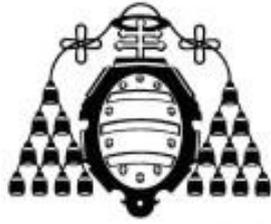


UNIVERSIDAD DE OVIEDO

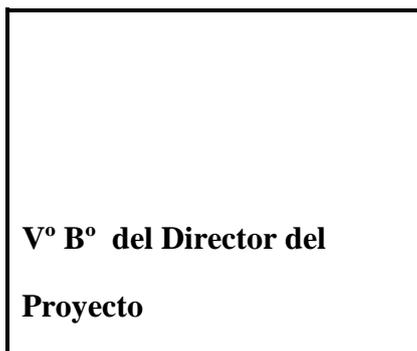


ESCUELA DE INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MASTER

“Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario”

DIRECTOR: Darío Álvarez Gutiérrez



AUTOR: Borja Antonio Trapiella Gil

Agradecimientos

A mis padres

A todos mis compañeros en Imagine800

Resumen

El contenido del proyecto, y su objetivo, se puede separar en dos partes claramente diferenciadas.

Por un lado presentamos un sistema de animaciones basado sobre la figura de un personaje en 3D, con un claro estilo caricaturizado. Este parte tiene una clara orientación hacia el ocio, que se completa con la posibilidad de que los usuarios generen su propio contenido dentro del sistema, ampliando de esta forma el número de posibilidades que ofrece la aplicación, y por tanto aumentando el interés que puede despertar en los potenciales usuarios y la duración de su “vida útil”.

Este último concepto es especialmente importante dentro del entorno objetivo del presente proyecto, puesto que este tipo de aplicaciones para dispositivos móviles se mueven dentro de un mercado muy cambiante, en el que el éxito depende en gran medida de factores como la novedad y la capacidad de llamar la atención, no solo en un principio, sino a lo largo del tiempo en el que se distribuye la aplicación.

En este caso concreto, todos estos factores se intentan cubrir no solo proporcionando los mecanismos para que los usuarios generen su propio contenido original (animaciones del personaje en base a grabaciones con su propia voz), sino también con la imagen del personaje que se utiliza, puesto que representa la caricatura de un personaje público, reconocible y con una importante presencia comercial/publicitaria.

El otro punto sobre el que gira el proyecto, es en la investigación y desarrollo de las diferentes alternativas que existen para los protocolos de intercambio de mensajes, tanto entre dispositivos móviles (independientemente de su sistema operativo), como clientes más enfocados al entorno web.

En este sentido se busca la solución más “global” posible teniendo en cuenta los requisitos anteriores, de forma que dé respuesta a todas nuestras necesidades optimizando los recursos y el trabajo necesario.

Además, esta parte no se ha enfocado de forma independiente a la que comentábamos en primer lugar, y se ha combinado con factores característicos tanto del entorno de comercialización y distribución de la aplicación anterior (dispositivos móviles, redes sociales, publicación de contenido), como del propio funcionamiento del sistema (generación de animaciones de manera dinámica).

Palabras Clave

Dispositivos móviles, Redes sociales, Protocolo de mensajería, XMPP, Android, C2DM, Notificaciones, Animaciones, Cliente, Servidor

Abstract

The content of the project, so is its objective, could be splitted up in two well defined parts.

On one hand, an animation system based on a character in 3D is shown in a clearly caricatured style. This part has an entertainment aim, which is completed by the possibility given to the users to generate their own content within the system, increasing this way the range of uses that the application offers, increasing, therefore, the interest that can be awoken within the potential users and its lifetime duration.

This last concept is specially important referring to the objective of this project, assuming that these kind of mobile phone applications are established in a quite changing market, in which its success depends on key points like their newness and their catching attention ability, not only in the beginning but also during the time when these applications are distributed.

In this specific case, all these key factors are achieved not only giving mechanisms to the users that allows them to generate their own original content (character animations based on the user's voice) but also using an image of the character they want to caricature because of the fact that who they are going to caricature is a well-known public person with an important public presence.

On the other hand, this project is based on research and further development of different alternatives of the protocols that already exist to exchange messages, either among mobile devices (without taking in account their OS) or among web focused clients.

Referring to this point, a solution as global as possible has been achieved focusing in these requirements, so it can give an answer to all our necessities optimizing resources and effort.

Furthermore, this part has not been focused in an independent way compared to the one that has been spoken of first, and it has been combined with characteristic factors not only from the commercialization and distribution range (mobile devices, social networks, content publication) but also from the system own operation mode (dynamic animation generation).

Keywords

Mobile devices, Social networks, Messaging protocol, XMPP, Android, C2DM, Notifications, Animations, Client, Server

Índice General

| | |
|---|-----------|
| CAPÍTULO 1. MEMORIA DEL PROYECTO..... | 21 |
| 1.1 RESUMEN DE LA MOTIVACIÓN, OBJETIVOS Y ALCANCE DEL PROYECTO | 21 |
| 1.2 RESUMEN DE TODOS LOS ASPECTOS | 23 |
| CAPÍTULO 2. INTRODUCCIÓN..... | 25 |
| 2.1 JUSTIFICACIÓN DEL PROYECTO | 25 |
| 2.2 OBJETIVOS DEL PROYECTO | 30 |
| 2.3 ESTUDIO DE LA SITUACIÓN ACTUAL | 31 |
| 2.3.1 <i>Sistema de animaciones</i> | 31 |
| 2.3.2 <i>Protocolo de mensajería</i> | 31 |
| CAPÍTULO 3. ASPECTOS TEÓRICOS..... | 35 |
| 3.1 ANDROID | 35 |
| 3.1.1 <i>Android SDK</i> | 36 |
| 3.1.2 <i>Android NDK</i> | 36 |
| 3.2 JAVACV | 37 |
| 3.3 SONIC..... | 37 |
| 3.4 A/SMACK | 38 |
| 3.5 MONETIZACIÓN..... | 38 |
| 3.5.1 <i>Google AdMob</i> | 38 |
| 3.5.2 <i>Mecanismos publicitarios propios</i> | 39 |
| 3.6 CONTROL ESTADÍSTICO DE LA APLICACIÓN – GOOGLE ANALYTICS..... | 39 |
| 3.7 MODUM..... | 40 |
| CAPÍTULO 4. PLANIFICACIÓN DEL PROYECTO Y RESUMEN DE PRESUPUESTOS | 41 |
| 4.1 PLANIFICACIÓN..... | 41 |
| 4.2 RESUMEN DEL PRESUPUESTO | 45 |
| CAPÍTULO 5. ANÁLISIS | 47 |
| 5.1 DEFINICIÓN DEL SISTEMA | 47 |
| 5.1.1 <i>Determinación del Alcance del Sistema</i> | 47 |
| 5.2 REQUISITOS DEL SISTEMA | 49 |
| 5.2.1 <i>Obtención de los Requisitos del Sistema</i> | 49 |
| 5.2.2 <i>Identificación de Actores del Sistema</i> | 51 |
| 5.2.3 <i>Especificación de Casos de Uso</i> | 51 |
| 5.3 IDENTIFICACIÓN DE LOS SUBSISTEMAS EN LA FASE DE ANÁLISIS | 56 |
| 5.3.1 <i>Descripción de los Subsistemas</i> | 56 |
| 5.3.2 <i>Descripción de los Interfaces entre Subsistemas</i> | 57 |
| 5.4 DIAGRAMA DE CLASES PRELIMINAR DEL ANÁLISIS..... | 59 |
| 5.4.1 <i>Diagrama de Clases</i> | 59 |
| 5.4.2 <i>Descripción de las Clases</i> | 60 |
| 5.5 ANÁLISIS DE CASOS DE USO Y ESCENARIOS | 64 |
| 5.5.1 <i>Reproducir vídeo</i> | 64 |
| 5.5.2 <i>Reproducir animación</i> | 65 |
| 5.5.3 <i>Generar animaciones propias</i> | 67 |

Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario

| | | |
|--|--|------------|
| 5.5.4 | Guardar vídeo | 69 |
| 5.5.5 | Compartir vídeo en ámbito general | 70 |
| 5.5.6 | Compartir vídeo ámbito del chat interno | 72 |
| 5.5.7 | Enviar mensaje con/sin conexión | 73 |
| 5.5.8 | Reproducir animaciones enviadas/recibidas | 75 |
| 5.6 | ANÁLISIS DE INTERFACES DE USUARIO | 77 |
| 5.6.1 | Descripción de la Interfaz | 77 |
| 5.6.2 | Descripción del Comportamiento de la Interfaz | 79 |
| 5.6.3 | Diagrama de Navegabilidad | 80 |
| 5.7 | ESPECIFICACIÓN DEL PLAN DE PRUEBAS | 81 |
| 5.7.1 | Pruebas unitarias | 81 |
| 5.7.2 | Pruebas de integración del sistema | 83 |
| 5.7.3 | Pruebas de usabilidad | 85 |
| CAPÍTULO 6. DISEÑO DEL SISTEMA..... | | 87 |
| 6.1 | ARQUITECTURA DEL SISTEMA..... | 87 |
| 6.1.1 | Diagramas de Paquetes | 87 |
| 6.1.2 | Diagramas de Componentes | 91 |
| 6.1.3 | Diagramas de Despliegue | 92 |
| 6.2 | DISEÑO DE CLASES | 93 |
| 6.2.1 | Diagrama de Clases | 93 |
| 6.3 | DIAGRAMAS DE SECUENCIA | 95 |
| 6.4 | DISEÑO DE LA BASE DE DATOS | 97 |
| 6.4.1 | Descripción del SGBD Usado | 97 |
| 6.4.2 | Integración del SGBD en Nuestro Sistema | 97 |
| 6.4.3 | Diagrama E-R | 98 |
| 6.5 | DISEÑO DE LA INTERFAZ..... | 99 |
| 6.6 | ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS | 102 |
| 6.6.1 | Pruebas Unitarias..... | 102 |
| 6.6.2 | Pruebas de Integración y del Sistema | 109 |
| 6.6.3 | Pruebas de Usabilidad y Accesibilidad | 115 |
| 6.6.4 | Pruebas de Rendimiento | 117 |
| CAPÍTULO 7. IMPLEMENTACIÓN DEL SISTEMA..... | | 119 |
| 7.1 | LENGUAJES DE PROGRAMACIÓN..... | 119 |
| 7.1.1 | Java | 119 |
| 7.2 | HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO | 120 |
| 7.2.1 | Eclipse | 120 |
| 7.2.2 | Android ADT Bundle | 120 |
| 7.2.3 | Rockmongo | 121 |
| 7.3 | CREACIÓN DEL SISTEMA..... | 122 |
| 7.3.1 | Problemas Encontrados | 122 |
| 7.3.2 | Descripción Detallada de las Clases | 126 |
| CAPÍTULO 8. DESARROLLO DE LAS PRUEBAS..... | | 149 |
| 8.1 | PRUEBAS UNITARIAS..... | 149 |
| 8.1.1 | Subsistema de animaciones | 149 |
| 8.1.2 | Subsistema de mensajería | 154 |
| 8.2 | PRUEBAS DE INTEGRACIÓN Y DEL SISTEMA..... | 157 |
| 8.3 | PRUEBAS DE USABILIDAD | 163 |

Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario

| | | |
|---|---|------------|
| 8.3.1 | <i>Cuestionarios del perfil 1</i> | 163 |
| 8.3.2 | <i>Cuestionarios del perfil 2</i> | 165 |
| 8.3.3 | <i>Cuestionarios del perfil 3</i> | 168 |
| 8.3.4 | <i>Cuestionarios del perfil 4</i> | 171 |
| 8.3.5 | <i>Cuestionarios del perfil 5</i> | 174 |
| 8.3.6 | <i>Cuestionarios del perfil 6</i> | 177 |
| 8.3.7 | <i>Cuestionarios del perfil 7</i> | 180 |
| 8.3.8 | <i>Cuestionarios del perfil 8</i> | 183 |
| 8.3.9 | <i>Conclusiones sobre la usabilidad</i> | 185 |
| 8.4 | PRUEBAS DE RENDIMIENTO | 186 |
| CAPÍTULO 9. MANUALES DEL SISTEMA | | 187 |
| 9.1 | MANUAL DE INSTALACIÓN | 187 |
| 9.2 | MANUAL DE USUARIO | 190 |
| 9.2.1 | <i>Sistema de animaciones</i> | 190 |
| 9.2.2 | <i>Sistema de mensajería</i> | 198 |
| 9.2.3 | <i>Interacción con elementos publicitarios y notificaciones</i> | 205 |
| CAPÍTULO 10. CONCLUSIONES Y AMPLIACIONES | | 207 |
| 10.1 | CONCLUSIONES | 207 |
| 10.2 | AMPLIACIONES..... | 208 |
| CAPÍTULO 11. PRESUPUESTO..... | | 209 |
| 11.1 | COSTE DE RECURSOS MATERIALES..... | 209 |
| 11.2 | COSTE DE RECURSOS HUMANOS..... | 211 |
| 11.3 | PRESUPUESTO COMPLETO | 213 |
| CAPÍTULO 12. REFERENCIAS BIBLIOGRÁFICAS | | 215 |
| 12.1 | LIBROS Y ARTÍCULOS..... | 215 |
| 12.2 | REFERENCIAS EN INTERNET | 216 |
| CAPÍTULO 13. APÉNDICES..... | | 217 |
| 13.1 | GLOSARIO Y DICCIONARIO DE DATOS..... | 217 |
| 13.2 | CONTENIDO ENTREGADO EN EL CD-ROM..... | 219 |
| 13.2.1 | <i>Contenidos</i> | 219 |

Índice de Figuras

| | | |
|-----------|--|-----|
| Figura 1 | Listado de aplicaciones "Talking" en Google Play | 25 |
| Figura 2 | Ejemplo "Talking Pocoyó" | 26 |
| Figura 3 | Imagen "Talking Tom Cat" | 27 |
| Figura 4 | Ejemplo de chat entre usuarios del juego Apalabrados..... | 28 |
| Figura 5 | Ejemplo del chat entre usuarios de la aplicación móvil de Skype | 28 |
| Figura 6 | Ejemplo de chat entre plataformas (PC - Móvil)..... | 29 |
| Figura 7 | Logo XMPP..... | 33 |
| Figura 8 | Logotipo de Android, conocido como 'Andy' | 35 |
| Figura 9 | Comparativa entre el Android presente en un dispositivo de Amazon, y la versión original sin modificar | 36 |
| Figura 10 | Logo JavaCV | 37 |
| Figura 11 | Logo Google Analytics | 40 |
| Figura 12 | Diagrama de Gantt de la planificación | 43 |
| Figura 13 | Detalle de tareas de la planificación | 44 |
| Figura 14 | Casos de Uso del Sistema de Animaciones | 52 |
| Figura 15 | Casos de uso del Sistema de Chat | 53 |
| Figura 16 | Diagrama de subsistemas | 58 |
| Figura 17 | Diagrama de clases preliminar | 59 |
| Figura 18 | Caso de uso 'Reproducir vídeo' | 64 |
| Figura 19 | Caso de uso 'Reproducir animación' | 65 |
| Figura 20 | Caso de uso 'Generar animaciones propias' | 67 |
| Figura 21 | Caso de uso 'Guardar vídeo' | 69 |
| Figura 22 | Caso de uso 'Compartir vídeo en ámbito general' | 70 |
| Figura 23 | Caso de uso 'Compartir vídeo en el ámbito del chat interno' | 72 |
| Figura 24 | Caso de uso 'Enviar mensaje con/sin conexión' | 73 |
| Figura 25 | Caso de uso 'Reproducir animaciones enviadas/recibidas' | 75 |
| Figura 26 | Esquema de la pantalla principal | 77 |
| Figura 27 | Esquema de la pantalla del listado de contactos del sistema de mensajería..... | 78 |
| Figura 28 | Esquema de la pantalla de conversación del sistema de mensajería | 79 |
| Figura 29 | Diagrama de navegabilidad..... | 80 |
| Figura 30 | Diagrama de componentes | 91 |
| Figura 31 | Diagrama de despliegue..... | 92 |
| Figura 32 | Diagrama de clases | 94 |
| Figura 33 | Diagrama de secuencia de la generación de una animación y envío mediante el sistema de mensajería..... | 96 |
| Figura 34 | Diagrama Entidad Relación | 98 |
| Figura 35 | Splash inicial de la aplicación | 99 |
| Figura 36 | Pantalla principal de la aplicación | 99 |
| Figura 37 | Menú de opciones de la pantalla principal..... | 100 |
| Figura 38 | Pantalla principal del sistema de mensajería con el listado de contactos..... | 100 |
| Figura 39 | Pantalla de una conversación del sistema de mensajería..... | 101 |
| Figura 40 | Logo Eclipse | 120 |
| Figura 42 | Menu de ajustes preinstalación | 188 |
| Figura 43 | Permisos de la aplicación | 188 |
| Figura 44 | Aplicación instalada correctamente | 189 |

Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario

| | |
|--|-----|
| Figura 45 . Captura que muestra los botones del interfaz deshabilitados | 191 |
| Figura 46 . Botones de las animaciones de vídeo | 192 |
| Figura 47 . Pitch agudo activado | 192 |
| Figura 48 . Pitch grave actviado | 192 |
| Figura 49 . Cambio de posición del personaje durante el proceso de grabación | 193 |
| Figura 50 . Opciones del menú de animaciones | 194 |
| Figura 51 . Captura de los vídeos generados en la galería multimedia del dispositivo | 195 |
| Figura 52 . Menú desplegado tras la opción de Compartir | 196 |
| Figura 53 . Animación enviada mediante el sistema de mensajería de la aplicaicón | 197 |
| Figura 54 . Botoón de ayuda en la pantalla de animaciones..... | 197 |
| Figura 55 . Pantalla de ayuda del sistema de animaciones..... | 198 |
| Figura 56 . Login con Facebook | 199 |
| Figura 57 . Listado de contactos extraídos de Facebook | 200 |
| Figura 58 . Campo enviar de la pantalla de conversación..... | 200 |
| Figura 59 . Ejemplo de mensaje no leído..... | 201 |
| Figura 60 . Mensaje formateado como animación en la conversación del chat..... | 202 |
| Figura 61 . Pantalla de reproducción de animacion desde el sistema de mensajería..... | 203 |
| Figura 62 . Antes y después de actualizar un contacto | 204 |
| Figura 63 . Eejemplo del banner para el modo de prueba | 205 |
| Figura 64 . Eejmplo de interstitial publicitario | 206 |

Capítulo 1. Memoria del Proyecto

En este primer capítulo se pretende hacer una explicación de las motivaciones del proyecto, los objetivos que se persiguen e indicar el alcance del mismo.

1.1 Resumen de la Motivación, Objetivos y Alcance del Proyecto

El origen de este proyecto se inicia en el comienzo de la realización de unas prácticas laborales entre la empresa “Imagine 800” y el alumno que firma la presente documentación.

Durante la duración de este periodo, se realizó una investigación acerca de las diferentes posibilidades que existían para desarrollar una aplicación para dispositivos móviles (Android), en la que se cumpliesen una serie de características:

- Reproducción fluida de animaciones
- Grabación de audio en el dispositivo
- Modificación de audio pregrabado
- Generación de animaciones en base a un audio pregrabado
- Generación de vídeos en base a un conjunto de fotogramas y un audio pregrabado
- Inclusión de elementos sociales
- Máxima optimización posible en diferentes gamas de dispositivos

Una de las principales dificultades que supuso abordar el conjunto anterior de requisitos/objetivos, fue la de lograr el mejor funcionamiento en todo tipo de dispositivos, teniendo en cuenta el ecosistema tan heterogéneo que existe en Android en cuestión de terminales (sobre todo si se compara con otros ecosistemas como el de iOS).

Además de esto, merece la pena llamar la atención sobre la modificación del audio y la generación de vídeos en base a fotogramas y audio por separados, algo que no está disponible de forma nativa en la plataforma Android, y que requirió un proceso previo de investigación y pruebas sobre las distintas alternativas posibles.

Teniendo en cuenta todo lo anterior, la motivación final que engloba a todo esta parte del proyecto es la de obtener un sistema que, cumpliendo con todo lo anterior, permita generar aplicaciones de este tipo utilizando diferentes diseños y estilos, de manera que se pueda expandir y generar nueva actividad comercial en torno a todo este proyecto. Especialmente teniendo en cuenta que este tipo de sistemas son muy habituales actualmente en este tipo de entornos, bajo importantes temáticas comerciales y publicitarias.

De especial importancia será también la inclusión de los diferentes mecanismos disponibles para poder monetizar la aplicación, enfocados siempre desde el punto de vista publicitario, ya sea acudiendo a soluciones de terceros o implementando mecanismos propios.

Por otro lado, centrándonos en la parte del proyecto que gira en torno a la implementación de un sistema propio de mensajería instantánea, la motivación y los objetivos que se persiguen con la investigación de aspectos sobre este tema se enfocan de otra manera.

Actualmente, la empresa “Imagine800” dispone de soluciones desarrolladas internamente que permiten implementar un sistema de videollamadas en diferentes plataformas (móvil, web, etc).

En base a esto, se consideraba importante la posibilidad de completar este producto con la inclusión de un sistema de mensajería instantánea, que añada una funcionalidad extra, y que cumpla con la misma característica fundamental: ser lo más multiplataforma posible.

Con esta motivación, se decidió emplear el mismo proceso de desarrollo del sistema anterior, para realizar las pruebas de concepto (a nivel práctico) de como incorporar un sistema de mensajería instantánea dentro de una aplicación móvil, empleando los elementos característicos de esta plataforma, y mostrando ejemplos de cómo completarlo añadiendo elementos propios del entorno donde se integra (en este caso el sistema de animaciones).

Además de esto, también se han desarrollado otras pruebas en la que se intercambian mensajes entre diferentes plataformas, con el objetivo de completar la investigación en torno a este punto de la forma más precisa posible.

En base a todo lo anterior, el alcance del proyecto abarcaría lo siguiente:

- Sistema de animaciones
 - Reproducción de animaciones
 - Grabación y edición de audio
 - Generación de vídeos
 - Generación de animaciones
 - Factores sociales
 - Optimización multidispositivo
 - Monetización
- Sistema de mensajería instantánea
 - Integrado y coherente en el sistema de animaciones
 - Compatible entre plataformas

1.2 Resumen de Todos los Aspectos

En este apartado se hará un resumen del proceso que se ha seguido durante la elaboración del proyecto para poder llevar a cabo el trabajo con la menor complejidad posible y de manera organizada.

El inicio del proyecto se centró en todo lo relacionado con la parte del sistema de las animaciones. El espíritu de este proceso de desarrollo siempre fue lo más práctico posible, ejecutando los avances y cambios (por pequeños que fuesen) constantemente en los distintos dispositivos de prueba utilizados. Desde mi punto de vista este factor es algo muy importante en los proyectos desarrollados sobre dispositivos móviles, puesto que los simuladores disponibles a nivel software no proporcionan una experiencia comparable a la de los dispositivos reales, y más teniendo en cuenta la importancia del rendimiento entre varios dispositivos en el sistema que nos ocupa.

Además, durante esta parte del proceso de desarrollo, se utilizaron diferentes aplicaciones a modo de “módulos” independientes, que se centraban en diferentes funcionalidades, y que posteriormente se irían integrando en la versión definitiva. La división en partes independientes se realizó de la siguiente manera:

- Reproducción de animaciones
- Grabación y modificación de audio
- Generación de animaciones en base a audio
- Generación de vídeos en base a fotogramas y audio independientes

Una vez que se finalizó esta parte del proyecto, se inició un proceso de investigación en torno a las distintas alternativas disponibles sobre protocolos, clientes y servidores, para la solución de mensajería instantánea que estábamos buscando, y en base a los requisitos que ya hemos comentado anteriormente.

Al igual que en la parte anterior del proyecto, las pruebas e implementaciones se hicieron de manera independiente, de manera que se pudiesen hacer todo tipo de pruebas y cambios en torno a esta funcionalidad.

Una vez las partes básicas estaban claramente definidas (conexiones, envío y recibimiento de mensajes, etc), se integró con el sistema definitivo, para posteriormente terminar de pulir el funcionamiento y los detalles del conjunto final.

Todo este proceso, y como suele ser habitual, se finalizó con una fase de pruebas de todos los elementos integrados, para afinar de manera precisa el funcionamiento del conjunto (las partes individuales eran probadas de manera independiente antes de integrarse en la versión definitiva).

Capítulo 2. Introducción

2.1 Justificación del Proyecto

En la actualidad, con la enorme expansión del mercado de dispositivos móviles y sus correspondientes “mercados” de aplicaciones, se han generado una gran cantidad de nuevos modelos de negocio que, en mayor o menor medida, han demostrado ser una gran posibilidad a la hora de explotar una idea, una imagen o un concepto.

El caso que nos ocupa se centra en las aplicaciones tipo “Talking”, en las que se centra todo el protagonismo a un personaje, centrado en la pantalla del dispositivo, el cual realizará una serie de acciones, normalmente con una intención cómica, ya sea como resultado de la acción del usuario o bien automáticamente como parte de la ejecución de la aplicación.



Figura 1 Listado de aplicaciones "Talking" en Google Play

El principal atractivo que presentan este tipo de sistemas, y que determina en la mayor parte de los casos su éxito o fracaso, es la forma de diseñar y explotar el personaje o personajes que se utilizan como protagonistas de la aplicación. En la mayoría de los casos, este modelo de negocio se utiliza para explotar una imagen publicitaria de personajes fácilmente reconocibles por la mayoría del público objetivo, ya sean series de televisión, películas, deportes, o vida pública en general, aunque también existen importantes excepciones que han sabido labrarse un sitio en lo alto del mercado de este tipo de aplicaciones, por su buen hacer, originalidad y buen trabajo en líneas generales.



Figura 2 Ejemplo "Talking Pocoyó"

En este sentido, y ante la posibilidad de disponer de un trabajo de diseño que cumplía especialmente con las premisas de esbozar a un personaje tanto reconocible como explotable desde una vía humorística, se decidió desarrollar una aplicación siguiendo la estela y las intenciones marcadas por otras aplicaciones similares, pero siempre intentando obtener nuestro propio estilo y marca de identidad en el proceso, marcándonos como meta la calidad alcanzada por uno de los referentes en este tipo de aplicaciones, como es la aplicación "Talking Tom Cat", ejemplo reseñable no sólo por su calidad, sino por ser una de las más exitosas en términos económicos, y sin requerir el uso de un personaje previamente reconocible (ellos hicieron reconocible a su propio diseño).



Figura 3. Imagen "Talking Tom Cat"

Por otro lado, otro aspecto a tratar son los sistemas de intercambio de mensajes que existen actualmente en las diferentes aplicaciones para móviles. Aunque son sobradamente conocidas aplicaciones como Whatsapp, Viber, o Line, en nuestro contexto particular nos centraremos más en las alternativas más "livianas" que se incluyen como parte de aplicaciones con otra finalidad, y que tienen como objetivo completar la funciones y posibilidades de esa función principal. Aún así, es importante mencionar esas aplicaciones centradas únicamente en la mensajería, puesto que nos servirán como grandes ejemplos a muchos niveles: diseño, implementación, uso, funciones, etc.

Dentro del sentido que estamos buscando, mensajería como complemento de cualquier aplicación, encontramos numerosos ejemplos, también ampliamente conocidos, y de muy diverso tipo: juegos, como Apalabrados, redes sociales, como Facebook, aplicaciones de videollamada, como Skype.

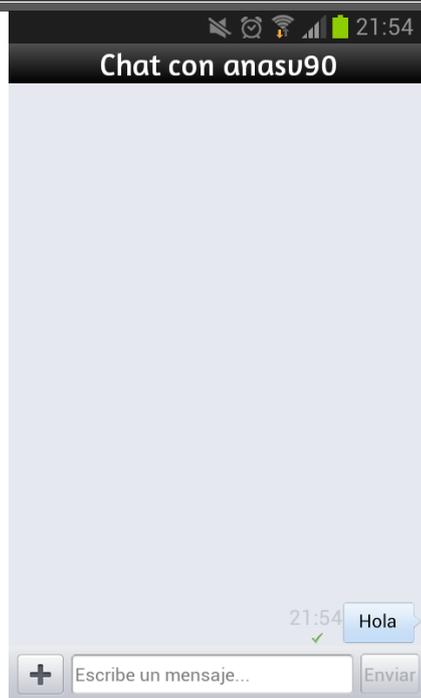


Figura 4 . Ejemplo de chat entre usuarios del juego Apalabrados

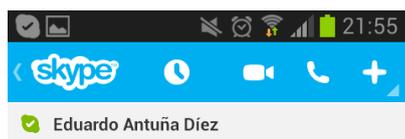


Figura 5 . Ejemplo del chat entre usuarios de la aplicación móvil de Skype

Esto nos muestra como existen numerosas aplicaciones, todas ellas con distintos objetivos concretos, pero que consideran de utilidad añadir la posibilidad de que sus usuarios intercambien mensajes con sus smartphones, e incluso, combinando estas plataformas con otras distintas en las que también disponen de soluciones de su producto (como Skype o Facebook).

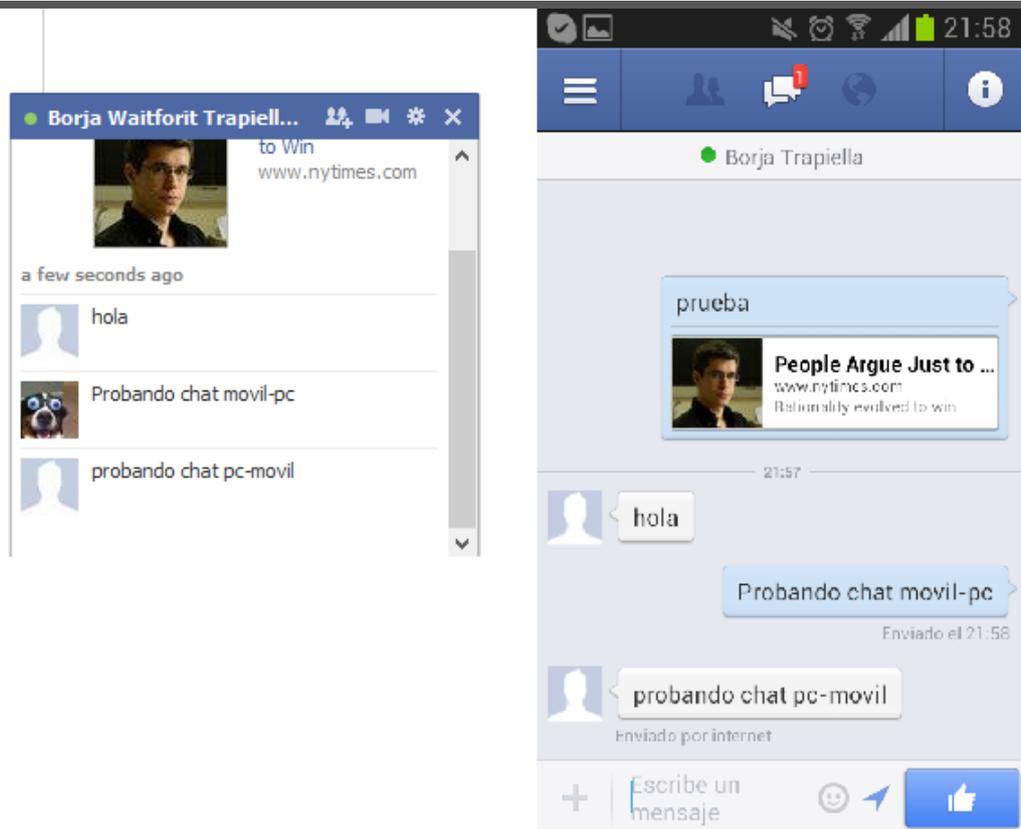


Figura 6 . Ejemplo de chat entre plataformas (PC - Móvil)

Todo esto nos muestra el amplio abanico de posibilidades que existe a la hora de incorporar un sistema de mensajería dentro del ecosistema de las aplicaciones móviles, justificando de esta forma el desarrollo de un sistema de este tipo, con el objetivo de poder ofrecerlo como complemento a cualquiera de nuestros clientes potenciales, independientemente del tipo de aplicación que les pueda interesar.

2.2 Objetivos del Proyecto

- Desarrollar una aplicación móvil que reproduzca animaciones y vídeos de manera fluida
- Conseguir un rendimiento similar en distintas gamas de teléfonos
- Generar vídeos en base a fotogramas y audio dentro del propio teléfono
- Grabar y modificar datos de audio dentro del propio teléfono
- Investigar y desarrollar una alternativa viable para un sistema de mensajería instantánea entre teléfonos y otros dispositivos
- Utilizar el sistema anterior para comunicar un Smartphone con un cliente Web en un PC
- Combinar el sistema de animaciones con el de mensajería, para permitir una nueva forma de compartir contenido entre los usuarios de la misma aplicación
- Implementar mecanismos de monetización de la aplicación desde un punto de vista publicitario

2.3 Estudio de la Situación Actual

En este apartado vamos a estudiar las distintas alternativas que podemos encontrar frente a la solución que se presenta en el proyecto. Al igual que en otros apartados de esta documentación, abordaremos de manera independiente lo que tiene que ver por un lado con el sistema de animaciones, y por otro con el protocolo de mensajería implementado.

2.3.1 Sistema de animaciones

Esta parte del proyecto es la que menos abierta está a evaluar posibilidades, puesto que no existen distintas opciones a la hora de desarrollarlo (salvo más allá de hacerlo para Android o iOS, algo que no evaluaremos como parte de este proyecto puesto que la versión para iOS ya se encuentra desarrollada de forma independiente a la presente).

La única alternativa sobre la que poder optar a la hora de desarrollar este sistema, fue la posibilidad de tratar de recrear fielmente la animación del personaje cuando se unifica con el audio grabado por el usuario (es decir, que los movimientos del personaje estén acompañados con el sonido que se reproduce), o por otro lado hacerlo de forma aleatoria.

En algunas de las aplicaciones similares “conocidas” que hemos mencionado anteriormente, se utilizan ambos sistemas, aunque principalmente tiene mucho que ver con el tipo de diseños empleados en cada una.

A efectos finales, no existe ninguna ventaja o inconveniente para utilizar uno u otro, más allá de que siempre parecerá algo más elaborado si se opta por la opción de realizar un análisis preciso y tratar de reproducirlo fielmente, por encima de una selección aleatoria de imágenes que no coincidirán con el resultado real.

Por este motivo, además de porque para nuestros diseños era la mejor opción, en nuestro proyecto optamos por la opción de analizar el audio de forma precisa y tratar de reproducir las oscilaciones de la amplitud sonora a la hora de generar las animaciones y videos posteriores.

2.3.2 Protocolo de mensajería

Esta es la parte del proyecto en la que realizaremos una evaluación más pormenorizada de las distintas alternativas de las que disponemos para implementar este sistema de mensajería.

Antes de empezar, es importante recordar que buscamos un sistema que nos permita comunicar, no solo teléfonos entre sí, sino también otras combinaciones más heterogéneas, como puede ser un móvil con un cliente web en un pc, o móviles de diferentes tipos (Android e iOS como principales vertientes).

Teniendo en cuenta la anterior, la elección de un protocolo u otro no dependerá solo de las características del mismo, sino también de las diferentes alternativas que presente a la hora de montar un servidor compatible y de desarrollar clientes para las distintas plataformas en las que estamos interesados.

Siendo esto así, las dos alternativas principales que hemos evaluado han sido: WebSocket y el protocolo XMPP.

2.3.2.1 WebSocket

2.3.2.1.1 Descripción

WebSocket es una tecnología de comunicación bidireccional que utiliza un único socket TCP.

Una de sus principales características es que está diseñado principalmente para utilizarse con navegadores y servidores web, siendo bastante reseñable el hecho de que esté siendo estandarizado por el W3C, y ha sido incluido en la reciente especificación del HTML5, lo que nos puede indicar la importancia que está adquiriendo este sistema.

Otra muestra de la relevancia de esta tecnología es que ha sido integrado en los principales navegadores: Mozilla Firefox, Chrome y Safari.

Aunque a nivel general esta opción nos permita enviar mensajes entre dispositivos (enviar datos en general), lo cierto es que no nos ofrece nada específico para implementar un sistema de mensajería instantánea, además de que las soluciones disponibles tanto para la figura del cliente como para la del servidor también están bastante limitadas en este sentido.

2.3.2.1.2 Ventajas

Las ventajas de la tecnología WebSocket son:

- Amplio reconocimiento en la actualidad en el entorno web
- Está siendo estandarizada por el W3C
- Integración en los principales navegadores

2.3.2.1.3 Inconvenientes

Los inconvenientes de esta tecnología son:

- Como tal no representa un sistema de intercambio de mensajes al estilo que buscamos
- Los clientes y servidores disponibles son bastante limitados para nuestros requisitos
- No hay un especial soporte para los dispositivos móviles

2.3.2.2 XMPP

2.3.2.2.1 Descripción

XMPP son las siglas de Extensible Messaging and Presence Protocol (Protocolo extensible de mensajería y comunicación de presencia en español). Representa un protocolo abierto basado en XML, y originalmente fue ideado para sistemas de mensajería instantánea, intentando trasladar la sencillez y adaptabilidad del XML.

Una de sus principales características es que al no ser propietario, puede ser utilizado libremente en cualquier proyecto, y existen amplias documentaciones del protocolo, así como numerosas soluciones cliente servidor que facilitan esta vía.

Este protocolo es usado por empresas tan conocidas e importantes como Google, Facebook o Tuenti, de manera que nos podemos hacer una idea de la calidad y posibilidades que proporciona el uso de este protocolo.



Figura 7 Logo XMPP

2.3.2.2.2 Ventajas

Las ventajas que nos ofrece el protocolo XMPP son:

- Concebido desde su origen para sistemas de mensajería instantánea
- Protocolo libre y abierto, lo que permite no solo utilizarlo libremente en cualquier proyecto, sino que es compatible con otros sistemas tan conocidos y extendidos como GTalk de Google
- Numerosas opciones de clientes y servidores, en función de lo que se necesite en cada caso
- Satisface nuestras necesidades de utilizarlo en entornos diferentes, tanto en móviles Android, iOS, o navegadores web
- En base a lo anterior, y teniendo en cuenta que llevan usándose tecnologías basadas en este protocolo desde el año 1998, resulta muy fácil encontrar soporte para dudas y problemas cuando nos embarcamos en un proyecto de este tipo

2.3.2.2.3 Desventajas

Las desventajas de este protocolo son:

- No puede manejar datos binarios por basarse enteramente en xml (esto se soluciona por otras vías, por lo que no supone un problema realizar operaciones como enviar ficheros por ejemplo)
- Escalabilidad, aunque es un problema recurrente en este tipo de sistemas de mensajería
- Sobrecarga en el uso de datos de presencia (datos de los usuarios conectados al sistema)

2.3.2.3 Comparativa entre las opciones analizadas

| | WebSocket | XMPP |
|--------------------------|-----------|----------|
| Heterogeneidad | Baja | Muy alta |
| Extensión y uso | Alta | Alta |
| Soluciones disponibles | Media | Muy alta |
| Facilidad de integración | Baja | Alta |

2.3.2.4 Justificación sobre el proyecto

Dado el análisis anterior que hemos realizado sobre las diferentes alternativas de las que disponemos, se puede llegar a una conclusión acerca de que solución se acerca más a nuestras necesidades.

Si atendemos a nuestro principal requisito, que como hemos comentado en varias ocasiones se trata de conseguir un sistema que nos permita comunicar no solo dispositivos móviles, sino también combinarlo con la presencia de clientes web para PCs, la mejor opción se trata del Protocolo XMPP.

La principal ventaja de esta alternativa es que desde el principio está concebida para desarrollar sistemas de mensajería instantánea, independientemente de la plataforma destino. Bajo esta premisa, disponemos de forma nativa de diferentes opciones propias de este tipo de sistemas, como listas de contactos, estados (disponibilidad, conexión, etc), concepto de mensaje como entidad, etc.

Además, la longevidad de esta tecnología, propicia dos elementos fundamentales. Por un lado, la documentación del protocolo, así como las vías para encontrar soporte y ayuda ante las diferentes dificultades que se nos puedan plantear, las cuales son amplias y extensas, facilitando de esta manera el proceso de desarrollo de un proyecto bajo este protocolo. Por otra parte, la cantidad de clientes y servidores de los que disponemos, en función de cuales sean nuestras necesidades, es muy elevada, ofreciéndonos todos ellos (una vez más) diferentes funcionalidades y opciones totalmente relacionadas con los sistemas de mensajería (contactos, envío de archivos, etc).

Capítulo 3. Aspectos Teóricos

En este apartado haremos una definición breve acerca de aquellas herramientas, tecnologías y conceptos que se han utilizado durante el desarrollo del proyecto.

3.1 Android

Android es un sistema operativo para dispositivos móviles basado en Linux, creado originalmente por Android Inc, empresa que sería comprada posteriormente por Google en 2005. Fue presentado en 2007 junto a la fundación Open Handset Alliance, un conjunto de empresas de los sectores software, hardware y telecomunicaciones, que tiene el objetivo de investigar y desarrollar sobre nuevos estándares abiertos para los dispositivos móviles. El primer teléfono que utilizaba Android como sistema operativo fue lanzado finalmente en el año 2008.



Figura 8 Logotipo de Android, conocido como 'Andy'

Los programas o aplicaciones que se desarrollan para este sistema operativo, se programan normalmente empleando el lenguaje de programación Java en combinación con el conjunto de herramientas de desarrollo de Android (conocido como SDK). Existe un mercado oficial para estas aplicaciones gestionado por la propia Google, conocido como "Google Play". Sin embargo, existen otros sitios además del oficial en los que poder adquirir estas aplicaciones, de empresas tan relevantes como Amazon o Samsung. En la actualidad existen más de 700000 aplicaciones disponibles mediante el sitio oficial de Android.

Otra de las particularidades que presenta Android es que está liberado bajo una licencia Apache, una licencia libre y de código abierto. Esto permite que cualquier fabricante pueda, no solo incluirlo como sistema operativo en sus dispositivos, sino también realizar modificaciones sobre él, que se centran en la mayoría de ocasiones en aspectos visuales y de interfaz, aunque existen ejemplos más drásticos como en el caso de los dispositivos distribuidos por Amazon.



Figura 9 Comparativa entre el Android presente en un dispositivo de Amazon, y la versión original sin modificar

La razón por la que hemos desarrollado la aplicación que ocupa el objetivo de este proyecto se basa en que es una de las principales vías (si no la principal) para entrar en el mercado de las aplicaciones móviles, teniendo en cuenta que copa más de un 50% de los teléfonos vendidos a nivel mundial.

3.1.1 Android SDK

El SDK de Android (Software Development Kit) es el conjunto de herramientas de desarrollo de Android que se utiliza en la mayoría de proyectos elaborados para ejecutarse sobre este sistema operativo.

