Proyecto de Software

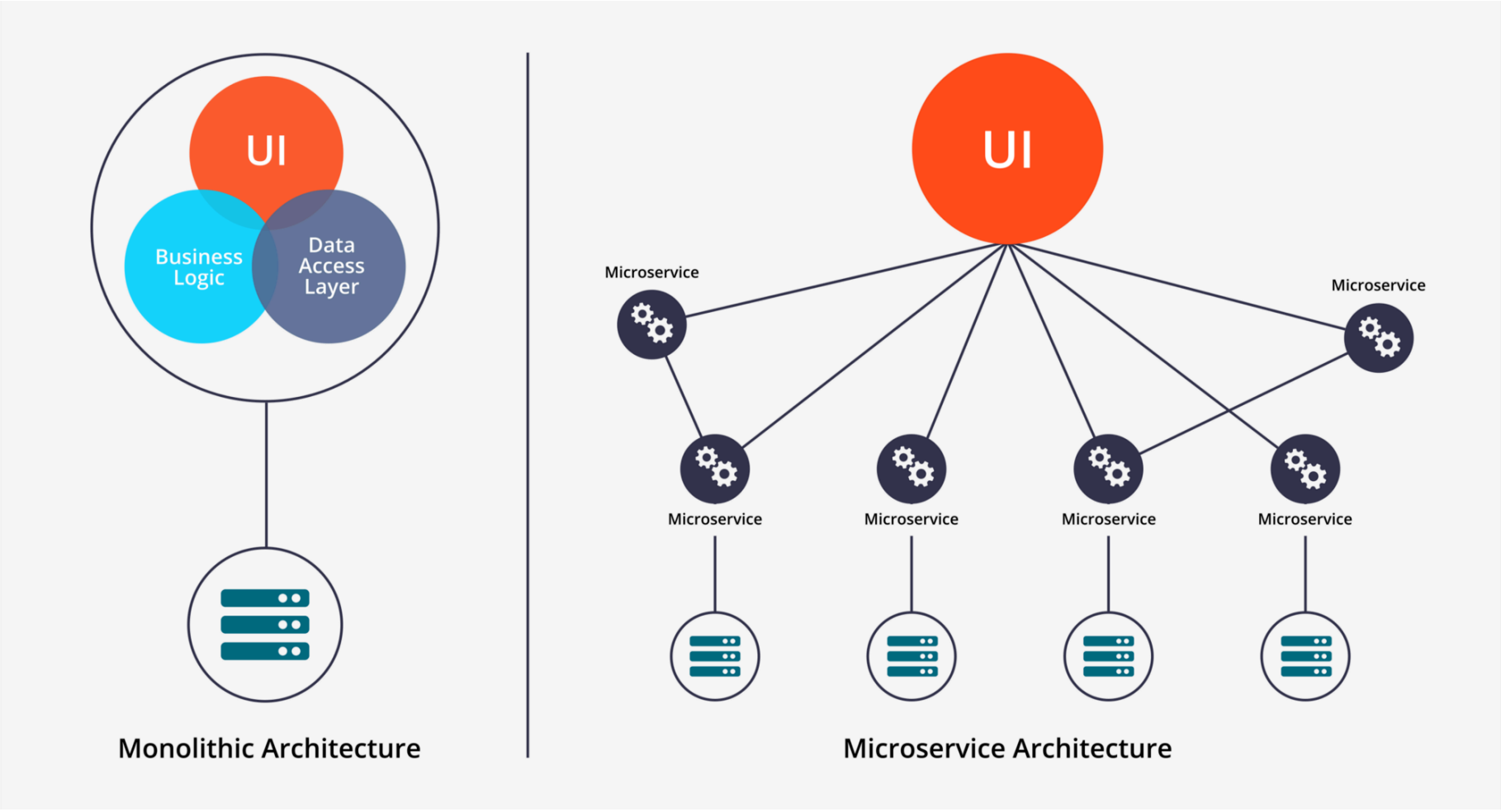
- Microservicios
- Frameworks JS
- Intro Vue

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.

