

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

📌 **GET:** <http://localhost:8080/api/saludo?lang=es>

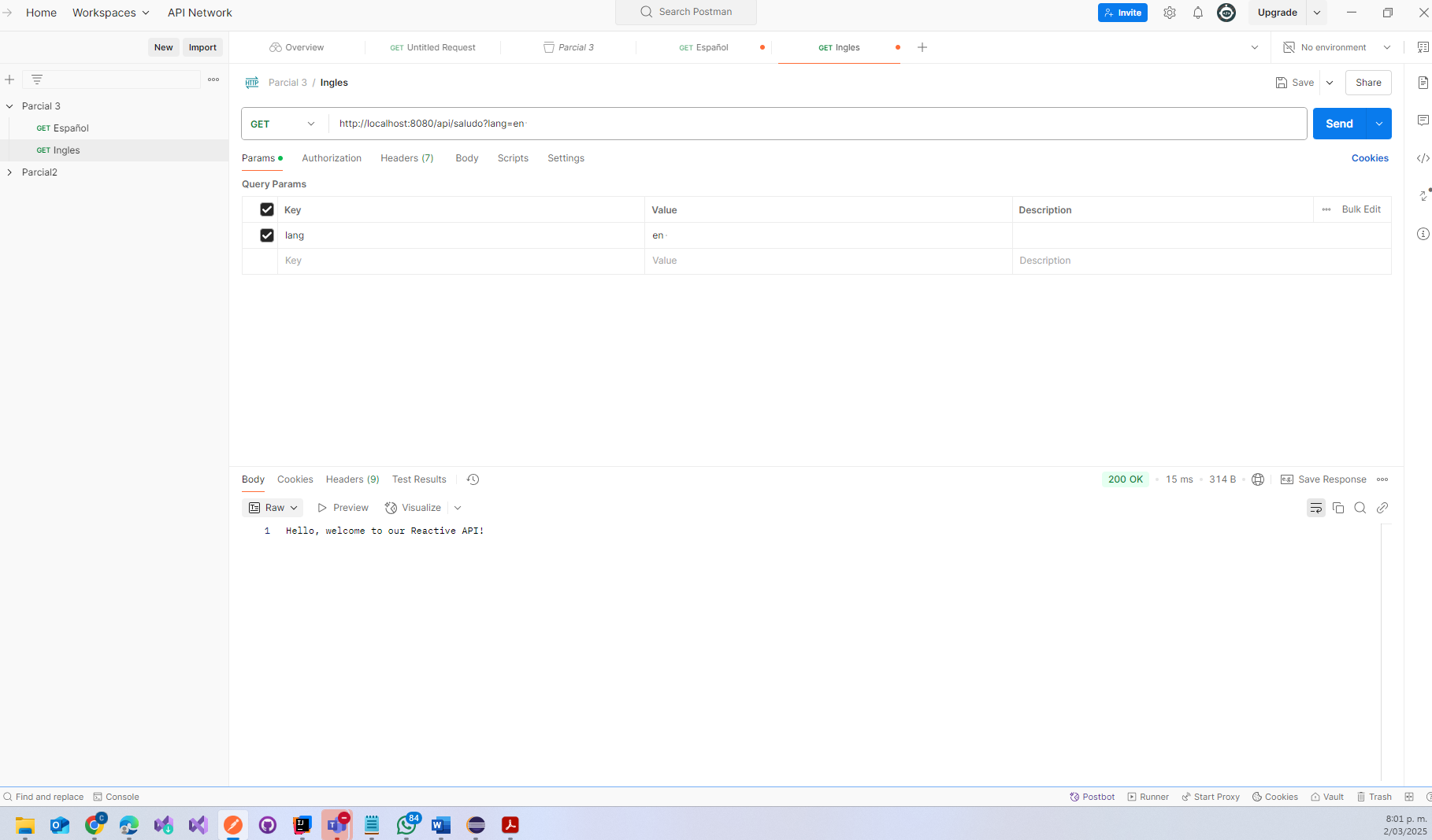
📌 **Respuesta:** "Hola, bienvenido a nuestra API Reactiva!"

