



PROYECTO AIRBNB

API REST de alojamientos turísticos

IES Monte Naranco

Proyecto Airbnb – 2º DAM

API REST de alojamientos turísticos

Descripción

A partir de la base de datos proporcionada:

- Crear y mapear las clases necesarias para representar el modelo.
- Desarrollar un API con SpringBoot que dé respuesta a los métodos que se piden más abajo.
- Desarrollar una capa de acceso a datos mediante repositorios que den servicio al API REST.
- Implementar seguridad mediante JWT en los métodos que así lo requieran.

Características

- La práctica podrá ser realizada de forma individual o por parejas.
- Se requerirá la realización de una demo o defensa de la práctica para mostrar el trabajo realizado.
- El proyecto debe estar subido a Github. Ahí se reflejará un histórico de commits y versiones que refleje el trabajo de los participantes.
- El plazo de realización de la práctica será hasta el 4 de febrero a las 23:59 horas.

CREACIÓN DE LA BASE DE DATOS

Se proporciona un script de creación de base de datos que creará 4 tablas:

- **Alojamientos:** para la información relacionada con los hoteles, apartamentos turísticos o pisos compartidos procedentes de Airbnb. Consta de:
 - Id: debe ser autogenerado por la base de datos.
 - Nombre: texto con el nombre descriptivo del alojamiento.
 - Latitud y longitud: permite ubicar el alojamiento en el mapa.
 - Precio: precio por noche de la estancia.
 - Puntuación: puntos concedidos por los usuarios (sobre 5).
 - Habitaciones: número de habitaciones del alojamiento.
 - Camas: número de camas disponibles.
 - Baños: número de baños disponibles.
 - Aseos: número de aseos disponibles.
 - Propietario_id: id del propietario del alojamiento.
 - Barrio_id: id del barrio donde se encuentra el alojamiento.
 - tipoHabitacion_id: id del tipo de alojamiento.
- **Propietarios:** son las personas dueñas del alojamiento. Constan de:
 - Id: se debe guardar el proporcionado en el fichero CSV.
 - Nombre: nombre de la persona o personas propietarias.
- **Barrios:** lista de barrios de NY donde hay alojamientos. Constan de:
 - Id: autogenerado por la base de datos.

- Nombre: nombre del barrio.
 - Distrito_id: nombre del distrito donde se encuentra el barrio.
- **Distritos:** lista de los distritos de NY donde hay alojamientos. Constan de:
 - Id: autogenerado por la base de datos.
 - Nombre: nombre del distrito.
- **TiposHabitacion:** lista con los tipos de alojamientos existentes. Constan de:
 - Id: autogenerado por la base de datos.
 - Nombre: nombre descriptivo del tipo de habitación o alojamiento.



CARGA DE LA INFORMACIÓN

Se proporciona un fichero para cargar la información correspondientes a los alojamientos de distintos distritos de NY en las distintas tablas.

CREACIÓN DEL API

El API deberá hacer una correcta gestión de los errores. Tendrá los siguientes métodos:

GET /distrito

Obtiene todos los distritos (id y nombre) presentes en la base de datos.

GET /alojamiento

Retorna un listado de todos los alojamientos existentes en la base de datos. Además, se incluirá al propietario de cada alojamiento y el tipo de habitación.

POST /alojamiento

Este método estará protegido mediante JWT.

Insertará un nuevo alojamiento con los siguientes campos:

- Nombre.
- Id del propietario.
- Id barrio.
- Latitud.
- Longitud.
- Id del tipo de habitación.
- Precio.
- Puntuación.
- Número de habitaciones.
- Número de camas.
- Número de baños.

PUT /alojamiento

Este método estará protegido mediante JWT.

Actualizará un nuevo alojamiento con los siguientes campos:

- Id.
- Nombre.
- Id del propietario.
- Id barrio.
- Latitud.
- Longitud.
- Id del tipo de habitación.
- Precio.
- Puntuación.
- Número de habitaciones.
- Número de camas.
- Número de baños.

