TESTING OF WEB INFORMATIONS SYSTEM

(WIS)



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

|  |  |  |  |
| --- | --- | --- | --- |
| Date | **Version** |  | **Description** |
| 20/02/2025 | 1.0 |  | Initial version of the Testing of WIS report |

Contents

[1. Introduction 1](#_Toc190957312)

[2. Known concepts about software testing 1](#_Toc190957313)

[3. Tools 2](#_Toc190957314)

[4. Conclusion 2](#_Toc190957315)

Group C1.056

## Introduction

A WIS (Web Information System) is an information system that uses web technologies to provide information and services to users or other applications or systems.

Testing a WIS is important because a system failure can have serious consequences for the client. While this is not always the case, it is essential to maintain the client’s reputation and deliver a satisfactory product that meets expectations and is free of obvious errors. Testing is conducted to detect these errors.

## Known concepts about software testing

The tests we have learned before are mainly centered around executing code with certain inputs to verify that the software behaves as expected.

Each test has something called the SUT (subject under test), that is the part of the software that we are focusing on testing.

A battery of tests is called test suite, it’s made of different classes that contains testing methods. Tests can be divided by their granularity, going from unitary tests (the most basic ones) to the exploratory ones.

Tests can be sociable (they will directly talk to collaborators) or solitary(they will replace all collaborators with doubles and avoid side-effects).

Mainly we have worked with unitary tests. We divide them into 3 parts: Arrange Act y Assert. Also, we have implemented both positive and negative test cases. Also, we know how to get a better coverture depending on if in the code there are sequences, conditionals, loops, or if the data has ranges, optionals, or collections.

Finally, we have seen some concepts about security and the main vulnerabilities like SQL injections for example and how to avoid them.

## Tools

The unit testing frameworks we have used are JUnit and Mockito. We have implemented tests by using assertions and test doubles (stubs, mocks, and fakes).

Swagger has also been quite useful sometimes for http methods.

## Conclusion

We have a solid base knowledge about testing. We have been introduced to the main topics, and we are able to prepare some basic tests if needed. However, we are not familiar with different tools, but we think with the already owned knowledge, it will be much easier to learn them.