**Documentación del Proyecto: Aplicación de Música**

**1. Introducción**

Este documento describe el desarrollo de una aplicación web llamada "MUSICA", realizada como parte del Trabajo Práctico Integrador (TPI) de la materia Programacion 3. La aplicación tiene como objetivo ofrecer una experiencia de usuario intuitiva para la gestión y exploración de canciones, artistas y álbumes. Está desarrollada en ReactJS y consume una API RESTful para la manipulación de datos.

**2. Descripción del Proyecto**

La aplicación MUSICA permite a los usuarios explorar un catálogo de música, gestionar su perfil, y realizar operaciones CRUD sobre las canciones. Los usuarios autenticados pueden agregar, editar y eliminar canciones. Además, la aplicación ofrece una interfaz de usuario moderna, con navegación intuitiva y segura mediante autenticación JWT.

**3. Requerimientos del Proyecto**

**3.1 General**

* **Framework:** ReactJS
* **Control de versiones:** Git (Repositorio en GitHub)
* **Entorno de desarrollo:** Local, desplegado en Vercel.
* **API:** HarmonyHub (API proporcionada por la cátedra).
* **Autenticación:** JWT.

**3.2 Autenticación y Autorización**

* **JWT:** La autenticación y manejo de sesiones se realiza mediante tokens JWT. Todas las solicitudes a la API están autenticadas.
* **Protección de rutas:** Rutas protegidas para áreas que requieren autenticación, como la página de perfil.

**3.3 UI e Implementación de Componentes**

* **Interfaz de Usuario:** Se prioriza una UI funcional y fácil de usar.
* **Componentes React:** Se utilizan hooks como useState, useEffect, y useContext para el manejo de estado.
* **Contexto:** AuthContext es utilizado para compartir el estado de autenticación entre componentes.
* **Estilos:** Se utilizó CSS puro para el diseño de la interfaz, con un enfoque en la simplicidad y la consistencia visual.

**3.4 Consumo de la API**

* **Operaciones CRUD:** La aplicación interactúa con varios endpoints para crear, leer, actualizar y eliminar canciones.
* **Manejo de errores:** Se implementan notificaciones visuales para informar al usuario sobre el estado de las operaciones, incluyendo mensajes de error.

**3.5 Enrutamiento**

* **React Router:** Se implementaron rutas para las secciones principales de la aplicación:
  + **/spotify** - Página de inicio con una lista de canciones.
  + **/login** - Página de inicio de sesión.
  + **/profile** - Página de perfil del usuario.
  + **/ -** Página principal de los datos

- Página de detalles de una canción específica.

* **Rutas protegidas:** Solo los usuarios autenticados pueden acceder a la página de perfil y realizar operaciones de edición y eliminación de canciones.
* **Página 404:** Se incluye una página de error para rutas no encontradas.

**4. Funcionalidades Implementadas**

**4.1 Registro e Inicio de Sesión**

* Los usuarios pueden registrarse y autenticarse mediante el formulario de inicio de sesión, que envía credenciales a la API para recibir un JWT.

**4.2 Exploración de Música**

* La página de inicio muestra una lista de canciones con la opción de ver detalles, editar o eliminar (si el usuario está autenticado).

**4.3 Gestión de Canciones**

* Los usuarios autenticados pueden agregar nuevas canciones, editar o eliminar las existentes. Las operaciones se realizan mediante formularios modales.

**4.4 Perfil de Usuario**

* La página de perfil muestra la información del usuario autenticado y permite modificar sus datos.

**5. Documentación del Código**

El código fuente de la aplicación está documentado con comentarios claros que explican la funcionalidad de cada componente y hook. Se sigue una estructura modular para facilitar el mantenimiento y la escalabilidad.

**6. Conclusión**

El proyecto MUSICA es una demostración del conocimiento adquirido durante el trascurso universitario, aplicando tecnologías modernas y mejores prácticas en el desarrollo web. Se alcanzaron todos los objetivos planteados, ofreciendo una aplicación funcional, segura y con una experiencia de usuario agradable.