

WIS Testing Report



Jesús Martín Vergara (jesmarver@alum.us.es)
Álvaro Navarro Rivera (alvnavriv@alum.us.es)
Mario Sánchez Naranjo (marsannar2@alum.us.es)
Gabriel Vacaro Goytia(gabvacgoy@alum.us.es)
Ignacio Warleta Murcia (ignwarmur@alum.us.es)

Diseño y Pruebas II
Grupo C1.02.01

<https://github.com/DP2-L1/ACME-L3-D01.git>

17/02/2023

Tabla de contenido

<i>Sumario</i>	2
<i>Historial de versiones</i>	3
<i>Introducción</i>	4
<i>Contenido</i>	5
<i>Requisito 1</i>	¡Error! Marcador no definido.
<i>Conclusiones</i>	6
<i>Bibliografía</i>	6

Sumario

Todo ingeniero de software debe entender que el testing es esencial para garantizar que se cumplan los requisitos de nuestro producto en un sistema de información web, que se documenten los resultados y se sigan buenas prácticas sobre el diseño de pruebas.

Según las capas de arquitectura con la que trabajemos (lógica de negocio, presentación o modelo de datos) y las tecnologías que usemos (frameworks, librerías, plugins,...) , podemos hacer uso de una gran variedad de herramientas y técnicas que nos ayudarán a pulir nuestro sistema a través de las pruebas.

Historial de versiones

Fecha	Versión	Descripción	Sprint
17/02/2023	1.0	Escritura del reporte	1

Introducción

En este reporte trataremos los conocimientos adquiridos sobre el testing en un sistema de información web (WIS).

Para abordar este tema, primero tenemos que entender la importancia del testing de un WIS, donde existen factores que influyen sobre este y en los que necesitamos profundizar con el fin de aprender buenas prácticas y metodologías. Posteriormente, veremos los tipos distintos de pruebas que se realizan sobre un sistema de información, cada uno con su función y propósito, para luego concluir con una breve conclusión sobre lo discutido en este reporte.

Contenido

Antes de empezar, cabe destacar que existen algunos conceptos que se suelen utilizar entre los profesionales que nos ayudarán a alcanzar un nivel mayor de comprensión del testing, como puede ser el sujeto bajo prueba (SUT), el objeto que se va a probar, el fake (objetos reales que se usan en el programa en ejecución pero simulan su comportamiento), el stub (un fake que devuelve una salida especificada por los testers) y los spies (un fake que recopila información sobre la ejecución de la prueba).

Para poder desarrollar buenas pruebas, tenemos que entender que una prueba está caracterizada por su nivel de granularidad y la validación de distintos aspectos del sistema. Una prueba puede validar distintas características de un sistema de información, como puede ser:

- Seguridad: verifica que la aplicación sea segura y protegida contra vulnerabilidades comunes de seguridad.
- Rendimiento: se evalúa la capacidad de respuesta y la velocidad de carga del sitio web, así como la capacidad de procesamiento y almacenamiento de la base de datos.
- Usabilidad: se verifica la facilidad de uso y navegación de la interfaz de usuario.
- Funcionalidad: se asegura de que todas las funciones y características de la aplicación estén funcionando como se espera.
- Interoperabilidad: se comprueba que la aplicación sea compatible con diferentes navegadores web, dispositivos y sistemas operativos.

Por último, tenemos que determinar la granularidad, es decir, el nivel de detalle que queremos cubrir con la prueba. La granularidad se puede dividir en varios niveles, desde pruebas de muy alto nivel que evalúan la funcionalidad general del sistema, hasta pruebas de muy bajo nivel que se enfocan en componentes individuales del sistema. Por lo general podemos clasificarlo en varios tipos:

- Pruebas Unitarias: pruebas a nivel de componente individual (métodos, funciones).
- Pruebas de integración: se enfocan en la interacción y compatibilidad entre los diferentes componentes del sistema.
- Pruebas de extremo a extremo: se prueba el sistema como un todo.
- Pruebas de Aceptación: evalúan si la aplicación cumple con los requisitos del usuario.
- Pruebas de exploración: buscan evaluar nuevas posibilidades.

Conclusiones

El testing o pruebas en un sistema de información web es un aspecto fundamental para garantizar su correcto funcionamiento y calidad. A través del testing se pueden identificar y corregir errores, evaluar la usabilidad, seguridad y rendimiento del sistema, entre otros aspectos.

Es importante que el testing se realice de forma sistemática y rigurosa, y que se documenten los resultados obtenidos para poder llevar a cabo una evaluación completa del sistema. De esta manera, se pueden identificar posibles problemas y corregirlos antes de que el sistema sea puesto en producción, lo que reducirá los costos y riesgos asociados con el mantenimiento y la corrección de errores en el futuro.

Bibliografía

Intencionalmente en blanco.