

## Rúbrica de Evaluación: Manejo Avanzado del DOM y Elementos ReactJS para el Proyecto del Hospital

Criterio de Evaluación	TL (1.5)	L (1)	ML (0.5)	NL (0)	Puntaje Máximo
<b>Manejo del DOM Virtual en ReactJS</b>	Implementa correctamente el DOM Virtual para optimizar el rendimiento en la renderización de datos (por ejemplo, doctores o servicios) y gestiona la actualización del DOM al cargar datos con <code>useEffect</code> o <code>componentDidMount</code> .	El DOM Virtual está implementado, pero con errores menores en la optimización o en la actualización de los datos.	La implementación del DOM Virtual presenta fallos que afectan el rendimiento del sistema.	No se utiliza el DOM Virtual o no se gestiona correctamente la actualización del DOM en ReactJS.	1
<b>Creación y Uso de Referencias en React</b>	Las <b>referencias</b> son implementadas correctamente para manipular elementos del DOM, como	Se utilizan referencias, pero con errores menores en su implementación o en el acceso a los elementos del DOM.	Las referencias están mal implementadas o no permiten manipular correctamente los elementos del DOM.	No se utilizan referencias o no se implementan correctamente en los componentes.	1.5

Criterio de Evaluación	TL (1.5)	L (1)	ML (0.5)	NL (0)	Puntaje Máximo
	campos de formularios o elementos de la interfaz del hospital, y se utilizan referencias de callback cuando es necesario.				
<b>Uso de Fragmentos y Contexto en ReactJS</b>	Se implementan <b>Fragmentos</b> para mejorar la estructura del DOM sin añadir nodos innecesarios. El <b>Contexto</b> es utilizado para gestionar datos globales, como el acceso a información de usuarios o doctores, sin necesidad de pasar props manualmente.	Se utilizan Fragmentos y Contexto, pero con errores menores en su implementación o en la gestión del estado global.	El uso de Fragmentos o Contexto está mal implementado o no facilita la gestión del estado global del proyecto.	No se utilizan Fragmentos o Contexto correctamente, afectando la estructura y gestión del estado en ReactJS.	1.5
<b>Verificación de Tipos con PropTypes</b>	Se implementan correctamente e las	Las PropTypes están presentes,	Las PropTypes están mal implementadas	No se implementan PropTypes para la	1

Criterio de Evaluación	TL (1.5)	L (1)	ML (0.5)	NL (0)	Puntaje Máximo
	<b>PropTypes</b> para verificar los tipos de datos pasados a los componentes , mostrando advertencias en caso de que el tipo de dato no sea el correcto.	pero no validan correctamente e todos los tipos de datos o presentan errores menores.	as o no se validan correctamente e los tipos de datos pasados a los componentes .	verificación de tipos en los componentes .	
<b>Uso de Componentes de Orden Superior y Portales</b>	Se implementa correctamente e un <b>Componente de Orden Superior (HOC)</b> para reutilizar la lógica de un componente y se utiliza un <b>portal</b> para renderizar un modal u otra interfaz sin interferir con la estructura del DOM principal.	Se implementa el HOC o el portal, pero con errores menores en la reutilización de lógica o en la gestión del DOM.	El HOC o el portal están mal implementados o no funcionan como se espera, afectando la estructura o la lógica de los componentes .	No se implementa un HOC o portal como se requiere para la gestión de la interfaz y reutilización de lógica.	1.5
<b>Optimización de Rendimiento</b>	Utiliza <b>Profiler</b> para identificar cuellos de	Se utiliza Profiler, pero con errores menores en	El uso de Profiler es limitado o no optimiza	No se utiliza Profiler o no se realiza ninguna	0.5

Criterio de Evaluación	TL (1.5)	L (1)	ML (0.5)	NL (0)	Puntaje Máximo
y Profiler en ReactJS	botella en la renderización de la aplicación del hospital y optimiza el rendimiento en componentes que cargan grandes volúmenes de datos.	la identificación de problemas de rendimiento o la optimización de la aplicación.	correctament e el rendimiento del sistema.	optimización del rendimiento en el sistema ReactJS.	

## Interpretación de los Resultados

- **Totalmente logrado (TL):** Los estudiantes han implementado correctamente todos los elementos avanzados de ReactJS, incluyendo el manejo del **DOM virtual**, referencias, **PropTypes**, y han optimizado el rendimiento mediante **Profiler**.
- **Logrado (L):** Los estudiantes han cubierto la mayoría de los requisitos, pero con algunos errores menores que afectan la optimización o la estructura del DOM.
- **Medianamente logrado (ML):** Algunas partes del proyecto están mal implementadas o incompletas, afectando el rendimiento y la gestión del DOM en el proyecto.
- **No logrado (NL):** No se cumplen los requisitos básicos del ejercicio, y las implementaciones avanzadas de ReactJS presentan errores importantes o están ausentes.

**Puntaje total: 7 puntos.**