

PROYECTO DE SOFTWARE

TEMARIO

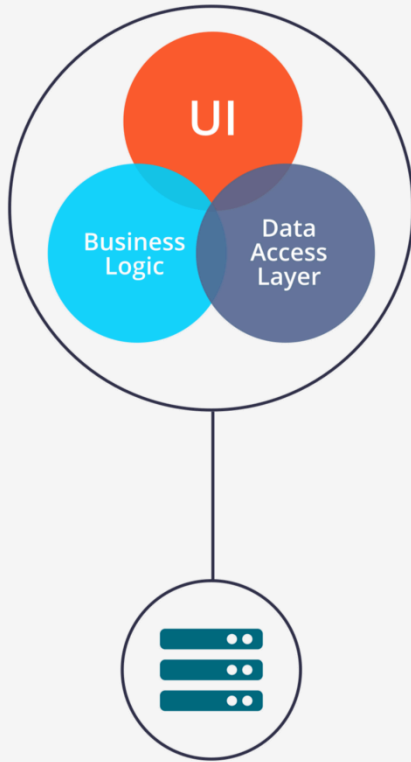
- Microservicios
- Frameworks JS
- Intro Vue

ARQUITECTURA DE MICROSERVICIOS

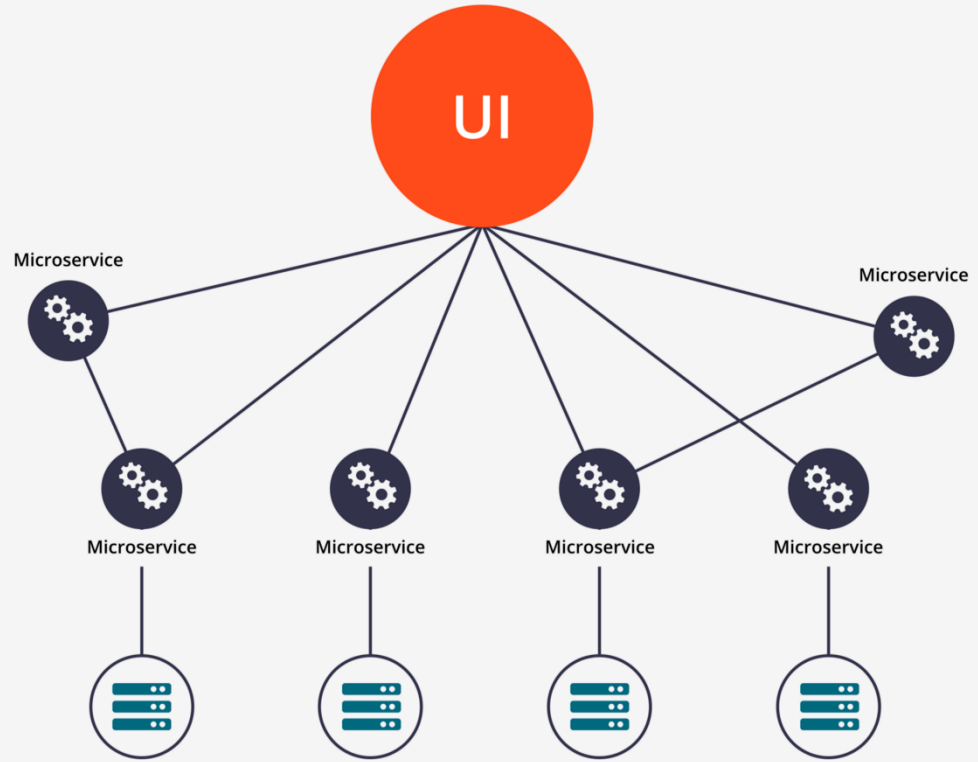
MICROSERVICIOS

- 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



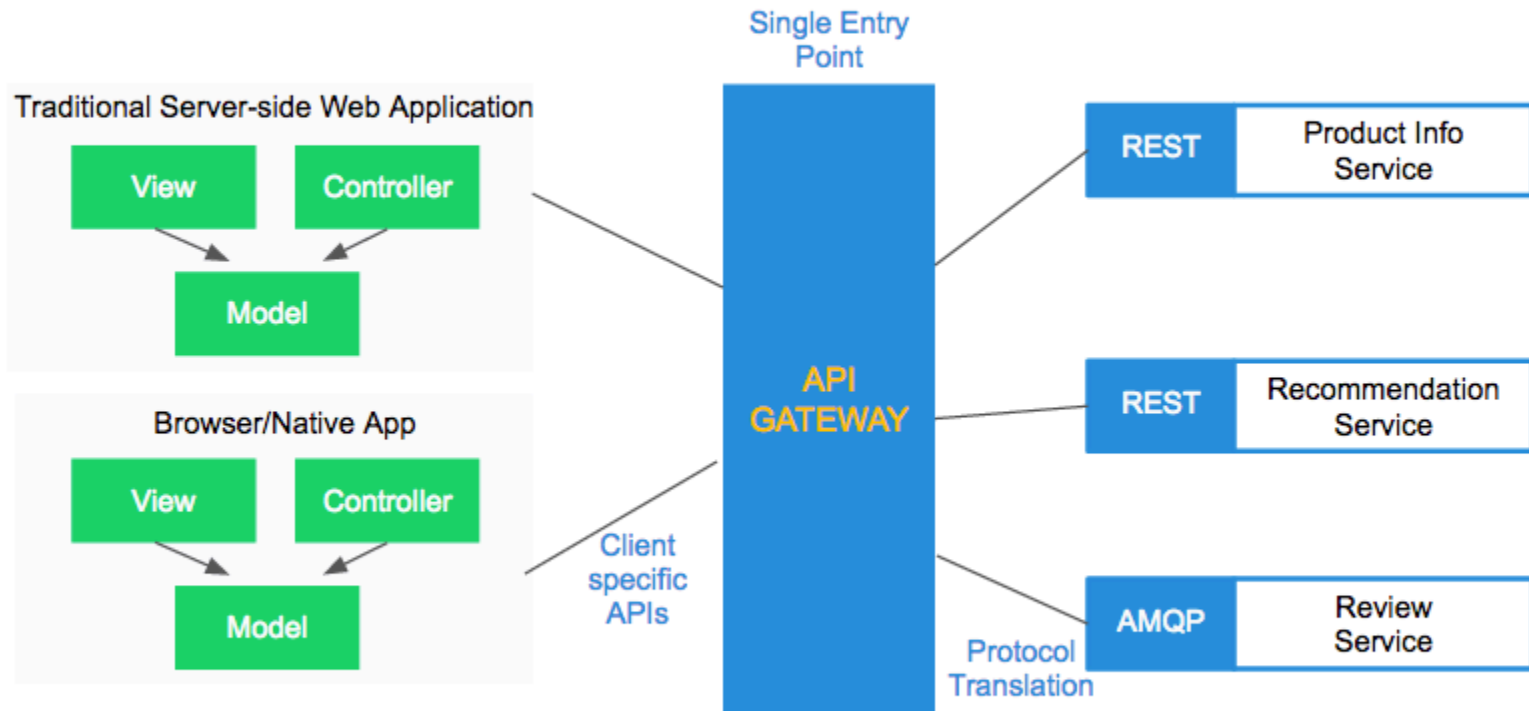
Microservice Architecture

¿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

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

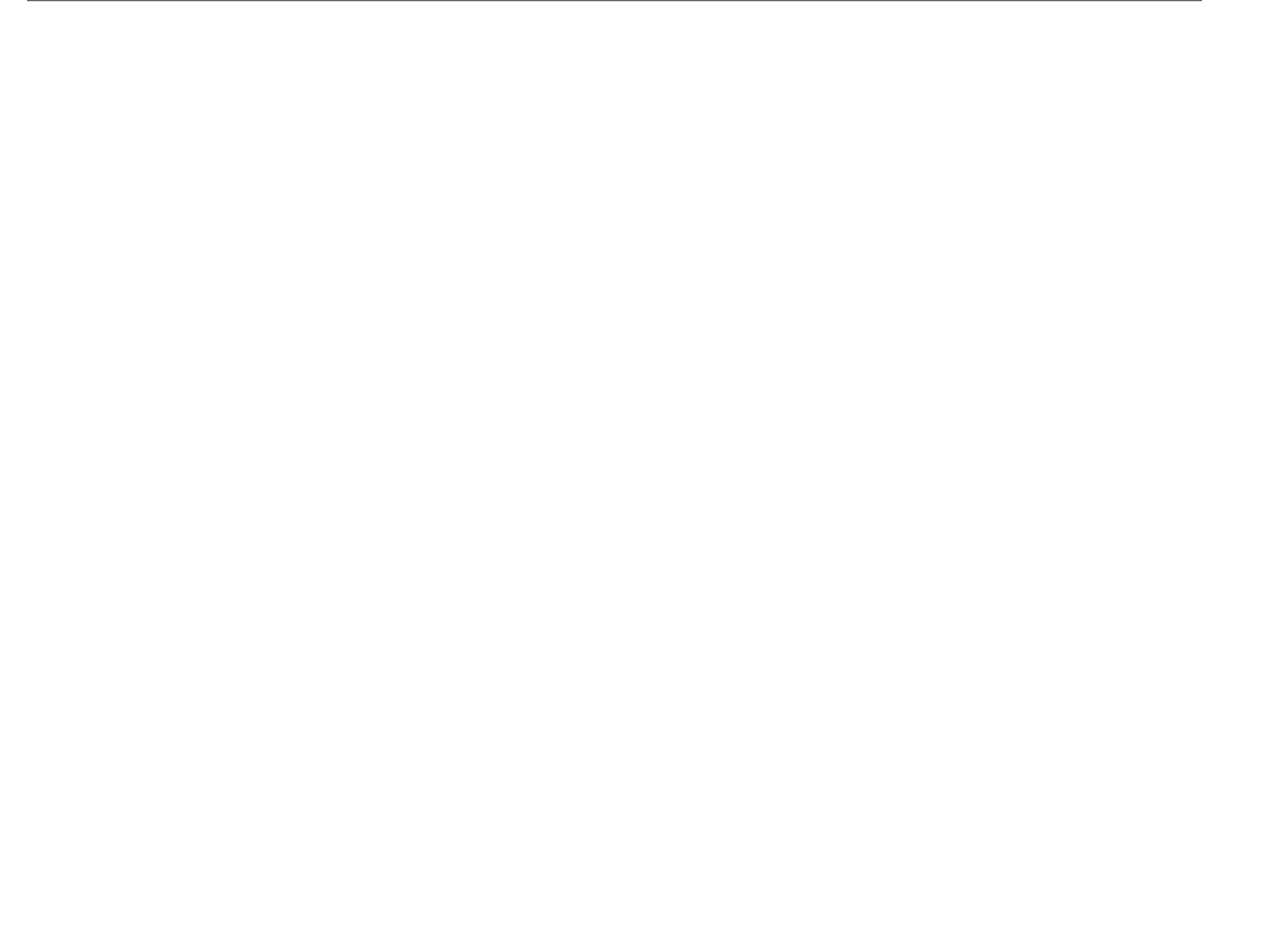
¿QUÉ ES UN FRAMEWORK JS?

- Trabajar con **JS pelado** es complejo. Por eso nació **jQuery**(2006) para facilitar el desarrollo.
- jQuery sigue estando muy extendido. **Stackoverflow survey** 2021, 2022, 2023 **y 2024**.
- 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.

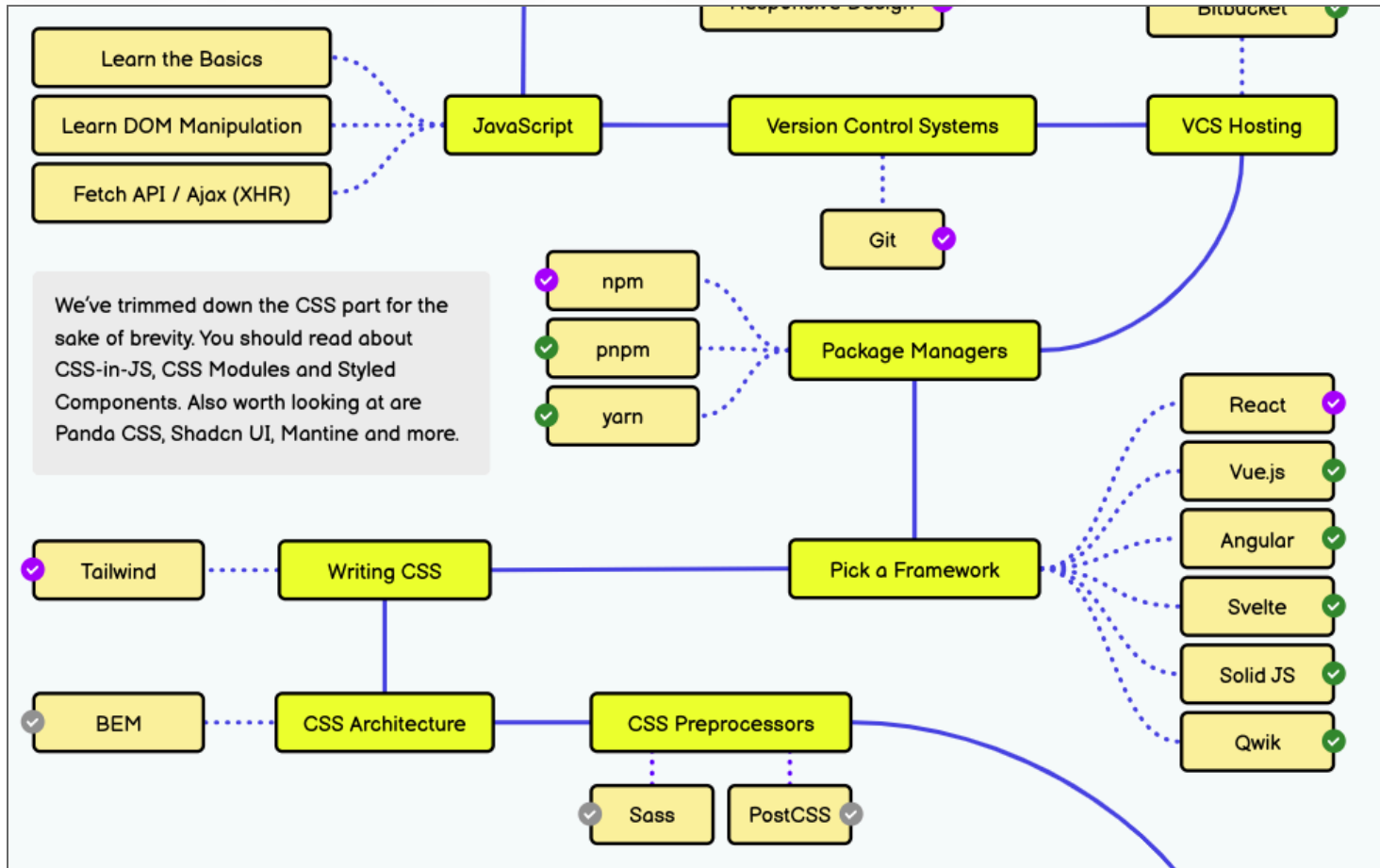
DEMASIADAS HERRAMIENTAS Y FRAMEWORKS JS:

[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

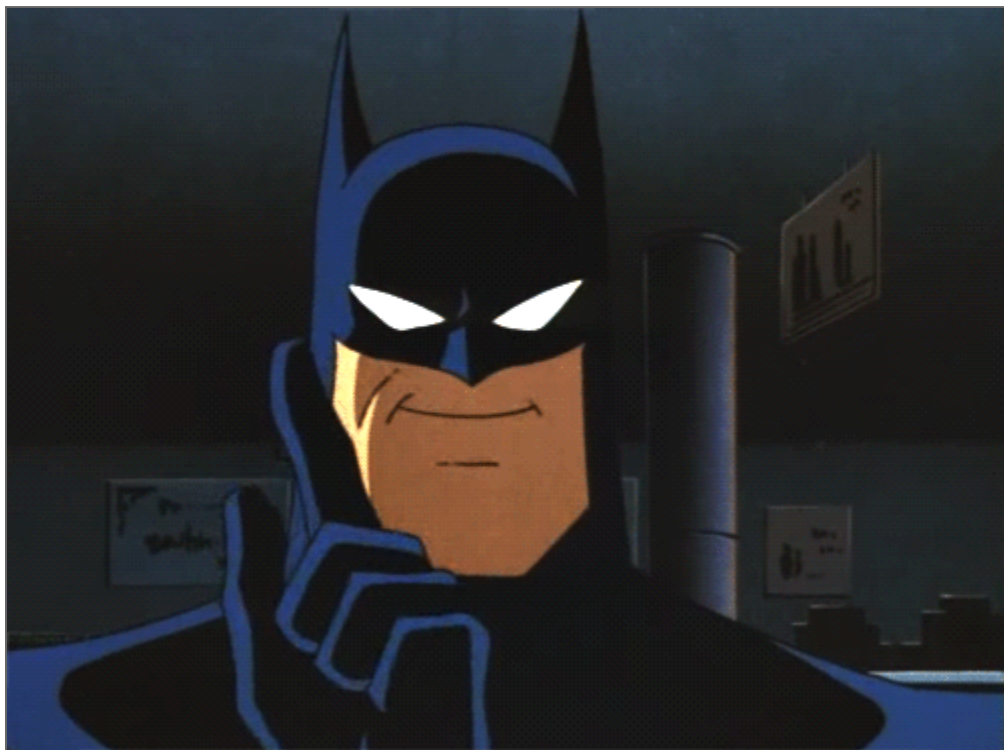


Developer Roadmap

¿POR QUÉ EXISTEN LOS FRAMEWORK JS?

SERÁ PORQUE ...

- Están basados en componentes.
- Tienen una comunidad sólida.
- Tienen muchas bibliotecas y componentes de terceros para manejar diferentes tareas.
- Tienen extensiones de navegador que ayudan a depurar.
- Son buenos para crear aplicaciones SPA (single page applications).
- Poseen una buena interfaz de línea de comandos (CLI) para facilitar el desarrollo.
- Realizan optimizaciones para mejorar el rendimiento.



LA VERDADERA RAZÓN:

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

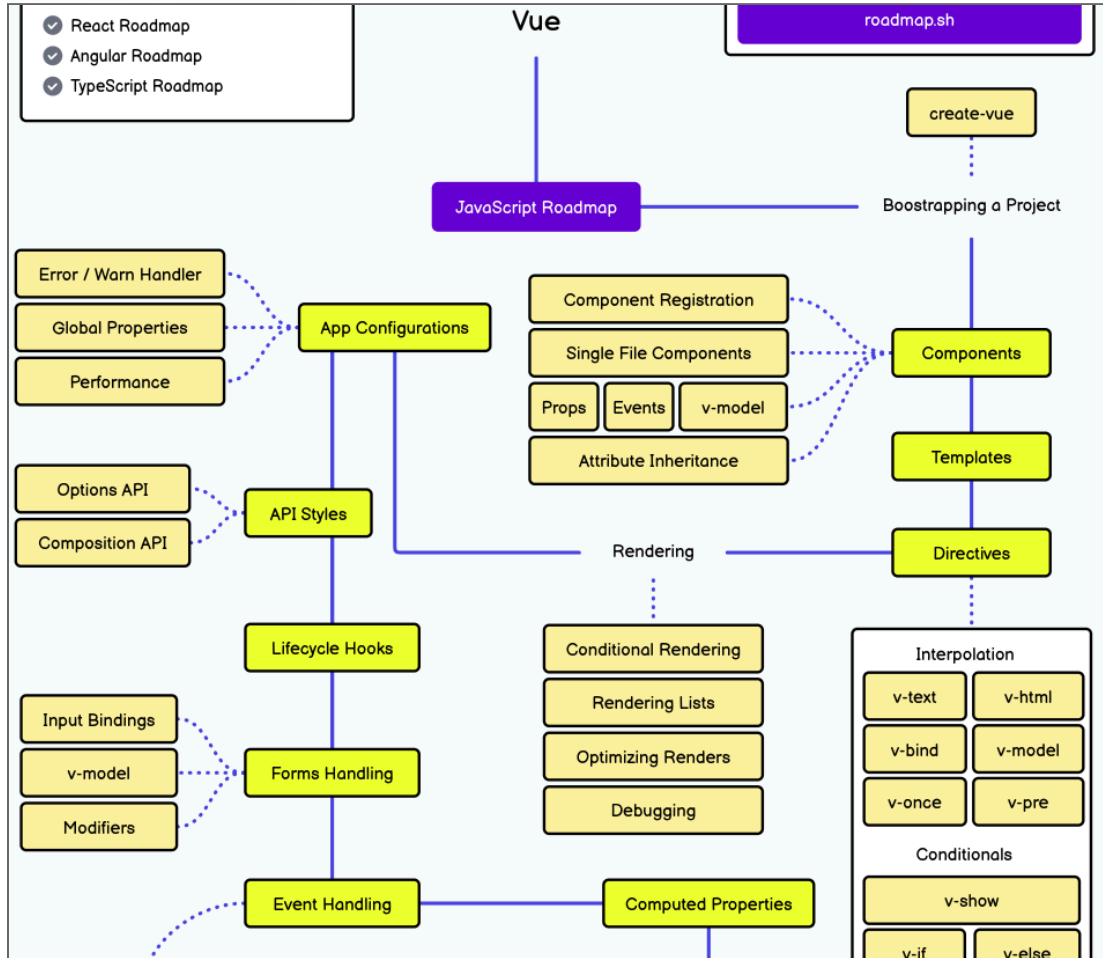
[Artículo Medium](#)

PARA SEGUIR LEYENDO: JS FRAMEWORKS

- Mejor Framework JS (2021): <https://codersera.com/blog/best-javascript-frameworks/>
- Top 10 JS Frameworks 2024: https://dev.to/delia_code/top-10-javascript-frameworks-to-learn-in-2024-41mn
- JS Frameworks Benchmark 2024: <https://dev.to/sfestus90/javascript-web-frameworks-benchmark-2024-an-in-depth-analysis-30om>
- State of JS: <https://stateofjs.com/> - últimas ediciones: 2022,2023
- Entendiendo los frameworks JS:
[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)

VUE.JS

VUE.JS ROADMAP



<https://roadmap.sh/vue>

<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 grán 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.

