

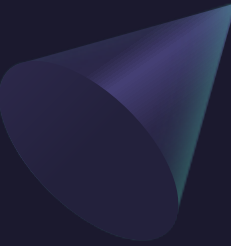
# Cadastro e avaliação de times

Desenvolvido por Kéwen Silva

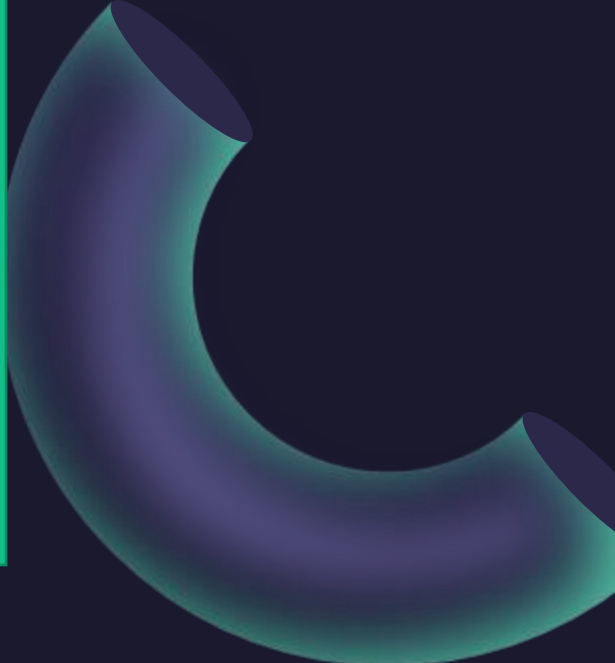
```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
  
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_objects  
data.objects[one.name].select  
print("please select exactly  
  
-- OPERATOR CLASSES --  
  
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
  
context):  
context.active_object is not
```



# Bibliotecas utilizadas



Além da biblioteca padrão `<stdio.h>`, foram utilizadas mais duas bibliotecas, são elas:

- ❑ `<string.h>` Biblioteca de funções relacionadas a strings
  - ❑ `<locale.h>` Biblioteca que traz a função `setlocale` que é onde passaremos os parâmetros (`LC_ALL`, `"Portuguese"`), que basicamente trazem o padrão de caracteres utilizado na língua portuguesa.
- 

# O código

```
void main(){
    setlocale(LC_ALL, "Portuguese");

    int op, cont, ctime, check, flag, flag2, direto;
    cont = ctime = flag = flag2 = direto = 0;
    char sair, auxNome[30], aux, op2;
    float calculo, caloria;

    struct jogador{
        char jogador[30], posicao[30], tabIMC[30];
        int idade;
        float peso, altura, imc;
    };

    struct time {
        char nome[30];
        struct jogador jogadores[20];
    } times[20];
```

Inicialmente, vale pontuar que o programa foi desenvolvido por completo na função main do tipo void.

Primeiramente, é passado o padrão de caracteres utilizado na língua portuguesa através da função setlocale.

Logo em seguida, começam-se as declarações de variáveis que serão necessárias para o funcionamento do programa. Ao longo do programa veremos estas sendo utilizadas, mas no momento, daremos uma atenção especial para as structs.

Na primeira struct, do tipo 'jogador', temos um conjunto de variáveis que irão compor as informações individuais dos jogadores cadastrados, como nome, peso altura etc.

Na segunda struct, do tipo 'time', teremos uma variável do tipo char que guardará os nomes dos times e um vetor do tipo jogador que guardará, no máximo, 20 jogadores por time.

