

A wide-angle photograph of a river flowing through a lush green valley. The river curves elegantly through the landscape, reflecting the bright sunlight. The surrounding hills and mountains are densely covered in green forests. The sky above is a vibrant blue, dotted with wispy, white clouds.

# Aggregates com Event Sourcing

Ivan Paulovich  
Software Craftmanship

Banco Olé Consignado – Setembro 2017

# Ivan Paulovich

- Software Craftmanship
- [1ooloop.com](http://1ooloop.com)
- [github.com/ivanpaulovich](https://github.com/ivanpaulovich)



De 2012 à 2014



# Agenda

- O que é Event Sourcing?
- Funcionamento
- Benefícios
- Implementação
- Dúvidas
- Referências

# O que é Event Sourcing?

Garantir que todas as modificações feitas no estado de uma agregação sejam armazenadas como uma sequência de eventos.

# Funcionamento

Eventos de Depósitos e Saques

1	Depositado	C/C: 4030-1 Valor: R\$ 100,00 Data: 17/08/2017
2	Depositado	C/C: 2060-0 Valor: R\$ 200,00 Data: 18/08/2017
3	Sacado	C/C: 4030-1 Valor: R\$ 40,00 Data: 19/08/2017
4	Sacado	C/C: 2060-0 Valor: R\$ 50,00 Data: 23/08/2017



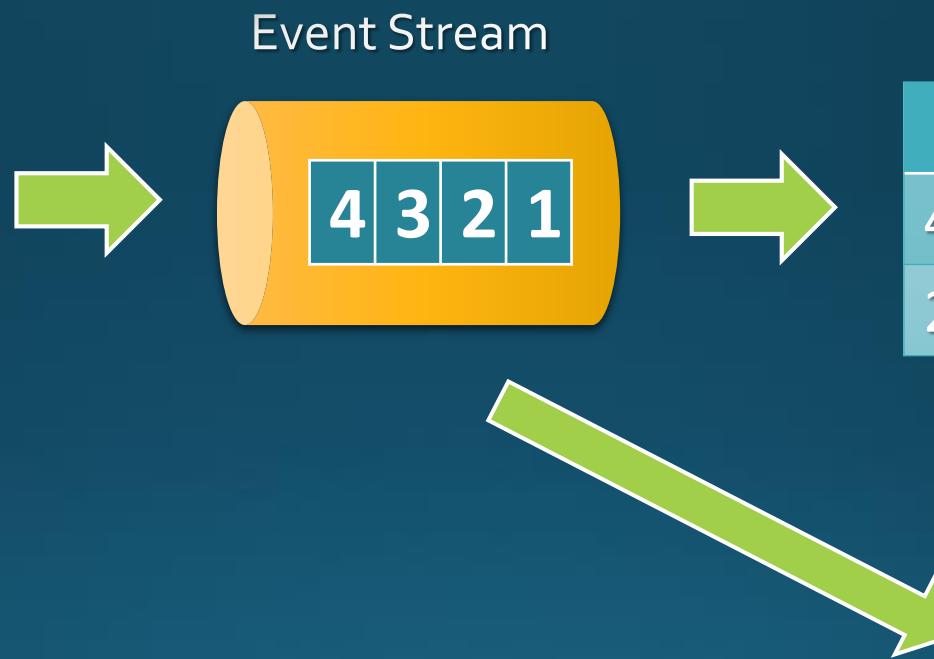
Atual Projeção do Saldo das Contas Correntes

C/C	Saldo
4030-1	R\$ 60,00
2060-0	R\$ 150,00

# Funcionamento

## Eventos de Depósitos e Saques

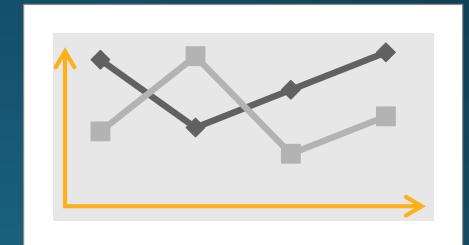
1	Depositado	C/C: 4030-1 Valor: R\$ 100,00 Data: 17/08/2017
2	Depositado	C/C: 2060-0 Valor: R\$ 200,00 Data: 18/08/2017
3	Sacado	C/C: 4030-1 Valor: R\$ 40,00 Data: 19/08/2017
4	Sacado	C/C: 2060-0 Valor: R\$ 50,00 Data: 23/08/2017



## Atual Projeção do Saldo das Contas Correntes

C/C	Saldo
4030-1	R\$ 60,00
2060-0	R\$ 150,00

## Base Analítica em Linha do Tempo



# Benefícios

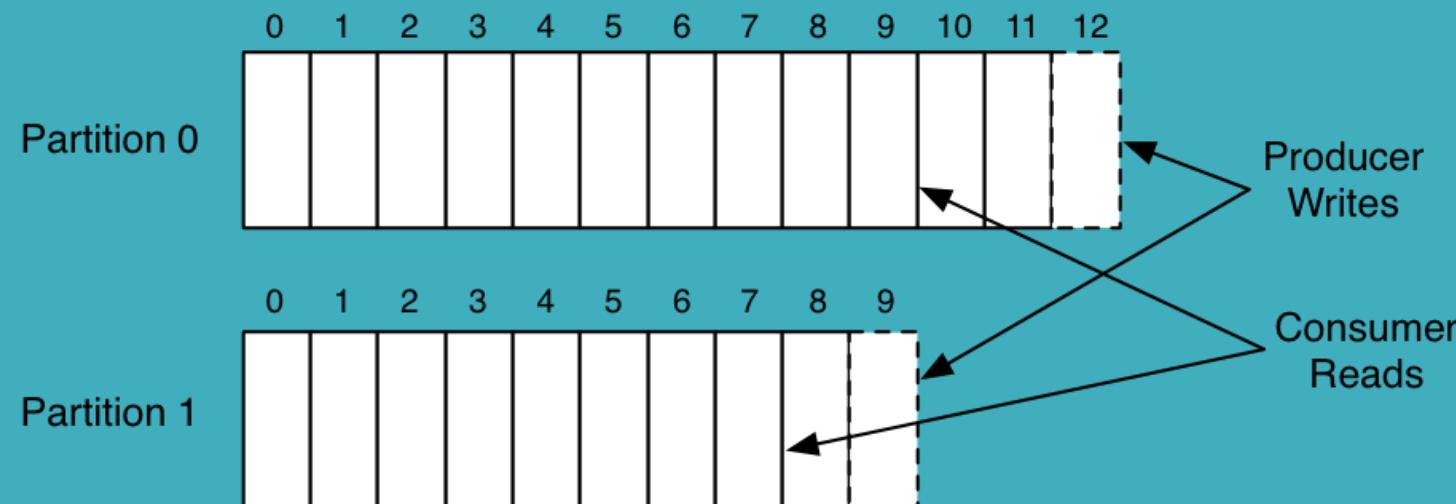
- Auditoria.
- Escalabilidade.
- Redundância.
- Pronto para Big Data!
- Novas funcionalidades de Debug
- Função replay!
- Consultas que voltam no tempo

# Auditoria

- Faz parte da Arquitetura de Event Sourcing registrar:
  - Quem realizou cada evento.
  - Quando o evento ocorreu.
  - E o motivo.
- As modificações são sempre incrementais.
- Sem operação DELETE.

# Escalabilidade

- Event Streams são somente escrita.
- Muito rápido.
- Escala quando particionado.



# Redundância

- Permite múltiplas projeções e replay.
- Geo-distribuído (On premise / Cloud).

Event Stream



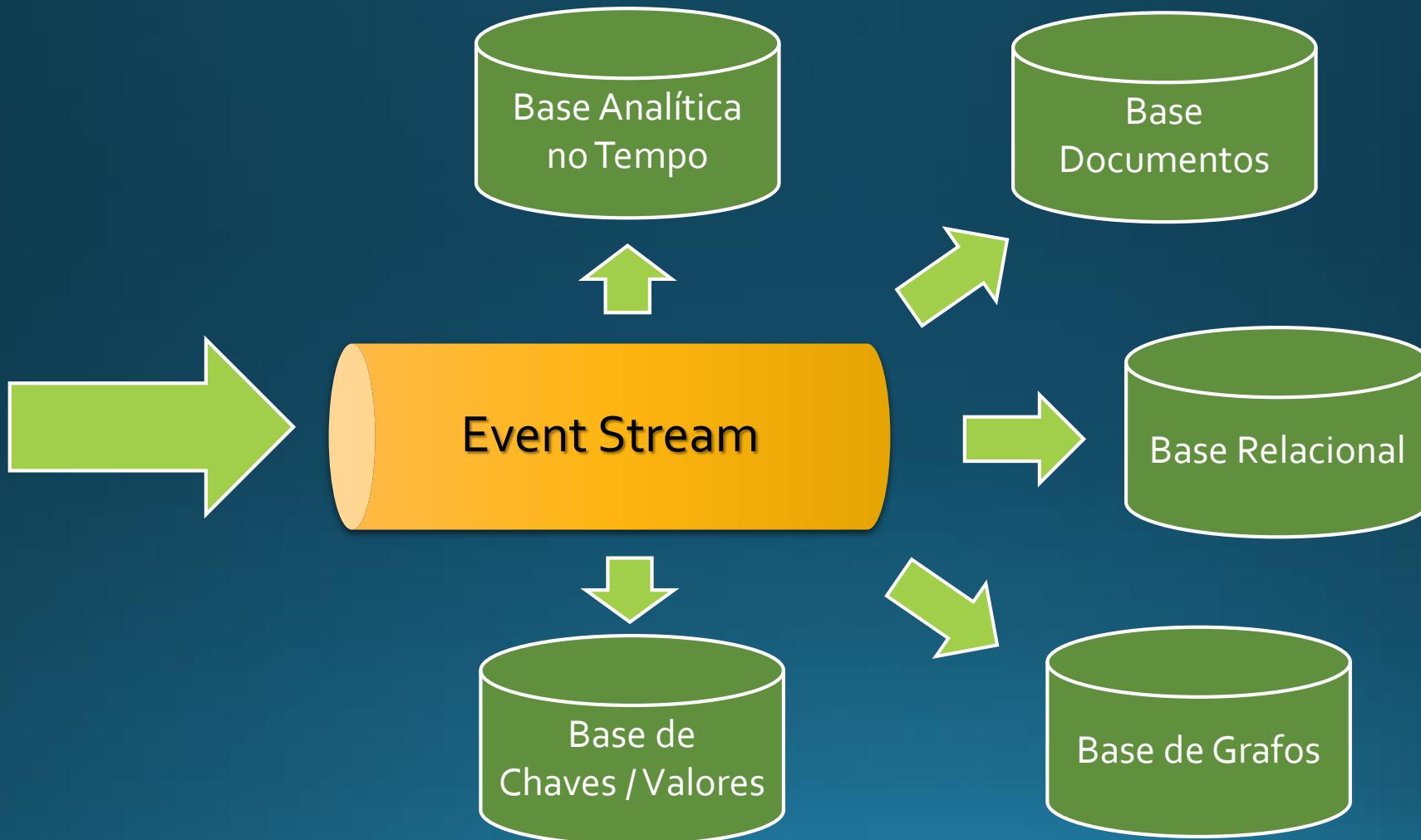
Atual Projeção do Saldo  
das Contas Correntes

