

Reporte semanal - Semana 8

22/03/2024

—

Ismael Montoro Peñasco
Fundación Goodlob

Acciones llevadas a cabo.....	2
Avances realizados.....	2
Labor 1: Practica con SQLite.....	2
Labor 2: Acceso a REST API.....	3
Labor 3: Conceptos de Python.....	4
Labor 3.1: Clase base abstracta.....	4
Labor 3.2: Manejo de archivos binarios.....	5
Labor 3.3: Tipado estatico.....	6
Labor 3.4: Scope.....	7
Labor 3.5: Tipo estático.....	8
Labor 3.6: Funciones anidadas.....	8
Labor 3.7: Iteradores con funciones.....	9
Labor 3.8: Iterador con funciones.....	10
Labor 4: Inteligencia artificial.....	11
Labor 5: Scapy.....	12
Planes para la próxima semana.....	13

Acciones llevadas a cabo

Seguiré estudiando Scapy e [Inteligencia Artificial](#), dando prioridad a la Inteligencia Artificial.

Avances realizados

Labor 1: Practica con SQLite

He practicado el acceso a la librería SQLite que viene incluida en la API de Python.

```
pokemon_API.py X
Librerías > SRC > pokemon_API.py > ...
1 import requests, sqlite3
2
3 try:
4
5     conexion = sqlite3.connect("Librerías\\SRC\\Informacion_Pokemon.sqlite")
6
7     conexion.execute("""
8         Create table Movimiento (
9             Nombre text primary key,
10            Descripcion text,
11            Puntos_Poder integer,
12            Precision integer,
13            tipo text CHECK (tipo IN ("Agua", "Bicho", "Dragón", "Eléctrico", "Fantasma",
14                                     "Fuego", "Hada", "Hielo", "Lucha", "Normal", "Planta",
15                                     "Psíquico", "Roca", "Sinistro", "Tierra", "Veneno",
16                                     "Volador", "???")),
17            Objetivo text
18        )
19    """)
20
21    conexion.execute("""
22        Create table Movimiento_Ataque (
23            Nombre text primary key,
24            Potencia integer,
25            Efecto text
26        )
27    """)
28
```

Labor 2: Acceso a REST API

He practicado el acceso a una REST API de Pokémon con la librería requests de Python.

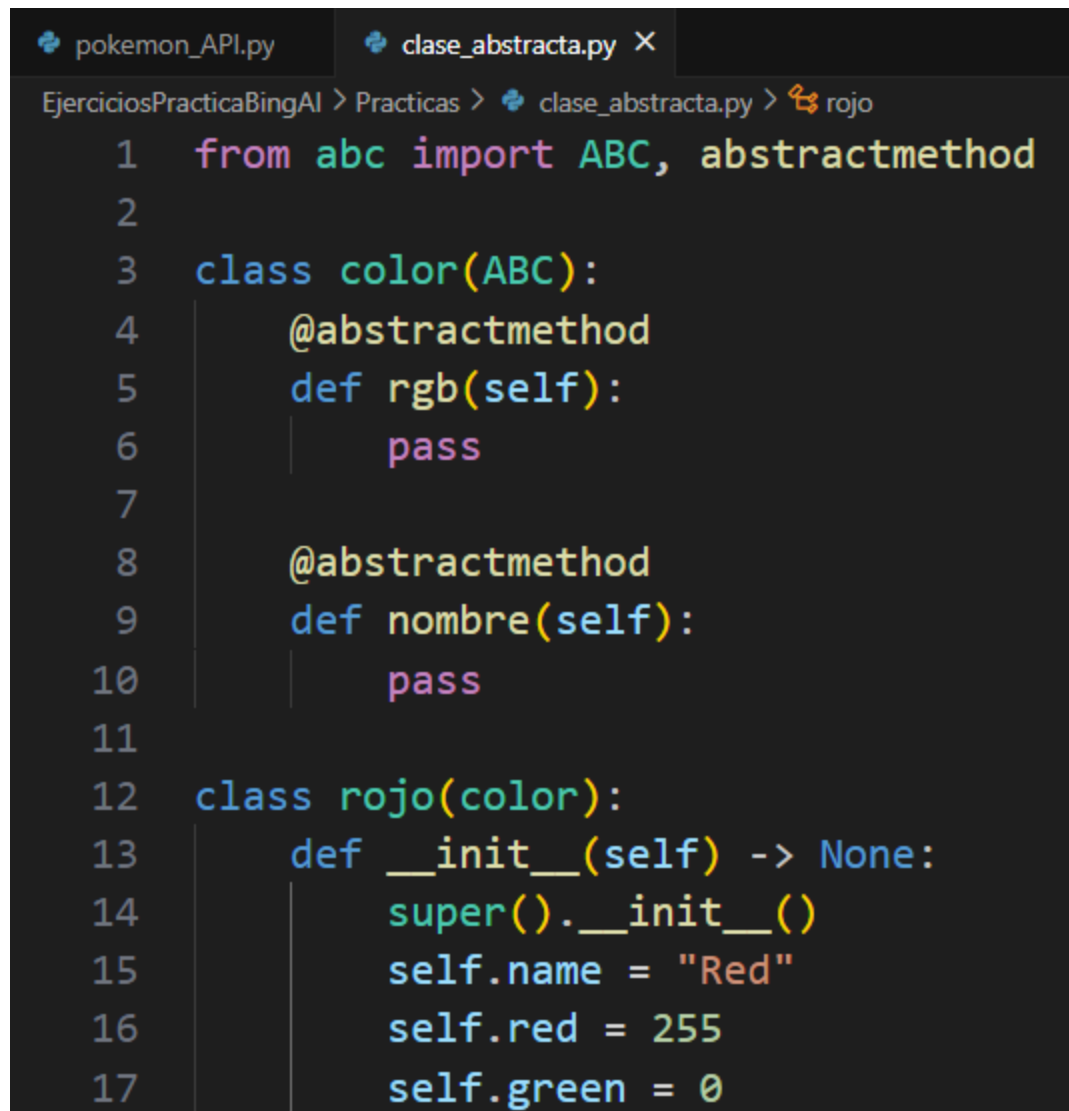
```
pokemon_API.py X
Librerías > SRC > pokemon_API.py > ...
69 for id in range(500, 505 + 1):
70     movimiento = f"https://pokeapi.co/api/v2/move/{id}"
71     info_movimiento = requests.get(movimiento).json()
72
73     name = info_movimiento['names'][5]['name']
74     tipo = info_movimiento['type']['name']
75     pp = info_movimiento['pp']
76     poder_ataque = info_movimiento['power']
77     precision = info_movimiento['accuracy']
78     prioridad = info_movimiento['priority']
79     prob_efecto_secundario = info_movimiento['effect_chance']
80     prob_golpe_critico = info_movimiento['meta']['crit_rate']
81     num_golpes_min = info_movimiento['meta']['min_hits']
82     num_golpes_max = info_movimiento['meta']['max_hits']
83     duracion_min = info_movimiento['meta']['min_turns']
84     duracion_max = info_movimiento['meta']['max_turns']
85     poder_drenaje = info_movimiento['meta']['drain']
86     objetivo = info_movimiento['target']['name']
87
```

Labor 3: Conceptos de Python

He practicado algunos conceptos de programación dentro del lenguaje de Python.

Labor 3.1: Clase base abstracta

He practicado el concepto de clase base abstracta en Python, con el fin de crear interfaces en Python, una interfaz en Java es como una especie de plantilla para generar clases con una estructura base en común.



```
pokemon_API.py  clase_abstracta.py X
EjerciciosPracticaBingAI > Practicas > clase_abstracta.py > rojo
1  from abc import ABC, abstractmethod
2
3  class color(ABC):
4      @abstractmethod
5      def rgb(self):
6          pass
7
8      @abstractmethod
9      def nombre(self):
10         pass
11
12  class rojo(color):
13      def __init__(self) -> None:
14          super().__init__()
15          self.name = "Red"
16          self.red = 255
17          self.green = 0
```

En el momento en el que una clase, que hereda de una clase abstracta, y esta clase no tenga todos los métodos declarados en la clase abstracta, dará un error.

```

PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

● PS C:\Users\ismae\Downloads\EstudiosPython> & c:/Users/ismae/Downloads/EstudiosPython/abstracta.py
Red 255 0 0
Azul (255, 0, 0)
Green 0 255 0
● PS C:\Users\ismae\Downloads\EstudiosPython> & c:/Users/ismae/Downloads/EstudiosPython/abstracta.py
Traceback (most recent call last):
  File "c:\Users\ismae\Downloads\EstudiosPython\EjerciciosPracticaBingAI\Practicas\abstracta.py", line 10, in <module>
    red = rojo()
          ^^^^^^
TypeError: Can't instantiate abstract class rojo with abstract method nombre
● PS C:\Users\ismae\Downloads\EstudiosPython> & c:/Users/ismae/Downloads/EstudiosPython/abstracta.py
Red 255 0 0
Azul (255, 0, 0)
Green 0 255 0
○ PS C:\Users\ismae\Downloads\EstudiosPython> █

