ESTRUTURAS COMPOSTAS

REGISTROS (Struct)

Profa. Dra. Elisa Yumi Nakagawa

2. Semestre de 2017

Slides inicialmente preparados pela Profa. Dra.Simone Senger Souza

Estruturas

- Vetor e Matriz
  - estruturas compostas homogêneas
- Registro
  - estruturas compostas heterogêneas

Registro

- Um registro (struct) é uma coleção de dados (que podem ser de tipos diferentes) sobre um objeto particular.



registro

campo

Cada campo pode ser de qualquer tipo (menos do tipo arquivo)

Registro - Exemplo

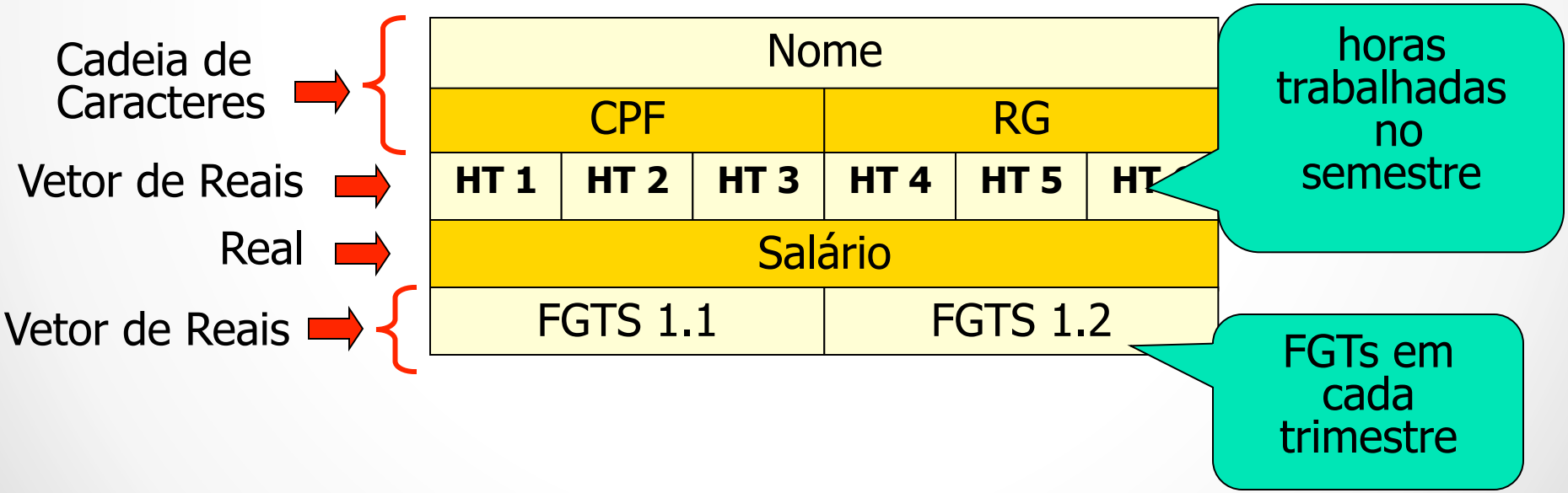
- Registro Aluno:
  - Nome
  - Número USP
  - Data de nascimento
  - Sexo
  - Endereço
  - Telefone
  - Email
  - ...

Registro - Exemplo

- Disciplina
  - Código da disciplina
  - Nome da disciplina
  - Ementa
  - Sistema de avaliação
  - Bibliografia
  - Cursos em que é oferecida
  - ...

Registro - Exemplo

- Registro de Pagamento



Registro

- Cada campo deve ter um nome e deve ser referenciado por este nome

- Não confundir com matriz e vetor onde todos os elementos são do mesmo tipo e são referenciados por um índice

Como definir uma variável tipo registro em C

```
typedef struct  {
        tipo1  campo1;
        tipo2  campo2;
           ...
         tipon campon;
} <nome do registro>;
```

Descrição dos campos

Não é uma variável, é um novo tipo de dado!!

