



STRUCT EM C++

Algoritmos e Estrutura de Dados I
Vânia de Oliveira Boreges



03 INTRODUÇÃO

04 COMPREENDER O CONCEITO DE `STRUCT` EM C++.

05 APRENDER A DECLARAR E UTILIZAR `STRUCTS`

06 ENTENDER A DIFERENÇA ENTRE `STRUCTS` E `CLASSES`

07 APLICAR `STRUCTS` EM EXEMPLOS PRÁTICOS

O QUE É UMA STRUCT?

Definição: Uma `struct` (abreviação de 'structure') é um tipo de dado composto que permite agrupar variáveis de tipos diferentes sob um único nome.

Analogias: Comparar uma `struct` a uma ficha de cadastro onde diferentes informações (nome, idade, endereço) estão organizadas em um único formulário.



DECLARAÇÃO DE UMA STRUCT

Sintaxe Básica:

```
struct NomeDaStruct {  
    Tipo variavel1;  
    Tipo variavel2;  
    // mais variáveis  
};
```

Exemplo Simples:

```
struct Pessoa {  
    std::string nome;  
    int idade;  
    float altura;  
};
```



ACESSANDO MEMBROS DE UMA STRUCT

Como instanciar uma struct.

Acessar e modificar os membros de uma `struct` usando o operador `.` (ponto).

Exemplo:

```
Pessoa p1;
```

```
p1.nome = "João";
```

```
p1.idade = 30;
```

```
p1.altura = 1.75;
```

