



Estrutura de Dados - I

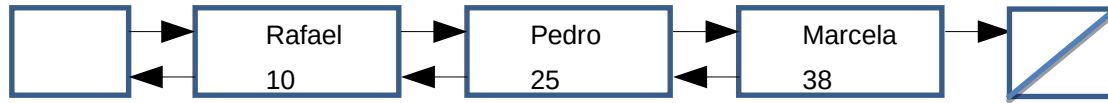
Listas Ligadas

Prof. MSc. Rafael Staiger Bressan
rafael.bressan@unicesumar.edu.br

Lista Duplamente Encadeada



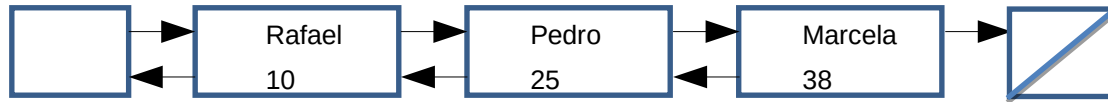
Lista Duplamente Encadeada Struct



```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<string.h>
4  struct cel{
5      char nome[100];
6      int idade;
7      struct cel *prox;
8      struct cel *ant;
9
10 };
11 typedef struct cel celula;
```

Lista Duplamente Encadeada

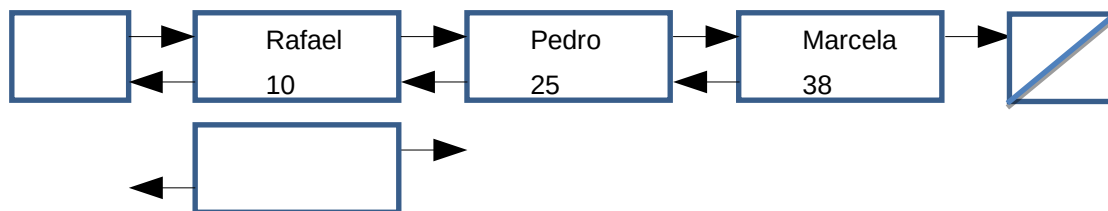
Inserir (Início)



```
--
13 // Insere elemento no começo da lista
14 void inserirInicio(char nome[], int idade, celula *ini) {
15     celula *nova; nova = malloc(sizeof(celula));
16     nova->idade = idade;
17     strcpy(nova->nome, nome);
18     if (ini->prox != NULL){
19         ini->prox->ant = nova;
20     }
21     nova->prox = ini->prox;
22     ini->prox = nova;
23     nova->ant = ini;
24 }
```

Lista Duplamente Encadeada

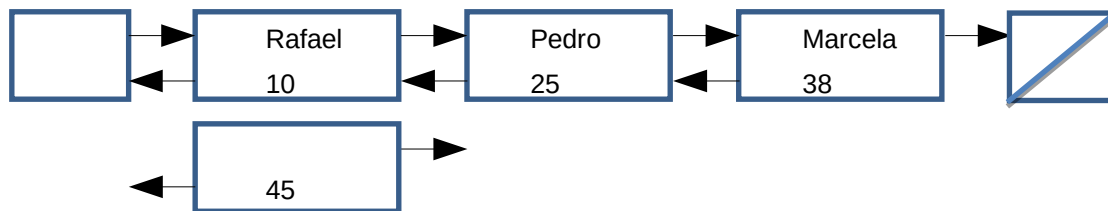
Inserir (Início)



```
13 // Insere elemento no começo da lista
14 void inserirInicio(char nome[], int idade, celula *ini) {
15     celula *nova; nova = malloc(sizeof(celula));
16     nova->idade = idade;
17     strcpy(nova->nome, nome);
18     if (ini->prox != NULL){
19         ini->prox->ant = nova;
20     }
21     nova->prox = ini->prox;
22     ini->prox = nova;
23     nova->ant = ini;
24 }
```

