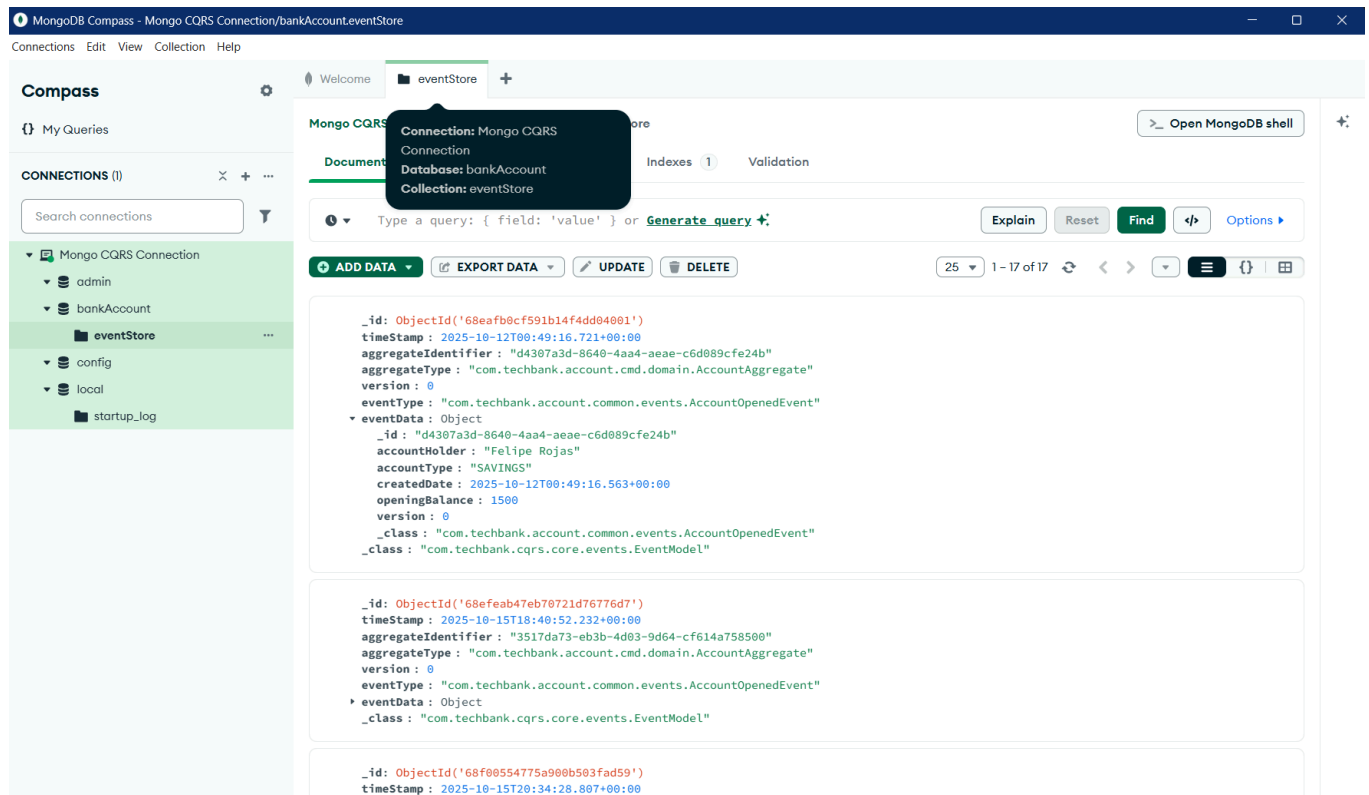


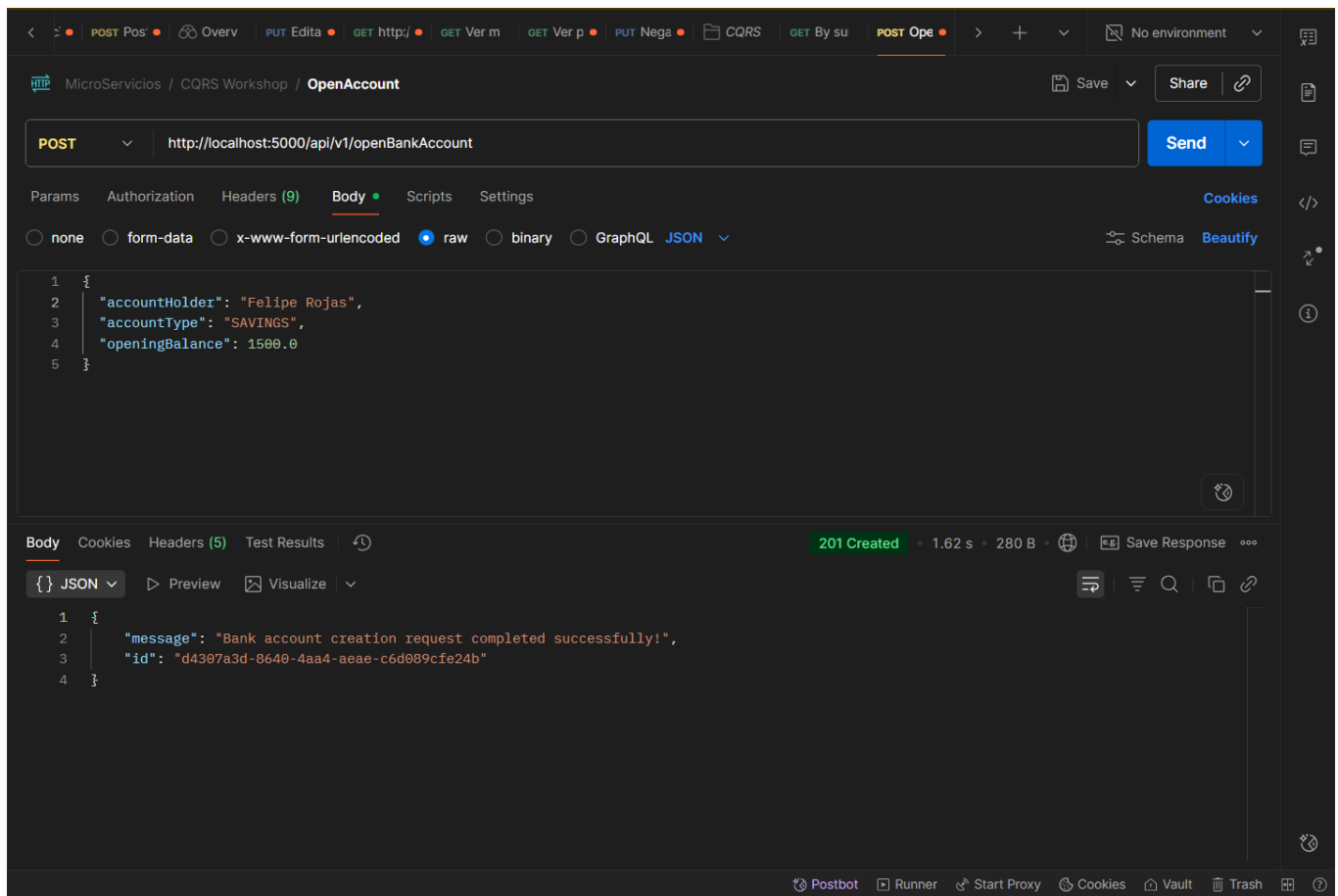
Felipe Rojas Prado - A00393918

