COMBINING MOCKITO WITH JUNIT FOR SOFTWARE QUALITY CONTROL

JOÃO BRITO, **M9984**
TOMÁS JERÓNIMO, **M9988**

QUALIDADE DE SOFTWARE
PROF. NUNO POMBO
20 DECEMBER 2019

INTRODUCTION

- Foundational concepts: Unit Testing;
- Two testing frameworks: Mockito and JUnit;
- Case study: Application overview;
- Unit tests;
- Results discussion.

UNIT TESTING

- **Individual components** are the test subjects;
- They are **examined in isolation** and should perform according to the accepted specifications;
- In the context of **OOP**, **each method is a unitary block**;

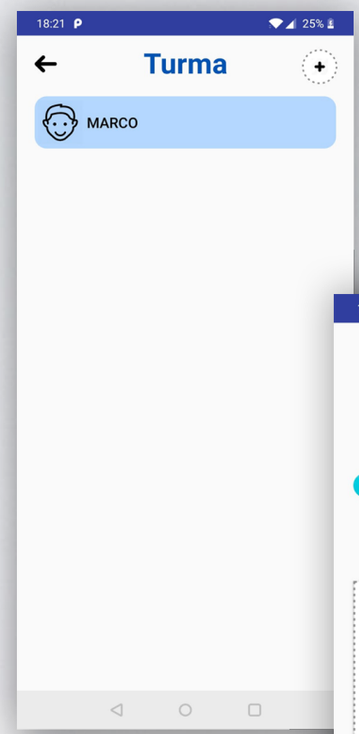
JUNIT

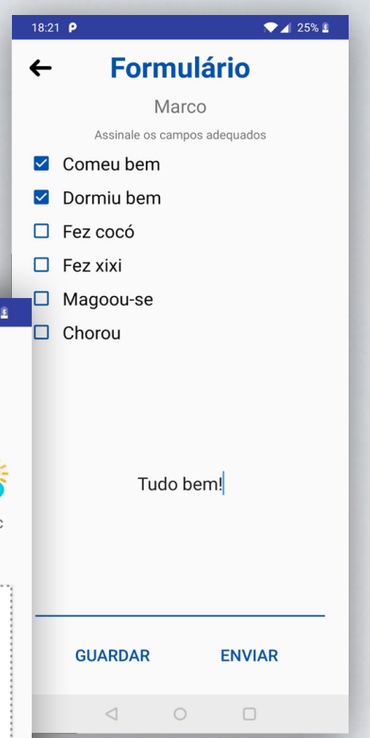
- **Open source testing framework**;
- Allows the creation of **automated tests**;
- A JUnit test is a **method** contained in a class;
- This method executes the code under test and **checks if the expected result is the same as the actual result**;
- Usually, two unit classes must be created for each requirement (positive test and negative test);
- Promotes the idea of **setting up test data before implementing it**;
- **Increases the stability** of program code and **reduces debugging time**.

