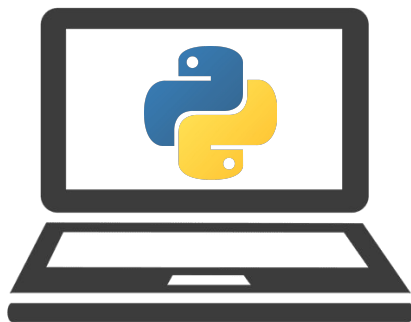




AUTOMATIZACIÓN DE TAREAS CON SELENIUM

2019



Hola!

Soy Carolina Gómez⁷

Security Developer en Fluid Attacks
Co - organizadora Django Girls Colombia y Python
Pereira



@caro_gomezt



@carogomezt



<https://www.linkedin.com/in/carolinagomezt>

/



cgomez@fluidattacks.com



Propósito

Encontrar **Todas** las Vulnerabilidades y
Reportarlas lo **Antes Posible**



Caso Fluid Attacks⁷

- Pruebas de regresión
- Generación de llaves de acceso
- Validación de ambientes



¿Cuándo automatizar?⁷

- Pruebas de regresión
- Tareas repetitivas en un navegador
- Validaciones de muchos campos de entrada

¿Qué es Selenium?

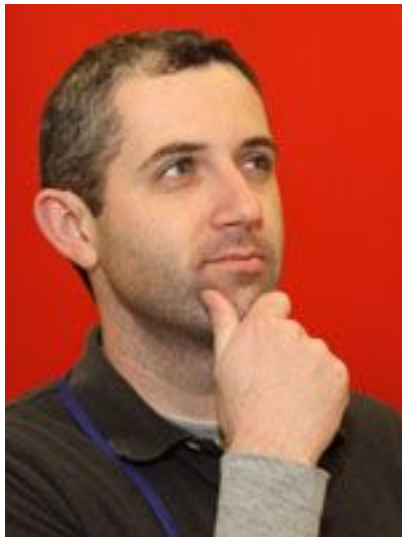


Selenium

Herramienta open source para la automatización de tareas en un navegador.

Imagen tomada de [hackernoon](https://hackernoon.com/selenium-what-is-it-and-how-to-use-it)

Historia⁷



Jason Huggins

Imagen tomada de [flourishconf](https://flourishconf.com/)





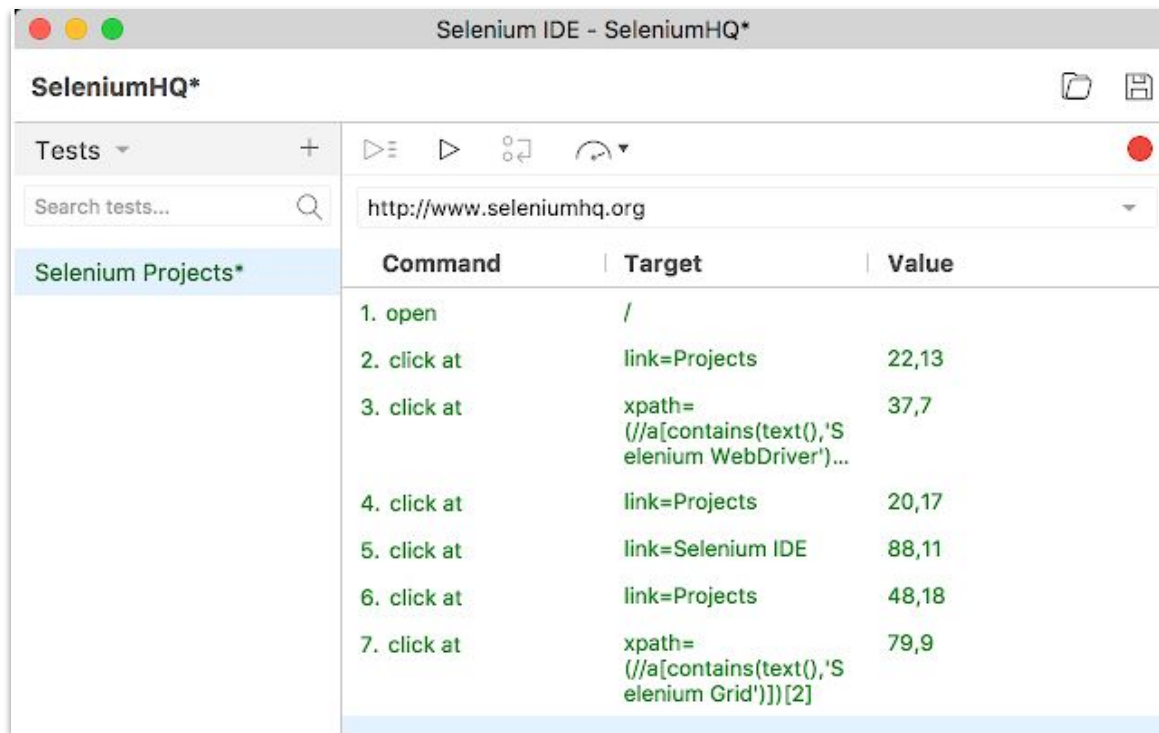
Herramientas⁷

**Selenium
IDE**

**Selenium
Web
Driver**

**Selenium
Grid**

Selenium IDE⁷



SeleniumHQ*

Tests ▾ +

Search tests... 🔍

http://www.seleniumhq.org ▾

Command	Target	Value
1. open	/	
2. click at	link=Projects	22,13
3. click at	xpath=	37,7
	//*[@contains(text(),'Selenium WebDriver')]	
4. click at	link=Projects	20,17
5. click at	link=Selenium IDE	88,11
6. click at	link=Projects	48,18
7. click at	xpath=	79,9
	//*[@contains(text(),'Selenium Grid'))][2]	

Ejemplo Selenium IDE

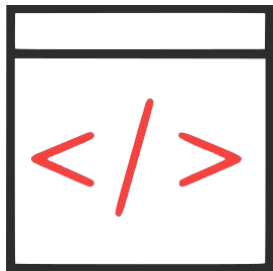
The screenshot displays the Selenium IDE interface. On the left is a browser window showing a scenic image of a lake with sailboats. The main panel on the right shows a test suite titled 'Python Colombia Test*'. The test suite is currently 'Executing'. Below the suite name, there is a table of test steps:

	Command	Target	Value
1	open	https://www.google.com/	
2	type	xpath=//input[@name='q']	PyCon Colombia 2019
3	click	css=div.FPdoLc.VicLAe > center > input[name='btnK']	
4	assert element present	css=cite.iUh30	https://www.pycon.co/
5	click	css=cite.iUh30	
6	wait for element not visible	css=div.FPdoLc.VicLAe > center > input[name='btnK']	50

Below the table, there is a section for the current step (click) with fields for Command, Target, Value, and Description. The Command is 'click', the Target is 'css=button.navbar-toggler', and the Value is empty. The Description field is also empty.

At the bottom of the interface, there is a status bar showing 'Runs: 1' and 'Failures: 0'. There are also tabs for 'Log' and 'Reference'.

Selenium Web Driver⁷



Test Script

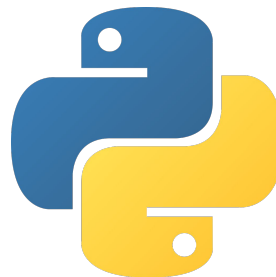
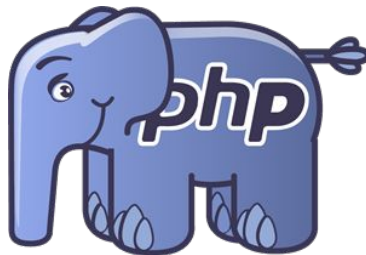
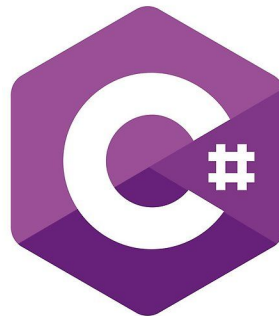