```

Labor 3.2: Manejo de archivos binarios

He practicado el manejo de archivos binarios en Python.

```

pokemon_API.py  archivos_binarios.py X
EjerciciosPracticaBingAI > Practicas > archivos_binarios.py > ...
1
2 array_bytes = None
3 with open("EjerciciosPracticaBingAI\\Practicas\\miktex.exe", "rb") as lectura:
4     array_bytes = lectura.read()
5     print(array_bytes[0:155])
6
7 with open("EjerciciosPracticaBingAI\\Practicas\\miktex_2.exe", "wb") as escritura:
8     escritura.write(array_bytes)

```

Labor 3.3: Tipado estatico

He practicado el tipado estático en Python.

EjerciciosPracticaBingAI > Practicas > estructura_typing.py > ...

```
1  # La libreria typing permite usar tipado estatico en Python
2  from typing import Dict, List, Set, Tuple
3
4  ids: Tuple[int] = (3121232, 5234234, 5435635, 7674565)
5  ids: Set[int] = {3121232, 5234234, 5435635, 7674565}
6  nombres: List[int] = ("Lucia", "Daniel", "Samuel", "Adrian")
7
8  empleados: Dict[int, str] = {
9      3121232: "Lucia",
10     5234234: "Daniel",
11     5435635: "Samuel",
12     7674565: "Adrian"
13 }
14
15 empresa: List[Dict[int, str]] = [
16     {
17         3121232: "Lucia",
18         5234234: "Daniel",
19         5435635: "Samuel",
20         7674565: "Adrian"
21     },
22     {
23         5164495: "Andres",
24         5324452: "Alejandro",
25         4848566: "Carla",
26         8390128: "Rafa"
27     },
28     {
29         4532644: "Alex",
30         5326432: "Jose",
```

Labor 3.4: Scope

He practicado el concepto de Scope en Python con ejemplos muy sencillos y los he incluido en los ejercicios de la semana.

```
# Variables globales
variable_global = 2
def devolver_variable_global():
    return variable_global

print("Variable", variable_global)
print("Funcion", devolver_variable_global())

def variable_global_funcion():
    global id # Puedes usar la palabra clave global para declarar
    id = 20000
    return id

print("Funcion", variable_global_funcion())
print("Variable", id)

# Variables dentro de funciones
def devolver_variable():
    v_funcion = 5
    return v_funcion

print(devolver_variable())
try:
    print(v_funcion)
except NameError:
    print("La variable de la funcion es inaccesible.")

# Variables dentro de bloques
condicion = False
if condicion:
    v_bloque = 4
    print(v_bloque)
```


Labor 3.5: Tipo estático

He practicado también algunos tipos de tipado estático en Python.

6º Usa la librería typing para practicar el tipado estatico en

```
# https://dev.to/geoannycode/conceptos-avanzados-de-python-pa
from typing import Dict, List, Set, Tuple

ids: Tuple[int] = (3121232, 5234234, 5435635, 7674565)
ids: Set[int] = {3121232, 5234234, 5435635, 7674565}
nombres: List[str] = ["Lucia", "Daniel", "Samuel", "Adrian"]

empleados: Dict[int, str] = {
    3121232: "Lucia",
    5234234: "Daniel",
    5435635: "Samuel",
    7674565: "Adrian"
}

empresa: List[Dict[int, str]] = [
    {
        3121232: "Lucia",
        5234234: "Daniel",
        5435635: "Samuel",
        7674565: "Adrian"
    }
]
```

Labor 3.6: Funciones anidadas

He creado funciones anidadas en Python.

8º Crea una funcion anidada (Nested Functions).

```
# https://dev.to/geoannycode/conceptos-avanzados-de-python-pa
def mi_funcion(num: int):
    resultado = num + 5
    def mi_funcion_anidada(result: int):
        resultado_anidado = result ** 7
        return resultado_anidado

    return mi_funcion_anidada(resultado)

print(mi_funcion(2))
```

823543

Labor 3.7: Iteradores con funciones

He trabajado con iteradores en Python con las funciones `iter()` y `next()`.