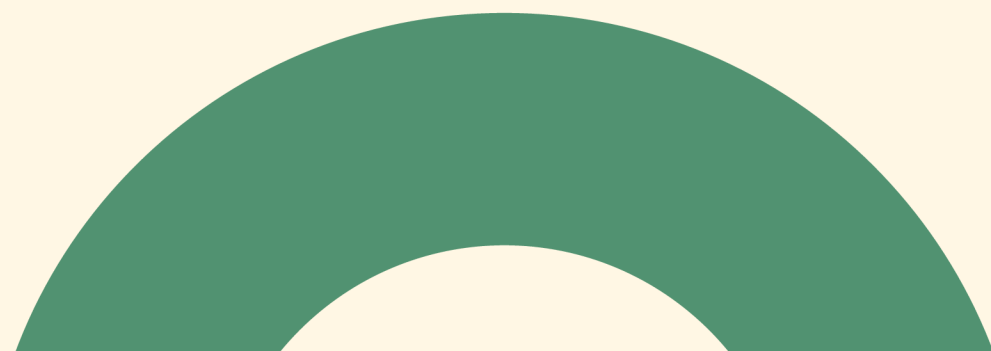
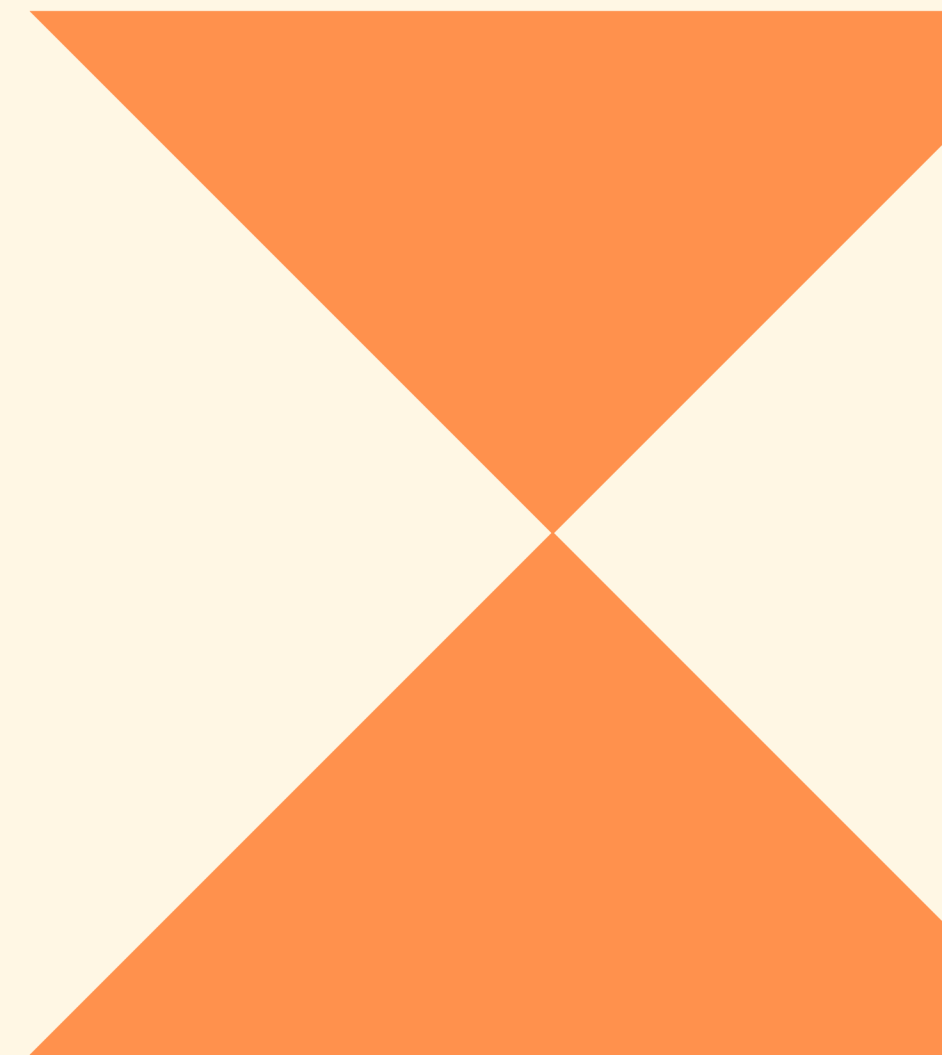
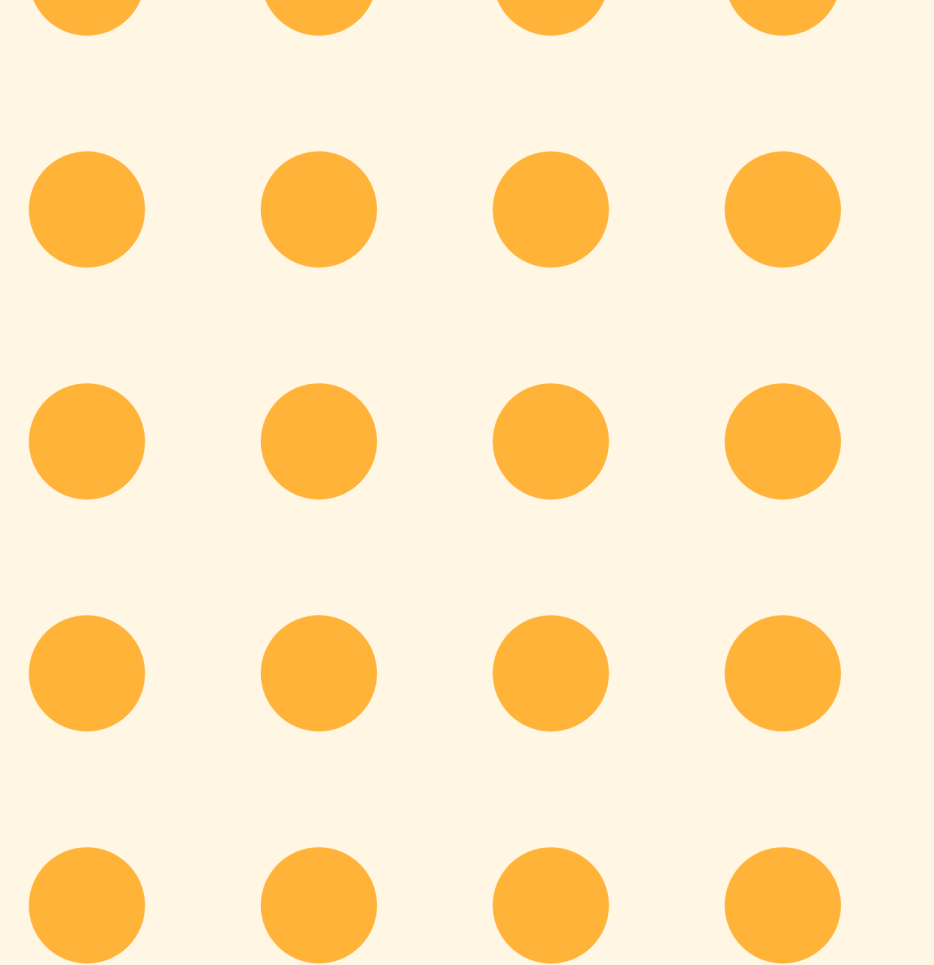
INICIALIZAÇÃO DE STRUCTS

Inicialização direta:

```
Pessoa p1 = {"João", 30, 1.75};
```

Lista de inicialização:

```
Pessoa p2{"Maria", 25, 1.65};
```



DIFERENÇA ENTRE STRUCTS E CLASSES

Por padrão, os membros de uma struct são públicos (public), enquanto os membros de uma `class` são privados (private).

