

Relatório sobre a implementação

1. Resumo

Tendo em vista um ecossistema de compra e venda de alimentos, pesquisou-se sobre o tema, a fim de programar um código funcional em Java. Para tanto, foi necessário juntar conhecimentos e experiências em Java, POO UML. Realizou-se, então, a confecção das classes e seus métodos e atributos. Diante disso, foi obtido um código em POO linguagem Java que permite todas as operações necessárias ao funcionamento de tal projeto desde a compra até a entrega de produtos passando por todas as múltiplas etapas medianas

2. O problema

O nosso sistema basicamente simula todos os passos na relação entre cliente e loja, dentro do tema e problemática já citados, a transformação das partes integrantes em objetos abstratos facilita muito a solução do problema e faz com que a programação seja mais orgânica e intuitiva, desse ponto de vista o uso de objetos com certeza melhora e facilita muito a vida do programador, permitindo também com que o usuário tenha acesso a um sistema de forma mais rápida e dinâmica, mas, nem tudo são flores a área do POO em si tem sim dificuldades a serem superadas, os programas em POO em geral são mais lentos do que programas imperativos por exemplo, graças a complexidade da representação dos objetos, e também tendem a ser menos eficientes graças a todos os desvios gerados na execução diferente da programação imperativa onde a solução é com poucas exceções linear e ocorre em poucos contextos, porém vale citar que esses problemas estão cada vez mais insignificantes graças ao poder computacional das máquinas modernas que fazem a diferença ser imperceptível.

3. Decisões

A sigla que escolhemos para o nosso projeto foi SUS, Sistema de usuário Satisfeito, as classes utilizadas foram: Cozinha, FormaPagamento, FotoProduto, Produto, Restaurante, Usuario, Pedido, ItemPedido, StatusPedido, Grupo, Permissao, Endereco, Cidade, Estado, GerarSenha, Responsavel.

As funções implementadas são: Todos os Getters, Setters e Construtores, addItem, delItem, atualizarFotoProduto, definirGrupos, cadastrarProduto, encontrarProduto, ativarProduto, desativarProduto, definirResponsavel.

O método de divisão de trabalho foi simples e intuitivo com cada pessoa do grupo tomando para si a responsabilidade sobre a parte que mais se sentia confiante.

4. Material e método

Toda a programação das classes foi feita utilizando o NetBeans IDE, e o Visual Studio Code, a linguagem escolhida como já citado foi o Java e o paradigma POO. Não foram utilizadas bibliotecas próprias, No entanto na programação das classes foram usadas classes pré existentes pertencentes ao ecossistema de bibliotecas do Java, foram elas: ArrayList, BigDecimal, JavaDate, BigDecimal, LocalDate, as classes pré existentes mostram o poder da linguagem Java e facilitaram em muito o desenvolvimento do projeto.

5. Resultados

Ao final do projeto tínhamos um programa Java que emulava perfeitamente (no que compete a estudantes da disciplina de POO) um ecossistema de compra e venda em Java. Seguem imagens do funcionamento do programa em uma execução das partes mais importantes do código:

Começamos declarando um Usuario e um Restaurante:

```
Usuario U1 = new Usuario("Pedro", "pedo@email.com", "minha-senha");
Restaurante R1 = new Restaurante("RangoLegal", "5.50", true, true,
```

Depois cadastramos alguns produtos:

```
R1.cadastrarProduto("X-burger", "Carne, queijo", "15.9", 21, "1", Restaurante.TipoCozinha.Americana);
R1.cadastrarProduto("X-salada", "Carne, queijo, salada", "19.9", 21, "2", Restaurante.TipoCozinha.Americana);
R1.cadastrarProduto("Coca-Cola Lata", "", "5", 11, "3", Restaurante.TipoCozinha.SCD);
R1.cadastrarProduto("Coca-Cola Zero Lata", "", "5", 11, "4", Restaurante.TipoCozinha.SCD);
R1.cadastrarProduto("Guarana Lata", "", "5", 11, "5", Restaurante.TipoCozinha.SCD);
```

Criamos um pedido:

```
Pedido Ped = new Pedido(R1, U1);
```

E adicionamos itens a esse pedido:

```
Ped.addItem("Guarana Lata", 1, "");
Ped.addItem("X-salada", 2, "");
Ped.delItem("X-salada", 1, "apenas 1");
```

Logo depois imprimimos o subtotal com o valor devido:

```
System.out.println("Subtotal:" + Ped.getSubtotal());
```

Depois de todas essas operações temo como saída a seguinte mensagem:

```
PS C:\Users\maic_OneDrive\Área de Trabalho\ProjetoPOO (1)> c:; cd 'c:\
ensioens\redhat.java-1.4.0\jre\17.0.2-win32-x86_64\bin\java.exe' '-XX:+S
ojettoPOO (1)\ProjetoPOO\target\classes' 'Projeto.Main'
Subtotal:24.9
```

Com o término da programação podendo testar e compreender toda a jornada e o funcionamento do código como um todo, com certeza todos os estudantes envolvidos se sentem mais preparados para os desafios a frente, não só na vida acadêmica, como também na vida profissional.