

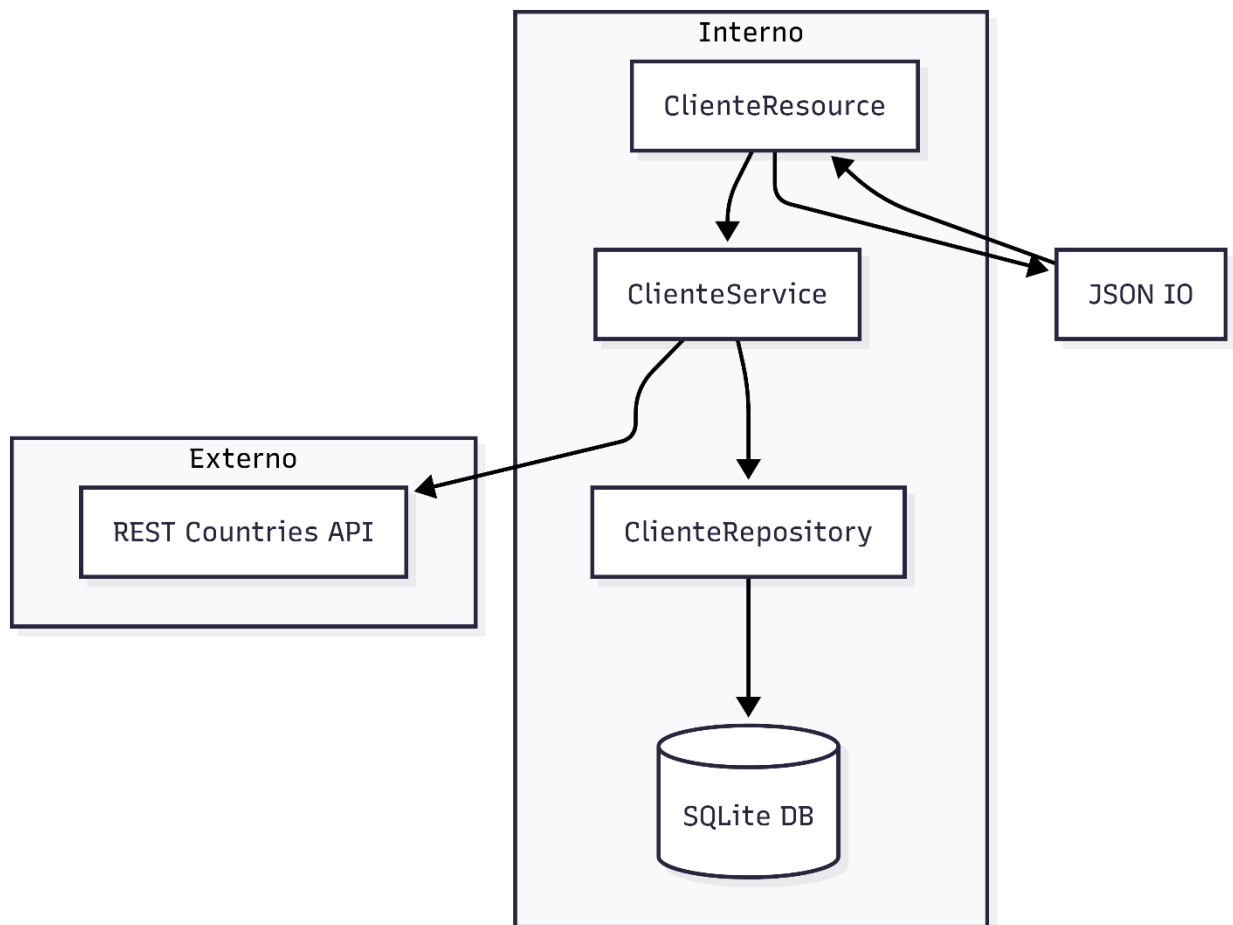
## Documentación Client Api – Ingeniero Backend Banreservas

Este proyecto implementa una API REST para la gestión de clientes, utilizando una arquitectura organizada en capas y conectándose a servicios externos colocados en el requerimiento de esta prueba.

### Arquitectura General

#### Enlace:

<https://www.mermaidchart.com/app/projects/935acf79-41ca-4c39-93d6-7aa35486f820/diagrams/45721a52-8645-4a63-b726-c6f517492df9/version/v0.1/edit>



## Componentes

**Internos - ClienteResource:** Punto de entrada de las solicitudes HTTP. Expone los endpoints de la API.

- **ClienteService:** Contiene la lógica de negocio y orquesta las llamadas al repositorio y a servicios externos.

