Teste DevJr Java-Spring SafeWay: Vinícius Vilas Boas Silva 20/01/2023

1- Clone o projeto em sua máquina e descreva os erros que você encontrou.
Obs.: Os erros podem ser desde código, estrutura de dados, boas práticas, experiência do usuário e regras de negócio.
2- Descreva como se estivesse repassando os ajustes para um programador.
3- Em caso de erros na regra de negócio, faça um relato para a empresa que solicitou o sistema, neste relato deve ser informando o erro e porquê acontece o erro.
4- Faça o máximo de ajustes no código, de forma que as falhas sejam corrigidas. Siga a seguinte ordem para o ajuste: regra de negócio, código, boas práticas, estrutura de dados e experiência do usuário.
5- Suba os ajustes no seu github (caso tenha feito apenas os descritivos, por favor desconsiderar).

- Estrutura geral da aplicação:
  - ✔ Classe *Main*:
    - Será responsável unicamente por "orquestrar" as chamadas da aplicação;
  - ✔ Pacote de Entidades:
    - Conterá as entidades da aplicação, ou seja, as classes que modelam os principais pontos das regras de negócio;
    - As Entidades não serão anêmicas, ou seja, possuirão comportamento específico de acordo com suas atribuições;
  - ✔ Pacote de Repositório:
    - Conterá os *DAOs* de cada entidade;
    - Implementará as lógicas de escrita e leitura nos *DataStores*;
  - ✔ Pacote de lógica:
    - As classes deste pacote serão responsáveis pela aplicação correta das regras de negócio;
  - ✔ Pacote de *Utils*:
    - Conterá classes responsáveis por apresentar as opções específicas na tela;

- Funcionamento geral da aplicação:
  - ✔ Classe *Main* chama os métodos das classes de *Utils*, de acordo com os parâmetros iniciais da aplicação e de acordo com o tipo de usuário que está logado;
  - ✔ De acordo com as escolhas do usuário as classes de lógica serão chamadas;
  - ✔ As classes de lógica manipulam as entidades, por exemplo: listam vendas, listam produtos, criam vendas, etc.
  - ✔ As classes de lógica também persistem essas mudanças, ou seja, são responsáveis pela leitura e escrita na *DataStore*;
  - ✔ As classes de repositório fornecem uma API para manipulação dos dados;

- Regras de negócio:
  - ✔ Corrigir o cálculo de saldo das empresas:
    - O saldo da empresa está considerando o valor total da venda, incluindo a taxa de comissão do sistema. É necessário subtrair a taxa de comissão do sistema na hora de adicionar o saldo da empresa;
- Separar classes por pacotes:
  - ✔ Adicionar um pacote para Entidades;

- ✔ Adicionar um pacote para lógicas ou casos de uso;
- ✔ Adicionar um pacote para os Repositórios das Entidades;
- ✔ Adicionar um pacote de *Utils* para funcionamentos do sistema;

➢ Lógica está toda na classe *Main*, separar;
- ✔ Lógica de Usuário: separar o Login e as opções de usuário;
- ✔ Lógica de Empresa: separar as ações de empresa;
- ✔ Lógica de Venda: separar o método *criarVenda*;
  - Como este é um comportamento da classe Venda, o método *criarVenda* deveria fazer parte dela, sendo chamado por uma classe do pacote de lógica no momento em que o usuário define, de fato, fechar a venda;

➢ Entidades anêmicas;
- ✔ As entidades deveriam possuir comportamentos próprios, separados da lógica de funcionamento do sistema;

➢ Modificação de Entidades;
- ✔ Método *criarVenda* dentro da Entidade Venda;
- ✔ Criação de uma nova Entidade: Carrinho;
  - Essa entidade servirá de intermediário para a seleção de produtos do usuário e a criação de uma venda;
  - Isso permitirá a expansão e extensão da aplicação no futuro, por exemplo: salvando o carrinho do usuário entre sessões, melhorando a usabilidade do sistema e a experiência do usuário;
- ✔ Estender Cliente e Empresa de Usuário:
  - Isso vai permitir remover os campos Cliente e Empresa de Usuário, fazendo melhor o uso de herança, e reformulando a lógica de login para auxiliar na legibilidade do código da aplicação;

➢ Inserir repositório para Entidades e deixá-los encapsulados em classes específicas de lógica;

➢ Boas práticas:
- ✔ Declarar parâmetros de métodos e construtores como final;
- ✔ Remover diversos condicionais de validações da classe *Main* do código e inseri-los nos contextos das lógicas e das entidades;
- ✔ Colocar o código em Inglês;

➢ Os chamados de *println* deveriam ser chamadas para *endpoints web*;