Lista Duplamente Encadeada

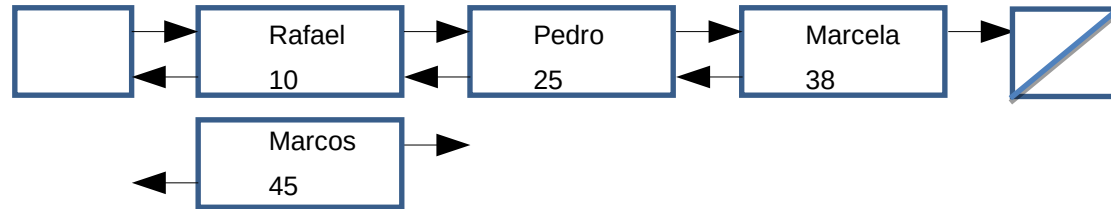
Inserir (Início)



```
13 // Insere elemento no começo da lista
14 void inserirInicio(char nome[], int idade, celula *ini) {
15     celula *nova; nova = malloc(sizeof(celula));
16     nova->idade = idade;
17     strcpy(nova->nome, nome);
18     if (ini->prox != NULL){
19         ini->prox->ant = nova;
20     }
21     nova->prox = ini->prox;
22     ini->prox = nova;
23     nova->ant = ini;
24 }
```

Lista Duplamente Encadeada

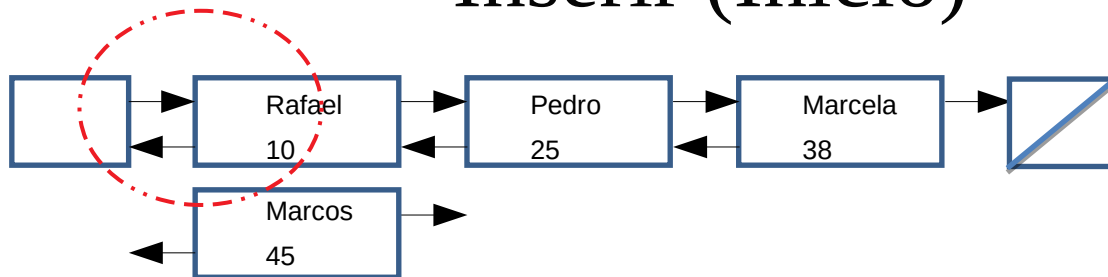
Inserir (Início)



```
13 // Insere elemento no começo da lista
14 void inserirInicio(char nome[], int idade, celula *ini) {
15     celula *nova; nova = malloc(sizeof(celula));
16     nova->idade = idade;
17     strcpy(nova->nome, nome);
18     if (ini->prox != NULL){
19         ini->prox->ant = nova;
20     }
21     nova->prox = ini->prox;
22     ini->prox = nova;
23     nova->ant = ini;
24 }
```

Lista Duplamente Encadeada

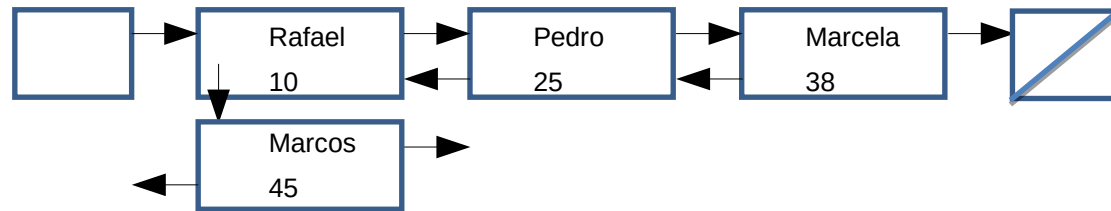
Inserir (Início)



```
13 // Insere elemento no começo da lista
14 void inserirInicio(char nome[], int idade, celula *ini) {
15     celula *nova; nova = malloc(sizeof(celula));
16     nova->idade = idade;
17     strcpy(nova->nome, nome);
18     if (ini->prox != NULL){
19         ini->prox->ant = nova;
20     }
21     nova->prox = ini->prox;
22     ini->prox = nova;
23     nova->ant = ini;
24 }
```


