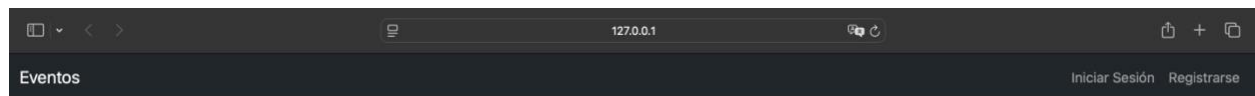


UIPR- Recinto de Arecibo
Programa de ciencias en computadora

Gestión de Eventos en línea
COMP 2025

Jorge J. Arce Vázquez
R00656626

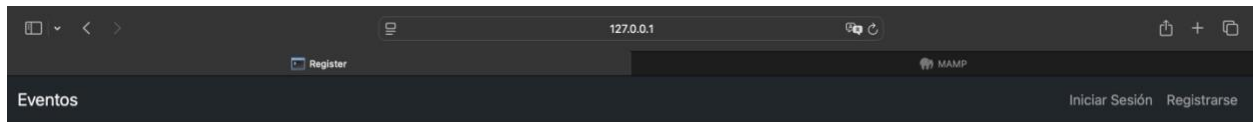
Pantalla principal una vez se ingrese a la página, la cual nos da la opción de iniciar sesión si ya tenemos una cuenta o crear una nueva.



Bienvenido al Administrador de Eventos

Por favor, inicia sesión o regístrate para continuar.

Esta es la interfaz de registro de usuario en la cual creamos un usuario nuevo con rol de participante



Registro de usuario

Nombre de usuario

Correo electrónico

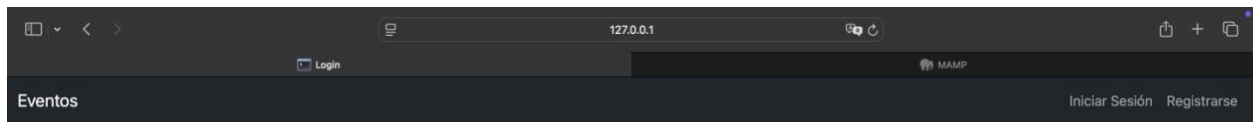
Contraseña

Confirmar contraseña

Role

Ya tiene un cuenta? [Inicie sesión aquí.](#)

Una vez nos registramos, se redirige a la interfaz de iniciar sesión en la cual colocamos los datos correo electrónico y contraseña, para luego tener acceso a los eventos disponibles si es que hay alguno.



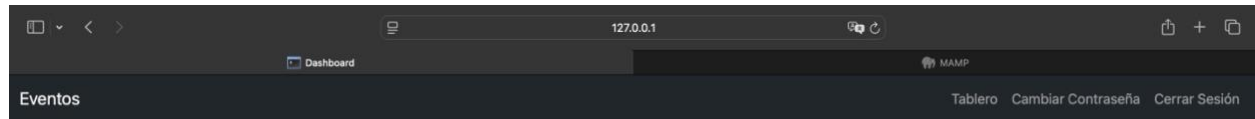
Iniciar sesión de usuario

Correo electrónico

Contraseña

No tiene una cuenta? [Regístrate aquí.](#)

Esta es la interfaz del participante en la cual no tiene permiso para crear, editar o borrar eventos solo verlos si es hay alguno.



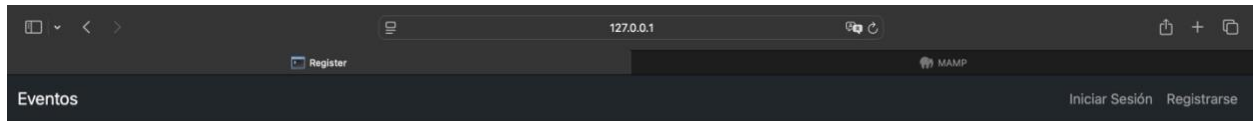
Administrador de Eventos

Nombre	Localización	Día y hora	Capacidad	Descripción	Acciones
--------	--------------	------------	-----------	-------------	----------

No hay eventos dispibles al momento.

Usted no tiene permiso para crear, actualizar o borrar eventos.

