Universidad de Ciencias y Humanidades



Docente:Eric Coronel

Alumna:Flores Cruz Melissa

2017

Introducción a Vue.js

Durante los últimos 5 años han surgido una cantidad de librerías y frameworks javascript para facilitar el desarrollo de aplicaciones para el frontend.

Entre las mas famosas se encuentra Angular.js, Ember.js, Knockout.js y recientemente React.js desarrollado por el equipo de Facebook.

Hoy quiero darte una introducción a una librería que promete mucho y que tendrá un gran auge en el 2016.

Estoy hablando de [Vue.js](http://vuejs.org/).

Que es Vue.js?

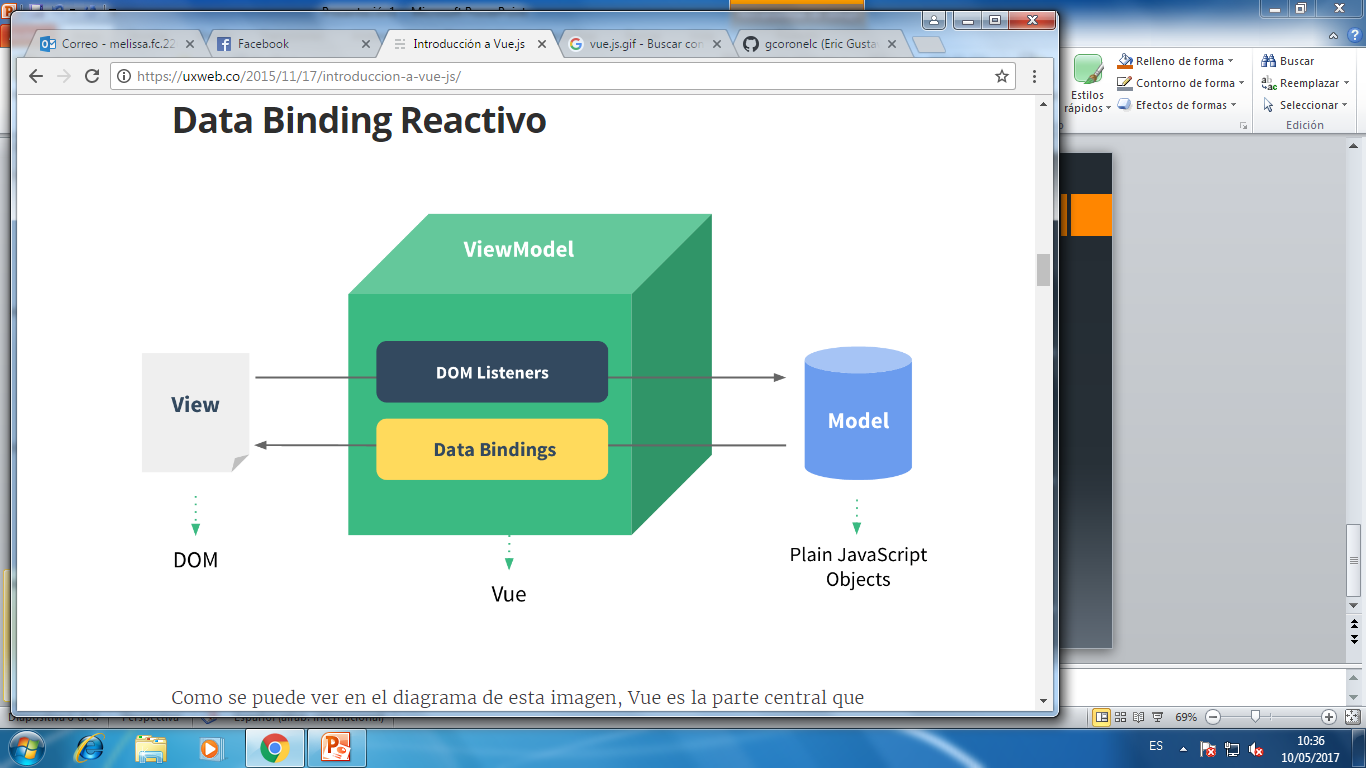
Vue.js es una librería javascript para construir interfaces web interactivas, la documentación dice que no es un framework por que solo se especializa en la capa de vista (VM - ViewModel).

Entonces es solo otro "framework" javascript?

Si, pero antes de que abandones este artículo

Las características que considero mas sobresalientes de Vue son:

* Aprenderlo es facil.
* Hace el código mas fácil de escribir, de entender y de mantener.
* Su sistema de data binding.
* Su sistema de componentes.
* Plugins increibles para hacer peticiones http, validación de formularios, router y soporte para transformadores como Browserify (vuefy) o WebPack para escribir javascript moderno.



Como se puede ver en el diagrama de esta imagen, Vue es la parte central que se encarga de mantener en sincronía los datos del modelo y la vista, este concepto es llamado data driven view.

