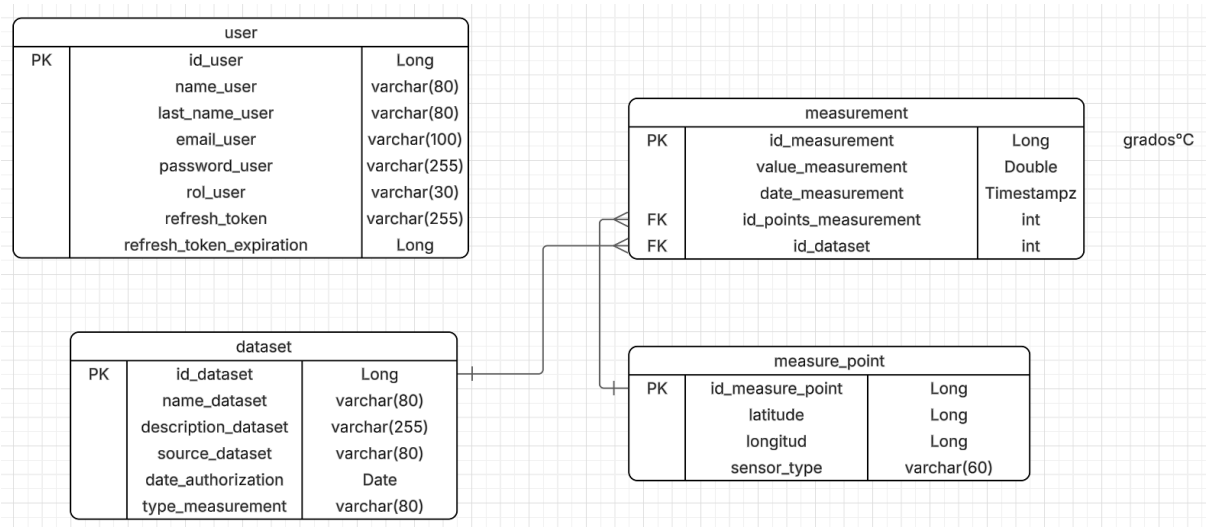
Diccionario de base de datos



user			
Columna	Tipo de Dato	Descripción	Llave
id_user	Long	Identificador único de cada usuario	Primaria
name_user	varchar(80)	Nombre del usuario	
lastName_user	varchar(80)	Apellido del usuario	
email_user	varchar(100)	Correo electrónico del usuario	
password_user	varchar(255)	Contraseña del usuario	
rol_user	varchar(30)	Rol del usuario	
refresh_token	varchar(255)	token de acceso jwt	
refrech_token_expiration	Long	determina cuándo expirará el token	

measurements			
Columna	Tipo de Dato	Descripción	Llave
id_measurement	Long	Identificador único de cada medición	Primaria
id_points_measurement	Long	Identificador de un punto de medición	Foránea
id_dataset	Long	Identificador de un dataset con mediciones	Foránea
value_measurement	Double	valor de la medición	
date_measurement	Timestampz	dia en que se tomó la medición	

measure_point			
Columna	Tipo de Dato	Descripción	Llave
id_measure_point	Long	Identificador único del punto de medición	Primaria
latitude	Long	latitud de la ubicación	
longitude	Long	longitud de la ubicación	
sensor_type	varchar(60)	tipo del sensor	

dataset			
Columna	Tipo de Dato	Descripción	Llave
id_dataset	Long	Identificador único del dataset	Primaria
name_dataset	varchar(80)	Nombre del dataset	
description_dataset	varchar(255)	Descripción del dataset	
source_dataset	varchar(80)	Fuente del dataset	
date_authorization	Date	Fecha de autorización del dataset	
type_mesurement	varchar(80)	Tipo de medición de un dataset	

Trigger: trg_calcular_anomalia_punto
Ubicación: ClimateChangeBackend/src/main/java/resources/loadData.sql

Descripción: Este trigger se encarga de calcular y notificar una "anomalía" en las mediciones cada vez que se inserta o actualiza un registro en la tabla measurements.

Detalles Técnicos:

- Evento: Se ejecuta AFTER INSERT OR UPDATE (después de insertar o actualizar).
- Tabla Afectada: measurements.
- Alcance: FOR EACH ROW (se ejecuta para cada fila afectada).
- Función Asociada: calcular_anomalia_punto().

Lógica de la Función (calcular_anomalia_punto):

1. Calcula el Promedio Histórico: Obtiene el promedio de todos los valores (value_measurement) registrados para el punto de medición (id_measure_points) de la nueva fila.
2. Calcula el Promedio del Último Año: Obtiene el promedio de los valores del mismo punto de medición, pero solo considerando los registros del último año (date_measurement >= CURRENT_DATE - INTERVAL '1 year').
3. Calcula la Anomalía: Resta el promedio histórico al promedio del último año (promedio_ultimo_anio - promedio_historico).
4. Notificación: Genera un mensaje de aviso en la base de datos (RAISE NOTICE) indicando el ID del punto y el valor de la anomalía calculada. Nota: Este resultado se muestra en la consola de la base de datos y no se guarda en una tabla persistente por defecto

Descripción de la implementación de JWT y autenticación

Se implementó un sistema de autenticación basado en JWT (JSON Web Tokens) con doble token: access token y refresh token.

El token se crea en la clase JwtUtil y almacena la siguiente información:

- subject: Rut del usuario
- firstName: Primer nombre del usuario
- lastName: Apellido del usuario
- email: Correo electrónico
- role: Rol del usuario (ROLE_USER, ROLE_ADMIN)
- authorities: Autoridades
- passwordHash: Hash de la contraseña
- issuedAt: Fecha de emisión
- expiration: Fecha de expiración (access token: 1 hora)

Algoritmo de firmado: HS512 (HMAC con SHA-512)

Autenticación

Para hacer la autenticación, se crearon los siguientes endpoints:

1. POST /api/v1/auth/register
    - Permite registrar a los nuevos usuarios
2. POST /api/v1/auth/login
   Autentica al usuario con RUT y contraseña
    - Genera dos tokens:
        1. SESSION (access token): Cookie con duración de 1 hora
        2. REFRESH (refresh token): Cooke con duración de 7 días
    - Retorna información del usuario y ambos el token en el body
3. POST /api/v1/auth/refreshtoken
    - Permite renovar el access token cuando expire
    - Valida el refresh token de la cookie
    - Genera un nuevo access token y refresh token
    - Extiende la sesión sin requerir credenciales
4. POST /api/v1/auth/logout
    - Invalida las cookies de sesión
    - Limpia el refresh token de la base de datos
    - Cierra la sesión del usuario

Seguridad

Validaciones en Login:

- Formato de RUT: ^[0-9]{7,8}-[0-9K]$
- Contraseña: Mínimo 8 caracteres
- Encriptación: BCrypt para almacenar contraseñas
- Validaciones en Register:

- RUT único y con formato válido
- Email único
- Contraseña con requisitos mínimos

Seguridad de los Tokens:

- HttpOnly: Las cookies no son accesibles desde JavaScript (protección contra XSS)
- SameSite: Lax: Protección contra CSRF
- Path específico: El refresh token solo se envía a su endpoint
- Secret Key: JWT firmado con clave secreta de 512 bits

Protección de Rutas en el Frontend:

- Rutas protegidas requieren meta: { requiresAuth: true }
- El router redirige a /login si no hay usuario autenticado
- Previene acceso a login/register si ya está autenticado