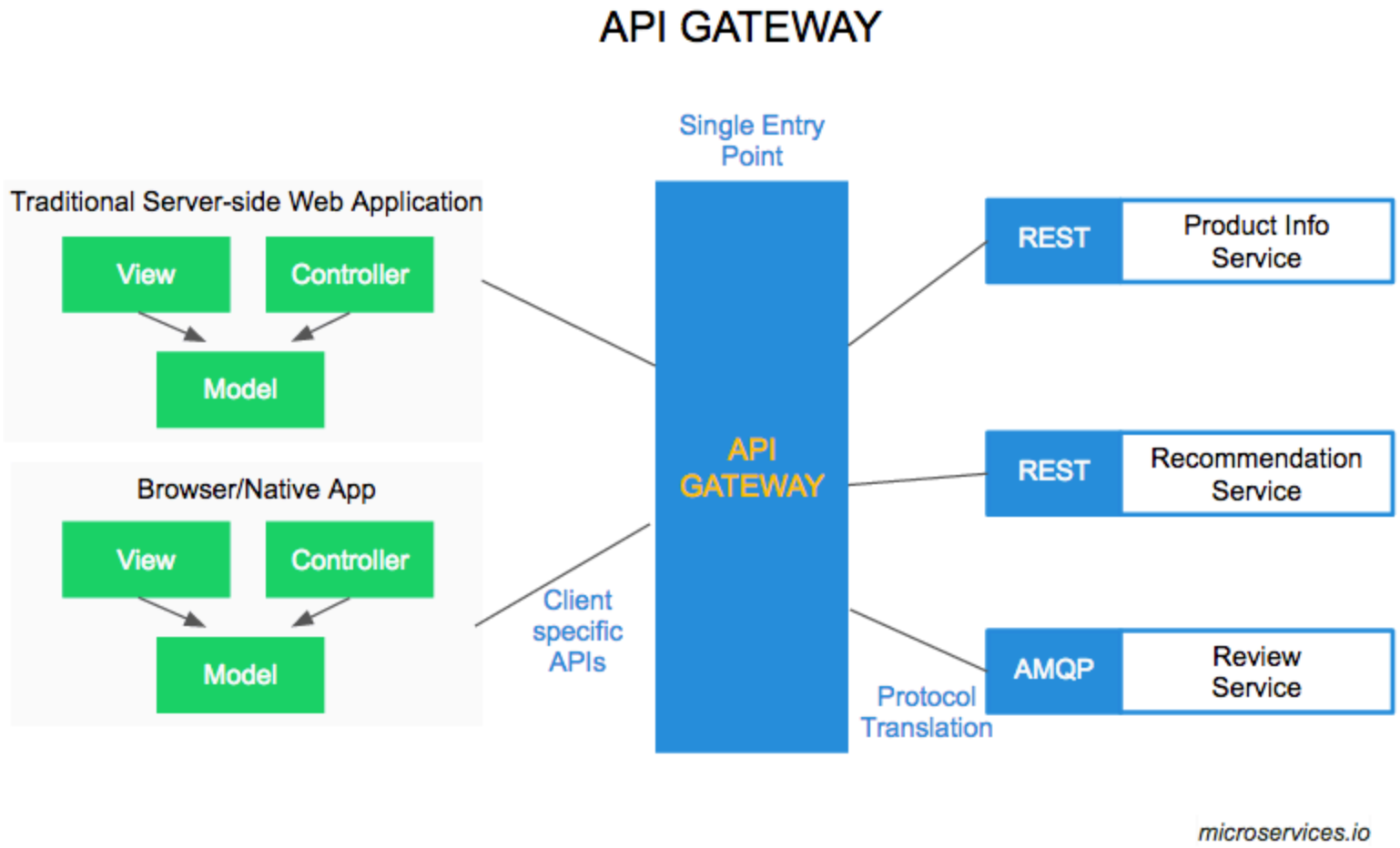
Arquitectura Monolítica vs Arquitectura de Microservicios



