

Lista Sequencial

Unidade II: Tipos Abstratos de Dados

Introdução

- Listas são um Tipo Abstrato de Dados (TAD) no qual podemos inserir e remover elementos em qualquer posição
- Uma lista sequencial basicamente tem os atributos *array* e contador; e métodos construtores, para inserção/remoção de elementos e para mostrar, pesquisar e ordenar
- *ArrayList* é a implementação C# da lista sequencial (ou linear)

Exercício Resolvido (1)

- Implemente uma classe lista sequencial sem usar o *ArrayList* e que permita a execução do código abaixo visto na Seção 1 da Unidade 1

```
Console.WriteLine("==== LISTA LINEAR ====");
Lista lista = new Lista(6);
int x1, x2, x3;
lista.InserirInicio(1);
lista.InserirFim(7);
lista.InserirFim(9);
lista.InserirInicio(3);
lista.Inserir(8, 3);
lista.Inserir(4, 2);

lista.Mostrar();

x1 = lista.RemoverInicio();
x2 = lista.RemoverFim();
x3 = lista.Remover(2);

Console.WriteLine(x1 + ", " + x2 + ", " + x3);
lista.Mostrar();
```

Exercício Resolvido (1)

- Implemente uma classe lista sequencial sem usar o *ArrayList* e que permita a execução do código abaixo visto na Seção 1 da Unidade 1

```
Console.WriteLine("==== LISTA LINEAR ====");  
Lista lista = new Lista(6);  
int x1, x2, x3;  
lista.InserirInicio(1);  
lista.InserirFim(7);  
lista.InserirFim(9);  
lista.InserirInicio(3);  
lista.Inserir(8, 3);  
lista.Inserir(4, 2);  
  
lista.Mostrar();  
  
x1 = lista.RemoverInicio();  
x2 = lista.RemoverFim();  
x3 = lista.Remover(2);  
  
Console.WriteLine(x1 + ", " + x2 + ", " + x3);  
lista.Mostrar();
```

Pause!

Exercício Resolvido (1)

```
class Lista {  
    int[] array;  
    int n;  
    Lista () {  
        array = new int[6];  
        n = 0;  
    }  
    Lista (int tamanho){  
        array = new int[tamanho];  
        n = 0;  
    }  
    void InserirInicio(int x) { ... }  
    void InserirFim(int x) { ... }  
    void Inserir(int x, int pos) { ... }  
    int RemoverInicio() { ... }  
    int RemoverFim() { ... }  
    int Remover(int pos) { ... }  
    void Mostrar() { ... }  
}
```

Exercício Resolvido (1)

```
class Lista {  
    int[] array;  
    int n;  
  
    Lista () {  
        array = new int[6];  
        n = 0;  
    }  
    Lista (int tamanho){  
        array = new int[tamanho];  
        n = 0;  
    }  
  
    void InserirInicio(int x) { ... }  
    void InserirFim(int x) { ... }  
    void Inserir(int x, int pos) { ... }  
    int RemoverInicio() { ... }  
    int RemoverFim() { ... }  
    int Remover(int pos) { ... }  
    void Mostrar() { ... }  
}
```

array

0	0	0	0	0	0
0	1	2	3	4	5

n

0

Exercício Resolvido (1)

```
class Lista {  
    int[] array;  
    int n;  
    Lista () {  
        array = new int[6];  
        n = 0;  
    }  
    Lista (int tamanho){  
        array = new int[tamanho];  
        n = 0;  
    }  
    void InserirInicio(int x) { ... }  
    void InserirFim(int x) { ... }  
    void Inserir(int x, int pos) { ... }  
    int RemoverInicio() { ... }  
    int RemoverFim() { ... }  
    int Remover(int pos) { ... }  
    void Mostrar() { ... }  
}
```

array

0	0	0	0	0	0
---	---	---	---	---	---

0 1 2 3 4 5

n

0

Exercício Resolvido (1)

// Exemplo: InserirInicio(1)

x

1

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

array

0

0

0

0

0

0

0

1

2

3

4

5

n

0

Exercício Resolvido (1)

```
// Exemplo: InserirInicio(1)
```

x

1

```
void InserirInicio(int x){
```

```
if (n >= array.Length)
```

```
    Environment.Exit(0);
```

```
//levar elementos para o fim do array
```

```
for (int i = n; i > 0; i--){
```

```
    array[i] = array[i-1];
```

```
}
```

```
array[0] = x;
```

```
n++;
```

```
}
```

false: 0 >= 6

array

0

0

0

0

0

0

0

1

2

3

4

5

n

0

Exercício Resolvido (1)

