TPO - PARTE II

Objetivo:

El objetivo de este trabajo es aplicar conceptos de calidad de producto en software mediante la evaluación de atributos de calidad, pruebas, integración continua y administración de la configuración. Se espera que los estudiantes analicen un software, propongan estrategias de aseguramiento de la calidad y documenten sus hallazgos.

Formato: Informe escrito + actividad práctica.

Contenido mínimo:

1. **Definición del software elegido** (caso real o caso ficticio).
2. **Análisis de atributos de calidad** con ISO 25010.
3. **Plan de pruebas** (validación, verificación y automatización).
4. **Integración continua** (definir un pipeline básico).
5. **Administración de la configuración** (gestión de cambios y trazabilidad).

Entrega: Informe en PDF + código/documentación en repositorio Git.

Descripción del Trabajo

Cada grupo deberá seleccionar un software (puede ser un desarrollo propio, de código abierto o una aplicación conocida sencilla) y realizar un análisis de calidad basado en los siguientes aspectos:

2. Descripción del Trabajo

Cada estudiante o grupo deberá seleccionar un software (puede ser un desarrollo propio, de código abierto o una aplicación conocida) y realizar un análisis de calidad basado en los siguientes aspectos:

2.1. Definición del Software

- Nombre y descripción del software.
- Objetivos y funcionalidades principales (al menos 3 requerimientos funcionales).
- Contexto de uso y usuarios finales.

2.2. Análisis de Atributos de Calidad

- Aplicar el modelo ISO/IEC 25010 para evaluar la calidad del software.
- Seleccionar al menos cinco atributos de calidad y justificarlos.
- Proponer métricas para evaluar estos atributos.

2.3. Plan de Pruebas

- Diseñar un plan de pruebas que incluya:
    - Pruebas unitarias (automatizadas preferentemente con herramientas como JUnit o PyTest).
    - Pruebas de integración.
    - Pruebas de sistema.
    - Estrategias de validación con usuarios.

2.4. Integración Continua

- Proponer una estrategia de integración continua para el software.
- Describir herramientas y procesos para la automatización de pruebas.
- Implementar (opcional) un pipeline básico utilizando GitHub Actions, Jenkins o GitLab CI.

2.5. Administración de la Configuración y Trazabilidad

- Describir el enfoque de gestión de versiones (uso de Git, estrategias de branching, etiquetado de versiones).
- Explicar la trazabilidad de requisitos hasta los casos de prueba.
- Usar un repositorio para evidenciar el control de cambios (opcional).

3. Entregables

Cada equipo deberá presentar los siguientes elementos:

1. **Informe escrito en PDF** con el desarrollo de los puntos anteriores (máximo 20 páginas).
2. **Código fuente y documentación** (si aplica) en un repositorio de GitHub o similar.
3. **EXPOSICIÓN: Presentación breve en la fecha indicada en cronograma de la asignatura,** en formato de diapositivas con los hallazgos principales. Participación de todos los integrantes del equipo. Límite de exposición por integrante: de 3 min. a 5 min máximo.

4. Criterios de Evaluación

El trabajo se evaluará con base en los siguientes criterios:

- **Claridad y profundidad del análisis (30%)**: Explicación detallada y fundamentada.
- **Aplicación de conceptos de calidad (30%)**: Uso adecuado de normas y metodologías.
- **Implementación práctica (20%)**: Uso de herramientas y evidencia de ejecución.
- **Presentación y documentación (20%)**: Organización, claridad y calidad del informe.

5. Plazo de Entrega

- Fecha límite: Indicado en cronograma.
- Modo de entrega: Subida del informe en el canal de Teams del equipo , con el link al repositorio en GitHub.