METODI E TECNOLOGIE PER LO SVILUPPO SOFTWARE

Nicola Bertazzo

nicola.bertazzo [at] unipd.it

Università degli Studi di Padova

Dipartimento di Matematica

Corso di Laurea in Informatica, A.A. 2021 – 2022

Cos’è Robot Framework?

Robot Framework is a **generic** open source **automation framework**. It can be used for **test automation** and **robotic process automation (RPA)**.

Robot Framework is actively supported, with many industry-leading companies using it in their software development.

Robot Framework **is open** and extensible and can be integrated with virtually any other tool to create powerful and flexible automation solutions. Being open source also means that Robot Framework **is free to use without licensing costs**.

Robot Framework **has easy syntax**, utilizing **human-readable keywords**. Its capabilities can **be extended by libraries implemented with Python or Java**. The framework has a **rich ecosystem around it, consisting of libraries and tools** that are developed as separate projects.

Robot Framework **project is hosted on GitHub** where you can find further **documentation, source code, and issue tracker**. **Downloads are hosted at PyPI**.

Robot Framework **is operating system and application independent**. The core **framework is implemented using Python** and also **runs on Jython (JVM) and IronPython (.NET)**.

Robot Framework itself is open source software released under **Apache License 2.0**, and **most of the libraries and tools in the ecosystem are also open source**.

The framework was **initially developed at Nokia Networks** and was open sourced in 2008.

https://robotframework.org/

Cos’è Robot Framework?

Generic test automation framework
– Utilizes the **keyword-driven** testing approach
– Suitable for both “normal” test automation and ATDD

Implemented with Python
– Runs also on Jython (JVM) and IronPython (.NET)
– Can be extended natively using Python or Java
– Other languages supported via a remote interface

Open source
– Hosted on GitHub, Apache 2 license
– Sponsored by Nokia Networks
– Rich ecosystem and very active community

Pekka Klärck

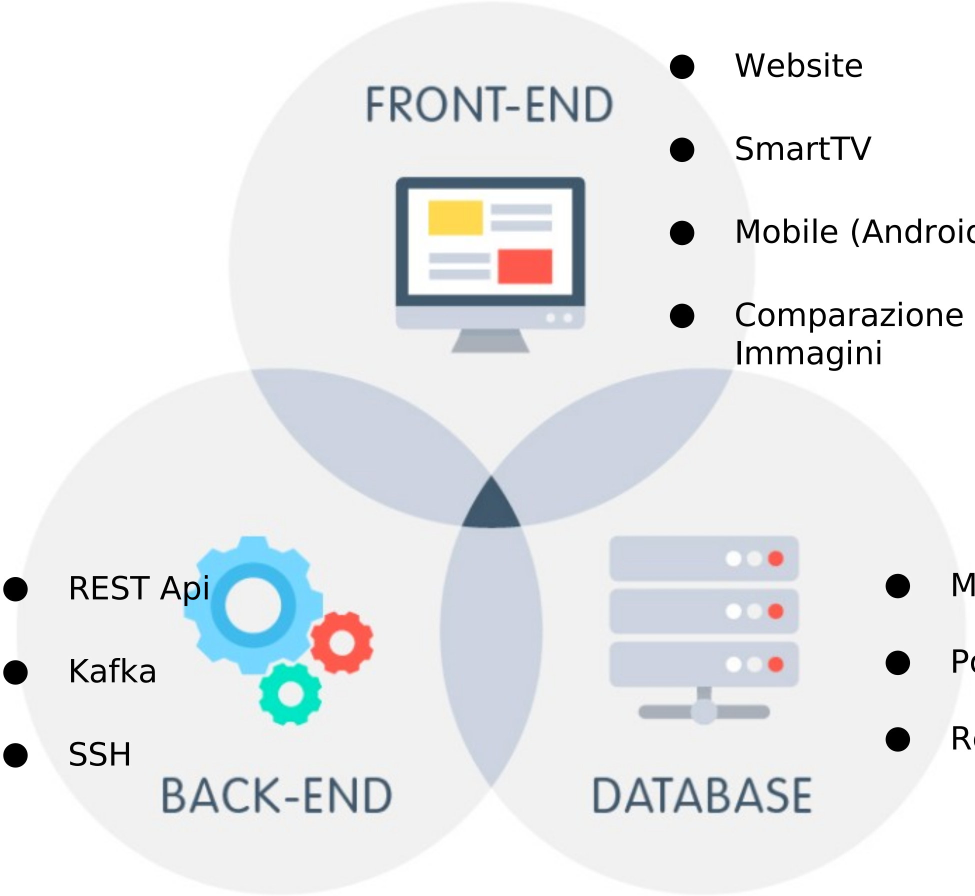
https://www.slideshare.net/pekkaklarck/robot-framework-introduction

Perchè Robot Framework?

