

Reporte semanal - Semana 2

09/02/2024

—
Ismael Montoro Peñasco
Fundación GoodJob



Acciones llevadas a cabo.....	2
Mínimo.....	2
Avances realizados.....	3
Labor 1: Proyecto con subprocess y openpyxl.....	3
Labor 2: Ejercicios de la semana y la API de Python.....	4
Labor 3: Fórmulas y Datos de Excel.....	5
Labor 4: Herramientas de IA.....	6
Labor 5: Librería de openai.....	7
Labor 6: Librería PyTest y pruebas unitarias.....	10
Labor 7: Librería Scapy.....	13
Planes para la próxima semana.....	13

Acciones llevadas a cabo

Aumentar el tiempo que le dedicaré, a medida que pasen los días le dedicaré un poco más de tiempo a descubrir nuevas librerías externas de Python y funciones nativas de la API de Python, y si es posible le dedicaré más tiempo a Javascript.

Empezaré los tutoriales de la suite Office empezando por Excel, me concentraré concretamente en la pestaña de Datos, Fórmulas y si da tiempo investigaré Python para Excel.

Empezaré a usar herramientas de inteligencia artificial en local para ir probando distintos modelos de IA.

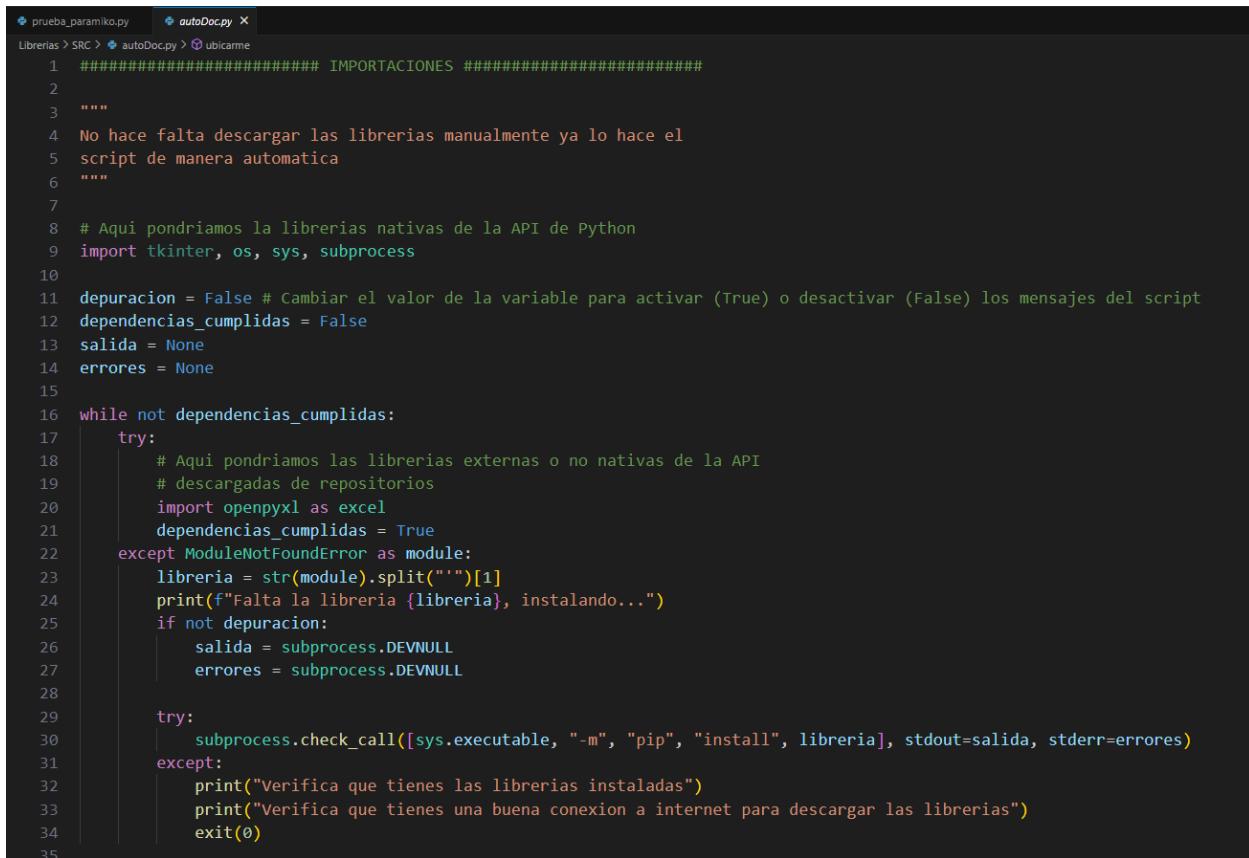
Mínimo

1. Investigar herramientas de IA.
2. Investigar Fórmulas y Datos de Excel.
3. Librerías Externas de Python.
4. API de Python.

