



# Linguagem de programação I

## Vetores em Java, vetores de objetos

MSc Romanuel Ramón Antunez.  
Isutic, 2013

- Existe apenas um bem: o conhecimento. Há apenas um mal, a ignorância.
- Se você quer um pedacinho do Paraíso, acredite em Deus. Mas se você quer conquistar o mundo, acredite em você porque Deus já te deu tudo o que você precisa para você vencer.

# Objetivos da aula de hoje

- Identificar as situações em que é necessário o uso de vetores.
- Declarar vetores no Java
- O uso de vetores na resolução de problemas.

# Personalidade de Hoje



- 23/06/1912 – 07/06/1954
- Londres
- Matemático, lógico, cientista de la computação, criptógrafo y filósofo britânico.
- Creado del modelo de máquina automática
  - Derivado depois teoria de autómatas

- 07/03/2011
- Cienfuegos, Cuba
- Pablo Javier Ramón Hernández



# Lembrando...

- **Ciclos precondicionales**

**while**(< condición de repetición >)  
[ Bloque de instrucciones]

# Lembrando...

- **Ciclos poscondicionales**

do

[ Bloque de instrucciones]

while(< condición de repetición>)



# Lembrando...

- **Ciclos for**

```
for(< enunciado1 >; < condición de  
repetición >; <enunciado2 >)  
[ Bloque de instrucciones]
```

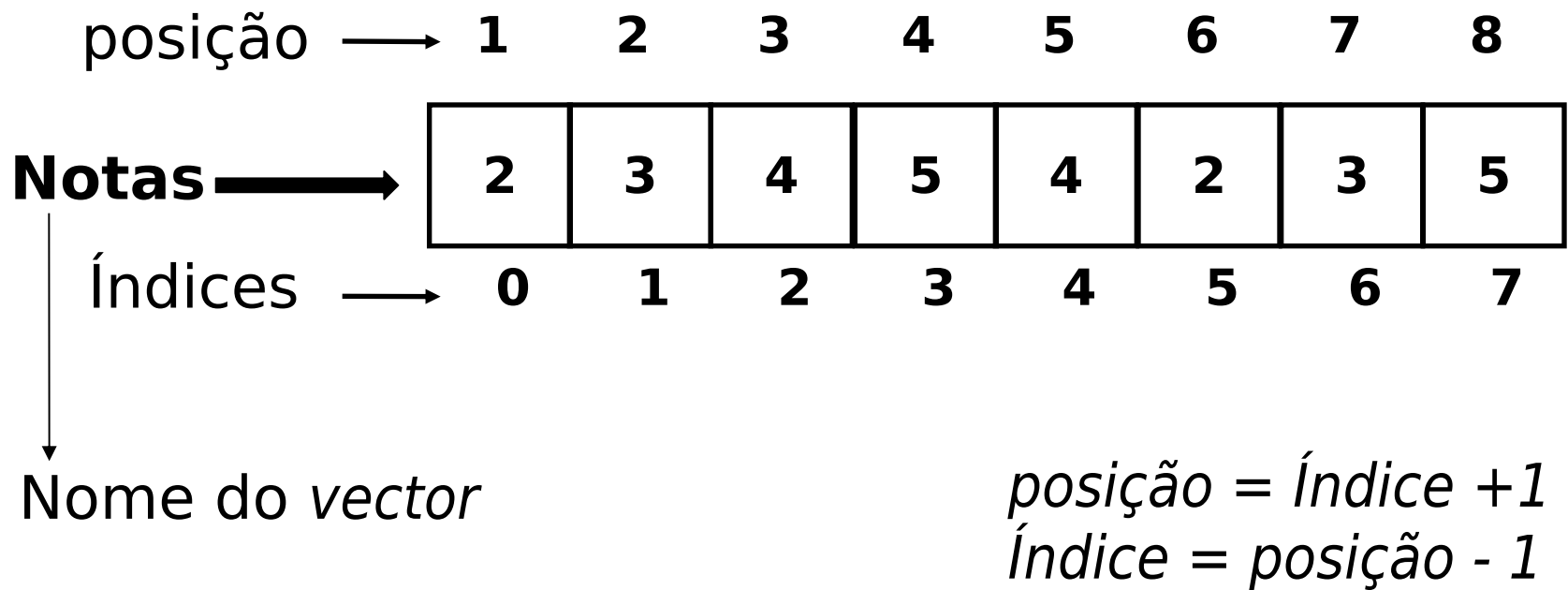


# Problema...

```
public class PromedioNotas
{
    public static void main ( String [] args )
    {
        int nota , suma = 0, promedio ;
        for(int i = 1; i <= 30; i++)
        {
            IO. print(" Entre una nota : ");
            nota = IO. readInt ();
            suma += nota ;
        }
        promedio = suma /30;
        IO. println ("El promedio de notas es: " + promedio );
    }
}
```

# Vector

Um vector é um grupo de locais na memória que armazena com o mesmo nome (identificador) para a coleta de dados do mesmo tipo



# Sintaxis no Java

```
<tipo>[] <variable>= new  
<tipo>[<cantidad de elementos>];
```

Para as notas fica assim

```
int [] notas = new int [30]; // Un array de 30  
valores enteros
```

Com o attributo publico lenght pode obter a longitude do vector.

# Trabajo con vectores

Sintaxis:

**<variable>[<expresión de tipo entero>];**

Os índices estão no intervalo de 0 .. n-1 e as posições de 1 .. n, onde n é o **comprimento ou o tamanho** do vector.

Exemplos:

**//Suma dos primeiros 3 elementos**

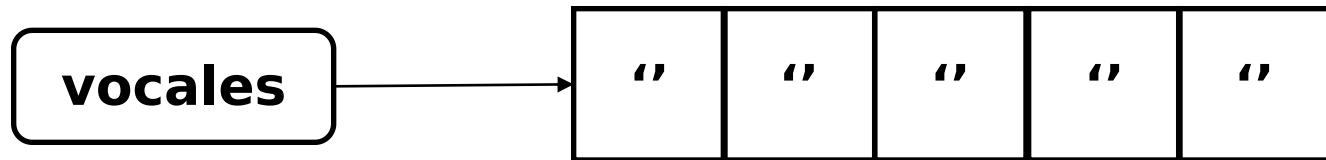
**suma = notas[0] + notas[1] + notas[2];**

**//Atribuir o valor 4 para o quarto elemento**

**notas[3] = 4;**

# Inicialização

Si creamos el siguiente *array*:  
**char[] vocales = new char[5];**

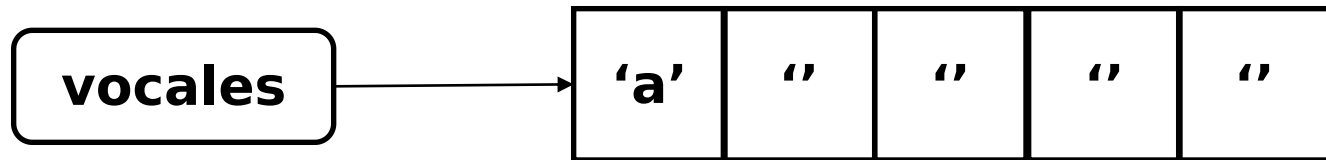


# Inicialização

Si creamos el siguiente *array*:

```
char[] vocales = new char[5];
```

```
vocales[0] = 'a';
```



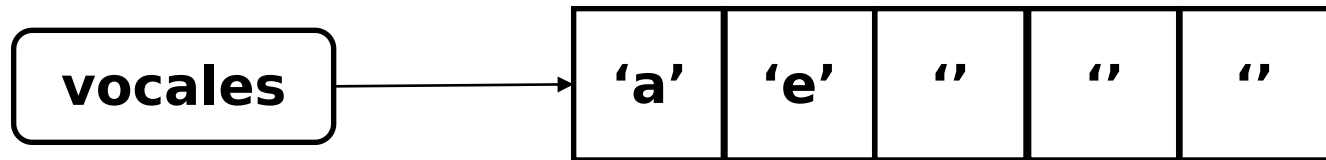
# Inicialização

Si creamos el siguiente *array*:

```
char[] vocales = new char[5];
```

```
vocales[0] = 'a';
```

```
vocales[1] = 'e';
```





# Inicialização

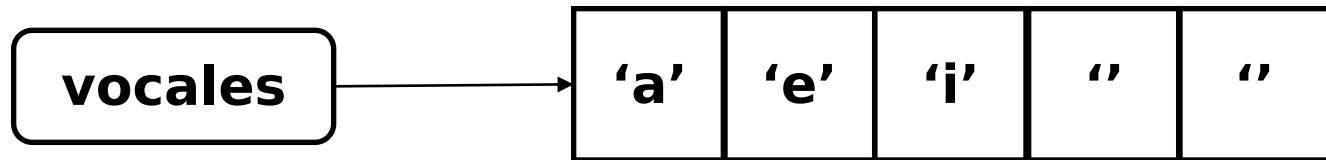
Si creamos el siguiente *array*:

```
char[] vocales = new char[5];
```

```
vocales[0] = 'a';
```

```
vocales[1] = 'e';
```

```
vocales[2] = 'i';
```



# Inicialização

Si creamos el siguiente *array*:

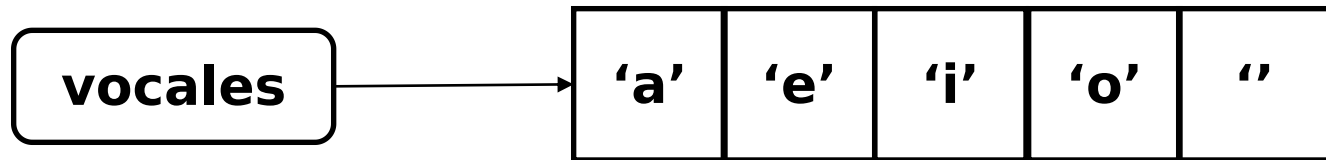
```
char[] vocales = new char[5];
```

```
vocales[0] = 'a';
```

```
vocales[1] = 'e';
```

```
vocales[2] = 'i';
```

```
vocales[3] = 'o';
```



# Inicialização

Si creamos el siguiente *array*:

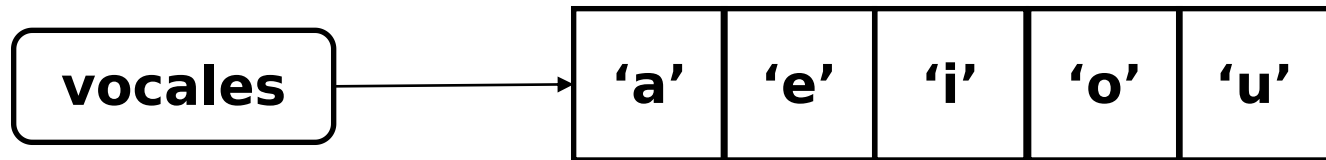
```
char[] vocales = new char[5];
```

```
vocales[0] = 'a';
```

```
vocales[1] = 'e';
```

```
vocales[2] = 'i';
```

```
vocales[3] = 'o';
```



# Inicialização

Si creamos el siguiente *array*:

```
char[] vocales = new char[5];
```

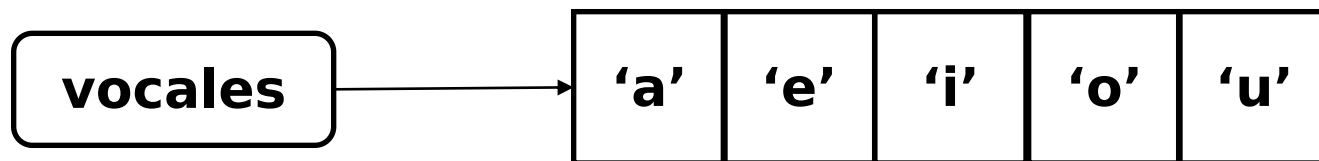
```
vocales[0] = 'a';
```

```
vocales[1] = 'e';
```

```
vocales[2] = 'i';
```

```
vocales[3] = 'o';
```

```
vocales[4] = 'u';
```



Isto é equivalente a:

```
vocales = {'a', 'e', 'i', 'o', 'u'};
```

① PARA CONSEGUIR UNA MUJER, NECESITAS TIEMPO Y DINERO, POR LO TANTO :

$$\textcircled{1} \quad \boxed{\text{MUJER} = \text{TIEMPO} \times \text{DINERO}}$$

② "EL TIEMPO ES ORO" POR LO TANTO "EL TIEMPO ES DINERO"

$$\textcircled{2} \quad \boxed{\text{TIEMPO} = \text{DINERO}}$$

③ REEMPLAZAMOS ② EN ①

$$\text{MUJER} = \text{DINERO} \times \text{DINERO}$$

$$\textcircled{3} \quad \boxed{\text{MUJER} = (\text{DINERO})^2}$$

④ "EL DINERO ES LA RAÍZ DE TODOS LOS PROBLEMAS"

$$\textcircled{4} \quad \boxed{\text{DINERO} = \sqrt{\text{PROBLEMAS}}}$$

⑤ REEMPLAZAMOS ④ EN ③

$$\text{MUJER} = (\sqrt{\text{PROBLEMAS}})^2$$

$$\textcircled{5} \quad \boxed{\text{MUJER} = \text{PROBLEMAS}}$$

OK.

# Asignacion y paso por parametros

```
class Calculadora
{
    public int Suma ( int [] A)
    {
        int suma = 0;
        for( int i = 0; i < A. length ; i++)
            suma += A[i];
        return suma ;
    }
    public double Promedio ( int [] A)
    {
        return Suma (A)/A. length ;
    }
}
```

# Asignacion y paso por parametros

- Ao passar um vetor como parâmetros, não cria-se uma cópia de cada um dos componentes do vetor de notas em cada um dos componentes do parâmetro formal A. O que realmente é atribuído a parâmetro formalmente A é uma cópia do mesma referência ao objeto vetor de que está em notas.



# Asignacion y paso por parametros

- Cada programador deve levar em conta a semântica da transferência de parâmetros-array, como quaisquer alterações feitas no parâmetro formal será refletido também no parâmetro real. O mesmo se aplica a atribuição semântica de vetores, isto é quando se faz:

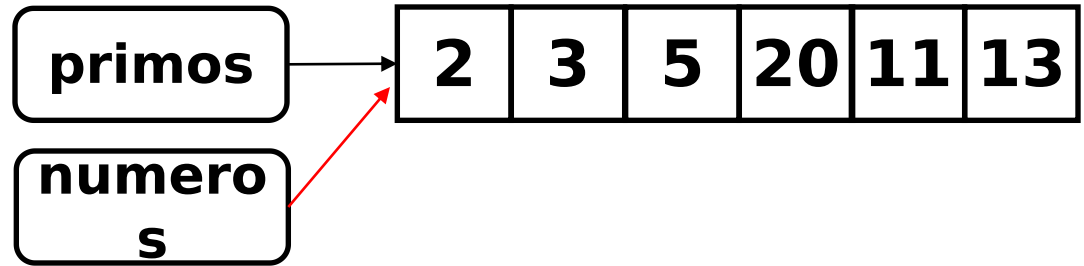
# Asignacion y paso por parametros

```
int[] primos = {2, 3, 5, 7, 11, 13};
```

```
    primos[3]=20
```

```
int[] numeros;
```

```
numeros = primos;
```



# Asignacion y paso por parametros

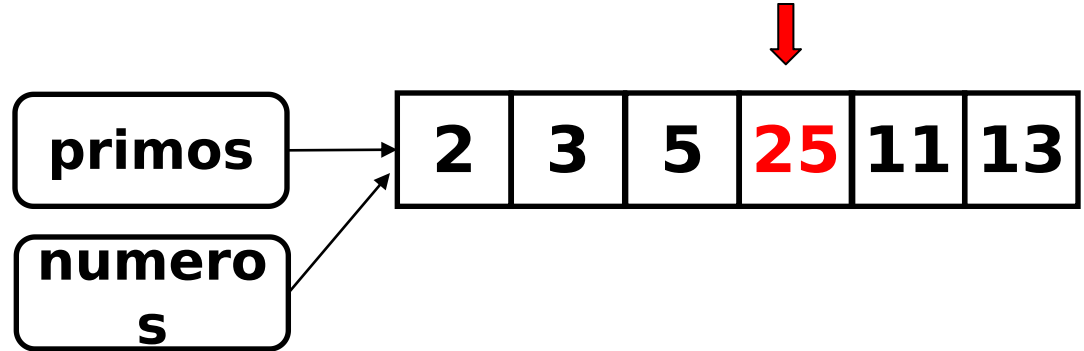
```
int[] primos = {2, 3, 5, 7, 11, 13};
```

```
    primos[3]=20
```

```
int[] numeros;
```

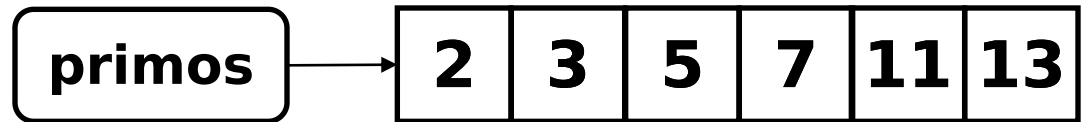
```
numeros = primos;
```

```
numeros[3] = 25;
```



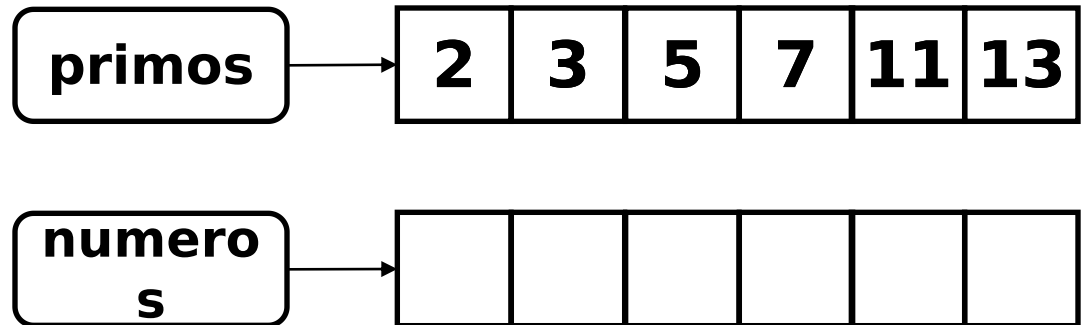
# Copia

```
int[] primos = {2, 3, 5, 7, 11, 13};
```



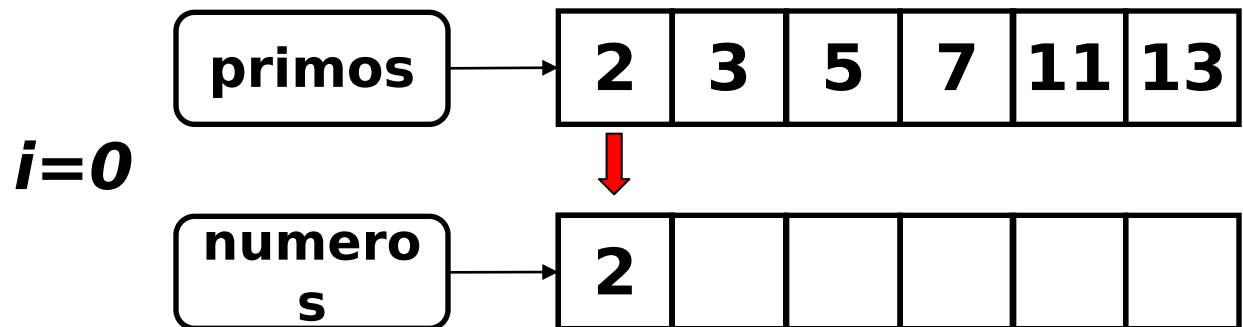
# Copia

```
int[] primos = {2, 3, 5, 7, 11, 13};  
int[] numeros = new int[primos.length];
```



