 **Facultad de Ingeniería – TDSC - UNSTA**

**Materia:** Fundamentos del Control de Calidad – 1° Trabajo Práctico.

**Profesor:** Ing. Tulio Ruesjas Martín. **Fecha de Entrega :** 20 de marzo de 2025

**Nombre y Apellido del Alumno:** Maximiliano Medina

**Objetivo:**

El objetivo de este trabajo práctico es que los alumnos comprendan los conceptos básicos del control de calidad en el desarrollo de software, así como la importancia de estas prácticas en la entrega de productos de software de alta calidad. También, se espera que los estudiantes identifiquen los roles y responsabilidades clave en el proceso de testing y reconozcan los beneficios que esta actividad aporta al desarrollo de software.

**Temario:**

1. **Introducción al Control de Calidad:** Investiga y redacta una breve introducción al concepto de control de calidad en el desarrollo de software. Incluye su definición y la importancia de aplicar prácticas de control de calidad en cada etapa del ciclo de vida del software.
2. **Importancia del Testing en el Proceso de Desarrollo:** Explica en qué consiste el testing en el contexto del desarrollo de software. Describe por qué el testing es una parte crucial del proceso de desarrollo de software y cómo contribuye a la entrega de productos de calidad.
3. **Objetivos y Beneficios del Testing:** Identifica cuales son los objetivos del testing en el desarrollo de software. Enumera y explica al los beneficios que el testing aporta al proceso de desarrollo de software.
4. **Roles y Responsabilidades en el Proceso de Testing:** Describe los roles clave en el proceso de testing de software. Detalla las responsabilidades asociadas a cada uno de los roles identificados.

**Formato del Trabajo:**

1. El trabajo deberá ser entregado en formato PDF.
2. Utiliza una estructura clara y coherente para cada sección del trabajo.
3. Cita adecuadamente las fuentes utilizadas para respaldar tus respuestas.
4. La extensión total del contenido del trabajo práctico no debe exceder las 6 páginas. El documento debe tener en su primera hoja el indice del documento.
5. El trabajo debe ser enviado a: tulio.ruesjas@unsta.edu.ar hasta las 14 horas del día 20 de marzo del 2025. TP enviado después de esa hora será rechazado.

**DESARROLLO**

1. El **control de calidad en el desarrollo de software es** un conjunto de Implementar prácticas de control de calidad en cada etapa del ciclo de vida del software—desde la planificación y el diseño hasta la implementación, pruebas y mantenimiento—permite detectar y corregir errores de manera temprana, reducir costos y mejorar la seguridad y el rendimiento del producto final. Métodos como revisión de código, pruebas automatizadas y auditorías de calidad son fundamentales para garantizar un software robusto.
2. El **Testing** en el desarrollo de software es el proceso de evaluar y verificar que un sistema o aplicación funcione correctamente y cumpla con los requisitos definidos. Consiste en ejecutar el software bajo diferentes condiciones para identificar defectos, verificar su desempeño y asegurar su calidad antes de su lanzamiento. Se pueden emplear diversas técnicas, como pruebas manuales o automatizadas, y distintos niveles de prueba, como pruebas unitarias, de integración, de sistema y de aceptación.

Es una parte crucial del proceso de desarrollo de software por varias razones:

**Detección temprana de errores:** Identificar defectos en las primeras etapas del desarrollo reduce los costos y el esfuerzo de corrección.

**Garantía de funcionalidad y usabilidad:** Permite asegurar que el software cumple con los requisitos del usuario y brinda una experiencia satisfactoria.

**Seguridad y estabilidad:** Detecta vulnerabilidades y previene fallos que puedan comprometer la integridad del sistema.

**Optimización del rendimiento:** Evalúa la eficiencia del software bajo diferentes condiciones de carga y uso.

**Facilidad de mantenimiento y escalabilidad:** Un software bien probado es más fácil de actualizar y expandir sin introducir nuevos errores.

El testing es fundamental para la entrega de productos de alta calidad, ya que permite validar que el software sea confiable, seguro y eficiente. Al aplicar pruebas en cada etapa del desarrollo, se minimiza el riesgo de fallos en la producción, mejorando la satisfacción del usuario y la reputación del producto en el mercado.

