



***CEFET-RJ - CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA***

Campus - Maria da graça

SIAMS0402 - Análise e Modelagem de Sistemas

Sistema Vai&Volta Viagens(VVV)

Grupo: Bruno Santana de Souza, Geise Camillo de Oliveira, Israel da Cunha de Oliveira e Jefferson Graciano da Silva

Professor: Sildenir Alves Ribeiro

Rio de Janeiro - RJ

Dezembro - 2024



Sumário

1 - Introdução.....	3
1.1 Sobre o trabalho.....	3
1.2 Sobre as atividades desenvolvidas.....	3
1.3 Sobre a interação do grupo e a divisão das tarefas.....	3
1.4 Descrição do Sistema.....	3
 2 - Análise e Modelagem.....	4
2.1 - Análise e especificação dos requisitos.....	4
2.1.1 - Requisitos funcionais.....	4
2.1.2 - Requisitos inversos.....	7
2.1.3 - Requisitos não-funcionais.....	7
2.2 - Diagrama de Casos de Uso.....	9
2.3 - Descrição dos Casos de Uso.....	10
 3 - Diagramas de Arquitetura.....	32
3.1 - Diagrama de Classes.....	32
3.2 - Diagrama de Atividades.....	33
3.3 - Diagrama de Sequências.....	37
3.3 - Diagrama Entidade-Relacionamento.....	42
 4 - Atividades.....	44
4.1 - Modelagem Física do Banco de Dados.....	44
4.2 - Implementação do Banco de Dados.....	47
4.2 - Implementação do RF001	49



1 - Introdução

1.1 - Sobre o trabalho

Mediante fictício contrato firmado e baseado no estudo de caso (Case) apresentado pelos fictícios Analistas de Negócio, este trabalho apresenta a análise e a modelagem do novo Sistema de Controle e Venda de Passagens (SCVP) da empresa Vai&Volta Viagens (VVV), bem como o desenvolvimento do mesmo. Toda a análise e especificação de requisitos, detalhamentos, Diagrama de Caso de Uso e outros Diagramas como de Classe, Sequência e Atividades foram elaborados e anexados no referido trabalho com as devidas documentações.

1.2 - Sobre as atividades desenvolvidas

Foi realizada a análise e a especificação dos requisitos, o diagrama de casos de uso e a descrição dos casos de uso do estudo de caso, o diagrama de classes, o diagrama de atividades e o diagrama de sequência.

1.3 - Sobre a interação do grupo e a divisão das tarefas

Os componentes do grupo participaram de forma pró ativa com o envolvimento de todos em todas as etapas do trabalho.

1.4 - Descrição do Sistema

O novo sistema de informação desenvolvido para a empresa VVV possui novas funcionalidades para vendas e controle de passagens, tais como vendas online, atendendo melhor todas as necessidades da empresa, de seus clientes e funcionários. O sistema é mais amigável e possibilita a satisfação de todos os envolvidos devido ao excelente desempenho do sistema. Ele permite a comercialização de passagens nacionais e internacionais, com opções de



diferentes companhias e modais (aviões, trens, ônibus e navios). As reservas, pagamentos, aprovações de venda, emissão de bilhetes e muito mais, podem ser realizados em suas lojas físicas ou pelo site da empresa.

2 - Análise e Modelagem

2.1 - Análise e especificação dos requisitos

2.1.1 - Requisitos funcionais

RF001: Cadastro cliente.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O cliente poderá ser cadastrado no sistema web.

RF002: Login do cliente.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Para comprar passagens, o usuário deverá estar logado no sistema web.

RF003: Cadastro de funcionários.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Os funcionários poderão ser cadastrados no sistema.

RF004: Login de funcionários.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Os funcionários terão a permissão de logar no sistema.



RF005: Cadastro de Passageiros

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Os passageiros deverão ser cadastrados no sistema.

RF006: Registro de locais de origem e destino.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Um funcionário poderá cadastrar locais de origem e de destino das viagens.

RF007: Registro de modais.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Os sistemas de modais poderão ser registrados no sistema.

RF008: Registro de pontos de venda.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Os funcionários terão permissão de configurar pontos presenciais de venda.

RF009: Reservas nos pontos de vendas.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

A compra de passagens das companhias, utilizando diferentes tipos de modais, será feita pelo funcionário.

RF010: Efetuar Reserva Online.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável



O usuários poderão fazer reservas online por meio do sistema web.

RF011: Pagamento das passagens.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir o pagamento das passagens no cartão de débito e de crédito.

RF012: Aprovação de reservas presenciais.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Um funcionário específico terá permissão para aprovar as reservas feitas presencialmente.

RF013: Aprovação de reservas online.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Aprovação das reservas online mediante aprovação do gerente de negócios virtuais e confirmação da operadora de cartões.

RF014: Emissão de bilhetes de passagens (tickets).

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Após confirmação do pagamento da reserva, o sistema deverá emitir bilhetes de passagens.

RF015: Acesso autorizado de funcionário em ponto de venda.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável



O Sistema deve permitir o funcionário trabalhar em até dois pontos de venda em dias diferentes, autorizado pelo gerente.

2.1.2 - Requisitos Inversos

RI001: Aprovação de vendas de Cartões.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deverá ter uma função para receber a confirmação do pagamento da operadora de cartões, para aprovação da venda.

RI002: Registro e Transferência de Vendas Online

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Após a conclusão de uma venda online, o sistema deve automaticamente registrar os detalhes da venda no banco de dados da VVV e transferi-los para as companhias proprietárias dos modais correspondentes.

2.1.3 - Requisitos Não-Funcionais

RNF001: Uso do Framework Java Spring Boot 3.4.0

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve ser desenvolvido utilizando o framework Java Spring Boot 3.4.0.

RNF002: Compatibilidade Multiplataforma

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável



O sistema deve ser compatível com navegadores modernos (Edge, Chrome, Safari, além de Firefox e Mozilla) e dispositivos móveis.

RNF003: Uso de Banco de Dados PostgreSQL 17 na AWS RDS

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve utilizar o banco de dados PostgreSQL 17, hospedado em uma instância RDS da AWS para garantir escalabilidade, segurança e alta disponibilidade. O banco de dados deverá ser configurado de acordo com as melhores práticas de segurança e otimização oferecidas pela AWS.

RNF004: Seguir rigorosamente a LGPD.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deverá garantir ao usuário os direitos que estão previstos na LGPD (Lei Geral de Proteção de Dados Pessoais), após concordar com os termos de privacidade e segurança.



2.2 - Diagrama de Casos de Uso

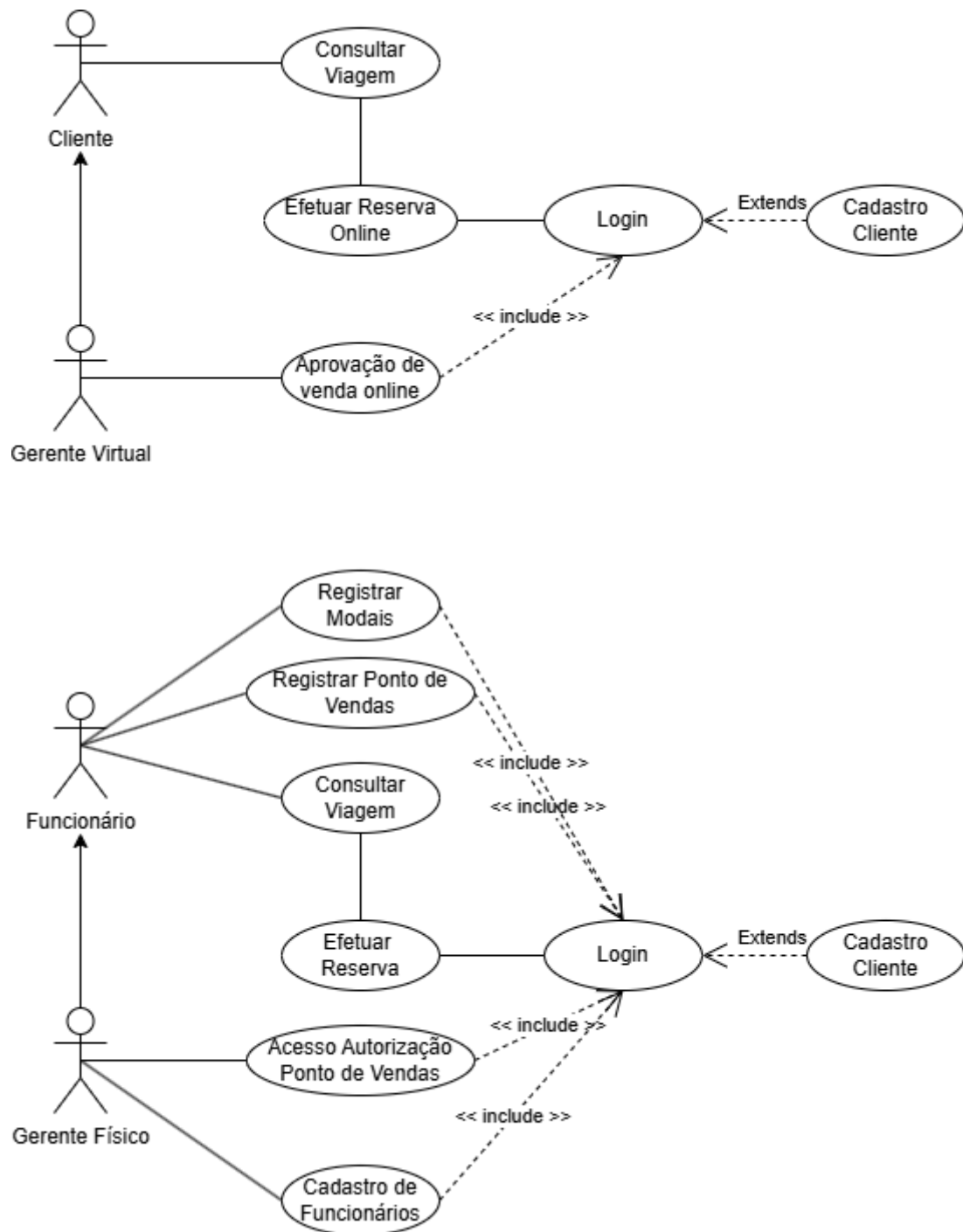


Figura 1: Diagrama de casos de uso



2.3 - Descrição dos Casos de Uso

2.3.1 - Caso de Uso 1: Cadastro cliente

Nome: Cadastro cliente.