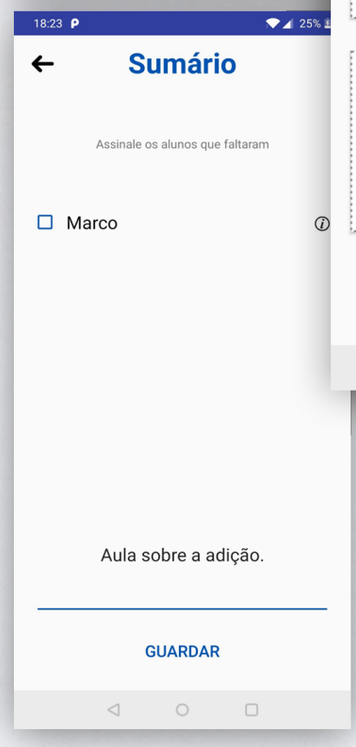
MOCKITO

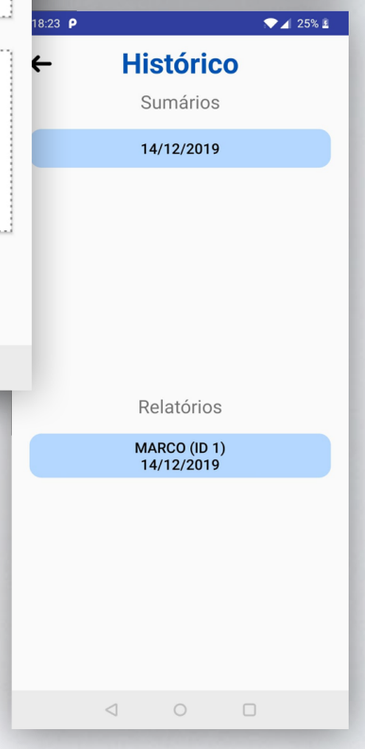
- **Mocking framework**;
- Used to **mock interfaces** to **test dummy functionalities**;
- Perform validations in a **controlled environment**;
- Two fundamental concepts: **mock** and **stub**;
- A mock is a **dummy/fake instance**;
- A stub is a **pre-determined behavior**, to free the class under test from its dependencies;
- Focus on **simplicity** and encourages clean and simple code.

