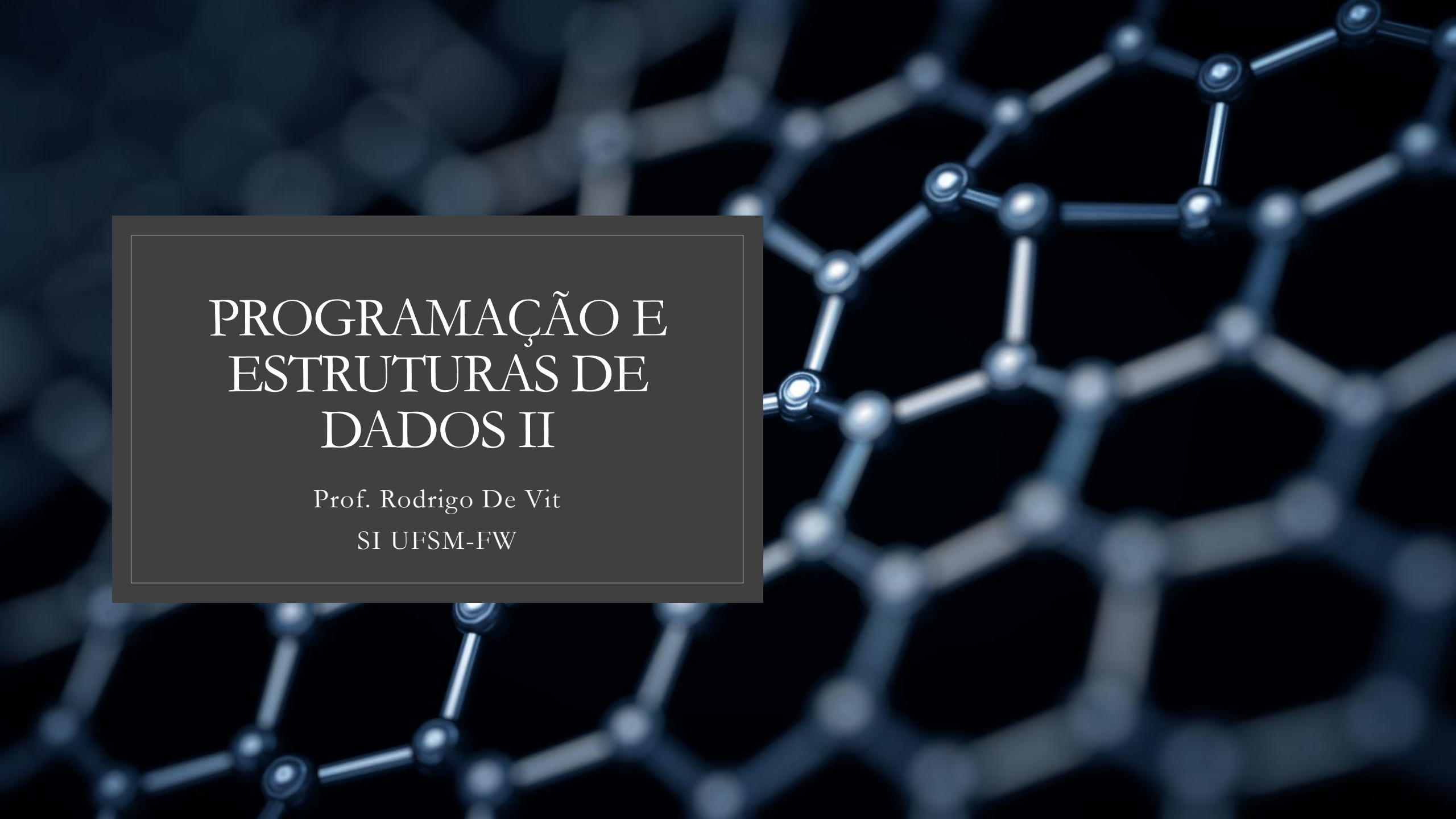

PROGRAMAÇÃO E ESTRUTURAS DE DADOS II

Prof. Rodrigo De Vit

SI UFSM-FW

Revisão

- **Aula (03), 20 e 21 de agosto e Aula (04), 27 e 28 de agosto:**
- UNIDADE 1 - ESTRUTURAS LINEARES E ENCADEADAS
- 1.2 - Estrutura Dinâmica- Listas Encadeadas.
- 1.2.1 - Conceituação e ponteiros.
- 1.2.2 - Listas lineares.
- 1.2.3 - Manipulação de pilhas e filas.
- 1.2.4 - Listas duplamente encadeadas.