- Enables easy-to-use tabular syntax for creating test cases in a uniform way.
- Provides ability to create reusable higher-level keywords from the existing keywords.
- Provides easy-to-read result reports and logs in HTML format.
- Is platform and application independent.
- Provides a simple library API for creating customized test libraries which can be implemented natively with either Python or Java.
- Provides a command line interface and XML based output files for integration into existing build infrastructure (continuous integration systems).
- Provides support for Selenium for web testing, Java GUI testing, running processes, Telnet, SSH, and so on.
- Supports creating data-driven test cases.
- Has built-in support for variables, practical particularly for testing in different environments.
- Provides tagging to categorize and select test cases to be executed.
- Enables easy integration with source control: test suites are just files and directories that can be versioned with the production code.
- Provides test-case and test-suite -level setup and teardown.
- The modular architecture supports creating tests even for applications with several diverse interfaces.

Perchè Robot Framework?

- Python → Portabile
- Open Source → No costi di licenza
- Full Stack
- Curva di apprendimento semplice
- Molte funzionalità Integrate
- Supporto dalla comunity

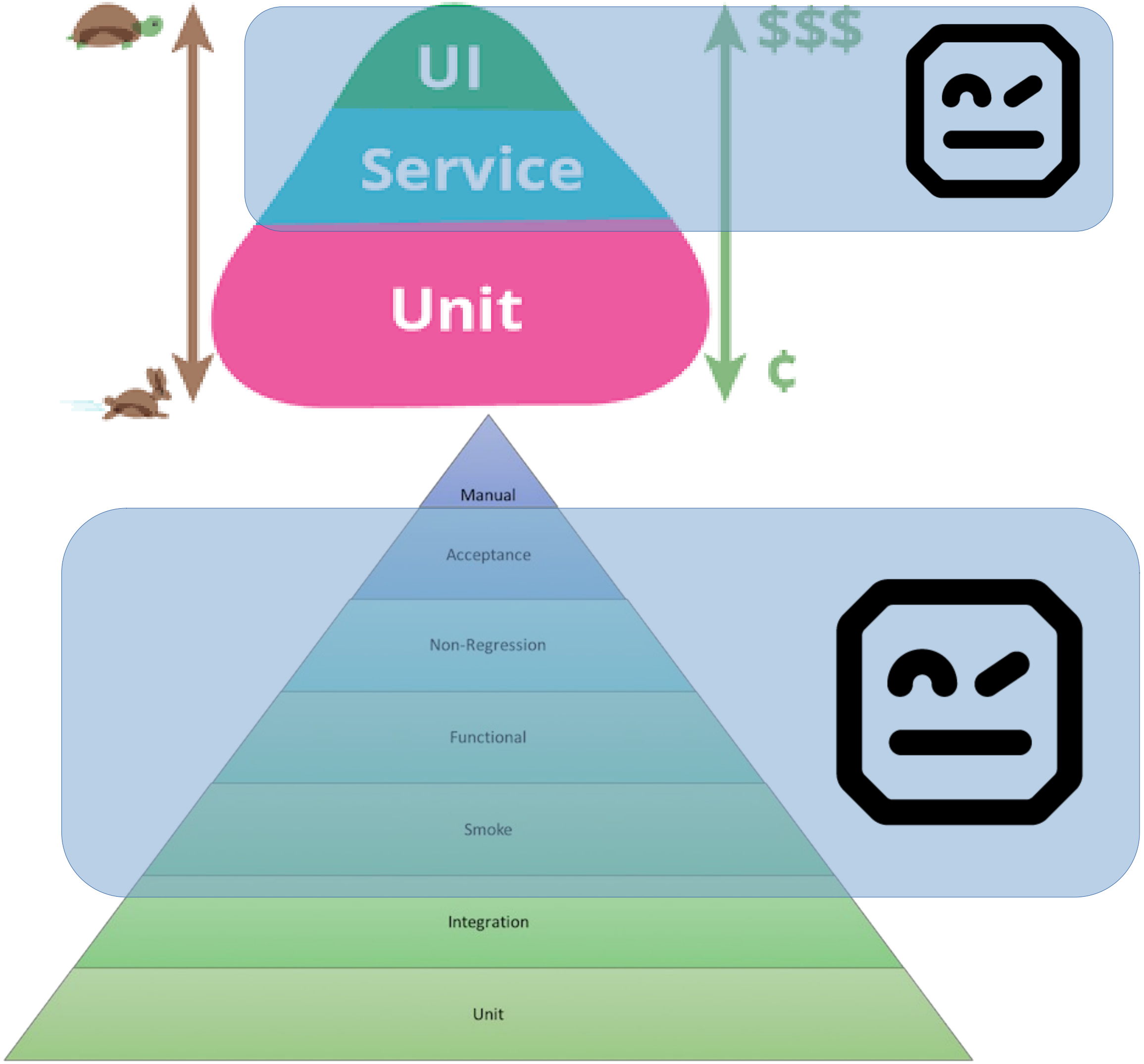


- Website
- SmartTV
- Mobile (Android, Ios)
- Comparazione Immagini

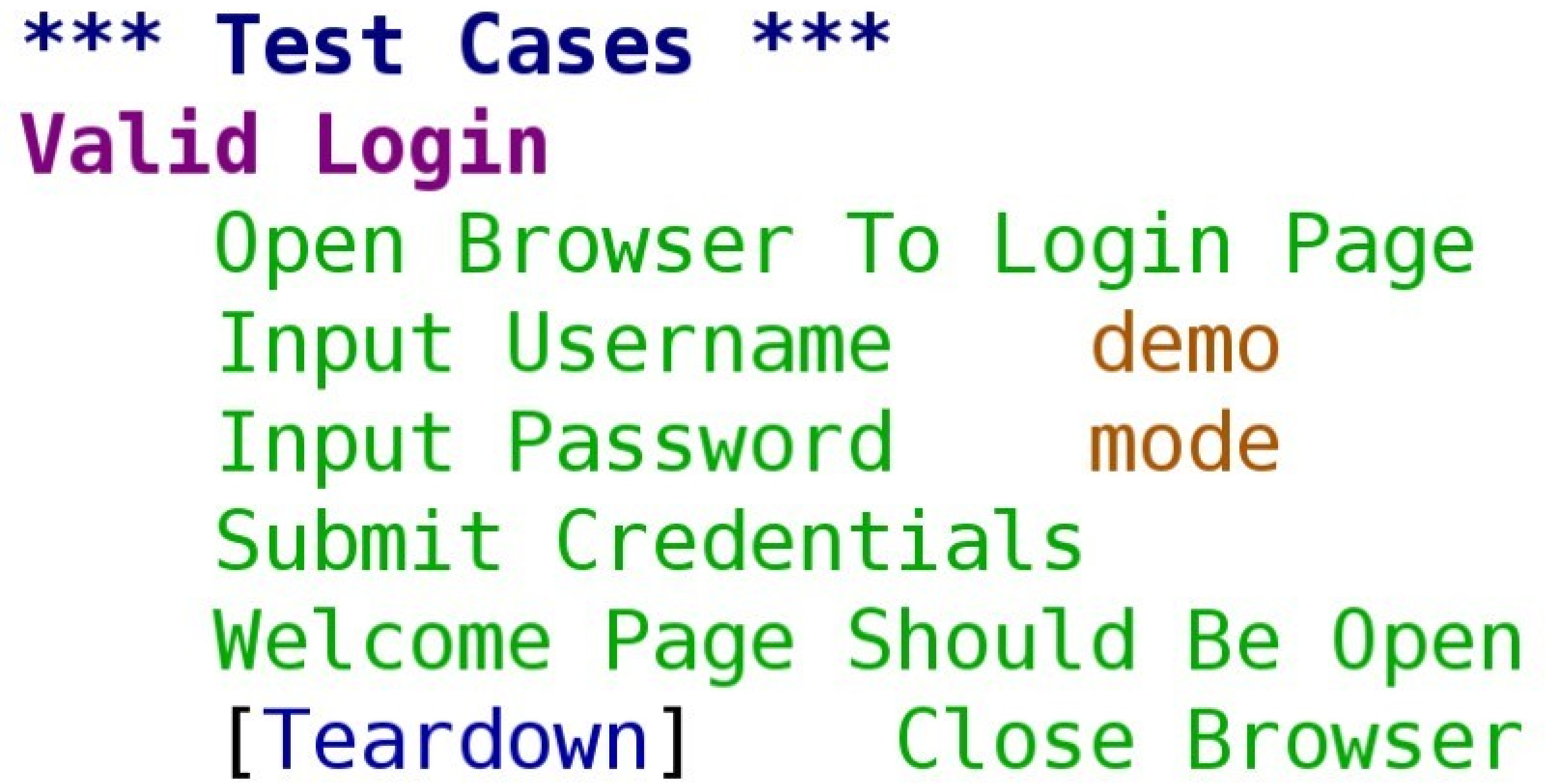
- REST Api
- Kafka
- SSH

- Mongo
- Posgres
- Redis

Quando Framework?

