

# PROYECTO DE SOFTWARE

# TEMARIO

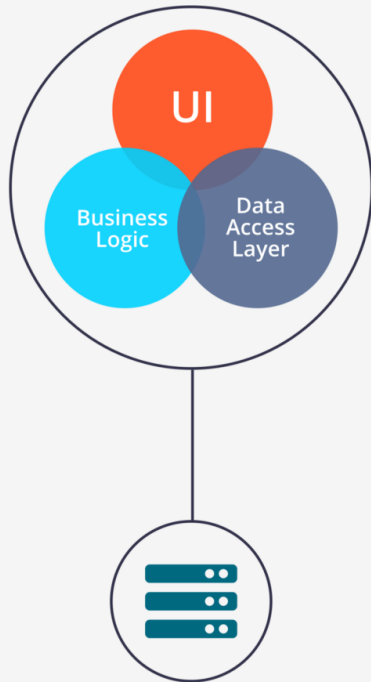
- Microservicios
- Frameworks JS
- Intro Vue

# ARQUITECTURA DE MICROSERVICIOS

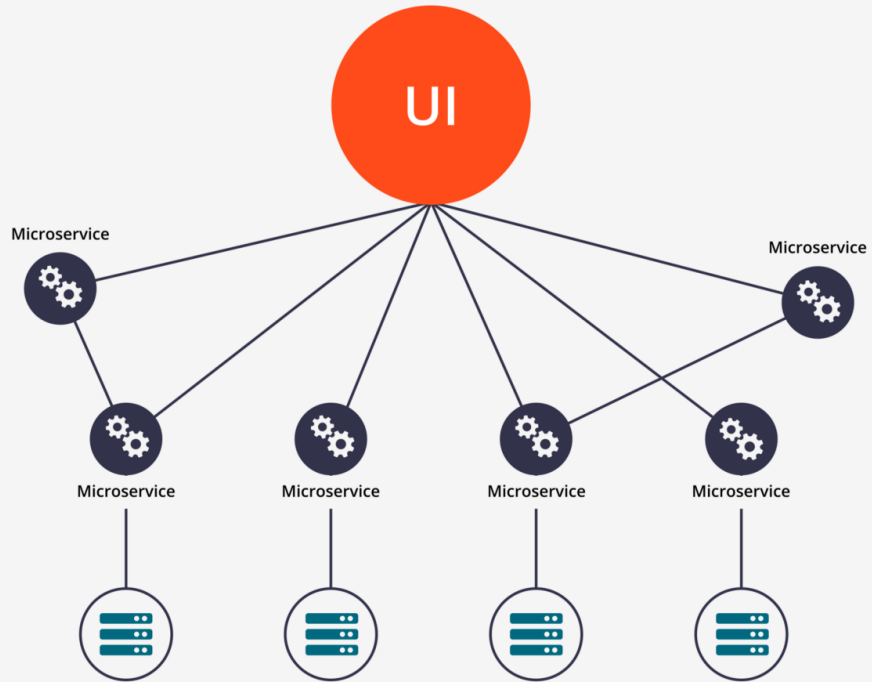
# MICROSERVICIOS

- Las aplicaciones se dividen en sus componentes más pequeños, independientes entre sí.
- A diferencia del enfoque tradicional y monolítico de las aplicaciones, en el que todo se encuentra en una única pieza.
- Los microservicios funcionan en conjunto para llevar a cabo las mismas tareas que la aplicación monolítica.
- Los microservicios facilitan la escalabilidad de todo el sistema, se despliegan según se vayan necesitando.
- Pueden tener distintas tecnologías entre sí.
- Al ser más pequeños, son mas simples de mantener y actualizar.

