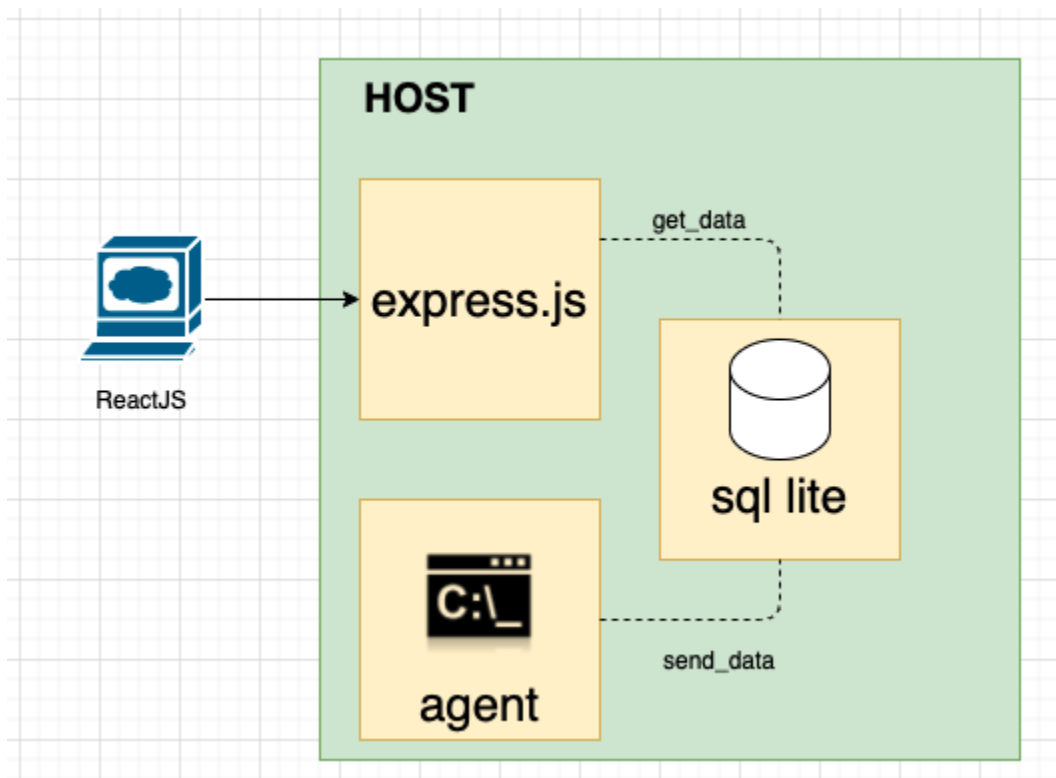


Full-stack Engineer

En esta prueba técnica se espera que pueda realizar en un día. No se espera que realice todo, si no que se espera saber cómo el programador se comporta, prioriza el desarrollo y como aplica buenas prácticas, en esta prueba se le pedirá realizar un pequeño proyecto basado en el framework ReactJS a nivel de frontend, con un backend en NodeJS con el framework ExpressJS, base de datos en SQLite 3 y un agente en NodeJS para enviar data hacia a la DBO para que sea poblada.

Modelo referencial:



Base de datos

1. Crear un modelo entidad relación del siguiente problema:

Seguridad ciudadana preocupada con el alza de la delincuencia, desea realizar un sistema que permita asegurar la seguridad en algunas entidades bancarias. Para ello informa algunos datos que resultan de interés:

- De cada banco se conoce su código, nombre. Los bancos pueden tener más de una sucursal dentro de la región.
- De las sucursales se conoce la dirección y el código de sucursal. Cada sucursal cuenta con 5 o más empleados, de los que se conoce su rut, nombre y edad. Cada sucursal tendrá asignado un vigilante que se preocupó de cuidar la seguridad del local. Un vigilante puede estar asociado a una o más sucursales, pero una sucursal solo tendrá un vigilante. Los datos de interés del vigilante son. Rut, nombre, edad. fecha de contratación.
- Dentro de las acciones que debe realizar el vigilante se encuentran realizar un reporte de la situación diaria de cada sucursal. Llamar a carabineros en caso de ver situaciones de riesgo, reportar incidente menor al Banco.
- Por otro lado cada empleado que ingrese a la sucursal fuera de horario debe registrarse en el sistema, realizando la misma acción cuando se retire de la sucursal. Si detecta anomalías dentro de la sucursal podrá realizar la observación correspondiente en el sistema.
- Tanto los empleados como los vigilantes deben autenticarse para poder ingresar al sistema.

