

Programación Orientada a Objetos

Arreglos

Ing. Eric Gustavo Coronel Castillo

<http://www.facebook.com/gcoronelc>

<http://gcoronelc.blogspot.com>

gcoronelc@gmail.com

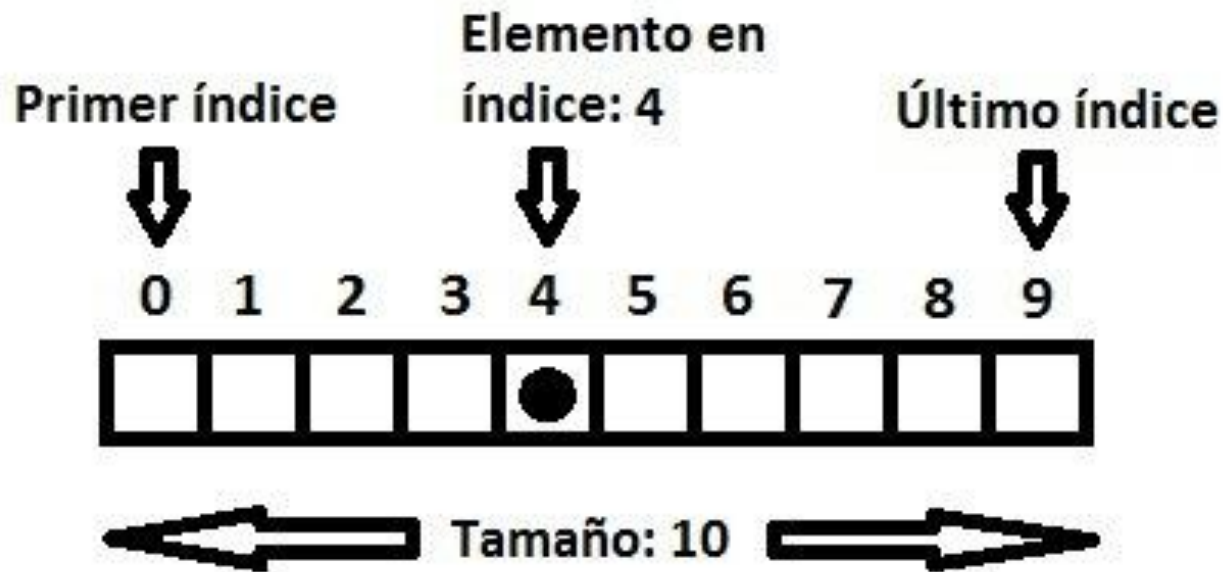
ÍNDICE

- ❖ Objetivos
- ❖ Estructura de Datos
- ❖ Arreglos
- ❖ Creación de Arreglos
- ❖ Resumen



OBJETIVOS

- ❖ Entender el concepto de arreglo.
- ❖ Utilizar arreglos para manejar colecciones de datos.



ESTRUCTURA DE DATOS

❖ Estructura de Datos

- Representa una colección de datos.

❖ Los tipos de datos más frecuentemente utilizados son:

- Datos Simples

- ✓ Entero, Real, Carácter, Lógico.

- Datos Estructurados

- ✓ Estáticos

- Arreglos (array), Registro, Archivo (fichero), Cadena.

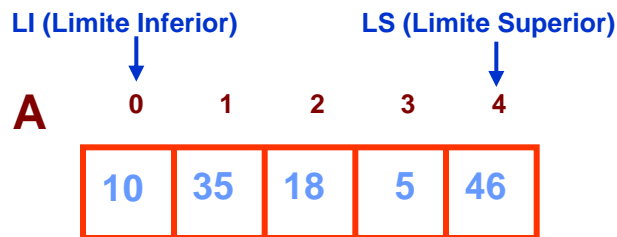
- ✓ Dinámicos

- Lista (pila/cola), Lista enlazada, Árbol, Grafo.

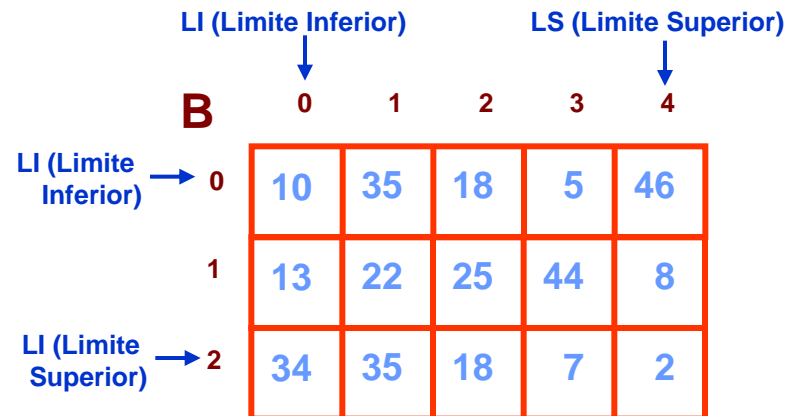
ARREGLOS

- ❖ Es un conjunto de datos homogéneos, es decir del mismo tipo, agrupados con un nombre y diferenciados por un índice (posición)
- ❖ Por lo general en muchos lenguajes de programación, la primera posición del arreglo inicia con 0, tales es el caso en Java.
- ❖ Una vez definida la dimensión y el tamaño del arreglo ya no puede ser modificarlo.