ESTRUTURAS LINEARES E ENCADEADAS
Filas

- Na estrutura de fila, os acessos aos elementos também seguem uma regra. **O que diferencia a *fila* da *pilha* é a ordem de saída dos elementos: enquanto na pilha "o último que entra é o primeiro que sai", na fila "o primeiro que entra é o primeiro que sai" (a sigla FIFO – *first in, first out* – é usada para descrever essa estratégia).** A ideia fundamental da fila é que só podemos inserir um novo elemento no final da fila e só podemos retirar o elemento do início.
- Um exemplo de utilização em computação é a **implementação de uma fila de impressão.** Se uma impressora é compartilhada por várias máquinas, deve-se adotar uma estratégia para determinar que documento será impresso primeiro. A estratégia mais simples é tratar todas as requisições com a mesma prioridade e imprimir os documentos na ordem em que foram submetidos – o primeiro submetido é o primeiro a ser impresso.
- De modo análogo ao que fizemos com a estrutura de pilha, neste capítulo discutiremos duas estratégias para a implementação de uma estrutura de fila: usando vetor e usando lista encadeada. **Para implementar uma fila, devemos ser capazes de inserir novos elementos em uma extremidade, o *fim*, e retirar elementos da outra extremidade, o *início*.**

ESTRUTURAS LINEARES E ENCADEADAS
Filas - interface

- Antes de discutirmos as duas estratégias de implementação, podemos definir a interface disponibilizada pela estrutura, isto é, definir quais operações serão implementadas para manipular a fila. Mais uma vez, para simplificar a exposição, consideraremos um estrutura que armazena valores reais.
- O arquivo `fila.h`, que representa a interface do tipo, pode conter o seguinte código:

```
typedef struct fila Fila;

Fila* cria (void);
void insere (Fila* f, float v);
float retira (Fila* f);
int vazia (Fila* f);
void libera (Fila* f);
```

- **criar uma estrutura de fila;**
- **inserir um elemento no fim;**
- **retirar o elemento do início;**
- **verificar se a fila está vazia;**
- **liberar a fila.**

ESTRUTURAS LINEARES E ENCADEADAS

Filas - Implementação de fila com vetor

- Como no caso da pilha, nossa primeira implementação de fila será feita usando um **vetor para armazenar os elementos**. Para isso, **devemos fixar o número máximo N de elementos na fila**. Podemos observar que o processo de inserção e remoção em extremidades opostas fará com que a fila **"ande" no vetor**. Por exemplo, se inserirmos os elementos 1.4, 2.2, 3.5, 4.0 e depois retirarmos dois elementos, a fila não estará mais nas posições iniciais do vetor. A Figura 11.1 ilustra a configuração da fila após a inserção dos primeiros quatro elementos e a Figura 11.2 após a remoção de dois elementos.

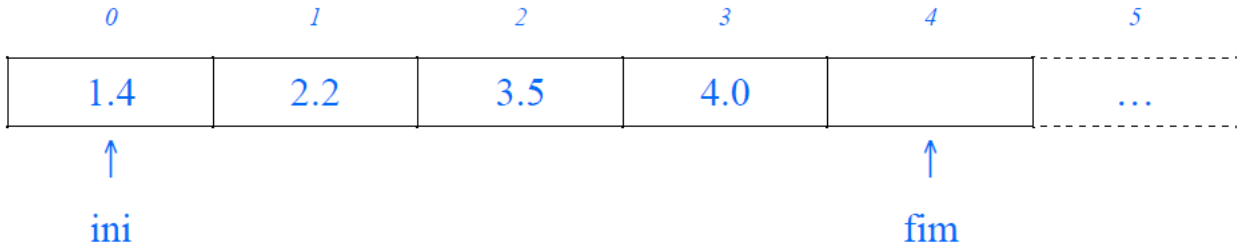

Figura 11.1: Fila após inserção de quatro novos elementos.

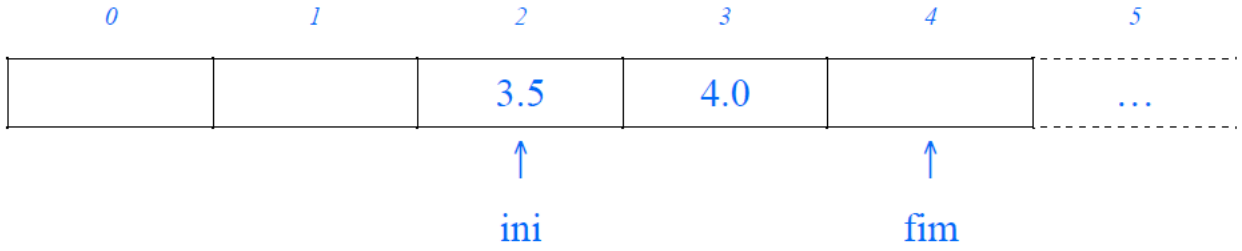

Figura 11.2: Fila após retirar dois elementos.

