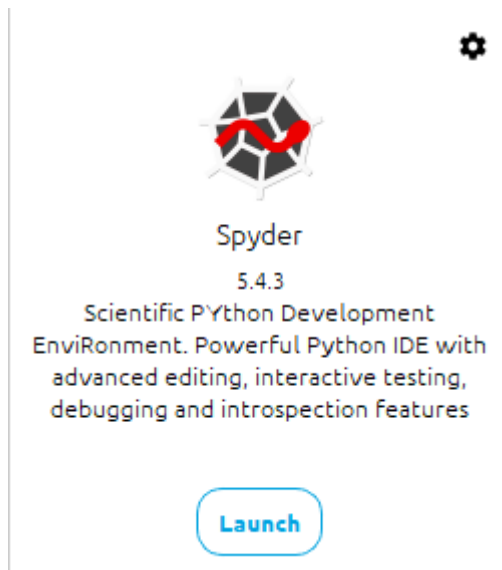
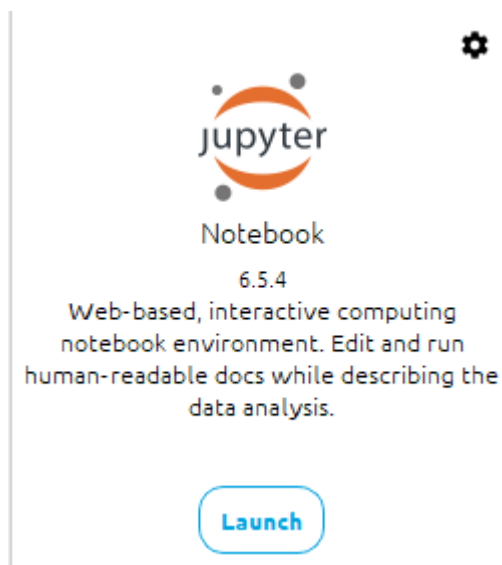


ANACONDA

Encontramos dentro de anaconda dos diferentes apartados donde podemos tanto ejecutar como usar Python, en este caso encontramos Spyder y Jupyter.

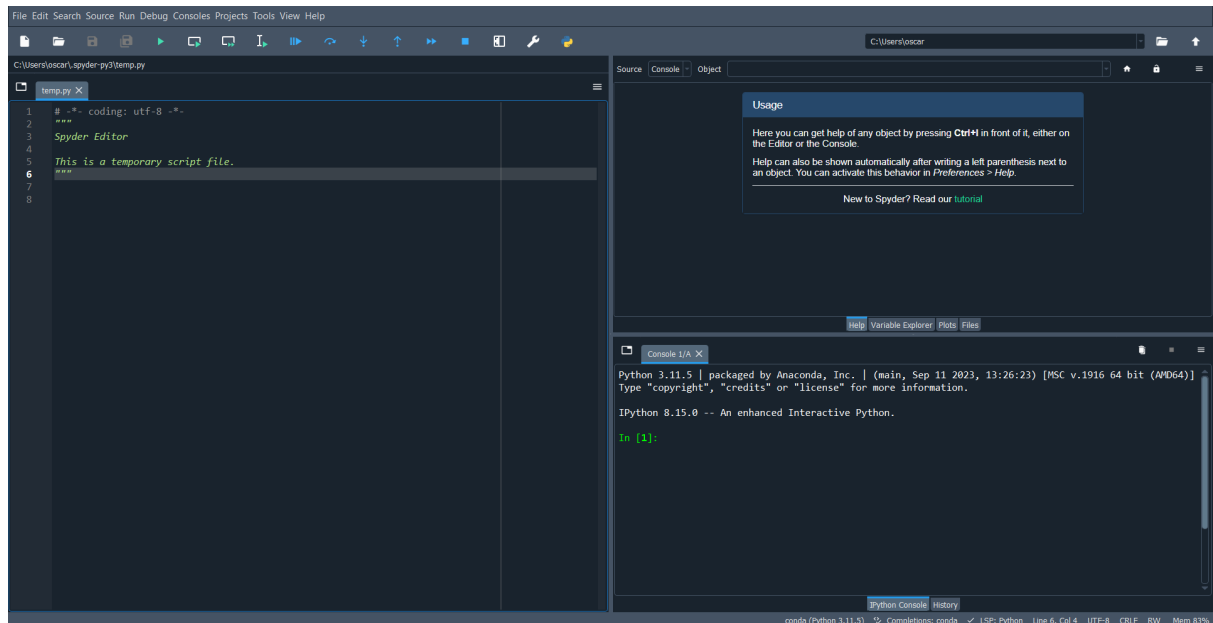


Este ofrece un editor de código, una consola interactiva, integración de librerías...

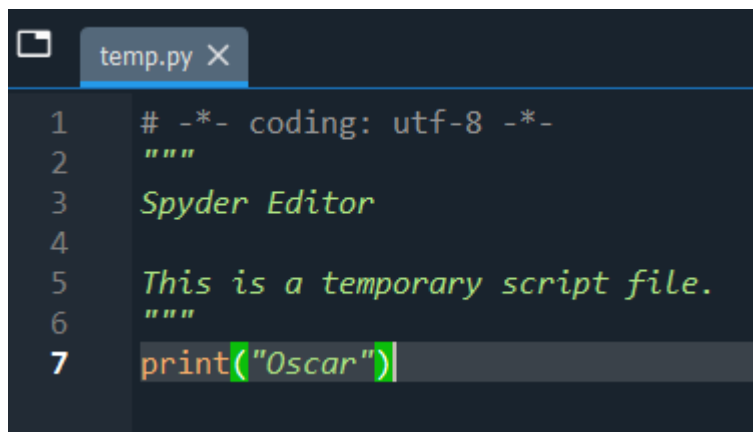


Python se utiliza en Jupyter para escribir y ejecutar código en celdas interactivas. Los usuarios pueden crear documentos enriquecidos que combinan código, texto y visualizaciones en tiempo real.

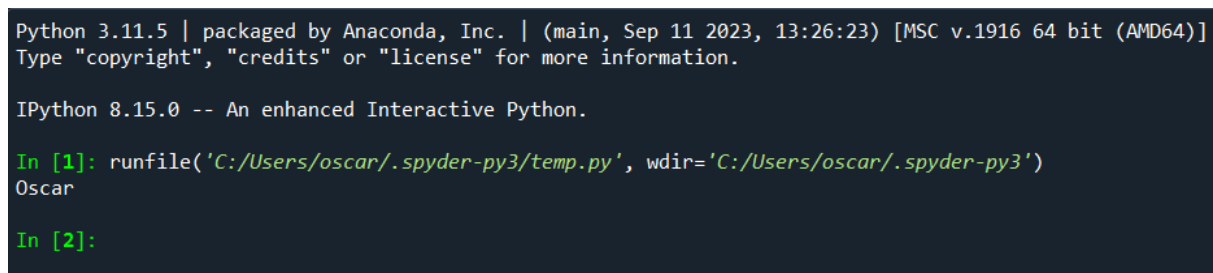
Interfaz de Spyder.



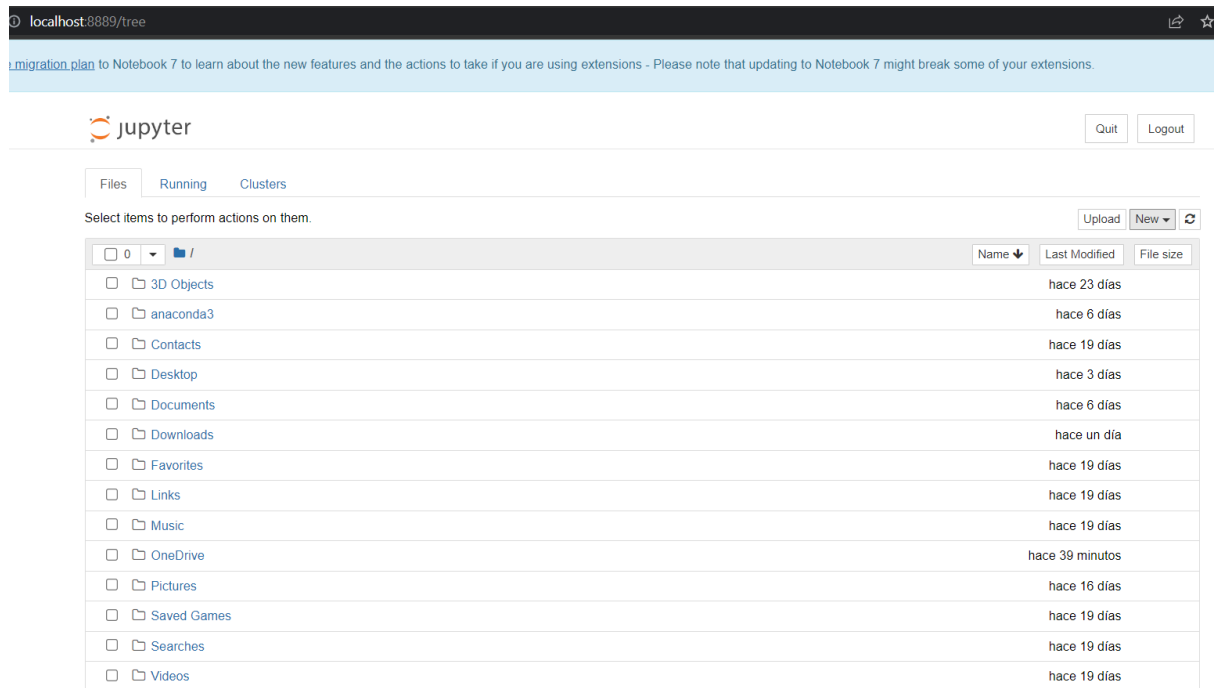
En este caso encontramos una mini consola donde podemos escribir nuestro código



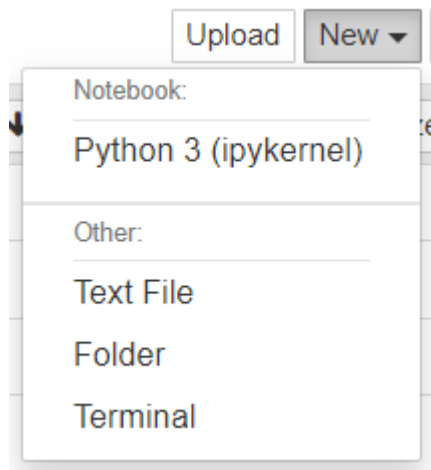
Por la parte derecha tenemos el apartado de ejecución donde podemos ver la ejecución de nuestro código



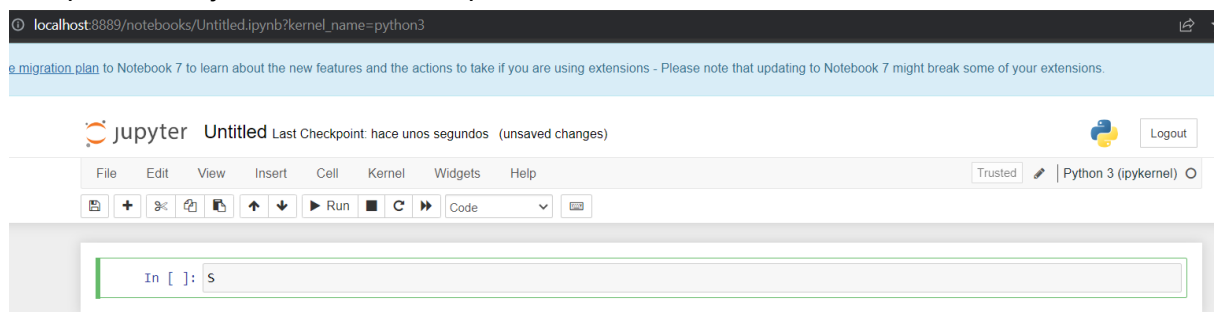
Interfaz de Jupyter



Para acceder a Python apretamos en “New” y le damos a “Python”



