



TRABAJO INTEGRADOR 2

Matemática

Título del proyecto: Conjuntos

Alumnos: Pablo Molinari y Leonel Mercorelli

Materia: Matemática

Profesor: Ernesto Klimovsky

Tutor: Sergio Adrián Maldonado

Comisión: 17

Fecha: 13/06/2025

Contenido

Introducción.....	3
Objetivo.....	3
PARTE 1 – Desarrollo Matemático (Conjuntos y Lógica)	4
1) DNI de cada integrante.....	4
2) Conjuntos de dígitos únicos	4
3) Operaciones entre los conjuntos.....	5
4) Diagramas de Venn	7
5) Expresiones lógicas redactadas en lenguaje natural	8
PARTE 2 – Desarrollo del Programa en Python	10
A. Operaciones con DNI's	10
B. Operaciones con años de nacimiento	10
PARTE 3 – Video de Presentación.....	11
Conclusiones	11
ANEXOS	12
Anexo 1: Video de presentación.....	13
Anexo 2: Repositorio del trabajo	13
Anexo 3: Fuentes	14

Introducción

El presente trabajo integrador busca articular los contenidos de teoría de conjuntos y lógica matemática con su implementación concreta mediante la programación en Python. El objetivo es explorar cómo los conceptos abstractos de la matemática pueden representarse y resolverse a través de algoritmos, estructuras condicionales, bucles y funciones.

A partir del análisis de los DNIs y años de nacimiento de los integrantes del grupo, se construyeron conjuntos numéricos, se aplicaron operaciones clásicas de la teoría de conjuntos y se evaluaron distintas condiciones lógicas. Todo este proceso fue traducido a código Python, permitiendo una visualización clara, dinámica y automatizada de los resultados.

Objetivo

El propósito principal es profundizar la integración entre los contenidos de Matemática (conjuntos y lógica) y Programación (estructuras condicionales, repetitivas y funciones), fortaleciendo el trabajo colaborativo, la comunicación efectiva y la responsabilidad individual en proyectos grupales.

PARTE 1 – Desarrollo Matemático (Conjuntos y Lógica)

Para este trabajo se parte de dos números de DNI, utilizados como fuente de datos para aplicar operaciones matemáticas.

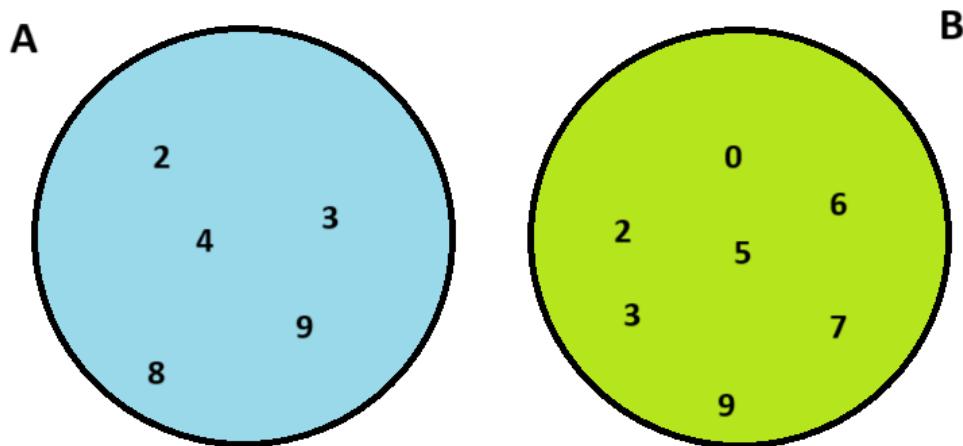
1) DNI de cada integrante

- Leonel: 42934838
- Pablo: 27099635

2) Conjuntos de dígitos únicos

Se extraen los dígitos únicos de cada DNI, formando así dos conjuntos.

