



DMS

DONATION MANAGEMENT SYSTEM

NOSSA EQUIPE



COLABORADORES

**FRANCISCO
MENEGUINI**

Responsável por desenvolver a carta de envolvimento externo para comprovação de relação, criação da EAP e criação do fluxograma.

**VITÓRIA
AFONSO**

Responsável em realizar relacionamento com organizações sociais e semelhantes, com intuito de envolvimento no projeto, criação de dicionário da EAP.

**ALEX
NOGUEIRA**

Responsável pela criação do documento plano de trabalho, e auxílio no desenvolvimento do Template de requisitos funcionais.

COLABORADORES

IGOR MODESTO

Responsável pela criação e bom funcionamento do Código/Sistema, criação do documento de Template de requisitos funcionais, e criação do cronograma.

MARIA LUIZA MELO

Responsável em desenvolver documento de apresentação de acordo o projeto realizado, e auxílio na criação do fluxograma.

INTRODUÇÃO

ORGANIZAÇÕES E CENTROS DE DOAÇÕES
ENFRENTAM DESAFIOS NA GESTÃO DE ESTOQUE
DEVIDO À FALTA DE UM SISTEMA INTEGRADO,
RESULTANDO EM UMA PERCA DE INFORMAÇÃO E PERCA
DE CONTROLE NO ESTOQUE

TENDO EM BASE AS INFORMAÇÕES SOBRE OS
DESAFIOS E PROBLEMAS EM GESTÃO INICIAMOS O
PROJETO DE EXTENSÃO COM O INTUITO DE CRIAR UM
AUXILIADOR PARA ESSAS ONG'S E CENTROS DE
DOAÇÕES.

OBJETIVOS

- 1 *Controle do estoque que mostre quais são as necessidades no momento para os usuários;*
- 2 *Auxiliar na administração, guardando informações digitalmente.*



JUSTIFICATIVA

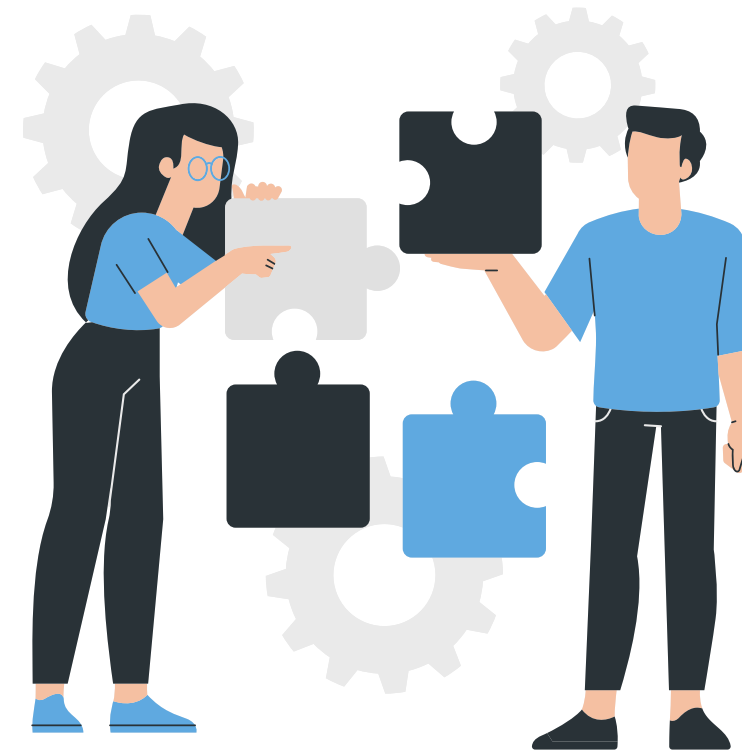
SOBRE QUAL ASSUNTO SE TRATA NOSSO TRABALHO?