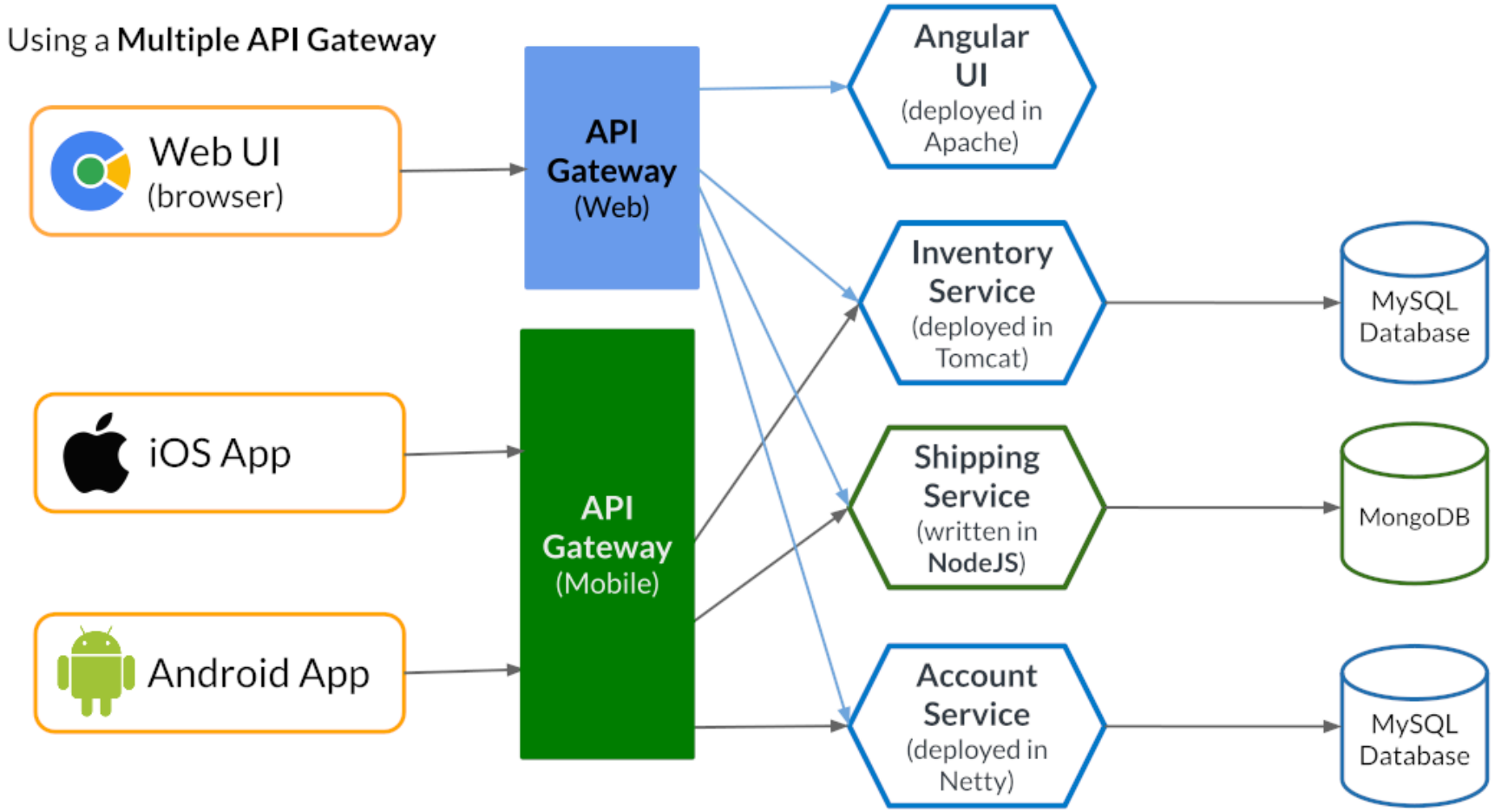
¿Cómo comunico un servicio con otro?

