

CHÁ

E-COMMERCE



DOCUMENTO DE VISÃO DE PROJETO

CHA_DVP_2022

Documento de Visão de Projeto desenvolvido para a Disciplina de
Laboratório de Engenharia de Software
Fatec Mogi | ADS
Aline dos Santos

HISTÓRICO DE VERSÕES

| DATA | VERSÃO | DESCRIÇÃO | AUTOR | REVISOR |
|------------|--------|-----------------------------------|--------------|---------|
| 27/02/2022 | 0.1.0 | Modelagem e Desenvolvimento | Aline Santos | - |
| 15/03/2022 | 0.1.1 | Análise dos Requisitos Levantados | Aline Santos | - |
| 12/04/2022 | 0.1.2 | Estudo das tecnologias utilizadas | Aline Santos | - |
| 17/05/2022 | 0.1.3 | Elaboração dos diagramas | Aline Santos | - |
| 07/06/2022 | 0.1.4 | Atualização dos diagramas | Aline Santos | - |
| 17/06/2022 | 0.1.5 | Atualização dos diagramas | Aline Santos | - |
| 28/06/2022 | 0.1.6 | Atualização dos diagramas | Aline Santos | - |

Cliente FATEC

Documento Documento de Visão de Projeto: chá.

Data 27 de fevereiro de 2022

Autor **Aline Santos**

aline.santos129@fatec.sp.gov.br

Revisado e Aprovado por: _____

28/06/2022

SUMÁRIO

| | |
|--|----|
| 1. Objetivo..... | 4 |
| 2. Necessidade do Negócio..... | 4 |
| 3. Objetivo do Projeto..... | 5 |
| 4. Declaração Preliminar de Escopo..... | 5 |
| 4.1. Descrição..... | 5 |
| 4.2. Produtos a serem entregues..... | 6 |
| 4.3. Requisitos..... | 6 |
| 4.3.1. Requisitos Funcionais..... | 6 |
| 4.3.2. Requisitos Não Funcionais..... | 7 |
| 4.3.3. Regras de Negócio..... | 8 |
| 5. Premissas..... | 8 |
| 6. Influência das Partes Interessadas..... | 8 |
| 7. Representação Arquitetural..... | 9 |
| 7.1. Restrições Arquiteturais..... | 13 |
| 8. Visão de Caso de Uso..... | 13 |
| 8.1. Visão geral do Sistema..... | 14 |
| 8.2. Realização dos Casos de Uso..... | 15 |
| 8.2.1. Gerenciar Produtos..... | 15 |
| 8.2.2. Gerenciar Pagamentos..... | 16 |
| 8.2.3. Comprar Produtos..... | 17 |
| 8.2.4. Solicitar Troca Produtos..... | 18 |
| 9. Visão de Lógica..... | 19 |
| 9.1. Camada de Domínio..... | 19 |
| 9.2. Camada de Controller..... | 21 |
| 9.3. Camada de Serviços..... | 22 |
| 9.4. Camada de Acesso aos Dados..... | 23 |
| 10. Visão de Implantação..... | 23 |
| 11. Visão de Implementação..... | 24 |
| 12. Visão de Dados..... | 24 |
| 13. Qualidade..... | 24 |
| 14. Referências..... | 24 |

1. Objetivo

Este documento apresenta uma visão do projeto que será desenvolvido na arquitetura em camadas dentro do ecossistema SpringBoot, permitindo a visualização da documentação das necessidades do cliente quanto ao uso de uma web application no contexto de uma loja online (e-commerce), justificativa do projeto e necessidades do negócio.

1.1. Escopo

Este documento apresenta o desenvolvimento de uma plataforma de vendas online que atenda às necessidades de um comércio de ervas de chá. Neste documento, apenas as partes mais significativas do ponto de vista da arquitetura serão apresentadas como forma de capturar e comunicar as decisões que foram tomadas em relação ao sistema aos desenvolvedores.

1.2. Referências

Este documento tem como referência o documento de requisitos **cha_drs_2022**.

2. Necessidade do Negócio

Para dar vazão à produção artesanal de ervas cultivadas pela agricultura familiar¹ faz-se necessário um sistema de vendas online de ervas de chá onde é seja possível gerenciar o produto vendido, o cliente, o processo de pagamento do cliente e a entrega do produto. O sistema ainda poderá ajudar no controle de estoque de produtos, geração de promoções por meio de cupons promocionais aos clientes usuais (top clientes).

¹ A ideia para esse modelo de negócios surge em função do artigo: Inovações na produção e beneficiamento de chás: o caso da agroindústria familiar “Consórcio Santa Gema de Plantas Medicinais”(PADILHA et al., 2021).

3. Objetivo do Projeto

Desenvolver uma web application que possa:

- Persistir dados em um SGDB
- Comunicar-se com outras partes do sistema por meio do protocolo HTTP
- Utilizado via Browser de forma responsiva

4. Declaração Preliminar de Escopo

Esta seção descreve o escopo do projeto. Para os detalhes dos requisitos de software vide documento de requisitos: **cha_drs_2022**.

4.1. Descrição

Este projeto tem como produto uma plataforma de comércio eletrônico (e-commerce), onde a oferta de produtos, venda e finalização do pagamento se dá completamente no formato digital. A plataforma de e-commerce está inserida no contexto de uma pequena loja de chás que tem como produto inicial ervas naturais para chás.

Cada produto ofertado possui embalagem única contendo a quantidade necessária para a infusão. A loja dispõe de diversos sabores para os produtos, classificados por categorias, assim o cliente que desejar deverá selecionar o produto e a quantidade desejada. Para que o cliente possa comprar e finalizar o pagamento é preciso estar logado em sua conta na plataforma do e-commerce e possuir um endereço de entrega cadastrado.

Obs. O Sistema de pagamentos utilizado até o momento da confecção deste documento é via Stripe, portanto não é necessário que o cliente, nesta etapa do projeto, cadastre o seu cartão de crédito.

4.2. Produtos a serem entregues

Nesta primeira etapa do projeto serão considerados produtos:

- CRUD Cliente
 - Implementado em conjunto: CRUD Cartão de Crédito (com especificações de tipo de bandeira), CRUD Endereço (com especificações de tipo de residência e tipo de endereço – entrega/cobrança)
- CRUD Produto
 - Implementado em conjunto: CRUD Categoria e Painel do Administrador (gerenciar os produtos, categorias, visualizar clientes cadastrados)
- Fluxo de venda
 - Implementado em conjunto: Carrinho de Compras, Sessão de Checkout (confirmar o pedido antes de prosseguir para o pagamento), Solicitação de Troca de pedido, Aplicação de cupom.
- Sessão de Pagamentos
 - Implementado em conjunto: Sessão de Checkout via Stripe API
 - Sistema de pagamento próprio da loja.

4.3. Requisitos

Os requisitos apresentados foram definidos no início do projeto e podem sofrer alterações. Para os detalhes dos requisitos, vide **cha_drs_2022**.

4.3.1. Requisitos Funcionais

O sistema deve permitir a inclusão, alteração, consulta e exclusão dos produtos ofertados no comércio.

O sistema deve permitir a associação de categorias aos produtos ofertados.

O sistema deve ser capaz de listar as categorias com mais produtos associados.

O sistema deve permitir a inclusão, alteração, consulta e exclusão de clientes.

O sistema deve ser capaz de permitir o login apenas com email e senha do usuário.

O sistema deve permitir a associação de endereços e cartões ao cliente cadastrado na plataforma.

O sistema deve ser capaz de exibir para o administrador o painel contendo o gerenciamento de produtos e clientes.

O sistema deve ser capaz de gerar um pedido associado ao cliente logado e adequado ao processamento de pagamentos.

O sistema deve ser capaz de permitir a troca de um dos itens do pedido e gerar um cupom no valor do item solicitado, mediante aprovação.

4.3.2. Requisitos Não Funcionais

O sistema deverá ser desenvolvido em Java 11, com framework SpringBoot MVC tendo como implementação da JPA o SpringData JPA por meio do Hibernate.

O sistema deverá utilizar driver de conexão para SGDB adequado ao MariaDB.

O sistema deverá utilizar o padrão de projeto MVC² e camadas.

O sistema deverá ser responsivo e funcionar nos navegadores: Firefox, Google Chrome, Edge, Vivaldi, Brave.

O sistema, deverá utilizar o conceito de token para guardar em cache o usuário logado na plataforma para retornar os dados referentes ao cliente associado.

