eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Ingeniería de Software

Desarrollo de una Plataforma Web Interactiva de Datos Musicales Obtenidos de la API de Spotify

Jon Ortega Goikoetxea

Dirección
Miren Bermejo Llopis

18 de noviembre de 2024

Agradecimientos

En caso de querer añadir agradecimientos, escribir aquí el texto.

En caso de no querer este apartado, comentalo en el fichero *main.tex*.

Resumen

Escribe aquí el resumen.

Objetivos de Desarrollo Sostenible

Este proyecto se alinea con varios Objetivos de Desarrollo Sostenible (ODS) de las Naciones Unidas. En concreto, se toma como referencia el marco de la *EHUagenda 2030* de la UPV/EHU. Esta agenda, "recoge la contribución de la UPV/EHU a 12 de los 17 ODS de la Agenda 2030, al que ha sumado el su compromiso con la diversidad lingüística y cultural a través del ODS 17+1" [1]. A continuación, se detallan los objetivos específicos relacionados:

Figura 0.1: Los ODS alineados con este trabajo.

- **ODS 3: Salud y Bienestar**. La música juega un papel importante en el bienestar emocional y mental. La capacidad que ofrece esta aplicación de mejorar la experiencia musical y permitir a los usuarios conectar más profundamente con su música contibuye positivamente a su bienestar general.
- **ODS 9: Industria, Innovación e Infraestructura**. Como este proyecto implica la utilización de tecnologías modernas como React, Next.js y Vercel durante el desarrollo, se fomenta la innovación tecnológica y se contribuye al desarrollo de infraestructuras digitales eficientes.
- **ODS 18 (17+1): Garantizar la diversidad lingüística y cultural**. Al permitir que los usuarios exploren y aprecien música de diferentes culturas y en diversos idiomas, se facilita la exposición a estas, fomentando así el entendimiento y la apreciación cultural, alineándose así con el objetivo 18 propuesto por la UPV/EHU.

Índice de contenidos

Índice de contenidos IV

Índice de figuras VI

Índice de tablas VII

1 Introducción 1
- 1.1. Contexto . . . 1
- 1.2. Motivación . . . 2
- 1.3. Objetivos del Proyecto . . . 2
- 1.4. Estructura de la Memoria . . . 3

2 Contexto Competitivo 4

3 Planificación 6
- 3.1. Alcance . . . 6
  - 3.1.1. Funcionalidades Incluidas . . . 6
  - 3.1.2. Exclusiones . . . 6
  - 3.1.3. Limitaciones . . . 7
- 3.2. Gestión de Tareas . . . 7
  - 3.2.1. Descripción de Tareas . . . 7
  - 3.2.2. Dedicaciones . . . 11
  - 3.2.3. Dependencias entre Tareas . . . 12
  - 3.2.4. Periodos de Desarrollo . . . 13
  - 3.2.5. Hitos . . . 13
- 3.3. Gestión de Riesgos . . . 13
- 3.4. Gestión de Calidad . . . 13

4 Análisis 14
- 4.1. Estudio de la API de Spotify . . . 14
- 4.2. Requisitos Funcionales . . . 14
- 4.3. Requisitos No Funcionales . . . 14
- 4.4. Casos de Uso . . . 14

5 Diseño **15**
5.1. Arquitectura del Sistema . . . 15
5.2. Diagrama de Componentes de React . . . 15
5.3. Interfaz de Usuario . . . 15
5.4. Diagramas de Secuencia . . . 15
5.5. Seguridad . . . 15
5.6. Diseño de Pruebas . . . 15

6 Implementación **16**

7 Pruebas **17**

8 Despliegue **18**
8.1. Vercel . . . 18
8.2. CD/CI . . . 18

9 Conclusiones **19**

Apéndice **20**

Bibliografía **21**

Índice de figuras

2.1. Ejemplo de estadística de “oscuridad” de *Obscurify*. 5
2.2. Imagen generada por MusicScapes. 5

3.1. Diagrama EDT con los paquetes de trabajo del proyecto. 8
3.2. Diagrama de dependencias entre las tareas y paquetes de trabajo del proyecto. 12

Índice de tablas

2.1. Comparativa de funcionalidades ofrecidas por otros servicios afines. 4

3.1. Tabla con las estimaciones de tiempo por paquete de trabajo y tarea del proyecto. 11

Capítulo 1

Introducción

1.1. Contexto

El uso de datos personalizados es un componente esencial en el desarrollo de aplicaciones y servicios digitales actuales. Las plataformas buscan ofrecer experiencias ajustadas a las preferencias de los usuarios, utilizando grandes volúmenes de datos para generar contenido adaptado. En este contexto, *Spotify* se ha consolidado como una de las plataformas más destacadas de *streaming* musical, no solo por su extenso catálogo, sino también por su capacidad de ofrecer recomendaciones y estadísticas de uso personalizadas.

