

## Estrutura de Dados

Instituto Politécnico do Cávado e do Ave  
Escola Técnica Superior Profissional  
CTeSP – Tecnologia e Inovação Informática  
Ano Letivo: 2021/2022



# Relatório Trabalho Prático

Joana Freitas Pimenta – 22999

Diogo Manuel Marques – 23000

João Gabriel Teixeira – 23013

Ludgero Miguel Simões – 23135

## Índice

Relatório.....	1
Trabalho Prático .....	1
Tema.....	3
Objetivos .....	3
Métodos.....	4
Estruturas e Listas .....	4
Funções .....	6
Conclusão .....	10

## Índice de Imagens

Imagem 1 - Funções implementadas .....	3
Imagem 2 – Estruturas .....	4
Imagem 3 - Elementos das Listas .....	5
Imagem 4 - Listas .....	5
Imagem 5 - Output inicial.....	6
Imagem 6 - Demonstração do Login .....	7
Imagem 7 – Demonstração Criar Conta .....	8
Imagem 8 - Menu Cliente.....	9
Imagem 9 - Menu Gestor .....	9
Imagem 10 - Logo C.....	10

## Tema

O tema proposto consiste em implementar uma solução de gestão de clientes, stock e compras de produtos de uma cadeia de supermercados, em contexto pandêmico, de maneira a prover uma forma de adaptação à situação, possibilitando garantir os bens necessários.

## Objetivos

Neste projeto, foi-nos pedido, em função de cimentar os conhecimentos adquiridos durante as aulas, para desenvolver uma aplicação que pode ser utilizada como “cliente” e como “gestor”, utilizando estruturas dinâmicas, com as seguintes funcionalidades:

Estrutura	Função	
	Gestão de clientes	Gestão de produtos
<b>Gestor</b>	Adicionar novos clientes	Adicionar novos produtos
	Remover clientes	Remover produtos
	Consultar clientes	Consultar produtos
<b>Cliente</b>	Adicionar novo conta de cliente	Comprar produtos
	Remover conta de cliente	

*Imagem 1 - Funções implementadas*

Foi-nos também proposto desenvolver algumas funcionalidades extra, que demonstrem sentido ao contexto do programa, de maneira a complementar e a torná-lo mais útil.

# Métodos

## Estruturas e Listas

Foram utilizadas uma série de estruturas e listas, pondo em prática os conceitos dados em âmbito de aula e permitindo assim que a aplicação fosse o mais otimizada possível.

```
typedef struct {
    int dia, mes, ano;
} Data;

typedef struct {
    int id; // Preenchido automaticamente com o valor do arquivo "FILE_ID_ACCOUNT"
    char username[30];
    char password[30];
    int tipoConta; // 1 - Cliente, 2 - Gestor
    int idTipo; // Guarda o Id do cliente/gestor dependendo do tipo da conta
    int createdByAdmin; // 0 ou null se não for criado pelo admin, 1 se tiver sido
} ContasLogin;

typedef struct {
    int numCliente; // Preenchido automaticamente com o valor do arquivo "FILE_ID_CLIENTE"
    char nome[30];
    char morada[50];
    int nContribuinte;
    char nCartaoCredito[16];
} Cliente;

typedef struct {
    int codGestor; // Preenchido automaticamente com o valor do Arquivo "FILE_ID_GESTION"
    char nome[30];
} Gestor;

typedef struct {
    int codProduto; // Preenchido automaticamente com o valor do Arquivo "FILE_ID_PRODUCT"
    char nome[30];
    float preco;
} Produto;

typedef struct {
    int id; // Preenchido automaticamente com o valor do Arquivo "FILE_ID_STOCK"
    int productID;
    Data validade;
    int quantidadeStock;
} Stock;

typedef struct {
    int idProduto;
    int quantidade;
} CarrinhoCompras;

typedef struct {
    int id; // Preenchido automaticamente com o valor do Arquivo "FILE_ID_COMPRA"
    Data dataCompra;
    int idCliente;
    float precoTotal;
} Compra;

typedef struct {
    int id;
    int idCompra;
    int idProduto;
    int quantidade;
} ItemsCompra;
//
```

Imagem 2 – Estruturas

```

// Elementos das Listas
typedef struct elementoContasLogin {
    ContasLogin contaLogin;
    struct elementoContasLogin *prox;
} ElementoContasLogin;

typedef struct elementoCliente {
    Cliente cliente;
    struct elementoCliente *prox;
} ElementoCliente;

typedef struct elementoGestor {
    Gestor gestor;
    struct elementoGestor *prox;
} ElementoGestor;

typedef struct elementoProduto {
    Produto produto;
    struct elementoProduto *prox;
} ElementoProduto;

typedef struct elementoStock {
    Stock stock;
    struct elementoStock *prox;
} ElementoStock;

typedef struct elementoCarrinhoCompras {
    CarrinhoCompras carrinhoCompras;
    struct elementoCarrinhoCompras *prox;
} ElementoCarrinhoCompras;

typedef struct elementoCompra {
    Compra compra;
    struct elementoCompra *prox;
} ElementoCompra;

typedef struct elementoItemsCompra {
    ItemsCompra itemsCompra;
    struct elementoItemsCompra *prox;
} ElementoItemsCompra;
//

