ANGULAR VS REACT

## Primer motivo: ReactJS no es un framework, es una librería

El primer motivo es que **AngularJS se presenta como un framework total**. Un framework donde podemos hacer todo lo necesario en una aplicación front compleja. Podemos incluir ciertas extensiones al framework, pero por lo general, el core de AngularJS nos va a permitir realizar un porcentaje altísimo de aplicaciones sin tener que incluir grandes cambios.

**ReactJS en cambio se define como una librería. Una librería que tiene como fin un propósito muy definido**: ReactJS solo quiere encargarse de gestionar la vista de tu aplicación. Si te acercas a esta librería, te cansarás de que Facebook te diga que si nuestra aplicación está diseñada como un MVC, **ReactJS se encargará de dar forma a la V de nuestro patrón**.

Por esto está gustando tanto ReactJS. Es una librería ligera que solo se encarga de hacer bien una cosa y nada más. Como vemos es complicado comparar un framework contra una librería, porque sería como comparar una navaja multiusos con un cuchillo. La navaja sabemos que hace muchas cosas que el cuchillo no puede. Lo único que sabemos es que el cuchillo corta muy bien.

## Segundo motivo: El paradigma de ReactJS no es el de AngularJS

**ReactJS nació como una solución para realizar programación declarativ**a en la capa de interfaz de una aplicación. ReactJS se presenta como una opción muy válida para desarrollar aplicaciones JavaScript por medio del paradigma funcional. Es decir, aplicaciones que solo contengan un único flujo, donde las funciones no contengan estado y si lo contienen, este sea  inmutable, en la medida de lo posible.

Todo esto se traduce a que ReactJS crea interfaces por medio de componentes. Componentes mínima, con funcionalidad mínima que se van componiendo los unos a los otros para formar jerarquías de componentes. Donde los datos fluyen desde las capas más genéricas de componentes hacia los más simples y específicas, formando un árbol donde las entradas son datos y las salidas son html.

**Todas estas ideas no son consideradas en AngularJS. AngularJS apostó en su día por ser más tradicional y diseñarse dentro de un patrón estructural MVC que con el tiempo irá dando paso a un MVVM** y que a día de hoy tiene la flexibilidad suficiente como para permitir al desarrollador usar el que necesite, denominándose a esto el patrón MVW o MV\* (Model-View-Whatever).

## Y si ReactJS es una librería que se encarga de la vista ¿Por qué no lo usamos junto con AngularJS?

Al igual que se está poniendo muy de moda el comparar AngularJS con ReactJS. Existen muchos tutoriales de cómo usarlos juntos. Esto es algo que se puede hacer. Podemos hacer que ReactJS se encargue de gestionar la jerarquía de componentes visuales de nuestra aplicación y que AngularJS funcione en el resto de capas.

**Particularmente yo no lo recomiendo.** Creo que AngularJS cuenta con un motor de vistas por medio de directivas y componentes bastante potente y que se adapta mejor a las necesidades de su framework a como lo hace ReactJS. Creo que ReactJS pierde esencia usándose con AngularJS y creo que migrar aplicaciones a Angular 2 en el futuro se complicaría demasiado.

Fuente:

<https://elabismodenull.wordpress.com/2016/02/28/angularjs-vs-reactjs-como-confundir-la-velocidad-y-el-tocino/>

I think some of the most interesting differences are the size, community, debugging, DOM, and mobile. When talking about mobile React has its own way of building mobile apps through React Native. Angular doesn’t try to tackle it’s own mobile and allows others like Ionic Framework to do this for it.

While the AngularJS 1 community is very large, the Angular 2 community is just gaining ground. This is in part due to Angular 2 still being in beta, but the traction it has already gotten is pretty outstanding. React’s community continues to grow and become more diverse as well.

When looking at sizes, there are some distinct differences between React and Angular. While Angular 2 is currently at around 764k in size, they are going to be trimming a lot of that off in the coming months. React also does not do as much of the MVC model as Angular so it can get away with having a much smaller size. However if you add in the extras to React to make it fully MVC then you will still come out with a smaller framework size than Angular 2.

Debugging is sometimes the hardest thing to do. I think it is interesting how each framework handles debugging. Angular 2 and it’s runtime debugging tends to give you less information than React and it’s compile time debugging.

Finally, I wanted to talk a second about the Virtual DOM that React uses. This was a new concept to me, and I looked at it more in depth. This concept seems to be much faster since you don’t have to work with the heavyweight parts that the real DOM has. Manipulating and changing a virtual copy of your DOM that is lightweight, and then only pushing the changes that are shown when doing a diff on the DOM is pretty awesome.

After spending a few days researching and reading other articles on these two frameworks, I have found that there is a lot of give and take when it comes to comparing the two. On some things Angular is better and on others, React is better. As always you should asses the project that you are working on to find out what best suits your needs. Maybe this comparison chart can help you in that decision.

Fuente:

<https://www.pluralsight.com/guides/front-end-javascript/angular-vs-react-a-side-by-side-comparison>

DEBATE EN CLASE:

Porque preferir REACT.JS sobre Angular.JS

* menos pero mejores recursos
* librería que puede complementarse o combinar con otras librerías
* mas rápido y ligero
* más simple de manejar
* componentes pequeños de mejor calidad
* manejo de vista controlado
* se facilita la programación
* invierte menos tiempo en traducir
* solo se agregan librerías
* no necesita ejecutar el programa para detectar errores
* detecta errores y permite corrección en la programación
* si no compila no muestra el componente en el cliente
* se coloca el html dentro del js
* importar por completo la carpeta
* se compila en el dispositivo móvil, no se interpreta
* es aplicación nativa

Conclusión:

Llegamos a la conclusión de no optar específicamente por Angular.js o por React.js ya que la aplicación para cada parte puede ser diferente ya que React.js es una gran decisión para la programación en JavaScript pero es muy difícil la implementación de HTML al contrario de Angular.js es mejor en HTML pero se puede programar en JavaScript pero no mucho.