

Centro Universitário Senac  
Campus Santo Amaro

Gustavo Orlando Araújo Vergani  
João Victor Farias Teixeira  
Maria Mercedes da Silva Rodrigues  
Raissa Miranda Casilla

# Sistema Rodoviário ExpressoBus

São Paulo  
2024

Gustavo Orlando Araújo Vergani  
João Victor Farias Teixeira  
Maria Mercedes da Silva Rodrigues  
Raissa Miranda Casilla

Trabalho de Graduação  
apresentado à Disciplina: Projeto  
Integrador - Desenvolvimento de  
Sistemas Orientado a Objetos, no  
3º Semestre do curso: ANÁLISE  
E DESENVOLVIMENTO DE  
SISTEMAS - MATUTINO do  
Centro Universitário Senac –  
Santo Amaro.

Orientador: Marcos de Arruda  
Monteiro

São Paulo  
2024

## RESUMO

O presente projeto tem como objetivo aprofundar e apresentar todo o conteúdo abordado ao longo do terceiro semestre do curso Tecnologia em Análise e Desenvolvimento de Sistemas. Sua elaboração consiste no desenvolvimento de um sistema rodoviário, visando expandir os conhecimentos em Programação Orientada a Objeto de forma simples e dinâmica.

Ao longo do trabalho, vamos citar cada etapa de elaboração, desde a especificação da ideia até seus diagramas para o início da codificação, proporcionando assim, uma visão detalhada do desenvolvimento do projeto. É de suma importância ressaltar que o andamento do trabalho se sucedeu através das aulas práticas de POO e de Projeto Integrador, nas quais pudemos compreender o escopo do mesmo.

Sua coleta e análise de dados foram realizadas por meio de códigos e métodos em linguagem Java, utilizando bibliotecas Swing, SQL, Random, Java Time, entre outras classes e conceitos fundamentais da programação orientada a objetos. Esses elementos contribuíram para o desenvolvimento de um sistema que proporciona satisfação ao cliente ao realizar compras de passagens rodoviárias em regiões específicas do Brasil.

**Palavras-Chave:** POO, Linguagem Java, Swing. SQL.

# SUMÁRIO

Introdução.....	5
Planejamento / Cronograma.....	6
Responsabilidades da Equipe.....	7
Especificação do Sistema.....	8
Solução de Software.....	10
Informações Sobre a Codificação do Sistema.....	11
Cenário.....	12
Diagrama de Caso de Uso.....	13
Descrição de Caso de Uso.....	13
Diagrama de Classes.....	15
Modelagem de Dados.....	16
Conclusão.....	17

# Introdução

O atual projeto tem como objetivo principal expandir os conhecimentos na disciplina de Programação Orientada a Objeto, através da criação de um sistema rodoviário. Este sistema permite a compra de passagens para as regiões sul e sudeste do Brasil, levando em consideração as preferências individuais dos clientes.

Denominado como “ExpressoBus”, o projeto foi escolhido com base nos requisitos estabelecidos pelo nosso orientador, Marcos Arruda, onde teríamos que realizar um projeto que atendesse os requisitos de busca, compra, reserva, pagamento, cancelamento, alteração, gestão de conta e histórico de passagens. Com essas propostas, optamos rapidamente por desenvolver um sistema de compra de passagens, dada a existência de plataformas online que já oferecem aos clientes todos os critérios desejados, tornando-se ideal para o projeto em desenvolvimento.

A aplicação prática dos conhecimentos adquiridos na disciplina se reflete na forma como construímos todo o trabalho, utilizamos bibliotecas em Java, como Swing, conceitos da programação orientada a objetos, além de classes fundamentais para a criação de interfaces gráficas, como AWT (Abstract Window Toolkit). Adicionalmente, aproveitamos os conhecimentos adquiridos em semestres anteriores e adicionamos o Banco de Dados MySQL, possibilitando que nosso sistema se tornasse prático e seguro ao usuário final. Ao realizar este trabalho, buscamos não apenas expandir nossos conhecimentos, mas também desenvolver experiência e habilidades que nos impulse para o campo da tecnologia, de uma maneira prática na solução de problemas.

## Planejamento / Cronograma

Tarefa	Data
Concepção de ideias	18/04/2024
Planejamento e Alinhamento do projeto	19/04/2024
Organização dos integrantes do grupo	22/04/2024
Criação dos Diagramas	23/04/2024
Criação do Banco de Dados MySQL	25/04/2024
Desenvolvimento da Interface Gráfica	26/04 à 30/04
Desenvolvimento da Lógica do Sistema	30/04 à 14/05
Documentação do Projeto	20/05/2024

## Responsabilidades da Equipe

Após conversas e aprovação da proposta do projeto de um sistema rodoviário pela equipe, estabelecemos uma divisão de responsabilidades para cada membro do grupo, atribuindo a cada um uma função específica. Segue abaixo mais detalhes sobre cada função e seu respectivo responsável.

