Diseño y desarrollo de una aplicación web para la recomendación y el descubrimiento de música

Design and development of a web-based application for music discovery and recommendation



Trabajo de Fin de Grado
Curso 2025–2026

Calificación: 9.0

Autor
Martina Águeda García

Directores
Alejandro Hernández Cerezo
Miguel Isabel Márquez

Doble Grado en Ingeniería Informática y Administración y Dirección de Empresas

Facultad de Informática

Universidad Complutense de Madrid

Diseño y desarrollo de una aplicación web para la recomendación y el descubrimiento de música

Design and development of a web-based application for music discovery and recommendation

Trabajo de Fin de Grado del Grado en Ingeniería Informática

Autor
Martina Águeda García

Directores
Alejandro Hernández Cerezo
Miguel Isabel Márquez

Convocatoria: Septiembre 2026

Doble Grado en Ingeniería Informática y Administración y Dirección de Empresas

Facultad de Informática

Universidad Complutense de Madrid

8 de septiembre de 2026

Dedicatoria

A mi abuelo José Luis,

por darnos la música.

Agradecimientos

Este Trabajo de Fin de Grado representa el cierre de mi etapa universitaria y no podía ser de otra forma que con la música como hilo conductor.

En primer lugar, quiero agradecer a mis padres y mis hermanos por ser el mejor club que existe, por enseñarme a ser persona y por acompañarme en todo.

A mi familia, mi mayor orgullo, gracias por estar siempre y por compartir la música conmigo.

A mis amigas, de mayor quiero ser como vosotras. Habéis sido lo mejor de estos años, gracias por todo.

A todos los que de una forma u otra han contribuido a que hoy esté donde estoy.

Y por supuesto, a Miguel y Alejandro, gracias por ser la mejor guía que podía haber tenido durante este año de locos.

Declaración de uso de Inteligencia Artificial

Durante el desarrollo del presente trabajo, se han usado herramientas de inteligencia artificial (IA) para asistir la elaboración de la memoria y el desarrollo de la aplicación.

Se usó Gemini para revisión de párrafos de la memoria, con el objetivo de detectar errores ortográficos o de redacción para su corrección.

Asimismo, se usó la herramienta Claude Code para agilizar la creación de código durante el desarrollo de prototipos y la aplicación, como recurso de consulta técnica y asistente de soporte para la resolución de errores que pudieran surgir.

Todo contenido generado, sugerido o modificado, fue revisado por el autor, el cual asume toda responsabilidad sobre la integridad y veracidad del software desarrollado y de la presente memoria. Asimismo, se declara expresamente que la IA no ha intervenido en la toma de decisiones relativas a la arquitectura del proyecto, a la metodología ni a la implementación de la lógica de negocio principal.

Resumen

Diseño y desarrollo de una aplicación web para la recomendación y el descubrimiento de música

Este trabajo nace de la insatisfacción con los sistemas de recomendación musical en la actualidad. La mayoría de usuarios recibe propuestas de la misma aplicación de *streaming* que utiliza para escuchar música y esto les provoca a menudo una sensación de estancamiento, así como de que se priorizan los intereses comerciales de la industria discográfica frente a sus gustos reales.

track[1] es una aplicación *full-stack* desarrollada con Next.js que permite a los usuarios registrar y valorar los álbumes que escuchan, compartir sus gustos con otros y recibir recomendaciones basadas en ellos a través de una interfaz cuidada y atractiva.

El sistema de recomendaciones que se encarga de generar las propuestas se apoya en una base de datos orientada a grafos de Neo4j sobre la cual se aplica la técnica de Generación Aumentada por Recuperación (RAG), que ofrece al LLM (*gpt-4o-mini* de OpenAI) un contexto acotado sobre el que decidir la propuesta. De esta forma, selecciona y argumenta la propuesta final entre un conjunto limitado de candidatos, mitigando el riesgo de alucinaciones.

El desarrollo ha consistido en una fase previa de investigación y dos de desarrollo iterativo, cada una culminada con evaluaciones con usuarios para validar el proyecto. Así, se ha obtenido como resultado una aplicación funcional cuya arquitectura y hallazgos se documentan en la presente memoria.

Palabras clave

Aplicación web, recomendación musical, RAG, base de datos orientada a grafos, LLM, Next.js, diseño UI/UX, evaluación con usuarios.

[1] Link a la app desplegada: track-tfg.netlify.app

Abstract

Design and development of a web-based application for music discovery and recommendation

This project grows out of the dissatisfaction with current music recommending systems. Most users receive suggestions from the same streaming application they use to listen to music, which often leads to a sense of standstil and the impression that the commercial interests of the record industry are prioritized over their actual preferences.

track[2] is a full-stack application developed with Next.js that enables users to log and rate the albums they listen to, share their tastes with others and receive recommendations based on those tastes through a refined and engaging interface.

The recommendation system responsible for generating these proposals relies on a Neo4j graph database, combined with a Retrieval-Augmented Generation (RAG) approach that provides the LLM (OpenAI's *gpt-4o-mini*) with a constrained context from which to select the proposal. In this way, it makes a choice and justifies it from a limited set of candidates, mitigating the risk of hallucinations.

The development process has consisted on an initial research phase followed by two iterative development cycles, each concluding with user evaluations to validate the project. Thus, the result is a functional application whose architecture and findings are documented in this dissertation.

Keywords

Web app, music recommendation, RAG, graph database, LLM, Next.js, UX/UI design, user evaluation.

[2]Link to the deployed app: track-tfg.netlify.app

Índice

1. Introducción **1**

1.1. Introducción 1

Motivación 1

Objetivos 1

Plan de trabajo 2

1.2. Introduction 3

Motivation 3

Goals 4

Work plan 4

2. Estado del Arte **7**

2.1. Estado del arte funcional 7

2.1.1. Recomendaciones de apps de reproducción 7

2.1.2. Otros sistemas de recomendación musical 10

2.2. Estado del arte tecnológico 11

2.2.1. Next.js 11

2.2.2. Neo4j 11

2.2.3. OpenAI 11

2.2.4. API de Discogs 12

2.2.5. Netlify 12

2.2.6. Tecnologías adicionales 12

3. Fase de investigación **15**

3.1. Justificación 15

3.2. Cuestionario 16

3.2.1. Objetivos del cuestionario 16

3.2.2. Elaboración y difusión 16
3.2.3. Análisis cualitativo 16
3.2.4. Análisis cuantitativo 20
3.3. Entrevistas 22
3.3.1. Preparación y realización de las entrevistas 22
3.3.2. Análisis y conclusiones de las entrevistas 24
3.4. Resultado de la fase de investigación 27

4. Diseño de la aplicación **29**
4.1. Diseño UI 29
4.2. Wireframes lo-fi 30
4.3. Wireframes hi-fi y prototipo 32

5. Arquitectura de la aplicación **37**
5.1. Arquitectura del sistema 37
5.2. Diseño del modelo de datos 38
5.2.1. Nodos y propiedades 39
5.2.2. Relaciones 39
5.3. Conexión con la API de Discogs 40

6. Sistema de recomendaciones **45**
6.1. RAG 45
6.1.1. GraphRAG 46
6.2. Arquitectura del sistema de recomendaciones 47
6.2.1. Ejemplo de ejecución 47

7. Desarrollo iterativo de la aplicación **51**
7.1. Primera iteración 51
7.1.1. Despliegue 53
7.1.2. Evaluación con usuarios 54
7.2. Segunda iteración 54
7.2.1. Mejoras 54
7.2.2. Sistema de recomendaciones 55
7.2.3. Evaluación con usuarios 55

8. Conclusiones y Trabajo Futuro **57**

8.1. Conclusiones y Trabajo Futuro 57

8.2. Conclusions and Future Work 59

Bibliografía **63**

A. Cuestionario **65**

A.1. Guión del cuestionario 65

B. Instrumentos de evaluación de usabilidad **69**

B.1. Fase 1: evaluación guiada 69

B.1.1. Preguntas previas 69

B.1.2. Tareas guiadas 69

B.1.3. Cuestionario System Usability Scale (SUS) 70

B.1.4. Preguntas de cierre 71

B.1.5. Resultados 71

B.2. Fase 2: evaluación de exploración libre 71

B.2.1. Resultados 71

Índice de figuras

1.1. Diagrama de Gantt del proceso de desarrollo 3

2.1. Mixes de Spotify 8
2.2. Playlists personalizadas de otros servicios de streaming 9
2.3. Musiio Tag de SoundCloud 10

3.1. Nube de palabras de las respuestas a la pregunta 10 del cuestionario . 18
3.2. Preguntas 11 y 12 del cuestionario 18
3.3. Gráfico circular de edades de los encuestados 21
3.4. Valoración de las recomendaciones según la frecuencia de estancamiento musical 22
3.5. Relación entre la apertura a la diversidad y el interés por recomendaciones basadas en el entorno 23
3.6. Interés en conectar con otros usuarios según la frecuencia con la que se comparte música 24

4.1. Elementos del *framework* de diseño 30
4.2. Wireframes lo-fi de la vista de Inicio 31
4.4. Wireframes lo-fi de la vista de Comunidad 31
4.3. Wireframe lo-fi de la vista de Explorar 31
4.5. Wireframes lo-fi de la vista de Perfil 32
4.6. Wireframe lo-fi de la vista de Usuario 32
4.7. Componentes reutilizables en Figma 33
4.8. Wireframes lo-fi de la vista de Perfil 34
4.10. Wireframe hi-fi de la vista de inicio de sesión 34
4.9. Wireframes hi-fi de las vistas de Perfil y Usuario 35

5.1. Arquitectura general de la aplicación 41

5.2. Esquema del modelo de datos en Neo4j 42
5.3. Diagrama de secuencia del flujo de llamada a la API de Discogs . . . 43

6.1. Diagrama conceptual del funcionamiento de un sistema con y sin RAG 46
6.2. Diagrama de secuencia del sistema de recomendaciones 49

7.1. Diagrama de secuencia del flujo de autenticación 52
7.2. Migración de Neo4j Desktop a Neo4j Aura 53

Capítulo 1

Introducción

1.1. Introducción

Motivación

Este trabajo nace de la necesidad recurrente de nuevas propuestas musicales y la insatisfacción de las que recibía hasta entonces. Es un problema que llevo reconociendo mucho tiempo tanto de primera mano como en mi entorno, y que transformé en la necesidad de dar con un sistema que entendiera al usuario lo mejor posible y le consiguiera recomendar aquello que sin saberlo buscaba.

A la hora de escuchar música, es habitual que la rotación de canciones, playlists y álbumes usual nos acabe resultando repetitiva y desinteresante. Intentar ampliar horizontes, bien dentro del mismo género o bien saliendo de lo habitual más drásticamente no suele ser fácil, pues el usuario necesita algún tipo de guía o punto de partida, que los sistemas de recomendación en la actualidad no siempre logran proporcionar. Por ello, salvo expertos en la materia, la población general se encuentra a menudo estancada en su gusto musical y con la pretensión de explorar nuevas posibilidades de escucha insaciada.

De cara a plantear mi Trabajo de Fin de Grado, quería dar con algo que no solo me fuera a permitir reflejar los conocimientos que he adquirido y los intereses que he desarrollado durante la carrera, sino hacerlo además en torno a alguna temática que me permitiera trabajar con una de mis mayores pasiones: la música.

Objetivos

Este trabajo tiene como objetivo mejorar la experiencia de recomendaciones musicales a través de una aplicación que combine los métodos que funcionan en la actualidad y corrija los que no para que el usuario reciba propuestas a medida de sus gustos y requerimientos.

En la actualidad, existen pocas aplicaciones dedicadas exclusivamente a perso-

nalizar recomendaciones musicales, sino que más bien se trata de funcionalidades dentro de aplicaciones de streaming de música. El principal problema que esto supone es que ajustar las propuestas a los requisitos de cada usuario es una prioridad secundaria para estos sistemas y por ello tienden a ser ineficaces. Muchos usuarios reportan también sentir que las recomendaciones coinciden con lo más comercial, pues es bien sabido que las grandes discográficas ponen grandes cantidades de dinero en promocionar sus productos en este tipo de espacios.

El propósito de este trabajo es democratizar las recomendaciones y refinar el algoritmo dedicado a ello para que la individualización de las propuestas sea conforme a lo que el usuario establezca y, por extensión, lo más satisfactorio posible. La intención es conseguirlo con una interfaz amigable y estéticamente atractiva, que ofrezca al usuario una experiencia agradable en conjunto.

Plan de trabajo

Si bien es cierto que con la elección del tema en mayo de 2025 tuve bastante espacio para imaginar cómo quería que fuera la aplicación y que, en realidad, las necesidades que quiero cubrir con ella se vienen manifestando desde hace tiempo, el trabajo como tal comenzó de manera efectiva con el curso académico.

Durante el primer cuatrimestre, me centré inicialmente en la fase de investigación de los sistemas de recomendación y de las fisuras con las que contaban, que podía intentar cubrir con mi trabajo. Después realicé cuestionarios y entrevistas para apoyarme en ellas y definir lo más estrictamente posible las necesidades que iban a suponer requisitos de la aplicación. También hice un análisis de la competencia con la que contaba mi aplicación para encontrar posibles problemas en ella a los que buscar dar solución con mi trabajo.

Paralelamente, me introduje en el mundo del diseño de UI, donde también dediqué bastante tiempo de exploración, un poco más a nivel de satisfacción e interés personal. Siempre me ha preocupado especialmente lo estético y la presentación al usuario del *software*, pues creo que hace mucho en la percepción del sistema. Por ello, investigué en revistas dedicadas al diseño UI, indagué en más detalle de lo que había tenido oportunidad hasta entonces en el ámbito del diseño UX y planteé las bases que quería para el posterior desarrollo de la aplicación.

Terminé el cuatrimestre con los bocetos de la aplicación aproximadamente a final de año, diseñando las vistas y planteando la estética que me gustaba para este propósito. Además, era importante para mí darle un nombre, una "marca" a mi aplicación para terminar de definir la idea e implicarme a fondo con el proyecto. Es entonces cuando nace ***track***.

Con el 2026, empezó el segundo cuatrimestre, en el que comenzaba el trabajo de implementación de la aplicación. Durante este tiempo llevé a cabo la implementación del *frontend* a partir de los diseños definidos en Figma, a la par que fui formalizando la arquitectura del sistema, según el desarrollo lo requería. Esta primera iteración se centró en la funcionalidad básica de la aplicación, dejando para una segunda

el sistema de recomendaciones y las posibles mejoras que pudieran surgir de una primera fase de evaluación con usuarios reales con la que concluyó la iteración.

Durante el verano incorporé los cambios derivados de los hallazgos de la primera fase de evaluación, además de una segunda fase de evaluación con usuarios para validarlos.

A medida que se avanzó en el proyecto se iba documentando en la memoria cada parte, por lo que el documento fue cobrando forma a lo largo de todo el curso académico.

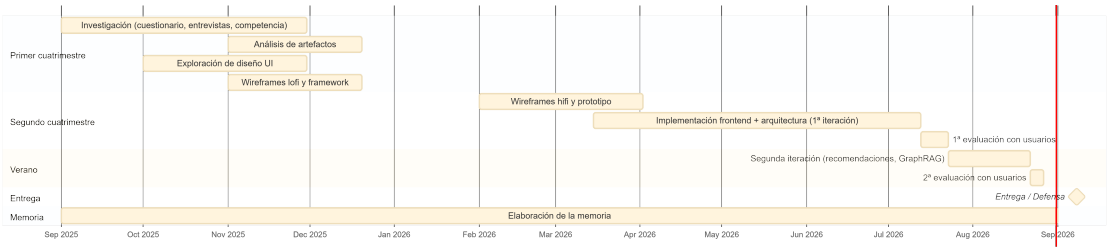


Figura 1.1: Diagrama de Gantt del proceso de desarrollo

1.2. Introduction

Motivation

This project stems from the recurring lack of new musical proposals and the dissatisfaction brought by the ones I was receiving. It is an issue I have been observing for a long time, both in myself and in my environment, and I was able to transform it into the need to develop a system that understood the user as best as possible and managed to recommend what they were unknowingly looking for.

In music listening, it is common that the usual rotation of songs, playlists and albums ends up feeling repetitive and uninteresting. Trying to broaden horizons, either inside the same genre or stepping out of the comfort zone in a more drastic manner is not an easy task, as the user needs some kind of guide or starting point that current recommending systems don't always manage to provide. Consequently, aside from experts in the matter, the general public is often found stuck in their musical taste and their appetite for new listening posibilities remains unsatisfied.

When it came to posing the topic for my Bachelor's Thesis, I wanted to come up with one that not only would allow me to reflect the knowledge and interests I have acquired throughout these years, but also do it on the subject of one of my biggest passions: music.

Goals

This project aims to improve the experience of receiving musical recommendations through an app that combines the methods that are currently working and rectify their shortcomings so as to provide the user with tailor-made proposals in accordance with their taste and requirements.

At present, there are few apps dedicated exclusively to personalizing musical recommendations, these functionalities are merely features within music streaming apps. The main problem this entails is that adjusting proposals to each user's requisites is a secondary priority to these systems, making them frequently ineffective. Many users also report feeling that recommendations often coincide with the most commercial content, since it is well known that big record labels place significant funds on promoting their products in these kind of spaces.

The goal of this project is to democratise recommendations and refine the algorithm dedicated to it so the individualisation of proposals is done appropriate to what the user establishes and, therefore, as satisfying as possible. The purpose is to achieve this alongside an aesthetically pleasing and friendly interface that offers the user an enjoyable experience altogether.

Work plan

Although it is true that when I had to choose the topic for my Bachelor's Thesis I had plenty of time to imagine how I wanted the app to turn out and that, in truth, the needs I want to meet with it have been manifesting for quite a while, the work itself began in earnest with the start of the academic year.

During the first term, I focused initially on the investigation phase of recommending systems and their gaps, which I would try to address with my work. Afterwards, I did questionnaires and interviews so I could lean on them and define as strictly as possible the shortcomings that were going to become requirements of the application. I also did a competitor analysis to find potential issues that I could find the solution for during the development phase.

In parallel, I delved into the world of UI design, to which I also dedicated a significant amount of exploration, a little more for personal fulfilment and interest. I have always cared deeply about the visual part of software, since I believe it does great a deal for the user's perception of the system. Thus, I consulted several magazines dedicated to UI design, I inquired in more detail than what I had been able to previously into UX design and laid the foundations of what I wanted for the later development stages of the app.

I ended the term by creating the wireframes for the app approximately at the end of 2025, designing the views and posing the aesthetic I liked for this purpose. Moreover, it was important for me to find a name, a brand for my app that would allow me to define the idea more precisely so that I could engage effectively with the project. It is at this point that the concept of ***track*** is born.

With 2026 came the implementation part of the development of the app. During this time I carried out the implementation of the frontend with the designs defined in Figma as my starting point, in parallel with the formalisation of the system's architechture, as the development would require. This first iteration focused on the basic functionality of the app, leaving the recommending system and the possible improvements derived from a first phase of evaluation with users for a second iteration.

During the summer I also incorporated the changes derived from the findings of the first phase of evaluation, as well as carrying out a second one to validate those changes.

As the project progressed, each section was documented in the dissertation, allowing the manuscript to take shape over the course of the academic year.

Capítulo 2

Estado del Arte

2.1. Estado del arte funcional

En la actualidad, la mayoría de usuarios obtienen recomendaciones musicales de las propias aplicaciones que utilizan para reproducir música. Las tres más usadas son Spotify, Apple Music y YouTube Music. Los métodos de recomendación varían para cada una, pero usuarios de las tres manifiestan deficiencias en las propuestas que ofrecen o explican no sentirse comprendidos por los sistemas de recomendación con los que cuentan (Baracskay et al., 2022).

A la hora de afrontar el trabajo, era especialmente importante conocer en la mejor medida posible cómo funcionan estos sistemas de recomendación. En este punto de investigación descubrí los conceptos de filtrado basado en contenido (*content-based filtering*) y filtrado colaborativo (*collaborative filtering*), que son los que usan estas plataformas para hacer propuestas a los usuarios. Mi formación en machine learning en el grado ha sido prácticamente nula, por lo que entender de qué manera se configuran las recomendaciones en los sistemas actuales supuso punto de inflexión en el enfoque que le daría a la investigación previa al diseño y, por extensión, al trabajo en su globalidad.

Con respecto a estas aplicaciones, me parece interesante destacar en este punto algunas de las funcionalidades con las que cuentan que me han llamado la atención, bien de manera positiva o negativa.

2.1.1. Recomendaciones de apps de reproducción

2.1.1.1. Spotify

Esta aplicación es la que utilizo prácticamente de manera ininterrumpida para reproducir música. Fundada en 2006, es actualmente una de las mayores proveedoras de servicios de reproducción de música y pódcasts. Ofrece lo que se conoce como servicio *freemium*, una estrategia de precios con la que los clientes pueden acceder a una versión básica de la app de forma gratuita, pero deben pagar para obtener la

versión completa.

Spotify ofrece reproducción de contenido con más de 100 millones de canciones y 7 millones de pódcasts para escoger y ha desarrollado un sistema de recomendaciones que combina filtrado basado en contenido y filtrado colaborativo. En la actualidad incorpora funciones de IA para ofrecer nuevas propuestas a sus usuarios.

Me informé de los criterios que tienen en cuenta para generar estas recomendaciones y me resultó especialmente interesante la selección editorial, en la que editores expertos deciden qué contenido incluir en las recomendaciones.

Por otro lado, en la parte personalizada de las recomendaciones, Spotify considera las preferencias de los usuarios, basándose en su comportamiento; información sobre estos, como personas a las que sigue, idiomas preferidos y otros datos de interés; y las tendencias actuales. Spotify tiene muy en cuenta lo que el público general está escuchando a la hora de generar propuestas, dando especial difusión a aquellas que cuentan con acuerdos promocionales. Además, hace uso de señales externas para comprender qué gusta. Así, por ejemplo, rastrea si un usuario llega a una canción desde TikTok o Instagram y le da un impulso mayor (Spotify, 2026).

Adicionalmente, me parece importante hacer un repaso por las funcionalidades más útiles e interesantes de Spotify para realizar propuestas. Una de ellas es el Blend: una función que genera listas compartidas entre 2 y hasta 10 usuarios, mezclando géneros, estados de ánimo y décadas favoritas de los participantes. No obstante, e irónicamente en el tiempo que llevo dedicando a este trabajo, he observado un deterioro en el desempeño de esta función: errores en la asignación de una canción a un usuario, recomendaciones que no se corresponden con ninguno, etc. Pese a ello, sigo considerando el Blend una muy buena opción para conectar con otros a través de la música.

Además, los *mixes* personalizados que propone Spotify abarcan un abanico muy amplio de conceptos: desde mixes por décadas, hasta mixes por artistas, pasando por mixes dedicados a distintos estados de ánimo y a contextos mucho más concretos como *Sunday Morning Easy Listening Mix* y *Getting Ready Shower Mix*.

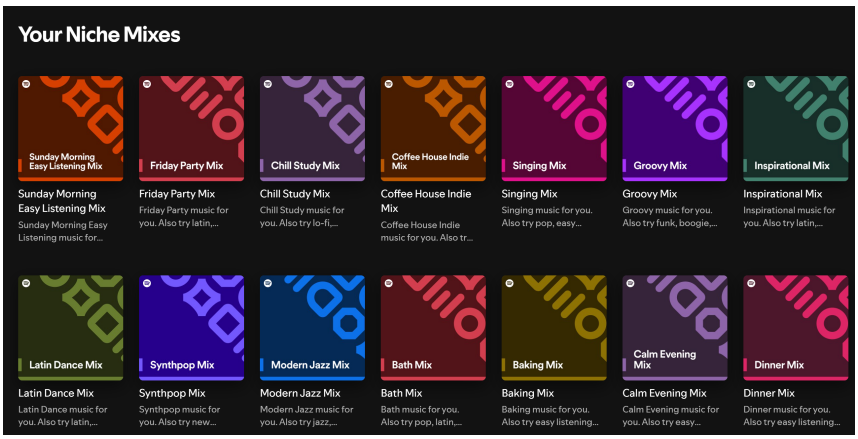
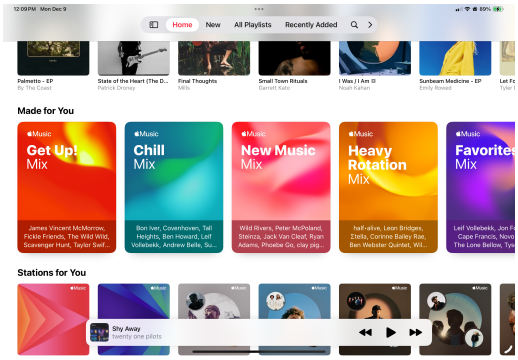
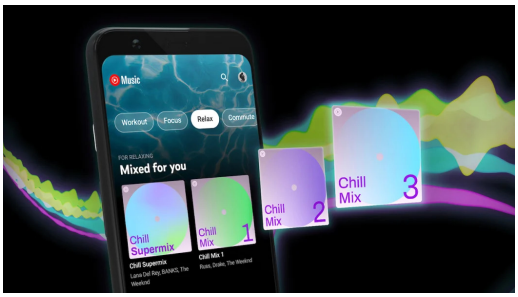


Figura 2.1: Mixes de Spotify

Otra de las funcionalidades que más me llama la atención de Spotify con respecto



(a) Apple Music



(b) Youtube Music

Figura 2.2: Playlists personalizadas de otros servicios de streaming

al descubrimiento de nuevas propuestas musicales es el relativamente reciente *DJ*, una función de inteligencia artificial que actúa, como su propio nombre indica, de DJ personalizado para los usuarios, intercalando música y comentarios contextuales, a modo de emisora de radio individualizada.

2.1.1.2. Apple Music

Apple Music es el servicio de *streaming* musical de Apple, lanzado en 2015 para sustituir a iTunes. A diferencia de Spotify y YouTube Music, solo funciona bajo suscripción de pago, no dispone de una versión gratuita con anuncios.

Su diferencia más notable frente a Spotify es el peso que tiene la selección humana en sus recomendaciones: Apple cuenta con cientos de editores que seleccionan manualmente el contenido de sus listas editoriales, y es el comportamiento de los usuarios sobre esas listas lo que después retroalimenta las recomendaciones algorítmicas y personalizadas. Se trata, por tanto, de un modelo combinado en el que la selección editorial no es independiente del algoritmo, sino su punto de partida.

2.1.1.3. YouTube Music

YouTube Music es un sistema de reproducción desarrollado por YouTube, pensado para sustituir a Google Play Music y Google Podcasts. También ofrece un servicio *freemium*, permitiendo ver vídeos musicales entre los que se insertan vídeos publicitarios en su versión gratuita y la reproducción sin publicidad, la reproducción de fondo de solo audio (lo cual reduce el consumo de datos para el usuario) y la descarga de canciones para la reproducción sin conexión en la versión de pago.

YouTube Music cuenta con un único mix personal con los artistas y canciones principales recientes de cada usuario, que se muestra en su canal y se actualiza diariamente. Además, se pueden crear experiencias de mix personalizadas, administrando funciones como la variedad y la familiaridad de los artistas (Google, 2025).

El rasgo que más lo diferencia de Spotify y Apple Music es que, al ser un servicio de Google, el algoritmo cruza la información de la app con el historial de búsqueda

y visualización del usuario en YouTube en general.

2.1.2. Otros sistemas de recomendación musical

Más allá de las principales plataformas de *streaming*, me informé sobre otros sistemas de recomendación y descubrimiento que entrarían en una clasificación de competencia más directa. Entre ellas, me llamaron especialmente la atención las APIs de SoundCloud Musiio Tag y Musiio Search.

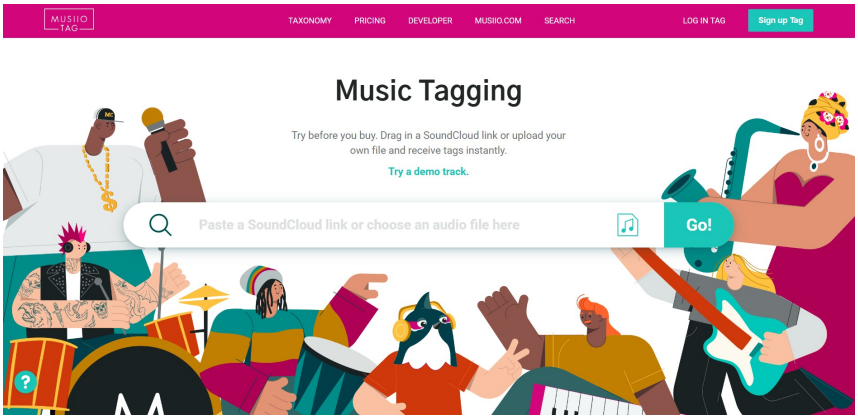


Figura 2.3: Musiio Tag de SoundCloud

Musiio combina inteligencia artificial con conocimiento musical experto para resolver problemas de catálogos musicales. Han conseguido automatizar el etiquetado y la búsqueda avanzada por similitud sonora y ofrecer informes para mejorar la gestión y el descubrimiento.

Por otro lado, me resultó muy interesante la base de datos de Discogs, que contiene información contrastada, actualizada y diversa sobre casi toda la discografía comercial existente. Además, disponen de una API que proporciona acceso a información en formato JSON de objetos de la base de datos.

La investigación relativa al estado del arte funcional se ha centrado especialmente en los algoritmos y sistemas de recomendación automatizada, si bien es cierto que adicionalmente se ha tenido en cuenta un amplio catálogo de sistemas de recomendación de selección humana como revistas, emisoras de radio y divulgadores, para estudiar qué hace que funcionen diferente a los sistemas automáticos y de qué manera estas ventajas se pueden incluir en el trabajo. En este contexto tomé como referencias especialmente interesantes la revista Pitchfork y la emisora de RTVE Radio 3. La principal ventaja que tienen este tipo de sistemas va desde la comprensión del contexto cultural de las propuestas hasta la capacidad de sorprender con recomendaciones basadas en el propio gusto personal del recomendador.

Esto último se hizo pensando en la posibilidad de incluir perfiles de referencia en las formas de recomendación, como pueden ser los propios artistas favoritos de los usuarios y otro tipo de expertos en la materia, que actuarían como autoridad cultural de la que los usuarios pudieran obtener propuestas.

2.2. Estado del arte tecnológico

Esta sección recoge las tecnologías elegidas para el desarrollo del proyecto, así como el proceso de evaluación de las alternativas y el motivo de la decisión final.

2.2.1. Next.js

Next.js es un *framework* basado en React para el desarrollo *full-stack* de aplicaciones web. Mientas que las aplicaciones desarrolladas con *frameworks* alternativos, como React puro o Vue, solo pueden renderizar contenido en el lado del cliente, Next.js permite también renderizar las vistas en el servidor, mejorando los tiempos de carga inicial mediante *bundles* (Next.js, 2026). De esta forma, se unifica el *frontend* con buena parte de la lógica de backend, con lo que se hace posible, por ejemplo, integrar el *driver* a Neo4j y las peticiones a la API de Discogs sin depender de una capa independiente.

Next.js cuenta con dos tipos de *routers* o enrutadores: el *Pages Router*, su sistema original, ya prácticamente deprecado, basado en la estructura de carpetas dentro de un directorio `pages/`, y el que yo he escogido para mi aplicación, el *App Router*, que incorpora *React Server Components* y ofrece un modelo más flexible al permitir combinar renderizado en servidor y en cliente dentro de una misma vista.

2.2.2. Neo4j

Neo4j es un modelo de base de datos de utiliza estructuras de grafos para almacenar los datos. Esto permite obtener una mayor rapidez en consultas, frente a alternativas relacionales como PostgreSQL o documentales como MongoDB. Utiliza el lenguaje de consulta Cypher, un lenguaje declarativo con patrones muy intuitivo visualmente.

En una aplicación de estas características, en las que las consultas principales son relaciones, un modelo relacional para la base de datos exigiría tablas intermedias para cada una de las relaciones. El modelo de grafo es una representación más directa de los datos que se usan, reduciendo la distancia entre el modelo mental de una red social como esta y el modelo de datos en sí mismo.

2.2.3. OpenAI

La API de OpenAI se utiliza para la parte generativa del sistema de recomendaciones. En concreto, se emplea el modelo *gpt-4o-mini*, más ligero que otros, dado que es una tarea relativamente sencilla y acotada, por lo que aporta la capacidad justa con un coste reducido.

En comparación con otras opciones, como la API de Anthropic (Claude) o Google Gemini, al tratarse de una tarea relativamente sencilla y acotada, no se justificaba

el coste más elevado.

2.2.4. API de Discogs

Discogs tiene una REST API que permite acceder a su base de datos para obtener información en formato JSON de *releases*, artistas y sellos. Aunque también puede gestionar colecciones de usuarios y *wishlists*, para este propósito se va a precisar únicamente la información relativa a los álbumes (título, artista, canciones, género, etc.).

2.2.5. Netlify

El despliegue de la aplicación se realizó a través de Netlify, una plataforma de desarrollo que ofrece servicios de *hosting* y despliegue y se conecta directamente con el repositorio de GitHub, automatizando el proceso de *build* con cada *commit*.

Además de usar el servicio de *hosting* para mi aplicación, utilicé Netlify para gestionar el guardado de las variables de entorno necesarias, como las credenciales de conexión a Neo4j y el token de la API de Discogs. El proceso de despliegue resultó sencillo, sin apenas configuración adicional a la que ofrece la plataforma por defecto.

2.2.6. Tecnologías adicionales

2.2.6.1. Figma

Figma es una de las herramientas de *software* de diseño de interfaces más utilizadas en la actualidad. Inicialmente la utilicé para la creación de los *wireframes* de la aplicación, tanto de baja como de alta fidelidad, pese a no ser la mejor opción para los bocetos de baja fidelidad, frente a opciones como Balsamiq, elegí usar Figma para unificar la etapa de diseño. Además, hice uso de sus herramientas de prototipado para plantear el flujo entre vistas.

A nivel de diseño, como se desarrolla en el capítulo 4, Figma me permitió definir el *framework* de diseño de la aplicación: colores, tipografía y los diferentes componentes reutilizables. De cara al desarrollo, esto me permitió abstraerme inicialmente de la funcionalidad para establecer cómo quería que se viera la aplicación, y posteriormente trasladarlo al código de la aplicación.

2.2.6.2. Bibliotecas de componentes

Se trata de colecciones de bloques de UI (interfaz de usuario), como botones, menús, barras de navegación, etc., que se pueden insertar directamente en el código del *frontend*. Cuando empecé a investigar sobre Radix UI, descubrí que había transicionado a Base UI, que es la biblioteca que he estado usando para algunos de mis componentes.

2.2.6.3. Entornos de desarrollo

Para el desarrollo de la aplicación se usó el IDE de JetBrains PyCharm y VS Code para el proyecto de la memoria en LaTeX, dado el límite de compilación de alternativas como Overleaf.

Para el control de versiones se utilizó Git, con GitHub como repositorio remoto. Además de la rama principal (`main`), donde se centralizó el desarrollo de la aplicación, se mantuvo otra separada para el desarrollo del sistema de recomendaciones, y otra (`memoria`) dedicada exclusivamente a la redacción de este documento, de forma que se mantuvieran aislados el historial de cambios del código y del de la memoria.

2.2.6.4. Jupyter Notebooks

Adicionalmente, cabe mencionar el uso de Jupyter Notebooks para realizar un análisis del cuestionario más exhaustivo que el que ofrece Google Forms. En él se importaron las bibliotecas pandas, numpy, matplotlib y seaborn para llevar a cabo análisis combinando estadística descriptiva con estudio de correlaciones entre variables relevantes del cuestionario de la fase de investigación.

Capítulo 3

Fase de investigación

3.1. Justificación

Basándome en lo aprendido por un lado en la asignatura de Diseño de Sistemas Interactivos y por otro en asignaturas de Marketing de ADE, consideré oportuno llevar a cabo una fase de investigación que me permitiera sentar las bases para lo que iban a ser requisitos funcionales de mi aplicación. Si bien el proyecto no sigue la metodología de Diseño Guiado por Objetivos vista en Diseño de Sistemas Interactivos, una fase de investigación previa al desarrollo permite conocer las motivaciones y comportamientos del usuario, imprescindibles a tener en cuenta en las fases posteriores.

La metodología del *Goal-Oriented Design* de Cooper (2014) justifica la fase de investigación desde un enfoque dedicado a entender al usuario y sus problemas para llegar a través de ello a la innovación. En esta ocasión, tuve la suerte de ser yo misma usuario objetivo de la aplicación, lo que me supuso una ventaja a la hora de conseguir ese entendimiento. No obstante, consideré necesario de igual forma realizar tanto un cuestionario como entrevistas en profundidad que me otorgaran puntos de vista que podía no estar teniendo en cuenta o simplemente otra perspectiva del problema.

En el ámbito del marketing, la investigación comercial se define como un proceso de identificación, valoración, selección y tratamiento de información para comprender el entorno, tomar decisiones o evaluar resultados. En este caso, pensando en la aplicación como el producto, lo que se busca es comprender el entorno para decidir cómo enfocar su desarrollo.

Los cuestionarios desde el punto de vista de la investigación comercial se comprenden como una técnica de recolección de datos de un grupo representativo para detectar tendencias de comportamiento y otros objetivos. En cuanto a las entrevistas, son parte de las técnicas de recogida de información cualitativas, con las que se pretende analizar el comportamiento y las opiniones de las personas de manera profunda y extensa.

La fase de investigación me permitió terminar de delimitar el alcance de la aplicación, así como definir con más detalle las funcionalidades que iba a tener, apoyán-

dome en lo que observé que era necesario según los resultados de mi análisis.

3.2. Cuestionario

3.2.1. Objetivos del cuestionario

La intención principal de llevar a cabo esta técnica era conocer los hábitos de escucha de los participantes. Por ello, cuestiones como cuánto tiempo dedicaban diariamente a escuchar música o a través de que medio lo hacían me parecieron pertinentes para obtener información que poder analizar posteriormente.

A un nivel más específico, con la difusión del formulario buscaba definir con mayor detalle las diferencias de perfil de escucha que puede haber, que ya existían en mi imaginario, pero necesitaba datos reales en los que basarme para definirlos más concretamente. Por otro lado, quería analizar la posible existencia de situaciones específicas que provoquen una mayor sensación de "estancamiento musical".

Además, entender cómo escucha música la gente (medio, aplicación, género, etc.), en qué situaciones o contextos lo hace o con qué frecuencia, su experiencia actual con las recomendaciones y hasta qué punto la música es algo que usen para socializar de algún modo, también me parecía relevante de cara al definir lo que iba a ser mi aplicación.