- Conjunto A (Leonel): {2, 3, 4, 8, 9}
(Dígitos únicos del 42934838, sin repetir elementos)
- Conjunto B (Pablo): {0, 2, 3, 5, 6, 7, 9}
(Dígitos únicos del 27099635, sin repetir elementos)

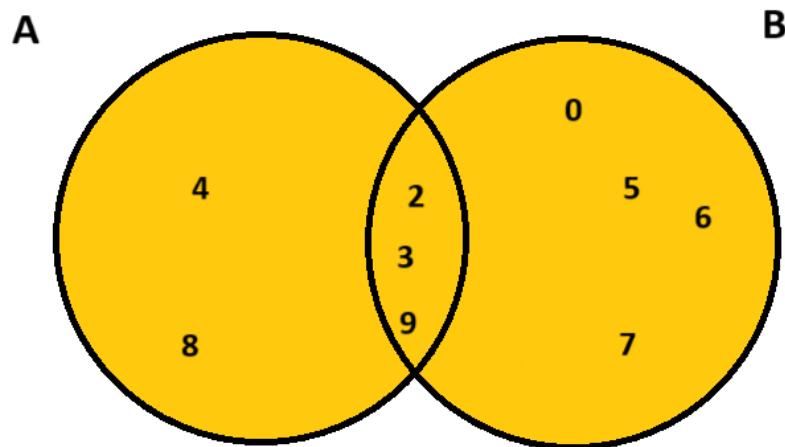


A partir de estos, se aplican operaciones básicas de teoría de conjuntos:

3) Operaciones entre los conjuntos

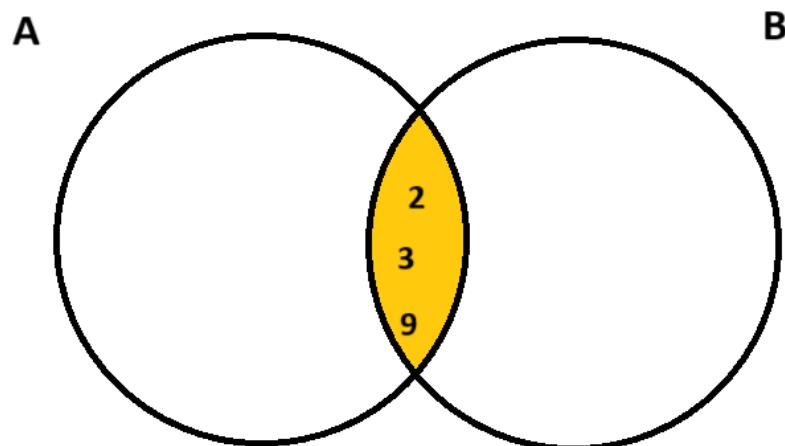
- a. **Unión ($A \cup B$):** $\{0, 2, 3, 4, 5, 6, 7, 8, 9\}$

La unión recoge todos los elementos de ambos conjuntos, sin repetir los que son comunes



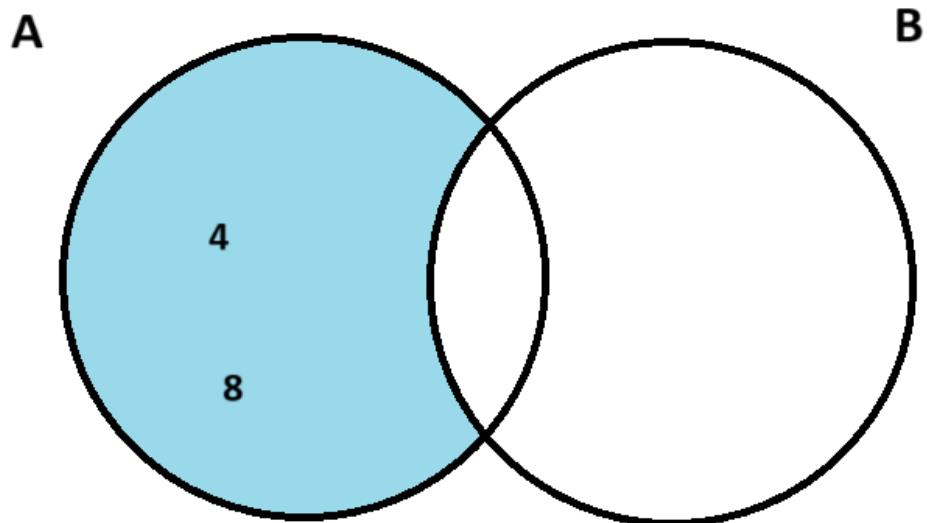
- **Intersección ($A \cap B$):** $\{2, 3, 9\}$

La intersección solo incluye los elementos que están en ambos conjuntos



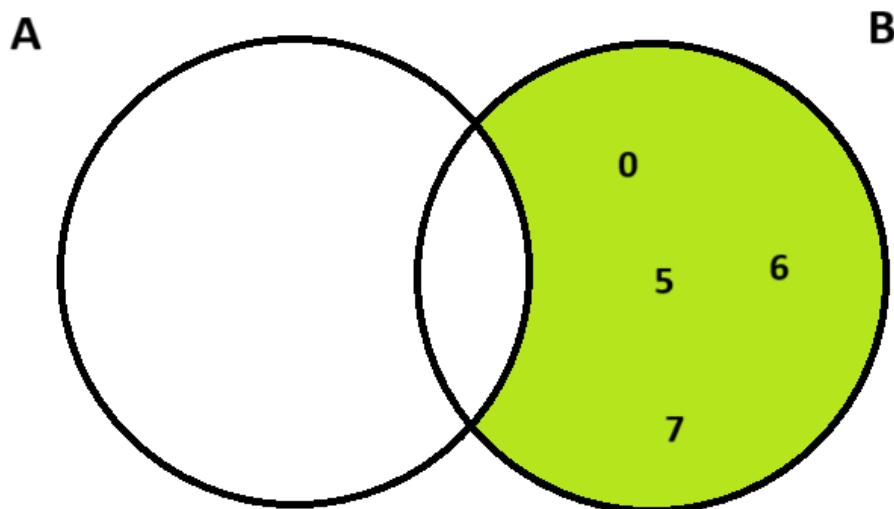
- Diferencia ($A - B$): $\{4, 8\}$

La diferencia ($A - B$) contiene los elementos que están en A, pero no en B.



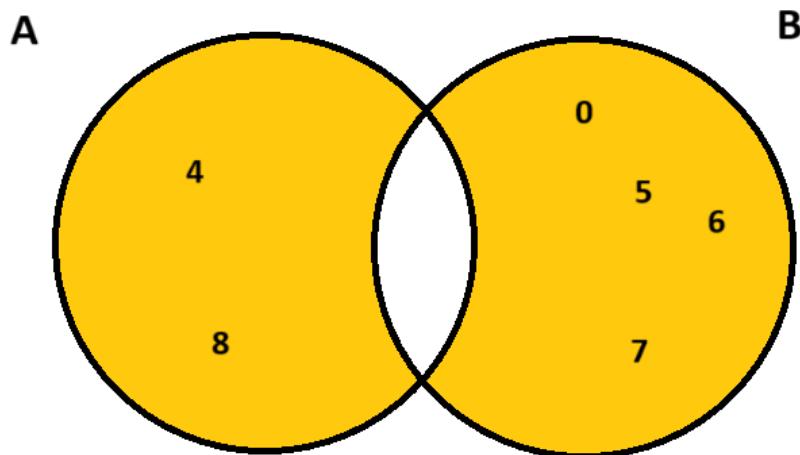
- b. Diferencia ($B - A$): $\{0, 5, 6, 7\}$

La diferencia ($B - A$) contiene los elementos que están en B, pero no en A.



- **Diferencia simétrica ($A \Delta B$): $\{0, 4, 5, 6, 7, 8\}$**

La diferencia simétrica contiene los elementos que están en A o en B, pero no en la intersección de ambos.



4) Diagramas de Venn

Los diagramas de Venn fueron elaborados y presentados a lo largo del desarrollo de los puntos 1, 2 y 3, acompañando cada una de las operaciones entre conjuntos para facilitar su comprensión visual.

5) Expresiones lógicas redactadas en lenguaje natural

• Expresión 1:

Enunciado:

Si algún dígito aparece en todos los conjuntos, se lo marca como “dígito común”.

(Esta condición permite detectar valores compartidos entre todos los conjuntos formados por los integrantes del grupo.)

Resultado esperado:

Los dígitos 2, 3 y 9 aparecen en ambos conjuntos.

Resultado obtenido:

Los dígitos comunes entre A y B son: {2, 3, 9}

• Expresión 2:

Enunciado:

Si hay más conjuntos con cantidad par de elementos que con cantidad impar, se etiqueta al grupo como “grupo par”.

(Evalúa la paridad de la cantidad de elementos en cada conjunto y clasifica el grupo en función de la mayoría.)

Resultado esperado:

Conjunto A: tiene 5 elementos → Tiene 3 elementos en total “par” → Grupo “par”

Conjunto B: tiene 7 elementos → Tiene 3 elementos “par” → Grupo “impar”

→ Ambos tienen cantidad impar → no se cumple la condición → grupo impar

Resultado obtenido:

Ambos conjuntos tienen cantidad impar de elementos.

→ El grupo se clasifica como: grupo impar.

♦ **Expresión 3:**

Enunciado:

Si dos conjuntos tienen exactamente la misma cantidad de elementos, se consideran “conjuntos equilibrados”.

Y si, además, la suma total de sus dígitos es igual, se los considera “conjuntos equivalentes”.

(Evalúa tanto la cantidad de elementos como el valor total de sus dígitos.)

Resultado esperado:

Conjunto A → 5 elementos, suma: $2 + 3 + 4 + 8 + 9 = 26$

Conjunto B → 7 elementos, suma: $0 + 2 + 3 + 5 + 6 + 7 + 9 = 32$

→ No cumplen ninguna de las condiciones.

Resultado obtenido:

No tienen la misma cantidad de elementos ni suman lo mismo.

→ Se los clasifica como: conjuntos dispares.

PARTE 2 – Desarrollo del Programa en Python

El programa fue estructurado en base a funciones reutilizables, que permiten modularizar las tareas como detección de bisiesto, suma de dígitos y comparación de conjuntos.

A continuación detallaremos las características principales de las 2 partes del programa:

A. Operaciones con DNI's

- Ingreso de los DNI's (reales o ficticios).
- Generación automática de los conjuntos de dígitos únicos correspondientes.
- Cálculo y visualización de la unión, intersección, diferencias y diferencia simétrica entre los conjuntos.
- Conteo de frecuencia de cada dígito en cada DNI, mediante estructuras repetitivas.
- Cálculo de la suma total de los dígitos de cada DNI.
- Evaluación de condiciones lógicas programadas en base a expresiones formuladas en lenguaje natural, tales como

B. Operaciones con años de nacimiento

- Ingreso de los años de nacimiento de los integrantes.
- Cálculo de cuántos nacieron en años pares e impares, utilizando estructuras repetitivas.
- Evaluación de condiciones especiales, tales como:
 - “Grupo Z” si todos nacieron después del año 2000.
 - “Tenemos un año especial” si al menos uno nació en un año bisiesto.
- Implementación de una función específica para determinar si un año es bisiesto.
- Cálculo del producto cartesiano entre el conjunto de años y el conjunto de edades actuales.

Estas funcionalidades permiten analizar la información de manera eficiente, conectando conceptos matemáticos con lógica computacional.

PARTE 3 – Video de Presentación (Anexo I)

Se realizó un video grupal de presentación y explicación del trabajo, cumpliendo con todas las pautas establecidas. La duración final fue de aproximadamente 10 minutos.

En el video, cada integrante se presentó en cámara, explicó su aporte al proyecto y mostró en funcionamiento el programa desarrollado en Python. Además, se comentaron brevemente los aprendizajes obtenidos al integrar contenidos de matemática y programación, destacando la utilidad de combinar el razonamiento lógico con la resolución algorítmica.

Este video no solo permitió mostrar el resultado final, sino también reflejar el proceso de trabajo colaborativo y la aplicación concreta de los conceptos teóricos.

Trabajo en grupo

Ambos integrantes participaron activamente en todas las etapas del trabajo, tanto en el desarrollo teórico como en la implementación práctica. Sin embargo, se asignaron lineamientos generales para organizar mejor las tareas.

Leonel Mercorelli:

Se enfoco principalmente en la parte teórica del trabajo. Analizo los DNIs, genero los conjuntos de dígitos únicos y realizo las operaciones entre conjuntos como la unión, intersección y diferencias. Redacto las expresiones lógicas que luego se implementaron en el programa. También elaboro los diagramas de Venn y participo en la redacción del informe escrito.

Pablo Molinari:

Estuve principalmente a cargo de la implementación del programa en Python. Desarrollo las funciones para trabajar con los conjuntos de dígitos, programo el conteo de frecuencia, la suma de los dígitos y las condiciones lógicas. También implemento la lógica relacionada con los años de nacimiento, incluyendo la detección de años bisiestos y el cálculo del producto cartesiano. Colaboro además en el diseño general del programa y la validación de los resultados.

Conclusiones

El desarrollo de este proyecto integrador nos permitió consolidar de forma práctica los conocimientos adquiridos en Matemática y Programación. Profundizamos nuestra comprensión sobre la relación entre los conjuntos, la lógica y las condiciones matemáticas con la implementación estructurada de algoritmos en Python.

Durante el diseño del programa, reforzamos la importancia de la precisión en la codificación, la necesidad de planificar correctamente las estructuras lógicas, y el impacto que tiene una buena representación matemática al momento de trasladarla a un lenguaje de programación.

Además, trabajar en equipo fue fundamental para distribuir responsabilidades, revisar el código entre pares y explicarnos mutuamente los conceptos más complejos. La colaboración no solo mejoró el resultado final, sino que también enriqueció el proceso de aprendizaje.

Este proyecto fortaleció nuestros conocimientos técnicos y promovió habilidades clave como el trabajo colaborativo, la comunicación efectiva y el desarrollo del pensamiento lógico-computacional. Fue una experiencia completa que integró teoría, práctica y trabajo en equipo de manera significativa.

ANEXOS

Anexo I: Video de presentación

Dentro de este enlace podrán encontrar el video de presentación del trabajo subido a la plataforma de YouTube

Enlace: [www.](http://www.youtube.com)

Anexo II: Repositorio del trabajo

El repositorio contiene todos los archivos del proyecto desarrollados para el Trabajo Integrador 2 de Matemática.

En este espacio podrán encontrar:

- El programa completo en Python (main.py), que implementa todas las operaciones y condiciones lógicas abordadas en el informe.
- Este mismo archivo PDF, con el desarrollo teórico, operaciones entre conjuntos y expresiones lógicas.
- El documento adicional que detalla las responsabilidades de cada integrante y la relación entre teoría y código.
- El archivo README.md, con una descripción general del proyecto, objetivos, herramientas utilizadas y estructura del repositorio.

Este repositorio permite acceder fácilmente a los materiales desarrollados, revisar el código fuente y verificar cómo se integraron los conceptos matemáticos con la lógica de programación.

Enlace: https://github.com/LeoMercorelli/integrador_2-matematica.git

Anexo III: Fuentes

Este trabajo se desarrolló a partir de los contenidos trabajados en clase dentro de las materias **Matemática y Programación I** de la Tecnicatura Universitaria en Programación (UTN). En Matemática, se aplicaron nociones de **teoría de conjuntos, lógica proposicional, operaciones entre conjuntos y razonamiento lógico**. En Programación I, se integraron conceptos como **estructuras condicionales, bucles, funciones, estructuras de datos y la lógica algorítmica**.

Todo el desarrollo fue realizado en lenguaje **Python**, conectando la abstracción matemática con su representación computacional.