

## **PHASE 3**

### **Software Testing Techniques**

**Juan David Nieves Galindo**

**Diego Andrés Catilla Lázaro**

**Paul Alejandro Ortiz**

**Grupo Colaborativo  
204027\_4**

**Tutor**

**MARÍA BENARDA PARDO**

**UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA (UNAD).  
ESCUELA DE CIENCIAS BÁSICAS TECNOLOGÍA E INGENIERA  
TECNOLOGIA EN DESARROLLO DE SOFTWARE**

**BARBOSA**

**2022**

## Contenido

Grupo Colaborativo 204027_4 .....	1
SOFTWARE TESTING (PRUEBAS DE SOFTWARE) .....	2
¿Necesario? .....	3
ROLES Y MÁS .....	3
Test Lead/Manager (Gerente de pruebas).....	3
Test engineers/QA testers (Ingenieros especializados en pruebas) .....	3
Software Tester (Ensayador de software).....	3
Software Test Automator Role (Encargado de la automatización).....	3
TIPOS DE PRUEBAS .....	4
Pruebas funcionales .....	4
Pruebas no funcionales .....	4
Pruebas de la interfaz gráfica de usuario .....	4
Pruebas de regresión .....	5
¿Cuáles son las fases del ciclo de vida de las pruebas de software? Describa cada fase. ....	5
Requisitos: .....	5
Planificación: .....	5
Diseño:.....	6
Implementación .....	6
Ejecución: .....	6
Cierre: .....	6
BIBLIOGRAFÍA.....	7

## SOFTWARE TESTING (PRUEBAS DE SOFTWARE)

En todo proyecto de software siempre hay fases que se distinguen de las demás por su gran importancia para el desarrollo de nuevos aplicativos de software que disciernen de unos especialistas encargados (de manera general) de; planear, investigar, documentar, automatizar y ejecutar diferentes formas de probar las opciones que tiene un sistema para ejecutar una serie de procesos de manera sistematizada.

Respecto a la diferencias que hay entre la garantía de calidad y las pruebas técnicas de software, no hay muchas, solo que entre una y otra se enfocan en procesos diferentes y a aspectos diferentes en la finalización de un software. En esencia la calidad refiere a evaluar

cómo se ejecutó el ciclo de vida del software en sí. Que por otro lado, las pruebas técnicas, referencias al funcionamiento final y esencial de la aplicación, es decir al código, así que por ende la podemos inter-clasificar, dentro de la pruebas de calidad.

También podemos identificarlo como la identificación de falencias reales, de manera rigurosa. Que desean encontrar aquellas cosas que no se miran a simple vista, también, entra en juego un poco la compatibilidad y las interfaces que tiene un software.

### ¿Necesario?

Es completamente necesario, es bien sabido que muchos robos, mal entendidos, errores de registros etc. Los cuales, son debido a la cantidad de bugs que solían quedar en las aplicaciones, esto no solamente causa que un dato se vaya mal, sino que un usuario cualquiera pueda vulnerar la seguridad, o por el contrario el kernel de cualquier aplicación, sacando de ésta manera, provecho de todas las debilidades de un sistema. Puede conllevar a robo de BigDatas y dinero de bancos o cualquier tienda en general.

## ROLES Y MÁS

### Test Lead/Manager (Gerente de pruebas)

Define de manera generalizada el cronograma de actividades para los demás miembro del equipo (subordinados). Por otro lado, asigna las responsabilidades de cada persona, ya sea, tester, ingenieros encargados, supervisores, automatizado, todo esto de acuerdo al perfil profesional. También en esencia verifica que cada fase vaya de buena manera con los recursos asignados para el proceso de implementación. También actualiza los requerimientos según como va avanzando el proyecto, para así de ésta manera encontrar un balance adecuado para la documentación del mismo.

### Test engineers/QA testers (Ingenieros especializados en pruebas)

Por decirlo de alguna forma, ellos son los encargados de leer todos los documentos anteriormente bien documentados para el desarrollo de la aplicación; a tiempo en que definen los casos de pruebas como un equipo, también se encargan de pasar factura de las cosas, implementos, softwares y licencias. Igualmente, son encargados de ejecutar y priorizar casos de prueba de manera correcta.

Todo esto converge en la documentación de las correcciones que se han hecho para el perfeccionamiento del software. Llevando de manera controlada las regresiones y complicaciones tenidas durante el proceso de testeo.

### Software Tester (Ensayador de software)

De manera general, el ingeniero de ensayo de software, es en esencia un analista de datos y de detalles, pues es capaz de analizar y crear casos de pruebas para el mejoramiento del software.

### Software Test Automator Role (Encargado de la automatización)

Al igual que el ingeniero general, éste planea y se encarga de ejecutar los casos de pruebas, pero en éste caso él está especializado en la automatización mediante vpl o diferentes entornos de software.

