
DESARROLLO DE UNA APLICACIÓN MÓVIL PARA COMPRA-VENTA DE INSTRUMENTOS MUSICALES

Jorge Onecha Blanco
Máster Universitario en Ingeniería Informática
Desarrollo de Aplicaciones Sobre Dispositivos Móviles

Jordi Ceballos Villach
Jordi Almirall López
Robert Clarisó Viladrosa

10/06/2020



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

B) GNU Free Documentation License (GNU FDL)

Copyright © 2020 Jorge Onecha Blanco.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (Jorge Onecha Blanco)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.



FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Desarrollo de una aplicación móvil para compra-venta de instrumentos musicales (Lost Instrument)</i>
Nombre del autor:	Jorge Onecha Blanco
Nombre del consultor/a:	Jordi Ceballos Villach Jordi Almirall López
Nombre del PRA:	Robert Clarisó Viladrosa
Fecha de entrega:	06/2020
Titulación:	Máster Universitario en Ingeniería Informática
Área del Trabajo Final:	Desarrollo de aplicaciones sobre dispositivos móviles
Idioma del trabajo:	Castellano
Palabras clave	Compra-Venta Instrumento Android
Resumen del Trabajo	
<p>El presente Trabajo de Fin de Master (TFM) se centra en realizar el análisis, diseño e implementación de una aplicación móvil dedicada a la compra-venta de productos e instrumentos musicales. La aplicación deberá facilitar la interacción entre los usuarios, permitiendo la posibilidad de cargar productos en la misma, para poder compartirlos y que sean visibles y accesibles para los demás. La comunicación entre dos usuarios se llevará a cabo mediante un chat privado.</p> <p>Para la implementación, se ha decidido que el sistema operativo elegido sea Android, debido al gran número de dispositivos que lo utilizan, a su integración con herramientas de google como son Firebase y Google Maps, y a que es un sistema operativo de código abierto. Además, se ha decidido usar metodología ágil debido al alto grado de mejoras existentes, que favorecen el desarrollo del proyecto con continuas mejoras.</p> <p>En el último punto de la memoria se incluye un anexo, con información de la instalación/compilación del proyecto y un manual de usuario.</p>	
Abstract:	
<p>This Master's Thesis focuses on analyzing, designing and implementing a mobile application dedicated to the sale and purchase of musical products and instruments. The application should make the interaction between users easier, allowing the possibility of uploading products into it, in order to share them and make them visible and accessible to others. Communication between two users will be carried out through a private chat.</p> <p>For the implementation, it has been decided that the chosen operating system is Android, due to the large number of devices that use it, its integration with Google tools such as Firebase and Google Maps, and the fact that it is an open source operating system. In addition, It has decided to use agile methodology due to the high degree of existing improvements, which enhance the development of the project with continuous improvements.</p> <p>In the last point of the memory an annex is included, with information of the installation/compilation of the project and a user manual.</p>	



Contenido

1	Contenido	III
2	Tablas	IV
3	Figuras	V
1	Capítulo 1 – Introducción.....	1
1.1	Estructura de la memoria.....	1
1.2	Estado del Arte	2
2	Capítulo 2 – Plan de Trabajo.....	10
2.1	Contexto y justificación del trabajo	10
2.2	Objetivos del Trabajo	10
2.3	Enfoque y método seguido.....	12
2.4	Planificación del Trabajo	13
3	Capítulo 3 – Caso de Estudio [9]	15
3.1	Análisis.....	15
3.2	Diseño Conceptual	21
3.3	Prototipado	24
3.4	Evaluación.....	30
3.5	Definición de los casos de uso	32
3.6	Diseño de la arquitectura.....	38
4	Capítulo 4 - Implementación.....	45
4.1	Desarrollo	45
4.2	Pruebas	54
5	Capítulo 5 – Conclusiones.....	59
5.1	Conclusiones y líneas futuras de trabajo	59
5.2	Logros alcanzados y conocimientos adquiridos.....	60
5.3	Posibles trabajos futuros.....	60
6	Capítulo 6 – Referencias	62
6.1	Referencias	62
7	Capítulo 7 – Anexos.....	64
7.1	Instrucciones de instalación y compilación.....	64
7.2	Manual de usuario	66



Tablas

Tabla 1. Características de Android	6
Tabla 2. Versiones de Android.....	9
Tabla 3. Cuota de mercado Versiones Android	9
Tabla 4. Estimación de tareas.....	14
Tabla 5. Estimación de horas dedicadas al proyecto.....	14
Tabla 6. Estimación de costes del proyecto	14
Tabla 7. Resultados fase de indagación	21
Tabla 8. Casos de uso. Escenario 1	22
Tabla 9. . Casos de uso. Escenario 2	22
Tabla 10. . Casos de uso. Escenario 3	22
Tabla 11. CU-001- Registrar Usuario	33
Tabla 12. CU-002- Login en la App	33
Tabla 13. CU-003- Acceso al menú de la aplicación.....	33
Tabla 14. CU-004- Buscar Producto	34
Tabla 15. CU-005- Acceder a la ficha del producto	34
Tabla 16. CU-006- Acceder a mis productos	35
Tabla 17. CU-007- Subir producto.....	35
Tabla 18. CU-008- Visualizar Chats	36
Tabla 19. CU-009- Visualizar mapa de productos	36
Tabla 20. CU-0010- Editar mi perfil.....	37
Tabla 21. CU-0011- Actualizar contraseña.....	37
Tabla 22. CU-0012- Eliminar usuario de la aplicación.....	38
Tabla 23. CU-0013- Cerrar sesión	38
Tabla 24. Relaciones entre las entidades.....	43
Tabla 25. Tabla User	43
Tabla 26. Tabla Products.....	43
Tabla 27. Tabla Conversación.....	43
Tabla 28. Tabla Chat	43
Tabla 29. Tabla Token	43
Tabla 30. Modelo Vista Controlador.....	44
Tabla 31. Recursos técnicos utilizados para el desarrollo.....	45
Tabla 32. Definición de los casos de prueba	57
Tabla 33. Resultado de los casos de prueba	58



Figuras

Ilustración 1. Arquitectura de Android	8
Ilustración 2. Diagrama de Gantt	14
Ilustración 3. Flujos de interacción	23
Ilustración 4. Sketches.....	24
Ilustración 5. Prototipo: Pantalla de inicio	25
Ilustración 6. . Prototipo: Pantalla de registro de usuario	25
Ilustración 7. Prototipo: Menú principal	26
Ilustración 8. Prototipo: Búsqueda de productos	26
Ilustración 9. Prototipo: Pantalla Mis Productos.....	27
Ilustración 10. Prototipo: Pantalla ficha producto.....	27
Ilustración 11. Prototipo: Pantalla subir producto.....	28
Ilustración 12. Prototipo: Pantalla Listado de Conversaciones.....	28
Ilustración 13. Prototipo: Pantalla chat	28
Ilustración 14. Prototipo: Pantalla de mapa de productos	29
Ilustración 15. Prototipo: Pantalla Mi perfil.....	29
Ilustración 16.. Prototipo: Botón Cerrar sesión.....	29
Ilustración 17. Prototipo: Diagrama general de Casos de Uso.....	32
Ilustración 18. Cloud Firestore.....	41
Ilustración 19. Diagrama Entidad-Relación	42
Ilustración 20. Firebase Authentication	46
Ilustración 21. Ejemplo Authentication: Cuentas registradas en la app.....	47
Ilustración 22. Firebase Storage	48
Ilustración 23. Firebase Cloud Messaging	48
Ilustración 24. Código fuente carpeta SRC.....	49
Ilustración 25. Carpeta de recursos RES	52
Ilustración 26. Dispositivo Virtual utilizado	54
Ilustración 27. Características dispositivo virtual utilizado	54
Ilustración 28. Android Studio	64
Ilustración 29. Vista principal IDE Android Studio.....	65
Ilustración 30. Versiones disponibles SDK Android Studio.....	65
Ilustración 31. Splash Screen	66
Ilustración 32. Pantalla index	66
Ilustración 33. Pantalla Login	67
Ilustración 34. Pantalla Registro Usuario.....	68
Ilustración 35. Acceso a la aplicación	68
Ilustración 36. Acceso con Facebook.....	69
Ilustración 37. Guardar Credenciales	69
Ilustración 38. Pantalla Menú Principal	70
Ilustración 39. Seleccionar y acceder a un producto del usuario autenticado.....	71
Ilustración 40. Eliminación producto subido	71
Ilustración 41. Crear nuevo producto.....	72
Ilustración 42. Subir imagen a la aplicación. Parte I	73
Ilustración 43. Subir imagen a la aplicación. Parte II	73
Ilustración 44. Visualizar producto subido	74
Ilustración 45. Acceso a Productos cerca.....	75
Ilustración 46. Búsqueda de un producto	75
Ilustración 47. Ficha de un producto de un usuario	76



Ilustración 48. Marcar como favorito un producto	77
Ilustración 49. Visualizar listado de productos favoritos	77
Ilustración 50. Abrir un Chat desde la ficha de un producto.....	78
Ilustración 51. Acceder a conversación iniciada	79
Ilustración 52. Mapa de productos	79
Ilustración 53. Modificar datos	80
Ilustración 54. Actualizar contraseña. Parte I	81
Ilustración 55. Actualizar contraseña. Parte II	81
Ilustración 56. Eliminar usuario de la App	82



Capítulo 1 – Introducción

1.1 Estructura de la memoria

La memoria está compuesta por siete capítulos, más un breve resumen inicial situado al principio de dicha memoria, redactado en castellano y en inglés. A continuación, detallaremos el contenido principal de cada apartado:

Resumen: Resumen del contenido del Trabajo de Fin de Máster

Capítulo 1 – Introducción: El objetivo de esta sección es realizar una introducción de los puntos principales en los que se basará el Trabajo de Fin de Máster.

- **Estructura de la memoria:** Se pretende facilitar al lector el orden en el que se tratarán los puntos clave de la memoria.
- **Estado del arte:** El objetivo de este punto es realizar un análisis de los sistemas operativos móviles que existen en la actualidad, analizando pros y contras y justificando los motivos de la elección del sistema operativo para dicho trabajo.

Capítulo 2 – Plan de Trabajo: Se divide en los siguientes puntos:

- **Contexto y justificación del trabajo:** Es el punto de partida del trabajo. En él, se definirá las necesidades a cubrir y los resultados que se desean obtener. También se justificarán las decisiones tomadas para su desarrollo.
- **Objetivos del proyecto:** Listado de los objetivos del trabajo, indicando a alto nivel los requerimientos funcionales y no funcionales de la aplicación móvil, así como la plataforma y los dispositivos a los que va dirigido.
- **Enfoque y método seguido:** En este apartado se indicarán cuales son las estrategias para llevar a cabo el trabajo. Se realizarán una valoración justificando los motivos de la elección.
- **Planificación del trabajo:** Descripción de los recursos necesarios, las tareas a realizar y la planificación temporal que se llevarán a cabo para realización del proyecto.

Capítulo 3 – Caso de estudio: Durante este capítulo se aplicará la metodología de Diseño Centrado en el Usuario (DCU). Se divide en los siguientes apartados:

- **Usuario y contexto de uso:** En este punto nos centraremos en conocer las características de los usuarios, sus necesidades y objetivos, así como el contexto de uso. El objetivo es detectar las funcionalidades de la aplicación.
- **Diseño conceptual:** A partir de la fase anterior, se elaborarán los escenarios de uso para conocer desde el punto de vista del usuario como utilizará el producto en un contexto concreto. El objetivo es detectar las necesidades de los usuarios y de diseño.
- **Prototipado:** Tomando como referencia los flujos de interacción descritos, se realizará un prototipo horizontal de alta fidelidad. El objetivo es representar la aplicación para poder realizar una evaluación antes de desarrollar el producto final.
- **Evaluación:** En este punto se planificará la evaluación del prototipo. Se recopilarán preguntas para obtener información referente al usuario que realiza el test, las tareas que deberán realizar, así como las preguntas referentes a las tareas.
- **Definición de los casos de uso:** El objetivo de este apartado es el de definir los casos de uso de la aplicación tomando como referencia lo aprendido al realizar el Diseño



Centrado en el Usuario. El objetivo será el de establecer las funcionalidades de la aplicación.

- **Diseño de la arquitectura:** En este punto se definirá la arquitectura del sistema, identificando las entidades que lo componen.

Capítulo 4 – Implementación: El objetivo de este apartado es centrarse en el producto que se tiene que desarrollar, documentando las decisiones tomadas durante el desarrollo y las pruebas realizadas.

- **Desarrollo:** En esta etapa se construirá una aplicación móvil que satisfaga los objetivos planteados dentro del proyecto. Se añadirá una breve descripción de los elementos utilizados en el desarrollo de la aplicación justificando las decisiones tomadas.
- **Pruebas:** Se definirá el entorno de pruebas utilizado detallando los resultados de las diferentes pruebas ejecutadas. El objetivo de este apartado es el de comprobar el correcto funcionamiento de la aplicación.

Capítulo 5 – Conclusiones: Apartado que contendrá las conclusiones obtenidas tras la realización del Trabajo de Fin de Master.

- **Resultado final:** En este punto se definirán las impresiones y sensaciones personales sobre la finalización y consecución del trabajo.
- **Logros alcanzados y conocimientos adquiridos:** Se definirán los logros alcanzados tras la finalización del proyecto, así como el conocimiento adquirido durante el desarrollo del mismo.
- **Posibles trabajos futuros:** Líneas de trabajo futuras para mejorar y evolucionar la aplicación

Capítulo 6 – Referencias: Bibliografía del material usado para la realización del proyecto.

Capítulo 7 – Anexos: Información complementaria que no tiene cabida en ninguno de los puntos anteriores: instalación/compilación del proyecto y manual de usuario.

1.2 Estado del Arte

En este apartado, el objetivo principal será el de describir y comparar los sistemas operativos móviles que en la actualidad tienen una mayor repercusión entre los usuarios. En el segundo apartado, nos centraremos en describir y analizar el sistema operativo **Android**.

1.2.1 Principales sistemas operativos móviles

A continuación, describiremos brevemente los principales sistemas operativos analizando sus ventajas y sus inconvenientes frente a los demás sistemas.

1.2.1.1 Android [1]

Android es un sistema operativo móvil desarrollado por Google, basado en Kernel de Linux y otros softwares de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes, automóviles (Android Auto) y televisores (Android TV). Inicialmente fue desarrollado por Android, empresa que Google respaldó económicamente y que adquirió en 2005. Android fue presentado en 2007



junto con la fundación del Open Handset Alliance (un consorcio de compañías) para avanzar en los estándares abiertos de los dispositivos móviles. Al ser una plataforma de código abierto, la convierte en el sistema operativo móvil más utilizado en el mundo, con una cuota de mercado superior al 80%. Es utilizado por numerosos dispositivos de multitud de marcas.

Puntos fuertes de Android

- El código de Android es abierto. Google liberó Android bajo licencia Apache. Esto es una gran ventaja ya que permite que cualquier persona pueda realizar una aplicación para Android de forma libre.
- Hoy día hay más de 650.000 aplicaciones disponibles para teléfonos Android de las cuales aproximadamente 2/3 son gratis. Además, la libertad de código permite adaptar Android a multitud de dispositivos no solo teléfonos móviles. Está implantado en tablets, GPS, relojes, etc.
- El sistema Android es capaz de hacer funcionar a la vez varias aplicaciones y se encarga de gestionarlas, dejarlas en modo suspensión si no se utilizan e incluso cerrarlas si llevan un periodo determinado de inactividad. De esta manera se evita un consumo excesivo de batería.

Inconvenientes

- Al disponer de tanto mercado de terminales, cada uno de estos podrá tener una versión y unas características diferentes a las cuales se tiene que adaptar Android. Esta gran fragmentación, afecta de manera perjudicial a la optimización de los dispositivos y esto se traduce en pérdida de velocidad, problemas de seguridad e incompatibilidades [2].
- Al ser un sistema abierto le hace más inseguro y más propenso a los ataques aprovechando sus vulnerabilidades. Cualquier persona puede crear una aplicación y subirla al *Market* sin unos exámenes exhaustivos de seguridad. Mayor facilidad para subir una aplicación maliciosa.
- Incompatibilidad entre versiones. Este inconveniente va unido al de la fragmentación, ya que, debido al gran número de versiones de sistemas operativos, puede existir incompatibilidad entre versiones en algunas aplicaciones.

1.2.1.2 IOS [3]

Sistema operativo móvil de la multinacional Apple. iOS se deriva de macOS, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo tipo Unix. Originalmente fue desarrollado para el *iPhone* (iPhone OS). Después, se ha usado en dispositivos como el *iPod touch* y el *iPad*. No permite la instalación de iOS en hardware de terceros.

Una de las características principales es su aspecto tan personal que ha marcado tendencia en el mundo de la tecnología a lo largo de los últimos 13 años. Otro de los puntos claves es el aprovechamiento de los recursos al ser la propia Apple la creadora tanto del Software como del Hardware. A continuación, analizamos los puntos fuertes y los inconvenientes de este sistema operativo. [4]

Puntos fuertes de iOS

- Gran diseño, funcionalidad y gran experiencia de usuario debido a su facilidad de uso.
- Al ser un sistema cerrado, es bastante seguro y no todo el mundo puede subir una aplicación a su mercado. Las aplicaciones pasan unos exámenes exhaustivos de seguridad y de funcionalidad antes de publicarse, por lo tanto, son muy seguras.



- Integración y sincronización sencilla y rápida entre otros dispositivos que lleven iOS utilizando el sistema *airdrop* (traspaso de archivos, reproducción de música con *homePod*, transmisión de contenido a la TV. mediante *Android TV*).
- Perfecta integración con su Hardware, consiguiendo iguales o mejores resultados que los dispositivos de la competencia con muchos menos recursos (memoria RAM, velocidad del procesador, etc.).

Desventajas de iOS

- No es posible personalizar ni modificar muchos de los parámetros, debido a que no es posible el acceso al sistema.
- Solo utilizan este sistema operativo los dispositivos de la marca Apple y estos dispositivos poseen unos precios muy elevados.
- Sistema operativo cerrado. Desde el punto de vista del desarrollo es una contra debido a que no están sencillo realizar apps para iOS comparándolo con Android.

1.2.2 Sistema Operativo Android

El sistema operativo elegido para desarrollar la aplicación del Trabajo de Fin de Master ha sido Android.

Android es un sistema operativo, basado en el kernel de Linux y diseñado principalmente para dispositivos móviles con pantalla táctil (Smartphone y Tablets). Uno de los aspectos fundamentales del sistema operativo de Android fue su orientación a la multiplataforma, algo realmente novedoso en su momento. Esta y otras características innovadoras, consiguieron que Android se convirtiera en el sistema operativo más utilizado.

1.2.2.1 Características de Android [5]

Diseño de dispositivo	La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.
Almacenamiento	SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos.
Conectividad	Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+, NFC y WiMAX, GPRS, UMTS y HSDPA+.
Mensajería	SMS y MMS son formas de mensajería, incluyendo mensajería de texto, además del servicio de Firebase Cloud Messaging (FCM) siendo la nueva versión de Google Cloud Messaging (GCM) bajo la marca Firebase con los nuevos SDK para realizar el desarrollo de mensajería en la nube mucho más sencillo.



Navegador web	El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome.
Soporte de Java	Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y se ejecuta en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. A partir de la versión 5.0, se utiliza el Android Runtime (ART). El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner.
Soporte multimedia	Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.
Soporte para <i>streaming</i>	Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player.
Soporte para hardware adicional	Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
Entorno de desarrollo	Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. Inicialmente el entorno de desarrollo integrado (IDE) utilizado era Eclipse con el plugin de Herramientas de Desarrollo de Android (ADT). Ahora se considera como entorno oficial Android Studio, descargable desde la página oficial de desarrolladores de Android.
Google Play	Google Play es un catálogo de aplicaciones gratuitas y de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.
Multi-táctil	Android tiene soporte nativo para pantallas capacitivas con soporte multitáctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel (posiblemente para evitar infringir patentes de otras compañías). ⁴³ Más tarde, Google publicó una actualización para el



	Nexus One y el Motorola Droid que activa el soporte multitáctil de forma nativa
Bluetooth	El soporte para A2DF y AVRCP fue agregado en la versión 1.5; el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0; y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2.
Videollamada	Android soporta videollamada a través de Hangouts (antiguo Google Talk) desde su versión HoneyComb.
Multitarea	Multitarea real de aplicaciones. Las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj.
Características basadas en voz	La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.
Tethering	Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o inferiores mediante aplicaciones disponibles en Google Play (por ejemplo, PdaNet). Para permitir a un PC usar la conexión de datos del móvil Android se podría requerir la instalación de software adicional.

Tabla 1. Características de Android

1.2.2.2 Arquitectura de Android [6]

La arquitectura de Android se basa en la pila de Software. Este tipo de arquitectura se caracteriza por estar formada por diferentes capas, donde cada una de ellas utiliza elementos de la capa inferior para poder realizar sus funciones. A continuación, entraremos en detalle de cada uno de las capas que forman la arquitectura:

Aplicaciones del sistema: Se incluye un conjunto de apps centrales (correo electrónico, mensajería SMS, calendarios, navegación en Internet, contactos, etc.). Estas Apps incluidas en la plataforma vienen por defecto y brindan a los usuarios capacidades esenciales para el manejo del dispositivo.

Marco de trabajo de la API de Java: Todo el conjunto de funciones del SO Android está disponible mediante API escritas en el lenguaje Java. Estas API son los cimientos que se necesitan para crear apps de Android simplificando la reutilización de componentes del sistema y servicios centrales y modulares, como los siguientes:

- **Un sistema de vista enriquecido y extensible** que se puede usar para compilar la IU de una app. Se incluyen listas, cuadrículas, cuadros de texto, botones e incluso un navegador web integrable.



- **Un administrador de recursos** que brinda acceso a recursos sin código, como strings, gráficos y archivos de diseño.
- **Un administrador de notificaciones** que permite que todas las apps muestren alertas personalizadas en la barra de estado.
- **Un administrador de actividad** que se encarga de administrar el ciclo de vida de las apps y proporciona una pila de retroceso de navegación común.
- **Proveedores de contenido** que permiten que las apps accedan a datos desde otras apps (app de Contactos).

Los desarrolladores tienen acceso total a las mismas API del marco de trabajo que usan las apps del sistema Android.

Bibliotecas C/C++ nativas: Muchos componentes y servicios centrales del sistema Android, como el ART y la HAL, se basan en código nativo que requiere bibliotecas nativas escritas en C y C++. La plataforma Android proporciona la API del *framework* de Java para exponer la funcionalidad de algunas de estas bibliotecas nativas a las apps. Por ejemplo se puede acceder a OpenGL ES a través de la API Open GL del *framework* de Java.

Runtime de Android: Para los dispositivos con Android 5.0 (nivel de API 21) o versiones posteriores, cada app ejecuta sus propios procesos con sus propias instancias del tiempo de ejecución de Android (ART). El ART está escrito para ejecutar varias máquinas virtuales en dispositivos de memoria baja ejecutando archivos DEX, un formato de código de bytes diseñado especialmente para Android y optimizado para ocupar un espacio de memoria mínimo. Crea cadenas de herramientas, como Jack, y compila fuentes de Java en código de bytes DEX que se pueden ejecutar en la plataforma Android.

Estas son algunas de las funciones principales del ART:

- **Compilación** ahead-of-time (AOT) y just-in-time (JIT)
- **Recolección** optimizada de elementos no utilizados (GC)
- **Conversión** de **archivos** de formato ejecutable (**DEX**) de un paquete de aplicaciones a un código de máquina más compacto en Android 9 (nivel de API 28) y versiones posteriores,
- **Mejora la compatibilidad** con la **depuración**, utilizando el generador de perfiles de muestras dedicado, las excepciones de diagnóstico detalladas y los informes de fallos, y la capacidad de establecer puntos de control para supervisar campos específicos

Antes de Android 5.0 (nivel de API 21), Dalvik era el entorno de ejecución del sistema operativo. En Android, también se incluye un conjunto de bibliotecas de la máquina virtual que proporcionan la mayor parte de la funcionalidad del lenguaje de programación Java.

Capa de abstracción de hardware (HAL): La capa de abstracción de hardware (HAL) brinda interfaces estándares que exponen las capacidades de hardware del dispositivo al *framework* de la API de Java de nivel más alto. La HAL consiste en varios módulos de biblioteca y cada uno de estos implementa una interfaz para un tipo específico de componente de hardware, como el módulo de la cámara o de Bluetooth. Cuando el marco de trabajo de una API realiza una llamada para acceder al hardware del dispositivo, el sistema Android carga el módulo de biblioteca para el componente de hardware en cuestión.



Kernel de Linux: La base de la plataforma Android es el kernel de Linux. Por ejemplo, el tiempo de ejecución de Android (ART) se basa en el kernel de Linux para funcionalidades subyacentes, como la generación de subprocesos y la administración de memoria de bajo nivel. El uso del kernel de Linux permite que Android aproveche funciones de seguridad claves y, al mismo tiempo, permite a los fabricantes de dispositivos desarrollar controladores de hardware para un kernel conocido.

A continuación, se muestra una ilustración obtenida en la página de developer de Android, donde se visualiza de forma gráfica cada capa que forma la estructura de la arquitectura de un sistema Android:



Ilustración 1. Arquitectura de Android

1.2.2.3 Versiones de Android [7]

A continuación, se mostrará una tabla con todas las versiones que han existido hasta la actualidad de Android.



Nombre	Número versión	Fecha de lanzamiento	Nivel de API
Apple Pie	1.0	23 de septiembre de 2008	1
Banana Bread	1.1	9 de febrero de 2009	2
Cupcake	1.5	25 de abril de 2009	3
Donut	1.6	15 de septiembre de 2009	4
Eclair	2.0 – 2.1	26 de octubre de 2009	5 – 7
Froyo	2.2 – 2.2.3	20 de mayo de 2010	8
Gingerbread	2.3 – 2.3.7	6 de diciembre de 2010	9 – 10
Honeycomb	3.0 – 3.2.6	22 de febrero de 2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.5	18 de octubre de 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	9 de julio de 2012	16 – 18
KitKat	4.4 – 4.4.4	31 de octubre de 2013	19 – 20
Lollipop	5.0 – 5.1.1	12 de noviembre de 2014	21 – 22
Marshmallow	6.0 – 6.0.1	5 de octubre de 2015	23
Nougat	7.0 – 7.1.2	15 de junio de 2016	24 – 25
Oreo	8.0 – 8.1	21 de agosto de 2017	26-27
Pie	9.0	6 de agosto de 2018	28
Android 10	10.0	3 de septiembre de 2019	29

Tabla 2. Versiones de Android

	Última versión
	Versión antigua pero vigente
	Versión descontinuada

1.2.2.4 Cuota de mercado de las versiones de Android [8]

Como se ha comentado en los puntos anteriores, una de las características de Android es que es un sistema con un alto nivel de fragmentación. Esto se traduce en que los usuarios que utilizan dispositivos Android no utilizan la misma versión del sistema operativo. La cuota de mercado de Android, hasta finales del año 2018 se reparte de la siguiente manera (sin contar el nuevo sistema operativo Android 10 que su lanzamiento ha sido en fecha posterior):

Nombre	Distribución
Gingerbread	0.3%
Ice Cream Sandwich	0.3%
Jelly Bean	3.2%
KitKat	6.9%
Lollipop	14.5%
Marshmallow	16.9%
Nougat	19.2%
Oreo	28.3%
Pie	10.4%

Tabla 3. Cuota de mercado Versiones Android



Capítulo 2 – Plan de Trabajo

2.1 Contexto y justificación del trabajo

El objetivo del proyecto es realizar una aplicación móvil para dispositivos Android dedicada a la compra-venta de productos musicales.

Actualmente, el mercado de las aplicaciones móviles de compra-venta ha sufrido un boom considerable que se ha extendido entre la población debido a las facilidades que aportan los Smartphone actuales para este tipo de Apps. Gracias a las funcionalidades de estos dispositivos, podemos comprar de manera rápida y directa un producto de segunda mano que sabemos que está cerca de nuestra ubicación y poder hacerlo en persona en nuestra misma ciudad.

La mayoría de apps de compra – venta son genéricas, es decir, no están especializadas en un área en concreto. El punto negativo es que, si queremos comprar un producto específico que necesite más información de la que la app nos puede suministrar, hay que contactar con el vendedor para que por privado nos proporcione más información sobre el producto. Por ejemplo, existe la app *Wallapop* que es genérica donde se venden coches (entre otros muchos productos) y existe la app *coches.net* especializada 100% en la compra-venta de coches. Esta última permite aportar mayor información técnica que la de la app genérica *Wallapop*.

Aquí es donde nace la idea de nuestro trabajo. Hacer una app a medida para el sector musical en la que los compradores y vendedores puedan interactuar con la aplicación sin ningún tipo de límite. En el caso de la compra-venta de productos musicales, el mercado no dispone de una app de referencia que ofrezca información específica sobre el producto que se desea vender.

Para resolver este problema vamos a crear una app de compra-venta de productos musicales, la cual permita aportar al posible comprador el máximo de información necesaria para realizar la transacción con garantías de que el producto que se está vendiendo es el que realmente quiere comprar.

2.2 Objetivos del Trabajo

A continuación, se describen los requerimientos de la aplicación. Se dividen en requerimientos funcionales y requerimientos no funcionales

2.2.1 Requerimientos funcionales de alto nivel

2.2.1.1 Requerimientos funcionales (comportamiento del sistema)

- **Gestión de la autenticación de los usuarios:** El sistema deberá gestionar la entrada de los usuarios al sistema, identificándolos y permitiéndoles actuar según sus características. Comprobará para acceder al mismo, si el nombre de usuario y su correspondiente contraseña introducida son válidos.
- **Gestión de usuarios:** El sistema deberá ofrecer las funcionalidades necesarias para poder gestionar de manera completa los usuarios pertenecientes a la app. Debe permitir



darse de alta en el sistema, modificar sus datos personales y darse de baja en la app, así como otras tareas relacionadas con dicha gestión.

- **Gestión de Productos:** El sistema deberá ofrecer las funcionalidades necesarias para poder gestionar de manera completa los productos en la App. Deberá permitir la creación de un producto nuevo con sus correspondientes datos. También deberá permitir la eliminación de dicho producto.
- **Gestión de imágenes y fotos:** El sistema deberá ofrecer las funcionalidades necesarias para poder gestionar de manera completa las fotografías e imágenes de la App. Debe permitir la subida, visualización y eliminación de las fotografías, así como otras tareas relacionadas con dicha gestión.
- **Gestión de mensajes:** El sistema deberá ofrecer las funcionalidades necesarias para poder gestionar de manera completa los mensajes de la App entre los usuarios.
- **Gestión de las notificaciones:** El sistema deberá ofrecer las funcionalidades necesarias para poder gestionar de manera completa las notificaciones de la App.
- **Gestión de la Geolocalización:** El sistema deberá ofrecer las funcionalidades necesarias para poder gestionar de manera completa las ubicaciones de los productos, las ubicaciones de las personas (ubicación actual del dispositivo). También deberá mostrarlo en un mapa.

2.2.1.2 Requerimientos no funcionales

- **Requerimientos de seguridad:** El sistema deberá ser seguro para los usuarios (seguridad de datos, autenticidad de la información, privacidad, etc.).
- **Requerimientos de usabilidad:** El sistema no deberá obligar al usuario a realizar un esfuerzo alto para aprender, usar, introducir datos e interpretar los resultados obtenidos en la app.
- **Requerimientos de eficiencia:** El sistema deberá ser eficiente en cuanto a tiempo de respuesta, número de operaciones por segundo, consumo de recursos de memoria, etc.
- **Requerimientos de entorno:** El sistema deberá desenvolverse en dispositivos con sistema operativo Android.
- **Requerimientos de desarrollo:** El sistema estará desarrollado en el lenguaje de programación Java (Android).

2.2.2 Elección de la Plataforma

La plataforma elegida para desarrollar **de forma nativa** la App ha sido **Android**. Uno de los motivos principales de la elección es por el conocimiento previo que se dispone (tras haber cursado la asignatura Tecnología y Desarrollo en Dispositivos Móviles). Otro motivo de peso en la elección de este sistema operativo ha sido por las facilidades que ofrece el ecosistema Android para los desarrolladores. Android ofrece resultados de gran rendimiento con una experiencia de usuario satisfactoria. Por este motivo, se han descartado de la elección opciones híbridas que, aunque puedan resultar más sencillas, el resultado a nivel de rendimiento es bastante peor que el de una app nativa pura.

2.2.3 Dispositivos a los que se dirige

Los dispositivos podrán ser Smartphone y Tablets en versiones superiores a la API 21, correspondiente a Android 5.0 - Lollipop.



2.3 Enfoque y método seguido

Para poder afrontar este proyecto, previamente se estuvo realizando un estudio de las apps de compra-venta más exitosas del mercado para poder obtener ideas y conclusiones.

La estrategia para llevar a cabo el trabajo ha sido la de desarrollar el producto desde cero. Basándonos en dichas apps de compra-venta que existen en la actualidad, añadiendo funcionalidades exclusivas para que nuestra app sea única, intentando dar un “toque” de personalidad y mejorar lo que ya existe en el mercado (dentro del sector musical).

Ésta será la estrategia más apropiada para cumplir los objetivos ya que toda app de compra-venta debe tener funcionalidades en común. Nos inspiraremos en lo que ya hay hecho y que funciona (*wallapop*, *coches.net*, *mil anuncios*, etc.). Hacer algo diferente a lo que ya funciona, podría llevarnos más tiempo y no dar con la clave. Una vez que tengamos la funcionalidad básica de una app de compra-venta, añadiremos funcionalidades específicas y una interfaz gráfica exclusiva de esta app que la diferencie de las demás y la especialice dentro del sector deseado.

Se ha decidido usar metodología ágil debido al tipo de proyecto que se va a desarrollar, ya que será una app móvil con un alto grado de mejoras posibles durante el transcurso de la misma. También podrá existir cambios de requisitos durante el desarrollo del proyecto. Durante la fase de diseño se utilizan conceptos de la metodología del diseño centrado en el usuario (DCU).

Fases establecidas para el desarrollo del trabajo:

Fase 1 – Plan de Trabajo

Alcance y los objetivos del proyecto que formarán el plan de trabajo.

Fase 2 – Diseño y Arquitectura

Durante esta fase se realiza el diseño, los prototipos de la aplicación y la evaluación. También se definen los casos de uso y la arquitectura técnica de la App. Se estructura de la siguiente manera:

- **Diseño centrado en el usuario**
 - **Usuarios y contexto de uso:** Conocer las características de los usuarios, sus necesidades, sus objetivos para poder detectar las funcionalidades que deberá tener la aplicación.
 - **Diseño conceptual:** Elaboración de los escenarios de uso.
 - **Prototipado:** Se creará la representación de las pantallas de la aplicación, que permitirá comunicar decisiones de diseño y evaluarlo antes de desarrollar el producto final.
 - **Evaluación:** Definir las preguntas y las tareas de evaluación para los usuarios.
- **Definición de los casos de uso:** Se definen los casos de uso para establecer las funcionalidades de la aplicación.
- **Diseño de la arquitectura:** En esta fase se definirá la arquitectura del sistema, identificando las entidades que se representarán en la base de datos, las clases y objetos que se utilizarán para gestionar los diferentes procesos y la estructura de la API que servirá para realizar las peticiones enviadas al servidor desde un cliente.



Fase 3 – Implementación

Se compondrá del desarrollo de la App y las pruebas correspondientes. Como vamos a utilizar metodología ágil, tras la finalización de cada Sprint se dispondrá del entregable correspondiente. Consta de:

- **Desarrollo:** Codificación de la aplicación.
- **Pruebas:** Se definirán las pruebas realizadas para probar el funcionamiento de la aplicación.
- **Aceptación del producto:** Tras cada entrega funcional, se evaluará y se aceptará el entregable.
- **Documentación de la memoria:** Tras cada entrega, se añadirá a la memoria el correspondiente avance y los resultados obtenidos.

Fase 4 – Publicación y entrega final.

Se finalizará todo lo que quede pendiente y se entregará el producto con su memoria y su presentación finalizada.

2.4 Planificación del Trabajo

2.4.1 Recursos y estimación de las tareas

Recursos necesarios para la realización del trabajo:

- Entorno de desarrollo Android Studio
- Librerías
- Firebase
- Software desarrollo de prototipos
- Software de modelado
- Procesador de texto
- Software de diapositivas para realizar la presentación

A continuación, se detalla la estimación de las tareas medidas en horas:

Nombre de la Tarea	Duración	Comienzo	Fin
• Elección del Tema	2 horas	21/02/20	21/02/20
• Creación de la Propuesta	2 horas	24/02/20	24/02/20
FASE 1 - Plan de Trabajo	20 horas	26/02/20	04/03/20
• Lectura de Documentos	4 horas	26/02/20	28/02/20
• Objetivos y requerimientos	10 horas	29/02/20	02/03/20
• Planificación del TFM	6 horas	03/03/20	04/03/20
FASE 2 - Diseño y Arquitectura	83 horas	05/03/20	01/04/20
• Análisis Funcional de la App	15 horas	05/03/20	08/03/20
• Diseño Conceptual	18 horas	09/03/20	12/03/20
• Prototipado	15 horas	12/03/20	16/03/20
• Evaluación	5 horas	17/03/20	17/03/20
• Definición de los Casos de Uso	20 horas	18/03/20	26/03/20
• Diseño de la Arquitectura	10 horas	27/03/20	01/04/20
FASE 3 - Implementación	110 horas	02/04/20	13/05/20
• Desarrollo del Back-End	60 horas	02/04/20	13/05/20
• Desarrollo del Front-End	40 horas	02/04/20	13/05/20
• Pruebas	10 horas	02/04/20	13/05/20
FASE 4 - Entrega final (3 Entregables)	50 horas	14/05/20	10/06/20



Preparación defensa TFM	8 horas	22/06/20	26/06/20
Total	275 horas		

Tabla 4. Estimación de tareas

*La fase 3 de implementación será iterativa y se repetirán a lo largo de cada Sprint.

Estimo una **dedicación de 275 horas** dedicadas al proyecto. En **días laborables** se dedicarán aproximadamente **4 horas**. En **fin de semana** la dedicación será de aproximadamente **6 horas**.

Diagrama de Gantt con las tareas: [10]

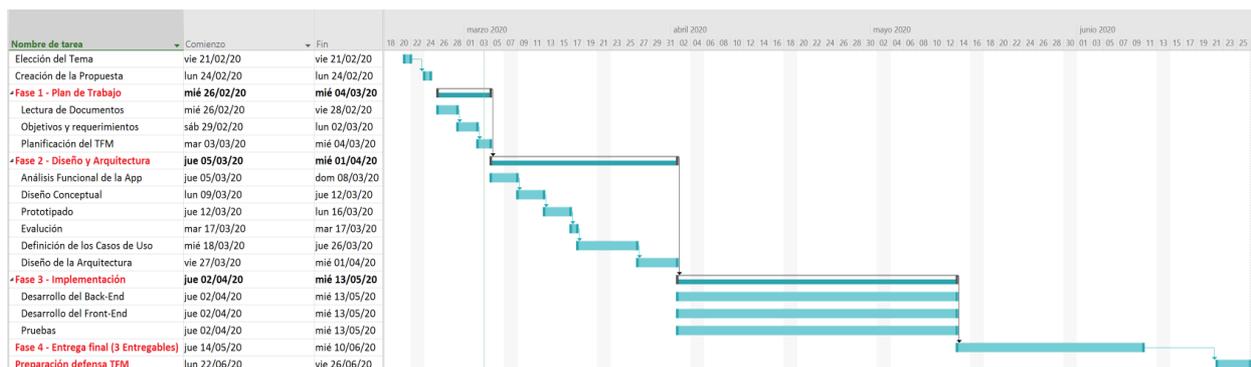


Ilustración 2. Diagrama de Gantt

2.4.2 Análisis de costes

En este apartado se presenta de una forma simulada el presupuesto necesario para llevar a cabo el proyecto si éste tuviera un carácter comercial.

En la siguiente tabla se muestran las horas dedicadas al proyecto. Sumando todas las fases, el resultado determina una dedicación de **263 horas**. En este cálculo no contamos ni el tiempo dedicado a la elección del plan ni las horas dedicadas a preparar la defensa.

Fase	Horas
Plan de Trabajo	20 horas
Diseño y arquitectura	83 horas
Implementación	110 horas
Publicación y entrega final	50 horas

Tabla 5. Estimación de horas dedicadas al proyecto

Se debería tener en cuenta el coste del material necesario (equipos, licencias y otros costes asociados), pero al ser un Trabajo de Fin de Master y no un proyecto dedicado a fines comerciales, se ha decidido obviarlo.

Realizando la estimación del precio a la hora de trabajo de un ingeniero Informático a dedicación única, hemos calculado unos 45€/hora. Lo que multiplicados por las horas dedicadas (sin tener en cuenta los materiales), la cifra asciende a 11.835€.

En la siguiente tabla mostramos el presupuesto global del proyecto:

Concepto	Importe
Coste personal	11.835€
Coste material	0€
Base Imponible	11.835€
I.V.A 21%	2.485,35€
Total	14.320,35€

Tabla 6. Estimación de costes del proyecto



Capítulo 3 – Caso de Estudio [9]

3.1 Análisis

El objetivo principal de la fase de análisis será el de conocer las características de los usuarios, sus necesidades, sus objetivos y el contexto de uso de la aplicación con el fin de poder detectar correctamente las funcionalidades que deberá tener la aplicación y poder satisfacer a los usuarios.

3.1.1 Justificación de los métodos de indagación

Para empezar el análisis y poder conocer a los usuarios, sus necesidades y sus objetivos, intentaremos responder a una serie de preguntas que serán claves para conocer el tipo de usuario al que nos enfrentamos:

- ¿Quiénes son los usuarios que utilizarán la aplicación?
- ¿Por qué utilizarán la aplicación?
- ¿Qué tareas realizarán con la aplicación?
- ¿Están familiarizados con aplicaciones similares?
- ¿Qué tipo de dispositivos utilizan?

Para poder realizar esta tarea, utilizaremos varios métodos de indagación:

- Observación e investigación contextual
- Encuestas mediante formularios breves en las cuales preguntaremos sobre el nivel de acogida de las apps similares a la nuestra.

Utilizaremos estas técnicas debido a que al ser una aplicación de compra-venta de productos musicales, podemos realizar la observación de los usuarios con aplicaciones de compra-venta genéricas, como puede ser *wallapop* o *milanuncios*. Al tener el mismo funcionamiento que la app que se va a desarrollar (salvo que está especializada en música) podremos obtener ciertos datos reales de como podría funcionar la app entre el público. Por lo tanto, en las entrevistas se preguntará si están familiarizados con las aplicaciones de compra-venta y si se fían de esta forma de comprar y vender. Con la parte de observación e investigación contextual, se observará a mi círculo más cercano y les preguntaré su opinión sobre este tipo de aplicaciones y si realmente las utilizan en su vida cotidiana.

También creo conveniente realizar una comparación con las aplicaciones ya existentes (*mil anuncios*, *coches.net*, *chicfy*, etc.), analizando como las valoran los usuarios y que mejoras podrían tener. Al extraer y al analizar estas informaciones, podremos obtener un resultado realista sobre la acogida que tendría esta aplicación entre el público.

3.1.1.1 Planteamiento

Cómo hemos comentado en el punto anterior, la investigación la hemos empezado a realizar entre las personas que conforman mi círculo más cercano. Se trataba de observar a los usuarios en su entorno habitual. Después de observar a mis hermanos, amigos, pareja, primos, padres y tíos, he llegado a las siguientes conclusiones:

- La aplicación tendría más éxito entre las personas de edad joven y de edad media.
- La aplicación tendría más éxito entre las personas de menor poder económico.



Ni los padres, ni los tíos (alrededor de 55-60 años) tenían en sus dispositivos alguna aplicación de compra-venta, y por lo que comentaron no se fiaban de este tipo de compras. En cambio, primos, amigos, hermanos y pareja tenían todos al menos una aplicación instalada en sus teléfonos y habían realizado alguna transacción con ella. Esto nos proporciona una primera estimación del perfil de usuario que podría utilizar nuestra aplicación.

A continuación, hemos realizado entrevistas informales a gente anónima. Hay personas que les hemos parado a la salida de tiendas musicales, y otras que les hemos preguntado paseando por las calles. Las entrevistas tienen la siguiente estructura:

Entrevistas informales/Encuestas

Datos Personales

- Edad
- Sexo
- Localidad
- Trabajo

Preguntas

1. ¿Dispone de Smartphone/Tablet?
2. ¿Hay alguna aplicación en su Smartphone/Tablet de compra-venta de productos?
3. En caso afirmativo, ¿las ha utilizado alguna vez? ¿Cómo ha sido la experiencia?
4. ¿Sabe usted tocar algún instrumento musical?
5. ¿Tiene planes de adquirir en un periodo corto de tiempo un instrumento musical?
6. En caso afirmativo, ¿qué preferiría para su economía: instrumentos nuevos de gama baja o usados de gama media?
7. ¿Se fiaría usted de este tipo de apps y de las facilidades que aportar a los usuarios para realizar las transacciones?

3.1.1.2 Desarrollo de las entrevistas

Se ha entrevistado a cinco personas. De estas cinco personas, dos a la salida de una tienda de instrumentos, otras dos por la calle (a priori sin relación con la música), y otra a un batería retirado de una banda de rock. Los resultados han sido los siguientes:

Formulario 1

Datos Personales

- **Edad:** 22.
- **Sexo:** Femenino
- **Localidad:** Madrid
- **Trabajo:** Estudiante

Preguntas

1. ¿Dispone de Smartphone/Tablet? Si.
2. ¿Hay alguna aplicación en su Smartphone/Tablet de compra-venta de productos? Si, tengo dos. Wallapop y coches.net



3. **En caso afirmativo, ¿las ha utilizado alguna vez? ¿Cómo ha sido la experiencia?** Si. La he utilizado un par de veces para comprar un microondas y una plancha, ya que vivo en un piso compartido. La experiencia muy positiva. Por muy poco dinero, hemos adquirido dos productos que funcionan perfectamente y que el dueño los iba a tirar porque se quería comprar electrodomésticos nuevos.
4. **¿Sabe usted tocar algún instrumento musical?** No
5. **¿Tiene planes de adquirir en un periodo corto de tiempo un instrumento musical?** Si. Me gustaría aprender a tocar el ukelele.
6. **En caso afirmativo, ¿Qué preferiría para su economía instrumentos nuevos de gama baja o usados de gama media?** Debido a que los ukeleles son bastante baratos, creo que me decantaré por ir a una tienda musical y que me recomienden uno y comprármelo. En un futuro, me gustaría aprender a tocar la guitarra. Ahí creo que si que me decantaré por comprarme una de segunda mano de buena calidad antes que una barata de calidad pésima.
7. **¿Se fiaría usted de este tipo de apps y de las facilidades que aportar a los usuarios para realizar las transacciones?** Si me baso en mis experiencias puedo afirmar que este tipo de apps son fiables y que los usuarios (por regla general) cumplen con lo acordado previamente.

Formulario 2

Datos Personales

- **Edad:** 35
- **Sexo:** Masculino
- **Localidad:** Madrid
- **Trabajo:** Ingeniero informático

Preguntas

- **¿Dispone de Smartphone/Tablet?** Si. Tanto de Smartphone como de Tablet
- **¿Hay alguna aplicación en su Smartphone/Tablet de compra-venta de productos?** Si. Tengo varias. Mil Anuncios y Wallapop
- **¿En caso afirmativo, las ha utilizado alguna vez? ¿Cómo ha sido la experiencia?** Si. Tengo dos guitarras (una acústica y otra eléctrica) y las compré por Wallapop a un chico que se había comprado unas guitarras mejores y por eso vendía las suyas. Me salieron baratitas y son buenas.
- **¿Sabe usted tocar algún instrumento musical?** Si. La guitarra y el bajo
- **¿Tiene planes de adquirir en un periodo corto de tiempo un instrumento musical?** Si. Estaría interesado en comprarme una guitarra eléctrica de la marca Gibson. Una Les Paul, aunque su precio ronda los 1200€, espero conseguirla de segunda mano por 600 o así.
- **En caso afirmativo, ¿Qué preferiría para su economía: instrumentos nuevos de gama baja o usados de gama media?** Pienso que para la gente que no disponemos de mucho dinero, es mejor realizar una búsqueda y comprar un producto de segunda mano a un precio bastante inferior al nuevo, siempre y cuando el producto esté en condiciones, claro está.
- **¿Se fiaría usted de este tipo de apps y de las facilidades que aportar a los usuarios para realizar las transacciones?** Si. Me fio totalmente. Mis experiencias han sido positivas.

Formulario 3

Datos Personales

- **Edad:** 56
- **Sexo:** Masculino
- **Localidad:** San Sebastián de los Reyes
- **Trabajo:** Autónomo



Preguntas

- **¿Dispone de Smartphone/Tablet?** Si. Por mi trabajo me paso pegado a él las 24 horas.
- **¿Hay alguna aplicación en su Smartphone/Tablet de compra-venta de productos?** No. Nunca he utilizado ese tipo de aplicaciones
- **¿En caso afirmativo, las ha utilizado alguna vez? ¿Cómo ha sido la experiencia?**
- **¿Sabe usted tocar algún instrumento musical?** Si. Toqué la batería en un grupo durante muchos años. Aún la sigo conservando y toco cuando mi trabajo me da una tregua y me lo permite.
- **¿Tiene planes de adquirir en un periodo corto de tiempo un instrumento musical?** No lo descarto. Me gusta mucho el violín.
- **En caso afirmativo, ¿Qué preferiría para su economía: instrumentos nuevos de gama baja o usados de gama media?** Prefiero instrumentos nuevos y pagar un poco más a instrumentos de otra persona que se desconoce como los ha tratado. Soy un poco maniático para esas cosas, y me gusta comprarme las cosas nuevas.
- **¿Se fiaría usted de este tipo de apps y de las facilidades que aportar a los usuarios para realizar las transacciones?** No, la verdad que no me fio de ese tipo de app. Puedes dar con una persona buena o puedes dar con una que te engañe. Prefiero la tienda de toda la vida y poder reclamar a alguien en el caso de que el instrumento salga defectuoso.

Formulario 4

Datos Personales

- **Edad:** 30
- **Sexo:** Femenino
- **Localidad:** Madrid
- **Trabajo:** Ingeniera de telecomunicaciones

Preguntas

1. **¿Dispone de Smartphone/Tablet?** Si. De ambas. Utilizo el Smartphone para el día a día y la Tablet para ver series en la cama.
2. **¿Hay alguna aplicación en su Smartphone/Tablet de compra-venta de productos?** Si. En concreto solo tengo *Wallapop*.
3. **¿En caso afirmativo, las ha utilizado alguna vez? ¿Cómo ha sido la experiencia?** Si. Nunca he tenido ningún problema. Lo último que he vendido ha sido unas gafas de sol que me quedaban grandes y estaban nuevas. Me las quería quitar de en medio por lo que el que me las compró pilló un buen chollo.
4. **¿Sabe usted tocar algún instrumento musical?** No, nunca he tenido la paciencia suficiente para aprender.
5. **¿Tiene planes de adquirir en un periodo corto de tiempo un instrumento musical?** No, en estos momentos no tengo pensado aprender a tocar ningún instrumento, aunque me gusta mucho la música.
6. **En un hipotético caso, ¿Qué preferiría para su economía si usted deseara comprar: instrumentos nuevos de gama baja o usados de gama media?** Instrumentos usados de una calidad superior
7. **¿Se fiaría usted de este tipo de apps y de las facilidades que aportar a los usuarios para realizar las transacciones?** Si.



Formulario 5

Datos Personales

- **Edad:** 35
- **Sexo:** Masculino
- **Localidad:** Palencia
- **Trabajo:** Taxista

Preguntas

1. **¿Dispone de Smartphone/Tablet?** Si. De ambas. Aunque la Tablet casi no la utilizo. El Smartphone a diario por mi trabajo, con las aplicaciones de navegación. También tengo *Spotify* y me conecto al bluetooth del taxi, cuando no llevo gente. Me encanta la música.
2. **¿Hay alguna aplicación en su Smartphone/Tablet de compra-venta de productos?** Si. Una de coches de segunda mano.
3. **En caso afirmativo, las ha utilizado alguna vez? ¿Cómo ha sido la experiencia??** Si. Pero para tantear el mercado y ver diferencias entre concesionarios y particulares. Todavía no la he utilizado, ya que un coche no se compra todos los días.
4. **¿Sabe usted tocar algún instrumento musical?** Si, toco el bajo. Voy a clases para aprender. Acabo de salir de la tienda de música de comprarme unas cuerdas.
5. **¿Tiene planes de adquirir en un periodo corto de tiempo un instrumento musical?** Si. En cuanto acabe las clases me quiero comprar un bajo semi-profesional.
6. **En caso afirmativo, ¿qué preferiría para su economía si usted deseara comprar: instrumentos nuevos de gama baja o usados de gama media?** Instrumentos usados de una calidad superior. No me puedo permitir el dinero que cuesta el bajo nuevo que me quiero comprar. Tendré que conformarme con uno de segunda mano.
7. **¿Se fiaría usted de este tipo de apps y de las facilidades que aportar a los usuarios para realizar las transacciones?** Si. Nunca he comprado, pero tengo muchos amigos que son “adictos” a comprar cosas y al rato venderlas. Siempre les ha ido bien por lo que si que me fiaría.

3.1.1.3 Resultados y Conclusiones

El principal resultado de la fase de análisis es determinar el tipo de usuarios que utilizarán nuestra aplicación. Con lo extraído en estos puntos, podemos identificar el perfil de usuarios que en gran medida utilizará nuestra app: será un perfil de joven-mediana edad, que se maneje con el mundo de las tecnologías y que se fie de ellas. Su nivel económico será bajo-medio. Hemos comprobado que este tipo de app entre la gente de más de 50 años no suele llamar mucho la atención por varios motivos. El primero porque a nivel económico (por regla general), están mejor que cualquier persona de entre 20 y 30 años, por lo cual se pueden permitir comprar instrumentos nuevos. El segundo motivo es que la gente de esa edad, suele ser más reacio a la compra por internet, gustándole más lo tradicional. Es decir, lo de toda vida: ir a una tienda, probar el instrumento, comprarlo y si tengo problemas con él, saber a quien reclamarlo y que se tengan ciertas garantías ante cualquier desperfecto del producto. También son reacios a la compra de productos usados, por lo que creo que estos usuarios no serían por regla general usuarios de nuestra App.

El perfil del vendedor será similar al del comprador, ya que la mayoría de gente que utilice la app lo utilizará para comprar lo que necesita y vender lo que no necesita para recuperar dinero.



3.1.2 Usuarios y contexto de uso

Después de realizar las entrevistas y la observación en la fase anterior, se ha detectado un único perfil, aunque engloba varios sub-perfiles. El perfil se basa en una agrupación de usuarios con características similares. Estas características sobre todo son edad y nivel económico. Al ser personas de una edad joven-media, deducimos que tienen un conocimiento básico-intermedio de los dispositivos móviles. También presuponemos que, en su mayoría, utilizarán esta app personas que no tengan demasiado poder económico. Dentro de este perfil podremos separar varios sub-perfiles:

- Joven sin poder económico.
- Persona que tiene muchos gastos y no puede permitirse pagar lo que vale un instrumento nuevo de calidad.
- Vendedor que vende su producto porque necesita el dinero para poder adquirir algo mejor.

Estos sub-perfiles tienen muchas similitudes entre si, por lo tanto, se considerará que existe un tipo de perfil que use la app para comprar y vender por diferentes motivos. La mayoría de usuarios serán en algún momento compradores y en algún momento vendedores.

Para el uso de esta aplicación no se requiere que los usuarios posean un conocimiento experto en dispositivos móviles, pero si que se manejen con el Smartphone para poder descargar, utilizar e interactuar con la aplicación.

A continuación, en la tabla siguiente, se describen los siguientes puntos obtenidos en la fase de indagación y que servirán como punto de partida para la fase de diseño:

- **Características:** Principales características de los usuarios que pertenecen a los perfiles detectados
- **Contextos de uso:** Se detallan los contextos de uso en los cuales los usuarios utilizarán la aplicación: ¿Dónde?, ¿cuándo?, ¿en qué entorno?
- **Análisis de tareas:** Se explican las principales tareas que proporcionará la aplicación para alcanzar sus objetivos.
- **Listado de características descubiertas:** Listado de características y elementos que gracias a la fase de indagación se han descubierto que deben estar presentes en la interfaz de la aplicación.

Perfil	Cliente de la APP
Características	<ul style="list-style-type: none"> • Personas con edad comprendida entre los 16 y los 50 años. • Género indistinto. • Clase social media-baja. • Ámbito profesional músicos y gente que se esté iniciando. • Experiencia con los dispositivos móviles: baja-media • Motivaciones: Comprar lo que necesitan a un precio más económico. Vender todo aquello que no se use y que ocupe espacio en casa para recuperar una parte de lo que costó.
Contextos de uso	<ul style="list-style-type: none"> • ¿Dónde? En cualquier lugar con conexión a Internet. • ¿Cuándo? En tiempo de ocio y a diario. • ¿Qué entorno? En entornos domésticos.
Análisis de tareas	<ul style="list-style-type: none"> • Tarea 1 – Registro del usuario • Tarea 2 - Acceso del usuario en la app • Tarea 3 – Consulta del menú • Tarea 4 – Buscar productos



	<ul style="list-style-type: none"> • Tarea 5 – Visualizar ficha del producto • Tarea 6 – Comunicación Chat (Comprar/Vender) • Tarea 7 – Subir producto para vender • Tarea 8 – Modificar perfil de usuario • Tarea 9 – Darse de baja de la app
Listado de características descubiertas	<ul style="list-style-type: none"> • Interfaz más sencilla e intuitiva. • Facilidad de interacción • Abrir el abanico de productos no solo a instrumentos musicales sino también tendría éxito componentes musicales como pueden ser cascos o auriculares, mesas de dj, altavoces, micrófonos, etc. • Añadir una opción para que cada usuario pueda marcar un producto como favorito y que se añada automáticamente a una lista para poder realizar un seguimiento de los mismos. • Facilitar la autenticación con otras plataformas mediante el Single Sign On.

Tabla 7. Resultados fase de indagación

3.2 Diseño Conceptual

El principal objetivo de esta fase es la de realizar el diseño de la aplicación teniendo en cuenta todo lo descubierto en los puntos anteriores. Este apartado estará dividido en dos partes:

- Escenarios de uso
- Diagramas de interacción

3.2.1 Escenarios de uso

Un escenario de uso describe desde el punto de vista del usuario como utilizará la aplicación móvil en un contexto concreto. Con su construcción se podrá determinar necesidades de los usuarios y de diseño. Una vez definidos los escenarios, serán de utilidad para conceptualizar la estructura de la aplicación y los flujos de interacción.

Escenario 1	
Perfil de usuario	Usuario de la App
Personaje	Raúl Gutiérrez tiene 18 años, es estudiante y vive en casa de sus padres. Raúl, como la mayoría de los jóvenes, usa a diario el móvil y se maneja de forma fluida con los dispositivos y sus apps. La app no sería un problema para él ya que el pertenece a la “generación digital”.
Descripción	Al ser joven y no trabajar, no dispone de mucho dinero para comprarse una guitarra eléctrica para poder empezar a aprender a tocar. Tiene dos opciones: comprarse una guitarra nueva pero bastante mala o comprarse una guitarra eléctrica de una marca mejor, pero de segunda mano. Raúl se descarga la app y se pone a buscar una guitarra eléctrica de segunda mano, que esté bien conservada a un precio asequible que no se lo podría permitir si la guitarra estuviera nueva.
Contexto	En el entorno doméstico, un día de diario. Lleva tiempo esperando y cree que ha llegado el momento de comprarse una guitarra.
Objetivos	Comprar una guitarra eléctrica de calidad media a un precio bajo para aprender a tocarla.
Tareas	Tarea 6 – Comunicación Chat con otro usuario para comprar producto.
Necesidad de información	Registro en el sistema e identificación
Funcionalidades necesarias	<ul style="list-style-type: none"> • Registrar el usuario en la app • Autenticarse en la app • Buscar el producto deseado • Realizar la compra mediante acuerdo con el vendedor



Desarrollo de las tareas	Logarse en el sistema y realizar las búsquedas necesarias para encontrar el producto deseado y comprarlo.
---------------------------------	---

Tabla 8. Casos de uso. Escenario 1

Escenario 2	
Perfil de usuario	Usuario de la App
Personaje	Daniel Marcos es un chico de 30 años. Tiene su trabajo y un grupo de música. Lleva tocando la batería desde los 15 años. Utiliza a diario el teléfono móvil para las redes sociales y plataformas de música en streaming. La app no es un problema, ya que ha crecido en la era de internet y confía en este tipo de aplicaciones.
Descripción	A lo largo de estos 15 años en los que Daniel lleva tocando la batería, se ha comprado 3 baterías. Según iba aprendiendo a tocar ha ido comprándose una mejor hasta acabar con una profesional. Tiene 2 baterías prácticamente nuevas en el trastero. Es una pena porque las podría utilizar una persona que estuviera aprendiendo. Aprovecha que existe la app, para quitárselas de en medio, ganarse un dinero y dar la oportunidad a otra persona de comprarlas que no tenga dinero para ello.
Contexto	En su casa, un día de diario, cansado de acumular instrumentos que tienen valor fuera de su casa.
Objetivos	Vender los instrumentos que tiene en el trastero que nadie utiliza.
Tareas	Tarea 7 - Cargar la información del producto para poder venderlo. Tarea 6- Esperar a que le comunique algún usuario su interés.
Necesidad de información	Registro en el sistema y cargar la información del producto (fotos, detalles, características).
Funcionalidades necesarias	<ul style="list-style-type: none"> • Registrar el usuario en la app • Autenticarse en la app • Cargar la información, detalles, fotos del producto a vender
Desarrollo de las tareas	Identificación y validación en el sistema y registro de la información del producto.

Tabla 9. Casos de uso. Escenario 2

Escenario 3	
Perfil de usuario	Usuario de la App
Personaje	Jesús Sánchez es un señor de 57 años que trabaja en una empresa de alimentación. Siempre le ha gustado tocar la guitarra y componer sus propias canciones. Su manejo con los dispositivos móviles es bastante pobre, ya que no le gustan las tecnologías y siempre que tiene algún problema su hija se lo soluciona.
Descripción	Jesús quiere vender las guitarras que no utiliza debido a que necesita el dinero y no quiere que le ocupen espacio. Su hija le habla de nuestra aplicación, pero el desconfía de lo virtual y le da “pereza” mostrar atención para aprender a crearse un perfil y subir la ficha del producto, con su descripción y sus fotos. Su hija, le anima a hacerlo ayudándole y Jesús consigue vender las 3 guitarras a un precio bastante interesante tanto para él como para el comprador.
Contexto	En su casa, un sábado por la tarde hablando con su hija surge la idea.
Objetivos	Vender los instrumentos que ya no utiliza para conseguir dinero.
Tareas	<ul style="list-style-type: none"> • Tarea 7 – Subir producto para vender
Necesidad de información	Registro en el sistema y cargar la información del producto
Funcionalidades necesarias	<ul style="list-style-type: none"> • Registrar el usuario en la app • Autenticarse en la app • Cargar la información, detalles, fotos del producto a vender
Desarrollo de las tareas	Identificación y validación en el sistema y registro de la información del producto.

Tabla 10. Casos de uso. Escenario 3



Flujos de interacción

A continuación, presentamos el flujo de interacción de la aplicación con las tareas que describimos en la fase anterior. [11]

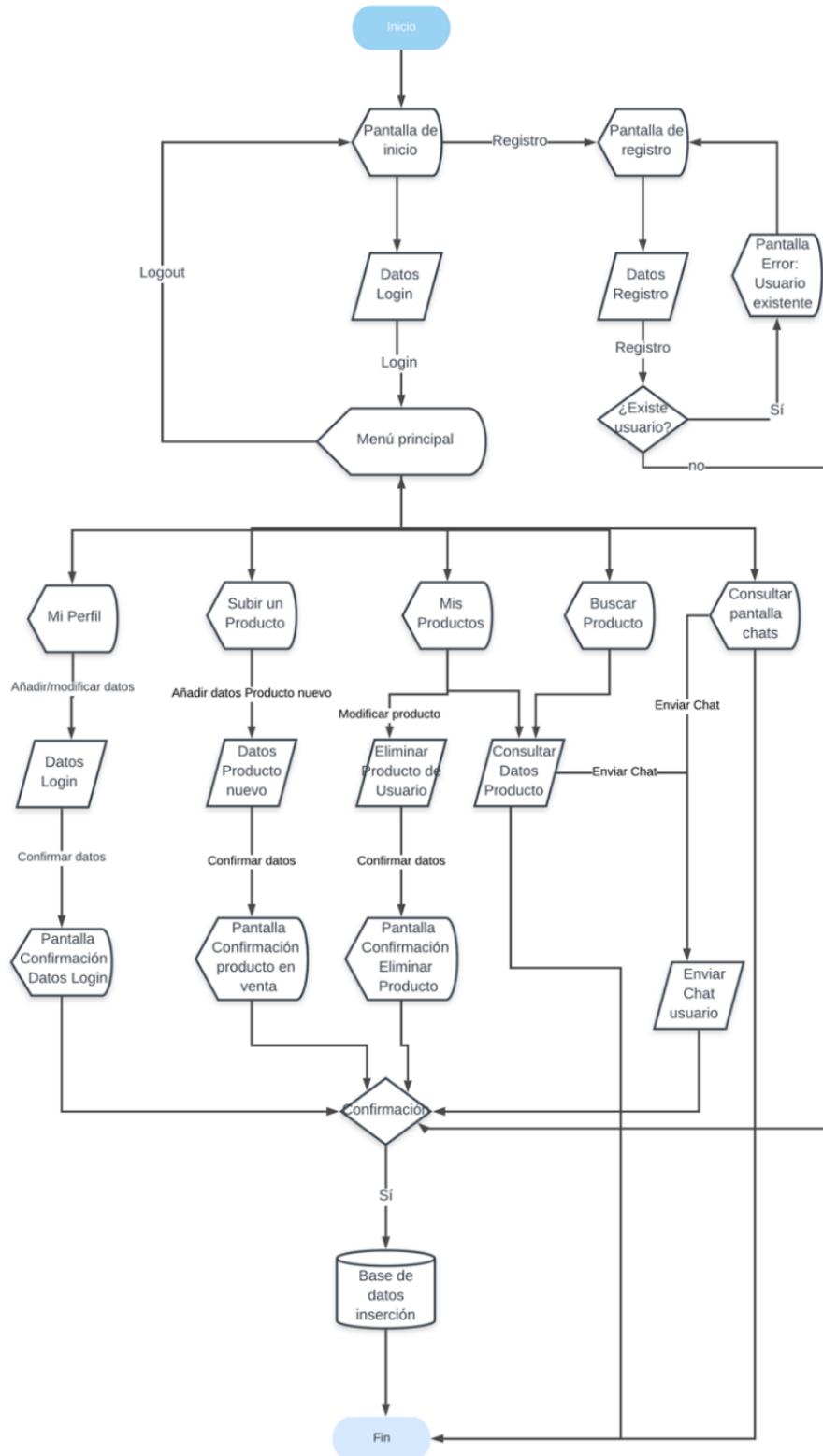


Ilustración 3. Flujos de interacción



3.3 Prototipado

Tomando los flujos de interacción definidos en la fase anterior, realizaremos un prototipo horizontal de alta fidelidad. Un prototipo es una representación de la aplicación, que permite comunicar decisiones de diseño y evaluarlo antes de desarrollar el producto final. Su versatilidad hace que sea sencillo y económico introducir modificaciones en su diseño e iterar incorporando mejoras fruto de la discusión con los miembros del equipo o de los resultados que obtenidos en la evaluación. Para realizar el prototipo se han tenido en cuenta los conceptos expuestos en la metodología “Diseño Centrado en el Usuario”. (DCU). Este apartado se divide en dos partes, comenzando por los sketches y posteriormente se definirá el prototipo horizontal de alta fidelidad.

3.3.1 Sketches

En este punto se definirán los Sketches, que son los primeros argumentos de diseño que tendrá la aplicación, hechos a mano alzada.

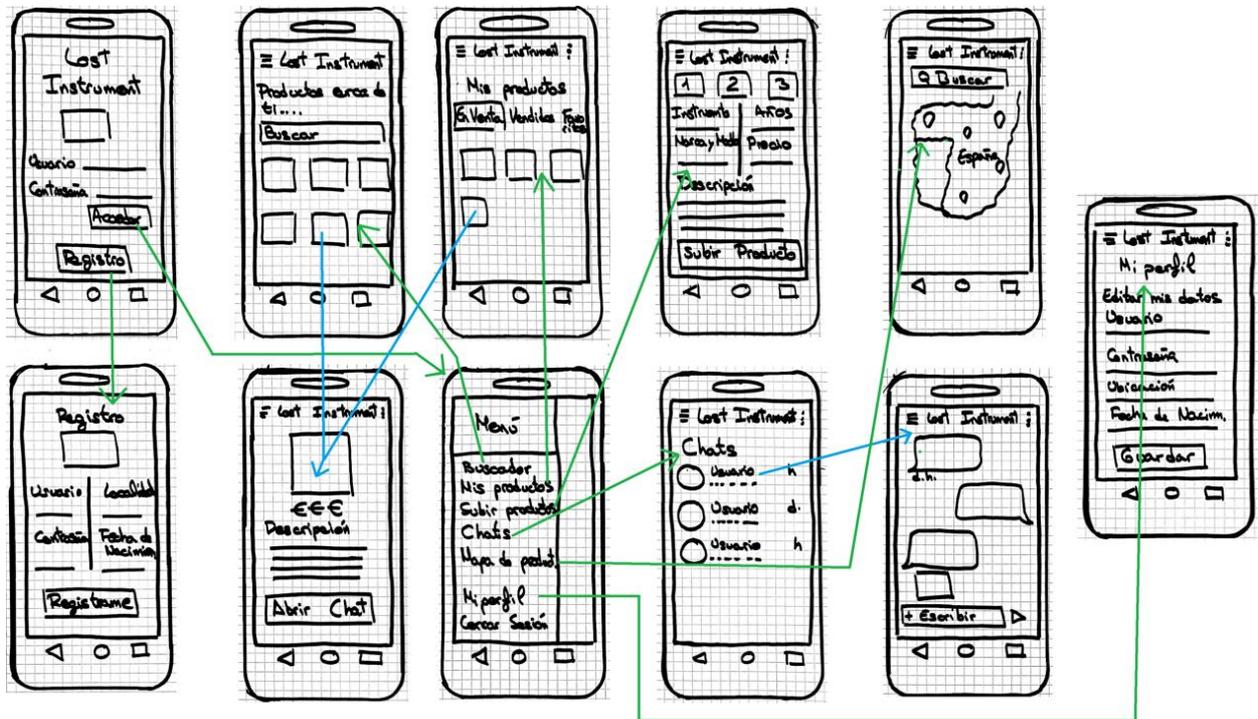


Ilustración 4. Sketches

3.3.2 Prototipos [12]

A continuación, presentaremos las pantallas del prototipo, explicando los elementos principales que forman cada diseño.



3.3.2.1 Pantalla de inicio

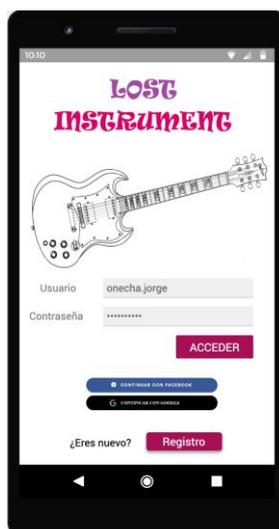


Ilustración 5. Prototipo: Pantalla de inicio

Esta pantalla será la primera pantalla de la aplicación. En ella el usuario se podrá logar introduciendo su nombre de usuario y contraseña, o mediante algún método de single sign on (Facebook, Google, etc.). Pulsando el botón acceder podremos entrar en la aplicación. Si no estamos registrados podremos acceder a la pantalla de registro mediante el botón Registro.

3.3.2.2 Pantalla de registro de usuario

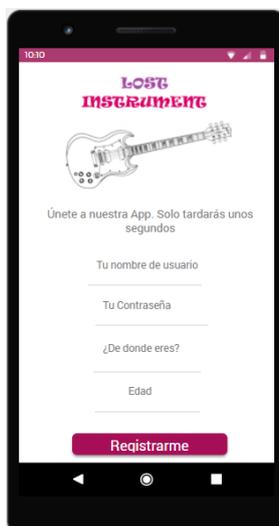


Ilustración 6. . Prototipo: Pantalla de registro de usuario

En esta pantalla se podrá crear un nuevo usuario rellendo los datos que se solicitan en el formulario (nombre de usuario, contraseña, fecha de nacimiento, etc.). Una vez relleno, se pulsará el botón “Registrarme” y el usuario se almacenará en la base de datos pudiendo acceder a la aplicación.



3.3.2.3 Menú principal

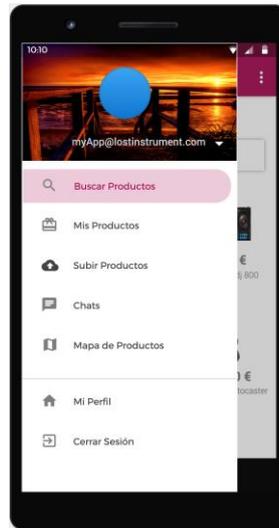


Ilustración 7. Prototipo: Menú principal

Tendremos un icono con forma de tres rayitas horizontales en la parte superior izquierda y cada vez que se pulse, se desplegará el menú principal de la aplicación con sus opciones. Desde este menú se podrá acceder a cualquier lugar de la aplicación: Buscar Productos, Mis Productos, Subir Productos, Chats, Mapa de Productos, Mi perfil y Cerrar sesión.

3.3.2.4 Buscar productos

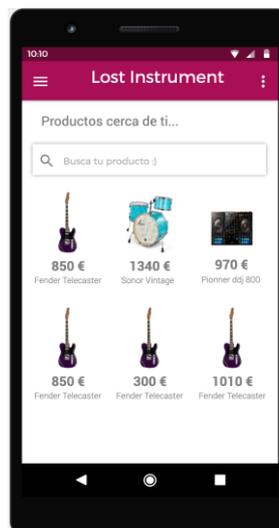


Ilustración 8. Prototipo: Búsqueda de productos

Será la primera pantalla que nos aparezca una vez realizado el login en la aplicación. Automáticamente la aplicación recogerá nuestra geolocalización y nos mostrará por pantalla los productos más cercanos a nosotros.



3.3.2.5 Mis Productos

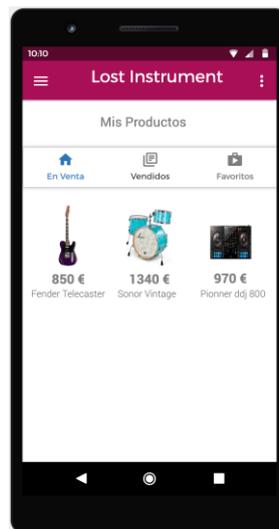


Ilustración 9. Prototipo: Pantalla Mis Productos

En esta pantalla se mostrará todos los productos que el usuario autenticado haya subido a la App. También nos mostrará los productos que se han marcado como favoritos, de tal manera que nos permita realizar un seguimiento.

3.3.2.6 Ficha de Producto



Ilustración 10. Prototipo: Pantalla ficha producto

Si el usuario pincha en cualquier producto, la aplicación le mostrará la ficha con las características de dicho producto: fotos, nombre, precio, descripción, etc.



3.3.2.7 Subir Productos

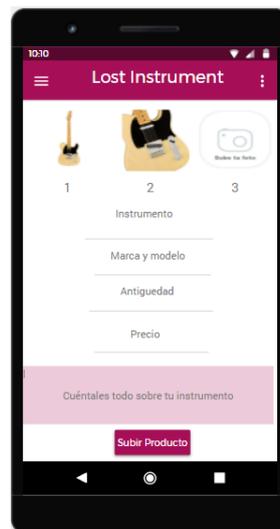


Ilustración 11. Prototipo: Pantalla subir producto

Desde esta pantalla el usuario autenticado podrá subir los productos que desee vender. Para ello, deberá subir unas imágenes del producto, y rellenar un formulario en el cual se indiquen ciertas características: nombre del instrumento, marca, modelo, antigüedad y el precio. También deberá escribir un comentario detallando el producto.

3.3.2.8 Pantalla de Conversaciones y Ventana de chat

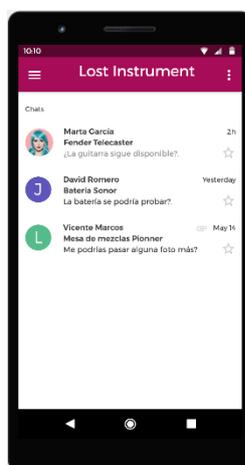


Ilustración 12. Prototipo: Pantalla Listado de Conversaciones.

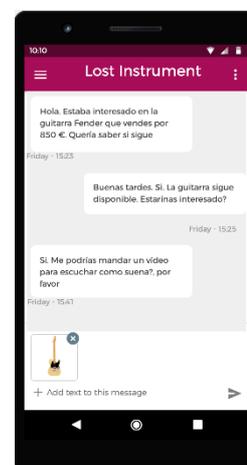


Ilustración 13. Prototipo: Pantalla chat

En esta pantalla se podrán acceder a los chats que se tengan en curso. Para que se habilite de primeras el chat, se deberá tener una conversación iniciada.



3.3.2.9 Mapa de Productos



Ilustración 14. Prototipo: Pantalla de mapa de productos

En esta pantalla se mostrará al usuario la ubicación de los productos subidos a la aplicación.

3.3.2.10 Mi Perfil

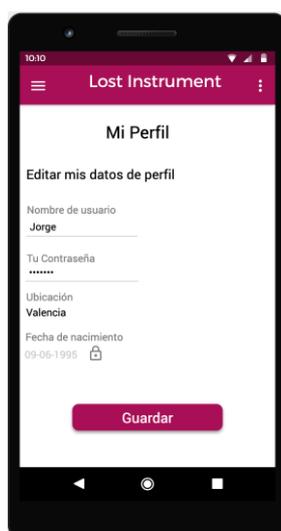


Ilustración 15. Prototipo: Pantalla Mi perfil

Desde esta pantalla, el usuario autenticado podrá modificar sus datos.

3.3.2.11 Cerrar Sesión



Ilustración 16.. Prototipo: Botón Cerrar sesión

Botón para realizar una salida ordenada y segura de nuestra app.



3.4 Evaluación

El objetivo de la última fase es la planificar la evaluación del prototipo. El proceso de DCU es un proceso iterativo y, por tanto, iremos evaluando los diseños y corrigiendo los errores de manera iterativa en el transcurso del proyecto.

Las preguntas que haremos a los usuarios para obtener la información correspondiente al usuario serán las mismas preguntas que hicimos en la fase de análisis

3.4.1 Preguntas para los usuarios

Datos Personales

- Edad
- Sexo
- Localidad
- Frecuencia con que utiliza aplicaciones móviles

3.4.2 Tareas que los usuarios deberán realizar

Tarea 1: Registrarse, acceder a la APP y cerrar sesión.

Tarea 2: Logarse y buscar el instrumento musical más cercano a nuestra ubicación y abrir un chat al usuario vendedor.

Tarea 3: Subir un producto nuevo rellenando toda la información

3.4.3 Preguntas referentes a las tareas

Se realizaron una serie de preguntas a los usuarios con el objetivo de medir el grado de satisfacción de los usuarios con la app. Las preguntas que se hicieron fueron:

- ¿Valora del 1 al 10 la facilidad de manejo de la aplicación?
- ¿Qué tarea le ha resultado más complicada?
- ¿Qué es lo que menos le ha gustado de la aplicación?
- ¿Considera atractiva la aplicación visualmente?
- ¿Cómo piensa que será la acogida de la app por parte de los usuarios de Smartphone?

A continuación, se presenta el resultado de tres test realizados:

TEST 1

Datos Personales

- **Edad:** 25
- **Sexo:** Femenino
- **Localidad:** Madrid
- **Trabajo:** Arquitecta

Preguntas

1. ¿Valora del 1 al 10 la facilidad de manejo de la aplicación? 9. Me ha parecido bastante sencilla e intuitiva.



2. **¿Qué tarea le ha resultado más complicada?** Ninguna. Las tres tareas las he realizado sin ningún problema.
3. **¿Qué es lo que menos le ha gustado de la aplicación?** Echo de menos alguna opción en el caso de que quieras comprar un producto que no esté en tu localidad. Si el envío lo gestionara la app, sería perfecto.
4. **¿Considera atractiva la aplicación visualmente?** Bastante. Es muy bonita, aunque se podría mejorar.
5. **¿Cómo piensa que será la acogida de la app por parte de los usuarios de Smartphone?** Creo que entre los músicos podría tener bastante éxito.

TEST 2

Datos Personales

- **Edad:** 51
- **Sexo:** Masculino
- **Localidad:** Alcobendas
- **Trabajo:** Conductor de autobuses

Preguntas

1. **¿Valora del 1 al 10 la facilidad de manejo de la aplicación?** 7. Me ha parecido sencillo el manejo, aunque hay alguna pantalla que al principio me ha confundido un poco.
2. **¿Qué tarea le ha resultado más complicada?** Subir un producto nuevo. No sabía muy bien como se subían las fotos que tenía en el teléfono ya hechas, pero al final lo he conseguido.
3. **¿Qué es lo que menos le ha gustado de la aplicación?** Los colores. No me gusta mucho el colorido.
4. **¿Considera atractiva la aplicación visualmente?** Si, pero hubiera puesto unos colores más sobrios. Un azul oscuro o negro y blanco.
5. **¿Cómo piensa que será la acogida de la app por parte de los usuarios de Smartphone?** Creo que entre la gente más joven gustará bastante.

TEST 3

Datos Personales

- **Edad:** 38
- **Sexo:** Masculino
- **Localidad:** Valladolid
- **Trabajo:** Camarero

Preguntas

1. **¿Valora del 1 al 10 la facilidad de manejo de la aplicación?** 9. La aplicación es muy sencilla y no tiene muchas ventanas que puedan confundir al usuario. Lo justo y lo necesario.
2. **¿Qué tarea le ha resultado más complicada?** Ninguna.
3. **¿Qué es lo que menos le ha gustado de la aplicación?** Realmente no hay nada que no me haya gustado, pero creo que la aplicación tiene mucho margen de mejora. Se puede explotar mucho más.
4. **¿Considera atractiva la aplicación visualmente?** Si. Muy vistosa y sencilla. Me ha gustado.
5. **¿Cómo piensa que será la acogida de la app por parte de los usuarios de Smartphone?** Pienso que puede ser bueno, aunque uno nunca sabe. A veces las cosas nuevas suelen generar desconfianza al principio hasta que se vayan apuntando más usuarios y corra la voz. La paciencia será la clave.



3.5 Definición de los casos de uso

A continuación, detallaremos el diagrama general de los casos de uso. Estos casos de uso servirán para establecer las funcionalidades de la aplicación. Con el siguiente diagrama, podremos identificar de forma gráfica actores y el flujo de la aplicación: [11]

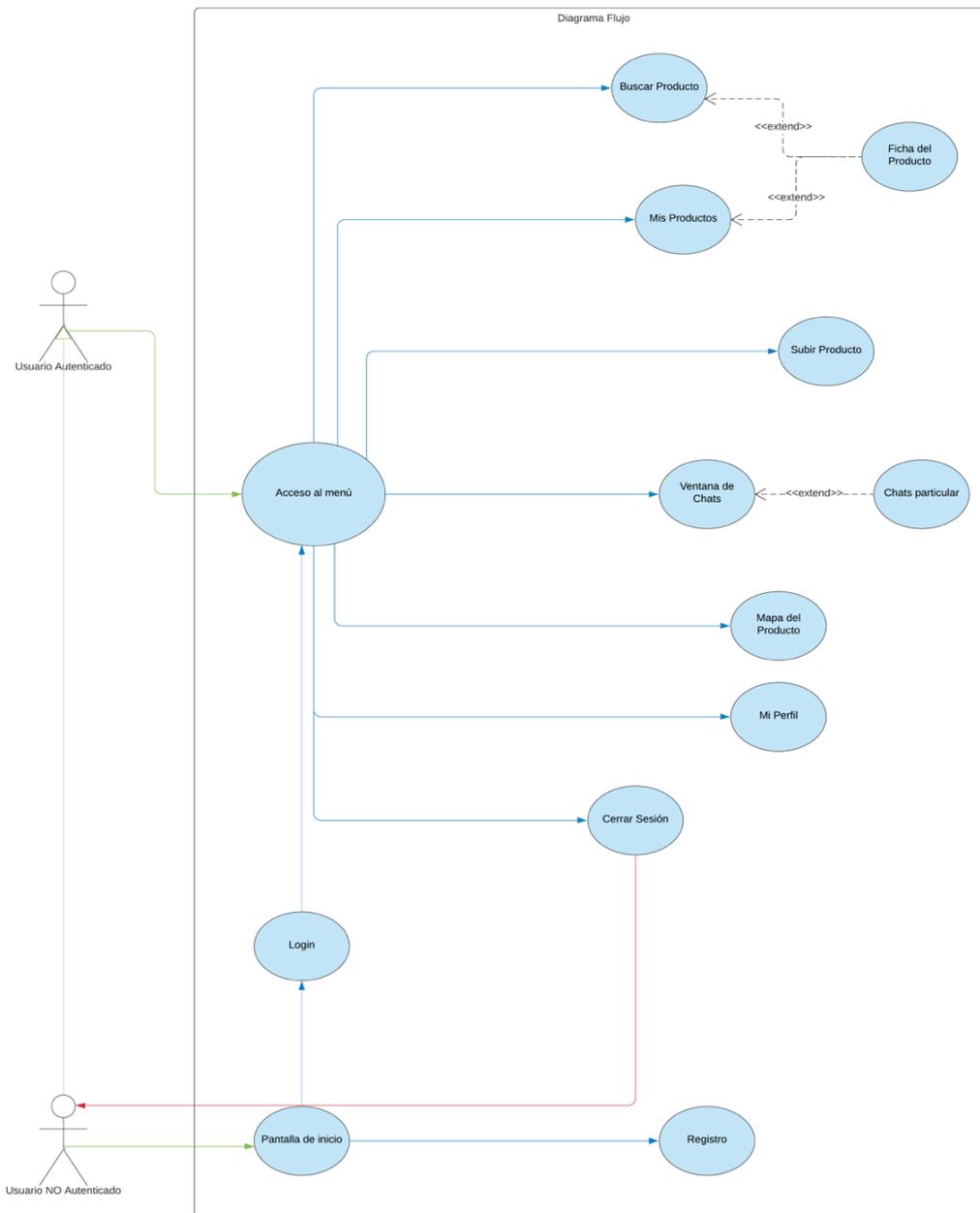


Ilustración 17. Prototipo: Diagrama general de Casos de Uso

3.5.1 Detalle de los casos de uso

En este apartado detallaremos el listado de los casos de uso de la aplicación donde se especifican los actores, precondiciones, flujo y postcondiciones:



Identificador	CU-001
Nombre	Registrar usuario
Prioridad	Alta
Descripción	El caso de uso consiste en el registro de un usuario en el sistema, necesario para poder usar la aplicación.
Actores	Cliente de la app
Precondiciones	El usuario debe tener al menos un correo electrónico para poder registrarse en la aplicación.
Iniciado por	Usuario no registrado
Flujo	Paso 1: Abrir la aplicación. Paso 2: En la pantalla de inicio pulsar sobre “Registro”. Paso 3: Rellenar el formulario que se presenta en pantalla. Paso 4: Una vez relleno todos los campos, pulsar “Registrarme”.
Postcondiciones	Se registra el usuario en la aplicación
Notas	Si el usuario ya se encuentra creado deberá dar un aviso y no creará el usuario.

Tabla 11. CU-001- Registrar Usuario

Identificador	CU-002
Nombre	Login en la app
Prioridad	Alta
Descripción	El caso de uso consiste en el login del usuario en la aplicación, para poder acceder.
Actores	Cliente de la app
Precondiciones	El usuario debe de estar registrado en la app (CU-001)
Iniciado por	Usuario registrado no autenticado.
Flujo	Paso 1: El usuario accede a la pantalla de inicio de la aplicación. Paso 2: El usuario introduce sus credenciales de acceso. Pase 3: El usuario pulsa el botón “acceder”
Postcondiciones	El usuario accede a la aplicación.
Notas	Si el usuario no se autentica de manera correcta, la app debería mostrar error en la autenticación.

Tabla 12. CU-002- Login en la App

Identificador	CU-003
Nombre	Acceso al menú de la aplicación
Prioridad	Alta
Descripción	El caso de uso consiste en acceder al menú de la aplicación, que será el punto de partida donde se seleccionen las funcionalidades de la aplicación
Actores	Usuario de la app
Precondiciones	El usuario debe de estar dado de alto en el sistema (CU-001) El usuario debe de estar logado en el sistema (CU-002)
Iniciado por	Usuario autenticado en la aplicación.
Flujo	Paso 1: El usuario pulsa el icono del menú (tres rayas horizontales en la parte superior de la pantalla) Paso 2: Se despliega el menú con el listado de funcionalidades de la app.
Postcondiciones	El usuario podrá acceder al menú de la aplicación.
Notas	

Tabla 13. CU-003- Acceso al menú de la aplicación



Identificador	CU-004
Nombre	Buscar Producto
Prioridad	Alta
Descripción	El caso de uso consiste en realizar una búsqueda de un producto en la aplicación.
Actores	Usuario de la app
Precondiciones	El usuario debe de estar dado de alto en el sistema (CU-001) El usuario debe de estar logado en el sistema (CU-002)
Iniciado por	Usuario autenticado en la aplicación tras haber desplegado el menú.
Flujo	Paso 1: El usuario accede al menú y selecciona “Productos cerca de mi”. Paso 2: Una vez que estamos en la pantalla deseada, accedemos al botón “Productos Cerca” situado en la parte inferior de la pantalla (botonera de tres pulsadores). Paso 3: Escribimos el nombre del producto que se desea buscar en el campo de texto preparado para ello y pulsamos en la lupa. Paso 4: La aplicación devolverá resultados solo si existen coincidencias en la base de datos. Paso 5: En caso afirmativo, podremos pulsar sobre el producto interesado y acceder a la ficha del producto
Postcondiciones	La aplicación nos devolverá el resultado de la búsqueda
Notas	El resultado de la búsqueda se devolverá vacío en el caso de que no existan coincidencias en la base de datos.

Tabla 14. CU-004- Buscar Producto

Identificador	CU-005
Nombre	Acceder a la ficha del producto
Prioridad	Alta
Descripción	El caso de uso consiste en acceder por parte del usuario a la ficha de un producto donde se podrá visualizar sus características.
Actores	Usuario de la App
Precondiciones	El usuario debe de estar dado de alto en el sistema (CU-001) El usuario debe de estar logado en el sistema (CU-002) El usuario debe de haber buscado el producto (CU-003) o haber accedido mediante la sección “Mis Productos” (CU-006).
Iniciado por	Usuario autenticado en la aplicación tras haber desplegado el menú y haber elegido una de las opciones donde se permite acceder a visitar los productos.
Flujo	Paso 1: El usuario busca el producto y pulsa sobre él, para poder acceder a su ficha. Paso 2: Se presenta al usuario la ficha del producto seleccionado con sus imágenes y características.
Postcondiciones	El usuario podrá visualizar la ficha del producto con todas sus características.
Notas	

Tabla 15. CU-005- Acceder a la ficha del producto



Identificador	CU-006
Nombre	Acceder a mis productos
Prioridad	Alta
Descripción	El caso de uso consiste en acceder por parte del usuario a una pantalla donde queden reflejados todos los productos que vende.
Actores	Usuario de la App
Precondiciones	El usuario debe de estar dado de alto en el sistema (CU-001) El usuario debe de estar logado en el sistema (CU-002)
Iniciado por	Usuario autenticado en la aplicación tras haber desplegado el menú para acceder a “mis productos”.
Flujo	Paso 1: El usuario pulsa en el menú y accede a la sección “Mis Productos” Camino alternativo: Paso 2: El usuario accede al menú y selecciona “Productos cerca de mí”. Paso 3: Una vez que estamos en la pantalla deseada, accedemos al botón “Mis Productos” situado en la parte inferior de la pantalla. Paso 4: La aplicación mostrará al usuario el listado de los productos que haya subido.
Postcondiciones	Se visualiza la pantalla de mis productos.
Notas	Si el usuario no ha comprado ni vendido ningún producto, la sección de mis productos aparecerá vacía.

Tabla 16. CU-006- Acceder a mis productos

Identificador	CU-007
Nombre	Subir producto
Prioridad	Media
Descripción	El caso de uso consiste en subir a la aplicación un producto por parte del usuario, para que lo pueda anunciar a los demás con el fin de venderlo.
Actores	Usuario de la app
Precondiciones	El usuario debe de estar dado de alto en el sistema (CU-001) El usuario debe de estar logado en el sistema (CU-002)
Iniciado por	Usuario autenticado en la aplicación tras haber desplegado el menú y haber accedido a la sección “Subir producto”.
Flujo	Paso 1: El usuario pulsa el menú y accede a la sección “Subir Producto”. Paso 2: El usuario rellena el formulario que se presenta en pantalla con los datos del producto, adjuntando varias imágenes del mismo. Paso 3: El usuario pulsa en “subir producto” y confirma la operación. Flujos alternativos: <ul style="list-style-type: none">• Si el usuario no introduce ninguna imagen• Si deja algún campo vacío No se subirá el producto hasta que se verifique que todos los datos introducidos son correctos
Postcondiciones	El usuario sube un producto que es visible para todos los usuarios de la app.
Notas	Si el usuario introduce algún dato erróneo, al pulsar el botón de “Subir Producto” la aplicación lanzará error indicando el campo erróneo.

Tabla 17. CU-007- Subir producto



Identificador	CU-008
Nombre	Visualizar chats
Prioridad	Media
Descripción	<p>El caso de uso consiste en la visualización por parte del usuario de todos sus chats en curso. Se podrá tener las siguientes conversaciones:</p> <ul style="list-style-type: none">• Conversación iniciada por otro usuario que se ha comunicado con el usuario autenticado mediante un producto subido.• El usuario autenticado inicia una conversación con otro usuario debido a que se encuentra interesado en un producto de otro usuario. <p>En la ventana conversaciones aparecerán los chats mezclados independientemente de que el usuario autenticado haya iniciado la conversación o no.</p>
Actores	Usuario de la app
Precondiciones	El usuario debe de estar dado de alto en el sistema (CU-001) El usuario debe de estar logado en el sistema (CU-002)
Iniciado por	Usuario autenticado en la aplicación tras haber desplegado el menú y haber accedido a la sección de “chats”.
Flujo	<p>Paso 1: El usuario pulsa el menú y accede a la sección “Conversaciones” Paso 2: La aplicación muestra al usuario por pantalla los chats que tiene en curso. Pase 3: Pulsando sobre la columna del chat deseado se podrá acceder a él. Paso 4: Recuperaremos por pantalla la conversación previa (si hubiera) y podremos comunicarnos con el usuario.</p>
Postcondiciones	El usuario visualiza los chats que tiene en curso.
Notas	Si no tenemos ningún chat en curso, no aparecerá ninguno y no dará error al cargar la pantalla de chats.

Tabla 18. CU-008- Visualizar Chats

Identificador	CU-009
Nombre	Visualizar el mapa de productos
Prioridad	Media
Descripción	El caso de uso consiste en mostrar al usuario la geolocalización de los productos de la aplicación.
Actores	Usuario de la app
Precondiciones	El usuario debe de estar dado de alto en el sistema (CU-001) El usuario debe de estar logado en el sistema (CU-002)
Iniciado por	Usuario autenticado en la aplicación tras haber desplegado el menú y haber accedido a la sección “Visualizar el mapa de productos”.
Flujo	<p>Paso 1: El usuario pulsa el menú principal y accede a la sección “Mapa de Productos”. Paso 2: Se muestra el mapa con una chincheta ubicada en el punto geográfico donde se haya registrado un producto. Paso 3: Si el usuario pulsa sobre la chincheta parecerá el nombre del producto para poder conocer el producto.</p>
Postcondiciones	Se visualiza un mapa en el cual se indican donde se geolocaliza el producto buscado.
Notas	

Tabla 19. CU-009- Visualizar mapa de productos



Identificador	CU-0010
Nombre	Editar mi perfil
Prioridad	Media
Descripción	El caso de uso consiste en permitir al usuario modificar los datos de su perfil.
Actores	Usuario de la app
Precondiciones	El usuario debe de estar dado de alto en el sistema (CU-001) El usuario debe de estar logado en el sistema (CU-002)
Iniciado por	Usuario autenticado en la aplicación tras haber desplegado el menú y haber accedido a la sección “Mi Perfil”
Flujo	Paso 1: El usuario pulsa el menú y accede a la sección “Mi Perfil” para editar algún dato. Paso 2: El sistema muestra al usuario sus datos personales y la posibilidad de modificarlos. Paso 3: El usuario actualiza los datos nuevos. Paso 4: El sistema informa que los cambios se han realizado con éxito.
Postcondiciones	Modificación del perfil de usuario realizada con éxito.
Notas	Si se produce un error al actualizar los datos, la aplicación lanzará un error al usuario.

Tabla 20. CU-0010- Editar mi perfil

Identificador	CU-0011
Nombre	Actualizar contraseña
Prioridad	Media
Descripción	El caso de uso consiste en permitir al usuario actualizar la contraseña
Actores	Usuario de la app
Precondiciones	El usuario debe de estar dado de alto en el sistema (CU-001) El usuario debe de estar logado en el sistema (CU-002)
Iniciado por	Usuario autenticado en la aplicación tras haber desplegado el menú y haber accedido a la sección “Mi Perfil”
Flujo	Paso 1: El usuario accede al menú y selecciona la sección “Mi Perfil” para editar los datos. Paso 2: El sistema muestra al usuario sus datos personales y la posibilidad de actualizar la contraseña. Paso 3: El usuario selecciona actualizar contraseña y automáticamente la aplicación le enviará un email, para que actualice la contraseña desde el enlace proporcionado.
Postcondiciones	Actualización de la contraseña realizada con éxito.
Notas	Para poder iniciar sesión deberá introducir la nueva contraseña, inhabilitando la anterior.

Tabla 21. CU-0011- Actualizar contraseña



Identificador	CU-0012
Nombre	Eliminar usuario de la aplicación
Prioridad	Media
Descripción	El caso de uso consiste en permitir al usuario darse de baja en el sistema
Actores	Usuario de la app
Precondiciones	El usuario debe de estar dado de alto en el sistema (CU-001) El usuario debe de estar logado en el sistema (CU-002)
Iniciado por	Usuario autenticado en la aplicación tras haber desplegado el menú y haber accedido a la sección “Mi Perfil”
Flujo	Paso 1: El usuario accede al menú y selecciona la sección “Mi Perfil” para editar los datos. Paso 2: El sistema muestra al usuario sus datos personales y la posibilidad de eliminar la cuenta, dándose de baja en el sistema. Paso 3: El usuario selecciona eliminar cuenta y automáticamente la aplicación le solicita confirmación, avisándole de que la acción no es reversible. Si el usuario confirma, la aplicación borra el usuario de la base de datos y sale de la aplicación regresando al menú de login.
Postcondiciones	El usuario no se podrá volver a autenticar con esa cuenta en la aplicación, hasta que no se vuelva a dar de alta.
Notas	

Tabla 22. CU-0012- Eliminar usuario de la aplicación

Identificador	CU-013
Nombre	Cerrar sesión
Prioridad	Alta
Descripción	El caso de uso consiste en permitir al usuario cerrar la sesión de una forma segura y ordenada
Actores	Usuario de la app
Precondiciones	El usuario debe de estar dado de alto en el sistema (CU-001) El usuario debe de estar logado en el sistema (CU-002)
Iniciado por	Usuario autenticado en la aplicación tras haber desplegado el menú y haber accedido a la sección “Cerrar sesión”
Flujo	Paso 1: El usuario pulsa el menú y dentro del menú selecciona “Cerrar sesión”. Paso 2: El sistema se cierra y redirige a la pantalla del login.
Postcondiciones	El usuario cierra correctamente su sesión.
Notas	

Tabla 23. CU-0013- Cerrar sesión

3.6 Diseño de la arquitectura

El objetivo de este apartado es el de definir la arquitectura del sistema, identificando las entidades que se representarán en la base de datos, las clases y objetos que se utilizarán para gestionar los diferentes procesos y la estructura de la aplicación, que servirá para realizar las peticiones al servidor.



3.6.1 Base de datos

Para poder desarrollar la aplicación, necesitaremos hacer uso de una base de datos donde poder almacenar y alojar todos los datos subidos por los usuarios. Para ello, se ha utilizado una herramienta que nos proporciona *Google* y que se integra de una manera perfecta con Android. Esta herramienta se llama *Firestore*.

3.6.1.1 Firestore [13]

Firestore es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada por Google en 2014. Es una plataforma ubicada en la nube, integrada con Google Cloud Platform, que usa un conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad, haciendo posible el crecimiento del número de usuarios y dando resultado también a la obtención de una mayor monetización.

Los desarrolladores obtienen una serie de ventajas al usar esta plataforma:

- Sincronización de forma fácil los datos de los proyectos sin tener que administrar conexiones o escribir lógica de sincronización compleja.
- Firestore usa un conjunto de herramientas multiplataforma: se integra fácilmente para plataformas web como en aplicaciones móviles. Es compatible con grandes plataformas, como IOS, Android, aplicaciones web, Unity y C++.
- Usa la infraestructura de Google y escala automáticamente para cualquier tipo de aplicación, desde las más pequeñas hasta las más potentes.
- Posibilidad de crear proyectos sin necesidad de un servidor: Las herramientas se incluyen en los SDK para los dispositivos móviles y web, por lo que no es necesario la creación de un servidor para el proyecto.

Firestore dota a sus usuarios de una gran documentación para crear aplicaciones usando esta plataforma. Aparte de esto, ofrece soporte gratuito mediante correo electrónico para todos sus usuarios, y además sus desarrolladores participan activamente en plataformas como Github y StackOverflow. Gracias a todas estas funcionalidades, cualquier desarrollador puede combinar y adaptar la plataforma a medida de sus necesidades.

Los servicios que presenta Firestore a sus usuarios son:

- **Desarrollo:** Firestore permite la creación de aplicaciones, minimizando el tiempo de optimización y desarrollo mediante diferentes funciones, entre las que destacan la detección de errores y el testeo, lo cual supone poder dar un salto de calidad a la aplicación. Las características más destacadas de Firestore son la posibilidad de poder almacenar toda la información en la nube y configurarla de manera distribuida.
- **Firestore Analytics:** es una aplicación gratuita que proporciona una visión profunda sobre el uso de la aplicación por parte de los usuarios.
- **Firestore Cloud Messaging:** Antiguamente conocido como Google Cloud Messaging (GCM), Firestore Cloud Messaging (FCM) es una plataforma para mensajes y notificaciones para Android, iOS, y aplicaciones web que actualmente puede ser usada de forma gratuita.



- **Firestore Auth:** Firestore Auth es un servicio que puede autenticar los usuarios utilizando únicamente código del lado del cliente. Incluye la autenticación mediante proveedores de inicio de sesión como Facebook, GitHub, Twitter, Google, Yahoo y Microsoft; así como los métodos clásicos de inicio de sesión mediante correo electrónico y contraseña. Además, incluye un sistema de administración del usuario por el cual los desarrolladores pueden habilitar la autenticación de usuarios con email y contraseña que se almacenarán en Firestore. Este servicio busca facilitar la creación de sistemas de autenticación, a la vez que mejora la incorporación, acceso y seguridad para los usuarios.
- **Firestore Database:** Firestore proporciona una base de datos **en tiempo real**, back-end y organizada en forma de árbol JSON. El servicio proporciona a los desarrolladores de aplicaciones una API que permite que la información de las aplicaciones sea sincronizada y almacenada en la nube de Firestore. La compañía habilita integración con aplicaciones Android, iOS, JavaScript, Java, etc. La sincronización en tiempo real de esta base de datos permite que los usuarios accedan a la información de sus datos desde cualquier dispositivo en tiempo real, compartiendo una instancia de Firestore Database, y cada vez que un usuario realice una modificación en esta, se almacena dicha información en la nube y se notifica simultáneamente al resto de dispositivos.

Una funcionalidad interesante de esta base de datos, es que, si un usuario realiza cambios y pierde a la vez su conexión a Internet, el SDK de la plataforma usa una caché local en el dispositivo donde guarda estos cambios; y una vez que vuelva a tener conexión, automáticamente se sincronizan los datos locales.

- **Firestore Storage:** Firestore Storage proporciona cargas y descargas seguras de archivos para aplicaciones Firestore, sin importar la calidad de la red. El desarrollador lo puede utilizar para almacenar imágenes, audio, vídeo, o cualquier otro contenido generado por el usuario. Firestore Storage se basa en el almacenamiento de Google Cloud Storage.
- **Firestore Cloud Firestore:** Cloud Firestore es un servicio de almacenamiento de datos derivado de Google Cloud Platform, adaptado a la plataforma de Firestore. Al igual que Firestore Database, es una base de datos NoSQL, aunque presenta diversas diferencias. Se organiza en forma de documentos agrupados en colecciones, y en ellos se pueden incluir tanto campos de diversos tipos (cadenas de texto, números, puntos geográficos, referencias a la propia base de datos, arrays, booleanos, marcas de tiempo, e incluso objetos propios) como otras subcolecciones. Entre sus limitaciones más destacadas encontramos la de no soportar las búsquedas de texto tipo "LIKE", eso es, buscar por subcadenas del texto almacenado, y la de no poder filtrar las búsquedas con condiciones que impliquen más de un campo, si no es por búsquedas por el texto exacto.

De todos estos servicios que nos aporta Firestore, en los siguientes apartados nos centraremos en explicar y detallar los que se han utilizado para administrar la base de datos de la aplicación:

- Firestore Cloud Firestore
- Firestore Database

3.6.1.2 **Firestore Cloud Firestore [14]**

Firestore Cloud Firestore proporciona una base de datos NoSQL flexible, escalable y en la nube con el fin de almacenar y sincronizar datos para la programación en el lado del cliente y del servidor.



Al igual que Firebase Realtime Database, mantiene los datos sincronizados entre apps cliente a través de agentes de escucha en tiempo real y ofrece asistencia sin conexión para dispositivos móviles y aplicaciones web, por lo que se pueden compilar apps con capacidad de respuesta que funcionan sin importar la latencia de la red ni la conectividad a Internet. Cloud Firestore también ofrece una integración sin interrupciones con otros productos de Firebase y Google Cloud Platform, incluido Cloud Functions.

Funcionamiento

A partir del modelo de datos NoSQL de Cloud Firestore, se almacenan los datos en documentos que contienen campos que se asignan a valores. Estos documentos se almacenan en colecciones, que son contenedores que se usan para organizar datos y compilar consultas. Los documentos admiten varios tipos de datos diferentes, desde strings y números simples, hasta objetos anidados complejos. También se pueden crear subcolecciones dentro de documentos y crear estructuras de datos jerárquicas que se ajustan según va creciendo la base de datos. El modelo de datos de Cloud Firestore admite cualquier estructura de datos.

Las consultas de Cloud Firestore son expresivas, eficientes y flexibles permitiendo crear consultas superficiales para recuperar datos en el nivel del documento, sin la necesidad de recuperar la colección completa ni las subcolecciones anidadas. Permite agregar criterios de orden, filtros y límites a las consultas o cursores para paginar los resultados. Para mantener actualizados los datos de la app sin tener que recuperar toda la base de datos cada vez que haya una actualización, se agregan agentes de escucha en tiempo real. Estos detectan y recuperan solo los cambios nuevos.



Ilustración 18. Cloud Firestore

3.6.1.3 Realtime Database [15]

Firebase Realtime Database es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Cuando se compilan apps multiplataforma con los SDK de iOS, Android y JavaScript, todos los clientes comparten una instancia de Realtime Database y reciben actualizaciones automáticamente con los datos más recientes.

Funcionamiento

Firebase Realtime Database permite el acceso seguro a la base de datos directamente desde el código del cliente. Los datos persisten de forma local. Además, incluso cuando no hay conexión, se siguen activando los eventos en tiempo real, lo que proporciona una experiencia adaptable al usuario final. Cuando el dispositivo vuelve a conectarse, Realtime Database sincroniza los cambios de los datos locales con las actualizaciones remotas que ocurrieron mientras el cliente estuvo sin conexión, lo que combina los conflictos de forma automática.



Proporciona un lenguaje flexible de reglas basadas en expresiones, llamado reglas de seguridad, para definir cómo se deberían estructurar los datos y en qué momento se pueden leer o escribir.

Realtime Database es una base de datos NoSQL y, como tal, tiene diferentes optimizaciones y funcionalidades en comparación con una base de datos relacional. La API de Realtime Database **está diseñada para permitir solo operaciones que se puedan ejecutar rápidamente**. Eso permite crear una excelente experiencia de tiempo real que puede servir a millones de usuarios sin afectar la capacidad de respuesta.

3.6.2 Diseño de la base de datos.

Para diseñar la base de datos, se ha realizado un diagrama entidad-relación. Este diagrama permite describir el tipo y las relaciones entre los datos. Se representan las entidades relevantes de una base de datos, así como sus interrelaciones. [11]

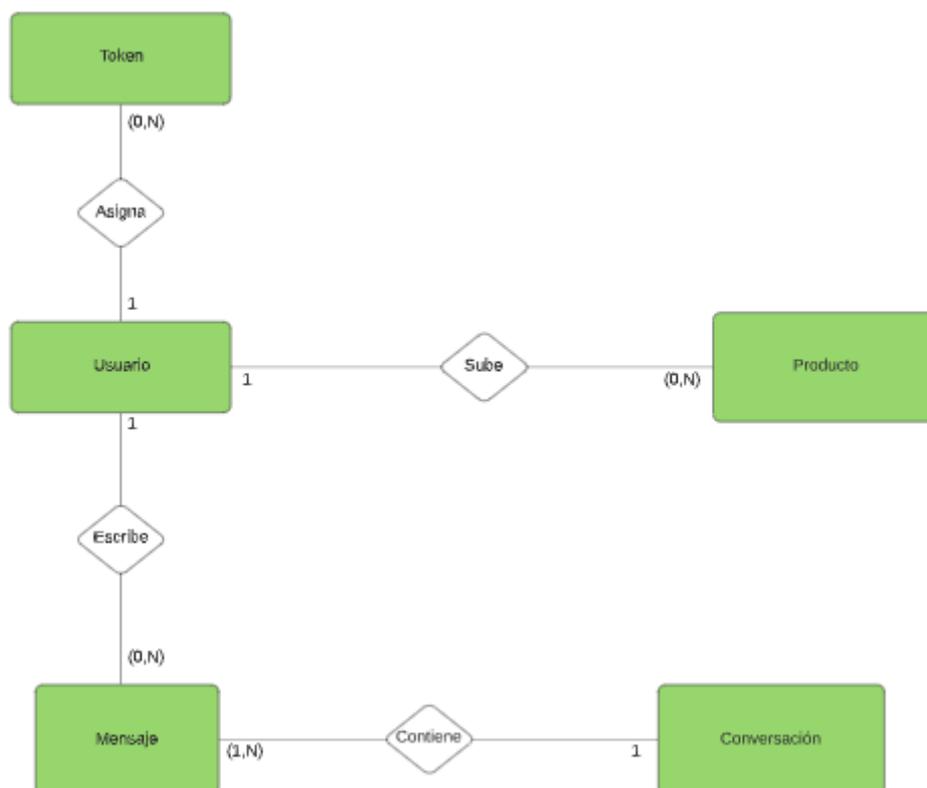


Ilustración 19. Diagrama Entidad-Relación



Relaciones entre las entidades:

Entidades	Relación
Token – Usuario	1 token estará asignado a un único usuario
	1 usuario podrá tener o cero o múltiples tokens (cero si está registrado en la aplicación, pero no se ha logado nunca, y múltiples si dispone de varios dispositivos logados con ese mismo usuario)
Usuario – Producto	1 usuario podrá subir ó 0 ó múltiples productos
	Ese producto con ese id, solo podrá ser subido una vez por un único usuario
Usuario – Mensaje	1 usuario podrá escribir ó 0 ó múltiples mensajes
	1 mensaje solo podrá ser escrito por un único usuario
Mensaje – Conversación	1 mensaje está contenido en una única conversación
	1 conversación podrá tener desde 1 hasta N mensajes

Tabla 24. Relaciones entre las entidades

El modelo relacional resultante es el siguiente

Tablas:

User
GUID_user
Date_birth
email
Productos_favoritos
Numero_id
password
User_name

Tabla 25. Tabla User

Conversacion
GUID_user_origen
GUID_user_destino
nombre_user_origen
nombre_user_destino
numero_relacion_user

Tabla 27. Tabla Conversación

Products
Key_product
UID_usuario
Nombre_instrumento
Marca
Antiguedad
Latitud
Longitud
Comentario
Precio
Nombre_imagenes
URL_imagenes

Tabla 26. Tabla Products

Chat (Mensaje)
fotoPerfil
guidOrigen
guidDestino
hora
mensaje
nombre
numeroRelacionAleatorio
type_mensaje

Tabla 28. Tabla Chat

Token (Notificacion)
tokenUser
userGUID

Tabla 29. Tabla Token

3.6.3 Arquitectura

A continuación, describiremos la arquitectura elegida para desarrollar nuestra arquitectura.



3.6.3.1 Modelo Vista Controlador [19]

El Modelo – Vista – Controlador (en adelante MVC) ha sido el paradigma elegido para la arquitectura de la aplicación.

MVC es un patrón de arquitectura de software que desacopla la vista del modelo para poder mejorar la reusabilidad y que las posibles modificaciones de las vistas tengan menor impacto en la lógica de negocio y en los datos. Este patrón se divide en 3 componentes: Modelo, Vista y Controlador.

- **Modelo:** Es la representación específica de la información, con la cual va a operar la aplicación (datos de la aplicación).
- **Vista:** La vista es la capa de la aplicación que ve el usuario en un formato adecuado para interactuar (interfaz gráfica).
- **Controlador:** El Controlador es la capa que controla todo lo que puede realizar nuestra aplicación. Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Funcionamiento

El controlador de nuestra aplicación recibirá la petición del usuario, el cual llamará al modelo y éste se encargará de pedir la información que necesitamos a la base de datos. La base de datos, devolverá el resultado que ha obtenido al modelo y éste, se lo devolverá al controlador, para que procese ese resultado y se lo envíe a la vista. La vista lo procesa y se lo envía al usuario.

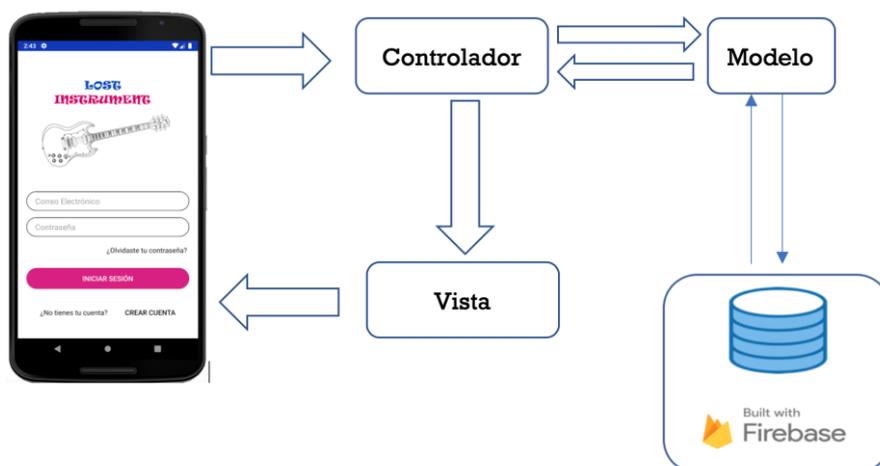


Tabla 30. Modelo Vista Controlador



Capítulo 4 - Implementación

4.1 Desarrollo

En esta etapa se construirá la aplicación móvil que cumpla con los objetivos que definimos en las etapas anteriores. Se codificará la aplicación aprovechando los frameworks, librerías y APIs más adecuados para la problemática a resolver.

4.1.1 Recursos utilizados en el desarrollo

Los recursos técnicos utilizados para el desarrollo de la aplicación han sido los siguientes:

Recursos	
Lenguaje de programación	Java
Entorno de desarrollo	Android Studio en su versión 3.6.2
Librerías requeridas	<ul style="list-style-type: none">• Librería de Firebase: Para poder utilizar Firebase. Nos proporciona una facilidad considerable a la hora de conectarse con Android Studio.• Servicios de Google Maps: Nos permite poder interaccionar nuestra aplicación con los mapas de Google.• Librería Glide y Picasso: Nos proporciona facilidad a la hora de gestionar las imágenes de la aplicación.• Librería uCrop: Al utilizar esta librería nos ha proporcionado la habilidad de poder cortar las imágenes al tamaño deseado antes de subirlas a la aplicación.• Librería EasySlider: Librería que nos permite de una forma rápida y fácil montar un Slider de imágenes. Esto le da un aspecto mucho más interactivo a la hora de visualizar los productos de nuestra aplicación.• Librería Volley: Para gestionar las notificaciones de la aplicación.
Entorno de ejecución y pruebas	Pruebas funcionales: Se han realizado pruebas funcionales mediante el emulador AVD Manager incorporado en el IDE de Android Studio.

Tabla 31. Recursos técnicos utilizados para el desarrollo

4.1.2 Puntos de interés del desarrollo

En este apartado explicaremos las funcionalidades utilizadas para la gestión de la autenticación de los usuarios, el almacenamiento del contenido y las notificaciones push de la aplicación. A continuación, se detallan estas funcionalidades aportadas por Firebase:



4.1.2.1 **Firebase Authentication [16]**

Firebase Authentication proporciona servicios de backend, SDK fáciles de usar y bibliotecas de IU ya elaboradas para autenticar a los usuarios en la app. Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federada populares (Google, Facebook y Twitter, entre otros muchos). Firebase Authentication se integra estrechamente en otros servicios de Firebase aprovechando los estándares de la industria como OAuth 2.0 y OpenID Connect. Como resultado se consigue una integración fácil con el backend de la aplicación.

Autenticación del SDK de Firebase

Para la autenticación de los usuarios de la aplicación se han utilizado dos métodos:

- **Autenticación basada en correo electrónico y contraseña:** Autentica a los usuarios con sus direcciones de correo electrónico y contraseñas. El SDK de Firebase Authentication nos proporciona métodos para poder crear y administrar usuarios que utilizan sus direcciones de correo electrónico y contraseñas para acceder (mediante un registro en la aplicación). Firebase Authentication también puede realizar el envío de correos electrónicos para restablecer la contraseña.
- **Integración con proveedores de identidad federada:** Autentica usuarios mediante la integración con proveedores de identidad federada. El SDK de Firebase Authentication proporciona métodos que permiten a los usuarios acceder con sus cuentas de Google, Facebook, Twitter y GitHub.

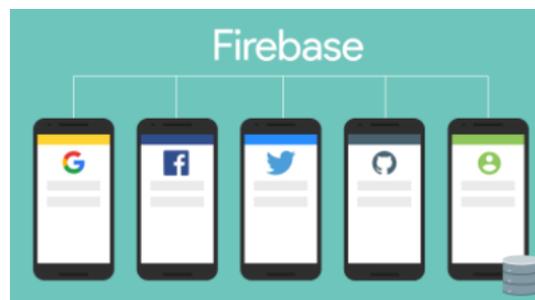


Ilustración 20. Firebase Authentication

Funcionamiento de la autenticación

Para que un usuario pueda acceder a la app, primero deberá obtener las credenciales de autenticación del usuario. Estas credenciales pueden ser la dirección de correo electrónico y la contraseña del usuario, o un token OAuth de un proveedor de identidad federada (Facebook, Google, etc.). A continuación, se envían esas credenciales al SDK de Firebase Authentication y los servicios de Firebase verificarán esas credenciales, mostrando una respuesta al cliente.

Cuando el acceso se realice correctamente, se podrá visualizar la información básica del perfil del usuario y controlar el acceso del usuario a los datos almacenados en otros productos de Firebase. Los usuarios autenticados podrán leer y escribir datos en Firebase Realtime Database y Cloud Storage (ambos utilizados en la aplicación). Podremos controlar el acceso de los usuarios modificando las reglas que nos proporciona Firebase.



A continuación, mostramos un ejemplo de cuentas registradas en la aplicación *Lost Instrument*:

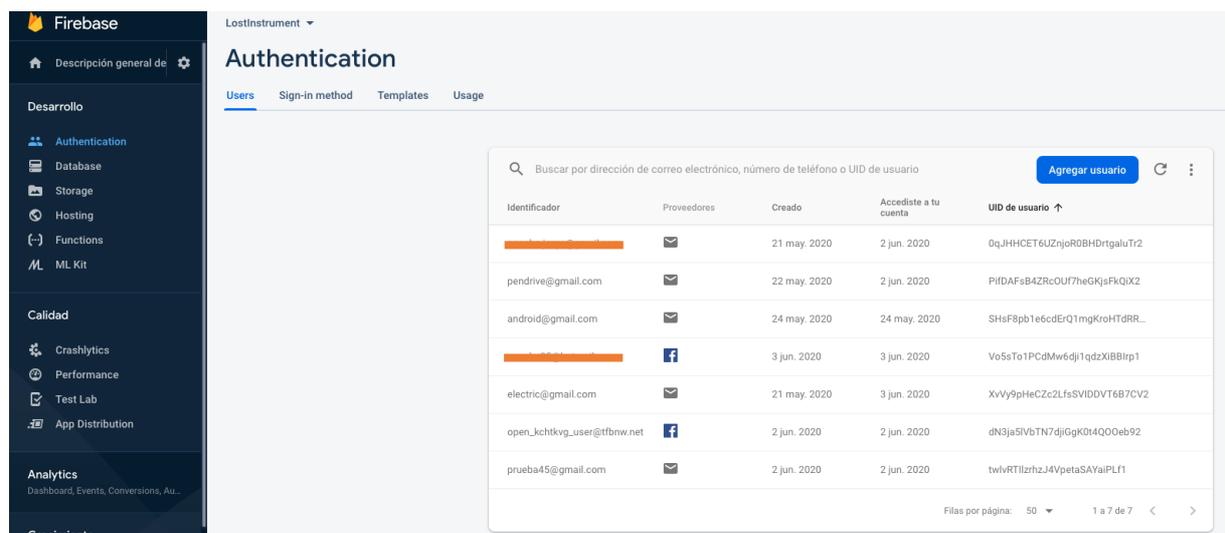


Ilustración 21. Ejemplo Authentication: Cuentas registradas en la app

4.1.2.2 Firebase Storage [17]

La aplicación utiliza Firebase Storage para el almacenamiento de las imágenes y las fotos que carguen los usuarios. Los SDK de Firebase para Cloud Storage agregan la seguridad de Google a las operaciones de carga y descarga de archivos para la aplicación, sin importar la calidad de la red. Se puede usar el SDK que nos facilita Firebase para almacenar imágenes, audio, video y otros tipos de contenido generado. A continuación, detallamos los puntos fuertes de Firebase Storage:

- **Operaciones robustas:** Los SDK de Firebase para Cloud Storage realizan las operaciones de carga y descarga sin importar la calidad de la red. Las cargas y descargas son robustas, lo que significa que se reinician en el punto en el que se interrumpieron para así ahorrar tiempo y ancho de banda a los usuarios.
- **Seguridad sólida:** Los SDK se integran con Firebase Authentication a fin de brindar autenticación intuitiva y sencilla para los programadores. Poseen un modelo de seguridad declarativa para permitir el acceso según el nombre de archivo, el tamaño, el tipo de contenido y otros metadatos.
- **Gran escalabilidad:** Está diseñada para escalar a exabytes si la aplicación se vuelve muy solicitada.

Funcionamiento

Cloud Storage almacena los archivos en un depósito de Google Cloud Storage y los hace accesibles a través de Firebase y Google Cloud. Esto permite tener la flexibilidad para subir y descargar archivos de clientes móviles a través de los SDK de Firebase y realizar procesamiento en el servidor. Cloud Storage se escala automáticamente, lo que significa que no es necesario migrar a ningún otro proveedor.



Los SDK de Firebase para Cloud Storage se integran perfectamente en Firebase Authentication con el fin de identificar a los usuarios y configurar controles de acceso para archivos individuales o grupos de archivos, de manera que se pueda hacer que los archivos sean públicos o privados según se desee.

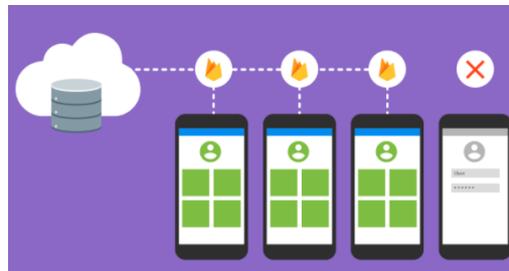


Ilustración 22. Firebase Storage

4.1.2.3 Firebase Cloud Messaging [18]

Firebase Cloud Messaging (FCM) es una solución de mensajería multiplataforma que permite enviar mensajes de forma segura y gratuita. Proporciona una conexión segura y de bajo consumo de batería entre el servidor y los dispositivos, lo que permite enviar y recibir mensajes y notificaciones. Utilizamos FCM, para notificar a un usuario que ha recibido un mensaje nuevo a través del canal de conexión confiable de FCM que consume poca batería.

Una implementación de FCM incluye dos componentes principales para enviar y recibir datos:

- Un entorno de confianza como Cloud Functions para Firebase
- Una app cliente de iOS, Android o Web (JavaScript) que reciba mensajes



Ilustración 23. Firebase Cloud Messaging

4.1.3 Estructura de la aplicación

En este punto mostraremos la estructura de archivos correspondiente al proyecto de Android implementado.

Como en todo proyecto de Android, se destacan dos carpetas:

- src : Carpeta que contiene todas las clases implementadas en java
- res: Contiene los recursos utilizados por la aplicación.



