



Estudio de programación en Python y sobre CS en general

FECHA: 12/09/2021

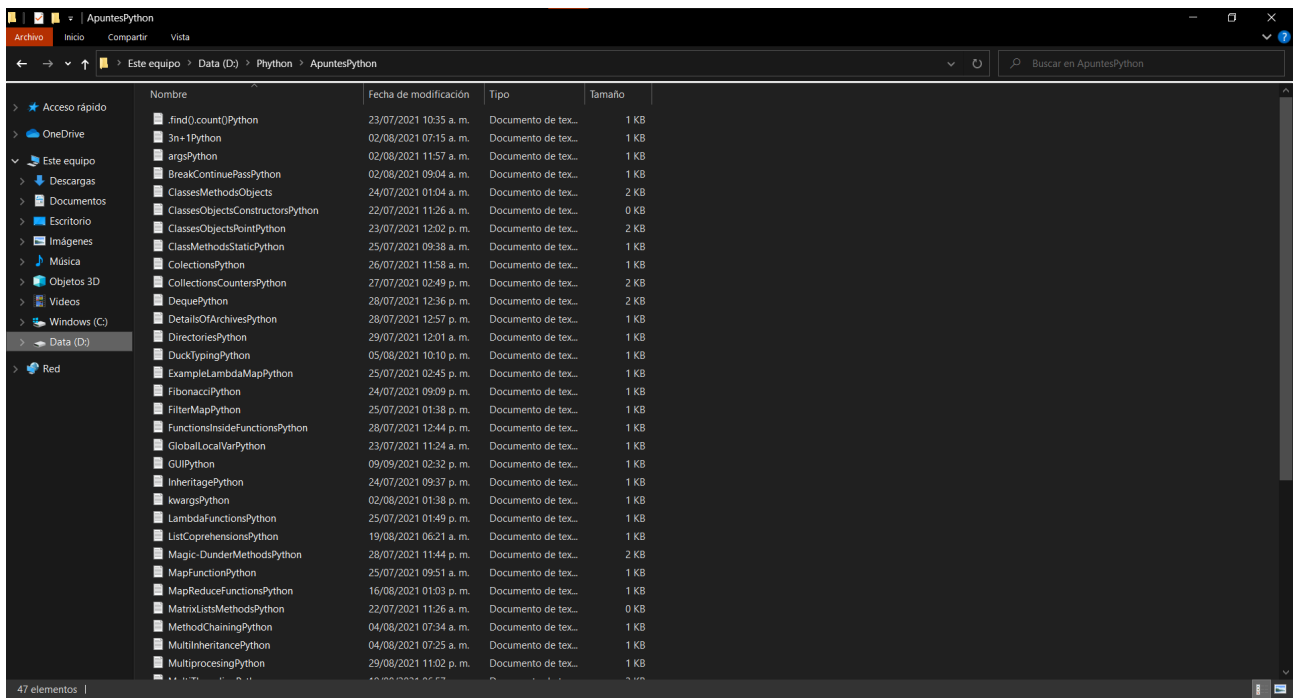
OBJETIVO CAS: Mejorar mis capacidades de estudio dirigido y motivado.

OBJETIVO PERSONAL: Aprender más sobre el funcionamiento de las computadoras y aprender a programar a nivel avanzado.

HORAS A REALIZAR: 42 horas de creatividad

En el verano con la clase de Computer Science, me dí un segundo intento con la programación, la cual ya la había probado hacía un par de años pero que no terminé de entender y de poder estudiar por la falta de una buena computadora y la limitante del idioma, ya que la gran mayoría de información que se consigue en internet está en inglés y es sino hasta hace poco que comencé a ser fluido en el lenguaje. Pero este reencuentro me hizo saber que la computación y la programación son muchas veces mi flow, un término que se nos presentó en la clase de tutorio, y que para el cual yo no tenía una respuesta tan clara. Esta misma pasión que le he tomado a la programación ha llevado a que he remplazado los videojuegos y la mayor parte de las series que veía para dedicarme mucho más a la programación, llegando a realizar en promedio unos 50 min de estudio al día desde la penúltima semana de julio, dejando un total de 42 horas de creatividad.

La programación me ha demostrado lo capaz que soy de estudiar de manera dedicada y centrada por gusto, no para prepararse para un concurso o para un examen, y me han mostrado un nuevo modo de entender y redactar mis pensamientos, dónde no va un tema solo de programación que me haya evocado poco interés, por lo mismo he realizado bastantes proyectos y apuntes, dejo a continuación una captura de pantalla de mis apuntes de Python, ejemplos de los mismos, y una parte de mi último proyecto, un juego de gato(tic-tac-toe) con varios tipos de jugadores y una computadora invencible:



Directorio de Apuntes de Python que he realizado para ayudarme a repasar temas de manera más concisa.

```
PassWordGeneratorPython: Bloc de notas
Archivo Edición Formato Ver Ayuda
#Password Verificador
def requisitos(psswr):
    low_alph = "abcdefghijklmnopqrstuvwxyz"
    up_alph = low_alph.upper()
    num = "0123456789"
    min = 0
    max = 0
    cif = 0
    ret = False #Estado de validación de contraseña
    '''if psswr.lower() == "skip":
        break'''
    for char in list(psswr):
        if low_alph.find(char) >=0:
            min +=1
        elif up_alph.find(char) >=0:
            max +=1
        elif num.find(char) >=0:
            cif +=1
        else:
            pass
    if min < 4:
        print("Usa más minúsculas.")
    elif max<2:
        print("Usa más mayúsculas.")
    elif cif<4:
        print("Usa más dígitos.")
    else:
        print("Contraseña Valida")
        ret = True

    return ret

veri = False
while not veri:
    psswr = input("Password: ")
    veri = requisitos(psswr)
print("Contraseña Guardada.")
```

Apunte sobre las comprehensions de los objetos tipo string con un pequeño ejercicio de revisión de contraseñas.

The screenshot shows the Visual Studio Code interface with a Python project named 'juego.py'. The Explorer sidebar on the left shows the project structure, including files like 'gato.py', 'juego.py', 'map_function.py', 'testing.py', and 'workspace.code-work...'. The main editor displays the code for 'juego.py', which is a Python class named 'Gato' that implements a game logic. The code includes imports for 'Usuario', 'RMG', and 'Terminator', and a 'time' module. The class has an 'init' method, a 'print tablero' method, and a 'tabla' static method. The terminal at the bottom shows the output of the program, which is a game of Gato (Tic-Tac-Toe) played by a human (h) and a computer (c). The terminal output shows the game board, the player's move, and the computer's response.

```
1 from gato import Usuario, RMG, Terminator
2 import time
3
4 class Gato:
5     def __init__(self):
6         self.tablero = [" " for i in range(9)] #Tomamos al tablero como una lista de 9, aunque sea un 3x3
7         self.ganador = None #Saber si hay un ganador
8
9     def print tablero(self):
10         for fila in [self.tablero[i*3:(i+1)*3] for i in range(3)]:
11             print(" | ".join(fila)+" |")
12
13
14     @staticmethod
15     def tabla():
16         numeros = [[str(i) for i in range(j*3, (j+1)*3)] for j in range(3)]
17         for fila in numeros: print(" | ".join(fila)+" |")
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/pscore6>

PS D:\VSCode Projects> & 'C:\Program Files\Python39\python.exe' 'c:\Users\ainbc\.vscode\extensions\ms-python.python-2021.9.1230869389\pythonFiles\lib\python\debugpy\launcher' '66921' '-.' 'd:\VSCode Projects\PythonVSC\juego.py'

Juego de Gato por Omikron
Seleccione el tipo de jugador de las letras:
(h)Humano or (c)Computadora (t)I-1000:
Jugador de X: t
Jugador de O: t
0	1	2
3	4	5
6	7	8
X tiró en la posición 5		
		X

Proyecto de Gato, con el output de terminal en la parte inferior, en el lado izquierdo el directorio de todos los proyectos que he realizado en Python con ayuda se VisualStudio Code, y en la parte superior el código fuente del proyecto.

NOMBRE COMPLETO DEL ALUMNO

Ain Bolaños Cortés