### - **Raissa Miranda Casilla**

Raissa ficou encarregada de liderar o processo de geração de ideias. Sua responsabilidade foi identificar os requisitos estabelecidos pelo orientador Marcos Arruda e conceber uma proposta que os atendesse. Ela também elaborou os diagramas de caso de uso e de classes, além da modelagem de dados, para facilitar o entendimento do grupo e servir como base para a implementação. Além disso, desenvolveu a documentação do projeto, garantindo que todas as etapas fossem registradas de forma clara e organizada.

### - **Gustavo Orlando Araújo Vergani**

Gustavo assumiu a responsabilidade pela implementação do banco de dados, seguindo a modelagem já feita anteriormente. O mesmo implementou tabelas de preços para facilitar o gerenciamento do programa, além de armazenar informações de forma organizada. Nosso banco de dados conta com tabelas específicas para clientes, histórico de passagens e registros de passagens já adquiridas, todas interligadas através de identificadores automáticos para cada cliente, garantindo um controle e estrutura sólida e eficiente para o desenvolvimento do projeto.

### - **Maria Mercedes da Silva Rodrigues**

Maria desempenhou o papel da interface gráfica do projeto. Utilizando bibliotecas em Java, como Swing e classes AWT. Ela incorporou elementos como botões, caixas de seleção, janelas e contêineres, além de atribuir eventos de ação, como cliques em botões e marcação em caixas de seleção, garantindo assim uma interação dinâmica. Isso resultou em uma interface intuitiva e visualmente atraente, assegurando a fluidez do projeto para os usuários finais.

### - **João Victor Farias Teixeira**

João foi o responsável pelo desenvolvimento de toda a lógica do sistema rodoviário. Ele dedicou-se à implementação de todas as funcionalidades do projeto, assegurando a máxima eficiência e precisão em cada método. Para que isso fosse possível, utilizou diversas classes e pacotes essenciais, como conexão SQL, gerador de números aleatórios (Random) e o pacote Java.Time para manipulação de datas e horários. Além disso, aplicou raciocínios lógicos para o controle dos dados, compreendendo os requisitos do projeto e transformando-os em códigos funcionais em linguagem Java.

# Especificação do Sistema

O sistema desenvolvido tem como um dos objetivos ampliar nossos conhecimentos de maneira prática. Com esse propósito, concluímos desenvolver um projeto de um sistema rodoviário que atendesse aos requisitos propostos e que fosse verdadeiramente utilizado no cotidiano da população.

O ExpressoBus, tem como finalidade disponibilizar a compra de passagens de ônibus para as regiões sul e sudeste do Brasil, oferecendo preços acessíveis e tipos de acomodações de viagem. Além disso, priorizamos a segurança, eficiência e a satisfação do cliente.

Desde o princípio, nosso sistema propõe a segurança ao exigir que os clientes tenham uma conta para acessá-lo, isso nos permite compreender melhor as preferências de cada usuário. Para os clientes que já possuem uma conta, basta fazer o login utilizando o e-mail e a senha associados. Para aqueles que não possuem uma conta, é necessário criar uma nova.

Após efetuar o login, o cliente é direcionado ao menu principal, onde encontrará opções de compra, viagens programadas, histórico de passagens, perfil e saída. Cada uma dessas opções possui conteúdos em páginas distintas, proporcionando uma visualização limpa e agradável.

Para que tudo isso aconteça, existe uma grande estrutura back-end por trás. Para um melhor entendimento de cada cenário, iremos especificar cada opção de menu por tópicos:

## 1. Login

Como mencionado anteriormente, a primeira tela a ser visualizada pelo cliente será a de login, desenvolvida para garantir a segurança e preferência de cada cliente. Essa tela serve como tela de entrada, onde possui campos para serem preenchidos, com email e senha. Caso o cliente não possua uma conta, ele terá a opção de 'Criar Conta', localizada no canto direito da tela. Ao selecionar essa opção, será direcionado para um formulário de cadastro no qual preencherá seus dados, incluindo: nome completo, RG, telefone, email e senha a ser criada. É importante ressaltar que o RG passará por uma validação para verificar sua autenticidade.

## 2. Menu Principal

Efetuando o login, ou criando uma nova conta, o cliente será designado para o menu principal, onde terá acesso às seguintes serviços:

- **Comprar Passagem** - Esta ferramenta permite aos usuários visualizar destinos no sul e sudeste do Brasil, oferecendo variedades de viagens. Os clientes podem escolher entre acomodações padrão ou leito-cama, além de selecionar a data, o assento e o horário desejados para a sua viagem.  
O preço da passagem varia de acordo com as preferências do cliente, incluindo o tipo de acomodação e o destino escolhido. Optar pelo leito-cama implica em um acréscimo de 100 reais ao valor da passagem.



