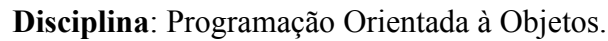


Curso: Bacharelado em Ciência da Computação.

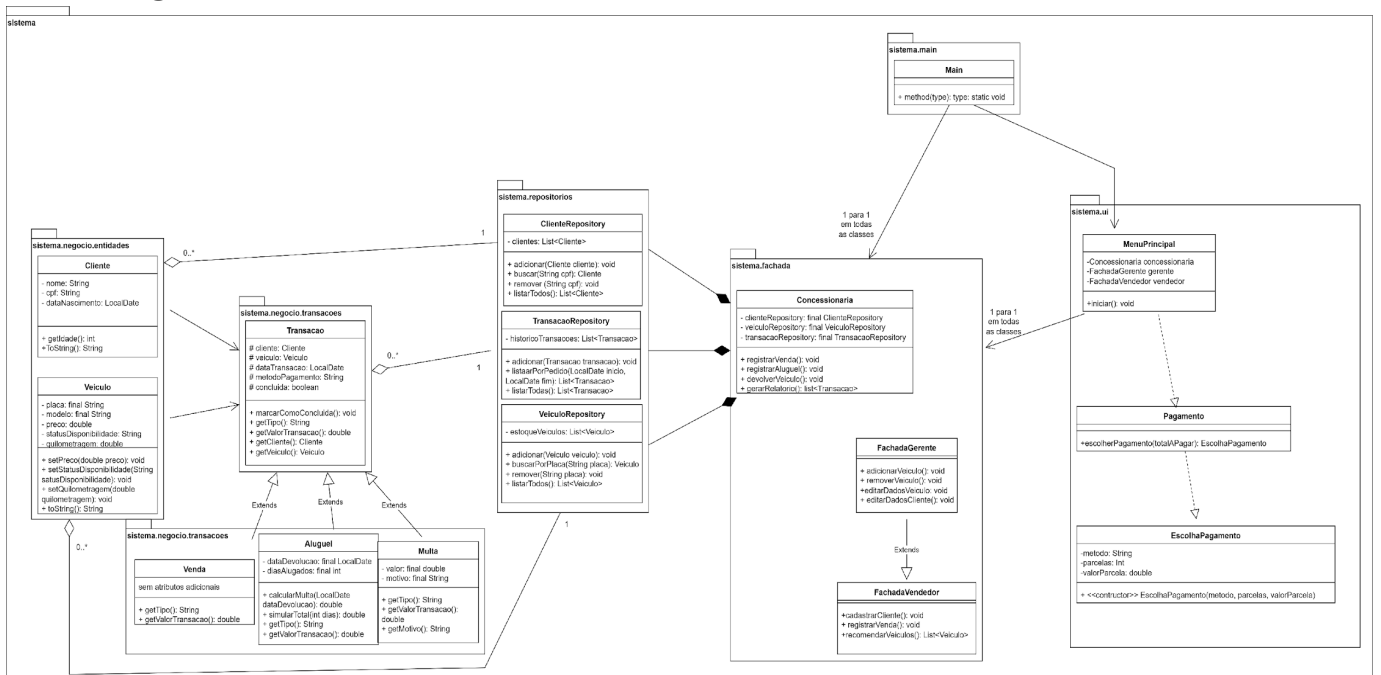


**Professora:** Thaís Alves Burity Rocha.

Semestre: 2025.1.

**Equipe:** Emanuel Bezerra, Pedro Arthur e Cleita Emanuela.

- **Diagrama de classes UML:**



- **Contribuições individuais:**

### **Visão Geral da Arquitetura em Camadas**

A divisão em camadas separa as responsabilidades do sistema, as camadas são:

- **Camada de Apresentação/Interface do Usuário:** Representada pelas classes Main, MenuPrincipal e Pagamento. É a camada com a qual o usuário interage diretamente. A classe Main é o ponto de entrada do sistema, e as classes de menu e pagamento gerenciam a interface e as opções disponíveis para o usuário.
- **Camada de Negócio/Serviço:** A classe Concessionária atua como a principal classe dessa camada. Ela orquestra as operações do sistema, chamando métodos de outras classes para realizar tarefas complexas, como registrar um aluguel ou devolver um veículo. É aqui que a lógica de negócios central reside.
- **Camada de Acesso a Dados (Repository):** As classes ClienteRepository, TransacaoRepository e VeiculoRepository representam essa camada. Elas são responsáveis pela persistência dos dados, ou seja, pela leitura, escrita e manipulação das informações de clientes, transações e veículos em um banco de dados ou outro tipo de armazenamento. A separação garante que a lógica de negócio na camada superior não precise saber os detalhes de como os dados são armazenados.

### **O Padrão de Projeto Fachada**

A classe Concessionária age como a fachada do sistema:

- **O Papel da Fachada:** A Concessionária simplifica o acesso a um conjunto de subsistemas mais complexo. Em vez de a Main ou o MenuPrincipal precisarem se comunicar diretamente com cada um dos repositórios (ClienteRepository, TransacaoRepository, etc.), eles interagem apenas com a Concessionária.
- **Vantagens:** Isso reduz o acoplamento entre as camadas. A Concessionária "esconde" a complexidade de gerenciar e coordenar as operações dos diferentes repositórios, oferecendo uma interface única e simplificada. Por exemplo, para "registrar um aluguel", a lógica pode envolver a verificação do cliente (ClienteRepository), a disponibilidade do veículo (VeiculoRepository) e o registro da transação (TransacaoRepository), mas a camada de apresentação só precisa chamar o método registrarAluguel() na Concessionária.

---

### **Classes e Relacionamentos:**

- **Classes de Negócio (Cliente, Veiculo, Transacao, Aluguel, Multa):** Estas classes representam os objetos centrais do domínio do problema. Elas contêm atributos (como nome, cpf, placa) e métodos (como getInfoCliente()) que modelam o comportamento e as características desses objetos.

- **Classes de Repositório:** As classes de repositório gerenciam a persistência dos objetos de negócio. Elas têm métodos como buscar, adicionar e remover, que são essenciais para interagir com o armazenamento de dados.
- **Relacionamentos:**
  - **Associação:** Indica que as classes estão relacionadas. Por exemplo, a classe Transacao está associada a Cliente e Veiculo.
  - **Associação de Agregação/Composição:** A linha com um losango na ponta indica que um objeto "contém" ou "é parte de" outro. Por exemplo, Concessionária tem associações de composição (losango preenchido) com os três repositórios, indicando que a Fachada é responsável pela criação e ciclo de vida desses objetos. Já a classe Transacao tem associações de agregação (losango vazio) com Cliente e Veiculo, o que significa que uma transação "usa" um cliente e um veículo, mas não é responsável por sua existência.
  - **Herança/Generalização:** A seta com um triângulo na base (como de Aluguel e Multa para Transacao) indica herança. Isso significa que Aluguel e Multa são tipos específicos de Transacao, herdando seus atributos e comportamentos.
  - **Dependência:** As linhas tracejadas com setas indicam uma dependência. Por exemplo, MenuPrincipal e Pagamento dependem de Main, o que significa que o código de Main pode usar ou chamar o código dessas classes. Da mesma forma, Concessionária depende dos repositórios para executar suas operações.
  - **Realização de Interface:** As linhas tracejadas que chegam na interface RetirarPagamento indicam que as classes PagamentoCartao e PagamentoDinheiro implementam essa interface

**Tabela das contribuições individuais:**

Estórias de Usuário	Responsável
<b>Vendedor</b>	
Como vendedor, eu preciso cadastrar um novo cliente, para que eu possa registrar suas informações e iniciar o processo de venda ou aluguel.	Emanuel
Como vendedor, eu preciso consultar os modelos de veículos disponíveis, para que eu possa oferecer opções aos clientes.	Pedro
Como vendedor, preciso analisar a faixa etária dos clientes e os veículos que eles adquiriram. O objetivo é recomendar veículos mais adequados a clientes de faixas etárias semelhantes.	Emanuel
Como vendedor, eu preciso registrar o método de pagamento de uma transação, para que a venda ou aluguel seja finalizado corretamente.	Pedro, Emanuel
Como vendedor, eu preciso registrar a venda ou aluguel, para que a transação seja documentada e o histórico do cliente atualizado.	Pedro, Emanuel
Como vendedor, eu preciso registrar o método de compra do veículo - venda ou aluguel -, para que eu possa distinguir os serviços.	Pedro
<b>Gerente</b>	
Como gerente, eu preciso adicionar um novo modelo de veículo ao estoque, para que ele esteja disponível para venda ou aluguel.	Pedro

Como gerente, eu preciso remover um modelo de veículo do estoque, para que o inventário esteja sempre atualizado.	Pedro
Como gerente, eu preciso consultar os modelos de veículos no estoque, para que eu possa monitorar o inventário.	Pedro
Como gerente, eu preciso consultar as informações de um cliente existente, para que eu possa acessar seu histórico ou atualizar seus dados.	Emanuel
Como gerente, eu preciso editar os dados de um cliente, para que as informações estejam sempre atualizadas.	Emanuel
Como gerente, eu preciso remover um cliente do sistema, para que dados desnecessários sejam excluídos.	Emanuel
Como gerente, eu preciso gerar relatórios de vendas e aluguéis, para que eu possa analisar o desempenho e tomar decisões estratégicas.	Pedro, Emanuel