Grupo: C1.064

20/02/2025

Repositorio: <https://github.com/paugonpae67/Acme-ANS-D01>

Integrantes: Paula Rosa González Páez ([paugonpae@alum.us.es](mailto:paugonpae@alum.us.es)), Lidia Ning Fernández Casillas ([lidfercas@alum.us.es](mailto:lidfercas@alum.us.es)), María Auxiliadora Quintana Fernández ([marquifer@alum.us.es](mailto:marquifer@alum.us.es)), Fernando Partal García ([ferpargar@alum.us.es](mailto:ferpargar@alum.us.es)), Iratxe Parra Moreno ([iraparmor@alum.us.es](mailto:iraparmor@alum.us.es)).

TESTING OF A WIS REPORT

acme ans – c1.064

|  |
| --- |
| TABLA DE CONTENIDOS |

[TABLA DE CONTENIDOS 1](#_Toc1813262021)

[RESUMEN EJECUTIVO 2](#_Toc1459581904)

[TABLA DE REVISIONES 3](#_Toc2111742646)

[INTRODUCCIÓN 4](#_Toc1767302031)

[CONTENIDO 4](#_Toc1382698068)

[IMPORTANCIA DE LAS PRUEBAS 5](#_Toc1166859189)

[Tipos de Pruebas y su Aplicación 5](#_Toc1554929408)

[Herramientas Utilizadas 6](#_Toc71773101)

[Documentación en el Entorno de Testing 6](#_Toc1136015694)

[Conclusión 7](#_Toc770383916)

[Bibliografía 8](#_Toc1354139355)

|  |
| --- |
| RESUMEN EJECUTIVO |

Este documento presenta los conocimientos previos a cursar la asignatura Diseño y Pruebas 2 en relación con las pruebas de los Sistemas de Información Web (WIS). Se detalla la importancia de las pruebas en el desarrollo de software, abordando conceptos básicos como la verificación y validación de software, el objetivo de detectar errores y garantizar la calidad del producto final.

Además, se describe el conocimiento previo sobre la estructura y organización de las pruebas, incluyendo pruebas unitarias, de integración y de sistema.

En el ámbito tecnológico, se identifican herramientas utilizadas previamente como JUnit para pruebas unitarias en Java, Mockito para test doubles o Spring Boot Test para la automatización de pruebas en aplicaciones web. Por último, también se explora el uso de herramientas de integración continua como GitHub Actions para la automatización de pruebas aunque nuestros conocimientos al respecto sean reducidos.

Este informe representa un punto de partida para evaluar el progreso en nuestra comprensión sobre las pruebas de software. A medida que avancemos en la materia, esperamos consolidar un enfoque más estructurado, integrando herramientas avanzadas y metodologías efectivas para la gestión de calidad en el desarrollo de aplicaciones web.

|  |
| --- |
| TABLA DE REVISIONES |

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSIÓN** | **FECHA** | **MIEMBRO DEL EQUIPO** | **DESCRIPCIÓN** |
| 1.0 | 20/02/2025 | Fernando Partal García | Redacción del documento. |

|  |  |  |
| --- | --- | --- |
| INTRODUCCIÓN |  |  |

Las pruebas de software son un pilar fundamental en el desarrollo de aplicaciones, ya que permiten garantizar la calidad, fiabilidad y correcto funcionamiento del sistema. En el contexto de los Sistemas de Información Web (WIS), las pruebas adquieren especial relevancia debido a la complejidad de su arquitectura y la necesidad de asegurar la interacción adecuada entre sus distintos componentes.

La estructura del informe sigue un enfoque progresivo: en primer lugar, se expone la importancia de las pruebas en el desarrollo de software, analizando cómo estas influyen en la calidad del producto final y en la eficiencia del proceso de desarrollo. Posteriormente, se detallan los diferentes tipos de pruebas y su aplicación en un WIS, destacando las pruebas unitarias, de integración y de sistema, así como su rol en la detección y corrección de errores.

También se examinan las herramientas que previamente hemos utilizado en el proceso de pruebas, incluyendo JUnit, Mockito y Spring Boot Test para la validación de servicios, así como React Testing Library para pruebas en el frontend. Se aborda además la automatización e integración continua mediante GitHub Actions.

Finalmente, se analiza la importancia de la documentación en el entorno de testing, con énfasis en la planificación y estructuración de las pruebas a través del Plan de Pruebas, un documento clave para la trazabilidad y organización del proceso de validación en el desarrollo de un WIS.

|  |  |  |
| --- | --- | --- |
| CONTENIDO |  |  |

### IMPORTANCIA DE LAS PRUEBAS

El desarrollo de software requiere mecanismos de validación y verificación para garantizar su correcto funcionamiento. Gracias a la asignatura Diseño y Pruebas 1, comprendemos que las pruebas juegan un papel esencial en la detección de errores y en la garantía de calidad del producto final.