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

📌 **GET:** http://localhost:8080/api/saludo?lang=en

📌 **Respuesta:** "Hello, welcome to our Reactive API!"



Interfaz de usuario gráfica, Texto, Aplicación

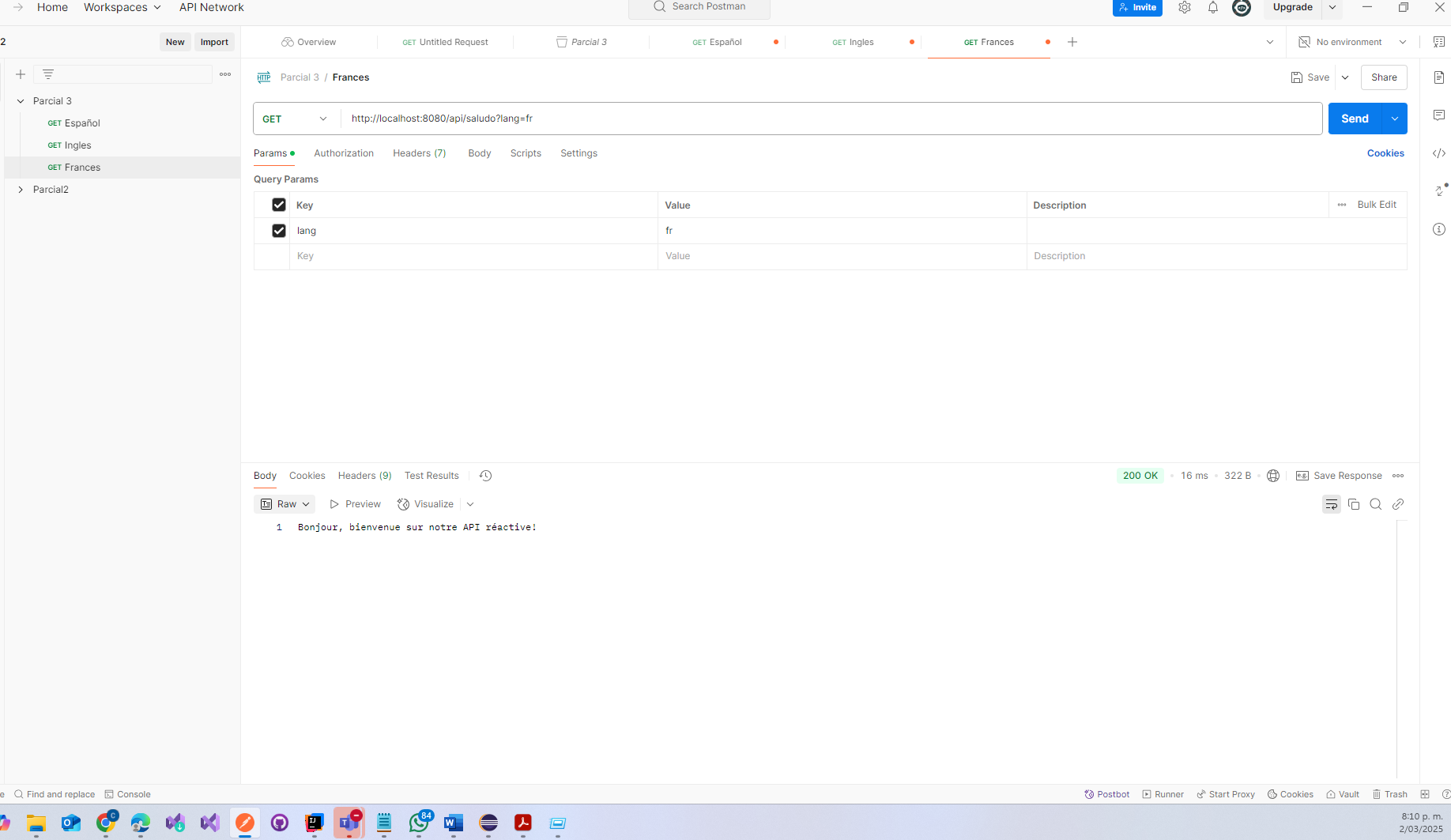
El contenido generado por IA puede ser incorrecto.

