

DOCUMENTO DE ALCANCE

YGGDRASOUND

YGGSO

Atteneri de Jesús Torres González

Nombre del fichero:	DAW_PRW_YGGSO_UT01.1. Documento de alcance.odt
Fecha de esta versión:	21/12/2025

Historial de revisiones

Fecha	Descripción	Autor
20/12/2025	Creación del documento de alcance	Atteneri de Jesús Torres González
21/12/2025	Modificación del rol de administrador tras realizar el diagrama de casos de uso	Atteneri de Jesús Torres González

ÍNDICE

1 INTRODUCCIÓN.....	4
2 REQUISITOS DE USUARIO.....	4
2.1 Necesidad del negocio.....	4
2.2 Objetivos del negocio y del proyecto.....	4
2.3 Perfil de usuarios potenciales.....	5
3 DESCRIPCIÓN DEL SISTEMA.....	5
4 REQUISITOS PRINCIPALES.....	7
5 CRITERIOS DE ACEPTACIÓN Y PLANIFICACIÓN.....	9
5.1 Tabla de Gantt.....	10

1 INTRODUCCIÓN

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.

2 REQUISITOS DE USUARIO

2.1 Necesidad del negocio

Yggdrasound surge para responder a una necesidad real en el mercado del coleccionismo musical: la experiencia de compra fragmentada. Actualmente, los aficionados deben saltar entre múltiples webs (tiendas, redes, páginas de conciertos, etc) para completar una decisión, lo que ralentiza el proceso y dificulta el descubrimiento.

El propósito de ***Yggdrasound*** es centralizar y unificar esa experiencia, integrando en una sola plataforma la exploración del catálogo, la gestión de compra, la comunidad y la búsqueda de conciertos.

2.2 Objetivos del negocio y del proyecto

El objetivo principal es desarrollar e implementar la plataforma web ***Yggdrasound***, una tienda online de vinilos que integre la gestión de un catálogo extenso con un sistema de descubrimiento de conciertos en vivo, ofreciendo una experiencia de compra unificada y de valor añadido para el coleccionista.

Objetivos Específicos

- Diseñar e implementar una base de datos relacional que almacene y gestione el catálogo de vinilos, usuarios, pedidos, listas de deseos y comentarios.
- Desarrollar un backend con API RESTful que gestione la lógica de negocio, la autenticación de usuarios y el proceso de compra simulado.
- Crear un frontend web responsive que permita a los usuarios navegar por el catálogo, gestionar su perfil, interactuar con la comunidad y utilizar la función "Buscar Conciertos".

- Implementar el módulo de scraping ("Buscar Conciertos") que consulte la página de Bandcamp del artista y muestre la disponibilidad de conciertos próximos.
- Desplegar la aplicación completa en un servicio en la nube, garantizando su accesibilidad pública y funcionamiento integrado.

2.3 Perfil de usuarios potenciales

Dadas las características del sistema de tienda de vinilos, se requieren los siguientes roles de usuarios que acceden a la plataforma:

- **Invitado:** Representa a cualquier visitante que navega por la tienda sin identificarse. Puede explorar el catálogo y ver detalles de productos y comentarios, pero no puede realizar acciones que requieran identificación personal.
- **Cliente / Usuario Registrado:** Persona que se ha registrado en la plataforma para gestionar su perfil, realizar compras (añadir al carrito o checkout simulado), consultar su historial, publicar comentarios y utilizar funciones adicionales como la búsqueda de conciertos. Requiere autenticación en el sistema.
- **Administrador:** Posee acceso total, incluyendo un panel de gestión para administrar el catálogo de productos y moderar los comentarios de los usuarios. Requiere autenticación en el sistema.

3 DESCRIPCIÓN DEL SISTEMA

Yggdrasound es una aplicación web de comercio electrónico completa que permite a los usuarios explorar, gestionar y comprar vinilos dentro de una experiencia integrada. El sistema se alimenta de una base de datos con un catálogo inicial extenso y ofrece, además de las funciones básicas de una tienda online, un módulo especial que muestra los conciertos del artista.

Este sistema requerirá al menos de 3 puntos de acceso o roles principales. A continuación se clasifican y describen las operaciones necesarias:

1. Front-end Cliente / Usuario Registrado

1. Iniciar sesión.
2. Explorar y visualizar el catálogo completo de vinilos.

3. Ver la ficha detallada de un vinilo (imagen, artista, álbum, precio, localidad y tracklist).
4. Gestionar su carrito de compra (añadir/eliminar productos).
5. Gestionar su lista de deseos.
6. Completar el proceso de checkout simulado para generar un pedido.
7. Consultar su historial de pedidos.
8. Publicar y leer comentarios en las fichas de los vinilos.
9. Utilizar la función "Buscar Conciertos" de un artista desde la ficha del vinilo.

2. Front-end Invitado / Usuario No Registrado

1. Explorar y visualizar el catálogo completo de vinilos.
2. Ver la ficha detallada de un vinilo.
3. Leer los comentarios publicados por otros usuarios.
4. Registrarse en el sistema para obtener un perfil de cliente.

