

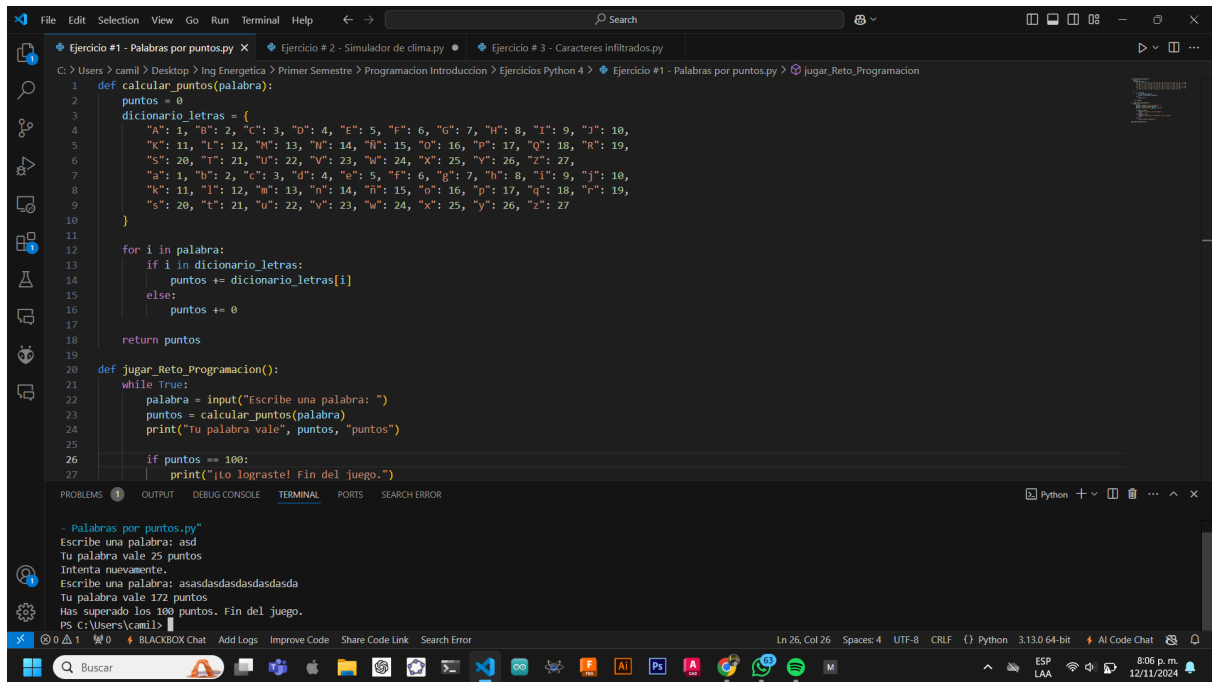
TALLER #4 PYTHON (3 EJERCICIOS)

CAMILO ALEXANDER FERNANDEZ VALVERDE
INTRODUCCIÓN A LA PROGRAMACIÓN

ING. CRISTHIAN ALEJANDRO CAÑAR MUÑOZ
PROFESOR DE ÁREA

CORPORACION UNIVERSITARIA AUTONOMA DEL CAUCA
INGENIERÍAS, ING ENERGÉTICA
POPAYÁN, CAUCA
2024

EJERCICIO # 1:



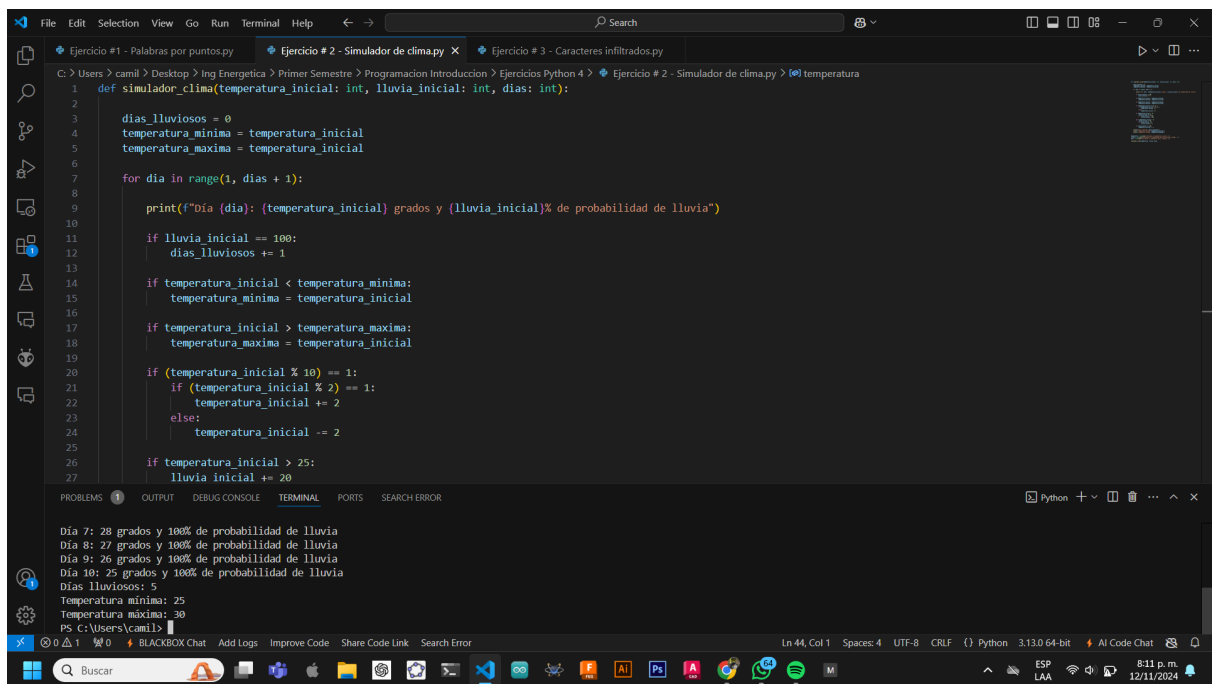
The screenshot shows a VS Code editor with a Python file named 'Ejercicio #1 - Palabras por puntos.py'. The script defines a function 'calcular_puntos(palabra)' that calculates the score of a word based on a predefined dictionary of letter values. It also defines a function 'jugar_Reto_Programacion()' that prompts the user to enter a word and prints the score. The terminal output shows the user entering 'asd' and 'asasdasdasdasdasda', with scores of 25 and 172 respectively. The user has also entered '100' to end the game.

```
1 def calcular_puntos(palabra):
2     puntos = 0
3     diccionario_letras = {
4         "A": 1, "B": 2, "C": 3, "D": 4, "E": 5, "F": 6, "G": 7, "H": 8, "I": 9, "J": 10,
5         "K": 11, "L": 12, "M": 13, "N": 14, "O": 15, "P": 16, "Q": 17, "R": 18,
6         "S": 20, "T": 21, "U": 22, "V": 23, "W": 24, "X": 25, "Y": 26, "Z": 27,
7         "a": 1, "b": 2, "c": 3, "d": 4, "e": 5, "f": 6, "g": 7, "h": 8, "i": 9, "j": 10,
8         "k": 11, "l": 12, "m": 13, "n": 14, "o": 15, "p": 16, "q": 17, "r": 18, "s": 20,
9         "t": 21, "u": 22, "v": 23, "w": 24, "x": 25, "y": 26, "z": 27
10    }
11
12    for i in palabra:
13        if i in diccionario_letras:
14            puntos += diccionario_letras[i]
15        else:
16            puntos += 0
17
18    return puntos
19
20 def jugar_Reto_Programacion():
21     while True:
22         palabra = input("Escribe una palabra: ")
23         puntos = calcular_puntos(palabra)
24         print("Tu palabra vale", puntos, "puntos")
25
26         if puntos == 100:
27             print("¡lo lograste! Fin del juego.")
```

Terminal Output:

```
- Palabras por puntos.py
Escribe una palabra: asd
Tu palabra vale 25 puntos
Intenta nuevamente.
Escribe una palabra: asasdasdasdasdasda
Tu palabra vale 172 puntos
Has superado los 100 puntos. Fin del juego.
PS C:\Users\camil>
```

EJERCICIO # 2:



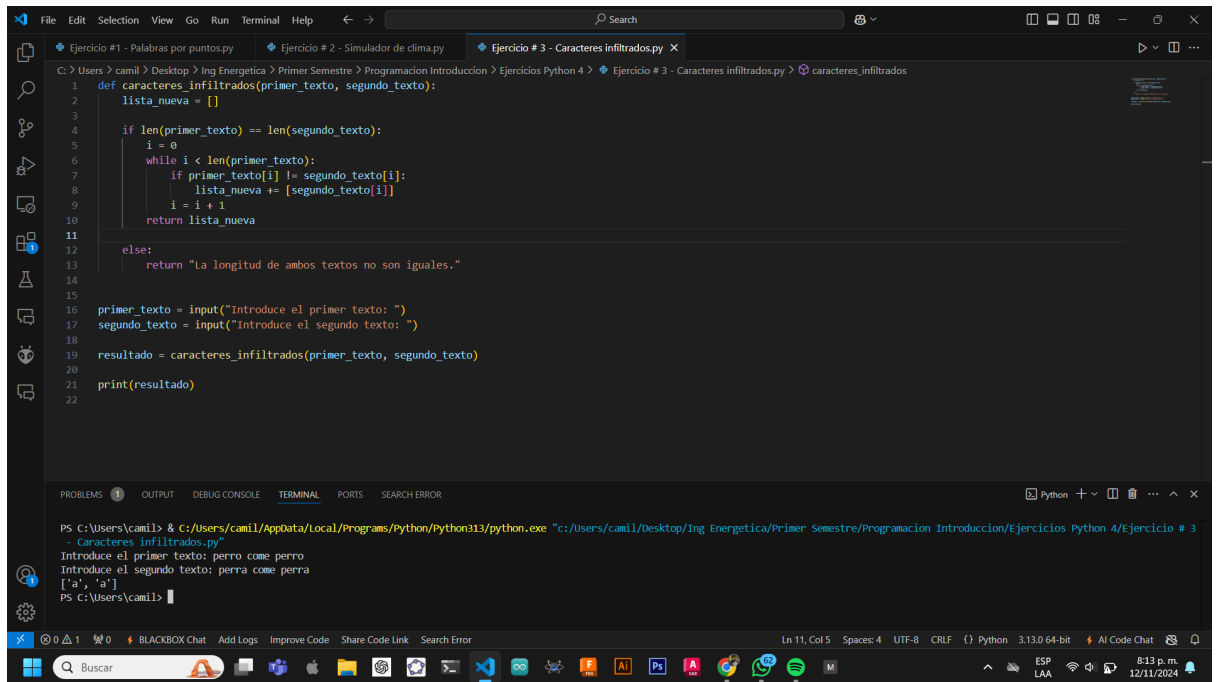
The screenshot shows a VS Code editor with a Python file named 'Ejercicio #2 - Simulador de clima.py'. The script defines a function 'simulador_clima(temperatura_inicial, lluvia_inicial, dias)' that simulates a climate over a given number of days. It updates the temperature and the number of rainy days based on certain conditions. The terminal output shows the simulation results for 10 days, including the number of rainy days and the temperature range.

```
1 def simulador_clima(temperatura_inicial: int, lluvia_inicial: int, dias: int):
2
3     dias_lluviosos = 0
4     temperatura_minima = temperatura_inicial
5     temperatura_maxima = temperatura_inicial
6
7     for dia in range(1, dias + 1):
8
9         print(f"Día {dia}: {temperatura_inicial} grados y {lluvia_inicial}% de probabilidad de lluvia")
10
11         if lluvia_inicial == 100:
12             dias_lluviosos += 1
13
14         if temperatura_inicial < temperatura_minima:
15             temperatura_minima = temperatura_inicial
16
17         if temperatura_inicial > temperatura_maxima:
18             temperatura_maxima = temperatura_inicial
19
20         if (temperatura_inicial % 10) == 1:
21             if (temperatura_inicial % 2) == 1:
22                 temperatura_inicial += 2
23             else:
24                 temperatura_inicial -= 2
25
26         if temperatura_inicial > 25:
27             lluvia_inicial += 20
```

Terminal Output:

```
Día 7: 28 grados y 100% de probabilidad de lluvia
Día 8: 27 grados y 100% de probabilidad de lluvia
Día 9: 26 grados y 100% de probabilidad de lluvia
Día 10: 25 grados y 100% de probabilidad de lluvia
Días lluviosos: 5
Temperatura mínima: 25
Temperatura máxima: 30
PS C:\Users\camil>
```

EJERCICIO # 3:



The image shows a Visual Studio Code editor window with a Python file named 'caracteres_infiltrados.py'. The script defines a function 'caracteres_infiltrados' that takes two strings as input and returns a list of characters that are present in both strings. The function uses a while loop to iterate through the characters of the first string and checks if they are in the second string. If they are, they are added to a new list. The function also handles the case where the strings are of different lengths by returning a message. The script prompts the user to enter two strings and prints the result.

```
1 def caracteres_infiltrados(primer_texto, segundo_texto):
2     lista_nueva = []
3
4     if len(primer_texto) == len(segundo_texto):
5         i = 0
6         while i < len(primer_texto):
7             if primer_texto[i] in segundo_texto[i]:
8                 lista_nueva += [segundo_texto[i]]
9                 i = i + 1
10            return lista_nueva
11
12     else:
13         return "La longitud de ambos textos no son iguales."
14
15
16 primer_texto = input("Introduce el primer texto: ")
17 segundo_texto = input("Introduce el segundo texto: ")
18
19 resultado = caracteres_infiltrados(primer_texto, segundo_texto)
20
21 print(resultado)
22
```

The terminal output shows the execution of the script. The user enters 'perro' as the first text and 'perro' as the second text. The output is ['a', 'a'], which appears to be a mistake in the terminal output or a typo in the input.

```
PS C:\Users\camil> & C:\Users\camil\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/camil/Desktop/Ing Energetica/Primer Semestre/Programacion Introduccion/Ejercicios Python 4/Ejercicio # 3 - Caracteres Infiltrados.py"
Introduce el primer texto: perro
Introduce el segundo texto: perro
['a', 'a']
PS C:\Users\camil>
```