3.2.2. Elaboración y difusión

La elaboración del artefacto en sí se realizó en Google Forms[1] y la difusión una vez terminado fue a través de WhatsApp, con un mensaje transmitido por varios grupos en el que pedía a mis contactos que respondieran y, en la medida de lo posible, lo reenviaran como pudieran. El resultado fue una muestra de tamaño inesperadamente grande (343 respuestas).

Cabe destacar que, en retrospectiva, dado que la elaboración no se hizo pensando en el análisis posterior, o quizás por no esperar un volumen de respuestas tan elevado, el proceso de análisis resultó más complejo de lo que podía haber sido. Sin embargo, por el contrario, gracias a este número tan alto de participaciones, se pudieron hacer estadísticas y estudios más profundos y representativos de lo que se esperaba.

3.2.3. Análisis cualitativo

Al observar las respuestas, y pensando en un análisis cuantitativo, contaba con la dificultad de haber formulado demasiadas preguntas de manera abierta. No obstante, este tipo de preguntas permiten que los participantes se salgan del guión y mencionen

[1] Ver Apéndice A

aspectos que pueden no estar siendo considerados en la pregunta ni en las respuestas cerradas.

Como se cuenta con una base de respuestas tan amplia, considero oportuno destacar algunas de las respuestas abiertas que más me han llamado la atención, bien por que simplemente me hayan abierto la mente a perspectivas diversas, o bien porque han contribuido a la definición del funcionamiento de la aplicación en sí misma.

La segunda pregunta del formulario (*¿Qué géneros de música son tus preferidos?*), contaba con opciones de selección múltiple con los géneros más comunes y una opción adicional en la que se podían añadir otros. Por lo general, los encuestados seleccionaban alguno o varios de los géneros que aparecían como opción y en algunos casos añadieron otros. Entre estos destacan: indie, R&B, folk/country, bossa nova y música católica/religiosa/de alabanza.

Tras observar las respuestas, he llegado a la conclusión de que hacer una pregunta dedicada al dispositivo empleado para la escucha y otra sobre el medio puede dar lugar a equívoco, haciendo que se mezclen las respuestas. De nuevo, ambas se hicieron con la opción "Otro: " que permitía que los participantes añadieran otras opciones. Los resultados de estas no son notables.

La pregunta número 5 se enfocaba en aquello a lo que los encuestados otorgaban mayor importancia a la hora de elegir una plataforma (*¿Qué es lo más importante para ti al elegir esta plataforma?*). Entre las respuestas que añadieron se observa que hay bastantes que encajan con las que ya estaban formuladas, y en el caso de las que no, se pueden resumir en una opción que no se consideró al formular la pregunta que tiene que ver con la comodidad o la costumbre y el esfuerzo que supondría cambiar a otra plataforma.

Las respuestas abiertas de las preguntas 6 y 8 aportan información adicional que no termina de ser relevante para la definición de los requisitos funcionales de la aplicación. No obstante, sí son interesantes a nivel cuantitativo por lo que se estudiarán en ese sentido.

Tras la pregunta sobre si la música es una parte importante de su vida, los participantes encontraban la posibilidad de justificar su respuesta. Me parece especialmente importante para el propósito de mi trabajo analizar aquellas justificaciones de quienes consideran que sí.

Por lo general, los participantes alegaban que la música actúa en sus vidas como una herramienta de gestión emocional: mejora el estado de ánimo, les relaja y ofrece consuelo y refugio en momentos peores y los conecta a sus emociones. Además, les acompaña cuando lo necesitan, está presente en sus actividades más rutinarias y en momentos más especiales, por lo que en ocasiones despierta recuerdos y nos transporta a ellos.

Se puede extraer también que la mayoría se apoyan en la música para amenizar tareas del día a día, concentrarse o desconectar y evadirse por un momento de la realidad. Por otro lado, muchos describen cómo la música les permite conectar con su entorno, bien compartiendo propuestas, en eventos sociales como pueden ser los



Figura 3.1: Nube de palabras de las respuestas a la pregunta 10 del cuestionario

conciertos e incluso cantando o tocando con gente. Esta última premisa es una de las razones por las que quiero llevar a cabo este proyecto, porque creo firmemente que la música es una de las mejores formas de conectar con los demás y sentirse acompañado.

Adicionalmente, dediqué una sección del cuestionario a saber sobre las recomendaciones que recibían los participantes. Para ello, primero pregunté por la frecuencia con la que se sentían estancados en su gusto musical y de qué manera resolvían esta situación en la actualidad. Las respuestas a esta última pregunta iban desde quienes no intentaban resolverlo por no encontrarle importancia suficiente o no considerarlo negativo, hasta quienes acudían a su entorno, su aplicación de reproducción de confianza o emisoras de radio para recibir propuestas.

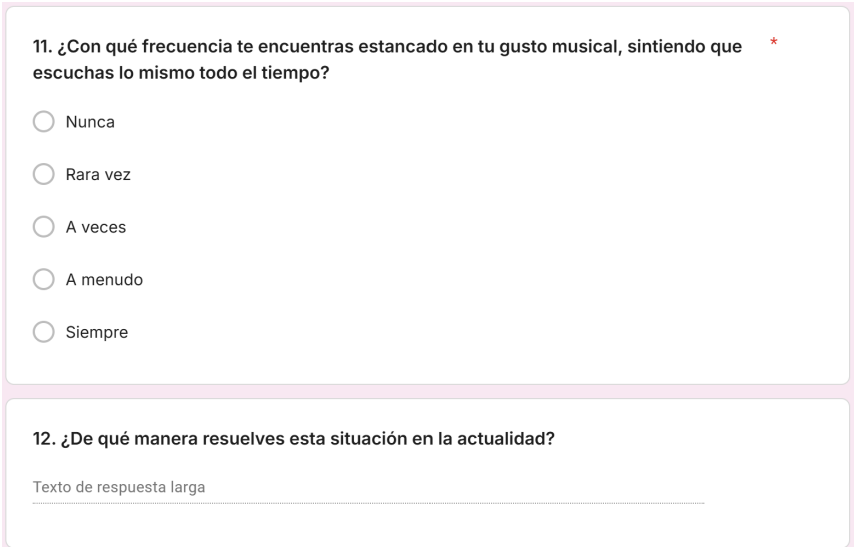


Figura 3.2: Preguntas 11 y 12 del cuestionario

La siguiente pregunta abierta que figuraba en el cuestionario estaba dirigida a conocer cómo consideraban los encuestados que su fuente de recomendaciones podía mejorar esta experiencia (*¿De qué manera podría tu fuente de recomendaciones*

mejorar esta experiencia?). Las gran mayoría no supo dar una respuesta, pero entre los que sí lo hicieron, las más destacables son, reunidas por similitud:

- "No repitiendo las mismas canciones constantemente.", "Ampliando las recomendaciones, no solo lo que cree que me va a gustar."
- "Creo que el algoritmo de spotify se ha vuelto muy mainstream y capitalizado. Antes hacía mejores recomendaciones y de artistas que de verdad no conocía. Me gustaría que me recomendara artistas por descubrir", "No buscando beneficio comercial"
- "Preguntándome", "Adivinar mis emociones para saber que me apetece.", "Si recibiese mi "feedback"de la recomendación o confirmando si vuelvo a escuchar esa recomendación.", "Quizás que me de recomendaciones basando en el momento del dia o mi estado de ánimo.", "Siento que a veces no tiene en cuenta música que a lo mejor escucho menos frecuentemente pero que aún así me gusta, es música más concreta para algunos momentos y que nunca me recomienda."
- "En el caso de personas físicas, no se puede mejorar. En el caso de algoritmos de plataformas, le falla el caracter humano y la cercanía personal."
- "Por ejemplo, que desde la propia plataforma haya una opción de conocer música nueva ya sea o bien atendiendo a las canciones/autores/estilos que ya escuchas (y así buscar algo similar pero nuevo) o la opción de encontrar algo totalmente distinto."
- "Ampliando mi fuente, es decir, compartiendo mis inquietudes con más personas.", "Vínculandose con las recomendaciones que recibo de mis amigos por la misma app."

Todas estas inquietudes, me sirven de motivación y fundamento para determinar las funcionalidades que quiero que mi aplicación tenga para cubrir este tipo de necesidades.

En el primer punto, se observa la sensación de estancamiento en las mismas recomendaciones. El siguiente apartado refleja la sensación de mercantilización de las propuestas, limitándose cada vez más aquello que se desea que venda más. El tercer conjunto de respuestas apela a una posible funcionalidad en la que, a través de un breve cuestionario, el usuario pueda transmitirle a la aplicación su estado anímico y/o el tipo de música que le apetece escuchar y en base a ello, esta genere una propuesta a medida.

La siguiente respuesta conecta con la posibilidad de incluir perfiles de usuarios públicos "expertos" que recomienden álbumes a sus seguidores. El quinto punto de la lista hace referencia a la idea de que en ocasiones buscamos descubrir opciones parecidas a lo que ya conocemos y sin embargo en otras se prefiere salir de esa *zona de confort* y explorar otros géneros y/o artistas.

Por último, destaca la necesidad de socializar en torno a la música, bien para "ampliar fuentes", o bien para conectar con nuestro entorno a través de recomendaciones.

Finalmente, y aunque pueda resultar redundante, hice otra pregunta para entender cómo a los encuestados les gustaría que fueran las recomendaciones que reciben. Algo que entre las respuestas se repitió mucho fue que fueran basadas en sus gustos y no en los de otros. Otros declaraban que buscaban abrir horizontes con recomendaciones que se salieran de lo que escuchan en la actualidad. También hubo bastantes respuestas que querían que estuvieran ligadas a los gustos de su entorno. Por último, cabe destacar aquellas respuestas de quienes buscan alternar entre lo que ya escuchan y lo que no.

En definitiva, incluir preguntas abiertas en mi cuestionario, aunque con un volumen de respuestas de tan considerable tamaño haya sido complicado de analizar, me ha permitido saber con mayor detalle sobre los hábitos de escucha de una muestra muy heterogénea. Con ellas, no solo encuentro apoyo para basar el diseño de mi aplicación, sino que además me sirven de refuerzo de la motivación de mi trabajo, porque me recuerdan por qué es necesaria una aplicación como esta, que conecte personas a través de la música y haga lo más difícil posible el sentirse estancado en la rotación habitual de álbumes y playlists.

3.2.4. Análisis cuantitativo

Para esta sección, y, de nuevo, dado el inesperado nivel de respuestas conseguido, consideré insuficiente el análisis automático que ofrece Google Forms y opté por la elaboración de un Jupyter Notebook en el que importé las librerías *pandas*, *numpy*, *matplotlib*, *seaborn* y *scipy* para llevar a cabo un análisis cualitativo más detallado de las respuestas al cuestionario.

Para ello, primero se han de limpiar y codificar las respuestas. Esta parte me llevó más tiempo del deseado, una vez más dado que al elaborar el cuestionario no tuve en cuenta este posible análisis posterior. Esto fue debido, principalmente, al hecho de que este tipo de análisis para muestras pequeñas no tiene mucho sentido, ya que no son suficientemente representativas.

Atendiendo al estándar de los cuestionarios de investigación comercial, la sección final del formulario estaba dedicada a conocer la muestra de forma demográfica. Conocer el perfil demográfico de la muestra no tiene demasiada utilidad para este propósito en la práctica, pero sí es interesante conocer cómo está configurada. También en este aspecto el tamaño tan grande de respuestas permite que sea mucho más heterogénea, lo que aporta representatividad a la muestra y credibilidad a los resultados.

Cabe destacar de la distribución por edades que el grupo más numeroso es el de 46-55 años, lo que no es usual en encuestas online de este tipo. En cuanto a otros demográficos como el género, destaca una participación femenina de casi el 70

En cuanto al resto de datos cuantitativos, me parece valioso el volumen de res-

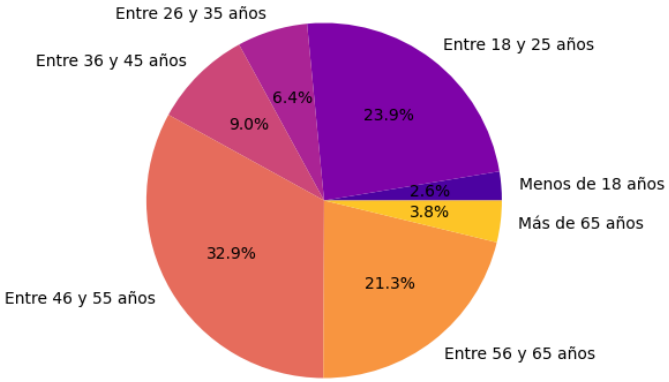


Figura 3.3: Gráfico circular de edades de los encuestados

puestas recibido para llevar a cabo un análisis exhaustivo y realizar algún tipo de análisis de datos que permita analizar posibles correlaciones entre variables que pueda utilizar como fundamento para los requisitos de la aplicación.

Además de algunos intentos fallidos de encontrar correlaciones y contrastar hipótesis, traté de encontrar alguna relación entre la sensación de estancamiento con el uso de las tres aplicaciones principales, pero las muestras de usuarios de Apple Music y YouTube Music no eran representativas.

A continuación, estudié si la frecuencia con la que los encuestados se sentían estancados en su gusto musical se relacionaba de alguna forma con la valoración que le daban a las recomendaciones que recibían. El resultado tampoco fue estadísticamente significativo ($\rho = -0{,}049$, $p = 0{,}40$, $n = 299$): la valoración media es muy similar para los que nunca se sienten estancados y quienes lo están constantemente. Sin embargo, esto sugiere que el estancamiento musical no depende de cómo se valoren las recomendaciones recibidas, sino de otros factores, reforzando la necesidad de una forma de fomentar la exploración.

Lo siguiente que quise comprobar es si el interés por recibir propuestas que se alejen de los gustos habituales y por recibir recomendaciones en base a los gustos del entorno son actitudes opuestas o compatibles. La correlación resultó positiva y significativa ($\rho = 0{,}241$, $p < 0{,}001$, $n = 299$).

Al dividir la muestra en cuadrantes, un 12,7 % se interesa por la diversidad y por lo social de manera simultámea, frente a un 17,4 % que puntúa alto en una dimensión y bajo en la otra; el 53,2 % restante se otorga valores intermedios de interés a ambas. Estos resultados descartan que exista una relación entre buscar diversidad y valorar lo social a la hora de recibir recomendaciones, lo que implica que no es necesario hacer una separación entre un enfoque de descubrimiento basado en la diversidad y uno basado en el entorno.

Finalmente, analicé la relación entre la frecuencia con la que los participantes comparten música con su entorno y su interés en conectar con otras personas a través de la música. Esta fue la correlación más fuerte de las tres ($\rho = 0{,}432$, $p < 0{,}001$,

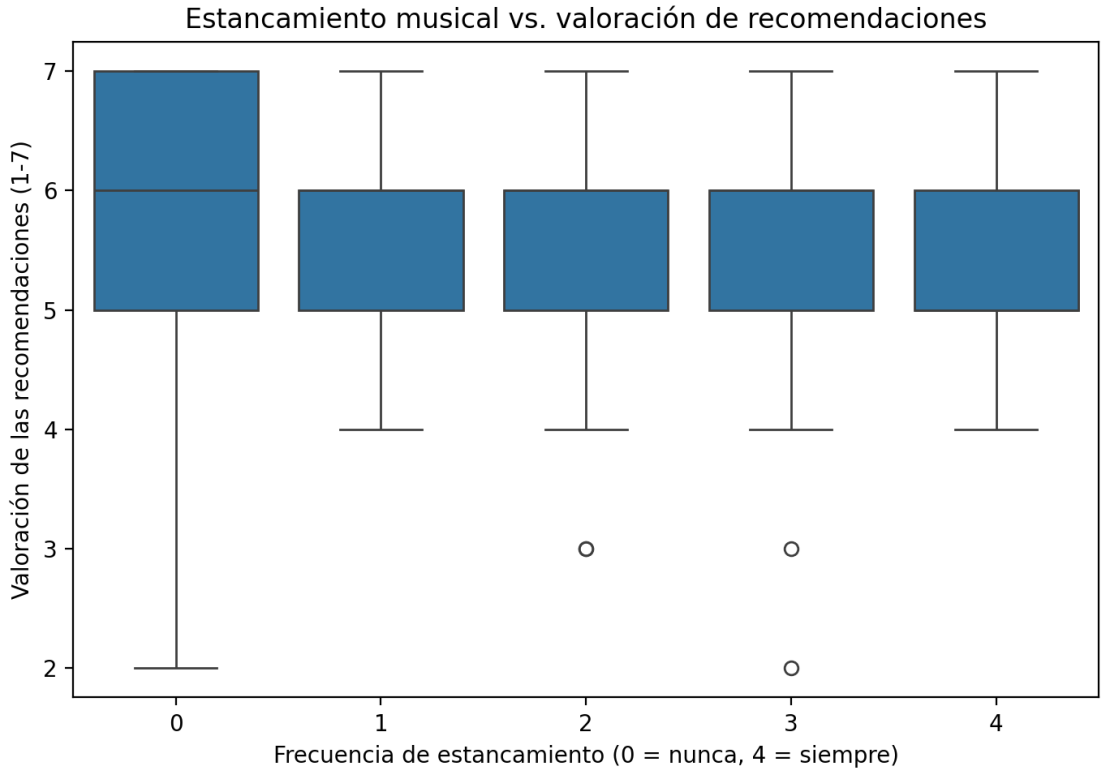


Figura 3.4: Valoración de las recomendaciones según la frecuencia de estancamiento musical

$n = 343$). De los tres análisis, es el que apoya cuantitativamente en mayor medida a la inclusión de funcionalidades sociales en la aplicación.

En definitiva, estos análisis confirman parte de lo observado en las respuestas abiertas del cuestionario y en las entrevistas, además de aportar una base cuantitativa que respalda dos de las decisiones funcionales más relevantes del trabajo: mantener unido el descubrimiento diverso y el descubrimiento social, y dar relevancia a la dimensión comunitaria de la aplicación.

3.3. Entrevistas

3.3.1. Preparación y realización de las entrevistas

Para la realización de las entrevistas y teniendo en cuenta las limitaciones de carácter temporal, decidí que era interesante hacérselas a dos perfiles diferentes que me proporcionaran relatos diversos de cara a plantear los requisitos. El objetivo que perseguía con las entrevistas era conocer la experiencia de escucha y descubrimiento de algunos individuos de manera más exhaustiva y detallada. Las entrevistas fueron grabadas (una audio y la otra vídeo) para su posterior análisis.