- **ClienteRepository:** Se encarga de interactuar con la base de datos SQLite utilizando Panache de Quarkus.

- **SQLite DB:** Base de datos liviana embebida para persistencia de datos.

**Externos - REST Countries API:** API externa utilizada para obtener la nacionalidad del cliente a partir del código de país. (Requerido en el documento).

- **JSON IO:** Canal de entrada/salida de datos en formato JSON entre el cliente y el recurso ClientResource

## Tecnologías Utilizadas

- Java 17+

- Quarkus

- Jakarta REST (JAX-RS)

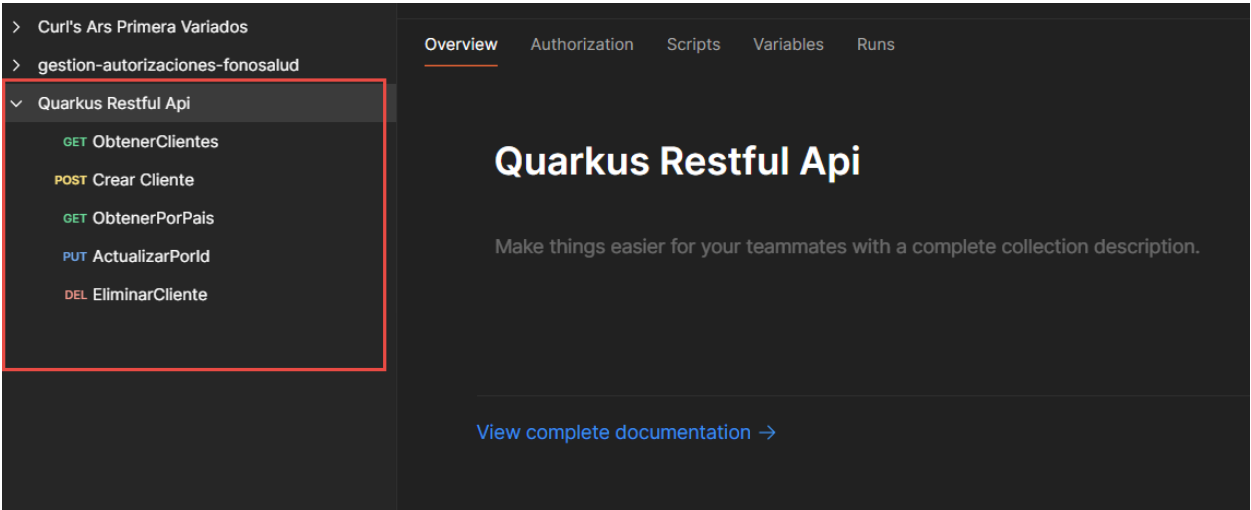
- SQLite - Panache ORM

- HTTP Client (Java 11+)

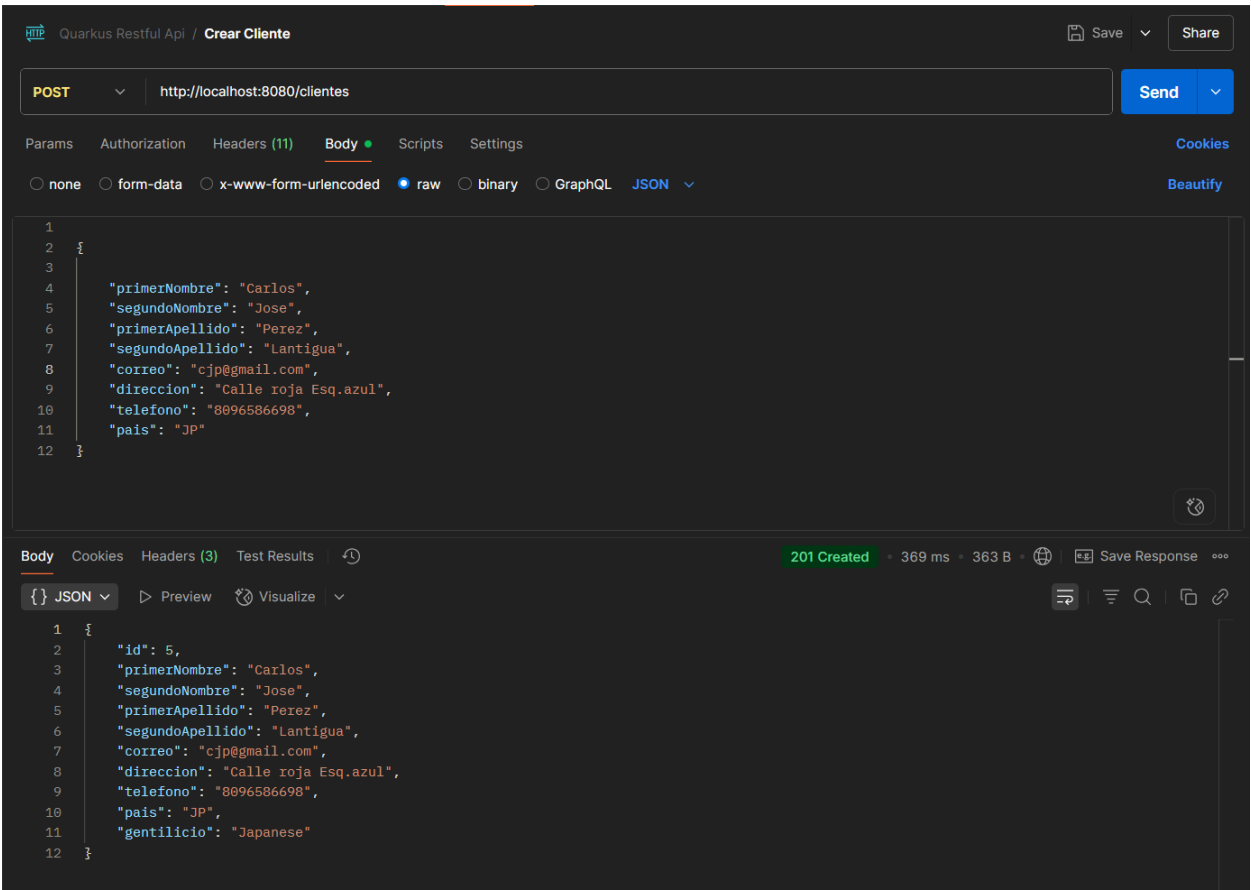
- REST Countries API (<https://restcountries.com>)

# Endpoints Principales

Postman Collections (localhost):



- POST /clientes: Crear un nuevo cliente:



- GET /clientes: Listar todos los clientes

Quarkus Restful Api / ObtenerClientes

SaveShare

GEThttp://localhost:8080/clientesSend

ParamsAuthorizationHeaders (8)BodyScriptsSettingsCookies

noneform-datax-www-form-urlencodedrawbinaryGraphQL

This request does not have a body

BodyCookiesHeaders (2)Test Results200 OK765 ms1.2 KBSave Response

JSONPreviewVisualize

1[2{3  "id": 1,4  "primerNombre": "Alonzo",5  "segundoNombre": "Luis",6  "primerApellido": "Martinez",7  "segundoApellido": "Perez",8  "correo": "alonzo@gmail.com",9  "direccion": "Calle Walla Esq. City",10  "telefono": "18095554466",11  "pais": "DO",12  "gentilicio": "Dominican"13  },14  {15    "id": 2,16    "primerNombre": "Joel",17    "segundoNombre": "Alberto",18    "primerApellido": "Estepan",19    "segundoApellido": "Ramon",20    "correo": "MAG@GMAIL.COM",21    "direccion": "SANTIAGO",22    "telefono": "809565563",23    "pais": "do",24    "gentilicio": "Dominican"25  },26  ]

- GET /clientes/{id}: Obtener un cliente por ID

Quarkus Restful Api / ObtenerClientePorId

SaveShare

GET

http://localhost:8080/clientes/2

Send

ParamsAuthorizationHeaders (8)BodyScriptsSettings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

BodyCookiesHeaders (2)Test Results

200 OK · 24 ms · 303 B · Save Response

JSON

PreviewVisualize

```
1 {
2   "id": 2,
3   "primerNombre": "Joel",
4   "segundoNombre": "Alberto",
5   "primerApellido": "Esteban",
6   "segundoApellido": "Ramon",
7   "correo": "MAG@GMAIL.COM",
8   "direccion": "SANTIAGO",
9   "telefono": "809565563",
10  "pais": "do",
11  "gentilicio": "Dominican"
12 }
```

- PUT /clientes/lid/: Actualizar un cliente existente

En este caso el requerimiento especifica que solo se pueden actualizar estos 4 campos, lo que hice fue utilizar un DTO para el request que solamente muestre los campos actualizables y en la respuesta traer el registro completo actualizado.

