



UNIVERSIDAD DE CÓRDOBA

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE  
INFORMÁTICA Y ANÁLISIS NUMÉRICO

# PROGRAMACIÓN DECLARATIVA

INGENIERÍA INFORMÁTICA

CUARTO CURSO

PRIMER CUATRIMESTRE



Tema 9.- Elementos básicos de Prolog



Primera  
parte:  
Scheme

Tema 1.- Introducción al lenguaje Scheme

Tema 2.- Expresiones y funciones

Tema 3.- Predicados y sentencias condicionales

Tema 4.- Iteración y recursión

Tema 5.- Tipos de datos compuestos

Tema 6.- Abstracción de datos

Tema 7.- Lectura y escritura

Segunda  
parte: Prolog

Tema 8.- Introducción al lenguaje Prolog

Tema 9.- Elementos básicos de Prolog

Tema 10.- Listas

Tema 11.- Reevaluación y el "corte"

Tema 12.- Entrada y salida

Segunda parte: **Prolog**

**Tema 8.-** Introducción al lenguaje Prolog

**Tema 9.-** Elementos básicos de Prolog

**Tema 10.-** Listas

**Tema 11.-** Reevaluación y el "corte"

**Tema 12.-** Entrada y salida

# Índice

1. Términos
2. Operadores aritméticos
3. Operadores relacionales
4. Definición de nuevos operadores

# Índice

1. Términos
2. Operadores aritméticos
3. Operadores relacionales
4. Definición de nuevos operadores

# 1. Término

- **Definición**
  - Los tipos de datos en Prolog se denominan "términos".
- **Tipos**
  - Números
  - Átomos
  - Variables
  - Estructuras
  - Listas
  - Cadenas de caracteres

# 1. Término

- **Números**

- **Enteros**

- ... , -3, -2, -1, 0, 1, 2, 3 ...

- **Reales**

- ..., -3.001, ..., -2,5, ..., 0.0, ..., 1.956, ...

- **Observación**

- Es un tipo de dato constante.
- Los rangos de los valores numéricos dependen del intérprete.

# 1. Término

- **Átomos**

- **Definición**

- Son constantes que **no** tienen valores numéricos.



# 1. Término

- **Átomos**

- **Sintaxis (1/3)**

- Primera forma:

- Compuestos por letras, números o el símbolo “\_”.

- Deben comenzar por una letra **minúscula**.

- Ejemplos

- ✓ juan, ana, oro, agua

- ✓ fin\_de\_mes, dato\_1, paga\_extra

# 1. Término

- **Átomos**

- **Sintaxis (2/3)**

- Segunda forma:

- Cualquier secuencia de caracteres delimitados por comillas simples.

- Ejemplos

- ✓ 'Juan Lara Luque'

- ✓ 'Fin de mes'

- ✓ '1Ba'

# 1. Término

- **Átomos**

- **Sintaxis (3/3)**

- Tercera forma:

- Cualquier secuencia de uno o más de los siguientes caracteres especiales:

+ - \* / > < = & # @ :

- Ejemplos

- ✓ +++

- ✓ +-

- ✓ >

# 1. Término

- **Átomos**

- **No** son átomos

- Uno, Marta, Vaso:

- Comienzan por una letra mayúscula.

- **23ab**:

- Comienza por un dígito.

- **\_hombre**:

- Comienza por “\_”

# 1. Términos

- **Variables**

- **Definición**

- Tipo de dato que puede modificar su valor.

- **Sintaxis**

- Su nombre está compuesto por letras, números o el símbolo de “\_”.
- Su nombre debe comenzar por una letra **mayúscula** o el símbolo “\_”.

# 1. Términos

- **Variables**

- **Ejemplos**

- X, Y, N, Respuesta, Cola, Cabeza
- Pago\_anual, Clave\_secreta, ...
- Dato\_1, Dato\_2, ...
- \_dato, \_control, ...

# 1. Términos

- **Variables**

- **Variable anónima: símbolo “\_”**

- Cada aparición de la variable anónima es **independiente** de las demás.
- La variable anónima se utiliza para
  - indicar que **existe un valor que hace verdadero un predicado,**
  - pero **no** interesa saber qué valor es.

# 1. Términos

- **Variables**

- **Variable anónima: símbolo “\_”**

- Ejemplo (1/3)

*tiene (juan, bicicleta).*

*tiene (juan, coche).*

*tiene (ana, motocicleta).*

*tiene (ana, coche).*

*tiene (pedro, barco).*



# 1. Términos

- **Variables**

- **Variable anónima: símbolo “\_”**

- Ejemplo (2/3)

- ¿**Qué** cosas tiene Juan?

?- *tiene(juan, X).*

*X = bicicleta ;*

*X = coche.*

- ¿Tiene Juan **alguna** cosa?

