

Aquí se muestra el diagrama de la arquitectura general del sistema para su solución de canalización de datos. Ilustra los componentes, incluida la base de datos PostgreSQL, la capa de servicio Spring Boot, los puntos finales de la API REST y las aplicaciones cliente que accederán a los datos del punto de acceso WiFi público.

**1. Fuente de datos: Datos de WiFi públicos.**

Los datos sin procesar provienen de datos abiertos proporcionados por el gobierno de la Ciudad de México, específicamente sobre puntos de acceso WiFi públicos en toda la ciudad.

Estos datos generalmente incluyen detalles como la ubicación (latitud, longitud), fecha de instalación, tipo de programa y otros atributos relevantes para cada punto de acceso WiFi.

**2. Capa de ingesta de datos.**

Esta capa maneja la importación de los datos sin procesar al sistema. Puede ser un script o un proceso ETL (Extracción, Transformación, Carga) que descarga, limpia y formatea los datos WiFi para su almacenamiento.

Los datos limpios luego se insertan en la base de datos PostgreSQL en un formato estructurado, lo que hace que sea fácil de consultar.

**3. Base de datos: PostgreSQL.**

La base de datos PostgreSQL se utiliza para almacenar los datos estructurados de puntos de acceso WiFi. La elección de PostgreSQL brinda los siguientes beneficios:

Compatibilidad con datos geoespaciales: PostgreSQL con PostGIS (una extensión para manejar objetos geográficos) es beneficioso aquí, especialmente porque uno de los requisitos es filtrar los resultados por proximidad a una coordenada determinada.

Persistencia de datos: garantiza que los datos se almacenen de forma permanente y que la API pueda consultarlos rápidamente.

**4. Backend: Spring Boot REST API.**

El núcleo de la aplicación está creado con Spring Boot, que proporciona una interfaz API RESTful para interactuar con los datos. La API consta de varios puntos finales que permiten a los usuarios recuperar datos de puntos de acceso WiFi con diferentes opciones de filtrado:

Obtener todos los puntos WiFi: un punto final paginado para enumerar todos los puntos de acceso WiFi.

Obtener punto WiFi por ID: recupera detalles de un punto de acceso WiFi específico utilizando su ID único.

Obtener puntos WiFi por colonia: filtra los puntos de acceso WiFi según el nombre de la colonia y devuelve resultados paginados.

Obtener puntos WiFi por proximidad: devuelve una lista de puntos de acceso WiFi ordenados por proximidad a una coordenada específica (latitud, longitud). Esta función utiliza consultas espaciales para calcular distancias.

Capa de servicio: contiene la lógica empresarial para procesar las solicitudes. Interactúa con la base de datos para obtener y estructurar los datos según los requisitos de la API.

Capa de repositorio: conecta la aplicación Spring Boot a la base de datos PostgreSQL mediante JPA (Java Persistence API). Esta capa maneja todas las interacciones de la base de datos y las operaciones CRUD.

**5. Puntos finales de la API.**

Estos son los puntos finales de acceso público a través de los cuales las aplicaciones cliente pueden interactuar con los datos de WiFi. Cada punto final corresponde a una consulta o filtro diferente, cumpliendo con los requisitos de la solución.

Ejemplos de puntos finales:

GET /http://localhost:8090/api/wifies/page/{page}: recupera una lista paginada de todos los puntos WiFi.

GET /http://localhost:8090/api/wifies/{id}: recupera información de un punto WiFi específico por ID.

GET http://localhost:8090/api/wifi/filtrar-colonia/page/{term}: recupera puntos WiFi filtrados por colonia.

GET /http://localhost:8090/api/wifi/proximidad?lat=19.4326&lon=-99.1332: enumera los puntos WiFi por proximidad a la latitud y longitud proporcionadas.

**6. Clientes.**

Diferentes aplicaciones cliente pueden acceder a la API para obtener información sobre puntos de acceso WiFi. Algunos clientes posibles incluyen:

Aplicación web: una interfaz que permite a los usuarios explorar y buscar puntos de acceso WiFi en un mapa o como una lista.

Aplicación móvil: una aplicación móvil donde los usuarios pueden encontrar puntos WiFi cercanos sobre la marcha.

Herramienta de administración interna: para administrar o monitorear la base de datos de puntos de acceso WiFi, aunque no es un requisito específico, podría ser útil para el mantenimiento del sistema.

**7. Seguridad y manejo de errores.**

Seguridad: si bien no se muestra explícitamente en el diagrama, lo ideal es que la API REST esté protegida para evitar el acceso no autorizado. Las prácticas comunes incluyen claves de API, OAuth2 o autenticación basada en tokens.

Manejo de errores: garantiza un manejo elegante de errores, como identificaciones no válidas, datos no disponibles o solicitudes mal formadas, con códigos de estado HTTP y mensajes apropiados.

Resumen del flujo de datos:

Los datos se ingieren y almacenan en PostgreSQL.

La aplicación Spring Boot proporciona puntos finales RESTful.

Los clientes interactúan con estos puntos finales para recuperar datos de puntos de acceso WiFi, que se paginan, se filtran por colonia o se ordenan por proximidad.

Esta configuración proporciona una arquitectura sólida, escalable y flexible para brindar datos WiFi públicos de manera eficiente, lo que permite una fácil expansión de funciones o integración con nuevos clientes.