Tema B

Sistema de Gestão de máquina de vending de snacks

Trabalho realizado por:

Napame Sá 29210

Ricardo Barbosa 30003

Tiago Cambão 30542

Professor

Carlos Edgar Novo

<Programação-Engenharia Mecatrónica>

11 de junho de 2023

Sumário

1 Introdução

Este projeto foi executado através do software DevC++ . O sistema foi projetado com o objetivo de simular a operação e controlo de uma máquina de vending, dividindo as suas funcionalidades em duas partes distintas: uma destinada ao usuário comum e outra reservada ao administrador, permitindo uma gestão total da máquina.
O sistema foi projetado com o objetivo de simular a operação e controlo de uma máquina de vending, dividindo as suas funcionalidades em duas partes distintas: uma destinada ao usuário comum e outra reservada ao administrador, permitindo uma gestão total da máquina.
Para o usuário comum, o sistema oferece um stock de 48 tipos diferentes de produtos, organizados em uma matriz de 6 filas e 8 colunas, sendo que ao realizar uma compra, são fornecidas código do produto, nome, tipo, informação adicional, calorias, preço, data de validade e quantidade disponível. Já o administrador possui privilégios adicionais, como a capacidade de adicionar novos produtos, verificar as quantidades e validades, atualizar os preços unitários e do stock em geral, bem como consultar todas as informações de faturação e realizar o levantamento dos valores contidos nela. Para começar o trabalho e dividir as tarefas fizemos fluxogramas para o admistrador, o utilizador e o momento de escolha entre os dois. Nas figuras a seguir encontram-se os fluxogramas que usamos como base para programar este projeto. Na pasta zipada onde se encontra este relatório também terá um pasta chamada `Fluxogramas` onde pode abrir o ficheiro e ver com mais detalhe o nosso fluxograma.

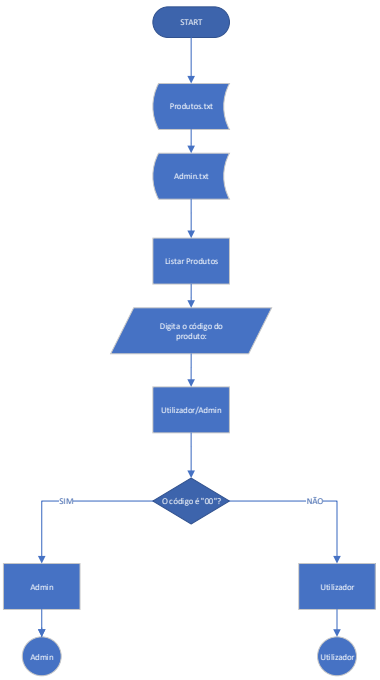

Figura 1: Fluxograma Utilizador/Admin

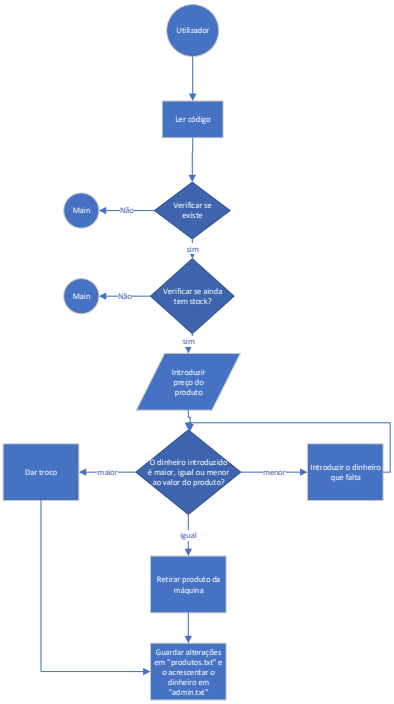

Figura 2: Fluxograma Utilizador

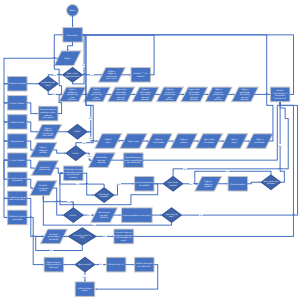

Figura 3: Fluxograma Admin

2 Funções necessárias para a execução do programa:

Figura 4: Funções

3 Estruturas utilizadas

Para este projeto usamos uma estrutura principal (produtos) e duas complementares. Na estrutura principal declaramos as variáveis importantes sobre o produto separando a data de validade ,as informações extras e calorias em outras estruturas separadas.

```
typedef struct data{
    int dia,mes,ano;
}DATA;
```

Figura 5: Struct data

```
typedef struct info{
    char informacao[200];
    float calorias;
}INFO;
```

Figura 6: Struct info

```
typedef struct produtos{
    char codigo[3];
    char nome[30];
    char tipo_produto[20];
    INFO info_produto;
    DATA data_validade;
    float price;
    int quantidade;
}PRODUTO;
```

Figura 7: Struct produtos

Typdef struct data:
Esta estrutura (5) permitiu-nos criar variáveis que nos fosse possível armazenar e verificar o dia mês e ano de uma determinada data.

Typdef struct info:
Estrutura criada (6) para declarar variáveis ao qual são guardadas as informações mais detalhadas como os ingredientes que um determinado produto leva e as calorias do mesmo.

Typdef struct produtos:
Estrutura criada (7) para declarar as variáveis com todas as informações importantes dos produtos, ou seja, a criação do código associado a um tipo de produto e sua localização para o utilizador escolher no momento da compra, nome do artigo o tipo de produto.

4 Adicionar Produtos

```
//////////////////////////ADICIONAR PRODUTO////////////////////////////////
void adicionar_produto(PRODUTO produtos[MAX], int* num_produto){
```

Figura 8: Início da função adicionar produto

Esta função, como o nome indica, permite ao administrador adicionar um novo produto á maquina ou, caso a maquina esteja já lotada, eliminar as informações de um produto escolhido pelo admin deixando espaço em branco para depois editar.

Caso a máquina ainda aceite produtos esta função perguntará por:

- • Código da localização que pretende colocar o item;
- • Tipo de produto (categoria);

- • Informações (ingredientes);
- • Calorias;
- • Prazo de validade;
- • Preço;
- • Quantidade que pretende adicionar.

Após introduzidas todas essas variáveis, as informações serão armazenadas na estrutura de dados chamada `"PRODUTO"`. Dessa forma, os itens serão chamados com o símbolo de ponto no início de cada um.

5 Guardar produto no ficheiro de texto

```
//////////////////////GUARDAR PRODUTO NO FICHEIRO//////////////////////
void guardar_produto(PRODUTO produtos[MAX],int* num_produto){
```

Figura 9: Início da função guardar produtos no ficheiro

Esta função tem como principal objetivo guardar as informações introduzidas sobre o produto num ficheiro de texto designado por `"produtos.txt"`, em que a função percorre o `"produtos"` armanezados em `RAM` e escreve cada informação no ficheiro separando por ponto e virgula. Assim, esta informação, que seria perdida após o fecho do programa, fica guardado permanentemente num ficheiro.

6 Reeabastecer máquina

```
//////////////////////REEABASTECER MAQUINA//////////////////////////////
void reeabastecer_maquina(PRODUTO produtos[MAX], int* num_produto){
```

Figura 10: Funçao reabastecer máquina

A função permite reabastecer/adicionar uma quantidade de um determinado produto na máquina. A função em primeiro lugar mostra a lista de produtos disponíveis e solicita ao administrador que insira o código da localização do produto a ser reabastecido e a quantidade a ser adicionada.// Após introduzido todos os valores, a função atualiza a quantidade do produto correspondente na lista de produtos e chama a função guardar produto guardar as alterações no ficheiro de texto correspondente.

7 Listar produtos no menu administrador e utilizador comum

```
//////////////////////////LISTAR PRODUTOS PARA ADMIN DE FORMA MAIS SIMPLES///////
void listar_produtos_admin(PRODUTO produtos[MAX], int* num_produto){
```

Figura 11: Função listar produtos para admin

```
//////////////////////////LISTAR PRODUTOS////////////////////////////////
void listar_produtos(PRODUTO produtos[MAX], int* num_produto){
```

Figura 12: Função listar produtos para utilizador

A função "listar_produtos_admin" percorre a lista de produtos e mostra os produtos de forma simplificada, exibindo apenas o código, nome, tipo, informações, calorias, data de validade, preço e quantidade de cada produto, sem especificar o que cada coluna significa como acontece no "listar_produtos" onde, por questões estéticas, é mostrado o que cada coluna corresponde conforme na figura 13.

```
*****************************************CNR.Lda*************************************************************************
_____________________________________________________________________________________________________________________
CÓDIGO:  |  NOME DO PRODUTO:                |  TIPO DO PRODUTO:  |  DATA DE VALIDADE:  |  PREÇO:  |  QUANTIDADE:
A1          Kit-Kat                          Chocolate            01/01/2024            1,50       10
A2          Kinder-Bueno                     Chocolate            01/12/2023            1,50       10
A3          Kinder-Delice                    Chocolate            01/12/2023            1,50       10
A4          Mars                             Chocolate            01/11/2023            1,80       10
```

Figura 13: Aparência da função listar produtos

8 Listar validade dos produtos

```
////////////////LISTAR VALIDADE DOS PRODUTOS ////////////////////////
void listar_validade_produtos(PRODUTO produtos[MAX], int* num_produto){
```

Figura 14: Função que lista a validade dos produtos

Esta função lista os produtos que estão fora da validade. Para isso, primeiro solicita ao administrador que insira a data atual, após introduzida a data, a função compara com a data de validade de cada produto, caso algum produto ultrapasse a data de validade esta exibe os produtos que já não podem estar armazenados na máquina. O administrador tem a possibilidade de remover um produto específico digitando o código correspondente e a quantidade a ser removida, caso queira remover mais produtos o programa perguntará e caso afirmativo repetirá a pergunta pelo código. Por questões de lógica, o programa não deixa o admistrador retirar mais produtos do que há de quantidade daquele produto.

Caso não haja produtos fora de prazo o administrador tem sempre a possibilidade de verificar as datas de validades de todo o stock contido na máquina. As alterações são sempre guardadas no ficheiro usando a função "guardar_produto".

9 Guardar compra no ficheiro de texto

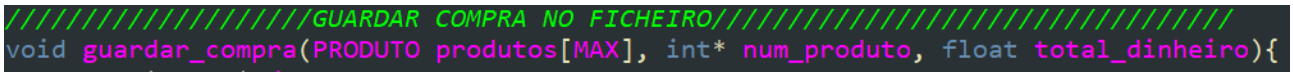

Figura 15: Função guardar compra

Função responsável por guardar as informações dos produtos após uma compra no ficheiro de texto `"produtos.txt"`. Além disso, ela também atualiza o valor total de dinheiro guardado pela máquina no ficheiro de texto `"admin.txt"` com base no total de dinheiro recebido.

10 Calculo Troco

Figura 16: Função calculo troco

Função responsável por realizar todo o cálculo do troco a ser dado ao cliente após uma compra. Nesta função estão todos os casos possíveis quando se introduz o valor para pagar como:

- • Caso o utilizador coloque o dinheiro certo;
- • Caso o utilizador coloque dinheiro a mais, devolvendo o troco;
- • Caso o utilizador coloque dinheiro insuficiente, pedindo o dinheiro que falta até colocar o dinheiro certo ou a mais, devolvendo o troco no final se necessário;

Para verificar todos esse casos possíveis programamos várias funções `if` dentro de um `loopdo while()` até que o valor introduzido pelo utilizador fosse igual ou maior que o preço associado ao produto.

Ela recebe como parâmetros o array `"produtos"`, e o número total de produtos `"num_produto"`, o preço introduzido anteriormente pelo utilizador, o `i` que corresponde ao `num_produto` encontrado no ciclo `for` anterior e o `dinheiro_total` onde depois será adicionado ao ficheiro `admin.txt` onde guarda o dinheiro acumulado.

11 Comprar produtos

Figura 17: Função comprar produto

A função permite ao utilizador realizar a compra e saber mais informações sobre o produto. Para podermos apresentar ao utilizador todas as informações a função antes usa a `listar_produtos` para listar ao utilizador todos os produtos e as informações.De seguida, pergunta pelo código do produto onde depois a função vai percorrer toda a lista de produtos até encontrar o código igual. Se o produto for encontrado e tiver quantidade disponível, pergunta ao utilizador se quer sabes mais informações sobre o produto que selecionou.
Caso for afirmativo exibe as informações e as calorias do produto. Caso seja negativo passa diretamente para a parte onde pede ao utilizador para colocar o dinheiro correspondente os produto que escolheu guardando na variável `"preco_utilizador"`.
Em seguida, chama a função `calculo_troco` (10) para calcular o troco e atualizar o valor total de dinheiro acumulado.

