

MANUAL DO PROGRAMADOR

António Matos

TGPSI 1L 20/23

CONEXÃO À BASE DE DADOS (MYSQL)

- Os dados da conexão à base de dados, podem ser alterados dentro do código, nas primeiras linhas, o username, a password, e a base de dados. A host da base de dados pode ser alterada através do ficheiro "ficheiro.txt".

```
char *server = host;
char *user = "root";
char *pass = "5603";
char *database = "psi";
con = mysql_init(NULL);
if(!mysql_real_connect(con, server, user, pass, database, 0, NULL, 0))
{
    fprintf(stderr, "%s\n", mysql_error(con));
}
```

MENU PRINCIPAL

- No menu principal, ele só vai dar print da secção de administrador caso a query seja executada com sucesso, ou seja caso o utilizador esteja com a coluna "grupo" marcada como "1" na base de dados. Se estiver "0", esse "printf" não aparecerá.

```
printf("+-----+\\n");
printf("| " Vermelho "Bem-vindo, escolha uma das opções abaixo. "Amarelo" |\\n");
printf("+-----+\\n");
printf("| 1 - "Ciano"Solicitar cartão" "Amarelo" |\\n");
printf("| 2 - "Ciano"Verificar dados/saldo conta bancária "Amarelo" |\\n");
printf("| 3 - "Ciano"Verificar dados cartão" "Amarelo" |\\n");
printf("| 4 - "Ciano"Transferir dinheiro" "Amarelo" |\\n");
printf("| 5 - "Ciano"Logout" "Amarelo" |\\n");
printf("| 6 - "Ciano"Sair" "Amarelo" |\\n");
if(mysql_fetch_row(res) != NULL)
{
    printf("| 7 - "Ciano"Admin menu" "Amarelo" |\\n");
}
```

GERADOR DE CARTÃO

- O gerador de cartão, funciona com rand, ou seja, demora para gerar. A query teve de ser dividida em 2, foi uma string não tem capacidade suficiente para aguentar tantos caracteres.

```
sprintf(mysqlvalues.query3, "UPDATE `contas` SET Cartao_solicitado = 1, Cartao_codigo = '%s', Cartao_entidade = '%i%i' WHERE Username = '%s';",  
        codigocartaochar, entidade1, entidade2, login.username);  
if(mysql_query(con, mysqlvalues.query3))  
{  
    printf("VermelhoVermelhoQuery failed: %s\n", mysql_error(con));  
    exit(0);  
}  
sprintf(mysqlvalues.query4, "UPDATE `contas` SET Cartao_referencia = '%s', Cartao_numero = '%i%i%i%i', Cartao_ccv = '%s' WHERE Username = '%s';",  
        referencialchar, cartao1, cartao2, cartao3, cartao4, cvvchar, login.username);  
if(mysql_query(con, mysqlvalues.query4))  
{  
    printf("VermelhoVermelhoQuery failed: %s\n", mysql_error(con));  
    exit(0);  
}
```

REGISTO DO UTILIZADOR

- No registo do utilizador, o Iban é automaticamente gerado, ou seja, mais uma vez, a conta do utilizador demorará a ser criada. É utilizado um vetor para armazenar 5 valores, pois o Iban não cabe num int, já que o int só consegue armazenar até 35.676,00.

```
int iban[4];
for(int i = 0; i < 5; i++)
{
    iban[i] = (rand() % (9999 + 1 - 1000)) + 1000;
    sleep(5);
}
int iban1 = (rand() % (999 + 1 - 100)) + 100;
sprintf(mysqlvalues.query6, "INSERT INTO contas (username, password, iban) VALUES ('%s', '%s', 'PT%i%i%i%i%i%i');",
        registo.username, registo.password, iban[0], iban[1], iban[2], iban[3], iban[4], iban1);
if(mysql_query(con, mysqlvalues.query6))
{
    printf("Vermelho"Query failed: %s\n", mysql_error(con));
    exit(0);
}
```

PRIVADO.H

- É um header desenvolvido por mim ao longo do tempo de forma a facilitar o código e a aumentar a eficiência. Meu código que me facilita, imports para não ocupar espaço no código main.c. O sistema de cores, não utilizei cores da biblioteca conio.h. Utilizei o meu próprio sistema de cores "escape colors". É necessário chamar a função "vtlfunc" uma vez, depois pode ser removido do código, serve somente para ativar uma funcionalidade de leitura deste tipo de cores, uma simples modificação no regedit.

```
//Início do Código
```

```
#define Reset   "\x1B[0m"  
#define Vermelho "\x1B[31m"  
#define Verde   "\x1B[32m"  
#define Amarelo "\x1B[33m"  
#define Azul    "\x1B[34m"  
#define Magenta "\x1B[35m"  
#define Ciano   "\x1B[36m"  
#define Branco  "\x1B[37m"
```

```
void title(nometitle)  
{  
    char msg[60];  
    sprintf(msg, "Executando o programa %s...", nometitle);  
    SetConsoleTitle(msg);  
}
```

```
void loadingfunc()  
{  
    printf("A carregar, por favor aguarde");  
    SetConsoleTitle("Loading - [!]");  
    printf(".");  
    sleep(1);  
    SetConsoleTitle("Loading - [/]");  
    printf(".");  
    sleep(1);  
    SetConsoleTitle("Loading - [-]");  
    printf(".");  
    sleep(1);  
    SetConsoleTitle("Loading - [\\]");  
    sleep(1);  
}
```

```
void fullscreen()  
{  
    ShowWindow(GetConsoleWindow(), SW_SHOWMAXIMIZED);  
}
```

```
//Início dos Imports
```

```
#include <stdlib.h>  
#include <stdio.h>  
#include <windows.h>  
#include <math.h>  
#include <string.h>  
#include <locale.h>  
#include <stddef.h>  
#include <stdbool.h>  
#include <mysql.h>
```

```
void cls()  
{  
    system("cls");  
}
```

```
bool startsWith(const char *haystack, const char *needle)  
{  
    for (size_t i = 0; needle[i] != '\0'; i++)  
    {  
        if (haystack[i] != needle[i])  
        {  
            return false;  
        }  
    }  
    return true;  
}
```