

Introducción del Proyecto

Este proyecto es un backend desarrollado en Spring Boot que simula el sistema de registro para una planta de reciclaje. Su propósito principal es gestionar las entregas de materiales reciclables, permitiendo a los usuarios registrar sus entregas y a los administradores gestionar los datos. El sistema calcula una compensación monetaria para el usuario basada en el tipo y peso del material entregado. La API REST expone operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para las entregas y funcionalidades adicionales para generar reportes diarios o semanales.

Los componentes principales del proyecto son:

- **API REST:** Para la interacción con el sistema.
- **Capa de persistencia:** Manejada con Spring Data JPA.
- **Base de datos:** MySQL para el almacenamiento de datos.

Configuración del Entorno de Desarrollo

A continuación, se detallan los requerimientos y configuraciones para levantar el proyecto en un entorno de desarrollo local.

Dependencias clave de pom.xml

El proyecto utiliza Maven para la gestión de dependencias. Las dependencias más importantes son:

- **Spring Boot Starter Web:** Para la creación de aplicaciones web y API REST.
- **Spring Boot Starter Data JPA:** Para la persistencia de datos con JPA y Hibernate.
- **MySQL Connector/J:** Driver para la conexión con la base de datos MySQL.
- **Lombok:** Para reducir el código repetitivo en clases de modelo (getters, setters, constructores).
- **MapStruct:** Para el mapeo automático entre objetos DTO (Data Transfer Objects) y entidades JPA.
- **Springdoc OpenAPI (spring-boot-starter):** Para la generación automática de la documentación de la API en formato OpenAPI 3 y la interfaz de usuario de Swagger.

Requerimientos para configurar el proyecto en IntelliJ IDEA

1. **IDE:** IntelliJ IDEA 2025.2 o superior.
2. **JDK:** Java 21 o superior.
3. **Base de datos:** Una instancia de MySQL en ejecución.

4. **Clonar el repositorio:** Obtener el código fuente desde el repositorio en GitHub.
5. **Instalar dependencias:** Abrir el proyecto en IntelliJ IDEA y permitir que Maven descargue todas las dependencias definidas en el pom.xml.

Configuración de la base de datos MySQL en application.properties

El archivo src/main/resources/application.properties contiene la configuración necesaria para conectar la aplicación con la base de datos MySQL.

code Properties

downloadcontent_copy

expand_less

```
spring.datasource.url=jdbc:mysql://localhost:3306/reciclaje_db
spring.datasource.username=root
spring.datasource.password=tu_contraseña
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# Configuración de JPA y Hibernate
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

# Inicialización de la base de datos
spring.sql.init.mode=always
```

- **spring.datasource.url:** La URL de conexión a la base de datos. Asegúrate de que el nombre de la base de datos (reciclaje_db) exista en tu servidor MySQL.
- **spring.datasource.username y spring.datasource.password:** Credenciales de acceso a la base de datos.
- **spring.jpa.hibernate.ddl-auto=update:** Hibernate actualizará el esquema de la base de datos automáticamente si detecta cambios en las entidades.
- **spring.sql.init.mode=always:** Spring Boot ejecutará los scripts SQL (schema.sql y data.sql si existen en src/main/resources) al iniciar la aplicación.

Arquitectura del Proyecto

El proyecto sigue una arquitectura en capas, lo que facilita la separación de responsabilidades y el mantenimiento del código.

- **Controller:** Esta capa es responsable de manejar las solicitudes HTTP, procesar las entradas y devolver las respuestas al cliente. Utiliza DTOs para recibir y enviar datos.

- **Service:** Contiene la lógica de negocio principal de la aplicación. Es invocada por los controladores y se comunica con la capa de repositorio para acceder a los datos.
- **Repository:** Define las interfaces para interactuar con la base de datos. Spring Data JPA implementa automáticamente estas interfaces.
- **DTOs (Data Transfer Objects):** Son objetos planos que se utilizan para transferir datos entre las capas, especialmente entre el cliente y el controlador. Esto evita exponer las entidades de la base de datos directamente en la API.

API REST

La API permite realizar operaciones CRUD sobre las entregas y generar reportes.

Descripción de los Endpoints de Entregas

- **GET /entregas:** Obtiene un listado de todas las entregas registradas.
- **GET /entregas/{id}:** Obtiene una entrega específica por su ID.
- **POST /entregas:** Crea una nueva entrega.
- **PUT /entregas/{id}:** Actualiza una entrega existente.
- **DELETE /entregas/{id}:** Elimina una entrega.
- **GET /entregas/reporte:** Genera un reporte de entregas, ya sea diario o semanal, según los parámetros proporcionados.

Generación de Reportes

La funcionalidad de reportes permite obtener un resumen de las entregas en un intervalo de tiempo específico.

- **Endpoint:** GET /entregas/reporte
- **Parámetros de la solicitud (Query Params):**
 - **fecha:** La fecha de referencia para el reporte en formato YYYY-MM-DD.
 - **tipo:** El tipo de reporte a generar. Puede ser diario o semanal.

Ejemplo de Uso (curl)

- **Reporte Diario:** Para obtener las entregas del 6 de octubre de 2025.

code Bash

downloadcontent_copy

expand_less

```
curl -X GET "http://localhost:8080/entregas/reporte?fecha=2025-10-06&tipo=diario"
```

- **Reporte Semanal:** Para obtener las entregas de la semana que contiene el 6 de octubre de 2025.

code Bash

downloadcontent_copy

expand_less

```
curl -X GET "http://localhost:8080/entregas/reporte?fecha=2025-10-06&tipo=semanal"
```

Validación de Entradas

Spring Boot, a través de la dependencia spring-boot-starter-validation, permite validar los datos de entrada en las solicitudes. En los DTOs, se pueden utilizar anotaciones de Jakarta Bean Validation (como @NotNull, @Size, @Min) para definir las reglas de validación. Si una solicitud no cumple con estas reglas, la API devolverá una respuesta 400 Bad Request con los detalles de los errores de validación.

1. [GitHub - gc130041/ProyectoFinal_PlantaDeReciclaje at development](https://github.com/gc130041/ProyectoFinal_PlantaDeReciclaje)