Documento de Despliegue

YGGDRASOUND

YGGSO

Atteneri de Jesús Torres González

Nombre del fichero:	DAW_PRW_YGGSO_5. Documento de despliegue.odt
Fecha de esta versión:	24/02/2026

Historial de revisiones

Fecha	Descripción	Autor
24/02/2026	Creación del documento de despliegue	Atteneri de Jesús Torres González

ÍNDICE

1 INTRODUCCIÓN

Este despliegue corresponde a la versión 1.0.0 (Release inicial) de la plataforma **Yggdrasound**. Se trata de la primera puesta en producción de la aplicación Full-Stack, que permite la gestión completa de un comercio electrónico de vinilos integrado con un sistema de consulta de conciertos en tiempo real. La arquitectura se basa en un cliente React y un servidor API REST bajo Node.js, utilizando PostgreSQL como persistencia de datos.

2 DETALLE DE LAS MEJORAS

2.1 Mejoras técnicas

- **Arquitectura Desacoplada:** Separación total entre Frontend (React) y Backend (Express) para facilitar escalabilidad.
- **Gestión de Estado Ligera**: Implementación de Zustand para el manejo eficiente del carrito y autenticación en el cliente.
- **Seguridad Robusta:** Almacenamiento de credenciales mediante hashing con bcrypt y protección de rutas vía JWT.
- **ORM Seguro**: Uso de Sequelize para prevenir ataques de inyección SQL mediante consultas parametrizadas.

2.2 Mejoras funcionales

- **Catálogo Interactivo:** Visualización paginada y filtrable de vinilos con detalles técnicos y tracklists.
- **Sistema "Buscar Shows":** Integración de scraping bajo demanda de Bandcamp para localizar conciertos del artista consultado.
- **Flujo de Compra Completo:** Gestión de carrito, lista de deseos e historial de pedidos para usuarios registrados.
- **Panel de Administración:** Herramientas CRUD para que el administrador gestione el catálogo y modere comentarios.

3 SERVIDOR DE APLICACIONES

3.1 Identificación del servidor

- **Nombre del servidor en la red:**
  - yggdrasound_backend: Instancia de servidor de aplicaciones en **Render** encargada de ejecutar el runtime de Node.js y la API REST.
  - yggdrasound: Instancia de hosting estático en **Netlify** encargada de servir los activos compilados de la interfaz React.

- **Dominio**:
  - **Backend (API):** https://yggdrasound-backend.onrender.com.
  - **Frontend (Cliente)**: https://yggdrasound.netlify.app/.

- **Directorio raíz del host virtual:**
  - **Backend:** /backend (Carpeta raíz del servicio en Render donde reside la lógica de Node.js).
  - **Frontend:** /frontend/dist (Ubicación de los archivos estáticos generados tras la compilación con Vite que Netlify distribuye).

3.2 Requisitos mínimos

- **Servidor de aplicaciones:** Node.js.
- **Versión mínima**: 22.18.0.
- **Otros:** npm versión 10.9.3 y motor de empaquetado Vite 7.2.7.

3.3 Configuración recomendada

- **Tiempo máximo de procesamiento:** 5 segundos (especialmente para la función de scraping de conciertos para evitar timeouts).
- **Memoria de usuario recomendada**: Mínimo 512MB RAM para el entorno de ejecución del backend.
- **Otros**: Configuración de variables de entorno (.env) para gestionar secretos, configuraciones, conexiones y puertos.

4 SERVIDOR DE BASE DE DATOS

4.1 Identificación del servidor

- **Nombre del servidor en la red:** dpg-d6ea26jh46gs73chcla0-a (Hostname interno asignado por Render).
- **Esquema o código de la base de datos**: yggdrasound_db (Esquema lógico: public).

4.2 Requisitos mínimos

- **Servidor de base de datos:** PostgreSQL.
- **Versión mínima:** 15.0 (vía imagen Docker postgres:15).
- **Otros**: Puerto por defecto 5432.

4.3 Scripts

- **Ubicación**: Raíz del proyecto, archivo yggdrasound_dump.sql.
- **Contenido**:
  - **Estructura completa**: Crea las 8 tablas principales (usuarios, vinilos, pedidos...) y las 4 tablas necesarias para las relaciones (como los géneros de cada vinilo).
  - **Base de datos alimentada:** Incluye los INSERT con el catálogo completo de vinilos, géneros, comentarios y usuarios de prueba. Está lista para usar nada más importar el archivo.

5 FICHEROS O BINARIOS A DESPLEGAR

5.1 Ubicación

Los archivos y scripts necesarios para el despliegue se encuentran en la raíz y en las carpetas principales del proyecto entregado:

- **Scripts de Base de Datos:**
  - yggdrasound_dump.sql: Situado en la raíz. Contiene el volcado completo (estructura y datos) necesario para inicializar PostgreSQL.
- **Servidor Backend (API):**
  - Carpeta /backend: Contiene el código fuente en Node.js, incluyendo la lógica de modelos (Sequelize), controladores y rutas.
  - backend/Dockerfile: Configuración para despliegue mediante contenedores.
- **Aplicación Frontend (Interfaz):**
  - Carpeta /frontend: Código fuente en React + Vite.
  - Para el despliegue en producción, se deberá generar la carpeta /frontend/dist mediante el comando npm run build (estos son los "artefactos" finales).
- **Documentación y Diseño:**
  - Carpeta /docs: Incluye el histórico de documentos de alcance, diseño técnico y los diagramas ER y de casos de uso requeridos en la rúbrica.
- **Configuración de Orquestación:**
  - docker-compose.yml: Localizado en la raíz para facilitar el despliegue de los servicios en conjunto.