Los componentes que incluye este SDK son: un depurador de código, un simulador de teléfono en el que correr las aplicaciones, ejemplos de código y tutoriales. Además, cada nueva versión del SDK permite utilizar las librerías de cualquier versión de Android desde la primera de ellas, para que se puedan desarrollar aplicaciones con el grado de compatibilidad hacia atrás que los desarrolladores deseen.

En este caso, nosotros hemos puesto el "límite" en la versión 2.2 del sistema operativo, (conocido como Froyo), algo que suele ser habitual en la mayoría de aplicaciones actuales, y que fue lanzada en el año 2010.

3.1.2 Android NDK

El NDK de Android (Native Development Kit) permite cargar en nuestras aplicaciones bibliotecas escritas en código C (u otros lenguajes), y poder utilizar sus funciones desde las funciones nativas de Android y el propio lenguaje de programación Java.

En este proyecto hemos hecho uso del NDK de Android puesto que fue necesario la utilización de determinadas librerías escritas en C que nos permitiesen implementar determinadas funcionalidades no presentes de forma nativa como parte del SDK de Android (generación de vídeos en base a fotogramas y audio, y la modificación de algunas características del audio, como su tono, velocidad, etc).

3.2 JavaCV

JavaCV es una librería para Java que proporciona una serie de “wrappers” o contenedores para poder utilizar las funciones propias de una serie de bibliotecas que son habitualmente utilizadas por los desarrolladores en operaciones relacionadas con el manejo y tratamiento de imágenes por ordenador. Algunas de las bibliotecas para las que proporciona soporte JavaCV son: OpenCV, FFmpeg, libdc1394, PGR FlyCapture, OpenKinect, videoInput, and ARToolKitPlus.

En el caso que nos ocupa, estamos interesados en las funciones relacionadas con la biblioteca FFmpeg.

Cabe destacar que se han realizado alguna serie de modificaciones para permitir la inclusión de una pista de audio en los vídeos generados, puesto que en la última versión disponible para JavaCV aún no se estaba teniendo en cuenta esta posibilidad.

Al ser una librería pensada para ser utilizada en entornos donde se use el lenguaje Java, nos permite incluirlo perfectamente en nuestro proyecto Android y hacer uso de las ventajas que supone para nuestra aplicación.



Figura 10 Logo JavaCV

3.3 Sonic

Sonic es un software gratuito para el tratamiento de datos de audio, que en esencia lo que nos permite es modificar el tono y la velocidad de dicho audio.

Al estar desarrollado en C, nos permite hacer una integración dentro de nuestra aplicación Android utilizando el NDK para ejecutar código nativo.

Se caracteriza por la simplicidad en su planteamiento, lo que deriva también en la simpleza a la hora de usarlo (pocas funciones y muy concretas).

En el concepto específico de nuestra aplicación se integra de una manera muy directa, ya que se han incluido mecanismos para analizar el audio a nivel de byte, algo que encaja perfectamente con la implementación de esta herramienta.

3.4 a/Smack

Smack es una librería de código abierto que permite implementar las funciones necesarias para desarrollar un cliente para un sistema de mensajería XMPP. Está desarrollado en Java, y ofrece una amplia variedad de funcionalidades que nos permiten desarrollar soluciones a todos los niveles, desde las más sencillas hasta las más completas.

El hecho de que esté desarrollado completamente en Java, hace que las posibilidades de hacerlo compatible con Android sean muy elevadas. En este sentido, hemos utilizado en nuestro proyecto la librería aSmack, que es una adaptación de la original para optimizarla y poder manejarla desde un entorno Android, aprovechando precisamente el carácter de código abierto de la librería original.

3.5 Monetización

Monetizar una aplicación móvil hoy en día se está convirtiendo en uno de los principales retos para las empresas que se dedican a desarrollar aplicaciones para estos dispositivos.

Se trata de un mercado muy grande, extremadamente cambiante, y donde la competencia va desde las empresas más grandes (la propia Google), hasta el desarrollador independiente que trabaja por su cuenta.

Teniendo esto en cuenta, a la hora de monetizar una aplicación hay que encontrar el difícil equilibrio entre el todo gratis, frente al todo de pago, de manera que nuestra aplicación sea lo más usable posible.

Esto es de especial importancia en aplicaciones del estilo a la que se presenta en este proyecto, ya que tienen un carácter más liviano, de uso esporádico, frente a lo que pueden ser otras aplicaciones como el Whatsapp, juegos de alto presupuesto, etc. En este sentido es de vital importancia encontrar el camino correcto de la explotación publicitaria, y evitar en lo posible las transacciones monetarias desmesuradas.

En este sentido abarcaremos dos tipos de plataformas o mecanismos publicitarios: banners distribuidos a través de Google AdMob, e interstitials gestionados internamente por la empresa Imagine800.

3.5.1 Google AdMob

AdMob es la plataforma publicitaria gestionada por Google, y por tanto una de las más utilizadas actualmente. Una de sus principales ventajas es su marcado carácter multiplataforma, siendo compatible tanto con sitios web, como con las principales plataformas móviles como son Android e iOS.

En este proyecto utilizaremos el sdk que ponen a disposición de los desarrolladores de Android, y cuya integración nos permitirá mostrar un “banner” publicitario (aleatorio) en la disposición que nosotros dispongamos dentro de la interfaz de nuestra aplicación.

3.5.2 Mecanismos publicitarios propios

Aunque AdMob nos permite incluir publicidad de manera rápida y directa en nuestra aplicación, existen otros mecanismos publicitarios que esa vía no abarca y que podemos abordar de manera independiente utilizando las herramientas internas de la empresa.

Una de las principales ausencias de AdMob es la posibilidad de mostrar interstitials de una forma tan directa como el caso de los banners.

Un interstitial es otro tipo de banner publicitario, pero de mayor tamaño y presencia visual, lo que lo hace más interesante de cara a mostrar publicidad más llamativa y atractiva para los usuarios. Sin embargo, por esta misma característica, no es un objeto publicitario que se pueda mantener constantemente en pantalla (a diferencia del banner tradicional), por lo que habrá que controlar su visionado para que sea siempre de manera esporádica y limitando en lo posible la molestia al usuario.

En este sentido, nuestra aplicación implementará un mecanismo para mostrar este tipo de publicidad casi a pantalla completa, mediante las herramientas de la propia empresa, de manera que seamos nosotros quienes controlemos cuando se distribuye esa publicidad, a que dispositivos, según el tipo de usuario, o incluso de marca y modelo de dispositivo.

3.6 Control estadístico de la aplicación – Google Analytics

Actualmente es algo muy común el hecho de incorporar herramientas o mecanismos que nos permitan controlar el flujo de los eventos de la ejecución de nuestras aplicaciones. Esto nos permite hacernos una idea de como usan los usuarios nuestros sistemas, de cosas que puedan estar fallando, o funcionalidades y elementos que pueden llamar más la atención sobre el resto.

Una de las herramientas más populares para este tipo de tareas, tanto a nivel de aplicaciones web como móvil (iOS y Android), es Google Analytics. Dicha herramienta nos permite hacer una ágil integración del control de eventos en nuestra herramienta, además de obtener otros datos como geolocalización de los accesos, tipos de dispositivos que los realizan, etc.

Además, todo esto es fácilmente accesible desde el correspondiente panel web de la herramienta, la cual incorpora facilidades para analizar los datos como gráficas, calendarios, análisis en tiempo real, etc.



Figura 11 . Logo Google Analytics

3.7 Modum

Modum es el nombre por el que se conoce internamente en la empresa Imagine800 a la plataforma que controla de manera genérica todo lo que tiene que ver con las aplicaciones y proyectos desarrollados internamente.

Esta plataforma integra una serie de herramientas, encabezada por una base de datos MongoDB, que nos permite a los desarrolladores disponer de mecanismos de registro de usuarios, datos de aplicaciones, etc, desde el primer momento en el que comienza un proceso de desarrollo de cualquier proyecto.

Como parte característica de su diseño, hay que destacar que implementa una estructura que diferencia de manera estricta los datos de cada aplicación (llamadas “universo” en la plataforma), de manera que los datos de cada una de ellas solo son visibles por ese “universo”, pero no por los demás.

Todo este conjunto de características facilitan mucho el proceso de desarrollo, dando además libertad en cada caso de ampliar el conjunto de funcionalidades (en servidor) de cada una de ellas según sus propias necesidades.

Otra de las herramientas importantes presentes en Modum es el control de notificaciones para cada plataforma móvil. Disponemos de los mecanismos necesarios para enviar notificaciones a cada dispositivo móvil registrado en nuestras aplicaciones, ya sea utilizando el GCM en Android, o APNS en el caso de iOS.

Este control de notificaciones nos permite, entre otras cosas, poder implementar mecanismos publicitarios propios y controlados internamente, como los que hemos descrito en el apartado anterior, pudiendo controlar aspectos tales como modelo y marca de dispositivo, idioma del terminal, grado de actividad del usuario, etc.

Además, y aunque para el presente proyecto no se han utilizado, la plataforma también ofrece mecanismos para el control y seguimiento de compras dentro de las aplicaciones, control de membresía, etc.

Capítulo 4. Planificación del Proyecto y Resumen de Presupuestos

4.1 Planificación

En esta sección se define la planificación temporal de todas las tareas que compone el proyecto, con una duración total de 70 días.

Atendiendo al diagrama de Gantt incluido en este apartado, puede apreciarse que el desarrollo del proyecto se ha estructurado en diferentes partes, siendo la principal diferencia la distinción entre el cliente (la aplicación móvil) y la parte de servidor.

Centrándonos de manera más específica en la parte de la aplicación móvil, se puede observar claramente la distinción entre las tareas relacionadas con aspectos de la gestión de animaciones, y las tareas de implementación de funcionalidades del sistema de mensajería. Esta distinción es la que nos marcará la pauta, tanto en la organización y consecución de las diferentes tareas de las que consta la planificación, como durante el desarrollo de la presente documentación.

Otro aspecto importante a destacar dentro de la estructura de la planificación, es la evaluación interna de los avances por parte de los directores de proyecto en el lado de la empresa. Esta forma de trabajar nos permitía no solo ir validando todos los avances conseguidos, sino también ejecutar procesos de pruebas intermedios de cada subsistema, llevados a cabo por terceras personas ajenas al desarrollo del sistema (algo de especial relevancia).

En cuanto a la organización general de las tareas, se ha seguido un esquema habitual que alterna tareas consecutivas con algunas otras tareas en paralelo, dependiendo de las relaciones de cada una con las funcionalidades pendientes del resto. Es de especial importancia la tarea número 20, “Diseños de Interfaces”, ya que es una tarea realizada de manera paralela junto con prácticamente el resto de puntos de la planificación. Esto se debe a que en esta tarea se pretende englobar todos los trabajos relacionados con la parte de interfaz de la aplicación, un aspecto altamente susceptible a modificaciones constantes, y que prácticamente se realiza diariamente durante el desarrollo del proyecto.

Por último, comentar que el proceso de desarrollo del proyecto finalizó con un parte de integración completa y una fase de pruebas general de todo el sistema, la cual pudo hacerse de forma rápida dado que como hemos comentado durante todo el proceso de desarrollo se hicieron fases de pruebas de los subsistemas de la aplicación que ayudaron a disminuir la cantidad de errores presentes en el sistema final.

Finalmente, el proceso termino con una evaluación tanto internamente por parte de los directores de la empresa, como externa por parte del director del Trabajo Fin de Master en la Universidad.

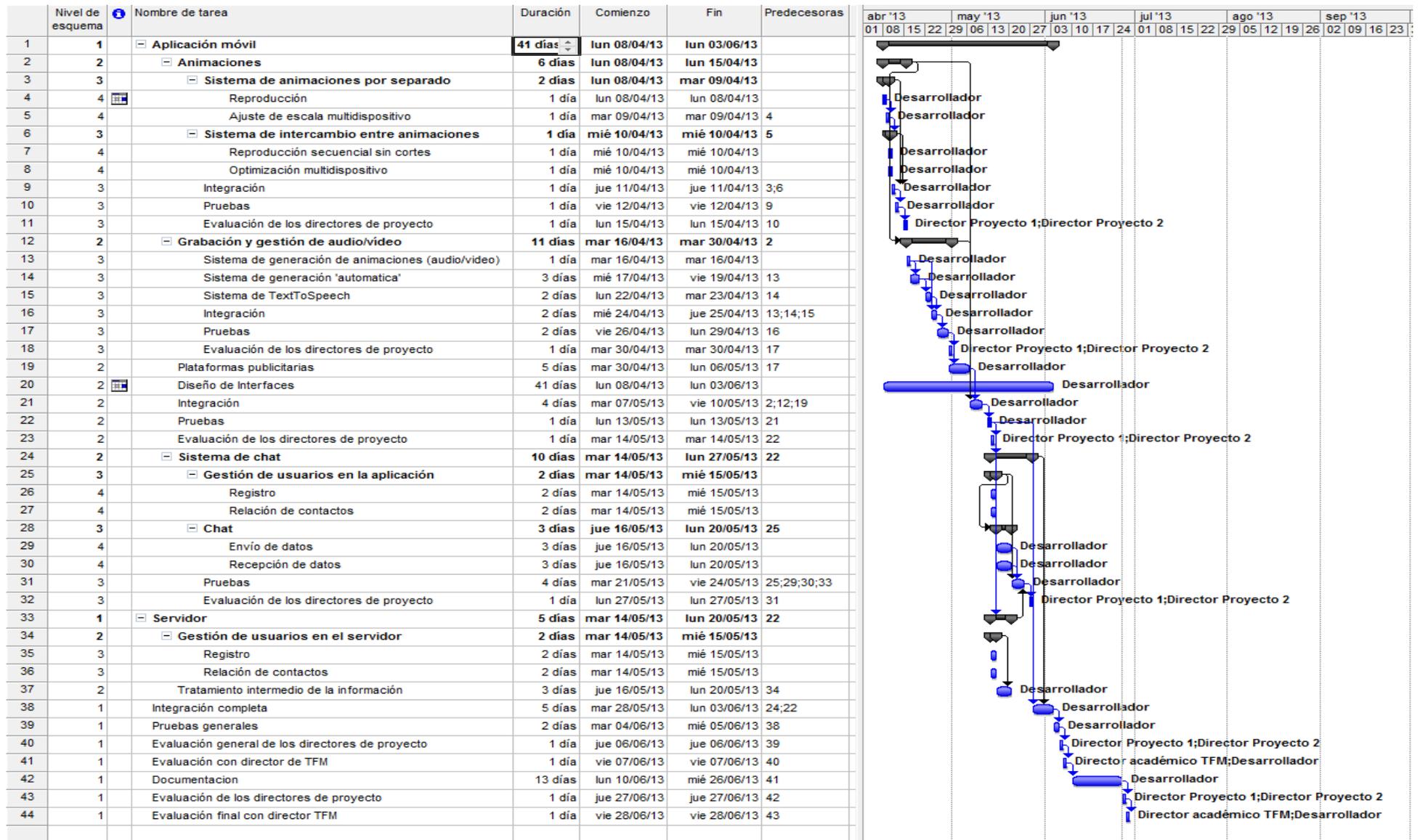


Figura 12 . Diagrama de Gantt de la planificación

| | Nivel de esquema | Nombre de tarea | Duración | Comienzo | Fin | Predecesoras | Demora permisible | Nombres de los recursos |
|----|------------------|--|----------|--------------|--------------|--------------|-------------------|---|
| 1 | 1 | [-] Aplicación móvil | 41 días | lun 08/04/13 | lun 03/06/13 | | 0 días | Desarrollador |
| 2 | 2 | [-] Animaciones | 6 días | lun 08/04/13 | lun 15/04/13 | | 0 días | Desarrollador |
| 3 | 3 | [-] Sistema de animaciones por separado | 2 días | lun 08/04/13 | mar 09/04/13 | | 0 días | Desarrollador |
| 4 | 4 | Reproducción | 1 día | lun 08/04/13 | lun 08/04/13 | | 0 días | Desarrollador |
| 5 | 4 | Ajuste de escala multidispositivo | 1 día | mar 09/04/13 | mar 09/04/13 | 4 | 0 días | Desarrollador |
| 6 | 3 | [-] Sistema de intercambio entre animaciones | 1 día | mié 10/04/13 | mié 10/04/13 | 5 | 0 días | Desarrollador |
| 7 | 4 | Reproducción secuencial sin cortes | 1 día | mié 10/04/13 | mié 10/04/13 | | 0 días | Desarrollador |
| 8 | 4 | Optimización multidispositivo | 1 día | mié 10/04/13 | mié 10/04/13 | | 0 días | Desarrollador |
| 9 | 3 | Integración | 1 día | jue 11/04/13 | jue 11/04/13 | 3;6 | 0 días | Desarrollador |
| 10 | 3 | Pruebas | 1 día | vie 12/04/13 | vie 12/04/13 | 9 | 0 días | Desarrollador |
| 11 | 3 | Evaluación de los directores de proyecto | 1 día | lun 15/04/13 | lun 15/04/13 | 10 | 0 días | Director Proyecto 1;Director Proyecto 2 |
| 12 | 2 | [-] Grabación y gestión de audio/vídeo | 11 días | mar 16/04/13 | mar 30/04/13 | 2 | 0 días | Desarrollador |
| 13 | 3 | Sistema de generación de animaciones (audio/vídeo) | 1 día | mar 16/04/13 | mar 16/04/13 | | 0 días | Desarrollador |
| 14 | 3 | Sistema de generación 'automática' | 3 días | mié 17/04/13 | vie 19/04/13 | 13 | 0 días | Desarrollador |
| 15 | 3 | Sistema de TextToSpeech | 2 días | lun 22/04/13 | mar 23/04/13 | 14 | 0 días | Desarrollador |
| 16 | 3 | Integración | 2 días | mié 24/04/13 | jue 25/04/13 | 13;14;15 | 0 días | Desarrollador |
| 17 | 3 | Pruebas | 2 días | vie 26/04/13 | lun 29/04/13 | 16 | 0 días | Desarrollador |
| 18 | 3 | Evaluación de los directores de proyecto | 1 día | mar 30/04/13 | mar 30/04/13 | 17 | 4 días | Director Proyecto 1;Director Proyecto 2 |
| 19 | 2 | Plataformas publicitarias | 5 días | mar 30/04/13 | lun 06/05/13 | 17 | 0 días | Desarrollador |
| 20 | 2 | [-] Diseño de Interfaces | 41 días | lun 08/04/13 | lun 03/06/13 | | 19 días | Desarrollador |
| 21 | 2 | Integración | 4 días | mar 07/05/13 | vie 10/05/13 | 2;12;19 | 0 días | Desarrollador |
| 22 | 2 | Pruebas | 1 día | lun 13/05/13 | lun 13/05/13 | 21 | 0 días | Desarrollador |
| 23 | 2 | Evaluación de los directores de proyecto | 1 día | mar 14/05/13 | mar 14/05/13 | 22 | 33 días | Director Proyecto 1;Director Proyecto 2 |
| 24 | 2 | [-] Sistema de chat | 10 días | mar 14/05/13 | lun 27/05/13 | 22 | 0 días | Desarrollador |
| 25 | 3 | [-] Gestión de usuarios en la aplicación | 2 días | mar 14/05/13 | mié 15/05/13 | | 0 días | Desarrollador |
| 26 | 4 | Registro | 2 días | mar 14/05/13 | mié 15/05/13 | | 0 días | |
| 27 | 4 | Relación de contactos | 2 días | mar 14/05/13 | mié 15/05/13 | | 0 días | |
| 28 | 3 | [-] Chat | 3 días | jue 16/05/13 | lun 20/05/13 | 25 | 0 días | Desarrollador |
| 29 | 4 | Envío de datos | 3 días | jue 16/05/13 | lun 20/05/13 | | 0 días | Desarrollador |
| 30 | 4 | Recepción de datos | 3 días | jue 16/05/13 | lun 20/05/13 | | 0 días | Desarrollador |
| 31 | 3 | Pruebas | 4 días | mar 21/05/13 | vie 24/05/13 | 25;29;30;33 | 0 días | Desarrollador |
| 32 | 3 | Evaluación de los directores de proyecto | 1 día | lun 27/05/13 | lun 27/05/13 | 31 | 0 días | Director Proyecto 1;Director Proyecto 2 |
| 33 | 1 | [-] Servidor | 5 días | mar 14/05/13 | lun 20/05/13 | 22 | 0 días | Desarrollador |
| 34 | 2 | [-] Gestión de usuarios en el servidor | 2 días | mar 14/05/13 | mié 15/05/13 | | 0 días | Desarrollador |
| 35 | 3 | Registro | 2 días | mar 14/05/13 | mié 15/05/13 | | 0 días | |
| 36 | 3 | Relación de contactos | 2 días | mar 14/05/13 | mié 15/05/13 | | 0 días | |
| 37 | 2 | Tratamiento intermedio de la información | 3 días | jue 16/05/13 | lun 20/05/13 | 34 | 0 días | Desarrollador |
| 38 | 1 | Integración completa | 5 días | mar 28/05/13 | lun 03/06/13 | 24;22 | 0 días | Desarrollador |
| 39 | 1 | Pruebas generales | 2 días | mar 04/06/13 | mié 05/06/13 | 38 | 0 días | Desarrollador |
| 40 | 1 | Evaluación general de los directores de proyecto | 1 día | jue 06/06/13 | jue 06/06/13 | 39 | 0 días | Director Proyecto 1;Director Proyecto 2 |
| 41 | 1 | Evaluación con director de TFM | 1 día | vie 07/06/13 | vie 07/06/13 | 40 | 0 días | Director académico TFM;Desarrollador |
| 42 | 1 | Documentación | 13 días | lun 10/06/13 | mié 26/06/13 | 41 | 0 días | Desarrollador |
| 43 | 1 | Evaluación de los directores de proyecto | 1 día | jue 27/06/13 | jue 27/06/13 | 42 | 0 días | Director Proyecto 1;Director Proyecto 2 |
| 44 | 1 | Evaluación final con director TFM | 1 día | vie 28/06/13 | vie 28/06/13 | 43 | 0 días | Director académico TFM;Desarrollador |

Figura 13 . Detalle de tareas de la planificación

4.2 Resumen del Presupuesto

En este apartado incluiremos un resumen del presupuesto del proyecto, teniendo en cuenta tanto el coste de materiales como el coste de recursos humanos.

| Item | Concepto | TOTAL |
|------|------------------|-------------------|
| 001 | Aplicación Móvil | 1.295,00 € |
| 002 | Integración | 175,00 € |
| 003 | Pruebas | 70,00 € |
| 004 | Documentación | 455,00 € |
| 005 | Hardware | 190,61 € |
| 006 | Software | 152,82 € |
| | SUBTOTAL | 2.338,43 € |

Capítulo 5. Análisis

Este apartado contendrá toda la especificación de requisitos y toda la documentación del análisis de la aplicación, a partir de la cual se elaborará posteriormente el diseño.

5.1 Definición del Sistema

5.1.1 Determinación del Alcance del Sistema

El alcance del sistema se puede identificar atendiendo a sus dos objetivos claros.

Por un lado implementar un sistema de gestión y creación de animaciones de manera dinámica, dentro de un dispositivo móvil. La consecución de este objetivo conllevará la gestión de elementos multimedia como vídeos, imágenes y audio, con tareas como la creación, análisis, edición y almacenamiento de algunos de estos elementos.

Por otro lado, se pretende crear un sistema de chat de texto entre usuarios, que permita intercambiar mensajes tanto entre dispositivos móviles (independientemente de su plataforma) como con otros dispositivos, como puede ser un cliente web en un PC. En este sentido, este sistema se integrará con la parte anterior para darle también sentido dentro del contexto general del proyecto.

Teniendo en cuenta esto, el alcance del sistema contempla:

- Sistema de animaciones
 - Reproducción de vídeo
 - Reproducción de animaciones
 - Grabación de audio
 - Análisis de audio
 - Modificación de audio
 - Almacenamiento de audio
 - Edición de vídeo
 - Almacenamiento de vídeo
 - Publicación de vídeo

- Sistema de chat
 - Gestión de usuarios
 - Intercambio de mensajes
 - Control de conexión
 - Envío de elementos derivados del sistema anterior dentro del contexto del chat
- Sistema general de control de usuarios, notificaciones y publicidad
 - Registro de terminal en el sistema de la empresa
 - Identificación del terminal en el sistema de la empresa
 - Recepción de notificaciones
 - Recepción de notificaciones publicitarias

5.2 Requisitos del Sistema

5.2.1 Obtención de los Requisitos del Sistema

| Código | Nombre Requisito | Descripción del Requisito |
|--------|---|---|
| R1.1 | Reproducir animaciones de vídeo | Se deben reproducir animaciones en formato de vídeo, sin que se aprecie ningún corte entre las imágenes estáticas y la animación, de manera que parezca un proceso de animación fluido |
| R1.2 | Reproducir animaciones en base a fotogramas | Se deben reproducir animaciones a partir de un conjunto de imágenes independientes, sin que se aprecie ningún corte entre las imágenes estáticas y la animación, de manera que parezca un proceso de animación fluido |
| R1.3 | Grabación de audio desde el terminal | Grabar pistas de audio en el teléfono desde la propia aplicación, en base a eventos del usuario |
| R1.4 | Análisis de audio | El audio grabado ha de analizarse para ser capaz de dividirlo en fragmentos que representen un solo fotograma, y analizar la amplitud de onda de ese fotograma y obtener un valor equivalente a la gesticulación del personaje empleado como base de las animaciones |
| R1.5 | Modificación de audio | Se debe poder analizar el audio grabado a nivel de bytes, y a partir de esta información modificar sus características para conseguir modificaciones sonoras (tono, velocidad, etc) |
| R1.6 | Almacenamiento de audio | Se deben implementar mecanismos que nos permitan guardar el audio en un archivo con el formato adecuado, a partir de cualquiera de las fuentes generadas (grabación con el terminal, modificación propia, etc) |
| R1.7 | Edición de vídeo | Se debe implementar un mecanismo que nos permite crear vídeos en base a un conjunto de fotogramas determinados dinámicamente en función de cada ejecución, además de distintas posibilidades de audio, todo ello sincronizado para conseguir la comentada equivalencia entre el audio |

| | | |
|------|--|---|
| | | y los movimientos de la animación del personaje caricaturizado |
| R1.8 | Almacenamiento de vídeo | Al igual que en el caso del audio, las secuencias de vídeo que generemos deberán ser guardadas en el formato adecuado, de manera que puedan ser reproducidas por el usuario independientemente de la aplicación |
| R1.9 | Publicación del vídeo | Se debe permitir al usuario poder publicar el vídeo en diferentes medios, en base a las posibilidades que ofrezca el contexto concreto de su dispositivo |
| R2.1 | Gestión de usuarios | Se debe realizar una gestión adecuada de las personas con los que el usuario del sistema podrá intercambiar mensajes a través de nuestro sistema de chat |
| R2.2 | Intercambio de mensajes | Requisito básico para cualquier tipo de sistema de chat. Este intercambio de mensajes habrá de ser posible entre distintas plataformas como ya se ha comentado |
| R2.3 | Control de conexión | Se deberá realizar un control de la conexión del terminal, para que en el caso de que un mensaje no pueda enviarse por algún tipo de fallo en la red, se envíe automáticamente una vez esta vuelva a estar disponible |
| R2.4 | Envío de elementos generados con el sistema de animaciones | Debe proporcionarse un nexo de unión entre los dos sistemas, que le dé un mayor sentido al proyecto y a la unificación de sus dos partes. Para esto se podrán intercambiar las animaciones generadas con los usuarios disponibles en el chat de la aplicación |
| R3.1 | Registro de terminal en la base de datos | Registrar el terminal en la base de datos del sistema, lo que nos permitirá llevar control estadístico de instalaciones, tipo de dispositivos, versiones instaladas, enviar notificaciones, etc |
| R3.2 | Identificación del terminal en la base de datos | Complemento del requisito anterior, que nos permitirá tener en el dispositivo toda la información generada por nuestro sistema durante el registro, para poder emplearla en caso de ser necesario. |
| R3.3 | Recepción de notificaciones | La aplicación ha de poder recibir notificaciones, tanto estándar, como con metadatos que permitan |

| | | |
|--|--|---|
| | | mostrar en la aplicación contenido de carácter publicitario |
|--|--|---|

- **Requisitos de Usuario:** la aplicación habrá de ser sencilla, y no requerirá ningún tipo de conocimiento específico para manejarla. De todas formas se incluirá una breve sección de ayuda que sirva de guía o para solventar las dudas que puedan aparecer.
- **Requisitos Tecnológicos:** El sistema funcionará en cualquier dispositivo Android a partir de la versión 2.2 (aproximadamente lanzada en el año 2010), al igual que la mayoría de aplicaciones para este sistema operativo móvil.
- **Requisitos de Usabilidad:** Para la grabación de audio será recomendable un ambiente con poco ruido para garantizar los mejores resultados posibles. El sistema de chat requerirá la identificación a través de una red social conocida, de manera que se puedan sincronizar los contactos del usuario como contactos de nuestro chat.
- **Requisitos de Tiempo de Respuesta:** En el caso del sistema de animaciones todo habrá de ir tan fluido como las capacidades técnicas del dispositivo nos lo permitan, y habrá que tomar las medidas necesarias para que esto se cumpla.

5.2.2 Identificación de Actores del Sistema

A la hora de identificar los actores del sistema que estamos desarrollando, nos centraremos como en el resto de casos en los dos sistemas claramente diferenciados de nuestro proyecto. Sin embargo es importante remarcar que a nivel general nos referiremos siempre a un actor general (el usuario del dispositivo o de la aplicación), remarcando alguna diferencia de manera excepcional cuando la funcionalidad o elemento que estemos describiendo así lo requiera.

- Sistema de animaciones
 - Usuario de la aplicación: en este sentido solo hay que tener en cuenta al usuario del dispositivo móvil que está reproduciendo las animaciones y generando las suyas propias
- Sistema de chat: en el caso del sistema de chat tendremos en cuenta los dos lados posibles de un intercambio de mensajes, aunque ambos se puedan identificar con el actor del sistema anterior. De esta forma podremos modelar los diagramas posteriores teniendo en cuenta las dos partes del flujo
 - Usuario emisor del mensaje
 - Usuario receptor del mensaje

5.2.3 Especificación de Casos de Uso

Las distintas funcionalidades del sistema están definidas por los actores que hemos descrito anteriormente.

Cabe destacar que las funcionalidades estrechamente ligadas con la plataforma Modum, no se han descrito como casos de uso, puesto que en ningún momento requieren la interacción del

usuario, y no suponen procesos lo suficientemente “completos” como para ser descritos mediante un caso de uso particular.

A continuación se mostrarán esas funcionalidades y su relación con los actores del sistema.

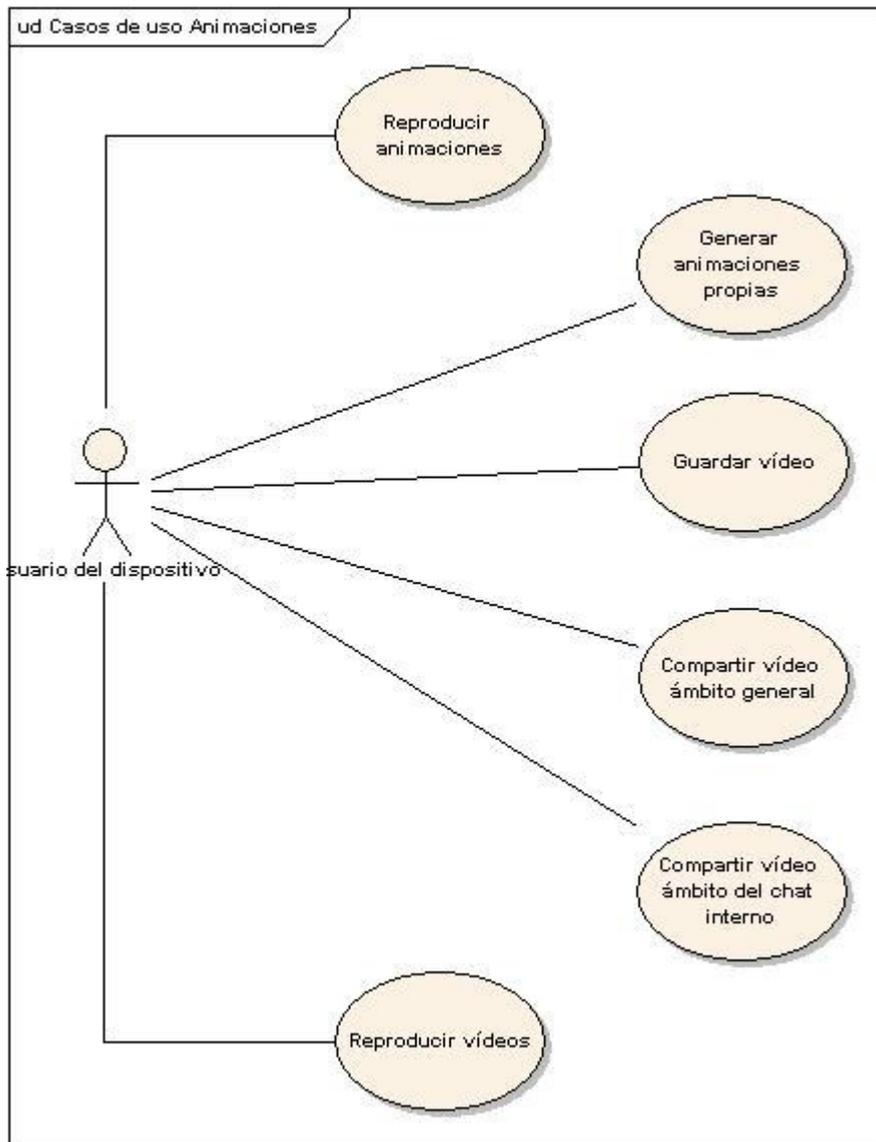


Figura 14 Casos de Uso del Sistema de Animaciones

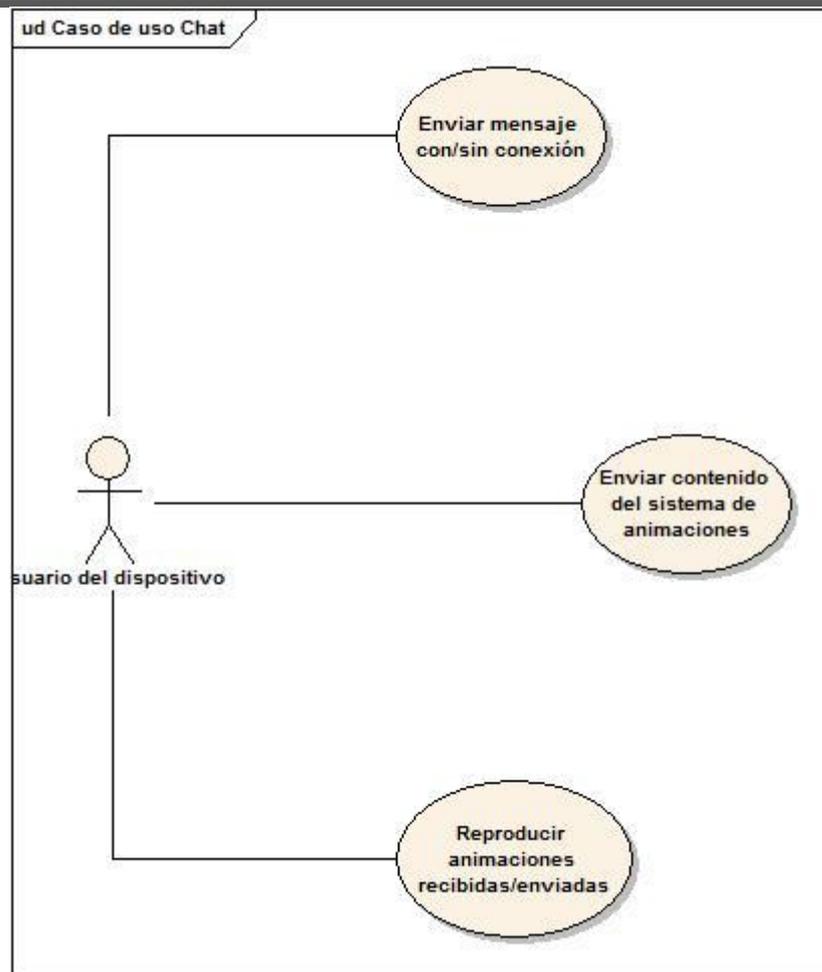


Figura 15 Casos de uso del Sistema de Chat

| |
|--|
| Nombre del Caso de Uso |
| Reproducir animaciones |
| Descripción |
| Se corresponde con el requisito R1.2, y hace referencia a la reproducción de animaciones en base a fotogramas independientes |

| |
|---|
| Nombre del Caso de Uso |
| Reproducir vídeos |
| Descripción |
| Se corresponde con el requisito R1.1, y hace referencia a la reproducción de animaciones a partir de vídeos con la animación completa |

| |
|---|
| <u>Nombre del Caso de Uso</u> |
| Generar propias animaciones |
| Descripción |
| Engloba los requisitos R1.3, R1.4, R1.5, R1.6, R1.7, y hace referencia al proceso completo por el que la aplicación permite al usuario generar animaciones en base al audio grabado |

| |
|---|
| <u>Nombre del Caso de Uso</u> |
| Guardar vídeo |
| Descripción |
| Se corresponde con el requisito R1.8, y hace referencia a la posibilidad de guardar en formato de vídeo la animación generada |

| |
|--|
| <u>Nombre del Caso de Uso</u> |
| Compartir vídeo en ámbito general |
| Descripción |
| Se corresponde con el requisito R1.9, y hace referencia a la posibilidad de publicar el vídeo guardado en diferentes medios, según lo que permita el teléfono de cada usuario en cada caso |

| |
|--|
| <u>Nombre del Caso de Uso</u> |
| Compartir vídeo en el ámbito del chat |
| Descripción |
| Se corresponde con el requisito R2.4, y hace referencia a la funcionalidad que permite enviar el contenido generado por el sistema de animaciones a través del sistema de chat propio de la aplicación |

| |
|--------------------------------------|
| <u>Nombre del Caso de Uso</u> |
| Enviar mensaje con/sin conexión |
| Descripción |

Se corresponde con los requisitos R2.1, R2.2, R2.3 y hace referencia al funcionamiento normal del sistema de mensajería

| Nombre del Caso de Uso |
|-------------------------------|
|-------------------------------|

| |
|---|
| Enviar contenido del sistema de animaciones |
|---|

| Descripción |
|--------------------|
|--------------------|

| |
|---|
| Se corresponde con el requisito R2.4, y hace referencia a la funcionalidad que permite enviar el contenido generado por el sistema de animaciones a través del sistema de chat propio de la aplicación. Es equivalente al caso de uso "Compartir video en el ámbito del chat" |
|---|

| Nombre del Caso de Uso |
|-------------------------------|
|-------------------------------|

| |
|---|
| Reproducir animaciones recibidas/enviadas |
|---|

| Descripción |
|--------------------|
|--------------------|

| |
|--|
| Se corresponde con los requisitos R1.2 Y R2.4, y hace referencia la funcionalidad de reproducir animaciones compartidas dentro de las conversaciones del sistema de mensajería |
|--|

5.3 Identificación de los Subsistemas en la Fase de Análisis

El objetivo de esta sección es analizar el sistema para poder descomponerlo en sistemas más pequeños (subsistemas) que faciliten su posterior análisis.

5.3.1 Descripción de los Subsistemas

A lo largo de la presente documentación, ya hemos realizado en varias ocasiones la distinción entre los sistemas que conforman la elaboración del proyecto.

En este apartado abordaremos de una manera más pormenorizada esta división, identificando claramente los subsistemas con los que estamos trabajando, como se organizan, y que objetivo tienen.

5.3.1.1 Subsistema de animaciones

Contiene todas las funciones que nos permiten llevar a cabo las operaciones necesarias para implementar la parte de la aplicación relacionada con la reproducción y generación de animaciones.

Todas estas funcionalidades pueden verse perfectamente diseccionadas en la identificación de requisitos.

Los elementos que conforman este subsistema se localizan únicamente en la aplicación móvil que se ejecuta en el dispositivo.

5.3.1.2 Subsistema de chat

Este subsistema abarca todos los componentes y funcionalidades relacionados con el sistema de mensajería propio que incluye la aplicación. Este subsistema se centra principalmente en la conexión del dispositivo al servidor central de mensajería, y los consiguientes intercambios de mensajes entre dispositivos (tanto enviar como recibir paquetes).

En el caso de este subsistema sus componentes se dividen en la parte del cliente (que forma parte de la aplicación móvil), y la parte de servidor.

5.3.1.3 Subsistema de registro de dispositivo

Aunque este subsistema no focalice ninguna funcionalidad crítica para los objetivos principales del alcance del proyecto, es de especial importancia nombrarlo ya que es un sistema que se incluye en todas las aplicaciones desarrolladas por la empresa, y que permite llevar un control interno del estado de las aplicaciones distribuidas y los dispositivos/usuarios que las utilizan.

Además este subsistema nos servirá como respaldo en caso de necesitar algún tipo de información extra para complementar alguna de las funcionalidades más propias de los objetivos del proyecto.

Al igual que en el subsistema anterior, en este también debemos hacer una división teniendo en cuenta la parte que se ejecuta dentro de la aplicación móvil, y las gestiones y funcionalidades llevadas a cabo en el servidor correspondiente.

5.3.2 Descripción de los Interfaces entre Subsistemas

Para analizar la interacción entre los subsistemas descritos en el apartado anterior, es de especial importancia la matización que hicimos en cada uno de ellos acerca de donde se situaban el grueso de sus operaciones.

Atendiendo a lo anterior, se puede observar con claridad que la interacción entre los subsistemas se realizará principalmente de dos maneras distintas:

- Internamente dentro de la propia aplicación: en este sentido el subsistema de las animaciones se maneja exclusivamente dentro del dispositivo en el que ejecute la aplicación, pero también puede comunicarse con los “clientes” de los otros dos subsistemas para interactuar con sus funcionalidades (compartir animaciones en el caso del chat, o registrar y obtener información del usuario en el caso del registro del dispositivo)
- Comunicación cliente servidor
 - HTTP: el subsistema de registro de dispositivo utilizará las clásicas peticiones http para ejecutar las operaciones oportunas en el servidor, y cuya objetivo principal será el de manejar la información del usuario y su dispositivo
 - XMPP: como se ha comentado en otros apartados, el protocolo escogido para implementar el subsistema de chat ha sido el protocolo abierto XMPP, y por tanto será el que se utilice para realizar la comunicación entre el cliente del subsistema de chat en la aplicación y el servidor responsable de su funcionalidad.

