



Universidad de Valladolid

**ESCUELA DE INGENIERÍA INFORMÁTICA  
DE SEGOVIA**

**Grado en Ingeniería Informática de  
Servicios y Aplicaciones**

---

**BanDate: Aplicación móvil multiplataforma para la  
gestión y organización de los eventos de una banda de  
música**

---

**Alumno: Iván Acero Sánchez**

**Tutor: José Vicente Álvarez Bravo**



# Agradecimientos

Me gustaría expresar mi más sincero agradecimiento a todos los que de una manera o de otra contribuyeron a la elaboración de este TFG.

En primer lugar, quisiera dar las gracias a mi tutor José Vicente Álvarez Bravo por su orientación, paciencia y constante apoyo en este proyecto. Su sabiduría y experiencia han sido vitales para el éxito de este trabajo.

Mis padres y familiares merecen ser mencionados por su amor incondicional y por estar siempre ahí brindándome tanto apoyo emocional como financiero para completar mis estudios.

A mis amigos, quienes me acompañaron a lo largo del desarrollo de este trabajo, proporcionando no solo distracción y compañía para hacer de este camino algo más llevadero y divertido, sino también sus valiosos consejos y opiniones.

Por último, pero no menos importante, dar las gracias a todos los compañeros y profesores quienes han contribuido a mi formación académica y personal durante esos años de estudio.



# Resumen

La gestión eficiente de eventos y asistencia en una banda de música es crucial para su organización y funcionamiento. Tradicionalmente, la Banda de Música de Segovia conocida como Unión Musical Segoviana, utilizaba métodos anticuados y manuales para gestionar todos estos aspectos, lo cual resultaba ineficiente y daba lugar a muchos errores.

Este trabajo aborda esta problemática mediante el desarrollo de una aplicación móvil, que se decidió llamar BanDate. Esta aplicación está diseñada para facilitar la gestión de eventos, tanto ensayos como actuaciones, permitiendo un control centralizado y accesible para todos los miembros de la banda.

Para su desarrollo se ha utilizado *React Native*, lo que permite que la app pueda estar disponible en los principales sistemas operativos móviles, tanto iOS como Android. Además, se ha empleado *Firebase* para manejar los datos en tiempo real, proporcionando una sincronización inmediata de la información y mejorando la eficiencia en la comunicación y coordinación dentro de la banda.

La aplicación incorpora funcionalidades clave como la programación y gestión de eventos y la confirmación de asistencia de los distintos miembros. Estas características mejoran significativamente la organización interna de la banda, permitiendo una planificación más efectiva y reduciendo la carga administrativa.

La implementación del proyecto se ha llevado a cabo utilizando un *stack tecnológico* avanzado, garantizando una solución robusta, escalable y fácil de usar.

**Palabras clave:** BanDate, gestión de eventos, multiplataforma, sincronización en tiempo real, optimización.



# Abstract

Efficient event and attendance management in a music band is crucial for its organization and operation. Traditionally, the Segovia Music Band, known as Unión Musical Segoviana, used outdated and manual methods to manage all these aspects, which resulted in inefficiency and numerous errors.

This work addresses this issue by developing a mobile application, which was decided to be named BanDate. This application is designed to facilitate the management of events, both rehearsals and performances, allowing centralized control accessible to all band members.

React Native was used for its development, allowing the app to be available on the main mobile operating systems, both iOS and Android. Additionally, Firebase has been employed to handle real-time data, providing immediate information synchronization and improving communication and coordination efficiency within the band.

The application incorporates key functionalities such as scheduling and managing events and confirming attendance of various members. These features significantly enhance the internal organization of the band, allowing for more effective planning and reducing administrative burden.

The project implementation has been carried out using an advanced technological stack, ensuring a robust, scalable, and user-friendly solution.

**Keywords:** BanDate, event management, cross-platform, real-time synchronization, optimization.



# Índice general

Introducción .....	4
1.1. Motivación .....	5
1.2. Objetivos y alcance del proyecto .....	5
1.3. Estructura del documento .....	7
Estado del arte y entorno de trabajo .....	10
2.1 Propuestas software existentes .....	11
2.2 Comparativa .....	13
2.3 Herramientas y tecnologías utilizadas .....	14
Gestión del proyecto .....	21
3.1 Metodología de trabajo .....	21
3.2 Planificación .....	23
3.3 Presupuesto .....	31
3.4 Balance final del proyecto .....	33
Análisis .....	38
4.1 Características del sistema .....	38
4.2 Actores .....	40
4.3 Requisitos de usuario .....	41
4.4 Requisitos funcionales .....	68
4.5 Requisitos no funcionales .....	69
4.6 Requisitos de información .....	72
4.7 Restricciones del sistema .....	73
Diseño .....	75
5.1 Arquitectura lógica .....	75
5.2 Arquitectura física .....	78
5.3 Diagramas de secuencia .....	80
5.4 Diseño de la interfaz .....	82
5.5 Modelo lógico de datos .....	97
Implementación .....	101
6.1 Estructura del proyecto .....	101

6.2 Detalles de implementación .....	108
Pruebas .....	113
7.1 Pruebas de caja negra .....	113
7.2 Otras pruebas realizadas .....	116
Documentación .....	118
8.1 Manual de usuario .....	118
8.2 Manual de despliegue .....	130
Conclusiones .....	136
9.1 Valoración personal .....	136
9.2 Líneas de trabajo futuras .....	137
Referencias (webgrafía) .....	139
Índice de figuras.....	141
Índice de tablas.....	144



# Capítulo 1

## Introducción

Hoy en día estamos evolucionando rápidamente gracias a los avances tecnológicos y cambios sociales, y la forma de adaptarse y adoptar las nuevas tecnologías que están surgiendo es también cada vez más la prioridad número uno para mantener la competitividad y aumentar la eficiencia en la organización, ya sea una banda de música u otro grupo. La efectividad en el uso de las nuevas tecnologías no solo mejora las operaciones administrativas, sino que también coordina los ensayos, las actuaciones y la gestión de recursos de la manera más efectiva posible.

Tanto la eficiencia operativa como la coordinación eficiente son de suma importancia en la gestión de las asistencias de las bandas de música para un buen desempeño y una gestión sin problemas. Generalmente, estos procesos se han basado en gran medida en la intervención personal y la interacción cara a cara de las partes interesadas, que, aunque han demostrado ser muy beneficiosas en la mayoría de los aspectos, también se han demostrado que contienen fallas y limitaciones logísticas.

Los administradores y líderes se encuentran incapaces de ser tan ágiles como lo demanda un entorno en el que la innovación necesariamente tiene que ser continua. Deben emplear soluciones y técnicas que no solo los hagan procesar de manera eficiente desde una perspectiva operativa, sino que también estimulen la creatividad y la cooperación entre los individuos del equipo. Es en este momento que el despliegue adecuado de la gestión tradicional, con la incorporación de las últimas tecnologías, termina siendo una fuerza especial que tiene la promesa de asegurar el éxito de cualquier organización en el futuro cercano.

El foco principal de este proyecto se centra en la implementación de un sistema de gestión de asistencia para la Unión Musical Segoviana que utilice las nuevas tecnologías y la mejora de la comunicación interna, programación de eventos y logística de manera precisa y eficiente, y que no solo facilite la operación diaria de esta banda, sino que ayude en gran medida a la posibilidad de que crezcan.

BanDate no es solo una aplicación móvil, es la digitalización revolucionaria de una banda que puede ayudar a trabajar en una dirección más organizada y competitiva centrada en el rendimiento, con cada ensayo y actuación llevados a cabo de la manera más profesional y cuidadosa. Enfocado en ser escalable y usabilidad, se construirá este proyecto de tal manera que se desarrolle con el tiempo y el crecimiento de la banda.

## 1.1. Motivación

Como miembro activo de la banda Unión Musical Segoviana, se ha sido testigo directo de las dificultades que se tiene debido a métodos de gestión anticuados y comunicaciones fragmentadas. Coordinar ensayos, planificar actuaciones y gestionar la asistencia de nuestros miembros se convierte a menudo en un laberinto de confusiones y esfuerzos innecesarios.

Fue esto lo que encendió una chispa para solucionar el problema mediante el desarrollo de una aplicación móvil llamada BanDate. Inspirado por el potencial de la tecnología, se vio la oportunidad de crear una solución integral y accesible que revitalizara y mejorara los procesos internos. Diseñada con una arquitectura multiplataforma y una base de datos ágil, esta aplicación promete simplificar la coordinación en tiempo real y fomentar una correcta organización entre todos los integrantes de la banda.

Más allá de mejorar la logística diaria, BanDate aspira a ser un cambio cultural en nuestra organización. No solo se busca optimizar la eficiencia operativa, sino también fortalecer el espíritu colaborativo. Cada uno de los músicos deben tener un papel de responsabilidad para no cargar trabajo innecesario sobre otros, promoviendo así un ambiente de trabajo más amigable y a la vez profesional.

En un mundo que avanza rápidamente hacia lo digital, este proyecto representa un punto importante de mejora para optimizar la eficiencia y organización de una banda. Con BanDate, se va a ir marcando el compás hacia una gestión más eficiente y una experiencia musical más organizada y profesional dentro de la comunidad musical.

## 1.2. Objetivos y alcance del proyecto

A continuación, se van a presentar los objetivos principales de este trabajo, de los cuales los principales van dirigidos a mejorar la gestión de eventos y asistencias en la Unión Musical Segoviana. Estos objetivos son:

- Implementar una aplicación móvil multiplataforma utilizando React Native, asegurando disponibilidad tanto en iOS como en Android.
- Integrar Firebase para facilitar la gestión de datos en tiempo real, optimizando la comunicación y coordinación dentro de la banda.
- Simplificar la planificación y gestión de eventos mediante una interfaz intuitiva y sencilla para todos los miembros de la banda.
- Posibilidad de generar y exportar un PDF con los asistidos de un evento específico.
- Reducir la carga administrativa al automatizar procesos como la confirmación de asistencias y la actualización instantánea de información relevante.
- Promover un entorno de trabajo colaborativo y dinámico.
- Establecer roles claros y responsabilidades dentro de la plataforma.

- Implementar un sistema de registro controlado, garantizando que solo los miembros autorizados de la banda puedan acceder a la plataforma.
- Según el rol, se podrá marcar o desmarcar asistencias de otros miembros de la banda.
- Utilizar un *stack tecnológico* robusto y seguro que garantice la fiabilidad y escalabilidad de BanDate a medida que crece la demanda y la complejidad operativa.
- Implementar prácticas de seguridad y validaciones de datos.
- Detallar exhaustivamente todas las fases del desarrollo de BanDate.

Además de todo esto, hay que detallar los aspectos específicos que abarca el alcance del proyecto y aquellos que se dejan fuera para futuras versiones.

El proyecto se desarrollará con un enfoque *full stack*, para crear una aplicación multiplataforma utilizando React Native y que sea así compatible tanto en dispositivos iOS como Android, para que una gran variedad de usuarios pueda acceder a la plataforma sin importar su dispositivo. Por otro lado, se integrará Firebase para facilitar la gestión de datos en tiempo real, permitiendo así que todos los datos estén siempre actualizados y accesibles de inmediato.

En cuanto a la gestión de eventos y asistencias, BanDate contará con una interfaz de usuario intuitiva, diseñada para mejorar la planificación y organización de los distintos eventos y permitir a su vez que todos los miembros de la banda puedan acceder, crear y actualizar información facilitando el control.

Esta aplicación también se enfocará en fomentar la colaboración dentro de la banda mediante la definición de unos roles claros para cada miembro. Esto promoverá la participación y garantizará una comunicación transparente. Esto es para que el entorno de trabajo sea mucho más colaborativo y dinámico, lo que mejorará el espíritu de equipo y la eficacia en la coordinación de actividades y eventos.

Para garantizar la fiabilidad de la plataforma a medida que crece la demanda y la complejidad operativa, se utilizará un *stack tecnológico* robusto y seguro, y se implementarán prácticas de seguridad de datos estrictas para proteger la integridad y confidencialidad de la información de los usuarios y de los eventos organizados a través de la plataforma, como por ejemplo un inicio de sesión para que puedan acceder a las funcionalidades de la aplicación todos aquellos miembros que estén registrados.

Otro aspecto importante es la manera en que los usuarios pueden registrarse en nuestra plataforma. A diferencia de muchas aplicaciones móviles donde cualquiera puede crear una cuenta libremente, el enfoque de este proyecto es más exclusivo y controlado, ya que para asegurar que solo los miembros de la banda tengan acceso se debe implementar un sistema donde es necesario que un administrador cree las cuentas de los usuarios previamente. Así los administradores tienen la responsabilidad de registrar a cada miembro de la banda, lo que significa que se evita la entrada de personas ajenas al grupo.

Finalmente, se llevará a cabo con esta memoria una documentación exhaustiva de todas las fases del desarrollo de BanDate. Esta documentación abarcará desde la planificación inicial hasta la implementación final y las pruebas en un entorno de producción para asegurar así que todos los aspectos del desarrollo estén claramente registrados, facilitando el mantenimiento y futuras actualizaciones de la plataforma.

## 1.3. Estructura del documento

El documento está organizado en varios capítulos, cada uno centrado en diferentes aspectos del proyecto para facilitar su comprensión y análisis. Aquí se ofrece una descripción general de lo que se puede encontrar en cada capítulo:

- **Capítulo 1: Introducción**

Este capítulo comienza explicando la motivación detrás del proyecto, es decir, las razones que llevaron a emprenderlo. Luego, se detallan los objetivos y el alcance del proyecto, indicando qué se espera lograr y los límites del trabajo realizado.

- **Capítulo 2: Estado del arte y entorno de trabajo**

En este capítulo, se revisan las soluciones de software similares que ya existen, proporcionando una comparación entre ellas. Además, se describen las herramientas y tecnologías que se utilizaron durante el desarrollo del proyecto, dando al lector una idea del entorno de trabajo.

- **Capítulo 3: Gestión del proyecto**

Aquí se explica la metodología de trabajo que se siguió, así como la planificación del proyecto, incluyendo cronogramas y tareas. También se presenta un análisis financiero, mostrando el presupuesto del proyecto. Finalmente, se ofrece una evaluación final del proyecto, considerando los resultados obtenidos y cómo se ajustaron a los planes iniciales.

- **Capítulo 4: Análisis**

Este capítulo describe las características del sistema desarrollado y los actores involucrados, es decir, los usuarios y otras entidades que interactúan con el sistema. Se detallan los requisitos de usuario, las funcionalidades que el sistema debe cumplir, los atributos de calidad (requisitos no funcionales), la información que maneja el sistema, y las limitaciones y restricciones del proyecto.

- **Capítulo 5: Diseño**

En este capítulo se presenta la estructura conceptual del sistema (arquitectura lógica) y su configuración física (arquitectura física). También se incluyen diagramas de secuencia que muestran las interacciones dentro del sistema, el diseño de la interfaz de usuario y la estructura lógica de los datos manejados.

- **Capítulo 6: Implementación**

Aquí se describe cómo se organizó el código y los componentes del proyecto, y se ofrece una explicación técnica de cómo se llevó a cabo la implementación. Este capítulo es esencial para entender cómo se materializó el diseño en un producto funcional.

- **Capítulo 7: Pruebas**

En este capítulo se detallan las pruebas realizadas al sistema. Se describen las pruebas de caja negra, que se realizan sin conocimiento del funcionamiento interno del sistema, y otras pruebas adicionales que se llevaron a cabo para asegurar la calidad del producto final.

- **Capítulo 8: Documentación**

Este capítulo incluye el manual de usuario, con instrucciones claras y detalladas para el usuario final sobre cómo utilizar el sistema, y el manual de despliegue, que guía en la instalación y configuración del sistema.

## ▪ **Capítulo 9: Conclusiones**

El documento cierra con una valoración personal del autor sobre el trabajo realizado y los resultados obtenidos, y se sugieren posibles líneas de trabajo futuras para mejorar o expandir el proyecto en el futuro.



## Capítulo 2

# Estado del arte y entorno de trabajo

La evolución y creación de aplicaciones móviles para la gestión de bandas de música ha transformado significativamente la forma en que estas organizaciones operan y coordinan sus actividades. En un entorno donde la eficiencia y la precisión son cruciales, la adopción de tecnologías móviles ha permitido a las bandas optimizar procesos administrativos, mejorar la comunicación y coordinar eventos de manera más efectiva. BanDate se presenta en este contexto como una solución integral para la gestión de asistencias y eventos, adaptándose a las necesidades específicas de la Unión Musical Segoviana.

Actualmente, una gran cantidad de bandas de música alrededor del mundo utilizan aplicaciones móviles o páginas web para gestionar sus actividades y eventos. Aproximadamente el 70% de las bandas de música medianas y grandes han implementado alguna ayuda de tecnología móvil para facilitar la coordinación de sus actividades. Estas aplicaciones permiten a los miembros de la banda verificar sus horarios y compromisos desde cualquier lugar y en cualquier momento, proporcionando una flexibilidad y accesibilidad sin precedentes.

Las principales funcionalidades de estas aplicaciones incluyen la gestión de calendarios, el seguimiento de asistencias, la distribución de partituras y materiales, entre otras cosas. La posibilidad de tener toda esta información centralizada y accesible desde un dispositivo móvil facilita enormemente la organización y la toma de decisiones.

Sin embargo, a pesar de los beneficios evidentes de todas las aplicaciones que puedan existir, hay desafíos y oportunidades de mejora para alcanzar los requisitos propios de la banda de música de Segovia. Algunas áreas de mejora incluyen la integración de funciones más avanzadas, como la gestión de extras, creación de roles específicos para los cargos de los músicos y la exportación de documentos con los asistentes de los eventos ya realizados.

En este capítulo, se explorará el estado actual del arte sobre algunas propuestas existentes de aplicaciones móviles para la gestión de bandas de música que se han encontrado con características similares a las de este proyecto. Analizaremos el funcionamiento, las ventajas y desventajas de estas propuestas y, finalmente, se realizará una comparativa entre ellas. Con esta investigación, hay que identificar las mejores prácticas y los aspectos a considerar en el diseño y desarrollo de BanDate, con el objetivo de ofrecer una experiencia única y satisfactoria a los usuarios de la Unión Musical Segoviana.

## 2.1 Propuestas software existentes

A continuación, se describen algunas de las propuestas software estudiadas que son similares a la de este proyecto:

### 2.1.1. BandMule

Band Mule[1] es una plataforma digital diseñada para simplificar la organización y gestión de ensayos, conciertos y giras para bandas musicales. Los usuarios se pueden registrar en la plataforma y crear un perfil personalizado. Además, se pueden crear distintos grupos musicales e invitar a otros usuarios para que formen parte de ellos y comunicarse a través de un chat.

La gestión de actividades y el almacenamiento de listas y canciones son las funcionalidades clave de esta aplicación.

Los miembros de la banda pueden programar ensayos especificando la fecha, hora y lugar, y también cada usuario puede indicar su disponibilidad para futuras fechas, lo que facilita la planificación y asegura la disponibilidad de todos para los eventos programados. Otra cosa que se puede hacer es subir y compartir partituras y letras de canciones, facilitando la preparación y optimizando el tiempo de ensayo.

En cuanto a la organización de conciertos y giras, Band Mule ofrece un calendario detallado donde puedes ver los días que hay eventos, incluyendo conciertos y giras.

La comunicación es otro pilar fundamental de esta aplicación, ya que la plataforma incluye un sistema de mensajería interna que facilita la comunicación entre los miembros de la banda y otros colaboradores.

La interfaz de Band Mule es intuitiva y está diseñada para ser fácil de usar, y es compatible solamente en dispositivos móviles, tanto Android como iOS.



Figura 1. Imagen de BandMule [1]

## 2.1.2. Glissando

Glissando[2] es una aplicación tanto móvil como web diseñada para facilitar la gestión de grupos musicales y sus miembros. Esta herramienta es especialmente útil para bandas, orquestas, coros y cualquier otro tipo de conjunto musical.

Esta aplicación permite a los administradores de grupos agregar, editar y eliminar miembros con facilidad. Cada miembro tiene un perfil personalizado donde se pueden almacenar datos personales como nombre, instrumento/s que toca, contacto, y roles específicos dentro del grupo. Esta funcionalidad es esencial para mantener una organización clara y eficiente, asegurando que todos los integrantes estén al tanto de sus responsabilidades y contribuciones.

Una de las características más destacadas de Glissando es su capacidad para que los administradores programen ensayos, conciertos y otros eventos importantes, notificando automáticamente a todos los miembros del grupo y que estos puedan agregarlo a su calendario propio de Google. Los miembros pueden confirmar su asistencia a los eventos o, en caso de no poder asistir, escribir una razón específica. Además, si los usuarios lo desean, pueden agregar automáticamente los nuevos eventos publicados a su propio calendario de Google, permitiendo así una mejor organización personal y asegurando que no se olviden de los compromisos importantes. Esta integración con Google Calendar facilita la sincronización de horarios y mejora la eficiencia en la gestión del tiempo para todos.

Esta aplicación también permite crear y gestionar listas de repertorio, facilitando el acceso a partituras y grabaciones de referencia. Los músicos pueden revisar las piezas asignadas para un evento en concreto desde cualquier lugar para no olvidarse de ello y así poder practicar lo establecido, mejorando la preparación individual y colectiva para aumentar la eficiencia durante los ensayos y actuaciones.

Cada grupo musical tiene sus propias necesidades y dinámicas. Glissando permite que los administradores configuren y creen los grupos deseados, adaptando cada uno a las particularidades de su funcionamiento.

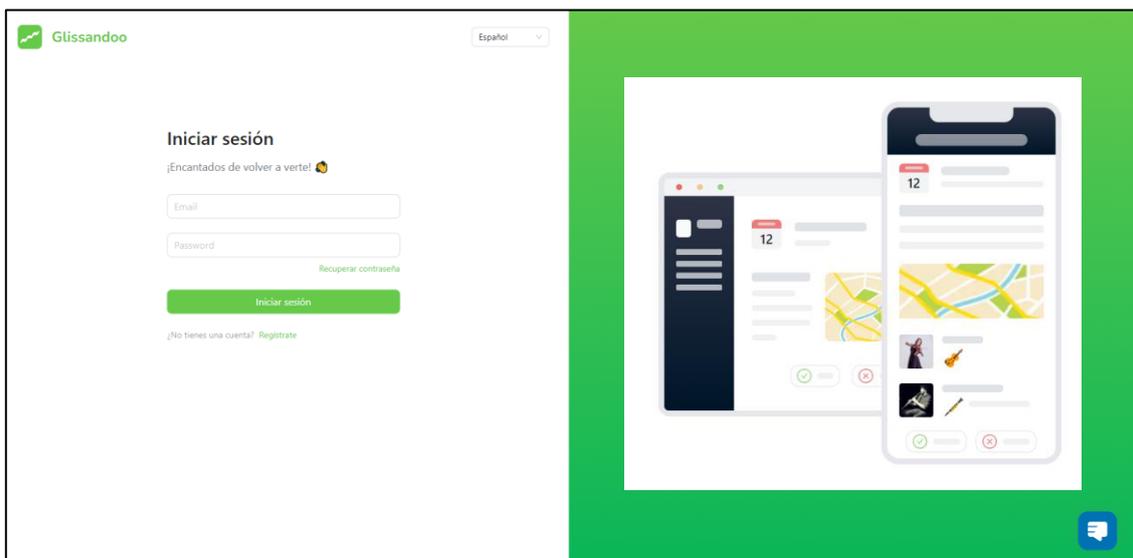


Figura 2. Imagen de Glissando [2]

## 2.2 Comparativa

Una vez analizadas las distintas aplicaciones existentes en el mercado y siendo las más parecidas a BanDate, se ha realizado un análisis de las funcionalidades que ofrecen estas alternativas y se ha creado esta tabla comparativa a modo resumen que explica de manera clara y concisa las diferencias y similitudes:

	BandMule	Glissandoo	BanDate
Creación y gestión de eventos musicales	✓	✓	✓
Aplicación web y móvil	✗	✓	✗
Registro de usuarios libre	✓	✓	✗
Existencia de distintos roles	✗	✓	✓
Creación de múltiples grupos musicales	✓	✓	✗
<b>Agregar músicos extras en casos especiales</b>	✗	✗	✓
Crear y gestionar listas de repertorio	✓	✓	✗
Control y gestión de los miembros registrados	✗	✓	✓
Personalización del perfil	✓	✓	✓
<b>Gestión de eventos ya realizados</b>	✗	✗	✓
<b>Exportar PDF de asistidos</b>	✗	✗	✓
Calendario con los eventos musicales	✓	✓	✗

Tabla 1. Tabla comparativa del proyecto con otras apps

Al observar la tabla comparativa, se pueden identificar varios puntos fuertes que tiene la aplicación BanDate frente a las otras dos:

1. **Agregar músicos extras en casos especiales:** BanDate es la única de las tres aplicaciones que permite agregar músicos extras en situaciones especiales. Esta funcionalidad es crucial para gestionar imprevistos y garantizar la flexibilidad en la organización de eventos musicales, algo que no ofrecen las otras dos aplicaciones.
2. **Gestión de eventos ya realizados:** Otra ventaja significativa de BanDate es su capacidad para gestionar eventos ya realizados. Esta función facilita el seguimiento de eventos pasados, permitiendo a los usuarios aprender y mejorar la planificación futura, un beneficio que no está disponible en Band Mule ni en Glissandoo.
3. **Exportar PDF de asistidos:** La posibilidad de exportar PDFs de los asistentes es otra funcionalidad exclusiva de BanDate. Esta característica es esencial para llevar un registro formal y detallado de la participación en los eventos, proporcionando documentación útil para futuras referencias y análisis.

## 2.3 Herramientas y tecnologías utilizadas

En esta sección se expondrán las herramientas y tecnologías que se han utilizado para el desarrollo de este trabajo.

### 2.3.1. Visual Studio Code



*Figura 3. Visual Studio Code*

Visual Studio Code comúnmente conocido como VS Code es un editor de código fuente desarrollado por Microsoft, que se ha convertido en una de las herramientas más populares entre los desarrolladores debido a su versatilidad y eficiencia.

Es un editor ligero pero potente, diseñado para trabajar con una amplia variedad de lenguajes de programación y tecnologías. Ofrece características avanzadas como autocompletado de código, resaltado de sintaxis, depuración integrada y terminal de comandos, todo en una interfaz limpia y fácil de usar. Una de las mayores ventajas de VS Code es su capacidad de ser altamente personalizable mediante extensiones que se pueden instalar desde su marketplace. Estas extensiones permiten agregar funcionalidades específicas que se adaptan a las necesidades particulares de cada proyecto, desde soporte para distintos lenguajes de programación hasta herramientas de productividad y colaboración.

Además, VS Code se integra de manera excelente con sistemas de control de versiones como Git, lo que facilita el seguimiento de los cambios en el código y la colaboración con otros desarrolladores. También es compatible con una amplia gama de sistemas operativos, incluyendo Windows, macOS y Linux, lo que lo hace accesible para desarrolladores en diversas plataformas.

### 2.3.2. JavaScript



*Figura 4. JavaScript*

JavaScript es un lenguaje de programación interpretado que permite la creación de páginas web dinámicas y aplicaciones interactivas. Lo que hace a JavaScript particularmente especial es su capacidad para ejecutarse tanto en el lado del cliente (navegador) como en el lado del servidor. Esta dualidad ha permitido utilizar un único lenguaje para manejar tanto la lógica de la aplicación como la interacción con el servidor.

Uno de los aspectos más atractivos de JavaScript es su vasto ecosistema de bibliotecas y frameworks, como React, Angular y Vue.js, que simplifican y aceleran el proceso de desarrollo.

JavaScript también es conocido por su comunidad activa y en constante crecimiento. La gran cantidad de recursos disponibles, como tutoriales, foros y documentación detallada, hizo que el aprendizaje y la resolución de problemas fueran más accesibles. Además, la naturaleza asíncrona de JavaScript, manejada a través de promesas y `async/await`, permitió que esta aplicación manejara múltiples tareas simultáneamente, mejorando la experiencia del usuario.

### 2.3.3. Expo



*Figura 5. Expo*

Expo es una plataforma excelente que simplifica y potencia la creación de aplicaciones junto con React Native. Expo proporciona un conjunto de herramientas y servicios que facilitan el proceso de desarrollo, desde la inicialización del proyecto hasta la publicación en las tiendas de aplicaciones.

Esta plataforma es especialmente útil porque elimina muchas de las complejidades asociadas con el desarrollo móvil. Con Expo, no es necesario configurar un entorno de desarrollo nativo para iOS o Android, lo que ahorra una cantidad significativa de tiempo y esfuerzo. La plataforma proporciona un conjunto de APIs preconfiguradas que permiten acceder a características nativas del dispositivo, como la cámara, el GPS y las notificaciones, sin necesidad de escribir código específico para cada sistema operativo.

Una de las características más destacadas de Expo es Expo Go, una aplicación que permite probar la aplicación en tiempo real en un dispositivo físico o emulador simplemente escaneando un código QR. Esto acelera considerablemente el ciclo de desarrollo y prueba, ya que los cambios en el código se reflejan inmediatamente en la aplicación.

Además, Expo cuenta con Expo CLI, una interfaz de línea de comandos que facilita la creación, gestión y despliegue de proyectos. Con comandos sencillos, es posible inicializar un nuevo proyecto, ejecutar la aplicación en un emulador o dispositivo real, y generar *builds* listas para ser publicadas en las tiendas de aplicaciones.

La comunidad de Expo es otro de sus grandes puntos fuertes ya que, con una documentación exhaustiva, foros activos y numerosos ejemplos de código, siempre hay recursos disponibles para resolver dudas y aprender nuevas técnicas y habilidades. Además, Expo se mantiene actualizado con las últimas versiones de React Native, lo que garantiza que las aplicaciones desarrolladas con esta plataforma sean modernas y estén optimizadas.

## 2.3.4. React Native

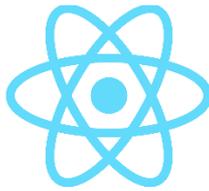


Figura 6. React Native

React Native es un framework de código abierto desarrollado por Facebook que permite construir aplicaciones móviles utilizando JavaScript y React. Este framework ha revolucionado el desarrollo móvil al permitir que los desarrolladores usen un único código base para crear aplicaciones nativas para iOS y Android.

React Native se basa en React, una biblioteca de JavaScript para construir interfaces de usuario, pero en lugar de renderizar componentes web renderiza componentes nativos. Esto significa que con React Native se pueden crear aplicaciones que ofrecen la experiencia y el rendimiento de una aplicación nativa, pero con la eficiencia de desarrollo de una aplicación web.

Uno de los mayores beneficios de React Native es la capacidad de reutilizar el código. Alrededor del 90% del código puede ser compartido entre las versiones de iOS y Android, lo que reduce significativamente el tiempo y el esfuerzo necesarios para desarrollar aplicaciones para ambas plataformas. Esta característica fue especialmente valiosa para el proyecto, ya que permitió enfocar más tiempo en mejorar las funcionalidades y la experiencia del usuario, en lugar de duplicar esfuerzos.

También se destaca el rendimiento, ya que, aunque utiliza JavaScript, los componentes se traducen en vistas nativas, lo que garantiza que la aplicación funcione de manera rápida y fluida. Además, el gran crecimiento de React Native ha permitido que exista una gran cantidad de bibliotecas y componentes reutilizables, lo que facilita la integración de funcionalidades avanzadas sin necesidad de escribir código desde cero.

Otra ventaja significativa es el desarrollo en caliente (*Hot Reloading*). Esta característica permite ver los cambios en el código de inmediato, sin necesidad de recompilar la aplicación. Esto agiliza enormemente el proceso de desarrollo y depuración, haciendo que iterar sobre el diseño y la funcionalidad sea mucho más eficiente.

## 2.3.5. Firebase



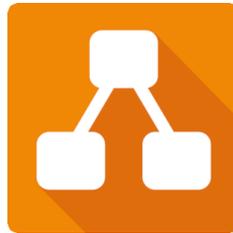
Figura 7. Firebase

En el desarrollo de esta aplicación móvil, Firebase ha sido una herramienta indispensable. Es una plataforma de desarrollo de aplicaciones móviles y web creada por Google que proporciona una amplia gama de servicios y herramientas para mejorar y facilitar el desarrollo de aplicaciones.

Firestore destaca por ofrecer una infraestructura en la nube que permite gestionar diversas funcionalidades sin necesidad de configurar servidores complejos. Uno de los servicios más importantes que se ha utilizado es Cloud Firestore, una base de datos NoSQL que permite almacenar y sincronizar datos en tiempo real. Esta funcionalidad es crucial para aplicaciones que requieren actualizaciones instantáneas y colaborativas, como chats o aplicaciones de seguimiento en tiempo real.

Otra característica poderosa de Firebase es su sistema de autenticación, Firebase Authentication. Este servicio permite gestionar de manera sencilla y segura el registro y la autenticación de usuarios, soportando métodos como el inicio de sesión con correo electrónico y contraseña, Google, Facebook, entre otros. Implementar una autenticación robusta y segura fue un aspecto esencial para garantizar la protección de los datos de los usuarios en la aplicación.

### 2.3.6. draw.io



*Figura 8. Draw.io*

A la hora de crear diagramas y representaciones visuales de manera sencilla y efectiva, se hizo un uso extensivo de draw.io. Es una herramienta en línea que ha permitido diseñar diagramas de caso de uso, diagramas de flujo, estructuras de datos y otros, esenciales para visualizar la funcionalidad de la aplicación móvil que estoy desarrollando.

Lo que más atrae de draw.io es su facilidad de uso, ya que posee una interfaz intuitiva y su simplicidad fue crucial para plasmar las ideas y comunicarlas de manera clara.

Además draw.io funciona completamente en el navegador web, lo que ayuda mucho a la hora de trabajar desde cualquier lugar, sin necesidad de instalar software adicional.

La capacidad de exportar los diagramas en diferentes formatos, como imágenes PNG o PDF, fue otra ventaja significativa. Esto permitió integrar fácilmente los diagramas en la documentación del proyecto y presentarlos de manera profesional.

### 2.3.7. MicrosoftProject



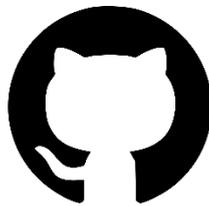
*Figura 9. MicrosoftProject*

Microsoft Project es una herramienta de gestión de proyectos desarrollada por Microsoft, diseñada para ayudar a planificar, asignar recursos, realizar un seguimiento del progreso y gestionar presupuestos. Para este trabajo se ha utilizado para la planificación detallada del proyecto.

Esta herramienta ha permitido desglosar el proyecto en tareas específicas, establecer relaciones y dependencias entre ellas, y asegurar que todas las actividades se organicen de manera lógica y eficiente. Además, ha sido fundamental para la creación de diagramas de Gantt, que ofrecen una representación visual del calendario del proyecto, facilitando el seguimiento y la comunicación del progreso a todos los involucrados.