Una de las características más valoradas por los usuarios de *Spotify* es la posibilidad de acceder a estadísticas personales que les permiten comprender mejor sus hábitos de escucha. En 2016, como producto de una campaña de márketing viral, *Spotify* lanzó *Spotify Wrapped*, un resumen anual de los datos musicales de cada usuario, que generó un gran interés y participación en las redes sociales. Sin embargo, el acceso a estas estadísticas solo está disponible en el mes de diciembre, lo que genera una oportunidad para desarrollar herramientas complementarias que ofrezcan este tipo de análisis de manera más frecuente.

El acceso a estos datos es posible gracias a la *Web API* pública que *Spotify* pone a disposición de los desarrolladores. De esta manera, es posible obtener una amplia gama de datos sobre el comportamiento del usuario, como sus canciones más escuchadas, artistas favoritos y playlists. Además, la API ofrece acceso a muchos datos adicionales no utilizados en *Spotify Wrapped*, pero que, realizando diferentes combinaciones, permiten crear nuevas formas enriquecidas de análisis que presenten la información de manera más atractiva y personalizada.

También es necesario mencionar, que el crecimiento de las aplicaciones web interactivas ha sido posible gracias a tecnologías y frameworks modernos, los cuales permiten crear interfaces rápidas y eficientes, optimizando el rendimiento y la experiencia de usuario y simplificando el desarrollo. Además, los sistemas de integración y despliegue continuo (CI/CD) facilitan la entrega de aplicaciones de manera automatizada y escalable, garantizando que estén siempre actualizadas y accesibles para los usuarios.

1.2. Motivación

La elección de este Trabajo de Fin de Grado surge de un interés personal que he tenido desde el inicio de mi carrera universitaria. Desde el primer año, he tenido la idea base para implementar un servicio en torno a la *Web API* de *Spotify*. Sin embargo, en ese momento no contaba con los conocimientos necesarios para llevarlo a cabo. Ahora, tras completar la carrera, me siento con las capacidades necesarias para poder realizar dicho proyecto y materializar esta idea.

Como usuario habitual de *Spotify*, siempre me ha fascinado la función de *Spotify Wrapped*. No obstante, me resulta limitante que esta información solo esté disponible durante un breve periodo del año. Existen otras herramientas y páginas web que analizan datos de *Spotify*, pero suelen ofrecer estadísticas genéricas y demasiado básicas que carecen de profundidad. Estoy convencido de que es posible obtener estadísticas mucho más interesantes y, conversando con compañeros y otros usuarios, existe un interés general por acceder a estas estadísticas en cualquier momento, no solo una vez al año. La música que cada individuo escucha es algo personal y, a menudo, forma parte de su identidad. Por ello, considero que este proyecto no solo es interesante y motivador para mí, sino que también puede aportar valor a otros usuarios que comparten esta misma idea.

La posibilidad de crear un proyecto que me interese de principio a fin, y que yo mismo desearía utilizar, es una gran motivación. Además, la selección de tecnologías que he decidido emplear, como se detallará más adelante, no solo son ampliamente demandadas en el mercado actual, sino que también contribuyen a mi crecimiento profesional y enriquecen mis conocimientos.