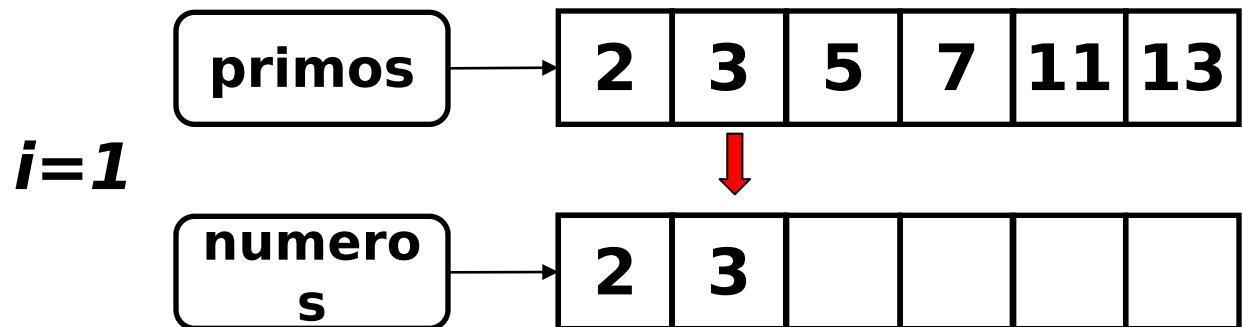
# Copia

```
int[] primos = {2, 3, 5, 7, 11, 13};  
int[] numeros = new int[primos.length];  
for(int i = 0; i < primos.length; i++)  
    numeros[i] = primos[i];
```



# Copia

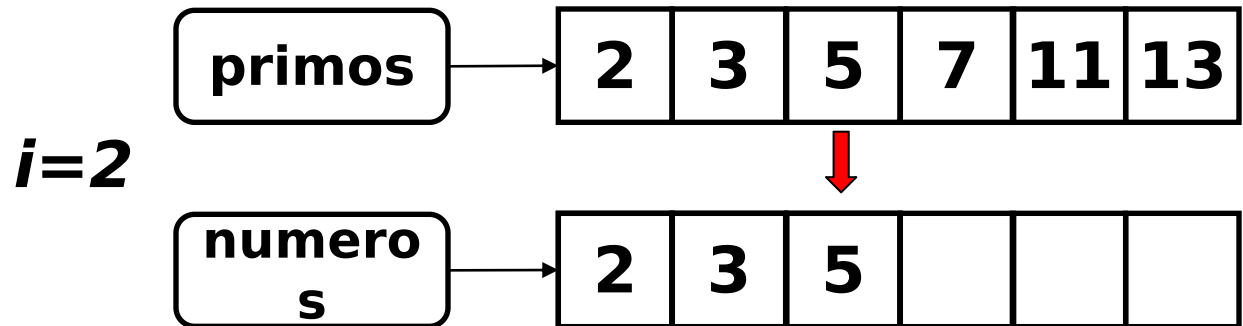
```
int[] primos = {2, 3, 5, 7, 11, 13};  
int[] numeros = new int[primos.length];  
for(int i = 0; i < primos.length; i++)  
    numeros[i] = primos[i];
```





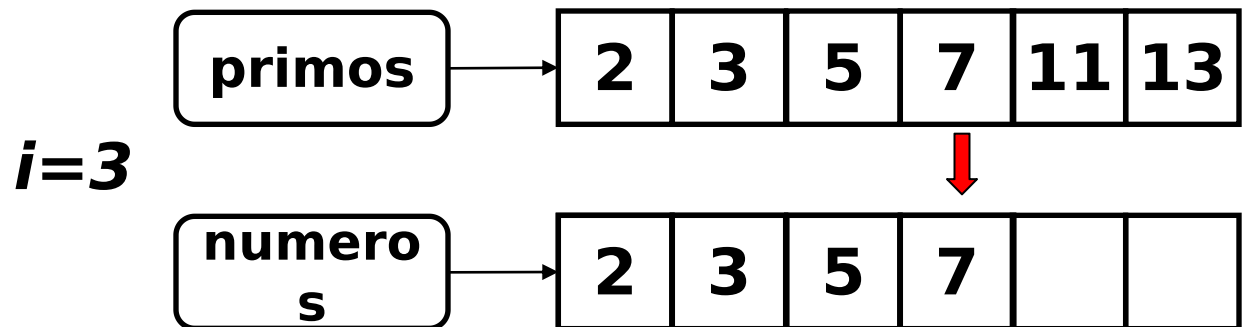
# Copia

```
int[] primos = {2, 3, 5, 7, 11, 13};  
int[] numeros = new int[primos.length];  
for(int i = 0; i < primos.length; i++)  
    numeros[i] = primos[i];
```



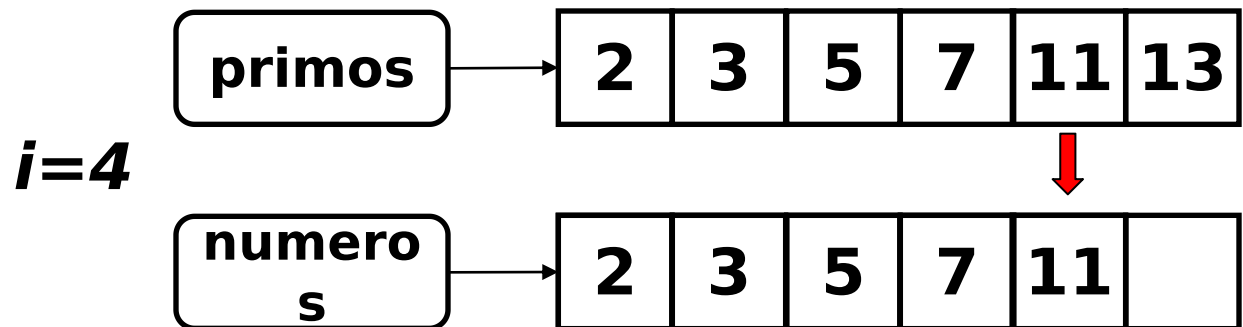
# Copia

```
int[] primos = {2, 3, 5, 7, 11, 13};  
int[] numeros = new int[primos.length];  
for(int i = 0; i < primos.length; i++)  
    numeros[i] = primos[i];
```



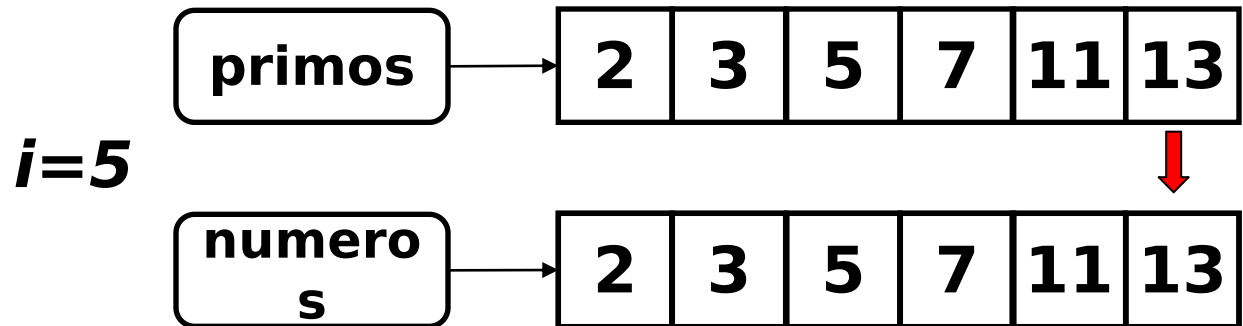
# Copia

```
int[] primos = {2, 3, 5, 7, 11, 13};  
int[] numeros = new int[primos.length];  
for(int i = 0; i < primos.length; i++)  
    numeros[i] = primos[i];
```



# Copia

```
int[] primos = {2, 3, 5, 7, 11, 13};  
int[] numeros = new int[primos.length];  
for(int i = 0; i < primos.length; i++)  
    numeros[i] = primos[i];
```



# Clase Estudiante

```
class Estudiante {  
    private String nombre;    //Nombre del estudiante  
    private int[] notas;      //Notas del estudiante  
    private int cantRealNotas; //Cantidad de notas  
  
    public Estudiante(String nombre, int cantMaxNotas){  
        this.nombre = nombre;  
        notas = new int[cantMaxNotas];  
        CantRealNotas = 0;  
    }  
    public int CantNotas(int nota) {  
        ...  
    }  
    public double Promedio() {  
        ...  
    }  
    public int ObtenerNota(int pos) {  
        return notas[pos-1];  
    }  
}
```

# Vetores de objetos

- Também pode criar vetores cujos elementos são objetos de uma classe particular. A declaração e a construção de um vetor de objetos é semelhante a como se explica, apenas o tipo de dados especificado deve ser o nome da classe.

<Nome da Classe>[] <variable>= new <Nome da classe>[<cantidad de elementos>];

Estudiante estudiantes = new Estudiantes [5];

# Vectores de objetos

- A operação `new` vai separar espaço para cinco objetos do tipo `Estudiante`, mas não cria objetos do tipo `Estudiante` para cada elemento do vetor, senão em cada elemento do vetor estudantes deixa o valor `null`

```
estudiantes [0] = new Estudiantes (" Hector", 5);
```

```
estudiantes [1] = new Estudiantes (" Raquel", 5);
```

```
estudiantes [2] = new Estudiantes (" Rafael", 5);
```

```
estudiantes [3] = new Estudiantes (" Paula", 5);
```

```
estudiantes [4] = new Estudiantes (" Rolando ", 5);
```



# Vectores de objetos

```
Estudiante [] estudiantes = {new Estudiante ("
    H'ector", 5) ,
new Estudiante (" Raquel", 5) ,
new Estudiante (" Rafael", 5) ,
new Estudiante (" Paula", 5) ,
new Estudiante (" Rolando ", 5) };
```

# Vectores de objetos

```
public class Programa
{
    public static void main ( String [] args )
    {
        /* Declaração , criação e inicialização do array de
        objetos de tipo Estudante */
        int nota ;
        for (int i = 0; i < 5; i++)
        {
            IO. println ("Del estudante " + estudantes [i].
            Nombre ());
```