// Exemplo: InserirInicio(1)

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

x

1

i

0

array

0

0

0

0

0

0

0

1

2

3

4

5

n

0

Exercício Resolvido (1)

// Exemplo: InserirInicio(1)

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

x 1

i 0

false: 0 > 0

array

0	0	0	0	0	0
---	---	---	---	---	---

0 1 2 3 4 5

n

0

Exercício Resolvido (1)

// Exemplo: InserirInicio(1)

x

1

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

array

1

0

0

0

0

0

0

1

2

3

4

5

n

0

Exercício Resolvido (1)

// Exemplo: InserirInicio(1)

x

1

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

array

1

0

0

0

0

0

0

1

2

3

4

5

n

1

Exercício Resolvido (1)

// Exemplo: InserirInicio(1)

x

1

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
    n++;
```

```
}
```

array

1

0

0

0

0

0

0

1

2

3

4

5

n

1

Exercício Resolvido (1)

// Exemplo: InserirInicio(3)

x

3

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

array

1

0

0

0

0

0

0

1

2

3

4

5

n

1

Exercício Resolvido (1)

// Exemplo: InserirInicio(3)

x 3

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
```

```
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
```

```
        array[i] = array[i-1];
```

```
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

false: 1 >= 6

array

1	0	0	0	0	0
---	---	---	---	---	---

0 1 2 3 4 5

n

1

Exercício Resolvido (1)

// Exemplo: InserirInicio(3)

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

x

3

i

1

array

1

0

0

0

0

0

0

1

2

3

4

5

n

1

Exercício Resolvido (1)

// Exemplo: InserirInicio(3)

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
    n++;
```

```
}
```

x 3

i 1

true: 1 > 0

array

1	0	0	0	0	0
0	1	2	3	4	5

n 1

Exercício Resolvido (1)

```
// Exemplo: InserirInicio(3)
```

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
```

```
        array[i] = array[i-1];
```

```
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

x

3

i

1

array[1] <- array[0]

array

1

1

0

0

0

0

0

1

2

3

4

5

n

1

Exercício Resolvido (1)

// Exemplo: InserirInicio(3)

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

x

3

i

0

array

1

1

0

0

0

0

0

1

2

3

4

5

n

1

Exercício Resolvido (1)

// Exemplo: InserirInicio(3)

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
    n++;
```

```
}
```

x 3

i 0

false: $0 > 0$

array

1	1	0	0	0	0
0	1	2	3	4	5

n 1

Exercício Resolvido (1)

// Exemplo: InserirInicio(3)

x 3

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

array

3	1	0	0	0	0
---	---	---	---	---	---

0 1 2 3 4 5

n

1

Exercício Resolvido (1)

// Exemplo: InserirInicio(3)

x 3

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

array

3	1	0	0	0	0
0	1	2	3	4	5

n

2

Exercício Resolvido (1)

// Exemplo: InserirInicio(3)

x 3

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
    n++;
```

```
}
```

array

3	1	0	0	0	0
0	1	2	3	4	5

n

2

Exercício Resolvido (1)

// Exemplo: InserirInicio(5)

x

5

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

array

3

1

0

0

0

0

0

1

2

3

4

5

n

2

Exercício Resolvido (1)

// Exemplo: InserirInicio(5)

x 5

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
```

```
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
```

```
        array[i] = array[i-1];
```

```
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

false: 2 >= 6

array

3

1

0

0

0

0

0

1

2

3

4

5

n

2

Exercício Resolvido (1)

// Exemplo: InserirInicio(5)

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

x

5

i

2

array

3

1

0

0

0

0

0

1

2

3

4

5

n

2

Exercício Resolvido (1)

// Exemplo: InserirInicio(5)

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
    n++;
}
```

x

5

i

2

true: 2 > 0

array

3

1

0

0

0

0

0

1

2

3

4

5

n

2

Exercício Resolvido (1)

```
// Exemplo: InserirInicio(5)
```

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
```

```
        array[i] = array[i-1];
```

```
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

x

5

i

2

array[2] <- array[1]

array

3

1

1

0

0

0

0

1

2

3

4

5

n

2

Exercício Resolvido (1)

// Exemplo: InserirInicio(5)

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

x

5

i

1

array

3

1

1

0

0

0

0

1

2

3

4

5

n

2

Exercício Resolvido (1)

// Exemplo: InserirInicio(5)

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
    n++;
}
```

x

5

i

1

true: 1 > 0

array

3

1

1

0

0

0

0

1

2

3

4

5

n

2

Exercício Resolvido (1)

```
// Exemplo: InserirInicio(5)
```

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
```

```
        array[i] = array[i-1];
```

```
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

x

5

i

1

array[1] <- array[0]

array

3

3

1

0

0

0

0

1

2

3

4

5

n

2

Exercício Resolvido (1)

