

## **Discusión y Análisis de Casos: Aplicación de ReactJS en el Proyecto del Hospital**

### **Integrantes:**

Pedro Guerrero  
Edwin Maureira  
Bastián Ortega

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

*Explica qué es ReactJS y justifica su uso en el proyecto del hospital.*

*Argumenta por*

*qué su naturaleza de librería declarativa y su enfoque basado en componentes es ideal*

*para construir una SPA que maneje múltiples secciones (Home, Servicios, Equipo*

*Médico, Contacto) de manera eficiente y dinámica.*

ReactJS es una librería de JavaScript desarrollada por Facebook en 2013, con el objetivo de simplificar el desarrollo de interfaces de usuario. Su creación responde a la necesidad de manejar grandes cantidades de datos dinámicos en interfaces interactivas. React está diseñado para ser modular, eficiente y flexible, permitiendo a los desarrolladores crear componentes reutilizables que gestionan su propio estado

Que React sea declarativo quiere decir que nosotros le decimos qué deseamos lograr (se declara el resultado) y React se encarga de renderizar y gestionar la implementación de la manera más eficiente posible. Esto en la página de hospital daría mayor simplicidad, eficiencia y claridad al código.

// Ejemplo Declarativo  
const element = <h1>Hello, world</h1>

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

*Analiza por qué utilizar una SPA desarrollada con ReactJS sería beneficioso para el sistema del hospital. Discute la necesidad de una navegación fluida, carga rápida de datos, y la optimización de la experiencia de usuario para acceder a información médica, servicios, doctores, etc., sin recargar la página.*

Crear la página del hospital como una SPA (Single Page Application) permite desarrollar una aplicación más fluida y rápida en comparación con las aplicaciones tradicionales basadas en múltiples páginas, ya que evita la recarga completa de la página cada vez que el usuario interactúa.

Trabajar con una app de “hospital” en React ofrece otros beneficios, como la facilidad para integrar componentes y APIs relacionados con pacientes, doctores y citas. También mejora el rendimiento en dispositivos móviles y facilita la incorporación de nuevas funcionalidades de manera eficiente. Además al tener todo el código en una sola página, el mantenimiento y las actualizaciones de estos se vuelven más sencillos.

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

*Explica cómo el DOM virtual de React puede mejorar el rendimiento de la aplicación web del hospital. Discute cómo la actualización eficiente de los componentes, sin recargar toda la página, puede hacer que el sistema funcione de manera más rápida y eficiente cuando los usuarios navegan entre las distintas secciones (consultas, citas, servicios).*

En React, el contenido se renderiza utilizando el DOM Virtual, una representación en memoria del DOM real que permite que React haga cambios de forma más eficiente.

Un DOM virtual permitirá en la página de hospital mejorar y optimizar la carga de pacientes, doctores, citas y otros elementos dinámicos de la página, Un ejemplo de una interacción puede ser si un usuario de la página decide hacer un cambio en un elemento dinámico como una cita, solo se renderiza ese elemento, sin recargar otros elementos de la página.

## Comparación de ReactJS con Otros Frameworks para el Proyecto

*Compara ReactJS con otras tecnologías como Angular o VueJS, y analiza por qué*

*React sería más adecuado para la construcción del sistema del hospital.*

*Considera*

*aspectos como la facilidad de integración con otras herramientas (por ejemplo, bases de*

*datos, API REST para manejo de citas), el manejo del ciclo de vida de los componentes*

*y la modularización.*

Característica	ReactJS	Angular	Vue
Renderizado	Virtual DOM	Real DOM	Virtual DOM
Aprendizaje	Fácil de iniciar	Complejo	Intermedio
Flujo de datos	Unidireccional	Bidireccional	Bidireccional
Github Stars	230k	96.4k	208k
Framework o librería?	Librería	Framework	Framework

Para la aplicación de hospital React tiene algunas ventajas frente a Angular o VueJS:

- React tiene una comunidad y ecosistema más grande que otras tecnologías
- React y Vue son tecnologías utilizadas a la hora de hacer SPA por su flexibilidad
- Al ser una librería da mayor flexibilidad al usar componentes y APIs
- El **Virtual DOM** es una técnica utilizada por bibliotecas como **React** para mejorar la eficiencia al manipular la UI.

## **Características Clave de ReactJS para el Desarrollo del Hospital**

*Identifica las características clave de ReactJS que facilitarían la implementación del proyecto del hospital:*

- *Componentización: Cómo dividir la interfaz del hospital en componentes reutilizables.*
- *Flujo de datos unidireccional: Cómo el control de flujo de información (por ejemplo, desde el backend al frontend) mejora la consistencia del sistema.*
- *JSX: Cómo facilita la creación de interfaces dinámicas y personalizables, mostrando datos médicos, doctores y citas.*

En la página de hospital aprovecharemos las diferentes ventajas de React como son la componentización, el flujo de datos unidireccional y el uso de JSX. Dividiremos la página en diferentes componentes de fragmentos de páginas, herramientas en JSX y llamados de APIs, pensando en la reutilización de estos. Aprovecharemos el flujo unidireccional de los padres a los hijos para herramientas de la página como lista de doctores, citas y servicios, y el envío de estos a componentes hijos al querer mostrarlos en la página.

El uso de JSX mejorará la simplicidad, legibilidad y optimización en el código JS y HTML.

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

*Explica cómo configurar el entorno de ReactJS para el desarrollo del sistema del hospital. Considera los prerequisites, la configuración local o el uso de CDN, y la utilización de herramientas como Webpack, Babel y Redux para manejar la lógica de negocio del hospital (citas, doctores, horarios).*

Para proyectos más grandes como el de la página de Hospital, se recomienda instalar React en el entorno local utilizando npm o yarn. Esto permite el uso de herramientas como Webpack y Babel y Redux.

En este caso para empaquetar módulos usaremos Vite porque es más rápido de instalar y usa ES Modules nativos.

Estos son los pasos para instalar un proyecto de react con Vite

1. Primero instalamos Vite y React

```
// npm create vite@latest nombre-del-proyecto --template react
```

2. luego las dependencias del proyecto

```
// npm install
```

3. y luego iniciamos el servidor

```
// npm run dev
```

4. Vite trae por defecto Babel y otra transformación de códigos llamado esbuild

5. Para configurar Redux utilizaremos el siguiente código en la terminal

```
// npm install redux react-redux
```

Después, configuras el store de Redux, lo proporcionas a la aplicación mediante el Provider, y usas el estado global en tus componentes con useSelector y useDispatch.