2. Realizar 5 queries al modelo, utilizando distintas cláusulas y operadores (inner join, left join, etc).

Javascript

1. **Crear una rutina que permita ordenar números (ascendente y descendente). Ejemplo**

input: 10, 30, 9, 7, 15

output: 7, 9, 10, 15, 30

2. **Crea una función recursiva que sea capaz de encontrar un número dentro de una colección.**

```
const numbers = [1,3,6,90,39,4]
```

```
console.log(searchNumber(39,..... )
```

NodeJS

1. Crea un agente que envíe datos del host (uuid, memoria, nombre del host..)

- Usa EventEmitter que deberá tener siguientes métodos:
 - connect
 - disconnect
 - addMetric
 - removeMetric
- Usa SQLite 3 para insertar/eliminar datos.
- El agente deberá enviar datos cada 1 segundo
- El agente deberá desconectarse a los 30 min después que se haya iniciado.
(debe estar en un archivo de configuración MIN_DISCONNECT=30)

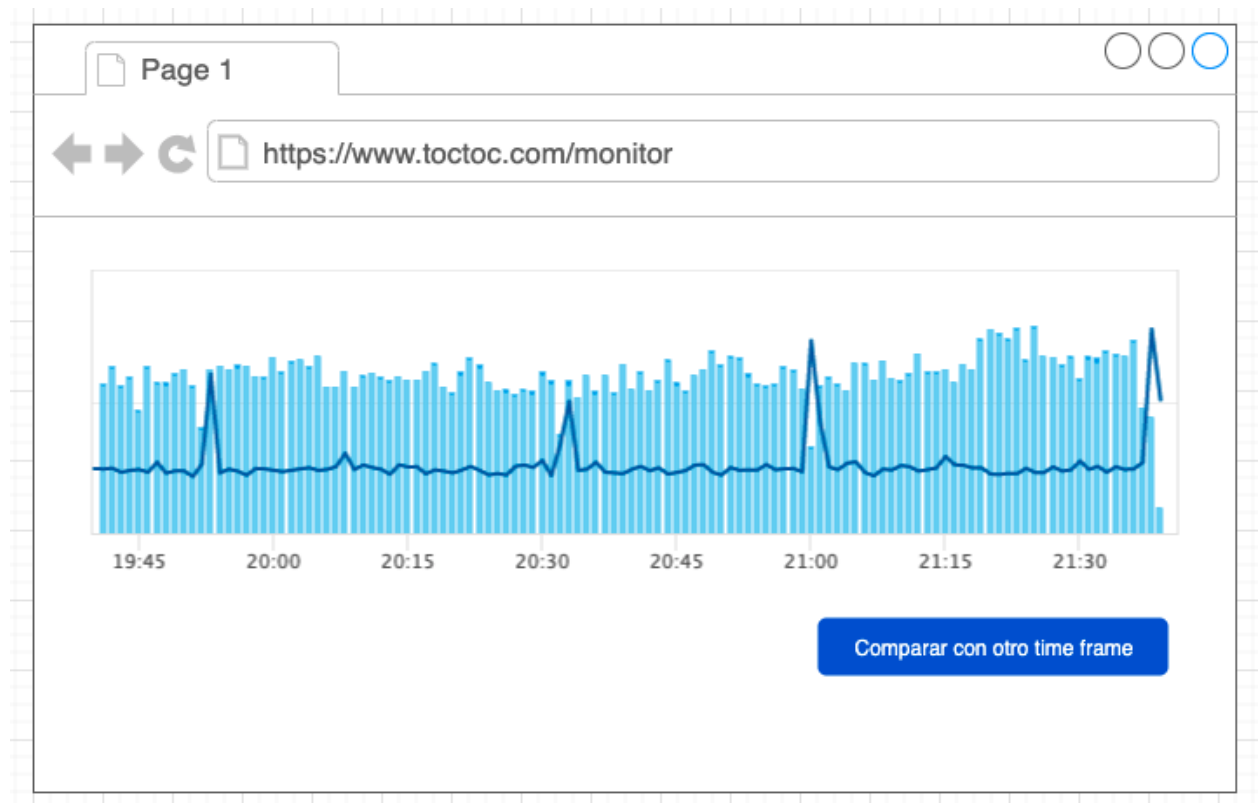
2. Crea un endpoint llamado /get/resource/memory

