proyecto:** Se enumeran los objetivos de nuestro proyecto, mostrando tanto los marcados por los requisitos de usuario, cómo los de carácter técnico al llevar el proyecto a la práctica.
- **Conceptos teóricos:** Cómo la aplicación no consta de unos conceptos teóricos en sí mismos, se realiza una breve descripción del funcionamiento de nuestro tema y sus reglas.
- **Técnicas y Herramientas:** Se presenta una descripción de las herramientas utilizadas en el desarrollo del software y de las técnicas metodológicas usadas para llevarlo a cabo.
- **Aspectos relevantes del desarrollo:** Se presentarán los puntos más importantes del proyecto software, los aspectos relevantes del análisis, diseño e implementación, acompañado de una descripción funcional de la aplicación obtenida.
- **Trabajos relacionados:** Se mostrará algún proyecto relacionado que trabaje sobre el mismo ámbito.
- **Conclusiones y líneas futuras:** En este apartado se mostrarán las conclusiones obtenidas del proyecto y se marcarán unos posibles pasos de mejora.
- **Bibliografía:** Se mostrarán los elementos usados en el proyecto para la documentación y realización del mismo.

2. OBJETIVOS DEL PROYECTO

El objetivo principal de este proyecto es la realización de una aplicación para móviles que permita gestionar el juego o evento conocido como amigo invisible. Más concretamente se pretende:

- Organización de eventos.
- Gestión de grupos y participantes permitiendo la selección de estos a través de la agenda telefónica del Smartphone.
- Selección de las restricciones a aplicar en la realización del sorteo. Por ejemplo para especificar que Pepe no puede regalar a María.
- Comunicar inmediatamente y de forma confidencial a los participantes el resultado del sorteo.
- Permitir que los participantes puedan añadir a la agenda de sus móviles recordatorios sobre el evento.

Los objetivos de carácter técnico debidos a la realización de este proyecto son:

- Inmersión y aprendizaje del lenguaje de programación Android, para ello debe realizarse el aprendizaje de Java en el cual se basa éste, y de su IDE, Eclipse.
- Aprendizaje del sistema de almacenamiento en base de datos de un dispositivo móvil con Android (SQLite).
- Aplicación de una metodología para la orientación a objetos (ya que el lenguaje de programación utilizado así lo requiere).

3. CONCEPTOS TEÓRICOS

El único concepto teórico, previo para la comprensión del proyecto, será la comprensión del juego abordado y sus reglas.

AMIGO INVISIBLE

Se cree que sus primeros orígenes provienen de Sudamérica, a principios del siglo pasado, pero su extensión le ha llevado a convertirse en uno de los juegos populares más famosos.

En el juego participarán varias personas que se harán regalos entre ellas sin saber quién ha sido. Suelen concurrir dos requisitos:

- Que los participantes se conozcan entre sí
- Que su número no sea muy elevado

Normalmente suele realizarse en alguna festividad como por ejemplo en Navidad.

Para organizarlo es necesario que los participantes se reúnan antes de la fecha acordada para la entrega y realicen un sorteo para asignar a quién irá cada regalo, no pudiendo como es lógico regalarselo a uno mismo. En el sorteo, el organizador escribirá todos los nombres en papeles y después los introducirá en una pequeña bolsa. Después, uno por uno, los participantes irán introduciendo la mano en la bolsa y sacando un papel con el nombre de alguien a quién deberán hacer el regalo.

Finalmente, el día acordado, se reunirán todos los participantes realizando la entrega de regalos, donde alguien irá cogiendo regalos y entregándolos a la persona cuyo nombre figure en la tarjeta.

Existen muchas variedades de este juego, desde la introducción de una temática para el regalo, hasta el envío de cartas anónimas con seudónimos para intentar averiguar quién hace el regalo.

4. LOS DISPOSITIVOS MÓVILES CONECTADOS Y ANDROID

La aplicación que presentaremos estará encaminada al uso en dispositivos móviles conectados y correrán sobre el sistema operativo Android por lo que haremos una pequeña aproximación a los mismos.

4.1 LOS DISPOSITIVOS MÓVILES CONECTADOS

Se denomina dispositivo móvil conectado o teléfono inteligente al dispositivo que ofrece más funciones que un teléfono móvil convencional, es decir, es construido sobre una plataforma informática móvil y dispone de una mayor capacidad de computación y conectividad. El término inteligente hace referencia a la capacidad de usarse como un computador de bolsillo, pudiendo llegar a remplazar a un ordenador personal.

Una de las principales características de casi todos los teléfonos inteligentes es que permiten la instalación de programas para incrementar el procesamiento de datos y la conectividad, pudiendo ser desarrolladas dichas aplicaciones por el fabricante del dispositivo o por otra persona con los conocimientos necesarios. Se pueden realizar tareas al mismo tiempo sin necesidad de interrumpir ninguna de ellas, al igual que en un ordenador.

Las características comunes a estos dispositivos son:

- Cuentan con algún sistema operativo
- Permiten la instalación de programas de terceros
- Tienen GPS
- Acceso a internet
- Soportan correo electrónico

- Poseen agenda digital
- Permiten leer documentos en distintos formatos

4.2 ¿QUÉ ES ANDROID?

Android es una plataforma de software y un sistema operativo basado en Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como Smartphones o tabletas. Esta plataforma ofrece un sistema operativo y herramientas para el desarrollo de aplicaciones móviles.

Fue inicialmente desarrollado por Android Inc. y más tarde, en 2005, Google compró la empresa. Fue presentado en 2007 y el primer móvil con Android fue el HTC Dream (salió a la venta en 2008). A partir de ahí, este sistema operativo ha ido en aumento colocándose en el primer puesto de cuota de mercado.

Existe una gran comunidad de desarrolladores, realizando aplicaciones para extender la funcionalidad de los dispositivos. Estas aplicaciones están mayoritariamente en la tienda oficial de Android: Google Play, que es la tienda en línea administrada por Google.

Con el anuncio del sistema Android y la creación de la Open Handset Alliance (compañías dedicadas al desarrollo de estándares abiertos para dispositivos móviles), Google liberó la mayoría del código Android bajo la licencia Apache, una licencia libre y de código abierto.

El sistema operativo en sí, está compuesto por lenguajes de programación como C, C++, Java o XML.

Los programas están escritos en el lenguaje de programación Java y se ejecutan en un sistema Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de este lenguaje, con compilación en tiempo de ejecución. La ejecución se realizará en una máquina virtual que

es específica de Android, Dalvik, optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados.

4.3 ARQUITECTURA DE ANDROID

Mostraremos a continuación la estructura interna de Android, comentando las diferentes capas de las que consta, pudiendo apreciarse en la figura :

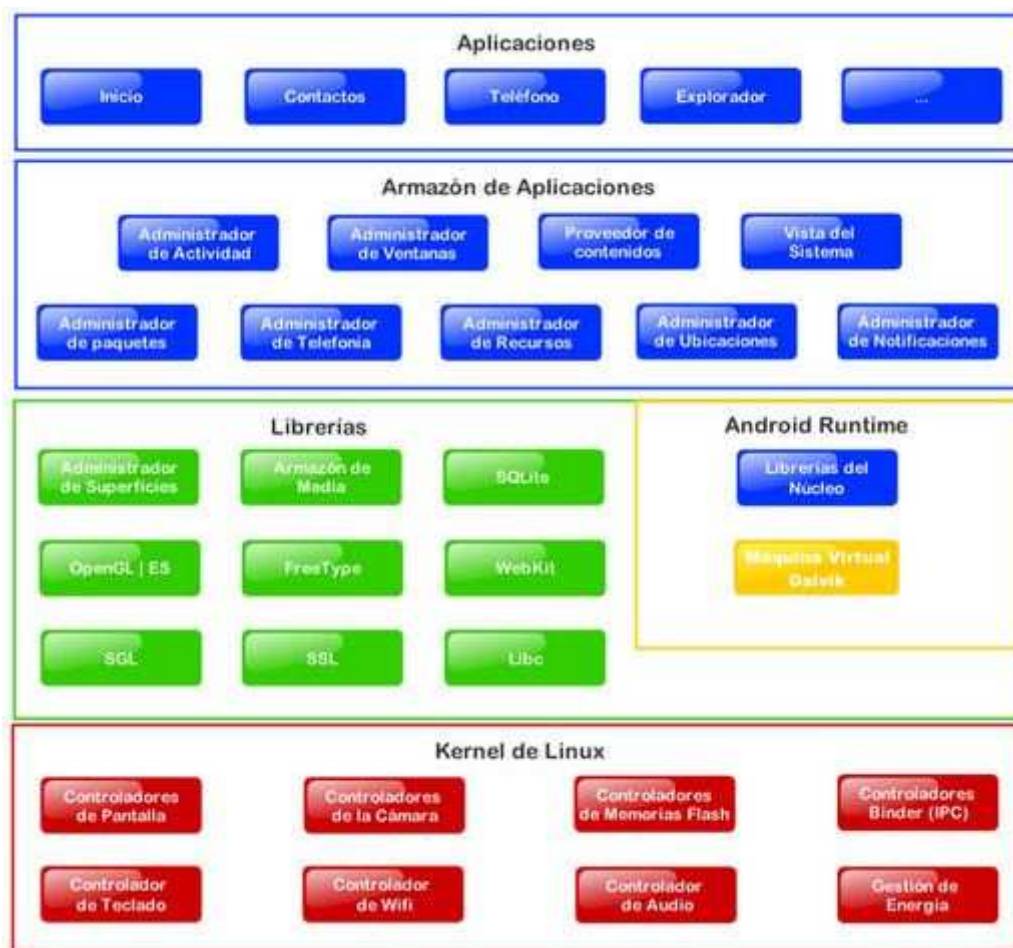


Figura 1. Arquitectura de Android

4.3.1 Kernel de Linux

En la parte más cercana al hardware se encuentra el kernel, núcleo de Linux. A pesar de estar basado en el núcleo de Linux, Android no es una distribución de este. Esta parte proporciona una serie de servicios esenciales (seguridad, memoria, procesos, etc).

