

Programação Multimédia

Iteracção

Conceitos Básicos de Um Programa

- Memória (armazenamento de dados)
 - Guardar dados temporários (durante a execução do programa)
- Selecção (...de caminhos de execução)
 - O nosso programa pode ter ramos que são executados em determinadas circunstâncias
- **Iteracção (execução repetida das instruções)**
 - **Executar várias vezes a mesma acção (sobre dados diferentes)**
- Módulos

Vectores

- Aumentar e diminuir a dimensão de um vector

Problema

- Criar um programa que anime linhas no ecrã. De cada vez que um ponto da linha “bate” na borda do ecrã, outra linha aparece.
- O número de linhas aumenta!
 - Como dimensionamos o vector?

Exemplo #1

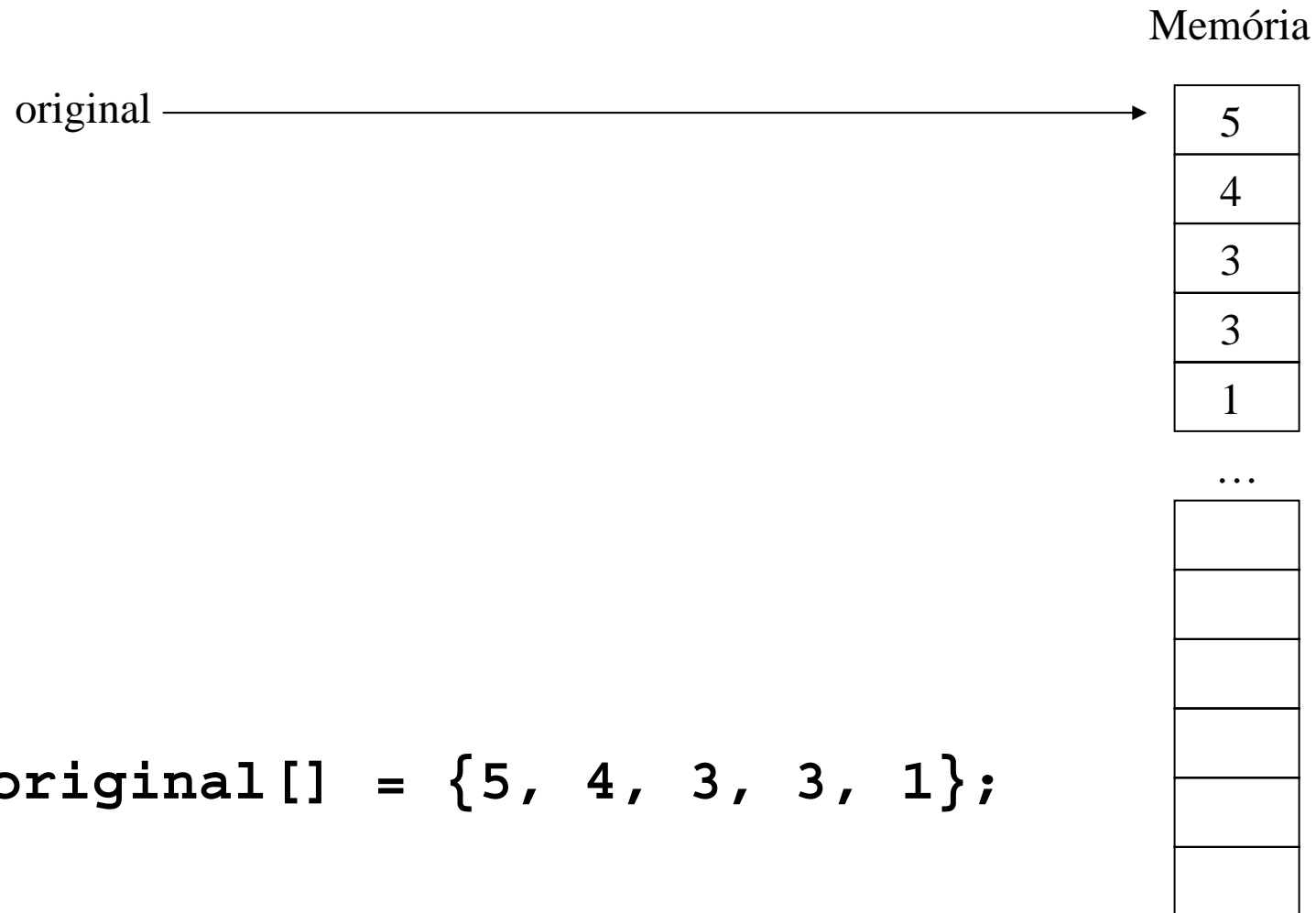
Aumentar um Vector

- Aumentar o tamanho de um vector
 - Como aumentar o tamanho de um vector mantendo o conteúdo?
 - `[45, 12, 43, 10, 2]` para `[45, 12, 43, 10, 2, 0]`
 - `int original[] = {45, 12, 43, 10, 2}; // 5 elementos`
- Não podemos fazer
 - `original = new int [6];`
 - Porquê?

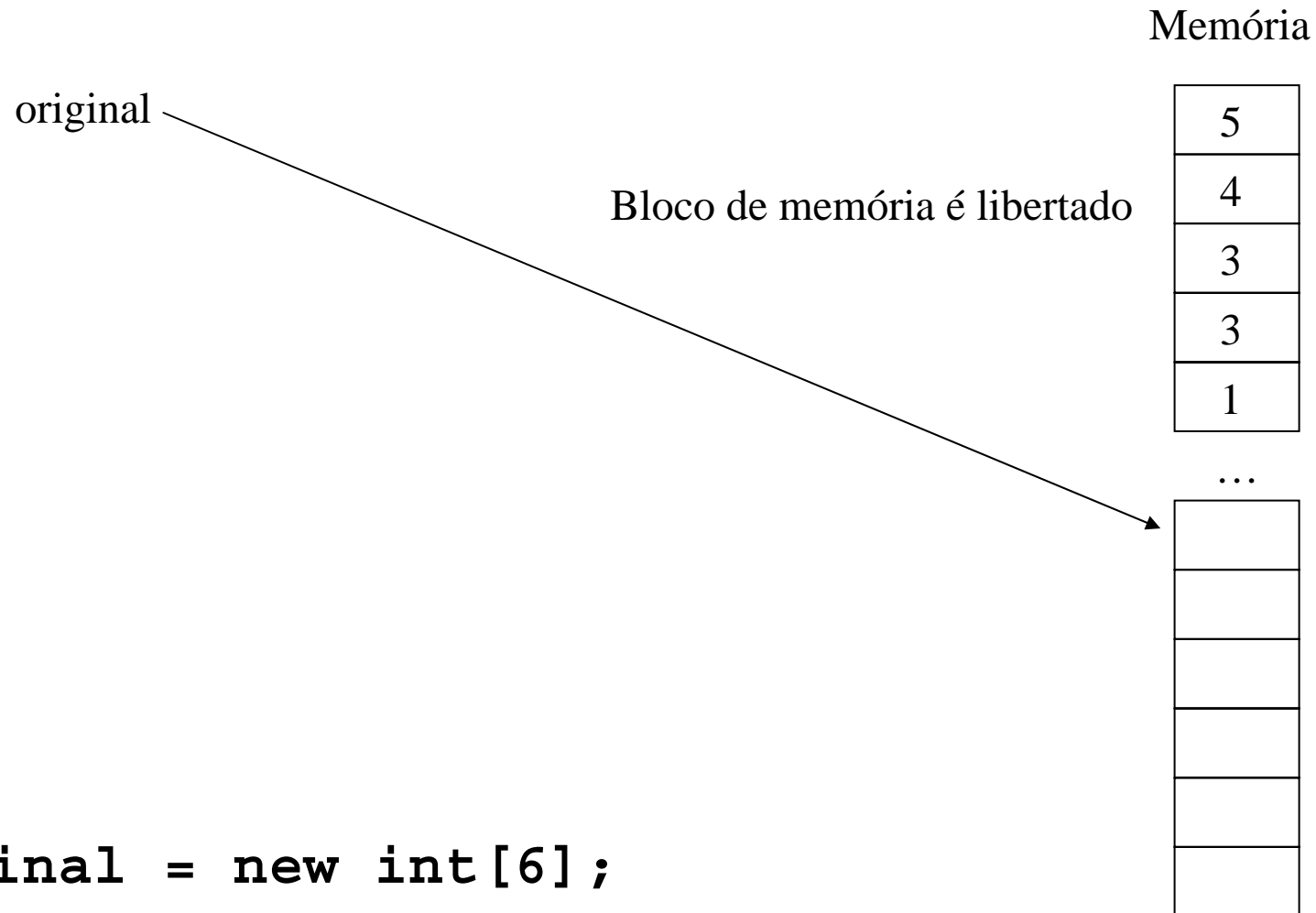
Aumentar um Vector

- Ao fazer
 - `new int[6];`
- Estamos a reservar um novo bloco de memória
- Os elementos anteriores perdem-se...

