

Desafío - Mi gestión de cursos

- Para realizar este desafío debes haber estudiado previamente todo el material disponibilizado 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.
- Desarrollo desafío:
 - El desafío se debe desarrollar de manera Grupal.
 - Para la realización del desafío necesitarás apoyarte del archivo *Apoyo Desafío - Mi gestión de cursos*

Capítulos

El desafío está basado en los siguientes capítulos de la lectura:

- API REST
- POST (CREATE) & GET (READ)
- PUT (UPDATE) & DELETE (DELETE)

Descripción

La academia necesita gestionar sus cursos de programación y solicitó la ayuda de uno de sus estudiantes full stack developer para crear una aplicación que permita hacer un CRUD de los cursos usando una API REST propia.

En este desafío deberás crear una API REST con Express y el paquete pg para servir el backend de una aplicación tipo CRUD en el lado del cliente.

En el **Apoyo Desafío - Mi gestión de cursos** se disponibiliza el `index.html` con la aplicación cliente. No obstante, siéntete libre de modificar la interfaz y crear tu propia maqueta, siempre y cuando cumpla con los requisitos y continúe la temática planteada.

Utilizar el siguiente objeto como referencia para el modelo de datos:


```
curso = {  
  nombre: "Express Js",  
  nivel: 5,  
  fecha: "2021-01-20",  
  duracion: "20 dias"  
}
```

La siguiente imagen te servirá como referencia de la interfaz gráfica que encontrarás en el Apoyo Desafío:

Agregar Cursos

Nombre

Nivel técnico

Fecha de inicio 

Duración

Tabla de Cursos

#	Nombre	Nivel Técnico	Fecha de Inicio	Duración	-
1	Python	3	2021-01-27	6	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
2	Javascript	5	2021-02-26	12	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Utiliza las siguientes instrucciones SQL para la creación de la base de datos y la tabla que deberás usar en este desafío:

```
CREATE DATABASE cursos;
```

```
CREATE TABLE cursos (id SERIAL PRIMARY KEY, nombre VARCHAR(50), nivel  
INT, fecha DATE, duracion INT);
```

Requerimientos

1. Crear una ruta **POST /curso** que reciba un payload desde el cliente con los datos de un nuevo curso y los ingrese a la tabla **cursos**.
2. Crear una ruta **GET /cursos** que consulte y devuelva los registros almacenados en la tabla **cursos**.
3. Crear una ruta **PUT /curso** que reciba un payload desde el cliente con los datos de un curso ya existente y actualice su registro en la tabla **cursos**.
4. Crear una ruta **DELETE /cursos** que reciba el id de un curso como parámetro de la ruta y elimine el registro relacionado en la tabla **cursos**.