- 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 petisión verifique si el cliente tiene permisos de acceso.

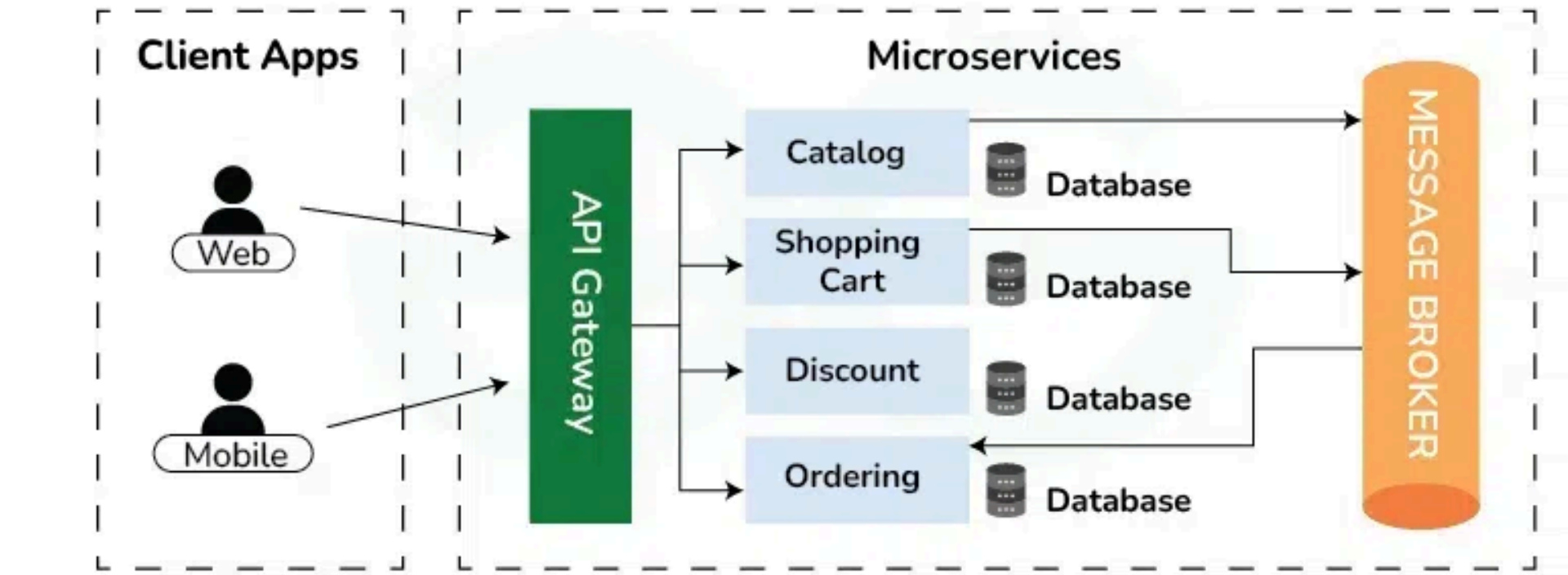
Microservicios: API Gateway



Microservicios: Múltiples API Gateway



Microservicios: Message Broker



Microservices Architecture

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

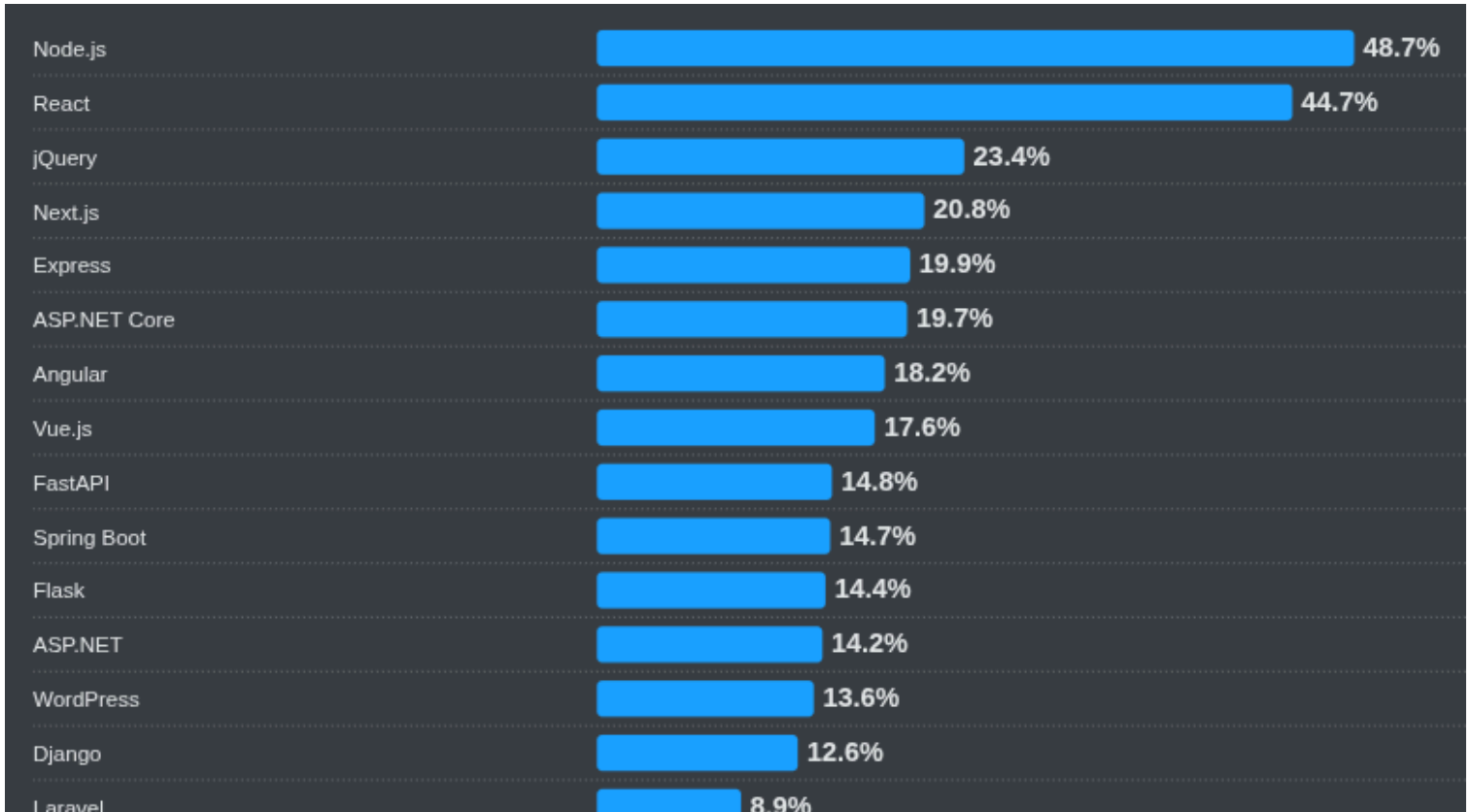
Frameworks JS

¿Qué és un framework JS?

- Trabajar con **JS pelado** es complejo. Por eso nació **jQuery**(2006) para facilitar el desarrollo.
- jQuery sigue estando **muy** extendido.
- 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.

