

calidad de software

Evaluación 3



17 de diciembre de 2024

gabriel salinas

INSTALACIÓN DE LINTER

Para instalar un linter en un proyecto React, Procedo a ejecutar el código **NPM INSTALL STANDARD -D** en la terminal al igual que en el video de ejemplo, lo que agregó las dependencias necesarias de desarrollo. Posteriormente, creé un archivo de configuración. eslintrc.json donde definí las reglas de revisión de código, como extender configuraciones recomendadas de React. Para verificar su funcionamiento, agregué scripts en package.json como "lint": "eslint src/\*\*/\*.{js,jsx}" que me permiten ejecutar la revisión de código estático y detectar potenciales problemas de sintaxis o buenas prácticas.

Requerimiento Funcional

La idea central de mi aplicación era crear un sistema que permitiera mostrar tarjetas de los personajes de “Los Picapiedras” con información que puede cambiar. Para lograr esto, implementé un componente **“CARD”** completamente configurable que recibe props como el título, la descripción y la imagen. Esto me permite generar diferentes tarjetas con contenido único sin tener que cambiar la estructura base del componente. Este enfoque hace que el código sea más flexible y reutilizable, ya que puedo crear tarjetas para distintos personajes (como los Picapiedras) simplemente cambiando los valores de las props, manteniendo una presentación consistente y facilitando la expansión de la aplicación

|  |
| --- |
|  |

Requerimiento No Funcional

Como requerimiento no funcional, mi proyecto se centra en asegurar una experiencia fácil y fluida. Para lograr esto, optimicé el rendimiento utilizando técnicas como la importación estática de imágenes para reducir los tiempos de carga, aplicando estilos CSS minimizados y optimizados, y empleando **StrictMode** de React para detectar posibles problemas de renderizado.

|  |
| --- |
|  |

Además, diseñé componentes responsivos con CSS que se ajustan a diferentes tamaños de pantalla, garantizando una presentación consistente en cualquier dispositivo, lo que mejora tanto la usabilidad como la accesibilidad de la aplicación.

|  |
| --- |
|  |

Análisis OWASP

Desde el punto de vista de seguridad de OWASP, revisé mi proyecto para identificar posibles riesgos, especialmente en el manejo de imágenes y rutas. Para reducir estos riesgos, utilicé importaciones estáticas para las imágenes, lo que ayuda a evitar algún tipo de código malicioso. También protegí la información al crear una estructura de componentes que no muestra detalles innecesarios, y me aseguré de controlar bien las importaciones de recursos para evitar posibles fallos de seguridad. Todo esto siguiendo las recomendaciones de seguridad de OWASP para asegurar que la aplicación sea más mucho más segura.

Estándar de Calidad ISO/IEC 25010

Siguiendo el modelo de calidad de software ISO/IEC 25010, evalué mi proyecto en varias áreas clave. Verifiqué que los componentes cumpliera con los requisitos de renderizado, asegurando una funcionalidad adecuada. revisé la compatibilidad en diferentes plataformas, garantizando que los componentes de React se muestren de manera correcta en varios dispositivos.