Após selecionar seus critérios de viagem, o cliente recebe uma mensagem de confirmação contendo todos os detalhes da passagem, contendo a opção de confirmar ou cancelar a compra. Ao optar por confirmar, o cliente recebe seu código de pagamento via Pix, realiza o pagamento e pronto: a viagem é marcada, os dados são salvos no banco de dados e a reserva fica disponível para visualização na seção 'Viagens Programadas'

- **Viagens Programadas** - Nesta opção teremos a tela que apresenta ao usuário suas viagens que já estão com datas marcadas, onde há a possibilidade do cliente cancelar, reagendar ou finalizar a respectiva viagem..
- **Histórico de Passagens** - Este recurso oferece aos clientes a oportunidade de visualizar um registro de suas viagens passadas. Os detalhes abrangem origem, destino, data, horário, tipo de acomodação, número do assento e preço correspondente, fornecendo informações sobre as preferências de viagem do cliente.
- **Perfil** - Ao acessar a seção "Perfil", os clientes terão a possibilidade de atualizar e gerenciar suas informações pessoais, garantindo uma experiência personalizada e segura. Antes de realizar qualquer atualização, os clientes serão solicitados a inserir a sua senha atual para garantir segurança de seus dados.
- **Sair** - Opção que encerra os serviços do sistema ExpressoBus, retornando para a tela de login.

## **Solução de Software**

Em nosso sistema rodoviário implementamos inúmeras soluções para efetuar a compra de passagens para regiões sul e sudeste do Brasil, de maneira eficaz e prática, visando sempre a boa experiência do nosso usuário ao realizar uma viagem.

Nossas soluções foram:

- Login de acesso ou Cadastro de novos clientes
- Compra de passagens rodoviárias
- Confirmação da compra antes de realizar o pagamento
- Pagamento via código copia e cola pix
- Visualização das viagens Programadas
- Finalizar, cancelar ou reagendar a viagem
- Histórico de passagens
- Gerenciamento de perfil

## Informações Sobre a Codificação do Sistema

A estruturação do sistema foi realizada por meio da utilização de classes e ferramentas da linguagem Java, seguindo os princípios da programação orientada a objetos. Utilizamos as classes Swing, Awt, SQL, Random, Time e JOptionPane para garantir total funcionalidade e manipulação necessária.

Durante o processo de desenvolvimento, optamos por criar classes específicas para cada tipo de codificação, priorizando a organização e a legibilidade do código. No total, foram desenvolvidas 12 classes, das quais sete foram dedicadas à criação das telas e suas respectivas lógicas de funcionamento, duas para a conexão e manipulação do banco de dados, uma para validação de RG e outra para verificar datas em relação ao desenvolvimento do projeto. A seguir detalharemos cada classe e suas principais manipulações dentro delas:

- **ChecaData**

A classe 'ChecaData' oferece funcionalidades para auxiliar na compra e no histórico de passagens dos usuários finais. Através de duas importações *do pacote java.time (LocalDate e Temporal.ChronoUnit)*, a classe fornece dois métodos essenciais, sendo eles: ProxDatas e DifDias.

- **ProxDatas:** Atualiza as datas de viagens disponíveis a cada 2 dias, garantindo que os usuários sempre tenham passagens disponíveis.
- **DifDias:** Calcula a diferença entre o dia atual e o dia da viagem, fornecendo informações para a regra de negócio do sistema rodoviário. Como por exemplo, a manipulação de cancelamentos e reagendamentos de viagens.

- **CompraPassagem**

A classe 'CompraPassagem' importa classes dos *pacotes Swing, AWT e Random* para personalizar e manipular a interface gráfica e a funcionalidade dos códigos. Utiliza o conceito de encapsulamento para criar um ID privado para cada cliente, garantindo proteção e controle de acesso no banco de dados.

Nos métodos, a classe implementa a lógica de compra de passagens, confirmação e pagamento, além de tratar possíveis erros ou ações inesperadas do usuário final.

- **ConectaMySQL**

A classe 'ConectaMySQL' trata a conexão com o banco de dados, através do *pacote 'java.sql.\*'*. Seus métodos garantem a abertura e o fechamento da conexão com a linguagem Java, além de prevenir falhas e otimizar o acesso à informação.

- **CriaComponentes**

A classe 'CriaComponentes' foi desenvolvida utilizando as *classes dos pacotes Swing e AWT, estendendo a 'JFrame'*. Essa classe facilita e organiza o processo de criação de interfaces gráficas em Java, ela encapsula a lógica necessária para criar e configurar diversos tipos de componentes, como botões, campos de texto, rótulos, painéis, entre outros.

- **CriaConta**

A classe 'CriaConta' oferece uma interface gráfica para a criação de novas contas de usuários no sistema. A mesma possui herança em '*Jframe*' e estende a interface '*ActionListener*', além das importações do pacote *Swing e AWT* para construção da interface.

Dentro da classe, são instanciados objetos para facilitar a criação de componentes visuais, lidar com ações SQL e validar o número de RG inserido pelo usuário. Após a configuração inicial da interface, o *método 'ActionPerformed'* manipula as informações fornecidas pelo usuário, como nome, RG, telefone, e-mail e senha, que são armazenadas em um banco de dados. Cada conta criada é associada a um ID privado único para identificação posterior.

