

TP9

Projeto de Bloco: Engenharia de Softwares Escaláveis



Guilherme di Savoia Nascimento

Prof. Armênio Torres

Data:30/12/2022

TP1

1. Descrever o domínio do sistema a ser desenvolvido.

O sistema a ser desenvolvido é um CRUD de geração de clientes para uma Petshop, para que os funcionários façam o cadastro dos clientes no banco de dados da empresa.

2. Listar e descrever os *Building Blocks* abaixo (incluindo os atributos) do modelo de domínio (pelo menos dois de cada):

1. Entidades

2. Objetos de Valor

3. Repositórios

O sistema possui duas entidades, ClientModel e MongoHelper, três objetos de valor, LojaValueObject, StoreAddressVO e ProductVO. Por fim, teremos um repositório, para armazenar as informações do cliente, o ClienteRepositorio.

A entidade ClienteModel é a entidade principal, cujo o microserviço é criado em volta, enquanto a entidade MongoHelper foi criada para configurar a conexão com o MongoDB.

Os objetos de valor LojaValueObject, StoreAddressVO e ProductVO foram criados para possibilitar que o microserviço possa buscar a lista de lojas, fazendo a chamada no microserviço StoreService, do aluno Natan Borges.

3. Fazer pelo menos três Casos de Uso do domínio.

Caso de uso 1: Criar Cliente

Ator : Operador do Sistema(Funcionário)

1. Ator recebe informações do cliente
2. Ator insere informações nos parâmetros de chamada da API
3. Ator envia chamada
4. Sistema recebe informações
5. Sistema cria o cliente

Caso de uso 1: Deletar Cliente

Ator : Operador do Sistema(Funcionário)

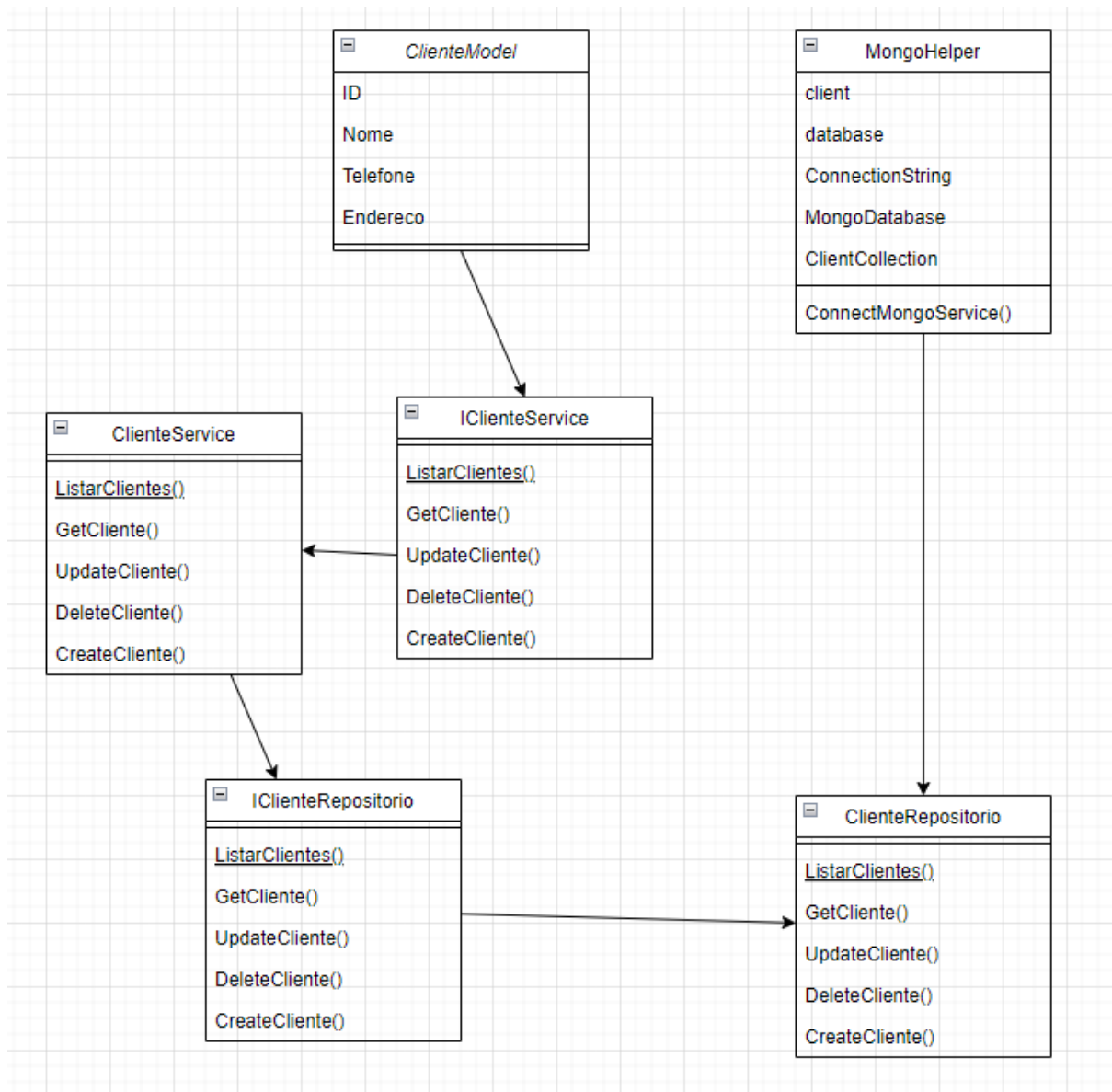
1. Receber pedido de deleção do cliente
2. Inserir ID do cliente na chamada da API
3. Enviar chamada
4. Sistema deleta o cliente.