Stackoverflow survey

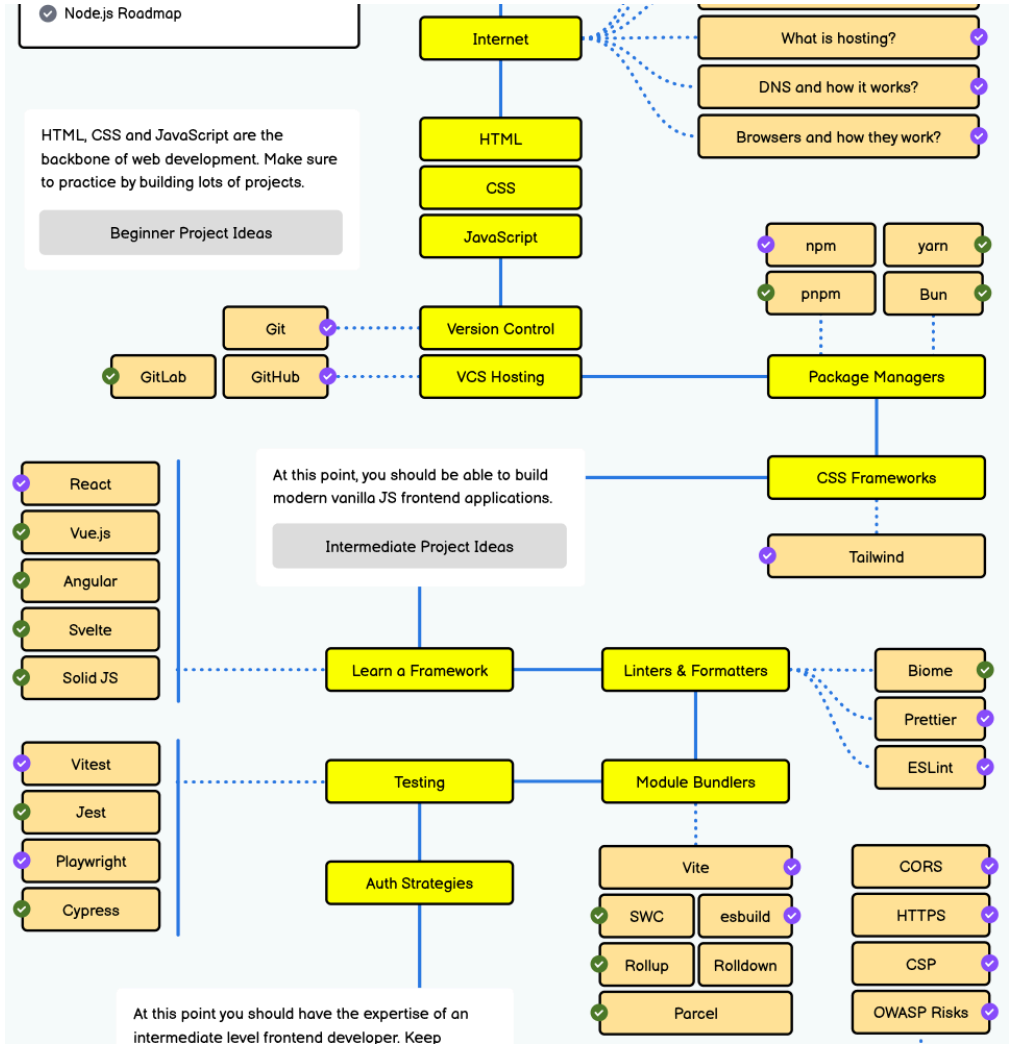
- https://survey.stackoverflow.co/
- Resultados de frameworks y tecnologías Web
  - 2021 | 2022 | 2023 | 2024 | 2025



Demasiadas herramientas y frameworks JS:

https://en.wikipedia.org/wiki/List_of_JavaScript_libraries

Elección de un framework JS: Front-end path

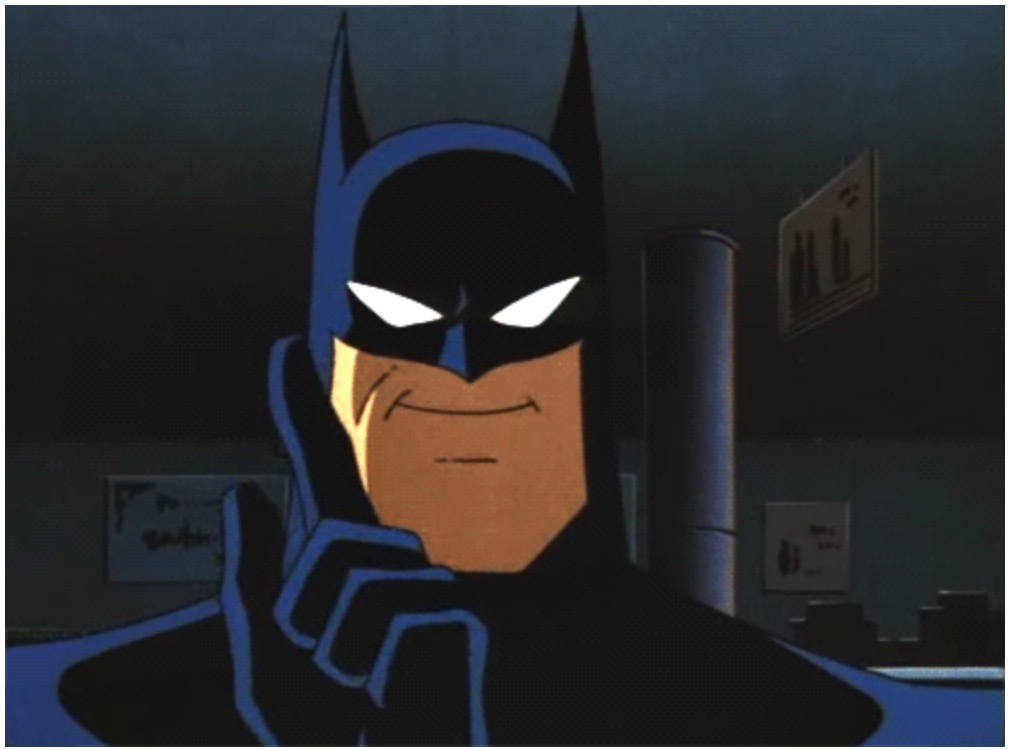


Developer Roadmap

¿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:

Artículo Medium

Angular vs. React vs. Vue

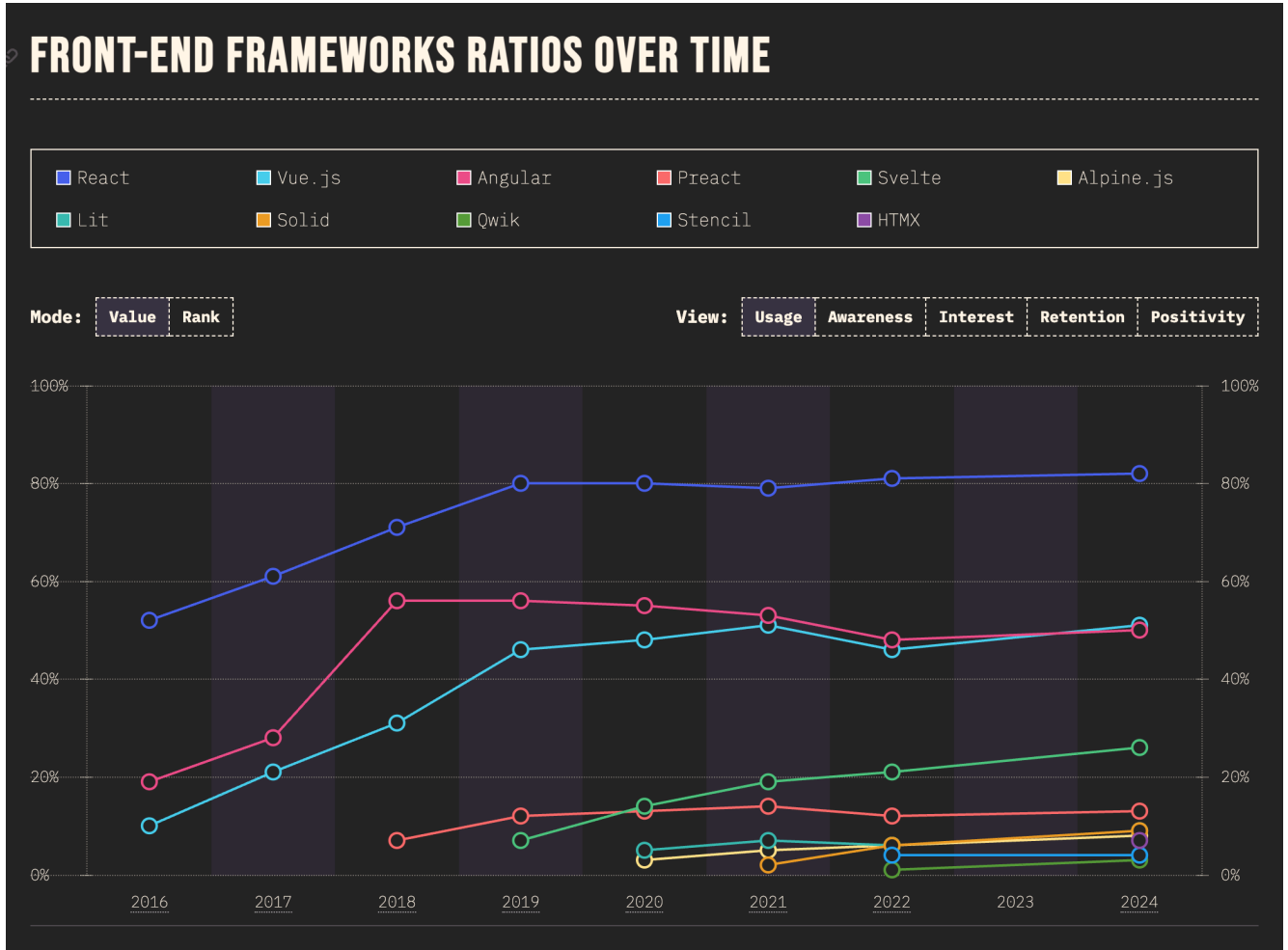
Factors	Angular	React	Vue
Learning Curve	Significant, due to TypeScript and complexity	Moderate, JSX may be unfamiliar initially	Easiest, HTML-based syntax, progressive
Architecture	Prescriptive, "batteries-included"	Modular, based on unidirectional data flow	Progressive, component-based
Data Binding	Two-way binding, automated synchronization	One-way binding, component rendering triggered by data	One-way and two-way binding, flexible
Mobile Solutions	Ionic framework for complete mobile development	React Native for native mobile apps	Quasar Framework and Vue Native
User Friendliness	Rigid, TypeScript learning curve	Simple, component-based, flexible	Easiest, progressive, HTML familiarity
Syntax	Declarative template syntax, TypeScript integration	JSX for HTML-like code, supports plain JavaScript	Template-based, HTML-aligned, modular Integration
Integration	Tightly integrated, full-stack features	Seamless integration, component-based	Incremental adoption, flexible
Performance	Two-way data binding may impact large apps	Virtual DOM for efficient rendering	Virtual DOM, reactivity system
Ecosystem	Full-stack features, TypeScript ecosystem	Vibrant community, extensive ecosystem	Progressive, flexible, active community
Community and Support	Strong, established community, Google-backed	Large, active community, Facebook-backed	Welcoming, approachable, active community
Migration	Incremental upgrades, challenges between major versions	Seamless, backward compatibility	Incremental adoption, backward compatibility

Para seguir leyendo: JS Frameworks

- Reactividad en JS (2021)
- Mejor Framework JS (2021)
- Top 10 JS Frameworks 2024
- JS Frameworks Benchmark 2024
- Entendiendo los frameworks JS
- Comparativa React vs. Angular vs. Vue (2019)

