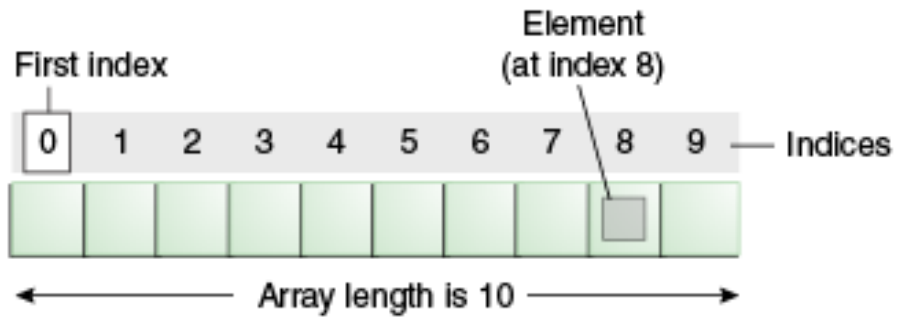


Fundamentos de Java

Arrays y Loops

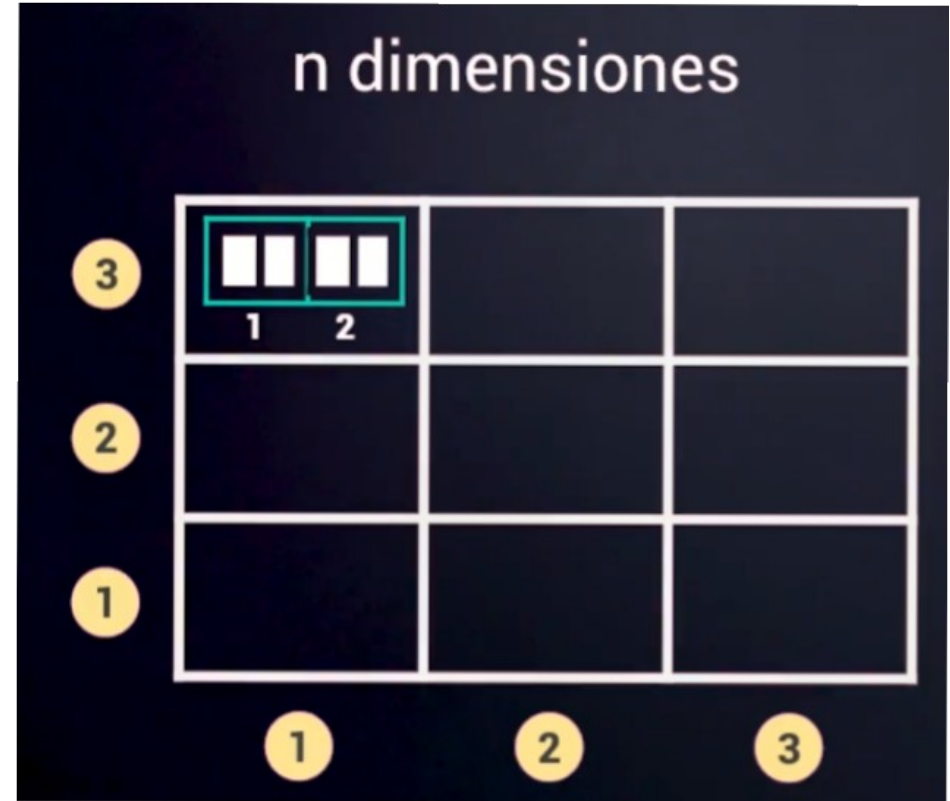
Arreglos

Un arreglo es una variable que nos permite almacenar mas de un elemento.



Arreglos multidimensionales

Los arrays son de una sola dimensión, pero si guardamos arrays dentro de otros arrays podemos obtener arrays multidimensionales.



Definiendo arreglos

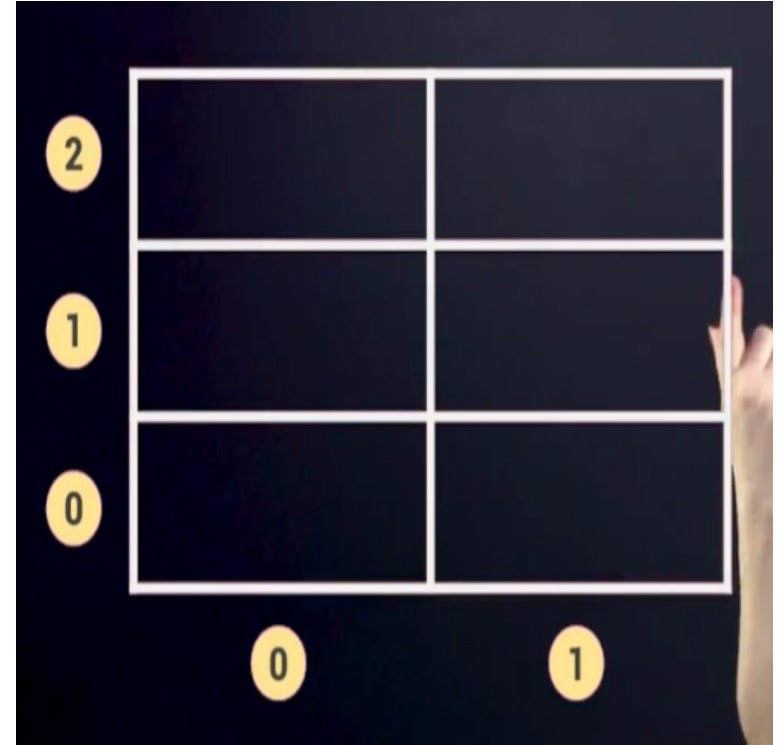
```
// 1. Definir el nombre de la variable y el tipo de datos
// que va a contener, cualquiera de las siguientes dos
// opciones es válida:
TipoDato[] nombreVariable;
TipoDato nombreVariable[];

// 2. Definir el tamaño del array, la cantidad de elementos
// que podemos guardar en el array:
TipoDato[] nombreVariable = new TipoDato[capacidad];

// Array de dos dimensiones:
TipoDato[][] cities = new String[númeroFilas][númeroColumnas];
```

Indices y búsquedas en los Arrays

Los índices son variables simples que nos ayudan a identificar las posiciones en un arreglo. Estas variables siempre guardan números, comienzan en 0 e incrementan de abajo a arriba y de izquierda a derecha a medida que guardamos más elementos en nuestros arrays.

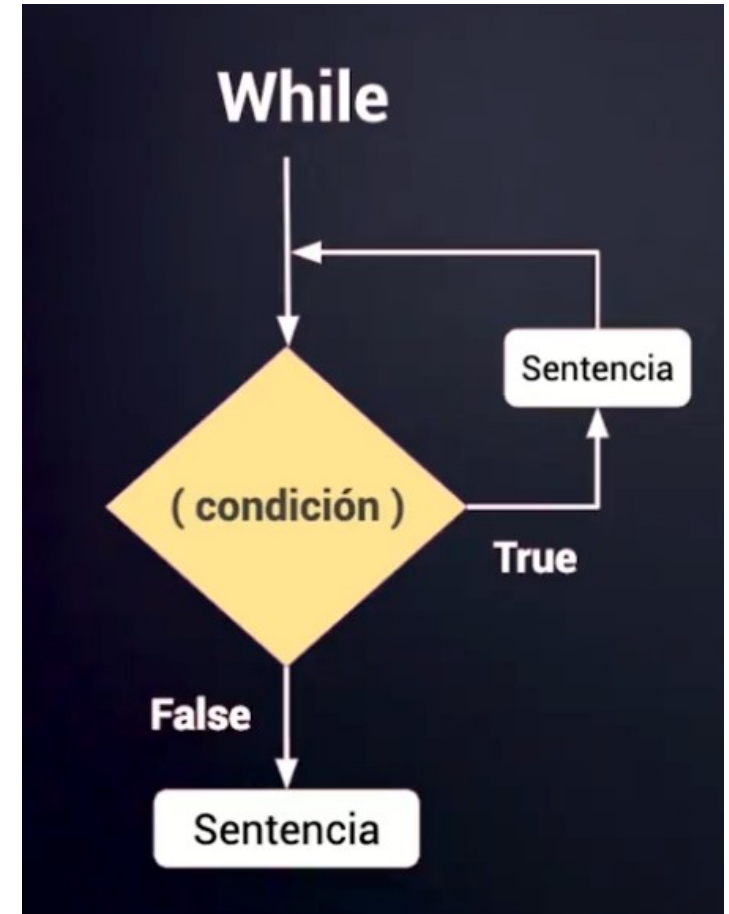


Indices y búsquedas en los Arrays



Bucle While

Los bucles (ciclos) nos ayudan a ejecutar una parte de nuestro código una cantidad de veces mientras que se cumpla alguna condición y podemos continuar con la ejecución de nuestro código.



Do While

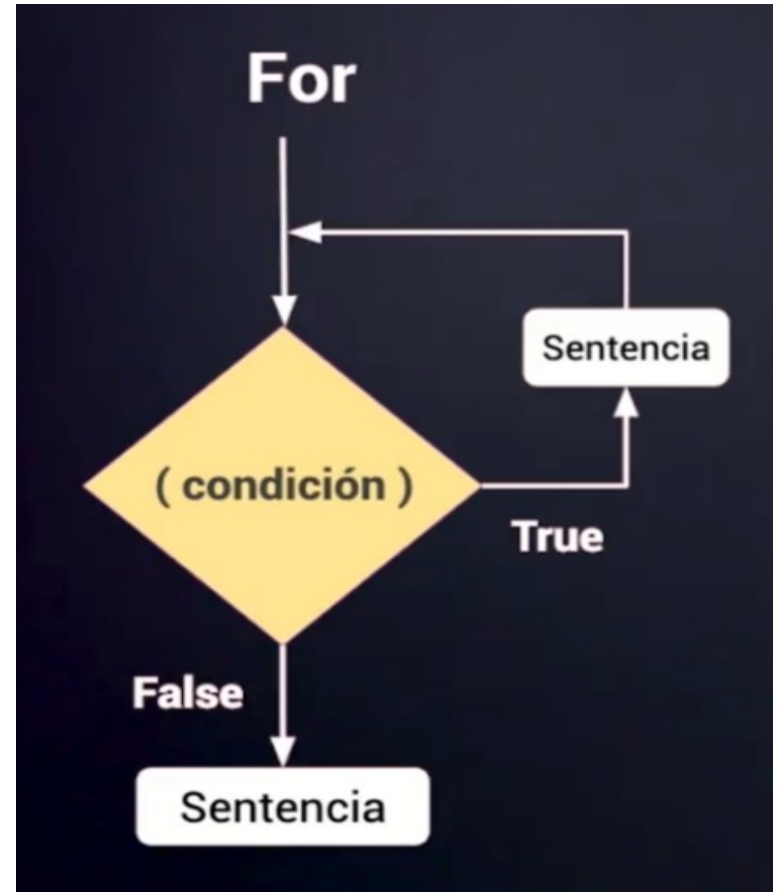
```
do {  
  
    //instrucciones  
  
} while (condicion);
```


While

```
while (condicion) {  
    //instrucciones  
}
```

Ciclos For

Los ciclos FOR ejecutan una parte de nuestro código todas las veces que sean necesarias hasta que una condición se cumpla, por ejemplo, que un número supere o iguale cierta cantidad.



For

```
for(inicializacion; condicion; incremento){  
    //instrucciones  
}
```

Break, Continue y Return

- Break: Un break es capaz de detener el flujo en el código, en este caso detendremos el ciclo como tal terminándolo y haciendo que saltemos a la siguiente instrucción después del ciclo.
- Continue: En cierto modo también nos va a servir para detener un ciclo pero en lugar de terminarlo como en el caso de break, este volverá directo a la condición.
- Return: En Java es para salir de Funciones, cuando lo usamos aquí siempre viene acompañado de un valor, el cuál indica el dato que se estará devolviendo en la función.