Para la elaboración del guión consideré oportuno organizarlo en varias secciones:

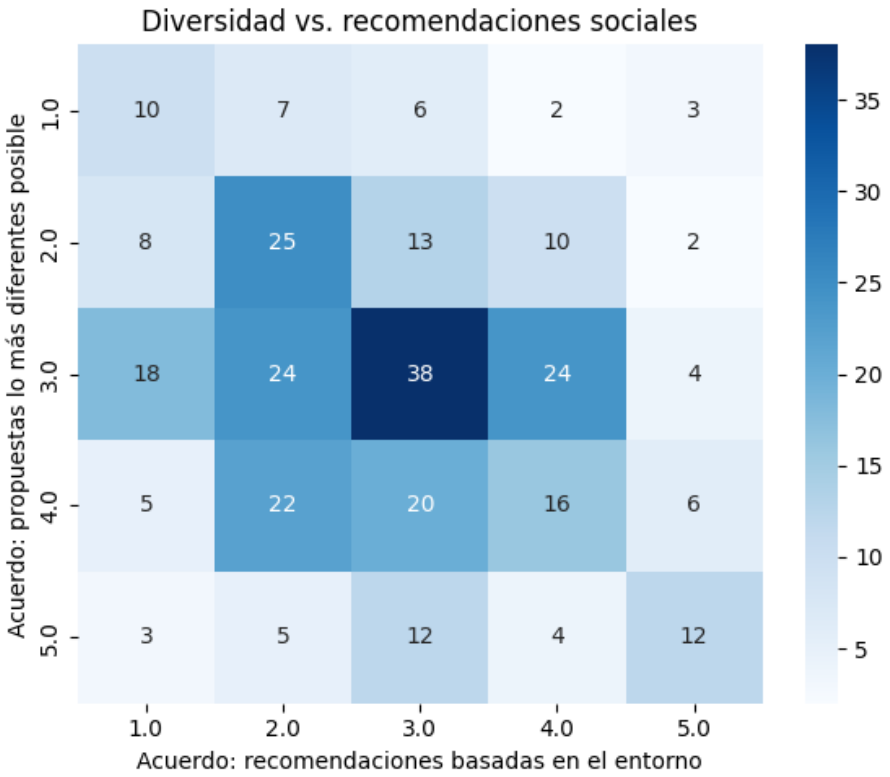


Figura 3.5: Relación entre la apertura a la diversidad y el interés por recomendaciones basadas en el entorno

1. Contexto y hábitos generales de escucha: dedicada a entender en qué situaciones la música acompaña a las personas encuestadas.

2. Descubrimiento de música nueva: dedicada a conocer cómo descubren nuevas propuestas de escucha y si las fuentes actuales les satisfacen o les producen frustración.

3. Registro musical: para saber cómo guardan la música que han escuchado para poder recuperarla en un futuro.

4. Estadísticas: de cara a una posible sección de estadísticas en la aplicación.

5. Otras necesidades: sección de cierre más abierta para explorar necesidades y otros temas que no hayan sido contemplados y la persona encuestada desee tratar.

Lo utilicé como base sobre la que estructurar la entrevista, pero teniendo en mente la posibilidad de salir de él para explorar posibles temas que fueran surgiendo.

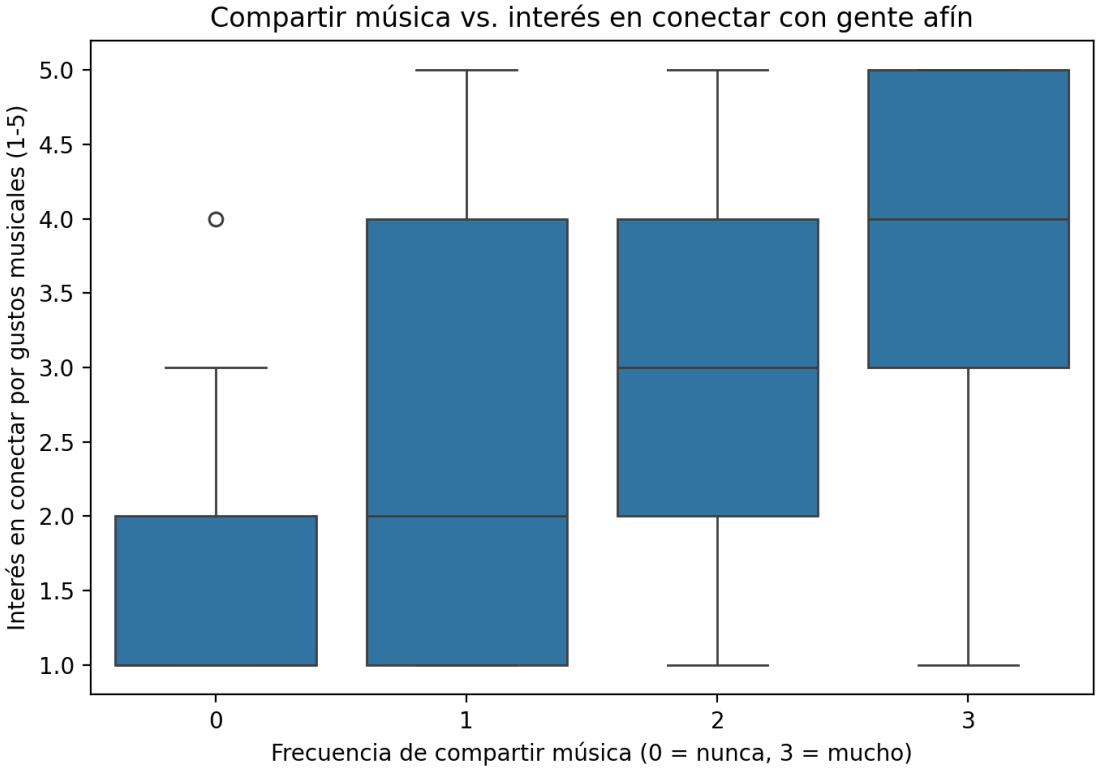


Figura 3.6: Interés en conectar con otros usuarios según la frecuencia con la que se comparte música

3.3.2. Análisis y conclusiones de las entrevistas

3.3.2.1. Entrevista A

Perfil, contexto y hábitos generales

La persona entrevistada es una mujer de 21 años, estudiante y usuaria de Apple Music. Describe la música como un elemento imprescindible en su día a día, y describe su papel como la "banda sonora de mi vida". Escucha música de manera constante, principalmente a través del móvil o en el coche.

Hábitos de descubrimiento

Su proceso de descubrimiento de nuevas propuestas es puramente social. Valora especialmente la recomendación de su círculo cercano, sobre todo de aquellas persona que considera que tienen un criterio específico según el género o los artistas que comparte con ellos. También aprecia las recomendaciones de personas de referencia en redes sociales, como pueden ser otros artistas. Por otro lado, utiliza herramientas como Shazam para guardar música que le gusta que escucha en espacios públicos.

Cuando obtiene propuestas de artistas nuevos por cualquiera de estos medios empieza escuchando su contenido más *mainstream* para decidir si quiere profundizar.

En este punto, no se mostró especialmente crítica sobre el algoritmo de recomen-

daciones, pero sí explicó que cuando descubre a un artista concreto de un género, tiene la sensación de que la aplicación tiende a llenarle de más música de ese mismo género sin entender por qué le ha gustado ese artista en particular. A veces acierta, y a veces le sale algo que "no escucharía nunca". Es un matiz que puede resultar importante a la hora de plantear la recomendación.

Hábitos de registro

Manifiesta que le gustaría poder llevar un registro de la procedencia de la música en su biblioteca, que cuando descubre una canción utiliza la función de "Favoritos" para evitar que caiga en el olvido. Añade que en ocasiones también recurre a la app de Notas de su móvil para apuntar canciones cuando no puede reproducirlas en el momento, pero se encuentra con el mismo problema de falta de contexto.

Sobre compartir con su entorno, cuenta que suele incorporar música en lo que comparte en redes sociales, porque es lo que la acompaña en todos los momentos. Además cuenta con amistades con las que suele intercambiar recomendaciones muy a menudo, en ocasiones son propuestas que puede que no le gusten, pero que identifica con el estilo de cierta persona.

Estadísticas

En esta sección de la entrevista, menciona la reciente incorporación de la sección de estadísticas mensuales de Apple Music, como una funcionalidad interesante para conocer más sobre sus hábitos de escucha. Echa en falta poder saber cuántas veces ha escuchado una canción, como se muestra en el Wrapped de Spotify, pero para todas.

Por otro lado, no muestra especial necesidad en diferenciar por géneros en estadísticas, porque su gusto es ecléctico y no hace esa separación, pero sí que le podría interesar conocer qué generos escucha más a menudo de esta forma.

Conclusiones

Como sección final algo más abierta, cuando se le preguntó qué echaba de menos en el proceso de escuchar, descubrir y recomendar música, habló de que le gustaría poder usar una red social únicamente dedicada a compartir música, en la que incluso profesionales como artistas, productores, etc., compartieran propuestas para otros.

En conclusión, esta persona depende en gran medida de las recomendaciones de su entorno en su proceso de descubrimiento musical y como aspectos a mejorar en esta experiencia, destaca la necesidad de mantener algún sistema que le permita tener contexto sobre la procedencia de dichas recomendaciones.

3.3.2.2. Entrevista B

Perfil, contexto y hábitos generales

La persona entrevistada en esta ocasión es una mujer de 54 años, usuaria asidua de Spotify, además de oyente de emisoras de radio como Radio 3 y Radio Clásica. La música ocupa un espacio central en su vida, acompañándola en todo momento:

en los desplazamientos en coche, en el trabajo, en casa, etc.

Hábitos de descubrimiento

Su proceso de descubrimiento se caracteriza por combinar métodos que se pueden considerar más "tradicionales" con otros más recientes como el propio algoritmo de Spotify. También en redes sociales obtiene recomendaciones de divulgadores o expertos y emplea Shazam para identificar música que escucha, ya que además le permite tenerlo registrado y no depender de su memoria.

De Spotify valora especialmente la lista de *Descubrimiento Semanal*, pues es donde más descubrimientos hace.

Hábitos de registro

En este ámbito explica que la gran cantidad de listas de reproducción que mantiene, con un número de canciones por lista muy elevado, le genera una sensación de desorganización que le abruma. Le resulta muy complicado encontrar canciones en su biblioteca y, cuando hace algún descubrimiento, bien a través de RRSS o bien con Shazam, lo guarda mediante una captura de pantalla a la que después debe acudir para recordarlo.

Destaca también la idea que expone de que la percepción de una propuesta varía por momentos, y puede que un día haga un descubrimiento y cuando pasado el tiempo lo vuelva a escuchar no le guste nada. Además me resulta especialmente relevante el hecho de que, para ella, un registro por álbumes es mucho menos descriptivo que hacerlo por canciones individuales.

Le gusta compartir lo que escucha con su entorno a través de recomendaciones que personaliza para cada persona porque piensa que le puede gustar y le gusta recibir recomendaciones de la misma forma.

Estadísticas

No muestra especial interés en contar con estadísticas que describan sus patrones y hábitos.

Conclusiones

La parte final de esta entrevista le dio sentido a realizar entrevistas en profundidad además del cuestionario, ya que permitió salir del guión y conocer inquietudes que no tenía contempladas y que ayudaron a plantear nuevas posibles funcionalidades.

Es interesante la idea que tiene esta persona de la promoción de ciertas propuestas, tanto en apps de reproducción como en la propia radio, que le hace sentir que lo que recibe es porque se está pagando para ser recomendado y plantearse la posibilidad de hacer una investigación propia para conseguir hacer descubrimientos al margen de este sistema comercial.

Por otro lado, expresa que su aplicación mágica soñada en este ámbito de descubrimiento y escucha sería una que fuera capaz de entender en cada momento lo que querría escuchar en función de la situación en que se encuentra. De esta idea surge la posible funcionalidad "Generar recomendación", algo parecido a un cuestionario

en el que los usuarios pudieran describir brevemente su situación, de manera que la aplicación generara una recomendación adaptada a ella.

3.4. Resultado de la fase de investigación

En conjunto, las técnicas de investigación empleadas me han permitido bajar a tierra las ideas que tenía planteadas pero que necesitaban definición, al combinar lo que manifestaron los participantes del cuestionario con lo que se profundizó en las dos entrevistas.

Los hallazgos confirman que los usuarios no solo buscan salir del estancamiento musical mediante algoritmos más transparentes y personalizados, sino que valoran el factor humano y la trazabilidad de las propuestas que reciben.

No obstante, no todos los usuarios comparten las mismas prioridades. Por un lado, se identifica un perfil de escucha más dependiente del factor social, cuyo proceso de descubrimiento depende casi por completo de su entorno cercano y de figuras de referencia en redes sociales, y que además comparte música activamente como forma de expresión personal. Este perfil se beneficiaría especialmente de contar con una red social integrada en la aplicación.

En segundo lugar, un perfil más autónomo y crítico con los sistemas de recomendación comerciales, que combina fuentes más tradicionales, como la radio, con el algoritmo de la propia plataforma, pero que desconfía de hasta qué punto las propuestas que recibe responden a intereses comerciales antes que a sus gustos reales. Este perfil es el que más necesita de un sistema de registro y trazabilidad sólido que le permita recuperar el contexto de sus descubrimientos.

Finalmente, un tercer perfil, más transversal, de quienes se sienten estancados en su gusto musical con independencia de cómo valoren las recomendaciones que reciben. Para ellos, mejorar el algoritmo de recomendación no basta, ya que necesitan además una vía activa para expresar en qué momento se encuentran, lo que conecta directamente con la funcionalidad de "Generar recomendación".

A partir de estas conclusiones, se cierra la fase de investigación con la traducción de las necesidades detectadas a los requisitos funcionales de la aplicación:

- **Sistema de registro y valoración de álbumes**: la funcionalidad central de ***track***. Permite catalogar, puntuar y llevar un registro de los álbumes que se han escuchado.
- **Red social para conectar a través de la música**: interactuar con otros usuarios y recibir recomendaciones en base a lo que escuchan otros.
- **Sistema de recomendación mejorado**: que ofrezca recomendaciones personalizadas basadas en lo valorado por el usuario y lo valorado por los amigos de este.

De este modo, la investigación me ha permitido justificar la existencia de la aplicación más allá de una motivación personal y me ha ayudado a definir los requisitos funcionales de cara a poner en marcha el diseño de la arquitectura del proyecto y su desarrollo.

Capítulo 4

Diseño de la aplicación

4.1. Diseño UI

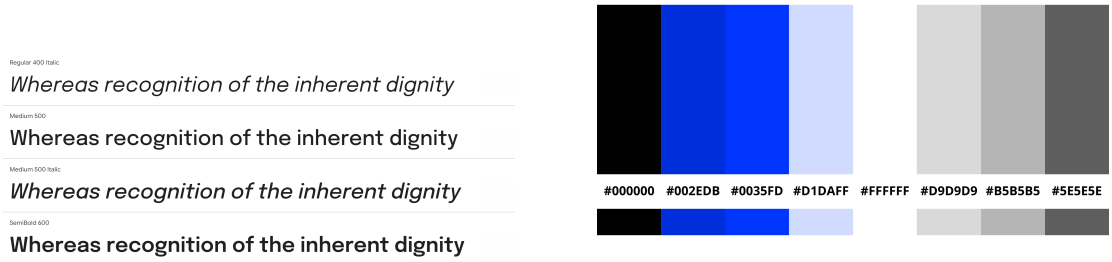
El proceso de diseño de la aplicación comenzó una vez finalizada la fase de investigación. Dada mi escasa formación en diseño gráfico y UI, pero mi elevado interés en la materia, llevé a cabo una investigación de las tendencias actuales y de principios de diseño que era importante tener en cuenta. A la hora de plantear un proyecto de estas características, me resulta imprescindible cuidar el aspecto visual y hacerlo lo más intuitivo y atractivo posible, y es por ello que dediqué un tiempo a asegurarme de que el diseño de la aplicación se alineara con esta idea.

Tenía claro que quería una interfaz minimalista, en la que lo importante fuera el contenido, manteniendo vistas sencillas y usables. Lo primero que necesitaba era un nombre con el que identificar a mi aplicación, para ir poco a poco definiendo mi idea más concretamente. Después de un proceso de *brainstorming*, en el que buscaba dar con un nombre conciso pero descriptivo, llegué al homónimo en inglés ***track***, que significa por un lado *pista*, como canción dentro de un álbum, y por otro *buscar* o *rastrear*, lo que me pareció la mezcla perfecta.

Para mantener la consistencia entre vistas a través de la aplicación, definí un *framework* de diseño relativamente básico, que se compone de una paleta de colores, tipografía y componentes reutilizables con un estilo común.

La fuente elegida para el proyecto es Epilogue, combinando distintos pesos y tamaños para apoyar la jerarquía visual. La paleta de colores parte de un tema oscuro como base, con un azul eléctrico de acento, combinado con blanco y varios tonos de gris, consiguiendo el minimalismo deseado para la estética general, dado que las portadas de los álbumes son el punto que lo rompe.

Finalmente, para acompañar el nombre de la aplicación y darle identidad, diseñé un isotipo que actúa de icono de ***track*** en espacios concretos. Consiste de un círculo en el color azul eléctrico de acento de todo el *framework* y sobre este el símbolo clave, que parte de la "t" de ***track***, enroscada sobre sí misma de manera que termina convirtiéndose en una representación de un tocadiscos, presentado en blanco. Junto al isotipo, forma el imagotipo el nombre de la aplicación, en minúsculas, para obtener



(a) Fuente Epilogue

(b) Paleta de colores

(c) Imagotipo

Figura 4.1: Elementos del *framework* de diseño

una imagen del proyecto limpia y moderna.

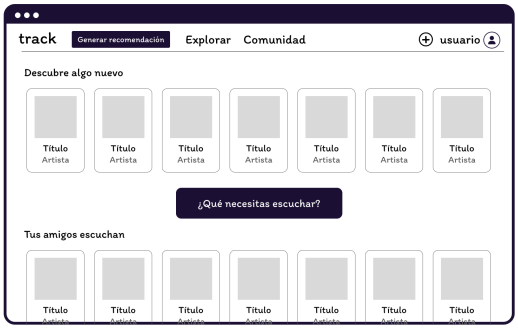
Dado el alcance de un proyecto como este, la aplicación en este punto se plantea como una aplicación web diseñada para ser usada en dispositivos de pantalla grande como ordenadores y tablets. No obstante, se añadieron algunas directivas de *responsiveness* para que se pueda adaptar a diferentes tamaños, pese a no ser un objetivo en esta etapa.

4.2. Wireframes lo-fi

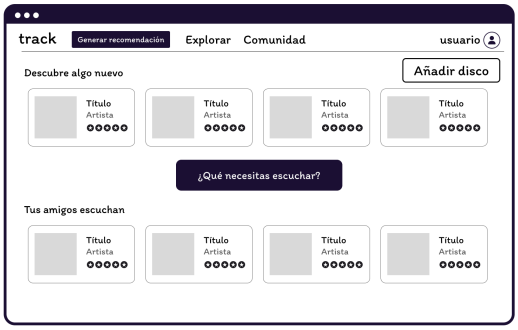
El uso de *wireframes lo-fi* o de baja fidelidad sirve para validar la arquitectura de la información previo a la definición de los elementos de diseño, así como para organizar la estructura de las vistas y elementos que contienen.

En las fases iniciales, contar con bocetos de lo que iba a ser la aplicación me permitió plasmar en vistas y flujos las ideas que tenía para las funcionalidades. Con el desarrollo iterativo, algunas de esas ideas más primitivas cambiaron de forma para ajustarse al flujo de uso y la simplicidad.

Varias vistas tienen dos diseños alternativos. En el caso de la página de Inicio, acabé decidiéndome por la segunda opción, ya que la información de los álbumes me resultó más atractiva en horizontal, decisión que apliqué para el resto de las vistas.

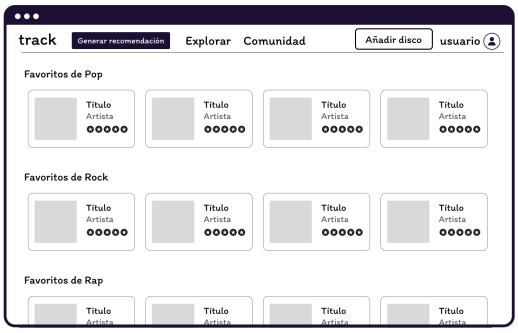


(a) Opción 1 para la vista de Inicio

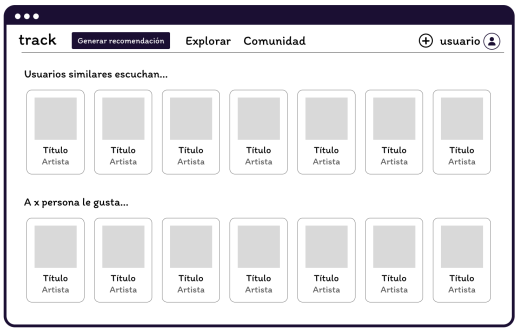


(b) Opción 2 para la vista de Inicio

Figura 4.2: Wireframes lo-fi de la vista de Inicio



(a) Opción 1 para la vista de Comunidad



(b) Opción 2 para la vista de Comunidad

Figura 4.4: Wireframes lo-fi de la vista de Comunidad

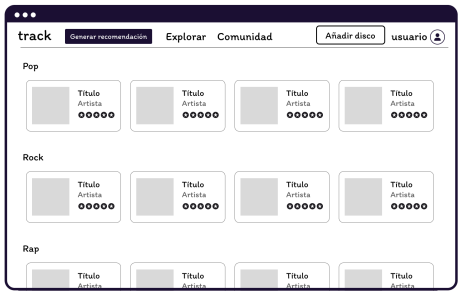


Figura 4.3: Wireframe lo-fi de la vista de Explorar

El concepto para la página de Explorar es el mismo que la de Inicio, pero se plantea que sea aquí donde el usuario reciba las recomendaciones en base a sus gustos, frente a una vista más general en la segunda.

La vista final de Comunidad cuenta con una disposición muy diferente a los *wireframes* de baja fidelidad iniciales, con *tabs* o pestañas para diferenciar novedades de los amigos del usuario, una posible funcionalidad futura de recomendaciones de expertos y una última para descubrir usuarios similares.



(a) Opción 1 para la vista de Perfil (b) Opción 2 para la vista de Perfil

Figura 4.5: Wireframes lo-fi de la vista de Perfil



Figura 4.6: Wireframe lo-fi de la vista de Usuario

Las vistas para las páginas de Usuario y Perfil son muy similares, con la diferencia de que la segunda tiene algún elemento adicional que es privado para el usuario.

4.3. Wireframes hi-fi y prototipo

Para la elaboración de *wireframes* de alta fidelidad y el prototipo se utilizó Figma. Partiendo de los de baja fidelidad y del *framework* de diseño descrito al principio, fui definiendo con más detalle los bocetos de cara a las vistas definitivas, ajustando la disposición de los elementos a medida que se iba precisando cada funcionalidad. También utilicé componentes en Figma (figura 4.7), que en la traducción al *frontend* del código se mantuvieron bastante similares.

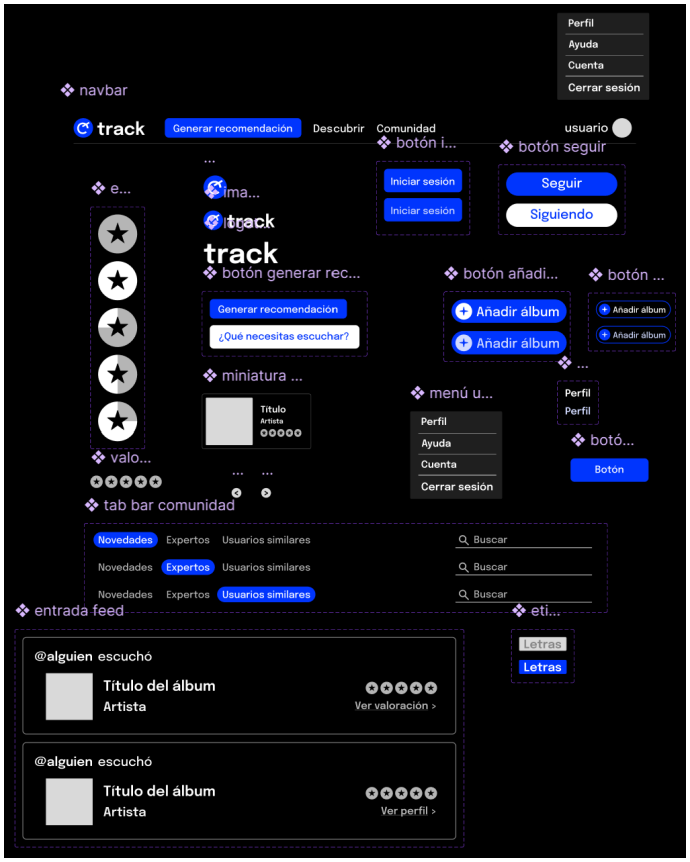
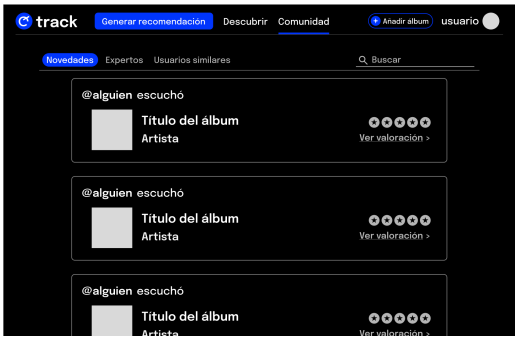


Figura 4.7: Componentes reutilizables en Figma

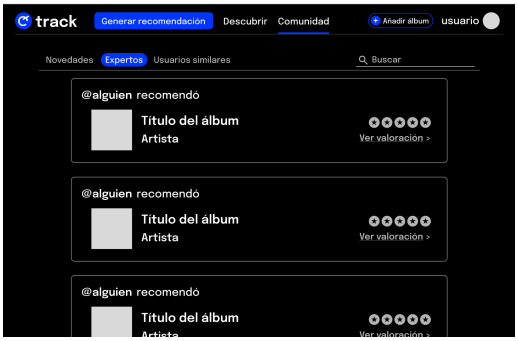
A nivel estructural, las vistas de Comunidad y Perfil/Usuario fueron las que experimentaron los cambios más significativos respecto a sus *wireframes lo-fi*. En el caso de Comunidad (figura 4.8), pasó de tener una disposición idéntica a la de Inicio y Explorar (posteriormente Descubrir), a tener una organización en pestañas para las secciones de Novedades, Expertos y Usuarios Similares. En cuanto a Perfil y Usuario (figura 4.9), se terminó eligiendo la opción 1, una vista dividida en tres secciones para asegurar la simplicidad de ambas páginas.

En las figuras se muestra el botón de "Añadir álbum" en el *header*. Finalmente, para mantener las vistas más despejadas, se optó por sacarlo y ponerlo en la esquina superior derecha.

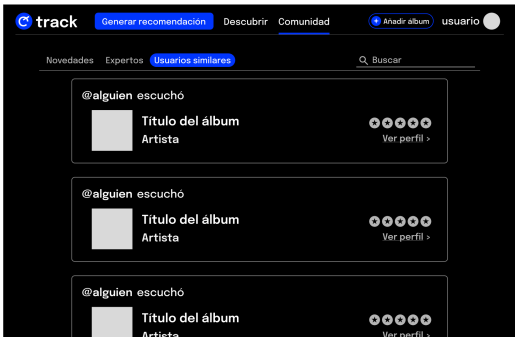
Adicionalmente, para las vistas de inicio de sesión y registro (figura 4.10), se optó por usar un elemento llamativo como es el vinilo en el azul característico del *framework* de diseño para protagonizarlas. Al tratarse de las primeras vistas con las que entra en contacto el usuario, un elemento visual fuerte como este comunica de inmediato el concepto de la aplicación y despierta la curiosidad del usuario, sin depender de texto ni de contenido adicional.



(a) Pestaña "Novedades" de la vista de Comunidad



(b) Pestaña "Expertos" de la vista de Comunidad



(c) Pestaña "Similares" de la vista de Comunidad

Figura 4.8: Wireframes lo-fi de la vista de Perfil

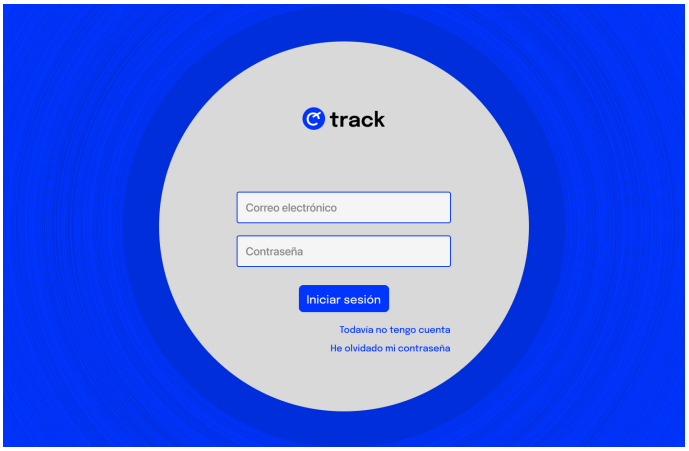


Figura 4.10: Wireframe hi-fi de la vista de inicio de sesión

Una vez terminadas las vistas, utilicé las herramientas de prototipado de Figma para conectarlas y determinar la navegación entre ellas de forma interactiva, validando los flujos principales de la aplicación, y así después trasladarlos a la estructura de rutas de Next.js en la implementación.



(a) Vista de Perfil hi-fi



(b) Vista de Usuario hi-fi

Figura 4.9: Wireframes hi-fi de las vistas de Perfil y Usuario

Capítulo 5

Arquitectura de la aplicación

5.1. Arquitectura del sistema

Una vez definida la interfaz de la aplicación, era necesario concretar cómo se iban a organizar y comunicar entre sí los distintos componentes que la conforman. En la figura 5.1 se recoge la arquitectura general, desde la petición que realiza el cliente hasta el acceso a los servicios externos a los que accede la aplicación.

En el lado del cliente, el navegador renderiza los componentes de React, que combinan *Server Components* y *Client Components*. Todas las peticiones atraviesan primero el `proxy.ts` de Next.js, que se encarga de verificar, mediante la biblioteca *jose*, el JWT almacenado en la cookie de sesión del usuario, protegiendo así las rutas que requieren autenticación y redirigiendo al inicio de sesión en caso de que esta no se cumpla.

La carpeta de rutas `app/` se organiza en dos *route groups* (grupos de rutas): `(init)`, que agrupa las vistas de registro e inicio de sesión previas a acceder a la aplicación, y `(main)`, que contiene el resto (Inicio, Descubrir, Comunidad, Álbum, Perfil, Valoración, etc.), accesibles únicamente tras superar la autenticación y verificación del *proxy*.

Para las operaciones de autenticación (inicio de sesión, registro, cierre de sesión y borrado de cuenta) la aplicación utiliza *Server Actions*, definidas en archivos `actions.ts` en el directorio de cada vista y en `lib/auth/`. Estas funciones acceden directamente a los repositorios de `lib/` para validar credenciales, y manipulan la cookie de sesión mediante la API `cookies()` de `next/headers`. De hecho, la aplicación aprovecha una de las principales ventajas del *App Router* de Next.js al no contar con capas de API REST propias: tanto las operaciones de autenticación como el resto de operaciones sobre datos se resuelven desde componentes de servidor y *Server Actions* que acceden directamente a la capa de acceso a datos.

La conexión con la base de datos se encuentra en el directorio `lib/`. Ahí, los repositorios (`neo4j/repos`) encapsulan las consultas Cypher necesarias para cada tipo de entidad, devolviendo a las vistas objetos adaptados a lo que necesitan. Estos repositorios se apoyan en una conexión *singleton* al *driver* de Neo4j (`neo4j/driver`).

5.2. Diseño del modelo de datos

En una aplicación como ***track***, las relaciones entre usuarios y álbumes son el contenido central de gran parte de la funcionalidad. Un modelo relacional para la base de datos exigiría tablas intermedias para cada una de estas relaciones. El modelo de grafo es una representación más directa de los datos que se usan, reduciendo la distancia entre el modelo mental de una red social como esta y el modelo de datos en sí mismo (figura 5.2).

En este contexto cobra importancia la *index-free adjacency*, una característica de Neo4j en la que cada nodo mantiene punteros directos a los nodos con los que está relacionado. Atravesar una relación tiene coste constante y, por tanto, el rendimiento de las consultas solo depende del número de relaciones atravesadas, no del tamaño total del grafo (Robinson et al., 2015). En un modelo relacional, cada `JOIN` adicional degrada el rendimiento a medida que las tablas crecen, a pesar de tener índices bien diseñados.

Un ejemplo es la pestaña de novedades de la vista de Comunidad: para mostrar a un usuario las valoraciones recientes de sus amigos, se recorre en un único patrón la relación `(Usuario)-[AMIGO_DE {estado: 'aceptada'}]-(Usuario)-[REALIZA]->(Valoracion)-[SOBRE]->(Album)`, combinando en un solo salto a un usuario con su red social, sus valoraciones y los álbumes correspondientes. Esta misma consulta en un modelo relacional exigiría al menos tres combinaciones `JOIN` sobre una tabla de amistades que, tendría que resolver su propia bidireccionalidad adicionalmente.

En las recomendaciones basadas en géneros favoritos y en álbumes bien valorados por otros usuarios, se recurre a `NOT EXISTS` en el patrón de búsqueda para excluir de los resultados los álbumes que el usuario ya ha valorado o marcado como favorito. Esta comprobación de no existencia sustituye lo que en SQL requeriría una subconsulta o un `LEFT JOIN` con filtro sobre valores nulos, resolviéndose como parte natural del propio patrón de coincidencia.

Además, el modelo permite resolver en una única consulta operaciones que combinan varias escrituras relacionadas entre sí. Por ejemplo, al registrar una valoración, la misma sentencia Cypher crea o actualiza el nodo correspondiente, fija la fecha en la relación de autoría, elimina la marca de "guardado para después" si el álbum estaba pendiente, sustituye la canción favorita anterior si la hubiera, y recalcula la puntuación media del álbum a partir de todas sus valoraciones.

Por último, varias relaciones almacenan propiedades que después se explotan en las propias consultas, como la fecha de la relación de autoría de una valoración, que se usa tanto para ordenar el perfil de un usuario y el feed de novedades como para filtrar valoraciones de un año concreto.

5.2.1. Nodos y propiedades

El modelo de datos de la aplicación se organiza en siete tipos de nodo, con sus respectivas propiedades:

- **Usuario**: `email`, `nombre`, `password`.
- **Album**: `discogs_id`, `titulo`, `url_port`, `puntuacion` (media de las valoraciones recibidas).
- **Artista**: `nombre`.
- **Cancion**: `titulo`, `cancion_id`.
- **Genero**: `nombre`.
- **Estilo**: `nombre`.
- **Valoracion**: `puntuacion`, `comentario`.

La unicidad de los distintos elementos del modelo está garantizada mediante *constraints* definidas en Neo4j. Existen restricciones de unicidad (*node property uniqueness*) sobre `Album.discogs_id`, `Artista.nombre`, `Genero.nombre`, `Estilo.nombre`, `Cancion.cancion_id`, `Usuario.email` y `Usuario.nombre`.

El código de los repositorios evita valoraciones duplicadas mediante `MERGE` al escribir sobre **Valoracion**, emparejando por el par (`usuario`, `idAlbum`), de manera que si el usuario ya había valorado ese álbum, se actualiza la información del nodo existente en lugar de crear uno nuevo. Esto se refuerza en la base de datos mediante una restricción de tipo *node key* sobre ese par de propiedades, indexando automáticamente la combinación evitando que cada `MERGE` tenga que recorrer todos los nodos **Valoracion** para comprobar si ya hay uno y protege la integridad del modelo frente a escrituras futuras que no pasen por ese código en el repositorio.

5.2.2. Relaciones

En cuanto a las relaciones entre los nodos anteriores:

- **(Artista)-[PUBLICA {anyo}]->(Album)**.
- **(Album)-[CONTIENE]->(Cancion)**.
- **(Album)-[PERTENECE_A]->(Genero)** y **(Album)-[EN_ESTILO]->(Estilo)**: clasificación del álbum según lo dispuesto en la base de datos de Discogs.
- **(Usuario)-[REALIZA {fecha}]->(Valoracion)-[SOBRE]->(Album)**.
- **(Usuario)-[FAVORITO {fecha}]->(Album)**.

- **(Usuario)-[GUARDADO_PARA_DESPUES {fecha}]->(Album)**.
- **(Usuario)-[AMIGO_DE {estado, fecha}]->(Usuario)**: la propiedad `estado` representa las solicitudes de amistad pendientes y aceptadas, evitando la necesidad de un nodo o tipo de relación adicional.
- **(Valoracion)-[CANCION_FAVORITA]->(Cancion)**.

5.3. Conexión con la API de Discogs

Por otro lado, el fichero `discogs.ts` actúa como cliente de la API REST de Discogs, al contener las llamadas necesarias y encapsulando las funciones (`searchDiscogsMasters` y `getMaster`), que devuelven directamente los datos que utiliza la aplicación.

La búsqueda de álbumes sigue una estrategia *local-first*: `searchMasters` consulta los álbumes ya existentes en Neo4j y solo recurre a `searchDiscogsMasters` cuando no encuentra el resultado en local. De esta forma, cada álbum se importa desde Discogs únicamente una vez. Cuando se selecciona un resultado externo, `ingestAndReturn` obtiene los datos completos con `getMaster` y los introduce en la base de datos (`insertDiscogs`) y así las búsquedas posteriores del mismo álbum se resuelven al completo en Neo4j. Este enfoque, frente a la alternativa de poblar la base de datos directamente con el catálogo completo de Discogs, permite que la base de datos crezca a medida que los usuarios añaden álbumes nuevos.

