**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS**

**CAMPUS OURO BRANCO**

**TÉCNICO INTEGRADO EM INFORMÁTICA**

**Julia Fernandes Melo e Bernardo Nunes Mendes**

**Trabalho Prático II - Módulo de Vendas (Interface Gráfica)**

**Ouro Branco**

**27 de Setembro de 2020**

**Introdução**

Esse software consiste em uma aplicação simples e eficiente, capaz de gerenciar as vendas do caixa de uma mercearia. Por meio de uma interface gráfica dinâmica, o operador do caixa consegue realizar a venda dos produtos disponibilizados no estoque do estabelecimento. O vendedor consegue obter, em tempo real, o valor total da compra e uma tabela com as informações dos produtos adicionados ao carrinho, somente por meio da inserção de seu código de barras e a quantidade requisitada desse produto, sem precisar realizar nenhuma conta manual ou qualquer outro tipo de consulta para isso.

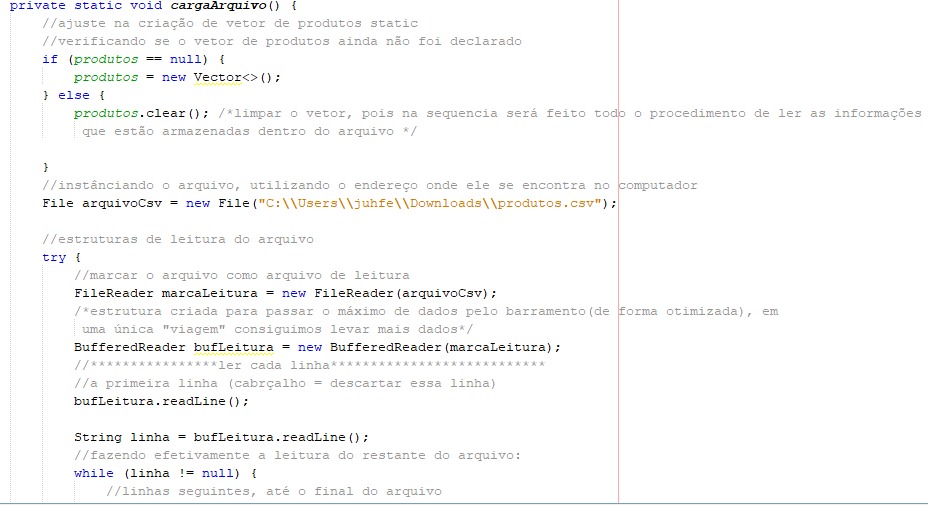
**Implementação**

**Descrição sucinta sobre o desenvolvimento do trabalho:** Para o desenvolvimento do software, resolvemos utilizar a Programação Orientada a Objetos em Java. Nosso programa não é conectado a nenhum Banco de Dados, mas possui uma classe com intuito de simular um (conectado à memória RAM do Computador). Desenvolvemos esse trabalho com o auxílio das aulas do prof. Saulo no intuito de criar um modulo de vendas (interface gráfica).

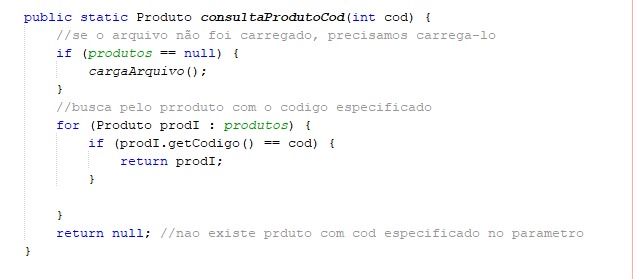
**Descrição dos módulos e sua inter-dependência:** No desenvolvimento do nosso sofwtware, criamos 3 classes programadas Orientadas a Objeto: ModeloTabelaCompra, Produto e fakeBancoDados, as quais dependem uma das outras. A classe produto define o comportamento e as características da mercadoria que está sendo vendida. Porém, as características dos produtos que são vendidos nessa mercearia são dadas por um arquivo, o qual será lido e terá as informações processadas por meio da classe fakeBancoDados antes. Portanto, a classe Produto depende da classe fakeBancoDados. Já a classe ModeloTabelaCompra depende da classe Produto, pois necessita de seus métodos, que definem o comportamento dos produtos, para realizar a manipulação da tabela na interface gráfica.