📌 **GET:** http://localhost:8080/api/saludo?lang=fr

📌 **Respuesta:** Bonjour, bienvenue sur notre API réactive!

Interfaz de usuario gráfica, Texto, Aplicación

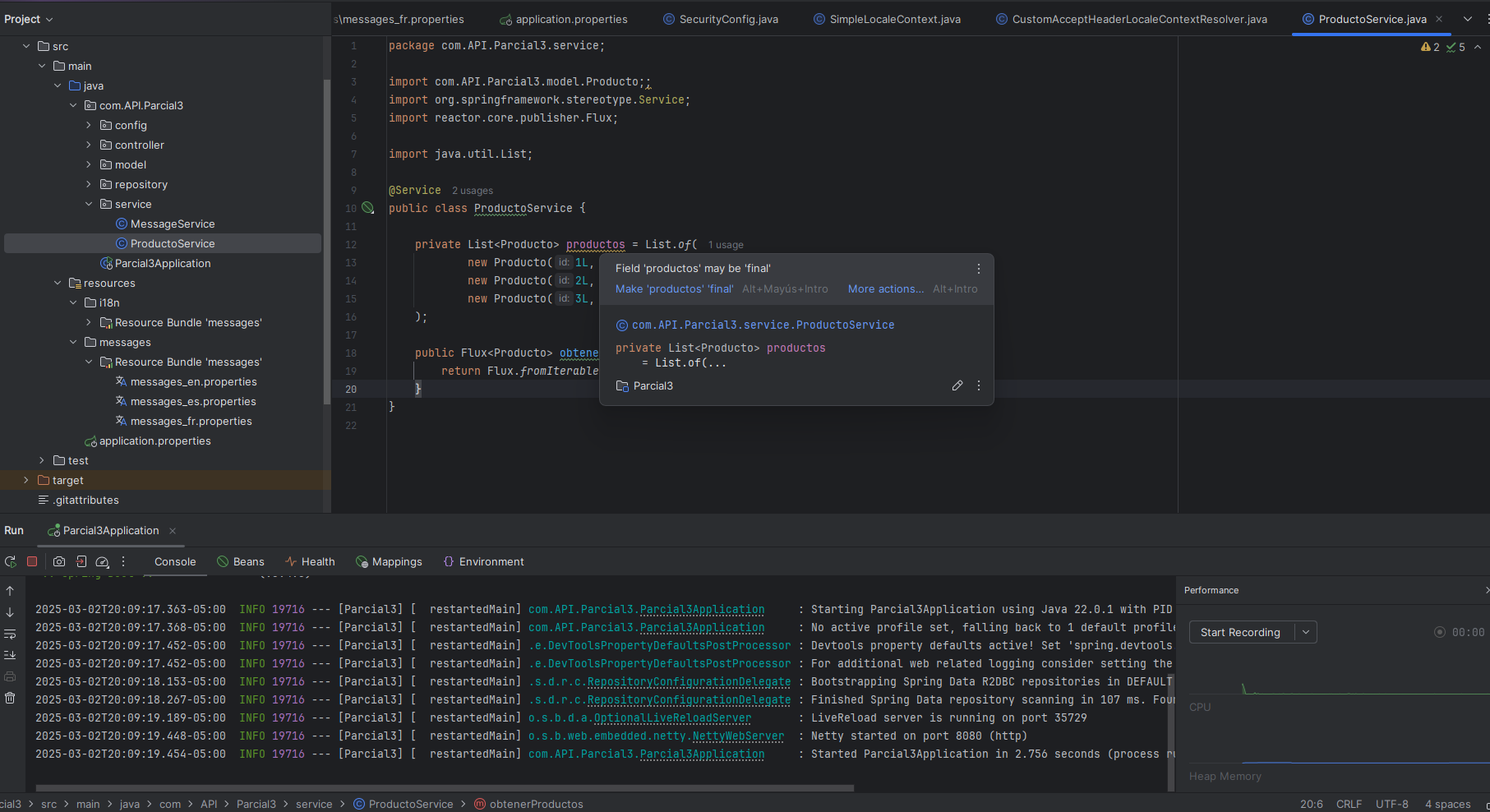
El contenido generado por IA puede ser incorrecto.

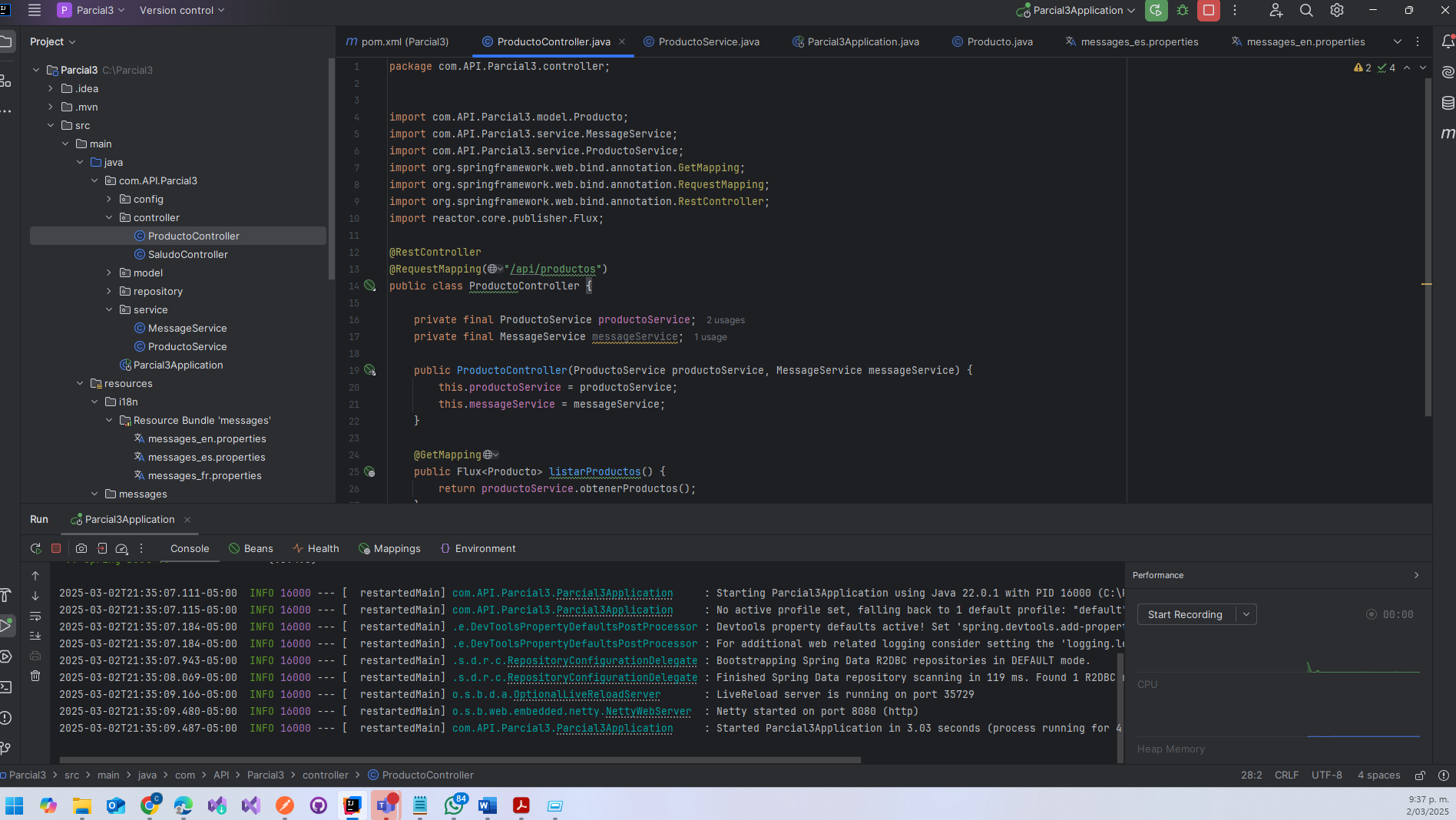


Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**5. Extensión: Agregar un Servicio Reactivo con Base de Datos**





Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

📌 Reto Final:

*Crea un servicio de pedidos con WebFlux que permita crear y listar pedidos en diferentes idiomas.*

**Clase Pedido (Asegurar constructores y Lombok)**

package com.API.Parcial3.model;

import jakarta.validation.constraints.NotBlank;

import jakarta.validation.constraints.NotNull;

import jakarta.validation.constraints.PositiveOrZero;

import lombok.Data;

import lombok.NoArgsConstructor;

import java.time.LocalDateTime;

import java.util.List;

import java.util.UUID;

@Data

@NoArgsConstructor

public class Pedido {

private String id;

@NotBlank(message = "{cliente.requerido}")

private String cliente;

@NotNull(message = "{productos.requeridos}")

private List<Producto> productos;

@PositiveOrZero(message = "{total.invalido}")

private double total;

private LocalDateTime fecha;

// Constructor para crear pedidos manualmente (sin ID ni fecha)

public Pedido(String cliente, List<Producto> productos) {

this.cliente = cliente;

