

MANUAL DE INSTALACIÓN PARA PRUEBAS AUTOMATIZADAS



ÍNDICE

¿Qué es Robot Framework?	3
Herramientas a utilizar para la práctica:	3
Instalación de Python	4
Instalación de PyCharm	7
Instalación de Robot Framework	12
Instalación de los WebDrivers	14
Creación del Proyecto de Pruebas	20
Instalación del plugin de Intellibot	23
Verificación de la Instalación	26

¿Qué es Robot Framework?

Es un framework de automatización de pruebas de desarrollo dirigido y de automatización de procesos robóticos. Es de código abierto y está basado en Python, permite hacer pruebas a aplicaciones distribuidas y muy diversas. El entorno que se ha creado está compuesto por múltiples librerías y herramientas ya construidas.

Robot Framework tiene una sintaxis sencilla, utilizando palabras clave (keywords) fáciles de entender por humanos.

Herramientas a utilizar para la práctica:

- PyCharm
- Robot Framework
- Librería Selenium

Instalación de Python

Ya que el framework está basado en el lenguaje Python, es necesario tenerlo instalado para poder trabajar con él.

Python puede ser descargado desde su [página oficial](#). Se descargará la última versión disponible que se ajuste a su sistema operativo:

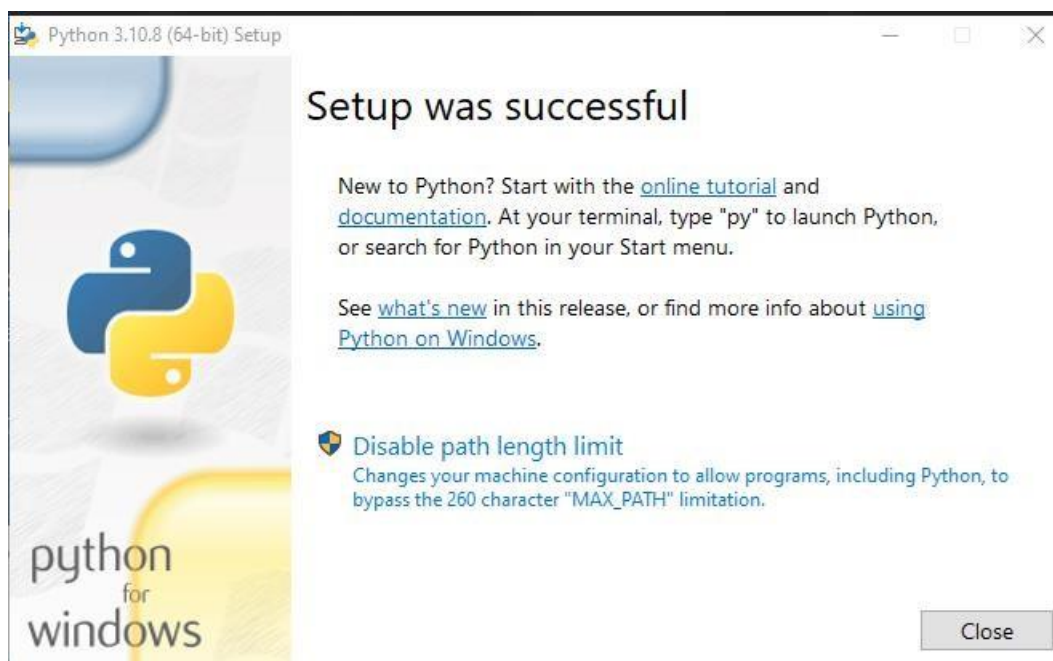


Una vez descargado se procede a ejecutar el instalador.

Cuando comience la instalación es importante seleccionar la opción de “Add python.exe to PATH” para que el sistema operativo pueda detectarlo sin importar la carpeta donde se encuentre.



Clic en “Close” para cerrar el instalador.



Una vez concluida la instalación se puede comprobar desde la terminal emitiendo el comando:

```
python --version
```

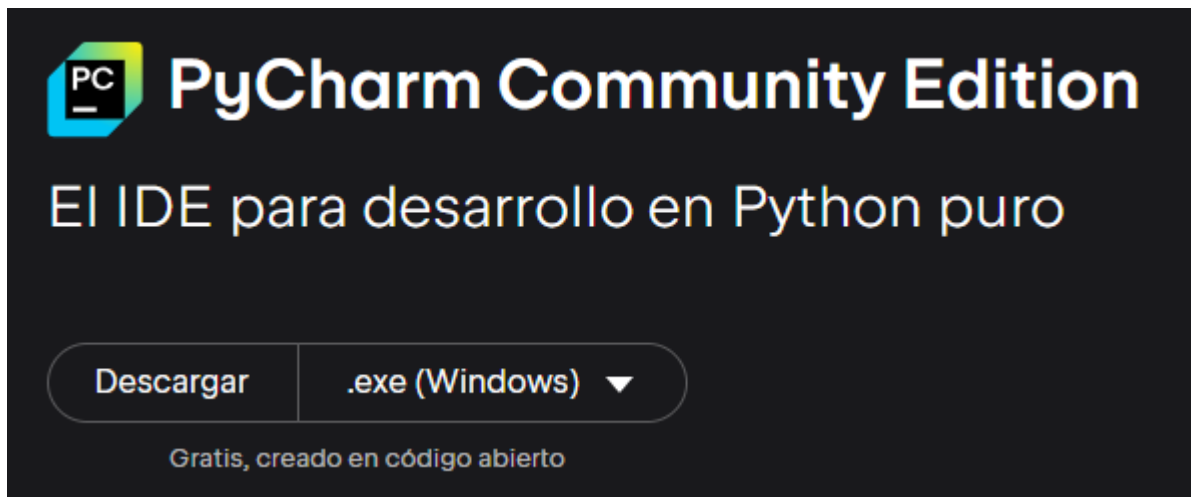
Y aparecerá como resultado la versión que acabamos de instalar:

```
C:\Users\jgzarate>python --version  
Python 3.11.4
```

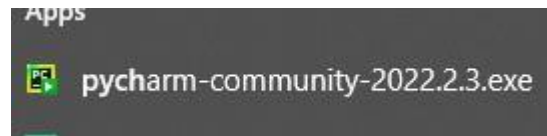
Instalación de PyCharm

Como se trabajará con Python el IDE que se utilizará es PyCharm Community Edition.

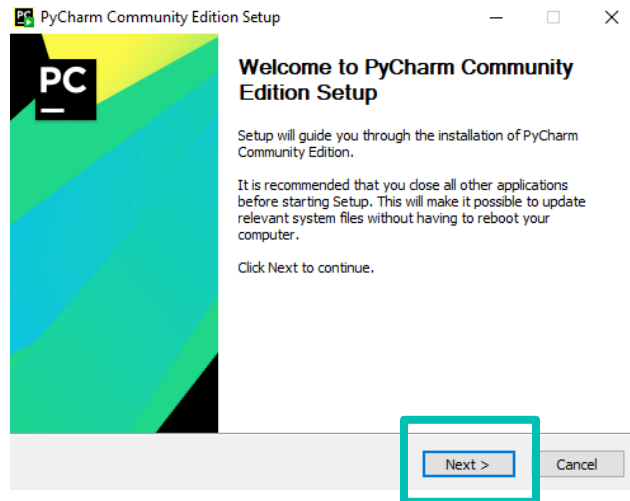
El IDE puede descargarse desde la [página web oficial de JetBrains](#), la compañía desarrolladora, ya sea la versión profesional o la de comunidad que es de código Abierto. En este caso se procederá con la versión de comunidad:



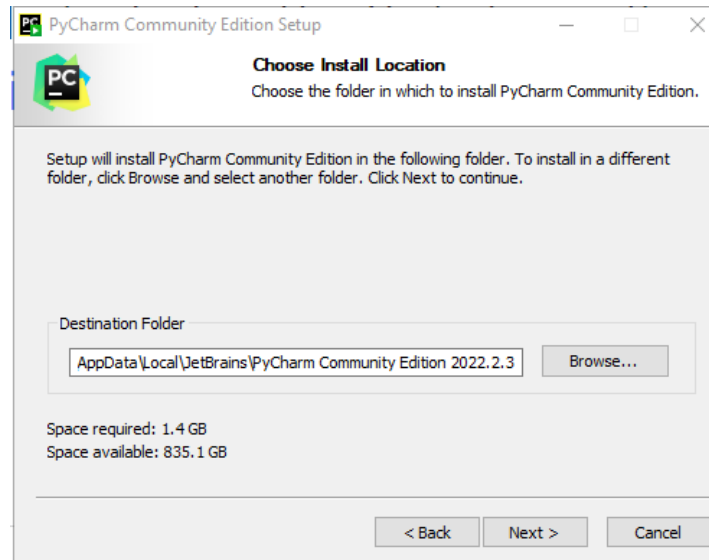
Una vez descargado el instalador se procederá a ejecutarlo:



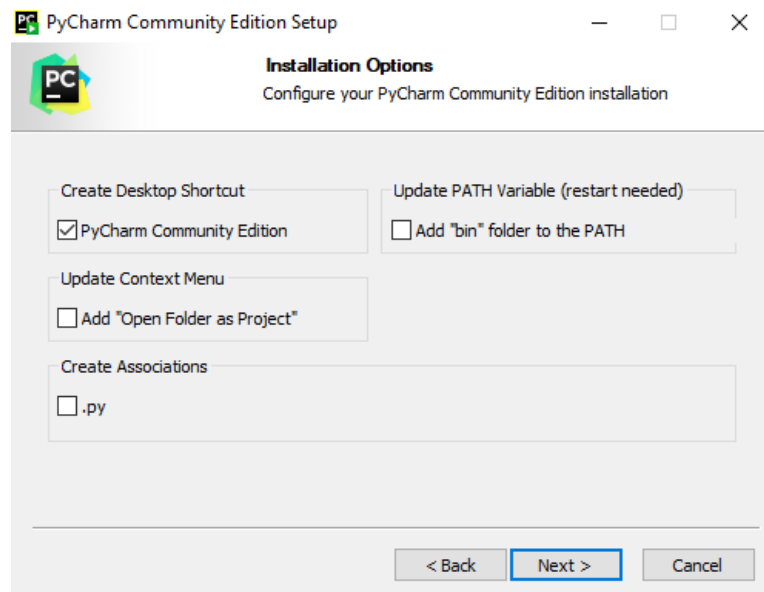
Al iniciar el instalador se seleccionará la opción de “Next” para continuar:



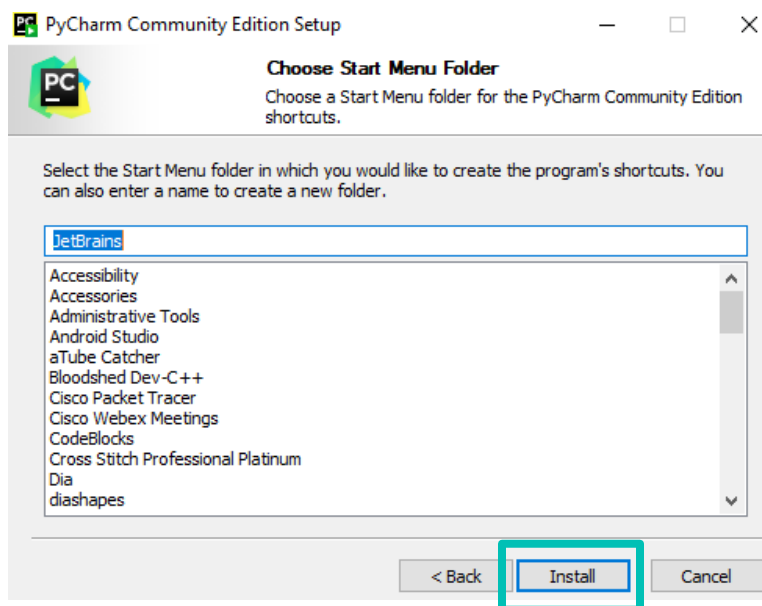
Se seleccionará la carpeta donde será instalado y se hará clic en “Next”:



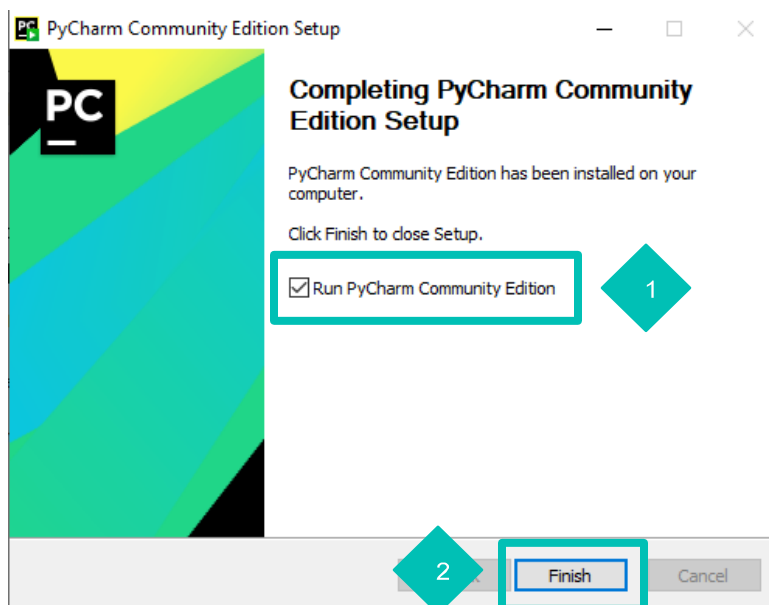
Aparecerán las opciones de instalación, se pueden seleccionar los que se deseen, pues no afecta a la práctica. Se hace clic en “Next”:



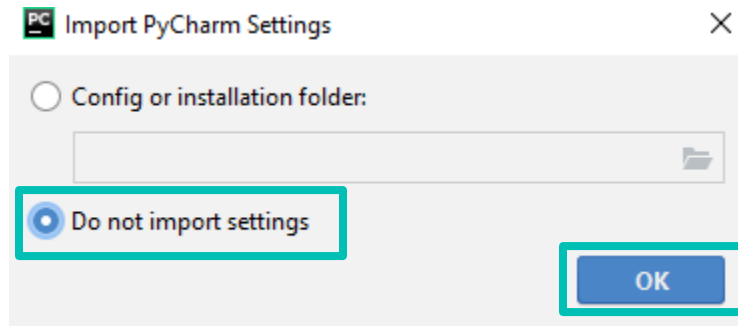
Se selecciona la opción de “Instalar”:



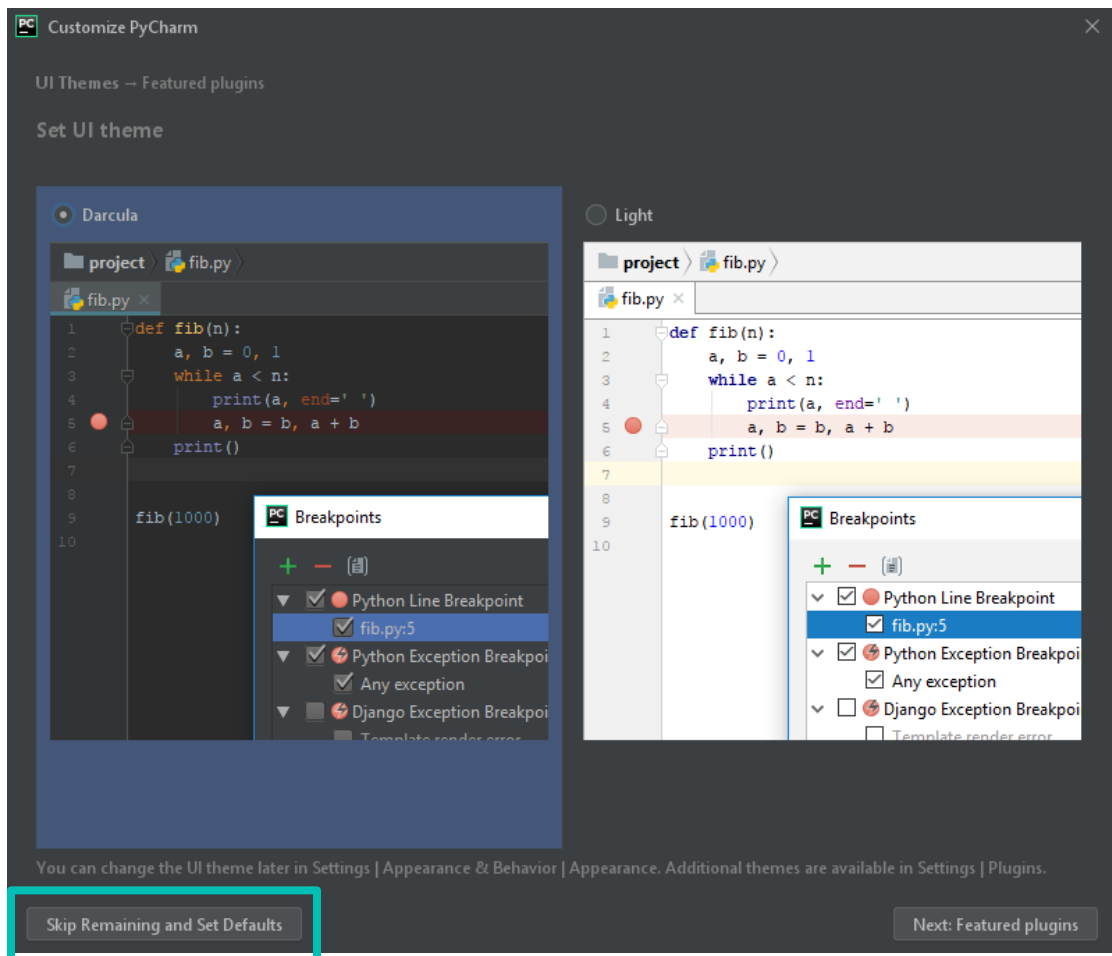
Una vez finalizada la instalación se selecciona la opción de “Finish”, se puede seleccionar la opción de “Run PyCharm Community Edition” para que el programa se ejecute al cerrar la ventana de instalación:



Se seleccionará la opción de “Do not import settings” y posteriormente “Ok”:



Por último, aparecerá la ventana que permite elegir el tema de la interfaz de usuario para trabajar en PyCharm, se selecciona el deseado y finalmente se hace clic en la opción de “Skip Remaining and Set Defaults”:

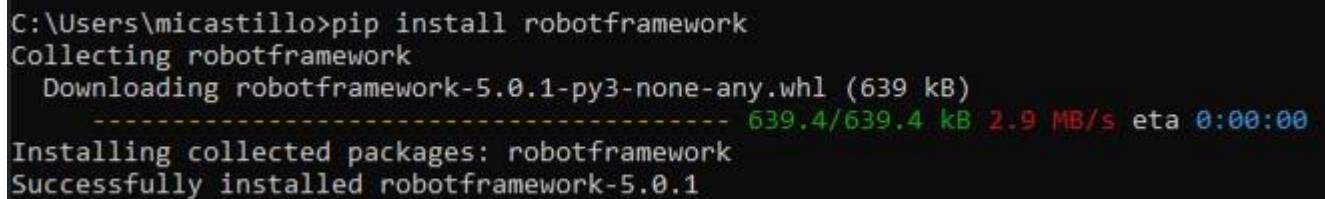


Instalación de Robot Framework

La instalación de RobotFramework es muy sencilla, se hace mediante el comando pip en la terminal.

Primero se emite el comando:

```
pip install robotframework
```

A screenshot of a Windows command prompt terminal window. The prompt is 'C:\Users\micastillo>'. The user has entered 'pip install robotframework'. The output shows 'Collecting robotframework', followed by 'Downloading robotframework-5.0.1-py3-none-any.whl (639 kB)'. A progress bar is shown with '639.4/639.4 kB' in green, '2.9 MB/s' in red, and 'eta 0:00:00' in blue. The final output is 'Installing collected packages: robotframework' and 'Successfully installed robotframework-5.0.1'.

```
C:\Users\micastillo>pip install robotframework
Collecting robotframework
  Downloading robotframework-5.0.1-py3-none-any.whl (639 kB)
----- 639.4/639.4 kB 2.9 MB/s eta 0:00:00
Installing collected packages: robotframework
Successfully installed robotframework-5.0.1
```

Instalación de RobotFramework

Después se ingresa el comando:

```
pip install robotframework-seleniumlibrary
```

Que corresponde a la librería que se va a utilizar para realizar el caso de prueba.

```

C:\Users\micastillo>pip install robotframework-seleniumlibrary
Collecting robotframework-seleniumlibrary
  Downloading robotframework_seleniumlibrary-6.0.0-py2.py3-none-any.whl (95 kB)
----- 95.2/95.2 kB 5.7 MB/s eta 0:00:00
Collecting robotframework-pythonlibcore>=2.2.1
  Downloading robotframework_pythonlibcore-3.0.0-py2.py3-none-any.whl (9.9 kB)
Requirement already satisfied: robotframework>=3.2.2 in c:\users\micastillo\appdata\local\programs\python\python
Collecting selenium>=4.0.0
  Downloading selenium-4.5.0-py3-none-any.whl (995 kB)
----- 995.2/995.2 kB 4.5 MB/s eta 0:00:00
Collecting urllib3[socks]>=1.26
  Downloading urllib3-1.26.12-py2.py3-none-any.whl (140 kB)
----- 140.4/140.4 kB 396.8 kB/s eta 0:00:00
Collecting trio>=0.17
  Downloading trio-0.22.0-py3-none-any.whl (384 kB)
----- 384.9/384.9 kB 23.4 MB/s eta 0:00:00
Collecting certifi>=2021.10.8
  Downloading certifi-2022.9.24-py3-none-any.whl (161 kB)
----- 161.1/161.1 kB 4.9 MB/s eta 0:00:00
Collecting trio-websocket>=0.9
  Downloading trio_websocket-0.9.2-py3-none-any.whl (16 kB)
Collecting async-generator>=1.9
  Downloading async_generator-1.10-py3-none-any.whl (18 kB)
Collecting exceptiongroup>=1.0.0rc9
  Downloading exceptiongroup-1.0.0rc9-py3-none-any.whl (12 kB)
Collecting outcome
  Downloading outcome-1.2.0-py2.py3-none-any.whl (9.7 kB)
Collecting cffi>=1.14
  Downloading cffi-1.15.1-cp310-cp310-win_amd64.whl (179 kB)
----- 179.1/179.1 kB 5.4 MB/s eta 0:00:00
Collecting sortedcontainers
  Downloading sortedcontainers-2.4.0-py2.py3-none-any.whl (29 kB)
Collecting idna
  Downloading idna-3.4-py3-none-any.whl (61 kB)
----- 61.5/61.5 kB ? eta 0:00:00
Collecting attrs>=19.2.0
  Downloading attrs-22.1.0-py2.py3-none-any.whl (58 kB)
----- 58.8/58.8 kB 3.0 MB/s eta 0:00:00
Collecting sniffio
  Downloading sniffio-1.3.0-py3-none-any.whl (10 kB)
Collecting wsproto>=0.14
  Downloading wsproto-1.2.0-py3-none-any.whl (24 kB)
Collecting PySocks!=1.5.7,<2.0,>=1.5.6
  Downloading PySocks-1.7.1-py3-none-any.whl (16 kB)
Collecting pycparser
  Downloading pycparser-2.21-py2.py3-none-any.whl (118 kB)
----- 118.7/118.7 kB 6.8 MB/s eta 0:00:00
Collecting h11<1,>=0.9.0
  Downloading h11-0.14.0-py3-none-any.whl (58 kB)
----- 58.3/58.3 kB ? eta 0:00:00
Installing collected packages: sortedcontainers, urllib3, sniffio, robotframework-pythonlibcore, PySocks, pycpar
tframework-seleniumlibrary
Successfully installed PySocks-1.7.1 async-generator-1.10 attrs-22.1.0 certifi-2022.9.24 cffi-1.15.1 exceptiong
ry-6.0.0 selenium-4.5.0 sniffio-1.3.0 sortedcontainers-2.4.0 trio-0.22.0 trio-websocket-0.9.2 urllib3-1.26.12 w

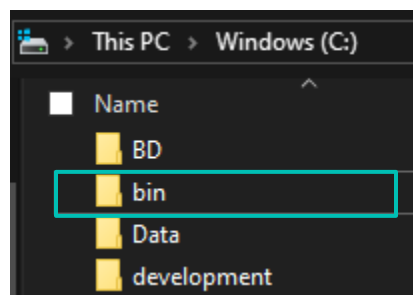
```

Instalación de librería de Selenium

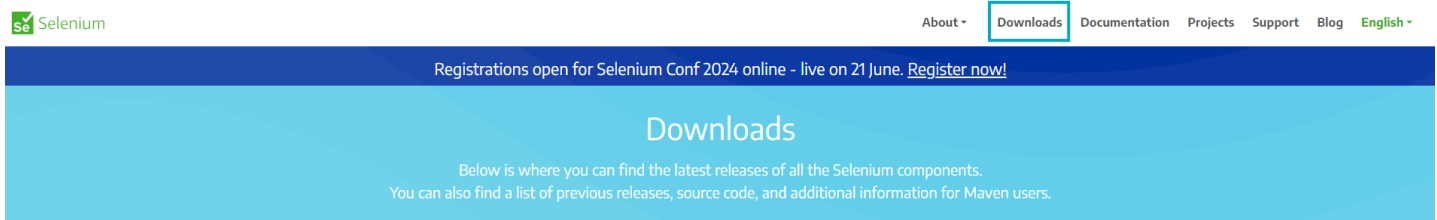
Instalación de los WebDrivers

Los casos de prueba que se realizarán en este manual serán sobre una página web, por lo que para poder ejecutarlos será necesario que se cuente con los WebDrivers del navegador que se decida utilizar, ya sea Chrome, Firefox, Edge, etcétera.

Para ello se creará una carpeta llamada bin en el disco local C:\ y dentro de ella se colocarán los WebDrivers.



Estos pueden ser encontrados desde la página web de Selenium en la sección de “Descargas”. La cual nos redirigirá a las respectivas páginas donde se encuentran los WebDrivers de los navegadores.



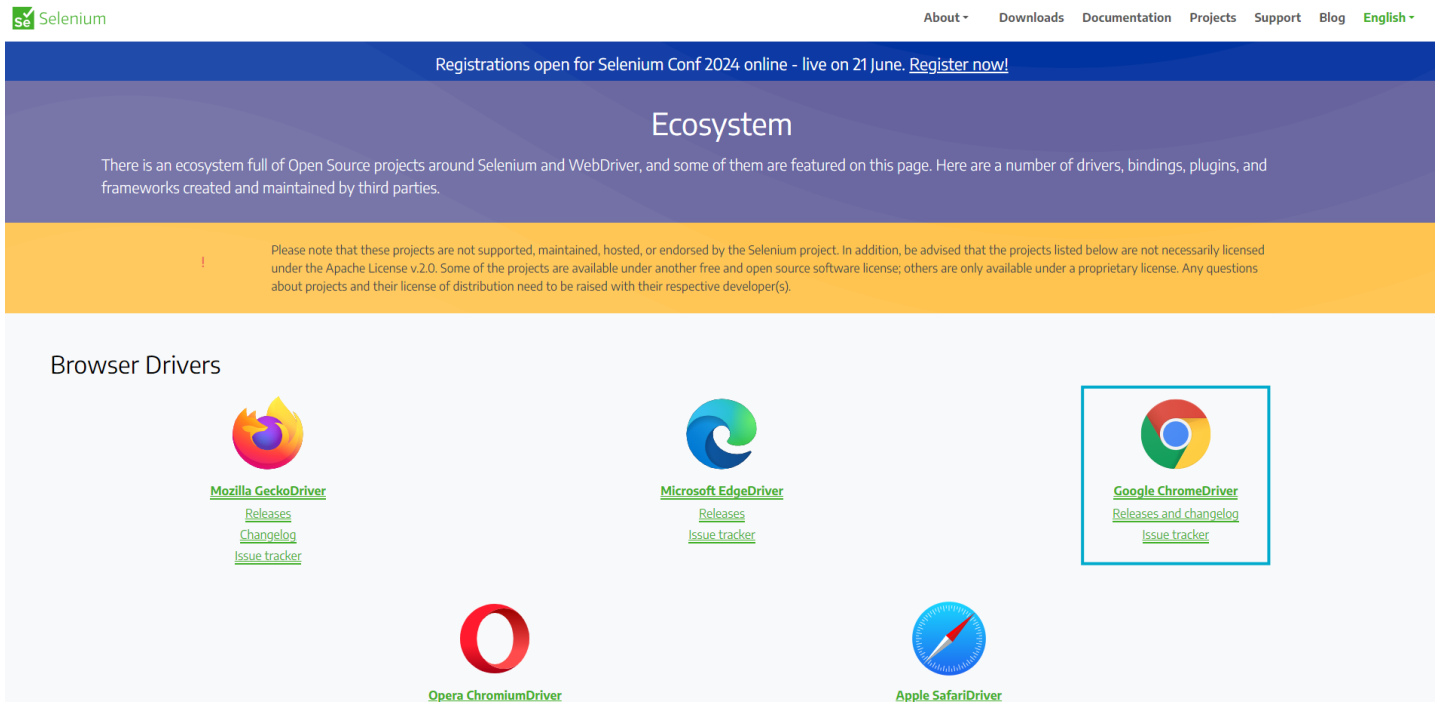
En la sección de “Selenium Clients and WebDriver Language Bindings” la cual se encuentra al inicio de la página de descargas, se dará clic en el link de “other languages exist”.

Selenium Clients and WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers.

While language bindings for [other languages exist](#), these are the core ones that are supported by the main project hosted on GitHub.

Para efectos de la práctica se utilizará Chrome. Pero se pueden descargar los drivers necesarios para pruebas futuras siguiendo los pasos en los links para cada navegador.



Selenium

About Downloads Documentation Projects Support Blog English






Registrations open for Selenium Conf 2024 online - live on 21 June. [Register now!](#)

Ecosystem

There is an ecosystem full of Open Source projects around Selenium and WebDriver, and some of them are featured on this page. Here are a number of drivers, bindings, plugins, and frameworks created and maintained by third parties.

Please note that these projects are not supported, maintained, hosted, or endorsed by the Selenium project. In addition, be advised that the projects listed below are not necessarily licensed under the Apache License v2.0. Some of the projects are available under another free and open source software license; others are only available under a proprietary license. Any questions about projects and their license of distribution need to be raised with their respective developer(s).

Browser Drivers

- 
[Mozilla GeckoDriver](#)
[Releases](#)
[Changelog](#)
[Issue tracker](#)
- 
[Microsoft EdgeDriver](#)
[Releases](#)
[Issue tracker](#)
- 
[Google ChromeDriver](#)
[Releases and changelog](#)
[Issue tracker](#)
- 
[Opera ChromiumDriver](#)
- 
[Apple SafariDriver](#)

En la sección de “Latest ChromeDriver binaries” se dará clic en el link de [“the Chrome for Testing availability dashboard”](#).

Latest ChromeDriver binaries

- Starting with M115 the latest Chrome + ChromeDriver releases per release channel (Stable, Beta, Dev, Canary) are available at [the Chrome for Testing availability dashboard](#). For automated version downloading one can use the convenient [JSON endpoints](#).
- The older releases can be found at the [Downloads](#) page.

Se irá a la sección de “Stable” y se buscará la URL del archivo binario de acuerdo con nuestro sistema operativo.

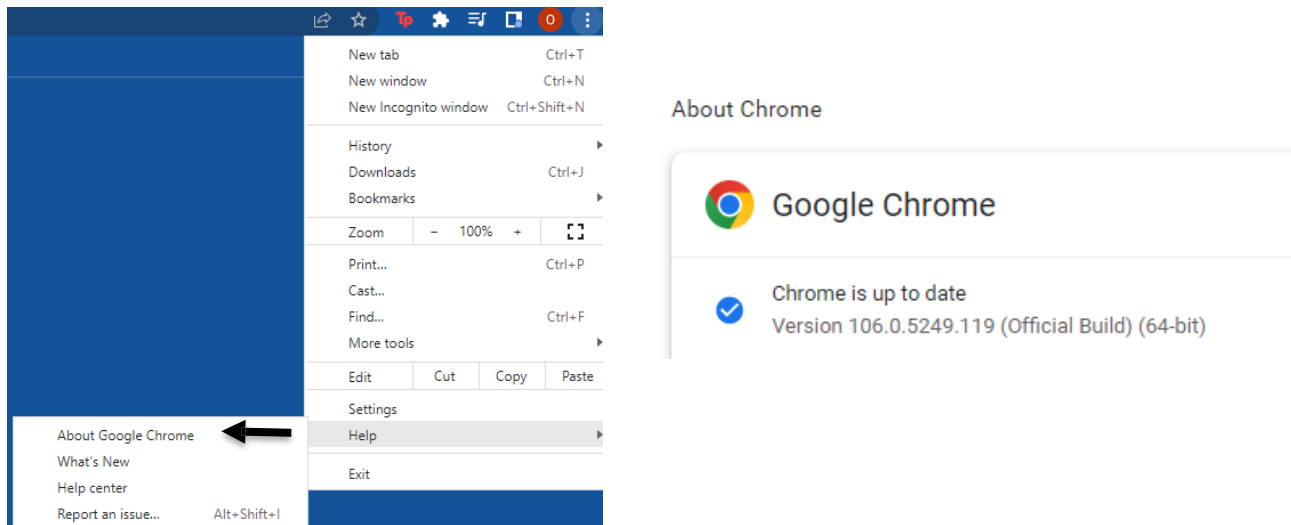
Stable

Version: 126.0.6478.126 (r1300313)

Binary	Platform	URL	HTTP status
chrome	linux64	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/linux64/chrome-linux64.zip	200
chrome	mac-arm64	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/mac-arm64/chrome-mac-arm64.zip	200
chrome	mac-x64	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/mac-x64/chrome-mac-x64.zip	200
chrome	win32	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/win32/chrome-win32.zip	200
chrome	win64	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/win64/chrome-win64.zip	200
chromedriver	linux64	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/linux64/chromedriver-linux64.zip	200
chromedriver	mac-arm64	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/mac-arm64/chromedriver-mac-arm64.zip	200
chromedriver	mac-x64	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/mac-x64/chromedriver-mac-x64.zip	200
chromedriver	win32	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/win32/chromedriver-win32.zip	200
chromedriver	win64	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/win64/chromedriver-win64.zip	200
chrome-headless-shell	linux64	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/linux64/chrome-headless-shell-linux64.zip	200
chrome-headless-shell	mac-arm64	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/mac-arm64/chrome-headless-shell-mac-arm64.zip	200
chrome-headless-shell	mac-x64	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/mac-x64/chrome-headless-shell-mac-x64.zip	200
chrome-headless-shell	win32	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/win32/chrome-headless-shell-win32.zip	200
chrome-headless-shell	win64	https://storage.googleapis.com/chrome-for-testing-public/126.0.6478.126/win64/chrome-headless-shell-win64.zip	200

Se copiará e irá a la URL seleccionada y comenzará la descarga del archivo ZIP.

Es importante verificar que la versión “Stable” concuerde con la versión del navegador Chrome instalada. Para revisar la versión de Chrome se va a Ajustes > Ayuda > Acerca de Google Chrome.



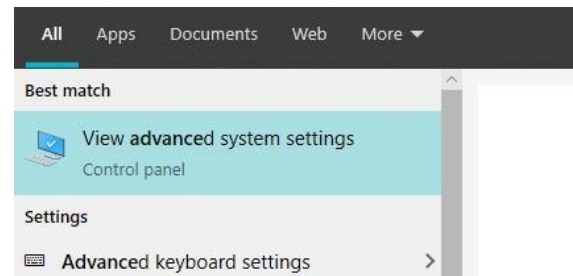
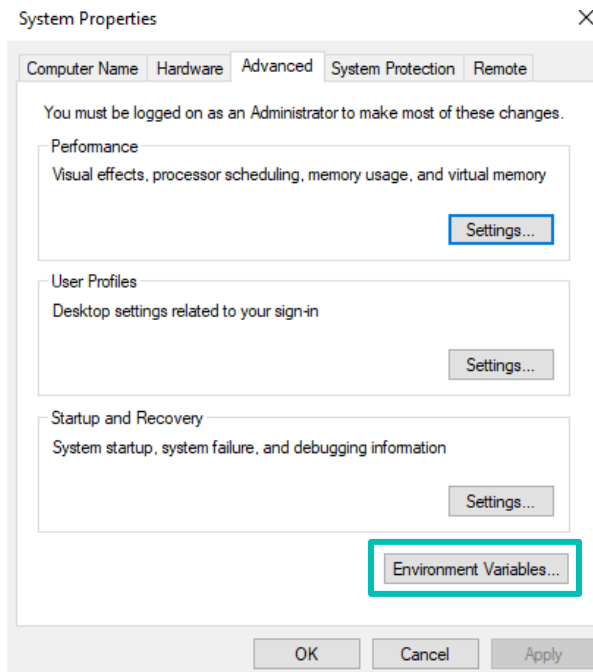
Una vez instalado el driver de Chrome, se procede a repetir los mismos pasos anteriores, pero para el navegador de Firefox, Internet Explorer y Edge.

Una vez descargados los WebDrivers a utilizar se procederá a descomprimir los archivos y a colocarlos dentro de la carpeta creada con anterioridad (C:\bin).

This PC > Windows (C:) > bin				
Name	Date modified	Type	Size	
chromedriver.exe	5/1/2020 5:56 PM	Application	8,666 KB	
geckodriver.exe	10/11/2019 12:06 AM	Application	2,947 KB	
IEDriverServer.exe	10/17/2019 2:11 PM	Application	3,037 KB	
msedgedriver.exe	7/1/2020 9:10 AM	Application	10,582 KB	

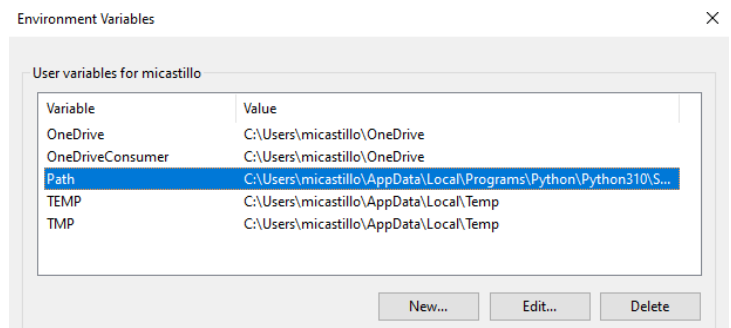
El paso siguiente es agregarla carpeta donde se encuentran los WebDrivers al PATH de nuestras variables del sistema.

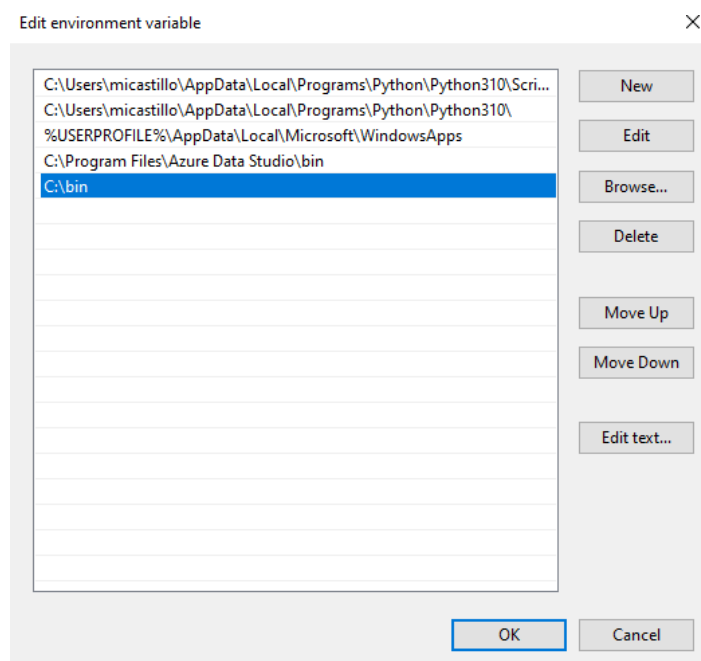
Para ello se debe ingresar a la Configuración Avanzada del Sistema.