Esta es la interfaz de registro de usuario en la cual creamos uno con roles de administrador



Registro de usuario

Nombre de usuario

Jorge

Correo electrónico

Jorge@gmail.com

Contraseña

••••••••

Confirmar contraseña

••••••••

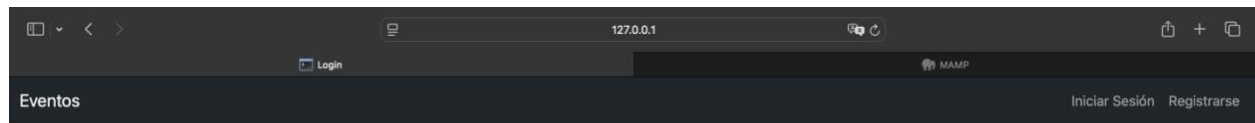
Role

Admin

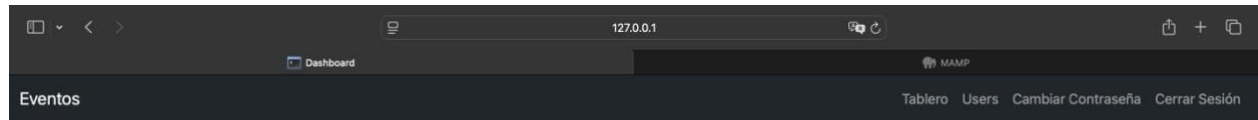
Register

Ya tiene un cuenta? [Inicie sesión aquí.](#)

Interfaz en la cual podemos iniciar sesión luego de habernos registrado con éxito.



En esta interfaz ya podemos ejercer nuestros roles de administrador como crear, borrar y editar eventos.



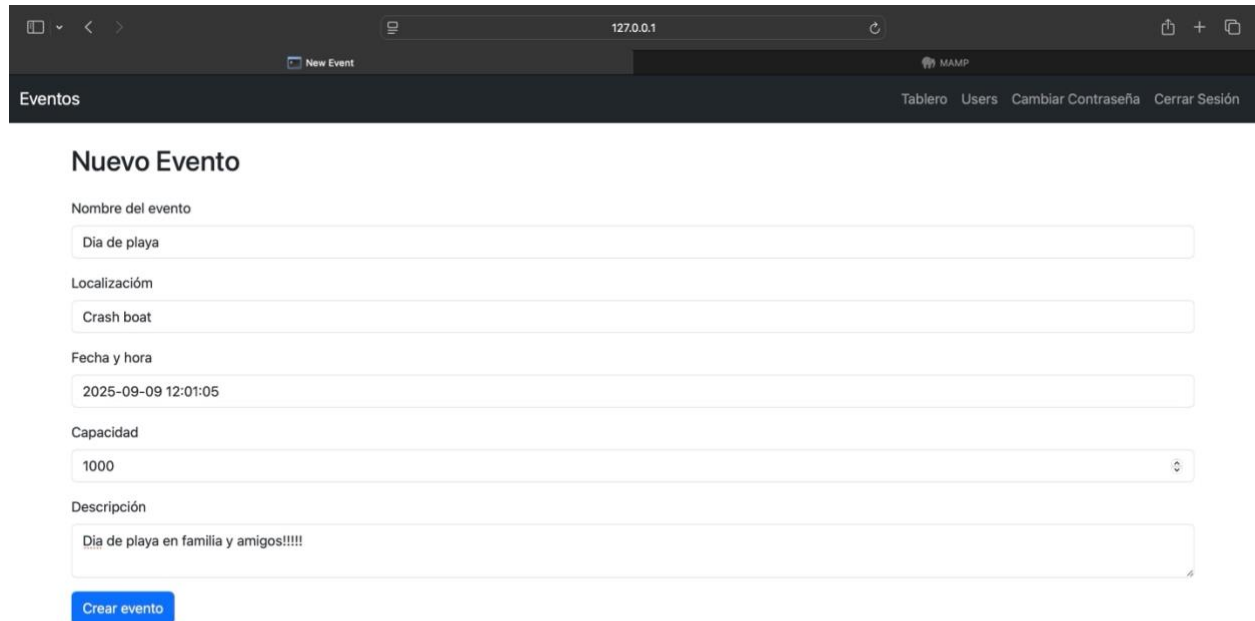
Administrador de Eventos

[+ Añadir Evento](#)

Nombre	Localización	Día y hora	Capacidad	Descripción	Acciones
--------	--------------	------------	-----------	-------------	----------

No hay eventos dispibles al momento.