Caso de uso 1: Editar Cliente

Ator : Operador do Sistema(Funcionário)

1. Ator insere ID do cliente a ser alterado
2. Sistema retorna informações do cliente
3. Ator altera as informações desejadas
4. Ator envia chamada
5. Sistema altera dados do cliente.

4. Fazer o Diagrama de Classes do modelo de domínio.



5. Identificar pelo menos dois serviços que podem ser oferecidos pelo domínio.

O domínio pode oferecer a criação de um cadastro na loja e a localização de uma loja próxima ao cliente.

TP3

6. Listar e descrever os *Building Blocks* abaixo (incluindo os atributos) do modelo de domínio (pelo menos dois de cada):

1. Agregados

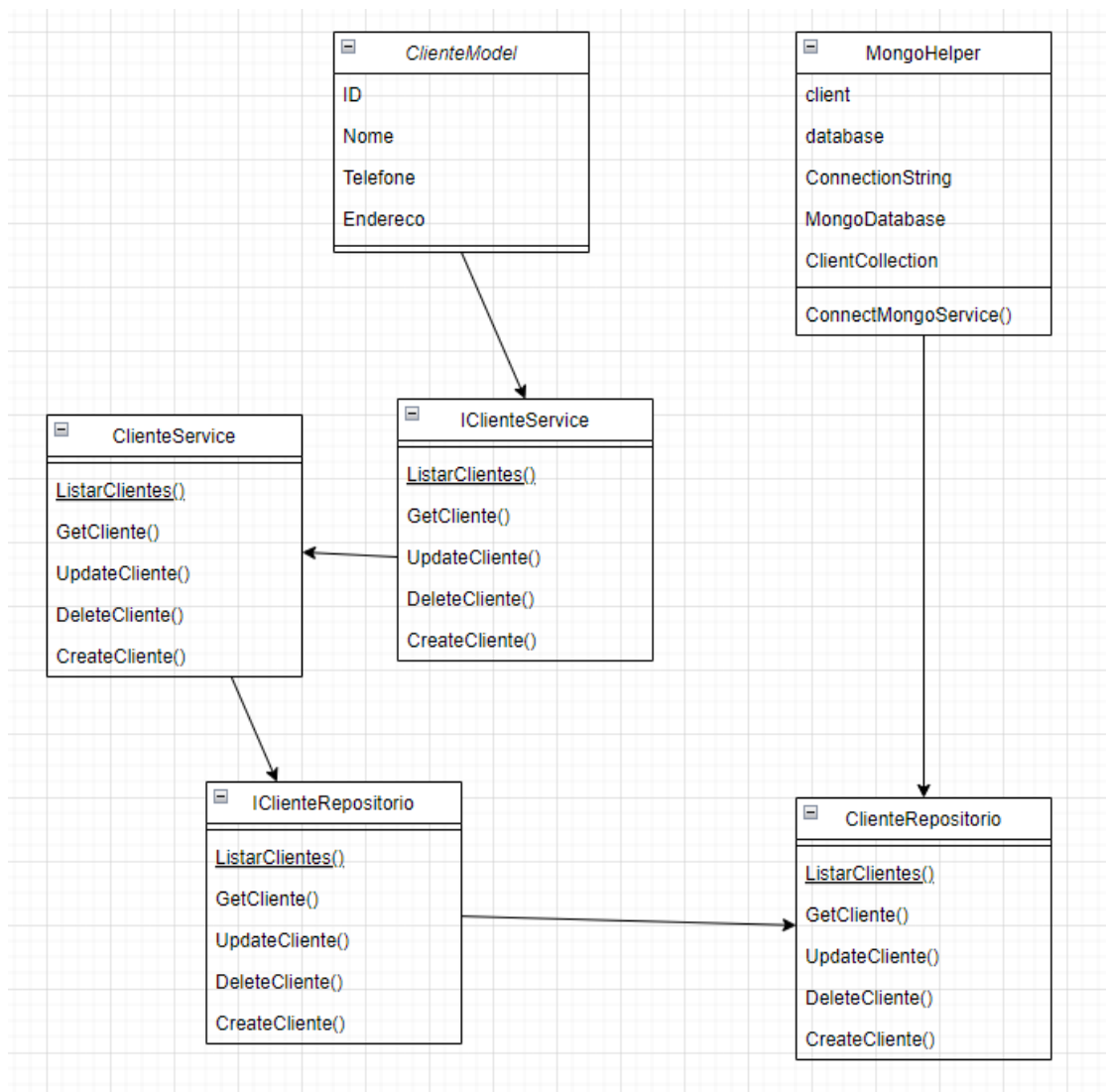
2. Serviços

O domínio não possui agregados mas possui dois serviços, o LojaService e o ClienteService. O ClienteService faz a comunicação entre o controller da API e o repositório, para fazer o armazenamento no banco de dados. o LojaService é utilizado para consumir a API StoreService, para poder fazer a listagem de lojas no banco de dados do microserviço.

