Prueba Técnica para Desarrollador Full Stack Junior

Objetivo:

Desarrollar una pequeña aplicación que permita a las comercializadoras listar y agregar operaciones.

Agregar documentación mediante un archivo README.md, donde se añada cómo instalar y desplegar la aplicación, así como comentarios que hayan ido surgiendo en el desarrollo, tomas de decisiones y el motivo de dichas decisiones.

Los lenguajes y frameworks especificados en la cabecera son obligatorios,sin embargo, las tecnologías adicionales, podéis utilizar las que prefiráis; sin embargo, os recomiendo considerar las que utilizamos en Sercomgas, aunque su uso es opcional.

Front-end (React + TypeScript):

Tareas:

1. Crear una interfaz de usuario: Debe incluir un formulario para añadir operaciones y una lista para mostrar las operaciones existentes.
2. Conectar con el back-end: Utilizar fetch o Axios para obtener y enviar datos al servidor.
3. Opcional: Implementar React Query (Sercomgas).

Back-end (Node + TypeScript):

Tareas:

1. Configurar un servidor básico: Puede ser un servidor Express.js o Fastify (Sercomgas).
2. Rutas API:
   - GET /operations: Devuelve todas las operaciones.
   - POST /operations: Permite agregar una nueva operación.
3. Conexión a PostgreSQL: Implementar funciones para consultar y modificar la base de datos. Usando un ORM como TypeORM (Sercomgas)

Base de Datos (PostgreSQL):

Tareas:

1. Diseñar un esquema simple:
    a. Ejemplo, tabla marketers (comercializadoras)
        i. id
        ii. name
        iii. created_at
        iv. updated_at
    b. Ejemplo, tabla operations:
        i. id
        ii. marketer_id (comercializadora que crea el contrato)
        iii. client_id (comercializadora con la que comercia)
        iv. type (si es compra o venta)
        v. amount (Cantidad de gas)
        vi. price (precio al que se vende)

Evaluación:

Funcionalidad: La aplicación debe funcionar según lo previsto.

Código: Claridad, uso adecuado de TypeScript, y estructura del proyecto.

Extra: Cualquier mejora adicional como Docker, testing, o mejoras en la UI serán valoradas pero no son obligatorias.

Notas:

Se valorará la funcionalidad básica por encima de las características adicionales. Es importante destacar que una prueba que esté incompleta pero incluya características avanzadas como Docker o testing se considerará menos favorablemente que una prueba que, aunque no tenga estas características adicionales, esté completa y cumpla con la funcionalidad básica requerida. Prioriza el desarrollo de las funciones esenciales y evita invertir tiempo en detalles que no sean cruciales para los objetivos de la prueba.

La entrega se realizará proporcionando un enlace a un repositorio de GitHub o GitLab.