4.1.3.1 Carpeta de código fuente “src”

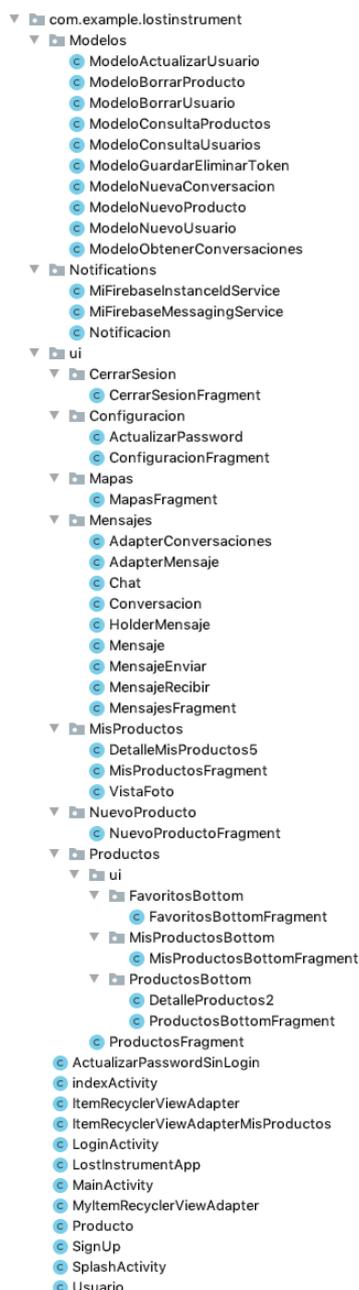


Ilustración 24. Código fuente carpeta SRC

A continuación, se detalla cada paquete que forma la estructura de la aplicación y cada clase que la compone:

Paquete Modelos

- **ModeloActualizarUsuario:** Modelo que se encargará de actualizar los datos del usuario en la base de datos
- **ModeloBorrarProducto:** Modelo que borrará de base de datos el producto que le pasemos como argumento
- **ModeloBorrarUsuario:** Modelo que borrará de base de datos el usuario que le pasemos como argumento



- **ModeloConsultaProductos:** Modelo que consultará en la base de datos el producto que le pasemos como argumento
- **ModeloConsultaUsuarios:** Modelo que consultará en la base de datos el usuario que le pasemos como argumento
- **ModeloGuardarEliminarToken:** Modelo que podrá guardar y/o eliminar el token asignado al usuario
- **ModeloNuevaConversacion:** Modelo que añadirá en la base de datos una conversación con los datos que le pasemos como argumento
- **ModeloNuevoProducto:** Modelo que añadirá en la base de datos un producto que le pasemos como argumento
- **ModeloNuevoUsuario:** Modelo que añadirá en la base de datos un nuevo usuario que le pasemos como argumento
- **ModeloObtenerConversaciones:** Modelo que consultará en la base de datos las conversaciones de un usuario que le pasemos como argumento

Paquete Notifications

- **MiFirebaseMessagingService:** Clase que se encargará de generar el nuevo token, guardarlo en la base de datos y recibir otros tokens.
- **Notificación:** Clase notificación con sus métodos.

Paquete Productos (Menú principal de la aplicación)

Paquete Cerrar Sesión

- **CerrarSesionFragment:** Fragmento que se encargará de realizar el cierre de la sesión y la salida de la aplicación.

Paquete Configuración

- **ActualizarPassword:** Método que enviará un correo al usuario para permitirle restablecer la password.
- **ConfiguracionFragment:** Fragmento que permitirá modificar los datos del usuario, actualizar su password y eliminar definitivamente la cuenta.

Paquete Mapas

- **MapasFragment:** Fragmento que nos mostrará en el mapa la geolocalización de todos los productos.

Paquete Mensajes

- **Conversación:** Clase que contiene los métodos de la pantalla de conversación del Chat.
- **AdapterConversaciones:** Clase que se encargará de mostrar las conversaciones en el fragment del menú Conversaciones.
- **AdapterMensaje:** Clase que mostrará un RecyclerView de mensajes.
- **HolderMensaje:** Esta clase controla los datos de cada mensaje. Primero instancia el layout con atributos java. Luego se crean métodos de get y de Set.
- **Mensaje:** Clase que gestiona los métodos y los constructores del objeto Mensaje.
- **MensajeRecibir:** Clase que gestiona la recepción de mensajes desde la base de datos y los carga en el recycler para mostrárselos al usuario.
- **MensajeEnviar:** Clase que gestiona el método del objeto Mensaje a Enviar, que se produce en cuanto el usuario pulsa en enviar el mensaje.
- **Chat:** Clase que gestionará el funcionamiento del chat.
- **MensajesFragment:** Fragmento que cargará todas las conversaciones que el usuario tenga en curso.



Paquete MisProductos

- **DetalleMisProductos:** Clase que gestiona la recepción de mensajes desde la base de datos y los carga en el recycler para mostrárselos al usuario.
- **MisProductosFragment:** Fragmento que se encargará de mostrarnos en un recyclerView los productos subidos por ese usuario.

Paquete Nuevo Producto

- **NuevoProductoFragment:** Fragmento que será el encargado de presentar un formulario al usuario para que introduzca los datos al usuario y cargarlos en la base de datos.

Paquete Productos

- **ProductosFragment:** Fragmento que cargará el Bottom Fragment de Productos Cerca de ti.

Paquete FavoritosBottom

- **FavoritosBottomFragment:** Fragmento que presentará los productos favoritos del usuario.

Paquete MisProductosBottom

- **MisProductosBottomFragment:** Fragmento que presentará en forma de cuadrícula los productos subidos del usuario.

Paquete ProductosBottom

- **ProductosBottomFragment:** Fragmento que buscará los productos que hay en la base de datos y se los mostrará al usuario ordenándolos por cercanía a su geolocalización. También permite realizar búsqueda de productos.
- **DetalleProductos:** Fragment que recibirá los detalles de un producto seleccionado y los mostrará en una vista.

ActualizarPasswordSinLogin: Esta actividad se encargará de recordar a un usuario su password desde fuera de la app, sin estar autenticado.

IndexActivity: Actividad Inicial de la aplicación. Presentará la opción de iniciar sesión mediante las credenciales de la app o desde Facebook.

ItemRecyclerViewAdapter: Clase que se encargará de crear el RecyclerView para los Items de los productos de los usuarios.

ItemRecyclerViewAdapterMisProductos: Clase que se encargará de crear el RecyclerView para los Items del botón Mis Productos con el GRID.

LoginActivity: Clase que permitirá a un usuario logarse en la aplicación con las credenciales de la App.

LostInstrumentApp: Clase que utiliza FB para hacer el login.

MainActivity: Clase Main. Será la primera pantalla una vez que el usuario se ha logado. Presentará un menú lateral y por defecto redirigirá al Fragment Mis Productos.

MyItemRecyclerViewAdapter: Clase que se encargará de crear el RecyclerView para los Items de Mis Productos.

Producto: Clase Producto con sus métodos.

SignUp: Clase que permitirá a un usuario registrarse en la App.

SplashActivity: Clase Splash que será la primera que se ejecute cuando se inicie la aplicación mostrando el logo de la APP.

Usuario: Clase Usuario con su constructor y sus métodos.



4.1.3.2 Carpeta de recursos res



Ilustración 25. Carpeta de recursos RES

Activity_index: Vista correspondiente a la actividad de Index.

Activity_login: Vista correspondiente a la actividad del login del usuario.

Activity_main: Vista correspondiente al main activity que se encargará de cargar el menú principal de la aplicación.

Activity_productos_with_bottom: Vista común del paquete productos.

Activity_sign_up: Vista correspondiente al registro del usuario en la aplicación.

Actualizacion_password: Vista para que el usuario pueda actualizar la contraseña.



App_bar_main: Vista para cargar el action bar de la aplicación.

Content_main: Vista para cargar el fragmento del main activity.

Detalle_productos2: Vista que cargará los detalles de un producto de otro usuario.

Detalles_misproductos: Vista que cargará los detalles de un producto subido por el usuario autenticado.

Fragment_cerrar_sesion: Vista que permite al usuario cerrar la sesión.

Fragment_configuracion: Vista correspondiente al menú configuración de la cuenta del usuario.

Fragment_favoritos_bottom: Vista correspondiente al botón productos favoritos.

Fragment_item: Vista en card view de un único producto del usuario en el RecyclerView.

Fragment_item2: Vista de un único producto de otro usuario en el RecyclerView.

Fragment_item_list: Vista de todos los productos que ha subido el usuario colocados dentro del RecyclerView.

Fragment_item_list2: Vista de todos los productos de otros usuarios que han subido a la aplicación, colocados dentro de un grid de recycler view.

Fragment_mapas: Vista donde el usuario puede ver la localización de todos los productos localizados en el mapa.

Fragment_mensajes: Vista de todas las conversaciones del usuario colocadas dentro de un recycler view.

Fragment_mis_productos_bottom: Vista del fragmento mis productos dentro del bottom Navigation Activity sobre el que se cargará el recycler view.

Fragment_mis_productos: Vista del fragmento mis productos sobre la que se cargará el recycler view.

Fragment_nuevo_producto: Vista correspondiente al formulario para generar un nuevo producto en la aplicación.

Fragment_productos_bottom: Vista del fragmento productos dentro del bottom Navigation Activity sobre el que se cargará el recycler view.

Fragment_productos: Vista del fragmento productos sobre la que se cargará el Bottom Navigation Activity.

Item_mensaje_conversacion: Vista que se represente a un único mensaje dentro del chat en el recycler view.

Item_mensaje: Vista que se corresponde con una única conversación dentro del recycler view.

Nav_header_main: Cabecera del menú principal.

Pantalla_conversacion: Vista en el que se cargarán todos los ítem_mensaje _conversacion colocados dentro de un recycler view.

4.1.4 Buenas prácticas de codificación

Durante la elaboración de la aplicación, se ha intentado seguir criterios básicos que tratan sobre la calidad del código. También se ha seguido la metodología del Diseño Centrado en el Usuario, procurando cumplir con los criterios tanto de seguridad como de usabilidad.

A continuación, indicamos las principales buenas prácticas de programación que se han utilizado:

Durante la construcción del producto, se han seguido criterios básicos relativos a la calidad del código como son: modularidad, abstracción y el uso de patrones de diseño. Además, se ha seguido una metodología de Diseño Centrado en el usuario, procurando cumplir de manera ágil, con los criterios de usabilidad de interfaces de usuario y seguridad del producto.



4.2 Pruebas

En esta etapa, tomando como referencia el análisis que se hizo anteriormente de los casos de uso, detallaremos las pruebas realizadas para determinar el funcionamiento correcto de la aplicación.

4.2.1 Entorno de pruebas

Para la ejecución de las pruebas, se ha utilizado el siguiente software:

- Herramienta Android Device Manager (AVD) de Android Studio. Este emulador está contenido dentro del IDE de Android Studio, permitiéndonos escoger y configurar entre múltiples dispositivos de la marca Google. El terminal de simulación elegido ha sido un Nexus 6 con las siguientes características:

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Nexus 6 API 28		1440 x 2560: 560dpi	28	Android 9.0 (Google APIs)	x86	3,5 GB	

Ilustración 26. Dispositivo Virtual utilizado

Verify Configuration

AVD Name: Nexus 6 API 28
AVD Id: Nexus_6_API_28

Nexus 6 5.96 1440x2560 560dpi

Pie Android 9.0 x86

Startup orientation: Portrait Landscape

Camera Front: Emulated Back: VirtualScene

Network Speed: Full Latency: None

Emulated Performance Graphics: Automatic
Boot option: Cold boot Quick boot Choose from snapshot
(no snapshots)

Multi-Core CPU 2

Memory and Storage
RAM: 1536 MB
VM heap: 384 MB
Internal Storage: 800 MB
SD card: Studio-managed 512 MB

Ilustración 27. Características dispositivo virtual utilizado



4.2.2 Pruebas Funcionales

Se han realizado pruebas de integración sobre el simulador de la aplicación. El objetivo de estas pruebas es el de probar las funcionalidades básicas imprescindibles para considerar que ha finalizado con éxito cada iteración. El resultado de estas pruebas debe ser satisfactorio para que el producto sea capaz de brindar al usuario unas funcionalidades básicas y una calidad mínima exigible.

También se han realizado pruebas de regresión, tras finalizar cada implementación, para comprobar que tanto las nuevas como las anteriores funcionalidades continuaban funcionando de manera correcta.

Definición de los casos de prueba

Código CP	Caso de Prueba	Funcionalidad	Pasos:
CP_1.1	Registro de un usuario desde la App	Registro de usuario	<p>Paso 1: Abrir la aplicación.</p> <p>Paso 2: En la pantalla de inicio pulsar sobre “Registro”.</p> <p>Paso 3: Rellenar el formulario que se presenta en pantalla.</p> <p>Paso 4: Una vez completado todos los campos, pulsar “Registrarme”.</p> <p>Flujo alternativo (opciones):</p> <ul style="list-style-type: none">• Email ya registrado en la base de datos o con formato incorrecto.• No introducir ninguno de los campos obligatorios• La edad debe tener el formato correcto y el usuario debe tener menos de 100 años. <p>No se registrará un usuario hasta que se verifique que todos los datos introducidos son correctos</p>
CP_2.1	Acceso a la aplicación con un usuario de la App	Login en la App	<p>Paso1: El usuario accede a la pantalla de inicio de la aplicación.</p> <p>Paso 2: El usuario introduce sus credenciales de acceso.</p> <p>Pase 3: El usuario pulsa el botón “acceder”</p> <p>Flujo alternativo:</p> <p>Paso1: El usuario accede a la pantalla de inicio de la aplicación.</p> <p>Paso 2: El usuario introduce sus credenciales de acceso incorrectos.</p> <p>Paso 3: La aplicación muestra el error.</p>
CP_2.2	Acceso a la aplicación probando el login de FB	Login en la App	<p>Paso1: El usuario accede a la pantalla de index de la aplicación y pulsa “Acceder mediante FB”.</p> <p>Paso 2: La aplicación muestra una ventana de FB donde el usuario introduce sus credenciales de acceso de FB</p> <p>Pase 3: El usuario pulsa el botón “acceder”.</p> <p>Flujo alternativo:</p> <p>Paso1: El usuario accede a la pantalla de index de la aplicación y pulsa “Acceder mediante FB”.</p>



			<p>Paso 2: La aplicación muestra una ventana de FB donde el usuario introduce sus credenciales de acceso de FB de forma incorrecta.</p> <p>Paso 3: La aplicación muestra el error.</p>
CP_3.1	Acceder al menú principal de la aplicación	Acceso al menú de la aplicación	<p>Paso 1: El usuario pulsa el icono del menú (tres rayas horizontales en la parte superior de la pantalla)</p> <p>Paso 2: Se despliega el menú con el listado de funcionalidades de la app.</p>
CP_4.1	Buscar un producto en concreto	Buscar producto	<p>Paso 1: El usuario accede al menú y selecciona “Productos cerca de mi”.</p> <p>Paso 2: Una vez que estamos en la pantalla deseada, accedemos al botón “Productos Cerca” situado en la parte inferior de la pantalla (botón de 3 pulsadores).</p> <p>Paso 3: Escribimos el nombre del producto en el campo de texto preparado para ello y pulsamos en la lupa.</p> <p>Paso 3: La aplicación devolverá resultados solo si existen coincidencias en la base de datos.</p>
CP_4.2	Acceso a los detalles de un producto buscado	Buscar Producto Acceso a la ficha del producto	<p>Paso 1: El usuario busca el producto y si hay resultados, pulsa sobre él, para poder acceder a su ficha.</p> <p>Paso 2: Se presenta al usuario la ficha del producto seleccionado con sus imágenes y características.</p>
CP_4.3	Acceso a los detalles de un producto cualquiera (sin búsqueda previa)	Acceso a la ficha del producto	<p>Paso 1: El usuario accede al menú y selecciona “Productos cerca de mi”.</p> <p>Paso 2: Una vez que estamos en la pantalla deseada, accedemos al botón “Productos Cerca” situado en la parte inferior de la pantalla.</p> <p>Paso 3: El usuario selecciona un producto de los primeros que le aparecen (ordenados por cercanía).</p> <p>Paso 4: Se presenta al usuario la ficha del producto seleccionado con sus imágenes y características.</p>
CP_5.1	Acceso al listado de Mis Productos	Acceder a Mis Productos	<p>Paso 1: Según accedamos a la aplicación la primera pantalla que se mostrará al usuario será el listado con los productos subidos</p> <p>Flujo alternativo:</p> <p>Paso 1: El usuario accede al menú y selecciona “Productos cerca de mi”.</p> <p>Paso 2: Una vez que estamos en la pantalla deseada, accedemos al botón “Mis Productos” situado en la parte inferior de la pantalla.</p> <p>Paso 3: La aplicación mostrará al usuario un listado de los productos que haya subido en forma de grid.</p>
CP_6.1	Subir un producto a la aplicación	Subir un producto	<p>Paso1: El usuario pulsa el menú y accede a la sección “Subir Producto”.</p> <p>Paso 2: El usuario rellena el formulario que se presenta en pantalla con los datos del producto, adjuntando varias fotos del producto.</p> <p>Paso 3: El usuario pulsa en “subir producto” y confirma la operación.</p> <p>Flujos alternativos:</p> <ul style="list-style-type: none">• Si el usuario no introduce ninguna imagen• Si deja algún campo vacío <p>No se subirá el producto hasta que se verifique que todos los datos introducidos son correctos</p>



CP_7.1	Abrir Chat a usuario	Visualizar Chat	<p>Paso 1: El usuario accede a la ficha de un producto (CP_4.3)</p> <p>Paso 2: El usuario se dirige a la parte inferior de la ventana, hasta encontrar el botón de color verde “Abrir Chat” y pulsarlo.</p> <p>Paso 3: Se abrirá una venta de Chat, la cual permitirá al usuario comunicarse con el otro usuario.</p>
CP_7.2	Visualizar Chats de usuario en curso	Visualizar Chat	<p>Paso 1: El usuario pulsa el menú y accede a la sección “Chats”</p> <p>Paso 2: La aplicación muestra al usuario por pantalla los chats que tenga en curso.</p> <p>Pase 3: Pulsando sobre la columna del chat deseada podremos acceder a él.</p> <p>Paso 4: Recuperaremos por pantalla la conversación previa (si hubiera) y podremos comunicarnos con el usuario.</p>
CP_8.1	Visualizar el mapa donde se ubican los productos	Visualizar el mapa de productos	<p>Paso 1: El usuario pulsa el menú y accede a la sección “Mapa de Productos”.</p> <p>Paso 2: Se muestra el mapa con una chincheta ubicada en el punto geográfico donde se haya registrado el producto.</p> <p>Paso 3: Si el usuario pulsa sobre la chincheta parecerá el nombre del producto.</p>
CP_9.1	Editar los datos del perfil	Editar Mi Perfil	<p>Paso 1: El usuario accede al menú y selecciona la sección “Mi Perfil” para editar los datos.</p> <p>Paso 2: El sistema muestra al usuario sus datos personales y la posibilidad de modificarlos.</p> <p>Paso 3: El usuario guarda los datos nuevos.</p> <p>Paso 4: El sistema pide confirmación e informa que los cambios se han realizado con éxito.</p> <p>Flujo alternativo:</p> <ul style="list-style-type: none">• El email no se puede modificar
CP_9.2	Actualizar la contraseña del usuario	Editar Mi Perfil	<p>Paso 1: El usuario accede al menú y selecciona la sección “Mi Perfil” para editar los datos.</p> <p>Paso 2: El sistema muestra al usuario sus datos personales y la posibilidad de actualizar la contraseña.</p> <p>Paso 3: El usuario selecciona actualizar contraseña y automáticamente la aplicación enviará un email al usuario, para que actualice la contraseña desde el enlace proporcionado.</p>
CP_9.3	Eliminar usuario de la aplicación	Eliminar usuario	<p>Paso 1: El usuario accede al menú y selecciona la sección “Mi Perfil” para editar los datos.</p> <p>Paso 2: El sistema muestra al usuario sus datos personales y la posibilidad de eliminar la cuenta, dándose de baja en el sistema.</p> <p>Paso 3: El usuario selecciona eliminar cuenta y automáticamente la aplicación le pide la confirmación y le avisa de que la acción no es reversible. Si el usuario confirma, la aplicación borra el usuario de la base de datos y sale de la aplicación regresando al menú de login.</p>
CP_10.1	Cerrar sesión	Cerrar Sesión	<p>Paso1: El usuario pulsa “Cerrar sesión” desde el menú principal.</p> <p>Paso 2: El sistema se cierra y redirige a la pantalla del login.</p>

Tabla 32. Definición de los casos de prueba



Resultado de la ejecución de los casos de prueba

Código CP	Caso de Prueba	Resultado	Detalles
CP_1.1	Registro de un usuario desde la App	OK	Se crea un usuario nuevo en la base de datos, con los datos introducidos en el formulario de registro
CP_2.1	Acceso a la aplicación con un usuario de la App	OK	Se consigue acceder a la aplicación con los datos de un usuario registrado
CP_2.2	Acceso a la aplicación probando el login de FB	OK	Se consigue acceder a la aplicación con los datos de la cuenta de FB.
CP_3.1	Acceder al menú principal de la aplicación	OK	Se muestra el botón que al pulsarlo se despliega el menú principal con las funcionalidades de la APP.
CP_4.1	Buscar un producto en concreto	OK	Se consigue encontrar un producto que coincide con la descripción introducida
CP_4.2	Acceso a los detalles de un producto buscado	OK	La aplicación nos muestra toda la información del producto seleccionado
CP_4.3	Acceso a los detalles de un producto cualquiera (sin búsqueda previa)	OK	La aplicación nos muestra toda la información del producto seleccionado
CP_5.1	Acceso al listado de Mis Productos	OK	La aplicación nos muestra el listado de los productos subidos por el usuario autenticado.
CP_6.1	Subir un producto a la aplicación	OK	Se crea un producto nuevo en la base de datos, que corresponde con los datos introducidos en el formulario de creación.
CP_7.1	Abrir Chat a usuario	OK	Se visualiza el chat del usuario y se consigue enviar mensajes y fotos al usuario.
CP_7.2	Visualizar Chats de usuario en curso	OK	La aplicación nos muestra todas las conversaciones en curso del usuario.
CP_8.1	Visualizar el mapa donde se ubican los productos	OK	Se muestra el mapa con los productos ubicados en forma de chincheta de Google.
CP_9.1	Editar los datos del perfil	OK	Se muestra un mensaje confirmando los nuevos datos. Si volvemos a entrar en el perfil, se comprueba que los datos se han actualizado de forma correcta.
CP_9.2	Actualizar la contraseña del usuario	OK	Se muestra la confirmación de que la contraseña se ha actualizado. Para comprobarlo, iniciamos sesión con la nueva contraseña, permitiéndonos acceder.
CP_9.3	Eliminar usuario de la aplicación	OK	Se muestra un mensaje confirmando la eliminación ya continuación la aplicación nos redirige a la venta de login. Si intentamos volver a entrar con el usuario eliminado, nos indicará que no existe en la base de datos.
CP_10.1	Cerrar sesión	OK	La aplicación nos redirige a la ventana de Login. Para volverse a logar deberemos introducir el usuario y la contraseña.

Tabla 33. Resultado de los casos de prueba



Capítulo 5 – Conclusiones

5.1 Conclusiones y líneas futuras de trabajo

5.1.1 Resultado final

El resultado final del proyecto ha sido la realización del análisis, diseño e implementación de una aplicación de Compra-Venta de productos musicales, que proporcionará a los usuarios la posibilidad de poder comprar y vender productos musicales de segunda mano, así como la comunicación entre ellos para dicho fin. El único requisito indispensable que deberán cumplir los usuarios para interactuar con la aplicación será el de disponer de un dispositivo Android con una versión de sistema operativo superior a 5.0 Lollipop, y una conexión a Internet.

La experiencia del desarrollo de la aplicación en Android ha sido satisfactoria tanto en tiempo como en funcionalidades. Para poder cubrir ciertas necesidades de la aplicación, he necesitado acceder a fuentes de información externas. La búsqueda de esta información ha resultado sencilla debido a que existen multitud de información en la red donde dan solución a cientos de problemas que se pueden plantear durante el desarrollo. También hay multitud de foros de programación, y vídeos colgados en las plataformas de reproducción de contenido donde he podido obtener respuesta a todos los problemas y dudas que han ido surgiendo.

Durante el desarrollo del proyecto, los requisitos iniciales definidos en las primeras fases, se han ido afinando y adaptándose a las nuevas necesidades que iban surgiendo durante el desarrollo, aunque en líneas generales ha variado muy sutilmente lo obtenido con lo establecido en la fase inicial.

En lo referente a las dificultades encontradas durante el desarrollo de la aplicación, estas, aparecieron sobre todo a la hora de implementar el chat de usuario. Era un reto nuevo, ya que desconocía como debía realizarlo. Después de buscar información en la red, visualizar múltiples vídeos y leer algún que otro ejemplo, pude implementar correctamente el chat de usuario. La utilización de Firebase y su herramienta RealTime, fue un acierto, debido a que hicieron que se simplificara bastante el desarrollo.

Después de superar los problemas que fueron apareciendo, algunos por mi desconocimiento y otras por la propia dificultad de la funcionalidad que se deseaba implementar, se ha podido comprobar las posibilidades que nos ofrece Android y Firebase. Su sencillez a la hora de vincularlos y como simplifican tareas que se podrían complicar bastante realizándolas con otras herramientas.

Por último, hacer una mención especial a la importancia de haber trabajado de forma correcta la fase de planificación, análisis y diseño guiadas por el diseño centrado en el usuario (DCU). Gracias a ello, se ha podido realizar un producto mucho más orientado a las necesidades del usuario final.



5.2 Logros alcanzados y conocimientos adquiridos

El logro principal alcanzado tras la finalización del TFM ha sido: conseguir trasladar las funcionalidades realizadas y pensadas en la teoría, a la aplicación desarrollada. De estas funcionalidades, algunas eran desconocidas totalmente al inicio del TFM. Para ello, se ha profundizado en el estudio del sistema operativo Android, descubriendo en él un gran potencial con respecto al desarrollo de aplicaciones. También se ha consolidado conocimientos inicialmente adquiridos en Java, y se han aprendido otros de cero, como la integración con Firebase. Aunque durante el desarrollo del Máster se ha cursado una asignatura dedicada al aprendizaje de Android, desde mi punto de vista, se veían las cosas básicas de este sistema operativo, quedando un camino bastante largo por recorrer para estar a la altura de una aplicación como la desarrollada en este TFM.

