FUNCTIONAL AND NON-FUNCTIONAL TESTING

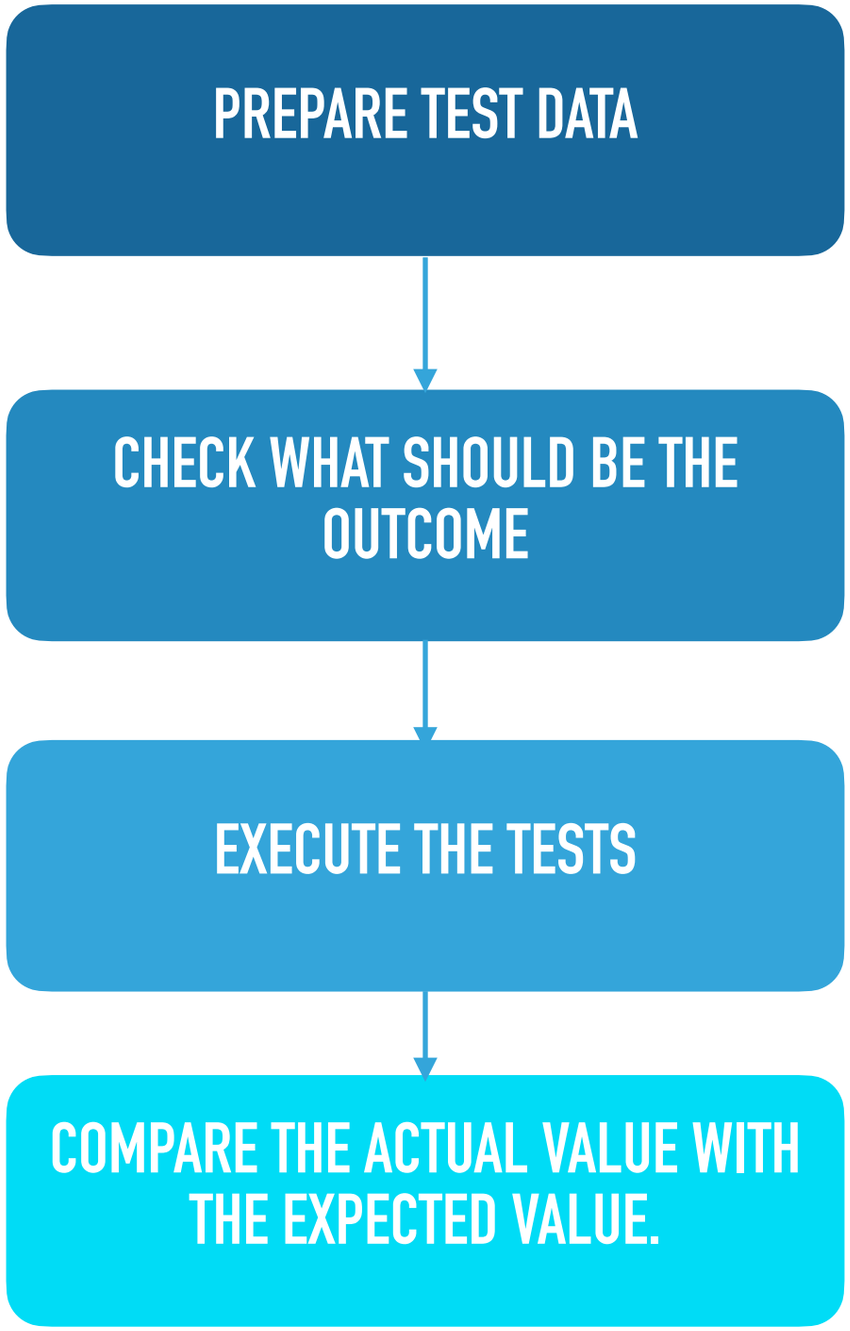
FUNCTIONAL TESTING

FOR YOU, WHAT IS FUNCTIONAL TESTING?

FUNCTIONAL TESTING

- Verifies each function of the software.
- Project Requirements.
- Simulate actual system usage.
- Automated or manual?

Steps to do functional testing:



FUNCTIONAL TESTING

- Types of functional testing:
  - Unit Testing
  - Smoke Testing
  - Sanity Testing
  - Integration Testing
  - System Testing
  - Acceptance Testing
    - UAT
    - Alpha Testing
    - Beta Testing
    - Contract and regulatory testing
  - Negative Testing

Unit Testing

- Basic level of testing.
- Smaller building blocks of a program or system.
- Automated.
- Developers.

EXAMPLE

```java
/**
 * Created by josea.wolff on 1/19/19.
 */
public class ExampleTest {

    public String getTextInUpperCase(String text){
        return text.toUpperCase();
    }

    public String getTextInLowerCase(String text){
        return text.toLowerCase();
    }
}
```

```java
import Example.ExampleTest;
import org.testng.Assert;
import org.testng.annotations.Test;

/**
 * Created by josea.wolff on 1/19/19.
 */
public class Tests {

    @Test
    public void upperCaseTest(){
        System.out.println("\n########################## upperCaseTest ####################################");
        String expectedResult = "INTRO TO DEVOPS";
        ExampleTest ex =  new ExampleTest();
        String actualResult = ex.getTextInUpperCase("Intro to devops");
        System.out.println("Expected Result: " + expectedResult + "\nActualResult: " + actualResult);
        System.out.println("##########################################################################");
        Assert.assertEquals(actualResult,expectedResult);
    }

    @Test
    public void lowerCaseTest(){
        System.out.println("\n########################## lowerCaseTest ####################################");
        String expectedResult = "intro to devops";
        ExampleTest ex =  new ExampleTest();
        String actualResult = ex.getTextInLowerCase("Intro to devops");
        System.out.println("Expected Result: " + expectedResult + "\nActualResult: " + actualResult);
        System.out.println("##########################################################################");
        Assert.assertEquals(actualResult,expectedResult);
    }
}
```

