

Trabajo Práctico Integrador II

Matemáticas y Programación

Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional

Alumnos – Comisión 12

Cornejo, Diego

Cuquejo, Mauro

Dantur, Daniel

Díaz de Quintana, Melisa

Docente Titular

Vanina Durrutty

Docente Tutor

María Teresa Brizzi

13 de junio de 2025

Índice

Introducción	3
Objetivos	4
Consignas	5
Desarrollo.....	8
Conclusión	27

Introducción

Los **conjuntos** y la **lógica** son fundamentales para estructurar el pensamiento deductivo y resolver problemas de manera sistemática. Los conjuntos permiten agrupar elementos bajo ciertas propiedades, mientras que la lógica proporciona las herramientas para razonar sobre la veracidad o falsedad de proposiciones. Juntos, forman la base de muchas áreas, donde su implementación mediante programación abre puertas a soluciones eficientes y escalables.

En este trabajo, se explora cómo conceptos abstractos de teoría de conjuntos y lógica matemática pueden implementarse en Python, haciendo uso de estructuras fundamentales de programación y permitiendo resolver problemas computacionales con precisión.

Objetivos

Profundizar la integración entre los contenidos de Matemática (conjuntos y lógicos) y Programación (estructuras condicionales, repetitivas y funciones), fortaleciendo también el trabajo en equipo, la comunicación clara y la responsabilidad individual en proyectos colaborativos.

Consignas

El trabajo debe hacerse en grupo y todos los integrantes deben pertenecer a la misma comisión.

La conformación de grupos tiene como objetivo fomentar la colaboración entre pares, una habilidad fundamental que todo programador debe desarrollar para integrarse eficazmente en proyectos de gran envergadura.

Cada integrante debe asumir responsabilidades específicas dentro del proyecto, explicar su parte en el video y entregar por escrito una descripción de las tareas que realizó.

Parte 1 – *Desarrollo Matemático (Conjuntos y Lógica)*

1. Cada integrante debe anotar su número de DNI.
2. A partir de los DNIs, se deben formar tantos conjuntos de dígitos únicos como integrantes tenga el grupo.
3. Realizar entre esos conjuntos las siguientes operaciones: unión, intersección, diferencia (entre pares) y diferencia simétrica.
4. Para cada una de estas operaciones, se debe realizar un diagrama de Venn (a mano o digital), que debe incluirse en la entrega.
5. Redactar al menos dos expresiones lógicas en lenguaje natural, que puedan luego implementarse en Python y escribir en la documentación que van a presentar cuál sería el resultado con los conjuntos que tienen.

Estas expresiones deben incluirse en el archivo PDF de la parte teórica y se espera que al menos una de ellas se implemente directamente como lógica en el programa Python.

Parte 2 – *Desarrollo del Programa en Python*

El programa debe implementar varias de las ideas trabajadas en papel. Debe incluir:

A. Operaciones con DNIs

- Ingreso de los DNIs (reales o ficticios).
- Generación automática de los conjuntos de dígitos únicos.
- Cálculo y visualización de: unión, intersección, diferencias y diferencia simétrica.
- Conteo de frecuencia de cada dígito en cada DNI utilizando estructuras repetitivas.
- Suma total de los dígitos de cada DNI.
- Evaluación de condiciones lógicas (condicionales), vinculadas con las expresiones escritas.

B. Operaciones con años de nacimiento

- Ingreso de los años de nacimiento (Si dos o más integrantes del grupo tienen el mismo año, ingresar algún dato ficticio, según el caso).
- Contar cuántos nacieron en años pares e impares utilizando estructuras repetitivas.
- Si todos nacieron después del 2000, mostrar "Grupo Z".
- Si alguno nació en año bisiesto, mostrar "Tenemos un año especial".
- Implementar una función para determinar si un año es bisiesto.

- Calcular el producto cartesiano entre el conjunto de años y el conjunto de edades actuales.

Parte 3 – Video de Presentación

Duración estimada entre 5 y 10 minutos. Todos los integrantes deben presentarse en cámara, mostrar el programa funcionando y explicar la parte que realizaron. También deben comentar brevemente qué aprendieron al combinar matemática y programación.

Entrega final

1. Archivo PDF con: desarrollo de conjuntos y operaciones, todos los diagramas de Venn, expresiones lógicas redactadas, y tareas de cada integrante explicadas por escrito.
2. Archivo con extensión .py que contenga el programa en Python.
3. Video grupal subido en lo posible a YouTube.
4. Documento adicional con los nombres de los integrantes, descripción de lo que hizo cada uno y la relación entre las expresiones lógicas escritas y el código implementado.

Calificación

La calificación será numérica y para aprobar deberá ser mayor o igual a **6**.