C/C	Saldo
4030-1	R\$ 60,00
2060-0	R\$ 150,00

Fonte da Verdade

Uma das possíveis projeções

# Pronto para Big Data!

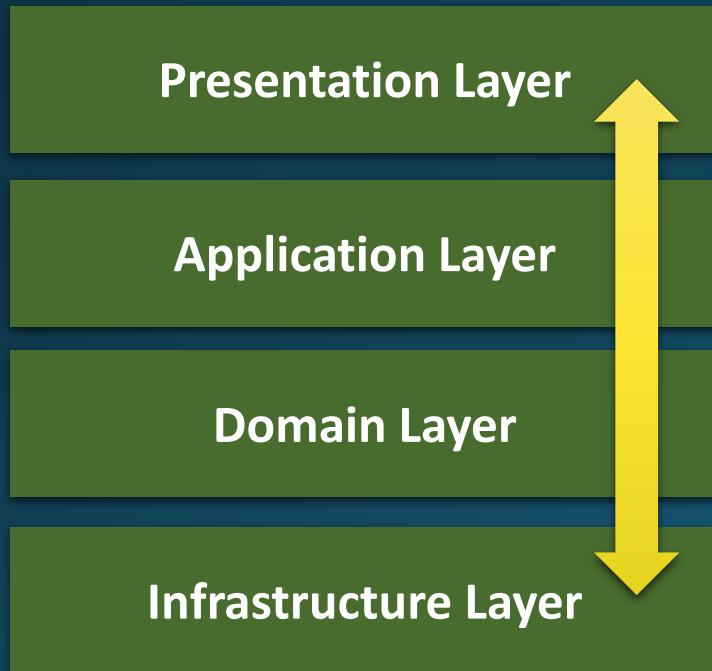


# Implementação

- CQRS
- Domain-Driven-Design
- Aggregates
- Domain Events
- Barramento
- Tratamento de Conflitos

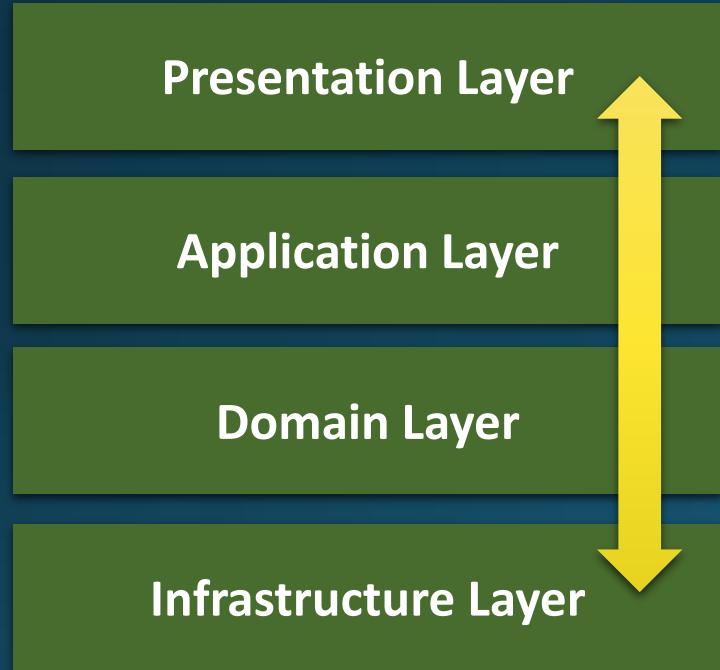
# CQRS

Arquitetura Tradicional em  
Camadas

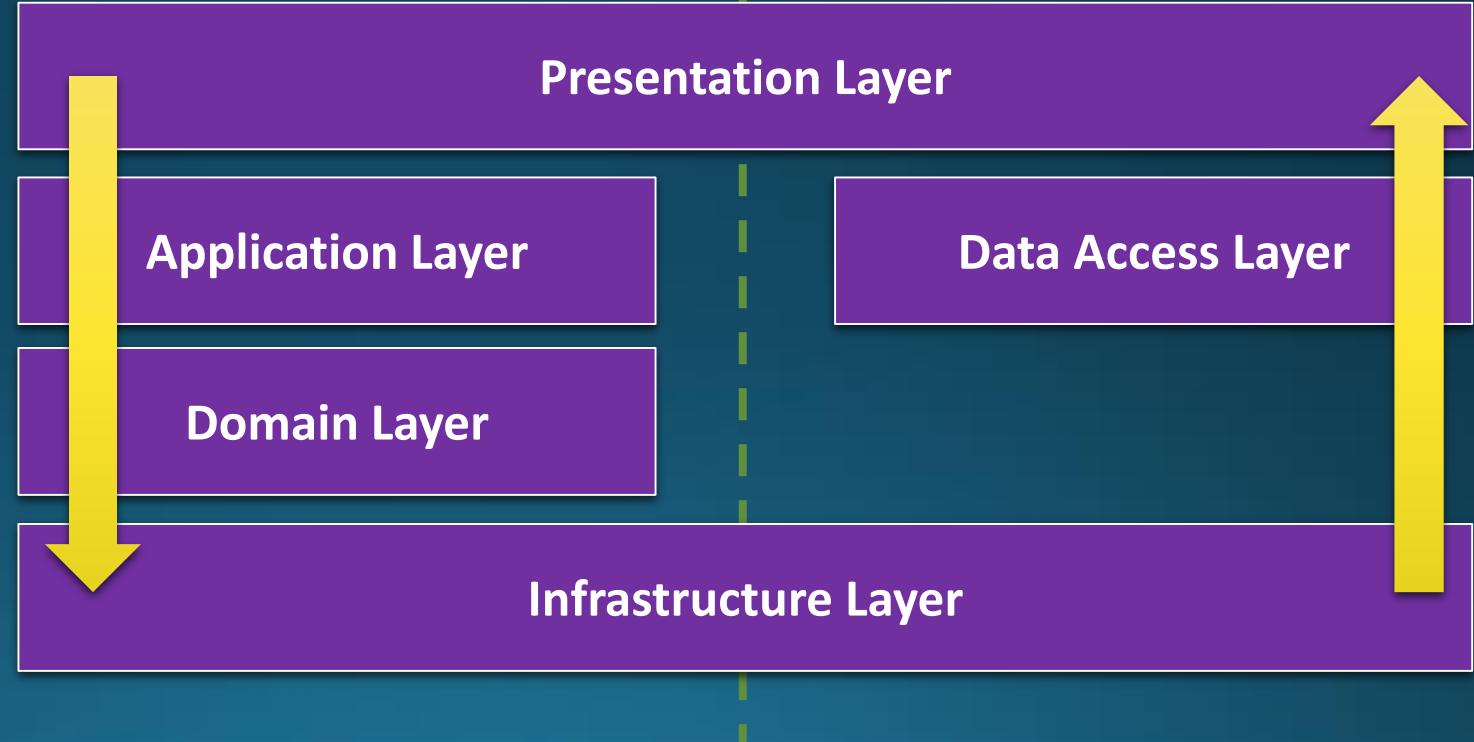


# CQRS

Arquitetura Tradicional em  
Camadas

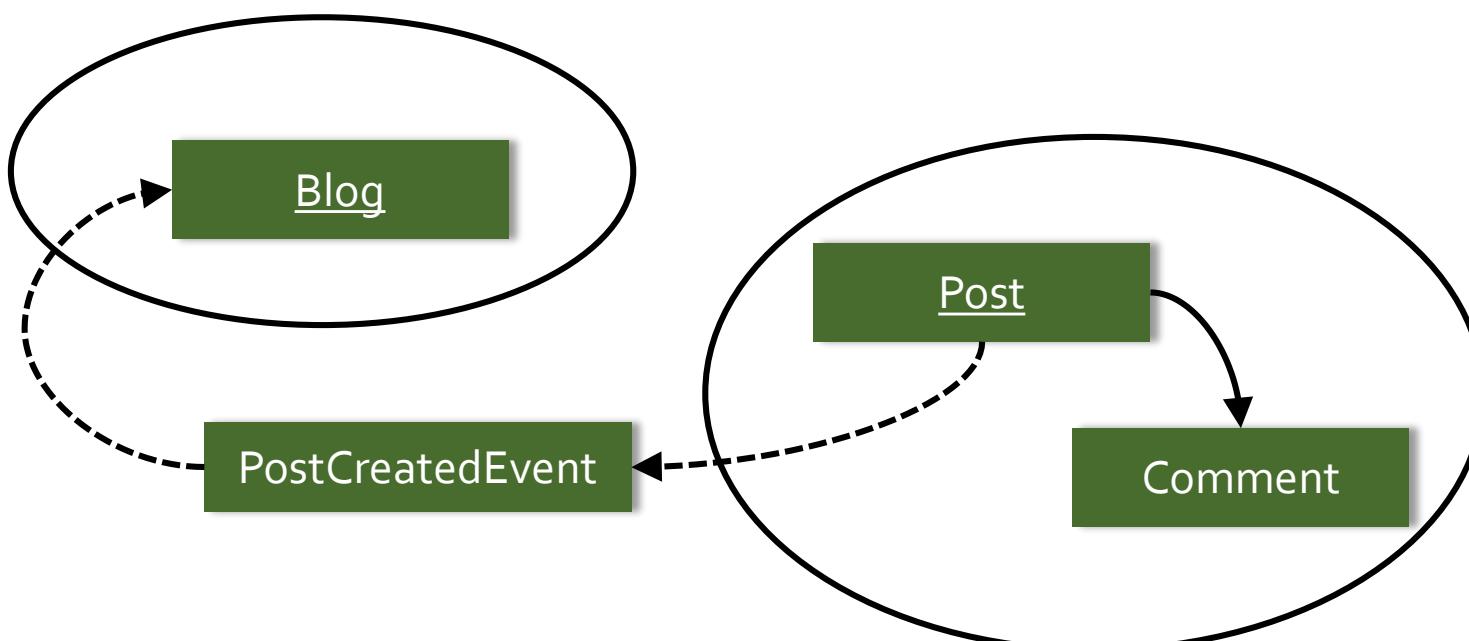


CQRS



# Domain-Driven-Design

- Usa-se a modelagem tática.
- Aggregações lançam eventos de domínio.
- Investe-se na persistência dos Eventos de Domínio



# Aggregates e Domain Events

- Aggregates
  - Protegem o quanto possível o grafo de entidades de acesso externo.
  - Garantem que o estado das entidades filhas são sempre consistentes.
  - Determinam um escopo de transação.

