

Documento Adicional – Trabajo Práctico Integrador II

Materias

Matemática y Programación I

Grupo

Comisión N°8

Fecha de entrega

09/06/2025

Objetivo

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

Integrantes del Grupo y roles principales en el desarrollo del proyecto

Nombre y Apellido	Rol principal en el proyecto	Puntos realizados
Ramallo Gerónimo	Parte 1: Desarrollo Matemático: Conjuntos.	Puntos 1 al 3
Mubilla Yanela	Parte 1: Desarrollo Matemático: Lógica.	Punto 4
Lahoz Cristian	Parte 2.A: Desarrollo del Programa en Python Operaciones con DNIs.	Puntos 1 al 3
Lagos Alejandro	Parte 2.A: Desarrollo del Programa en Python Operaciones con DNIs.	Puntos 4 al 6
Maldonado Ariana	Parte 2.B: Desarrollo del Programa en Python Operaciones con años de nacimiento.	Puntos 1 al 6

Descripción de tareas realizadas

Ramallo Gerónimo: Desarrolló la primera parte del proyecto, desde el punto 1 hasta el punto 3. Realizó un análisis utilizando los números de DNI de cada integrante del equipo, formó los conjuntos de dígitos únicos y aplicó las operaciones de unión, intersección, diferencia y diferencia simétrica. También elaboró los diagramas de Venn correspondientes, utilizando la herramienta online <https://venngage.com> para lograr una visualización clara y ordenada.

Mubilla Yanela: Trabajó en la primera parte del proyecto, específicamente en el punto 4. Redactó las expresiones lógicas en lenguaje natural basadas en los conjuntos creados, analizó los resultados esperados y llevó a cabo su implementación en código Python.

Lahoz Cristian: Programó la segunda parte del proyecto, Parte 2A, específicamente los puntos 1 al 3. Implementó el ingreso de los DNIs, la generación automática de conjuntos de dígitos únicos y las operaciones de conjuntos en Python (unión, intersección, diferencia y diferencia simétrica).

Lagos Alejandro: Programó la segunda parte del proyecto, Parte 2A, específicamente los puntos 4 al 6. Implementó el conteo de frecuencia de los dígitos, la suma total de los dígitos de cada DNI.

Maldonado Ariana: Programó la Parte 2B del proyecto. Ingresó los años de nacimiento de los integrantes del equipo, contó cuántos nacieron en años pares e impares, verificó si el grupo correspondía al “Grupo Z”, detectó años bisiestos, implementó la función para identificar si un año es bisiesto y calculó el producto cartesiano entre los años de nacimiento y las edades actuales de los integrantes.

Expresiones lógicas desarrolladas y su implementación

A lo largo del trabajo, fuimos planteando distintas expresiones lógicas en lenguaje natural, basadas en datos reales del grupo (documentos y años de nacimiento), y las fuimos implementado en Python. De esta manera, unimos la teoría con la práctica, aplicando la lógica y los conjuntos al análisis de información concreta.

- **Parte 1: Conjuntos y Lógica**

En el punto 5 de esta parte, se trabajaron expresiones lógicas que luego se tradujeron a código. Las que desarrollamos fueron las siguientes:

5. 1 “Si todos los conjuntos contienen al menos n dígitos en común, entonces el grupo tiene n dígitos comunes”.

Para verificar esta condición, se realizó la intersección entre todos los conjuntos de dígitos obtenidos a partir de los DNIs. Si esa intersección tenía n elementos, se consideró que esos eran los dígitos comunes en el grupo.

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

Para esta condición, se recorrieron todos los conjuntos contando cuántos tenían una cantidad impar de dígitos. Luego, se comparó esa cantidad con los que tenían cantidad par, y si los impares eran mayoría, se mostraba el mensaje correspondiente.

Estas expresiones fueron pensadas en un principio a partir del análisis matemático, y después se transformaron en instrucciones en Python.

- **Parte 2A – Aplicación en Python: Operaciones con DNIs**

En esta sección se implementa en Python el análisis de los documentos de identidad (DNIs) de los integrantes del grupo, utilizando conceptos de conjuntos, funciones y estructuras de control. El programa permite ingresar manualmente los DNIs o utilizar un conjunto predefinido para automatizar pruebas y presentaciones.

Una vez ingresados los DNIs, el programa extrae los dígitos únicos de cada uno y genera conjuntos matemáticos individuales, representando así la información numérica de cada integrante de forma abstracta. A partir de estos conjuntos, se ejecutan diversas operaciones fundamentales de la teoría de conjuntos:

- **Unión:** para obtener el conjunto total de dígitos presentes en el grupo.
- **Intersección:** para identificar los dígitos que se repiten en todos los DNIs.
- **Diferencia simétrica:** para detectar los dígitos exclusivos de algún conjunto (que no están en todos).
- **Diferencias entre pares:** se calcula la diferencia de elementos entre cada combinación posible de conjuntos.

Además, se implementan funciones adicionales que permiten analizar los datos desde distintos enfoques:

- Se cuenta la frecuencia de aparición de cada dígito tanto a nivel individual (por DNI) como global (en todos los conjuntos).
- Se suma la totalidad de los dígitos presentes en cada documento como medida numérica individual.
- Se evalúan condiciones lógicas desarrolladas previamente en lenguaje natural, como:

- Si todos los conjuntos tienen al menos un dígito en común.
- Si hay más conjuntos con cantidad impar de elementos que pares, en cuyo caso el grupo se clasifica como "grupo impar".

El diseño modular del programa, basado en funciones específicas para cada tarea, permite mantener un código legible, reutilizable y alineado con los contenidos tanto de Matemática como de Programación. Esta integración entre teoría y práctica refuerza el objetivo pedagógico del trabajo, demostrando cómo es posible analizar información concreta del grupo mediante herramientas computacionales.

- **Parte 2B – Aplicación en Python: Operaciones con años de nacimiento**

En esta última sección se usaron expresiones relacionadas con los años de nacimiento de cada uno de los integrantes del equipo.

A continuación, las mencionamos en detalle:

B.3 “Si todos nacieron después del 2000, mostrar "Grupo Z”.

Se usó la función **all()** para verificar que todos los años ingresados fueran mayores a 2000. Si se cumplía, el programa mostraba “Grupo Z”. Además, agregamos una opción alternativa en caso de que no se cumpliera esta condición. En ese caso, el programa imprime “No todos pertenecen al Grupo Z” como resultado de nuestro análisis, para evitar que no se muestre ningún mensaje. Esta decisión se tomó para que siempre haya una respuesta clara por parte del programa, sea positiva o negativa.

B.4 “Si alguno nació en año bisiesto, mostrar "Tenemos un año especial”.

Se creó una función llamada **año_bisiesto()** que evaluaba si un año cumplía con las condiciones para ser considerado bisiesto. Después con la función **any()**, se verificó si al menos uno de los años del grupo lo era. Al igual que en el caso anterior, se contempló la opción contraria, si ninguno de los años era bisiesto, el programa imprimía el mensaje “No tenemos un año especial”, para completar la lógica y evitar silencios en la ejecución.