UNIVERSIDAD EUROPEA DEL ATLÁNTICO

GRADO EN

Ingeniería Informática

Sistemas Distribuidos y Programación en Paralelo

Proyecto Final:

Sistema de Gestión de Eventos

Trabajo realizado por:

Brenda Lopes Ventura de Souza

Manuela Grizoni Smelan

Sistema de Gestión de Evento

Requisitos básicos para el funcionamiento

Este proyecto consiste en desarrollar una aplicación para la gestión de eventos. Se utiliza un backend en C# usando el framework .NET que expone una API REST, que se comunica con JSON, para manejar eventos, inscripciones y recursos, usando una base de datos en MySQL Workbench y Postman para probar los endpoints.
Se utiliza un frontend en HTML para la interacción del usuario, que tiene un formulario para registrar eventos, inscripción de usuarios y visualización de eventos disponibles.

Gestión de Eventos

Se puede crear, editar y listar eventos. Cada evento tiene informaciones como nombre, fecha, ubicación y capacidad máxima. lista de participantes

Inscripción de Participantes

Los usuarios pueden registrarse en un evento mediante un formulario, y los datos registrados se guardan en una base de datos.

Base de Datos

Todas las informaciones se quedan almacenadas en una base de datos, esto permite gestionar y consultar la información de forma eficiente

