# PROGRAMACIÓN

# UT3: Arrays unidimensionales (III)



## 1. ¿Qué es un array?

Una matriz o array es una estructura de datos indexada que contiene una colección de valores del mismo tipo a las que se hace referencia por medio de un nombre común.

Sirven para almacenar valores que normalmente tienen una relación entre sí.

¿Por qué son necesarios?

Normalmente nos vamos a encontrar con la necesidad de almacenar más de un valor, como si fuese una lista o una tabla, por ejemplo, el nombre de los alumnos de una clase, las calificaciones de una asignatura de un alumno, los meses de un año, …

Con el uso de arrays unidimensionales solucionamos ese problema, ya que nos permiten almacenar todos estos valores en una sola variable.

## 2. Estructura de un array

En los arrays encontramos dos atributos:

- El índice; nos indica la posición del array.
- El elemento; nos indica qué contiene (el valor) el array en una determinada posición.

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| “Enero” | “Febero” | “Marzo” | “Abril” | “Mayo” | “Junio” |

| ÍNDICE |
|---|
| ELEMENTO |

NOTA: ¡La primera posición de un array siempre es el índice 0!

Los arrays pueden ser de cualquier tipo, pero TODOS los elementos deben ser del mismo.

Es decir, si un array es de enteros, todos los elementos deben ser enteros.

## 3. Declaración e inicialización de un array

### DECLARACIÓN

Para declarar un array en java seguiremos la siguiente estructura, pero debemos tener en cuenta que los arrays son “listas” estáticas, por lo que **siempre debemos indicarles cual va a ser su longitud al crearlos.**

NO SE PUEDEN AÑADIR MÁS ELEMENTOS DE LOS QUE TENGA EL ARRAY.

<tipo_datos> [ ] <nombre_variable> = new <tipo_datos> [<longitud>]

Con [ ] (los corchetes) indicamos que la variable va a ser un array.

Ejemplo:

```
String [ ] arrray_Strings = new String [10]; // Array de 10 posiciones

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

### INICIALIZACIÓN

Al crear el array, sino le asignamos valores se inicializa a unos valores predeterminados en función del tipo del array.

- Variables numéricas a **cero**.
- Variables de caracteres a **\u0000**.
- Variables boleanas a **false**.
- Objetos a **null**.

Tenemos dos formas de inicializar un array, ir uno a uno dando valor a sus elementos, o inicializarlos todos a la vez.

Ejemplo:

String [ ] arrray_Strings = new String [12]

array_Strings[0] = “Enero”;

array_Strings[1] = “Febrero”;

array_Strings[2] = “Marzo”;

…

array_Strings[11] = “Diciembre”;

Ejemplo2:

int [ ] array_Enteros = new int [3];

array_Enteros[0] = 4;

array_Enteros[1] = 32;

array_Enteros[2] = -3;

Ejemplo3: // Declaramos el array de golpe:

int [ ] array_Enteros = new int [3];

array_Enteros = {4, 32, -3};

## 4. Uso de un array

Podemos guardar el valor de cada posición del array en una variable para su posterior uso.

Del mismo modo que le damos valor, lo solicitamos:

Ejemplo:

```
int num = array_Enteros [2];
```

Otra función útil de los arrays es conocer por código su longitud.

Esto es posible usando "**length**"

Si escribimos

```
int longitud = array_Enteros.length
```

La variable longitud almacenará un 3.

Esto es muy útil cuando queremos recorrer un array entero, con el uso del bucle for.

Ejemplo:

```
for (int i = 0 ; i < EjemploArray.length ; i++){
    System.out.println( EjemploArray[i] );
}
```