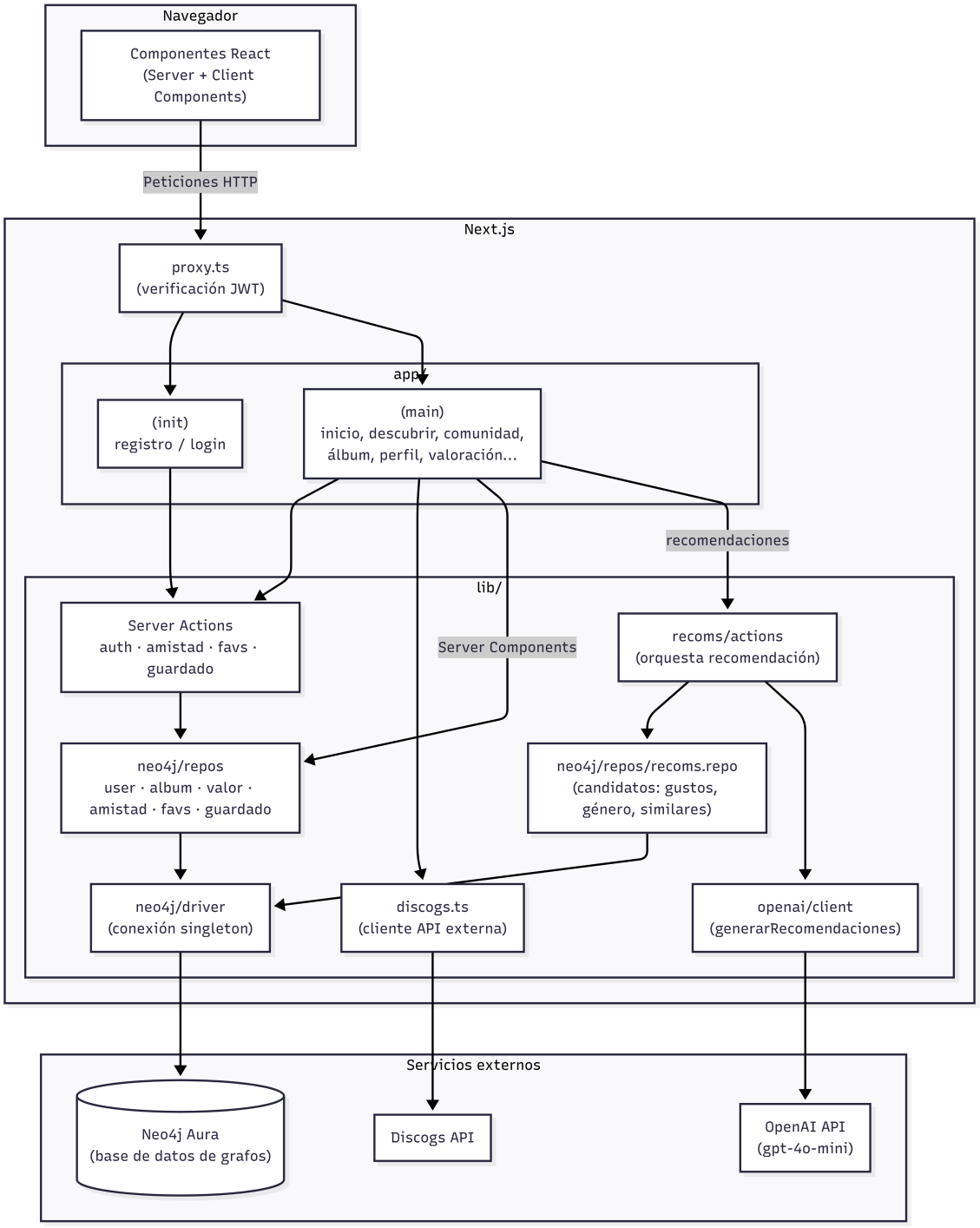


Figura 5.1: Arquitectura general de la aplicación

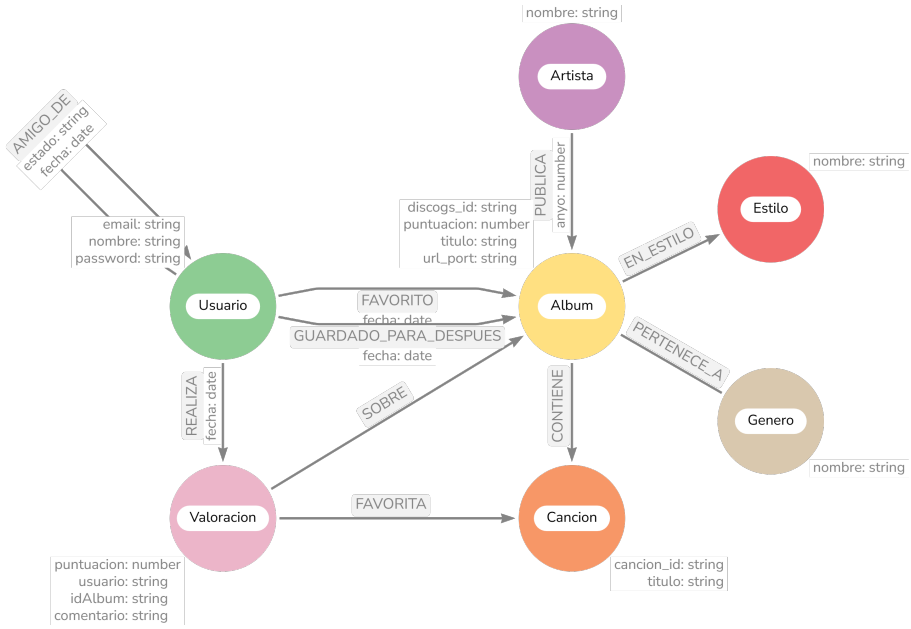


Figura 5.2: Esquema del modelo de datos en Neo4j

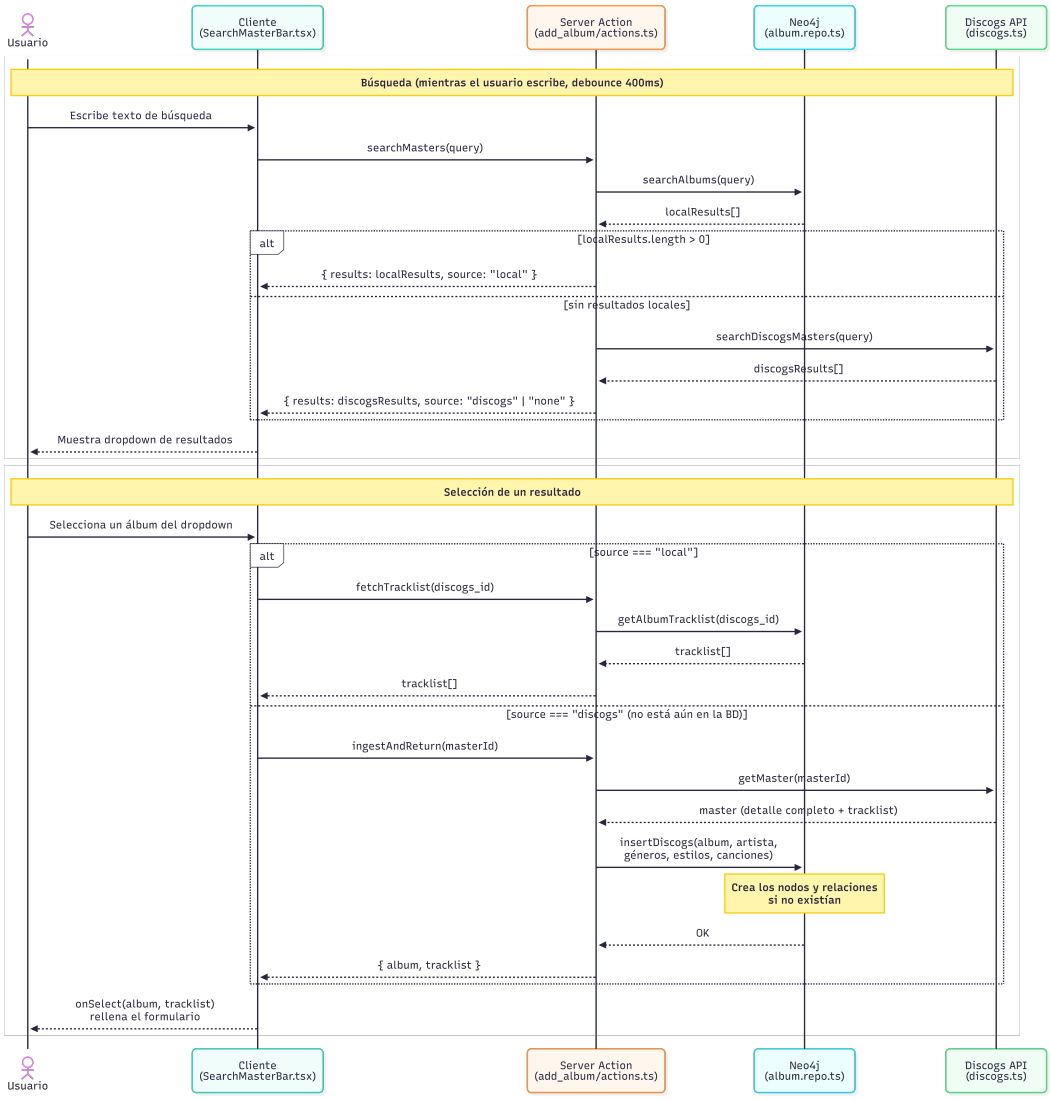


Figura 5.3: Diagrama de secuencia del flujo de llamada a la API de Discogs

Capítulo 6

Sistema de recomendaciones

Los LLMs (*Large Learning Models* o Modelos de Lenguaje de Gran Escala) son uno de los avances más significativos de la Inteligencia Artificial Generativa por su capacidad para procesar, resumir y generar texto propio de lenguaje humano. En su nivel más básico, un LLM opera como una máquina compleja de texto predictivo, cuya función es calcular probabilísticamente la continuación más viable de una secuencia de palabras basándose en correlaciones estadísticas aprendidas de sus datos de entrenamiento.

Esta naturaleza probabilística supone restricciones notables en entornos que exigen precisión absoluta. Dado que el entrenamiento de un LLM se realiza en un punto específico en el tiempo, sus datos son estáticos. El modelo carece de acceso nativo a información generada tras su fecha de corte o a bases de datos privadas y propietarias. Además, cuando se le solicita información que no posee en su conjunto de entrenamiento, tiende a generar textos altamente coherentes y convincentes, pero carentes de veracidad factual, conocidos como alucinaciones. Este comportamiento es sensible al parámetro de temperatura del modelo; una temperatura elevada incrementa la aleatoriedad y la probabilidad de generar declaraciones falsas, mientras que una temperatura baja favorece respuestas más precisas y consistentes.

Por otro lado, los LLM actúan habitualmente como "cajas negras". Sus procesos de decisión son complejos y dificultan la interpretación del razonamiento exacto detrás de una respuesta.

Para superar estas limitaciones, la práctica de *prompt engineering* recurre a una técnica denominada "anclaje", que consiste en suministrar un contexto explícito con la consulta para limitar y guiar la generación del modelo, forzándolo a depender de la evidencia empírica ofrecida en lugar de sus pesos probabilísticos.

6.1. RAG

La Generación Recuperada por Contexto (RAG) es un diseño arquitectónico desarrollado para sistematizar el proceso de anclaje. RAG extrae información relevante

de fuentes externas y la inyecta directamente en el contexto del *prompt*, mitigando el riesgo de las alucinaciones mencionadas sin necesidad de incurrir en procesos de reentrenamiento mucho más costosos (Lewis et al., 2020).

El flujo de trabajo estándar de la arquitectura RAG se estructura en tres etapas secuenciales. Primero, el sistema procesa la entrada en lenguaje natural del usuario para interpretar semánticamente la consulta. A continuación, un componente especializado conocido como *retriever* (recuperador) ejecuta una búsqueda en las fuentes externas del sistema para seleccionar los fragmentos que poseen mayor relevancia respecto a la consulta original. Finalmente, el sistema concatena los fragmentos recuperados en el prompt de entrada del LLM. Con este contenido, el LLM lee el contexto inyectado y sintetiza una respuesta final.

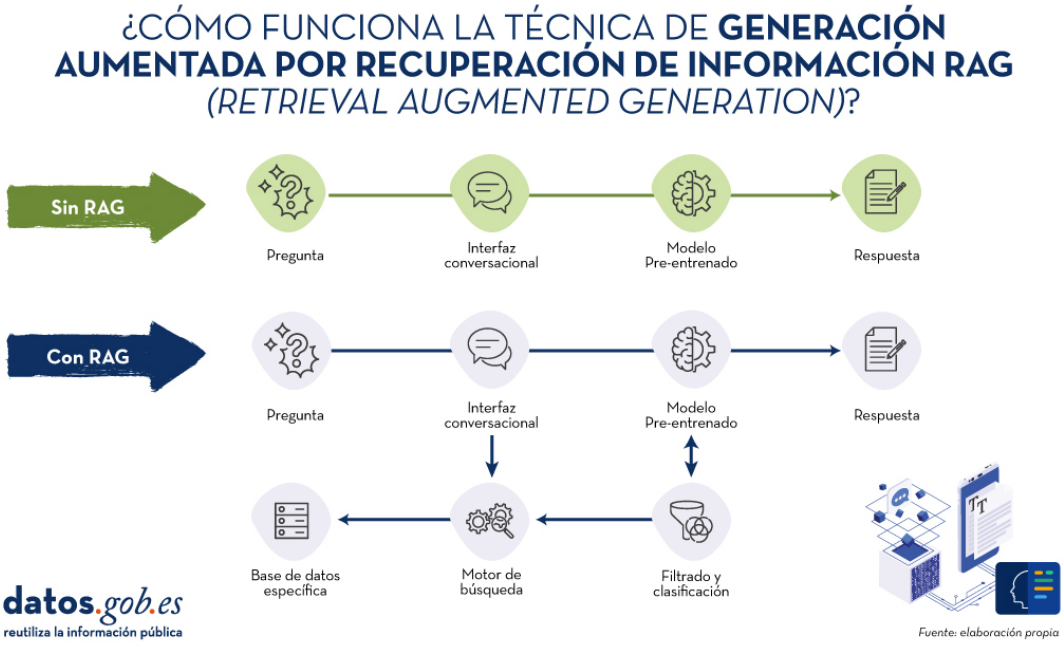


Figura 6.1: Diagrama conceptual del funcionamiento de un sistema con y sin RAG (*Fuente: Datos abiertos del Gobierno de España*)

6.1.1. GraphRAG

GraphRAG es una variante de RAG en la que la fuente externa de la que se recupera contexto es un grafo de conocimiento (Edge et al., 2024). En lugar de seleccionar los fragmentos más relevantes por similitud semántica, el *retriever* recorre relaciones explícitas del grafo mediante Cypher, para aprovechar que el propio grafo codifica qué entidades están relacionadas entre sí y cómo.

En ***track***, el paso de recuperación se resuelve en el momento de la consulta, recorriendo las relaciones del usuario y del catálogo de álbumes para extraer un subgrafo acotado de candidatos, junto con las razones concretas que los conectan con el usuario. Ese subgrafo, se serializa como contexto estructurado y es lo que se inyecta

en el *prompt* del LLM. Es entonces cuando queda restringido a seleccionar y justificar su elección únicamente a partir de esas señales, sin posibilidad de recomendar un álbum que no forme parte del subgrafo recuperado, evitando las alucinaciones.

6.2. Arquitectura del sistema de recomendaciones

El sistema de recomendaciones se organiza en tres capas con responsabilidades independientes, tal y como recoge la figura 6.2: una primera de recuperación sobre el grafo, una de orquestación que decide cuándo recurrir al LLM, y otra de generación que produce la selección final y su justificación.

La capa de recuperación se encuentra en `recomendacion.repo.ts` y expone tres funciones, cada una asociada a un origen distinto (`getCandidatosGustos`, `getCandidatosPorGenero` y `getCandidatosSimilares`). Las tres siguen el mismo patrón: a partir de las relaciones `(Usuario)-[REALIZA]->(Valoracion)-[SOBRE]->(Album)` y `(Album)-[PERTENECE_A]->(Genero)`/`(Album)-[EN_ESTILO]->(Estilo)`, obtienen el perfil de gustos para la consulta y recorren el catálogo para identificar candidatos que compartan género, estilo o artista, o que hayan sido valorados positivamente por un amigo. El resultado es un conjunto de candidatos junto con la evidencia concreta que los conecta con la consulta: los géneros o estilos compartidos y, en su caso, el amigo que valoró el álbum y su puntuación, y es este el contexto que se traslada a la capa de generación.

La segunda capa, en `lib/recomendaciones/actions.ts`, decide si la selección final requiere intervención del LLM. Cuando el número de candidatos recuperados es igual o menor al número de resultados que se van a mostrar, el conjunto es lo bastante reducido como para delegar en el modelo una elección sobre él, por lo que se devuelven directamente los candidatos ordenados por el propio Cypher, sin invocar la API. El LLM solo se invoca cuando existe un conjunto de candidatos mayor sobre el que realmente hay que priorizar, de forma que su uso queda acotado a los casos en los que aporta valor añadido.

La capa de generación (`lib/openai/client.ts`), recibe los candidatos y una instrucción textual específica de la consulta y los traslada al modelo *gpt-4o-mini* mediante la funcionalidad de *structured outputs* de la API de OpenAI. La respuesta se restringe a un esquema JSON en el que el identificador de cada álbum recomendado (`discogs_id`) se define como una enumeración cerrada con los identificadores de los candidatos recibidos, de forma que el modelo no puede seleccionar ningún álbum que no forme parte de esta lista. El título, artista, portada y puntuación de cada álbum recomendado se recuperan directamente del grafo.

6.2.1. Ejemplo de ejecución

Para ilustrar de manera sencilla el funcionamiento de las tres capas, se considera un usuario que ha valorado positivamente dos álbumes: *Rumours* y *Graceland*, ambos clasificados con género *Pop* y estilo *Pop Rock*. Además, un amigo suyo ha valorado

con un 5 el álbum *Blue*.

La función `getCandidatosGustos` recorre el grafo a partir de esas valoraciones y devuelve los siguientes candidatos, junto con la evidencia que los conecta con el usuario:

Álbum	**Género/estilo común**	**Amigo**
The Rise And Fall Of A Midwest P...	Pop, Pop Rock	—
Cabildo Y Juramento	Pop, Pop Rock	—
Blue	Pop, Pop Rock	Raquel (5)

Como el número de candidatos (3) supera al número de resultados a mostrar (2), la capa de orquestación decide invocar al LLM en lugar de devolver los candidatos ordenados directamente por Cypher.

El contexto recuperado se serializa e inyecta en el prompt, restringiendo al modelo a elegir únicamente entre los `discogs_id` recibidos:

```
Candidatos:
- discogs_id: 3255691, titulo: "The Rise And Fall Of A Midwest Prin-
  cess", genero/estilo compartido: Pop, Pop Rock
- discogs_id: 2329972, titulo: "Cabildo Y Juramento", genero/estilo
  compartido: Pop, Pop Rock
- discogs_id: 47744, titulo: "Blue", genero/estilo compartido:
  Pop, Pop Rock, valorado por amigo: Raquel (5)

Selecciona 2 álbumes de la lista anterior y justifica brevemente
cada elección usando únicamente la evidencia proporcionada.
```

El modelo responde con un objeto JSON en el que `discogs_id` está restringido a los tres valores anteriores:

```
{
  "recomendaciones": [
    {
      "discogs_id": 47744,
      "justificacion": "Comparte genero y estilo con tus valoraciones
      y fue valorado con un 5 por tu amigo Raquel."
    },
    {
      "discogs_id": 2329972,
      "justificacion": "Comparte genero (Pop) y estilo (Pop Rock)
      con los albumes que has valorado positivamente."
    }
  ]
}
```

La aplicación recupera del grafo el título, artista, portada y puntuación de cada `discogs_id` devuelto, y los muestra en la sección correspondiente.

El sistema se integra en tres vistas de la aplicación. En la página de Inicio, en la sección "Explora nuevas propuestas", se utiliza `getRecomendacionesGustos`. En la página de Descubrir conviven las recomendaciones basadas en gustos, el ranking de mejor valorados del año sin generación, y recomendaciones del género más frecuente entre las valoraciones positivas del usuario (`getRecomendacionesGenero`). Por último, en la vista de detalle de cada álbum, la sección "Álbumes similares" emplea `getSimilaresAAlbum`.

