



# PARADIGMAS DE PROGRAMACIÓN

Licenciatura en Informática  
Programador Universitario



## UNIDAD I

### PARADIGMAS DE PROGRAMACIÓN

#### PROLOG

#### Paradigma Lógico

## Paradigma Lógico

### Lenguajes Lógicos

constan de

- **Predicados**
- **Cláusulas**
- **Mecanismos de búsqueda y reconocimiento**
- **Estructuras de datos**, tales como átomos, variables, listas, ...

# Lenguaje PROLOG

**a) Predicados (hechos):** se utilizan para expresar propiedades de los objetos

`predicado(argumento).`

Los objetos deben ir entre ( ) separados por ,

Debe comenzar con minúscula

`predicado(arg1,arg2,..., argn).`

```
programador(luis).    % Luis es programador
hijo(luis,juan,ana).  % Luis es hijo de Juan y Ana
casado(juan,ana).     % Juan está casado con Ana
almuerza(jorge, X).  % Jorge almuerza cualquier cosa
```

3

# Lenguaje PROLOG

**b) Cláusulas:** Definen reglas lógicas que permiten inferir otros conceptos al aplicarlas a una Base de Conocimiento.

`<parte-izq-regla> :- <parte-der-regla>.`

SI <parte-der-regla> ENTONCES <parte-izq-regla>

```
hombre(X) :- varon(X), adulto(X).
progenitor(X,Y) :- padre(X,Y); madre(X,Y).
vuela(X):- pajaro(X), \+ pinguino(X), \+ avestruz(X)
```

Conjunción (y)

,

Disyunción (o)

;

Negación

\+

4

# Lenguaje PROLOG

```
hombre(X) :- varon(X), adulto(X).  
progenitor(X,Y) :- padre(X,Y); madre(X,Y).  
vuela(X):- pajaro(X), \+ pinguino(X), \+ avestruz(X)
```

¿Cómo escribimos la relación abuelo?

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

```
 $\forall X, Z \exists Y :$   
(X es hombre  $\wedge$  X es progenitor de Y  $\wedge$  Y es progenitor de Z)  $\Rightarrow$  X es abuelo de Z
```

5

# Lenguaje PROLOG

**c) Mecanismos de Búsqueda:** la búsqueda en la Base de Conocimiento se efectúa de arriba a abajo, usando el algoritmo de **DFS** (Deep First Search) con aplicación de **Backtracking**. Las reglas se resuelven de izquierda a derecha.

```
numero(0):-write(0).  
numero(N):-N1 is N-1, numero(N1), write(N).
```

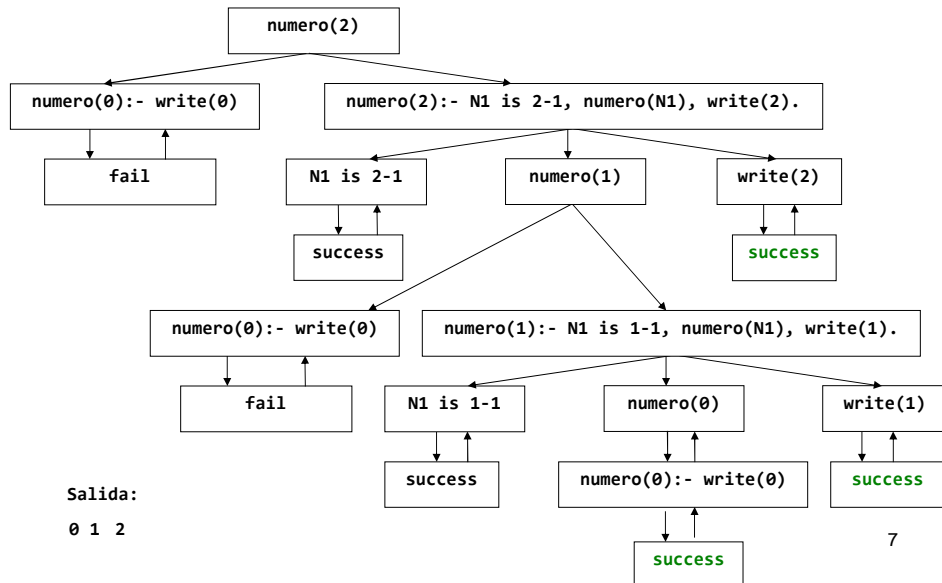
## Consulta

```
?.-numero(2).  
012
```

6

## Mecanismos de Búsqueda

```
numero(0):-write(0).
numero(N):-N1 is N-1, numero(N1), write(N).
```



## Lenguaje PROLOG

Base del Conocimiento

```
empleado(juan).
empleado(luis).
secretaria(rosa).
jefe(pedro).
jefe(julio).
supervisor(X,Y) :- jefe(X),empleado(Y).
supervisor(X,Y) :- jefe(X),secretaria(Y).
```

¿Cuál será la respuesta de Prolog a la pregunta ?

? - supervisor(X,Y).