- **EventosSQL**

A classe 'EventosSQL' importa o *pacote 'sql'* e *herda a classe 'ConectaSQL'*. A mesma possui 591 linhas e 18 métodos que tratam das verificações no banco de dados.

- **HistóricoPassagens**

A classe 'HistoricoPassagens' utiliza *importações da biblioteca Swing e AWT* para criação gráfica, exibindo o histórico de passagens de um cliente. Ela *herda funcionalidades da classe 'JFrame'* e *implementa o 'ActionListener'* para lidar com eventos de botão. A janela exibe informações sobre as passagens do cliente, incluindo seu id de passagem, código do pagamento, origem, destino, data e horário da viagem, assento, tipo de acomodação e seu respectivo valor.

- **Login**

A classe 'Login' importa as *bibliotecas Swing e AWT*, além de utilizar as classes '*criaComponentes*' e '*EventosSQL*' para criar componentes gráficos e lidar com eventos de interação. Seu método é uma *sobrescrita de 'ActionPerformed'* para realizar as ações de acordo com a escolha do usuário.

Sua classe permite a conexão dos dados do cliente e fornece ao menu principal do programa. A mesma busca no banco de dados as informações inseridas no campo de texto de login, faz tratamentos de erros e possui um direcionamento para tela de criação de nova conta se o cliente não obter sucesso em seu login.

- **Perfil**

A classe 'Perfil' importa as *bibliotecas Swing e AWT*, e *herda funcionalidades da classe 'JFrame' e implementa o 'ActionListener'*. Ao instanciar essa classe, uma janela é exibida para o usuário com campos pré-preenchidos contendo os dados atuais do cliente, como nome, RG, telefone, e-mail e senha. Estes dados são obtidos do banco de dados utilizando *métodos da classe 'EventosSQL'*.

O principal recurso é editar o perfil do cliente, sendo assim, ao clicar no botão "Editar", o usuário pode fazer alterações nos campos de texto correspondentes às informações do perfil. No entanto, antes de confirmar as mudanças, uma janela de confirmação é exibida, solicitando que o usuário insira sua senha.

Uma vez validada a senha, as informações do perfil são atualizadas no banco de dados com os novos valores fornecidos pelo usuário.

- **TelaInicial**

Essa classe é responsável por exibir o menu principal do sistema para o cliente, oferecendo uma série de opções e funcionalidades disponíveis. A mesma também *herda de 'JFrame' e implementa 'ActionListener'* para lidar com eventos de interação com o usuário.

No construtor da classe, é configurado as propriedades visuais da janela, e em seguida são criados os elementos gráficos utilizando os *métodos da classe 'CriaComponentes'*. Esses elementos são posicionados na interface utilizando o *método 'setBounds'*.

O menu principal é dividido em duas seções, os botões de passagens (Compra Passagem, Viagens Programas e Histórico de Passagens) e os de ações (Perfil e Sair). Cada botão é associado a um evento de clique (ActionListener), que direciona o cliente para a tela correspondente à ação escolhida.

- **ValidaRG**

A classe 'ValidaRG' tem como finalidade validar o dígito verificador de um número de RG.

Ela contém um construtor padrão e um método chamado 'validar', que recebe como parâmetro uma string representando o número do RG a ser validado.

O método multiplica cada dígito do RG pela sua posição inversa, soma os resultados, calcula o resto da divisão dessa soma por 11 e compara o resultado com o dígito verificador. Se coincidir, retorna indicando que o RG é válido.

- **ViagensProgramadas**

Por fim, temos a classe 'ViagensProgramadas', responsável por exibir as viagens do cliente, permitindo que ele consulte, finalize, cancele ou remarque o dia e horário de sua viagem.

O método construtor define a parte gráfica da página, além de ouvir os eventos dos botões, os demais métodos obtêm a *sobrescrita de 'ActionPerformed'* e permite às manipulações dos botões..

- **Finalizar:** Solicita ao usuário o ID da passagem a ser finalizada. Se o ID for válido, verifica se o usuário confirma a finalização da viagem. Se confirmado, move a passagem para o histórico de viagens e o remove da lista de viagens programadas.

- **Cancelar:** Solicita ao usuário o ID da passagem a ser cancelada. Se o ID for válido, verifica se o usuário ainda está dentro do prazo para cancelar a viagem. Se sim, confirma com o usuário e notifica sobre o estorno do valor e remove a passagem da lista de viagens programadas.
- **Reagendamento:** Solicita ao usuário o ID da passagem a ser alterada. Se o ID for válido, verifica se o usuário ainda está dentro do prazo para reagendar sua viagem. Se sim, solicita ao usuário uma nova data e horário disponíveis no sistema e atualiza os dados da viagem.
- **Voltar:** Retorna à tela inicial do sistema.