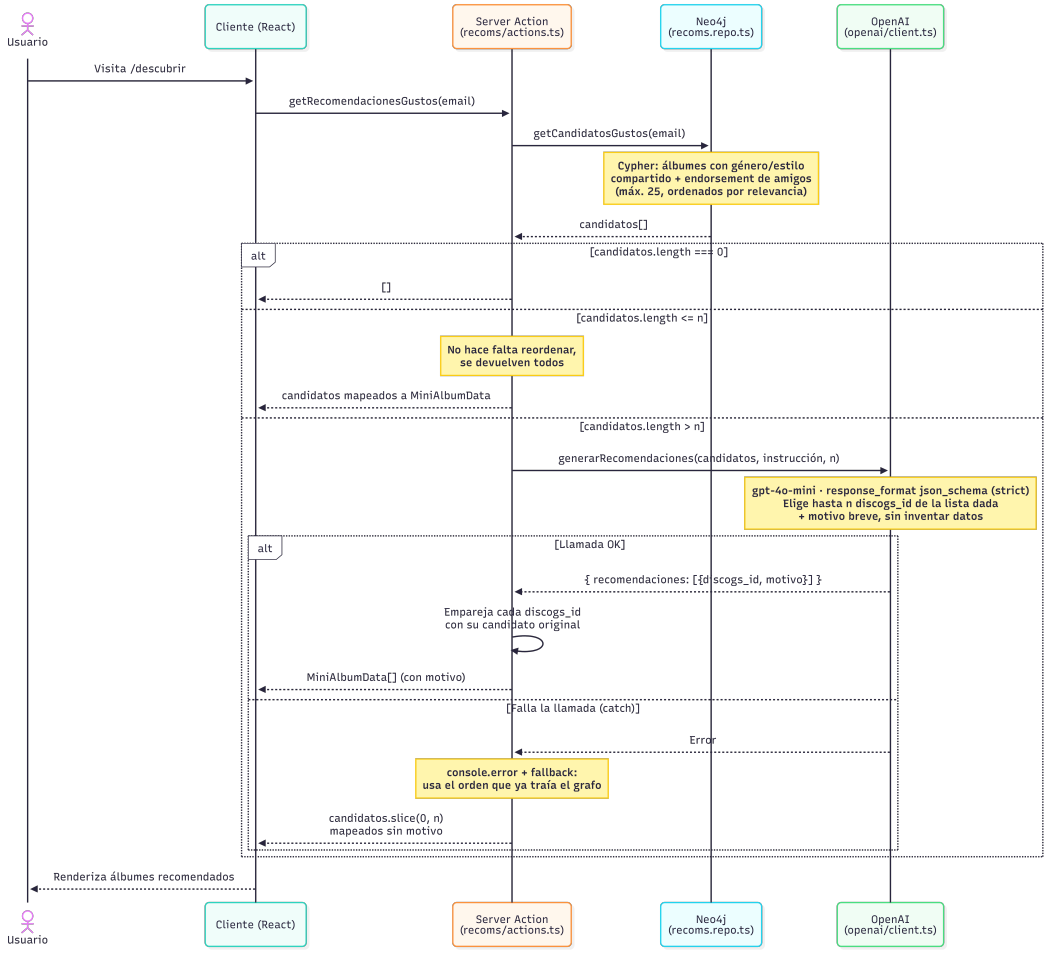


Figura 6.2: Diagrama de secuencia del sistema de recomendaciones

Capítulo 7

Desarrollo iterativo de la aplicación

7.1. Primera iteración

La primera parte de la fase de desarrollo tuvo como objetivo construir la versión inicial de la aplicación, con la funcionalidad básica completa, dejando fuera únicamente el sistema de recomendaciones.

El punto de partida fue la implementación del *frontend* a partir de los *wireframes* de alta fidelidad de Figma, traduciendo cada vista a componentes de React. Una vez se tuvo, había que determinar la arquitectura de la solución de autenticación para el proyecto, así como definir el modelo de datos al detalle. Además, es en este punto donde se realiza una investigación acerca de los posibles abordajes para la autenticación.

En un primer enfoque, se contempló usar *next-auth*, un paquete de `npm` que proporciona la lógica de autenticación para aplicaciones de Next.js. Sin embargo, terminé descartándolo ya que me pareció que añadía demasiada complejidad a un sistema de autenticación basado únicamente en correo y contraseña.

En su lugar, acabé decidiendome por implementar esta lógica con JWT (*Jason Web Token*), un estándar que define una forma de transmitir información de forma segura entre partes como un objeto JSON (Jones et al., 2015). Para ello utilicé dos bibliotecas:

- *bcryptjs*: para el almacenamiento de las contraseñas, que transforma en un *hash*. En el inicio de sesión, utiliza `bcrypt.compare()` para asegurar que se introduce la correcta.
- *jose*: para la emisión y verificación del token JWT. Esta biblioteca se encarga de gestionar la firma HMAC-SHA256 del token. Al iniciar sesión, la función `SignJWT` genera un token firmado con un secreto (`JWT_SECRET`) que contiene únicamente los datos necesarios para identificar al usuario y una caducidad de dos horas.

La verificación se centraliza en un único punto: el archivo `proxy.ts`, que se

ejecuta en prácticamente cualquier petición a la aplicación, comprueba si la ruta solicitada requiere sesión iniciada y, en ese caso, usa `jwtVerify` para validar que el token de la cookie es correcto y no ha caducado antes de dejar pasar la petición. En caso de que el token sea incorrecto o que haya caducado, redirige al inicio de sesión. Asimismo, si el usuario ya tiene la sesión iniciada e intenta acceder a las páginas de inicio de sesión o registro, se le redirige directamente a la vista principal. De esta forma, toda la lógica de autorización de la aplicación queda concentrada en un único fichero.

Este enfoque me permitió simplificar en gran medida la arquitectura de esta parte del código, y centrarme más en refinar el diseño del modelo de datos y el resto de la funcionalidad de la aplicación.

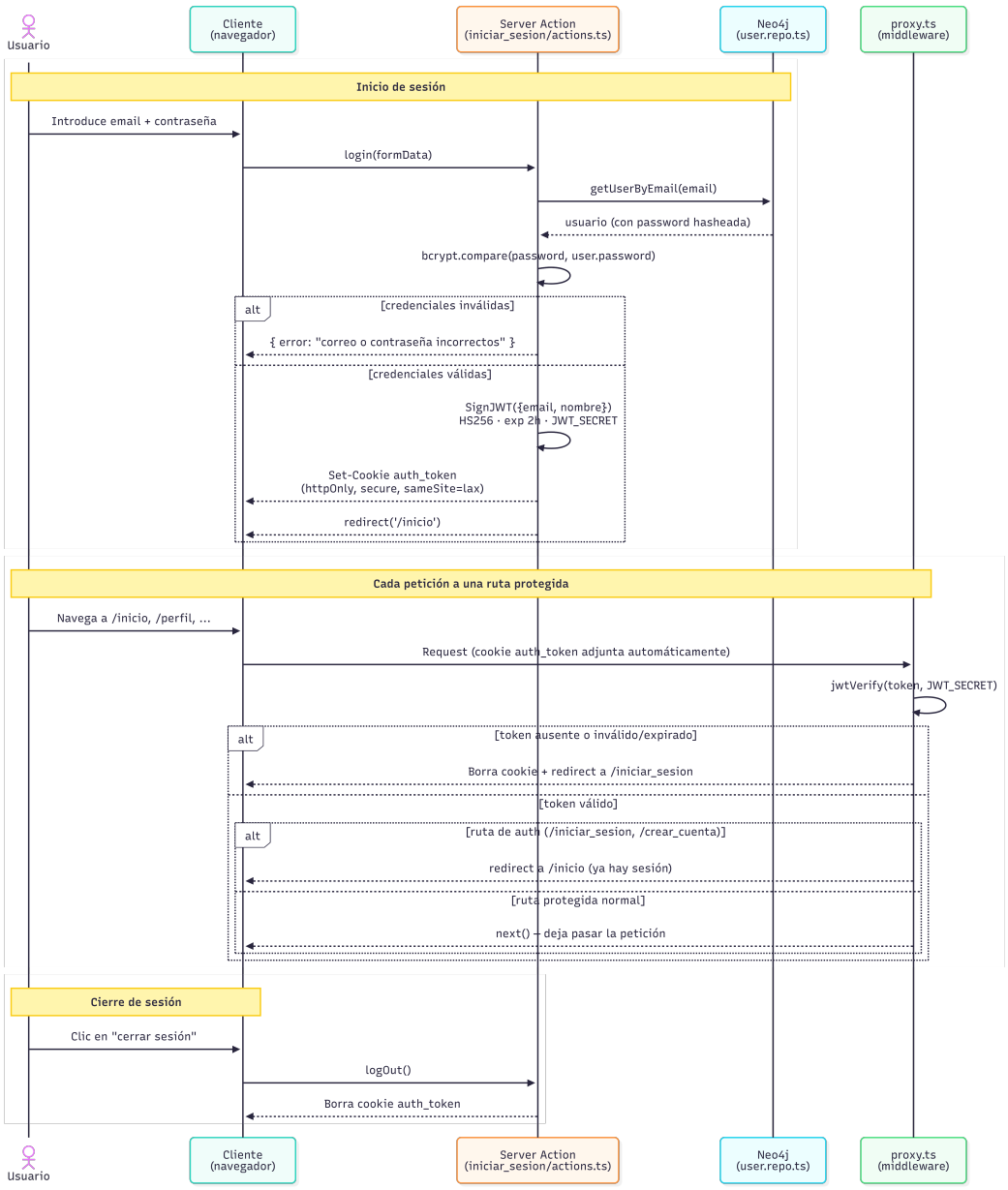


Figura 7.1: Diagrama de secuencia del flujo de autenticación

En la misma línea, y de acuerdo con la convención de Next.js para las rutas dinámicas, la página de confirmación tras valorar un álbum (`/valoracion/[idAlbum]/[emailToken]`) no incluye el correo del usuario en la URL, sino un token cifrado con AES-256-GCM, que evita que quede expuesto.

Para la base de datos, comencé trabajando con Neo4j Desktop en local. Una vez tuve claro el modelo de datos y lo había traducido a nodos y relaciones de Neo4j, con sus respectivas propiedades, definí las *constraints* para asegurar la unicidad de ciertos parámetros y la integridad del modelo en conjunto.

A continuación me dediqué a definir la interacción con la API de Discogs. En este punto se planteó la posibilidad de prescindir de la API y poblar la base de datos directamente, pero se descartó por el elevado volumen de álbumes con los que cuenta la base de datos de Discogs, y puesto que se vio que el flujo de llamadas era relativamente sencillo y eficiente.

Teniendo la base de datos y la conexión con la API de Discogs, llegó el proceso de desarrollo de la funcionalidad de la aplicación. Partiendo de las vistas que había traducido de los *wireframes* de Figma, fui conectando cada una a sus datos reales gradualmente: primero el modelo de álbum, enlazando la ficha de cada disco (`app/(main)/album/[IDalbum]`) y el formulario para añadir una valoración (`app/(main)/add_album`) con `album.repo.ts` y el cliente de Discogs. Les siguieron las valoraciones y las relaciones sociales del grafo, que alimentan las vistas de Comunidad, Descubrir y Perfil.

7.1.1. Despliegue

Hasta este punto, las pruebas se habían estado haciendo ejecutando el proyecto en local con `npm`, pero de cara a las evaluaciones, y en especial a la entrega del trabajo, había que hacer el despliegue en un entorno que permitiera publicar la app.

El despliegue se resolvió con Netlify, una vez se hubo migrado la base de datos a Neo4j Aura (figura 7.2). El proceso se llevó a cabo de forma sencilla y fluida.

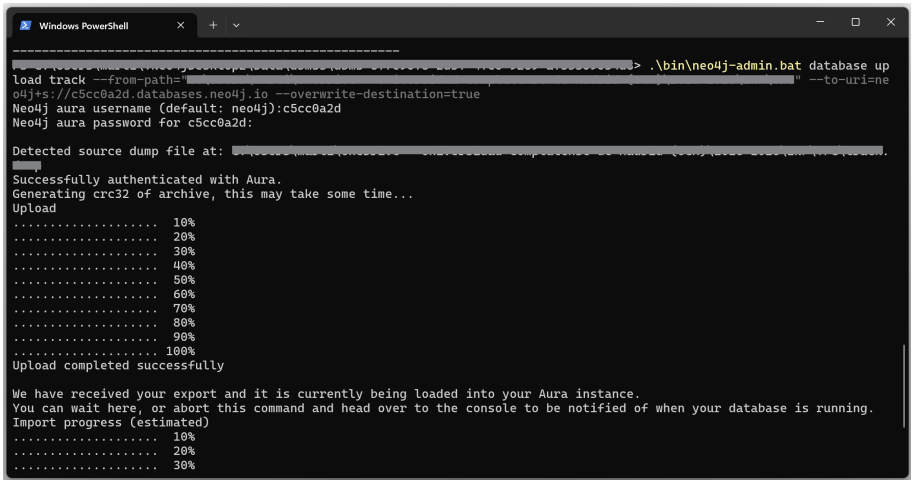


Figura 7.2: Migración de Neo4j Desktop a Neo4j Aura

7.1.2. Evaluación con usuarios

Una vez la app fue funcional, a excepción, como se ha mencionado, de la parte de las recomendaciones, se procedió a llevar a cabo una primera fase de evaluaciones para validar la usabilidad de ***track*** con cinco participantes: dos de ellos, sin conocimientos técnicos, y otros tres con perfil técnico/de desarrollo. La evaluación consistió de varias tareas guiadas con la técnica de *think aloud* y, al finalizar, el cuestionario SUS (*System Usability Scale*) (Brooke, 1996).

La puntuación del cuestionario SUS para esta primera fase osciló entre 32,5 y 100 puntos, con una media de 77,5 (véase Apéndice B), resultado que sitúa la usabilidad percibida en un nivel aceptable, aunque fuertemente condicionado por el participante que encontró más incidencias técnicas.

7.1.2.1. Hallazgos

De las evaluaciones con usuarios de la primera iteración surgieron los siguientes hallazgos, agrupados por área:

- **Puntuación por estrellas**: el mecanismo de incremento secuencial en cuartos generó confusión y frustración.
- **Botón "Generar recomendación"**: al no estar implementado, el botón no ofrecía ningún tipo de retroalimentación, lo que desconcertó a los usuarios.
- **Contraste y legibilidad**: se señalaron problemas de contraste algunos elementos.
- **Formularios**: ausencia de indicaciones sobre los requisitos de la contraseña, campos que no se vaciaban tras un error de validación, así como la falta de retroalimentación durante el envío.
- **Comunidad**: dificultad para localizar la búsqueda de usuarios y para acceder al perfil de otro usuario desde una tarjeta de la comunidad, además de un truncado incorrecto del contenido cuando este era extenso.

7.2. Segunda iteración

La segunda iteración del desarrollo de la aplicación se centró en implementar las mejoras derivadas de los hallazgos de la primera fase de evaluaciones, así como toda la lógica relativa al sistema de recomendaciones.

7.2.1. Mejoras

A raíz de los hallazgos obtenidos, se realizaron un conjunto de mejoras. En el componente dedicado a darle puntuación a una valoración (`puntuar.tsx`), sustituí

el incremento secuencial por un manejador de clic independiente para cada estrella (`handleStarClick`), que calcula el relleno local de la estrella pulsada y permite ciclar sus cuartos de forma individual; de este modo la puntuación responde a la estrella concreta sobre la que se hace clic, acelerando de manera considerable la tarea de realizar una valoración. El botón *"Generar recomendación"* se sustituyó por un nuevo componente (`gen_button.tsx`) que abre un cuadro de diálogo informando de que la función está en desarrollo, dando así al usuario la retroalimentación que antes faltaba.

En cuanto a la vista de Comunidad, enlacé el nombre de usuario de cada tarjeta a su perfil, y corregí el truncado incorrecto del título y el artista para corregir los problemas de adaptación al contenedor. También mejoré el contraste de la barra de búsqueda de usuarios, sustituyendo los tonos grises más oscuros del icono, el marcador de posición y el borde del desplegable de resultados por otros más claros.

Por otro lado, en los formularios de inicio de sesión, registro y eliminación de cuenta añadí una indicación visual del envío en curso, deshabilité el botón mientras la petición estaba pendiente y sustituí su texto por uno que refleja el estado. Añadí también, bajo el campo de contraseña del formulario de registro, la indicación del número mínimo de caracteres requerido.

7.2.2. Sistema de recomendaciones

Al tiempo que se llevó a cabo el proceso de evaluaciones con usuarios de la primera iteración se desarrollaba en una rama alternativa (`recoms`) la lógica del sistema de recomendaciones. El proceso de implementación consistió en separar la lógica en dos fases.

En primer lugar, una capa de acceso a datos en Neo4j (`recoms.repo.ts`) que, para cada usuario, genera mediante Cypher el conjunto de candidatos. En segundo lugar, la conexión con OpenAI para la selección final y su justificación.

Sobre este mismo patrón se implementaron las tres variantes: recomendaciones según los gustos generales del usuario, según su género favorito y álbumes similares a uno concreto.

Además, fue necesario añadir la clave de la API a las variables de entorno con las que ya contaba el proyecto de Netlify (`OPENAI_API_KEY`).

7.2.3. Evaluación con usuarios

Una vez incorporadas las mejoras anteriores y desarrollado el sistema de recomendaciones, se llevó a cabo una segunda fase de evaluaciones. Esta vez se optó por una exploración libre de la aplicación completa seguida del cuestionario SUS, ampliado con cuatro preguntas específicas sobre la percepción del sistema de recomendaciones.

Los resultados denotan una mejora considerable de la percepción de usabilidad,

puesto que la puntuación SUS media alcanzó 97,5 (véase Apéndice B) sobre 100 (frente a los 77,5 de la primera fase). Las cuatro preguntas específicas sobre las recomendaciones obtuvieron también valoraciones altas, en torno a 4,5 sobre 5 de media, destacando especialmente el descubrimiento de contenido nuevo y la naturalidad con la que las recomendaciones se integran en el resto de la aplicación.

7.2.3.1. Hallazgos

Los comentarios abiertos recogidos fueron mayoritariamente positivos y sin incidencias relevantes ("hacía mucho tiempo que era necesaria una aplicación como esta", "me ha encantado, es algo novedoso y súper útil", "app necesaria para la gente interesada por la música, que busca nuevos estímulos"). Varios participantes destacaron positivamente detalles concretos de la interacción, como la selección de la canción favorita de un álbum mediante un desplegable en lugar de escribirla manualmente. Al tratarse de una exploración libre, no se registraron problemas de usabilidad que impidieran completar la navegación por la aplicación. Con esta segunda fase se dio por validada tanto la resolución de los hallazgos de la primera evaluación como la calidad percibida del sistema de recomendaciones, cerrando así la fase de desarrollo del proyecto.

Capítulo 8

Conclusiones y Trabajo Futuro

8.1. Conclusiones y Trabajo Futuro

Conclusiones

Este trabajo buscaba darle respuesta a una necesidad muy concreta y personal: recibir propuestas de escucha que vencieran la sensación de estancamiento que tanto yo como mi entorno veníamos experimentando por parte de las plataformas de *streaming* habituales, que no dependieran del apoyo de grandes discográficas, apoyadas en una interfaz amigable y sencilla.

En retrospectiva, este trabajo me ha permitido entrar en contacto con la totalidad del proceso de elaboración de una aplicación como ***track***. Inicialmente, no era consciente de la profundidad de la fase de investigación previa al desarrollo de una aplicación. Fue durante el primer cuatrimestre, tanto con este trabajo como con la asignatura de Diseño de Sistemas Interactivos, cuando descubrí todo lo que implica, así como el diseño UI/UX, por el cual he adquirido un interés profundo durante este tiempo.

Trabajar individualmente en todo el proceso de desarrollo de *software*, desde la investigación hasta la implementación *full-stack* y la evaluación iterativa, pasando por decisiones de arquitectura y de diseño, es una experiencia que me ha aportado una comprensión mucho más amplia de este tipo de sistemas, dado que durante la carrera solo las había abordado de forma parcial o de manera aislada.