4.3.2 Bibliotecas nativas

Por encima del núcleo tendremos las bibliotecas nativas de Android. Estas son una serie de bibliotecas codificadas en C/C++ utilizadas por diversos componentes del sistema que pueden ser accedidas por los desarrolladores a través del framework de aplicación. Se encuentran aquí bibliotecas que permiten el acceso a la pantalla, reproducción de audio y video, etc.

4.3.3 Android Runtime

Comprende los elementos de las partes referentes al tiempo de ejecución:

- Máquina virtual Dalvik: Android tiene su propia implementación de la máquina virtual de Java. Además está preparada para trabajar con archivos .dex (clases de java convertidas en tiempo de compilación).
- Librerías del núcleo: interfaces de programación de aplicaciones que proporcionan una potente y simple plataforma de desarrollo. Estas bibliotecas son las que proporcionan la mayoría de las funcionalidades disponibles en el núcleo del lenguaje Java.

4.3.4 Framework de aplicación

Se encuentran las estructuras que soportan las aplicaciones creadas. Permiten la comunicación entre el hardware y las aplicaciones.

Existen una serie de objetos (Manager) que se encargan de manejar los servicios ofrecidos por el hardware y devolver los recursos necesarios para su utilización.

4.3.5 Aplicaciones

Comprenden las aplicaciones básicas que instalamos y utilizamos en nuestro teléfono. Entre ellas se encuentran tales como cliente de correo electrónico, calendario, navegador, etc. Todas se encuentran escritas en Java.

4.4 ARQUITECTURA DE UNA APLICACIÓN

En Android pueden utilizarse elementos de una aplicación en otras siempre y cuando se permita. Para alcanzarlo, el sistema debe poder ejecutar un proceso de una aplicación cuando se requiera, por lo que las aplicaciones no disponen de un solo punto de entrada, sino que pueden iniciarse cuando se necesiten.

Los componentes que formarán una aplicación serán:

- **Activity:** representa una interfaz gráfica para una acción que el usuario puede realizar. Las aplicaciones pueden estar compuestas por una o varias actividades, cada una independiente del resto e implementada como subclase de la clase Activity. Cada activity tiene una ventana para dibujar sobre ella, cuyo contenido visual se realiza por medio de una jerarquía de vistas.
- **Services:** son procesos que se ejecutan de forma invisible de cara al usuario durante un tiempo indefinido, careciendo de interfaz gráfica. Son de gran utilidad para hacer una serie de tareas mientras el usuario se dedica a hacer otras. La comunicación con ellas se realiza

a través de una interfaz proporcionada por el mismo servicio. Hereda de la clase Service.

- **Content Provider:** elemento que permite el intercambio de información entre aplicaciones. Las clases creadas al efecto heredan de ContentProvider y han de implementar los métodos necesarios para permitir a otras aplicaciones leer o escribir información. Las aplicaciones usan un objeto de tipo ContentResolver que se encarga de ello, el cual maneja la comunicación entre procesos.
- **Broadcast Receiver:** estos objetos heredan de la clase BroadcastReceiver y tienen como tarea principal reaccionar ante una señal o un mensaje. No poseen interfaz gráfica, pero pueden lanzar actividades respondiendo a información recibida.
- **Intent:** es un mensaje de llamada asíncrono que sirve para iniciar actividades, servicios y receptores de difusión. Si se inicia una actividad o servicio hay que especificar la acción a realizar y la dirección de la información a enviar; si se inician receptores de difusión, el Intent da nombre a la acción que va a ser anunciada. En general, cada pantalla en la aplicación, se corresponde con una nueva actividad lanzada por un Intent.
- **AndroidManifest:** Cada aplicación tiene un archivo "AndroidManifest.xml" que presenta información esencial sobre esta en el sistema operativo. Tiene las siguientes funciones:
 - Se nombra el paquete Java de la aplicación, identificándolo únicamente.
 - Se describen los componentes de la aplicación.
 - Se enumeran las bibliotecas vinculadas.
 - Se asocia el icono y la etiqueta principal.

- Se declaran los permisos de la aplicación. Si una aplicación quiere acceder a los recursos hardware del dispositivo, debe solicitarlo a través de este archivo.
- Se declara el nivel mínimo de la API que se requiere para la aplicación.

5. TÉCNICAS Y HERRAMIENTAS

En este apartado describiremos las técnicas y herramientas utilizadas en el desarrollo del proyecto.

5.1 ECLIPSE

Es un entorno de desarrollo integrado de código abierto y multiplataforma, para desarrollar lo que se llaman "Aplicaciones de Cliente Enriquecido". Eclipse también es una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas.

Fue desarrollado originalmente por IBM, pero ahora es desarrollado por la Fundación Eclipse, una organización sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Este entorno de desarrollo usa módulos (plug-in) para proporcionar toda su funcionalidad, a diferencia de otros entornos, lo que permite a Eclipse extenderse usando otros lenguajes de programación como C/C++ y Python.

El SDK de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador interno y un modelo completo de los archivos fuente de Java, lo que permite técnicas de refactorización avanzadas.

Fue utilizada para la implementación de el proyecto software que se construyó, ya que es la herramienta más utilizada para la creación de aplicaciones Android y Java.

5.2 JAVA

Desarrollado por James Gosling (Sun Microsystems), es un lenguaje de programación orientado a objetos y basado en clases, en la que su sintaxis deriva de C y C++, pero con menos facilidades a bajo nivel que los anteriores.

Las aplicaciones creadas en Java pueden ser ejecutadas en cualquier máquina virtual Java sin importar la arquitectura subyacente, lo que da a este lenguaje potencia y uso en cualquier ámbito de la programación.

El lenguaje fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es que los programadores escriban un programa una única vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que un código que es ejecutado en una plataforma no tiene que ser recompilado en otra.

Java es actualmente uno de los lenguajes de programación más usados a nivel mundial, particularmente en aplicaciones cliente-servidor de web. También es la base de Android, por lo que tuvo que ser aprendido para realizar la programación en dicho sistema operativo.

5.3 SDK ANDROID

Es un conjunto de herramientas de desarrollo software que permiten al programador crear aplicaciones para un sistema concreto. Es una API creada para permitir la creación de aplicaciones en Android.

Incluye soporte para todo lo necesario por parte de un programador como entornos de desarrollo, bibliotecas, documentación, tutoriales, etc.

Se trata de un conjunto de herramientas muy completo y libre con el que es posible desarrollar aplicaciones y ejecutar un emulador del

sistema Android en la versión que se elija. Será usado para el desarrollo de la aplicación, su depuración y su comprobación en el emulador proporcionado.

5.4 REM

REM es una herramienta experimental gratuita de gestión de requisitos diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo software de acuerdo con la metodología definida en la tesis doctoral "Un entorno metodológico de ingeniería de requisitos para sistemas de información", presentada por Armando Durán en septiembre de 2000.

El objetivo de esta metodología es la definición de las tareas a realizar, los productos a obtener y las técnicas a emplear durante la actividad de análisis de requisitos de la fase de ingeniería de requisitos del ciclo de vida de la ingeniería del software mediante tablas que recopilarán los datos necesarios para la creación del proyecto.

Utilizaremos esta herramienta para una fácil recolección de los requisitos de nuestra aplicación y su posterior uso en el avance del ciclo de vida.

5.5 VISUAL PARADIGM

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos. Presenta licencia gratuita y comercial.

Será utilizado para crear documentación asociada al proyecto, tales como diagramas, debido a su facilidad.

5.6 UML (LENGUAJE UNIFICADO DE MODELADO)

Es el lenguaje de modelado de sistemas software más conocido y utilizado en la actualidad; está respaldado por el OMG. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema y aspectos concretos como expresiones de lenguajes de programación. Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir, en otras palabras, es el lenguaje en el que está descrito el modelo.

5.7 GANTT PROJECT

Es una herramienta libre de gestión de proyectos que permite la creación de diagramas de Gantt.

Este diagrama presentará la planificación temporal de un proyecto software a desarrollar, además de servir para gestionar los recursos del mismo. A pesar de esto, el diagrama de Gantt no indicará las relaciones existentes entre actividades.

Se usó por la facilidad de la creación de un eje temporal de la planificación del proyecto.

