Manual Splinter

Prerrequisitos

- Debe tener configurado y creado el proyecto miblog según el documento django-from-scratch. De no tenerlo, también puede descargar el proyecto finalizado del siguiente repositorio GIT:
- https://github.com/ulima-is2/pygames/tree/master/splinter
- (original https://github.com/ulima-is2/blog-django)
- Tener creado y activado un virtualenvironment.

Instalación

Seguiremos la documentación inicial de https://splinter.readthedocs.io/en/latest/
Desde un shell, instalar la librería splinter

```
$ py -m pip install splinter
```

Para usar Google Chrome, descargar el Chrome Driver desde
https://sites.google.com/a/chromium.org/chromedriver/downloads
P.e. Descargamos para windows
https://chromedriver.storage.googleapis.com/2.30/chromedriver_win32.zip
Descomprimimos y colocamos en la ruta donde está instalado Google Chrome
P.e. C:\Program Files (x86)\Google\Chrome
En caso no se tenga permisos, ubicar el archivo chromedriver.exe en una carpeta cualquiera.

Copiar el siguiente código a un archivo test.py

```
from splinter import Browser

executable_path = {'executable_path':'C:\Program Files (x86)\Google\Chrome\chromedriver.exe'}
with Browser('chrome', **executable_path) as browser:
    # Visit URL
    url = "http://www.google.com"
    browser.visit(url)
    browser.fill('q', 'splinter - python acceptance testing for web applications')
    # Find and click the 'search' button
    button = browser.find_by_name('btnG')
    # Interact with elements
    button.click()
    if browser.is_text_present('splinter.readthedocs.io'):
        print("Yes, the official website was found!")
    else:
        print("No, it wasn't found... We need to improve our SEO techniques")
```

Archivo test.py

El executable_path debe ser la ruta donde se ubica el chromedriver.exe

Ejecutamos en consola

```
$ py test.py
```

Comprobamos

Pruebas de aceptación

Para realizar una prueba de aceptación con Splinter, probaremos ingresar al administrador de Django, para ello creamos al archivo test_admin.py

```
from splinter import Browser

executable_path = {'executable_path':'C:\Program Files (x86)\Google\Chrome\chromedriver.exe'}
with Browser('chrome', **executable_path) as browser:
    # Visit URL
    url = "http://localhost:8000/admin"
    browser.visit(url)
            #test login
    browser.fill("username", "admin")
    browser.fill("password", "admin12345")
    button = browser.find_by_xpath('//*[@id="login-form"]/div[3]/input')
    button.click()

    if browser.is_text_present('Django administration'):
        print("Login: Success")
    else:
        print("Login: Error")

    browser.click_link_by_partial_href('/admin/posts/usuario/add/')
    browser.fill("nombre", "satoshi")
    browser.fill("usuario", "satoshi")
    browser.fill("password", "satoshi")
    button = browser.find_by_xpath('//*[@id="usuario_form"]/div/div/input[1]')
    button.click()

    if browser.is_text_present('was added successfully'):
        print("Create Usuario: Success")
    else:
        print("Create Usuario: Error")

    input('Waiting a key...')
```

Archivo test_admin.py

Considerando que la clave del administrador es admin12345
(o si no, cambiar usando python manage.py changepassword admin)
Comprobamos que se ingresa al administrador y se crea un nuevo usuario.