Estruturas

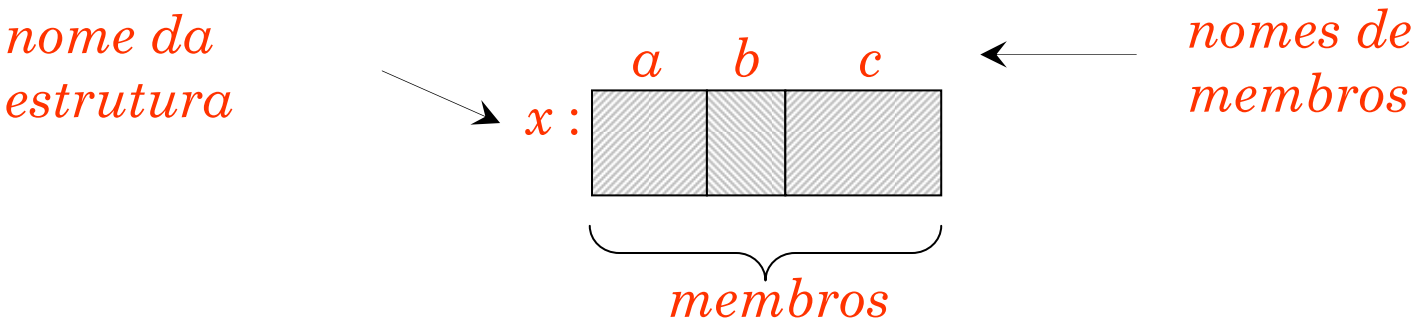
Prof. Dr. Silvio do Lago Pereira

Departamento de Tecnologia da Informação

Faculdade de Tecnologia de São Paulo

Estruturas

- **Uma estrutura é um tipo de dados agregado e hetero-gêneo, cujos itens são identificados por nomes.**



Exemplo: estrutura data

```
typedef struct {
    int dia;
    int mes;
    int ano;
} data;

Podemos escrever:

data hoje;
hoje.dia = 19;
hoje.mes = 7;
hoje.ano = 2000;
```

Inicialização de estrutura

- A estrutura deve ser global ou estática.
- Os valores iniciais devem ser constantes.

```
static data hoje = { 19, 7, 2000 };
```

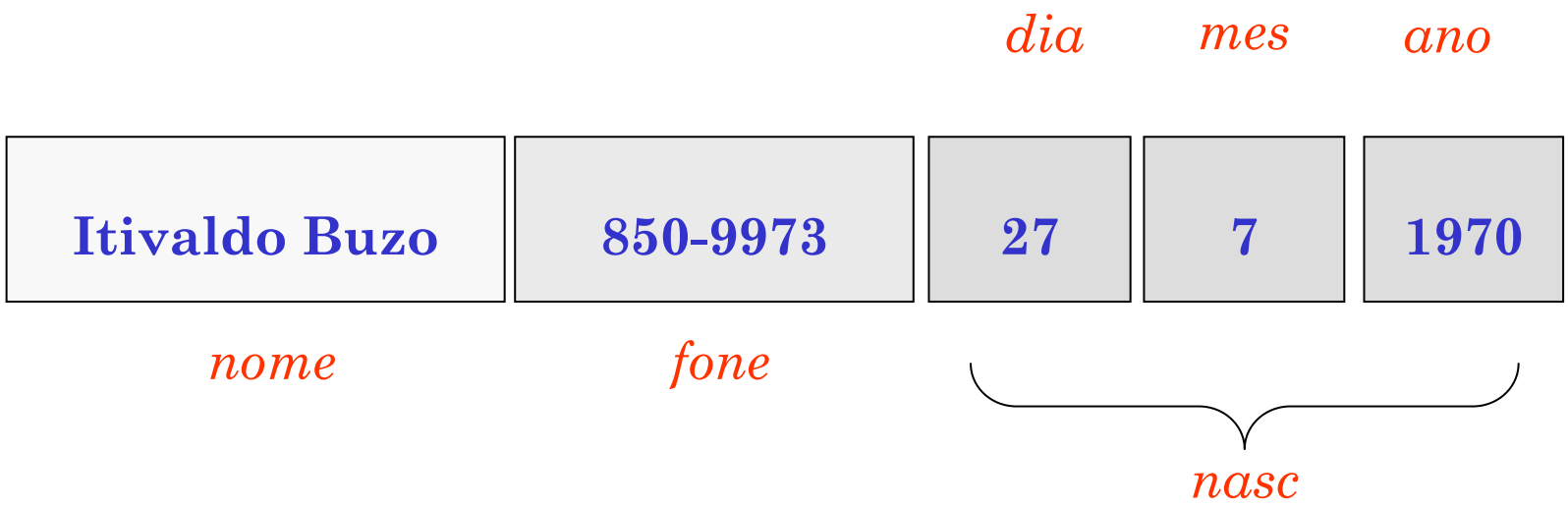
Exemplo: estrutura pessoa

```
typedef struct {
    int dia;
    int mes;
    int ano;
} data;

typedef struct {
    char nome[31];
    char fone[21];
    data nasc;
} pessoa;
```