EXAMPLE

```
Example                                                    9ms
  <no name>                                                9ms
    Tests                                                  9ms
      lowerCaseTest                                        9ms
      upperCaseTest                                        0ms
```

```
All 2 tests passed – 9ms
/Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java ...
objc[32713]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMach
  .jdk/Contents/Home/bin/java (0x106cc24c0) and /Library/Java/JavaVirtualMachines/jdk1.8
  .jdk/Contents/Home/jre/lib/libinstrument.dylib (0x106d4e4e0). One of the two will be u

########################### lowerCaseTest ####################################
Expected Result: intro to devops
ActualResult: intro to devops
##############################################################################

########################### upperCaseTest ####################################
Expected Result: INTRO TO DEVOPS
ActualResult: INTRO TO DEVOPS
##############################################################################

===============================================
Example
Total tests run: 2, Failures: 0, Skips: 0
===============================================

Process finished with exit code 0
```

EXAMPLE

```
/**
 * Created by josea.wolff on 1/19/19.
 */
public class ExampleTest {

    public String getTextInUpperCase(String text){
        return text.toUpperCase();
    }

    public String getTextInLowerCase(String text){
        return text.toUpperCase();
    }
}
```

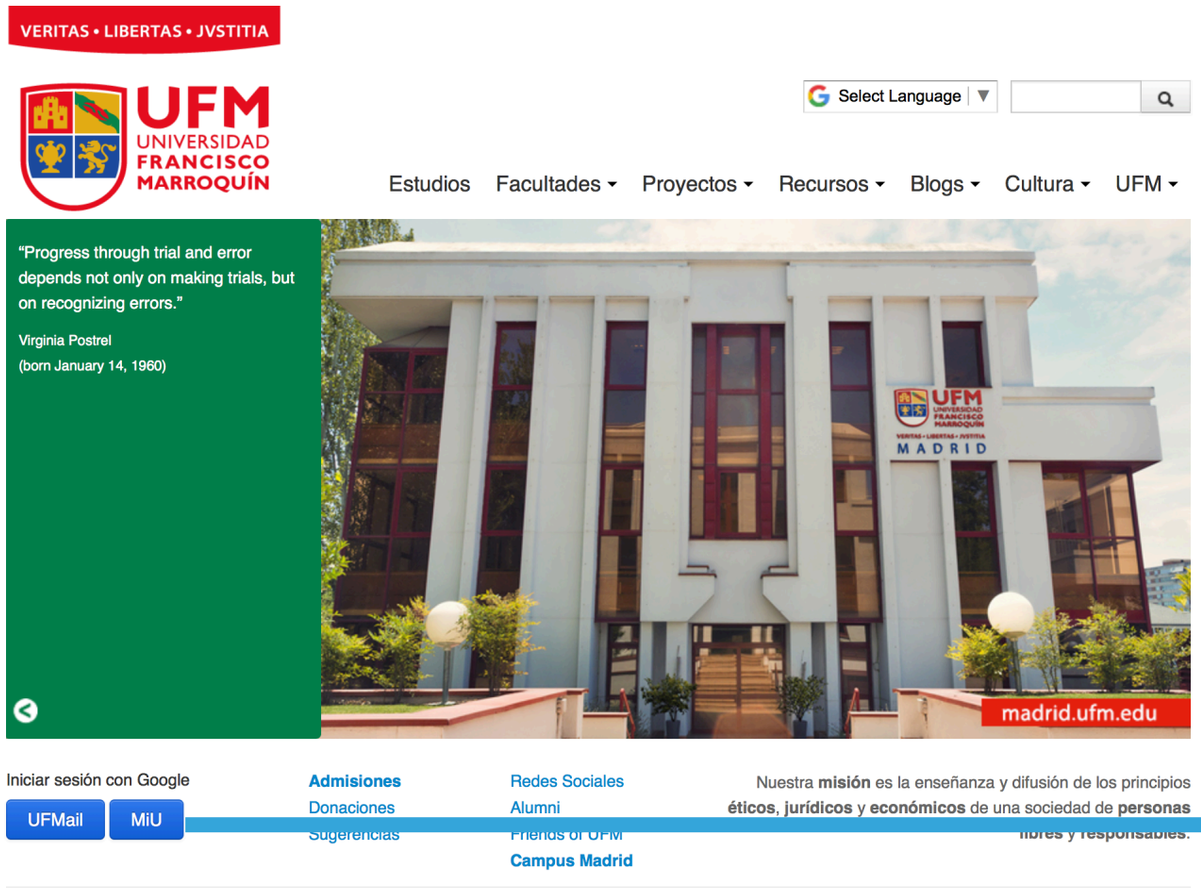
```
Example                                  10ms
  <no name>                              10ms
    Tests                                10ms
      lowerCaseTest                      10ms
      upperCaseTest                       0ms
```