7. Identificar pelo menos dois microserviços que podem ser oferecidos pelo domínio.

O domínio da Pet Friends pode oferecer microserviços de Assinaturas e de Lojas

8. Modelar o contexto delimitado de cada microserviço usando o diagrama de classes.



TP5

Descreva a estrutura das APIs disponibilizadas em cada Microserviço do seu projeto.

A descrição deverá conter:

9. Endpoints (paths)

Caminho, parâmetros e métodos HTTP disponibilizados.

Para o microserviço de cliente, foram criados quatro endpoints:

1. GET: <http://cliente.live/api/cliente>.
2. GET: <http://cliente.live/api/cliente/{id}>
3. POST: <http://cliente.live/api/cliente>
4. PUT: <http://cliente.live/api/cliente/{id}>
5. DELETE: <http://cliente.live/api/cliente/{id}>
6. GET: <http://cliente.live/api/cliente/loja>

10. Descrição textual dos Endpoints.

Os primeiros 2 endpoints são GETs, o primeiro sendo o endpoint para listar todos os clientes armazenados no banco de dados e o segundo GET é para buscar um cliente específico, pelo seu Id. O terceiro endpoint é um POST, para publicar as informações de um cliente no banco de dados. O quarto endpoint é um PUT, onde a informação do cliente é coletada pelo seu ID e editada. O quinto endpoint é um DELETE, funcionando similarmente ao PUT, que busca as informações do cliente pelo ID e deleta. O sexto e último endpoint é uma busca em outra API, a API de loja, retornando a lista de todas as lojas salvas no banco de dados do microserviço.

TP7

Para cada Microserviço previsto, liste e descreva:

11. Comandos (*Commands*)
12. Filas de Comandos
13. Consultas (*Queries*)
14. Eventos de Domínio (*Domain Events*)
15. Filas de Eventos

Não foi utilizada mensageria no microserviço pois foi acordado em aula que não seria necessário a utilização.

TP9

16. Implementar os dois Microserviços previstos em qualquer linguagem.

Organizar o código de pelo menos um dos Microserviços seguindo as camadas propostas pelo *Domain Driven Design* (*Application, Domain e Infrastructure*) e na camada de Domínio (*Domain*) diretórios com os nomes dos *Building Blocks* (*Entities, ValueObjects, Repositories, Factories, Services, Aggregates*).

O microserviço implementado neste projeto foi o microserviço de Cliente, desenvolvido em C# .NET. A aplicação foi separada em três camadas, com o ClienteController constituindo a camada de Aplicação, o ClienteModel, MongoHelper, LojaValueObject, ProductVO e StoreAddressVO constituindo o domínio. Por fim, o ClienteService, LojaService, ClienteRepositório e suas respectivas interfaces compõem a camada de infraestrutura.

- 17. Cada Microserviço deverá conter pelo menos um *resource* com no mínimo dois métodos implementados (GET, POST, PUT e DELETE) que faça uso de um serviço de domínio.**

Obs: Para Microserviço de CRUD (*Create, Read, Update e Delete*), fazer TODOS os métodos.

O microserviço de cliente possui todos os métodos do CRUD implementados, o ClienteController referencia o ClienteService, que por sua parte referencia o ClienteRepositório, que faz o armazenamento no banco de dados utilizando o MongoDB.

- 18. Consumir todos os métodos de pelo menos um dos Microserviços ou colocar documentação (Ex: Swagger). O consumo pode ser via Aplicação Web, Console, Mobile, Desktop ou um Microserviço consumindo o outro.**

O ClienteController consome o microserviço de Loja, possibilitando o cliente a fazer uma pesquisa de todas as lojas presentes no banco de dados da API. O plano inicial era fazer uma comparativa de lojas e clientes por CEP, mas não foi possível.

- 19. Implementar pelo menos um dos Microserviços usando Comandos (*Commands*), Consultas (*Queries*) e Filas (*Queues*). Pode ser fila em memória (Ex: Queue<T>), núvem, RabbitMQ etc.**

Não foi utilizada mensageria no microserviço pois foi acordado em aula que não seria necessário a utilização.

20. Implementar pelo menos um Microserviço com segurança (Ex: JWT) OU gerar evidências de deploy em cloud (prints de telas etc).

Para ter o deploy em cloud do microserviço de cliente, foi utilizado o ArgoCD, com ajuda do Professor Felipe Fink. Com isto, o deploy em cloud está sincronizado com o github, para que a aplicação esteja sempre sincronizada com as novas versões.