Uso de `struct` geralmente para dados simples, enquanto `class` é mais usada em conceitos de Programação Orientada a Objetos (POO).

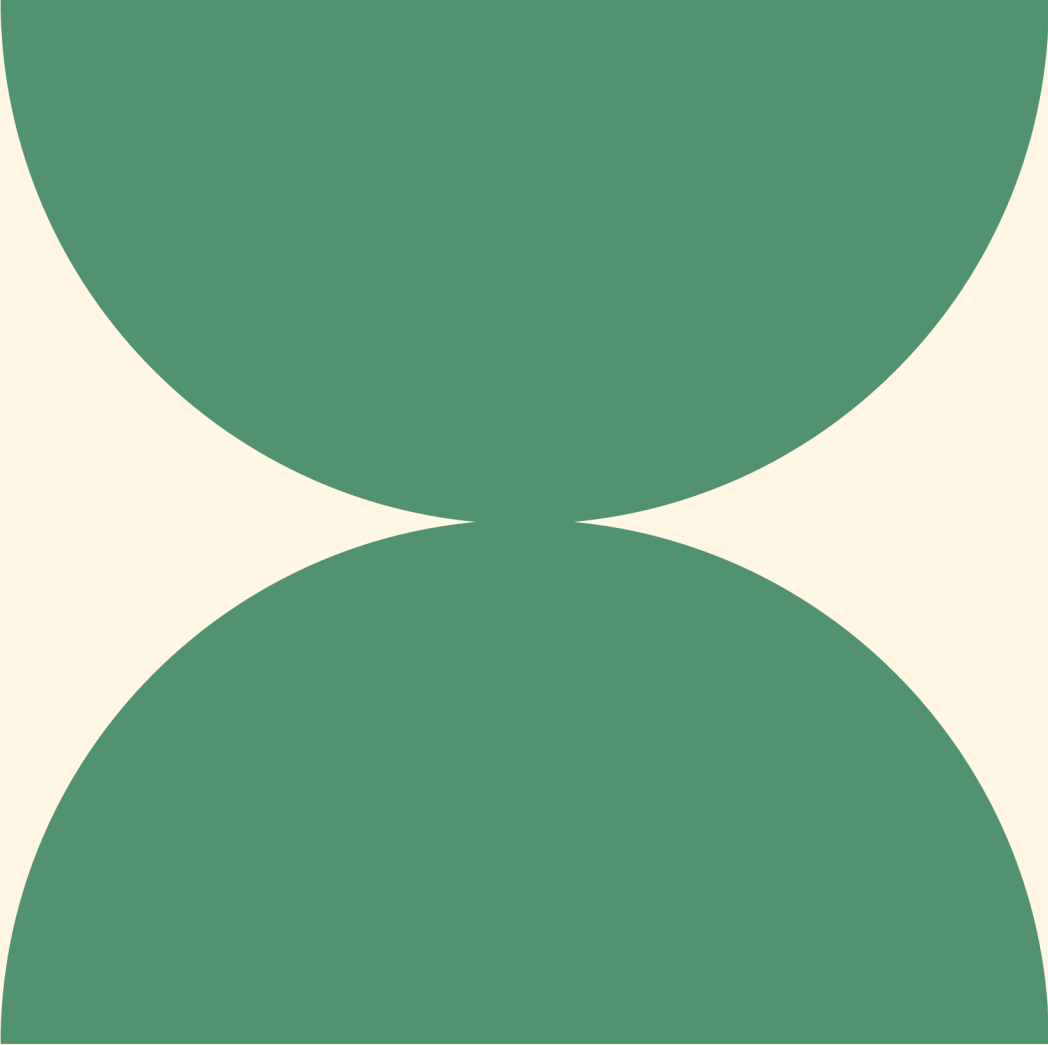


STRUCTS ANINHADAS

Declaração de `structs` dentro de outras `structs` para modelar dados mais complexos.

