YGGDRASOUND

YGGSO

Atteneri de Jesús Torres González

Nombre del fichero:	DAW_PRW_YGGSO_UT01 Plan de proyecto
Fecha de esta versión:	20/12/2025

Historial de revisiones

Fecha	Descripción	Autor
20/12/2025	Creación del documento “Plan de Proyecto”	Atteneri de Jesús Torres González

ÍNDICE

1 Resumen ejecutivo..... 4
2 Alcance del proyecto..... 4
3 Requisitos del proyecto..... 5
4 Estructura de trabajo y cronograma..... 7
5 Recursos y roles del equipo..... 8
6 Gestión de riesgos.....9
7 Conclusión..... 10

1 Resumen ejecutivo

Yggdrasound es una plataforma web de comercio electrónico especializada en la venta de vinilos, diseñada para coleccionistas que buscan una experiencia unificada. La aplicación integra un catálogo extenso con gestión de compras (carrito, historial, deseos) y una función única que muestra, directamente en la ficha de cada artista, información en tiempo real sobre sus próximos conciertos.

El propósito de ***Yggdrasound*** es centralizar y unificar la experiencia del coleccionista de vinilos, eliminando la fragmentación actual al integrar en una sola plataforma la exploración del catálogo, la gestión de compra, la comunidad y la búsqueda de conciertos de los artistas, facilitando así el descubrimiento y la decisión de compra.

2 Alcance del proyecto

El objetivo general es resolver la fragmentación en la experiencia de compra de vinilos online, ofreciendo una plataforma centralizada donde los coleccionistas puedan explorar un catálogo, gestionar sus compras y, además, descubrir al instante si los artistas tienen conciertos programados.

Los objetivos específicos son:

- Permitir el registro y autenticación de usuarios.
- Gestionar un catálogo de productos con imágenes, precios y listas de canciones.
- Implementar un carrito de compra, lista de deseos e historial de pedidos.
- Integrar un sistema de comentarios por producto.
- Desarrollar la funcionalidad “Buscar Shows” que consulte en Bandcamp la disponibilidad de conciertos del artista.

Los límites del desarrollo serán:

- No se desarrollará una pasarela de pago real (se simulará el proceso de confirmación de pedido).
- No se realizará scraping masivo o continuo. La búsqueda de eventos será una consulta puntual y bajo demanda.

3 Requisitos del proyecto

Los requisitos funcionales serán:

- **Gestión de Usuarios**
  - El sistema debe permitir el registro de nuevos usuarios (email, nombre, contraseña).
  - El sistema debe permitir el inicio y cierre de sesión de usuarios registrados.
  - Los usuarios autenticados deben poder editar su perfil básico.
- **Gestión del Catálogo**
  - El sistema debe mostrar una lista paginada y filtrable de todos los vinilos.
  - El sistema debe mostrar una página de detalle para cada vinilo con: artista, álbum, imagen, precio, localidad del artista, lista de canciones y sección de comentarios.
  - (Admin) El sistema debe permitir a un administrador realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los productos del catálogo.
- **Funciones de Compra (Simuladas)**
  - Los usuarios autenticados deben poder añadir/eliminar vinilos de un carrito de compra.
  - Los usuarios autenticados deben poder añadir/eliminar vinilos de una lista de deseos.
  - El sistema debe simular un proceso de checkout, generando un pedido y registrándose en el historial de pedidos del usuario.
- **Sistema de Comentarios**
  - Los usuarios autenticados deben poder publicar un comentario en la página de detalle de un vinilo.
  - El sistema debe mostrar los comentarios ordenados por fecha (más recientes primero).
- **Búsqueda de Conciertos**
  - En la página de detalle de un vinilo, el sistema debe ofrecer un botón/acción "Buscar Shows".
  - Al activarlo, el sistema debe consultar (scraping) la página de Bandcamp del artista y mostrar claramente al usuario si hay conciertos disponibles o no.

Los requisitos no funcionales serán:

- **Usabilidad**
  - La interfaz debe ser intuitiva y responsive, funcionando correctamente en dispositivos de escritorio y móviles.
- **Rendimiento**
  - La consulta de scraping ("Buscar Shows") debe tener un tiempo de espera máximo de 5 segundos antes de dar una respuesta (éxito o error).
- **Seguridad**
  - Las contraseñas de usuario deben almacenarse de forma encriptada en la base de datos.
- **Fiabilidad**
  - El sistema debe manejar adecuadamente los errores (ej: artista no encontrado en Bandcamp, timeout) en la función "Buscar Conciertos", mostrando un mensaje claro al usuario sin caerse.

Los requisitos técnicos serán:

- **Arquitectura**: Aplicación web basada en una arquitectura cliente-servidor (frontend, backend, base de datos).
- **Frontend**: Desarrollado con React.js para una interfaz dinámica y de una sola página (SPA).
- **Backend**: Desarrollado con Node.js y Express.js para crear la API RESTful.
- **Base de Datos:** PostgreSQL en un contenedor Docker para el entorno de desarrollo, haciendo uso de Sequelize como ORM para el almacenamiento relacional.
- **Scraping**: Uso de la biblioteca como Cheerio para la obtención de datos de conciertos desde Bandcamp.
- **Despliegue**: La aplicación será desplegada en un servicio en la nube como Render, Railway o Vercel.
- **Control de versiones:** El código se gestionará con Git y se alojará en un repositorio en GitHub.