// Exemplo: InserirInicio(5)

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
    n++;
}
```

x

5

i

0

array

3

3

1

0

0

0

0

1

2

3

4

5

n

2

Exercício Resolvido (1)

// Exemplo: InserirInicio(5)

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
    n++;
```

```
}
```

x

5

i

0

false: $0 > 0$

array

3

3

1

0

0

0

0

1

2

3

4

5

n

2

Exercício Resolvido (1)

// Exemplo: InserirInicio(5)

x 5

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

array

5

3

1

0

0

0

0

1

2

3

4

5

n

2

Exercício Resolvido (1)

// Exemplo: InserirInicio(5)

x 5

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
```

```
    n++;
```

```
}
```

array

5	3	1	0	0	0
---	---	---	---	---	---

0 1 2 3 4 5

n

3

Exercício Resolvido (1)

// Exemplo: InserirInicio(5)

x 5

```
void InserirInicio(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > 0; i--){
        array[i] = array[i-1];
    }
```

```
    array[0] = x;
    n++;
```

```
}
```

array

5	3	1	0	0	0
---	---	---	---	---	---

0 1 2 3 4 5

n

3

Exercício Resolvido (1)

```
class Lista {  
    int[] array;  
    int n;  
    Lista () {  
        array = new int[6];  
        n = 0;  
    }  
    Lista (int tamanho){  
        array = new int[tamanho];  
        n = 0;  
    }  
    void InserirInicio(int x) { ... }  
    void InserirFim(int x) { ... }  
    void Inserir(int x, int pos) { ... }  
    int RemoverInicio() { ... }  
    int RemoverFim() { ... }  
    int Remover(int pos) { ... }  
    void Mostrar() { ... }  
}
```

array

5	3	1	0	0	0
0	1	2	3	4	5

n

3

Exercício Resolvido (1)

// Exemplo: InserirFim(9)

x

9

```
void InserirFim(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    array[n] = x;
    n++;
```

```
}
```

array

5

3

1

0

0

0

0

1

2

3

4

5

n

3

Exercício Resolvido (1)

```
// Exemplo: InserirFim(9)
```

x 9

```
void InserirFim(int x){
```

```
    if (n >= array.Length)
```

```
        Environment.Exit(0);
```

```
    array[n] = x;
```

```
    n++;
```

```
}
```

false: 3 >= 6

array

5

3

1

0

0

0

0

1

2

3

4

5

n

3

Exercício Resolvido (1)

// Exemplo: InserirFim(9)

x 9

```
void InserirFim(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    array[n] = x;
```

```
    n++;
```

```
}
```

array

5

3

1

9

0

0

0

1

2

3

4

5

n

3

Exercício Resolvido (1)

// Exemplo: InserirFim(9)

x 9

```
void InserirFim(int x){
```

```
    if (n >= array.Length)
        Environment.Exit(0);
```

```
    array[n] = x;
```

```
    n++;
```

```
}
```

array

5

3

1

9

0

0

0

1

2

3

4

5

n

4

Exercício Resolvido (1)

// Exemplo: InserirFim(9)

x 9

```
void InserirFim(int x){  
    if (n >= array.Length)  
        Environment.Exit(0);  
  
    array[n] = x;  
    n++;  
}
```

array

5	3	1	9	0	0
0	1	2	3	4	5

n

4

Exercício Resolvido (1)

```
class Lista {  
    int[] array;  
    int n;  
    Lista () {  
        array = new int[6];  
        n = 0;  
    }  
    Lista (int tamanho){  
        array = new int[tamanho];  
        n = 0;  
    }  
    void InserirInicio(int x) { ... }  
    void InserirFim(int x) { ... }  
    void Inserir(int x, int pos) { ... }  
    int RemoverInicio() { ... }  
    int RemoverFim() { ... }  
    int Remover(int pos) { ... }  
    void Mostrar() { ... }  
}
```

array

5	3	1	9	0	0
0	1	2	3	4	5

n

4

Exercício Resolvido (1)

// Exemplo: Inserir(4,2)

pos

2

x

4

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

array

5

3

1

9

0

0

0

1

2

3

4

5

n

4

Exercício Resolvido (1)

// Exemplo: Inserir(4,2)

pos

2

x

4