Web Driver



Navegador

Test Script⁷



Navegadores⁷



**Google
Chrome**



Firefox



**Internet
Explorer**



Opera



Safari



**Microsoft
Edge**

Localización de Elementos⁷

- id
- name
- xpath
- link text
- partial link text
- tag name
- class name
- css selector

Instancia Web Driver⁷

```
1 from selenium import webdriver
2 from selenium.webdriver.firefox.options import Options
3
4
5 def driver_initialize(url):
6     """
7     Starts a Firefox driver in the defined URL in headless mode
8
9     :param url: URL to load once the driver starts
10    """
11    options = Options()
12    options.headless = True
13    driver = webdriver.Firefox(options=options)
14    driver.get(url)
15    return driver
```

Selenium Grid¹

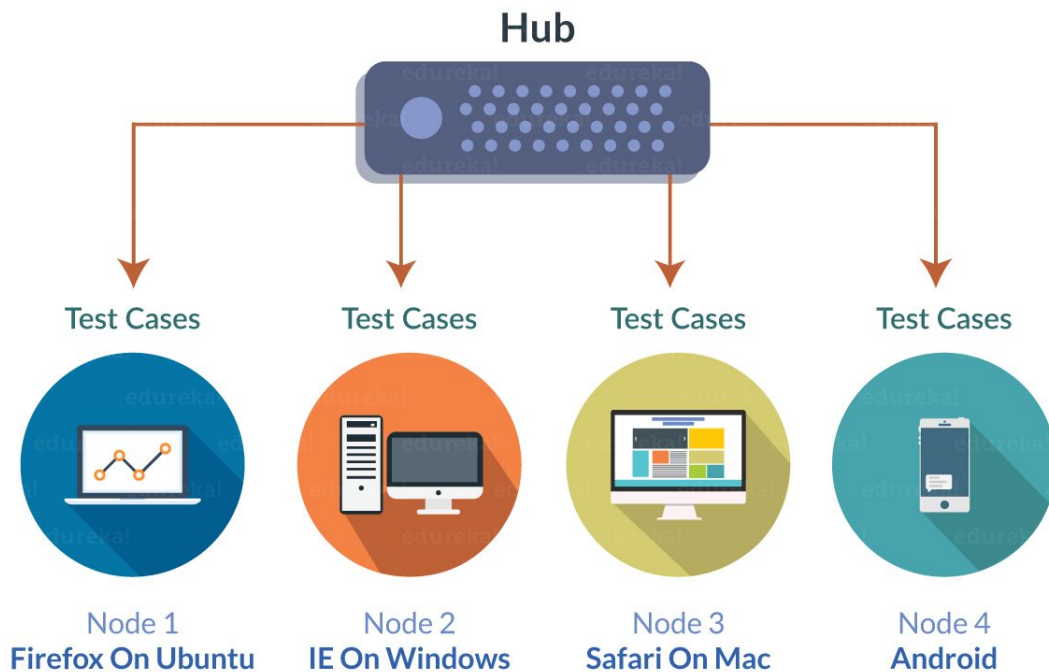
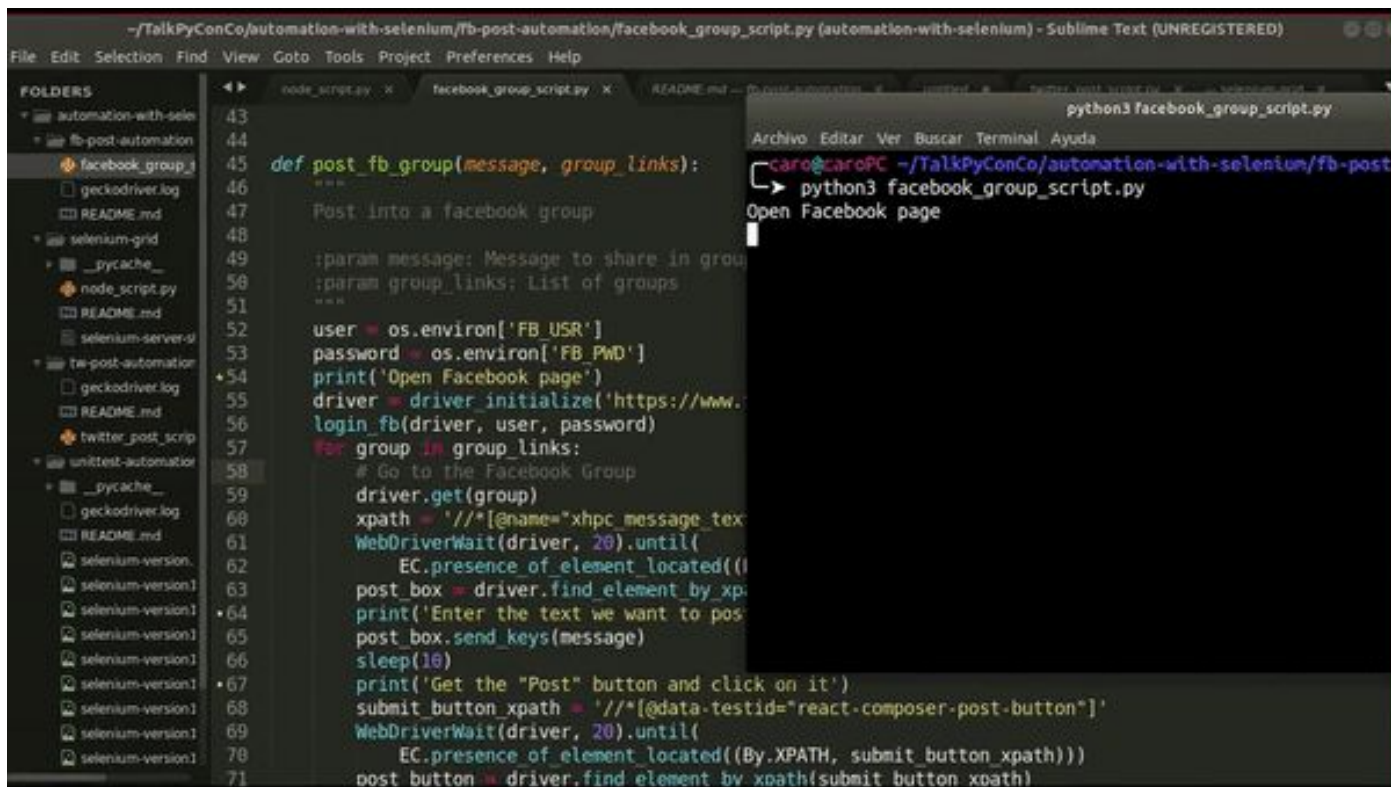