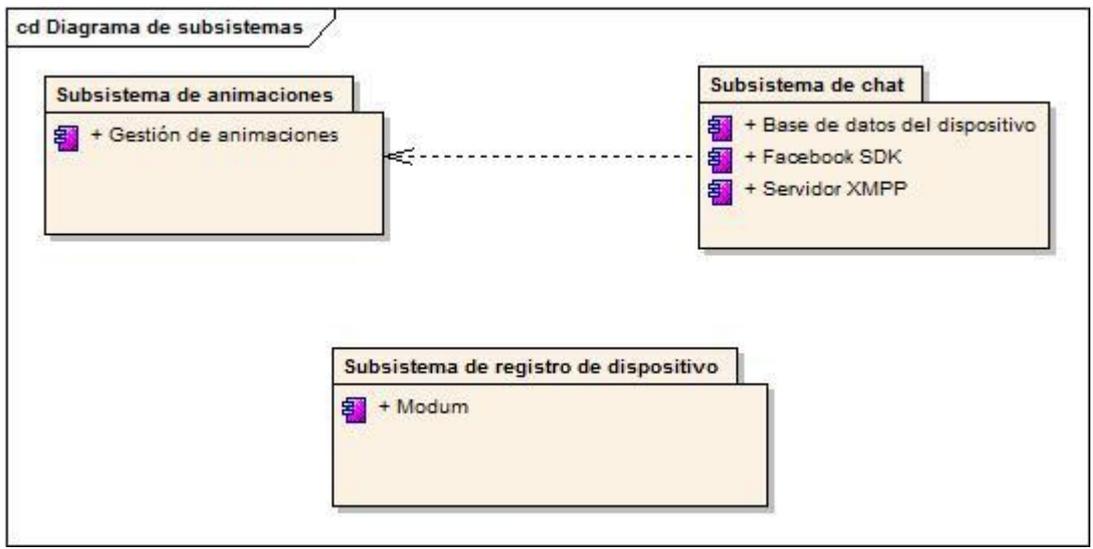


Figura 16 . Diagrama de subsistemas

5.4 Diagrama de Clases Preliminar del Análisis

5.4.1 Diagrama de Clases

A continuación se muestra un diagrama de clases que tiene como objetivo dar una primera idea aproximada de cómo va a ser la estructura final del sistema. Además, para este proyecto es especialmente importante tener en cuenta las particularidades a las que nos vemos sometidos por el uso del sistema operativo Android, que condicionaran fuertemente el diagrama de clases definitivo.

En este diagrama se tiene en cuenta la distinción de subsistemas realizada en apartados anteriores.

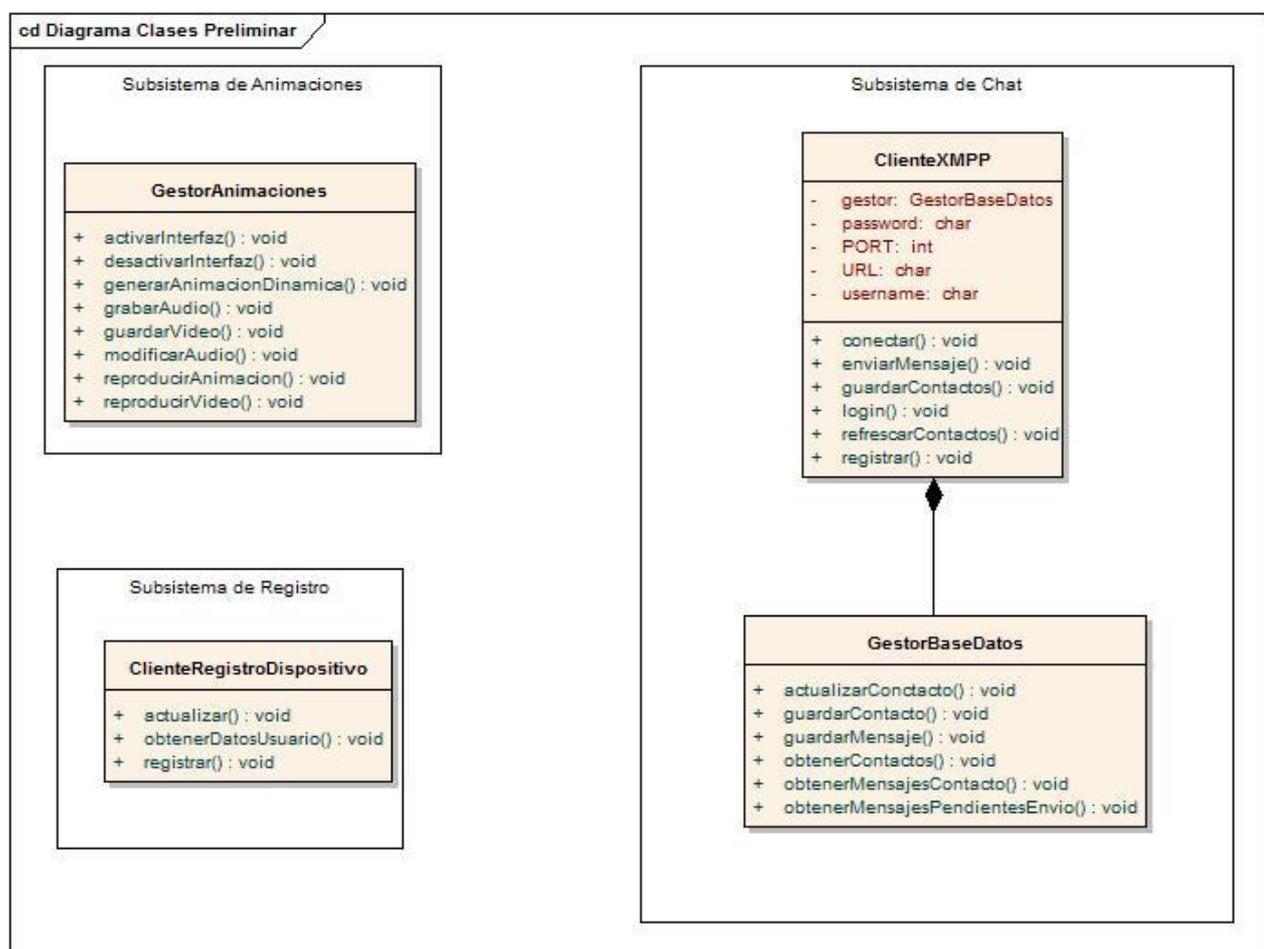


Figura 17 .Diagrama de clases preliminar

5.4.2 Descripción de las Clases

En los siguientes apartados se detallaran las clases que aparecen en el diagrama anterior, teniendo en cuenta una vez más la distinción establecida entre los subsistemas del proyecto. Para cada clase se realizará una descripción de la misma, de sus responsabilidades dentro de la aplicación, y de sus atributos y métodos.

5.4.2.1 Subsistema de animaciones

| |
|--|
| Nombre de la Clase |
| GestorAnimaciones |
| Descripción |
| Es la clase que implementa todas las funcionalidades relacionadas con la reproducción y generación de animaciones dentro de la aplicación. |
| Responsabilidades |
| Ha de manejar todo el contenido multimedia de la aplicación, tanto el propio como el que genera el usuario con el uso. Esto implica controlar tanto elementos multimedia dentro del teléfono, como parte de su hardware, además de elementos de interfaz de la aplicación |
| Métodos Propuestos |
| activarInterfaz: activa todos los elementos de la interfaz con lo que el usuario puede interactuar |
| desactivarInterfaz: desactiva todos los elementos de la interfaz con lo que el usuario puede interactuar |
| generarAnimacionDinamica: controla el proceso por el cual se analiza el audio grabado, se obtienen los valores equivalentes para los frames de la aplicación, y se “juntan” estos frames para obtener la animación a reproducir tras la finalización de todo el proceso |
| grabarAudio: graba el audio del usuario con el micro del dispositivo para su posterior análisis |
| guardarVideo: convierte la animación generada en un archivo de vídeo con el formato adecuado y reproducible de forma externa la aplicación |
| modificarAudio: permite modificar el audio grabado a nivel de byte con el objetivo de modificar sus características sonoras |
| reproducirAnimacion: controla todo el proceso que permite reproducir una secuencia de frames en la aplicación, manteniendo la estabilidad de todos los elementos visuales |
| reproducirVideo: controla todo el proceso que permite reproducir archivos de vídeo en la aplicación, manteniendo la estabilidad de todos los elementos visuales |

5.4.2.2 Subsistema de chat

| |
|---|
| Nombre de la Clase |
| ClienteXMPP |
| Descripción |
| Es la clase que implementa el cliente del sistema de intercambio de mensajes dentro de la aplicación |
| Responsabilidades |
| Su principal responsabilidad es la de permitir tanto enviar como recibir mensajes utilizando el protocolo XMPP y apoyándose en el servidor preparado para tal efecto. Sin embargo, como parte de esta funcionalidad, abarca otros aspectos como la gestión de contactos del usuario con los que podrá comunicarse, y la gestión de los mensajes enviados y recibidos, empleando para tal fin la base de datos interna del dispositivo |
| Atributos Propuestos |
| gestor: es el objeto que se utilizará para ejecutar las funcionalidades relacionadas con la base de datos password: contraseña utilizada para el login del usuario en el servidor XMPP PORT: puerto de conexión XMPP URL: url de conexión XMPP username: nombre de usuario utilizado para el login en el servidor XMPP |
| Métodos Propuestos |
| conectar: método que realiza la conexión del dispositivo al servidor XMPP enviarMensaje: envío de mensaje dentro del sistema de chat implementado guardarContactos: guardar listado de contactos del usuario login: identificación del usuario en el servidor XMPP refrescarContactos: actualización del listad de contactos del usuario registrar: registro del usuario en el servidor XMPP |

| |
|---|
| Nombre de la Clase |
| GestorBaseDatos |
| Descripción |
| Es la clase que implementa todas las operaciones a realizar por la aplicación en la base de datos interna del dispositivo |
| Responsabilidades |
| Las operaciones que implementará serán las encargadas de manejar en la base de datos la información relativa a los contactos del usuario, y a los mensajes que recibe y envía dentro del sistema de chat implementado |
| Métodos Propuestos |
| <p>actualizarContacto: actualiza algún dato de un usuario ya presente en la base de datos</p> <p>guardarContacto: añade un nuevo contacto a la base de datos</p> <p>guardarMensaje: guarda un nuevo mensaje en la base de datos, que puede ser enviado o no enviado</p> <p>obtenerContactos: devuelve un listado de todos los contactos registrados en la base de datos</p> <p>obtenerMensajesPendientesEnvio: devuelve un listado de todos los mensajes que no se llegaron a enviar definitivamente</p> <p>obtenerMensajesContacto: devuelve un listado de todos los mensajes intercambiados con un contacto (tanto los recibidos como los enviados)</p> |

5.4.2.3 Subsistema de registro

| |
|--|
| Nombre de la Clase |
| ClienteRegistroDispositivo |
| Descripción |
| Clase que se encarga de registrar al usuario y los datos de su dispositivo en la base de datos de la empresa |
| Responsabilidades |
| Implementar las funcionalidades que permitan registrar al usuario cuando es nuevo, obtener |

sus datos cuando ya se encuentra en el sistema, y actualizar los que hayan cambiado

Métodos Propuestos

actualizar: comprueba si los datos que tiene el usuario registrados concuerdan con los actuales, y en caso contrario actualiza los necesarios

obtenerDatosUsuario: obtiene toda la información registrada para el usuario y su dispositivo

registrar: registra en el sistema al usuario y a su dispositivo en caso de que sean nuevos en el mismo (primera instalación)

5.5 Análisis de Casos de Uso y Escenarios

5.5.1 Reproducir vídeo

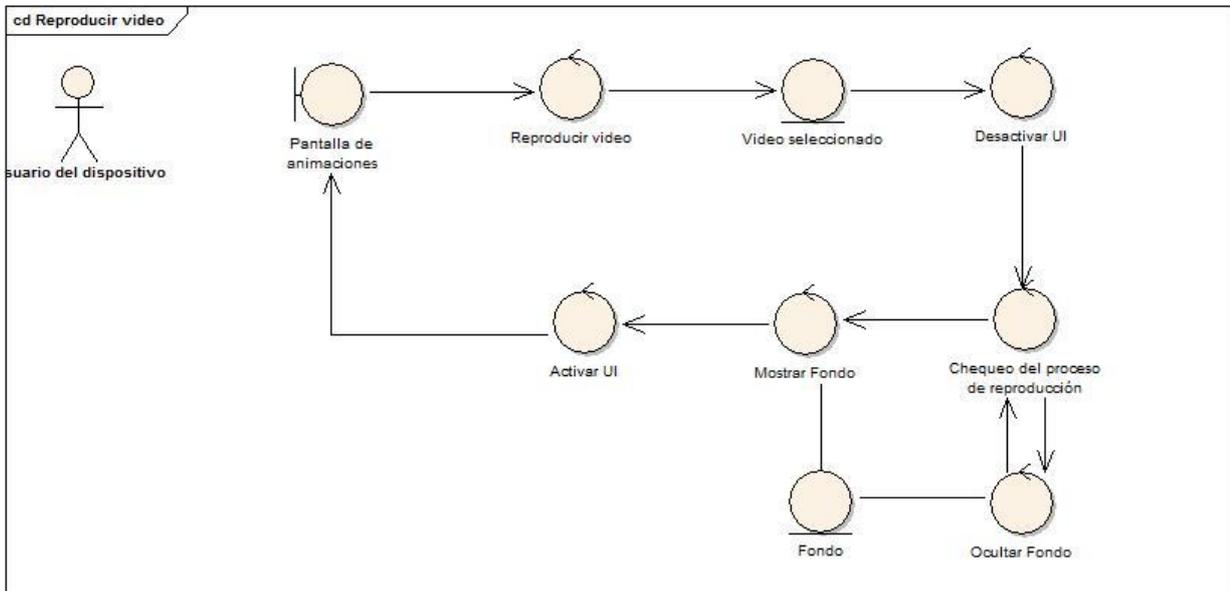


Figura 18 . Caso de uso 'Reproducir video'

| Reproducir vídeo | |
|-----------------------|---|
| Precondiciones | No puede estar reproduciéndose ningún vídeo o animación, generando animaciones, grabando audio, etc. Es decir, la pantalla de animaciones ha de estar en reposo |
| Poscondiciones | El proceso debe finalizar con la aplicación regresando al mismo estado inicial: en reposo, con todos los procesos terminados y la interfaz habilitada |
| Actores | Iniciado por el usuario del dispositivo al hacer click en un botón de animación de vídeo |
| Descripción | <p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la pantalla principal de la aplicación, con los botones correspondientes a las respectivas animaciones 2. Pulsará uno de los botones presentes en la interfaz 3. Comenzará el proceso de reproducción del vídeo, el cual desactivara temporalmente todos los componentes del interfaz, y realizará un análisis del proceso de reproducción para conseguir el mejor resultado posible 4. Si todo se ha realizado correctamente, el vídeo se habrá reproducido de forma fluida, sin parpadeos, y la aplicación retomará su estado inicial, con el personaje estático y los componentes de la interfaz reactivados |

| | |
|--------------------------------------|---|
| Variaciones (escenarios secundarios) | - |
| Excepciones | <ul style="list-style-type: none"> ○ El dispositivo no es compatible con el formato de vídeo: la reproducción no se puede llevar a cabo porque el móvil no es compatible con su formato |
| Notas | - |

5.5.2 Reproducir animación

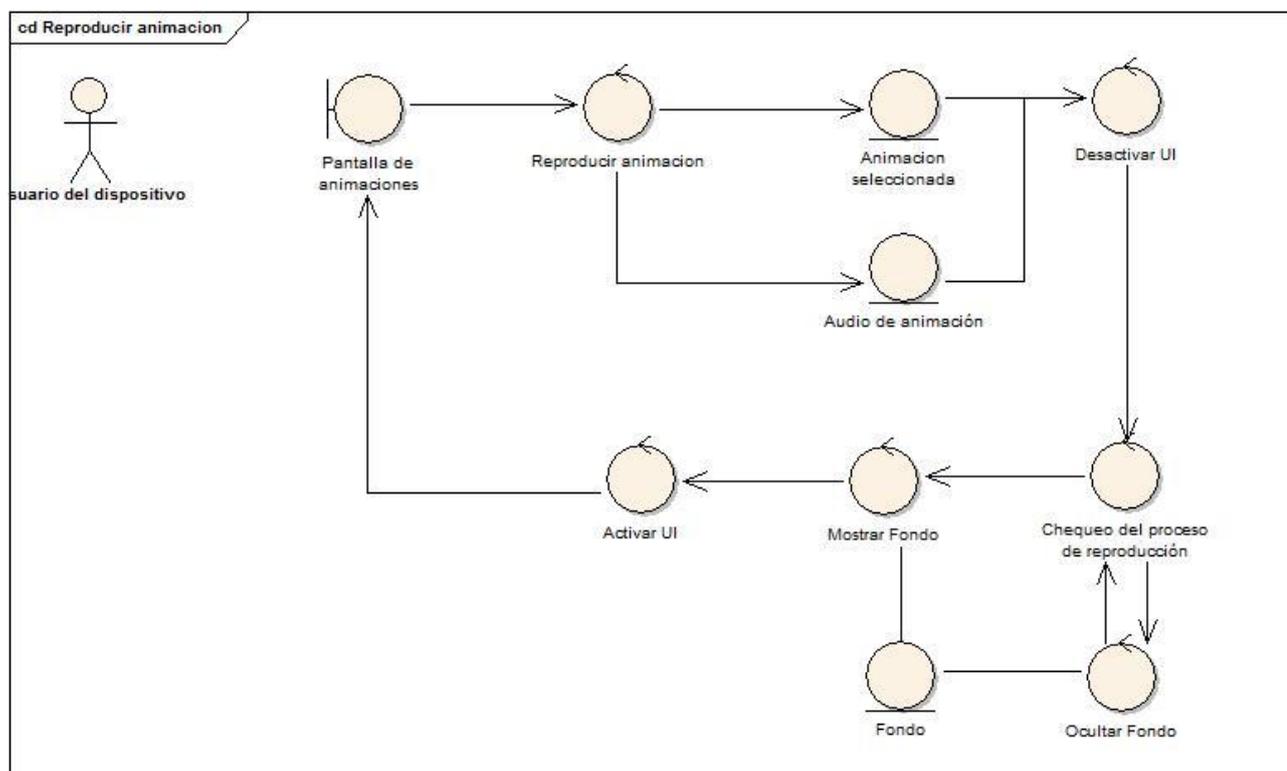


Figura 19 . Caso de uso 'Reproducir animación'

| Reproducir animación | |
|-----------------------|---|
| Precondiciones | No puede estar reproduciéndose ningún vídeo o animación, generando animaciones, grabando audio, etc. Es decir, la pantalla de animaciones ha de estar en reposo |
| Poscondiciones | El proceso debe finalizar con la aplicación regresando al mismo estado |

| | |
|---|---|
| | inicial: en reposo, con todos los procesos terminados y la interfaz habilitada |
| Actores | Iniciado por el usuario del dispositivo al hacer click en la reproducción de una animación |
| Descripción | <p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la pantalla principal de la aplicación, donde el click en ciertas partes de la pantalla lanzan una animación, o bien reproducirá una de sus animaciones posteriormente a grabarla 2. Pulsará el elemento de la interfaz que lanza la reproducción de la animación 3. Comenzará por un lado la reproducción de la secuencia de frames, y por otro la del audio asociado. 4. Si todo se ha realizado correctamente, la animación y el audio se habrán reproducido de forma fluida, sin parpadeos o interrupciones, y la aplicación retomará su estado inicial, con el personaje estático y los componentes de la interfaz reactivados |
| Variaciones (escenarios secundarios) | - |
| Excepciones | - |
| Notas | - |

5.5.3 Generar animaciones propias

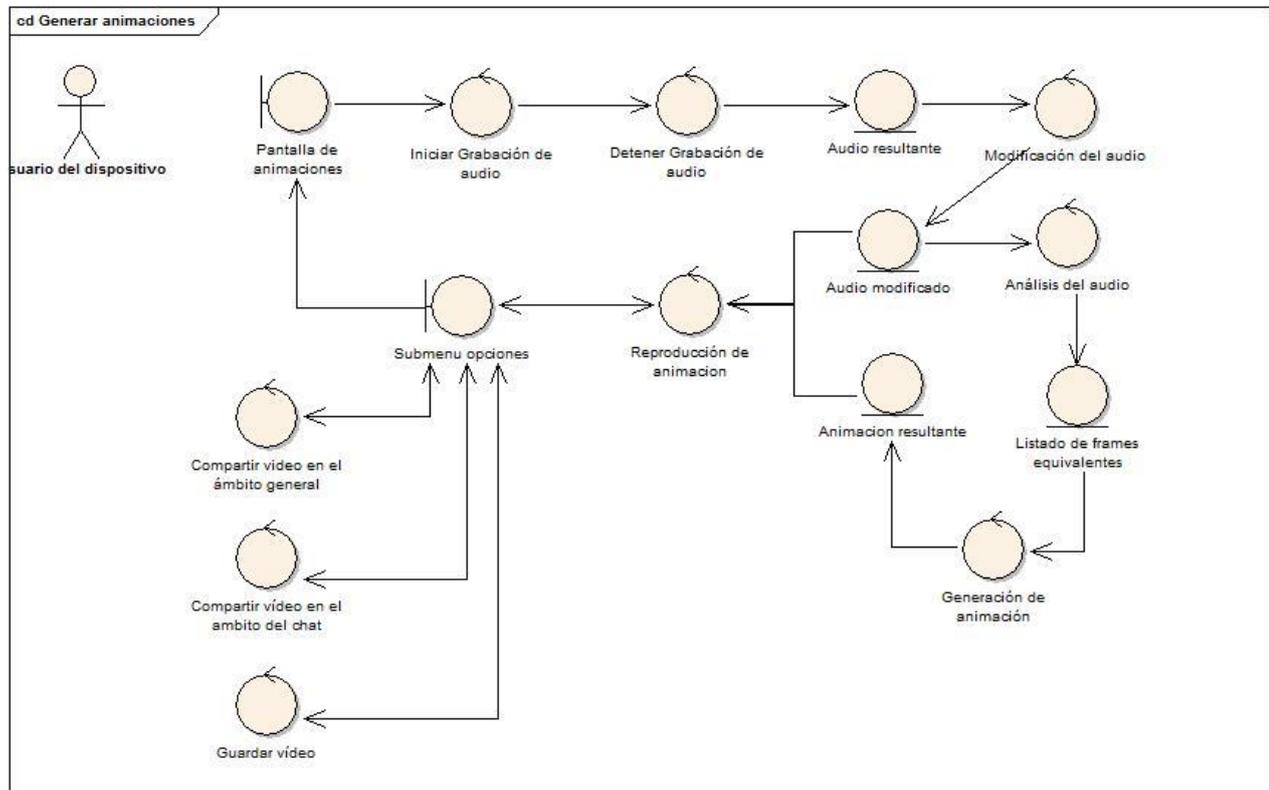


Figura 20 . Caso de uso 'Generar animaciones propias'

| Generar animaciones propias | |
|-----------------------------|---|
| Precondiciones | No puede estar reproduciéndose ningún vídeo o animación, generando animaciones, grabando audio, etc. Es decir, la pantalla de animaciones ha de estar en reposo |
| Poscondiciones | El proceso de grabación inicial debe finalizar, al menos, con la obtención de un audio que poder analizar para obtener el listado de frames equivalentes y poder reproducir la animación de forma automática al menos una vez. El resto de interacciones dependen del usuario y las opciones que ejecute en el submenú desplegado |
| Actores | El usuario del dispositivo al hacer click en el botón de grabación |
| Descripción | El usuario: <ol style="list-style-type: none"> 1. El usuario se encuentra en la pantalla inicial de la aplicación 2. Hace click en el botón de grabación y se inicia la generación del audio. Se muestra un cambio en la interfaz que así lo indica |

| | |
|---|--|
| | <ol style="list-style-type: none"> 3. Vuelve a hacer click en el botón de grabación. Se cambia la interfaz para indicar que se ha detenido la grabación 4. Una vez se analiza el audio y se genera la animación, esta se reproduce automáticamente. 5. Al finalizar se despliega el submenú de opciones 6. Una vez descartado el submenú anterior se vuelve a la pantalla origina, con la aplicación en reposo |
| Variaciones (escenarios secundarios) | <ul style="list-style-type: none"> • Escenario Alternativo 1: La grabación alcanza los 45 segundos <ul style="list-style-type: none"> ○ El proceso se detiene automáticamente ○ El proceso prosigue de forma normal desde el punto 4 |
| Excepciones | - |
| Notas | - |

5.5.4 Guardar vídeo

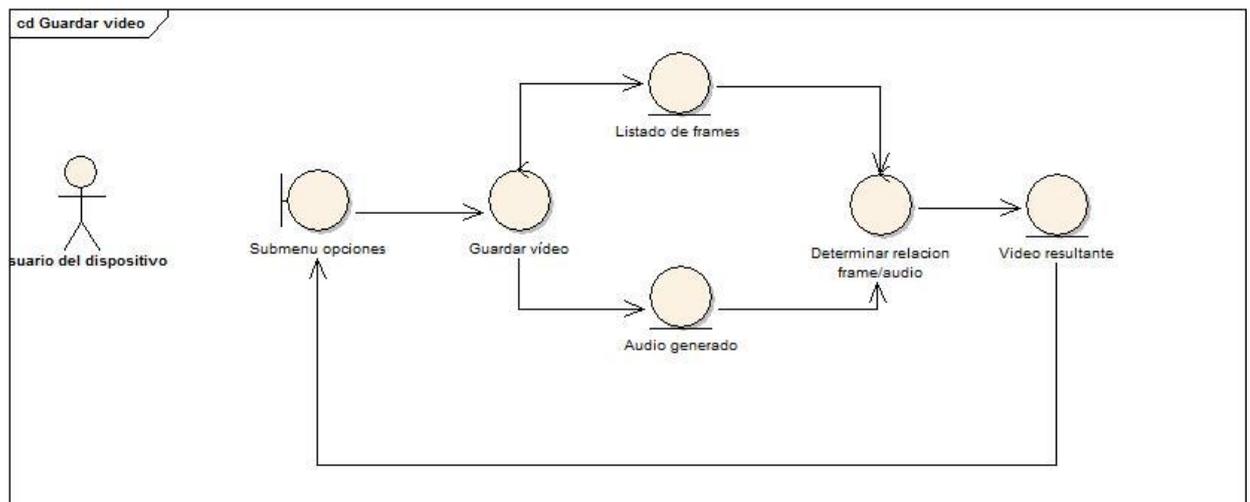


Figura 21 . Caso de uso 'Guardar vídeo'

| Guardar vídeo | |
|---|---|
| Precondiciones | Se debe estar presente en el submenú de opciones posterior a realizar una generación de animación propia, por lo que ya disponemos de un listado de frames equivalentes (y por tanto de una animación) y de un audio generado por el usuario |
| Poscondiciones | El vídeo se ha generado correctamente y se encuentra guardado en la carpeta destinada para tal efecto. El vídeo ha de ser reproducible por el propio dispositivo al margen de la aplicación |
| Actores | Usuario del dispositivo |
| Descripción | <p>El usuario:</p> <ol style="list-style-type: none"> 1. Se encuentra en el submenú de opciones tras haber generado su propia animación y haber visto la reproducción al menos una vez 2. Hace click en la opción guardar vídeo del submenú 3. Comenzará el proceso que relaciona cada frame con un fragmento concreto del audio, de manera que estén lo más sincronizado posible 4. El vídeo ha guardado en una carpeta que determina la aplicación y se ha vuelta a desplegar el menú de opciones |
| Variaciones (escenarios secundarios) | <ul style="list-style-type: none"> • Escenario Alternativo 1: El usuario decide pulsar en primer lugar la opción que permite compartir el vídeo en el ámbito general del dispositivo |

| | |
|--------------------|---|
| | <ul style="list-style-type: none"> ○ Se detecta esta situación y se lanza el proceso de generación del vídeo. Se notifica visualmente al usuario que el proceso está teniendo lugar ○ Una vez terminado se continua con los pasos propios del proceso de compartir en el ámbito general |
| Excepciones | <ul style="list-style-type: none"> ● El vídeo ya ha sido guardado: se notifica al usuario de que el presente vídeo ya ha sido generado y guardado correctamente en el dispositivo |
| Notas | - |

5.5.5 Compartir vídeo en ámbito general

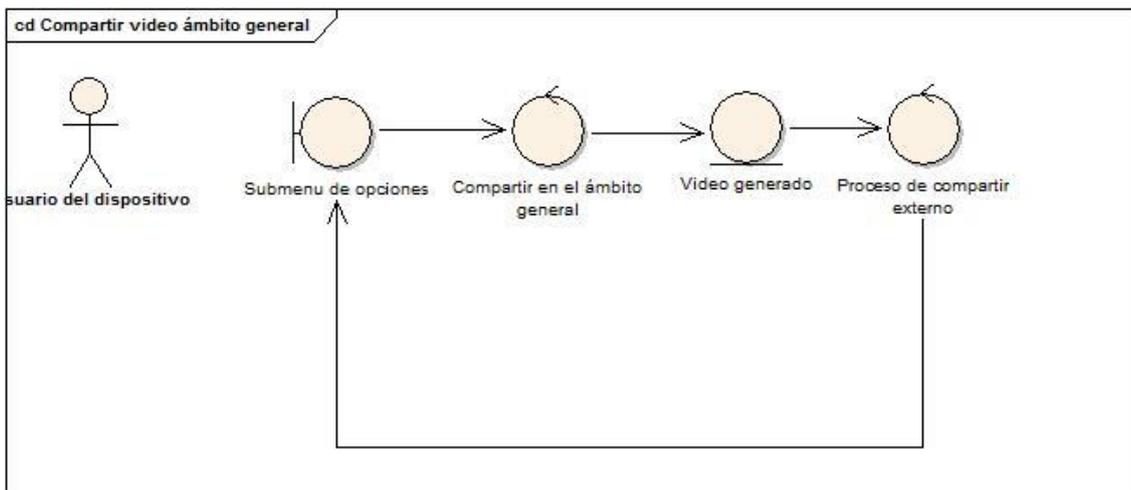


Figura 22 . Caso de uso 'Compartir vídeo en ámbito general'

| Compartir vídeo en ámbito general | |
|--|--|
| Precondiciones | Debe existir un vídeo generado correspondiente a la animación en curso |
| Poscondiciones | Se debe volver a desplegar el submenú de opciones |
| Actores | Usuario del dispositivo |
| Descripción | <p>El usuario:</p> <ol style="list-style-type: none"> 1. Se encuentra en el submenú de opciones tras haber generado su propia animación y haber visto la reproducción al menos una vez. 2. Hace click en la opción de compartir en el ámbito general del submenú |

Análisis | **Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario**

| | |
|---|--|
| | <ol style="list-style-type: none">3. Se lanzará el proceso de compartir interno del propio teléfono, y que es independiente de la aplicación y su funcionamiento4. El vídeo se ha compartido correctamente en el punto de destino, y la aplicación vuelve al punto en el que se encontraba al principio (con el submenú desplegado) |
| Variaciones (escenarios secundarios) | <ul style="list-style-type: none">• Escenario Alternativo 1: El usuario no ha guardado el vídeo antes de intentar compartirlo<ul style="list-style-type: none">○ Notificar el inicio del proceso de generación del vídeo○ Una vez terminado, proseguir con el proceso actual |
| Excepciones | - |
| Notas | - |

5.5.6 Compartir vídeo ámbito del chat interno

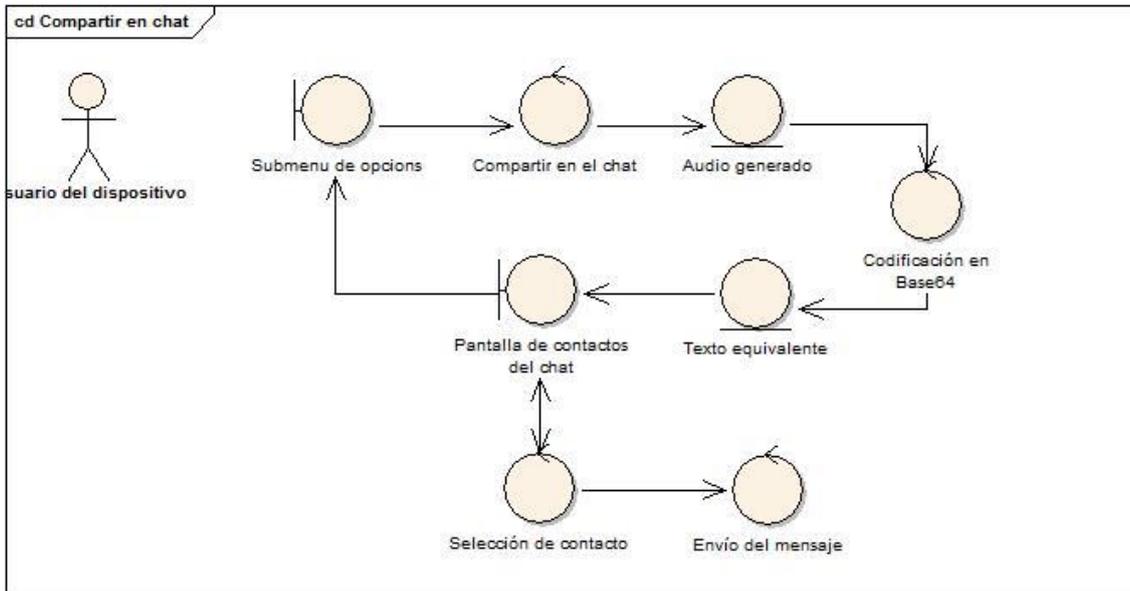


Figura 23 . Caso de uso 'Compartir vídeo en el ámbito del chat interno'

| Compartir vídeo en el ámbito del chat interno | |
|---|--|
| Precondiciones | Se debe haber generado una animación y encontrarse en el submenú de opciones correspondiente desplegado al final del proceso |
| Poscondiciones | Se debe volver a estar en el estado inicial, con el submenú aún desplegado |
| Actores | Usuario del dispositivo |
| Descripción | <p>El usuario:</p> <ol style="list-style-type: none"> 1. Se encuentra en el submenú de opciones desplegado tras realizar una grabación 2. Hace click en la opción de compartir en el chat 3. Se genera el texto equivalente al audio mediante la codificación en Base64 4. Se accede al listado de contactos de la aplicación 5. Se seleccionado un contacto del listado anterior 6. El mensaje es enviado automáticamente y se muestra la conversación completa con ese contacto (con la animación como último mensaje enviado) 7. Se pueden repetir los puntos 4, 5, y 6 sin volver a la pantalla de animaciones con el submenú desplegado 8. Finalmente se vuelve a la pantalla de animaciones, con el submenú de opciones aún desplegado |

| | |
|---|--|
| Variaciones (escenarios secundarios) | <ul style="list-style-type: none"> • Escenario Alternativo 1: El usuario hasta el momento no había usado nunca el sistema de chat. En ese caso debe pasar por el proceso de previo de cargar su listado de contactos |
| Excepciones | - |
| Notas | - |

5.5.7 Enviar mensaje con/sin conexión

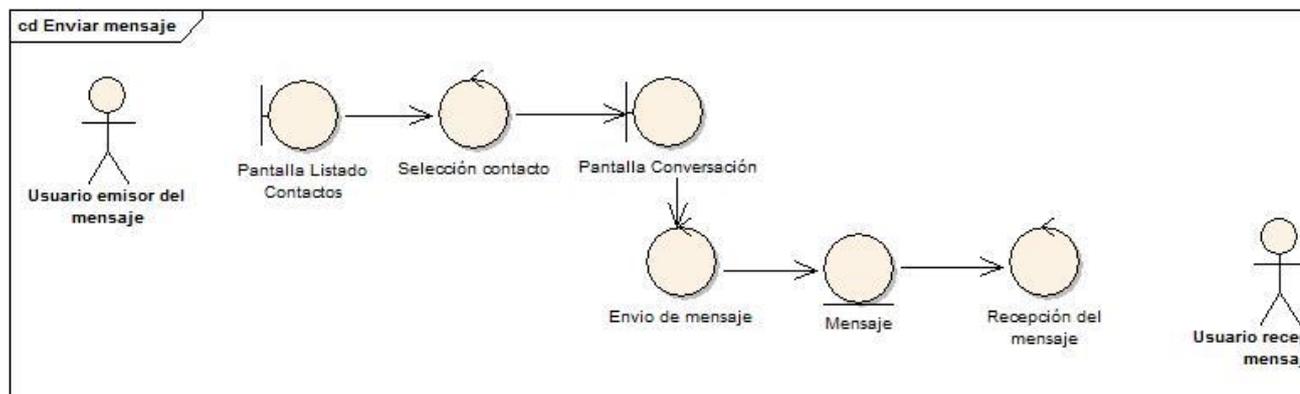


Figura 24 . Caso de uso 'Enviar mensaje con/sin conexión'

| Enviar mensaje con/sin conexión | |
|---------------------------------|---|
| Precondiciones | Se ha de disponer de un listado de contactos disponible |
| Poscondiciones | El nuevo mensaje ha de aparecer el último en el listado de la conversación con ese contacto |
| Actores | Los dos usuarios participantes en la conversación |
| Descripción | El usuario: <ol style="list-style-type: none"> 1. Accederá a la pantalla del listado de contactos 2. En caso de no estar conectado, habrá de conectarse para disponer de su listado de contactos 3. Seleccionará uno de sus contactos 4. Una vez dentro de la conversación podrá escribir el mensaje y enviarlo |

| | |
|---|--|
| | <p>5. Posteriormente ese mensaje aparecerá en el listado de mensajes de esa conversación para el usuario receptor</p> |
| Variaciones (escenarios secundarios) | <ul style="list-style-type: none"> • Escenario Alternativo 1: El usuario receptor no está comprobando la conversación en ese momento. En tal caso verá el mensaje como recibido en el listado de contactos inicial. Una vez acceda a la conversación verá el mensaje de forma normal como parte del listado <ul style="list-style-type: none"> ○ |
| Excepciones | <ul style="list-style-type: none"> • El usuario no se encuentra conectado al servidor XMPP: esto es transparente para el usuario, y el mensaje se listará dentro de la conversación como enviado. El mensaje quedará como pendiente de envío, y se procesará la próxima vez que el usuario se conecte al sistema de chat |
| Notas | - |

5.5.8 Reproducir animaciones enviadas/recibidas

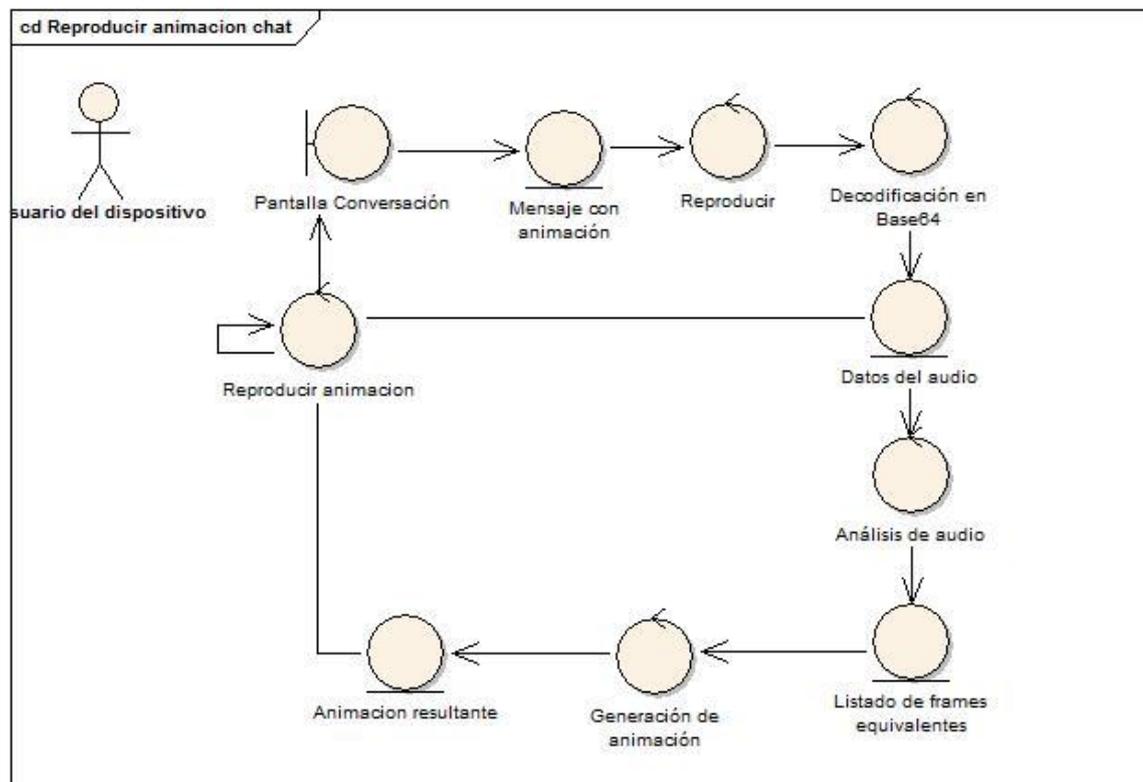


Figura 25 . Caso de uso 'Reproducir animaciones enviadas/recibidas'

| Reproducir animaciones enviadas/recibidas | |
|---|---|
| Precondiciones | Debe de haber alguna animación enviada o recibida en el listado de mensajes de la conversación |
| Poscondiciones | La aplicación ha de seguir funcionando correctamente tras la reproducción de la animación, sobre todo en lo referido a la pantalla principal de las animaciones |
| Actores | Usuario emisor y usuario receptor del mensaje, según en qué lado de la conversación se encuentren. En el diagrama hemos resumido como Usuario del dispositivo a nivel general |
| Descripción | El usuario: <ol style="list-style-type: none"> 1. Accederá a la conversación de alguno de sus contactos 2. Seleccionará algún mensaje de animación, ya sea enviado o recibido 3. El contenido del mensaje será decodificado para obtener el valor del audio original, y se iniciará el proceso de su análisis 4. Una vez analizado se generará la animación de la misma forma que se haría en la pantalla principal |

| | |
|---|--|
| | <p>5. La reproducción de la animación es automática la primera ocasión, y se puede repetir cuantas veces se desee</p> <p>6. El usuario volverá a la pantalla de la conversación con normalidad cuando haya terminado</p> |
| Variaciones (escenarios secundarios) | - |
| Excepciones | - |
| Notas | - |

5.6 Análisis de Interfaces de Usuario

En este apartado trataremos acerca de la interfaz de usuario que presenta el sistema, es decir, como ofreceremos al usuario la manera de interactuar con las funciones que ofrece la aplicación.

5.6.1 Descripción de la Interfaz

El diseño general que se pretende que tenga la aplicación, puede apreciarse en los siguientes esquemas.