```

*Imagem 3 - Elementos das Listas*

```

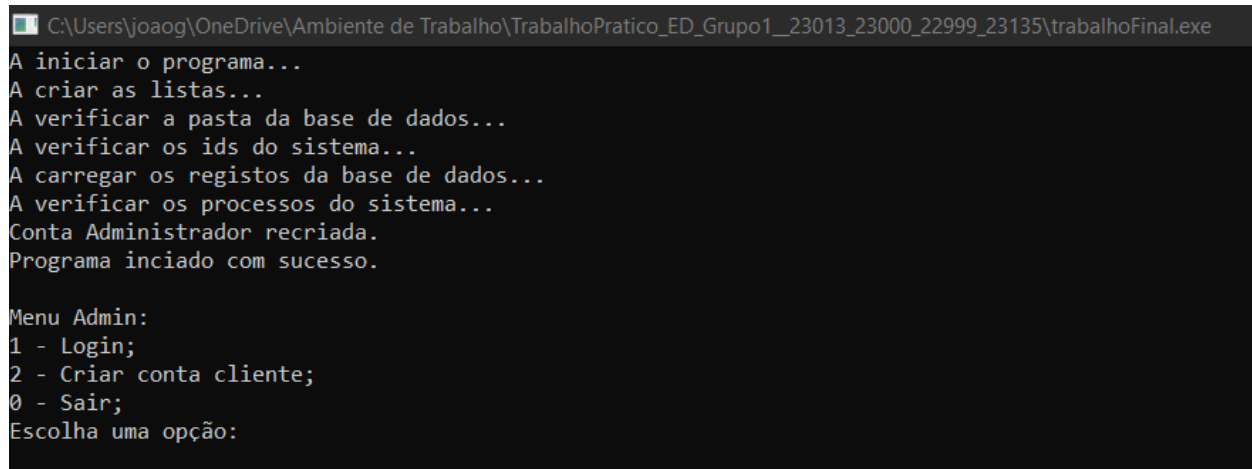
// Listas
typedef ElementoContasLogin* ListaContasLogin;
typedef ElementoCliente* ListaCliente;
typedef ElementoGestor* ListaGestor;
typedef ElementoProduto* ListaProduto;
typedef ElementoStock* ListaStock;
typedef ElementoCarrinhoCompras* ListaCarrinhoCompras;
typedef ElementoCompra* ListaCompra;
typedef ElementoItemsCompra* ListaItemsCompra;
//

```

*Imagem 4 - Listas*

## Funções

Ao executar a aplicação, a função *startProgram()* é chamada, e tem como objetivo iniciar todos os processos necessários para a execução da aplicação. A função permite também saber, através do *return*, se está tudo pronto e sem erros, retorna 1 para **sucesso** e 0 para **problema**.



```
C:\Users\joaog\OneDrive\Ambiente de Trabalho\TrabalhoPratico_ED_Grupo1_23013_23000_22999_23135\trabalhoFinal.exe
A iniciar o programa...
A criar as listas...
A verificar a pasta da base de dados...
A verificar os ids do sistema...
A carregar os registos da base de dados...
A verificar os processos do sistema...
Conta Administrador recriada.
Programa iniciado com sucesso.

Menu Admin:
1 - Login;
2 - Criar conta cliente;
0 - Sair;
Escolha uma opção:
```

Imagem 5 - Output inicial

O 'Menu Admin' permite aos clientes e aos gestores, 'Iniciar sessão', 'Criar uma conta Cliente' ou 'Sair da aplicação'.

Ao seleccionar a 1ª opção, será pedido ao utilizador as credenciais da sessão, através da função *Login()*, que verifica se os dados correspondem,. Ao introduzir a password, ela está oculta por motivos de privacidade e segurança, sendo apenas possível, através da função *lerPassowrd()*. Estando então tudo verificado o utilizador pode entrar prosseguir com a utilização da sua conta, caso contrário é pedido ao utilizador para introduzir novamente os dados, como apresenta a Imagem 6.

É possível também, criar uma nova conta, como demonstra a Imagem 7.

C:\Users\joaog\OneDrive\Ambiente de Trabalho\TrabalhoPratico\_ED\_Grupo1\_\_23013\_23000\_22999\_23135\trabalhoFinal.exe

A iniciar o programa...  
A criar as listas...  
A verificar a pasta da base de dados...  
A verificar os ids do sistema...  
A carregar os registos da base de dados...  
A verificar os processos do sistema...  
Programa iniciado com sucesso.

Menu Admin:

1 - Login;  
2 - Criar conta cliente;  
0 - Sair;

Escolha uma opção: 1

Introduza o username da conta: JonasG

Introduza a password da conta: \*\*\*\*\*

Falha ao entrar.

