# Lista Simplesmente Encadeada Circular

## Estrutura de Dados — QXD0010

Prof. Atílio Gomes Luiz
gomes.atilio@ufc.br

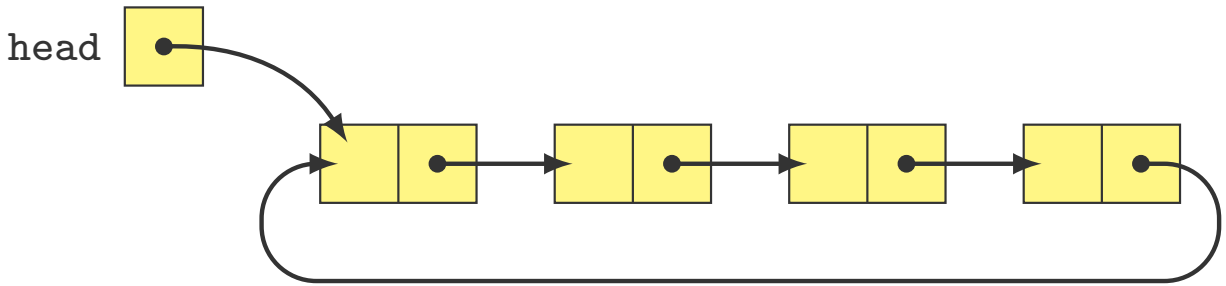
Universidade Federal do Ceará
Campus Quixadá

2º semestre/2022

# Introdução

# Lista simplesmente encadeada circular

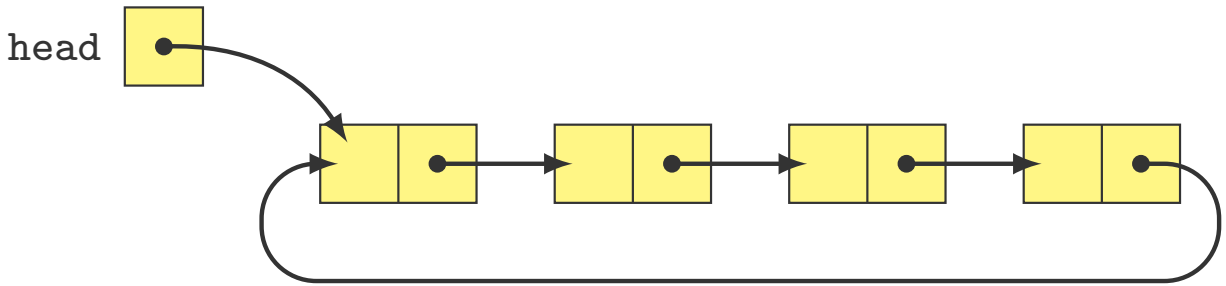
Lista circular (sem nó cabeça):



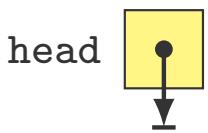
# Lista simplesmente encadeada circular

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Lista circular (sem nó cabeça):



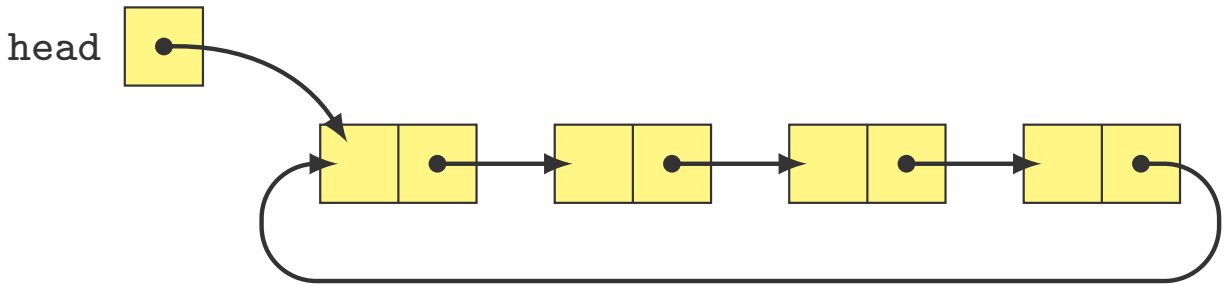

Lista circular vazia: ponteiro `head` é nulo.



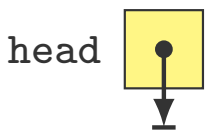
# Lista simplesmente encadeada circular

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Lista circular (sem nó cabeça):




Lista circular vazia: ponteiro `head` é nulo.

