

Engenharia de Softwares Escaláveis

Domain-Driven Design (DDD) e Arquitetura de
Softwares Escaláveis com Java

Agenda

Etapa 5: Armazenamento, Publicação e Persistência de Domain Events em Java utilizando Event Sourcing.

- Laboratório.





Laboratório

Suponha que temos um aplicativo de contabilidade. A função desse aplicativo é registrar todas as transações que ocorrem em uma conta. A conta é um agregado nesse aplicativo.

Agora, vamos supor que as seguintes transações ocorram em uma determinada conta:

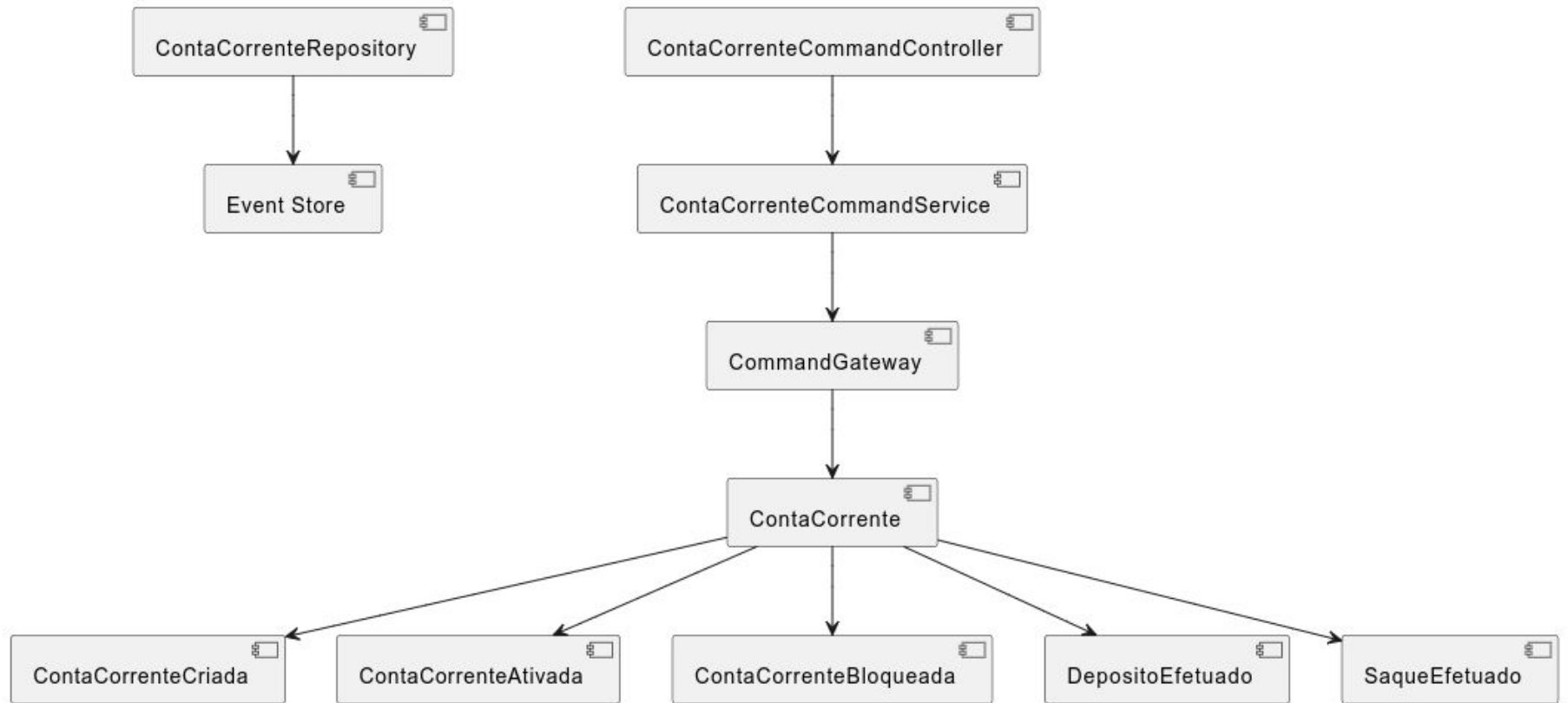
1. Criação de conta com saldo inicial de 100 dinheiros.
2. Conta definida como ATIVA.
3. Um amigo devolveu parte do dinheiro por ter dividido uma refeição. Depósito de 55 dinheiros na conta.
4. Você comprou um sorvete para sua namorada. Saque de 20 dinheiros.
5. A namorada não gostou do sorvete. Então, você teve que comprar outro. Saque de 50 dinheiros.
6. Você se esqueceu de pagar o aluguel. O proprietário aplicou uma multa. Saque de 100 dinheiros.
7. Conta bloqueada devido a saldo negativo.
8. Deposite 100 dinheiros na conta. Respire aliviado.
9. Conta definida como ATIVA.

