



**Educación**  
Secretaría de Educación Pública



TECNOLÓGICO  
NACIONAL DE MÉXICO



# **TECNOLÓGICO NACIONAL DE MÉXICO**

## **INSTITUTO TECNOLÓGICO DE TLAXIACO**

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

**Yael de Jesús Santiago Ortiz**

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

**INTELIGENCIA ARTIFICIAL**

**1U**

**8° US**

**ING SISTEMAS  
COMPUTACIONALES**

**05 MARZO DE 2025**

## 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
```

```
PS C:\Users\YaelS\Documents\GitHub\IA_Practicas> cd E_U1_IA
PS C:\Users\YaelS\Documents\GitHub\IA_Practicas\E_U1_IA> python -m venv venv
```

### Paso 2: Activación del entorno virtual

Se utilizó el siguiente comando:

```
.\venv\Scripts\activate
```

```
PS C:\Users\YaelS\Documents\GitHub\IA_Practicas\E_U1_IA> .\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
```

```
(venv) PS C:\Users\YaelS\Documents\GitHub\IA_Practicas\E_U1_IA> pip install nltk
>>
Collecting nltk
  Downloading nltk-3.9.1-py3-none-any.whl.metadata (2.9 kB)
Collecting click (from nltk)
  Downloading click-8.1.8-py3-none-any.whl.metadata (2.3 kB)
Collecting joblib (from nltk)
  Downloading joblib-1.4.2-py3-none-any.whl.metadata (5.4 kB)
Collecting regex<=2021.8.3 (from nltk)
  Downloading regex-2024.11.6-cp312-cp312-win_amd64.whl.metadata (41 kB)
Collecting tqdm (from nltk)
  Downloading tqdm-4.67.1-py3-none-any.whl.metadata (57 kB)
Collecting colorama (from click->nltk)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Downloading nltk-3.9.1-py3-none-any.whl (1.5 MB)
  1.5/1.5 MB 8.0 MB/s eta 0:00:00
Downloading regex-2024.11.6-cp312-cp312-win_amd64.whl (273 kB)
Downloading click-8.1.8-py3-none-any.whl (98 kB)
Downloading joblib-1.4.2-py3-none-any.whl (301 kB)
Downloading tqdm-4.67.1-py3-none-any.whl (78 kB)
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: regex, joblib, colorama, tqdm, click, nltk
Successfully installed click-8.1.8 colorama-0.4.6 joblib-1.4.2 nltk-3.9.1 regex-2024.11.6
tqdm-4.67.1

[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

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)')
```

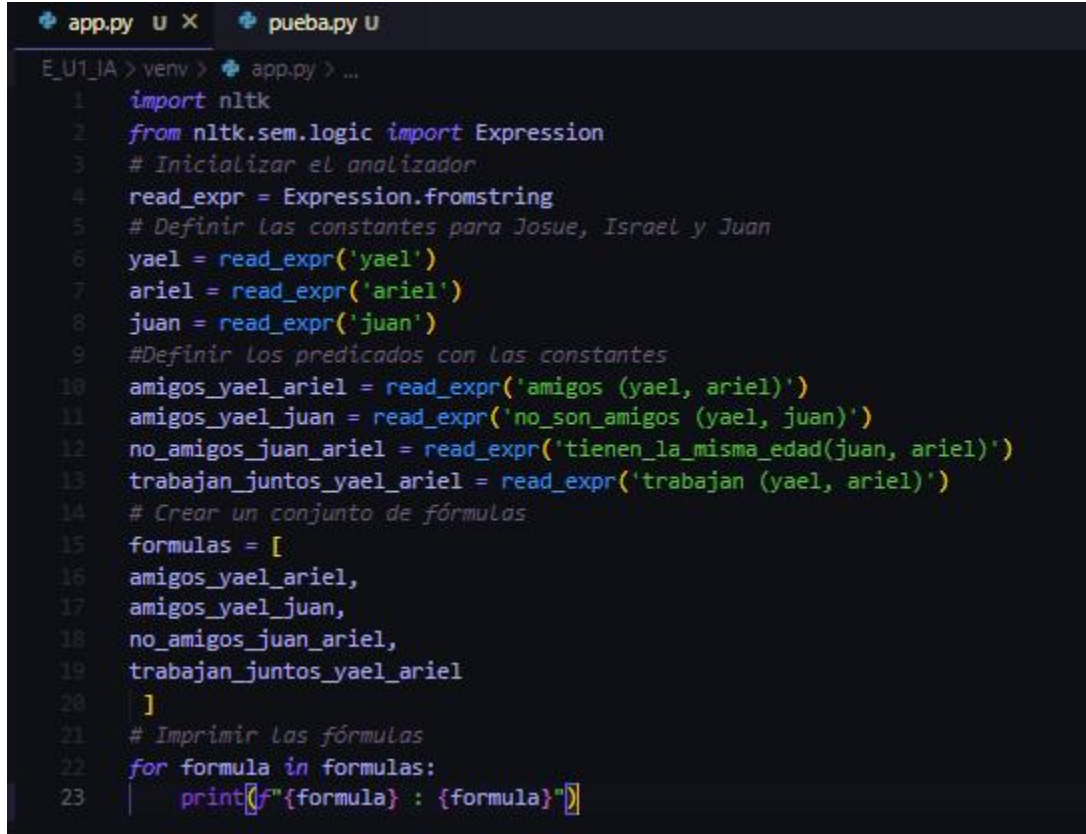
```
trabajan_juntos_yael_ariel = read_expr('trabajan(yael, ariel)')
```

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:

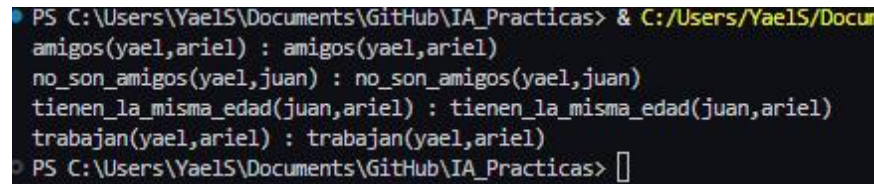


```
app.py U X pueba.py U
E_U1_IA > venv > app.py > ...
1 import nltk
2 from nltk.sem.logic import Expression
3 # Inicializar el analizador
4 read_expr = Expression.fromstring
5 # Definir las constantes para Josue, Israel y Juan
6 yael = read_expr('yael')
7 ariel = read_expr('ariel')
8 juan = read_expr('juan')
9 #Definir los predicados con las constantes
10 amigos_yael_ariel = read_expr('amigos (yael, ariel)')
11 amigos_yael_juan = read_expr('no_son_amigos (yael, juan)')
12 no_amigos_juan_ariel = read_expr('tienen_la_misma_edad(juan, ariel)')
13 trabajan_juntos_yael_ariel = read_expr('trabajan (yael, ariel)')
14 # Crear un conjunto de fórmulas
15 formulas = [
16 amigos_yael_ariel,
17 amigos_yael_juan,
18 no_amigos_juan_ariel,
19 trabajan_juntos_yael_ariel
20 ]
21 # Imprimir las fórmulas
22 for formula in formulas:
23     print(f"{formula} : {formula}")
```

Ejecución del programa

El programa se ejecutó con el siguiente comando:

python app.py



```
PS C:\Users\YaelS\Documents\GitHub\IA_Practicas> & C:/Users/YaelS/Docum
amigos(yael,ariel) : amigos(yael,ariel)
no_son_amigos(yael,juan) : no_son_amigos(yael,juan)
tienen_la_misma_edad(juan,ariel) : tienen_la_misma_edad(juan,ariel)
trabajan(yael,ariel) : trabajan(yael,ariel)
PS C:\Users\YaelS\Documents\GitHub\IA_Practicas> 
```

## 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/>