

MINICURSO DE ALGORITMOS 2

PET-SIMC

STRUCT

PET de Sistemas de Informação do Campus Monte Carmelo

@petsimc





CRONOGRAMA:

1. Definição
2. Sintaxe
 - a. Criando
 - b. Instanciando
 - c. Acessando
 - d. String
3. Exercícios



O que são structs



O pallet aparece novamente... Mas agora para agrupar variáveis de diferentes tipos.

@petsimc

O que são structs



Então basicamente essa é uma estrutura para agrupar variáveis relacionadas. Por exemplo, uma pessoa poderia ser representada como uma struct.

Sintaxe:

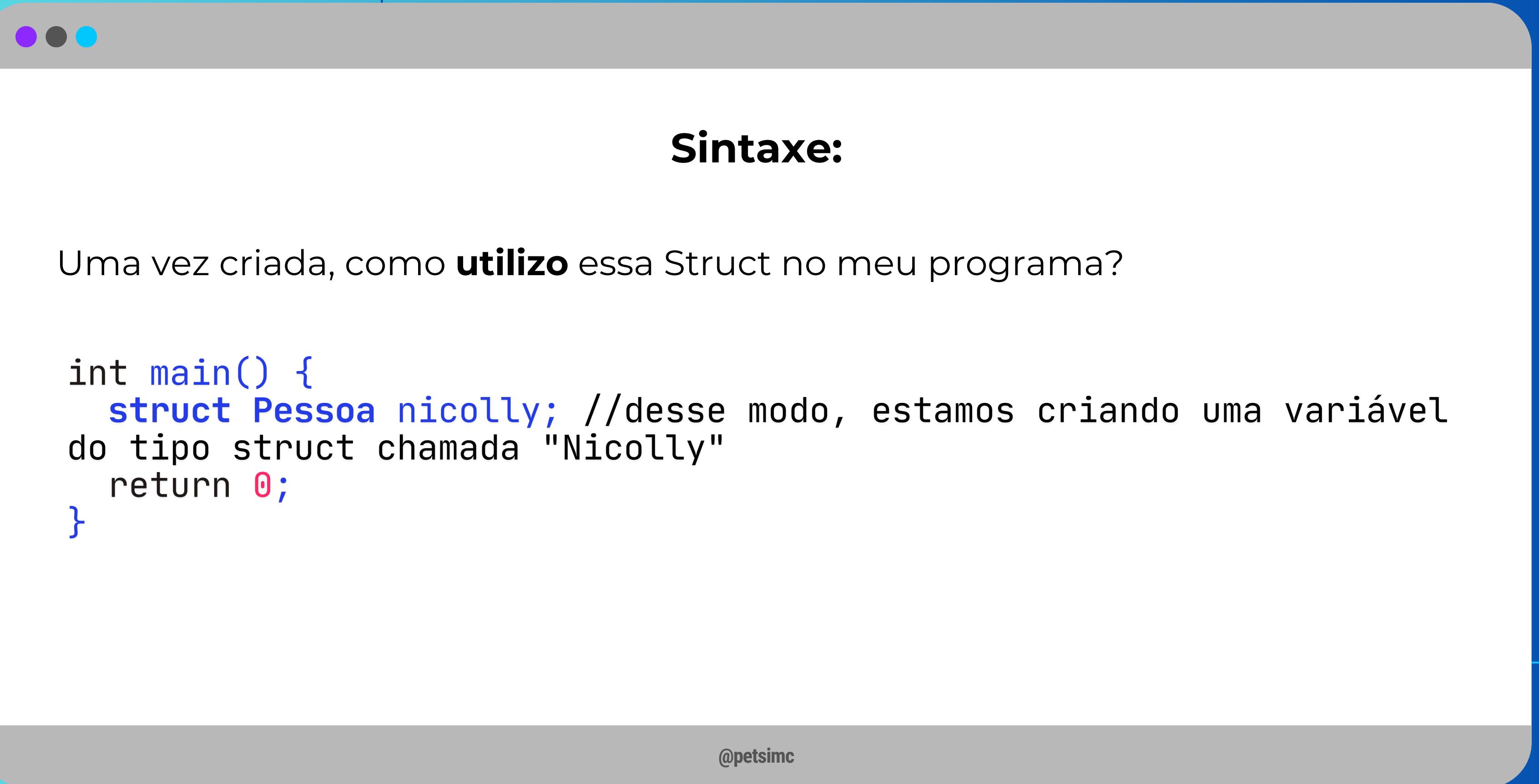
Para **criar uma Struct**. Usamos a seguinte declaração:

```
struct Pessoa { //declaração e nome  
    int idade;  
    char inicialNome;  
};
```