1.3. Objetivos del Proyecto

Para empezar a caracterizar el proyecto de manera concreta, en este apartado se definen los objetivos que guiarán el desarrollo. El **objetivo general** de este proyecto es desarrollar una plataforma web interactiva que permita a los usuarios de *Spotify* acceder a las visualizaciones y análisis de sus datos musicales personales cuando lo deseen. Para alcanzar este objetivo general, se plantean los siguientes **objetivos específicos**:

- Analizar la API de *Spotify* para obtener los datos necesarios.
- Desarrollar la lógica necesaria para filtrar, organizar y transformar los datos obtenidos, preparándolos para su representación gráfica.
- Diseñar una interfaz de usuario intuitiva, interactiva y adaptable a diferentes dispositivos.
- Adoptar tecnologías modernas como *Next.js* y *TypeScript* para beneficiarse de las ventajas que ofrecen.
- Implementar prácticas de CI/CD usando la plataforma de *Vercel*.
- Implementar políticas de seguridad con respecto a los datos de usuarios establecidas por *Spotify* para desarrolladores.
- Documentar el proceso de desarrollo.

1.4. Estructura de la Memoria

La estructura de esta memoria refleja las diferentes etapas por las que atraviesa un proyecto de desarrollo software. Cada capítulo aborda aspectos clave que contribuyen al logro de los objetivos propuestos.

En la Introducción se ha establecido el contexto del proyecto, la motivación que impulsa su realización y los objetivos. Para terminar de definir el contexto en el que se está desarrollando, en el Contexto Competitivo se realizará un análisis de las soluciones ya existentes. A continuación, en la Planificación, se define el alcance del proyecto y se aborda lodo lo relacionado con la gestión de tareas, riesgos y calidad del proyecto. Además, se presentan las tecologías que se han utilizado durante todo el desarrollo.

El Análisis se centra en el estudio de la API de *Spotify*, el cual es **fundamental para el desarrollo del proyecto**. Se especifican los requisitos y se presentan los casos de uso que guiarán el desarrollo de la aplicación. En conjunto con el capítulo anterior, en el Diseño se describe la arquitectura del sistema y la estructura de la interfaz de usuario (UI), tanto a nivel técnico como visual. Además, se abordan aspectos de seguridad y se planifica el diseño de las pruebas.

La Implementación detalla el proceso de desarrollo de la aplicación, incluyendo las decisiones tomadas durante esta fase. Se describen los retos enfrentados y cómo se han resuelto. Seguido, en el capítulo de las Pruebas se exponen las pruebas realizadas y se analizan los resultados obtenidos. Como consecuencia lógica, en Despliegue se explica el proceso de puesta en producción de la aplicación y las estrategias de CI/CD implementadas.

Finalmente, en Conclusiones se hace una reflexión sobre los resultados alcanzados, evaluando el cumplimiento de los objetivos iniciales, el seguimiento en comparación a la planificación inicial y se discuten las lecciones aprendidas y se proponen líneas futuras de trabajo.

Mediante este flujo, se espera que el público lector no tenga problemas para seguir la línea de pensamiento que se ha llevado a cabo y que cada apartado quede aclarado o, en menor medida, justificado por los capítulos anteriores.

Capítulo 2

Contexto Competitivo

Antes de proceder con la planificación del proyecto, es un buen ejercicio analizar las alternativas similares que actualmente existen en el mercado. Como se menciona en la sección de Contexto, la mayoría de otras webs ofrecen estadísticas muy básicas, como visualizar los *top géneros*, *artistas* y *álbumes* del usuario, además de su *historial de reproducción* más reciente. Cada una de estas funcionalidades corresponde a llamadas directas a la *Web API* de *Spotify*, siendo estadísticas triviales de implementar, con una mínima necesidad de procesamiento de datos.

Estadísticas	stats.fm	Obscurify	Stats for Spotify	Replayify	Zodiac Affinity	Music Scapes
Básicas	Top canciones Top artistas Top géneros Historial reciente	Top canciones Top artistas Top géneros	Top canciones Top artistas Top géneros Historial reciente	Top canciones Top artistas Historial reciente		
Avanzadas	Tiempo escuchado Número de artistas Número de álbumes	Porcentaje por décadas				
Creativas	Escuchas por hora del día	Rating de "obscurity" Top artistas "oscuros" Top canciones "oscuras" Tus estados de ánimo			5 canciones según tu signo zodiacal	Imagen basada en las propiedades de las canciones que más escuchas

Tabla 2.1: Comparativa de funcionalidades ofrecidas por otros servicios afines.