Aumentar um Vector



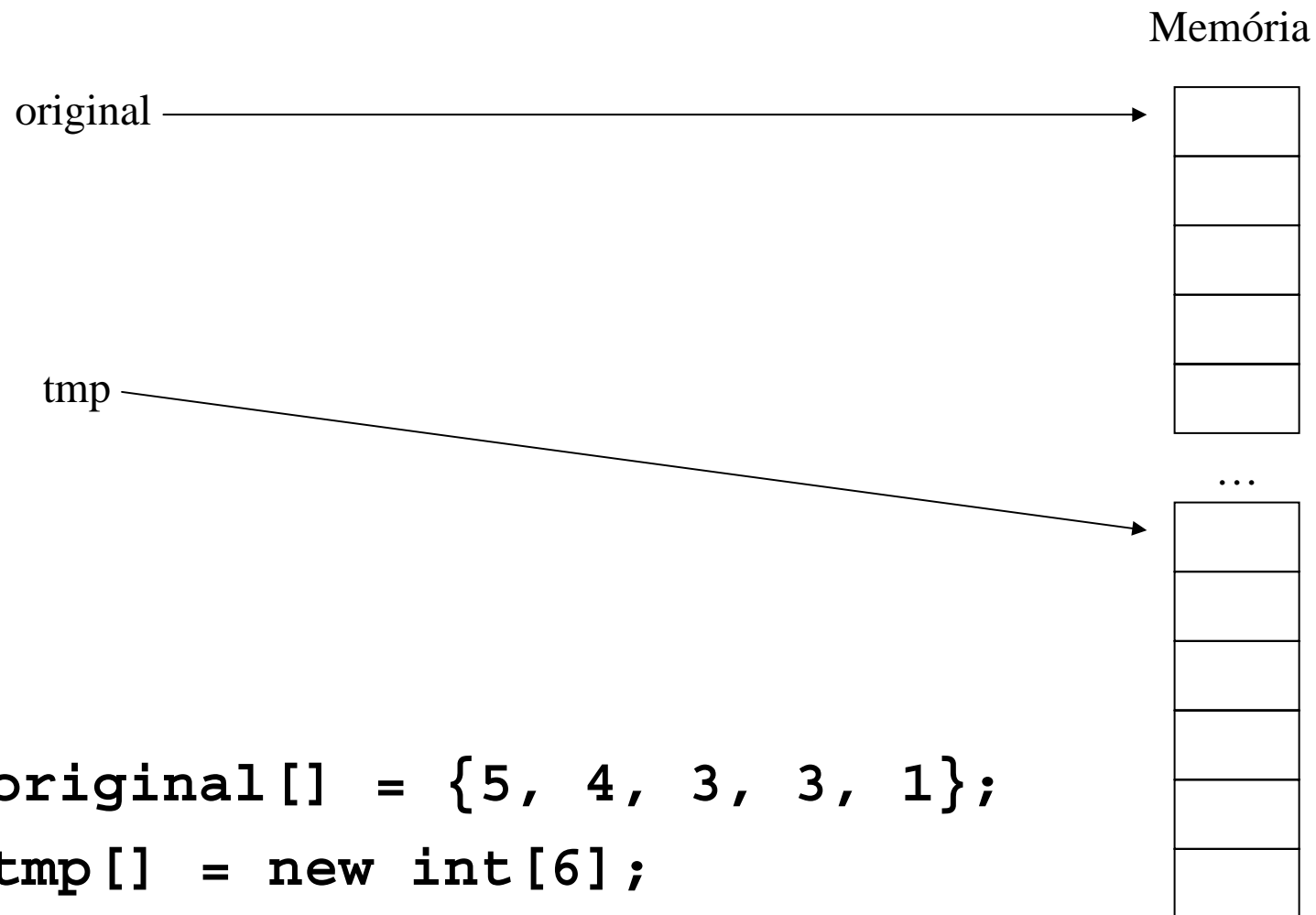
Aumentar um Vector



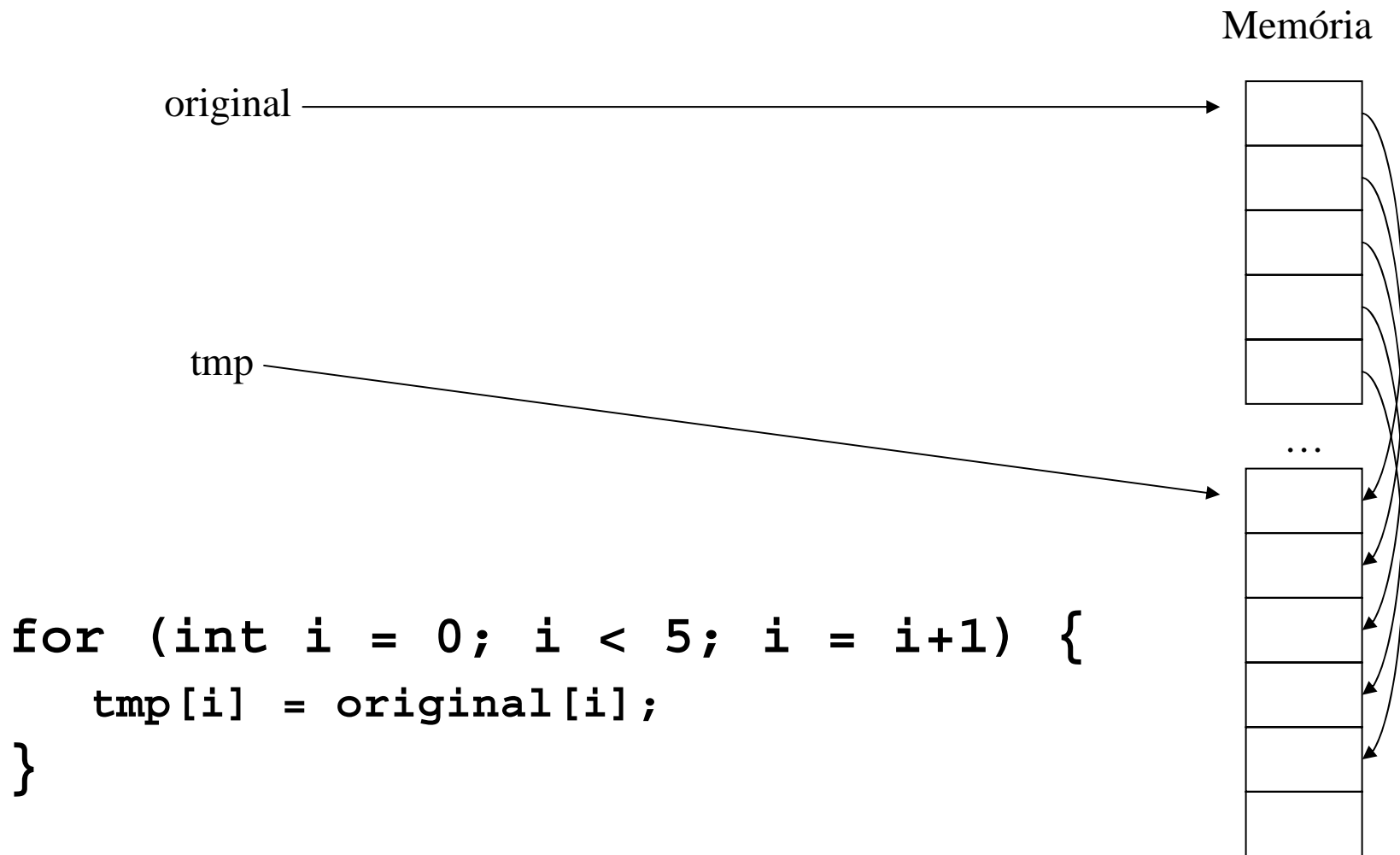
Aumentar um Vector

- Solução
 - Criar um novo vector com 6 elementos
 - Copiar o vector antigo para as primeiras posições do novo
 - Atribuir o novo vector ao antigo