Gracias a estas funcionalidades, se ha podido gestionar de manera efectiva los tiempos y recursos, asegurando el éxito del proyecto.

### 2.3.8. GitHub



*Figura 10. GitHub*

GitHub es una plataforma de desarrollo colaborativo y control de versiones basada en Git, ideal tanto para proyectos individuales como colaborativos. En este proyecto se ha utilizado GitHub para gestionar el código fuente y controlar las versiones del proyecto.

Esta plataforma ha permitido almacenar y organizar el código de manera segura, realizar un seguimiento detallado de los cambios y gestionar diferentes versiones del proyecto de forma eficiente. Ha sido esencial para mantener un flujo de trabajo estructurado, permitiendo revertir cambios si era necesario y asegurar la integridad del proyecto.

### 2.3.9. Word



*Figura 11. Word*

Microsoft Word es un procesador de textos ampliamente utilizado desarrollado por Microsoft. Esta herramienta ha sido esencial en el proyecto para la creación y edición de documentos. Con Word, se ha podido redactar informes detallados, preparar documentación del proyecto y elaborar presentaciones.

La facilidad de uso de Word ha permitido dar formato profesional a los documentos, integrar imágenes y tablas, y asegurar que toda la información se presente de manera clara y organizada. Además, sus funcionalidades de revisión y colaboración han facilitado la revisión y corrección de textos, mejorando la calidad del contenido final.



# Capítulo 3

## Gestión del proyecto

En esta sección se explicará cómo se ha desarrollado la aplicación, estimando tanto su tamaño total como el tiempo necesario para completar el proyecto. También se detallará cómo se ha planificado el uso de los recursos disponibles para asegurar que se termine en el tiempo más eficiente posible, y tras esto se incluirá y describirá un presupuesto aproximado que cubra los costos de todas las herramientas requeridas para desarrollar y desplegar la aplicación.

### 3.1 Metodología de trabajo

Después de revisar exhaustivamente las metodologías y enfoques más usados en el desarrollo de proyectos de software, se ha optado por implementar en este proyecto el modelo iterativo-incremental[3]. Esta metodología combina elementos tanto del modelo en cascada como del desarrollo basado en prototipos.

El modelo iterativo-incremental se caracteriza por desarrollar la herramienta de manera evolutiva mediante la planificación de múltiples bloques de trabajo, conocidos como iteraciones o fases. Durante estas iteraciones, se amplía y perfecciona continuamente la funcionalidad y las características del producto final.

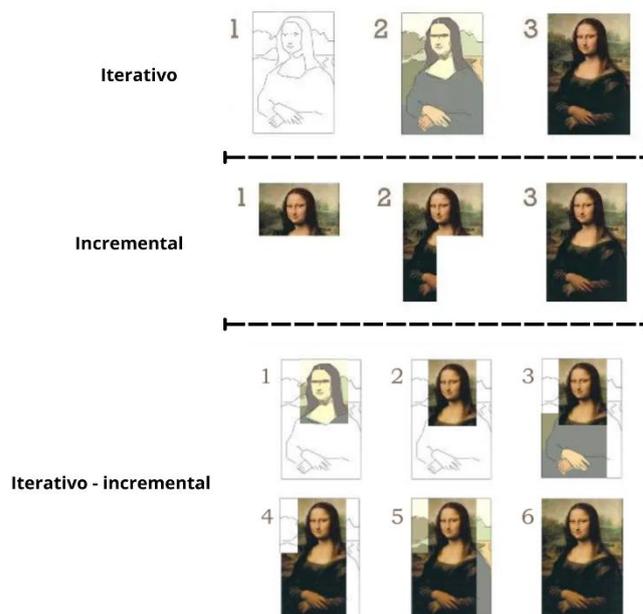


Figura 12. Metodología de trabajo [3]

Así como se ha mencionado, el desarrollo se organizará en varias fases que incluirán la introducción de nuevas funcionalidades (incrementos) y la adaptación y mejora de las funcionalidades existentes (iteraciones).

El proyecto se dividirá en **9 fases**: la primera consistirá en el aprendizaje de las herramientas a utilizar, creación del prototipado, análisis y planificación, seguida por 8 incrementos que añadirán funcionalidades a la aplicación y mejorarán las existentes, junto con el despliegue de la app, y en la fase final se realizará en mayor medida la documentación del proyecto, que se llevará a cabo de manera continua y al mismo tiempo que todo el desarrollo del software para dejar todo bien explicado y detallado.

- **Fase inicial del proyecto:** En esta fase inicial, el objetivo principal es adquirir un profundo conocimiento de las herramientas y tecnologías clave necesarias para el proyecto BanDate, al mismo tiempo que se avanzará en la creación de un prototipo funcional que guiará el diseño y desarrollo futuro. Se realizarán pruebas y pequeños proyectos de ejemplo para familiarizarnos plenamente con estas tecnologías y asegurar que el equipo de desarrollo esté completamente preparado para abordar los desafíos que surgirán en las próximas etapas del proyecto. También se han realizado actividades de análisis de requisitos y planificación, asegurando que todos los aspectos y alcance del sistema estén documentados y claros para el correcto desarrollo del proyecto.
- **Primer incremento:** En esta fase la tarea principal será la implementación de funcionalidades relacionadas con la gestión de eventos, específicamente la creación, edición, eliminación y visualización detallada de los ensayos y actuaciones.
- **Segundo incremento:** La siguiente fase abordará la funcionalidad de registro y gestión de cuentas, incluyendo la definición de roles específicos, y registro de extras.
- **Tercer incremento:** Aquí se implementarán las funcionalidades esenciales de autenticación, recuperación de contraseña y cierre de sesión.
- **Cuarto incremento:** Este incremento es crucial para asegurar la seguridad y la correcta distribución de responsabilidades dentro de la banda. Se enfocará en la implementación de un sistema de control de roles y permisos dentro de la aplicación, para garantizar que solo los usuarios autorizados puedan acceder a ciertas ventanas y funcionalidades exclusivas según su rol.
- **Quinto incremento:** Esta fase es crucial ya que se centrará en la implementación de la funcionalidad para marcar asistencias dentro de los detalles de los eventos, ya sea ensayos o actuaciones. Esta característica permitirá a los administradores, jefes de cuerda y músicos gestionar y registrar las asistencias de manera eficiente y ordenada. Es importante controlar los roles ya que determinará qué miembros pueden marcar asistencias y de quiénes pueden hacerlo.
- **Sexto incremento:** se desarrollará la funcionalidad de la pantalla de listados, que incluirá un listado de eventos ya realizados y un listado de usuarios registrados. Esta pantalla permitirá a los administradores gestionar tanto los eventos pasados como los usuarios, con funcionalidades de eliminación, ascenso y descenso de roles. Además, se implementará la capacidad de ver los asistentes de eventos pasados y eliminar eventos antiguos si es necesario.

- **Séptimo incremento:** Aquí se añadirán dos funcionalidades clave, la capacidad de generar y exportar un PDF con los asistentes a los eventos realizados y guardar la sesión en local en el móvil para mejorar la experiencia del usuario al evitar que tengan que iniciar sesión repetidamente.
- **Octavo incremento:** En esta fase afinamos los detalles para asegurarnos de que todo funcione de manera fluida y segura mediante la implementación de validaciones de datos en formularios, la validación de la conectividad a internet y la base de datos, así como en la realización de pruebas exhaustivas y el despliegue final de la aplicación.
- **Documentación del proyecto:** Este enfoque garantiza que el desarrollo de la aplicación sea ordenado y avance de manera continua, mejorando constantemente a medida que el proyecto progresa. Además, permite detectar y solucionar problemas rápidamente, fomentando un proceso de retroalimentación constante y asegurar así que el producto final sea de alta calidad y fácil de mantener.

Durante el desarrollo se ha utilizado técnicas y herramientas comunes en la industria de la programación, en particular se ha implementado flujo de trabajo y gestión de versiones llamado *gitflow*. Este método implica hacer commits y push con ayuda de GitHub en una rama de desarrollo llamada *develop* y luego fusionar esa rama con *master* con cada nueva versión o lanzamiento.

## 3.2 Planificación

La planificación del proyecto se dividió en dos fases para asegurarnos de que todo se hiciera de manera organizada y eficiente. En la primera fase se hizo una estimación inicial del proyecto para entender su tamaño y el tiempo necesario para completarlo, incluyendo la documentación necesaria. Esto nos dio una vista general de como de grande y complejo sería el proyecto.

Después se desglosó el proyecto en tareas más específicas siguiendo los incrementos definidos en nuestra metodología iterativa-incremental. Evaluamos cada tarea individualmente y le asignamos un tiempo estimado de desarrollo basado en nuestra estimación inicial.

Toda esta planificación detallada se plasmó en un diagrama de Gantt, que nos da una representación visual clara del calendario del proyecto y muestra las dependencias entre las tareas. Gracias a esto se ha podido mantener el desarrollo del proyecto de manera controlada para alcanzar los objetivos, asegurando que se cumplieran con los puntos clave del proyecto en cada etapa.

### 3.2.1. Estimación por Puntos de Función

Existen multitud de técnicas para calcular de manera fiable el tamaño total de un proyecto de software. En este proyecto se va a hacer uso de unas de las técnicas de estimación estudiadas en el grado: técnica de **Albrecht** usando **puntos de función**[18], ya que permite cuantificar la funcionalidad que se pretende entregar al usuario, independientemente de los recursos empleados para ello.

Este método consta de **tres etapas principales**:

- En la primera se determinan los **componentes funcionales** de software y su complejidad.
- En la segunda se calculan los puntos de función no ajustados.
- En la tercera se hará el ajuste de los puntos función.

En la primera etapa primero hay que identificar los elementos clave para calcular los puntos de función. Empezamos enumerando todos los componentes de cada tipo: entradas externas, salidas externas, grupos lógicos de datos internos, grupos lógicos de datos de interfaz y consultas externas. Luego se evalúa la complejidad de cada uno de estos componentes de forma individual y se utilizan tablas predefinidas que nos indican si un componente es de complejidad alta, complejidad media o baja.

A continuación, se describe brevemente cada componente que se tiene en cuenta y a su vez se realiza la identificación de estos y su complejidad:

- **Entradas externas:** Son todos aquellos grupos de datos o mandatos de control de usuario que entran en la aplicación y añaden o cambian información en un grupo lógico de datos interno. Para el cálculo de su complejidad nos basaremos en la Tabla 3.

Entradas externas	Complejidad
Registrar cuentas de usuario	Media
Crear eventos	Media
Sistema de autenticación	Media
Recuperación de contraseña	Baja
Cerrar sesión de la cuenta	Baja
Eliminar eventos	Baja
Modificar eventos	Media
Eliminar cuentas de usuario	Baja
Ascender rol de un miembro	Media
Retirar rol de un miembro	Media
Modificar perfil de usuario	Baja
Eliminar eventos realizados	Baja
Marcado de asistencias	Media

Tabla 2. Entradas externas

		Tipos de datos elementales		
		1 a 4	5 a 15	16 ó más
Ficheros referenciados	0 ó 1	Baja	Baja	Media
	2	Baja	Media	Alta
	3 ó más	Media	Alta	Alta

Tabla 3. Criterios de evaluación de entradas externas y consultas

- **Salidas externas:** Son todos aquellos grupos de datos o mandatos de control de usuario que salen de la aplicación. Para el cálculo de su complejidad nos basaremos en la Tabla 5.

Salidas externas	Complejidad
Generar PDF con asistidos de un evento realizado	Baja

Tabla 4. Salidas externas

		Tipos de datos elementales		
		1 a 5	6 a 19	20 ó más
Ficheros referenciados	0 ó 1	Baja	Baja	Media
	2 ó 3	Baja	Media	Alta
	4 ó más	Media	Alta	Alta

Tabla 5. Criterios de evaluación de salidas externas

- **Consultas:** Son entradas de usuario que producen inmediatamente una salida, de forma que no se modifican los datos internos. Para el cálculo de su complejidad hay que seguir la misma tabla que la de entradas externas, la Tabla 3.

Consultas	Complejidad
Listado de eventos	Media
Detalles de un evento	Media
Información del perfil	Baja
Listado de usuarios registrados	Baja
Listado de eventos realizados	Baja
Listado de asistidos	Media

Tabla 6. Consultas

- **Ficheros lógicos internos:** Son aquellos grupos lógicos de datos o información de control interna que se generan, son usados y mantiene la aplicación. Para el cálculo de su complejidad nos basaremos en la Tabla 8.

Ficheros lógicos internos	Complejidad
Sesión del usuario	Baja

Tabla 7. Ficheros lógicos internos

		Tipos de datos elementales		
		1 a 19	20 a 50	51 ó más
Ficheros referenciados	0 ó 1	Baja	Baja	Media
	2 a 5	Baja	Media	Alta
	4 ó más	Media	Alta	Alta

Tabla 8. Criterios de evaluación de ficheros lógicos internos y externos.

- **Ficheros lógicos externos:** Son aquellos grupos lógicos de datos o información que el sistema utiliza para el procesamiento de datos. Para el cálculo de su complejidad hay que seguir la misma tabla que la de ficheros lógicos internos, la Tabla 8.

Ficheros lógicos externos	Complejidad
Cloud Firestore Database (Firebase)	Alta
Authentication (Firebase)	Media

Tabla 9. Ficheros lógicos externos

A continuación, después de establecer todas las complejidades requeridas por parte del sistema, es momento de calcular los puntos de función sin ajustar (PFNA) como se muestra en la Tabla 10:

Puntos de Función Sin Ajustar (PFSA)			
Elemento	Complejidad y peso	Número	Total
Entradas	Baja (x3)	6	18
	Media (x4)	7	28
	Alta (x6)	0	0
Salidas	Baja (x4)	1	4
	Media (x5)	0	0
	Alta (x7)	0	0
Consultas	Baja (x3)	3	9
	Media (x4)	3	12
	Alta (x6)	0	0
Ficheros Lógicos Internos	Baja (x7)	0	0
	Media (x10)	1	10
	Alta (x15)	0	0
Ficheros Lógicos Externos	Baja (x5)	0	0
	Media (x7)	1	7
	Alta (x10)	1	10
<b>Total</b>			<b>98</b>

Tabla 10. Correspondencias y puntos de función sin ajustar (PFNA)

Una vez definidos los puntos de función sin ajustar (PFNA) es necesario ajustarlos de usando un Factor de Ajuste (FA). Este Factor de Ajuste depende de catorce características comunes de los sistemas. Cada una de estas características tiene asignado un valor de 0 a

5 en función de la influencia que posea cada una en el desarrollo del proyecto BanDate, que vienen especificados en la Tabla 11:

Factor de Ajuste	Valor
Comunicación de datos	0
Funciones distribuidas	0
Prestaciones	2
Gran USO de la configuración	0
Velocidad de las transacciones	2
Entrada de datos en línea	5
Diseño para la eficiencia del usuario final	2
Actualización de datos en línea	4
Complejidad del proceso lógico interno de la aplicación	3
Reusabilidad del código	3
Facilidad de instalación	1
Facilidad de operación	1
Localizaciones múltiples	0
Facilidad de cambios	2
<b>Total</b>	<b>25</b>

Tabla 11. Factores de complejidad y puntuación

Ahora se va a calcular el factor de ajuste (FA) a partir de la suma de los 14 factores de complejidad (FC) haciendo uso de la siguiente ecuación:

$$FA = 0,65 + (0,01 \cdot FC) = 0,65 + (0,01 \cdot 25) = 0,9$$

Se calculan ahora los Puntos de Función Ajustados (PFA) como el producto de los Puntos de Función Sin Ajustar y el Factor de Ajuste:

$$PFA = PFSA \cdot FA = 98 \cdot 0,9 = 88,2$$

Para finalizar la estimación es necesario calcular la **duración del proyecto**. La aplicación estará diseñada y desarrollada al 100% con JavaScript. Si consultamos las líneas de código por punto de función que corresponden para JavaScript, equivale aproximadamente a **55 LDC/PF**. Con esto, se puede calcular el número total de LDC de la siguiente forma:

$$\text{Total de LDC} = PFA \cdot 55 = 88,2 \cdot 55 = 4851 \text{ LDC}$$

Ahora para estimar el tiempo total necesario para desarrollar el proyecto, hay que calcular el promedio de líneas de código que se pueden escribir por hora. Según la experiencia y habilidades de las que se dispone con el lenguaje JavaScript se estima que se pueden escribir alrededor de **45 líneas de código por hora**. De esta manera, el tiempo total de programación sería el siguiente:

$$\text{Tiempo de programación} = \frac{4851 \text{ LDC}}{45 \text{ LDC/hora}} = 107,8 \text{ horas}$$

Dado que solo hay una persona en el equipo, y que aproximadamente la codificación supondrá el **35%** del esfuerzo total y el resto será tiempo dedicado al análisis, documentación y pruebas, la estimación de la duración del proyecto sería la siguiente:

$$\text{Duración (horas)} = \frac{107,8 \text{ h} \cdot 100}{35} = \mathbf{308 \text{ horas}}$$

### 3.2.2. Desglose temporal

Teniendo en cuenta la estimación realizada en el apartado anterior y estimando que la dedicación al proyecto va a ser una media de 2 horas por día de lunes a viernes, si el proyecto se inicia a mediados de octubre de 2023 se prevé que finalice a principios de junio de 2024, aproximadamente **8 meses**.

Como se mencionó en la sección sobre la metodología utilizada, se ha dividido este proyecto en 9 partes principales. Cada una de estas partes está compuesta por pequeñas tareas (subtareas) que van a añadir nuevas funciones a la solución que estamos desarrollando. A continuación, se va a mostrar en la Figura 13 estas tareas y las relaciones en función de un criterio seguido (en caso de que 2 tareas tengan relación, una tarea no empieza hasta que la otra no termina):

- 1.Fase inicial del proyecto	31 días	lun 16/10/23	mar 28/11/23	
- 1.1 Aprendizaje	25 días	lun 16/10/23	lun 20/11/23	
Tecnologías como React Native, Expo y Firebase	25 días	lun 16/10/23	lun 20/11/23	
- 1.2 Prototipado	16 días	mié 01/11/23	mié 22/11/23	
Diseño funcional y técnico	4 días	mié 01/11/23	lun 06/11/23	11
Creación de la base de datos	4 días	mar 07/11/23	vie 10/11/23	5
Creación de un prototipo	12 días	mar 07/11/23	mié 22/11/23	5
Navegación simple entre pantallas	2 días	lun 13/11/23	mar 14/11/23	6
- 1.3 Análisis	11 días	lun 16/10/23	mar 31/10/23	
Análisis de los requisitos	6 días	lun 16/10/23	lun 23/10/23	
Validación de los requisitos	5 días	mar 24/10/23	mar 31/10/23	10
Especificación funcional	4 días	mar 24/10/23	lun 30/10/23	10
- 1.3 Planificación	6 días	mar 21/11/23	mar 28/11/23	
Planificación del alcance, tiempo y coste	6 días	mar 21/11/23	mar 28/11/23	3
- 2.Primer incremento	8 días	mié 29/11/23	jue 14/12/23	
- 2.1 Gestión de eventos	8 días	mié 29/11/23	jue 14/12/23	
Formulario de creación de eventos	3 días	mié 29/11/23	vie 01/12/23	14
Listado de eventos	3 días	jue 07/12/23	mar 12/12/23	17
Botones de acción como eliminar y editar	2 días	mié 13/12/23	jue 14/12/23	18
Detalles de un evento	2 días	mié 13/12/23	jue 14/12/23	18
- 3.Segundo incremento	9 días	vie 15/12/23	vie 29/12/23	
- 3.1 Gestión de usuarios	6 días	vie 15/12/23	mar 26/12/23	
Formulario de creación de cuentas	4 días	vie 15/12/23	mié 20/12/23	20
Sistema de registro controlado	2 días	jue 21/12/23	mar 26/12/23	23
Gestión de roles	2 días	jue 21/12/23	mar 26/12/23	23
- 3.2 Gestión de extras	2 días	jue 21/12/23	mar 26/12/23	
Formulario de registro de extras	2 días	jue 21/12/23	mar 26/12/23	23
- 3.3 Perfil del usuario	3 días	mié 27/12/23	vie 29/12/23	
Pantalla del perfil	2 días	mié 27/12/23	jue 28/12/23	25
Funcionalidad de editar perfil	1 día	vie 29/12/23	vie 29/12/23	29
- 4.Tercer incremento	7 días	mar 02/01/24	mié 10/01/24	
- 4.1 Sistema de autenticación	4 días	mar 02/01/24	vie 05/01/24	
Pantalla y funcionalidad del login	3 días	mar 02/01/24	jue 04/01/24	30
Funcionalidad de cierre de sesión	1 día	vie 05/01/24	vie 05/01/24	33
- 4.2 Recuperación de contraseña	3 días	lun 08/01/24	mié 10/01/24	
Pantalla y funcionalidad de recuperación de contraseña	3 días	lun 08/01/24	mié 10/01/24	34
- 5.Cuarto incremento	3 días	jue 11/01/24	lun 15/01/24	
- 5.1 Control de roles	3 días	jue 11/01/24	lun 15/01/24	
Control de visualización según el rol	3 días	jue 11/01/24	lun 15/01/24	36
Acceso a funcionalidades según el rol	3 días	jue 11/01/24	lun 15/01/24	36
- 6.Quinto incremento	7 días	mar 16/01/24	mié 24/01/24	
- 6.1 Sistema de marcado de asistencia	7 días	mar 16/01/24	mié 24/01/24	
Componentes específicos según el rol	4 días	mar 16/01/24	vie 19/01/24	40
Funcionalidad de marcar/desmarcar asistencia	3 días	lun 22/01/24	mié 24/01/24	43
- 7.Sexto incremento	16 días	jue 25/01/24	jue 15/02/24	
- 7.1 Listado de usuarios	6 días	jue 25/01/24	jue 01/02/24	
Pantalla con el listado de todos los usuarios registrados	3 días	jue 25/01/24	lun 29/01/24	44
Funcionalidad para eliminar, ascender o descender cuentas de usuario	3 días	mar 30/01/24	jue 01/02/24	47
- 7.2 Listado de eventos realizados	10 días	vie 02/02/24	jue 15/02/24	
Pantalla con el listado de eventos realizados	4 días	vie 02/02/24	mié 07/02/24	48
Ventana con un listado de los asistentes de un evento pasado	4 días	jue 08/02/24	mar 13/02/24	50
Funcionalidad para eliminar eventos	2 días	mié 14/02/24	jue 15/02/24	51
- 8.Séptimo incremento	6 días	vie 16/02/24	vie 23/02/24	
- 8.1 Exportación de PDF	3 días	vie 16/02/24	mar 20/02/24	
Funcionalidad para exportar un archivo PDF con los asistentes de un evento	3 días	vie 16/02/24	mar 20/02/24	52
- 8.2 Sesión en local	3 días	mié 21/02/24	vie 23/02/24	
Implementación de AsyncStorage para guardar la sesión del usuario en local	3 días	mié 21/02/24	vie 23/02/24	55
- 9.Octavo incremento	34 días	lun 26/02/24	vie 12/04/24	
- 9.1 Validaciones y seguridad	14 días	lun 26/02/24	vie 15/03/24	
Implementación de validaciones	4 días	lun 26/02/24	jue 29/02/24	57
Control seguro de datos	4 días	vie 01/03/24	mié 06/03/24	60
Implementación de alertas y mensajes ante posibles errores	6 días	jue 07/03/24	vie 15/03/24	61
- 9.2 Verificar conectividad	6 días	lun 18/03/24	lun 25/03/24	
Funcionalidad de comprobación de conectividad con Firebase	3 días	lun 18/03/24	mié 20/03/24	62
Funcionalidad de comprobación activa de conexión a internet	3 días	jue 21/03/24	lun 25/03/24	64
- 9.3 Pruebas	8 días	mar 26/03/24	jue 04/04/24	
Pruebas exhaustivas en simulador Android	3 días	mar 26/03/24	jue 28/03/24	65
Pruebas exhaustivas en simulador iOS	3 días	mar 26/03/24	jue 28/03/24	65
Corrección de posibles fallos	5 días	vie 29/03/24	jue 04/04/24	68
- 9.4 Despliegue de la app	6 días	vie 05/04/24	vie 12/04/24	
Despliegue en Android	3 días	vie 05/04/24	mar 09/04/24	69
Despliegue en iOS	3 días	mié 10/04/24	vie 12/04/24	71
- 10.Documentación	30 días	lun 15/04/24	vie 24/05/24	
- 10.1 Memoria del trabajo	30 días	lun 15/04/24	vie 24/05/24	
Documentación elaborada de la memoria	22 días	lun 15/04/24	mar 14/05/24	72
Explicación del manual de usuario y despliegue	8 días	mié 15/05/24	vie 24/05/24	75

Figura 13. Tareas con tiempos estimados

Como se puede observar en la anterior figura anterior a su vez se ha estimado el tiempo necesario para completar cada una de estas tareas, y la estimación es casi idéntica a la de la planificación realizada anteriormente ya que el proyecto finaliza cerca de principios de junio, el 24 de mayo. Seguidamente vamos a plasmar en un diagrama de Gantt todo esto, la Figura 14:

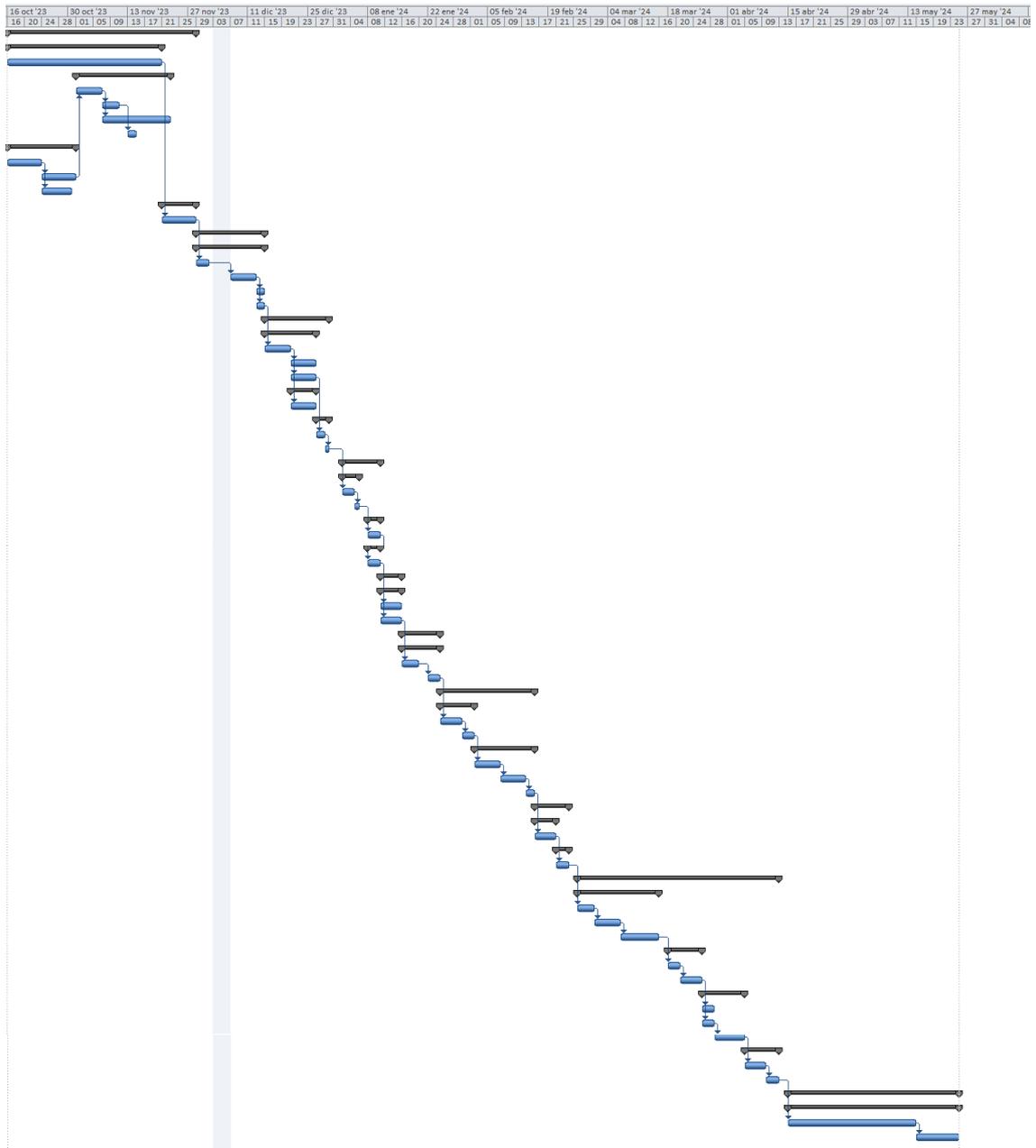


Figura 14. Diagrama de Gantt estimado del proyecto

Además de esto hay que hablar también del calendario seguido, ya que hay ciertos días no laborales que hay que considerar y que se han establecido previamente en el proyecto de MicrosoftProject. Estas excepciones se han cogido de referencia según el Calendario Académico de la Organización Académica del Grado en Ingeniería Informática de Servicios y Aplicaciones para el curso 2023-2024:

Excepciones		Semanas laborales	
	Nombre	Comienzo	Fin
1	San Frutos	25/10/2023	25/10/2023
2	Día no lectivo	04/12/2023	05/12/2023
3	Constitución	06/12/2023	06/12/2023
4	Inmaculada	08/12/2023	08/12/2023
5	Día no lectivo 2	22/12/2023	22/12/2023
6	Navidad	25/12/2023	25/12/2023
7	Noche vieja	31/12/2023	31/12/2023
8	Año nuevo	01/01/2024	01/01/2024
9	Festivo	08/03/2024	08/03/2024
10	Días no lectivos 3	27/05/2024	29/05/2024

Figura 15. Excepciones en el calendario laboral del proyecto estimado

## 3.3 Presupuesto

Para obtener el presupuesto total del proyecto según lo planificado con la metodología por puntos de función que se ha visto anteriormente, se ha dividido en tres partes bien diferenciadas: recursos humanos, recursos materiales y otros gastos.

### 3.3.1. Recursos humanos

En los proyectos de desarrollo software de hoy en día, en la mayoría suelen haber unos roles diferenciados, cada uno con sus funciones:

- El **jefe de proyecto** es el encargado de dirigir el equipo y coordinar los diferentes esfuerzos requeridos.
- El **analista** de datos el que participa en las diferentes fases de análisis de requisitos, diseño y documentación.
- Los **desarrolladores** son los encargados de programar e implementar las diferentes funcionalidades del proyecto.

En el caso de este proyecto únicamente ha participado una persona, ejerciendo los 3 diferentes roles. Por ello, para calcular un salario estimado se ha tomado como referencia el salario medio de un ingeniero junior y el salario medio percibido por un ingeniero senior. El salario de un ingeniero junior es de unos 21.000€ anuales y el salario de un ingeniero senior es de 31.000€ anuales trabajando una jornada laboral completa. Así, se ha establecido que el salario medio por un ingeniero informático en estas condiciones es de 26.000€ anuales en 12 pagas, es decir en 12 meses:

Salario Anual	Salario Mensual
26.000€	2.166,7€

Tabla 12. Coste de personal a jornada completa

No obstante, esto no es así ya que estaba planificado que el personal trabajase una media de 2 horas diarias de lunes a viernes. Entonces, se debe calcular lo que realmente ganaría en un mes bajo estas condiciones.

Si la jornada de trabajo va a ser de unas 2 horas diarias de lunes a viernes, y los días laborables en un mes es de promedio 21 días, nos queda que se realizarían unas 42 horas/mes. Si por el contrario fuera a jornada completa (8 horas diarias) serían 168 horas/mes, por lo que la proporción de horas trabajadas sería:

$$\frac{42 \text{ h/mes}}{168 \text{ h/mes}} = 0,25$$

Entonces nos queda el salario mensual ajustado así:

$$2.166,7\text{€} * 0,25 = 541,67\text{€/mes}$$

Por lo tanto, el salario mensual que realmente ganaría el ingeniero sería de **541,67€**, y según el tiempo estimado calculado sería: **4.333,36€**.

### 3.3.2. Recursos Materiales

En este apartado se detallan los costes de los diferentes recursos materiales utilizados, tanto de la parte hardware como software atendiendo a los factores de impacto correspondientes a la duración del proyecto.

#### Coste hardware:

	Coste (€)	Vida útil (media)	Uso (%)	Coste estimado (€) del tiempo planificado
<b>Ordenador Portátil</b>	1.100€	6 años	18%	198€
<b>Ratón</b>	18,25€	3 años	24%	4,38€
<b>Teclado</b>	24,60€	3 años	24%	5,9€
<b>Monitor</b>	147€	5 años	18%	26,46€
<b>Disco duro Externo</b>	40€	6 años	12%	4,8€
<b>Dispositivo móvil</b>	696€	4 años	35%	234,6€
<b>Total</b>				<b>474,14€</b>

Tabla 13. Total estimado de los costes hardware

#### Coste software:

	Coste (€)	Vida útil (media)	Uso (%)	Coste estimado (€) del tiempo planificado
<b>Licencia Windows 10</b>	13€	Ilimitada	5%	0,65€
<b>Licencia Microsoft Project</b>	19,40€	Ilimitada	5%	4,38€
<b>Licencia office</b>	16€	1 año	65%	10,4€
<b>Total</b>				<b>15,43€</b>

Tabla 14. Total estimado real de los costes software

El coste total de los recursos materiales utilizados sería de **489,57€**.

