

## **Trabajo Práctico Integrador N° 2 - Matemática y Programación**

### **Conectando Conjuntos, Lógica y Código: Desarrollo Colaborativo**

#### **Parte 1**

#### **Alumnos - Grupo N° 1**

Alex Pedro Dauria, Silvia Giardini, Marina Giselle Cordero

Luciano de la Rubia, Laureano Escoca.

**Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional.**

#### **Docente Titular**

Matemática: Carina Jovanovich

Programación I: AUS Bruselario, Sebastián

#### **Docente Tutor**

Matemática: Sebastián Marinier

Programación I: Candia, Verónica

10 de Junio de 2025

## Tabla de contenido

<b>Introducción</b>	2
<b>Desarrollo</b>	3
A) Operaciones con DNIs	3
B) Operaciones con años de nacimiento	9
<b>Contribuciones</b>	14
<b>Conclusión</b>	16
<b>Bibliografía</b>	17

## Introducción

El presente trabajo integrador representa una síntesis entre los fundamentos de matemática y programación, específicamente la teoría de conjuntos y la lógica proposicional, con las estructuras esenciales de Python (condicionales, repetitivas y funciones).

Nuestro principal objetivo es profundizar la integración entre estos contenidos, buscando desarrollar a través del código lo aprendido en matemática y poniendo en práctica lo visto en Programación I. Además, este proyecto está diseñado para fortalecer el trabajo en equipo, la comunicación clara y la responsabilidad individual en el contexto de un proyecto colaborativo.

La conformación de grupos para este trabajo no solo busca fomentar la colaboración entre pares, una habilidad fundamental para cualquier programador, sino también asegurar que cada integrante asuma responsabilidades específicas, contribuya activamente al proyecto y pueda explicar su aporte tanto en el video final como en la descripción escrita de sus tareas.

## Desarrollo

### Desarrollo Matemático (Conjuntos y Lógica)

#### A) Operaciones con DNIs

1. Cada integrante debe anotar su número de DNI.

Integrante 1: 42.103.090

Integrante 2: 31.058.096

Integrante 3: 36.661.622

Integrante 4: 42.700.909

Integrante 5: 40.109.349

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

#### Conjuntos

	Conjunto	Ordenado ascendente
A	{4, 2, 1, 0, 3, 9}	{0, 1, 2, 3, 4, 9}
B	{3, 1, 0, 5, 8, 9, 6}	{0, 1, 3, 5, 6, 8, 9}
C	{3, 6, 1, 2}	{1, 2, 3, 6}
D	{4; 2; 7; 0; 9}	{0, 2, 4, 7, 9}
E	{4, 0, 1, 9, 3}	{0, 1, 3, 4, 9}

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

### Unión ( $\cup$ )

Unión de A y B	$A \cup B = \{0, 1, 2, 3, 4, 9\} \cup \{0, 1, 3, 5, 6, 8, 9\} = \{0, 1, 2, 3, 4, 5, 6, 8, 9\}$
Unión de C y D	$C \cup D = \{1, 2, 3, 6\} \cup \{0, 2, 4, 7, 9\} = \{0, 1, 2, 3, 4, 6, 7, 9\}$
Unión de D y E	$D \cup E = \{0, 2, 4, 7, 9\} \cup \{0, 1, 3, 4, 9\} = \{0, 1, 2, 3, 4, 7, 9\}$
Unión de todos A, B, C, D, E	$A \cup B \cup C \cup D \cup E = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

### Intersección ( $\cap$ )

Intersección de A y B	$A \cap B = \{0, 1, 2, 3, 4, 9\} \cap \{0, 1, 3, 5, 6, 8, 9\} = \{0, 1, 3, 9\}$
Intersección de C y D	$C \cap D = \{1, 2, 3, 6\} \cap \{0, 2, 4, 7, 9\} = \{2\}$
Intersección de D y E	$D \cap E = \{0, 2, 4, 7, 9\} \cap \{0, 1, 3, 4, 9\} = \{0, 4, 9\}$
Intersección de todos A, B, C, D, E	$A \cap B \cap C \cap D \cap E$ $= \{0, 1, 2, 3, 4, 9\} \cap \{0, 1, 3, 5, 6, 8, 9\} \cap \{1, 2, 3, 6\} \cap \{0, 2, 4, 7, 9\} \cap \{0, 1, 3, 4, 9\}$ $= \emptyset \text{ (conjunto vacío)}$

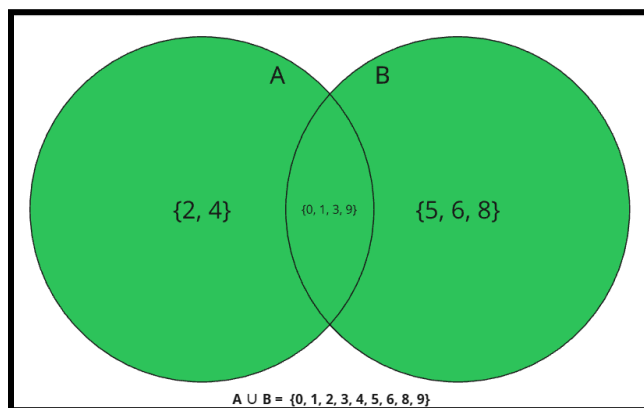
### Diferencia ( $-$ )

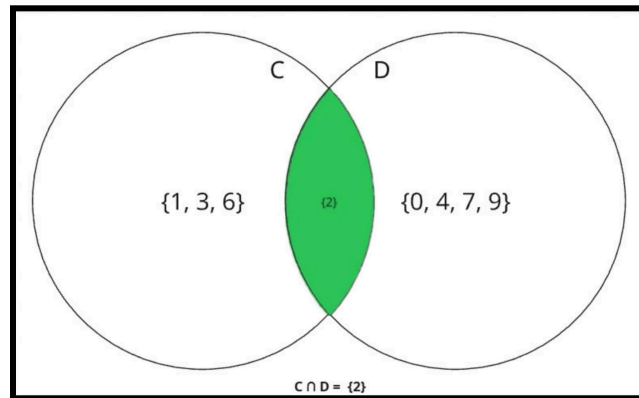
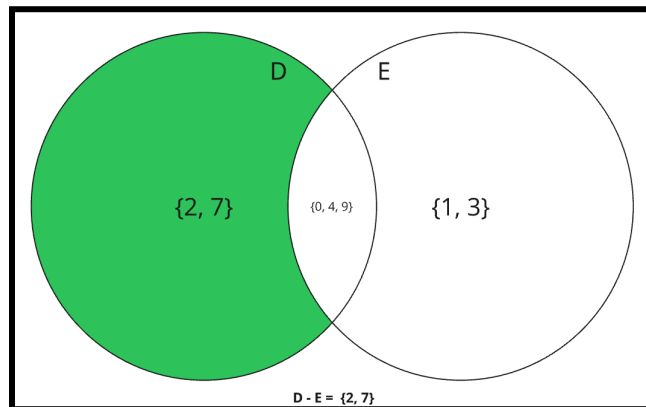
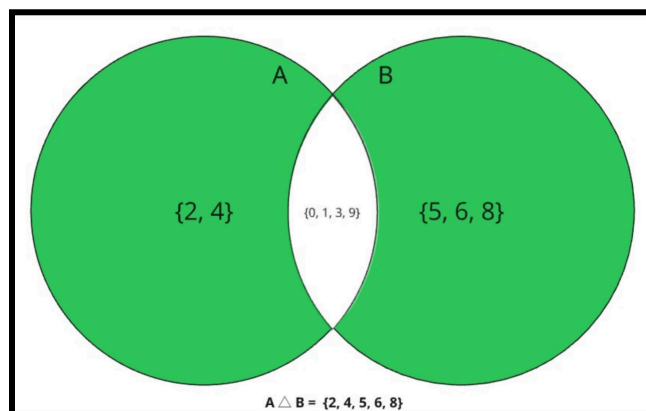
Diferencia de A y B	$A - B = \{0, 1, 2, 3, 4, 9\} - \{0, 1, 3, 5, 6, 8, 9\} = \{2, 4\}$
Diferencia de C y D	$C - D = \{1, 2, 3, 6\} - \{0, 2, 4, 7, 9\} = \{1, 3, 6\}$
Diferencia de D y E	$D - E = \{0, 2, 4, 7, 9\} - \{0, 1, 3, 4, 9\} = \{2, 7\}$

**Diferencia Simétrica ( $\Delta$ )**

Diferencia Simétrica de A y B	$A \Delta B = \{0, 1, 2, 3, 4, 9\} \Delta \{0, 1, 3, 5, 6, 8, 9\} = \{2, 4, 5, 6, 8\}$
Diferencia Simétrica de C y D	$C \Delta D = \{1, 2, 3, 6\} \Delta \{0, 2, 4, 7, 9\} = \{0, 1, 3, 4, 6, 7, 9\}$
Diferencia Simétrica de D y E	$D \Delta E = \{0, 2, 4, 7, 9\} \Delta \{0, 1, 3, 4, 9\} = \{1, 2, 3, 7\}$
Diferencia Simétrica de todos	$A \Delta B \Delta C \Delta D \Delta E$ $= \{0, 1, 2, 3, 4, 9\} \Delta \{0, 1, 3, 5, 6, 8, 9\} \Delta \{1, 2, 3, 6\} \Delta \{0, 2, 4, 7, 9\} \Delta \{0, 1, 3, 4, 9\}$ $= \{2, 4, 5, 7, 8\}$ <p>Para múltiples conjuntos, la diferencia simétrica incluye solo los elementos que aparecen un número impar de veces.</p>

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

**Unión de A y B**

**Intersección de C y D****Diferencia de D y E****Diferencia Simétrica de A y B**

**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.**

### Expresión lógica 1

**Dígitos que están en el conjunto A o en el conjunto B, pero no están en el conjunto C**

- **Operación:**  $(A \cup B) - C$
- **Cálculo:**
  - Unión de A y B:  $\{1, 2, 3, 4, 5, 6, 7, 8\} \cup \{1, 2, 3, 4\} = \{1, 2, 3, 4, 5, 6, 7, 8\}$
  - Diferencia con C:  $\{1, 2, 3, 4, 5, 6, 7, 8\} - \{0, 1, 2, 5, 6, 7, 8, 9\} = \{3, 4\}$
- **Resultado esperado:**  $[3, 4]$
- **Explicación:** Los dígitos 3 y 4 están en A o B (o ambos) y no aparecen en C.

### Proposición Lógica:

$$(x \in A \vee x \in B) \wedge x \notin C$$

Explicación paso a paso:

- $x \in A \vee x \in B$ : significa que el dígito pertenece a alguno de los dos conjuntos.
- $x \notin C$ : Pero no debe pertenecer a C.
- La conjunción de ambos nos da la condición deseada.



**Expresión lógica 2**

**Dígitos que están en el conjunto D, pero no están en la intersección de los conjuntos C y E**

- **Operación:**  $D - (C \cap E)$
- **Cálculo:**
  - *Intersección de C y E:*  $\{0, 1, 2, 5, 6, 7, 8, 9\} \cap \{1, 2, 3, 4\} = \{1, 2\}$
  - *Diferencia con D:*  $\{0, 3, 4, 5, 6, 7, 8, 9\} - \{1, 2\} = \{0, 3, 4, 5, 6, 7, 8, 9\}$
- **Resultado esperado:**  $[0, 3, 4, 5, 6, 7, 8, 9]$
- **Explicación:** Los dígitos 0, 3, 4, 5, 6, 7, 8, 9 están en D y no en la intersección de C y E (que contiene 1 y 2).

**Proposición Lógica (para un elemento x):**

$$(x \in D) \wedge \neg(x \in C \wedge x \in E)$$

Explicación paso a paso:

- $x \in D$ : significa que el dígito pertenece al conjunto D.
- $x \in C \wedge x \in E$ : significa que el dígito pertenece a ambos conjuntos (intersección).
- $\neg (...)$ : Y el dígito no puede estar en ambos conjuntos C y E al mismo tiempo..

## B) Operaciones con años de nacimiento.

### Consigna:

Desarrolle un programa que cumpla con los siguientes requisitos:

- Que permita el ingreso de los años de nacimiento de los integrantes del grupo. Si dos o más integrantes comparten el mismo año de nacimiento, se debe ingresar un dato ficticio para diferenciarlos, según corresponda.
- Utilice estructuras repetitivas para contar cuántos integrantes nacieron en años pares y cuántos en años impares.
- Determine si todos los integrantes nacieron después del año 2000. En caso afirmativo, muestre el mensaje "Grupo Z".
- Identifique si alguno de los años de nacimiento corresponde a un año bisiesto y, de ser así, muestre el mensaje "Tenemos un año especial".
- Implemente una función que determine si un año es bisiesto (un año es bisiesto si es divisible por 4, pero no por 100, salvo que también sea divisible por 400).
- Calcule el producto cartesiano entre el conjunto de años de nacimiento y el conjunto de edades actuales de los integrantes (calculadas restando cada año de nacimiento al año actual, 2025).

## Desarrollo Matemático (Conjuntos y Lógica)

La consigna y posterior desarrollo de código, aplica conceptos de matemáticas, específicamente de teoría de conjuntos y lógica.

### 1. Teoría de Conjuntos:

- Los años de nacimiento (*años*) y edades (*edades*) se tratan como conjuntos finitos.
- *producto\_cartesiano* implementa el producto cartesiano conjunto A x conjunto B, generando todas las combinaciones de años y edades.
- *verificar\_grupo\_z* verifica si el conjunto de años es un subconjunto de los años posteriores a 2000.
- *cont\_pares\_impares* divide el conjunto de años en subconjuntos disjuntos (no tienen elementos en común) (pares e impares) y calcula sus cardinalidades.

### 2. Lógica:

- *ingreso\_año\_nacimiento()* Esta función solicita años de nacimiento y valida las entradas. Las condiciones lógicas son:

**Condición para aceptar un año ingresado:**

- "El valor ingresado es un número entero Y el año no está en la lista de años previos Y el año es mayor o igual a 1900 Y menor o igual a 2025."

**Proposición Lógica:**

$$(\text{año} \in \mathbb{Z}) \wedge \neg(\text{año} \in \text{años previos}) \wedge (\text{año} \geq 1900) \wedge (\text{año} \leq 2025))$$

**Explicación:**

- **año  $\in \mathbb{Z}$**  : El año pertenece al grupo de los enteros.  
( $\mathbb{Z}$  representa el conjunto de todos los números enteros, positivos, negativos y el cero).
- **$\wedge \neg(\text{año} \in \text{años previos})$** : Y el año no debe haber sido ingresado antes.  
Es decir, no debe estar en la lista de años ya registrados.
- **$\wedge (\text{año} \geq 1900)$**  : Y el año debe ser 1900 o mayor.
- **$\wedge (\text{año} \leq 2025)$** : Y el año debe ser 2025 o menor.
- ***cont\_pares\_impares(anios)*** Esta función cuenta años pares e impares. La condición lógica determina si un año es par o impar.

**Condición para clasificar un año:**

- "Si el año es divisible por 2 (es decir, el resto de dividirlo por 2 es 0), entonces es par; de lo contrario, es impar."

**Proposición Lógica:**

$$(a \bmod 2 = 0) \rightarrow \text{Par}(a)$$

$$\neg(a \bmod 2 = 0) \rightarrow \text{Impar}(a):$$

**Explicación primera proposición:**

**a mod 2 = 0:** significa que el residuo de dividir el número 'a' entre 2 es igual a 0.

$\rightarrow$  **Par(a):** Esto implica que: si se cumple la condición, 'a' es un número par.

**Explicación segunda proposición:**

**a mod 2 = 0:** significa que el residuo de dividir el número 'a' entre 2 es igual a 0.

**$\neg(a \bmod 2=0)$ :** Negamos lo anterior, o sea: El número no es divisible por 2

→ Esto implica que al dividir 'a' entre 2, el residuo es distinto de 0.

→ **Impar(a)**: Entonces, el número es impar.

- **verificar\_grupo\_z(anios)** Esta función verifica si todos los años son posteriores al 2000 (Generación Z).

#### Condición para ser Generación Z:

- "Todos los años de la lista deben ser mayores a 2000."

Equivalente: "No debe existir ningún año en la lista que sea menor o igual a 2000."

#### Proposición Lógica:

$$\forall a \in \text{anios} \wedge (a > 2000)$$

#### Explicación de símbolos:

- $\forall$ : "Para todo" o "Para cada".
  - $a \in \text{anios}$ : El elemento a pertenece al conjunto de años llamados **anios**.
  - $\wedge (a > 2000)$ : Y el año debe ser mayor que 2000.
- **es\_bisiesto(anio)** Esta función determina si un año es bisiesto.

#### Condición para que un año sea bisiesto:

- "El año es bisiesto si es divisible por 4 Y no es divisible por 100, O si es divisible por 400."

#### Proposición Lógica:

$$((a \bmod 4 = 0) \wedge (a \bmod 100 \neq 0)) \vee (a \bmod 400 = 0)$$

**Explicación:**

- **$a \bmod 4 = 0$** : El año es divisible por 4 (o sea, su resto al dividirlo entre 4 es 0).
  - **$\wedge (a \bmod 100 \neq 0)$** : y el año **no** es divisible por 100 (el resto no es 0).
  - **$\vee (a \bmod 400 = 0)$** : O el año es divisible por 400 (resto 0 al dividirlo entre 400).
- ***verificar\_bisiesto(anios)*** Esta función verifica si hay al menos un año bisiesto en la lista.

**Condición para imprimir un mensaje:**

- "Si existe al menos un año en la lista que sea bisiesto, entonces imprime 'Tenemos un año especial'."

**Proposición Lógica:**

$\exists a \in \text{anios}: ((a \bmod 4 = 0 \wedge a \bmod 100 \neq 0) \vee (a \bmod 400 = 0)) \rightarrow$   
 Imprimir("Tenemos un año especial")

**Explicación:**

- **$\exists a \in \text{anios}$** : Existe al menos un año  $a$  dentro del conjunto o lista **anios**.
- **$(a \bmod 4 = 0 \wedge a \bmod 100 \neq 0)$** : tal que el año es divisible por 4 **y** no divisible por 100.
- **$\vee (a \bmod 400 = 0)$** : o el año es divisible por 400.
- **$\rightarrow \text{Imprimir("Tenemos un año especial")}$** : Entonces, se imprime el mensaje: 'Tenemos un año especial'.

## Contribuciones

Cabe destacar que todos los integrantes participaron activamente, no solo realizando una parte específica de las actividades sino también, supervisando el trabajo en general; desde los cálculos matemáticos iniciales hasta la depuración y validación el código Python, garantizando que las funciones de conjuntos arrojen los valores correctos según la lógica establecida.

Buscando errores, corrigiendolos y acompañándonos entre todos para llevar a cabo este proyecto de forma exitosa.

### Contribuciones Individuales

#### **Silvia Giardini**

- Creó la estructura base del informe en PDF.
- Redactó la introducción y la versión preliminar del “Documento complementario sobre contribuciones del equipo”.
- Formó los conjuntos de dígitos a partir de los DNIs del grupo.
- Realizó las operaciones de unión e intersección (total y entre pares).
- Propuso las expresiones lógicas iniciales.
- Colaboró en el planteo general del código basado en lógica de conjuntos.

#### **Alex Pedro Dauria**

- Se encargó del desarrollo de los cálculos de diferencia y diferencia simétrica entre conjuntos.
- Diseñó los diagramas de Venn correspondientes.
- Reformuló las expresiones lógicas iniciales.

**Marina Giselle Cordero**

- Supervisó cada consigna y corrigió errores en la redacción del informe.
- Revisó expresiones lógicas y colaboró con el análisis matemático del ejercicio B (años de nacimiento).

**Luciano de la Rubia**

- Participó activamente en el desarrollo de la parte 2

**Laureano Escoca**

- Participó activamente en el desarrollo de la parte 2



## Conclusión

Esta primera parte del trabajo permitió aplicar conceptos fundamentales de teoría de conjuntos y lógica matemática en un contexto práctico y colaborativo. A través del análisis de los DNIs y la construcción de conjuntos, realizamos operaciones como unión, intersección, diferencia y diferencia simétrica, visualizándose mediante diagramas de Venn. Además, redactamos e interpretamos expresiones lógicas que reflejan relaciones entre los conjuntos, una base esencial para su posterior implementación en Python. La actividad reforzó no solo los contenidos teóricos, sino también la importancia del trabajo en equipo y la división de tareas dentro de un proyecto.

## Bibliografía

### Universidad Tecnológica Nacional

#### Lógica en Programación

<https://tup.sied.utn.edu.ar/mod/resource/view.php?id=3627>

#### Material Teórico Lógica

<https://tup.sied.utn.edu.ar/mod/resource/view.php?id=5262>

<https://tup.sied.utn.edu.ar/mod/resource/view.php?id=5263>

#### Teoría de Conjuntos

[https://youtu.be/q1tB2H\\_Kgk](https://youtu.be/q1tB2H_Kgk)

#### Producto cartesiano y relaciones binarias

<https://youtu.be/hr1c3HF7p0A>

#### Material Teórico Conjuntos

<https://tup.sied.utn.edu.ar/mod/resource/view.php?id=3410>

<https://tup.sied.utn.edu.ar/mod/resource/view.php?id=3411>

#### Lectura Sugerida Conjuntos

<https://tup.sied.utn.edu.ar/mod/resource/view.php?id=6316>

### Fuentes Externas

#### Teoría de Conjuntos

<https://roa.cedia.edu.ec/webappscode/21/index.html>

#### Aplicación de Conjuntos en la Computación

<https://prezi.com/yddt0aj1c5oe/aplicaciones-de-conjuntos-en-la-computacion/>

### Uso de Inteligencia Artificial

Durante la elaboración de este trabajo académico, se utilizó inteligencia artificial, en particular ChatGPT de OpenAI y Grok, para asistir en la organización del contenido, redacción de explicaciones técnicas y apoyo en la estructuración del informe. Estas herramientas facilitaron la generación de textos claros, la explicación detallada de comandos y la propuesta de metodología acorde al tema tratado.

Se reconoce que el uso de IA fue un complemento y apoyo en la construcción del trabajo, pero las decisiones sobre los comandos utilizados, interpretaciones del escaneo y análisis fueron realizadas por los autores.