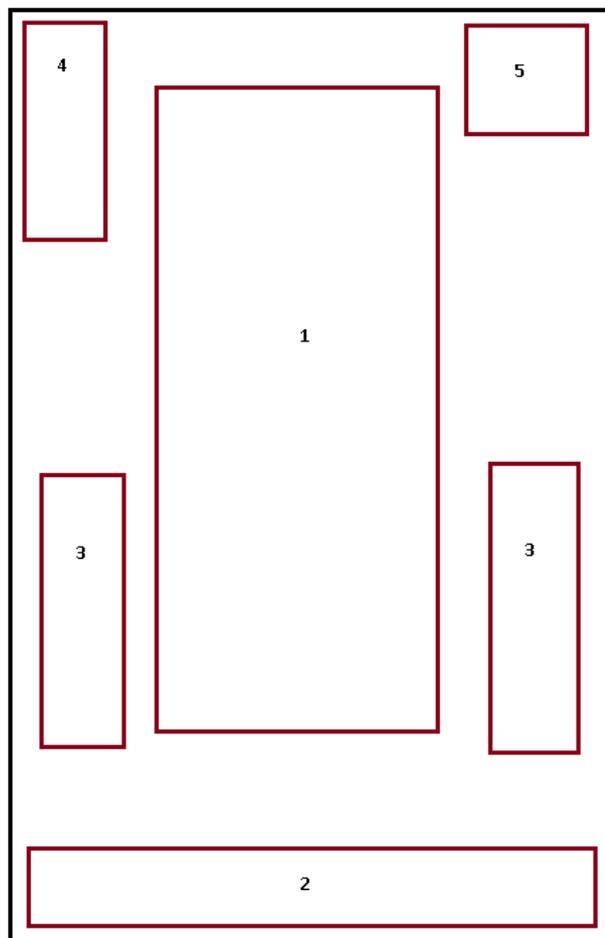


Figura 26 . Esquema de la pantalla principal

- 1: Posición del personaje principal
- 2: Banner publicitario estático
- 3: Botones de las animaciones principales
- 4: Botones de funciones secundarias
- 5: Botón de grabación

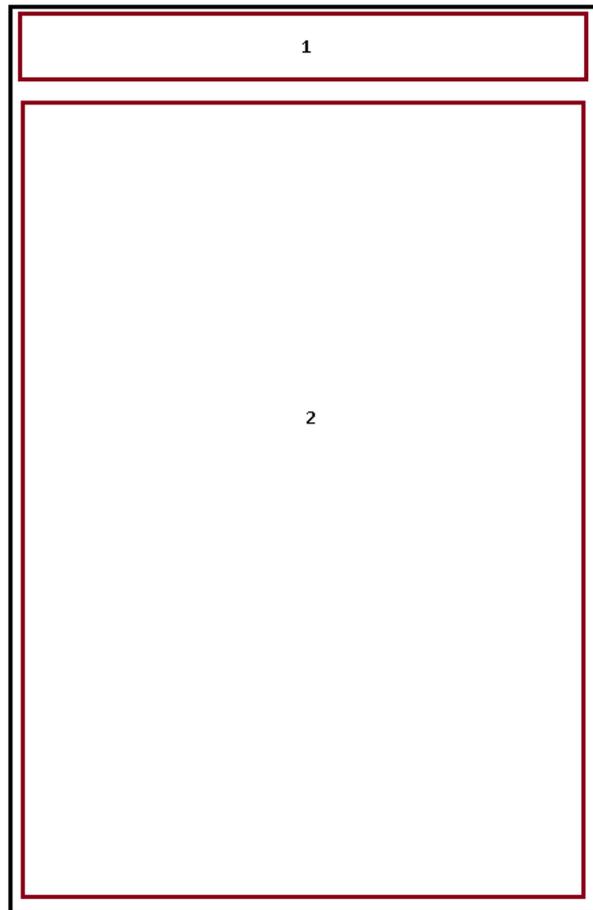


Figura 27 . Esquema de la pantalla del listado de contactos del sistema de mensajería

- 1: Botones de funciones extras del sistema de mensajería
- 2: Listado de contactos del sistema de mensajería

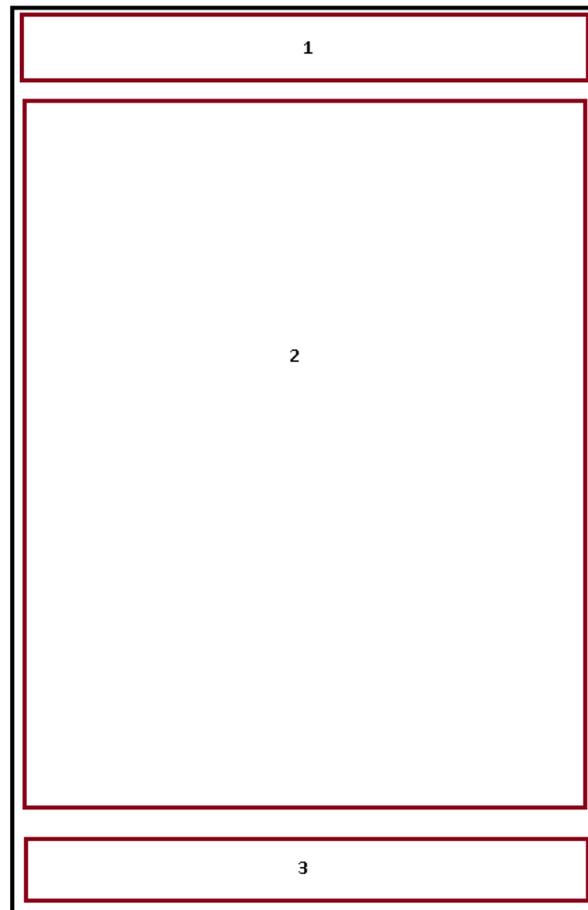


Figura 28 . Esquema de la pantalla de conversación del sistema de mensajería

- 1: Información del contacto con el que se establece la conversación
- 2: Listado de mensajes de la conversación
- 3: Controles para el envío de mensajes

Lo que se pretende a la hora de diseñar e implementar las acciones que ejecutan cada elemento del interfaz, es mantener la mayor simplicidad, remarcando la funcionalidad de cada botón de la mejor manera posible, de manera que cada elemento no arroje ninguna duda acerca de su objetivo.

Salvo por algunos diseños propios de la aplicación, se ha respetado la guía de estilos propia de Android, utilizando en algunos casos los elementos disponibles para funciones de la interfaz claramente reconocibles.

5.6.2 Descripción del Comportamiento de la Interfaz

En general se ha tratado de implementar una interfaz, en la que cada botón desate un evento claramente reconocible, como es en el caso de las animaciones, de la grabación de audio o de

la generación de vídeo, casos en los cuales siempre se aporta algún elemento extra a la interfaz para que quede remarcado el hecho de que la aplicación se encuentra ejecutando algo.

En líneas generales, el comportamiento del interfaz es el esperado en cada caso, y no existe ningún elemento que pueda arrojar dudas acerca de su funcionalidad. Además, para facilitar el uso a los usuarios se han mantenido conceptos propios de otras aplicaciones con funcionalidades similares.

5.6.3 Diagrama de Navegabilidad

En el siguiente diagrama de navegabilidad pueden verse reflejadas todas las pantallas de las que consta la interfaz del sistema, y el tipo de transiciones que pueden existir entre cada una de ellas.

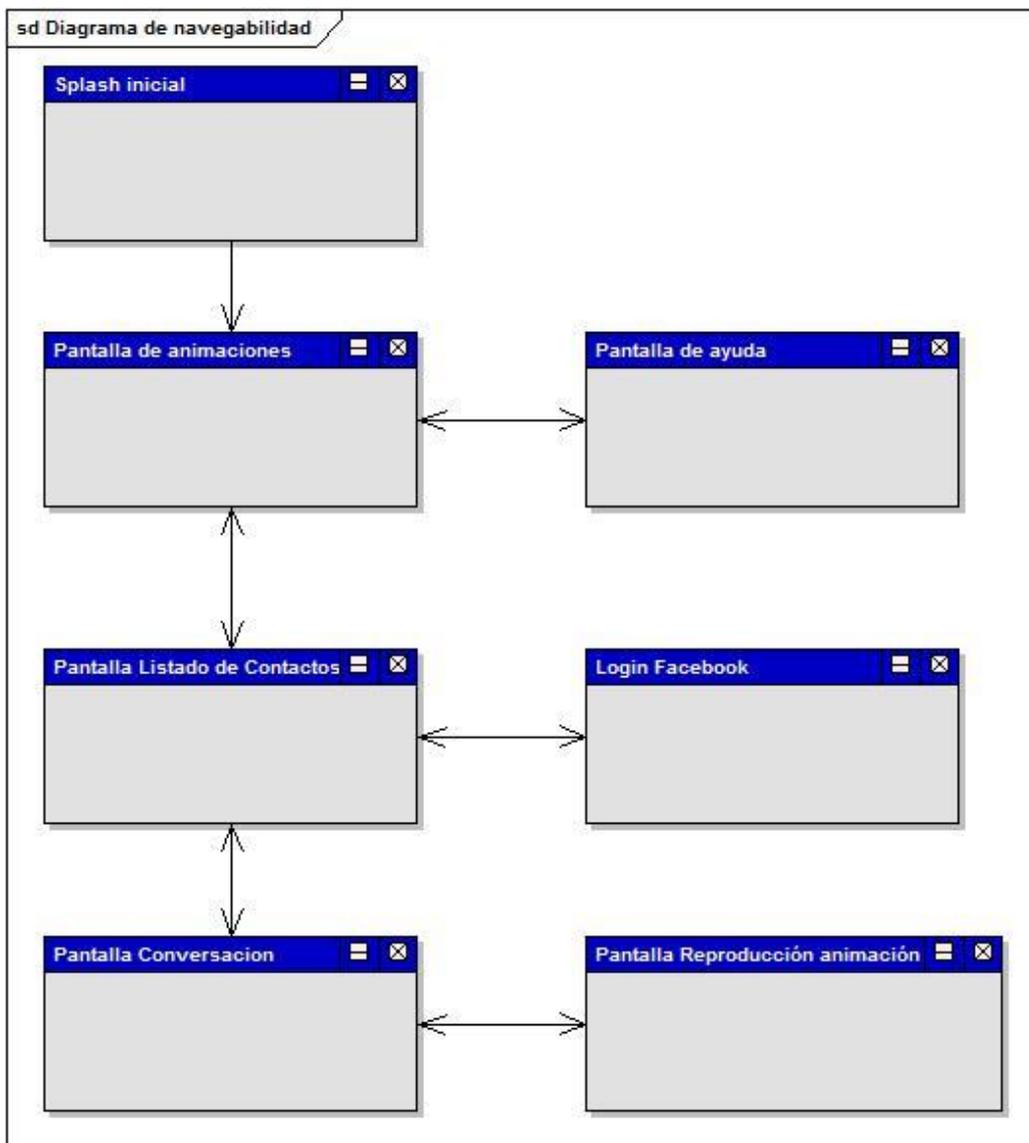


Figura 29 . Diagrama de navegabilidad

5.7 Especificación del Plan de Pruebas

En esta sección crearemos y diseñaremos el plan de pruebas de la aplicación y sus funciones, así como todos los mecanismos que utilizaremos para detectar errores y corregirlos ya en la fase de implementación.

Las pruebas contemplarán aspectos tanto de funcionalidad de la aplicación como de aspectos de los usuarios o clientes de la misma.

Se tendrán en cuenta varios tipos de pruebas:

5.7.1 Pruebas unitarias

Este tipo de pruebas se centran en las partes que se implementan de manera individual, antes de obtener el sistema completo. En esta fase estas partes son: la gestión de animaciones y el sistema de chat

| <i>Caso de Uso : Reproducir vídeos</i> | |
|---|---|
| Prueba | Resultado Esperado |
| Reproducir el vídeo alternando con la imagen estática de base | El vídeo se reproduce de manera fluida y no se aprecian parpadeos al cambiar la vista de la imagen por la del vídeo y viceversa |

| <i>Caso de Uso : Reproducir animaciones</i> | |
|---|---|
| Prueba | Resultado Esperado |
| Reproducir una secuencia de frames como una animación utilizando el sistema propio de Android | En base a unas imágenes definidas, se reproduce la animación correctamente y a la velocidad que esperamos |
| Prueba | Resultado Esperado |
| Reproducir una animación alternando con la imagen estática de base | La animación se reproduce correctamente, sin apreciarse ningún parpadeo al modificar la vista de la imagen por la de la animación y viceversa |

| <i>Caso de Uso : Generar animaciones propias</i> | |
|---|---------------------------|
| Prueba | Resultado Esperado |

| | |
|---|--|
| Grabar audio con el teléfono | Obtenemos un archivo con el audio grabado durante el proceso y en el directorio que hemos indicado |
| Prueba | Resultado Esperado |
| Analizar un archivo de audio | Obtener el audio como un array de bytes, además de otras características del sonido |
| Prueba | Resultado Esperado |
| Modificar las propiedades de un audio | Obtener el mismo audio pero con un tono cambiado (grave o agudo) |
| Prueba | Resultado Esperado |
| Analizar audio para obtener frames equivalentes | Los frames obtenidos ofrecen un resultado que hace que la animación sincronice el habla del personaje de los diseños con el del sonido grabado |

| <i>Caso de Uso : Guardar vídeo</i> | |
|---|--|
| Prueba | Resultado Esperado |
| Generar un archivo de vídeo en base a unas imágenes y un audio localizado por la aplicación | Obtenemos un archivo de audio reproducible por el teléfono de manera independiente a nuestra aplicación |
| Prueba | Resultado Esperado |
| Generar un archivo de vídeo utilizando los frames de la aplicación y el audio modificado | Obtenemos un archivo de vídeo reproducible por el teléfono de manera independiente a nuestra aplicación, y además en el resultado obtenido las imágenes y el audio están correctamente sincronizadas |

| <i>Caso de Uso : Enviar mensaje con/sin conexión</i> | |
|---|---|
| Prueba | Resultado Esperado |
| Enviar mensajes utilizando como servidor el GTalk | La aplicación básica del teléfono nos permite comunicarnos con el chat del gmail en un navegador web o en la aplicación móvil |
| Prueba | Resultado Esperado |
| Enviar mensajes entre terminales utilizando nuestro propio servidor | Los mensajes se intercambian correctamente entre ambos teléfonos |

| Prueba | Resultado Esperado |
|---|--|
| Enviar mensajes entre terminal y un cliente web | Los mensajes se intercambian correctamente entre ambos nodos |

5.7.2 Pruebas de integración del sistema

En este caso, las pruebas se hacen sobre todas las partes integradas del sistema, teniendo que tener en cuenta aspectos como el consumo de memoria del dispositivo, especialmente crítico en aplicaciones que hacen tanto uso de elementos multimedia, y la interacción entre los subsistemas de animaciones y de chat (enviar animaciones, reproducirlas en las conversaciones, etc).

| <i>Caso de Uso : Reproducir animaciones / Reproducir videos</i> | |
|---|--|
| Prueba | Resultado Esperado |
| Reproducir animaciones y vídeos de forma consecutiva e intercalándolos, como parte del funcionamiento normal que se espera de la aplicación | Todas las reproducciones se hacen de forma fluida y sin parpadeos. No hay problemas de memoria |

| <i>Caso de Uso : Generar animaciones propias / Guardar video / Compartir ámbito general /Compartir en chat</i> | |
|---|---|
| Prueba | Resultado Esperado |
| Generar una animación dentro de la aplicación final | El proceso se realiza sin interrupciones y la animación generada se sincroniza correctamente con el audio grabado. |
| Prueba | Resultado Esperado |
| Guardar video de la animación | El vídeo guardado es equivalente a la animación, sobre todo en aspectos de sincronización imagen/audio. Es reproducible directamente desde el teléfono. |
| Prueba | Resultado Esperado |
| Compartir vídeo guardado en el ámbito general | El funcionamiento del sistema de publicación nativo de Android es el esperado y nos permite enviar nuestro vídeo por los diferentes que nuestro terminal nos ofrece |

| Prueba | Resultado Esperado |
|--|--|
| Compartir vídeo guardado en el chat de la aplicación | La codificación se hace de manera correcta y el mensaje llega adecuadamente al terminal receptor |

| <i>Caso de Uso : Enviar mensajes sin/con conexión</i> | |
|--|--|
| Prueba | Resultado Esperado |
| Enviar un mensaje con conexión a uno de nuestros contactos, esperando verse en la pantalla de conversación | Los contactos se obtienen de forma adecuada y el mensaje es recibido adecuadamente. Al volver a la pantalla de contactos, el mensaje aparece como el último de la conversación y además marcado correctamente como leído |
| Prueba | Resultado Esperado |
| Enviar un mensaje con conexión a uno de nuestros contactos, esperando verse en la pantalla de contactos | El mensaje ha de colocarse como último mensaje recibido de esa conversación, y verse claramente destacada la conversación como pendiente de leer |
| Prueba | Resultado Esperado |
| Enviar varios mensajes sin conexión y conectarse posteriormente | El envío de los mensajes se hace sin ningún impedimento. Al volver al conectarse todos los mensajes listados como no enviados son enviados correctamente |

| <i>Caso de Uso : Reproducir animaciones enviadas/recibidas</i> | |
|---|---|
| Prueba | Resultado Esperado |
| Reproducir una animación enviada por nosotros mismos | La animación se reproduce como cabría esperar y de la misma manera que en la pantalla principal de animaciones. Al volver a la pantalla principal, la gestión de animaciones sigue siendo la correcta |
| Prueba | Resultado Esperado |
| Reproducir una animación recibida | La animación se reproduce como cabría esperar y de la misma manera que en la pantalla principal de animaciones. Al volver a la pantalla principal, la gestión de animaciones sigue siendo la correcta |

5.7.3 Pruebas de usabilidad

Estas pruebas están destinadas a garantizar que el sistema sea usable para una amplia gama de usuarios independientemente de su nivel o capacidad, sobre todo teniendo en cuenta la temática de la aplicación que hace que nos encontremos ante un público objetivo de un rango de edad muy amplio.

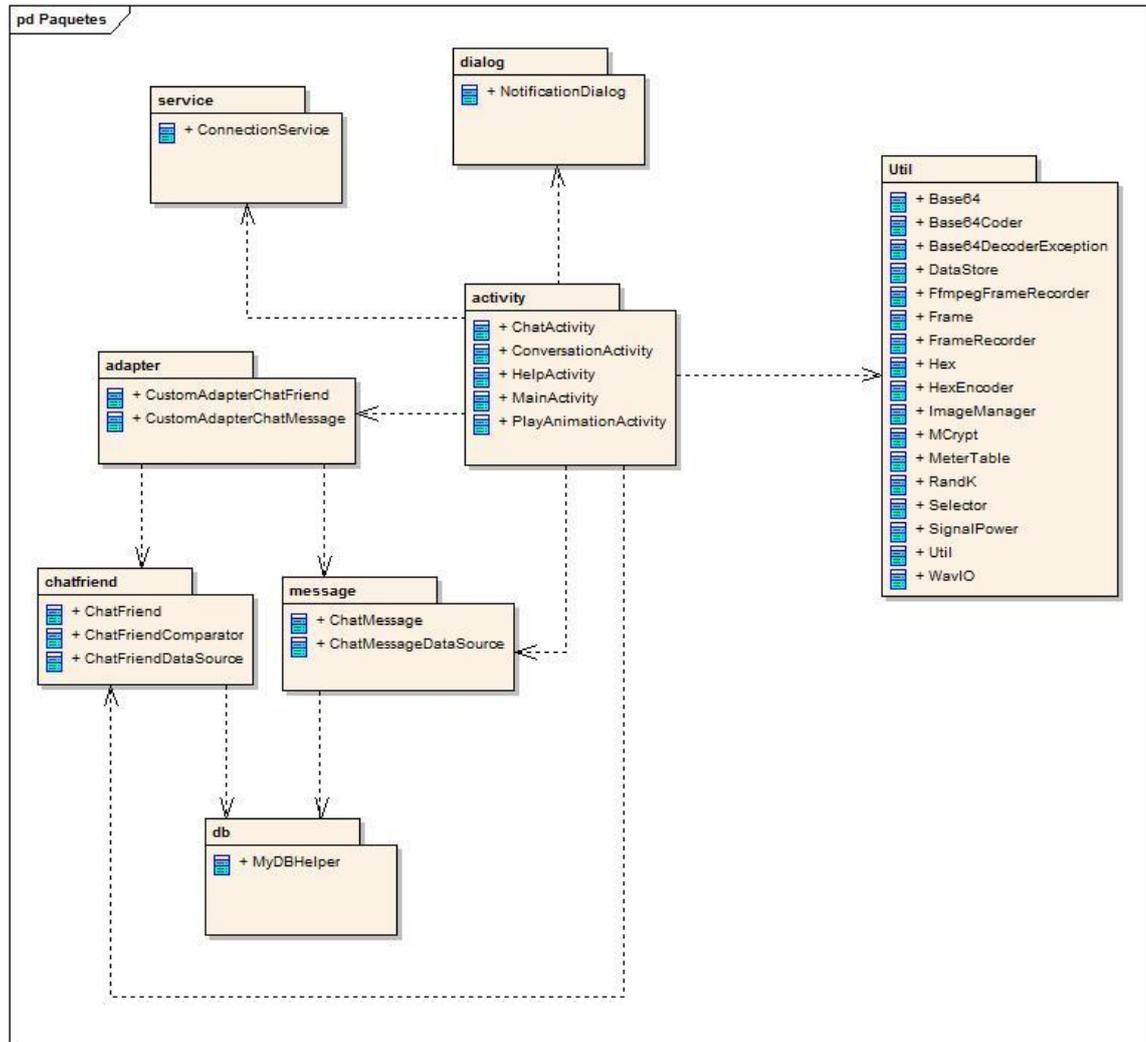
Estas pruebas de usabilidad se centrarán en realizar tests de la aplicación con personas de diferentes características, ya sea atendiendo a su edad, formación, capacidad de uso de nuevas tecnologías, etc.

Además, también tendremos en cuenta como factor para garantizar una buena usabilidad el hecho de que la aplicación se ejecute correctamente en la mayoría de terminales posibles, atendiendo sobre todo a aspectos de rendimiento, y manteniendo siempre presente las limitaciones a las que se ven sometidas normalmente en este ámbito las aplicaciones destinadas a terminales con el sistema operativo Android.

Capítulo 6. Diseño del Sistema

6.1 Arquitectura del Sistema

6.1.1 Diagramas de Paquetes



6.1.1.1 Paquete Activity

Es el paquete que determina la funcionalidad “visible” de la aplicación. Contiene todas las clases “Activity” del proyecto, que en Android habitualmente equivalen a cada pantalla de la aplicación.

Cada una de estas clases implementa todos los comportamientos de la aplicación, como por ejemplo que texto se va a poner en determinado momento, donde, cuando y porque, o el comportamiento de los botones y demás elementos de interfaz.

Como se puede apreciar en el diagrama tiene dependencias con todos los demás paquetes incluidos en el proyecto, los cuales complementan lo implementado en estas Activities.

El contenido concreto del paquete es:

- **ChatActivity:** es la clase que representa la pantalla con el listado de contactos
- **ConversationActivity:** es la clase que representa la pantalla con el listado de mensajes de una conversación
- **HelpActivity:** es la clase que representa la pantalla de ayuda
- **MainActivity:** es la clase que representa la pantalla principal de la aplicación con el sistema de animaciones
- **PlayAnimationActivity:** es la clase que representa la pantalla en la que se reproducen las animaciones enviadas por el chat

6.1.1.2 Paquete útil

Este paquete contiene todas las clases que proporcionan funciones auxiliares o de apoyo para el resto de elementos de la aplicación. En el diagrama lo hemos relacionado solamente con el paquete activity por simplificar, pero podría estar enlazado con todos los de la aplicación.

El contenido de este paquete es:

- **Base64, Base64Coder, Base64DecoderException:** estas 3 clases proporcionan funciones de codificación en base64 para operaciones relacionadas con funciones a MODUM
- **DataStore:** esta clase se utiliza como referencia para almacenar variables de utilidad temporal
- **Frame, FrameRecorder, FfmpegFrameRecorder:** estas clases son una redefinición de unas clases presentes en las librerías de OpenCV, para modificar ciertos aspectos de la implementación original.
- **Hex, HexEncoder:** al igual que las 3 primeras clases de Base64, ofrecen funcionalidades de codificación hexadecimal para funciones relacionadas con MODUM
- **MeterTable, Selector, SignalPower:** estas clases ofrecen funciones para realizar el análisis del audio y su amplitud de onda
- **WavIO:** esta clase ofrece funcionalidades para analizar ficheros de audio en formato wav

6.1.1.3 Paquete dialog

En este paquete contenemos una entidad que nos sirve para formatear los Interstitials de las notificaciones con contenido publicitario.

El contenido de este paquete es:

- **NotificationDialog:** esta clase representa los objetos que mostrar cuando recibimos notificaciones de carácter publicitario en la aplicación.

6.1.1.4 Paquete service

En este paquete localizamos los servicios que se usan dentro de la aplicación. En el caso de este proyecto solo utilizamos uno, que gestiona toda la actividad relacionada con el chat y el protocolo XMPP: conexión, envío y recepción de mensajes, desconexión, login, etc.

El contenido de este paquete es:

- **ConnectionService**: esta es la clase en la que modelamos no solo los métodos relacionados con las funciones XMPP, sino también con el mecanismo adecuado para instanciar el proyecto en segundo plano de manera correcta dentro del sistema de Android

6.1.1.5 Paquete adapter

En Android un adapter se utiliza para mostrar elementos de un listado de una forma particular y definida por nosotros mismos. En nuestro caso particular tenemos dos listados que formatear: el listado de contactos, y el listado de mensajes de cada conversación.

El contenido de este paquete es:

- **CustomAdapterChatFriend**: esta es la clase que implementa la forma en la que se muestra cada elemento del listado de contactos. Se encarga entre otras cosas de mostrar el último mensaje de cada conversación, o determinar si una conversación tiene mensajes no leídos.
- **CustomAdapterChatMessage**: esta clase es otro adapter, pero en este caso para cada mensaje de una conversación del sistema de mensajería. Determina si los mensajes son enviados, recibidos, o si se deben mostrar como una animación a reproducir.

6.1.1.6 Paquete chatfriend

En este paquete se modela la información relacionada con cada contacto del sistema de mensajería, además de los mecanismos para manejar sus datos en la base de datos de la aplicación.

El contenido de este paquete es:

- **ChatFriend**: esta clase determina el contenido y los atributos de cada contacto del chat
- **ChatFriendComparator**: esta clase nos permite determinar el orden en el que queremos mostrar los contactos; en nuestro caso se hace en orden alfabético.
- **ChatFriendDataSource**: esta clase implementa todos los métodos que ejecutan acciones contra la base de datos, siempre relacionadas con información referente a los contactos del sistema de mensajería.

6.1.1.7 Paquete message

Este paquete es idéntico al anterior, pero en lo referente a los mensajes recibidos en el sistema de chat. La diferencia radica principalmente en que en este caso no es necesario determinar un orden particular para los mensajes más allá del orden en el que se reciben.

El contenido de este paquete es:

- `ChatMessage`: modela los atributos de cada mensaje enviado y recibido
- `ChatMessageDataSource`: implementa las funcionalidades que realizan consultas, inserciones y actualizaciones contra la base de datos.

6.1.1.8 Paquete db

En este paquete se encuentran los elementos que nos ayudan a modelar e instanciar la base de datos de nuestra aplicación. Aquí introducimos la información referente a las tablas que queremos utilizar, las columnas, las claves, etc.

El contenido de este paquete es:

- `MyDBHelper`: Es la clase que se encarga de instanciar nuestra base de datos junto con todas sus características. Además, será utilizada por los dos paquetes anteriores para poder lanzar cada llamada a la base de datos (inserts, queries, etc)

6.1.2 Diagramas de Componentes

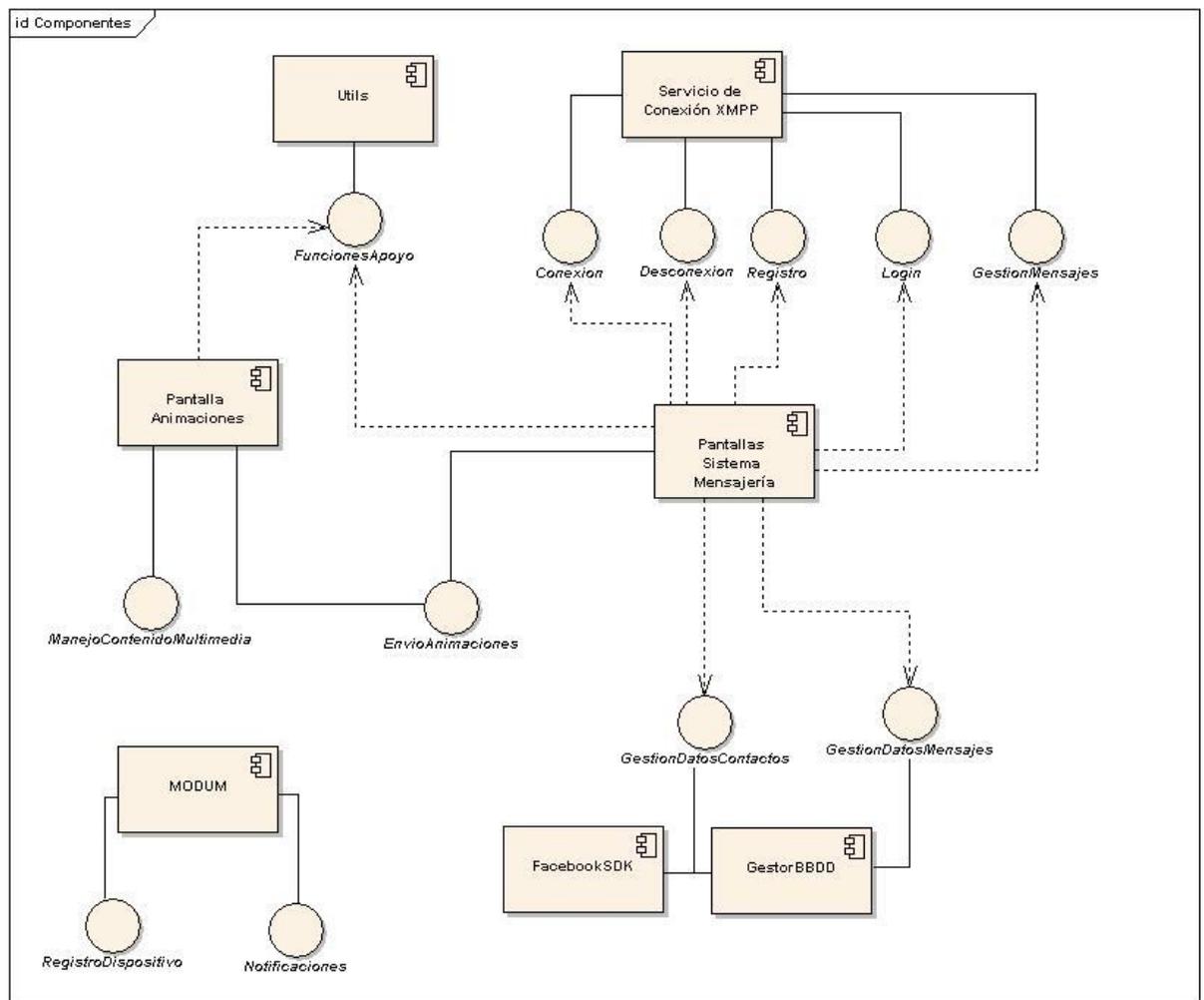


Figura 30 . Diagrama de componentes

6.1.3 Diagramas de Despliegue

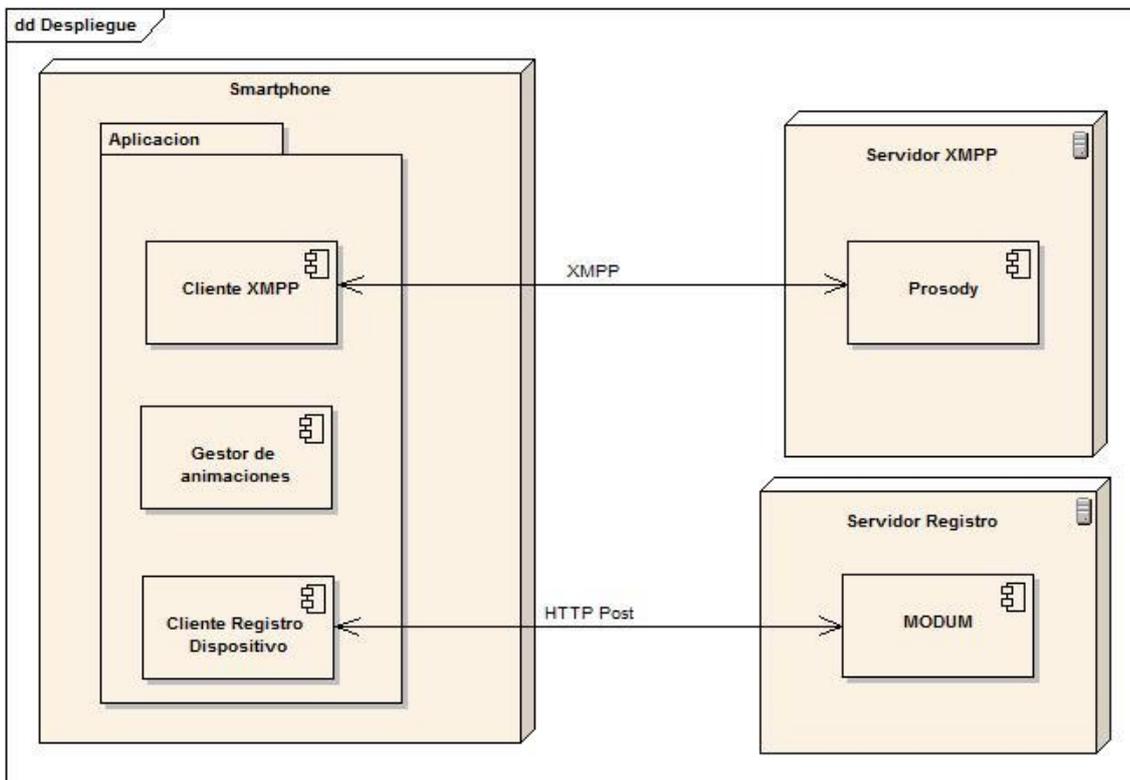


Figura 31 . Diagrama de despliegue

6.1.3.1 Smartphone

El smartphone es el nodo donde se ejecutará la aplicación. Por tanto, dentro de este elemento podemos localizar cada parte de implementada del sistema: el gestor de animaciones, el cliente para el sistema de mensajería XMPP, y el cliente para registrar los datos del dispositivo en MODUM.

6.1.3.2 Servidor XMPP

En este servidor es donde tenemos desplegado el servidor Prosody para el sistema de mensajería XMPP.

Dado que no es un proyecto en producción, utilizamos el servidor de pruebas, pero si en un futuro se lanzan aplicaciones utilizando este sistema de mensajería probablemente el servidor se establecería en el mismo en el que corre MODUM para proyectos en producción.

6.1.3.3 Servidor de registro

Denominamos servidor de registro al servidor en el que se encuentra funcionando la plataforma MODUM. Al igual que en el caso anterior, para este proyecto utilizamos el servidor de pruebas.

6.2 Diseño de Clases

En este apartado estudiaremos el diseño de las principales clases de la aplicación.

6.2.1 Diagrama de Clases

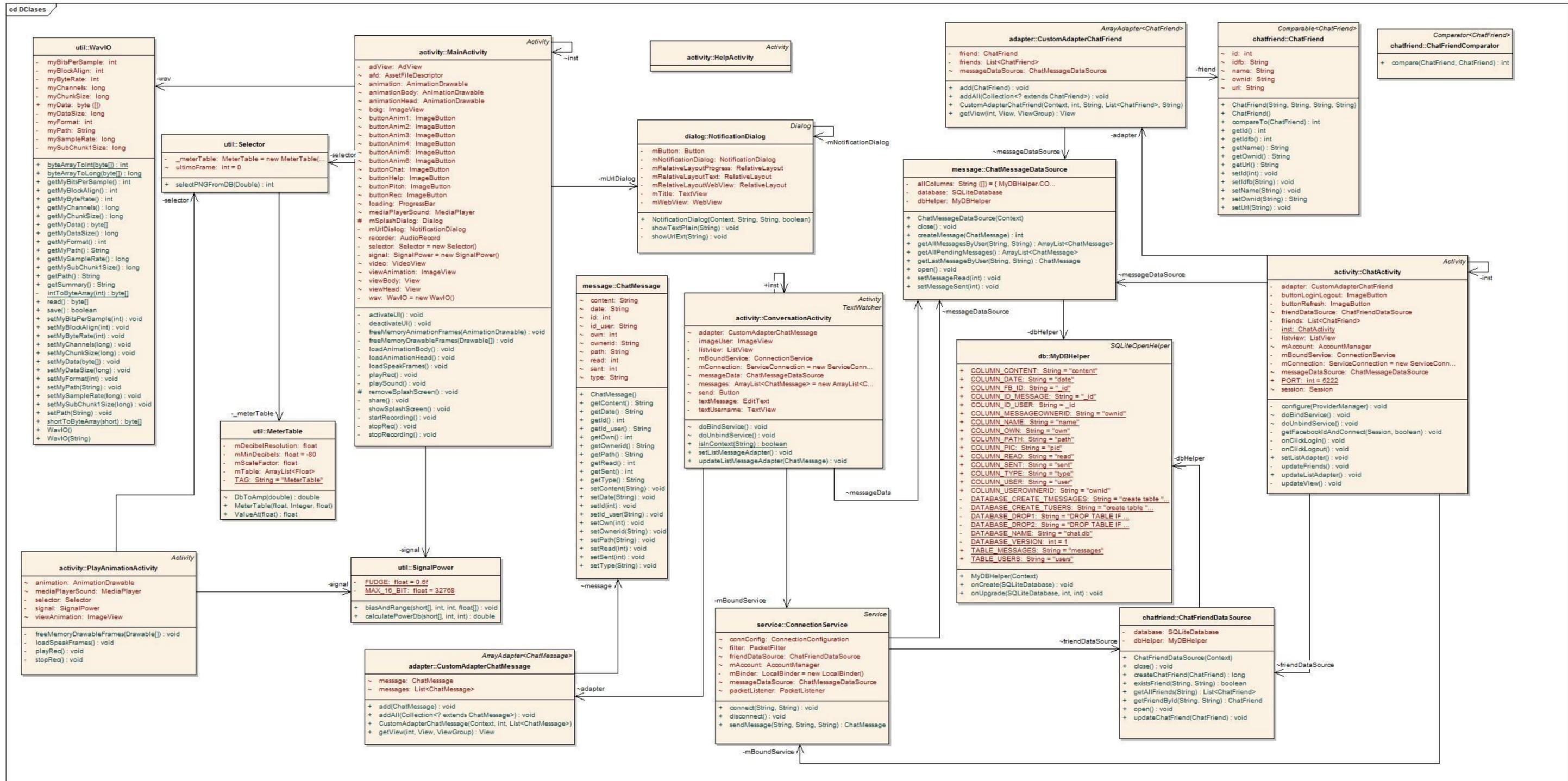


Figura 32 . Diagrama de clases

6.3 Diagramas de Secuencia

En este apartado vamos a desarrollar los diagramas de secuencia de los casos de uso teniendo en cuenta como hemos descrito el diseño del sistema hasta este punto.

Nos centraremos en aquellos casos de uso en los que intervienen un mayor número de elementos dentro del sistema, y que por tanto tenga más sentido diseccionar mediante un diagrama de secuencia.

Teniendo en cuenta lo anterior, hemos decidido elaborar el diagrama de secuencia que refleja toda la actividad entre elementos del sistema en el proceso que va desde generar una animación por parte del usuario, hasta compartirla en una de las conversaciones del sistema de mensajería.

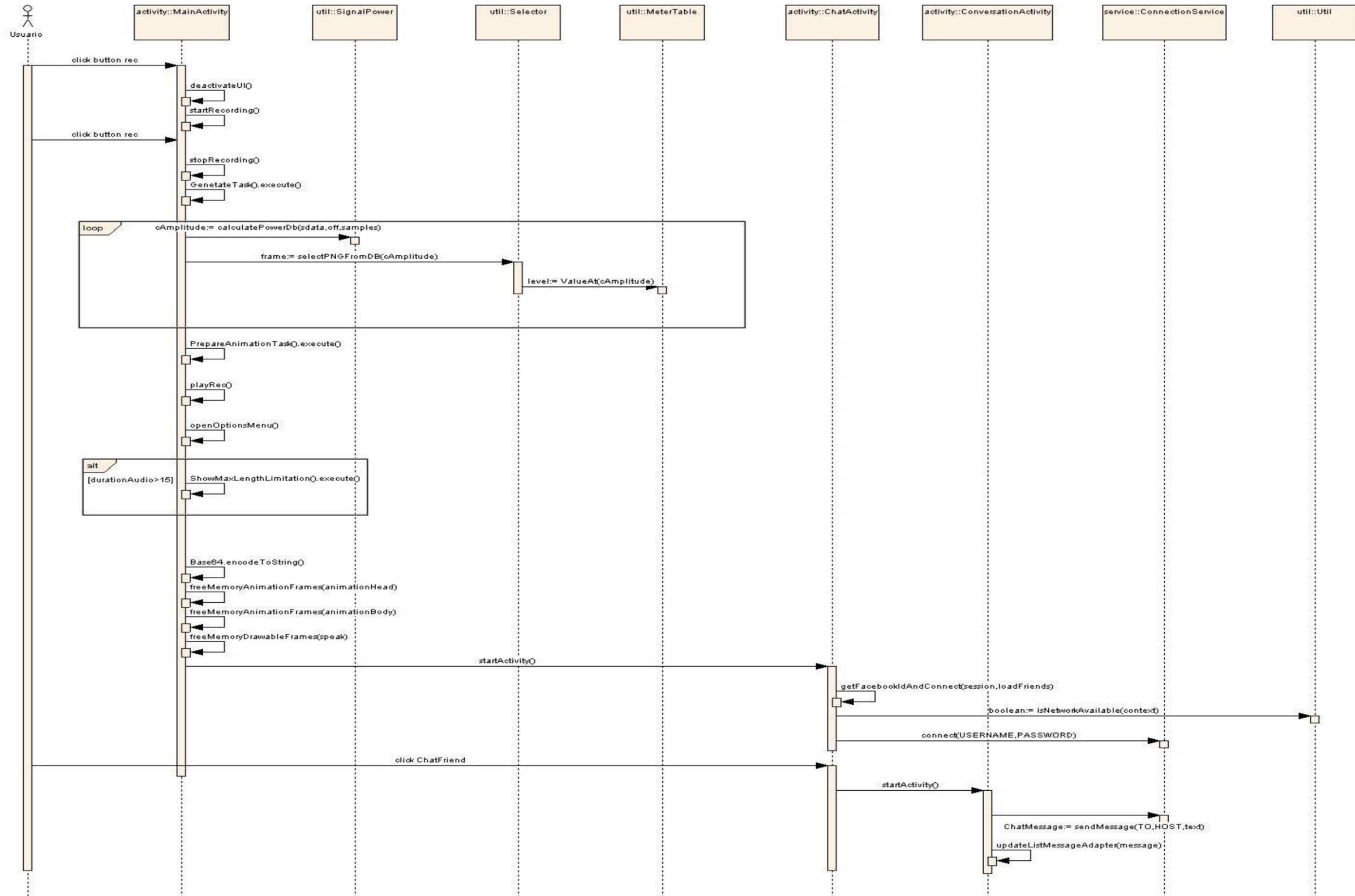


Figura 33 - Diagrama de secuencia de la generación de una animación y envío mediante el sistema de mensajería

6.4 Diseño de la Base de Datos

En este apartado abordaremos todo lo relacionado con la base de datos que utiliza nuestra aplicación para trabajar con los datos generados por el sistema de chat.