² MVC considera três papéis: o modelo é um objeto que representa as informações do domínio (não visual e possui o comportamento POJO). A view apresenta o modelo para a interface do usuário por meio da renderização das tecnologias de frontend. O Controller orquestra as entradas do usuário, manipula o modelo e reflete as mudanças na view. Assim o padrão MVC apresenta, principalmente, duas camadas de separação: separação da apresentação do modelo e separação da view e controller (FOWLER, 2003).

4.3.3. Regras de Negócio

O sistema deverá garantir que para cada ofertado, uma categoria estará associada. A associação de categorias deverá ser feita pelo administrador da plataforma no momento do cadastro do produto, permitindo a adição de novas categorias.

O sistema deverá garantir que o cadastro de endereços do cliente deverá consistir na composição dos dados cadastrais (vide cha_drs_2022 RN0023).

5. Premissas

O fluxo de desenvolvimento do projeto será acompanhado pelo board disponibilizado no Trello.

6. Influência das Partes Interessadas

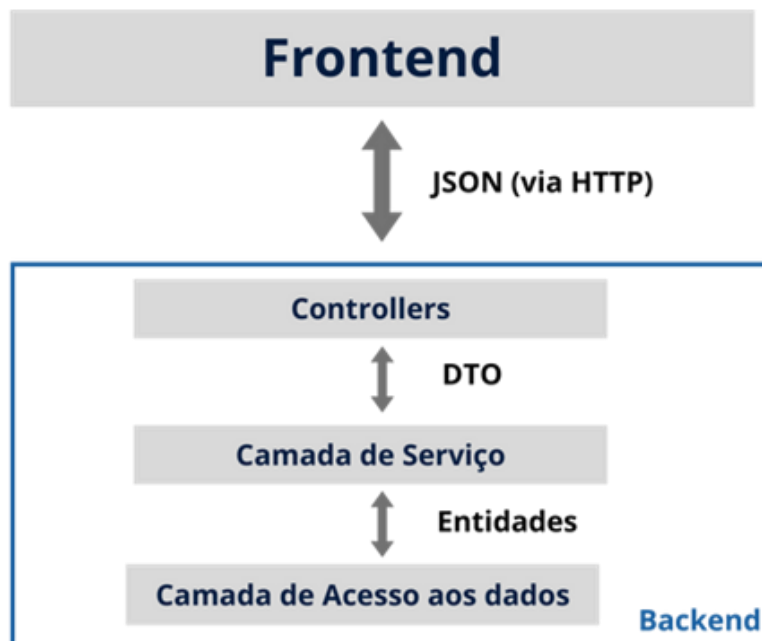
Apresentação dos Stakeholders (envolvidos com a plataforma):

- Usuário Administrador
 - Perfil: Ana Clara Prestes, administradora da cooperativa “MaisChá” e coordenadora do grupo de vendas e distribuição de chás artesanais produzidas pelo cultivo de ervas através da agricultura familiar na região do Vale do Ribeira.
- Usuário Cliente:
 - Perfil: Guilherme Torres, historiador e fotógrafo profissional, viaja pelo mundo em busca de conhecer a cultura local e, por meio da fotografia, dar voz ao mundo. Amante de chás tradicionais da Índia, busca incentivar a produção interna preferindo o produto mais natural e feito em solo brasileiro.

7. Representação Arquitetural

A plataforma desenvolvida para o comércio eletrônico seguirá o padrão de arquitetura de sistemas em camadas, em um nível de mais alta abstração as camadas são apresentadas na figura 1:

Figura 1 – Visão da arquitetura em camadas do sistema.



Fonte: o autor.

Nessa perspectiva, o padrão arquitetural permite modelar as classes de domínio em pacotes “model”, e possuem como aspecto global os dados mais relevantes ao contexto do negócio.

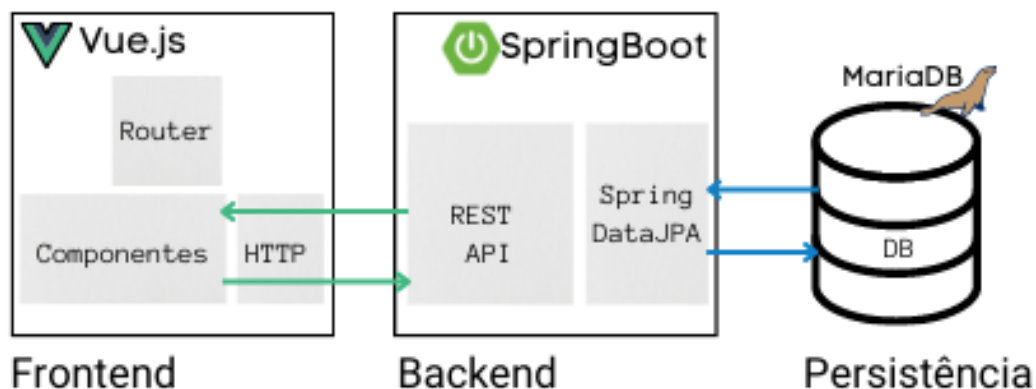
As interações realizadas na view, por meio de template views apresentados ao usuário, a comunicação ocorre por meio do protocolo HTTP, e as classes controllers são responsáveis por orquestrar as requisições e retornando uma resposta para a view.

Neste projeto, uma camada de Service foi aplicada para auxiliar a classe Controller, assim as requisições que chegam são recebidas pela controller que aciona a camada de serviços que aplicar a lógica e as regras de negócios necessárias ao contexto do comércio eletrônico.

Vale ressaltar ainda que, para a persistência de dados o acesso é realizado por uma classe especializada em persistência por meio de Repositórios, e o padrão

MVC representa a forma como as classes estão organizadas internamente, a forma com a qual a aplicação web funciona implementa o padrão REST. O diagrama na figura 2 representa em um alto nível de abstração a arquitetura completa do sistema:

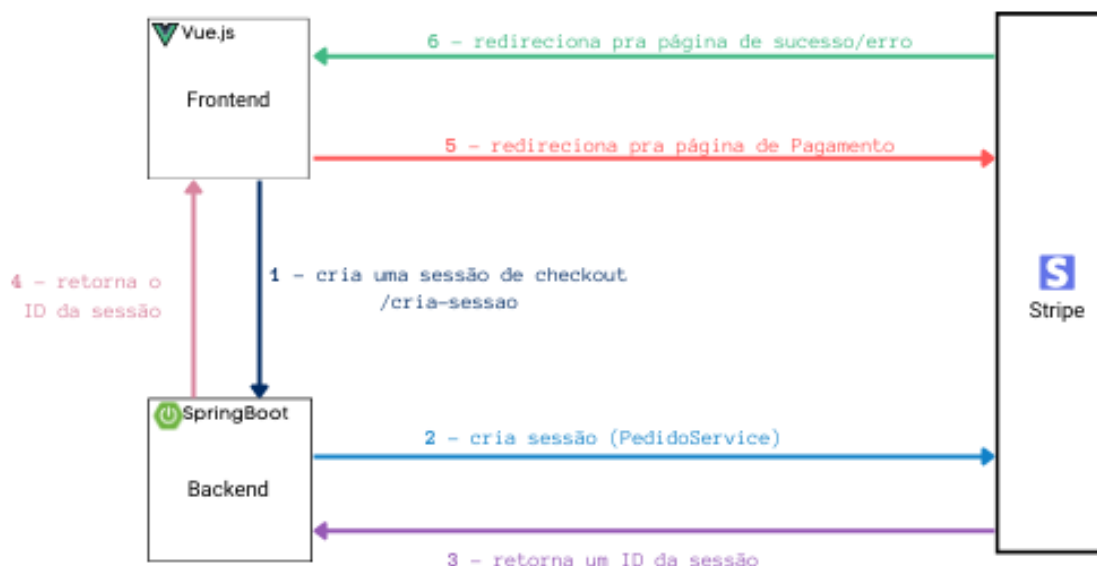
Figura 2 – Visão geral do sistema da aplicação web.



Fonte: Autor.

No momento atual do projeto, há ainda o uso de um serviço especializado em pagamentos e é possível observar como a comunicação entre o sistema desenvolvido e o serviço de pagamentos ocorre através da figura 3, que apresenta uma visão geral do funcionamento:

Figura 3 – Esquema fluxo de funcionamento do Sistema de Pagamento integrado ao projeto.



Fonte: Autor.

Vale ainda apresentar que a implementação do backend em Java utilizando-se do padrão MVC faz uso de padrões de projeto que permitem sua representação em diagrama de blocos para melhor visualizar suas camadas:

Figura 4 – Abstração em camadas da arquitetura geral do projeto.