TEMPLATE



VIEW



Changes to Model
update View

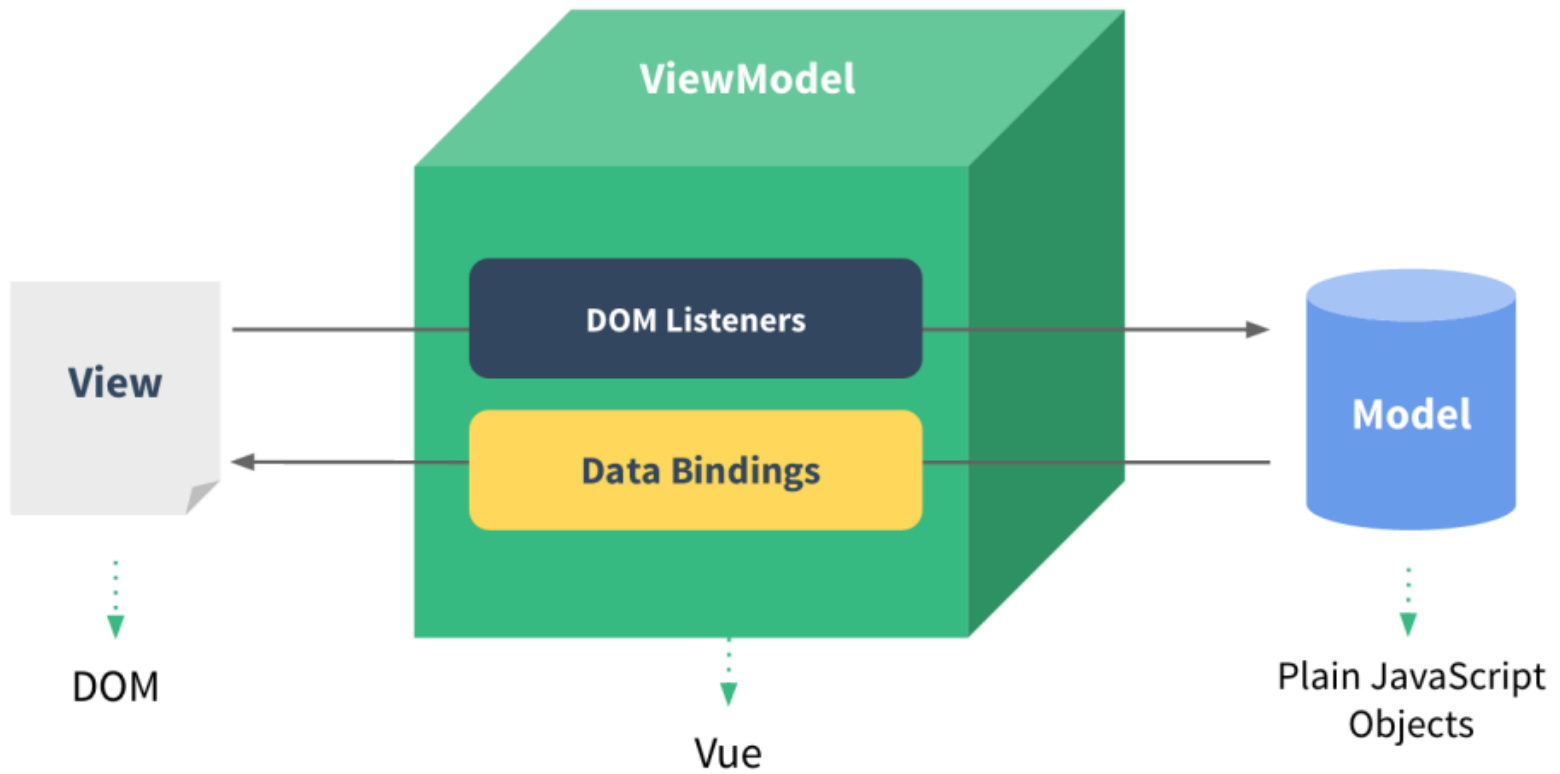
MODEL

Changes to View
update Model



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** **devtools.vuejs.org** 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 3 (Versión actual):

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

Vue 2 (Versión LST - finalizó el 31/12/2023):

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

No recomendada para nuevos proyectos.

INSTALACIÓN VUEJS:

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

```
npm create vue@latest
```

Se utiliza un setup guiado basado en Vite.

```
cd NOMBRE_PROYECTO  
npm install  
npm run dev
```

- Esta es la opción recomendada para para **proyectos más grandes**.
- Node.js versión ≥ 18.3 requerida.

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...

Speaker notes