

SALÃO MARCELUS

SISTEMA DE CONTROLE DE ESTOQUE

Sumário

LINKS



REPOSITÓRIO DO PROJETO:

<https://github.com/l7sReact/controle-de-estoque>

- 1 Introdução e detalhes do projeto
- 2 Descrição geral do sistema
- 3 Requisitos do sistema
- 4 Funções do código

Introdução e detalhes do projeto

Nessa parte falaremos do projeto em geral, seus objetivos e para quem faremos



[Voltar para o slide de Temas](#)

Introdução e detalhes do projeto

O projeto tem o tema de criar um sistema de controle de estoque na linguagem C, o salão de beleza Marcelus é localizado no bairro Fazendinha (Curitiba), o salão é composto por 3 cabeleireiros e 2 manicures. Todos os colaboradores gerenciam o estoque dos produtos.

Objetivos

Tem como objetivo controlar o estoque de um salão de beleza, podendo-se adicionar, remover ou verificar os produtos de um banco de dados, para que se verifique caso tenha falta, evitar o vencimento e fazer a remoção de produtos vencidos que podem causar danos a saúde de seus clientes. Pois a verificação é feita de modo manual. No final, podendo prejudicar o salão judicialmente e financeiramente.

Regras de negócio do salão

- Os produtos são comprados através de distribuidoras oficiais das marcas;
- O salão utiliza duas formas de controle de estoque, o PEPS e PVPS. O PEPS significa que, quando o produto entra no estoque, já é posto nas prateleiras na frente para ser vendido. O PVPS é quando o produto que está mais perto de vencer tem prioridade para ser utilizado ou vendido primeiro;
- Controlar a movimentação dos produtos para uso interno e linha home-care (revenda) é extremamente importante para o controle de estoque. Com essas informações, se pode fazer o planejamento de compras mais eficientes. Durante a entrada dos produtos é necessária que seja feita uma conferência rigorosa das mercadorias. Fazendo isso, deve-se conferir as quantidades e as datas de validade;
- O local de armazenamento deve ser em um local arejado, fora do alcance da luz solar e em temperatura ambiente;
- Classificar o estoque é saber quais produtos de maior uso interno e venda. Isso é importante para que o salão saiba quais itens devem ser repostos com frequência, quais precisam serem melhor trabalhados ou no seu uso ou na sugestão de venda.

