

Documentación Client Api – Ingeniero Backend Banreservas

Descripción General

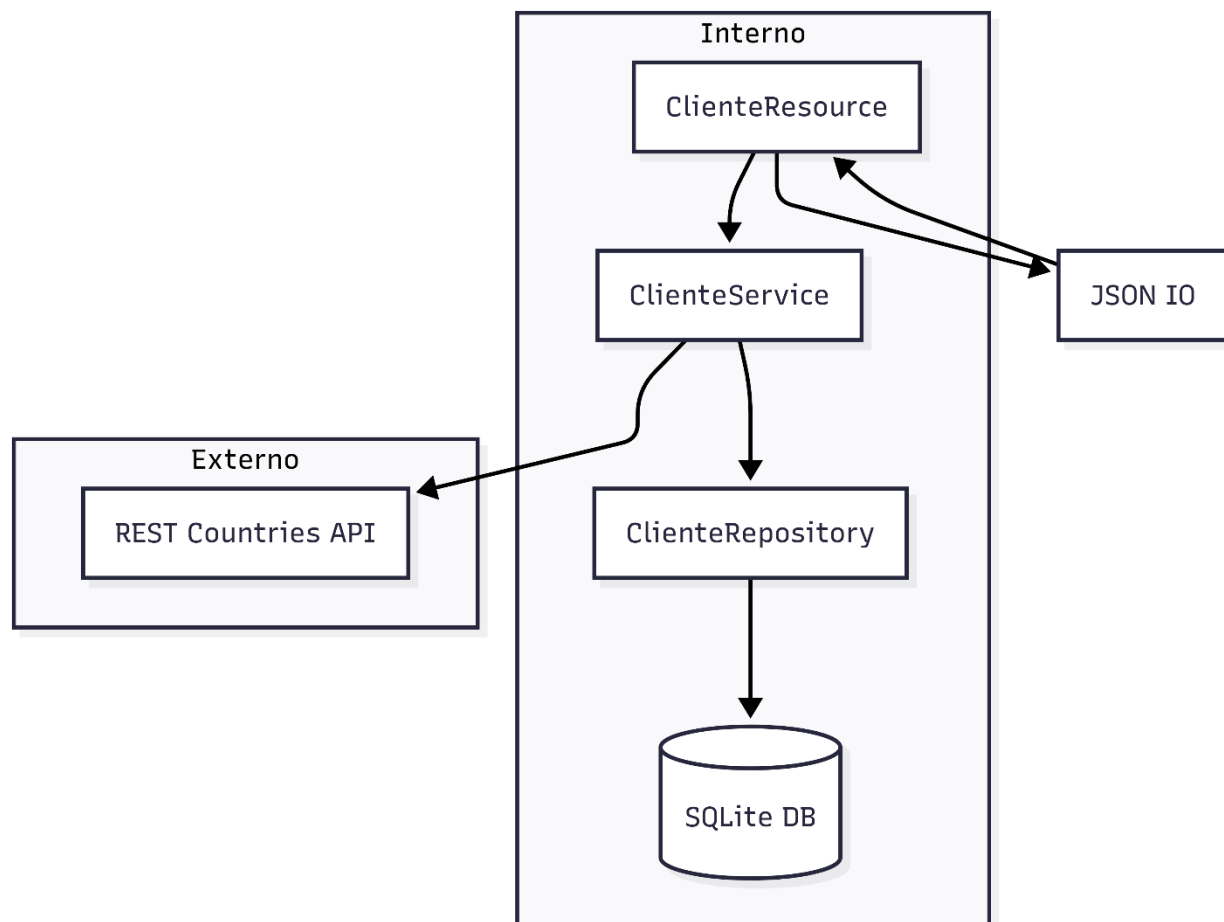
El proyecto creado (Client-API) desarrollado con Quarkus permite gestionar un grupo de clientes almacenados en una base de datos.