# Vectores de objetos

```
for (int j = 0; j < 5; j++)  
{  
    IO. print(" Entre la nota " + (j+1));  
    nota = IO. readInt ();  
    estudiantes [i]. AdicionarNota (nota );  
}
```

```
}
```

```
int contador = 0;
```

```
for (int i = 0; i < 5; i++)
```

```
    if ( estudiantes [i]. Promedio () > 85)  
        contador ++;
```

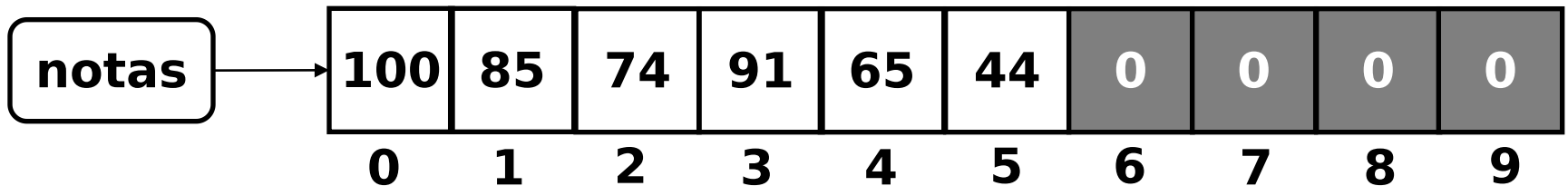
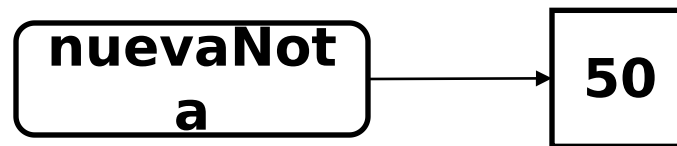
```
IO. println ("Hay " + contador + " estudiantes con  
promedio mayor que 85");
```

```
}
```

```
}
```

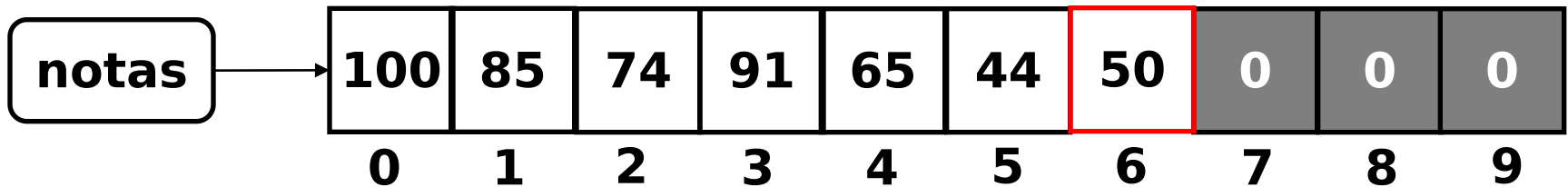
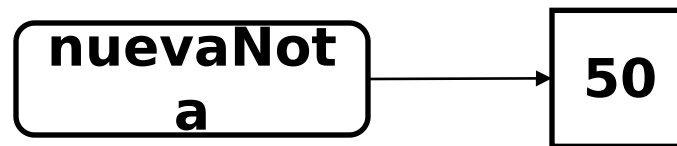
# Algoritmo Add

```
public void AdicionarNota(int nuevaNota )  
{  
    if(cantRealNotas < notas.length)  
        notas[cantRealNotas++] = nuevaNota ;  
}
```



# Algoritmo Add

```
public void AdicionarNota(int nuevaNota )  
{  
    if(cantRealNotas < notas.length)  
        notas[cantRealNotas++] = nuevaNota ;  
}
```



# Algoritmo Busca

```
public int BuscarNota(int nota)
{
    int i = 0;
    while ((i < cantRealNotas )&&( notas[i] != nota ))
        i++;
    if (i == cantRealNotas)
        return -1
    return i;
}
```

cantRealNotas

6

nota

91

i

notas

100

85

74

91

65

44

0

0

0

0

# Algoritmo Busca

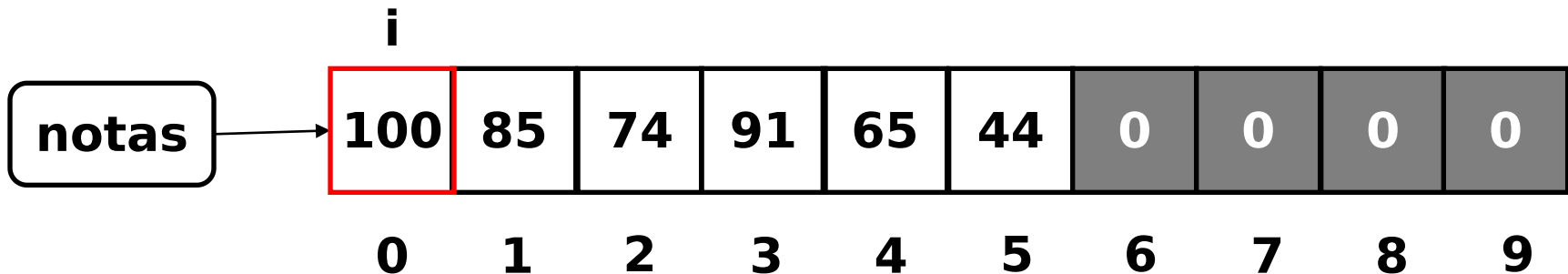
```
public int BuscarNota(int nota)
{
    int i = 0;
    while ((i < cantRealNotas )&&( notas[i] != nota ))
        i++;
    if (i == cantRealNotas)
        return -1
    return i;
}
```

cantRealNotas

6

nota

91



# Algoritmo Busca

```
public int BuscarNota(int nota)
{
    int i = 0;
    while ((i < cantRealNotas )&&( notas[i] != nota ))
        i++;
    if (i == cantRealNotas)
        return -1
    return i;
}
```

cantRealNotas

6

nota

91

notas

100

85

74

91

65

44

0

0

0

0

0

1

2

3

4

5

6

7

8

9



# Algoritmo Busca

```
public int BuscarNota(int nota)
{
    int i = 0;
    while ((i < cantRealNotas ) &&( notas[i] != nota ))
        i++;
    if (i == cantRealNotas)
        return -1;
    return i;
}
```

cantRealNotas

6

nota

91

notas

100

85

74

91

65

44

0

0

0

0

0

1

2

3

4

5

6

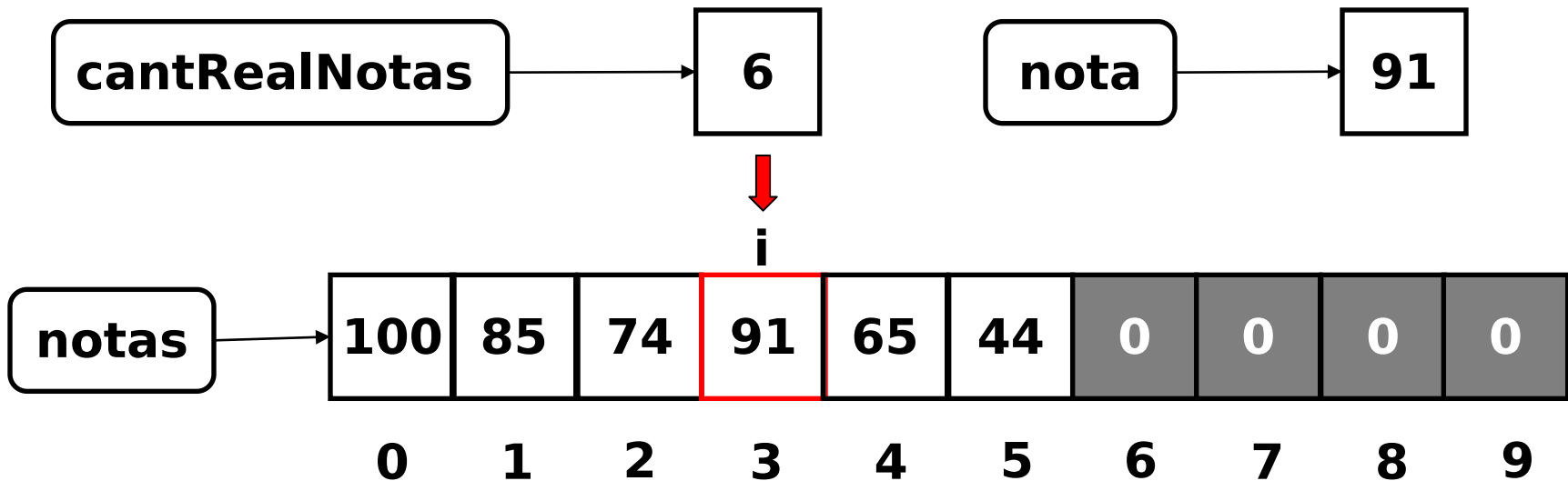
7

8

9

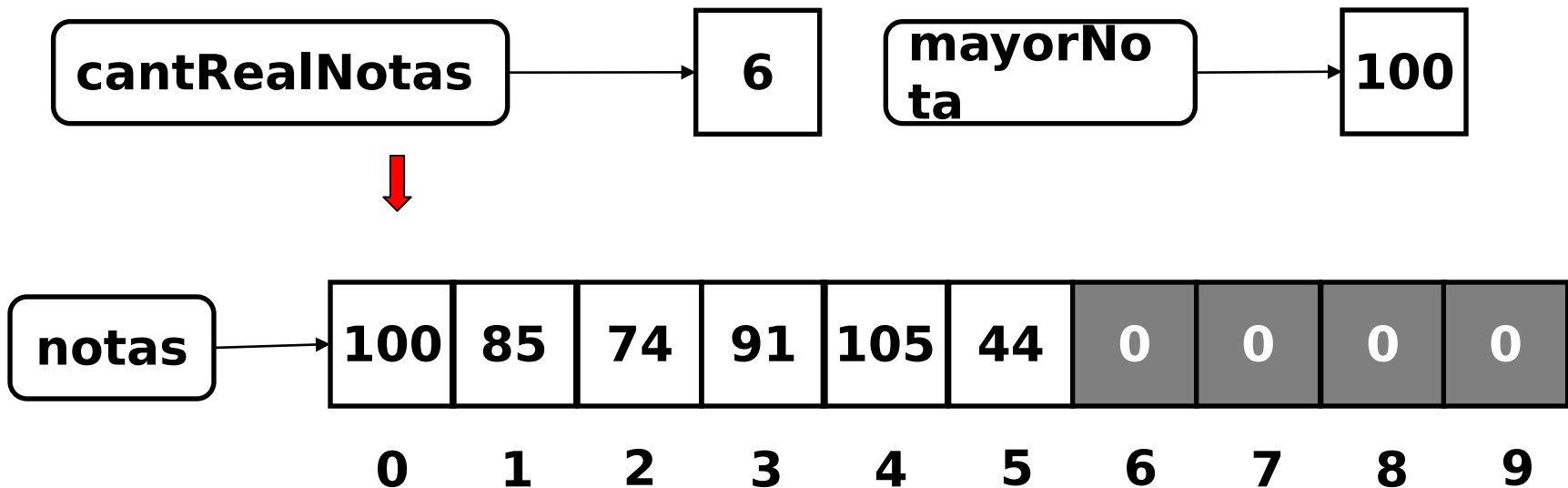
# Algoritmo Busca

```
public int BuscarNota(int nota)
{
    int i = 0;
    while ((i < cantRealNotas ) &&( notas[i] != nota ))
        i++;
    if (i == cantRealNotas)
        return -1;
    return i;
}
```