# Menu de cadastro

```
printf("1 - Inserir time\n0 - Sair");  
printf("\nSelecione a opção desejada: ");  
scanf("%d", &op);  
system("cls");
```

Nesta parte é exibido um menu que inicialmente fornece apenas duas opções. Após coletado o valor da opção desejada, é feita uma limpeza no console através da função `system('cls')`

# Opção 1

```
switch(op){
    case 1:
        printf("Digite o nome do time: ");
        fflush(stdin);
        gets(times[ctime].nome);
        printf("\n");
        do{
            printf("Digite o nome do jogador: ");
            fflush(stdin);
            gets(times[ctime].jogadores[cont].jogador);

            printf("Digite a idade: ");
            scanf("%d", &times[ctime].jogadores[cont].idade);

            printf("Digite o peso em kg: ");
            scanf("%f", &times[ctime].jogadores[cont].peso);

            printf("Digite a altura em metros: ");
            scanf("%f", &times[ctime].jogadores[cont].altura);

            printf("Digite a posição em que o jogador atua: ");
            fflush(stdin);
            gets(times[ctime].jogadores[cont].posicao);

            times[ctime].jogadores[cont].imc = times[ctime].jogadores[cont].peso
            / pow(times[ctime].jogadores[cont].altura,2);

            printf("Deseja continuar? <s|n> ");
            scanf("%c", &sair);
            system("cls");
            cont++;
        } while(sair == 's' || sair == 'S');
        ctime++;
        cont = 0;
    break;
}
```

Na primeira opção faz-se o cadastro do time e seus jogadores. Nesta etapa, é possível cadastrar até 20 jogadores que é o máximo de jogadores permitidos por time. O loop encerra após o usuário responder 'n' na pergunta "Deseja continuar?".

Temos também a presença de dois contadores, são eles:

**ctime** -> Vai definir a posição em que serão inseridos os times no vetor struct times e também servirá para sabermos a quantidade de times cadastrados;

**cont** -> Vai definir a posição em serão inseridos os jogadores no vetor de struct jogadores dentro do do vetor times.

Também, nessa mesma opção, temos o cálculo do índice de massa corporal do jogador.

Após sair da opção 1, o usuário é questionado se deseja voltar ao menu de cadastro ou se deseja avançar para o menu de estatísticas

```
printf("\nDeseja avançar para o menu de estatísticas e informações ou voltar ao menu de cadastro? <1 - 2> ");
scanf("%d", &check);
fflush(stdin);
system("cls");
```

# Menu de cadastro completo

```
printf("1 - Inserir novo time\n2 - Modificar nome do time\n3 - Inserir novos jogadores em um time\n4 - Modificar todos os dados do jogador\n");  
printf("5 - Modificar dados especificos do jogador\n6 - Avançar para o menu de estatísticas\n0 - Sair");  
printf("\nEscolha o esporte: ");  
scanf("%d", &op);  
system("cls");
```

Agora o menu de cadastro será exibido e serão acrescentadas mais 5 opções

# Opção 2

```
case 2:
    printf("\nDigite o time que deseja alterar o nome: ");
    fflush(stdin);
    gets(auxNome);
    for(int x = 0; x < ctime; x++){
        if(strcmp(auxNome, times[x].nome) == 0){
            flag = 1;
            printf("\nDigite o novo nome do time: ");
            fflush(stdin);
            gets(times[x].nome);
        }
    }
    if(!flag)
        printf("\nEste time não está cadastrados no banco de dados");
    break;
```

Na opção 2 será solicitado que o usuário digite o nome do time que ele deseja editar. Assim, que esse nome for fornecido será verificado se o time já foi cadastrado, caso sim, será disponibilizada opção do usuário inserir o novo nome do time.



# Opção 3

```
case 3:
printf("\nDigite o time em que deseja inserir novos jogadores: ");
fflush(stdin);
gets(auxNome);
for(int x = 0; x < ctime; x++){
    if(strcmp(auxNome, times[x].nome) == 0){
        flag = 1;
        for(int y = 0; y < 20; y++){
            if(times[x].jogadores[y].idade == 0 && times[x].jogadores[y].peso == 0 && times[x].jogadores[y].altura == 0){
                printf("Digite o nome do jogador: ");
                fflush(stdin);
                gets(times[x].jogadores[y].jogador);

                printf("Digite a idade: ");
                scanf("%d", &times[x].jogadores[y].idade);

                printf("Digite o peso em kg: ");
                scanf("%f", &times[x].jogadores[y].peso);

                printf("Digite a altura em metros: ");
                scanf("%f", &times[x].jogadores[y].altura);

                printf("Digite a posição em que o jogador atua: ");
                fflush(stdin);
                gets(times[x].jogadores[y].posicao);

                break;
            }
        }
    }
}
if(!flag)
printf("\nEste time não está cadastrados no banco de dados");
break;
```

Na terceira opção teremos a possibilidade de inserir novos jogadores em um time que já foi cadastrado.

