**TypeScript**

**Tuplas.**

**Documentación:**

[**https://www.typescriptlang.org/es/play#example/tuples**](https://www.typescriptlang.org/es/play#example/tuples)

[**https://www.typescriptlang.org/docs/handbook/basic-types.html#tuple**](https://www.typescriptlang.org/docs/handbook/basic-types.html#tuple)

[**https://www.typescriptlang.org/docs/handbook/2/objects.html#tuple-types**](https://www.typescriptlang.org/docs/handbook/2/objects.html#tuple-types)

**<https://www.typescriptlang.org/docs/handbook/2/everyday-types.html>**

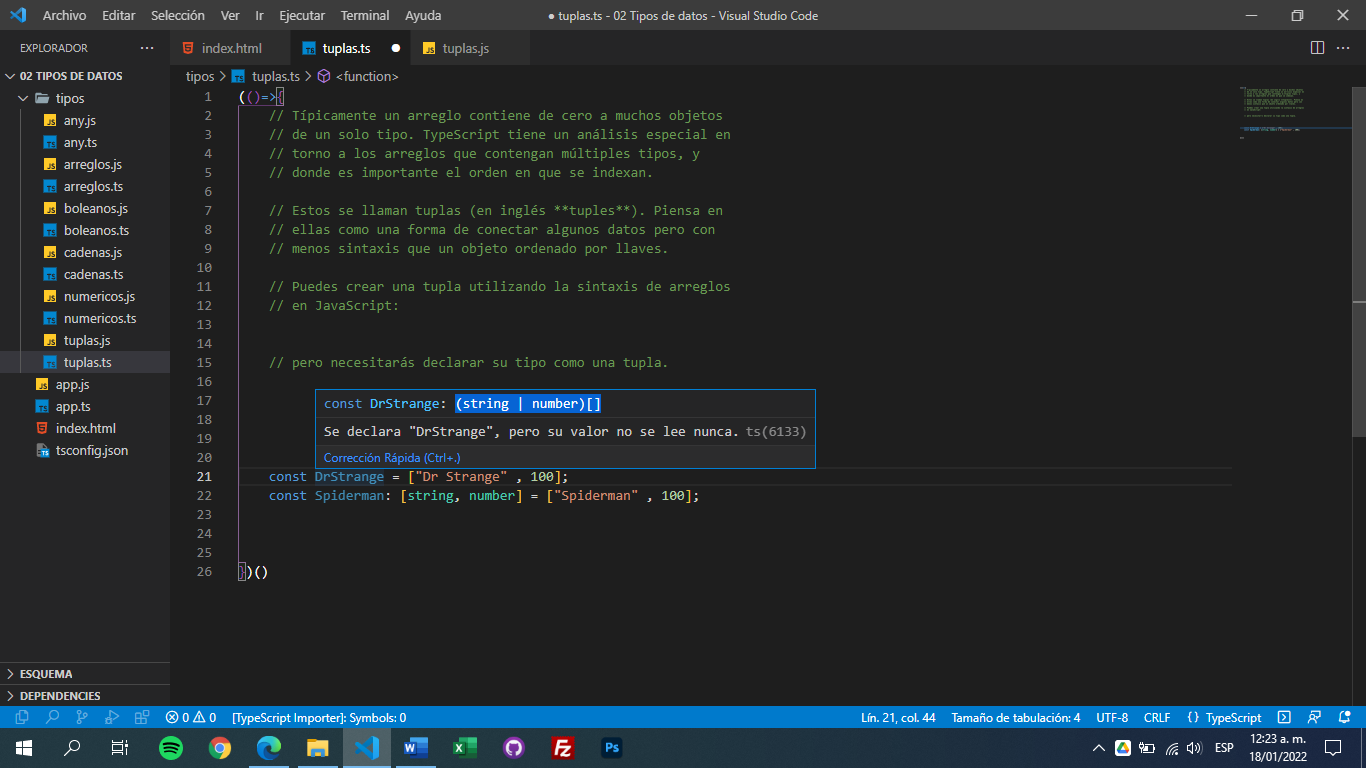
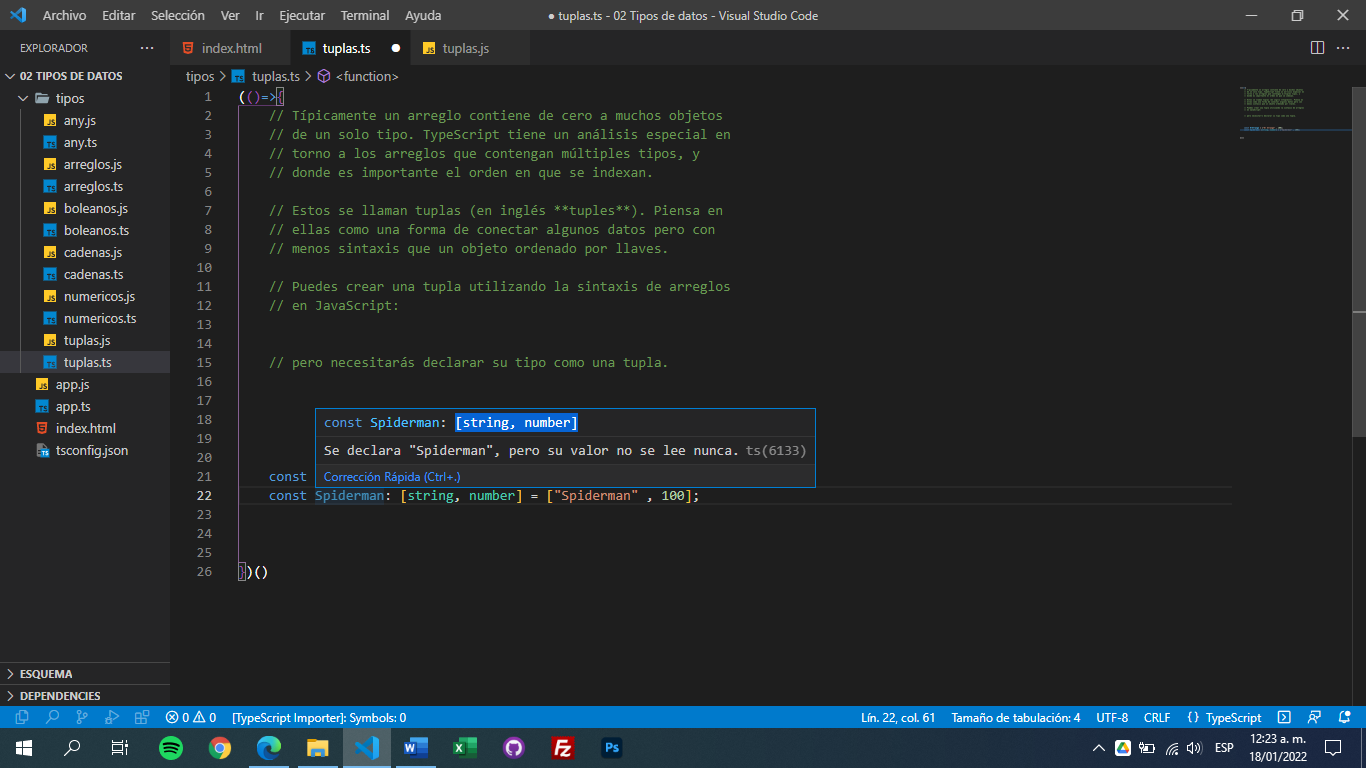
Las tuplas son algo nuevo y propio de TS que no tiene JS. Típicamente un arreglo contiene de cero a muchos objetos de un solo tipo. TS tiene un análisis especial en torno a los arreglos que contengan múltiples tipos y donde es importante el orden en el que se indexan.

A estos se llaman tuplas (tuples). Piensa en ellas como una forma de conectar algunos datos pero con menos sintaxis que un objeto ordenado por llaves. Puedes crear una tupla utilizando la sintaxis de arreglos. Ejemplo:

    const DrStrange = ["Dr Strange" , 100];

pero necesitarás declarar su tipo como una tupla. Ejemplo:

    const Spiderman: [string, number] = ["Spiderman" , 100];

Al pasar el cursor sobre los nombres de ambas variables puedes ver la diferencia entre la variable de tipo arreglo **( (string | number)[] ) y la tupla ( [string, number] ).**

**En un arreglo, el orden no es importante por lo que un elemento en cualquier índice puede ser una cadena de texto o un número. En la tupla, el orden y la longitud son garantizados. Esto significa que TS proporcionará el tipo correcto en el índice adecuado e incluso generará un error si intenta acceder a un objeto en un índice no declarado. Ejemplo:**

const DrStrange = ["Dr Strange" , 100];

const Spiderman: [string, number] = ["Spiderman" , 100];

DrStrange[0] = 100;

DrStrange[1] = "Dr Strange";

Spiderman[0] = 100;         // Error: El tipo 'number' no se puede asignar al tipo 'string'

Spiderman[1] = "Spiderman"; // Error: El tipo 'string' no se puede asignar al tipo 'number'