

# UNIVERSIDAD POLITECNICA SALESIANA

## Investigación y manual de PROLOG

Wilson Conce

wconcel@est.ups.edu.ec

**Resumen:** Este documento está desarrollado para guiar el uso de PROLOG (programmation Logique) como lenguaje de programación, que es uno de los lenguajes que fue desarrollado a los inicios del año 1972 en Francia. Lastimosamente prolog es usado mayormente dentro del área de inteligencia artificial que a continuación estaremos brindando el uso de este lenguaje, basándonos de algunos ejemplos prácticos, a más de una instalación y configuración.

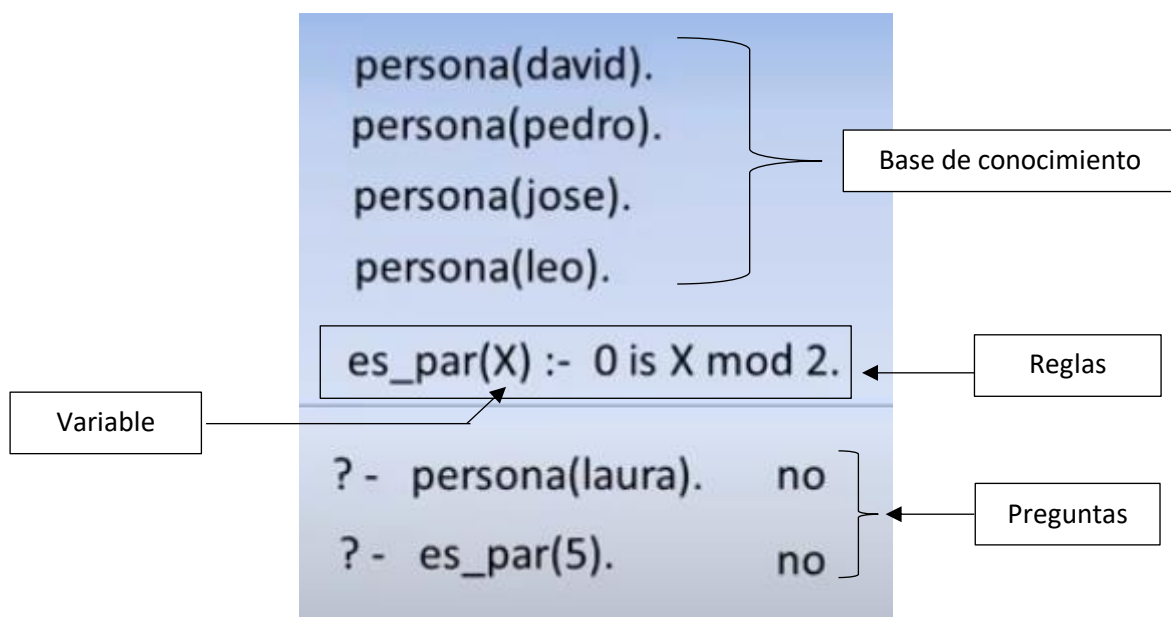
### INTRODUCCION

Prolog es una lenguaje de programación lógico que posee en la actualidad varios compiladores, como lo son: SWI Prolog, GNU Prolog, Visual Prolog, en su actualidad la más popular si así decirlo, es SWI Prolog, esta cuenta con una librería extensa con respuesta rápida que permite generar ejecutables nativos, prolog usa un paradigma descriptivo o declarativo.

### ¿Qué es Prolog?

Es un lenguaje que está orientado a la programación lógica e interpretada, en solución a problemas mediante cálculos predicados, que se encuentran basadas en preguntas a la DB (base de datos), y pruebas matemáticas. Estos resultados se dan a través de búsqueda lógica de predicados, en donde el conocimiento es representado mediante reglas.

- **Construir una base de conocimientos**
  - o Una base de conocimiento de forma con hechos, estos hechos siempre serán verdaderos.
- **Variables y constantes**
  - o Las variables siempre serán escritas en mayúsculas, en cado de escribir con minúsculas estas ya no serían contadas como variables si no sería unas constantes.



- **Reglas:** son usadas para significar que un hecho depende de uno o más.

## Programando en Prolog

Existen dos cuerpos principales, declaración de los hechos y las preguntas, con esta información, el usuario puede hacer preguntas respecto a las especificaciones.

Prolog busca en la base de conocimientos las preguntas para ver si coinciden con algún hecho que está en la base de conocimiento, dando como respuesta un sí o no.