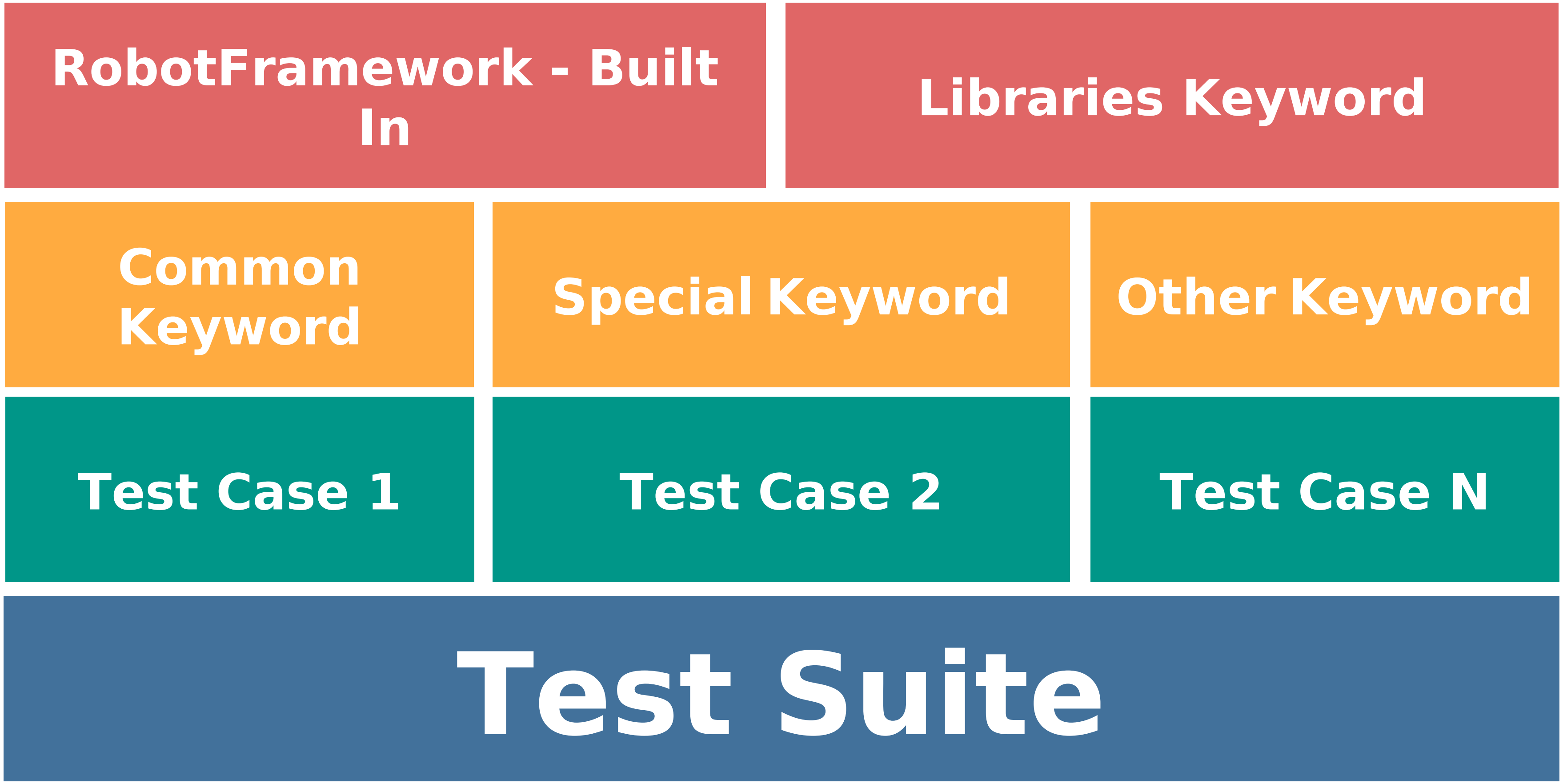


Esempio di Test Case



```
*** Test Cases ***
Valid Login
    Open Browser To Login Page
    Input Username    demo
    Input Password    mode
    Submit Credentials
    Welcome Page Should Be Open
    [Teardown]    Close Browser
```

Come funziona? (keyword-driven testing approach)



RobotFramework - Built In

Libraries Keyword

Common Keyword

Special Keyword

Other Keyword

Test Case 1

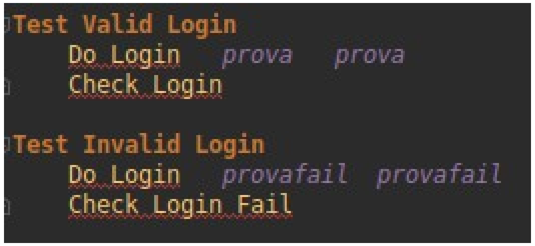
Test Case 2

Test Case N

Test Suite

Robot Framework

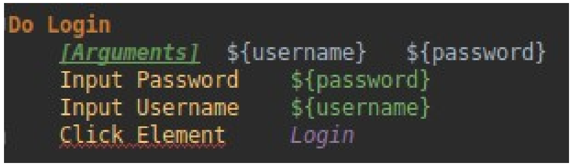
*** Test Case ***



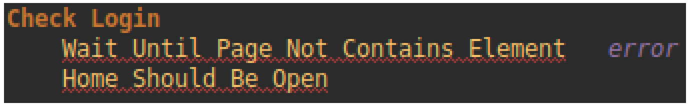
