



UNIVERSIDAD  
Popular del cesar

# Ingeniería de Sistemas

**ESPECIALIZACION EN INGENIERIA DE SOFTWARE**  
**MODULO PATRONES DE DISEÑO DE SOFTWARE**

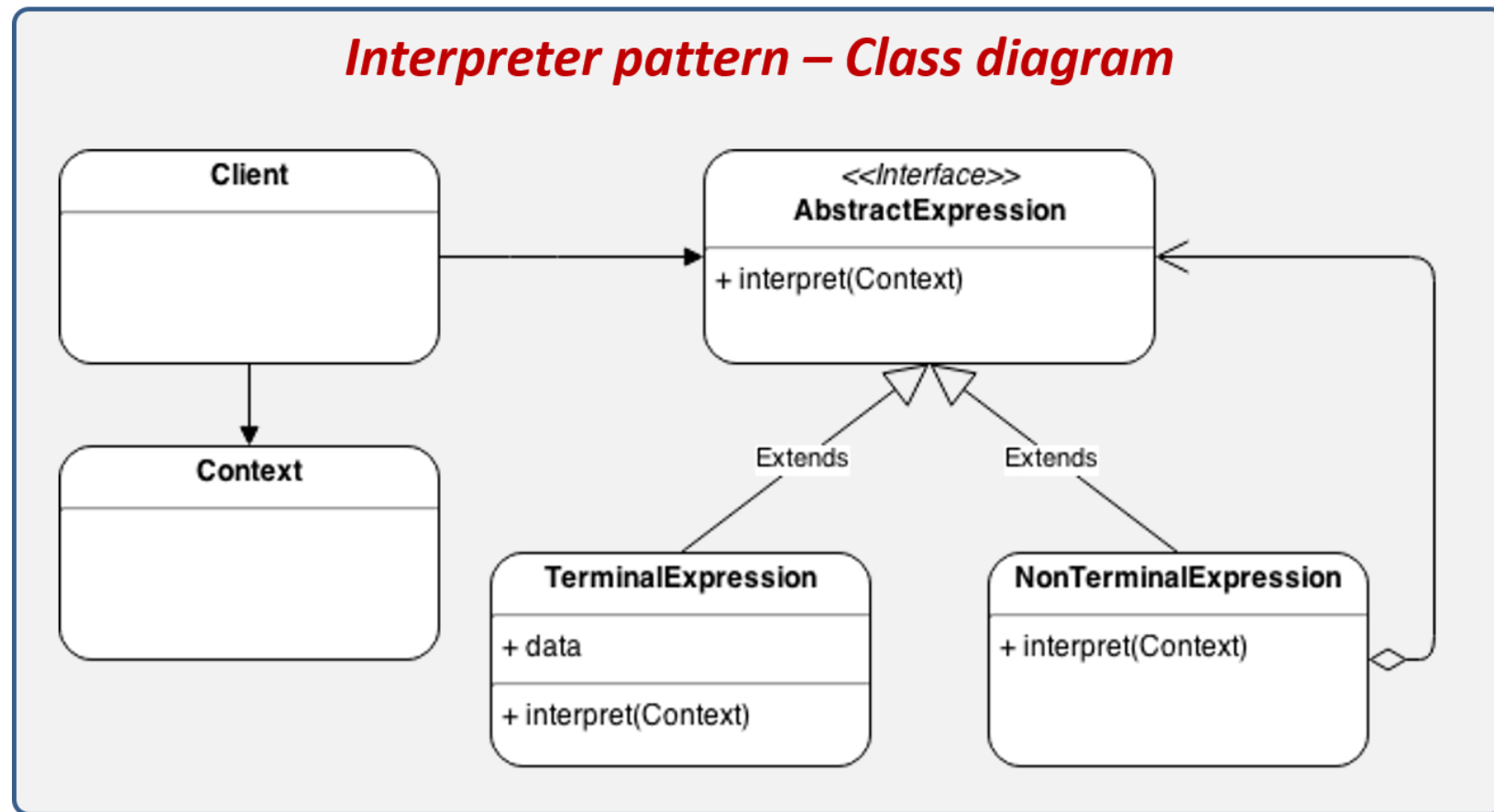
INTEGRANTES:  
CRISTIAN CONTRERAS  
RAFAEL PEDRAZA



## Patrones Diseño – Interpreter

**Interpreter** es un patrón de diseño de comportamiento que define una gramática para interpretar un lenguaje y proporciona una forma de evaluar sentencias en dicho lenguaje..

### *Interpreter pattern – Class diagram*



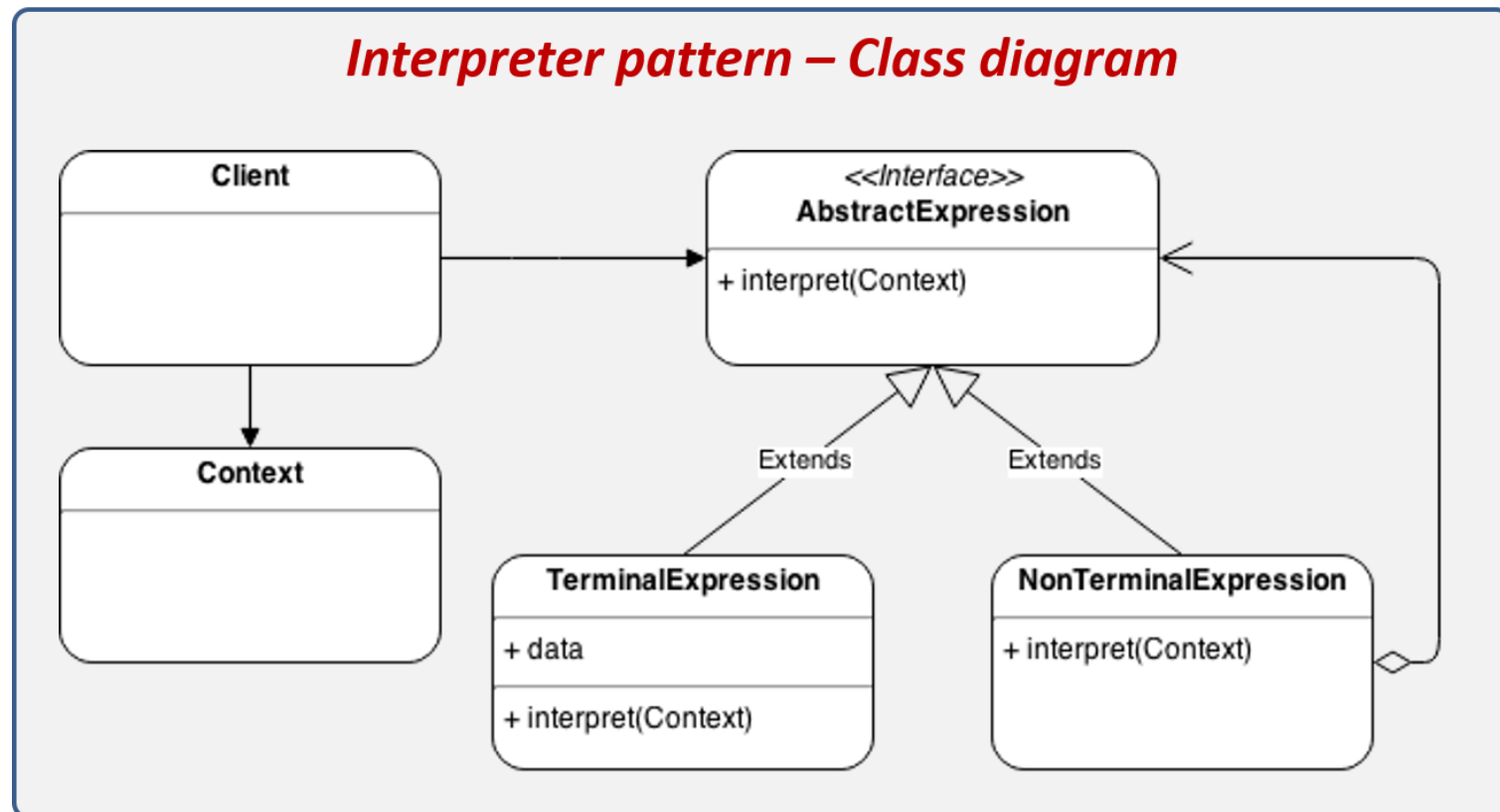
## Patrones diseño – Interpreter

•**Client**: Actor que dispara la ejecución del interpreter.