# Aggregates e Domain Events

- Aggregates
  - Protegem o quanto possível o grafo de entidades de acesso externo.
  - Garantem que o estado das entidades filhas são sempre consistentes.
  - Determinam um escopo de transação.
- Domain Events
  - Algo que ocorreu no passado.
  - Uma vez armazenados se tornam imutáveis.
  - Todo o processamento relativo ao evento já foi realizado

# Aggregates e Domain Events

- Aggregates
  - Protegem o quanto possível o grafo de entidades de acesso externo.
  - Garantem que o estado das entidades filhas são sempre consistentes.
  - Determinam um escopo de transação.
- Domain Events
  - Algo que ocorreu no passado.
  - Uma vez armazenados se tornam imutáveis.
  - Todo o processamento relativo ao evento já foi realizado

Event Sourcing != Command Sourcing

# Barramento

- Publish / Subscribe
- Streaming de dados em tempo real
- Armazenamento seguro e replicado



# Event Sourcing

- Não é simples
- É a cereja do bolo dos livros do Evans, Vernon e Greg Young
- Requer grande investimento em modelagem
- Mais fácil começar seguindo um modelo e orientação

- É um caminho longo.



- É um caminho longo.
- Interessante implementar rápido!



ivanpaulovich/jambo: DD Ivan

[GitHub, Inc. \[US\] | https://github.com/ivanpaulovich/jambo/](https://github.com/ivanpaulovich/jambo/)

This repository Search Pull requests Issues Marketplace Explore Unwatch 10 Unstar 25 Fork 10

Code Issues 0 Pull requests 0 Projects 3 Wiki Settings Insights

DDD + Aggregates + Event Sourcing com .NET Core, Kafka e MongoDB (Aplicações Produtora e Consumidora independentes) Edit

ddd ddd-architecture solid solid-principles event-sourcing event-driven dotnet-core kafka aggregate mongodb microservices microservice webapi bearer-authentication cqrs cqrs-es domain-driven-design service-bus csharp optimistic-updates Manage topics

196 commits 2 branches 2 releases 1 contributor Apache-2.0

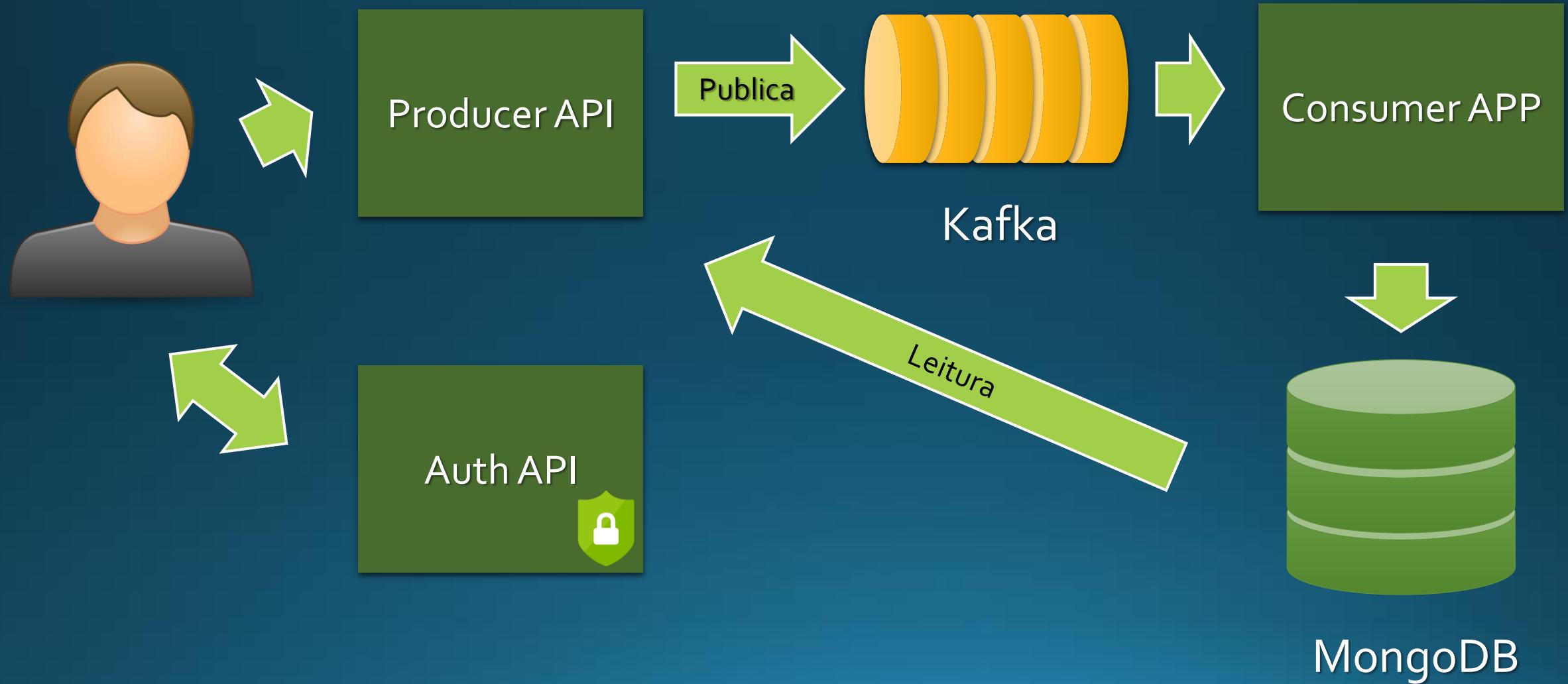
Branch: master New pull request Create new file Upload files Find file Clone or download

ivanpaulovich committed on GitHub ok Latest commit ed3afdd 6 days ago

docs	ok	12 days ago
images	Ok	12 days ago
install	ok	8 days ago
source	ok	8 days ago
.gitattributes	OK	11 months ago
.gitignore	Initial commit	11 months ago
CODE_OF_CONDUCT.md	Código de conduta	12 days ago
CONTRIBUTING.md	guide	12 days ago
LICENSE	Initial commit	11 months ago

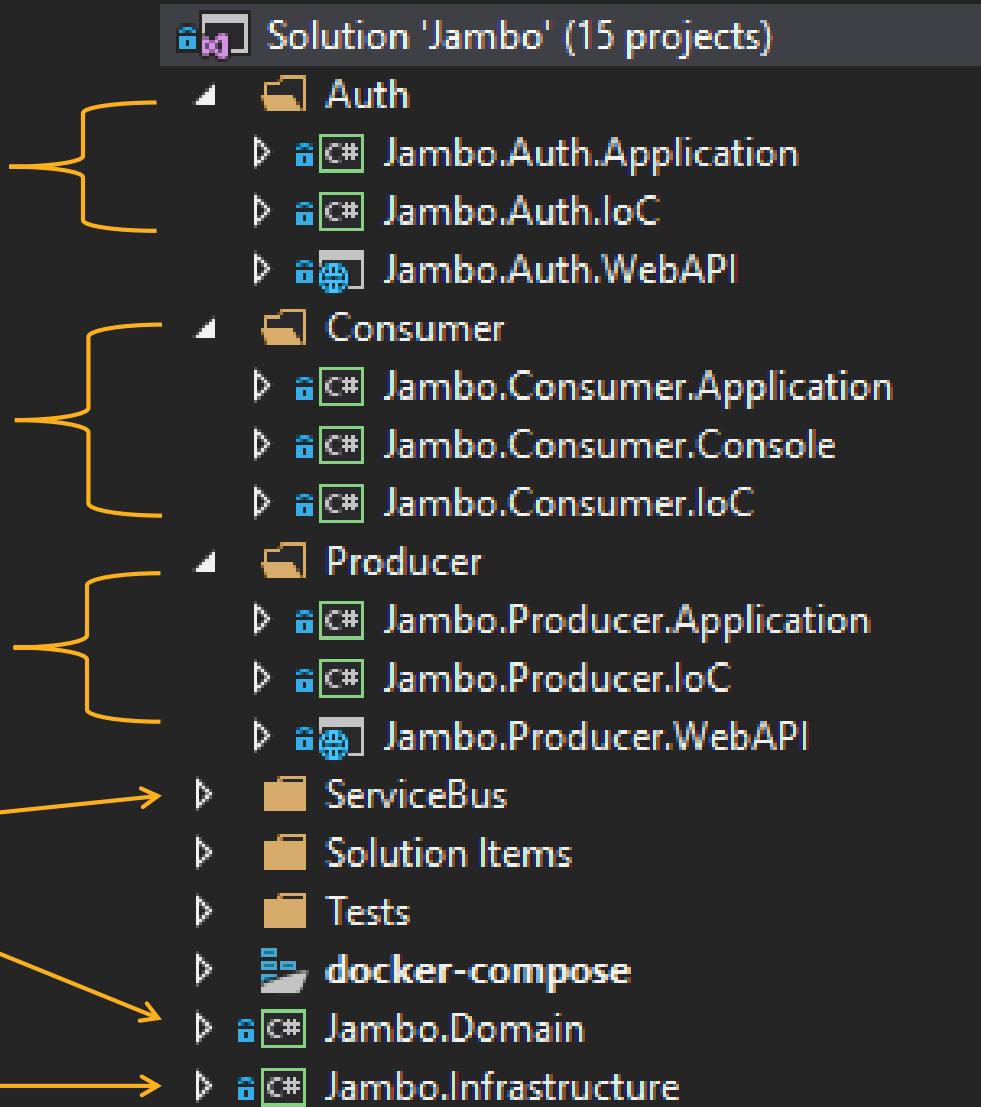
<https://github.com/ivanpaulovich/jambo/>

# Visão Geral



# A Solution

Gera tokens de autenticação



Escuta por mensagens, aplica nas agregações e modifica o MongoDB

Recebe mensagens, processa pelos CommandHandlers e publica DomainEvents no Kafka

Barramento Kafka

Domínio

Mapeamento com MongoDB e repositórios

# Producer (Web API)

Apresentação

- Recebe Commands
- Publica Commands internamente (MediatR)