## **Cenário**

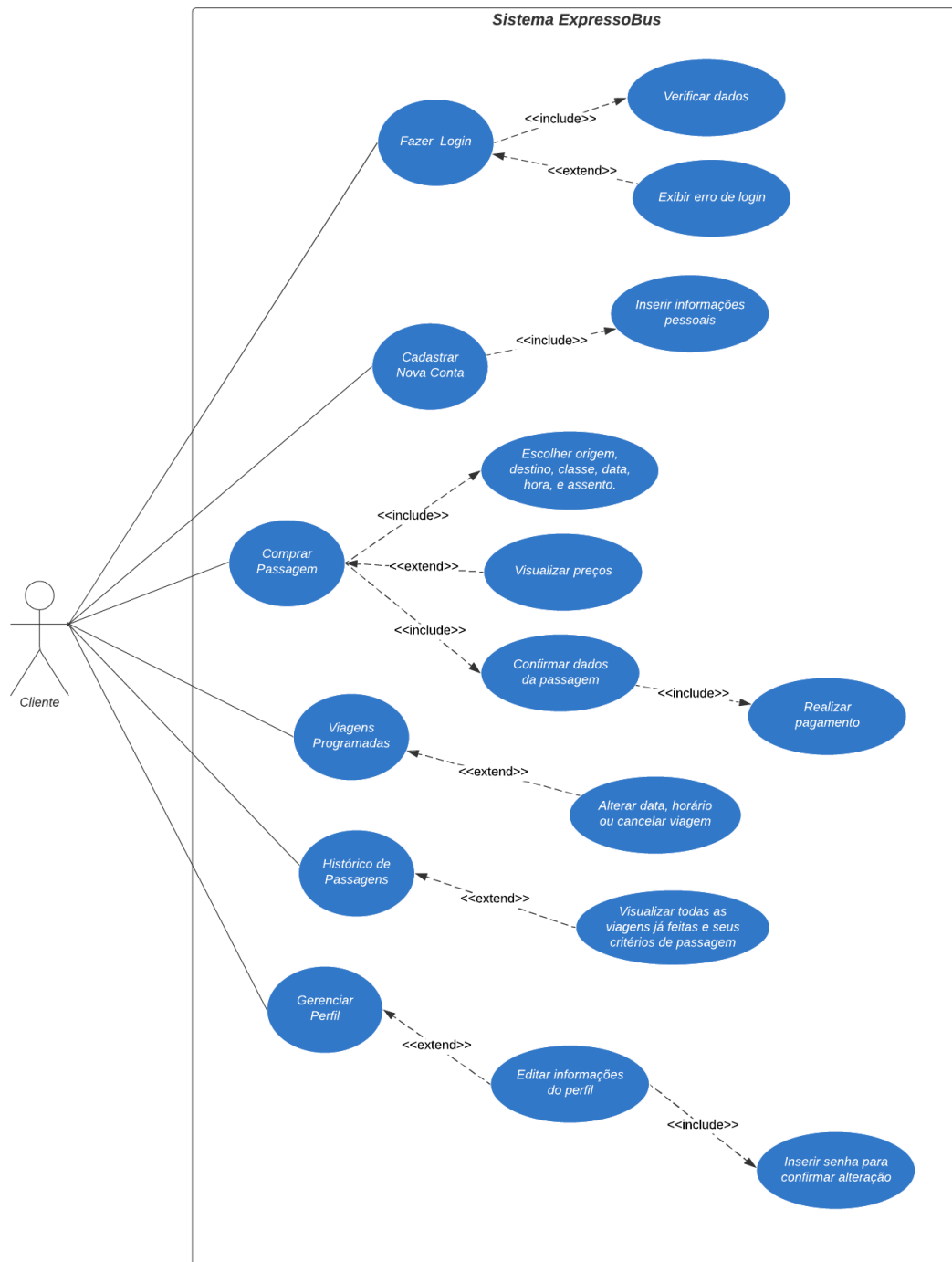
Visando o desenvolvimento do projeto, para realizar uma compra precisamos passar por algumas etapas, para isso criaremos uma situação hipotética de um cliente que deseja viajar de São Paulo para Santa Catarina.

Seus passos serão:

1. Acessar o sistema utilizando o email e senha cadastrados.
2. Escolher a opção "Comprar Passagem" para iniciar o processo de compra.
3. Selecionar São Paulo como origem e Santa Catarina como destino.
4. Escolher o tipo de acomodação desejada (Leito-Cama).
5. Agendar data, horário e seleção do número de assento.
6. Verificar o preço da passagem e clicar em "Comprar".
7. Revisar e confirmar os dados da compra, avançando para o pagamento.
8. O sistema gera um código de pagamento via PIX.
9. O cliente realiza o pagamento utilizando o código recebido.
10. O sistema confirma o pagamento e a viagem é agendada.

Após a confirmação, o cliente pode visualizar todos os detalhes da viagem na seção "Viagens Programadas", com a opção de remarcar ou cancelar sua viagem até 1 dias antes do embarque.

# Diagrama de Caso de Uso





# Descrição de Caso de Uso

## Ator Principal:

- Cliente

## Interessados:

- Cliente deseja comprar passagem para regiões sul e sudeste do Brasil.
- Sistema responsável por gerenciar as reservas e vendas de passagens.