Aquí ingresamos toda la información del nuevo evento que estamos creando



The screenshot shows a web browser window with a dark theme. The address bar shows '127.0.0.1'. The browser tab is labeled 'New Event'. The page has a dark header with the word 'Eventos' on the left and navigation links 'Tablero', 'Users', 'Cambiar Contraseña', and 'Cerrar Sesión' on the right. The main content area is titled 'Nuevo Evento' and contains several form fields: 'Nombre del evento' with the value 'Día de playa', 'Localización' with the value 'Crash boat', 'Fecha y hora' with the value '2025-09-09 12:01:05', 'Capacidad' with the value '1000', and 'Descripción' with the value 'Día de playa en familia y amigos!!!!'. A blue button labeled 'Crear evento' is at the bottom left of the form.

Eventos

Tablero Users Cambiar Contraseña Cerrar Sesión

Nuevo Evento

Nombre del evento

Día de playa

Localización

Crash boat

Fecha y hora

2025-09-09 12:01:05

Capacidad

1000

Descripción

Día de playa en familia y amigos!!!!

Crear evento

Una vez ingresada toda la información nos aparece un mensaje en pantalla que muestra que el evento fue creado con éxito y la información del mismo.



The screenshot shows a web browser window with a dark theme. The address bar displays '127.0.0.1'. The page has a header with 'Dashboard' and 'MAMP' links. A navigation bar at the top contains 'Eventos', 'Tablero', 'Users', 'Cambiar Contraseña', and 'Cerrar Sesión'. A light blue message box states 'Evento creado exitosamente.' Below this, the 'Administrador de Eventos' section features a '+ Añadir Evento' button and a table with one event entry.

Nombre	Localización	Día y hora	Capacidad	Descripción	Acciones
Día de playa	Crash boat	2025-09-09 12:01:05	1000	Día de playa en familia y amigos!!!!	 

En esta interfaz podemos editar el evento y guardar los cambios

127.0.0.1

Dashboard

MAMP

Eventos

TableroUsersCambiar ContraseñaCerrar Sesión

Evento actualizado exitosamente.

Administrador de Eventos

+ Añadir Evento





Nombre	Localización	Día y hora	Capacidad	Descripción	Acciones
Día de playa actualizado	Crash boat	2025-09-09 12:01:05	10	Día de playa en familia y amigos!!!!	 
Día de pesca	Manati	2025-09-09 12:12:12	50	Ven y disfruta de un día de pesca con los empleados de la empresa	 

Aquí borramos un evento y guardamos los cambios

Evento actualizado exitosamente.

Administrador de Eventos

+ Añadir Evento

Nombre	Localización	Día y hora	Capacidad	Descripción	Acciones
Día de playa actualizado	Crash boat	2025-09-09 12:01:05	10	Día de playa en familia y amigos!!!!	 
Día de pesca	Manatí	2025-09-09 12:12:12	50	Ven y disfruta de un día de pesca con los empleados de la empresa	 