Avances realizados

Labor 1: Proyecto con subprocess y openpyxl

Antes de empezar el lunes decidí hacer un pequeño proyecto para leer celdas de hojas de cálculo de Excel desde Python, con la librería subprocess y openpyxl.



```

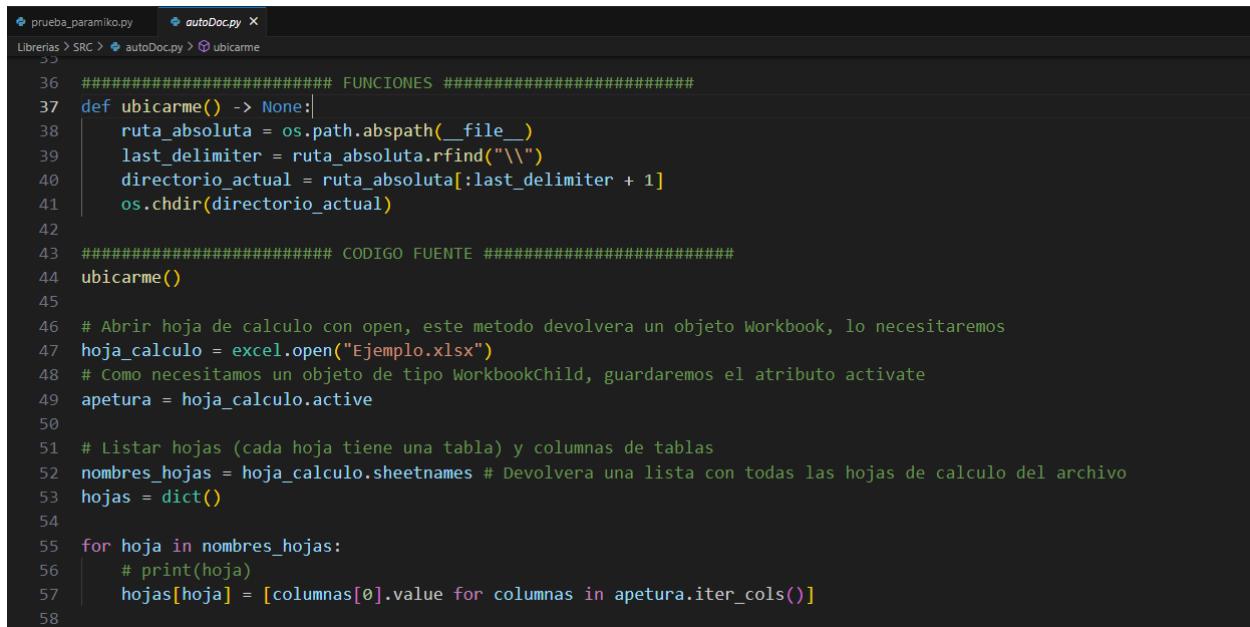
 ❶ prueba_paramiko.py ❷ autoDoc.py X
Librerias > SRC > ❸ autoDoc.py > ❹ ubicarme
1 ##### IMPORTACIONES #####
2 """
3 """
4 No hace falta descargar las librerias manualmente ya lo hace el
5 script de manera automatica
6 """
7
8 # Aqui pondriamos las librerias nativas de la API de Python
9 import tkinter, os, sys, subprocess
10
11 depuracion = False # Cambiar el valor de la variable para activar (True) o desactivar (False) los mensajes del script
12 dependencias_cumplidas = False
13 salida = None
14 errores = None
15
16 while not dependencias_cumplidas:
17     try:
18         # Aqui pondriamos las librerias externas o no nativas de la API
19         # descargadas de repositorios
20         import openpyxl as excel
21         dependencias_cumplidas = True
22     except ModuleNotFoundError as module:
23         libreria = str(module).split(" ")[1]
24         print(f"!Falta la libreria {libreria}, instalando...")
25         if not depuracion:
26             salida = subprocess.DEVNULL
27             errores = subprocess.DEVNULL
28
29     try:
30         subprocess.check_call([sys.executable, "-m", "pip", "install", libreria], stdout=salida, stderr=errores)
31     except:
32         print("Verifica que tienes las librerias instaladas")
33         print("Verifica que tienes una buena conexion a internet para descargar las librerias")
34         exit(0)
35

```

La librería subprocess me sirve para instalar la librería openpyxl en caso de que no esté instalada.

Puede encontrar información al respecto de este truco gracias a [Bing IA](#).

También hago el uso de la librería os para declarar al script el directorio donde tiene que trabajar, y usó la librería openpyxl para realizar las lecturas a la hoja de cálculo.



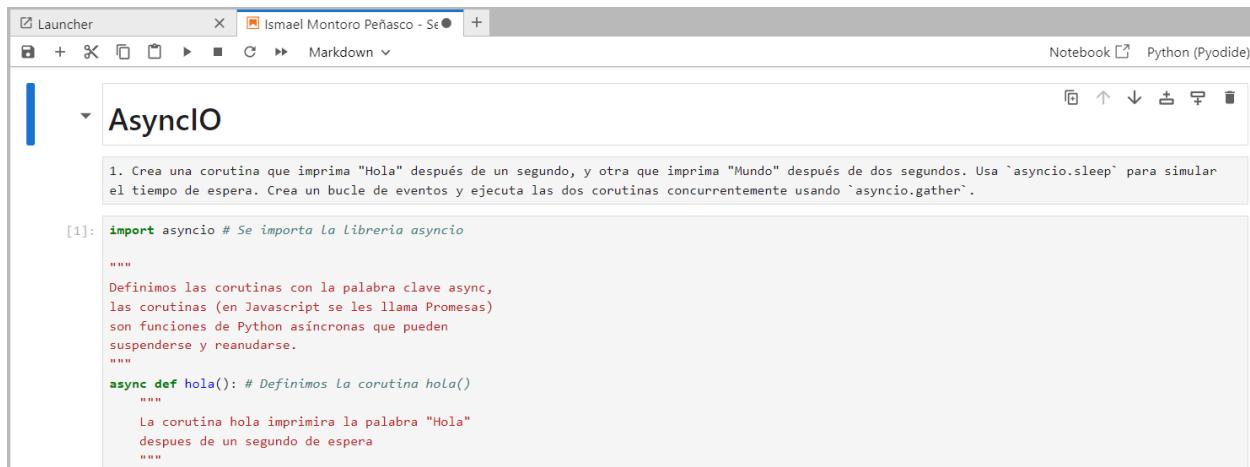
```

prueba_paramiko.py autoDoc.py
Librerias > SRC > autoDoc.py > ubicarme
36 ##### FUNCIONES #####
37 def ubicarme() -> None:
38     ruta_absoluta = os.path.abspath(__file__)
39     last_delimiter = ruta_absoluta.rfind("\\")
40     directorio_actual = ruta_absoluta[:last_delimiter + 1]
41     os.chdir(directorio_actual)
42
43 ##### CODIGO FUENTE #####
44 ubicarme()
45
46 # Abrir hoja de calculo con open, este metodo devolvera un objeto Workbook, lo necesitaremos
47 hoja_calculo = excel.open("Ejemplo.xlsx")
48 # Como necesitamos un objeto de tipo WorkbookChild, guardaremos el atributo activate
49 apertura = hoja_calculo.active
50
51 # Listar hojas (cada hoja tiene una tabla) y columnas de tablas
52 nombres_hojas = hoja_calculo.sheetnames # Devolvera una lista con todas las hojas de calculo del archivo
53 hojas = dict()
54
55 for hoja in nombres_hojas:
56     # print(hoja)
57     hojas[hoja] = [columnas[0].value for columnas in apertura.iter_cols()]
58

```

Labor 2: Ejercicios de la semana y la API de Python

Mediante los ejercicios de la semana fui fortaleciendo mi conocimiento de la API de Python, he investigado lo que son las librerías asyncio para realizar corutinas y threading para realizar programación paralela con hilos en Python.



1. Crea una corutina que imprima "Hola" después de un segundo, y otra que imprima "Mundo" después de dos segundos. Usa `asyncio.sleep` para simular el tiempo de espera. Crea un bucle de eventos y ejecuta las dos corutinas concurrentemente usando `asyncio.gather`.

```

[1]: import asyncio # Se importa La Libreria asyncio
"""
Definimos las corutinas con la palabra clave async,
las corutinas (en Javascript se les llama Promesas)
son funciones de Python asíncronas que pueden
suspenderse y reanudarse.
"""

async def hola(): # Definimos la corutina hola()
"""
    La corutina hola imprimira la palabra "Hola"
    despues de un segundo de espera
"""

```

Labor 3: Fórmulas y Datos de Excel

He investigado algunas funciones de Excel para realizar labores básicas, algunas funciones de manejo de texto y fechas, y dos funciones potentes IF para aplicar condicionales y BUSCARV y BUSCARH para buscar datos dentro de una matriz.

Nombre	Funcionalidad	Ejemplo	Datos de prueba
CARÁCTER	Nos da un carácter dependiendo del decimal de la tabla Ascii	A	
IZQUIERDA	Extrae los caracteres de una cadena por la derecha	I	
MAYUSCULAS	Devuelve la cadena introducida a mayúscula	DEVUELVE LA CADENA INTRODUCIDA A MAYUSCULA	Numeros
MINUSCULAS	Devuelve la cadena introducida a minúscula	devuelve la cadena introducida a minúscula	Texto
IGUAL	Devuelve Verdadero si la cadena es igual a la otra	VERDADERO	
MATRIZATEXTO	Crea una concatenación de varios textos a partir de una matriz completa	Nos da un carácter dependiendo del decimal de la tabla Ascii; Devuelve la cadena introducida a mayúscula; Devuelve la cadena introducida a minúscula; Devuelve Verdadero si la cadena es igual a la otra	
DERECHA	Extrae los caracteres de una cadena por la derecha	Verdadero si la cadena es igual a la otra	
IZQUIERDA	Extrae los caracteres de una cadena por la izquierda	la	2 a
CONCAT	Concatena texto	Ho	1 dasdadas
REEMPLAZAR	Subtrae una parte de la cadena y la sustituye por una tuya	HolaMundo!	3 dasadasd515 aksmdk!
HOY	Devuelve la fecha actual, cambia de valor cada día	Holmknddo!	

He investigado también algunas opciones de la pestaña datos, cómo extraer datos de una web, como aplicar un filtro a una tabla, como solucionar los errores de toda una columna, aplicar ordenador de datos.