Ao entrar na opção é pedido para que o usuário digite o nome do time que ele inserirá o jogador, e depois que forneça os dados do mesmo.

Uma observação interessante para este caso é a forma que é feito o segundo loop 'for'. Este foi feito da seguinte forma: `for(int y = 0; y < 20; y++)`

O número 20 é a quantidade máxima de jogadores que o vetor 'jogadores' pode ter. Para parar o loop utilizasse o 'If' que vai verificar se os valores da altura, peso e idade são iguais a 0, se forem, é executado o 'break' que saíra do loop.

Essa estrutura é utilizada diversas vezes no código.



# Opção 4

```
case 4:
    printf("\nDigite o time em que o jogador atua: ");
    fflush(stdin);
    gets(auxNome);
    for(int x = 0; x < ctime; x++){
        if(strcmp(auxNome, times[x].nome) == 0){
            flag = 1;
            printf("\nDigite o nome do jogador que deseja modificar os dados: ");
            fflush(stdin);
            gets(auxNome);
            for(int y = 0; y < 20; y++){
                if(times[x].jogadores[y].idade == 0 && times[x].jogadores[y].peso == 0 && times[x].jogadores[y].altura == 0)
                    break;

                else if(strcmp(auxNome, times[x].jogadores[y].jogador) == 0){
                    flag2 = 1;
                    printf("\n****Atenção****\nNessa etapa todos os dados do jogador serão modificados\n");
                    printf("\nDigite nome do jogador: ");
                    fflush(stdin);
                    gets(times[x].jogadores[y].jogador);

                    printf("Digite a idade: ");
                    scanf("%d", &times[x].jogadores[y].idade);

                    printf("Digite o peso em kg: ");
                    scanf("%f", &times[x].jogadores[y].peso);

                    printf("Digite a altura em metros: ");
                    scanf("%f", &times[x].jogadores[y].altura);

                    printf("Digite a posição em que o jogador atua: ");
                    fflush(stdin);
                    gets(times[x].jogadores[y].posicao);

                    break;
                }
            }
        }
    }
}
```

```
if(!flag)
    printf("\nEste time não está cadastrados no banco de dados");
if(!flag2)
    printf("\nEste jogador não foi cadastrado");
break;
```

Na Quarta opção é disponibilizado para o usuário modificar todos os dados do jogador desejado. A sua estrutura é semelhante a opção 3, porém agora se faz a pesquisa e verificação do jogador escolhido.

# Opção 5

```
case 5:
    printf("\nDigite o time em que o jogador atua: ");
    fflush(stdin);
    gets(auxNome);
    for(int x = 0; x < ctime; x++){
        if(strcmp(auxNome, times[x].nome) == 0){
            flag = 1;
            printf("\nDigite o nome do jogador que deseja modificar os dados: ");
            fflush(stdin);
            gets(auxNome);
            for(int y = 0; y < 20; y++){
                if(times[x].jogadores[y].idade == 0 && times[x].jogadores[y].peso == 0 && times[x].jogadores[y].altura == 0)
                    break;

                else if(strcmp(auxNome, times[x].jogadores[y].jogador) == 0){
                    flag2 = 1;
                    printf("\nA - Modificar o nome\nB - Modificar a idade\nC - Modificar o peso\nD - Modificar a altura\nE - Modificar a posição");
                    printf("\nSelecione a opção desejada: ");
                    fflush(stdin);
                    scanf("%c", &op2);
                    switch(op2){
                        case 'a':
                        case 'A':
                            printf("\nDigite o nome atualizado do jogador: ");
                            fflush(stdin);
                            gets(times[x].jogadores[y].jogador);
                            break;

                        case 'b':
                        case 'B':
                            printf("Digite a idade atualizada: ");
                            scanf("%d", &times[x].jogadores[y].idade);
                            break;

                        case 'c':
                        case 'C':
                            printf("Digite o peso em kg atualizado: ");
                            scanf("%f", &times[x].jogadores[y].peso);

                            break;

                        case 'd':
                        case 'D':
                            printf("Digite a altura em metros atualizada: ");
                            scanf("%f", &times[x].jogadores[y].altura);
                            break;

                        case 'e':
                        case 'E':
                            printf("Digite a posição em que o jogador atua atualizada: ");
                            fflush(stdin);
                            gets(times[x].jogadores[y].posicao);
                            break;

                        default:
                            printf("\nOpção inválida");
                    }
                    break;
                }
            }
        }
    }

    if(!flag)
        printf("\nEste time não está cadastrados no banco de dados");
    if(!flag2)
        printf("\nEste jogador não foi cadastrado");
    break;
```