Conta#	Quantia	Moeda	Status
ABCD567823	85,00	dinheiros	ATIVO

ID_Agregado	Evento	Carga útil do evento
ABCD567823	Evento de ativação da conta	{...}
ABCD567823	Evento de atualização do saldo da conta	{...}
ABCD567823	Evento de Conta Retida	{...}
ABCD567823	Evento de atualização do saldo da conta	{...}
ABCD567823	Evento de atualização do saldo da conta	{...}
ABCD567823	Evento de atualização do saldo da conta	{...}
ABCD567823	Evento de atualização do saldo da conta	{...}
ABCD567823	Evento de ativação da conta	{...}
ABCD567823	Evento de atualização do saldo da conta	{...}
ABCD567823	Evento de criação de conta	{...}

Event Sourcing com Axon Framework

Command Side



1

```
application.properties x
Source History
1 spring.application.name=DR4_E5_A1
2 spring.datasource.url=jdbc:h2:file:./h2_databases/DR4_1
3 spring.datasource.driverClassName=org.h2.Driver
4 spring.datasource.username=sa
5 spring.datasource.password=password
6 spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
7 spring.h2.console.enabled=true
8 spring.jpa.hibernate.ddl-auto=update
```

spring-boot-starter-parent → 3.3.2
spring-boot-starter-actuator
spring-boot-starter-data-jpa
spring-boot-starter-web
axon-spring-boot-starter → 4.10.0
com.h2database
springdoc-openapi-starter-webmvc-ui
spring-boot-starter-test

2

```
pom.xml [DR4_E5_A1] x
1 <?xml version="1.0" encoding="UTF-8" ?>
20
21 <dependency>
22   <groupId>org.springframework.boot</groupId>
23   <artifactId>spring-boot-starter-actuator</artifactId>
24 </dependency>
25 <dependency>
26   <groupId>org.springframework.boot</groupId>
27   <artifactId>spring-boot-starter-data-jpa</artifactId>
28 </dependency>
29 <dependency>
30   <groupId>org.springframework.boot</groupId>
31   <artifactId>spring-boot-starter-web</artifactId>
32 </dependency>
33 <dependency>
34   <groupId>org.axonframework</groupId>
35   <artifactId>axon-spring-boot-starter</artifactId>
36   <version>4.10.0</version>
37   <exclusions>
38     <exclusion>
39       <groupId>org.axonframework</groupId>
40       <artifactId>axon-server-connector</artifactId>
41     </exclusion>
42   </exclusions>
43 </dependency>
44 <dependency>
45   <groupId>com.h2database</groupId>
46   <artifactId>h2</artifactId>
47   <scope>runtime</scope>
48 </dependency>
49 <dependency>
50   <groupId>org.springdoc</groupId>
51   <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
52   <version>2.6.0</version>
53 </dependency>
```

3

```
Estado.java x
Source History
1 package br.edu.infnet.contas.domain;
2
3 public enum Estado {
4     CRIADA, ATIVADA, BLOQUEADA
5 }
```

Criação do
agregado

4

```
Estado.java x ContaCorrente.java x
Source History
1 package br.edu.infnet.contas.domain;
2
3 import ...8 lines
11
12 @Aggregate
13 public class ContaCorrente {
14
15     @AggregateIdentifier
16     private String id;
17     private String titular;
18     private double saldo;
19     private String estado;
20
21     public ContaCorrente() {
22     }
23
24     @CommandHandler
25     public ContaCorrente(CriarContaCommand comando) {
26         AggregateLifecycle.apply(new ContaCorrenteCriada(comando.id, comando.titular, comando.saldo));
27     }
}
```


5

```
BaseCommand.java x
Source History
1 package br.edu.infnet.contas.comandos;
2
3 import org.axonframework.modelling.command.TargetAggregateIdentifier;
4
5 public class BaseCommand<T> {
6
7     @TargetAggregateIdentifier
8     public final T id;
9
10    public BaseCommand(T id) {
11        this.id = id;
12    }
13
14 }
```