Imagen tomada de [edureka](https://www.edureka.io/)

1. Casos prácticos del uso de Selenium

Automatización de publicaciones en Facebook⁷



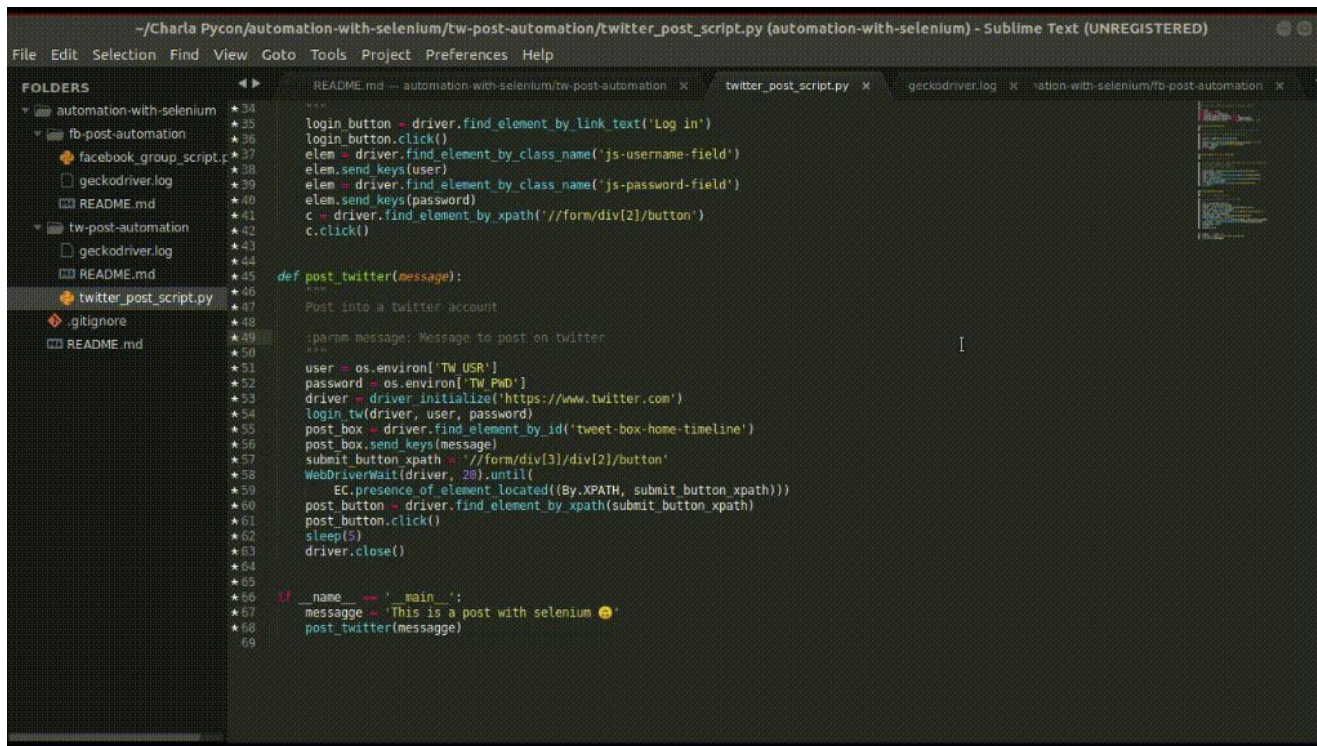
```
~/TalkPyConCo/automation-with-selenium/Fb-post-automation/facebook_group_script.py (automation-with-selenium) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
+ automation-with-selenium
+ fb-post-automation
  + facebook_group_1
    geckodriver.log
    README.md
  selenium-grid
  + _pycache_
  node_script.py
  README.md
  selenium-server-9
+ fb-post-automation
  geckodriver.log
  README.md
  twitter_post_script.py
+ unittest-automation
  + _pycache_
  geckodriver.log
  README.md
  selenium-version.
  selenium-version.1
  selenium-version.1
  selenium-version.1
  selenium-version.1
  selenium-version.1
  selenium-version.1
  selenium-version.1
  selenium-version.1
  selenium-version.1
  selenium-version.1

43
44
45 def post_fb_group(message, group_links):
46     """
47     Post into a facebook group
48
49     :param message: Message to share in group
50     :param group_links: List of groups
51     """
52     user = os.environ['FB_USER']
53     password = os.environ['FB_PWD']
54     print('Open Facebook page')
55     driver = driver_initialize('https://www.
56     login_fb(driver, user, password)
57     for group in group_links:
58         # Go to the Facebook Group
59         driver.get(group)
60         xpath = '//*[@name="xhpc message text"]'
61         WebDriverWait(driver, 20).until(
62             EC.presence_of_element_located((
63                 By.XPATH, xpath)
64         )
65         post_box = driver.find_element_by_xpath(xpath)
66         print('Enter the text we want to post')
67         post_box.send_keys(message)
68         sleep(10)
69         print('Get the "Post" button and click on it')
70         submit_button_xpath = '//*[@data-testid="react-composer-post-button"]'
71         WebDriverWait(driver, 20).until(
72             EC.presence_of_element_located((By.XPATH, submit_button_xpath)))
73         post_button = driver.find_element_by_xpath(submit_button_xpath)
74         post_button.click()
75
python3 facebook_group_script.py
Archivo Editar Ver Buscar Terminal Ayuda
caro@caroPC ~/TalkPyConCo/automation-with-selenium/fb-post-automation:
python3 facebook_group_script.py
Open Facebook page
```

Video ejemplo [automatización en Facebook](#)