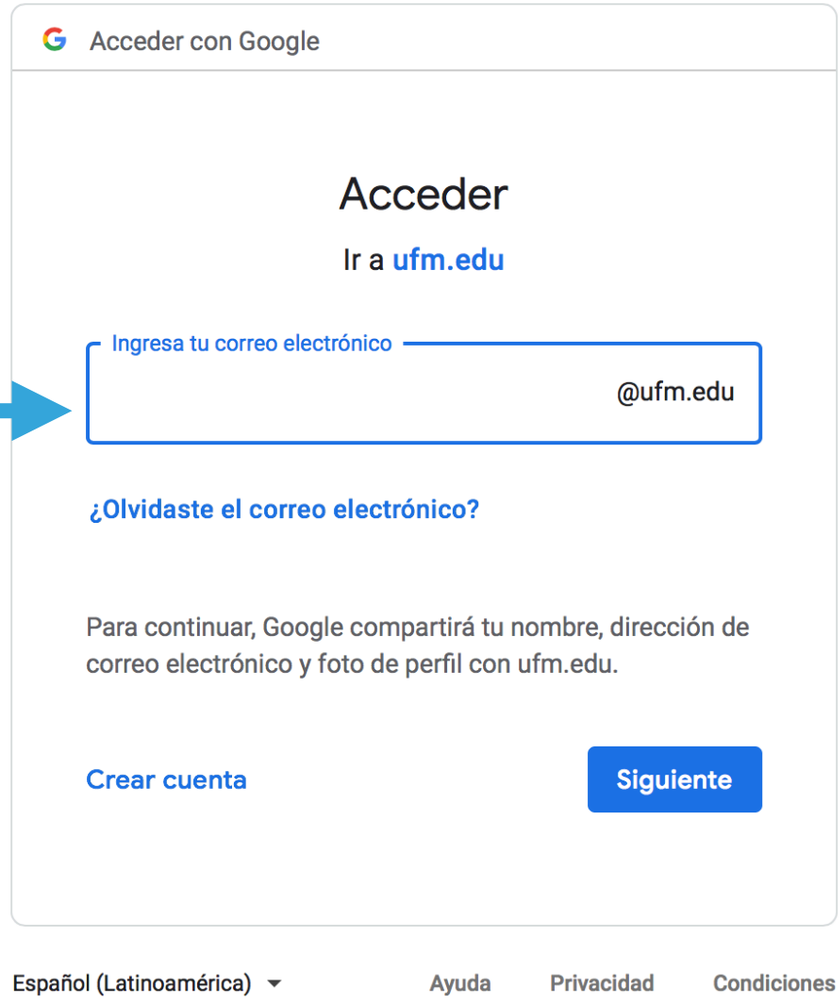
```
/Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/ja
objc[32816]: Class JavaLaunchHelper is implemented in both /Library/Jav
  and /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/
  one is undefined.

############################ lowerCaseTest ############################
Expected Result: intro to devops
ActualResult: INTRO TO DEVOPS
###################################################################################

java.lang.AssertionError: expected [intro to devops] but found [INTRO TO DEVOPS]
Expected :intro to devops
Actual   :INTRO TO DEVOPS
 <Click to see difference>


    at org.testng.Assert.fail(Assert.java:96)
    at org.testng.Assert.failNotEquals(Assert.java:776)
    at org.testng.Assert.assertEqualsImpl(Assert.java:137)
    at org.testng.Assert.assertEquals(Assert.java:118)
    at org.testng.Assert.assertEquals(Assert.java:453)
    at org.testng.Assert.assertEquals(Assert.java:463)
    at Tests.lowerCaseTest(Tests.java:29) <16 internal calls>
    at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
    at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:84) <4 internal calls>
    at org.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:72)
    at org.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:127)


############################ upperCaseTest ############################
Expected Result: INTRO TO DEVOPS
ActualResult: INTRO TO DEVOPS
###################################################################################

===============================================
Example
Total tests run: 2, Failures: 1, Skips: 0
===============================================


Process finished with exit code 0
```

Smoke Testing

- Any idea?
- Help to decide if is feasible to continue with the rest of the tests.
- Use the Pareto principle.
- Manual or Automated.
- Normally used on:
    - Integration and System Level.
- Pesticide paradox principle - Important