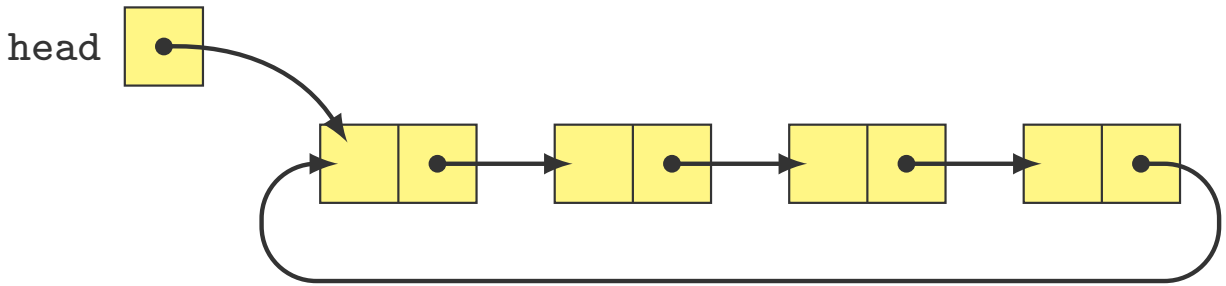



Exemplo de aplicações:

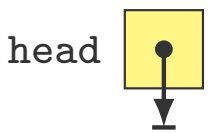
# Lista simplesmente encadeada circular

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Lista circular (sem nó cabeça):




Lista circular vazia: ponteiro `head` é nulo.



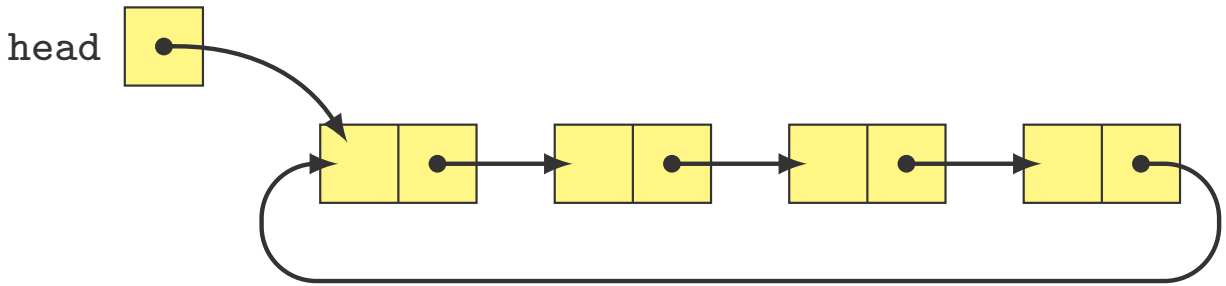

Exemplo de aplicações:

- Execução de processos no sistema operacional

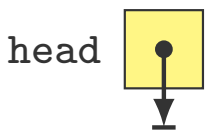
# Lista simplesmente encadeada circular

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Lista circular (sem nó cabeça):




Lista circular vazia: ponteiro `head` é nulo.




Exemplo de aplicações:

- Execução de processos no sistema operacional
- Controlar de quem é a vez em um jogo de tabuleiro

# Implementação em C++

# Node.h



```
1  #ifndef NODE_H
2  #define NODE_H
3
4  using Item = double;
5
6  struct Node {
7      Item value;
8      Node* next;
9
10     Node(const Item& val, Node *nextPtr) {
11         value = val;
12         next = nextPtr;
13     }
14 };
15
16 #endif
```

# List.h — Arquivo de Cabeçalho



```
1  #ifndef CLIST_H
2  #define CLIST_H
3  #include <string>
4  #include "Node.h"
5
6  class CircularList {
7  private:
8      Node *m_head {nullptr};
9      int m_size {0};
10 public:
11     CircularList() = default;
12     bool empty() const { return m_head == nullptr; }
13     int size() const { return m_size; }
14     void push_back(const Item& val);
15     void pop_back();
16     Item& operator[](int index);
17     const Item& operator[](int index) const;
18     std::string toString() const;
19     void clear();
20     ~CircularList();
21 };
22
23 #endif
```