Descrição: O sistema deve provisionar uma forma de cadastrar um cliente. Para cadastrar um cliente, são requeridos os seguintes dados: nome, data de nascimento, e-mail, senha, CPF, profissão, endereço e telefone.

Atores: Cliente

Pré-Condições: O cliente não está previamente cadastrado no sistema.

Pós-Condições: O cliente é cadastrado com sucesso no sistema.

Fluxo Básico:

- 1 - O Cliente acessa a tela de cadastro de clientes.
- 2 - O Cliente preenche os campos necessários para o cadastro.
- 3 - O Cliente clica no botão "Cadastre-se".
- 4 - O sistema registra o cliente com sucesso e atribui um código único a ele.

Fluxos Alternativos:

(A1) Alternativa ao passo 3 - Dados incorretos, cliente já cadastrado ou senha inválida

- 1.a O sistema detecta uma falha no cadastro, como cpf inválido.
- 1.b O sistema alerta ao cliente sobre a falha e fornece orientações.
- 1.c O sistema retoma ao passo 2.

Estrutura de dados:

(ED1) Cliente

- 1.1. Nome
- 1.2. Data de Nascimento
- 1.3. E-mail
- 1.4. Senha
- 1.5. CPF
- 1.6. Profissão



1.7. Endereço

1.8. Telefone

1.9. ID do Cliente

Regras de Negócio:

(RN1) Cada cliente tem informações como nome, endereço, CPF, telefone e profissão.

2.3.2 - Caso de Uso 2: Login do cliente

Nome: Login do cliente.

Descrição: O sistema deve provisionar uma forma do Cliente se autenticar e conseguir acessar funcionalidades e realizar ações.

Atores: Cliente

Pré-Condições: O cliente deve estar previamente cadastrado no sistema.

Pós-Condições: O cliente está autenticado no sistema e é redirecionado para a página inicial.

Fluxo Básico:

1. O cliente acessa a tela de login de clientes.
2. O cliente preenche os campos de login com suas credenciais cadastradas.
3. O cliente clica no botão "Login".
4. O cliente é redirecionado à página inicial.

Fluxos Alternativos:

(A1) Alternativa ao Passo 3 - Cliente não cadastrado ou dados incorretos

- 1.a O sistema detecta uma falha de autenticação.
- 1.b O sistema alerta o cliente sobre a falha de autenticação e fornece orientações sobre como proceder.
- 1.c O sistema retoma ao passo 2.

Estrutura de dados:

(ED1) Cliente

- 1.1. Nome



1.2. Data de Nascimento

1.3. E-mail

1.4. Senha

1.5. CPF

1.6. Profissão

1.7. Endereço

1.8. Telefone

1.9. ID do Cliente

Regras de Negócio:

2.3.3 - Caso de Uso 3: Cadastro de funcionários

Nome: Cadastro de funcionários.

Descrição: O sistema deve permitir o cadastro de novos funcionários, incluindo informações como nome, endereço e cargo. Os cargos disponíveis são Gerente, Gerente de Negócios Virtuais, Atendente, Registrador de Viagens e funcionários.

Atores: Registrador de Viagens e funcionários.

Pré-Condições: O funcionário não está previamente cadastrado no sistema.

Pós-Condições: O Registrador de Viagens e funcionários é registrado no sistema e redirecionado para a tela de cadastro, com todos os campos em branco.

Fluxo Básico:

1 - O Registrador de Viagens e funcionários acessa a tela de cadastro de funcionários.

2 - O Registrador de Viagens e funcionários preenche os campos necessários para o cadastro.

3 - O Registrador de Viagens e funcionários clica no botão "Cadastrar funcionário".

4 - O sistema registra o funcionário com sucesso e atribui um código único a ele.

**Fluxos Alternativos:**

(A1) Alternativa ao passo 3 - Dados incorretos ou funcionário já cadastrado.

1.a O sistema detecta uma falha no cadastro.

1.b O sistema alerta ao Registrador de Viagens e funcionários sobre a falha de autenticação e fornece orientações.

1.c O sistema retoma ao passo 2.

Estrutura de dados:**(ED1) Funcionário**

1.1. ID

1.2. Nome

1.3. Endereço

1.4. Cargo (enumerado: Gerente, Gerente de Negócios Virtuais, Atendente, Registrador de Viagens e funcionários)

Regras de Negócio:

(RN1) Os funcionários da empresa possuem código, nome, e endereço.

2.3.4 - Caso de Uso 4: Login de funcionários

Nome: Login de funcionários.

Descrição: O sistema deve provisionar uma forma do Funcionário se autenticar e conseguir acessar as funcionalidades e realizar ações restringidas a funcionários.

Atores: Funcionário

Pré-Condições: O funcionário deve estar previamente cadastrado no sistema.

Pós-Condições: O funcionário está autenticado no sistema, é redirecionado para a página inicial e apenas consegue acessar as funções, informações e ações restritas ao seu cargo.

Fluxo Básico:

1. O funcionário acessa a tela de login de funcionários.



2. O funcionário preenche os campos de login com suas credenciais cadastradas.
3. O funcionário clica no botão "Login".
4. O funcionário é redirecionado para a página inicial de funcionários.

Fluxos Alternativos:

(A1) Alternativa ao Passo 3 - Funcionário não cadastrado ou dados incorretos

- 1.a O sistema detecta uma falha de autenticação.
- 1.b O sistema alerta o funcionário sobre a falha de autenticação e fornece orientações sobre como proceder.
- 1.c O sistema retoma ao passo 2.

Estrutura de dados:

(ED1) Funcionário

- 1.1. ID
- 1.2. Nome
- 1.3. Endereço
- 1.4. Cargo (enumerado: Gerente, Gerente de Negócios Virtuais, Atendente, Registrador de Viagens e funcionários)

Regras de Negócio:

2.3.5 - Caso de Uso 5: Cadastro de Cliente por funcionário

Nome: Cadastro de Cliente por funcionário.

Descrição: O sistema deve permitir que o funcionário da loja física cadastre um cliente.

Atores: Funcionário.

Pré-Condições:

O funcionário deve estar previamente logado no sistema.

O passageiro não está previamente cadastrado no sistema.

Pós-Condições: O cliente é cadastrado com sucesso no sistema.

Fluxo Básico:

- 1- O funcionário entra na tela de cadastro



2- O funcionário preenche os dados da tela de cadastro

3- O funcionário clica em enviar dados.

Fluxos Alternativos:

(A1) Alternativa ao passo 3 - Dados incorretos.

1.a O sistema detecta uma falha no cadastro.

1.b O sistema alerta ao responsável sobre a falha de autenticação e fornece orientações.

1.c O sistema retoma ao passo 2.

Estrutura de dados:

(ED1) Cliente

1.1. Nome

1.2. Idade

1.3. E-mail

1.4. Senha

1.5. CPF

1.6. Profissão

1.7. Endereço

1.8. Telefone

1.9. ID do Cliente

Regras de Negócio:

(RN1) Cada cliente tem informações como nome, endereço, CPF, telefone e profissão.

2.3.6 - Caso de Uso 6: Registro de locais de origem e destino

Nome: Registro de locais de origem e destino.

Descrição: O sistema deve provisionar que o funcionário possa ser capaz de registrar os locais de origem e destino, cadastrando cidades, aeroportos, estações e portos

Atores: Funcionário.

Pré-Condições: O Funcionário deve estar logado para realizar o registro.

Pós-Condições: O registro das cidades, aeroportos, estações e portos



é feito no sistema, incluindo os devidos códigos e identificadores.

Fluxo Básico:

- 1 - O funcionário acessa o sistema.
- 2 - O funcionário seleciona a opção de registrar local de origem ou destino..
- 3 - O funcionário especifica a cidade, aeroporto, estação e porto de origem e de destino.
- 7 - O funcionário confirma as informações.
- 8 - O sistema registra as informações com sucesso.

Fluxos Alternativos:

Estrutura de dados:

(ED1) Cidade

- 1.1. Código (único)
- 1.2. ID.

(ED2) Aeroporto

- 1.1. Código (único)

Regras de Negócio:

(RN1) As passagens aéreas devem incluir o código do aeroporto.

2.3.7 - Caso de Uso 7: Registro de modais.

Nome: Registro de modais.

Descrição: O sistema deve permitir que os funcionários registrem modais no sistema. Para registrar um modal, são requeridos os seguintes dados: tipo, código, modelo, capacidade, ano de fabricação, nome da empresa associada, manutenção.

Atores: Funcionário

Pré-Condições: O funcionário tem acesso autorizado ao sistema de registro de modais.

Pós-Condições: O modal é registrado com sucesso no sistema, com todas as informações necessárias.

Fluxo Básico:

- 1 - O funcionário acessa o sistema de registro de modais.



2 - O funcionário seleciona a opção para cadastrar um novo modal.

3 - O funcionário insere os seguintes dados do modal: tipo, código, modelo, capacidade, ano de fabricação, nome da empresa associada e manutenção.

4 - O sistema registra o modal com sucesso, confirma a operação ao funcionário e atribui um código único a ele.

Fluxos Alternativos:

(A1) Alternativa ao passo 3 - Dados inseridos não válidos ou campos obrigatórios em branco.

1.a O sistema detecta que algum dado inserido é inválido ou existe um campo obrigatório em branco.

1.b O sistema exibe uma mensagem de erro.

1.c O sistema solicita que o funcionário forneça as informações em falta.

1.d O sistema retoma ao passo 3.