Como ya hemos comentado en otros apartados, la base de datos utilizada es la interna que proporciona a los desarrolladores el propio sistema operativo Android.

6.4.1 Descripción del SGBD Usado

Los sistemas de gestión de bases de datos (SGBD) representan un tipo de software que proporciona un nexo de unión entre una base de datos, y los usuarios y aplicaciones que la utilizan.

En este caso, lo que nos ofrece Android a los desarrolladores es una base de datos SQLite, interna del teléfono, de manera que cualquier aplicación puede crear o instanciar una base de datos para manejar su información de forma privada.

A nivel general, SQLite es un SGBD relacional que se caracteriza por estar contenida en una biblioteca de escaso tamaño y escrita en C. Su ligereza, optimización de rendimiento en las operaciones, y el hecho de que sea un proyecto de dominio público, la hacen la candidata para ser incluida dentro de un sistema operativo para móviles como es Android.

Cabe destacar que Android utiliza la versión 3.4.0 de SQLite

6.4.2 Integración del SGBD en Nuestro Sistema

A la hora de integrar la gestión con la base de datos en nuestro sistema se ha intentado independizar lo más posible del resto de funciones de la aplicación.

De esta manera podemos distinguir claramente diferentes partes, cada una con un objetivo concreto:

- **MyDBHelper:** esta clase es la que se encarga de instanciar la base de datos y todos sus elementos (tablas, columnas, etc). Será la encargada de crear la base de datos originalmente, actualizarla o eliminarla en caso de que sea necesario. Además, también se usará como referencia posteriormente para realizar las correspondientes consultas.
- **DataSource:** este elemento se crea por cada entidad que queremos manejar dentro de la base de datos, que en nuestro caso serán mensajes y usuarios. Cada uno de los DataSource implementados contiene una referencia a la clase MyDBHelper, e implementa todos los métodos que manejan los datos de las entidades que representan (inserciones, consultas, actualizaciones, borrados).

Mediante esta división clara de las funcionalidades, podemos establecer una organización adecuada del código que trabaja con la base de datos, y emplearlo posteriormente de manera más cómoda y clara en el resto de partes de la aplicación que requieran estos datos.

6.4.3 Diagrama E-R

Con un diagrama entidad relación se pretende representar las entidades relevantes de un sistema de información (en este caso de nuestra base de datos), las propiedades o atributos de estas entidades, además de las conexiones que establecen entre ellas.

En nuestro caso, los elementos o entidades relevantes son:

- Usuarios: la tabla de usuarios representará a los contactos que mostraremos en el chat al usuario de nuestra aplicación. Estos contactos serán obtenidos desde la propia cuenta de Facebook del usuario de nuestra aplicación (en caso de que nos otorgue el permiso necesario), en base a que también tengan instalada nuestra aplicación. La información que obtendremos de estos usuarios será simplemente su nombre, su identificador interno e imagen dentro de la citada red social. Cabe destacar a nivel de privacidad que esta información solo se almacenará internamente en el terminal, y en ningún caso será extraída del teléfono para otros usos.
- Mensajes: el registro de mensajes que hagamos en la base de datos es crucial para el correcto funcionamiento del sistema de mensajería. Principalmente tendremos que distinguir si los mensajes son nuestros o no, y en cualquier caso marcar claramente la dupla Emisor-Receptor, para poder estructurarlos de forma correcta dentro de cada conversación a la que correspondan. Además de esto, y como es evidente, también habrá que controlar el contenido de los mensajes, y la fecha concreta en la que se registró como información añadida.

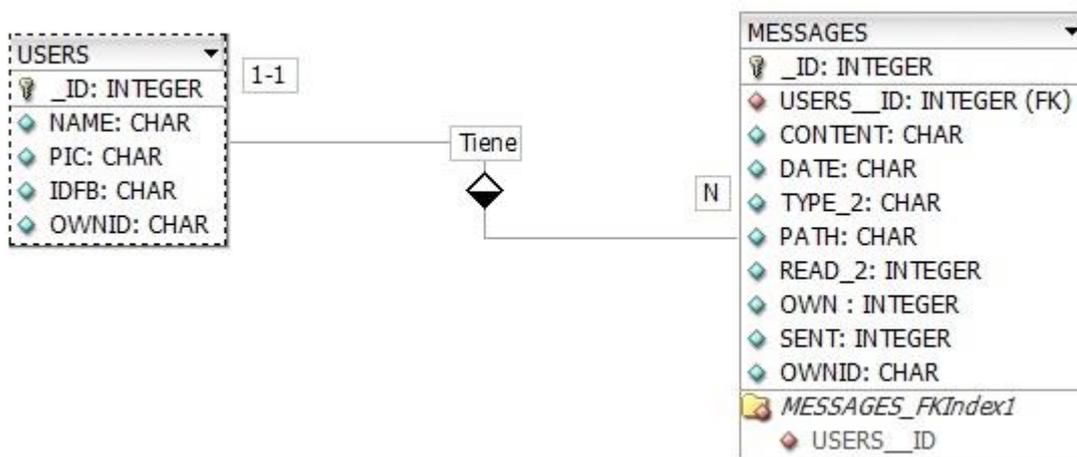


Figura 34 . Diagrama Entidad Relación

6.5 Diseño de la Interfaz

En este apartado mostraremos el diseño final de las principales pantallas de la aplicación, teniendo en cuenta las principales características que hemos definido durante la fase de análisis.

Cabe destacar que durante las pruebas realizadas se sacó la conclusión general de que los botones presentes en la interfaz, fundamentalmente los de la pantalla de animaciones, deberían ser más grandes, y en las capturas posteriores se verá el resultado final que se consideró finalmente correcto.

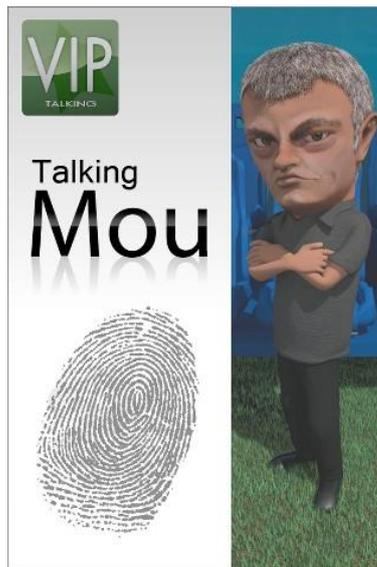


Figura 35 . Splash inicial de la aplicación



Figura 36 . Pantalla principal de la aplicación



Figura 37 . Menú de opciones de la pantalla principal

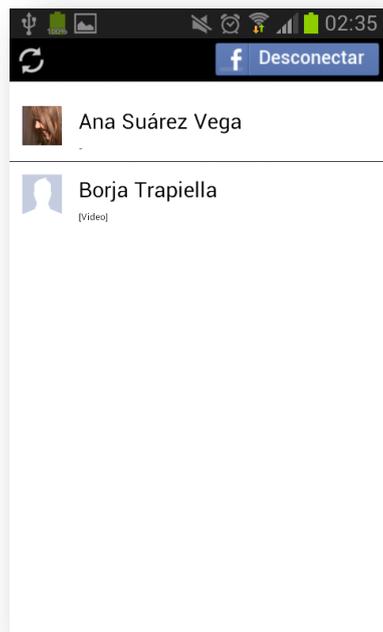


Figura 38 . Pantalla principal del sistema de mensajería con el listado de contactos



Figura 39 . Pantalla de una conversación del sistema de mensajería

6.6 Especificación Técnica del Plan de Pruebas

El proceso de pruebas se extiende durante todo el desarrollo de la aplicación, y en este apartado describiremos como aplicar las pruebas diseñadas durante este proceso de desarrollo.

Hay que tener en cuenta, como ya se ha mencionado en capítulos anteriores, que el sistema se ha desarrollado por partes que fueron ensambladas posteriormente para conformar el sistema completo. Por esto, hay que distinguir entre las pruebas que se aplicaron a las partes individuales y las que se realizaron sobre el sistema completo.

Todas las pruebas llevadas a cabo durante el proceso de desarrollo se realizaron bajo las siguientes condiciones:

- Equipo con sistema operativo Ubuntu 12 de 32 bits
- Android ADT
- Samsung Galaxy S – CyanogenMod 4.1.2
- Samsung Galaxy S2 – Adroid 4.1.2
- Samsung Galaxy S3 – Adroid 4.1.2
- Samsung Galaxy Ace – Android 2.3.5
- Nexus One – Android 2.3.5

6.6.1 Pruebas Unitarias

Estas pruebas son las que se realizan sobre los módulos independientes de la aplicación: por una parte el sistema de animaciones y por otra el de sistema de mensajería (la división por subsistemas definida durante el análisis).

6.6.1.1 Animaciones

En este punto las pruebas se realizan generando diferentes aplicaciones de prueba, en las que se simplificaban al máximo los elementos presentes de manera que solo nos tuviésemos que centrar en los procesos concretos que pretendíamos probar o experimentar.

| <i>Caso de Uso : Reproducir vídeos</i> | |
|---|--|
| Prueba | Resultado Esperado |
| Reproducir el vídeo alternando con la imagen estática de base | <p>Al mismo tiempo que se lanza la reproducción del vídeo, se lanza un proceso que comprueba cada poco tiempo el estado de la reproducción (momento del metraje en el que se encuentra).</p> <p>Este proceso trata de asegurar que el vídeo ya se encuentre en marcha, aunque sea de manera mínima, de forma que podamos ocultar la imagen que nos sirve como momento de</p> |

| | |
|--|--|
| | <p>reposo para las animaciones del personaje, y nos permite además optimizar este proceso para una amplia gama de dispositivos (no todos los terminales reproducen un vídeo a la misma velocidad)</p> |
| | <p>Esta prueba será de especial atención en los terminales de gama más baja dentro de los que utilizamos para las pruebas.</p> <p>En caso de apreciarse algún parpadeo entre la imagen estática y la animación, o de producirse fallos durante la reproducción del vídeo, habrá que revisar algunos factores fundamentales:</p> <ul style="list-style-type: none"> • La coordinación del proceso que chequea el estado de la reproducción del vídeo • La manera en la que realizamos la sustitución de los elementos visuales • El formato de los vídeos incluidos en el proyecto |

Caso de Uso : Reproducir animaciones

| Prueba | Resultado Esperado |
|--|--|
| <p>Reproducir una secuencia de frames como una animación utilizando el sistema propio de Android</p> | <p>Debemos generar una animación nativa de android (el elemento de interfaz que nos proporciona para reproducir secuencias de frames), en base a una serie de imágenes, ya sea incluidas en el proyecto o presentes en la propia memoria del teléfono.</p> <p>Debemos ser capaces de controlar el flujo de reproducción de esta animación en base a eventos del usuario (iniciar, parar, calcular final, etc).</p> |
| | <p>En caso de no producirse el resultado esperado, deberemos analizar detenidamente el uso del objeto de animación de Android, y sus particularidades, sobre todo a la hora de “cargar” las imágenes que formarán la animación, y de ejecutar eventos sobre dicho objeto (play, stop, etc)</p> |
| Prueba | Resultado Esperado |

| | |
|---|---|
| <p>Reproducir una animación alternando con la imagen estática de base</p> | <p>Esta prueba tiene el mismo objetivo que lo que se esperaba en el caso del vídeo, con la salvedad de que tiene como objetivo conseguir una mayor velocidad de respuesta en la ejecución de la animación.</p> <p>En este caso el evento será lanzado cuando el usuario toque zonas concretas del personaje animado, por lo que pretendemos dar una sensación más rápida y dinámica en el movimiento de la animación.</p> <p>Por otro lado, y al igual que ocurría en el caso de los vídeos, esperamos que no se produzcan parpadeos o interrupciones en la gestión y el manejo de los dos elementos visuales (estático y en movimiento)</p> <hr/> <p>Los pasos a seguir en caso de no reproducir el resultado esperado en esta funcionalidad, serán muy similares a las pruebas con el vídeo, pero aplicado al caso concreto del objeto que gestiona la animación:</p> <ul style="list-style-type: none"> • La coordinación del proceso que chequea el estado de la reproducción de la animación • La manera en la que realizamos la sustitución de los elementos visuales <p>Sin embargo, para este caso obviaremos las comprobaciones sobre el proceso de reproducción de la animación, puesto que de ello se encarga la prueba descrita anteriormente a esta.</p> |
|---|---|

| <i>Caso de Uso : Generar animaciones propias</i> | |
|---|---|
| Prueba | Resultado Esperado |
| <p>Grabar audio con el teléfono</p> | <p>Debemos ser capaces de realizar una grabación de audio con el propio dispositivo, y obtener ese audio tanto en formato “físico” (es decir, como un archivo dentro del propio teléfono) como a nivel de datos manejables directamente por la aplicación.</p> <p>De esta forma nos garantizamos dos formas diferentes de manejar el audio, para los procesos que realizaremos posteriormente.</p> <hr/> <p>En caso de no obtener los resultados esperados, podemos</p> |

| | <p>revisar los siguientes puntos de posible conflicto:</p> <ul style="list-style-type: none"> • Uso incorrecto del hardware del dispositivo. Puede deberse a un problema de permisos insuficientes para la aplicación • Errores en el manejo de los datos obtenidos. Este factor puede afectarnos tanto para obtener el fichero resultante como los datos equivalentes del audio. En ambos casos habrá que tener en cuenta el formato resultante de la salida del audio • Errores en la escritura de ficheros. En este caso puede deberse a un error en el manejo de los directorios, o como se ha comentado en puntos anteriores, por falta de permisos para la aplicación. |
|-------------------------------------|---|
| Prueba | Resultado Esperado |
| <p>Analizar un archivo de audio</p> | <p>Debemos de ser capaces de analizar un archivo de audio (en formato WAV) desde la propia aplicación, y obtener el conjunto de datos equivalente y manejable desde nuestro propio código.</p> <p>Junto con los datos puramente de audio, también serán de utilizados los metadatos con la información asociada a este audio (duración, samplerate, etc).</p> <p>Para obtener el resultado esperado y comprobar que el análisis del audio se realiza correctamente, la mejor forma es utilizar diferentes fuentes de audio:</p> <ul style="list-style-type: none"> • Un audio wav obtenido de cualquier fuente externa • El audio obtenido por la grabadora nativa del dispositivo • El audio obtenido por el caso de uso anterior <p>Además, para asegurarnos de que el proceso de análisis se realiza correctamente, deberemos realizar el proceso en la dirección contraria:</p> <ul style="list-style-type: none"> • Reproducir los datos de audio obtenidos directamente con el MediaPlayer de Android desde dentro de la propia aplicación (esto además incluye una funcionalidad de nuestro propio sistema) • Guardar el audio analizado como un nuevo fichero distinto, y tratar de reproducirlo directamente desde el teléfono u otra plataforma como un PC |

| Prueba | Resultado Esperado |
|--|---|
| <p>Modificar las propiedades de un audio</p> | <p>A partir de un conjunto de datos de audio (obtenidos por el proceso del caso anterior), debemos modificar el sonido final en base a una serie de características predefinidas (velocidad, tono, etc).</p> <p>Para este proceso nos apoyaremos en el uso de una librería externa, la cual habrá de estar correctamente configurada previamente.</p> <hr/> <p>Las comprobaciones del resultado esperado en este caso son similares a las de la prueba anterior, en tanto que debemos comprobar que las modificaciones realizadas realmente se traducen en un cambio de las características sonoras del audio original.</p> <p>Los posibles errores en este proceso vendrán ocasionados muy probablemente por un error en la configuración de la librería externa dentro de nuestro proyecto, o bien por un uso incorrecto de los parámetros en base a los cuales se realizarán las transformaciones (excesivos o demasiado pequeños).</p> <p>Será de especial importancia las pruebas con la demo original de la propia librería para ajustar estos detalles</p> |
| Prueba | Resultado Esperado |
| <p>Analizar audio para obtener frames equivalentes</p> | <p>Esta prueba se basará en gran medida en los resultados obtenidos en el proceso de análisis de audio en general, puesto que utilizará los datos arrojados por este análisis para posteriormente realizar este punto.</p> <p>Se trata de conseguir “traducir” el conjunto de datos de audio obtenidos a partir de un archivo wav, en un conjunto de “números” que representarán cada uno un frame determinado, teniendo en cuenta que cuanto menor sea el número menor será la amplitud de ese fragmento de audio (y por tanto condicionará de una forma u otra el movimiento del personaje en la animación resultante)</p> <hr/> <p>En este punto la comprobación de los resultados se centrará principalmente en comprobar que obtenemos un número de frames equivalentes acorde a la relación fps/duración del audio, y que a primera vista no apreciamos grandes desfases entre el audio analizado y las amplitudes obtenidas.</p> <p>En fases posteriores con un mayor grado de integración de los subsistemas, se podrán realizar las comprobaciones</p> |

| | |
|--|--|
| | directamente con la reproducción de las animaciones generadas. |
|--|--|

| <u>Caso de Uso : Guardar vídeo</u> | |
|---|--|
| Prueba | Resultado Esperado |
| Generar un archivo de vídeo en base a unas imágenes y un audio localizado por la aplicación | Pretendemos obtener un archivo de vídeo en un formato adecuado para la reproducción en dispositivos móviles (mp4, 3gp) que sea resultado de la combinación de un conjunto de imágenes y un archivo (o datos) de audio. Al igual que en el caso de la modificación de audio, esta funcionalidad utilizará las funciones de una librería externa al proyecto, y habrá que tener en cuenta la integración correcta de la misma dentro de nuestra aplicación. |
| | En este caso, al tratarse de un conjunto de imágenes y un audio localizados directamente, los errores producidos durante estas pruebas se deberán probablemente a un uso incorrecto de la librería externa. |
| Prueba | Resultado Esperado |
| Generar un archivo de vídeo utilizando los frames de la aplicación y el audio modificado | Esta prueba busca avanzar un paso más en la integración de las funcionalidades de este subsistema, combinando de esta forma la grabación, modificación y análisis del audio, junto con el punto anterior, para obtener como resultado un vídeo equivalente a las animaciones dinámicas que generará la aplicación final. |
| | En este punto, y una vez dado por hecho que las funcionalidades convergentes en este punto están correctamente desarrolladas, los errores encontrados se deberán probablemente a algún fallo en la integración de las partes. Errores comunes pueden ser debidos a fallos a la hora de apuntar a los elementos que debemos utilizar en la creación del vídeo (imágenes o audio). También habrá que prestar especial atención a la sincronización del audio con las imágenes, revisando siempre |

| | |
|--|---------------------------|
| | la relación fps/duración. |
|--|---------------------------|

6.6.1.2 Sistema de mensajería

| <u>Caso de Uso : Enviar mensaje con/sin conexión</u> | |
|---|---|
| Prueba | Resultado Esperado |
| <p>Enviar mensajes utilizando como servidor el GTalk del correo de Google</p> | <p>Empleando dos cuentas de correo de Gmail (una emisora y otra receptora), enviaremos mensajes desde nuestro cliente en el dispositivo (registrado como es lógico con las credenciales correspondientes de gmail).</p> <p>En este punto realizaremos el proceso en primer lugar entre nuestro cliente y el cliente web de Gmail, de manera que nos aseguremos que el cliente al menos funciona, y posteriormente estableciendo la comunicación directamente entre dos terminales utilizando nuestra aplicación (dándole ya toda la responsabilidad de enviar y recibir todos los mensajes)</p> <p>El resultado esperado es simplemente el correcto intercambio de mensajes entre las partes en los dos casos testados.</p> <p>Al igual que ha ocurrido en casos anteriores, al tratarse de una librería externa a nuestro proyecto habrá que prestar atención a la inclusión de la misma en nuestro trabajo, además del uso correcto de sus funcionalidades (más teniendo en cuenta que está es mucho más compleja y completa que las dos mencionadas anteriormente)</p> |
| Prueba | Resultado Esperado |
| <p>Enviar mensajes entre terminales utilizando nuestro propio servidor</p> | <p>En esta prueba esperamos obtener los mismos resultados que en la anterior (exclusivamente entre terminales), pero dejando de un lado el servidor del Gtalk, y empleando directamente el nuestro.</p> <p>Esto implica controlar aspectos como el registro y la identificación de usuarios, además de la correcta conexión a nuestro servidor, algo que anteriormente no entraba dentro de lo implementado</p> |

| | <p>En cuanto a intercambio de mensajes el resultado esperado es el mismo que en el anterior, y por tanto si el cliente se comportaba correctamente en el caso anterior, en este también debería de hacerlo.</p> <p>En caso de no apreciar los resultados esperados, deberemos de analizar con detenimiento las acciones ejecutadas en el servidor, puesto que es el punto diferencial con el caso anterior.</p> <p>Las nuevas funcionalidades implementadas en el cliente en esta parte (registro, login, etc) también deberán de tener una especial atención.</p> |
|--|---|
| Prueba | Resultado Esperado |
| <p>Enviar mensajes entre terminal y un cliente web</p> | <p>Aunque no es una funcionalidad propia de nuestro sistema, si es un requisito indispensable a la hora de seleccionar el protocolo a utilizar en el proyecto, por lo que se realizarán pruebas exhaustivas sobre esta funcionalidad.</p> <p>Se trata de obtener los mismos resultados que en el primer caso, entre nuestro dispositivo y el cliente web de Gmail, pero empleando en nuestro caso tanto un cliente propio como nuestro servidor de las pruebas anteriores.</p> <p>El resultado esperado en este punto es el del correcto funcionamiento del sistema de intercambio de mensajes, una vez hemos depurado todo el proceso en el caso anterior.</p> <p>En este caso habrá que prestar especial atención al funcionamiento del proceso en la parte del cliente web (dando por sentado que las pruebas anteriores fueron correctas)</p> |

6.6.2 Pruebas de Integración y del Sistema

En este apartado realizamos un bloque de pruebas similar al anterior, pero dentro del marco de ambos subsistemas integrados. En este contexto, disponemos de una aplicación similar a la definitiva, en la que los subsistemas pueden interactuar entre sí mediante los mecanismos descritos en otros apartados.

Al igual que en la definición que hicimos en el análisis de este bloque de pruebas, cabe destacar que hemos agrupado ciertos casos de uso por tener una relación funcional muy estrecha, la cual hacía necesaria las pruebas conjuntas de ambos casos.

| <i>Caso de Uso : Reproducir animaciones / Reproducir videos</i> | |
|---|---|
| Prueba | Resultado Esperado |
| Reproducir animaciones y vídeos de forma consecutiva e intercalándolos, como parte del funcionamiento normal que se espera de la aplicación | <p>El resultado esperado en esta prueba es fruto de la combinación de los resultados anteriores en los que probábamos las animaciones y los vídeos de forma independiente.</p> <p>Tanto la reproducción de los vídeos como las secuencias de imágenes han de reproducirse sin parpadeos, sin apreciarse el cambio entre la imagen estática y el comienzo del movimiento, y obteniendo una respuesta más rápida en el caso de las secuencias de imágenes.</p> <p>En esta fase de integración habrá que tener en cuenta los problemas de memoria derivados de la carga de todos los elementos multimedia al mismo tiempo. Como es evidente, no se deberán producir tales errores</p> <p>En caso de no producirse los resultados esperados, habrá que atender a dos factores principales (teniendo en cuenta que habremos solucionado los anteriores problemas durante las pruebas unitarias):</p> <ul style="list-style-type: none"> • Los problemas de memoria. Habrá que vigilar especialmente lo que cargamos en la memoria del dispositivo en todo momento, como liberarlo, etc • Puede ser que los ajustes de sincronización que funcionaban durante la fase unitaria se vean resentidos en este punto por el mayor peso del sistema completo. En este caso se deberán volver a revisar estos ajustes, para adaptarlos a las nuevas circunstancias |

| <i>Caso de Uso : Generar animaciones propias / Guardar video / Compartir ámbito general /Compartir en chat</i> | |
|---|--|
| Prueba | Resultado Esperado |
| Generar una animación dentro de la aplicación final | Se espera la ejecución completa y sin fallos del siguiente proceso, fruto de la integración de gran parte del subsistema de animaciones: |

| | |
|--------------------------------------|--|
| | <ul style="list-style-type: none"> • Inicio de grabación de audio • Detención de grabación de audio • Análisis de audio • Modificación de audio • Equivalencia audio/frames • Generación de animación en base a imágenes • Reproducción de animación sincronizada con audio • Mostrar submenú de opciones |
| | <p>El resultado esperado es el del conjunto global de cada una de las funcionalidades que participan en el proceso.</p> <p>Como he comentado en otras ocasiones, una vez dados por buenos los procesos de manera individual en las pruebas unitarias, en caso de fallos habrá que atender especialmente a la integración de las partes y los puntos en los que confluyen</p> |
| Prueba | Resultado Esperado |
| <p>Guardar video de la animación</p> | <p>Esta prueba también se realizó durante la fase de pruebas unitarias, por lo que el resultado esperado es el mismo que en esa ocasión.</p> <p>En este caso habrá que tener en cuenta que el proceso se realizará en conjunto con el total de la aplicación, y que habrá de saber localizar en todo momento los datos necesarios para generar el vídeo.</p> <p>Se ha de poder generar el vídeo en diferentes puntos o momentos de la aplicación, bloqueando el uso de la interfaz durante la duración del proceso, y volviendo a mostrar siempre al terminar las opciones del submenú correspondiente.</p> <p>En caso de no obtener el resultado esperado, atender especialmente a los puntos donde se está ejecutando el proceso de generación del vídeo, y la forma en la que hacemos referencia a los elementos que lo conforman (audio y vídeo)</p> |
| Prueba | Resultado Esperado |

| | |
|---|---|
| <p>Compartir vídeo guardado en el ámbito general</p> | <p>El propio sistema Android nos permite lanzar su pantalla genérica de publicación de contenido, y se adecua a las aplicaciones presentes en el dispositivo.</p> <p>Como parte de la integración del sistema, se añade el caso extra de que si el vídeo aún no se ha generado y se intenta lanzar esta opción, se ejecuta previamente el proceso de creación del vídeo antes de lanzar la opción de compartir.</p> <p>En caso de no obtener el resultado esperado, prestar especial atención al proceso de creación del vídeo y al directorio donde estemos intentando acceder al mismo</p> |
| <p>Prueba</p> | <p>Resultado Esperado</p> |
| <p>Compartir vídeo guardado en el chat de la aplicación</p> | <p>El audio grabado por el usuario se codifica correctamente en Base64 y es enviado correctamente a través del sistema de chat propio de la aplicación.</p> <p>Es especialmente importante que se mantenga el concepto del proceso “compartir en el chat”, con un inicio y un final, ya que siendo estrictos la aplicación se encuentra en la parte del sistema de animaciones y no del chat.</p> <p>En este sentido se habrá de controlar que una vez salga del chat, la pantalla de animaciones siga en el mismo estado que inició el proceso de compartir (con el submenú desplegado y aun manejando la misma animación).</p> <p>El audio habrá de recibirse correctamente en el receptor (aunque esto es responsabilidad del subsistema de mensajería) y deberá reproducirse la animación desde la propia conversación tanto en el emisor como en el receptor.</p> <p>Este punto será tratado en pruebas sucesivas.</p> <p>El análisis de los errores producidos en este caso es de especial dificultad porque es el caso crítico donde más confluyen ambos subsistemas.</p> <p>De todas formas, además de revisar la integración entre ambas partes y comprobar que el flujo del proceso es el esperado (cambios de pantalla, mantener los datos entre ellas, etc), habrá que prestar especial atención a la parte de la codificación/decodificación en Base64, puesto que se trata del añadido extra respecto a la fase de pruebas unitarias.</p> <p>De especial atención será realizar la codificación y decodificación de la misma forma, para conseguir unos datos</p> |

| | |
|--|--|
| | equivalentes al final de cada proceso. |
|--|--|

Caso de Uso : Enviar mensajes sin/con conexión

| Prueba | Resultado Esperado |
|--|--|
| Enviar un mensaje con conexión a uno de nuestros contactos, esperando verse en la pantalla de conversación | <p>Se trata de la funcionalidad básica que podemos esperar del sistema de mensajería implementado.</p> <p>En este parte de la integración sin embargo, tienen que confluír correctamente las siguientes partes:</p> <ul style="list-style-type: none"> • Conexión correcta al servidor XMPP • Registro o Login del usuario (según el caso) • Carga de contactos del usuario (desde Facebook o desde la propia base de datos del terminal) • Carga de conversación con el contacto seleccionado • Envío correcto del mensaje (que será registrado en la base de datos del dispositivo) <p>En este punto, en caso de no producirse el resultado esperado deberíamos atender al resto de elementos complementarios del sistema de mensajería que al propio envío del mensaje (aunque sin dejar de comprobarlo evidentemente).</p> <p>Al igual que en otros casos, deberemos prestar especial atención a los elementos incluidos como parte del proceso de integración.</p> |
| Prueba | Resultado Esperado |
| Enviar un mensaje con conexión a uno de nuestros contactos, esperando verse en la pantalla de contactos | <p>El resultado esperado es el mismo al caso anterior, con la salvedad de que en lugar de mostrar el mensaje en la conversación del contacto, el sistema ha de realizar la gestión del mensaje por debajo (de manera transparente al usuario) y actualizar el listado de contactos de forma correcta.</p> <p>El nuevo mensaje recibido deberá mostrarse como último mensaje en la conversación con ese contacto concreto, y visualmente destacarla para dejar claro que existe un nuevo mensaje que no se ha visto (en la parte del receptor).</p> |

| | |
|---|---|
| | En este caso, y al tratarse de una gestión que ha de realizar el sistema sin intervención alguna del usuario, los posibles problemas que aparezcan tendrán que ver con el sistema de gestión de la base de datos, y sobre todo la forma de refrescar la información y la interfaz en el momento de recibir un nuevo mensaje en ese momento de la aplicación. |
| Prueba | Resultado Esperado |
| Enviar varios mensajes sin conexión y conectarse posteriormente | <p>Cada vez que se produce una reconexión al servidor XMPP, ya sea por reiniciar la aplicación, una caída de la red, etc, se deben enviar de forma automática todos los mensajes listados en la base de datos del terminal como no enviados.</p> <p>Estos mensajes ya deberían de aparecer en sus correspondientes conversaciones listados correctamente, y de acuerdo a lo definido en las dos pruebas anteriores.</p> <p>En caso de no obtener el resultado esperado en este punto, el problema puede deberse a algún fallo a la hora de listar los mensajes no enviados, o de registrarlos como tal en la base de datos.</p> |

| <i>Caso de Uso : Reproducir animaciones enviadas/recibidas</i> | |
|---|--|
| Prueba | Resultado Esperado |
| Reproducir una animación enviada por nosotros mismos | <p>Se espera exactamente el mismo resultado que cuando se reproduce una animación en la pantalla principal del personaje animado.</p> <p>En este caso no cabe esperar un resultado erróneo fruto de problemas en la comunicación XMPP, por lo que los posibles fallos pueden deberse a una mala integración de la reproducción de animaciones en este punto del sistema.</p> |
| Prueba | Resultado Esperado |
| Reproducir una animación recibida | <p>Se espera exactamente el mismo resultado que cuando se reproduce una animación en la pantalla principal del personaje animado.</p> <p>En este caso, además de posibles errores en la integración de la reproducción de animaciones, habrá que prestar atención al proceso de comunicación XMPP, y de la decodificación de</p> |

| | |
|--|---|
| | los mensajes recibidos como audio. Hay que tener en cuenta que estos mensajes deberán venir marcados correctamente para ser interpretados como una animación y no como un mensaje estándar. |
|--|---|

6.6.3 Pruebas de Usabilidad y Accesibilidad

Ya se ha comentado en ocasiones, que el carácter de la aplicación desarrollada (sobre todo en tono lúdico y de humor) hace que el rango de público objetivo de la aplicación sea muy amplio, tanto si atendemos a factores como la edad y el género, o a las habilidades de cada usuario para utilizar aplicaciones de este tipo o en general en dispositivos móviles.

En este sentido realizaremos una serie de pruebas sobre la aplicación, con el fin de evaluar el grado de usabilidad de la misma en diferentes perfiles de usuario, teniendo en cuenta que no todos estos perfiles tendrán sentido en todas las funcionalidades de la aplicación (el usuario tipo del sistema de chat no tiene que ser el mismo que el de la parte de animaciones necesariamente). Por esta razón, y dependiendo del usuario que haga las pruebas y su perfil, el cuestionario se realizará tanto para el sistema de animaciones como para el sistema de mensajería

Antes de realizar cuestiones relativas a la aplicación, se someterá a un cuestionario previo a los usuarios de prueba para determinar su nivel de conocimiento o experiencia dentro de este ámbito.

| |
|---|
| ¿Usa un smartphone frecuentemente? |
| <ol style="list-style-type: none">1. Todos los días2. Varias veces a la semana3. Ocasionalmente4. Nunca o casi nunca |
| ¿Qué tipo de actividades realiza con su teléfono? |
| <ol style="list-style-type: none">1. Es parte de mi trabajo o profesión2. Lo uso básicamente para ocio (Multimedia, Internet en general, etc)3. Únicamente para llamar4. Leo el correo y navego ocasionalmente |
| ¿Ha utilizado alguna vez una aplicación de este tipo? (Tipo Talking) |
| <ol style="list-style-type: none">1. Sí, en muchas ocasiones2. Sí, pero no habitualmente |

| |
|--|
| 3. No, nunca |
| ¿Qué busca Vd. Principalmente cuando instala una nueva aplicación en su teléfono? |
| <ol style="list-style-type: none"> 1. Que sea fácil de usar 2. Que no tenga publicidad molesta 3. Que sea rápida 4. Que funcione correctamente en mi dispositivo |

Una vez determinado el perfil de los usuarios de prueba, se puede proceder a realizar el cuestionario acerca de las características funcionales de la aplicación. De esta forma, seremos capaces de valorar en su justa medida los resultados de mismo.

| Facilidad de Uso | Siempre | Frecuentemente | Ocasionalmente | Nunca |
|--|----------------|-----------------------|-----------------------|--------------|
| <i>¿Sabe dónde está dentro de la aplicación?</i> | | | | |
| <i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i> | | | | |
| <i>¿Le resulta sencillo el uso de la aplicación?</i> | | | | |
| Funcionalidad | Siempre | Frecuentemente | Ocasionalmente | Nunca |
| <i>¿Funciona cada tarea como Vd. espera?</i> | | | | |
| <i>¿El tiempo de respuesta de la aplicación es muy grande?</i> | | | | |
| <i>¿Echa en falta algún elemento que complemente la funcionalidad?</i> | | | | |
| <i>¿Se comunican de manera correcta los eventos de la aplicación?</i> | | | | |
| <i>¿Ha detectado algún error o detalle susceptible de mejora en el comportamiento del sistema?</i> | | | | |
| Calidad del Interfaz | | | | |
| Aspectos gráficos | Muy | Adecuado | Poco | Nada |

| | Adecuado | | Adecuado | Adecuado |
|---|----------|-----------|-----------|----------------|
| <i>El tipo y tamaño de letra es</i> | | | | |
| <i>Los iconos e imágenes usados son</i> | | | | |
| <i>Los colores empleados son</i> | | | | |
| <i>El tamaño general es</i> | | | | |
| <i>La distribución de los elementos es</i> | | | | |
| <i>La transición entre las pantalla es</i> | | | | |
| <i>La adaptación a la pantalla de mi terminal es (Modelo:)</i> | | | | |
| Diseño de la Interfaz | | Si | No | A veces |
| <i>¿Le resulta fácil de usar?</i> | | | | |
| <i>¿El diseño de las pantallas es claro y atractivo?</i> | | | | |
| <i>¿Cree que los iconos son claros respecto a la funcionalidad que presentan?</i> | | | | |
| <i>¿Cree que el diseño podría mejorar en algún momento determinado?</i> | | | | |
| Observaciones | | | | |
| Cualquier comentario del usuario | | | | |

6.6.4 Pruebas de Rendimiento

Como se ha comentado en otros apartados de la documentación, para el desarrollo de este proyecto habrá que tener muy en cuenta el rendimiento en función del modelo de dispositivo, fundamentalmente en lo que hace referencia al sistema de animaciones.

La parte más crítica a comprobar en cuestiones de rendimiento es la reproducción de vídeos y animaciones, puesto que debido al diseño de la aplicación y a como se ha pensado su funcionamiento, existe la necesidad de alternar visualmente imágenes estáticas con imágenes en movimiento (vídeos o animaciones) de manera que de una sensación fluida y sin cortes. Es decir, se trata de que la ejecución de las animaciones parezca lo más posible en tiempo real, y no basada en la selección de diferentes escenas animadas.

Todo lo relacionado con este punto es especialmente crítico en aspectos de rendimiento, puesto que no todos los terminales reproducen un vídeo o una animación a la misma velocidad, por lo que decisiones o mecanismos del estilo “Click en play – Oculto imagen – Muestra vídeo”, difícilmente nos proporcionarán resultados satisfactorios.

En este sentido, todas las pruebas definidas en los apartados anteriores y relacionadas con estos aspectos, se han realizado en el mayor número posible de dispositivos, atendiendo especialmente como se ha comentado a las especificaciones de los mismos, no solo de hardware sino también de software.

Los dispositivos utilizados para calibrar el rendimiento de esta parte han sido:

- Samsung Galaxy S – CyanogenMod 4.1.2 y 4.2.1 (Dispositivo rooteado con una ROM no oficial de Android)
- Samsung Galaxy S2 – Android 4.1.2 – Android 4.0
- Samsung Galaxy S3 – Android 4.1.2
- Samsung Galaxy Ace – Android 2.3.5
- Nexus One – Android 2.3.5

El uso de diferentes versiones del sistema operativo Android se debe principalmente a actualizaciones realizadas durante el proceso de desarrollo.

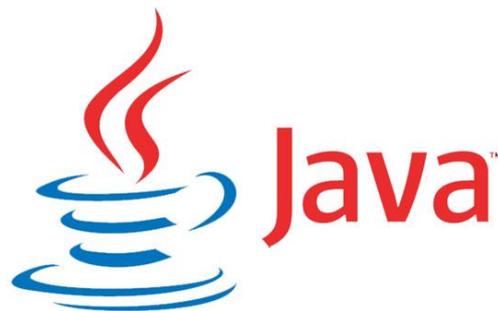
Capítulo 7. Implementación del Sistema

7.1 Lenguajes de Programación

7.1.1 Java

Java es un lenguaje de programación desarrollado originalmente por Sun Microsystems y publicado en 1995. Se trata de un lenguaje cuya sintaxis deriva mucho de lenguajes como C y C++, pero con una orientación de mucho más alto nivel. Es un lenguaje de programación de propósito general, basado en clases y orientado a objetos, que fue diseñado con la intención de que los programas desarrollados en Java, una vez compilados pudiesen ser utilizados en cualquier dispositivo o plataforma.

Todas estas características, y probablemente el hecho de que sea uno de los lenguajes más extendidos y asimilados actualmente, son la razón de que Google decidiese utilizarlo como el lenguaje de programación a utilizar para desarrollar aplicaciones para Android, haciendo de esta forma el proceso de entrada en la plataforma mucho más directo para millones de desarrolladores que ya tienen conocimientos de Java.



7.2 Herramientas y Programas Usados para el Desarrollo

7.2.1 Eclipse

Eclipse es un entorno de desarrollo de código abierto que incluye un conjunto de herramientas de programación que permiten desarrollar proyectos en un gran número de lenguajes de programación diferentes, y en multitud de plataformas (Linux, Windows, Mac, etc).

Fue originalmente desarrollado por IBM como parte de su conjunto de herramientas de desarrollo, pero en la actualidad es coordinado por la Fundación Eclipse, una entidad sin ánimo de lucro que trata de fomentar y potenciar la comunidad de desarrollo de código abierto.

Su principal característica es que permite incorporar numerosos “plugins”, que añaden multitud de funcionalidades, pero especialmente la posibilidad de desarrollar proyectos para distintos tipos de lenguajes.

Por esta razón es utilizado como uno de los principales entornos a la hora de desarrollar para Android, ya que no solo es compatible con Java, sino que también permite añadir los propios complementos de Android para realizar proyectos en esta plataforma (gestor de versiones, gestor de emuladores, editor gráfico, etc)



Figura 40 . Logo Eclipse

7.2.2 Android ADT Bundle

Durante el desarrollo del proyecto, Google publicó en la web para desarrolladores de Android el ADT Bundle.

Este editor suponía una versión de Eclipse “pre-editada” por Google, y que te proporcionaba un Eclipse totalmente funcional y preparado para empezar a trabajar con Android desde el primer momento, sin necesidad de perder tiempo en configurar complementos añadidos.

Aunque el proyecto se comenzó inicialmente utilizando una versión de Eclipse tradicional, posteriormente proseguí el desarrollo utilizando las distintas versiones del ADT Bundle que se iban publicando.

7.2.3 Rockmongo

Rockmongo es una herramienta de gestión de la base de datos NoSql MongoDB, desarrollada en PHP 5 y por tanto pensada para ser accesible desde un navegador web.

Sus principales características son que es de código libre, fácil y rápida de instalar, dispone de control de usuarios (login), y permite manejar la base de datos de diversas formas (no solo hacer consultas, sino también inserciones, actualizaciones y borrados).

Además incluye una consola javascript que te permite interactuar con la base de datos de una forma más sofisticada, y codificando de manera más concreta las operaciones que se deseen hacer con la base de datos.

Esta herramienta ha sido utilizada para controlar el registro de dispositivos de la aplicación en el sistema de la empresa (MODUM), y gestionar el registro de envío de notificaciones C2DM a la aplicación.

7.3 Creación del Sistema

7.3.1 Problemas Encontrados

En este apartado enumeramos los problemas encontrados en el desarrollo y las soluciones que se aplicaron para solventarlos.

7.3.1.1 Sincronización de imagen estática e imágenes en movimiento

Lo primero que se abordó al comenzar el proyecto por la parte de las animaciones fue lo que a priori parecía lo más sencillo: reproducir las animaciones tanto de vídeo como en base a fotogramas.

