

*Universidad Tecnológica de la*

*Riviera Maya*

**Desarrollo de Aplicaciones III**

**Profesor:**

*Lic. Julián Villegas*

**Autor(es):**

*Domínguez Juárez Juan Antonio*

*Solidaridad, Quintana Roo, 26 de marzo del 2018*

**Carrera:**

*TSU en Tecnologías de la Información y Comunicación Área Sistemas Informáticos (TI51)*

**Tabla de contenido**

[**Definición** 2](#_Toc509876460)

[**Ejemplo** 3](#_Toc509876461)

[**Fuentes de Información** 5](#_Toc509876462)

## **Definición**

TypeScript es un lenguaje de programación de alto nivel que implementa muchos de los mecanismos más habituales de la programación orientada a objetos, pudiendo extraer grandes beneficios que serán especialmente deseables en aplicaciones grandes, capaces de escalar correctamente durante todo su tiempo de mantenimiento.

La característica fundamental de TypeScript es que compila en Javascript nativo, por lo que se puede usar en todo proyecto donde se esté usando Javascript.

TypeScript proporciona autocompletado avanzado, navegación y refactorización. Tener tales herramientas es casi un requisito para grandes proyectos. Sin ellos, el miedo a cambiar el código pone la base del código en un estado de semi-lectura solamente, y hace que las refactorizaciones a gran escala sean muy arriesgadas y costosas.

TypeScript no es el único lenguaje escrito que se compila en JavaScript. Hay otros lenguajes con sistemas de tipo más fuertes que en teoría pueden proporcionar herramientas absolutamente fenomenales. Pero en la práctica, la mayoría de ellos no tienen nada más que un compilador. Esto se debe a la creación de herramientas de desarrollo enriquecidas tiene que ser un objetivo explícito desde el primer día, que ha sido para el equipo de TypeScript. Es por eso que crearon servicios de lenguaje que pueden ser utilizados por los editores para proporcionar verificación de tipos y autocompletado. Si se ha preguntado por qué hay tantos editores con excelentes soportes de TypeScript, la respuesta son los servicios de idiomas.

## **Ejemplo**

Básicamente necesitamos descargar dos programas. El primero es NodeJS, no porque necesitemos desarrollar con Node, sino porque el compilador de TypeScript está desarrollado en NodeJS.

Desde el sitio de NodeJS encontramos las opciones para la instalación en nuestro sistema operativo, por medio de un típico instalador con su asistente (siguiente, siguiente…) Más información en el Manual de NodeJS.

Luego necesitarás el TSC (Command-line TypeScript Compiler), la herramienta que nos permite compilar un archivo TypeScript a Javascript nativo. Este software es el que está realizado con NodeJS y su instalación se realiza vía npm con el siguiente comando:

**npm install -g typescript**

Cómo crear y compilar un archivo TypeScript

El archivo TypeScript lo creas con cualquier editor de texto para programadores, como habitualmente haces con cualquier otro lenguaje. Solo que generalmente usarás la extensión ".ts".

Algo muy importante: Cualquier código Javascript compila en TypeScript. Esto quiere decir que el código que tenemos en Javascript lo podemos usar también perfectamente dentro de Javascript. Por tanto, para trabajar con TS podemos usar perfectamente los conocimientos que tenemos en Javascript, agregando diversas ayudas que nos ofrece TypeScript.

Este sería un posible código TypeScript donde hemos colocado básicamente unos cuantas declaraciones de tipos en variables. Además de una función que se ha declarado devuelve "void" (nada).

*var a:number = 9;*

*a += 4;*

*function mostrar(b:string) :void{*

*console.log(b);*

*}*

*mostrar('hola');*

Para compilarlo usarás el mencionado TSC y lanzarás un sencillo comando que convertirá el código a Javascript nativo.

tsc ejemplo.ts

La herramienta TSC incluye además "watchers" que permiten vigilar cambios en los archivos TS, compilando su código a JS en el instante sin que tengamos que intervenir. Además es habitual que el compilador, se use en conjunto con otras herramientas como Gulp para mejorar aún más el flujo de desarrollo.

Existe asimismo un archivo de configuración llamado "tsconfig.json" en el que podemos indicar todas las opciones que queramos para el compilador (por ejemplo el estándar EcmaScript al que deseamos compilar el código, el tipo de reporte de errores que se desea, las rutas donde colocar los archivos compilados, etc).

## **Fuentes de Información**

<https://desarrolloweb.com/articulos/introduccion-a-typescript.html>

<http://todomvc.com/examples/typescript-angular/#/>

<https://vsavkin.com/writing-angular-2-in-typescript-1fa77c78d8e8>