## Objetos de datos

### Tipos de datos

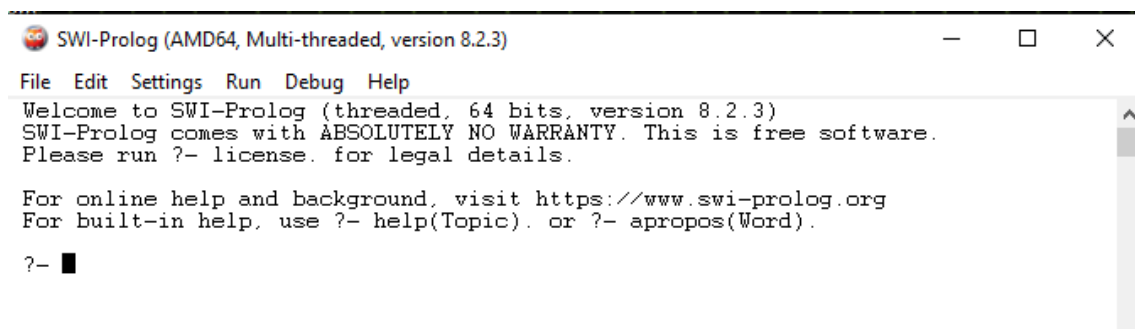
- Enteros
- Reales
- Caracteres
- Átomos: constantes y variables de cadena.
- Listas, representadas entre [ ].
- Tipos definidos por el usuario. Las reglas para definir relaciones pueden actuar como tipos de usuario

## Ventajas y desventajas

V	D
Cálculo procedural	Resolución automática no siempre eficiente
Fácil de programar	Es incapaz de reconocer problemas inaplicables
Modularidad	Motor de inferencia es limitado
Búsqueda independiente	

## Instalación de Prolog Windows

La instalación en este caso en Windows es súper fácil y básica, con dar siguiente y elegir una ruta de instalación y listo, tenemos nuestro sistema instalado.



## Instalación Prolog Linux (Linux Mint 19.3)

Dentro de Linux lo tenemos digitar las siguientes líneas:

- % sudo apt-get install software-properties-common
- % sudo apt-add-repository ppa:swi-prolog/stable
- % sudo apt-get update
- % sudo apt-get install swi-prolog

Después creamos un archivo con extensión .pl, en mi caso lo he nombrado ejemplo\_1.pl, y para abrir nos ubicamos en la ruta donde creamos dicho archivo y ejecutamos la siguiente línea:

- `swipl -s ejemplo_1.pl` (ejemplo\_1.pl es el nombre de mi archivo creado)

```
(base) wilson@wilson-vm:~/Escritorio/Prolog$ swipl -s ejemplo_1.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- 
```

### Ejemplos desarrollados en Prolog

Dados los siguientes hechos:

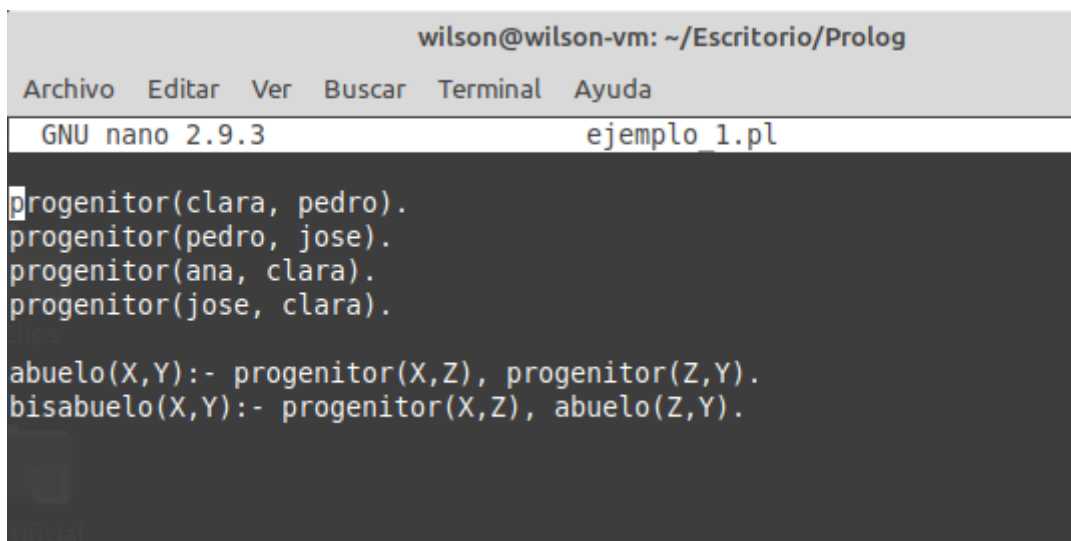
- Clara es madre de Pedro
- Pedro es padre de José
- Ana es madre de Clara
- José es padre de Clara

¿Es clara abuela de José?

¿Es Pedro abuelo de Clara?