Como definir uma variável tipo registro em C

- **Outra forma de fazer (sem typedef):**

```
struct  REGALUNO {
    char numUSP[10];
    char nome[30];
    char sexo;
    float renda_familiar;
} ;
```

- **Na <u>declaração de variáveis</u> coloca-se:**

struct REGALUNO ALUNO;

Como definir uma variável tipo registro em C

- **No início do programa (em geral, fora de todas as funções):**

```
typedef struct {
    char numUSP[10];
    char nome[30];
    char sexo;
    float renda_familiar;
} REGALUNO;
```

- **Na <u>declaração de variáveis</u> coloca-se:**

REGALUNO ALUNO;

Como fazer Referência a um Elemento de uma Variável do Tipo Struct

<nome-da-variável-tipo-struct> . <nome do campo>

REGALUNO aluno;

- **No Caso do Exemplo**

  uma atribuição de valores poderia ser:

```
ALUNO . numUSP = "1842655";
ALUNO . nome = "Pedro Henrique";
ALUNO . sexo = 'M';
ALUNO . Renda_familiar = 8.500,00 ;
```

Como fazer Referência a um Elemento de uma Variável do Tipo Struct

- **No Caso do Exemplo**

  uma leitura de valores poderia ser:

```
printf("\n\nEntre com o numero USP:");
    gets(ALUNO.numUSP);
    printf("\n\nEntre com o nome:");
    gets(ALUNO.nome);
    printf("\n\nEntre com o sexo(M/F):");
    scanf("%c", &ALUNO.sexo);
    printf("\n\nEntre com a renda familiar:");
    scanf("%f", &ALUNO.renda_familiar);
```

Manipulação da Variável Tipo Struct

- As variáveis do tipo *struct* podem ser manipuladas do mesmo modo que outros dados na memória

- Exemplo:

```
SALBR:=40 * REGEMPR.SALARIO;
```

Exibição da Variável Tipo Struct

- As variáveis do tipo *struct* podem ser exibidas.

- Exemplo:

```
printf ("nome do aluno: %s", ALUNO.NOME);
```

Estrutura dentro de estrutura

```
struct tipo_endereco
 {
    char rua [50];
    int numero;
    char bairro [20];
    char cidade [30];
    char sigla_estado [3];
    char CEP [10];
 };
```

```
struct ficha_pessoal
    {
      char nome [50];
      long int telefone;
      struct tipo_endereco
                        endereco;
     };
```

Estrutura dentro de estrutura

```
typedef struct {
    int dia, mes, ano;
}REGDATA;
typedef struct{
        char nome[20];
        REGDATA nasc, contrato;
        float salario;
        } REGF;
REGF dados[100];
```

```
dados[i].nasc.dia = 1;
dados[i].nasc.mes = 12;
dados[i].nasc.ano = 1980;
...
dados[i].contrato.dia = 5;
dados[i].contrato.mes = 3;
dados[i].contrato.ano = 2007;
```