Lista Duplamente Encadeada

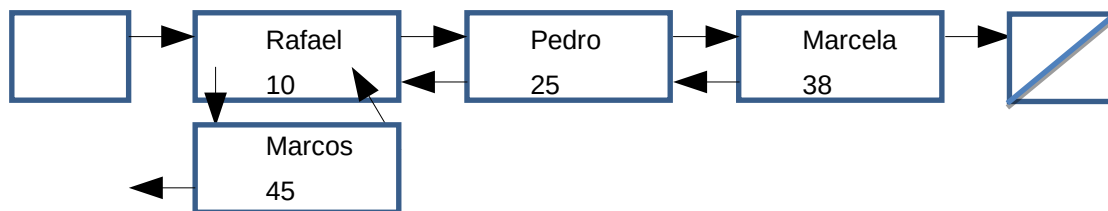
Inserir (Início)



```
--  
13 // Insere elemento no começo da lista  
14 void inserirInicio(char nome[], int idade, celula *ini) {  
15     celula *nova; nova = malloc(sizeof(celula));  
16     nova->idade = idade;  
17     strcpy(nova->nome, nome);  
18     if (ini->prox != NULL){  
19         ini->prox->ant = nova;  
20     }  
21     nova->prox = ini->prox;  
22     ini->prox = nova;  
23     nova->ant = ini;  
24 }
```

Lista Duplamente Encadeada

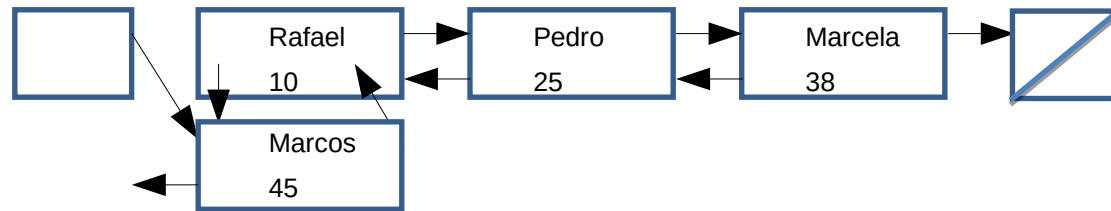
Inserir (Início)



```
13 // Insere elemento no começo da lista
14 void inserirInicio(char nome[], int idade, celula *ini) {
15     celula *nova; nova = malloc(sizeof(celula));
16     nova->idade = idade;
17     strcpy(nova->nome, nome);
18     if (ini->prox != NULL){
19         ini->prox->ant = nova;
20     }
21     nova->prox = ini->prox;
22     ini->prox = nova;
23     nova->ant = ini;
24 }
```

Lista Duplamente Encadeada

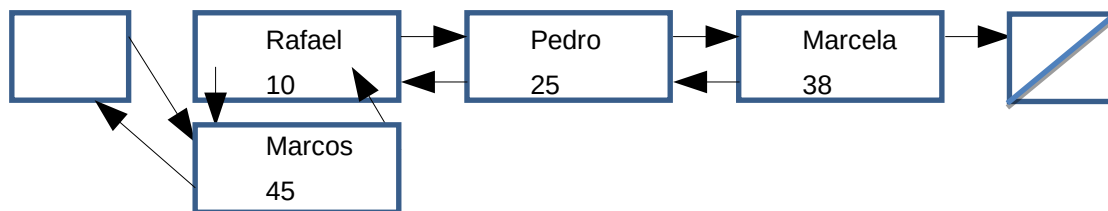
Inserir (Início)



```
--
13 // Insere elemento no começo da lista
14 void inserirInicio(char nome[], int idade, celula *ini) {
15     celula *nova; nova = malloc(sizeof(celula));
16     nova->idade = idade;
17     strcpy(nova->nome, nome);
18     if (ini->prox != NULL){
19         ini->prox->ant = nova;
20     }
21     nova->prox = ini->prox;
22     ini->prox = nova;
23     nova->ant = ini;
24 }
```