Na Quinta opção é disponibilizado para o usuário modificar um dado específico do jogador. Para isso, é disponibilizado mais um menu que vai de A a E listando todas as informações que podem ser alteradas.

# Opção 6, 0 e default

```
case 6:
    direto = 1;
    break;

case 0:
    printf("\nSaindo do programa...");
    break;

default:
    printf("\nOpção inválida");
```

Na opção 6 o usuário tem a possibilidade de avançar diretamente para o menu de estatísticas.

No case 6 é inserido o valor '1' para a flag 'direto', isso vai possibilitar checkar se entrou na opção e pular a pergunta que é feita ao sair do case.

Se a flag 'direto' for igual a 1 o programa pula o if e entra no else onde a flag 'check' vai receber o valor 1 e se o valor de 'check' é 1 é disponibilizado o menu de estatísticas.

Na opção 0 é exibida uma mensagem "saindo do programa...". Essa opção controla o loop mais externo do programa, ou seja, quando ela é selecionada o programa é encerrado.

E no default a mensagem de "opção inválida".

```
if(!direto){
    printf("\nDeseja avançar para o menu de estatísticas e informações ou voltar ao menu de cadastro? <1 - 2> ");
    scanf("%d", &check);
    fflush(stdin);
    system("cls");
} else{
    check = 1;
}
if(check == 1){
```

# Menu de estatísticas e informações

```
if(check == 1){  
    do{  
        printf("A - Exibir times cadastrados\nB - Exibir jogadores do time\nC - Exibir dados de um jogador\nS - Sair");  
        printf("\nDigite a opção desejada: ");  
        fflush(stdin);  
        scanf("%c", &sair);  
        system("cls");  
    }
```

O menu de estatísticas e informações será controlado pela variável 'sair' que é do tipo 'char'.  
Como no menu anterior, após selecionada a opção desejada é feita uma limpeza no console

# Opção A

```
switch(sair){
case 'a':
case 'A':
printf("\n-----Lista de times cadastrados-----");
for(int x = 0; x < ctime; x++){
    printf("\n%s", times[x].nome);
    for(int y = 0; y < 20; y++){
        if(times[x].jogadores[y].idade == 0 && times[x].jogadores[y].peso == 0 && times[x].jogadores[y].altura == 0)
            break;
        else
            cont++;
    }
    printf("\nQuantidade de jogadores: %d", cont);
    cont = 0;
}
printf("\n");
break;
```

Na opção A é exibida uma lista de times cadastrados e a quantidade de jogadores que tem em cada time.

