

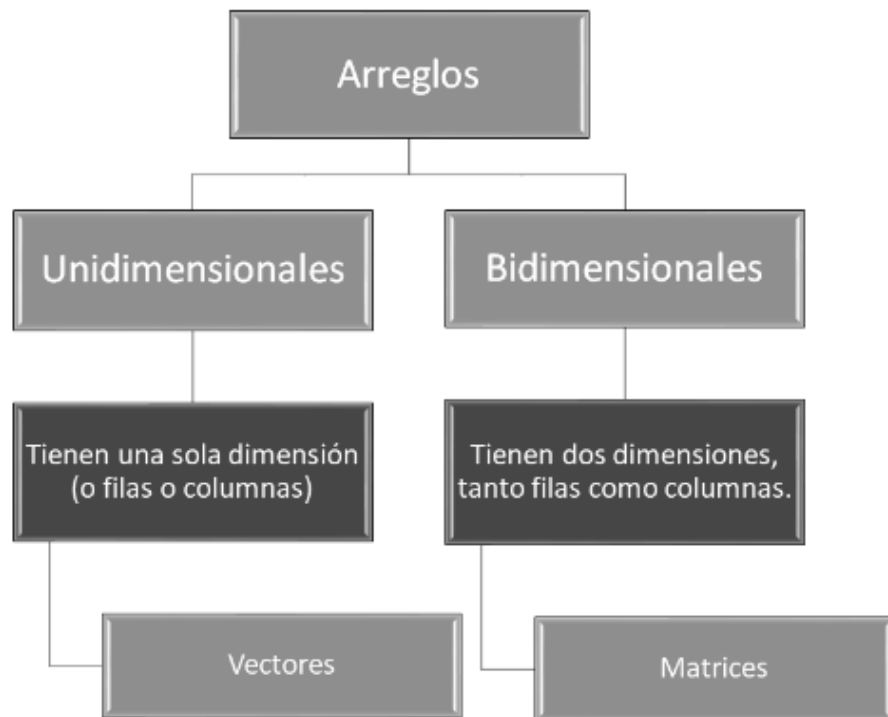


Vectores y Matrices

Arrays (Arreglos)

- Son un conjunto de datos que se almacenan en memoria de forma contigua bajo un mismo nombre pero con diferentes índices para identificar a cada uno de ellos.
- Son estructuras estáticas. Una vez que se declaran e inicializan tienen que mantener su tamaño durante toda la ejecución de un programa.
- Cada arreglo puede ser declarado con **un solo tipo de dato**, por lo que solo puede almacenar datos de ese tipo.

Tipos de Arreglos



Vectores

- Los vectores son arrays unidimensionales que se caracterizan por solo tener o filas o columnas.
- Cada posición posee un **índice** que permite identificarla. La numeración de los índices arranca en 0 y termina en el tamaño del array - 1. (Por ejemplo: 6 - 1).

ÍNDICES					
0	1	2	3	4	5
2	32	45	68	21	123

Vectores (Arrays unidimensionales)

// Declaración e inicialización

```
{ } int numeros[] = new int [6]; // declaro e inicializo un array de 6 posiciones
```

// Carga (manual)

```
{ } numeros[0] = 2;  
numeros[1] = 32;  
numeros[2] = 45;  
numeros[3] = 68;  
numeros[4] = 21;  
numeros[5] = 123;
```

// Recorrido

```
{ } //recorremos un vector y mostramos que hay en cada posición  
for (int i=0; i<numeros.length; i++) {  
    System.out.println("Estoy en el índice: " + i);  
    System.out.println("Tengo guardado un " + numeros [i]);  
}
```

// Carga por teclado

```
{ } Scanner tecla = new Scanner (System.in);  
  
//recorremos cargamos cada posición por teclado  
for (int i=0; i<numeros.length; i++) {  
    System.out.println("Ingrese el valor para la posición: " + i);  
    numeros[i] = tecla.nextInt();  
}
```

Matrices

Las matrices son arreglos bidimensionales que se caracterizan por tener dos índices. Uno para manejar las filas y otro para manejar las columnas.

A 3x3 matrix is shown with teal cells. Above the matrix, an orange bracket labeled 'ÍNDICES COLUMNAS' spans the columns, with indices 0, 1, and 2 below it. To the left of the matrix, an orange bracket labeled 'ÍNDICES FILAS' spans the rows, with indices 0, 1, and 2 to its left.

	0	1	2
0	2	32	45
1	68	21	123
2	5	77	69

// Declaración e inicialización

{ }

```
int matriz[][] = new int [3][3]; // declaro e inicializo una matriz de  
3 filas y 3 columnas.
```

// Carga (manual)

{ }

```
numeros[0][0] = 52;  
numeros[0][1] = 133;  
numeros[0][2] = 24;  
...  
numeros[2][2] = 86;
```

// Recorrido

```
//recorremos una y mostramos que hay en cada posición
//para las matrices usamos dos for, uno para filas y otro para columnas
for (int f=0; f<=2; f++) {
    for (int c=0; c<=2; c++) {
        System.out.println("Fila: " + f + "Columna: " + c);
        System.out.println("Tengo guardado un " + matriz[f][c]);
    }
}
```

// Carga por teclado

```
Scanner tecla = new Scanner (System.in);

//recorremos y cargamos la matriz al mismo tiempo
for (int f=0; f<=2; f++) {
    for (int c=0; c<=2; c++) {
        System.out.println("Ingrese el valor para la posición: " + i);
        matriz[f][c] = tecla.nextInt();
    }
}
```

Errores por desbordamiento

Los errores por desbordamiento se producen cuando intentamos hacer referencia a una posición en nuestro array (sea vector o matriz) que no existe o que está fuera de los límites del tamaño que declaramos. Por ejemplo: