

## Trabajo Integrador N.º 2: Matemática - Conjuntos

HERNÁN CASALDERREY - CHRISTIAN BUSTAMANTE - Comisión 6 – 2025

Youtube: <https://youtu.be/hfKS668CIGE>

DNI Christian : 31644815

DNI Hernan: 30576434

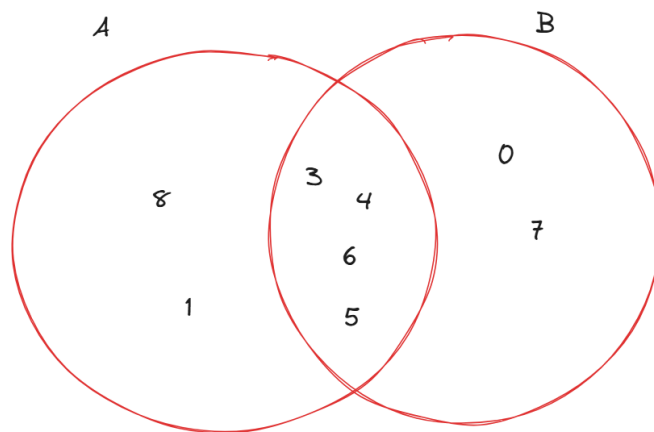
### Parte A: Operaciones con DNIs

si tomamos los dígitos de cada número:

- $31644815 \rightarrow \{1, 3, 4, 5, 6, 8\}$
- $30576434 \rightarrow \{0, 3, 4, 5, 6, 7\}$

Entonces podríamos hacer:

- **Unión:** todos los dígitos únicos de ambos  $\rightarrow \{0, 1, 3, 4, 5, 6, 7, 8\}$
- **Intersección:** dígitos comunes  $\rightarrow \{3, 4, 5, 6\}$
- **Diferencia A-B** ( $31644815 - 30576434$ ):  $\{1, 8\}$
- **Diferencia B-A** ( $30576434 - 31644815$ ):  $\{0, 7\}$



A: DNI Christian  
B: DNI Hernan

### Evaluación de expresiones lógicas

1. **Diversidad numérica:** Sí (ambos conjuntos tienen al menos 5 elementos)
2. **Combinación amplia:** No (A no tiene más elementos que B)
3. **Grupo sin ceros:** No (el conjunto B contiene el dígito 0)
4. **Dígito común:** Sí (hay varios dígitos en común: 3, 4, 5, 6)
5. **Grupo par:** Sí (ambos conjuntos tienen una cantidad par de elementos)
6. **Dígito representativo:** No (la intersección tiene más de un elemento)

### *Lenguaje natural:*

#### **Ejemplo 1**

- ❖ **Lenguaje natural:** “Los dígitos que están en A y no están en B”
  - ❖ **Expresión lógica:**  $A - B$
  - ❖ **Resultado** (si  $A = \{1, 3, 4, 5, 6, 8\}$  y  $B = \{0, 3, 4, 5, 6, 7\}$ ):  
→  $\{1, 8\}$
- 

#### **Ejemplo 2**

- ❖ **Lenguaje natural:** “Los dígitos que están en ambos conjuntos”
  - ❖ **Expresión lógica:**  $A \cap B$
  - ❖ **Resultado:**  
→  $\{3, 4, 5, 6\}$
- 

#### **Ejemplo 3**

- ❖ **Lenguaje natural:** “Los dígitos que están en al menos uno de los conjuntos”
  - ❖ **Expresión lógica:**  $A \cup B$
  - ❖ **Resultado:**  
→  $\{0, 1, 3, 4, 5, 6, 7, 8\}$
- 

#### **Ejemplo 4**

- ❖ **Lenguaje natural:** “Los dígitos que están en B pero no en A”
  - ❖ **Expresión lógica:**  $B - A$
  - ❖ **Resultado:**  
→  $\{0, 7\}$
- 

#### **Ejemplo 5**

- ❖ **Lenguaje natural:** “Los dígitos que no se repiten entre A y B”
- ❖ **Expresión lógica:**  $(A - B) \cup (B - A)$
- ❖ **Resultado:**  
→  $\{0, 1, 7, 8\}$

**Explicaciones de cada programa y conclusión.**

Este programa tiene como objetivo aplicar operaciones de conjuntos y lógica matemática a partir de dos números (por ejemplo, DNI), permitiendo explorar relaciones entre sus dígitos y evaluar condiciones lógicas expresadas tanto en lenguaje formal como en lenguaje natural.

### Funcionalidades principales:

#### 1. Conversión a conjuntos de dígitos

Cada número ingresado se convierte en un conjunto de dígitos únicos. Esto permite trabajar con los elementos sin repeticiones y aplicar operaciones de teoría de conjuntos.

#### 2. Operaciones de conjunto

Se realizan operaciones clásicas como:

- ❖ **Unión:** todos los dígitos presentes en al menos uno de los números.
- ❖ **Intersección:** dígitos que aparecen en ambos números.
- ❖ **Diferencia:** dígitos que están en un número pero no en el otro.

#### 3. Conteo de frecuencia de dígitos

Se cuenta cuántas veces aparece cada dígito en cada número, utilizando estructuras repetitivas. Esto permite detectar repeticiones y patrones.

#### 4. Suma total de dígitos

Se calcula la suma de todos los dígitos de cada número, lo cual puede ser útil para análisis numéricos o validaciones.

#### 5. Evaluación de condiciones lógicas

Se implementan expresiones condicionales como:

- ❖ Si todos los conjuntos tienen al menos 5 elementos → **Alta diversidad numérica**
- ❖ Si algún conjunto tiene más de 6 elementos → **Diversidad numérica alta**
- ❖ Si algún dígito aparece en todos los conjuntos → **Dígito compartido**
- ❖ Si algún conjunto contiene un número impar → **Combinación amplia**
- ❖ Si ningún conjunto contiene el dígito 0 → **Grupo sin ceros**
- ❖ Si la intersección tiene exactamente un elemento → **Dígito representativo**

#### 6. Expresiones en lenguaje natural

Se incluyen descripciones comprensibles para interpretar los resultados, como:

- ❖ “Los dígitos que están en A y no están en B”
- ❖ “Los dígitos que están en ambos conjuntos”
- ❖ “Los dígitos que no se repiten entre A y B”

### Operaciones con años de nacimiento”

*Este programa fue desarrollado con el objetivo de realizar distintos análisis y operaciones lógicas a partir de una lista de años de nacimiento ingresados por el usuario. A continuación, se detallan las funcionalidades implementadas:*

### 1. Ingreso de datos

El usuario debe ingresar los años de nacimiento de los integrantes del grupo, separados por espacios. En caso de que dos personas compartan el mismo año, se puede ingresar un dato ficticio para mantener la unicidad si es necesario.

### 2. Conteo de años pares e impares

El programa recorre la lista de años utilizando una estructura repetitiva (for) y cuenta cuántos de ellos son pares y cuántos impares. Esto permite obtener una primera caracterización del grupo.

### 3. Detección de "Grupo Z"

Se evalúa si **todos los integrantes nacieron después del año 2000**. Si esta condición se cumple, se imprime el mensaje:

**"Grupo Z"**, haciendo referencia a la generación nacida en el siglo XXI.

### 4. Verificación de años bisiestos

Se implementa una función llamada **es bisiesto** que determina si un año es bisiesto, aplicando las reglas del calendario gregoriano. Luego, se verifica si **alguno de los años ingresados es bisiesto**. Si es así, se muestra el mensaje:

**"Tenemos un año especial"**.

### 5. Cálculo de edades actuales

El programa calcula la edad actual de cada integrante restando su año de nacimiento al año actual (2023). Esto genera un nuevo conjunto de datos: las edades.

### 6. Producto cartesiano

Finalmente, se calcula el **producto cartesiano** entre el conjunto de años de nacimiento y el conjunto de edades. Esto genera todas las combinaciones posibles entre ambos conjuntos, lo cual puede ser útil para análisis cruzados o visualizaciones.

### Conclusión

A lo largo de este trabajo práctico se desarrollaron dos programas en Python que abordan problemáticas distintas pero complementarias desde el punto de vista lógico y computacional.

En primer lugar, el análisis de **años de nacimiento** permitió aplicar estructuras repetitivas, condicionales y funciones personalizadas para clasificar datos, detectar patrones (como años pares, impares o bisiestos), y generar combinaciones útiles mediante el producto cartesiano. Este enfoque no solo refuerza conceptos de programación, sino que también permite interpretar datos personales de forma estructurada y significativa.

En segundo lugar, el análisis de **conjuntos de dígitos** a partir de números identificatorios (como DNI) permitió aplicar operaciones de conjuntos, conteo de frecuencias, y evaluación de condiciones lógicas expresadas tanto en lenguaje formal como en lenguaje natural. Esta parte del trabajo destaca la importancia de la lógica matemática en la programación y su aplicación en contextos cotidianos.

Ambos programas demuestran cómo la programación puede ser una herramienta poderosa para organizar, analizar y extraer conclusiones a partir de datos simples. Además, fomentan el pensamiento lógico, la resolución de problemas y la capacidad de traducir situaciones reales a estructuras computacionales claras y eficientes.