Taller de Micro Servicios

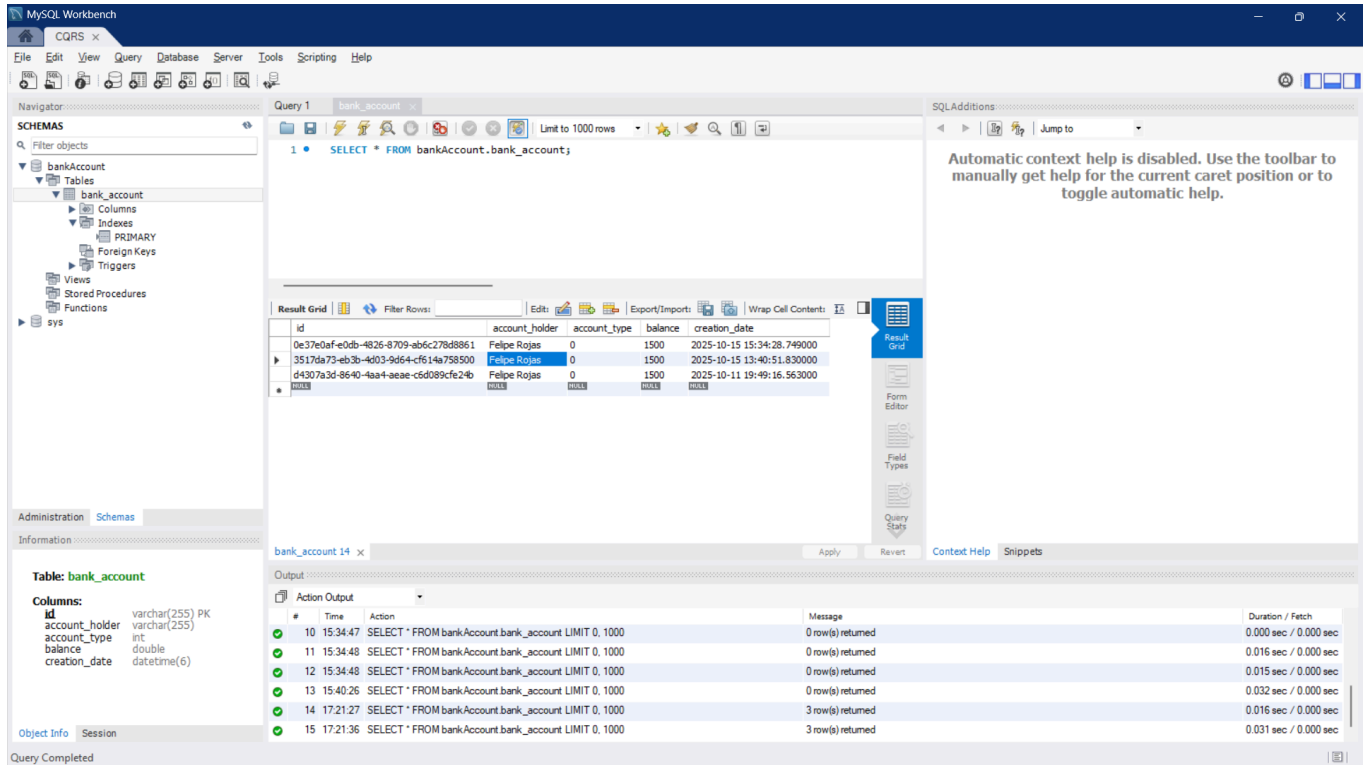
Capturas de pantalla

Mongo Compass

