

Prática offline 2 – Loja de Carros

1. Descrição geral

O objetivo do projeto é o desenvolvimento de um sistema cliente/servidor para gerenciamento de carros em uma loja. O sistema consiste de um mecanismo de armazenamento de carros e seus dados associados como renavan, nome, categoria, ano de fabricação, quantidade disponível e preço.

1.2. Sobre os carros

Nessa loja, os carros são agrupados em três categorias: econômico, intermediário e executivo.

Exemplos de carros econômicos: fiat novo uno, chevrolet onix, ford ka, hyundai hb20, nissan march.

Exemplos de carros intermediários: ford ka sedan, chevrolet onix plus, hyundai hb20s, renault logan, toyota etios.

Exemplos de carros executivos: toyota corolla, honda civic, chevrolet cruze, ford fusion, audi a3.

2. User Stories

User Stories são uma forma de expressar requisitos funcionais desejados para o sistema (o que o sistema deve fazer). Elas focam nos objetivos dos usuários e como eles conseguem alcançá-los. *User stories* devem ser curtas, simples e escritas sob o ponto de vista do usuário. Observe que, no mundo real, o cliente poderá mudar de ideia com respeito a esses requisitos funcionais.

User Story	Título	Descrição
1.	Adicionar carro	Um usuário pode adicionar carros ao sistema da loja. Para adicionar, os seguintes atributos são fornecidos: renavan, nome, categoria, ano de fabricação e preço. Atualizar quantidade disponível.
2.	Apagar carro	Um usuário pode apagar registros de carros da loja. Todos os atributos são removidos a partir do nome do carro ou quando a quantidade disponível chegar em zero.
3.	Listar carros	Um usuário pode listar os carros da loja (com todos os atributos). A listagem pode ocorrer por categoria ou de forma geral e

		deve ser apresentada em ordem alfabética dos nomes.
4.	Pesquisar (consultar) carro	Um usuário pode realizar uma busca por carro a partir de seu nome ou do renavan.
5.	Alterar atributos de carros	Um usuário pode alterar atributos de carros armazenados. Exemplo: um cadastro pode ter sido feito de forma errada (nome ou data de fabricação incorretos, etc).
6.	Atualizar listagem de carros enviada aos clientes conectados	Adicionar, apagar e alterar atributos de carros são operações que fazem com que o servidor tenha que atualizar os clientes.
7.	Exibir quantidade de carros	Um usuário pode consultar o sistema para saber quantos carros estão armazenados em um dado momento.
8.	Comprar carro	Um usuário pode efetuar a compra de um carro após consulta e análise de preço.

3. Detalhamento dos Requisitos.

- A relação entre as entidades Carro, Econômico, Intermediário e Executivo deve ser implementada no projeto respeitando os conceitos de POO. O mesmo deve acontecer com as entidades Cliente e Funcionário.
- Usuários podem ser clientes ou funcionários da loja (autenticação é necessária).
- Clientes podem listar, pesquisar, exibir a quantidade e comprar carros.
- Funcionários podem fazer tudo que os clientes fazem. Além disso, podem adicionar, apagar e alterar atributos dos carros.
- Clientes e funcionários devem realizar as operações listadas via conexão com um servidor. O servidor pode armazenar os dados ou utilizar um servidor auxiliar específico para guardar os dados.
 - Esquema: cliente <---> servidor.
- Doze carros, sendo três de cada tipo, devem ser adicionados no início da execução da aplicação.

3. Observações.

- O prazo para a entrega dos projetos expira em 05/04/2023 às 23:59h, via SIGAA. Portanto, certifiquem-se do arquivo que vão enviar.
- Avaliação: o projeto vale 40% da nota da 2ª unidade.
 - Metade para o código rodando corretamente.
 - Perguntas individuais podem ser feitas sobre o código e a apresentação.
- Para os que enviarem por e-mail, depois do prazo, o projeto valerá 20% a menos.
- O projeto deve ser desenvolvido de forma individual ou em dupla.
- Os trabalhos devem utilizar as tecnologias vistas, até o momento, na disciplina para desenvolver o projeto (*Threads*, *Sockets* e *RMI*).
- Correção:
 - Análise da gravação de um vídeo com a apresentação do projeto.
 - Um roteiro com as instruções necessárias será disponibilizado para ajudar na gravação.

- Sabe-se que a estrutura de projetos dessa natureza pode ser muito comum. No entanto, a lógica de funcionamento, o armazenamento e a visualização das informações da loja podem ser bem particulares. Cuidado com códigos iguais. A penalidade é a nota ZERO.