

Metodología: Para el desarrollo de la parte de del TP separamos la consigna en 2 partes:

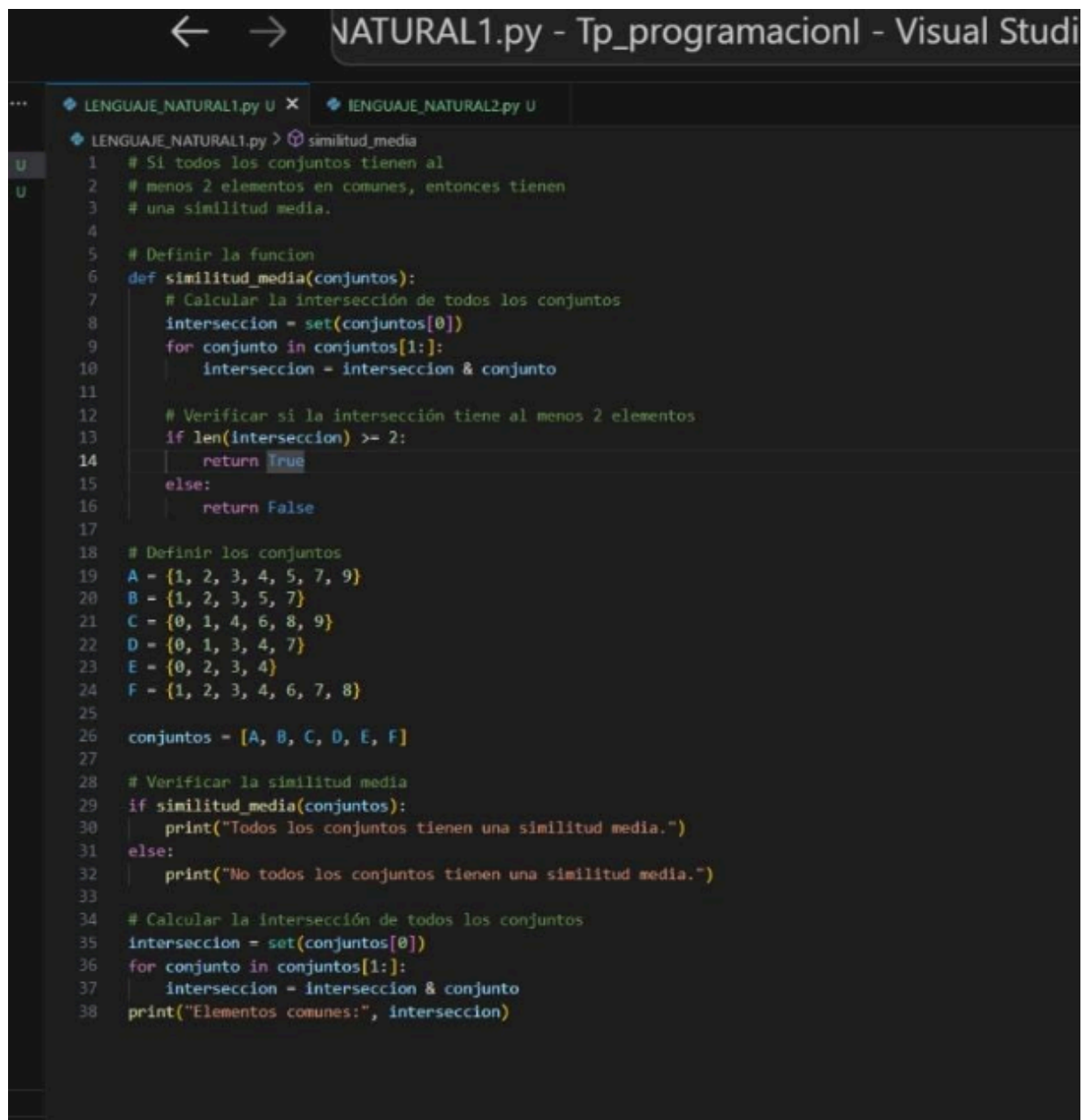
- Realización de los diagramas de venn de diferencia y diferencia simétrica
- Realización de lenguaje natural a lenguaje Python y desarrollo del programa.

PUNTO 5

En este punto se combinó el ejercicio 5 de expresiones matemáticas y una parte del ejercicio 1 de implementación en python.

