

Projeto: Sistema de Informações para Concessionária de Automóveis desenvolvido em Java.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE – UFRN

INSTITUTO METRÓPOLE DIGITAL – IMD

DIM0116 - LINGUAGEM DE PROGRAMAÇÃO II

Docente: Francimar Carlos de Macêdo

Alunos: André Augusto Fernandes & Edvaldo Dantas de Medeiros Júnior

1. Introdução

Este projeto foi desenvolvido para a disciplina de Linguagem de Programação II, como meio de avaliar os conhecimentos adquiridos pelo aluno ao longo da disciplina.

Esta etapa do projeto consiste de uma entrega parcial do projeto, com apresentação do conceito e funcionalidades implementadas até o momento.

2. Objetivo

Os requisitos da avaliação (2ª Unidade) eram desenvolver uma aplicação em JAVA na qual fosse possível aplicar os conceitos abaixo:

1. Classes e Objetos;
2. Herança;
3. Classes Abstratas;
4. Interfaces;
5. Composição;
6. Polimorfismo;
7. Tratamento de Exceção;
8. Coleções;

3. Proposta

Para nosso projeto, criamos um sistema de informações para concessionárias de automóveis, no qual será possível administrar os departamentos de:

- Recursos Humanos;
- Clientes;
- Estoques;
- Vendas.

Tais departamentos serão dispostos em forma de menus para o usuário do sistema, em que cada menu possuirá sub-menus para realizar operações específicas para cada área, como cadastrar clientes, cadastrar funcionários, abastecer estoque, realizar vendas, entre outras.

4. Instruções de Compilação e Execução/Uso

4.1. Compilação

Para compilar o programa Java na linha de comando, siga as etapas abaixo:

1. Certifique-se de ter o Java Development Kit (JDK) instalado no seu sistema. Você pode verificar isso digitando `java -version` no prompt de comando. Se o JDK estiver instalado corretamente, você verá a versão do Java sendo exibida. Certifique-se também que o make está instalado no seu sistema. Você pode verificar isso digitando `make -v` no prompt de comando. Se o make estiver instalado corretamente, você verá a versão do make sendo exibida. Caso o JDK ou o make não estejam instalados, você pode instalá-los através dos links abaixo:
 - [JDK](#)
 - [make](#)
2. Baixe os arquivos deste repositório para uma pasta local de sua preferência.
3. Abra um prompt de comando ou terminal no seu sistema operacional.
4. Navegue até o diretório raiz do projeto, onde se encontra o arquivo Makefile, e execute o seguinte comando: `make -B` O comando acima irá compilar os arquivos

.java e gerar os arquivos .class. OBS: Caso o comando acima não funcione, tente executar o seguinte comando: `javac -d bin src/*.java` A seguinte tela será exibida:

```
Windows PowerShell
PS C:\Users\arqan\Dropbox\GitHub\LP2-TRABALHO-U3\LP2-Trabalho-U3> make -B
javac -d bin -sourcepath src src/App.java
javac -d bin -sourcepath src src/Carro.java
javac -d bin -sourcepath src src/Cliente.java
javac -d bin -sourcepath src src/Funcionario.java
javac -d bin -sourcepath src src/Loja.java
javac -d bin -sourcepath src src/Pessoa.java
PS C:\Users\arqan\Dropbox\GitHub\LP2-TRABALHO-U3\LP2-Trabalho-U3> |
```

Os arquivos serão compilados na estrutura padrão de diretórios de um projeto Java:

```
|— Makefile
|— bin
|   |— (arquivos .class serão gerados aqui)
|— src
|   |— (arquivos-fonte .java aqui)
```

4.2. Execução

Para executar o programa, dirija-se ao diretório `/bin` e digite `java App` no prompt de comando.