```
Test Valid Login
    Do Login    prova    prova
    Check Login

Test Invalid Login
    Do Login    provafail    provafail
    Check Login Fail
```

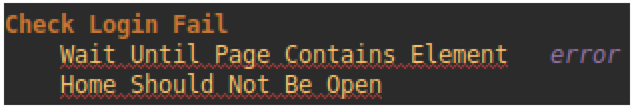
*** Keywords ***



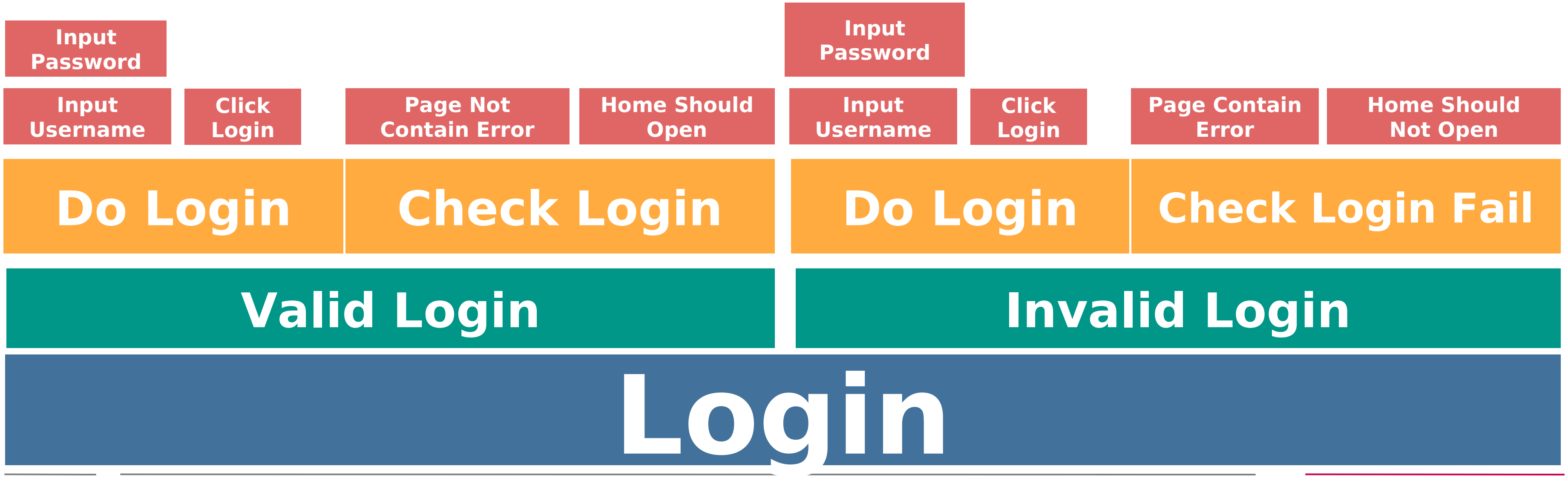
```
Do Login
    [Arguments]    ${username}    ${password}
    Input Password    ${password}
    Input Username    ${username}
    Click Element     Login
```



```
Check Login
    Wait Until Page Not Contains Element    error
    Home Should Be Open
```



```
Check Login Fail
    Wait Until Page Contains Element    error
    Home Should Not Be Open
```