### 3.3.3. Otros gastos

Estos costes no pueden olvidarse, ya que están ahí, aunque no lo parezca. Son los referidos a servicios básicos como la conexión a Internet, que son imprescindibles para el funcionamiento eficiente del proyecto. A continuación, se muestran en la Tabla 15:

	Coste (€/mes)	Coste estimado (€) del tiempo planificado
<b>Electricidad</b>	30€	240€
<b>Fibra óptica (600 Mbps de subida y bajada)</b>	27,99€	223,92€
<b>Tarifa móvil</b>		
<b>Papel</b>	1,5€	12€
<b>Total</b>		<b>475,92€</b>

Tabla 15. Total estimado de los otros gastos

Una vez obtenido todo el presupuesto desglosado por partes, se puede obtener el presupuesto total estimado del proyecto como se puede ver en la Tabla 16:

Coste total estimado		
Tiempo planificado	Costes	Coste final
8 meses	- Recursos humanos: 4.333,36€ - Recursos materiales: 489,57€ - Otros gastos: 475,92€	<b>5.298,85€</b>

Tabla 16. Coste total estimado del proyecto

## 3.4 Balance final del proyecto

En este apartado se va a hablar lo que ha sucedido en realidad con el desarrollo del proyecto y cómo se ha visto modificado el presupuesto final. A continuación, se va a mostrar en la Figura 16 las tareas con sus duraciones reales, y en la Figura 17 un diagrama de Gantt:

- 1.Fase inicial del proyecto	31 días	lun 16/10/23	mar 28/11/23	
- 1.1 Aprendizaje	25 días	lun 16/10/23	lun 20/11/23	
Tecnologías como React Native, Expo y Firebase	25 días	lun 16/10/23	lun 20/11/23	
- 1.2 Prototipado	16 días	vie 03/11/23	vie 24/11/23	
Diseño funcional y técnico	4 días	vie 03/11/23	mié 08/11/23	11
Creación de la base de datos	6 días	jue 09/11/23	jue 16/11/23	5
Creación de un prototipo	12 días	jue 09/11/23	vie 24/11/23	5
Navegación simple entre pantallas	2 días	vie 17/11/23	lun 20/11/23	6
- 1.3 Análisis	13 días	lun 16/10/23	jue 02/11/23	
Análisis de los requisitos	7 días	lun 16/10/23	mar 24/10/23	
Validación de los requisitos	6 días	jue 26/10/23	jue 02/11/23	10
Especificación funcional	4 días	jue 26/10/23	mar 31/10/23	10
- 1.3 Planificación	6 días	mar 21/11/23	mar 28/11/23	
Planificación del alcance, tiempo y coste	6 días	mar 21/11/23	mar 28/11/23	3
- 2.Primer incremento	10 días	mié 29/11/23	lun 18/12/23	
- 2.1 Gestión de eventos	10 días	mié 29/11/23	lun 18/12/23	
Formulario de creación de eventos	3 días	mié 29/11/23	vie 01/12/23	14
Listado de eventos	5 días	jue 07/12/23	jue 14/12/23	17
Botones de acción como eliminar y editar	2 días	vie 15/12/23	lun 18/12/23	18
Detalles de un evento	2 días	vie 15/12/23	lun 18/12/23	18
- 3.Segundo incremento	10 días	mar 19/12/23	jue 04/01/24	
- 3.1 Gestión de usuarios	7 días	mar 19/12/23	vie 29/12/23	
Formulario de creación de cuentas	4 días	mar 19/12/23	mar 26/12/23	20
Sistema de resgistro controlado	2 días	mié 27/12/23	jue 28/12/23	23
Gestión de roles	3 días	mié 27/12/23	vie 29/12/23	23
- 3.2 Gestión de extras	2 días	mié 27/12/23	jue 28/12/23	
Formulario de registro de extras	2 días	mié 27/12/23	jue 28/12/23	23
- 3.3 Perfil del usuario	3 días	mar 02/01/24	jue 04/01/24	
Pantalla del perfil	2 días	mar 02/01/24	mié 03/01/24	25
Funcionalidad de editar perfil	1 día	jue 04/01/24	jue 04/01/24	29
- 4.Tercer incremento	7 días	vie 05/01/24	mar 06/02/24	
- 4.1 Sistema de autenticación	4 días	vie 05/01/24	jue 01/02/24	
Pantalla y funcionalidad del login	3 días	vie 05/01/24	mié 31/01/24	30
Funcionalidad de cierre de sesión	1 día	jue 01/02/24	jue 01/02/24	33
- 4.2 Recuperación de contraseña	3 días	vie 02/02/24	mar 06/02/24	
Pantalla y funcionalidad de recuperacion de contraseña	3 días	vie 02/02/24	mar 06/02/24	34
- 5.Cuarto incremento	3 días	mié 07/02/24	vie 09/02/24	
- 5.1 Control de roles	3 días	mié 07/02/24	vie 09/02/24	
Control de visualización según el rol	3 días	mié 07/02/24	vie 09/02/24	36
Acceso a funcionalidades según el rol	3 días	mié 07/02/24	vie 09/02/24	36
- 6.Quinto incremento	7 días	lun 12/02/24	mar 20/02/24	
- 6.1 Sistema de marcado de asistencia	7 días	lun 12/02/24	mar 20/02/24	
Componentes específicos según el rol	4 días	lun 12/02/24	jue 15/02/24	40
Funcionalidad de marcar/desmarcar asistencia	3 días	vie 16/02/24	mar 20/02/24	43
- 7.Sexto incremento	16 días	mié 21/02/24	jue 14/03/24	
- 7.1 Listado de usuarios	6 días	mié 21/02/24	mié 28/02/24	
Pantalla con el listado de todos los usuarios registrados	3 días	mié 21/02/24	vie 23/02/24	44
Funcionalidad para eliminar, ascender o descender cuentas de usuario	3 días	lun 26/02/24	mié 28/02/24	47
- 7.2 Listado de eventos realizados	10 días	jue 29/02/24	jue 14/03/24	
Pantalla con el listado de eventos realizados	4 días	jue 29/02/24	mar 05/03/24	48
Ventana con un listado de los asistentes de un evento pasado	4 días	mié 06/03/24	mar 12/03/24	50
Funcionalidad para eliminar eventos	2 días	mié 13/03/24	jue 14/03/24	51
- 8.Séptimo incremento	6 días	vie 15/03/24	vie 22/03/24	
- 8.1 Exportación de PDF	3 días	vie 15/03/24	mar 19/03/24	
Funcionalidad para exportar un archivo PDF con los asistidos de un evento	3 días	vie 15/03/24	mar 19/03/24	52
- 8.2 Sesión en local	3 días	mié 20/03/24	vie 22/03/24	
Implementación de AsyncStorage para guardar la sesión del usuario en local	3 días	mié 20/03/24	vie 22/03/24	55
- 9.Octavo incremento	34 días	lun 25/03/24	jue 09/05/24	
- 9.1 Validaciones y seguridad	14 días	lun 25/03/24	jue 11/04/24	
Implementación de validaciones	4 días	lun 25/03/24	jue 28/03/24	57
Control seguro de datos	4 días	vie 29/03/24	mié 03/04/24	60
Implementación de alertas y mensajes ante posibles errores	6 días	jue 04/04/24	jue 11/04/24	61
- 9.2 Verificar conectividad	6 días	vie 12/04/24	vie 19/04/24	
Funcionalidad de comprobación de conectividad con Firebase	3 días	vie 12/04/24	mar 16/04/24	62
Funcionalidad de comprobación activa de conexión a internet	3 días	mié 17/04/24	vie 19/04/24	64
- 9.3 Pruebas	8 días	lun 22/04/24	mié 01/05/24	
Pruebas exhaustivas en simulador Android	3 días	lun 22/04/24	mié 24/04/24	65
Pruebas exhaustivas en simulador iOS	3 días	lun 22/04/24	mié 24/04/24	65
Corrección de posibles fallos	5 días	jue 25/04/24	mié 01/05/24	68
- 9.4 Despliegue de la app	8 días	jue 02/05/24	lun 13/05/24	
Despliegue en Android	3 días	jue 02/05/24	lun 06/05/24	69
Despliegue en iOS	5 días	mar 07/05/24	lun 13/05/24	71
- 10.Documentación	33 días	mar 14/05/24	mar 09/07/24	
- 10.1 Memoria del trabajo	33 días	mar 14/05/24	mar 09/07/24	
Documentación elaborada de la memoria	25 días	mar 14/05/24	jue 27/06/24	72
Explicación del manual de usuario y despliegue	8 días	vie 28/06/24	mar 09/07/24	75

Figura 16. Tareas con tiempos reales

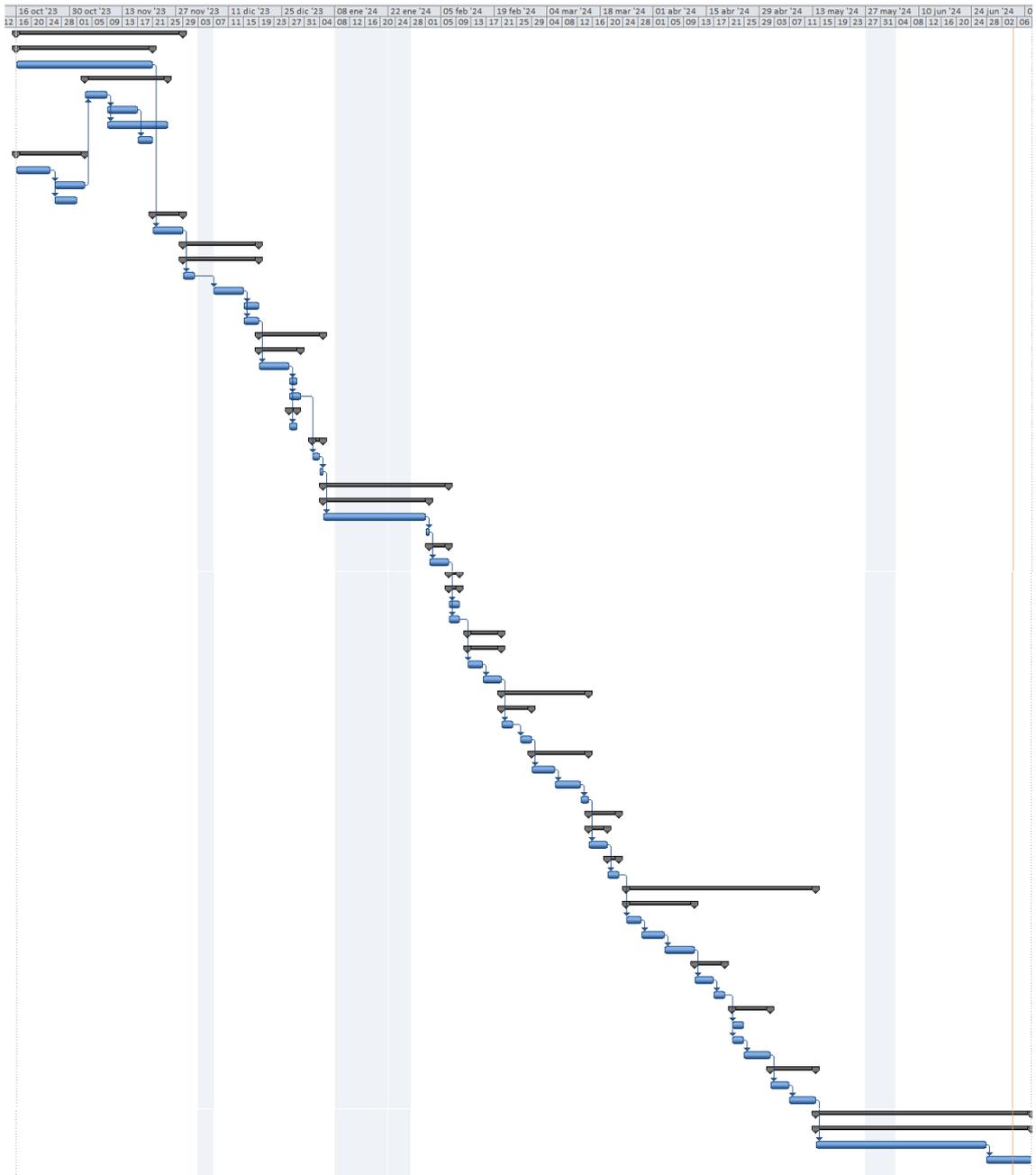


Figura 17. Diagrama de Gantt real del proyecto

Como vimos en el apartado de planificación, en el desglose temporal se estimó que el proyecto terminaría a principios de junio aproximadamente, para ser exactos el **24 de mayo de 2024**. Pero no esto no ha sido así, ya que como se ha podido observar en la Figura 16 y Figura 17 el proyecto ha finalizado realmente el **9 de julio**.

Todo esto es debido a múltiples factores que a continuación se van a explicar:

- **Nuevo calendario laboral:** El calendario laboral estándar para los recursos de este proyecto se ha visto afectado, ya que se tuvo que dedicar tiempo adicional a la preparación de exámenes y para la realización de estos, lo que redujo las horas disponibles para trabajar en el proyecto. El calendario laboral de este proyecto quedó finalmente del a siguiente manera:

	Nombre	Comienzo	Fin
1	San Frutos	25/10/2023	25/10/2023
2	Día no lectivo	04/12/2023	05/12/2023
3	Constitución	06/12/2023	06/12/2023
4	Inmaculada	08/12/2023	08/12/2023
5	Día no lectivo 2	22/12/2023	22/12/2023
6	Navidad	25/12/2023	25/12/2023
7	Noche vieja	31/12/2023	31/12/2023
8	Año nuevo	01/01/2024	01/01/2024
9	Periodo de evaluación ordinario primer cuatrimes	08/01/2024	29/01/2024
10	Festivo	08/03/2024	08/03/2024
11	Días no lectivos 3	27/05/2024	29/05/2024
12	Preparación y realización examen segundo cuatr	30/05/2024	05/06/2024

Figura 18. Excepciones en el calendario laboral del proyecto real

- **Retrasos en las tareas:** El proyecto experimentó varios retrasos en distintas fases y tareas. Esto es debido a que surgieron diversas complicaciones:
  - Tiempo de recolección de requisitos: La fase inicial del proyecto, que involucraba la recolección de requisitos, tomó más tiempo del anticipado. Reunir toda la información necesaria y asegurarse de que los requisitos fueran válidos y precisos fue un proceso más lento de lo esperado.
  - Complicaciones tecnológicas: Durante el desarrollo del proyecto nos encontramos con diversas complicaciones tecnológicas. Algunos de los sistemas y herramientas utilizadas no funcionaron como se esperaba, lo que nos obligó a buscar soluciones alternativas y hacer ajustes que tomaron tiempo adicional como fue la creación de la base de datos y su integración con el código fuente.
  - Bugs y errores inesperados: Como es común en proyectos de desarrollo surgieron bugs y errores inesperados que requirieron atención inmediata y así surgieron más retrasos.
  - Despliegue en iOS: Hubo un retraso significativo al realizar el despliegue en los dispositivos iOS ya que se requería de un Mac, el cual no se disponía en ese momento. Fue necesario solicitar uno prestado, lo que llevó un tiempo adicional y afectó el cronograma previsto. Todo esto finalmente no sirvió de nada ya que no se pudo realizar el despliegue, suponía unos costes elevados innecesarios.
  - Documentación de la memoria: La documentación de la memoria del proyecto también se retrasó debido a su densidad y complejidad. Redactar y revisar toda la información necesaria para una documentación exhaustiva y precisa fue un proceso que requirió más tiempo del planificado.

Con todo esto, el proyecto se ha visto retrasado en 46 días, lo que supone unas 68 horas de retraso trabajando 2 horas diarias de lunes a viernes como se explicó en la sección de planificación. Debido a estas horas de retraso se ha de multiplicar por 0,22 el presupuesto total planificado, pues las 68 horas de esfuerzo no planificado equivalen al 22% del tiempo planificado, lo que es igual a 1.165,75€ extra.

Esto supone que el presupuesto real del proyecto sea de **6.464,6€**.



# Capítulo 4

## Análisis

Este capítulo contiene toda la información obtenida durante la fase de análisis, la cual se enfoca en comprender con mayor precisión los requerimientos del cliente, en este caso de una banda de música. Se describen en primer lugar las características del sistema, los usuarios que interactuarán con la aplicación, los requisitos de usuario y un modelo de Casos de Uso con la especificación de cada uno. Además, se derivan los requisitos funcionales a partir de los requisitos de usuario y se detallan los requisitos no funcionales. Por último, se muestran un modelo Entidad-Relación a modo de representación de los requisitos de información y a posteriori las restricciones del sistema más importantes.

### 4.1 Características del sistema

Las características de un sistema son las funciones esenciales y principales que debe tener para alcanzar los objetivos del proyecto. A continuación, se detallan las características que este proyecto debe cumplir:

- **CAR-01 (Gestión de eventos)**
  - **CAR-01.1 (Crear eventos):** Esta funcionalidad permite la creación de nuevos eventos para la banda musical, proporcionando detalles como la fecha, hora, lugar y descripción del evento.
  - **CAR-01.2 (Ver detalles):** Los usuarios pueden consultar información detallada sobre los nuevos eventos publicados.
  - **CAR-01.3 (Eliminar evento):** Para eliminar eventos que ya no son relevantes o que han sido cancelados.
  - **CAR-01.4 (Editar evento):** Esta función permite modificar los detalles de un evento existente, facilitando la actualización de información importante.
- **CAR-02 (Gestión de usuarios)**
  - **CAR-02.1 (Iniciar sesión):** Los usuarios pueden acceder a la aplicación ingresando su correo electrónico y contraseña.
  - **CAR-02.2 (Cerrar sesión):** Permite a los usuarios salir de su cuenta de manera segura.
  - **CAR-02.3 (Visualizar perfil):** Los usuarios pueden ver su perfil, y su información personal al detalle.
  - **CAR-02.4 (Modificar perfil):** Esta funcionalidad permite a los usuarios actualizar su información personal, como el nombre completo y número de teléfono.

- **CAR-02.5 (Recuperar contraseña):** Permite al usuario ya registrado recuperar las credenciales en caso de que haya olvidado la contraseña.
- **CAR-03 (Asistencias)**
  - **CAR-03.1 (Marcar asistencia):** Los usuarios pueden indicar si asistirán a un evento, lo cual ayuda a la organización a prever el número de participantes.
  - **CAR-03.2 (Eliminar asistencia):** Esta función permite a los usuarios cancelar su asistencia a un evento si ya no pueden asistir.
  - **CAR-03.3 (Generar PDF de asistidos):** Permite generar un documento PDF con la lista de asistentes de un evento pasado, útil para una buena organización.
- **CAR-04 (Gestión de nuevos miembros y extras)**
  - **CAR-04.1 (Dar de alta a nuevos miembros):** Permite registrar nuevos miembros en la banda, asegurando que tengan acceso a unas las funcionalidades específicas de la aplicación.
  - **CAR-04.2 (Dar de alta a un extra):** Funcionalidad para registrar usuarios externos que asistirán a eventos específicos.
  - **CAR-04.3 (Administrar usuarios):** Permite a los administradores gestionar los usuarios registrados, incluyendo dar de baja a cualquier usuario, ascender o revocar el rango de jefe de cuerda de un miembro.
- **CAR-05 (Distintos listados disponibles)**
  - **CAR-05.1 (Consultar listado de usuarios y extras):** Los usuarios pueden ver una lista completa de todos los usuarios y extras registrados en la aplicación.
  - **CAR-05.2 (Consultar listado de eventos):** Los usuarios pueden ver una lista completa de todos los nuevos eventos publicados en la aplicación.
  - **CAR-05.3 (Consultar listado de eventos pasados):** Los usuarios pueden ver una lista completa de todos los eventos realizados en la aplicación.
  - **CAR-05.4 (Consultar asistidos de un evento pasado):** Los organizadores pueden ver una lista de todos los usuarios que han asistieron a un evento ya realizado.

### 4.1.1. Árbol de características

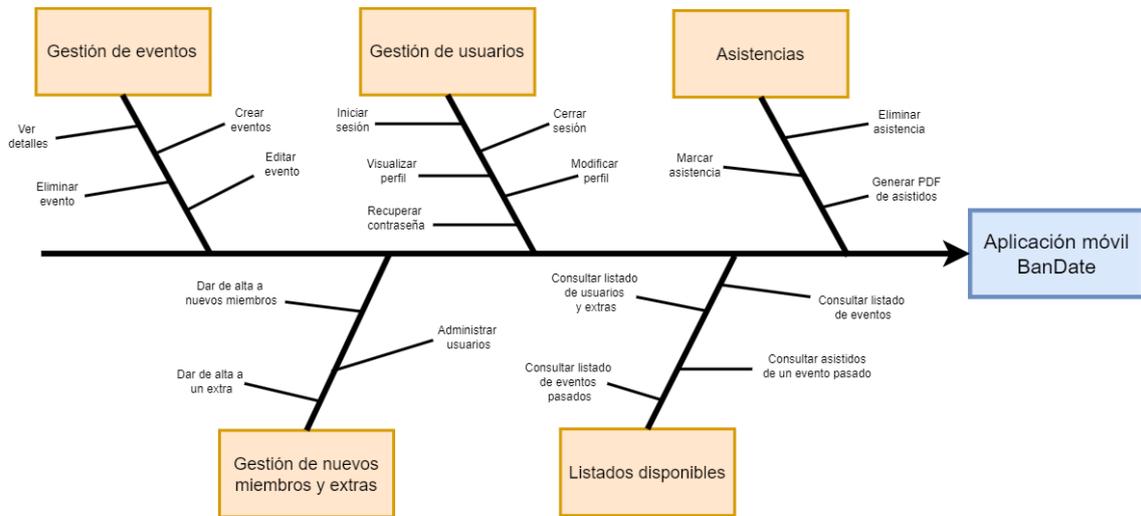


Figura 19. Árbol de características de la aplicación

## 4.2 Actores

Se puede definir como las entidades (personas u otros sistemas externos) que interactúan con el sistema. En este caso estos actores van a ser siempre personas físicas capaces de interactuar con alguna de las pantallas y componentes de la aplicación móvil.

La Figura 20 muestra un diagrama de actores con las relaciones existentes entre ellos.

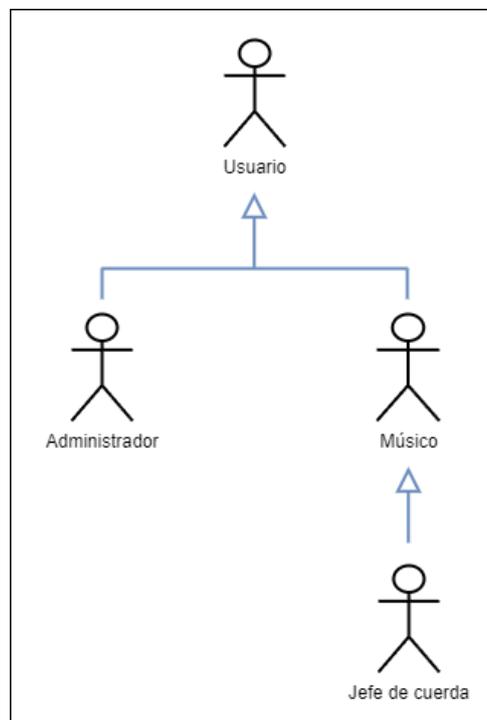


Figura 20. Diagrama de relación entre actores

Estas relaciones tienen un sentido, y es que el actor principal del que todos los otros parten es el **Usuario**. El usuario se refiere a que todos los que estén por debajo de él pueden realizar las mismas funcionalidades básicas de la aplicación, como por ejemplo la más importante, iniciar sesión. El **Administrador** es el actor encargado de gestionar los eventos y usuarios del sistema. El **Músico** es un actor simple que utiliza el sistema para las funcionalidades básicas como marcado de asistencias y visualización de listados. El **Jefe de cuerda** tiene una relación directa con el músico ya que este hereda todo lo que puede realizar, pero aparte tienes otras funciones y más responsabilidades que el propio músico.

Actores		
ID	Nombre	Descripción
A1	Usuario	El usuario describe al conjunto de todos los miembros de la banda, que tras identificarse en el sistema puede hacer uso de la aplicación móvil, ya sea para ver el perfil, modificarlo o cerrar sesión.
A2	Administrador	Este actor es capaz de acceder a la funcionalidad completa de la aplicación, y es el encargado de crear, editar o eliminar los eventos de la banda en la aplicación, obtener un listado con las personas que van a asistir a estos, modificar las asistencias de todos los miembros y administrar las cuentas de los usuarios registrados, además de poder crearlas.
A3	Músico	Es el actor que tiene más restricciones. Es el que utiliza la aplicación móvil para informarse sobre los eventos musicales que se publican, y para marcar asistencia en caso de poder asistir. Podrá visualizar listados, pero no podrá acceder a sus funcionalidades.
A4	Jefe de cuerda	Cada grupo de los distintos instrumentos va a disponer de un jefe, el cual va a poder marcar y eliminar las asistencias de los miembros de su cuerda, y agregar nuevas personas externas al listado de extras si es necesario.

*Tabla 17. Actores*

### 4.3 Requisitos de usuario

Los Requisitos de Usuario son las acciones que un tipo de usuario podrá realizar dentro del sistema y reflejan lo que los usuarios esperan obtener al usar la aplicación.

En esta sección los identificaremos y enumeraremos y para especificarlos utilizaremos los Casos de Uso, ya que estos permiten describir cómo interactúa un usuario con el sistema y también se incluirá varios Diagramas de Casos de Uso divididos por actores. Finalmente, detallaremos las especificaciones de los Casos de Uso para aclarar y definir las interacciones entre los usuarios y el sistema.

<b>Requisitos de Usuario</b>	
<b>CU-01</b>	Iniciar sesión
<b>CU-02</b>	Visualizar perfil
<b>CU-03</b>	Cerrar sesión
<b>CU-04</b>	Modificar perfil
<b>CU-05</b>	Recuperar contraseña
<b>CU-06</b>	Crear evento
<b>CU-07</b>	Consultar nuevos eventos publicados
<b>CU-08</b>	Editar evento
<b>CU-09</b>	Eliminar evento
<b>CU-10</b>	Ver detalles de un evento
<b>CU-11</b>	Marcar cualquier asistencia
<b>CU-12</b>	Eliminar cualquier asistencia
<b>CU-13</b>	Dar de alta a un extra
<b>CU-14</b>	Dar de alta a nuevos miembros de la banda
<b>CU-15</b>	Consultar usuarios de la aplicación
<b>CU-16</b>	Dar de baja a cualquier usuario
<b>CU-17</b>	Ascender músico a jefe de cuerda
<b>CU-18</b>	Quitar rango de jefe de cuerda
<b>CU-19</b>	Consultar todos los eventos realizados
<b>CU-20</b>	Ver asistidos de un evento
<b>CU-21</b>	Generar PDF con los asistidos
<b>CU-22</b>	Marcar asistencia
<b>CU-23</b>	Eliminar asistencia
<b>CU-24</b>	Marcar asistencia de usuarios de su cuerda
<b>CU-25</b>	Eliminar asistencia de usuarios de su cuerda

*Tabla 18. Requisitos de usuario*

### 4.3.1. Diagramas de Casos de Uso

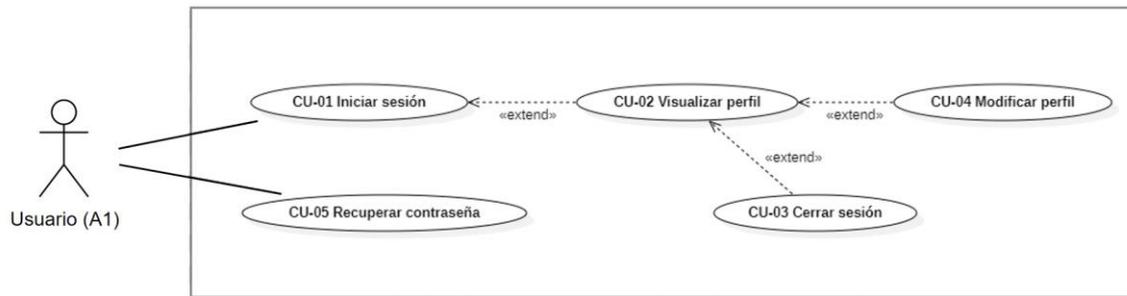


Figura 21. Diagrama de casos de uso del actor Usuario

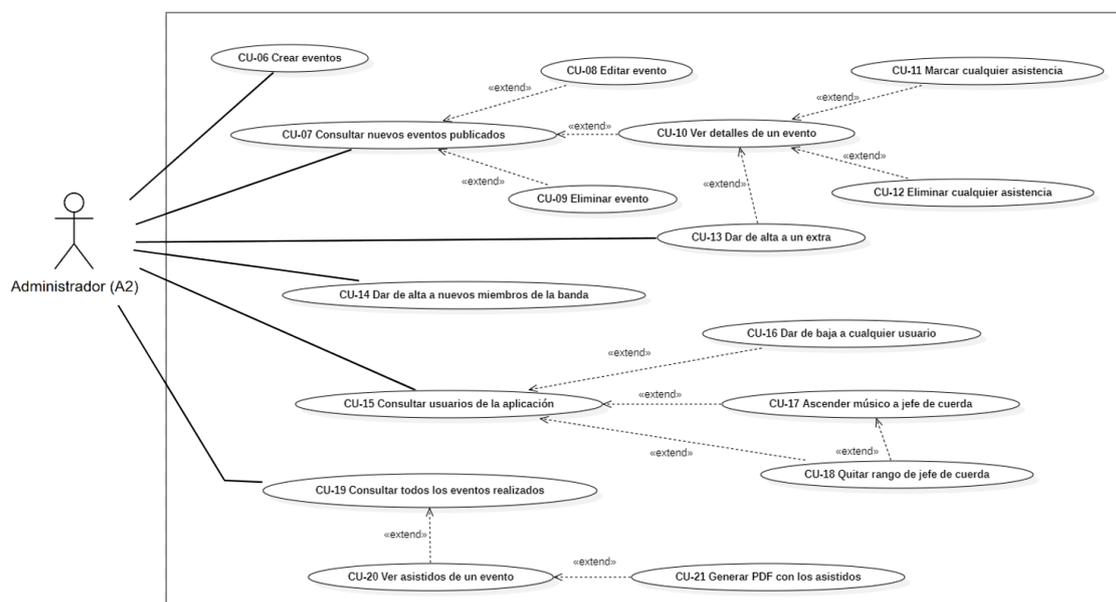


Figura 22. Diagrama de casos de uso del actor Administrador

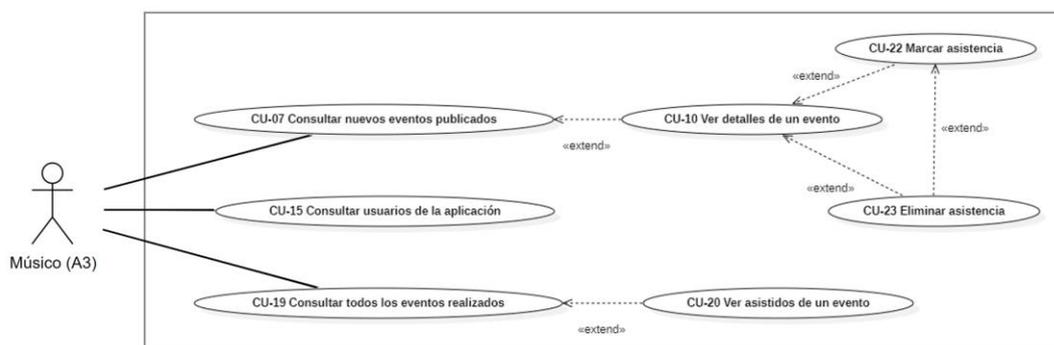


Figura 23. Diagrama de casos de uso del actor Músico

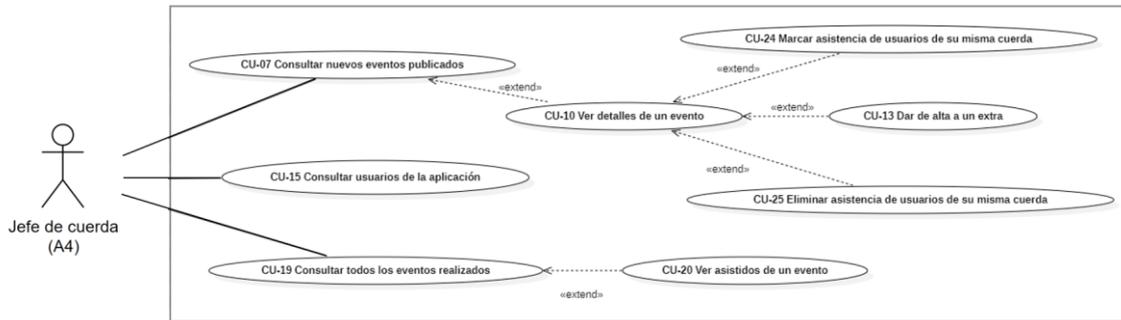


Figura 24. Diagrama de casos de uso del actor Jefe de cuerda

### 4.3.2. Especificación de Casos de Uso

<b>CDU-01</b>	<b>Iniciar sesión</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Usuario (A1)
<b>Descripción</b>	El usuario introduce los datos de su cuenta registrada para acceder a la aplicación.
<b>Precondiciones</b>	El usuario debe tener una cuenta registrada en la base de datos y descargada la aplicación en su dispositivo móvil.
<b>Secuencia Normal</b>	1.- El usuario ingresa su nombre de usuario y contraseña en los campos correspondientes.
	2.- El usuario hace clic en el botón "Iniciar Sesión".
	3.- El sistema valida las credenciales del usuario.
	4.- Si las credenciales son válidas, el sistema redirige al usuario a la página principal de la plataforma.
	5.- Si las credenciales son inválidas, se muestra un mensaje de error indicando que las credenciales ingresadas son incorrectas.
<b>Postcondiciones</b>	El usuario ha iniciado sesión correctamente y puede acceder a las funciones y características de la aplicación.
<b>Excepciones</b>	1. El usuario deja los campos de nombre de usuario o contraseña en blanco. Se muestra un mensaje de error indicando que ambos campos son obligatorios.
	2. El usuario ingresa un nombre de usuario válido pero una contraseña incorrecta. Se muestra un mensaje de error indicando que la contraseña ingresada es incorrecta.
	3. El usuario ingresa un nombre de usuario que no está registrado en la base de datos. Se muestra un mensaje de error indicando que el usuario no existe.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la cantidad de veces que los usuarios necesiten acceder a sus cuentas en la plataforma.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que proporciona acceso seguro a las funciones y datos del usuario en la plataforma.

Tabla 19. Caso de uso "Iniciar sesión"

<b>CDU-02</b>	<b>Visualizar perfil</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Usuario (A1)
<b>Descripción</b>	El usuario accede al menú de perfil para visualizar su perfil de usuario.
<b>Precondiciones</b>	El usuario ha iniciado sesión en la aplicación.
<b>Secuencia Normal</b>	<p>1.- El usuario pulsa sobre el icono de "Perfil" en la parte superior derecha de la aplicación.</p> <p>2.- El sistema carga y muestra la información del perfil del usuario, incluyendo nombre, correo electrónico, teléfono, entre otros detalles.</p> <p>3.- Si lo desea, el usuario puede realizar cambios en su perfil, como actualizar la información de contacto.</p>
<b>Postcondiciones</b>	El usuario ha visualizado con éxito su perfil en la aplicación.
<b>Excepciones</b>	Si hay un problema de conexión o error en el sistema, se muestra un mensaje de error indicando que no se puede acceder al perfil en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del usuario de revisar o actualizar su información en la plataforma.
<b>Importancia</b>	La importancia de este caso de uso es moderada, ya que proporciona al usuario acceso a su información personal y le permite realizar cambios según sus necesidades.

*Tabla 20. Caso de uso "Visualizar perfil"*

<b>CDU-03</b>	<b>Cerrar sesión</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Usuario (A1)
<b>Descripción</b>	El usuario utiliza la función de cerrar su sesión para salir de su cuenta y volver a la pantalla de inicio de sesión.
<b>Precondiciones</b>	El usuario ha iniciado sesión en la aplicación.
<b>Secuencia Normal</b>	<p>1.- El usuario pulsa en el icono de "Perfil" en la parte superior derecha de la aplicación.</p> <p>2.- El sistema carga y muestra la información del perfil del usuario, incluyendo nombre, correo electrónico, teléfono, entre otros detalles.</p> <p>3.- Si lo desea, el usuario puede realizar cambios en su perfil, como actualizar la información de contacto.</p>
<b>Postcondiciones</b>	El usuario ha visualizado con éxito su perfil en la aplicación.
<b>Excepciones</b>	Si hay un problema de conexión o error en el sistema, se muestra un mensaje de error indicando que no se puede acceder al perfil en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del usuario de revisar o actualizar su información en la plataforma.
<b>Importancia</b>	La importancia de este caso de uso es moderada, ya que proporciona al usuario acceso a su información personal y le permite realizar cambios según sus necesidades.

*Tabla 21. Caso de uso "Cerrar sesión "*

<b>CDU-04</b>	<b>Modificar perfil</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Usuario (A1)
<b>Descripción</b>	El usuario modifica la información de su perfil de usuario.
<b>Precondiciones</b>	El usuario ha iniciado sesión en la aplicación y está en la ventana de visualización de su perfil.
<b>Secuencia Normal</b>	1.- El usuario pulsa sobre el icono de "Perfil" en la parte superior derecha de la aplicación.
	2.- El usuario pulsa en la opción "Modificar perfil".
	3.- El sistema carga un formulario prellenado con la información actual del perfil del usuario.
	4.- El usuario modifica la información que desea cambiar, como nombre, dirección de correo electrónico, etc.
	5.- El usuario confirma los cambios pulsando en el botón "Guardar".
	6.- El sistema valida los cambios y actualiza la información del perfil del usuario en la base de datos.
	7.- Se muestra un mensaje indicando que los cambios se han guardado correctamente.
<b>Postcondiciones</b>	La información del perfil del usuario ha sido modificada y actualizada en la plataforma.
<b>Excepciones</b>	1.- Si hay errores en los datos ingresados por el usuario, como un formato de correo electrónico incorrecto, el sistema muestra un mensaje de error indicando los campos que necesitan corrección.
	2.- Si hay un problema de conexión o error en el sistema al intentar guardar los cambios, se muestra un mensaje de error indicando que no se pueden guardar los cambios en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del usuario de actualizar su información en la plataforma.
<b>Importancia</b>	La importancia de este caso de uso es moderada, ya que proporciona al usuario la capacidad de mantener actualizada su información personal en la plataforma según sus necesidades y preferencias.

*Tabla 22. Caso de uso "Modificar perfil"*

<b>CDU-05</b>	<b>Recuperar contraseña</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Usuario (A1)
<b>Descripción</b>	El usuario solicita recuperar su contraseña mediante el ingreso de su correo electrónico registrado.
<b>Precondiciones</b>	El usuario debe tener una cuenta registrada y un correo electrónico asociado en la base de datos.
<b>Secuencia Normal</b>	<p>1.- El usuario pulsa sobre el enlace "¿Olvidaste tu contraseña?" en la pantalla de inicio de sesión.</p> <p>2.- El sistema muestra una pantalla solicitando el ingreso del correo electrónico registrado.</p> <p>3.- El usuario ingresa su correo electrónico y pulsa en el botón "Recuperar contraseña".</p> <p>4.- El sistema verifica que el correo electrónico ingresado esté registrado en la base de datos.</p> <p>5.- Si el correo electrónico es válido, el sistema envía un correo electrónico con un enlace para recuperar la contraseña.</p> <p>6.- El usuario recibe el correo electrónico y pulsa en el enlace proporcionado.</p> <p>7.- El sistema muestra una pantalla para ingresar una nueva contraseña.</p> <p>8.- El usuario ingresa y confirma la nueva contraseña, luego pulsa en el botón "Guardar".</p> <p>9.- El sistema valida la nueva contraseña y actualiza la información en la base de datos.</p> <p>10.- Se muestra un mensaje indicando que la contraseña se ha actualizado correctamente y redirige al usuario a la pantalla de inicio de sesión.</p>
<b>Postcondiciones</b>	La contraseña del usuario ha sido actualizada correctamente, y el usuario puede iniciar sesión con la nueva contraseña.
<b>Excepciones</b>	<p>1.- Si el usuario deja el campo de correo electrónico en blanco, se muestra un mensaje de error indicando que el campo es obligatorio.</p> <p>2.- Si el usuario ingresa un correo electrónico que no está registrado en la base de datos se muestra un mensaje de error indicando que el correo electrónico no está asociado a ninguna cuenta.</p> <p>3.- Si el enlace de recuperación de contraseña ha expirado o ya ha sido utilizado, se muestra un mensaje de error indicando que el enlace no es válido.</p> <p>4.- Si hay un problema de conexión o error en el sistema al intentar guardar la nueva contraseña se muestra un mensaje de error indicando que no se pueden guardar los cambios en este momento.</p>
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la decisión del usuario de dejar de utilizar la plataforma de forma permanente.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que proporciona al usuario la capacidad de cerrar definitivamente su cuenta y eliminar todos sus datos de la plataforma, garantizando su privacidad y seguridad.

Tabla 23. Caso de uso "Recuperar contraseña"

<b>CDU-06</b>	<b>Crear evento</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2)
<b>Descripción</b>	Un administrador puede crear y añadir un nuevo evento para que todos los usuarios de la aplicación puedan verlo.
<b>Precondiciones</b>	El administrador ha iniciado sesión en la plataforma y tiene permisos para crear eventos.
<b>Secuencia Normal</b>	1.- El administrador selecciona la ventana de crear eventos en el menú inferior.
	2.- Se muestra un formulario con campos para ingresar los detalles del evento.
	3.- El administrador completa los campos del formulario con la información del evento.
	4.- El usuario administrador guarda el evento pulsando en el botón "Guardar".
	5.- El sistema valida los cambios y actualiza la información del nuevo evento en la base de datos.
	6.- Se muestra un mensaje indicando que el evento se ha guardado correctamente.
<b>Postcondiciones</b>	El evento se agrega a la lista de eventos por realizar en la plataforma.
<b>Excepciones</b>	1.- El administrador deja campos sin completar. Se muestra un mensaje de error indicando los campos faltantes.
	2.- El formato de los datos ingresados no cumple con las especificaciones requeridas. Se muestra un mensaje de error indicando el formato incorrecto.
	3.- Si hay un problema de conexión o error en el sistema al intentar crear el evento, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la cantidad de eventos que el organizador desee agregar a la plataforma.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite que los organizadores añadan eventos a la plataforma para su promoción y venta de entradas, generando así contenido y atrayendo a los usuarios a la plataforma.

*Tabla 24. Caso de uso "Crear evento"*

<b>CDU-07</b>	<b>Consultar nuevos eventos publicados</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2), Músico (A3), Jefe de cuerda (A4)
<b>Descripción</b>	Un usuario utiliza la ventana de inicio para consultar los eventos más recientemente publicados.
<b>Precondiciones</b>	El usuario ha iniciado sesión en la plataforma y existe al menos un evento publicado en la aplicación.
<b>Secuencia Normal</b>	1.- El usuario accede a la ventana principal de la aplicación
	2.- El sistema muestra una lista de eventos disponibles, ordenados por fecha del evento.
	3.- El usuario puede seleccionar un evento para ver más detalles, marcar asistencia...
<b>Postcondiciones</b>	El usuario ha visualizado con éxito los nuevos eventos publicados.
<b>Excepciones</b>	1.- Si no hay nuevos eventos publicados, se muestra un mensaje indicando que no hay eventos disponibles en este momento.
	2.- Si hay un problema de conexión o error en el sistema al intentar consultar eventos, se muestra un mensaje de error indicando que no se pueden cargar los eventos en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la actividad de publicación de eventos en la plataforma y de la necesidad del usuario de buscar nuevos eventos.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que proporciona a los usuarios acceso a información actualizada sobre eventos disponibles, lo que puede influir en sus decisiones de participar en ellos y en su experiencia general en la plataforma.

*Tabla 25. Caso de uso "Consultar nuevos eventos publicados"*

<b>CDU-08</b>	<b>Editar evento</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2)
<b>Descripción</b>	Un administrador puede editar la información de un evento publicado en la aplicación.
<b>Precondiciones</b>	El usuario ha iniciado sesión en la plataforma y existe al menos un evento publicado en la aplicación.
<b>Secuencia Normal</b>	1.- El usuario accede a la ventana principal de la aplicación
	2.- El sistema muestra una lista de eventos disponibles, ordenados por fecha del evento.
	3.- El usuario pulsa sobre el icono de editar de un evento para modificar los datos.
	4.- El administrador modifica los campos del evento que desea cambiar, como la hora, fecha, lugar...
	5.- El usuario administrador guarda los cambios pulsando en el botón "Guardar".
	6.- El sistema valida los cambios y actualiza la información del evento modificado en la base de datos.
	7.- Se muestra un mensaje indicando que los cambios del evento se han guardado correctamente.
<b>Postcondiciones</b>	La información del evento ha sido modificada y actualizada en la plataforma.
<b>Excepciones</b>	1.- El administrador elimina campos que anteriormente estaban completados. Se muestra un mensaje de error indicando los campos faltantes.
	2.- El formato de los datos ingresados no cumple con las especificaciones requeridas. Se muestra un mensaje de error indicando el formato incorrecto.
	3.- Si hay un problema de conexión o error en el sistema al intentar editar el evento, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la cantidad de eventos que el organizador desee agregar a la plataforma.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite que los organizadores añadan eventos a la plataforma para su promoción y venta de entradas, generando así contenido y atrayendo a los usuarios a la plataforma.

*Tabla 26. Caso de uso "Editar evento"*

<b>CDU-09</b>	<b>Eliminar evento</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2)
<b>Descripción</b>	Un administrador puede eliminar un evento publicado en la aplicación.
<b>Precondiciones</b>	El usuario ha iniciado sesión en la plataforma y existe al menos un evento publicado en la aplicación.
<b>Secuencia Normal</b>	1.- El usuario accede a la ventana principal de la aplicación
	2.- El sistema muestra una lista de eventos disponibles, ordenados por fecha del evento.
	3.- El usuario pulsa sobre el icono de eliminar de un evento para borrar los datos.
	4.- El sistema muestra una confirmación de la acción de eliminación, explicando que la acción es irreversible.
	5.- El usuario administrador confirma el borrado.
	6.- El sistema elimina el evento de la base de datos y lo retira de la lista de eventos disponibles en la plataforma.
	7.- Se muestra un mensaje confirmando que el evento ha sido eliminado exitosamente.
<b>Postcondiciones</b>	El evento ha sido eliminado permanentemente de la plataforma y ya no está disponible para los usuarios.
<b>Excepciones</b>	1.- Si hay un problema de conexión o error en el sistema al intentar eliminar el evento, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del organizador de retirar eventos de la plataforma.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite a los administradores eliminar aquellos eventos que no quieren que el resto de usuarios vean por algún motivo.

*Tabla 27. Caso de uso "Eliminar evento"*

<b>CDU-10</b>	<b>Ver detalles de un evento</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2), Músico (A3), Jefe de cuerda (A4)
<b>Descripción</b>	Un usuario utiliza la plataforma para ver todos los detalles completos de un evento específico.
<b>Precondiciones</b>	El usuario ha iniciado sesión en la plataforma y existe al menos un evento publicado en la aplicación.
<b>Secuencia Normal</b>	<p>1.- El usuario accede a la ventana principal de la aplicación</p> <p>2.- El sistema muestra una lista de eventos disponibles, ordenados por fecha del evento.</p> <p>3.- El usuario pulsa sobre el icono de más detalles de un evento para mostrar más información.</p> <p>4.- El sistema carga y muestra una página con los detalles completos del evento seleccionado, como la fecha, hora, información, y dependiendo del rol del usuario muestra unos botones u otros.</p>
<b>Postcondiciones</b>	El usuario ha visualizado los detalles completos del evento seleccionado.
<b>Excepciones</b>	<p>1.- Si el evento ha sido eliminado o no está disponible, se muestra un mensaje de error indicando que el evento no se puede encontrar.</p> <p>2.- Si hay un problema de conexión o error en el sistema al intentar cargar los detalles del evento, se muestra un mensaje de error indicando que no se pueden cargar los detalles en este momento.</p>
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la curiosidad del usuario y la cantidad de eventos que desee explorar en detalle.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que proporciona a los usuarios la información necesaria para decidir si desean participar en un evento, lo cual es crucial para saber la asistencia.

*Tabla 28. Caso de uso "Ver detalles de un evento"*

<b>CDU-11</b>	<b>Marcar cualquier asistencia</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2)
<b>Descripción</b>	Un administrador puede indicar quién asistirá a un evento específico publicado en la aplicación.
<b>Precondiciones</b>	El administrador ha iniciado sesión en la plataforma y ha seleccionado un evento del cual desea marcar su asistencia o cualquier otra.
<b>Secuencia Normal</b>	<p>1.- El administrador accede a la ventana principal de la aplicación</p> <p>2.- El sistema muestra una lista de eventos disponibles, ordenados por fecha del evento.</p> <p>3.- El administrador pulsa sobre el icono de más detalles de un evento para mostrar más información.</p> <p>4.- El sistema carga y muestra una página con los detalles completos del evento seleccionado y un listado con todos los usuarios de los cuales puede marcar cualquier asistencia.</p> <p>5.- El administrador pulsa sobre el botón "Desapuntado" que aparece al lado de los usuarios que no han marcado su asistencia para ese evento.</p> <p>6.- El sistema registra la asistencia del usuario en la base de datos y se muestra un mensaje confirmando que la asistencia ha sido marcada exitosamente.</p>
<b>Postcondiciones</b>	La asistencia del usuario en un evento ha sido registrada en la plataforma.
<b>Excepciones</b>	1.- Si hay un problema de conexión o error en el sistema al intentar marcar una asistencia, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del administrador de marcar su asistencia o la de otros usuarios a eventos específicos.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite al administrador indicar su participación en eventos y también la de otros usuarios por si se han olvidado de hacerlo, lo cual ayuda a planificar mejor los eventos.

*Tabla 29. Caso de uso "Marcar cualquier asistencia"*

<b>CDU-12</b>	<b>Eliminar cualquier asistencia</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2)
<b>Descripción</b>	Un administrador puede eliminar cualquier asistencia de cualquier usuario registrado en la plataforma de un evento específico (incluyendo la suya).
<b>Precondiciones</b>	El administrador ha iniciado sesión en la plataforma y ha seleccionado un evento del cual desea eliminar su asistencia o cualquier otra, además de que debe de haber registrada alguna asistencia.
<b>Secuencia Normal</b>	<p>1.- El administrador accede a la ventana principal de la aplicación</p> <p>2.- El sistema muestra una lista de eventos disponibles, ordenados por fecha del evento.</p> <p>3.- El administrador pulsa sobre el icono de más detalles de un evento para mostrar más información.</p> <p>4.- El sistema carga y muestra una página con los detalles completos del evento seleccionado y un listado con todos los usuarios de los cuales puede marcar cualquier asistencia.</p> <p>5.- El administrador pulsa sobre el botón "Apuntado" que aparece al lado de cualquier miembro que esté apuntado que va a asistir para ese evento.</p> <p>6.- El sistema muestra una confirmación de la acción de eliminación.</p> <p>7.- El usuario administrador confirma el borrado.</p> <p>8.- El sistema elimina la asistencia del usuario de la base de datos y se muestra un mensaje confirmando que la asistencia ha sido eliminada exitosamente.</p>
<b>Postcondiciones</b>	La asistencia del usuario en un evento ha sido eliminada de la plataforma.
<b>Excepciones</b>	1.- Si hay un problema de conexión o error en el sistema al intentar eliminar la asistencia, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del administrador de cancelar su asistencia o la de otros usuarios a eventos específicos.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite al administrador eliminar su participación en eventos y también la de otros usuarios por si se han olvidado de hacerlo, lo cual ayuda a planificar mejor los eventos.

Tabla 30. Caso de uso "Eliminar cualquier asistencia"

<b>CDU-13</b>	<b>Dar de alta a un extra</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2) y Jefe de cuerda (A4)
<b>Descripción</b>	Un administrador utiliza la plataforma para registrar y dar de alta a un extra, que es un usuario externo a la aplicación y que no puede iniciar sesión ni nada.
<b>Precondiciones</b>	El administrador ha iniciado sesión en la plataforma y ha seleccionado un evento de la pantalla principal del cual desea agregar un nuevo extra que no está registrado, o accede mediante la pantalla de crear cuentas.
<b>Secuencia Normal</b>	<p>1.- El administrador accede a la ventana principal de la aplicación</p> <p>2.- El sistema muestra una lista de eventos disponibles, ordenados por fecha del evento.</p> <p>3.- El administrador pulsa sobre el icono de más detalles de un evento para mostrar más información.</p> <p>4.- El sistema carga y muestra una página con los detalles completos del evento seleccionado y un listado con todos los usuarios, y abajo aparece un botón de "Dar de alta a un nuevo extra".</p> <p>5.- El usuario administrador pulsa sobre el botón y el sistema muestra un modal para rellenar los datos del extra.</p> <p>6.- El administrador completa los campos del formulario con la información del extra y lo registra pulsando en el botón "Registrar extra".</p> <p>7.- El sistema valida los datos ingresados y registra al extra en la base de datos.</p> <p>8.- Se muestra un mensaje indicando que el extra se ha registrado correctamente.</p> <p>9.- Otra forma es que el administrador acceda a la ventana de creación de cuentas de la aplicación</p> <p>10.- Ahí aparece un botón "Dar de alta a un nuevo extra" que si pulsa se sigue la misma secuencia que a partir del paso 5.</p>
<b>Postcondiciones</b>	El extra ha sido registrado y está disponible en la plataforma para los propósitos correspondientes.
<b>Excepciones</b>	<p>1.- El administrador deja campos obligatorios sin completar. Se muestra un mensaje de error indicando los campos faltantes.</p> <p>2.- El formato de los datos ingresados no cumple con las especificaciones requeridas. Se muestra un mensaje de error indicando el formato incorrecto.</p> <p>3.- Si hay un problema de conexión o error en el sistema al intentar dar de alta al extra, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.</p>
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad de la plataforma de registrar nuevos extras.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite a los administradores ingresar en la aplicación nuevos extras para agregarlos a ciertos eventos en los que se necesite personal.

Tabla 31. Caso de uso "Dar de alta a un extra"

<b>CDU-14</b>	<b>Dar de alta a nuevos miembros de la banda</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2)
<b>Descripción</b>	Un administrador utiliza la plataforma para registrar y dar de alta a nuevos miembros de una banda para que puedan usarla ellos también.
<b>Precondiciones</b>	El administrador ha iniciado sesión en la plataforma y ha accedido a la pantalla de crear cuentas.
<b>Secuencia Normal</b>	<p>1.- El usuario accede a la ventana de creación de cuentas de la aplicación.</p> <p>2.- El sistema muestra distintas opciones para que el administrador elija.</p> <p>3.- El usuario administrador pulsa sobre el botón "Dar de alta a un nuevo extra" y el sistema muestra un modal para rellenar los datos del extra.</p> <p>4.- El administrador completa los campos del formulario con la información del extra y lo registra pulsando en el botón "Registrar miembro".</p> <p>5.- El sistema valida los datos ingresados y registra al miembro en la base de datos.</p> <p>6.- Se muestra un mensaje indicando que el usuario se ha registrado correctamente.</p>
<b>Postcondiciones</b>	El usuario ha sido registrado y podrá acceder a la app como miembro de la banda.
<b>Excepciones</b>	<p>1.- El administrador deja campos obligatorios sin completar. Se muestra un mensaje de error indicando los campos faltantes.</p> <p>2.- El formato de los datos ingresados no cumple con las especificaciones requeridas. Se muestra un mensaje de error indicando el formato incorrecto.</p> <p>3.- Si hay un problema de conexión o error en el sistema al intentar dar de alta al nuevo miembro, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.</p>
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad de la plataforma de registrar nuevos miembros.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite a los administradores registrar en la aplicación nuevos miembros de la banda para que puedan acceder a toda la funcionalidad.

Tabla 32. Caso de uso "Dar de alta a nuevos miembros de la banda"

<b>CDU-15</b>	<b>Consultar usuarios de la aplicación</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2), Músico (A3), Jefe de cuerda (A4)
<b>Descripción</b>	Un usuario utiliza la plataforma para consultar la lista de todos los usuarios registrados en la aplicación.
<b>Precondiciones</b>	El usuario ha iniciado sesión en la plataforma.
<b>Secuencia Normal</b>	1.- El usuario accede a la ventana de listados de la aplicación
	2.- El sistema muestra unos botones para elegir.
	3.- El usuario pulsa sobre el botón "Listado de usuarios".
	4.- El sistema muestra una lista de usuarios registrados, que puede incluir detalles como nombre, instrumento, teléfono, etc.
	5.- El usuario administrador puede visualizar unos botones para dar rangos.
<b>Postcondiciones</b>	El usuario ha consultado y obtenido la información de los usuarios registrados en la plataforma.
<b>Excepciones</b>	1.- Si hay un problema de conexión o error en el sistema al intentar consultar la lista de usuarios, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del usuario de revisar la lista de usuarios de la plataforma.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite a los usuarios ver y mantener un control y gestión de los usuarios registrados.

Tabla 33. Caso de uso "Consultar usuarios de la aplicación"

<b>CDU-16</b>	<b>Dar de baja a cualquier usuario</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2)
<b>Descripción</b>	Un administrador puede editar la información de un evento publicado en la aplicación.
<b>Precondiciones</b>	El administrador ha iniciado sesión en la plataforma y ha accedido a la pantalla de listados.
<b>Secuencia Normal</b>	1.- El administrador accede a la ventana de listados de la aplicación
	2.- El administrador pulsa sobre el botón "Listado de usuarios".
	3.- El administrador busca al usuario que desea eliminar y pulsa sobre el botón que aparece al lado "Dar de baja".
	4.- El sistema solicita una confirmación de la acción de dar de baja, explicando las consecuencias de esta acción.
	6.- El administrador confirma la acción de dar de baja.
	7.- El sistema procede a eliminar la cuenta del usuario y actualiza la base de datos.
	8.- Se muestra un mensaje confirmando que el usuario ha sido dado de baja exitosamente.
	<b>Postcondiciones</b>
<b>Excepciones</b>	1.- Si hay un problema de conexión o error en el sistema al intentar dar de baja al usuario, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del administrador de gestionar y mantener la base de usuarios, eliminando cuentas según sea necesario.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite a los administradores gestionar eficazmente la base de usuarios de la plataforma, asegurando que solo los usuarios autorizados y activos tengan acceso y que se mantenga la integridad y seguridad de la plataforma.

Tabla 34. Caso de uso "Dar de baja a cualquier usuario"

<b>CDU-17</b>	<b>Ascender músico a jefe de cuerda</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2)
<b>Descripción</b>	Un administrador utiliza la plataforma para ascender a un músico registrado al rol de "Jefe de Cuerda" en la aplicación.
<b>Precondiciones</b>	El administrador ha iniciado sesión en la plataforma y el músico debe estar registrado en la plataforma como miembro de la banda.
<b>Secuencia Normal</b>	1.- El administrador accede a la ventana de listados de la aplicación
	2.- El administrador pulsa sobre el botón "Listado de usuarios".
	3.- El administrador busca al usuario que desea ascender y pulsa sobre el botón que aparece al lado "Ascender a jefe de cuerda".
	4.- El sistema solicita una confirmación de la acción, explicando las consecuencias de esta acción.
	6.- El administrador confirma la acción de ascender a jefe de cuerda.
	7.- El sistema procede a la actualización del usuario y actualiza la base de datos.
	8.- Se muestra un mensaje confirmando que el usuario ha sido ascendido exitosamente.
	<b>Postcondiciones</b>
<b>Excepciones</b>	1.- Si hay un problema de conexión o error en el sistema al intentar ascender al músico, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad de la banda de asignar o cambiar jefes de cuerda.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite a los administradores de la banda gestionar de manera efectiva la estructura organizativa y asignar roles clave que son esenciales para el funcionamiento y liderazgo dentro de la banda. Además de que significa otorgar más privilegios a ese usuario ascendido.

Tabla 35. Caso de uso "Ascender músico a jefe de cuerda"

<b>CDU-18</b>	<b>Quitar rango de jefe de cuerda</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2)
<b>Descripción</b>	Un administrador utiliza la plataforma para quitar a un músico registrado el rol de "Jefe de Cuerda" en la aplicación para que pase a ser músico.
<b>Precondiciones</b>	El administrador ha iniciado sesión en la plataforma y el jefe de cuerda debe estar registrado en la plataforma.
<b>Secuencia Normal</b>	<p>1.- El administrador accede a la ventana de listados de la aplicación</p> <p>2.- El administrador pulsa sobre el botón "Listado de usuarios".</p> <p>3.- El administrador busca al usuario que desea descender y pulsa sobre el botón que aparece al lado "Quitar rango de jefe de cuerda".</p> <p>4.- El sistema solicita una confirmación de la acción, explicando las consecuencias de esta acción.</p> <p>6.- El administrador confirma la acción de descender al jefe de cuerda.</p> <p>7.- El sistema procede a la actualización del usuario y actualiza la base de datos.</p> <p>8.- Se muestra un mensaje confirmando que el usuario ha sido descendido exitosamente.</p>
<b>Postcondiciones</b>	La cuenta del usuario ha sido descendida en la plataforma, y el usuario tiene menos permisos de los que ya tenía.
<b>Excepciones</b>	1.- Si hay un problema de conexión o error en el sistema al intentar descender al usuario, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad de la banda de descender o cambiar los jefes de cuerda.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite a los administradores de la banda gestionar de manera efectiva la estructura organizativa y asignar roles clave que son esenciales para el funcionamiento y liderazgo dentro de la banda. Además de que significa retirar privilegios a ese usuario descendido.

Tabla 36. Caso de uso "Quitar rango de jefe de cuerda"

<b>CDU-19</b>	<b>Consultar todos los eventos realizados</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2), Músico (A3), Jefe de cuerda (A4)
<b>Descripción</b>	Un usuario utiliza la plataforma para consultar la lista de todos los eventos realizados en la aplicación.
<b>Precondiciones</b>	El usuario ha iniciado sesión en la plataforma.
<b>Secuencia Normal</b>	1.- El usuario accede a la ventana de listados de la aplicación
	2.- El sistema muestra unos botones para elegir.
	3.- El usuario pulsa sobre el botón "Listado de eventos".
	4.- El sistema muestra una lista de eventos realizados, que puede incluir detalles como fecha, hora, lugar, etc.
	5.- El sistema muestra al lado de cada evento realizado un botón "Mostrar asistidos", que si lo pulsas se muestra pantalla con un listado de los usuarios y extras que asistieron.
	6.- El usuario administrador puede visualizar un botón para exportar un PDF con los asistidos.
<b>Postcondiciones</b>	El usuario ha consultado y obtenido la información de los realizados en la plataforma.
<b>Excepciones</b>	1.- Si hay un problema de conexión o error en el sistema al intentar consultar la lista de eventos, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del usuario de consultar la lista de eventos realizados de la plataforma.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite a los usuarios ver y mantener un control y gestión de los eventos realizados con sus asistencias.

Tabla 37. Caso de uso "Consultar todos los eventos publicados"

<b>CDU-20</b>	<b>Ver asistidos de un evento</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2), Músico (A3), Jefe de cuerda (A4)
<b>Descripción</b>	Un usuario utiliza la plataforma para consultar una lista con todos los asistidos de un determinado evento realizado en la aplicación.
<b>Precondiciones</b>	El usuario ha iniciado sesión en la plataforma y ha accedido a la ventana de listados de la aplicación
<b>Secuencia Normal</b>	1.- El usuario pulsa sobre el botón "Listado de eventos".
	2.- El sistema muestra una lista de eventos realizados, que puede incluir detalles como fecha, hora, lugar, etc.
	3.- El sistema muestra al lado de cada evento realizado un botón "Mostrar asistidos".
	4.- El usuario pulsa sobre ese botón y se muestra una pantalla con un listado de usuarios y extras que asistieron a ese evento.
<b>Postcondiciones</b>	El usuario ha consultado la lista de asistentes del evento.
<b>Excepciones</b>	1.- Si no hay asistentes registrados para el evento, se muestra un mensaje indicando que no hay asistentes en este momento.
	2.- Si hay un problema de conexión o error en el sistema al intentar cargar la lista de asistentes, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del usuario de conocer quiénes asistirán al evento.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que proporciona a los usuarios información valiosa sobre la participación en eventos y poder gestionar mejor el control de los pagos.

Tabla 38. Caso de uso "Ver asistidos de un evento"

<b>CDU-21</b>	<b>Generar PDF con los asistidos</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Administrador (A2)
<b>Descripción</b>	Un administrador utiliza la plataforma para generar un documento PDF que contiene la lista de asistentes a un evento específico.
<b>Precondiciones</b>	El administrador ha iniciado sesión en la plataforma y ha accedido a la ventana de listados de eventos y ha pulsado sobre el botón "Mostrar asistidos" de un evento realizado específico.
<b>Secuencia Normal</b>	1.- El administrador visualiza un listado con los asistentes de un evento realizado, y debajo un botón que dice "Generar PDF".
	2.- El administrador pulsa sobre el botón.
	3.- El sistema recopila la lista de asistentes registrados para el evento, incluyendo información relevante, y genera un documento PDF.
	4.- El sistema muestra un mensaje confirmando que el PDF ha sido generado y descargado exitosamente.
<b>Postcondiciones</b>	El administrador ha generado y descargado un documento PDF con la lista de asistentes de un evento.
<b>Excepciones</b>	1.- Si el evento no tiene asistentes registrados, se muestra un mensaje indicando que no hay asistentes disponibles para generar el PDF.
	2.- Si hay un problema de conexión o error en el sistema al intentar generar el PDF, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del administrador de generar reportes de asistencia en formato PDF.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite a los administradores generar reportes formales de asistencia, que pueden ser útiles para la planificación, documentación, y análisis de eventos.

*Tabla 39. Caso de uso "Generar PDF con los asistidos"*

<b>CDU-22</b>	<b>Marcar asistencia</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Músico (A3)
<b>Descripción</b>	Un músico utiliza la plataforma para indicar su asistencia a un ensayo, concierto, o cualquier otro evento relacionado con la banda.
<b>Precondiciones</b>	El músico ha iniciado sesión en la plataforma y ha seleccionado un evento del cual desea marcar su asistencia.
<b>Secuencia Normal</b>	1.- El músico accede a la ventana principal de la aplicación
	2.- El sistema muestra una lista de eventos disponibles, ordenados por fecha del evento.
	3.- El músico pulsa sobre el icono de más detalles de un evento para mostrar más información.
	4.- El sistema carga y muestra una página con los detalles completos del evento seleccionado y un listado con todos los usuarios que van a asistir.
	5.- El músico pulsa sobre el botón "Voy a asistir" que aparece debajo del listado de los usuarios que van a asistir.
	6.- El sistema registra la asistencia del usuario en la base de datos y se muestra un mensaje confirmando que la asistencia ha sido marcada exitosamente.
<b>Postcondiciones</b>	La asistencia del músico en un evento ha sido registrada en la plataforma.
<b>Excepciones</b>	1.- Si hay un problema de conexión o error en el sistema al intentar marcar asistencia, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la cantidad de eventos de la banda a los que el músico desee asistir.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite a los músicos gestionar su participación en ensayos, conciertos y otros eventos de la banda, facilitando la organización y la planificación por parte de los administradores de la banda.

*Tabla 40. Caso de uso "Marcar asistencia"*

<b>CDU-23</b>	<b>Eliminar asistencia</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Músico (A3)
<b>Descripción</b>	Un músico utiliza la plataforma para cancelar su asistencia previamente marcada a un ensayo, concierto, o cualquier otro evento relacionado con la banda.
<b>Precondiciones</b>	El músico ha iniciado sesión en la plataforma y ha marcado previamente su asistencia a un evento específico.
<b>Secuencia Normal</b>	1.- El músico accede a la ventana principal de la aplicación
	2.- El sistema muestra una lista de eventos disponibles, ordenados por fecha del evento.
	3.- El músico pulsa sobre el icono de más detalles de un evento para mostrar más información.
	4.- El sistema carga y muestra una página con los detalles completos del evento seleccionado y un listado con todos los usuarios que van a asistir.
	5.- El músico pulsa sobre el botón "Eliminar asistencia" que aparece debajo del listado de los usuarios que van a asistir.
	6.- El sistema actualiza la base de datos eliminando la asistencia del músico a ese evento y se muestra un mensaje confirmando que la asistencia ha sido borrada exitosamente.
<b>Postcondiciones</b>	La asistencia del músico en un evento ha sido eliminada de la plataforma.
<b>Excepciones</b>	1.- Si hay un problema de conexión o error en el sistema al intentar eliminar asistencia, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del músico de cancelar su asistencia a eventos específicos.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite a los músicos gestionar de manera flexible su participación en eventos de la banda, ayudando a los administradores a tener una visión más precisa de la asistencia esperada y mejorando la organización de la banda.

*Tabla 41. Caso de uso "Eliminar asistencia"*

<b>CDU-24</b>	<b>Marcar asistencia de usuarios de su misma cuerda</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Jefe de cuerda (A4)
<b>Descripción</b>	Un jefe de cuerda puede indicar quién asistirá de su misma cuerda (mismo instrumento) a un evento específico publicado en la aplicación.
<b>Precondiciones</b>	El jefe de cuerda ha iniciado sesión en la plataforma y ha seleccionado un evento del cual desea marcar su asistencia o cualquier otra.
<b>Secuencia Normal</b>	<p>1.- El jefe de cuerda accede a la ventana principal de la aplicación</p> <p>2.- El sistema muestra una lista de eventos disponibles, ordenados por fecha del evento.</p> <p>3.- El jefe de cuerda pulsa sobre el icono de más detalles de un evento para mostrar más información.</p> <p>4.- El sistema carga y muestra una página con los detalles completos del evento seleccionado y un listado con todos los usuarios de los cuales puede marcar cualquier asistencia de los que tengan registrado el mismo instrumento que el jefe de cuerda.</p> <p>5.- El jefe de cuerda pulsa sobre el botón "Desapuntado" que aparece al lado de los usuarios de su cuerda que no han marcado su asistencia para ese evento, o sobre el mismo.</p> <p>6.- El sistema registra la asistencia del usuario en la base de datos y se muestra un mensaje confirmando que la asistencia ha sido marcada exitosamente.</p>
<b>Postcondiciones</b>	La asistencia del usuario en un evento ha sido registrada en la plataforma.
<b>Excepciones</b>	1.- Si hay un problema de conexión o error en el sistema al intentar marcar una asistencia, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del jefe de cuerda de marcar su asistencia o la de otros usuarios de su misma cuerda a eventos específicos.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite al jefe de cuerda indicar su participación en eventos y también la de otros usuarios de su misma cuerda por si se han olvidado de hacerlo, lo cual ayuda a planificar mejor los eventos.