this.productos = productos;

calcularTotal(); // Calcula el total, pero no genera ID ni fecha

}

// Método para inicializar ID y fecha ANTES de guardar el pedido

public void inicializarPedido() {

this.id = UUID.randomUUID().toString();

this.fecha = LocalDateTime.now();

}

public void calcularTotal() {

this.total = productos.stream()

.mapToDouble(Producto::getPrecio)

.sum();

}

}

**Clase PedidoService (Generar ID y fecha antes de guardar)**

@Service

public class PedidoService {

private final CopyOnWriteArrayList<Pedido> pedidos = new CopyOnWriteArrayList<>();

public Mono<Pedido> crearPedido(Pedido pedido) {

pedido.inicializarPedido(); // Genera ID y fecha

pedidos.add(pedido);

return Mono.just(pedido);

}

public Flux<Pedido> listarPedidos() {

return Flux.fromIterable(pedidos);

}

}

**Controlador PedidoController (Llamar a inicializarPedido)**

@PostMapping

public Mono<ResponseEntity<String>> crearPedido(

@RequestBody Pedido pedido,

@RequestHeader(name = "Accept-Language", required = false) Locale locale

) {

pedido.calcularTotal();

return pedidoService.crearPedido(pedido)

.map(p -> ResponseEntity.ok(

messageSource.getMessage("pedido.creado", null, locale)

