

# TypeScript

## 1) Generalización simbólica: ¿Cuáles son las reglas escritas del lenguaje?

### Variables y tipos

TypeScript permite declarar variables con `let`, `const` y `var`

Tipos primitivos: `number`, `string`, `boolean`, `null`, `undefined`.

Tipos compuestos (estructurados): `arrays`, `tuples`, `objects`.

Tipos derivados: `type` y `interface`

### Estructura de control

Secuenciales: ejecución de instrucciones en orden.

Condicionales: `if`, `else if`, `else`, `switch`

Bucles: `for`, `while`, `do...while`

### Funciones

- Declaración de funciones como bloques autocontenidos que reciben parámetros y retornan valores.
- No se consideran objetos ni métodos de clases (porque nos limitamos a programación estructurada).

## Expresiones y operadores

- Operadores aritméticos (+, -, \*, /, %).
- Operadores lógicos (&&, ||, !) y relacionales (==, !=, <, >).
- Precedencia de operaciones y reglas de combinación de tipos.

## 2) Creencias de los profesionales: ¿Qué características particulares del lenguaje se cree que sean "mejores" que en otros lenguajes?

### Seguridad de tipos

TypeScript ofrece tipado estático, lo que permite detectar errores de tipo en tiempo de compilación.

Creencia: Esto reduce bugs y hace el código más robusto y predecible.

## Autocomplete y documentación

Herramientas como VSCode aprovechan los tipos para ofrecer autocompletado y validación de parámetros.

Creencia: Esto acelera el desarrollo y reduce errores de sintaxis y de lógica.

Mejor mantenimiento y escalabilidad.

Compatibilidad con ecosistema JS → flexibilidad sin sacrificar robustez.

Claridad en estructuras → flujo de datos más comprensible.

Disciplina estructurada → código más predecible y depurable.