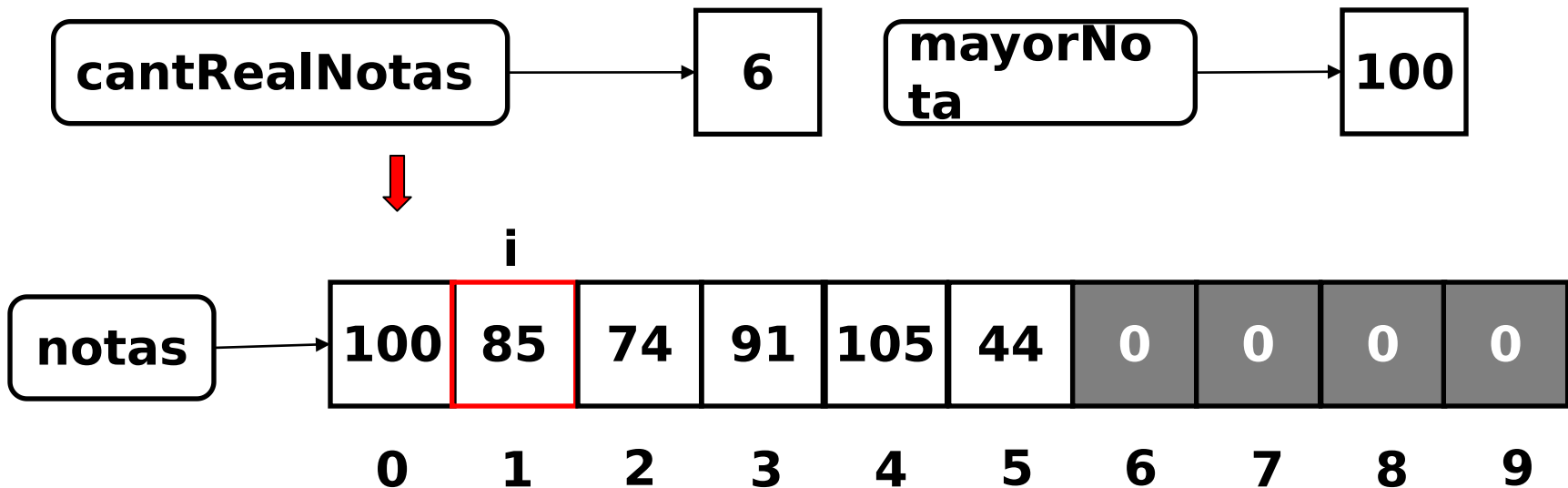
# Algoritmo Maior ou Menor

```
public int MayorNota()  
{  
    int mayorNota = notas[0];  
    for(int i = 1; i < cantRealNotas ; i++)  
        if(notas[i] > mayorNota)  
            mayorNota = notas[i];  
    return mayorNota ;  
}
```



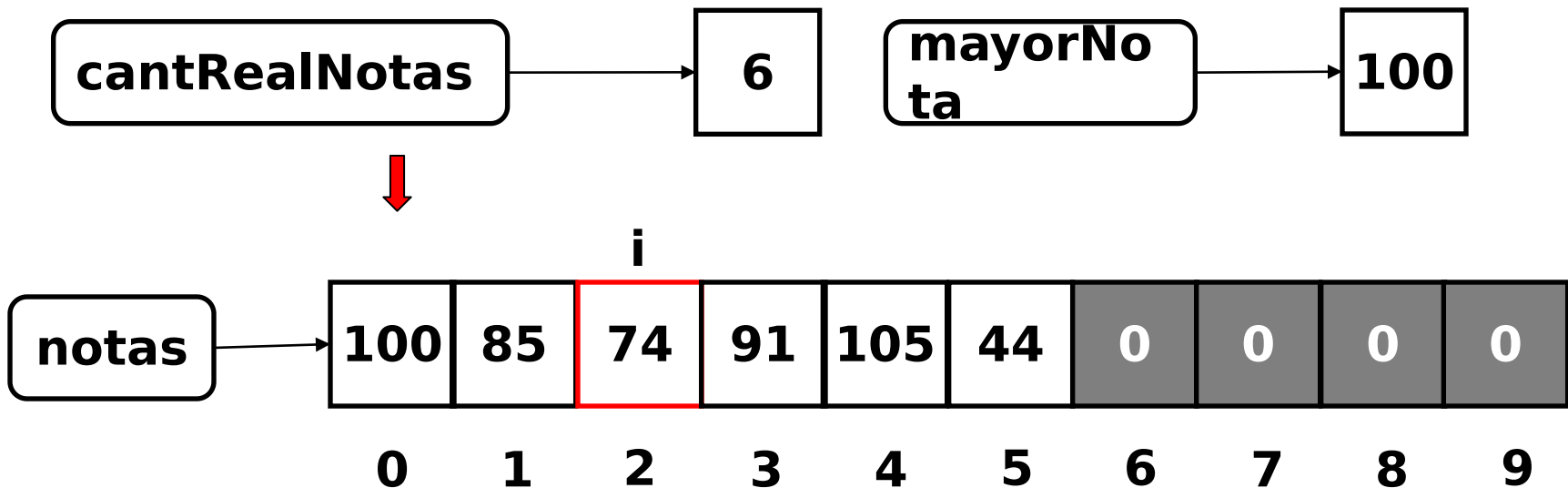
# Algoritmo Maior ou Menor

```
public int MayorNota()  
{  
    int mayorNota = notas[0];  
    for(int i = 1; i < cantRealNotas ; i++)  
        if(notas[i] > mayorNota)  
            mayorNota = notas[i];  
    return mayorNota ;  
}
```



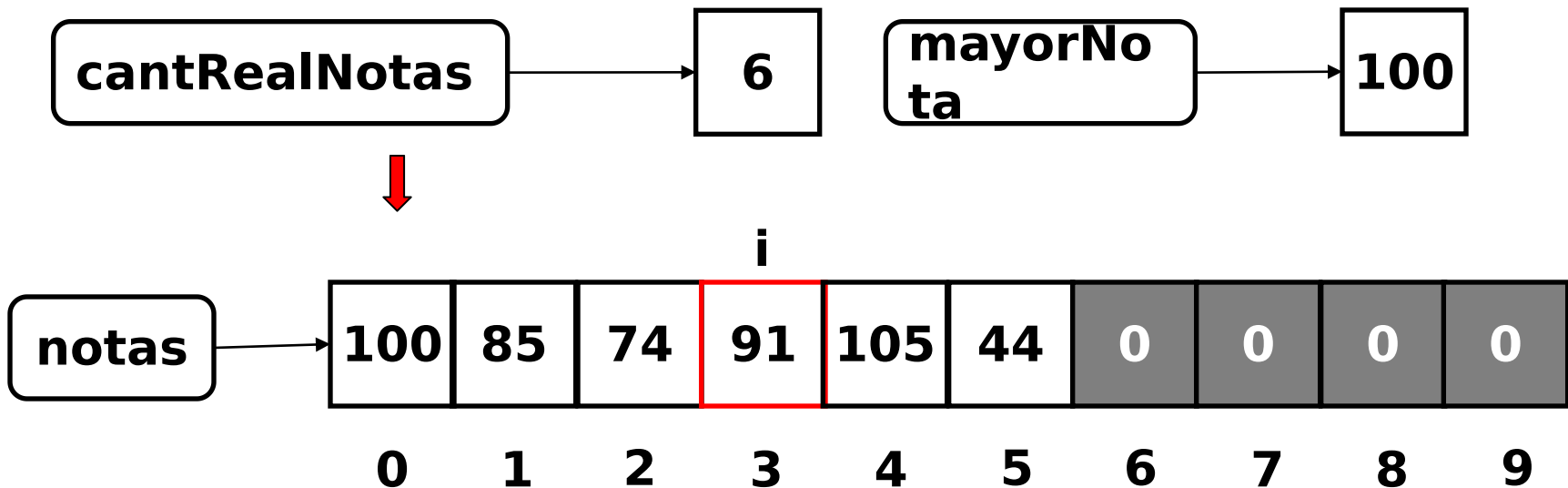
# Algoritmo Maior ou Menor

```
public int MayorNota()  
{  
    int mayorNota = notas[0];  
    for(int i = 1; i < cantRealNotas ; i++)  
        if(notas[i] > mayorNota)  
            mayorNota = notas[i];  
    return mayorNota ;  
}
```