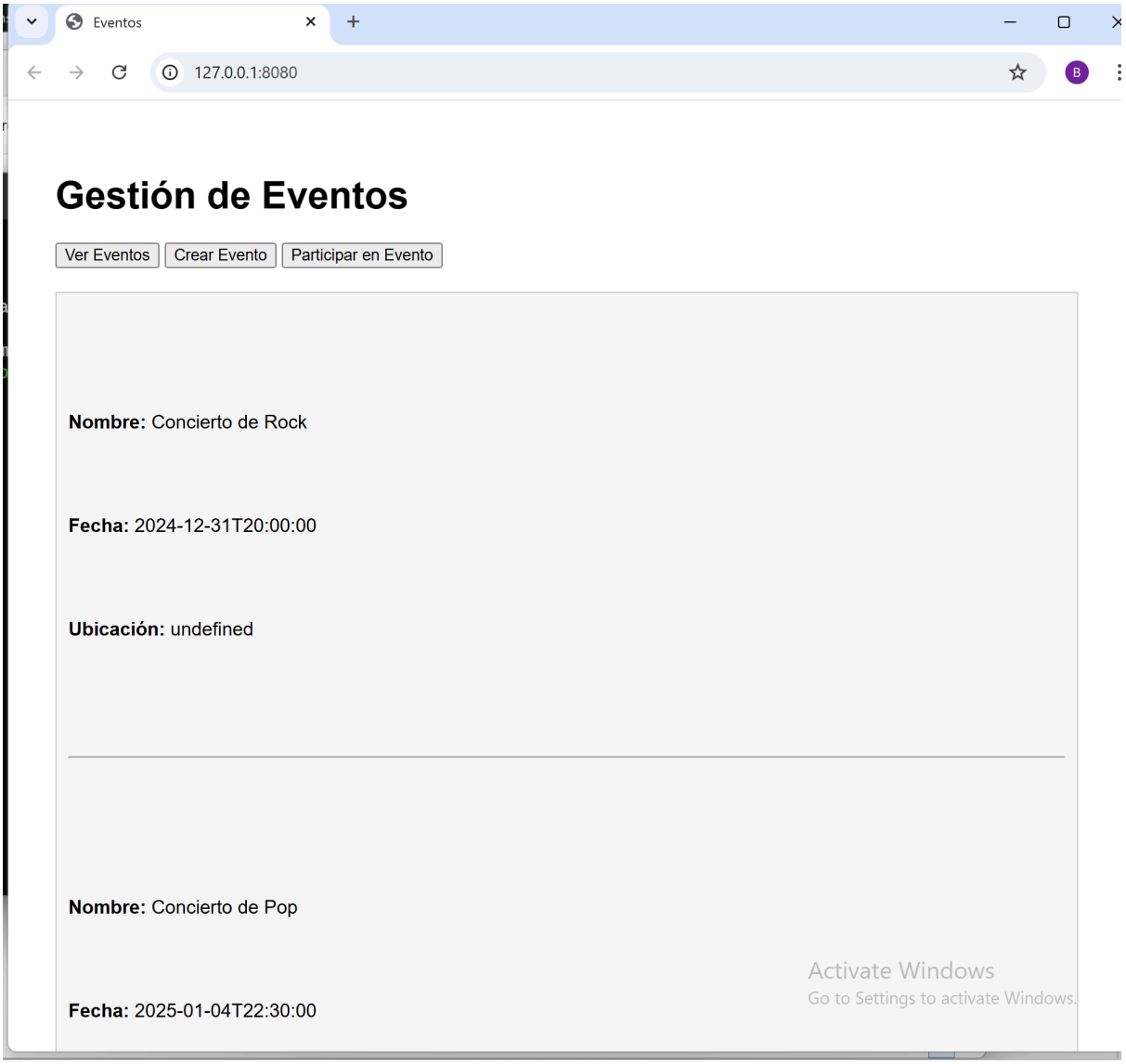
Interfaz de Usuario

Creación de la Interfaz de Usuario para el Proyecto

Se diseñó una interfaz de usuario sencilla utilizando **HTML**, **CSS** y **JavaScript puro** para interactuar con la API REST desarrollada en .NET Core. El objetivo fue crear un frontend funcional que permitiera realizar las operaciones básicas de gestión de eventos.

1. **Herramientas Utilizadas**:
    - **HTML/CSS**: Para estructurar y estilizar la interfaz.
    - **JavaScript**: Para realizar solicitudes HTTP (`fetch`) a la API.
    - **live-server**: Para servir el frontend localmente.
2. **Funcionalidades Implementadas**:
    - **Ver Eventos**: Un botón que realiza una solicitud `GET` a la API y lista los eventos en formato `Nombre`, `Fecha` y `Ubicación`.

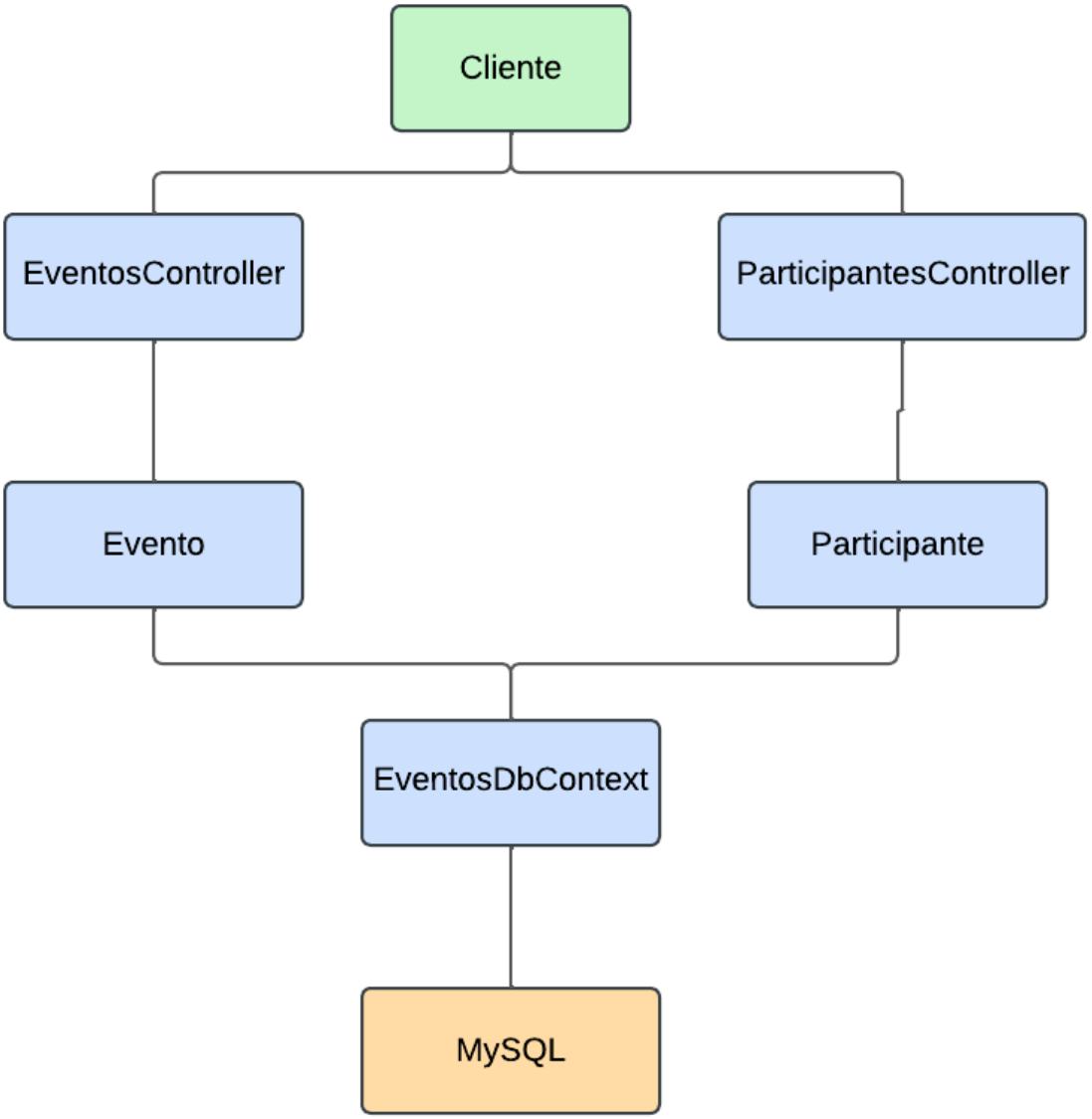
- **Crear Evento**: Un formulario que envía los datos del nuevo evento mediante una solicitud `POST` a la API.
- **Participar en Evento**: Un formulario para registrar un participante en un evento específico utilizando una solicitud `POST`.

