

Reporte Quincenal

19/04/2024

Ismael Montoro Peñasco
Fundación Goodlob

Acciones llevadas a cabo.....	2
Avances realizados.....	2
Curso de Inteligencia Artificial.....	2
Lunes 25 de Marzo.....	2
Día 26 y 27 de Marzo.....	2
Domingo 31 de Marzo.....	2
Día 2 y 3 de Abril.....	2
Curso Python de SEPE.....	3
Examen Intermedio.....	4
Investigaciones.....	5
Labor 1: Scapy en Kali.....	5
Labor 2: Prácticas de Poké Api y SQLite3.....	6
Labor 3: Masterclass de GIT.....	7
Labor 4: Métodos de ejecución y compilación.....	8
Labor 5: Algoritmo de codificación.....	9
Labor 6: Leer bit a bit.....	10
Labor 7: Generador de abecedarios.....	10
Labor 8: Configuraciones al repositorio.....	11
Planes para la próxima semana.....	11

Acciones llevadas a cabo

He realizado mis investigaciones en Python, sobre conceptos básicos de la programación, me he apuntado a un curso de Python proveído por el SEPE para poder realizar la certificación del PCAP y PCPP, también he ido avanzando con el curso de Inteligencia Artificial que me ofrece la universidad de Harvard.

AVISO: el curso de la universidad de Harvard, es un curso sin modalidad Streaming, es decir, son una serie de vídeos subidos a la plataforma EDX, es decir, no tengo un horario fijo por lo que puedo ir haciendo el curso poco a poco, y centrarme en el curso del SEPE que si tengo un horario fijo de 8:45 am a 15:00 pm.

→ Curso de Python:

<https://sepe.pue.es/cursos/programacion-avanzada-python-curso-oficial-pcap-pcpp-1-python-institute/ed1546>

→ Curso de Inteligencia Artificial:

<https://www.edx.org/es/learn/artificial-intelligence/harvard-university-cs50-s-introduction-to-artificial-intelligence-with-python>

Avances realizados

Curso de Inteligencia Artificial

Lunes 25 de Marzo

He mirado conceptos básicos del tema 1 en el Curso, he ido realizando apuntes para poder estudiar mejor.

Día 26 y 27 de Marzo

He avanzado un poco con el tema 1 (las búsquedas), me han estado comentando sobre los múltiples problemas que tiene una IA a la hora de realizar búsquedas.

Domingo 31 de Marzo

He avanzado un poco con el tema 1 (las búsquedas), por el momento he podido ver en toda su magnitud los problemas de un IA a la hora de realizar una búsqueda, y algunas soluciones para solventarlas.

Día 2 y 3 de Abril

He avanzado un poco con el tema 1 con respecto a las soluciones o algoritmos que hay sobre las búsquedas a nivel conceptual.

Curso Python de SEPE

En este curso he repasado conceptos algunos conceptos que vi en Nanfor, además de ver conceptos nuevos que no sabía sobre Python.

- He instalado PyCharm dado que Visual Studio Code es insuficiente para proyectos pequeños.
- He visto operadores de Python a un nivel más extenso, junto con la prioridad de cada uno de ellos.
- He visto colecciones.
 - Las listas.
 - Las tuplas.
 - Los sets junto con las operaciones de conjuntos.
 - Los diccionarios.
- He visto conceptos interesantes como las listas comprimidas.
- He visto las excepciones de Python tanto a nivel básico como a nivel avanzado.
- He visto las aserciones.
- He visto una buena cantidad de funciones nativas de Python, algunos ejemplos que podría nombrar son la librería string, math, random y platform.
- He visto lo que son los paquetes de Python, y las diferentes maneras de importar una librería.
- He visto Programación Orientada a Objetos en profundidad.
 - He visto lo que es el MRO.
 - He visto las diferencias entre Python y Java, ejemplo en Python no existe la sobrecarga de métodos.
 - He visto la sobrescritura con los métodos `__str__` y `__del__`.
 - He visto sus modos de acceso que aunque no lo parezca, existen los modos de acceso `protected`, y no solo `public` o `private`.
 - He visto el modo de acceso estático.
- He visto cómo crear excepciones personalizadas.
- He visto funciones como `filter`, `map` y `reduce`, junto con las lambdas.
- He visto ficheros, y sus modos de apertura.
- He visto los módulos `os`, `datetime`, `time` y `calendar`.
- He visto cómo crear iteradores con funciones de Python.
- He visto también de forma más avanzada el módulo `pandas`.