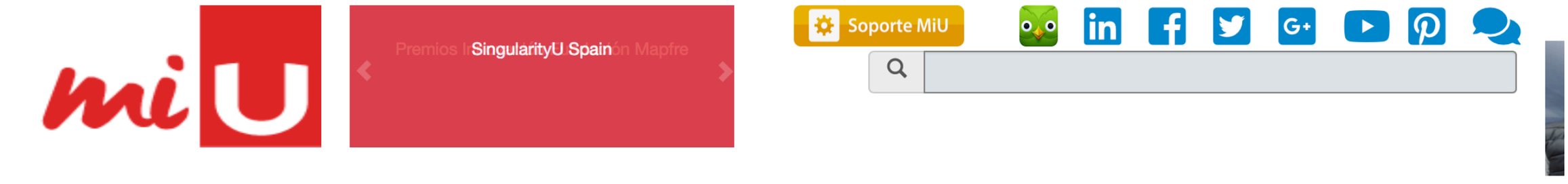
EXAMPLE

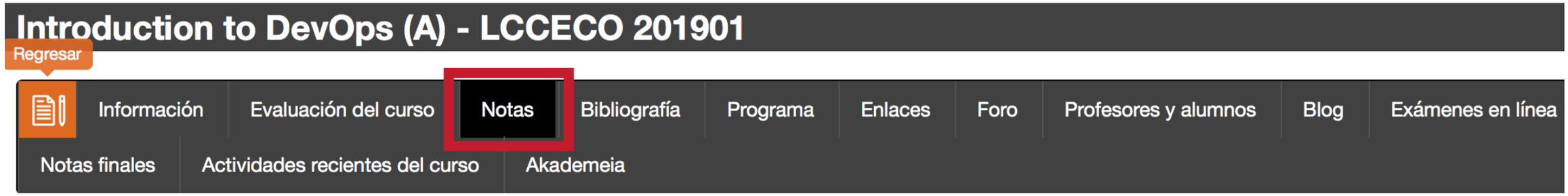




Integration Level

EXAMPLE



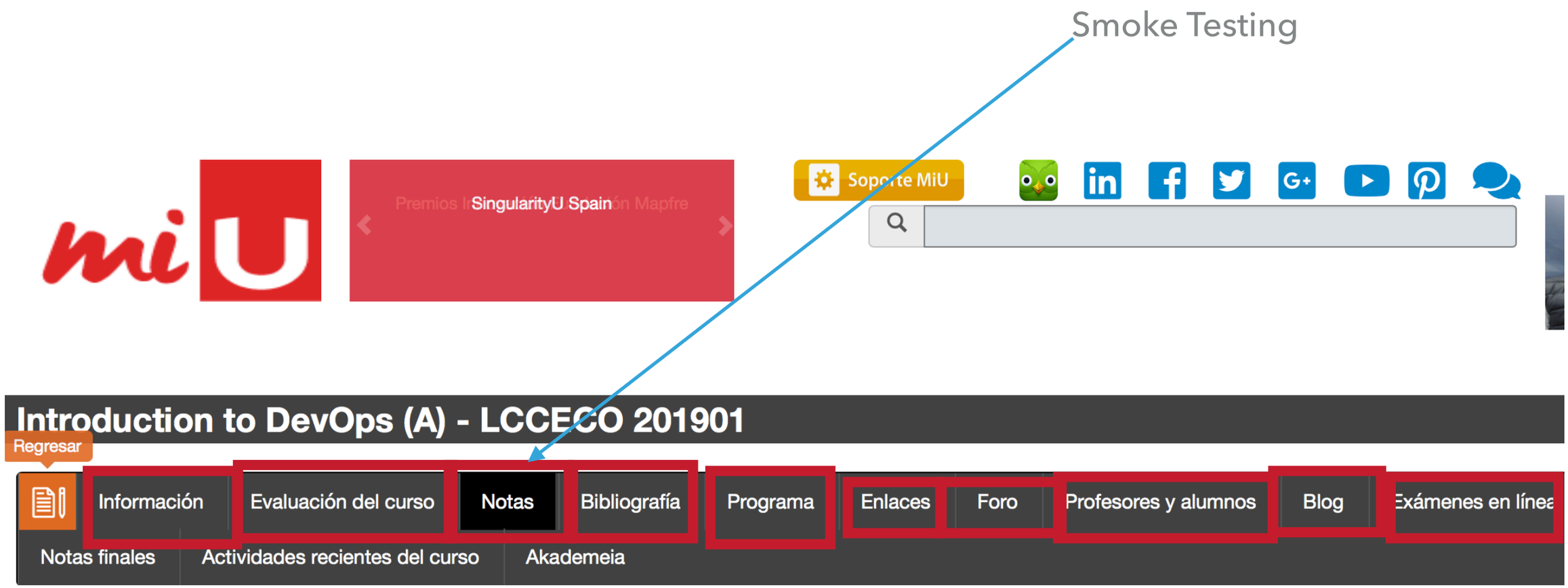


System Level

Sanity Testing

- Checks bugs reported in the previous build.
- Checks if breaks another functionalities.
- Checks that the expected functionality works fine.
- Avoid wasting time and money.
- Subset of Acceptance Criteria.
- Checks the main functionalities without going deeper.

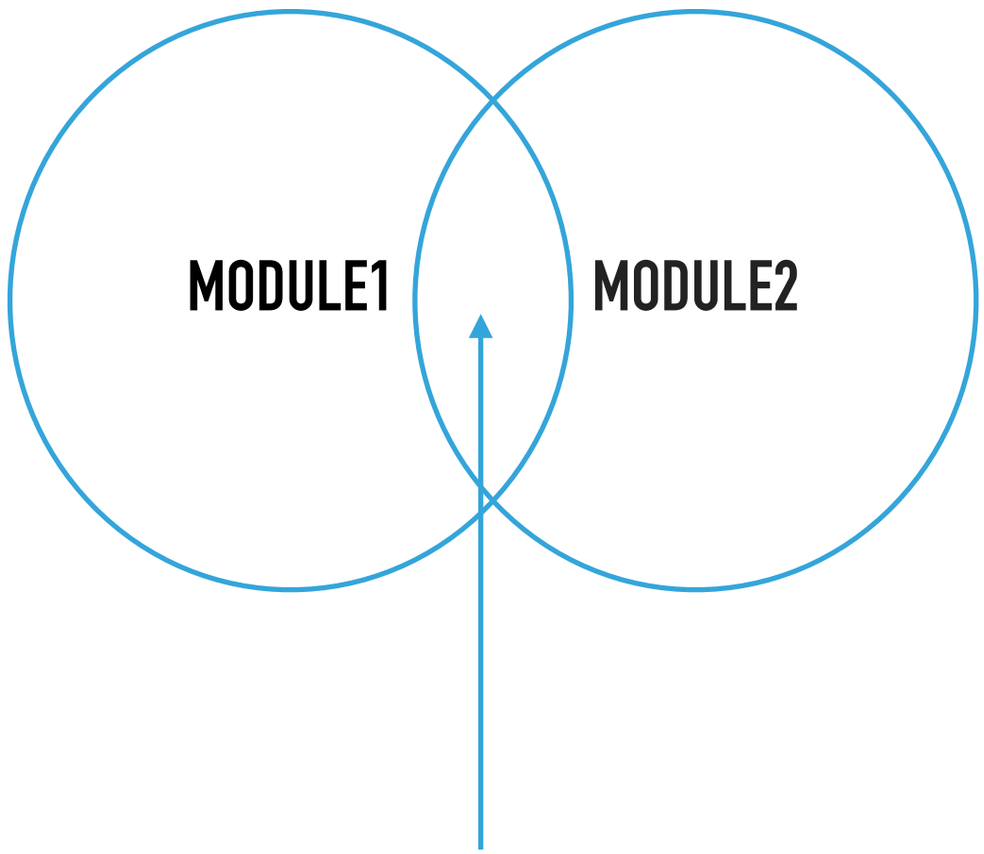
EXAMPLE


Smoke Testing
SingularityU Spain
Soporte MiU
Introduction to DevOps (A) - LCCECO 201901
Regresar
Información
Evaluación del curso
Notas
Bibliografía
Programa
Enlaces
Foro
Profesores y alumnos
Blog
Exámenes en línea
Notas finales
Actividades recientes del curso
Akademeia