- CONTROLE DE ESTOQUE;
- CONTROLE DE DESPERDÍCIO;
- USABILIDADE;



PROBLEMAS

- 1 *Processos manuais;*
- 2 *Falta de controle dos produtos doados;*
- 3 *Má gestão em relação ao acompanhamento de armazenamento de produtos;*
- 4 *Desvios de dinheiro ou produtos.*



BENEFÍCIOS ESPERADOS

- Ter um melhor controle das demandas das doações;
- Evitar desvio de dinheiro e doações;



CRITÉRIOS DE AVALIAÇÃO

● SATISFAÇÃO DAS ONGS E PARCEIROS:

Coleta-se feedback das ONGs sobre a eficácia do sistema e a satisfação geral com ele.

● CUMPRIMENTO DE PRAZOS:

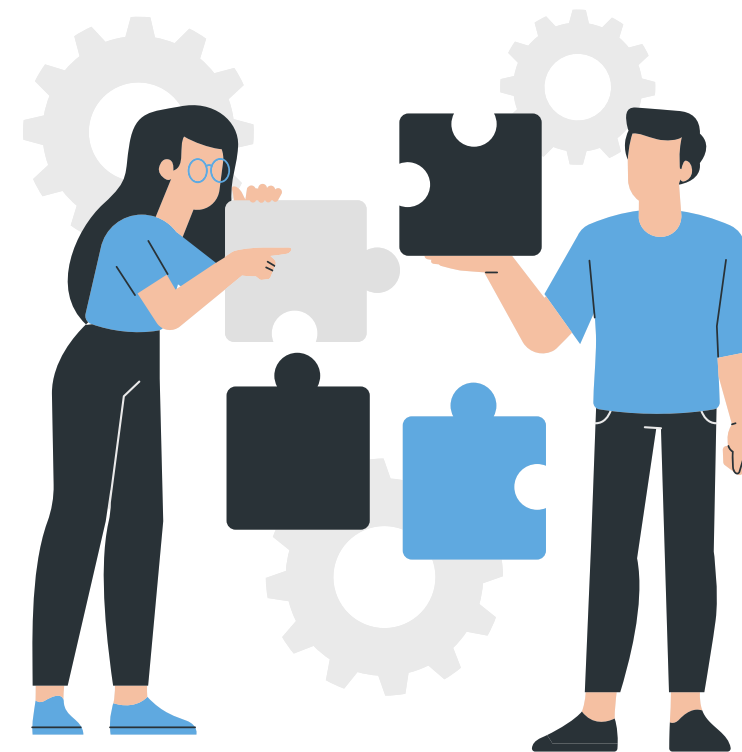
Verifica-se se o projeto foi entregue dentro do prazo, avaliando a gestão do projeto.

Critérios de Sucesso

Atividade	Não		Desafios
	Cumprida	Cumprida	
Criação do termo de abertura	X		
EAP	X		
Cronograma	X		
Fluxograma	X		
Código em C	X		
Dicionário EAP	X		
Carta de envolvimento	X		
Criação do plano de trabalho	X		
Prova de conceito	X		
Criação de documento de apresentação	X		

Conceitos usados

- 1 *Decisão*
- 2 *Repetição*
- 3 *Vetor*
- 4 *Funções*
- 5 *Arquivos*



CRIAÇÃO DA STRUCT

```
// Estrutura para representar um produto
typedef struct {
    char nome[50];
    int quantidade;
    char dataValidade[11]; // DD/MM/AAAA
    char doador[50];
} Produto;
```

ESTRUTURA DE TEXTO

```
void exibirLogin() {
    printf("=====\n");
    printf("          TELA DE LOGIN          \n");
    printf("=====\n");
}

void exibirMenu() {
    printf("\n=====\n");
    printf("          MENU PRINCIPAL          \n");
    printf("=====\n");
    printf("1. Cadastrar produto\n");
    printf("2. Imprimir estoque\n");
    printf("3. Remover produto\n");
    printf("4. Sair\n");
    printf("Escolha uma opcao: ");
}
```

FUNÇÕES

```
// Função para salvar produtos em um arquivo
void salvarProdutos(Produto produtos[], int numProdutos) {
    FILE *file = fopen("produtos.txt", "w");
    if (file == NULL) {
        printf("Erro ao abrir o arquivo para escrita.\n");
        return;
    }
    for (int i = 0; i < numProdutos; i++) {
        fprintf(file, "%s,%d,%s,%s\n", produtos[i].nome, produtos[i].quantidade, produtos[i].dataValidade, produtos[i].doador);
    }
    fclose(file);
}
```

FUNÇÕES

```
// Função para imprimir o estoque
void imprimirEstoque(Produto produtos[], int numProdutos) {
    printf("=====\n");
    printf("          ESTOQUE ATUAL          \n");
    printf("=====\n");
    for (int i = 0; i < numProdutos; i++) {
        printf("Produto %d:\n", i + 1);
        printf("Nome: %s\n", produtos[i].nome);
        printf("Quantidade: %d\n", produtos[i].quantidade);
        printf("Data de Validade: %s\n", produtos[i].dataValidade);
        printf("Doador: %s\n", produtos[i].doador);
        printf("-----\n");
    }
}
```

FUNÇÕES

```
// Função para carregar produtos de um arquivo
void carregarProdutos(Produto produtos[], int *numProdutos) {
    FILE *file = fopen("produtos.txt", "r");
    if (file == NULL) {
        printf("Nenhum arquivo de produtos encontrado. Iniciando com estoque vazio.\n");
        return;
    }
    while (fscanf(file, "%49[^,],%d,%10[^,],%49[^\n]\n", produtos[*numProdutos].nome, &produtos[*numProdutos].quantidade,
        (*numProdutos)++);
    }
    fclose(file);
}
```


FUNÇÕES

```
void limparTela() {  
    system("clear");  
}  
void esperarTecla() {  
    printf("Pressione Enter para continuar...");  
    getchar(); // Aguarda a entrada do usuário  
    getchar(); // Aguarda a entrada do usuário  
}
```

FUNÇÕES

```
// Função para remover uma quantidade específica de um produto
void removerProduto(Produto produtos[], int *numProdutos) {
    char nome[50];
    int quantidadeRemover;
    int i;
    // Solicita o nome do produto a ser removido e a quantidade
    printf("Digite o nome do produto: ");
    scanf("%49s", nome);
    printf("Digite a quantidade a ser removida: ");
    scanf("%d", &quantidadeRemover);
    // Procura o produto pelo nome
    for (i = 0; i < *numProdutos; i++) {
        if (strcmp(produtos[i].nome, nome) == 0) {
            // Verifica se a quantidade a ser removida é maior do que a disponível
            if (quantidadeRemover > produtos[i].quantidade) {
                printf("Quantidade a ser removida é maior do que a disponível.\n");
                return;
            } else if (quantidadeRemover == produtos[i].quantidade) {
                // Remove o produto se a quantidade a ser removida for igual à disponível
                for (int j = i; j < *numProdutos - 1; j++) {
                    produtos[j] = produtos[j + 1];
                }
                (*numProdutos)--; // Reduz o número de produtos
                printf("Produto removido com sucesso.\n");
                return;
            } else {
                // Reduz a quantidade do produto
                produtos[i].quantidade -= quantidadeRemover;
                printf("Quantidade removida com sucesso. Nova quantidade: %d\n", produtos[i].quantidade);
                return;
            }
        }
    }
    // Se o produto não for encontrado
    printf("Produto não encontrado.\n");
}
```

SWITCH CASE

```
do {
    limparTela();
    exibirMenu();
    scanf("%d", &opcao);
    switch (opcao) {
        case 1:
            cadastrarProduto(produtos, &numProdutos);
            salvarProdutos(produtos, numProdutos); // Salvar após o cadastro
            esperarTecla();
            break;
        case 2:
            imprimirEstoque(produtos, numProdutos);
            esperarTecla();
            break;
        case 3:
            removerProduto(produtos, &numProdutos);
            salvarProdutos(produtos, numProdutos); // Salvar após a remoção
            esperarTecla();
            break;
        case 4:
            printf("Saindo...\n");
            break;
        default:
            printf("ERRO! Código inserido inválido.\n");
            esperarTecla();
            break;
    }
} while (opcao != 4);
return 0;
```

