Somos un ecosistema de desarrolladores de software

Vue JS

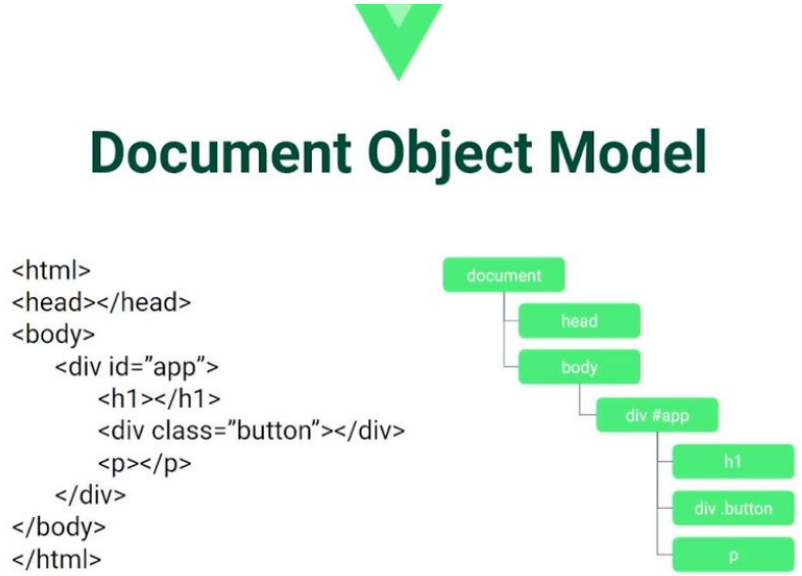
Vue.js

Vue 3

Vue.js es un framework progresivo para construir interfaces de usuario. Fue creado por Evan You y es utilizado para desarrollar interfaces de usuario (UI) y aplicaciones de una sola página (SPA). Su núcleo se centra solamente en la capa de vista, pero es fácil integrarlo con otras librerías o proyectos existentes

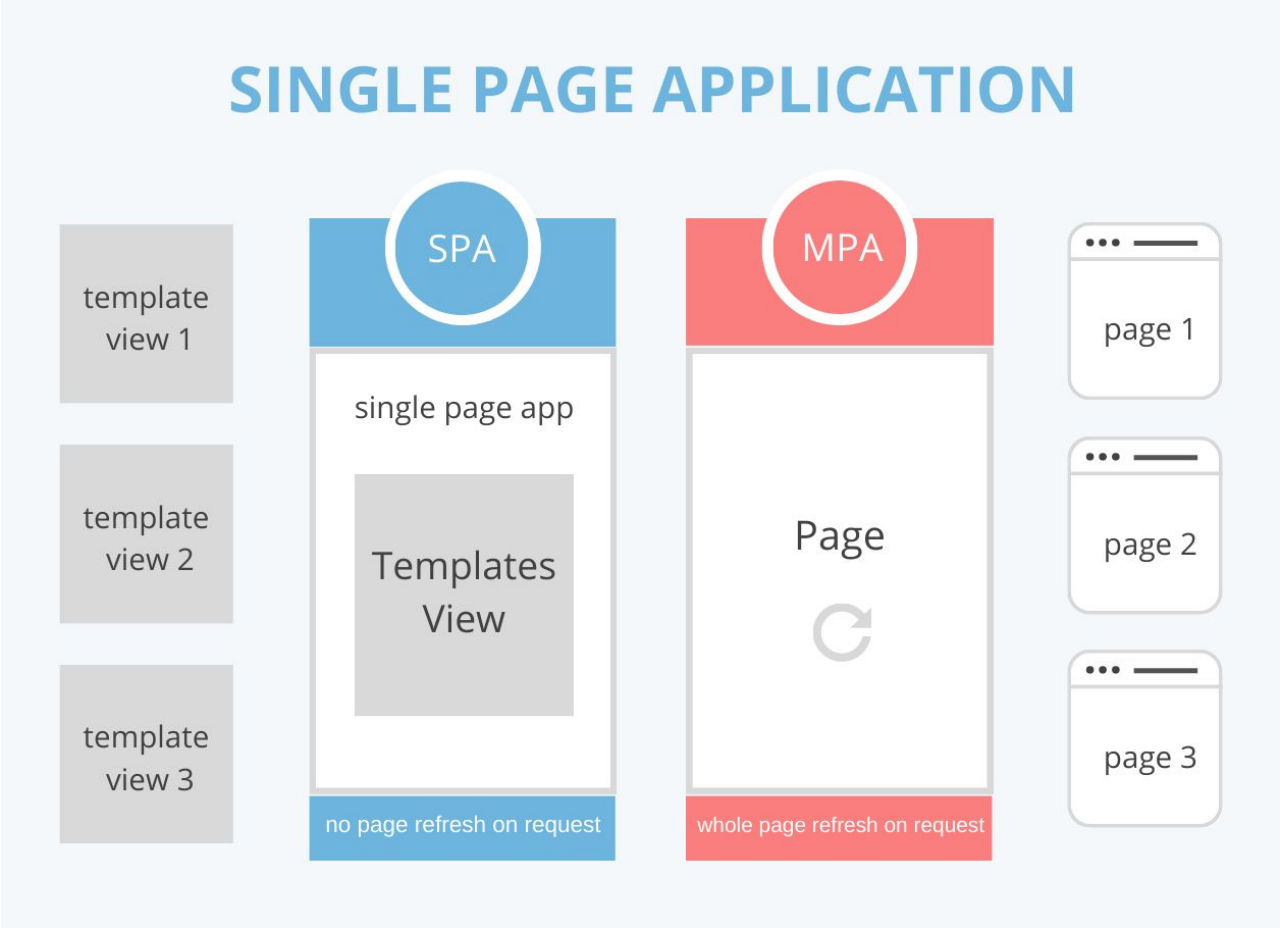
Características de Vue 3

- **Reactividad:** Vue maneja automáticamente la actualización de los componentes de la UI cuando los datos subyacentes cambian, lo que facilita el desarrollo.
- **Componentes:** Permite construir aplicaciones grandes y complejas a través de pequeños, autocontenidos y reutilizables componentes.
- **Enlace de datos bidireccional:** Vue facilita el enlace de datos bidireccional con su sintaxis v-model, lo que simplifica la manipulación de los formularios.
- **Transiciones y animaciones:** Vue proporciona varias maneras de aplicar transiciones a elementos HTML y componentes cuando son añadidos/actualizados/eliminados del DOM.
- **Directivas:** Ofrece una serie de directivas incorporadas como v-if, v-for, v-bind, y v-on, que permiten realizar tareas comunes con menos código.



Que es una SPA?

- Una SPA (Single Page Application, o Aplicación de Una Sola Página en español) es un tipo de aplicación web o sitio web que interactúa con el usuario reescribiendo dinámicamente la página actual con nuevos datos del servidor web, en lugar de cargar páginas web enteras nuevas.

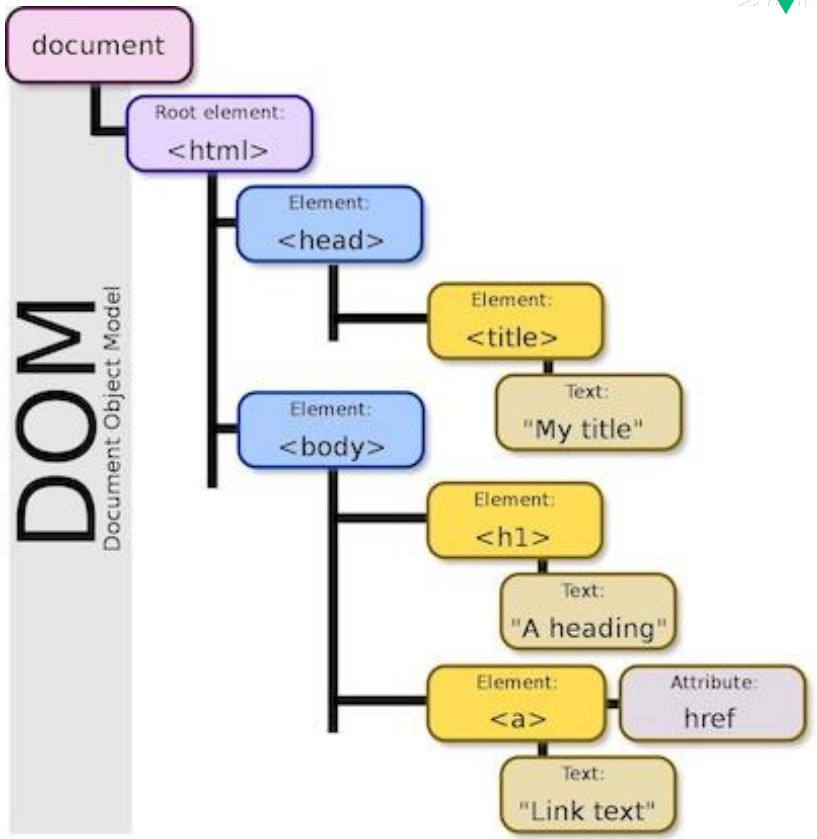


DOM Virtual

El DOM virtual en Vue.js es una parte fundamental de su arquitectura reactiva que ayuda a optimizar las actualizaciones de la interfaz de usuario y a mejorar el rendimiento general de las aplicaciones.

Funcionamiento:

1. Creación del DOM Virtual
2. Renderizado y Reactividad
3. Diffing del DOM Virtual
4. Parcheo del DOM Real



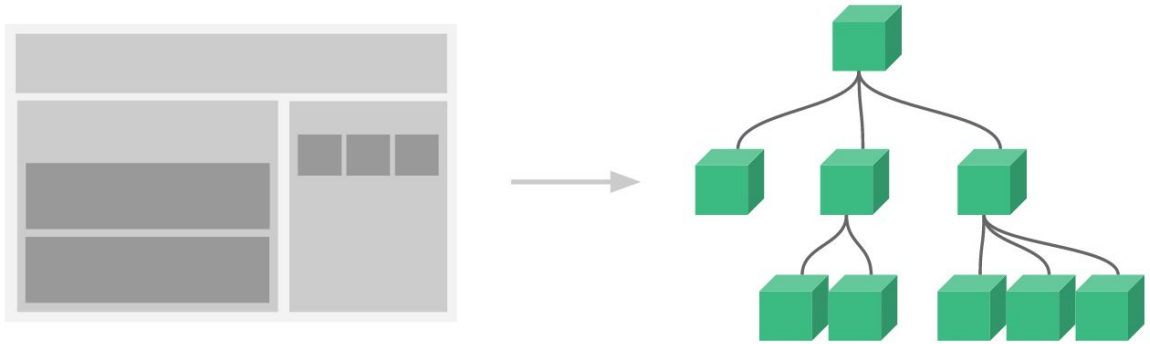
Que es un componente en Vue JS?

Pequeñas piezas de interfaz, como botones o formularios, que puedes reutilizar en tu web.

Independientes: Cada componente incluye todo lo que necesita para funcionar: partes visuales, código y estilos.

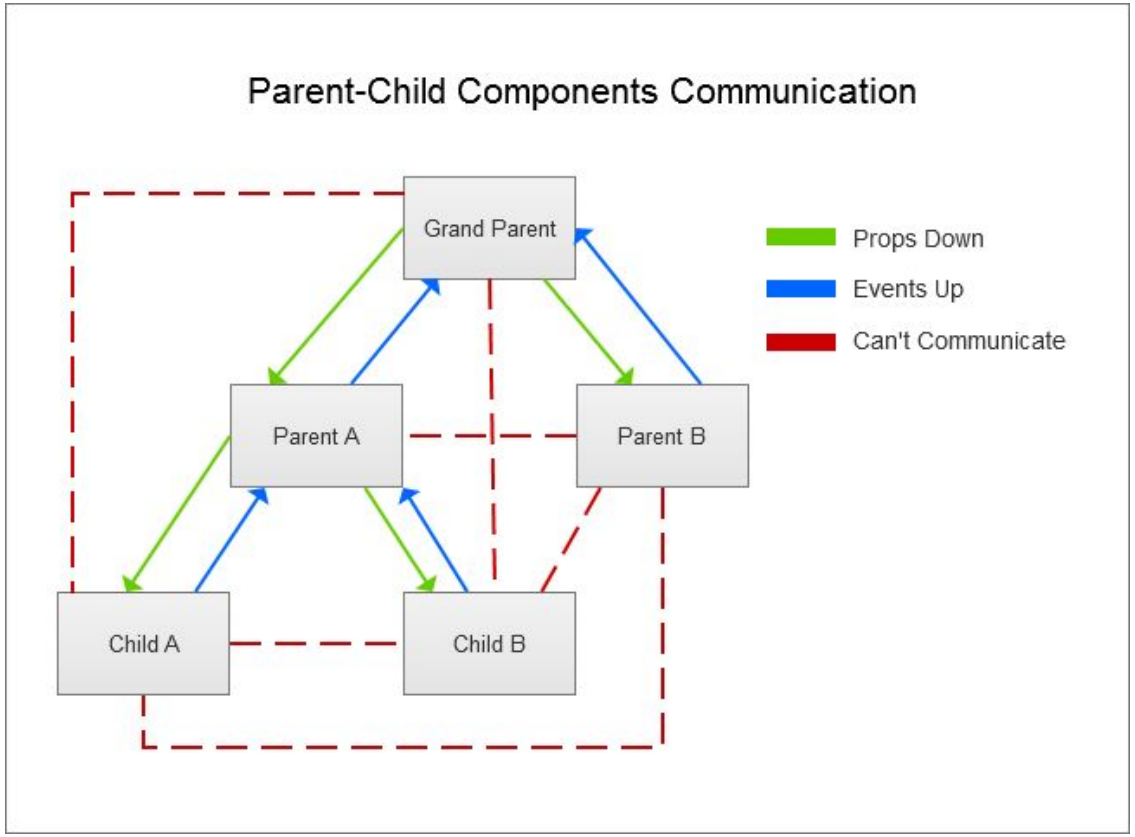
Reactivos: Se actualizan automáticamente cuando la información que muestran cambia.

Comunicativos: Los componentes pueden recibir información y enviar mensajes, facilitando la organización de la app.



Comunicación entre componentes Vue Js

La comunicación entre componentes en Vue.js es fundamental para construir aplicaciones complejas y escalables. Esta comunicación se maneja principalmente de dos maneras: de padres a hijos mediante props y de hijos a padres mediante eventos. Además, para la comunicación entre componentes que no tienen una relación directa de padre-hijo



Estructura de un componente en Vue Js

Template: La sección de template define la estructura HTML del componente. Aquí es donde especificas cómo debe mostrarse la interfaz de usuario del componente.

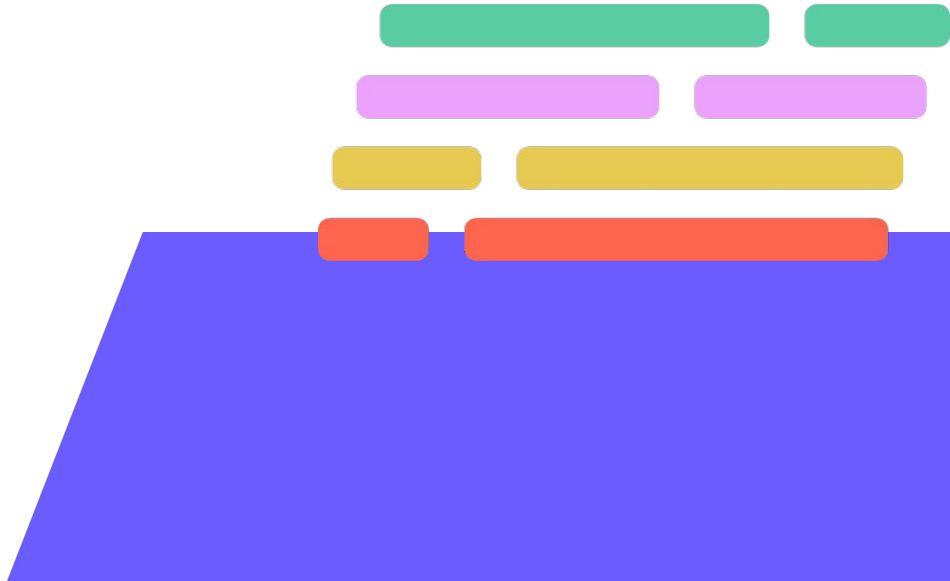
Script: La sección de script es donde defines la lógica del componente, como las propiedades de datos, métodos, propiedades computadas, observadores, y hooks de ciclo de vida.