Una de las opciones más populares es la aplicación web *stats.fm*[1] (anteriormente *Spotistats*). Esta ofrece las funcionalidades básicas mencionadas a todos los usuarios, pero, mediante un plan de pago, se habilitan gráficas más avanzadas. Estas gráficas no se obtienen directamente de la API, sino que requieren que el usuario descargue manualmente los datos históricos guardados por *Spotify* y los suba a la página web como un archivo comprimido.

[1] stats.fm: https://spotistats.app/

De esta manera, se generan estadísticas relativamente más complejas, pero que aún carecen de "creatividad" en su presentación.

Otro servicio notable es *Obscurify*[2], que se centra en la "oscuridad" o "rareza" de la música que el usuario escucha. El concepto principal de *Obscurify* es identificar las canciones y artistas que son menos populares entre otros usuarios, clasificándolos como más "oscuros". Esta funcionalidad permite que el usuario se sienta especial al escuchar música que no es común. Sin embargo, su enfoque está limitado en su mayoría a este concepto, lo que deja fuera otras formas de visualización o análisis más amplios y variados.

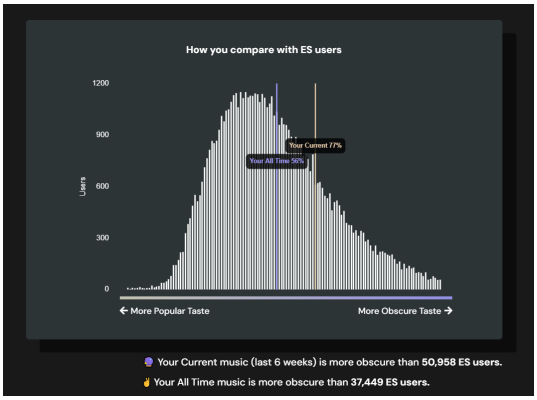

Figura 2.1: Ejemplo de estadística de "oscuridad" de *Obscurify*.

Además de estas páginas principales, existen otras que se enfocan toda su funcionalidad en una única característica original. Dos ejemplos destacados son *Zodiac Affinity*[3] y *MusicScapes*[4]. La primera genera una recomendación de cinco canciones en base a los hábitos de escucha del usuario y su signo zodiacal, mientras que la segunda crea de manera procedural una imagen basada en diferentes propiedades de las canciones que el usuario ha escuchado recientemente. Estas páginas no ofrecen ninguna funcionalidad adicional, limitándose a esa única característica. Aunque hay otros servicios similares, he elegido estos dos ejemplos por su aspecto más "acabado", ya que muchas otras alternativas se presentan más como una prueba de concepto o una demo, en lugar de páginas completamente desarrolladas.

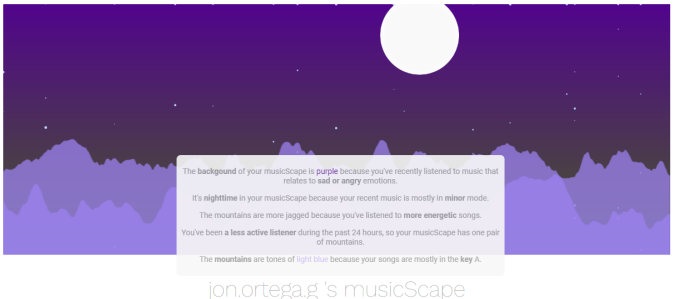

Figura 2.2: Imagen generada por MusicScapes.

[2]Obscurify: https://www.obscurifymusic.com/
[3]Zodiac Affinity: https://zodiacaffinity.eu/
[4]MusicScapes: https://musicscapes.herokuapp.com/

Capítulo 3

Planificación

3.1. Alcance

Definir con precisión el alcance es fundamental para asegurar que el desarrollo se ajuste a los objetivos propuestos y se realice dentro de los recursos y tiempos establecidos. Para ello, en esta sección se delimitarán las funcionalidades, exclusiones y limitaciones que se esperan en este proyecto.

3.1.1. Funcionalidades Incluidas

En la plataforma web se ofrecen las siguientes funcionalidades principales:

- Autenticación segura mediante las credenciales de *Spotify*.
- *Home* o Panel Inicial donde se muestran la información básica de la cuenta.
- Análisis detallado y visualizaciones gráficas avanzadas, interactivas y actualizadas de sus datos musicales.
- Interfaz adaptativa, intuitiva y responsiva.
- Cierre de sesión seguro.