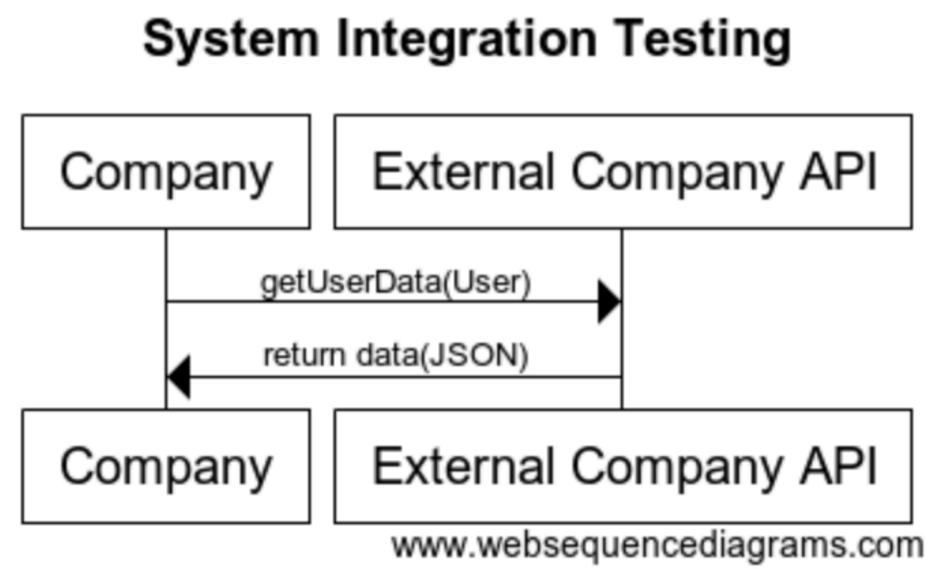
Integration Testing

- After Unit Testing occurs.
- Small unit are combined.
- Two levels:
  - Component Integration Testing.
  - System Integration Testing.
- Testers/Developers.

EXAMPLE



Component Integration Testing



System Testing

- After Integration Testing occurs.
- Behavior and capabilities of a whole system.
- End-to-End Testing.
- Produces information that is used by stakeholders to make release decisions.
- Test Environment.
- Automated or manual.
- Testers.

Analogy of System Testing

Acceptance Testing

- Acceptance criteria is needed.
- Focuses on the behavior of the whole system.
- System complete and will work as expected.
- Significant number of defects in some cases is considered a major project risk.
- Common forms to do acceptance testing:
  - UAT.
  - Alpha testing.
  - Beta testing.
  - Contractual and regulatory acceptance testing

UAT

- Environment ≈ Production.
- Main Objective:
  - Building confidence that the user can use the system to meet their needs.
- Techniques:
  - Error Guessing.
  - Exploratory Testing.
  - Check List based Testing
- Who does these tests?
  - Real users or a separate team in the company.

Alpha Testing

- Environment ≈ Production.
- Dedicated test team in the company.
- Not open to the market.
- Black box testing and white box testing.
- Reduce risk.

Beta Testing

- Environment = Production.
- Real Users.
- Open to the market.
- Black box testing.
- Detection of defects related to the conditions and environment(s)
- In some cases happens after alpha testing occurs.

Contractual and regulatory acceptance testing

- Contractual Testing
  - Contract's acceptance criteria -
  - Real Users or independent testers.
- Regulatory acceptance testing:
  - Example: Legal or security.
  - Real users, independent testers, regulatory agencies.

Negative Testing

- Error path testing or failure testing.
- Validate the behavior of the application with invalid data.
- Real and more scenarios.
- How do you can define the valid and invalid data?
    - Equivalence partitioning.
    - Boundaries Values.

EXAMPLE

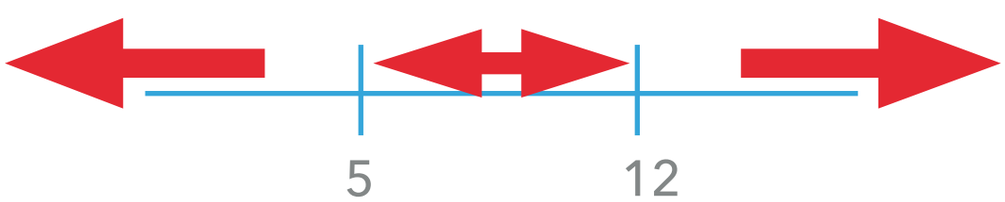
Specifications:

No symbols and numbers should be accepted.

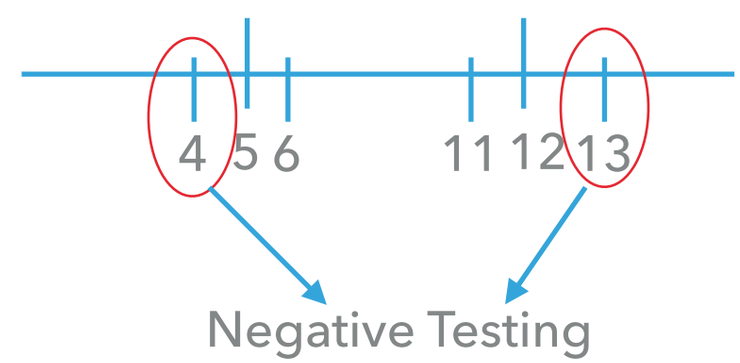
5 to 12 characters.