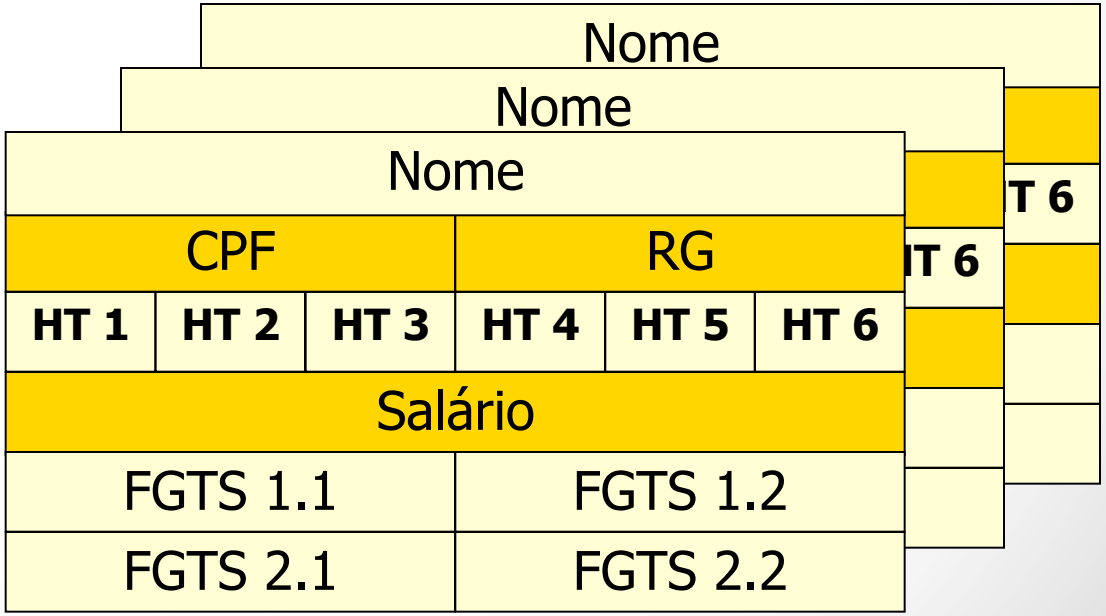
Vetor de Registros



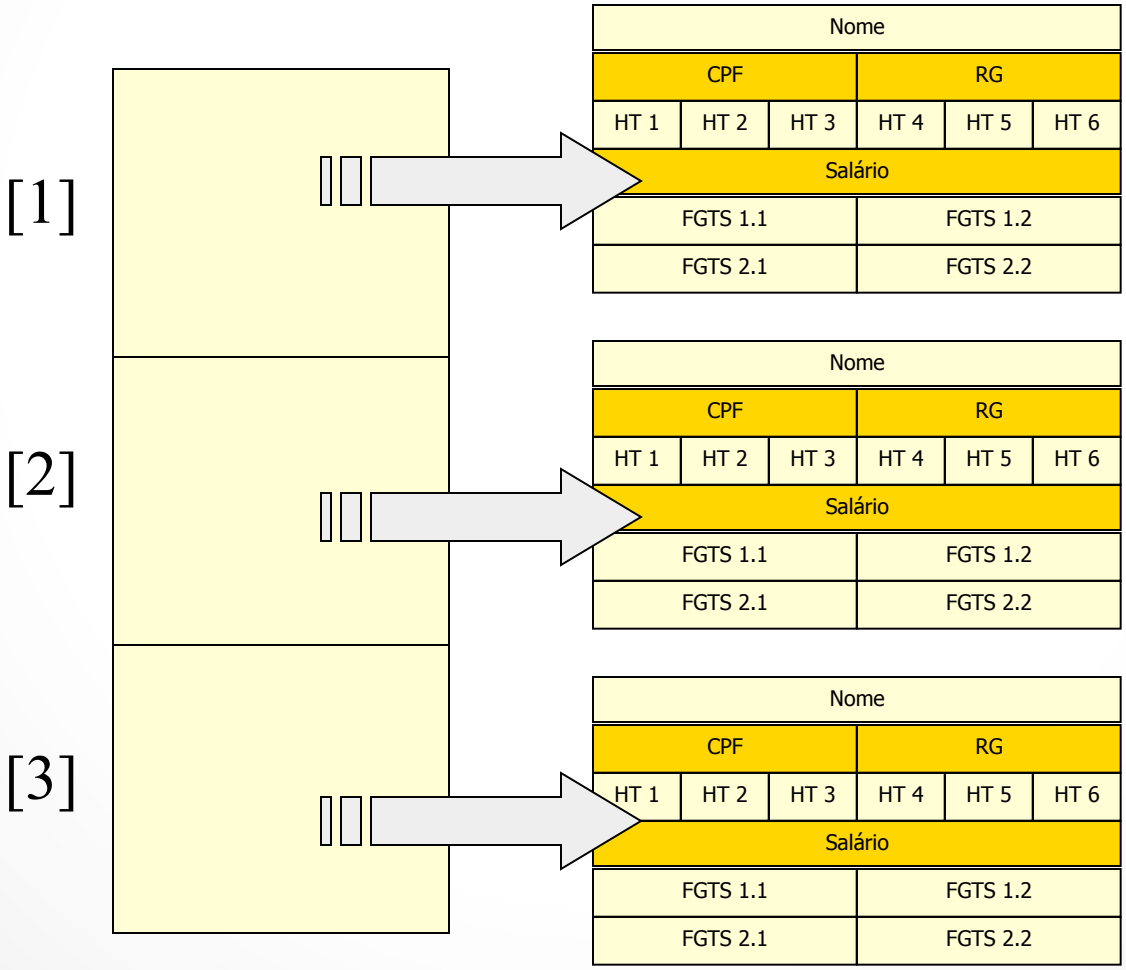
- Se, ao invés de um <u>única</u> ficha do empregado, quisermos cadastrar <u>várias</u> fichas?