ReactJS

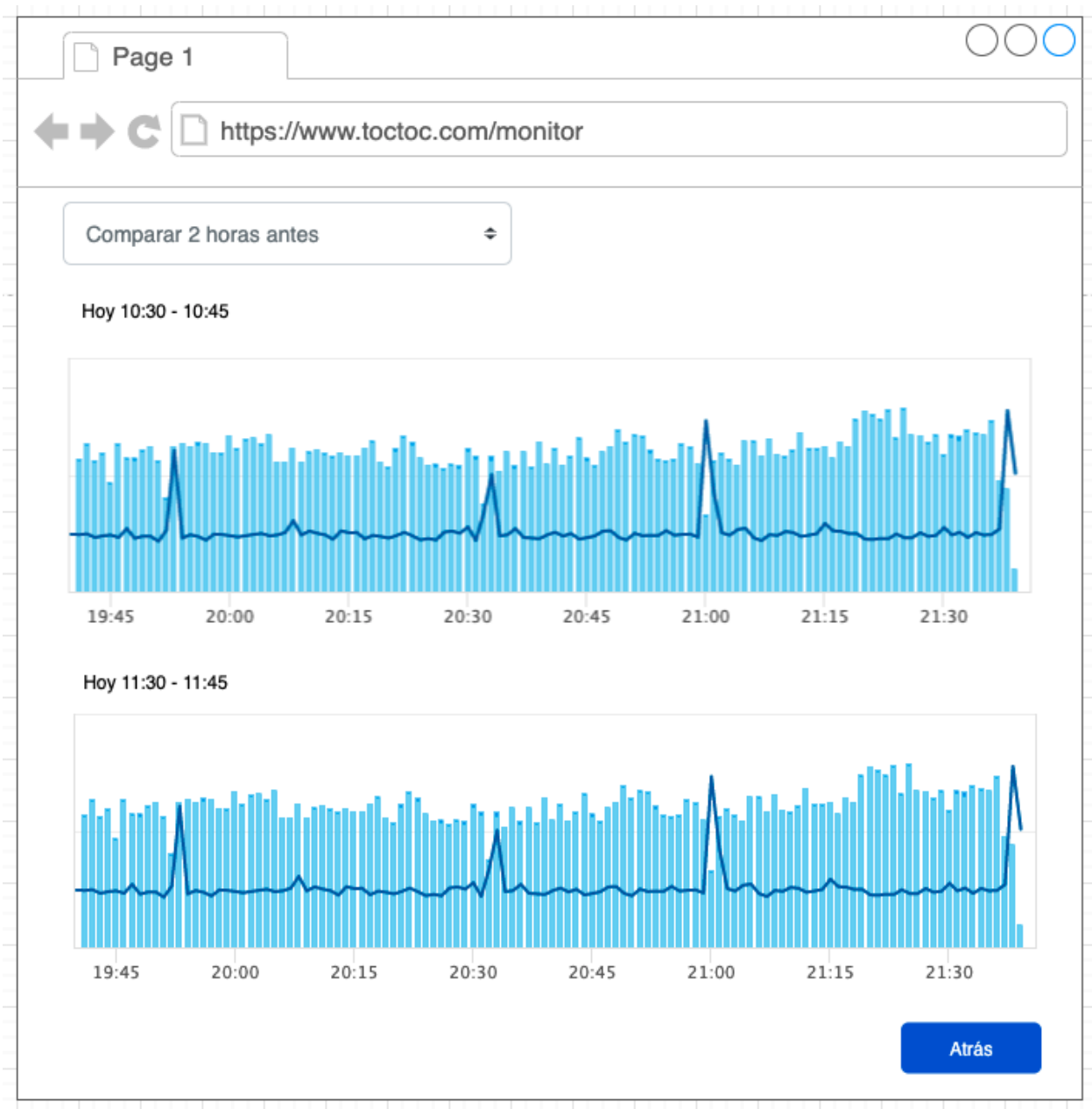
1. En esta página se visualizarán los últimos 30 minutos de los datos ya enviados por el agente.

Usa una librería a tu gusto para visualizar los datos enviados.

Como por ejemplo: <https://www.chartjs.org/>



2. En esta página debes comparar dos times frame ya sea de 1, 2, 3 horas o minutos.



3. Usa un setInterval para llamar al server y refrescar los estados de los gráficos.