Relatório A3

Nome: Gabriel de Santana Coimbra Pereira RA: 12724145259

Nome: Wesllen dos Santos Correia RA: 12724148801

Nome: Esdras Alves de Andrade Silva RA: 1272320690

Nome: Alisson da silva Lopes RA: 12723211997

Nome: Matheus Dias Marinho RA: 12724161117

Introdução

O tema do projeto escolhido pelo grupo é um sistema de concessionária, que, apesar de nosso contexto ser uma elaboração simples do que poderia ser um sistema real, foi escolhido por estar alinhado com os objetivos do trabalho e ser facilmente compreendido pela maioria das pessoas. O principal objetivo do projeto é desenvolver um sistema CRUD (Create, Read, Update, Delete) utilizando a linguagem Java. Para alcançar esse objetivo, decidimos utilizar os ambientes de desenvolvimento Visual Studio Code e IntelliJ IDEA.

O código foi estruturado em dez classes principais, cada uma com um papel específico no sistema. A seguir, detalhamos o propósito de cada classe:

* **Principal.java**: Esta classe contém a função main(), que é o ponto de entrada do programa. Nela inicializamos o que será nosso banco de dados, representado por um ArrayList<>, e instanciamos as entidades iniciais conforme os requisitos do trabalho.
* **Menu.java**: A classe Menu é responsável por apresentar o menu de opções ao usuário, permitindo que ele interaja com o sistema de forma intuitiva e saiba quais operações estão disponíveis. Além disso, instanciamos os objetos das classes Controle que possuem os métodos de suas respectivas entidades.
* **Pessoa**.java: A classe pessoa é a Super Classe das classes Cliente.java e Funcionario.java.
* **Cliente.java**: Representa a entidade cliente, contendo os atributos e métodos (Getters e Setters) necessários para manipular os dados dos clientes.
* **Funcionario.java**: Representa a entidade funcionário, com atributos e métodos (getters e setters) para gerenciar os dados dos funcionários da concessionária.
* **Veiculo.java**: Representa a entidade veículo, incluindo os atributos e métodos (getters e setters) para tratar os dados dos veículos.
* **ControleCliente.java**: Esta classe é responsável por implementar os métodos relacionados à entidade cliente, tais como adicionar, atualizar, remover e listar clientes.
* **ControleFuncionario.java**: Esta classe é responsável por implementar os métodos relacionados à entidade funcionário, tais como adicionar, atualizar, remover e listar funcionários.
* **ControleVeiculo.java**: Esta classe é responsável por implementar os métodos relacionados à entidade veículo, tais como adicionar, atualizar, remover e listar veículos.
* **ControleVenda.java**: Responsável pela operação de venda de veículos, esta classe relaciona as três principais entidades do projeto (cliente, funcionário e veículo).

Fundamentação Teórica

O projeto foi desenvolvido na linguagem Java, fazendo uso extensivo da Programação Orientada a Objetos (POO). No contexto deste projeto, que visa criar um sistema para uma concessionária, os principais conceitos de POO aplicados foram:

* **Classes**: São a base da POO, representando a estrutura e o comportamento de um tipo de objeto. Uma classe é uma abstração do objeto, enquanto o objeto é uma instância da classe.
* **Herança**: Permite que uma classe herde atributos e métodos de outra classe, promovendo reutilização de código e hierarquia.
* **Encapsulamento**: É o princípio de agrupar dados e métodos que operam sobre esses dados em uma única unidade, ou classe. Ele esconde os detalhes internos e só expõe o que é necessário, melhorando a modularidade e a segurança.
* **Abstração**: Consiste em representar as características essenciais de um objeto, ignorando detalhes específicos. Facilita a compreensão e modelagem de sistemas complexos.

Java é uma linguagem reconhecida por sua robustez, segurança e portabilidade. Esses atributos a tornam ideal para o desenvolvimento de sistemas, como o nosso sistema de concessionária.

* **Tratamento de Exceções**: Java possui um sistema robusto para tratamento de erros, obrigando o desenvolvedor a lidar com possíveis exceções e tornando o código mais seguro e confiável.
* **Segurança**: Java é fortemente tipada, reduzindo erros de tipo.
* **Portabilidade**: Java é executado em qualquer dispositivo que tenha a JVM(Java Virtual Machine), permitindo que o código seja executado em diferentes plataformas sem modificações.

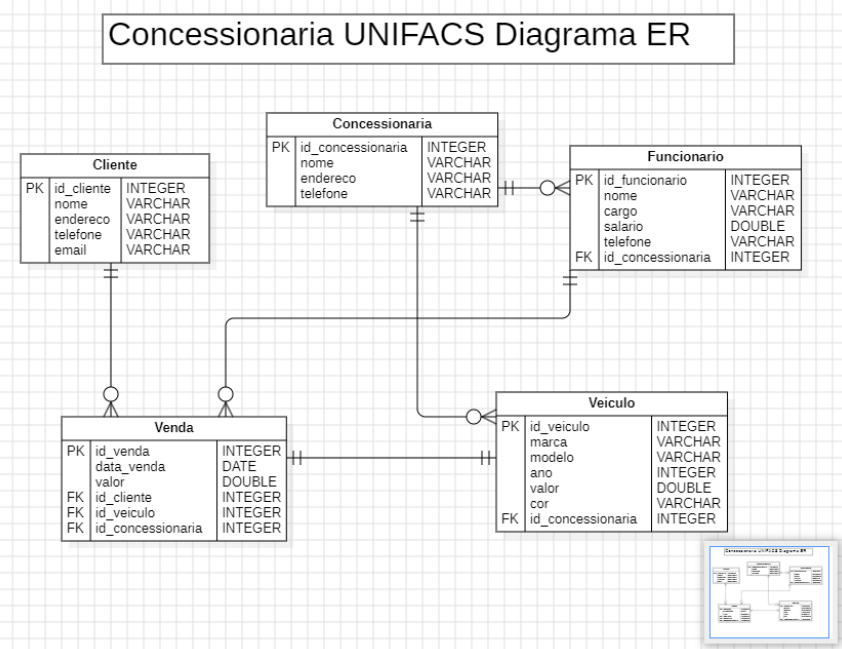
As IDEs Utilizadas para o desenvolvimento do projeto, foram utilizadas as seguintes IDEs (Integrated Development Environments):

* **Visual Studio Code**: Um editor de código-fonte que suporta várias linguagens, incluindo Java. Possui integração com o Git e diversas extensões que facilitam o desenvolvimento.
* **IntelliJ IDEA**: Uma IDE amplamente utilizada para desenvolvimento em Java. Oferece ferramentas para encontrar e corrigir erros, além de suporte a plugins que melhoram a produtividade do desenvolvedor.

O projeto é um sistema CRUD (Create, Read, Update, Delete), que são as operações fundamentais para o funcionamento de diversas aplicações, especialmente na web.

* **Create (Criar)**: Adicionar novos registros ao sistema, como novos clientes ou veículos.
* **Read (Ler)**: Exibir dados existentes no sistema, permitindo a visualização de informações sobre clientes, funcionários e/ou veículos.
* **Update (Atualizar)**: Modificar dados já existentes no sistema, como atualizar informações cliente.
* **Delete (Deletar)**: Remover registros do sistema, como excluir um cliente que não utiliza mais os serviços da concessionária.

Projeto de implementação



Esquemático feito usando o software de modelagem StarUML.

O esquemático nos mostra as relações entre as entidades do projeto e a concessionária.

O cliente pode fazer várias compras por meio da venda, mas uma venda é somente atruída a um cliente. Portanto, a relação é (1, N).

A concessionária ela pode ter vários funcionários mas o funcionário só trabalha para uma concessionária. Portanto, a relação é (1, N).

A concessionária ela pode ter vários veículos mas o veículo só pertence a uma concessionária. Portanto, a relação é (1, N).

Um funcionário realiza várias vendas mas uma venda só é feita por um funcionário. Portanto, a relação é (1, N).

Um veículo só é vendido por uma venda. Portanto, a relação é (1, 1).

Considerações finais

Apesar da simplicidade do projeto, ainda foi desafiador criar um sistema CRUD, desenvolver noções de projeto com método MVC e fazer uma tratativa de erros. Futuramente seria possível aplicar uma interface gráfica e utilizar outro banco de dados ao invés do ArrayList. Portanto, esse sistema nos deu uma visão de como poderia funcionar um sistema real de uma concessionária.

Bibliografia

<https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos>

<https://www.dio.me/articles/vantagens-e-desvantagens-da-linguagem-java>

https://www.treinaweb.com.br/blog/principais-ides-para-desenvolvimento-java

https://querobolsa.com.br/revista/crud-o-que-e#:~:text=O%20que%20%C3%A9%20um%20sistema,atualiza%C3%A7%C3%A3o%20e%20exclus%C3%A3o%20de%20dados.