SOLUÇÃO

criar um <u>vetor</u> de <u>registros</u> !



Vetor de Registros



Como definir um vetor de registros

- Na seção de declaração de tipo coloca-se:

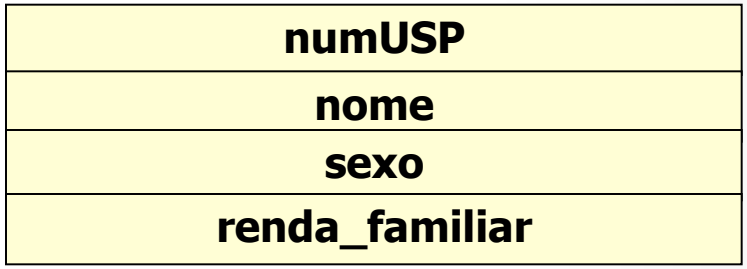
```
typedef struct  {
                        tipo1  campo1;
                        tipo2  campo2;
                           ...
                         tipon campon;
              } <nome do registro>;

<nome-do-registro> <nome-da-variavel [dimensao]>
```

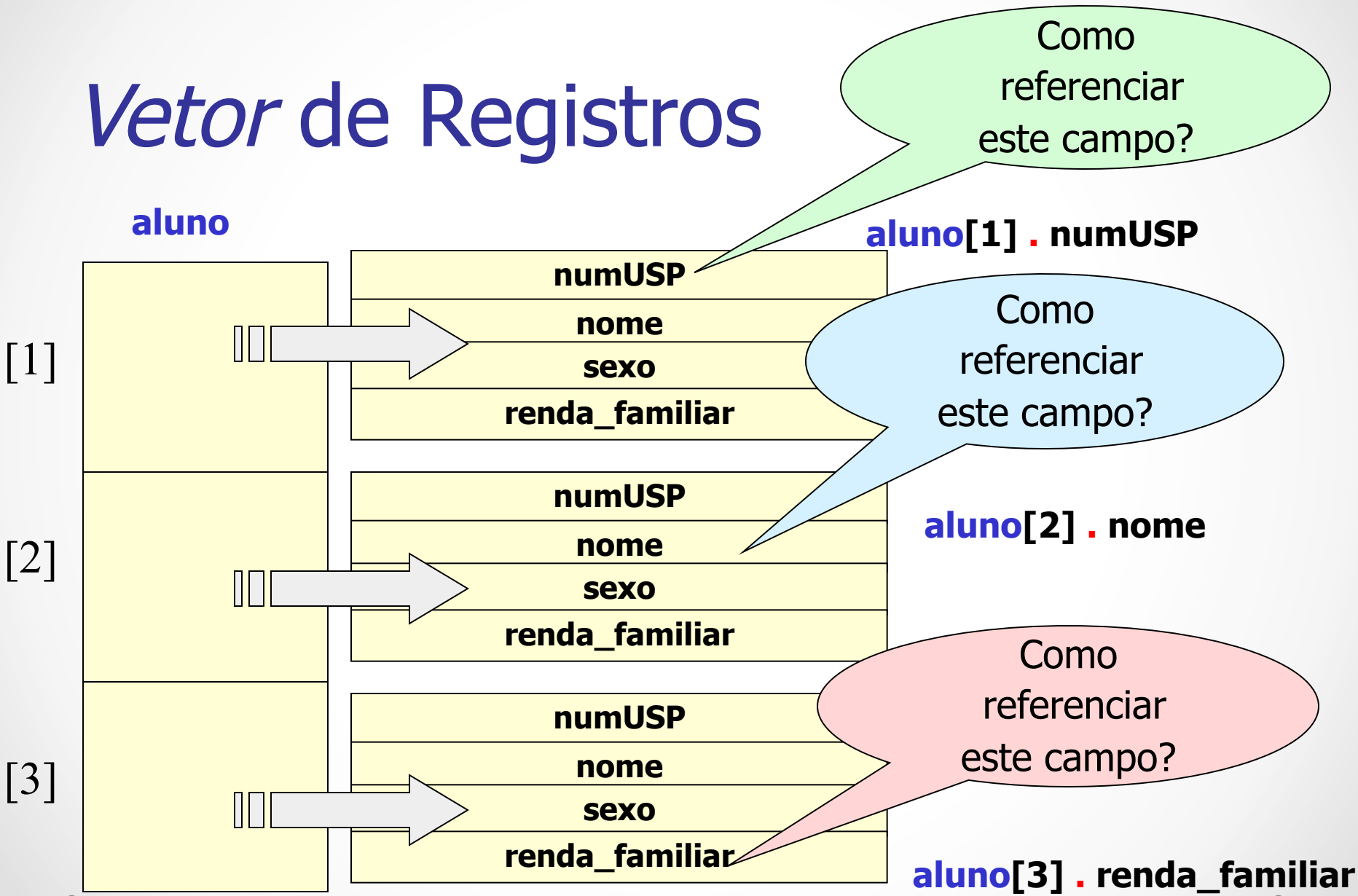
No Exemplo

```
typedef struct {
      char numUSP[10];
      char nome[30];
      char sexo;
      float renda_familiar;
   } REGALUNO;
```

