



Pruebas de Software

Diego Araque - A01026037

Marco Torres - A01025334

Uriel Aguilar - A01781698

25 de Agosto del 2022

1. Realiza una investigación de las siguientes pruebas de software

Pruebas locales.

Las pruebas unitarias consisten en aislar una parte del código y comprobar que funciona a la perfección. Son pequeños tests que validan el comportamiento de un objeto y la lógica, como probar la funcionalidad de una función o una clase.

Pruebas de integración.

Las pruebas de integración dentro del software testing chequean la integración o interfaces entre componentes, interacciones con diferentes partes del sistema, como un sistema operativo, sistema de archivos y hardware o interfaces entre sistemas.

Pruebas alfa.

Por ejemplo, podríamos desarrollar un prototipo de una parte del sistema que no estamos seguros de cómo implementar la interfaz. Si las pruebas alpha devuelven unos comentarios positivos entonces podríamos seguir por esa vía.

Pruebas de regresión.

Son pruebas para que cada vez que exista un cambio en el código, se revise que no se haya generado un cambio negativo en este. Para estas pruebas se implementan, por ejemplo, las pruebas locales o de integración. De esta manera los corremos cada vez que existen cambios en el software, y revisamos si falla o no.

Pruebas dinámicas de validación.

Estas consisten en analizar el comportamiento dinámico del código. Estos ocurren en tiempo real, y suelen implementarse para ver cómo se comporta en el mundo real. Un ejemplo es si te registras en una página y el usuario necesita ingresar la contraseña y después confirmar la contraseña. En tiempo real se le dice al usuario si la contraseña es igual o no a la que se dio previamente.

Pruebas bajo condiciones frontera.

Este tipo de pruebas consiste más bien en una especie de paradigma. Con esto nos referimos a que al hacer pruebas locales, definimos un rango de los valores que vamos a probar. Estos valores estarán en los límites, es decir si se tiene un rango de 10 a 100; se probará el 10, el 100, el 9 y el 101.

2. Investigar las herramientas de pruebas automatizadas existentes actualmente para cada uno de los diferentes tipos de pruebas, de la lista dada arriba (comerciales y open source), identificando para cada una: Nombre, tipo de prueba, licencia (comercial u open source), plataforma y/o lenguajes, particularidades:

Pruebas Locales

Para pruebas locales (unitarias), existen una gran cantidad de librerías que nos permiten realizar estas pruebas.

- MochaJS: Esta librería nos permite hacer pruebas tanto en el backend, como en el frontend. Es totalmente gratis, y se instala desde npm.
- JEST: Esta librería es usada normalmente para aplicaciones de react. This means that it is a javascript library.
- unittest: Este es un framework de python que de igual manera nos permite hacer pruebas unitarias. Cualquiera lo puede usar, para probar su software.

Pruebas de Integración:

- Hound: Es una librería para node o react que permite hacer pruebas automatizadas de integración, puede correr varias sesiones de browsers al mismo tiempo y soporta los browsers más importantes.
- The-book: Es una librería para hacer tests de integración en JavaScript que tiene una experiencia de usuario bastante buena porque es user friendly.

Pruebas de Regresión:

- **Subject7:** Esta es una herramienta “Cloud Based” que tiene la filosofía “true code-less”, lo cuál significa que puedes hacer todas las pruebas de regresión de tu aplicación sin escribir una línea de código, todo se haría desde la interfaz de Subject7
- **Cerberus Testing:** Esta herramienta es open source y soporta aplicaciones web, mobile, APIs, desktops y databases.

Pruebas Dinámicas de Validación:

- **AddressSanitizer:** Esta herramienta es para C y C++, se centra principalmente en encontrar errores de stackoverflow, memory leaks y otros errores de memoria cuando la aplicación se usa por usuarios.
- **IBM Security AppScan:** Esta herramienta sirve para Web y mobile, identifica fallos de seguridad, genera reportes y toma las medidas necesarias para arreglar las vulnerabilidades.

- 3. Seleccionar una herramienta de pruebas y aplicar a un tipo de prueba (de la lista de arriba) a un producto/componente de software, preferentemente en alguna de las aplicaciones móviles que desarrollaste como actividad en el módulo 4:
Desarrollo de aplicaciones móviles. Diseña y documenta el caso de prueba:**

Para esta actividad hicimos 2 tipos de pruebas. Las primeras fueron pruebas unitarias, en las cuales probamos diferentes funciones que nos permiten traducir los valores de tipo mexicano a tipo estadounidense (esta función traduce las fechas, nombres, correos, etc). Además de esto, probamos otra función individual que hace la conversión de calificaciones por separado.

Usamos JEST para probar nuestros componentes de React. Primero que todo, checamos que nuestro main container se renderiza en la página. Para el componente de estudiantes checamos si se renderiza el título que se le pasa. Otro test que hicimos que consideramos muy importante, es que si se sube un archivo checar que si se está subiendo de verdad, de esa manera se pueden evitar bugs que pueden ser difíciles de encontrar.

Referencias:

- Pittet, S. (s.f.). Los distintos tipos de pruebas de software.
<https://www.atlassian.com/es/continuous-delivery/software-testing/types-of-software-testing>
- Threat Stack. (s.f.). Best Testing Tools.
<https://www.threatstack.com/blog/50-best-integration-testing-tools>