



UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI  
Campus Senador Helvídio Nunes de Barros - CSHNB  
Curso Bacharelado em Sistemas de Informação

## **ALGORITMOS E PROGRAMAÇÃO II**

Glauber Dias Gonçalves  
ggoncalves@ufpi.edu.br

# Revisão Tipos Abstratos de Dados (TAD)

- Conceito
- Codificação em C (struct)
- Vetor de TAD

# CONCEITO

- Tipo Abstrato de Dados (**TAD**)
  - coleção de variáveis/tipos
  - possivelmente de tipos diferentes
  - logicamente relacionadas para formar um TAD



# CONCEITO

- Tipo Abstrato de Dados (**TAD**)
  - coleção de variáveis/tipos
  - possivelmente de tipos diferentes
  - logicamente relacionadas para formar um TAD
- Os itens de um TAD são chamados de campos.
- Exemplos:
  - Funcionário de uma empresa
    - nome, cargo, salário
  - Aluno universitário
    - matrícula, nome, curso
  - Endereço
    - CEP, logradouro, numero, bairro, cidade

# Conceito

- Declaração da estrutura de dados básica

estrutura nome\_do\_tipo:

<tipo>: <variável>

<tipo>: <variável>

...

fim\_estrutura

Nome do tipo  
do registro.

Tipos primitivos (int, float, ...)  
do campo do registro.

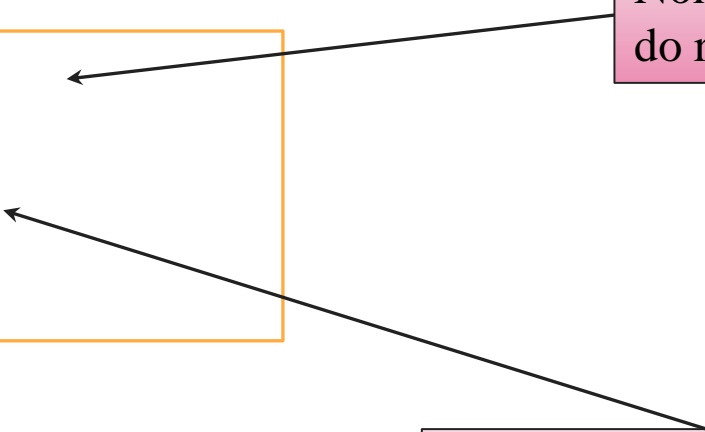
# Codificação em C

- Declaração de um struct (TAD).

## Exemplo:

```
struct funcionario{  
    char nome[80];  
    char cargo[10];  
    float salario;  
};
```

Nome do tipo  
do registro.



Tipos primitivos (int, float, ...)  
do campo do registro.

# ESTRUTURA DE DADOS

- Instanciar variáveis de um TAD
  - Após declaração da estrutura do TAD

```
struct funcionario{  
    char nome[80];  
    char cargo[10];  
    float salario;  
};  
  
int main(){  
    struct funcionario func1;  
    ...  
}
```

# ESTRUTURA DE DADOS

- O acesso ao campo de um TAD com a sintaxe:

**nome\_da\_variável.nome\_do\_campo**

```
struct funcionario{
    char nome[80];
    char cargo[10];
    float salario;
};

int main(){
    struct funcionario func1;
    scanf("%[^\\n]s", func1.nome);
    printf("Nome: %s", func1.nome);
    ...
}
```



# PRÁTICA

- Fazer o TAD carro com os campos: nome, montadora e autonomia dada em km/litro.
- Fazer um algoritmo para ler dois carros e imprimir o carro mais econômico.