5.8 SQLITE

Es un sistema de gestión de bases de datos relacional, contenida en una relativamente pequeña biblioteca escrita en C. SQLite es un proyecto de dominio público creado por Richard Hipp.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices y los propios datos), son guardados como un sólo fichero estándar en la máquina del host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción. En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño.

Es usado para la inclusión en la aplicación de datos permanentes y de los que no sabemos a priori el número a almacenar.

5.9 SQLITE MANAGER

Es un software que permite la visualización de bases de datos SQLite. Pasando un archivo de este tipo de base de datos presenta los datos almacenados, los campos y sus valores.

Es muy útil para la comprobación del correcto almacenamiento de los datos de nuestra aplicación.

5.10 SHARED PREFERENCES

Las preferencias compartidas permiten al usuario almacenar una serie de datos (no muy elevado) en un fichero en la propia aplicación para agilizar el acceso a ciertos datos. Si el número de datos no es muy elevado, podremos sustituir una base de datos SQLite por este tipo de almacenamiento. Se almacenan en un archivo de preferencias a través de un par clave-valor (boolean, int, float...) que nos permitirá después recuperar el valor de lo almacenado de diferentes formas y en cualquier parte de la aplicación. Estos valores tienen un almacenamiento permanente como una base de datos, pero con un acceso mucho más rápido. Los datos se almacenan en un archivo con extensión .xml.

5.11 MICROSOFT WORD

Es un software destinado al procesamiento de textos. Fue creado por la empresa Microsoft, y actualmente viene integrado en la suite ofimática Microsoft Office.

Originalmente fue desarrollado por Richard Brodie para el computador de IBM bajo sistema operativo DOS en 1983. Versiones subsecuentes fueron programadas para muchas plataformas. Ha llegado a ser el procesador de textos más famoso del mundo.

Utilizado para la creación de todos los documentos de texto asociados a la documentación del proyecto.

5.12 PAINT

Es un programa editor de fotografía desarrollado por Microsoft. Paint ha acompañado al sistema operativo Microsoft Windows desde la versión 1.0. Siendo un programa básico, se incluye en todas las nuevas versiones de este sistema. Por su simplicidad, rápidamente se

convirtió en una de las aplicaciones más usadas de las primeras versiones de Windows y es todavía fuertemente asociado con la inmediata usabilidad de Windows.

5.13 BALSAMIQ MOCKUPS

Es una herramienta para crear interfaces de aplicaciones tanto web como de escritorio. Permite la creación de un boceto de la interfaz gráfica de nuestra aplicación mediante un simple menú que permite añadir todo tipo de elementos que nos podremos encontrar en nuestro software. Da una primera visión de la interfaz del proyecto y ayuda al desarrollador a organizar los elementos.

La aplicación es de pago, pero cuenta con una versión de prueba de 30 días.

Fue utilizada para un primer acercamiento a como sería la interfaz de nuestra aplicación.

5.14 DEKNOP

Es un gestor de botones que permite la creación de estos para un sitio web, presentaciones o archivos de ayuda.

Proporciona una interfaz paso a paso y fácil de usar que permite seleccionar el texto, estilo de fuente, color, tamaño y muchos otros aspectos. Puede visualizarse cualquier cambio que se haga al instante, por lo que es fácil experimentar hasta que se decida un diseño final.

Los archivos pueden almacenarse en formato JPG, BMP, PNG y GIF, además de poder guardar proyectos y continuarlos cuando se desee.

Utilizado para crear todos los botones que aparecen en nuestra aplicación.

6. ASPECTOS RELEVANTES DEL DESARROLLO

En este apartado describiremos los aspectos más importantes del desarrollo en las distintas fases del ciclo de vida de nuestro proyecto.

6.1 EL CICLO DE VIDA

En el desarrollo de este proyecto se ha seguido un modelo de ciclo de vida en el cual se siguieron una serie de pasos que llevan al software desde su concepción inicial hasta su entrega en este caso.

El proyecto fue abarcado definiendo una serie de requisitos inicialmente, que posteriormente fueron añadidos de forma incremental para obtener una mayor funcionalidad y refinar el mismo.

Al tener un proyecto ya definido, con una serie de requisitos que queríamos que nuestra aplicación tuviese, se fueron añadiendo en el orden establecido que se preparó de antemano, necesario para el correcto orden de los pasos de la aplicación.

El primer paso fue la parte de la introducción del evento, que es en el flujo de la aplicación el primer paso para realizar un evento. Se plantearon los diagramas de casos de uso y se continuó con su diseño y su implementación. Al final de dicho incremento se realizaron las pruebas necesarias para comprobar el correcto funcionamiento de esta parte en temas de validación de los eventos y en el almacenamiento de estos eventos en las preferencias compartidas.

El segundo incremento trató acerca de la introducción de grupos a la aplicación en un almacenamiento permanente, procediéndose de igual manera que en el primer incremento. Tras su análisis se introdujo la comprobación de nombres para que a la hora de añadir no pudieran repetirse un nombre de grupo ya escogido y evitar errores, además se

introdujo una validación del grupo para el número de participantes. Después del diseño y la implementación se comprobó el funcionamiento de dicha parte, encontrando un fallo que permitía a la hora de tener un grupo creado y querer modificarlo, eliminar participantes y salir del grupo volviendo atrás pudiendo quedar al grupo con menos participantes de los obligatorios. Dicho fallo fue corregido.

En el tercer incremento abarcamos la introducción de los participantes que como cabe esperar se realiza también para un almacenamiento permanente y que está solapado con el punto anterior de la creación de un grupo. En el análisis planteamos la restricción de que haya unos datos obligatorios para el posterior envío de los datos, con lo cual verificaremos los datos al introducir cada participante en la implementación. Al finalizar se realizaron una serie de pruebas para la verificación de este punto sin ningún error sustancial.

El cuarto incremento sigue los pasos de los anteriores, se realizaron las exclusiones entre participantes del grupo para el sorteo (si las hubiere). En esta parte hubo una serie de propuestas para realizar las exclusiones, ya que era un tema complicado para después poder usar los datos en el sorteo. Tras el análisis, diseño e implementación se comprobó esta parte sin grandes fallos.

El quinto mantiene lo visto en los incrementos anteriores, se encargará de la realización del sorteo, donde se llevará a cabo el envío para la información del evento. Tras la realización del análisis, diseño e implementación se llevó a cabo una serie de pruebas para la comprobación de la correcta asignación entre participantes incluyendo el caso de la realización de exclusiones, teniendo problemas de asignación en la implementación. Fueron llevados a cabo las correcciones oportunas y solventado los problemas.

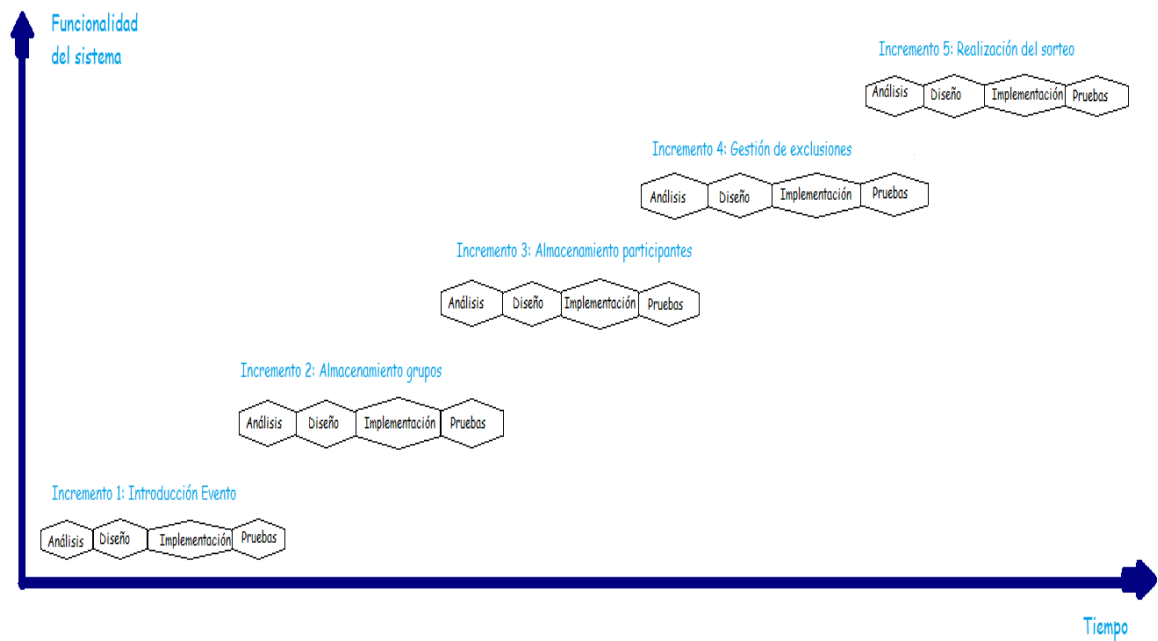


Figura 2. Incrementos de la aplicación

El tiempo empleado en cada incremento se puede observar en el Diagrama de Gantt, contenido en el Anexo 1.