Are you sure you want to delete this event?
Cancel OK

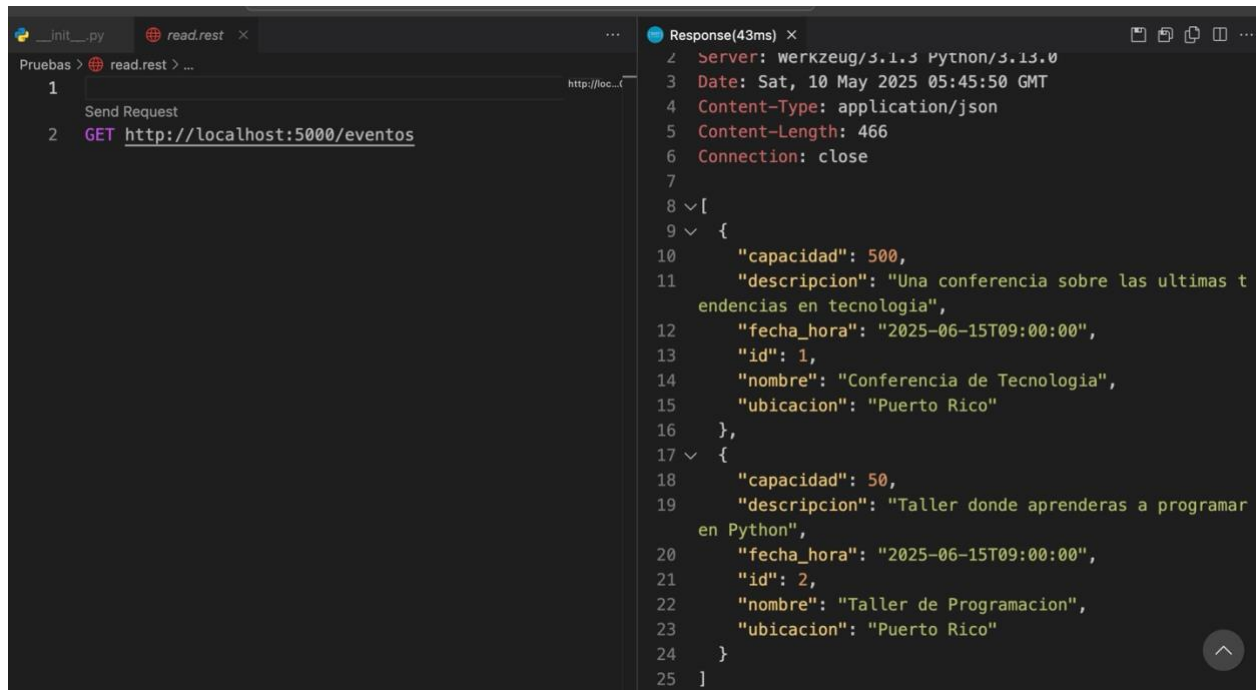
Evento eliminado exitosamente.

Administrador de Eventos

+ Añadir Evento

Nombre	Localización	Día y hora	Capacidad	Descripción	Acciones
Día de pesca	Manatí	2025-09-09 12:12:12	50	Ven y disfruta de un día de pesca con los empleados de la empresa	 

read.rest



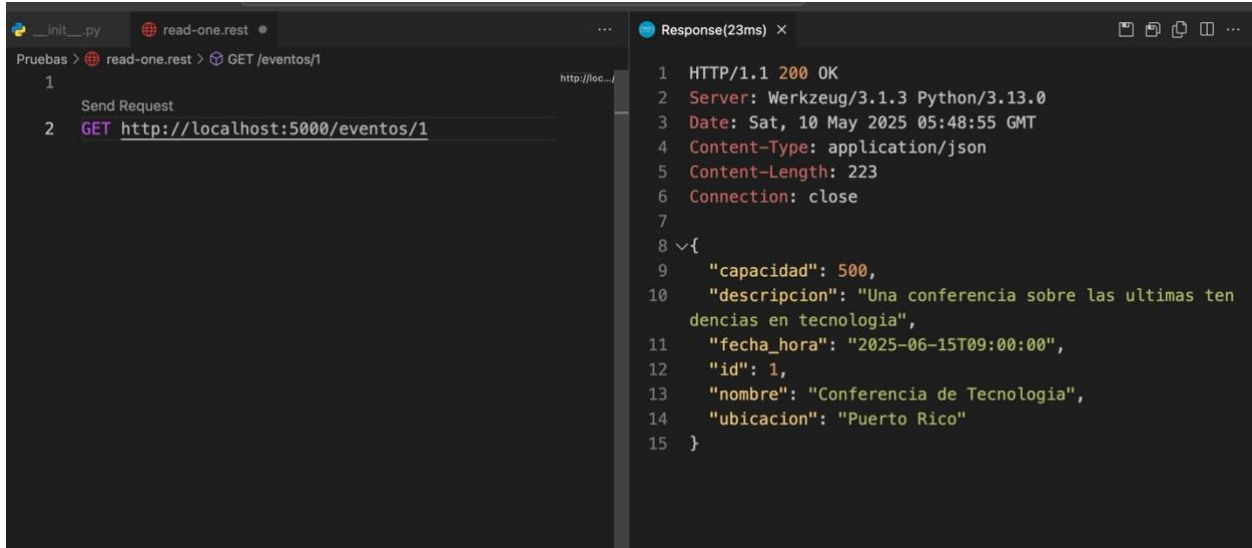
The screenshot displays the read.rest application interface. On the left, the 'Pruebas' (Tests) tab is active, showing a list of requests. The first request is a GET request to 'http://localhost:5000/eventos'. The right pane shows the 'Response(43ms)' for this request. The response headers indicate it's from Werkzeug/3.1.3 Python/3.13.0, dated Sat, 10 May 2025 05:45:50 GMT, with a Content-Type of application/json and a Content-Length of 466. The response body is a JSON array of two event objects.

```
1 Send Request
2 GET http://localhost:5000/eventos

Response(43ms)
2 Server: Werkzeug/3.1.3 Python/3.13.0
3 Date: Sat, 10 May 2025 05:45:50 GMT
4 Content-Type: application/json
5 Content-Length: 466
6 Connection: close
7
8 [
9   {
10    "capacidad": 500,
11    "descripcion": "Una conferencia sobre las ultimas t
12    endencias en tecnologia",
13    "fecha_hora": "2025-06-15T09:00:00",
14    "id": 1,
15    "nombre": "Conferencia de Tecnologia",
16    "ubicacion": "Puerto Rico"
17  },
18  {
19    "capacidad": 50,
20    "descripcion": "Taller donde aprenderas a programar
21    en Python",
22    "fecha_hora": "2025-06-15T09:00:00",
23    "id": 2,
24    "nombre": "Taller de Programacion",
25    "ubicacion": "Puerto Rico"
26  }
27 ]
```