12 Carregar produtos no ficheiro de texto

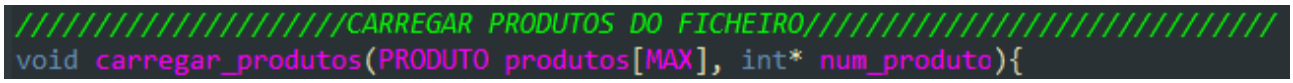

Figura 18: Função carregar produtos num txt

A função extrai os produtos a partir de um ficheiro de texto chamado `"produtos.txt"`, onde a função lê as informações da memória permamente do disco do computador de cada produto e os armazena em `RAM` para poder ser usado pelo programa. O número total de produtos `"num_produto"`é atualizado com base na quantidade de produtos introduzidos.
Esta função encontra-se na `main` antes de começar o programa pois é preciso primeiro fazer esse carregamento para depois ser possível listar os produtos.

13 Levantamento do dinheiro

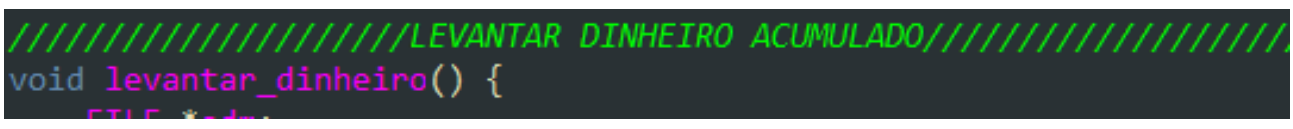

Figura 19: Função levantar dinheiro

Esta função permite ao administrador da máquina retirar o dinheiro acumulado.
Ela lê o valor atual de dinheiro guardado contido no ficheiro de texto `"admin.txt"`. Em seguida, exibe o valor e pergunta ao administrador se ele deseja fazer o levantamento. Se a resposta for afirmativa,

o valor do dinheiro é automaticamente retirado do ficheiro e definido como zero e atualizando esse mesmo ficheiro. Caso seja negativo o valor é inalterado e volta ao menu de `admin`.

14 Atualizar preço de um ou vários produtos

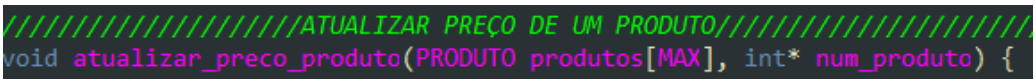

Figura 20: Função atualizar preço de um produto

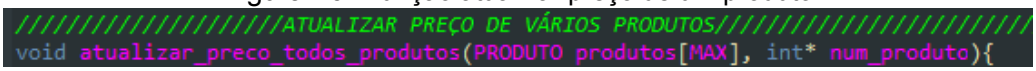

Figura 21: Função atualizar preço de todos os produto

A função `"atualizar_preco_produto"` 20 permite ao administrador atualizar o preço de um produto específico. Ela exibe a lista de todos os produtos disponíveis e solicita ao administrador que insira o código do produto a ser atualizado. Em seguida, pede ao administrador que introduza o novo preço. Por fim, pergunta se quer mudar o preço de outros produtos. Se o admistrador o quiser o processo recomeça de novo.

Já a função `"atualizar_preco_todos_produto"` 21 permite ao administrador atualizar o preço de todos os produtos com base numa percentagem de inflação. O administrador informa a percentagem de inflação e a função atualiza o preço de cada produto com base nesse valor. Todas as alterações são guardadas no ficheiro `"produtos.txt"`.
Como forma de segurança o código somente deixa que o valor da inflação seja entre 1% e 100%. Isso é sempre repetido até que seja inserido um valor dentro do intervalo permitido.