6.2 ASPECTOS IMPORTANTES DE LAS FASES

A continuación se presentan las distintas disciplinas abarcadas en el desarrollo del software, mostrando los artefactos más importantes obtenidos en ellas:

6.2.1 Requisitos

Mediante la exposición del tema abarcado en el proyecto y los distintos tipos de aplicaciones ya existentes, se identificaron una serie de requisitos software que deseábamos que nuestra aplicación contuviera, se documentaron los requisitos funcionales, no funcionales y de información, y se tuvo una primera idea clara de la manera de construir el sistema.

6.2.2 Análisis

Se modelaron los casos de uso y se añadieron requisitos funcionales tales como la comprobación del evento y una notificación de la validez o de la no validez de este, la comprobación del nombre del grupo introducido, etc. Como puede verse en la figura 3, los casos de uso más significativos son los siguientes:

-Introducir Evento: Permite al usuario introducir los datos de un evento para que sea realizado, o la selección de un evento que ya hayamos introducido y que no haya finalizado. Realizará el almacenamiento del evento y la validación de este.

-Añadir Grupo: Permite al usuario introducir un nuevo grupo de participantes a la aplicación. Pide un nombre de grupo y lo verifica, además realiza la verificación del grupo antes de añadirlo a la base de datos (número válido de participantes).

-Añadir Participante: Es una extensión del caso de uso anterior, ya que la inclusión de un nuevo participante se realiza dentro del grupo en cuestión que estemos trabajando. Permite la introducción de un nuevo participante al grupo o la modificación de uno existente, ya que este caso de uso es lanzado también a la hora de modificar algún participante. También realiza la verificación para la validez de la introducción del participante.

-Buscar Participante: El usuario puede seleccionar un participante ya introducido y modificarlo o eliminarlo del grupo, permitiendo la flexibilidad de cara al usuario de la aplicación.

-Buscar Grupo: La misma flexibilidad para los grupos. El usuario podrá modificar dicho grupo (añadir más participantes, eliminar algunos, modificarlos,...), eliminar directamente el grupo de la base de datos o seleccionar dicho grupo para la realización del evento.

-Seleccionar Exclusiones: Permite al usuario la selección o no de exclusiones entre los participantes del grupo, en cuyo caso dará la opción de añadir exclusiones, ver las exclusiones realizadas y eliminar cualquiera de las hechas.

-Generar Sorteo: Realiza la asignación entre los participantes del grupo seleccionado. Da al usuario la opción del envío del resultado del sorteo a través del dispositivo móvil y realiza el envío obligatorio a través del correo electrónico. Además dará la opción al usuario de poder añadir al calendario del teléfono un recordatorio del evento.

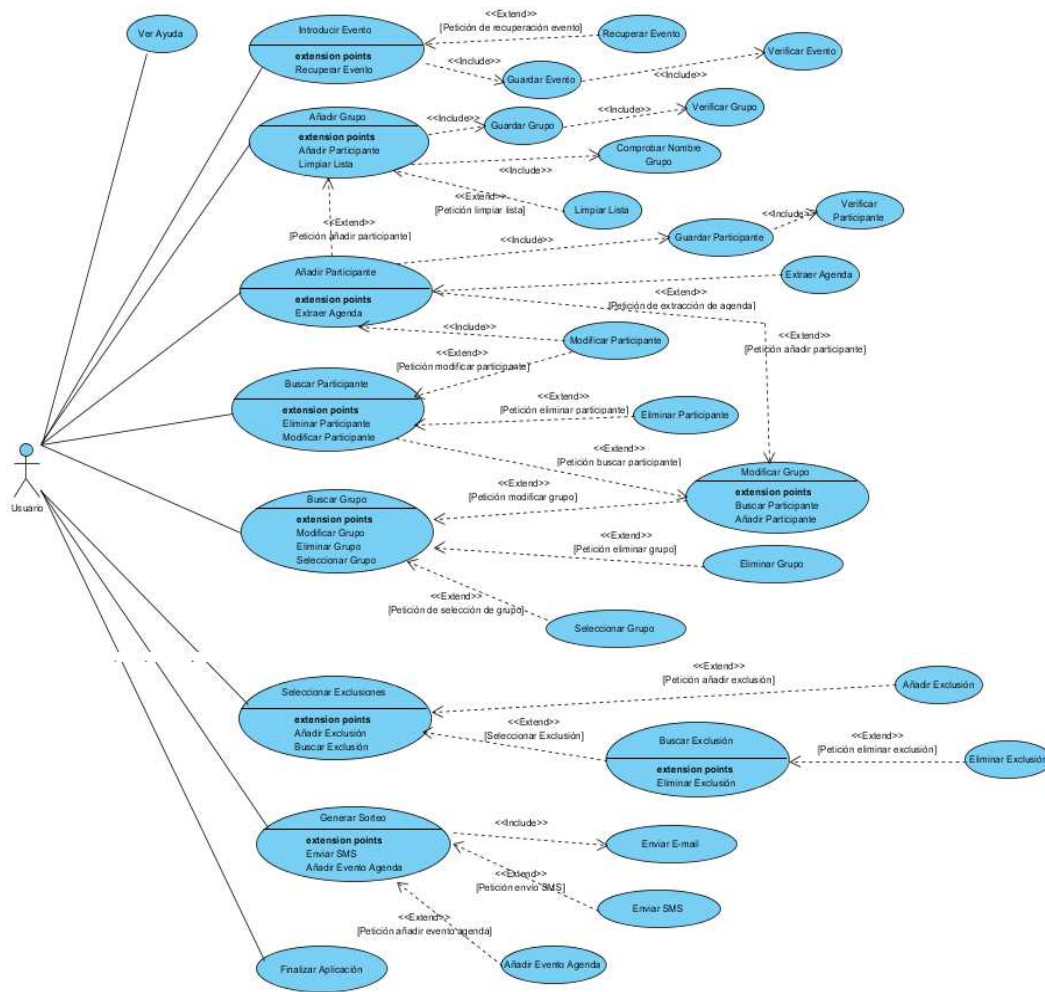


Figura 3. Diagrama de casos de uso

En el anexo 2 puede encontrarse una explicación más detallada de los puntos anteriores.

6.2.3 Diseño

Supone la entrada al dominio de la solución del problema, se realiza una división en paquetes del proyecto. Contaría con los paquetes Principal (contiene las clases de inicio de la aplicación, la pantalla principal), Ayuda (clase que presenta por pantalla una ayuda), Evento (clases para la introducción de un evento), Selección Grupos (clases para las opciones de los grupos), Grupo (clases para las opciones dentro de un grupo), Participantes (clases para la inclusión de participantes), Exclusiones (clases para el manejo de las exclusiones), Generar Sorteo (clases para la realización del sorteo y el envío de los datos), siendo la estructura de la figura 4:

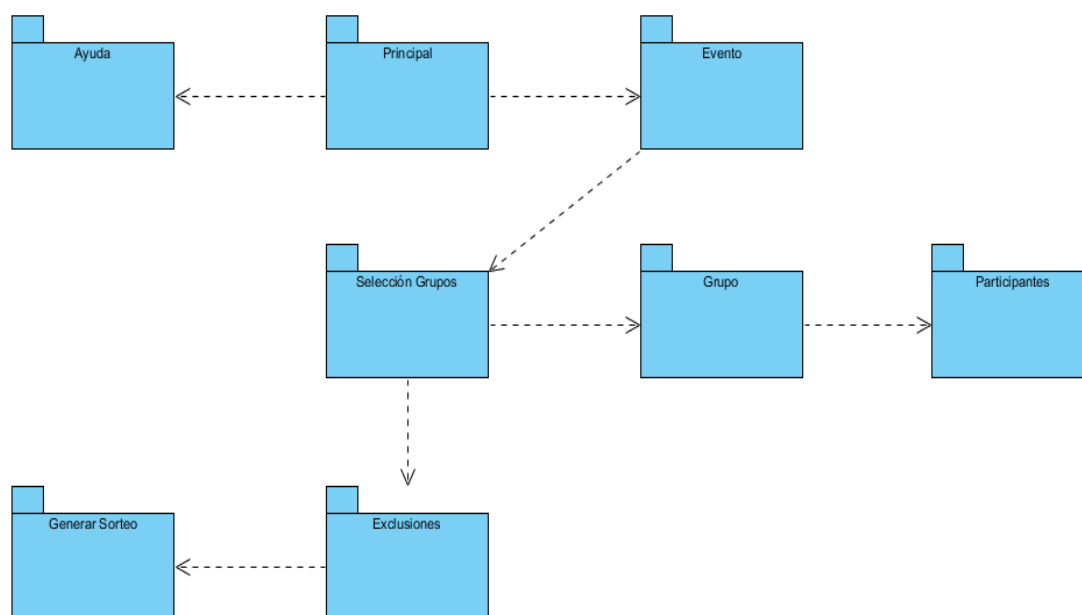


Figura 4. Diagrama de paquetes

En la figura anterior mostramos el diagrama de paquetes general, ahora describiremos el diagrama de paquetes completo, con sus clases asociadas (ver figura 4). Las clases pertenecientes a la interfaz gráfica se

definieron con la palabra "Pantalla" y a continuación el elemento que la identifica:

-Principal: Está compuesto por la clase "Pantalla Principal" que es la clase con la cual se inicia la aplicación. A partir de esta se podrá acceder a la clase "Pantalla Ayuda" que mostrará una pequeña ayuda de la aplicación y a la creación de un nuevo evento, además de la posibilidad de finalizar la aplicación.