Pretende tentar novamente?[s/S]: s

Introduza o username da conta: JonasG

Introduza a password da conta: \*\*\*\*\*

Bem-Vindo(a) de volta Joao Gabriel.

Menu Cliente:

1 - Consultar dados;  
2 - Alterar dados;  
3 - Ver produtos;  
4 - Consultar produto;  
5 - Verificar carrinho de compras;  
6 - Comprar produto;  
7 - Alterar quantidade produto do carrinho de compras;  
8 - Remover produto do carrinho de compras;  
9 - Finalizar Compra;  
10 - Limpar carrinho;  
11 - Consultar compras feitas;  
12 - Apagar conta;  
0 - Logout;

Escolha uma opção:

Imagem 6 - Demonstração do Login

```
C:\Users\joaog\OneDrive\Ambiente de Trabalho\TrabalhoPratico_ED_Grupo1_23013_23000_22999_23135\trabalhoFinal.exe
A iniciar o programa...
A criar as listas...
A verificar a pasta da base de dados...
A verificar os ids do sistema...
A carregar os registos da base de dados...
A verificar os processos do sistema...
Programa iniciado com sucesso.

Menu Admin:
1 - Login;
2 - Criar conta cliente;
0 - Sair;
Escolha uma opção: 2
Criar conta do cliente:
Introduza o nome: Ludgero Miguel
Introduza a morada: Braga
Introduza o número de contribuinte: 291302394
Introduza o número do cartão de crédito: 4556437811903420
Introduza o username da conta: LuD
Introduza a password da conta: *****
Conta criada com sucesso.

Menu Admin:
1 - Login;
2 - Criar conta cliente;
0 - Sair;
Escolha uma opção:
```

*Imagem 7 – Demonstração Criar Conta*

Caso o utilizador selecione a terceira opção a função *endProgram()* liberta todos os processos e a aplicação é encerrada.

Após o Cliente efetuar o login, tem acesso á um menu como apresenta a [Imagem 8](#), onde é possível 'Consultar os meus dados', 'Alterar os meus dados', 'Visualizar catálogo', 'Consultar produto', 'Visualizar carrinho', 'Adicionar produto', 'Alterar quantidade de produtos', 'Remover produto do carrinho', 'Efetuar compra', 'Esvaziar carrinho', 'Histórico de compras' e 'Eliminar Conta'. Todas as funções são validadas e verificadas de forma a não ser possível burlar.



```
Menu Cliente:
1 - Consultar os meus dados;
2 - Alterar os meus dados;
3 - Visualizar catálogo;
4 - Consultar produto;
5 - Visualizar carrinho;
6 - Adicionar produto;
7 - Alterar quantidade de produtos;
8 - Remover produto do carrinho;
9 - Efetuar compra;
10 - Esvaziar carrinho;
11 - Histórico de compras;
12 - Eliminar conta;
0 - Logout;
Escolha uma opção:
```

*Imagem 8 - Menu Cliente*

Após o Gestor efetuar o login, tem acesso á um menu como apresenta a [Imagem 9](#), onde é possível 'Consultar os meus dados', 'Alterar os meus dados', 'Consultar clientes', 'Procurar cliente', 'Inserir cliente', 'Eliminar cliente', 'Consultar gestores', 'Procurar gestor', 'Inserir gestor', 'Eliminar gestor', 'Consultar produtos', 'Procurar produto', 'Inserir produto', 'Remover produto', 'Inserir stock', 'Consultar stock produto', 'Remover stock', 'Consultar compras', 'Procurar compra', 'Eliminar conta. Todas as funções são validadas e verificadas de forma a não ser possível burlar.

```
Menu gestor:
1 - Consultar dados;
2 - Alterar dados;
3 - Consultar clientes;
4 - Procurar cliente;
5 - Criar cliente;
6 - Remover cliente;
7 - Consultar gestores;
8 - Procurar gestor;
9 - Criar gestor;
10 - Remover gestor;
11 - Consultar produtos;
12 - Procurar produto;
13 - Adicionar produto;
14 - Remover produto;
15 - Adicionar stock;
16 - Consultar stock produto;
17 - Remover stock;
18 - Consultar compras;
19 - Procurar compra;
20 - Apagar conta;
0 - Logout;
Escolha uma opção:
```

*Imagem 9 - Menu Gestor*

## Conclusão

Embora tenha sido um trabalho árduo e complexo, que exigiu muita dedicação da parte do nosso grupo, consideramos que concluímos este projeto com bastante destreza. Sentimo-nos bastante satisfeitos conosco. Com este relatório podemos afirmar que atingimos todos os objetivos propostos.

De maneira geral, pusemos em prática os conhecimentos adquiridos em contexto de aula, aprofundando os mesmos, assim como adquirindo novos conhecimentos através de pesquisas e esclarecimento de dúvidas.

Algumas das dificuldades enfrentadas foram: A implementação da função ler Password(), pois a função apagava mais texto do que o necessário; Na função que valida a data de validade do produto e Esconder os primeiros 14 números do cartão de crédito.



*Imagem 10 - Logo C*