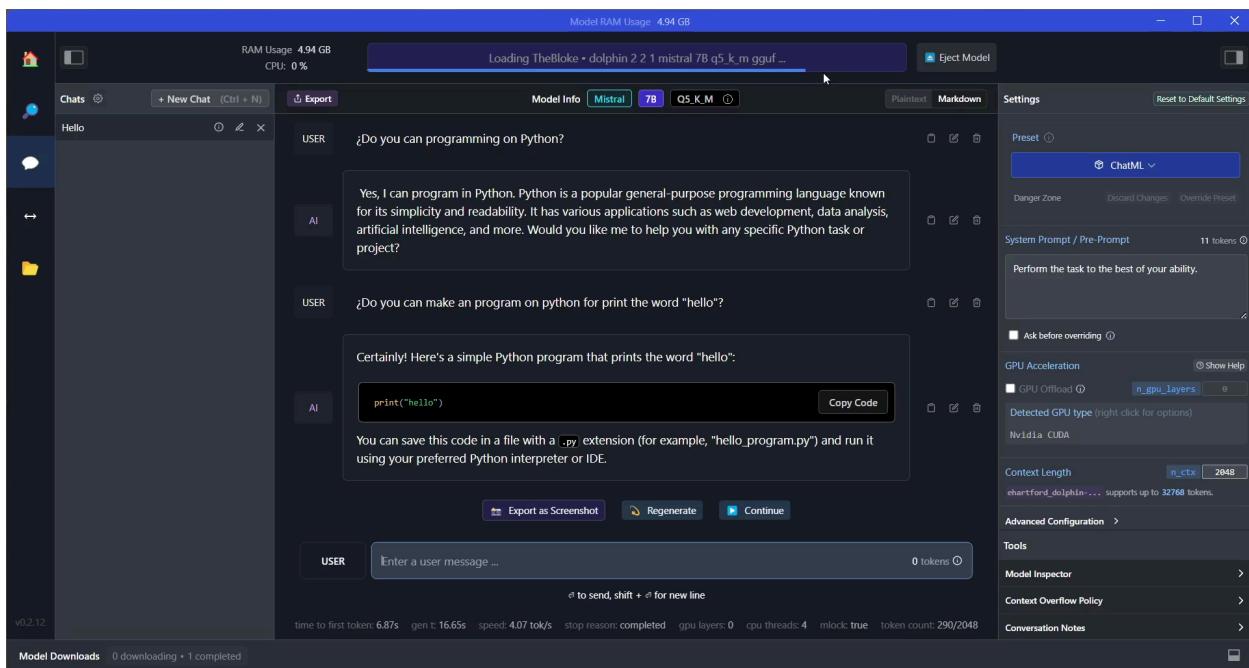
Z	Símbolo	Elemento	Grupo	Periodo	Peso atómico	Densidad/g/cm³	Fusión/K	Ebullición/K	Cci/(g·K)	Neg	Abundancia	Año	Descubridor(es)	Origen del nombre
1	H	Hidrógeno	1	1	1.008	0.00008988	14.01	20.28	14.304	2.20	1400	1766	Henry Cavendish	(griego) hydro- + -gen, 'helio'
2	He	Helio	18	1	4.002602	0.0001785	0.95	4.22	5.193	-	0.008	1895	Ramsay y Cleve	(mitología) helios, de la
3	Li	Litio	1	2	6.94	0.534	453.69	1560	5.582	0.98	20	1817	Arfvedson	(griego) lithos, roca de
4	Be	Berilio	2	9.0121831	1.85	1560	2742	1.825	1.57	2.8	1797	Vauquelin	(griego) beryllios, piedra	
5	B	Boro	13	2	10.81	2.34	2349	4200	1.026	2.04	10	1808	Davy y Gay-Lussac	(árabe) 'buraq, borax, u
6	C	Carbono	14	2	12.011	2.267	3800	4300	0.709	2.55	200	1772	Prehistoria	(latín) carbo, 'carbón'.
7	N	Nitrogeno	15	2	14.007	0.0012508	63.15	77.36	1.04	3.04	19	1772	Rutherford	(griego) nitron y -gen, 'e
8	O	Oxígeno	16	2	15.999	0.001429	54.36	90.20	0.918	3.44	461000	1774	Priestley y Scheele	(griego) oxy-, tanto 'for
9	F	Flúor	17	2	18.99840316	0.001696	53.53	85.03	0.824	3.98	585	1886	Moissan	(latín) fluer, 'fluir'.
10	Ne	Neón	18	2	20.1797	0.0008999	24.56	27.07	1.03	-	0.005	1898	Ramsay y Travers	(griego) neos, 'nuevo'.
11	Na	Sodio	1	3	22.98976928	0.971	370.87	1156	1.228	0.93	23600	1807	Davy	(árabe) sawdā, negra; 's
12	Mg	Magnesio	2	3	24.303	1.738	923	1363	1.023	1.31	23300	1755	Black	(griego) de Magnesia, c
13	Al	Aluminio	13	3	26.9815385	2.698	933.47	2792	0.897	1.61	82300	1825	Oersted	(latín) alumen, 'alumina'
14	Si	Silicio	14	3	28.085	2.3296	1687	3538	0.705	1.9	282000	1824	Berzelius	(latín) silex (originariamente)
15	P	Fósforo	15	3	30.9737619981.82	2.067	317.30	550	0.769	2.19	1050	1669	Brand	(griego) phosphorus, '
16	S	Azufre	16	3	32.06	2.067	388.36	717.87	0.71	2.58	350	1801	Prehistoria	(latín) sulphur, 'sulfuro'.
17	Cl	Cloro	17	3	35.45	0.003214	171.6	239.11	0.479	3.16	145	1774	Scheele	(griego) chloros, 'amarillo'
18	Ar	Argón	18	3	39.948	0.0017837	83.80	87.30	0.52	-	3.5	1894	Ramsay y Rayleigh	(griego) argos, 'inactivo'
19	K	Potasio	1	4	39.0983	0.862	336.53	1032	0.757	0.82	20900	1807	Davy	(inglés) del inglés pot as
20	Ca	Calcio	2	4	40.078	1.54	1115	1757	0.647	1	41500	1808	Davy	(griego) calx, 'caliza' (Ca)
21	Sc	Escandio	3	4	44.955908	2.989	1814	3109	0.568	1.36	22	1879	Nilson	(lugar) Scandia, nombre
22	Ti	Titánio	4	4	47.867	4.54	1941	3560	0.523	1.54	5650	1791	Gregor y Klapproth	(mitología) de los Titanes
23	V	Vanadio	5	4	50.9415	6.11	2183	3680	0.489	1.63	120	1801	del Rio	(mitología) de la diosa e
24	Cr	Cromo	6	4	51.9961	7.15	2180	2944	0.449	1.66	102	1797	Vauquelin	(griego) chroma, 'color'
25	Mn	Manganoso	7	4	54.938044	7.44	1519	2334	0.479	1.55	950	1774	Gahn	(griego) corrupción del

Labor 4: Herramientas de IA

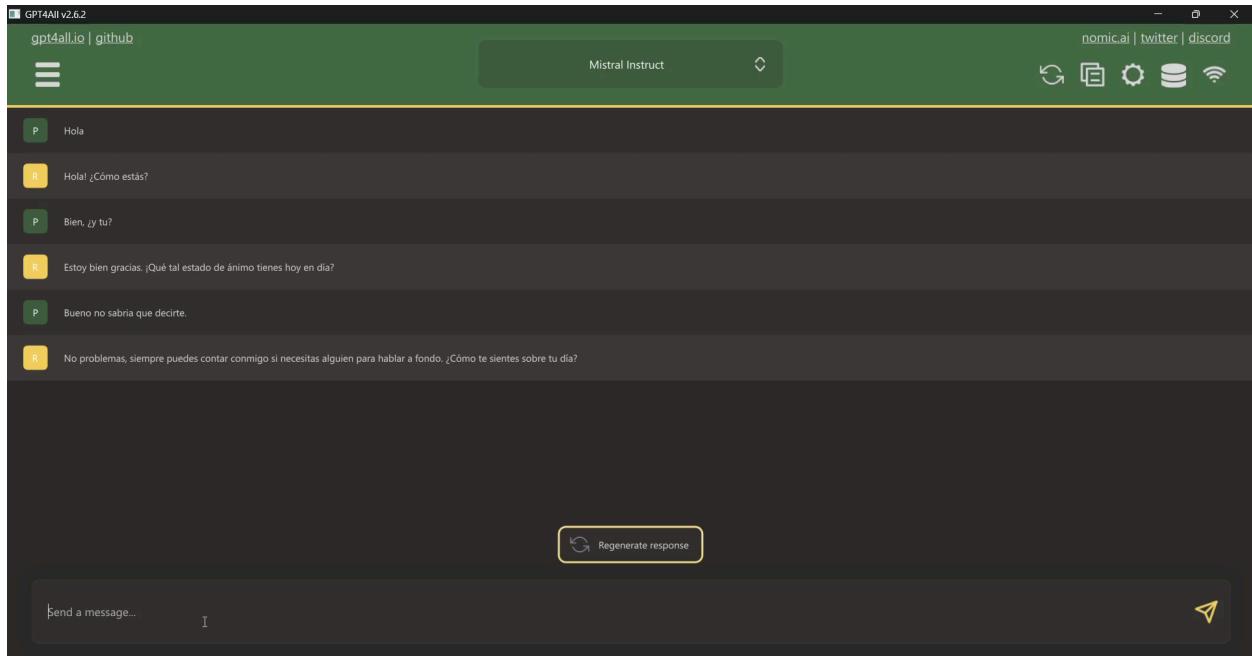
En la Masterclass de Inteligencia Artificial realizada por GoodJob le comente a Román Ramírez Giménez que había herramientas de IA locales, y como le encantó aquel aporte, me propuse usarlas, empezando por LM Studio, que él me sugirió después de que investigara, la herramienta que le propuse que se llama chat4all.

Esta herramienta de inteligencia tiene cositas interesantes como:

- Buscador de modelos de inteligencia artificial, donde puedes buscar repositorios con modelos ya listos para usar, e instalarlos dentro de LM Studio. Los repositorios se descargan de <https://huggingface.co/>.
- La posibilidad de ejectar o desactivar el modelo que estás usando.
- La posibilidad de hacerse un servidor http para realizar consultas a la IA desde fuera de la interfaz de LM Studio con una API de Python.
- Un listado de los modelos descargados.
- La posibilidad Configurar los modelos ya instalados.



También investigue chat4all, la diferencia que tiene con LM Studio, es que es bastante simple, no tiene muchas opciones de configuración, además no encontré ninguna opción para crear un servidor web para hacer consultas al modelo de IA que tengas instalado y hacer consultas desde una API.

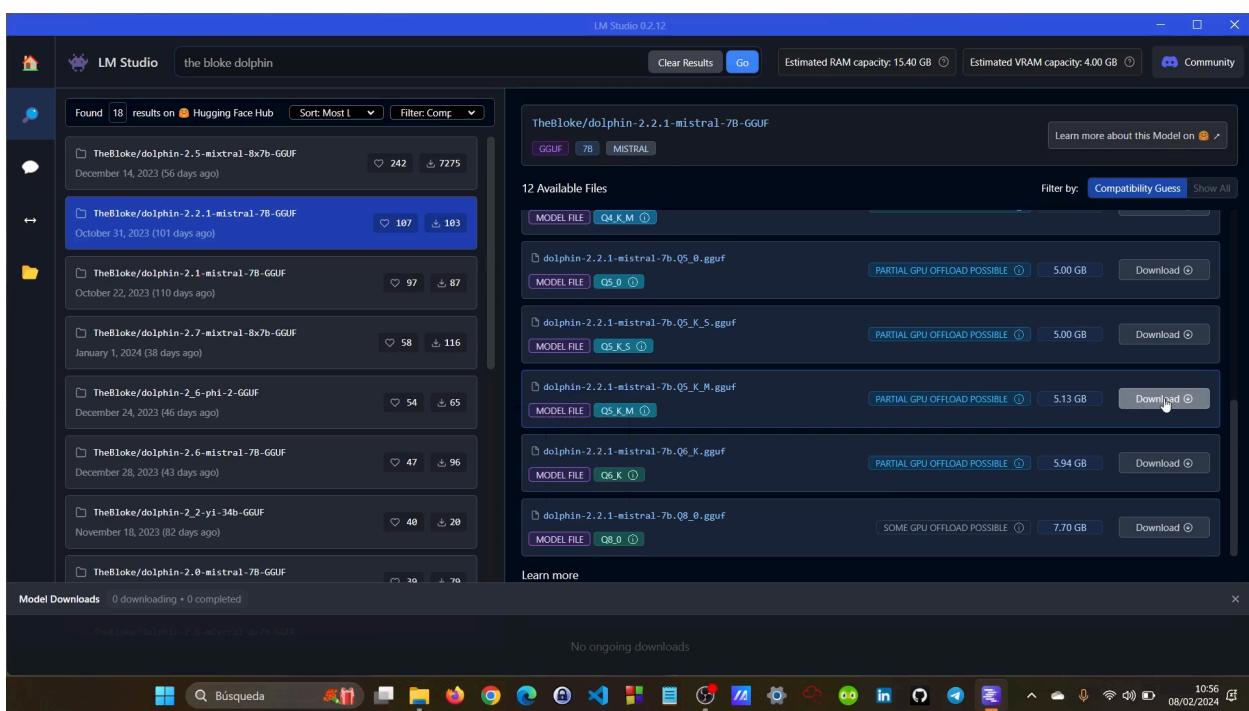
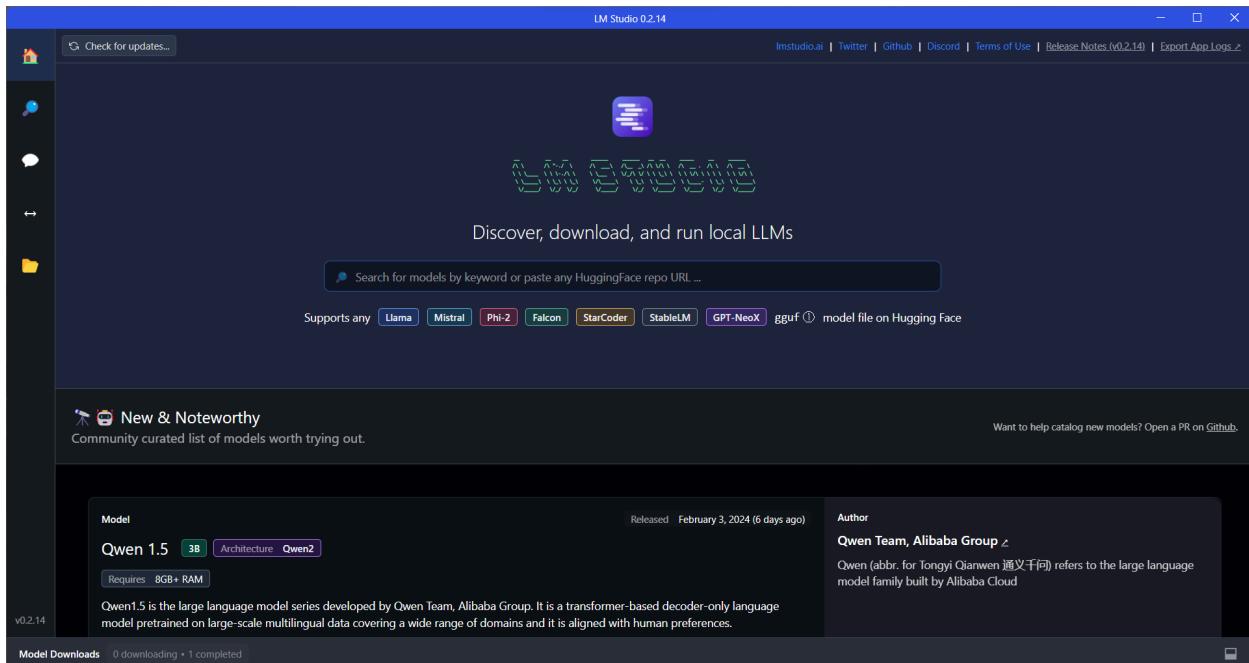