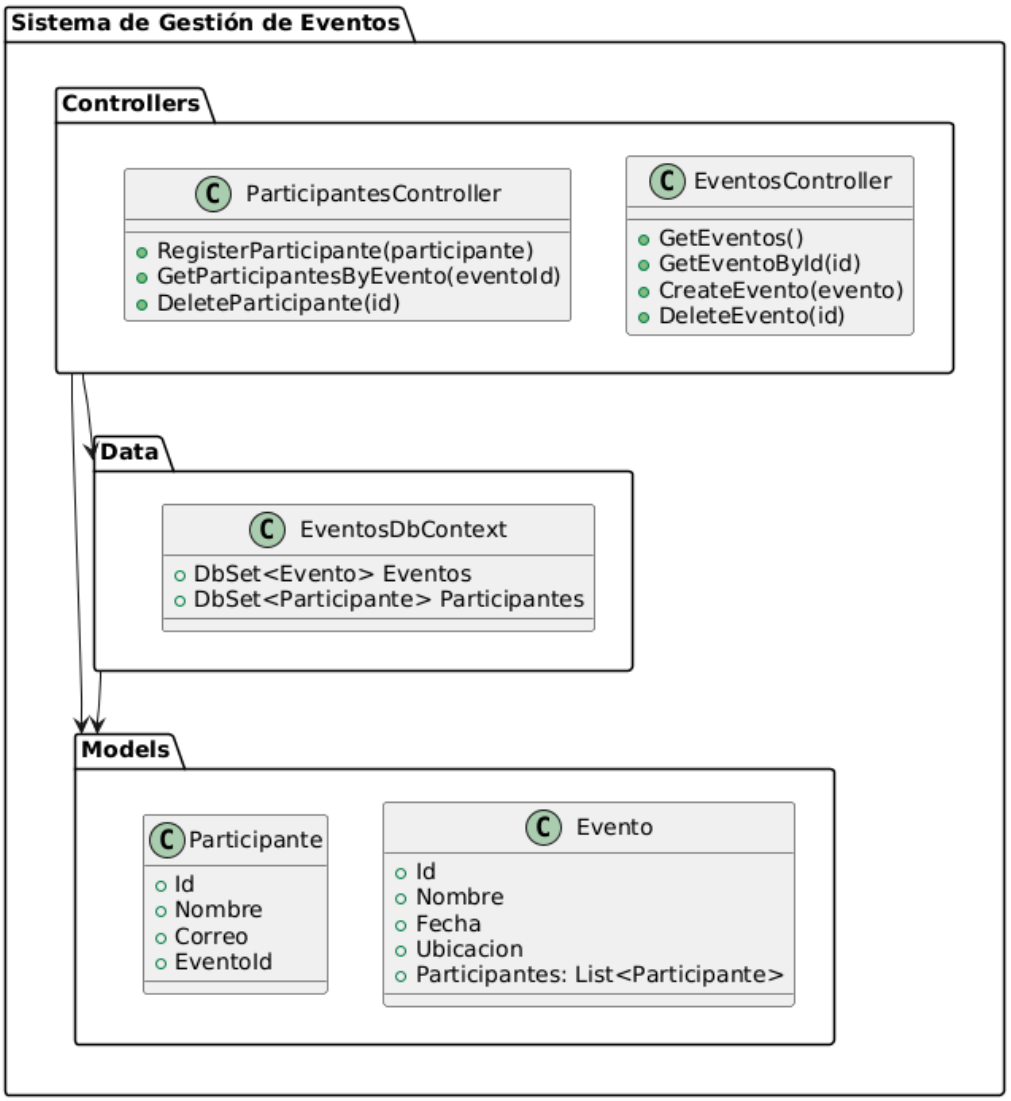


Programación Asíncrona

En EventosController y ParticipantesController se usa programación asíncrona, teniendo el recurso compartido la base de datos, que nos aseguras que las consultas no bloqueen el hilo principal, permitiendo que otras solicitudes se procesan simultáneamente.
Las consultas y modificaciones a la base de datos como ToListAsync, AddAsync, SaveChangeAsync son asíncronas lo que nos asegura un mejor rendimiento y cumplimiento con los requisitos del sistema.

Diagrama de la arquitectura

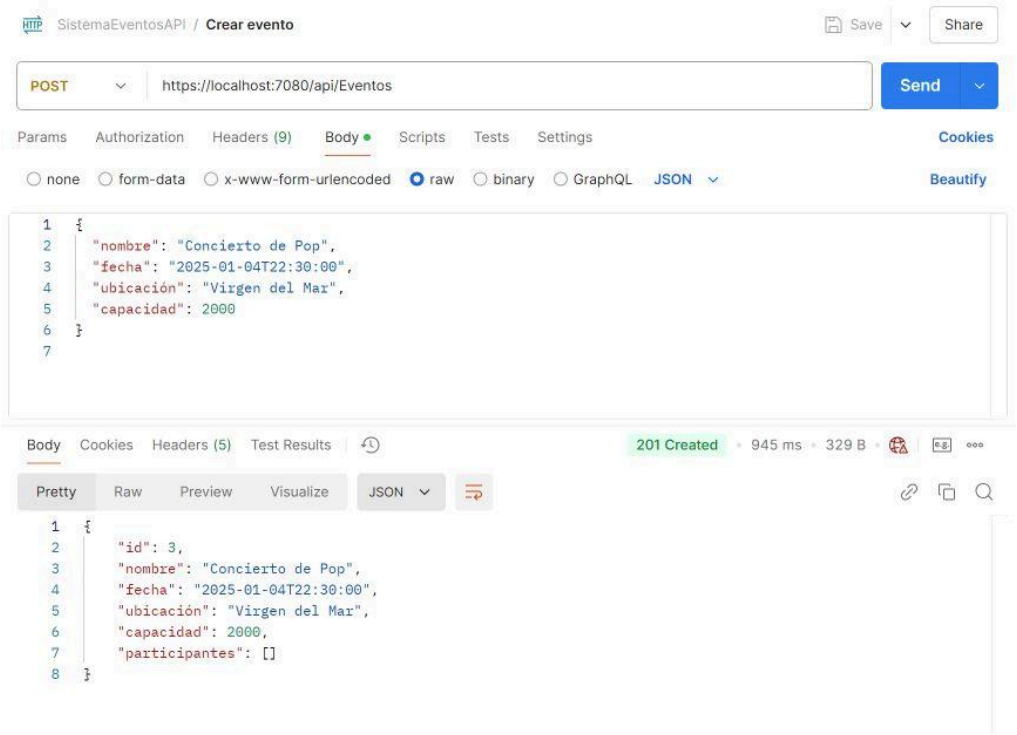




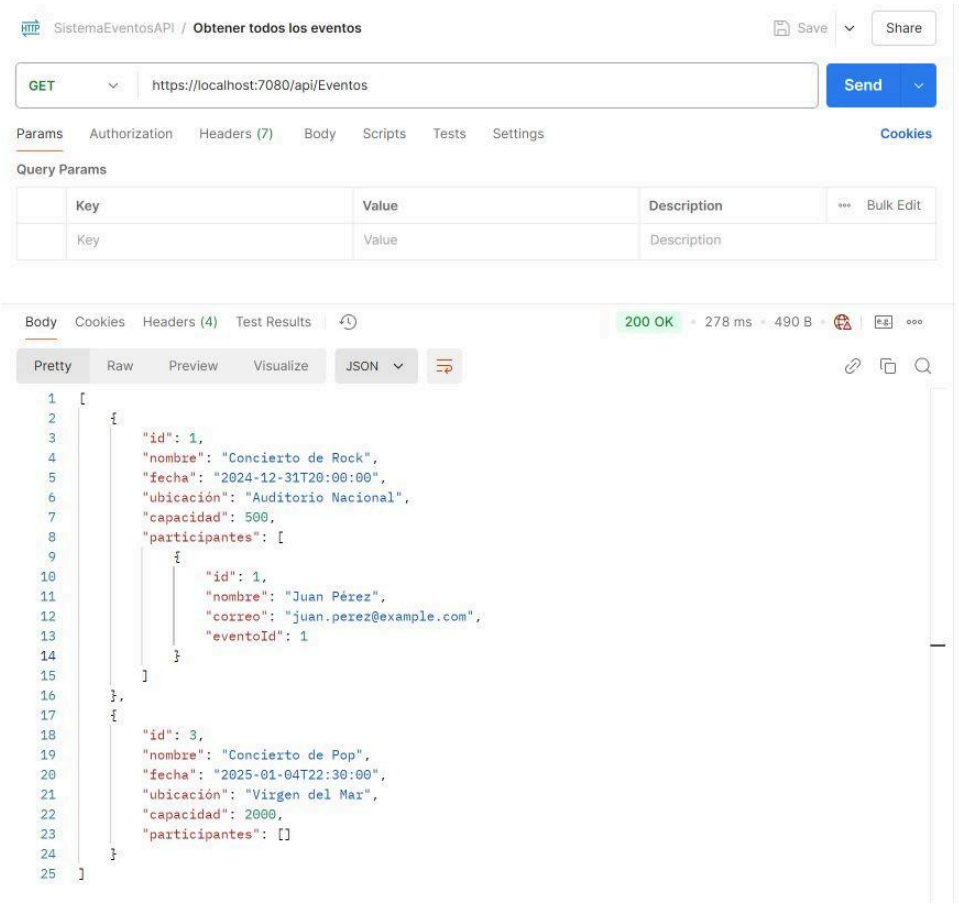
Pruebas Endpoint

Se realizaron pruebas GET, POST y DELETE, mediante el Postman para garantizar la calidad y el correcto funcionamiento del sistema:

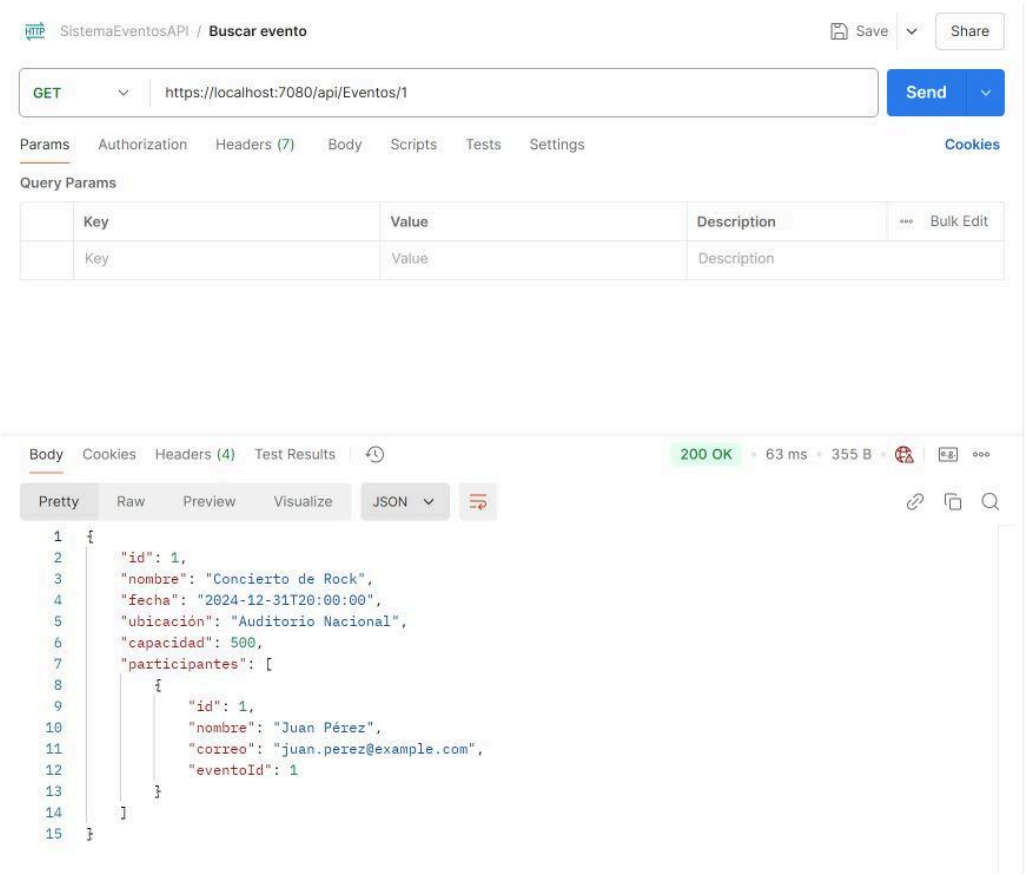
1. **POST/api/eventos:**



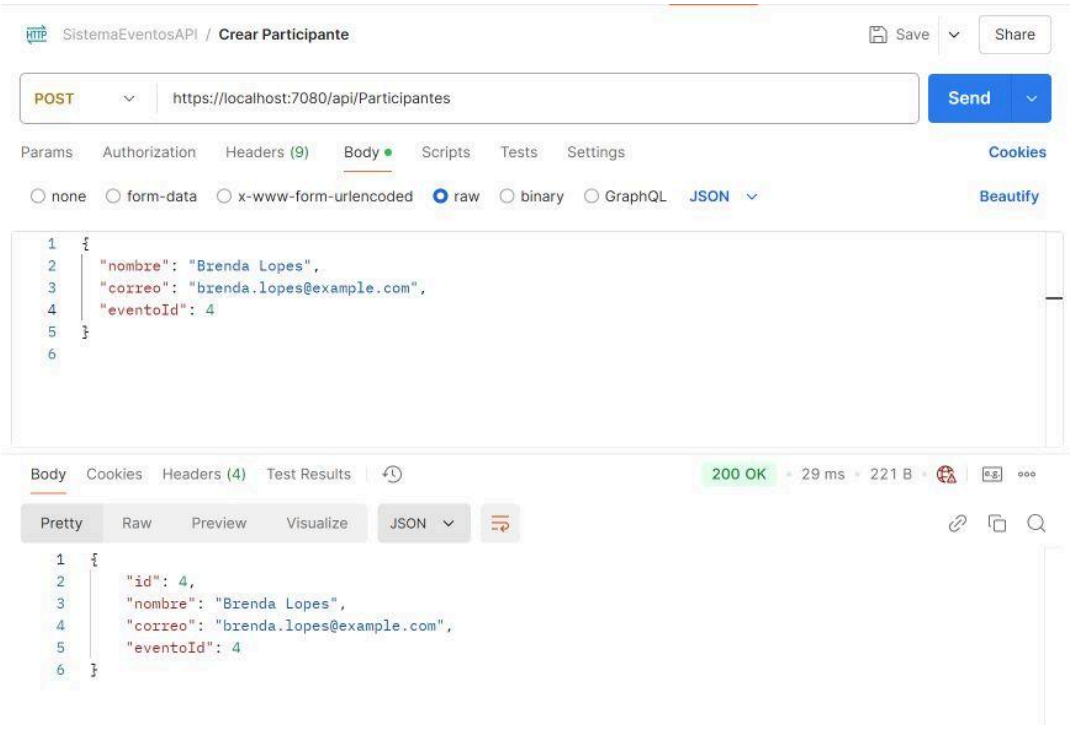
2. GET/api/eventos:



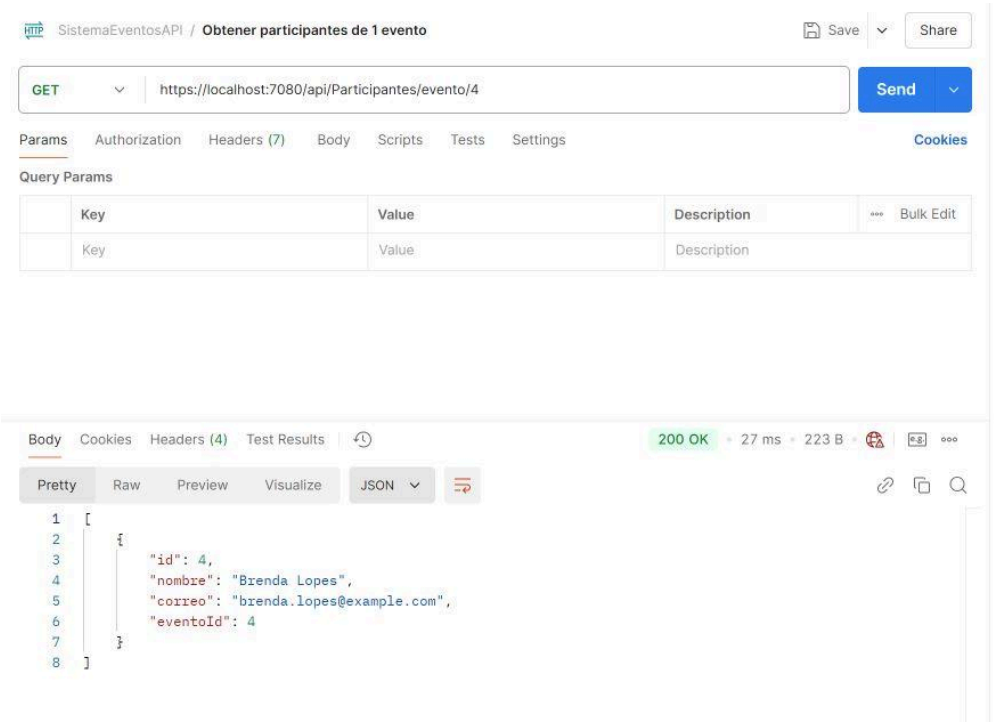
3. GET/api/eventos{id}



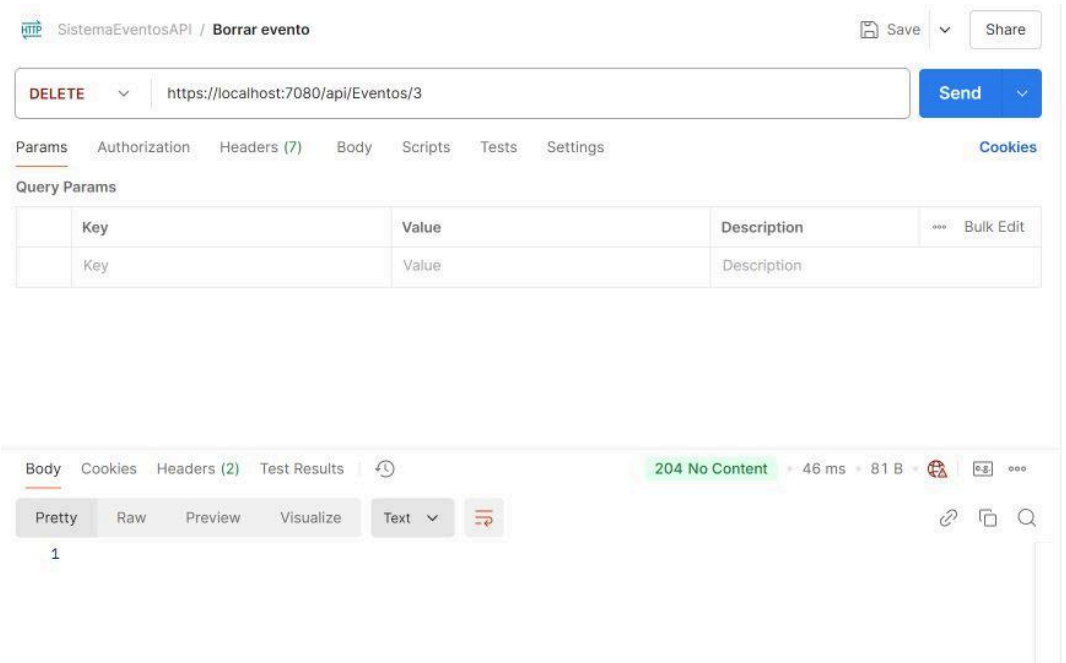
4. POST/api/Participantes



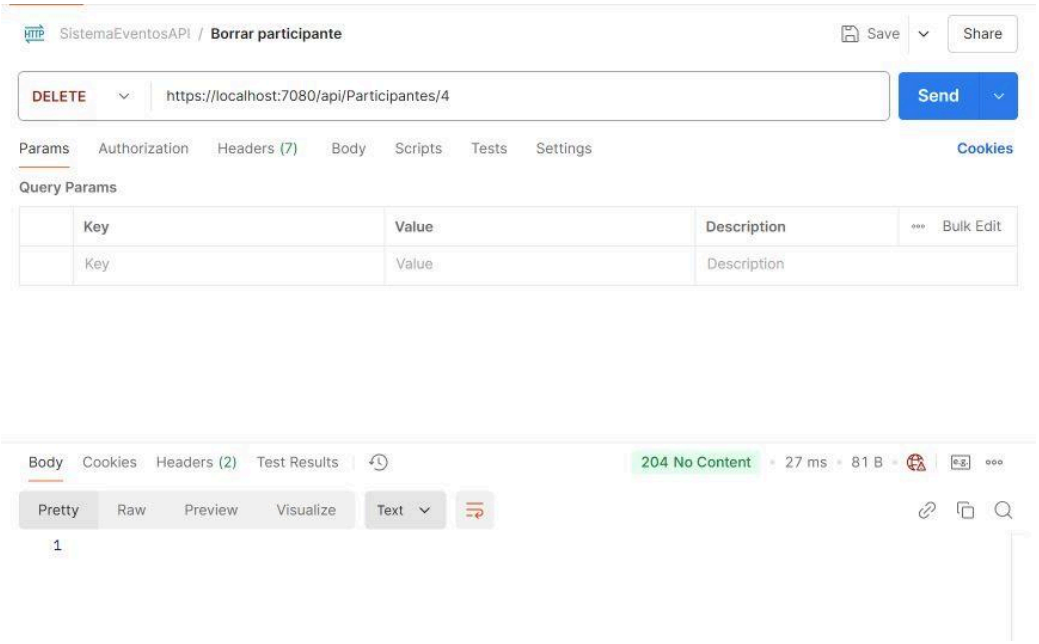
5. GET/api/participantes/evento/{eventoId}



6. DELETE /api/eventos/{id}



7. DELETE /api/participantes/{id}


SistemaEventosAPI / Borrar participante
Save
Share
DELETE
https://localhost:7080/api/Participantes/4
Send
Params
Authorization
Headers (7)
Body
Scripts
Tests
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Key
Value
Description

Body
Cookies
Headers (2)
Test Results
204 No Content
27 ms
81 B
Pretty
Raw
Preview
Visualize
Text
1