**Documentação breve de alguns pontos implementados no Teste**

**Fluxo do cliente**

* **Empresas fora de ordem na hora de realizar a compra, na minha visão isso prejudica a experiência do usuário**

**2 - SafeWay Padaria**

**1 - Level Varejo**

**3 - SafeWay Restaurante**

**Causa do problema para o Programador:**

O problema que ocorre na saída das empresas fora de ordem no console está relacionado com o stream. O método forEach em um stream não garante a ordem dos elementos.

**Correção:** Para garantir a ordem das empresas ao imprimir no console, podemos usar um for loop em vez de forEach antes de imprimir.

System.*out*.println("Para realizar uma compra, escolha a empresa onde deseja comprar: ");  
for (Empresa empresa : empresas) {  
 System.*out*.println(empresa.getId() + " - " + empresa.getNome());  
}

* **Criação de uma nova classe chamada: Util**.

Para acoplar todos métodos necessários na aplicação para melhorar a reusabilidade, legibilidade e facilidade de manutenção em um futuro.



* **Criações de novos métodos chamados: *escolhaProdutos*,** **mostrarMeusProdutos, criarVenda, mostrarComprasEfetuadas, mostrarVendasEfetuadas, inicializarEmpresas, inicializarProdutos, inicializarClientes, inicializarUsuarios dentro da Util.**

Para para melhorar a reusabilidade, legibilidade e facilidade de manutenção em um futuro**.**

**Fluxo da Empresa**

* **Estoque da empresa não estava sendo atualizado após a compra dos clientes.**

Após verificação no fonte, foi implementado essa lógica dentro do método criarVenda e também foi implmentado uma validação para o estoque(que deve ser melhorada).

**Trecho implementado:**

Double total = 0.0;  
boolean estoqueSuficiente = true;  
  
for (Produto produto : carrinho) {  
 total += produto.getPreco();  
 for (Produto produtoEstoque : produtos) {  
 if (produtoEstoque.getId().equals(produto.getId())) {  
 // Update quantity in stock  
 int novaQuantidade = produtoEstoque.getQuantidade() - 1;  
 produtoEstoque.setQuantidade(novaQuantidade);  
 break;  
 }  
 }  
}  
  
for (Produto produto : carrinho) {  
 for (Produto produtoEstoque : produtos) {  
 if (produtoEstoque.getId().equals(produto.getId()) && produtoEstoque.getQuantidade() <= 0) {  
 estoqueSuficiente = false;  
 break;  
 }  
 }  
 if (!estoqueSuficiente) {  
 System.*out*.println("Desculpe, não há estoque suficiente para um ou mais produtos no carrinho.");  
 return null;  
 }  
}

* **A empresa deve vender apenas produtos que ela esteja relacionada**

Isso resolvido com uma implementação que Verifica se o produto pertence à Empresa escolhida que Itera sobre a lista de produtos para verificar se o produto escolhido pertence à empresa selecionada. Se encontrar um produto correspondente, ele é adicionado ao carrinho. Se não encontrar ou se o usuário escolher '0', a execução é interrompida. Se não encontrar e a escolha não for '0', exibe uma mensagem de erro.

**Trecho implementado:**

for (Empresa empresa : empresas) {  
 System.*out*.println(empresa.getId() + " - " + empresa.getNome());  
}  
  
Integer escolhaEmpresa = sc.nextInt();  
  
// Mostrar produtos da empresa escolhida  
Util.*mostrarProdutos*(produtos, escolhaEmpresa);  
  
Integer escolhaProduto;  
boolean produtoEncontrado = false;  
  
do {  
 System.*out*.print("Escolha um produto (ou 0 para sair): ");  
 escolhaProduto = sc.nextInt();  
  
 // Verificar se o produto pertence à empresa escolhida  
 for (Produto produto : produtos) {  
 if (produto.getId().equals(escolhaProduto) && produto.getEmpresa().getId().equals(escolhaEmpresa)) {  
 carrinho.add(produto);  
 produtoEncontrado = true;  
 break;  
 }  
 }  
  
 if (!produtoEncontrado && escolhaProduto != 0) {  
 System.*out*.println("Produto não encontrado ou não pertence à empresa escolhida.");  
 }  
  
} while (escolhaProduto != 0 && !produtoEncontrado);

* **O saldo da empresa deve ser alterado já refletindo as taxas**

Isso não estava ocorrendo, o saldo da empresa não considerava a taxa. Fiz um ajuste no método **criarVenda**.

**Trecho implementado:**

Double comissaoSistema = total \* empresa.getTaxa();  
Double valorLiquido = total - comissaoSistema;  
  
int idVenda = vendas.isEmpty() ? 1 : vendas.get(vendas.size() - 1).getCódigo() + 1;  
Venda venda = new Venda(idVenda, carrinho.stream().toList(), total, comissaoSistema, empresa, cliente);  
empresa.setSaldo(empresa.getSaldo() + valorLiquido);  
vendas.add(venda);  
return venda;