



# Array de datos

TypeScript

<https://www.typescriptlang.org/>



Juan Vladimir  
@juanvladimir13

# Agenda

1. Que es un array
2. Características de un array
3. Recorrido de un array de datos primitivos
4. Objetos en typescript
5. Recorrido de array de objetos

# Que es un array

Es una estructura de datos que permite **almacenar múltiples elementos del mismo tipo bajo un mismo nombre**, a los cuales se accede mediante un índice numérico.

## Usos comunes de los array/arreglos:

- ✓ Almacenar grandes conjuntos de datos del mismo tipo, como listas de nombres, resultados de exámenes, etc.
- ✓ Implementar algoritmos que requieren acceso rápido a elementos basados en su posición.
- ✓ Representar matrices y otras estructuras de datos multidimensionales.

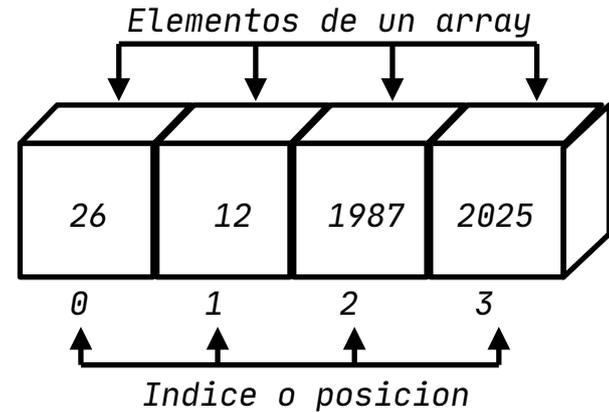
# Características de un array

**Homogeneidad:** Todos los elementos de un arreglo deben ser del mismo tipo de dato.

**Orden:** Los elementos se almacenan en un orden específico.

**Tamaño/dimensión:** En muchos lenguajes de programación, los arrays tienen un tamaño definido al momento de su creación y no pueden cambiar de tamaño durante la ejecución del programa.

**Acceso por índice:** El acceso a un elemento específico se realiza mediante su índice, que generalmente comienza en 0 para el primer elemento.



```
const nombre notas: tipo de dato number[] = [ tamaño / dimension [ 51 , 64 , 75 ] ;  
                                     índice / posición 0     1     2
```

```
let friends:string[] = [ "Steve", "Bill" ] ;
```

# Recorrido de un array de datos primitivos

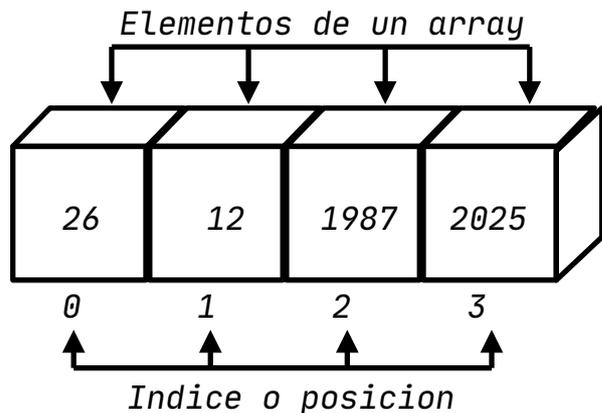


Diagram illustrating the structure of an array declaration. The code is: `const notaAnual = [ 51 , 64 , 75 ];`. Above the variable name `notaAnual`, the text "nombre" is written with a bracket. Above the opening square bracket `[`, the text "tamaño / dimension" is written with a bracket. Below the values `51`, `64`, and `75`, the indices `0`, `1`, and `2` are written with brackets. Below the closing square bracket `]`, the text "indice / posicion" is written with a bracket.

## Estructura de datos while

```
let index = 0;
while (index < notaAnual.length) {
  console.log(notaAnual[index]);
  index++;
}
```

## Estructura de datos for

```
for (let index = 0;
     index < notaAnual.length;
     index++)
{
  console.log(notaAnual[index]);
}
```

# Objetos en typescript

## Definición de un Objeto

```
interface Judagor {  
  nombre: string;  
  edad: number;  
  retirado: boolean;  
}
```

## Creación de un Objeto

```
const cr7: Judagor = {  
  nombre: "Cristiano Ronaldo",  
  edad: 38,  
  retirado: false,  
};
```

## (Setters) Modifica datos del objeto

```
cr7.nombre = "El Bicho";  
cr7.edad = 39;
```

## (Getters) Obtiene datos del objeto

```
const nombre = cr7.nombre;  
const edad = cr7.edad;
```

# Recorrido de array de objetos

## Definición de un interface

```
interface Calificacion {  
  estudiante: string;  
  nota: number;  
}
```

## Creación de un array de objetos

```
const calificaciones: Calificacion[] = [  
  { estudiante: "Juan", nota: 85 },  
  { estudiante: "Ana", nota: 90 },  
  { estudiante: "Luis", nota: 78 },  
];
```

## Recorrido de array utilizando indices

```
for (let index = 0; index < calificaciones.length; index++) {  
  const calificacion = calificaciones[index];  
  console.log(calificacion.estudiante, calificacion.nota);  
}
```

## Recorrido de array utilizando un iterador

```
for (const calificacion of calificaciones) {  
  console.log(calificacion.estudiante, calificacion.nota);  
}
```