Por separado no suponía ningún tipo de problema, pero una vez se comenzaron a integrar las diferentes partes que conformaban el sistema de animaciones, se fue haciendo patente que conseguir manejarlo de forma fluida no era algo trivial, y más si se ejecutaba la aplicación en terminales menos potentes.

La solución se me ocurrió después de haber trabajado en HTML5 en un proyecto con un requisito similar. En este caso la limitación que provocaba los posibles errores de sincronización y parpadeos era la conexión a Internet, y se consiguió resolver utilizando los eventos de reproducción del componente de vídeo nativo de HTML5. La idea era forzar a que el proceso de reproducción llevase en marcha al menos unos segundos, de manera que cuando hiciésemos el cambio de imagen estática a movimiento, hubiese contenido que mostrar y no un fondo negro vacío (aunque fuese durante una milésima de segundo).

En base a esto, se trató de adoptar la misma solución en este proyecto, o al menos la misma esencia de la idea. Se da la circunstancia de que en Android el reproductor multimedia no dispone del mismo evento que nos permitía implementar el mecanismo en HTML5, por lo que hubo de hacerse de forma manual, realizando comprobaciones cada X tiempo de la situación actual de la reproducción; si el proceso ya había avanzado lo suficiente para mostrar el parpadeo, podíamos realizar el intercambio entre las imágenes.

Aunque en general este sistema funcionó de forma correcta, el cambio a la hora de ponerlo en práctica supuso otra serie de problemas derivados que requirieron mayor control de todo el proceso en general: problemas de memoria con el reproductor, procesos que se ejecutaban de manera indefinida, etc.

También hay que destacar que tras la reproducción de vídeo se hacía necesario volver a mostrar la imagen estática, puesto que la imagen final de los vídeos una vez finalizados no siempre mostraba el mejor resultado posible, y la ejecución consecutiva de animaciones terminaba arrojando un resultado irregular.

Para este caso se utilizaron una vez más los eventos de vídeo, en el que podíamos ejecutar una determinada tarea justo cuando un video terminase. En este caso no era necesario un control de rendimiento, puesto que el evento era interno del propio reproductor y no podía haber

fallo en su notificación. De esta manera, cada vez que un vídeo finalizase, podríamos volver a tener visible la imagen estática con el personaje en reposo.

Todos estos mecanismos fueron utilizados de manera similar en el caso de la reproducción de animaciones en base a secuencias de fotogramas, pero con las particularidades características de este tipo de animaciones.

7.3.1.2 Generar vídeos a partir de imágenes y audio

Como ya se ha comentado en otros apartados, la posibilidad de crear vídeos en base a contenido multimedia como imágenes y audio no es una funcionalidad que Android soporte de forma nativa.

Siendo uno de los requisitos fundamentales de la aplicación, hubo que investigar las posibilidades que existían para poder incluirlo de alguna forma, encontrando de esta forma las librerías de JavaCV y las compilaciones de código nativo del Ffmpeg (mediante el Android NDK).

Una vez encontrada esta vía, se inició un proceso (documentado en las pruebas unitarias) en el que comprobamos las posibilidades de la librería, como podíamos importarla a nuestro proyecto, etc.

Durante este proceso, hubo que solventar otra serie de inconvenientes como que la librería no permitía añadir audio a los vídeos generados, o algunos otros detalles de configuración que o bien no nos dejaban ejecutarla, o bien el resultado era un vídeo sin contenido.

Finalmente, mediante pruebas de ensayo y error, y con la ayuda de los foros de soporte de la librería, conseguí incluirlo y utilizarlo de manera exitosa en el proyecto.

7.3.1.3 Análisis del audio y obtención de frames equivalentes

Una de las partes del proyecto, en lo relacionado con las animaciones, que más trabajo ocasionó fue la de analizar el audio para calcular su amplitud y obtener los frames equivalentes, de manera que el personaje sincronizase los movimientos de su boca con los del audio grabado.

Desde un principio dispuse de algoritmo que devolvía las equivalencias, usado en otros proyectos de la empresa, algunos también desarrollados por mí mismo, y con buenos resultados.

Sin embargo, durante la fase de pruebas entre varias personas, diferentes terminales, diferentes versiones de Android, etc, parecían sucederse algunas animaciones demasiado erróneas, en las que el movimiento del personaje era muy exagerado (pasaba de ninguna amplitud a la máxima de un fotograma a otro).

Estos errores no parecían deberse a nada en concreto, ya que bien podías generar 10 animaciones consecutivas totalmente correctas y con una sincronización casi perfecta, que a la primera saltar el fallo y generar una desastrosa.

Tras numerosas pruebas, en las que llegue a comparar los bytes de audio con el resultado del análisis arrojado de manera minuciosa, pude encontrar el error que se estaba produciendo.

Principalmente se debía a que en un primer momento durante el proceso de cálculo de los frames equivalentes se tenía muy en cuenta la duración del audio grabado, a partir de la cual se calculaban el número de imágenes necesarias. Es decir, todo era muy exacto y estaba calculado totalmente a la medida del audio grabado por el usuario.

El problema aparecía en algunos casos puntuales, en los que la relación duración/framerate ocasionaba que el audio se analizase de forma incorrecta, y se generaban muy seguidos picos de silencio con otros de máximo nivel.

La solución final se consiguió utilizando un valor fijo, proporcionado por el propio micrófono del teléfono, y que nos permite leer el audio exactamente en los mismos tramos en todas las ocasiones que ejecutemos el proceso.

Como parte negativa se pierde en ocasiones la sincronización tan exacta que se obtenía al utilizar una relación directa entre el framerate y la duración, pero el resultado en líneas generales es mucho más satisfactorio.

7.3.1.4 Modificación de las características del audio

Al igual que en el caso de la generar el vídeo, la modificación de un contenido sonoro no es una funcionalidad incluida de forma natural en sistema operativo Android.

Para realizar esto, se llevó a cabo el mismo proceso de investigación por el que localicé la librería JavaCV, encontrando en este caso la librería Libsonic.

Por suerte en el caso de esta librería el proceso consiguiente de importarla y utilizarla dentro del proyecto no fue tan problemático como en el caso del vídeo.

7.3.1.5 Carga de imágenes y problemas de memoria

Es un “problema” conocido y ampliamente comentado por todos los desarrolladores de Android, el cómo afecta a la memoria del dispositivo la carga continua de imágenes (peor cuanto más grandes), y sobre todo cuando se “crean” dentro de la aplicación como elementos a utilizar por el interfaz.

Existen varias formas de utilizar las imágenes en una aplicación Android: como resource (fuente) de la aplicación directamente, o bien crear un objeto Bitmap a partir del origen de la imagen.

El problema deriva de que cuando se hace esto de manera recurrente, y no se libera en ningún momento la memoria que utilizan esos bitmaps, la aplicación puede llegar a saturar la memoria del dispositivo y cerrarse automáticamente.

El mecanismo más habitual, incluso aconsejado por Google en las guías de desarrolladores de Android, es “reciclar” esos Bitmaps creados una vez que no son necesarios. De esta forma se libera memoria y se pueden cargar nuevas imágenes cuando la aplicación lo requiera.

Sin embargo, en el caso de este proyecto, todos los bitmaps que se utilizan son necesarios en memoria, puesto que si no lo están no se pueden cargar en pantalla, y por tanto no tendríamos ni las animaciones basadas en secuencias de imágenes, ni los fotogramas que se utilizan de base para las animaciones creadas por el usuario. Además, cargar y borrar de memoria las imágenes que conforman las animaciones no es una solución posible en nuestro caso, puesto que perderíamos mucho tiempo en realizar estos procesos y una de las razones para emplear las animaciones y no los vídeos es precisamente la de aumentar la velocidad en la interacción con el usuario.

En estas condiciones, la solución fue tratar de ajustar al máximo el número y el tamaño de las imágenes que cargábamos en memoria, reaprovechando las que fuesen similares, y asegurándonos una vez más de que los ajustes eran válidos incluso para los móviles de menos capacidad.

Tras conseguir solucionar este punto para la pantalla principal (la que lleva todo el peso de imágenes de la aplicación), el problema reapareció inesperadamente al implementar la reproducción de animaciones dentro de las conversaciones del sistema de mensajería.

Sin embargo, en este caso nada nos impedía borrar de la memoria aquellas imágenes no necesarias para ese momento de la aplicación (principalmente las de las animaciones del personaje), por lo que para este caso si se implementó el mecanismo de “reciclaje”, que libera (o más bien reorganiza) la memoria al entrar y salir de la pantalla principal de la aplicación.

7.3.1.6 Control de la conexión XMPP

En un determinado momento, cuando llego el punto del proyecto en el que me dispuse a integrar el sistema de mensajería dentro del resto de la aplicación, surgió el problema de cómo mantener y controlar el objeto que manejaba la conexión de la aplicación al servidor XMPP.

En las aplicaciones de test individuales este problema nunca se había presentado, puesto que simplemente era una pantalla en la que recibías y enviabas mensajes. Sin embargo, al integrarlo en el resto de la aplicación, nos encontrábamos ante varias pantallas: una con el listado de contactos, y otra que representaba cada posible conversación con cada uno de ellos.

Además de esto, estaba la necesidad de realizar determinados cambios o actualizaciones a nivel de interfaz, según los eventos que sucediesen en relación a esta conexión (envío y llegada de mensajes).

Tras varios intentos de crear y mantener el objeto de la conexión directamente a través de las pantallas de la aplicación, quedó claro que se debía optar por una alternativa más “elegante”, a la par que efectiva, puesto que todos estos intentos siempre terminaban dando problemas y perdiendo el control de la conexión activa.

La solución final fue la de crear un servicio que se encargará totalmente de manejar todo lo relacionado con la conexión XMPP: tanto de crearla, registrar al usuario, identificarle, enviar mensajes, recibirlos, es decir lo más cercano a la mensajería XMPP, como de realizar el resto de tareas derivadas como actualizar los mensajes de la base de datos o modificar los elementos oportunos en el interfaz de usuario.

Todo esto se consiguió utilizando los propios servicios en background de los que dispone Android, que permiten ser utilizados por las clases principales de la aplicación como si fuesen interfaces, con una serie de métodos que puedan llamar cuando resulten necesarios (además del control de crear y detener el servicio).

Además de esto, las pantallas principales del sistema de mensajería fueron diseñadas para que dispusiesen una serie de métodos al "exterior", de manera que el servicio también pudiese realizar acciones relacionadas con la interfaz en base a eventos o sucesos de su propia actividad.

7.3.2 Descripción Detallada de las Clases

En esta sección debemos describir las clases más relevantes de nuestra aplicación de manera que se detallen todos sus atributos y métodos así como su descripción y función exacta.

7.3.2.1 *MainActivity*

Descripción detallada

Activity que contiene la pantalla principal de la aplicación, con todo lo relacionado con la reproducción y manejo de animaciones, video y audio.

Documentación de las funciones

private void activateUI()

Método utilizado para habilitar todos los elementos visuales de la interfaz.

private void deactivateUI()

Método utilizado para deshabilitar todos los elementos visuales de la interfaz.

private void freeMemoryAnimationFrames(AnimationDrawable animation)

Método para "liberar" los frames que contiene una animación.

Parámetros:

- ***animation*** de tipo AnimationDrawable

private void freeMemoryDrawableFrames(Drawable[] pics)

Método para "liberar" el array de frames pasado como parámetro

Parámetros:

- **pics** array de tipo Drawable
-

private void loadAnimationBody()

Método para cargar los frames asociados a la animación que se reproduce al pulsar sobre el torso del personaje.

private void loadAnimationHead()

Método para cargar los frames asociados a la animación que se reproduce al pulsar sobre la cabeza del personaje.

private void loadSpeakFrames()

Método para cargar los frames utilizados para la creación de animaciones personalizadas en base al audio.

private void playRec()

Método que ejecuta la reproducción del último audio grabado por el usuario.

private void playSound()

Método que ejecuta la reproducción aleatoria del sonido asociado a las animaciones hechas mediante frames independientes.

protected void removeSplashScreen()

Método para ocultar la pantalla del Splash al inicio de la aplicación

private void share()

Método para compartir mediante los mecanismos de los que dispone el teléfono el video generado.

private void showSplashScreen()

Método para mostrar el splash al iniciar la aplicación.

private void startRecording()

Método encargado de iniciar la grabación de audio previa a generar una nueva animación.

private void stopRec()

Método que ejecuta la reproducción del último audio grabado por el usuario.

private void stopRecording()

Método encargado de detener el proceso de grabación de audio.

Documentación de los datos

private AdView adView

Es el elemento que nos permite hacer peticiones al servicio publicitario del AdMob

AnimationDrawable animation

Elemento que contendrá las animaciones generadas por el usuario

AnimationDrawable animationBody

Objeto que contendrá la animación ejecutada al pulsar sobre el cuerpo del personaje

AnimationDrawable animationHead

Objeto que contendrá la animación ejecutada al pulsar sobre la cabeza del personaje

ImageView bckg

Objeto que representará en pantalla la imagen estática del personaje

ImageButton buttonAnim1

Elemento que lanza una de las animaciones de vídeo de la pantalla principal

ImageButton buttonAnim2

Elemento que lanza una de las animaciones de vídeo de la pantalla principal

ImageButton buttonAnim3

Elemento que lanza una de las animaciones de vídeo de la pantalla principal

ImageButton buttonAnim4

Elemento que lanza una de las animaciones de vídeo de la pantalla principal

ImageButton buttonAnim5

Elemento que lanza una de las animaciones de vídeo de la pantalla principal

ImageButton buttonAnim6

Elemento que lanza una de las animaciones de vídeo de la pantalla principal

ImageButton buttonChat

Elemento que inicia el sistema de mensajería desde la pantalla principal

ImageButton buttonHelp

Objeto que abre la información de la sección de ayuda desde la pantalla principal

ImageButton buttonPitch

Objeto que permite cambiar el tipo de pitch a utilizar en la generación de animaciones

ImageButton buttonRec

Elemento que inicia y detiene (según el momento) el proceso de grabación de audio para generar una nueva animación

ProgressBar loading

Elemento que se utiliza para mostrar gráficamente cuando se está ejecutando un proceso por el cual es necesario esperar cierto tiempo

MediaPlayer mediaPlayerSound

Objeto MediaPlayer para reproducir el audio necesario en cada momento de la aplicación

Dialog mSplashDialog

Objeto utilizado para generar y mostrar el splash al inicio de la aplicación

NotificationDialog mUrlDialog

Elemento utilizado para mostrar los interstitials de carácter publicitario

AudioRecord recorder

Objeto utilizado para grabar audio en cada proceso de generación de animaciones

Selector selector

Objeto empleado para obtener el frame equivalente a un fragmento de audio

SignalPower signal

Elemento auxiliar para obtener la intensidad de un fragmento de audio

VideoView video

Elemento de interfaz para mostrar y reproducir las animaciones de vídeo

ImageView viewAnimation

Elemento de interfaz para mostrar y reproducir las animaciones de frames

View viewBody

Vista para encuadrar el torso del personaje principal

View viewHead

Vista para encuadrar la cabeza del personaje principal

WavIO wav

Objeto que nos proporciona una serie de funciones extra para analizar archivos de audio en formato WAV

7.3.2.2 HelpActivity

Activity que representa el apartado de ayuda del sistema de animaciones. Únicamente muestra una imagen, sin más funciones o atributos.

7.3.2.3 ChatActivity

Descripción detallada

Activity que se encarga de representar la pantalla con el listado de contactos del sistema de mensajería. Nos permite acceder a cada una de las conversaciones, ya estén empezadas o no.

Documentación de las funciones

private void configure(ProviderManager pm)

Método de configuración de las principales características del sistema de mensajería.

Parámetros:

- **pm** objeto ProviderManager
-

private void getFacebookIdAndConnect(final Session session,final boolean loadFriends)

Método encargado de obtener el ID de usuario de Facebook en función de la sesión actual en la aplicación, y de actualizar el listado de contactos en base a este ID en caso de ser indicado.

Parámetros:

- **sesión** objeto sesión de Facebook
 - **loadFriends** indicador de si es necesario recargar el listado de contactos durante la operación
-

private void onClickLogin()

Método que implementa las acciones a realizar al pulsar el botón de Login en Facebook.

private void onClickLogout()

Método que implementa las acciones a realizar al pulsar el botón de Logout en Facebook

private void loadAnimationBody()

Método para cargar los frames asociados a la animación que se reproduce al pulsar sobre el torso del personaje.

public void setListAdapter()

Método para establecer el objeto Adapter asociado a nuestro listado de contactos.

private void updateFriends()

Método para actualizar el listado de contactos almacenados en la base de datos.

public void updateListAdapter()

Método para actualizar el contenido del objeto Adapter asociado a nuestro listado de contactos.

private void updateView()

Método encargado de actualizar los elementos de la interfaz, en función del estado de la sesión en Facebook y la antigüedad de nuestro listado de contactos.

Documentación de los datos

private CustomAdapterChatFriend adapter

Es el elemento que nos permite hacer peticiones al servicio publicitario del AdMob

ImageButton buttonLoginLogout

Elemento que contendrá las animaciones generadas por el usuario

ImageButton buttonRefresh

Objeto que contendrá la animación ejecutada al pulsar sobre el cuerpo del personaje

ChatFriendDataSource friendDataSource

Objeto que contendrá la animación ejecutada al pulsar sobre la cabeza del personaje

List<ChatFriend> friends

Objeto que representará en pantalla la imagen estática del personaje

ChatActivity inst

Elemento que lanza una de las animaciones de vídeo de la pantalla principal

ListView listview

Elemento que lanza una de las animaciones de vídeo de la pantalla principal

ConnectionService mBoundService

Elemento que lanza una de las animaciones de vídeo de la pantalla principal

ServiceConnection mConnection

Elemento que lanza una de las animaciones de vídeo de la pantalla principal

ChatMessageDataSource messageDataSource

Elemento que lanza una de las animaciones de vídeo de la pantalla principal

Session session

Elemento que lanza una de las animaciones de vídeo de la pantalla principal

7.3.2.4 ConversationActivity

Descripción detallada

Activity que contiene cada conversación del sistema de mensajería. Permite tanto leer los mensajes recibidos como enviados en la conversación con un contacto determinado.

Documentación de las funciones

public static boolean isInContext(String iduser)

Método utilizado para determinar si un mensaje recibido pertenece a la conversación cargada en el Activity.

Parámetros:

- **iduser** String que queremos comparar con el del usuario con el que se mantiene la conversación activa

Retorno

- **verdadero o falso** según el resultado de la comparación
-

public void setListAdapter()

Método para establecer el objeto Adapter asociado a nuestro listado de mensajes.

public void updateListAdapter(final ChatMessage message)

Método para actualizar el contenido del objeto Adapter asociado a nuestro listado de mensajes.

Parámetros:

- **message** el objeto mensaje nuevo a añadir en la conversación

Documentación de los datos

CustomAdapterChatMessage adapter

Objeto adapter que contendrá y formateará el listado de mensajes de la conversación cargada en pantalla.

ImageView imageUser

Imagen del usuario con el que se mantiene la conversación actual.

ListView listview

Objeto listview al que se cargará el adapter descrito anteriormente para mostrar los mensajes de la conversación correctamente formateados.

ConnectionService mBoundService

Referencia al servicio en background que maneja el sistema de mensajería XMPP.

ServiceConnection mConnection

Objeto que sirve para manejar la referencia al servicio anterior, y poder utilizar los métodos que proporciona adecuadamente.

ChatMessageDataSource messageData

Elemento DataSource para manejar los datos en la base de datos relacionados con los mensajes en el sistema de mensajería.

ArrayList<ChatMessage> messages

Listado de mensajes que se mostrarán en la conversación.

Button send

Botón para confirmar el envío de mensajes

EditText textMessage

Campo de texto para escribir el contenido de los mensajes a enviar

TextView textUsername

Elemento de la interfaz que contiene el nombre del usuario con el que se mantiene la conversación

7.3.2.5 PlayAnimationActivity

Descripción detallada

Clase que implementa la pantalla en la que reproducimos las animaciones enviadas a través del sistema de mensajería de la aplicación.

Documentación de las funciones

private void freeMemoryDrawableFrames(Drawable[] pics)

Método para "liberar" el array de frames pasado como parámetro.

Parámetros:

- **pics** array de elementos de tipo Drawable a liberar
-

private void loadSpeakFrames()

Método para cargar los frames utilizados para la creación de animaciones personalizadas en base al audio.

private void playRec()

Método que ejecuta la reproducción del último audio grabado por el usuario

private void stopRec()

Método que ejecuta la reproducción del último audio grabado por el usuario.

Documentación de los datos

AnimationDrawable animation

Objeto que contiene la animación a reproducir

MediaPlayer mediaPlayerSound

Reproductor multimedia para reproducir el audio asociado a la animación

Selector selector

Elemento que nos dará un frame equivalente en base a la intensidad de un fragmento del audio grabado

SignalPower signalPower

Elemento que servirá para obtener la intensidad de un fragmento de audio concreto. Será empleado por el elemento anterior

ImageView viewAnimation

Objeto imageView que contendrá la animación a reproducir

7.3.2.6 CustomAdapterChatFriend

Descripción detallada

Clase que implementa un adapter propio para formatear de la manera que nos interesa los contactos disponibles en el sistema de mensajería.

Documentación de las funciones

public void add(ChatFriend friend)

Método para añadir un nuevo elemento ChatFriend al contenido del adapter.

Parámetros:

- **friend** objeto de tipo ChatFriend a añadir
-

public void addAll(Collection<ChatFriend> friends)

Método para añadir un listado de contactos al contenido del adapter.

Parámetros:

- **friends** colección de objetos de tipo ChatFriend a añadir

Documentación de los datos

ChatFriend friend

Objeto utilizado para tratar los datos de un contacto de manera individual

List<ChatFriend> friends

Listado de contactos que constituye el contenido inicial que formateará el adapter

ChatMessageDataSource messageDataSource

Objeto DataSource para manejar información en la base de datos relacionada con los mensajes que queramos mostrar

7.3.2.7 CustomAdapterChatMessage

Descripción detallada

Clase que implementa un adapter propio para formatear de la manera que nos interesa los mensajes intercambiados en el sistema de mensajería.

Documentación de las funciones

public void add(ChatMessage message)

Método para añadir un nuevo elemento ChatMessage al contenido del adapter.

Parámetros:

- **message** objeto de tipo ChatMessage a añadir
-

public void addAll(Collection<ChatMessage> messages)

Método para añadir un listado de mensajes al contenido del adapter.

Parámetros:

- **messages** colección de objetos de tipo ChatMessage a añadir
-

Documentación de los datos

ChatMessage message

Objeto utilizado para tratar los datos de un mensaje de manera individual

List<ChatMessage> messages

Listado de mensajes que constituye el contenido inicial que formateará el adapter

7.3.2.8 MyDBHelper

Descripción detallada

Clase utilizada para instanciar la base de datos SQLite de la aplicación. Contiene la información de las tablas, sus correspondientes columnas, etc.

Documentación de las funciones

public void onCreate(SQLiteDatabase db)

Llamado al crear la base de datos. Este método es llamado automáticamente por Android cuando necesite hacer alguna operación con la base de datos y no se encuentre creada.

Parámetros:

- **db** referencia la base de datos SQLite

public void onUpgrade(SQLiteDatabase db , int oldVersion, int newVersion)

Llamado al actualizar la base de datos. Este método es llamado por Android cuando necesita actualizar la base de datos.

Parámetros:

- **db** referencia la base de datos SQLite
- **oldVersion** referencia a la versión antigua de la base de datos
- **newVersion** referencia a la nueva versión de la base de datos

Documentación de los datos

Los datos contenidos en esta clase son todos de tipo String, que definen las tablas de la base de datos, sus respectivas columnas, y las órdenes a la base de datos para crear estas nuevas tablas con la estructura adecuada (columnas, clave primaria, tipos de datos, etc)

7.3.2.9 NotificationDialog

Descripción detallada

Elemento Dialog que se utiliza para formatear las notificaciones con contenido publicitario.

7.3.2.10 ConnectionService

Descripción detallada

Clase que implementa el Servicio que se utiliza para manejar las conexiones y las operaciones realizadas con el sistema de mensajería XMPP.

Documentación de las funciones

public void connect(String USERNAME, String PASSWORD)

Método para realizar la conexión al servidor XMPP. Si el usuario no está registrado, crea su cuenta y se loguea.

Parámetros:

- **USERNAME** nombre de usuario para ejecutar la conexión/registro/login
- **PASSWORD** contraseña del usuario para ejecutar la conexión/registro/login

public void disconnect()

Método para realizar la desconexión del servidor XMPP.

public ChatMessage sendMessage(String TO, String HOST, String text)

Método para enviar un mensaje a través del sistema XMPP.

Parámetros:

- **TO** nombre de usuario destinatario del mensaje
- **HOST** host XMPP que estamos utilizando
- **Text** Cadena a enviar en el mensaje

Retorno:

- **ChatMessage message** el objeto mensaje insertado con todas las características correspondientes (leído, enviado, animación, fecha, etc)

Documentación de los datos

ConnectionConfiguration connConfig

Objeto empleado para configurar la conexión XMPP

PacketFilter filter

Elemento empleado para recibir los mensajes entrantes por el sistema de mensajería XMPP

ChatFriendDataSource friendDataSource

Objeto DataSource para manejar datos de contactos en la base de datos de la aplicación

AccountManager mAccount

Objeto que se utiliza para gestionar los registros de usuarios en el sistema XMPP

LocalBinder mBinder

Referencia utilizada para poder utilizar las funciones asociadas al servicio desde el resto de puntos de la aplicación.

ChatMessageDataSource messageDataSource

Objeto DataSource para manejar datos de mensajes en la base de datos de la aplicación, tanto de los enviados como de los recibidos

PacketListener packetListener

Elemento empleado para recibir los mensajes entrantes por el sistema de mensajería XMPP

7.3.2.11 ChatFriend

Descripción detallada

Clase que diseña la entidad que representa a un contacto/amigo del sistema de mensajería.

Documentación de las funciones

Todas las funciones presentes en esta clase son getters y setters de los atributos definidos en el siguiente apartado.

Documentación de los datos

Int int

Id del contacto respecto a su inserción en la base de datos.

String idfb

Id del contacto en facebook. Será lo que lo referencia dentro del sistema de mensajería.

String name

Nombre completo del contacto

String url

URL de la que se obtiene la imagen del usuario

String ownerid

ID del usuario de la aplicación que carga este contacto

7.3.2.12 ChatFriendDataSource

Descripción detallada

Clase que implementa los mecanismos para manejar los datos en la base de datos de los contactos del sistema de mensajería.

Documentación de las funciones

public int createChatFriend(ChatFriend friendToInsert)

Método para insertar en la base de datos un nuevo contacto.

Parámetros:

- **friendToInsert** contacto a insertar en la base de datos

Retorno:

- **id** identificador del usuario insertado
-

public boolean existsFriend(String id,String ownerId)

Método que devuelve si existe un contacto en base a su Id y el de su propietario.

Parámetros:

- **id** id de Facebook del contacto buscado
- **ownerId** identificador del usuario que cargó el contacto buscado

Retorno:

- **true** si el usuario ya existe en la base de datos
 - **false** si el usuario no existe en la base de datos
-

public List<ChatFriend> getAllFriends(String ownerId)

Método para obtener todos los contactos de la base de datos, en base al identificador del usuario "propietario" que los insertó.

Parámetros:

- **ownerId** identificador del usuario del cual queremos obtener el listado de contactos

Retorno:

- **friends** listado de contactos presentes en la base de datos
-

public ChatFriend getFriendById(String id, String ownerId)

Método que devuelve un contacto en base a su Id y el de su propietario.

Parámetros:

- **id** id de Facebook del contacto buscado
- **ownerId** identificador del usuario que cargó el contacto buscado

Retorno:

- **friend** el objeto chatfriend del contacto buscado

Documentación de los datos

SQLiteDataBase database

Referencia a la base de datos para ejecutar las operaciones contra ella.

MyDBHelper dbHelper

Objeto MyDBHelper para obtener referencias adecuadas a la base de datos

7.3.2.13 ChatMessage

Descripción detallada

Clase que diseña la entidad que representa a un mensaje del sistema de mensajería.

Documentación de las funciones

Todas las funciones presentes en esta clase son getters y setters de los atributos definidos en el siguiente apartado.

Documentación de los datos

Int int

Id del mensaje respecto a su inserción en la base de datos.

String content

Contenido del mensaje.

String date

Fecha y hora de la recepción o envío del mensaje

String id_user

Id del usuario con el que se mantiene la conversación

Int own

Entero usado a modo de booleano para determinar si el mensaje lo hemos enviado nosotros

String ownerid

Id del usuario al que pertenece ese contacto y esa conversación. Utilizado para distinguir entre posibles conversaciones con un mismo contacto por parte de diferentes usuarios del dispositivo.

Int read

Entero usado a modo de booleano para determinar si el mensaje se ha visto

Int sent

Entero usado a modo de booleano para determinar si el mensaje se ha enviado

7.3.2.14 *ChatMessageDataSource*

Descripción detallada

Clase que implementa los mecanismos para manejar los datos en la base de datos de los mensajes del sistema de mensajería.

Documentación de las funciones

public int createMessage(ChatMessage messageToInsert)

Método para crear un nuevo mensaje en la base de datos.

Parámetros:

- **messageToInsert** mensaje a insertar en la base de datos

Retorno:

- **id** entero identificador del mensaje insertado
-

public ArrayList<ChatMessage> getAllMessagesByUser(String iduser,String ownerId)

Método que devuelve el listado de mensajes intercambiado entre dos usuarios.

Parámetros:

- **iduser** id de Facebook del contacto del que buscamos los mensajes
- **ownerId** identificador del usuario que cargó el contacto buscado y que forma parte de la conversación buscada

Retorno:

- **messages** listado de mensajes que conforman la conversación que buscamos en base a los parámetros introducidos
-

public ArrayList<ChatMessage> getAllPendingMessages(String ownerId)

Método que devuelve un listado de todos los mensajes pendiente de enviar.

Parámetros:

- **ownerid** id de Facebook del usuario al que pertenecen esos mensajs pendientes

Retorno:

- **messages** listado de todos los mensajes guardados como no enviados.
-

public ChatMessage getLastMessageByUser(String iduser, String ownerId)

Método que devuelve el último mensaje intercambiado entre dos usuarios.

Parámetros:

- **idUser** id de Facebook del contacto buscado
- **ownerId** identificador del usuario que cargó el contacto buscado y que forma parte de la conversación buscada

Retorno:

- **message** objeto mensaje con la información del último mensaje de la conversación buscada.

public void setMessageSent(int idMessage)

Método para modificar un mensaje concreto como enviado.

Parámetros:

- **idMessage** identificador del mensaje a modificar

public void setMessageRead(int idMessage)

Método para modificar un mensaje concreto como leído.

Parámetros:

- **idMessage** identificador del mensaje a modificar

Documentación de los datos

SQLiteDataBase database

Referencia a la base de datos para ejecutar las operaciones contra ella.

MyDBHelper dbHelper

Objeto MyDBHelper para obtener referencias adecuadas a la base de datos

Capítulo 8. Desarrollo de las Pruebas

En este capítulo se documentaran los resultados obtenidos en las pruebas que se documentaron en los capítulos 5 y 6.

8.1 Pruebas Unitarias

En este apartado vamos a describir los resultados obtenidos al realizar las pruebas unitarias sobre los módulos independientes de la aplicación.

8.1.1 Subsistema de animaciones

| <i>Caso de Uso : Reproducir vídeos</i> | |
|---|--|
| Prueba | Resultado Esperado |
| Reproducir el vídeo alternando con la imagen estática de base | Al mismo tiempo que se lanza la reproducción del vídeo, se lanza un proceso que comprueba cada poco tiempo el estado de la reproducción (momento del metraje en el que se encuentra). Este proceso trata de asegurar que el vídeo ya se encuentre en marcha, aunque sea de manera mínima, de forma que podamos ocultar la imagen que nos sirve como momento de reposo para las animaciones del personaje, y nos permite además optimizar este proceso para una amplia gama de dispositivos (no todos los terminales reproducen un vídeo a la misma velocidad) |
| | Resultado obtenido |
| | El cambio entre la imagen estática y el movimiento se realiza correctamente, pero la aplicación (y el terminal) empieza a ejecutarse de manera muy lenta hasta que termina cerrándose por un error |
| | Solución |
| | Tras comprobar con ayuda de los logs que es lo que estaba pasando, descubrí que el proceso que realizaba las comprobaciones del estado de la reproducción nunca se detenía, y seguía ejecutándose en background de manera infinita, con los correspondientes problemas de memoria. Se pudo solucionar “matando” el proceso una vez ha cumplido su función, y añadiendo un control extra de |

| | |
|--|--|
| | ejecuciones para evitar este tipo de problemas |
|--|--|

| <u>Caso de Uso : Reproducir animaciones</u> | |
|---|---|
| Prueba | Resultado Esperado |
| Reproducir una secuencia de frames como una animación utilizando el sistema propio de Android | <p>Debemos generar una animación nativa de android (el elemento de interfaz que nos proporciona para reproducir secuencias de frames), en base a una serie de imágenes, ya sea incluidas en el proyecto o presentes en la propia memoria del teléfono.</p> <p>Debemos ser capaces de controlar el flujo de reproducción de esta animación en base a eventos del usuario (iniciar, parar, calcular final, etc).</p> |
| | Resultado obtenido |
| | La animación no se llega a reproducir, y permanece siempre estática en el primer frame |
| | <p>Solución</p> <p>Tras investigar el problema, encontré que las animaciones nativas de Android han de iniciarse de una forma muy concreta, más como si fueran procesos en vez de dar una simple orden como en el caso del reproductor multimedia.</p> <p>Después de modificar la operación para iniciar la reproducción de la forma adecuada, se solucionó el problema</p> |
| Prueba | Resultado Esperado |
| Reproducir una animación alternando con la imagen estática de base | <p>Esta prueba tiene el mismo objetivo que lo que se esperaba en el caso del vídeo, con la salvedad de que tiene como objetivo conseguir una mayor velocidad de respuesta en la ejecución de la animación.</p> <p>En este caso el evento será lanzado cuando el usuario toque zonas concretas del personaje animado, por lo que pretendemos dar una sensación más rápida y dinámica en el movimiento de la animación.</p> <p>Por otro lado, y al igual que ocurría en el caso de los vídeos, esperamos que no se produzcan parpadeos o interrupciones en la gestión y el manejo de los dos elementos visuales</p> |

| | |
|--|--|
| | (estático y en movimiento) |
| | Resultado obtenido |
| | En este caso, y teniendo en cuenta los dos errores anteriores, todo el proceso se realizó de manera correcta |

Caso de Uso : Generar animaciones propias

| Prueba | Resultado Esperado |
|---------------------------------------|--|
| Grabar audio con el teléfono | <p>Debemos ser capaces de realizar una grabación de audio con el propio dispositivo, y obtener ese audio tanto en formato “físico” (es decir, como un archivo dentro del propio teléfono) como a nivel de datos manejables directamente por la aplicación.</p> <p>De esta forma nos garantizamos dos formas diferentes de manejar el audio, para los procesos que realizaremos posteriormente.</p> |
| | Resultado obtenido |
| | El archivo se crea correctamente en el directorio previsto para ello. Al reproducirlo desde el teléfono la calidad es correcta y todo está como debería |
| Prueba | Resultado Esperado |
| Analizar un archivo de audio | <p>Debemos de ser capaces de analizar un archivo de audio (en formato WAV) desde la propia aplicación, y obtener el conjunto de datos equivalente y manejable desde nuestro propio código.</p> <p>Junto con los datos puramente de audio, también serán de utilizados los metadatos con la información asociada a este audio (duración, samplerate, etc).</p> |
| | Resultado obtenido |
| | Podemos obtener los datos equivalentes tanto de archivos wav en general, como de los grabados por nosotros desde la propia aplicación. Los datos clave, como el array de bytes y la duración del audio se obtienen perfectamente. |
| Prueba | Resultado Esperado |
| Modificar las propiedades de un audio | A partir de un conjunto de datos de audio (obtenidos por el proceso del caso anterior), debemos modificar el sonido final en base a una serie de características predefinidas (velocidad, |

| | |
|--|---|
| | <p>tono, etc).</p> <p>Para este proceso nos apoyaremos en el uso de una librería externa, la cual habrá de estar correctamente configurada previamente.</p> <p>Resultado obtenido</p> <p>Obtenemos un error al intentar ejecutar las funciones de la librería externa, y la aplicación se cierra como consecuencia del mismo.</p> <p>Solución</p> <p>El problema residía en la configuración de la librería a la hora de importarla en nuestro proyecto. No se habían respetado los nombres de los paquetes al incluirla, y no estaba localizándola dentro del proyecto.</p> <p>Se solucionó cambiando los nombres de los paquetes por los necesarios</p> |
| Prueba | Resultado Esperado |
| <p>Analizar audio para obtener frames equivalentes</p> | <p>Esta prueba se basará en gran medida en los resultados obtenidos en el proceso de análisis de audio en general, puesto que utilizará los datos arrojados por este análisis para posteriormente realizar este punto.</p> <p>Se trata de conseguir “traducir” el conjunto de datos de audio obtenidos a partir de un archivo wav, en un conjunto de “números” que representarán cada uno un frame determinado, teniendo en cuenta que cuanto menor sea el número menor será la amplitud de ese fragmento de audio (y por tanto condicionará de una forma u otra el movimiento del personaje en la animación resultante)</p> <p>Resultados obtenidos</p> <p>En varias ocasiones hubo resultados muy deficientes, aunque fueron por diversos motivos.</p> <p>La característica común de estos vídeos era, no solo una clara falta de sincronización, sino frames llevados al extremo (mínima amplitud seguida de máxima)</p> <p>Soluciones</p> |

| | |
|--|---|
| | <p>En los primeros intentos, los fallos se debían a un desajuste en la “tabla de equivalencias” entre los datos que reflejaba el análisis del audio y los frames que le correspondían.</p> <p>La solución de esta circunstancia fue un proceso de ensayo y error, analizando los datos generados para un audio con solo silencio, otro con sonido razonable, y otro con un sonido muy ruidoso. De esta forma pudimos determinar el mínimo, el máximo y la zona media, consiguiendo un ajuste y unos resultados muy satisfactorios.</p> <p>El otro problema se focalizó en algo mucho más extraño en el que solo se obtenían resultados erróneos de manera muy puntual.</p> <p>El problema resultó estar en la forma en la que se dividía el audio por tramos para su análisis, que mostraba errores y discordancias muy claros en algunos casos puntuales.</p> <p>La solución fue cambiar el criterio por el que se dividía el audio en tramos.</p> |
|--|---|

| <i>Caso de Uso : Guardar vídeo</i> | |
|---|---|
| Prueba | Resultado Esperado |
| Generar un archivo de vídeo en base a unas imágenes y un audio localizado por la aplicación | <p>Pretendemos obtener un archivo de vídeo en un formato adecuado para la reproducción en dispositivos móviles (mp4, 3gp) que sea resultado de la combinación de un conjunto de imágenes y un archivo (o datos) de audio.</p> <p>Al igual que en el caso de la modificación de audio, esta funcionalidad utilizará las funciones de una librería externa al proyecto, y habrá que tener en cuenta la integración correcta de la misma dentro de nuestra aplicación.</p> |
| | Resultado obtenido |

| | |
|--|--|
| | <p>El vídeo es generado, con una duración acorde a lo que se espera, pero sin embargo no contiene audio y solo vemos un fondo verde en lugar de las imágenes introducidas.</p> |
| | <p>Solución</p> <p>Las herramientas de la librería externa estaban funcionando, pero no estaba utilizando correctamente los parámetros de configuración del formato de vídeo, por lo que el contenido generado estaba muy posiblemente corrupto (atendiendo al resultado)</p> <p>Empleando la configuración adecuada el problema se resolvió y el vídeo era perfectamente reproducible con el contenido esperado.</p> |
| Prueba | Resultado Esperado |
| Generar un archivo de vídeo utilizando los frames de la aplicación y el audio modificado | Esta prueba busca avanzar un paso más en la integración de las funcionalidades de este subsistema, combinando de esta forma la grabación, modificación y análisis del audio, junto con el punto anterior, para obtener como resultado un vídeo equivalente a las animaciones dinámicas que generará la aplicación final. |
| | <p>Resultado obtenido</p> <p>El vídeo se genera correctamente una vez superados los problemas de las pruebas con el caso genérico</p> |

8.1.2 Subsistema de mensajería

| <u>Caso de Uso : Enviar mensaje con/sin conexión</u> | |
|--|--|
| Prueba | Resultado Esperado |
| Enviar mensajes utilizando como servidor el GTalk del correo de Google | <p>Empleando dos cuentas de correo de Gmail (una emisora y otra receptora), enviaremos mensajes desde nuestro cliente en el dispositivo (registrado como es lógico con las credenciales correspondientes de gmail).</p> <p>En este punto realizaremos el proceso en primer lugar entre nuestro cliente y el cliente web de Gmail, de manera que nos aseguremos que el cliente al menos funciona, y posteriormente estableciendo la comunicación directamente</p> |

| | |
|--|---|
| | <p>entre dos terminales utilizando nuestra aplicación (dándole ya toda la responsabilidad de enviar y recibir todos los mensajes)</p> |
| | <p>Resultado obtenido</p> |
| | <p>Los mensajes se ven correctamente, tanto en el cliente del teléfono como en el cliente web del gmail</p> |
| Prueba | Resultado Esperado |
| <p>Enviar mensajes entre terminales utilizando nuestro propio servidor</p> | <p>En esta prueba esperamos obtener los mismos resultados que en la anterior (exclusivamente entre terminales), pero dejando de un lado el servidor del Gtalk, y empleando directamente el nuestro.</p> <p>Esto implica controlar aspectos como el registro y la identificación de usuarios, además de la correcta conexión a nuestro servidor, algo que anteriormente no entraba dentro de lo implementado</p> |
| | <p>Resultados obtenidos</p> |
| | <p>En este punto encontramos numerosos problemas, todos ellos relacionados con la configuración del XMPP.</p> <p>Estos errores eran fallos al registrar usuarios, al identificarlos, y al enviar mensajes a otros contactos que no se encontraban activos en ese momento</p> |
| | <p>Solución</p> |
| | <p>Todos los errores se solucionaron modificando el archivo de configuración del servidor, y activando los paquetes y permisos necesarios para hacer uso de esa funcionalidad</p> |
| Prueba | Resultado Esperado |
| <p>Enviar mensajes entre terminal y un cliente web</p> | <p>Aunque no es una funcionalidad propia de nuestro sistema, si es un requisito indispensable a la hora de seleccionar el protocolo a utilizar en el proyecto, por lo que se realizarán pruebas exhaustivas sobre esta funcionalidad.</p> <p>Se trata de obtener los mismos resultados que en el primer caso, entre nuestro dispositivo y el cliente web de Gmail,</p> |

| | |
|--|---|
| | <p>pero empleando en nuestro caso tanto un cliente propio como nuestro servidor de las pruebas anteriores.</p> |
| | <p>Resultado obtenido</p> |
| | <p>Al disponer de un cliente móvil y un servidor correctamente configurado, esta parte del proceso funcionó correctamente (obviando las dificultades derivadas de desarrollar el cliente web)</p> |

8.2 Pruebas de Integración y del Sistema

En este apartado abordamos las pruebas que se realizan una vez se han integrado los módulos mencionados en el apartado anterior.