**Descrição do algoritmo em pequenos blocos de código:**

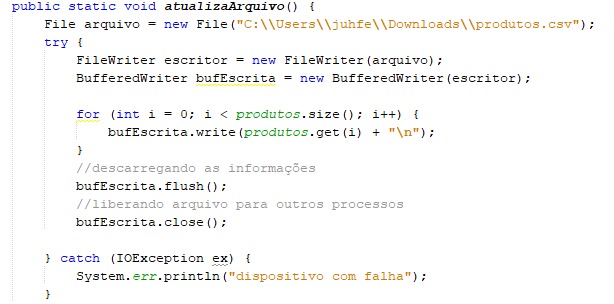
**Primeiro Código: CargaArquivo();** Segue abaixo uma pequena parte do código **CargaArquivo(),** o qual tem a função de ler todas as informações armazenadas do arquivo do Excel e processar esse texto, extraindo cada informação que precisamos para o Produto.

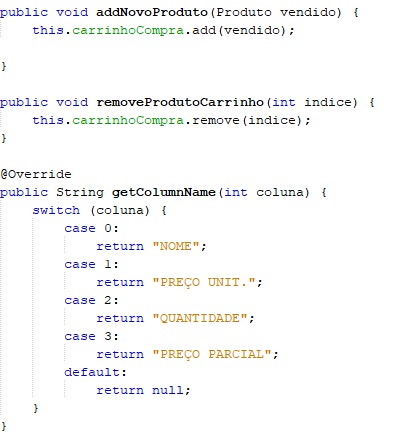
****

**Segundo Código: consultaProdutCod():** esse método realiza a consulta do Produto por meio do seu código.

****

**Terceiro Código: atualizaArquivo():** toda vez que a janela principal for fechada, esse método atualizará o documento do Excel com as informações dos produtos.

****

**Quarto/ Quinto/Sexto Códigos: addNovoProduto(), removeProdutoCarinho() e getColumnName():**

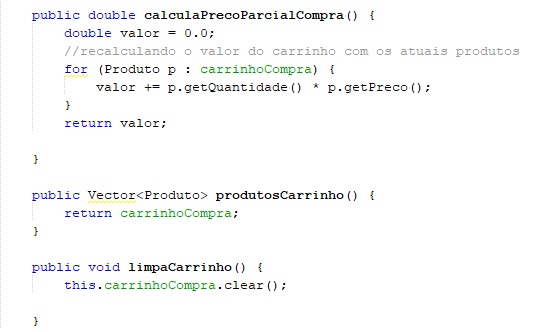
addNovoProduto() é responsável por inserir na tabela, o produto informado pelo usuário e sua respectiva quantidade ao Carrinho de Compras.

removeProdutoCarrinho() é o método responsável por excluir a linha da tabela referente ao produto que se deseja remover.

getColumnName() é o código que insere o título um cada coluna da tabela.

**Sétimo Código: calculaPrecoParcialCompra()**

Esse código é responsável por multiplicar o preço unitário de cada produto com sua respectiva quantidade requisitada, calculando assim,o preço parcial.

****

**Descrição de cada classe:**

Nosso projeto é composto por um frame, o qual nomeamos de **“JanelaPrincipal”.** É nele onde encontramos a estrutura da interface e as transições de tela.

Logo após, construímos dois paineis:

* O primeiro chamado **“CompraGUI”,** interface responsável pela realização de todo processo de compra. Ele possui uma tabela em seu centro, a qual se refere ao Carrinho de Compras (onde todos os produtos comprados são exibidos com suas respectivas informações). Há também nele, campos para inserção/exibição de informações (código de barras , nome,quantidade do produto, preço parcial). Por fim, ele também é constituido de botões (como add Produto, Estoque, Finalizar, Remover) os quais nos permitem manipular as informações contidas na tabela.
* O Segundo chamado **Estoque**, o qual não possui ainda muitas funcionalidades.