Una vez ahí se selecciona la opción de las Variables de Entorno.

En la sección de Variables del Usuario se va a seleccionar la fila de "Path" y posteriormente la opción de "Editar".

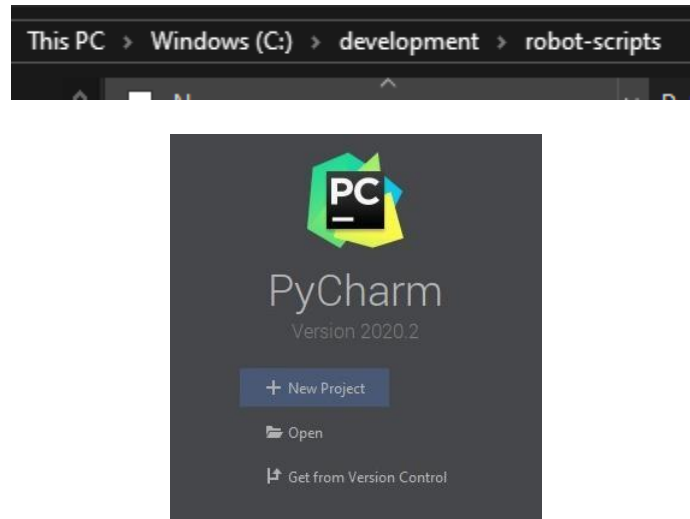




A continuación, debe seleccionarse la opción de “Nueva” y se escribirá la dirección de la carpeta “C:\bin” en el campo vacío que se muestra al final de la lista, una vez finalizado deberá seleccionarse la opción de “OK” para terminar de agregarla al PATH.

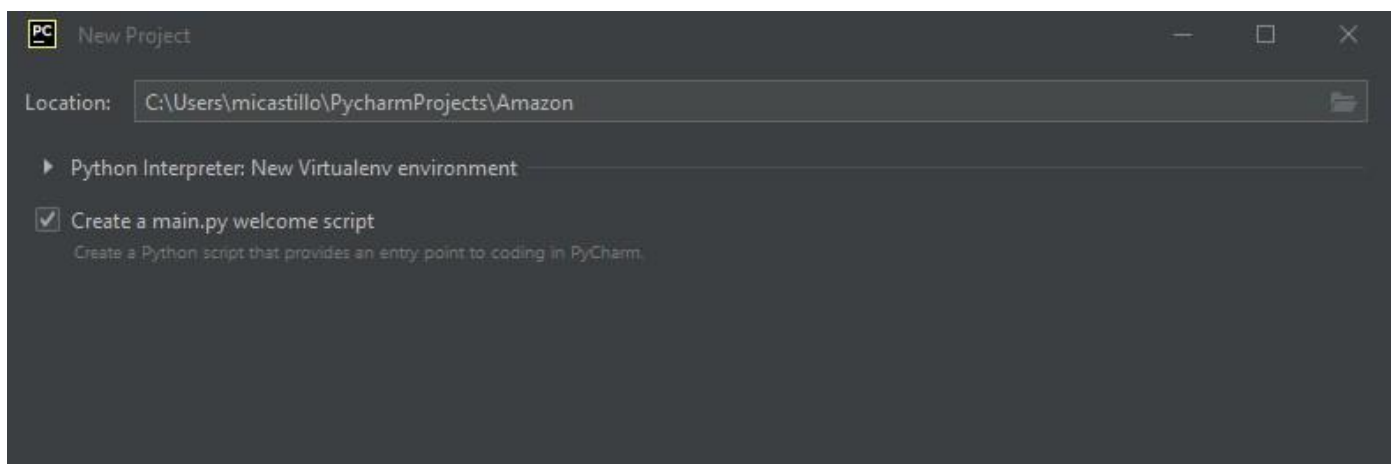
Creación del Proyecto de Pruebas

De preferencia para tener una mayor facilidad de acceso a los proyectos de pruebas se creará la carpeta “development” en el disco local C: y a su vez se creará una carpeta anidada llamada “robot-scripts”, que es donde se almacenará el caso de prueba.

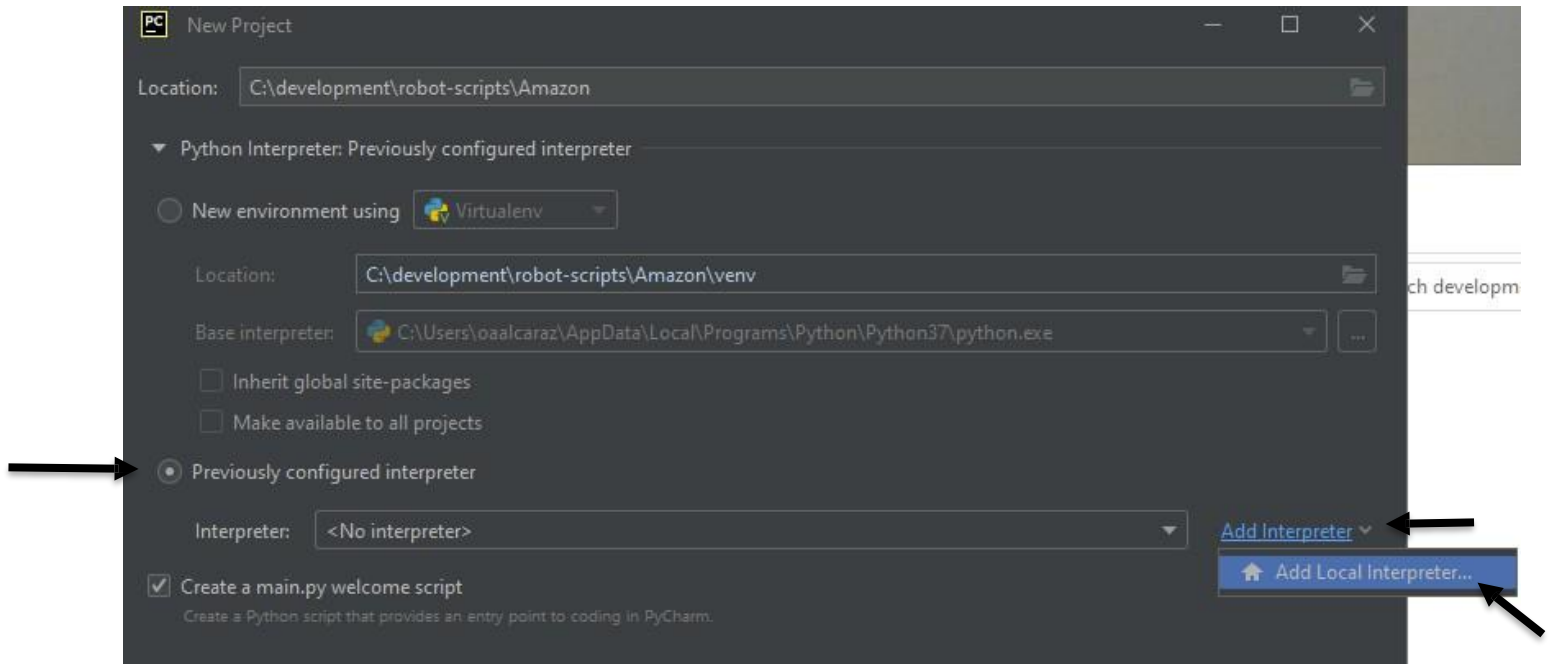


Se ingresará al IDE de PyCharm, y se seleccionará la opción de “Crear un Nuevo Proyecto”.

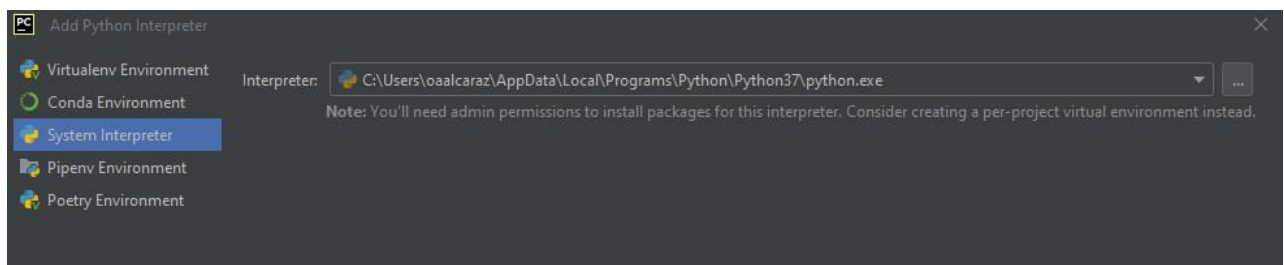
Se seleccionará el lugar donde se almacenará el proyecto y se nombrará “amazon”. Posteriormente se dará click el menú de la opción “Python Interpreter”.



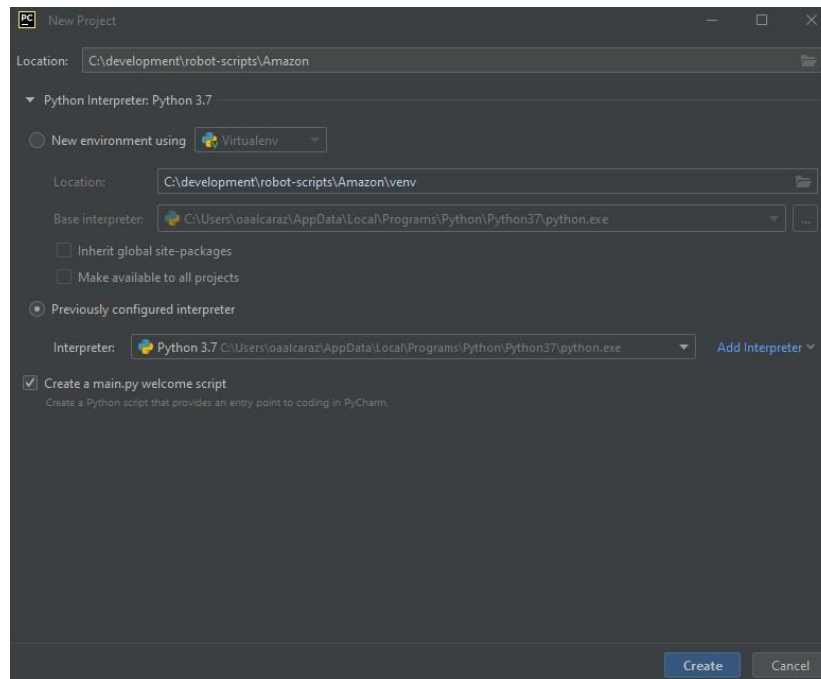
Se seleccionará la opción de “Previously configured interpreter” y después se dará clic en “Add Interpreter” y después “Add Local Interpreter...”.



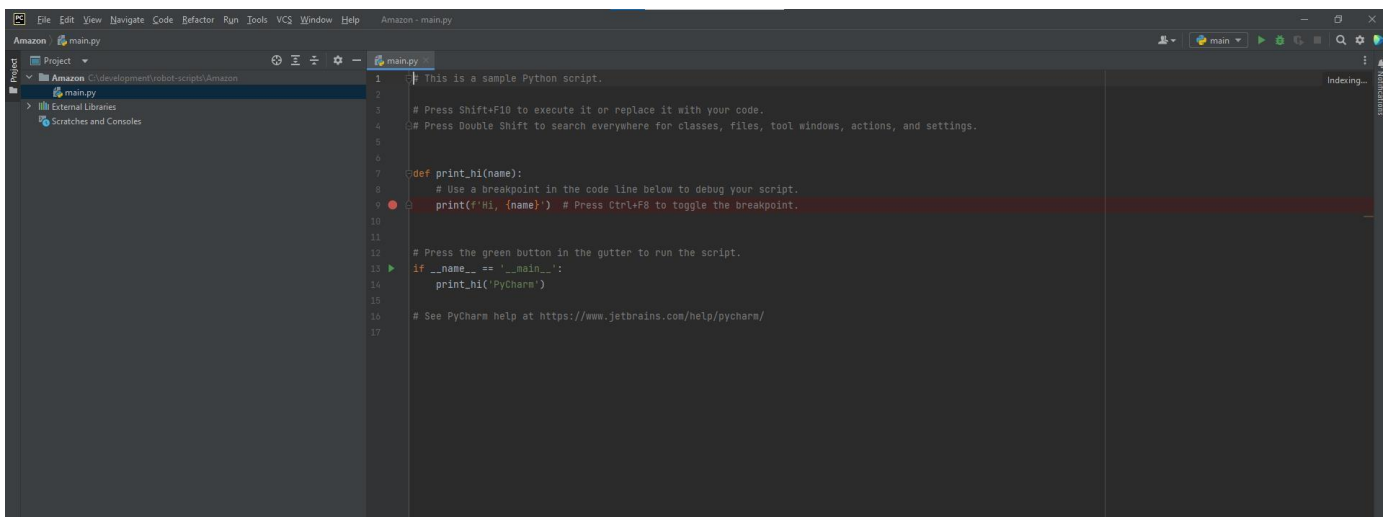
Posteriormente se ingresará en la sección de “System Interpreter” y se mostrará aquel que se presentó cuando se instaló Python en el sistema. Se dejará esa opción y se seleccionará “OK”.



Una vez definido el intérprete se finalizará la creación del proyecto seleccionando la opción de “Crear”.



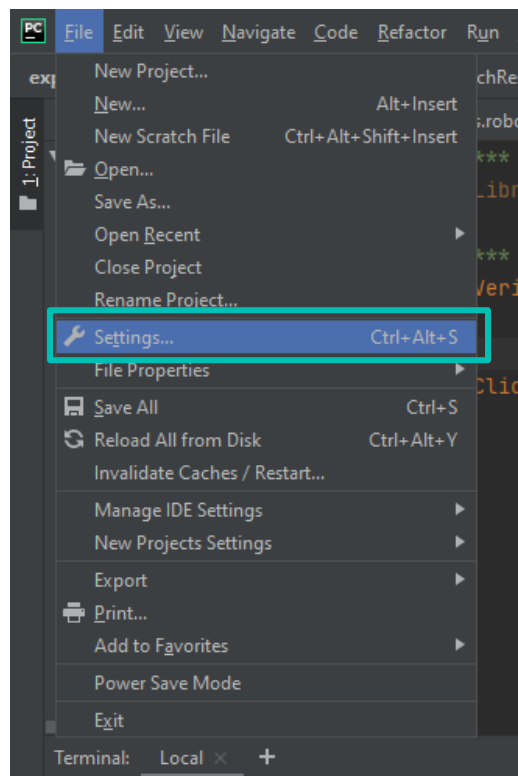
Al hacer clic en “Crear”, se verá el área de trabajo de la siguiente manera.



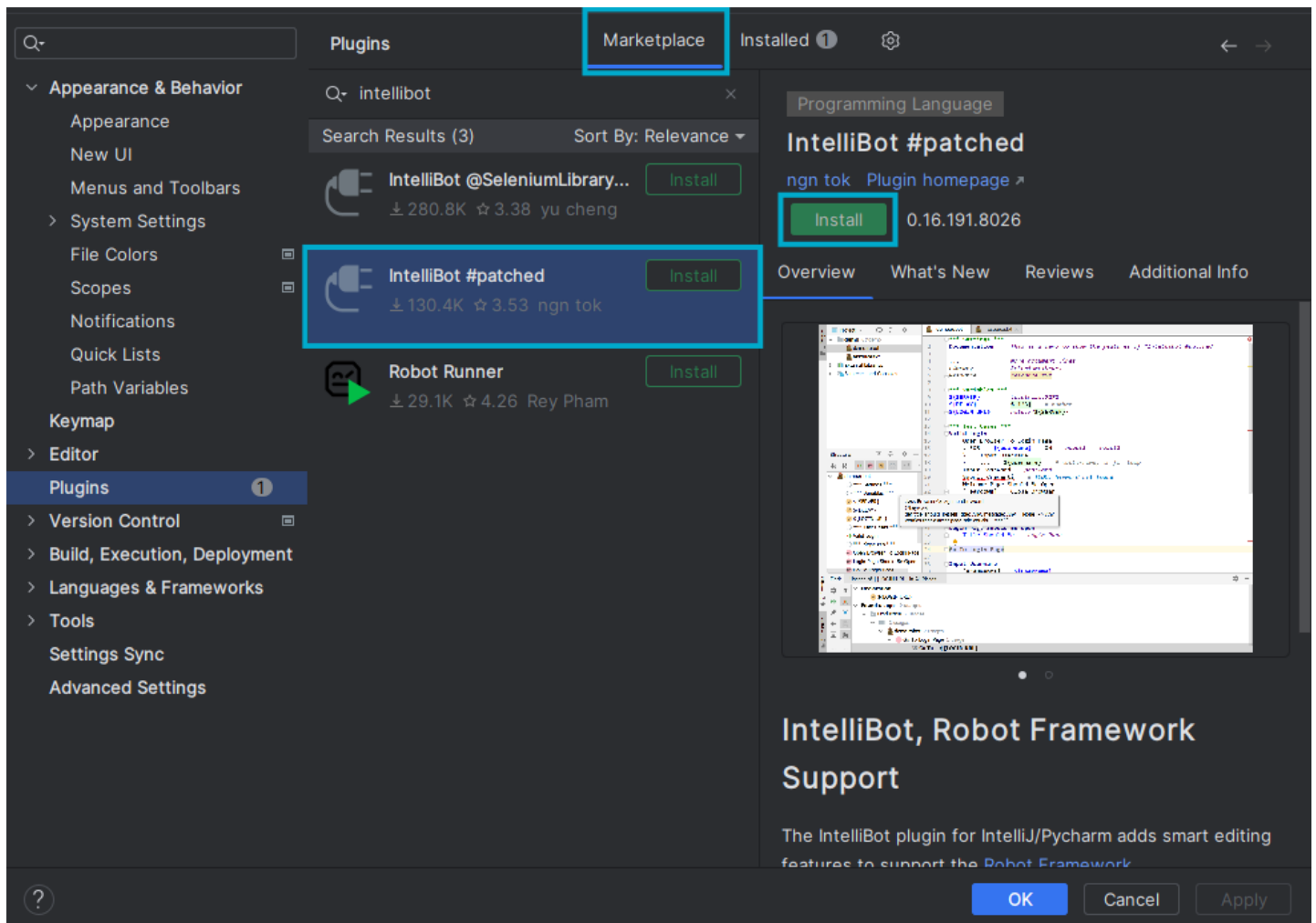
Instalación del plugin de Intellibot

Se instalará el plugin en el IDE de PyCharm que mejora tanto la imagen de los casos de prueba, la sintaxis e incluye también la sugerencia de las palabras clave, que volverán la experiencia del usuario algo más sencillo y eficiente.

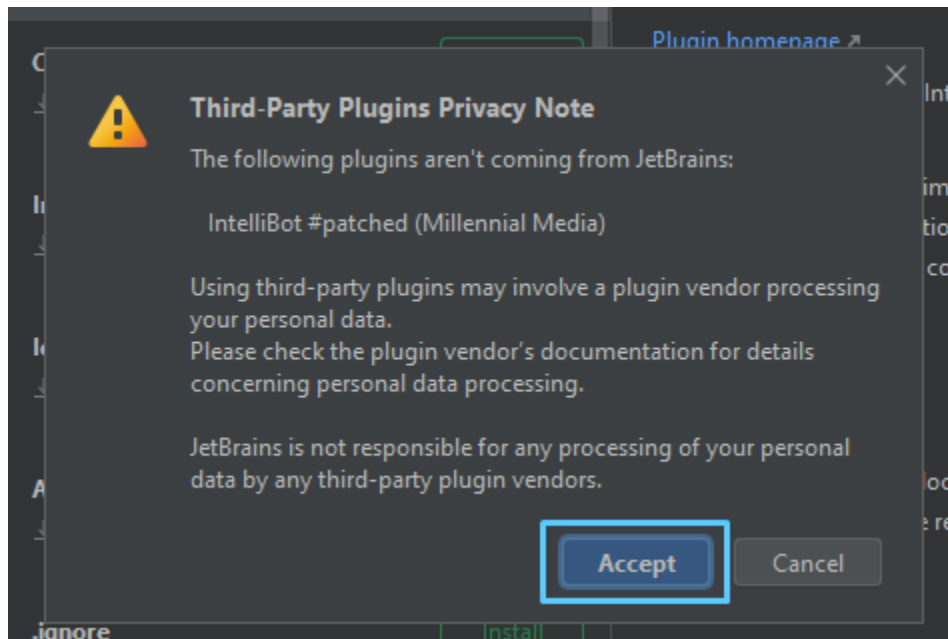
Para ello se deberá ingresar al IDE de PyCharm, al menú “File” y seleccionar la opción de “Settings...”.



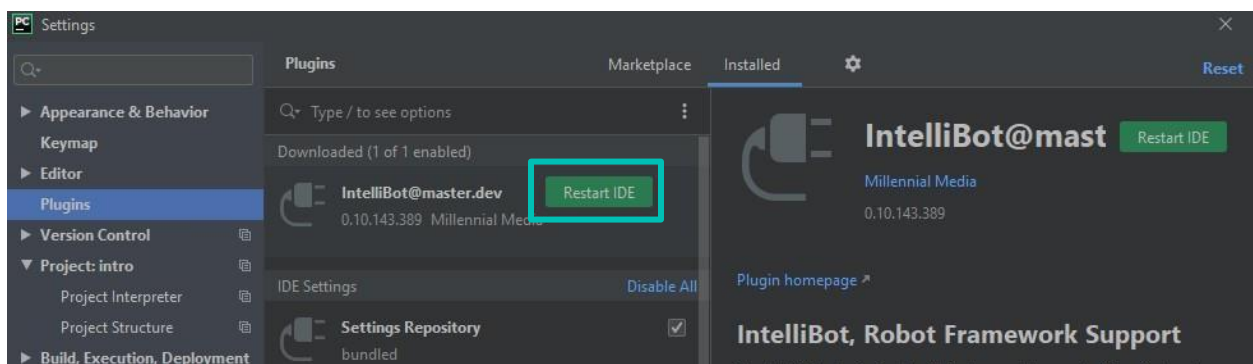
Dentro de la ventana de “Settings” se ingresará a la sección de “Plugins”, se seleccionará la pestaña de “Marketplace”, el plugin “IntelliBot #patched” y damos clic a “Install”.



Saldrá una advertencia a la que se procederá a dar clic en “Accept”.



Una vez instalado aparece la opción de “Restart IDE” para reiniciar el IDE y ver reflejados los cambios. Se selecciona la opción y una vez que se vuelve a abrir el IDE, está listo para empezar a crear los casos de prueba.



El IDE Pycharm se reiniciará, y el Setup estará listo para utilizarse.

Verificación de la Instalación

Para verificar que Robot Framework y las librerías necesarias se han instalado correctamente, puede crear un archivo de prueba simple en el directorio raíz del proyecto de prueba (Amazon) llamado “test.robot” con el siguiente contenido:

```
*** Settings ***
Library      SeleniumLibrary

*** Test Cases ***
Open Amazon Test
    Open Browser    https://www.amazon.com    chrome
    Sleep           5 seconds
    [Teardown]     Close All Browsers
```

Ejecute la prueba utilizando el siguiente comando desde el directorio raíz del proyecto:

```
robot test.robot
```

Si todo está configurado correctamente, un navegador debería abrirse, navegar a <https://www.amazon.com> y cerrarse después de 5 segundos.