## 

## 

## **DP2-Report on Prior Knowledge in Web Information System (WIS) Testing**

## 

**Group:** C1.027  
**Repository:** <https://github.com/DP2-C1-027/AirNav-Logistics>  
**Group Members:** Garcia de Tejada Delgado, Jose; Peñaloza Friqui, Nora; Niza Cobo, Manuel Jesús; Gomez Claraco, Nicolas; Campos Diez, Lucia.

**Corporate Emails:** [josgardel8@alum.us.es](mailto:josgardel8@alum.us.es), [norpennfri@alum.us.es](mailto:norpennfri@alum.us.es) , [mannizcob@alum.us.es](mailto:mannizcob@alum.us.es) , [nicgomcla@alum.us.es](mailto:nicgomcla@alum.us.es) , [luccamdie@alum.us.es](mailto:luccamdie@alum.us.es)   
**Date:** 02/15/2025

**Table of Contents**

1. Executive Summary
2. Revision Table
3. Introduction
4. Contents
5. Conclusions
6. Bibliography

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## **Executive Summary**

This document describes the prior knowledge regarding testing in a Web Information System (WIS) before taking this course. It details general concepts, types of tests, known tools and techniques, as well as limitations in previous knowledge. Specifically, it addresses the initial awareness of unit, integration, and functional testing, highlighting the primary focus on backend unit tests. Additionally, it mentions familiarity with tools such as JUnit and Mockito, albeit with limited depth of application. Furthermore, the document identifies gaps in knowledge regarding advanced testing strategies, including end-to-end and security testing, as well as best practices for ensuring software quality in a structured manner.

Moreover, it highlights the lack of hands-on experience with automated testing frameworks and continuous integration pipelines, which are essential for modern software development. The understanding of performance and load testing was also minimal, with no prior exposure to tools like JMeter or Gatling. The document further emphasizes the need for a more in-depth comprehension of test-driven development (TDD) and behavior-driven development (BDD) methodologies to enhance software reliability and maintainability.

## 

## **Revision Table**

|  |  |  |
| --- | --- | --- |
| **Revision Number** | **Date** | **Description** |
| 1.0 | 02/15/2025 | Initial version of the document |

## 

## 

## 

## 

## **Introduction**

Before taking this course, knowledge regarding testing in a Web Information System (WIS) was limited to basic software testing concepts. There was a general understanding of the importance of verifying the correct functionality of an application, but there was no structured approach or specific tools for conducting tests effectively.

This document is structured as follows: the first section contains the executive summary, followed by a revision table. Then, the contents section details prior knowledge in software testing. Finally, conclusions and bibliography are presented.

## **Content**

### **1. Basic Testing Concepts**

* It was understood that software testing is essential to ensure product quality.
* The difference between manual and automated testing was known.
* It was recognized that testing helps detect errors and improve system stability.

### **2. Types of Tests**

* There was basic knowledge of unit, integration, and functional tests, although without a deep understanding of their implementation.
* It was known that unit tests verify small code fragments, while integration tests validate interactions between modules.

### **3. Web Application Testing**

* It was recognized that web applications require specific tests to evaluate their behavior across different browsers and devices.
* The importance of testing security and scalability was understood.
* Awareness of tools such as Selenium for automating interface testing existed, though they had not been used in depth.

### **4. Frameworks and Tools**

* The use of JUnit for unit testing in Java was known.
* Awareness of Mockito for creating mock objects existed, but without deep exploration of its application.
* It was understood that Spring Boot includes mechanisms for testing, but there was no detailed knowledge of their use.
* Most of the testing approach focused on unit testing in the backend.
* There was limited exposure to automated testing frameworks and continuous integration pipelines, which are crucial in modern software development.
* Performance and load testing knowledge was minimal, with no prior experience using tools like JMeter or Gatling.
* There was a need for a deeper understanding of test-driven development (TDD) and behavior-driven development (BDD) methodologies to enhance software reliability and maintainability.

## **Conclusions**

Before taking this course, knowledge about testing in a Web Information System was basic and lacked a systematic approach. While the importance of testing and some fundamental concepts were understood, advanced techniques and specific tools for effectively implementing them in a professional environment were largely unknown. This course has been key to expanding knowledge and strengthening skills in conducting structured and efficient software tests.

## **Bibliography**

Intentionally blank.