

Instituto Tecnológico de Costa Rica  
Introducción al desarrollo de páginas web  
IC-8057

## **VueJs y TypeScript**

Bryan Brenes Rojas  
Lery Sánchez Calderón  
Marvin Castro Roldán  
Stephanie Delgado

Prof.: Ericka Solano Fernández

Verano 2019 - 2020  
15 enero 2019

# Tabla de contenido

<b>Typescript</b>	<b>3</b>
Algunos avances	3
Características	4
Ventajas	4
Desventajas	4
Plataformas en la que se utiliza	4
Mecanismos de instalación	5
Aspectos relevantes de su uso	5
Compilador	6
Especificación de tipos	7
<b>Vue.Js</b>	<b>8</b>
Historial de versiones	8
Características	9
Ventajas	9
Desventajas	9
Plugins importantes	10
Páginas con vue js	10
Ejemplos	10
Ejemplo #1	10
Ejemplo #2	12
Mecanismos de instalación	15
Versión independiente	15
Versión NPM	16
Versión CLI	16
<b>Referencias</b>	<b>17</b>

# Typescript

Uno de los lenguajes favoritos en la creación de aplicaciones es JavaScript, y aunque en la actualidad se presenta como un lenguaje muy evolucionado, en el pasado tuvo grandes dificultades para ser utilizado en bases de códigos grandes, aplicaciones de gran escala y proyectos con muchos años de desarrollo, debido a que carecía de conceptos de programación como clases y módulos, y por estas carencias no tenía herramientas para optimizar el flujo de desarrollo.

Al inicios, Microsoft tuvo la necesidad de realizar una mejora a JavaScript, para el desarrollo de productos como Bing y Office 365, así como otros productos escalables, por lo que Microsoft implementó un proyecto conocido en sus orígenes como Strada, en donde se tenía como objetivo dar solución al problema de JavaScript en el desarrollo de aplicaciones a gran escala, por lo que Steve Luco en conjunto con un equipo de alrededor de 50 personas, incluido Anders Hejlsberg quien era arquitecto en C#, y creador de Delphi y Turbo Pascal, luego de 2 años de desarrollo logran implementar TypeScript en el año 2012.

## Algunos avances

- En el 2013 fue posible que editores de texto como Netbeans y Sublime Text lo soportará.
- En el 2014, se escribió un nuevo compilador de TypeScript que fue 5 veces más veloz.
- En el 2015 se añade soporte para módulos ES6.
- El 16 de Septiembre del 2015, el framework React soporta TypeScript.
- El 14 de Setiembre del 2016, el framework Angular soporta TypeScript.

## Características

- Es un lenguaje de programación libre y de código abierto.
- Desarrollado y mantenido por Microsoft.
- Es un superconjunto de Javascript.
- Añade tipado estático y objetos basados en clases.

- Sirve tanto para el cliente como para el servidor.
- Extiende Javascript (cualquier código Javascript funciona en Typescript, ya que un compilador de TypeScript se traduce a Javascript original).
- Tiene genéricos y decoradores.

## Ventajas

- Cuenta con muchas herramientas y una gran comunidad.
- Hace que el código se entienda mejor.
- Tiene finalización de código y IntelliSense, que ayuda a completar el código.
- Puede escribir código orientado a objetos de una manera más fácil.
- Tiene una curva de aprendizaje liviana, debido a que tiene una sintaxis basada en JavaScript.
- Detecta errores en tiempo de compilación.

## Desventajas

- Tiene mayor complejidad en la configuración de nuestro proyecto.
- Toma más tiempo y dedicación para escribir el código.

## Plataformas en la que se utiliza

Typescript como se mencionó en secciones pasadas es un superconjunto de javascript lo que quiere decir que está basado en javascript y por ende, cualquier código de javascript es válido en Typescript, también se debe tener presente que los navegadores no comprenden typescript y por esto, todo código escrito en este lenguaje debe ser transcrito a javascript mediante un compilador. Dicho esto, se puede ver en realidad que typescript no “corre” en ninguna plataforma, sino que es traducido, por esto, se debe enfocar en ver qué navegadores son compatibles con cuales versiones de javascript.

Como se mencionó, no podemos hablar explícitamente de las plataformas en las que se puede utilizar typescript, sin embargo, si se pueden mencionar un conjunto de frameworks que lo utilizan. Entre estos hay tanto frameworks para front-end como para back-end. Entre los frameworks para el lado del cliente se tienen ReactJs, Angular, Vue.Js entre otros. Por otro lado, para el lado del servidor entre los más importantes se encuentran nodeJs, NestJs.

Este lenguaje es muy utilizado en frameworks populares, por ello existen una gran cantidad de aplicaciones que han sido desarrolladas con este lenguajes, entre estas podemos encontrar la página de de configuración de automóviles de la BMW de Estados Unidos, la del banco Santander, el editor de latex de Overleaf, Google, Github, Adobe, Netflix, Walmart, JetBrains, Microsoft, Slack, entre muchas otras.

## Mecanismos de instalación

La instalación de de Typescript es relativamente sencilla, ya que este se puede instalar mediante gestor de paquetes de nodeJs (npm). Una vez instalado nodejs y verificando que se tiene npm se utiliza la siguiente instrucción:

```
npm install -g typescript
```

Esta instrucción instalará de manera global en el sistema typescript, lo que dará acceso al compilador que este lenguaje “tsc”. Con este se puede generar un archivo de javascript a partir de uno de typescript, así como muchas más funcionalidades.

## Aspectos relevantes de su uso

En esta sección se muestra aspectos importantes en el uso de Typescript como instrucciones principales del compilador para empezar a trabajar con este lenguaje, así como aspectos de sintaxis.

### Compilador

Como se mencionó, los navegadores no pueden entender typescript, por ello, se debe transcribir de alguna manera el código hecho en typescript en javascript; esta es la función del compilador “tsc”. La instrucción más básica es la de compilar un programa de typescript, para esto se usa lo siguiente:

```
tsc <archivo.ts>
```

En la siguiente figura se muestra un programa en typescript y su salida en javascript. La fig. 1 muestra una clase con atributos tipados, así como parámetros de una función

tipados. El código de la fig. 2 muestra como al pasarlo a javascript se omiten los tipos y algunas funcionalidades como la concatenación se realizan de una manera distinta.

```
class persona {  
    private nombre: string;  
    private apellido: string;  
  
    constructor(nombre: string, apellido: string) {  
        this.nombre = nombre;  
        this.apellido = apellido;  
    }  
  
    public saludar(): void {  
        console.log(`Hola! mi nombre es ${this.nombre} ${this.apellido}`);  
    }  
}  
  
var p1 = new persona("lery", "Sanchez");  
p1.saludar();
```

**Figura 1.** Muestra de código escrito en Typescript. Fuente: propia.

```
"use strict";  
var persona = /** @class */ (function () {  
    function persona(nombre, apellido) {  
        this.nombre = nombre;  
        this.apellido = apellido;  
    }  
    persona.prototype.saludar = function () {  
        console.log("Hola! mi nombre es " + this.nombre + " " + this.apellido);  
    };  
    return persona;  
})();  
var p1 = new persona("lery", "Sanchez");  
p1.saludar();
```

**Figura 2.** Muestra de código javascript generado por el compilador tsc. Fuente: propia.

El método anterior puede ser un poco tedioso ya que se debe compilar cada vez que se genera un cambio en el archivo. El compilador brinda una funcionalidad para observar cambios en el archivo y compilar cada vez que se guarda un cambio. Esto se puede hacer mediante la bandera -w (watch) en el compilador como se muestra a continuación., lo que mostrará una salida como en la fig. 3.

```
tsc -w <archivo.ts>
```

```
[4:53:21 PM] Starting compilation in watch mode...  
[4:53:23 PM] Found 0 errors. Watching for file changes.
```

**Figura 3.** Mensaje mostrado por el compilador al observar cambios en archivos. Fuente: propia.

Con esto surge la pregunta, ¿Debo compilar cada archivo por separado?. La respuesta es no, el compilador brinda la posibilidad de compilar todos los archivos a la vez mediante la generación de un proyecto de typescript de la siguiente manera:

```
tsc --init
```

Esta instrucción generará un archivo tsconfig.json que incluye información para que el compilador compile el proyecto. Tener este archivo permite que cuando utilizemos los comandos anteriores se aplique a todos los archivos de typescript generados.

## Especificación de tipos

Esta es una de las características por las que TypeScript es popular ya que permite la definición de tipos, lo que ayuda a la detección de errores entre otras cosas. Para definir variables con tipos se utiliza el carácter ":" como se muestra en el siguiente ejemplo:

```
let nombre:string = "Rodrigo"
let list: number[] = [1, 2, 3];
let list: Array<number> = [1, 2, 3];
function getNombre(): string{
    return "Felicia";
}
```

Los tipos que tiene este lenguaje son los siguientes:

- Boolean
- Number
- String
- Array
- Tuple
- Enum
- Any
- Void
- Null and Undefined
- Never
- Object

# Vue.js

Vue fue creado por Evan You ex trabajador de Google, que también fue uno de los desarrolladores de Angular. Su primera versión fue lanzada en el 2014. Aunque inicialmente fue pensado para ser una biblioteca personal, la comunidad hizo que el proyecto creciera a un ritmo impresionante.

Vue es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde el inicio para ser adoptado incrementalmente. La biblioteca principal se enfoca solo en la capa de la vista, y es muy simple de utilizar e integrar con otros proyectos o bibliotecas existentes. Por otro lado, Vue también es perfectamente capaz de soportar aplicaciones sofisticadas de una sola página (en inglés *single-page-application* o SPA) cuando se utiliza en combinación con herramientas modernas y librerías compatibles.

## Historial de versiones

- 0.9 Animatrix en 2014
- 0.10 Blade Runner en 2014
- 0.11 Cowboy Bebop en 2014
- 0.12 Dragon Ball en 2015
- 1.0 Evangelion en 2015
- 2.0 Ghost in the Shell en 2016
- 2.1 Hunter X Hunter en 2016
- 2.2 Initial D en 2017
- 2.3 JoJo's Bizarre Adventure en 2017
- 2.4 Kill la Kill en 2017
- 2.5 Level E en 2017
- 2.6 Macross en 2019

## Características

- Es Open Source.
- Accesible.
- Versátil: Su núcleo es bastante pequeño y se escala a través de plugins.
- Escalable por el mismo tema de la versatilidad
- Optimizado.
- Tiene una gran comunidad.
- Vue trabaja con componentes, en términos sencillos se podría decir que es un elemento el cual se encapsula código reutilizable.



- Reactivo que utiliza objetos JavaScript simples y una representación optimizada.
- Cuenta con templates.

## Ventajas

- Tiene gran similitud con Angular. Tiene muchas similitudes con Angular, una de ellas es que se trabaja con componentes.
- Documentación detallada. Como tiene una gran comunidad, la documentación pasa en constante cambio y cada día se agrega más documentación.
- Adaptabilidad.
- Excelente integración. Debido a que es muy flexible cuenta con una excelente integración.
- Gran escalado, puede ser de un tamaño relativamente pequeño o bien puede llegar a escalar un gran proyecto.
- Pequeño tamaño. En comparación con Angular y React vue.js es de un tamaño muy pequeño.

## Desventajas

- Falta de recursos: todavía tiene poca cuota de mercado comparado con Angular o React.
- Riesgo de excesiva flexibilidad.
- Falta de documentación completa en inglés. Como es relativamente muy nueva, se podría tener dificultad de encontrar documentación completa en inglés.

## Plugins importantes

Vue.js cuenta con una gran cantidad de plugins, unos de los más importantes son:

- VUE-ROUTER: Sirve para gestionar las rutas.
- Conectar a servicios externos: Como vue-axios o vuexfire para conectar con bases de datos.
- TEST: Se pueden hacer test unitarios y funcionales.
- VUEX: Es una implementación de la arquitectura de aplicación FLUX.

## Páginas con vue js

Algunas páginas realizadas con vue js son las siguientes:

- Louis Vuitton
- Nintendo

## Ejemplos

En esta sección se mostrarán algunos ejemplos muy sencillos para ir conociendo un poco más de vue.js.

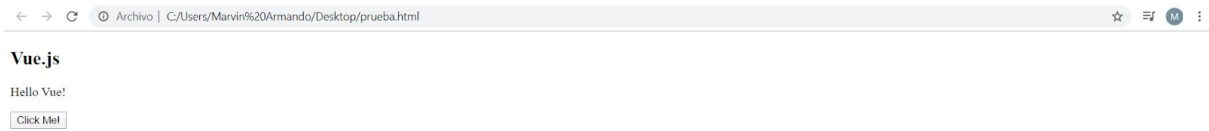
### Ejemplo #1

El ejemplo es bastante sencillo, consiste en mostrar un botón en una página que al presionar este mostrará el nombre que se escribió en la línea 24.

```
1  <!DOCTYPE html>
2  <html>
3    <script src="https://cdn.jsdelivr.net/npm/vue"></script>
4
5  <body>
6
7    <h2>Vue.js</h2>
8
9    <div id="app">
10      {{ message }}
11    </div>
12
13    <p>
14      <button onclick="myFunction()">Click Me!</button>
15    </p>
16
17    <script>
18      var myObject = new Vue({
19        el: '#app',
20        data: { message: 'Hello Vue!' }
21      })
22
23      function myFunction() {
24        myObject.message = "Marvin Castro Roldán";
25      }
26    </script>
27
28  </body>
29
30  </html>
```

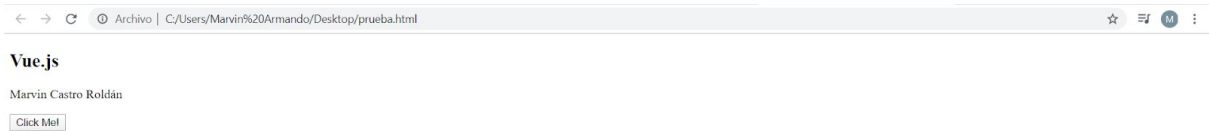
**Figura 4.** Muestra de código.

Fuente: [https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis\\_vue\\_1](https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_vue_1).



**Figura 5.** Se muestra el navegar..

Fuente: [https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis\\_vue\\_1](https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_vue_1).



**Figura 6.** Al presionar el botón el nos mostrará el nombre que estaba en myObject.message (Línea 24) Fuente: [https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis\\_vue\\_1](https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_vue_1).

## Ejemplo #2

En este ejemplo se desarrolla un componente que permite crear una libreta de los pagos que tenemos en un mes.

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="utf-8">
6      <title>Vue Example</title>
7      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
8          integrity="sha384-BVYiiSIFeK1dGmJRAKycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">
9  </head>
10
11 <body>
12     <div class="container">
13         <div class="row">
14             <div class="col-xs-12">
15                 <h1 class="jumbotron">VueJs</h1>
16                 <div id="app">
17                     <table class="table table-striped">
18                         <thead>
19                             <tr>
20                                 <th style="width:40px;"></th>
21                                 <th>Descripción</th>
22                                 <th style="width:200px;">Monto</th>
23                                 <th style="width:100px;">¿Cancelado?</th>
24                             </tr>
25                         </thead>
26                         <tbody>
27                             <tr>
28                                 <td></td>
29                                 <td></td>
30                                 <td><input type="text" class="form-control" v-model="newEntry.name" /></td>
31                                 <td><input type="text" class="form-control" v-model="newEntry.amount" /></td>
32                                 <td><button @click="add()" type="button" class="btn btn-success btn-block">Agregar</button></td>
33                             </tr>
34                             <tr v-if="items.length === 0">
35                                 <td colspan="4" class="text-center">No hay registros por mostrar</td>
36                             </tr>
37                             <tr v-for="item, index in items">
38                                 <td><button @click="remove(index)" type="button" class="btn btn-danger btn-xs"><i class="glyphicon glyphicon-trash"></i></td>
39

```

Figura 7. Sección de código Fuente: <https://anexsoft.com/ejemplo-tutorial-practico-de-vuejs>

```

50         </button>
51     </td>
52     <td>{{ item.name }}</td>
53     <td>{{ item.amount.toFixed(2) }}</td>
54     <td class="text-center" :title="item.paid ? 'Si' : 'No'"
55         <button @click="changeToPaid(item)" class="btn btn-default btn-sm"
56             :class="{ 'btn-success': item.paid }">
57
58             <i v-if="item.paid" class="glyphicon glyphicon-ok"></i>
59             <i v-if="!item.paid" class="glyphicon glyphicon-remove"></i>
60
61         </button>
62     </td>
63 </tr>
64 </tbody>
65 <tfoot>
66     <tr class="text-danger">
67         <td></td>
68         <td class="text-right">Por pagar</td>
69         <td>{{ totalAmount(0) }}</td>
70         <td></td>
71     </tr>
72     <tr class="text-success">
73         <td></td>
74         <td class="text-right">Pagado</td>
75         <td>{{ totalAmount(1) }}</td>
76         <td></td>
77     </tr>
78     <tr class="text-info">
79         <td></td>
80         <td class="text-right">Total</td>
81         <td>{{ totalAmount(2) }}</td>
82         <td></td>
83     </tr>
84 </tfoot>
85 </table>
86 </div>
87 </div>
88 </div>
89 </div>
90 <script src="https://unpkg.com/vue"></script>
91 <script>
92     var app = new Vue({
93         el: '#app', // Elemento
94         data: { // Parametros
95             newEntry: {
96                 name: '',

```

Figura 8. Sección de código Fuente: <https://anexsoft.com/ejemplo-tutorial-practico-de-vuejs>

```

97         amount: 0
98     },
99     items: [{
100         name: 'Servicios',
101         amount: 200,
102         paid: false
103     },
104     {
105         name: 'Hosting de Anexsoft',
106         amount: 90,
107         paid: true
108     },
109     ]
110 },
111 methods: {
112     remove: function (index) {
113         this.items.splice(index, 1);
114     },
115     // EcmaScript 6 syntax
116     add() {
117         this.items.push({
118             name: this.newEntry.name,
119             amount: parseFloat(this.newEntry.amount),
120             paid: false
121         });
122
123         this.newEntry.name = '';
124         this.newEntry.amount = 0;
125     },
126     changeToPaid(item) {
127         item.paid = !(item.paid);
128     },
129     totalAmount(t) {
130         var total = this.items.reduce(function (a, b) {
131             switch (t) {
132                 case 0: return a + (!b.paid ? b.amount : 0);
133                 case 1: return a + (b.paid ? b.amount : 0);
134                 case 2: return a + b.amount;
135             }
136         }, 0);
137
138         return total.toFixed(2);
139     }
140 }
141 });
142 </script>
143 </body>



```

**Figura 9.** Sección de código Fuente: <https://anexsoft.com/ejemplo-tutorial-practico-de-vuejs>

Como resultado obtenemos lo siguiente.

← → ↻ Archivo | C:/Users/Marvin%20Armando/Desktop/prueba.html




VueJs

Descripción	Monto	¿Cancelado?
<input type="text"/>	<input type="text" value="0"/>	<button>Agregar</button>
 Servicios	200.00	<input type="checkbox"/>
 Hosting de Anexsoft	90.00	<input checked="" type="checkbox"/>
	Por pagar 200.00	
	Pagado 90.00	
	Total 290.00	

**Figura 10.** Resultado Fuente:Propio

← → ↻ Archivo | C:/Users/Marvin%20Armando/Desktop/prueba.html

VueJs

Descripción	Monto	¿Cancelado?
<input type="text"/>	<input type="text" value="0"/>	<button>Agregar</button>
 Servicios	200.00	<input type="checkbox"/>
 Hosting de Anexsoft	90.00	<input checked="" type="checkbox"/>
 Internet	38.00	<input type="checkbox"/>
	Por pagar 238.00	
	Pagado 90.00	
	Total 328.00	

**Figura 11.** Al ingresar otra tarea, la página actualiza automáticamente el precio por pagar  
Fuente:Propio

## Mecanismos de instalación

La instalación de vue js es muy sencilla, cuenta con diferentes maneras de poder hacer uso de ella, las cuales se mostraran a continuación.

### Versión independiente

Está es la manera mas sencilla de instalar vue js, ingresando al siguiente link:<https://es-vuejs.github.io/vuejs.org/v2/guide/installation.html>, la cual consiste en descargar un archivo y posteriormente incluirla en nuestro proyecto por medio de una etiqueta script.

## Versión independiente

---

Simplemente descargala e incluyela con una etiqueta script. **Vue** se registrará como una variable global.



No utilices la versión minificada durante el desarrollo. ¡Perderás todas las advertencias para los errores comunes!

**Versión de desarrollo**

Con todas las advertencias y el modo depuración.

**Versión de producción**

Advertencias eliminadas, 24.72kb min+gzip

## Versión NPM

La siguiente forma de instalar vue js es mediante el comando NPM, este método se recomienda cuando se va a construir una aplicación de gran escala.

