

UEM/CTC – Departamento de Informática
Curso: Ciência da Computação
Professor: Flávio Rogério Uber

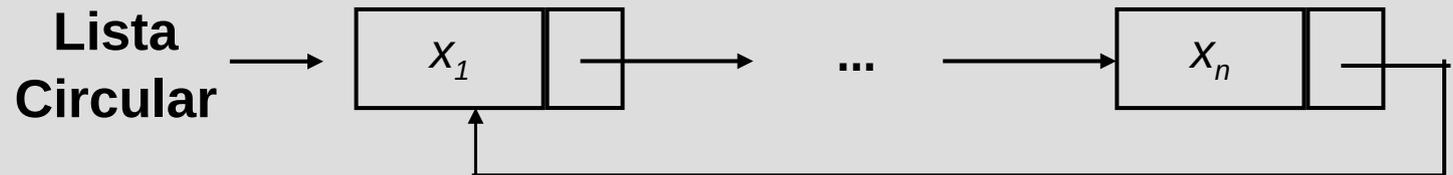
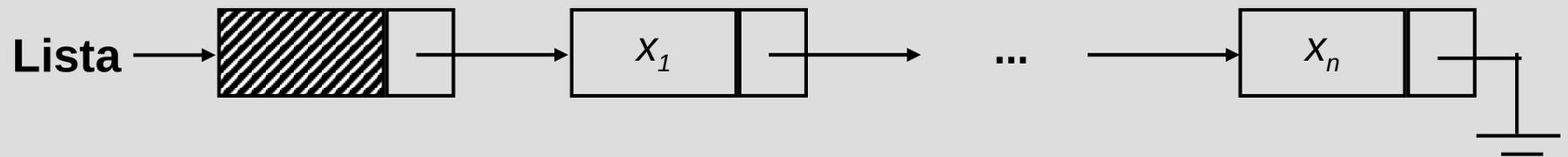
Estrutura de Dados
(6884/3 e 4)

Listas Circulares

Listas Circulares

- Lista encadeada onde o último nó é ligado ao primeiro
- Mesmo sendo circular existe a identificação, através de um ponteiro, do primeiro elemento da lista