Desarrollo**Parte 1** – Desarrollo Matemático (Conjuntos y Lógica)

1. Cada integrante debe anotar su número de DNI.

Número de D.N.I. de cada integrante:

- Melisa Díaz de Quintana D.N.I. **35.357.229**
- Daniel Dantur D.N.I. **31.081.403**
- Diego Cornejo D.N.I. **45.024.397**
- Mauro Cuquejo D.N.I. **35.539.349**

2. A partir de los DNIs, se deben formar tantos conjuntos de dígitos únicos como integrantes tenga el grupo.

Conjuntos de dígitos únicos:

$$A = \{2, 3, 5, 7, 9\}$$

$$B = \{0, 1, 3, 4, 8\}$$

$$C = \{0, 2, 3, 4, 5, 7, 9\}$$

$$D = \{3, 4, 5, 9\}$$

3. Realizar entre esos conjuntos las siguientes operaciones: unión, intersección, diferencia (entre pares) y diferencia simétrica.

Unión conjuntos:

La unión de dos conjuntos es el conjunto formado por los elementos que pertenecen al menos a uno de ellos o a los dos. Omitimos las viceversas de las combinaciones, ya que el resultado es el mismo: $A \cup B = B \cup A$.

$A \cup B = \{0, 1, 2, 3, 4, 5, 7, 8, 9\}$	$A \cup C = \{0, 2, 3, 4, 5, 7, 9\}$
$A \cup D = \{2, 3, 4, 5, 7, 9\}$	$B \cup C = \{0, 1, 2, 3, 4, 5, 7, 8, 9\}$
$B \cup D = \{0, 1, 3, 4, 5, 8, 9\}$	$C \cup D = \{0, 2, 3, 4, 5, 7, 9\}$

Intersección conjuntos

La intersección de conjuntos son los elementos comunes a los dos conjuntos. Omitimos las viceversas de las combinaciones, ya que el resultado es el mismo: $A \cap B = B \cap A$.

$A \cap B = \{3\}$	$A \cap C = \{2, 3, 5, 7, 9\}$
$A \cap D = \{3, 5, 9\}$	$B \cap C = \{0, 3, 4\}$
$B \cap D = \{3, 4\}$	$C \cap D = \{3, 4, 5, 9\}$

Diferencia conjuntos

La diferencia de conjuntos dan como resultado los elementos del primer conjunto que no se encuentran en el segundo conjunto. En éste caso la inversión de la expresión si tiene un resultado distinto: $A - B \neq B - A$.

$A - B = \{2, 5, 7, 9\}$	$A - C = \{\}$
$A - D = \{2, 7\}$	$B - A = \{0, 1, 4, 8\}$
$B - C = \{1, 8\}$	$B - D = \{0, 1, 8\}$
$C - A = \{0, 4\}$	$C - B = \{2, 5, 7, 9\}$
$C - D = \{0, 2, 7\}$	$D - A = \{4\}$
$D - B = \{5, 9\}$	$D - C = \{\}$

Diferencia Simétrica

La diferencia simétrica de dos conjuntos es el conjunto formado por los elementos que pertenecen a uno o a otro conjunto, pero no a ambos a la vez. Omitimos las viceversas de las combinaciones, ya que el resultado es el mismo: $A \triangle B = B \triangle A$.

$A \triangle B = \{0, 1, 2, 4, 5, 7, 8, 9\}$	$A \triangle C = \{0, 4\}$
--	----------------------------

$A \triangle D = \{2, 4, 7\}$	$B \triangle C = \{1, 2, 5, 7, 8, 9\}$
$B \triangle D = \{0, 1, 5, 8, 9\}$	$C \triangle D = \{0, 2, 7\}$

4. Para cada una de estas operaciones, se debe realizar un diagrama de Venn (a mano o digital), que debe incluirse en la entrega.

Uniones

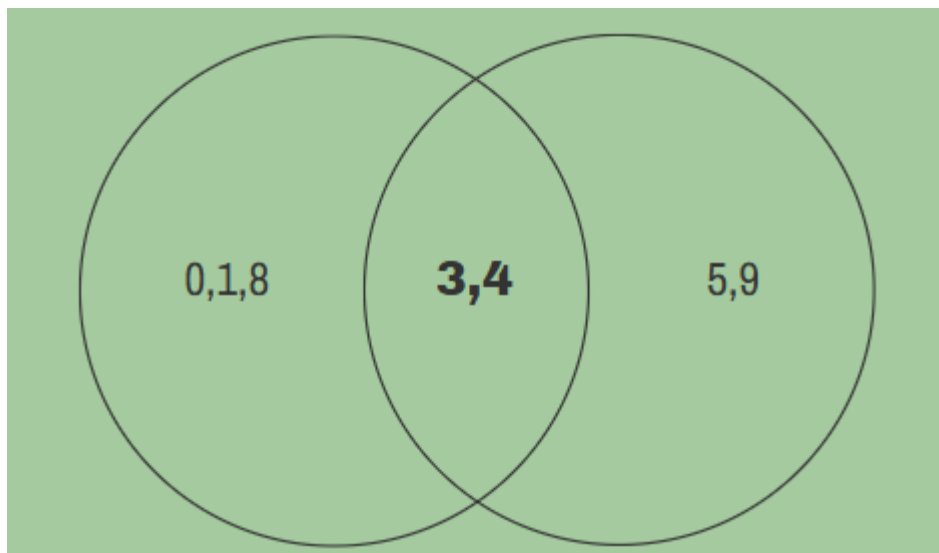
$$A \cup B = \{0, 1, 2, 3, 4, 5, 7, 8, 9\}$$