Examen Intermedio

Realice el examen intermedio de Python, la nota sacada es un sobresaliente.

 Español - Internacional (es) ▾

TR-8725 - IFCD093PO - MACHINE LEARNING APLICADO USANDO PYTHON

[Área personal](#) / [Cursos](#) / [TR-8725](#) / [Examen intermedio](#) 📅 MARTES 16/04/24 / [Examen Intermedio](#)

Navegación por el cuestionario

1	2	3	4	5	6	7	8
✓	✓	✓	✓	✓	✓	✓	✓
9	10						
✓	✓						

[Finalizar revisión](#)

Comenzado el	martes, 16 de abril de 2024, 14:02
Estado	Finalizado
Finalizado en	martes, 16 de abril de 2024, 14:09
Tiempo empleado	7 minutos 51 segundos
Calificación	10,00 de 10,00 (100%)

 Español - Internacional (es) ▾

TR-8725 - IFCD093PO - MACHINE LEARNING AF

[Área personal](#) / [Cursos](#) / [TR-8725](#) / [Examen intermedio](#) 📅 MARTES 16/04/24 / [Examen Intermedio](#)

Navegación por el cuestionario

1	2	3	4	5	6	7	8
✓	✓	✓	✓	✓	✓	✓	✓
9	10						
✓	✓						

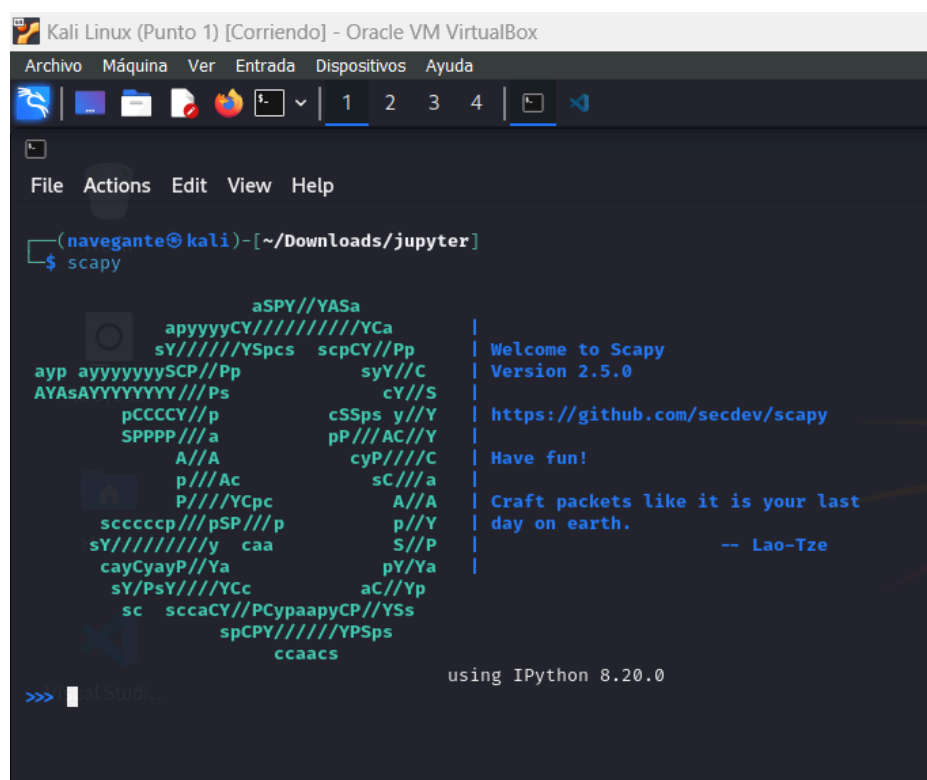
[Finalizar revisión](#)

Comenzado el	martes, 16 de abril de 2024, 14:02
Estado	Finalizado
Finalizado en	martes, 16 de abril de 2024, 14:09
Tiempo empleado	7 minutos 51 segundos
Calificación	10,00 de 10,00 (100%)