Armazenamento

```
static pessoa p = {
    "Itivaldo Buzo",
    "850-9973",
    {27, 7, 1970}
};
```



Acesso aos campos da estrutura

```
pessoa amigo;

strcpy(amigo.nome, "Itivaldo Buzo");

strcpy(amigo.fone, "850-9973");

amigo.nasc.dia = 27;

amigo.nasc.mes = 7;

amigo.nasc.ano = 1970;
```

Vetores de estruturas: tabelas

```
static pessoa agenda[5] = {
  {"Itivaldo Buzo",  "850-9973", {27, 7,1970}},
  {"Roberto Soares", "266-0879", {15,11,1971}},
  {"Márcia Ueji",    "576-8292", { 9, 5,1966}},
  {"Silvio Lago",    "851-7715", {18, 3,1968}},
  {"Mie Kobayashi",  "834-0192", { 4,12,1973}}
};
```

Armazenamento da tabela

	nome	fone	nasc
0	Itivaldo Buzo	850-9973	27/07/1970
1	Roberto Soares	266-0879	15/11/1971
2	Márcia Ueji	576-8292	09/05/1966
3	Silvio Lago	851-7715	18/03/1968
4	Mie Kobayashi	834-0192	04/12/1973

Acesso aos campos na tabela

```
strcpy(agenda[1].nome,"Roberta Soares");
strcpy(agenda[1].fone, "266-0879");


agenda[1].nasc.dia = 15;
agenda[1].nasc.mes = 11;
agenda[1].nasc.ano = 1971;
```

Exercício

Codifique uma função para preencher a agenda.

```
void preenche(pessoa a[], int n) {
 int i;

 for(i=0; i<n; i++) {
  printf("\n%da pessoa\n",i);
  printf("\nNome? "); gets(a[i].nome);
  printf("\nFone? "); gets(a[i].fone);
  printf("\nAniv? ");
  scanf("%d/%d/%d%*c", &a[i].nasc.dia,
                       &a[i].nasc.mes,
                       &a[i].nasc.ano,
 }
}
```

Exercício

Codifique uma função para exibir a agenda.

```
void exibe(pessoa a[], int n) {
 int i;

 for(i=0; i<n; i++) {
  printf("\n%da pessoa\n", i);
  printf("\nNome: %s", a[i].nome);
  printf("\nFone: %s", a[i].fone);
  printf("\nAniv: %02d/%02d/%d",
          a[i].nasc.dia,
          a[i].nasc.mes,
          a[i].nasc.ano,
 }
}
```

Exercício

Codifique uma função para ordenar agenda por nome.

```
void ordena(pessoa a[], int n) {
 int i, j;
 pessoa p;

 for(i=1; i<n; i++)
  for(j=0; j<n-i; j++)
   if( strcmp(a[j].nome,a[j+1].nome)>0 ) {
      p = a[j];
      a[j] = a[j+1];
      a[j+1] = p;
   }
}
```

Exercício

Codifique programa que lê e exibe agenda ordenada.

```
void main(void) {
  pessoa a[5];

  preenche(a,5);
  ordena(a,5);
  exibe(a,5);
}
```

Fim