

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define TAM_MAX_STR 40
#define TAM_MAX_RGT 3
#define MAX_FAZENDAS 100
#define TAM_MAX_ANIMAL 20

typedef struct _Fazenda {
    unsigned short int matricula;
    char dono[TAM_MAX_STR];
    char animais[TAM_MAX_RGT][TAM_MAX_ANIMAL];
} Fazenda;

Fazenda fazendas[MAX_FAZENDAS];
int contadorFazendas = 0;

void cadastrarFazendas();
void listarFazendas();
void atualizarFazenda();
void gravarDados();
void excluirFazenda();
void sair();

void enter();
int menu();
void opcao_menu(int opcao);

int main() {
    int opcao_escolhida;

    do {
        opcao_escolhida = menu();
        opcao_menu(opcao_escolhida);
    } while (opcao_escolhida != 6);

    return 0;
}

int menu() {
    int opcao;

    printf("====MENU====\n");
    printf("1 - Cadastrar nova fazenda: \n");
    printf("2 - Listar Fazendas cadastradas: \n");
    printf("3 - Atualizar dados da Fazenda: \n");
    printf("4 - Excluir Fazenda do registro: \n");
    printf("5 - Gravar dados \n");
    printf("6 - Sair\n");

    printf("Digite uma opção: ");

```

```

scanf("%d", &opcao);
getchar();

return opcao;
}

void opcao_menu(int opcao) {
    switch (opcao) {
        case 1:
            cadastrarFazendas();
            break;
        case 2:
            listarFazendas();
            break;
        case 3:
            atualizarFazenda();
            break;
        case 4:
            excluirFazenda();
            break;
        case 5:
            gravarDados();
            break;
        case 6:
            sair();
            break;
        default:
            printf("Digite uma opção válida!\n");
            break;
    }
}

void enter() {
    printf("Pressione ENTER para continuar...");
    getchar();
}

void cadastrarFazendas() {
    if (contadorFazendas >= MAX_FAZENDAS) {
        printf("Capacidade máxima de fazendas atingida!\n");
        enter();
        return;
    } else {
        Fazenda f;
        printf("Informe a matrícula da fazenda: ");
        scanf("%hu", &f.matricula);
        getchar();

        printf("Informe o nome do dono da fazenda: ");
        fgets(f.dono, TAM_MAX_STR, stdin);
        f.dono[strcspn(f.dono, "\n")] = 0; // Remove a nova linha
    }
}

```

```

        for (int i = 0; i < TAM_MAX_RGT; i++) {
            printf("Informe o nome do animal %d: ", i + 1);
            fgets(f.animais[i], TAM_MAX_ANIMAL, stdin);
            f.animais[i][strcspn(f.animais[i], "\n")] = 0; // Remove a nova linha
        }

        fazendas[contadorFazendas++] = f;
        printf("Fazenda cadastrada com sucesso!\n");
        enter();
    }
}

void listarFazendas() {
    if (contadorFazendas == 0) {
        printf("Nenhuma fazenda cadastrada.\n");
        enter();
        return;
    } else {
        printf("Listando Fazendas:\n");
        for (int i = 0; i < contadorFazendas; i++) {
            printf("Matrícula: %hu | Dono: %s | Animais: ",
                fazendas[i].matricula, fazendas[i].dono);
            for (int j = 0; j < TAM_MAX_RGT; j++) {
                printf("%s", fazendas[i].animais[j]);
                if (j < TAM_MAX_RGT - 1) {
                    printf(", ");
                }
            }
            printf("\n");
        }
        enter();
    }
}

void atualizarFazenda() {
    unsigned short int matricula;

    printf("Informe a matrícula da fazenda que deseja atualizar: ");
    scanf("%hu", &matricula);
    getchar();

    int is_fazenda = 0;
    for (int i = 0; i < contadorFazendas; i++) {
        if (fazendas[i].matricula == matricula) {
            is_fazenda = 1;
            printf("Atualizando fazenda do dono %s:\n", fazendas[i].dono);
            printf("Novo nome do dono: ");
            fgets(fazendas[i].dono, TAM_MAX_STR, stdin);
            fazendas[i].dono[strcspn(fazendas[i].dono, "\n")] = 0; // Remove a nova linha

            for (int j = 0; j < TAM_MAX_RGT; j++) {
                printf("Novo nome do animal %d: ", j + 1);
            }
        }
    }
}

```

```

        fgets(fazendas[i].animais[j], TAM_MAX_ANIMAL, stdin);
        fazendas[i].animais[j][strcspn(fazendas[i].animais[j], "\n")] = 0; // Remove a nova linha
    }

    printf("Fazenda atualizada com sucesso!\n");
    enter();
    break;
}
}

if (!is_fazenda) {
    printf("Fazenda não encontrada!\n");
    enter();
}
}

void excluirFazenda() {
    unsigned short int matricula;
    printf("Informe a matrícula da fazenda que deseja remover: ");
    scanf("%hu", &matricula);

    int is_fazenda = 0;
    for (int i = 0; i < contadorFazendas; i++) {
        if (fazendas[i].matricula == matricula) {
            is_fazenda = 1;
            for (int j = i; j < contadorFazendas - 1; j++) {
                fazendas[j] = fazendas[j + 1];
            }
            contadorFazendas--;
            printf("Fazenda removida com sucesso!\n");
            break;
        }
    }

    if (!is_fazenda) {
        printf("Fazenda não encontrada!\n");
    }
}

void gravarDados() {
    FILE *arquivo = fopen("fazendas.txt", "w");
    if (arquivo != NULL) {
        for (int i = 0; i < contadorFazendas; i++) {
            fprintf(arquivo, "%hu %s", fazendas[i].matricula, fazendas[i].dono);
            for (int j = 0; j < TAM_MAX_RGT; j++) {
                fprintf(arquivo, " %s", fazendas[i].animais[j]);
            }
            fprintf(arquivo, "\n");
        }
        fclose(arquivo);
        printf("Dados salvos com sucesso!\n");
    } else {

```

```
        printf("Erro ao salvar os dados!\n");
    }
}

void sair() {
    printf("Encerrando.....\n");
    printf("Programa finalizado!\n");
    exit(0); // Finaliza o programa
}
```