Investigaciones

Labor 1: Scapy en Kali

Me di cuenta de que desarrollar con la librería Scapy en Windows es bastante más difícil debido a lo poco integrado que está todo el ecosistema de dependencias, así que instalé Kali Linux en VirtualBox y ahora desarrollo Scapy desde ahí e incluso he podido instalar el paquete PyX, una dependencia que no me funcionaba en Windows.



```

Kali Linux (Punto 1) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
File  Actions  Edit  View  Help
(navegante@kali)-[~/Downloads/jupyter]
$ scapy

      aSPY//YASa
      apyyyyCY/////////YCa
      sy////////YSpCs  scpCY//Pp
ayp ayyyyyySCP//Pp      syY//C
AYAsAYYYYYYYY///Ps      cY//S
      pCCCCY//p      cSSps y//Y
      SPPPP///a      pP///AC//Y
      A//A      cyP///C
      p///Ac      sC///a
      P///YCpc      A//A
      scccccp///pSP///p      p//Y
      sY/////////y  caa      S//P
      cayCyayP//Ya      pY/Ya
      sY/PsY///YCc      aC//Yp
      sc  sccaCY//PCyapaPyCP//YSs
      spCPY/////////YPSps
      ccaacs

      |
      | Welcome to Scapy
      | Version 2.5.0
      | https://github.com/secdev/scapy
      | Have fun!
      | Craft packets like it is your last
      | day on earth.
      | -- Lao-Tze
      |

using IPython 8.20.0
  
