

Final Practicas Profesionalizantes ||

"Un Mar de Música" es una plataforma web creada para facilitar la administración de canciones y álbumes de artistas. Su propósito principal es brindar a los músicos una herramienta que les permita organizar y difundir su material musical.

Esta aplicación web ofrece la posibilidad de crear y personalizar un perfil, subir pistas, agruparlas en álbumes y gestionar toda la biblioteca musical de forma práctica y efectiva.

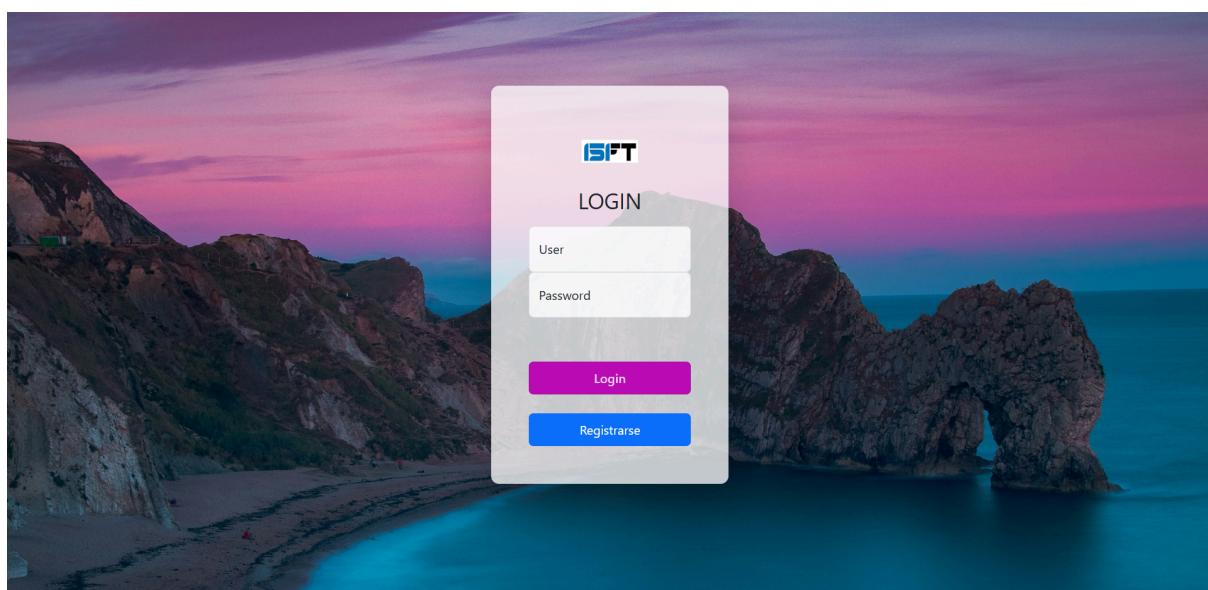
El desarrollo del backend se realizó con Node.js, utilizando MySQL como base de datos para almacenar la información de los usuarios. En el frontend se emplea EJS para generar vistas dinámicas, junto con Bootstrap para ofrecer un diseño moderno y adaptable a distintos dispositivos.

Github:[Repositorio](#)

Introducción del proyecto:[Un Mar de Musica](#)