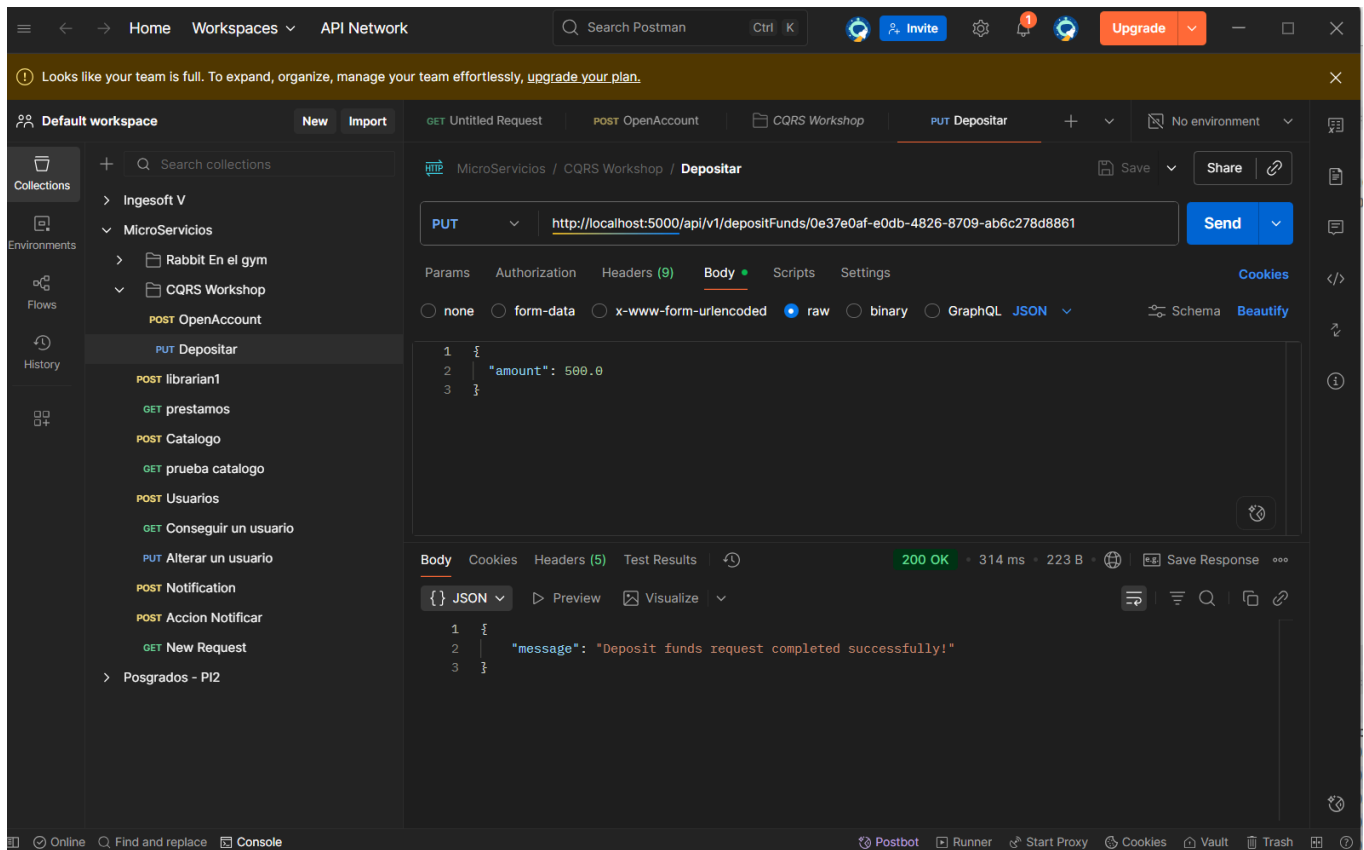




Revision en MySQL Workbench:



Deposito



Cambio del balance:

Query 1 **bank_account**

Limit to 1000 rows

```
1 • SELECT * FROM bankAccount.bank_account;
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	id	account_holder	account_type	balance	creation_date
▶	0e37e0af-e0db-4826-8709-ab6c278d8861	Felipe Rojas	0	2000	2025-10-15 15:34:28.749000
	3517da73-eb3b-4d03-9d64-cf614a758500	Felipe Rojas	0	1500	2025-10-15 13:40:51.830000
	d4307a3d-8640-4aa4-aeae-c6d089cfe24b	Felipe Rojas	0	1500	2025-10-11 19:49:16.563000
*	NULL	NULL	NULL	NULL	NULL

Sacar dinero

GET Untitled Reque | POST OpenAccoun | CQRS Workshc | PUT Depositar | PUT Sacar dinero | No environment

MicroServicios / CQRS Workshop / Sacar dinero

PUT <http://localhost:5000/api/v1/withdrawFunds/0e37e0af-e0db-4826-8709-ab6c278d8861> Send

Params | Authorization | Headers (9) | Body | Scripts | Settings | Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ☐ Schema Beautify

```
1 {
2   "ammount": 167
3 }
```

Body | Cookies | Headers (5) | Test Results | 200 OK • 51 ms • 224 B • Save Response

{ } JSON Preview Visualize

```
1 {
2   "message": "Withdraw funds request completed successfully!"
3 }
```

Postbot | Runner | Start Proxy | Cookies | Vault | Trash

Retiro del dinero:

Query 1 **bank_account** x

Limit to 1000 rows

1 • **SELECT * FROM bankAccount.bank_account;**

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | **Result Grid**

	id	account_holder	account_type	balance	creation_date
▶	0e37e0af-e0db-4826-8709-ab6c278d8861	Felipe Rojas	0	1833	2025-10-15 15:34:28.749000
	3517da73-eb3b-4d03-9d64-cf614a758500	Felipe Rojas	0	1500	2025-10-15 13:40:51.830000
	d4307a3d-8640-4aa4-aeae-c6d089cfe24b	Felipe Rojas	0	1500	2025-10-11 19:49:16.563000
*	NULL	NULL	NULL	NULL	NULL

Form Editor
Field Types
Query Stats

bank_account 44 x Apply Revert

Output

Action Output

#	Time	Action	Message
✓ 40	18:41:28	SELECT * FROM bankAccount.bank_account LIMIT 0, 1000	3 row(s) returned
✓ 41	18:41:28	SELECT * FROM bankAccount.bank_account LIMIT 0, 1000	3 row(s) returned
✓ 42	18:41:28	SELECT * FROM bankAccount.bank_account LIMIT 0, 1000	3 row(s) returned
✓ 43	18:41:29	SELECT * FROM bankAccount.bank_account LIMIT 0, 1000	3 row(s) returned
✓ 44	18:43:01	SELECT * FROM bankAccount.bank_account LIMIT 0, 1000	3 row(s) returned
✓ 45	18:46:26	SELECT * FROM bankAccount.bank_account LIMIT 0, 1000	3 row(s) returned

Cerrar cuenta

DELETE

http://localhost:5000/api/v1/closeBankAccount/3517da73-eb3b-4d03-9d64-cf614a758500

Send

Params Authorization Headers (7) Body Scripts Settings

Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

Schema Beautify

1 Ctrl+Alt+P for Postbot

Body Cookies Headers (5) Test Results

200 OK • 66 ms • 230 B • Save Response

{ } JSON

Preview Visualize

```
1 {  
2   "message": "Bank account closure request successfully completed!"  
3 }
```

Query 1 bank_account ×

Limit to 1000 rows

1 • `SELECT * FROM bankAccount.bank_account;`

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: IA

	id	account_holder	account_type	balance	creation_date
▶	0e37e0af-e0db-4826-8709-ab6c278d8861	Felipe Rojas	0	1833	2025-10-15 15:34:28.749000
	d4307a3d-8640-4aa4-aeae-c6d089cfe24b	Felipe Rojas	0	1500	2025-10-11 19:49:16.563000
*	NULL	NULL	NULL	NULL	NULL

bank_account 45 × Apply Revert

Obtener todas las cuentas

MicroServicios / CQRS Workshop / **Obtener todas las cuentas**

GET http://localhost:5001/api/v1/bankAccountLookup/ Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (5) Test Results 200 OK • 169 ms • 561 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "message": "Successfully returned 2 bank account(s)!",
3   "accounts": [
4     {
5       "id": "0e37e0af-e0db-4826-8709-ab6c278d8861",
6       "accountHolder": "Felipe Rojas",
7       "creationDate": "2025-10-15T20:34:28.749+00:00",
8       "accountType": "SAVINGS",
9       "balance": 1833.0
10    },
11    {
12      "id": "d4307a3d-8640-4aa4-aeae-c6d089cfe24b",
13      "accountHolder": "Felipe Rojas",
14      "creationDate": "2025-10-12T00:49:16.563+00:00",
15      "accountType": "SAVINGS",
16      "balance": 1500.0
17    }
18  ]
19 }
```

Postbot Runner Start Proxy Cookies Vault Trash

Buscar por ID

MicroServicios / CQRS Workshop / **Buscar por ID** Save Share

GET http://localhost:5001/api/v1/bankAccountLookup/byId/0e37e0af-e0db-4826-8709-ab6c278d8861 Send

Params Authorization Headers (8) **Body** Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