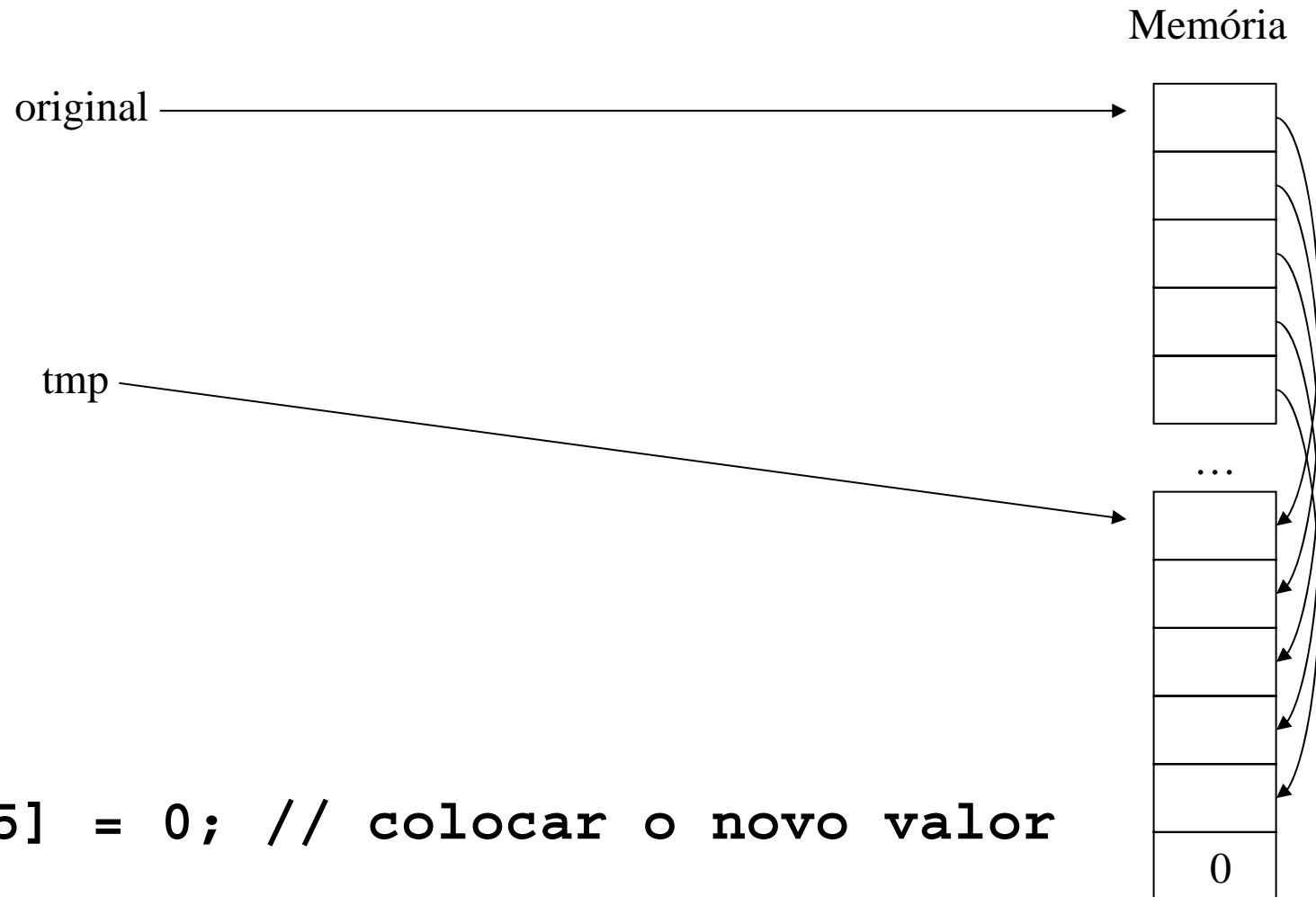
Aumentar um Vector: passo #1



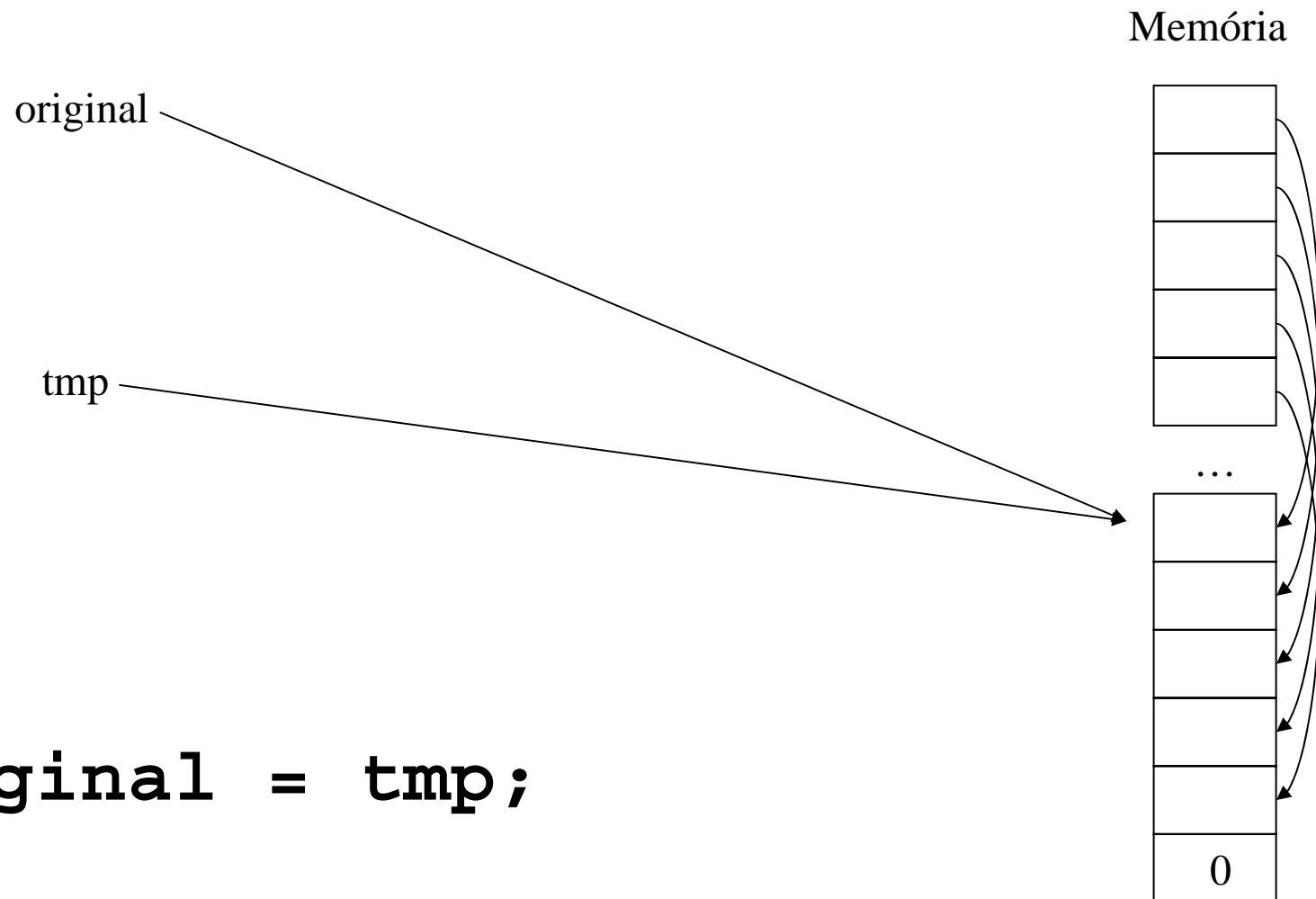
Aumentar um Vector: passo #2



Aumentar um Vector: passo #3



Aumentar um Vector: passo #4



Aumentar um Vector

```
int original[] = {5, 4, 3, 3, 1};
int tmp[] = new int[6];
int i;
for (i = 0; i < 5; i = i+1) {
    tmp[i] = original[i];
}
tmp[5] = 0; // colocar o novo valor
original = tmp;
```

