

Apuntes TypeScript y Vue.js

Antecedentes de TypeScript

- 👤 Creado, en 2012, por Steve Luco y Anders Heijlsberg
- 👤 Microsoft realizar Strada para corregir los problemas que presentaba JavaScript en aplicaciones a gran escala

Características

- 👤 Es de código abierto
- 👤 Superconjunto de JavaScript Herramienta para administrar paquetes, en 2001 se hizo compatible con windows, hubo una separación del equipo pero se volvieron a unir.

Ventajas

- 👤 Tiene mucha documentación
- 👤 Código más entendible
- 👤 Finalización de código e IntelliSense
- 👤 Se puede realizar código OO de una manera más sencilla
- 👤 Curva de aprendizaje liviana
- 👤 Detecta errores en tiempo de compilación

Desventajas

- 👤 Mayor complejidad en la configuración de nuestro proyecto
- 👤 Toma más tiempo y dedicación para escribir código

Frameworks que lo utilizan

- 👤 Banco Santander
- 👤 GitHub
- 👤 Walmart
- 👤 Slack
- 👤 JetBrains
- 👤 OverLeaf
- 👤 Google
- 👤 Adobe
- 👤 Netflix

Compilación

- 👤 Solo “entiende” javascript
- 👤 Pasa de TypeScript a JavaScript
- 👤 `tsc <archivo.ts>` (genera archivo js, hay que editar después de compilar)
- 👤 `tsc -w <archivo.js>` (watch, se queda esperando cambios, compila despues de save)
- 👤 `tsc --init` (trar archivo json donde estan todas las configuraciones y vienen los archivos que hay que compila)

Antecedentes Vue.js

- 📖 Creado, en 2004, por Evan You
- 📖 Creado para facilitar el programar
- 📖 Pensado, en sus inicios, como una biblioteca personal

Características

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

Componente

- 📖 Template
- 📖 Script
- 📖 Estilo

Ventajas

- 📖 Similar a Angular
- 📖 Documentación detallada
- 📖 Adaptabilidad
- 📖 Excelente integración
- 📖 Gran escalada
- 📖 Menor tamaño que Angular y React.js

Desventajas

- 📖 Falta de recursos
- 📖 Riesgo de excesiva flexibilidad cuando se trata de proyectos grandes.
- 📖 Falta de documentación

Plugins importantes

- 📖 Vue Router: Gestor de rutas
- 📖 Vuefire: Conexión con base de datos
- 📖 Vue-axios: Conexión con REST APIs
- 📖 Vuex: Gestor de componentes

Estructura

- 📖 Public: los html y los componentes se agregan allí
- 📖 Assets: imágenes, fuentes, demás recursos multimedia
- 📖 Components: aquí se crean los componentes
- 📖 Router: especifica las rutas para mostrar todas las vistas.
- 📖 App.vue contiene todos los componentes, los cuales 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

Instalación para el Taller

La instalación de los elementos necesarios para el taller fue bastante sencilla. Constaba de ejecutar dos comandos para instalar lo necesario.

Para TypeScript se debía ejecutar

```
npm install -g typescript
```

Para Vue.Js

```
npm install -g @vue/cli
```

Lo cual fue bastante fácil y rápido.

Taller

El taller consistió en la realización de un sistema para administrar tareas o actividades, donde se incluyen todas las operaciones de leer, crear, modificar y eliminar para los ítems de tareas.

El sistema consta de dos pantallas, una pantalla inicial donde se explica en que consiste el taller y la pantalla de “Tareas por Hacer”



[Acerca de este proyecto](#) [Por hacer](#)

Taller de Vue.js con TypeScript

Este taller fue realizado para proponer un ejercicio para ser desarrollado en clase, de manera que se muestren las funcionalidades que propone el framework VueJs utilizando TypeScript como lenguaje de programación. La aplicación que se desarrolla en este taller consiste en un sistema para administrar tareas o actividades, donde se incluyen todas las operaciones de leer, crear, modificar y eliminar para los ítems de tareas.



Tareas por hacer

☐ Terminar el laboratorio 2☐ Ponerle con el proyecto final! :O☐ Marcar todos como terminados

2 pendientes

[Todos](#)[Pendientes](#)[Terminados](#)