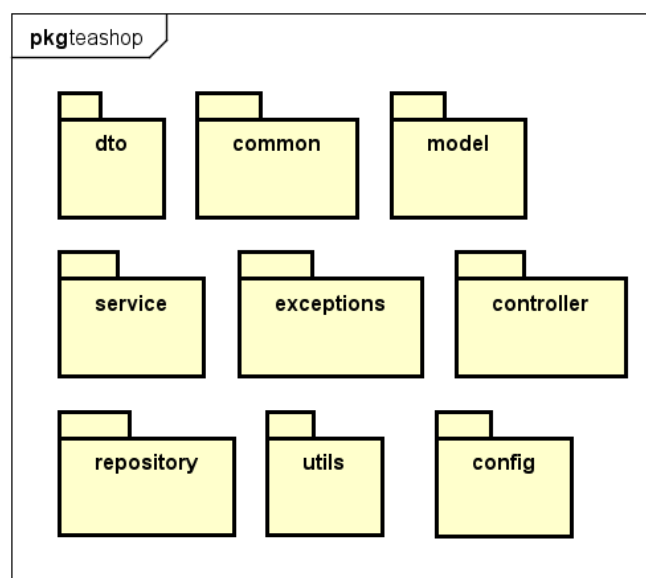


Fonte: autor.

As classes de domínio do projeto são modeladas considerando a arquitetura apresentada e para transferir os dados dos objetos ao longo da aplicação utilizou-se classes do padrão Data Transfer Object (DTO), para reduzir a exposição de dados desnecessários da classe de domínio.

A organização das classes de domínio é apresentada em diagramas de pacotes na figura 5:

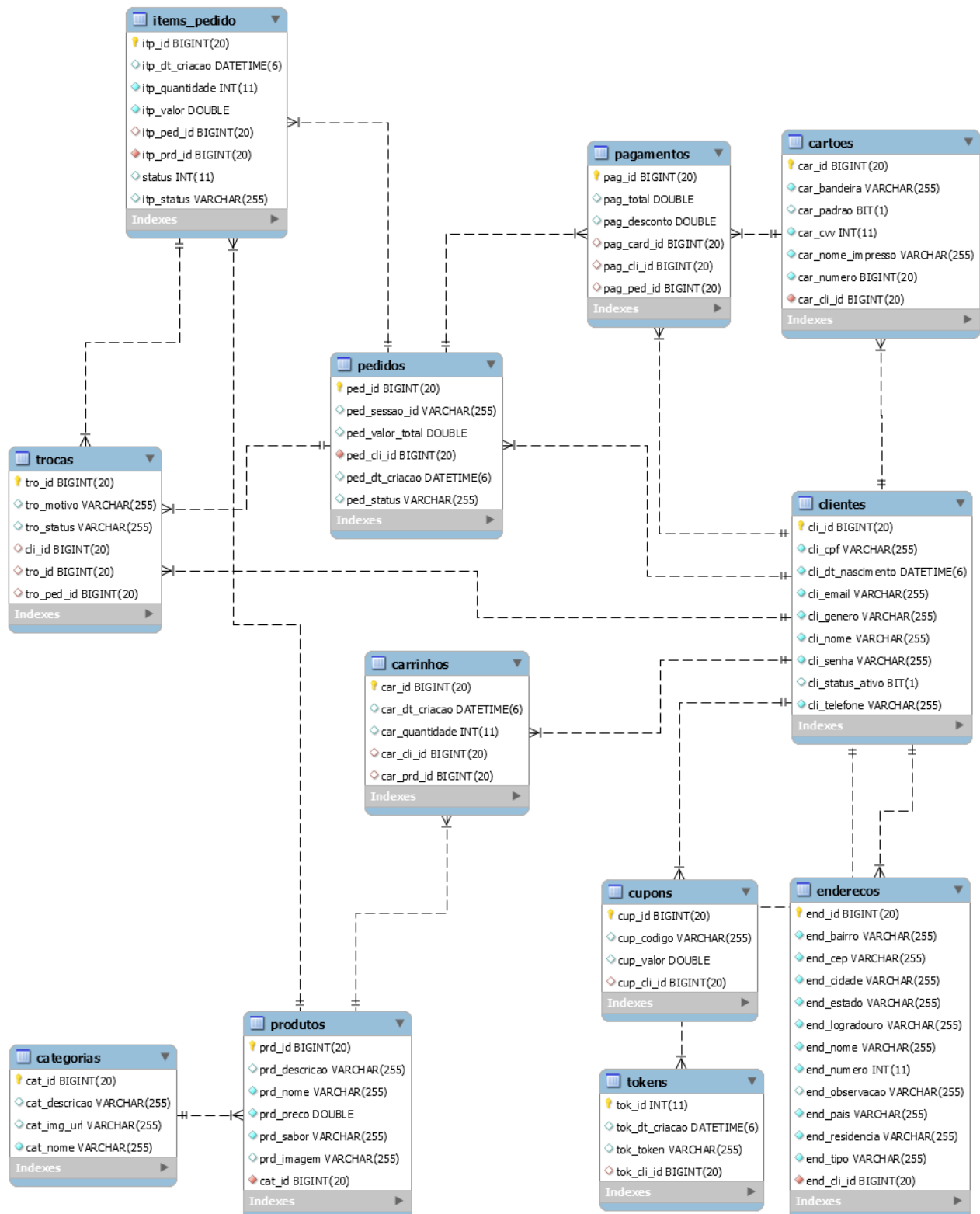
Figura 5 – Representação geral do sistema na perspectiva de pacotes



Fonte: autor, gerado AstahUML

No que se refere à persistência de dados, o banco foi modelado considerando o contexto de negócios e o diagrama de entidade relacionamentos é apresentado na figura 7:

Figura 7 – Representação geral do banco de dados modelado para o sistema



Fonte: autor, gerado via MySQL Workbench por meio de engenharia reversa

7.1. Restrições Arquiteturais

Para o desenvolvimento do sistema as seguintes tecnologias foram utilizadas:

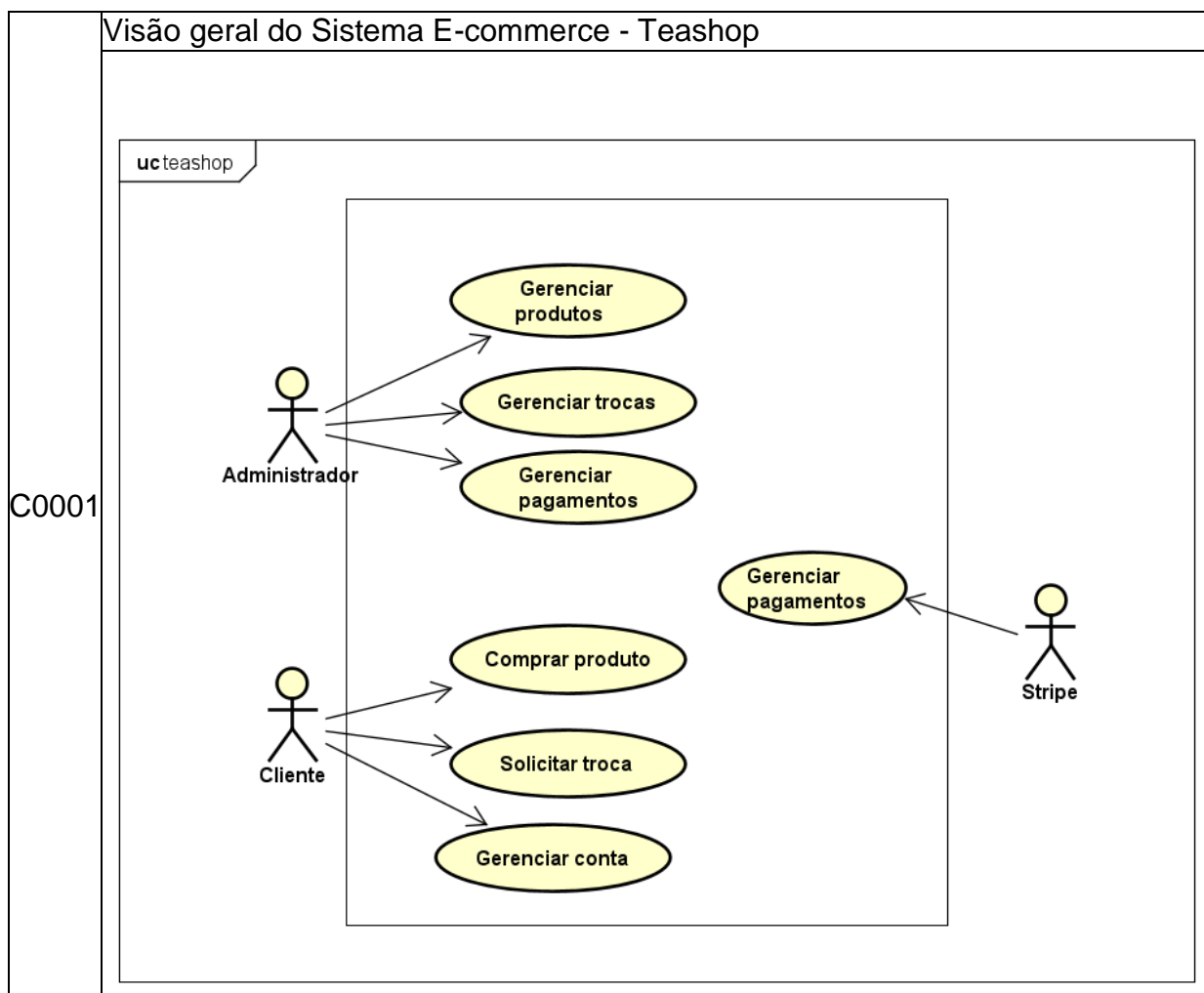
- JDK 1.8
- Java 11
- Framework SpringBoot 2.6.4
- Maven 4.0
- MariaDB
- Vue 3 / Axios
- Bootstrap 4.0
- Node v15

