

## Rúbrica de Evaluación: Mejorando la Web del Hospital con APIs, TypeScript, y Seguridad

Criterio de Evaluación	TL (2)	L (1.5)	ML (1)	NL (0)	Puntaje Máximo
<b>Consumo de APIs usando Fetch API o Axios</b>	El consumo de la API está completamente implementado o con <b>Fetch API</b> o <b>Axios</b> . Se gestionan correctamente e las solicitudes <b>GET, POST, PUT, y DELETE</b> , y los datos se muestran dinámicamente en la interfaz React. Maneja errores de manera efectiva y muestra mensajes claros al usuario.	El consumo de la API está implementado, pero con errores menores en la gestión de solicitudes o en la visualización de datos. El manejo de errores está presente, pero con algunas limitaciones.	La implementación del consumo de API tiene errores importantes que afectan la interacción con los datos o el manejo de errores.	No se implementa correctamente e el consumo de la API, lo que impide el acceso o la visualización de los datos.	2
<b>Integración de TypeScript</b>	TypeScript está correctamente	TypeScript está integrado,	La integración de TypeScript	No se utiliza TypeScript de manera	2

Criterio de Evaluación	TL (2)	L (1.5)	ML (1)	NL (0)	Puntaje Máximo
<b>en Componentes Clave</b>	e integrado en los componentes clave, definiendo tipos, props, y estados de manera adecuada. Se utilizan interfaces y clases para modelar los datos, lo que mejora la estructura y la robustez del código.	pero con errores menores en la definición de tipos o props. Las interfaces o clases están parcialmente implementadas.	es incompleta o presenta errores significativos que afectan la estructura o el funcionamiento del código.	efectiva, o no se integra en los componentes .	
<b>Mejoras en la Seguridad del Front-End</b>	Se implementan correctamente medidas de seguridad como la protección de rutas con <b>React Router DOM</b> , el uso de <b>JWT</b> para asegurar peticiones a la API, y la validación de formularios para evitar ataques XSS. Se asegura	La seguridad está implementada, pero con pequeños errores en la protección de rutas, la autenticación, o la validación de formularios.	La seguridad está mal implementada o es incompleta, dejando la aplicación expuesta a vulnerabilidades.	No se implementan medidas de seguridad o la implementación es incorrecta.	2

Criterio de Evaluación	TL (2)	L (1.5)	ML (1)	NL (0)	Puntaje Máximo
	la encriptación de datos sensibles antes de enviarlos al servidor.				
<b>Optimización con Hooks y Manejo de Errores</b>	Se usan correctamente <b>useState</b> y <b>useEffect</b> para gestionar el estado y efectos secundarios. El <b>Hook personalizado</b> resuelve una funcionalidad recurrente, y el manejo de errores es efectivo, mostrando mensajes claros al usuario.	Los Hooks están implementados, pero con errores menores en la gestión del estado o efectos secundarios. El Hook personalizado funciona, pero con limitaciones.	La implementación de Hooks o el manejo de errores es incompleta o presenta errores importantes que afectan el funcionamiento de la aplicación.	No se implementan Hooks correctamente o no se maneja adecuadamente el control de errores.	1

## Interpretación de los Resultados

- **Totalmente logrado (TL):** Los estudiantes han implementado correctamente el consumo de **APIs**, la integración de **TypeScript**, las mejoras de **seguridad**, y la optimización mediante **Hooks**, logrando una aplicación funcional, segura y bien estructurada.

- **Logrado (L):** La mayoría de las funcionalidades están implementadas correctamente, con pequeños errores que no afectan gravemente la seguridad o el funcionamiento del proyecto.
- **Medianamente logrado (ML):** Varias áreas del proyecto presentan errores importantes o están incompletas, lo que afecta la funcionalidad o la seguridad de la aplicación.
- **No logrado (NL):** No se implementan correctamente las funcionalidades clave, y la aplicación no funciona o está expuesta a vulnerabilidades graves.

---

**Puntaje total: 7 puntos.**