

Estudio de la popularidad del framework VueJS

Autor

Raül de Arriba Garcia

rauldearriba@gmail.com

Abstract—El número de páginas web que tenemos disponibles en Internet cada día es mayor y el número de lenguajes de programación, también. Cada tecnología nos ofrece una serie de ventajas e inconvenientes que tendremos que evaluar para cada proyecto. En este artículo veremos VueJS como framework progresivo para el desarrollo front end y que ventajas ofrece respecto a otros frameworks basados en JavaScript más utilizados actualmente por la comunidad y grandes empresas.

Keywords— VueJS, JavaScript, framework, modular, MVVM, Virtual DOM

I. INTRODUCCIÓN

VUEJS es un framework progresivo basado en el lenguaje de programación JavaScript — las funcionalidades principales, que son el renderizado y el sistema de componentes, se encuentran en una biblioteca con las funcionalidades justas para que funcione, eso permite un renderizado en un tiempo corto. Las demás funcionalidades se consiguen añadiendo bibliotecas externas —

Fue creado por Evan You, antiguo desarrollador de Google, como proyecto personal con el objetivo de facilitar el funcionamiento de AngularJS. [1] En este artículo se presenta el crecimiento de popularidad de este framework en la comunidad de desarrolladores. Veremos que esto viene dado por la propuesta que hace Vue para simplificar el funcionamiento de otros frameworks como Angular o ReactJS, sin perder funcionalidades ni tener dependencias entre bibliotecas a la hora de utilizarlas. A demás, compararemos Angular y VueJS para ver a que se debe esta evolución en un tiempo de solamente 4 años y medio.

El artículo presenta la siguiente estructura, en la sección 2 se presenta la evolución de VueJS, en la sección 3 se muestra que son los componentes, en la sección 4 se describe el patrón que utiliza, en la sección 5 se habla del Virtual DOM y React, en la sección 6 compararemos los frameworks y finalmente las conclusiones obtenidas.

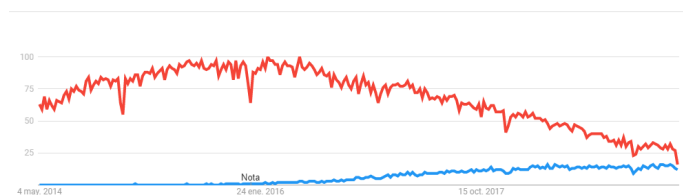


Fig. 1. Evolución de popularidad según Google Trends desde 2014 (rojo: Angular, azul: Vue)

II. LA EVOLUCIÓN DE VUEJS

VueJS, framework creado en 2014 sobre JavaScript, surgió a partir de que su creador, que anteriormente, fue desarrollador Angular para la empresa Google, opinara sobre la lenta curva de aprendizaje que debía de tener con Angular para poder iniciar un desarrollo y comenzó el proyecto de VueJS con la finalidad de conseguir un framework para el desarrollo de interfaces de usuarios, sencillo de utilizar y de rápido aprendizaje. VueJS, como podéis ver en el gráfico de popularidad de la figura 1, está igualando el nivel de popularidad de Angular, a nivel de búsquedas. Este rápido crecimiento se debe a cuatro motivos.

La curva de aprendizaje rápida, con hacer los ejemplos de la documentación se aprecia su facilidad. Con eso ya dispones de los conocimientos base para iniciar un proyecto.

El rápido renderizado de componentes con el uso de Virtual DOM. Esta es su característica principal y la propia página web de VueJS documenta esta característica [2]. A demás da compatibilidad a navegadores antiguos.

La documentación, todo y que la comunidad todavía no es muy amplia, VueJS dispone de unos ejemplos y explicaciones excelentes.

Finalmente, VueJS es un framework progresivo. Las librerías están desacopladas, esto permite desarroyar el proyecto por módulos y no tener dependencias. En su corta etapa de vida, VueJS consiguió el primer puesto en popularidad en el año 2016 en github. [3]

III. LOS COMPONENTES

La propia web de VueJS nos describe los componentes de la siguiente manera, “Los componentes son una de las características más poderosas de Vue. Te permiten extender elementos HTML básicos para encapsular código reutilizable.”[4] Para que la definición que da la pagina web sea posible, los componentes deben seguir una serie de características.

Como norma general, los componentes de Vue deben ser independientes — no deben depender de componentes adicionales para su utilización — y reutilizables. Cumplen el principio de responsabilidad única, se crean para un objetivo y solo debe hacer ese trabajo. No deben interactuar con el exterior directamente, pueden interactuar a partir de eventos para evitar dependencia de datos. Deben tener la capacidad de poder cambiar los estilos de los componentes desde fuera para adaptarlos a nuestras necesidades sin tocar la base del componente. Finalmente, se evita crear componentes demasiado pequeños, como una imagen o un texto, para no sobrecargar de componentes y dependencias los proyectos. [5]

IV. EL PATRÓN MODEL - VIEW - VIEWMODEL

AngularJS es uno de los frameworks, creados sobre JavaScript igual que VueJS, front end de los más populares actualmente gracias al soporte de la gran compañía que lo creó, Google. Angular, lanzado a la luz en 2009, trabaja bajo el patrón Modelo-Vista-Controlador (MVC) pero con una estructuración poco marcada debido a que en este, el modelo tiene mucha relación con la vista, tanta que la vista puede modificar al modelo y el modelo puede modificar a la vista. [6]

En cambio, VueJS utiliza el patrón Model-View-ViewModel (MVVM) o también conocido como Model-View-Whatever. Este patrón tiene como objetivo simplificar el desarrollo y el mantenimiento del software. MVVM se divide en 3 partes. Modelo, igual que en otros patrones, representa la capa de datos, contiene la información pero nunca la modifica. Vista, su función es la de mostrar la información que los usuarios verán. El punto de conexión entre los 3 componentes comentados anteriormente es el ViewModel o modelo de la vista, donde su función es obtener los datos y manipularlos para que se muestren de la forma deseada en la vista. Normalmente hay una ViewModel para cada vista.[7]

V. EL VIRTUAL DOM Y REACT

React es el otro framework de JavaScript que junto con Angular, tienen los niveles de popularidad y utilización para desarrollo front end más elevados.