Labor 5: Librería de openai

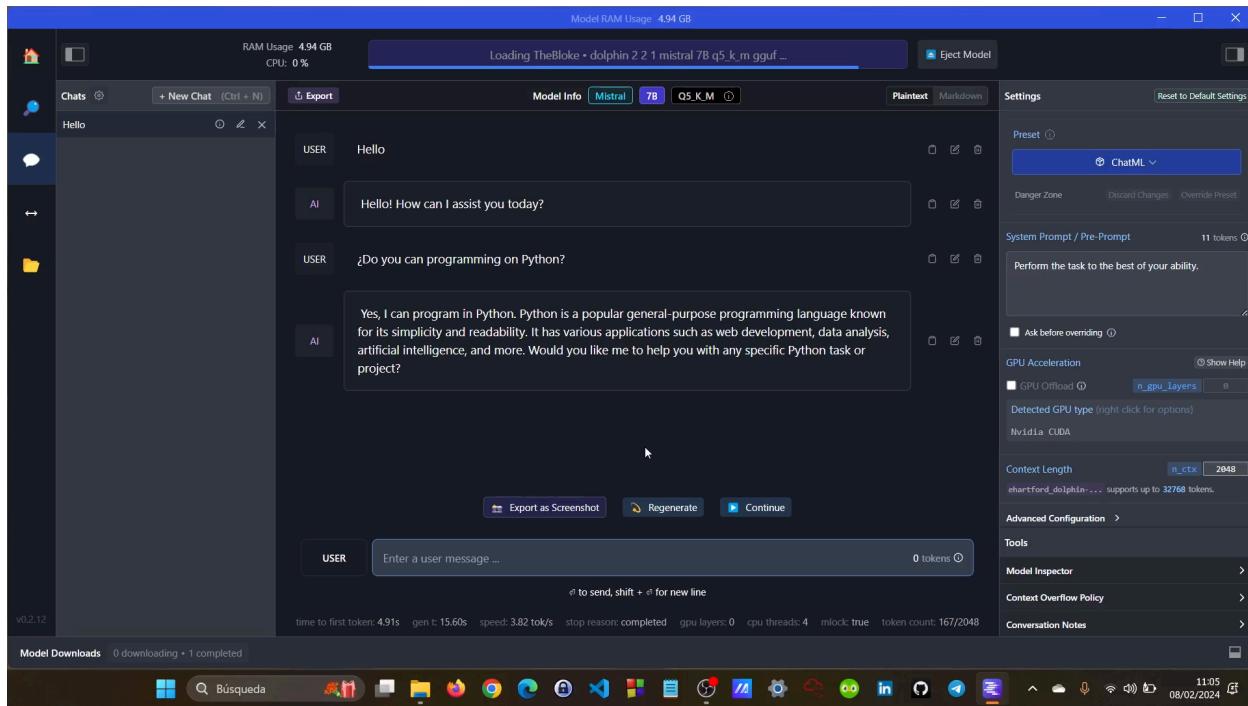
Disponible de pypi.org en este enlace "<https://pypi.org/project/openai/>", esta librería me permite realizar peticiones a la IA si es que tengo el servidor activo.

Aquí dejo un procedimiento, que si bien está guiado, no recomiendo hacerlo a no ser que tengas un hardware bastante potente.

