



Algoritmos e Programação II

Prof.^a Noeli A. Pimentel Vaz
Prof. Joilson dos Reis Brito

CRONOGRAMA AVALIAÇÕES

2o BIMESTRE

Atividade Prática Avaliativa 3 (APA3) - 06/11/2023

Atividade Prática Avaliativa 4 (APA3) - 27/11/2023

Prova - 11/12/2023

Registros

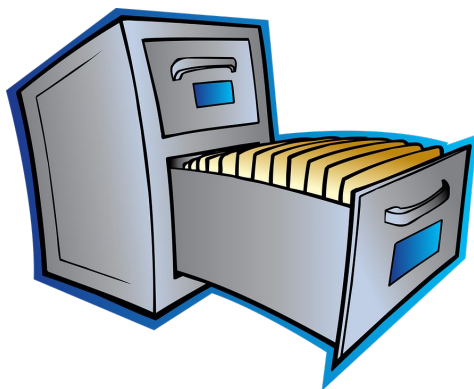


Diagram illustrating a data record structure (Registro) with field labels (Etiqueta de campo) and field data (Dados de campo).

ID do produto	FB9
Nome do produto	Pão de
Categoria	Pães
Preço unitário	\$ 7,95
Estoque	12
Desconto	\$ 1,59

REGISTROS

Vetores e Matrizes são variáveis compostas homogêneas porque todos os seus elementos são do mesmo tipo(caracter, inteiro, real ou lógico).

Vetor Idades

18	88	46	21	10	64
----	----	----	----	----	----

Vetor NomeEscritor

S	t	a	n		L	e	e	\0
---	---	---	---	--	---	---	---	----

Matriz Notas alunos

8,9	7,4	4,7	10,0
0,5	5,6	4,0	3,0
10,0	9,5	9,8	8,9
6,5	7,0	6,5	8,1

REGISTROS

Variáveis compostas heterogêneas são uma coleção de uma ou mais variáveis, sendo possível ser de tipos diferentes, colocadas juntas sob um único nome.

Em algoritmo são chamadas de Registros e em C são criadas com o comando **STRUCT**.

Variáveis compostas heterogêneas são utilizadas para representar um conjunto de informações que estão logicamente relacionadas.

Super-herói	Data da criação
Superman	1938

REGISTROS

Exemplos de informações que são armazenadas em registros:

- Uma ficha de cadastro de uma empresa tem as informações sobre um funcionário, necessárias para a empresa.
- Um cartão de ponto de uma empresa possui informações sobre um funcionário e seus horários de entrada e saída.
- Uma ficha de um consultório médico possui dados e características do estado de saúde dos pacientes.

REGISTROS

Suponha que uma fazenda chamada Rancho Alegre possui uma ficha de cadastro dos funcionários que trabalham na fazenda conforme formato abaixo:

FICHA CADASTRAL FAZENDA RANCHO ALEGRE	
Número Funcionário:	Nome:
Nascimento:	
Data:	País:
Estado:	Cidade:
Qtde dependentes:	Fumante (S/N):
Escolaridade:	Profissão:
Data Ingresso:	Salário:

campo

REGISTROS

Percebe-se que a ficha possui um conjunto de informações relacionadas, ou seja, todas são dados de um funcionário pertencente a fazenda Rancho Alegre. A cada informação dar-se-á o nome de **campo**.

Algumas informações da ficha possuem o mesmo tipo de dado, por exemplo:

- nome, cidade, estado, país, profissão e fumante são do tipo **cadeia de caracteres**.
- Qtde dependentes é um valor do tipo **int** (Ex.: 10 dependentes).
- Salário é uma informação do tipo **float** (Ex.: R\$ 1000,00)

REGISTROS

Para representar esse tipo de estrutura utilizamos o conceito de variáveis compostas heterogêneas ou registro (**struct** em C).

Variáveis compostas heterogêneas (registros) são um conjunto de informações (campos) relacionadas, onde cada campo pode ser de um tipo de dado diferente (**int**, **double**, **char**).

REGISTROS

Criação de um novo tipo:

Até o momento são conhecidos os tipos de dados básicos: inteiro (int), real (float ou double), caracter (char).

Quando for necessário utilizar um registro há a necessidade de definir um novo tipo.

