

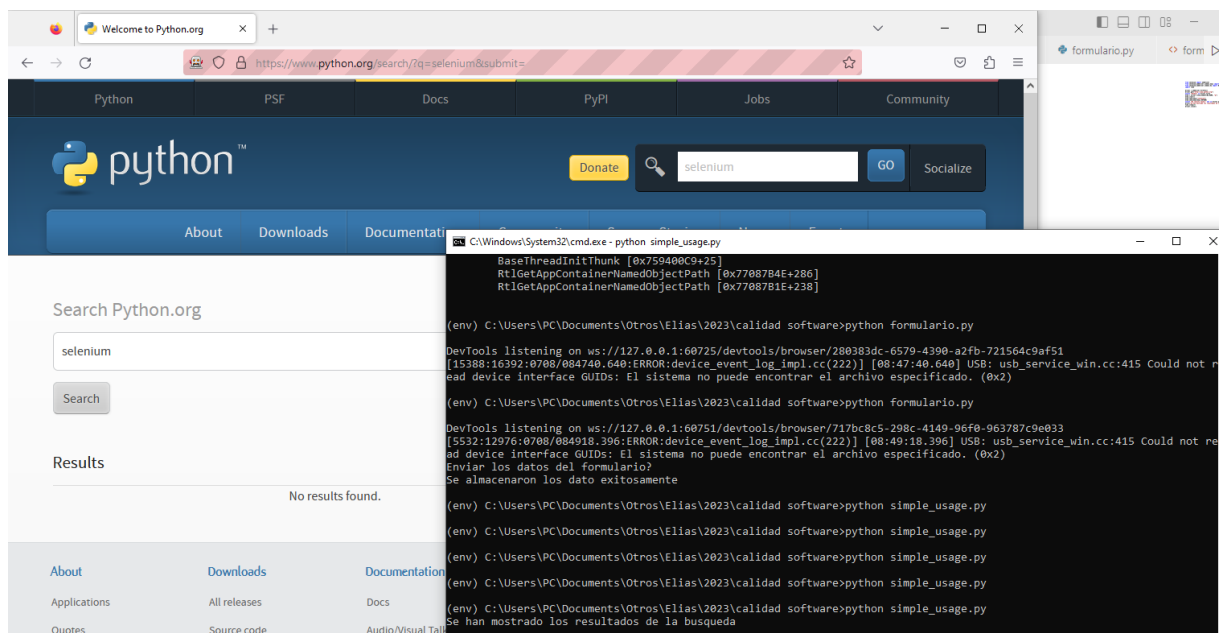
## Selenium

Herramienta utilizada para la automatización de pruebas en aplicaciones web.

Un uso simple:

En este caso se usa selenium para abrir una url en Firefox y buscar en el titulo la palabra Python, para posteriormente, buscar selenium dentro de la página de Python.

```
simple_usage.py > ...
1  from selenium import webdriver
2  from selenium.webdriver.common.keys import Keys
3  from selenium.webdriver.common.by import By
4  import time
5
6  driver = webdriver.Firefox()
7  driver.get("http://www.python.org")
8  assert "Python" in driver.title
9  elem = driver.find_element(By.NAME, "q")
10 elem.clear()
11 elem.send_keys("selenium")
12 elem.send_keys(Keys.RETURN)
13 assert "No results found." not in driver.page_source
14 print("Se han mostrado los resultados de la búsqueda")
15 time.sleep(3)
16 driver.close()
```



Test unitarios:

Selenium también se puede usar para test unitarios

En este caso, se usan 2 navegadores, uno para acceder a la web de Python desde Firefox, y el otro test para acceder a Google.com desde Chrome y hacer una búsqueda de selenium:

```
simple_unit_test.py > PythonOrgSearch > test_search_in_python_org
1  import unittest
2  from selenium import webdriver
3  from selenium.webdriver.common.keys import Keys
4  from selenium.webdriver.common.by import By
5  import time
6
7  class PythonOrgSearch(unittest.TestCase):
8
9      def setUp(self):
10         self.driver = webdriver.Firefox()
11
12     def test_search_in_python_org(self):
13         driver = self.driver
14         driver.get("http://www.python.org")
15         self.assertIn("Python", driver.title)
16         time.sleep(3)
17         self.assertNotIn("No results found.", driver.page_source)
18
19     def test_search_in_python_org2(self):
20         driver = self.driver
21         driver.get("http://www.google.com")
22         self.assertIn("Google", driver.title)
23         elem = driver.find_element(By.NAME, "q")
24         elem.send_keys("selenium")
25         elem.send_keys(Keys.RETURN)
26         time.sleep(3)
27         self.assertNotIn("No results found.", driver.page_source)
28
29
30     def tearDown(self):
31         self.driver.close()
32
33 if __name__ == "__main__":
34     unittest.main()
```

Al ejecutar el código se obtuvo el siguiente mensaje de éxito al ejecutar las pruebas unitarias:

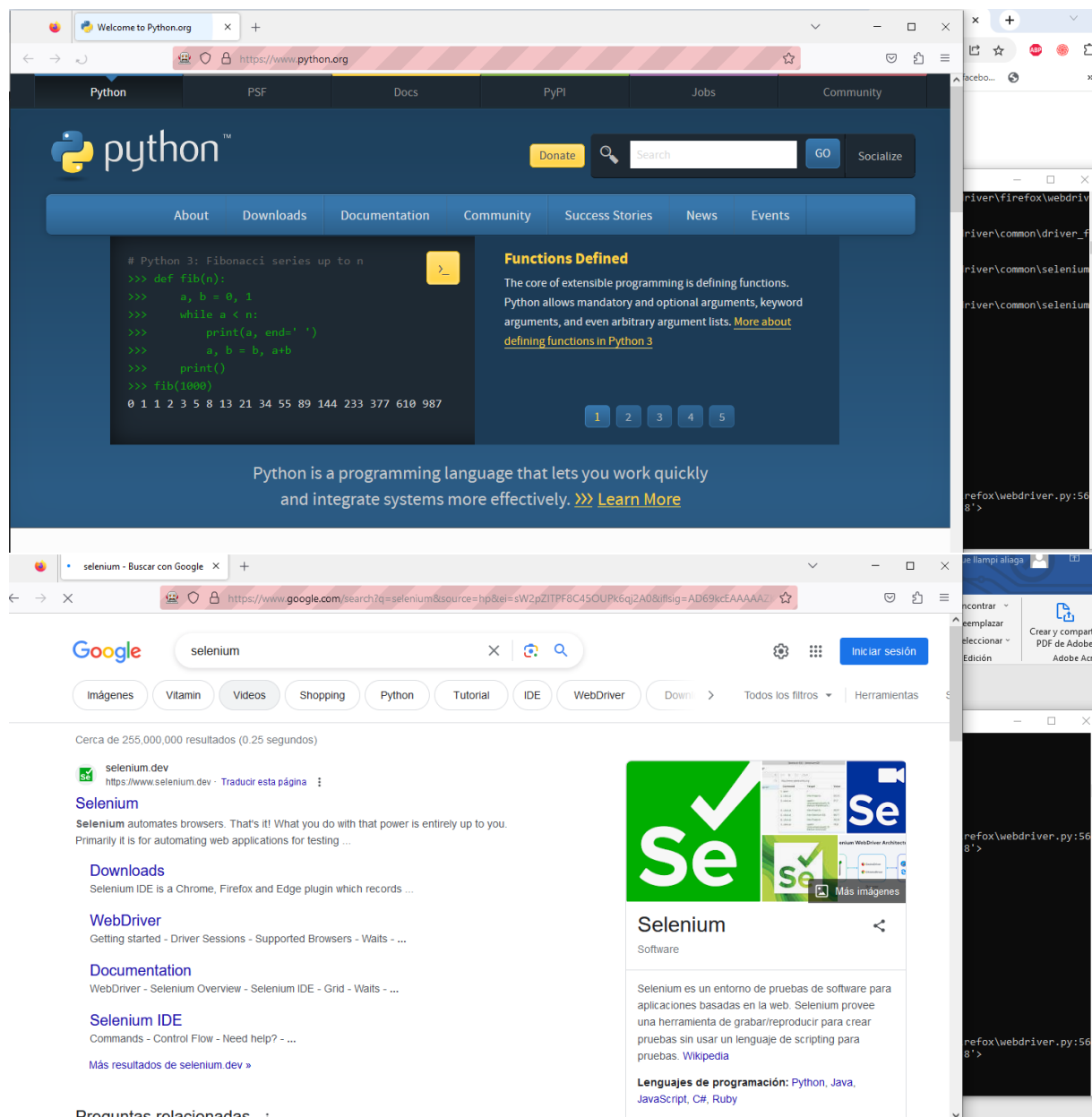
```
.....
Ran 2 tests in 23.067s

FAILED (failures=1)

(env) C:\Users\PC\Documents\Otros\Elias\2023\calidad software>python simple_unit_test.py
C:\Users\PC\Documents\Otros\Elias\2023\calidad software\env\lib\site-packages\selenium\webdriver\firefox\webdriver.py:56:
ResourceWarning: unclosed file <_io.TextIOWrapper name='geckodriver.log' mode='a+' encoding='utf-8'>
  self.service = service if service else Service()
ResourceWarning: Enable tracemalloc to get the object allocation traceback
..
.....
Ran 2 tests in 25.158s

OK

(env) C:\Users\PC\Documents\Otros\Elias\2023\calidad software>
```