Criação dos
comandos

6

```
BaseCommand.java x CriarContaCommand.java x
Source History
1 package br.edu.infnet.contas.comandos;
2
3 public class CriarContaCommand extends BaseCommand<String> {
4
5     public final String titular;
6     public final double saldo;
7
8     public CriarContaCommand(String id, String titular, double depositoInicial) {
9         super(id);
10        this.titular = titular;
11        this.saldo = depositoInicial;
12    }
13 }
```

Criação dos
comandos

7

```
DepositarCommand.java x
Source History
1 package br.edu.infnet.contas.commands;
2
3 public class DepositarCommand extends BaseCommand<String> {
4
5     public final double valor;
6
7     public DepositarCommand(String id, double valor) {
8         super(id);
9         this.valor = valor;
10    }
11 }
```

8

```
DepositarCommand.java x SacarCommand.java x
Source History
1 package br.edu.infnet.contas.commands;
2
3 public class SacarCommand extends BaseCommand<String> {
4
5     public final double valor;
6
7     public SacarCommand(String id, double valor) {
8         super(id);
9         this.valor = valor;
10    }
11 }
```

Criação dos
eventos

9

```
BaseEvent.java x
Source History
1 package br.edu.infnet.contas.eventos;
2
3 public class BaseEvent<T> {
4
5     public final T id;
6
7     public BaseEvent(T id) {
8         this.id = id;
9     }
10 }
11
```

10

```
BaseEvent.java x ContaCorrenteCriada.java x
Source History
1 package br.edu.infnet.contas.eventos;
2
3 public class ContaCorrenteCriada extends BaseEvent<String> {
4
5     public final String titular;
6     public final double saldo;
7
8     public ContaCorrenteCriada(String id, String titular, double depositoInicial) {
9         super(id);
10        this.titular = titular;
11        this.saldo = depositoInicial;
12    }
13 }
```

Criação dos
eventos

11

```
DepositoEfetuado.java x
Source History
1 package br.edu.infnet.contas.events;
2
3 public class DepositoEfetuado extends BaseEvent<String> {
4     public final double valor;
5
6     public DepositoEfetuado(String id, double valor) {
7         super(id);
8         this.valor = valor;
9     }
10 }
```

12

```
DepositoEfetuado.java x SaqueEfetuado.java x
Source History
1 package br.edu.infnet.contas.events;
2
3 public class SaqueEfetuado extends BaseEvent<String> {
4     public final double valor;
5
6     public SaqueEfetuado(String id, double valor) {
7         super(id);
8         this.valor = valor;
9     }
10 }
```