REGISTROS

Sintaxe:

```
struct NOME_DO_NOVO_TIPO
{
    tipo1 campo11, campo12, ..., campo1N;
    tipo2 campo21, campo22, ..., campo2N;
    ...
    tipoM campoM1, campoM2, ... campo MN;
};
```

Onde:

NOME_DO_NOVO_TIPO é o identificador que especifica o nome do registro criado (novo tipo de dados). A partir desse nome podem ser criadas variáveis desse tipo.

tipo1, tipo2, tipo3, ... tipoM são os tipos de cada um dos campos do registro.

campo11, campo12, ..., campoMN são os nomes dos campos do registro.

REGISTROS

Exemplo:

```
struct FichaFuncionario
{
    int Numero, Nascimento, Dependentes, DataIngresso;
    char Nome[40], Pais[40], Estado[40], Cidade[40];
    char Fumante, Escolaridade[40], Profissao[40];
    float Salario;
};
```

REGISTROS

Declaração da variável do tipo registro:

NOME_DO_NOVO_TIPO regVar1, regVar2, ..., regVarN;

Onde:

regVar1, regVar2, ..., regVarN são as variáveis do tipo **NOME_DO_NOVO_TIPO** definido anteriormente.

Exemplo:

FichaFuncionario Funcionario;

REGISTROS

Acesso a um campo do registro:

O acesso aos campos das variáveis do tipo registro é realizado através do operador ponto " . ", conforme abaixo:

nome Variável.campo



símbolo para acesso aos
campos (operador ponto)

Exemplos:

```
Funcionario.Nascimento=01041989;
```

```
gets(Funcionario.Nome);
```

```
printf("Nome: %s Data Nascimento: %i", Funcionario.Nome, Funcionario.Nascimento);
```

RESUMINDO REGISTROS

FichaFuncionario é um tipo criado no programa no qual o registro é criado.

Funcionario é uma variável do tipo **FichaFuncionario**.

Funcionario.Nascimento é um campo da variável **Funcionario** que armazena a data de nascimento de um funcionário.

VETORES DE REGISTROS

Mas se fosse necessário ler as informações de 100 funcionários, eu iria criar 100 variáveis ?



VETORES DE REGISTROS



Exemplo com o registro de funcionários:

```
FichaFuncionario Funcionarios[100];
```

Vetores de Registros

O acesso aos campos será feita normalmente com o índice do vetor:

`Funcionarios[Indice].Nome`

`Funcionarios[Indice].Salario`

Exemplo

Programa que lê os nomes e salários de vários funcionários da Fazenda Rancho Alegre e escreve o nome do com o maior salário.

Exemplo

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

```
#include<stdlib.h>
```

```
#include<locale.h>
```

```
#define Tamanho 3
```

```
// Criação do tipo com as informações do registro
```

```
struct FichaFuncionario
```

```
{
```

```
    int Numero, Nascimento, Dependentes,DataIngresso;
```

```
    char Nome[40],Pais[40],Estado[40],Cidade[40];
```

```
    char Fumante,Escolaridade[40],Profissao[40];
```

```
    float Salario;
```

```
};
```

```
int main()
```

Exemplo

```
{
```

```
    setlocale(LC_ALL, "Portuguese");
```

```
    FichaFuncionario Funcionarios[Tamanho];
```

```
    int l,IndiceMaior;
```

```
    float MaiorSalario;
```

```
    for(l=0;l<Tamanho;l++)
```

```
    {
```

```
        printf("Funcionário %d\n", l+1);
```

```
        fflush(stdin);
```

```
        printf("Nome: ");gets(Funcionarios[l].Nome);
```

```
        printf("Salario: ");scanf("%f",&Funcionarios[l].Salario);
```

```
        system("cls");
```

```
    }
```

```
    MaiorSalario = Funcionarios[0].Salario;
```

```
    IndiceMaior = 0;
```

Exemplo

```
for(l=1;l<Tamanho;l++)  
    if(Funcionarios[l].Salario>MaiorSalario)  
    {  
        MaiorSalario = Funcionarios[l].Salario;  
        IndiceMaior = l;  
    }  
printf("Funcionário com maior salário: %s \n",Funcionarios[IndiceMaior].Nome);  
system("pause");  
return 0;  
}
```