Arreglo Unidimensional (Vector)



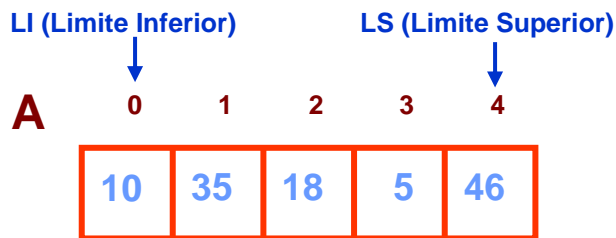
Arreglo Bidimensional (Matriz)



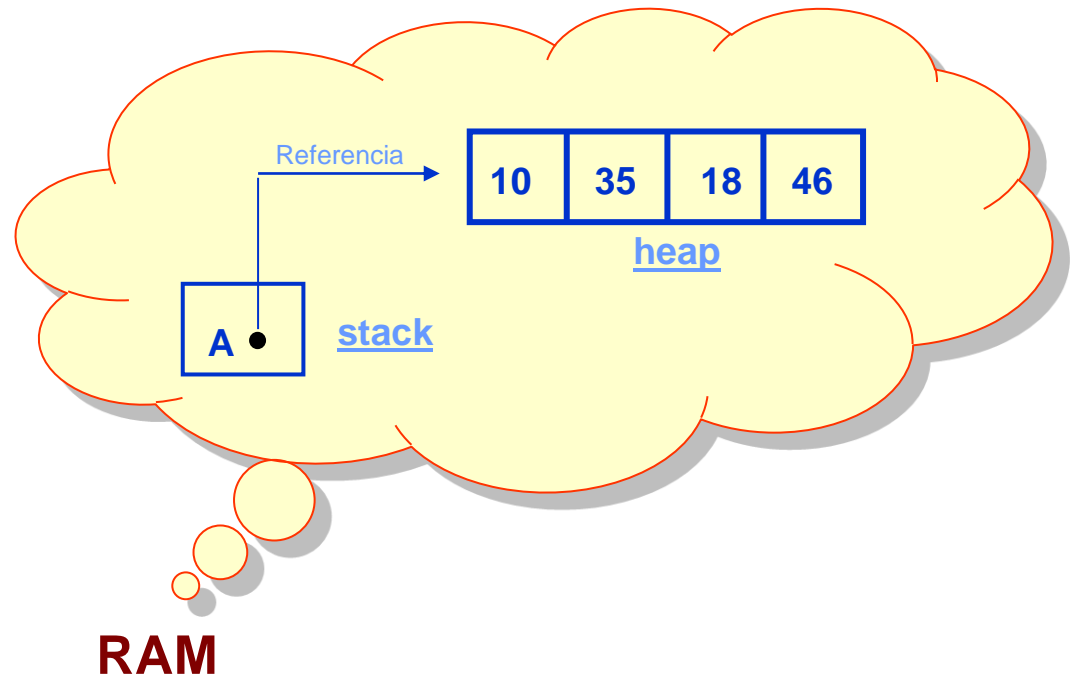
ARREGLOS

- ❖ Los arreglos (array) en Java es un tipo de clase especial que hereda en forma implícita de Object (java.lang.Object).
- ❖ En un arreglo puede almacenar datos primitivos o complejos (datos referenciados)

Arreglo Unidimensional (Vector)



```
int A[] = new int[5];  
A[0] = 10;  
A[1] = 35;  
A[2] = 18;  
A[3] = 5;  
A[4] = 46;
```