```

Labor 2: Prácticas de Poké Api y SQLite3

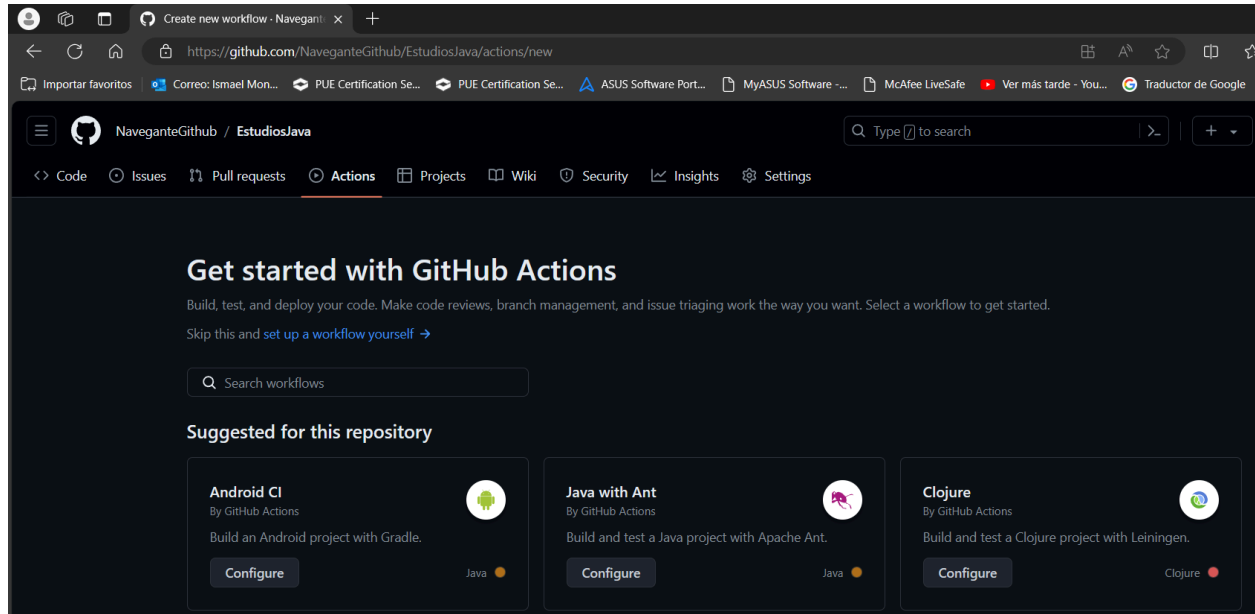
He estado practicando los microservicios con la Poké Api y SQL con Python 3 en SQLite3.

```
pokemon_API.py x
Librerías > SRC > Pokemon > pokemon_API.py > [?] conexion

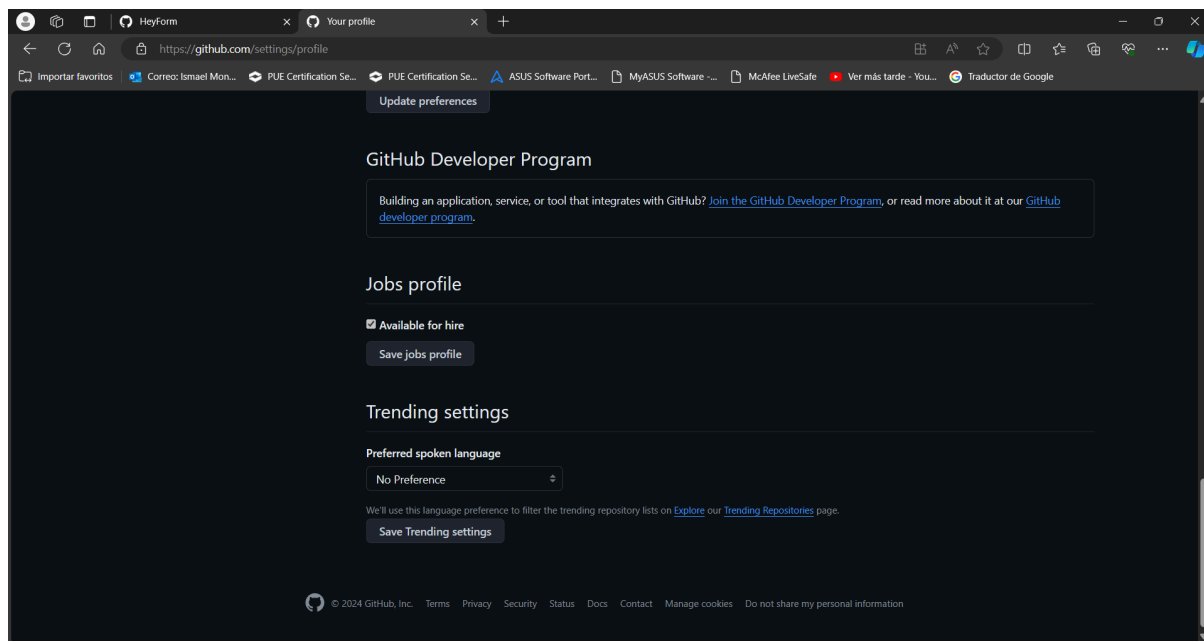
58 try:
59
60     conexion = sqlite3.connect(bd)
61
62     for id in range(1, 915 + 1):
63         movimiento = f"https://pokeapi.co/api/v2/move/{id}"
64         info_movimiento = requests.get(movimiento).json()
65
66         name = None
67         try:
68             num_nombres = len(info_movimiento['names'])
69             if num_nombres > 5:
70                 name = info_movimiento['names'][5]['name']
71             elif num_nombres > 0:
72                 name = info_movimiento['names'][0]['name']
73
74         except IndexError:
75             conexion.execute(consulta,
76                               (str(id), None, None, None, None, None, None, None, None, None,
77                                None, None, None, None, None, None, None))
78             print("Movimiento no encontrado.")
79             continue
80
81         descripcion = None
82         descripcion_efecto = None
83         try:
84             if len(info_movimiento['flavor_text_entries']) > 0:
85                 posicion = 0
```

Labor 3: Masterclass de GIT

He asistido a la masterclass de Git y Repositorios impartida por Roman. Lo más interesante que aprendí de esta Masterclass son las Actions, que sirven para poder desplegar el software de un repositorio en un servicio como AWS, Azure, Alibaba, etc...



También una cosa interesante que aprendí que me sorprendió mucho es que Github se puede utilizar como una bolsa de empleo.



Labor 4: Métodos de ejecución y compilación

He investigado sobre los métodos `eval()`, `exec()` y `compile()`, para realizar la ejecución de un código que esté declarado dentro de una cadena de texto.

```
1  # https://es.stackoverflow.com/questions/309128
2  # https://stackoverflow.com/questions/44704229/
3  # https://stackoverflow.com/questions/6883033/a
4  '''
5  eval() es una funcion de python que me sirve
6  para ejecutar UNA sola linea de texto como si
7  fuera un codigo de Python
8  '''
9  cadena = "2 + 5 ** 2"
10 resultado_eval = eval(cadena)
11 print(resultado_eval)
12
13 #####
14
15 cadena = "print(2 + 5 ** 2)"
16 eval(cadena)
17
18 #####
19
20 '''
21 exec() es una funcion de python que me sirve
22 para ejecutar VARIAS sola linea de texto como
23 si fueran un codigo de Python
24 '''
25
26 codigo = f"""
27 x = 5
28 print(2 + x)
29 """
30 exec(codigo)
31
```