Application

- Captura Commands (Command Handlers)
- Processa as mensagens e integrações

Domain

- Regras de negócio
- Lança Domain Events

Infraestrutura

- Escreve no Banco de Dados

# Consumer (Console APP)

Apresentação

- Console APP que escuta Domain Events

Application

- Processa Domain Events e controla a transação

Domain

- Aplica Domain Events

Infraestrutura

- Lê do Banco de Dados

# Produzindo Domain Events

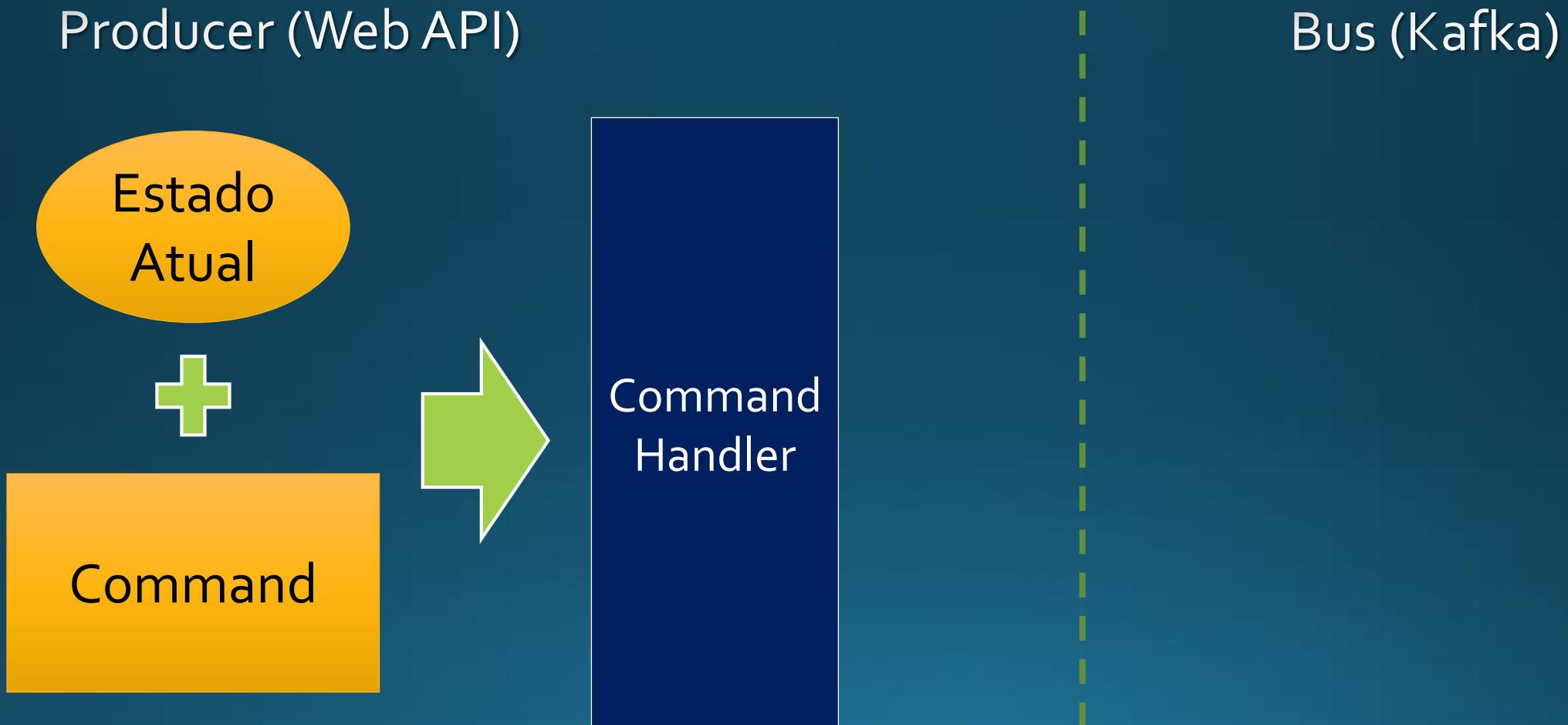
Producer (Web API)

Command  
Handler

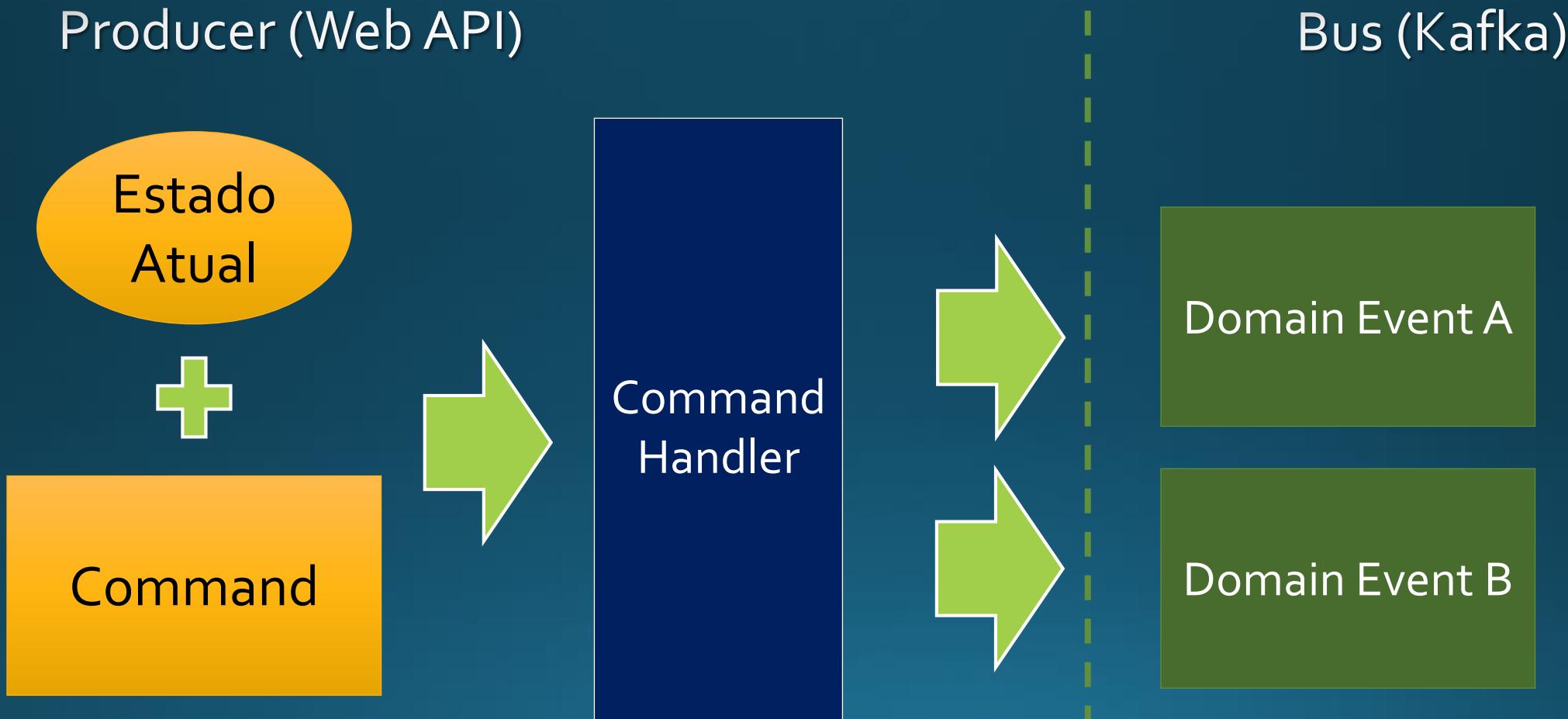
Bus (Kafka)



# Produzindo Domain Events



# Produzindo Domain Events



# Mediator.Send(...)

- Publica mensagens no barramento interno da aplicação
- Diminui o acoplamento

```
[HttpPatch("UpdateContent")]
0 references | Ivan Paulovich, 17 days ago | 1 author, 3 changes | 0 requests | 0 exceptions
public async Task<IActionResult> UpdateContent([FromBody]UpdatePostContentCommand command)
{
    await mediator.Send(command);
    return (IActionResult)Ok();
}
```

# Commands

- Criado pela requisição ao serviço
- Contem Value Types
- Recomendável possuir identificação do usuário
- MediatR

```
UpdatePostContentCommand.cs x
0
7     namespace Jambo.Producer.Application.Commands.Posts
8     {
9         [DataContract]
10        3 references
11         public class UpdatePostContentCommand : CommandBase, IRequest
12         {
13             [DataMember]
14             2 references
15             public Guid Id { get; private set; }
16             [DataMember]
17             2 references
18             public string Title { get; private set; }
19             [DataMember]
20             2 references
21             public string Content { get; private set; }
22
23             1 reference
24             public UpdatePostContentCommand()
25             {
26
27             }
28
29             0 references
30             public UpdatePostContentCommand(Guid id, string title, string content) : this()
31             {
32                 Id = id;
33                 Title = title;
34                 Content = content;
35             }
36         }
37     }
```

# Command Handlers

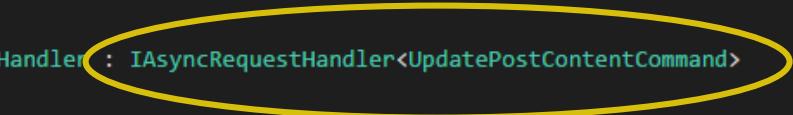
- Específicos para cada Command
- Assinado no MediatR
- Faz parte do negócio

```
UpdateContentCommandHandler.cs ✘
10
11  namespace Jambo.Production.Application.CommandHandlers.Posts
12  {
13      0 references
14      public class UpdateContentCommandHandler : IAsyncRequestHandler<UpdatePostContentCommand>
15      {
16          2 references
17          private readonly IPublisher bus;
18          2 references
19          private readonly IPostReadOnlyRepository postReadOnlyRepository;
20
21          0 references
22          public UpdateContentCommandHandler(
23              IPublisher bus,
24              IPostReadOnlyRepository postReadOnlyRepository)
25          {
26              this.bus = bus ??
27                  throw new ArgumentNullException(nameof(bus));
28              this.postReadOnlyRepository = postReadOnlyRepository ??
29                  throw new ArgumentNullException(nameof(postReadOnlyRepository));
30
31          0 references
32          public async Task Handle(UpdatePostContentCommand message)
33          {
34              Post post = await postReadOnlyRepository.GetPost(message.Id);
35              post.UpdateContent>Title.Create(message.Title), Content.Create(message.Content));
36
37          }
38      }
39  }
```

# Command Handlers

- Específicos para cada Command
- Assinado no MediatR
- Faz parte do negócio

```
UpdateContentCommandHandler.cs ✘
10
11  namespace Jambo.Production.Application.CommandHandlers.Posts
12  {
13      0 references
14      public class UpdateContentCommandHandler : IAsyncRequestHandler<UpdatePostContentCommand>
15      {
16          2 references
17          private readonly IPublisher bus;
18          2 references
19          private readonly IPostReadOnlyRepository postReadOnlyRepository;
20
21          0 references
22          public UpdateContentCommandHandler(
23              IPublisher bus,
24              IPostReadOnlyRepository postReadOnlyRepository)
25          {
26              this.bus = bus ??
27                  throw new ArgumentNullException(nameof(bus));
28              this.postReadOnlyRepository = postReadOnlyRepository ??
29                  throw new ArgumentNullException(nameof(postReadOnlyRepository));
30
31          0 references
32          public async Task Handle(UpdatePostContentCommand message)
33          {
34              Post post = await postReadOnlyRepository.GetPost(message.Id);
35              post.UpdateContent>Title.Create(message.Title), Content.Create(message.Content));
36
37          }
38      }
39  }
```