# VETOR DE TAD

---

# VETOR DE TAD

- Declarado da mesma forma que vetores de tipos primitivos
  - Define-se o TAD
  - Declara-se o vetor do TAD criado
- Exemplo:
  - estrutura funcionario: func\_empresaA[28]
  - estrutura carro: carro\_lojaB[100]

# VETOR DE TAD

- Para manipular os dados do vetor, devem ser fornecidos o índice e o campo
- Exemplo:
  - `func_empresaA[0].salario = 1100.00`
  - `func_empresaA[1].salario = 2500.00`
  - `carro_lojaB[0].autonomia = 10;`
  - `carro_lojaB[1].autonomia = 15;`

# VETOR DE TAD

- Exemplo:

```
#include <stdio.h>

struct funcionario{
    char nome[80];
    char cargo[10];
    float salario;
};
```

# VETOR DE TAD

- Exemplo:

```
#include <stdio.h>

struct funcionario{
    char nome[80];
    char cargo[10];
    float salario;
};
```

```
int main(){
    struct funcionario empresaA[10];
    int i, num_func;

    num_func = 2;
    for( i=0; i<num_func; i++ ){
        setbuf(stdin, NULL);
        printf("Nome:");
        scanf("%[^\n]s", empresaA[i].nome);
        printf("Cargo:");
        setbuf(stdin, NULL);
        scanf("%[^\n]s", empresaA[i].cargo);
        printf("Salario:");
        setbuf(stdin, NULL);
        scanf("%f", &empresaA[i].salario);
    }

    printf("Funcionarios:\n");
    for( i=0; i<num_func; i++ ){
        printf("Nome: %s\n", empresaA[i].nome);
        printf("Cargo: %s\n", empresaA[i].cargo);
        printf("Salario: %.2f\n", empresaA[i].salario);
    }
    return 0;
}
```

# ALTERNATIVAS PARA DECLARAÇÃO DE TADS

- Exemplo para 1a. alternativa:

```
#include <stdio.h>

struct funcionario{
    char nome[80];
    char cargo[10];
    float salario;
};

typedef struct funcionario Funcionario;

int main(){
    Funcionario empresaA[10];
    int i, num_func;

    num_func = 2;
    for( i=0; i<num_func; i++ ){
        setbuf(stdin, NULL);
        printf("Nome:");
        scanf("%[^\n]s", empresaA[i].nome);
        printf("Cargo:");
        setbuf(stdin, NULL);
        scanf("%[^\n]s", empresaA[i].cargo);
        printf("Salario:");
        setbuf(stdin, NULL);
        scanf("%f", &empresaA[i].salario);
    }

    printf("Funcionarios:\n");
    for( i=0; i<num_func; i++ ){
        printf("Nome: %s\n", empresaA[i].nome);
        printf("Cargo: %s\n", empresaA[i].cargo);
        printf("Salario: %.2f\n", empresaA[i].salario);
    }
    return 0;
}
```

# ALTERNATIVAS PARA DECLARAÇÃO DE TADS

- Exemplo para 2a. alternativa:

```
#include <stdio.h>

typedef struct{
    char nome[80];
    char cargo[10];
    float salario;
} Funcionario;
```

```
int main(){
    Funcionario empresaA[10];
    int i, num_func;

    num_func = 2;
    for( i=0; i<num_func; i++ ){
        setbuf(stdin, NULL);
        printf("Nome:");
        scanf("%[^\n]s", empresaA[i].nome);
        printf("Cargo:");
        setbuf(stdin, NULL);
        scanf("%[^\n]s", empresaA[i].cargo);
        printf("Salario:");
        setbuf(stdin, NULL);
        scanf("%f", &empresaA[i].salario);
    }

    printf("Funcionarios:\n");
    for( i=0; i<num_func; i++ ){
        printf("Nome: %s\n", empresaA[i].nome);
        printf("Cargo: %s\n", empresaA[i].cargo);
        printf("Salario: %.2f\n", empresaA[i].salario);
    }
    return 0;
}
```



# PRÁTICA

- Fazer o TAD carro com os campos: nome, montadora e autonomia dada em km/litro.
- Fazer um algoritmo para ler 5 carros de uma loja de veiculos e imprimir o carro mais econômico.

# SUMÁRIO

- Tipos Abstratos de Dados (TAD)
  - coleção de tipos relacionados entre si
  - formam um único tipo ou TAD
- Os tipos que compõem um TAD podem ser:
  - primitivos (inteiro, real, caracter)
  - outros TADs (Endereco, Aluno, Registro)
- Campos de um TAD: tipos que o compõe
  - Acessados pela variável que instancia o TAD  
Ex: endereco01.rua, alunoA.nome
- **Lista de Atividades 04 – Questões 1 - 3**