GET /alojamiento/distrito/{numero}

Retorna un listado de todos los alojamientos presentes en un determinado distrito. Por ejemplo:

GET /alojamiento/distrito/1

Retornará todos los alojamientos en el distrito 1.

GET /alojamiento/distrito/{numero}/min/{numero}/max/{numero}

Retorna un listado de todos los alojamientos presentes en un determinado distrito en el rango de precios definido por min y max. Por ejemplo:

GET /alojamiento/distrito/1/min/500/max/1000

Retornará todos los alojamientos en el distrito 1 con precio entre 500 y 1000 (ambos incluidos).

GET /alojamiento/distrito/{numero}/habitaciones/{numero}

Devuelve todos los alojamientos presentes en un determinado distrito con un número de habitaciones determinado. Por ejemplo:

GET /alojamiento/distrito/1/habitaciones/2

Retornará todos los alojamientos en el distrito 1 con 2 habitaciones.

GET /alojamiento/distrito/{numero}/habitaciones/{numero}/banos/{numero}

Devuelve todos los alojamientos presentes en un determinado distrito con un número de habitaciones y baños determinado. Por ejemplo:

GET /alojamiento/distrito/1/habitaciones/2/banos/1

Retornará todos los alojamientos en el distrito 1 con 2 habitaciones y 1 baño.

Además, se deben implementar otros dos métodos de filtro libres que consideres que pueden tener interés para un usuario buscando alojamiento en Nueva York.

INTERFAZ DE USUARIO

Se proporciona una sencilla interfaz de usuario en HTML+CSS+JS. La interfaz incorpora un mapa creado con la librería Leaflet, que es una librería JavaScript para la creación de mapas.

Cuando carga la página, se producen dos peticiones al API. Por un lado, se hace una petición GET /distrito, para obtener todos los distritos y así, rellenar los desplegados con dichos valores. El código que se encarga de hacer la petición y crear los <option> para asignar a ambos desplegados está en la función rellenarDesplegadosDistrito() de main.js.

Por otro lado, se hace una petición GET /alojamiento para obtener todos los alojamientos presentes en la base de datos. Dado que la respuesta es un poco pesada (muchos datos), tarda unos segundos en llegar al cliente, de ahí que el mapa aparezca inicialmente con un icono de “cargando”.

Una vez resueltas las dos peticiones anteriores, el cliente ya puede realizar las búsquedas que deseé. Para ello, dispone de dos formularios en la parte izquierda para probar el API:

- Por un lado, podrás buscar por distrito y rango de precio: es decir, podrá indicar un distrito, y, opcionalmente, un precio mínimo y máximo del alojamiento. Al pulsar en el

botón de “Buscar” correspondiente, se le mostrarán en el mapa solo los marcadores relativos a dichos alojamientos.

- Por otro, por distrito y número de habitaciones y baños: esta búsqueda puede ser solo por distrito, por distrito y habitaciones, o por distrito, habitaciones y baños.

Para acelerar la visualización de los puntos, y para conseguir una mayor claridad en el mapa cuando hay muchos puntos ubicados muy cerca, se utilizan clusters de puntos. Es decir, los puntos que están cerca en el mapa se juntan en nubes de puntos. Estas nubes tienen un número que indica cuántos puntos o marcadores hay en esa zona. Si el usuario hace zoom y se acerca, estas nubes se van “disipando” de manera que van descomponiéndose en marcadores. Sobre cada marcador se puede pulsar y se muestra un popup con su información.