Lista Duplamente Encadeada

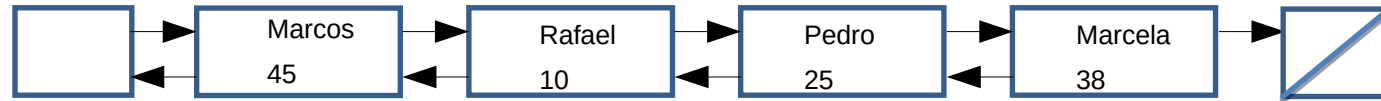
Inserir (Início)



```
--
13 // Insere elemento no começo da lista
14 void inserirInicio(char nome[], int idade, celula *ini) {
15     celula *nova; nova = malloc(sizeof(celula));
16     nova->idade = idade;
17     strcpy(nova->nome, nome);
18     if (ini->prox != NULL){
19         ini->prox->ant = nova;
20     }
21     nova->prox = ini->prox;
22     ini->prox = nova;
23     nova->ant = ini;
24 }
```

Lista Duplamente Encadeada

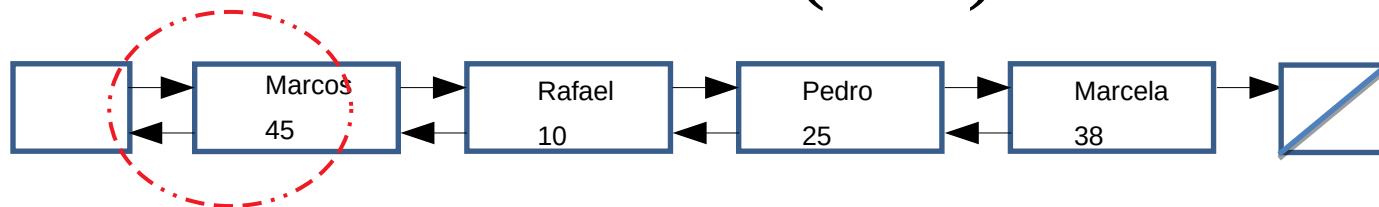
Inserir (Início)



```
--
13 // Insere elemento no começo da lista
14 void inserirInicio(char nome[], int idade, celula *ini) {
15     celula *nova; nova = malloc(sizeof(celula));
16     nova->idade = idade;
17     strcpy(nova->nome, nome);
18     if (ini->prox != NULL){
19         ini->prox->ant = nova;
20     }
21     nova->prox = ini->prox;
22     ini->prox = nova;
23     nova->ant = ini;
24 }
```

Lista Duplamente Encadeada

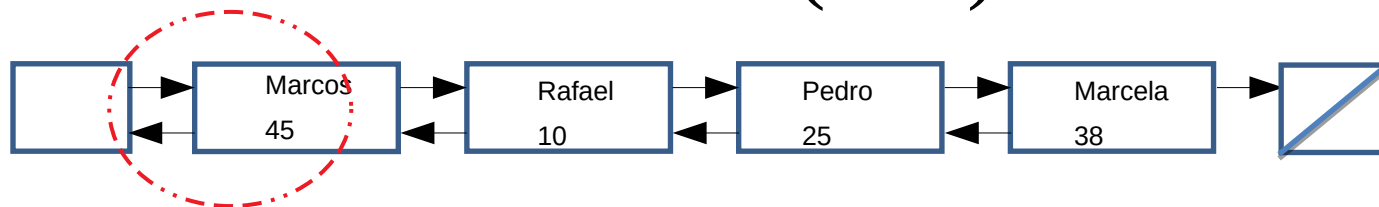
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```