$$A \cup D = \{2, 3, 4, 5, 7, 9\}$$



$$B \cup D = \{0, 1, 3, 4, 5, 8, 9\}$$



$$A \cup C = \{0, 2, 3, 4, 5, 7, 9\}$$



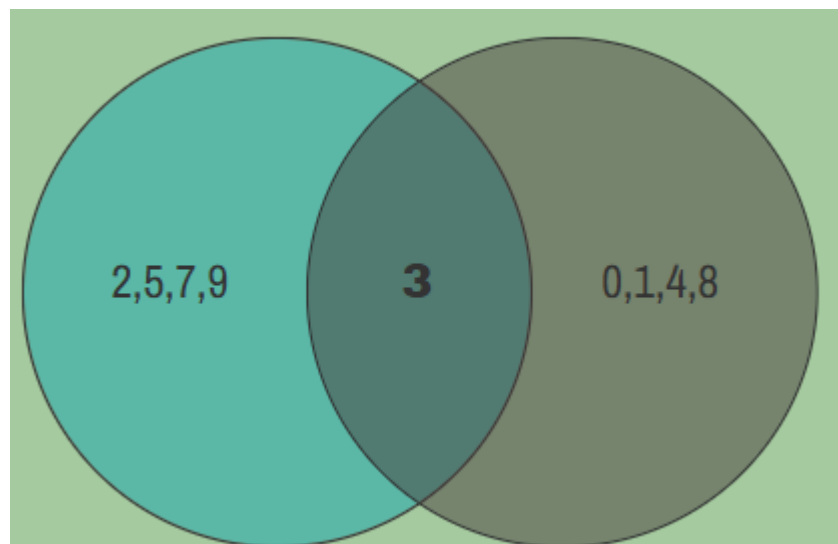
$$B \cup C = \{0, 1, 2, 3, 4, 5, 7, 8, 9\}$$