3.1.2. Exclusiones

Para establecer expectativas claras sobre el alcance del proyecto, se detallan a continuación las funcionalidades que **no** serán incluidas en la plataforma web:

- No se desarrollarán aplicaciones nativas de otras plataformas como móvil, PC, Mac o Linux; el acceso será exclusivamente a través de la web.
- Aunque se seguirá un diseño intuitivo, no se implementarán funcionalidades específicas de accesibilidad avanzadas como compatibilidad con lectores de pantalla o navegación por teclado.
- La plataforma se enfoca exclusivamente en la integración con *Spotify*; se excluyen todos los otros servicios de streaming como *Apple Music*, *Deezer*, etc.

- No se almacenarán de forma persistente datos personales del usuario en servidores propios más allá de lo necesario para la sesión actual; todos los datos se obtendrán directamente de la API de *Spotify* y se manejarán en tiempo real.
- Se excluye el desarrollo de funcionalidades relacionadas con la interacción social (envío de mensajes, compartir estadísticas, rankings entre usuarios, etc.) dentro o a través de la plataforma, ya que superarían el alcance recogido dentro de un TFG.

3.1.3. Limitaciones

Durante el desarrollo del proyecto, se han identificado las siguientes limitaciones que han afectado al alcance y a las funcionalidades de la web:

- Las políticas de seguridad de *Spotify* impiden el almacenamiento persistente de datos personales, limitando funcionalidades que requieran conservar información del usuario entre sesiones.
- El procesamiento de los datos se ve limitado por los recursos computacionales que la nube de *Vercel* ofrece, descartando técnicas avanzadas como el aprendizaje automático.
- El tiempo y los recursos disponibles para el desarrollo del proyecto son finitos, lo que ha obligado a priorizar funcionalidades esenciales y descartar características adicionales.
- Al hacer uso de una API de terceros, todas las funcionalidades necesitan una conexión activa a Internet para poder funcionar de manera correcta.

3.2. Gestión de Tareas

Una vez definido el alcance, es necesario detallar las tareas requeridas para el desarrollo del proyecto. En esta sección se caracterizará todo lo necesario en relación a las tareas, incluyendo su definición, relaciones y tiempos asignados, para asegurar una gestión estructurada y alineada con los objetivos del proyecto.

3.2.1. Descripción de Tareas

Para gestionar de manera efectiva el conjunto de actividades, se ha elaborado una Estructura de Desglose de Trabajo (EDT). Esta EDT (figura 3.1) proporciona una visión general de las principales áreas de trabajo, desglosando el proyecto en paquetes específicos que abarcan cada tarea esencial, facilitando así la gestión.

En este caso, el proyecto se organiza en cinco áreas principales, que abarcan todas las fases del desarrollo de la aplicación; abordando tanto las tareas relacionadas con la creación de la aplicación en sí misma (el producto final) como aquellas enfocadas en la gestión y redacción del proyecto para su documentación. Esta estructura garantiza una distribución clara de las tareas, cubriendo tanto los aspectos técnicos como los organizativos.