Installazione

Installare:

- Python
- Robot Framework
- Visual Studio Code
- Git

Come indicato nella guida:

installazione-robot-framework-win.pdf

Demo

Primo Test

Creare una cartella

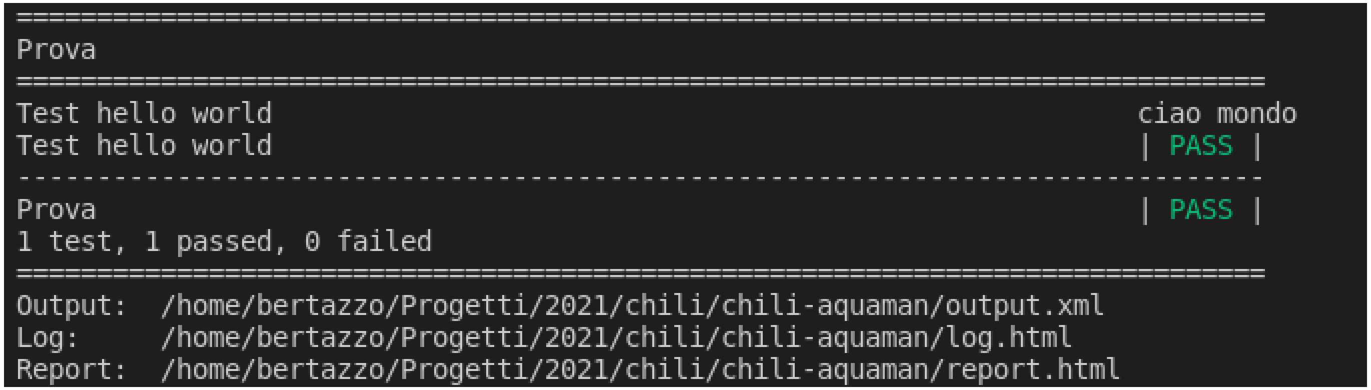
Aprire la cartella con Visual Studio Code

Creare il file prova.robot

```
***Test Cases***
Test hello world
    Log To Console  ciao mondo
```

- Aprire il terminale da Visual Studio Code ed eseguire il seguente comando:

robot prova.robot



```
==============================================================================
Prova
==============================================================================
Test hello world                                                      ciao mondo
Test hello world                                                      | PASS |
------------------------------------------------------------------------------
Prova                                                                 | PASS |
1 test, 1 passed, 0 failed
==============================================================================
Output:  /home/bertazzo/Progetti/2021/chili/chili-aquaman/output.xml
Log:     /home/bertazzo/Progetti/2021/chili/chili-aquaman/log.html
Report:  /home/bertazzo/Progetti/2021/chili/chili-aquaman/report.html
```

Caso Complesso (1)

Seguire:

https://github.com/robotframework/QuickStartGuide/blob/master/QuickStart.rst

Clonare il progetto

Aprire una nuova finestra in Visual Studio Code

Selezionare controllo del codice sorgente

Clonare il seguente repository:

https://github.com/robotframework/QuickStartGuide.git

Analizzare e provare l'applicazione da verificare (SUT)

Seguire la guida Demo application:

https://github.com/robotframework/QuickStartGuide/blob/master/QuickStart.rst#demo-application

Demo

Caso Complesso (2)

