



DOCUMENTAÇÃO

Teste Selenium/Python/Notepad++

Por: João Emanuel Ramos

Pedro Nishiyama

SUMÁRIO

- 1.Gravar os Testes
 - I. Abra o Windows PowerShell
 - II. Inserir marca d'água
 - III. Abrir o Navegador de testes

- 2.Nomear projeto e inserir um URL
 - I. Abra o Selenium IDE
 - II. Grave o Teste
 - III. Nomeie o Projeto
 - IV. Insira um "URL"

- 3.Iniciando uma gravação
 - I. Confira se estar gravando
 - II. Crie o roteiro
 - III. Nomear o teste

- 4.Importando Teste
 - I. Importe o arquivo
 - II. Importando em Python Pytest
 - III. Selecionar pasta

- 5.Abrindo arquivo importado
 - I. Entendendo o Notepad++

6. Editando arquivo para o uso
 - I. Oque usar?
 - II. Abra o arquivo de molde

7. Realizando o teste.
 - I. Abra o PowerShell
 - II. Realizando o teste

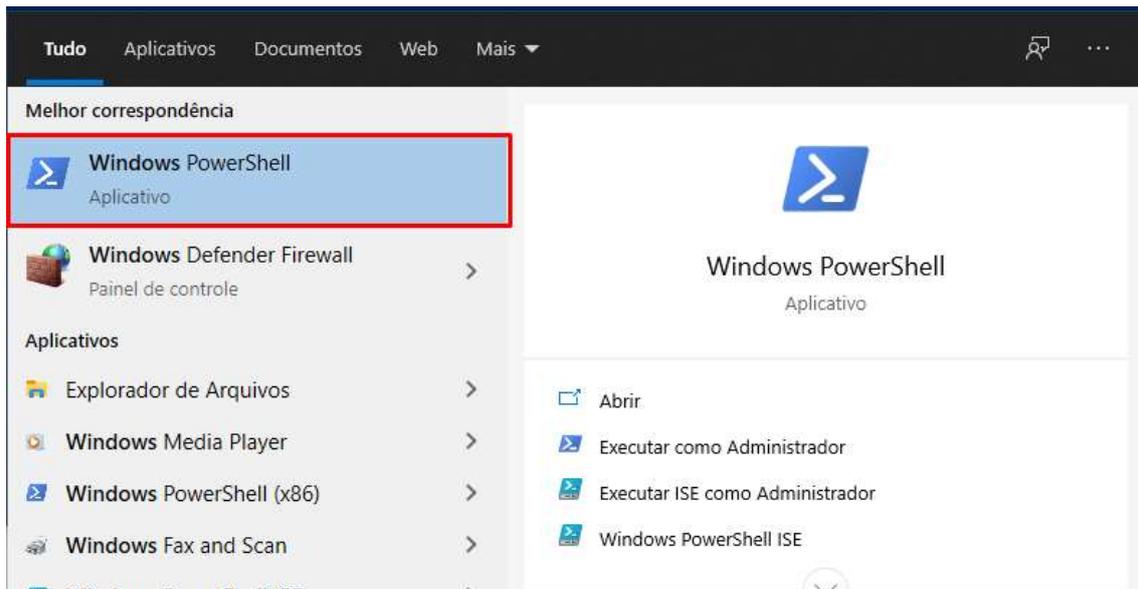
1. Gravar os testes

I – Abra o Windows PowerShell

Clique na barra de pesquisa do Windows e escreva “Windows PowerShell”.



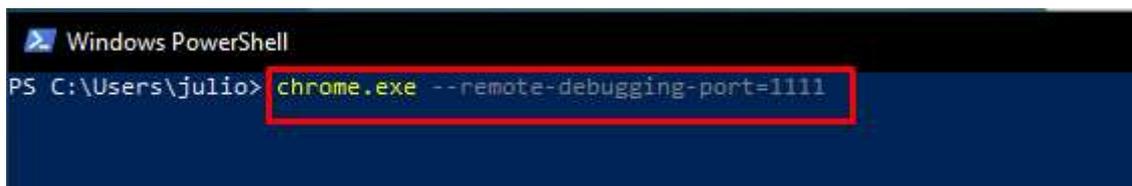
Clique para abrir o programa.