Exemplo:

```
struct Endereco {  
    std::string rua;  
    int numero;  
};  
  
struct Pessoa {  
    std::string nome;  
    int idade;  
    Endereco endereco;  
};
```

USOS COMUNS DE STRUCTS

Modelagem de dados simples, como coordenadas, datas, registros de banco de dados.

Código limpo e organizado.



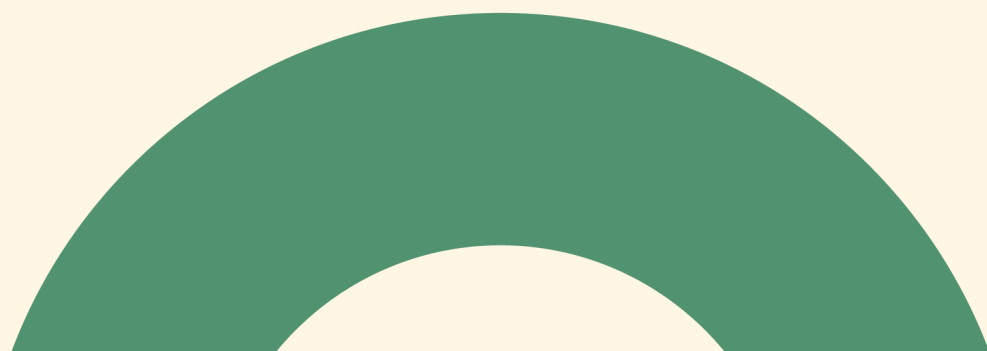
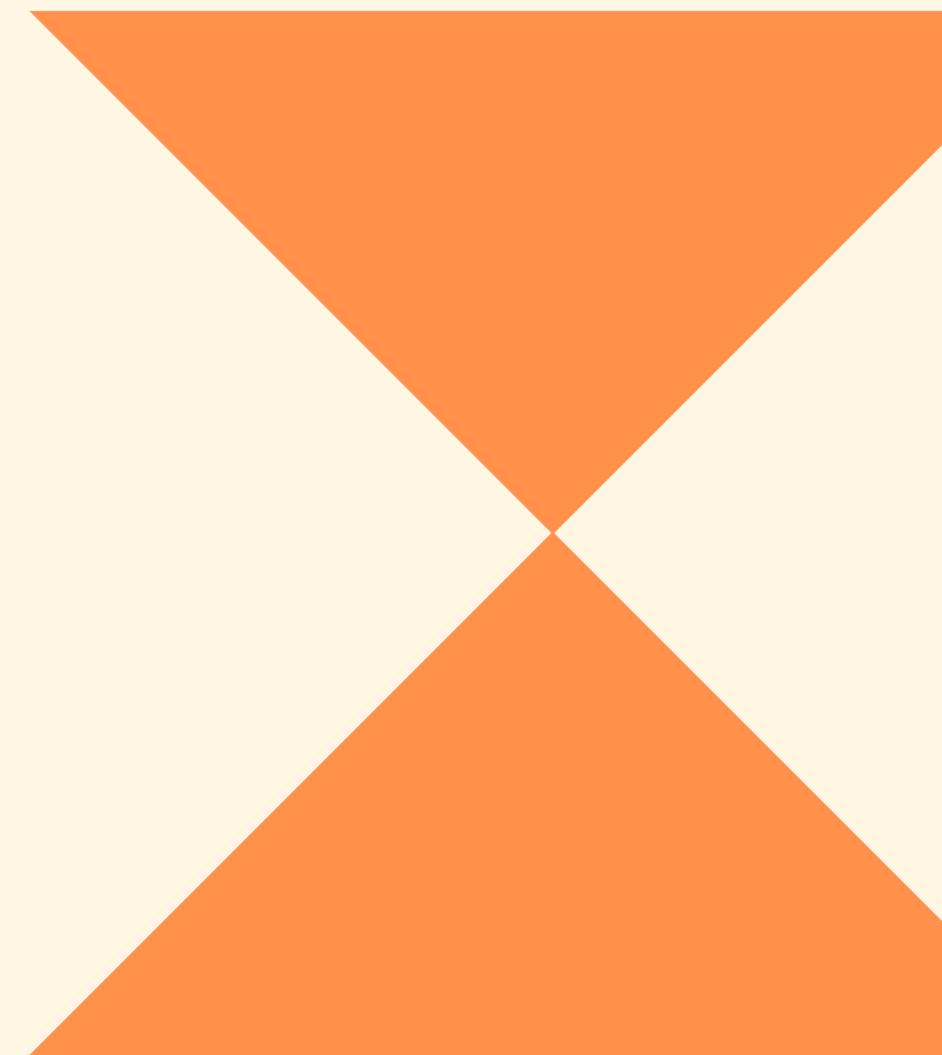
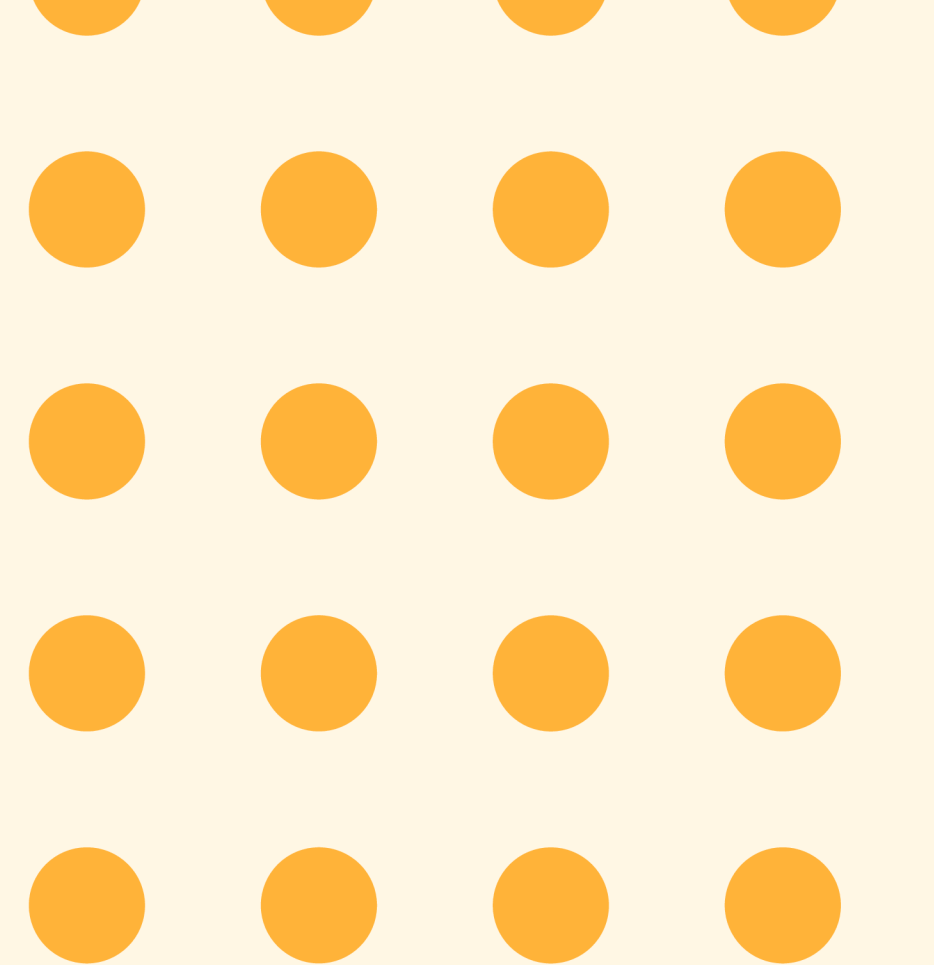
INICIALIZAÇÃO DE STRUCTS

Inicialização direta:

```
Pessoa p1 = {"João", 30, 1.75};
```

Lista de inicialização:

```
Pessoa p2{"Maria", 25, 1.65};
```



RESULTADOS

Exercício 1: Crie uma struct chamada Produto que contenha três membros: nome (string), preco (float), e quantidade (int). No programa principal (main), instancie um objeto dessa struct, atribua valores a cada membro e exiba as informações do produto no console.

Exercício 2: Crie uma struct chamada Estudante que contenha os seguintes membros: nome (string), idade (int) e notaFinal (float). No programa principal (main), crie um objeto dessa struct, atribua valores aos membros e imprima os detalhes do estudante no console.

RESULTADOS

Exercício 3: Crie uma ``struct`` para representar um ponto no espaço 2D (com ``x`` e ``y`` como membros). Em seguida, crie uma função que calcule a distância entre dois pontos.

Exercício 4: Crie um programa que utilize struct para armazenar informações sobre um livro (título, autor, ano de publicação), leia os dados e após armazenamento os imprima na tela.

Exercício 5: Faça um programa que utilize structs aninhadas para armazenar informações sobre um aluno (nome, idade, endereço, notas) e calcule a média das notas.