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

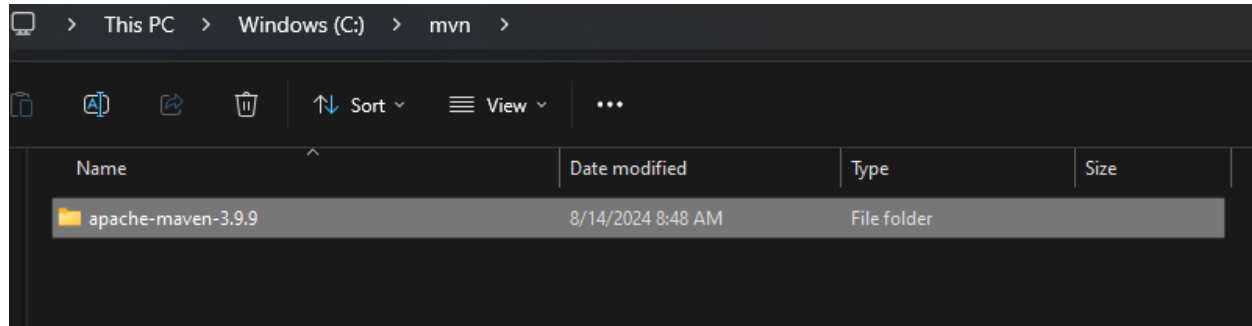
Tecnologías Utilizadas

- Java 17+
- Quarkus
- Jakarta REST (JAX-RS)
- SQLite - Panache ORM
- HTTP Client (Java 11+)
- REST Countries API (<https://restcountries.com>)

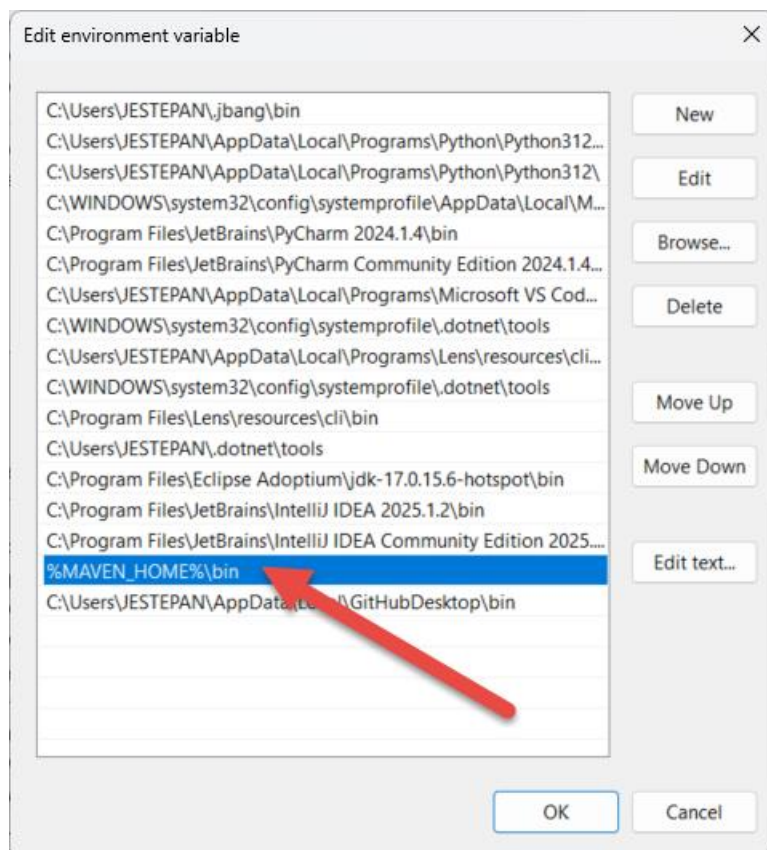
Instalación y ejecución del proyecto

Para la creación, ejecución y prueba de este proyecto, utilicé el framework sugerido en el documento de requerimientos el cual te permite implementar aplicaciones de manera rápida y te permite utilizar los componentes y extensiones que necesites.

Para su ejecución utilicé **Maven** instalado localmente



Variables de sistema:



System variables	
Variable	Value
JAVA_HOME	C:\Program Files\Eclipse Adoptium\jdk-17.0.15.6-hotspot
MAVEN_HOME	C:\mvn\apache-maven-3.9.9
NUMBER_OF_PROCESSORS	12

Versión utilizada de maven:

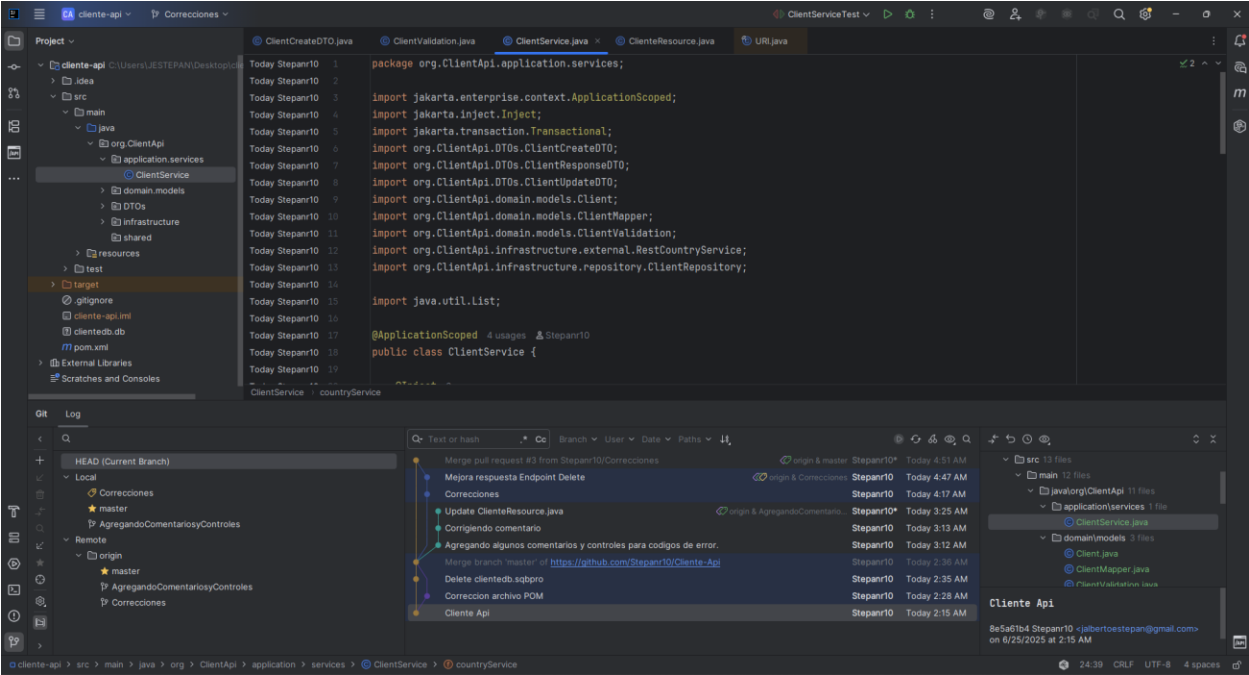
```
Terminal Local x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\JESTEPAN\Desktop\cliente-api> mvn -v
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfcd97d260186937)
Maven home: C:\mvn\apache-maven-3.9.9
Java version: 17.0.15, vendor: Eclipse Adoptium, runtime: C:\Program Files\Eclipse Adoptium\jdk-17.0.15.6-hotspot
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
PS C:\Users\JESTEPAN\Desktop\cliente-api>
```

IDE utilizado:

IntelliJ DEA (C.E.)



Para ejecutar el proyecto de manera local:

Clonar el repositorio alojado en

- <https://github.com/Stepanr10/Cliente-API.git>

Ejecutar en la terminal estando en la base del proyecto:

Mvn compile quarkus: dev

Esto habilitará el local host <http://localhost:8080/>

Para swagger: <http://localhost:8080/q/swagger-ui/#/>

```
PS C:\Users\JESTEPAN\Desktop\cliente-api> mvn compile quarkus:dev
[INFO] Scanning for projects...
[INFO]
[INFO] ----- org.example:cliente-api -----
[INFO] Building cliente-api 1.0.0-SNAPSHOT
[INFO] from pom.xml
[INFO] ----- [ jar ] -----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ cliente-api ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO]
[INFO] --- quarkus:3.9.3:generate-code (default) @ cliente-api ---
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ cliente-api ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- quarkus:3.9.3:dev (default-cli) @ cliente-api ---
[INFO] Invoking resources:3.3.1:testResources (default-testResources) @ cliente-api
[INFO] skip non existing resourceDirectory C:\Users\JESTEPAN\Desktop\cliente-api\src\test\resources
[INFO] Invoking compiler:3.13.0:testCompile (default-testCompile) @ cliente-api
[INFO] Nothing to compile - all classes are up to date.
[INFO] Listening for transport dt_socket at address: 5005
[INFO]
2025-06-25 10:20:24,117 WARN [io.qua.dat.dep.dev.DevServicesDatabaseProcessor] (build-3) Unable to start Dev Services for default datasource as this datasource type (other) does not support Dev Services
--
2025-06-25 10:20:28,982 WARN [io.qua.agr.run.AgroalConnectionConfigurer] (JPA Startup Thread) Agroal does not support detecting if a connection is still usable after an exception for database kind: other
2025-06-25 10:20:30,183 INFO [io.quarkus] (Quarkus Main Thread) cliente-api 1.0.0-SNAPSHOT on JVM (powered by Quarkus 3.9.3) started in 8.412s. Listening on: http://localhost:8080
2025-06-25 10:20:30,185 INFO [io.quarkus] (Quarkus Main Thread) Profile dev activated. Live Coding activated.
2025-06-25 10:20:30,187 INFO [io.quarkus] (Quarkus Main Thread) Installed features: [agroal, cdi, hibernate-orm, hibernate-orm-panache, hibernate-validator, narayana-jta, rest, rest-jackson, smallrye-context-propagation, smallrye-openapi, swagger-ui, vertx]
--
Tests paused
Press [a] to edit command line args (currently ''), [r] to resume testing, [o] toggle test output, [:] for the terminal, [h] for more options>
```

En el swagger se le colocaron anotaciones para detallar el objetivo de cada endpoint.