-Ayuda: Compuesto por la clase "Pantalla Ayuda" que muestra una breve descripción de la aplicación.

-Evento: Compuesto por la clase "Pantalla Evento" que representa la interfaz donde introduciremos el evento, la clase "Recuperar Evento" que permitirá la carga de un evento guardado con anterioridad y la clase "D. Evento" que comprenderá la verificación de los datos del evento y el almacenamiento de estos.

-Selección Grupos: Está compuesto por la clase "Pantalla Selección Grupos" que muestra la interfaz de esta pantalla, y la clase "Grupo" que permite la interacción del usuario para crear grupos, modificar los ya existentes, eliminarlos o seleccionarlos para el desarrollo del evento.

-Grupo: Estará compuesto por la clase "Pantalla Grupo" que muestra la interfaz del grupo en el que nos encontremos, la clase "Participante" que permitirá al usuario el añadir un nuevo participante, la modificación de un participante del grupo o el borrado de estos. Contará también con la clase "Limpiar Lista", que proporcionará un borrado completo de todos los participantes asociados al grupo.

-Participantes: Lo compondrán la clase "Pantalla Participante" que mostrará la interfaz de introducción de datos para un participante y la clase

"D. Participante" que recogerá los datos introducidos, verificará dichos datos y añadirá el participante a la base de datos.

-Exclusiones: Estará compuesto por la clase "Pantalla Exclusiones" que mostrará la interfaz donde podrán realizarse las exclusiones, la clase "Exclusión Grupo" que permitirá dos opciones para el evento, "Sin Exclusiones" que no realizará exclusiones entre los participantes, y "Con Exclusiones" que permitirá la realización de exclusiones, donde tendremos dos para interactuar, "Añadir" que añadirá una exclusión entre participantes y "Eliminar" que eliminará exclusiones ya realizadas.

-Generar Sorteo: Contendrá la clase "Pantalla Generar Sorteo" donde se encontrará la interfaz de la aplicación, la clase "Sorteo" que realizará la asignación entre los participantes, y la clase "Envío Datos" donde se realizará el envío a través de los medios dispuestos de los datos del evento y del sorteo.

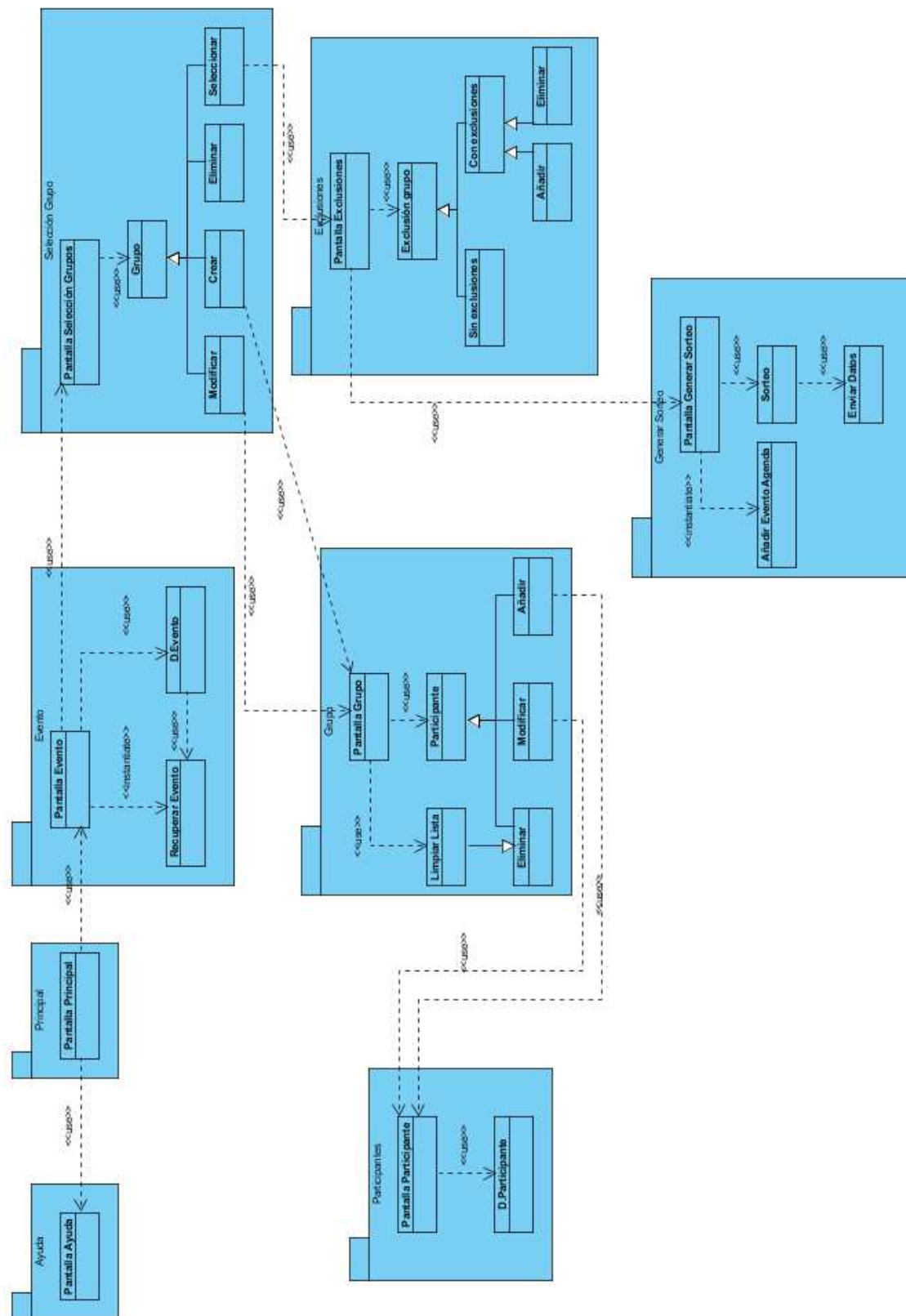


Figura 5. Diagrama de clases

Se tomaron una serie de decisiones también acerca de la interfaz del usuario. Al ser la aplicación en una serie de pasos el número de pantallas fue predefinido al principio del desarrollo de la aplicación y la concepción de cada una de las pantallas se mantuvo.

Se especificó un desarrollo preliminar a través de una herramienta diseñada para ello y se cargaron los elementos que serían necesarios para llevar a cabo la funcionalidad definida. Se llevó a cabo una división de interfaces para los distintos pasos de el flujo de la aplicación, es decir, una pantalla para el evento, otra para la selección del grupo, exclusiones, etc.

Los botones fueron dispuestos de tal manera que en todas las pantallas coincidieran en lugar y se tuviera abajo a la izquierda orientado los botones de retroceso en la aplicación y en la parte inferior derecha los de avance en esta. Los colores tanto de los botones como de las pantallas no variarán.

En la figura 6 se muestran las interfaces de usuario que han sido realizadas:

6.2.4 Implementación

Esta fase se ha llevado a cabo dividiendo los sistemas en pequeños subsistemas para su implementación, de acuerdo a su funcionalidad: almacenamiento evento, almacenamiento participantes, validación grupos,...

Puede consultarse en el Anexo 4.

6.2.5 Pruebas

Las pruebas han sido realizadas en todas las interacciones llevadas a cabo, verificando todos los requisitos de la aplicación y comprobando que eran cumplidos.

Fueron corregidos algunos errores, tales como cuando realizábamos la modificación de algún grupo y podíamos salir de él sin verificar que seguía cumpliendo la restricción de tener un número de participantes mínimo. Se realizaron el mayor número de pruebas en la fase de implementación, ya que a la hora de contener código era mucho más propenso a los errores.

Para comprobar el correcto funcionamiento de la aplicación final se llevó a cabo la realización de un evento completo realizado con varios usuarios reales que querían llevar a cabo el desarrollo del juego, cumpliendo los objetivos marcados y notificando a cada participante correctamente del evento y de su receptor del regalo.

6.3 VERSIÓN ANDROID

Un aspecto importante que cabe destacar es la versión elegida. En nuestra aplicación se buscó llegar al máximo número posible de usuarios lo que llevó a plantear como versión elegida la 2.2, una versión anterior a la que actualmente llevan los Smartphones, pero al ser las aplicaciones de Android retro-compatibles (una aplicación puede ejecutarse en un dispositivo que tenga una versión Android igual o superior), se decidió aplicar esta versión.

6.4 EVALUACIÓN DEL EVENTO

Una parte importante en el desarrollo del proyecto fue la validación de un evento introducido, ya que la aplicación no permite avanzar a realizar ninguna otra acción si no se introducen los datos mínimos de un evento.

Se tomaron como base los elementos que aparecían en otras aplicaciones similares, pero introduciendo una serie de cambios respecto a ellas como la no obligatoriedad de introducir precios mínimo y máximo (otras en cambio si lo requieren) o la inclusión de la temática del evento (tampoco es obligatoria). Sin embargo, si verificaremos que haya unos datos como mínimo para un nombre de evento, el lugar de la realización, una fecha y una hora para tener claros los datos de la reunión del grupo.