Name:

Equivalence Partitioning.



Boundaries values.



NON-FUNCTIONAL TESTING

FOR YOU, WHAT COULD BE A NON-FUNCTIONAL ASPECT?

NON-FUNCTIONAL TESTING

Some non-functional aspects:

- Performance.
- Usability.
- Security.
- Visual.
- Resilience

NON-FUNCTIONAL TESTING

- How well the system behaves.
- Common is performed at all levels.
- Late discovery = dangerous to the success of the project.
- Black-box testing.
    - Example: Boundary value analysis can be used to define the stress conditions for performance testing.
- Involve special skills.

PERFORMANCE TESTING

- The performance of an application is measured from different perspectives.
- Different types of performance testing.
  - Load testing.
  - Stress testing.
- Continuously monitored.

LOAD TESTING

- Modeling the expected usage of the application.
- Simulate multiple users concurrently.
- Checks how the application behave during normal or hight usage.
- Helps to determine how many users can be handle.
- Find bugs that are not able to detect using another testing technique.

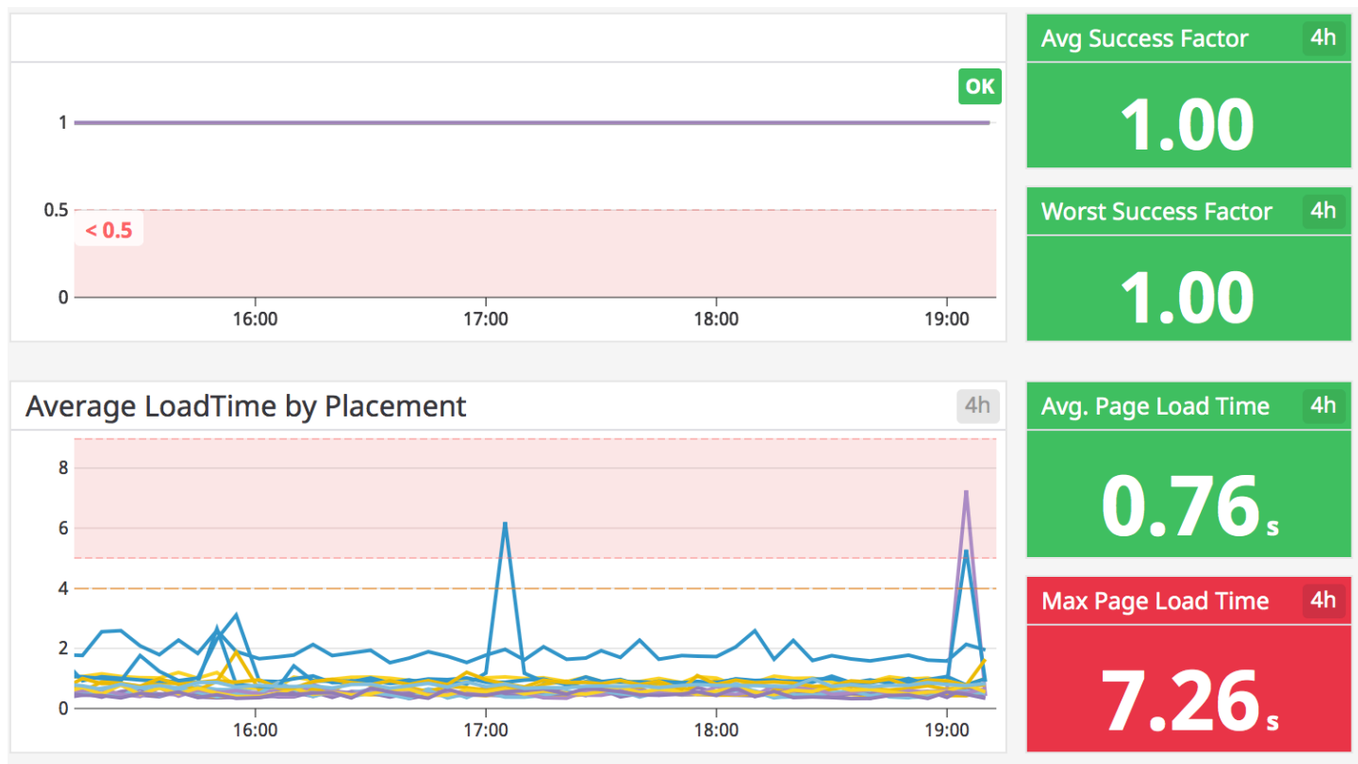
STRESS TESTING

- Load placed on the system is increased beyond the normal usage.
- How the application response.
- Long period of time. (Hours or days).
- Test the robustness of the hardware system.

WHAT'S THE MAIN DIFFERENCE BETWEEN LOAD AND STRESS TESTING.

PERFORMANCE MONITORING

- System Owners.
- Performance degradations. (Example?).
- Tools.



USABILITY TESTING

- Evaluate the ease of use which the end users of the system access the application.
- ISO 9214–11
  - Effectiveness.
  - Efficiency.
  - Satisfaction.
- Key principles of usability testing:
  - Communication.
  - Provide information users wants.
  - Offer simple.