13

```
ContaCorrente.java x
Source History

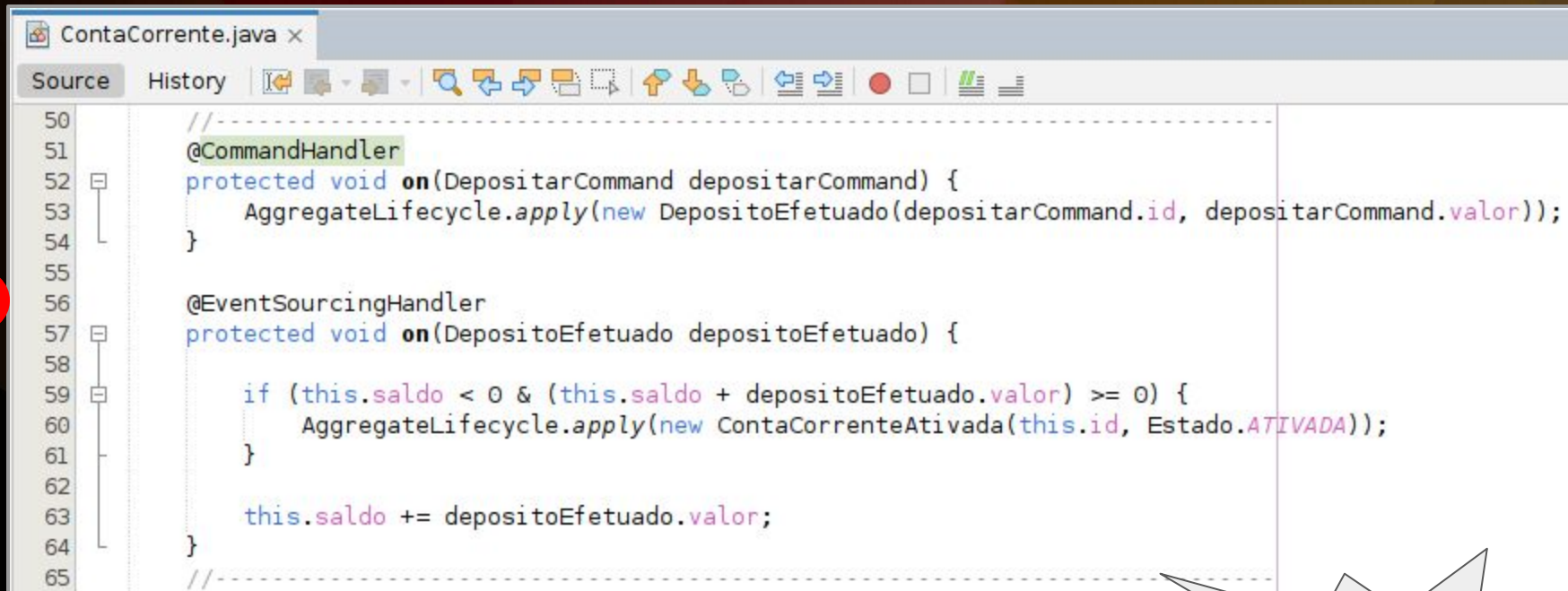
33 //-----
34 @CommandHandler
35 public ContaCorrente(CriarContaCommand comando) {
36     AggregateLifecycle.apply(new ContaCorrenteCriada(comando.id, comando.titular, comando.saldo))
37 }
38
39 @EventSourcingHandler
40 protected void on(ContaCorrenteCriada evento) {
41     this.id = evento.id;
42     this.titular = evento.titular;
43     this.saldo = evento.saldo;
44     this.estado = String.valueOf(Estado.CRIADA);
45
46     AggregateLifecycle.apply(new ContaCorrenteAtivada(this.id, Estado.ATIVADA));
47 }
48
49 @EventSourcingHandler
50 protected void on(ContaCorrenteAtivada evento) {
51     this.estado = String.valueOf(evento.estado);
52 }
53 //-----
54 //-----
55 @CommandHandler
56 protected void on(DepositarCommand depositarCommand) {
57     AggregateLifecycle.apply(new DepositoEfetuado(depositarCommand.id, depositarCommand.valor));
58 }
59
60 @EventSourcingHandler
61 protected void on(DepositoEfetuado depositoEfetuado) {
62
63     if (this.saldo < 0 & (this.saldo + depositoEfetuado.valor) >= 0) {
64         AggregateLifecycle.apply(new ContaCorrenteAtivada(this.id, Estado.ATIVADA));
65     }
66 }
```

Criação dos
manipuladores

```
ContaCorrente.java x
Source History
29 //-----
30 @CommandHandler
31 public ContaCorrente(CriarContaCommand comando) {
32     AggregateLifecycle.apply(new ContaCorrenteCriada(comando.id, comando.titular, comando.saldo));
33 }
34
35 @EventSourcingHandler
36 protected void on(ContaCorrenteCriada evento) {
37     this.id = evento.id;
38     this.titular = evento.titular;
39     this.saldo = evento.saldo;
40     this.estado = String.valueOf(Estado.CRIADA);
41
42     AggregateLifecycle.apply(new ContaCorrenteAtivada(this.id, Estado.ATIVADA));
43 }
44
45 @EventSourcingHandler
46 protected void on(ContaCorrenteAtivada evento) {
47     this.estado = String.valueOf(evento.estado);
48 }
49 //-----
```

Criação dos
manipuladores

15



```
50 //-----
51 @CommandHandler
52 protected void on(DepositarCommand depositarCommand) {
53     AggregateLifecycle.apply(new DepositoEfetuado(depositarCommand.id, depositarCommand.valor));
54 }
55
56 @EventSourcingHandler
57 protected void on(DepositoEfetuado depositoEfetuado) {
58
59     if (this.saldo < 0 & (this.saldo + depositoEfetuado.valor) >= 0) {
60         AggregateLifecycle.apply(new ContaCorrenteAtivada(this.id, Estado.ATIVADA));
61     }
62
63     this.saldo += depositoEfetuado.valor;
64 }
65 //-----
```

