



Anhanguera

*Aqui o seu esforço
ganha força.*



Anhanguera

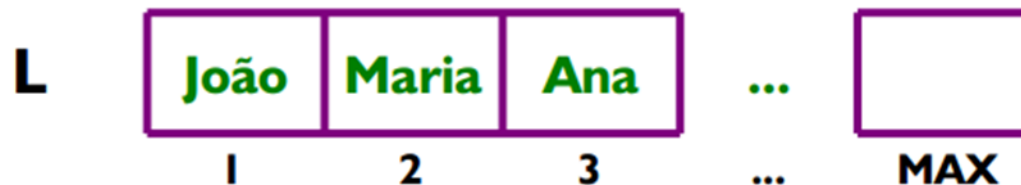
Listas Simplesmente Encadeadas

Prof. Esp. Rodrigo Hentz



Tipos de listas

- Veremos os conceitos de aplicação de dois tipos de listas:
- Sequencial e Encadeada
- Na lista sequencial os nós são armazenados em endereços sequencias, como um vetor.



- Na lista encadeada os nós são sequenciados através de ponteiros.



Listas Simplesmente Encadeadas

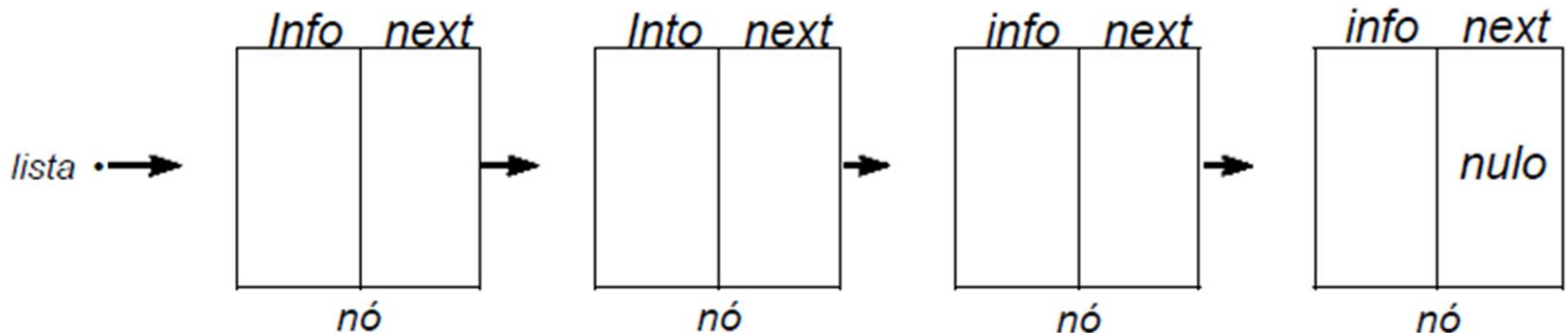
- Também encontradas na literatura com o nome de listas ligadas.
- Em uma lista simplesmente encadeada, para cada novo elemento inserido na estrutura, alocamos um espaço de memória para armazená-lo.
- Fazemos uso da alocação dinâmica de memória.
- Desta forma, o espaço total de memória gasto pela estrutura é proporcional ao número de elementos nela armazenado.

Listas Simplesmente Encadeadas

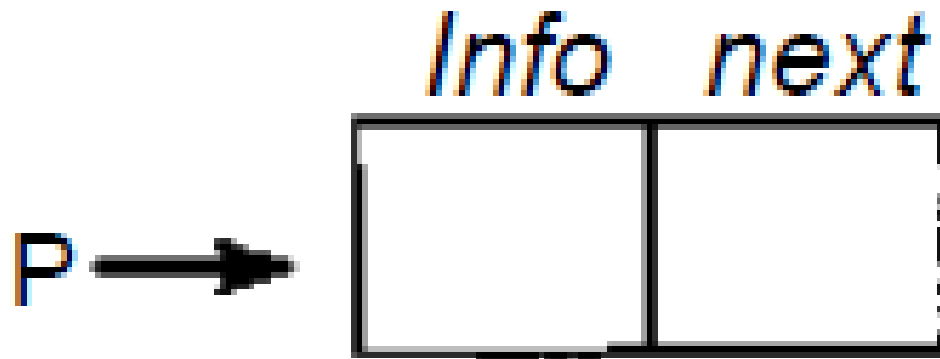
- No entanto, não podemos garantir que os elementos armazenados na lista ocuparão um espaço de memória contíguo, portanto não temos acesso direto aos elementos da lista.
- Para que seja possível percorrer todos os elementos da lista, devemos explicitamente guardar o encadeamento dos elementos, o que é feito armazenando-se, junto com a informação de cada elemento, um ponteiro para o próximo elemento da lista.

Listas Simplesmente Encadeadas

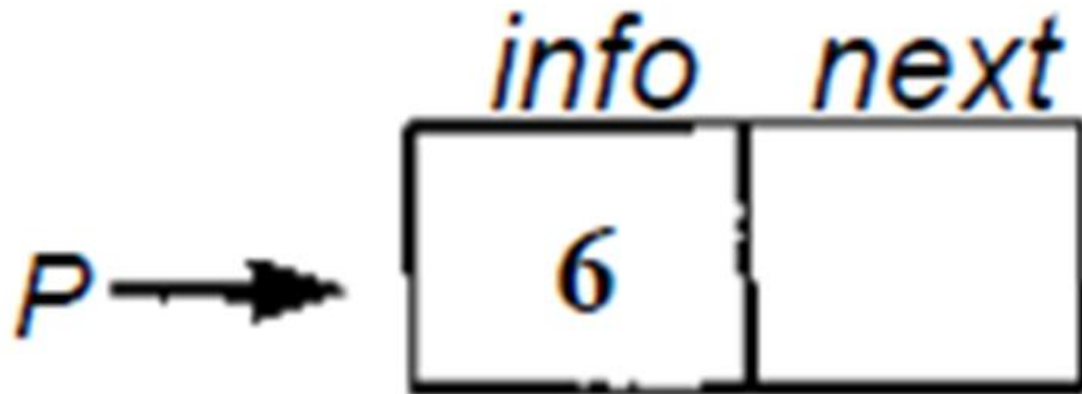
- Cada item na lista é chamada de nó e contém dois campos, um campo de informação e um campo do endereço do próximo nó.



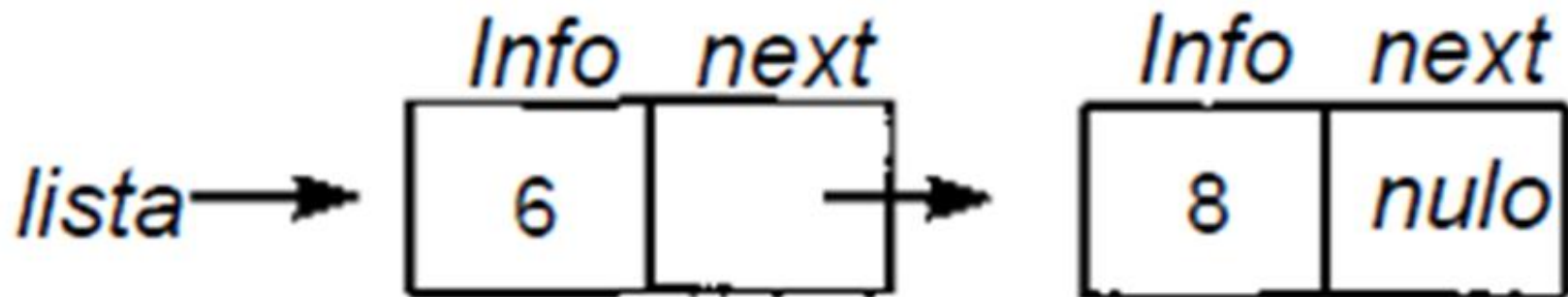
Nó



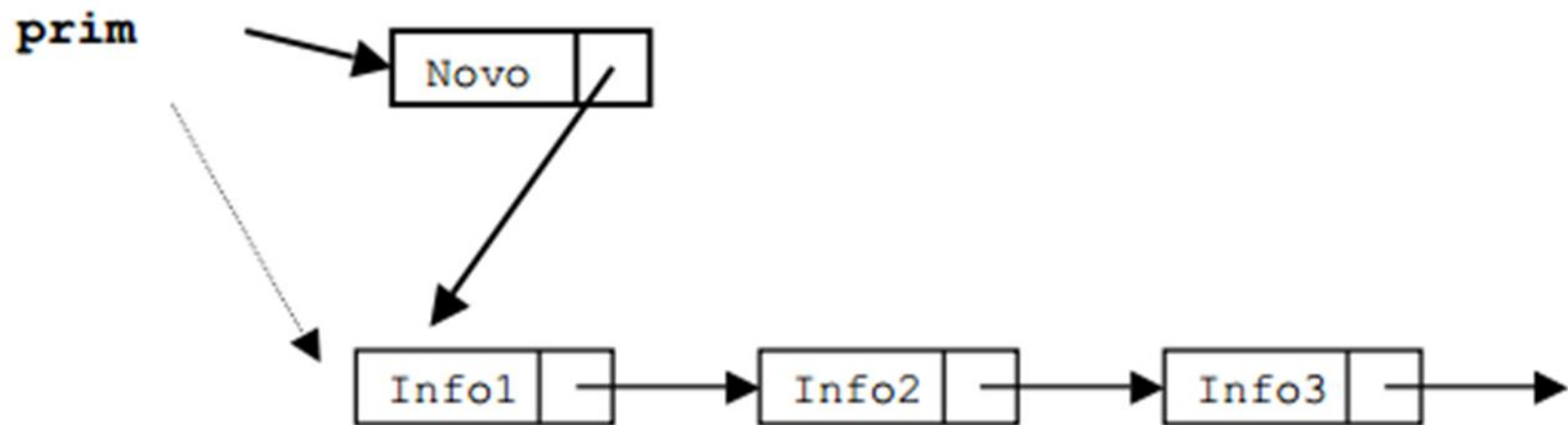
Nó



Nó

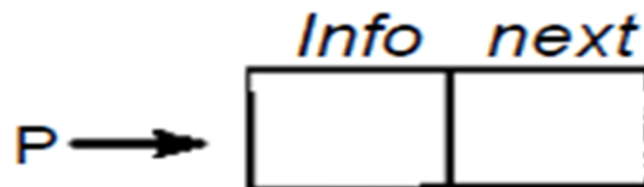


Incluir nó no início



Criação da estrutura nó

```
typedef struct no {  
    int info;  
    struct no* next;  
} sLista, sNo;
```



Iniciar a lista

```
sLista* inicializaLista()  
{  
    printf("\nLista criada.");  
    return NULL;  
}
```

Criar nó

```
sNo* criarNo(int valor)
{
    sNo* p = (sNo*)malloc(sizeof(sNo));
    p->info = valor;
    p->next = NULL;
    return p;
}
```

Inserir no início

```
sLista* inserirInicio(sNo* no, int valor)
{
    sNo* novo = criarNo(valor);
    novo->next = no;
    return novo;
}
```

Imprimir lista

```
void imprimirLista(sLista* lista)
{
    sNo* p;
    for (p = lista; p != NULL; p = p->next)
        printf("info = %d\n", p->info);
}
```

Inserir no final da lista

```
sLista* inserirFinal(sLista* lista, int valor)
{
    sLista* p = lista;
    if (lista == NULL) return inserirInicio(lista, valor);
    else
    {
        while(p->next != NULL) p = p->next;
        p->next = criarNo(valor);
        return lista;
    }
}
```




Anhanguera

*Aqui o seu esforço
ganha força.*