Estrutura de dados:

(ED1) Modal

1.1. Tipo

1.2. Código (único)

1.3. Modelo

1.4. Capacidade

1.5. Ano de Fabricação

1.6. Nome da Empresa

1.7. Manutenção

Regras de Negócio:

(RN1) Os modais passam por manutenção periódica, durante a qual ficam não-operacionais

(RN2) A manutenção é realizada pela empresa proprietária dos modais, que informa quando o modal entrará em manutenção.

2.3.8 - Caso de Uso 8: Registro de pontos de venda.

Nome: Registro de pontos de venda.



Descrição: O sistema deve permitir que os funcionários registrem pontos de vendas no sistema. Para registrar um ponto de venda, são requeridos os seguintes dados: código, nome, CNPJ, endereço, telefone e um funcionário atribuído como Gerente.

Atores: Funcionário

Pré-Condições: O funcionário tem acesso autorizado ao sistema de registro de pontos de venda.

O ponto de venda não está previamente cadastrado no sistema.

Pós-Condições: O ponto de venda é registrado com sucesso no sistema, com todas as informações necessárias.

Fluxo Básico:

- 1 - O funcionário acessa o sistema de registro de pontos de venda.
- 2 - O funcionário seleciona a opção para cadastrar um novo ponto de venda.
- 3 - O funcionário insere os seguintes dados do ponto de venda: código, nome, CNPJ, endereço e telefone.
- 4 - O funcionário seleciona o funcionário que será atribuído como gerente do ponto de venda.
- 5 - O sistema registra o ponto de venda com sucesso, confirma a operação ao funcionário e atribui um código único a ele.

Fluxos Alternativos:

(A1) Alternativa ao passo 3 - Dados inseridos não válidos ou campos obrigatórios em branco

1.a O sistema detecta que algum dado inserido é inválido ou existe um campo obrigatório em branco.

1.b O sistema exibe uma mensagem de erro.

1.c O sistema solicita que o funcionário forneça as informações em falta.

1.d O sistema retoma ao passo 3.

(A2) Alternativa ao passo 3 - CNPJ já cadastrado no sistema

2.a O sistema detecta que o CNPJ já está cadastrado no sistema.

2.b O sistema exibe uma mensagem de erro.

2.c O sistema solicita que o funcionário forneça um CNPJ diferente.



2.d O sistema retoma ao passo 3.

(A3) Alternativa ao passo 4 - Funcionário não disponível ou válido

3.a O sistema detecta que o funcionário não está disponível ou não é válido.

3.b O sistema exibe uma mensagem de erro.

3.c O sistema solicita que outro funcionário seja escolhido como gerente.

3.d O sistema retoma ao passo 4.

Estrutura de dados:

(ED1) Ponto de Venda

1.1. Código (único)

1.2. Nome

1.3. CNPJ

1.4. Endereço

1.5. Telefone

1.6. Gerente (funcionário atribuído)

Regras de Negócio:

(RN1) Cada ponto de venda possui um funcionário que é um gerente, e os funcionários podem trabalhar em até dois postos de venda e em dias diferentes, mediante autorização do gerente.

2.3.9 - Caso de Uso 9: Reservas nos pontos de vendas.

Nome: Reservas nos pontos de vendas.

Descrição: O sistema deve permitir ao funcionário logado localmente realizar reservas de clientes já cadastrados no sistema, informado os dados de origem, destino e se tem escala ou não.

Atores: Funcionário

Pré-Condições:

O Funcionário deve estar logado para realizar a reserva.

O cliente deve estar cadastrado.

Pós-Condições:

O pagamento da reserva deve ser realizado pelo cliente.



O pagamento da reserva deve ser aprovado pelo banco.

O funcionário deve confirmar o pagamento

O funcionário deve emitir o ticket.

A reserva deve ser associada ao modal, para evitar overbooking respeitando a capacidade do modal.

As reservas devem ser consultadas e armazenadas no banco de dados da VVV e transferidas para a companhia dos modais.

Fluxo Básico:

- 1- O funcionário entra na tela de reserva
- 2- O funcionário escolhe um roteiro de viagem
- 3 - O funcionário localiza o cliente com os dados do cliente
- 4 - O funcionário insere os dados da reserva
- 5- O funcionário clica em fazer reserva
- 6 - O cliente realiza o pagamento

Fluxos Alternativos:

(A1) Alternativa ao passo 3 - Cliente não cadastrado

Caso o cliente não esteja cadastrado, o funcionário cadastra o cliente no sistema utilizando a tela de cadastro.

Estrutura de dados:

(ED1) Reserva

- 1.1. Código da reserva
- 1.2. Data da reserva
- 1.3. Status da reserva
- 1.4. Valor da reserva
- 1.5. Nome do cliente
- 1.6. Código do cliente
- 1.7. Origem
- 1.8. Destino
- 1.9. Tipo de passagem
- 1.10. Hora de partida
- 1.11. Hora de chegada



1.12. Tempo de viagem

1.13. Localizador

Regras de Negócio:

(RN1) Cada Reserva deve ser de um único passageiro.

2.3.10 - Caso de Uso 10: Efetuar reserva online.

Nome: Efetuar reserva online.

Descrição: O sistema deve permitir ao cliente logado realizar reservas através do site da VVV, informado os dados de origem, destino e se tem escala ou não.

Atores: Cliente

Pré-Condições: O cliente deve estar logado para realizar a reserva.

Pós-Condições: O pagamento da reserva deve ser realizado pelo cliente e aprovado pelo gerente de negócios virtuais.

A reserva deve ser associada ao modal, para evitar overbooking respeitando a capacidade do modal.

As reservas devem ser consultadas e armazenadas no banco de dados da VVV e transferidas para a companhia dos modais.

Fluxo Básico:

- 1- O cliente entra na tela de reserva
- 2- O cliente insere os dados da reserva, local de origem e destino
- 3- O cliente escolhe um roteiro de viagem
- 4- O cliente clica em fazer reserva

Fluxos Alternativos:

Estrutura de dados:

(ED1) Reserva

- 1.1. Código da reserva
- 1.2. Data da reserva
- 1.3. Status da reserva
- 1.4. Valor da reserva



- 1.5. Nome do cliente
- 1.6. Código do cliente
- 1.7. Origem
- 1.8. Destino
- 1.9. Tipo de passagem
- 1.10. Hora de partida
- 1.11. Hora de chegada
- 1.12. Tempo de viagem
- 1.13. Localizador

Regras de Negócio:

(RN1) Cada Reserva deve ser de um único passageiro.

2.3.11 - Caso de Uso 11: Pagamento das passagens.

Nome: Pagamento das passagens.

Descrição: O sistema deve permitir ao cliente logado realizar o pagamento das passagens no cartão de crédito e de débito ou pix.

Atores: Cliente

Pré-Condições:

O cliente deve estar logado para realizar o pagamento.

O cliente deve estar cadastrado.

Pós-Condições:

O pagamento da reserva deve ser realizado pelo cliente.

O pagamento da reserva deve ser aprovado pelo banco.

O sistema deve confirmar o pagamento com a operadora de cartões.

O sistema deve emitir o ticket.

Fluxo Básico:

- 1- O cliente abrirá a tela de compra de passagens.
- 2- O cliente escolherá dentre as opções de passagens disponíveis.
- 3- O cliente vai ser direcionado a página de pagamento.



4- O cliente cadastra o cartão de débito ou/e crédito caso seja cartão e caso seja pix, realizada o pagamento no pix.

5- O cliente clica em efetuar o pagamento.

6- O sistema deve identificar o pagamento da passagem junto a confirmação da operadora de cartões.

Fluxos Alternativos:

(A1) Alternativa ao passo 6 - Pagamento não aprovado

Caso o pagamento não tenha sido aprovado, o cliente retorna ao passo 3.

Estrutura de dados:

(ED1) Passagem online

1.1. Código da passagem

1.2. Data da passagem

1.3. Status da passagem

1.4. Valor da passagem

1.5. Nome do cliente

1.6. Código do cliente

1.7. Origem

1.8. Destino

1.9. Tipo de passagem

1.10. Hora de partida

1.11. Hora de chegada

1.12. Tempo de viagem

1.13. Localizador

Regras de Negócio:

(RN1) Os meios de pagamento das reservas são feitos com cartão em débito automático ou de crédito.

(RN2) Os pagamentos via cartão de crédito podem ser parcelados sem juros em até 4 vezes, acima de 4 vezes serão cobrados juros de 5% sobre o valor da passagem.

2.3.12 - Caso de Uso 12: Aprovação de reservas presenciais.



Nome: Aprovação de reservas presenciais.

Descrição: O sistema deve ser capaz de possibilitar um funcionário de aprovar uma reserva.

Atores: Funcionário

Pré-Condições:

A reserva deve existir.

O funcionário deve estar logado.

O pagamento deve estar aprovado.

Pós-Condições:

Fluxo Básico:

- 1- O Funcionário deve abrir a tela de aprovações de reserva
- 2- O Funcionário deve localizar a reserva através dos dados do Cliente da reserva.
- 3- O Funcionário deve verificar se o pagamento está aprovado.
- 4- O Funcionário deve aprovar a reserva clicando em aprovar.

Fluxos Alternativos:

(A1) Alternativa ao passo 2 - Reserva não localizada.

Caso o Atendente não localize a reserva, o fluxo é interrompido e é necessário fazer uma nova reserva.

(A2) Alternativa ao passo 3 - Pagamento não aprovado.

Caso o pagamento não tenha sido aprovado, a reserva deve ser cancelada.

Estrutura de dados:

(ED1) Reserva

- 1.1. Código da reserva
- 1.2. Data da reserva
- 1.3. Status da reserva
- 1.4. Valor da reserva
- 1.5. Nome do cliente
- 1.6. Código do cliente
- 1.7. Origem



1.8. Destino

1.9. Tipo de passagem

1.10. Hora de partida

1.11. Hora de chegada

1.12. Tempo de viagem

1.13. Localizador

Regras de Negócio:

(RN1) Cada Reserva deve ser de um único passageiro.