En este archivo de prueba utilizamos el metodo GET y recibimos todos los eventos existentes.

read-one.rest



The screenshot shows a REST client interface with two panes. The left pane displays the request details, and the right pane shows the response details.

Request Details:

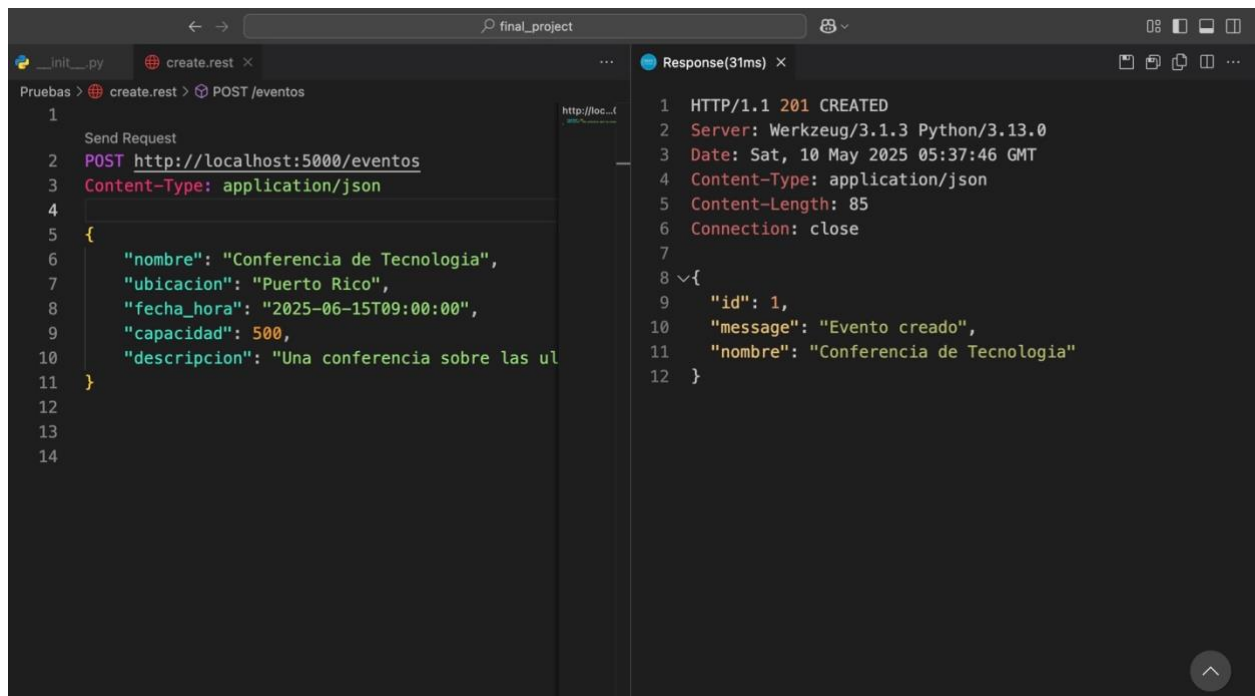
- Pruebas > read-one.rest > GET /eventos/1
- 1 Send Request
- 2 GET http://localhost:5000/eventos/1