Sintaxe:

Apelidando a struct:

```
struct Pessoa { //declaração e nome  
    int idade;  
    char inicialNome;  
}p; //definindo um apelido
```



Sintaxe:

Uma vez criada, como **utilizo** essa Struct no meu programa?

```
int main() {  
    struct Pessoa nicolly; //desse modo, estamos criando uma variável  
    do tipo struct chamada "Nicolly"  
    return 0;  
}
```

Sintaxe:

Uma vez criada, como **utilizo** essa Struct no meu programa?

```
int main() {  
    struct Pessoa nicolly;  
    nicolly.idade=20;  
    nicolly.inicialNome='n';  
    struct Pessoa lavinia={21,'l'};  
    return 0;  
}
```

@petsimc

Sintaxe:

E posso ter um **vetor** que contém várias "Pessoas"?

```
int main() {
    struct Pessoa pessoas[5]; //esse vetor contém 5 "objetos" pessoas
    for(int i=0; i<5; i++){
        pessoas[i].idade=20+i;
        pessoas[i].inicialNome='a';
    }
    return 0;
}
```

Sintaxe:

Quando você declara uma struct com um apelido, você está criando uma variável global daquele tipo struct.

```
int main() {  
  
    struct Pessoa {  
        int idade;  
        char inicialNome;  
    }p; //definindo um apelido  
  
    // Acesso DIRETO sem precisar instanciar  
    p.idade = 25;  
    p.inicialNome = 'J';  
  
    printf("Idade: %d\n", p.idade);  
    printf("Inicial: %c\n", p.inicialNome);  
  
    return 0;  
}
```

E quanto às strings em estruturas?

Lembre-se de que strings em C são, na verdade, uma matriz de caracteres e, infelizmente, você não pode atribuir um valor a uma matriz como esta:

```
int main() {  
    struct Pessoa {  
        int idade;  
        char inicialNome;  
        char nome[32];  
    };  
  
    struct Pessoa p1;  
    p1.idade = 25;  
    p1.inicialNome = 'J';  
    // Dessa forma dará erro  
    p1.nome = "nicolly"  
    // Executem e vejam a saída  
    printf("Nome: %s\n", p1.nome);  
  
    return 0;  
}
```