## Pré-Condições:

- O cliente possui uma conta válida no sistema de reserva de passagens rodoviárias.
- O cliente deseja realizar uma viagem de ônibus.

## Fluxo Principal:

1. O caso de uso começa quando o cliente faz login no sistema.
2. O cliente seleciona a opção compra de passagem e especifica a origem, destino, classe, data, hora, assento e tipo de acomodação desejada.
3. O sistema fornece os detalhes da viagem e solicita a confirmação da compra.
4. O cliente confirma e realiza o pagamento através do código disponibilizado pelo sistema.
5. O sistema registra a compra e atualiza dados, exibindo detalhes da viagem na opção de “viagens programadas”.

## Pós-Condições:

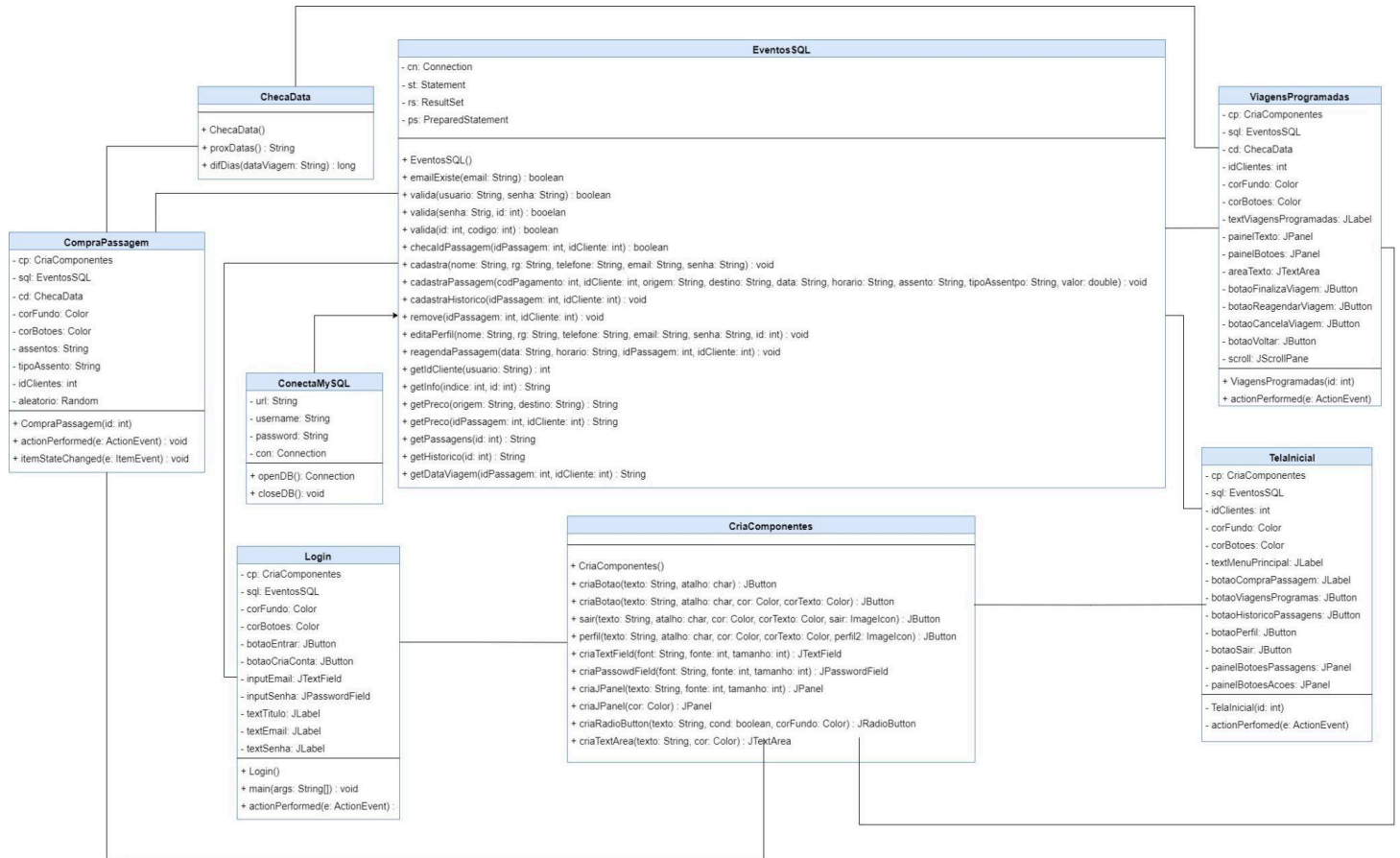
- A compra de passagem é registrada no sistema associada ao ID do cliente.
- O perfil do cliente é atualizado com informações sobre a viagem programada