Body Cookies Headers (5) Test Results 200 OK • 65 ms • 391 B • Save Response

{ } JSON Preview Visualize

```
1 {
2   "message": "Successfully returned bank account!",
3   "accounts": [
4     {
5       "id": "0e37e0af-e0db-4826-8709-ab6c278d8861",
6       "accountHolder": "Felipe Rojas",
7       "creationDate": "2025-10-15T20:34:28.749+00:00",
8       "accountType": "SAVINGS",
9       "balance": 1833.0
10    }
11  ]
12 }
```

Postbot Runner Start Proxy Cookies Vault Trash

Buscar por nombre

MicroServicios / CQRS Workshop / **Buscar por nombre** Save Share

GET http://localhost:5001/api/v1/bankAccountLookup/byHolder/Felipe%20Rojas Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (5) Test Results 200 OK • 288 ms • 720 B • Save Response

{} **JSON** Preview Visualize

```
8      "accountType": "SAVINGS",
9      "balance": 1833.0
10    },
11    {
12      "id": "d4307a3d-8640-4aa4-aeae-c6d089cfe24b",
13      "accountHolder": "Felipe Rojas",
14      "creationDate": "2025-10-12T00:49:16.563+00:00",
15      "accountType": "SAVINGS",
16      "balance": 1500.0
17    },
18    {
19      "id": "d88b1582-7558-49c6-94b0-3d565633af9b",
20      "accountHolder": "Felipe Rojas",
21      "creationDate": "2025-10-16T00:18:37.601+00:00",
22      "accountType": "SAVINGS",
23      "balance": 100.0
24    }
25  ]
26 }
```

Postbot Runner Start Proxy Cookies Vault Trash + ?

Buscar por balance mayor a 1000

MicroServicios / CQRS Workshop / Por balance (Mayor a 1000)

GET http://localhost:5001/api/v1/bankAccountLookup/withBalance/GREATER_THAN/1000.0 Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (6) Test Results 200 OK • 21 ms • 605 B Save Response

{ JSON Preview Visualize

```
4 {
5   "id": "0e37e0af-e0db-4826-8709-ab6c278d8861",
6   "accountHolder": "Felipe Rojas",
7   "creationDate": "2025-10-15T20:34:28.749+00:00",
8   "accountType": "SAVINGS",
9   "balance": 1833.0
10 },
11 {
12   "id": "d4307a3d-8640-4aa4-aeae-c6d089cfe24b",
13   "accountHolder": "Felipe Rojas",
14   "creationDate": "2025-10-12T00:49:16.563+00:00",
15   "accountType": "SAVINGS",
16   "balance": 1500.0
17 }
18 ]
19 }
```

Postbot Runner Start Proxy Cookies Vault Trash

Buscar por balance menor a 1000

MicroServicios / CQRS Workshop / **Por balance (Menor a 1000)**

GET Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

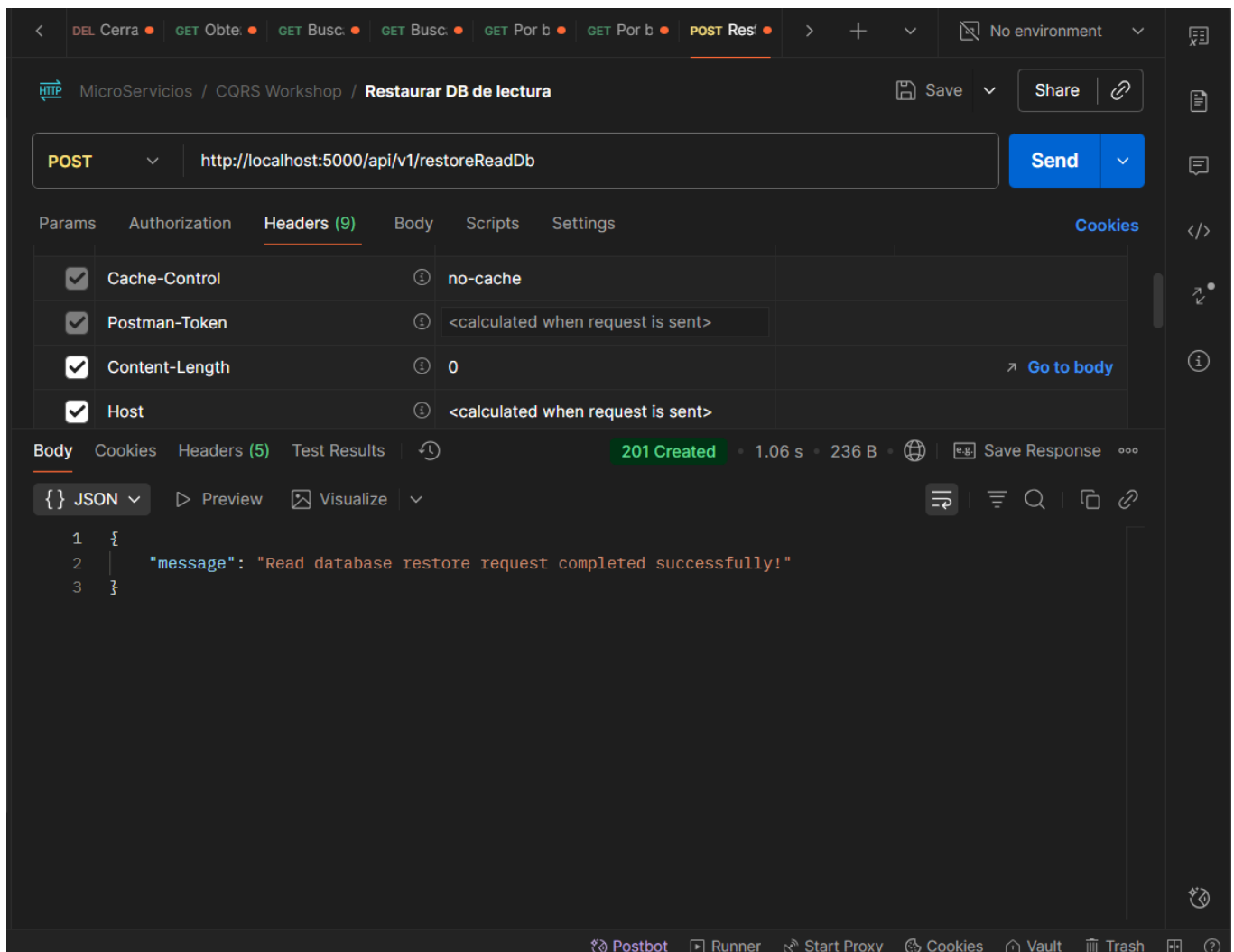
Body Cookies Headers (6) Test Results 200 OK • 24 ms • 439 B • Save Response

