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

1.      Cada integrante debe anotar su número de DNI.

PF = 39580665

MF = 44611290

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

PF = {0, 3, 5, 6, 8, 9}

MF = {0, 1, 2, 4, 6, 9}

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

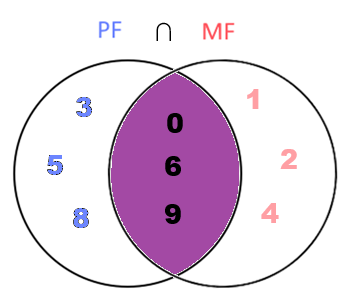
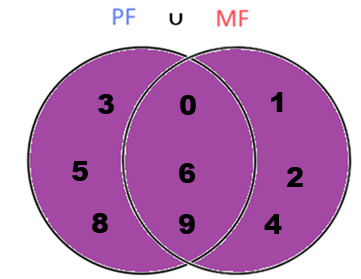
4.      Para cada una de estas operaciones, se debe realizar un diagrama de Venn (a mano o digital), que debe incluirse en la entrega.

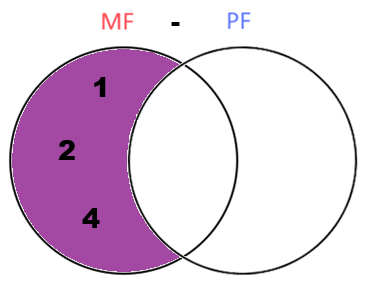
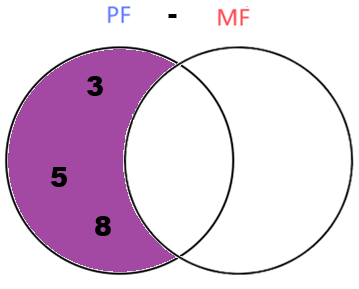
Diagrama, Diagrama de Venn

El contenido generado por IA puede ser incorrecto.Diagrama, Diagrama de Venn

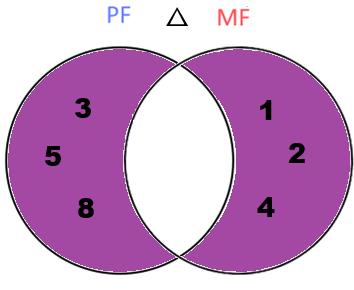
El contenido generado por IA puede ser incorrecto.

**Unión Intersección**

**Diferencia**



**Diferencia Simétrica**



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 cual seria el resultado con los conjuntos que tienen.

Ejemplos de expresiones lógicas:

* Si todos los conjuntos tienen al menos 5 elementos, entonces se considera que hay una alta diversidad numérica.·
* Si el conjunto A tiene más elementos que el conjunto B y el conjunto C contiene al menos un número impar, entonces se cumple la condición de combinación amplia.·
* Si ningún conjunto tiene el número 0, entonces se considera un grupo sin ceros.·
* Si algún dígito aparece en todos los conjuntos, se marca como dígito común.·
* Si hay más conjuntos con cantidad par de elementos que con cantidad impar, entonces se etiqueta como "grupo par".·
* Si la intersección entre todos los conjuntos tiene exactamente un elemento, se considera un dígito representativo del grupo.

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.

* Dígitos que no se encuentran en ningún conjunto del grupo.
  + Ej.: Si ningún conjunto posee el número 7, entonces se considera un grupo sin sietes.
* Si todos los conjuntos tienen al menos 5 elementos, entonces se considera que hay una alta diversidad numérica.
* Si la cantidad de conjuntas es par y tienen una cantidad de elementos par, nos encontramos frente a un grupo “Par de Pares”.

**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.

Ejemplos:

·         Si un dígito aparece en todos los conjuntos, mostrar "Dígito compartido".

·         Si algún conjunto tiene más de 6 elementos, mostrar "Diversidad numérica alta".

**B. Operaciones con años de nacimiento**

·         Ingreso de los años de nacimiento (Si dos o mas 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.**

**E J E M P L O**

Supongamos que en el grupo hay 3 integrantes, con los siguientes DNIs:  
  
● Persona A: 23456789  
● Persona B: 22334455  
● Persona C: 98765432  
  
De cada DNI se toman los dígitos únicos y se arma un conjunto (sin repetir elementos).  
  
● Conjunto A = {2, 3, 4, 5, 6, 7, 8, 9}  
● Conjunto B = {2, 3, 4, 5}  
● Conjunto C = {2, 3, 4, 5, 6, 7, 8, 9}  
  
Ejemplo A y B:  
  
● Unión (A ∪ B): todos los elementos de A y B sin repetir → {2, 3, 4, 5, 6, 7, 8, 9}  
  
● Intersección (A ∩ B): elementos que están en ambos → {2, 3, 4, 5} ● Diferencia (A - B): elementos de A que no están en B → {6, 7, 8, 9}  
  
● Diferencia simétrica (A Δ B): elementos que están en A o en B pero no en ambos → {6, 7, 8, 9} (En este caso, B no tiene ningún dígito que A no tenga, así que la diferencia simétrica es igual a la diferencia A - B.)  
  
Diagrama de Venn: Hay herramientas online como [https://venngage.com](https://venngage.com/) o pueden hacerlo en PowerPoint, Paint o a mano en papel.  
  
Expresiones lógicas en lenguaje natural Estas son frases que expresan condiciones que se pueden traducir a código Python.  
  
Ejemplo 1:Lenguaje natural: “Los dígitos que están en A y no están en B”  
Resultado con los conjuntos del ejemplo: {6, 7, 8, 9}  
  
Ejemplo 2: Lenguaje natural: “Los dígitos que están en A o en B, pero no en ambos”  
Resultado: {6, 7, 8, 9}  
  
Ejemplo 3: “Los dígitos que aparecen en los tres DNIs a la vez”  
Resultado: {2, 3, 4, 5}