**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA**

**UNIDADE EDUCACIONAL SÃO GABRIEL**

**Bacharelado em Sistemas de Informação**

**Matheus Ryuji Matsutane**

**TRABALHO PRÁTICO**

Belo Horizonte

2020

**Matheus Ryuji Matsutane**

**TRABALHO PRÁTICO**

Trabalho Prático apresentado como requisito parcial à aprovação na disciplina Algoritmos e Estruturas de Dados

Professores: Rosilane Ribeiro da Mota

Belo Horizonte

2020

**SUMÁRIO**

1. Objetivo do programa 4

2. Detalhes da implementação 4

2.1. Métodos desenvolvidos 4

2.2. Como executar o programa 5

2.3. Testes realizados 6

2.4. Decisões de implementação 6

* **Objetivo do programa**

O programa, batizado de “Máscara Tech” e desenvolvido em Java, tem como objetivo oferecer um ambiente de fácil utilização para que produtores locais de máscaras sejam capazes de manter suas vendas atualizadas.

Foram implementadas funcionalidades como: registro de vendas, atualização de estoque, atualização de custos de produção, atualização de preços de venda, geração de relatório diário via arquivo de texto.

Além dos objetivos básicos, também foram implementados os objetivos extras, como a elaboração de uma classe especial e o uso de arquivos de texto para gerar relatórios.

* **Detalhes de implementação**

Dentro do código, seguiu-se o padrão de usar a língua inglesa para nomear variáveis, métodos, classes e os demais elementos. Também foi levada em consideração a norma padrão “Camel Case” para a nomenclatura dos elementos.

* **Métodos desenvolvidos**

Na classe ApplicationStore.java, temos:

O **código principal (main)**, que cria uma flag do tipo boolean “exitFlag” inicialmente falsa, recebe os valores das máscaras iniciais através do vetor “masks’ e então mantém um loop para que o programa possa manter-se ativo ou encerrar.

O **método mainMenu (Menu de Operações),** que funciona como um menu principal. Nele, o usuário deve escolher uma entre 6 opções – registrar venda, atualizar estoque, atualizar preço de produção, atualizar preço de venda, gerar relatório ou sair. Retorna um boolean, que se for True, encerra o programa. Sendo False, mantém o programa ativo.

O **método maskMenu (Menu de Máscaras)** é um submenu, exibido a partir do momento em que o usuário deseja escolher uma entre os 4 tipos de máscara – máscara infantil lisa, máscara infantil estampada, máscara adulto lisa, máscara adulto estampada. Retorna um valor int, que representa a opção escolhida pelo usuário.

O **método masksReport (Geração de Relatórios)** tem a função de gerar um arquivo report.txt e enumerar a descrição, estoque, quantidade vendida e lucro referentes a cada máscara.

Na classe Mask.java, temos:

A **classe Mask (Máscara)** declarada, que define separadamente cada elemento: id (identificação), description (descrição), stock (estoque), dailySale (vendas diárias), productionPrice (preço de produção) e salePrice (preço de vendas).

O **método registerDailySale**, que verifica se a quantidade vendida é maior que 0, registra as vendas e informa o sucesso da operação. Se o número de unidades vendidas for maior que o estoque, uma mensagem de erro é exibida ao usuário.

O **método getDailySale** **(Obter Vendas Diárias)** retorna a quantidade de vendas diária, como valor do tipo int.

O **método setDailyStock** **(Definir Estoque Diário)** atualiza o estoque baseado na quantidade fornecida como parâmetro pelo usuário.

O **método getDailyStock (Obter Estoque Diário)** retorna a quantidade de máscaras em estoque, como valor do tipo int.

O **método setProductionPrice (Definir Preço de Produção)** atualiza o custo de produção baseado no valor fornecido como parâmetro pelo usuário.

O **método getProductionPrice (Obter Preço de Produção)** retorna o custo de produção, com valor do tipo float.

O **método** **setSalePrice (Definir Preço de Vendas)** atualiza o preço de vendas baseado no valor fornecido como parâmetro pelo usuário.

O **método getSalePrice (Obter Preço de Produção)** retorna o preço de vendas, com valor do tipo float.

O **método getDescription (Obter Descrição)** retorna a descrição do produto, com valor do tipo string.

* **Como executar o programa**

Dentro do diretório compactado enviado no Canvas, foi incluído o executável MascaraTech.jar que pode ser executado através de uma linha de comando, como o Prompt de Comando ou PowerShell no Windows através do comando:

java -jar .\MascaraTech.jar\



Alternativamente, também é possível executar o programa a partir do seu código fonte, que está disponível em **MascaraTech/src/ApplicationStore**, com o auxílio da IDE de preferência.

* **Testes realizados**

O programa foi testado exaustivamente por ambos os membros do grupo, colocando à prova todas as possibilidades que ele traz através da navegação de menus e inserção de valores quando preciso.

Dessa forma, chegou-se à conclusão de que o código atende o objetivo inicialmente proposto, e está livre de falhas básicas.

Entretanto, quando o usuário insere dados incorretamente ou executa ações impossíveis, mensagens de erro são exibidas.

* **Decisões de implementação**

Todas as funcionalidades foram detalhadas acima no documento e através de comentários no código. Também foi realizado o esforço de tornar o nome das variáveis, métodos e classes o mais claro possível, evitando dúvidas por parte de terceiros.

Foram usados vetores, diversos métodos e diversas classes para cumprir com os objetivos propostos, todos explicados da forma mais clara possível ao longo da documentação do código.