

# **Actividad 3 - Aseguramiento De La Calidad**

## **Ingeniería en Desarrollo de Software**

**Tutor: Jessica Hernández**

**Alumno: José Luis Rodríguez Blancas**

**Fecha: 4/05/2024**

## Índice

Introducción .....	3
Descripción .....	3
Justificación .....	4
Desarrollo .....	5
Plan de Pruebas Integral .....	5
Reglas para el versionado e incidencias .....	8
Conclusión .....	9

## **Introducción**

En el contexto del desarrollo de software, la calidad es un factor crucial que determina el éxito de un proyecto. Una parte fundamental para garantizar la calidad es la realización de pruebas exhaustivas y sistemáticas. En esta actividad, nos enfocaremos en diseñar un plan integral de pruebas que nos permita validar el software de manera efectiva y eficiente.

El plan de pruebas integral no solo incluirá la definición de las pruebas a aplicar, sino también un cronograma detallado que establecerá los tiempos y recursos necesarios para cada etapa del proceso. Asimismo, se definirán los roles y responsabilidades de cada miembro del equipo de desarrollo, asegurando una distribución clara de tareas y una comunicación efectiva.

Además, se establecerán las reglas para el versionado del software, lo cual es fundamental para mantener un control preciso de las diferentes versiones y cambios realizados durante el desarrollo. Asimismo, se definirán las reglas para el manejo correcto de incidencias, garantizando que cualquier problema identificado durante las pruebas sea registrado, evaluado y solucionado de manera adecuada.

En resumen, esta actividad busca establecer un marco de trabajo sólido y estructurado que nos permita llevar a cabo las pruebas de forma eficiente y con altos estándares de calidad.

## **Descripción**

En el contexto presentado, se destaca la importancia de diseñar un plan de pruebas integral para garantizar la calidad del software desarrollado. Este plan no solo implica la definición de las pruebas a realizar, sino también la creación de un cronograma detallado y la asignación de roles dentro del equipo de desarrollo.

Esto asegura que las pruebas se realicen de manera organizada y efectiva, optimizando así los recursos y el tiempo dedicados a esta etapa crucial del desarrollo.

Además, se hace hincapié en la necesidad de establecer reglas claras para el versionado del software y el manejo de incidencias. El versionado permite mantener un registro preciso de los cambios realizados en el código, facilitando la identificación y corrección de errores. Por otro lado, el manejo correcto de incidencias garantiza que los problemas detectados durante las pruebas sean abordados de manera adecuada, evitando que afecten la calidad final del producto.

En resumen, esta actividad busca asegurar que el proceso de pruebas se realice de manera estructurada y eficiente, contribuyendo así a la entrega de un software de alta calidad que cumpla con las expectativas y necesidades de los usuarios.

## **Justificación**

La utilización de un plan de pruebas integral es fundamental en el desarrollo de software por varias razones. En primer lugar, ayuda a identificar y corregir errores en el código antes de que el producto final se ponga en producción, lo que reduce significativamente los costos y el tiempo dedicado a corregir problemas en etapas más avanzadas del desarrollo.

Además, un plan de pruebas integral permite asegurar que el software cumpla con los requisitos funcionales y no funcionales especificados, garantizando así la satisfacción del cliente y la calidad del producto final. También facilita la detección temprana de posibles problemas de rendimiento, seguridad o usabilidad, lo que contribuye a mejorar la experiencia del usuario y la reputación de la empresa.

Por otro lado, el establecimiento de reglas para el versionado y el manejo de incidencias permite mantener un control riguroso sobre los cambios realizados en el código y facilita la colaboración entre los miembros del equipo de desarrollo. En resumen, emplear un plan de pruebas integral es fundamental para garantizar la calidad, la eficiencia y el éxito de un proyecto de desarrollo de software.

Desarrollo

Plan de Pruebas Integra		
**Actividad**	**Responsable**	**Fecha**
Preparación		
Definir alcance y objetivos de las pruebas	Gerente de Proyecto	Semana 1
Configurar el entorno de pruebas	Líder de Pruebas	Semana 1
Ejecución de Pruebas		
Pruebas unitarias	Equipo de Pruebas	Semana 2
Pruebas de integración	Equipo de Pruebas	Semana 3
Pruebas de sistema	Equipo de Pruebas	Semana 4
Análisis de Resultados y Correcciones		
Evaluar resultados de las pruebas	Líder de Pruebas	Semana 5
Corregir errores identificados	Equipo de Desarrollo	Semana 5
Finalización		
Documentar resultados y conclusiones	Líder de Pruebas	Semana 6
Preparar informe final	Gerente de Proyecto	Semana 6

Plan de Pruebas Integral

1. Objetivo del Plan de Pruebas:
- Validar la funcionalidad, rendimiento, seguridad y usabilidad del software desarrollado, garantizando que cumple con los requisitos especificados.

## 2. Cronograma de Actividades:

- Semana 1: Preparación
  - Definir alcance y objetivos de las pruebas.
  - Configurar el entorno de pruebas.
- Semana 2-4: Ejecución de Pruebas
  - Pruebas unitarias: Semana 2.
  - Pruebas de integración: Semana 3.
  - Pruebas de sistema: Semana 4.
- Semana 5: Análisis de Resultados y Correcciones
  - Evaluar resultados de las pruebas y corregir errores identificados.
- Semana 6: Finalización
  - Documentar resultados y conclusiones.
  - Preparar informe final.

## 3. Roles:

- Gerente de Proyecto: Responsable de la supervisión general del plan de pruebas y la coordinación del equipo.
- Líder de Pruebas: Encargado de diseñar, ejecutar y supervisar las pruebas, así como de gestionar los recursos necesarios.
- Equipo de Desarrollo: Responsable de corregir los errores identificados durante las pruebas.
- Equipo de Pruebas: Encargado de ejecutar las pruebas según lo planificado y reportar los resultados.

## 4. Estándares de Calidad:

- Cumplir con los requisitos funcionales y no funcionales especificados en el documento de requisitos.
- Utilizar buenas prácticas de programación y diseño.
- Garantizar la seguridad y privacidad de los datos.
- Cumplir con los estándares de usabilidad y accesibilidad.

#### 5. Herramientas de Pruebas:

- Se utilizarán herramientas como Selenium para pruebas automatizadas, JUnit para pruebas unitarias, y JIRA para el seguimiento de incidencias y el versionado del software.

#### 6. Comunicación:

- Se realizarán reuniones regulares para informar sobre el progreso de las pruebas y discutir posibles problemas.
- Se utilizarán herramientas de comunicación como Slack y correo electrónico para mantener al equipo informado.

#### 7. Documentación:

- Se generará documentación detallada de las pruebas realizadas, los resultados obtenidos y las correcciones aplicadas.

Este plan de pruebas integral garantiza que el software sea probado de manera exhaustiva y estructurada, siguiendo estándares de calidad rigurosos y asegurando la entrega de un producto final de alta calidad y confiabilidad.

## Reglas para el versionado e incidencias

### Reglas para el Versionado:

1. Se utilizará un sistema de control de versiones (Git) para gestionar el código fuente del software.
2. Se seguirá la convención de nomenclatura "Major.Minor.Patch" para las versiones del software, donde:
  - Major: Incrementa cuando se realizan cambios significativos que pueden afectar la compatibilidad hacia atrás.
  - Minor: Incrementa cuando se agregan nuevas funcionalidades de forma compatible con versiones anteriores.
  - Patch: Incrementa cuando se realizan correcciones de errores menores o mejoras que no afectan la compatibilidad.
3. Se crearán ramas (branches) separadas para el desarrollo de nuevas funcionalidades (feature branches) y para la corrección de errores (bugfix branches).
4. Antes de fusionar una rama de desarrollo en la rama principal (master), se realizará una revisión de código (code review) por parte de al menos dos miembros del equipo.
5. Se utilizarán etiquetas (tags) para marcar versiones específicas del software que se consideren estables y listas para ser implementadas en producción.

### Reglas para el Manejo de Incidencias:

1. Todas las incidencias identificadas durante las pruebas serán registradas en un sistema de seguimiento de incidencias (JIRA).



2. Cada incidencia deberá incluir una descripción detallada del problema, pasos para reproducirlo, y cualquier información relevante.
3. Las incidencias se clasificarán según su gravedad (crítica, alta, media, baja) y su prioridad (urgente, alta, normal, baja).
4. Cada incidencia asignada a un miembro del equipo de desarrollo deberá ser resuelta dentro de un plazo acordado, dependiendo de su prioridad y gravedad.
5. Una vez resuelta una incidencia, se realizará una verificación por parte del equipo de pruebas para asegurar que el problema ha sido corregido satisfactoriamente antes de cerrarla.
6. Se mantendrá una comunicación constante entre el equipo de desarrollo y el equipo de pruebas para garantizar una resolución rápida y eficiente de las incidencias identificadas.

## **Conclusión**

La elaboración de un plan de pruebas integral, junto con la definición de reglas para el versionado y manejo de incidencias, es fundamental en el campo laboral del desarrollo de software y tiene un impacto significativo en la calidad y éxito de los proyectos.

En mi vida cotidiana, estos principios también son relevantes, ya que la planificación cuidadosa y la organización son clave para alcanzar metas y objetivos de manera efectiva. La importancia de establecer reglas y procedimientos claros se refleja en la necesidad de mantener un orden y una estructura que faciliten el logro de los resultados deseados.

Además, la aplicación de estos principios en el ámbito laboral fomenta la colaboración y comunicación efectiva entre los miembros del equipo, lo que no solo mejora la calidad del trabajo realizado, sino que

también contribuye al desarrollo de habilidades de liderazgo, resolución de problemas y toma de decisiones.

En resumen, la implementación de un plan de pruebas integral y el establecimiento de reglas claras para el versionado y manejo de incidencias son prácticas fundamentales que pueden tener un impacto positivo tanto en el ámbito laboral como en la vida cotidiana, al promover la eficiencia, la calidad y el éxito en las actividades que realizamos.