Tabla 42. Caso de uso "Marcar asistencia de usuarios de su misma cuerda"

<b>CDU-25</b>	<b>Eliminar asistencia de usuarios de su misma cuerda</b>
<b>Versión</b>	1.0.0
<b>Actor</b>	Jefe de cuerda (A4)
<b>Descripción</b>	Un jefe de cuerda puede eliminar cualquier asistencia de cualquier usuario de su misma cuerda registrado en la plataforma de un evento específico (incluyendo la suya).
<b>Precondiciones</b>	El jefe de cuerda ha iniciado sesión en la plataforma y ha seleccionado un evento del cual desea eliminar su asistencia o cualquier otra algún usuario de su misma cuerda.
<b>Secuencia Normal</b>	<p>1.- El jefe de cuerda accede a la ventana principal de la aplicación</p> <p>2.- El sistema muestra una lista de eventos disponibles, ordenados por fecha del evento.</p> <p>3.- El jefe de cuerda pulsa sobre el icono de más detalles de un evento para mostrar más información.</p> <p>4.- El sistema carga y muestra una página con los detalles completos del evento seleccionado y un listado con todos los usuarios de los cuales puede eliminar cualquier asistencia de aquellos de su misma cuerda.</p> <p>5.- El jefe de cuerda pulsa sobre el botón "Apuntado" que aparece al lado de los usuarios de su cuerda que estén apuntados que van a asistir para ese evento, o sobre el mismo.</p> <p>6.- El sistema muestra una confirmación de la acción de eliminación.</p> <p>7.- El jefe de cuerda confirma el borrado.</p> <p>8.- El sistema elimina la asistencia del usuario de la base de datos y se muestra un mensaje confirmando que la asistencia ha sido eliminada exitosamente.</p>
<b>Postcondiciones</b>	La asistencia del usuario en un evento ha sido eliminada de la plataforma.
<b>Excepciones</b>	1.- Si hay un problema de conexión o error en el sistema al intentar eliminar la asistencia, se muestra un mensaje de error indicando que no se puede realizar la acción en este momento.
<b>Frecuencia</b>	La frecuencia de este caso de uso depende de la necesidad del jefe de cuerda de cancelar su asistencia o la de otros usuarios de su misma cuerda a eventos específicos.
<b>Importancia</b>	La importancia de este caso de uso es alta, ya que permite al jefe de cuerda eliminar su participación en eventos y también la de otros usuarios de su misma cuerda por si se han olvidado de hacerlo y no pueden asistir finalmente, lo cual ayuda a planificar mejor los eventos.

Tabla 43. Caso de uso "Eliminar asistencia de usuarios de su misma cuerda"

## 4.4 Requisitos funcionales

Aquí se van a detallar en tablas los Requisitos Funcionales, que indican cómo debe comportarse el sistema bajo determinadas situaciones:

**RF-01:** El sistema debe permitir a los usuarios iniciar sesión para acceder a la aplicación.

**RF-02:** El sistema debe validar las credenciales del usuario al iniciar sesión para garantizar la existencia de la cuenta y la precisión de los datos ingresados.

**RF-03:** El sistema debe proporcionar mensajes de error claros en caso de credenciales incorrectas o datos inválidos.

**RF-04:** El sistema debe permitir a los usuarios modificar su perfil según sea necesario.

**RF-05:** El sistema debe permitir al administrador crear nuevos eventos.

**RF-06:** El sistema debe validar los datos ingresados al crear un nuevo evento, asegurando que se proporcionen todos los campos obligatorios y que sean válidos.

**RF-07:** El sistema debe permitir al administrador editar la información de los eventos existentes, y verificar que la información de los eventos se actualice correctamente.

**RF-08:** El sistema debe permitir al administrador eliminar eventos según sea necesario.

**RF-09:** El sistema debe permitir al administrador marcar o eliminar cualquier asistencia de los participantes a los eventos, asegurando que se registren correctamente y se apliquen las actualizaciones necesarias en la base de datos.

**RF-10:** El sistema debe permitir al administrador crear nuevas cuentas de usuario según sea necesario.

**RF-11:** El sistema debe verificar la validez de los datos ingresados al crear nuevas cuentas de usuario, asegurando que se proporcionen todos los campos requeridos y que los datos sean válidos.

**RF-12:** El sistema debe permitir al administrador asignar o revocar el rango de jefe de cuerda a usuarios específicos.

**RF-13:** El sistema debe permitir al administrador ver la lista de asistentes a un evento específico.

**RF-14:** El sistema debe proporcionar la funcionalidad para que el administrador exporte la información de un evento en formato PDF.

**RF-15:** El sistema debe permitir al usuario consultar los eventos publicados.

**RF-16:** El sistema debe permitir al jefe de cuerda marcar o eliminar la asistencia de los músicos de su cuerda, asegurando que solo pueda modificar la asistencia de los músicos de su cuerda y que se registren correctamente en el sistema.

**RF-17:** El sistema debe permitir a los usuarios ver detalles específicos de los eventos.

**RF-18:** El sistema debe permitir al músico la marcación o eliminación de su propia asistencia a eventos, asegurando que solo pueda modificar su propia asistencia y que se registre correctamente en el sistema.

**RF-19:** El sistema debe permitir a los administradores y jefes de cuerda registrar nuevos extras al sistema.

**RF-21:** El sistema debe permitir que todos los usuarios puedan ver el listado de usuarios registrados en el sistema.

**RF-22:** El sistema debe permitir que todos los usuarios pueden ver el listado de eventos ya realizados.

**RF-23:** El sistema debe permitir que los usuarios puedan cerrar sesión en el sistema.

## 4.5 Requisitos no funcionales

Los requisitos no funcionales son aquellas restricciones del sistema que afectan a su desarrollo o funcionamiento. Aquí se detallan algunos tipos distintos de requisitos no funcionales:

### 4.5.1. Integridad

La integridad describe la capacidad del sistema para evitar la corrupción y la pérdida de los datos que maneja:

ID	Requisitos de integridad
INT-01	El sistema debe garantizar la integridad de los datos de los eventos, evitando modificaciones no autorizadas o pérdida de información.
INT-02	El sistema debe asegurar la integridad de los datos de los usuarios registrados, protegiendo su información personal y evitando alteraciones no autorizadas.
INT-03	El sistema debe realizar validaciones de integridad en los datos ingresados por los usuarios al registrar eventos, asegurando que cumplan con las reglas y restricciones establecidas.

*Tabla 44. Requisitos no funcionales de integridad*

### 4.5.2. Robustez

La robustez mide la capacidad del sistema para funcionar correctamente en situaciones anómalas:

ID	Requisitos de robustez
ROB-01	El sistema debe ser capaz de manejar situaciones anómalas de manera adecuada y recuperarse de ellas sin comprometer su funcionalidad.
ROB-02	Se deben implementar mecanismos de tolerancia a fallos para garantizar la continuidad del servicio en caso de errores o interrupciones.
ROB-03	El sistema debe ser capaz de manejar cargas de trabajo elevadas sin degradar su rendimiento o provocar errores.

*Tabla 45. Requisitos no funcionales de robustez*

### 4.5.3. Seguridad

La seguridad asegura la protección del sistema frente a ataques maliciosos:

ID	Requisitos de seguridad
SEG-01	La plataforma debe implementar medidas de seguridad adecuadas para proteger los datos de los usuarios
SEG-02	Se debe implementar un sistema de autenticación y autorización basado en roles para garantizar el acceso adecuado a las diferentes funcionalidades de la aplicación

*Tabla 46. Requisitos no funcionales de seguridad*

#### 4.5.4. Usabilidad

La usabilidad comprende todos aquellos factores que hacen que la aplicación sea friendly o fácil de usar:

ID	Requisitos de usabilidad
USAB-01	El sistema debe contar con una interfaz de usuario atractiva y fácil de usar, que proporcione una experiencia intuitiva para los usuarios.
USAB-02	Las funcionalidades principales deben ser accesibles de forma clara y sencilla, con una navegación fluida y coherente.

*Tabla 47. Requisitos no funcionales de usabilidad*

#### 4.5.5. Disponibilidad

La disponibilidad da una medida del tiempo en el que los servicios del sistema están operativos:

ID	Requisitos de disponibilidad
DISP-01	El sistema debe estar disponible para su acceso y uso en la mayor parte del tiempo, con un tiempo de inactividad planificado mínimo para tareas de mantenimiento.
DISP-02	Se deben implementar mecanismos de equilibrio de carga para distribuir la carga de trabajo de manera eficiente y evitar puntos de congestión.

*Tabla 48. Requisitos no funcionales de disponibilidad*

#### 4.5.6. Rendimiento

El rendimiento mide la respuesta del sistema ante las solicitudes y acciones de los usuarios:

ID	Requisitos de rendimiento
REND-01	El sistema debe ser capaz de responder a las solicitudes de los usuarios en un tiempo razonable, idealmente en un máximo de 10 segundos.
REND-02	Se deben realizar pruebas de carga y rendimiento para verificar que el sistema pueda manejar la carga esperada sin degradar significativamente su rendimiento.
REND-03	El sistema debe ser capaz de manejar un gran número de usuarios concurrentes sin degradar significativamente su rendimiento.

*Tabla 49. Requisitos no funcionales de rendimiento*

### 4.5.7. Escalabilidad

La escalabilidad determina la capacidad del sistema para gestionar un número creciente de usuarios, datos, transacciones, servidores...

ID	Requisitos de escalabilidad
ESC-01	El sistema debe ser diseñado y desarrollado de manera que pueda manejar un incremento de demanda a medio plazo sin comprometer su funcionalidad o rendimiento.
ESC-02	Se deben utilizar tecnologías y arquitecturas escalables que permitan la adición de recursos y la distribución de la carga de trabajo según sea necesario.

*Tabla 50. Requisitos no funcionales de escalabilidad*

### 4.5.8. Portabilidad

La portabilidad cuantifica el esfuerzo necesario para migrar el sistema entre dos entornos operativos:

ID	Requisitos de rendimiento
POR-01	El sistema debe ser compatible con diferentes plataformas y navegadores webs modernos, asegurando una correcta visualización y funcionalidad en diferentes dispositivos.
POR-02	El sistema debe ser compatible con múltiples sistemas operativos, Android e iOS.
POR-03	Se debe proporcionar una documentación clara sobre los requisitos de software y hardware necesarios para desplegar el sistema en diferentes entornos.

*Tabla 51. Requisitos no funcionales de portabilidad*

## 4.6 Requisitos de información

Aquí se empleará el modelo Entidad-Relación como esquema conceptual de datos específico para ilustrar las características de los datos que manejará el sistema, seguido de un diccionario de datos para detallar los campos de cada una de las entidades que componen la base de datos de este proyecto.

### 4.6.1. Modelo conceptual (Entidad-Relación)

En los proyectos de desarrollo software de hoy en día, en la mayoría suelen haber unos roles diferenciados, cada uno con sus funciones:

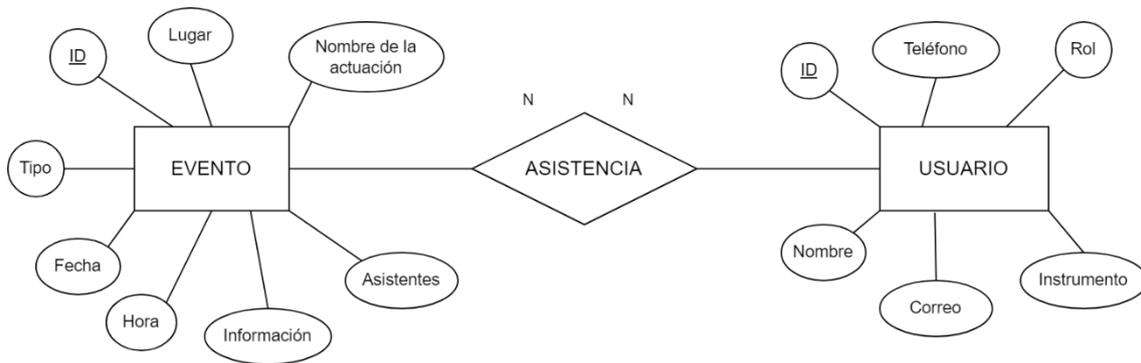


Figura 25. Modelo Entidad-Relación

### 4.6.2. Diccionario de datos

Entidad "Usuario"			
Atributo	Tipo	Único (unique)	Required
correo	string	Si	Si
instrumento	string	No	Si
nombre	string	No	Si
telefono	string	Si	Si

Tabla 52. Diccionario de datos de la entidad Usuario

Entidad "Evento"			
Atributo	Tipo	Único (unique)	Required
fecha	string	No	Si
hora	string	No	Si
información	string	No	Si
lugar	string	No	Si
nombreActuacion	string	No	Si
tipo	string	No	Si
asistentes	array[string]	No	No

Tabla 53. Diccionario de datos de la entidad Evento

## 4.7 Restricciones del sistema

- **Control de sesiones:** Si un usuario no inicia sesión por medio del formulario de inicio de sesión no podrá acceder a las funcionalidades del sistema.
- **Correo electrónico:** No pueden existir en la aplicación dos usuarios registrados con el mismo correo electrónico.
- **Nombre de usuario:** El nombre de usuario debe ser entre 5-
- **Control de contraseñas:** Las contraseñas de los usuarios deben tener una longitud mínima de 6 caracteres.
- **Gestión de roles:** Dentro de la aplicación, cada usuario tendrá un rol y por ello unos accesos u otros a ciertas funcionalidades y pantallas.



# Capítulo 5

## Diseño

En esta sección vamos a describir el diseño de la aplicación mediante la arquitectura lógica y física, unos diagramas de secuencia que sirven para representar de manera visual cómo interactúan entre sí los diferentes elementos del sistema a lo largo del tiempo, destacando el orden en el que ocurren los eventos. Al final se detallará la estructura de la interfaz de la aplicación junto a sus características principales, y un modelo explicativo de cómo es y cómo está estructurada la base de datos de este trabajo.

### 5.1 Arquitectura lógica

Para representar esta arquitectura se ha utilizado un modelo de arquitectura por capas, el cual divide el software en diferentes niveles que interactúan entre sí. Para este proyecto, se ha decidido seguir el modelo de tres capas:

- **Capa de Presentación:** Muestra los datos al usuario a través de una interfaz.
- **Capa de Negocio:** Capa responsable de procesar los datos y gestionar las interacciones.
- **Capa de Datos:** Facilita el acceso a los datos almacenados en una base de datos.

Además de utilizar la arquitectura multinivel, para estructurar la aplicación se ha implementado el patrón de diseño **MVC**[6] (**Modelo-Vista-Controlador**). Este patrón se compone de tres partes fundamentales:

- **Modelo:** Maneja los datos y la lógica de negocio. Se encarga de gestionar el acceso y actualización de los datos, generalmente interactuando con la base de datos.
- **Vista:** Es la interfaz de usuario que presenta los datos. Su función es mostrar la información de forma clara y atractiva, facilitando la interacción del usuario con la aplicación.
- **Controlador:** Actúa como intermediario entre el Modelo y la Vista. Recibe las entradas del usuario, procesa las solicitudes y actualiza tanto el Modelo como la Vista según sea necesario.

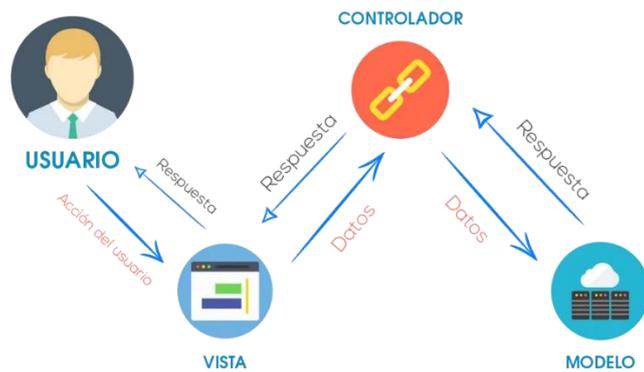


Figura 26. Patrón de diseño MVC [6]

Este enfoque no solo mejora la organización del código, sino que también facilita su escalabilidad y mantenimiento, aspectos cruciales en el desarrollo de cualquier aplicación.

Para representar esta arquitectura lógica se han creado dos diagramas de tres capas. El primero denominado “**Arquitectura Lógica Simplificada**” ofrece una visión a más alto nivel y el segundo proporciona un análisis más detallado, denominado “**Arquitectura Lógica Detallada**”.

### 5.1.1. Arquitectura Lógica Simplificada

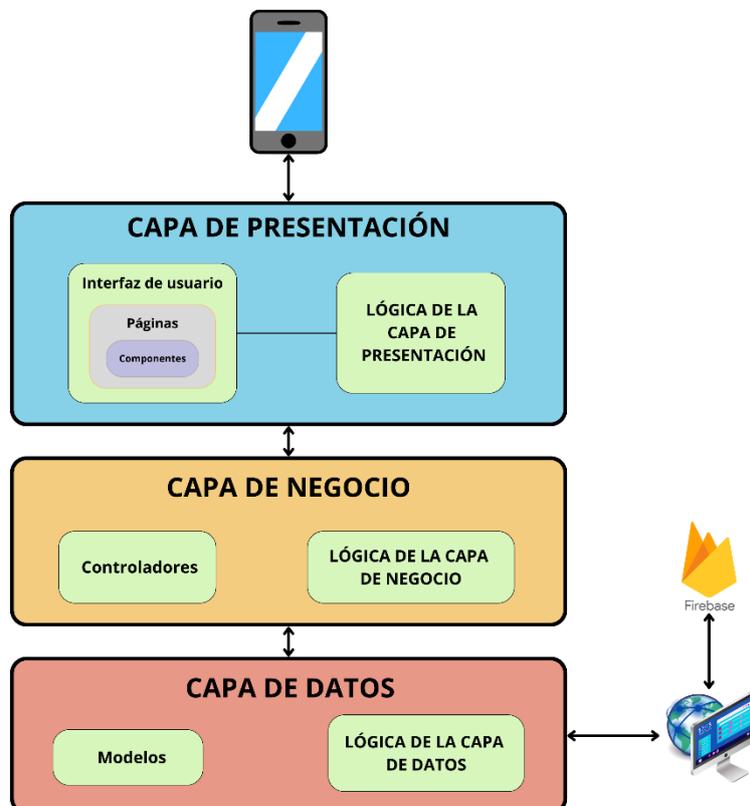


Figura 27. Diagrama de arquitectura lógica simplificada

## 5.1.2. Arquitectura Lógica Detallada

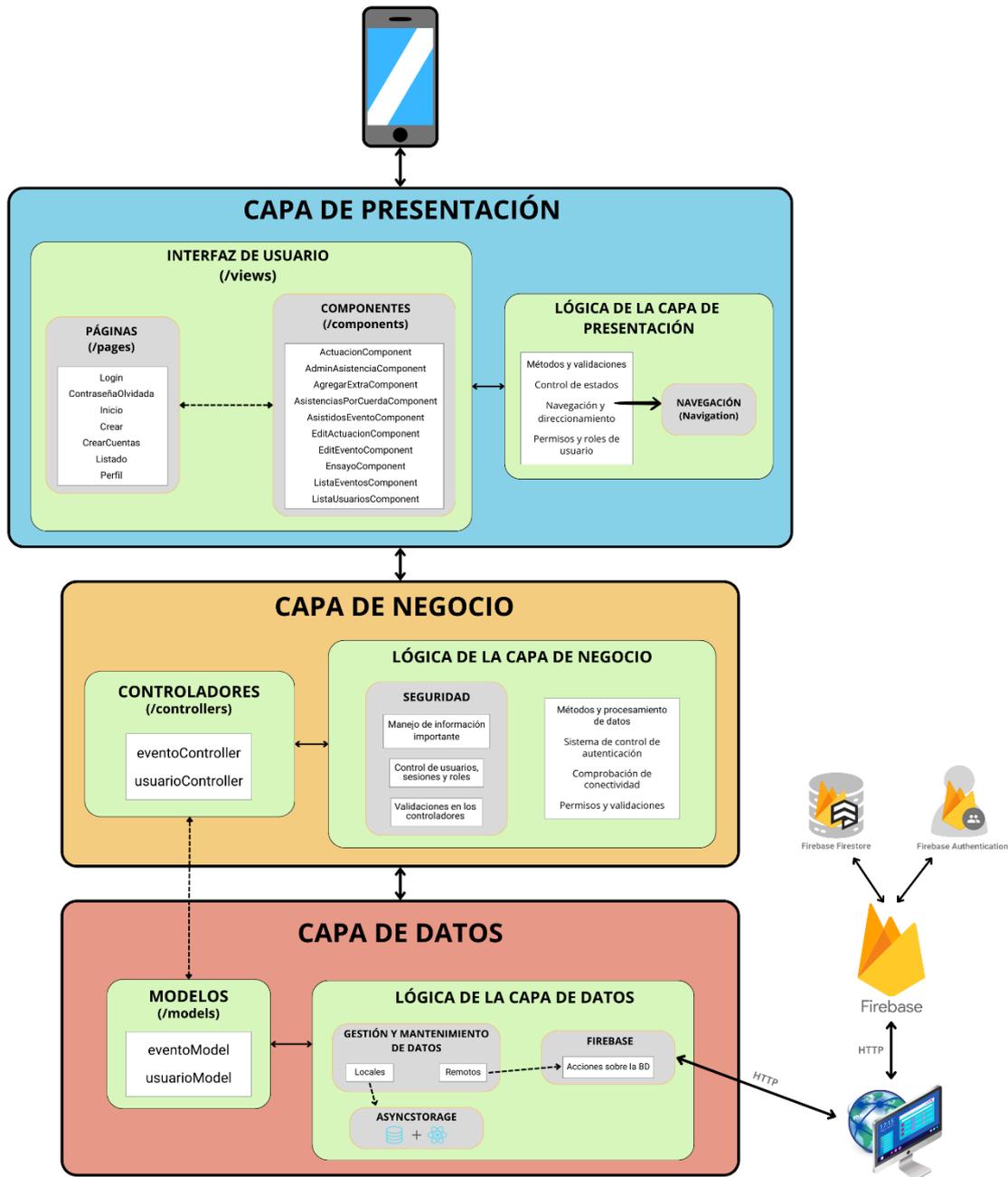


Figura 28. Diagrama de arquitectura lógica detallada

El diagrama presentado en la Figura 28 muestra la arquitectura lógica de una aplicación móvil organizada en tres capas bien diferenciadas. Cada una de estas capas cumple un papel fundamental y específico en la estructura de la aplicación, permitiendo una separación clara de responsabilidades y facilitando el mantenimiento y escalabilidad del sistema:

- **Capa de Presentación**
  - **Interfaz de Usuario (/views):** Aquí se encuentran las páginas que el usuario puede navegar.
  - **Componentes (/components):** Son partes reutilizables de la interfaz que encapsulan funcionalidades específicas.
  - **Lógica de la Capa de Presentación:** Incluye métodos y validaciones necesarias para la interacción del usuario, como control de estados de la interfaz, navegación y direccionamiento entre páginas y permisos y roles de usuario para controlar el acceso a diferentes partes de la aplicación.
- **Capa de Negocio**
  - **Controladores (/controllers):** Son los responsables de gestionar las solicitudes que vienen de la capa de presentación y convertirlas en acciones, interactuando con la capa de datos cuando sea necesario.
    - eventoController: Controlador para la gestión de eventos.
    - usuarioController: Controlador para la gestión de usuarios.
  - **Lógica de la Capa de Negocio:** Aquí se definen los métodos y procesos que manipulan y procesan los datos, además del manejo de información sensible, control de acceso basado en roles y validaciones de seguridad, sistemas de control de autenticación de usuarios y una constante verificación de la conectividad y permisos necesarios para acceder a los datos.
- **Capa de Datos**
  - **Modelos (/models):** Definen la estructura de los datos y cómo se almacenan.
  - **Lógica de la Capa de Datos:** Se encarga de la gestión y mantenimiento de los datos, tanto locales como remotos, usando:
    - AsyncStorage: Almacenamiento local de la sesión del usuario en el dispositivo.
    - Firebase: Plataforma que gestiona la base de datos y la autenticación de los usuarios. Las acciones sobre la base de datos se realizan a través de peticiones HTTP a Firebase, que a su vez sincroniza los datos con el servidor en la nube.

## 5.2 Arquitectura física

Esta arquitectura se encarga de detallar cómo está configurada e implementada la tecnología que se ha utilizado. Se mostrarán y explicarán los componentes físicos y las configuraciones necesarias para que la arquitectura lógica funcione correctamente. Esto es como el esqueleto que soporta y da forma a la estructura lógica, asegurando que todos los objetivos establecidos se puedan alcanzar.

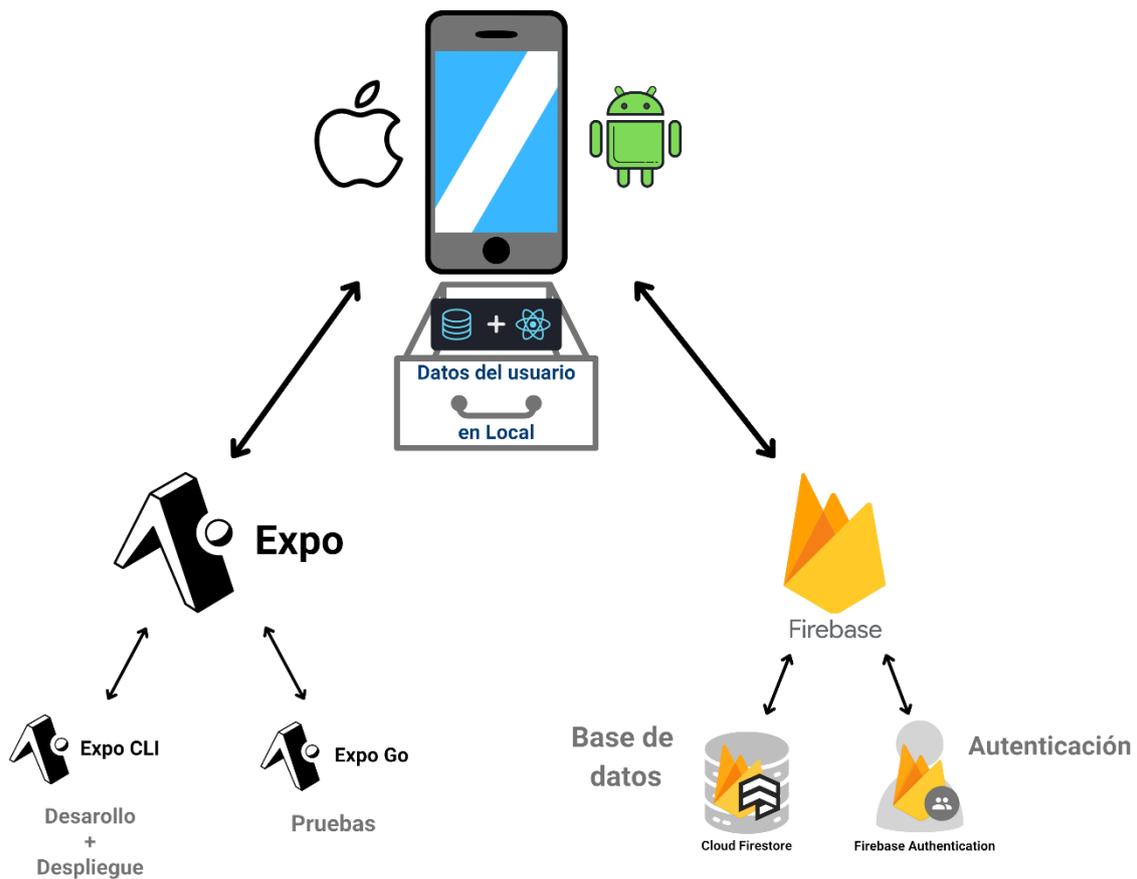


Figura 29. Diagrama de arquitectura física

En este proyecto la arquitectura física queda definida por un patrón cliente-servidor. El cliente estará definido por el dispositivo móvil utilizado por el usuario, ya sea con un sistema operativo iOS o Android. El lado del servidor en cambio se encuentra definido por dos componentes principales:

- **Expo para el desarrollo, pruebas y despliegue:**  
Es una herramienta que facilita el desarrollo, pruebas y despliegue de la aplicación.
  - Expo CLI: Utilizado para el desarrollo y despliegue.
  - Expo Go: Utilizado para pruebas, permitiendo a los desarrolladores ver cambios en tiempo real en sus dispositivos.
- **Servicios Backend (Firebase):**  
Para la base de datos se ha utilizado:
  - Cloud Firestore mediante la utilización de Firestore, una base de datos NoSQL (no relacional) en la nube que sirve para almacenar y sincronizar datos en tiempo real.
En cambio, para la autenticación se ha utilizado:
  - Firebase Authentication, que gestiona la autenticación de usuarios, permitiendo iniciar sesión de distintas maneras (la elegida ha sido correo y contraseña, pero se puede con diferentes proveedores como Google, Facebook...).

## 5.3 Diagramas de secuencia

En este apartado se han decidido realizar los diagramas de secuencia más representativos e importantes de la aplicación, debido a que el resto de flujos tendrán una secuencia de eventos y comunicación similares a los que se van a mostrar a continuación.

### 5.3.1. Crear evento (CU-06)

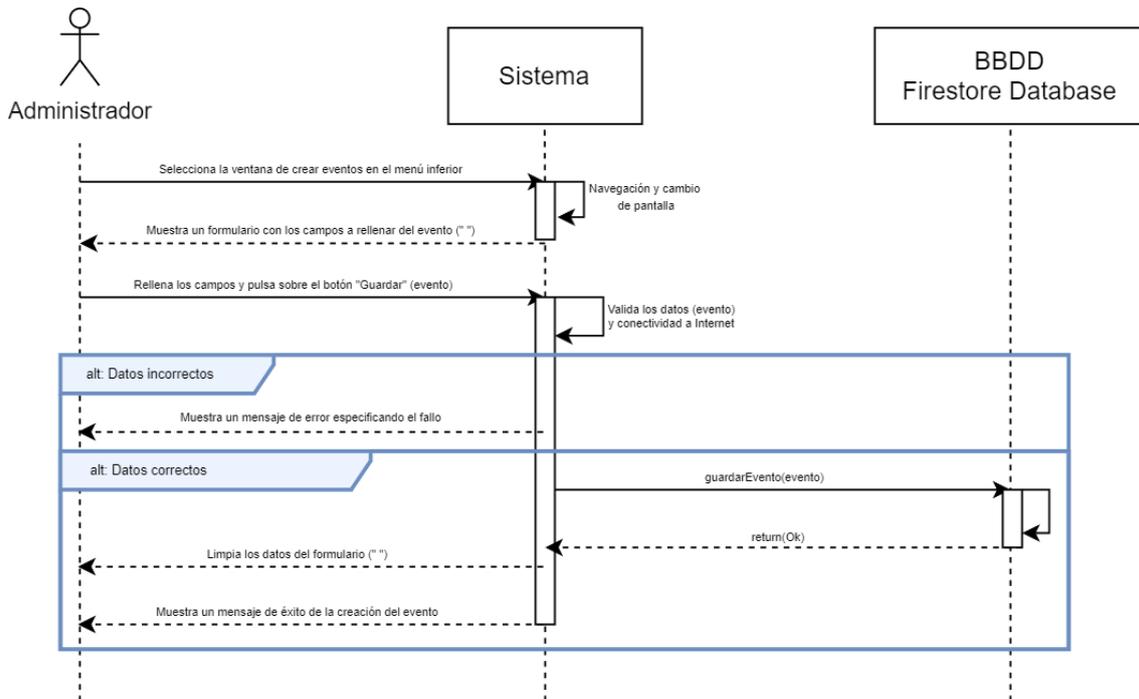


Figura 30. Diagrama de secuencia – Crear evento

### 5.3.2. Consultar nuevos eventos publicados (CU-07)

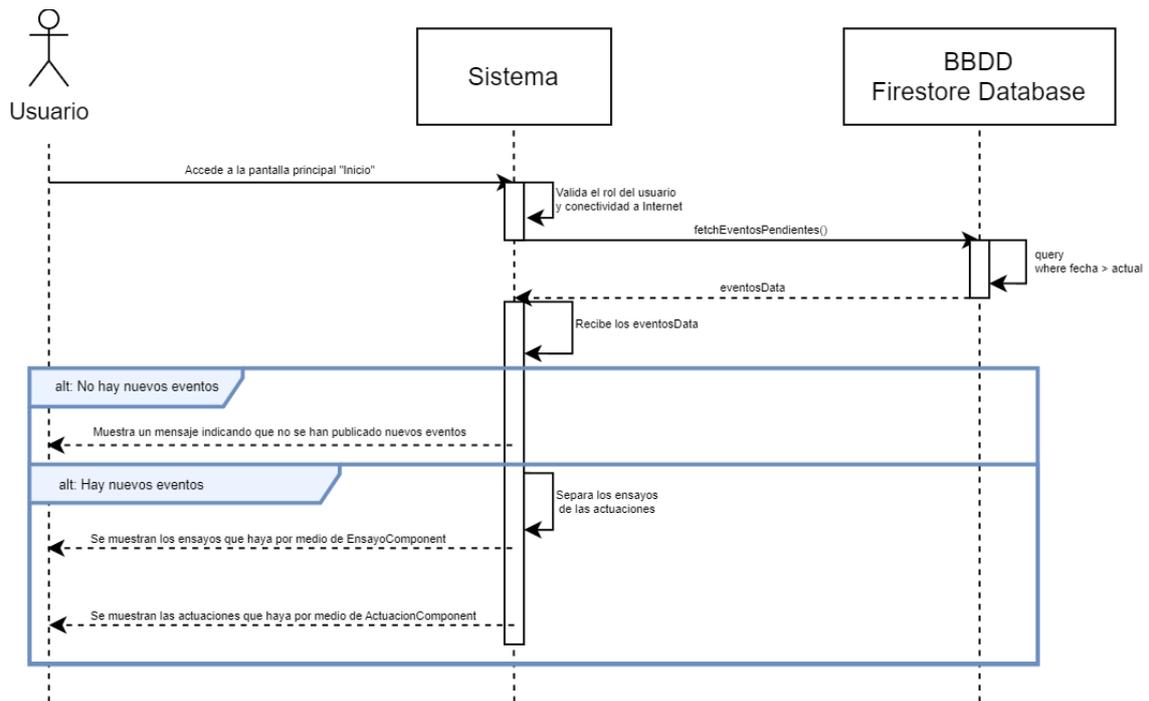


Figura 31. Diagrama de secuencia – Consultar nuevos eventos publicados

### 5.3.2. Dar de alta a nuevos miembros de la banda (CU-14)

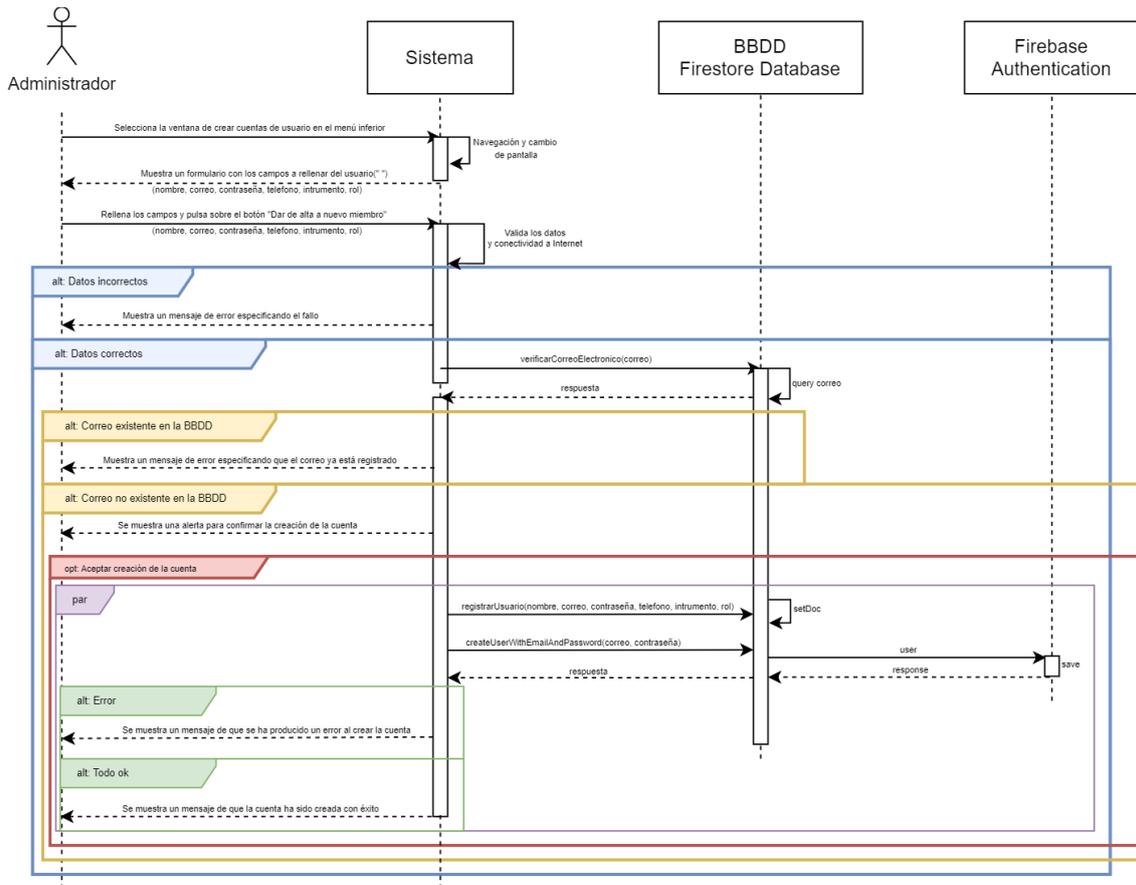


Figura 32. Diagrama de secuencia – Dar de alta a nuevos miembros de la banda

## 5.4 Diseño de la interfaz

En este apartado se detallan las interfaces de usuario que contiene la aplicación móvil de este proyecto mediante bocetos. Para su diseño se ha utilizado Draw.io, una herramienta en línea.