15 Editar produto

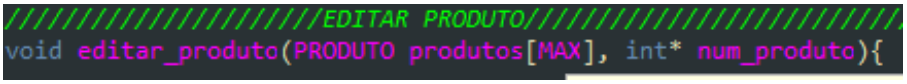

Figura 22: Função editar produto

Esta função é usada para quando o admistrador quiser editar as informações de um produto já existente ou, caso um produto seja eliminado porque já atingiu a capacidade máxima da máquina, para inserir novas informações. Como todos as funções do admistrador no final é executada a função `guardar_produto` 5 onde todas as alterações são guardadas no ficheiro de texto `produtos.txt`.

16 Eliminar Produto

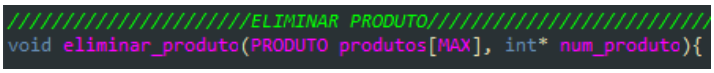

Figura 23: Função eliminar produto

Esta função permite ao administrador eliminar um determinado produto do stock atual da máquina, onde em primeiro lugar é solicitado o código que deseja retirar, em seguida inicia um `loop` na sequência percorrendo todos os produtos presentes da lista, e comparando assim o código introduzido com o código do produto, quando encontrado, todas as informações são redefinidas. Após removido o produto, a função `"guardar_produto"` (5) é chamada para guardar todas as alterações feitas na lista de produtos. Caso o código do produto não seja encontrado em nenhum dos produtos da lista, é exibida a mensagem ”Não encontrado o produto”.

17 Menu/utilizador administrador

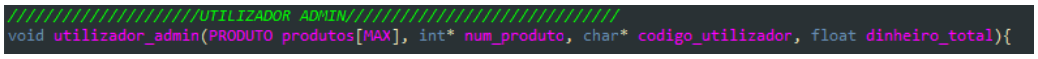

Figura 24: Função utilizador admin

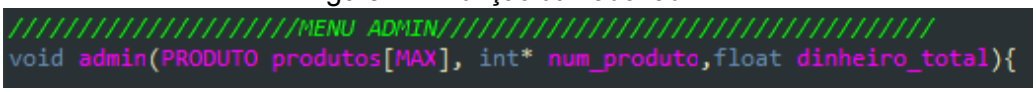

Figura 25: Função menu administrador

A função `"utilizador_admin"` redireciona usuário para o menu de administrador caso no inicio digite o código ”00”, aqui o administrador pode fazer toda a gestão da máquina , já a função `"admin` representa o menu de opções para o administrador, onde exibe um menu com várias opções, como adicionar produto, listar produtos, reabastecer a máquina, listar validade dos produtos, atualizar preço de um produto ou o preço de todos.

18 Conclusão

Durante a realização deste trabalho, encontramos algumas dificuldades que gostaríamos de relatar. Uma das principais dificuldades que enfrentamos foi a necessidade de criar um `loop` para solicitar os inputs aos usuários até que fossem inseridos corretamente, de acordo com as especificações requeridas. Após várias tentativas sem sucesso, procuramos a ajuda dos nossos colegas e professor, e juntos conseguimos resolver esse problema.

Uma dificuldade que encontramos foi como organizar o código de forma esteticamente apelativa e também como apresentar a lista de 48 produtos para o utilizador de uma forma que seria fácil para perceber que informações eram o tipo de produtos, data de validade, etc.. Graças ao `ChatGPT` consegui-nos ajudar a como fazer com que o texto ficasse todo alinhado com as colunas. Embora que seja um projeto onde não será usado pelo público geral, queríamos trazer certa parte de realismo para, também, de uma forma, treinar para as exigências do mercado de trabalho.

Em resumo, enfrentamos desafios ao longo do desenvolvimento deste trabalho. Pesquisamos bastante, consultamos nossas anotações e recorremos ao `chatGPT` e ao professor quando as dúvidas persistiam. No entanto, antes de tudo, aprendemos a importância de procurar ajuda.
Além disso, este projeto permitiu nos aplicar e aprofundar os conhecimentos adquiridos ao longo do semestre. As pesquisas realizadas para solucionar os problemas também nos proporcionaram a descoberta de novos conceitos e técnicas. Além disso, este projeto colaborativo trouxe uma valiosa experiência de trabalho em equipa.
No geral, este trabalho fez-nos enfrentar e superar desafios, ampliar os nossos conhecimentos e adquirir habilidades valiosas. Continuaremos a aplicar estes conhecimentos nos nossos futuros projetos, aproveitando ao máximo as oportunidades de aprendizado e crescimento profissional.