•**Context**: Objeto con información global que será utilizada por el intérprete para leer y almacenar información global entre todas las clases que conforman el patrón, este es enviado al interpreter el cual lo replica por toda la estructura.

•**AbstractExpression**: Interface que define la estructura mínima de una expresión.

### Interpreter pattern – Class diagram

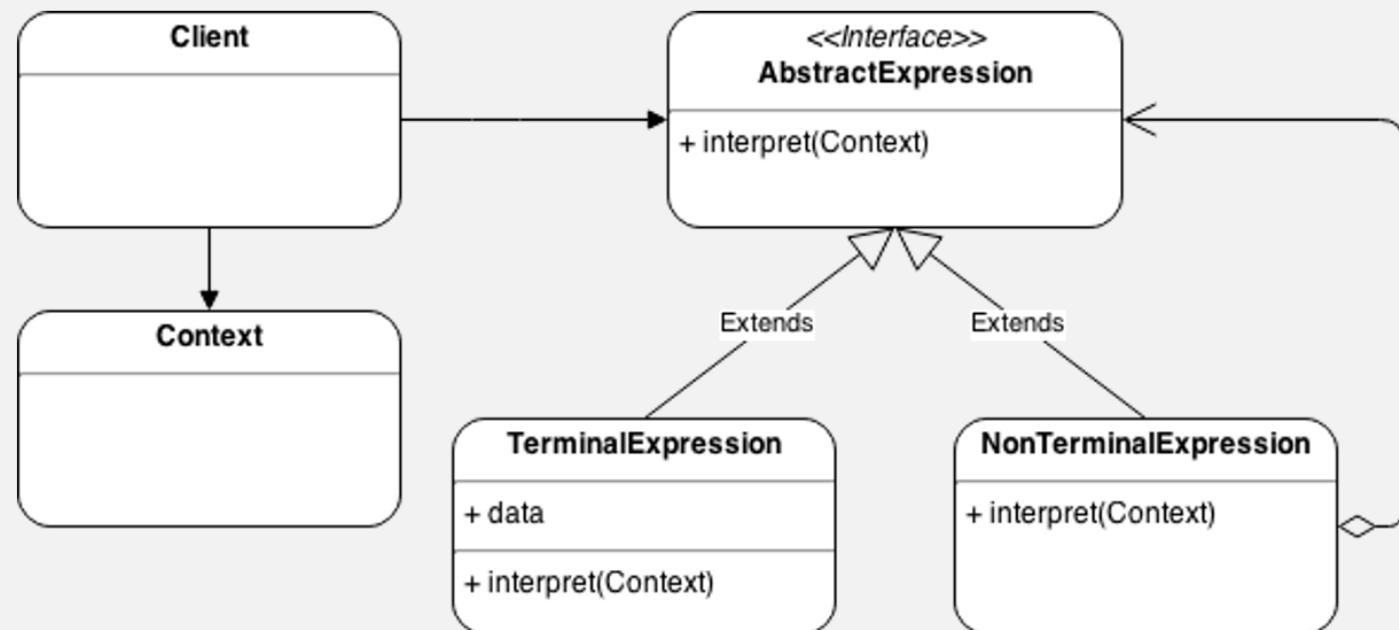


## Patrones diseño – Interpreter

•**TerminalExpression**: Se refiere a expresiones que no tienen más continuidad y al ser evaluadas o interpretadas terminan la ejecución de esa rama. Estas expresiones marcan el final de la ejecución de un sub-árbol de la expresión.

