

Prueba Técnica Fullstack: CRUD de Empleados

Objetivo

Desarrollar una aplicación web con un frontend en **Next.js** y un backend en **NestJS** que permita gestionar empleados (crear, editar, listar, eliminar). La interfaz debe incluir **modales** para crear y editar empleados.

Backend - NestJS

Requerimientos:

1. Crear un módulo llamado employees.
2. Modelo Employee con los siguientes campos:
 - id: number (autoincremental)
 - firstName: string
 - lastName: string
 - email: string
 - position: string

Endpoints:

- GET /employees: Lista todos los empleados.
- POST /employees: Crea un nuevo empleado.
- PUT /employees/:id: Actualiza un empleado.
- DELETE /employees/:id: Elimina un empleado.

Puedes usar una base de datos SQLite con TypeORM o una implementación en memoria (array[]) si quieres simplificar.

Frontend - Next.js (App Router)

Requerimientos:

1. Página principal: localhost:3000/employees
2. Mostrar la lista de empleados en una tabla.
3. Botón Nuevo Empleado que abre un modal con formulario.
4. Cada fila debe tener botones de Editar (abre modal con datos) y Eliminar.
5. Validación básica de formularios (campos requeridos).

Stack sugerido:

- React + Next.js (App Router)
 - TailwindCSS
 - Axios o Fetch
 - react-hook-form
 - zustand
-

Criterios de evaluación

- Organización del código (modularización, buenas prácticas).
 - Uso correcto de modales y formularios controlados.
 - Uso de rutas RESTful.
 - Lógica clara y mantenible.
 - Buenas UX/UI mínimas (feedback en errores, loading, etc.)
-

Employee CRUD

New Employee

Name	Email	Position	Actions
John Doe	john.doe@example.com		
Jane Smith	jane.smith@example.com		<div>EditDele</div>
Alice Johnson	alice.johnson@example.com		<div>EditDele</div>
Bob Brown	bob.brown@example.com		<div>EditDele</div>