

# Relatório do Sistema de Gestão de Aluguer de Automóveis

## Sistemas Distribuídos

Jailson Varela-40699  
Thawila Simbine-49183

### **Introdução**

O presente relatório descreve o desenvolvimento de um Sistema de Gestão de Aluguer de Automóveis (veículos), criado para gerir eficientemente o registo, aluguer e consulta de veículos numa central de aluguer distribuída. Este sistema foi desenvolvido como parte de uma simulação académica para demonstrar conceitos de sistemas distribuídos, incorporando um **servidor central** e dois tipos de cliente: **Cliente Geral e Cliente Administrador**.

O principal objetivo do sistema é proporcionar uma solução prática para a gestão de uma frota de veículos, permitindo funcionalidades como o registo de novos veículos, aprovação de veículos para aluguer, criação de registos de aluguer e consultas detalhadas.

### **Metodologia**

O desenvolvimento do sistema seguiu uma abordagem estruturada, dividida nas seguintes etapas:

#### **1. Planeamento:**

- Identificação dos requisitos funcionais e técnicos do sistema.
- Escolha de tecnologias adequadas, como Java, PostgreSQL e middleware java RMI.

## 2. Análise :

- Modelação do sistema com base nos princípios de orientação objetos (solid).
- Separação de responsabilidades através de pacotes, como: **entidades, serviços, repositórios e exceções**, e um tipo de arquitetura que é chamada de MVC(model views controller).

## 3. Implementação:

- Desenvolvimento das funcionalidades principais e mais algumas que achamos relevantes, e tratamento de erros.
- Integração da comunicação entre cliente e servidor.

## 4. Testes:

- Testes funcionais para validar as operações e garantimos que o sistema funcione como está especificado .

# Desenvolvimento

No trabalho consigo fazer as implementações das funcionalidades mencionadas no enunciado e adicionamos mais algumas funcionalidades e estruturas de dados como:

- Estruturas de dados para :
  - Tipo estado administrativo ;
  - Tipo estado aluguer ;
  - Tipo de unidade de tempo;
  - Tipo de veículo.
- Funcionalidades como:
  - naoAprovarVeiculo - que serve justamente para mudar o estado administrativo de um veículo para Não Aprovado

- INTERFACES como:
  - Serviço Auxiliar interface;
  - Serviço cliente geral;
  - Serviço cliente administrador;
  - Serviço principais do sistema interface;

Para implementar a componente distribuída em Java decidimos usar, no nosso projeto, o Java Remote Method Invocation (Java RMI) que executa PROXIS e SKELETON relativos aos STUB E PUB.

## **Conclusão**

O Sistema de Gestão de Aluguer de Veículos demonstrou ser uma solução eficiente e escalável para a gestão de uma frota de veículos em várias localizações com estados de administração diferentes. A separação de responsabilidades entre Cliente Administrador, Cliente Geral e Servidor, o uso de middleware java RMI, a integração com uma base de dados PostgreSQL sólida e a comunicação adaptada através do terminal garantiram a qualidade da implementação.

Embora o sistema esteja funcional, algumas melhorias podem ser consideradas para versões futuras, como a implementação de autenticação de utilizadores e a integração web para facilitar o uso por operadores, apesar de já ser algo bastante agradável visualmente pelo terminal. Também, a implementação de funcionalidades para a entrega do veículo ou o término do aluguer ficaram por implementar.