# Command Handlers

- Específicos para cada Command
- Assinado no MediatR
- Faz parte do negócio

Busca o estado atual

Realiza uma operação na Agregação

Publica os Domain Events

```
UpdateContentCommandHandler.cs ✘
10
11  namespace Jambo.Production.Application.CommandHandlers.Posts
12  {
13      0 references
14      public class UpdateContentCommandHandler : IAsyncRequestHandler<UpdatePostContentCommand>
15      {
16          2 references
17          private readonly IPublisher bus;
18          2 references
19          private readonly IPostReadOnlyRepository postReadOnlyRepository;
20
21          0 references
22          public UpdateContentCommandHandler(
23              IPublisher bus,
24              IPostReadOnlyRepository postReadOnlyRepository)
25          {
26              this.bus = bus ??
27                  throw new ArgumentNullException(nameof(bus));
28              this.postReadOnlyRepository = postReadOnlyRepository ??
29                  throw new ArgumentNullException(nameof(postReadOnlyRepository));
30
31          0 references
32          public async Task Handle(UpdatePostContentCommand message)
33          {
34              Post post = await postReadOnlyRepository.GetPost(message.Id);
35              post.UpdateContent>Title.Create(message.Title), Content.Create(message.Content));
36              await bus.Publish(post.GetEvents(), message.Header);
37          }
38      }
39  }
```

# Events

- Possui Value Types
- Serializável (JSON)
- MediatR (de novo!)

```
PostContentUpdatedDomainEvent.cs •  
1  using System;  
2  
3  namespace Jambo.Domain.Model.Posts.Events  
4  {  
5      8 references  
6      public class PostContentUpdatedDomainEvent : DomainEvent  
7      {  
8          2 references  
9              public Title Title { get; private set; }  
10             2 references  
11             public Content Content { get; private set; }  
12  
13             1 reference  
14             public PostContentUpdatedDomainEvent(Guid aggregateRootId, int version,  
15                 DateTime createdDate, Header header, Title title, Content content)  
16                 : base(aggregateRootId, version, createdDate, header)  
17             {  
18                 Title = title;  
19                 Content = content;  
20             }  
21  
22             1 reference  
23             public static PostContentUpdatedDomainEvent Create(AggregateRoot aggregateRoot,  
24                 Title title, Content content)  
25             {  
26                 if (aggregateRoot == null)  
27                     throw new ArgumentNullException("aggregateRoot");  
28  
29                     PostContentUpdatedDomainEvent domainEvent = new PostContentUpdatedDomainEvent(  
30                         aggregateRoot.Id, aggregateRoot.Version, DateTime.UtcNow, null, title, content);  
31  
32                     return domainEvent;  
33             }  
34         }  
35     }  
36 }
```

# O Aggregate

- Regras de negócio
- Lança Eventos
- Aplica Eventos

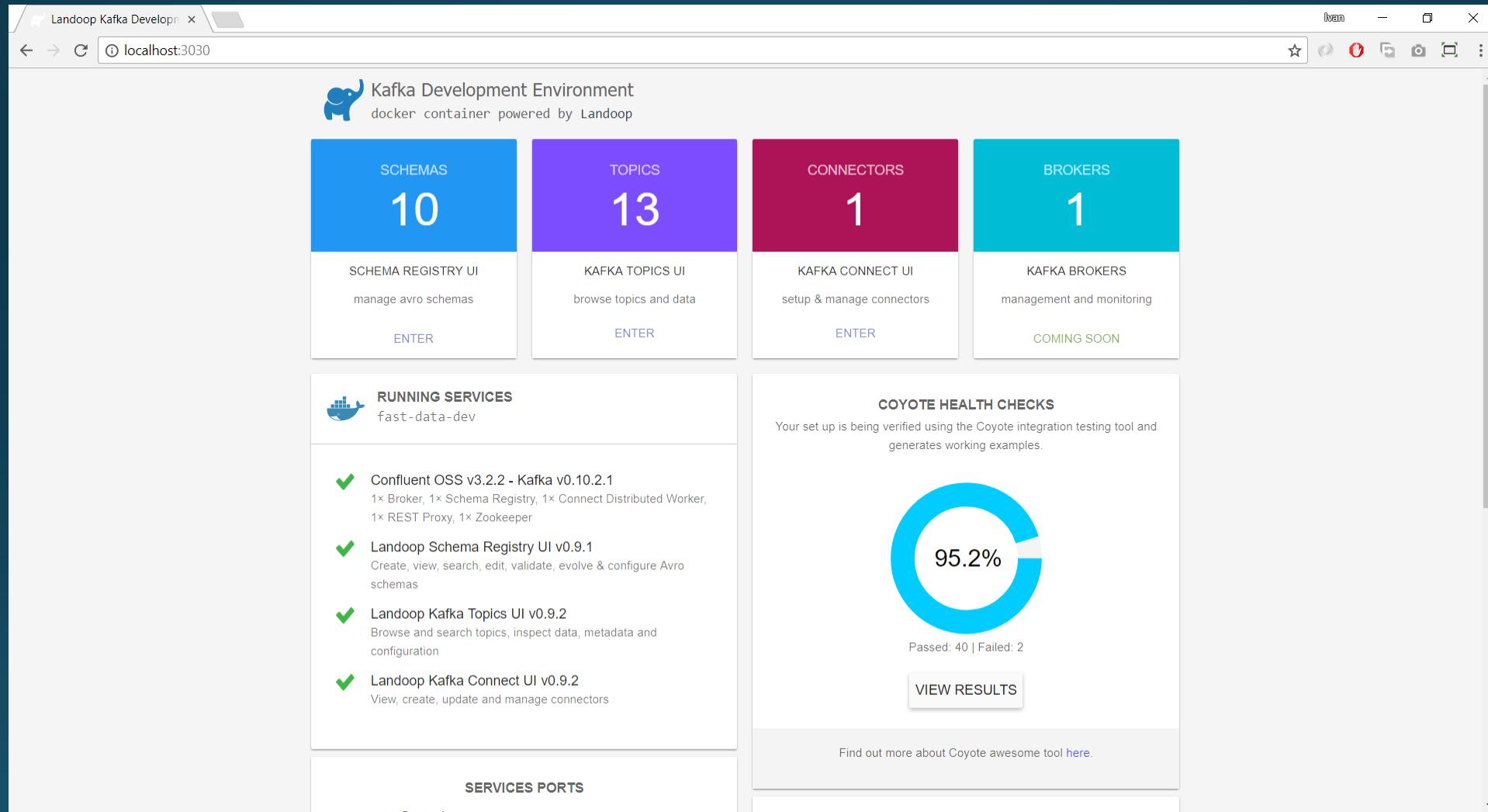
30 references | Ivan Paulovich, 15 days ago | 2 authors, 8 changes

```
public class Post : AggregateRoot
{
    2 references | Ivan Paulovich, 15 days ago | 1 author, 1 change | 0 exceptions
    public void UpdateContent>Title title, Content content)
    {
        if (enabled == false)
        {
            throw new BlogDomainException(
                "The blog is disabled. Enable this before making any changes.");
        }

        Raise(PostContentUpdatedDomainEvent.Create(this, title, content));
    }
}

1 reference | Ivan Paulovich, 16 days ago | 1 author, 3 changes | 0 exceptions
protected void When(PostContentUpdatedDomainEvent @event)
{
    title = @event.Title;
    content = @event.Content;
}
```

# Como eu acesso o Kafka?



The screenshot shows the Landoop Kafka Development Environment UI running in a browser window titled "Landoop Kafka Develop". The URL is "localhost:3030". The interface includes a header with the Landoop logo and the text "Kafka Development Environment" and "docker container powered by Landoop". Below the header are four cards: "SCHEMAS" (10, Schema Registry UI, ENTER), "TOPICS" (13, Kafka Topics UI, ENTER), "CONNECTORS" (1, Kafka Connect UI, ENTER), and "BROKERS" (1, Kafka Brokers, COMING SOON). A "RUNNING SERVICES" section lists Confluent OSS v3.2.2 - Kafka v0.10.2.1, Landoop Schema Registry UI v0.9.1, Landoop Kafka Topics UI v0.9.2, and Landoop Kafka Connect UI v0.9.2, all marked as green checkmarks. A "COYOTE HEALTH CHECKS" section shows a 95.2% success rate with 40 passed and 2 failed tests, with a "VIEW RESULTS" button. At the bottom, there's a "SERVICES PORTS" section and a footer with copyright information.

Landoop Kafka Develop

localhost:3030

Kafka Development Environment  
docker container powered by Landoop

SCHEMAS  
**10**  
SCHEMA REGISTRY UI  
manage avro schemas  
ENTER

TOPICS  
**13**  
KAFKA TOPICS UI  
browse topics and data  
ENTER

CONNECTORS  
**1**  
KAFKA CONNECT UI  
setup & manage connectors  
ENTER

BROKERS  
**1**  
KAFKA BROKERS  
management and monitoring  
COMING SOON

RUNNING SERVICES  
fast-data-dev

- ✓ Confluent OSS v3.2.2 - Kafka v0.10.2.1  
1x Broker, 1x Schema Registry, 1x Connect Distributed Worker, 1x REST Proxy, 1x Zookeeper
- ✓ Landoop Schema Registry UI v0.9.1  
Create, view, search, edit, validate, evolve & configure Avro schemas
- ✓ Landoop Kafka Topics UI v0.9.2  
Browse and search topics, inspect data, metadata and configuration
- ✓ Landoop Kafka Connect UI v0.9.2  
View, create, update and manage connectors

COYOTE HEALTH CHECKS  
Your set up is being verified using the Coyote integration testing tool and generates working examples.

95.2%

Passed: 40 | Failed: 2

VIEW RESULTS

SERVICES PORTS

2022-06-21 11:51:31 PM

# Como eu acesso o Kafka?

The screenshot shows the Kafka Topics UI - Browse interface at [localhost:3030/kafka-topics-ui/#/cluster/fast-data-dev/topic/n/jambov32/](http://localhost:3030/kafka-topics-ui/#/cluster/fast-data-dev/topic/n/jambov32/). The left sidebar lists 8 topics: coyote\_test\_avro, coyote\_test\_binary, coyote\_test\_json, jambov32, nyc\_yellow\_taxi\_trip\_data, logs-broker, sea\_vessel\_position\_reports, and reddit\_posts. The right panel is focused on the jambov32 topic, which has 1 partition. The DATA tab is selected, showing 4 total messages fetched. The first message is displayed with its key and value. The key is a Java class definition, and the value is a JSON object representing a domain event. The partition offset is 0.