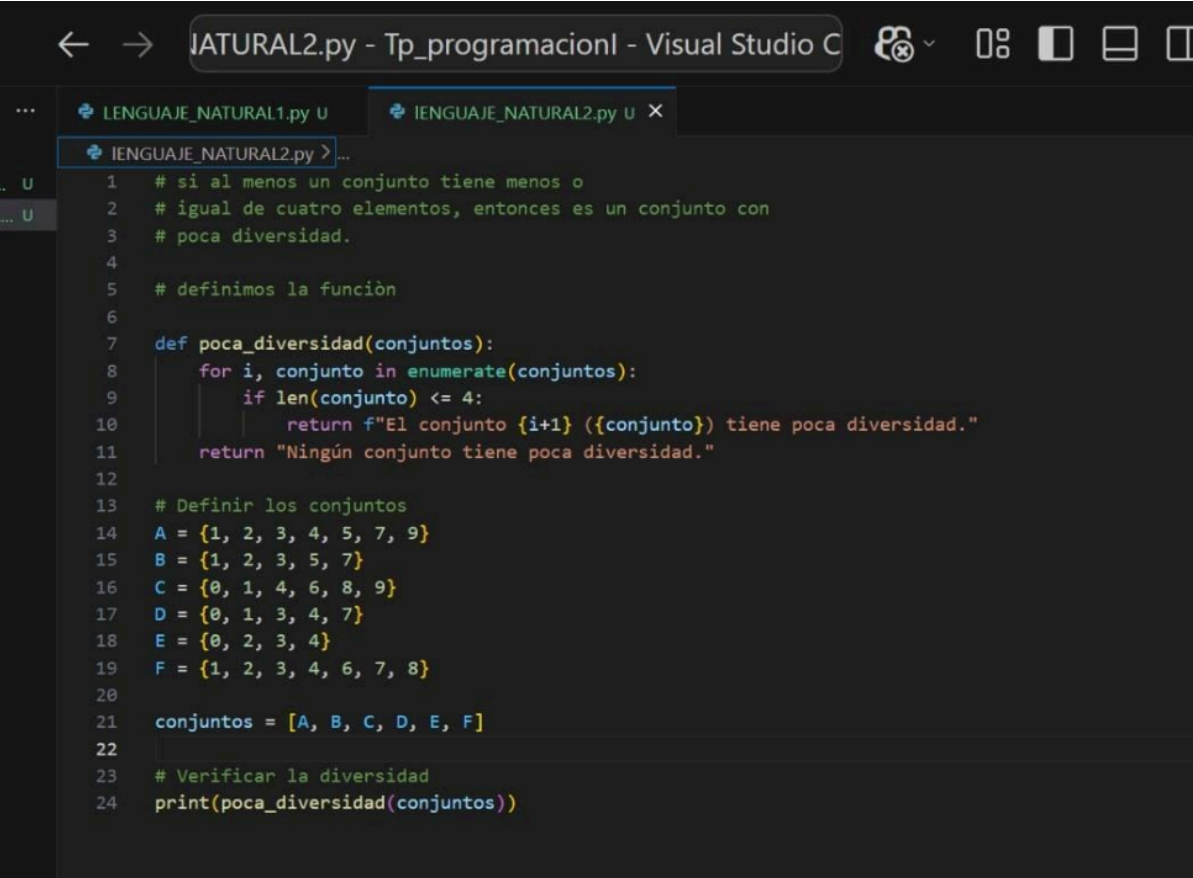
IMPLEMENTACIÓN EN PYTHON ejercicio 2.A.6: este ejercicio pertenece a la parte 2, implementación en python, pero fue movido a esta parte del trabajo así todos los integrantes del grupo realizamos operaciones con código.

A) Si todos los conjuntos tienen al menos 2 elementos en comunes, entonces tienen una similitud media.



```
← → NATURAL1.py - Tp_programacionI - Visual Studi
... LENGUAJE_NATURAL1.py U × LENGUAJE_NATURAL2.py U
LENGUAJE_NATURAL1.py > similitud_media
1 # Si todos los conjuntos tienen al
2 # menos 2 elementos en comunes, entonces tienen
3 # una similitud media.
4
5 # Definir la funcion
6 def similitud_media(conjuntos):
7     # Calcular la intersección de todos los conjuntos
8     interseccion = set(conjuntos[0])
9     for conjunto in conjuntos[1:]:
10         interseccion = interseccion & conjunto
11
12     # Verificar si la intersección tiene al menos 2 elementos
13     if len(interseccion) >= 2:
14         return True
15     else:
16         return False
17
18 # Definir los conjuntos
19 A = {1, 2, 3, 4, 5, 7, 9}
20 B = {1, 2, 3, 5, 7}
21 C = {0, 1, 4, 6, 8, 9}
22 D = {0, 1, 3, 4, 7}
23 E = {0, 2, 3, 4}
24 F = {1, 2, 3, 4, 6, 7, 8}
25
26 conjuntos = [A, B, C, D, E, F]
27
28 # Verificar la similitud media
29 if similitud_media(conjuntos):
30     print("Todos los conjuntos tienen una similitud media.")
31 else:
32     print("No todos los conjuntos tienen una similitud media.")
33
34 # Calcular la intersección de todos los conjuntos
35 interseccion = set(conjuntos[0])
36 for conjunto in conjuntos[1:]:
37     interseccion = interseccion & conjunto
38 print("Elementos comunes:", interseccion)
```

B) Si al menos un conjunto tiene menos o igual de cuatro elementos, entonces es un conjunto con poca diversidad.



```
1 # si al menos un conjunto tiene menos o
2 # igual de cuatro elementos, entonces es un conjunto con
3 # poca diversidad.
4
5 # definimos la función
6
7 def poca_diversidad(conjuntos):
8     for i, conjunto in enumerate(conjuntos):
9         if len(conjunto) <= 4:
10             return f"El conjunto {i+1} ({conjunto}) tiene poca diversidad."
11     return "Ningún conjunto tiene poca diversidad."
12
13 # Definir los conjuntos
14 A = {1, 2, 3, 4, 5, 7, 9}
15 B = {1, 2, 3, 5, 7}
16 C = {0, 1, 4, 6, 8, 9}
17 D = {0, 1, 3, 4, 7}
18 E = {0, 2, 3, 4}
19 F = {1, 2, 3, 4, 6, 7, 8}
20
21 conjuntos = [A, B, C, D, E, F]
22
23 # Verificar la diversidad
24 print(poca_diversidad(conjuntos))
```

