

## Desafío Evaluado - Mi repertorio

- Para realizar este desafío debes haber estudiado previamente todo el material disponible correspondiente a la unidad.
- Una vez terminado el desafío, comprime la carpeta que contiene el desarrollo de los requerimientos solicitados y sube el `.zip` en el LMS.
- Puntaje total: 10 puntos
- Desarrollo desafío:
  - El desafío se debe desarrollar de manera Grupal.
  - Para la realización del desafío necesitarás apoyarte del archivo *Apoyo Desafío - Mi Repertorio*.

### Capítulos

El desafío está basado en el siguiente capítulo de la lectura:

- Levantando un servidor con conexión a PostgreSQL.
- Insertando registros.
- Consultado registros.
- Actualizando registros.
- Eliminando registros.

### Descripción

La escuela de música “E-Sueño” está motivando a sus estudiantes de canto a presentarse en vivo y se puso en contacto con el restaurante del sector para utilizar su tarima e iniciar un calendario de presentaciones. Para conocer y gestionar las canciones que cantarán sus estudiantes, la escuela contrató a un desarrollador freelance para la creación de una aplicación tipo CRUD.

En este desafío deberás desarrollar un servidor con Express que utilice el paquete pg para conectarse con PostgreSQL y utilice funciones asíncronas para hacer las consultas a la base de datos.

El servidor deberá disponibilizar las siguientes rutas:

- **POST /cancion:** Recibe los datos correspondientes a una canción y realiza la inserción en la tabla **repertorio**.
- **GET /canciones:** Devuelve un JSON con los registros de la tabla **repertorio**.
- **PUT /cancion:** Recibe los datos de una canción que se desea editar y ejecuta una función asíncrona para hacer la consulta SQL y actualice ese registro de la tabla **repertorio**.
- **DELETE /cancion:** Recibe por queryString el id de una canción y realiza una consulta SQL a través de una función asíncrona para eliminarla de la base de datos.

Tienes a disposición un **Apoyo Desafío - Mi Repertorio** con la aplicación cliente que se muestra en la siguiente imagen, lista para el consumo de estas rutas, por lo que deberás enfocarte solo en el desarrollo backend.

🎵 Mi repertorio 🎵

Canción:

Artista:

Tono:

Agregar

Tabla de canciones 🎤

#	Canción	Artista	Tono	
1	A dios le pido	Juanes	Dm	<button>Editar</button> <button>Eliminar</button>
2	Música Ligera	Soda Estéreo	Bm	<button>Editar</button> <button>Eliminar</button>

Si es de tu agrado, puedes crear tu propia maqueta siempre y cuando cumpla con los requerimientos del desafío. Para la creación de la base de datos y la tabla **repertorio** utiliza las siguientes consultas SQL

```
CREATE DATABASE repertorio ;
```

```
CREATE TABLE canciones (id SERIAL, titulo VARCHAR(50), artista  
VARCHAR(50), tono VARCHAR(10));
```

## Requerimientos

1. Crear una ruta **POST /cancion** que reciba los datos correspondientes a una canción y realice a través de una función asíncrona la inserción en la tabla **repertorio**.
2. Crear una ruta **GET /canciones** que devuelva un JSON con los registros de la tabla **repertorio**.
3. Crear una ruta **PUT /cancion** que reciba los datos de una canción que se desea editar, ejecuta una función asíncrona para hacer la consulta SQL correspondiente y actualice ese registro de la tabla **repertorio**.
4. Crear una ruta **DELETE /cancion** que reciba por queryString el id de una canción y realiza una consulta SQL a través de una función asíncrona para eliminarla de la base de datos.