

1. Configuración y dependencias

1.1 Propiedades

- `spring.application.name=policies-api` Establece el nombre de la aplicación. Es útil para la identificación en los registros y sistemas de monitoreo.
- `server.port=8080` Define el puerto en el que se ejecutará la aplicación. El puerto por defecto es 8080, pero se puede cambiar según las necesidades.

1.1 Configuración de la Base de Datos

- `spring.datasource.url=jdbc:postgresql://localhost:5432/development`
Especifica la URL de conexión a la base de datos PostgreSQL llamada `development` en el puerto 5432 del `localhost`.
- `spring.datasource.username=${DEVELOPMENT_POSTGRES_USERNAME}`
Nombre de usuario para acceder a la base de datos, obtenido de una variable de entorno para mantener la seguridad.
- `spring.datasource.password=${DEVELOPMENT_POSTGRES_PASS}`
Contraseña para acceder a la base de datos, también almacenada en una variable de entorno.
- `spring.datasource.driver-class-name=org.postgresql.Driver` Especifica el controlador JDBC utilizado para PostgreSQL.

1.2 Configuración de JPA/Hibernate

- `spring.jpa.hibernate.ddl-auto=update` Automatiza la actualización del esquema de la base de datos según las entidades JPA definidas. El valor `update` sincroniza el esquema cada vez que se ejecuta la aplicación.
- `spring.jpa.database-platform=org.hibernate.dialect.PostgreSQLDialect`
Indica a Hibernate el dialecto SQL que debe utilizar, en este caso, PostgreSQL.

1.3 Configuración de Logging

- `logging.level.com.coppel.policies_api=ERROR` Establece el nivel de registro para el paquete `com.coppel.policies_api` en `ERROR`, lo que significa que solo se registrarán mensajes de error, reduciendo el volumen de logs.

2. Configuración de Variables de Entorno

Las variables de entorno `DEVELOPMENT_POSTGRES_USERNAME` y `DEVELOPMENT_POSTGRES_PASS` son esenciales para mantener seguras las credenciales de la base de datos. A continuación, se explica cómo configurarlas en diferentes sistemas operativos.

2.1 En macOS y Linux

- A. Abre la terminal.
- B. Ejecuta los siguientes comandos para establecer las variables de entorno temporalmente:
 - `export DEVELOPMENT_POSTGRES_USERNAME=tu_usuario`
 - `export DEVELOPMENT_POSTGRES_PASS=tu_contraseña`
- C. Para que las variables persistan en futuras sesiones, agrégalas al archivo de configuración de tu shell (`~/.bash_profile`, `~/.bashrc` o `~/.zshrc`):
 - `echo 'export
DEVELOPMENT_POSTGRES_USERNAME=tu_usuario' >>
~/.bash_profile`
 - `echo 'export DEVELOPMENT_POSTGRES_PASS=tu_contraseña'
>> ~/.bash_profile`
- D. Recarga el perfil:
 - `source ~/.bash_profile`

2.2 En Windows

- A. **Ve a Configuración del sistema:**
 - Windows 10/11: Haz clic derecho en Este equipo > Propiedades > Configuración avanzada del sistema.
- B. En la pestaña **Opciones avanzadas**, haz clic en **Variables de entorno**.
- C. En **Variables de usuario y del sistema**, haz clic en **Nuevo** y crea las variables:
 - **Nombre de la variable:** `DEVELOPMENT_POSTGRES_USERNAME`
Valor de la variable: `tu_usuario`
 - **Nombre de la variable:** `DEVELOPMENT_POSTGRES_PASS` **Valor de la variable:** `tu_contraseña`
- D. Asegúrate de cerrar y abrir nuevamente cualquier terminal o IDE para que los cambios surtan efecto.

3. Base de datos

3.1 Crear base de datos

- CREATE DATABASE development;

3.2 Crear funciones para el manejo de consultas por el servidor.

- Los archivos correspondientes se encuentran en la carpeta psq.
- Ejecutar los queries en la base de datos development

4. Descarga, compilación y Ejecución de proyecto.

4.1 Limpia y construye el proyecto con Maven:

- mvn clean install

4.2 Ejecuta el proyecto:

- mvn spring-boot:run

4.3 Verifica que la aplicación esté funcionando accediendo a:

- <http://localhost:8080>