Criação dos
manipuladores

ContaCorrente.java x

Source History

```
66 //-----
67 @CommandHandler
68 protected void on(SacarCommand sacarCommand) {
69     AggregateLifecycle.apply(new SaqueEfetuado(sacarCommand.id, sacarCommand.valor));
70 }
71
72 @EventSourcingHandler
73 protected void on(SaqueEfetuado saqueEfetuado) {
74
75     if (this.saldo >= 0 & (this.saldo - saqueEfetuado.valor) < 0) {
76         AggregateLifecycle.apply(new ContaCorrenteBloqueada(this.id, Estado.BLOQUEADA));
77     }
78     this.saldo -= saqueEfetuado.valor;
79 }
80
81 @EventSourcingHandler
82 protected void on(ContaCorrenteBloqueada contaCorrenteBloqueada) {
83     this.estado = String.valueOf(contaCorrenteBloqueada.estado);
84 }
85 //-----
```

Criação dos
manipuladores

	@CommandHandler	@EventSourcingHandler
Propósito	Receber comandos, tomar decisões de negócios e aplicar eventos.	Receber eventos e modificar o estado do agregado.
Entrada	Um objeto de Comando.	Um objeto de Evento.
Resultado	Chama AggregateLifecycle.apply(NewEvent)	Modifica as variáveis de estado.
Altera Estado	Não. Ele nunca deve modificar o estado diretamente.	Sim. Este é o único local onde o estado deve ser alterado.
Quando é invocado?	Quando um novo Comando é despachado para o agregado.	<ol style="list-style-type: none"> 1. Imediatamente após um CommandHandler aplicar um evento. 2. Durante a recarga do agregado, para cada evento no histórico.
Obrigatoriedade	Necessário para permitir que o agregado reaja a comandos.	Necessário para construir o estado do agregado a partir dos eventos.

17

```
ContaCorrenteCommandService.java x
Source History
1 package br.edu.infnet.contas.services;
2
3 import br.edu.infnet.contas.domain.ContaCorrente;
4 import java.util.concurrent.CompletableFuture;
5
6 public interface ContaCorrenteCommandService {
7     public CompletableFuture<String> abrirContaCorrente(ContaCorrente cc);
8     public CompletableFuture<String> depositar(String id, double valor);
9     public CompletableFuture<String> sacar(String id, double valor);
10 }
```

18

```
ContaCorrenteCommandServiceImpl.java x
Source History
1 package br.edu.infnet.contas.services;
2
3 import ...9 lines
4
5 @Service
6 public class ContaCorrenteCommandServiceImpl implements ContaCorrenteCommandService {
7
8     @Autowired
9     private final CommandGateway commandGateway;
10
11     public ContaCorrenteCommandServiceImpl(CommandGateway commandGateway) {
12         this.commandGateway = commandGateway;
13     }
14
15     @Override
16     public CompletableFuture<String> abrirContaCorrente(ContaCorrente cc) {
17         return commandGateway.send(new CriarContaCommand(
18             UUID.randomUUID().toString(),
19             cc.getTitular(),
20             cc.getSaldo()));
21     }
22
23     @Override
24     public CompletableFuture<String> depositar(String id, double valor) {
25         return commandGateway.send(new DepositarCommand(id, valor));
26     }
27
28     @Override
29     public CompletableFuture<String> sacar(String id, double valor) {
30         return commandGateway.send(new SacarCommand(id, valor));
31     }
32 }
```

Criação do
serviço de
comandos

Criação da API Rest de comandos

19

```
ContaCorrenteCommandController.java x
Source History
1 package br.edu.infnet.contas.controllers;
2
3 import ...9 lines
12
13 @RestController
14 @RequestMapping(value = "/contas")
15 public class ContaCorrenteCommandController {
16
17     private final ContaCorrenteCommandService service;
18
19     public ContaCorrenteCommandController(ContaCorrenteCommandService service) {
20         this.service = service;
21     }
22
23     @PostMapping
24     public CompletableFuture<String> abrirContaCorrente(@RequestBody ContaCorrente cc) {
25         return service.abrirContaCorrente(cc);
26     }
27
28     @PutMapping(value = "/depositar")
29     public CompletableFuture<String> depositar(@RequestBody Map<String, Object> deposito){
30         return service.depositar((String) deposito.get("id"), (Double) deposito.get("valor"));
31     }
32
33     @PutMapping(value = "/sacar")
34     public CompletableFuture<String> sacar(@RequestBody Map<String, Object> saque){
35         return service.sacar((String) saque.get("id"), (Double) saque.get("valor"));
36     }
37 }
```