Es importante destacar que únicamente se mostrará lo que ve un administrador, ya que tanto los músicos como los jefes de cuerda tienen las mismas opciones que el administrador, pero con restricciones de acceso a algunas funcionalidades y pantallas.

Además, al final de todas las pantallas y bocetos de la aplicación, se plasmará la paleta de colores utilizada por la plataforma final, el logo propio de esta y se explicará a niveles generales cómo se han obtenido las ideas para la parte visual de la aplicación y hacerla más atractiva.

Inicio de sesión	
<b>Descripción</b>	Pantalla que permite a cualquier usuario registrado acceder a su cuenta personal mediante sus credenciales
<b>Activación</b>	- Al iniciar la ejecución de la aplicación - Al cerrar sesión
<b>Eventos</b>	- Pulsación sobre el botón para el envío de credenciales - Pulsación sobre el enlace "Recupérala" para acceder a la pantalla "Recuperar contraseña"
<b>Boceto</b>	

Tabla 54. Diseño de la interfaz de inicio de sesión

Recuperar contraseña	
<b>Descripción</b>	Pantalla que permite a cualquier usuario registrado recuperar su contraseña por medio de su correo electrónico
<b>Activación</b>	- Desde la pantalla de "Inicio de sesión"
<b>Eventos</b>	<ul style="list-style-type: none"> <li>- Pulsación sobre la flecha de la esquina superior izquierda para volver a la pantalla de "Inicio de sesión"</li> <li>- Pulsación sobre el botón para el envío del correo de cambio de contraseña</li> <li>- Pulsación sobre el enlace "Volver a inicio de sesión" para volver a la pantalla de "Inicio de sesión"</li> </ul>
<b>Boceto</b>	

Tabla 55. Diseño de la interfaz de recuperar contraseña

Inicio - Pantalla Principal	
<b>Descripción</b>	Pantalla que permite visualizar los últimos eventos publicados
<b>Activación</b>	<ul style="list-style-type: none"> <li>- Desde la pantalla de "Inicio de sesión"</li> <li>- Desde la barra de navegación (abajo)</li> </ul>
<b>Eventos</b>	<ul style="list-style-type: none"> <li>- Pulsación sobre el icono de la esquina superior derecha para ir a la pantalla "Perfil"</li> <li>- Un ScrollView para subir y bajar la pantalla y así poder visualizar todos los eventos que haya por hacer</li> <li>- Pulsación sobre el icono de los tres puntos de un componente para mostrar un modal con los detalles del evento en cuestión</li> <li>- Pulsación sobre el icono de papelera de basura de un componente para mostrar un modal y eliminar el evento en cuestión</li> <li>- Pulsación sobre el icono del lápiz de un componente para mostrar un modal y editar el evento en cuestión</li> <li>- Barra de navegación (abajo) para acceder a otras pantallas de la aplicación</li> </ul>
<b>Boceto</b>	

Tabla 56. Diseño de la interfaz de pantalla principal

Detalles de un evento	
<b>Descripción</b>	Modal que muestra los detalles y asistidos de un evento específico
<b>Activación</b>	- Desde la pantalla de “Inicio – Pantalla principal”
<b>Eventos</b>	<ul style="list-style-type: none"> <li>- Pulsación sobre el icono de la esquina superior derecha para cerrar el modal</li> <li>- Pulsación sobre el botón Apuntar/Desapuntar de un usuario para marcar o eliminar su asistencia a ese evento</li> <li>- Un ScrollView para subir y bajar el Container de los miembros y así poder visualizar todos los usuarios registrados</li> <li>- Pulsación sobre el botón inferior “Añadir nuevo extra” para abrir un modal con un formulario</li> </ul>
<b>Boceto</b>	

*Tabla 57. Diseño de la interfaz de detalles de un evento*

Editar evento	
<b>Descripción</b>	Modal que permite editar los datos de un evento
<b>Activación</b>	- Desde la pantalla de “Inicio – Pantalla principal”
<b>Eventos</b>	<ul style="list-style-type: none"> <li>- Pulsación sobre el icono de la esquina superior derecha para cerrar el modal</li> <li>- Escribir para modificar los distintos TextInput</li> <li>- La fecha y la hora son DateTimePicker</li> <li>- Pulsación sobre el botón inferior de la derecha “Guardar cambios” para guardar los cambios hechos y cerrar el modal</li> <li>- Pulsación sobre el botón inferior de la izquierda “Cancelar” para cerrar el modal</li> </ul>
<b>Boceto</b>	

Tabla 58. Diseño de la interfaz de editar un evento

Eliminar evento	
<b>Descripción</b>	Modal que permite eliminar un evento
<b>Activación</b>	- Desde la pantalla de “Inicio – Pantalla principal”
<b>Eventos</b>	- Pulsación sobre el botón inferior de la derecha “Aceptar” para borrar el evento y cerrar el modal - Pulsación sobre el botón inferior de la izquierda “Cancelar” para cerrar el modal
<b>Boceto</b>	

Tabla 59. Diseño de la interfaz de eliminar un evento

Perfil	
<b>Descripción</b>	Pantalla que muestra la información del perfil del usuario
<b>Activación</b>	- Desde cualquier pantalla de la barra de navegación
<b>Eventos</b>	- Pulsación sobre el icono de la esquina superior derecha para cerrar sesión - Pulsación sobre el botón inferior "Modificar perfil" para abrir un modal y modificar datos del usuario
<b>Boceto</b>	

Tabla 60. Diseño de la interfaz de perfil

Modificar perfil	
<b>Descripción</b>	Modal que permite editar los datos de un evento
<b>Activación</b>	- Desde la pantalla de “Inicio – Pantalla principal”
<b>Eventos</b>	- Escribir para modificar los distintos TextInput - Pulsación sobre el botón inferior de la derecha “Guardar cambios” para guardar los cambios hechos y cerrar el modal - Pulsación sobre el botón inferior de la izquierda “Cancelar” para cerrar el modal
<b>Boceto</b>	

Tabla 61. Diseño de la interfaz de editar perfil

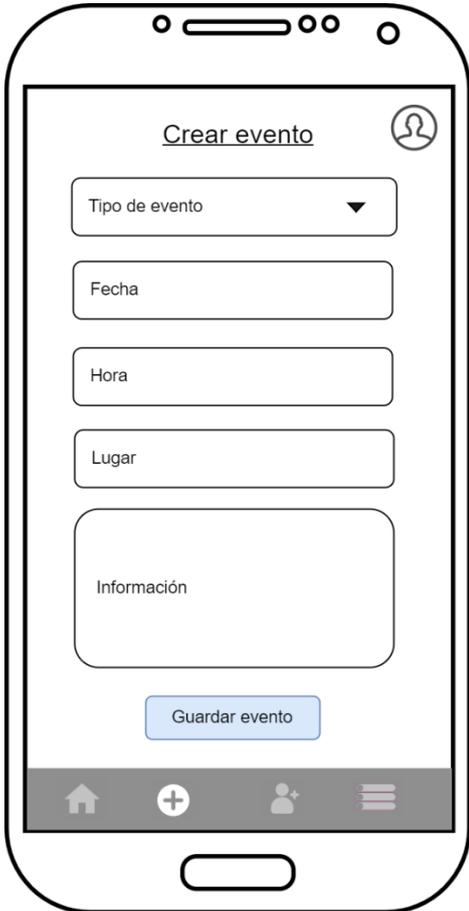
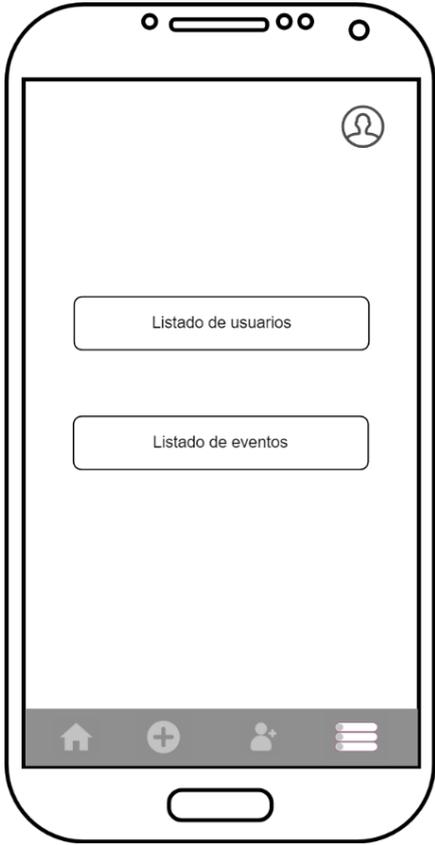
Crear evento	
<b>Descripción</b>	Pantalla que contiene un formulario que permite crear nuevos eventos para que lo puedan visualizar todos los usuarios registrados en la aplicación
<b>Activación</b>	- Desde la barra de navegación (abajo)
<b>Eventos</b>	<ul style="list-style-type: none"> <li>- Pulsación sobre el icono de la esquina superior derecha para ir a la pantalla "Perfil"</li> <li>- Pulsación sobre el "Picker" para seleccionar el tipo de evento (ensayo o actuación)</li> <li>- La fecha y la hora son DateTimePicker para elegir correctamente</li> <li>- TextInputs a rellenar con los datos necesarios</li> <li>- Pulsación sobre el botón inferior "Guardar evento" para guardar los datos introducidos del nuevo evento en la base de datos</li> </ul>
<b>Boceto</b>	

Tabla 62. Diseño de la interfaz de crear evento

Registro de cuentas	
<b>Descripción</b>	Pantalla que contiene un formulario que permite dar de alta a nuevos miembros y extras para que puedan utilizar la aplicación (salvo los extras)
<b>Activación</b>	- Desde la barra de navegación (abajo)
<b>Eventos</b>	<ul style="list-style-type: none"> <li>- Pulsación sobre el icono de la esquina superior derecha para ir a la pantalla "Perfil"</li> <li>- Pulsación sobre los "Picker" para seleccionar un instrumento y rol específicos</li> <li>- TextInputs a rellenar con los datos necesarios del usuario</li> <li>- Pulsación sobre el botón inferior "Dar de alta a nuevo miembro" para guardar los datos introducidos del nuevo miembro en la base de datos</li> <li>- Pulsación sobre el botón inferior "Dar de alta a extra" para mostrar un modal con un formulario para un nuevo extra</li> </ul>
<b>Boceto</b>	

Tabla 63. Diseño de la interfaz de registro de cuentas

Listados	
<b>Descripción</b>	Ventana que contiene unos botones para mostrar unas ventanas con los listados específicos, usuarios y eventos realizados
<b>Activación</b>	- Desde la barra de navegación (abajo)
<b>Eventos</b>	<ul style="list-style-type: none"> <li>- Pulsación sobre el icono de la esquina superior derecha para ir a la pantalla "Perfil"</li> <li>- Pulsación sobre el botón superior "Listado de usuarios" para mostrar una ventana con el listado de todos los usuarios registrados en la aplicación</li> <li>- Pulsación sobre el botón superior "Listado de eventos" para mostrar una ventana con el listado de los eventos ya realizados</li> </ul>
<b>Boceto</b>	

*Tabla 64. Diseño de la interfaz de listados*

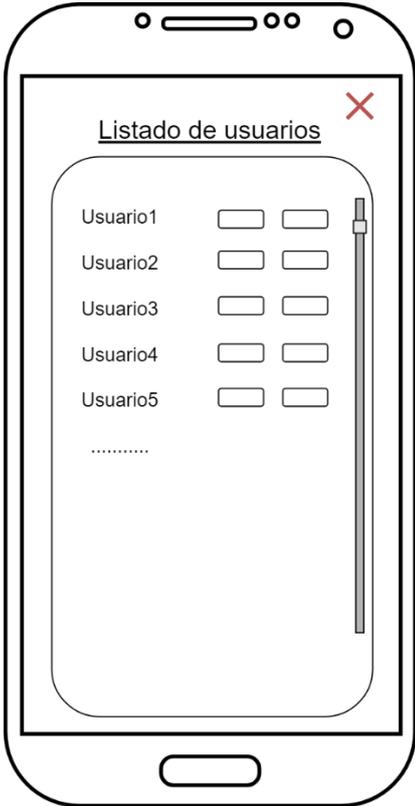
Listado de usuarios	
<b>Descripción</b>	Pantalla que contiene un listado con todos los usuarios registrados en la aplicación
<b>Activación</b>	- Desde la pantalla "Listado"
<b>Eventos</b>	<ul style="list-style-type: none"> <li>- Pulsación sobre el icono de la esquina superior derecha para cerrar el modal</li> <li>- Un ScrollView para subir y bajar el Container de los miembros y así poder visualizar todos los usuarios registrados</li> <li>- Pulsación sobre uno de los botones de un usuario para dar de baja y eliminar al usuario de la aplicación</li> <li>- Pulsación sobre uno de los botones de un usuario para ascender o descender según el rol que tenga a un rango superior o inferior</li> </ul>
<b>Boceto</b>	

Tabla 65. Diseño de la interfaz de listado de usuarios

Listado de eventos realizados	
<b>Descripción</b>	Pantalla que contiene un listado con los eventos ya realizados de la aplicación
<b>Activación</b>	- Desde la pantalla "Listado"
<b>Eventos</b>	<ul style="list-style-type: none"> <li>- Pulsación sobre el icono de la esquina superior derecha para cerrar el modal</li> <li>- Un ScrollView para subir y bajar la vista y así poder visualizar todos los eventos realizados</li> <li>- Pulsación sobre el botón exportar PDF para generar un PDF con los asistidos del evento específico</li> <li>- Pulsación sobre el icono de papelera de basura de un evento para eliminar el evento en cuestión</li> </ul>
<b>Boceto</b>	

Tabla 66. Diseño de la interfaz de listado de eventos realizados

### 5.4.1. Ideas generales del diseño

Esta aplicación móvil se ha diseñado con una idea centrada en la experiencia del usuario, tomando de referencia e inspiración las redes sociales populares que todos utilizamos a diario. Este enfoque familiar busca que los usuarios se sientan cómodos y encuentren la app intuitiva desde el primer momento, haciendo que por ejemplo el inicio de sesión y visualización de perfiles sean claros y concisos, que la barra de navegación tenga elementos familiares para una navegación intuitiva, que haya iconos y botones reconocibles con un diseño visual limpio y moderno, etcétera.

Todo esto es para dar al usuario una apariencia atractiva, agradable y a la vez profesional desde el primer uso que se le da a la aplicación.

### 5.4.1. Paleta de colores

La paleta de colores mostrada en la Figura 33 sigue los colores más representativos de la banda para la que se ha desarrollado esta aplicación móvil. Estos colores fueron seleccionados principalmente para reflejar la identidad visual de la banda y crear una experiencia de usuario coherente y atractiva. Los tonos elegidos no solo mejoran la estética de la aplicación, sino que también refuerzan el vínculo emocional con los miembros de la banda, asegurando que la aplicación sea visualmente alineada con la imagen y espíritu de la banda.

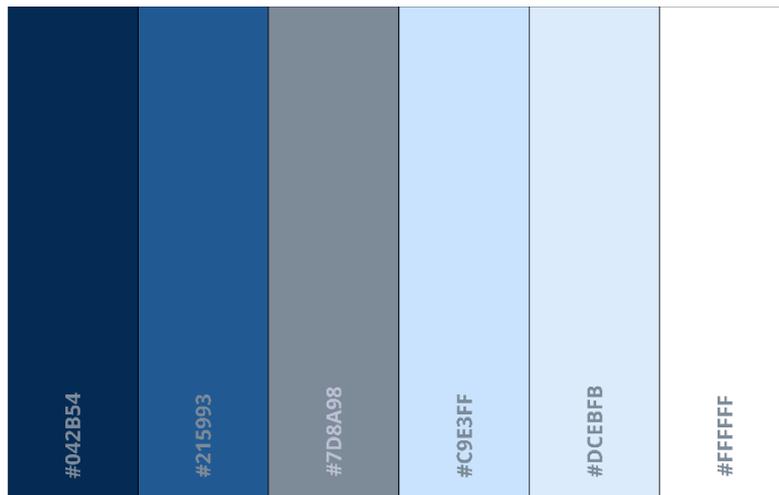


Figura 33. Paleta de colores de BanDate

### 5.4.2. Logo de BanDate

Con ayuda de draw.io, se ha creado el siguiente logo propio para este proyecto:



Figura 34. Logo de la aplicación

## 5.5 Modelo lógico de datos

Para comprender cómo se estructura y se diseña la base de datos de la aplicación móvil desarrollada, vamos a desglosar estos conceptos y las diversas entidades involucradas. Además, se hará especial énfasis en el tipo de base de datos utilizado: **NoSQL**.

El sistema se basa en Firebase, una plataforma digital que ofrece una variedad de servicios para el desarrollo de aplicaciones, aplicaciones móviles tanto para Android como iOS. Específicamente, el servicio utilizado para proporcionar la funcionalidad de base de datos es Cloud Firestore.

Cloud Firestore es una base de datos no relacional, también conocida como NoSQL, distinta de las bases de datos relacionales o SQL. Las diferencias entre estos dos tipos de bases de datos son significativas, pero en esta sección nos enfocaremos en la estructura y el modelado de los datos:

- **Cloud Firestore (NoSQL):** Almacena la información en colecciones de documentos y tiene flexibilidad en el modelado de los datos (documentos compuestos por diferentes tipologías de datos)
- **Bases de datos relacionales (SQL):** Almacena la información en tablas mediante filas y columnas, y tiene una estricta integridad referencial entre tablas, relaciones entre tablas e índices.

Así, la estructura de los datos de la base de datos NoSQL se ha definido de la siguiente forma como se muestra en la Figura 35, donde aparecen representadas las principales colecciones de documentos que hacen referencia a las diferentes entidades:



Figura 35. Estructura jerárquica de datos en Cloud Firestore

Dentro de estas colecciones se crean documentos de forma individual, los cuales se encuentran definidos por un identificador único y campos o pares clave-valor que soportan multitud de tipologías de datos para almacenar la información, como se puede ver en la siguiente imagen:

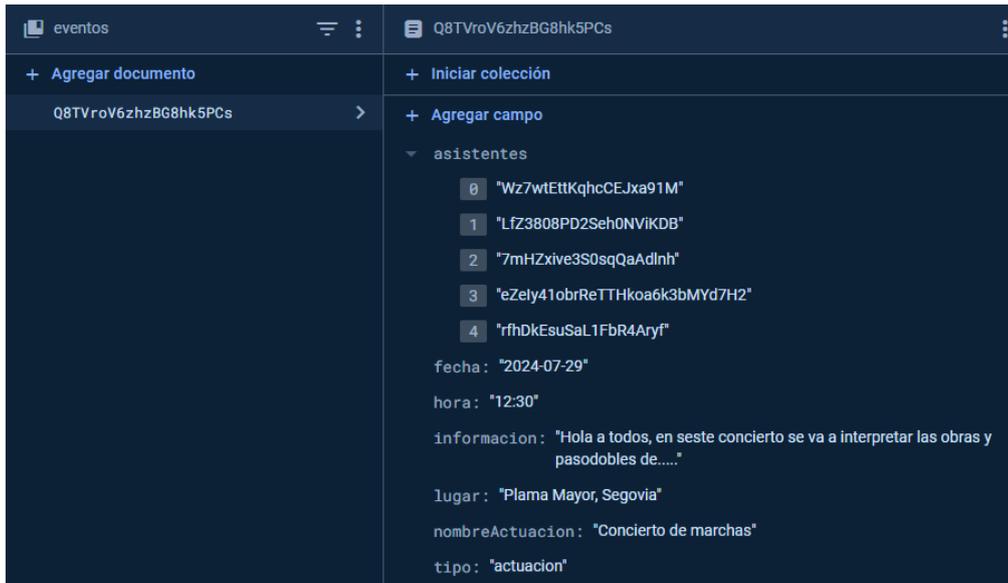


Figura 36. Estructura de documento en Cloud Firestore

Una vez explicado y comprendido los principios fundamentales de Cloud Firestore y las bases de datos no relacionales (NoSQL), vamos a incorporar un modelo lógico de datos al proyecto. Este modelo es independiente del tipo de base de datos y nos permitirá visualizar con mayor claridad las relaciones entre las diferentes entidades y sus atributos.

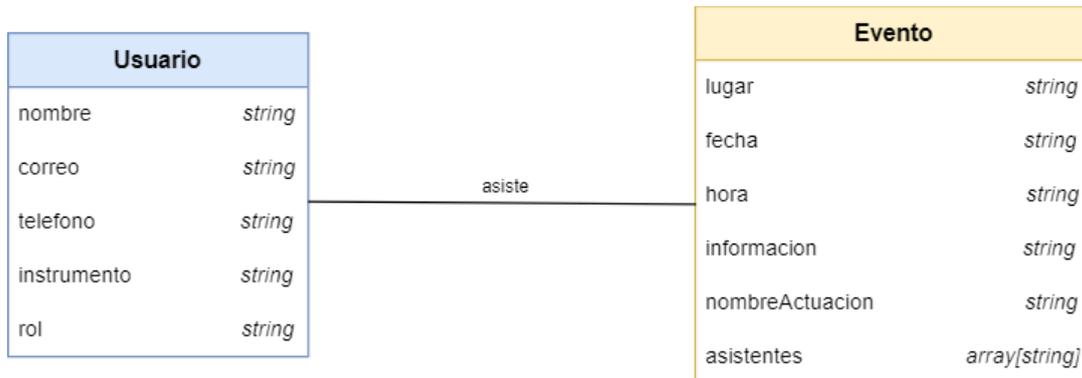


Figura 37. Estructura de documento en Cloud Firestore

Cada documento de usuario y evento tendrá su identificador propio, son únicos e irrepetibles. La relación entre ambos se establece en el array asistentes, donde se van guardando los identificadores de los usuarios que marcan asistencia para el evento que ellos requieran.

### 5.5.1. La utilidad de modelos con esquemas

La utilidad del modelado mediante esquemas es importante ya que evitan la llamada inyección de código. En la construcción del esquema se debe describir la colección y sus propiedades, especificando para cada propiedad sus tipos de datos y restricciones si tiene. De este modo, todas las acciones sobre la base de datos que impliquen una colección se ciñen a su esquema, no permitiendo la entrada de otros datos que no sean los especificados en el esquema.

Así pues, dado que la inyección se lleva a cabo enviando datos maliciosos (cadenas de caracteres que empiezan por \$ y son sentencias), al usar esquemas no sucederá nada que no queremos que ocurra. Un ejemplo utilizado es modelo para la colección Evento, representado en la Figura 38:

```
class Evento {
  /**
   * @param {string} tipo - Tipo de evento (ensayo, actuación).
   * @param {string} lugar - Lugar del evento.
   * @param {string} informacion - Información adicional sobre el evento.
   * @param {string} fecha - Fecha del evento en formato YYYY-MM-DD.
   * @param {string} hora - Hora del evento en formato HH:mm.
   * @param {string} nombreActuacion - Nombre de la actuación.
   * @param {Array<string>} asistentes - Lista de asistentes al evento.
   */
  constructor(tipo, lugar, informacion, fecha, hora, nombreActuacion, asistentes = []) {
    this.tipo = tipo;
    this.lugar = lugar;
    this.informacion = informacion;
    this.fecha = fecha;
    this.hora = hora;
    this.nombreActuacion = nombreActuacion;
    this.asistentes = asistentes;
  }

  // Método para convertir el objeto a un formato adecuado para guardar en Firestore
  toFirestoreObject() {
    return {
      tipo: this.tipo,
      lugar: this.lugar,
      informacion: this.informacion,
      fecha: this.fecha,
      hora: this.hora,
      nombreActuacion: this.nombreActuacion,
      asistentes: this.asistentes
    };
  }
}

export default Evento;
```

Figura 38. Captura de pantalla del modelo-esquema de la colección Eventos



# Capítulo 6

## Implementación

En este capítulo vamos a explicar cómo está organizado y cómo funciona el proyecto en la práctica. Primero, se dará una visión general de su estructura, describiendo cómo se han organizado las distintas carpetas con sus archivos. Luego, se explicará algunas de las funcionalidades más importantes implementadas.

### 6.1 Estructura del proyecto

En este apartado se describe la estructura de la solución y la funcionalidad de alguno de los módulos más importantes que componen la aplicación y el porqué de la estructura y organización elegida.

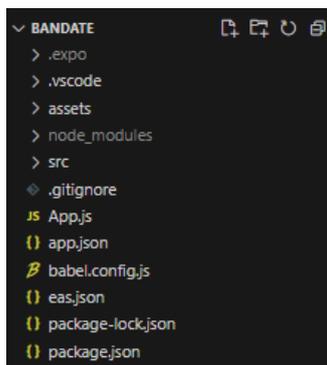


Figura 39. Captura de pantalla de la estructura del proyecto

En la imagen anterior se puede visualizar la estructura completa de toda la aplicación. La lógica que nos interesa y más importante de la aplicación se encuentra en las carpetas “assets” y “src”, y en el archivo App.js, que todo ello se detallará más adelante.

A continuación, se van a mostrar las carpetas y archivos menos importantes pero que aun así están ahí y hacen que este proyecto funcione correctamente:

- **expo:** Carpeta donde Expo guarda configuraciones específicas y datos del proyecto. No se ha manipulado esta carpeta en ningún momento durante el proyecto.
- **vscode:** Contiene configuraciones específicas del editor Visual Studio Code, como ajustes y extensiones recomendadas para el proyecto.
- **node\_modules:** Contiene todas las dependencias y módulos instalados a través de npm o yarn. Esta carpeta se genera automáticamente al instalar los paquetes.

- **gitignore:** Archivo que especifica qué archivos y carpetas deben ser ignorados por Git. Esto es útil para evitar subir archivos innecesarios o sensibles al repositorio.
- **app.json:** Archivo de configuración para la aplicación Expo. Aquí se definen aspectos como el nombre de la app, icono, permisos necesarios, etc.
- **babel.config.js:** Archivo de configuración de Babel, que es un compilador de JavaScript. Aquí se especifican los presets y plugins que Babel debe usar.
- **eas.json:** Archivo de configuración para EAS (Expo Application Services), utilizado para configuraciones de compilación y despliegue en la nube de Expo. Este archivo se explicará más en detalle en el apartado de despliegue.
- **package-lock.json:** Archivo generado automáticamente que asegura que las mismas versiones de las dependencias sean instaladas en cada instalación del proyecto. Ayuda a mantener la consistencia del entorno de desarrollo.
- **package.json:** Archivo que contiene metadatos del proyecto y las dependencias necesarias. También incluye scripts para ejecutar tareas comunes, como iniciar el servidor de desarrollo o compilar el proyecto.

En la carpeta “assets” se almacenan los recursos estáticos del proyecto. En este caso se almacenan los iconos y logos propios de la aplicación BanDate, y, además, se ha creado una carpeta extra llamada “iconos” como se puede ver en la Figura 40:

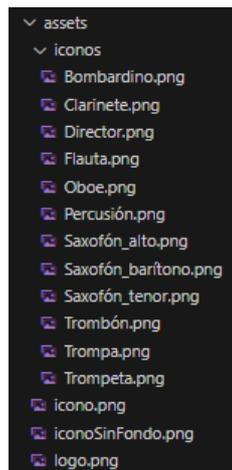


Figura 40. Captura de pantalla de la carpeta assets

Esta carpeta contiene imágenes de los distintos instrumentos que contempla la aplicación, que se podrán visualizar en el perfil de usuario según el instrumento que este tenga asignado. Ejemplo del icono “Trompa.png”:



Figura 41. Icono Trompa.png

La carpeta “*src*” es la carpeta principal del código fuente de la aplicación. Aquí se organizan los componentes, pantallas y otros archivos de código.

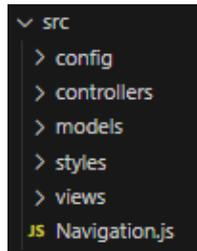


Figura 42. Captura de pantalla de la carpeta *src*

Para la organización de los archivos que contienen el código, se ha seguido el patrón **MVC o modelo vista controlador**, el cual es un patrón de diseño que permite hacer una separación del código según sus distintas responsabilidades, para una mejor organización y mantenibilidad.

La arquitectura MVC propone la separación de los componentes de una aplicación en tres grupos:

- **Modelos:** En la carpeta “*models*” se encuentran los modelos con los esquemas de las colecciones de la base de datos:

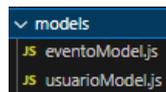


Figura 43. Captura de pantalla de la carpeta *src/models*

- **Vistas:** En la carpeta “*views*” están las distintas pantallas/páginas que ve y con las que interactúa el usuario, a parte de los componentes que contienen estas:

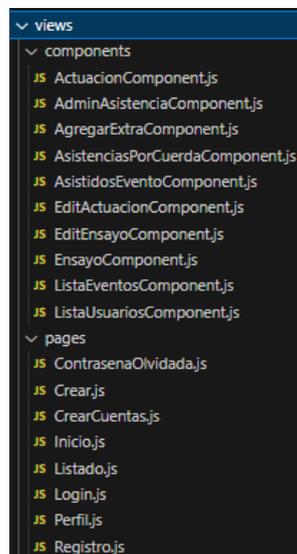


Figura 44. Captura de pantalla de la carpeta *src/views*

- **Controladores:** En la carpeta “*controllers*” se encuentra la lógica necesaria para responder las acciones que se realizan en la aplicación, ya sean funciones para crear, leer, actualizar y eliminar datos en la base de datos. Se han dividido en dos archivos, uno para cada colección que contiene la base de datos:

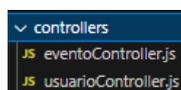


Figura 45. Captura de pantalla de la carpeta *src/controllers*

A parte de estas carpetas anteriores de la carpeta “src”, en la carpeta “config” se encuentra el archivo *firebase.js*.

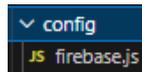


Figura 46. Captura de pantalla de la carpeta *src/config*

```
import { initializeApp } from "firebase/app";
import { getFirestore } from "firebase/firestore";
import { getAuth } from "firebase/auth";
import AsyncStorage from '@react-native-async-storage/async-storage';

// Firebase configuración
const firebaseConfig = {
  apiKey: " ",
  authDomain: " ",
  projectId: " ",
  storageBucket: " ",
  messagingSenderId: " ",
  appId: " "
};

const app = initializeApp(firebaseConfig);
const database = getFirestore(app);
const auth = getAuth(app);

// AsyncStorage para persistencia de sesión
const SESSION_STORAGE_KEY = '@MyApp:session';
const getSession = async () => {
  try {
    const sessionJSON = await AsyncStorage.getItem(SESSION_STORAGE_KEY);
    return sessionJSON ? JSON.parse(sessionJSON) : null;
  } catch (e) {
    console.error('Error al obtener la sesión desde AsyncStorage:', e);
    return null;
  }
};

const setSession = async (session) => {
  try {
    await AsyncStorage.setItem(SESSION_STORAGE_KEY, JSON.stringify(session));
  } catch (e) {
    console.error('Error al almacenar la sesión en AsyncStorage:', e);
  }
};

const removeSession = async () => {
  try {
    // Eliminar la sesión de AsyncStorage utilizando la clave específica
    await AsyncStorage.removeItem(SESSION_STORAGE_KEY);
  } catch (error) {
    console.error('Error al eliminar la sesión de AsyncStorage:', error);
  }
};

export { database, auth, getSession, setSession, removeSession };
```

Figura 47. Captura de pantalla del archivo *src/config/config.js*

Este archivo es el que configura Firebase en este proyecto, permitiendo usar Cloud Firestore (base de datos) y Firebase Auth (autenticación). También gestiona la persistencia de la sesión del usuario utilizando AsyncStorage para guardar, obtener y eliminar la información de sesión de manera local. Se han tapado las claves de configuración de la base de datos por privacidad y seguridad.

En la carpeta “styles” hay únicamente un archivo, el cual contiene la paleta de colores que se explicó en el apartado diseño y que se utiliza en toda parte visual de la aplicación.

