

# TypeScript

## *Estructuras de datos*

CertiDevs

# êndice de contenidos

1. Estructuras de datos en TypeScript .....	1
1.1. Arrays .....	1
1.2. Tuplas .....	1
1.3. Objetos .....	1
1.4. Mapas .....	2

# 1. Estructuras de datos en TypeScript

TypeScript admite varias estructuras de datos que facilitan la manipulación y organización de datos en aplicaciones. Algunas de las estructuras de datos más utilizadas en TypeScript son: arrays, tuplas, objetos y mapas. Vamos a analizar cada una de ellas en detalle.

## 1.1. Arrays

Los arrays son colecciones ordenadas de elementos del mismo tipo. En TypeScript, los arrays pueden ser de dos tipos: homogéneos, donde todos los elementos son del mismo tipo, y heterogéneos, donde los elementos pueden ser de diferentes tipos. Los arrays se pueden declarar utilizando corchetes ([]) o utilizando la palabra clave Array.

```
const numbers: number[] = [1, 2, 3, 4, 5];
const names: Array<string> = ['Alice', 'Bob', 'Charlie'];
```

Se puede acceder a los elementos de un array utilizando corchetes y un índice.

```
const numbers: number[] = [1, 2, 3, 4, 5];
const firstNumber = numbers[0]; // 1
```

Los arrays en TypeScript ofrecen una amplia variedad de métodos y propiedades para manipular y acceder a los datos, como length, push, pop, splice, slice, map, filter, reduce y muchos más.

## 1.2. Tuplas

Las tuplas son similares a los arrays, pero con una longitud fija y tipos de elementos específicos. Las tuplas se pueden utilizar para representar una colección ordenada de valores con diferentes tipos de datos. Se pueden declarar utilizando corchetes ([]).

```
const person: [string, number] = ['Alice', 30];
```

Al igual que con los arrays, se puede acceder a los elementos de una tupla utilizando corchetes y un índice.

```
const person: [string, number] = ['Alice', 30];
const name = person[0]; // 'Alice'
const age = person[1]; // 30
```

## 1.3. Objetos

Los objetos en TypeScript son colecciones de propiedades clave-valor, donde las claves son strings y los valores pueden ser de cualquier tipo. Los objetos se pueden declarar utilizando llaves ({}).

```
const person = {  
  name: 'Alice',  
  age: 30,  
};
```

Se puede acceder a las propiedades de un objeto utilizando la notaci3n de puntos o la notaci3n de corchetes.

```
const person = {  
  name: 'Alice',  
  age: 30,  
};  
  
const name = person.name; // 'Alice'  
const age = person['age']; // 30
```

TypeScript tambi3n admite la definici3n de interfaces y tipos para describir la estructura de un objeto.

```
interface Person {  
  name: string;  
  age: number;  
}  
  
const person: Person = {  
  name: 'Alice',  
  age: 30,  
};
```

## 1.4. Mapas

Los mapas en TypeScript son una estructura de datos que permite almacenar pares clave-valor, donde las claves pueden ser de cualquier tipo (no solo strings). Los mapas se pueden declarar utilizando la clase Map.

```
const map = new Map<string, number>();  
map.set('one', 1);  
map.set('two', 2);  
  
console.log(map.get('one')); // 1  
console.log(map.has('two')); // true
```

Los mapas ofrecen varios m3todos y propiedades para manipular y acceder a los datos, como size, get, set, delete, has, keys, values y entries.