

**PRÁCTICA NO. 1: LÓGICA DE PREDICADOS**

**PROFESORA: ING. JOSUÉ ISRAEL VÁZQUEZ MARTÍNEZ**

**INTELIGENCIA ARTIFICIAL**

**1U**

**8° US**

**ING SISTEMAS COMPUTACIONALES**

**05 MARZO DE 2025**

**YAEL DE JESUS SANTIAGO ORTIZ**

**TECNOLÓGICO NACIONAL DE MÉXICO**

**INSTITUTO TECNOLÓGICO DE TLAXIACO**

**INTRODUCCIÓN**

El análisis lógico en programación permite modelar relaciones entre elementos mediante fórmulas matemáticas y lógicas. En este reporte, se documenta el proceso de creación e implementación de un analizador lógico utilizando la librería NLTK en Python 3.12, con el propósito de representar y manipular expresiones lógicas sencillas.

**OBJETIVO**

El objetivo de esta práctica es implementar un sistema lógico básico utilizando NLTK para definir constantes, predicados y fórmulas lógicas, así como ejecutar un programa que imprima las relaciones definidas entre distintas constantes.

**MATERIAL Y EQUIPO NECESARIO**

- Computadora con editor de texto (Visual Studio Code).

- Python 3.12.

- Librería NLTK.

- Acceso a internet (para investigar si es necesario).

DESARROLLO

Paso 1: Creación del entorno virtual

Se creó una carpeta llamada `E\_U1\_IA` y dentro de ella se generó un entorno virtual mediante el siguiente comando:

python -m venv venv



Paso 2: Activación del entorno virtual

Se utilizó el siguiente comando:

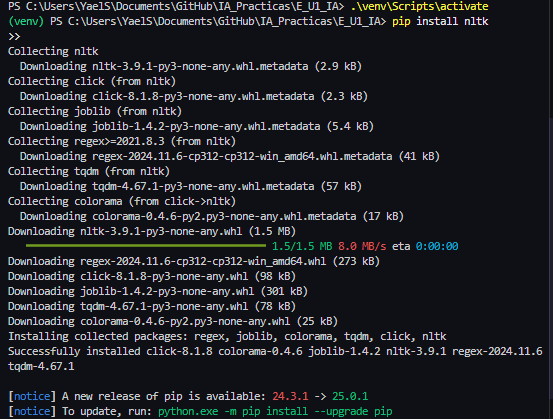
.\venv\Scripts\activate



Paso 3: Instalación de la librería NLTK

Con el entorno virtual activado, se instaló NLTK usando el siguiente comando:

pip install nltk



Paso 4: Creación del archivo `app.py`

Se creó el archivo `app.py` donde se programaron las expresiones lógicas y las fórmulas.

Paso 5: Inicialización del analizador lógico

Se importó la función `Expression.fromstring` de NLTK y se creó un alias `read\_expr` para convertir cadenas de texto en objetos de tipo `Expression`:

Paso 6: Definición de constantes lógicas

Se definieron las constantes lógicas para representar a las personas "yael", "ariel" y "juan":

Paso 7: Creación de predicados

Se definieron predicados que establecen relaciones entre las constantes:

amigos\_yael\_ariel = read\_expr('amigos(yael, ariel)')

amigos\_yael\_juan = read\_expr('no\_son\_amigos(yael, juan)')

no\_amigos\_juan\_ariel = read\_expr('tienen\_la\_misma\_edad(juan, ariel)')

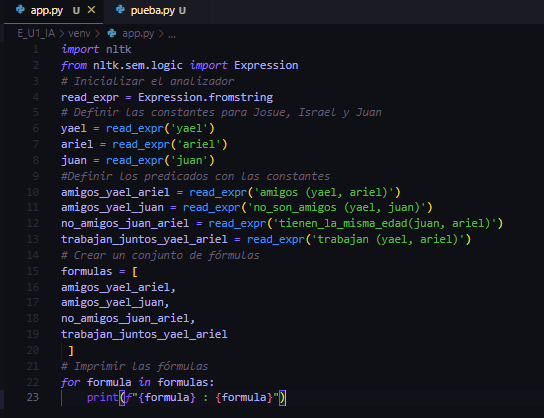
trabajan\_juntos\_yael\_ariel = read\_expr('trabajan(yael, ariel)')

Paso 8: Creación del conjunto de fórmulas

Se almacenaron las fórmulas en una lista llamada `formulas`:

Paso 9: Impresión de las fórmulas

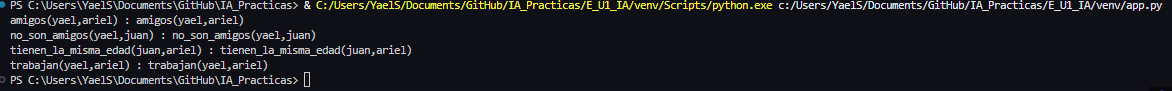
Finalmente, se imprimieron las fórmulas para verificar su correcta interpretación:



Ejecución del programa

El programa se ejecutó con el siguiente comando:

python app.py



**CONCLUSIÓN**

La implementación del analizador lógico básico con NLTK permitió comprender cómo definir constantes, crear predicados y almacenar fórmulas lógicas para modelar relaciones simples entre elementos. El uso de un entorno virtual garantizó un ambiente aislado y controlado, evitando conflictos con otras bibliotecas. Este ejercicio sienta las bases para futuros proyectos de procesamiento lógico más complejos.

BIBLIOGRAFIA

Python Software Foundation. (2025). venv — Creation of virtual environments. Python 3.12. https://docs.python.org/3/library/venv.html

NLTK Team. (2025). Natural Language Toolkit Documentation. https://www.nltk.org/