```
# última versión estable
$ npm install vue
```

Shell

## Versión CLI

Vue.js provee una CLI oficial para poder estructurar de manera rápida aplicaciones de una SPA.



```
# Instala vue-cli
$ npm install --global vue-cli
# Crea un nuevo proyecto usando la plantilla "webpack"
$ vue init webpack my-project
# Instala las dependencias y ¡listo!
$ cd my-project
$ npm install
$ npm run dev
```

## Referencias

Microsoft (s.f.). Quick Start · TypeScript. Disponible en:

<https://www.typescriptlang.org/Samples/>

Alberca, C. (2018). Typescript vs javascript. Retrieved from

<https://www.adictosaltrabajo.com/2018/08/01/typescript-vs-javascript/>

Azaustre, C. (2017). **Qué es lo que me gusta de vue.js**. Retrieved from

<https://carlosazaustre.es/que-es-lo-que-me-gusta-de-vue-js/>

Coding Potions. (2019). Introducción a vue JS » qué es y sus característica. Retrieved from

<https://codingpotions.com/que-es-vue>

Colina, C. (2019). Ventajas y desventajas de TypeScript. Retrieved from

<https://time.graphics/es/line/155170>

Dongil, J. (2018). **Por qué elegir VueJS: 5 razones para considerarlo nuestro próximo framework de referencia**. Retrieved from

<https://www.genbeta.com/desarrollo/por-que-elegir-vuejs-5-razones-para-considerarlo-nuestro-proximo-framework-de-referencia>

Font, J. (2019). **ReactJS vs Angular5 vs vue.js ¿Cuál elegir?**. Retrieved from

<https://medium.com/@javifont/reactjs-vs-angular5-vs-vue-js-cuál-elegir-a9358068cf94>

Hernandez, U. (2018). **Qué es TypeScript**. Retrieved from

<https://codigofacilito.com/articulos/typescript>

Kevin. (2019). **Typescript**. Retrieved from <https://time.graphics/es/line/155170>

Rosa, J. (2017). ¿Qué es vue.js? Retrieved from <https://openwebinars.net/blog/que-es-vuejs/>

Vue.js. (2015). **Vue.js 0.12 released!**. Retrieved from

<https://vuejs.org/2015/06/11/012-release/>

Vue.js. (2019). Instalación vue.js. Retrieved from

<https://es-vuejs.github.io/vuejs.org/v2/guide/installation.html>

Vue.js. (2019). **Reactividad en profundidad**. Retrieved from

<https://es-vuejs.github.io/vuejs.org/v2/guide/reactivity.html>

w3schools.com. (2018). **Javascript versions**. Retrieved from

[https://www.w3schools.com/js/js\\_versions.asp](https://www.w3schools.com/js/js_versions.asp)

w3schools.com. (2018). What is vue.js. Retrieved from

[https://www.w3schools.com/whatis/whatis\\_vue.asp](https://www.w3schools.com/whatis/whatis_vue.asp)

Web. (2017). **Made with vue.js**. Retrieved from <https://madewithvuejs.com/websites>