# ARQUITECTURA MONOLÍTICA VS ARQUITECTURA DE MICROSERVICIOS



Monolithic Architecture

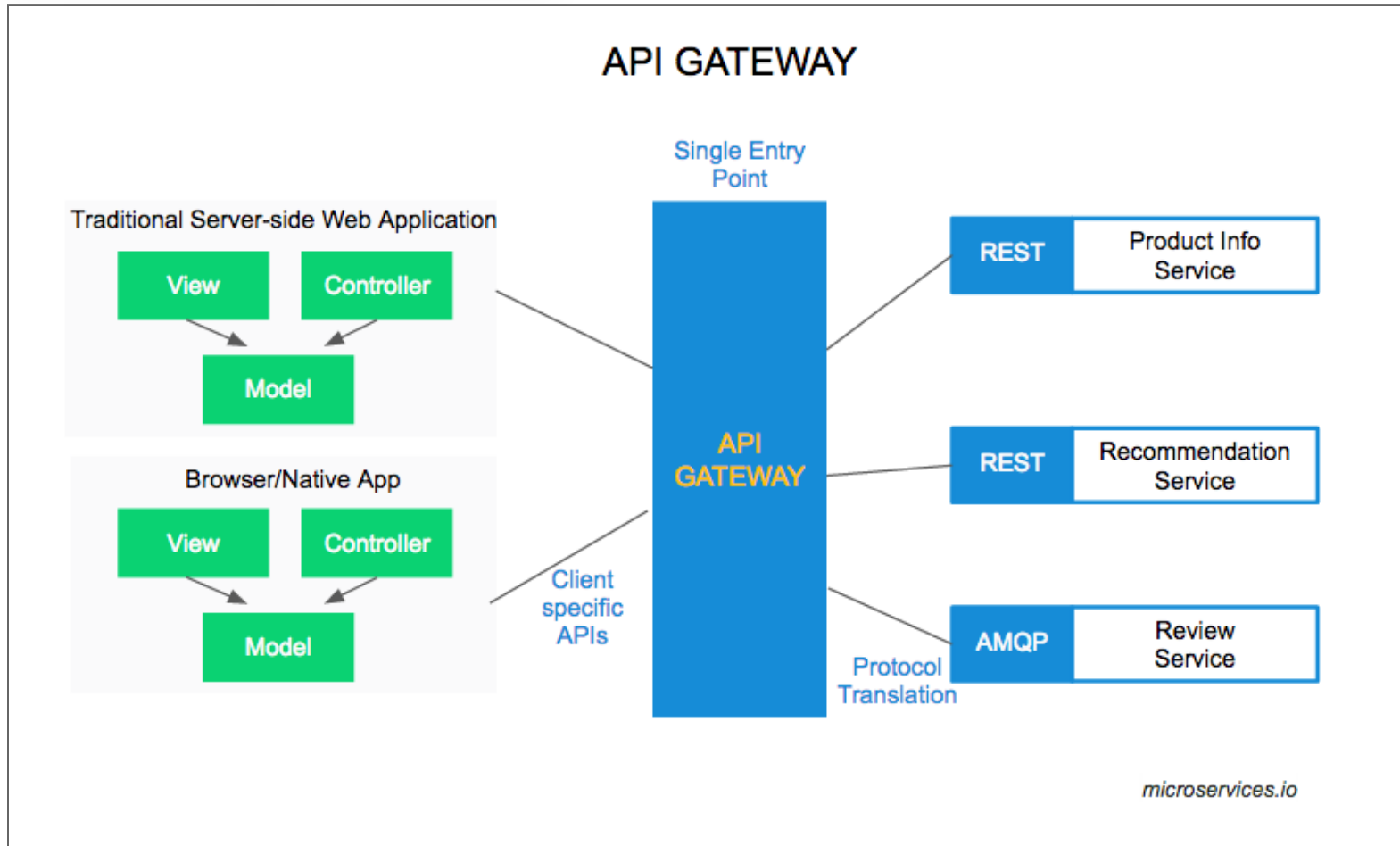


Microservice Architecture

## ¿CÓMO COMUNICO UN SERVICIO CON OTRO?

- Actualmente la opción más utilizada es mediante APIs HTTP/REST con JSON.
- Incluso puede centralizarse la comunicación utilizando un API Gateway.
- En dicho API Gateway puede implementarse una capa de seguridad, que ante una petición verifique si el cliente tiene permisos de acceso.

