TAD Lista Simplesmente Encadeada com Nó Cabeçalho

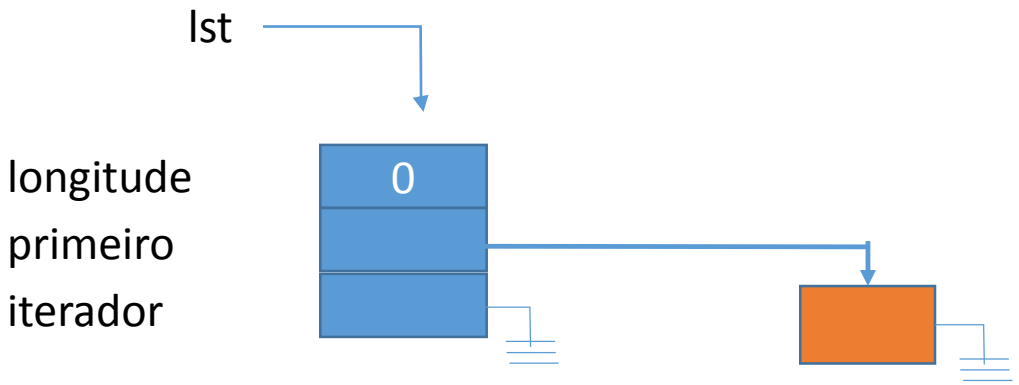
Prof. D.Sc. Saulo Ribeiro

Lista simplesmente encadeada com Nó Cabeçalho

- Esta maneira de representar a lista resolve o problema de inserção, colocando o **apontador** uma **posição atrás** em relação ao elemento que se quer indicar.
- Com o objetivo de que o primeiro elemento não se converta em um caso particular, se coloca um Nó adicional, sem informação válida no começo da lista(Nó cabeçalho).

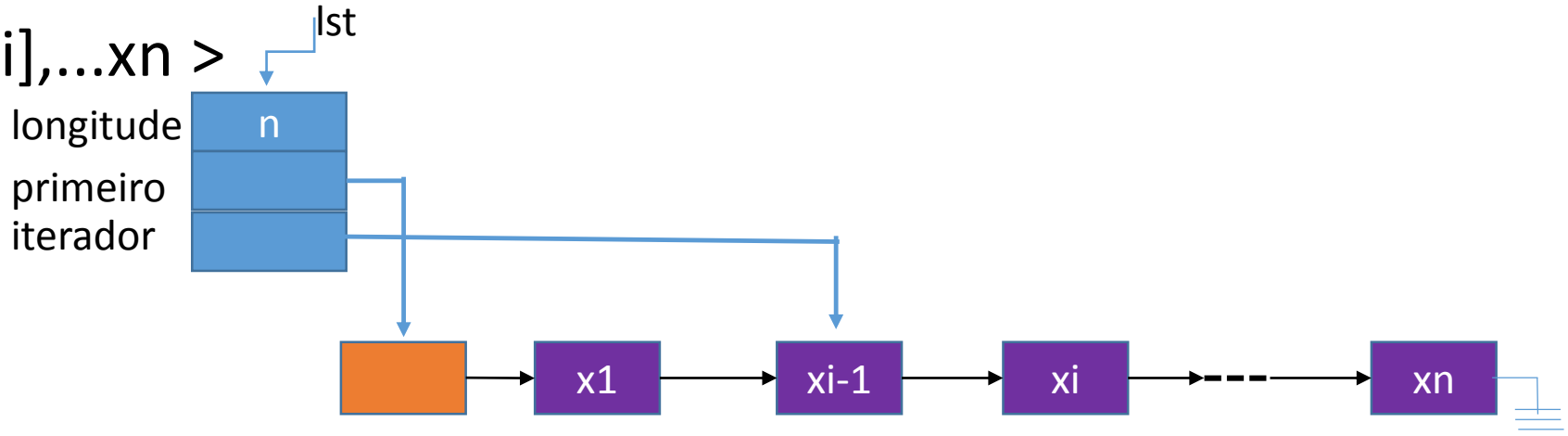
Lista simplesmente encadeada com Nó Cabeçalho

- **Lista vazia**: o **iterador** está indefinido: não aponta para ninguém (NULL). Primeiro aponta para o nó cabeçalho.
- lst = < > [ ]



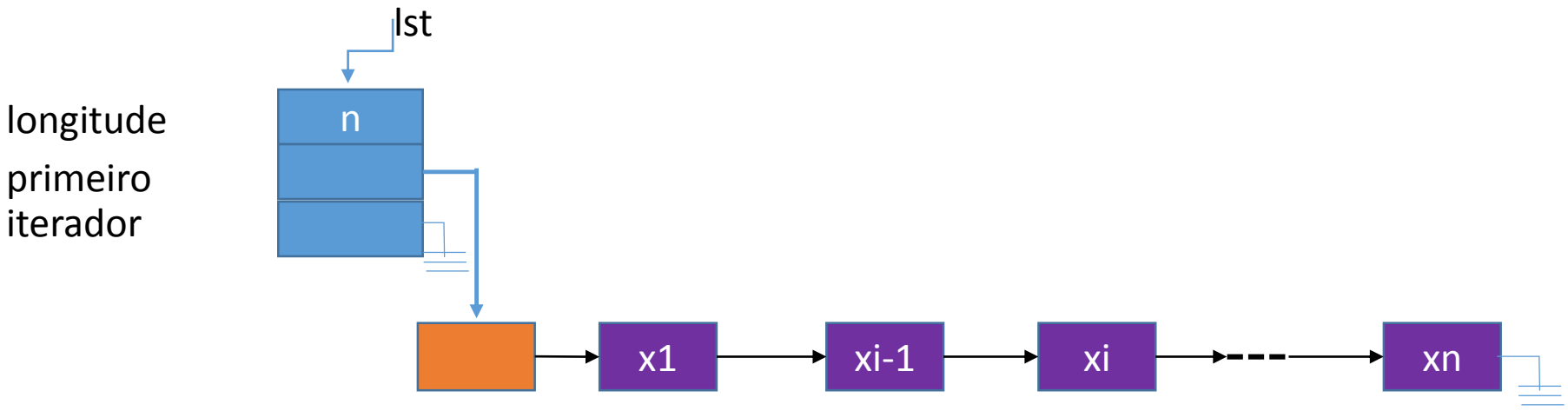
Lista simplesmente encadeada com Nó Cabeçalho

- **Lista cheia com o iterador definido** (iterador aponta para um elemento da lista, [xi]). Primeiro aponta para o nó cabeçalho. Mas, sei que o primeiro elemento válido da lista, vem logo após.
- O iterador fica posicionado uma posição atrás(xi-1) em relação ao elemento para qual realmente está sob o iterador(xi).
- lst = < x1,...,[xi],...xn >



Lista simplesmente encadeada com Nó Cabeçalho

- **Lista cheia com o iterador indefinido** (iterador aponta NULL).
- Primeiro aponta para o nó cabeçalho.
- lst = < x1,...,xi,...xn >[ ]



Lista simplesmente encadeada com Nó Cabeçalho


```
typedef int TipoL;

typedef struct ListaNo{
        TipoL info;
        struct ListaNo *prox;
} *pListaNo;

typedef struct{
        pListaNo primeiro, iterador;
        int longitude;
} Tlista, * Lista;
```