6 OPERATIVA

6.1 Operaciones de base de datos

Para garantizar la persistencia de datos se debe realizar los siguientes pasos:

1. **Preparación de la instancia:** Disponer de un servidor PostgreSQL (v15 o superior) activo.
2. **Creación del contenedor lógico:** Ejecutar la sentencia SQL: CREATE DATABASE yggdrasound_db;
3. **Importación del volcado (Dump):** Cargar el archivo yggdrasound_dump.sql ubicado en la raíz del proyecto para generar el esquema y los datos iniciales.
   - **Comando recomendado:** psql -U [usuario] -d yggdrasound_db -f yggdrasound_dump.sql
4. **Permisos:** Asegurar que el usuario de conexión tenga privilegios completos sobre el esquema public.

6.2 Operaciones de despliegue

El despliegue de **Yggdrasound** puede realizarse mediante dos modalidades. Se recomienda encarecidamente la **Opción A** por su alta fidelidad y facilidad de configuración.

Opción A: Despliegue Automatizado (Docker) - RECOMENDADO

Esta modalidad utiliza contenedores para asegurar que el entorno sea idéntico en cualquier máquina, automatizando la instalación de Node.js, PostgreSQL y la configuración de redes.

1. **Requisitos previos:** El operador debe tener instalado **Docker Desktop** (que incluye Docker Compose) y asegurarse de que los puertos 3000, 5173 y 5432 estén libres.
2. **Preparación de variables:** Antes de iniciar, es indispensable que exista el archivo ./backend/.env con las credenciales por defecto. Para que Docker conecte correctamente, los valores deben ser:
   - DB_NAME=yggdrasound_db
   - DB_USER=postgres
   - DB_PASSWORD=yggdrasound_password
   - DB_HOST=postgres
3. **Ejecución del despliegue:** Abrir una terminal en la raíz del proyecto (donde se ubica el archivo docker-compose.yml) y ejecutar el comando: docker-compose up --build
4. **Automatización de Datos:** El sistema detectará el archivo yggdrasound_dump.sql y lo importará automáticamente en la base de datos durante el primer arranque, por lo que no es necesario ejecutar scripts SQL manualmente.
5. **Acceso al sistema:** Una vez que la terminal indique que los servicios están "running", se podrá acceder a la aplicación mediante la URL: http://localhost:5173.

Opción B: Despliegue Manual (Paso a paso)

Para despliegues en entornos donde Docker no esté disponible, se deberán seguir estos pasos de forma secuencial:

Paso 1: Configuración del Backend

1. Acceder a la carpeta /backend e instalar las dependencias con el comando **npm install.**
2. Crear un archivo .env basándose en el archivo .env.example. Es crítico configurar correctamente el DB_PASSWORD y el JWT_SECRET. Para ejecuciones locales fuera de Docker, el DB_HOST debe establecerse como localhost.
3. Iniciar el servidor de la API ejecutando **npm start.**

Paso 2: Configuración del Frontend

1. Acceder a la carpeta /frontend e instalar las dependencias con **npm install.**
2. Crear un archivo .env y definir la variable VITE_API_URL con la dirección del backend (ejemplo: http://localhost:3000).
3. Generar los artefactos de producción con el comando **npm run build.**
4. Para la puesta en marcha, se debe servir el contenido de la carpeta /frontend/dist mediante un servidor web o ejecutar **npm run dev** para una previsualización rápida.

6.3 Condiciones y verificación de éxito

Para confirmar que el despliegue se ha completado correctamente, realizar las siguientes comprobaciones:

1. **Salida de la API:** Acceder a https://yggdrasound-backend.onrender.com/api/vinyl/. Se debe recibir un objeto JSON con el catálogo de vinilos cargado desde el dump.
2. **Carga de Interfaz:** Al acceder a la URL del frontend (https://yggdrasound.netlify.app/), se debe visualizar la página de inicio con el contenido y el grid de productos correctamente renderizado.
3. **Flujo de Autenticación:** Realizar un Login con las credenciales de prueba. Si se genera el token JWT y se accede al perfil, la conexión Backend-BD es exitosa.
4. **Gestión de Estado:** Añadir un producto al carrito y navegar por diferentes rutas. Si el contador del carrito persiste, la sincronización entre **Zustand** y la lógica de la aplicación es correcta.

7 CONTROL DE VERSIONES

- **Repositorio Remoto:** https://github.com/attetorres/yggdrasound/commits/main/
- **Rama principal:** main.
- **Descripción del historial:** El repositorio contiene el historial completo de commits realizados durante el desarrollo, organizados cronológicamente, lo que permite verificar la evolución del proyecto y el cumplimiento de los hitos establecidos.

8 OBSERVACIONES

- **Configuración de Entorno:** El correcto funcionamiento de la conexión Sequelize depende de que las variables en el archivo .env del backend coincidan exactamente con las de la base de datos PostgreSQL. Cualquier discrepancia en DB_PASSWORD o DB_HOST impedirá el arranque del servidor.
- **Orquestación con Docker:** El archivo docker-compose.yml incluido permite levantar todo el stack (Base de Datos, Backend y Frontend) de forma automática. Es la opción recomendada para evaluar el proyecto en local sin necesidad de configurar dependencias manualmente.

- **Acceso y Seguridad (JWT):** Para testear las funcionalidades protegidas y el panel de administración, se ha habilitado una cuenta de acceso total en el script de datos iniciales:
  - Usuario Administrador: admin@yggdrasound.com
  - Contraseña: 123456789
  - Este usuario posee el rol de administrador is_admin: true, necesario para superar el middleware de autenticación en las rutas /admin.
- **Visualización de Assets:** Las portadas de los vinilos se sirven mediante URLs externas. Se requiere conexión a internet activa durante la navegación para que la interfaz cargue correctamente todos los recursos visuales del catálogo.