

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?

Un cuantificador existencial es aquel que ayuda a denotar al menos un elemento existente dentro de un conjunto y se denota como " $\exists x$ "

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

Un cuantificador universal es aquel que denota a todos los elementos del universo, y su simbología es " $\forall x$ "

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.

Falsa

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

a) $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

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

b) $\neg(p \rightarrow q) \equiv p \wedge \neg q$

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

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

1. El conjunto de los meses se le denota como $A = \{\text{enero, febrero, marzo, abril, mayo, junio, julio, agosto, septiembre, octubre, noviembre, diciembre}\}$ consta de 12 elementos

2. El conjunto de los números Reales denotado como $C = \{\text{enteros, naturales, racionales, irracionales}\}$ consta de 4 elementos

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.

En la carpeta de EjerciciosP1

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.

En la carpeta de EjerciciosP1

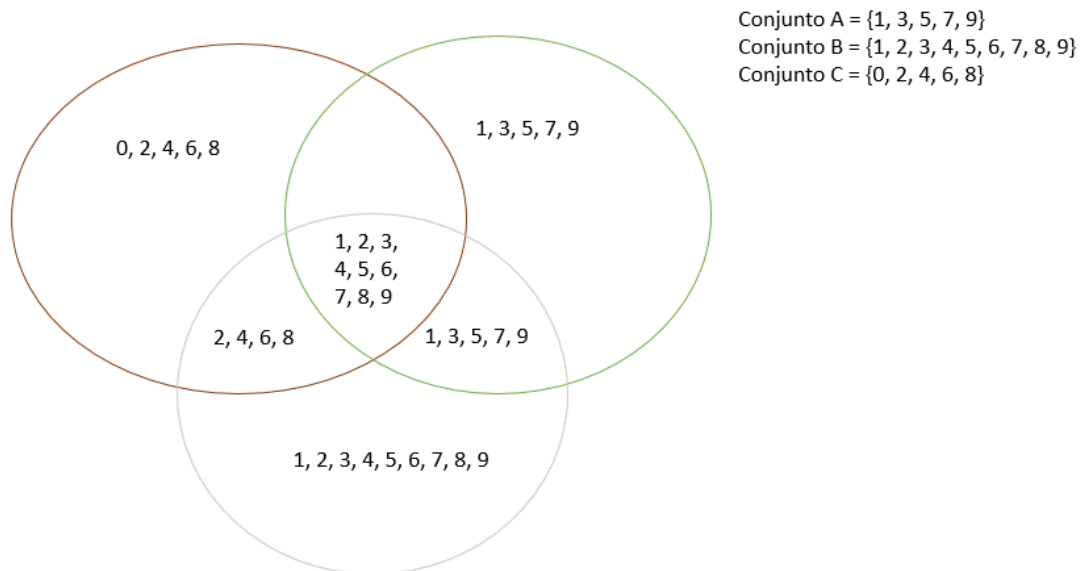
3. Diseña un programa en Python que solicite al usuario ingresar dos conjuntos y muestre por pantalla si el primero es un subconjunto del segundo.

En la carpeta de EjerciciosP1

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.

En la carpeta de EjerciciosP1

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.

En la carpeta de EjerciciosP1