Al comenzar el proyecto, mis conocimientos de Android eran básicos, lo que implicó un aprendizaje intensivo, y un inicio del proyecto un tanto costoso y repleto de dificultades. Este handicap, se fue solucionando según fui recopilando información por la red y experiencia con Android. Según fueron pasando las semanas, cada vez me fui encontrando mucho más hábil y veloz a la hora de ir implementando las funcionalidades, lo cual supuso un empuje de motivación bastante grande para poder finalizar en tiempo este trabajo.

Se han usado librerías de terceros que han ayudado a realizar funciones que al principio desconocía como las iba a realizar. Por ejemplo, a la hora de seleccionar una foto para subirla a la aplicación, se necesitaba que la foto tuviera un tamaño específico, por lo que se usó una librería que nos ayudara a que el usuario durante el proceso de subida, pudiera cortar la imagen, dejándola preparada para la subida a los servidores de Firebase.

Al elaborar este proyecto, se ha adquirido un conocimiento mucho más serio y profesional de cómo se debe realizar un proyecto de tal magnitud cumpliendo todas las etapas hasta la finalización de la aplicación.

Por último, valorar como está dirigida y gestionada la asignatura Trabajo de Fin de Máster. Gracias a su organización en diferentes entregas de las PEC (plan de trabajo, diseño y arquitectura, implementación y entrega final), la memoria se ha ido realizando fase por fase, lo cual ha ayudado a una asimilación de los contenidos bastante importante y, sobre todo, se ha ido desarrollando durante estos meses, dejando un camino bastante asequible para la entrega final, sin tener que acudir a la presión y a los nervios por no llegar en tiempo a dicha entrega.

5.3 Posibles trabajos futuros

Aunque la aplicación posee una funcionalidad bastante extensa, necesita estar en constante evolución para no quedarse obsoleta. Los posibles trabajos futuros serán los de añadir funcionalidades que no han tenido cabida en esta versión de la aplicación, ya sea porque no entraron en la planificación inicial debido a que se alargaban en el tiempo o porque surgieron durante el desarrollo de la aplicación, siendo bastante costoso su cabida en dicha versión:

- Gestión del envío logístico de los productos. La aplicación se podría hacer cargo del envío mediante la gestión de un acuerdo con empresas de mensajería.



- Posibilidad de suscripción a alertas personalizadas. Cuando un usuario suba un producto que cumpla ciertos requisitos previamente guardados, enviar un aviso al usuario.
- Acuerdos con tiendas musicales como intermediarias para la venta del instrumento. El usuario deja el instrumento en la tienda física y ésta se encarga de la puesta a punto del mismo y de garantizar y certificar que el instrumento cumple ciertos criterios de calidad y no presenta desperfectos ocultos. La tienda lo podría poner a la venta llevándose una comisión por las gestiones.
- Mejorar la interfaz gráfica, haciendo uso de la metodología DCU. Encuestas para comprobar cómo está siendo la aceptación de la aplicación entre los usuarios y que se puede mejorar.



Capítulo 6 – Referencias

6.1 Referencias

Recursos teóricos

- [1] Wikipedia – Definición de Android
<https://es.wikipedia.org/wiki/Android>
- [2] Fragmentación Android
<https://techdroy.com/como-afecta-fragmentacion-android/>
- [3] Wikipedia – Definición de IOS
<https://es.wikipedia.org/wiki/IOS>
- [4] Pros y contras de iOS
<https://socialtecno.com/2018/09/04/ventajas-y-desventajas-de-android-y-ios-en-2018/>
- [5] Wikipedia – Características de Android
<https://es.wikipedia.org/wiki/Android#Caracter%C3%ADsticas>
- [6] Arquitectura de Android
<https://developer.android.com/guide/platform>
- [7] Versiones de Android
<https://es.wikipedia.org/wiki/Android#Versiones>
- [8] Cuota de mercado Android
<https://developer.android.com/about/dashboards/>
- [9] Material Trabajo Fin de Master UOC
 - Trabajo final de máster
 - Módulo 1: Introducción al trabajo final
 - Módulo 2: El trabajo final como proyecto
 - Módulo 3: La gestión del proyecto a lo largo del trabajo final.
 - Diseño centrado en el usuario: Muriel Garreta Domingo/Enric Mor Pera
 - Redacción de textos científicos-técnicos
 - Presentación de documentos y elaboración de presentaciones
 - Exposición de contenidos en vídeo
 - Tecnología y desarrollo en dispositivos móviles

Recursos Software

- [10] Diagrama de Gantt
<https://www.ganttproject.biz>
- [11] Diagramas UML
<https://www.lucidchart.com/>



- [12] Herramienta para la creación de prototipos
<https://www.justinmind.com/>
- [13] Firebase
<https://firebase.google.com/>
- [14] Firebase Cloud Firestore
<https://firebase.google.com/products/firestore?hl=es-419>
- [15] Realtime Database
<https://firebase.google.com/products/realtime-database>
- [16] Firebase Authentication
<https://firebase.google.com/products/auth>
- [17] Firebase Storage
<https://firebase.google.com/products/storage>
- [18] Firebase Cloud Messaging
<https://firebase.google.com/products/cloud-messaging>
- [19] Modelo Vista Controlador
<https://solidgeargroup.com/componentes-de-arquitectura-de-android-de-mvc-a-mvvm/>
- [20] Android Studio
<https://developer.android.com/studio>



Capítulo 7 – Anexos

7.1 Instrucciones de instalación y compilación

7.1.1 Instalación del entorno de programación

Para realizar la instalación del entorno de programación para poder compilar la aplicación, se deberán realizar los siguientes pasos: [20]

Paso 1. Descargar e instalar Android Studio (recomendable la última versión disponible)

<https://developer.android.com/studio>

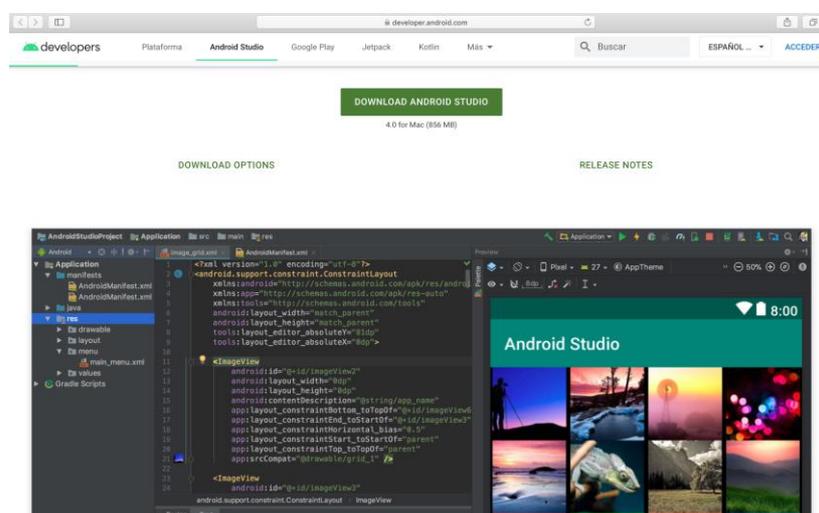


Ilustración 28. Android Studio

El paquete descargado incluye todas las herramientas necesarias para empezar a desarrollar la app:

- Android Studio IDE
- Herramientas SDK de Android
- Herramientas de la plataforma Android
- La última versión de la plataforma Android
- La última imagen del sistema para el emulador

Paso 2.

- Descomprimir el fichero “LostInstrument.zip” e importar proyecto en Android Studio.
- La estructura del proyecto quedará de la siguiente manera:

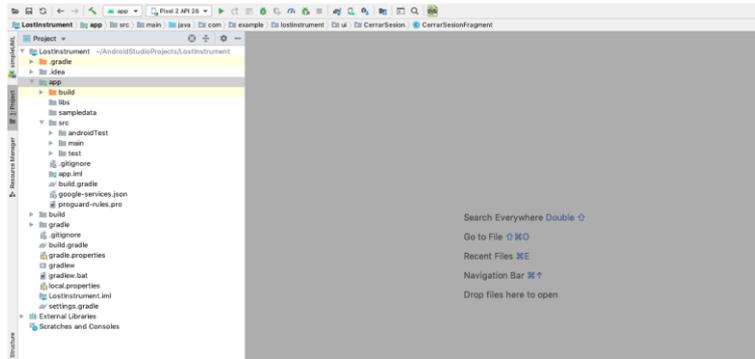


Ilustración 29. Vista principal IDE Android Studio

Paso 3. SDK Manager - Existen múltiples versiones de la plataforma Android, y cada una proporciona diferentes APIs y funcionalidades a los desarrolladores a través de un SDK (Software Development Kit). El SDK Manager es una herramienta de Android Studio que nos permite configurar qué versiones del SDK están disponibles en nuestro entorno de desarrollo. Para la realización de la práctica hemos utilizado la siguiente configuración

compileSdkVersion 28
minSdkVersion 21

CompileSdkVersion especifica el SDK de Android donde compila nuestra aplicación. En este caso es API 28 que corresponde a Android PIE.

minSdkVersion es 21 (Lollipop), lo que significa que **no** podremos utilizar la aplicación en un dispositivo que utilice un SDK menor de API 21.

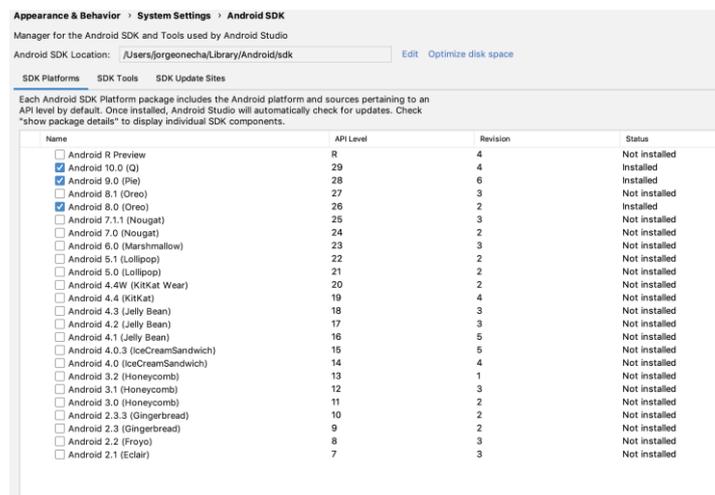


Ilustración 30. Versiones disponibles SDK Android Studio

Paso 4. Para poder ejecutar la aplicación, tendremos que definir el simulador. Utilizamos la herramienta AVD Manager, para crear un dispositivo virtual. Es importante que nuestro dispositivo sea superior a la API 21 – Lollipop.

Paso 5. Ejecutamos la opción Run App (Triángulo verde en forma de “Play”)



7.2 Manual de usuario

7.2.1 Inicializar la aplicación

Al pulsar sobre el icono de la aplicación situado en el Launcher del dispositivo Android, se iniciará la aplicación. Mientras se carga la aplicación, se muestra un Splash Screen:



Ilustración 31. Splash Screen

Una vez que esté cargada la aplicación, se mostrará al usuario la pantalla inicial de la aplicación:



Ilustración 32. Pantalla index



- **Botón iniciar sesión:** Nos permitirá acceder a la pantalla de login para acceder a la aplicación introduciendo el correo electrónico y la contraseña. Si no disponemos de una cuenta, podremos crearnos un usuario en la pantalla de login.
- **Iniciar con Facebook:** Nos permitirá acceder a la aplicación accediendo a Facebook e introduciendo las credenciales de autenticación.

7.2.2 Pantalla Login

Desde esta pantalla podremos crear una cuenta en la aplicación pulsando “Crear Cuenta”, o si ya disponemos de una cuenta podremos acceder a la aplicación introduciendo las siguientes credenciales:

- **Correo electrónico:** Correo con el que el usuario se ha registrado.
- **Contraseña:** Contraseña de la cuenta.

A continuación, se muestra la pantalla:

11:46

**LOST
INSTRUMENT**

Correo Electrónico

Contraseña

¿Olvidaste tu contraseña?

INICIAR SESIÓN

¿No tienes tu cuenta? **CREAR CUENTA**

Ilustración 33. Pantalla Login

7.2.3 Registrar un usuario

Para poder acceder a la aplicación necesitaremos disponer de una cuenta. Si no disponemos de ninguna, podremos crearnos una desde la pantalla de login pulsando en CREAR CUENTA. La aplicación nos dirigirá a un formulario que tendremos que rellenar para realizar el registro del usuario. A continuación, presentamos la pantalla para el registro del usuario:

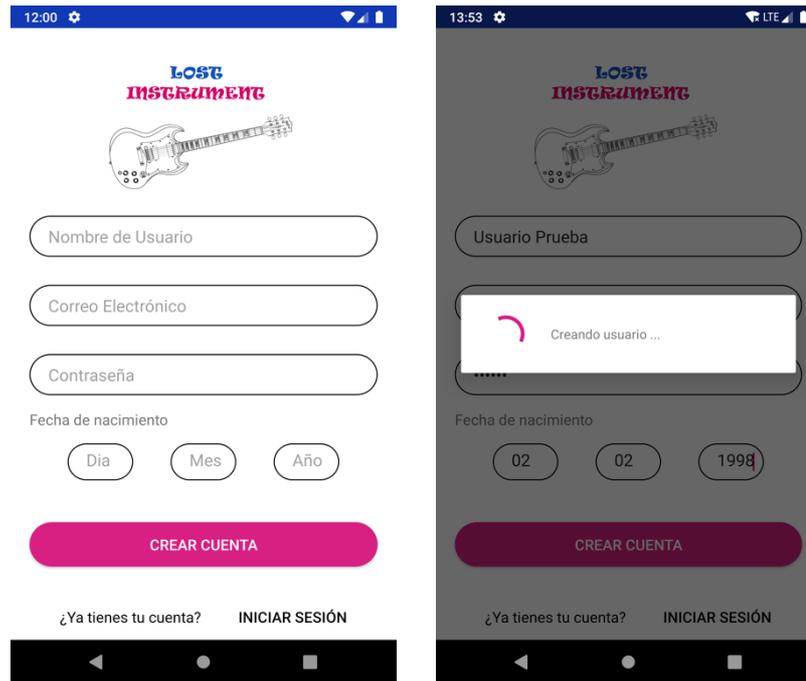


Ilustración 34. Pantalla Registro Usuario

7.2.4 Acceso a la aplicación

Para acceder a la aplicación tendremos que rellenar correctamente el correo electrónico del usuario junto a la contraseña y pulsar en iniciar sesión:



Ilustración 35. Acceso a la aplicación



Si hemos optado por la opción de inicio con FB:

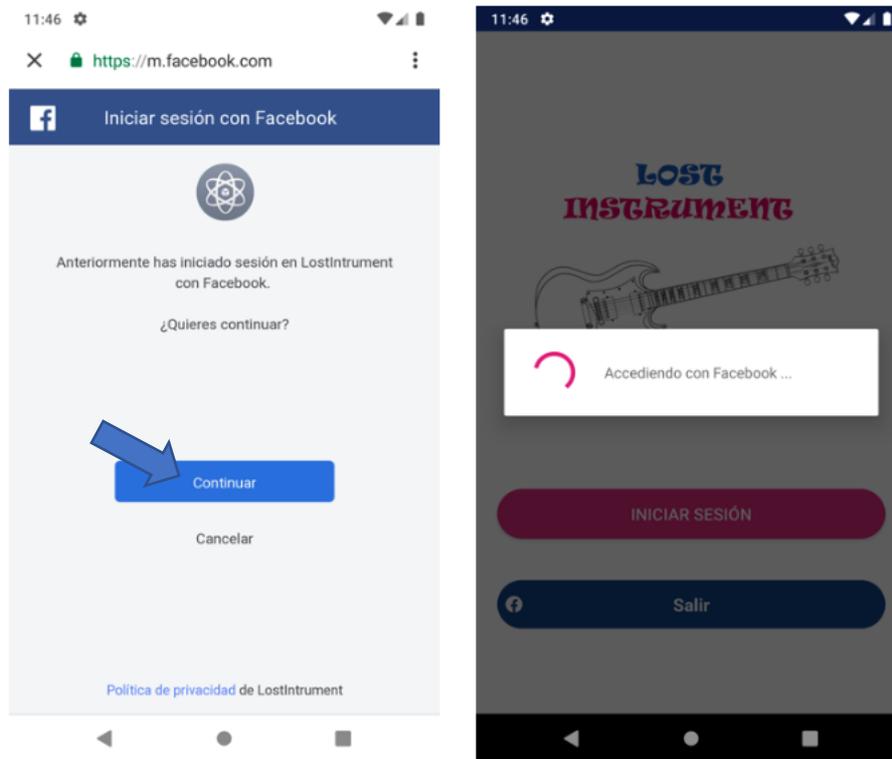


Ilustración 36. Acceso con Facebook

Cuando accedamos por primera vez, la aplicación nos pedirá confirmación para guardar las credenciales con la finalidad de que la próxima vez que volvamos a entrar, nos saltemos el paso de autenticarnos:

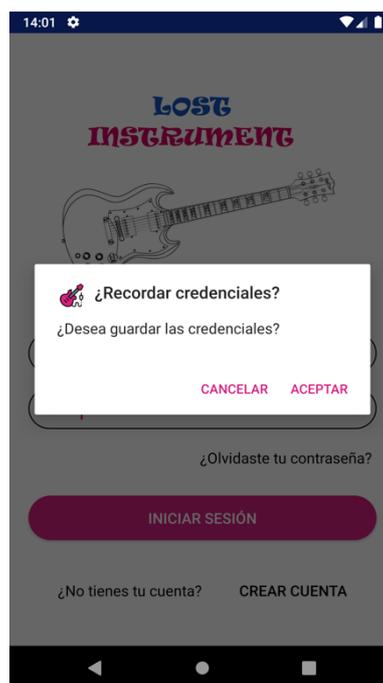


Ilustración 37. Guardar Credenciales



7.2.5 Menú principal

Pinchando en el icono de las tres rayitas (conocido como *hamburger*), se desplegará el menú principal de la aplicación. Desde este menú podremos acceder a todas las funcionalidades que nos presenta la aplicación. El menú tiene una orientación vertical y dentro de él se presenta de forma clara y concisa todas las opciones con un icono representativo de cada funcionalidad:

- Productos cerca de ti.
- Mis productos.
- Nuevo producto.
- Conversaciones
- Mapa de Productos
- Configuración
- Cerrar Sesión

En la parte superior del menú aparecerá el nombre del usuario y su correo electrónico junto con el logo de la app.

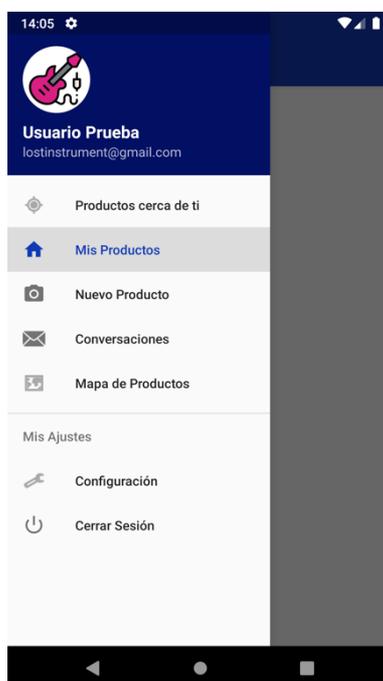


Ilustración 38. Pantalla Menú Principal

7.2.6 Visualizar mis productos subidos

Será la primera pantalla que se muestre al usuario según acceda a la aplicación. Desde esta pantalla se podrán consultar los productos que el usuario autenticado haya subido a la aplicación. Se mostrarán en forma de lista, en la cual aparecerá la siguiente información por cada producto:

- Foto principal del producto
- Nombre del instrumento
- Precio del instrumento.



La pantalla tiene el siguiente aspecto:

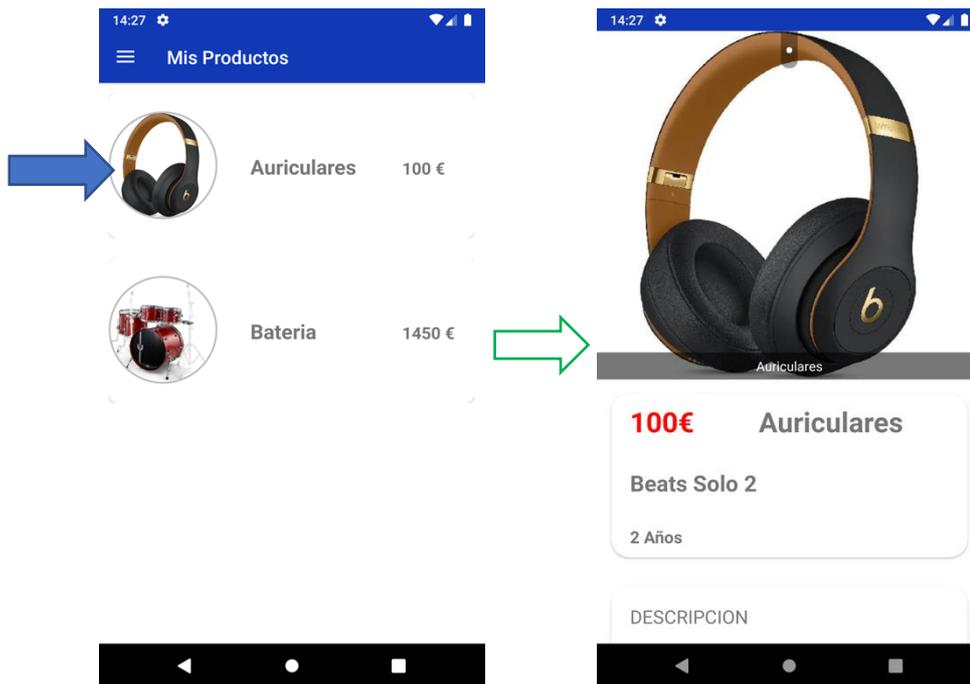


Ilustración 39. Seleccionar y acceder a un producto del usuario autenticado

7.2.7 Eliminar un producto subido

Desde la ficha del producto (pantalla anterior), podremos eliminar un producto que ya no nos interese tenerlo subido, ya sea porque lo hemos vendido o nos hayamos equivocado a la hora de subirlo. Para poderlo eliminar nos desplazaremos hasta debajo de la pantalla, hasta visualizar un botón llamado Eliminar Producto. Si lo pulsamos, saldrá un aviso para confirmar la operación. Esta acción no se podrá revertir.

