



Appium e Python: iniciando automação de testes em dispositivos móveis

Maria Clara
Learning session - Dia 2

02.Outubro.2019

Retrospectiva

- **Contexto** de qualidade e de automação (riscos e benefícios)
- **Setup** do nosso ambiente (IDE, Android Studio, Appium Desktop)
- **Instância** de um dispositivo Android emulado (Android P)
- Comandos **ADB**
- Funcionalidades básicas do **Appium** (tap, gestos, recording)
- Instalamos a aplicação **Casas Bahia**
- ***Desired capabilities*** no Appium
- Replicamos isso em **código** via Python



E hoje?

- **Retrospectiva** rápida do que vimos no nosso último encontro
- Abrir novamente nosso **ambiente**: Android Studio, Appium, IDE
- **Inicializar** a instância do nosso celular
- Ver se tá tudo **funcionando**
- Definir **nosso primeiro teste**
- Escrever o **código** do primeiro teste
- Aplicar alguns **frameworks e padrões**: pytest, unittest, pageobjects.



Page Objects

- **POM (Page Object Model):** Padrão de projeto bastante utilizado em automação.
- Um ***page object*** é uma classe orientada a objeto.
- **Benefícios:** a alteração do código vai acontecer apenas no método relacionado ao funcionamento, não nos métodos do teste.
- Tudo relacionado a UI vai estar **num único lugar**.
- Testes e mapeamentos de elementos acontecem **em lugares diferentes**.
- O **código** fica mais amigável e de manutenção mais fácil.



UnitTest

- **Framework** de testes unitários.
- É bastante usado em automação de testes por oferecer compartilhamento de **setup()**, **tearDown()**, agregar execução de testes por collections:
 - **test fixture:** toda configuração necessária para executar uma coleção de testes;
 - **test case:** identifica a classe que servirá de base para os casos de teste
 - **test suite:** uma coleção de casos de teste
 - **test runner:** quem vai guiar o conjunto de execuções.



Pytest

- **Framework** que facilita a execução e organização de códigos menores.
- Possibilita o **assertion**.

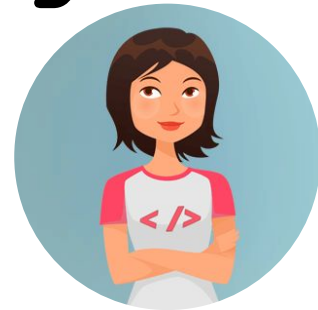
```
>> pip install pytest
```



Bora melhorar esse código?



Bora melhorar esse código?



```
from appium import webdriver
```

```
desired_caps = {  
    "platformName": "Android",  
    "deviceName": "Appium1",  
    "appPackage": "com.novapontocom.casasbahia",  
    "appActivity": "br.com.viavarejo.feature.home.HomeActivity"  
}
```

```
driver = webdriver.Remote("http://localhost:4723/wd/hub", desired_caps)
```


Sugestão para próximos passos

- **PEP8** - Ótimo padrão para escrita e revisão de código
- **PyTest** - Framework do Python para automação de testes
- **Unittest** - Framework para testes unitários
- **MoT** - Comunidade de teste de software de Recife

Referências

1. [Appium official page](#)
2. [Android Official page](#)
3. [Appium official repository on Github](#)
4. [Pytest official page](#)
5. [Evozi apk-downloader](#)
6. [Meetups of Ministry of Testing - Recife](#)
7. [Syllabus Apostila CTFL - BSTQB](#)
8. [Blog do CESAR School](#)
9. [PEP8 - Padrão oficial](#)
10. [Pytest](#)
11. [Page Object Model \(POM\)](#)
12. [Unittest](#)
13. [Docker Android Project](#)



C . E . S . A . R

www.cesar.org.br

Rua Bione. 220 | Cais do Apolo |

Bairro do Recife Recife/PE | CEP:

50.030-390 - Brasil

contato@cesar.org.br

+55 81 3425.4700



C . E . S . A . R
centro de estudos e sistemas
avançados do recife



Obrigada! <3

Appium e Python: iniciando automação de teste mobile

Maria Clara
Learning session - Dia 2

02.Outubro.2019