Labor 5: Algoritmo de codificación

He creado un nuevo formato de codificación llamado alfablock una codificación que tiene como objetivo principal, convertir cualquier dato en un bloque de caracteres alfabéticos.

```

152 def codificar_a_alfablock2(self, texto = "hello") -> str:
153     """
154     Esta codificacion me permitira codificar texto
155     alfanumerico con metacaracteres a un formato
156     alfabetico basico en mayusculas, cada caracter
157     se codificara en un bloque de 2 letras en mayusculas
158     """
159     abecedario = {
160         "A": "AA", "B": "AB", "C": "AC", "D": "AD", "E": "AE",
161         "F": "AF", "G": "AG", "H": "AH", "I": "AI", "J": "AJ",
162         "K": "AK", "L": "AL", "M": "AM", "N": "AN", "Ñ": "AÑ",
163         "O": "AO", "P": "AP", "Q": "AQ", "R": "AR", "S": "AS",
164         "T": "AT", "U": "AU", "V": "AV", "W": "AW", "X": "AX",
165         "Y": "AY", "Z": "AZ",
166         "a": "BA", "b": "BB", "c": "BC", "d": "BD", "e": "BE",
167         "f": "BF", "g": "BG", "h": "BH", "i": "BI", "j": "BJ",
168         "k": "BK", "l": "BL", "m": "BM", "n": "BN", "ñ": "BÑ",
169         "o": "BO", "p": "BP", "q": "BQ", "r": "BR", "s": "BS",
170         "t": "BT", "u": "BU", "v": "BV", "w": "BW", "x": "BX",
171         "y": "BY", "z": "BZ",
172         "0": "CA", "1": "CB", "2": "CC", "3": "CD", "4": "CE",
173         "5": "CF", "6": "CG", "7": "CH", "8": "CI", "9": "CJ",
174         " ": "CK", "=": "CL", "!": "CM", "\"": "CN", "'": "CÑ",
175         "$": "CO", "€": "CP", "%": "CQ", "/": "CR", "&": "CS",
176         "(": "CT", ")": "CU", ".": "CV", "-": "CW", "_": "CX",

```

Labor 6: Leer bit a bit

He aprendido a leer los bits de un archivo.

```
1 def leer_unos_y_ceros(archivo):
2     with open(archivo, 'rb') as f:
3         while True:
4             byte = f.read(1) # Leer un byte del ar
5             if not byte: # Si no hay más bytes par
6                 break
7             # Convertir el byte a su representación
8             # bits = bin(ord(byte))[2:] # .zfill(8)
9             bits_dec = ord(byte)
10            # print(bits) # Imprimir los bits en u
11            print(bits_dec)
12
13 # Ejemplo de uso
14 archivo = "Src\\pruebas\\abecedario_unicode.txt" #
15 leer_unos_y_ceros(archivo)
16
```

Labor 7: Generador de abecedarios

He estudiado la codificación Unicode, y he averiguado cómo tratar con dicha codificación.

```
5 """
6 https://home.unicode.org/technical-quick-start-guide/
7 https://www.unicode.org/versions/stats/charcountv15\_1.html
8 https://www.unicode.org/versions/stats/charcountv13\_0.html
9 """
10
11 # v1 = 96000
12 v1 = 150000
13 try:
14     for i in range(v1):
15         contador_str = str(contador).zfill(5)
16         bloque = "".join([letras[int(num)] for num in contador_str])
```

Labor 8: Configuraciones al repositorio

He configurado mi repositorio de Github para que esté más organizado, con ello he creado un repositorio especial para subir mis apuntes de Python, y recursos que he ido recolectando sobre Python.

<https://github.com/NaveganteGithub/EstudiosPython>

Planes para la próxima semana

Me centraré en el curso de PCAP y PCPP, para conseguir el voucher para realizar el examen del PCAP, y si tengo tiempo, seguiré con el curso de Inteligencia Artificial.