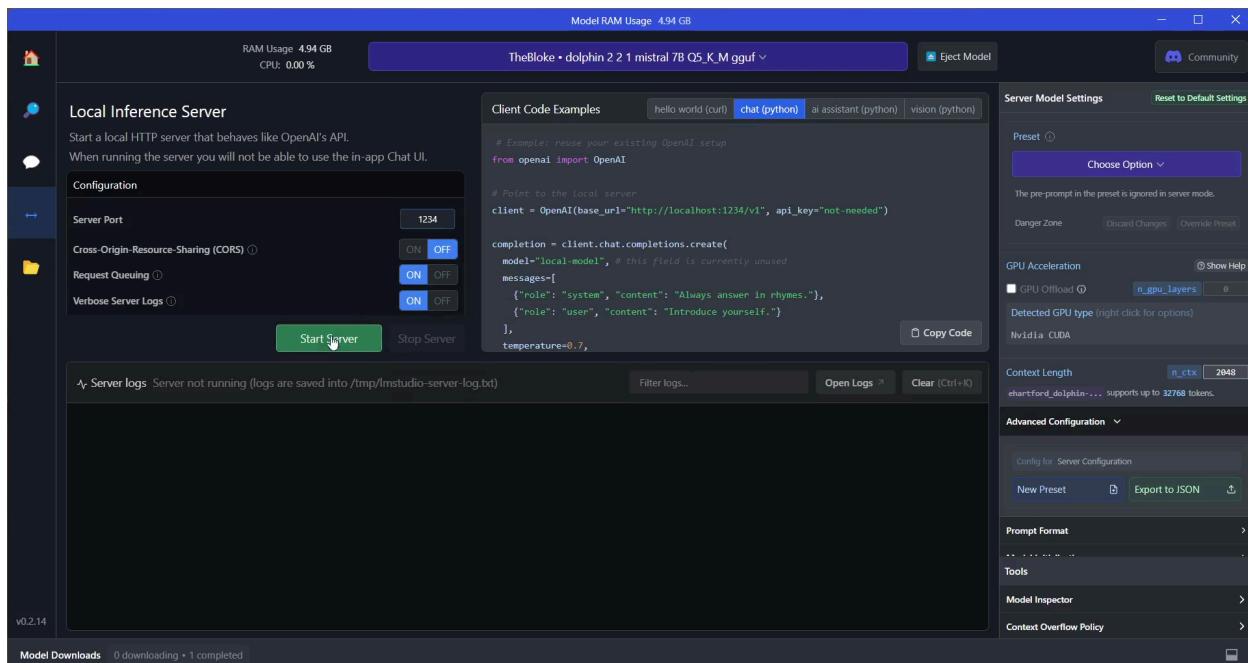
1. Descargamos e instalamos el modelo de inteligencia artificial, a través del buscador de LM Studio, el cual utiliza el sitio de <https://huggingface.co/>.



- Probamos para ver si el modelo funciona, haciéndole algunas preguntas, desde el apartado de chats.



- Activamos el servidor, desde Local Server → Start Server



- Instalamos la librería de openai con “pip install openai”. Ojo, la librería no funciona en entornos virtuales de Python, tiene que ser instalada en la máquina.

- Realizamos un script de python a partir del ejemplo que nos ofrecen en LM Studio y la ejecutamos. Cuidado porque ejecutar el script requiere de muchos recursos hardware.

Dev_Ai > LM_Studio.py > ...

```
9 while True:
10
11     prompt = input("Que quieres decirle a la IA: ")
12
13     if prompt == "Exit now":
14         break
15
16     completion = client.chat.completions.create(
17         model="local-model", # this field is currently unused
18         messages=[
19             {"role": "system", "content": "Always answer in rhymes."},
20             {"role": "user", "content": "Hello, my name Ismael"}
21             {"role": "user", "content": prompt}
22         ],
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

PS C:\Users\ismae\git\programacion_avanzada> & C:/Users/ismae/AppData/Local/Programs/Python/Python311/python.exe c:/User...
studio.py
Que quieres decirle a la IA: Hello
Hi

Que quieres decirle a la IA: █

Labor 6: Librería PyTest y pruebas unitarias

PyTest es una librería de Python que nos permite realizar pruebas unitarias, es decir, pruebas para ver si el código realmente funciona como queremos.

```
1 import pytest
2 # Si utilizas esta importacion pytest dara error, pytest no admite importaciones relativas
3 # from .mi_persona import Persona
4 from SRC.PyTest.Persona import Persona
5
6 class TestPersona:
7
8     def test_prueba(self):
9         assert 0 == 0 # No da ningun error porque la condicion es correcta
10        """
11            Da error porque la condicion no es correcta,
12            y si da error detiene la prueba unitaria en
13            donde se sucedio dicho error
14        """
15        # assert 0 != 0
16        assert 2 > 0
```

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
prueba_pytest.py .

PS C:\Users\ismael\Downloads\EstudiosPython\Librerias\SRC\PyTest> pytest prueba_pytest.py
=====
platform win32 -- Python 3.11.5, pytest-8.0.0, pluggy-1.4.0
rootdir: C:\Users\ismael\Downloads\EstudiosPython\Librerias\SRC\PyTest
collected 1 item

prueba_pytest.py .

=====
1 passed in 0.06s =====
PS C:\Users\ismael\Downloads\EstudiosPython\Librerias\SRC\PyTest> pytest prueba_pytest.py
=====
platform win32 -- Python 3.11.5, pytest-8.0.0, pluggy-1.4.0
rootdir: C:\Users\ismael\Downloads\EstudiosPython\Librerias\SRC\PyTest
collected 1 item

prueba_pytest.py .

=====
1 passed in 0.06s =====
PS C:\Users\ismael\Downloads\EstudiosPython\Librerias\SRC\PyTest> pytest prueba_pytest.py
=====
platform win32 -- Python 3.11.5, pytest-8.0.0, pluggy-1.4.0
rootdir: C:\Users\ismael\Downloads\EstudiosPython\Librerias\SRC\PyTest
collected 1 item

prueba_pytest.py F
=====
FAILURES =====
TestPersona.test_prueba
=====
self = <prueba_pytest.TestPersona object at 0x0000029FE5CCF350>

def test_prueba(self):
    # assert 0 == 0 # No da ningun error porque la condicion es correcta
>     assert 0 != 0
E     assert 0 != 0

prueba_pytest.py:10: AssertionError
=====
short test summary info =====
FAILED prueba_pytest.py::TestPersona::test_prueba - assert 0 != 0
=====
1 failed in 0.17s =====
PS C:\Users\ismael\Downloads\EstudiosPython\Librerias\SRC\PyTest>

