

**PROYECTO FINAL:**

## **Módulo II**

# **Backend Developer**

## **Descripción del Proyecto Final:**

El proyecto final consiste en desarrollar una aplicación fullstack compuesta por una API REST construida con Node.js y Express (usando MongoDB o MySQL como base de datos) y una web responsive desarrollada con React (en JSX o TSX). La web debe interactuar con la API para realizar operaciones CRUD completas. Además, se debe incluir una documentación pública de la API (estilo Swagger), limitada únicamente a los endpoints GET, para que otros puedan consumirla.

## Backend (API REST)

- Servidor funcional con Node.js y Express.
- Conexión a base de datos (MongoDB con *mongoose* o MySQL con *mysql2*).
- Rutas REST obligatorias:
  - GET (listar todos, filtrar/buscar).
  - POST (crear nuevo).
  - PUT (reemplazar completo).
  - PATCH (actualizar parcial).
  - DELETE (eliminar).
- Validaciones de datos y manejo de errores.
- Middleware de CORS y JSON correctamente configurado.

## Para sumar puntos

Con la finalidad de obtener más puntos en la entrega final se puede optar por sumar alguna de las siguientes características a la API REST trabajada:

- Sistema de usuarios que controle quién puede acceder a ciertos endpoints
- Sistema de creación y validación de usuarios (bcrypt, registro/login)
- Uso de variables de entorno (dotenv)

## Frontend (Web con React)

- Sitio web trabajado con React (JSX o TSX).
- Diseño responsive (adaptado a desktop y mobile).
- Consumo de la API REST usando fetch o axios.
- Funcionalidades mínimas:
  - Listar registros.
  - Buscar/filtrar.
  - Crear, editar, eliminar registros.
  - Código ordenado, con componentes reutilizables.

## Para sumar puntos

- Tema claro/oscuro
- Feedback de estados (errors/loading/success)
- Placeholder o Skeletons
- Uso de useContext
- Creación de Hooks personalizados (estilo useFetch o useAxios)

## Documentación pública de la API

- Documentación accesible desde la web (por ejemplo, en /public-api).
- Estilo Swagger.
- Describe todos los endpoints disponibles.
- Incluye parámetros, ejemplos de uso y respuestas esperadas.

Esta sección debe estar limitada solo a métodos GET (sin POST, PUT, PATCH ni DELETE).

## Anotaciones

Se pueden sumar los accesos a los endpoints POST, PUT, PATCH y DELETE solo si se trabajó el sistema de usuarios y se verifique previamente que quien está consumiendo la API sea un usuario verificado.