1. El testing en el desarrollo de software tiene varios objetivos clave, entre ellos:

**Detectar y corregir errores :** Identificar defectos en el software antes de su implementación en producción.

**Verificar el cumplimiento de requisitos :** Asegurar que el software funcione según las especificaciones establecidas.

**Garantizar la calidad del producto :** Mejorar la estabilidad, seguridad y rendimiento del software.

**Reducir costos y tiempos de desarrollo :** Detectar problemas temprano evita gastos elevados en correcciones posteriores.

**Facilitar el mantenimiento y evolución del software :** Asegurar que futuras modificaciones no introduzcan nuevos errores.

**Mejorar la experiencia del usuario :** Garantizar que el software sea intuitivo, funcional y libre de fallos críticos.

El **testing** aporta múltiples **beneficios** al proceso de desarrollo de software, entre los que destacan:

* Mayor confiabilidad y estabilidad
* Asegúrese de que el software funcione de manera predecible y sin errores críticos.
* Optimización del rendimiento
* Permite evaluar cómo responde el software bajo diferentes condiciones y cargas de trabajo.
* Reducción de costos a largo plazo
* Identificar y corregir errores en etapas tempranas minimizar los gastos de mantenimiento.
* Mayor seguridad
* Detecta vulnerabilidades que podrían ser explotadas por atacantes.
* Entrega de productos con mayor calidad.
* Asegúrese de que el software cumpla con los estándares de calidad antes de su lanzamiento.
* Facilitar la integración y compatibilidad
* Permite verificar que el software funciona correctamente en diferentes entornos, dispositivos o plataformas.
* Mejora la satisfacción del usuario
* Un software libre de errores y con un buen rendimiento proporciona una mejor experiencia al usuario final.

1. **Roles Clave en el Proceso de Testing de Software**

**Tester / Analista de Calidad**

**Responsabilidades:**

* Diseñar y ejecutar casos de prueba manuales o automatizados.
* Identificar, documentar y hacer seguimiento de errores o defectos.
* Validar que el software cumpla con los requisitos funcionales y no funcionales.
* Colaborar con desarrolladores y otros equipos para mejorar la calidad del software.

**Ingeniero de Calidad de Software**

**Responsabilidades:**

* Desarrollar estrategias y planes de prueba.
* Automatizar pruebas para mejorar la eficiencia del testing.
* Diseñar e implementar frameworks de pruebas automatizadas.
* Identificar mejoras en los procesos de calidad y asegurar su implementación.

**Líder o Gerente de Pruebas**

**Responsabilidades:**

* Definir la estrategia de pruebas y planifica
* Gestionar los recursos y asignar tara
* Supervisar la calidad del software.
* Coordinar la comunicación entre testers, d
* Evaluar y reportar el progreso de las pruebas.

**Desarrollador de Software en Pruebas (SDET – Desarrollador de Software en Pruebas)**

**Responsabilidades:**

* Diseñar, desarrollar y mantener herramientas y frameworks de testing automatizados.
* Implementar pruebas a nivel de código (unitarias, integración, regresión)
* Colaborar con desarrolladores en la mejora del código y la calidad del software.
* Asegurar que las pruebas se integran en el proceso de desarrollo.

**Probador de rendimiento (Tester de Rendimiento)**

**Responsabilidades:**

* Evaluar la capacidad de respuesta, estabilidad y escalabilidad del software.
* Realice pruebas de carga, estrés y volumen para medir el rendimiento del sistema.
* Identificar cuellos de botella y proponer mejoras de optimización.
* Usar herramientas especializadas como JMeter, LoadRunner, entre otras.

**Probador de Seguridad**

**Responsabilidades:**

* Evaluar la seguridad del software y detectar vulnerabilidades.
* Realizar pruebas de penetración y análisis de riesgos.
* Implementar estrategias de pruebas para prevenir ataques y brechas de seguridad.
* Colaborar con equipos de desarrollo y seguridad para mejorar la protección del software.

**Tester de Aceptación de Usuario**

**Responsabilidades:**

* Verifique que el software cumpla con los requisitos del usuario final.
* Realizar pruebas en entornos reales de negocio antes de la implementación.
* Proporcionar comentarios sobre la usabilidad y funcionalidad del software.
* Validar que el producto final es adecuado para su lanzamiento.