Swagger UI

/q/openapi

Explore

cliente-api API

1.0.0-SNAPSHOT

OAS 3.0

/q/openapi

Cientes

Con este Api se busca gestionar un CRUD de clientes utilizando una base de datos SQLite.

GET

/clientes

Obtener todos los clientes.

POST

/clientes

Crear un cliente.

GET

/clientes/pais/{codigo}

Obtener un cliente según el país.

GET

/clientes/{id}

Obtener un cliente según su ID en la base de datos.

PUT

/clientes/{id}

Actualizar datos de un cliente filtrandolo por su ID. (telefono, correo, dirección, pais)

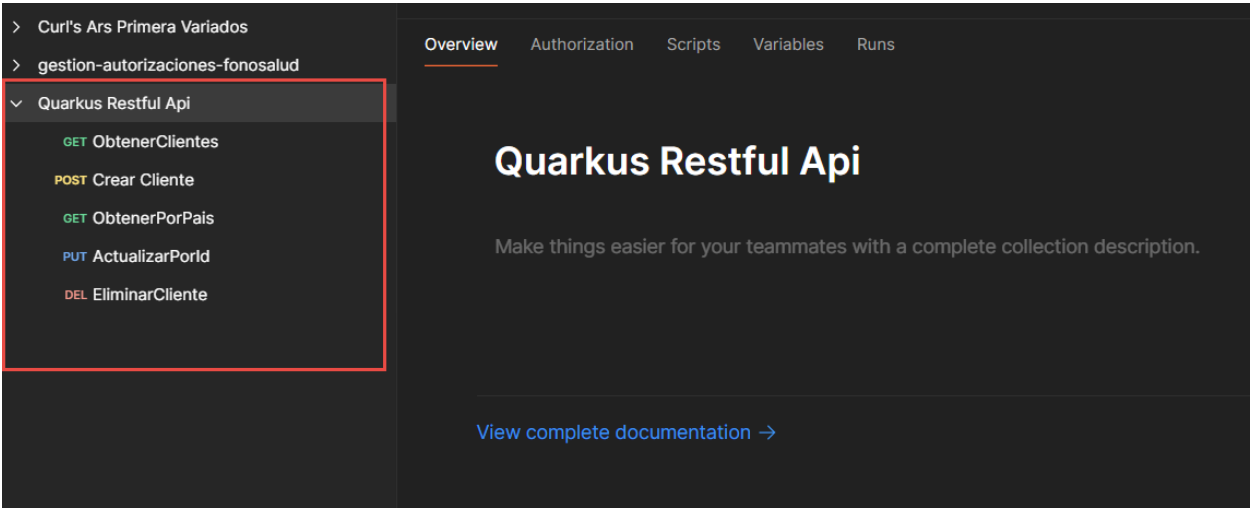
DELETE

/clientes/{id}

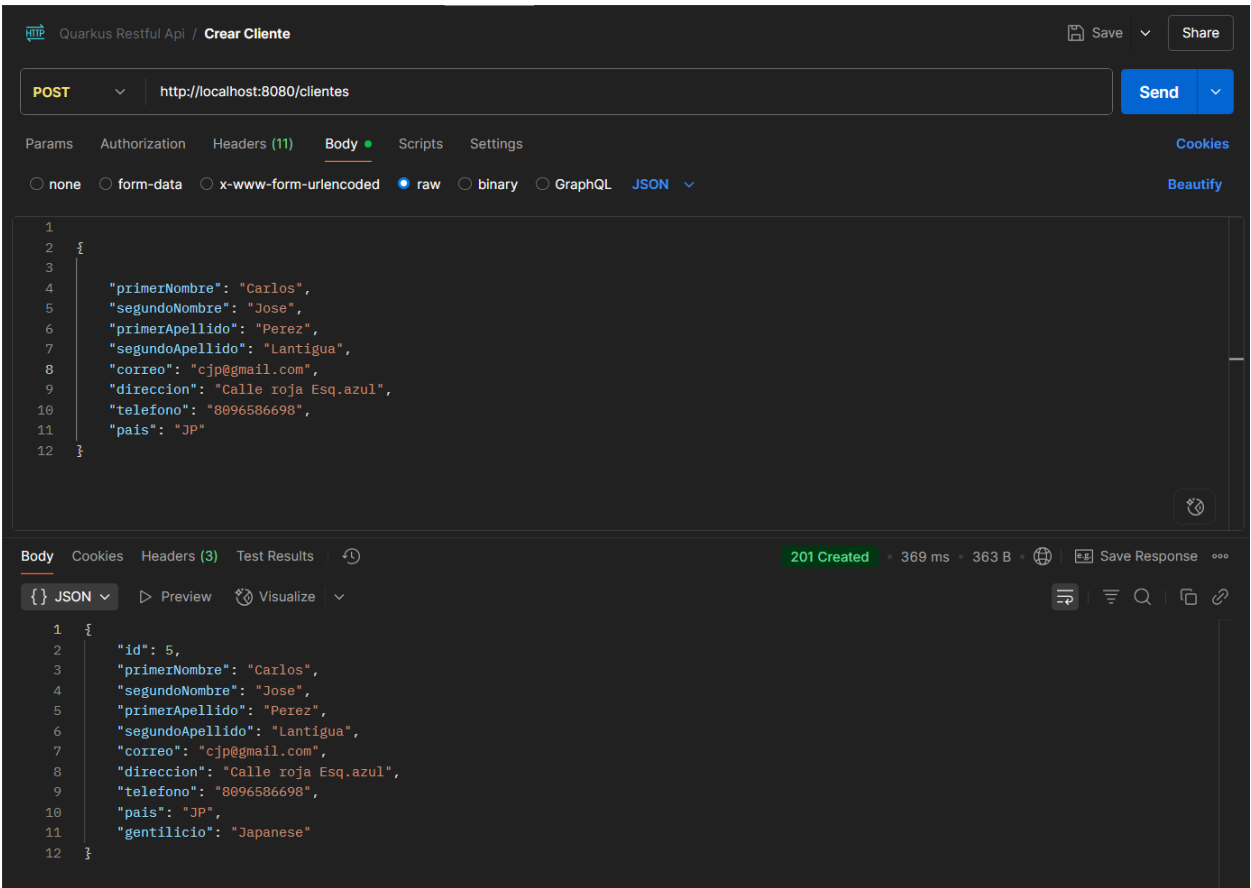
Eliminar un cliente buscandolo por su ID.

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": "Estepan",
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 a PUT request to `http://localhost:8080/clientes/2`. The request body is a JSON object with four fields: `correo`, `direccion`, `telefono`, and `pais`. The response is a 200 OK status with a JSON