

DP2 2021-2022

Knowledge of WIS' testing



Repositorio:

<https://github.com/mpadillatabuenca/Acme-Toolkit.git>

Miembros:

- José Manuel Bejarano Pozo (josbezipoz@alum.us.es)
- Mario Espinosa Rodríguez (maresprod5@alum.us.es)
- Andrea Meca Sánchez (andmecs@alum.us.es)
- Manuel Padilla Tabuenca (maresprod5@alum.us.es)
- Ezequiel Pérez Sosa (ezepersos@alum.us.es)
- Javier Terroba Orozco (javteroro@alum.us.es)

GRUPO G1-E2-05

Versión 1.0.0

25/02/2022

Tabla de contenidos

Tabla de contenidos	1
Historial de versiones	2
Resumen Ejecutivo	2
Introducción	2
Contenido	2
Funcionales	2
Unitarias	3
Integración	3
No funcionales y rendimiento	3
Aspectos finales de las pruebas en relación con distintos frameworks.	3
Conclusión	4
Bibliografía	4

Historial de versiones

Fecha	Versión	Descripción de los cambios
27/02/2022	1.0.0	Creación y finalización del documento.

Resumen Ejecutivo

Dada la brevedad del documento estimo poco acertada la realización de un resumen ejecutivo.

Introducción

En este documento realizado por Ezequiel Pérez Sosa se explica los conocimientos previos del autor sobre las pruebas realizables sobre productos software, previo a la realización de la asignatura. Durante todo el documento se tiene un lenguaje informativo pero todo basado en el conocimiento del autor.

Este conocimiento se basa en el proceso de aprendizaje tanto en la carrera universitaria cómo en el ámbito laboral.

Contenido

Existen variedad de pruebas software y motivos para su realización y ejecución. Las pruebas se realizan dependiendo de la etapa de desarrollo en la que se encuentre el proyecto en cuestión.

Independientemente del lenguaje a utilizar o el framework, podemos diferenciar las siguientes pruebas:

- Funcionales
- Unitarias
- Integración
- No funcionales y rendimiento

Ahora se explicará en que consisten estas pruebas y para qué son utilizadas.

Funcionales

Estas pruebas consisten en comprobar que cierta funcionalidad del producto funciona de forma correcta, sin entrar en detalles cómo la validación del tipo de información que puede ser enviada en un campo. Por ejemplo, supongamos que debemos comprobar que el proceso de registro funciona correctamente. No importan los campos que vamos a introducir (por ejemplo, si un correo electrónico es del formato esperado) sino si al final de dicho

proceso podemos asegurar que el usuario ha podido realizar el registro de forma correcta y no ha ocurrido nada inesperado durante dicho proceso.

Este tipo de pruebas no intenta comprobar un sistema completo sino un módulo de esta.

Unitarias

Estas pruebas consisten en comprobar los elementos o piezas de código más pequeñas del producto. Difieren de las pruebas funcionales en que comprobamos funciones concretas o líneas y no procesos completos. Por ejemplo, imaginemos que tenemos una función que debe sacar dos posibles salidas dependiendo de la entrada. Podemos construir pruebas que dadas unas entradas concretas compruebe la salida esperada.

Integración

Estas pruebas consisten en comprobar el correcto funcionamiento de distintos módulos de la aplicación y cómo estos se complementan. Comprobamos así con estas pruebas que cuando un usuario utilice cierto módulo de la aplicación, esta se comuniquen de forma correcta con el resto de la aplicación. Por ejemplo, si un usuario inicia sesión, el resto de los módulos de la aplicación deberían ejecutarse en función de la respuesta del módulo de inicio de sesión.

No funcionales y rendimiento

Las pruebas no funcionales consisten en comprobar ciertos aspectos de la aplicación, cómo por ejemplo la seguridad del sistema o la fiabilidad de este. Haciendo uso de hacking ético y atacando la aplicación pueden encontrarse vulnerabilidades en el sistema. Este tipo de pruebas se realizan en una fase avanzada del proyecto.

Las pruebas de rendimiento son pruebas donde se miden ciertos aspectos variados, cómo la concurrencia durante una operación que se ejecuta a la misma vez que otra que acceden a los mismo recursos o pruebas de carga donde una gran cantidad de usuarios realizan una operación.

Aspectos finales de las pruebas en relación con distintos frameworks.

Los frameworks que ha utilizado el autor de este documento donde se han realizado pruebas son los siguientes:

- Spring Boot (Uso de librerías de Junit)
- Django

En vista de que en la asignatura utilizaremos el lenguaje Java nos centraremos en los conocimientos sobre Junit.

Con respecto a Junit el autor de este documento lo ha utilizado en varios proyectos. Con estas librerías se han realizado pruebas parametrizadas para realizar pruebas con distintas entradas. Durante toda la carrera universitaria se han adquirido conocimientos (aunque no en profundidad) sobre todos los tipos de prueba en el lenguaje de programación JAVA.

Conclusión

Al final de la realización de este documento, podemos concluir que los conocimientos son repartidos en el ámbito de la realización y ejecución de pruebas. Conocimiento que se espera profundizar y pulir en la realización de la asignatura.

Bibliografía

1. Apuntes de la asignatura de Diseño y Pruebas 1 (ev.us.es)