



Avanzados y Funcione...



Curso de Fundamentos de TypeScript



→ CONFIGURADO NU...

TypeScript vs. JavaScript 2/24



RECURSOS

MARCADORES

¿TypeScript es diferente a JavaScript? ¿Un desarrollador en TypeScript es diferente a uno en JavaScript? La respuesta a ambas es sí, sin embargo, no hay una notable diferencia. Uno (TypeScript) se base en el otro (JavaScript) añadiendo elementos para mejorar la detección de bugs y experiencia de desarrollo.

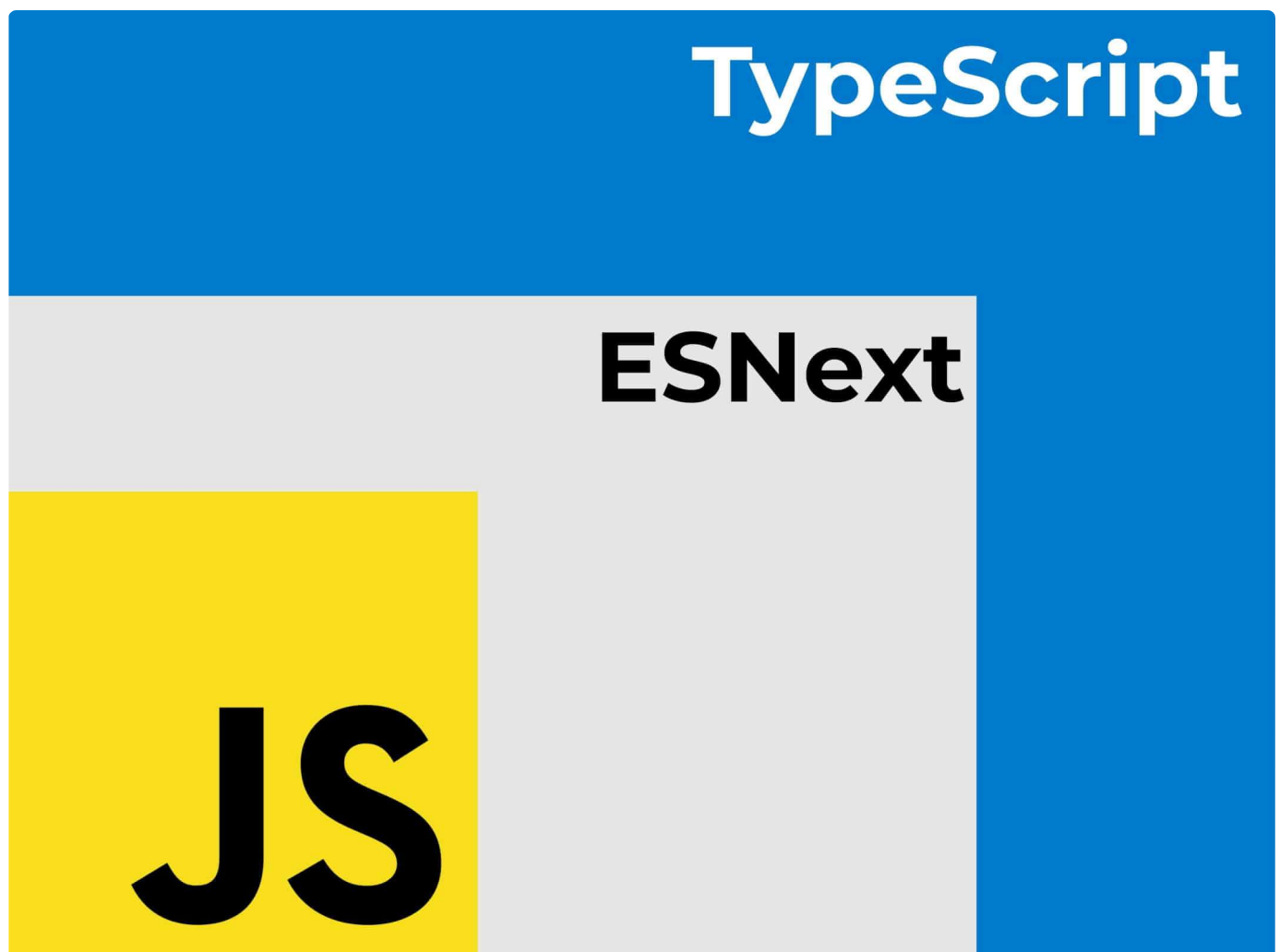
## Panorama

JavaScript ha sufrido un incremento exponencial en su uso, pues se puede usar en Frontend, Backend, IoT, entre otros. No obstante, este no fue creado como un lenguaje maduro desde el inicio, fue con el tiempo que ha ido mejorando hasta lo que es hoy en día.

En JavaScript solo te das cuenta de que tienes un error hasta el momento en que lo ejecutas, sea en el navegador o en un entorno de ejecución como NodeJS, más no antes. Lo que queremos como desarrolladores es obtener retroalimentación lo antes posible para tener la menor cantidad de errores en producción

## El aporte de TypeScript

TypeScript abarca todo lo que tiene JavaScript, más las nuevas versiones de ECMAScript, y añade análisis estático a nuestro código.



## ¿Qué significa análisis de código estático?

*Entre más rápido encuentres un error, más fácil será solucionarlo*

En el libro *Software Engineering at Google*[1] señalan ciertas capas para detectar fallas en el desarrollo de programas:

1. **Análisis de código estático:** corre en el editor de código en busca de un typo (error en la escritura de un término), llamadas incorrectas a funciones y brinda autocompletado de código
2. **Pruebas Unitarias (Unit Tests):** se realiza pruebas para verificar si una parte del código hace lo que queremos que ejecute
3. **Pruebas de Integración (Integration Tests):** vemos como todo el código funciona en conjunto y que se ejecute cómo deseamos
4. **Revisión de código (Code Review):** se verifica si se ha seguido con las normas, estándares y mejores prácticas establecidas por el equipo

[1] Software Engineering at Google. Lessons Learned from Programming Over Time - Titus Winters, Tom Manshreck y Hyrum Wright.

*Contribución creada por: Martín Álvarez.*

### Lecturas recomendadas

What I learned from Software Engineering at Google | Swizec Teller  
<https://swizec.com/blog/what-i-learned-from-software-engineering-at-google>



### Clases relacionadas



Conceptos básicos de TypeScript para usar

Conceptos básicos de TypeScript para usar Angular



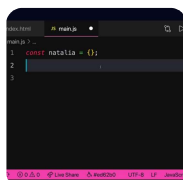
Curso de Fundamentos de Angular



Uso con TypeScript



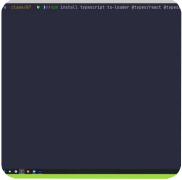
Curso de Vite.js



Objetos literales y prototipos en JavaScript



Curso Básico de Programación Orientada a Objetos con Ja...



## Integración con TypeScript



Curso Práctico de Webpack



## Repaso a TypeScript: tipos y POO



Curso de Backend con NestJS