Taller sobre validación de documentos

Heidy Dayana Otagrí Rodríguez

Tecnología en Análisis y Desarrollo de Software

Servicio Nacional de Aprendizaje (SENA)

Ficha 2627060

Ing. Yerman Augusto Hernández Sáenz

  
  
Bogotá D.C., marzo del 2023

1. **¿Qué es un artefacto en programación?**

Un artefacto se refiere a cualquier componente tangible o intangible que surge durante el proceso de desarrollo de software y contribuye al producto final. Estos elementos son representativos de las diferentes etapas del ciclo de vida del desarrollo, desde la concepción y el diseño hasta la implementación y el despliegue. Los artefactos encapsulan información esencial, como lógica de negocio, configuraciones técnicas, flujos de datos y algoritmos.

1. **Tipos de artefactos**

Los artefactos pueden clasificarse en diversas categorías, como:

* Código Fuente: Los archivos de código que contienen las instrucciones escritas por los programadores en lenguajes como Java, Python, C++, etc.
* Documentación Técnica: Manuales técnicos, comentarios en el código, guías de usuario y especificaciones de diseño.
* Archivos de Configuración: Archivos que definen cómo se comporta el software en diferentes entornos, como archivos YAML, JSON o XML.
* Archivos Binarios: Los ejecutables y las bibliotecas compartidas que permiten la ejecución del software.
* Casos de Prueba: Conjuntos de datos de prueba y scripts que verifican el comportamiento esperado del software.
* Diagramas de Arquitectura: Representaciones visuales de la estructura del sistema, como diagramas de clases, diagramas de flujo y diagramas de componentes.

1. **Evaluación de artefactos**

La evaluación de artefactos es un proceso crítico para garantizar la calidad, la eficiencia y la funcionalidad del software. Implica la revisión exhaustiva de los artefactos para identificar problemas y asegurarse de que se adhieran a los estándares y prácticas recomendadas. Esto incluye evaluar la legibilidad y la coherencia del código, verificar el rendimiento del software y garantizar que se cumplan los requisitos funcionales y no funcionales.

1. **¿Cómo se realizan las evaluaciones de artefactos?**

Las evaluaciones de artefactos son llevadas a cabo por un equipo de desarrollo o por revisores expertos. Estas evaluaciones pueden involucrar:

* Revisión por Pares: Los desarrolladores revisan y comentan el código y otros artefactos creados por sus colegas.
* Análisis Estático: El uso de herramientas automatizadas para examinar el código en busca de problemas potenciales, como vulnerabilidades de seguridad y malas prácticas.
* Pruebas Unitarias y de Integración: Ejecución de pruebas automatizadas para validar la funcionalidad y detectar fallos.
* Auditorías de Código: Evaluaciones exhaustivas realizadas por revisores con experiencia para garantizar la calidad y la coherencia del código.

1. **Instrumentos utilizados en la evaluación**

En la evaluación de artefactos, se emplean varias herramientas y enfoques técnicos, tales como:

* **Herramientas de Análisis Estático:** Ejemplos incluyen SonarQube, ESLint, Checkstyle, que analizan el código en busca de problemas de calidad y estilo.
* **Herramientas de Pruebas Automatizadas:** Como JUnit (Java) y pytest (Python) para realizar pruebas unitarias y de integración.
* **Herramientas de Revisión de Código**: GitHub, GitLab y Bitbucket proporcionan funcionalidades para revisiones colaborativas.
* **Herramientas de Automatización de Despliegue:** Como Jenkins y Travis CI, que aseguran la calidad durante la implementación continua.

1. **Resultados de la evaluación**

Los resultados de la evaluación de artefactos incluyen

* **Informe de Errores:** Identificación y documentación de errores, vulnerabilidades y problemas de rendimiento.
* **Comentarios y Sugerencias:** Retroalimentación sobre cómo mejorar la estructura del código, la eficiencia y la legibilidad.
* **Veredicto de Calidad:** Determinación de si los artefactos cumplen con los estándares y las expectativas de calidad.
* **Recomendaciones de Mejora**: Propuestas para optimizar el rendimiento, la seguridad y la eficiencia del software.