# MICROSERVICIOS: API GATEWAY



## PARA SEGUIR LEYENDO DE MICROSERVICIOS

- AWS: <https://aws.amazon.com/es/microservices/>
- Redhat: <https://www.redhat.com/es/topics/microservices>
- Video en español: <https://www.youtube.com/watch?v=9R2hFwIPGnQ>



# FRAMEWORKS JS

# ¿QUÉ ES UN FRAMEWORK JS?

- Trabajar con **JS pelado** es complejo. Por eso nació **jQuery**(2006) para facilitar el desarrollo.
- jQuery sigue estando muy extendido. **Stackoverflow survey 2021, 2022**.
- El poder de cómputo de los clientes web **aumenta día a día**.
- Las aplicaciones web implementan **cada vez más funcionalidades y complejidad**, con lo que jQuery se queda corto.

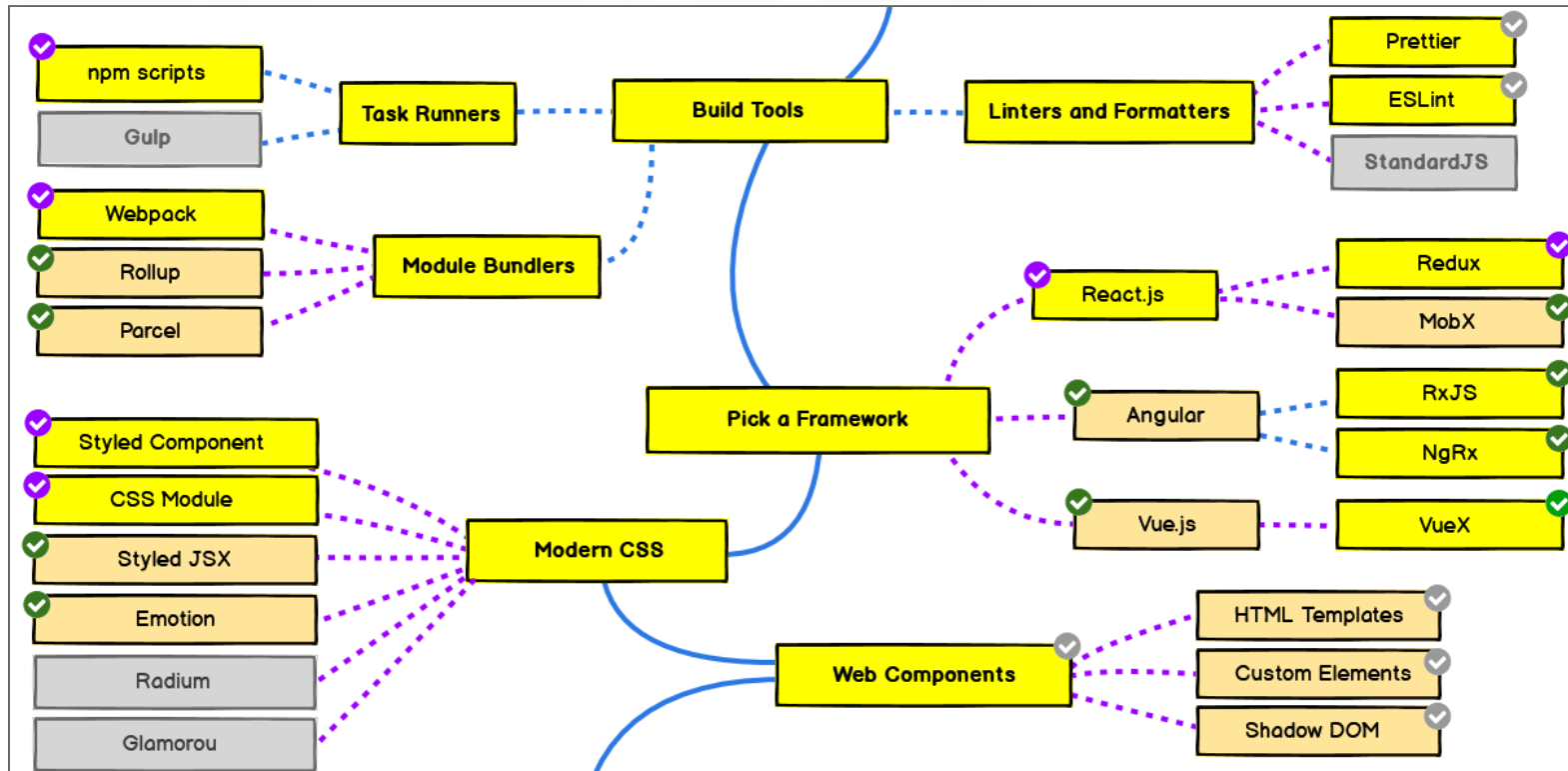
## **DEMASIADAS HERRAMIENTAS Y FRAMEWORKS JS:**

