

Rúbrica Analítica para Evaluación de Proyectos de Software - Ingeniería de Sistemas

Rúbrica Analítica | Ingeniería | Ingeniería de sistemas | 4 niveles

Descripción

Esta rúbrica está diseñada para evaluar proyectos de software desarrollados por estudiantes de educación técnica y tecnológica en Ingeniería de Sistemas. Cada criterio se evalúa de forma individual para identificar con precisión las fortalezas y áreas de mejora en el proyecto.

Rúbrica

Rúbrica Analítica para Evaluación de Proyectos de Software - Ingeniería de Sistemas

Esta rúbrica está diseñada para evaluar proyectos de software desarrollados por estudiantes de educación técnica y tecnológica en Ingeniería de Sistemas. Cada criterio se evalúa de forma individual para identificar con precisión las fortalezas y áreas de mejora en el proyecto.

Criterios de Evaluación	Excelente (4)	Bueno (3)	Aceptable (2)	Bajo (1)
1. Cumplimiento de Requisitos Funcionales	El software cumple completamente con todos los requisitos funcionales establecidos, sin errores.	El software cumple la mayoría de los requisitos funcionales, con mínimos errores o faltantes.	El software cumple algunos requisitos funcionales, pero con errores notables y funcionalidades incompletas.	El software no cumple con los requisitos funcionales o presenta fallas graves que impiden su uso.
2. Calidad del Código	Código bien estructurado, legible, documentado y sigue buenas prácticas de programación.	Código generalmente claro y estructurado, con documentación básica y buenas prácticas en su mayoría.	Código con estructura pobre, poca documentación y prácticas de programación inconsistentes.	Código desorganizado, difícil de entender, sin documentación ni adherencia a buenas prácticas.

Criterios de Evaluación	Excelente (4)	Bueno (3)	Aceptable (2)	Bajo (1)
3. Diseño y Arquitectura del Software	Diseño robusto y modular que facilita la escalabilidad y mantenimiento del sistema.	Diseño funcional y modular con algunas áreas que podrían mejorarse para escalabilidad.	Diseño poco modular, con dificultades para escalar o mantener el software.	Diseño inadecuado que afecta gravemente la funcionalidad y mantenimiento del software.
4. Pruebas y Validación	Realización exhaustiva de pruebas con resultados documentados y corrección de errores identificados.	Pruebas aplicadas adecuadamente con documentación parcial y corrección de errores principales.	Pruebas limitadas, poca documentación y errores importantes sin corregir.	No se realizaron pruebas o estas son insuficientes, sin corrección de errores.
5. Documentación Técnica	Documentación completa, clara y organizada que facilita la comprensión y uso del software.	Documentación adecuada, aunque con algunos detalles o secciones poco claras.	Documentación incompleta o poco clara, dificultando la comprensión del proyecto.	Documentación ausente o insuficiente para entender el software o su uso.
6. Gestión del Proyecto	Excelente planificación y control del proyecto, con cumplimiento puntual de objetivos y entregables.	Buena gestión del proyecto, con leves retrasos o ajustes en la planificación.	Gestión irregular, con retrasos y falta de seguimiento en algunos aspectos del proyecto.	Mala gestión, incumplimiento de objetivos y entregables sin control ni seguimiento.
7. Usabilidad y Experiencia de Usuario	Interfaz intuitiva, accesible y que ofrece una experiencia de usuario clara y satisfactoria.	Interfaz funcional con algunos detalles que podrían mejorar la experiencia del usuario.	Interfaz poco amigable o con dificultades para la navegación y uso del software.	Interfaz confusa o inoperable que afecta negativamente la experiencia del usuario.
8. Innovación y Creatividad	El proyecto presenta ideas innovadoras y soluciones creativas que aportan valor significativo.	El proyecto incluye algunos elementos creativos o mejoras innovadoras moderadas.	El proyecto tiene pocas ideas originales, siguiendo principalmente soluciones estándar.	El proyecto carece de creatividad o innovación, replicando soluciones básicas sin aportes propios.