{ } JSON Preview Visualize

```
1 {
2   "message": "Successfully returned 1 bank account(s)!",
3   "accounts": [
4     {
5       "id": "d88b1582-7558-49c6-94b0-3d565633af9b",
6       "accountHolder": "Felipe Rojas",
7       "creationDate": "2025-10-16T00:18:37.601+00:00",
8       "accountType": "SAVINGS",
9       "balance": 100.0
10    }
11  ]
12 }
```

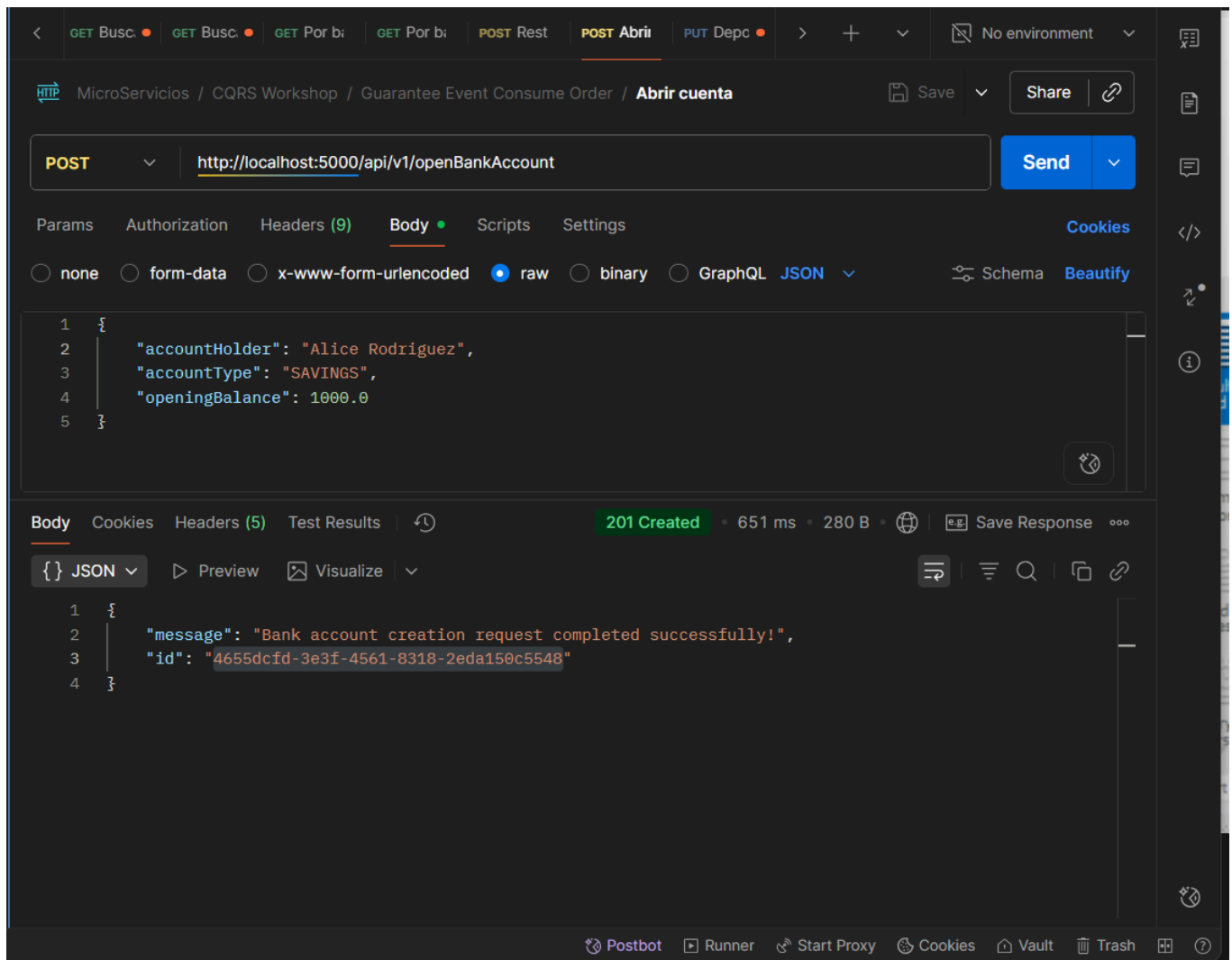
Postbot Runner Start Proxy Cookies Vault Trash