4 Estructura de trabajo y cronograma

División en fases o tareas principales:

Fase 1: Planificación, Investigación y Preparación de Datos

- Investigación del código fuente de Bandcamp para identificar selectores HTML y estructuras de datos para el scraping.
- Desarrollo y ejecución de scripts de scraping (Python con beautifulsoup) para obtener los datos iniciales de los vinilos.
- Creación de scripts de limpieza y transformación de datos para adaptarlos al esquema de la base de datos PostgreSQL.
- Diseño de la base de datos (modelo ER).

Fase 2: Configuración y Desarrollo del Backend

- Configuración del proyecto Node.js/Express.
- Configuración de PostgreSQL con Sequelize y Docker.
- Implementación de modelos y migraciones basadas en el diseño ER.
- Carga inicial de los datos limpios en la base de datos.
- Desarrollo de la API REST.
- Implementación del endpoint del módulo "Buscar Shows" (scraping en tiempo real de Bandcamp con Cheerio).

Fase 3: Desarrollo del Frontend

- Configuración del proyecto React.
- Implementación de rutas y layout principal.
- Desarrollo de páginas: catálogo (con filtros), detalle de producto (con sección de comentarios y botón "Buscar Shows"), carrito, lista de deseos, perfil e historial.
- Integración con la API (Axios) para todas las funcionalidades.
- Estilizado (Tailwind) para una UI responsive.

Fase 4: Integración, Pruebas, Despliegue y Documentación

- Integración completa frontend-backend.
- Pruebas funcionales de todos los flujos de usuario (registro, compra, comentarios, scraping).
- Pruebas de rendimiento y corrección de errores.
- Despliegue en la nube (Render, Railway, etc.) de la aplicación completa.
- Pruebas finales en el entorno de producción.

Las fechas estimadas serán las siguientes:

- Semana 1-2: Fase 1 – Planificación, Investigación y Preparación de Datos
- Semana 3-4: Fase 2 – Configuración y Desarrollo del Backend
- Semana 5-6: Fase 3 – Desarrollo del Frontend
- Semana 7: Fase 4 – Integración y Pruebas
- Semana 8: Fase 4 – Despliegue, Pruebas Finales y Documentación

Herramienta o formato de planificación:

Al ser un proyecto individual, voy a seguir las fases en el orden que hemos definido, completando cada una antes de pasar a la siguiente. Iré controlando mi avance semana a semana, comprobando que voy cumpliendo con los plazos marcados para cada etapa usando kanban del proyecto github.

5 Recursos y roles del equipo

El personal asignado consta solo de mi misma como desarrolladora Full-Stack y seré responsable de todas las fases del proyecto: análisis, diseño, desarrollo backend y frontend, gestión de la base de datos, implementación del scraping, pruebas, despliegue y documentación.

Recursos técnicos y materiales necesarios:

Hardware/Software:

- Ordenador de desarrollo con conexión a internet.
- Software y herramientas de Desarrollo:
- Editor de código (VSCode).
- React.js para el frontend.
- Cliente API para peticiones HTTP (Nuncius o Insomnia).
- Node.js y npm para el entorno de ejecución.
- PostgreSQL y Docker para la base de datos.
- Git y GitHub para control de versiones.
- Python para el scraping inicial de datos (con beautifulsoup) y para limpieza y transformación de los datos.

Servicios Externos:

- Plataforma de despliegue en la nube (Render, Railway o Vercel).
- Acceso a Bandcamp para la obtención de datos de productos y conciertos (scraping ético con fines educativos).

6 Gestión de riesgos

Los posibles problemas o retrasos podrían ser:

- Cambios en la estructura de Bandcamp: El código fuente de las páginas scrapeadas (tanto para el catálogo inicial como para la búsqueda de conciertos) podría modificarse, rompiendo los selectores HTML y causando fallos en la obtención de datos.
- Problemas de despliegue: Dificultades al configurar el entorno en la nube o al conectar todos los servicios (backend, frontend, base de datos) pueden retrasar la puesta en producción.

Plan de contingencia o prevención:

Para cambios en el scraping:

- **Prevención**: Diseñar los scripts de scraping con selectores HTML lo más robustos y genéricos posible.
- **Contingencia**: Mantener un script de respaldo que permita re-ejecutar el proceso de scraping, limpieza y carga de datos en la base de datos de forma semiautomática, minimizando el tiempo de corrección.

Para retrasos técnicos:

- **Prevención**: Priorizar el desarrollo de una versión mínima viable (MVP) de cada funcionalidad en primer lugar.

7 Conclusión

Yggdrasound es un proyecto realista y completo que encaja perfectamente como trabajo final de DAW. Es viable porque cubro todas las tecnologías que necesito dominar (base de datos, backend y frontend).

Lo más valioso es que no es solo un ejercicio técnico: simula una web real con una función útil (avisar de conciertos) que le da personalidad. Al terminarlo, tendré un proyecto entero que enseñar, desde la idea hasta la web publicada, demostrando que puedo resolver problemas de desarrollo de principio a fin.