# main.cpp — Programa Cliente

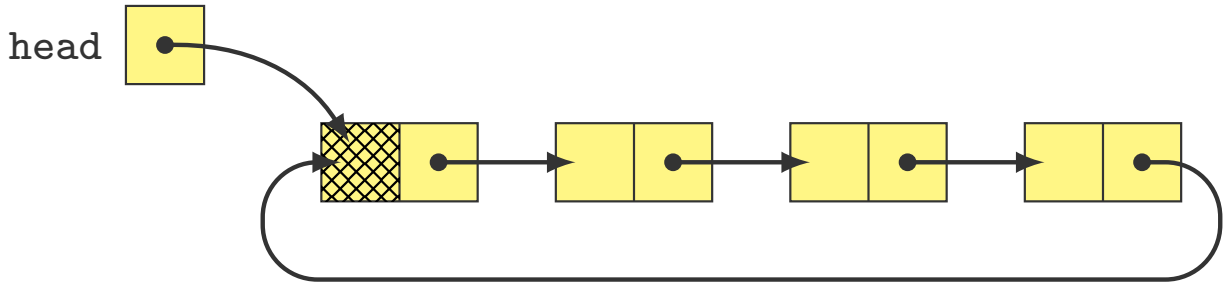


```cpp
#include <iostream>
#include "CircularList.h"
using namespace std;

void print(const CircularList& lst) {
    for(int i = 0; i < lst.size(); ++i)
        cout << lst[i] << " ";
    cout << endl;
}

int main() {
    CircularList lst;

    for(int i = 1; i <= 9; ++i)
        lst.push_back(i * 0.5);

    cout << lst.toString() << endl;
}
```

# Variações

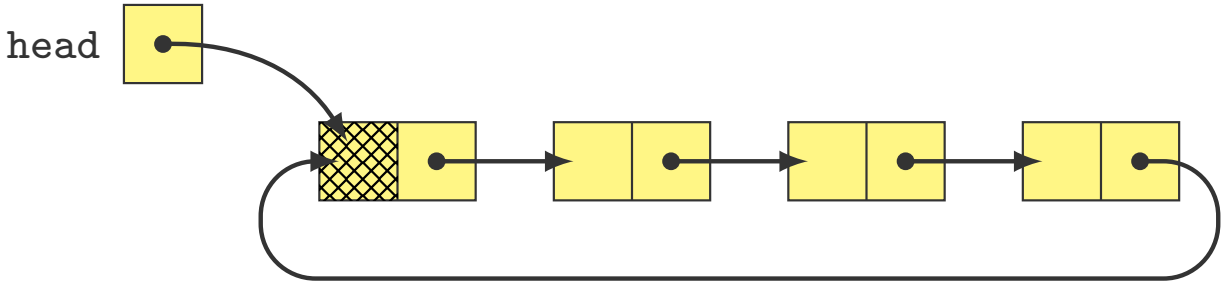
# Variações — Listas circulares com nó sentinela

Lista circular com nó sentinela:

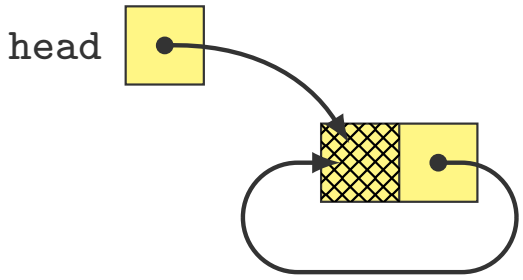


# Variações — Listas circulares com nó sentinela

Lista circular com nó sentinela:



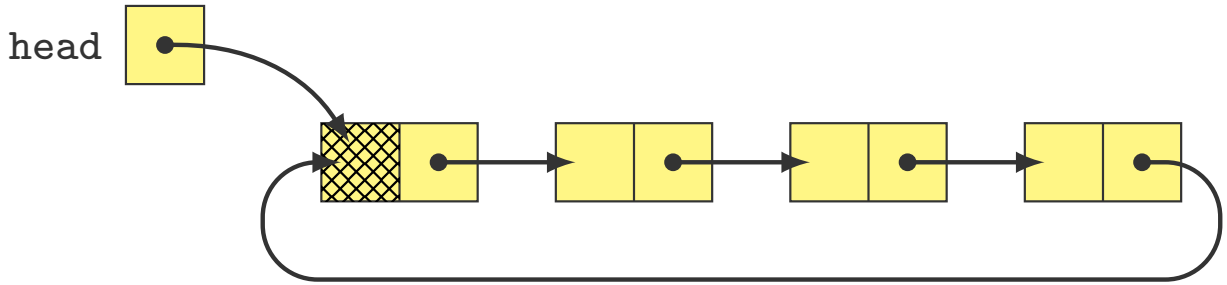

Lista circular vazia com nó sentinela:



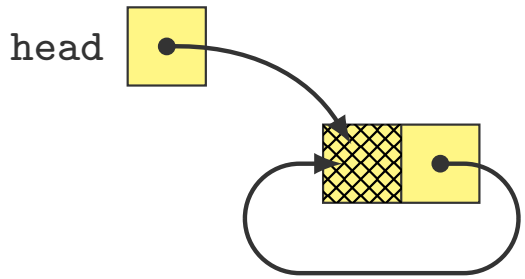
# Variações — Listas circulares com nó sentinela



Lista circular com nó sentinela:




Lista circular vazia com nó sentinela:

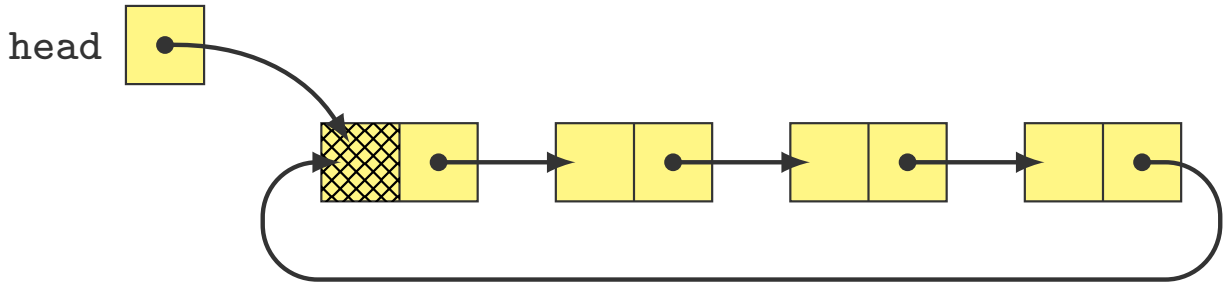



Diferenças para a versão sem nó sentinela:

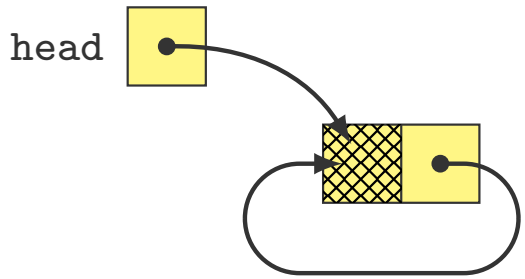
# Variações — Listas circulares com nó sentinela

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Lista circular com nó sentinela:




Lista circular vazia com nó sentinela:



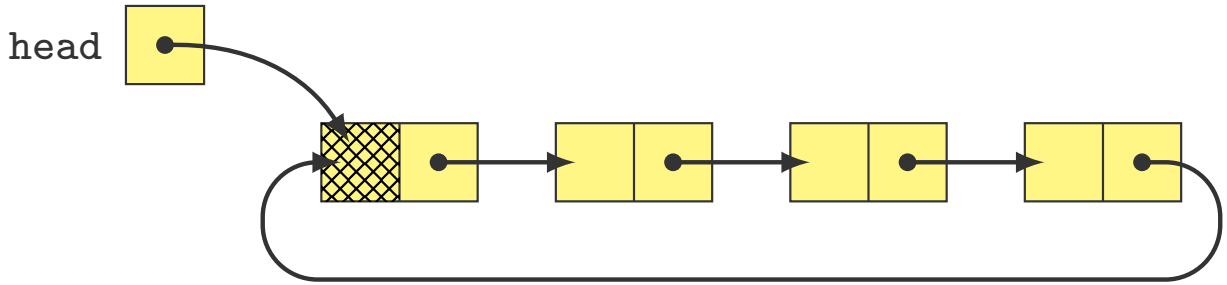

Diferenças para a versão sem nó sentinela:

- ponteiro `head` sempre aponta para nó sentinela

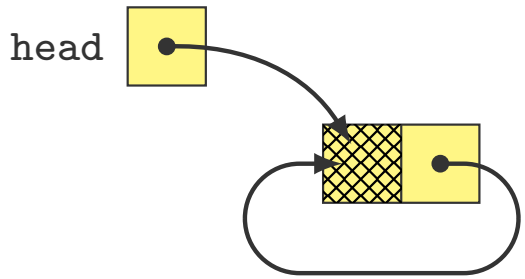
# Variações — Listas circulares com nó sentinela

