El futuro digital es de todos | MinTIC

UNIVERSIDAD EL BOSQUE | MisiónTIC 2022

Ciclo 2

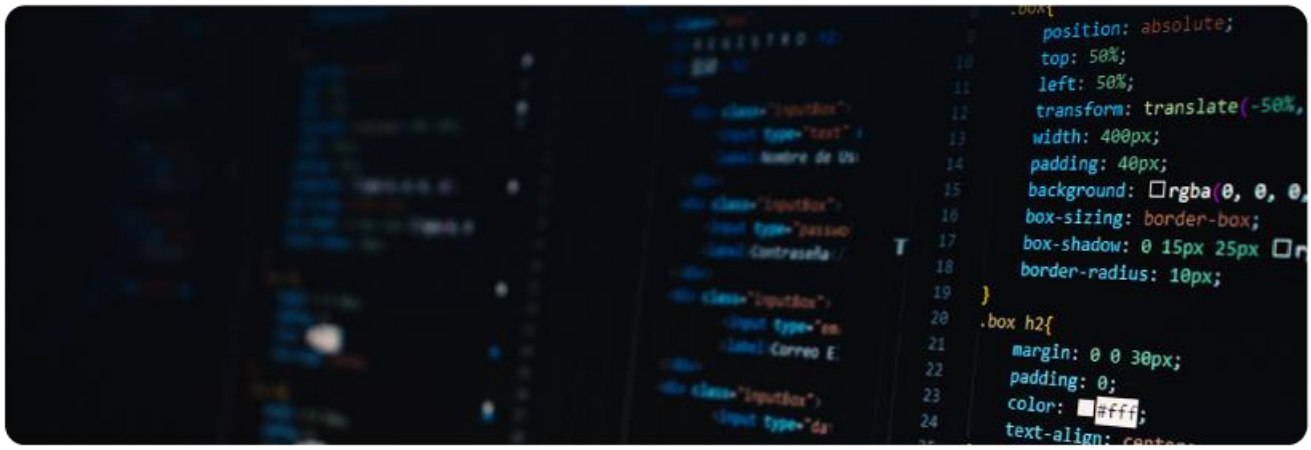
Semana 7

Pruebas de código e interfaz gráfica de usuario

Lectura 1 - Proceso de pruebas de una solución

Convenio de asociación no. 777 de 2021 suscrito entre el Fondo Único de Tecnologías de la Información y las Comunicaciones y la Universidad el Bosque.

Proceso de pruebas de una solución



Las pruebas de software o software testing son parte integral del ciclo de vida del desarrollo de software y hacen referencia a todas aquellas actividades que están orientadas a entregar una información objetiva de la calidad del producto de software desarrollado, brindando seguridad de la funcionalidad, el rendimiento y la experiencia del usuario, e identificando errores que no solo ahorraran tiempo de mantenimiento sino y más importante asegurando que la aplicación de software este revisada y auditada antes de ser ejecutada por el usuario final.

Estas pruebas se orientan en ayudar al programador a explorar, conocer y entender aún más el producto que está desarrollando, buscando igualmente reducir la cantidad de errores que se puedan presentar, y evitando que estos errores sean evidentes o visibles ante el usuario.

Así como el desarrollo de aplicaciones ha ido cambiando con el paso del tiempo, las pruebas de software también han tenido cambios en sus metodologías, donde anteriormente se realizaban de forma manual y solamente una vez se terminaba la codificación del software, pero la automatización y las pruebas orientadas a la prevención han traído consigo pruebas que se hacen de manera automática y en todos los ámbitos de la aplicación.

Este proceso de pruebas ha ganado mucha importancia, tanto que hasta el mismo proceso de codificación también ha sufrido cambios, dando nacimiento a desarrollos orientados a pruebas, creando software testeable como un requisito indispensable para tener un código de calidad.

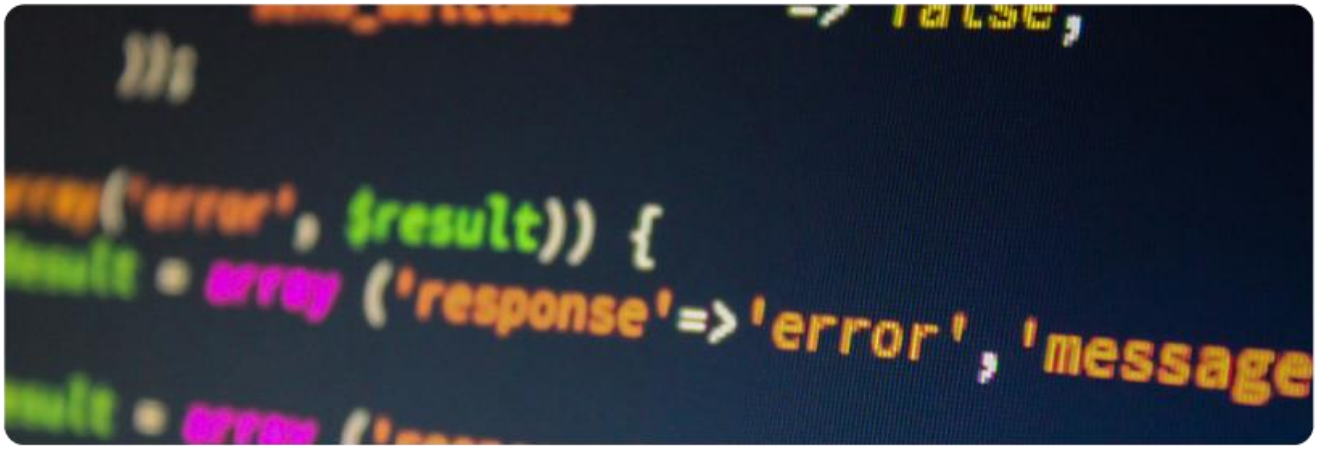
Estas pruebas se convierten por tanto en un elemento crítico que garantiza el correcto funcionamiento de la aplicación, y que tiene objetivos no solo relacionados con la detección de defectos en el software y su corrección, sino que verifica temas como la integración de

componentes y el cumplimiento de requisitos implementados, pero se debe tener claro que se usan para mostrar errores mas no para demostrar su ausencia, teniendo en cuenta también en que momento implementarlas y en qué momento parar de ejecutarlas, haciendo pruebas no solo con el uso de las condiciones de entrada válidas y esperadas sino también para condiciones no válidas e inesperadas.

Pero, así como existen diferentes tipos de aplicaciones de software, y no solo esto sino diferentes aspectos dentro de un mismo desarrollo, así mismo existen diferentes tipos de pruebas que se pueden aplicar y que se ejecutan en diferentes etapas o con distintos objetivos, como lo son pruebas de rendimiento, de escalabilidad, de integración, pruebas unitarias y mucha otras más, que nos ofrecerán una visibilidad integral de la aplicación, desde el código hasta la experiencia del usuario.

Existen pruebas de tipo manual que son las que realiza un usuario paso a paso, o pruebas automáticas que son aquellas que se ejecutan con el uso de otro software. Estas pruebas se pueden ejecutar en distintos niveles creando otra clasificación dependiendo de su implementación pues pueden ir desde las que se ejecutan para probar un módulo especifico hasta pruebas que verifican el sistema completo, por tanto, existe muchos tipos de pruebas como pruebas de sistema, de integración, de rendimiento, de regresión, de aceptación de usuario, entre muchas otras, pero en este ciclo solo hablaremos de las pruebas unitarias.

Pruebas unitarias



Comprueban el correcto funcionamiento de un módulo de código, asegurando que todos los módulos o partes del sistema desarrollado funcionan correctamente por separado. Una prueba

unitaria consiste en un fragmento de código que ejecuta una funcionalidad específica en nuestro código para probar su comportamiento y estado.

Estas consisten en aislar una parte del código y revisar su funcionamiento como un pequeño test que valida el comportamiento de un objeto y su lógica. Suelen ser usadas durante la fase de desarrollo y normalmente es ejecutada por parte del programador, pero también puede ser realizada por el equipo de calidad, y aunque para algunos pueda ser tiempo perdido en realidad puede ayudar a ahorrar tiempo y dinero por errores que solo se podrían ver en fases avanzadas o pruebas del sistema completo.

Estas pruebas unitarias nos ayudan a demostrar que la lógica del código es el correcto y que funcionará en todos los casos, aumentando la legibilidad del código, ayudando además a los desarrolladores a entender más claramente el código base, facilitando el hacer cambios más rápidamente.

Existe un concepto conocido como pruebas unitarias de las tres A´s o A´s unit testing, que consiste en un proceso dividido en tres pasos: Arrange (organizar), Act (actuar), Assert (afirmar). En el primer paso Arrange se definen los requisitos que el código debe cumplir, Act en donde se ejecuta el test de prueba que mostrará los resultados a analizar, y finalmente el paso Assert en donde se comprueba si aquellos resultados que se obtuvieron en el paso anterior son los esperados, en cuyo caso se puede validar y seguir adelante, o de lo contrario se deberán hacer las correcciones que correspondan.

Para que las pruebas unitarias tengan un buen nivel de calidad, lo ideal es que estas sean automatizables, que no necesiten de una intervención manual, esto en especial para tener una integración continua. Otras características importantes son que sean completas cubriendo la mayor cantidad de código posible, independientes para que una ejecución no afecte a otra, Repetibles o Reutilizables para que no sean pruebas que se ejecuten una sola vez, y finalmente Profesionales, estando al nivel del código mismo siendo manejadas con la misma profesionalidad y documentación.

Estas pruebas ofrecen ventajas como la refactorización o el fomento al cambio pues no solo facilitan al programador la detección fácil de errores, sino el cambio de código para mejorar su estructura, al poder hacer pruebas sobre los cambios realizados, asegurando así que los mismos no tengan defectos. Además, simplifican la integración pues al llegar a esta fase se tiene un mayor grado de seguridad del funcionamiento del código. Otra ventaja es que la prueba misma hace parte de la documentación del código puesto que ahí se puede ver cómo usar el programa.

Igualmente se debe tener en claro que las pruebas unitarias por si solas no funcionan ya que, aunque confirmen el correcto funcionamiento de una parte del programa, esto no asegura que

al integrarse con las otras partes el desarrollo funcione, así que hasta la integración no será fácil detectar problemas generales, de rendimiento u otros que afecten el sistema completo.