

# Linguagem de Programação C



**PROFESSOR: DIEGO RICARDO KROHL**  
**`diego.krohl@ifc-videira.edu.br`**

# Linguagem C – Registros (Structs)



- São variáveis compostas heterogêneas;
- São conjuntos de dados logicamente relacionados, mas de tipos diferentes (inteiro, real, string, etc.);
- Os elementos dos registros são chamados de campos.
- Exemplo: Dados sobre funcionários de uma empresa:
  - Nome
  - Idade
  - Salário

# Registros - Declaração



- ```
struct nome_do_tipo_do_registro {  
    tipo1 campo1;  
    tipo2 campo2;  
    tipo3 campo3;  
    // ...  
    tipoN campoN;  
};
```
- ```
struct funcionario {  
    char nome[50];  
    int idade;  
    float salario;  
};
```

# Acesso a campos de um registro



- Pode ser realizado através da seguinte sintaxe:
  - `nome_do_registro.nome_do_campo`
- Para uma variável `f` do tipo `funcionario`:
  - `struct funcionario f;`
- O campo `nome` é acessado assim:
  - `f.nome`

# Exemplo:



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

struct funcionario {
    char nome[50];
    int idade;
    float salario;
};

main() {
    struct funcionario f;

    strcpy(f.nome, "Regis");
    f.idade = 18;
    f.salario = 1000;

    printf("Nome: %s\n", f.nome);
    printf("Idade: %d\n", f.idade);
    printf("Salario: %.2f\n", f.salario);
}
```

# Vetor de Registros



- Declaração:

- `struct nome_registro nome_da_variavel[tamanho_do_vetor];`

- Uso:

- `nome_da_variavel[indice].nome_do_campo;`

# Exemplo



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

struct pessoa {
    char nome[50];
    int idade;
};

main() {
    struct pessoa p[2];

    strcpy(p[0].nome, "Regis");
    p[0].idade = 18;
    strcpy(p[1].nome, "Maria");
    p[1].idade = 25;

    printf("Nome: %s - Idade: %d\n", p[0].nome, p[0].idade);
    printf("Nome: %s - Idade: %d\n", p[1].nome, p[1].idade);

}
```

# Funções e registros



- Funções podem aceitar registros como parâmetros e devolver registros como resultado;
- Regra simples: `struct nome` é, para o C, um tipo tal qual `int` e `float`, e pode ser usado como os primitivos nas funções.



# Exemplo



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

struct pessoa {
    char nome[50];
    int idade;
};

void imprime_pessoa(struct pessoa p) {
    printf("nome: %s\n", p.nome);
    printf("idade: %i\n", p.idade);
}

main() {
    struct pessoa aluno1;
    ...
    imprime_pessoa(aluno1);
}
```

# Funções e registros



```
#include <stdio.h>
#include <string.h>
#define MAX 5

    struct produto {
        char descricao[40];
        float valor;
    };

main() {

    struct produto prods[MAX];
    int i;

    for (i=0; i<MAX; i++){
        puts("Nome do Produto?");
        gets(prods[i].descricao);
        puts("Valor do Produto?");
        scanf("%f", &prods[i].valor);
    }

    for (i=MAX-1; i>=0; i--) {
        printf("Descricao: %s - Valor: R$ %.2f\n",
            prods[i].descricao, prods[i].valor);
    }
}
```