8. Visão de Caso de Uso

Nesta seção, os casos de uso de maior relevância para o entendimento do sistema serão apresentados em formatos de diagramas UC (Use Case), sendo eles: Gerenciamento de produtos, Gerenciamento de trocas, Gerenciamento de pagamentos, Realização de Compras, Solicitações de trocas, gerenciamento de Contas e Gerenciamento de Pagamentos, representando assim os casos de uso de configuração, condução e análise.

O Gerenciamento de pagamentos neste projeto pode ser realizado por meio de um sistema externo, apresentado no diagrama, ou por meio interno que será gerenciado pelo Administrador em Gerenciar Pagamentos.

8.1. Visão geral do Sistema



| Caso de Uso de Configuração: Cadastrar Produtos | |
|---|-----------------------|
| Identificação | Descrição |
| C0002 | Cadastrar produto |
| C0003 | Atualizar produto |
| C0004 | Remover produto |
| C0005 | Listar/exibir produto |

| Caso de Uso de Análise: Gerenciar Pagamento de Produtos | |
|---|------------------------------|
| Identificação | Descrição |
| C0006 | Receber pedido de pagamento |
| C0007 | Aceitar pedido de pagamento |
| C0008 | Rejeitar pedido de pagamento |
| C0009 | Listar pedidos de pagamento |

| Caso de Uso de Condução: Comprar Produtos | |
|---|--|
| Identificação | Descrição |
| C0010 | Exibir lista de produtos |
| C0011 | Selecionar quantidade de produtos |
| C0012 | Adicionar produto ao carrinho de compras |
| C0013 | Confirmar pagamento |
| C0014 | Solicitar troca de item do pedido |
| C0015 | Utilizar cupom de troca em pagamento de pedido |

8.2. Realização dos Casos de Uso

Esta seção apresentará a descrição dos casos de uso C002, C006 e C010-15

8.2.1. Gerenciar Produtos

| | |
|-----------------|---|
| Nome do Cenário | Cadastrar Produto |
| Ator | Administrador |
| Diagrama | DC001 |
| Caso de Uso | C0002 |
| Pré-Condição | Categoria de produtos cadastrada. Vide RN0011 do cha_dsr_2022. |
| Fluxo Normal | <ol style="list-style-type: none"> 1. O administrador acessa o painel de gerenciamento de produtos. 2. O sistema exibe a tela de opções de gerenciamento. |

| | |
|-------------------|---|
| | <ol style="list-style-type: none"> 3. O administrador acessa o formulário de cadastro de novos produtos através do botão “Adicionar Produto” 4. O administrador preenche o formulário com os dados exigidos. 5. O administrador salva os dados inseridos através do botão “salvar”. 6. O sistema salva os dados no banco de dados e exibe uma mensagem de sucesso ao administrador. 7. O sistema redireciona o administrador para a tela inicial do painel de gerenciamento. |
| Fluxo alternativo | - |
| Pós-condição | O produto cadastrado ficará disponível para compras. |

8.2.2. Gerenciar Pagamentos

| | |
|-------------------|--|
| Nome do Cenário | Receber pedido de Pagamento |
| Ator | Cliente |
| Diagrama | DC001 |
| Caso de Uso | C0006 |
| Pré-Condição | Realizar compra de itens adicionados ao carrinho (RF0033) |
| Fluxo Normal | <ol style="list-style-type: none"> 1. O sistema recebe o pedido de pagamento do cliente. 2. O sistema realiza verificações junto ao sistema de cartões de crédito. 3. O sistema valida o cartão utilizado como forma de pagamento. 4. O sistema informa que o pagamento foi aceito pela operadora do cartão do cliente. 5. O sistema informa que o pedido foi aprovado. |
| Fluxo alternativo | - |

| | |
|--------------|---|
| Pós-condição | Pedido segue para o processo de entrega ao endereço do cliente. |
|--------------|---|

8.2.3. Comprar Produtos

| | |
|-----------------|--|
| Nome do Cenário | Comprar produtos |
| Ator | Cliente |
| Diagrama | DC001 |
| Caso de Uso | C0010-13 |
| Pré-Condição | Cliente cadastrado na plataforma (RF0021), produto cadastrado (RF0011). |
| Fluxo Normal | <ol style="list-style-type: none">1. O sistema exige autenticação do usuário.2. Após autenticação o sistema exibe a tela inicial do comércio eletrônico contendo os produtos.3. O cliente acessa o produto desejado clicando sobre o card do produto.4. O sistema exibe os detalhes do produto.5. O cliente define a quantidade de itens deste produto ele deseja comprar no campo “quantidade”.6. O cliente adiciona o produto ao carrinho de compras através do botão “adicionar ao carrinho”.7. O sistema atualiza o carrinho de compras do cliente contabilizando o valor total até o momento.8. O cliente acessa o carrinho de compras clicando no ícone do carrinho de compras.9. O sistema exibe a tela com todos os produtos adicionados ao carrinho com o valor total para cada item (quantidade * item) e o custo total de todos os itens adicionados ao carrinho.10. O cliente confirma a compra através do botão “confirmar”. |

| | |
|-------------------|--|
| | <p>11. O sistema exibe a tela de informações sobre o sistema de pagamentos.</p> <p>12. O cliente consente com o sistema de pagamentos.</p> <p>13. O sistema redireciona o cliente para uma sessão de pagamentos para efetuar o pagamento.</p> <p>14. O sistema exibe uma tela com a descrição da compra e o formulário para pagamento.</p> <p>15. O cliente insere os dados solicitados para o pagamento e confirma o pagamento por meio do botão “pagar”.</p> <p>16. O sistema redireciona o cliente para a tela de “sucesso”.</p> <p>17. O sistema direciona o cliente para a lista de pedidos realizados.</p> |
| Fluxo alternativo | <p>Caso o cliente decida não confirmar a compra na etapa 10:</p> <p>11. O sistema redireciona o cliente para a tela de “erro”.</p> <p>12. O sistema mantém os itens no carrinho.</p> |
| Pós-condição | O pedido ficará disponível para aprovação de pagamento. |

8.2.4. Solicitar Troca Produtos

| | |
|-----------------|--|
| Nome do Cenário | Solicitar troca de produto. |
| Ator | Cliente |
| Diagrama | DC001 |
| Caso de Uso | C0014-15 |
| Pré-Condição | Pedido realizado. |
| Fluxo Normal | <ol style="list-style-type: none"> 1. O sistema exige autenticação do usuário. 2. Após autenticação o sistema exibe a tela inicial do comércio eletrônico contendo os produtos. 3. O cliente acessa os pedidos realizados acessando o link “Meus Pedidos” |

| | |
|-------------------|---|
| | <ol style="list-style-type: none">4. O sistema exibe a lista de pedidos realizados.5. O cliente acessa os detalhes do pedido clicando sobre o número do pedido.6. O cliente solicita troca do item desejado por meio do botão “Solicitar troca”.7. O sistema apresenta a tela de solicitação de troca.8. O cliente insere a descrição do motivo da troca e envia a solicitação para ser aprovada por meio do botão “enviar”.9. O sistema redireciona o cliente para a tela de pedidos. |
| Fluxo alternativo | n/a |
| Pós-condição | A solicitação de troca ficará disponível para aprovação. Um cupom de troca ficará disponível para utilização em compras futuras. |

