

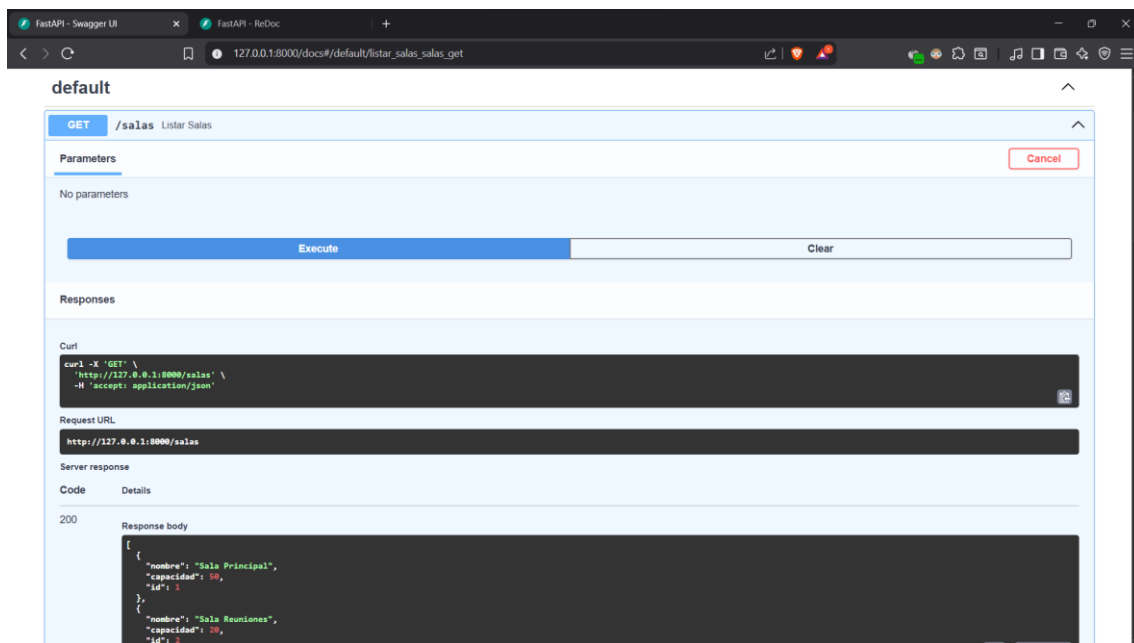


PRÁCTICA DE APLICACIONES WEB
TAREA API REST GRUPAL

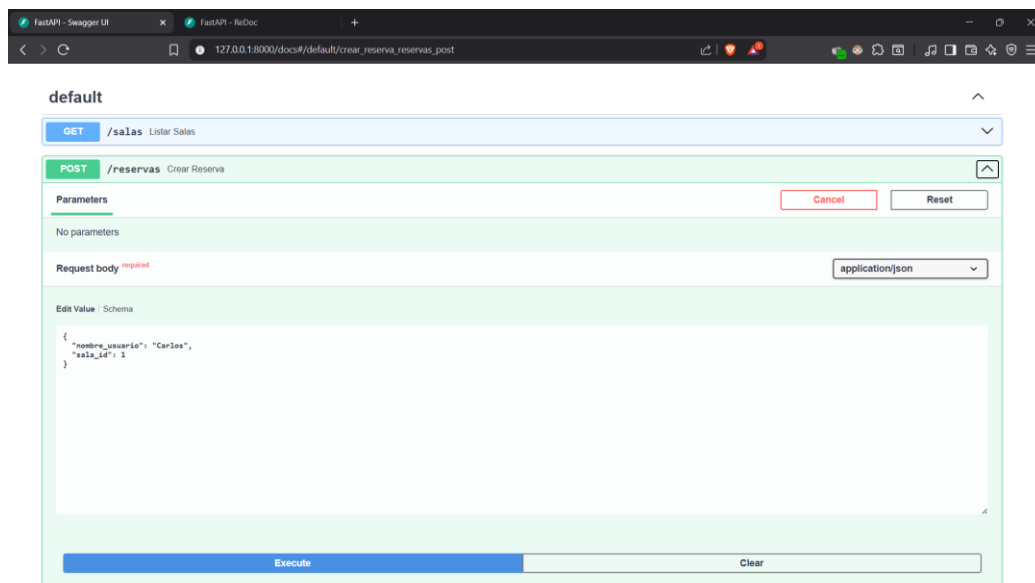
Integrantes: Fabián Simbaña, Ozzy Loachamin, Brayan Ortiz, Luis Pérez, Elias Cazar

1. Capturas de pantalla API funcionando en Swagger UI

Get /salas



Post /reservas





The screenshot shows the Swagger UI interface for a REST API. The browser address bar displays the URL `127.0.0.1:8000/docs#/default/crear_reserva_reservas_post`. The 'Responses' tab is selected, showing a `200` status code. The response body is a JSON object: `{ "nombre_usuario": "Carlos", "sala_id": 1, "id": 12 }`. The response headers include `content-length: 47`, `content-type: application/json`, `date: Mon, 21 Jul 2025 02:06:10 GMT`, and `server: unicorn`. Below the response details, a table lists the response with a `200` status and the description 'Successful Response'. A dropdown menu for 'Media type' is set to 'application/json'.