Swagger UI

localhost:8080/swagger-ui/index.html#/conta-corrente-controller/abrirContaCorrente

Administrativo | Conteúdo | Ferramentas | Livros | Todos os favoritos

conta-corrente-controller

POST /contas

Parameters Cancel Reset

No parameters

Request body required application/json

```
{
  "id": "12345678",
  "titular": "Machado de Assis",
  "saldo": 3456.78,
  "estado": ""
}
```

Execute Clear



Swagger UI H2 Console

localhost:8080/h2-console/login.do?jsessionid=b8d29f52426e8d5de302c584fb30e7b0

Administrativo Conteúdo Ferramentas Livros Todos os favoritos

Auto commit Auto complete Off Auto select On

jdbc:h2:file:/home/armeniocardos

Run Run Selected Auto complete Clear SQL statement:

ASSOCIATION_VALUE_ENTRY DEAD_LETTER_ENTRY DOMAIN_EVENT_ENTRY SAGA_ENTRY SNAPSHOT_EVENT_ENTRY TOKEN_ENTRY INFORMATION_SCHEMA Sequences Users

H2 2.2.224 (2023-09-17)

SELECT * FROM DOMAIN_EVENT_ENTRY

SELECT * FROM DOMAIN_EVENT_ENTRY;

GLOBAL_INDEX	EVENT_IDENTIFIER	META_DATA
1	d6cb0297-5252-42eb-8a5c-2ed9bc07ec70	3c6d6574612d646174613e3c656e7472793e3c737472696e673e747261636549643c2f737472696e673e3c737472696e673e313666626164333332
2	c664c6dd-2cef-4741-92fe-058cfb51a4e8	3c6d6574612d646174613e3c656e7472793e3c737472696e673e747261636549643c2f737472696e673e3c737472696e673e313666626164333332
3	c6742fc9-3098-46bf-8114-9e7b6a7b1381	3c6d6574612d646174613e3c656e7472793e3c737472696e673e747261636549643c2f737472696e673e3c737472696e673e31393137663665632
4	00c12117-2a75-4092-8d9b-829180d6a2b8	3c6d6574612d646174613e3c656e7472793e3c737472696e673e747261636549643c2f737472696e673e3c737472696e673e31393137663665632

(4 rows, 3 ms)

	PAYLOAD_REVISION	PAYLOAD_TYPE	TIME_STAMP	AGGREGATE_IDENTIFIER	SEQUENCE_NUMBER	TYPE
4654372696164613e	null	br.edu.infnet.contas.eventos.ContaCorrenteCriada	2024-08-21T22:51:27.156Z	676f96f2-be74-4118-b3ce-6b1d040076f4	0	ContaCorrente
	null	br.edu.infnet.contas.eventos.ContaCorrenteAtivada	2024-08-21T22:51:27.160Z	676f96f2-be74-4118-b3ce-6b1d040076f4	1	ContaCorrente
4654372696164613e	null	br.edu.infnet.contas.eventos.ContaCorrenteCriada	2024-08-21T22:51:35.177Z	b99be171-ac15-457d-877c-af07663829c9	0	ContaCorrente
	null	br.edu.infnet.contas.eventos.ContaCorrenteAtivada	2024-08-21T22:51:35.177Z	b99be171-ac15-457d-877c-af07663829c9	1	ContaCorrente



```
ContaCorrenteQueryService.java x ContaCorrenteQueryServiceImpl.java x
Source History
1 package br.edu.infnet.contas.services;
2
3 import java.util.List;
4
5 public interface ContaCorrenteQueryService {
6     public List<Object> listarEventos(String id);
7 }
```

```
ContaCorrenteQueryController.java x ContaCorrenteQueryServiceImpl.java x
Source History
1 package br.edu.infnet.contas.services;
2
3 import java.util.List;
4 import java.util.stream.Collectors;
5 import org.axonframework.event sourcing.eventstore.EventStore;
6 import org.springframework.stereotype.Service;
7
8 @Service
9 public class ContaCorrenteQueryServiceImpl implements ContaCorrenteQueryService {
10
11     private final EventStore eventStore;
12
13     public ContaCorrenteQueryServiceImpl(EventStore eventStore) {
14         this.eventStore = eventStore;
15     }
16
17     @Override
18     public List<Object> listarEventos(String id) {
19         List<Object> retorno = eventStore.readEvents(id, 0)
20             .asStream()
21             .map(record -> record.getPayload()).collect(Collectors.toList());
22         return retorno;
23     }
24 }
```