Ambos partes se integran para conformar el sistema final, por lo que algunas de las pruebas anteriores se repetirán de forma similar pero teniendo en cuenta el aspecto global de todo el sistema, y además se añadirán algunas nuevas en las que intervienen ambas partes del proyecto.

| Caso de Uso : Reproducir animaciones / Reproducir videos | |
|---|---|
| Prueba | Resultado Esperado |
| Reproducir animaciones y vídeos de forma consecutiva e intercalándolos, como parte del funcionamiento normal que se espera de la aplicación | El resultado esperado en esta prueba es fruto de la combinación de los resultados anteriores en los que probábamos las animaciones y los vídeos de forma independiente. Tanto la reproducción de los vídeos como las secuencias de imágenes han de reproducirse sin parpadeos, sin apreciarse el cambio entre la imagen estática y el comienzo del movimiento, y obteniendo una respuesta más rápida en el caso de las secuencias de imágenes. En esta fase de integración habrá que tener en cuenta los problemas de memoria derivados de la carga de todos los elementos multimedia al mismo tiempo. Como es evidente, no se deberán producir tales errores |
| | Resultado obtenido |
| | Tras realizar una combinación de la reproducción de una animación y posteriormente la de un vídeo, lo que debería serla imagen estática del personaje en reposo aparece como un fondo negro |
| | Solución |
| | Existe un problema de coordinación entre los procesos que coordinan los eventos de los vídeos por un lado, y los de las animaciones de imágenes por otro. Tras un análisis pormenorizado y paso por paso de cada uno de los procesos, se localiza el punto en el que no se estaba llevando a cabo la restauración de la imagen estática como visible de nuevo |

Caso de Uso : Generar animaciones propias / Guardar video / Compartir ámbito general

| <u>/Compartir en chat</u> | |
|---|---|
| Prueba | Resultado Esperado |
| Generar una animación dentro de la aplicación final | <p>Se espera la ejecución completa y sin fallos del siguiente proceso, fruto de la integración de gran parte del subsistema de animaciones:</p> <ul style="list-style-type: none"> • Inicio de grabación de audio • Detención de grabación de audio • Análisis de audio • Modificación de audio • Equivalencia audio/frames • Generación de animación en base a imágenes • Reproducción de animación sincronizada con audio • Mostrar submenú de opciones |
| | Resultado obtenido |
| | Tras la resolución de los problemas durante las pruebas unitarias de las partes individuales que conforman el proceso completo, la funcionalidad arrojó el resultado final que estábamos esperando |
| Prueba | Resultado Esperado |
| Guardar video de la animación | <p>Esta prueba también se realizó durante la fase de pruebas unitarias, por lo que el resultado esperado es el mismo que en esa ocasión.</p> <p>En este caso habrá que tener en cuenta que el proceso se realizará en conjunto con el total de la aplicación, y que habrá de saber localizar en todo momento los datos necesarios para generar el vídeo.</p> <p>Se ha de poder generar el vídeo en diferentes puntos o momentos de la aplicación, bloqueando el uso de la interfaz durante la duración del proceso, y volviendo a mostrar siempre al terminar las opciones del submenú correspondiente.</p> |
| | Resultado obtenido |
| | Una vez más el trabajo realizado durante las pruebas unitarias nos permite obtener un resultado correcto en esta parte de la integración del sistema |
| Prueba | Resultado Esperado |

| | |
|---|--|
| <p>Compartir vídeo guardado en el ámbito general</p> | <p>El propio sistema Android nos permite lanzar su pantalla genérica de publicación de contenido, y se adecua a las aplicaciones presentes en el dispositivo.</p> <p>Como parte de la integración del sistema, se añade el caso extra de que si el vídeo aún no se ha generado y se intenta lanzar esta opción, se ejecuta previamente el proceso de creación del vídeo antes de lanzar la opción de compartir.</p> <p>Resultado obtenido</p> <p>El vídeo no se comparte porque no se ha generado previamente (caso especial) y a la hora de lanzar el proceso responsable de esta función no se realiza su ejecución de manera correcta (se inicia la generación del video y de la pantalla de compartir al mismo tiempo, y no se localiza el archivo necesario)</p> <p>Solución</p> <p>Se lanza el proceso de creación del vídeo de manera “bloqueante”, para obligar al flujo de la aplicación a esperar por la finalización del proceso anterior antes de mostrarle al usuario las opciones de publicación</p> |
| <p>Prueba</p> | <p>Resultado Esperado</p> |
| <p>Compartir vídeo guardado en el chat de la aplicación</p> | <p>El audio grabado por el usuario se codifica correctamente en Base64 y es enviado correctamente a través del sistema de chat propio de la aplicación.</p> <p>Es especialmente importante que se mantenga el concepto del proceso “compartir en el chat”, con un inicio y un final, ya que siendo estrictos la aplicación se encuentra en la parte del sistema de animaciones y no del chat.</p> <p>En este sentido se habrá de controlar que una vez salga del chat, la pantalla de animaciones siga en el mismo estado que inició el proceso de compartir (con el submenú desplegado y aun manejando la misma animación).</p> <p>El audio habrá de recibirse correctamente en el receptor (aunque esto es responsabilidad del subsistema de mensajería) y deberá reproducirse la animación desde la propia conversación tanto en el emisor como en el receptor.</p> |

| | |
|--|---|
| | Este punto será tratado en pruebas sucesivas. |
| | Resultado obtenido |
| | La decodificación de audio en el momento de tratar el contenido del mensaje enviado (o recibido si nos referimos al receptor) no se realiza de forma correcta, y al tratar de reproducirlo se emite un ruido como claro síntoma del error |
| | Solución |
| | El mecanismo de codificación y decodificación en Base64 no se estaba haciendo de manera correcta, ya que se obtenían resultados diferentes para un mismo audio inicial. Se solventó esta irregularidad especificando a la funcionalidad que realizaba este proceso que utilizase las mismas características de codificación en ambos puntos. |

| <i>Caso de Uso : Enviar mensajes sin/con conexión</i> | |
|--|---|
| Prueba | Resultado Esperado |
| Enviar un mensaje con conexión a uno de nuestros contactos, esperando verse en la pantalla de conversación | Se trata de la funcionalidad básica que podemos esperar del sistema de mensajería implementado. En este parte de la integración sin embargo, tienen que confluir correctamente las siguientes partes: <ul style="list-style-type: none"> • Conexión correcta al servidor XMPP • Registro o Login del usuario (según el caso) • Carga de contactos del usuario (desde Facebook o desde la propia base de datos del terminal) • Carga de conversación con el contacto seleccionado • Envío correcto del mensaje (que será registrado en la base de datos del dispositivo) |
| | Resultado obtenido |
| | Todas las partes del proceso funcionan correctamente, y el mensaje es enviado, insertado en la base de datos de la aplicación y listado correctamente en la conversación actual |
| Prueba | Resultado Esperado |
| Enviar un mensaje con conexión a uno de nuestros | El resultado esperado es el mismo al caso anterior, con la salvedad de que en lugar de mostrar el mensaje en la |

| | |
|--|--|
| <p>contactos, esperando verse en la pantalla de contactos</p> | <p>conversación del contacto, el sistema ha de realizar la gestión del mensaje por debajo (de manera transparente al usuario) y actualizar el listado de contactos de forma correcta.</p> <p>El nuevo mensaje recibido deberá mostrarse como último mensaje en la conversación con ese contacto concreto, y visualmente destacarla para dejar claro que existe un nuevo mensaje que no se ha visto (en la parte del receptor).</p> <p>Resultado obtenido</p> <p>Desde el punto de vista de la conversación con el contacto el proceso parece correcto.</p> <p>Sin embargo desde la pantalla del contacto que recibe el mensaje, estando en la pantalla del listado de contactos, la llegada del nuevo mensaje no actualiza ni la información de la vista ni los detalles visuales</p> <p>Solución</p> <p>El servicio encargado de controlar los eventos relacionados con el sistema de mensajería no se estaba comunicando correctamente con la clase responsable de manejar los elementos de interfaz.</p> <p>Se realizaron los cambios pertinentes que permitiesen a las clases principales que manejan las vistas “ofrecer” métodos al servicio responsable de controlar los mensajes, y actualizarlo todo de manera pertinente</p> |
| <p>Prueba</p> | <p>Resultado Esperado</p> |
| <p>Enviar varios mensajes sin conexión y conectarse posteriormente</p> | <p>Cada vez que se produce una reconexión al servidor XMPP, ya sea por reiniciar la aplicación, una caída de la red, etc, se deben enviar de forma automática todos los mensajes listados en la base de datos del terminal como no enviados.</p> <p>Estos mensajes ya deberían de aparecer en sus correspondientes conversaciones listados correctamente, y de acuerdo a lo definido en las dos pruebas anteriores.</p> <p>Resultado obtenido</p> |

| | |
|--|--|
| | El gestor de la base de datos de la aplicación lista correctamente los mensajes marcados como pendientes, y se envían apropiadamente a sus destinatarios correspondientes en base a la información con la que fueron registrados |
|--|--|

| <u>Caso de Uso : Reproducir animaciones enviadas/recibidas</u> | |
|--|--|
| Prueba | Resultado Esperado |
| Reproducir una animación enviada por nosotros mismos | Se espera exactamente el mismo resultado que cuando se reproduce una animación en la pantalla principal del personaje animado. |
| Prueba | Resultado Esperado |
| Reproducir una animación recibida | Se espera exactamente el mismo resultado que cuando se reproduce una animación en la pantalla principal del personaje animado. |
| Resultado obtenido | |
| Se producen dos problemas de forma variable. O se reproduce la animación correctamente y al volver a la pantalla principal y al tratar de ejecutar las animaciones del personaje la aplicación se cierra, o bien se cierra directamente al intentar reproducirla en primera instancia. | |
| Solución | |
| <p>En ambos casos se está produciendo un problema de memoria en la aplicación por la carga de nuevas imágenes en la interfaz del sistema.</p> <p>La solución pasó por hacer una gestión activa de la memoria, en la que se descartan las imágenes empleadas por la pantalla principal de la aplicación (cuando se abandona esta vista), y se cargan las necesarias cuando queremos reproducir una animación desde el chat.</p> <p>Al volver a la vista principal, el proceso se realiza de forma inversa</p> | |

8.3 Pruebas de Usabilidad

Dentro del diseño de las pruebas del capítulo 6, se determina que para las pruebas de usabilidad se van a realizar una serie de cuestionarios a sujetos de prueba de cara a evaluar las características de la aplicación.

El primero de los cuestionarios tiene como objetivo determinar las “capacidades” del usuario concreto, mientras que el segundo es el que afecta directamente a su opinión acerca de las propiedades de la aplicación.

8.3.1 Cuestionarios del perfil 1

- Hombre de 51 años
- Sector de la alimentación

| |
|---|
| ¿Usa un smartphone frecuentemente? |
| <ol style="list-style-type: none">1. Todos los días2. Varias veces a la semana3. Ocasionalmente4. Nunca o casi nunca |
| ¿Qué tipo de actividades realiza con su teléfono? |
| <ol style="list-style-type: none">1. Es parte de mi trabajo o profesión2. Lo uso básicamente para ocio (Multimedia, Internet en general, etc)3. Únicamente para llamar4. Leo el correo y navego ocasionalmente |
| ¿Ha utilizado alguna vez una aplicación de este tipo? (Tipo Talking) |
| <ol style="list-style-type: none">1. Sí, en muchas ocasiones2. Sí, pero no habitualmente3. No, nunca |
| ¿Qué busca Vd. Principalmente cuando instala una nueva aplicación en su teléfono? |
| <ol style="list-style-type: none">1. Que sea fácil de usar2. Que no tenga publicidad molesta3. Que sea rápida4. Que funcione correctamente en mi dispositivo |

| Facilidad de Uso | Siempre | Frecuentemente | Ocasionalmente | Nunca |
|--|---------------------|-----------------------|-----------------------|----------------------|
| <i>¿Sabe dónde está dentro de la aplicación?</i> | ✓ | | | |
| <i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i> | | ✓ | | |
| <i>¿Le resulta sencillo el uso de la aplicación?</i> | | ✓ | | |
| Funcionalidad | Siempre | Frecuentemente | Ocasionalmente | Nunca |
| <i>¿Funciona cada tarea como Vd. espera?</i> | | ✓ | | |
| <i>¿El tiempo de respuesta de la aplicación es muy grande?</i> | | | | ✓ |
| <i>¿Echa en falta algún elemento que complemente la funcionalidad?</i> | | | | ✓ |
| <i>¿Se comunican de manera correcta los eventos de la aplicación?</i> | ✓ | | | |
| <i>¿Ha detectado algún error o detalle susceptible de mejora en el comportamiento del sistema?</i> | | | ✓ | |
| Calidad del Interfaz | | | | |
| Aspectos gráficos | Muy Adecuado | Adecuado | Poco Adecuado | Nada Adecuado |
| <i>El tipo y tamaño de letra es</i> | | ✓ | | |
| <i>Los iconos e imágenes usados son</i> | | ✓ | | |
| <i>Los colores empleados son</i> | | ✓ | | |
| <i>El tamaño general es</i> | | ✓ | | |
| <i>La distribución de los elementos es</i> | | ✓ | | |

| | | | | |
|--|---|-----------|-----------|----------------|
| La transición entre las pantalla es | | ✓ | | |
| La adaptación a la pantalla de mi terminal es (Modelo: Samsung Galaxy SII) | ✓ | | | |
| Diseño de la Interfaz | | Si | No | A veces |
| ¿Le resulta fácil de usar? | | ✓ | | |
| ¿El diseño de las pantallas es claro y atractivo? | | ✓ | | |
| ¿Cree que los iconos son claros respecto a la funcionalidad que presentan? | | ✓ | | |
| ¿Cree que el diseño podría mejorar en algún momento determinado? | | | | ✓ |
| Observaciones | | | | |
| - | | | | |

8.3.2 Cuestionarios del perfil 2

- Mujer de 45 años
- Ama de casa

| |
|--|
| ¿Usa un smartphone frecuentemente? |
| <ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca |
| ¿Qué tipo de actividades realiza con su teléfono? |
| <ol style="list-style-type: none"> 1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio (Multimedia, Internet en general, etc) 3. Únicamente para llamar 4. Leo el correo y navego ocasionalmente |

| ¿Ha utilizado alguna vez una aplicación de este tipo? (Tipo Talking) |
|--|
| <ol style="list-style-type: none"> 1. Sí, en muchas ocasiones 2. Sí, pero no habitualmente 3. No, nunca |
| ¿Qué busca Vd. Principalmente cuando instala una nueva aplicación en su teléfono? |
| <ol style="list-style-type: none"> 1. Que sea fácil de usar 2. Que no tenga publicidad molesta 3. Que sea rápida 4. Que funcione correctamente en mi dispositivo |

| Facilidad de Uso | Siempre | Frecuentemente | Ocasionalmente | Nunca |
|---|---------|----------------|----------------|-------|
| ¿Sabe dónde está dentro de la aplicación? | ✓ | | | |
| ¿Existe ayuda para las funciones en caso de que tenga dudas? | | | ✓ | |
| ¿Le resulta sencillo el uso de la aplicación? | | ✓ | | |
| Funcionalidad | Siempre | Frecuentemente | Ocasionalmente | Nunca |
| ¿Funciona cada tarea como Vd. espera? | | ✓ | | |
| ¿El tiempo de respuesta de la aplicación es muy grande? | | | ✓ | |
| ¿Echa en falta algún elemento que complemente la funcionalidad? | | | | ✓ |
| ¿Se comunican de manera correcta los eventos de la aplicación? | | | ✓ | |
| ¿Ha detectado algún error o detalle susceptible de mejora en el comportamiento del sistema? | | | | ✓ |
| Calidad del Interfaz | | | | |
| Aspectos gráficos | Muy | Adecuado | Poco | Nada |

Desarrollo de las Pruebas | Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario

| | Adecuado | | Adecuado | Adecuado |
|--|----------|-----------|-----------|----------------|
| <i>El tipo y tamaño de letra es</i> | | ✓ | | |
| <i>Los iconos e imágenes usados son</i> | | ✓ | | |
| <i>Los colores empleados son</i> | | ✓ | | |
| <i>El tamaño general es</i> | ✓ | | | |
| <i>La distribución de los elementos es</i> | | ✓ | | |
| <i>La transición entre las pantalla es</i> | | ✓ | | |
| <i>La adaptación a la pantalla de mi terminal es (Modelo: Samsung Galaxy Ace)</i> | | ✓ | | |
| Diseño de la Interfaz | | Si | No | A veces |
| <i>¿Le resulta fácil de usar?</i> | | ✓ | | |
| <i>¿El diseño de las pantallas es claro y atractivo?</i> | | ✓ | | |
| <i>¿Cree que los iconos son claros respecto a la funcionalidad que presentan?</i> | | | | ✓ |
| <i>¿Cree que el diseño podría mejorar en algún momento determinado?</i> | | | | ✓ |
| Observaciones | | | | |
| - | | | | |

8.3.3 Cuestionarios del perfil 3

- Hombre de 40 años
- Informático

| |
|--|
| ¿Usa un smartphone frecuentemente? |
| <ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca |
| ¿Qué tipo de actividades realiza con su teléfono? |
| <ol style="list-style-type: none"> 1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio (Multimedia, Internet en general, etc) 3. Únicamente para llamar 4. Leo el correo y navego ocasionalmente |
| ¿Ha utilizado alguna vez una aplicación de este tipo? (Tipo Talking) |
| <ol style="list-style-type: none"> 1. Sí, en muchas ocasiones 2. Sí, pero no habitualmente 3. No, nunca |
| ¿Qué busca Vd. Principalmente cuando instala una nueva aplicación en su teléfono? |
| <ol style="list-style-type: none"> 1. Que sea fácil de usar 2. Que no tenga publicidad molesta 3. Que sea rápida 4. Que funcione correctamente en mi dispositivo |

| Facilidad de Uso | Siempre | Frecuentemente | Ocasionalmente | Nunca |
|---|---------|----------------|----------------|-------|
| <i>¿Sabe dónde está dentro de la aplicación?</i> | ✓ | | | |
| <i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i> | | | ✓ | |
| <i>¿Le resulta sencillo el uso de la aplicación?</i> | ✓ | | | |

Desarrollo de las Pruebas | Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario

| Funcionalidad | Siempre | Frecuentemente | Ocasionalmente | Nunca |
|---|--------------|----------------|----------------|---------------|
| ¿Funciona cada tarea como Vd. espera? | | ✓ | | |
| ¿El tiempo de respuesta de la aplicación es muy grande? | | | | ✓ |
| ¿Echa en falta algún elemento que complemente la funcionalidad? | | | | ✓ |
| ¿Se comunican de manera correcta los eventos de la aplicación? | ✓ | | | |
| ¿Ha detectado algún error o detalle susceptible de mejora en el comportamiento del sistema? | | | | ✓ |
| Calidad del Interfaz | | | | |
| Aspectos gráficos | Muy Adecuado | Adecuado | Poco Adecuado | Nada Adecuado |
| El tipo y tamaño de letra es | | ✓ | | |
| Los iconos e imágenes usados son | | ✓ | | |
| Los colores empleados son | | ✓ | | |
| El tamaño general es | | ✓ | | |
| La distribución de los elementos es | | | ✓ | |
| La transición entre las pantalla es | | ✓ | | |
| La adaptación a la pantalla de mi terminal es (Modelo: Samsung Galaxy S) | ✓ | | | |
| Diseño de la Interfaz | | Si | No | A veces |
| ¿Le resulta fácil de usar? | | ✓ | | |

| | | | |
|---|---|--|---|
| <i>¿El diseño de las pantallas es claro y atractivo?</i> | ✓ | | |
| <i>¿Cree que los iconos son claros respecto a la funcionalidad que presentan?</i> | ✓ | | |
| <i>¿Cree que el diseño podría mejorar en algún momento determinado?</i> | | | ✓ |
| Observaciones | | | |
| El usuario indica que sería positivo aumentar el tamaño de los botones de las animaciones en la pantalla principal, para hacerlos más directos e intuitivos | | | |

8.3.4 Cuestionarios del perfil 4

- Hombre de 28 años
- Ingeniero de Telecomunicaciones

| |
|--|
| ¿Usa un smartphone frecuentemente? |
| <ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca |
| ¿Qué tipo de actividades realiza con su teléfono? |
| <ol style="list-style-type: none"> 1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio (Multimedia, Internet en general, etc) 3. Únicamente para llamar 4. Leo el correo y navego ocasionalmente |
| ¿Ha utilizado alguna vez una aplicación de este tipo? (Tipo Talking) |
| <ol style="list-style-type: none"> 1. Sí, en muchas ocasiones 2. Sí, pero no habitualmente 3. No, nunca |
| ¿Qué busca Vd. Principalmente cuando instala una nueva aplicación en su teléfono? |
| <ol style="list-style-type: none"> 1. Que sea fácil de usar 2. Que no tenga publicidad molesta 3. Que sea rápida 4. Que funcione correctamente en mi dispositivo |

| Facilidad de Uso | Siempre | Frecuentemente | Ocasionalmente | Nunca |
|--|---------|----------------|----------------|-------|
| ¿Sabe dónde está dentro de la aplicación? | ✓ | | | |
| ¿Existe ayuda para las funciones en caso de que tenga dudas? | | | ✓ | |

| | | | | |
|--|---------------------|-----------------------|-----------------------|----------------------|
| <i>¿Le resulta sencillo el uso de la aplicación?</i> | ✓ | | | |
| Funcionalidad | Siempre | Frecuentemente | Ocasionalmente | Nunca |
| <i>¿Funciona cada tarea como Vd. espera?</i> | | ✓ | | |
| <i>¿El tiempo de respuesta de la aplicación es muy grande?</i> | | | | ✓ |
| <i>¿Echa en falta algún elemento que complemente la funcionalidad?</i> | | | | ✓ |
| <i>¿Se comunican de manera correcta los eventos de la aplicación?</i> | | ✓ | | |
| <i>¿Ha detectado algún error o detalle susceptible de mejora en el comportamiento del sistema?</i> | | | | ✓ |
| Calidad del Interfaz | | | | |
| Aspectos gráficos | Muy Adecuado | Adecuado | Poco Adecuado | Nada Adecuado |
| <i>El tipo y tamaño de letra es</i> | | ✓ | | |
| <i>Los iconos e imágenes usados son</i> | ✓ | | | |
| <i>Los colores empleados son</i> | | ✓ | | |
| <i>El tamaño general es</i> | | ✓ | | |
| <i>La distribución de los elementos es</i> | | ✓ | | |
| <i>La transición entre las pantalla es</i> | | ✓ | | |
| <i>La adaptación a la pantalla de mi terminal es (Modelo: Samsung Galaxy SIII)</i> | | ✓ | | |
| Diseño de la Interfaz | | Si | No | A veces |
| <i>¿Le resulta fácil de usar?</i> | | ✓ | | |

Desarrollo de las Pruebas | Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario

| | | | |
|---|---|--|---|
| <i>¿El diseño de las pantallas es claro y atractivo?</i> | ✓ | | |
| <i>¿Cree que los iconos son claros respecto a la funcionalidad que presentan?</i> | ✓ | | |
| <i>¿Cree que el diseño podría mejorar en algún momento determinado?</i> | | | ✓ |
| Observaciones | | | |
| - | | | |

8.3.5 Cuestionarios del perfil 5

- Hombre de 28 años
- Informático

| |
|--|
| ¿Usa un smartphone frecuentemente? |
| <ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca |
| ¿Qué tipo de actividades realiza con su teléfono? |
| <ol style="list-style-type: none"> 1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio (Multimedia, Internet en general, etc) 3. Únicamente para llamar 4. Leo el correo y navego ocasionalmente |
| ¿Ha utilizado alguna vez una aplicación de este tipo? (Tipo Talking) |
| <ol style="list-style-type: none"> 1. Sí, en muchas ocasiones 2. Sí, pero no habitualmente 3. No, nunca |
| ¿Qué busca Vd. Principalmente cuando instala una nueva aplicación en su teléfono? |
| <ol style="list-style-type: none"> 1. Que sea fácil de usar 2. Que no tenga publicidad molesta 3. Que sea rápida 4. Que funcione correctamente en mi dispositivo |

| Facilidad de Uso | Siempre | Frecuentemente | Ocasionalmente | Nunca |
|---|---------|----------------|----------------|-------|
| <i>¿Sabe dónde está dentro de la aplicación?</i> | ✓ | | | |
| <i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i> | | | ✓ | |
| <i>¿Le resulta sencillo el uso de la aplicación?</i> | | ✓ | | |

Desarrollo de las Pruebas | Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario

| Funcionalidad | Siempre | Frecuentemente | Ocasionalmente | Nunca |
|---|--------------|----------------|----------------|---------------|
| ¿Funciona cada tarea como Vd. espera? | ✓ | | | |
| ¿El tiempo de respuesta de la aplicación es muy grande? | | | | ✓ |
| ¿Echa en falta algún elemento que complemente la funcionalidad? | | | | ✓ |
| ¿Se comunican de manera correcta los eventos de la aplicación? | ✓ | | | |
| ¿Ha detectado algún error o detalle susceptible de mejora en el comportamiento del sistema? | | | | ✓ |
| Calidad del Interfaz | | | | |
| Aspectos gráficos | Muy Adecuado | Adecuado | Poco Adecuado | Nada Adecuado |
| El tipo y tamaño de letra es | | ✓ | | |
| Los iconos e imágenes usados son | | ✓ | | |
| Los colores empleados son | ✓ | | | |
| El tamaño general es | ✓ | | | |
| La distribución de los elementos es | ✓ | | | |
| La transición entre las pantalla es | | ✓ | | |
| La adaptación a la pantalla de mi terminal es (Modelo: Nexus One) | ✓ | | | |
| Diseño de la Interfaz | | Si | No | A veces |
| ¿Le resulta fácil de usar? | | ✓ | | |

| | | | |
|---|---|--|---|
| <i>¿El diseño de las pantallas es claro y atractivo?</i> | ✓ | | |
| <i>¿Cree que los iconos son claros respecto a la funcionalidad que presentan?</i> | | | ✓ |
| <i>¿Cree que el diseño podría mejorar en algún momento determinado?</i> | | | ✓ |
| Observaciones | | | |
| - | | | |

8.3.6 Cuestionarios del perfil 6

- ✓ Niño de 3 años (este cuestionario se centrará principalmente en las funcionalidades de la pantalla de animaciones)

| |
|--|
| ¿Usa un smartphone frecuentemente? |
| <ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca |
| ¿Qué tipo de actividades realiza con su teléfono? |
| <ol style="list-style-type: none"> 1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio (Multimedia, Internet en general, etc) 3. Únicamente para llamar 4. Leo el correo y navego ocasionalmente |
| ¿Ha utilizado alguna vez una aplicación de este tipo? (Tipo Talking) |
| <ol style="list-style-type: none"> 1. Sí, en muchas ocasiones 2. Sí, pero no habitualmente 3. No, nunca |
| ¿Qué busca Vd. Principalmente cuando instala una nueva aplicación en su teléfono? |
| <ol style="list-style-type: none"> 1. Que sea fácil de usar 2. Que no tenga publicidad molesta 3. Que sea rápida 4. Que funcione correctamente en mi dispositivo |

| Facilidad de Uso | Siempre | Frecuentemente | Ocasionalmente | Nunca |
|--|---------|----------------|----------------|-------|
| ¿Sabe dónde está dentro de la aplicación? | ✓ | | | |
| ¿Existe ayuda para las funciones en caso de que tenga dudas? | - | - | - | - |

| | | | | |
|--|---------------------|-----------------------|-----------------------|----------------------|
| <i>¿Le resulta sencillo el uso de la aplicación?</i> | ✓ | | | |
| Funcionalidad | Siempre | Frecuentemente | Ocasionalmente | Nunca |
| <i>¿Funciona cada tarea como Vd. espera?</i> | ✓ | | | |
| <i>¿El tiempo de respuesta de la aplicación es muy grande?</i> | | | | ✓ |
| <i>¿Echa en falta algún elemento que complemente la funcionalidad?</i> | - | - | - | - |
| <i>¿Se comunican de manera correcta los eventos de la aplicación?</i> | - | - | - | - |
| <i>¿Ha detectado algún error o detalle susceptible de mejora en el comportamiento del sistema?</i> | - | - | - | - |
| Calidad del Interfaz | | | | |
| Aspectos gráficos | Muy Adecuado | Adecuado | Poco Adecuado | Nada Adecuado |
| <i>El tipo y tamaño de letra es</i> | - | - | - | - |
| <i>Los iconos e imágenes usados son</i> | ✓ | | | |
| <i>Los colores empleados son</i> | ✓ | | | |
| <i>El tamaño general es</i> | | | ✓ | |
| <i>La distribución de los elementos es</i> | ✓ | | | |
| <i>La transición entre las pantalla es</i> | ✓ | | | |
| <i>La adaptación a la pantalla de mi terminal es (Modelo: Samsung Galaxy S)</i> | ✓ | | | |
| Diseño de la Interfaz | | Si | No | A veces |
| <i>¿Le resulta fácil de usar?</i> | | ✓ | | |

Desarrollo de las Pruebas | Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario

| | | | |
|---|---|--|--|
| <i>¿El diseño de las pantallas es claro y atractivo?</i> | ✓ | | |
| <i>¿Cree que los iconos son claros respecto a la funcionalidad que presentan?</i> | ✓ | | |
| <i>¿Cree que el diseño podría mejorar en algún momento determinado?</i> | ✓ | | |
| Observaciones | | | |
| <p>Parece que resulta más atractivo uno de los sonidos aleatorios de las animaciones centrales frente al otro, por lo que podría ser una buena idea darle mayor presencia a este sonido.</p> <p>Los iconos en general serían mucho más accesibles para este tipo de usuario con un mayor tamaño</p> | | | |

8.3.7 Cuestionarios del perfil 7

- Mujer de 22 años
- Química

| |
|--|
| ¿Usa un smartphone frecuentemente? |
| <ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca |
| ¿Qué tipo de actividades realiza con su teléfono? |
| <ol style="list-style-type: none"> 1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio (Multimedia, Internet en general, etc) 3. Únicamente para llamar 4. Leo el correo y navego ocasionalmente |
| ¿Ha utilizado alguna vez una aplicación de este tipo? (Tipo Talking) |
| <ol style="list-style-type: none"> 1. Sí, en muchas ocasiones 2. Sí, pero no habitualmente 3. No, nunca |
| ¿Qué busca Vd. Principalmente cuando instala una nueva aplicación en su teléfono? |
| <ol style="list-style-type: none"> 1. Que sea fácil de usar 2. Que no tenga publicidad molesta 3. Que sea rápida 4. Que funcione correctamente en mi dispositivo |

| Facilidad de Uso | Siempre | Frecuentemente | Ocasionalmente | Nunca |
|---|---------|----------------|----------------|-------|
| <i>¿Sabe dónde está dentro de la aplicación?</i> | ✓ | | | |
| <i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i> | | ✓ | | |
| <i>¿Le resulta sencillo el uso de la aplicación?</i> | ✓ | | | |

Desarrollo de las Pruebas | Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario

| Funcionalidad | Siempre | Frecuentemente | Ocasionalmente | Nunca |
|---|--------------|----------------|----------------|---------------|
| ¿Funciona cada tarea como Vd. espera? | | ✓ | | |
| ¿El tiempo de respuesta de la aplicación es muy grande? | | | | ✓ |
| ¿Echa en falta algún elemento que complemente la funcionalidad? | | | | ✓ |
| ¿Se comunican de manera correcta los eventos de la aplicación? | | ✓ | | |
| ¿Ha detectado algún error o detalle susceptible de mejora en el comportamiento del sistema? | | | ✓ | |
| Calidad del Interfaz | | | | |
| Aspectos gráficos | Muy Adecuado | Adecuado | Poco Adecuado | Nada Adecuado |
| El tipo y tamaño de letra es | ✓ | | | |
| Los iconos e imágenes usados son | ✓ | | | |
| Los colores empleados son | ✓ | | | |
| El tamaño general es | ✓ | | | |
| La distribución de los elementos es | | ✓ | | |
| La transición entre las pantalla es | | ✓ | | |
| La adaptación a la pantalla de mi terminal es (Modelo: Samsung Galaxy Ace) | | ✓ | | |
| Diseño de la Interfaz | | Si | No | A veces |
| ¿Le resulta fácil de usar? | | ✓ | | |

| | | | |
|---|---|--|---|
| <i>¿El diseño de las pantallas es claro y atractivo?</i> | ✓ | | |
| <i>¿Cree que los iconos son claros respecto a la funcionalidad que presentan?</i> | ✓ | | |
| <i>¿Cree que el diseño podría mejorar en algún momento determinado?</i> | | | ✓ |
| Observaciones | | | |
| - | | | |

8.3.8 Cuestionarios del perfil 8

- Hombre de 36 años
- Profesional TIC

| |
|--|
| ¿Usa un smartphone frecuentemente? |
| <ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca |
| ¿Qué tipo de actividades realiza con su teléfono? |
| <ol style="list-style-type: none"> 1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio (Multimedia, Internet en general, etc) 3. Únicamente para llamar 4. Leo el correo y navego ocasionalmente |
| ¿Ha utilizado alguna vez una aplicación de este tipo? (Tipo Talking) |
| <ol style="list-style-type: none"> 1. Sí, en muchas ocasiones 2. Sí, pero no habitualmente 3. No, nunca |
| ¿Qué busca Vd. Principalmente cuando instala una nueva aplicación en su teléfono? |
| <ol style="list-style-type: none"> 1. Que sea fácil de usar 2. Que no tenga publicidad molesta 3. Que sea rápida 4. Que funcione correctamente en mi dispositivo |

| Facilidad de Uso | Siempre | Frecuentemente | Ocasionalmente | Nunca |
|--|---------|----------------|----------------|-------|
| ¿Sabe dónde está dentro de la aplicación? | ✓ | | | |
| ¿Existe ayuda para las funciones en caso de que tenga dudas? | | | ✓ | |

| | | | | |
|--|---------------------|-----------------------|-----------------------|----------------------|
| <i>¿Le resulta sencillo el uso de la aplicación?</i> | ✓ | | | |
| Funcionalidad | Siempre | Frecuentemente | Ocasionalmente | Nunca |
| <i>¿Funciona cada tarea como Vd. espera?</i> | | ✓ | | |
| <i>¿El tiempo de respuesta de la aplicación es muy grande?</i> | | | ✓ | |
| <i>¿Echa en falta algún elemento que complemente la funcionalidad?</i> | | | | ✓ |
| <i>¿Se comunican de manera correcta los eventos de la aplicación?</i> | | ✓ | | |
| <i>¿Ha detectado algún error o detalle susceptible de mejora en el comportamiento del sistema?</i> | | | | ✓ |
| Calidad del Interfaz | | | | |
| Aspectos gráficos | Muy Adecuado | Adecuado | Poco Adecuado | Nada Adecuado |
| <i>El tipo y tamaño de letra es</i> | | ✓ | | |
| <i>Los iconos e imágenes usados son</i> | | ✓ | | |
| <i>Los colores empleados son</i> | | ✓ | | |
| <i>El tamaño general es</i> | | ✓ | | |
| <i>La distribución de los elementos es</i> | | ✓ | | |
| <i>La transición entre las pantalla es</i> | | ✓ | | |
| <i>La adaptación a la pantalla de mi terminal es (Modelo: Samsung Glaxy Note 2)</i> | | | ✓ | |
| Diseño de la Interfaz | | Si | No | A veces |
| <i>¿Le resulta fácil de usar?</i> | | ✓ | | |

| | | | |
|--|---|--|---|
| ¿El diseño de las pantallas es claro y atractivo? | ✓ | | |
| ¿Cree que los iconos son claros respecto a la funcionalidad que presentan? | ✓ | | |
| ¿Cree que el diseño podría mejorar en algún momento determinado? | | | ✓ |
| Observaciones | | | |
| Este usuario en concreto indicó que sería más interesante reproducir la animación directamente después de generarla, en lugar de crear el vídeo inicialmente y obligar al usuario a esperar a que terminase el proceso (netamente más largo que generar solo la animación) | | | |

8.3.9 Conclusiones sobre la usabilidad

Mediante los perfiles anteriores se ha intentado conseguir una cierta variedad en las características (sexo, edad, estudios, profesión) y capacidades de cada sujeto de prueba, de manera que las conclusiones extraídas de las respuestas a los cuestionarios anteriores sobre la usabilidad de nuestro sistema nos proporcionen una estimación general de la cercanía de nuestra aplicación a todo tipo de usuarios, atiendo a factores como facilidad de uso, calidad de diseño, buen funcionamiento, etc.

En lo relacionado con la parte del diseño y estructura de la interfaz de la aplicación, se pueden sacar conclusiones muy positivas, obteniendo resultados favorables en lo referente a todas las cuestiones sobre este aspecto. Como excepción cabe destacar lo reflejado en las pruebas realizadas por el niño de 3 años y el informático de 40 (su padre), en las que se apreció como una posible mejora muy positiva la de aumentar el tamaño de todos los botones del sistema, haciendo más fácil y directa la interacción con el mismo.

Respecto a aspectos de funcionalidad, cabe destacar únicamente la observación del profesional TIC, que indicó que sería mejor que el usuario generase el vídeo a posteriori y como una opción voluntaria, y no obligarle desde el principio a esperar todo el proceso; de esta forma, mostrándole primero la animación (de manera mucho más rápida y ágil), podría decidir si la animación es la que deseaba y posteriormente guardarla como vídeo.

8.4 Pruebas de Rendimiento

Como ya se ha comentado anteriormente, las principales dificultades a nivel de rendimiento con las que nos enfrentamos durante el desarrollo del proyecto, estaban relacionadas con el sistema de gestión de animaciones, y más concretamente con la forma de alternar de manera imperceptible el contenido en movimiento con el contenido estático.

Entre todos los utilizados para las pruebas, los principales terminales tomados como referencia mínima fueron el Samsung Galaxy Ace y el Nexus One.

Los resultados de las pruebas con estos teléfonos nos proporcionaron la información necesaria para tomar dos decisiones muy importantes en lo referente a la implementación del gestor de animaciones:

- Utilizar parámetros de transparencia en lugar de visibilidad parece ser mucho más efectivo a nivel de rendimiento durante la ejecución de la aplicación.
- Implementar un mecanismo de “chequeo” de la información del reproductor para garantizar una mejor sincronización a la hora de reproducir las animaciones de vídeo.

Capítulo 9. Manuales del Sistema

9.1 Manual de Instalación

Al tratarse de una aplicación para dispositivos Android, el proceso de instalación es muy rápido y sencillo, aunque si consta de alguna peculiaridad que tendremos en cuenta en este apartado.

Como condición inicial, cabe destacar que se requiere un dispositivo con una versión mínima de Android 2.2.

La forma más sencilla de instalar una aplicación en un dispositivo Android es acudir directamente al “Play Store”, que es el principal distribuidor oficial de aplicaciones Android. Sin embargo, como estamos hablando de una aplicación independiente creada para este mismo proyecto, deberemos ejecutar la instalación directamente desde el archivo original de la aplicación, presente en uno de estos dos lugares:

- Enlace a internet donde esté alojado el apk final
- Directorio de la entrega donde esté presente el apk final

Una vez obtenido el fichero .apk de uno de los dos puntos anteriores, simplemente hay que abrirlo dentro del teléfono e iniciará la instalación. Sin embargo, antes de esto, habrá que activar en nuestro terminal la opción “Fuentes desconocidas”, que nos permite instalar aplicaciones de orígenes distintos al Play Store oficial.

Para ello, basta con ir en nuestro teléfono a la sección “Ajustes” --> “Seguridad” --> “Fuentes desconocidas”, y activar el “tick” de dicha opción.

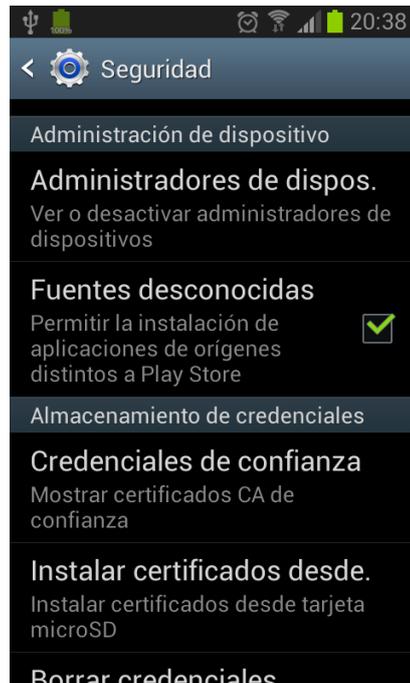


Figura 41 . Menu de ajustes preinstalación

Tras esto se iniciará el proceso de instalación, aunque previamente nos pedirá que confirmemos los permisos de la aplicación. Estos permisos se han limitado al máximo posible teniendo en cuenta las funcionalidades del sistema, y no se encuentra ninguno que pueda ser considerado abusivo.



Figura 42 . Permisos de la aplicación

Una vez aceptados los permisos, se iniciará la instalación y cuando esta termine podremos abrir ya la aplicación y comenzar a utilizarla sin problemas, y sin descargas adicionales.

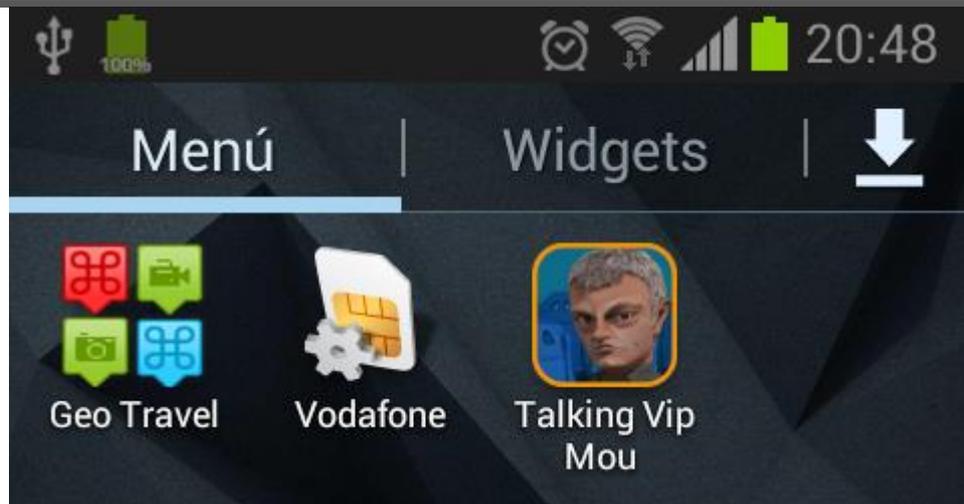


Figura 43 . Apliación instalada correctamente

9.2 Manual de Usuario

En este manual de usuario trataremos de reflejar y explicar de manera clara y sencilla el uso de todas las funciones de la aplicación, aunque como se ha remarcado en muchas ocasiones durante esta documentación todas ellas han sido implementadas de la manera más intuitiva posible, de forma que su uso no ofrezca dudas o problemas en ningún momento.