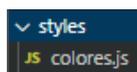


Figura 48. Captura de pantalla de la carpeta *src/styles*

### 6.1.1. Implementación de la navegación

Por último y no menos importante aparece en la carpeta principal también el archivo “*Navigation.js*”.

Este archivo configura y administra la estructura de navegación, utilizando *react-navigation* para definir las rutas y pantallas.

En él se utiliza el componente *AuthenticatedUserProvider*, que maneja el estado del usuario verificando la autenticación con Firebase y cargando la sesión que hay en local si existe. También verifica periódicamente la conexión a internet y muestra una alerta si no hay conexión con ayuda de la biblioteca importada **NetInfo**[16].

```
useEffect(() => {
  const loadSession = async () => {
    const session = await getSession();
    if (session) {
      setUser(session);
      try {
        const userDocRef = doc(database, 'usuarios', session.uid);
        const userDoc = await getDoc(userDocRef);
        if (userDoc.exists()) {
          const userData = userDoc.data();
          setUserRole(userData.rol);
          setUserDatos(userData);
        }
      } catch (error) {
        Alert.alert('Error', 'Error al obtener los datos. Inténtalo de nuevo más tarde.');
```

```
        console.error('Error al obtener datos de usuario:', error.message);
      }
    }
    setIsLoading(false);
  };

  const unsubscribeAuth = onAuthStateChanged(auth, async (authenticatedUser) => {
    if (authenticatedUser) {
      setUser(authenticatedUser);
      setIsLoading(false);

      try {
        const userDocRef = doc(database, 'usuarios', authenticatedUser.uid);
        const unsubscribeUser = onSnapshot(userDocRef, (doc) => {
          if (doc.exists()) {
            const userData = doc.data();
            console.log(userData.rol);
            setUserRole(userData.rol);
            setUserDatos(userData);
          }
        });
        return () => unsubscribeUser();
      } catch (error) {
        Alert.alert('Error', 'Error al obtener los datos. Inténtalo de nuevo más tarde.');
```

```
        console.error('Error al obtener datos de usuario:', error.message);
      }
    } else {
      await loadSession();
    }
    setIsLoading(false);
  });

  // Función para verificar la conexión y mostrar alerta
  const checkConnection = () => {
    NetInfo.fetch().then(state => {
      if (!state.isConnected) {
        Alert.alert('Sin conexión', 'No hay conexión a internet. Inténtalo de nuevo más tarde.');
```

```
      }
    });
  };

  // Verificar la conexión cada 10 segundos
  const intervalId = setInterval(checkConnection, 20000);

  return () => {
    unsubscribeAuth();
    clearInterval(intervalId);
  };
}, []);
```

Figura 49. Captura de pantalla del archivo *src/Navigation.js* (estado del usuario y conectividad a Internet)

El componente principal *Navigation* envuelve la aplicación en *AuthenticatedUserProvider* y *NavigationContainer*, asegurando que la navegación tenga acceso al contexto del usuario autenticado. Dependiendo del estado de autenticación del usuario, *RootNavigator* decide qué pantallas mostrar: una pantalla de inicio de sesión para usuarios no autenticados o una navegación tabulada para usuarios autenticados.

```

export default function Navigation() {
  return (
    <AuthenticatedUserProvider>
      <NavigationContainer>
        <RootNavigator />
      </NavigationContainer>
    </AuthenticatedUserProvider>
  );
}

function RootNavigator() {
  const { user } = useContext(AuthenticatedUserContext);

  if (!user) {
    return (
      <Stack.Navigator>
        <Stack.Screen
          name="Login"
          component={LoginScreen}
          options={{ headerShown: false }}
        />
        <Stack.Screen
          name="ContraseñaOlvidada"
          component={ContraseñaOlvidada}
          options={{ title: 'Recuperar Contraseña' }}
        />
      </Stack.Navigator>
    );
  }

  return (
    <Stack.Navigator>
      <Stack.Screen
        name="BanDate"
        component={MainTabNavigator}
        options={{ navigation }} => ({
          headerRight: () => (
            <TouchableOpacity onPress={() => navigation.navigate('Perfil')}>
              <Icon name="person-circle-outline" size={36} color={colores.background} style={{ marginRight: 15 }} />
            </TouchableOpacity>
          ),
          headerStyle: {
            backgroundColor: colores.primary,
          },
          headerTintColor: colores.text,
        })
      />
      <Stack.Screen
        name="Perfil"
        component={ProfileScreen}
        options={{
          title: 'Perfil',
          headerStyle: { backgroundColor: colores.primary },
          headerTintColor: colores.text,
          animationEnabled: false,
        }}
      />
    </Stack.Navigator>
  );
}

```

Figura 50. Captura de pantalla del archivo *src/Navigation.js* (método principal)

Otro punto importante es la diferencia en la navegación dependiendo del rol del usuario (por ejemplo, solo los administradores pueden acceder a ciertas pestañas). El archivo también gestiona la navegación entre las pantallas de inicio, creación de cuentas, listado de eventos y el perfil del usuario, ajustando la interfaz según el estado del usuario.

```
function MainTabNavigator() {
  const { userRole } = useContext(AuthenticatedUserContext);

  return (
    <Tab.Navigator
      initialRouteName="Inicio"
      screenOptions={{
        tabBarActiveTintColor: colores.text,
        tabBarStyle: {
          display: 'flex',
          backgroundColor: colores.primary,
          borderTopWidth: 0
        }
      }}>
      <Tab.Screen
        name="Inicio"
        component={InicioScreen}
        options={{
          tabBarLabel: () => null, headerShown: false, tabBarIcon: ({ color }) => (
            <Icon name="home" size={25} color={color} />
          ),
        }}
      />
      {userRole === 'admin' && (
        <Tab.Screen
          name="Crear"
          component={CrearScreen}
          options={{
            tabBarLabel: () => null, headerShown: false, tabBarIcon: ({ color }) => (
              <Icon name="add-circle" size={25} color={color} />
            ),
          }}
        />
      )}
      {userRole === 'admin' && (
        <Tab.Screen
          name="CrearCuentas"
          component={CrearCuentas}
          options={{
            tabBarLabel: () => null, headerShown: false, tabBarIcon: ({ color }) => (
              <Icon name="person-add" size={25} color={color} />
            ),
          }}
        />
      )}
      <Tab.Screen
        name="Listado"
        component={ListadoScreen}
        options={{
          tabBarLabel: () => null, headerShown: false, tabBarIcon: ({ color }) => (
            <Icon name="list" size={25} color={color} />
          ),
        }}
      />
    </Tab.Navigator>
  );
}
```

Figura 51. Captura de pantalla del archivo `src/Navigation.js` (barra de navegación)

## 6.1.2. Archivo principal

Este archivo es el que habíamos visto en la estructura del proyecto y no se había hablado de él aún. Es el archivo principal de la aplicación e importa el componente *Navigation* que acabamos de explicar anteriormente, y lo utiliza como el componente principal que será renderizado cuando la aplicación se inicie.

```
import Navigation from "../src/Navigation.js";

export default function App() { return <Navigation/>; }
```

Figura 52. Captura de pantalla del archivo `App.js`

## 6.2 Detalles de implementación

Aquí se van a mencionar 3 partes del código especialmente relevantes y de cierta complejidad técnica a nivel de implementación.

### 6.2.1. Inicio de sesión

El proceso de inicio de sesión es uno de los pilares fundamentales de nuestra aplicación. Aquí se detallan los mecanismos y algoritmos que garantizan la autenticación segura de los usuarios, incluyendo la validación de credenciales, la gestión de sesiones y la protección contra intentos de acceso no autorizados.

```
const onHandleLogin = async () => {
  if (!isConnected) {
    Alert.alert('Error', 'Parece que ocurrió un problema inesperado. Por favor, inténtalo de nuevo más tarde.');
```

```
    return;
  }
  setError(null);
  setIsLoading(true);

  if (email === "") {
    setError("El correo no puede estar vacío.");
    setIsLoading(false);
    return;
  }
  if (!validateEmail(email)) {
    setError("El correo no es válido.");
    setIsLoading(false);
    return;
  }
  if (password === "") {
    setError("La contraseña no puede estar vacía.");
    setIsLoading(false);
    return;
  }
  if (password.length < 6) {
    setError("La contraseña debe tener al menos 6 caracteres.");
    setIsLoading(false);
    return;
  }

  try {
    const result = await inicioSesion(email, password);
    setIsLoading(false);
    if (!result.success) {
      setError(result.message);
      return;
    }
    console.log("Inicio de sesión correcto");
  } catch (err) {
    setError("Error al iniciar sesión. Comprueba tus credenciales.");
    setIsLoading(false);
    console.error("Error en el inicio de sesión:", err);
  }
};
```

Figura 53. Captura de pantalla de la función que ejecuta el iniciar sesión de la vista `src/views/pages/Login.js`

Esta función se ejecuta al pulsar sobre el botón “Iniciar sesión”. Se hacen todas la validaciones primero, y seguidamente se llama a la función ***inicioSesion(email, password)***, que se encuentra en el controlador “*usuarioController.js*” y hace lo siguiente:

```
export const inicioSesion = async (email, password) => {
  try {
    const usuariosRef = collection(database, 'usuarios');
    const q = query(usuariosRef, where('correo', '=', email));
    const querySnapshot = await getDocs(q);

    if (!querySnapshot.empty) {
      await signInWithEmailAndPassword(auth, email, password);
      const user = auth.currentUser; //Obtener el usuario autenticado y almacenar la sesión en AsyncStorage
      if (user) {
        const session = user;
        await setSession(session);
      }
      return { success: true, message: 'Inicio de sesión correcto' };
    } else {
      return { success: false, message: 'Esta cuenta de usuario no existe...' };
    }
  } catch (error) {
    console.error('Error al iniciar sesión:', error.code);
    throw error;
  }
};
```

Figura 54. Captura de pantalla de la función `inicioSesion` del controlador `src/controllers/usuarioController.js`

Lo que hace esta función mostrada en la Figura 54 es primero buscar en la base de datos un documento de usuario que coincida con el correo electrónico proporcionado. Si encuentra el usuario, intenta iniciar sesión con el correo electrónico y la contraseña usando *signInWithEmailAndPassword*. Si el inicio de sesión es exitoso, almacena la sesión del usuario actual, y si no encuentra el usuario, devuelve un mensaje indicando que la cuenta no existe. En caso de error durante el proceso, se captura y se muestra.

## 6.2.2. La Agrupación y ordenamiento de usuarios por instrumento

El objetivo es mostrar a los usuarios agrupados por su instrumento y ordenados alfabéticamente dentro de cada grupo para que el administrador pueda marcar asistencias dentro de un evento fácilmente. Este proceso no solo organiza los datos de manera lógica y accesible, sino que también mejora significativamente la experiencia del usuario.

- **Agrupación de usuarios:** El código recorre la lista de usuarios y los agrupa según el instrumento que tocan. Si el instrumento del usuario no está en el objeto *usuariosPorInstrumento*, se crea una nueva entrada para ese instrumento:

```
const usuariosPorInstrumentoCopy = {};  
usuarios.forEach(usuario => {  
  if (!usuariosPorInstrumentoCopy[usuario.instrumento]) {  
    usuariosPorInstrumentoCopy[usuario.instrumento] = [];  
  }  
  usuariosPorInstrumentoCopy[usuario.instrumento].push(usuario);  
});
```

Figura 55. Captura de pantalla del componente. *src/views/components/AdminAsistenciaComponent.js*

- **Ordenamiento alfabético:** Una vez que los usuarios están agrupados por instrumento, se ordenan alfabéticamente dentro de cada grupo. Esta ordenación facilita la búsqueda y selección de usuarios dentro de la interfaz de la aplicación:

```
Object.keys(usuariosPorInstrumentoCopy).forEach(instrumento => {  
  usuariosPorInstrumentoCopy[instrumento].sort((a, b) => {  
    return a.nombre.localeCompare(b.nombre);  
  });  
});
```

Figura 56. Captura de pantalla2 del componente. *src/views/components/AdminAsistenciaComponent.js*

- **Visualización estructurada:** El resultado de la agrupación y ordenamiento se presenta en la interfaz de usuario, con cada instrumento como un encabezado y los usuarios listados debajo de su respectivo instrumento. Esto no solo organiza los datos de manera coherente, sino que también mejora la navegabilidad y la experiencia de usuario:

```

<ScrollView style={styles.asistidos}>
  {Object.keys(usuariosPorInstrumento).sort((a, b) => a.localeCompare(b)).map(instrumento => [
    <View key={instrumento}>
      <Text style={styles.instrumentoTitle}>{instrumento}</Text>
      {usuariosPorInstrumento[instrumento].map(usuario => {
        const isAsistente = asistentes.includes(usuario.id);
        const usuarioLogueado = usuario.id === userId;
        const nombre = usuarioLogueado ? 'Tú' : usuario.nombre;
        return (
          <View
            key={usuario.id}
            style={[
              styles.usuarioItem,
              usuario.rol === 'extra' ? styles.extraUsuarioItem : null,
              usuarioLogueado ? { backgroundColor: '#87F5F1' } : null
            ]}
          >
            <View style={styles.usuarioInfo}>
              <View style={styles.row}>
                <Icon
                  name={isAsistente ? 'checkbox' : 'close-circle'}
                  color={isAsistente ? '#52D858' : 'red'}
                  style={styles.icon}
                />
                <Text style={styles.usuarioText} numberOfLines={1} ellipsizeMode='tail'>{nombre}</Text>
              </View>
              <Text style={styles.telefonoText}>{usuario.telefono}</Text>
            </View>
            <TouchableOpacity
              style={styles.button, { backgroundColor: buttonColor[usuario.id] ? buttonColor[usuario.id].toString() : '' }}
              onPress={() => handleToggleAsistencia(usuario.id)}
            >
              <Text style={styles.buttonText}>{buttonText[usuario.id] ? buttonText[usuario.id].toString() : ''}</Text>
            </TouchableOpacity>
          </View>
        );
      })};
    </View>
  ])}
</ScrollView>

```

Figura 57. Captura de pantalla del componente. `src/views/components/AdminAsistenciaComponent.js`

### 6.2.3. Exportar un PDF con los asistentes

En esta sección, explicamos cómo se ha implementado la funcionalidad para exportar un PDF con la lista de asistentes de un evento pasado. Esta funcionalidad es esencial para que los administradores puedan generar informes y compartirlas fácilmente.

- **Preparar el contenido del PDF:** El proceso comienza con la creación del contenido HTML que se convertirá en el PDF. Este contenido se organiza en una estructura que incluye el tipo de evento, la fecha y una lista de asistentes agrupados por instrumento:

```

let content = `
<html>
<head>
  <style>
    body { font-family: Arial, sans-serif; padding: 20px; }
    h1 { text-align: center; text-decoration: underline; }
    h3 { margin-top: 25px; margin-bottom: 2px; }
    .usuario { margin-bottom: 8px; }
    .extra { background-color: #FFEE3; padding: 5px; }
    .nombre { font-weight: 500; } /* Añade esta clase */
  </style>
</head>
<body>
  <h1>${tipoEvento[eventoTipo]} - ${eventoFecha}</h1>

  // Iterar sobre los asistentes y agregar su información al contenido del PDF
  Object.keys(usuariosPorInstrumento).forEach(instrumento => {
    const usuariosDelInstrumento = usuariosPorInstrumento[instrumento].filter(usuario => asistentes.includes(usuario.id));
    if (usuariosDelInstrumento.length > 0) {
      content += `<h3>${instrumento}</h3>`;
      usuariosDelInstrumento.forEach(usuario => {
        content += `<div class="usuario ${usuario.rol === 'extra' ? 'extra' : ''}">
          <span class="nombre">${usuario.nombre}</span> - ${usuario.telefono} ${usuario.rol === 'extra' ? '- ${usuario.rol}' : ''}
        </div>`;
      });
    }
  });

  content += `</body></html>`;

```

Figura 58. Captura de pantalla del componente. `src/views/components/AsistidosEventoComponent.js`

- **Generar el PDF:** Una vez que el contenido HTML está listo, utilizamos la biblioteca *expo-print* para convertirlo en un archivo PDF. Esta biblioteca proporciona una función *printToFileAsync* que toma el HTML y genera un archivo PDF:

```
// Generar el PDF utilizando expo-print
const { uri } = await Print.printToFileAsync({ html: content });
```

Figura 59. Captura de pantalla2 del componente. *src/views/components/AsistidosEventoComponent.js*

- **Guardar y compartir el PDF:** Después de generar el PDF, es importante almacenarlo correctamente y ofrecer opciones para compartirlo. Utilizamos *expo-file-system* para mover el archivo PDF a una ubicación más adecuada con un nombre descriptivo:

```
// Definir el nombre del archivo basado en el evento y la fecha
const fileName = `${tipoEvento[eventoTipo].replace(/ /g, '_')}_${eventoFecha}.pdf`;
const newUri = `${FileSystem.documentDirectory}${fileName}`;

// Mover el archivo PDF a una nueva ubicación con el nombre deseado
await FileSystem.moveAsync({
  from: uri,
  to: newUri
});
```

Figura 60. Captura de pantalla3 del componente. *src/views/components/AsistidosEventoComponent.js*

Finalmente, se usa *expo-sharing* para abrir un diálogo de compartir, permitiendo que el usuario pueda compartir el PDF mediante diversas aplicaciones:

```
// Compartir el PDF utilizando expo-sharing
await Sharing.shareAsync(newUri, {
  mimeType: 'application/pdf',
  dialogTitle: 'Compartir PDF',
  UTI: 'com.adobe.pdf'
});
```

Figura 61. Captura de pantalla4 del componente. *src/views/components/AsistidosEventoComponent.js*



# Capítulo 7

## Pruebas

Este capítulo muestra las pruebas que se han realizado para demostrar el correcto funcionamiento del sistema.

### 7.1 Pruebas de caja negra

Las pruebas realizadas son de caja negra y tienen un carácter funcional. Sirven para probar las entradas y salidas del sistema, sin considerar los procesos internos que ocurren.

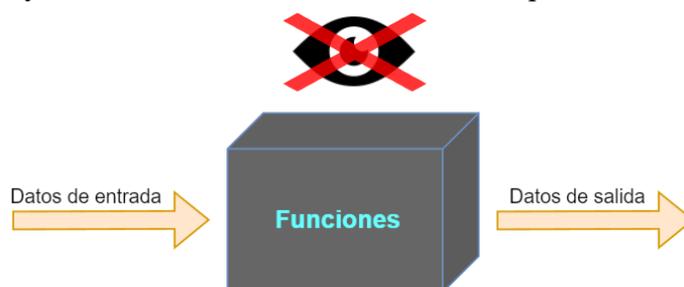


Figura 62. Pruebas de caja negra

A continuación, se adjuntan seis pruebas de caja negra realizadas:

PCN-01: Inicio de sesión con una cuenta no registrada	
<b>Propósito</b>	Verificar que el sistema no permita el inicio de sesión con una cuenta que no está registrada en el sistema
<b>Prerrequisitos</b>	- Acceso al formulario de inicio de sesión de la aplicación - El sistema debe estar operativo
<b>Datos de entrada</b>	Email no registrado y contraseña inventada
<b>Resultado esperado</b>	El sistema debe mostrar un mensaje de error indicando que la cuenta no existe y no debe permitir el acceso al sistema.
<b>Resultado obtenido</b>	<div style="border: 1px solid black; padding: 10px; display: inline-block;"><p style="text-align: center;"><b>Inicio de sesión</b></p><p>manuela@gmail.com</p><p>.....</p><p style="color: red; font-size: small;">Esta cuenta de usuario no existe...</p><p style="text-align: center; background-color: #003366; color: white; padding: 5px;">Iniciar sesión</p></div> <span style="color: green; font-size: 2em; vertical-align: middle;">✓</span>

Tabla 67. Prueba de caja negra "Inicio de sesión incorrecto"

PCN-02: Listado de eventos vacío	
<b>Propósito</b>	Verificar que el sistema muestre un mensaje adecuado cuando no haya eventos nuevos publicados en la pantalla de inicio
<b>Prerrequisitos</b>	- Acceso al sistema y a la pantalla de inicio - No debe haber eventos nuevos publicados en el sistema
<b>Datos de entrada</b>	Acceso a la pantalla de inicio con un usuario registrado
<b>Resultado esperado</b>	El sistema debe mostrar un mensaje indicando que no hay eventos nuevos publicados
<b>Resultado obtenido</b>	

Tabla 68. Prueba de caja negra "Listado de eventos vacío"

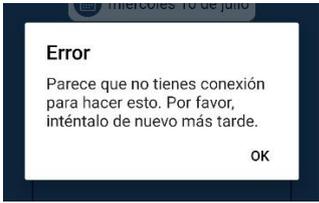
PCN-03: Creación de nuevos eventos sin conexión	
<b>Propósito</b>	Verificar que el sistema no permita la creación de nuevos eventos cuando no hay conexión a Internet y muestre un mensaje adecuado al usuario
<b>Prerrequisitos</b>	- Acceso al sistema y a la funcionalidad de creación de eventos. - Desconexión de Internet o simulación de pérdida de conexión.
<b>Datos de entrada</b>	Datos del evento que se desea crear
<b>Resultado esperado</b>	El sistema debe detectar la falta de conexión a Internet y mostrar un mensaje de error adecuado
<b>Resultado obtenido</b>	

Tabla 69. Prueba de caja negra "Creación de nuevos eventos sin conexión"

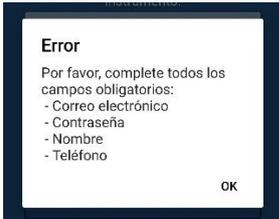
PCN-04: Dar de alta a un nuevo miembro con campos vacíos	
<b>Propósito</b>	Verificar que el sistema no permita dar de alta a un nuevo miembro cuando los campos obligatorios están vacíos y muestre mensajes de error adecuados
<b>Prerrequisitos</b>	Acceso al sistema y a la funcionalidad de registro de nuevos miembros
<b>Datos de entrada</b>	Campos del formulario de registro de nuevos miembros dejados vacíos
<b>Resultado esperado</b>	El sistema debe mostrar mensajes de error indicando que los campos obligatorios no pueden estar vacíos
<b>Resultado obtenido</b>	

Tabla 70. Prueba de caja negra "Registrar miembro con campos vacíos"

PCN-05: Modificación exitosa del perfil de usuario	
<b>Propósito</b>	Verificar que el sistema muestre un mensaje de éxito al usuario después de modificar correctamente su perfil
<b>Prerrequisitos</b>	Acceso al sistema y a la pantalla de perfil
<b>Datos de entrada</b>	Datos del usuario actualizados para modificar los antiguos
<b>Resultado esperado</b>	El sistema debe mostrar un mensaje de éxito indicando que la modificación del perfil se realizó correctamente
<b>Resultado obtenido</b>	

Tabla 71. Prueba de caja negra "Modificación exitosa del perfil de usuario"

PCN-06: Visualización del listado de usuarios registrados	
<b>Propósito</b>	Verificar que el sistema muestre correctamente todos los usuarios registrados agrupados por instrumentos, y que el usuario logueado aparezca destacado visualmente
<b>Prerrequisitos</b>	- Acceso al sistema con un usuario registrado - Usuarios registrados en diferentes instrumentos dentro de la aplicación
<b>Datos de entrada</b>	Acceso a la pantalla de visualización de usuarios registrados
<b>Resultado esperado</b>	Los usuarios deben estar agrupados por instrumentos de manera clara y ordenada. El usuario logueado debe ser fácilmente identificable con un estilo visual diferente que lo destaque del resto de usuarios
<b>Resultado obtenido</b>	

Tabla 72. Prueba de caja negra "Listado de usuarios registrados"

## 7.2 Otras pruebas realizadas

Además de las pruebas de la sección anterior que hicimos para verificar diferentes partes de la aplicación y que el valor esperado fuera igual que el real, también se decidió hacer pruebas más cercanas a cómo las personas reales usarían la aplicación.

Para eso, elegimos a tres personas al azar de la banda de música y se les pidió que usaran la aplicación durante un día completo. Queríamos ver cómo interactuaban con ella en situaciones reales: si encontraban alguna dificultad para navegar por las funciones, si notaban algún problema que las pruebas automáticas podrían no haber detectado, o si tenían sugerencias para mejorar la experiencia.

Estas pruebas ayudaron a obtener comentarios valiosos y observar cómo la aplicación se comporta en el mundo real, con usuarios reales. Así, combinamos la precisión de las pruebas automatizadas con la perspectiva humana y la experiencia práctica, asegurándonos de que la aplicación no solo funcione bien en teoría, sino también en la práctica cotidiana de nuestros usuarios.

Pero, además, con estas pruebas se ha extraído que todas las tareas que ofrece la aplicación funcionan correctamente, pero también se han identificado posibles mejoras para el futuro. Entre estas mejoras se incluyen la integración de un calendario visual donde se puedan ver los eventos, la inclusión en los perfiles de los ensayos y actuaciones a los que ha asistido cada miembro, y la capacidad de integrar documentos con el repertorio a tocar. Estos cambios potenciales prometen enriquecer aún más la funcionalidad y usabilidad de la aplicación, alineándose con las necesidades y deseos de los usuarios.

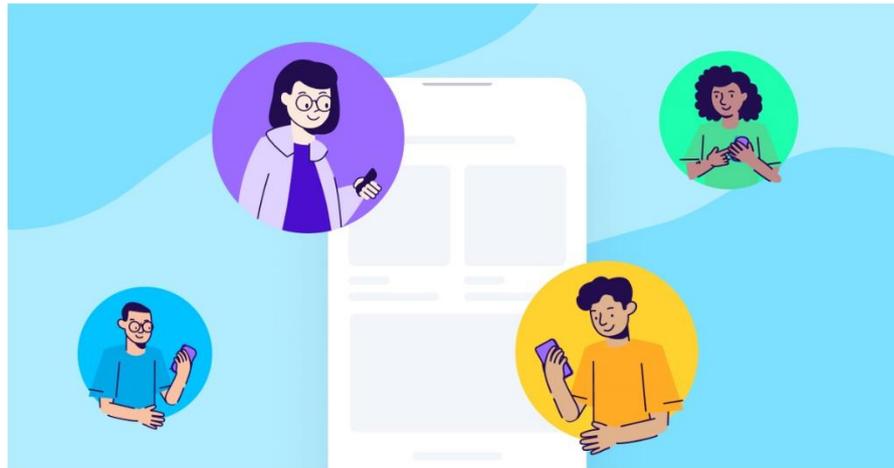


Figura 63. Pruebas con usuarios finales reales [17]



# Capítulo 8

## Documentación

En este capítulo se habla sobre los dos manuales que se han preparado para que se pueda entender mejor cómo usar y poner en marcha la aplicación móvil BanDate.

Ambos manuales están diseñados para que tengan toda la información que se necesita, ya sea para usar la aplicación de manera efectiva o para desplegarla correctamente en un entorno técnico.

### 8.1 Manual de usuario

El manual de usuario explica cómo funciona la aplicación desde el punto de vista de los distintos usuarios que hay. Te guía paso a paso para que se sepa cómo utilizar todas las funciones y cómo sacarle el máximo provecho.

#### 8.1.1. Todos los usuarios

En primer lugar, se muestran las pantallas y las tareas básicas que todos los usuarios pueden llevar a cabo, independientemente de su rol.

- **Inicio de sesión:** Será la pantalla que aparezca al iniciar la aplicación. En ella se encuentra el formulario para introducir los datos de usuario registrado, la opción de iniciar sesión y la de recuperar contraseña.

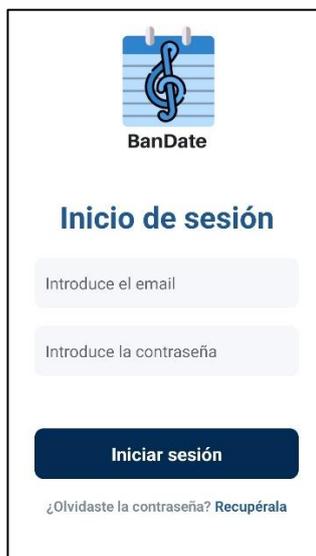


Figura 64. Pantalla “Inicio de sesión”

- **Recuperar contraseña:** En la pantalla para recuperar la contraseña del usuario se encuentra el formulario con el campo para introducir el correo electrónico, y las opciones de enviar el correo para cambiar la contraseña y la opción para volver a la pantalla de inicio de sesión.



Figura 65. Pantalla “Recuperar contraseña”

No existe pantalla de registro, ya que únicamente podrán crear las cuentas los administradores como se verá más adelante. Esto es para garantizar privacidad y seguridad, y no se pueda registrar ninguna persona externa a la banda de música de Segovia.

- **Barra de navegación inferior:** En cuanto a la barra de navegación, todos los usuarios tienen la misma, salvo los administradores que pueden acceder a unas pantallas extra. La barra de navegación para músicos y jefes de cuerda es la siguiente:



Figura 66. Barra de navegación de los músicos y jefes de cuerda

Mediante esta barra de navegación, los usuarios pueden acceder a las pantallas “Inicio” y “Listados”.

- **Visualizar perfil:** Una vez se ha iniciado sesión, se puede acceder al perfil de usuario desde cualquier pantalla principal de la aplicación, pulsando sobre el icono de usuario que se muestra en la Figura 67:



Figura 67. Barra superior de la aplicación

En esta pantalla se podrá visualizar los datos del usuario *logueado*, cerrar sesión o modificar algunos datos si es necesario, como se puede ver en la Figura 68:



Figura 68. Pantalla "Perfil"

- **Editar perfil:** Desde el perfil se pueden modificar los datos, y saldrá un modal como el de la Figura 69. Este formulario te muestra los datos actuales del usuario y si desea cambiar alguno nuevo, puede hacerlo pulsando en el botón guardar cambios. Sino, en el botón cancelar para cerrar el modal y cancelar el cambio de datos.



Figura 69. Modal "Editar perfil"

## 8.1.2. Manual del Administrador

El administrador tiene todos los permisos y acceso a todas las funcionalidades de la aplicación.

Como se había mencionado anteriormente, la barra de navegación es distinta a la del resto de usuarios y pueden acceder a pantallas exclusivas:



Figura 70. Barra de navegación del administrador

Este puede acceder a las pantallas “Inicio”, “Creación de eventos”, “Registro de cuentas” y “Listados”.

- **Pantalla Inicio:** Esta pantalla está disponible para todos los usuarios y es donde se visualizan los nuevos eventos publicados, pero los administradores tienen acciones adicionales que pueden realizar.



Figura 71. Pantalla “Inicio” del administrador

Las acciones que puede realizar este, aparte de acceder a los detalles de un evento específico (se explicará al final), puede eliminarlo o editarlo como se muestran en la Figura 72 y Figura 73 respectivamente:



Figura 72. Pantalla “eliminar evento”



Figura 73. Pantalla “editar evento”

- Creación de eventos:** En esta pantalla el admin puede crear nuevos eventos, tanto ensayos como actuaciones. Deberá indicar todos los campos que se muestran, si elige que el tipo del evento es “actuación”, se revelará un nuevo campo para indicar el nombre del evento (como se ve en la Figura 74).



Figura 74. Pantalla “Crear eventos” del administrador

- **Registro de cuentas:** A través de esta pantalla, el administrador puede dar de alta a nuevos miembros de la banda, e incluso a extras para ocasiones especiales. Deberá indicar todos los campos que se muestran, y lo más importante, seleccionar un instrumento y rol adecuados para el usuario a registrar:

Figura 75. Pantalla “Registro de cuentas” del administrador

Los extras que se registren en la aplicación no podrán acceder mediante el login, simplemente quedan guardados sus datos por si necesita contar con ellos en eventos que se requiera de personal.

Figura 76. Modal “Registro de extras” del administrador

- **Listados:** Se muestran dos botones en esta pantalla de listados y es común para todos los usuarios. Pueden visualizar tanto un listado con todos los usuarios registrados en la aplicación (extras y miembros), como de todos los eventos ya realizados.



Figura 77. Pantalla “Listados”

En el **listado de usuarios**, el administrador tiene el poder de dar de baja a cualquier usuario, y ascender o descender a jefe de cuerda o músico respectivamente.



Figura 78. Ventana “Listado de usuarios” del administrador

En el **listado de eventos pasados** el administrador puede eliminar un evento específico, y a la vez que visualiza los asistentes de este, puede exportar un PDF con los asistentes y así compartirlo con quien desee.



Figura 79. Ventana "Listado de eventos pasados" del administrador



Figura 80. Ventana asistidos del administrador

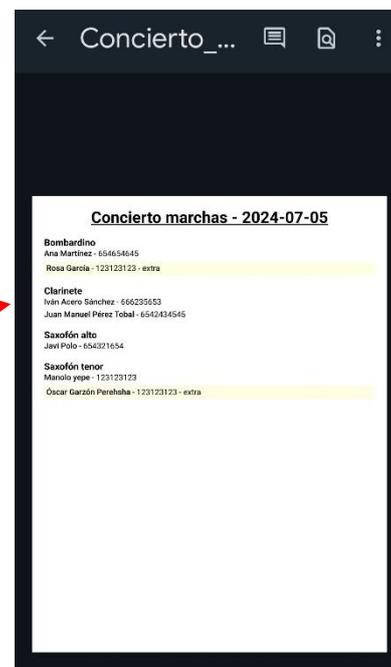


Figura 81. Captura del PDF exportado de asistidos

Volviendo a la “Pantalla Inicio”, se va a mostrar la ventana de **detalles del evento**, ya que esta contiene la funcionalidad más importante de esta aplicación.

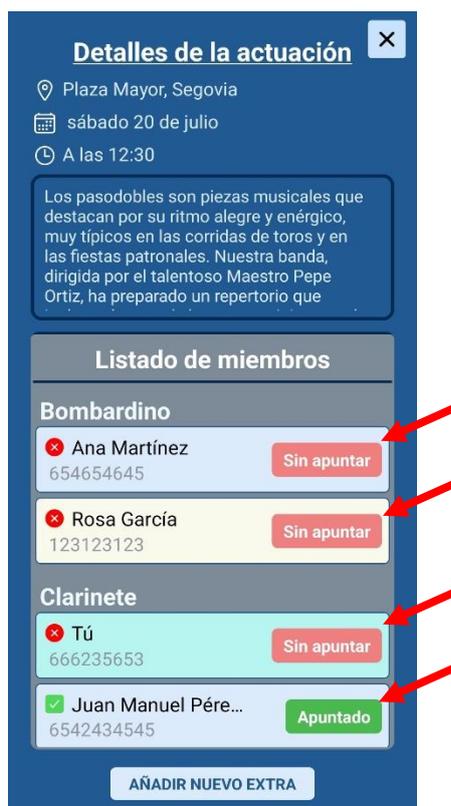


Figura 82. Ventana “Detalles del evento” del administrador

En esta ventana, el administrador puede marcar y desmarcar cualquier asistencia de cualquier miembro o extra que esté registrado en el sistema pulsando los botones que aparecen al lado de cada usuario (incluyéndose a él mismo). Para ver todos los miembros, aunque no se muestre en la Figura 82, hay un scroll que al deslizar el dedo sube y baja el listado de miembros.