La respuesta será

```
X=pedro, Y=juan
X=pedro, Y=luís
X=julio, Y=juan
X=julio, Y=luís
X=pedro, Y=rosa
X=julio, Y=rosa
```

Para suprimir el mecanismo de Backtracking rescribimos las reglas con el operador ! (cut)

```
supervisor(X,Y) :- jefe(X), empleado(Y), !.
supervisor(X,Y) :- jefe(X), secretaria(Y), !.
```

La respuesta será

```
X=pedro, Y=juan
```

9

## Lenguaje PROLOG

### d) Estructuras de datos:

Constantes: Átomos, literales ó Valores enteros

```
luís, silla, manzana
'hola'
20, 50, 123
```

### Variables

```
X, Y, N1, Hola
```

### Listas

```
[a, b, c, d]
[rojo, verde, azul]
```

10

# Lenguaje PROLOG

Lista → `[cabeza | resto]`

```
vocales([a, e, i, o, u]).
```

```
? - vocales(X).  
X = [a, e, i, o, u]
```

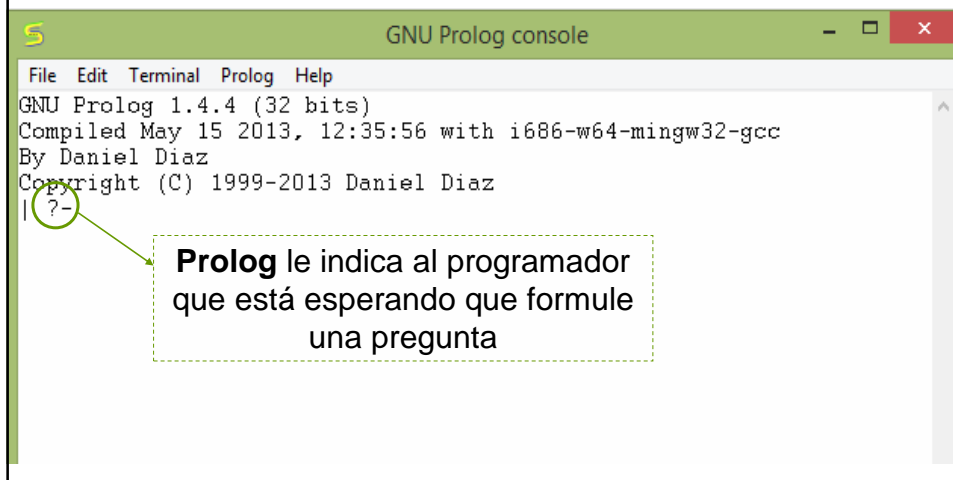
```
? - vocales([H | T]).  
H = a  
T = [e, i, o, u]
```

```
? - vocales ([X | _ ]).  
X = a
```

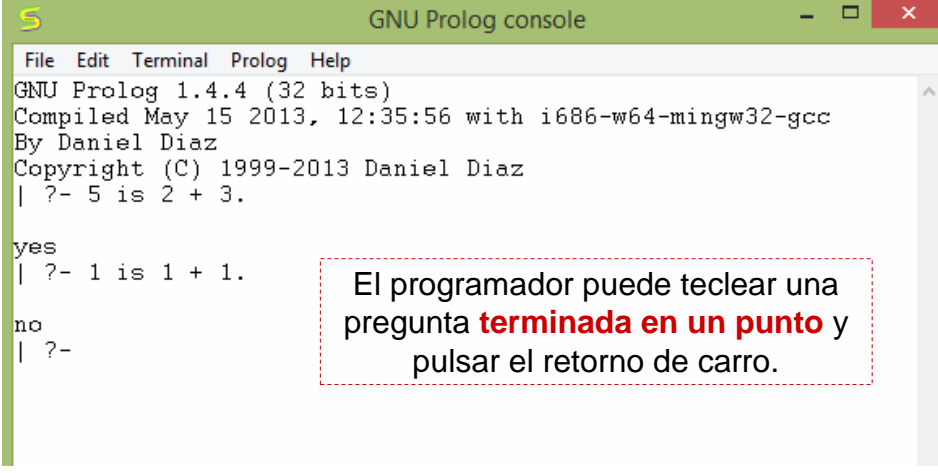
11

# Lenguaje PROLOG

**Prolog** es un lenguaje conversacional



# Lenguaje PROLOG



The screenshot shows the GNU Prolog console window. The title bar is green and says "GNU Prolog console". The menu bar includes File, Edit, Terminal, Prolog, and Help. The console text is as follows:

```
GNU Prolog 1.4.4 (32 bits)
Compiled May 15 2013, 12:35:56 with i686-w64-mingw32-gcc
By Daniel Diaz
Copyright (C) 1999-2013 Daniel Diaz
| ?- 5 is 2 + 3.

yes
| ?- 1 is 1 + 1.

no
| ?-
```

A red dashed box highlights the text: "El programador puede teclear una pregunta **terminada en un punto** y pulsar el retorno de carro."

13

# Lenguaje PROLOG

```
?- 5 is 2+.3.
uncaught exception:
  error(syntax_error('user_input:15
(char:7) . or operator expected after
expression'),read_term/3)
```

Las preguntas son términos Prolog y deben ajustarse a una **sintaxis formal concreta**. Si la pregunta en cuestión no es un término correcto, habremos cometido un error sintáctico.

Prolog detecta tales errores y nos avisará que no entiende la pregunta formulada.

14

# Lenguaje PROLOG

En la actualidad PROLOG cuenta con estrategias y características que **NO** se corresponden con el paradigma declarativo.