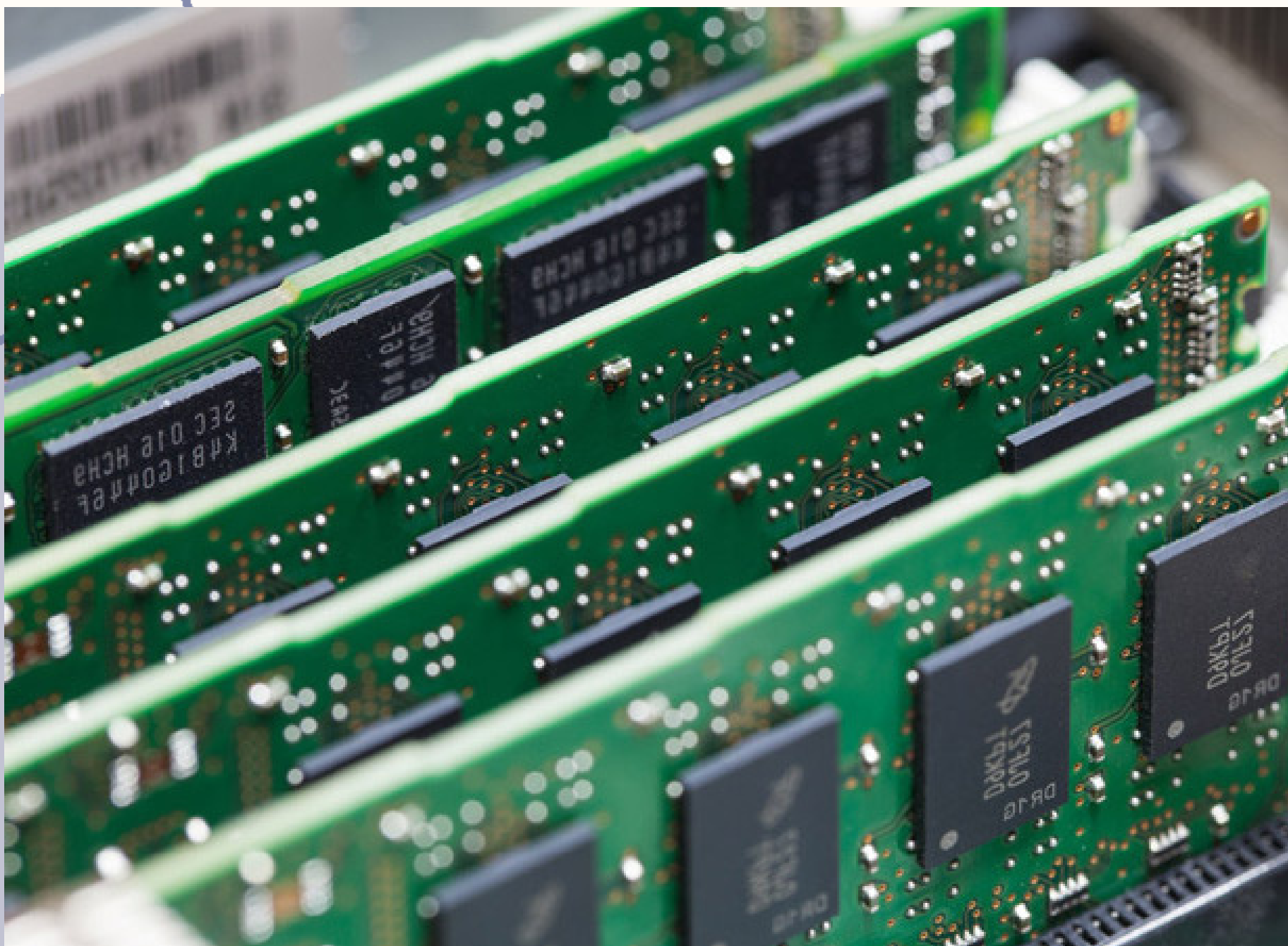
Descrição geral do sistema

[Voltar para o slide de Temas](#)

Descrição geral do sistema

O sistema é desenvolvido em linguagem C utilizando o VSCodium e Dev C++, tendo a capacidade de controlar o estoque do salão Marcelus, permitindo:

- Cadastro de novos produtos;
- Controle de entrada e saída de produtos;
- Exclusão de produtos;
- Relatório geral de estoque;
- Relatórios de produtos que estão abaixo do estoque mínimo;
- Relatórios de produtos vencidos;
- Relatório de aumento do preço de compra do fornecedor.