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Lista circular com nó sentinela:




Lista circular vazia com nó sentinela:

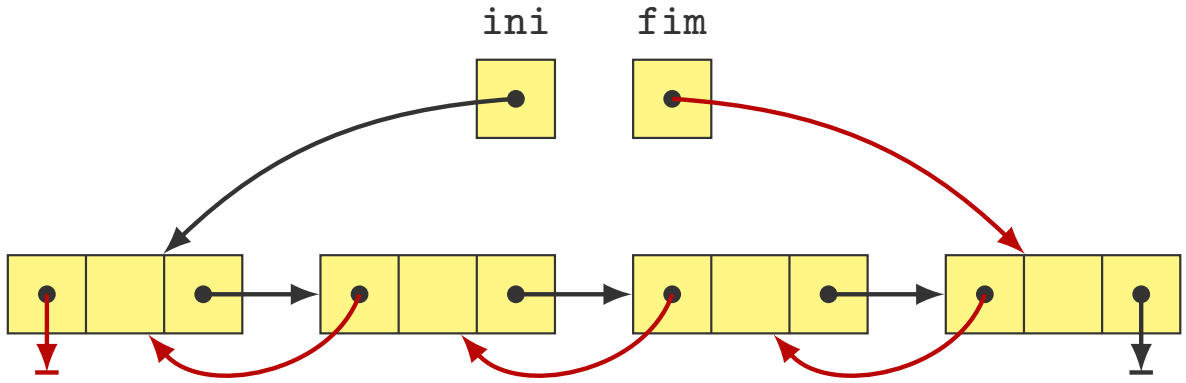



Diferenças para a versão sem nó sentinela:

- ponteiro `head` sempre aponta para nó sentinela
- código de inserção e de remoção mais simples

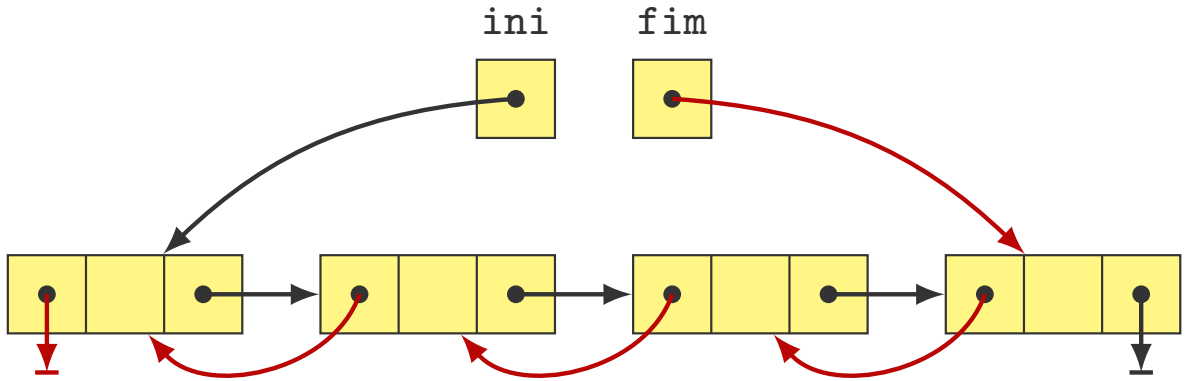
# Variações - Lista duplamente encadeada