2.3.13 - Caso de Uso 13: Aprovação de reservas online.

Nome: Aprovação de reservas online.

Descrição: O sistema deve ser capaz de aprovar uma reserva online.

Atores: Cliente e Gerente de negócios virtuais.

Pré-Condições:

A reserva deve existir.

O Gerente de negócios virtuais deve estar logado.

O pagamento deve estar aprovado.

Pós-Condições:

O pagamento da reserva deve ser realizado pelo cliente.

O pagamento da reserva deve ser aprovado pelo banco.

O sistema deve confirmar o pagamento com a operadora de cartões.

O sistema deve emitir o ticket.

Fluxo Básico:

1- O cliente abrirá a tela de reservas.

2- O cliente escolherá dentre as opções de reserva disponíveis.

3- O cliente vai ser direcionado a página de pagamento.

4- O cliente cadastra o cartão de débito ou/e crédito.

5- O cliente clica em efetuar o pagamento.

6- O gerente de negócios virtuais deve identificar o pagamento da passagem junto a confirmação da operadora de cartões.

Fluxos Alternativos:



(A1) Alternativa ao passo 6 - Pagamento não aprovado

Caso o pagamento não tenha sido aprovado, o cliente será retornado ao passo 3.

Estrutura de dados:

(ED1) Reserva online

- 1.1. Código da reserva
- 1.2. Data da reserva
- 1.3. Status da reserva
- 1.4. Valor da reserva
- 1.5. Nome do cliente
- 1.6. Código do cliente
- 1.7. Origem
- 1.8. Destino
- 1.9. Tipo de passagem
- 1.10. Hora de partida
- 1.11. Hora de chegada
- 1.12. Tempo de viagem
- 1.13. Localizador

Regras de Negócio:

(RN1) Cada Reserva deve ser de um único passageiro.

(RN2) Antes da venda, é necessário fazer uma reserva que só é confirmada mediante o pagamento e aprovação do gerente de negócios virtuais.

2.3.14 - Caso de Uso 14: Emissão de bilhetes de passagens (tickets).

Nome: Emissão de bilhetes de passagens (tickets).

Descrição: O sistema deve ser capaz de realizar a emissão de bilhetes de passagens quando o pagamento tiver sido aprovado.

Atores: Cliente, funcionário

Pré-Condições: A reserva e o pagamento devem ter sido confirmados.

Pós-Condições:

**Fluxo Básico:**

- 1- Na tela de confirmação de reserva deve escolher a opção “emitir ticket”.
- 2- Após o pagamento ter sido aprovado, o sistema deve emitir o ticket.

Fluxos Alternativos:

(A1) Alternativa ao passo 2 - Pagamento não aprovado

Caso o pagamento não tenha sido aprovado, o cliente ou funcionário serão informados e redirecionados para o passo de pagamento..

Estrutura de dados:**(ED1) Tickets**

- 1.1. Número do ticket
- 1.2. Tipo de passagem
- 1.3. Hora de partida
- 1.4. Hora de chegada
- 1.5. Tempo de viagem
- 1.6. Localizador

Regras de Negócio:.**2.3.15 - Caso de Uso 15: Autorizar acesso de funcionário em ponto de venda.**

Nome: Acesso autorizado de funcionário em ponto de venda.

Descrição: O Sistema deve permitir o funcionário trabalhar em até dois pontos de venda, em dias diferentes, com a devida autorização do gerente de cada ponto de venda.

Atores: Gerente do ponto de venda

Pré-Condições:

O Funcionário deve ter solicitado o trabalho nos devidos pontos de venda.

Os pontos de venda devem estar cadastrados.

O Funcionário deve estar cadastrado.

O gerente deve estar cadastrado e logado.

Pós-Condições:



O Funcionário será aprovado de trabalhar em dois pontos de venda.

Fluxo Básico:

- 1- O Gerente de Ponto de Venda abre a tela de cadastro de funcionários
- 2- O Gerente de Ponto de Venda localiza o registro do Funcionário
- 3- O Gerente de Ponto de Venda seleciona os pontos de venda e clica em "Aprovar solicitação de trabalhar em dois pontos de venda".

Fluxos Alternativos:

(A1) Alternativa ao passo 3 - Funcionário não aprovado.

Caso o funcionário não seja aprovado, o sistema retornará ao passo 2.

Estrutura de dados:

(ED1) Funcionário

- 1.1. ID
- 1.2. Nome
- 1.3. Endereço
- 1.4. Cargo (enumerado: Gerente, Gerente de Negócios Virtuais, Atendente, Registrador de Viagens, Registrador de Clientes)
- 1.5. Pontos de venda

Regras de Negócio:

(RN1) Os funcionários podem trabalhar em até dois postos de venda e em dias diferentes, mediante autorização do gerente.

2.3.16 - Caso de Uso 16: Aprovação de vendas de Cartões.

Nome: Aprovação de vendas de Cartões.

Descrição: O sistema deve ter uma função para receber a confirmação do pagamento da operadora de cartões, permitindo a aprovação da venda realizada.

Atores: Sistema de Vendas e Operadora de Cartões

Pré-Condições: O cliente deve ter efetuado uma compra utilizando um cartão de crédito ou débito.

Pós-Condições: A venda é aprovada e o pedido é processado.

Fluxo Básico:



- 1 - O cliente efetua a compra utilizando um cartão de crédito ou débito.
- 2 - O sistema de vendas envia os detalhes da transação para a operadora de cartões.
- 3 - A operadora de cartões processa o pagamento.
- 4 - A operadora de cartões envia uma resposta de confirmação para o sistema de vendas.
- 5 - O sistema de vendas aprova a venda e continua com o processamento do pedido.

Fluxos Alternativos:

(A1) Alternativa ao passo 4 - Pagamento não aprovado.

1.a - A operadora de cartões não aprova o pagamento.

1.b - O sistema de vendas exibe uma mensagem de erro e solicita que o cliente tente novamente ou utilize outro método de pagamento

1.c O sistema retoma ao passo 1.

Estrutura de dados:

Regras de Negócio:

2.3.17 - Caso de Uso 17: Registro de viagem

Nome: Registro de viagem.

Descrição: O sistema deve provisionar que o funcionário possa ser capaz de registrar e manipular roteiros de viagens que contém um local de origem e destino, modal de transporte utilizado, escalas e o número de reservas disponíveis.

Atores: Funcionário.

Pré-Condições: Existência de pelo menos uma cidade de origem e uma cidade de destino cadastradas no sistema.

Disponibilidade dos modais de transporte necessários para a realização da viagem.

Possibilidade de reserva de passagens para a viagem.



Pós-Condições: O registro completo da viagem é feito no sistema, incluindo informações sobre o modal de transporte utilizado, cidades de origem e destino, e reservas associadas.

Fluxo Básico:

- 1 - O funcionário acessa o sistema de registro de viagens.
- 2 - O funcionário seleciona a opção de registrar uma nova viagem.
- 3 - O funcionário especifica a cidade de origem e a cidade de destino da viagem.
- 4 - O funcionário escolhe o modal de transporte a ser utilizado para a viagem (por exemplo, avião, trem, ônibus).
- 5 - Se a viagem envolver escalas, o funcionário adiciona as cidades de escala conforme necessário.
- 6 - O funcionário indica o número máximo de reservas que estão associadas a essa viagem.
- 7 - O funcionário confirma as informações da viagem.
- 8 - O sistema registra a viagem com sucesso e atribui um código único a ela.

Fluxos Alternativos:

(A1) Alternativa ao passo 3 - Cidade de origem ou destino não cadastrada no sistema

- 1.a O sistema detecta que a cidade de origem ou destino não está cadastrada no sistema.
- 1.b O sistema exibe uma mensagem de erro.
- 1.c O sistema solicita que o funcionário insira uma cidade válida.
- 1.d O sistema retoma ao passo 3.

(A2) Alternativa ao passo 4 - Modal não disponível

- 2.a O sistema detecta que o modal não está disponível.
- 2.b O sistema exibe uma mensagem de erro.
- 2.c O sistema solicita que o funcionário insira um modal disponível.
- 2.d O sistema retoma ao passo 4.

Estrutura de dados:

(ED1) Viagem



- 1.1. Código da viagem (único)
- 1.2. Cidade de origem
- 1.3. Cidade de destino
- 1.4. Modal de transporte
- 1.5. Cidades de escala (se houver)
- 1.6. Número de reservas associadas.

Regras de Negócio:

(RN1) Cada viagem possui um roteiro que inclui uma cidade de origem e uma cidade de destino, podendo ser direta ou com escalas.

(RN2) As reservas podem ser feitas apenas por meio físico nas lojas da empresa ou online por cartão de crédito ou débito.

(RN3) Uma viagem pode ser realizada usando um ou mais tipos de modais (aviões, trens, ônibus e navios).

(RN4) As passagens aéreas devem incluir o código do aeroporto.