En cuanto a los objetivos planteados inicialmente, la aplicación ofrece una experiencia visual y de interacción cuidada, validada y refinada con usuarios reales. El modelo de datos permite una personalización real de las propuestas en función de las valoraciones, las relaciones sociales y el comportamiento de cada usuario, y no de criterios ajenos a ellos y sus preferencias. En ese sentido, la aspiración de democratizar la experiencia de recomendación está presente en la arquitectura de la aplicación en global.

No obstante, es importante poner de manifiesto que, dado el alcance de un tra-

bajo de estas características, no es posible demostrar una mejora notable en la experiencia de recomendación musical del usuario en este punto. El sistema de recomendaciones descrito requiere de un uso más exhaustivo de la aplicación para poder ofrecer propuestas de valor para el usuario.

En definitiva, termino este proyecto con la satisfacción de haber dedicado casi un año a un trabajo propio, nacido de un interés personal genuino, que he convertido en una aplicación funcional y que me ha permitido aprender de todos los ámbitos que rodean el desarrollo de una aplicación como ***track***.

Trabajo futuro

En base a los requisitos identificados en la fase de investigación y los hallazgos obtenidos de las evaluaciones con usuarios, se extraen algunas líneas de mejora que pueden ser objeto de trabajo futuro:

- **Escalabilidad del sistema**: tanto las características de la infraestructura de despliegue actual como el modelo de lenguaje escogido son adecuadas para el alcance de este trabajo, pero presentan limitaciones a tener en cuenta de cara a una posible continuación del proyecto. Sería interesante evaluar la migración a planes de pago de estas tecnologías o a una infraestructura autogestionada, así como revisar el coste y los límites de las llamadas a las APIs de OpenAI y Discogs a mayor escala.

  El uso del modelo *gpt-4o-mini* y de *structured outputs* con una enumeración cerrada de identificadores impide que el modelo alucine fuera del subgrafo de candidatos, pero no garantiza la calidad ni la naturalidad de las justificaciones que genera. Habría que evaluar la posibilidad de utilizar un modelo más capaz, o el ajuste fino de uno más ligero sobre ejemplos del dominio musical, mejorando esas justificaciones evitando disparar el coste.

- **Cuestionario de perfil inicial**: un breve cuestionario en el proceso de creación de cuenta que permita definir un perfil de gustos de partida para cada usuario, mitigando el problema de arranque en frío del sistema de recomendaciones y mejorando la personalización desde el principio.

- **Funcionalidad "Generar recomendación"**: una de las conclusiones más relevantes que se obtuvo de una de las entrevistas llevadas a cabo en la fase de investigación fue la de desarrollar una funcionalidad adicional que, a través de un pequeño cuestionario de la situación puntual del usuario, hiciera una recomendación para el momento en concreto.

- **Evaluación con usuarios del prototipo**: el prototipo interactivo de alta fidelidad en Figma solo se validó de forma interna antes de pasar a la implementación. Someterlo a una evaluación con usuarios reales en esa fase, habría permitido detectar problemas de usabilidad de forma más temprana y económica, y llegar a la primera iteración de desarrollo con un flujo ya validado para evitar tener que corregirlo más adelante.

- **Registro de emociones sugeridas**: incorporar la posibilidad de que el usuario indique qué emociones le suscita un álbum, enriqueciendo así la información con la que el sistema aprende de sus preferencias.
- **Recomendaciones de expertos**: la pestaña de Comunidad se diseñó contemplando un espacio futuro para recomendaciones curadas por expertos.
- **Canciones y playlists**: incluir la posibilidad de valorar canciones de forma individual y *playlists* creadas por los usuarios y que se tengan en cuenta para las recomendaciones.
- **Edición de los datos del usuario**: de forma adicional, se plantea un posible desarrollo de la lógica de edición de los datos del usuario, como una imagen para mostrar en el perfil o los datos de la cuenta.

8.2. Conclusions and Future Work

Conclusions

This work sought to fulfil a very specific and personal need: receiving listening suggestions that overcome the sense of standstil that both I and those around me had been experiencing with habitual streaming platforms. They had to be suggestions that did not rely on the backing of major record labels, supported by a friendly and straightforward interface.

In hindsight, this project has allowed me to engage with the entire creation process of an application like ***track***. Initially, I was unaware of the depth of the research phase required prior to developing an application. It was during the first semester, through both this project and the Interactive Systems Design course, that I discovered all that it entails, as well as UI/UX design, in which I have developed a keen interest over this time.

Working individually across the entire software development lifecycle, from research to full-stack implementation and iterative evaluation, through architecture and design decisions, is an experience that has provided me with a much broader understanding of these systems, given that during my degree I had only approached them partially or isolated from the rest of the process.

Regarding the initial objectives, the application offers a refined visual and interaction experience, validated and polished with real users. The data model enables genuine personalisation of suggestions based on each user's ratings, social connections, and behaviour, rather than criteria external to them and their preferences. In that sense, the ambition to democratise the recommendation experience is embedded within the overall architecture of the application.

However, it is important to point out that, given the scope of a project of these characteristics, it is not possible to demonstrate a notable improvement in the user's

music recommendation experience at this stage. The recommendation system described requires more extensive use of the application to offer meaningful proposals to the user.

Ultimately, I complete this project with the satisfaction of having devoted nearly a year to my own work, born from a genuine personal interest, which I have managed to transform into a functional application and which has allowed me to learn from every domain surrounding the development of an application like ***track***.

Future Work

Based on the requirements identified during the research phase and the findings obtained from user evaluations, several avenues for improvement have been identified for future work:

- **System scalability**: both the characteristics of the current deployment infrastructure and the chosen language model are suitable for the scope of this project, but they present limitations to consider regarding a possible continuation of the project. It would be worthwhile to evaluate migrating to paid tiers for these technologies or to a self-managed infrastructure, as well as reviewing the cost and rate limits of the OpenAI and Discogs APIs at a larger scale.

  The use of the *gpt-4o-mini* model and structured outputs with a closed set of identifiers prevents the model from hallucinating outside the candidate subgraph, but it does not guarantee the quality or naturalness of the justifications it generates. It would be valuable to assess using a more capable model, or fine-tuning a lighter one on music-domain examples, thereby improving these justifications without driving up costs.

- **Initial profile questionnaire**: a brief onboarding questionnaire during account creation to define a baseline preference profile for each user, mitigating the cold-start problem of the recommendation system and enhancing personalisation from the outset.

- **"Generate recommendation" feature**: one of the most relevant conclusions drawn from the research interviews was the development of an additional feature that, through a short questionnaire about the user's current situation, provides a recommendation tailored to that specific moment.

- **User evaluation of the prototype**: the high-fidelity interactive prototype in Figma was only validated internally before moving to implementation. Subjecting it to user testing at that stage would have allowed usability issues to be identified earlier and more cost-effectively, reaching the first development iteration with a pre-validated flow to avoid having to correct it later.

- **Evoked emotion logging**: enabling users to indicate the emotions an album evokes, enriching the information the system uses to learn their preferences.

- **Expert recommendations**: the Community tab was designed with a dedicated space for expert-curated recommendations in mind.
- **Tracks and playlists**: allowing users to rate individual tracks and user-created playlists, incorporating these ratings into the recommendation engine.
- **User profile editing**: additionally, implementing the logic for editing user data such as uploading a profile picture or updating account details is proposed.

Bibliografía

Baracskay, I., Baracskay III, D. J., Iqbal, M. y Knijnenburg, B. P. The diversity of music recommender systems. En *Companion Proceedings of the 27th International Conference on Intelligent User Interfaces*, páginas 97–100. 2022.

Brooke, J. SUS: A quick and dirty usability scale. En *Usability Evaluation in Industry* (editado por P. W. Jordan, B. Thomas, B. A. Weerdmeester y I. L. McClelland), páginas 189–194. Taylor & Francis, 1996.

Cooper, A., Reimann, R., Cronin, D. y Noessel, C. *About Face: The Essentials of Interaction Design*. Wiley, 4th edición, 2014.

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S. y Larson, J. From local to global: A graph RAG approach to query-focused summarization. `https://arxiv.org/abs/2404.16130`, 2024.

Google. Escucha mixes propios o personalizados - Ayuda de YouTube Music. `https://support.google.com/youtubemusic/answer/15165061?hl=es-419`, 2025.

Jones, M., Bradley, J. y Sakimura, N. JSON web token (JWT). RFC 7519, IETF, `https://datatracker.ietf.org/doc/html/rfc7519`, 2015.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S. y Kiela, D. Retrieval-augmented generation for knowledge-intensive NLP tasks. En *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, páginas 9459–9474. 2020.

Next.js. Next.js Docs. `https://nextjs.org/docs`, 2026.

Robinson, I., Webber, J. y Eifrem, E. *Graph Databases: New Opportunities for Connected Data*. O'Reilly Media, 2nd edición, 2015.

Spotify. Cómo funcionan las recomendaciones en spotify. `https://www.spotify.com/safetyandprivacy/understanding-recommendations`, 2026.

Apéndice A

Cuestionario

A.1. Guión del cuestionario

Investigación sobre hábitos de escucha y descubrimiento musical

Este cuestionario está diseñado para estudiar la manera en que los aficionados a la música descubren nuevas propuestas para ampliar su rango. La información recopilada es completamente anónima y únicamente será utilizada para mejorar la experiencia de usuario de una aplicación de recomendaciones musicales que se está desarrollando como Trabajo de Fin de Grado en la Facultad de Informática de la UCM.

Este cuestionario no tardará más de 5 minutos en completarse y tu respuesta es de gran ayuda para el desarrollo de la aplicación. ¡Muchas gracias por tu participación y tiempo!

SECCIÓN 1

1. ¿Te gusta escuchar música a diario? Marca solo un óvalo. Sí No

2. ¿Qué géneros de música son tus preferidos? Selecciona todos los que correspondan. Pop Rock Reggaeton / Latino Rap / Hip Hop Clásica Jazz Flamenco Electrónica Otro

3. ¿Qué dispositivo utilizas principalmente para escuchar música? Selecciona todos los que correspondan. Teléfono móvil Ordenador Tablet Altavoz inteligente Sistema de audio Reproductor del coche Otro

4. ¿Qué medio/s utilizas actualmente para escuchar música? Selecciona todos los que correspondan. Spotify Apple Music Amazon Music YouTube SoundCloud Otro

5. ¿Qué es lo más importante para ti al elegir esta plataforma? Selecciona todos los que correspondan. Precio Calidad de sonido El catálogo disponible El algoritmo de recomendaciones La facilidad de uso de la app La posibilidad de crear/compartir playlists Otro

6. ¿De qué manera sueles escuchar música? Selecciona todos los que correspon-

dan. Álbumes completos Playlists hechas por mí Playlists hechas por otros / por la plataforma Listas de éxitos (Top 50, Novedades, etc.) Otro

SECCIÓN 2: Hábitos de escucha

7. ¿Cuántas horas al día sueles escuchar música? Marca solo un óvalo. Menos de 1h Entre 1h y 3h Más de 3h

8. ¿En qué contextos o situaciones sueles hacerlo? Selecciona todos los que correspondan. Mientras viajo o conduzco En el trabajo o estudiando Haciendo ejercicio Para relajarme Antes de dormir En eventos sociales Otro

9. ¿Consideras que la música es una parte importante de tu vida? Marca solo un óvalo. Sí No

10. ¿Por qué?

11. ¿Con qué frecuencia te encuentras estancado en tu gusto musical, sintiendo que escuchas lo mismo todo el tiempo? Marca solo un óvalo. Nunca Rara vez A veces A menudo Siempre

12. ¿De qué manera resuelves esta situación en la actualidad?

13. ¿Te gusta recibir recomendaciones musicales? Marca solo un óvalo. Sí (*Salta a la pregunta 14*) No (*Salta a la pregunta 21*)

SECCIÓN 3: Experiencia actual con las recomendaciones

A continuación, te haremos algunas preguntas que nos ayudarán a perfeccionar las recomendaciones en nuestra aplicación.

14. ¿De qué manera sueles recibir recomendaciones de nueva música? Selecciona todos los que correspondan. De mi aplicación de reproducción de música De amigos / familia De TikTok / Instagram De la radio Otro

15. Del 1 al 7, ¿cómo valoras las recomendaciones que recibes? Marca solo un óvalo. Muy negativo 1 2 3 4 5 6 7 Muy positivo

16. ¿De qué manera podría tu fuente de recomendaciones mejorar esta experiencia?

17. ¿Cuál de las siguientes se parece más a tu experiencia con estas recomendaciones? Marca solo un óvalo. No suelen alejarse demasiado de lo que ya escucho Se alejan demasiado de mi estilo No me suelen gustar Mi fuente de recomendaciones no suele equivocarse Otro

18. Muestra tu grado de acuerdo con la siguiente afirmación: "Me gustaría recibir propuestas lo más diferentes posible a mi gusto actual para abrirme a otros estilos."Marca solo un óvalo. Totalmente en desacuerdo 1 2 3 4 5 Totalmente de acuerdo

19. Muestra tu grado de acuerdo con la siguiente afirmación: "Me gustaría que las recomendaciones fueran basadas en lo que le gusta a la gente de mi entorno."Marca solo un óvalo. Totalmente en desacuerdo 1 2 3 4 5 Totalmente de acuerdo

20. Describe con más detalle, si lo deseas, cómo te gustaría que fueran las recomendaciones que recibas.

SECCIÓN 4

21. ¿Te gusta compartir la música que escuchas con tu círculo? Marca solo un óvalo. Sí No

22. ¿Con qué frecuencia compartes música (canciones, playlists, artistas) con otras personas? Marca solo un óvalo. Nunca Rara vez Ocasionalmente Con frecuencia

23. Muestra tu grado de acuerdo con la siguiente afirmación: "Mi gusto musical está influido por mi entorno."Marca solo un óvalo. Totalmente en desacuerdo 1 2 3 4 5 Totalmente de acuerdo

24. Muestra tu grado de acuerdo con la siguiente afirmación: "Me gustaría conectar con personas con las que comparta gustos musicales."Marca solo un óvalo. Totalmente en desacuerdo 1 2 3 4 5 Totalmente de acuerdo

---

SECCIÓN 5: Datos de interés

Por último, unas preguntas adicionales para refinar la investigación. Te recordamos que son datos completamente anónimos y en ningún caso serán utilizados más allá de los fines académicos del estudio.

25. Género Marca solo un óvalo. Femenino Masculino Prefiero no decirlo Otro

26. ¿Qué edad tienes? Marca solo un óvalo. Menos de 18 años Entre 18 y 25 años Entre 26 y 35 años Entre 36 y 45 años Entre 46 y 55 años Entre 56 y 65 años Más de 65 años

27. ¿Cuál es tu ocupación principal? Selecciona todos los que correspondan. Trabajo Estudio Otro

28. ¿Cuál es tu máximo nivel de estudios alcanzado? Marca solo un óvalo. Ninguno Educación Primaria Educación Secundaria Obligatoria Bachillerato Formación Profesional de Grado Medio Formación Profesional de Grado Superior Diplomatura o Grado Universitario Máster Doctorado

Apéndice B

Instrumentos de evaluación de usabilidad

B.1. Fase 1: evaluación guiada

B.1.1. Preguntas previas

Antes de comenzar las tareas guiadas, se recogieron los siguientes datos de cada participante:

1. Formato de la evaluación. Marca solo un óvalo: Presencial / Remoto.
2. ¿Escucha música con frecuencia? Marca solo un óvalo: Sí / No / A veces.
3. ¿Usa apps similares (Letterboxd, RYM, Last.fm...)?

B.1.2. Tareas guiadas

Se plantearon cuatro tareas guiadas mediante la técnica de *think aloud*. Para cada una de ellas se registraron los mismos cinco indicadores: **resultado** (Éxito / Éxito con ayuda / Fracaso), **tiempo empleado**, **número de errores**, **facilidad percibida** (escala de 1 a 7, de "Muy difícil" a "Muy fácil") y **observaciones** del evaluador.

Tarea 1 – Sign in & Log in

1. Crea una cuenta.
2. Cierra sesión.
3. Inicia sesión de nuevo.

Tarea 2 – Añadir y valorar un álbum

1. Añade la valoración de un álbum que te guste mucho o hayas escuchado recientemente.
2. Dirígete a donde deberías ver la valoración que acabas de hacer.

Tarea 3 – Descubrir / comunidad

1. Dirígete a donde querrías ver las recomendaciones generadas (no funcional todavía).
2. Encuentra al usuario `@martina.ag` y envíale una solicitud de amistad.

Tarea 4 – Perfil / cuenta

1. Dirígete a la vista de tu perfil.
2. Dirígete a la vista de tu cuenta.

B.1.3. Cuestionario System Usability Scale (SUS)

Tras completar las tareas guiadas, los participantes respondieron al cuestionario SUS, puntuando cada afirmación del 1 (*Totalmente en desacuerdo*) al 5 (*Totalmente de acuerdo*):

1. Creo que me gustaría usar esta aplicación con frecuencia.
2. He encontrado la aplicación innecesariamente compleja.
3. Pensé que la aplicación era fácil de usar.
4. Creo que necesitaría ayuda de alguien con conocimientos técnicos para poder usarla.
5. Encontré que las distintas funciones de la aplicación estaban bien integradas.
6. Pensé que había demasiada inconsistencia en la aplicación.
7. Me imagino que la mayoría de la gente aprendería a usar esta aplicación muy rápidamente.
8. Encontré la aplicación muy incómoda/complicada de usar.
9. Me sentí muy segura/o usando la aplicación.
10. Necesité aprender muchas cosas antes de poder manejarme con la aplicación.

B.1.4. Preguntas de cierre

1. ¿Qué es lo que más te ha gustado?
2. ¿Qué cambiarías o te ha resultado más incómodo?
3. ¿Echaste algo en falta?

B.1.5. Resultados

Participante	**Puntuación SUS**
EV1	100
EV2	32,5
EV3	90
EV4	77,5
EV5	87,5
Media	**77,5**

Tabla B.1: Puntuaciones SUS individuales – Fase 1

B.2. Fase 2: evaluación de exploración libre

En la segunda fase, tras una exploración libre de la aplicación completa, los participantes respondieron de nuevo al cuestionario SUS, ampliado con cuatro preguntas específicas sobre la percepción del sistema de recomendaciones, puntuadas también del 1 (*Totalmente en desacuerdo*) al 5 (*Totalmente de acuerdo*):

1. Las recomendaciones que me mostró la app eran relevantes para mis gustos.
2. Descubrí algo que no conocía gracias a las recomendaciones.
3. Confío en que las recomendaciones mejorarán cuanto más use la app.
4. Las recomendaciones se integran de forma natural en la aplicación.

También se incluyó una pregunta abierta de cierre: "Añade, si lo deseas, cualquier observación que te haya surgido de usar la app".

B.2.1. Resultados

Participante	**Puntuación SUS**
EV1	97,5
EV2	100
EV3	100
EV4	97,5
EV5	100
EV6	95
EV7	97,5
EV8	92,5
EV9	95
EV10	100
Media	**97,5**

Tabla B.2: Puntuaciones SUS individuales – Fase 2

Pregunta	**Media**
Relevancia para mis gustos	4,5
Descubrimiento de algo nuevo	4,5
Confianza en la mejora futura	4,7
Integración natural en la aplicación	4,7

Tabla B.3: Medias de las preguntas específicas sobre recomendaciones – Fase 2

¿Os parezco esquiva? Puedo serlo, a veces. También soy dulce y tranquilizadora, y disonante e iracunda, y difícil, y simple, apaciguadora como arena que se vierte, penetrante como el alfiler. Soy la Música...

Una música prodigiosa
Mitch Albom