**[https://en.wikipedia.org/wiki/List of JavaScript libraries](https://en.wikipedia.org/wiki/List_of_JavaScript_libraries)**





# ELECCIÓN DE UN FRAMEWORK JS: FRONT-END PATH



## Developer Roadmap

**¿POR QUÉ EXISTEN LOS FRAMEWORK JS?**

**LA VERDADERA RAZÓN:**

**KEEPING THE UI  
IN SYNC  
WITH THE STATE  
IS HARD**

[Artículo Medium](#)

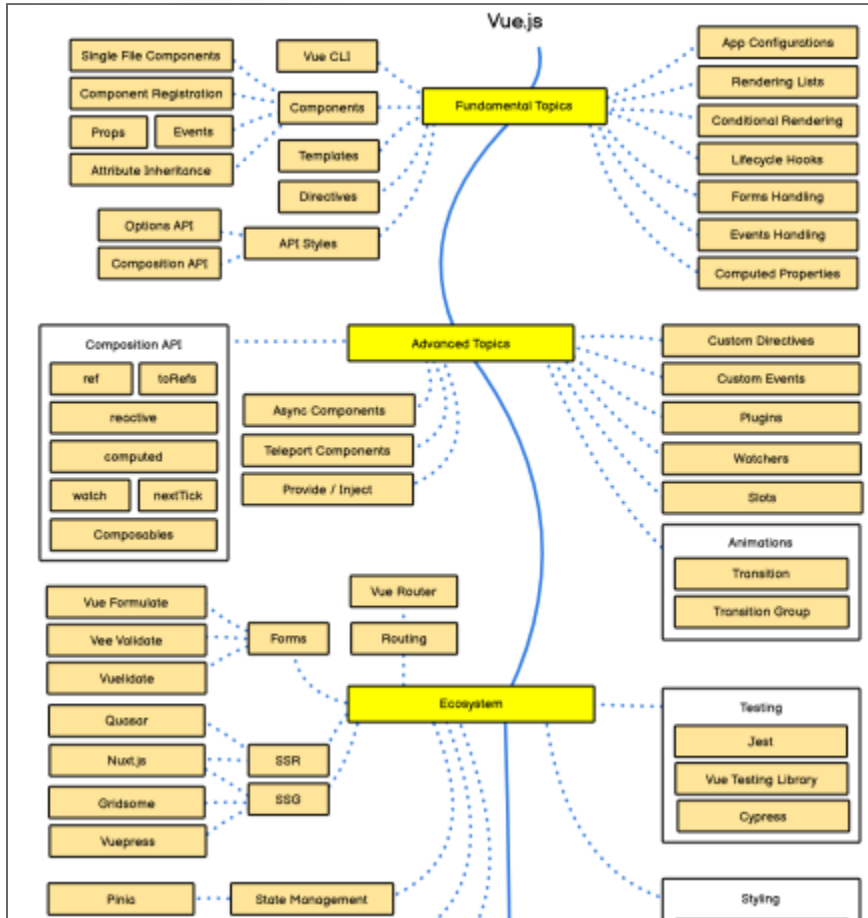


## PARA SEGUIR LEYENDO: JS FRAMEWORKS

- Comparación de Frameworks JS:  
<https://tinyurl.com/comparacionLibsJS>
- <https://platzi.com/blog/stack-javascript-2020/>
- <https://2021.stateofjs.com/en-US/libraries>
- [https://developer.mozilla.org/es/docs/Learn/Tools and testing/Client-side JavaScript frameworks](https://developer.mozilla.org/es/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks)
- <https://codersera.com/blog/best-javascript-frameworks/>