Get Reservas/mis-reservas (se ve Carlos, la creada anteriormente)

The screenshot shows the Swagger UI interface for a REST API. The browser address bar displays the URL `127.0.0.1:8000/docs#/default/obtener_reservas_reservas_mis_reservas_get`. The 'Responses' tab is selected, showing a `200` status code. The response body is a JSON array of reservation objects. The first object is for 'Carlos' with `sala_id: 1` and `id: 12`. The response headers include `content-length: 345`, `content-type: application/json`, `date: Mon, 21 Jul 2025 02:07:48 GMT`, and `server: unicorn`. Below the response details, a table lists the response with a `200` status and the description 'Successful Response'. A dropdown menu for 'Media type' is set to 'application/json'.



Post /menu

The screenshot shows the Swagger UI interface for the POST /menu endpoint. The endpoint is titled "Create Menu". The request body is required and set to application/json. The request body is a JSON object with the following structure:

```
{  "name": "hamburguesa",  "description": "carne, pan, lechuga y queso"}
```

The interface includes an "Execute" button and a "Clear" button.

Get /menu (se observa el menú creado anteriormente “hamburguesa”)

The screenshot shows the Swagger UI interface for the GET /menu endpoint. The endpoint is titled "Get Menu". The response is a JSON object with the following structure:

```
{  "name": "hamburguesa",  "description": "carne, pan, lechuga y queso",  "id": 1}
```

The interface includes an "Execute" button and a "Clear" button. The response is displayed in the "Responses" section, showing the status code 200 and the response body.



Post /tables

The screenshot shows the Swagger UI interface for the POST /tables endpoint. The endpoint is titled "Create Table". The "Parameters" section is empty, indicating no query or path parameters. The "Request body" section is marked as "required" and has a dropdown menu set to "application/json". Below this, the "Edit Value" section shows a JSON schema for the request body:

```
{  "table_name": "mesaClientesNuevos",  "location": "UID"}
```

At the bottom of the interface, there are "Execute" and "Clear" buttons.

Get /tables (se muestra la mesa creada anteriormente)

The screenshot shows the Swagger UI interface for the GET /tables endpoint, titled "Get Tables". The "Parameters" section is empty. Below the "Execute" and "Clear" buttons, the "Responses" section is expanded, showing the response for a 200 status code. The response body is a JSON object:

```
{  "table_name": "mesaClientesNuevos",  "location": "UID",  "id": 1}
```

Below the response body, there is a "Download" button. The "Request URL" section shows the URL: `http://127.0.0.1:8000/tables`. The "Curl" section shows the command to execute the request:

```
curl -X 'GET' \  'http://127.0.0.1:8000/tables' \  -H 'accept: application/json'
```



Post /customers

FastAPI - Swagger UI | FastAPI - ReDoc

127.0.0.1:8000/docs/#/default/create_customer_customers_post

POST /customers Create Customer

Parameters Cancel Reset

No parameters

Request body required application/json

Edit Value | Schema

```
{  "name": "Fabian Chancusig",  "phone_number": "0986152164"}
```

Get /customers

FastAPI - Swagger UI | FastAPI - ReDoc

127.0.0.1:8000/docs/#/default/get_customers_customers_get

GET /customers Get Customers

Parameters Cancel

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \  'http://127.0.0.1:8000/customers' \  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/customers
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "name": "Fabian Chancusig", "phone_number": "0986152164",}</pre>



Post /bookings

The screenshot shows the Swagger UI interface for a POST request to the endpoint `/bookings`. The request body is a JSON object:

```
{  "customer_id": 1,  "table_id": 1,  "date": "2025-07-21T02:17:25.243Z",  "notes": "agendar de mesa nueva"}
```

Below the request body, there is an `Execute` button and a `Clear` button. The `Responses` section is empty. The `Curl` section shows the following command:

```
curl -X 'POST' \  http://127.0.0.1:8000/bookings \  -H 'accept: application/json' \  -H 'Content-Type: application/json' \  -d '{  "customer_id": 1,  "table_id": 1,  "date": "2025-07-21T02:17:25.243Z",  "notes": "agendar de mesa nueva"  }'
```

Get /bookings

The screenshot shows the Swagger UI interface for a GET request to the endpoint `/bookings`. The `Parameters` section is empty. Below it, there is an `Execute` button and a `Clear` button. The `Responses` section is empty. The `Curl` section shows the following command:

```
curl -X 'GET' \  http://127.0.0.1:8000/bookings \  -H 'accept: application/json'
```

The `Request URL` section shows the URL `http://127.0.0.1:8000/bookings`. The `Server response` section shows a `200` status code. The `Response body` section shows the following JSON object:

```
{  "customer_id": 1,  "table_id": 1,  "date": "2025-07-20T21:17:25.243000",  "notes": "agendar de mesa nueva",  "id": 1}
```

There is a `Download` button next to the response body.