(RN5) Os modais passam por manutenção periódica, durante a qual ficam não-operacionais



3 - Diagramas de Arquitetura

3.1 - Diagrama de Classes

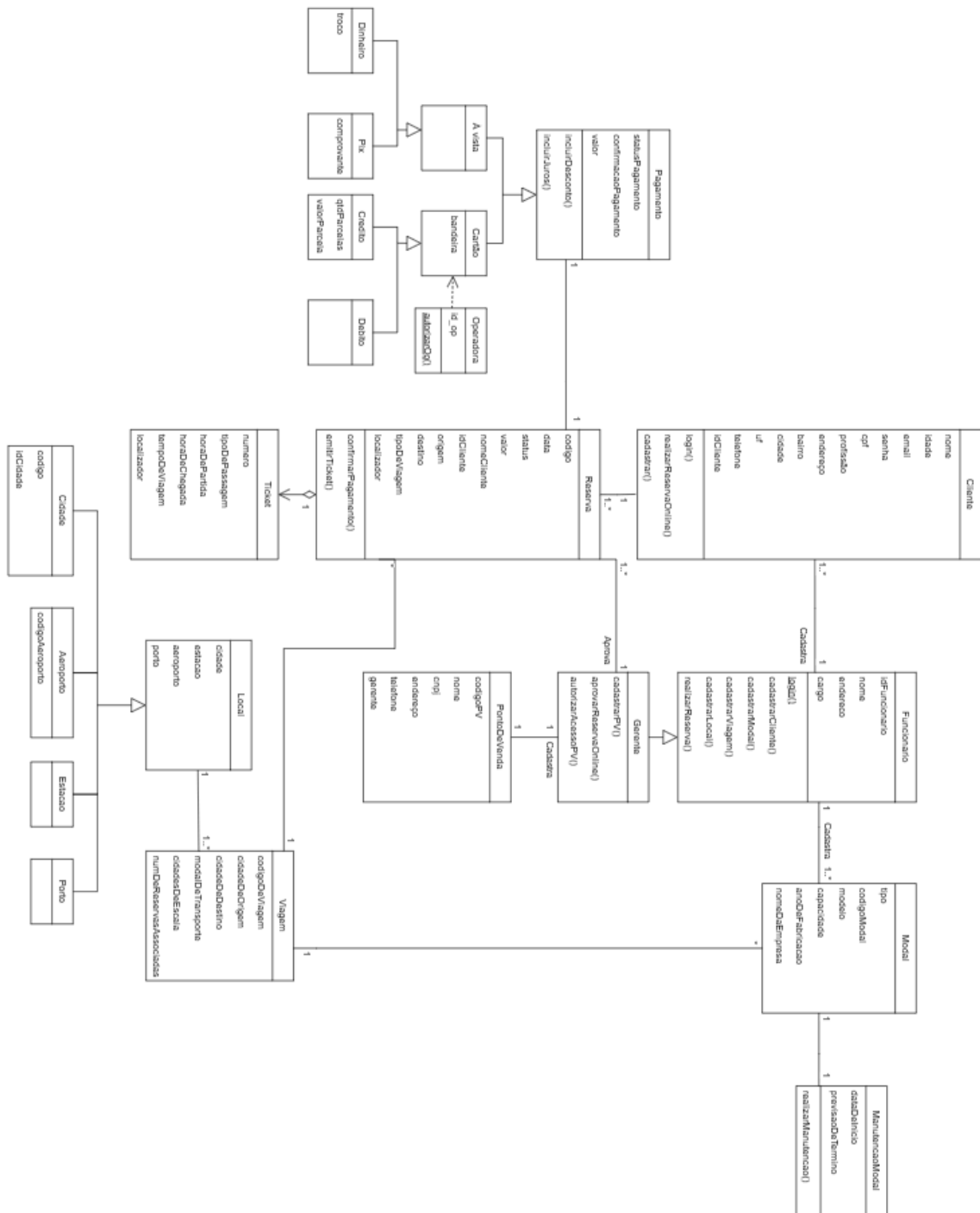
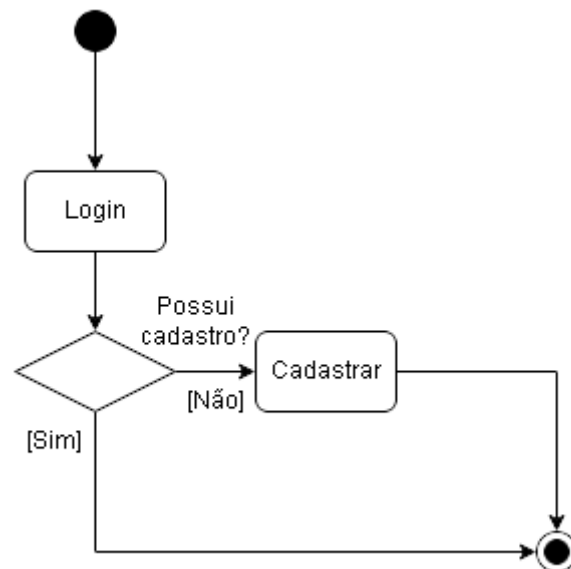


