



INSTITUTO FEDERAL
MINAS GERAIS

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS
GERAIS
CAMPUS OURO BRANCO
TÉCNICO INTEGRADO EM INFORMÁTICA**

**Mariana do Carmo Moreira
Patrick Carvalho Moreira**

Mercearia

**Ouro Branco - MG
27 de Setembro de 2020**

Mariana do Carmo Moreira
Patrick Carvalho Moreira

Mercearia

Segundo Trabalho apresentado a disciplina de Tópicos de Desenvolvimento em Software, como parte dos requisitos necessários à obtenção de nota.

Professor(a): Saulo Henrique
Disciplina: Tópicos em Desenvolvimento de Software
Turma: Informática III

Ouro Branco - MG
27 de Setembro de 2020

Resumo

Neste novo cenário de pandemia de Corona Vírus, Seu José resolveu abrir uma mercearia para suprir as necessidades dos moradores do seu bairro de Ouro Branco. Porém, seus produtos ficavam listados em uma planilha do Excel e uma simples venda acabava levando muito tempo. Com isso, foi requisitado a elaboração de um software com interface gráfica para a venda dos produtos para o operador de caixa da mercearia do Seu José. Para a implementação desse software usamos o NetBeans.

Lista de ilustrações

Figura 1 – cargaArquivo()	6
Figura 2 – consultaProdutCod()	7
Figura 3 – atualizaArquivo()	7
Figura 4 – addNovoProduto() e removeProdutoCarrinho()	8
Figura 5 – setValueAt()	8
Figura 6 – calculaPrecoParcialCompra()	9
Figura 7 – Tela Inicial do Software - Compra	10
Figura 8 – Inserindo o código do produto	11
Figura 9 – As informações do produto são inseridas automaticamente por conta do código já informado	11
Figura 10 – Alteração da quantidade do produto e do preço parcial	12
Figura 11 – Adição do produto ao Carrinho de Compra (Tabela)	12
Figura 12 – Alteração da quantidade na própria tabela, mais a requisição da senha do gerente para aprovar a modificação	13
Figura 13 – Quantidade e preços alterados	13
Figura 14 – Adição de um novo produto	14
Figura 15 – Exclusão desse novo produto, com a requisição da senha do gerente para a exclusão do produto	14
Figura 16 – Produto já excluído	15
Figura 17 – Tela limpa depois da compra ser finalizada	15

Sumário

1	Introdução	5
1.1	Programa	5
2	Classe, JFrame e JPanel	6
2.1	Classe	6
2.1.1	Produtos	6
2.1.2	FakeBancoDeDados	6
2.1.3	ModeloTabelaCompra	7
2.2	JFrame e JPanel	9
2.2.1	janelaPrincipal	9
2.2.2	Compra	9
2.2.3	Estoque	10
3	Entradas e Saídas	11
4	Dificuldades	16
5	Conclusão	17

1 Introdução

O “software” armazena em uma tabela todo os produtos que serão vendidos ao cliente, facilitando até mesmo o valor total da compra, pois, o programa automaticamente ao armazenar os produtos vendidos já faz o somatório total da compra daquele cliente. Ademais, o “software” realiza o controle do estoque, ou seja, a cada compra de um produto específico, a quantidade do produto em estoque é abatida. Para tratar o controle de estoque, foi elaborado uma classe de um banco de dados.

1.1 Programa

Foi desenvolvido com o uso de Programação Orientada Objeto com os métodos, para tratar os eventos dos botões. O programa possui três classes orientadas objeto: Produto, FakeBancoDeDados e ModeloTabelaCompra. Um frame: janelaPrincipal. E dois painéis: Compra e Estoque.

O operador de caixa irá utilizar o software informando o código e a quantidade de cada produto que o cliente da mercearia desejar comprar. Esses dados irão para a tabela existe na Interface Gráfica, automaticamente, ao ser enviado para a tabela, o cálculo da compra já estará sendo realizado, facilitando também na questão do pagamento.

2 Classe, JFrame e JPanel

2.1 Classe

As classes utilizadas no desenvolvimento do software foram:

2.1.1 Produtos

A classe Produto representa cada produto que está sendo vendido ou disponível em estoque na mercearia do Seu José. Ela possui todos os atributos necessários do produto: código, nome, preço e a quantidade. Com os seus, respectivos, getters e setters. Contendo também o construtor e o toString.

2.1.2 FakeBancoDeDados

Essa é a classe que representa o estoque da mercearia do Seu José. É por meio dela que conseguimos verificar se ainda existe produto disponível em estoque para efetuar a venda para o cliente. Antes o estoque era controlado por uma planilha do Excel.

Métodos Principais:

O método cargaArquivo() “pega” o arquivo Excel com os produtos e passa eles para o Banco de Dados.

Figura 1 – cargaArquivo()

```
private static void cargaArquivo(){  
  
    //Ajuste na criação do vetor de produto static  
    if(produtos == null){  
        produtos = new Vector<>();  
    }else{  
        produtos.clear();  
    }  
  
    File arquivoCsv = new File("C:\\Users\\Mariana\\Documents\\IFMG - 2020\\Desenvolvimento de Tópicos de Software\\Exercicio
```

O método consultaProdutCod() procura qual o produto referente ao código.

Figura 2 – consultaProdutCod()

```
public static Produto consultaProdutCod(int cod){
    //Se o arquivo ainda não foi carregado precisamos carregar
    if(produtos == null){
        cargaArquivo();
    }
    //Busca pelo produto com o código especificado
    for(Produto prodI : produtos){
        if(prodI.getCodigo() == cod){
            return prodI;
        }
    }
    //Não existe produto com o código especificado no parâmetro
    return null;
}
```

O método `atualizaArquivo()` atualiza o arquivo Excel toda vez que o evento `formWindowClosing` é pressionado.

Figura 3 – atualizaArquivo()

```
public static void atualizaArquivo(){
    File arquivo = new File("C:\\Users\\Mariana\\Documents\\IFMG - 2020\\Desenvolvimento de Tópicos de Software\\Exercicio Av
    try {
        FileWriter escritor = new FileWriter(arquivo);

        BufferedWriter bufEscrita = new BufferedWriter(escritor);

        for(int i = 0; i < produtos.size(); i++){
            bufEscrita.write(produtos.get(i) + "\n");
        }

        bufEscrita.flush();
        bufEscrita.close();

    } catch (IOException ex) {
        System.out.println("Dispositivo com falha");
    }
}
```

2.1.3 ModeloTabelaCompra

Essa classe funciona como um carrinho de compras. Cada produto que o cliente está comprando é adicionado na tabela. Essa classe está ligada diretamente com a tabela que aparece na Interface Gráfica.

Métodos Principais:

O método `addNovoProduto()` adiciona o produto que o cliente deseja comprar na tabela Carrinho de Compra e o método `removeProdutoCarrinho()` remove o produto caso o cliente desista de comprar.

Figura 4 – addNovoProduto() e removeProdutoCarrinho()

```
public void addNovoProduto(Produto vendido) {  
    this.carrinhoCompra.add(vendido);  
}  
  
public void removeProdutoCarrinho(int indice){  
    this.carrinhoCompra.remove(indice);  
}
```

O método setValueAt() permite a modificação da coluna quantidades diretamente na Tabela Carrinho de Compra, como questão de segurança, para tal modificação, é requisitado a senha do gerente.

Figura 5 – setValueAt()

```
@Override  
public void setValueAt(Object novoValor, int linha, int coluna) {  
    if(coluna == 2){  
        //requisitando a senha do gerente para permissão de modificação da quantidade  
        String senha = JOptionPane.showInputDialog(null, "Informe a senha do gerente",  
            "operação restrita", JOptionPane.INFORMATION_MESSAGE);  
  
        if(senha != null && senha.equals("ifmg")){  
            carrinhoCompra.get(linha).setQuantidade((int)novoValor);  
  
            //a tabela e o valor da compra são atualizados  
            this.painel.atualizaQuantidades();  
        }  
    }  
}
```

O método calculaPrecoParcialCompra() calcula o preço parcial de cada produto, fazendo a multiplicação do preço unitário com a quantidades de produtos que o cliente deseja comprar.

