

Nombre: Trejo Alex

NRC: 9714

1.1 Parte Teórica (40%):

**1. Responde las siguientes preguntas sobre cuantificadores existencial y universal:**

**a) ¿Qué es un cuantificador existencial y cómo se denota?**

Es un termino que se usa para expresar el grado en que se aplica una propiedad o relación, en este caso el cuantificador se utiliza para afirmar que existe al menos un conjunto en un conjunto que satisface una proposición dada, se denota por  $\exists$ .

**b) ¿Qué es un cuantificador universal y cómo se denota?**

El cuantificador universal se utiliza para afirmar que una proposición es verdadera para todos los elementos de un conjunto, se denota por  $\forall$ .

**2. Indica si las siguientes afirmaciones son verdaderas o falsas en relación a la validez y los sistemas de prueba en predicados:**

**a) La validez de una fórmula depende de su forma lógica.**

Verdadero

**b) Un sistema de prueba en predicados se utiliza para verificar la validez de una fórmula.**

Falsa

**c) La regla de inferencia de modus ponens es comúnmente utilizada en los sistemas de prueba en predicados.**

Verdadero

**3. Completa las siguientes equivalencias lógicas utilizando las leyes de la lógica proposicional:**

**a)  $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$**

$$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$$

**b)  $\neg(p \rightarrow q) \Leftrightarrow p \wedge \neg q$**

$$\neg(p \rightarrow q) \Leftrightarrow p \wedge \neg q$$

**4. Enumera dos ejemplos de conjuntos utilizando la técnica de enumeración, donde cada conjunto contenga al menos tres elementos**

1. Conjunto A: {1,3,5,7,9,11}

2. Conjunto B: {paloma, águila, gorrión}

## 1.2 Parte Práctica (60%):

**1. Escribe un programa en Python que determine la intersección de dos conjuntos ingresados por el usuario. Muestra el resultado por pantalla.**

```
2
3 def leer_conjunto():
4     conjunto = set()
5     elementos = input("Ingrese los elementos del conjunto separados por comas: ")
6     elementos = elementos.split(",")
7     for elemento in elementos:
8         conjunto.add(elemento.strip())
9     return conjunto
10
11
12 print("Ingrese los elementos del primer conjunto:")
13 conjunto1 = leer_conjunto()
14
15 print("Ingrese los elementos del segundo conjunto:")
16 conjunto2 = leer_conjunto()
17 interseccion = conjunto1.intersection(conjunto2)
18
19 print("Conjunto A: ", conjunto1)
20 print("Conjunto B: ", conjunto2)
21
22
23
24 print("La intersección de los conjuntos es: ", interseccion)
25
```

```
Ingrese los elementos del primer conjunto:
Ingrese los elementos del conjunto separados por comas: 4,5,6,4,5,3,4,5
Ingrese los elementos del segundo conjunto:
Ingrese los elementos del conjunto separados por comas: 3,4,5,6,7,6,5
Conjunto A: {'4', '6', '3', '5'}
Conjunto B: {'3', '4', '7', '6', '5'}
La intersección de los conjuntos es: {'4', '5', '6', '3'}
```

**2. Crea una función en Python que reciba dos conjuntos como parámetros y devuelva un nuevo conjunto que contenga los elementos presentes en el primer conjunto pero no en el segundo. Prueba la función con conjuntos de tu elección.**

```

16
17 def leer_conjunto():
18     conjunto = set()
19     elementos = input("Ingrese los elementos del conjunto separados por comas: ")
20     elementos = elementos.split(",")
21     for elemento in elementos:
22         conjunto.add(elemento.strip())
23     return conjunto
24
25 def DiferenciaDeConjuntoAB(conjunto1, conjunto2):
26     conjunto3 = conjunto1.copy()
27     conjunto3.difference_update(conjunto2)
28     return conjunto3
29
30
31 print("Ingrese los elementos del primer conjunto:")
32 conjunto1 = leer_conjunto()
33
34 print("Ingrese los elementos del segundo conjunto:")
35 conjunto2 = leer_conjunto()
36
37 conjunto3=DiferenciaDeConjuntoAB(conjunto1, conjunto2)
38
39 print("La diferencia de A - B:",conjunto3)
40

```

```

Ingrese los elementos del primer conjunto:
Ingrese los elementos del conjunto separados por comas: 5,6,7,8,9
Ingrese los elementos del segundo conjunto:
Ingrese los elementos del conjunto separados por comas: 5,1,2,3
La diferencia de A - B: {'7', '9', '6', '8'}

```

**3. Diseña un programa en Python que solicite al usuario ingresar dos conjuntos y muestre por pantalla si el primero es un subconjunto del segundo.**

```

import random
import string

def leer_conjunto():
    conjunto = set()
    elementos = input("Ingrese los elementos del conjunto separados por comas: ")
    elementos = elementos.split(",")
    for elemento in elementos:
        conjunto.add(elemento.strip())
    return conjunto

print("Ingrese los elementos del primer conjunto:")
conjunto1 = leer_conjunto()

print("Ingrese los elementos del segundo conjunto:")
conjunto2 = leer_conjunto()

print("Conjunto A:",conjunto1)
print("Conjunto B:",conjunto2)

if conjunto1.issubset(conjunto2):
    print("El primer conjunto", 'A', "es un subconjunto del segundo conjunto", 'B')
else:
    print("El primer conjunto", 'A', "no es un subconjunto del segundo", 'B')