Requisitos do sistema

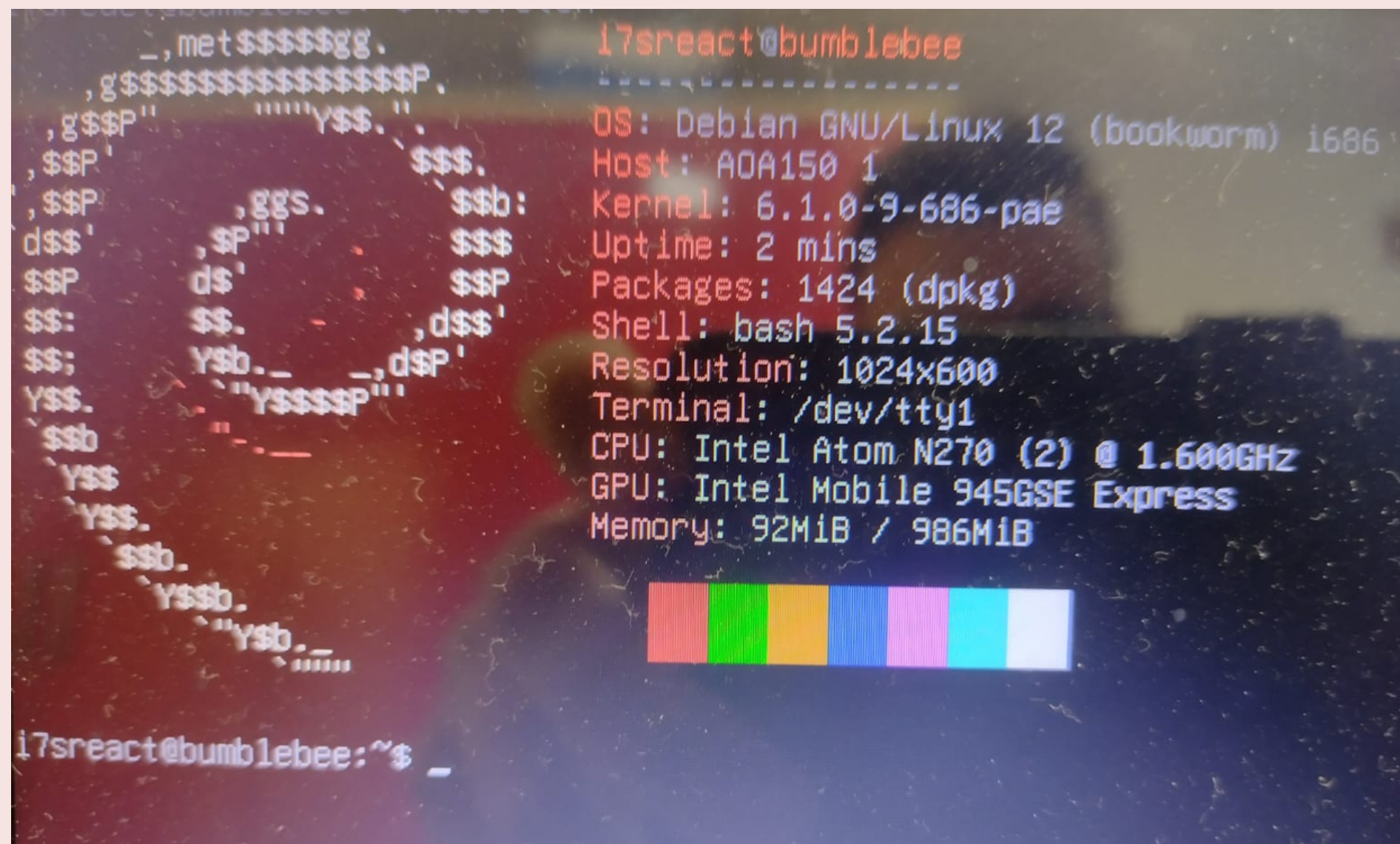
AMBIENTES DE DESENVOLVIMENTO E BIBLIOTECAS DO SISTEMA

- DEV C++ (Versão 5.11);
 - VSCodium (Versão 1.79.0);
 - GIT (Versão 2.34.1);
 - GCC (Versão 11.3.0).
-
- `stdio.h`;
 - `stdlib.h`;
 - `string.h`;
 - `time.h`;
 - `ctype.h`.

Requisitos do sistema

Onde o sistema pode rodar

Por ser um sistema leve e sem interface gráfica, é possível rodarem em máquinas antigas e que possuem poucos recursos, por exemplo, uma máquina que possui um Intel Atom de 32bits com frequência de 1.60GHz e 1GB de RAM.



Funções do código

```
void sortProduct(Tproduct inventory[])  
  
// Function to add a new product  
void addProduct(Tproduct inventory[])  
  
// Function to exhibit the general report  
void generalReport(Tproduct inventory[])  
  
// Function to exhibit a report of the products  
void reportPriceList(Tproduct inventory[])  
  
// Function to make a change in a product  
void changeProduct(Tproduct inventory[])
```


Funções do código

Função para validar as datas de vencimento do produto na hora de cadastrar

```
int dateExp(struct tm dateE);
```

Substituto do fflush(stdin)

```
void cleanStdin(void);
```

Função para ler o arquivo inventory.dat

```
void readFile(Tproduct inventory[], int *size);
```

Função para escrever os dados no arquivo inventory.dat

```
void writeFile(Tproduct inventory[], int size);
```

Função da busca binária

```
int binarySearch(Tproduct inventory[], int key, int size);
```

Função auxiliar que verifica se o vetor está vazio

```
int null(int size);
```

Funções do código

Função para ordenar os produtos que estão dentro de um vetor

```
void sortProduct(Tproduct inventory[], int size);
```

Função para adicionar um novo produto

```
void addProduct(Tproduct inventory[], int *size);
```

Função para exibir um relatório geral dos produtos cadastrados

```
void generalReport(Tproduct inventory[], int size);
```

Função para exibir um relatório dos preços dos produtos

```
void reportPriceList(Tproduct inventory[], int size);
```

Função para alterar dados de um produto

```
void changeProduct(Tproduct inventory[], int size);
```

Função para deletar um produto

```
void deleteProduct(Tproduct inventory[], int *size);
```


Funções do código

Função para achar um produto através de seu código

```
void consultProduct(Tproduct inventory[], int size);
```

Função para achar um produto através do seu nome

```
int consultProductByName(Tproduct inventory[], int size);
```

Função para abrir o menu de relatórios especiais

```
void espMenu(Tproduct inventory[], int size);
```

Função para mostrar produtos vencidos

```
void reportExpiredProduct(Tproduct inventory[], int size);
```

Função para aumentar o preço de compra dos produtos de um fornecedor

```
void increaseSupplierPrices(Tproduct inventory[], int size);
```

Função para mostrar um relatório de produtos abaixo da quantidade mínima do estoque

```
void reportMinimalInventory(Tproduct inventory[], int size);
```

Funções do código

Função para alterar as datas de vencimento do produto (Não deu certo)

```
void changeExpiredMenu(Tproduct inventory[], int size);
```

Função para alterar TODAS as informações de um produto (Não conseguimos esse também)

```
void changeFunction(Tproduct inventory[], int size);
```

CONSIDERAÇÕES FINAIS

Com esse programa, é possível monitorar os produtos de um salão de beleza, resultando em um melhor gerenciamento do estoque.

Primeiramente, a fiscalização do sistema fez-se necessária, pois sem a devida organização dos dados podem ocasionar em acidentes que prejudiquem a saúde do cliente e abalar a reputação do estabelecimento.

Ademais, ter um sistema que te notifique caso haja um problema com o produto a ser vendido ou utilizado indica ser um dos melhores métodos para um trabalho harmônico. Portanto, possuir um sistema de controle de estoque é essencial para o funcionamento de um salão de beleza.

INTEGRANTES

DA EQUIPE

1

DOUGLAS NASCIMENTO

2

ANDRÉ HENRIQUE FIATKOSKI LUSTOSA

3

PABLO CELESTINO DA ROCHA LOBO



OBRIGADO POR
ASSISTIREM!