

## Prueba Técnica – Desarrollador Fullstack Junior (React, React Native, Express, PostgreSQL)

### 1. Introducción

Esta prueba tiene como objetivo evaluar tus habilidades técnicas, organización del código, nociones de arquitectura y buenas prácticas de desarrollo. Dispondrás de **3 días** para completarla.

---

### 2. Objetivo

Construir un sistema de **gestión de tareas colaborativas** con usuarios, autenticación, roles, comentarios en tareas y sincronización entre web y móvil.

---

### 3. Requerimientos funcionales

#### Backend (Node.js + Express + PostgreSQL)

- **Autenticación y autorización:**
  - Registro y login con JWT.
  - Roles: user y admin.
  - Middleware para proteger rutas y validar roles.
- **Gestión de tareas:**
  - Crear, listar, actualizar y eliminar tareas.
  - Cada tarea tiene: título, descripción, estado (pendiente, en progreso, completada), fecha de vencimiento y usuario asignado.
- **Comentarios en tareas:**
  - Un usuario puede agregar comentarios a una tarea.
  - Relación 1:N entre tarea y comentarios.
- **Notificaciones simuladas:**
  - Cuando una tarea cambia de estado, registrar un log en una tabla notifications (no es necesario enviar push real, solo guardar en DB).
- **Subida de archivos:**
  - Permitir adjuntar un archivo a una tarea (por ejemplo, un PDF o imagen).
  - Guardar el archivo en una carpeta local del servidor (/uploads).
  - Guardar en la base de datos la ruta del archivo.
  - Endpoint para descargar/ver el archivo.

- **Base de datos:**
    - Tablas: users, tasks, comments, notifications.
    - Relaciones:
      - users 1:N tasks
      - tasks 1:N comments
      - tasks 1:N notifications
- 

## Frontend Web (React)

- **Pantallas:**
    1. Login / Registro.
    2. Lista de tareas con filtros por estado y fecha de vencimiento.
    3. Formulario para crear/editar tarea con opción de subir archivo.
    4. Vista de detalle de tarea con comentarios y archivo adjunto.
    5. Panel de administración (solo para admin) para ver todas las tareas de todos los usuarios.
  - **Características:**
    - Guardar token JWT en localStorage.
    - Mostrar notificaciones recientes.
    - Manejo de errores y mensajes al usuario.
    - Descargar/ver archivo adjunto.
- 

## App Móvil (React Native con Expo)

- **Pantallas:**
  1. Login / Registro.
  2. Lista de tareas asignadas al usuario.
  3. Detalle de tarea con comentarios y archivo adjunto.
  4. Posibilidad de cambiar estado de la tarea.
- **Características:**
  - Guardar token JWT en AsyncStorage.
  - Sincronizar cambios con backend.
  - Mostrar notificaciones recientes.

- Descargar/ver archivo adjunto.

---

#### 4. Requerimientos técnicos

- **Backend:**
  - Node.js + Express.
  - PostgreSQL, ORM a elección.
  - JWT para autenticación.
  - Variables de entorno (.env).
  - Validaciones con Joi o similar.
- **Estructura modular:**
  - backend/
  - src/
  - routes/
  - controllers/
  - models/
  - middlewares/
  - services/
  - config/
- **Frontend Web:**
  - React, framework a elección.
  - React Router.
  - Componentes reutilizables.
- **Mobile:**
  - React Native con Expo.
  - React Navigation.

---

#### 6. Criterios de evaluación

1. Organización del código y separación de responsabilidades.
2. Buenas prácticas (nombres claros, manejo de errores, validaciones).
3. Seguridad básica (JWT, no exponer credenciales).
4. Capacidad de implementar relaciones y lógica de negocio.

---

## 7. Entregables

- Repositorio en GitHub con:
  - backend/
  - frontend/
  - mobile/
- Script SQL o migraciones.

---

## 8. Plazo

- **Duración:** 3 días.