Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**WIS Testing Report**



Grado en Ingeniería Informática – Ingeniería del Software  
Diseño y Pruebas 2.

Curso 2024 – 2025

| **Grupo de prácticas: C1.004** | | |
| --- | --- | --- |
| **Autores por orden alfabético** | **Rol** | **Descripción del rol** |
| Cortés Carrasco, Claudio - 77929527 | Developer | Persona encargada de desarrollar el código. |
| Delgado Pallares, David - 29519510E | Tester | Persona encargada de realizar pruebas sobre el código. |
| Fernández Román, Santiago- 30276668C | Project Manager | Persona encargada de tomar decisiones de diseño y vigilar el correcto desarrollo |
| García León, Guillermo -  17486052A | Developer | Persona encargada de desarrollar el código. |

**Control de Versiones**

| **Fecha** | **Versión** | **Descripción** |
| --- | --- | --- |
| 20/02/2025 | v1.0.0 | Desarrollo de la primera versión. |
|  |  |  |
|  |  |  |
|  |  |  |

**Índice de contenido**

[**1. Resumen ejecutivo**](#_wkdgh14d2jlq) **4**

[**2. Introducción**](#_14h6bglc6s60) **5**

[**3. Contenidos**](#_icebub40rzkm) **6**

[2.1. Testing en el Backend con Spring Boot](#_2oh3sz1cnhwe) 6

[2.2. Testing en el Frontend con React](#_bf6zwu2sdqmy) 6

[**4. Conclusiones**](#_471n1tejp1g1) **7**

[**5. Bibliografía**](#_syrhz6ozjbez) **8**

## Resumen ejecutivo

Este documento presenta un resumen de las estrategias de testing en Web Information Systems (WIS), abarcando los conceptos clave aprendidos en la asignatura Diseño y Pruebas 1 (DP1). Se detalla la implementación de pruebas en un sistema basado en la arquitectura Spring Boot + React, explorando cómo estas tecnologías se combinan para garantizar la calidad y fiabilidad de aplicaciones web modernas.

Asimismo, este documento ha sido elaborado para garantizar el cumplimiento del Requisito 22 del proyecto, asegurando que la información presentada esté alineada con los criterios establecidos. Se han estructurado los contenidos de manera clara y concisa, proporcionando una visión general efectiva del testing en WIS y su aplicación en el desarrollo de software.

## Introducción

El presente informe tiene como objetivo recopilar y estructurar los conocimientos adquiridos sobre las estrategias de testing en Web Information Systems (WIS) en el marco de la asignatura Diseño y Pruebas 1 (DP1). A lo largo del curso, hemos trabajado en la implementación de pruebas en un sistema basado en la arquitectura Spring Boot + React, explorando cómo estas tecnologías se combinan para garantizar la calidad y fiabilidad de aplicaciones web modernas.

En este documento, se detallarán los conceptos clave abordados, desde las pruebas en el backend con Spring Boot, incluyendo pruebas unitarias, de integración y de aceptación, hasta las pruebas en el frontend con React, destacando el uso de herramientas para testing de componentes y pruebas end-to-end.

Durante el curso DP1, se utilizaron herramientas como Mockito, Spring Test y @DataJpaTest para la validación de la lógica del backend, asegurando el correcto funcionamiento de los servicios y la persistencia de datos.

Este reporte servirá como una síntesis del aprendizaje adquirido y como referencia para el desarrollo futuro de aplicaciones basadas en WIS con buenas prácticas de testing.

## Contenidos

El testing en una aplicación basada en la arquitectura Spring Boot y React se divide en pruebas para el backend y el frontend, asegurando el correcto funcionamiento de cada capa.

#### 2.1. Testing en el Backend con Spring Boot

Spring Boot ofrece diversas herramientas para realizar pruebas automatizadas:

* **Pruebas Unitarias:** Se utilizan JUnit y Mockito para probar componentes individuales como servicios y controladores.
* **Pruebas de Integración:** Se usan para verificar la interacción entre diferentes capas del sistema, utilizando Spring Test y bases de datos en memoria como H2.
* **Pruebas de Persistencia:** Se emplea @DataJpaTest para probar la correcta funcionalidad de los repositorios JPA y su integración con la base de datos.
* **Pruebas de Aceptación:** Se implementan con herramientas como Cucumber para verificar el cumplimiento de requisitos funcionales.

Spring Boot también permite la simulación de llamadas a APIs REST con MockMvc, lo que facilita la validación de endpoints antes de la integración con el frontend.

#### 2.2. Testing en el Frontend con React

React cuenta con diversas bibliotecas para realizar pruebas de componentes y flujos de usuario:

* **Testing de Componentes:** Se emplea Jest junto con React Testing Library para validar la renderización y el comportamiento de los componentes.
* **Pruebas de Interfaz de Usuario (UI):** Se simulan eventos con herramientas como Cypress o Playwright para probar la interacción del usuario con la aplicación.
* **Pruebas End-to-End (E2E):** Se ejecutan pruebas automatizadas que recorren la aplicación completa, verificando la integración con el backend.

Para garantizar la estabilidad del código, es recomendable integrar estas pruebas en pipelines de integración continua (CI/CD) utilizando herramientas como GitHub Actions o Jenkins.

## Conclusiones

El testing en Web Information Systems con Spring Boot y React es fundamental para garantizar la calidad, estabilidad y seguridad de las aplicaciones web modernas. A lo largo de este curso, hemos aprendido a estructurar un conjunto de pruebas para cada capa del sistema, utilizando herramientas especializadas para cada tipo de validación.

El uso de Mockito, Spring Test y @DataJpaTest en DP1 ha permitido una mayor comprensión de las estrategias de prueba en el backend, asegurando una correcta validación de la lógica de negocio y la persistencia de datos.

Este informe sintetiza los conocimientos adquiridos en DP1 y servirá como referencia para la implementación de pruebas en futuros proyectos de desarrollo web con estas tecnologías.

## Bibliografía

Apuntes de la asignatura Diseño y Pruebas 1 (DP1) contenidos en la plataforma Enseñanza Virtual.