React debe su popularidad al Virtual Document Object Model (virtual DOM). Un DOM Virtual es una representación del árbol DOM (figura 2) Actualizar el DOM real de manera frecuente es una tarea muy pesada, esta sobre el orden de $O(n^3)$, de ahí el DOM Virtual. Actualizar un objeto JavaScript es una tarea libiana en comparación, gracias a las heurísticas utilizadas, la complejidad, disminuye al orden de $O(n)$. [8]

VueJS a partir de la versión 2.0, adoptó el DOM virtual creado por React por completo, sin ninguna diferencia. Esto fue posible gracias a que ambos frameworks basan su arquitectura en componentes.

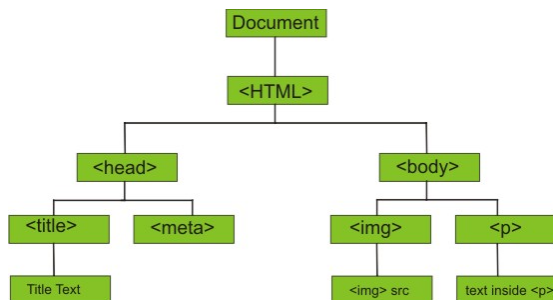


Fig. 2. Árbol DOM (figura de mialtowe.es)

VI. COMPARACIÓN

Como hemos visto, Angular utiliza un patrón MVC poco definido a diferencia de VueJS que se basa en MVVM. Esto facilita el mantenimiento de código y el aprendizaje al

tener una estructura clara. Angular 2, tiene una ventaja clara respecto a vueJS, está diseñado para aplicaciones de gran escala y una base de código grande. Lo utilizan empresas como Google, Paypal o Airbnb. VueJS todavía no se ha utilizado para un gran proyecto por lo que no se ha podido comprobar la robustez. [9]

Podemos ver que VueJS y React tienen muchas cosas en común, como la arquitectura basada en componentes y la utilización del DOM Virtual. Pero a la hora de hacer las plantillas, Vue utiliza HTML tradicional, esto le da mayor facilidad para integrarse en aplicaciones ya existentes. React en cambio, recomienda el uso de extensión de sintaxis a JavaScript (JSX) que es JS con sintaxis de XML. React Native, parte de React que da la capacidad de utilizar JavaScript, CSS y HTML para crear aplicaciones móviles nativas. Es un componente del que Vue no dispone todavía. [10]

CONCLUSIÓN

En este artículo se han visto los motivos por los cuales un framework, con apenas 4 años de vida, ha conseguido un crecimiento a nivel de popularidad muy rápido en comparación con otras tecnologías asentadas, como lo es Angular o React. Esto es gracias a que Vue ha sabido coger lo mejor de cada uno de ellos. Como conclusión los 3 frameworks tienen un gran rendimiento por lo que si el proyecto es muy grande, se debería plantear Angular. Si el proyecto tiene código creado anteriormente, Vue es la mejor opción. Para desarrollo de aplicaciones móviles React Native. Pero por juntar lo mejor de Angular y React, VueJS está subiendo su popularidad hasta estos niveles tan altos.

REFERENCIAS

- [1] José Antonio Dongil Sánchez. "Por qué elegir VueJS" <https://www.genbeta.com/desarrollo/por-que-elegir-vuejs-5-razones-para-considerarlo-nuestro-proximo-framework-de-referencia> Accedido en: Abril, 2019.
- [2] VueJS. "Comparison with Other Frameworks" <https://vuejs.org/v2/guide/comparison#search-form> Accedido en: Abril, 2019.
- [3] Kendru Estrada. "Por que es Vue.js el nuevo framework de moda" <https://medium.com/blog-apside/por-que-es-vue-js-es-el-nuevo-framework-de-moda-79de70e13ef5> Accedido en: Mayo, 2019.
- [4] VueJS. "Componentes" <https://es-vuejs.github.io/vuejs.org/v2/guide/components.html> Accedido en: Abril, 2019.
- [5] Eder Negrete. "Mi Arquitectura en Vue" <https://platzi.com/tutoriales/1145-vuejs-profesional/1410-mi-arquitectura-en-vue/> Accedido en: Abril, 2019.
- [6] AngularJS. "What are Scopes?" <https://docs.angularjs.org/guide/scope> Accedido en: Abril, 2019.
- [7] theshallowbay. "El patrón MVVM - Introducción" <https://theshallowbay.github.io/tutoriales/2017/07/02/patron-mvvm-introduccion/> Accedido en: Abril, 2019.
- [8] Ruphesh Mishra. "Virtual DOM in ReactJS" <https://hackernoon.com/virtual-dom-in-reactjs-43a3fdb1d130/> Accedido en: Mayo, 2019.
- [9] Uriel Hernandez. "Angular, React O Vue ¿Cuál Elegir?" <https://codigofacilito.com/articulos/angular-react-vue> Accedido en: Mayo, 2019.
- [10] Daniel Ortego Delgado. "Diferencias entre Vue.js y React" <https://openwebinars.net/blog/diferencias-entre-vuejs-y-react/> Accedido en: Mayo, 2019.