5. Actualizar un cliente existente, permitiendo solo la modificación de su correo, dirección, teléfono y país.

The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** http://localhost:8080/clientes/2
- Body (Request):**

```
1 {
2   "correo": "jalbertoestepan@GMAIL.COM",
3   "direccion": "Santo Domingo Este",
4   "telefono": "8094028717",
5   "pais": "DO"
6 }
```
- Status:** 200 OK
- Body (Response):**

```
1 {
2   "id": 2,
3   "primerNombre": "Joel",
4   "segundoNombre": "Alberto",
5   "primerApellido": "Estepan",
6   "segundoApellido": "Ramon",
7   "correo": "jalbertoestepan@GMAIL.COM",
8   "direccion": "Santo Domingo Este",
9   "telefono": "8094028717",
10  "pais": "DO",
11  "gentilicio": "Dominican"
12 }
```

- DELETE /clientes/{id}: Eliminar un cliente

Quarkus Restful Api / EliminarCliente

SaveShare

DELETE

http://localhost:8080/clientes/5

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (2)

Test Results

200 OK

43 ms

102 B

Save Response

...

{ } JSON

Preview

Visualize

1

Cliente eliminado

En caso de no existir:

Quarkus Restful Api / EliminarCliente

SaveShare

DELETE

http://localhost:8080/clientes/88

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (2)

Test Results

404 Not Found

21 ms

160 B

Save Response

...

{ } JSON

Preview

Visualize

1

{

2

"error": "No se encontró un cliente para eliminar con este id: 88"

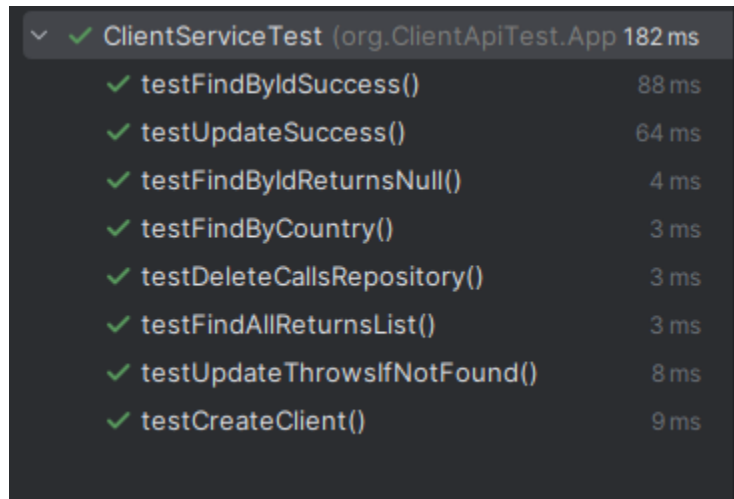
3

}

## Pruebas

----- Las pruebas unitarias y de integración están desarrolladas usando @QuarkusTest y se ubican en el directorio src/test/java.

**(intelli J- test)**

A screenshot of the IntelliJ IDEA test results window. It shows a list of test methods for the class ClientServiceTest, all of which passed successfully, indicated by green checkmarks. The total execution time for the class is 182 ms. The methods and their individual execution times are: testFindByIdSuccess() (88 ms), testUpdateSuccess() (64 ms), testFindByIdReturnsNull() (4 ms), testFindByCountry() (3 ms), testDeleteCallsRepository() (3 ms), testFindAllReturnsList() (3 ms), testUpdateThrowsIfNotFound() (8 ms), and testCreateClient() (9 ms).

✓ ClientServiceTest (org.ClientApiTest.App 182 ms)	
✓ testFindByIdSuccess()	88 ms
✓ testUpdateSuccess()	64 ms
✓ testFindByIdReturnsNull()	4 ms
✓ testFindByCountry()	3 ms
✓ testDeleteCallsRepository()	3 ms
✓ testFindAllReturnsList()	3 ms
✓ testUpdateThrowsIfNotFound()	8 ms
✓ testCreateClient()	9 ms