## Fluxo Alternativo:

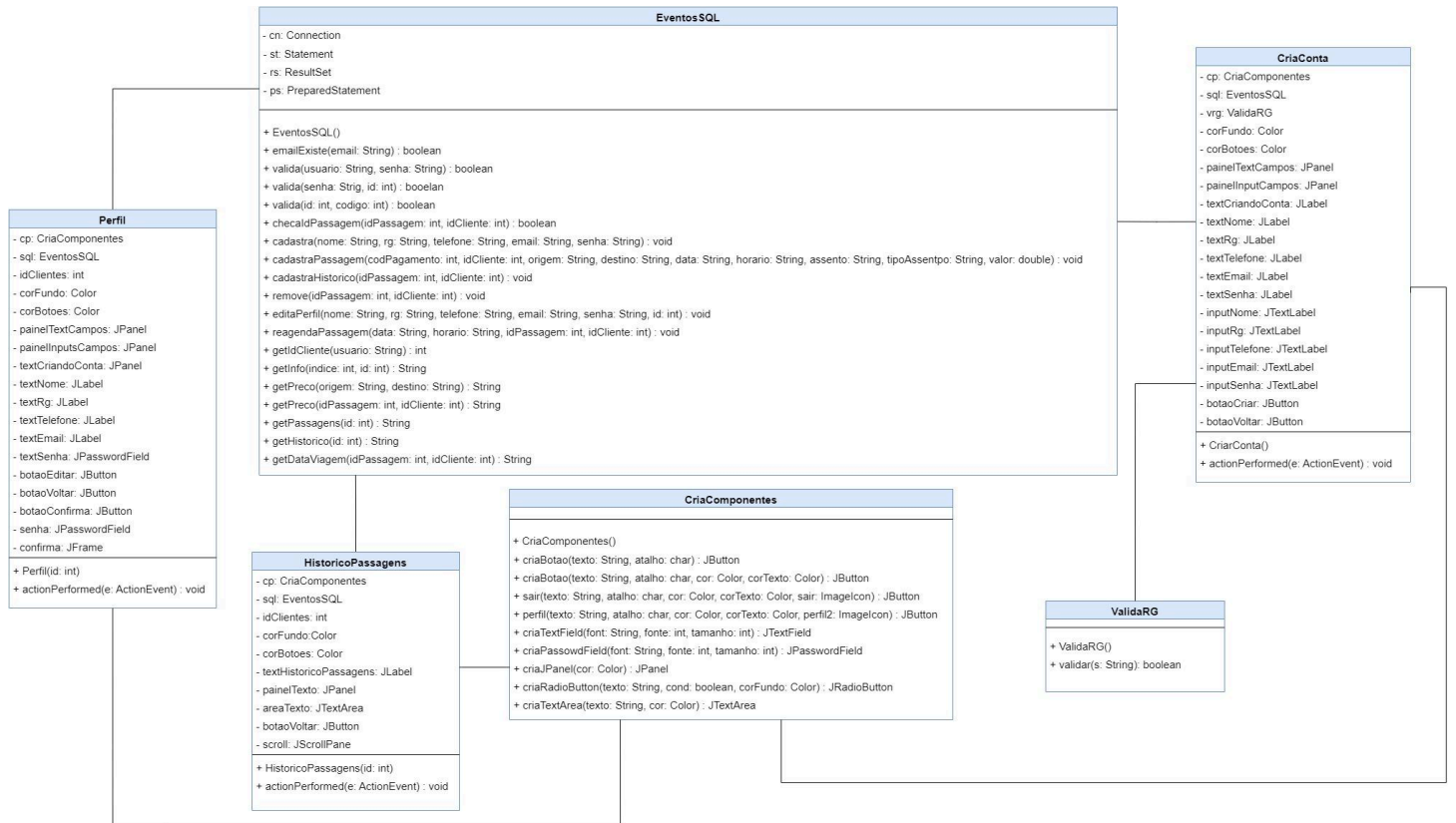
- Se as informações de login forem inválidas ou inexistentes, o sistema exibe uma mensagem de erro e orienta o cliente a corrigir as informações ou realizar uma nova conta.
- Se o cliente optar por cancelar a compra em qualquer etapa do processo, a transação é interrompida e o sistema retorna à página inicial anterior.

