

# Informe de Integrantes y Relación entre Expresiones Lógicas y Código.

## Nombres y Tareas realizadas:

**Rapetti Sebastián:** Redacción de expresiones lógicas, implementación del código en Python, realización de los diagramas de Venn, realización del primer archivo PDF.

**Rosas Alejo:** Evaluación de condiciones lógicas, operaciones entre conjuntos, funciones y condicionales. Análisis de años de nacimiento para determinar pares e impares. Edición de video + Informe.

## Relación entre expresiones lógicas y código:

### Expresiones lógicas:

**Unión:**  $A \cup B = \{0,2,4,5,6,7\}$

**Intersección:**  $A \cap B = \{4,6,7\}$

**Diferencia:**  $A - B = \{5\}$

**Diferencia simétrica:**  $A \Delta B = \{0,2,5\}$

```
# 3) CALCULO Y VISUALIZACION DE:  
# unión, intersección, diferencia y diferencia simétrica  
union = set.union(*conjunto_digitos) # se usa para calcular la unión de los conjuntos de dígitos únicos  
interseccion = set.intersection(*conjunto_digitos) # se usa para calcular la interseccion de dos conjuntos  
diferencia = conjunto_digitos[0] - conjunto_digitos[1]  
diferencia_simetrica = set.symmetric_difference(conjunto_digitos[0],conjunto_digitos[1])
```

## Expresiones Lógicas redactadas con código:

***“Si todos los conjuntos tienen menos de 3 números se considera que no tienen mucha densidad numérica.”***

```
# El condicional se encarga de ver si todos los conjuntos tiene al menos 5 digitos unicos.
if all(len(conjunto) >= 5 for conjunto in conjunto_digitos):
# En caso de que se cumpla la condicion, este mismo imprime alta diversidad numerica.
    print("Alta diversidad numérica.")
# En caso de que no se cumpla la condicion imprime no hay diversidad numerica
else:
    print("No hay diversidad numérica suficiente.")
```

***“Si ninguno de los dos conjuntos tiene sólo números pares, entonces no es un conjunto de números pares.”***

```
# En esta porcion de codigo se cuenta cuantos conjuntos tienen una cantidad par de elementos.
pares = sum(1 for conj in conjunto_digitos if len(conj) % 2 == 0)
# Tambien se compara con los que tienen cantidad impar
impares = len(conjunto_digitos) - pares
```

***“Si la intersección tiene un único elemento, se considera representativo del grupo”***

```
# Aqui se utiliza la interseccion utilizada anteriormente.
if len(interseccion) == 1:
# Si solo hay un digito comun entre los conjuntos se lo considera "Representativo del grupo"
    print(f"Dígito representativo del grupo: {list(interseccion)[0]}")
# Si hay mas de uno o directamente no hay ninguno se muestra "No hay un unico digito representativo"
else:
    print("No hay un único dígito representativo.")
```