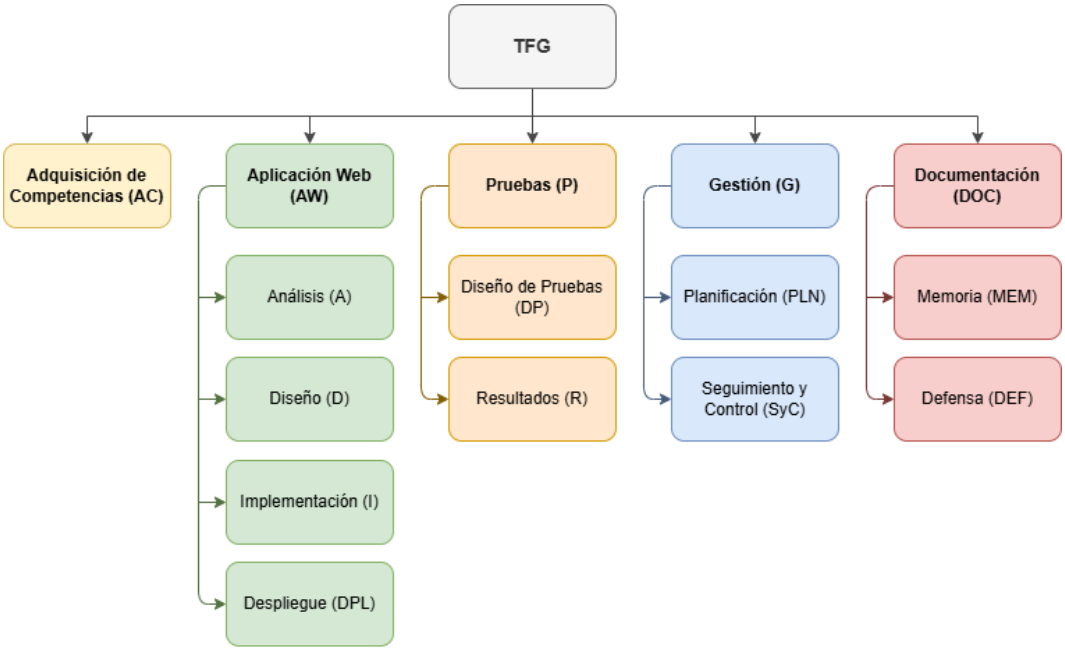

Figura 3.1: Diagrama EDT con los paquetes de trabajo del proyecto.

A continuación se detalla cada paquete de trabajo y las tareas correspondientes contenidas en cada una:

3.2.1.1. Adquisición de Competencias (AC):

Este paquete de trabajo incluye todas las tareas necesarias para adquirir el conocimiento sobre las tecnologías y herramientas clave para el desarrollo del proyecto.

- **AC.1:** Aprender *TypeScript*, *React.js* y *Next.js* para el desarrollo de la aplicación web.
- **AC.2:** Estudiar el uso de *Vercel* para el hosting y despliegue de la aplicación.
- **AC.3:** Hacer un reconocimiento inicial de la *Web API* de *Spotify*.

3.2.1.2. Aplicación Web (AW):

En este paquete se engloban todas las fases de desarrollo de la aplicación web, desde la planificación inicial hasta el despliegue final.

- **Análisis (A):**
  - **AW.A.1:** Estudiar y analizar en profundidad la *Web API* de *Spotify* para determinar el alcance y sus limitaciones.
  - **AW.A.2:** Definir los requisitos funcionales y no funcionales del sistema.
  - **AW.A.3:** Desarrollar los principales casos de uso del sistema.
- **Diseño (D):**

- **AW.D.1:** Diseñar la arquitectura del sistema.
- **AW.D.2:** Crear un diagrama de componentes React para establecer la jerarquía y realizar un diseño general de la interfaz de usuario.
- **AW.D.3:** Definir los diagramas de secuencia de los casos principales.
- **AW.D.4:** Realizar una gestión de la seguridad y asegurar que se implementarán las medidas definidas por *Spotify* para el uso de la API.

- **Implementación (I):**
  - **AW.I.1:** Implementar el login de la página web, usando el protocolo OAuth 2.0 implementado por *Spotify*.
  - **AW.I.2:** Implementar el panel inicial (dashboard) de la web.
  - **AW.I.3:** Implementar la sección principal de estadísiticas.
  - **AW.I.4:** Implementar la funcionalidad de cerrar sesión.
  - **AW.I.5:** Realizar optimizaciones y correcciones en la implementación.
- **Despliegue (DPL):**
  - **AW.DPL.1:** Configurar el despliegue en *Vercel* para crear un proceso automático de despliegue.
  - **AW.DPL.2:** Monitorear el funcionamiento del despliegue y los logs.

3.2.1.3. Pruebas (P):

Este paquete agrupa las tareas relacionadas con la verificación y validación de la aplicación, garantizando su correcto funcionamiento y calidad.

- **Diseño de Pruebas (DP):** Planificar y diseñar pruebas unitarias, de integración y de carga para evaluar el rendimiento y la estabilidad de la aplicación.
- **Resultados (R):**
  - **P.R.1:** Realizar las pruebas planificadas y documentar los errores encontrados.
  - **P.R.2:** Definir y, en caso de que sea posible, implementar las correcciones necesarias.