Lista Duplamente Encadeada

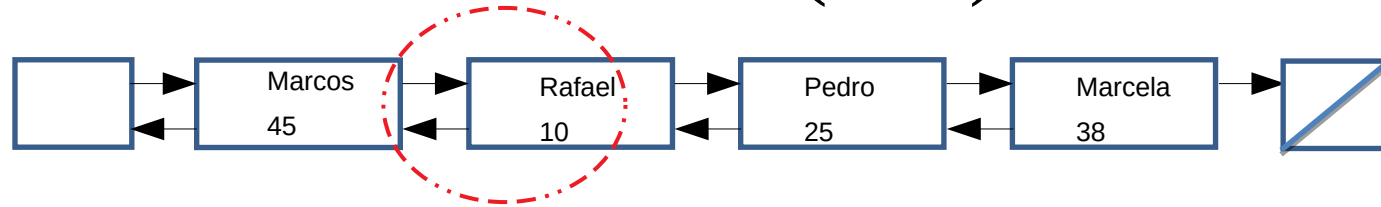
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```

Lista Duplamente Encadeada

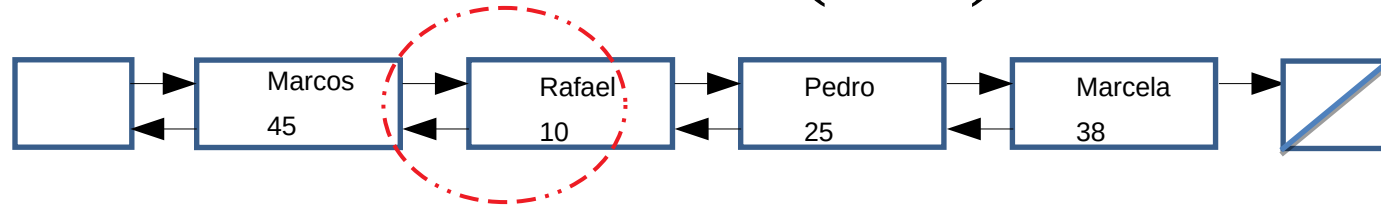
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```


Lista Duplamente Encadeada

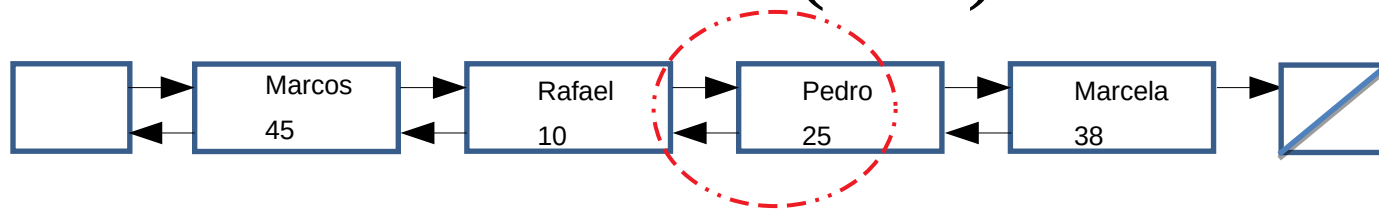
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```

Lista Duplamente Encadeada

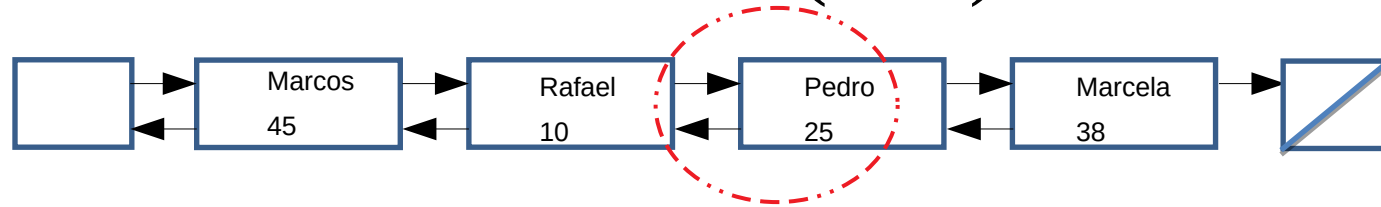
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```