ESTRUTURAS LINEARES E ENCADEADAS

Filas - Implementação de fila com vetor – incremento circular

- Com essa estratégia, é fácil observar que, em um dado instante, a parte ocupada do vetor pode chegar à última posição. Para reaproveitar as primeiras posições livres do vetor sem implementarmos uma re-arrumação trabalhosa dos elementos, **podemos incrementar as posições do vetor de forma "circular": se o último elemento da fila ocupa a última posição do vetor, inserimos os novos elementos a partir do início do vetor.** Desta forma, em um dado momento, poderíamos ter quatro elementos, 20.0, 20.8, 21.2 e 24.3, distribuídos dois no fim do vetor e dois no início.

- Para essa implementação, os índices do vetor são incrementados de maneira que seus valores progridam "circularmente". Desta forma, se temos 100 posições no vetor, os valores dos índices assumem os seguintes valores:

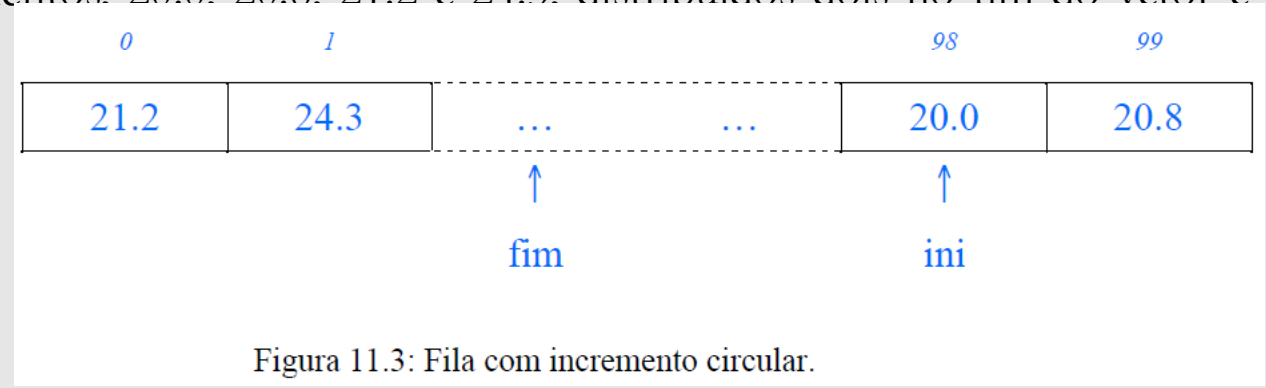

Figura 11.3: Fila com incremento circular.

0, 1, 2, 3, …, 98, 99, 0, 1, 2, 3, …, 98, 99, 0, 1,…

FilaV.c

ESTRUTURAS LINEARES E ENCADEADAS

Filas - Implementação de fila com lista

- Vamos agora ver como implementar uma fila através de uma lista encadeada, que será, como nos exemplos anteriores, uma lista simplesmente encadeada, em que cada nó guarda um ponteiro para o próximo nó da lista. Como teremos que inserir e retirar elementos das extremidades opostas da lista, que representarão o início e o fim da fila, teremos que usar dois ponteiros, `ini` e `fim`, que apontam respectivamente para o primeiro e para o último elemento da fila. Essa situação é ilustrada na figura ao lado:

- A operação para retirar um elemento se dá no início da lista (fila) e consiste essencialmente em fazer com que, após a remoção, ini aponte para o sucessor do nó retirado. (Observe que seria mais complicado remover um nó do fim da lista, porque o antecessor de um nó não é encontrado com a mesma facilidade que seu sucessor.) A inserção também é simples, pois basta acrescentar à lista um sucessor para o último nó, apontado por fim, e fazer com que fim aponte para este novo nó.