Post /orders

The screenshot shows the Swagger UI for the POST /orders endpoint. The URL bar indicates the API is running at 127.0.0.1:8000. The endpoint is configured with a JSON body containing the following fields:

```
{  "order_type": "mesa para cliente fabia",  "menu_item_id": 1,  "customer_id": 1,  "table_id": 1}
```

Below the body editor, there are 'Execute' and 'Clear' buttons. The 'Responses' section is currently empty. The 'Curl' section shows the following command:

```
curl -X 'POST' \  'http://127.0.0.1:8000/orders' \  -H 'accept: application/json' \  -H 'Content-Type: application/json' \  -d '{  "order_type": "mesa para cliente fabia",  "menu_item_id": 1,  "customer_id": 1,  "table_id": 1  }'
```

The 'Request URL' section shows the endpoint: `http://127.0.0.1:8000/orders`.

Get /orders

The screenshot shows the Swagger UI for the GET /orders endpoint. The URL bar indicates the API is running at 127.0.0.1:8000. The endpoint is configured with the following details:

- Method:** GET
- Path:** /orders
- Parameters:** No parameters
- Execute:** Button to execute the request
- Clear:** Button to clear the request
- Responses:** Section showing the response details
- Curl:**

```
curl -X 'GET' \  'http://127.0.0.1:8000/orders' \  -H 'accept: application/json'
```
- Request URL:** `http://127.0.0.1:8000/orders`
- Server response:** Section showing the response details
- Code:** 200
- Details:** Section showing the response body
- Response body:**

```
{  "order_type": "mesa para cliente fabia",  "menu_item_id": 1,  "customer_id": 1,  "table_id": 1,  "id": 1}
```
- Response headers:** Section showing the response headers



Delete /reservas/{id}

DELETE /reservas/{id} Eliminar Reserva

Parameters

Name	Description
id * required Integer (path)	8

Execute Clear

Responses

200

Response body

```
{ "ok": true }
```

Response headers

```
content-length: 11  
content-type: application/json  
date: Mon, 21 Jul 2025 02:36:30 GMT  
server: nginx
```

Get /reservas/mis-reservas (se eliminó la reserva con id: 8)

GET /reservas/mis-reservas Obtener Reservas

Parameters

No parameters

Execute Clear

Responses

200

Response body

```
[  
  {  
    "nombre_usuario": "string!!",  
    "sala_id": 1,  
    "id": 4  
  },  
  {  
    "nombre_usuario": "string!!",  
    "sala_id": 1,  
    "id": 5  
  },  
  {  
    "nombre_usuario": "string!!",  
    "sala_id": 2,  
    "id": 6  
  },  
  {  
    "nombre_usuario": "string!!",  
    "sala_id": 2,  
    "id": 10  
  },  
]
```

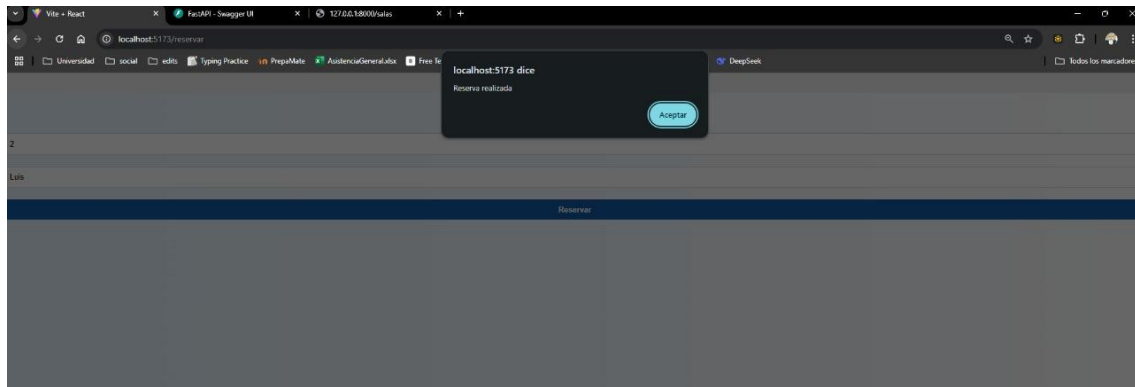



2. Capturas de pantalla Frontend

Pantalla de las salas

Salas		
Listado de Salas		
Nombre:	Sala Principal	Capacidad: 50
Nombre:	Sala Reuniones	Capacidad: 20

Reserva de sala



Lista de mis reservas

Salas		
Reservar		
Mis Reservas		
Usuario: Brayan - Sala: 1		Eliminar
Usuario: Luis - Sala: 2		Eliminar