```
void Inserir(int x, int pos){
```

```
if (n >= array.Length || pos < 0 || pos > n)
```

```
    Environment.Exit(0);
```

```
//levar elementos para o fim do array
```

```
for (int i = n; i > pos; i--){
```

```
    array[i] = array[i-1];
```

```
}
```

```
array[pos] = x;
```

```
n++;
```

```
}
```

false: $4 \geq 6 \parallel 2 < 0 \parallel 2 > 4$

array

5

3

1

9

0

0

0

1

2

3

4

5

n

4

Exercício Resolvido (1)

// Exemplo: Inserir(4,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

pos

2

x

4

i

4

array

5

3

1

9

0

0

0

1

2

3

4

5

n

4

Exercício Resolvido (1)

// Exemplo: Inserir(4,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
    n++;
```

```
}
```

pos

2

x

4

i

4

true: 4 > 2

array

5

3

1

9

0

0

0

1

2

3

4

5

n

4

Exercício Resolvido (1)

// Exemplo: Inserir(4,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
```

```
        array[i] = array[i-1];
```

```
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

pos

2

x

4

i

4

array[4] <- array[3]

array

5

3

1

9

9

0

0

1

2

3

4

5

n

4

Exercício Resolvido (1)

// Exemplo: Inserir(4,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

pos

2

x

4

i

3

array

5

3

1

9

9

0

0

1

2

3

4

5

n

4

Exercício Resolvido (1)

// Exemplo: Inserir(4,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
    n++;
```

```
}
```

pos

2

x

4

i

3

true: 3 > 2

array

5

3

1

9

9

0

0

1

2

3

4

5

n

4

Exercício Resolvido (1)

// Exemplo: Inserir(4,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
```

```
        array[i] = array[i-1];
```

```
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

pos

2

x

4

i

3

array[3] <- array[2]

array

5

3

1

1

9

0

0

1

2

3

4

5

n

4

Exercício Resolvido (1)

// Exemplo: Inserir(4,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

pos

2

x

4

i

2

array

5

3

1

1

9

0

0

1

2

3

4

5

n

4

Exercício Resolvido (1)

// Exemplo: Inserir(4,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
    n++;
```

```
}
```

pos

2

x

4

i

2

false: 2 > 2

array

5

3

1

1

9

0

0

1

2

3

4

5

n

4

Exercício Resolvido (1)

// Exemplo: Inserir(4,2)

pos

2

x

4

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

array

5

3

4

1

9

0

0

1

2

3

4

5

n

4

Exercício Resolvido (1)

// Exemplo: Inserir(4,2)

pos

2

x

4

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

array

5

3

4

1

9

0

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(4,2)

pos

2

x

4

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
    n++;
```

```
}
```

array

5

3

4

1

9

0

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

pos

2

x

7

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
    n++;
```

```
}
```

array

5

3

4

1

9

0

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

pos

2

x

7

```
void Inserir(int x, int pos){
```

```
if (n >= array.Length || pos < 0 || pos > n)
```

```
    Environment.Exit(0);
```

```
//levar elementos para o fim do array
```

```
for (int i = n; i > pos; i--){
```

```
    array[i] = array[i-1];
```

```
}
```

```
array[pos] = x;
```

```
n++;
```

```
}
```

false: $5 \geq 6 \parallel 2 < 0 \parallel 0 > 4$

array

5

3

4

1

9

0

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

pos

2

x

7

i

5

array

5

3

4

1

9

0

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
    n++;
}
```

pos

2

x

7

i

5

true: 5 > 2

array

5

3

4

1

9

0

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

pos

2

x

7

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
```

```
        array[i] = array[i-1];
```

```
    }
```

```
    array[pos] = x;
    n++;
}
```

array[5] <- array[4]

i

5

array

5

3

4

1

9

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

pos

2

x

7

i

4

array

5

3

4

1

9

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
    n++;
```

```
}
```

pos

2

x

7

i

4

true: 4 > 2

array

5

3

4

1

9

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
```

```
        array[i] = array[i-1];
```

```
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

pos

2

x

7

i

4

array[4] <- array[3]

array

5

3

4

1

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

pos

2

x

7

i

3

array

5

3

4

1

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
    n++;
```

```
}
```

pos

2

x

7

i

3

true: 3 > 2

array

5

3

4

1

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

pos

2

x

7

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
```

```
        array[i] = array[i-1];
```

```
    }
```

array[3] <- array[2]

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

i

3

array

5

3

4

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

pos

2

x

7

i

2

array

5

3

4

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

//levar elementos para o fim do array

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
    n++;
```

```
}
```

pos

2

x

7

i

2

false: 2 > 2

array

5

3

4

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

pos

2

x

7

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

array

5

3

7

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

pos

2

x

7

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
```

```
    n++;
```

```
}
```

array

5

3

7

4

1

9

0

1

2

3

4

5

n

6

Exercício Resolvido (1)

// Exemplo: Inserir(7,2)

pos

2

x

7

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
    n++;
```

```
}
```

array

5

3

7

4

1

9

0

1

2

3

4

5

n

6

Exercício Resolvido (1)