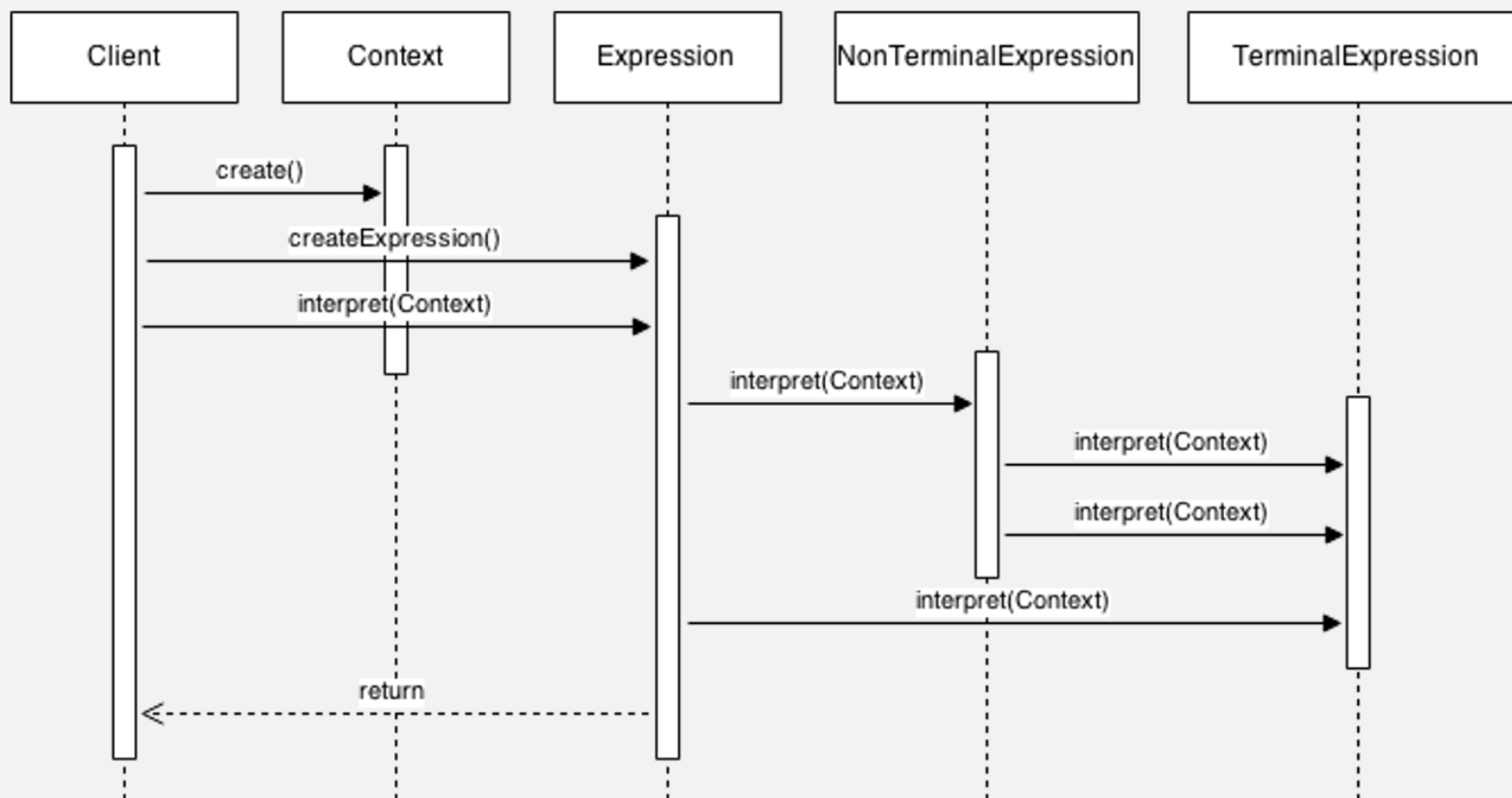
•**NonTerminalExpression**: Son expresiones compuestas y dentro de ellas existen más expresiones que deben ser evaluadas. Estas estructuras son interpretadas utilizando recursividad hasta llegar a una expresión Terminal.

