

## Rúbrica de Evaluación: Integración Completa del Sistema del Hospital con ReactJS

Criterio de Evaluación	TL (1.5)	L (1)	ML (0.5)	NL (0)	Puntaje Máximo
<b>Implementación de Vistas Complejas con ReactJS</b>	Las vistas <b>Home, Equipo Médico, y Citas</b> están correctamente implementadas y estructuradas con componentes reutilizables, validaciones, y un diseño claro.	Las vistas están implementadas, pero presentan errores menores o falta de validaciones en una de las secciones.	Las vistas están incompletas o presentan errores importantes en la estructura o funcionalidad de los componentes .	No se implementan correctamente las vistas o faltan componentes clave.	1.5
<b>Optimización del DOM Virtual y Uso de Fragmentos</b>	Se utiliza el <b>DOM Virtual</b> y <b>Fragmentos</b> de manera eficiente, actualizando solo los elementos necesarios en cada vista y evitando la creación de nodos innecesarios.	Se usan el DOM Virtual y Fragmentos, pero hay pequeños errores que afectan la estructura o eficiencia.	El DOM Virtual o los Fragmentos están mal implementados, afectando el rendimiento de la aplicación.	No se implementan correctamente el DOM Virtual o los Fragmentos.	1

Criterio de Evaluación	TL (1.5)	L (1)	ML (0.5)	NL (0)	Puntaje Máximo
<b>Uso de Referencias y Callbacks</b>	Las referencias se utilizan correctamente para interactuar con el DOM, como enfocar campos o gestionar desplazamientos en las vistas, y las referencias de callback están bien implementadas.	Se utilizan referencias y callbacks, pero con errores menores en su implementación o en su uso en el DOM.	Las referencias o callbacks están mal implementadas, limitando la interacción con los elementos del DOM.	No se implementan referencias ni callbacks para la gestión del DOM.	1.5
<b>Manejo de Datos con API REST Simulada</b>	Los datos del equipo médico y servicios se obtienen correctamente mediante una <b>API REST simulada</b> utilizando <b>fetch</b> o <b>axios</b> con <b>async/await</b> , y se gestionan de manera asíncrona en las vistas	Se simula el manejo de datos, pero hay errores menores en la carga o en la renderización de los datos en las vistas.	El manejo de datos asíncronos es deficiente, afectando la renderización correcta de los componentes en las vistas.	No se implementa correctamente la obtención de datos a través de una API REST simulada.	1.5

Criterio de Evaluación	TL (1.5)	L (1)	ML (0.5)	NL (0)	Puntaje Máximo
	correspondientes.				
<b>Optimización de Rendimiento y Uso de Profiler</b>	Se usa <b>Profiler</b> para identificar y resolver problemas de rendimiento, optimizando la renderización de las vistas que manejan grandes volúmenes de datos (como el listado de doctores o servicios).	Se utiliza Profiler, pero con errores menores en la identificación de problemas o en la optimización del rendimiento.	La optimización de rendimiento es deficiente o no se identifica adecuadamente con Profiler.	No se utiliza Profiler ni se realizan optimizaciones de rendimiento en la aplicación.	1
<b>Comprobación de Tipos con PropTypes</b>	Las <b>PropTypes</b> están correctamente implementadas en todos los componentes, verificando los tipos de datos y mostrando advertencias en caso de errores.	Las PropTypes están implementadas, pero no validan todos los tipos de datos o presentan pequeños errores en su implementación.	Las PropTypes están mal implementadas o no validan correctamente los tipos de datos en los componentes.	No se implementan PropTypes para la validación de tipos de datos en los componentes.	0.5

## Interpretación de los Resultados

- **Totalmente logrado (TL):** Los estudiantes han implementado correctamente las vistas, gestionado datos asíncronos mediante API, optimizado el rendimiento y utilizado técnicas avanzadas de ReactJS como referencias, Profiler y PropTypes.
  - **Logrado (L):** La mayoría de los requisitos están cubiertos con errores menores que no afectan significativamente la funcionalidad de la aplicación.
  - **Medianamente logrado (ML):** Hay varias áreas mal implementadas o incompletas, afectando el rendimiento y la funcionalidad de las vistas.
  - **No logrado (NL):** Los componentes clave no están implementados o presentan errores graves que impiden el correcto funcionamiento del sistema.
- 

**Puntaje total: 7 puntos.**