// Exemplo: Inserir(2,2)

pos

2

x

2

```
void Inserir(int x, int pos){
```

```
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);
```

```
    //levar elementos para o fim do array
```

```
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
```

```
    array[pos] = x;
    n++;
```

```
}
```

array

5

3

7

4

1

9

0

1

2

3

4

5

n

6

Exercício Resolvido (1)

// Exemplo: Inserir(2,2)

pos

2

x

2

```
void Inserir(int x, int pos){
```

```
if (n >= array.Length || pos < 0 || pos > n)
```

```
    Environment.Exit(0);
```

```
//levar elementos para o fim do array
```

```
for (int i = n; i > pos; i--){
```

```
    array[i] = array[i-1];
```

```
}
```

```
array[pos] = x;
```

```
n++;
```

```
}
```

true: 6 >= 6 || ... || ...

array

5

3

7

4

1

9

0

1

2

3

4

5

n

6



Exercício Resolvido (1)

```
class Lista {  
    int[] array;  
    int n;  
    Lista () {  
        array = new int[6];  
        n = 0;  
    }  
    Lista (int tamanho){  
        array = new int[tamanho];  
        n = 0;  
    }  
    void InserirInicio(int x) { ... }  
    void InserirFim(int x) { ... }  
    void Inserir(int x, int pos) { ... }  
    int RemoverInicio() { ... }  
    int RemoverFim() { ... }  
    int Remover(int pos) { ... }  
    void Mostrar() { ... }  
}
```

array

5	3	7	4	1	9
0	1	2	3	4	5

n

6

Exercício Resolvido (1)

```
// Exemplo: RemoverInicio()
```

```
int RemoverInicio(){
```

```
    if (n == 0)
        Environment.Exit(0);
```

```
    int resp = array[0];
    n--;
```

```
    for (int i = 0; i < n; i++){
        array[i] = array[i+1];
    }
```

```
    return resp;
}
```

array

5	3	7	4	1	9
0	1	2	3	4	5

n

6

Exercício Resolvido (1)

```
// Exemplo: RemoverInicio()
```

```
int RemoverInicio(){
```

```
    if (n == 0)
```

```
        Environment.Exit(0);
```

```
    int resp = array[0];
```

```
    n--;
```

```
    for (int i = 0; i < n; i++){
```

```
        array[i] = array[i+1];
```

```
    }
```

```
    return resp;
```

```
}
```

false: 6 == 0

array

5

3

7

4

1

9

0

1

2

3

4

5

n

6

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

resp

5

```
int RemoverInicio(){
```

```
    if (n == 0)
```

```
        Environment.Exit(0);
```

```
    int resp = array[0];
```

```
    n--;
```

```
    for (int i = 0; i < n; i++){
```

```
        array[i] = array[i+1];
```

```
    }
```

```
    return resp;
```

```
}
```

array

5

3

7

4

1

9

0

1

2

3

4

5

n

6

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

resp

5

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

array

5

3

7

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

5

i

0

array

5

3

7

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

5

i

0

true: $0 < 5$

array

5

3

7

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){
```

```
    if (n == 0)
        Environment.Exit(0);
```

```
    int resp = array[0];
    n--;
```

```
    for (int i = 0; i < n; i++){
```

```
        array[i] = array[i+1];
```

```
    }
```

```
    return resp;
```

```
}
```

resp

5

i

0

array[0] <- array[1]

array

3

3

7

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

5

i

1

array

3

3

7

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

5

i

1

true: $1 < 5$

array

3

3

7

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){
```

```
    if (n == 0)
        Environment.Exit(0);
```

```
    int resp = array[0];
    n--;
```

```
    for (int i = 0; i < n; i++){
```

```
        array[i] = array[i+1];
```

```
    }
```

```
    return resp;
```

```
}
```

resp

5

i

1

array[1] <- array[2]

array

3

7

7

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

5

i

2

array

3

7

7

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

5

i

2

true: $2 < 5$

array

3

7

7

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){
```

```
    if (n == 0)
        Environment.Exit(0);
```

```
    int resp = array[0];
    n--;
```

```
    for (int i = 0; i < n; i++){
```

```
        array[i] = array[i+1];
```

```
    }
```

```
    return resp;
```

```
}
```

resp

5

i

2

array[2] <- array[3]

array

3

7

4

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

5

i

3

array

3

7

4

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

5

i

3

true: $3 < 5$

array

3

7

4

4

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){
```

```
    if (n == 0)
        Environment.Exit(0);
```

```
    int resp = array[0];
    n--;
```

```
    for (int i = 0; i < n; i++){
```

```
        array[i] = array[i+1];
```

```
    }
```

```
    return resp;
```

```
}
```

