TAYROSOFT
Innovación en su máxima expresión

901 226 102
(+57) 3166550933
info@tayrosoft.com
Santa Marta, Colombia

Prueba técnica: Aplicación TODO con Node.js y React.js

TAYROSOFT
Innovación en su máxima expresión

901 226 102
(+57) 3043571431
info@tayrosoft.com
Santa Marta, Colombia

Santa Marta, miércoles, 26 de febrero de 2024.

Sres.
CANDIDATOS
Santa Marta, Magdalena.

REF: PRUEBA TÉCNICA: APLICACIÓN TODO CON NODE.JS Y REACT.JS

Cordial saludo,

A través de la presente les agradecemos su interés por hacer parte de nuestra organización, como parte del proceso de selección, nos gustaría solicitarte que completes una prueba técnica que te permitirá demostrar tus habilidades en Node.js y React.js.

En este documento encontrarás una descripción detallada de la prueba técnica, que consiste en crear una aplicación de gestión de tareas (TODO) utilizando Node.js para el backend y React.js para el frontend. La prueba incluye requisitos funcionales y técnicos que te guiarán en el desarrollo de la aplicación.

Una vez que hayas completado la prueba, por favor sube el código completo de la aplicación a un repositorio público en GitHub y comparte el enlace con nosotros a través del grupo de Microsoft teams.

Estamos emocionados por ver tu trabajo y evaluar tus habilidades en el desarrollo de aplicaciones web. Si tienes alguna pregunta sobre la prueba o necesitas aclaración sobre algún punto, no dudes en contactarnos.

¡Muchas gracias por tu interés en unirte a nuestro equipo! Esperamos con entusiasmo revisar tu prueba técnica y avanzar en el proceso de selección.

Atentamente:

JULIO CÉSAR MORENO MANJARRES
COO y Cofundador de Tayrosoft S.A.S

901 226 102
(+57) 3043571431
info@tayrosoft.com
Santa Marta, Colombia

1 Instrucciones

La prueba consiste en crear una aplicación de gestión de tareas (**TODO**) utilizando **Node.js** para el backend y **React.js** para el frontend.

La aplicación debe permitir a los usuarios crear, leer, actualizar y eliminar tareas. Utiliza una base de datos para almacenar las tareas. Puedes utilizar cualquier base de datos que prefieras (por ejemplo, MongoDB, PostgreSQL, SQLite, etc.).

Crea un repositorio en Git para la aplicación y realiza commits periódicos a medida que avanzas en el desarrollo.

Sube el código completo de la aplicación, incluyendo tanto el backend como el frontend, a un repositorio público en GitHub y comparte el enlace con nosotros.

1.1 Requerimientos funcionales

- ✓ La aplicación debe mostrar una lista de tareas pendientes al cargar la página principal.
- ✓ Los usuarios deben poder agregar nuevas tareas ingresando un título y una descripción.
- ✓ Cada tarea debe tener opciones para editar y eliminar.
- ✓ Los usuarios deben poder marcar una tarea como completada.
- ✓ Debe haber una opción para filtrar las tareas según su estado (pendientes, completadas, todas).

1.2 Requerimientos técnicos

1.2.1 Backend (Node.js):

1.2.1.1 API RESTful con Express.js

- ✓ Utiliza el framework Express.js para crear una API RESTful que maneje las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) de las tareas.
- ✓ Define rutas para cada operación CRUD, siguiendo las convenciones de RESTful API.
- ✓ Utiliza los métodos HTTP adecuados (GET, POST, PUT, DELETE) para cada tipo de operación

1.2.1.2 Interacción con la base de datos

- ✓ Utiliza un ORM (Object-Relational Mapping) o ODM (Object-Document Mapping) para interactuar con la base de datos.

901 226 102
(+57) 3043571431
info@tayrosoft.com
Santa Marta, Colombia

- ✓ Si decides utilizar MongoDB como base de datos, puedes usar Mongoose como ODM.
- ✓ Si prefieres una base de datos relacional como PostgreSQL o SQLite, puedes utilizar Sequelize como ORM.

1.2.1.3 Validación de datos

- ✓ Implementa validación de datos en el backend para garantizar la integridad de los datos.
- ✓ Asegúrate de manejar correctamente los errores de validación y devolver respuestas HTTP adecuadas en caso de errores.

1.2.2 Frontend (React.js):

1.2.2.1 Componentes de React

- ✓ Utiliza componentes de React para crear la interfaz de usuario de la aplicación.
- ✓ Divide la interfaz en componentes reutilizables y modulares para mejorar la mantenibilidad y la escalabilidad del código.

1.2.2.2 Gestión de estado

- ✓ Utiliza el estado local de los componentes de React para almacenar y gestionar el estado de la aplicación.
- ✓ Considera el uso de Context API o Redux si la aplicación requiere un manejo más complejo del estado o si necesita compartir estado entre múltiples componentes.

1.2.2.3 Peticiones HTTP

- ✓ Utiliza Axios u otra librería para realizar peticiones HTTP al backend y consumir la API RESTful que has creado.
- ✓ Gestiona correctamente las respuestas y los errores de las peticiones HTTP, mostrando mensajes de error al usuario cuando sea necesario.

1.2.2.4 Diseño responsive

- ✓ Implementa un diseño responsive utilizando CSS puro, CSS frameworks como Bootstrap o librerías de componentes como Material-UI.
- ✓ Asegúrate de que la aplicación sea accesible y se vea bien en diferentes dispositivos y tamaños de pantalla.

1.2.3 Extra (Opcional)

1.2.3.1 Autenticación de usuarios

901 226 102
(+57) 3043571431
info@tayrosoft.com
Santa Marta, Colombia

- ✓ Implementa un sistema de autenticación de usuarios utilizando JSON Web Tokens (JWT) u otro método seguro.
- ✓ Permite que los usuarios inicien sesión, se registren y gestionen su perfil de usuario.
- ✓ Asegura las rutas protegidas para que solo los usuarios autenticados puedan acceder a ciertas funcionalidades.

1.2.3.2 Funcionalidades adicionales

- ✓ Agrega funcionalidades adicionales según tu creatividad y las necesidades de la aplicación.
- ✓ Por ejemplo, podrías añadir etiquetas de color para las tareas, funcionalidades de búsqueda y filtrado avanzado, notificaciones en tiempo real, etc.

Nota Legal

JULIO CÉSAR MORENO MANJARRES
COO y Cofundador de Tayrosoft S.A.S