3.2.1.4. Gestión (G):

- **Planificación (PLN):**
  - **G.PLN.1:** Realizar una primera estimación de tiempos de las tareas generales.
  - **G.PLN.2:** Establecer el alcance inicial del proyecto según las características del producto seleccionadas.
  - **G.PLN.3:** Definir la planificación del proyecto.
  - **G.PLN.4:** Revisar y, si fuera necesario, modificar la planificación.
- **Seguimiento y Control (SyC):**

- **G.SyC.1:** Conversaciones y comentarios de la tutora a lo largo del desarrollo.
- **G.SyC.2:** Elaboración de un documento para registrar las actividades y dedicaciones realizadas a lo largo del proyecto.
- **G.SyC.3:** Comparación de los datos del seguimiento con los de la placificación, identificación de las desviaciones y riesgos significativos.

3.2.1.5. Documentación (DOC):

Este paquete agrupa las tareas necesarias para la elaboración de la memoria y la preparación de la defensa del proyecto.

- **Memoria (MEM):**
  - **DOC.MEM.1:** Preparar el entorno de trabajo en LaTeX utilizando *Visual Studio Code* y establecer la estructura básica de la memoria a partir de la plantilla proporcionada por la facultad.
  - **DOC.MEM.2:** Redactar la memoria.
- **Defensa (DEF):**
  - **DOC.DEF.1:** Identificar los puntos y conceptos clave que se presentarán en la defensa.
  - **DOC.DEF.2:** Crear los elementos visuales de apoyo para la defensa.
  - **DOC.DEF.3:** Preparar y ensayar la defensa.

3.2.2. Dedicaciones

A continuación, en la tabla 3.1, se presentan las horas estimadas para las tareas descritas en el apartado anterior. También se muestran las sumas de las dedicaciones por paquete de trabajo y la suma total de horas que se espera que lleve el desarrollo del proyecto completo.

	Tarea	Horas
	Adquisición de Competencias (AC)	**24**
	AC.1: TS, React, Next.js	18
	AC.2: Vercel	2
	AC.3: Spotify API	4
Aplicación Web (AW)	**Análisis (A)**	**30**
	AW.A.1: Estudio de Spotify API	16
	AW.A.2: Captura de requisitos	8
	AW.A.3: Casos de uso	6
	Diseño (D)	**37**
	AW.D.1: Aquitectura	5
	AW.D.2: Interfaz de usuario	10
	AW.D.3: Lógica de negocio	18
	AW.D.4: Seguridad	4
	Implementación (I)	**80**
	AW.I.1: Login	7
	AW.I.2: Home	15
	AW.I.3: Estadísticas	40
	AW.I.4: Logout	2
	AW.I.5: Optimizaciones	16
	Despliegue (DPL)	**3**
	AW.D.1: Configurar CI/CD	2
	AW.D.2: Monitorear logs	1
	Pruebas (P)	**30**
	P.DP: Diseñar pruebas	18
	P.R: Resultados	12
	Gestión (G)	**20**
	G.PLN: Planificación	10
	G.SyC: Seguimiento y Control	10
	Documentación (DOC)	**90**
	DOC.MEM: Memoria	80
	DOC.DEF: Defensa	10
	TOTAL	**314**

Tabla 3.1: Tabla con las estimaciones de tiempo por paquete de trabajo y tarea del proyecto.

3.2.3. Dependencias entre Tareas

En la figura 3.2 se muestra un diagrama representando las dependencias que existen entre las diferentes tareas. De esta manera, se puede apreciar de forma visual las tareas que requieren la finalización de una o varias tareas para su comiezo.

