



Desafío de Prueba Técnica

Para esta prueba, te recomendamos que puedas generar un repositorio en Git, subirlo público y luego brindarnos el link para que podamos descargarlo.

Estás trabajando en una aplicación de gestión de tareas. Se te pide que crees un componente de lista de tareas (TaskList) que muestre una lista de tareas y permita marcar las tareas como completadas. Cada tarea debe tener un botón para marcarla como completada.

Requisitos:

Componente TaskList con Funcionalidades Adicionales

En el contexto de una aplicación de gestión de tareas, se te solicita desarrollar un componente TaskList con funcionalidades adicionales para mejorar la experiencia del usuario.

Requisitos del Componente TaskList Mejorado:

- **Funcionalidad Básica:** El componente TaskList debe mostrar una lista de tareas y permitir a los usuarios marcarlas como completadas.
- **Filtro de Tareas:** Agrega la capacidad de filtrar las tareas por diferentes criterios, como por ejemplo, mostrar solo las tareas pendientes o solo las tareas completadas.
- **Ordenación Alfabética:** Permite ordenar las tareas alfabéticamente por su descripción, tanto en orden ascendente como descendente.
- **Ordenación por Fecha:** Implementa la posibilidad de ordenar las tareas por fecha de creación o fecha de vencimiento, en orden ascendente o descendente.
- Además de cumplir con los requisitos básicos, se espera que el componente TaskList sea eficiente, tenga una interfaz de usuario intuitiva y se adhiera a las mejores prácticas de desarrollo frontend.

Manejo de Estado:

- Utiliza el estado local del componente para almacenar la lista de tareas y su estado de completado.
- Asegúrate de manejar adecuadamente las actualizaciones de estado al completar una tarea.

Estilo:

- Agrega un estilo básico al componente para que sea estéticamente agradable.
- Cambia el estilo de una tarea completada para indicar visualmente que está completada (por ejemplo, tachado o cambio de color).

Buenas Prácticas:

- Utiliza buenas prácticas de desarrollo en JavaScript/React.
- Escribe código limpio, modular y fácil de entender.
- Siéntete libre de agregar comentarios si crees que es necesario para explicar tu código.
- **Descripción de los Campos:**

id: Identificador único de la tarea.

description: Descripción de la tarea.

dueDate: Fecha de vencimiento de la tarea.

completed: Indica si la tarea está completada o no. Es un valor booleano, donde true representa que la tarea está completada y false que está pendiente.

Entregables:

- Para los estilos, se pueden utilizar SCSS con módulos o styled components.
- Instrucciones claras sobre cómo ejecutar y probar el componente.
- Repositorio Público: Se requiere un enlace al repositorio público que contenga todo el código necesario para el proyecto. El repositorio debe estar correctamente organizado, incluyendo una estructura clara de carpetas y archivos que reflejen la arquitectura del software.
- Dockerfile: Se espera la entrega de un Dockerfile que contenga todas las configuraciones necesarias para construir y ejecutar el servicio en un contenedor Docker de manera eficiente. El Dockerfile debe incluir todos los pasos requeridos para la instalación de dependencias, la configuración del entorno y la puesta en marcha del servicio.

Notas Importantes:

- Puedes utilizar Create React App u otras herramientas según tu preferencia.
- No es necesario preocuparse por la persistencia de datos; puedes simular la lista de tareas como un array local.
- No dudes en agregar cualquier mejora o característica adicional que creas que demostrará tu habilidad en React.