Style: La sección style permite definir los estilos CSS específicos para este componente. Los estilos aquí definidos están encapsulados, lo que significa que solo afectan al componente actual

Directivas en Vue JS

Una directiva es una instrucción especial que se coloca en el marcado HTML para indicar a Vue.js que realice alguna acción en un elemento del DOM.

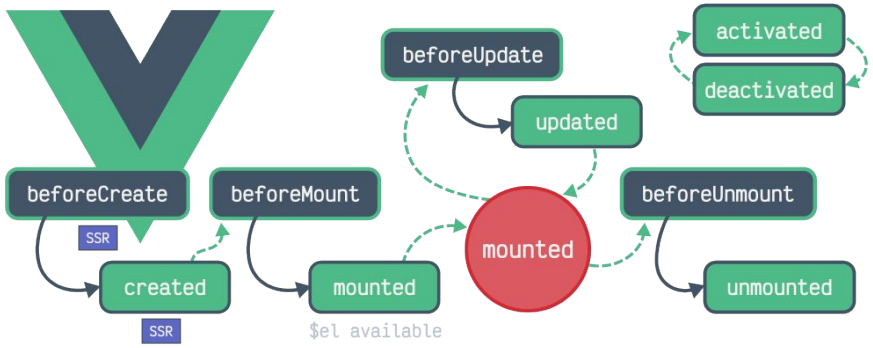
Directiva	Descripción
v-bind	Vincula dinámicamente uno o más atributos o una propiedad de componente a una expresión.
v-model	Crea una vinculación bidireccional en formularios HTML.
v-for	Renderiza una lista de elementos basada en un arreglo de datos.
v-if	Muestra un elemento solo si la expresión evaluada es verdadera.
v-else-if	Como v-if, pero se evalúa como una condición de 'else if'.
v-else	Usado para el bloque 'else' en condicionales v-if.
v-show	Alterna la visibilidad de un elemento cambiando la propiedad CSS 'display'.
v-pre	Salta la compilación de Vue para este elemento y todos sus hijos.
v-once	Renderiza el elemento y sus hijos una sola vez, ignorando futuras actualizaciones de datos.



Ciclo de vida de un componente

El ciclo de vida de un componente en Vue.js describe las distintas etapas por las que pasa desde su creación hasta su destrucción. Cada etapa del ciclo de vida es acompañada por un conjunto de "hooks", que son métodos que puedes definir en un componente para ejecutar código en puntos específicos durante la vida del componente.

Hook	Descripción
beforeCreate	Se ejecuta justo después de que la instancia ha sido inicializada, antes de la observación de datos y la inicialización de eventos.
created	Se llama después de que la instancia se ha creado completamente, con los observadores de datos, propiedades computadas, y eventos configurados.
beforeMount	Se activa justo antes de que el componente sea montado en el DOM.
mounted	Se ejecuta después de que el componente ha sido montado en el DOM, ideal para operaciones que requieren que el componente ya esté visible..
beforeUpdate	Se dispara después de que los datos cambian y el DOM virtual se re-renderiza pero antes de actualizar el DOM real.
updated	Se llama después de que los cambios en los datos se han reflejado en el DOM actualizado.
beforeDestroy	Se invoca justo antes de que la instancia de Vue sea destruida, mientras la instancia todavía es completamente funcional.
destroyed	Se llama después de que la instancia ha sido destruida, desvinculando directivas, removiendo event listeners y destruyendo subcomponentes.



Preprocesador de estilos

Vue.js se usa comúnmente con preprocesadores de estilos por varias razones estratégicas que mejoran el flujo de trabajo del desarrollo, la mantenibilidad del código y la escalabilidad de las aplicaciones.

Un preprocesador de estilos es una herramienta que extiende la funcionalidad de CSS con características adicionales como variables, funciones, mixins, y anidación de reglas, que no están disponibles en el CSS puro.

- Mejora de la Organización del Código
- Funcionalidades Avanzadas
- Mejora del Mantenimiento
- Productividad y Eficiencia
- Compatibilidad con Navegadores
- Componentes Estilizados de Forma Escalable

</Be a coder>

Susana Giraldo | Directora Comercial | 314 245 56 59