Listas Circulares

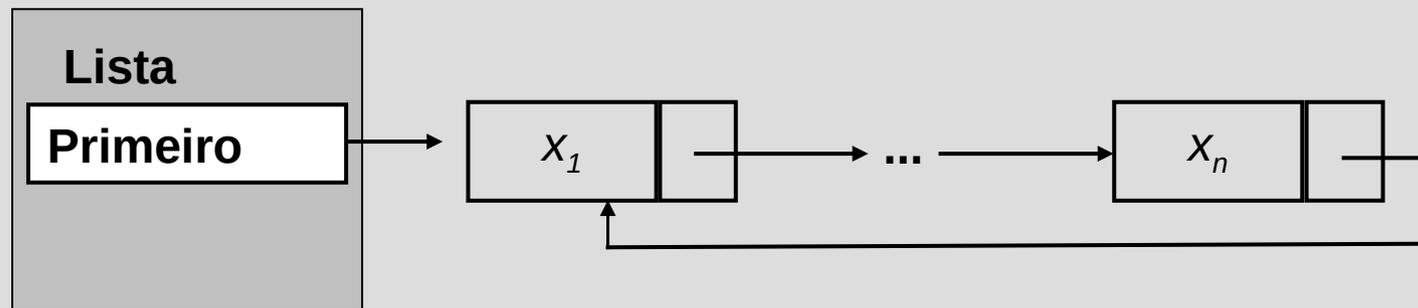


Listas Circulares

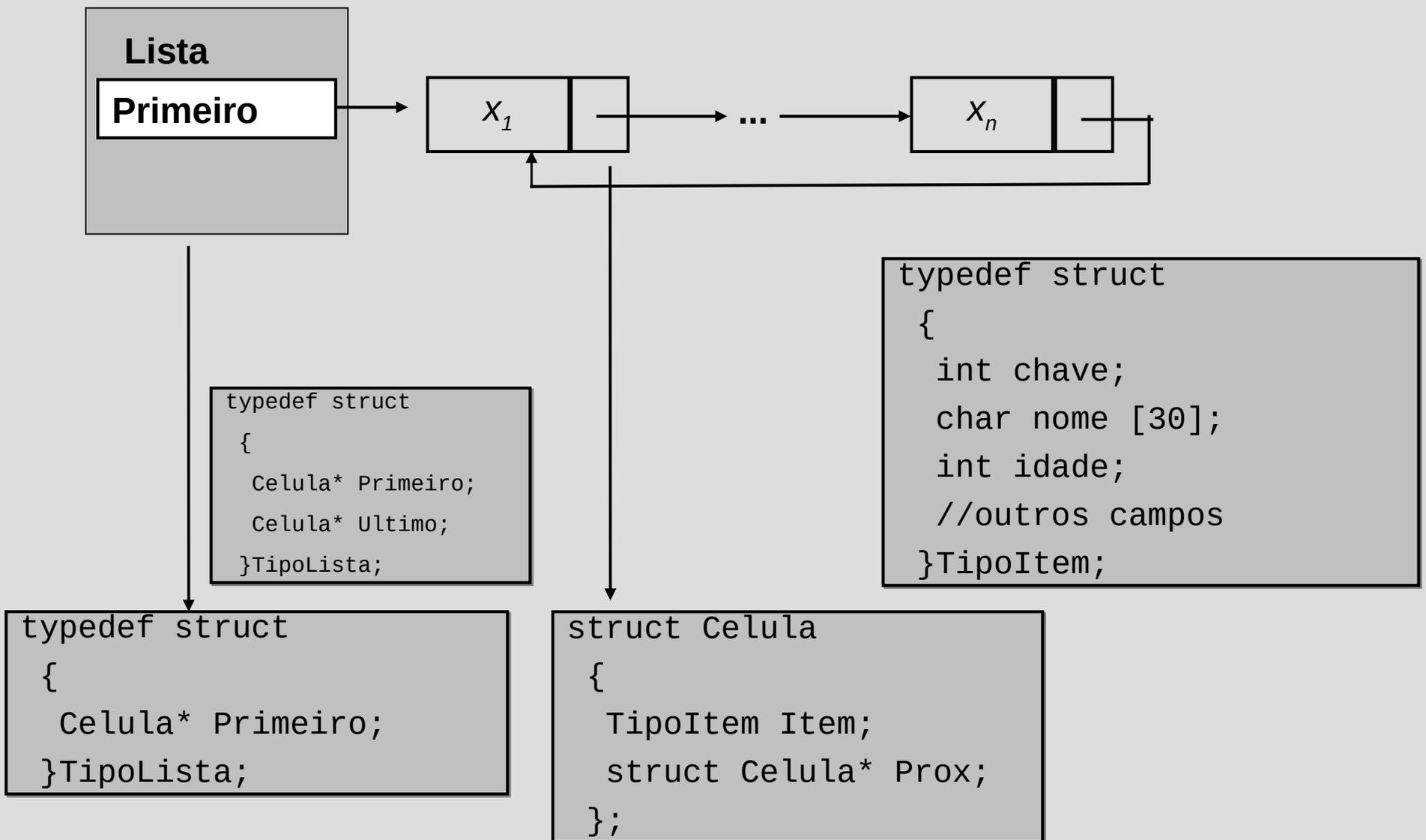
- Inserção/Remoção
 - Todo elemento tem um “próximo”
 - Não há casos especiais de inserção ou remoção no final da lista

Listas Circulares

- Implementação
 - O registro TipoLista não precisa mais de um apontador para a última célula da lista;
 - Também é desnecessário a cabeça da lista. Um único ponteiro determinando o início da mesma é suficiente.



Listas Circulares



Listas Circulares

```
void FLVazia (TipoLista * Lista);
```



```
void FLVazia (TipoLista *Lista)
{
    Lista -> Primeiro = (struct Celula*) malloc(sizeof(struct
Celula));
    Lista -> Ultimo = Lista -> Primeiro;
    Lista -> Primeiro -> Prox = NULL;
}
```

```
void FLVazia (TipoLista *Lista)
{
    Lista -> Primeiro = NULL;
}
```

Listas Circulares

int Vazia (TipoLista Lista);



?

```
int Vazia (TipoLista Lista)
{
    return (Lista.Primeiro == Lista.Ultimo);
}
```

```
int Vazia (TipoLista Lista)
{
    return (Lista.Primeiro == NULL);
}
```

```

void Insere (TipoItem x, TipoLista *Lista)
{
    struct *Celula pins;
    struct *Celula paux;
    pins = (struct *Celula) malloc(sizeof(struct Celula));
    if (Lista->Primeiro == NULL)
    {
        Lista->Primeiro = pins;
        pins->Prox = pins;
    }
    else
    {
        paux = Lista->Primeiro;
        while (paux->Prox != Lista->Primeiro)
        {
            paux = paux->Prox;
        }
        pins->Prox = Lista->Primeiro;
        paux->Prox = pins;
    }
    pins->Item = x;
}

```

```

void Insere (TipoItem x, TipoLista *Lista)
{
    Lista->Ultimo->Prox = (struct Celula*) malloc(sizeof(struct
Celula));
    Lista->Ultimo = Lista->Ultimo->Prox;
    Lista->Ultimo->Item = x;
    Lista->Ultimo->Prox = NULL;
}

```

Lista Circular

- 1) Faça um procedimento para contar os elementos de uma lista encadeada circular.
- 2) Faça um procedimento para criar uma segunda lista encadeada circular idêntica a uma já existente.