Lista Duplamente Encadeada

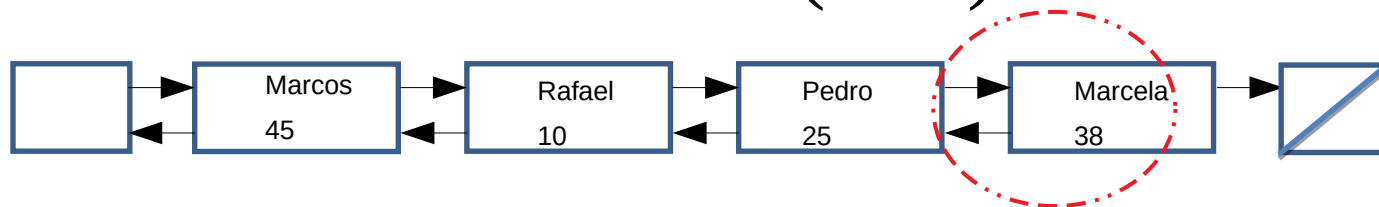
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```

Lista Duplamente Encadeada

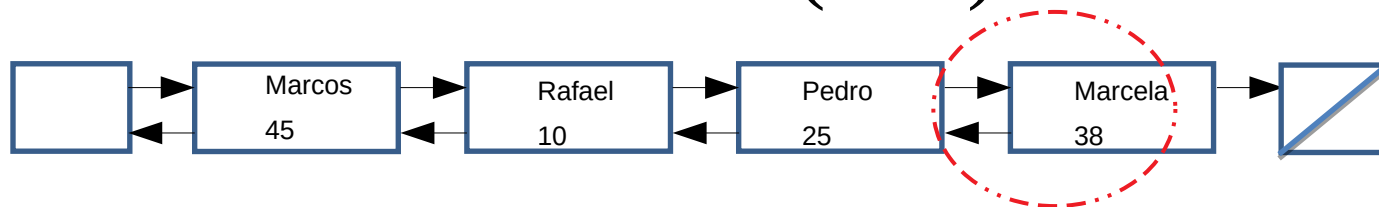
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```


Lista Duplamente Encadeada

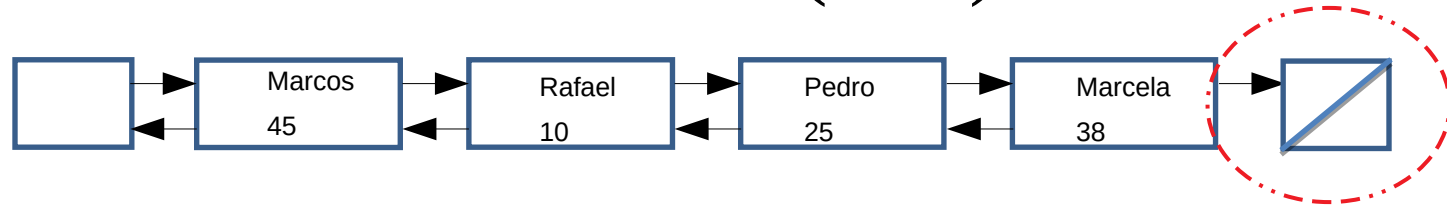
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```

Lista Duplamente Encadeada

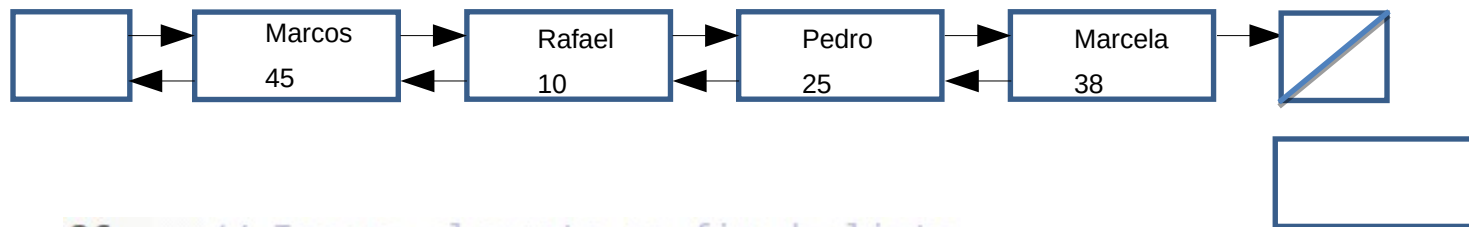
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```

