Challenge Natura

Sprint 01 - Programação Orientada a Objetos com Java e Web

Turma – 2SIS

Nome do Grupo: Tech Trend

Integrantes:

Enzo Luciano Duarte – **RM**: 552486

Francisco Henrique Lima – **RM**: 99545

João Victor Oliveira Avellar – **RM**: 550283

Murilo Santini Chequer – **RM**: 550198

Ronaldo Kozan Júnior – **RM**: 98865

São Paulo

2024

1. Descrição do problema:

O problema abordado é o desengajamento das consultoras da Natura, que muitas vezes enfrentam dificuldades devido à fragmentação das ferramentas digitais disponíveis.

Atualmente, é necessário o uso de três aplicativos, são eles:

- “Natura: Perfumes e Cosméticos";
- "Minha Consultoria";
- "Emana Pay.

Com a unificação de todos os aplicativos, não só simplificará o processo de vendas, mas também aumentará o engajamento das revendedoras, ao proporcionar uma plataforma centralizada, e principalmente, fácil de usar.

2. Funcionamento do Programa:

O programa inicia com o cadastro de clientes e consultoras na **classe SuperApp**. Os clientes se cadastram fornecendo nome, endereço e outros dados necessários. As consultoras também se cadastram, informando seu nome, endereço e outros detalhes.

As consultoras têm a função de mostrar os produtos disponíveis para os clientes. Elas acessam o SuperApp para listar os produtos que estão oferecendo.

Quando um cliente decide comprar algum produto, ele entra em contato com a consultora de sua escolha. A consultora mostra os produtos disponíveis e o cliente escolhe o que deseja comprar. O cliente realiza o pedido, informando a consultora escolhida e os produtos desejados. Esse pedido é registrado no SuperApp.

Cada pedido é registrado na **classe Pedido** com as informações do cliente, consultora e produtos selecionados. O pedido é adicionado à lista de pedidos na **classe SuperApp**.

Após a realização do pedido, o estoque dos produtos é atualizado. O método **atualizarEstoque()** da **classe Produto** é chamado para reduzir a quantidade disponível do produto pedido.

Os clientes podem visualizar seus pedidos através do método **visualizarPedidos()** da **classe Cliente**. As consultoras também podem ver os pedidos recebidos através do método **visualizarPedidos()** da **classe ConsultoraDeBeleza**.

2.1 - Interconexão das Classes:

As **classes Cliente**, **ConsultoraDeBeleza**, **Produto** e **Pedido** estão todas conectadas através da **classe SuperApp**. Os métodos públicos dessas classes são utilizados para interagir entre si e realizar as operações necessárias para a venda dos produtos.

3. Identificação dos objetos envolvidos na aplicação, ou seja, aqueles que serão utilizados na resolução do problema:

Classes: Consultora, Cliente, Produto, Pedido e SuperApp.

4. Identificação das Classes e seus atributos:

Classe: Consultora

Atributos:

- - IdConsultora : int,
- - nomeConsultora : string,
- - enderecoConsultora : string,
- - idadeConsultora : int,
- - pontuacaoConsultora : int,
- - prodDisponiveis : list

Classe: Cliente

Atributos:

- - idCliente : int,
- - nomeCliente : string,
- - enderecoCliente : string,
- - idadeCliente : int,
- - listaConsultorasFavoritas : string

Classe: Produto

Atributos:

- - idProduto : int,
- - nomeProduto : string,
- - descricaoProduto : string,
- - preco : double,
- - categoria : string,
- - estoque : int

Classe: Pedido

Atributos:

- - idPedido : int,
- - cliente : Cliente,
- - consultora : Consultora,
- - produtos : list,
- - data : datetime

Classe: SuperApp

Atributos:

- - clientes : list
- - consultoras : list
- - produtos : list
- - pedidos : list

5. DIAGRAMA DE CLASSES: