

El motivo del presente informe es realizar un análisis de REACT JS y su impacto al implementarlo en proyecto sitio web del Hospital.

## **1. ReactJS y su Aplicación en el Proyecto del Hospital**

React es una librería JS de naturaleza declarativa, su enfoque basado en componentes es ideal para construir una SPA que maneje múltiples secciones. En el proyecto del hospital nos permite ahorrar código HTML y JS, realizar cambios reactivos en tiempo real, controlar los estados de la aplicación, carga dinámica de datos, modularizar el sitio mediante componentes, entre otros.

## **2. Ventajas de las SPA en un Sistema de Hospital**

Desarrollar un SPA es ideal debido a que nos permite una navegación fluida, reactividad al renderizar los cambios del DOM en la aplicación, carga de datos dinámica (entre otros), lo cual permite una mejor experiencia de usuario.

## **3. Manejo del DOM Virtual en la Web del Hospital**

El DOM virtual es una característica clave de React que mejora el rendimiento y la eficiencia de las actualizaciones de la interfaz de usuario. Con esto, React puede realizar actualizaciones por lotes, minimizar los reflujos y los repintados y aplicar los cambios de manera eficiente. En nuestro caso, elementos estáticos como header y footer al ser modularizados mejora el rendimiento de la página, elementos dinámicos como carga de datos, carga de médicos, carga de especialidades son controlados por el uso de estados, lo cual no bloquea el flujo de la aplicación mejorando el rendimiento y experiencia de usuario.

## **4. Comparación de React JS con Otros Frameworks**

React es una buena opción para una gran variedad de proyectos debido a su enfoque declarativo, comunidad activa y su extenso ecosistema. A diferencia de Angular y Vue.js, React es conocido por su sencillez y adaptabilidad. Mientras que Angular tiene un enfoque más estructurado y basado en opiniones, React sigue una arquitectura basada en componentes, lo que permite a los desarrolladores crear componentes de interfaz de usuario reutilizables. Por otro lado, Vue.js encuentra un punto intermedio entre las dos opciones al proporcionar un proceso de integración paso a paso. Por lo tanto, para elegir React como librería se debe tener en cuenta la escala del proyecto, el nivel de complejidad, el grado deseado de supervisión y la sostenibilidad del proyecto a largo plazo.

## 5. Características Clave de ReactJS para el Desarrollo del Hospital

**Componentes:** React organiza la interfaz de usuario en componentes reutilizables, que pueden ser de distinto tipo y tamaño. De esta forma, se facilita la modularidad del código, simplificando la construcción y el mantenimiento de las aplicaciones, ya que cada componente se puede desarrollar y probar de forma aislada antes de integrarlo en el desarrollo final. En nuestro caso haremos uso de componentes como Footer, Header, CardDoctors, CardAppoinmets, DoctorList, AppoinmentsList, etc.

**Estados y Desarrollo Declarativo:** React permite al desarrollador describir cómo debe verse la interfaz dependiendo del estado de la aplicación. De este modo, cuando un componente cambia de estado, React hace un seguimiento de esos cambios y actualiza la interfaz de usuario para que aparezca acorde al nuevo estado. Esto, además de simplificar la gestión del estado, implica que la interfaz siempre estará sincronizada y actualizada con los datos de la aplicación.

**Virtual DOM:** React aplica las modificaciones primero en una representación virtual del DOM, de forma que posteriormente solo se apliquen al DOM real los cambios necesarios. Así, se reduce la carga en el navegador y se optimiza el rendimiento.

**JSX:** React utiliza JSX (JavaScript XML), que permite escribir código que se asemeja al marcado HTML. Gracias a ello, la creación de componentes se simplifica y mejora su legibilidad. JSX será aplicado en los componentes del hospital los cuales se estructurarán en carpetas haciendo una mejor legibilidad y mantención de código fuente.

**Unidireccionalidad de datos:** esto significa que los datos solo fluyen en una dirección a través de la aplicación. Esto facilita el seguimiento y el mantenimiento, con lo que se consigue un código más predecible y fácil de depurar.

**Herramientas complementarias:** React cuenta con numerosas opciones de herramientas y bibliotecas, como Redux, que facilitan y agilizan el desarrollo de plataformas en React.

## 6. Configuración y Ejecución de ReactJS en el Proyecto del Hospital

6.1. El proyecto se creará con el empaquetador VITE (requiere previa instalación de Node JS).

6.2. La implementación y comandos de Node JS para la creación, configuración y ejecución del sitio es (posteriormente se utilizará React + TypeScript):

```
npm create vite@latest  
react + javascript / proyecto-react-hospital  
cd proyecto-react-hospital  
npm install  
npm run dev  
npm run build
```

6.3. Se utilizará CDN para incorporar librería de iconos y Bootstrap (posteriormente serán incluidas mediante NPM).

6.4. Para manejar el estado de la aplicación entre páginas (Home, Personas, Citas, Contacto) utilizaremos Redux.

6.5. La estructura preliminar del sitio sería la siguiente:

/proyecto-react-hospital

```
├── README.md
├── node_modules
├── package.json
├── package-lock.json
├── .gitignore
├── public
│   ├── favicon.ico
│   ├── index.html
│   └── manifest.json
└── src
    ├── App.css
    ├── App.js
    ├── App.test.js
    ├── index.css
    ├── index.js
    └── components/
        ├── DoctorsCard.jsx
        ├── DoctorsList.jsx
        ├── DoctorsData.jsx
        ├── Appointments.jsx
        ├── ContactForm.jsx
        ├── Header.jsx
        ├── Footer.jsx
        └── MainContent.jsx
```