

## Rúbrica de Evaluación: Consumo de API en el Sistema del Hospital

Criterio de Evaluación	TL (1.5)	L (1)	ML (0.5)	NL (0)	Puntaje Máximo
<b>Implementación de Peticiones con useEffect y useState</b>	Se utilizan correctamente los Hooks <b>useEffect</b> y <b>useState</b> para realizar peticiones a la API y mostrar los datos de manera dinámica en la interfaz del hospital.	Se utilizan los Hooks, pero con errores menores en la gestión del estado o en la actualización de los datos.	Los Hooks están mal implementados, lo que impide que los datos se muestren correctamente.	No se implementan correctamente los Hooks, ni se realiza la petición a la API.	2
<b>Uso de Fetch API o Axios para el Consumo de la API</b>	Se implementa correctamente <b>Fetch API</b> o <b>Axios</b> para realizar las peticiones, manejando asincrónicamente los datos y mostrando los resultados en la interfaz.	Se utiliza <b>Fetch API</b> o <b>Axios</b> , pero con pequeños errores en la petición o en el manejo de los datos.	Las peticiones no se realizan correctamente, afectando la visualización de los datos.	No se utiliza correctamente <b>Fetch API</b> o <b>Axios</b> para realizar las peticiones.	1.5
<b>Peticiones Basadas en Eventos del Usuario</b>	Se permite al usuario realizar peticiones a	La interacción con el usuario para	La interacción con el usuario está	No se implementa la funcionalidad	1

Criterio de Evaluación	TL (1.5)	L (1)	ML (0.5)	NL (0)	Puntaje Máximo
	la API mediante una interacción, como un botón para recargar los datos, y los datos se actualizan correctamente en la interfaz.	realizar peticiones funciona, pero con errores menores en la actualización de los datos.	mal implementada o no actualiza correctamente los datos en la interfaz.	para realizar peticiones a partir de eventos del usuario.	
<b>Manejo de Errores en Peticiones Asíncronas</b>	Se implementa una estrategia de manejo de errores que muestra mensajes en la interfaz cuando la API falla, y se permite al usuario intentar nuevamente realizar la petición.	El manejo de errores funciona, pero con errores menores en la visualización o en la reintentar la petición.	El manejo de errores está mal implementado o no permite al usuario manejar la falla de la API.	No se implementa ninguna estrategia de manejo de errores en las peticiones a la API.	1
<b>Optimización del Rendimiento al Omitir Efectos en useEffect</b>	Se optimiza correctamente el uso de <b>useEffect</b> , evitando que las peticiones se realicen múltiples	La optimización está presente, pero hay pequeños errores que provocan	La optimización es deficiente y afecta el rendimiento de la aplicación.	No se implementa ninguna optimización en <b>useEffect</b> , provocando	0.5

Criterio de Evaluación	TL (1.5)	L (1)	ML (0.5)	NL (0)	Puntaje Máximo
	veces innecesariamente, limitando las peticiones solo a cambios relevantes.	recargas innecesarias.		recargas innecesarias.	

## Interpretación de los Resultados

- **Totalmente logrado (TL):** Los estudiantes han implementado correctamente el consumo de API utilizando **useEffect** y **useState**, optimizado el rendimiento, y manejado los errores de manera eficiente.
- **Logrado (L):** La mayoría de los requisitos están cubiertos con pequeños errores que no afectan significativamente el funcionamiento de la aplicación.
- **Medianamente logrado (ML):** Varias áreas del proyecto presentan errores que afectan el correcto funcionamiento de las peticiones y el manejo de los datos.
- **No logrado (NL):** No se cumplen los requisitos básicos, y las peticiones a la API o el manejo de errores no están implementados o son incorrectos.

**Puntaje total: 7 puntos.**