# Variações - Lista duplamente encadeada

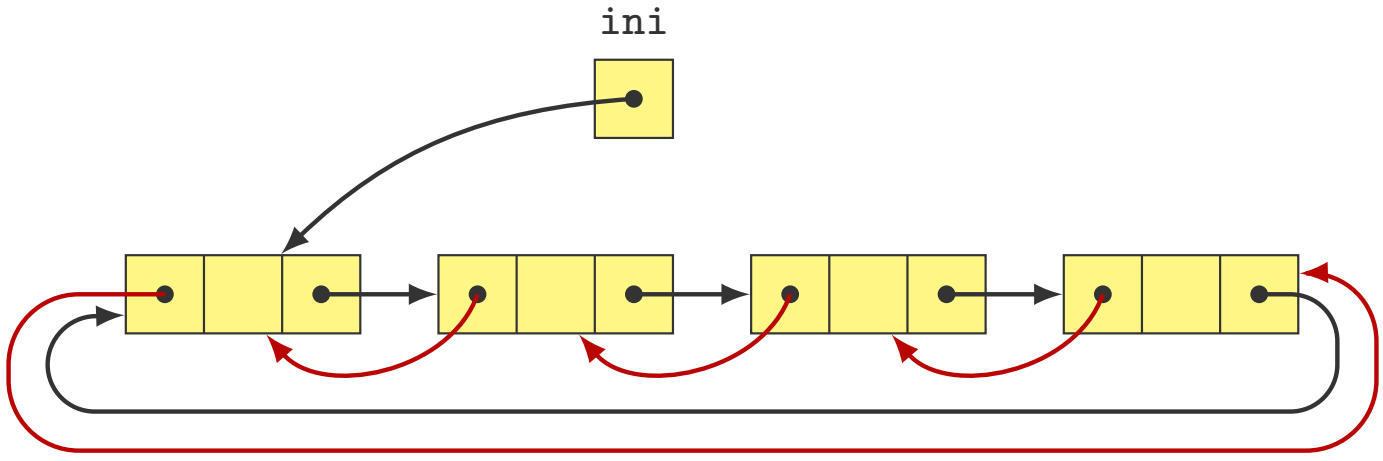
UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ




- Cada nó tem um ponteiro para o próximo nó e para o nó anterior.
- Se tivermos um ponteiro para o último elemento da lista, podemos percorrer a lista em ordem reversa.

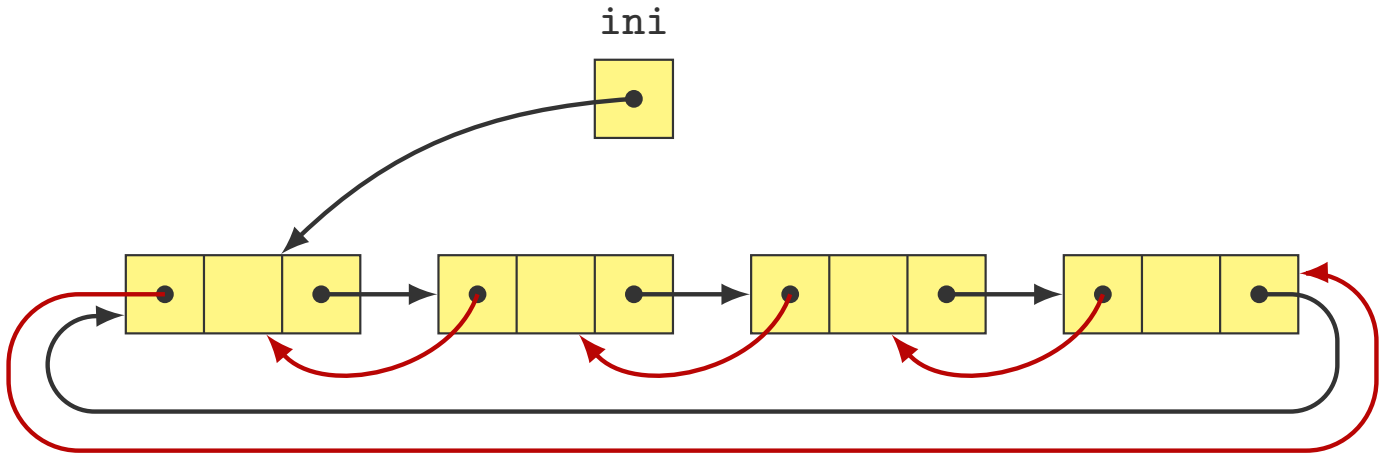
# Variações - Lista circular duplamente encadeada





# Variações - Lista circular duplamente encadeada

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ




Permite inserção e remoção em $O(1)$

Podemos ter uma lista dupla circular com cabeça também...

# Exercícios

# Exercícios



- Implemente uma lista duplamente encadeada com as operações:
  - inserir nó
  - remover nó
  - saber se há nó com dado valor
  - tamanho da lista
  - concatenar duas listas
  - imprimir lista de frente para trás ou reversamente

- Implemente uma lista circular duplamente encadeada com as operações:
  - inserir nó
  - remover nó
  - saber se há nó com dado valor
  - tamanho da lista
  - concatenar duas listas
  - imprimir lista de frente para trás ou reversamente

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

# FIM