resp

5

i

3

array[3] <- array[4]

array

3

7

4

1

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

5

i

4

array

3

7

4

1

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

5

i

4

true: $4 < 5$

array

3

7

4

1

1

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){
```

```
    if (n == 0)
        Environment.Exit(0);
```

```
    int resp = array[0];
    n--;
```

```
    for (int i = 0; i < n; i++){
```

```
        array[i] = array[i+1];
```

```
    }
```

```
    return resp;
```

```
}
```

resp

5

i

4

array[4] <- array[5]

array

3

7

4

1

9

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

5

i

5

array

3

7

4

1

9

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

5

i

5

false: $5 < 5$

array

3

7

4

1

9

9

0

1

2

3

4

5

n

5

Exercício Resolvido (1)

```
// Exemplo: RemoverInicio()
```

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);
```

```
    int resp = array[0];  
    n--;
```

```
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }
```

```
    return resp;
```

```
}
```

resp

5

array

3

7

4

1

9

9

0

1

2

3

4

5

n

5

Retornando o 5

Exercício Resolvido (1)

// Exemplo: RemoverInicio()

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

5

array

3

7

4

1

9

9

0

1

2

3

4

5

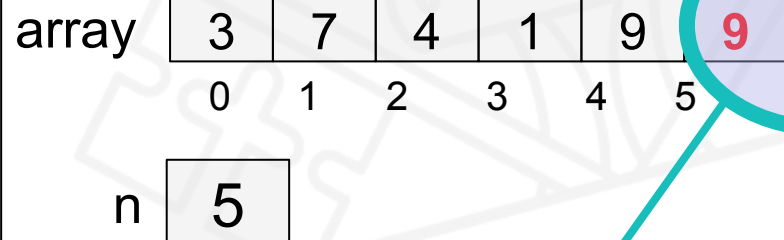
n

5

Exercício Resolvido (1)

```
// Exemplo: RemoverInicio()
```

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```



Este nove foi ou
não removido?

Exercício Resolvido (1)

```
// Exemplo: RemoverInicio()
```

```
int RemoverInicio(){
```

```
    if (n == 0)
```

```
        Environ
```

```
    int resp =
```

```
    n--;
```

```
    for (int i =
```

```
        array[i]
```

```
    }
```

```
    return res
```

```
}
```

array

3

7

4

1

9

9

Reflexões

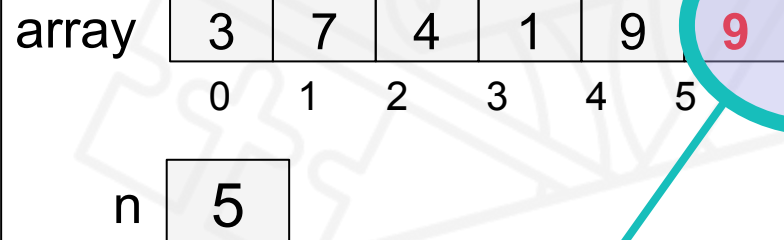
- 1) Diferencie remoção lógica e física
- 2) Diferencie formatação lógica (rápida) e física
- 3) O que acontece quando enviamos um arquivo para a lixeira? E quando o excluimos definitivamente?

nao removeido.

Exercício Resolvido (1)

```
// Exemplo: RemoverInicio()
```

```
int RemoverInicio(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    int resp = array[0];  
    n--;  
  
    for (int i = 0; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```




Este nove foi ou
não removido?

Exercício Resolvido (1)

```
class Lista {  
    int[] array;  
    int n;  
    Lista () {  
        array = new int[6];  
        n = 0;  
    }  
    Lista (int tamanho){  
        array = new int[tamanho];  
        n = 0;  
    }  
    void InserirInicio(int x) { ... }  
    void InserirFim(int x) { ... }  
    void Inserir(int x, int pos) { ... }  
    int RemoverInicio() { ... }  
    int RemoverFim() { ... }  
    int Remover(int pos) { ... }  
    void Mostrar() { ... }  
}
```

array

3	7	4	1	9	
---	---	---	---	---	---

0 1 2 3 4 5

n

5

Exercício Resolvido (1)