En palabras simples, si los datos cambian, la vista cambiara también.

Es por eso que en muchos lugares de la documentación veras que se refiere a Vue como el vm (view-model).

Para lograr esta sincronización, Vue utiliza directivas que se escriben como atributos html en los elementos que necesites enlazar al *vm*, y cuando los datos del modelo se modifiquen, todos los elementos enlazados serán actualizados también.

<input type="text" id="nombre" v-model="nombre"/>

En este ejemplo, la directiva es v-model, y como esta, existen otras y se escriben de forma muy parecida, primero v- y después el nombre de la directiva.

Vue esta inspirado por Angular, y esto se debe a que su [creador](https://twitter.com/youyuxi) lo utilizó mucho durante el tiempo que trabajó para Google.

Ejemplo

Vamos a hacer el ejemplo mas simple posible, de esta forma podremos observar el data binding en acción.

Aquí esta todo el markup que analizaremos para entender como funciona Vue:

 <!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<title>Comanzando con Vue.js</title>

</head>

<body>

<div id="app">

<input type="text" placeholder="Escribe tu nombre" v-model="nombre" />

<span>{{ nombre }}</span>

</div>

<script src="http://cdn.jsdelivr.net/vue/1.0.8/vue.min.js"></script>

<script>

vm = new Vue({

el: '#app',

data: {

nombre: ''

}

});

</script>

</body>

</html>

Antes que nada, incluye el javascript de Vue, puedes incluirlo desde cualquier CDN, en la documentación se sugieren dos [CDN](http://vuejs.org/guide/installation.html#CDN) que a la fecha apuntan a la versión 1.0.8, si estas siguiendo este ejemplo es posible que ya se encuentre en una version mas avanzada.

Esto no sera problema mientras la version menor no cambie, es decir, mientras sea 1.0.\*, ya que Vue utiliza versionamiento semántico (mayor.menor.hotfix), esto asegura que el API no cambia en versiones hotfix.

Para incluirlo solo agregaremos la url del CDN en un elemento script

<script src="http://cdn.jsdelivr.net/vue/1.0.8/vue.min.js"></script>.

Cada instancia que generes de Vue está enlazada a un elemento html, es por esta razón que he creado un elemento donde la instancia sera "montada" con este markup:

<div id="app">

<input type="text" placeholder="Escribe tu nombre" v-model="nombre" />

<span>{{ nombre }}</span>

</div>

Aquí estamos creando un div con el id app, de esta forma, Vue podrá encontrar el elemento y hacer toda la magia (observadores, eventos, etc) para mantener enlazados los datos con la vista, si, este markup es nuestra vista!.

La vista esta compuesta por un cuadro de texto y un elemento span.

El cuadro de texto tiene una propiedad v-model="nombre".

Esta propiedad es una directiva y hace que la propiedad nombre de nuestro modelo este enlazada a este elemento html.

El elemento span tiene un contenido diferente. Si desarrollas con Laravel notaras inmediatamente que las llaves {{ }} son las mismas que utilizas con el motor de plantillas Blade. Si, en realidad tanto Vue como Blade utilizan el mismo tipo de llaves para mostrar los datos del modelo en las vistas.

Por ahora solo estamos con el frontend, así que una colisión por utilizar las mismas llaves no puede ocurrir.

Estas llaves mostraran el valor actual de la propiedad del modelo que pongamos entre ellas, en este caso, la propiedad es nombre {{ nombre }}.

Y ahora la parte final, la que le da el sabor a este pequeño ejemplo:

var vm = new Vue({

el: '#app',

data: {

nombre: ''

}

});

Aquí estamos creando la instancia de Vue con new Vue(), como parámetro de construcción, recibe un objeto donde puedes especificar los datos, métodos, opciones y mas.

La opción **el**

Aquí puedes especificar un selector css del elemento donde se va a montar la instancia de Vue.

La opción **data**

Este es un objeto javascript que representa el modelo (los datos) de nuestra aplicación. Dentro de este podemos crear las propiedades que necesitemos.

En Resumen

Vue es un framework joven, recientemente se lanzo la version 1.0, un dato curioso es que parece que el creador tiene cierto gusto por las series de anime japonesas porque esta version tiene el sobrenombre *Evangelion*, y la version 0.12 *Dragon Ball* y la 0.11 *Cowboy Bebop*.

Pienso que la version 0.12, fue la que más retroalimentación recibió por parte de los usuarios y la que definió muchas de las nuevas características en la version 1.0.

El creador de Laravel ([Taylor Otwell](https://twitter.com/taylorotwell)) tiene una afinidad con este framework, es así que en su ultimo proyecto open source [Laravel Spark](https://github.com/laravel/spark) lo utilizó para toda la parte del frontend.