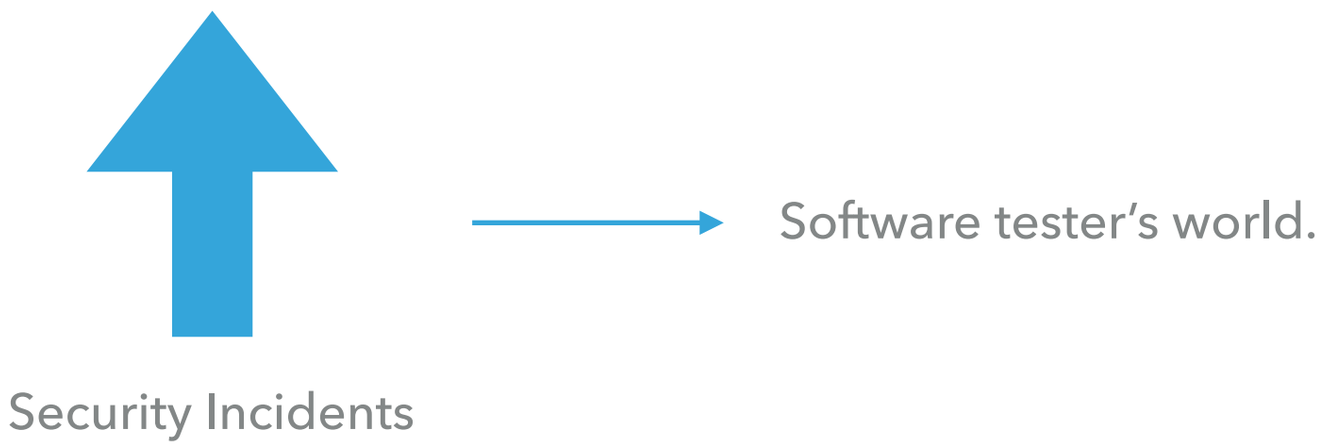
USABILITY TESTING

- Discover needs and expectations.
- Critical for usability testing:
  - Visibility.
  - Match between system and the real world.
  - Ease learning.
- Feedback direct from target audience.

USABILITY TESTING – EXAMPLE

SECURITY TESTING

- Was performed by network administrators or system developers.



- 3 of 4 web sites are vulnerable to an attack and 75% of the hacks occur at the application level (Gartner's reports).

SECURITY TESTING

- Main Objectives:
  - Security requirements.
  - Identify and solve vulnerabilities of the application.
- Should be developed in each phase of the SDLC.
- Manual or automated.
- Examples ?
  - IP Restriction.
  - Ports Enable.
  - Pen testing (Penetration testing).

VISUAL TESTING

- Main Objectives:
  - Visual requirements.
  - Rendering in different browsers/OS
- Manual or automated.
- Examples ?
  - Responsive
  - Element positions.

RESILIENCE TESTING

- Ensure the application perform well in real conditions.
- Acceptance service level to business.
- Recovery.
- Example:

EXERCISE

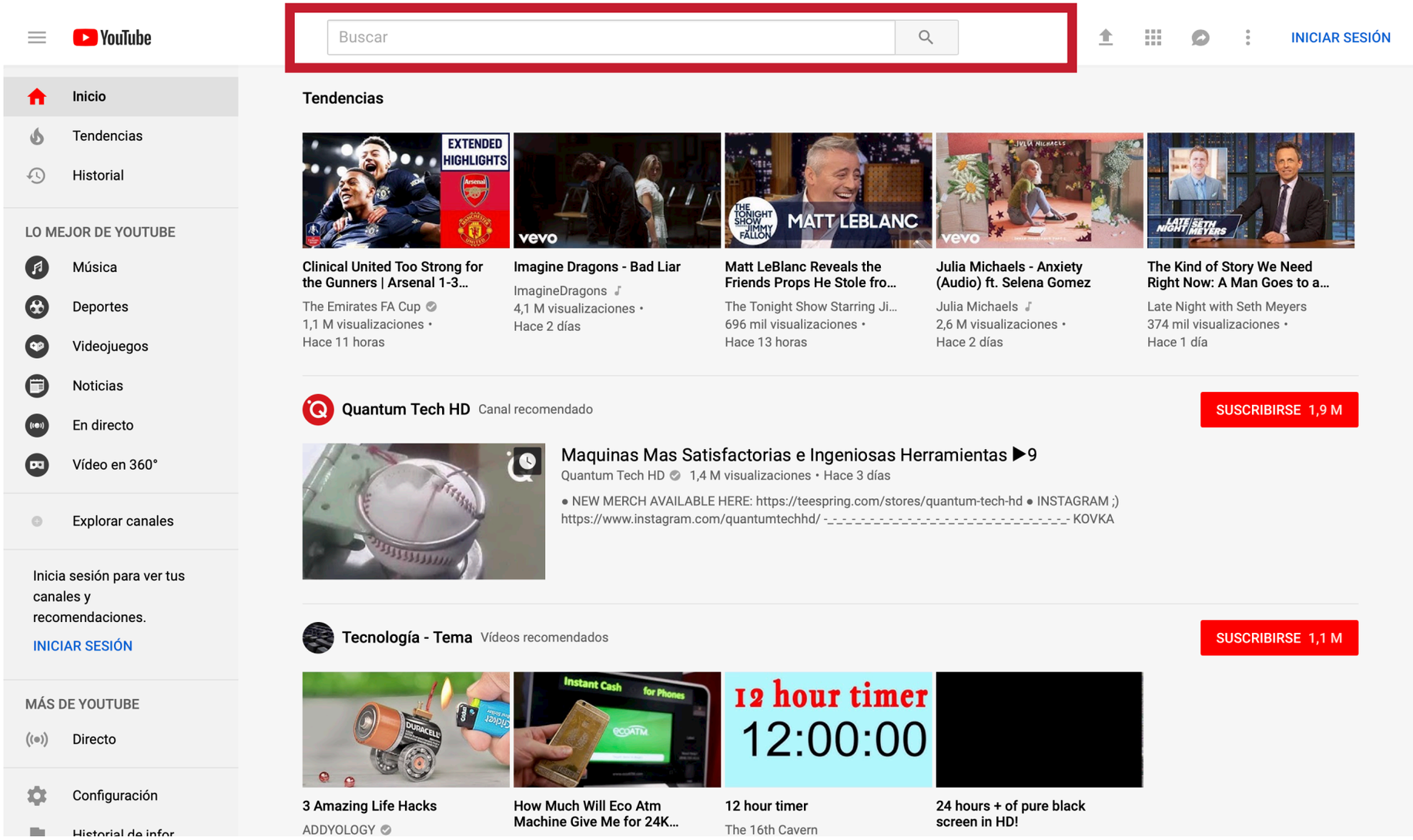
- Select one of the following websites.