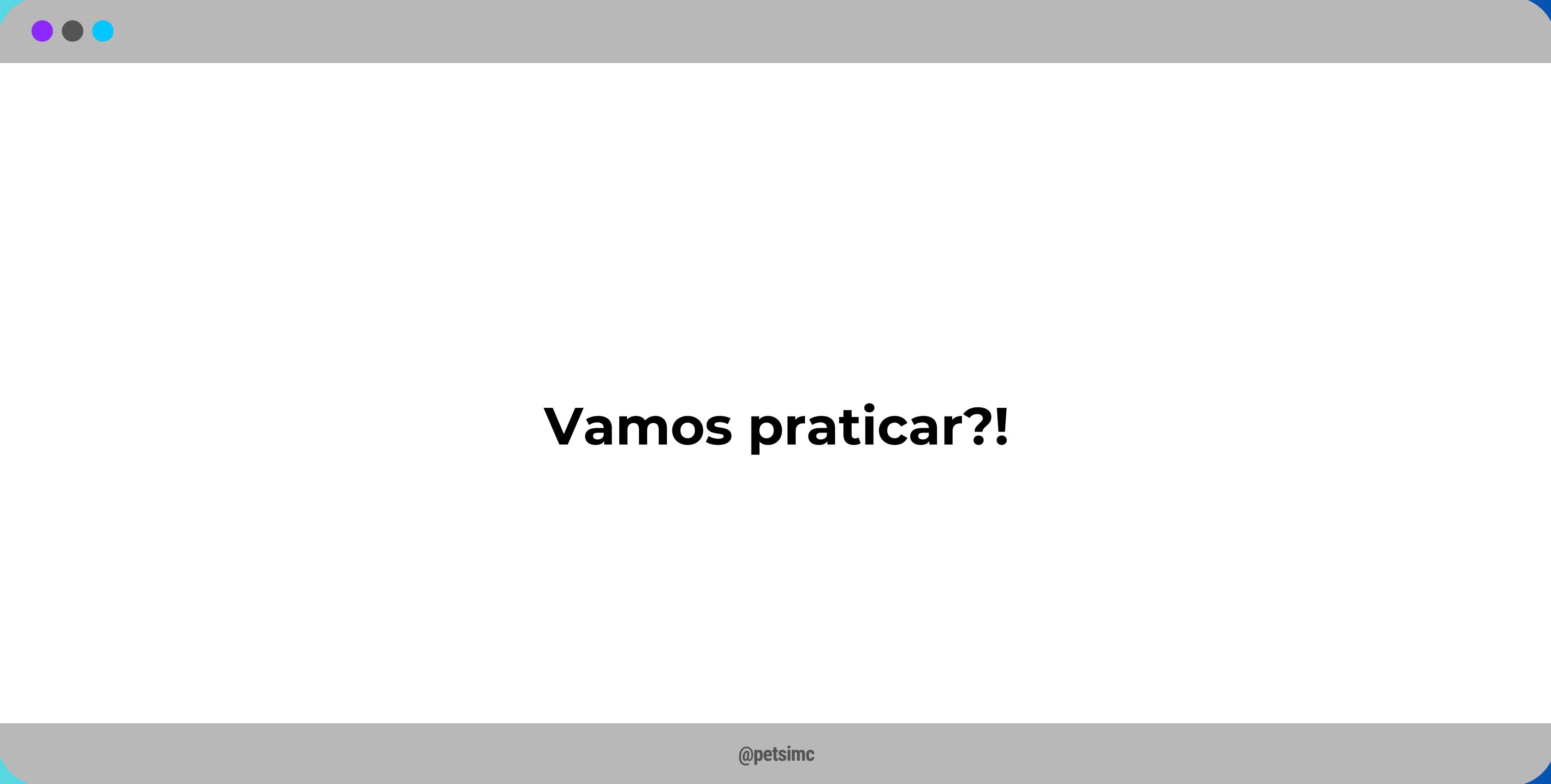
@petsimc

E quanto às strings em estruturas?

No entanto, existe uma solução para isso! Você pode usar a função strcpy()

```
int main() {  
  
    struct Pessoa p1;  
  
    p1.idade = 25;  
    p1.inicialNome = 'J';  
  
    // Usando a função strcpy()  
    strcpy(p1.nome, "Nicolly");  
  
    // Executem e vejam a saída  
    printf("Nome: %s\n", p1.nome);  
  
    return 0;  
}
```

A função strcpy() em C é usada para copiar strings de um local para outro. Ela recebe dois parâmetros: o destino (onde a string será copiada) e a origem (a string que será copiada).



Vamos praticar?!

@petsimc



Exercício 01 - Pessoas

Implemente um programa que leia o nome, a idade e o endereço de uma pessoa e armazene os dados em uma estrutura.

Exercício 02 - Cadastro de livros

Neste exercício, você deverá criar um programa em C para gerenciar um cadastro de livros utilizando vetores de structs. Primeiramente, defina uma estrutura chamada Livro contendo três campos: titulo (uma string), autor, valor e ano de publicação.

1. Declare um vetor com 3 livros.
2. Preencha os dados via input.
3. Imprima todos os livros publicados após o ano 2000.



Exercício 03 - Alunos

Crie uma estrutura representando os alunos de um determinado curso. A estrutura deve conter a matrícula do aluno, nome e média geral.

- (a) Permita ao usuário entrar com os dados de 5 alunos.
- (b) Encontre o aluno com maior média geral.
- (c) Encontre o aluno com menor média geral.
- (d) Para cada aluno diga se ele foi aprovado ou reprovado, considerando o valor 6 para aprovação.



CONTATO

Como entrar em contato em caso de dúvidas?



E-mail

laviniabd@ufu.br
nicolly.luz@ufu.br



Sala do PET- 1BMC206

UFU - Monte
Carmelo - Campus
Araras



Site do PET-SIMC

petsimc.facom.ufu.br



Grupo do Wpp