Clea

II – Abrir o Navegador de testes

Digite o comando {chrome.exe --remote-debugging-port=1111}



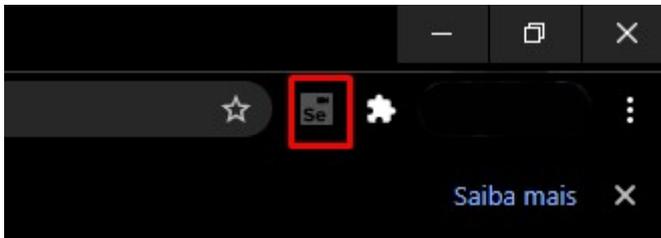
Logo após clique em “Enter” em seu teclado.

Ele irá abrir um navegador onde os testes serão feitos.

2. Nomear projeto e inserir um URL

I – Abra o Selenium IDE

Abra o Selenium IDE localizado na parta superior do “Google Crome”



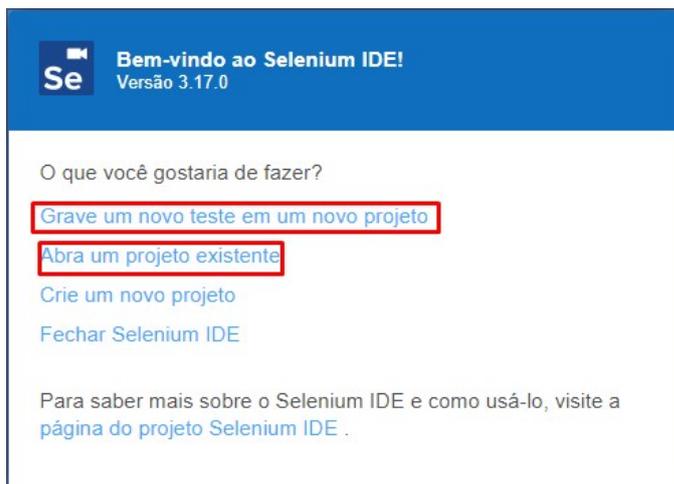
II – Grave o Teste

Clique em,;:

“Grave um novo teste em um novo projeto”

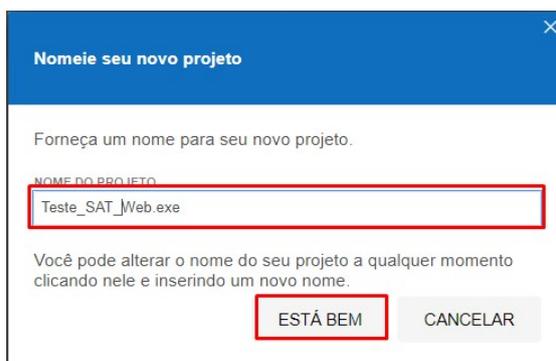
Ou se já tiver fazendo um projeto.

“Abra um projeto existente”



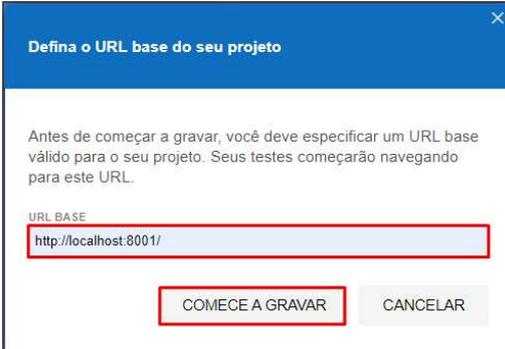
III – Nomeie o Projeto

De nome ao projeto e clique em “Está Bem”



IV – Insira um “URL”

Insira o seguinte URL (<http://localhost:8001/>)



Defina o URL base do seu projeto

Antes de começar a gravar, você deve especificar um URL base válido para o seu projeto. Seus testes começarão navegando para este URL.

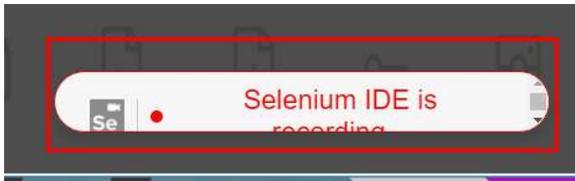
URL BASE

Depois clique em **(Começar a Gravar)**

3. Iniciando uma gravação

Quando for fazer qualquer teste vai ser necessário que grave desde da parte de fazer o login no SAT até no passo em começa os testes.

I – Confira se estar gravando



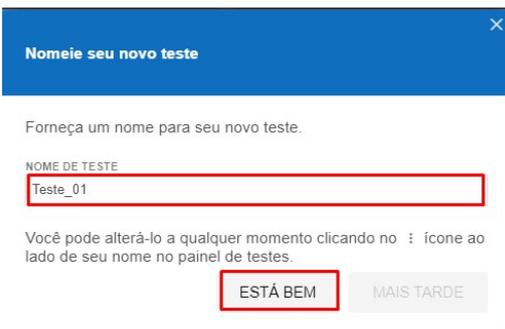
II – Crie o roteiro

Ao terminas de criar o roteiro Clique em “Stop” no selenium



III – Nomear o teste

Depois de criar um script, de um título a o teste e clique em “Está Bem”



Nomeie seu novo teste

Forneça um nome para seu novo teste.

NOME DE TESTE

Você pode alterá-lo a qualquer momento clicando no  ícone ao lado de seu nome no painel de testes.

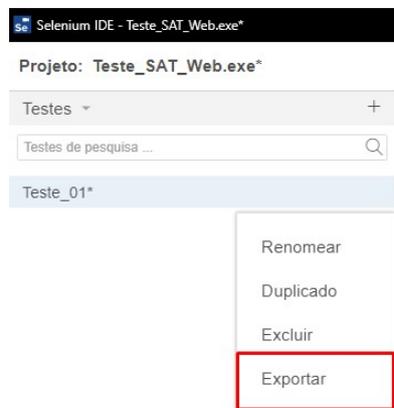
4. Importando Teste

I. Importe o arquivo

Clique no botão sublinhado abaixo

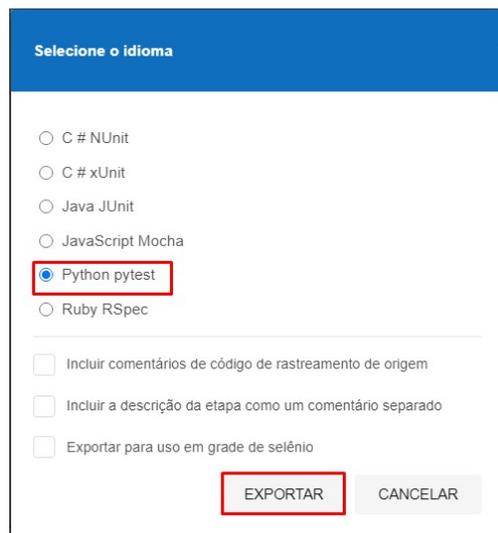


Logo após clique em Exportar



II. Importando em Python pytest

Selecione a opção Python Pytest depois clique em Exportar



III. Selecione pasta

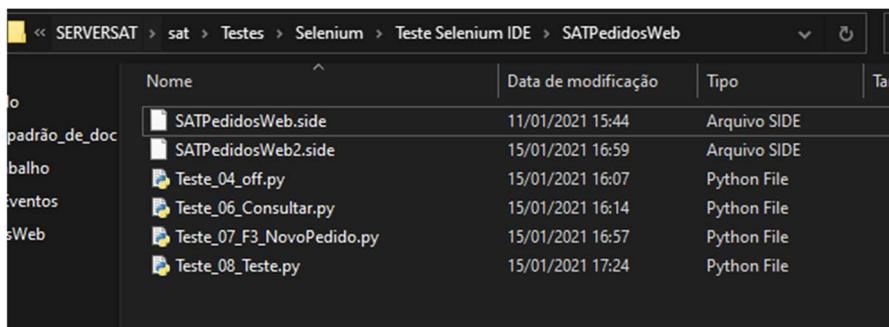
Obs. para um melhor uso das pastar

\\serversat\SAT\Testes\Selenium\Teste Selenium IDE

Insira nessa pasta o que é importado do selenium ide

Selecione a pasta ou crie sua pasta do teste em questão

Obs. Cada teste deve ser exportado separadamente.

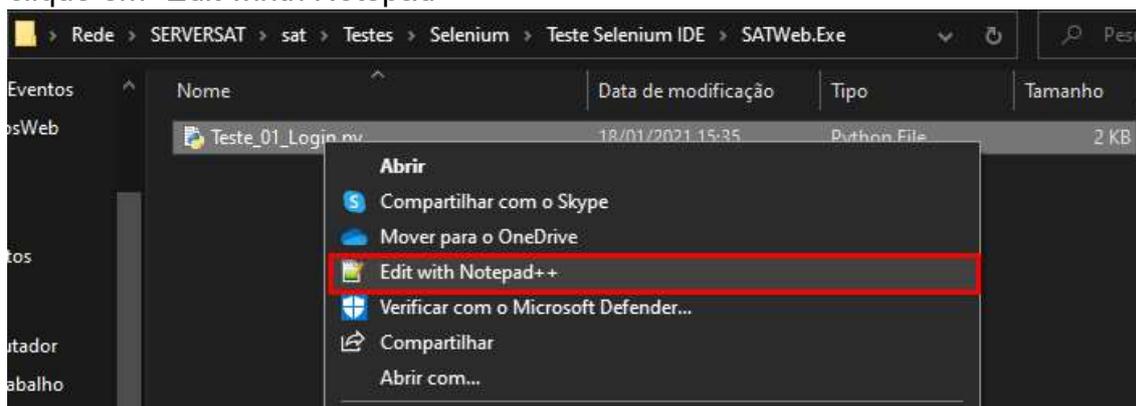


Obs. Para usar dessa maneira é necessário baixar e instalar esses programas

- Python 3.9 “64-bit”
- Notepad++
- Selenium ide

5. Abrindo arquivo importado

I. Selecione o arquivo do teste em questão e com botão direito do mouse clique em “Edit with Notepad++”



6. Editando arquivo para o uso

I. O que usar?

```

1 # Generated by Selenium IDE
2 import pytest
3 import time
4 import json
5 from selenium import webdriver
6 from selenium.webdriver.common.by import By
7 from selenium.webdriver.common.action_chains import ActionChains
8 from selenium.webdriver.support import expected_conditions
9 from selenium.webdriver.support.wait import WebDriverWait
10 from selenium.webdriver.common.keys import Keys
11 from selenium.webdriver.common.desired_capabilities import DesiredCapabilities
12
13 class TestTeste01():
14     def setup_method(self, method):
15         self.driver = webdriver.Chrome()
16         self.vars = {}
17
18     def teardown_method(self, method):
19         self.driver.quit()
20
21     def wait_for_window(self, timeout = 2):
22         time.sleep(round(timeout / 1000))
23         wh_now = self.driver.window_handles
24         wh_then = self.vars["window_handles"]
25         if len(wh_now) > len(wh_then):
26             return set(wh_now).difference(set(wh_then)).pop()
27
28     def test_teste01(self):
29         self.driver.get("http://localhost:8001/")
30         self.driver.set_window_size(1003, 1040)
31         self.driver.find_element(By.ID, "0101_id-inputEl").click() #<<< Apartir dessa linha
32         self.driver.find_element(By.ID, "0101_id-inputEl").send_keys("1236")
33         self.driver.find_element(By.ID, "0105_id-inputEl").send_keys("suporte")
34         self.driver.find_element(By.CSS_SELECTOR, "#0F1_id > img").click()
35         self.driver.find_element(By.ID, "button-1005-btnEl").click()
36

```

Obs. Sempre será depois da linha

`self.driver.set_window_size (1003, 1040)`

Copie as linhas depois da linha especificada a cima.

Obs. Como tem que gravar no início então selecione o lugar em questão e copie somente o teste.

```

def test_teste07F3NovoPedido(self):
    self.driver.get("http://localhost:8001/")
    self.driver.set_window_size(1936, 1056)
    self.driver.find_element(By.ID, "0101_id-inputEl").send_keys("1236")
    self.driver.find_element(By.ID, "0105_id-inputEl").send_keys("suporte")
    self.driver.find_element(By.ID, "0105_id-inputEl").send_keys(Keys.ENTER)
    self.driver.find_element(By.ID, "0105_id-inputEl").send_keys("suporte")
    self.driver.find_element(By.ID, "0105_id-inputEl").send_keys(Keys.ENTER)
    self.driver.find_element(By.ID, "0111_id-inputEl").send_keys(Keys.ENTER)
    self.driver.find_element(By.ID, "0105_id-inputEl").send_keys("suporte")
    self.driver.find_element(By.ID, "button-1005-btnEl").click()
    self.vars["window_handles"] = self.driver.window_handles
    self.driver.find_element(By.ID, "043_id-btnIconEl").click()
    self.vars["win3826"] = self.wait_for_window(2000)
    self.driver.switch_to.window(self.vars["win3826"])
    self.driver.find_element(By.ID, "tool-1265-toolEl").click()
    self.driver.find_element(By.ID, "07D_id-btnInnerEl").click()
    self.driver.find_element(By.ID, "01681_id-inputEl").click()
    self.driver.find_element(By.ID, "0279_id").click() #<<< Começo Aqui
    self.driver.find_element(By.ID, "01BB5_id").click()
    self.driver.find_element(By.CSS_SELECTOR, ".x-grid-item-focused > .x-grid-cell-inner").click()
    self.driver.find_element(By.ID, "button-1007-btnInnerEl").click()