Dicho paso llevó a la introducción de otra funcionalidad que simplificaría el uso por parte del usuario, el guardado del evento introducido y la posible recuperación de los datos de este si por cualquier cuestión no se finalizó (salida inesperada de la aplicación, posterior realización de esta, etc). Esto da más flexibilidad a la aplicación y al usuario.

Con lo cual, se ha llevado a cabo el almacenamiento de dichos datos en un archivo Shared Preferences, realizando en primer lugar una

verificación del evento introducido en el código para la correcta introducción de este y no dejando avanzar si la validación no fuese correcta, realizado con la pulsación del botón "Siguiente" para continuar al próximo paso de la aplicación. Si la verificación es correcta, los datos del evento (tanto opcionales como obligatorios) serán almacenados, estando disponibles para su posterior acceso y envío a los participantes o para su recuperación directa en el inicio de la aplicación cuando un evento que hemos comenzado a realizar no ha finalizado. Dicha opción de recuperación se mostrará al entrar en la creación de un nuevo evento, mostrando un cuadro de diálogo que permitirá la opción de recuperar los datos anteriores o no hacerlo, cargando en pantalla la información si seleccionáramos recuperarlos.

Los datos al realizar un evento completo serán borrados del fichero donde fueron almacenados.

6.5 MODELO DE DATOS

Otra de las partes principales de la aplicación es la asociación de un grupo de participantes a el evento que hayamos realizado para su correcto funcionamiento. Tras la observación de este punto se tomó la opción de almacenar los grupos de participantes de una manera permanente, es decir, tener disponibles los grupos de participantes ya introducidos por si se decidiesen usar en otras ocasiones, por lo que se permite asociar cualquier grupo ya usado en otra ocasión a un nuevo evento. Esto evitaría al usuario tener que volver a introducir todos los datos de los participantes que forman el grupo, ya que es una tarea muy pesada.

Se optó por almacenar dichos datos en un sistema gestor de bases de datos típica de Android (SQLite), ya que en principio no sabemos el

tamaño ni el número de grupos que el usuario almacenará, con lo cual al poder ser muy elevado se decantó por esta opción.

A partir de aquí, se podrá realizar cualquier tipo de acción con los participantes y grupos (modificación, eliminación y selección para la realización del evento), accediendo a esta base de datos.

->Para los grupos se almacenarán:

-ID: Es el identificador del grupo en la base de datos.

-Nombre: Será el nombre del grupo y no podrá repetirse.

->Para los participantes almacenaremos:

-ID: Es el identificador del participante en el grupo.

-Nombre: Es el nombre del participante asociado y no podrá repetirse dentro del grupo.

-E-mail: Será la dirección de correo electrónico del participante y no podrá estar en blanco, ya que es un dato obligatorio.

-Teléfono: Número de teléfono del participante. Será un dato extra, es decir, no es obligatorio que exista.

6.6 RESTRICCIONES DEL SORTEO

Una parte importante y también a destacar es la realización de las exclusiones entre participantes del grupo si el usuario las selecciona.

En la mayoría de aplicaciones por el estilo no se da esta opción para excluir a ciertos participantes con otros. Viendo que este punto era importante, se decidió incluir este apartado como un elemento más del flujo de la aplicación para que el usuario lo visualizase y pudiera elegir si realizarlas o no, pero siempre estando esta opción visible.

Los datos de las exclusiones realizadas serían almacenados en un archivo Shared Preferences, el cual contendría una serie de pares clave-valor que almacenaría el participante origen de la exclusión y otro par el participante destino, es decir:

OrigenX-Participante

DestinoX-Participante

con lo cual tendríamos para un valor de X, un sentido de exclusión entre dos participantes.

Dichas exclusiones presentan opciones para ser borradas y para ser realizadas en ambos sentidos, es decir, si dos participantes quieren ser excluidos mutuamente, no hará falta realizar dos exclusiones diferentes sino que se marcará la casilla que aparece para que estos sean excluidos mutuamente.

6.7 REALIZACIÓN DEL SORTEO

Otro tema a destacar es la realización de la asignación entre los participantes del sorteo. En este apartado hubo que tener en cuenta dos aspectos: si la realización del sorteo conllevaba exclusiones descritas por el usuario o si no las había.

En la opción de no seleccionar ningún tipo de exclusión, tendremos un método que realizará una asignación aleatoria para cada participante del grupo introducido, previniendo los casos de que se asignen a sí mismo o que dicho participante ya haya sido asignado. En la figura 7 mostraremos el diagrama de flujo asociado a la realización del sorteo para esta parte:

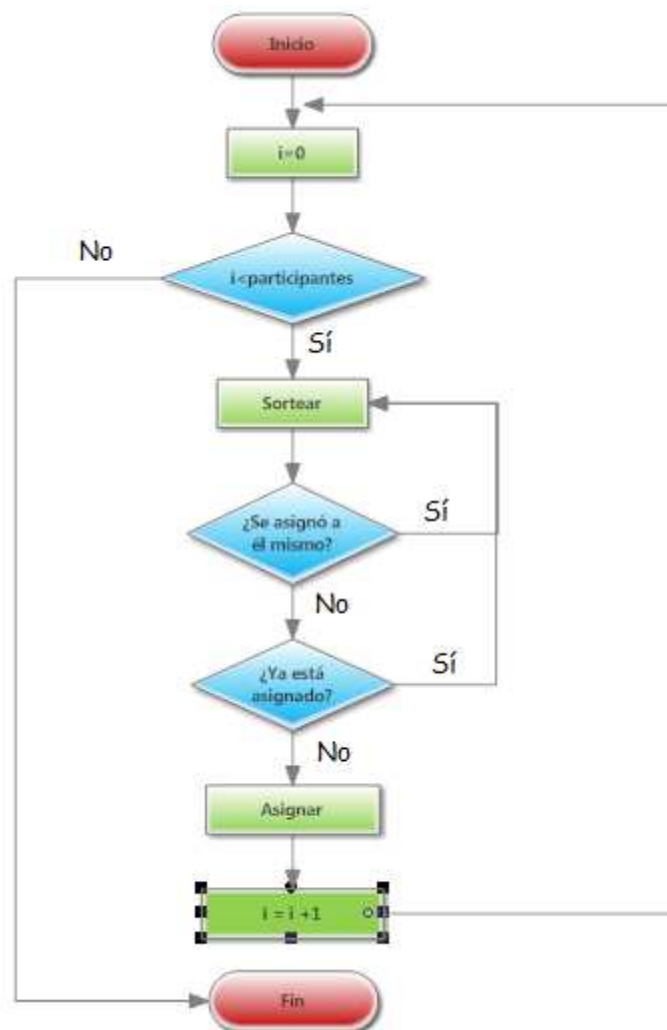


Figura 7. Diagrama flujo sin exclusiones

Para la opción de la realización del sorteo con exclusiones tendremos que realizar el mismo proceso que en el apartado anterior, pero incluyendo una parte nueva, la introducción cuando realizamos el sorteo de los participantes a los que no se podrá regalar. Cada persona del grupo podrá tener uno o varios participantes a los que no podrá regalar, por lo que irán introduciéndose según vaya realizándose la asignación de cada una y eliminándose cuando el participante tenga otro asignando. En la figura 8 mostraremos el diagrama de flujo de esta parte:

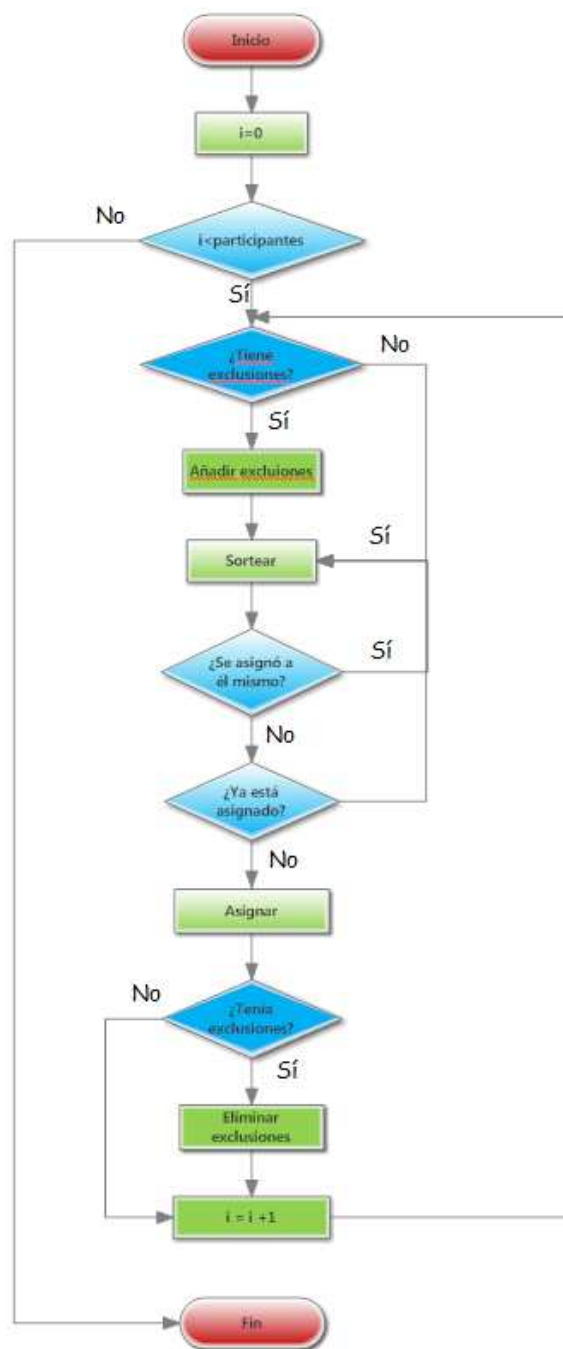


Figura 8. Diagrama flujo con exclusiones

6.8 NOTIFICACIÓN DE LOS RESULTADOS DEL SORTEO

Al ser nuestra aplicación totalmente confidencial (no mostrará los datos por pantalla para mantener en secreto el evento, incluso hasta para el organizador), deberemos tener algún método de envío que se realice obligatoriamente. El envío obligatorio deberá ser por correo electrónico, por lo que todos los participantes deberán tener un correo asociado cuando sean introducidos.