Como podemos observar, la interfaz es bastante sencilla, nos ofrece una linea de código la cual podemos ejecutar cuando nos plazca



```
In [1]: print("Oscar")  
  
Oscar
```

Para crear **entornos** hemos elegido “Pip”

1: Primero de todo accedemos al directorio donde queramos que se guarden nuestros entornos y seguidamente usamos el comando `python -m venv “nombre”` para crear el entorno dentro del directorio

```
C:\Users\oscar>cd C:\Users\oscar\Desktop\Envrioment  
C:\Users\oscar\Desktop\Envrioment>python -m venv prueba
```

Seguidamente una vez tenemos creado el directorio debemos de activarlo para que este funcione y podamos acceder a el

```
C:\Users\oscar\Desktop\Envrioment>myenv\Scripts\activate  
El sistema no puede encontrar la ruta especificada.
```

Seguidamente probamos a instalar un paquete dentro del entorno, para ello usamos el comando “`pip install requests`” para instalar en este caso, el paquete “requests”

```
(prueba) C:\Users\oscar\Desktop\Envrioment>pip install requests  
Collecting requests  
  Obtaining dependency information for requests from https://files.pythonhosted.org/packages/c0/09/bb/16/7a/1a26b2ce6cd6965bf26b47bc0bf44/requests-2.31.0-py3-none-any.whl.metadata  
  Downloading requests-2.31.0-py3-none-any.whl.metadata (4.6 kB)  
Collecting charset-normalizer<4,>=2 (from requests)  
  Obtaining dependency information for charset-normalizer<4,>=2 from https://files.pythonhosted.org/packages/18/64/fea4b02ef11da0a2e829ea7b365e09fae2a132314541ad01ae4/charset_normalizer-3.3.1-cp312-cp312-win_amd64.whl.metadata (33 kB)  
  Downloading charset_normalizer-3.3.1-cp312-cp312-win_amd64.whl.metadata (33 kB)  
Collecting idna<4,>=2.5 (from requests)  
  Downloading idna-3.4-py3-none-any.whl (61 kB)  
61.5/61.5 kB 1.7 MB/s eta 0:00:00
```

Para comprobar que se ha instalado con normalidad usamos el comando “`pip list`” para comprobar los paquetes que tenemos instalados dentro del entorno

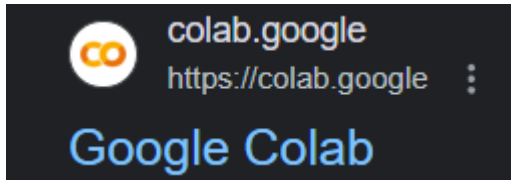
```
(prueba) C:\Users\oscar\Desktop\Envrioment>pip list  
Package           Version  
-----  
certifi            2023.7.22  
charset-normalizer 3.3.1  
idna                3.4  
pip                23.2.1  
requests           2.31.0  
urllib3            2.0.7
```

Una vez hemos terminado de trabajar usamos “`deactivate`” para salir del entorno.

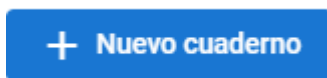
```
(prueba) C:\Users\oscar\Desktop\Envrioment>deactivate  
C:\Users\oscar\Desktop\Envrioment>
```

Por último, para ejecutar Python google nos da la aplicación web “**Collab**” la cual es bastante sencilla de usar.

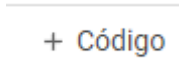
Simplemente nos dirigimos a google y ponemos “google collab”



Una vez dentro vamos a crear un nuevo cuaderno para hacer nuestros programas.



El uso es muy sencillo, cada vez que queramos escribir una línea simplemente tenemos que apretar en “código” para que nos añada una línea más de código



Prueba de funcionamiento:

