

# TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN – A DISTANCIA

Trabajo Integrador 2 – Matemática y Programación

COMISIÓN: M2025-4

---

## Datos Generales

**Título del trabajo:** Conjuntos, Lógica y Programación en Python

**Alumnos:** Ramiro Morales - Diego Raúl Montes

**Materia:** Matemática y Programación

**Profesor/a - Tutora:** Eduardo Monaco - Araceli Rojas

**Fecha de entrega:** 09/06/2025

---

## Índice

1. Desarrollo Matemático (**Conjuntos y Lógica**).
  2. Desarrollo en Python (**Captura del Algoritmo Python**).
  3. Metodología de Trabajo.
  4. Links del Video y Repositorio **Github**.
- 

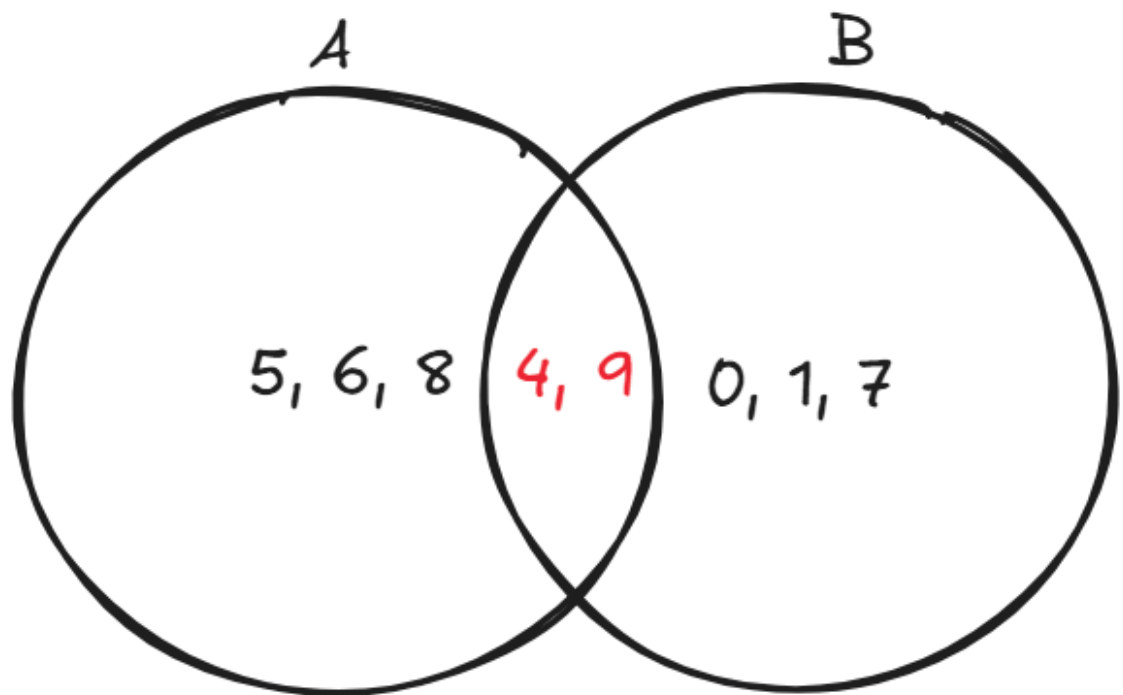
## 1. Desarrollo Matemático (Conjuntos y Lógica)

### 1.1. Conjuntos

- DNI: 46998595
- DNI: 44090717

- DNI **A**: {4, 5, 6, 8, 9}
- DNI **B**: {0, 1, 4, 7, 9}
- Operaciones realizadas entre conjuntos:
  - **Unión ( $A \cup B$ )**: todos los elementos de A y B sin repetir  $\rightarrow \{0, 1, 4, 5, 6, 7, 8, 9\}$
  - **Intersección ( $A \cap B$ )**: elementos que están en ambos  $\rightarrow \{4, 9\}$
  - **Diferencia ( $A - B$ )**: elementos de A que no están en B  $\rightarrow \{5, 6, 8\}$
  - **Diferencia ( $B - A$ )**: elementos de B que no están en A  $\rightarrow \{0, 1, 7\}$
  - **Diferencia simétrica ( $A \Delta B$ )**: elementos que están en A o en B pero no en ambos  $\rightarrow \{0, 1, 5, 6, 7, 8\}$