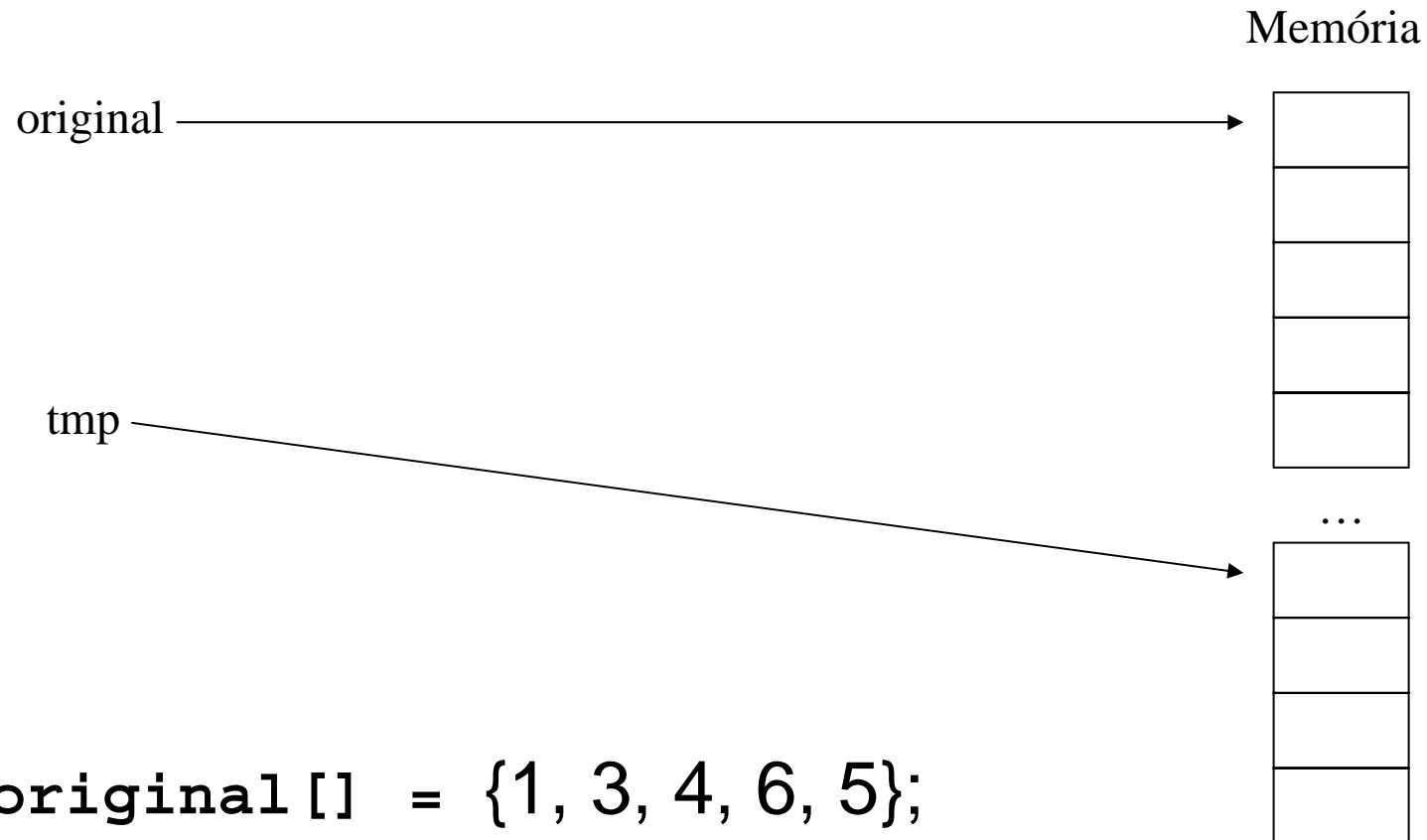
Reduzir um Vector

- Reduzir o tamanho de um vector
 - Eliminar um determinado elemento (representado pelo índice)
- Exemplo: Eliminar o elemento no índice 2 (valor 4) do vector
 - `int original[] = {1, 3, 4, 6, 5};`
- Ou seja, ficar com
 - `[1, 3, 6, 5]`

Reduzir um Vector

- Solução
 - Criar um novo vector com tamanho inferior ao original
 - Percorrer o vector original e copiar todos os elementos para o novo excepto o elemento a eliminar
 - Atribuir o novo vector ao antigo