```

II. Abra o arquivo de molde

Abra o arquivo usando o Notepad++

\\Serversat\sat\Testes\Selenium\Modelo Teste.py

Obs. toda vez que for usar. Selecione duplique e usa para que o roteiro não fique errado.

Insira as linhas copiada no lugar selecionado (Inicio do Script)

```

class SATTeste():
    def InitDrive(self):
        chrome_options = Options()
        chrome_options.add_experimental_option("debuggerAddress", "127.0.0.1:1111")
        chrome_driver = "C:\ChromeDrive\chromedriver.exe"
        self.driver = webdriver.Chrome(chrome_driver, chrome_options=chrome_option
        self.vars = {}

    def StartTest(self):
        self.driver.execute_script("window.stop();");

#### Inicio do script #####

# Objeto SATTeste é criado da classe SATTeste
SATTeste = SATTeste()
# Driver é iniciado
SATTeste.InitDrive()
# Inicia o teste
SATTeste.StartTest()

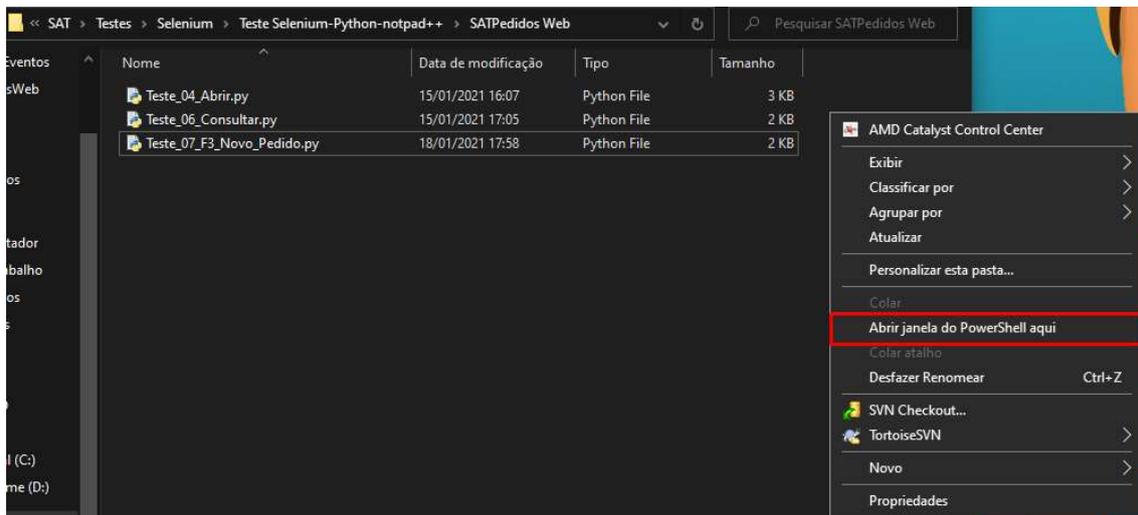
```

7. Realizando o teste.

I. Abra o PowerShell

Para dar certo clique com o (botão direito + shift) na pasta onde estar salvou os projetos e clique **“Abrir janela do PowerShell Aqui”**

Obs. nenhum item da pasta deve estar selecionado, (botão direito + shift) deve ser feito na parte vazia da tela.



II. Realizando o teste

Digite o comando `python.exe .\Teste_04_Abrir.py`

Pronto agora assista seu teste sendo feito.

Dica quando estiver cheiro PowerShell rode o comando (Clear)