Restaurar la DB de lectura



Parte 2

Crear cuenta



Depositar dinero

Query 1 bank_account x

Limit to 1000 rows

```
1 • SELECT * FROM bankAccount.bank_account;
```

Result Grid

	id	account_holder	account_type	balance	creation_date
	0e37e0af-e0db-4826-8709-ab6c278d8861	Felipe Rojas	0	497	2025-10-15 15:34:28.749000
▶	4655dcfd-3e3f-4561-8318-2eda150c5548	Alice Rodriguez	0	3500	2025-10-15 21:07:31.805000
	d4307a3d-8640-4aa4-aeae-c6d089cfe24b	Felipe Rojas	0	1500	2025-10-11 19:49:16.563000
	d88b1582-7558-49c6-94b0-3d565633af9b	Felipe Rojas	0	100	2025-10-15 19:18:37.601000
*	NULL	NULL	NULL	NULL	NULL

bank_account 56 x

Apply Revert

Result Grid

Form Editor

Field Types

Query Stats

Retirar dinero

GET Busc. • GET Por bi • GET Por bi • POST Rest • POST Abrii • PUT Depc • PUT Retir. •

> + ▾

No environment ▾

MicroServicios / CQRS Workshop / Guarantee Event Consume Order / Retirar dinero

Save ▾ Share

PUT ▾ http://localhost:5000/api/v1/withdrawFunds/4655dcfd-3e3f-4561-8318-2eda150c5548

Send ▾

Params Authorization Headers (9) Body • Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▾

Schema Beautify

```
1 {
2   "amount": 666.0
3 }
```

Body Cookies Headers (5) Test Results ↻

200 OK • 52 ms • 224 B • Save Response ⋮

{ } JSON ▾

Preview Visualize ▾

```
1 {
2   "message": "Withdraw funds request completed successfully!"
3 }
```

Postbot Runner Start Proxy Cookies Vault Trash ?

GET Por bi

GET Por bi

POST Rest

POST Abrii

PUT Depc

PUT Retira

DEL Cerra

> + v

No environment v

MicroServicios / CQRS Workshop / Guarantee Event Consume Order / Cerrar cuenta

Save vShare

DELETE vhttp://localhost:5000/api/v1/closeBankAccount/4655dcfd-3e3f-4561-8318-2eda150c5548Send v

ParamsAuthorizationHeaders (7)BodyScriptsSettingsCookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

BodyCookiesHeaders (5)Test Results🕒200 OK • 40 ms • 230 B • 🌐📄 Save Response ...

{ } JSON v

▶ Preview🖼 Visualize v

🔍📄🔗

1 {

2 | "message": "Bank account closure request successfully completed!"

3 }

🔄

PostbotRunnerStart ProxyCookiesVaultTrash ?

Cierre de cuenta satisfactorio

Query 1 | bank_account x

Limit to 1000 rows

```
SELECT * FROM bankAccount.bank_account;
```

	id	account_holder	account_type	balance	creation_date
▶	0e37e0af-e0db-4826-8709-ab6c278d8861	Felipe Rojas	0	497	2025-10-15 15:34:28.749000
	d4307a3d-8640-4aa4-aeae-c6d089cfe24b	Felipe Rojas	0	1500	2025-10-11 19:49:16.563000
	d88b1582-7558-49c6-94b0-3d565633af9b	Felipe Rojas	0	100	2025-10-15 19:18:37.601000
*	NULL	NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

Form Editor
Field Types
Query Stats

bank_account 58 x Apply Revert

Lectura de cuenta ya cerrada

MicroServicios / CQRS Workshop / Guarantee Event Consume Order / Leer DB

GET http://localhost:5001/api/v1/bankAccountLookup/byId/4655dcfd-3e3f-4561-8318-2eda150c5548 Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (3) Test Results 204 No Content • 141 ms • 112 B • Save Response

Raw Preview Visualize

1

Postbot Runner Start Proxy Cookies Vault Trash

Restore DB

La cuenta fue eliminada anteriormente

POST Rest

POST Abrii

PUT Depc

PUT Retira

DEL Cerrai

GET Leer

POST Retc

>

+

▼

No environment

MicroServicios / CQRS Workshop / Guarantee Event Consume Order / Retore DB

Save

Share

POST

http://localhost:5000/api/v1/restoreReadDb

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (5)

Test Results

201 Created

145 ms

236 B

Save Response

{ } JSON

Preview

Visualize

1 {

2 |

3 |

"message": "Read database restore request completed successfully!"

