

# Prueba de Candidatos Full-Stack Middle/Senior

En esta prueba analizaremos tus conocimientos generales e incluso tu velocidad de desarrollo. A continuación te explicaremos todo lo que será necesario. La prueba general se divide en pequeños ejercicios con los cuales se evaluará en práctica:

- Pensamiento lógico
- Resolución de problemas
- Tiempo de desarrollo
- Entre otros

Se incluyen bonos extra que podrán ser tomados en cuenta más no son obligatorios.

## Ejercicios Backend

### 1. Importación de data.

#### PROBLEMA:

El siguiente archivo .csv ([resource\\_accommodation.csv](#)) contiene una serie de datos relacionados con el comercio inmobiliario. Ejemplo (Dirección del apartamento/piso, Metros cuadrados, Características, entre otros) que deben residir en una tabla en base de datos MySQL.

#### REQUERIMIENTO:

El objetivo principal es crear un API endpoint en NodeJS (preferiblemente ExpressJS o NestJS) el cual sea capaz de leer el archivo .csv e importar la data a la tabla de la base de datos. **Utilizar NodeJS es mandatorio.**

BONO EXTRA: Proteja el endpoint utilizando JWT

### 2. Filtrar data.

#### PROBLEMA:

Basado en el ejercicio #1 ya tenemos una base de datos funcional. Ahora necesitamos poder filtrar la data.

## REQUERIMIENTO:

Se requiere un endpoint método GET el cual permita pasar atributos para poder filtrar el resultado de la data por:

1. Rango de precio mínimo y máximo.
2. Número de habitaciones.

BONO EXTRA: Proteja el endpoint utilizando JWT

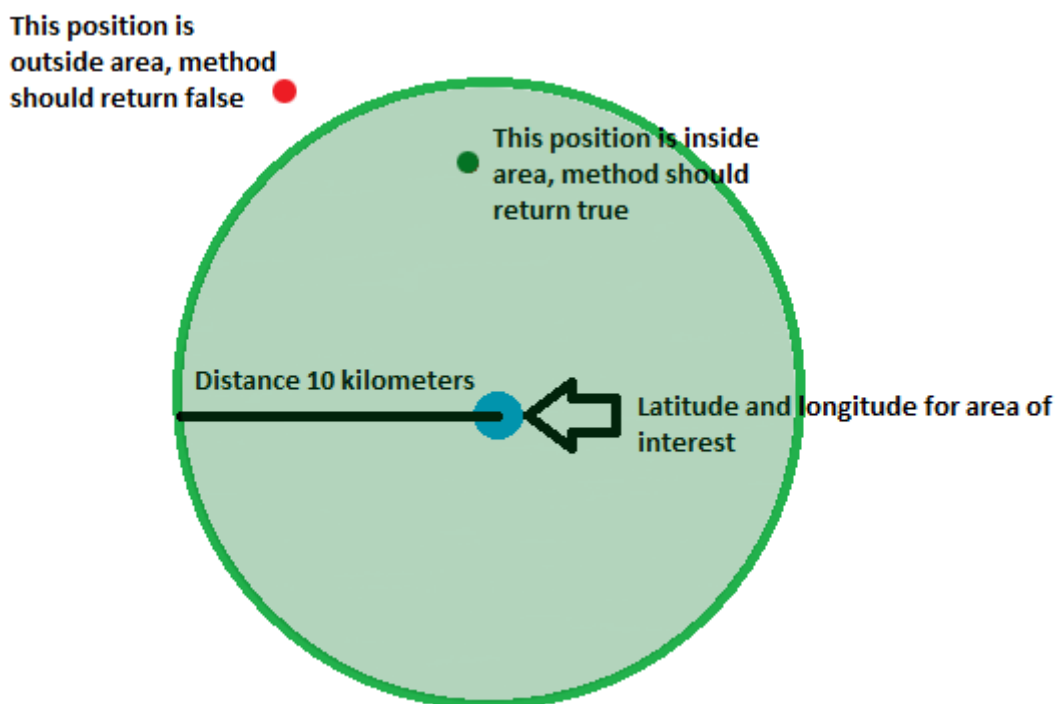
## 3. Procesar data.

### PROBLEMA:

En algunos casos necesitamos saber el precio del alquiler por zona. Para ello necesitamos procesar la información de nuestra base de datos.

### REQUERIMIENTO:

Se necesita endpoint método GET en el cual se pasen 3 atributos (Latitud, Longitud, Distancia en Kms = X), y esta retorne el precio promedio del metro cuadrado dentro de un radio de X kilómetros de la latitud y longitud indicados. Ver imagen:



BONO EXTRA: Provea un endpoint GET el cual utilice los mismos parámetros del ejercicio, pero devuelva una lista de las propiedades en formato JSON, que se encuentren dentro del área establecida.

## 4. Reportes.

### PROBLEMA:

En ocasiones se necesita generar reportes para el área administrativa.

### REQUERIMIENTO:

Se requiere un endpoint al cual se pasen los atributos de filtro, coordenadas y tipo de reporte (PDF, CSV) y dicho reporte generado se guarde en una carpeta.

# Ejercicios Frontend

Los siguientes ejercicios deben ser realizados utilizando React. Tratan sobre hacer una GUI protegida por autenticación de usuario/contraseña en donde se podrán cargar archivos .csv para alimentar la base de datos de propiedades y luego hacer consultas/filtros. Se puede utilizar cualquier framework (Preferiblemente NextJS) o librería adicional a **React** que considere conveniente para su desarrollo. **Lo único mandatorio es utilizar React.**

Se calificará presentación, formato y responsiveness.

## 1. Autenticación de Usuarios

### PROBLEMA:

Se debe contar con una tabla en la base de datos con los usuarios y contraseñas para autenticación del uso de la GUI.

### REQUERIMIENTO:

Crear una tabla con los usuarios y sus respectivas contraseñas y hacer una página de inicio de sesión para proteger todas las páginas subsiguientes en la GUI.

## 2. Despliegue de GUI

### REQUERIMIENTO:

Crear una GUI en donde el usuario que inició sesión exitosa pueda realizar las siguientes tareas:

- Cargar un archivo .csv para alimentar la lista de propiedades
- Ver una lista completa de propiedades y filtrarlas basado en los siguientes campos:
  - Cantidad de habitaciones
  - Rango de precio (mínimo y máximo)
  - Ubicación basada en rangos de kilómetros de un punto de referencia (zona perimetral)
  - Metraje de la propiedad
  - Posee balcón
  - Pet friendly
  - Piscina
  - Jardin
- Descargar un CSV o PDF con los resultados aplicados en el filtro

**BONO EXTRA:** Desplegar un mapa con las propiedades del filtro marcadas dentro del mismo

Cualquier duda estoy a las órdenes en telegram [@silviovo](#) o en el correo [silvio.orozco@voxdatacomm.com](mailto:silvio.orozco@voxdatacomm.com)

---

Revision #9

Created 20 April 2023 16:52:25 by Alejandro Mejia

Updated 27 February 2024 14:31:42 by Silvio Orozco