KAFKA TOPICS

8 TOPICS

Search topics:

- coyote\_test\_avro  
1 Partitions x 1 Replication
- coyote\_test\_binary  
1 Partitions x 1 Replication
- coyote\_test\_json  
1 Partitions x 1 Replication
- jambov32**  
1 Partitions x 1 Replication
- nyc\_yellow\_taxi\_trip\_data  
1 Partitions x 1 Replication | 2 configs
- logs-broker  
1 Partitions x 1 Replication
- sea\_vessel\_position\_reports  
3 Partitions x 1 Replication | 2 configs
- reddit\_posts  
5 Partitions x 1 Replication | 2 configs

Powered by [Landoop](#)

jambov32

DATA PARTITIONS 1 CONFIGURATION

Total Messages Fetched: 4. Data type: binary

filter

Partition 0 Seek to offset 0

TOPIC	TABLE	RAW DATA
		<p><b>Key:</b> Jambo.Domain.Model.Blogs.Events.BlogCreatedDomainEvent, Jambo.Domain, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null <b>Value:</b> { "AggregateRootId": "5bed6c5b-2f71-47d8-a27a-2084c3f21c3c", "Version": 0, "CreatedDate": "2017-09-20T01:02:19.5813594Z", "Header": { "CorrelationId": "6b7c1bd2-3e5e-46d5-8ab2-1529f8979dcb", "UserName": "string" } }</p> <p><b>Key:</b> Jambo.Domain.Model.Blogs.Events.BlogUrlUpdatedDomainEvent, Jambo.Domain, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null <b>Value:</b> { "Url": "http://www.100loop.com", "AggregateRootId": "5bed6c5b-2f71-47d8-a27a-2084c3f21c3c", "Version": 1, "CreatedDate": "2017-09-01T01:02:19.5838767Z", "Header": { "CorrelationId": "6b7c1bd2-3e5e-46d5-8ab2-1529f8979dcb", "UserName": "string" } }</p> <p><b>Key:</b> Jambo.Domain.Model.Posts.Events.PostCreatedDomainEvent, Jambo.Domain, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null <b>Value:</b> { "BlogId": "5bed6c5b-2f71-47d8-a27a-2084c3f21c3c", "BlogVersion": 2, "AggregateRootId": "6f1b10d4-e90f-469b-ad63-183a471e83cf", "Version": 0, "CreatedDate": "2017-09-20T01:03:59.4128395Z", "Header": { "CorrelationId": "2a98da16-ac06-4816-8241-c7e2f68ea9c7", "UserName": "string" } }</p> <p><b>Key:</b> Jambo.Domain.Model.Posts.Events.PostContentUpdatedDomainEvent, Jambo.Domain, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null <b>Value:</b> { "Title": { "Text": "Oi Mundo" }, "Content": { "Text": "Vamos ver no que d\u00e1!!" }, "AggregateRootId": "6f1b10d4-e90f-469b-ad63-183a471e83cf", "Version": 1, "CreatedDate": "2017-09-20T01:03:59.4148296Z", "Header": { "CorrelationId": "2a98da16-ac06-4816-8241-c7e2f68ea9c7", "UserName": "string" } }</p>

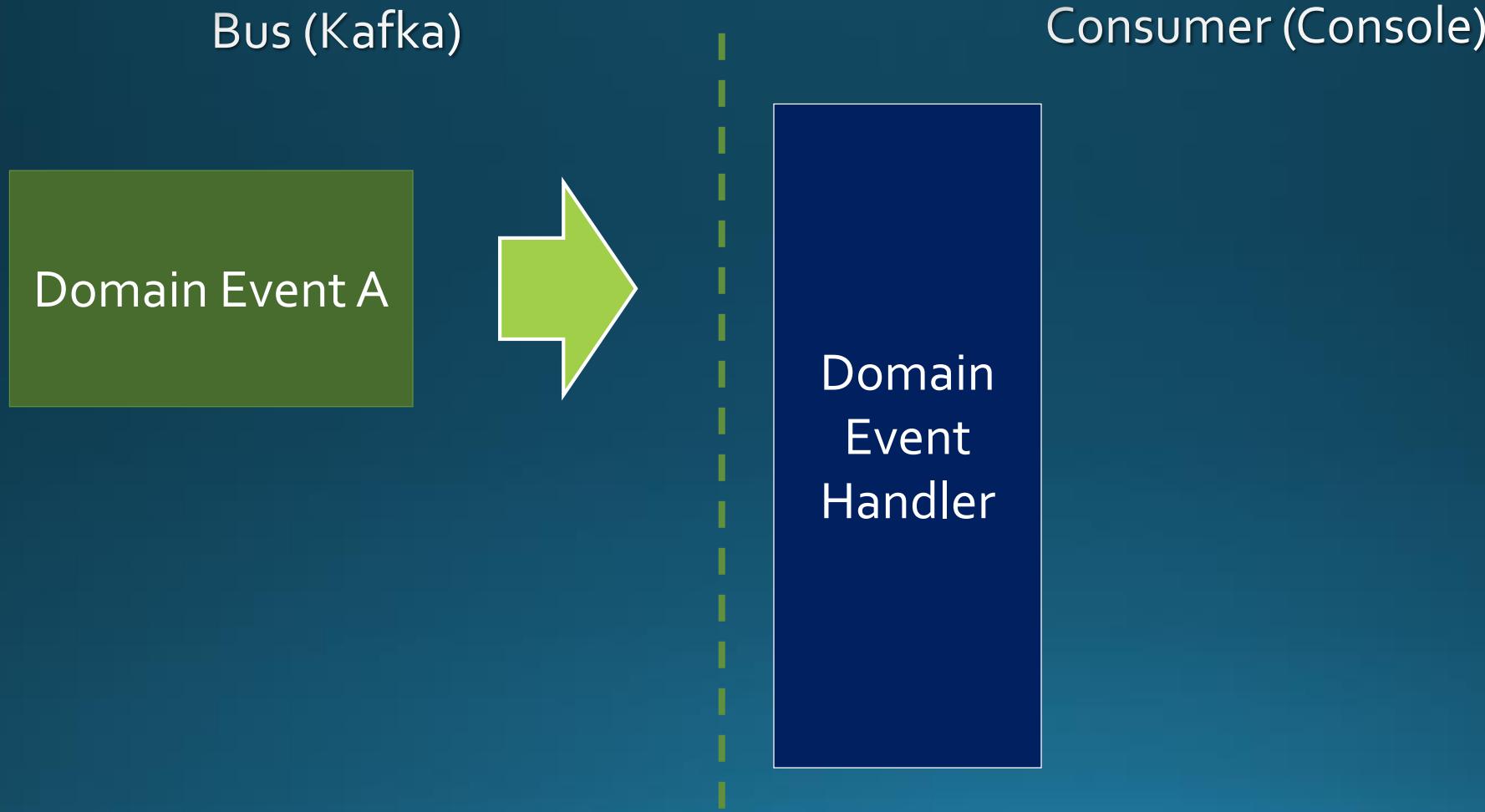
# Criando Projeções

Bus (Kafka)

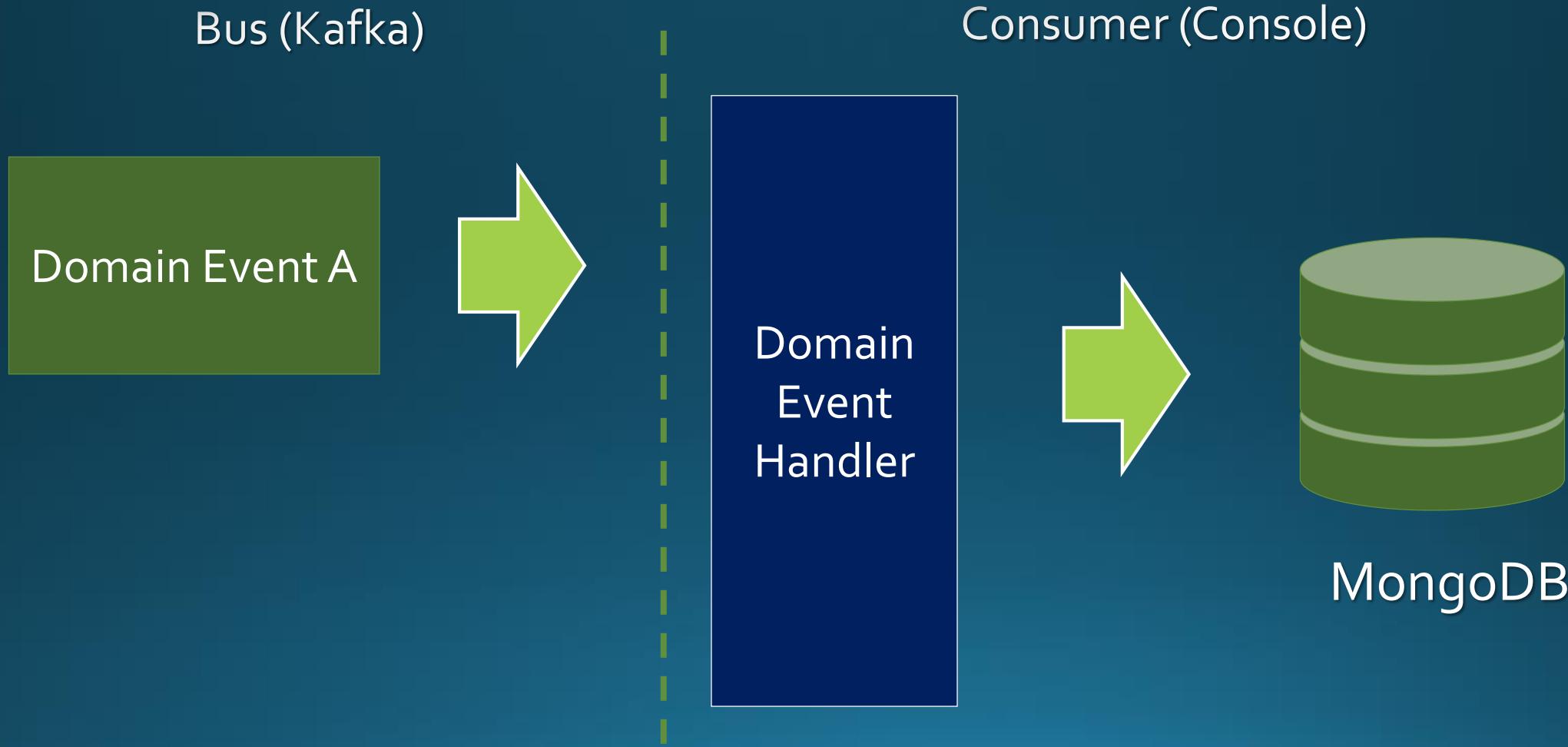
Consumer (Console)



