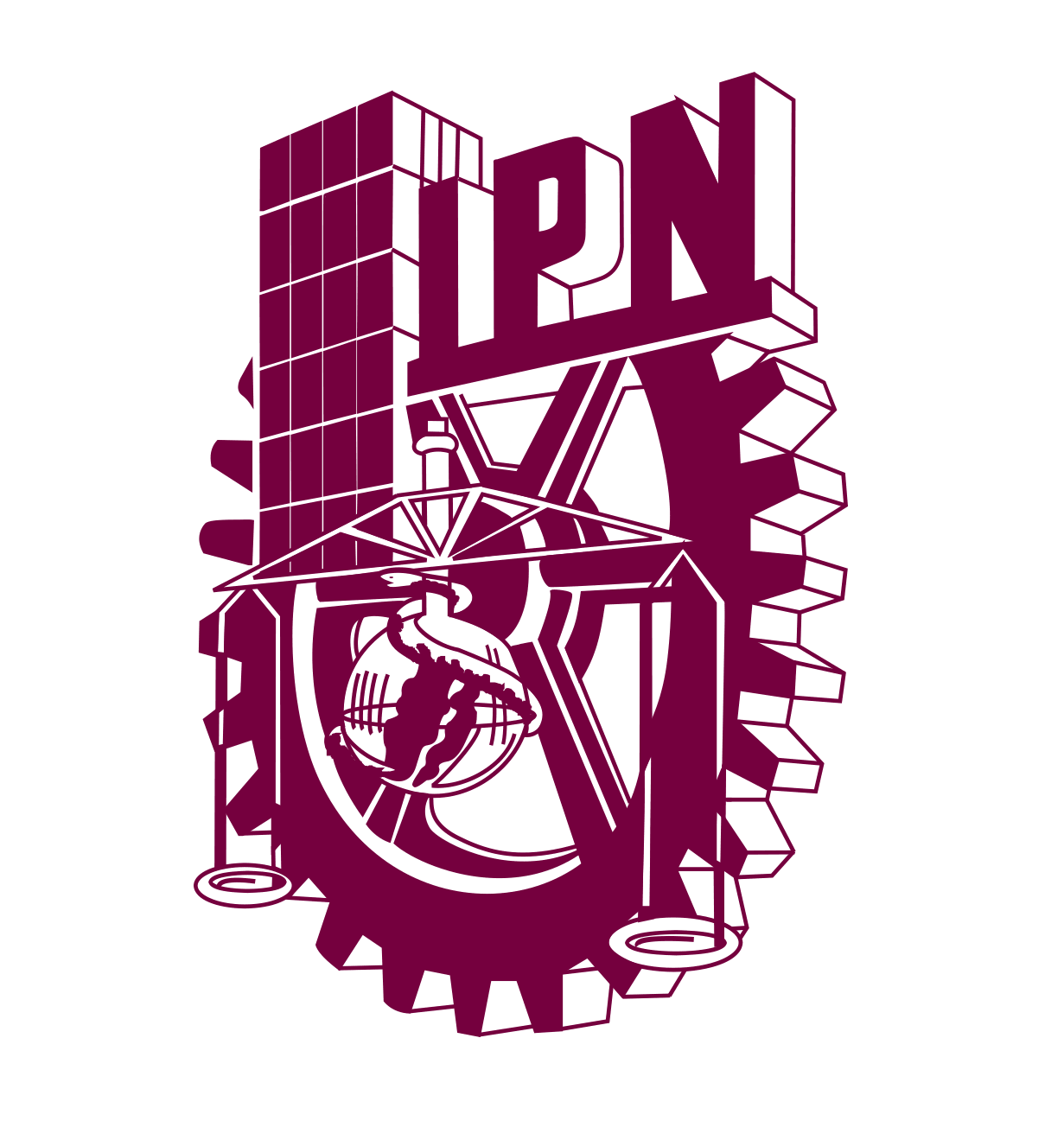
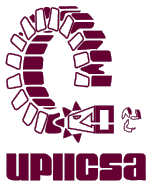
**INSTITUTO POLITÉCNICO NACIONAL**

**Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas**

**Ingeniería de Pruebas**

**Documento Técnico: Plan de Aseguramiento y Control de Calidad (ACS/SQA) - Calculadora en Python con Tkinter**

**Autores:**

* Contreras Mosco Cristobal
* Díaz Pérez Diego
* Hernández Aguirre Ricardo
* Monroy Muñoz Angel Yael
* Salazar Rocha Any Jennifer

**Fecha:** [09/09/2025]

**Versión:** 1.0

**Índice**

1. Introducción
2. Objetivo
3. Alcance
4. Estándares de Calidad Adoptados
5. Procedimientos de Aseguramiento de Calidad

5.1. Revisión de código

5.2. Revisión de interfaz

5.3. Control de versiones

5.4. Gestión de defectos

5.5. Pruebas de calidad

1. Roles y Responsabilidades

6.1. Analista/Desarrollador

6.2. Usuario final (Validador)

1. Actividades del Aseguramiento de Calidad
2. Métricas de Calidad
3. Reporte y Seguimiento de Defectos
4. Conclusiones

**Introducción**

El presente documento establece el Plan de Aseguramiento y Control de Calidad del Software (ACS/SQA) para el proyecto Calculadora con Interfaz Gráfica. El objetivo es garantizar que el producto final cumpla con los requerimientos funcionales y no funcionales definidos, asegurando su confiabilidad, precisión y facilidad de uso. El aseguramiento de la calidad es fundamental en el desarrollo de software, ya que permite minimizar errores, reducir riesgos y asegurar la satisfacción del usuario final.

**Objetivo**

Definir las políticas, estándares y procedimientos que aseguren la calidad del software. Establecer mecanismos de control para detectar y corregir errores en etapas tempranas. Verificar que el producto cumpla con los requerimientos establecidos en el documento de requisitos. Validar que el software entregue resultados correctos, confiables y en el tiempo esperado.

**Alcance**

El plan se aplicará a todas las fases del desarrollo del sistema de calculadora:

1. Análisis: Revisión y validación de requerimientos.
2. Diseño: Revisión de diagramas y estructura de la interfaz.
3. Implementación: Verificación de estándares de codificación (PEP8 en Python).
4. Pruebas: Ejecución de casos de prueba unitarios, de integración y de sistema.

**Estándares de Calidad Adoptados**

* Estándares de programación en Python: Cumplimiento de la guía PEP8.
* Interfaz gráfica: Uso de Tkinter con controles estandarizados.
* Pruebas: Planificadas de acuerdo con el estándar IEEE 829 para documentación de pruebas.
* Control de defectos: Registro de errores con descripción, pasos para reproducirlos y estatus de resolución.

**Procedimientos de Aseguramiento de Calidad**

**Revisión de código**

El desarrollador validará la sintaxis y consistencia de los módulos antes de integrarlos. Se usarán pruebas unitarias para cada método (ejemplo: realizar\_calculo, borrar\_ultimo\_caracter).

**Revisión de interfaz**

Validar que los botones tengan etiquetas claras y que la disposición sea intuitiva.

Verificar que los errores (ejemplo: división por cero) se muestren correctamente en la pantalla.

**Control de versiones**

Respaldo de versiones del código fuente en etapas clave (al final de cada fase).

**Gestión de defectos**

Registro de errores en una bitácora que incluye: descripción, prioridad, estado (pendiente, en corrección, resuelto).

**Pruebas de calidad**

Ejecución de pruebas con datos válidos e inválidos. Validación de precisión en resultados numéricos (dos decimales máximos). Validación del funcionamiento del historial.

**Roles y Responsabilidades**

**Analista/Desarrollador**

Implementar el sistema con estándares de calidad. Ejecutar pruebas unitarias e integrar los resultados en la bitácora. Documentar incidencias y soluciones.

**Usuario final (Validador)**

Probar la aplicación en condiciones reales. Reportar errores de usabilidad o mal funcionamiento.

**Actividades del Aseguramiento de Calidad**

| Actividad | Responsable | Momento de ejecución | Evidencia |
| --- | --- | --- | --- |
| Revisión de requerimientos | Analista | Fase de análisis | Documento validado |
| Revisión de diseño | Desarrollador | Fase de diseño | Diagramas revisados |
| Revisión de código | Desarrollador | Durante implementación | Código conforme a PEP8 |
| Ejecución de pruebas unitarias | Desarrollador | Durante pruebas | Reporte de resultados |
| Validación por usuario | Usuario final | Pruebas de aceptación | Retroalimentación |

**Métricas de Calidad**

Para asegurar un producto confiable, se utilizarán las siguientes métricas:

1. Densidad de defectos.
2. Cobertura de pruebas: % de funciones verificadas mediante pruebas.
3. Cumplimiento de requerimientos: % de requerimientos implementados frente a los definidos.
4. Tasa de corrección de errores: Tiempo promedio para resolver defectos.
5. Satisfacción del usuario: Valoración de usabilidad (escala 1–5).

**Reporte y Seguimiento de Defectos**

Todos los errores detectados se registrarán en una bitácora con los siguientes campos:

1. ID del error.
2. Fecha de detección.
3. Descripción del problema.
4. Módulo afectado.
5. Prioridad (Alta, Media, Baja).
6. Estado (Pendiente, En proceso, Resuelto).
7. Responsable.

Este registro permitirá llevar un control claro de los errores y verificar su resolución antes de la entrega final.

**Conclusiones**

El Plan ACS/SQA asegura que la calculadora desarrollada en Python/Tkinter cumpla con los estándares de calidad, confiabilidad y usabilidad. La correcta aplicación de este plan permitirá minimizar riesgos, garantizar la precisión de los cálculos y ofrecer una experiencia positiva al usuario final.