# Opção B

```
case 'b':
case 'B':
    printf("\nDigite o nome do Time: ");
    fflush(stdin);
    gets(auxNome);
    flag = 0;
    flag2 = 0;
    for(int x = 0; x < ctime; x++){
        if(strcmp(auxNome, times[x].nome) == 0){
            flag = 1;
            printf("\nJogadores do Time %s\n", times[x].nome);
            for(int y = 0; y < 20; y++){
                if(times[x].jogadores[y].idade == 0 && times[x].jogadores[y].peso == 0 && times[x].jogadores[y].altura == 0)
                    break;
                else {
                    flag2 = 1;
                    printf("\nJogador: %s", times[x].jogadores[y].jogador);
                    printf("\tPosição: %s", times[x].jogadores[y].posicao);
                    printf("\n");
                }
            }
        }
        if(!flag)
            printf("\nEste time não está cadastrados no banco de dados");
        if(!flag2)
            printf("\nNão há jogadores cadastrados nesse time");
    }
    break;
```

Na opção B são exibidos todos os jogadores e de um time e suas respectivas posições.

# Opção C

Na opção C temos a avaliação dos jogadores. Nela faz-se é recolhido o valor do IMC calculado na opção 1 do primeiro menu. A partir dele iremos verificar se o jogador está abaixo do peso, regular ou acima do peso.

**Cálculo de calorias ->** O cálculo de calorias utilizado aqui teve como base o cálculo disponibilizado pelo site '[tuasaude](#)', onde é disposta uma tabela que tem algumas variações como mulher, homem e idades, porém não foram utilizadas todas essas variações, pois a ideia do programa inicial não é ter a maior precisão possível e também foram feitos os cálculos com as variações e os resultados não eram muito discrepantes. Portanto, será utilizado aqui o cálculo para homens de 18 a 30 anos que é o seguinte:  $(15,057 \times \text{peso em kg}) + 692,2$ .

Esse cálculo vai nos disponibilizar o tanto de calorias que o corpo gasta diariamente, ou seja, para manter o peso atual o jogador só precisará manter o consumo dessas calorias, para perder peso será feita uma dieta com déficit calórico que aqui será de -500kcal que prometem a perda de 2kg por mês e para o ganho de peso a dieta será com superavit calórico de +500kcal que promete o ganho de 2kg mensais.

**Cálculo da Proteína ->** A proteína na dieta deve ser responsável pelo consumo de 10% a 35% das calorias diárias, aqui consideraremos como 30%, pois a ideia é cadastrar atletas esportivos.

**Cálculo das gorduras ou lipídios ->** A gordura terá 20% das calorias.

**Cálculo dos carboidratos ->** Os carboidratos terão 50% das calorias.



# Opção C

```
case 'c':
case 'C':
printf("\nDigite o nome do jogador: ");
fflush(stdin);
gets(auxNome);
for(int x = 0; x < ctme; x++){
    for(int y = 0; y < 20; y++){
        if(times[x].jogadores[y].idade == 0 && times[x].jogadores[y].peso == 0 && times[x].jogadores[y].altura == 0)
            break;

        else if(strcmp(auxNome, times[x].jogadores[y].jogador) == 0){
            flag = 1;
            printf("Jogador: %s", times[x].jogadores[y].jogador);
            printf("\nIdade: %d", times[x].jogadores[y].idade);
            printf("\nPeso: %.2f", times[x].jogadores[y].peso);
            printf("\nAltura: %.2f", times[x].jogadores[y].altura);
            printf("\nPosição: %s", times[x].jogadores[y].posicao);
            printf("\nIMC: %.2f", times[x].jogadores[y].imc);
            calculo = 15.057 * times[x].jogadores[y].peso + 692.2;
            // Tabela IMC
            if(times[x].jogadores[y].imc < 19){
                caloria = calculo + 500;
                printf("\nO jogador está abaixo do peso.\nDeve consumir %.2f kcal diárias para engordar 2kg por mês", caloria);
                printf("\n-----Tabela de macronutrientes a serem consumidos-----\n");
                printf("\n          Calorias      :      %.2fkcal", caloria);
                printf("\n          Proteína       :      %.2fkcal", caloria*0.30);
                printf("\n          Lipídios        :      %.2fkcal", caloria*0.20);
                printf("\n          Carboidratos    :      %.2fkcal", caloria*0.5);
            } else if(times[x].jogadores[y].imc < 25){
                printf("\nO jogador está com o peso regular/normal.\nDeve consumir %.2f kcal diárias para manter o peso atual", calculo);
                printf("\n-----Tabela de macronutrientes a serem consumidos-----\n");
                printf("\n          Calorias      :      %.2fkcal", calculo);
                printf("\n          Proteína       :      %.2fkcal", calculo*0.30);
                printf("\n          Lipídios        :      %.2fkcal", calculo*0.20);
                printf("\n          Carboidratos    :      %.2fkcal", calculo*0.5);
            }
        }
    }
}
```

```
} else if(times[x].jogadores[y].imc < 30){
    caloria = calculo - 500;
    printf("\nO jogador está com excesso de peso.\nDeve consumir %.2f kcal diárias para emagrecer 2kg por mês", caloria);
    printf("\n-----Tabela de macronutrientes a serem consumidos-----\n");
    printf("\n          Calorias      :      %.2fkcal", caloria);
    printf("\n          Proteína       :      %.2fkcal", caloria*0.30);
    printf("\n          Lipídios        :      %.2fkcal", caloria*0.20);
    printf("\n          Carboidratos    :      %.2fkcal", caloria*0.5);
} else if(times[x].jogadores[y].imc < 35){
    caloria = calculo - 500;
    printf("\nO jogador está com Obesidade classe I.\nDeve consumir %.2f kcal diárias para emagrecer 2kg por mês", caloria);
    printf("\n-----Tabela de macronutrientes a serem consumidos-----\n");
    printf("\n          Calorias      :      %.2fkcal", caloria);
    printf("\n          Proteína       :      %.2fkcal", caloria*0.30);
    printf("\n          Lipídios        :      %.2fkcal", caloria*0.20);
    printf("\n          Carboidratos    :      %.2fkcal", caloria*0.5);
} else if(times[x].jogadores[y].imc < 40){
    caloria = calculo - 500;
    printf("\nO jogador está com Obesidade classe II.\nDeve consumir %.2f kcal diárias para emagrecer 2kg por mês", caloria);
    printf("\n-----Tabela de macronutrientes a serem consumidos-----\n");
    printf("\n          Calorias      :      %.2fkcal", caloria);
    printf("\n          Proteína       :      %.2fkcal", caloria*0.30);
    printf("\n          Lipídios        :      %.2fkcal", caloria*0.20);
    printf("\n          Carboidratos    :      %.2fkcal", caloria*0.5);
} else {
    caloria = calculo - 500;
    printf("\nO jogador está com Obesidade classe III.\nDeve consumir %.2f kcal diárias para emagrecer 2kg por mês", caloria);
    printf("\n-----Tabela de macronutrientes a serem consumidos-----\n");
    printf("\n          Calorias      :      %.2fkcal", caloria);
    printf("\n          Proteína       :      %.2fkcal", caloria*0.30);
    printf("\n          Lipídios        :      %.2fkcal", caloria*0.20);
    printf("\n          Carboidratos    :      %.2fkcal", caloria*0.5);
}
printf("\n");
}
}
```

```
if(!flag)
    printf("\nEsse jogador não está cadastrado no banco de dados");
break;
```

# Opção S e default

```
case 's':
case 'S':
    printf("\nVoltando ao menu inicial...");
    system("cls");
    break;
default:
    printf("\nopção inválida");
}
printf("\n");
} while(sair != 's' && sair != 'S');
}
```

Na opção S teremos a mensagem "voltando ao menu inicial" e nela sairemos do loop do menu de estatísticas e voltaremos para o loop do menu de cadastro.  
O default exibe uma mensagem para opções não válidas.

# Fontes para os cálculos

---

Calorias: <https://www.feitodeiridium.com.br/como-calcular-sua-necessidade-de-macro-e-micronutrientes/>

---

Macronutrientes: <https://www.feitodeiridium.com.br/como-calcular-sua-necessidade-de-macro-e-micronutrientes/>

# Kéwen Silva

## Desenvolvedor

● <https://github.com/kewensilva-gif/cadastro-time>

