

# Antecedentes de TypeScript

Creado en 2012 por Steve Luco y Anders Heijlsberg

Microsoft hace el proyecto Strada para corregir los problemas que presentaba JavaScript en aplicaciones a gran escala, debido en gran medida a su falta de tipado estático.

## Características

- Es de código abierto
- Desarrollado por Microsoft
- Superconjunto de JavaScript, contiene todas las características de JavaScript y más

## Ventajas

1. Tiene mucha documentación.
2. Código más entendible.
3. Autocompletamiento de código.
4. puede escribir código orientado a objetos de una manera más coherente.
5. Curva de aprendizaje liviana debido a su fácil estructura.
6. Detecta errores en tiempo de compilación.

## Desventajas

1. Mayor complejidad en la configuración de nuestro proyecto
2. Toma más tiempo para escribir el código

## Frameworks y Marcas que lo utilizan

- Banco Santander
- GitHub
- Slack
- walmart
- JetBrains
- OverLeaf
- Google
- Adobe
- Netflix

## Compilación

Las herramientas web en realidad solo puede entender JavaScript, por cuanto se pasa de TypeScript a JavaScript mediante compilación

1. `tsc <archivo.ts>` (genera archivo js, hay que editar después de compilar)
2. `tsc -w <archivo.js>` (watch, se queda esperando cambios, compila después de save)
3. `tsc --init` (traz archivo json donde están todas las configuraciones y vienen los archivos que hay que compilar)

# Vue.Js

## Antecedentes

Creado por Evan You en 2004.

Se creó para programar más fácil.

Inicialmente fue pensado como una biblioteca personal.

## Características

- Open Source
- Accesible
- Versátil
- Gran Comunidad
- Se encapsula código reutilizable
- Cuenta con templates

## Componentes

Templates, script y estilo

## Ventajas

- Se parece a angular
- Documentación detallada
- Adaptabilidad
- Excelente integración
- Muy buena para aescalar
- Tamaño muy pequeño (más pequeño que Angular y React.js)

## Desventajas

- Falta de recursos
- Riesgo de excesiva flexibilidad. En proyectos grandes.
- Falta de documentación, toda está en inglés

## Plugins importantes

- Vue Router: gestor de rutas

- Vuefire: conectar con base de datos
- Vue-axios: conexión con REST APIs
- Vuex: gestor de componentes

## Primeros pasos

Para crear un proyecto ejecutamos  
 -vue create [nombre del proyecto]

## Estructura

- public: html y los componentes se agregan allí.
- assets: imágenes, fuentes...
- components: aquí se crean los componentes, según el grupo expositor la forma más versátil de crear componentes es con el simple file component
- router: especifica las rutas para mostrar todas las vistas.
- En el App.vue se tienen todos los componentes. Los componentes pueden ser utilizados con parámetros que se denominan props.
- En router-view dentro de App.vue se coloca la vista y route-to funciona como links entre puntos de la aplicación.
- main.ts: corresponde al archivo donde se conserva el punto de entrada