Response Details:

```
1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.3 Python/3.13.0
3 Date: Sat, 10 May 2025 05:48:55 GMT
4 Content-Type: application/json
5 Content-Length: 223
6 Connection: close
7
8 {
9   "capacidad": 500,
10  "descripcion": "Una conferencia sobre las ultimas ten
11  dencias en tecnologia",
12  "fecha_hora": "2025-06-15T09:00:00",
13  "id": 1,
14  "nombre": "Conferencia de Tecnologia",
15  "ubicacion": "Puerto Rico"
16 }
```

Aquí recibimos un solo evento dependiendo del id que invoquemos.

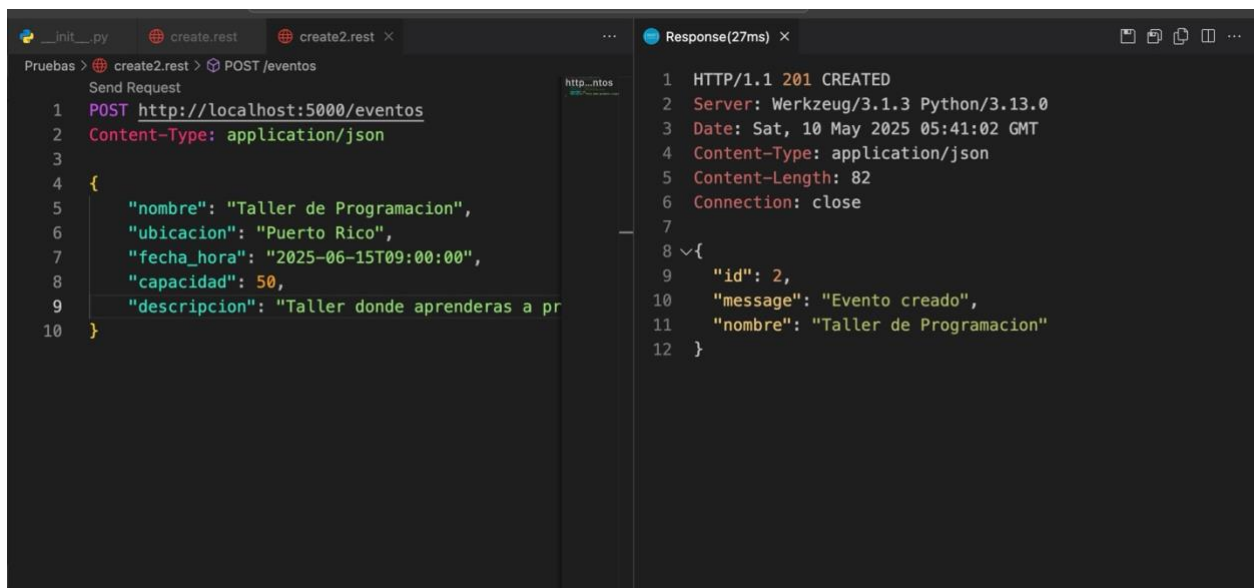
create.rest



The screenshot shows a REST client interface with a tab labeled 'create.rest'. The 'Send Request' section displays a POST request to 'http://localhost:5000/eventos' with a JSON body. The 'Response(31ms)' section shows the server's response, which is a 201 status code and a JSON object indicating the event was created.

```
1 Send Request
2 POST http://localhost:5000/eventos
3 Content-Type: application/json
4
5 {
6   "nombre": "Conferencia de Tecnologia",
7   "ubicacion": "Puerto Rico",
8   "fecha_hora": "2025-06-15T09:00:00",
9   "capacidad": 500,
10  "descripcion": "Una conferencia sobre las ult
11 }
12
```

```
1 HTTP/1.1 201 CREATED
2 Server: Werkzeug/3.1.3 Python/3.13.0
3 Date: Sat, 10 May 2025 05:37:46 GMT
4 Content-Type: application/json
5 Content-Length: 85
6 Connection: close
7
8 {
9   "id": 1,
10  "message": "Evento creado",
11  "nombre": "Conferencia de Tecnologia"
12 }
```



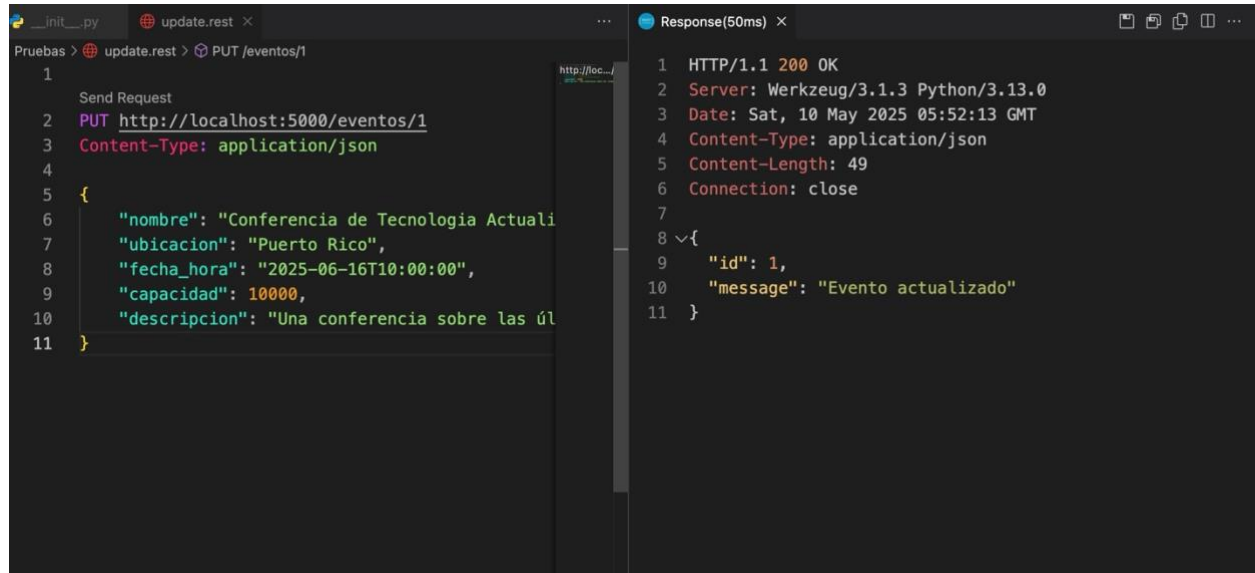
The screenshot shows a REST client interface with a tab labeled 'create2.rest'. The 'Send Request' section displays a POST request to 'http://localhost:5000/eventos' with a JSON body. The 'Response(27ms)' section shows the server's response, which is a 201 status code and a JSON object indicating the event was created.

```
1 Send Request
2 POST http://localhost:5000/eventos
3 Content-Type: application/json
4
5 {
6   "nombre": "Taller de Programacion",
7   "ubicacion": "Puerto Rico",
8   "fecha_hora": "2025-06-15T09:00:00",
9   "capacidad": 50,
10  "descripcion": "Taller donde aprenderas a pr
11 }
12
```

```
1 HTTP/1.1 201 CREATED
2 Server: Werkzeug/3.1.3 Python/3.13.0
3 Date: Sat, 10 May 2025 05:41:02 GMT
4 Content-Type: application/json
5 Content-Length: 82
6 Connection: close
7
8 {
9   "id": 2,
10  "message": "Evento creado",
11  "nombre": "Taller de Programacion"
12 }
```

utilizamos el metodo post en en cual le enviamos un nuevo evento al servidor para asi crear los mismos y añadirlos al administrador de eventos.

update.rest



The screenshot shows a REST client interface with two main panels. The left panel displays the request details, and the right panel shows the response details.

Request Details:

- Method: PUT
- URL: `http://localhost:5000/eventos/1`
- Content-Type: `application/json`
- Request Body (JSON):

```
{  "nombre": "Conferencia de Tecnologia Actuali",  "ubicacion": "Puerto Rico",  "fecha_hora": "2025-06-16T10:00:00",  "capacidad": 10000,  "descripcion": "Una conferencia sobre las úl"}
```

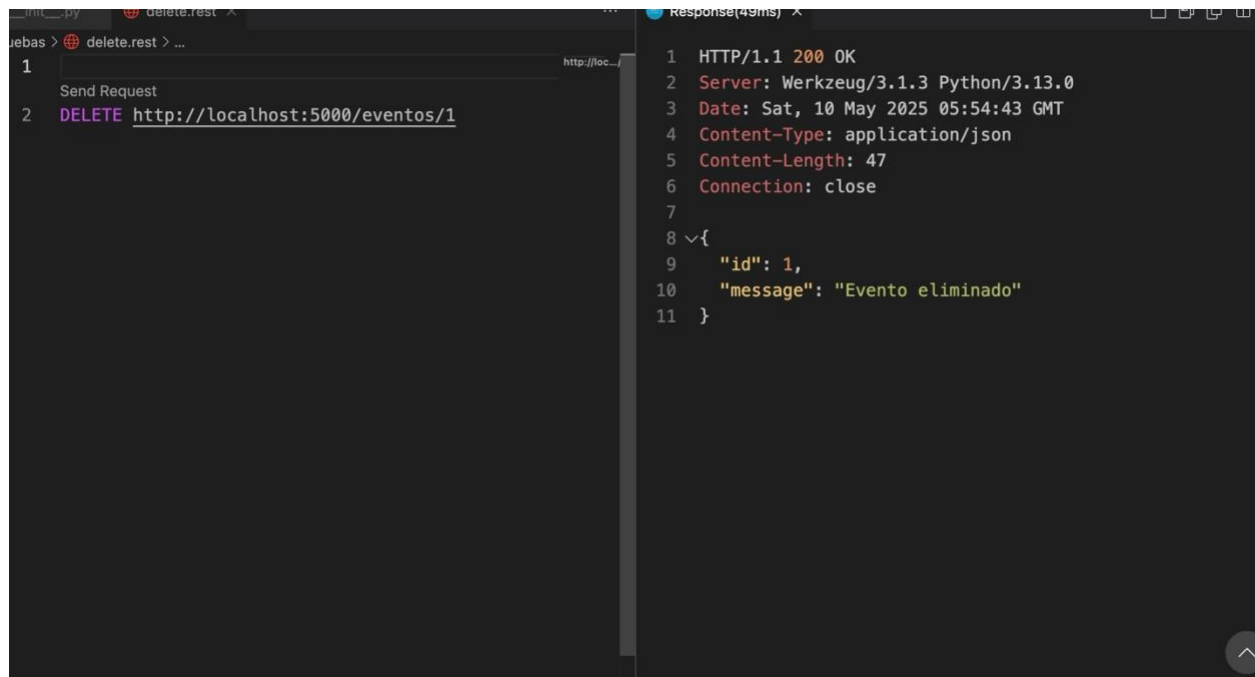
Response Details:

- Status: HTTP/1.1 200 OK
- Server: Werkzeug/3.1.3 Python/3.13.0
- Date: Sat, 10 May 2025 05:52:13 GMT
- Content-Type: `application/json`
- Content-Length: 49
- Connection: close
- Response Body (JSON):

```
{  "id": 1,  "message": "Evento actualizado"}
```

En este archivo utilizamos el metodo PUT el cual nos permite modificar información ya existente la cual en este caso fue el evento #1

delete.rest



The screenshot shows a REST client interface with two panels. The left panel, titled 'delete.rest', contains a list of requests. The second request is a DELETE method to the URL 'http://localhost:5000/eventos/1'. The right panel, titled 'Response(49ms)', displays the server's response. It starts with the status line 'HTTP/1.1 200 OK', followed by headers: 'Server: Werkzeug/3.1.3 Python/3.13.0', 'Date: Sat, 10 May 2025 05:54:43 GMT', 'Content-Type: application/json', and 'Content-Length: 47'. The response body is a JSON object: {'id': 1, 'message': 'Evento eliminado'}. The interface is dark-themed with a vertical scrollbar on the left panel.

```
1 1 Send Request http://loc...
2 2 DELETE http://localhost:5000/eventos/1

1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.3 Python/3.13.0
3 Date: Sat, 10 May 2025 05:54:43 GMT
4 Content-Type: application/json
5 Content-Length: 47
6 Connection: close
7
8 {
9   "id": 1,
10  "message": "Evento eliminado"
11 }
```

Utilizamos el método DELETE para borrar el evento #1

https://github.com/Jaelv04/comp2052/tree/main/Proyecto_final_eventos