

Prueba Técnica para Vacante de Practicante de Ingeniería de Software

Objetivo: Evaluar las habilidades técnicas para implementar una aplicación que registre a los pacientes de un hospital infantil. Esta prueba está diseñada para evaluar conocimientos en desarrollo web y la integración de herramientas modernas, con un enfoque en habilidades prácticas y aplicación real en un entorno de desarrollo profesional.

Descripción del Proyecto

Se desea desarrollar una aplicación de registro de pacientes para un hospital infantil. Los pacientes son enviados de diferentes hospitales, y el sistema debe permitir registrar la información relevante de cada paciente.

Requerimientos Técnicos del Proyecto

1. Modelo de Datos:

- Paciente:
 - Nombre completo
 - Edad
 - Sexo
 - Fecha de nacimiento
 - Ciudad de origen
 - Fecha de inscripción
 - Hospital de origen
 - Nombre del tutor
 - Teléfono del tutor

2. Requisitos Funcionales:

- Crear una interfaz simple que permita registrar nuevos pacientes.
- Visualizar una lista de pacientes registrados con la información básica.
- Permitir filtrar la lista de pacientes por nombre, ciudad de origen y hospital de origen.
- Editar y actualizar la información de un paciente registrado.
- Eliminar un registro de paciente.

3. Requisitos No Funcionales:

- El diseño de la aplicación debe ser intuitivo y fácil de usar.
- El código debe ser limpio, bien comentado y seguir las mejores prácticas.

Requisitos Técnicos para la Prueba

Backend:

- Implementa un backend usando **Node.js** con **Express.js**(opcional) o un framework similar.
- Desarrollar un modelo de base de datos relacional como **PostgreSQL** o **MySQL** que refleje adecuadamente las necesidades del sistema
- Crea una API RESTful con los siguientes endpoints:
 - **POST /patients**: Para registrar un nuevo paciente.
 - **GET /patients**: Para listar todos los pacientes.
 - **GET /patients/:id**: Para obtener la información detallada de un paciente específico.
 - **PUT /patients/:id**: Para actualizar la información de un paciente.
 - **DELETE /patients/:id**: Para eliminar un paciente.

Frontend:

- Usa **React** con **Next.js**(opcional) para el frontend.
- Crea una página de registro de pacientes donde el usuario pueda ingresar los datos requeridos.
- Crea una página de lista de pacientes que consuma la API del backend para mostrar todos los registros.
- Permitir la edición y eliminación de pacientes desde la interfaz de usuario.

Validaciones:

- Validar todos los campos de entrada en el frontend antes de enviarlos al servidor.
- Implementar validaciones en el backend para asegurarse de que los datos sean correctos y estén completos.

Entregables:

1. **Repositorio de GitHub:**
 - Subir todo el código de la aplicación a un repositorio de GitHub público.
 - Incluye un archivo **README.md** con instrucciones claras sobre cómo configurar e iniciar el proyecto.

Criterios de Evaluación:

- Correctitud y funcionamiento de la aplicación.
- Calidad del código (estructura, comentarios, y mejores prácticas).
- Estructura y diseño de la base de datos.
- Habilidad para manejar datos y realizar validaciones en el backend y frontend.
- Implementación de funciones de CRUD y su integración.

Duración:

Tiempo estimado para completar la prueba: **1 semana**.

Tecnologías Opcionales (Recomendadas):

Para mejorar la eficiencia y calidad del desarrollo, se recomiendan las siguientes herramientas y tecnologías opcionales:

- **TypeScript:** Utiliza TypeScript para mayor seguridad en el desarrollo y tipado estático.
- **React Hook Form o Formik:** Usa cualquiera de estas bibliotecas para manejar formularios y sus validaciones de manera eficiente.
- **MUI v5 (Material-UI) o PrimeReact:** Utiliza alguna de estas bibliotecas de componentes para construir una interfaz de usuario moderna y responsiva.
- **Tailwind CSS:** Usa esta biblioteca de utilidades para estilizar los componentes de la interfaz de usuario de manera eficiente.
- **Axios:** Utiliza Axios para realizar llamadas a la API del backend.
- **Prisma:** Usa Prisma como ORM para manejar la conexión a la base de datos y realizar operaciones CRUD de manera más sencilla.

Estas herramientas y tecnologías son opcionales, pero su uso es altamente recomendado para mejorar la experiencia de desarrollo, mantenimiento y escalabilidad del proyecto.