Además, podrá añadir directamente desde esta ventana un nuevo extra, pero es la misma funcionalidad y formulario que se mostró en la Figura 76.

### 8.1.3. Manual del Músico

El músico tiene accesos más restringidos, pero aun así puede hacer todo lo mencionado en la sección 8.1.1, e incluso acceder a la pantalla de listados (Figura 77).

- **Pantalla Inicio:** Esta pantalla es igual que la que ya se explicó para los administradores, pero no tiene las funcionalidades de eliminado y edición de eventos:

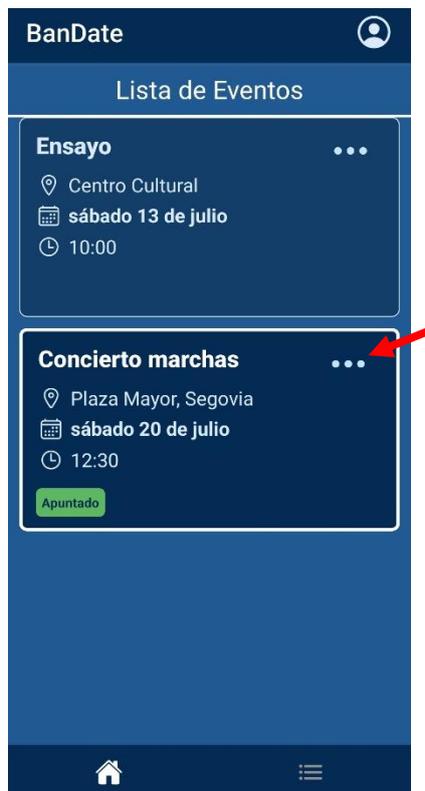


Figura 83. Pantalla "Inicio"

Si se accede a la ventana de detalles, se puede observar que la vista para marcar asistencia cambio pudiendo solamente marcar/desmarcar su asistencia con el botón inferior, y sólo podrá ver a los miembros de su cuerda:

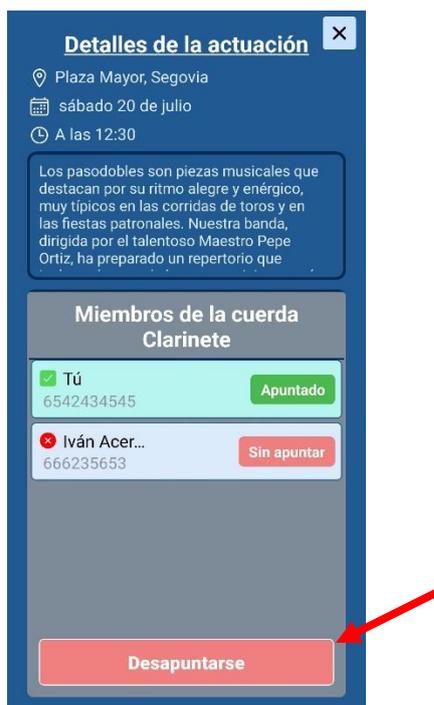


Figura 84. Ventana "Detalles del evento" del músico

Accediendo a la pantalla “Listados”, puede ver lo mismo que lo mostrado en la Figura 77.

Para el **listado de usuarios**, el músico únicamente podrá visualizarlo sin poder realizar ninguna acción sobre este:



Figura 85. Ventana “Listado de usuarios”

En el **listado de eventos pasados** el músico puede seleccionar evento específico, y visualizar los asistentes de este, sin poder ni eliminarlo ni exportar un PDF.

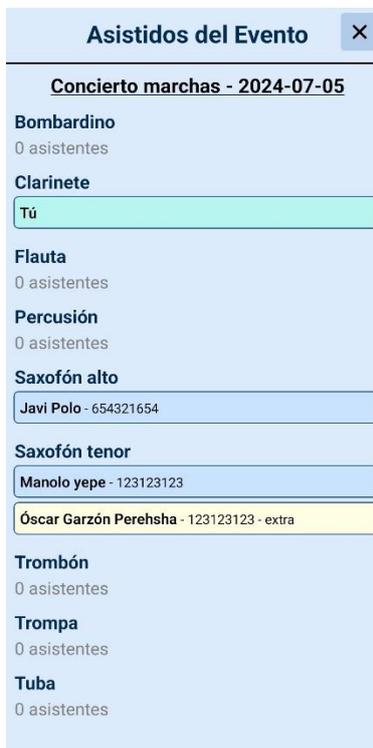


Figura 86. Ventana “Listado de eventos pasados”

## 8.1.4. Manual del Jefe de Cuerda

Este usuario no se queda atrás, es un punto intermedio entre el administrador y el músico.

El jefe de cuerda de un instrumento específico tiene exactamente los mismos accesos a pantallas y funcionalidades que el músico, solo que, si accede a la ventana de detalles, la vista para marcar/desmarcar asistencias cambia un poco:



Figura 87. Ventana “Detalles del evento” del jefe de cuerda

El jefe de cuerda en esta ventana puede marcar y desmarcar cualquier asistencia pulsando los botones que aparecen al lado de cada usuario (incluyéndose a él mismo) como el administrador, pero únicamente para los miembros de su misma cuerda (mismo instrumento).

Además, podrá añadir directamente desde esta ventana un nuevo extra como los administradores por si es necesario.

## 8.2 Manual de despliegue

El manual de despliegue te muestra los pasos técnicos necesarios para poner en funcionamiento la aplicación. Aquí se explicará cómo instalar y configurar todo lo que se necesita para que la aplicación esté lista en Google Play Store (para Android) y Apple Store (para iOS).

Aunque no se desplegó realmente la aplicación en las distintas tiendas de ambos sistemas operativos debido a los costes que suponía, se generó un APK simulándolo (se explicará cómo se ha hecho al final).

Aun así, se va a detallar por pasos y de forma breve que hay que hacer en caso de querer subir la aplicación móvil a las distintas tiendas para que cualquiera pueda descargarla.

### 8.2.1. Prerrequisitos

En esta sección se detallan los softwares y cuentas necesarias para el despliegue de la aplicación:

- **Software necesario:**
  - Node.js (recomendado usar la última versión estable)
  - npm (normalmente viene con Node.js)
  - Expo CLI (npm install -g expo-cli)
  - Git
  - Java Development Kit (JDK) para despliegue en Android
  - Xcode para despliegue en iOS (necesario un MAC)
- **Cuentas necesarias:**
  - Cuenta de Firebase
  - Cuenta en Google Play Store (para Android)
  - Cuenta en Apple Developer (para iOS)

### 8.2.2. Configuración inicial

Es el primer paso práctico para preparar el entorno de desarrollo. Para ello hay que:

- **Clonar el repositorio:**
  - `git clone <URL_DEL_REPOSITORIO>`
  - `cd <NOMBRE_DEL_PROYECTO>`
- **Instalar dependencias:**
  - Mediante el comando: `npm install`

### 8.2.3. Configuración de la base de datos

Esta sección describe cómo crear un proyecto en *Firebase*, añadir la aplicación para las plataformas Android e iOS, y descargar y configurar los necesarios para la integración con *Firebase*:

- **Crear un proyecto en Firebase:**
  1. Iniciar sesión en <https://firebase.google.com/?hl=es> y acceder a Firebase Console para crear un nuevo proyecto (Figura 88)



Figura 88. Crear nuevo proyecto en la consola de Firebase

- **Añadir apps a Firebase:**
  - **Para Android:**
    1. Selecciona "Añadir app" y elige "Android".
    2. Sigue los pasos para registrar tu aplicación, usando el paquete *com.tuapp.nombre*.
    3. Descarga el archivo *google-services.json* y colócalo en el directorio *android/app*.
  - **Para iOS:**
    1. Selecciona "Añadir app" y elige "iOS".
    2. Sigue los pasos para registrar tu aplicación, usando el paquete *com.tuapp.nombre*.
    3. Descarga el archivo *GoogleService-Info.plist* y colócalo en el directorio *ios/<NOMBRE\_DEL\_PROYECTO>/*.

#### 8.2.4. Configuración de Expo

- **Inicializar Expo:**
  - Mediante el comando *expo init <NOMBRE\_DEL\_PROYECTO>*
- **Configurar Firebase en Expo:** Asegúrate de seguir las instrucciones específicas para integrar Firebase con Expo usando *expo-firebase*, etc.

#### 8.2.5. Despliegue en Android

- **Configurar las credenciales:**
  1. Sigue las instrucciones oficiales de [Expo](#) para configurar las credenciales.
- **Construir la aplicación:**
  - Con ayuda del comando: *expo build:android*
- **Subir la APK/AAB a Google Play Console:**
  1. Ve a [Google Play Console](#).
  2. Sigue los pasos para subir y publicar la aplicación.

#### 8.2.5. Despliegue en iOS

- **Configurar las credenciales:**
  1. Sigue las instrucciones oficiales de [Expo](#) para configurar las credenciales.
- **Construir la aplicación:**
  - Con ayuda del comando: *expo build:ios*

- **Subir la IPA a App Store Connect:**
  1. Usa *Transporter* (disponible en Mac App Store) para subir el archivo “.ipa“ a App Store Connect.
  2. Ve a App Store Connect y sigue los pasos para subir y publicar la aplicación.

## 8.2.6. Generación de APK simple

Para las pruebas con usuarios reales, en vez de desplegarlo como se ha mostrado arriba, **se generó un APK** (Android Package Kit), un archivo que contiene toda la aplicación que se quiere instalar en un dispositivo Android de una manera fácil y rápida.

Sin embargo, la limitación que se ha presentado es que no puede ser utilizado en dispositivos iOS, ya que estos requieren un proceso diferente. En su lugar, **Expo Go** ha facilitado la prueba simultánea en ambos sistemas operativos, permitiendo una experiencia de usuario fluida sin necesidad de generar archivos específicos para iOS.

Los pasos que se han realizado para generar el **APK para Android** son los siguientes:

- Realizar la misma configuración inicial que la explicada en el apartado 8.2.2.
- Crear un proyecto en Firebase desde el Firebase Console, y elegimos el modo “Web” para la integración del SDK de Cloud Firestore en la aplicación.



Figura 89. Crear un proyecto en la consola de Firebase del tipo Web

- Una vez hecho esto se elige un nombre (en este caso BanDate), se registra la app y se genera la configuración de la base de datos, la cual se debe añadir en el archivo “*src/config/firebase.js*” del proyecto:

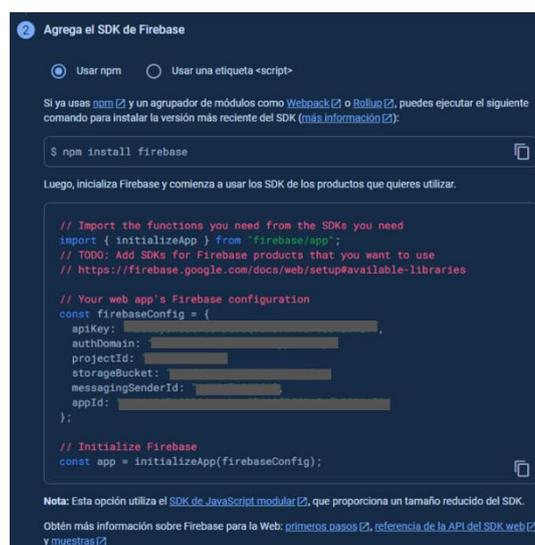


Figura 90. Configuración de Firebase para conectar la base de datos con la aplicación

Es importante no mostrar esta configuración al público porque contiene información sensible, como claves de API y credenciales de acceso. Exponer esta información podría comprometer la seguridad de la aplicación y los datos de los usuarios, por eso es crucial mantener esta información privada y segura para proteger la integridad y el funcionamiento adecuado de tu aplicación.

- Una vez hecho todo esto, lo siguiente que hay que hacer, es ejecutar el comando ***eas build:configure***, el cual es utilizado para configurar opciones específicas relacionadas con la construcción y despliegue de la aplicación:

```
→ buildTest git:(main) eas build:configure
💡 The following process will configure your iOS and/or Android project to be compatible with EAS Build. These changes only apply to your local project files and you can safely revert them at any time.

✓ Which platforms would you like to configure for EAS Build? > All

🚀 Your project is ready to build.

- Run eas build when you are ready to create your first build.
- Once the build is completed, run eas submit to upload the app to app stores.
```

Figura 91. Ejecución del comando “*eas build:configure*”

- De seguido hay que asegurarse de que en el archivo “*eas.json*” aparezca esto para poder generar el APK correctamente:

```
{
  "cli": {
    "version": ">= 6.0.0"
  },
  "build": {
    "development": {
      "developmentClient": true,
      "distribution": "internal"
    },
    "preview": {
      "distribution": "internal",
      "android": {
        "buildType": "apk"
      }
    },
    "production": {}
  },
  "submit": {
    "production": {}
  }
}
```

Figura 92. Contenido del archivo “*eas.json*”

- Por último y para terminar la generación del APK, hay que asegurarse que estén guardados todos los cambios hechos, y hacer un *push*. Con esto, hay que ejecutar este último comando:

```
→ buildTest git:(main) eas build -p android --profile preview
✓ Using remote Android credentials (Expo server)
✓ Using Keystore from configuration: Build Credentials mmCmecGoIL (default)

Compressing project files and uploading to EAS Build. Learn more
# Uploading to EAS Build (3.5 MB / 4.3 MB)
```

Figura 93. Ejecución del comando “*eas build -p android --profile preview*”

Cuando haya finalizado la ejecución del comando anterior, iremos a nuestra cuenta de Expo desde un dispositivo Android y ya podremos descargar la aplicación por medio de la APK generada:

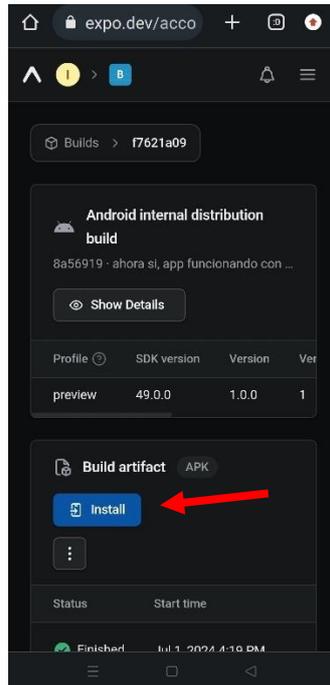


Figura 94. APK generada con Expo

## 8.2.7. Ejecución de la aplicación en línea con Expo Go

Ejecutar la aplicación en línea con **Expo Go** (anteriormente conocido como Expo Client) implica utilizar la aplicación móvil Expo Go, disponible en Android e iOS, para cargar y ejecutar el proyecto desarrollado con Expo de manera directa desde tu entorno de desarrollo. Esto permite ver y probar los cambios en tiempo real sin necesidad de compilar o generar un archivo APK o IPA separado como se ha explicado en todos los pasos de arriba.

- En primer lugar, para arrancar el proyecto basta con realizar lo mismo de antes, lo explicado en el apartado 8.2.2, y seguidamente la integración del SDK de Cloud Firestore en la aplicación explicado en el apartado 8.2.6.

Ya hecho esto, solamente hay que ejecutar el comando `npx expo start` y escanear el código QR generado con la aplicación Expo Go desde el dispositivo móvil.



Figura 95. Ejecución de la aplicación con Expo Go



# Capítulo 9

## Conclusiones

En conclusión, en este trabajo se presenta una herramienta software móvil multiplataforma totalmente funcional capaz de cumplir con todos los requisitos que se capturaron en los distintos incrementos del proyecto. La herramienta, que ha sido probada con usuarios reales, ha demostrado ser eficiente, accesible y usable. Esta herramienta, que ha sido implementada para usarse en un contexto real, será desplegada en breve para dar servicio a los distintos usuarios de la Unión Musical Segoviana.

### 9.1 Valoración personal

Después de completar este Trabajo de Fin de Grado, siento una profunda satisfacción por haber desarrollado una aplicación móvil que no solo amplía mis habilidades técnicas, sino que también por el impacto directo que va a tener en la banda de música a la que pertenezco. Este proyecto nació de la motivación de contribuir de manera significativa a mi banda de mi ciudad, Segovia, enfrentando el desafío de crear una herramienta útil y accesible para todos sus integrantes.

A lo largo de este proceso he experimentado tanto momentos de aprendizaje como de superación personal. Desde el dominio de nuevas tecnologías hasta la gestión eficiente del tiempo, cada obstáculo superado ha fortalecido mi determinación y capacidad para poder enfrentarme a futuros desafíos.

Mirando hacia adelante, mi aspiración es que esta aplicación pueda estar disponible en las tiendas Play Store y App Store, facilitando su uso a un público más amplio. Este trabajo no solo representa el fin de mi formación académica en el desarrollo de software, sino también un testimonio de perseverancia y dedicación que estoy seguro me servirá en mi crecimiento profesional y personal.

En definitiva, este proyecto ha sido una experiencia enriquecedora que ha consolidado mis conocimientos técnicos y me ha brindado la oportunidad de hacer una contribución valiosa a mi comunidad musical. Estoy emocionado por el futuro de esta aplicación y confiado en que seguirá evolucionando para beneficiar a quienes la utilicen.

## 9.2 Líneas de trabajo futuras

Aunque la implementación actual de la aplicación cumple correctamente con las funcionalidades previstas, se han identificado diversas áreas de mejora que podrían potenciar su rendimiento y utilidad en el futuro. A continuación, se detallan algunas propuestas de desarrollo que podrían enriquecer significativamente la experiencia del usuario.

### 9.2.1. Integración de un calendario visual

Una de las mejoras más inmediatas sería la incorporación de un calendario visual. Esta funcionalidad permitiría a los usuarios visualizar de manera clara y ordenada todos los eventos programados. Un calendario interactivo no solo facilitaría la planificación de ensayos y actuaciones, sino que también permitiría a los miembros de la banda gestionar mejor su tiempo y compromisos. Con esta herramienta, los usuarios podrían ver de un vistazo todas las actividades futuras, lo que redundaría en una mejor organización y coordinación.

### 9.2.2. Inclusión de ensayos y actuaciones en los perfiles de usuario

Otra mejora importante sería la inclusión de un historial de ensayos y actuaciones en los perfiles de los miembros de la banda. Esta funcionalidad proporcionaría un registro detallado de la participación de cada usuario en las actividades de la banda, permitiendo un seguimiento más preciso de la implicación y compromiso de los miembros. Además, este historial podría ser útil para evaluar el desempeño individual y colectivo, así como para planificar futuros eventos basándose en la disponibilidad y experiencia de los miembros.

### 9.2.3. Integración de documentos

La capacidad de integrar documentos con el repertorio a interpretar es otra mejora clave. Esta funcionalidad permitiría a los miembros de la banda acceder fácilmente a partituras, anotaciones y otros documentos relevantes directamente desde la aplicación. Con esta herramienta, se facilitaría la preparación de ensayos y actuaciones, ya que todos los miembros tendrían acceso a la información necesaria en un solo lugar. Esta integración no solo ahorraría tiempo, sino que también mejoraría la precisión y coherencia en la interpretación musical.

### 9.2.4. Sistema de notificaciones personalizado

Se propone también la implementación de un sistema de notificaciones personalizado. Este sistema alertaría a los usuarios sobre la creación de nuevos eventos y otras actividades importantes. Las notificaciones automáticas mejorarían significativamente la comunicación interna, asegurando que todos los miembros estén al tanto de las últimas actualizaciones y cambios en el calendario. Con esta funcionalidad, se reducirían los malentendidos y se mejoraría la puntualidad y asistencia a los ensayos y actuaciones.

### 9.2.5. Expansión de la aplicación

Una mejora más ambiciosa sería la expansión de la aplicación para permitir su uso por otras bandas. Esta expansión implicaría el desarrollo de un sistema de registro abierto, donde los administradores pudieran crear nuevos grupos e invitar a usuarios mediante un enlace o código de invitación. Este cambio haría que la aplicación fuera más versátil y accesible a un público más amplio, permitiendo que diversas bandas de música se beneficien de sus funcionalidades. La posibilidad de que múltiples bandas utilicen la misma plataforma promovería el intercambio de conocimientos y recursos entre diferentes agrupaciones musicales.

# Referencias (webgrafía)

[1] **BandMule**. Disponible en: <https://www.bandmule.com/>.  
Último acceso: 10 de julio de 2024.

[2] **Glissandoo**. Disponible en: <https://glissandoo.com/>.  
Último acceso: 8 de julio de 2024.

[3] **Metodología iterativa-incremental**. Disponible en:  
<https://blog.jmbeas.es/2019/09/22/desarrollo-iterativo-e-incremental/>.  
Último acceso: 9 de julio de 2024.

[4] **Cloud Firestore | Firebase Google**. Disponible en:  
<https://firebase.google.com/docs/firestore>.  
Último acceso: 11 de julio de 2024.

[5] **Firebase Authentication | Firebase Google**. Disponible en:  
<https://firebase.google.com/docs/auth>.  
Último acceso: 11 de julio de 2024.

[6] **Patrón de diseño MVC**. Disponible en:  
<https://www.precognis.com/blog/modelo-vista-controlador/>.  
Último acceso: 10 de julio de 2024.

[7] **Documentación GitHub**. Disponible en: <https://docs.github.com/es>.  
Último acceso: 12 de julio de 2024.

[8] **Tutorial de cómo generar un APK con Expo**.  
<https://youtu.be/FBv4PrW5wqY?si=qD6ZRYEkknMZNy04>.  
Último acceso: 9 de julio de 2024.

[9] **Curso React-Native**. <https://youtu.be/aIstXVQRDbM?si=qfpFYWzFfshsu9GK>.  
Último acceso: 11 de julio de 2024.

[10] **Generar PDF con Expo**. Disponible en:  
<https://youtu.be/6R4NH42e8gM?si=wKxGp53Mvd0Ials5>.  
Último acceso: 12 de julio de 2024.

[11] **Documentación React Native**. Disponible en: <https://reactnative.dev>.  
Último acceso: 10 de julio de 2024.

[12] **Iconos de la librería react-native-vector-icons**. Disponible en:  
<https://oblador.github.io/react-native-vector-icons/>.  
Último acceso: 11 de julio de 2024.

[13] **Documentación Asynstorage.** Disponible en:  
<https://reactnative.dev/docs/asyncstorage>.  
Último acceso: 12 de julio de 2024.

[14] **Documentación Expo.** Disponible en: <https://expo.dev>.  
Último acceso: 8 de julio de 2024.

[15] **Documentación para integrar Firebase en React-native.** Disponible en:  
<https://www.freecodecamp.org/espanol/news/como-construir-una-aplicacion-react-native-e-integrarla-con-firebase/>.  
Último acceso: 9 de julio de 2024.

[16] **Documentación Netinfo.** Disponible en:  
<https://docs.expo.dev/versions/latest/sdk/netinfo/>.  
Último acceso: 10 de julio de 2024.

[17] **Imagen usuarios.** Disponible en:  
<https://www.appsflyer.com/es/resources/reports/app-user-acquisition/>  
Último acceso: 9 de julio de 2024.

[18] **Planificación y estimación.** Disponible en:  
[https://administracionelectronica.gob.es/pae\\_Home/pae\\_Documentacion/pae\\_Metodologia/pae\\_Metrica\\_v3.html#.UoyQeeJ0WBo](https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodologia/pae_Metrica_v3.html#.UoyQeeJ0WBo)  
Último acceso: 9 de julio de 2024.

# Índice de figuras

Figura 1. Imagen de BandMule [1] .....	11
Figura 2. Imagen de Glissadoo [2].....	12
Figura 3. Visual Studio Code .....	14
Figura 4. JavaScript.....	14
Figura 5. Expo.....	15
Figura 6. React Native .....	16
Figura 7. Firebase.....	16
Figura 8. Draw.io .....	17
Figura 9. MicrosoftProject .....	18
Figura 10. GitHub .....	18
Figura 11. Word.....	19
Figura 12. Metodología de trabajo [3] .....	21
Figura 13. Tareas con tiempos estimados .....	29
Figura 14. Diagrama de Gantt estimado del proyecto .....	30
Figura 15. Excepciones en el calendario laboral del proyecto estimado .....	31
Figura 16. Tareas con tiempos reales .....	34
Figura 17. Diagrama de Gantt real del proyecto .....	35
Figura 18. Excepciones en el calendario laboral del proyecto real.....	36
Figura 19. Árbol de características de la aplicación .....	40
Figura 20. Diagrama de relación entre actores .....	40
Figura 21. Diagrama de casos de uso del actor Usuario .....	43
Figura 22. Diagrama de casos de uso del actor Administrador.....	43
Figura 23. Diagrama de casos de uso del actor Músico.....	43
Figura 24. Diagrama de casos de uso del actor Jefe de cuerda .....	44
Figura 25. Modelo Entidad-Relación.....	72
Figura 26. Patrón de diseño MVC [6].....	76
Figura 27. Diagrama de arquitectura lógica simplificada.....	76
Figura 28. Diagrama de arquitectura lógica detallada .....	77
Figura 29. Diagrama de arquitectura física .....	79
Figura 30. Diagrama de secuencia – Crear evento.....	80
Figura 31. Diagrama de secuencia – Consultar nuevos eventos publicados.....	81
Figura 32. Diagrama de secuencia – Dar de alta a nuevos miembros de la banda ...	82
Figura 33. Paleta de colores de BanDate .....	96
Figura 34. Logo de la aplicación.....	96
Figura 35. Estructura jerárquica de datos en Cloud Firestore.....	97
Figura 36. Estructura de documento en Cloud Firestore .....	98
Figura 37. Estructura de documento en Cloud Firestore .....	98
Figura 38. Captura de pantalla del modelo-esquema de la colección Eventos.....	99
Figura 39. Captura de pantalla de la estructura del proyecto .....	101
Figura 40. Captura de pantalla de la carpeta assets.....	102
Figura 41. Icono Trompa.png.....	102

Figura 42. Captura de pantalla de la carpeta src .....	103
Figura 43. Captura de pantalla de la carpeta src/models.....	103
Figura 44. Captura de pantalla de la carpeta src/views.....	103
Figura 45. Captura de pantalla de la carpeta src/controllers .....	103
Figura 46. Captura de pantalla de la carpeta src/config .....	104
Figura 47. Captura de pantalla del archivo src/config/config.js .....	104
Figura 48. Captura de pantalla de la carpeta src/styles .....	104
Figura 49. Captura de pantalla del archivo src/Navigation.js (estado del usuario y conectividad a Internet) .....	105
Figura 50. Captura de pantalla del archivo src/Navigation.js (método principal) ..	106
Figura 51. Captura de pantalla del archivo src/Navigation.js (barra de navegación) .....	107
Figura 52. Captura de pantalla del archivo App.js .....	107
Figura 53. Captura de pantalla de la función que ejecuta el iniciar sesión de la vista src/views/pages/Login.js.....	108
Figura 54. Captura de pantalla de la función inicioSesión del controlador src/controllers/usuarioController.js.....	108
Figura 55. Captura de pantalla del componente. src/views/components/AdminAsistenciaComponent.js.....	109
Figura 56. Captura de pantalla2 del componente. src/views/components/AdminAsistenciaComponent.js.....	109
Figura 57. Captura de pantalla3 del componente. src/views/components/AdminAsistenciaComponent.js.....	110
Figura 58. Captura de pantalla del componente. src/views/components/AsistidosEventoComponent.js .....	110
Figura 59. Captura de pantalla2 del componente. src/views/components/AsistidosEventoComponent.js .....	111
Figura 60. Captura de pantalla3 del componente. src/views/components/AsistidosEventoComponent.js .....	111
Figura 61. Captura de pantalla4 del componente. src/views/components/AsistidosEventoComponent.js .....	111
Figura 62. Pruebas de caja negra .....	113
Figura 63. Pruebas con usuarios finales reales [17].....	116
Figura 64. Pantalla “Inicio de sesión”.....	118
Figura 65. Pantalla “Recuperar contraseña” .....	119
Figura 66. Barra de navegación de los músicos y jefes de cuerda.....	119
Figura 67. Barra superior de la aplicación .....	119
Figura 68. Pantalla “Perfil” .....	120
Figura 69. Modal “Editar perfil” .....	120
Figura 70. Barra de navegación del administrador .....	121
Figura 71. Pantalla “Inicio” del administrador .....	121
Figura 72. Pantalla “eliminar evento”	
Figura 73. Pantalla “editar evento”	122
Figura 74. Pantalla “Crear eventos” del administrador .....	122
Figura 75. Pantalla “Registro de cuentas” del administrador .....	123
Figura 76. Modal “Registro de extras” del administrador .....	123
Figura 77. Pantalla “Listados” .....	124
Figura 78. Ventana “Listado de usuarios” del administrador.....	124
Figura 79. Ventana “Listado de eventos pasados” del administrador .....	125

Figura 80. Ventana asistidos del administrador	
Figura 81. Captura del PDF exportado de asistidos .....	125
Figura 82. Ventana “Detalles del evento” del administrador .....	126
Figura 83. Pantalla “Inicio” .....	127
Figura 84. Ventana “Detalles del evento” del músico .....	127
Figura 85. Ventana “Listado de usuarios” .....	128
Figura 86. Ventana “Listado de eventos pasados” .....	128
Figura 87. Ventana “Detalles del evento” del jefe de cuerda .....	129
Figura 88. Crear nuevo proyecto en la consola de Firebase .....	131
Figura 89. Crear un proyecto en la consola de Firebase del tipo Web .....	132
Figura 90. Configuración de Firebase para conectar la base de datos con la aplicación .....	132
Figura 91. Ejecución del comando “eas build:configure” .....	133
Figura 92. Contenido del archivo “eas.json” .....	133
Figura 93. Ejecución del comando “eas build -p android --profile preview .....	133
Figura 94. APK generada con Expo .....	134
Figura 95. Ejecución de la aplicación con Expo Go .....	134

# Índice de tablas

Tabla 1. Tabla comparativa del proyecto con otras apps .....	13
Tabla 2. Entradas externas .....	24
Tabla 3. Criterios de evaluación de entradas externas y consultas .....	24
Tabla 4. Salidas externas .....	25
Tabla 5. Criterios de evaluación de salidas externas.....	25
Tabla 6. Consultas .....	25
Tabla 7. Ficheros lógicos internos.....	25
Tabla 8. Criterios de evaluación de ficheros lógicos internos y externos. ....	26
Tabla 9. Ficheros lógicos externos .....	26
Tabla 10. Correspondencias y puntos de función sin ajustar (PFNA) .....	26
Tabla 11. Factores de complejidad y puntuación .....	27
Tabla 12. Coste de personal a jornada completa .....	31
Tabla 13. Total estimado de los costes hardware .....	32
Tabla 14. Total estimado real de los costes software .....	32
Tabla 15. Total estimado de los otros gastos .....	33
Tabla 16. Coste total estimado del proyecto .....	33
Tabla 17. Actores.....	41
Tabla 18. Requisitos de usuario .....	42
Tabla 19. Caso de uso "Iniciar sesión" .....	44
Tabla 20. Caso de uso "Visualizar perfil" .....	45
Tabla 21. Caso de uso "Cerrar sesión " .....	45
Tabla 22. Caso de uso "Modificar perfil" .....	46
Tabla 23. Caso de uso "Recuperar contraseña" .....	47
Tabla 24. Caso de uso "Crear evento" .....	48
Tabla 25. Caso de uso "Consultar nuevos eventos publicados" .....	49
Tabla 26. Caso de uso "Editar evento" .....	50
Tabla 27. Caso de uso "Eliminar evento" .....	51
Tabla 28. Caso de uso "Ver detalles de un evento" .....	52
Tabla 29. Caso de uso "Marcar cualquier asistencia" .....	53
Tabla 30. Caso de uso "Eliminar cualquier asistencia" .....	54
Tabla 31. Caso de uso "Dar de alta a un extra" .....	55
Tabla 32. Caso de uso "Dar de alta a nuevos miembros de la banda" .....	56
Tabla 33. Caso de uso "Consultar usuarios de la aplicación" .....	57
Tabla 34. Caso de uso "Dar de baja a cualquier usuario" .....	58
Tabla 35. Caso de uso "Ascender músico a jefe de cuerda" .....	59
Tabla 36. Caso de uso "Quitar rango de jefe de cuerda" .....	60
Tabla 37. Caso de uso "Consultar todos los eventos publicados" .....	61
Tabla 38. Caso de uso "Ver asistidos de un evento" .....	62
Tabla 39. Caso de uso "Generar PDF con los asistidos" .....	63
Tabla 40. Caso de uso "Marcar asistencia" .....	64
Tabla 41. Caso de uso "Eliminar asistencia" .....	65

Tabla 42. Caso de uso "Marcar asistencia de usuarios de su misma cuerda" .....	66
Tabla 43. Caso de uso "Eliminar asistencia de usuarios de su misma cuerda" .....	67
Tabla 44. Requisitos no funcionales de integridad .....	69
Tabla 45. Requisitos no funcionales de robustez .....	69
Tabla 46. Requisitos no funcionales de seguridad .....	70
Tabla 47. Requisitos no funcionales de usabilidad .....	70
Tabla 48. Requisitos no funcionales de disponibilidad .....	70
Tabla 49. Requisitos no funcionales de rendimiento .....	70
Tabla 50. Requisitos no funcionales de escalabilidad .....	71
Tabla 51. Requisitos no funcionales de portabilidad .....	71
Tabla 52. Diccionario de datos de la entidad Usuario .....	72
Tabla 53. Diccionario de datos de la entidad Evento .....	72
Tabla 54. Diseño de la interfaz de inicio de sesión .....	83
Tabla 55. Diseño de la interfaz de recuperar contraseña .....	84
Tabla 56. Diseño de la interfaz de pantalla principal .....	85
Tabla 57. Diseño de la interfaz de detalles de un evento .....	86
Tabla 58. Diseño de la interfaz de editar un evento .....	87
Tabla 59. Diseño de la interfaz de eliminar un evento .....	88
Tabla 60. Diseño de la interfaz de perfil .....	89
Tabla 61. Diseño de la interfaz de editar perfil .....	90
Tabla 62. Diseño de la interfaz de crear evento .....	91
Tabla 63. Diseño de la interfaz de registro de cuentas .....	92
Tabla 64. Diseño de la interfaz de listados .....	93
Tabla 65. Diseño de la interfaz de listado de usuarios .....	94
Tabla 66. Diseño de la interfaz de listado de eventos realizados .....	95
Tabla 67. Prueba de caja negra "Inicio de sesión incorrecto" .....	113
Tabla 68. Prueba de caja negra "Listado de eventos vacío" .....	114
Tabla 69. Prueba de caja negra "Creación de nuevos eventos sin conexión" .....	114
Tabla 70. Prueba de caja negra "Registrar miembro con campos vacíos" .....	114
Tabla 71. Prueba de caja negra "Modificación exitosa del perfil de usuario" .....	115
Tabla 72. Prueba de caja negra "Listado de usuarios registrados" .....	115