object containing all client details, including `id`, `primerNombre`, `segundoNombre`, `primerApellido`, `segundoApellido`, `correo`, `direccion`, `telefono`, `pais`, and `gentilicio`.

```
1 {
2   "correo": "jalbertoestepan@GMAIL.COM",
3   "direccion": "Santo Domingo Este",
4   "telefono": "8094028717",
5   "pais": "DO"
6 }
```

Body Cookies Headers (2) Test Results 200 OK • 755 ms • 326 B Save 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

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

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"

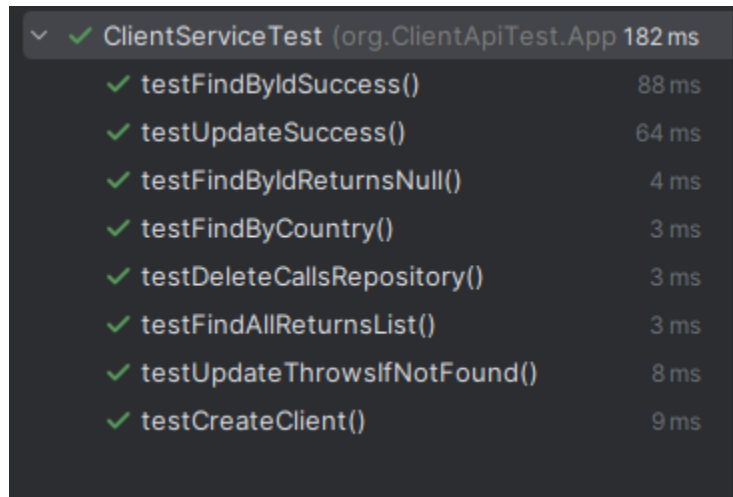
3

}

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: testFindByldSuccess() (88 ms), testUpdateSuccess() (64 ms), testFindByldReturnsNull() (4 ms), testFindByCountry() (3 ms), testDeleteCallsRepository() (3 ms), testFindAllReturnsList() (3 ms), testUpdateThrowsIfNotFound() (8 ms), and testCreateClient() (9 ms).

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