Las pruebas de software permiten asegurar que una aplicación cumpla con los requisitos funcionales y no funcionales establecidos, reduciendo el riesgo de fallos en producción. Además, ayudan a identificar posibles errores en las primeras etapas del desarrollo, disminuyendo los costos asociados a correcciones tardías.

Otro aspecto fundamental de las pruebas es su papel en el mantenimiento y la evolución del software. A medida que una aplicación crece y se implementan nuevas funcionalidades, las pruebas automatizadas facilitan la detección de regresiones, asegurando que los cambios no afecten el comportamiento previo del sistema. Esto es particularmente relevante en entornos ágiles y de integración continua, donde las actualizaciones se despliegan de manera frecuente, facilitando la entrega continua de software confiable.

### Tipos de Pruebas y su Aplicación

En la asignatura Diseño y Pruebas 1 adquirimos una comprensión sólida sobre los distintos tipos de pruebas y su aplicación en el desarrollo de software:

Las pruebas unitarias se enfocan en la verificación de componentes individuales del software, asegurando que cada unidad de código funcione correctamente de manera aislada. Utilizamos herramientas como JUnit para Java, lo que nos ha permitido validar métodos y clases con rapidez y precisión.

Las pruebas de integración garantizan que los diferentes módulos del sistema interactúen de manera correcta. En este nivel de prueba, hemos aprendido a simular dependencias y evaluar la comunicación entre componentes, identificando posibles fallos en la integración de servicios o acceso a bases de datos.

Las pruebas de sistema verifican el comportamiento global de la aplicación, evaluando su funcionalidad completa en un entorno similar al de producción. Entendemos que este tipo de prueba es crucial para detectar problemas que puedan surgir en la interacción entre diferentes partes del sistema y hemos visto su aplicación en el contexto de arquitecturas MVC.

Todo esto fue puesto en práctica en el desarrollo de un WIS de un videojuego, aplicando pruebas unitarias, de integración y de sistema para garantizar su correcto funcionamiento. Además, realizamos pruebas en el frontend utilizando React Testing Library, lo que nos permitió validar la interacción y el comportamiento de los componentes de la interfaz de usuario, asegurando su correcta integración con la lógica de negocio.

### Herramientas Utilizadas

Previamente, hemos trabajado con diversas herramientas para la automatización de pruebas en aplicaciones web. Utilizamos JUnit para la ejecución de pruebas unitarias en Java, así como Mockito para la creación de test doubles y simulación de dependencias. Además, empleamos Spring Boot Test para la validación de servicios en aplicaciones desarrolladas con Spring.

También tenemos experiencia en la ejecución de pruebas en interfaces web utilizando React Testing Library, lo que nos ha permitido evaluar la funcionalidad y comportamiento de los componentes de nuestra aplicación.

### Documentación en el Entorno de Testing

En el marco del desarrollo de pruebas para un WIS, tenemos experiencia en el desarrollo de un Plan de Pruebas, un documento esencial para la gestión del proceso de pruebas. En él se incluyen aspectos clave como la estrategia de pruebas, la cobertura de pruebas, la matriz de trazabilidad y la planificación de pruebas. Este documento permitió estructurar y organizar la ejecución de las pruebas de manera eficiente, asegurando que cada caso de prueba esté correctamente definido y alineado con los requisitos del sistema.

|  |  |  |
| --- | --- | --- |
| Conclusión |  |  |

El conocimiento previo que poseíamos sobre pruebas en un WIS se basaba en nociones adquiridas en la asignatura Diseño y Pruebas 1, donde aprendimos sobre pruebas unitarias, de integración y de sistema con herramientas como JUnit, Mockito y Spring Boot Test.

Durante el proyecto de la asignatura, aplicamos diferentes estrategias de prueba, logrando un 70% de cobertura del código con pruebas automatizadas. Además, utilizamos React Testing Library para validar la interfaz de usuario e implementamos GitHub Actions para la integración continua, asegurando estabilidad y detección temprana de errores.

Asimismo, documentamos el proceso en el Plan de Pruebas, lo que nos permitió organizar y estructurar la ejecución de pruebas de manera eficiente. Con esta experiencia, hemos fortalecido nuestras competencias en testing, mejorando la calidad y fiabilidad del software.

|  |  |  |
| --- | --- | --- |
| Bibliografía |  |  |

Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla (LSI) (2025) – *Diseño y Pruebas 1.*