



Sistema de Gestão de Aluguer de Automóveis

1. Objetivo

Desenvolver uma aplicação distribuída para simular um sistema de gestão de aluguer de veículos. O sistema deverá permitir o registo de veículos, gestão do estado de aluguer e realizar consultas e aprovações remotamente. A aplicação deverá incluir um servidor e dois clientes: um geral (para consultas e registos) e um administrativo (para gestão e aprovações), acessíveis remotamente para verificar e atualizar as informações dos veículos.

2. Descrição da Tarefa

A Central de Aluguer de Automóveis necessita de um sistema para gerir a frota de veículos disponíveis para aluguer em várias localizações. Este sistema deve permitir o registo dos veículos, o rastreamento das suas localizações e o estado atual de aluguer.

Cada veículo terá uma matrícula, modelo, tipo (ex.: compacto, SUV, camioneta), localização atual e estado de aluguer (disponível, alugado, em manutenção). Os veículos também terão um estado administrativo (aprovado ou não aprovado), que será gerido pelo cliente administrativo. Os dados de cada veículo deverão ser armazenados numa base de dados PostgreSQL.

Funcionalidades

Cliente Geral

- **Registo de um novo veículo:** Um funcionário pode adicionar um novo veículo da frota ao sistema. O veículo ficará com o estado administrativo de "não aprovado" até ser autorizado pelo administrador.
- **Registar clientes:** Permitir o registo de novos clientes no sistema, incluindo informações básicas como nome, documento de identificação e contacto.

- **Registar um novo aluguer realizado por um cliente, especificando o valor do pagamento:** Esta funcionalidade permite que um funcionário registre no sistema um aluguer efetuado por um cliente, associando o veículo alugado, o cliente, e o valor acordado para o aluguer. Esse registo deverá incluir também detalhes relevantes, como a data de início e a duração prevista do aluguer.
- **Listar veículos disponíveis para aluguer:** Com opção de filtros por localização e tipo de veículo.
- **Listar localizações de veículos alugados.**
- **Consultar o histórico de aluguer de um veículo específico,** identificado por um ID do veículo, incluindo datas, localizações dos alugueres e pagamentos recebidos.

Cliente Administrador

- **Listar veículos por estado administrativo** (aprovado/não aprovado).
 - **Aprovar um veículo** não aprovado para ser alugado.
-

3. Implementação

Requisitos Técnicos:

- Utilize uma camada de software intermediária (middleware) para facilitar a comunicação entre o cliente e o servidor, de modo a que estes possam comunicar-se sem depender de uma plataforma ou sistema operativo específico. O middleware deverá funcionar como uma ponte que permite a troca de informações entre as partes, tornando o sistema mais flexível e adaptável a diferentes ambientes.
- **(Bónus) Implementar técnicas distintas na construção dos clientes:** Utilizar técnicas distintas para cada cliente (por exemplo, diferentes tecnologias de comunicação ou bibliotecas) que demonstrem a flexibilidade e a independência do sistema face à plataforma.
- A interface será em linha de comandos, com menus para navegação das opções.
- Não é necessário implementar autenticação de utilizador nesta fase.

Configuração:

- Parâmetros de configuração devem ser externos ao código-fonte, utilizando argumentos de linha de comando ou ficheiros de propriedades (ex.: `java.util.Properties`).
 - A solução deve ser compatível com a plataforma `alunos.di.uevora.pt`, onde uma base de dados PostgreSQL deve ser configurada para registar as informações dos veículos e alugueres.
-

4. Entrega

- Enviar um ficheiro .zip no Moodle, contendo a pasta sd-t01-YYYYY-ZZZZZ (substituir YYYYY e ZZZZZ pelos números de aluno)
- A pasta deve conter o código-fonte, ficheiros de configuração, o script de execução das aplicações(makefile) e o *dump* da base de dados PostgreSQL configurada com as tabelas e dados necessários para o sistema de aluguer de veículos.
- Um relatório deve estar incluído com:
 - Identificação dos alunos.
 - **Justificação das escolhas feitas ao longo do trabalho** (ex.: escolha do middleware, estrutura das tabelas na base de dados e métodos de comunicação entre cliente e servidor)
 - Observações sobre o desenvolvimento e eventuais desafios encontrados.

Nota: Os alunos podem optar por trabalhar individualmente ou em pares.