Figura 6 – calculaPrecoParcialCompra()

```
public double calculaPrecoParcialCompra() {  
  
    double valor = 0.0;  
  
    //recalculando o valor do carrinho com os atuais produtos  
    for(Produto p : carrinhoCompra){  
        valor += p.getQuantidade() * p.getPreco();  
    }  
    return valor;  
}
```

2.2 JFrame e JPanel

O frame e os painéis utilizados no desenvolvimento do software foram:

2.2.1 janelaPrincipal

Esse frame é responsável pela estrutura da interface gráfica. Nele existe a implementação da barra de rolagem que se adequa a qualquer tamanho de tela. Ele é responsável também pelas transições de telas.

2.2.2 Compra

Esse painel é responsável pela interface gráfica do Carrinho de Compra. Ele consiste em preencher uma tabela, denominada Carrinho de Compras com os produtos que o cliente deseja comprar. Para isso, possui opções de entrada e botões para o tratamento do comportamento do software.

Figura 7 – Tela Inicial do Software - Compra

NOME	PREÇO UNITÁRIO	QUANTIDADE	PREÇO TOTAL
------	----------------	------------	-------------

Valor Total: R\$ 0,00

2.2.3 Estoque

Esse JPainel ainda não foi implementado no software.

FASTFORMAT

Você precisa comprar esse documento para remover a marca d'água.
Documentos de 10 páginas são gratuitos.

You need to buy this document to remove the watermark.
10-page documents are free.

3 Entradas e Saídas

O operador de caixa preenche o campo Código com o código do produto, precionando a tecla Enter o produto já é buscado no banco de dados e aparece o seu nome e preço na tela. Em seguida, o operador informa a quantidade e precionando o botão AddProduto, o produto é adicionado na tabela.

Figura 8 – Inserindo o código do produto

Código: 78564232

Nome:

Quantidade: 1

Preço Parcial:

Add Produto

NOME	PREÇO UNITÁRIO	QUANTIDADE	PREÇO TOTAL
------	----------------	------------	-------------

Valor Total: R\$ 0,00

Finalizar Estoque

Figura 9 – As informações do produto são inseridas automaticamente por conta do código já informado

Você precisar comprar esse documento para remover a marca d'água.
Documentos de 10 paginas são gratuitos.

Código: 78564232

Nome: Recheado

Quantidade: 1

Preço Parcial: 4,41

Add Produto

NOME	PREÇO UNITÁRIO	QUANTIDADE	PREÇO TOTAL
------	----------------	------------	-------------

Valor Total: R\$ 0,00

Finalizar Estoque

Figura 10 – Alteração da quantidade do produto e do preço parcial

The screenshot shows a web application interface with a yellow background. On the left, there are input fields for 'Código' (78564232), 'Nome' (Recheado), 'Quantidade' (6), and 'Preço Parcial' (26,46). Below these is an 'Add Produto' button. In the center is a table with columns: 'NOME', 'PREÇO UNITÁRIO', 'QUANTIDADE', and 'PREÇO TOTAL'. The table is currently empty. To the right of the table is a 'Remover' button. At the bottom left, there is a 'Valor Total: R\$ 0,00' label. At the bottom right, there are 'Finalizar' and 'Estoque' buttons.

Figura 11 – Adição do produto ao Carrinho de Compra (Tabela)

The screenshot shows the same web application interface as Figure 10, but with the product 'Recheado' added to the shopping cart table. The table now has one row with the following data: 'Recheado' (NOME), '4,41' (PREÇO UNITÁRIO), '6' (QUANTIDADE), and '26,46' (PREÇO TOTAL). A large, colorful watermark 'FASTFORMAT' is overlaid on the center of the image. Below the watermark, there is text in Portuguese and English: 'Você precisa comprar este documento para remover a marca d'água. Documentos de 10 páginas são gratuitos.' and 'You need to buy this document to remove the watermark. 10-page documents are free.'

Outra funcionalidade implementada é que conseguimos alterar a quantidade com o produto já no Carrinho de Compras, para isso, é necessário a senha do gerente. Também temos a opção de remover algum produto, caso o cliente desista de levar, requisitando a senha do gerente para a exclusão.