**VUE.JS**

# VUE.JS ROADMAP



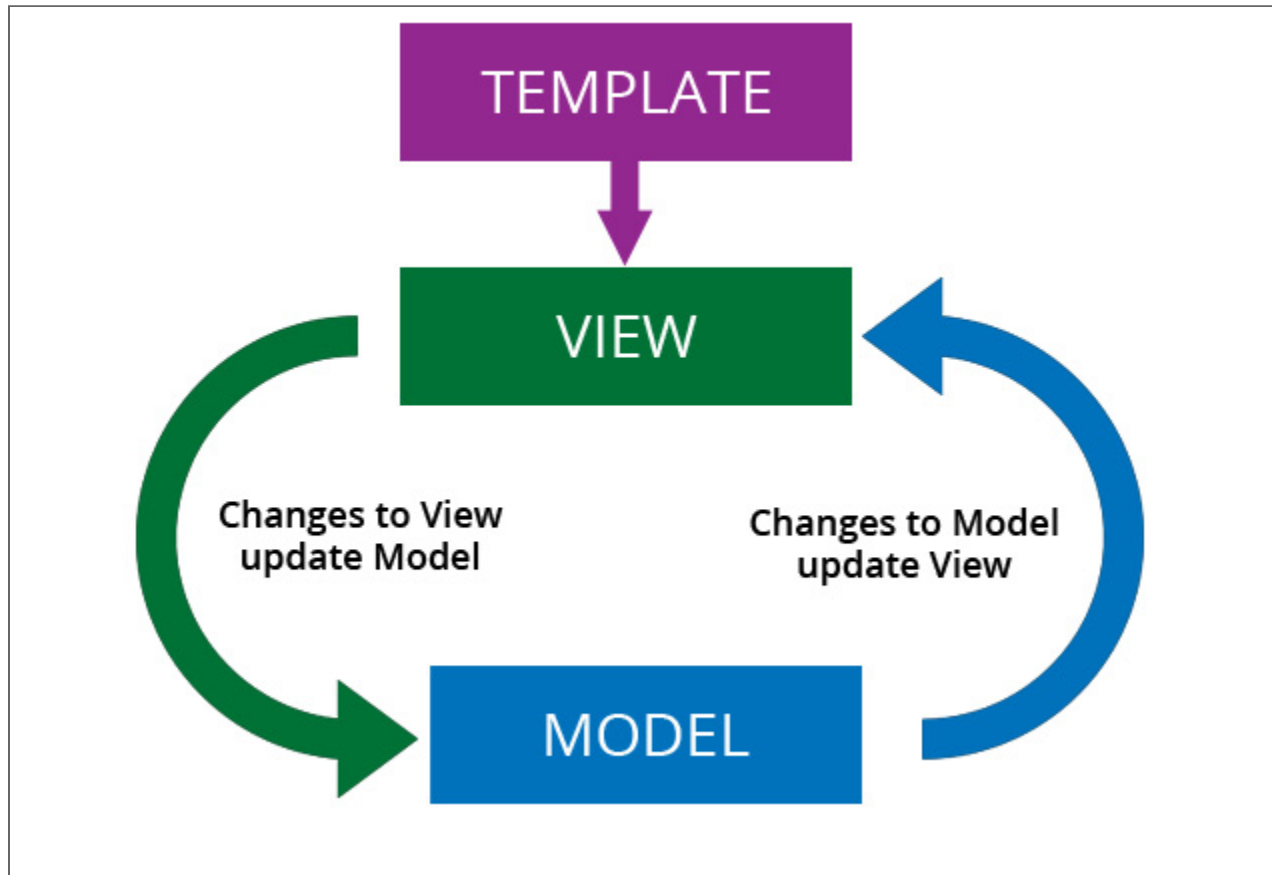
<https://roadmap.sh/vue>

# CARACTERÍSTICAS

- Vue.js es un framework de JavaScript para front-end.
- Facilidad de aprendizaje y uso con respecto a otros frameworks como **ReactJS**.
- Mejor rendimiento comparado con **AngularJS**.
- Vue.js es un framework **progresivo**. Tiene la facilidad para