- Identify 6 functional and non-functional aspects.
- Describe what type of functional and non-functional testing you would used and why.

EXAMPLE – YOUTUBE

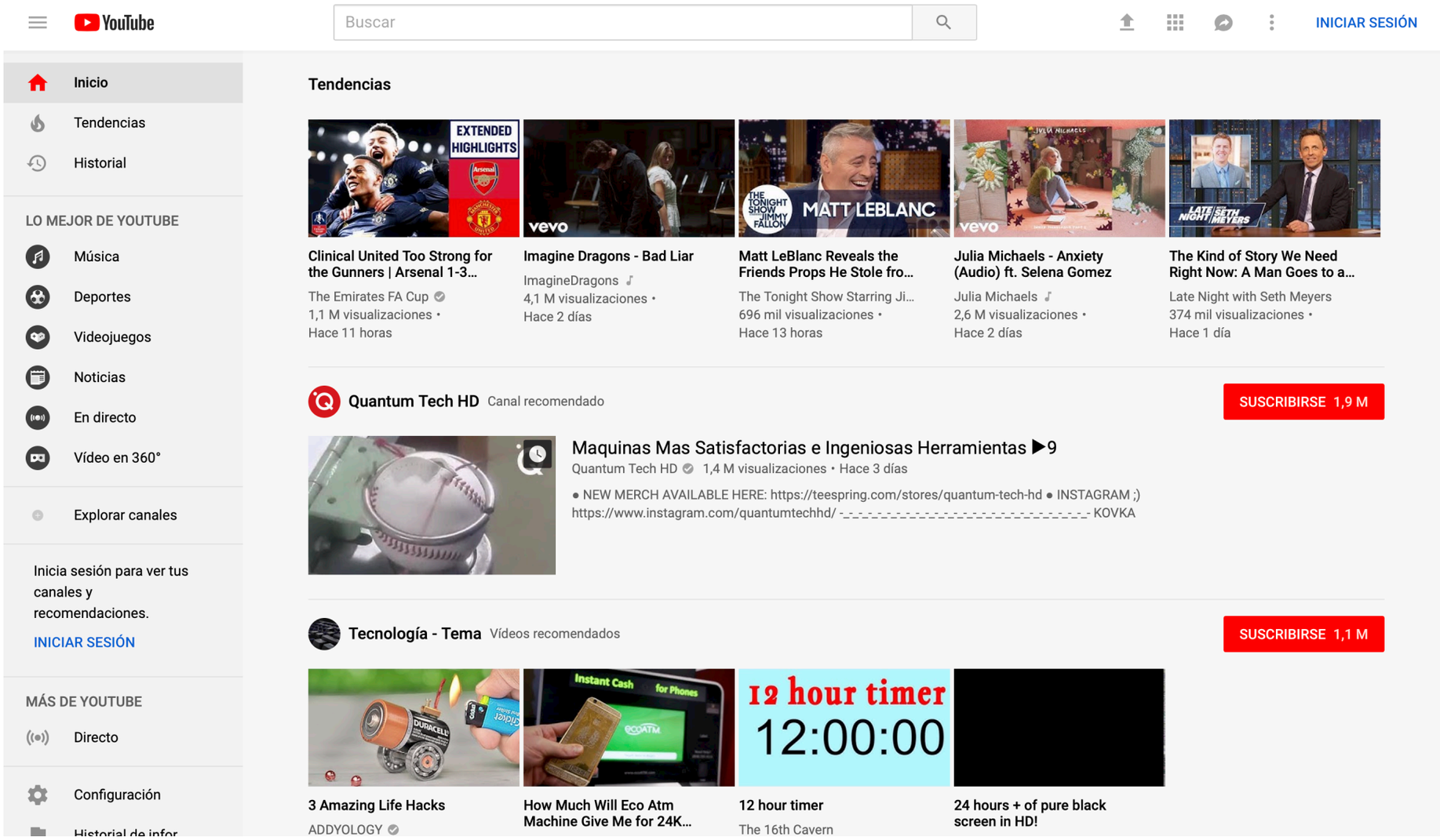
YouTube

YOUTUBE - FUNCTIONAL TEST



- Search Option:
    - Functional Type: Smoke Test
    - Why: Because for me the search option is the most important of youtube. If the user is not able to search video youtube doesn't accomplish their main use and smoke testing help us to check if is feasible continue with the rest of the tests (making sure that the most critical functionalities are working).

YOUTUBE - NON-FUNCTIONAL TEST



- Search Option:
  - Non-Functional Type: Load Test
  - Why: We have to be sure if youtube can make different searches simultaneous with a quick response. Load test simulate multiple users concurrently

THANK YOU.