```
// Exemplo: RemoverFim()
```


```
int RemoverFim(){
```

```
    if (n == 0)  
        Environment.Exit(0);
```

```
    return array[--n];
```

```
}
```

array

3	7	4	1	9	
0	1	2	3	4	5

n

5

Exercício Resolvido (1)

```
// Exemplo: RemoverFim()
```

```
int RemoverFim(){
```

```
    if (n == 0)
```


```
        Environment.Exit(0);
```

```
    return array[--n];
```

```
}
```

false: $5 == 0$

array

3	7	4	1	9	
0	1	2	3	4	5

n

5


Exercício Resolvido (1)

```
// Exemplo: RemoverFim()
```

```
int RemoverFim(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    return array[--n];  
}
```

Primeiro o --, depois
o retorno

array

3	7	4	1	9	
0	1	2	3	4	5

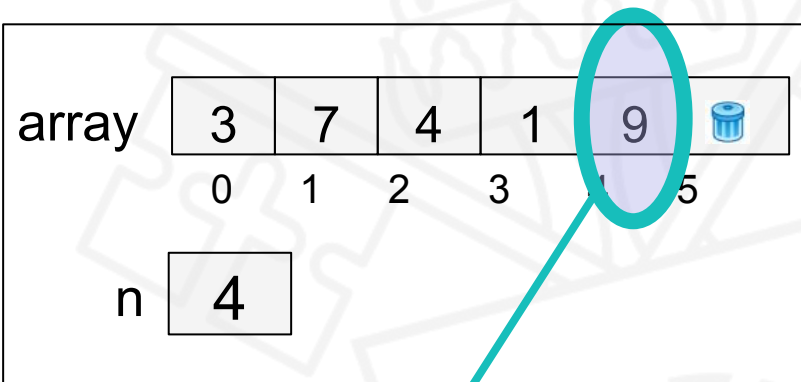
n

4

Exercício Resolvido (1)

```
// Exemplo: RemoverFim()
```

```
int RemoverFim(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    return array[--n];  
}
```





Retornando o 9

Exercício Resolvido (1)

```
// Exemplo: RemoverFim()
```

```
int RemoverFim(){  
    if (n == 0)  
        Environment.Exit(0);  
  
    return array[--n];  
}
```

array

3	7	4	1		
---	---	---	---	---	---

0 1 2 3 4 5

n

4

Exercício Resolvido (1)

```
class Lista {  
    int[] array;  
    int n;  
    Lista () {  
        array = new int[6];  
        n = 0;  
    }  
    Lista (int tamanho){  
        array = new int[tamanho];  
        n = 0;  
    }  
    void InserirInicio(int x) { ... }  
    void InserirFim(int x) { ... }  
    void Inserir(int x, int pos) { ... }  
    int RemoverInicio() { ... }  
    int RemoverFim() { ... }  
    int Remover(int pos) { ... }  
    void Mostrar() { ... }  
}
```

array

3	7	4	1		
---	---	---	---	---	---

0 1 2 3 4 5

n

4

Exercício Resolvido (1)

```
// Exemplo: Remover(2)
```

```
int Remover(int pos){
```

```
    if (n == 0 || pos < 0 || pos >= n)
        Environment.Exit(0);
```



```
    int resp = array[pos];
    n--;
```

```
    for (int i = pos; i < n; i++){
        array[i] = array[i+1];
    }
```

```
    return resp;
```

```
}
```

array

3	7	4	1		
0	1	2	3	4	5

n

4

Exercício Resolvido (1)

```
// Exemplo: Remover(2)
```

```
int Remover(int pos){
```

```
    if (n == 0 || pos < 0 || pos >= n)
```

```
        Environment.Exit(0);
```

```
    int resp = array[pos];
```

```
    n--;
```

```
    for (int i = pos; i < n; i++){
```

```
        array[i] = array[i+1];
```



```
    }
```

```
    return resp;
```

```
}
```

false: $4 == 0 \parallel 2 < 0 \parallel 2 \geq 4$

array

3	7	4	1		
---	---	---	---	---	---

0 1 2 3 4 5

n

4

Exercício Resolvido (1)

// Exemplo: Remover(2)

resp

4

```
int Remover(int pos){
```

```
    if (n == 0 || pos < 0 || pos >= n)
        Environment.Exit(0);
```

```
    int resp = array[pos];
```

```
    n--;
```

```
    for (int i = pos; i < n; i++){
        array[i] = array[i+1];
    }
```

```
    return resp;
```

```
}
```

array

3

7

4

1





0

1

2

3

4

5

n

4

Exercício Resolvido (1)

