

Ejercicio Práctico para Candidatos a Analista de Desarrollo: Creación de una Aplicación de Lista de Tareas (To-Do List)

Gracias por participar en nuestro proceso de selección. Como parte de nuestra evaluación, te invitamos a completar un ejercicio práctico que nos ayudará a conocer tus habilidades en el desarrollo y análisis de software. El objetivo es crear una aplicación de lista de tareas (to-do list) que te permita demostrar tus capacidades para abordar problemas, implementar soluciones eficientes y realizar análisis funcionales.

Requerimientos:

1. Frontend:

- Desarrolla una interfaz de usuario que permita a los usuarios:
- Crear tareas, definiendo una descripción y una fecha límite.
- Marcar tareas como completadas.
- Editar y eliminar tareas existentes.
- Filtrar tareas por estado (pendiente, completada).

2. Backend:

- Implementa un backend que permita manejar las siguientes funcionalidades:
- Creación, edición y eliminación de tareas.
- Marcar tareas como completadas o pendientes.
- Almacenar los datos en una base de datos simple (puede ser SQL o una base en memoria).

3. Integración:

- Asegúrate de que el frontend y el backend se comuniquen de manera efectiva.
- Utiliza un API RESTful para conectar ambas partes.

4. Funcionalidades Adicionales (Opcional):

- Agrega una funcionalidad para generar reportes de las tareas completadas y pendientes.
- Incluye la opción de establecer prioridad a las tareas (alta, media, baja).

5. Estilo y Diseño:

- Mejora la apariencia visual de la aplicación utilizando CSS.
- Asegúrate de que la aplicación sea responsiva y se vea bien en dispositivos móviles y de escritorio.

6. Pruebas:

- Realiza pruebas básicas para garantizar que las funcionalidades principales funcionen como se espera.

- Documenta cualquier prueba realizada, incluyendo casos de prueba exitosos y problemas encontrados.

7. Documentación y Presentación:

- Documenta brevemente tu código, destacando las decisiones de diseño y tecnologías utilizadas.
- Prepara una breve presentación (5-10 minutos) para explicar la arquitectura de la aplicación y el flujo de trabajo.

8. Versionado y Control de Código:

- Utiliza una plataforma de control de versiones como GitHub o GitLab para gestionar tu código.
- Realiza commits regulares con mensajes descriptivos que reflejen los cambios realizados.

Evaluación:

- Organización y claridad del código.
- Eficiencia en la implementación de las funcionalidades requeridas.
- Capacidad para resolver problemas y agregar valor con funcionalidades adicionales.
- Calidad del diseño de la interfaz de usuario.
- Claridad y efectividad en la presentación del proyecto.

Entrega:

La prueba debe ser entregada en un máximo de 3 días calendario.