usarlo y adaptarlo a proyectos tanto grandes como pequeños.
- Ha tenido un gran crecimiento. Veamos su comunidad en **Github**.

# CARACTERÍSTICAS

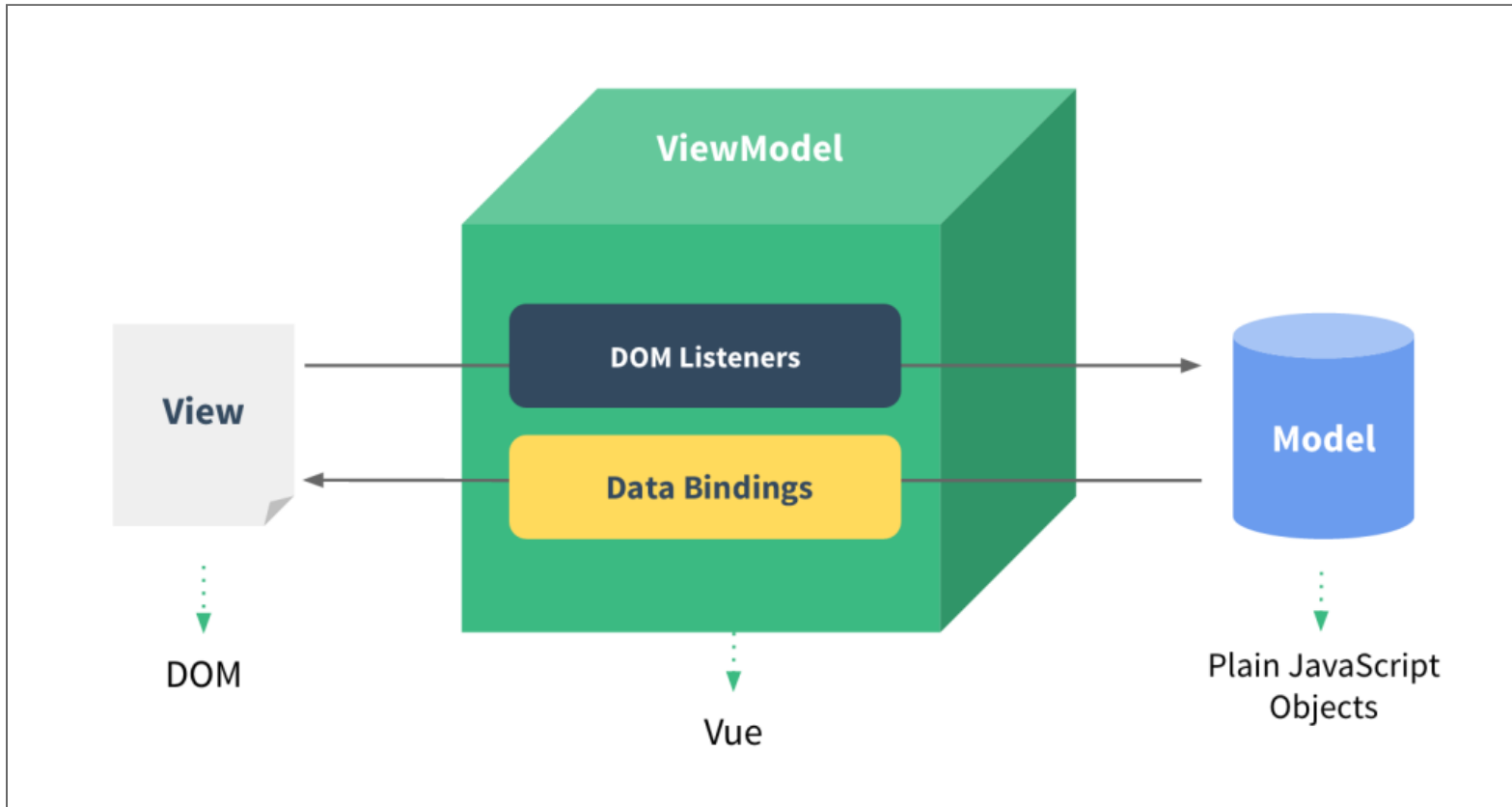
- Es un framework "reactivo" que implementa **"two way data-binding"**: enlace de datos en dos direcciones (entre la vista y el modelo) de una manera muy eficiente y rápida.