# Criando Projeções



# Criando Projeções



# Domain Event Handlers

- Específicos para cada Evento
- Assinado no MediatR
- Interage com o BD
- Controle transação

```
1 reference | Ivan Paulovich, 18 days ago | 1 author, 3 changes
public class PostContentUpdatedEventHandler : IRequestHandler<PostContentUpdatedDomainEvent>
{
    private readonly IPostReadOnlyRepository postReadOnlyRepository;
    private readonly IPostWriteOnlyRepository postWriteOnlyRepository;

    0 references | Ivan Paulovich, 18 days ago | 1 author, 2 changes | 0 exceptions
    public PostContentUpdatedEventHandler(
        IPostReadOnlyRepository postReadOnlyRepository,
        IPostWriteOnlyRepository postWriteOnlyRepository)
    {
        this.postReadOnlyRepository = postReadOnlyRepository ??
            throw new ArgumentNullException(nameof(postReadOnlyRepository));
        this.postWriteOnlyRepository = postWriteOnlyRepository ??
            throw new ArgumentNullException(nameof(postWriteOnlyRepository));
    }

    10 references | Ivan Paulovich, 18 days ago | 1 author, 3 changes | 0 exceptions
    public void Handle(PostContentUpdatedDomainEvent domainEvent)
    {
        Post post = postReadOnlyRepository.GetPost(domainEvent.AggregateRootId).Result;

        if (post.Version != domainEvent.Version)
            throw new TransactionConflictException(post, domainEvent);

        post.Apply(domainEvent);
        postWriteOnlyRepository.UpdatePost(post).Wait();
    }
}
```

# Domain Event Handlers

- Específicos para cada Evento
- Assinado no MediatR
- Interage com o BD
- Controle transação

```
1 reference | Ivan Paulovich, 18 days ago | 1 author, 3 changes
public class PostContentUpdatedEventHandler : IRequestHandler<PostContentUpdatedDomainEvent>
{
    private readonly IPostReadOnlyRepository postReadOnlyRepository;
    private readonly IPostWriteOnlyRepository postWriteOnlyRepository;

    0 references | Ivan Paulovich, 18 days ago | 1 author, 2 changes | 0 exceptions
    public PostContentUpdatedEventHandler(
        IPostReadOnlyRepository postReadOnlyRepository,
        IPostWriteOnlyRepository postWriteOnlyRepository)
    {
        this.postReadOnlyRepository = postReadOnlyRepository ??
            throw new ArgumentNullException(nameof(postReadOnlyRepository));
        this.postWriteOnlyRepository = postWriteOnlyRepository ??
            throw new ArgumentNullException(nameof(postWriteOnlyRepository));
    }

    10 references | Ivan Paulovich, 18 days ago | 1 author, 3 changes | 0 exceptions
    public void Handle(PostContentUpdatedDomainEvent domainEvent)
    {
        Post post = postReadOnlyRepository.GetPost(domainEvent.AggregateRootId).Result;

        if (post.Version != domainEvent.Version)
            throw new TransactionConflictException(post, domainEvent);

        post.Apply(domainEvent);
        postWriteOnlyRepository.UpdatePost(post).Wait();
    }
}
```

# Domain Event Handlers

- Específicos para cada Evento
- Assinado no MediatR
- Interage com o BD
- Controle transação

Busca o estado atual

Optimistic Concurrency

Aplica uma operação na Agregação

Atualiza em BD

```
1 reference | Ivan Paulovich, 18 days ago | 1 author, 3 changes
public class PostContentUpdatedEventHandler : IRequestHandler<PostContentUpdatedDomainEvent>
{
    private readonly IPostReadOnlyRepository postReadOnlyRepository;
    private readonly IPostWriteOnlyRepository postWriteOnlyRepository;

    0 references | Ivan Paulovich, 18 days ago | 1 author, 2 changes | 0 exceptions
    public PostContentUpdatedEventHandler(
        IPostReadOnlyRepository postReadOnlyRepository,
        IPostWriteOnlyRepository postWriteOnlyRepository)
    {
        this.postReadOnlyRepository = postReadOnlyRepository ??
            throw new ArgumentNullException(nameof(postReadOnlyRepository));
        this.postWriteOnlyRepository = postWriteOnlyRepository ??
            throw new ArgumentNullException(nameof(postWriteOnlyRepository));
    }

    10 references | Ivan Paulovich, 18 days ago | 1 author, 3 changes | 0 exceptions
    public void Handle(PostContentUpdatedDomainEvent domainEvent)
    {
        Post post = postReadOnlyRepository.GetPost(domainEvent.AggregateRootId).Result;
        if (post.Version != domainEvent.Version)
            throw new TransactionConflictException(post, domainEvent);
        post.Apply(domainEvent);
        postWriteOnlyRepository.UpdatePost(post).Wait();
    }
}
```

# Como eu acesso o MongoDB?

The screenshot shows a web browser window with the URL <https://robomongo.org>. The page features two main sections: "Robo 3T" on the left and "Studio 3T" on the right. Both sections include a green button labeled "Download" and a large green icon at the bottom.

**Robo 3T**

Robo 3T (formerly Robomongo) is the free lightweight GUI for MongoDB enthusiasts.

- MongoDB GUI with embedded shell

[Download Robo 3T](#)

**Studio 3T**

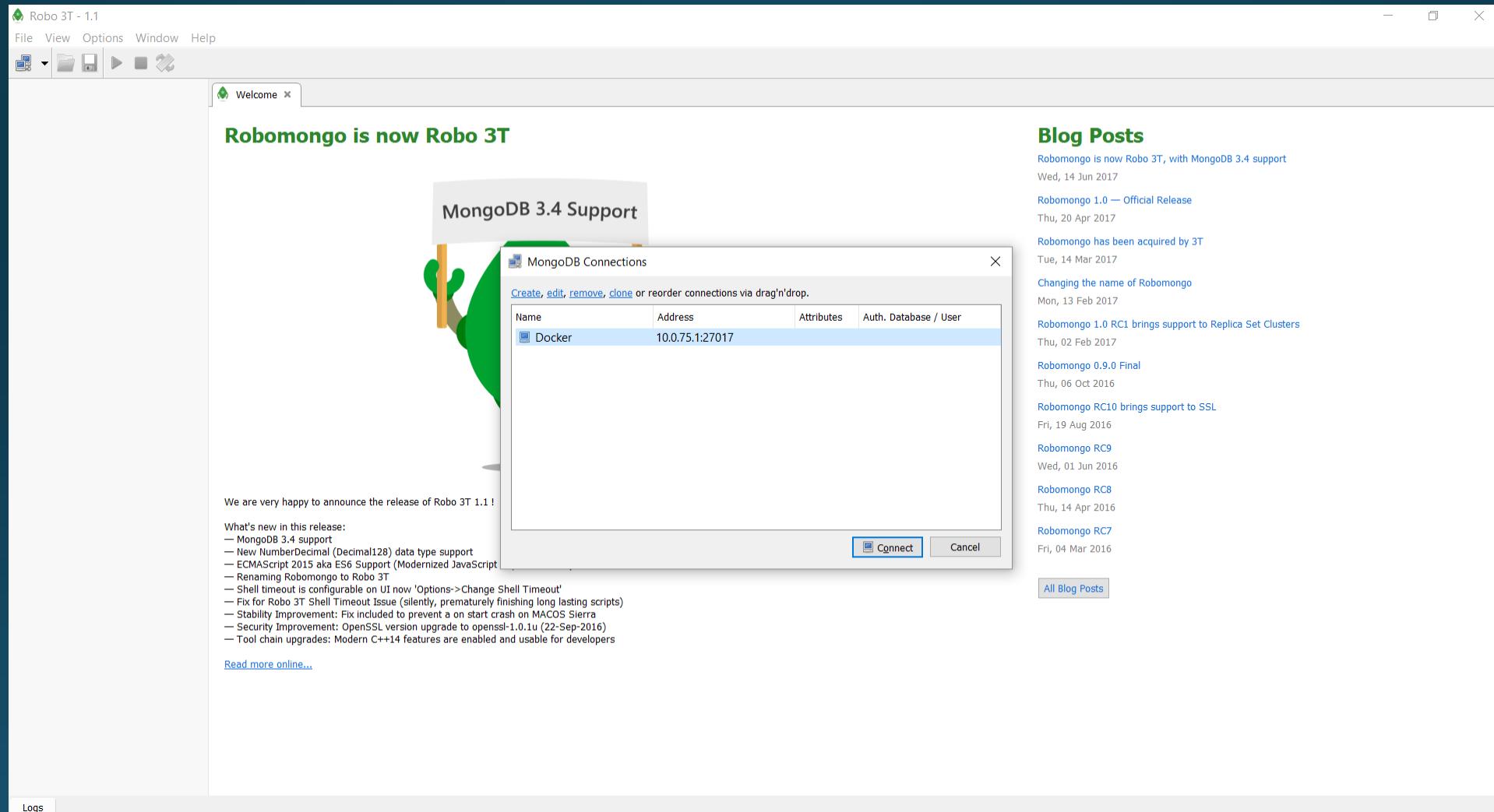
Are you serious about MongoDB? Choose Studio 3T - our fully featured IDE for MongoDB professionals.

- Fully featured IDE with embedded shell
- Visual Query Builder
- In-Place editing
- IntelliShell with Auto-Completion
- Query MongoDB with SQL
- Export to / import from SQL DB
- Aggregation Pipeline Editor
- And so much more...

