**Introducción al desarrollo de SPAs con React y Vue**

Nombre: Andrea Garcés

Curso: GR2SW

**Introducción**

Las Aplicaciones de Una Sola Página (SPA) han revolucionado el panorama del desarrollo web, brindando una experiencia de usuario fluida y altamente interactiva. En este trabajo de investigación, se analizarán dos de las bibliotecas de JavaScript más populares y poderosas para el desarrollo de SPAs: React y Vue.

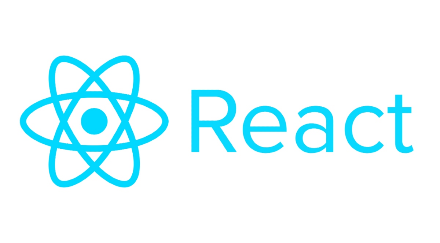
El objetivo de este informe es proporcionar una comprensión sólida de los conceptos básicos, las características y la arquitectura de React y Vue. Además, se presentará un ejemplo concreto de una aplicación "Hola Mundo" en ambas bibliotecas para ilustrar sus capacidades y diferencias.

**Conceptos básicos**

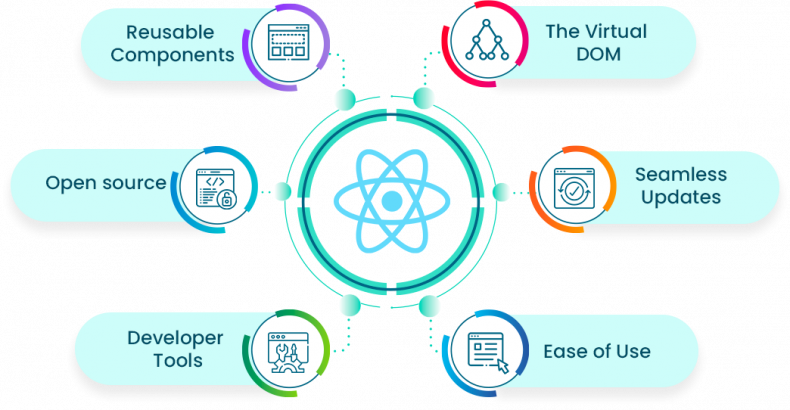
* ***¿Qué es una SPA?***

Una web SPA o single page application se refiere a una forma de desarrollo web en la que la página web está contenida en un único archivo. De esta forma, se carga todo en HTML y, mientras naveguemos por la página, irá solicitando el contenido al servidor vía ajax. Es decir, todo el contenido html, css y javascript de la página web al completo se carga una primera vez y nada más. De esta forma se mejoran los tiempos de respuesta y, por consiguiente, la experiencia de usuario.[1]

* ***React***



React es una biblioteca de JavaScript desarrollada por Facebook para construir interfaces de usuario (UI) interactivas y reactivas en aplicaciones web. Se utiliza para crear componentes de interfaz de usuario reutilizables y gestionar la actualización eficiente de la interfaz en función de los cambios en los datos.



1. Componentes: En React, una aplicación se construye mediante la creación de componentes. Los componentes son bloques de construcción reutilizables que encapsulan partes específicas de la interfaz de usuario. Pueden ser desde elementos simples, como botones o encabezados, hasta componentes más complejos, como formularios o tarjetas.
2. Estado (State): El estado es un concepto fundamental en React. Representa los datos que pueden cambiar en un componente a lo largo del tiempo. Cuando el estado de un componente cambia, React automáticamente se encarga de actualizar la interfaz de usuario para reflejar esos cambios.
3. Propiedades (Props): Las propiedades son datos que se pasan de un componente padre a un componente hijo en React. Las props permiten que los componentes compartan información y se vuelvan reutilizables. Los componentes pueden recibir props y utilizarlos para personalizar su comportamiento y contenido.
4. Manejo de eventos: React permite la gestión de eventos, como clics de botones o cambios de entrada, de manera similar a JavaScript tradicional. Los eventos pueden desencadenar cambios en el estado y, por lo tanto, en la interfaz de usuario.
5. Código abierto: React es una biblioteca de JavaScript de código abierto, lo que significa que su código fuente está disponible públicamente y es accesible para cualquier persona que desee utilizarlo, modificarlo o contribuir a su desarrollo.[2]

* ***Vue***



Vue.js, comúnmente conocido como Vue, es un framework de JavaScript progresivo utilizado para crear interfaces de usuario interactivas en aplicaciones web. Vue se enfoca en la capa de vista de una aplicación y proporciona un conjunto de herramientas para facilitar el desarrollo frontend.



1. Directivas: Vue utiliza directivas para agregar funcionalidad declarativa a los elementos HTML. Las directivas más comunes son ‘v-bind’, ‘v-model’, ‘v-if’, ‘v-for’, y ‘v-on’. Estas directivas permiten enlazar datos, controlar estructuras condicionales y realizar iteraciones de manera sencilla.
2. Reactividad: Vue utiliza un sistema de reactividad que detecta automáticamente los cambios en los datos y actualiza la interfaz de usuario de manera eficiente. Esto significa que no es necesario realizar manualmente las actualizaciones del DOM.
3. Interpolación: Vue permite la interpolación de datos directamente en las plantillas HTML utilizando la sintaxis de doble llave “{{}}”. Esto facilita la representación de datos dinámicos en la interfaz de usuario.
4. Directiva ‘v-model’: Esta directiva se utiliza para establecer una conexión bidireccional entre un elemento de formulario y una variable en el estado de Vue. Esto facilita la manipulación de datos de entrada y salida.
5. Ligereza y flexibilidad: Con un tamaño reducido y un enfoque progresivo, Vue permite a los desarrolladores integrar sus características de manera gradual en aplicaciones existentes o construir aplicaciones desde cero de manera eficiente. Vue es lo suficientemente versátil como para adaptarse a las necesidades específicas de cada proyecto.[3]

**Arquitectura**

* ***React***

La arquitectura de una aplicación construida con React se basa en el concepto de componentes. Cada componente es una unidad independiente que encapsula la lógica y la interfaz de usuario relacionada. Aquí se destacan algunos aspectos clave de la arquitectura de React:

1. Componentes: React fomenta la división de la interfaz de usuario en componentes reutilizables. Cada componente tiene su propio estado y propiedades.
2. Flujo de Datos Unidireccional: React sigue un patrón de flujo de datos unidireccional, donde los datos fluyen desde los componentes superiores a los componentes secundarios. Esto facilita la depuración y el mantenimiento de la aplicación.
3. Virtual DOM: React utiliza un Virtual DOM para mejorar el rendimiento. En lugar de actualizar directamente el DOM del navegador cada vez que cambia el estado de un componente, React compara el Virtual DOM con el DOM real y realiza las actualizaciones mínimas necesarias, lo que hace que la aplicación sea más eficiente.
4. Estructura de Carpetas: En cuanto a la estructura de carpetas, no existe una única convención, pero es común organizar los componentes en directorios y separar el código relacionado con el estado y las acciones en otro conjunto de archivos.
5. Enrutamiento: Para construir aplicaciones de una sola página con múltiples vistas, React se suele combinar con React Router, una biblioteca que facilita la navegación y la gestión de rutas en una aplicación React.[4]
6. Flujo unidireccional de datos: React sigue un patrón de flujo de datos unidireccional, lo que significa que los datos fluyen en una sola dirección, desde el componente padre al componente hijo. Esto facilita la depuración y el mantenimiento de la aplicación.[2]

* ***Vue***

La arquitectura de una aplicación Vue se basa en un enfoque más progresivo y flexible. Vue permite que los desarrolladores adopten gradualmente sus características según sea necesario. Aquí se destacan los aspectos clave de la arquitectura de Vue:

1. Componentes: Al igual que React, Vue fomenta la componentización de la interfaz de usuario. Los componentes de Vue son unidades autónomas que pueden contener su propio estado, métodos y plantillas.
2. Reactividad: Vue se destaca por su sistema de reactividad. Cuando los datos cambian, las partes relevantes de la interfaz de usuario se actualizan automáticamente sin necesidad de manipular el DOM directamente.
3. Integración Progresiva: Vue es conocido por su capacidad de integración progresiva. Esto significa que puedes utilizar Vue en una parte específica de tu aplicación sin tener que reconstruir toda la aplicación. Esto lo hace ideal para proyectos existentes.
4. Estructura de Carpetas: Al igual que en React, la estructura de carpetas en una aplicación Vue puede variar según las preferencias del desarrollador, pero generalmente se organiza en torno a componentes y módulos.[3]
5. Enrutamiento: Para crear aplicaciones de una sola página con navegación, Vue se combina comúnmente con Vue Router. Esta biblioteca permite la gestión de rutas y la navegación entre vistas de manera declarativa.[5]

**Bibliografía**

[1] Incentro. "¿Qué es una Web de Aplicaciones de Página Simple (SPA)?" [En línea]. Disponible en: <https://www.incentro.com/es-ES/blog/que-es-web-simple-page-applications>. [Acceso: 04/09/2023].

[2] React. "React – A JavaScript library for building user interfaces," [En línea]. Disponible en: <https://react.dev/learn>. [Acceso: 04/09/2023].

[3] Vue.js. "Vue.js - Introduction," [En línea]. Disponible en: <https://vuejs.org/guide/introduction.html>. [Acceso: 05/09/2023].

[4] React Router. "React Router: Declarative Routing for React" [En línea]. Disponible en: <https://reactrouter.com/en/main>. [Acceso: 04/09/2023].

[5] Vue Router. "Vue Router," [En línea]. Disponible en: <https://router.vuejs.org/>. [Acceso: 05/09/2023].