# CARACTERÍSTICAS

- Se basa en el patrón **Model-View-Viewmodel**



# CARACTERÍSTICAS

- Vue.js, está más enfocado hacia la vista, y **puede ser implementado en el HTML de cualquier proyecto web** sin requerir cambios drásticos .
- Los navegadores modernos poseen la extensión **Vue.js devtools** que nos asiste en el desarrollo.
- Vue.js soporta todos los browsers que sean compatibles con **ES5-compliant**. (Ver **Versiones JS**.)



# INSTALACIÓN VUEJS:

- Descargando el js e incluyéndolo directamente en un tag `<script>`.
- Linkear directamente desde una **CDN** (Content Delivery Network).

Vue 2:

```
<script src="https://cdn.jsdelivr.net/npm/vue@2"></script>
```

Vue 3:

```
<script src="https://unpkg.com/vue@3/dist/vue.global.js">  
</script>
```

# INSTALACIÓN VUEJS:

- Instalar Vue CLI via **npm** (manejador de paquetes por defecto para Node.js):

```
> npm install -g @vue/cli
> vue create <your-project-name>
. . .
> cd <your-project-name>
> npm install
> vue serve
o
> npm run dev
```

Esta es la opción recomendada para para **proyectos más grandes**.

# INCLUYENDO DE UNA CDN:

Veamos hello world.

```
<!DOCTYPE html>
<html>
<head>
  <title>Mi primera aplicaci&ocute;n Vue</title>
  <script
src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
</head>
<body>
  <div id="app">
    <h2>{{ message }}</h2>
  </div>

  <script>
    const HelloVueApp = {
      data() {
        return {
          message: 'Hola Vue!!!'
        }
      }
    }
    vue_app = Vue.createApp(HelloVueApp).mount('#app')
  </script>
</body>
</html>
```

Los datos y DOM están ahora relacionados utilizando `{{}}`.  
Modifiquemos `vue_app.message`.

**¿DUDAS?**

**SEGUIMOS CON VUEJS EN VIDEO Y LA PRÓXIMA SEMANA...**