```

```

Ingrese los elementos del primer conjunto:
Ingrese los elementos del conjunto separados por comas: 3,5,4,6,5,8
Ingrese los elementos del segundo conjunto:
Ingrese los elementos del conjunto separados por comas: 3,5,4,6,6,8,8,9,7
Conjunto A: {'4', '8', '3', '6', '5'}
Conjunto B: {'4', '9', '8', '3', '6', '7', '5'}
El primer conjunto A es un subconjunto del segundo

```

4. Implementa una función en Python que reciba una lista de conjuntos y devuelva el conjunto potencia de dicha lista. Prueba la función con conjuntos de tu elección.

```

def conjunto_potencia(lista_conjuntos):
    conjunto_resultado = set()
    conjunto_resultado.add(frozenset()) # Agrega el conjunto vacío

    for conjunto in lista_conjuntos:
        nuevos_subconjuntos = set()

        for subconjunto in conjunto_resultado:
            nuevos_subconjuntos.add(subconjunto)
            nuevos_subconjuntos.add(subconjunto.union(conjunto))

        conjunto_resultado = nuevos_subconjuntos

    return conjunto_resultado

def conjunto_interseccion(lista_conjuntos):
    conjuntos = [{1, 2}, {3, 4, 5}, {6}]
    resultado = conjunto_potencia(conjuntos)
    print(resultado)

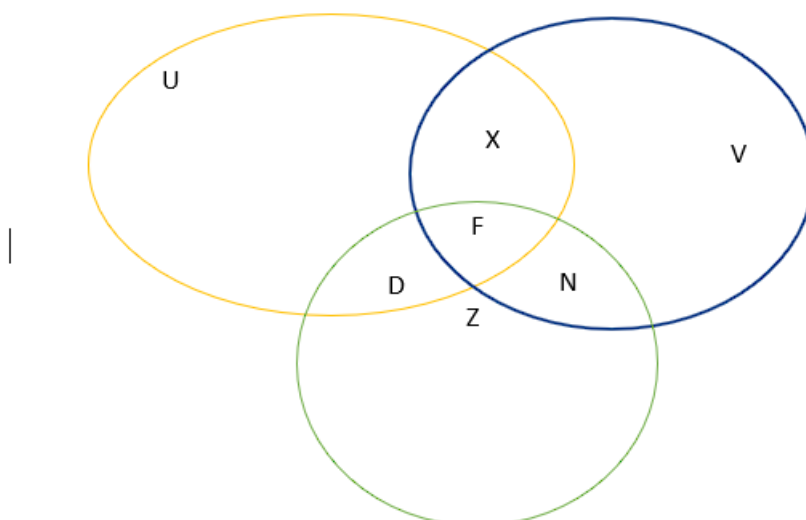
```

```

{frozenset({3, 4, 5, 6}), frozenset({1, 2}), frozenset({1, 2, 3, 4, 5}), frozenset({6}), frozenset({1, 2, 3, 4, 5, 6}), frozenset({3, 4, 5}), fr
ozenset(), frozenset({1, 2, 6})}

```

5. Dibuja un diagrama de Venn-Euler que represente la relación entre tres conjuntos de tu elección. Utiliza colores diferentes para cada conjunto y etiqueta los elementos correspondientes.



**6. Realiza la operación de unión de dos conjuntos utilizando una lista en Python. Muestra el resultado por pantalla.**

```
1
2 def leer_conjunto():
3     conjunto = []
4     elementos = input("Ingrese los elementos del conjunto separados por comas: ")
5     elementos = elementos.split(",")
6     for elemento in elementos:
7         conjunto.append(elemento.strip())
8     return conjunto
9
10 print("Ingrese los elementos del primer conjunto:")
11 conjunto1 = leer_conjunto()
12
13 print("Ingrese los elementos del segundo conjunto:")
14 conjunto2 = leer_conjunto()
15
16
17 unionOfSets=conjunto1 + conjunto2
18
19 print("Conjunto A:",conjunto1)
20 print("Conjunto B:",conjunto2)
21 print("Union de A y B:",unionOfSets)
22
```

```
Ingrese los elementos del primer conjunto:
Ingrese los elementos del conjunto separados por comas: 3,4,5,4,3
Ingrese los elementos del segundo conjunto:
Ingrese los elementos del conjunto separados por comas: 1,4,3,5
Conjunto A: ['3', '4', '5', '4', '3']
Conjunto B: ['1', '4', '3', '5']
Union de A y B: ['3', '4', '5', '4', '3', '1', '4', '3', '5']
```