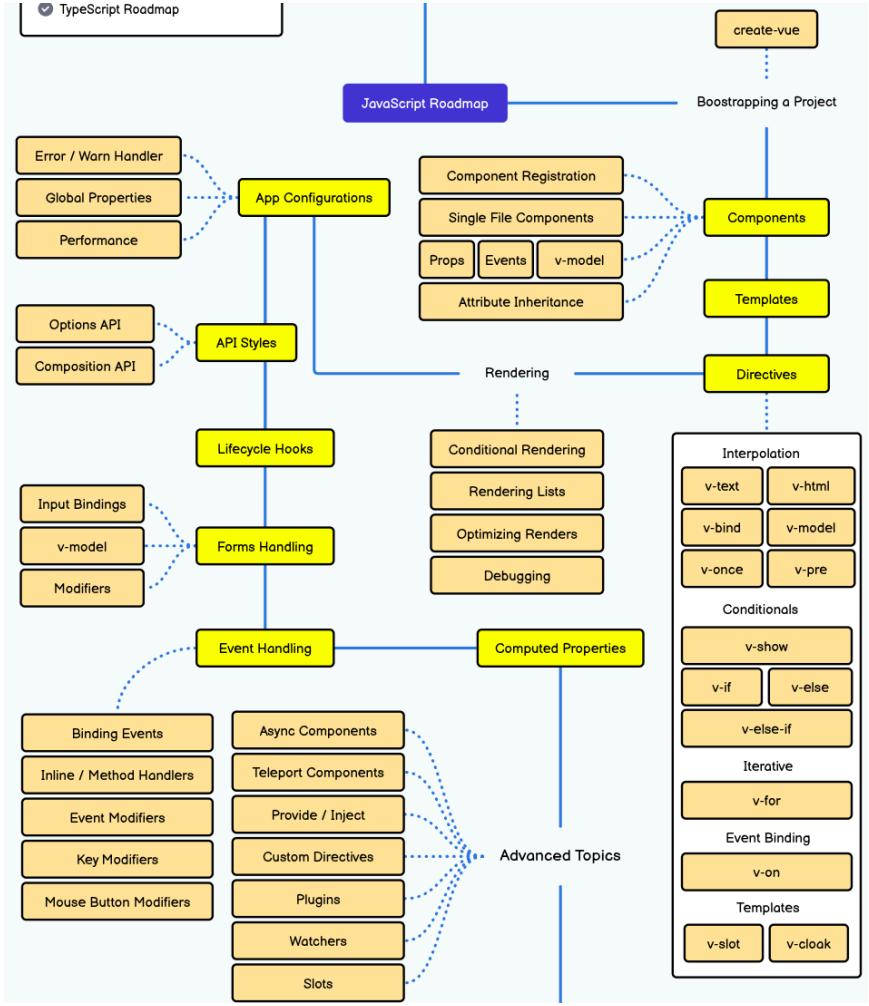
State of JS Survey

- https://stateofjs.com/
- Últimas ediciones: 2022, 2023, 2024



Vue.js

Vue.js roadmap



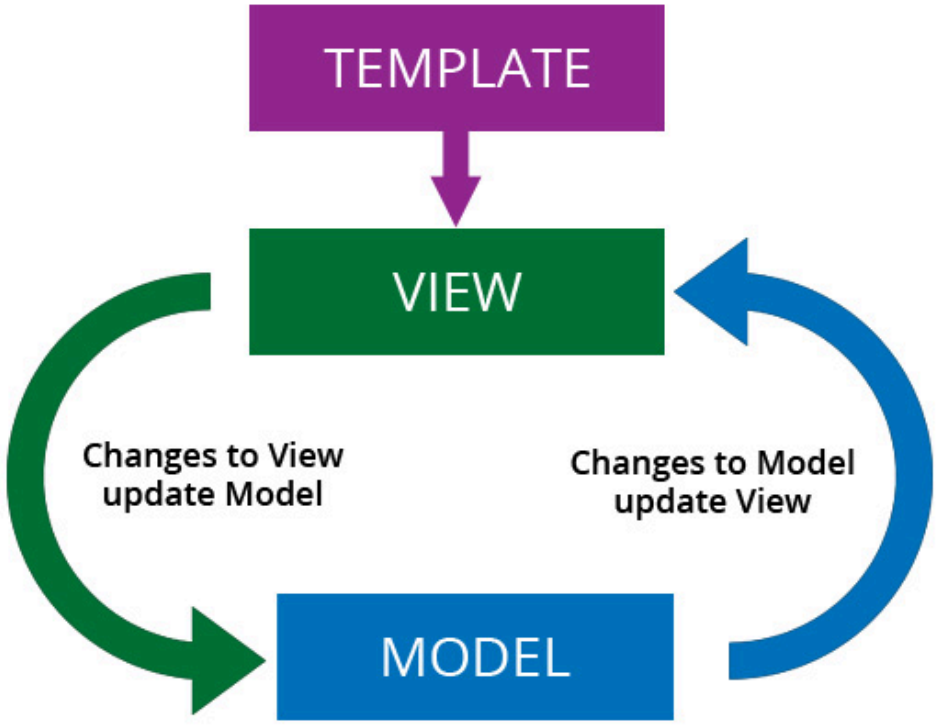
https://roadmap.sh/vue

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 grán crecimiento. Veamos su comunidad en Github.

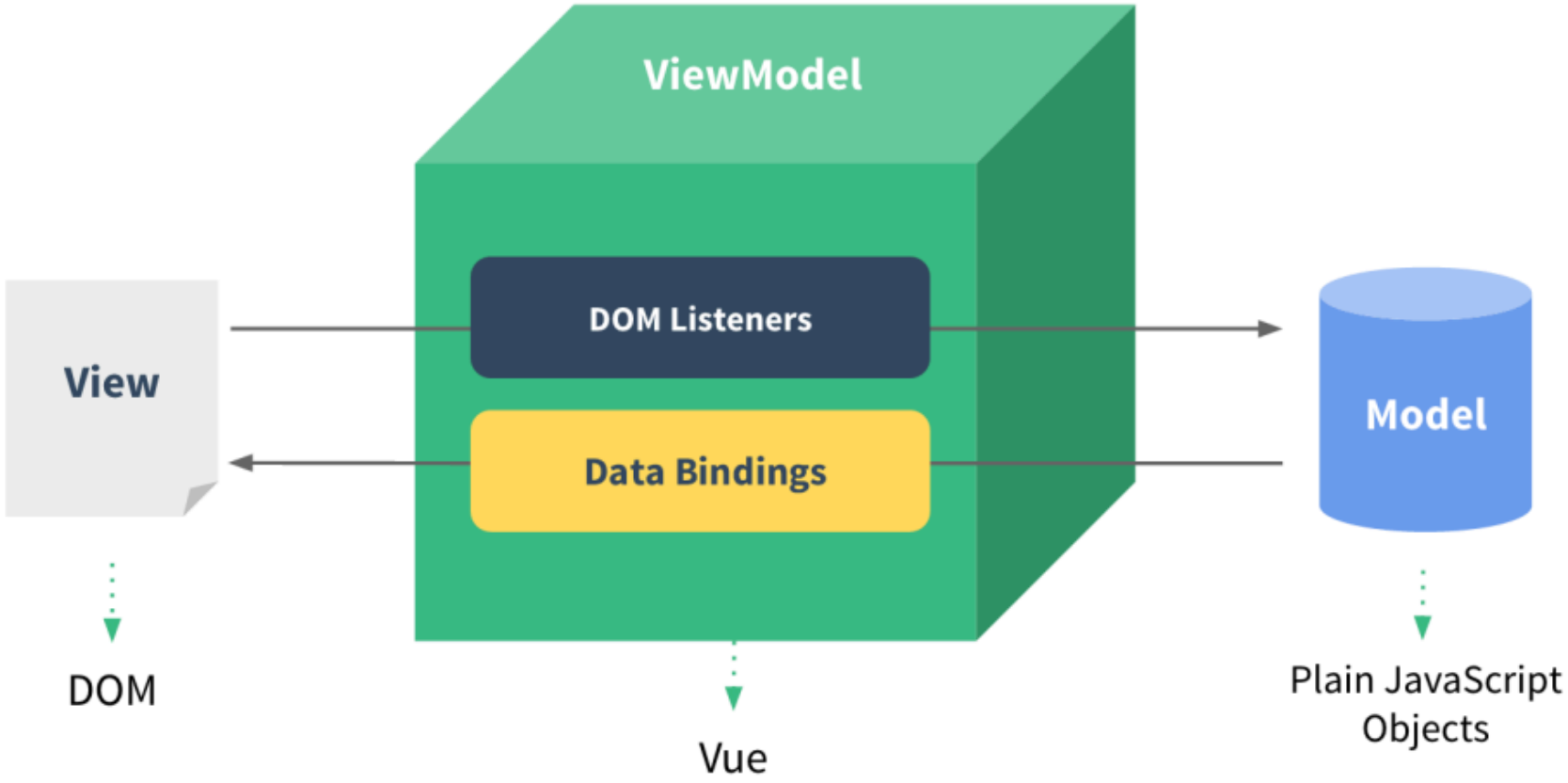
Características

- 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.



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**) - NO UTILIZAR:

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

Instalación VueJS:

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

```
npm create vue@latest
```

Se utiliza un setup guiado basado en Vite una herramienta para contruir código web moderno.

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

- Esta es la opción recomendada para para **proyectos más grandes**.
- Node.js versión requerida **^20.19.0 || >=22.12.0**.

Incluyendolo de una CDN:

Veamos hello_world. Los datos y DOM están ahora relacionados utilizando **{{}}**.

Modifiquemos **vue_app.message**.

```
  <title>Mi primera aplicaci&oacute;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>
```

Ejemplo Two way data-binding:

- Hace la relación bidireccional entre un input y los datos de la aplicación Vue.
- Veamos v-model.

```
<div id="two-way-binding">
  <p>{{ message }}</p>
  <input v-model="message">
</div>
<script>
  const App = {
    data() {
      return {
        message: 'Bienvenidos a Proyecto!!'
      }
    }
  }
 vue_app = Vue.createApp(App).mount('#two-way-binding')
</script>
```

¿Dudas?

Seguimos con vuejs en video y la próxima semana...

Referencias Vue

- Guia Oficial Vue: https://vuejs.org/guide/introduction.html