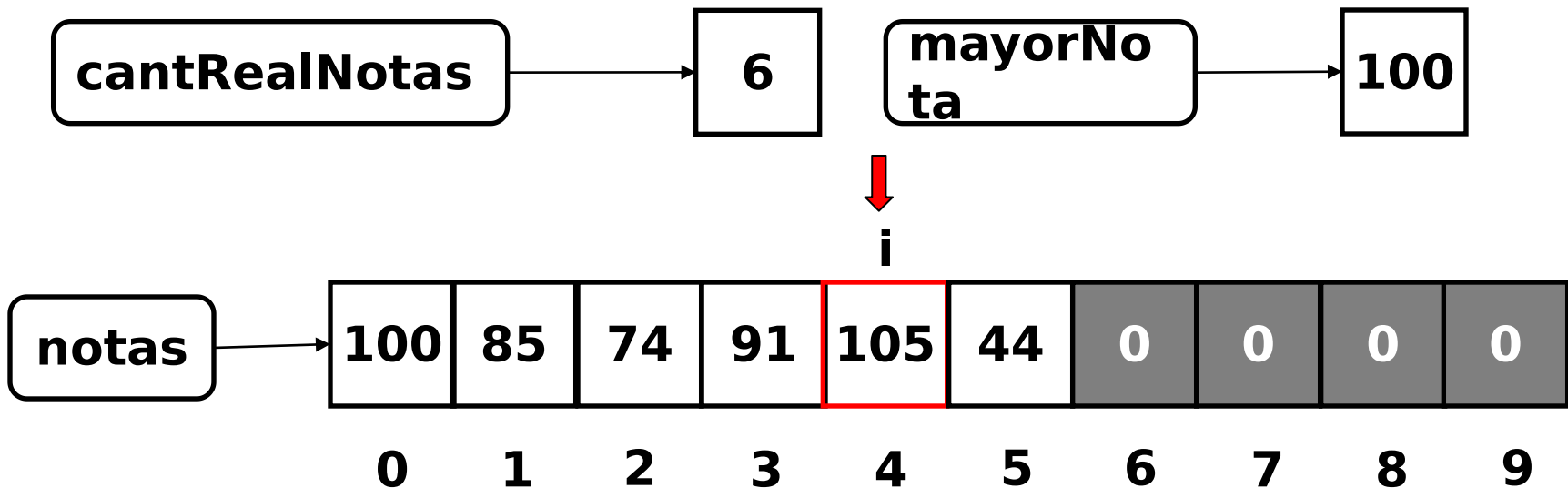
# Algoritmo Maior ou Menor

```
public int MayorNota()  
{  
    int mayorNota = notas[0];  
    for(int i = 1; i < cantRealNotas ; i++)  
        if(notas[i] > mayorNota)  
            mayorNota = notas[i];  
    return mayorNota ;  
}
```



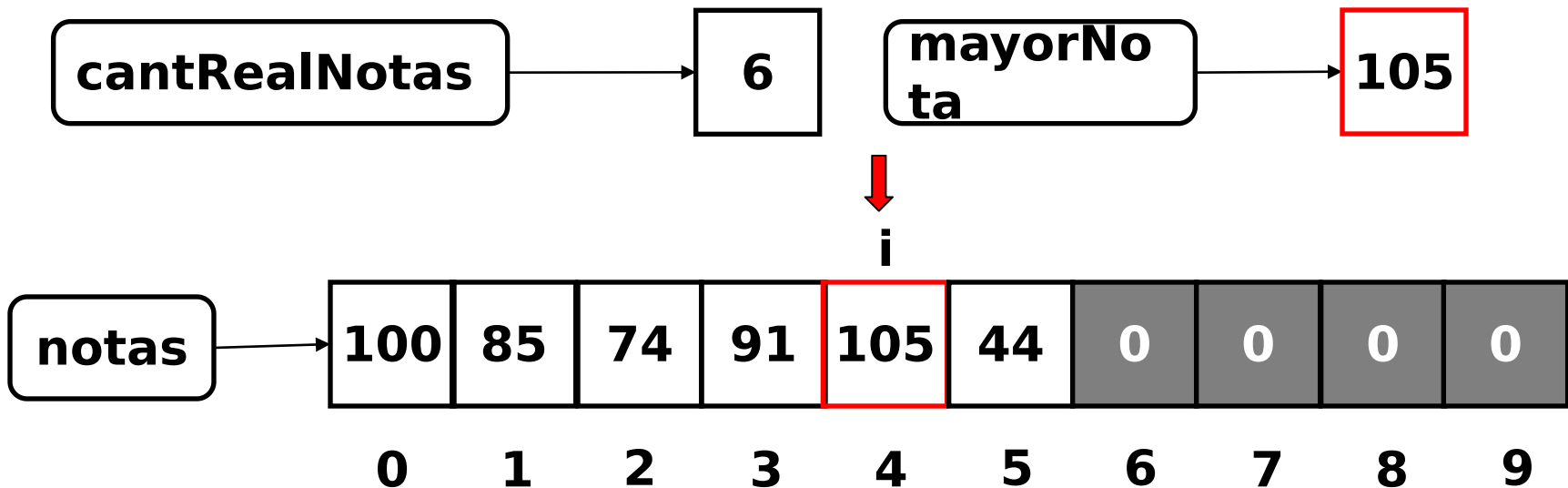
# Algoritmo Maior ou Menor

```
public int MayorNota()  
{  
    int mayorNota = notas[0];  
    for(int i = 1; i < cantRealNotas ; i++)  
        if(notas[i] > mayorNota)  
            mayorNota = notas[i];  
    return mayorNota ;  
}
```



# Algoritmo Maior ou Menor

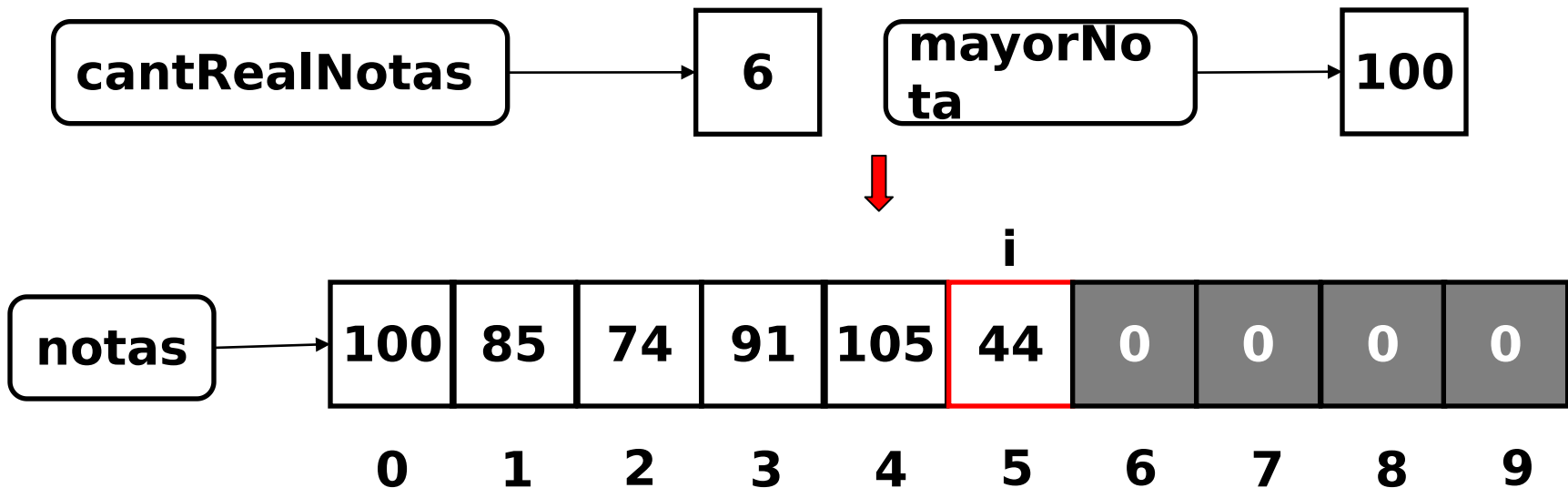
```
public int MayorNota()  
{  
    int mayorNota = notas[0];  
    for(int i = 1; i < cantRealNotas ; i++)  
        if(notas[i] > mayorNota)  
            mayorNota = notas[i];  
    return mayorNota ;  
}
```





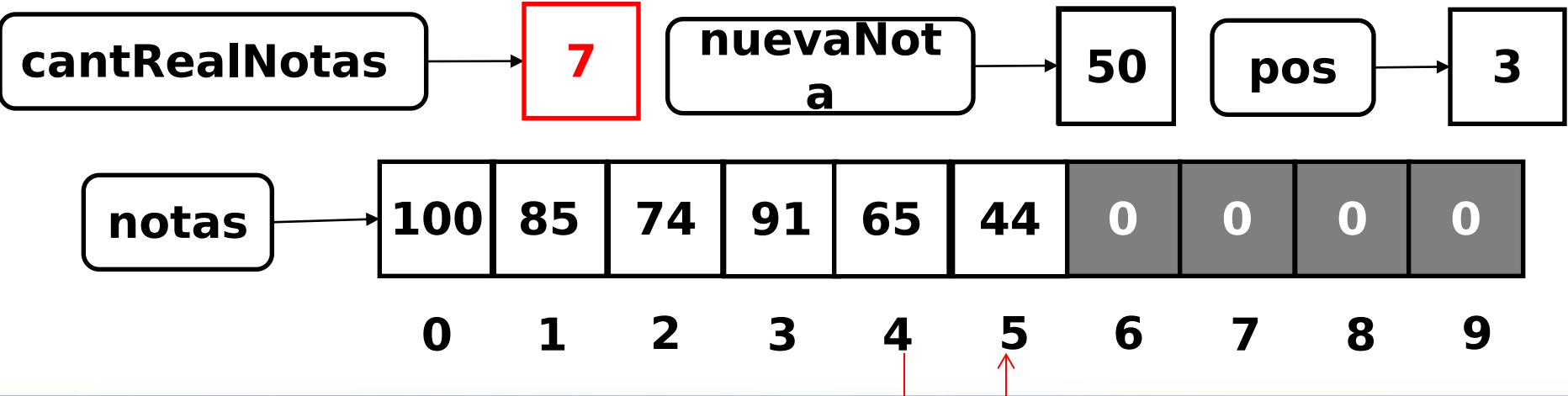
# Algoritmo Maior ou Menor

```
public int MayorNota()  
{  
    int mayorNota = notas[0];  
    for(int i = 1; i < cantRealNotas ; i++)  
        if(notas[i] > mayorNota)  
            mayorNota = notas[i];  
    return mayorNota ;  
}
```



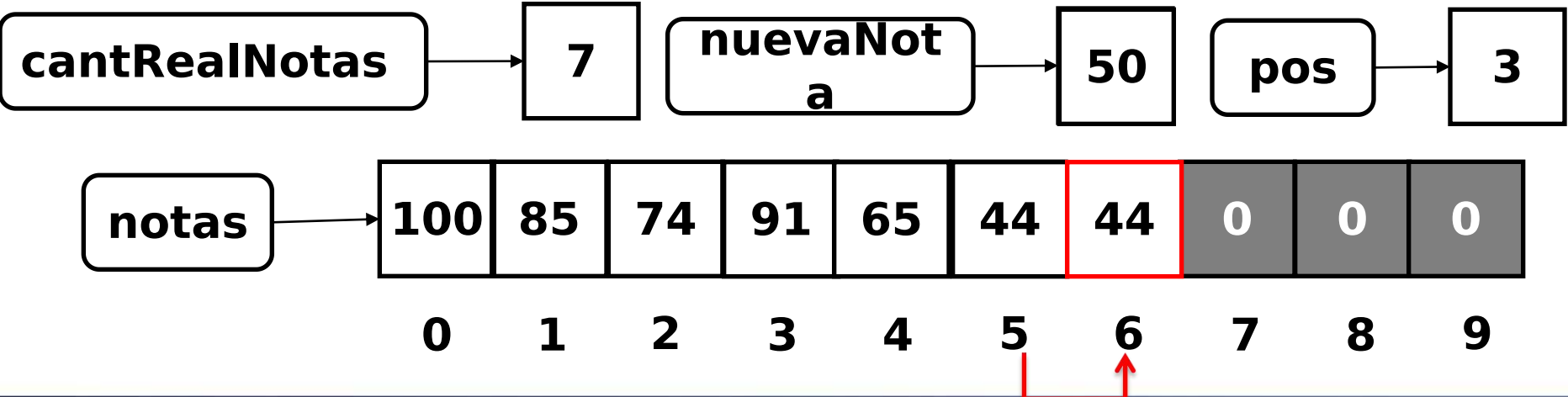
# Algoritmo Insertar

```
public void InsertarNota(int nuevaNota , int pos)
{
    if ((cantRealNotas < notas.length)
        &&( pos > 0) &&( pos <= cantRealNotas ))
    {
        cantRealNotas++;
        for(int i = cantRealNotas - 1; i > pos-1; i--)
            notas[i] = notas[i-1];
        notas[pos] = nuevaNota ;
    }
}
```



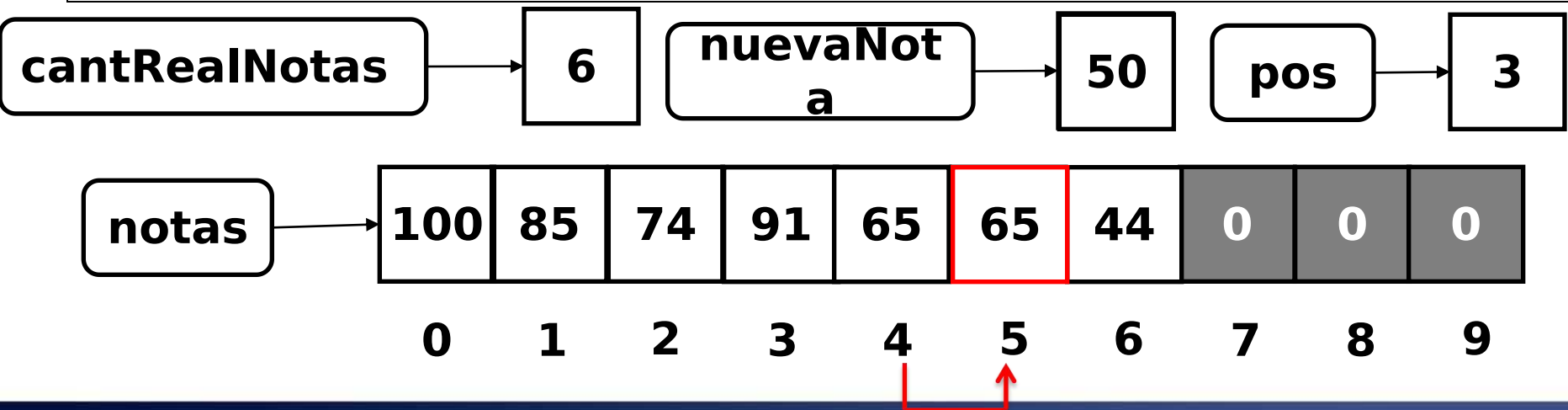
# Algoritmo Insertar

```
public void InsertarNota(int nuevaNota , int pos)
{
    if ((cantRealNotas < notas.length)
        &&( pos > 0) &&( pos <= cantRealNotas ))
    {
        cantRealNotas++;
        for(int i = cantRealNotas - 1; i > pos-1; i--)
            notas[i] = notas[i-1];
        notas[pos] = nuevaNota ;
    }
}
```



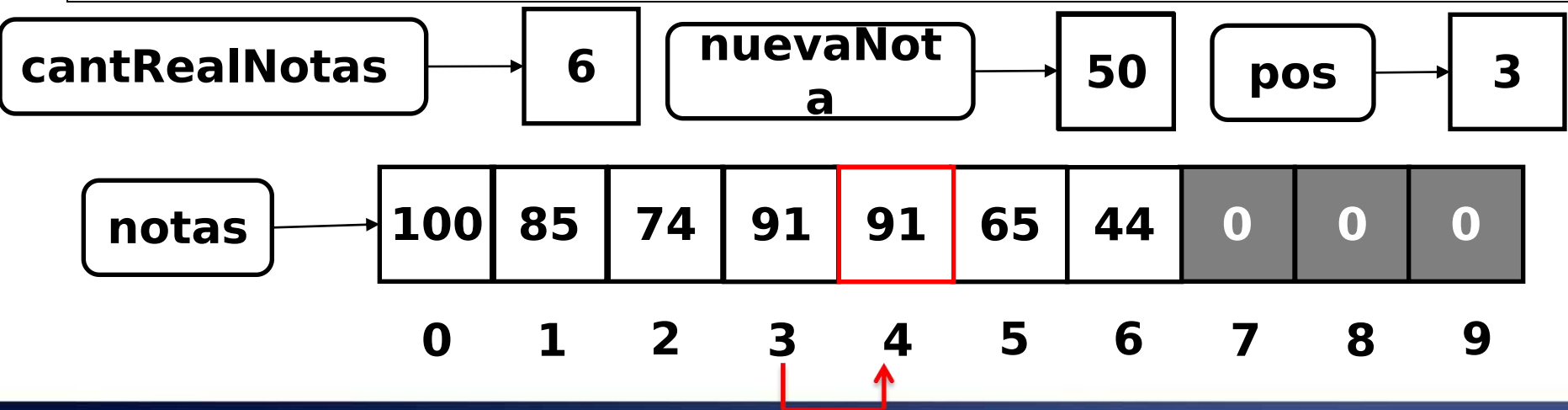
# Algoritmo Insertar

```
public void InsertarNota(int nuevaNota , int pos)
{
    if ((cantRealNotas < notas.length)
        &&( pos > 0) &&( pos <= cantRealNotas ))
    {
        cantRealNotas++;
        for(int i = cantRealNotas - 1; i > pos-1; i--)
            notas[i] = notas[i-1];
        notas[pos] = nuevaNota ;
    }
}
```



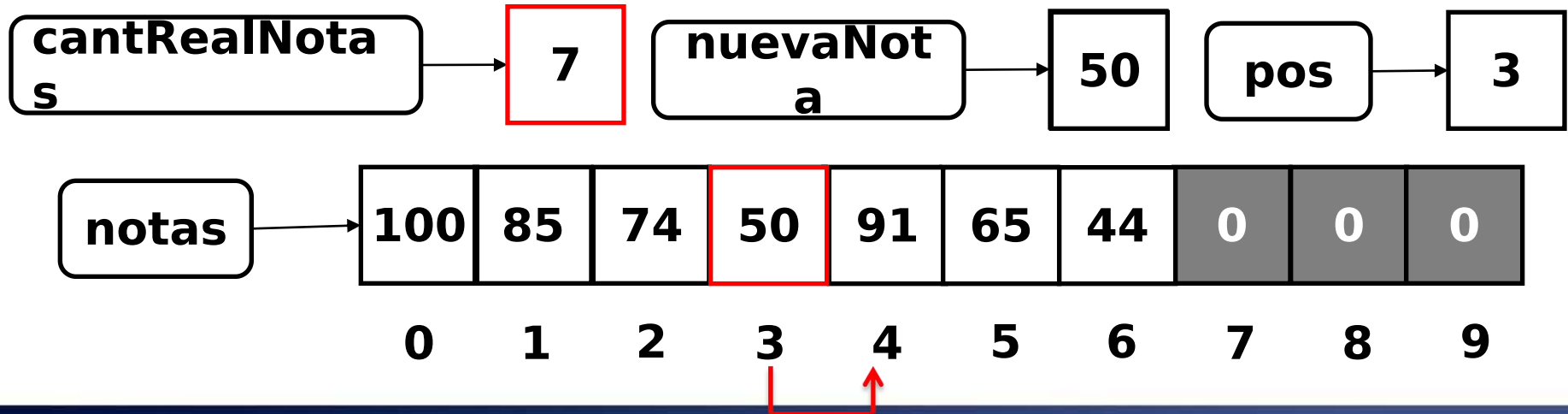
# Algoritmo Insertar

```
public void InsertarNota(int nuevaNota , int pos)
{
    if ((cantRealNotas < notas.length)
        &&( pos > 0) &&( pos <= cantRealNotas ))
    {
        cantRealNotas++;
        for(int i = cantRealNotas - 1; i > pos-1; i--)
            notas[i] = notas[i-1];
        notas[pos] = nuevaNota ;
    }
}
```