¿Es Ana bisabuela de alguien?

¿De quién?



```
wilson@wilson-vm: ~/Escritorio/Prolog
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 2.9.3 ejemplo_1.pl

progenitor(clara, pedro).
progenitor(pedro, jose).
progenitor(ana, clara).
progenitor(jose, clara).

abuelo(X,Y):- progenitor(X,Z), progenitor(Z,Y).
bisabuelo(X,Y):- progenitor(X,Z), abuelo(Z,Y).
```

*Ilustración 1 Base de conocimientos*

```
wilson@wilson-vm: ~/Escritorio/Prolog
Archivo  Editar  Ver  Buscar  Terminal  Ayuda

(base) wilson@wilson-vm:~/Escritorio/Prolog$ swipl -f ejemplo_1.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- abuelo(clara, jose).
true.

?- abuelo(pedro, clara).
true.

?- bisabuelo(ana, jose).
true.

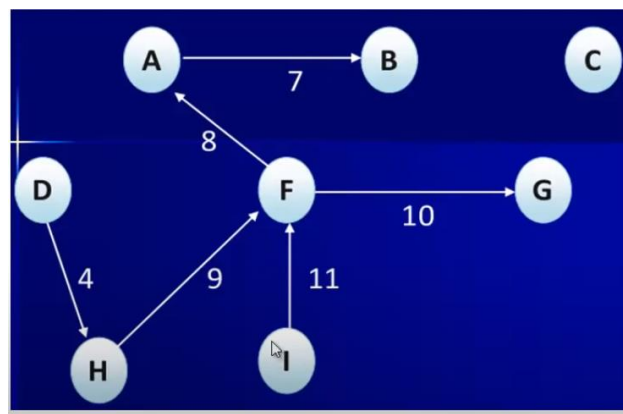
?- bisabuelo(ana, X).
X = jose.

?- 
```

*Ilustración 2 resultados*

### Ejemplo 2 desarrollado con Prolog

- ¿Existe una conexión entre A e I?
- Con que nodos está conectado F y Cual es el costo de cada conexión.
- Construir una regla para determinar si un nodo tiene aristas.
- Construir una regla para determinar cuál es el costo de ir de un nodo X a un Z pasando por Y



```
wilson@wilson-vm: ~/Escritorio/Prolog
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 2.9.3 ejemplo_2.pl

conexion(d,h,4).
conexion(h,f,9).
conexion(i,f,11).
conexion(f,g,10).
conexion(f,a,8).
conexion(a,b,7).
conexion(d,i,2).

tieneArista(X) :- conexion(X,_,_).
costoLlegada(Inicio,Destino,Inter,Costo):-
    conexion(Inicio,Int,C1), conexion(Int,Destino,C2), Costo is C1+C2.
```

```
wilson@wilson-vm: ~/Escritorio/Prolog
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- conexion(a,i,_).
false.

?- conexion(f,X,C).
X = g,
C = 10 ;
X = a,
C = 8.

?- tieneArista(f).
true ;
true.

?- costoLlegada(f,b,a,C).
C = 15.

?-
```

**Conclusiones:** PROLOG es un lenguaje que nos prepara para la interacción entre máquina y humano, ya que por medio de este podemos ingresar datos a la base de conocimiento; con hechos y relaciones además de aplicar reglas, para empezar a consultarle a la máquina y nos devuelva una respuesta o resultado

### Referencias:

Artificial intelligence through Prolog. (1988). *Choice Reviews Online*.  
<https://doi.org/10.5860/choice.26-0350>

Boizumault, P., Djamboulain, A. M., & Fattouh, J. (2015). Presentation Outline. In *The Implementation of Prolog*. <https://doi.org/10.1515/9781400863440.190>

Bowen, K. A. (1979). Prolog. *Proceedings of the 1979 Annual Conference, ACM 1979*.

<https://doi.org/10.1145/800177.810020>

Clocksin, W. F., & Mellish, C. S. (1994). Programming in Prolog. In *Programming in Prolog*.  
<https://doi.org/10.1007/978-3-642-97596-7>

PROLOG Programming for Artificial Intelligence. (1987). *Information and Software Technology*. [https://doi.org/10.1016/0950-5849\(87\)90035-8](https://doi.org/10.1016/0950-5849(87)90035-8)

Wielemaker, J., Schrijvers, T., Triska, M., & Lager, T. (2012). SWI-Prolog. *Theory and Practice of Logic Programming*. <https://doi.org/10.1017/S1471068411000494>

(“Artificial Intelligence through Prolog,” 1988; “PROLOG Programming for Artificial Intelligence,” 1987; Boizumault et al., 2015; Bowen, 1979; Clocksin & Mellish, 1994; Wielemaker et al., 2012)