Lista Duplamente Encadeada

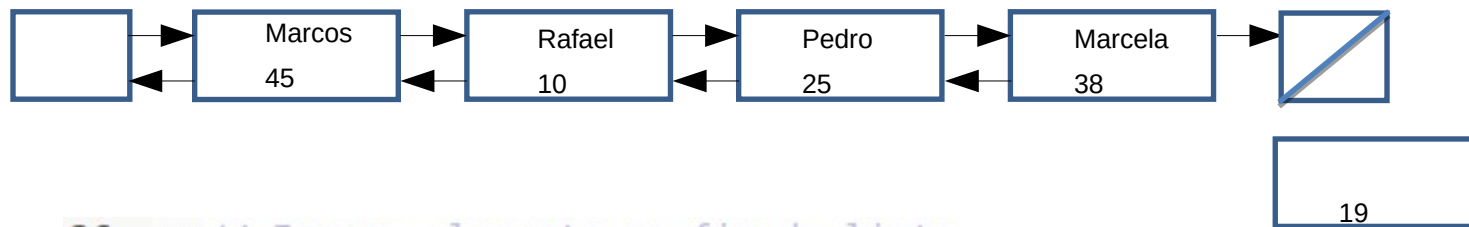
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```

Lista Duplamente Encadeada

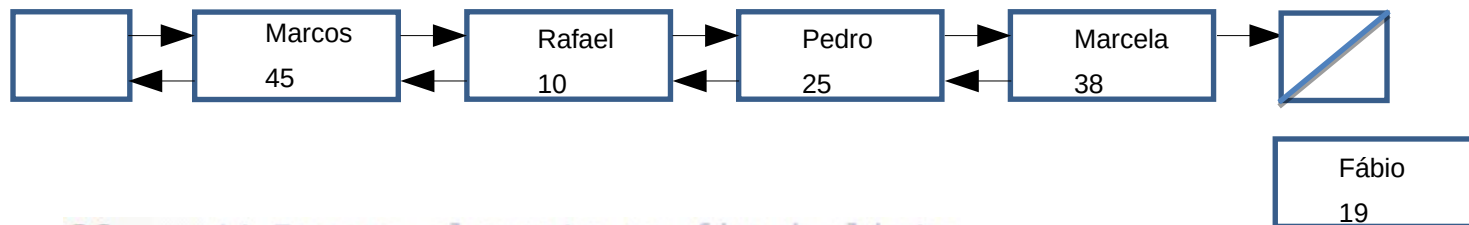
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```


Lista Duplamente Encadeada

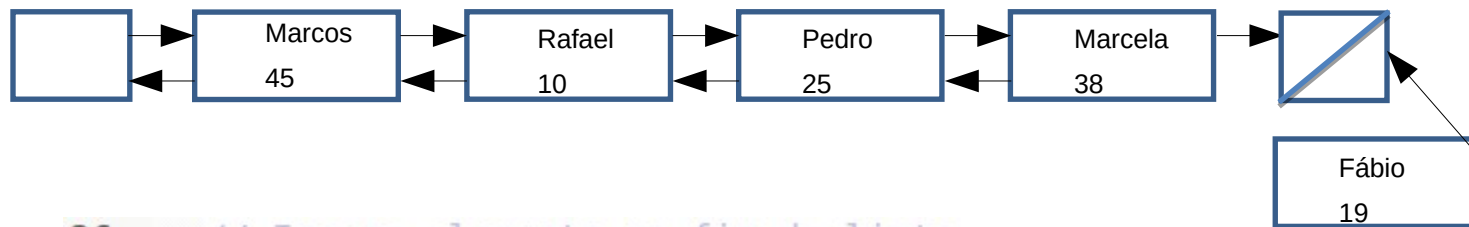
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```