# Diagrama de Classes

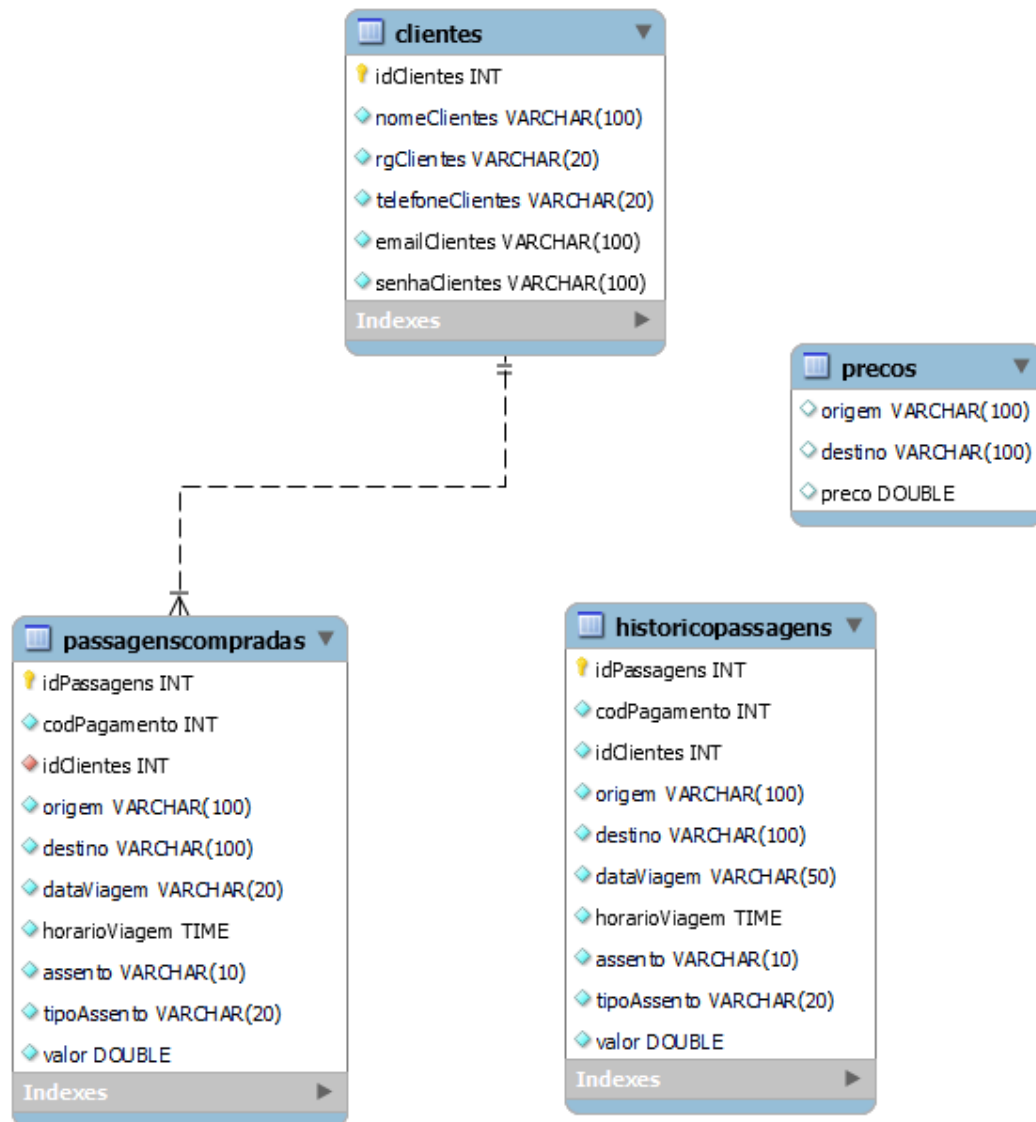
## Part. 1



## Part 2.



# Modelagem de Dados



## Conclusão

Durante a evolução do projeto, adquirimos profunda compreensão e habilidade na prática da programação estruturada, exploramos de maneira significativa os conceitos e a arquitetura da Programação Orientada a Objetos na linguagem Java.

O desenvolvimento do ExpressoBus foi essencial para o crescimento pessoal e profissional de cada membro do grupo. Através da colaboração em equipe, desenvolvemos trocas de ideias, estudos e analogias, onde cada membro contribuiu de forma significativa com suas responsabilidades. Resultando em um projeto fluido, onde alinhamos ideias e pensamentos sem dificuldades ou desistências, destacando uma habilidade crucial na área de tecnologia: o trabalho em equipe.

Compreendemos plenamente a necessidade e a importância de cada etapa, desde a concepção teórica até a sua implementação prática. Sendo assim, acentuamos a importância das aulas dedicadas à sua elaboração, que possibilitaram a comunicação presencial do grupo gerando novos aprendizados.

Os conhecimentos transmitidos ao longo do semestre, nas disciplinas de Projeto Integrador e Programação Orientada a Objetos, foram fundamentais para a realização deste trabalho. Através dessas aulas, conseguimos desenvolver um projeto sem falhas, alcançando todos os objetivos estabelecidos por nosso orientador. É importante mencionar que incorporamos alguns conceitos de semestres passados, como a ampliação de diagramas, banco de dados, manipulação de dados e utilizações extensas de classes do pacote Java, permitindo ter uma base sólida no desenvolvimento backend.

Agradecemos ao orientador, Marcos Arruda, pela sua proatividade em esclarecer dúvidas ao longo do processo. Seu amparo foi fundamental, assim como sua didática, que nos permitiu compreender toda a teoria de forma prática e aplicável.

Contudo, o desenvolvimento desse projeto nos permite adentrar cada vez mais no mundo da programação, integrando de maneira coerente com nosso cotidiano e organizando de maneira clara e concisa nossos conhecimentos ao longo de mais um semestre do curso Análise e Desenvolvimento de Sistemas.