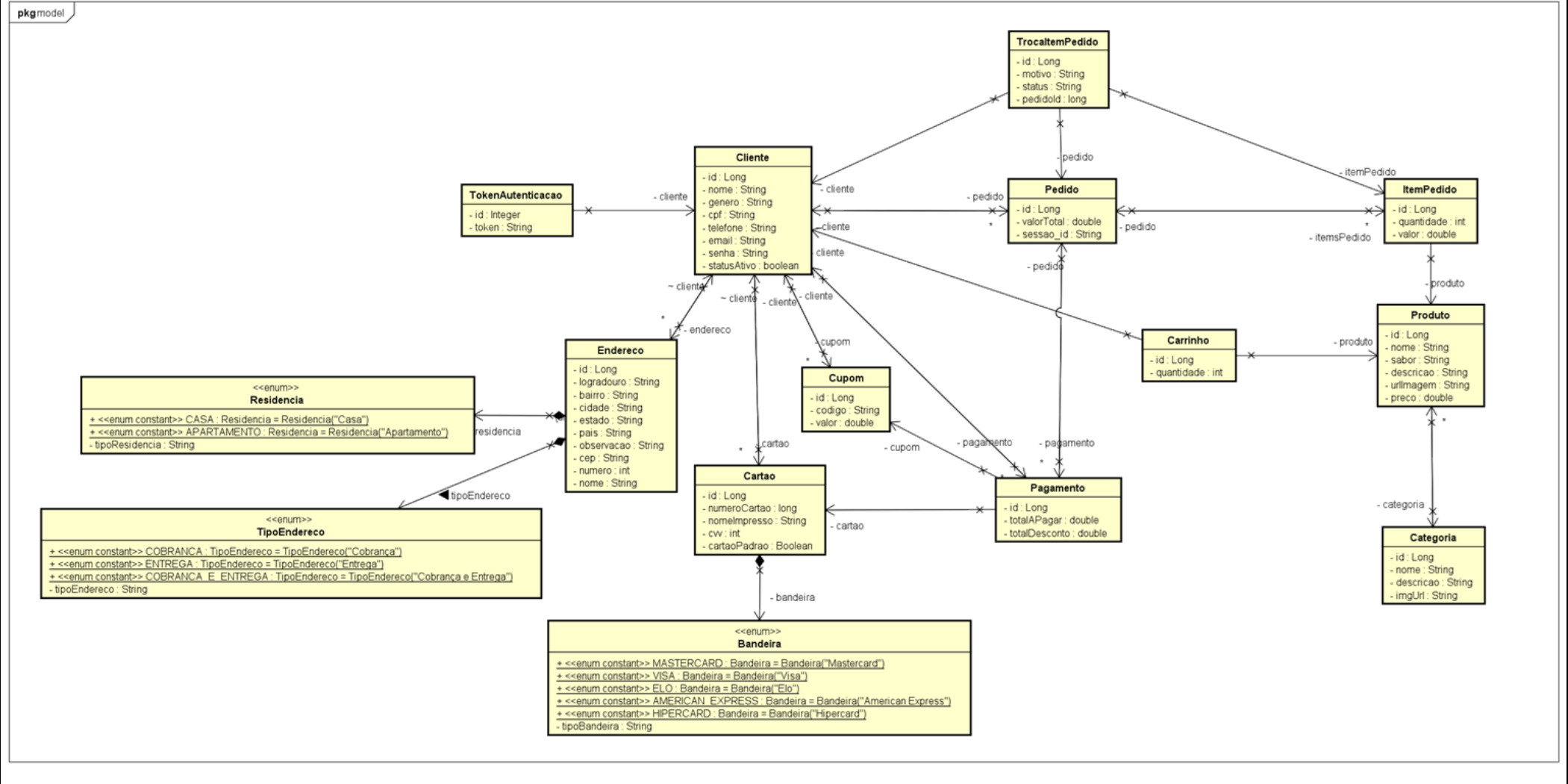
9. Visão de Lógica

Esta seção apresenta a visão lógica das camadas do sistema por meio da representação de diagramas de classes.

9.1. Camada de Domínio

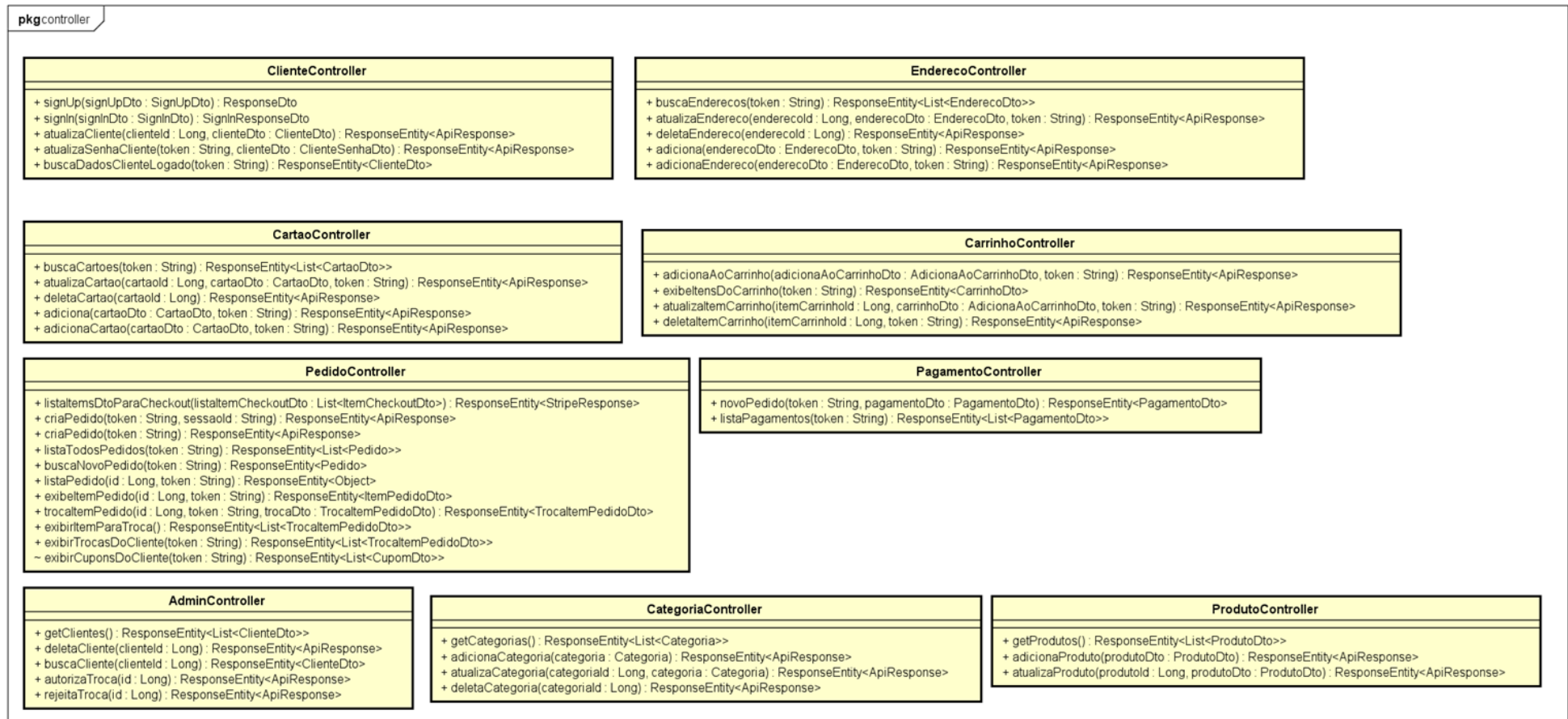
A camada de domínio apresenta visão das classes de referentes aos modelos do contexto do comércio eletrônico para chás e é representada pelo diagrama de classes DC0001:

DC0001 – Diagrama de Classes Camada de Domínio



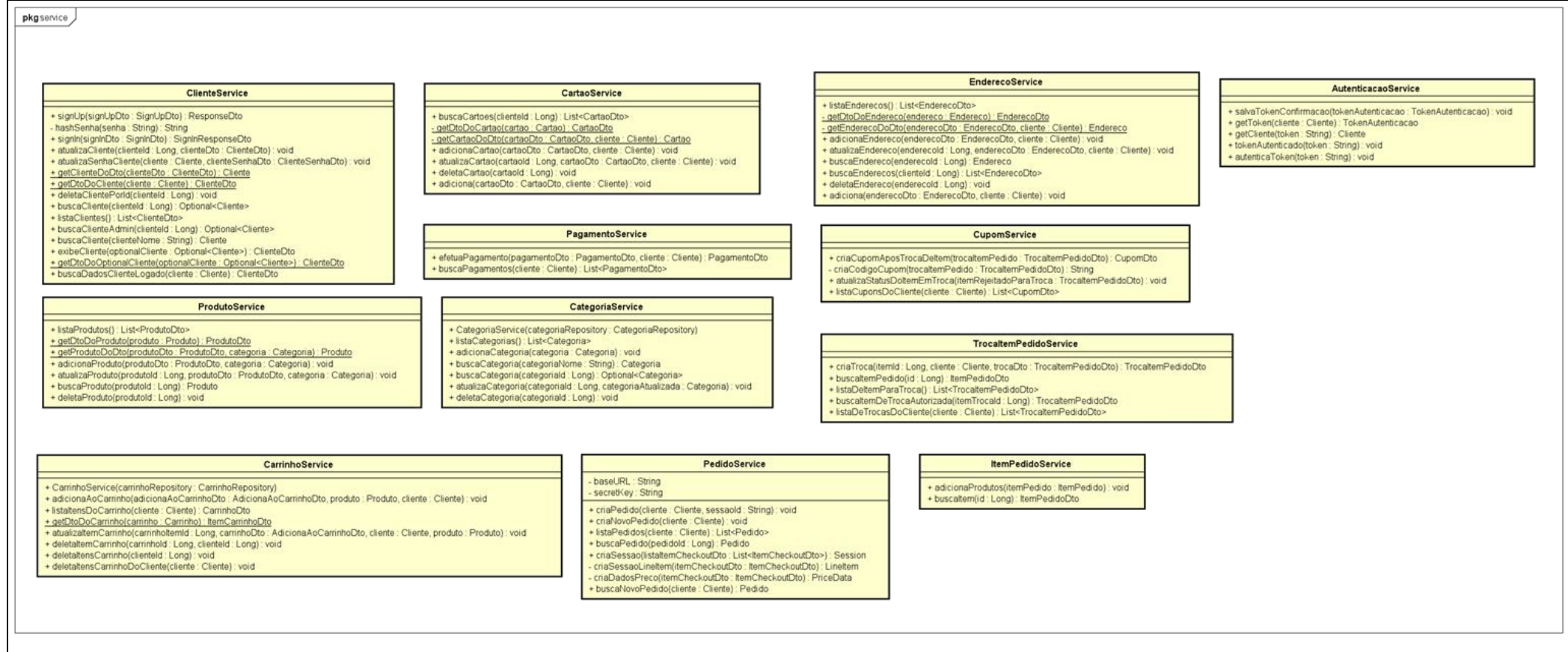
9.2. Camada de Controller

DC0002 – Diagrama de Classes da Camada de Controller



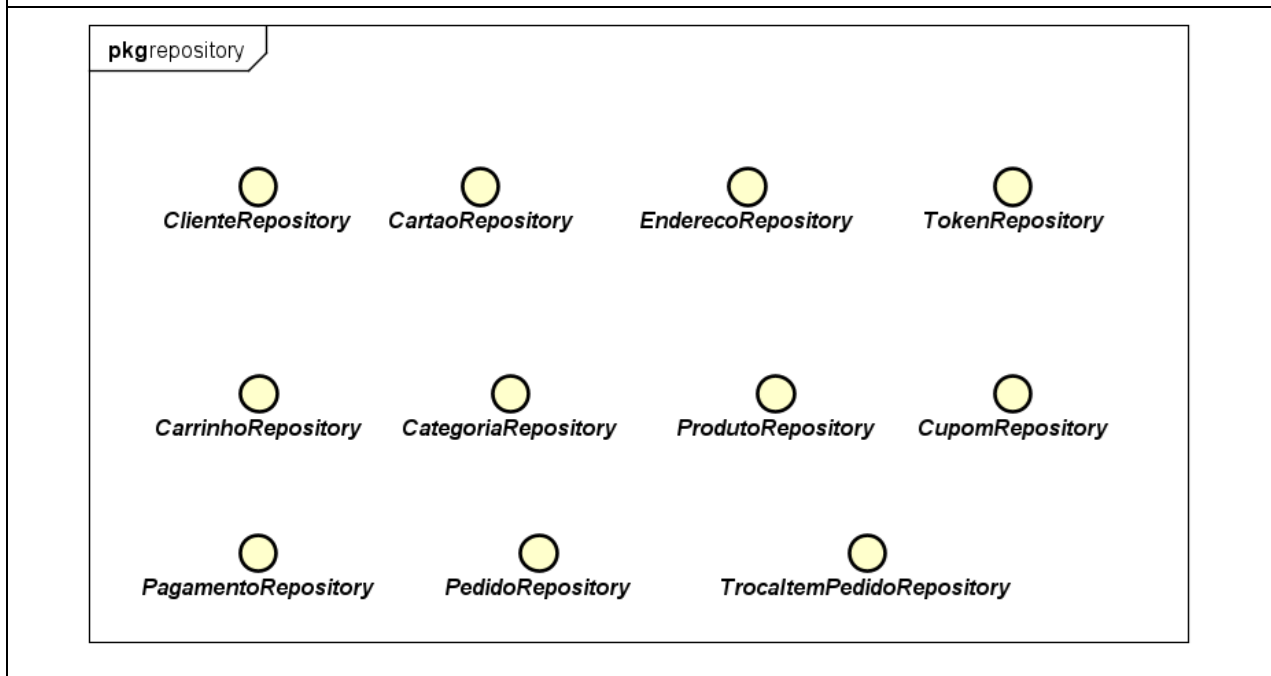
9.3. Camada de Serviços

DC0003 – Diagrama de Classes da camada de Serviços



9.4. Camada de Acesso aos Dados

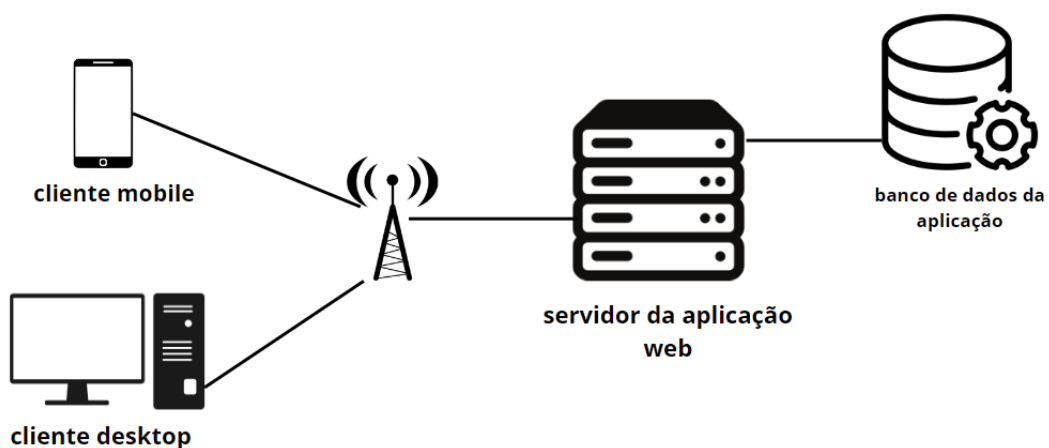
DC0004 – Diagrama de Classes da Camada de Acesso aos Dados



10. Visão de Implantação

Esta seção apresenta as configurações da rede física na qual o Teashop será implantado e executado. Apresenta a visão do Modelo de Implantação em um nível mais alto de abstração dos componentes que executarão o sistema.

Figura 8 – Implantação da Aplicação Web Teashop



Fonte: autor.

Os clientes mobile e desktop acessam a mesma aplicação web, que é apresentada conforme o tipo de dispositivo de acesso, sendo portanto uma aplicação única disponibilizada em diferentes formatos, utilizando o Vue.js como biblioteca para renderização do frontend tendo Bootstrap como principal framework para preparo da responsividade da camada de apresentação da aplicação.

A aplicação utiliza um único banco de dados para persistir todos os dados referentes às ações realizadas na aplicação pelo cliente ou pelo administrador da aplicação.

11. Visão de Implementação

A estrutura geral da aplicação para a Teashop está baseada na visão geral da camada Lógica.

12. Visão de Dados

A estrutura da camada de persistência de dados para a aplicação está apresentada na visão geral de dados, onde o modelo relacional foi obtido através da engenharia reversa do banco de dados criado por meio do Hibernate, implementação da especificação JPA. O mapeamento das entidades por meio do Hibernate permite a criação do banco de dados, conforme apresentado no script de criação em anexo.