### Interpreter pattern – Class diagram



# Patrones diseño – Interpreter

## *Interpreter pattern – Diagram of sequence*

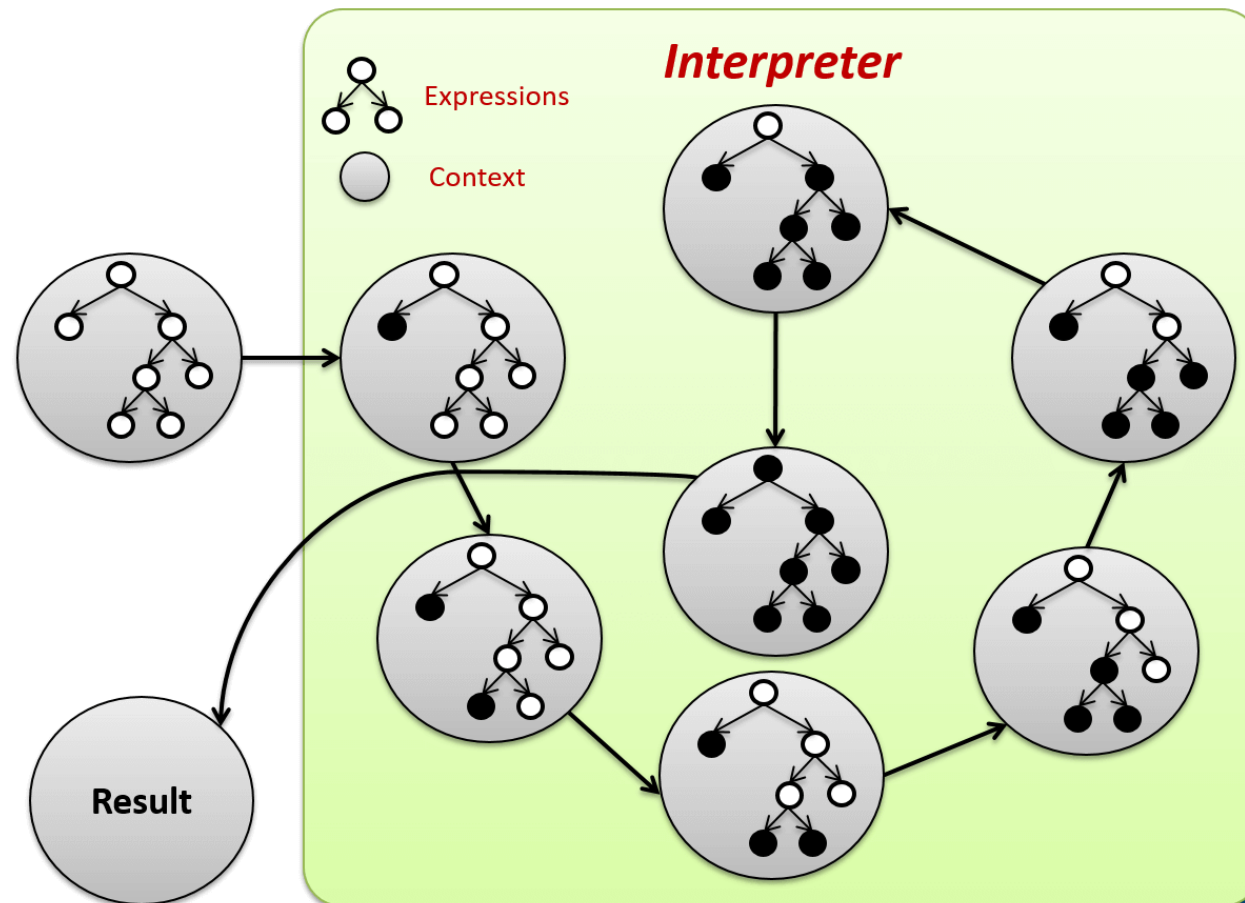




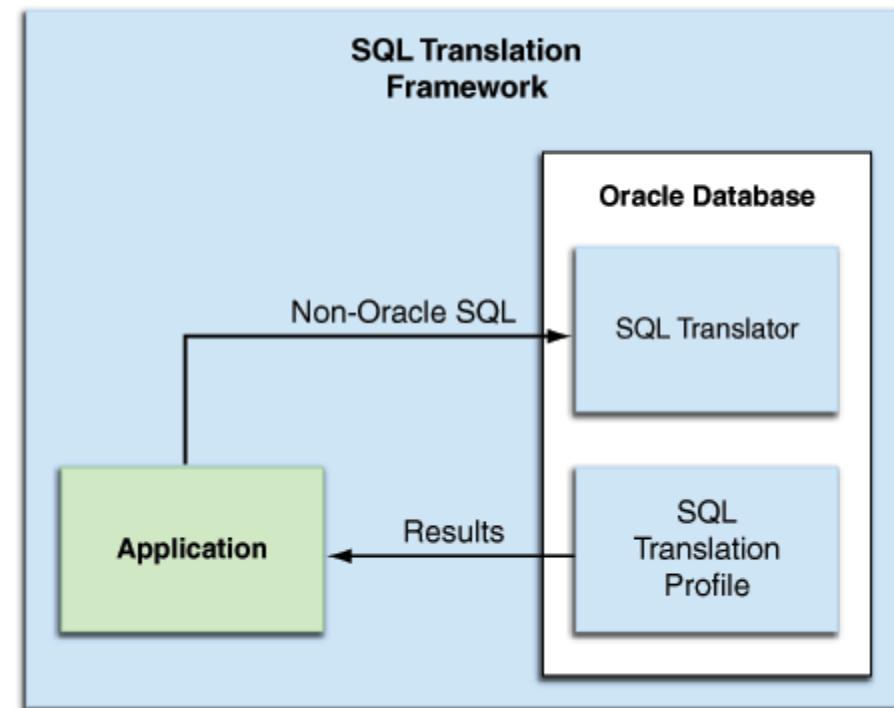
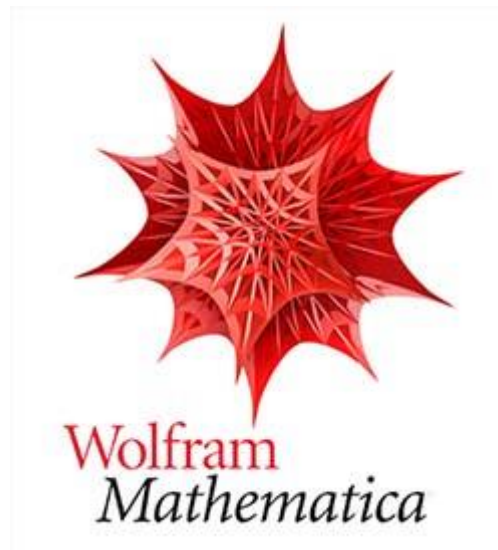
## Patrones diseño – Interpreter

### Ejemplo:

Mediante la implementación del patrón de diseño Interpreter construiremos una aplicación que interprete comandos SQL para realizar consultas sobre un Archivo de Excel, como si este se tratara de una base de datos relacional, en donde cada Hoja será vista como una tabla y las columnas de la hoja como columnas de la tabla. Para esto construiremos nuestra propia estructura de clases para representar el Lenguaje SQL, para finalmente ser interpretadas y nos arroje un resultado.



# Patrones diseño – Interpreter



**El patrón de diseño Interpreter se utiliza ampliamente en aplicaciones y sistemas donde es necesario interpretar y ejecutar expresiones o comandos en un lenguaje específico. Aunque las implementaciones exactas pueden variar según el contexto y la necesidad específica del proyecto, aquí te menciono algunos ejemplos de cómo y dónde se utiliza este patrón en empresas reconocidas:**

**Oracle Database (SQL Interpreter):**

En el contexto de bases de datos relacionales como Oracle, el lenguaje SQL (Structured Query Language) se interpreta utilizando un motor de consultas que sigue principios similares al patrón Interpreter. Esto permite ejecutar consultas complejas y manipular datos de manera eficiente.

**Sistemas de Inteligencia Artificial y Procesamiento de Lenguaje Natural:**

En empresas como Google (con Google AI) o Microsoft (con Azure Cognitive Services), se implementan interpreters para procesar y ejecutar modelos de inteligencia artificial y procesamiento de lenguaje natural. Estos sistemas interpretan y ejecutan comandos o consultas que permiten interactuar con capacidades avanzadas de IA.

**Software de Matemática Simbólica y Cálculo Automático:**

En aplicaciones como Wolfram Mathematica, que se utiliza para cálculos matemáticos avanzados y simbólicos, el patrón Interpreter puede utilizarse para interpretar y evaluar expresiones matemáticas complejas definidas por los usuarios.







**UNIVERSIDAD**  
Popular del cesar