APPLICATION OVERVIEW

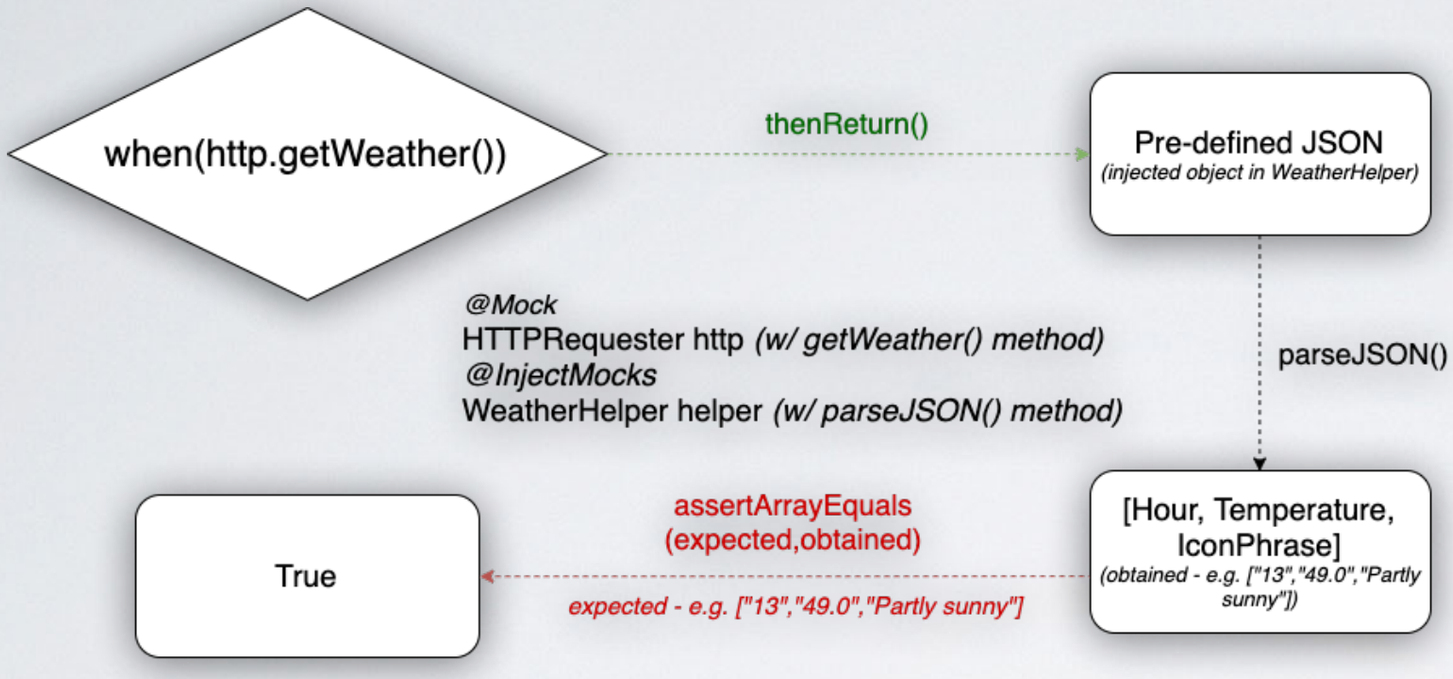








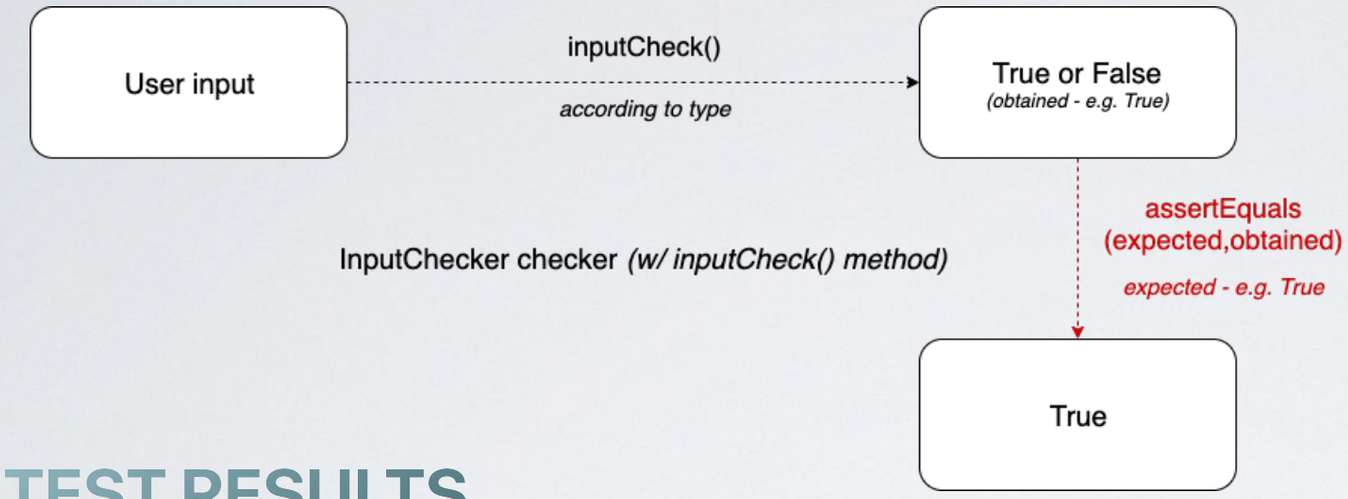
TEST EXTERNAL DEPENDENCIES



TEST RESULTS

Injected object	Expected result	Assert Result
{...”13”..., ...”Partly sunny”..., ... ”29.5”...}	[”13”, ”Partly sunny”, ”29.5”]	True

TEST USER INPUT



TEST RESULTS

Test input	Expected Result	Assert Result
P@ssword1	True	True
#passwoRD123	True	True
(empty input)	False	False
P@ss1	False	False
password	False	False
12345	False	False
#@%&=	False	False
password123	False	False
PASSWORD123	False	False

Test input	Expected Result	Assert Result
teste@gmail.com	True	True
teste@ubi.com	True	True
teste@hotmail.com	True	True
(empty input)	False	False
testehotmail.com	False	False
teste@gmailcom	False	False
teste@gmail.random	**False**	**True**
te%&@gmail.com	False	False
teste@hotm#?l.com	False	False

RESULTS DISCUSSION

- The goal was to **validate the implementation** of several functionalities;

- In the first test, **two main conclusions** were drawn:
  - Communication with AccuWeather API was successful;
  - The parseJSON() method was coded properly.

- In the user input test:
  - Three out of the four test groups fulfilled the expectations;
  - In the email test group, there was **one failed test**.

- The results were satisfactory;

- The **failed test allowed for the code errors to be easily identified** and fixed.

COMBINING MOCKITO WITH JUNIT FOR SOFTWARE QUALITY CONTROL

JOÃO BRITO, **M9984**
TOMÁS JERÓNIMO, **M9988**

QUALIDADE DE SOFTWARE
PROF. NUNO POMBO
20 DECEMBER 2019

QUESTIONS?