## TIPOS DE PRUEBAS

### Pruebas funcionales

Las pruebas funcionales son un tipo de pruebas de caja negra en las que los casos de prueba se preparan teniendo en cuenta las especificaciones. Esta forma de prueba se realiza para comprobar si el sistema cumple los requisitos del cliente. Básicamente, en el caso de las pruebas funcionales son importantes las siguientes comprobaciones:

1. El probador debe tener muy clara la funcionalidad que debe realizar la aplicación.
2. Para probar la aplicación es muy importante tener el conjunto correcto de datos (entradas válidas y no válidas)
3. La salida de la aplicación para los datos de prueba proporcionados debe comprobarse según la especificación funcional definida.
4. Los casos de prueba deben cubrir todas las pruebas en los posibles escenarios.
5. El resultado real de una entrada determinada debe registrarse y compararse con el resultado esperado.

Los tipos de pruebas funcionales incluyen:

1. Pruebas unitarias
2. Pruebas de integración
3. Pruebas del sistema
4. Pruebas de regresión
5. Pruebas de aceptación

### Pruebas no funcionales

En un sistema de software hay muchos requisitos, como el rendimiento, la seguridad, la interfaz de usuario, etc., que son de naturaleza no funcional, pero es muy importante probar estos atributos.

Las pruebas de software realizadas para comprobar los atributos no funcionales de un sistema se denominan pruebas no funcionales. Los tipos de metodologías de pruebas no funcionales son los siguientes:

1. Pruebas de rendimiento
2. Pruebas de usabilidad
3. Pruebas de seguridad

### Pruebas de la interfaz gráfica de usuario

El tipo de prueba que se realiza para comprobar si la interfaz gráfica de usuario de una aplicación funciona correctamente se denomina prueba de interfaz gráfica de usuario (GUI). No sólo se comprueba que la interfaz gráfica de usuario funcione correctamente, sino también que se ajuste a las normas de calidad definidas. A continuación, se enumeran algunos de los aspectos más importantes que deben comprobarse durante las pruebas de la interfaz gráfica de usuario:

- Diseño
- Colores
- Fuentes
- Tamaño de letra
- Etiquetas
- Funciones de la caja de pruebas
- Cómo se formatea el texto
- Leyendas
- Botones
- Listas
- Iconos
- Enlaces
- Contenido
- Llaves cortas
- El reloj de arena mientras el sistema procesa los datos

Esta forma de prueba puede ser manual o automática, según le convenga al probador. En ocasiones, las pruebas de la interfaz gráfica de usuario son realizadas por terceras organizaciones y no por el equipo de desarrollo o de pruebas de la empresa.

### **Pruebas de regresión**

Una vez que se detecta un defecto en el sistema, se envía inmediatamente para su corrección. Sin embargo, una vez solucionado el defecto, es importante realizar pruebas intensas para comprobar que los cambios realizados en el código no han afectado a ninguna otra área del sistema. Las pruebas de regresión se llevan a cabo para garantizar que la corrección de errores no ha causado ninguna violación de la funcionalidad o de la lógica de negocio. Las pruebas de regresión ayudan a minimizar las lagunas en el proceso de pruebas. Asegura que la aplicación no tiene defectos antes de que se envíe a la siguiente fase de pruebas.

**¿Cuáles son las fases del ciclo de vida de las pruebas de software? Describa cada fase.**

R/

Las fases del ciclo de vida para las pruebas de software son las siguientes:

#### **Requisitos:**

- En este proceso se crea una lista de requisitos, se hace la comprensión de la viabilidad de los requisitos y también de la automatización.

#### **Planificación:**

- Teniendo en cuenta los requisitos que se abordaran en el proyecto, se define el alcance del proyecto, se realiza el análisis de riesgo y una estimación de prueba.

**Diseño:**

- Se hace una investigación para los escenarios de prueba, se hace una priorización de los casos de prueba y se diseñan en consecuencia.

**Implementación**

- Se detalla la condición de prueba, se identifica los datos de la prueba y se crea las métricas de trazabilidad.

**Ejecución:**

- Se ejecutan los casos de prueba mediante una herramienta de seguimientos de errores, se identifican los comportamientos no deseados en el producto y se hace un documento con los resultados obtenidos; luego se procede a corregir los errores y se prueba de nuevo.

**Cierre:**

- Se verifica que se completen todas las pruebas, luego se evalúan los factores como la calidad, la cobertura de la prueba, el cronograma y el costo y al final se prepara un análisis casual y resolución (CAR).

## BIBLIOGRAFÍA

- Malik S. (2017). Software Testing: Essential Phase of SDLC and a Comparative Study of Software Testing Techniques. International Journal of System & Software Engineering. 38-45. <http://search.ebscohost.com/bibliotecavirtual.unad.edu.co/login.aspx?direct=true&db=aci&AN=128065924&lang=es&site=eds-live&scope=site>
- Morgan, P. (2010). *01: The Fundamentals Of Testing*. BCS Learning & Development Limited. <https://www.proquest.com/docview/1016284402/fulltext/5EF9699F70B046AAPQ/2?accountid=48784#>