SISTEMA DE LOGIN

```
int main() {
    while (tentativas > 0) {
        printf("Digite o nome de usuario: ");
        scanf("%19s", usuario); // Limita o tamanho da entrada
        printf("Digite a senha: ");
        scanf("%19s", senha); // Limita o tamanho da entrada

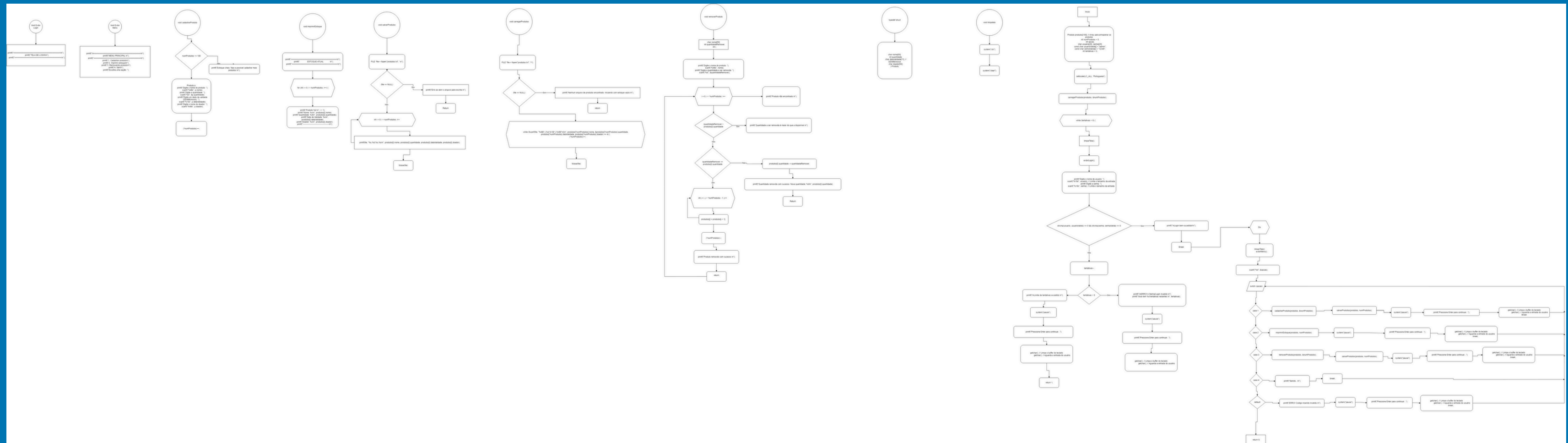
        if (strcmp(usuario, usuarioValido) == 0 && strcmp(senha, senhaValida) == 0) {
            printf("\nLogin bem-sucedido!\n");
            break;
        } else {
            tentativas--;
            if (tentativas > 0) {
                printf("\nERRO! A Senha/Login invalido.\n");
                printf("Voce tem %d tentativas restantes.\n", tentativas);
                #ifdef _WIN32
                    system("pause");
                #else
                    printf("Pressione Enter para continuar...");
                    getchar(); // Limpa o buffer do teclado
                    getchar(); // Aguarda a entrada do usuário
                #endif
            }
        }
    }

    if (tentativas == 0) {
        return 1; // Encerra o programa com erro
    }
}
```

```
    } else {
        printf("\nLimite de tentativas excedido.\n");
        #ifdef _WIN32
            system("pause");
        #else
            printf("Pressione Enter para continuar...");
            getchar(); // Limpa o buffer do teclado
            getchar(); // Aguarda a entrada do usuário
        #endif
    }
}

if (tentativas == 0) {
    return 1; // Encerra o programa com erro
}
```

CRIAÇÃO DO FLUXOGRAMA



OBRIGADO!

ALUNOS

ALEX NOGUEIRA
FRANCISCO MENEGUINI
IGOR MODESTO
MARIA LUIZA MELO
VITÓRIA AFONSO