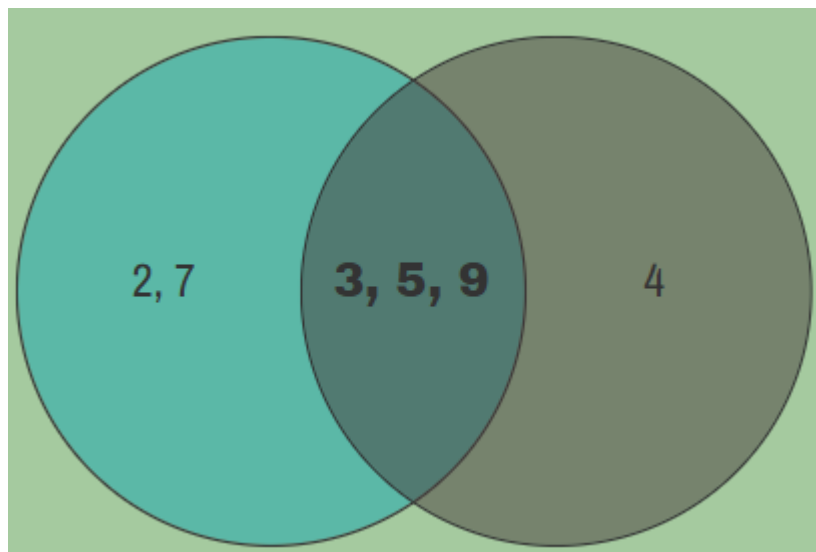
$$C \cup D = \{0, 2, 3, 4, 5, 7, 9\}$$

***Intersecciones***

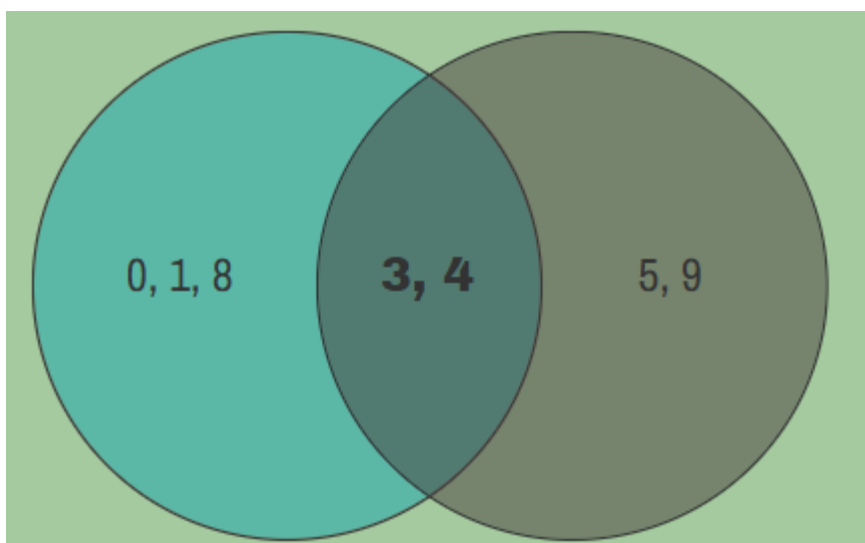
$$A \cap B = \{3\}$$



$$A \cap D = \{3, 5, 9\}$$



$$B \cap D = \{3, 4\}$$



$$C \cap D = \{3, 4, 5, 9\}$$



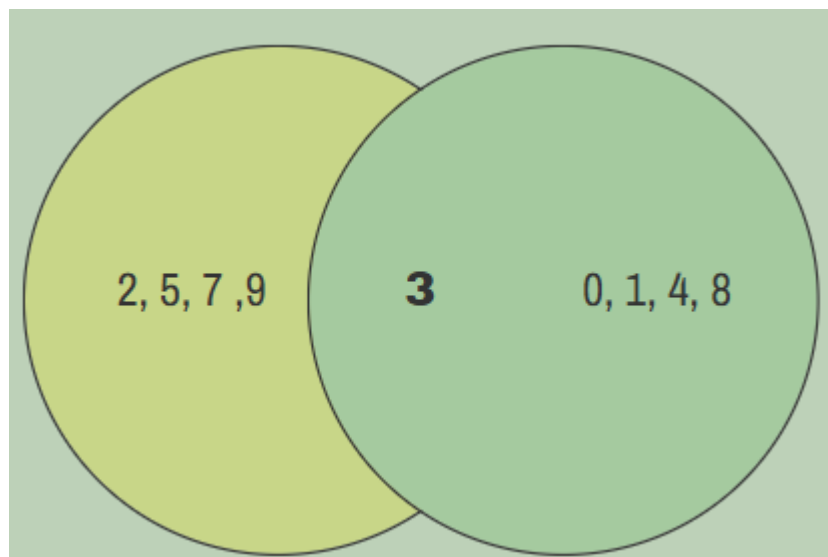