Lista Duplamente Encadeada

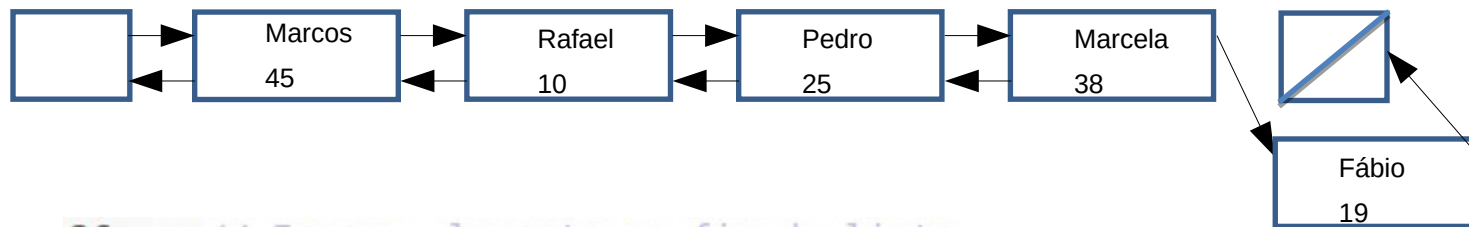
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```

Lista Duplamente Encadeada

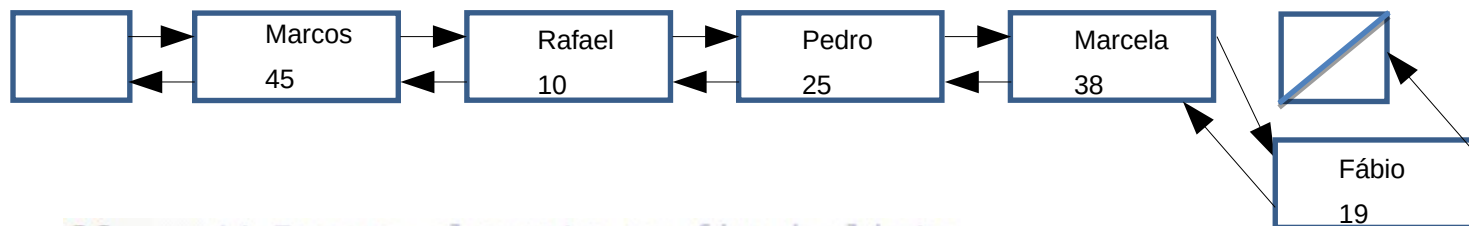
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```

Lista Duplamente Encadeada

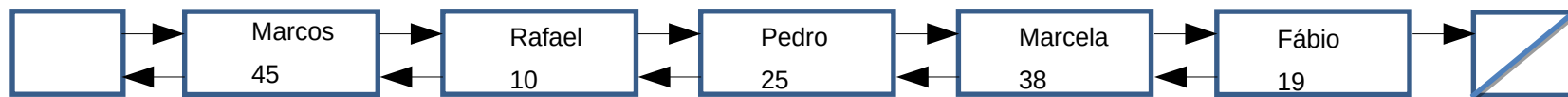
Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```


Lista Duplamente Encadeada

Inserir (Fim)



```
26 // Insere elemento no fim da lista
27 void inserirFim(char nome[], int idade, celula *ini) {
28     if (ini->prox == NULL){
29         celula *nova; nova = malloc(sizeof(celula));
30         nova->idade = idade;
31         strcpy(nova->nome, nome);
32         nova->prox = NULL;
33         ini->prox = nova;
34         nova->ant = ini;
35     }else{
36         inserirFim(nome, idade, ini->prox);
37     }
38 }
```



Atividade Prática

- Desenvolva um programa em C para manipulação de uma lista duplamente encadeada com os métodos apresentados no slide 27.
 - Inserção (Início | Fim | Meio)
 - Remoção (Início | Fim | Meio)
 - Percorrer (Exibir todos os elementos | Procurar um elemento)