Figura 2: Diagrama de Classes

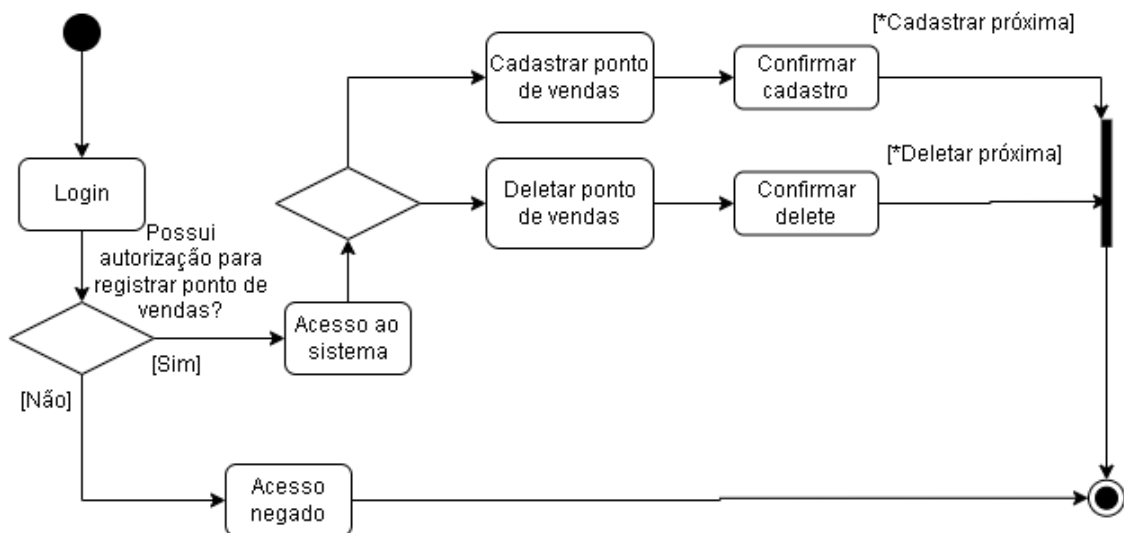


3.2 - Diagrama de Atividades

3.2.1 - Login de usuário/funcionário

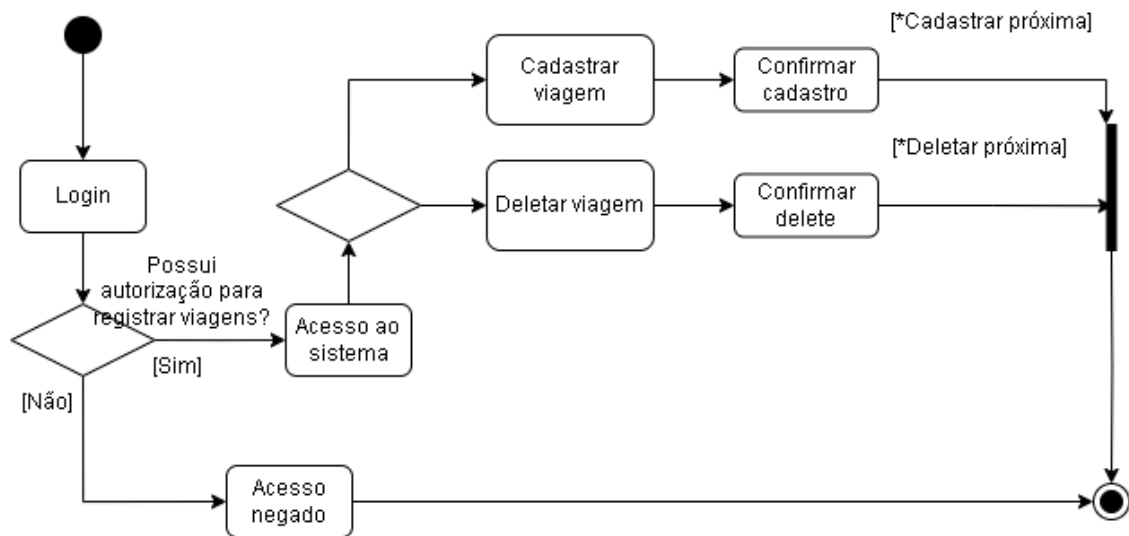


3.2.2 - Registrar/Deletar pontos de venda

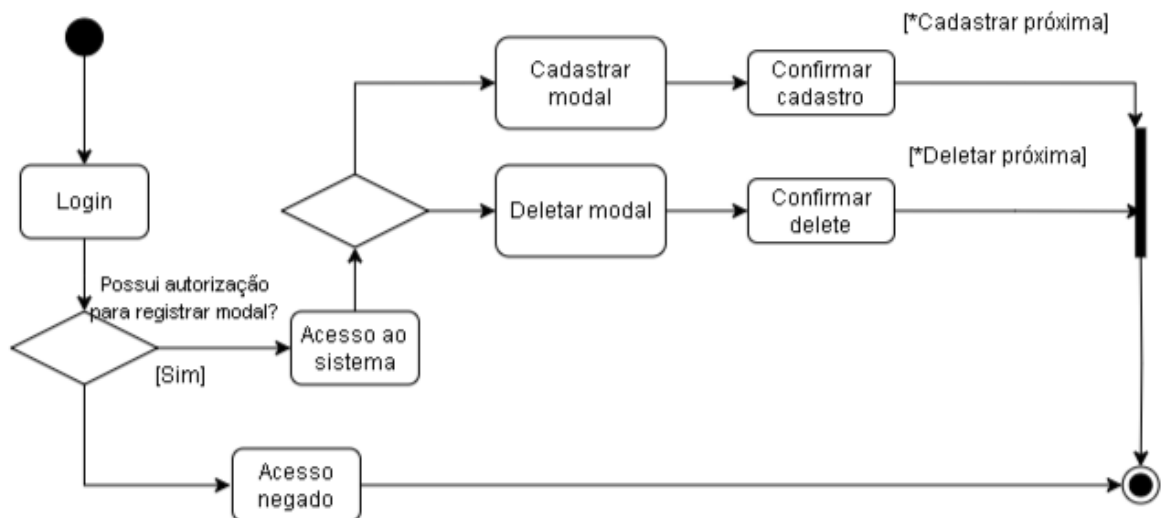




3.2.5 - Registrar/Deletar viagens

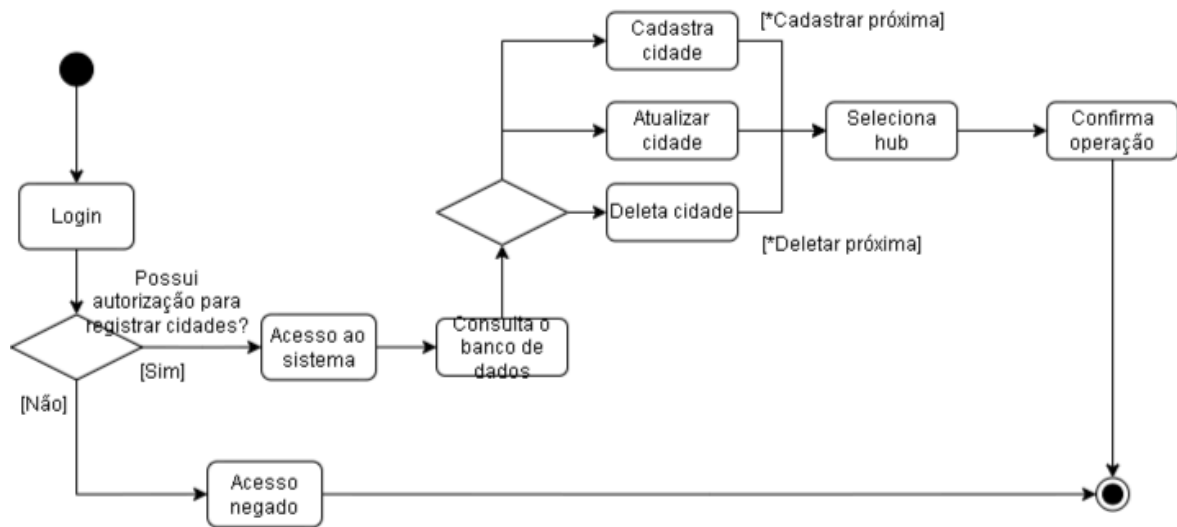


3.2.6 - Registrar/Deletar modal

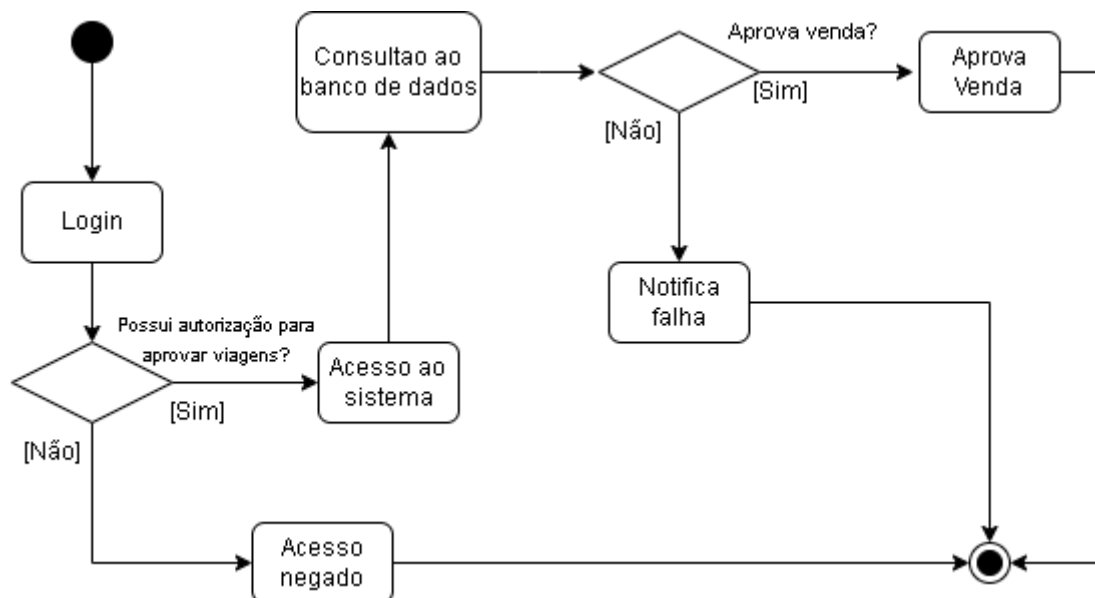




3.2.7 - Registrar/Deletar/Atualizar locais



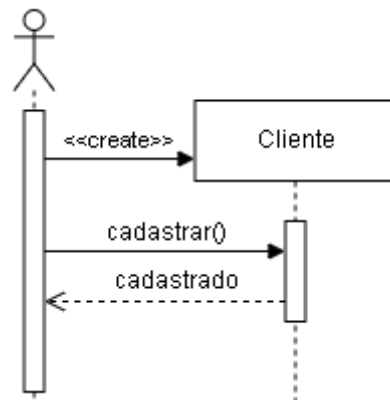
3.2.8 - Aprovar vendas online



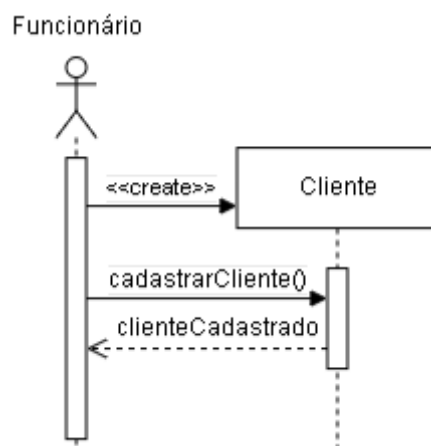


3.3 - Diagrama de Sequência

3.3.1 - Cliente se cadastra

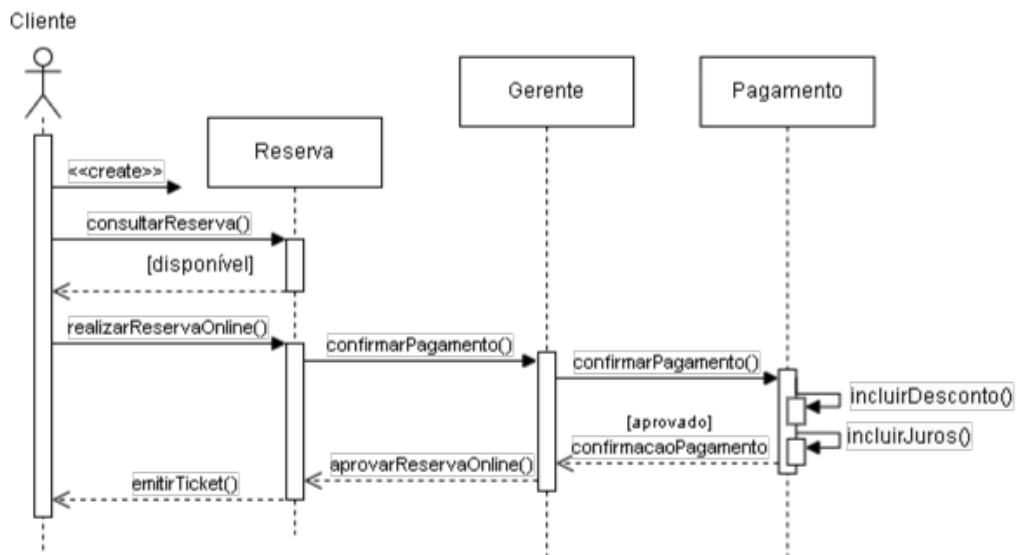


3.3.2 - Funcionário cadastra cliente

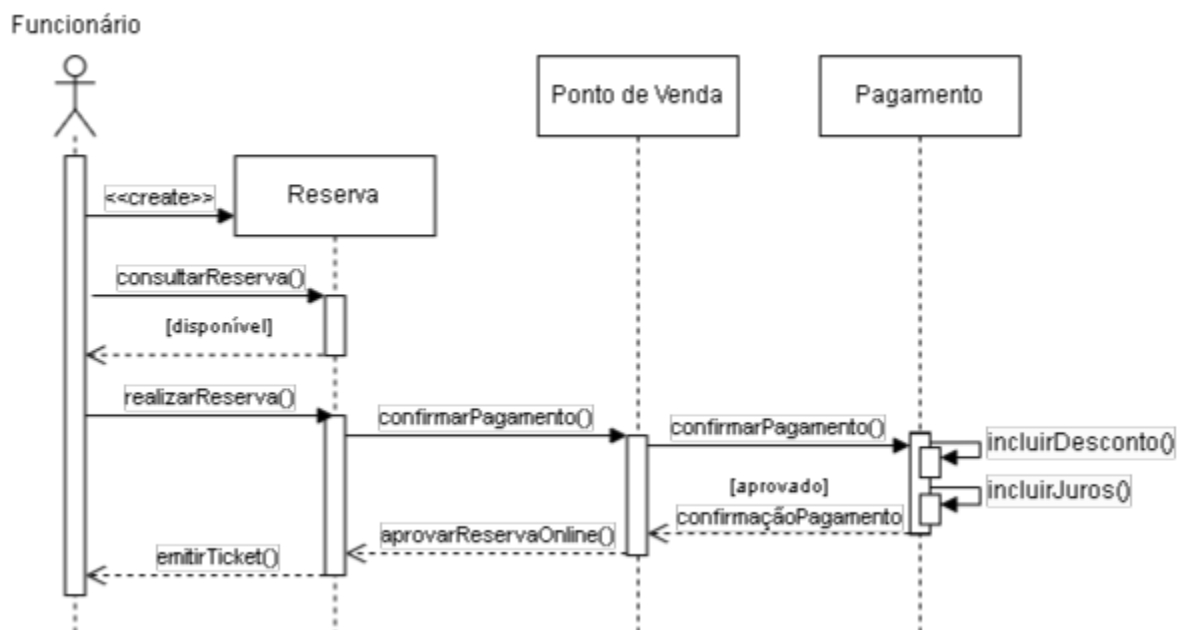




3.3.3 - Efetuar reserva online

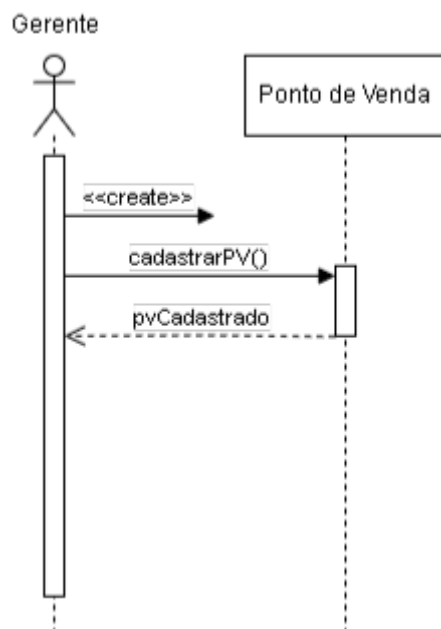


3.3.4 - Funcionário realiza reserva no Ponto de Venda

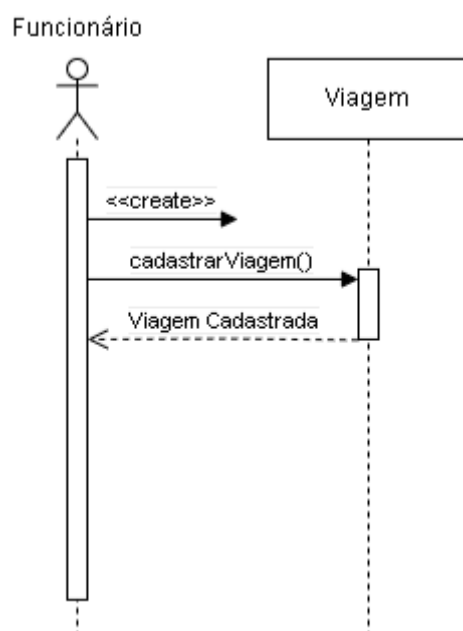




3.3.5 - Gerente cadastra ponto de venda

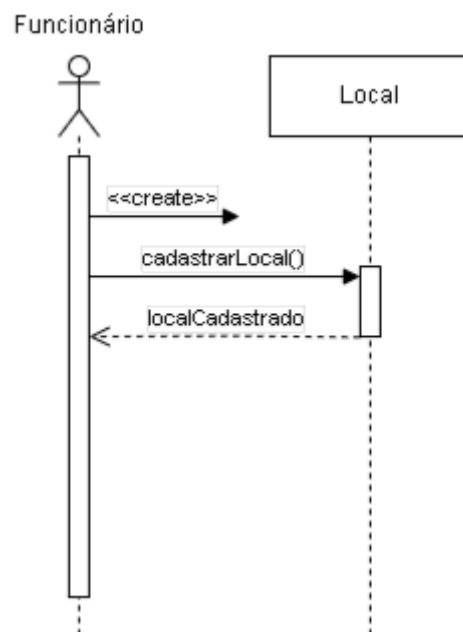


3.3.6 - Funcionário cadastra viagem

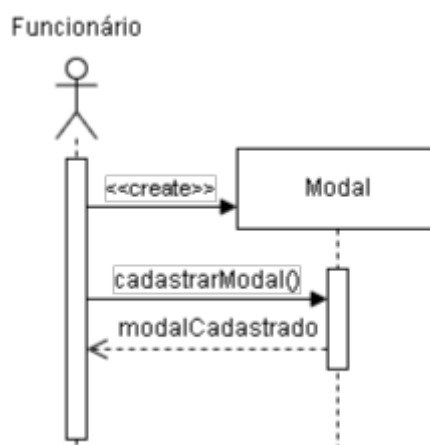




3.3.7 - Funcionário cadastra local

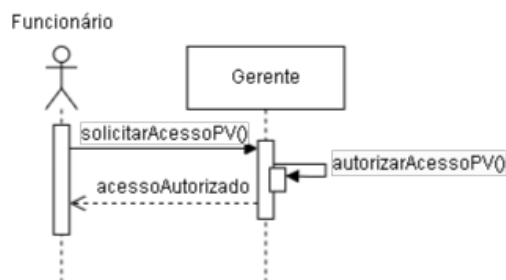


3.3.8 - Funcionário cadastra modal

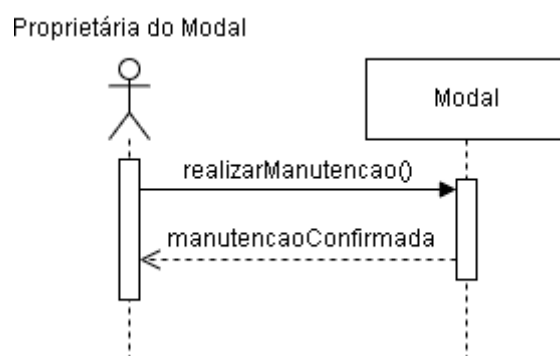




3.3.9 - Autorização do funcionário no Ponto de Vendas



3.3.10 - Manutenção do Modal





3.4 - Diagrama Entidade-Relacionamento

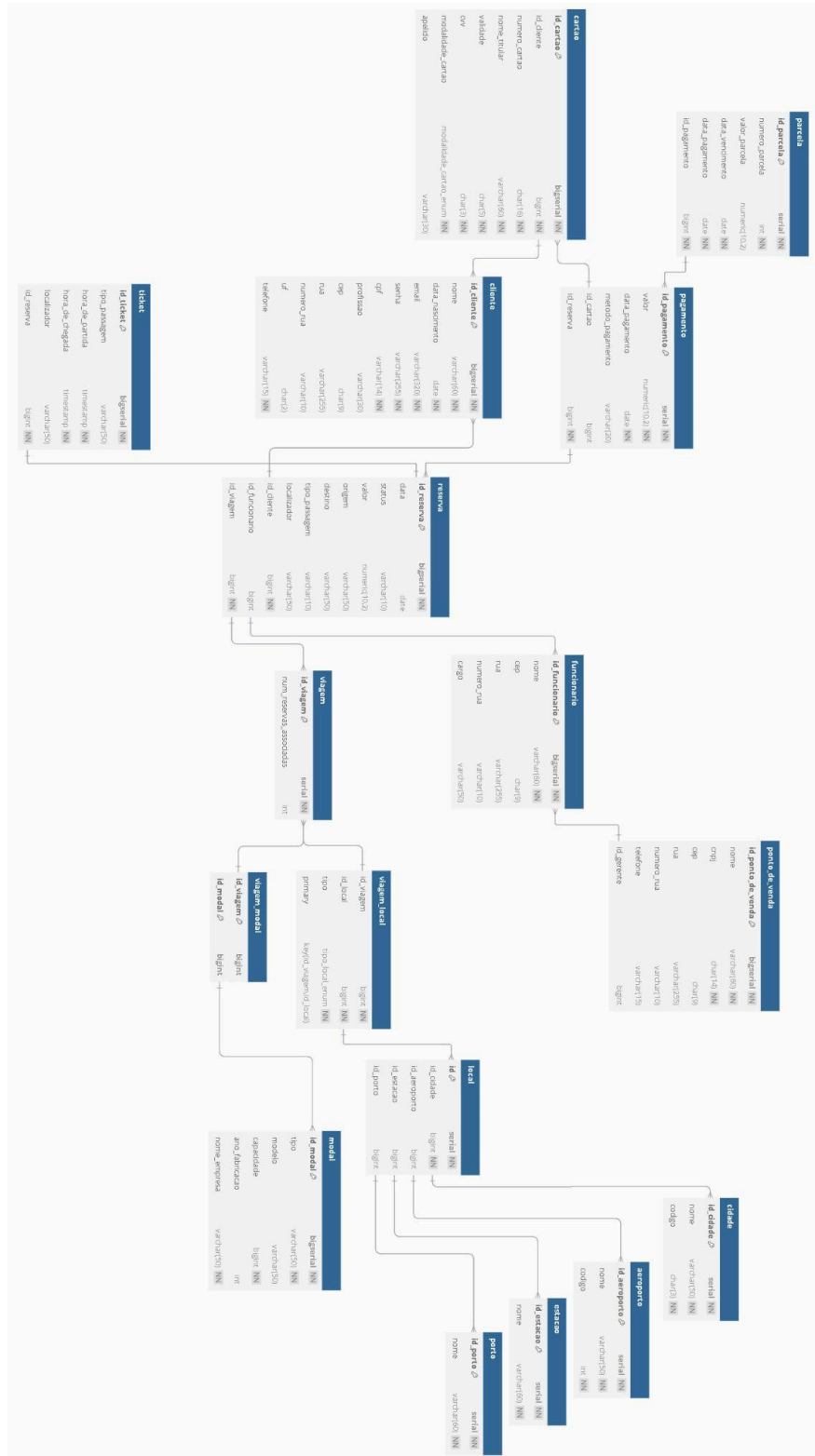


Figura 3: Diagrama Entidade-Relacionamento



4 - Atividades

4.1 - Modelagem Física do Banco de Dados

Abaixo está apresentado o script SQL responsável pela criação do modelo físico do banco de dados feito a partir do modelo Diagrama Entidade-Relacionamento. Esse código inclui as tabelas, atributos, tipos de dados e relacionamentos necessários para a implementação no SGBD escolhido.

```
-- Tabela Cliente
CREATE TABLE cliente (
    id_cliente BIGSERIAL PRIMARY KEY,
    nome VARCHAR(60) NOT NULL,
    data_nascimento DATE NOT NULL,
    email VARCHAR(320) NOT NULL,
    senha VARCHAR(255) NOT NULL,
    cpf VARCHAR(14) NOT NULL,
    profissao VARCHAR(30),
    cep CHAR(9),
    rua VARCHAR(255),
    numero_rua VARCHAR(10),
    uf CHAR(2),
    telefone VARCHAR(15) NOT NULL
);

CREATE TYPE modalidade_cartao_enum AS ENUM ('credito',
'debito');

-- Tabela Cartão
CREATE TABLE cartao (
    id_cartao BIGSERIAL PRIMARY KEY,
    id_cliente INT NOT NULL REFERENCES
cliente(id_cliente),
    numero_cartao CHAR(16) NOT NULL,
    nome_titular VARCHAR(60) NOT NULL,
    validade CHAR(5) NOT NULL,
```



```
        cvv CHAR(3) NOT NULL,
        modalidade_cartao modalidade_cartao_enum NOT NULL,
        apelido VARCHAR(30)
    );

    CREATE TYPE metodo_pagamento_enum AS ENUM ('pix',
'boleto', 'cartao');

-- Tabela Funcionario
CREATE TABLE funcionario (
    id_funcionario BIGSERIAL PRIMARY KEY,
    nome VARCHAR(60) NOT NULL,
    cep CHAR(9),
    rua VARCHAR(255),
    numero_rua VARCHAR(10),
    cargo VARCHAR(50)
);

-- Tabela Ponto de Venda
CREATE TABLE ponto_de_venda (
    id_ponto_de_venda BIGSERIAL PRIMARY KEY,
    nome VARCHAR(60) NOT NULL,
    cnpj CHAR(14) NOT NULL,
    cep CHAR(9),
    rua VARCHAR(255),
    numero_rua VARCHAR(10),
    telefone VARCHAR(15),
    id_gerente BIGINT REFERENCES funcionario
(id_funcionario)
);

-- Tabela: viagem
CREATE TABLE viagem (
    id_viagem SERIAL PRIMARY KEY,
    num_reservas_associadas INT
);