Automatización de publicaciones en Twitter⁷



```
~/Charla Pycon/automation-with-selenium/tw-post-automation/twitter_post_script.py (automation-with-selenium) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
  automation-with-selenium
  fb-post-automation
  facebook_group_script.py
  geckodriver.log
  README.md
  tw-post-automation
  geckodriver.log
  README.md
  twitter_post_script.py
  .gitignore
  README.md

34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69

login_button = driver.find_element_by_link_text('Log in')
login_button.click()
elem = driver.find_element_by_class_name('js-username-field')
elem.send_keys(user)
elem = driver.find_element_by_class_name('js-password-field')
elem.send_keys(password)
c = driver.find_element_by_xpath('//form/div[2]/button')
c.click()

def post_twitter(message):
    """
    Post into a twitter account
    """
    iparams message: Message to post on twitter
    """
    user = os.environ['TW_USR']
    password = os.environ['TW_PWD']
    driver = driver_initialize('https://www.twitter.com')
    login_tw(driver, user, password)
    post_box = driver.find_element_by_id('tweet-box-home-timeline')
    post_box.send_keys(message)
    submit_button_xpath = '//form/div[3]/div[2]/button'
    WebDriverWait(driver, 20).until(
        EC.presence_of_element_located((By.XPATH, submit_button_xpath)))
    post_button = driver.find_element_by_xpath(submit_button_xpath)
    post_button.click()
    sleep(5)
    driver.close()

if __name__ == '__main__':
    message = 'This is a post with selenium 🐍'
    post_twitter(message)
```

Video ejemplo [automatización en Twitter](#)

Pruebas de sistemas

```
caro@caroPC ~/TalkPyConCo/automation-with-selenium/unittest-automation <master*>
└─> python3 -m unittest -v unittest_automation_script.py 1
test_search_by_text (unittest_automation_script.SearchVersion) ... ok
test_selenium_last_release (unittest_automation_script.SearchVersion) ... FAIL
test_selenium_version (unittest_automation_script.SearchVersion) ... ok

=====
```

Video ejemplo [unittest](#)

Ejemplo Selenium Grid⁷

```
caro@caroPC ~/TalkPyConCo/automation-with-selenium/selenium-grid
```

```
9 class SearchVersion(unittest.TestCase):
10     PLATFORM = 'LINUX'
11     BROWSER = 'firefox'
12
13     def setUp(self):
14         desired_caps = {}
15         desired_caps['platform'] = self.PLATFORM
16         desired_caps['browserName'] = self.BROWSER
17         self.driver = webdriver.Remote(
18             command_executor='http://localhost:4444/wd/hub',
19             desired_capabilities=desired_caps)
```

Video ejemplo [selenium grid](#)

Software as a service con⁷ selenium



Automatizar la explotación o cierre de vulnerabilidades

- Falta de validación de datos
 - Enumeración de usuarios
 - Denegación de servicio
 - Cross-site scripting (XSS)
 - Inyección SQL

Ventajas

- Soporta múltiples sistemas operativos y navegadores
- Soporta pruebas en paralelo
- Es gratuito

Desventajas⁷

- Puede ser lento
- No genera informes automáticos
- Sólo se pueden probar aplicaciones web



¡Gracias!
¿Alguna Pregunta?

<https://github.com/carogomezt/automation-with-selenium>



¿El entorno de ejecución siempre requiere java?

- Utilizando solo web driver no
- Con selenium grid es necesario java para ejecutar el servidor

**¿Podemos automatizar⁷
navegadores en dispositivos
móviles?**



¿Se puede ejecutar selenium en un ambiente sin interfaz gráfica?

```
5  from selenium import webdriver
6
7
8  def driver_initialize(url):
9      """
10         Starts a Firefox driver in the defined URL in headless mode
11
12         :param url: URL to load once the driver starts
13         """
14         options = webdriver.FirefoxOptions()
15         options.add_argument('--headless')
16         driver = webdriver.Firefox(options=options)
17         driver.get(url)
18         return driver
```

¿Qué otras opciones existen en el mercado?



<https://idatalabs.com/tech/products/selenium>