REGALUNO aluno[42];



Vetor de Registros



Como definir variável tipo registro em C com vetor

- **No início do programa (em geral, fora de todas as funções):**

```
typedef struct {
        int passagem, num_poltrona;
        float dist;
        char origem[20], destino[20];
    } REGPASSAGEM;
```

- **Na <u>declaração de variáveis</u> coloca-se:**

REGPASSAGEM onibus[40];

Estruturas e Funções

- Estruturas são passadas por valor
- Exemplo:

```
typedef struct{
        char nome[20];
        int idade;
        } REG;
REG dados;
```

```
void ler(REG dados)

int main()
{
...
ler(dados);
}
```

Os dados não são alterados na função main()

Estruturas e Funções

- Ponteiros para estruturas: possibilita a passagem por referência
- Exemplo:

```
typedef struct{
        char nome[20];
        int idade;
        } REG;
REG dados;
```

```
void ler(REG *dados)

int main()
{
...
ler(&dados);
}
```

Estruturas e Funções

- Ponteiros para estruturas: possibilita a passagem por referência
- Exemplo:

```
typedef struct{
        char nome[20];
        int idade;
        } REG;
REG dados;
```

```
void ler(REG *dados)
{
    dados->nome= "Felipe"
    dados->idade = 16;
}
int main()
{
      ler (&dados);
}
```

Exercício 1

- Fazer uma função que recebe como parâmetro a variável ônibus (preenchida) e conta e retorna quantos passageiros tem o destino "São Paulo".

```
typedef struct {
        int passagem, num_poltrona;
        float dist;
        char fumante;
        char origem[20], destino[20];
} REGPASSAGEM;
REGPASSAGEM onibus[40];
```

Exercício 2

- Escrever um programa que:
  a) Lê uma tabela com nomes e idades de jogadores de futebol. O usuário entra com nomes até um nome vazio para encerrar a inclusão dos dados na tabela.
  b) Faça uma função que, dada uma idade, mostrar todos os jogadores que possuem essa idade.