3. Front-end Administrador

1. Panel de Usuarios

1. Visualizar el listado de usuarios.
2. Crear, modificar y eliminar usuario.

2. Panel de administración de Productos (Vinilos)

1. Visualizar el listado completo de productos.
2. Crear, modificar y eliminar un producto del catálogo.

3. Panel de administración de Comentarios

1. Visualizar todos los comentarios publicados.
2. Moderar o eliminar comentarios inapropiados.

4. Panel de administración de Pedidos

1. Visualizar el historial de pedidos de todos los clientes.

4. Sistema / Automatización (Backend)

1. Ejecutar, de forma programada o bajo demanda, el scraping inicial para poblar la base de datos.
2. Ejecutar, bajo demanda del cliente, la consulta en tiempo real a Bandcamp para la función "Buscar Conciertos".
3. Gestionar la conexión y sincronización entre el frontend, la API y la base de datos.

4 REQUISITOS PRINCIPALES

Los requisitos funcionales serán:

- **Gestión de Usuarios**
 - El sistema debe permitir el registro de nuevos usuarios (email, nombre y 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.

5 CRITERIOS DE ACEPTACIÓN Y PLANIFICACIÓN

El proyecto se considerará dentro del alcance cuando se cumplan las siguientes condiciones:

Catálogo Funcional:

- Se muestran todos los vinilos iniciales en una interfaz paginada o con scroll infinito.
- Cada vinilo muestra: imagen, artista, álbum, precio, localidad y lista de canciones.
- Los filtros básicos (por género, artista, precio) funcionan correctamente.

Gestión de Usuario y Compra Simulada:

- Un usuario puede registrarse, iniciar sesión y editar su perfil.
- Un usuario autenticado puede añadir/eliminar vinilos al carrito y a la lista de deseos.
- El proceso de checkout simulado genera un pedido registrado en el historial del usuario.

Sistema de Comentarios:

- Los usuarios autenticados pueden publicar un comentario en la ficha de un vinilo.
- Los comentarios se muestran públicamente ordenados por fecha (más reciente primero).

Función "Buscar Conciertos":

- En la ficha de un artista, existe un botón o sección "Buscar Conciertos".

- Al activarlo, el sistema consulta Bandcamp y muestra un mensaje claro indicando si hay o no hay conciertos disponibles (o un error controlado si falla la consulta).

Despliegue y Acceso:

- La aplicación está desplegada en la nube y es públicamente accesible mediante una URL.
- El frontend, el backend y la base de datos están conectados y funcionan en conjunto.

Responsive y Estabilidad Básica:

- La interfaz se visualiza y es usable correctamente en dispositivos móviles y de escritorio.

5.1 Tabla de Gantt

Id. Tarea	Tarea de Desarrollo	Fecha Inicio	Fecha Fin	Día Inicio	Día Fin	Duración
1	1.1. Análisis técnico del desarrollo (ficha técnica).	21/12/2025	21/12/2025	1	1	1
2	1.2. Investigación y script de scraping (Bandcamp).	24/12/2025	26/12/2025	3	5	3
3	1.3. Limpieza y transformación de datos para la BBDD.	29/12/2025	30/12/2025	6	7	2
4	1.4. Diseño del modelo de datos (ER) y wireframes.	31/12/2025	02/01/2026	8	10	3
5	2.1. Configuración del proyecto Node.js/ Express y BD PostgreSQL (Docker).	05/01/2026	06/01/2026	11	12	2
6	2.2. Implementación de modelos Sequelize y migraciones.	07/01/2026	08/01/2026	13	14	2
7	2.3. Carga inicial de datos en la base de datos.	09/01/2026	09/01/2026	15	15	1
8	2.4. Desarrollo de la API REST.	12/01/2026	16/01/2026	16	20	5
9	2.5. Implementación de autenticación.	19/01/2026	20/01/2026	21	22	2
10	2.6. Desarrollo del endpoint "Buscar Conciertos" (scraping con Cheerio).	21/01/2026	23/01/2026	23	25	3

11	3.1. Configuración, proyecto React y estructura de rutas.	26/01/2026	27/01/2026	26	27	2
12	3.2. Desarrollo de componentes: Layout, Catálogo, Filtros.	28/01/2026	30/01/2026	28	30	3
13	3.3. Desarrollo de páginas: Detalle de producto, Carrito, Perfil.	02/02/2026	05/02/2026	31	34	4
14	3.4. Integración con API (Axios) para todos los componentes.	06/02/2026	10/02/2026	35	39	5
15	3.5. Estilizado con Tailwind CSS (UI responsive).	11/02/2026	13/02/2026	40	42	3
16	4.1. Integración completa frontend-backend y pruebas funcionales.	16/02/2026	19/02/2026	43	46	4
17	4.2. Optimización y corrección de errores.	20/02/2026	23/02/2026	47	50	4
18	4.3. Despliegue en la nube (Render/Railway/Vercel).	24/02/2026	25/02/2026	51	52	2
19	4.4. Pruebas finales en producción y documentación.	26/02/2026	27/02/2026	53	53	1

