

## sprint.c

```
1  Sprint 1.1 :
2
3  #include <stdio.h>
4  #include <string.h>
5  #include <stdlib.h>
6  #include <locale.h>
7  #include <conio.h>
8  #include <time.h>
9  #include <ctype.h>      // para usar isalpha, digit, isspace, tolower
10 #include <stdbool.h>    // para usar booleano
11 #define MAX_USUARIOS 30 // quantidade maxima de usuarios
12 #define MAX_NOME 30     // tamanho maximo do nome
13 #define MAX_SENHA 20    // tamanho maximo da senha
14 #define MAX_VENDAS 200  // tamanho maximo do relatorio de vendas
15 #define MAX_LINHA 100   // tamanho maximo do buffer de leitura de uma linha de arquivo
16 #define MAX_PRODUTO 100 // quantidade maxima de produtos cadastrados
17 int idSessao = 0;       // variavel para controle de "sessao", identifica o usuario logado
18 int position;          // variavel de controle
19 int excedido = 0;       // variavel de controle
20 int idUsuario = 0;      // inicia com valor 0 antes da leitura dos usuarios ser feita e atribuida
    ao id
21 char senhaADM[] = "1234";
22 char senhaAdmin[MAX_SENHA];
23 char adm[] = "adm";
24 char caixa[] = "caixa";
25 typedef struct
26 { // estrutura de variaveis atribuidas a um usuario
27     char nome[MAX_NOME];
28     char senha[MAX_SENHA];
29     char cargo[MAX_NOME];
30     int id;
31 } estrutura;
32 estrutura user[MAX_USUARIOS]; // array com os valores da estrutura
33 int limpar_tela()              // funcao para limpar a tela
34 {
35     system("cls"); // comando do Windows para limpar a tela
36     return 1;
37 }
38 void limpabuffer() // funcao para limpar o buffer apos um endereco de memoria ser escrito
39 {
40     int c;
41     while ((c = getchar()) != '\n' && c != EOF)
42     {
43         // corpo vazio para limpar o endereco de memoria
44     }
45 }
46 void toLowerCase(char *str) // transforma maiusculo em minusculo
47 {
48     int i = 0;
49     while (str[i] != '\0')
50     {
51         str[i] = tolower(str[i]);
```

```
52     i++;
53 }
54 }
55 void remove_espacos(char *str) // remove espacos de uma string
56 {
57     char *dest = str; // ponteiro para o destino da string sem espacos
58     while (*str)
59     {
60         if (!isspace((unsigned char)*str)) // verifica se o caractere nao e um espaco
61         {
62             *dest++ = *str; // copia o caractere nao espaco para o destino
63         }
64         str++; // avanca para o proximo caractere
65     }
66     *dest = '\\0'; // adiciona o caractere nulo no final da nova string
67 }
```