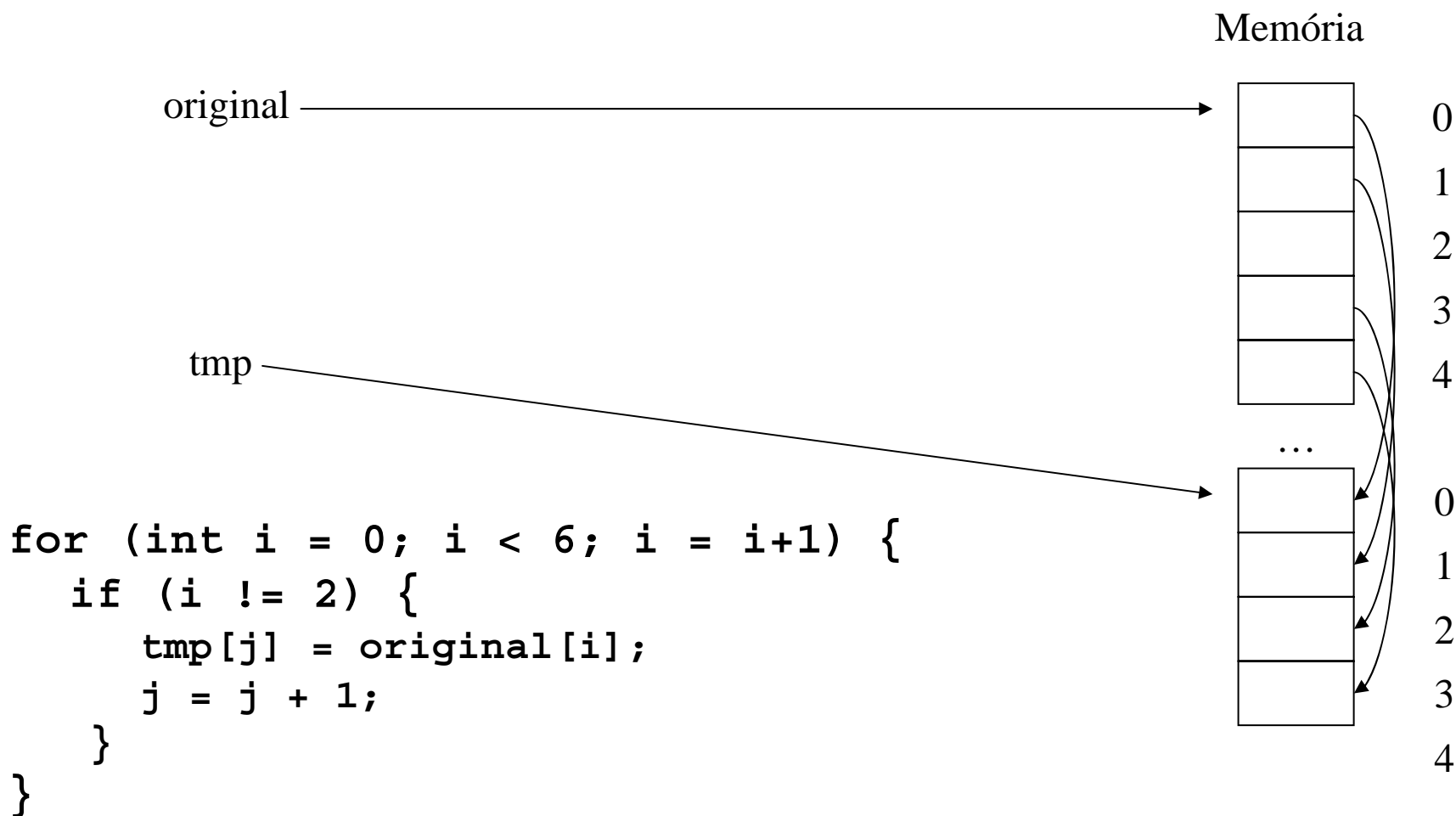
Reduzir um Vector: passo #1



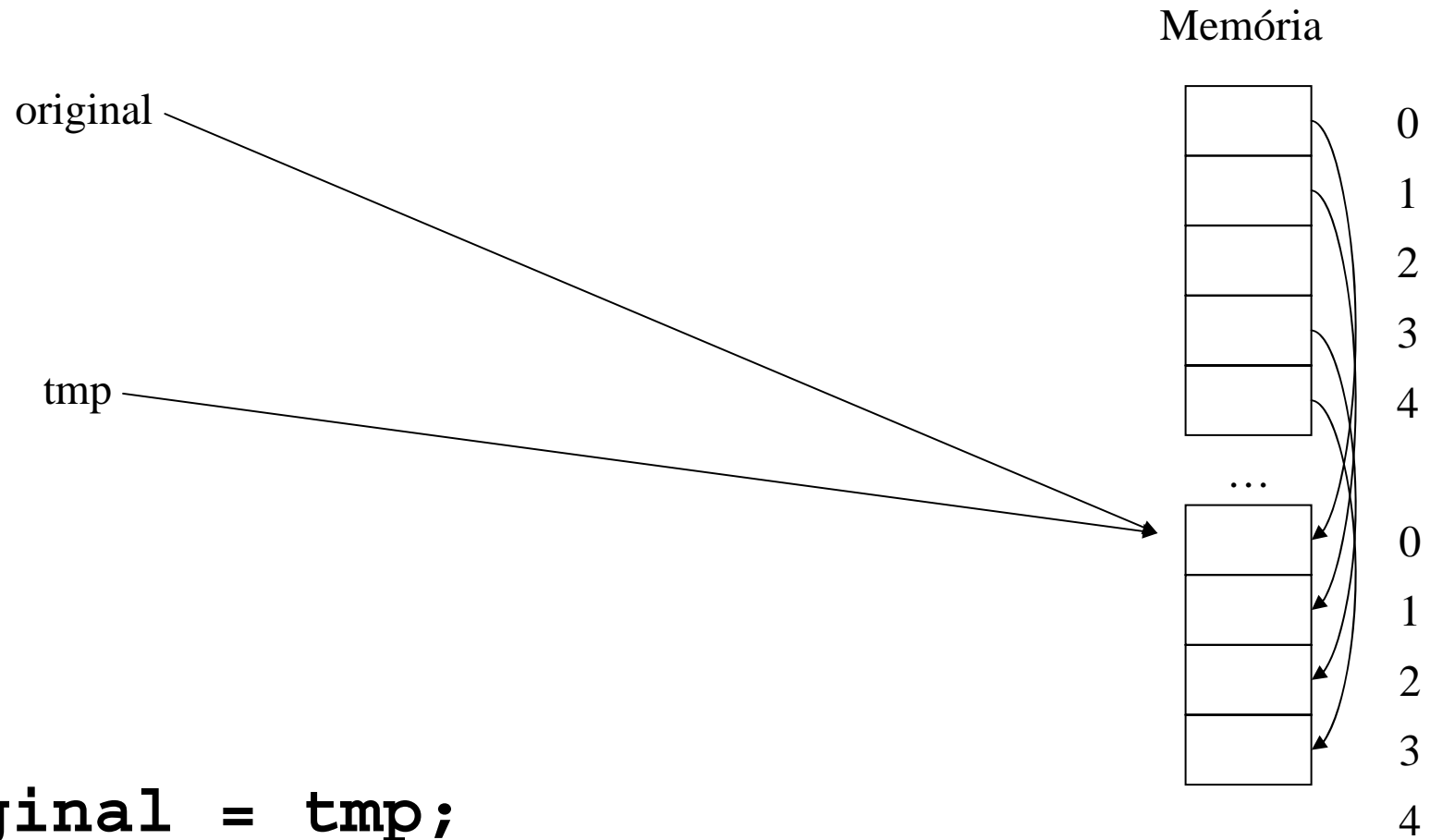
```
int original[] = {1, 3, 4, 6, 5};
```

```
int tmp[] = new int[4];
```

Reduzir um Vector: passo #2



Reduzir um Vector: passo #3



Reduzir um Vector

```
int original[] = {1, 3, 4, 6, 5, 10};
int tmp[] = new int[5];
int j = 0;
for (int i = 0; i < 6; i = i+1) {
    if (i != 2) {
        tmp[j] = original[i];
        j = j + 1;
    }
}
original = tmp;
```

Exercício #1

- Modificar o Exemplo #1 de forma a cada linha ter a sua própria cor.
 - A cor pode ser inicializada de forma aleatória

Intervalo

Matrizes

- Extensão de um vector para duas dimensões

Variável simples:

5

Vector:

i →						
	5	1	9	5	4	3

⌈

Matriz:

	i →					
j ↓	10	1	3	5	2	4
	5	9	5	2	8	5
	1	5	7	1	2	5
	2	8	1	5	7	1

⌈ ⌈

Matriz

```
int tabela[] [];
```

```
tabela = new int [10] [10];
```

- É preciso especificar as duas dimensões
 - Altura e largura

Percorrer Matrizes

- São precisos dois ciclos para percorrer uma matriz:

```
int i;
int j;
for (j = 0; j < 10; j = j + 1) {
    for (i = 0; i < 10; i = i + 1) {
        tabela[j][i] = 1;
    }
}
```

Exemplo #3

```
// contem a cor de cada pixel
int ecra[] [];

void setup() {
    size(200, 200);

    ecra = new int[200][200];

    // inicializar a matriz com valores
    // aleatorios
    for (int i = 0; i < 200; i++) {
        for (int j = 0; j < 200; j++) {
            ecra[i][j] =
color(random(255),          random(255),
random(255));
        }
    }
}

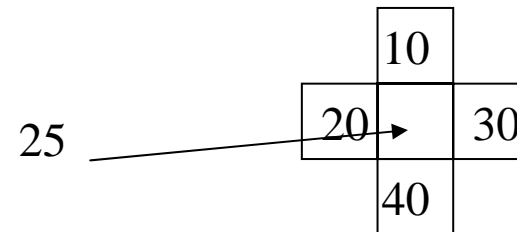
void draw() {
    for (int i = 0; i < 200; i++) {
        for (int j = 0; j < 200; j++) {
            stroke(ecra[i][j]);
            point(j, i);
        }
    }
}
```

Exercício #2

- Modifique o Exemplo #3 de forma a gerar cores diferentes em cada frame

Exercício #3

- Modifique o Exemplo #3 de forma a calcular, em cada frame, cada pixel da seguinte forma:
 - A média dos pixels adjacentes



Projecto Semanal

1. Modifique o Exercício #3 de forma a que a matriz comece com cores diferentes em regiões diferentes
 1. Divida a matriz em quatro regiões:

