**Pedro Henrique e Allan Teixeira**

**Módulo de Vendas**

Esse documento contém informações da criação dos módulos de vendas pelos alunos Allan e Pedro na matéria Tópicos em Desenvolvimento de Software com o auxilio do professor.

**Prof.:** Saulo Henrique Cabral Silva

**Ouro Branco, 28 de setembro de 2020.**

APRESENTAÇÃO

Nesse breve documento de texto, iremos apresentar características e aspectos de um trabalho feito pelos alunos Allan e Pedro do IFMG. O trabalho foi feito pelo auxílio do professor.

DOCUMENTAÇÃO

**Descrição do desenvolvimento do trabalho:**

O trabalho contempla uma linguagem um pouco complexa, pela sua criação ser de origem do professor, que pelo qual, fez vídeo aulas para nós pela plataforma Moodle, onde tivemos que acompanhar o raciocínio feito pelo mesmo. Por isso, o trabalho teve uma densidade grande, levando em consideração esses aspectos. O programa está na linguagem Java, feito especificamente no NetBeans. No início o trabalho parecia bastante simples e prático, mas com a efetuação do código, verificamos o quão complexo é, mas que foi instruído pelo professor.

**Descrição dos métodos:**

**Classe CompraGUI:**

**codTxtKeyTyped:** Esse método faz com que quando o usuário aperta a tecla "enter", o programa verifica se existe informação no campo código, depois avalia se o dado é numérico. Depois que o programa recebe o código, ele avalia se o dado informado e numérico, por ser o único tipo aceito. Ao concluir isso, o programa efetua a busca do produto com o código informado na GUI, e avalia se existe o produto com esse código. Se existir, ele atualiza os campos com dados do produto, caso o produto não esteja cadastrado, emite uma mensagem de "Produto não cadastrado".

**quantTxtKeyRealeased:** O usuário informa a quantidade a ser comprada, e esse método calcula o preço parcial desses produtos, exibindo-o no painel na área "Preço Parc.".

**addProdutoBtnMouseClicked:** Esse método avalia se o produto pode ser adicionado na tabela. Se adicioná-lo, atualiza o valor da compra do cliente e limpa as informação para que um produto seja adicionado. Depois disso adiciona o produto vendido à tabela.

**RemoverBtnActionPerformed:** Esse método remove algum produto listado na tabela, onde o mesmo já tinha sido adicionado ao carrinho. Ao clicar em remover ele pede a senha do gerente para que prossiga a remoção.

**verificaQuantidade:** Verifica se a quantidade de itens informada pelo usuário tem disponível no estoque, senão a venda não é realizada.

**FinalizarBtnActionPerformed:** Finaliza a compra do usuário, descontando o número de itens do carrinho de compra inicial no estoque e exibindo o valor a ser pago.

**atualizaQuantidades:** Esse método atualiza as informações da tabela e o valor da compra com os produtos no carrinho com base no edição da tabela por parte do usuário.

**Classe FakeBancoDados**

**cargaArquivo:** Esse método é responsável para ler e interpretar o arquivo.cvs, alocando-o no banco de dados falso (memória RAM).

**consultaProdutoCod:** Com esse método, ao enviarmos o código do produto, ele faz a busca do produto com base no código apresentado.

**atualizaArquivo:** Toda vez que clicamos em finalizar compra, todos os itens selecionados são descontados no banco de dados falso, e atualiza o arquivo, de acordo com o número de itens por cada produto.

**Classe JanelaPrincipal**

**meuInitComponents:** Nesse método é atualizado e formatado a barra de rolagem do painel principal, adaptando-a com base na necessidade da janela flutuante. Além de colocar o painel compra ao iniciar o programa em exibição para o usuário.

**efetuaTransicao:** Com esse método é possível efetuar a transição entre os painéis.

**Classe ModeloTabelaCompra**

Nessa classe há métodos que são responsáveis por obter as informações sobre o carrinho de compra do usuário (getRowCount e getColumnCount).

**getValueAt:** Esse método atualiza a tabela carrinho compra conforme o usuário vai adicionando produtos, exibindo o nome, preço, quantidade e o valor do total da compra carrinho.

Além dos métodos existentes nessa classe para obter as informações sobre a tabela existem os métodos para adicionar e remover os produtos da tabela carrinho compra que por meio das ações do usuário pode adicionar um novo item na tabela ou remover. Esses métodos são chamados por outro método de outra classe (EstoqueGUI) quando o usuário adiciona ou remove produtos.

**getColumnName:** Esse método e responsável por editar as informações default da tabela carrinho compra, colocando os nomes das colunas (nome, preço unit., quantidade e preço parcial).

**isCellEditable:** Deixa somente o campo de quantidade para o usuário poder editar, que o mesmo pode colocar o quanto de itens ele quer mesmo depois de já ter concluído a adição do produto no carrinho.

**setValueAt:** Toda vez que o usuário fizer uma alteração na coluna 2 da tabela, o sistema requisitará ao usuário a senha do gerente, que se for a mesma salva na memória RAM (IFMG), deixará as informações serem alteradas.

**getColumnClass:** Esse método edita a possibilidade de dados compatíveis com os dados da tabela carrinho compra, fixando o tipo de valor a ser exibido em cada coluna. Ex.: coluna zero será uma String.

**calculaPrecoParcialCompra:** Toda vez que o usuário inserir um novo produto na tabela carrinho compra, esse método irá atualizar o Valor Total da compra com base no Valor Parcial do(s) item(s) do usuário.

**limpaCarrinho:** Toda vez que o usuário finalizar a compra, esse método será chamado dentro do botão finalizar para limpar a tabela carrinho compra.

**Classe Produto**

Dentro dessa classe estão inseridos os métodos getters e setters que serão requisitados nas outras classes, que pelo qual, terão retorno de acordo com a sua funcionalidade. Ademais, existem também o método "toString" que será utilizado para converter o arquivo.csv em um bloco de notas, onde atuará como sendo o falso banco de dados.