-- Tabela: cidade
CREATE TABLE cidade (
    id_cidade SERIAL PRIMARY KEY,
    nome VARCHAR(50) NOT NULL,
```



```
        codigo CHAR(3) NOT NULL UNIQUE
    );

-- Tabela: aeroporto
CREATE TABLE aeroporto (
    id_aeroporto SERIAL PRIMARY KEY,
    nome VARCHAR(50) NOT NULL,
    codigo INT NOT NULL UNIQUE
);

-- Tabela: estacao
CREATE TABLE estacao (
    id_estacao SERIAL PRIMARY KEY,
    nome VARCHAR(60) NOT NULL
);

-- Tabela: porto
CREATE TABLE porto (
    id_porto SERIAL PRIMARY KEY,
    nome VARCHAR(60) NOT NULL
);

-- Tabela: local (ligação com cidade e pontos específicos)
CREATE TABLE local (
    id SERIAL PRIMARY KEY,
    id_cidade INT NOT NULL REFERENCES cidade
(id_cidade),
    id_aeroporto INT REFERENCES aeroporto
(id_aeroporto),
    id_estacao INT REFERENCES estacao (id_estacao),
    id_porto INT REFERENCES porto (id_porto)
);

CREATE TYPE tipo_local_enum AS ENUM ('origem', 'destino',
'escala');

-- Tabela: viagem_local (relação entre viagem e local)
CREATE TABLE viagem_local (
    id_viagem INT NOT NULL REFERENCES viagem(id_viagem),
    id_local INT NOT NULL REFERENCES local(id),
```



```
        tipo tipo_local_enum NOT NULL,
        PRIMARY KEY (id_viagem, id_local)
    );

-- Tabela Modal
CREATE TABLE modal (
    id_modal BIGSERIAL PRIMARY KEY,
    tipo VARCHAR(50) NOT NULL,
    modelo VARCHAR(50),
    capacidade INT NOT NULL,
    ano_fabricacao INT,
    nome_empresa VARCHAR(50) NOT NULL
);

-- Tabela Viagem_Modal (Relacionamento entre viagem e modal)
CREATE TABLE viagem_modal (
    id_viagem BIGINT REFERENCES viagem(id_viagem),
    id_modal BIGINT REFERENCES modal(id_modal),
    PRIMARY KEY (id_viagem, id_modal)
);

-- Tabela Reserva
CREATE TABLE reserva (
    id_reserva BIGSERIAL PRIMARY KEY,
    data DATE,
    status VARCHAR(10),
    valor NUMERIC(10, 2),
    origem VARCHAR(50),
    destino VARCHAR(50),
    tipo_passagem VARCHAR(10),
    localizador VARCHAR(50),
    id_cliente BIGINT NOT NULL REFERENCES cliente(id_cliente),
    id_funcionario BIGINT REFERENCES funcionario(id_funcionario),
    id_viagem BIGINT REFERENCES viagem(id_viagem)
);

-- Tabela Pagamento
CREATE TABLE pagamento (
```



```
        id_pagamento SERIAL PRIMARY KEY,
        valor NUMERIC(10,2) NOT NULL,
        data_pagamento DATE NOT NULL,
        metodo_pagamento VARCHAR(20),
        id_cartao INT REFERENCES cartao(id_cartao),
        id_reserva INT NOT NULL REFERENCES reserva(id_reserva)
    );

-- Tabela Parcelas
CREATE TABLE parcela (
    id_parcela SERIAL PRIMARY KEY,
    numero_parcela INT NOT NULL,
    valor_parcela NUMERIC(10,2),
    data_vencimento DATE NOT NULL,
    data_pagamento DATE NOT NULL,
    id_pagamento INT NOT NULL REFERENCES
pagamento(id_pagamento)
);

-- Tabela Ticket
CREATE TABLE ticket (
    id_ticket BIGSERIAL PRIMARY KEY,
    tipo_passagem VARCHAR(50),
    hora_de_partida TIMESTAMP NOT NULL,
    hora_de_chegada TIMESTAMP NOT NULL,
    localizador VARCHAR(50),
    id_reserva BIGINT NOT NULL REFERENCES reserva
(id_reserva)
);
```

4.2 - Implementação do Banco de Dados

Nesta etapa, foi realizada a implementação do banco de dados físico utilizando o **SGBD PostgreSQL** com o auxílio da ferramenta **PgAdmin 4**. A implementação incluiu a criação das tabelas, definição de atributos, chaves primárias, chaves estrangeiras e relacionamentos, conforme o modelo físico definido anteriormente



Ferramentas utilizadas:

SGBD: PostgreSQL 17

Ferramenta de Gerenciamento: PgAdmin 4

Linguagem Utilizada: SQL

Descrição do Processo de Implementação

1. Criação do Banco de Dados

Acessou o PgAdmin 4 e criou um novo banco de dados chamado VVV.

2. Criação das Tabelas

As tabelas foram criadas utilizando scripts SQL baseados no modelo físico.

3. Execução dos Scripts SQL

Os scripts SQL foram inseridos e executados no editor de consultas do PgAdmin 4.

4. Verificação da Estrutura

Após a execução dos scripts, verificou-se a estrutura das tabelas no PgAdmin 4, garantindo que os relacionamentos e atributos foram criados corretamente.



The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' shows a database structure with tables like 'cliente'. The main window displays a SQL query: `SELECT * FROM public.cliente ORDER BY id_cliente ASC`. Below the query, the 'Data Output' tab shows a single row of data for a client.

id_cliente	nome	data_nascimento	email	senha	cpf	profissao	cep
1	Fulano de Tal	2001-01-01	fulanodetal@gmail.com	fulano123	199.000.000-00	Desenvolvedor	20000-000

Figura 4: Consulta de clientes cadastrados pelo site da VVV no banco de dados implementado

4.3 - Implementação do RF001

Implementação do Requisito Funcional 001

Requisito: Cadastro cliente.

Objetivo: O cliente poderá ser cadastrado no sistema web.

Fluxo da Implementação

1. O usuário envia os dados via requisição HTTP (POST).
2. Os dados são validados para garantir que estão no formato correto.
3. Se forem válidos, os dados são armazenados no banco de dados.
4. A API retorna uma mensagem indicando sucesso ou erro.

Trecho do código da classe ClienteWebController

```
package group.vvv.controllers;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;
```



```
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
import group.vvv.models.Cliente;
import group.vvv.services.ClienteService;

@Controller
@RequestMapping("/web/clientes")
public class ClienteWebController {

    @Autowired
    private ClienteService clienteService;

    @GetMapping("/novo")
    public String exibirFormularioCadastro(Model model) {
        model.addAttribute("cliente", new Cliente());
        return "formulario-cliente";
    }

    @PostMapping
    public String salvarClienteWeb(@ModelAttribute Cliente cliente,
    Model model) {
        clienteService.salvar(cliente);
        model.addAttribute("mensagem", "Cliente cadastrado com
sucesso!");
        return "formulario-cliente";
    }
}
```

Testando a Implementação

Exemplo de Requisição (Entrada)

Método: POST

URL: /web/clientes

Corpo:



← → ↻ localhost:8082/web/clientes

Cadastro de Cliente

Nome
Fulano de Tal

Data de Nascimento
dd/mm/aaaa

Email
fulanodetal@gmail.com

Senha

CPF
199.000.000-00

Profissão
Desenvolvedor

CEP
20000-000

Rua
Beltrano

Número da Rua
123

UF
RJ

Telefone
2 19999999999

[Cadastrar](#)

Resposta de Sucesso (Saída)

Cliente cadastrado com sucesso!

Resultado no Banco de Dados

	id_cliente [PK] bigint	nome character varying (60)	data_nascimento date	email character varying (320)	senha character varying (255)	cpf character varying (14)	profissao character varying (30)	cep character varying (9)
1	1	Fulano de Tal	2001-01-01	fulanodetal@gmail.com	fulano123	199.000.000-00	Desenvolvedor	20000-000

Código Completo no GitHub

Para acessar o código completo, consulte o repositório no GitHub:

-> [Link do repositório](#)

-> [Função Cadastrar Usuário](#)