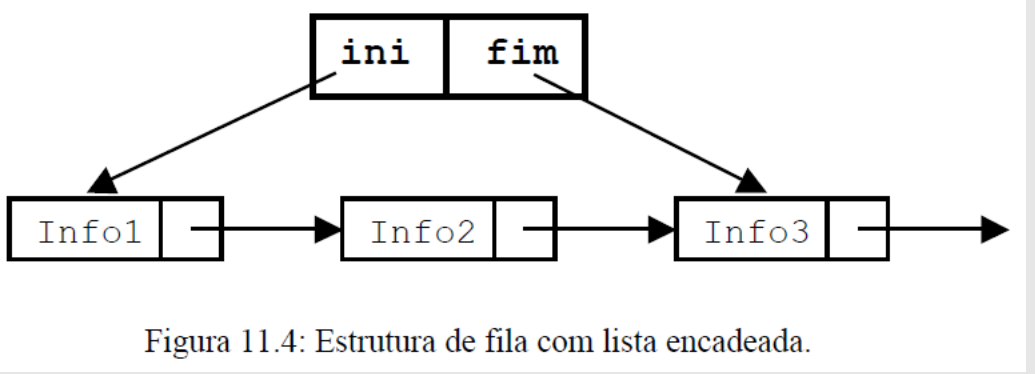

Figura 11.4: Estrutura de fila com lista encadeada.

FilaL.c

ESTRUTURAS LINEARES E ENCADEADAS
Fila dupla

- A estrutura de dados que chamamos de *fila dupla* consiste numa fila na qual é possível inserir novos elementos em ambas as extremidades, no início e no fim. Consequentemente, permite-se também retirar elementos de ambos os extremos. É como se, dentro de uma mesma estrutura de fila, tivéssemos duas filas, com os elementos dispostos em ordem inversa uma da outra.
- O arquivo `fila2.h`, que representa a interface do tipo, pode conter o seguinte código:

```
typedef struct fila2 Fila2;

Fila2* cria (void);
void insere_ini (Fila2* f, float v);
void insere_fim (Fila2* f, float v);
float retira_ini (Fila2* f);
float retira_fim (Fila2* f);
int vazia (Fila2* f);
void libera (Fila2* f);
```

- criar uma estrutura de fila dupla;
- inserir um elemento no início;
- inserir um elemento no fim;
- retirar o elemento do início;
- retirar o elemento do fim;
- verificar se a fila está vazia;
- liberar a fila.

ESTRUTURAS LINEARES E ENCADEADAS

Filas - Implementação de fila dupla com lista

- A implementação de uma fila dupla com lista encadeada merece uma discussão mais detalhada. A dificuldade que encontramos reside na implementação da função para retirar um elemento do final da lista. Todas as outras funções já foram discutidas e poderiam ser facilmente implementadas usando uma lista simplesmente encadeada. No entanto, na lista simplesmente encadeada, a função para retirar do fim não pode ser implementada de forma eficiente, pois, dado o ponteiro para o último elemento da lista, não temos como acessar o anterior, que passaria a ser o então último elemento.
- Para solucionar esse problema, temos que lançar mão da estrutura de lista duplamente encadeada. Nessa lista, cada nó guarda, além da referência para o próximo elemento, uma referência para o elemento anterior: dado o ponteiro de um nó, podemos acessar ambos os elementos adjacentes. Este arranjo resolve o problema de acessarmos o elemento anterior ao último. Devemos salientar que o uso de uma lista duplamente encadeada para implementar a fila é bastante simples, pois só manipulamos os elementos das extremidades da lista.

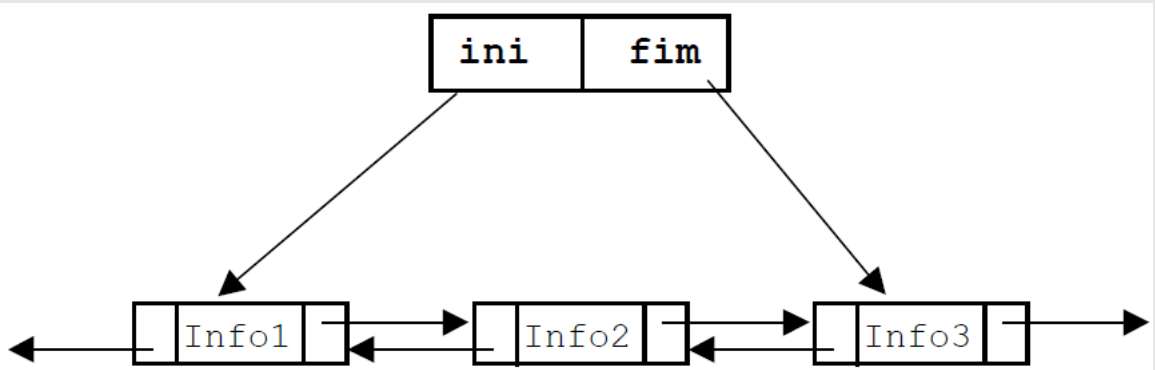

Figura 11.5: Arranjo da estrutura de fila dupla com lista.

FilaL2.c