Criação do
serviço de
consultas

```
ContaCorrenteQueryService.java x ContaCorrenteQueryServiceImpl.java x
Source History
1 package br.edu.infnet.contas.services;
2
3 import [...8 lines]
11
12 @Service
13 public class ContaCorrenteQueryServiceImpl implements ContaCorrenteQueryService {
14
15     private final EventStore eventStore;
16     private final ContaCorrenteRepository contaCorrenteRepository;
17
18     @Autowired
19     public ContaCorrenteQueryServiceImpl(EventStore eventStore, ContaCorrenteRepository contaCorrenteRepository) {
20         this.eventStore = eventStore;
21         this.contaCorrenteRepository = contaCorrenteRepository;
22     }
23
24     @Override
25     public List<Object> listarEventos(String id) {
26         List<Object> retorno = eventStore.readEvents(id, 0)
27             .asStream()
28             .map(record -> record.getPayload()).collect(Collectors.toList());
29         return retorno;
30     }
31
32     public Optional<ContaCorrente> obterPorId(String id) {
33         return contaCorrenteRepository.findById(id);
34     }
35 }
```

Criação do
serviço de
consultas

Criação da API Rest de consultas

```
ContaCorrenteQueryController.java x
Source History
1 package br.edu.infnet.contas.controllers;
2
3 import br.edu.infnet.contas.services.ContaCorrenteQueryService;
4 import java.util.List;
5 import org.springframework.web.bind.annotation.GetMapping;
6 import org.springframework.web.bind.annotation.PathVariable;
7 import org.springframework.web.bind.annotation.RequestMapping;
8 import org.springframework.web.bind.annotation.RestController;
9
10 @RestController
11 @RequestMapping(value = "/contas")
12 public class ContaCorrenteQueryController {
13
14     private final ContaCorrenteQueryService service;
15
16     public ContaCorrenteQueryController(ContaCorrenteQueryService service) {
17         this.service = service;
18     }
19
20     @GetMapping("/{id}/eventos")
21     public List<Object> listarEventos(@PathVariable(value = "id") String id){
22         return service.listarEventos(id);
23     }
24 }
```


Criação da API Rest de consultas

```
ContaCorrenteQueryController.java x
Source History
1 package br.edu.infnet.contas.controllers;
2
3 import ...7 lines
10
11 @RestController
12 @RequestMapping(value = "/contas")
13 public class ContaCorrenteQueryController {
14
15     private final ContaCorrenteQueryService service;
16
17     public ContaCorrenteQueryController(ContaCorrenteQueryService service) {
18         this.service = service;
19     }
20
21     @GetMapping("/{id}/eventos")
22     public List<Object> listarEventos(@PathVariable(value = "id") String id){
23         return service.listarEventos(id);
24     }
25
26     @GetMapping("/{id}")
27     public ContaCorrente obterPorId(@PathVariable(value = "id") String id){
28         return service.obterPorId(id).get();
29     }
30 }
```

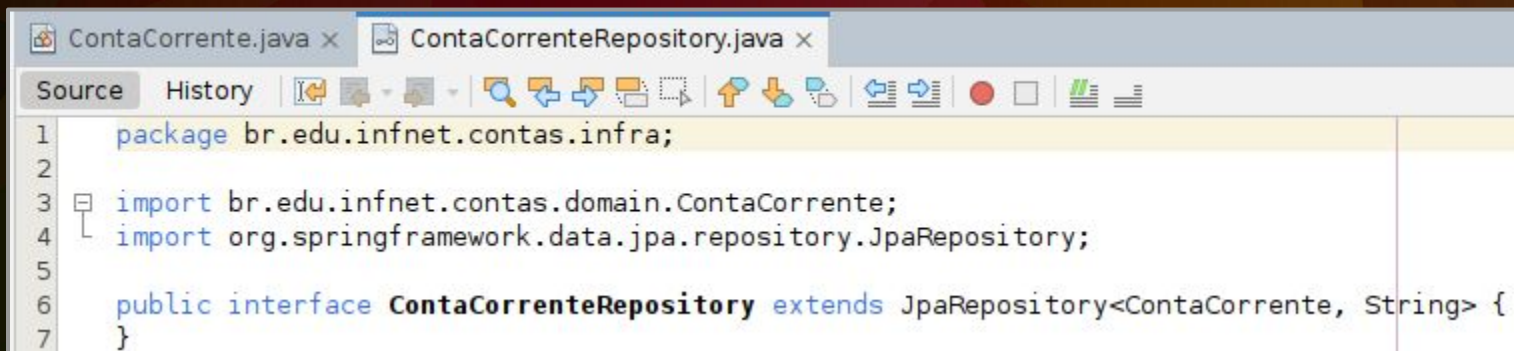


ContaCorrente.java x ContaCorrenteRepository.java x

Source History

```
1 package br.edu.infnet.contas.domain;
2
3 import ...15 lines
18
19 @Aggregate
20 @Entity
21 public class ContaCorrente {
22
23     @AggregateIdentifier
24     @Id
25     private String id;
26     private String titular;
27     private double saldo;
28     private String estado;
29
30     public ContaCorrente() {
31     }
```

Criação do repositório de dados



ContaCorrente.java x ContaCorrenteRepository.java x

Source History

```
1 package br.edu.infnet.contas.infra;
2
3 import br.edu.infnet.contas.domain.ContaCorrente;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface ContaCorrenteRepository extends JpaRepository<ContaCorrente, String> {
7 }
```


Swagger UI

localhost:8080/swagger-ui/index.html#/conta-corrente-query-controller/listarEventos

Administrativo Conteúdo Ferramentas Livros Todos os favoritos

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/contas/77020ec8-63d8-4f5b-9c76-db16216d87d5/eventos' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:8080/contas/77020ec8-63d8-4f5b-9c76-db16216d87d5/eventos
```

Server response

Code	Details
200	<div>Response body</div> <pre>[{ "id": "77020ec8-63d8-4f5b-9c76-db16216d87d5", "titular": "Rubem Fonseca", "saldo": 4567.89 }, { "id": "77020ec8-63d8-4f5b-9c76-db16216d87d5", "estado": "ATIVADA" }, { "id": "77020ec8-63d8-4f5b-9c76-db16216d87d5", "valor": 1000 }]</pre>



Swagger UI

localhost:8080/swagger-ui/index.html#/conta-corrente-command-controller/depositar

Administrativo | Conteúdo | Ferramentas | Livros | Todos os favoritos

PUT /contas/depositar

Parameters Cancel Reset

No parameters

Request body required application/json

```
{
  "id": "77020ec8-63d8-4f5b-9c76-db16216d87d5",
  "valor": 234.56
}
```

Execute Clear

Responses



Swagger UI

localhost:8080/swagger-ui/index.html#/conta-corrente-command-controller/depositar

Administrativo Conteúdo Ferramentas Livros

Todos os favoritos

200

Response body

```
[
  {
    "id": "77020ec8-63d8-4f5b-9c76-db16216d87d5",
    "titular": "Rubem Fonseca",
    "saldo": 4567.89
  },
  {
    "id": "77020ec8-63d8-4f5b-9c76-db16216d87d5",
    "estado": "ATIVADA"
  },
  {
    "id": "77020ec8-63d8-4f5b-9c76-db16216d87d5",
    "valor": 1000
  },
  {
    "id": "77020ec8-63d8-4f5b-9c76-db16216d87d5",
    "valor": 234.56
  }
]
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Fri, 23 Aug 2024 21:55:07 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Responses

Code	Description
200	OK

Media type

/

Controls Accept header.

Example Value Schema



Swagger UI x H2 Console x | +

localhost:8080/swagger-ui/index.html#/conta-corrente-query-controller/obterPorId

Administrativo Conteúdo Ferramentas Livros Todos os favoritos

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8080/contas/5817b242-2a9a-49af-8f50-f82483ce062a' \
-H 'accept: */*'
```

Request URL

```
http://localhost:8080/contas/5817b242-2a9a-49af-8f50-f82483ce062a
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "id": "5817b242-2a9a-49af-8f50-f82483ce062a", "titular": "Machado de Assis", "saldo": 1234.56, "estado": "ATIVADA" }</pre> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Fri, 23 Aug 2024 23:12:08 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>

Responses