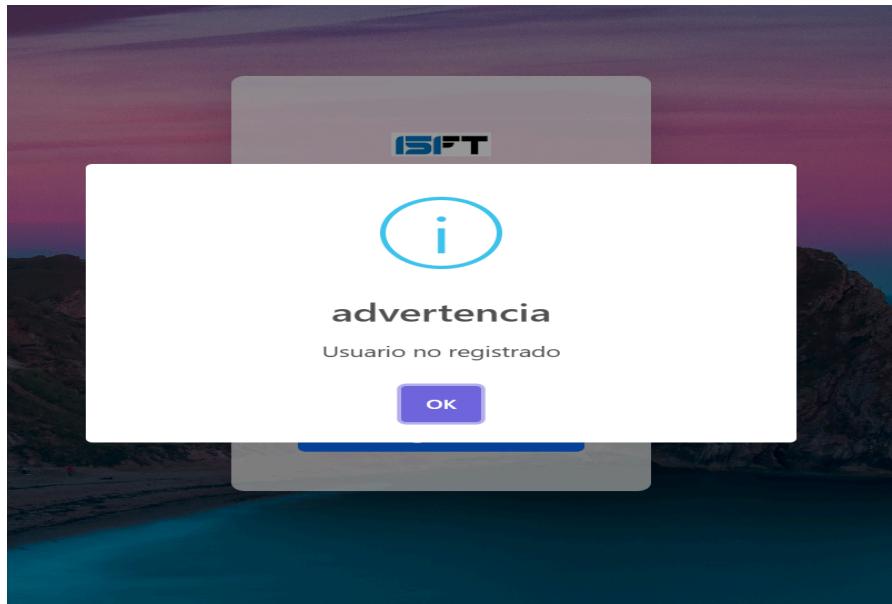
Cuando el usuario intenta entrar a la página web, el backend revisa la cookie con el JWT. Si el token es válido y corresponde a un usuario existente, se permite el acceso y la vista principal se muestra normalmente; si no, se redirige a la página de login.

Login

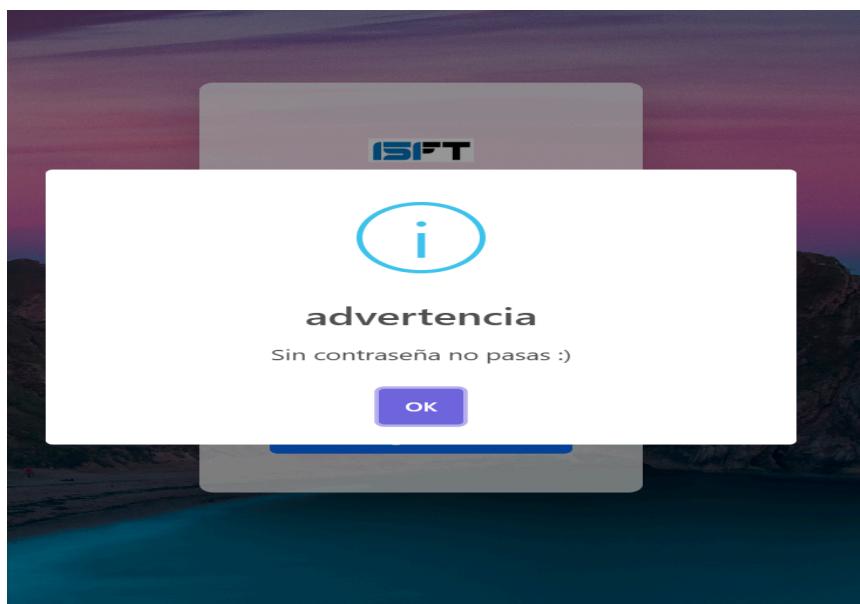


En el inicio de sesión, la vista muestra un formulario con campos para usuario y

contraseña. Si el usuario deja un campo vacío o el backend no encuentra ni el usuario ni la contraseña responde renderizando nuevamente la vista de login con una alerta de Bootstrap:

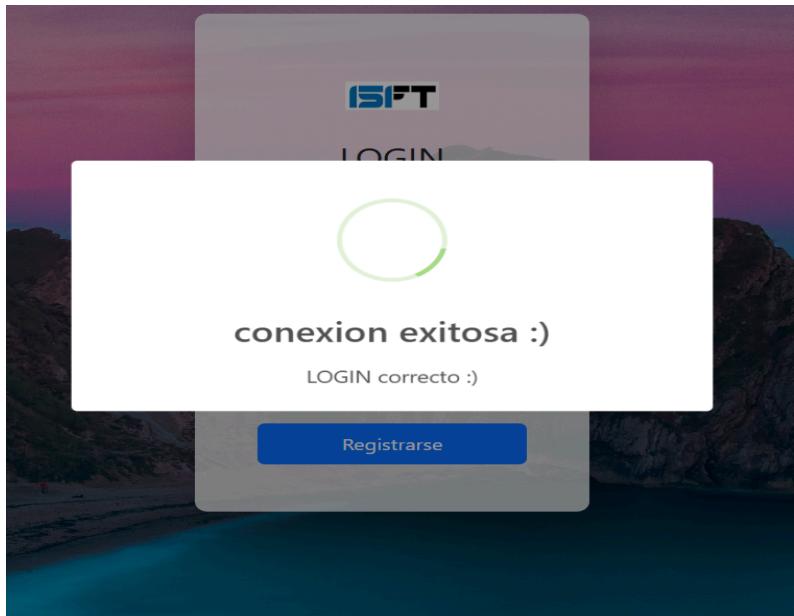


Si el nombre del usuario es válido pero la contraseña es incorrecta, se vuelve a mostrar la vista con otro aviso de error:



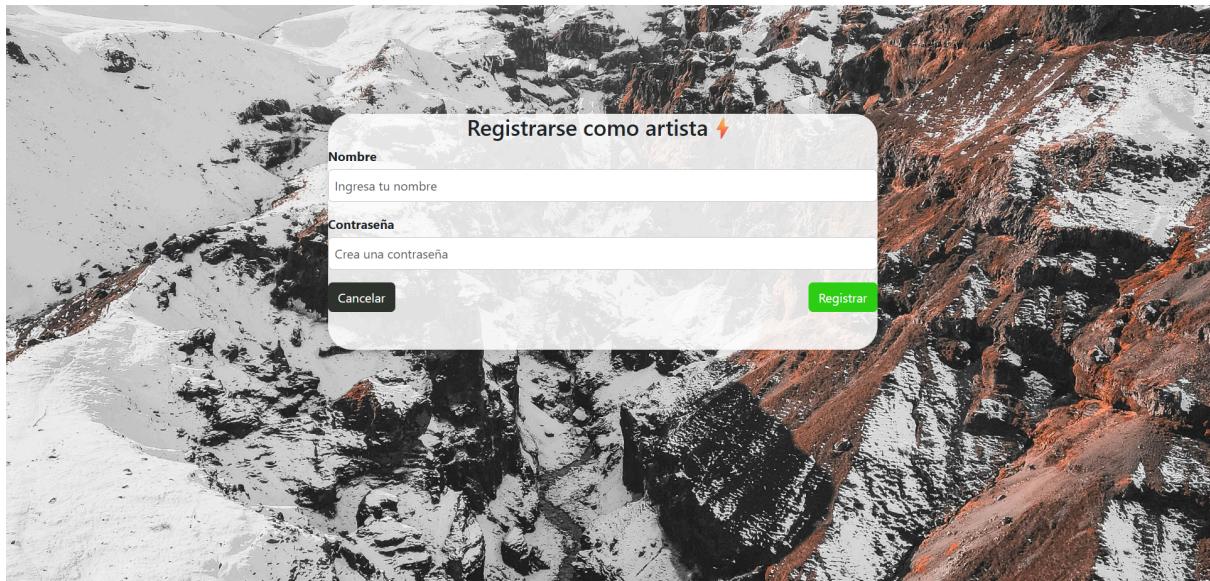
Si son correctas, el backend crea un token JWT, lo almacena como cookie y renderiza la misma vista, pero ahora con una alerta de éxito y un mensaje de

bienvenida que desaparece automáticamente gracias a los parámetros enviados desde el backend:



En caso de que el usuario no esté registrado, puede ingresar al formulario de registro a través del login:

Registro



En el registro, la vista incluye un formulario con campos para nombre, usuario y contraseña. Cuando el usuario completa estos campos y envía el formulario, la información se envía al servidor mediante una solicitud POST.

El backend recibe estos datos, encripta la contraseña con bcrypt js y guarda todo en la base de datos MySQL. Una vez completado, la vista redirige al login para que pueda ingresar su usuario y acceder a la página principal.



A screenshot of a MySQL database table titled 'Opciones extra'. The columns are labeled 'id', 'user', 'name', and 'pass'. There are four rows of data:

	<input type="button" value="←"/> <input type="button" value="→"/>	<input type="button" value="id"/>	<input type="button" value="user"/>	<input type="button" value="name"/>	<input type="button" value="pass"/>
<input type="checkbox"/>	<input type="button" value="Editar"/>	<input type="button" value="Copiar"/>	<input type="button" value="Borrar"/>	1 un pibe	tomas \$2a\$08\$wnnJpVyboomLKF6MBtEeEuW7Ww7ocncl8r6iPx1m.Qz...
<input type="checkbox"/>	<input type="button" value="Editar"/>	<input type="button" value="Copiar"/>	<input type="button" value="Borrar"/>	2 un pibe	tomas \$2a\$08\$A8iT2kzDEVtS1b4dk56T.ejJVVOq.DO864ft29XEpXu...
<input type="checkbox"/>	<input type="button" value="Editar"/>	<input type="button" value="Copiar"/>	<input type="button" value="Borrar"/>	3 unChango	chango \$2a\$08\$Uj67gYbijGvUopjzf07VCurmNYXM/BRwD24P8L8W6Yt...
<input type="checkbox"/>	<input type="button" value="Editar"/>	<input type="button" value="Copiar"/>	<input type="button" value="Borrar"/>	6 Franco	franco \$2a\$08\$d.v9wKPAU0vOLP9xmoobQ.l3nC0.dkLAM44Bjp0lNEW...

Dashboard



The dashboard interface has a sidebar on the left with the following menu items:

- Dashboard
- Subir Canciones
- Subir Álbum
- Buscar Artistas
- Borrar Cancion
- Borrar Álbum
- Cerrar Sesión

The main content area is titled 'Canciones Subidas' and lists three files:

- Tecno.wav
- GuitarSolo2.mp3
- GuitarSolo.wav

Each file entry includes a preview player (showing 0:00 / 1:00, 0:00 / 1:42, 0:00 / 1:10 respectively), a volume control, and a red 'Eliminar' (Delete) button.

Below this is a section titled 'Subir Canción' with a 'Seleccionar audio' input field containing 'Seleccionar archivo' and 'Sin archivos seleccionados', and a green 'Subir Canción' button.

Al ingresar con sus credenciales, el artista es recibido por una pantalla principal que muestra un acceso rápido a todas sus funciones. Desde aquí, puede subir nuevas canciones o álbumes, organizarlos en su biblioteca personal y visualizarlos en una lista de las mismas.

Cada canción o álbum cuenta con botones para reproducir su contenido directamente en el navegador y para eliminarlo de su cuenta si ya no lo desea. La reproducción se realiza de manera fluida mediante un reproductor integrado.

El reproductor integrado ofrece al usuario un control completo sobre su pista musical. Permite ajustar el volumen de cada pista en reproducción, descargar la

canción en caso de haber perdido el archivo original en su dispositivo y, si lo desea, reproducir varias canciones al mismo tiempo.

Próximamente, el dashboard incorporará una barra de búsqueda y un explorador de artistas. Esta nueva función permitirá buscar a otros músicos registrados, acceder a sus perfiles y escuchar sus canciones o ver sus álbumes publicados.

Finalmente, al cerrar sesión, se borra el token que identifica al usuario como logueado y redirige el usuario al login.

Dependencias:

- **bcryptjs**: encripta las contraseñas antes de guardarlas en la base de datos y para compararlas al iniciar sesión.
- **cookie-parser**: Permite que la aplicación lea y gestione cookies enviadas por el navegador.
- **dotenv**: Carga las variables de entorno desde un archivo .env a process.env para mantener datos sensibles fuera del código.
- **Ejs** : Es el motor de plantillas que permite generar HTML dinámico en el servidor usando datos pasados desde el backend.
- **express**: Se encarga de manejar rutas, peticiones HTTP, middleware y toda la estructura base del servidor.
- **express-session**: Es el módulo que gestiona sesiones de usuario en Express. Guarda información temporal en el servidor para mantener al usuario autenticado entre peticiones.
- **jsonwebtoken**: Permite crear y verificar tokens JWT (JSON Web Tokens) para que la autenticación sea segura sin tener que almacenar datos de sesión en el servidor.
- **Multer**: Es el middleware que maneja la subida de archivos en formularios HTML, como las canciones que se suben en el dashboard.
- **mysql**: Permite a Node.js conectarse y realizar consultas a bases de datos MySQL.
- **nodemon**: Es la herramienta que reinicia automáticamente el servidor de Node.js cada vez que detecta cambios en el código, para acelerar el desarrollo.