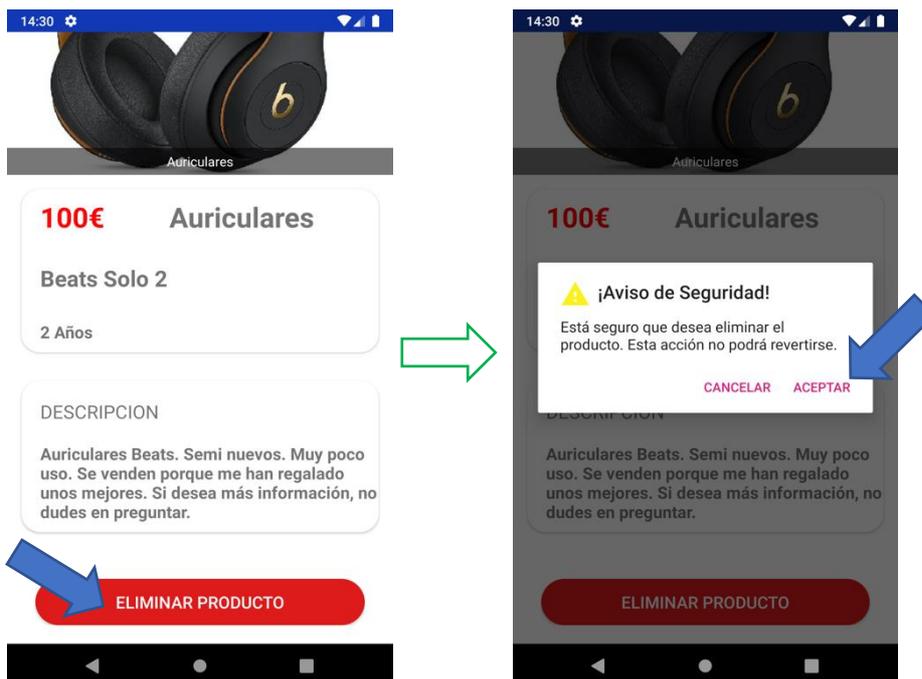


Ilustración 40. Eliminación producto subido



7.2.8 Nuevo Producto

Para poder subir un producto, el usuario deberá acceder al menú y allí seleccionar la opción nuevo producto. La aplicación nos mostrará un formulario para que el usuario introduzca los datos del producto. A continuación, se muestra el proceso:

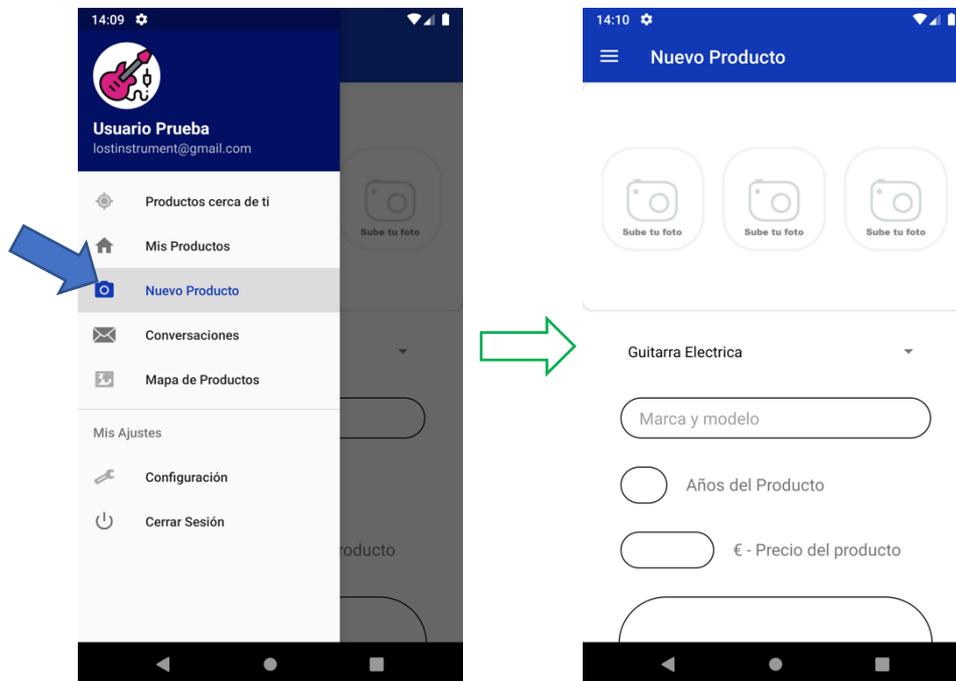


Ilustración 41. Crear nuevo producto

La aplicación nos pedirá los siguientes datos:

- **Subir foto:** El usuario podrá subir hasta 3 imágenes de cada producto. Para ello tendrá que pinchar en la imagen “Sube tu foto” y se abrirá un asistente donde el usuario podrá elegir si la imagen la sube de la galería o por el contrario desea abrir la cámara y hacer la foto en el acto. Una vez que el usuario haya elegido la foto, la aplicación le proporcionará una herramienta para realizar el recorte de la foto en el caso de que la imagen sea más grande de lo permitido. Una vez recortada, aparecerá en la pantalla. A continuación, se muestra el proceso:

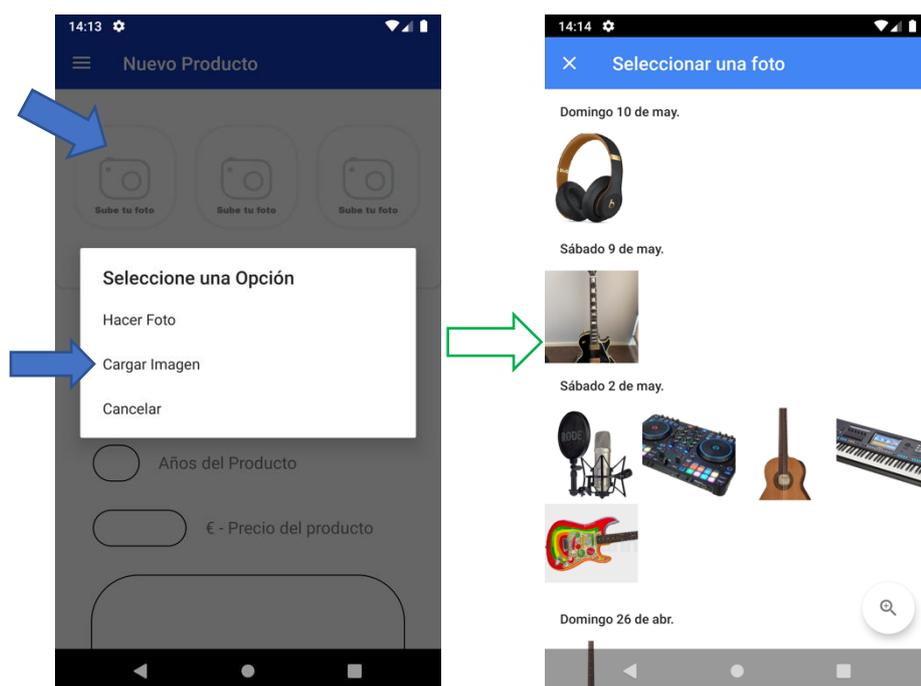


Ilustración 42. Subir imagen a la aplicación. Parte I

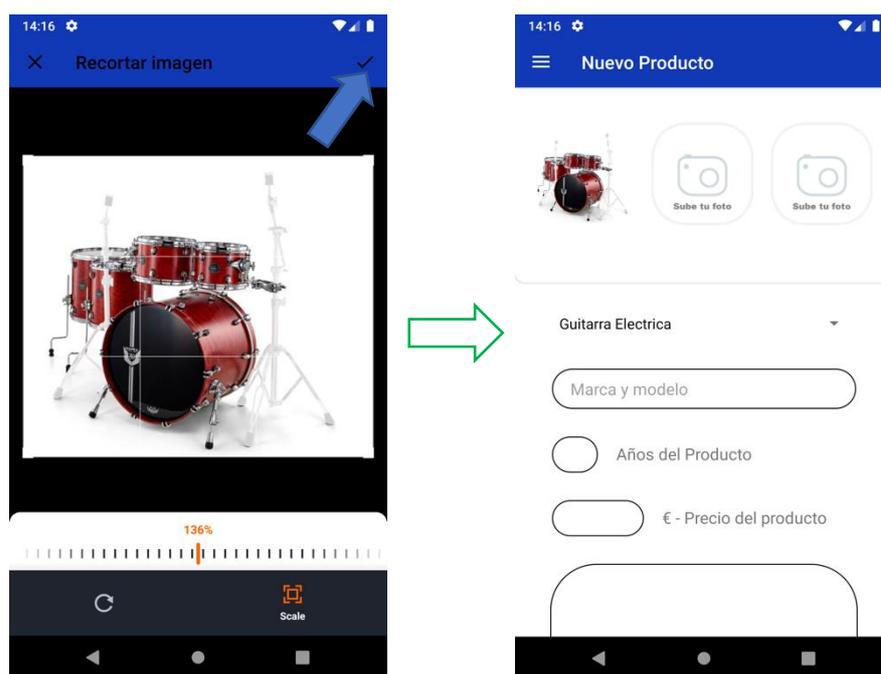


Ilustración 43. Subir imagen a la aplicación. Parte II

- El usuario dispondrá de un desplegable donde podrá elegir el nombre del producto que ha subido.
- Marca y modelo: El usuario deberá introducir la marca del producto y el modelo a subir.
- Precio del producto
- Comentario: En el último campo se le proporcionará al usuario un espacio para que explique los detalles y los datos a destacar del producto que desee subir.



Una vez que todo esté relleno, deberá pulsar en subir producto. La aplicación comprobará que no hay ningún campo vacío y que todo lo relleno es correcto, y subirá el producto a la aplicación.



Ilustración 44. Visualizar producto subido

7.2.9 Buscar un producto

Para buscar un producto, tendremos que desplegar el menú principal y acceder a la opción “Productos cerca de ti”. Una vez que estemos dentro, accederemos por defecto al botón “Productos Cerca” en el cual se cargarán los productos más cercanos a nuestra ubicación. Se podrá saber el tipo de instrumento que es, el precio, la distancia que hay desde la ubicación del usuario a la del producto e informará de la localidad donde se encuentra dicho producto. A continuación, mostramos un pantallazo de ejemplo:



Ilustración 45. Acceso a Productos cerca

Desde esta pantalla también podremos buscar un producto, introduciendo en el campo correspondiente el nombre del instrumento que buscamos. Para que empiece la búsqueda, pinchamos en la lupa.

A continuación, un ejemplo de una búsqueda:

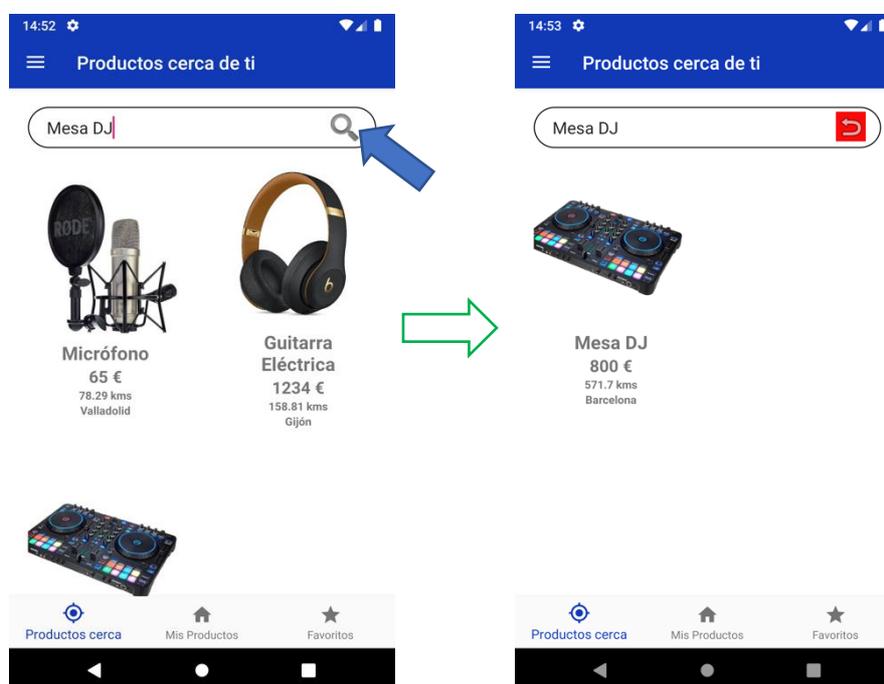


Ilustración 46. Búsqueda de un producto

Para eliminar la búsqueda bastará con volver a pinchar en la flecha roja y se restablecerán los productos.



7.2.10 Visualizar los detalles de un producto

Si el usuario desea visualizar los detalles de un producto de interés, deberá pulsarlo para acceder a dichos detalles. Ejemplo de una ficha de un producto:

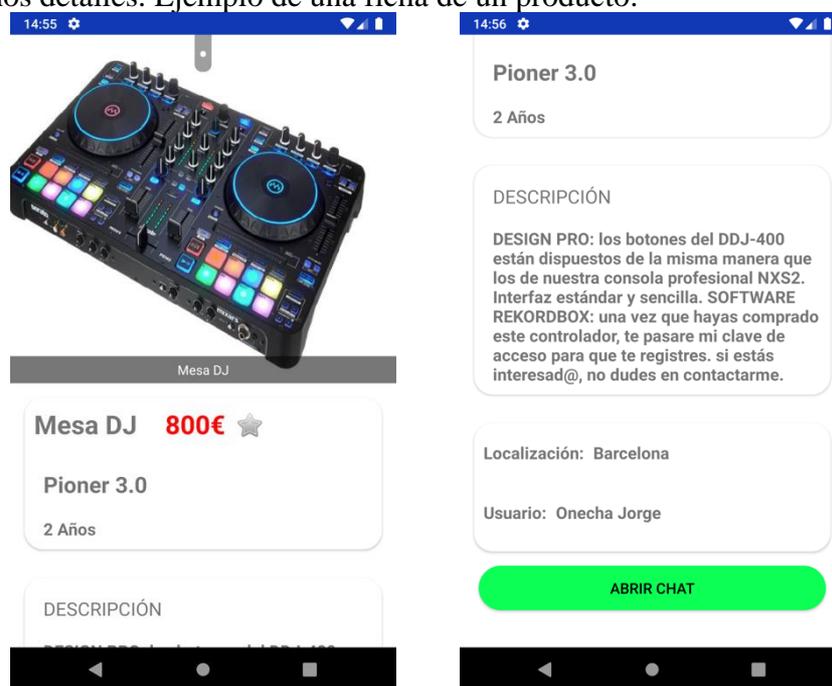


Ilustración 47. Ficha de un producto de un usuario

Desde esta ficha podremos ver los siguientes detalles:

- Imágenes del producto: La aplicación mostrará un Slider de fotos correspondientes al producto de interés.
- Nombre del instrumento
- Precio del instrumento
- Producto favorito: Estrella que marca si un producto está en nuestra lista de favoritos.
- Marca y detalles del nombre del instrumento
- Antigüedad del instrumento.
- Descripción del producto
- Localización: Nos informará de donde se encuentra el producto.
- Usuario: Usuario propietario del producto que estamos visitando.
- Abrir Chat: Será el botón que nos permita comunicarnos con el usuario propietario del producto.

7.2.11 Marcar como favorito un producto

El usuario mientras se encuentre visualizando los productos, podrá encontrar varios de su interés. Dichos productos los podrá guardar en una especie de listado para poder realizar un seguimiento. Para ello, deberá acceder a la ficha del producto y pulsar en la estrella de color gris para añadir ese producto a la lista de favoritos. Una vez pulsada, la estrella se cambiará a amarillo, la cual indica que el producto se ha añadido a nuestro listado de productos favoritos:



Ilustración 48. Marcar como favorito un producto

7.2.12 Visualizar el listado de productos favoritos

Para poder visualizar el listado de nuestros productos favoritos deberemos acceder a productos cerca de ti en el menú principal. Una vez que accedemos, desde los botones de la parte inferior tendremos que pulsar el de favoritos que tiene dibujado una estrella, para poder acceder al listado, tal y como se muestra a continuación:

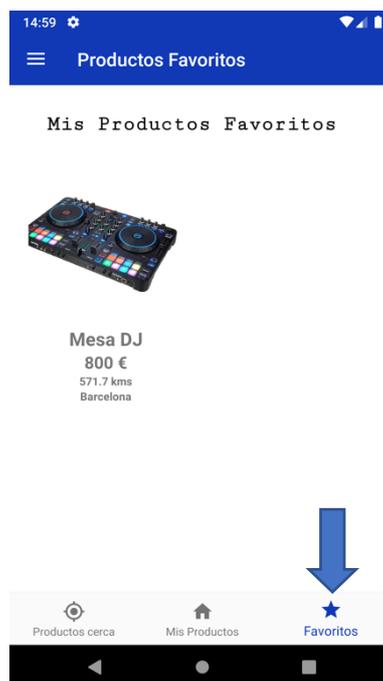


Ilustración 49. Visualizar listado de productos favoritos



7.2.13 Abrir un Chat

Cuando un producto sea de interés del usuario y esté decidido a comprarlo, la aplicación le proporciona una herramienta de chat para comunicarse con el propietario del producto. Para poder abrir un chat, el usuario se debe situar dentro de la ficha del producto y a continuación en la parte inferior de la pantalla, desplazándose hacia abajo podrá visualizar el botón “abrir chat” que le conducirá a una ventana de chat, donde podrá empezar a hablar con el usuario

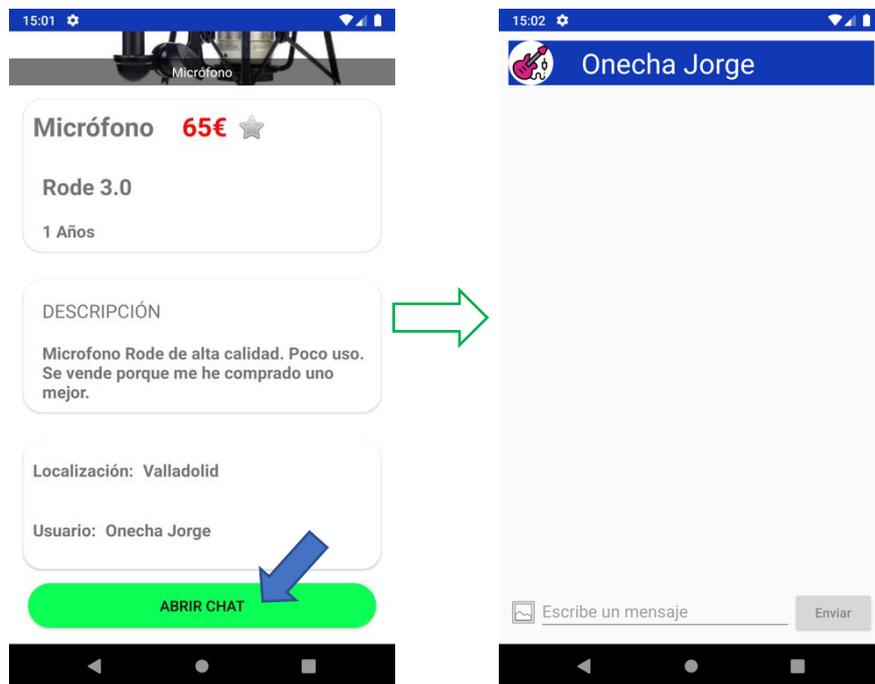


Ilustración 50. Abrir un Chat desde la ficha de un producto

7.2.14 Abrir una conversación iniciada

El usuario podrá acceder a todas las conversaciones que tiene en curso desde la pestaña Conversaciones del menú principal. Aparecerá un listado de todas ellas, y pulsando en la deseada, directamente se abrirá la ventana de chat recuperando los mensajes que se hayan escrito en el pasado.

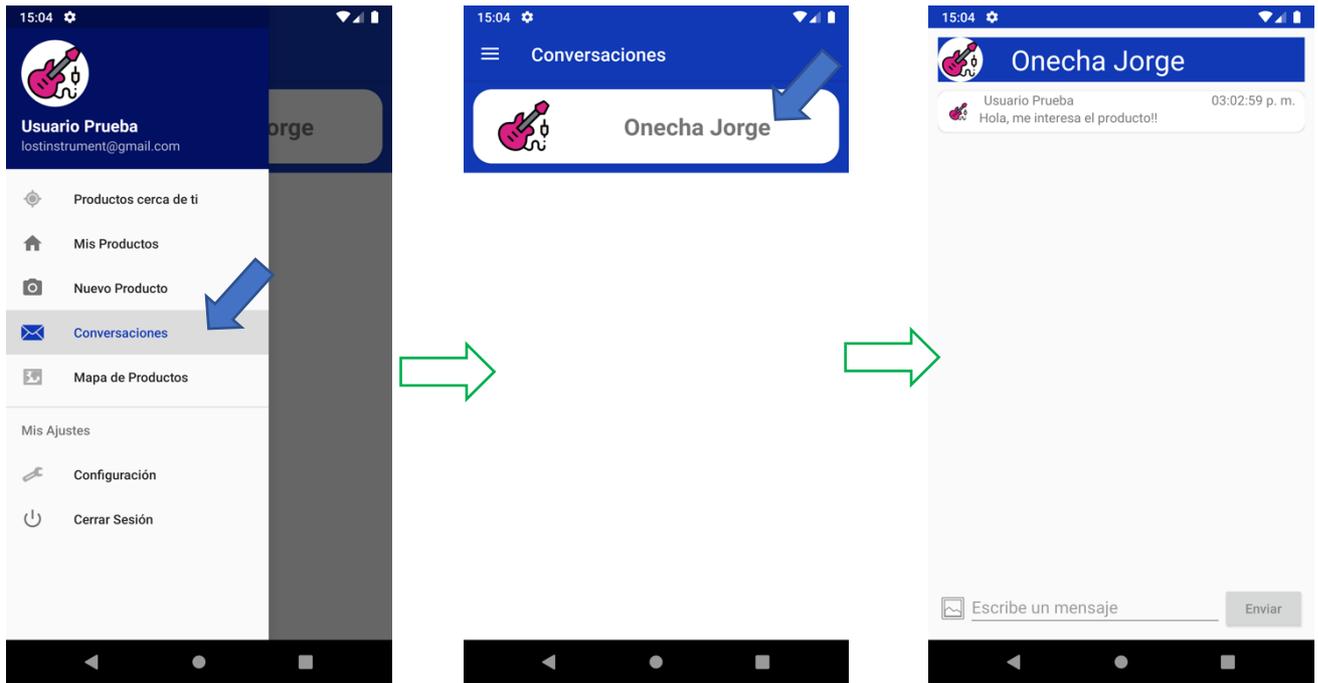


Ilustración 51. Acceder a conversación iniciada

7.2.15 Comprobar el mapa de productos

El usuario podrá acceder a la ventana de mapas, desde el menú principal, donde se encontrarán ubicados los productos que han subido los usuarios a la aplicación.



Ilustración 52. Mapa de productos



7.2.16 Modificar datos, actualizar contraseña y eliminar la cuenta.

El usuario podrá modificar sus datos personales, menos el email, que al ser único no se permite modificarlo. Para poder modificar los datos tendrá que dirigirse a la pantalla configuración desde el menú principal. Una vez que lo haya modificado, podrá guardar los cambios pulsando el botón “modifica tus datos” y confirmando el aviso que le saldrá por pantalla.



Ilustración 53. Modificar datos

7.2.17 Actualizar contraseña

Un usuario podrá cambiar la contraseña cuando lo desee. Para ello deberá acceder desde el menú principal a la ventana de configuración. Una vez que se encuentre allí deberá desplazarse hacia abajo hasta que encuentra el botón de actualizar contraseña:



Ilustración 54. Actualizar contraseña. Parte I

Una vez que lo pulse, la aplicación le conducirá a una nueva pantalla donde saldrá el email del usuario que será donde se envíe el correo electrónico. Desde el correo electrónico podrá acceder a una url que le permitirá actualizar su contraseña. Automáticamente se le sacará de la aplicación.

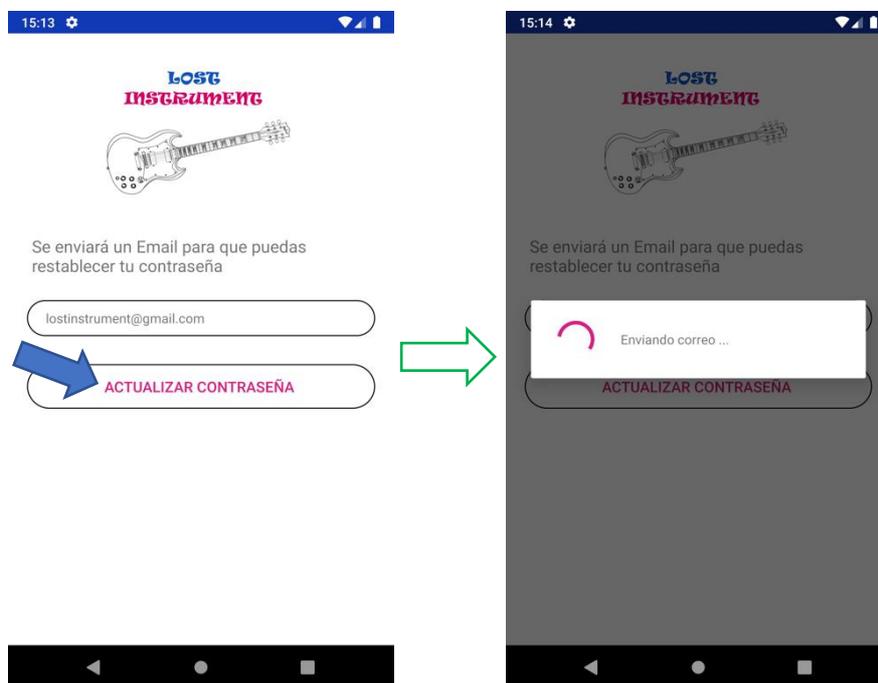


Ilustración 55. Actualizar contraseña. Parte II



7.2.18 Eliminar la cuenta

El usuario podrá eliminar su cuenta definitivamente cuando lo desee. Para poder realizar esta función, deberá acceder a la pantalla de configuración desde el menú principal y desplazándose hacia abajo en la pantalla encontrará un botón denominado “Eliminar la cuenta”. Al pulsarlo saldrá una confirmación. Si el usuario confirma, la aplicación realizará el borrado de la información y las imágenes del usuario. Una vez que termine, nos informará si ha tenido éxito y nos dirigirá a la pantalla inicial de la aplicación.

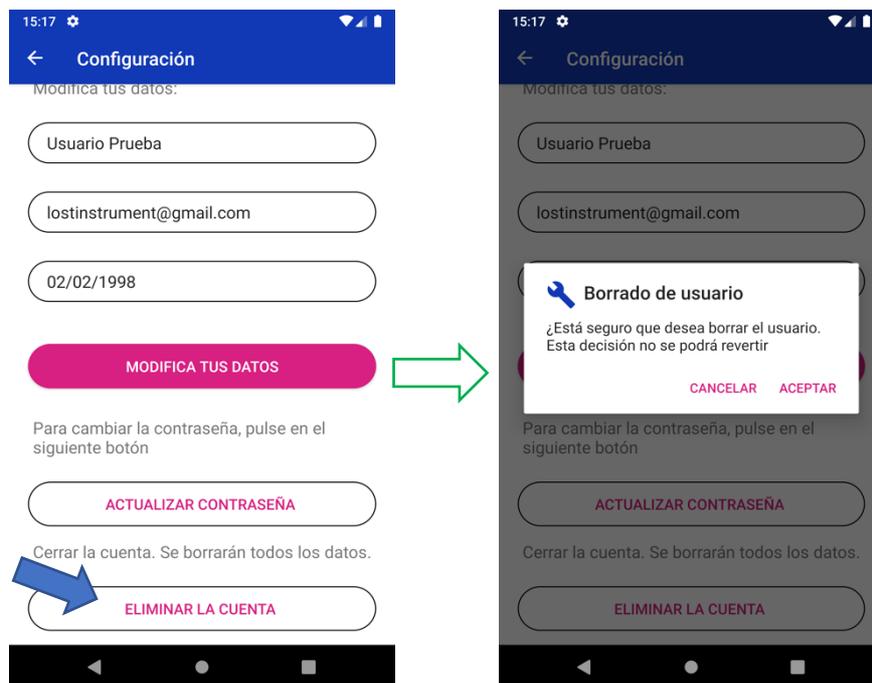


Ilustración 56. Eliminar usuario de la App