$$A \cap C = \{2, 3, 5, 7, 9\}$$



$$B \cap C = \{0, 3, 4\}$$

***Diferencias***

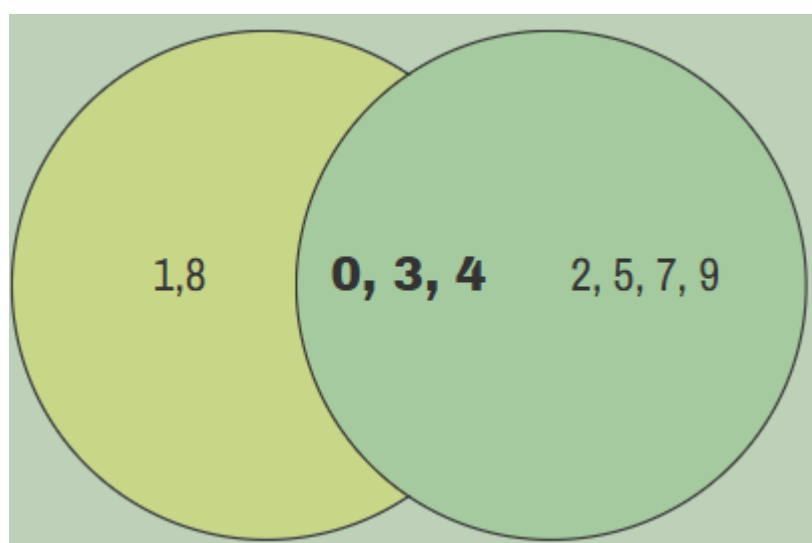
$$A - B = \{2, 5, 7, 9\}$$



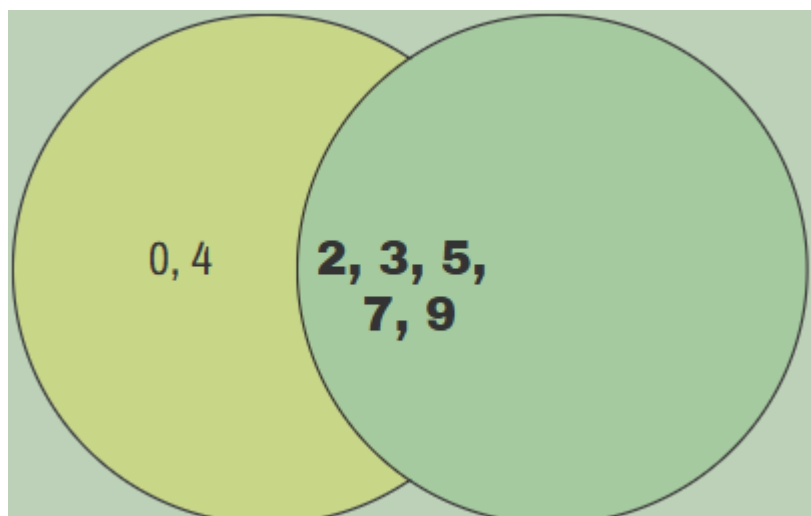
$$A - D = \{2, 7\}$$



$$B - C = \{1, 8\}$$



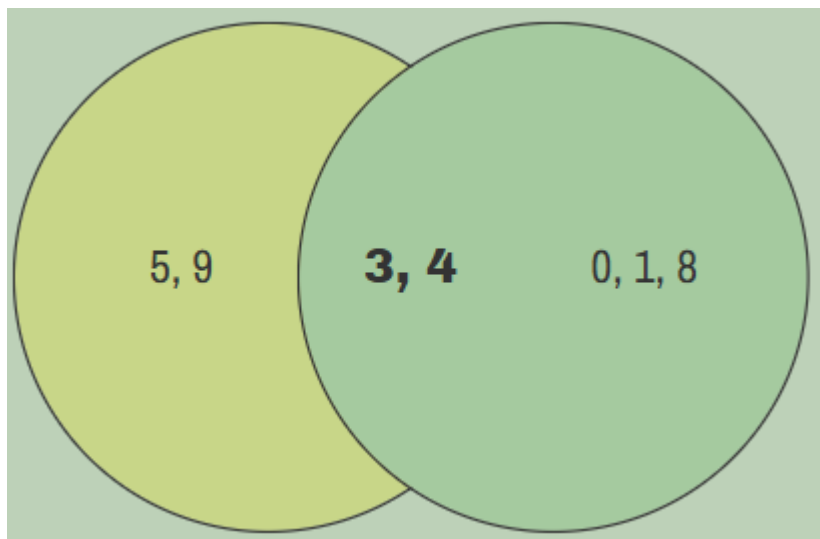
$$C - A = \{0, 4\}$$



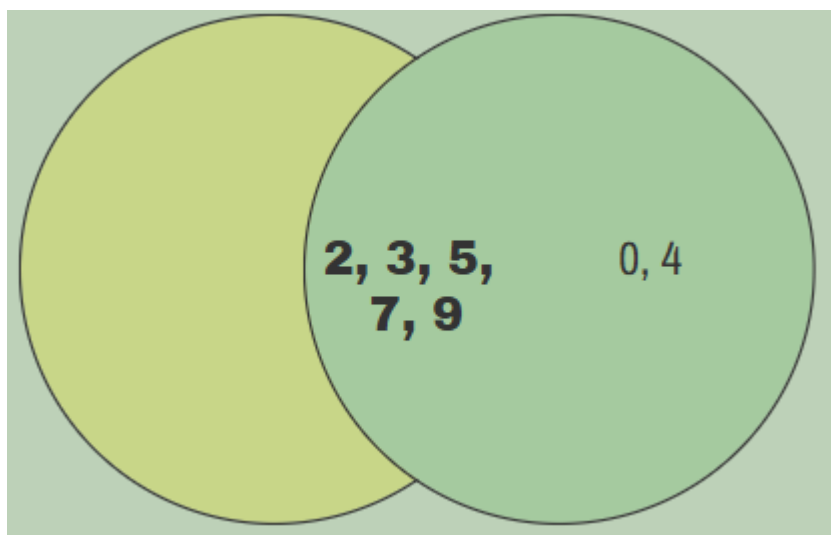
$$C - D = \{0, 2, 7\}$$



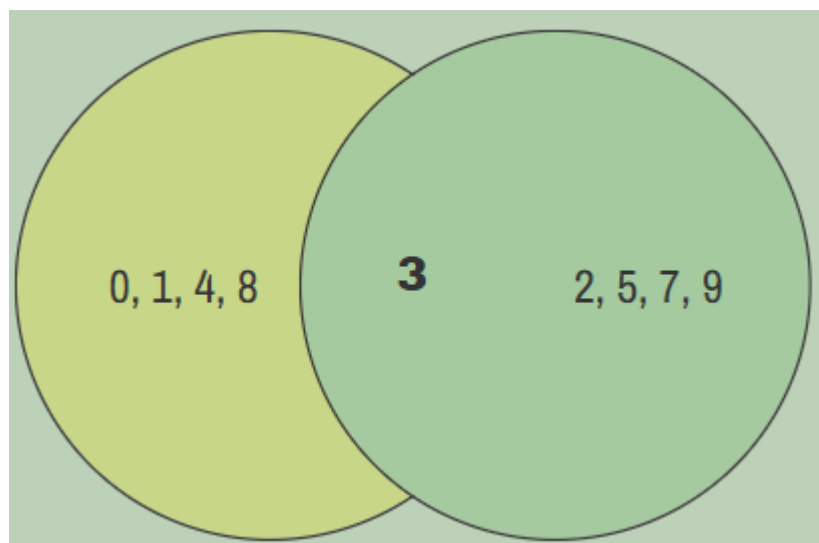
$$D - B = \{5, 9\}$$



$$A - C = \{ \}$$



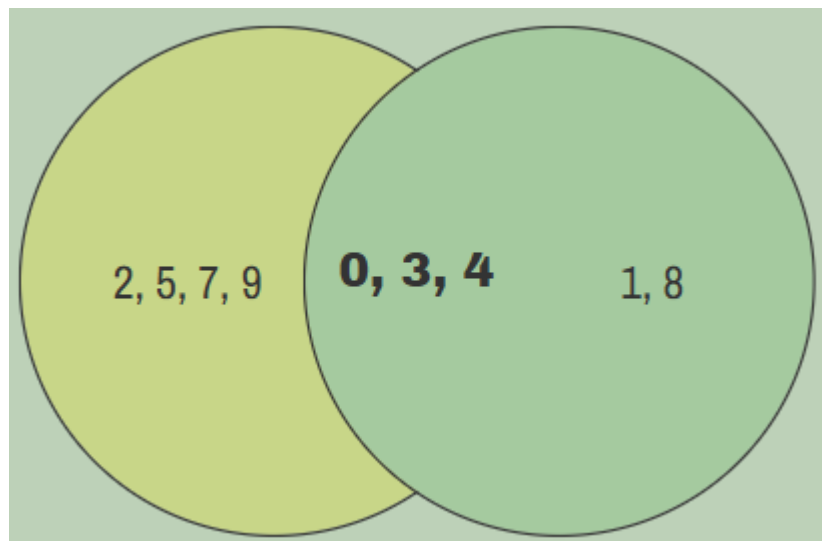
$$B - A = \{0, 1, 4, 8\}$$



$$B - D = \{0, 1, 8\}$$



$$C - B = \{2, 5, 7, 9\}$$



$$D - A = \{4\}$$

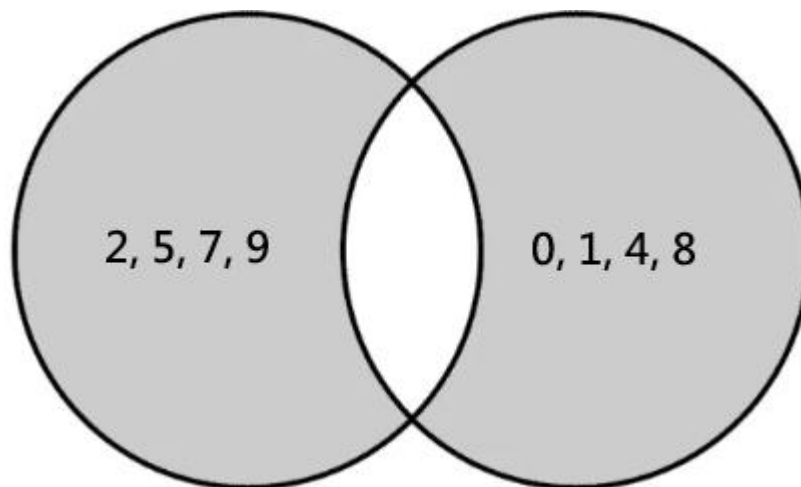


$$D - C = \{\}$$

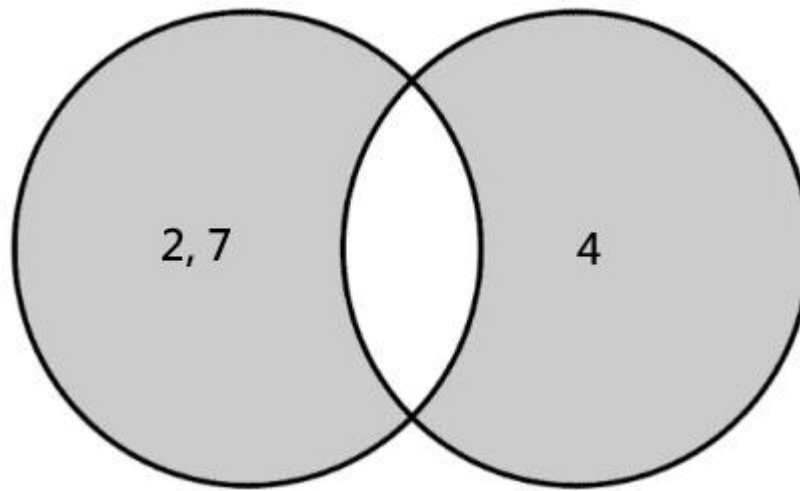


Diferencias simétricas

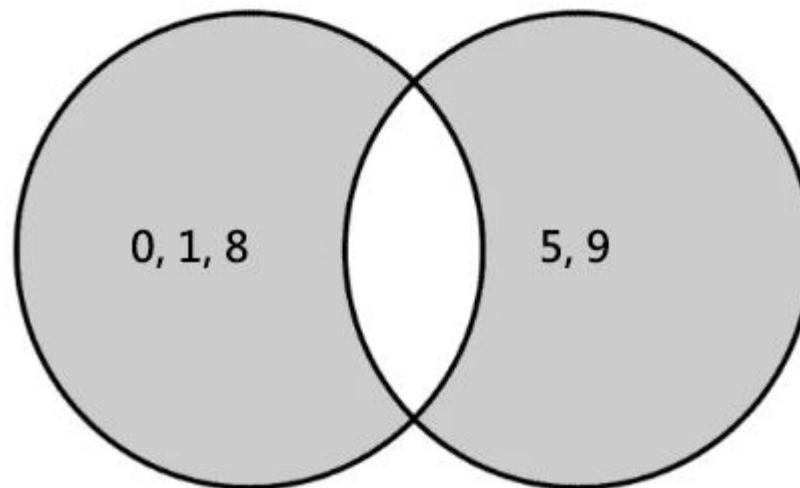
$$A \triangle B = \{0, 1, 2, 4, 5, 7, 8, 9\}$$



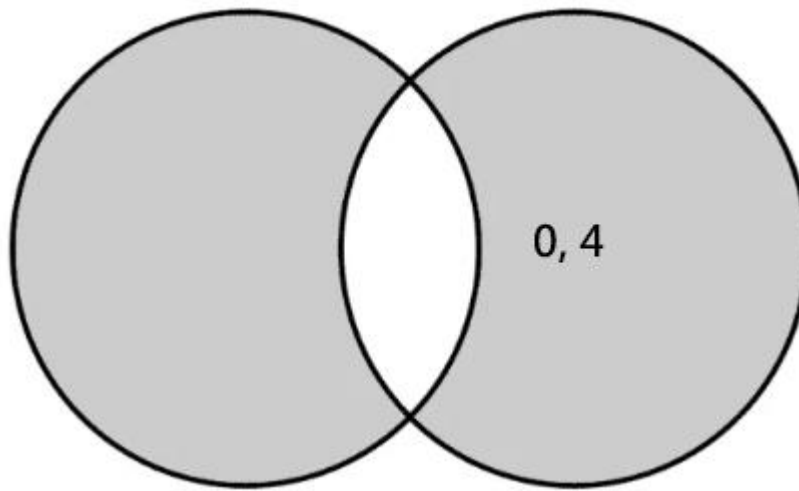
$$A \triangle D = \{2, 4, 7\}$$



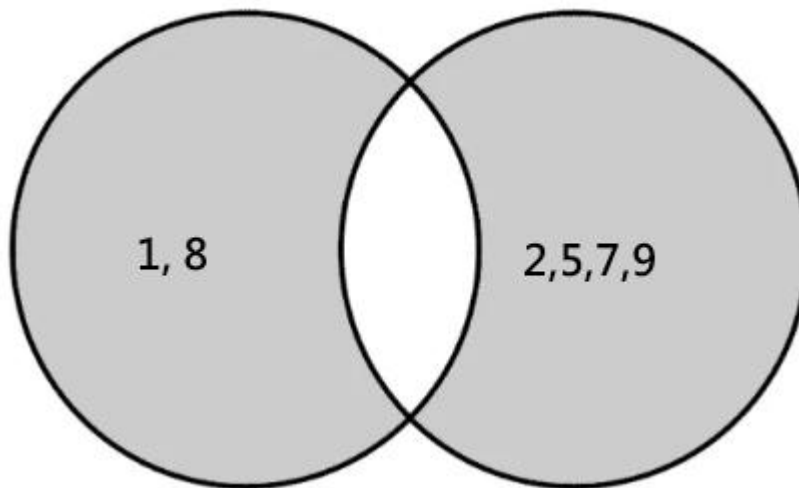
$$B \triangle D = \{0, 1, 5, 8, 9\}$$



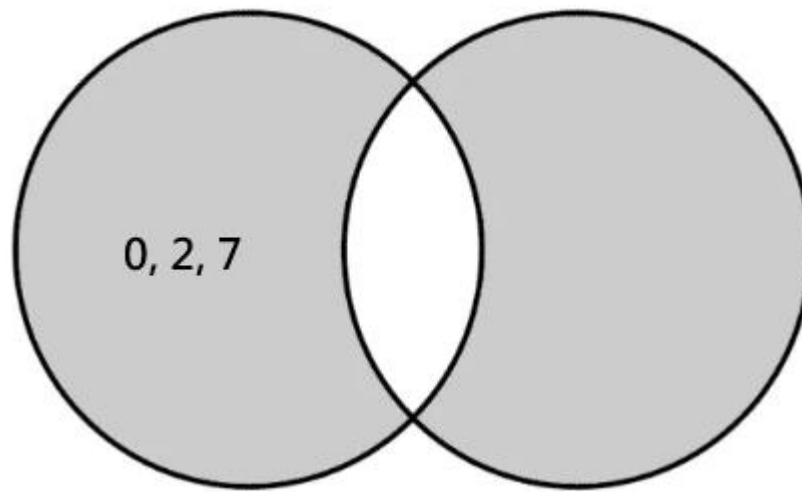
$$A \triangle C = \{0, 4\}$$



$$B \triangle C = \{1, 2, 5, 7, 8, 9\}$$



$$C \triangle D = \{0, 2, 7\}$$



5. Redactar al menos dos expresiones lógicas en lenguaje natural, que puedan luego implementarse en Python y escribir en la documentación que van a presentar cuál sería el resultado con los conjuntos que tienen.

Expresiones lógicas en lenguaje natural:

1. “Si todos los elementos de un conjunto pertenecen a otro conjunto que no es igual, entonces el primer conjunto es un subconjunto propio del segundo”

Implementación en Python:

La validación consiste en que la intersección de dos conjuntos es equivalente al segundo conjunto.

$$\mathbf{C = \{0,2,3,4,5,7,9\} \qquad D = \{3,4,5,9\}}$$

$$\mathbf{C \cap D = \{3,4,5,9\} \text{ es igual al conjunto D.}}$$

Otra forma de validarlo, es utilizando la diferencia de conjuntos. Si al conjunto D le resto el conjunto C da como resultado un conjunto vacío.

$$\mathbf{D - C = \{\}}$$

2. “Si un conjunto contiene más elementos que todos los demás, entonces se lo considera conjunto dominante”

Implementación en Python:

Comparamos la cantidad de elementos del conjunto C de forma iterativa contra el resto de los conjuntos, y verificamos que se cumple que el conjunto C tiene mayor cantidad de elementos que los conjuntos A, B y D.

A = {2,3,5,7,9} tiene *cinco* elementos **B** = {0,1,3,4,8} tiene *cinco* elementos

C = {0,2,3,4,5,7,9} tiene *siete* elementos **D** = {3,4,5,9} tiene *cuatro* elementos

El conjunto C es el conjunto dominante.

Conclusión

El desarrollo de este proyecto representó una instancia integradora donde se combinaron contenidos de matemática y programación para resolver un problema a partir de datos reales del grupo. A partir del uso de los DNIs y años de nacimiento de los integrantes, se lograron construir y analizar conjuntos, aplicar operaciones entre ellos, evaluar condiciones lógicas y realizar cálculos relevantes como frecuencias, sumas, clasificaciones y productos cartesianos.

El trabajo nos permitió representar información numérica mediante conjuntos de dígitos únicos, y luego aplicar operaciones clásicas como la unión, intersección, diferencia y diferencia simétrica, profundizando en su interpretación dentro del contexto del grupo. Asimismo, se incorporaron estructuras condicionales para verificar propiedades como la existencia de subconjuntos propios o la condición de conjunto dominante, lo cual reforzó la comprensión del razonamiento lógico y la formulación de expresiones en lenguaje natural llevadas al código.

Por otro lado, el análisis de los años de nacimiento complementó la mirada matemática: se clasificaron como pares o impares, se evaluó si todos los miembros pertenecen a la generación Z, se identificaron años bisiestos y se construyó el producto cartesiano con las edades, generando así un cruce de datos que ejemplifica la potencia del pensamiento combinatorio y la lógica matemática aplicada a situaciones concretas.

Además del contenido técnico, el proyecto promovió la organización del trabajo en equipo, la división de tareas, el testeo conjunto del programa y la puesta en común de conclusiones. Nos permitió tomar decisiones sobre cómo estructurar los datos, qué funciones desarrollar, y cómo interpretar los resultados dentro de un marco teórico. También fue una oportunidad para ejercitar habilidades de programación como el uso de listas, conjuntos, diccionarios, funciones auxiliares y estructuras de control, integrando la teoría aprendida con la práctica.