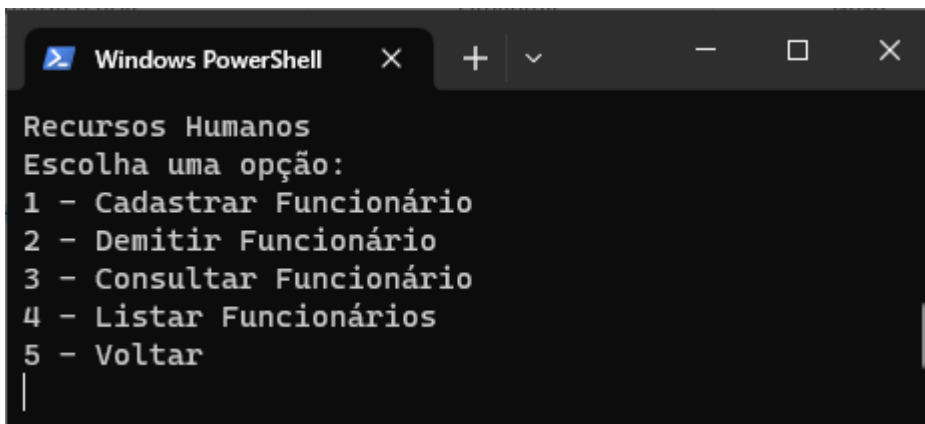
O seguinte menu será exibido:

```
Windows PowerShell
PS C:\Users\arqan\Dropbox\GitHub\LP2-TRABALHO-U3\LP2-Trabalho-U3\bin> java App
Bem vindo ao SYSCAR! Sistema de Informações para Concessionárias de Automóveis
Escolha uma opção:
1 - Recursos Humanos
2 - Clientes
3 - Abastecimento e Venda
4 - Estoque
5 - Sair
|
```

A partir do menu acima, o usuário iniciará sua jornada pelo SYSCAR - Sistema de Informações para Concessionárias de Automóveis.

4.2.1. 1 - Recursos Humanos

Ao entrar em recursos humanos, o usuário terá acesso às seguintes operações:

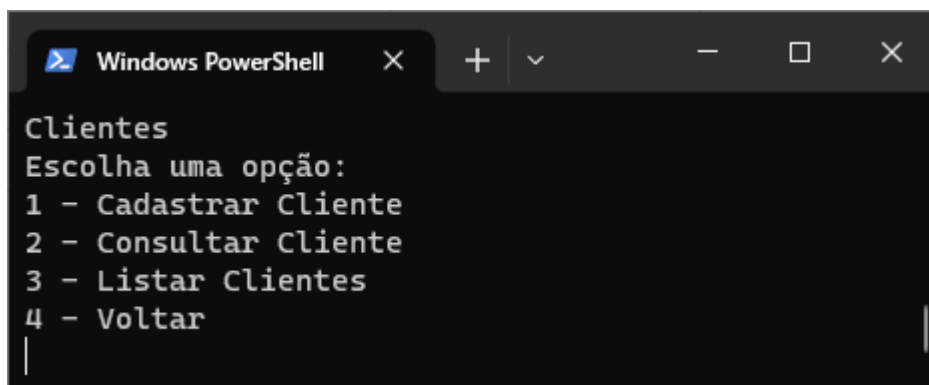
A screenshot of a Windows PowerShell terminal window. The title bar shows 'Windows PowerShell' with standard window controls. The terminal content displays a menu titled 'Recursos Humanos' followed by 'Escolha uma opção:'. Below this, there is a numbered list of five options: '1 - Cadastrar Funcionário', '2 - Demitir Funcionário', '3 - Consultar Funcionário', '4 - Listar Funcionários', and '5 - Voltar'. A vertical cursor is positioned at the end of the fifth option.

Aqui o usuário poderá:

- Cadastrar Funcionário: As informações do funcionário serão passadas pelo usuário ao programa e, em seguida, serão colocadas em um objeto funcionário, para então serem inseridas em uma lista no objeto loja.
- Demitir Funcionário: A instância de funcionário cuja matrícula for fornecida pelo usuário será excluído da lista funcionários em loja.
- Consultar Funcionário: A instância de funcionário cuja matrícula for fornecida pelo usuário será exibida na tela. Caso não exista um funcionário com a matrícula fornecida, o programa exibirá uma mensagem de erro.
- Listar Funcionários: Todos os funcionários cadastrados serão exibidos na tela. Caso não exista nenhum funcionário cadastrado, o programa exibirá uma mensagem de erro.

4.2.2. 2 - Clientes

Ao entrar em clientes, o usuário terá acesso às seguintes operações:

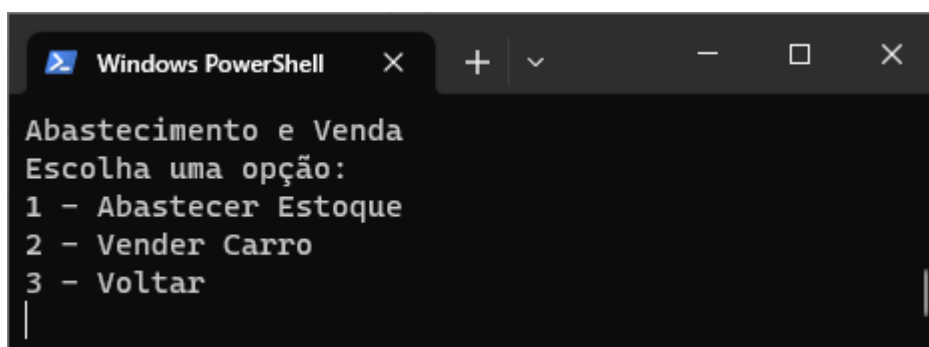


```
Windows PowerShell
Clientes
Escolha uma opção:
1 - Cadastrar Cliente
2 - Consultar Cliente
3 - Listar Clientes
4 - Voltar
```

- Cadastrar Cliente: Aqui, o sistema irá solicitar ao usuário a matrícula do funcionário que está cadastrando o cliente, para que seja possível realizar a operação. O programa irá verificar a lista de funcionários da loja e, caso a matrícula fornecida não exista, o programa exibirá uma mensagem de erro. Caso a matrícula exista, o programa irá solicitar as informações do cliente e, em seguida, irá colocá-las em um objeto cliente, para então serem inseridas em uma lista no objeto loja.
- Consultar Cliente: A instância de cliente cujo cadastro for fornecido pelo usuário será exibida na tela. Caso não exista um cliente com o CPF fornecido, o programa exibirá uma mensagem de erro.
- Listar Clientes: Todos os clientes cadastrados serão exibidos na tela. Caso não exista nenhum cliente cadastrado, o programa exibirá uma mensagem de erro.

4.2.3. 3 - Abastecimento e Vendas

Ao entrar em abastecimento e vendas, o usuário terá acesso às seguintes operações:



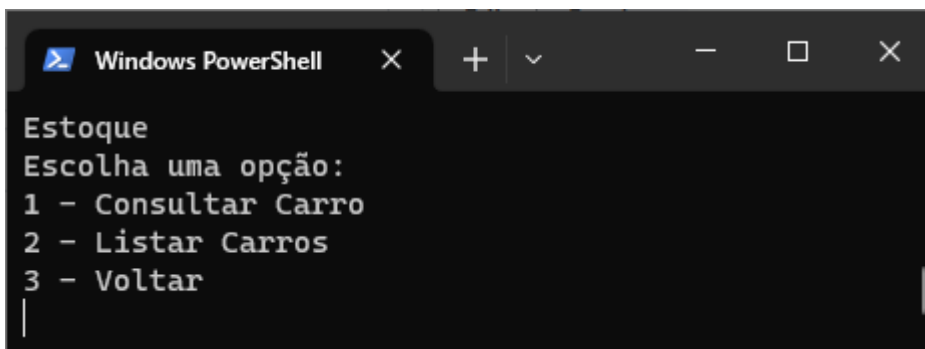
```
Windows PowerShell
Abastecimento e Venda
Escolha uma opção:
1 - Abastecer Estoque
2 - Vender Carro
3 - Voltar
```

- Abastecer Estoque: O usuário poderá abastecer o estoque da loja com novos automóveis. As informações do automóvel serão passadas pelo usuário ao programa e, em seguida, serão colocadas em um objeto carro, para então serem inseridas em uma lista no objeto loja.

- Vender Automóvel: O usuário poderá vender um automóvel para um cliente. O programa irá solicitar a matrícula do cliente e o chassi do automóvel. Caso o cliente não exista, o programa exibirá uma mensagem de erro. Caso o automóvel não exista, o programa exibirá uma mensagem de erro. Por fim, o programa irá exibir uma mensagem de sucesso e o automóvel será removido da lista de automóveis da loja e adicionado à lista de automóveis do cliente.

4.2.4. 4 - Estoque

Ao entrar em estoque, o usuário terá acesso às seguintes operações:

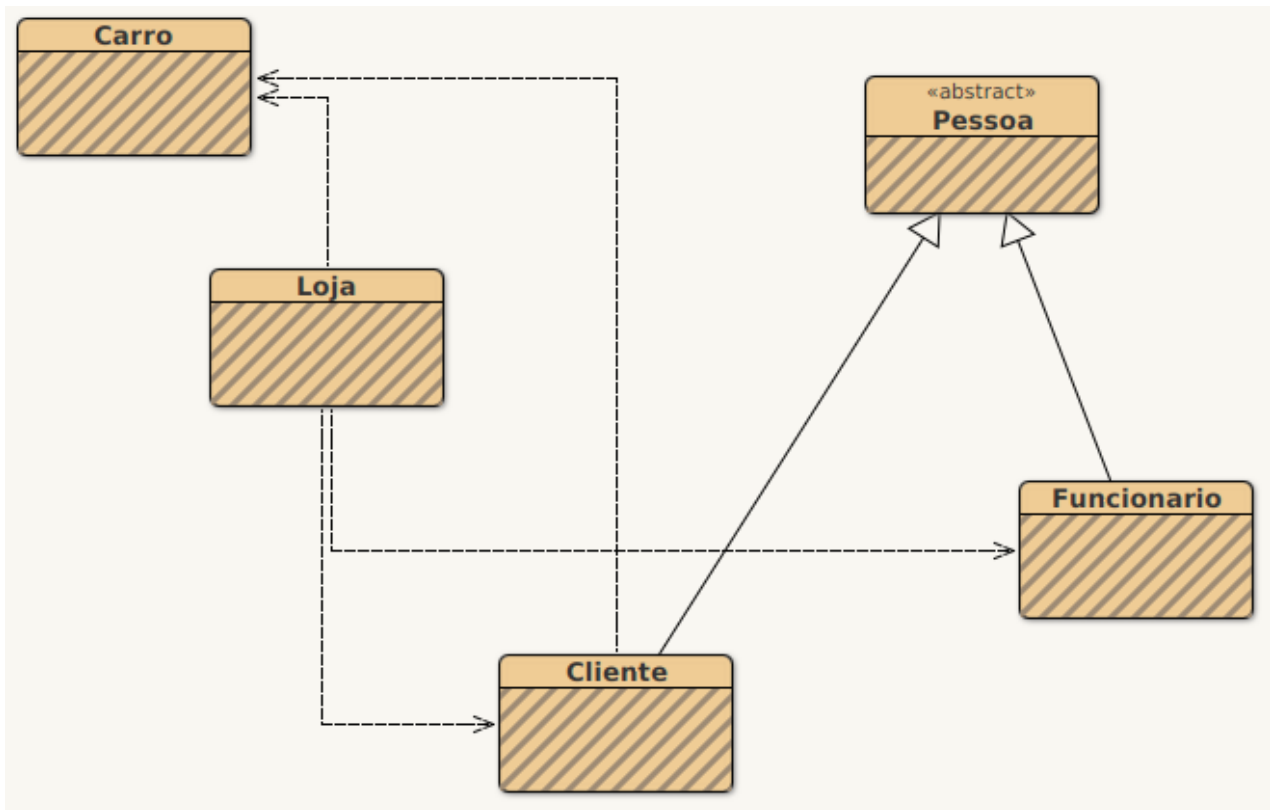


```
Windows PowerShell
Estoque
Escolha uma opção:
1 - Consultar Carro
2 - Listar Carros
3 - Voltar
|
```

- Consultar Automóvel: O usuário poderá consultar um automóvel pelo seu modelo. Caso o automóvel não exista, o programa exibirá uma mensagem de erro. Caso o automóvel exista, o programa exibirá as informações do automóvel.
- Listar Automóveis: Todos os automóveis cadastrados serão exibidos na tela. Caso não exista nenhum automóvel cadastrado, o programa exibirá uma mensagem de erro.

5. Implementação

O projeto possui a seguinte estrutura de classes:



- Pessoa: Pessoa é uma classe abstrata com atributos e métodos genéricos de uma pessoa física;
- Cliente, Funcionários: são classes herdeiras de Pessoa, com especializações próprias. Cliente possui uma coleção do tipo List<> para armazenar seus automóveis. Já funcionário, além de seus atributos especializados, possui também métodos próprios para fazer transações ordinárias da loja;
- Carro: classe que será utilizada para instanciar os automóveis da loja e dos clientes,;
- Loja: classe utilizada para instanciar o objeto loja. Loja possuirá coleções do tipo List<> para clientes, funcionários e carros, além de métodos para manipular seus atributos e coleções;
- App: Classe que contem o main, no qual será executado o programa.

6. Tecnologias utilizadas

Para desenvolver o sistema foi utilizada a linguagem de programação Java e o editor de código VS Code.

Para compilar o programa, foi utilizado o compilador javac e o make.

7. Conclusão

O projeto se encontra em estágio avançado de desenvolvimento e, até o momento, foi possível aplicar os conceitos de classes e objetos, herança, classes abstratas, interfaces, composição, polimorfismo, tratamento de exceção e coleções.