```
// Exemplo: Remover(2)
```

resp 4

```
int Remover(int pos){
```

```
    if (n == 0 || pos < 0 || pos >= n)
        Environment.Exit(0);
```

```
    int resp = array[pos];
```



```
    n--;
```

```
    for (int i = pos; i < n; i++){
        array[i] = array[i+1];
    }
```

```
    return resp;
```

```
}
```

array

3	7	4	1		
0	1	2	3	4	5

n

3

Exercício Resolvido (1)

// Exemplo: Remover(2)

```
int Remover(int pos){  
    if (n == 0 || pos < 0 || pos >= n)  
        Environment.Exit(0);  
  
    int resp = array[pos];  
    n--;  
  
    for (int i = pos; i < n; i++){  
        array[i] = array[i+1];  
    }  
  
    return resp;  
}
```

resp

4

i

2

array

3

7

4

1





0

1

2

3

4

5

n

3

Exercício Resolvido (1)

// Exemplo: Remover(2)

```
int Remover(int pos){
```

```
    if (n == 0 || pos < 0 || pos >= n)
        Environment.Exit(0);
```

```
    int resp = array[pos];
    n--;
```

```
    for (int i = pos; i < n; i++){
        array[i] = array[i+1];
    }
```

```
    return resp;
```

```
}
```

resp

4

i

2

true: 2 < 3

array

3

7

4

1





0

1

2

3

4

5

n

3

Exercício Resolvido (1)

// Exemplo: Remover(2)

```
int Remover(int pos){
```

```
    if (n == 0 || pos < 0 || pos >= n)
        Environment.Exit(0);
```

```
    int resp = array[pos];
    n--;
```

```
    for (int i = pos; i < n; i++){
        array[i] = array[i+1];
    }
```

```
    return resp;
```

```
}
```

resp

4

i

2

array[2] <- array[3]

array

3

7

1

1





0

1

2

3

4

5

n

3

Exercício Resolvido (1)

```
// Exemplo: Remover(2)
```

```
int Remover(int pos){
```

```
    if (n == 0 || pos < 0 || pos >= n)
        Environment.Exit(0);
```

```
    int resp = array[pos];
    n--;
```

```
    for (int i = pos; i < n; i++){
        array[i] = array[i+1];
    }
```

```
    return resp;
```

```
}
```

resp

4

i

3

array

3

7

1

1





0

1

2

3

4

5

n

3

Exercício Resolvido (1)

// Exemplo: Remover(2)

```
int Remover(int pos){
```

```
    if (n == 0 || pos < 0 || pos >= n)
        Environment.Exit(0);
```

```
    int resp = array[pos];
    n--;
```

```
    for (int i = pos; i < n; i++){
        array[i] = array[i+1];
    }
```

```
    return resp;
```

```
}
```

resp

4

i

3

false: 3 < 3

array

3

7

1

1





0

1

2

3

4

5

n

3

Exercício Resolvido (1)

```
// Exemplo: Remover(2)
```

```
int Remover(int pos){
```

```
    if (n == 0 || pos < 0 || pos >= n)
        Environment.Exit(0);
```

```
    int resp = array[pos];
    n--;
```

```
    for (int i = pos; i < n; i++){
        array[i] = array[i+1];
    }
```

```
    return resp;
```

resp

4

array

3

7

1

1





0

1

2

3

4

5

n

3

Retornando o 4

Exercício Resolvido (1)

// Exemplo: Remover(2)

```
int Remover(int pos){
```

```
    if (n == 0 || pos < 0 || pos >= n)
        Environment.Exit(0);
```

```
    int resp = array[pos];
    n--;
```

```
    for (int i = pos; i < n; i++){
        array[i] = array[i+1];
    }
```

```
    return resp;
```

```
}
```

resp

4

array

3

7

1



0

1

2

3

4

5

n

3

Exercício Resolvido (1)

```
class Lista {  
    int[] array;  
    int n;  
    Lista () {  
        array = new int[6];  
        n = 0;  
    }  
    Lista (int tamanho){  
        array = new int[tamanho];  
        n = 0;  
    }  
    void InserirInicio(int x) { ... }  
    void InserirFim(int x) { ... }  
    void Inserir(int x, int pos) { ... }  
    int RemoverInicio() { ... }  
    int RemoverFim() { ... }  
    int Remover(int pos) { ... }  
    void Mostrar() { ... }  
}
```

Exercício Resolvido (1)

```
void Mostrar(){  
    Console.Write("[ ");  
    for (int i = 0; i < n; i++){  
        Console.Write(array[i] + " ");  
    }  
    Console.WriteLine("]");  
}
```

Exercício: O que será mostrado na tela?

array

3	7	1			
0	1	2	3	4	5

n 3

Tela