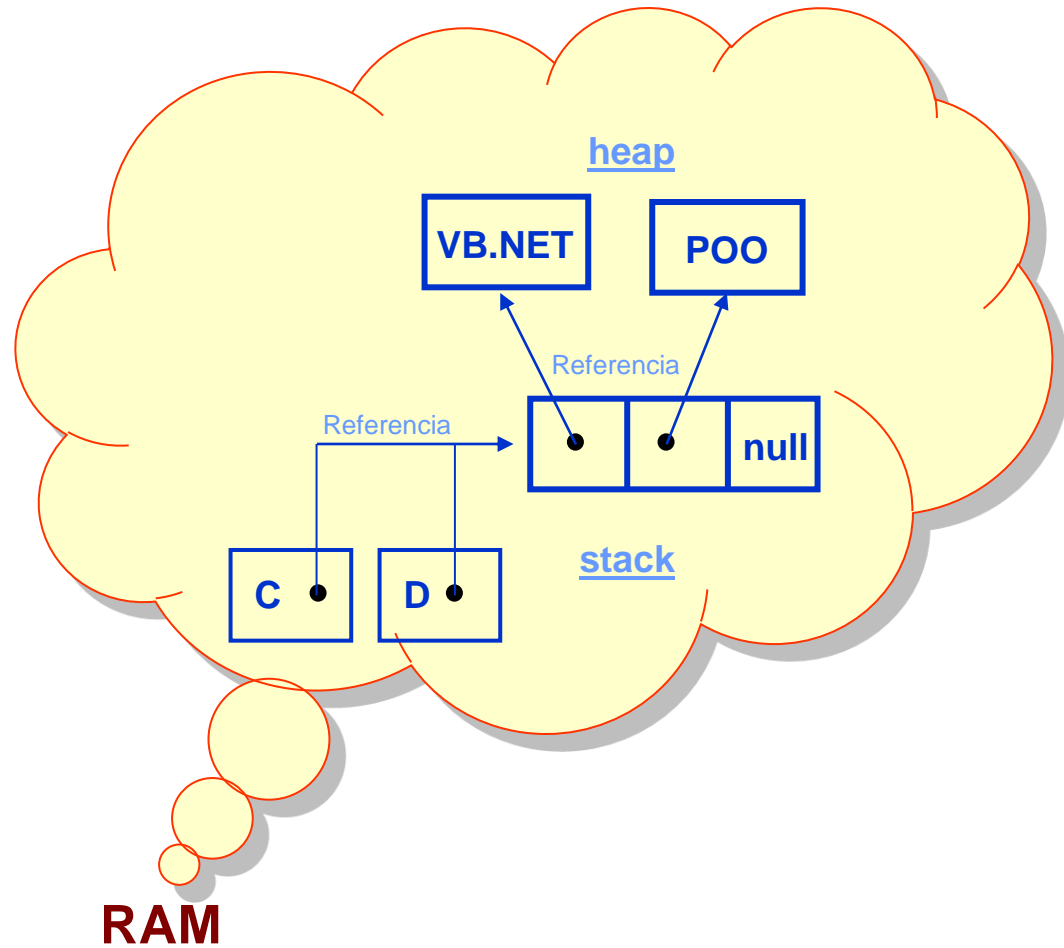


ARREGLOS

C	0	1	2
	Java	POO	null

```
String C[]=new String[3];  
C[0] = new String("Java");  
C[1] = new String("POO");
```

```
//Referencia  
String D[]=C;  
D[0] = "VB.NET";
```



CREACIÓN DE ARREGLOS

// VECTOR

// Forma 1

```
int A[];
```

```
A = new int[5];
```

// Forma 2

```
int A[]=new int[5];
```

//Forma 3

```
int A[] = {10,35,18,5,46};
```

// Forma 4

```
int A[];
```

```
A = new int[]{10,35,18,5,46};
```

// MATRIZ

// Forma 1

```
int B[][];
```

```
B = new int[3][5];
```

// Forma 2

```
int B[][] = new int[3][5];
```

// Forma 3

```
int B[][]={{10,35,18,5,46},  
           {13,22,25,44,8},  
           {34,35,18,7,2}};
```


RECORRIDO POR ARREGLOS

//VECTOR

//Sumar los elementos

```
int S=0;
```

//Formato indexada

```
for(int i=0;i<=4;i++)
```

```
    S += A[i];
```

//Formato colecciones

```
for(int n: A)
```

```
    S += n;
```

//MATRIZ

//Sumar los elementos

```
int S=0;
```

//Formato indexada

```
for(int i=0;i<=2;i++)
```

```
    for(int j=0;j<=4;j++)
```

```
        S += B[i][j];
```

//Formato usando length

```
for(int i=0;i<B.length;i++)
```

```
    for(int j=0;j<B[i].length;j++)
```

```
        S += B[i][j];
```

RESUMEN

- ❖ Para almacenar una colección de datos, cree arreglos de una dimensión (vector) o de dos dimensiones (matrices).
- ❖ Los arreglos son datos referenciados (memoria **stack**) y datos almacenados (memoria **heap**)
- ❖ Para realizar un recorrido por los valores de los arreglos use la estructura de control **for** en su formato indexado y formato de colecciones.

BIBLIOGRAFÍA