Figura 12 – Alteração da quantidade na própria tabela, mais a requisição da senha do gerente para aprovar a modificação

The screenshot shows a software interface with a yellow background. On the left, there are input fields for 'Código', 'Nome', 'Quantidade' (set to 1), and 'Preço Parcial', along with an 'Add Produto' button. In the center, a table displays product information:

NOME	PREÇO UNITÁRIO	QUANTIDADE	PREÇO TOTAL
Recheado	4,41	3	26,46

Below the table is a 'Remover' button. A modal dialog titled 'operação restrita' is open, prompting the user to 'Informe a senha do gerente' (Enter the manager's password) with 'OK' and 'Cancelar' buttons. At the bottom, there is a 'Valor Total' field showing 26,46, and 'Finalizar' and 'Estoque' buttons.

Figura 13 – Quantidade e preços alterados

This screenshot shows the same software interface as Figure 12, but with updated values. The 'Quantidade' field is still 1, but the 'Preço Parcial' field is now empty. The table shows the same product, but the 'PREÇO UNITÁRIO' is 4,41 and the 'QUANTIDADE' is 3, resulting in a 'PREÇO TOTAL' of 13,23. The 'Remover' button is still present. A large, semi-transparent watermark reading 'FASTFORMAT' is overlaid on the center of the screen. Below the watermark, there is text in Portuguese and English: 'Você precisa comprar esse documento para remover a marca d'água. Documentos de 10 páginas são gratuitos. You need to buy this document to remove the watermark. 10-page documents are free.' At the bottom, the 'Valor Total' field now shows 13,23, and the 'Finalizar' and 'Estoque' buttons are still visible.

Figura 14 – Adição de um novo produto

Código

Nome

Quantidade

Preço Parcial

Add Produto

NOME	PREÇO UNITÁRIO	QUANTIDADE	PREÇO TOTAL
Rachado	4,41	3	13,23
Bombom	5,19	12	62,28

Remove

Valor Total: 88,74

Finalizar Estoque

Figura 15 – Exclusão desse novo produto, com a requisição da senha do gerente para a exclusão do produto

Código

Nome

Quantidade

Preço Parcial

Add Produto

NOME	PREÇO UNITÁRIO	QUANTIDADE	PREÇO TOTAL
Rachado	4,41	3	13,23
Bombom	5,19	12	62,28

Remove

Valor Total: 88,74

Finalizar Estoque

FASTFORMAT

Confirma a exclusão?

Deseja remover o produto selecionado?

Sim Não

Você precisar comprar esse documento para remover a marca d'água.
Documentos de 10 páginas são gratuitos.

You need to buy this document to remove the watermark.
10-page documents are free

Figura 16 – Produto já excluído

The screenshot shows a software interface with a yellow background. On the left, there are input fields for 'Código', 'Nome', 'Quantidade' (set to 1), and 'Preço Parcial', along with an 'Add Produto' button. In the center, a table displays the product list:

NOME	PREÇO UNITÁRIO	QUANTIDADE	PREÇO TOTAL
Recheado	4,41	3	13,23

Below the table, there is a 'Valor Total:' field showing '13,23'. On the right, there is a 'Remove' button. At the bottom, there are 'Finalizar' and 'Estoque' buttons.

E quando terminar de informar o código dos produtos que o cliente deseja levar, tem o botão finalizar que finaliza a compra do cliente, limpando toda a Interface Gráfica.

Figura 17 – Tela limpa depois da compra ser finalizada

The screenshot shows the same software interface as Figure 16, but after the purchase has been finalized. The 'Valor Total:' field now shows '0,00'. A large, semi-transparent watermark 'FASTFORMAT' is overlaid across the center of the screen. The text 'Você precisa comprar esse documento para remover a marca d'água.' and 'Documentos de 10 páginas são gratuitos.' is also visible. The 'Finalizar' and 'Estoque' buttons are still present at the bottom.

4 Dificuldades

A maior dificuldade encontrada foi mexer com o Git e o GitHub. Mas, com o auxílio do aluno Caio, conseguimos enviar o projeto para o GitHub.



Você precisar comprar esse documento para remover a marca d'água.
Documentos de 10 páginas são gratuitos.

You need to buy this document to remove the watermark.
10-page documents are free.

5 Conclusão

O trabalho proposto foi essencial para a revisão da matéria de POO, com ele pudemos rever conceitos e aprender novas funcionalidades, tais como ligar um arquivo de memória RAM com uma classe no NetBeans.



**Você precisar comprar esse documento para remover a marca d'água.
Documentos de 10 páginas são gratuitos.**

**You need to buy this document to remove the watermark.
10-page documents are free.**