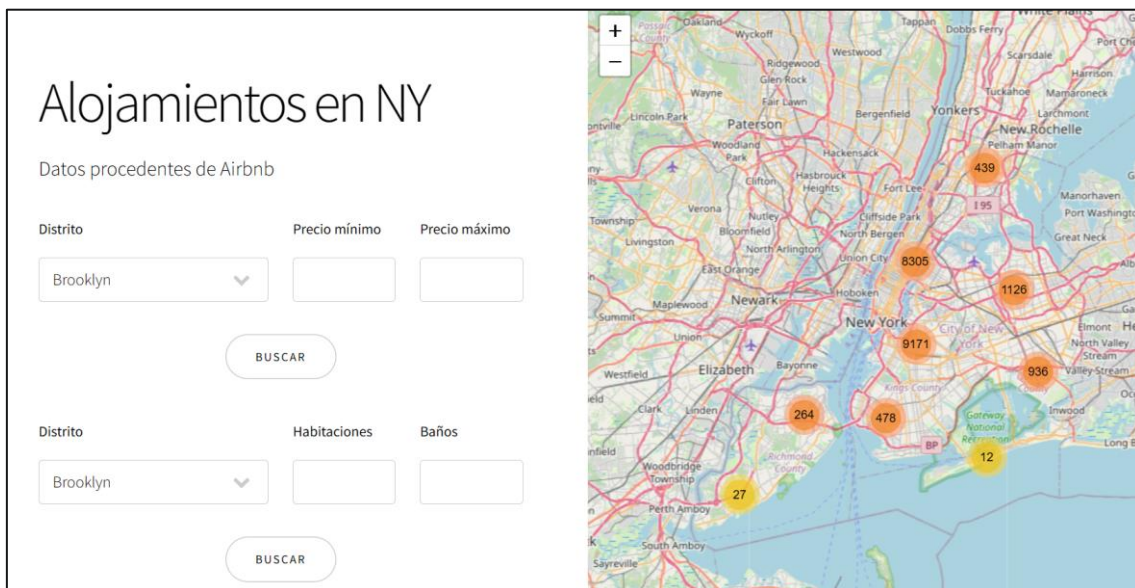


Ilustración 1. Pantalla inicial

Todos los marcadores que se pintan en el mapa tienen asociado un popup. Es decir, cuando se pulsa sobre ellos se abre un popup con información de ese alojamiento. Esa información consiste en el nombre, nombre del propietario, barrio donde se encuentra, tipo de la habitación, precio y puntuación.

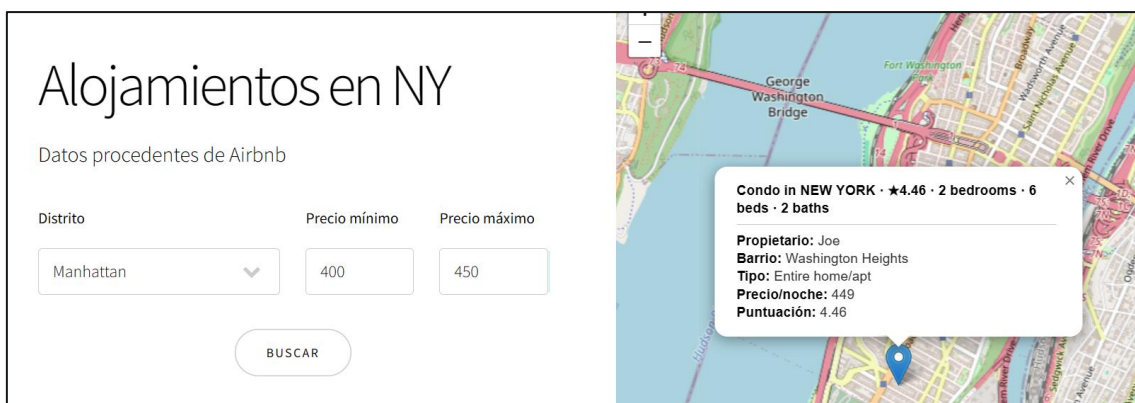


Ilustración 2. Búsqueda por distrito y precio y visualización del popup de un marcador

ENTREGA

Se debe entregar el proyecto desarrollado y un documento donde se añada:

- Nombre de los desarrolladores.
- Captura de pantalla de las pruebas generadas para cada caso.
- URL del enlace a Github.