A parte de este método de envío, se dará la opción de el envío también a través de un SMS, siendo este opcional ya que dicho mensaje podría tener coste para el organizador.

Cuando el usuario realice la generación del sorteo, se hará automáticamente el envío del correo a todos los participantes del grupo y si seleccionó la opción de SMS, también se hará este a los componentes del grupo que tengan un número de teléfono asociado.

6.9 DESCRIPCIÓN FUNCIONAL DE LA APLICACIÓN

Describiremos brevemente el funcionamiento de la aplicación.

Tendremos una pantalla principal de la aplicación (puede verse en la figura 9):



Figura 9. Pantalla Principal

Muestra las opciones disponibles:

- Nuevo Grupo: Nos llevará a la creación de un evento para la realización del cometido de la aplicación.
- Ayuda: Mostrará la pantalla donde aparecerá una breve descripción de la aplicación.
- Salir: Permitirá al usuario salir de la aplicación.

Mostrará la pantalla para introducir los datos de un nuevo evento:

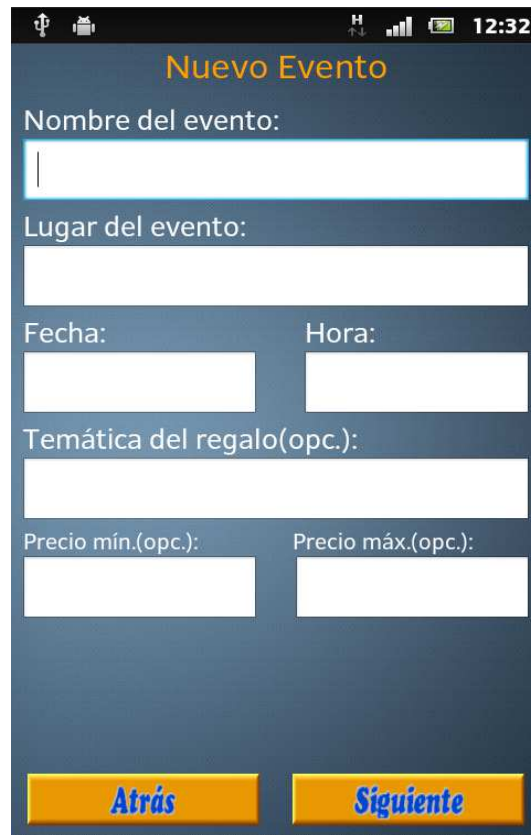


Figura 10. Pantalla Introducción Evento

En la figura 10 se muestra el formato para la introducción de los datos del evento. Contendrá la posibilidad de volver a la pantalla principal ("Atrás") o de continuar con el flujo de la aplicación ("Siguiente") siempre que los datos del evento sean válidos (realizaría la verificación y el almacenado de los datos del evento en un archivo para su posterior acceso).

Si tuviéramos un evento ya introducido de otra ocasión que estuviera almacenado como hemos comentado anteriormente, al acceder a esta pantalla nos aparecería un cuadro de diálogo para la recuperación y carga de esos datos.

Si pulsáramos "Recuperar", cargaríamos los datos guardados en cada uno de los recuadros para los que hubiera información, si se pulsara "No Recuperar" no se efectuaría ningún cambio.

Tras la pulsación de "Siguiente" en el apartado anterior, iremos a la siguiente pantalla:



Figura 11. Pantalla Selección Grupos

Nos aparecerán los grupos que hayamos introducido en la aplicación anteriormente (los cuales pinchando en ellos nos permitirán realizar diferentes opciones como modificar, eliminar o seleccionar el grupo para el evento), un botón ("Crear Grupo") para la creación de un nuevo grupo y un recuadro que mostrará el grupo que seleccionaremos para nuestro evento y que no podrá estar vacío cuando seleccionemos el botón "Siguiente" para continuar.

Si realizáramos la pulsación del botón "Crear Grupo" nos aparecería un cuadro de diálogo para la introducción del nombre del grupo.

Cuando realizamos la introducción de un nombre para el grupo y pulsemos "Continuar" la aplicación nos llevará a la creación de ese grupo (siempre y cuando el nombre no esté repetido o no esté en blanco).

Nos encontraremos en la pantalla del grupo seleccionado. A este apartado podremos llegar por dos caminos: a través de la creación de un nuevo grupo como comentamos en el apartado anterior, o por la selección de "Modificar" de un grupo ya introducido que aparecerán pinchando sobre él.

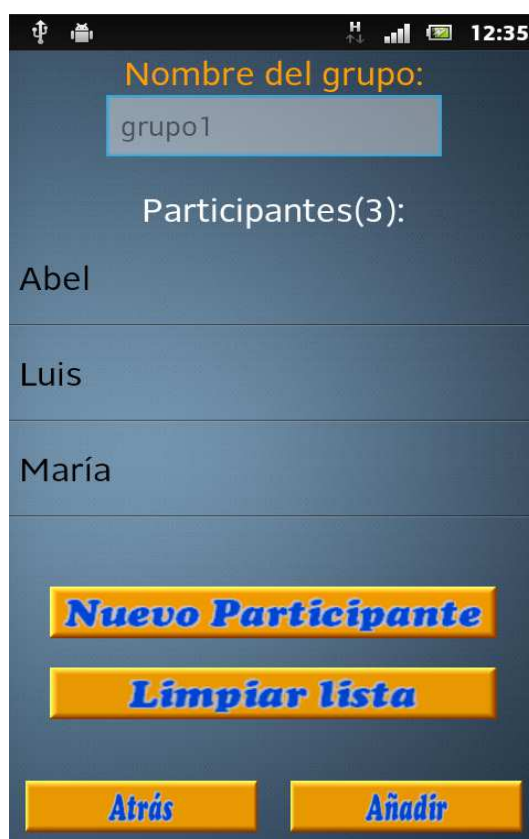


Figura 12. Pantalla Grupo

Nos encontraremos dentro del grupo, donde se realizará la asignación de los participantes. En este punto podremos añadir participantes al grupo

("Nuevo Participante"), o realizar cualquier acción con los ya introducidos (eliminar y modificar) pinchando sobre ellos, también tendremos la opción de eliminar todos los participantes asignados a la vez con la pulsación de "Limpiar Lista".

Una vez modificado o creado el grupo se pulsaría "Añadir" para la correcta creación, añadiéndolo a la base de datos y devolviéndonos a la pantalla anterior, donde aparecería dicho grupo.

Si en el apartado anterior se seleccionase la introducción de un nuevo participante, o si a un participante ya introducido se le seleccionase la opción de modificarlo, iríamos a la siguiente pantalla:



Figura 13. Pantalla Nuevo Participante

Aquí tendríamos las opciones necesarias para añadir un participante al grupo, además de la opción "Desde Agenda..." que permitirá cargar los

datos de un participante desde la agenda del teléfono. Cuando introduzcamos los datos pulsaremos "Agregar", que añadirá el participante a la base de datos(si los datos son correctos) y nos devolverá a la pantalla del grupo.

Cuando en la pantalla selección de grupos seleccionamos un grupo introducido y pulsamos "Siguiente" avanzaremos así hasta la pantalla de introducción de exclusiones:

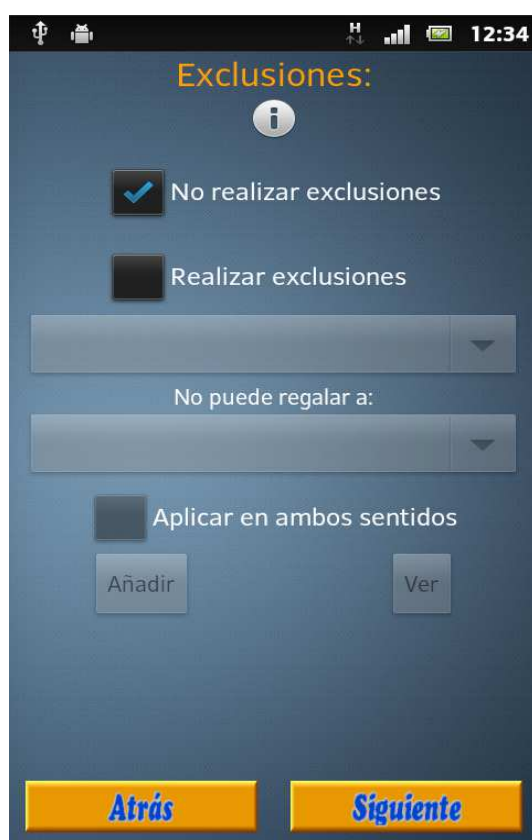


Figura 14. Pantalla Exclusiones

Tendremos en primer lugar un CheckBox para seleccionar si deseamos realizar exclusiones en el grupo o no.