Postbot

Runner

Start Proxy

Cookies

Vault

Trash

Y ahora podemos ver que fue restaurada con el mensaje

The screenshot shows a REST client interface with a dark theme. At the top, there's a breadcrumb trail: `MicroServicios / CQRS Workshop / Guarantee Event Consume Order / Leer DB`. Below this, the request method is `GET` and the URL is `http://localhost:5001/api/v1/bankAccountLookup/byId/853f4024-ed06-4712-ab74-9e9c07915bfa`. The `Send` button is highlighted in blue. Below the URL bar, there are tabs for `Params`, `Authorization`, `Headers (7)`, `Body`, `Scripts`, and `Settings`. The `Params` tab is active, showing a table with columns `Key`, `Value`, and `Description`. Below the table, there's a `Query Params` section. The `Body` tab is also visible, showing a `200 OK` status with a response time of `44 ms` and a size of `394 B`. The response body is in `JSON` format and contains the following data:

```
1 {
2   "message": "Successfully returned bank account!",
3   "accounts": [
4     {
5       "id": "853f4024-ed06-4712-ab74-9e9c07915bfa",
6       "accountHolder": "Alice Rodriguez",
7       "creationDate": "2025-10-16T01:57:15.585+00:00",
8       "accountType": "SAVINGS",
9       "balance": 6000.0
10    }
11  ]
12 }
```

At the bottom of the interface, there's a footer with icons for `Postbot`, `Runner`, `Start Proxy`, `Cookies`, `Vault`, `Trash`, and a help icon.

Query 1 **bank_account** x

Limit to 1000 rows

1 • **SELECT * FROM** bankAccount.bank_account;

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	account_holder	account_type	balance	creation_date
	0e37e0af-e0db-4826-8709-ab6c278d8861	Felipe Rojas	0	1332	2025-10-15 15:34:28.749000
▶	853f4024-ed06-4712-ab74-9e9c07915bfa	Alice Rodriguez	0	6000	2025-10-15 20:57:15.585000
	d4307a3d-8640-4aa4-aeae-c6d089cfe24b	Felipe Rojas	0	1500	2025-10-11 19:49:16.563000
	d88b1582-7558-49c6-94b0-3d565633af9b	Felipe Rojas	0	100	2025-10-15 19:18:37.601000
*	NULL	NULL	NULL	NULL	NULL

bank_account 59 x Apply Revert

Explicación de cómo se garantiza el orden de los eventos y por qué es importante en CQRS y Event Sourcing.

El orden de los eventos se garantiza asignando todas las events de una misma entidad (por ejemplo, una cuenta) a la misma «canaleta» de Kafka (partición) usando la misma key (aggregateId). Kafka mantiene el orden de los mensajes dentro de cada partición, por eso si siempre se publica con la misma key, los consumidores verán los eventos de esa entidad en el mismo orden en que fueron producidos. Por otro lado, esto es importante en CQRS + Event Sourcing porque la proyección (read model) se reconstruye aplicando los eventos en el mismo orden en que ocurrieron para obtener el mismo estado determinista de la entidad. Con lo que, ordenar = publicar con la misma key (aggregateId) para que Kafka entregue en orden dentro de la partición; es crítico para reproducir correctamente el estado de una entidad.

Resumen

Se centralizó la publicación de eventos en un único topic (account-events) y se garantizó el orden por entidad publicando siempre con key = aggregateId . Además se unificó el consumidor en el servicio de consulta para demultiplexar por un header eventType y se ajustó la lógica de republish/restore para conservar key y orden. Con esto la proyección (read model) puede reconstruirse determinísticamente desde el event store.

Objetivos

- Garantizar que todos los eventos de una misma entidad (aggregate) se consuman y apliquen en el mismo orden.
- Permitir reproducir (replay) todos los eventos desde el event store para reconstruir la read model.

Para Verificación:

1. Arrancar infra (Kafka, ZK, Mongo, MySQL, Kafdrop).
2. Arrancar servicios:
 - Command app (puerto 5000)
 - Query app (puerto 5001)
3. Crear cuenta (POST /api/v1/openBankAccount) — guardar `id`.
4. Hacer operaciones: PUT /api/v1/depositFunds/{id}, PUT /api/v1/withdrawFunds/{id}.
5. En Kafdrop, abrir `account-events` y comprobar:
 - Cada registro tiene `key == <aggregateId>`.
 - Header `eventType` presente.
6. Consultar read DB: GET <http://localhost:5001/api/v1/bankAccountLookup/byId/{id}> y verificar balances.
7. Test de replay:
 - Detener `account.query`, opcionalmente limpiar read DB, arrancar `account.query`.
 - POST <http://localhost:5000/api/v1/restoreReadDb> (republica eventos desde Mongo).
 - Verificar que los eventos republicados aparecen en Kafdrop y la read DB se reconstruye correctamente.

Repositorio del trabajo: <https://github.com/rojas435/cqrs-event-sourcing-kafka.git>