Eseguire i test

[Opzionale]Creare e attivare il venv

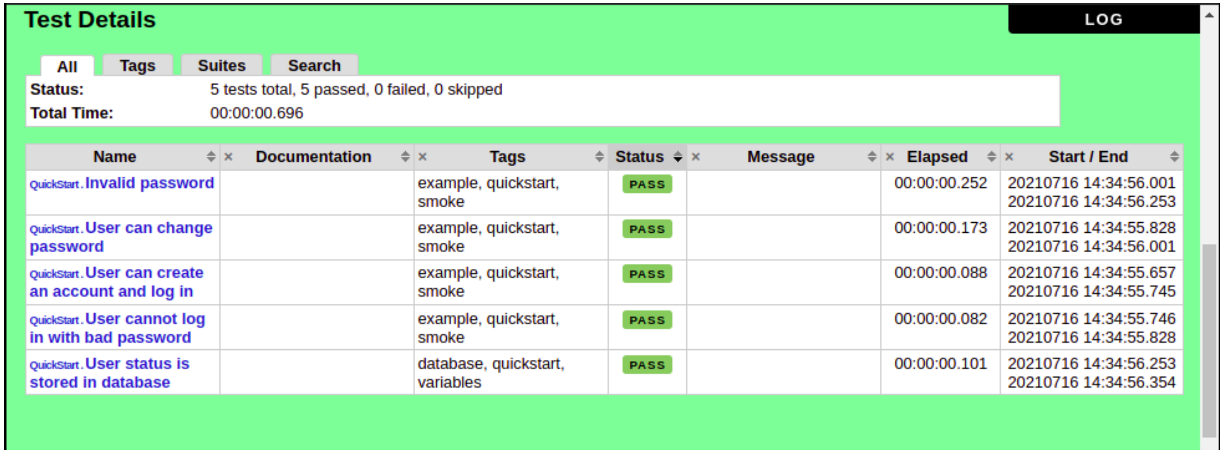
Installare le dipendenze:

pip install robotframework

pip install docutils

Da terminale di VSC:

robot QuickStart.rst



Test Details

Status:	5 tests total, 5 passed, 0 failed, 0 skipped
Total Time:	00:00:00.696

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
QuickStart.Invalid password		example, quickstart, smoke	PASS		00:00:00.252	20210716 14:34:56.001 20210716 14:34:56.253
QuickStart.User can change password		example, quickstart, smoke	PASS		00:00:00.173	20210716 14:34:55.828 20210716 14:34:56.001
QuickStart.User can create an account and log in		example, quickstart, smoke	PASS		00:00:00.088	20210716 14:34:55.657 20210716 14:34:55.745
QuickStart.User cannot log in with bad password		example, quickstart, smoke	PASS		00:00:00.082	20210716 14:34:55.746 20210716 14:34:55.828
QuickStart.User status is stored in database		database, quickstart, variables	PASS		00:00:00.101	20210716 14:34:56.253 20210716 14:34:56.354

Analizzare il risultato

Aprire il report prodotto dall’esecuzione con un browser

Caso Complesso2 (1)

Seguire:

https://github.com/robotframework/RobotDemo

Clonare il progetto

Aprire una nuova finestra in Visual Studio Code

Selezionare controllo del codice sorgente

Clonare il seguente repository:

https://github.com/robotframework/RobotDemo.git

Analizzare e provare l'applicazione da verificare (SUT)

Seguire la guida Demo application:

https://github.com/robotframework/RobotDemo#demo-application

Caso Complesso (2)

Eseguire i test

Creare e attivare il venv

```
python3 -m venv robotdemo
source ./robotdemo/bin/activate
pip install -r requirements.txt
```

Da terminale di VSC:

```
robot *.robot
```

Analizzare il risultato

Aprire il report prodotto dall’esecuzione con un browser

Demo

Community

Community ufficiale:

https://robotframework.org/#community

Commuinty italia:

Meetup Robot-Framework-Milano: https://www.meetup.com/Robot-Framework-Milano/

Fonti

https://robotframework.org/#introduction

https://robotframework.org/#examples

Robot Framework Introduction

https://istqb.ita-stqb.org/docs/ITASTQB-FLSY-2018.pdf

https://github.com/robotframework/robotframework/blob/master/INSTALL.rst

https://github.com/robotframework/QuickStartGuide/blob/master/QuickStart.rst

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#id518