9º Trabaja con iteradores en Python con la función `iter()` y la función `next()`.

```
# https://dev.to/geoannycode/conceptos-avanzados-de-python-para-ser-un-profesic
lista = [3121232, 5234234, 5435635, 7674565]
iterador = iter(lista)

print(type(iterador))

try:
    for _ in range(5):
        elemento_actual = next(iterador)
        print(elemento_actual)
except StopIteration:
    print("No hay más elementos")
```

```
<class 'list_iterator'>
3121232
5234234
5435635
7674565
No hay más elementos
```

Labor 3.8: Iterador con funciones

He practicado el uso de la palabra reservada yield de python para crear una función iterable.

10º Utiliza la palabra yield para generar una funcion que

```
[2]: # https://dev.to/geoannycode/conceptos-avanzados-de-python
def contar(inicio: int):
    inicio += 1
    yield inicio # yield es como return, pero con la dife
    inicio += 1
    yield inicio
    inicio += 1
    yield inicio
    inicio += 1
    yield inicio
    inicio += 1
    yield inicio

valor = 5
iterador = contar(valor)

try:
    for _ in range(valor + 1):
        valor_actual = next(iterador)
        print(valor_actual)
except StopIteration:
    print("La funcion no devuelve mas valores")
```

6
7
8
9
10
La funcion no devuelve mas valores

Labor 4: Inteligencia artificial

He avanzado con las clases de Introducción a la Inteligencia Artificial ofrecida por la universidad de [Harvard](#).

Introducción a la Inteligencia Artificial

Con la inteligencia artificial podemos realizar todo tipo de tareas como procesar patrones de datos de manera mucho más eficiente, reconocimiento de rostros, hacer entender a las máquinas el lenguaje natural de los humanos cuando les hablan, es decir, puedes que tu teléfono móvil te entienda a ti, hacer que una máquina juegue a un juego mejor que los humanos, y la lista sigue y sigue.

Dentro del campo de la Inteligencia Artificial hay muchos tipos de técnicas que nos permitirán de alguna manera dotar a los ordenadores de Inteligencia, y así que la máquina pueda realizar toda clase de tareas.

Para poder desarrollar e implementar una inteligencia artificial tenemos que realizar una serie de pasos, a modo de guía paso a paso, tenemos un listado de todos los conceptos por los que hay que pasar antes de llegar a la implementación de una Inteligencia Artificial.

Insertar un Roadmap de IA con los nombres de los siguientes conceptos

1. Search → El primer problema que debemos afrontar a la hora de desarrollar una Inteligencia Artificial funcional es, como haríamos para que la IA **busque** soluciones ante un problema que se presente, sin importar el tipo de problema, como por ejemplo, obtener indicaciones para llegar con un coche desde un punto A a un punto B, o tratar de descubrir cómo se juega a un juego, ya sea la tres en raya, el ajedrez, etc. Search es búsqueda en Inglés

Labor 5: Scapy

He practicado los valores por defecto en Scapy, resulta que los paquetes de Scapy tienen valores por defecto, al pasarle los parametros al paquete modificas los valores por defecto, sin embargo si utilizas la palabra reservada del de python sobre un parámetro del paquete puedes devolverle su valor por defecto.

```
154
155  paquete = IP(src="192.168.1.1", ttl=50)
156
157  print(paquete.src)
158  del paquete.src
159  print(paquete.src)
160
161  print(paquete.ttl)
162  del paquete.ttl
163  print(paquete.ttl)
```

Podemos realizar el mismo procedimiento con paquetes con varios protocolos.

```
165  paquete = IP(src="192.168.1.1", ttl=50)/ICMP(id=0x55)
166
167  print(paquete[IP].src)
168  del paquete[IP].src
169  print(paquete[IP].src)
170
171  print(paquete[IP].ttl)
172  del paquete[IP].ttl
173  print(paquete[IP].ttl)
174
175  print(paquete[ICMP].id)
176  del paquete[ICMP].id
177  print(paquete[ICMP].id)
```

Planes para la próxima semana

Seguiré estudiando Scapy e [Inteligencia Artificial](#), dando prioridad a la Inteligencia Artificial. A ser posible también estudiar otros temas relacionados con Python.