## 1.2. Diagramas de Venn



## 2. Desarrollo en Python (Captura del Algoritmo Python).

```
Integrador II.py X
Matematica > Integrador II.py > ...
You, hace 4 minutos | 2 authors (RamiroMoralesdev and one other)
1 print("\nEjercicio Integrador de Matemática")
2 while True:
3     a = input("\nIngrese un DNI: ")
4     b = input("Ingrese un segundo DNI: ")
5     if a.isdigit() and b.isdigit():
6         print("DNI válidos")
7         break
8     else:
9         print("ERROR PROBA DE NUEVO.")
10
11 def conjuntos(a, b):
12     a = list(sorted(set(a)))
13     b = list(sorted(set(b)))
14     print("\nConjunto A:", a)
15     print("Conjunto B:", b)
16
17     union = set(a) | set(b) # | = OR
18     # Convierto en set para eliminar elementos repetidos y sorted para ordenarlos
19     interseccion = list(sorted(set(a) & set(b)))
20     diferencia = list(sorted(set(a) - set(b)))
21     diferencia = list(sorted(set(b) - set(a)))
22     diferencia_simetrica = list(sorted(set(a) ^ set(b))) # ^ = XOR
23
24     # Imprimo los resultados
25     print(f"\nUnion:", union)
26     print(f"Interseccion:", interseccion)
27     print(f"Diferencia:", diferencia)
28     print(f"Diferencia Simetrica:", diferencia_simetrica)
29
30 conjuntos(a, b)
31
32 # Conteo de frecuencia de cada dígito en cada conjunto utilizando estructuras repetitivas. La frecuencia es la cantidad de veces que aparece un dígito en el conjunto.
33 print("\nConteo de frecuencia de cada dígito en cada conjunto")
34 for i in range(10):
35     frecuencia_a = a.count(str(i))
36     frecuencia_b = b.count(str(i))
37     print(f"Frecuencia del dígito {i} en el conjunto A: {frecuencia_a}")
38     print(f"Frecuencia del dígito {i} en el conjunto B: {frecuencia_b}")
```

```
Integrador II.py X
Matematica > Integrador II.py > ...
40 # Suma total de todos los dígitos en cada conjunto
41 def suma_recursiva_digitos(n):
42     if n == 0:
43         return 0
44     else:
45         return n % 10 + suma_recursiva_digitos(n // 10) # n % 10 = retorna el ultimo dígito, n // 10 = saca el resto
46
47 n = int(a)
48 print(f"\nLa suma de los dígitos de {n} es: {suma_recursiva_digitos(n)}")
49
50 n = int(b)
51 print(f"La suma de los dígitos de {n} es: {suma_recursiva_digitos(n)}")
52
53 # Operaciones con años de nacimiento
54 print("\nOperaciones con años de nacimiento")
55 a = int(input("Ingrese un año de nacimiento: "))
56 b = int(input("Ingrese un año de nacimiento: "))
57
58 print("\n")
59 def calcular_par_impar(a,b):
60
61     # Contar cuántos nacieron en años pares e impares utilizando estructuras repetitivas.
62     while a % 2 == 0 & b % 2 == 0: # Si a es par o b es par
63         print(f"Los años {a} y {b} son pares")
64         break
65     else:
66         print(f"Los años {a} y {b} son impares")
67
68     if a & b > 2000:
69         print("Grupo Z")
70
71     def es_bisiesto(a):
72         if a % 4 == 0 and (a % 100 != 0 or a % 400 == 0):
73             return True
74         else:
75             return False
76     # Verificar si el año es bisiesto
77     if es_bisiesto(a) == True or es_bisiesto(b) == True:
78         print("Tenemos un año especial")
79     else: print("Ninguno de los años es bisiesto")
80
81     calcular_par_impar(a,b)
```

```
Integrador II.py X
Matematica > Integrador II.py > ...

82
83  años = {a, b}
84
85  edades = set()
86  for anio in años:
87      edad = 2025 - anio
88      edades.add(edad)
89
90  print("\nProducto cartesiano entre años de nacimiento y edades:")
91  for anio in años:
92      for edad in edades:
93          print(f"({anio}, {edad})")
94
95
```

---

### 3. Metodología de Trabajo.

- Nos organizamos en dos grupos. Solo nos tomó cinco minutos conversar y ponernos de acuerdo sobre cómo íbamos a dividir el trabajo. Una vez definido esto, comenzamos a programar y a realizar el Diagrama de Venn.
- Herramientas utilizadas (Python, Venn diagrams, Excalidraw, ChatGPT(Casi nos arruina el trabajo jaja))
- Pasos seguidos: investigación, desarrollo matemático, codificación, prueba, grabación del video

---

### 4. Links del Video y Repositorio Github.

- <https://github.com/RamiroMoralesdev/codesUTN/blob/main/Matematica/Integrador%20II.py>
  - <https://youtu.be/WIKxTGZATd0>
-