

Algoritmo e Programação

Vetores, Matrizes, Structs e

Prof. Dr. Gilberto Fernandes Jr.





Structs

- Definição e criação de structs
- Manipulação de dados em structs

Por que utilizar estruturas?

- Imagine ter de declarar quatro cadastros para quatro pessoas diferentes:

```
char nome1[50], nome2[50], nome3[50],  
nome4[50];  
int idade1, idade2, idade3, idade4;  
char rua1[50], rua2[50], rua3[50], rua4[50];  
int numero1, numero2, numero3, numero4;
```

- Utilizando uma estrutura, o mesmo pode ser feito da seguinte maneira:

```
struct cadastro c1, c2, c3, c4;
```



Criação de uma struct

```
struct Cadastro {  
    char nome[30];  
    int idade;  
    char rua[50];  
    int numero;  
};
```

Declaração no main:

```
struct Cadastro c1;
```



```
typedef struct {  
    char nome[30];  
    int idade;  
    char rua[50];  
    int numero;  
} Cadastro;
```

Declaração no main:

```
Cadastro c1;
```

Acesso aos dados

- Cada campo (variável) da estrutura pode ser acessado usando o operador “.” (ponto)

```
struct Cadastro c1;
```

```
strcpy(c1.nome, "Joao");  
c1.idade = 32;
```

← Atribuição direta

```
fgets(c1.rua, 30, stdin);  
scanf("%d", &c1.numero);
```

← Inserção pelo teclado



Complementando...

Atribuição

```
struct Ponto{  
    int x;  
    int y;  
};
```

```
struct Ponto x;  
struct Ponto y = {1,3};
```

```
x = y;
```



Aninhamento

```
struct Endereco{  
    char rua[50];  
    int numero;  
};
```

```
struct cadastro{  
    char nome[50];  
    int idade;  
    struct Endereco end;  
};
```

Vetores

```
struct Cadastro c[4];  
  
c[0].idade = 18;
```



Vamos Praticar?

Vamos utilizar o Visual Studio Code para construir alguns algoritmos com estruturas na linguagem C