Criamos uma classe chamada **“Produto”**, a qual, a partir de suas propriedades e métodos, define o comportamento da mercadoria da mercearia.

Fizemos também, uma classe chamada **“fakeBancoDados”**, que, como o próprio nome sugere, serve para simulação de um Banco de Dados. Nessa classe carregamos as informações do estoque da mercearia, que estavam contidas em um arquivo pré-existente e implementamos inúmeros métodos de consulta necessários para realização da compra.

A última classe criada é nomeada como **“ModeloTabelaCompra”** e foi feita com intuito de criar métodos que ajudem na manipulação dos dados da tabela presente na interface gráfica da compra.

**Descrição do formato de entrada dos dados:** A entrada de dados para adicionar um produto ao carrinho consiste na inserção do seu código de barras. Logo após pressionar enter, o vendedor deve informar a quantidade de itens que o cliente deseja adquirir. Após adicionar um produto ou um lote de produtos ao carrinho, o vendedor pode alterar a quantidade de produtos requisitados ou remove-los, mas para isso, antes ele precisará inserir a senha correta.

**Descrição do formato de saída dos dados:** Logo após o vendedor adicionar um produto ao carrinho, o programa exibirá, na tabela que se encontra no centro da interface, o nome do produto, seu preço unitário, a quantidade requisitada e o preço parcial. Na parte inferior da tela, será exibido o valor total da compra.

Caso não haja a quantidade suficiente do produto para efetuar a compra ou o valor informado não seja condizente com o padrão esperado, o usuário será informado pela exibição de caixas de diálogo.

A partir de uma solicitação da alteração da quantidade do produto, realizada na tabela, todos esses valores serão modificados também.

Quando produto é removido, a linha na tabela referente a esse produto desaparece e o valor parcial é subtraído do valor total. Quando o usuário clica no botão finalizar, todos os produtos da tabela são removidos e o valor parcial exibido é igual à “0.0”.

**Explicação sobre como utilizar o programa:**  Para utilizar o programa, o vendedor, inicialmente, vai inserir o código de barras do produto que o cliente deseja comprar. Logo após será necessário que ele tecle Enter, para assim, modificar e inserir a quantidade requisitada pelo cliete.

Logo após preencher essas informações, basta adicionar no carrinho, selecionando o botão Add Produto. Esse produto será exibido na tabela como um item do carrinho de compras.

Caso desejado, o vendedor poderá remover o produto ou alterar a quantidade de produtos vendidos. Para alterar a quantidade, basta clicar sob o número referente a quantidade e inserir a senha requisitada para que essa ação ocorra. Logo após esse procedimento ele poderá modificá-la.

Para remover algum produto, selecione-o na tabela e pressione o botão remover. Depois confirme e insira a senha requisitada para que essa ação ocorra.

Assim que a compra for finalizada, basta clicar no botão “finalizar” e o programa estará pronto para um novo processidmento.

**Problemas encontrado:** Durante a realização deste trabalho tivemos bastante dificuldade para mexer com o Git, visto que ano passado haviamos usado poucas vezes em nosso trabalho Interdisciplinar. Porém, alguns colegas da turma se disponibilizaram a nos ajudar.

**Conclusão**

A realização desse trabalho foi de suma importância para que nós tenhamos conseguido revisar conceitos aprendidos ano passado, além de obtermos novos conhecimentos em relação ao Git Hub. Também foi útil para nos conseguíssemos aprender novas formas de manipular as informações presentes uma tabela, as quais nós ainda não haviamos aprendido.

**Bibliografia**

ZIVIANI, Nivio. **Solução e Documentação de Trabalhos Práticos.** 2002. Disponível em: https://homepages.dcc.ufmg.br/~nivio/cursos/aed2/roteiro/. Acesso em: 28 set. 2020.