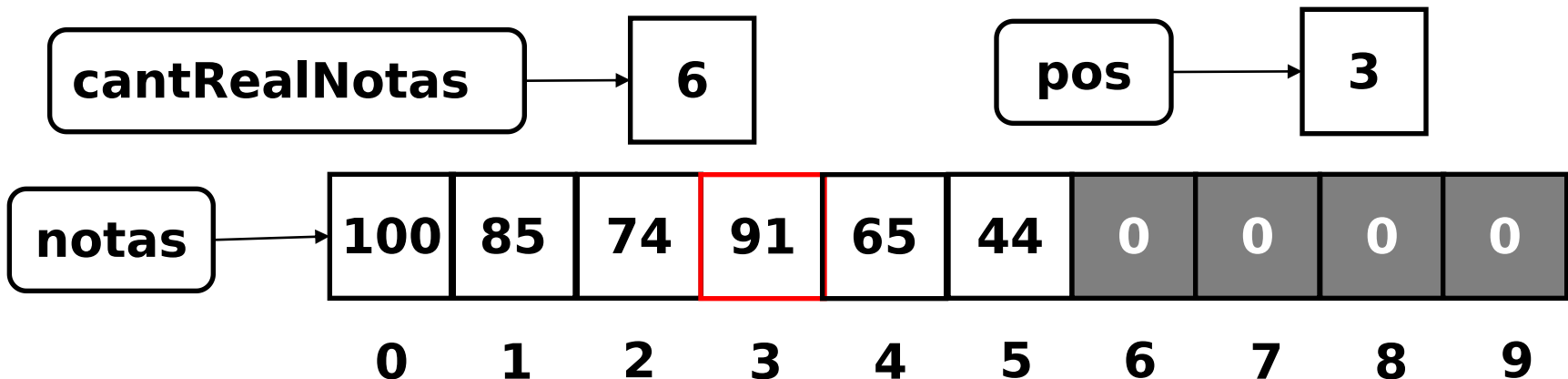
# Algoritmo Insertar

```
public void InsertarNota(int nuevaNota , int pos)
{
    if ((cantRealNotas < notas.length)
        &&( pos > 0) &&( pos <= cantRealNotas ))
    {
        cantRealNotas++;
        for(int i = cantRealNotas - 1; i > pos-1; i--)
            notas[i] = notas[i-1];
        notas[pos] = nuevaNota ;
    }
}
```



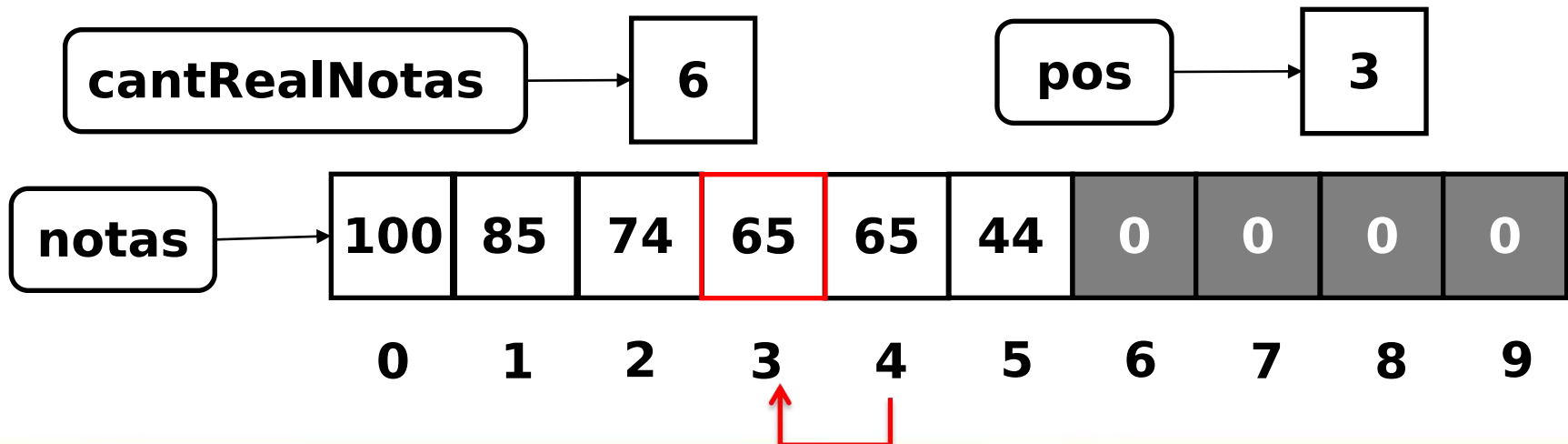
# Algoritmo Eliminar

```
public void EliminarNota(int pos)
{
    if ((cantRealNotas > 0) && (pos > 0) &&
        (pos <= cantRealNotas))
    {
        for(int i = pos-1; i < cantRealNotas ; i++)
            notas[i] = notas[i+1];
        cantRealNotas --;
    }
}
```



# Algoritmo Eliminar

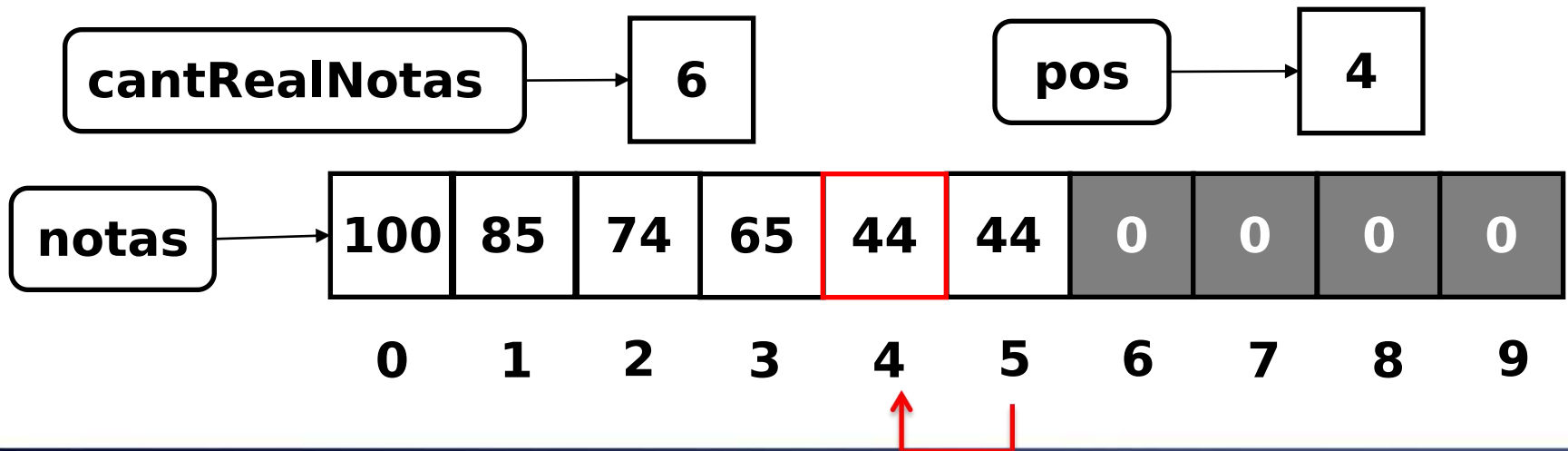
```
public void EliminarNota(int pos)
{
    if ((cantRealNotas > 0) && (pos > 0) &&
        (pos <= cantRealNotas))
    {
        for(int i = pos-1; i < cantRealNotas ; i++)
            notas[i] = notas[i+1];
        cantRealNotas --;
    }
}
```





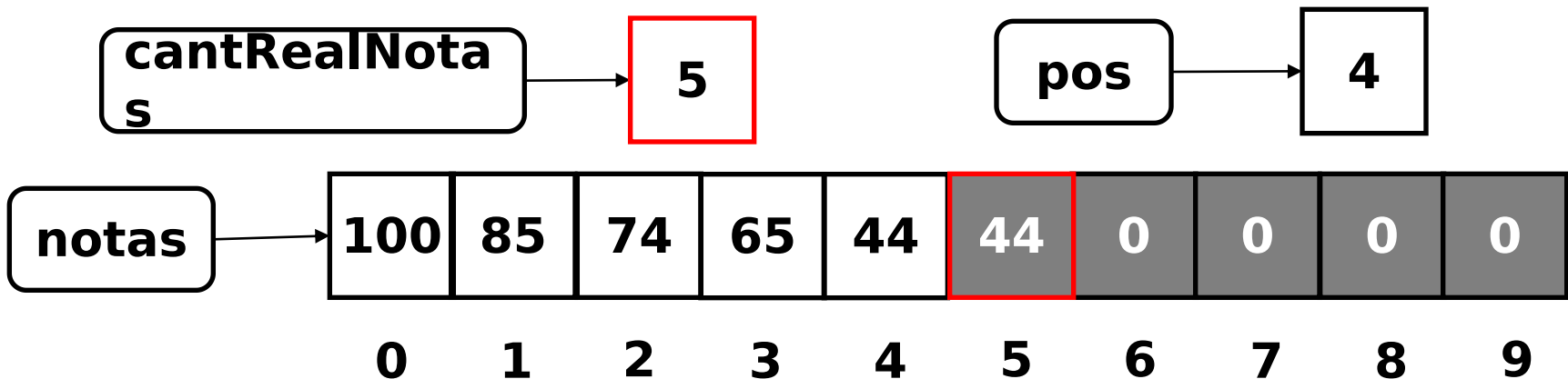
# Algoritmo Eliminar

```
public void EliminarNota(int pos)
{
    if ((cantRealNotas > 0) && (pos > 0) &&
        (pos <= cantRealNotas))
    {
        for(int i = pos-1; i < cantRealNotas ; i++)
            notas[i] = notas[i+1];
        cantRealNotas --;
    }
}
```



# Algoritmo Eliminar

```
public void EliminarNota(int pos)
{
    if ((cantRealNotas > 0) && (pos > 0) &&
        (pos <= cantRealNotas))
    {
        for(int i = pos-1; i < cantRealNotas ; i++)
            notas[i] = notas[i+1];
        cantRealNotas --;
    }
}
```





# Linguagem de programação I

## Vetores em Java, vetores de objetos

MSc Romanuel Ramón Antunez.  
Isutic, 2013