13. Qualidade

O sistema no estado atual não é considerado apto para produção pois falha em grandes aspectos de qualidade por não apresentar uma camada mais robusta de Segurança, uma camada de Interceptadores para alto volume de requisições, não possui uma camada mais robusta de tratamento de exceções, sendo assim um sistema aberto a melhorias.

14. Referências

FOWLER, M. **Patterns of enterprise application architecture**. Boston: Addison-Wesley, 2003.

PADILHA, A. C. M. et al. Inovações na produção e beneficiamento de chás: o caso da agroindústria familiar “Consórcio Santa Gema de Plantas Medicinais”. **Espacio Abierto**, v. 30, n. 1, p. 147–165, 2021.

Documento de Requisitos cha_dsr_2022.

ANEXO 01 – Script de criação do BD –

-- MySQL Script generated by MySQL Workbench

-- Tue Jun 28 09:05:53 2022

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO
ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- Schema mydb

-- Schema ecommerce_v2

-- Schema ecommerce_v2

CREATE SCHEMA IF NOT EXISTS `ecommerce_v2` DEFAULT CHARACTER SET latin1 ;

USE `ecommerce_v2` ;

-- Table `ecommerce_v2`.`categorias`

CREATE TABLE IF NOT EXISTS `ecommerce_v2`.`categorias` (

```
`cat_id` BIGINT(20) NOT NULL AUTO INCREMENT,  
`cat_descricao` VARCHAR(255) NULL DEFAULT NULL,  
`cat_img_url` VARCHAR(255) NULL DEFAULT NULL,  
`cat_nome` VARCHAR(255) NOT NULL,  
PRIMARY KEY (`cat_id`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 5  
DEFAULT CHARACTER SET = latin1;
```

```
-----  
-- Table `ecommerce_v2`.`produtos`  
-----
```

```
CREATE TABLE IF NOT EXISTS `ecommerce_v2`.`produtos` (  
`prd_id` BIGINT(20) NOT NULL AUTO_INCREMENT,  
`prd_descricao` VARCHAR(255) NULL DEFAULT NULL,  
`prd_nome` VARCHAR(255) NOT NULL,  
`prd_preco` DOUBLE NOT NULL,  
`prd_sabor` VARCHAR(255) NOT NULL,  
`prd_imagem` VARCHAR(255) NULL DEFAULT NULL,  
`cat_id` BIGINT(20) NOT NULL,  
PRIMARY KEY (`prd_id`),  
INDEX `FKmrlfxoenc3de9ju8w5on3j5uv` (`cat_id` ASC) VISIBLE,  
CONSTRAINT `FKmrlfxoenc3de9ju8w5on3j5uv`  
FOREIGN KEY (`cat_id`)  
REFERENCES `ecommerce_v2`.`categorias` (`cat_id`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 32  
DEFAULT CHARACTER SET = latin1;
```

```
-----  
-- Table `ecommerce_v2`.`clientes`  
-----
```

```
CREATE TABLE IF NOT EXISTS `ecommerce_v2`.`clientes` (  
  `cli_id` BIGINT(20) NOT NULL AUTO_INCREMENT,  
  `cli_cpf` VARCHAR(255) NOT NULL,  
  `cli_dt_nascimento` DATETIME(6) NOT NULL,  
  `cli_email` VARCHAR(255) NOT NULL,  
  `cli_genero` VARCHAR(255) NOT NULL,  
  `cli_nome` VARCHAR(255) NOT NULL,  
  `cli_senha` VARCHAR(255) NOT NULL,  
  `cli_status_ativo` BIT(1) NULL DEFAULT NULL,  
  `cli_telefone` VARCHAR(255) NOT NULL,  
  PRIMARY KEY (`cli_id`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 10  
DEFAULT CHARACTER SET = latin1;
```

```
-----  
-- Table `ecommerce_v2`.`carrinhos`  
-----
```

```
CREATE TABLE IF NOT EXISTS `ecommerce_v2`.`carrinhos` (  
  `car_id` BIGINT(20) NOT NULL AUTO_INCREMENT,  
  `car_dt_criacao` DATETIME(6) NULL DEFAULT NULL,  
  `car_quantidade` INT(11) NULL DEFAULT NULL,  
  `car_cli_id` BIGINT(20) NULL DEFAULT NULL,  
  `car_prd_id` BIGINT(20) NULL DEFAULT NULL,  
  PRIMARY KEY (`car_id`),  
  INDEX `FKslcibfhlj6k1hxmf7y1x836en` (`car_cli_id` ASC) VISIBLE,  
  INDEX `FKh88nu42vrjfnex3hsq0cm7x75` (`car_prd_id` ASC) VISIBLE,  
  CONSTRAINT `FKh88nu42vrjfnex3hsq0cm7x75`  
  FOREIGN KEY (`car_prd_id`)  
  REFERENCES `ecommerce_v2`.`produtos` (`prd_id`),  
  CONSTRAINT `FKslcibfhlj6k1hxmf7y1x836en`  
  FOREIGN KEY (`car_cli_id`)  
  REFERENCES `ecommerce_v2`.`clientes` (`cli_id`))
```

ENGINE = InnoDB

AUTO_INCREMENT = 103

DEFAULT CHARACTER SET = latin1;

-- Table `ecommerce_v2`.`cartoes`

CREATE TABLE IF NOT EXISTS `ecommerce_v2`.`cartoes` (

 `car_id` BIGINT(20) NOT NULL AUTO_INCREMENT,

 `car_bandeira` VARCHAR(255) NOT NULL,

 `car_padrao` BIT(1) NULL DEFAULT NULL,

 `car_cvv` INT(11) NOT NULL,

 `car_nome_impresso` VARCHAR(255) NOT NULL,

 `car_numero` BIGINT(20) NOT NULL,

 `car_cli_id` BIGINT(20) NOT NULL,

 PRIMARY KEY (`car_id`),

 INDEX `FKcyrssq02ils4a6gwbwy0fufwu` (`car_cli_id` ASC) VISIBLE,

 CONSTRAINT `FKcyrssq02ils4a6gwbwy0fufwu`

 FOREIGN KEY (`car_cli_id`)

 REFERENCES `ecommerce_v2`.`clientes` (`cli_id`))

ENGINE = InnoDB

AUTO_INCREMENT = 72

DEFAULT CHARACTER SET = latin1;

-- Table `ecommerce_v2`.`cupons`

CREATE TABLE IF NOT EXISTS `ecommerce_v2`.`cupons` (

 `cup_id` BIGINT(20) NOT NULL AUTO_INCREMENT,

 `cup_codigo` VARCHAR(255) NULL DEFAULT NULL,

 `cup_valor` DOUBLE NULL DEFAULT NULL,

 `cup_cli_id` BIGINT(20) NULL DEFAULT NULL,

```
PRIMARY KEY (`cup_id`),  
INDEX `FK6mydit3ae7fbnq9maam0n7xf5` (`cup_cli_id` ASC) VISIBLE,  
CONSTRAINT `FK6mydit3ae7fbnq9maam0n7xf5`  
FOREIGN KEY (`cup_cli_id`)  
REFERENCES `ecommerce_v2`.`clientes` (`cli_id`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 8  
DEFAULT CHARACTER SET = latin1;
```

```
-----  
-- Table `ecommerce_v2`.`enderecos`  
-----  
CREATE TABLE IF NOT EXISTS `ecommerce_v2`.`enderecos` (  
`end_id` BIGINT(20) NOT NULL AUTO_INCREMENT,  
`end_bairro` VARCHAR(255) NOT NULL,  
`end_cep` VARCHAR(255) NOT NULL,  
`end_cidade` VARCHAR(255) NOT NULL,  
`end_estado` VARCHAR(255) NOT NULL,  
`end_logradouro` VARCHAR(255) NOT NULL,  
`end_nome` VARCHAR(255) NOT NULL,  
`end_numero` INT(11) NOT NULL,  
`end_observacao` VARCHAR(255) NULL DEFAULT NULL,  
`end_pais` VARCHAR(255) NOT NULL,  
`end_residencia` VARCHAR(255) NOT NULL,  
`end_tipo` VARCHAR(255) NOT NULL,  
`end_cli_id` BIGINT(20) NOT NULL,  
PRIMARY KEY (`end_id`),  
INDEX `FKavbrj5wu8ujdejmv7ecosev0u` (`end_cli_id` ASC) VISIBLE,  
CONSTRAINT `FKavbrj5wu8ujdejmv7ecosev0u`  
FOREIGN KEY (`end_cli_id`)  
REFERENCES `ecommerce_v2`.`clientes` (`cli_id`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 68
```

DEFAULT CHARACTER SET = latin1;

-- Table `ecommerce_v2`.`pedidos`

CREATE TABLE IF NOT EXISTS `ecommerce_v2`.`pedidos` (
 `ped_id` BIGINT(20) NOT NULL AUTO INCREMENT,
 `ped_sessao_id` VARCHAR(255) NULL DEFAULT NULL,
 `ped_valor_total` DOUBLE NULL DEFAULT NULL,
 `ped_cli_id` BIGINT(20) NOT NULL,
 `ped_dt_criacao` DATETIME(6) NULL DEFAULT NULL,
 `ped_status` VARCHAR(255) NULL DEFAULT NULL,
 PRIMARY KEY (`ped_id`),
 INDEX `FKIt3tis6egug1b2p2i8h1c4y5n` (`ped_cli_id` ASC) VISIBLE,
 CONSTRAINT `FKIt3tis6egug1b2p2i8h1c4y5n`
 FOREIGN KEY (`ped_cli_id`)
 REFERENCES `ecommerce_v2`.`clientes` (`cli_id`))
ENGINE = InnoDB
AUTO INCREMENT = 91
DEFAULT CHARACTER SET = latin1;

-- Table `ecommerce_v2`.`items_pedido`

CREATE TABLE IF NOT EXISTS `ecommerce_v2`.`items_pedido` (
 `itp_id` BIGINT(20) NOT NULL AUTO INCREMENT,
 `itp_dt_criacao` DATETIME(6) NULL DEFAULT NULL,
 `itp_quantidade` INT(11) NOT NULL,
 `itp_valor` DOUBLE NOT NULL,
 `itp_ped_id` BIGINT(20) NULL DEFAULT NULL,
 `itp_prd_id` BIGINT(20) NOT NULL,
 `status` INT(11) NULL DEFAULT NULL,

```
`itp_status` VARCHAR(255) NULL DEFAULT NULL,  
PRIMARY KEY (`itp_id`),  
INDEX `FKsj7w4njwbrm4q619vf8l8ffy6` (`itp_ped_id` ASC) VISIBLE,  
INDEX `FKrnhgm0y4r5lrb8dssl3ynj6vs` (`itp_prd_id` ASC) VISIBLE,  
CONSTRAINT `FKrnhgm0y4r5lrb8dssl3ynj6vs`  
FOREIGN KEY (`itp_prd_id`)  
REFERENCES `ecommerce_v2`.`produtos` (`prd_id`),  
CONSTRAINT `FKsj7w4njwbrm4q619vf8l8ffy6`  
FOREIGN KEY (`itp_ped_id`)  
REFERENCES `ecommerce_v2`.`pedidos` (`ped_id`))  
ENGINE = InnoDB  
AUTO INCREMENT = 89  
DEFAULT CHARACTER SET = latin1;
```

```
-----  
-- Table `ecommerce_v2`.`lista_desejos`  
-----
```

```
CREATE TABLE IF NOT EXISTS `ecommerce_v2`.`lista_desejos` (  
`lst_id` BIGINT(20) NOT NULL AUTO INCREMENT,  
`lst_dt_criacao` DATETIME(6) NULL DEFAULT NULL,  
`lst_cli_id` BIGINT(20) NOT NULL,  
`lst_prd_id` BIGINT(20) NULL DEFAULT NULL,  
PRIMARY KEY (`lst_id`),  
INDEX `FK6xq1ioqtvmdxg1b42dmy5pny5` (`lst_cli_id` ASC) VISIBLE,  
INDEX `FK4qs8poogq4w56hntxensdew9f` (`lst_prd_id` ASC) VISIBLE,  
CONSTRAINT `FK4qs8poogq4w56hntxensdew9f`  
FOREIGN KEY (`lst_prd_id`)  
REFERENCES `ecommerce_v2`.`produtos` (`prd_id`),  
CONSTRAINT `FK6xq1ioqtvmdxg1b42dmy5pny5`  
FOREIGN KEY (`lst_cli_id`)  
REFERENCES `ecommerce_v2`.`clientes` (`cli_id`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

-- Table `ecommerce_v2`.`pagamentos`

```
CREATE TABLE IF NOT EXISTS `ecommerce_v2`.`pagamentos` (  
  `pag_id` BIGINT(20) NOT NULL AUTO INCREMENT,  
  `pag_total` DOUBLE NULL DEFAULT NULL,  
  `pag_desconto` DOUBLE NULL DEFAULT NULL,  
  `pag_card_id` BIGINT(20) NULL DEFAULT NULL,  
  `pag_cli_id` BIGINT(20) NULL DEFAULT NULL,  
  `pag_ped_id` BIGINT(20) NULL DEFAULT NULL,  
  PRIMARY KEY (`pag_id`),  
  INDEX `FKm5ahajrkmhouanhmikilof2pi` (`pag_card_id` ASC) VISIBLE,  
  INDEX `FKm2lmjo2m8wyixe0i0fnh491t` (`pag_cli_id` ASC) VISIBLE,  
  INDEX `FK6wjm3roqirp7xmx09m84e23dh` (`pag_ped_id` ASC) VISIBLE,  
  CONSTRAINT `FK6wjm3roqirp7xmx09m84e23dh`  
    FOREIGN KEY (`pag_ped_id`)  
    REFERENCES `ecommerce_v2`.`pedidos` (`ped_id`),  
  CONSTRAINT `FKm2lmjo2m8wyixe0i0fnh491t`  
    FOREIGN KEY (`pag_cli_id`)  
    REFERENCES `ecommerce_v2`.`clientes` (`cli_id`),  
  CONSTRAINT `FKm5ahajrkmhouanhmikilof2pi`  
    FOREIGN KEY (`pag_card_id`)  
    REFERENCES `ecommerce_v2`.`cartoes` (`car_id`))  
ENGINE = InnoDB  
AUTO INCREMENT = 8  
DEFAULT CHARACTER SET = latin1;
```

-- Table `ecommerce_v2`.`tokens`

```
CREATE TABLE IF NOT EXISTS `ecommerce_v2`.`tokens` (  
  `tok_id` INT(11) NOT NULL AUTO INCREMENT,  
  `tok_dt_criacao` DATETIME(6) NULL DEFAULT NULL,  
  `tok_token` VARCHAR(255) NULL DEFAULT NULL,  
  `tok_cli_id` BIGINT(20) NULL DEFAULT NULL,
```



```
PRIMARY KEY (`tok_id`),  
INDEX `fk_cli_id` (`tok_cli_id` ASC) VISIBLE,  
CONSTRAINT `fk_cli_id`  
FOREIGN KEY (`tok_cli_id`)  
REFERENCES `ecommerce_v2`.`clientes` (`cli_id`)  
ON DELETE CASCADE)  
ENGINE = InnoDB  
AUTO_INCREMENT = 10  
DEFAULT CHARACTER SET = latin1;  
  
-----  
-- Table `ecommerce_v2`.`trocas`  
-----  
  
CREATE TABLE IF NOT EXISTS `ecommerce_v2`.`trocas` (  
`tro_id` BIGINT(20) NOT NULL AUTO_INCREMENT,  
`tro_motivo` VARCHAR(255) NULL DEFAULT NULL,  
`tro_status` VARCHAR(255) NULL DEFAULT NULL,  
`cli_id` BIGINT(20) NULL DEFAULT NULL,  
`tro_id` BIGINT(20) NULL DEFAULT NULL,  
`tro_ped_id` BIGINT(20) NULL DEFAULT NULL,  
PRIMARY KEY (`tro_id`),  
INDEX `FK1dtyojdr6ulqf6ytw3cn8cnlm` (`cli_id` ASC) VISIBLE,  
INDEX `FKmet3incajapcm8ys5lnb7aw9g` (`tro_id` ASC) VISIBLE,  
INDEX `FK1pnrmcddb3amce1wsbrh9w2` (`tro_ped_id` ASC) VISIBLE,  
CONSTRAINT `FK1dtyojdr6ulqf6ytw3cn8cnlm`  
FOREIGN KEY (`cli_id`)  
REFERENCES `ecommerce_v2`.`clientes` (`cli_id`),  
CONSTRAINT `FK1pnrmcddb3amce1wsbrh9w2`  
FOREIGN KEY (`tro_ped_id`)  
REFERENCES `ecommerce_v2`.`pedidos` (`ped_id`),  
CONSTRAINT `FKmet3incajapcm8ys5lnb7aw9g`  
FOREIGN KEY (`tro_id`)  
REFERENCES `ecommerce_v2`.`items_pedido` (`itp_id`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 54
```

DEFAULT CHARACTER SET = latin1;

SET SQL_MODE=@OLD_SQL_MODE;

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;

SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;