Si no las deseamos lo dejaremos como aparece por defecto.

Si en cambio deseamos realizar exclusiones marcaremos la casilla "Realizar Exclusiones", lo que activará los elementos que aparecen debajo

para poder trabajar con ellos. Aparecerán dos listas desplegables que permitirán la selección de los participantes del grupo seleccionado, la primera seleccionará que participante no puede regalar y la segunda a quién no podrá. Una vez elegidos dos participantes se podrá marcar la casilla "Ambos Sentidos" si deseamos que la relación sea mutuamente excluyente, lo que agilizará la realización de estas.

Tendremos también la opción de visualizar las exclusiones que hemos realizado y la opción de eliminarlas pulsando en el botón "Ver" y pinchando en cualquiera de ellas.

Al finalizar las exclusiones y avanzar, tendremos la pantalla final donde se mostrarán todos los datos del evento introducido y los participantes asociados para la comprobación final de dichos datos:



The screenshot shows a mobile application interface with a dark blue background. At the top, there is a status bar with icons for signal, battery, and time (12:35). Below the status bar, the title "Descripción del evento:" is displayed in orange. The event name "Fallas" is shown in white. Below it, "Se celebrará en: Valencia" is displayed in white. The date and time are shown as "El día: 13/11/2013" and "A las: 17:53" in orange. The theme "Temática: A elección de los participantes" is shown in white. The gift value "Valor regalo: Sin definir" is shown in orange. The number of participants "Participantes(3):" is shown in orange. Below this, a list of participants is shown in white: Abel, Luis, and María. At the bottom, there are two orange buttons with white text: "Modificar Datos" and "Generar Sorteo".

Evento	Nombre	Fecha	Hora	Temática	Valor regalo	Participantes
Fallas	Valencia	13/11/2013	17:53	A elección de los participantes	Sin definir	Abel, Luis, María

Figura 15. Pantalla Datos Finales

Aquí tendremos dos opciones, "Modificar Datos" si los datos hemos comprobado que no son correctos o que nos falta alguno y "Generar Sorteo", el cual pedirá si se desea el envío de los datos a través de SMS y después realizará la asignación entre participantes, el envío de los datos y el borrado del evento introducido.

7. TRABAJOS RELACIONADOS

Aquí describiremos las diferencias de nuestro software con otro que trate sobre el mismo tema. Describiremos una de las aplicaciones que hemos encontrado, llamada "Amigo Invisible", desarrollada por Androvelopers, con fecha de la última versión 21 de noviembre de 2013 (2.0).

La aplicación dispone de una pantalla principal que tendrá dos botones, uno para empezar el desarrollo de la aplicación y otro para una ayuda. Tras pulsar "Empezar" comenzará lo que nos interesa, dispondrá de una pantalla para la introducción de los datos de un evento, siendo obligatorio introducir todos y cada uno de los datos para avanzar:

The screenshot shows a mobile application interface for entering event data. At the top, a blue header bar contains the text "1. Introduce los datos del evento". Below this, there are several input fields: a text field for "Nombre", a text field for "Lugar", two text fields for "Fecha" and "Hora", a currency selector showing "Moneda € - Euro" with a dropdown arrow, and two text fields for "Precio mínimo (€)" and "Precio máximo (€)". At the bottom of the form is a large, light gray button labeled "Siguiente". The status bar at the very top shows the time as 11:37 and various mobile connectivity icons.

Figura 16. Pantalla Evento

Tras la introducción de los datos y el avance pulsando el botón "Siguiente", se nos mostrará una pantalla para la introducción de los datos de los participantes asociados al evento:

2. Introduzca los datos de los participantes:

Nombre

Correo(opcional) @gmail.com ▼

Participantes (0)

Anterior Siguiente

Figura 17. Pantalla Incluir Grupo

En esta pantalla tendremos que introducir el nombre de los participantes asociados y el correo (opcionalmente), ya que el resultado se mostrará por pantalla. Tras agregar cada uno de nuestros participantes, pulsaremos "Siguiente", lo que nos llevará a la pantalla para realizar el sorteo:



Figura 18. Pantalla Resumen

Mostrará los datos de la aplicación y los participantes del sorteo para su comprobación. Tras esta comprobación se pulsará "Realizar Sorteo", lo que realizará la asignación, enviará al correo los datos (si el participante tiene correo asociado) y mostrará los resultados por pantalla. Tras esto, la aplicación finalizará su flujo correctamente.

Destacaremos que cuando se comenzó el desarrollo de la aplicación se encontraron aplicaciones similares pero todas carecían de algo que no las permitía desarrollar el objetivo del juego abordado al completo (falta de introducción de exclusiones, envío de datos a través de otros medios, total confidencialidad del sorteo incluso para el que lo realiza,...) por lo que en nuestra aplicación tratamos de coger cada uno de estos pasos y unificarlos como parte esencial de la misma, añadiendo además partes que dan al

usuario una mayor flexibilidad y facilidad de uso (extraer datos directamente de la agenda del teléfono, grupos de participantes almacenados permanentemente,...).

Mostraremos en la figura 19 una tabla comparativa con distintas funcionalidades:

	Nuestra aplicación	Amigo Invisible
Almacena Evento	Sí	No
Almacena grupos y participantes	Sí	No
Permite exclusiones	Sí	No
Muestra el resultado en pantalla	No	Sí
Envía datos por correo electrónico	Sí	Sí
Guarda el evento en el calendario	Sí	No

Figura 19. Tabla comparativa

8. CONCLUSIONES Y LÍNEAS FUTURAS

Al finalizar el proyecto tendremos una aplicación para la realización del sorteo del amigo invisible con las siguientes funcionalidades:

- Total confidencialidad del sorteo
- Total flexibilidad con grupos y participantes posibilitando la asociación a nuevos eventos de grupos ya creados
- Posibilidad de añadir al calendario del teléfono el evento para que sea recordado
- Posible recuperación de un evento que hayamos introducido con anterioridad y que por cualquier razón no haya finalizado
- Realización de exclusiones entre los participantes del sorteo
- Opción de envío de los datos a través de SMS

Se han aplicado conocimientos obtenidos en los años de estudio durante el desarrollo de la misma, además de la adquisición de una serie de conocimientos aprendidos por cuenta propia como son el aprendizaje del lenguaje Java y Android, muy utilizados en el mercado y válidos para nuestra formación en el futuro.

Se plantearán una serie de líneas futuras para la mejora de la aplicación en algunos aspectos que parecen interesantes:

- Funcionalidad en otras plataformas.
- Añadir un servidor en el cual los participantes puedan subir fotos y mediante el cual se pueda realizar el aviso a todos los demás participantes del grupo cuando han sido incluidos en un evento y de los datos del sorteo, siendo todo ello online.

- Envío de los datos por otras vías, por ejemplo, permitir el acceso a través de facebook pudiendo crear grupos con los usuarios del evento para informar sobre este.

9. BIBLIOGRAFÍA

Para el desarrollo de la memoria se han utilizado los siguientes documentos:

- García Peñalvo, F.J (2011). Apuntes de la asignatura "Ingeniería del Software". Departamento de Informática y Automática, Universidad de Salamanca.

- García Peñalvo, F.J, Maudes Raedo, J.M, Piarrini Velthuis, M.G, García-Bermejo Giner, J.R y Moreno García, M.N (2000). Proyecto de final de carrera, Ingeniería Técnica en Informática: Guía de realización y documentación. Departamento de Informática y Automática, Universidad de Salamanca.

- Durán Toro, A. y Bernárdez Jiménez, B. (2002). Metodología para la Elicitación de Requisitos de Sistemas Software. Departamento de Lenguajes y Sistemas Informáticos, Escuela Técnica Superior de Ingeniería Informática.

Así mismo se han utilizado una serie de páginas web como ayuda para la realización del sistema:

- Enciclopedia: <http://es.wikipedia.org>
- Página oficial de Eclipse: <http://www.eclipse.org>
- Página oficial de Android: <http://www.android.com>
- Curso de Android: <http://www.tutos4u.com>
- Curso de Android: <http://www.javaya.com.ar/androidya>
- Curso de Android: <http://www.youtube.com> (Perfil de Jesús Conde).
- Curso de Java: <http://www.youtube.com> (Perfil de Jesús Conde).
- Foro Android: <http://www.android-spa.com>
- Foro Android: <http://stackoverflow.com>