Ejemplo práctico: rellenar datos de un formulario:

Selenium permite realizar tareas repetitivas, como la de rellenar datos en un formulario y confirmar el envío.

En este caso, se tiene un formulario simple en html, con campos de nombre, apellido y correo.

Además, al hacer clic en enviar, aparecerá un mensaje de confirmación.

### Formulario

Nombre:

Apellido:

Correo electrónico:

C:\Users\PC\Documents\Otros\elias\2023\carrada72050software/formulario.html

ITING 3C Eachine H8 Mini He... caminos.udc.es/info... Curso de Analítica... https://www.facebo...

### Formulario

Nombre:

Apellido:

Correo electrónico:

Esta página dice

Enviar los datos del formulario?

Para realizar esta tarea de manera automática, se implementa en python el siguiente código:

```
formulario.py > ...
1  from selenium import webdriver
2  from selenium.webdriver.common.by import By
3  from selenium.webdriver.common.keys import Keys
4  from selenium.webdriver.common.alert import Alert
5  import time
6
7
8  driver = webdriver.Chrome()
9
10 driver.get("C:\\Users\\PC\\Documents\\Otros\\Elias\\2023\\calidad software\\formulario.html")
11 nombre_input = driver.find_element(By.ID, "nombre")
12 nombre_input.clear()
13 nombre_input.send_keys("Elias")
14
15
16 apellido_input = driver.find_element(By.ID, "apellido")
17 apellido_input.clear()
18 apellido_input.send_keys("Llampi")
19
20 correo_input = driver.find_element(By.ID, "correo")
21 correo_input.clear()
22 correo_input.send_keys("correo@gmail.com")
23
24 time.sleep(3)
25
26 confirmar = driver.find_element(By.ID, "confirm")
27 confirmar.click()
28
29 time.sleep(2)
30
31 alert = Alert(driver)
32
33 print(alert.text)
34 print ("Se almacenaron los dato exitosamente")
```

Esto rellena los campos del formulario con datos de prueba, luego da clic en el botón de enviar y finalmente confirma en el popup que aparece en ventana.