Challenge - Truora

Antes de iniciar recuerda
Si vas a poner el código en un repositorio público, **no uses el nombre de Truora en ninguna parte.**

Este es un **reto de aprendizaje** y no esperamos que conozcas las tecnologías necesarias para resolverla. Esperamos que puedas aprender y aplicar.

Es normal en Truora enfrentarse a desafíos de los que no sabemos mucho o a veces nada. Esta prueba tiene como objetivo presentar un reto similar en el que los ingenieros tienen que investigar una solución e implementarla en un periodo corto de tiempo.
Ésta prueba también te sirve cómo candidato para saber si es un trabajo que te va a gustar o no.

Si quedas seleccionado pagaremos $200 dólares por la prueba

Definición del problema

Crear una interfaz para buscar información de bases de datos correos.
La primera parte es indexar la base de datos de prueba y la segunda construir una interfaz para consultarla.

Parte 1: Indexar base de datos de correo electrónico

Primero descargar la base de datos de correos de Enron Corp:
http://www.cs.cmu.edu/~enron/enron_mail_20110402.tgz (423MB)
Después escribe un programa que indexe sus contenidos en la herramienta ZincSearch: https://zincsearch.com/

Por ejemplo:

```
$ ./indexer enron_mail_20110402
```

Parte 2: Profiling

Hazle profiling a tu indexer: https://go.dev/doc/diagnostics#profiling

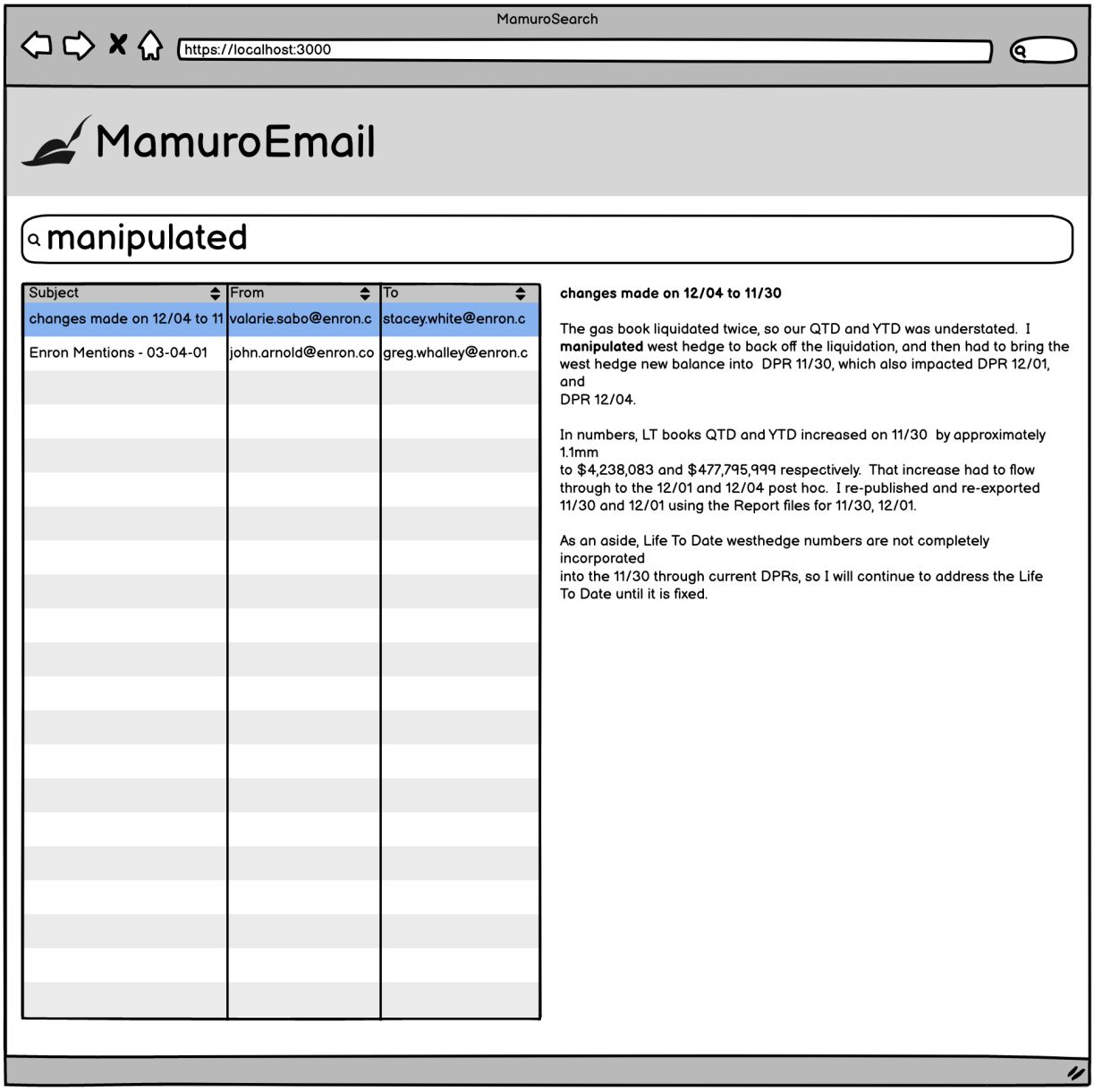
Genera el gráfico para analizarlo durante la sustentación.

Parte 3: Visualizador

Crea una interfaz simple para buscar los contenidos. Por ejemplo:

Por ejemplo:

```
$ ./mamuro -port 3000
Mamoru is running in http://localhost:3000
```



*este es un ejemplo, puedes hacer la interfaz como quieras.

Tecnologías

Las siguientes tecnologías **tienen** que ser usadas para resolver la prueba. Entendemos que es muy posible que no tengas experiencia usando estas tecnologías. Para nosotros, es muy importante medir la capacidad de aprendizaje que tiene un/a ingeniero/a. Esto además, ayuda a entender de mejor forma el talento.

Lenguaje Backend: Go
Base de Datos: ZincSearch

API Router: chi (pronunciado kai)
Interfaz: Vue 3
CSS: Tailwind

Si usas Windows, te recomendamos instalar
https://docs.microsoft.com/en-us/windows/wsl/install

Notas

1. El API puede ser REST
2. El diseño de UI/UX es libre
3. La prueba debe ser realizada por una sola persona y sustentada en vivo
4. No buscamos una solución absolutamente perfecta, buscamos la solución de cada persona respecto a su nivel. Si tu nivel es de Senior esperamos ver una solución de nivel Senior.
5. Limite de tiempo: 4 semanas (tiempo promedio: 1 semana)
6. Se puede seguir mejorando la prueba hasta el dia de la sustentación
7. Como mínimo debes presentar lo que se piden en este documento pero puedes agregar cualquier funcionalidad extra que quieras
8. Durante la sustentación vamos a revisar lo que hiciste corriendo en tu computador
9. Durante la sustentación vamos a hacer múltiples preguntas para medir tu capacidad de análisis y respuesta
10. Presentar el código terminado no garantiza que vas a ser seleccionado.
11. Si usas Windows, te recomendamos instalar https://docs.microsoft.com/en-us/windows/wsl/install
12. Puedes encontrar componentes de ejemplo de tailwind aquí: https://tailwindcomponents.com/components

Envío de tu prueba

- Una vez finalizada la prueba, envía el link de tu repositorio en GitHub a hiring@truora.com de la siguiente manera: Software Developer + tu nombre.

Recuerda

Si vas a tomar... no manejes.

Y si vas a poner el código en un repositorio público, **no uses el nombre de Truora en ninguna parte.**