```

Puedes utilizar más de una función dentro de la clase, porque cada una se interpretará como un objeto. “collected 2 ítems” nos indica que la clase de pruebas tiene dos funciones de pruebas unitarias declaradas, mientras que “collected 1 ítems” nos indica que la clase de pruebas tiene una función de pruebas unitarias declarada.

```

doc.txt prueba_pytest.py Persona.py
Liberias > SRC > PyTest > prueba_pytest.py > TestPersona > test_prueba
1 import pytest
2 # Si utilizas esta importacion pytest dara error, pytest no admite importaciones relativas
3 # from .mi_persona import Persona
4 from SRC.PyTest.Persona import Persona
5
6 class TestPersona:
7
8     def test_prueba(self):
9         assert 0 == 0 # No da ningun error porque la condicion es correcta
10        """
11            Da error porque la condicion no es correcta,
12            y si da error detiene la prueba unitaria en
13            donde se sucedio dicho error
14        """
15        # assert 0 != 0
16        assert 2 > 0
17
18    def test_constructor(self):
19        persona = Persona(nombre="Diego", edad=25)
20        assert persona.dar_nombre() == "Diego"
21        assert persona.dar_edad() == 25

```

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

(Librerías) PS C:\Users\ismae\Downloads\EstudiosPython\Librerías\SRC\PyTest> pytest prueba_pytest.py
=====
platform win32 -- Python 3.11.5, pytest-8.0.0, pluggy-1.4.0
rootdir: C:\Users\ismae\Downloads\EstudiosPython\Librerías\SRC\PyTest
collected 2 items

prueba_pytest.py ..

=====
= 2 passed in 0.05s =
● (Librerías) PS C:\Users\ismae\Downloads\EstudiosPython\Librerías\SRC\PyTest> pytest prueba_pytest.py
=====
platform win32 -- Python 3.11.5, pytest-8.0.0, pluggy-1.4.0
● rootdir: C:\Users\ismae\Downloads\EstudiosPython\Librerías\SRC\PyTest
collected 1 item

prueba_pytest.py .

=====
= 1 passed in 0.05s =
○ (Librerías) PS C:\Users\ismae\Downloads\EstudiosPython\Librerías\SRC\PyTest> █
```

Podemos realizar pruebas para comprobar constructores, getters, setters, si ciertas variables tienen ciertos contenidos o también para comprobar si se cumplen correctamente con ciertos cálculos.

The screenshot shows the PyCharm IDE interface. The top navigation bar has tabs for 'doc.txt', 'prueba_pytest.py', and 'Persona.py'. Below the tabs, the breadcrumb navigation shows 'Librerías > SRC > PyTest > prueba_pytest.py > TestPersona'. The main code editor displays a Python test script with two test functions: `test_constructor` and `test_asignacion`. The terminal at the bottom shows the command `pytest .\prueba_pytest.py` being run, and the output indicates 5 passed tests in 0.03 seconds.

```
doc.txt prueba_pytest.py Persona.py
Librerías > SRC > PyTest > prueba_pytest.py > TestPersona
22
23     # Podemos hacer pruebas de muchos tipos
24
25     def test_constructor(self): # Pruebas para ver si el constructor y sus getters funcionan
26         persona = Persona(nombre="Diego", edad=25)
27         assert persona.dar_nombre() == "Diego"
28         assert persona.dar_edad() == 25
29
30     def test_asignacion(self): # Pruebas para ver si los setters funcionan
31         persona = Persona(nombre="Diego", edad=25)
32         persona.asignar_nombre("Adriana")
33         persona.asignar_edad(28)
34
35         assert persona.dar_nombre() != "Diego"
36         assert persona.dar_edad() != 25
37

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
Python - PyTest + ▾

(librerías) PS C:\Users\ismael\Downloads\EstudiosPython\Librerías\SRC\PyTest> pytest .\prueba_pytest.py
=====
platform win32 -- Python 3.11.5, pytest-8.0.0, pluggy-1.4.0
rootdir: C:\Users\ismael\Downloads\EstudiosPython\Librerías\SRC\PyTest
collected 5 items

prueba_pytest.py ....
[100%]

=====
5 passed in 0.03s =====
(librerías) PS C:\Users\ismael\Downloads\EstudiosPython\Librerías\SRC\PyTest>
```

Fuente: <https://misovirtual.virtual.uniandes.edu.co/codelabs/tutorial-PyTest/index.html#0>

Labor 7: Librería Scapy

He mirado un poco la librería Scapy, no he llegado a profundizar del todo.

```

prueba_scapy.py 9+  doc.txt
Librerías > SRC > test_scapy > prueba_scapy.py > ...
1  from scapy.all import *
2
3  packet = IP()/TCP()
4  print(Ether()/packet)
5
6
7  p = Ether()/IP(dst="www.secdev.org")/TCP()
8  print(p.summary())
9
10 print(p.dst) # first layer that has an src field, here Ether
11 print(p[IP].src) # explicitly access the src field of the IP layer
12
13 # sprintf() is a useful method to display fields
14 print(p.strftime("%Ether.src% > %Ether.dst%\n%IP.src% > %IP.dst%"))
15
16
PROBLEMAS 12 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
PS C:\Users\ismae\Downloads\EstudiosPython> & c:/Users/ismae/Downloads/EstudiosPython/Librerías/Scripts/python.exe c:/Users/ismae/Downloads/EstudiosPython/Librerías/SRC/test_scapy/prueba_scapy.py
Ether / IP / TCP 127.0.0.1:ftp_data > 127.0.0.1:http S
Ether / IP / TCP 192.168.1.34:ftp_data > Net("www.secdev.org/32"):http S
44:48:b9:6a:c1:60
192.168.1.34
48:e7:da:54:a9:cf > 44:48:b9:6a:c1:60
192.168.1.34 > Net("www.secdev.org/32")

```

Fuentes:

<https://scapy.net/>

<https://github.com/secdev/scapy/blob/master/doc/notebooks/Scapy%20in%2015%20minutes.ipynb>

<https://codingninjablogs.tech/tryhackme-python-for-pentesters-47b7ce525b90>

Planes para la próxima semana

Seguiré trabajando con la librería Scapy. Iré descubriendo nuevas librerías de Python, trabajaré en la API de Python para seguir descubriendo nuevas funcionalidades, e investigar aún más Excel.

Si da tiempo, investigaré temas de ciberseguridad.