

Lab02S01: Modelagem do Sistema de Aluguel de Carros

1. Diagrama de Casos de Uso

O diagrama de casos de uso define os principais atores e as interações que eles têm com o sistema.

Atores:

- **Cliente:** Usuário individual que pode criar, modificar, consultar e cancelar pedidos de aluguel.
- **Agente (Bancos/Empresas):** Responsáveis por modificar e avaliar pedidos.
- **Sistema:** Responsável pelo processamento e armazenamento de dados.

Casos de Uso:

- Cadastro no sistema
- Login
- Criar pedido de aluguel
- Modificar pedido de aluguel
- Consultar pedido de aluguel
- Cancelar pedido de aluguel
- Analisar pedido de aluguel (Agente)
- Avaliar e aprovar/reprovar pedido (Agente)
- Gerar contrato
- Associar contrato a um banco (se aplicável)

2. Histórias do Usuário

História 1: Cadastro de Cliente

Como um cliente, eu quero me cadastrar no sistema informando meus dados pessoais, **para que** eu possa realizar aluguéis de veículos.

História 2: Criar Pedido de Aluguel

Como um cliente, eu quero criar um pedido de aluguel informando veículo e período, **para que** eu possa alugar um carro.

História 3: Modificar Pedido de Aluguel

Como um cliente, eu quero modificar um pedido de aluguel antes da aprovação, **para que** eu possa corrigir ou alterar detalhes.

História 4: Avaliar Pedido

Como um agente (banco/empresa), **eu quero** avaliar um pedido de aluguel verificando a situação financeira do cliente, **para que** eu possa aprová-lo ou reprová-lo.

História 5: Gerar Contrato

Como um sistema, **eu quero** gerar automaticamente um contrato após aprovação do pedido, **para que** o cliente e agente tenham um documento formalizado.

3. Diagrama de Classes

O diagrama de classes representa a estrutura do sistema e os relacionamentos entre os elementos.

Principais Classes:

- **Cliente** (id, nome, CPF, RG, endereço, profissão, empregador, renda[])
- **Agente** (id, nome, tipo, banco ou empresa)
- **Automóvel** (id, matrícula, ano, marca, modelo, placa)
- **Pedido** (id, cliente, automóvel, período, status)
- **Contrato** (id, pedido, cliente, agente, valor, status)
- **Sistema** (responsável por gerenciar clientes, pedidos e contratos)

4. Diagrama de Pacotes

A estrutura do sistema será baseada em uma arquitetura MVC.

Pacotes:

- **Controller:** Gerencia a lógica de negócio (PedidoController, ClienteController, ContratoController)
- **Model:** Representa os dados do sistema (Cliente, Pedido, Automóvel, Contrato, Agente)
- **View:** Responsável pela interface com o usuário (páginas web dinâmicas)
- **Persistence:** Gerencia o acesso ao banco de dados (DAO para Cliente, Pedido, Contrato, Automóvel)