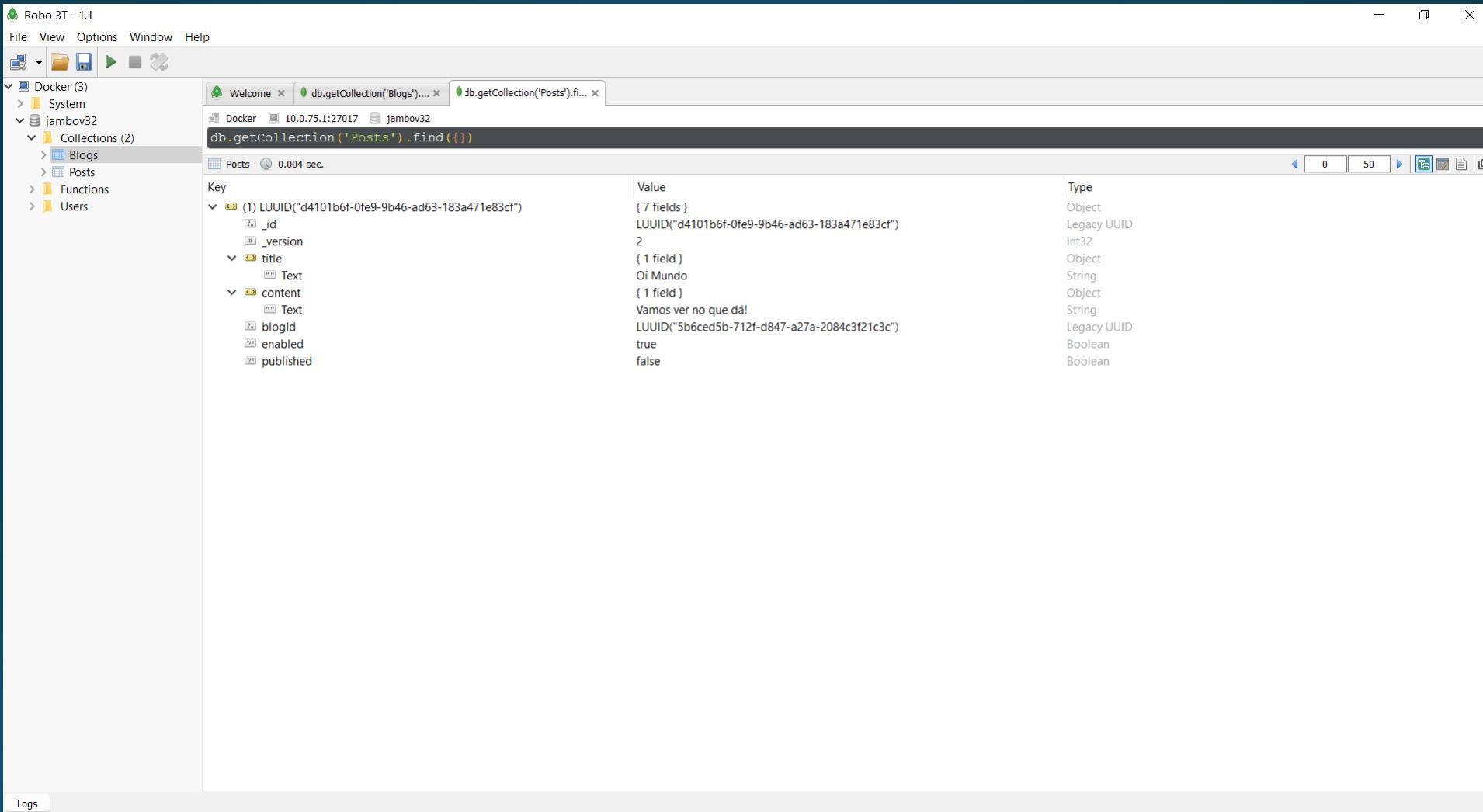
[Download Studio 3T](#)

Robomongo is Robo 3T

# Como eu acesso o MongoDB?



# Como eu acesso o MongoDB?

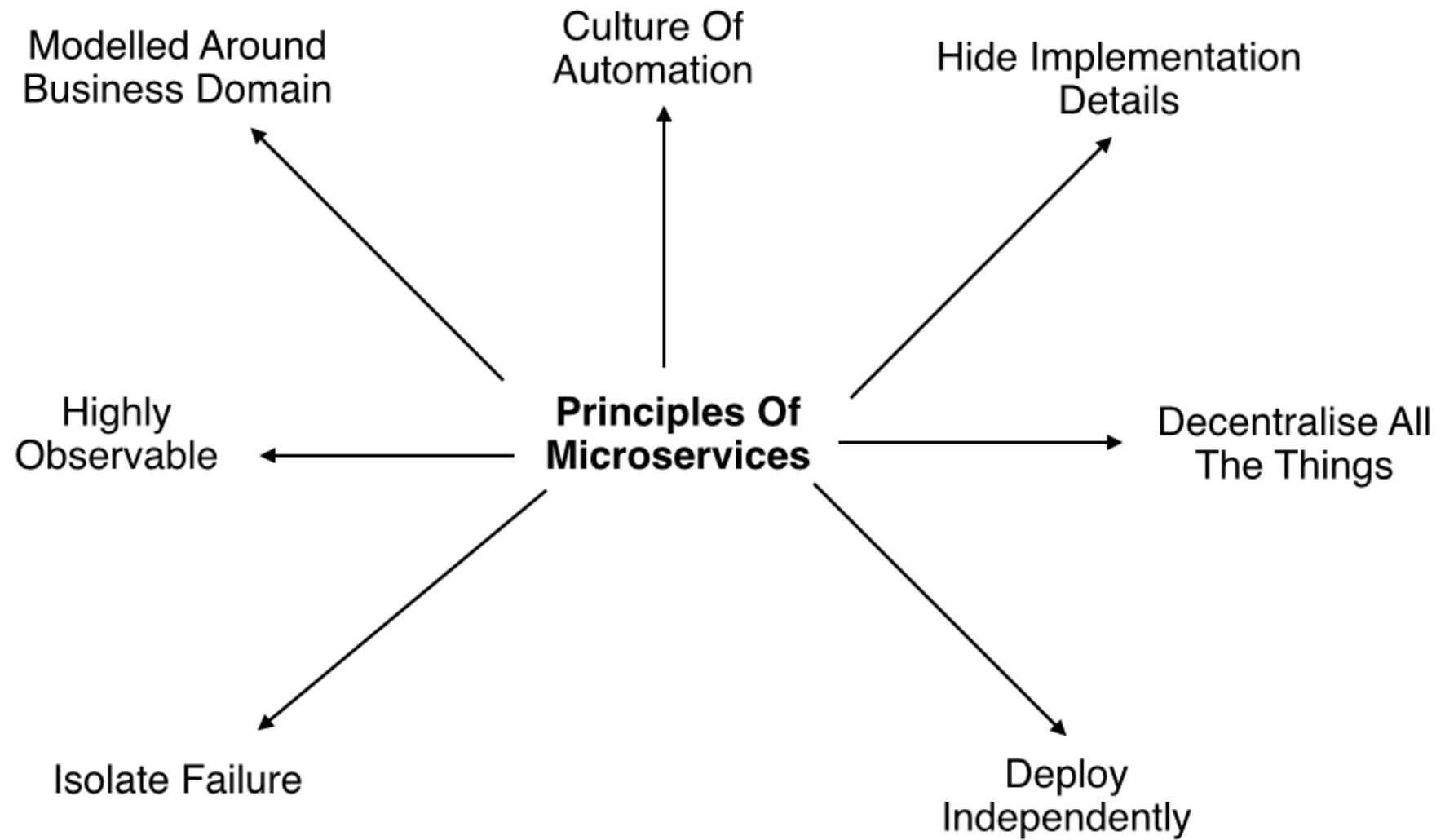


The screenshot shows the Robo 3T application interface. On the left is a sidebar with a tree view of databases and collections:

- Docker (3)
- System
- jambov32
  - Collections (2)
    - Blogs
    - Posts
  - Functions
  - Users

The main area has three tabs at the top: Welcome, db.getCollection('Blogs')..., and db.getCollection('Posts').fi... . The middle tab contains the command: db.getCollection('Posts').find({}). Below this is a table showing the results of the query:

Key	Value	Type
✓ (1) LUUID("d4101b6f-0fe9-9b46-ad63-183a471e83cf")	{ 7 fields }	Object
_id	UUID("d4101b6f-0fe9-9b46-ad63-183a471e83cf")	Legacy UUID
_version	2	Int32
title	{ 1 field }	Object
Text	Oi Mundo	String
content	{ 1 field }	Object
Text	Vamos ver no que dá!	String
blogId	UUID("5b6ced5b-712f-d847-a27a-2084c3f21c3c")	Legacy UUID
enabled	true	Boolean
published	false	Boolean



Como  
continuo o  
meu aprendizado?



ivanpaulovich/jambo: DD Ivan

[GitHub, Inc. \[US\] | https://github.com/ivanpaulovich/jambo/](https://github.com/ivanpaulovich/jambo/)

This repository Search Pull requests Issues Marketplace Explore

Unwatch 10 Unstar 25 Fork 10

Code Issues 0 Pull requests 0 Projects 3 Wiki Settings Insights

DDD + Aggregates + Event Sourcing com .NET Core, Kafka e MongoDB (Aplicações Produtora e Consumidora independentes) [Edit](http://www.100loop.com)

ddd ddd-architecture solid solid-principles event-sourcing event-driven dotnet-core kafka aggregate mongodb microservices  
microservice webapi bearer-authentication cqrs cqrs-es domain-driven-design service-bus csharp optimistic-updates Manage topics

196 commits 2 branches 2 releases 1 contributor Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

ivanpaulovich committed on GitHub ok Latest commit ed3afdd 6 days ago

docs	ok	12 days ago
images	Ok	12 days ago
install	ok	8 days ago
source	ok	8 days ago
.gitattributes	OK	11 months ago
.gitignore	Initial commit	11 months ago
CODE_OF_CONDUCT.md	Código de conduta	12 days ago
CONTRIBUTING.md	guide	12 days ago
LICENSE	Initial commit	11 months ago



<https://github.com/ivanpaulovich/jambo/>

ivanpaulovich/jambo: DD X Ivan - X

GitHub, Inc. [US] | https://github.com/ivanpaulovich/jambo/

This repository Search Pull requests Issues Marketplace Explore

Unwatch 10 Unstar 25 Fork 10

Code Issues 0 Pull requests 0 Projects 3 Wiki Settings Insights

DDD + Aggregates + Event Sourcing com .NET Core, Kafka e MongoDB (Aplicações Produtora e Consumidora independentes) Edit

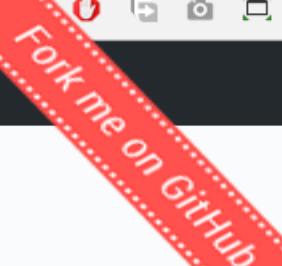
ddd ddd-architecture solid solid-principles event-sourcing event-driven dotnet-core kafka aggregate mongodb microservices  
microservice webapi bearer-authentication cqrs cqrs-es domain-driven-design service-bus csharp optimistic-updates Manage topics

196 commits 2 branches 2 releases 1 contributor Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

ivanpaulovich committed on GitHub ok Latest commit ed3afdd 6 days ago

docs	ok	12 days ago
images	Ok	12 days ago
install	ok	8 days ago
source	ok	8 days ago
.gitattributes	OK	11 months ago
.gitignore	Initial commit	11 months ago
CODE_OF_CONDUCT.md	Código de conduta	12 days ago
CONTRIBUTING.md	guide	12 days ago
LICENSE	initial commit	11 months ago



<https://github.com/ivanpaulovich/jambo/>

Dúvidas?

# Referências

- Implementing CQRS/ES in ASP.NET  
<https://www.exceptionnotfound.net/implementing-cqrs-in-net-part-1-architecting-the-application/>
- Pattern: Event sourcing by Chris Richardson  
<http://microservices.io/patterns/data/event-sourcing.html>
- Building Scalable Applications Using Event Sourcing and CQRS  
<https://initiate.andela.com/event-sourcing-and-cqrs-a-look-at-kafka-eoc1b9od17d8>