

## Sprint 1.1:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <locale.h>
#include <conio.h>
#include <time.h>
#include <ctype.h> // para usar isalpha, digit, isspace, tolower
#include <stdbool.h> // para usar booleano

#define MAX_USUARIOS 30 // quantidade maxima de usuarios
#define MAX_NOME 30 // tamanho maximo do nome
#define MAX_SENHA 20 // tamanho maximo da senha
#define MAX_VENDAS 200 // tamanho maximo do relatorio de vendas
#define MAX_LINHA 100 // tamanho maximo do buffer de leitura de uma linha de arquivo
#define MAX_PRODUTO 100 // quantidade maxima de produtos cadastrados

int idSessao = 0; // variavel para controle de "sessao", identifica o usuario logado
int position; // variavel de controle
int excedido = 0; // variavel de controle
int idUsuario = 0; // inicia com valor 0 antes da leitura dos usuarios ser feita e atribuida ao id

char senhaADM[] = "1234";
char senhaAdmin[MAX_SENHA];
char adm[] = "adm";
char caixa[] = "caixa";

typedef struct
{ // estrutura de variaveis atribuidas a um usuario
    char nome[MAX_NOME];
    char senha[MAX_SENHA];
    char cargo[MAX_NOME];
    int id;
} estrutura;

estrutura user[MAX_USUARIOS]; // array com os valores da estrutura

int limpar_tela() // funcao para limpar a tela
{
    system("cls"); // comando do Windows para limpar a tela
    return 1;
}

void limpabuffer() // funcao para limpar o buffer apos um endereco de memoria ser escrito
{
    int c;
    while ((c = getchar()) != '\n' && c != EOF)
```

```
{  
    // corpo vazio para limpar o endereco de memoria  
}  
}
```

```
void toLowerCase(char *str) // transforma maiusculo em minusculo  
{  
    int i = 0;  
    while (str[i] != '\0')  
    {  
        str[i] = tolower(str[i]);  
        i++;  
    }  
}
```

```
void remove_espacos(char *str) // remove espacos de uma string  
{  
    char *dest = str; // ponteiro para o destino da string sem espacos  
    while (*str)  
    {  
        if (!isspace((unsigned char)*str)) // verifica se o caractere nao e um espaco  
        {  
            *dest++ = *str; // copia o caractere nao espaco para o destino  
        }  
        str++; // avanca para o proximo caractere  
    }  
    *dest = '\0'; // adiciona o caractere nulo no final da nova string  
}
```