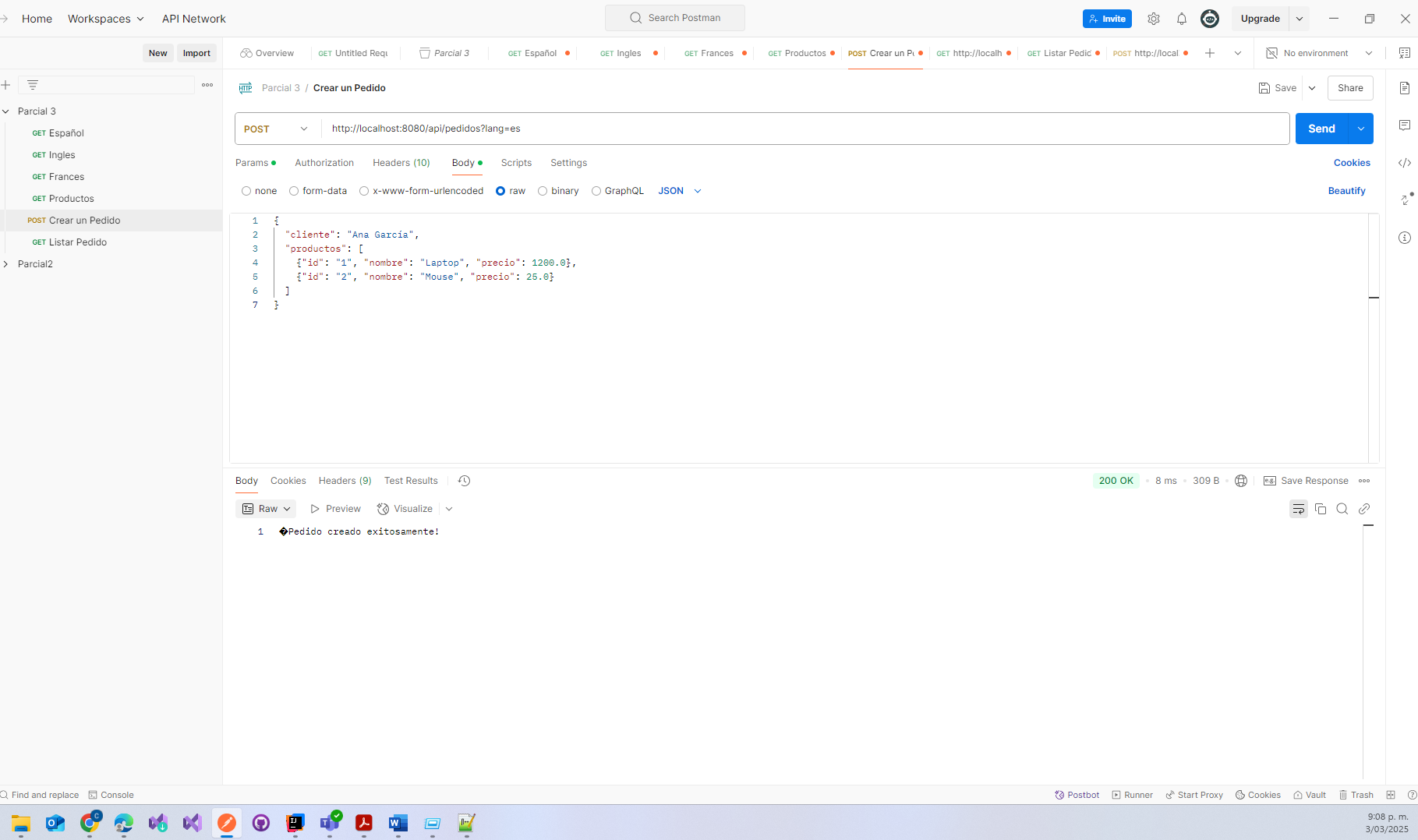
));

}

# Pruebas de Creación de Pedidos en Diferentes Idiomas

**A. Español (lang=es)**

* **Método: POST**
* **URL: http://localhost:8080/api/pedidos?lang=es**
* **Body (JSON):**



**B. Inglés (lang=en)**

* **Método**: POST
* **URL**: http://localhost:8080/api/pedidos?lang=en

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

**C. Francés (lang=fr)**

* **Método**: POST
* **URL**: http://localhost:8080/api/pedidos?lang=fr
* **Body**: Mismo JSON que arriba.
* **Respuesta Esperada**:

"Commande créée avec succès!" Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**2. Pruebas de Listado de Pedidos en Diferentes Idiomas**

**A. Español (lang=es)**

* **Método**: GET
* **URL**: http://localhost:8080/api/pedidos?lang=es
* **Respuesta (JSON)**:

