



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUÍA DE APRENDIZAJE: Guía – **Aplicación de calidad Software**

- Denominación del Programa de Formación: **Tecnólogo en Análisis y desarrollo de Software**
- Código del Programa de Formación: **228106**
- Nombre del Proyecto: **Sistema Integral Web Para Gestión De Procesos Educativos Del CSF.**
- Fase del Proyecto: **Evaluación.**
- Actividad de Proyecto: **Determinar la estructura lógica del sistema.**
- Competencia: **38369 - Controlar la calidad del servicio de software de acuerdo con los estándares técnicos.**
- Resultados de Aprendizaje Alcanzar: **593145 - 03 Realizar Actividades de Mejora de la Calidad del Software a Partir de los Resultados de la Verificación.**
- Duración de la Guía: **36 HORAS**

2. PRESENTACIÓN

- El uso de herramientas como **Codacy** y **SonarQube** es fundamental para garantizar la **calidad del software**, ya que permiten analizar el código fuente en busca de errores, vulnerabilidades de seguridad, duplicaciones y malas prácticas que afectan la mantenibilidad del sistema. Estas plataformas automatizan la revisión del código, fomentan la mejora continua y aseguran que los equipos de desarrollo cumplan con los **estándares de calidad definidos por la norma ISO/IEC 25010**, la cual establece los atributos esenciales del producto de software, como la confiabilidad, eficiencia, seguridad y mantenibilidad.

En consecuencia, su implementación contribuye directamente al desarrollo de soluciones más seguras, sostenibles y alineadas con buenas prácticas internacionales de ingeniería de software.



3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 – Foro - Base de datos. Duración: 3 horas - Trabajo: Individual

Para iniciar con esta actividad nos haremos las siguientes preguntas.

1. **¿Qué entiendes por calidad del software** y por qué consideras que es importante garantizarla durante el ciclo de desarrollo?
2. **¿Qué tipos de errores o defectos** puede detectar una herramienta de análisis de código como **SonarQube o Codacy**?
3. Según la **norma ISO/IEC 25010**, ¿cuáles son las **características principales** que debe tener un software de calidad?
4. **¿Por qué es importante integrar la revisión de código (Code Review)** dentro del proceso de desarrollo en equipo?

Esta investigación y conclusión se evidenciara un informe con la entrega final.

3.2 – Taller Práctico — Revisión de Calidad de Software Web. Duración: 33 horas.

Objetivo de la actividad

Desarrollar una aplicación web funcional con estructura completa (Login, Dashboard, CRUD de productos y tipos de productos), implementando buenas prácticas de programación y realizando una **revisión de calidad del código** mediante herramientas especializadas como **SonarQube o Codacy** evidenciando los siguiente items.

1. Aplicar los conceptos de autenticación, administración y manejo de datos en un entorno web.
2. Implementar operaciones CRUD (crear, leer, actualizar, eliminar) para la gestión de productos.
3. Integrar un sistema de control de versiones mediante GitHub.
4. Evaluar la calidad del código y realizar correcciones con base en un análisis automatizado.

Descripción del taller

El aprendiz deberá crear **una página web funcional** basada en el **tema otorgado por el instructor**, la cual cuenta con:

- Un **módulo de autenticación (Login)**.
- Un **dashboard o panel de administración**.
- Un **CRUD de dos tablas**:
 - productos
 - tipos_productos



El proyecto deberá publicarse en **GitHub** y someterse a una revisión de calidad con **SonarQube (Cloud o Desktop)** o **Codacy**.

Posteriormente, se analizarán los **errores encontrados antes y después de su corrección**, documentando el proceso.

Para el desarrollo de la actividad se deben seguir las siguientes instrucciones descritas en el material de apoyo suministrado por el instructor.



Entregables

1. Carpeta del proyecto web completo (código fuente).
2. Link del repositorio GitHub público.
3. Capturas de:

- ✓ Login
- ✓ Dashboard
- ✓ CRUD
- ✓ Escaneo en SonarQube o Codacy
- ✓ Correcciones de errores antes y después

4. Documento PDF con:

- ✓ Descripción del desarrollo.
- ✓ Resultados del análisis de calidad.
- ✓ Tabla comparativa de errores corregidos.
- ✓ Conclusiones personales.

4. PLANTEAMIENTO DE EVIDENCIAS DE APRENDIZAJE PARA LA EVALUACIÓN EN EL PROCESO FORMATIVO.

Fase del proyecto formativo	Actividad del proyecto formativo	Actividad de Aprendizaje	Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación



5. GLOSARIO DE TÉRMINOS

- **Mantenibilidad:** Capacidad de un software para ser modificado, corregido o mejorado de manera eficiente sin introducir nuevos errores. Una alta mantenibilidad facilita la evolución del sistema a lo largo del tiempo.
- **Confiabilidad:** Grado en que un sistema realiza sus funciones de manera consistente y sin fallos bajo condiciones establecidas. Un software confiable reduce los errores y aumenta la satisfacción del usuario.
- **Usabilidad:** Facilidad con la que los usuarios pueden aprender, comprender y operar un software. Involucra aspectos de diseño, claridad de interfaz y experiencia del usuario (UX).
- **Eficiencia del rendimiento:** Relación entre los recursos utilizados (tiempo, memoria, procesamiento) y el nivel de desempeño alcanzado. Un software eficiente optimiza sus recursos para ofrecer respuestas rápidas y estables.
- **Seguridad:** Capacidad del software para proteger la información y evitar accesos no autorizados, vulnerabilidades o pérdida de datos. Es uno de los pilares de la calidad definidos en la norma **ISO/IEC 25010**.

6. REFERENTES BIBLIOGRÁFICOS

- International Organization for Standardization. (2011). ISO/IEC 25010:2011 – Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Ginebra: ISO. Disponible en: <https://www.iso.org/standard/35733.html>
- Sommerville, I. (2016). Software Engineering (10th ed.). Pearson Education. Disponible en: <https://www.pearson.com/en-us/subject-catalog/p/software-engineering/P200000003758>

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Javier Leonardo Pineda Uribe	Instructor G19	Planta - Provisional	8 de Octubre de 2025

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					