
Tokko Broker

Av. del Libertador, 7208
Argentina (Buenos Aires)

Panel de Brokers de Bienes Raíces

10 de febrero del 2022

MOTIVACIÓN

Se debe construir un panel web de administración que permita la gestión mínima de brokers y propiedades de bienes raíces de forma sencilla.

Este ejercicio servirá para evaluar los conocimientos de los aplicantes a las oportunidades frontend de la organización y para introducir a los ingresantes al stack tecnológico que utilizamos en el equipo de ingeniería de TokkoBroker.

FUNCIONALIDADES

El panel debe permitir:

1. Listar los brokers existentes.
2. Crear nuevos Brokers.
3. Listar las propiedades existentes.
4. Crear nuevas propiedades a los brokers.

INTERFACES EXTERNAS

Para efectuar las operaciones requeridas en las funcionalidades descritas anteriormente, se proveerá una API GraphQL que permitirá cumplir con los requerimientos que será adjuntada en conjunto con este documento. Esta API se encuentra escrita en NodeJS y para poder instalarla y ejecutarla, se deben seguir las instrucciones que se encuentran en el archivo README.md del paquete.

Para identificar las queries y mutaciones necesarias para realizar las operaciones, debe correr la aplicación y buscar los docs que la interfaz de GraphQL expone.

REQUERIMIENTOS

1. Debe estar construido en cualquier framework/librería orientada a componentes(Preferentemente React + Typescript, este último es opcional).
2. Puede utilizar librerías de componentes que faciliten la creación de visuales tales como Material UI, jopi.js (librería de tokko que utilizamos para crear nuestras aplicaciones), o cualquier otra.
3. Debe consumir la API provista.

No será evaluado el diseño, por lo que a las pantallas respecta, siéntase libre de disponer la información que desee de la forma que desee.

RESOLUCIÓN Y ENTREGA

El proyecto se debe ser subido a cualquier repositorio git de carácter público y enviar el link a **bsabatino@navent.com**

Temas a evaluar

- Arquitectura del proyecto.
- Buenas prácticas de desarrollo.
- Legibilidad y calidad de código.
- Patrones de diseño(si aplica).
- Utilización de librerías orientadas a componentes.
- Routing(opcional).
- Apollo + GraphQL.
- Manejo de estado de aplicación

Valorables

- Manual de ejecución.
- Documento que justifique las decisiones tomadas.
- Utilización de docker & docker compose.
- Utilización de la librería jopi.js (<https://www.npmjs.com/package/@oneloop/jopijs>)