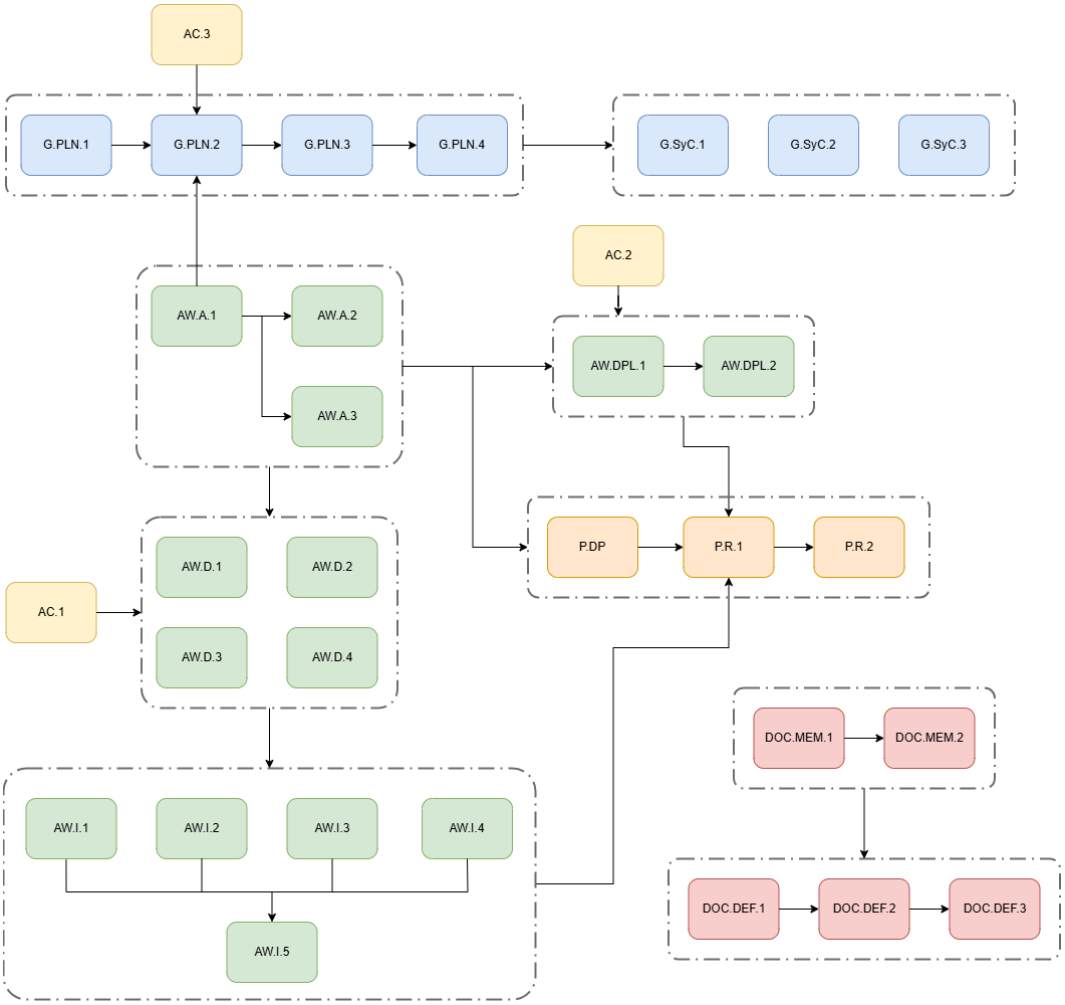

Figura 3.2: Diagrama de dependencias entre las tareas y paquetes de trabajo del proyecto.

Como se puede apreciar, el proyecto debe iniciarse con las tareas de la planificación (paquete de trabajo **G.PLN**) y análisis (**AW.A**), al igual que el desarrollo de las tareas relacionadas con la memoria (**DOC.MEM**), que se realizan desde el inicio del proyecto hasta casi la finalización del TFG. Para ordenar temporalmente todas las tareas, en la siguiente sección se tratarán los periodos de desarrollo de cada una.

3.2.4. Periodos de Desarrollo

3.2.5. Hitos

3.3. Gestión de Riesgos

3.4. Gestión de Calidad

Capítulo 4

Análisis

4.1. Estudio de la API de Spotify

4.2. Requisitos Funcionales

4.3. Requisitos No Funcionales

4.4. Casos de Uso

Capítulo 5

Diseño

5.1. Arquitectura del Sistema

5.2. Diagrama de Componentes de React

5.3. Interfaz de Usuario

5.4. Diagramas de Secuencia

5.5. Seguridad

5.6. Diseño de Pruebas

Capítulo 6

Implementación

Capítulo 7

Pruebas

Capítulo 8

Despliegue

8.1. Vercel

8.2. CD/CI

Capítulo 9

Conclusiones

Cita falsa [1]

Apéndice

Apéndices

Bibliografía

[1] A. Falso, “Título ficticio para pruebas,” 2024, cita ficticia para evitar errores de compilación en LaTeX. Ver página 19.