



Estruturas de Dados
Aula 9 -2: Listas
simplesmente encadeadas
sem sentinela e sem
cabeçalho

Fontes Bibliográficas

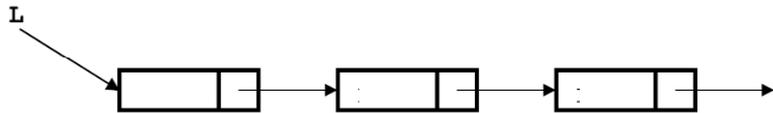


- Waldemar Celes, Renato Cerqueira, José Lucas Rangel, *Introdução a Estruturas de Dados*, Editora Campus (2004)
 - Capítulo 10 - Listas encadeadas

Lista Encadeada Simples



- Considere uma lista encadeada simples, sem célula cabeçalho e sem "sentinela":



```
typedef struct tipoitem TipoItem;  
typedef struct celula_str TipoLista;
```

Lista Encadeada Simples

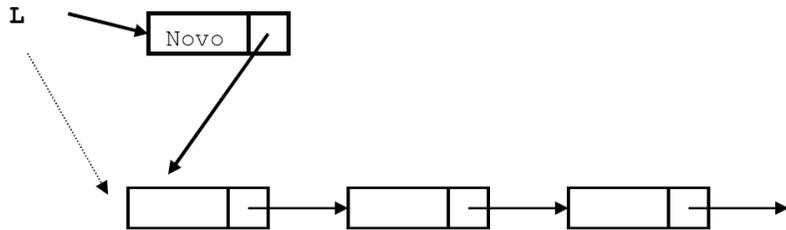


```
struct tipoitem{  
    int valor;  
    /* outros componentes */  
};  
  
struct celula_str{  
    TipoItem Item;  
    struct celula_str* Prox;  
};
```

Função Inserir



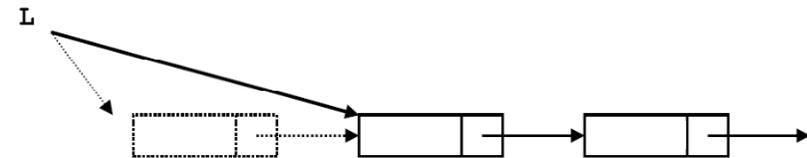
```
TipoLista* lst_inserere (TipoLista* l, TipoItem* item)
{
    /* cria uma nova celula */
    TipoLista* novo = (TipoLista*) malloc(sizeof(TipoLista));
    novo->Item = *item;
    novo->Prox = l;
    return novo;
}
```



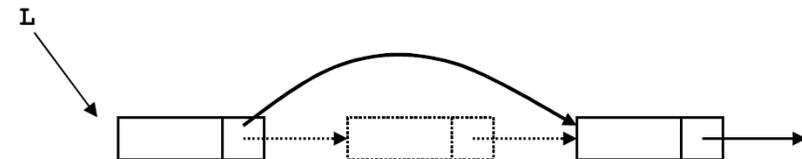
Função Retirar



- Recebe como entrada a lista e o valor do elemento a retirar
- Atualiza o valor da lista, se o elemento removido for o primeiro



- Caso contrário, apenas remove o elemento da lista



Função Retirar (código)



```
TipoLista* lst_retira (TipoLista* l, int v)
{
    TipoLista* ant = NULL;
    TipoLista* p = l;
    while (p != NULL && p->Item.valor != v)
    { ant = p;
      p = p->Prox;}
    if (p == NULL)
        return l;
    if (ant == NULL){
        l = p->Prox;}
    else{
        ant->Prox = p->Prox;}
    free(p);
    return l;
}
```

Outras Operações



- Imprime lista:

```
void lst_imprime (TipoLista* l)
{
    TipoLista* p;
    for (p=l; p!=NULL; p=p->Prox)
        printf ("Valor = %d\n", p->Item.valor);
}
```

- Busca elemento na lista:

```
TipoLista* busca (TipoLista* l, int v)
{
    TipoLista* p;
    for (p=l; p!=NULL; p = p->Prox) {
        if (p->Item.valor == v)
            return p;
    }
    return NULL; /*não encontrou o elemento */
}
```