Matemática y Programación - Trabajo Integrador N° 2
Conjuntos y lógica.

Grupo: N° 22 **Comisión**: 16.

Integrantes del grupo:

Pablo Mariasch – pablomariasch85@gmail.com

Francisco Barandiarán Pernuzzi – faugustobp@gmail.com

Tareas realizadas por cada integrante:

Pablo:

- Se encargó de realizar el documento en Word con toda la estructura del trabajo, que luego se pasó a formato PDF.
- Elaboró los diagramas de Venn para representar las operaciones entre conjuntos realizadas por Francisco.
- Redactó y analizó la segunda expresión lógica, relacionada con la detección de conjuntos completamente diferentes.
- En el código, trabajó en la estructura general del programa: diseño modular de funciones, el menú interactivo y la punto B de la Parte 2, que incluye las funciones para análisis lógicos, generación, producto cartesiano, años bisiestos, etc.

Francisco:

- Se encargó de realizar todas las operaciones entre conjuntos, como la unión, intersección, diferencia y diferencia simétrica.
- También redactó y analizó la primera expresión lógica, referida a la detección de conjuntos con tendencia a dígitos bajos.
- En el código, desarrolló el punto A de la Parte 2, que incluye la transformación de DNIs en conjuntos, las operaciones entre ellos y el análisis de frecuencia y suma de dígitos.

Relación entre Expresiones Lógicas y el Código

Primera Expresión Lógica (Francisco)
"Si al menos dos conjuntos tienen algún número menor que 3, se considera un grupo con tendencia a dígitos bajos."

Código relacionado:

```
 99   # Detecta personas con tendencia a usar dígitos bajos (<3)
100   def hay_tendencia_digitos_bajos(conjuntos):
101       return [(k, v) for k, v in conjuntos.items() if any(d < 3 for d in v)]
102
```

```
114   # Evalúa condiciones lógicas sobre los conjuntos de dígitos
115   def imprimir_condiciones_logicas(conjuntos):
116       print(color("\nEvaluación de condiciones lógicas:", "green"))
117
118       tendencia = hay_tendencia_digitos_bajos(conjuntos)
119       if len(tendencia) >= 2:
120           print("→ Grupo con tendencia a dígitos bajos (dígitos < 3):")
121           for nombre, conjunto in tendencia:
122               print(f"  {nombre}: {conjunto}")
123       else:
124           print("→ No hay suficiente tendencia a dígitos bajos.")
```

Explicación:
El código evalúa cada conjunto de dígitos únicos. Si encuentra al menos dos personas con algún dígito menor que 3, imprime que hay una "tendencia a dígitos bajos".

Segunda Expresión Lógica (Pablo)
“Si dos conjuntos no tienen ningún elemento en común entre sí, entonces se consideran conjuntos completamente diferentes.”

Código relacionado:

```
103  # Detecta pares de personas cuyos conjuntos no comparten ningún dígito
104  def hay_conjuntos_completamente_diferentes(conjuntos):
105      claves = list(conjuntos.keys())
106      resultados = []
107      for i in range(len(claves)):
108          for j in range(i + 1, len(claves)):
109              c1, c2 = claves[i], claves[j]
110              if conjuntos[c1].isdisjoint(conjuntos[c2]):
111                  resultados.append((c1, c2))
112      return resultados
113
```

```
126      disjuntos = hay_conjuntos_completamente_diferentes(conjuntos)
127      if disjuntos:
128          print("→ Hay conjuntos completamente diferentes:")
129          for par in disjuntos:
130              print(f"  {par[0]} y {par[1]}")
131      else:
132          print("→ No hay conjuntos completamente diferentes.")
133
```

Explicación:
El código busca pares de conjuntos que no comparten ningún dígito. Si encuentra alguno, imprime sus nombres.