Para ser más claros y organizar la información de la mejor manera posible, estructuraremos este manual en dos partes: por un lado el sistema de animaciones (la pantalla principal), y por otro el de mensajería (las dos pantallas siguientes).

9.2.1 Sistema de animaciones

Todo lo relacionado con el sistema de animaciones está presente en la pantalla principal de la aplicación, en la cual podemos observar al personaje en la zona central de la pantalla, y el resto de botones que implementan cada funcionalidad distribuidos a su alrededor.

9.2.1.1 Reproducción de animaciones “simples”

Entendemos como animaciones simples las que representan un movimiento rápido y corto del personaje. Estas dos animaciones se desencadenan cuando pulsamos en la cabeza o en el torso del personaje (una animación por cada una de estas partes).

Al pulsar sobre estos dos lugares, se iniciará un movimiento del personaje que simulará su reacción a la pulsación que acabamos de hacer, y mientras el corto periodo de tiempo que dure la reproducción de la animación el resto de elementos de la interfaz estarán desactivados. Este punto puede apreciarse fácilmente en este momento, y durante el uso general de la aplicación, ya que aquellos botones que se encuentren deshabilitados tendrán un color más “apagado” que el habitual.



Figura 44 . Captura que muestra los botones del interfaz deshabilitados

Estas dos animaciones irán acompañadas de un pequeño efecto sonoro aleatorio entre dos posibilidades (teniendo una más probabilidad de aparecer que la otra por los motivos explicados en apartados anteriores).

9.2.1.2 Reproducción de animaciones “complejas”

Entendemos como animaciones complejas aquellos que son más elaboradas, en las que el personaje realiza acciones más vistosas, y que en definitiva tienen una mayor duración o contenido que las anteriores.

Estas animaciones se accionan pulsando en alguno de los 6 botones de la parte inferior de la pantalla, distribuidos tanto a la izquierda como a la derecha de la misma. Además, al igual que en el caso anterior, durante la ejecución de estas animaciones el resto de botones de la aplicación permanecerán deshabilitados.



Figura 45 . Botones de las animaciones de vídeo

Es importante destacar que la reproducción de las animaciones no se puede detener o cancelar en ningún momento, y en caso de pulsar el botón “Atrás” o “Home” del terminal, la aplicación se cerrará por completo.

9.2.1.3 Grabación de animaciones propias

Esta es la principal funcionalidad de esta parte de la aplicación, y se ejecuta pulsando en el botón rojo de “Rec” de la parte superior derecha de la pantalla (el más grande y llamativo de toda la interfaz del sistema).

Cabe destacar que existen dos formas de obtener la modificación de audio posterior: en pitch (tono) grave, y en otro agudo. Esta modificación se determina con el botón posicionado en tercera posición, en la parte superior izquierda de la pantalla.



Figura 46 . Pitch agudo activado



Figura 47 . Pitch grave activado

Una vez se pulse en dicho botón, comenzará el proceso de grabación, identificado visualmente de forma clara ya que el personaje cambiará de posición, hacia un gesto que indica que está “escuchando”.



Figura 48 . Cambio de posición del personaje durante el proceso de grabación

Una vez se inicia el proceso, este se detendrá de dos formas diferentes: o bien el usuario vuelve a pulsar en el botón de REC para detener el proceso cuando termina de hablar, o bien el proceso se detiene automáticamente a los 30 segundos.

Cuando el proceso de grabación se detiene, el personaje volverá a su postura de reposo habitual, y comenzará el proceso de análisis y generación de la animación, identificado claramente por un icono reconocible de “cargando” o similar. La longitud de este proceso de espera dependerá en gran medida de la longitud del audio grabado, pero en líneas generales es un proceso considerablemente rápido e inmediato.

En el momento que se termina de generar la nueva animación creada, esta empezará a reproducirse automáticamente para que el usuario pueda ver el resultado del audio que acaba de grabar. Tras la reproducción, se desplegará desde la parte inferior de la pantalla, el menú de opciones de la animación, que completará el resto de funcionalidades de la aplicación, y que detallaremos en el siguiente apartado.

Cabe destacar que la reproducción de las animaciones del usuario puede detenerse pulsando en una ocasión el botón “Atrás” del terminal, y provocando que se despliegue automáticamente el menú de opciones de la aplicación.

9.2.1.4 Menú de opciones de la animación

Como hemos comentado en el punto anterior, este menú ofrece el resto de funcionalidades de la aplicación, y está estrechamente relacionado con la animación que se acaba de generar.

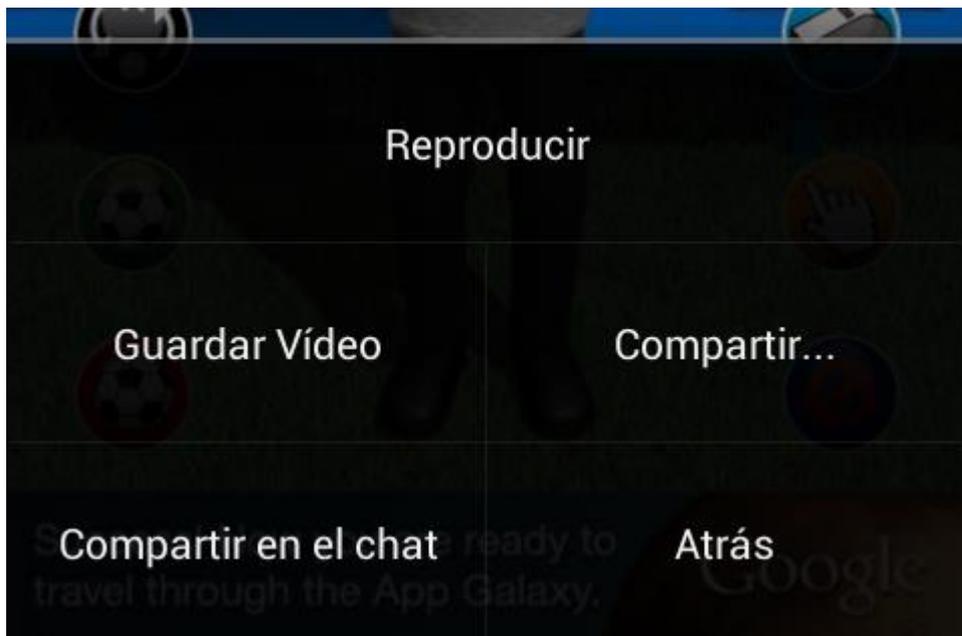


Figura 49 . Opciones del menú de animaciones

A continuación detallaremos por separado el funcionamiento de cada una de estas opciones.

9.2.1.4.1 Reproducir

Pulsando en la opción de reproducir tenemos la posibilidad de volver a ver (y escuchar) la animación generada las veces que deseemos.

La reproducción de la animación se puede detener pulsando en el botón “Atrás” del terminal.

9.2.1.4.2 Guardar vídeo

Esta opción nos permite generar un vídeo a partir de la animación creada, y nos lo guarda dentro del teléfono. Durante el proceso de creación, se muestra claramente un indicador de proceso en curso.

Posteriormente podremos acceder a todos los vídeos que hemos guardado dentro de la carpeta “Mou” en la galería multimedia de nuestro dispositivo.

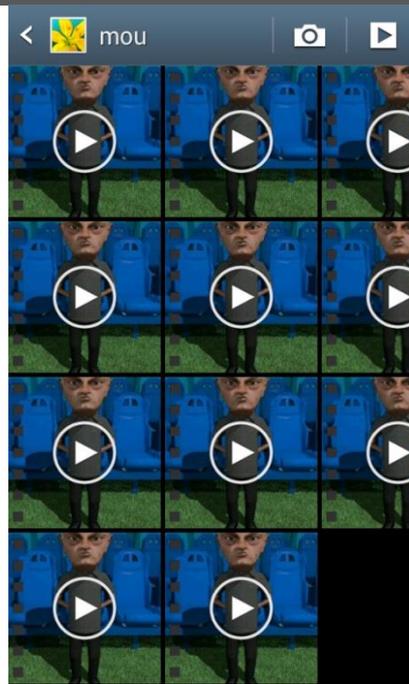


Figura 50 . Captura de los vídeos generados en la galería multimedia del dispositivo

La duración del proceso es más larga que en el caso de las animaciones, y dependerá una vez más de la longitud del audio grabado.

Cada animación solo puede guardarse como vídeo una sola vez, y en caso de reintentarlo se mostrará un mensaje informando de que ese vídeo ya se encuentra guardado en el dispositivo.

9.2.1.4.3 Compartir

Este botón nos da la posibilidad de compartir el vídeo resultante de nuestra animación a través de diferentes medios.

El mecanismo utilizado para ello es el que Android ofrece de manera nativa, y el menú desplegado no presenta diferencias respecto a lo que vemos cuando intentamos compartir cualquier elemento desde la propia interfaz del sistema operativo. Los medios disponibles para compartir el vídeo variarán según las aplicaciones instaladas en cada dispositivo.

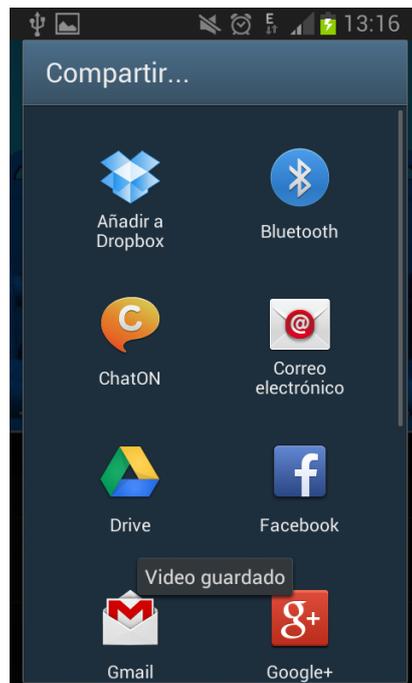


Figura 51 . Menú desplegado tras la opción de Compartir

Si al pulsar esta opción aún no hemos generado el vídeo, se iniciará previamente el proceso de creación del vídeo, y al finalizar se desplegará el menú para publicarlo.

9.2.1.4.4 Compartir en el chat

Esta opción es similar a la anterior, puesto que nos permite compartir la animación creada, pero tiene la particularidad de que la animación se publica como un mensaje más dentro del sistema de mensajería interno de la aplicación.

En este caso no se crea ningún vídeo, sino que se limita al máximo posible la transferencia de datos para hacer la operación más rápida y ligera para el usuario y el terminal.

Una vez se ha pulsado en la opción, el listado de conversaciones con amigos del chat es desplegado, y debemos seleccionar a uno de ellos para compartir la animación. Una vez seleccionado, el mensaje es enviado directamente a esa persona y se muestra la conversación completa con el contacto, con la animación como el último mensaje de la conversación.

Esta opción está limitada a una duración de 15 segundos para optimizar el proceso de envío y evitar errores inesperados de memoria.

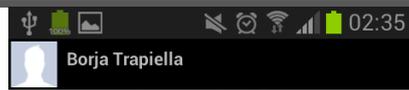


Figura 52 . Animación enviada mediante el sistema de mensajería de la aplicación

La reproducción de estas animaciones desde el propio chat se explicará en el apartado correspondiente del manual del sistema de mensajería.

9.2.1.4.5 Atrás

Esta opción simplemente cierra el menú, y nos devuelve al estado inicial de la aplicación.

9.2.1.5 Ayuda

Existe una sección de ayuda disponible con información a modo de resumen de las funcionalidades presentes en esta pantalla. Para abrir esta sección hay que pulsar en el botón de ayuda localizado en parte superior izquierda de la pantalla, en primera posición.



Figura 53 . Botoón de ayuda en la pantalla de animaciones

ayuda

- Háblale y deja que repita lo que dices
- Golpéale en la cabeza o en la tripa
- Graba videos y compártelos en Youtube, Facebook o vía email
- Pulsa los icones y verás que pasa!



Figura 54 . Pantalla de ayuda del sistema de animaciones

9.2.2 Sistema de mensajería

En este apartado describiremos el funcionamiento de todas las funciones del sistema de mensajería, de la misma forma que en el caso del uso de las animaciones.

9.2.2.1 Sincronización con Facebook

La sincronización de la cuenta de Facebook del usuario es esencial para el funcionamiento del chat, ya que los contactos con los que se podrá comunicar serán sus amigos de esta red social que también tengan instalada la aplicación.

Haciendo click en el botón “Conectar” de la parte superior derecha de la pantalla, se iniciará el proceso de Login a través de la propia aplicación de Facebook. En caso de que el usuario no la tenga instalada en su terminal, se desplegará un cuadro en la pantalla que le permitirá hacer el registro directamente desde este punto.



Figura 55 . Login con Facebook

Una vez se ha identificado con éxito, le solicitará los permisos que el usuario debe proporcionar a la aplicación sobre su cuenta de Facebook (acceder a su nombre y al listado de sus amigos), o le confirmará que ya se los ha proporcionado en caso de ser así. Una vez el proceso se ha confirmado, se iniciará la carga de los contactos en el chat en base a los amigos de Facebook del usuario de la aplicación. Este proceso será remarcado mediante una señal clara, y dependerá tanto de la calidad de la conexión como del número de contactos a proporcionar.

Este proceso no será necesario realizarlo cada vez que se abra el sistema de mensajería, puesto que existe una sesión Facebook del usuario, que se mantendrá activa durante un periodo de tiempo considerable (siempre y cuando el usuario no cierre su sesión Facebook dentro del propio terminal).

Para la carga de contactos siempre será necesaria la conexión a Internet, al menos durante la primera carga de datos en la misma ejecución de la aplicación, puesto que se requiere del identificador del usuario para obtener sus contactos internamente dentro de la base de datos de nuestra aplicación.

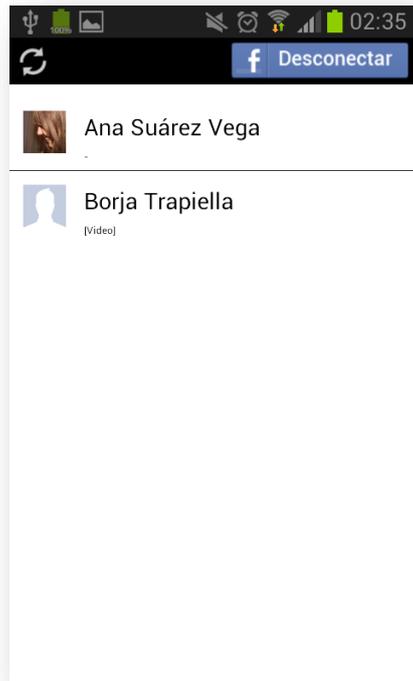


Figura 56 . Listado de contactos extraídos de Facebook

9.2.2.2 Envío de mensajes

El mecanismo de envío de mensajes funciona de la misma forma que cabría esperar en cualquier sistema de este tipo.

Tras seleccionar uno de los contactos de la pantalla inicial de la aplicación, se cargará la pantalla de la conversación con ese contacto, con todos los mensajes que hayamos intercambiado, y siempre mostrando los últimos de todo el listado.

En esta misma pantalla disponemos de un campo de texto en el que introducir nuestro mensaje, y de un botón “Enviar” para confirmar el envío al contacto receptor del mismo.



Figura 57 . Campo enviar de la pantalla de conversación

Una vez lo hayamos enviado, el mensaje será añadido como el último del listado de la conversación.

Es de especial importancia recalcar que en caso de no poder enviarse por falta de conexión, el mensaje será almacenado como pendiente, y será enviado a ese usuario la próxima vez que dispongamos de nuevo de conexión a Internet.

El último mensaje enviado, en caso de ser el último de la conversación, aparecerá en el listado de contactos, de forma reducida de bajo del nombre del contacto con el que está establecida dicha conversación.

9.2.2.3 Recepción de mensajes

Sobre la recepción de mensajes no hay que destacar ningún tipo de funcionamiento concreto o tarea que el usuario tenga que realizar.

Si el dispositivo tiene conexión a Internet, podrá recibir mensajes del resto de sus contactos, y estos aparecerán o bien en el listado de mensajes de la conversación, o bien como último mensaje recibido si estamos en la pantalla del listado de contactos o amigos del chat.

Cabe destacar que si nos encontramos en la pantalla de contactos, o en otra conversación, los mensajes del resto de usuarios llegarán y se añadirán a su correspondiente conversación, y se marcarán como no leídos de forma clara, cambiando el color de la celda de la conversación a la que pertenezcan esos mensajes pendientes de lectura.

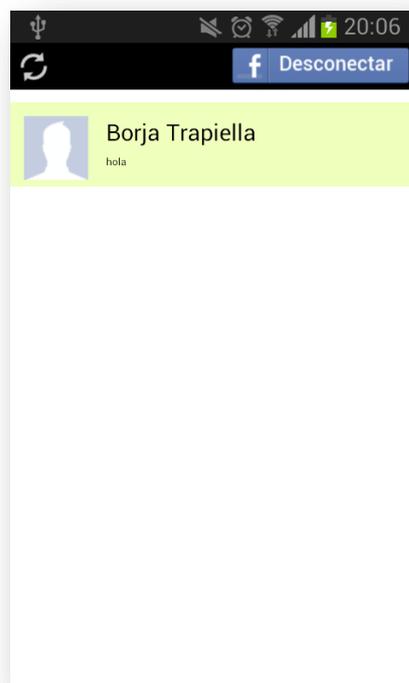


Figura 58 . Ejemplo de mensaje no leído

En el caso de no disponer de conexión, o de no tener la aplicación abierta, los mensajes se recibirán en ambos casos, ya sea cuando recuperemos la conexión o cuando volvamos a ejecutar la aplicación en nuestro terminal, y se comportarán de la misma forma que en el caso normal.

9.2.2.4 Reproducción de animaciones

La reproducción de animaciones generadas por la aplicación, ya sean enviadas por otros usuarios o por nosotros mismos, es muy sencilla y la funcionalidad está claramente señalizada.

El sistema detectará que mensajes contienen información referente a los datos de una animación, y los mostrará de forma diferente a como se tratan los mensajes de texto convencionales.

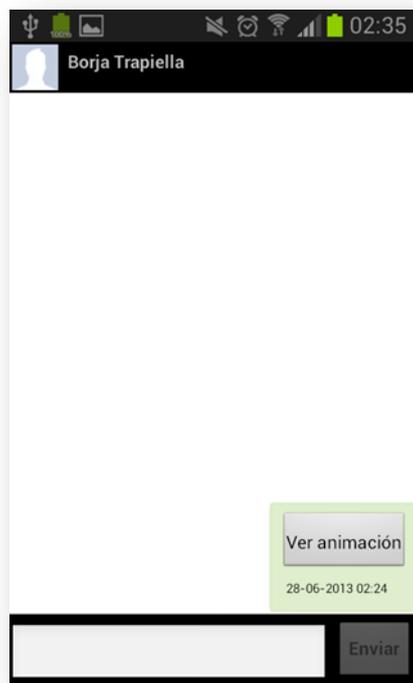


Figura 59 . Mensaje formateado como animación en la conversación del chat

Simplemente habrá que pulsar en el botón que se muestra en la captura anterior, y se lanzará la reproducción de la animación. Una vez en este punto, el uso de la aplicación sigue las mismas pautas que en la pantalla principal de animaciones: pulsando el botón "Atrás" se detendrá la reproducción de la animación y se mostrará el menú de opciones (en este caso solo permite volver a reproducirlo), y en caso de no detenerla el menú se desplegará cuando termine la reproducción de la animación.



Figura 60 . Pantalla de reproducción de animacion desde el sistema de mensajería

9.2.2.5 Actualización de contactos

El botón de la parte superior izquierda de la pantalla del listado de contactos, tiene como funcionalidad asignada refrescar el listado de contactos disponibles, volviendo a compararlos con la información que nos ofrece Facebook.

Esta tarea de actualización, respetará todos los contactos que ya teníamos (aunque ya no tengan instalada la aplicación), actualizando sus nombres de usuario o sus fotografías, y además nos añadirá aquellos nuevos amigos que se hayan instalado recientemente la aplicación.

La actualización se ejecutará automáticamente cada dos semanas, pero este botón se proporciona fundamentalmente por si alguno de esos nuevos amigos que se instalaron la aplicación posteriormente, deciden iniciar una conversación con nosotros y la aplicación nos los muestra como “Desconocido”. En ese caso, podemos utilizar el botón de actualizar, y esa conversación pasará a estar correctamente identificada, con el nombre del usuario y su fotografía, al igual que el resto de contactos ya conocidos.

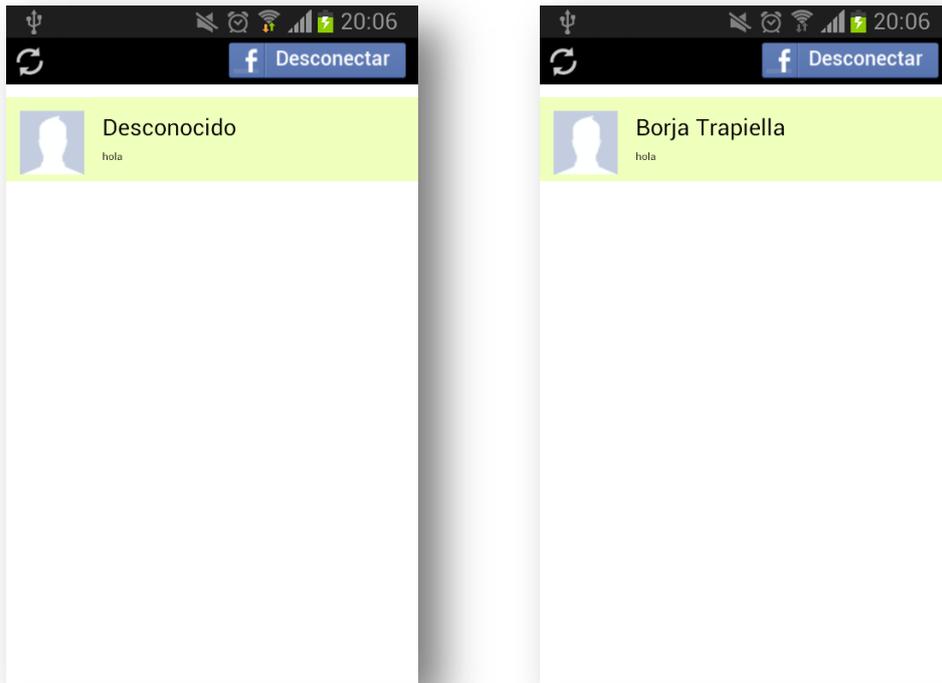


Figura 61 . Antes y después de actualizar un contacto

9.2.3 Interacción con elementos publicitarios y notificaciones

Como ya se ha comentado, la vía para monetizar esta aplicación está basada en los contenidos publicitarios, y este se distribuye al usuario dentro de la aplicación de dos formas diferentes: banner fijo, e interstitials puntuales.

En el caso del banner, se encuentra siempre localizado en la parte inferior de la pantalla principal de las animaciones. No interfiere en ningún momento para el uso normal de la aplicación, y el usuario solo verá cambios en caso de que haga una pulsación sobre el mismo.

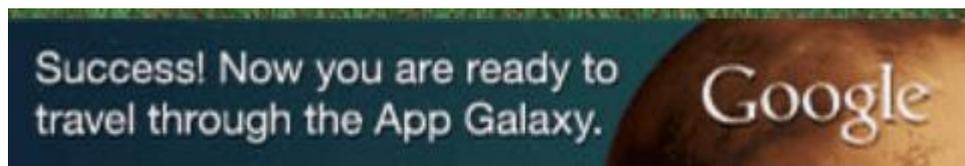


Figura 62 . Ejemplo del banner para el modo de prueba

En caso de que el usuario pulse sobre el banner porque considere interesante la publicidad que refleja, se le abrirá el navegador de su terminal (o el diálogo para seleccionar uno) con el contenido publicitado en la correspondiente web relacionada con el banner. Una vez abandone el navegador, volverá a la aplicación en el mismo punto donde lo había dejado.

El caso de los interstitials es algo más diferente. Este tipo de publicidad vendrá a través de notificaciones, que contendrán un texto orientativo acerca del contenido de esa publicidad. Si el usuario pulsa sobre la notificación, se lanzará la ejecución de la aplicación, y una vez abierta se mostrará el contenido publicitario que incluían los datos de la notificación.

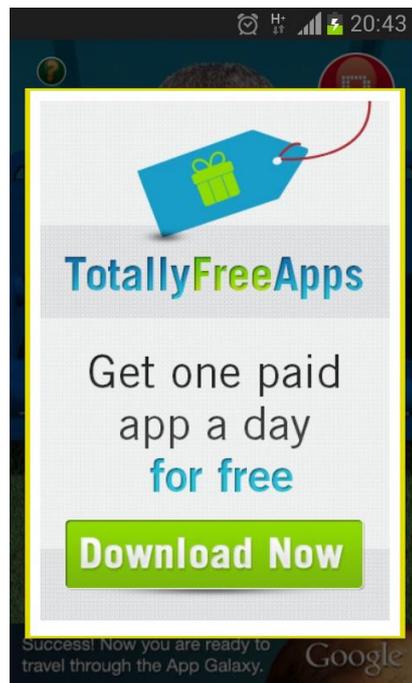


Figura 63 . Ejemplo de interstitial publicitario

Este contenido será mostrado mediante una página web incrustada dentro de la interfaz de la aplicación, y si el usuario pulsa sobre el mismo la acción resultante será igual que la del banner convencional.

Para cerrarlo simplemente tendrá que pulsar el botón “Atrás” de su terminal.

Capítulo 10. Conclusiones

Ampliaciones

y

10.1 Conclusiones

Al igual que llevo haciendo durante toda la documentación, me gustaría separar mis conclusiones por cada una de las dos funcionalidades diferentes que he desarrollado para este proyecto.

En el caso del sistema de animaciones me siento particularmente orgulloso de lo que he conseguido implementar, particularmente por todas las limitaciones y dificultades encontradas a cada paso.

Considero que el resultado final es muy correcto, y se ha logrado una solución de un alto nivel de calidad en todos los aspectos: reproducción de animaciones, compatibilidad entre dispositivos, análisis de audio, funcionalidad de generar vídeo, etc.

Como casi todo en esta profesión y en este ámbito puede ser mejorable, pero en definitiva creo que tanto el trabajo realizado como los resultados obtenidos, han sido muy satisfactorios.

Una pequeña prueba de ello es que la versión “recortada” de este proyecto, la que abarca únicamente el sistema de animaciones, se encuentra publicada en la tienda oficial de aplicaciones Android “Play Store”, y sin ningún tipo de promoción ha alcanzado en poco tiempo una cifra cercana a las 20000 descargas y una muy buena valoración por los usuarios (a pesar de contener algún error subsanado durante el desarrollo del presente proyecto).

En lo referente al sistema de mensajería, hay que tener en cuenta que los objetivos a corto plazo eran muy diferentes. Se trataba de hacer una investigación “práctica” de que plataformas o alternativas existían actualmente para dar respuesta a una serie de necesidades, descritas en los capítulos iniciales de esta documentación.

Teniendo en cuenta esto, creo que he podido realizar un muy buen acercamiento al objetivo buscado, implementando un sistema de mensajería para Android sencillo pero de un alto grado de funcionalidad, y demostrando como se puede adaptar un sistema de este tipo al contenido particular de una aplicación determinada.

En este sentido soy muy optimista de cara a las posibilidades de seguir trabajando por este camino, utilizando esta combinación de herramientas, protocolos, clientes, etc, para poder lograr sacar adelante objetivos y proyectos con mayores pretensiones y funcionalidades en caso de que sea necesario.

En definitiva, creo que el contenido y el trabajo contenidos en este proyecto han sido muy completos y variados en líneas generales, y sobre todo destacables por el uso e implementación de tecnologías, herramientas, dispositivos y funcionalidades que tienen la máxima presencia en el ámbito tecnológico e informático a día de hoy.

10.2 Ampliaciones

En el capítulo de ampliaciones, una vez más volveré a referirme por separado a cada funcionalidad implementada.

En el caso del sistema de animaciones, creo que ya es un proyecto muy trabajo y bastante cerrado de cara a implementar nuevas funcionalidades. Sin embargo, una de las más interesantes sería de la apoyar la compatibilidad entre dispositivos con una descarga dinámica del contenido en base a determinadas características del terminal, lo que nos permitiría por ejemplo lograr una compatibilidad mucho más refinada con todos los tablets y smartphones que se están lanzando en la actualidad con un tamaño de pantalla y unas resoluciones muy elevadas (y en los que ahora la calidad del contenido multimedia se ve bastante afectada).

Respecto al sistema de mensajería sí que se podrían plantear muchas más mejoras y ampliaciones, que lo acercasen más al resto de sistemas similares que hay en el mercado (en cuanto a funcionalidad), aunque si es cierto que en ese caso ya estaríamos tratando con un proyecto más individual centrado exclusivamente en ese tipo de funcionalidades. Algunas de estas mejoras serían:

- Conversaciones multiusuario
- Confirmación de envíos y llegadas
- Envío de archivos
- Personalización del interfaz
- Exportación del historial de mensajes
- Control de última conexión de contactos

Capítulo 11. Presupuesto

En este capítulo vamos a desarrollar el presupuesto del proyecto, estudiando por una parte el coste de los elementos materiales utilizados en el desarrollo (hardware y software) y por otra parte el total de trabajo (recursos humanos dedicados al proyecto) realizado durante los 70 días que ha durado todo el proceso.

11.1 Coste de recursos materiales

En este apartado analizaremos el conjunto de hardware y software empleado para el desarrollo del sistema de cara a incluir los costes correspondientes en el desglose del presupuesto.

El software empleado durante la realización del proyecto ha sido:

- Android ADT
- Enterprise Architect
- Microsoft Office 2010
- Microsoft Project 2007
- Prosody server

El hardware empleado en el desarrollo del proyecto es:

- Ordenador portátil Fujitsu Siemens Amilo
- Samsung Galaxy S
- Samsung Galaxy S2
- Samsung Galaxy S3
- Samsung Galaxy Ace
- Nexus One

Dado que el uso de estos recursos no se destina de manera exclusiva al desarrollo de este proyecto, debemos calcular la tasa de amortización a aplicar para poder determinar su valor específico dentro del marco del presente proyecto.

Para esto estimamos una vida útil de los anteriores recursos de 3 años, que sumado a los 70 días de duración reflejados en la planificación, nos permite calcular el valor para la citada amortización.

- Días laborables de los recursos: 3 años x 12 meses x 20 días laborables = 720 días
- La tasa de amortización se calcula dividiendo los 70 días de duración del proyecto entre los 720 días “útiles” de los recursos (70/720). El resultado es una tasa de amortización del 0,0972%, redondeado para simplificar el cálculo en el presupuesto a un 0,097%

Una vez realizado este cálculo, podemos detallar el presupuesto de los recursos en la siguiente tabla.

| Item | Subítem | Concepto | % de Amortización | Precio unitario | TOTAL |
|------|---------|-------------------------|-------------------|-----------------|-----------------|
| 001 | | Hardware | | | 190,61 € |
| | 001 | Ordenador Portátil | 0,097 | 700,00 € | 67,90 € |
| | 002 | Samsung Galaxy S | 0,097 | 120,00 € | 11,64 € |
| | 003 | Samsung Galaxy S2 | 0,097 | 300,00 € | 29,10 € |
| | 004 | Samsung Galaxy S3 | 0,097 | 450,00 € | 43,65 € |
| | 005 | Samsung Galaxy Ace | 0,097 | 160,00 € | 15,52 € |
| | 006 | Nexus One | 0,097 | 235,00 € | 22,80 € |
| 002 | | Software | | | 152,82 € |
| | 001 | Android ADT (gratis) | 0,097 | Gratis | 0,00 € |
| | 002 | Enterprise Architect | 0,097 | 136,55 € | 13,24 € |
| | 003 | Microsoft Office 2010 | 0,097 | 139,00 € | 13,48 € |
| | 004 | Microsoft Project 2007 | 0,097 | 1.300,00 € | 126,10 € |
| | 005 | Prosody Server(garuito) | 0,097 | Gratis | 0,00 € |
| | | | | TOTAL | 343,43 € |

11.2 Coste de recursos humanos

En este apartado analizaremos detalladamente el presupuesto relativo al coste de los recursos humanos necesarios para el desarrollo del proyecto, en este caso formados por una sola persona.

Para calcular este presupuesto, se van a tener en cuenta los siguientes elementos:

- El número de días laborables empleados en el desarrollo: 151 días
- Una media diaria estimada de 5 horas de trabajo
- Las tareas definidas en la planificación del proyecto en el capítulo 4 de esta documentación

En la tabla que se incluye a continuación se detallan los costes de los recursos humanos.

| Item | Subítem | Concepto | Días de trabajo | Precio unitario | TOTAL |
|------|---------|------------------------------------|-----------------|-----------------|-------------------|
| 001 | | Aplicación móvil | | | 1.295,00 € |
| | 001 | Sistema de animaciones | 6 | 35 € | 210,00 € |
| | 002 | Grabación y gestión de audio/video | 11 | 35 € | 385,00 € |
| | 003 | Plataformas publicitarias | 5 | 35 € | 175,00 € |
| | 004 | Integración | 4 | 35 € | 140,00 € |
| | 005 | Pruebas | 1 | 35 € | 35,00 € |
| | 006 | Sistema de chat | 10 | 35 € | 350,00 € |
| 002 | | Integración | | | 175,00 € |
| | 001 | Integración completa | 5 | 35 € | 175,00 € |
| 003 | | Pruebas | | | 70,00 € |
| | 001 | Pruebas generales | 2 | 35 € | 70,00 € |
| 004 | | Documentación | | | 455,00 € |
| | 001 | Elaboración de la documentación | 13 | 35 € | 455,00 € |
| | | | | TOTAL | 1.995,00 € |

11.3 Presupuesto completo

| Item | Subítem | Concepto | Días de trabajo | Precio unitario | TOTAL |
|------|---------|------------------------------------|-----------------|-----------------|-------------------|
| 001 | | Aplicación móvil | | | 1.295,00 € |
| | 001 | Sistema de animaciones | 6 | 35 € | 210,00 € |
| | 002 | Grabación y gestión de audio/video | 11 | 35 € | 385,00 € |
| | 003 | Plataformas publicitarias | 5 | 35 € | 175,00 € |
| | 004 | Integración | 4 | 35 € | 140,00 € |
| | 005 | Pruebas | 1 | 35 € | 35,00 € |
| | 006 | Sistema de chat | 10 | 35 € | 350,00 € |
| 002 | | Integración | | | 175,00 € |
| | 001 | Integración completa | 5 | 35 € | 175,00 € |
| 003 | | Pruebas | | | 70,00 € |
| | 001 | Pruebas generales | 2 | 35 € | 70,00 € |
| 004 | | Documentación | | | 455,00 € |
| | 001 | Elaboración de la documentación | 13 | 35 € | 455,00 € |

| Item | Subítem | Concepto | % de Amortización | Precio unitario | TOTAL |
|------|---------|-------------------------|-------------------|-----------------|-----------------|
| 001 | | Hardware | | | 190,61 € |
| | 001 | Ordenador Portátil | 0,097 | 700,00 € | 67,90 € |
| | 002 | Samsung Galaxy S | 0,097 | 120,00 € | 11,64 € |
| | 003 | Samsung Galaxy S2 | 0,097 | 300,00 € | 29,10 € |
| | 004 | Samsung Galaxy S3 | 0,097 | 450,00 € | 43,65 € |
| | 005 | Samsung Galaxy Ace | 0,097 | 160,00 € | 15,52 € |
| | 006 | Nexus One | 0,097 | 235,00 € | 22,80 € |
| 002 | | Software | | | 152,82 € |
| | 001 | Android ADT (gratis) | 0,097 | Gratis | 0,00 € |
| | 002 | Enterprise Architect | 0,097 | 136,55 € | 13,24 € |
| | 003 | Microsoft Office 2010 | 0,097 | 139,00 € | 13,48 € |
| | 004 | Microsoft Project 2007 | 0,097 | 1.300,00 € | 126,10 € |
| | 005 | Prosody Server(garuito) | 0,097 | Gratis | 0,00 € |

SUBTOTAL 2.338,43 €

Capítulo 12. Referencias Bibliográficas

12.1 Libros y Artículos

Libros y artículos usados de alguna forma durante el desarrollo del proyecto o su documentación.

[Korth07] Silberschatz, Abraham; F.Korth, Henry; Sudarshan, S. "Fundamentos de Diseño de Bases de Datos". Editorial MacGraw Gill. 2007. ISBN: 978-84-481-5671-8

12.2 Referencias en Internet

Páginas Web consultadas para cualquier aspecto relacionado con el desarrollo del sistema o su documentación.

[Hassan08] Hassan Montero, Y. "Guía de Evaluación Heurística de Sitios Web". <http://www.nosolousabilidad.com/articulos/heuristica.htm>

[MongoDoc] Web oficial del proyecto MongoDB. <http://www.mongodb.org/>

[SamDev] Web para desarrolladores relacionada con tecnologías de Samsung (o utilizadas por Samsung). <http://developer.samsung.com/>

[AndDev] Web para desarrolladores oficial de Android. <http://developer.android.com>

[Sonic] Web del proyecto Libsonic. <http://dev.vinux-project.org/sonic/>

[Javacv] Web del proyecto javacv. <http://code.google.com/p/javacv/>

[Xmpp] Web oficial del estándar XMPP y la fundación que regula el proyecto. <http://xmpp.org/>

[Smack] Web del cliente XMPP Smack. <http://www.igniterealtime.org/projects/smack/>

[aSmack] Web del proyecto que realiza una adaptación del cliente Smack al entorno de Android. <https://github.com/Flowdalic/asmack>

[Prosody] Web del servidor para utilizar el protocolo XMPP. <http://prosody.im/>

Capítulo 13. Apéndices

13.1 Glosario y Diccionario de Datos

Por orden alfabético, todos los términos que se consideren importantes en la aplicación con una descripción breve de su significado dentro de la aplicación.

- **AdMob:** Plataforma de publicidad proporcionada por Google
- **Android:** sistema operativo para dispositivos móviles sobre el que hemos desarrollado la aplicación del presente proyecto.
- **Animación de frames:** animación elaborada dentro de la aplicación en base a un conjunto de imágenes individuales.
- **Animación de video:** animación reproducida dentro de la aplicación en base a un archivo de vídeo.
- **Banner:** segmento rectangular (por lo menos es habitual que lo sea en los smarthpones) que se coloca habitualmente fijo dentro de la interfaz de una aplicación móvil para mostrar contenido publicitario. Se suele monetizar por número de impresiones (número de veces que se muestra un banner en la aplicación) y por número de pulsaciones (número de veces que un usuario pulsa sobre el anuncio).
- **Imagine800:** nombre de la empresa para la que ha sido desarrollado el presente proyecto.
- **Interstitial:** Similar a un banner, pero de mucho mayor tamaño. Puede ocupar toda la pantalla o gran parte de la misma. En este caso la monetización dependería del cliente con el que se contratase el servicio publicitario, ya que en este proyecto lo controlamos internamente en la empresa y no a través de terceros como con AdMob.
- **Modum:** plataforma empleada internamente en la empresa para el desarrollo de todos los proyectos. Incluye numerosas herramientas y mecanismos de control de usuarios, dispositivos, aplicaciones, notificaciones, y de todos los aspectos relacionados con cada proyecto en general.
- **Notificación:** alerta o mensaje que se envía a los smartphones, generados específicamente por cada aplicación. Aunque los puede mandar la empresa responsable de la aplicación, siempre pasan a través de las plataformas de Google. Se suelen utilizar para llamar la atención del usuario sobre la aplicación, notificar promociones, o como en nuestro caso distribuir contenido publicitario propio.
- **Pitch:** entendemos por pitch un valor numérico que determina el tono de un sonido. En el contexto de la aplicación lo utilizaremos para editar el audio grabado por el usuario, en dos medidas: pitch agudo o pitch grave.
- **Smartphone:** teléfono móvil de última generación, que se caracteriza por capacidades como entrar en internet, reproducir música, vídeos, y fundamentalmente tener acceso a un amplio catálogo de aplicaciones instalables. Es la base “hardware” sobre la que corre el proyecto.
- **Splash:** habitualmente se conoce por este término a la pantalla inicial que se muestra brevemente en el inicio de una aplicación a modo introductorio.

13.2 Contenido Entregado en el CD-ROM

13.2.1 Contenidos

Estructura

| Directorio | Contenido |
|--|--|
| <i>./ Directorio raíz del CD</i> | Contiene un fichero leeme.txt explicando todo el contenido del CD, además de la carpeta principal “Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario” |
| <i>./ Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario</i> | Contiene 3 carpetas: <ul style="list-style-type: none"> • Documentación • Presentación • Ejecutable |
| <i>./ Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario / Documentación</i> | Contiene toda la documentación asociada al proyecto. Incluye tanto el archivo Word docx, y el archivo PDF. A su vez contiene otras 2 carpetas: <ul style="list-style-type: none"> • Imágenes • Diagramas |
| <i>./ Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario / Documentación/ Imágenes</i> | Contiene todas las imágenes utilizadas en la elaboración de la documentación del proyecto |
| <i>./ Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario / Documentación/ Diagramas</i> | Incluye todas las imágenes de los diagramas utilizadas en la documentación. Además de esto, contiene dos ficheros más. <p>Por un lado el fichero generado por la herramienta Enterprise Architect, con la que se han realizado los diagramas UML.</p> <p>Por otro, un fichero xml de la herramienta DBDesigner, con la que se ha realizado el diagrama de Entidad-Relación</p> |
| <i>./ Aplicación de ocio para dispositivos móviles, centrada en el manejo de</i> | Contiene la presentación en <i>Powerpoint</i> que va a ser utilizada el día de la defensa del |

| Directorio | Contenido |
|--|--|
| <i>animaciones y contenido multimedia, con distribución a través de chat multiusuario / Presentación</i> | proyecto, además de otros elementos utilizados durante la misma |
| <i>./ Aplicación de ocio para dispositivos móviles, centrada en el manejo de animaciones y contenido multimedia, con distribución a través de chat multiusuario / Ejecutable</i> | Contiene el archivo .apk que sirve de instalador de la aplicación desarrollado en los dispositivos Android compatibles |