?- *tiene(juan, \_).*

*true*

# 1. Términos

- Variables

- Variable anónima: símbolo “\_”

- Ejemplo (3/3)

- ¿Qué cosas tiene en común Juan y Ana?  
*? tiene (juan, X), tiene (ana, X).*

*X = coche*

- ¿Tienen Juan y Pedro alguna cosa?

*?- tiene(juan,\_), tiene(pedro,\_).*

*true*

# 1. Términos

- **Estructuras**

- **Definición**

- Es un tipo de dato **compuesto** por uno o varios términos:
  - números, átomos, estructuras, listas o cadenas.
- Permiten organizar la información.

- **Sintaxis**

- *nombre\_estructura(atributo<sub>1</sub>, ..., atributo<sub>N</sub>)*

# 1. Términos

- Estructuras

- Ejemplos

- Formato de la estructura **libro**

*libro(título, autor, editorial, año)*

- Uso de la estructura "libro" con el predicado "tiene".

*tiene('Juan Lora', libro('Ana Karenina', 'Tolstoi', 'Luna', 2010)).*

*tiene('Juan Lora', libro('El Quijote', 'Cervantes', 'Fe', 2007)).*



**Predicado**



**Estructura**

# 1. Términos

- Estructuras

- Ejemplos

- **Títulos de los libros que tiene Juan Lora**

?- *tiene*('Juan Lora', *libro*(T, \_, \_,\_)).

T = 'Ana Karenina' ;

T = 'El Quijote'

Se teclea ";"

# 1. Términos

- Estructuras

- Observación

- No se deben confundir las estructuras con los predicados.

- ❑ Un **predicado** devuelve un valor de verdadero o falso.

- ❑ Una **estructura** no devuelve ningún valor, sino que solamente agrupa datos relacionados.

# 1. Términos

- Estructuras

- Observación

- Errores

?- *libro*('Ana Karenina', A, \_, \_).

*Error.*

# 1. Términos

- **Listas**

- **Definición**

- Una lista es un tipo especial de estructura compuesta por una secuencia ordenada de cero, uno o más elementos.
- Cada elemento puede ser cualquier término, incluso otra listas

- **Sintaxis**

*[elemento<sub>1</sub>, ..., elemento<sub>N</sub>]*

- Forma equivalente

*.(elemento<sub>1</sub>, .(elemento<sub>2</sub>, .(..., elemento<sub>N</sub>, **[]**)...)*



# 1. Términos

- Listas

- Ejemplos

$[]$

$[a]$

$[a, b]$

$[a, b, c]$

$.(a, [])$

$.(a, .(b, []))$

$.(a, .(b, .(c, [])))$

# 1. Términos

- **Listas**

- **Observación**

- Las listas serán explicadas en el tema nº 10.

# 1. Términos

- Cadenas de caracteres

- **Definición**

- Una cadena de caracteres es una secuencia de caracteres delimitados por comillas.

- **Observación**

- Prolog considera que una **cadena** de caracteres es un **caso especial de lista** compuesta por códigos ASCII.

- **Ejemplo**

*"Hola"*

[72, 111, 108, 97]

## Índice

1. Términos
2. Operadores aritméticos
3. Operadores relacionales
4. Definición de nuevos operadores

## 2. Operadores aritméticos

- Introducción
- Operadores prefijos
- Operadores infijos
- Ejemplos de predicados con operadores aritméticos

## 2. Operadores aritméticos

- Introducción
- Operadores prefijos
- Operadores infijos
- Ejemplos de predicados con operadores aritméticos

## 2. Operadores aritméticos

- **Introducción**

- Una expresión aritmética se puede considerar como una estructura.

- Ejemplo

- $2 * 10 + 1$  es equivalente a  $+(*(2, 10), 1)$

## 2. Operadores aritméticos

- **Introducción**

- Evaluación de expresiones

- Prolog utiliza la palabra clave "*is*" para evaluar una expresión aritmética.

- Ejemplo

?- *X is* 2 \* 10 + 1.

*X =* 21

?- *X is* +( \*(2,3), 1).

*X =* 7.



## 2. Operadores aritméticos

- **Introducción**

- **Observación**

- Si una variable tiene un valor entonces **no** se le puede asignar un valor nuevo.

- Ejemplo

?- *X is 1, A is 2 \* X, X is A + 1.*

*false.*

?- *X is 1, A is 2 \* X, Y is A + 1.*

*X = 1,*

*A = 2,*

*Y = 3.*

## 2. Operadores aritméticos

- Introducción
- Operadores prefijos
- Operadores infijos
- Ejemplos de predicados con operadores aritméticos

## 2. Operadores aritméticos

- Operadores prefijos

- +, -

- Ejemplos

?-  $X$  *is* + 9.

$X = 9$

?-  $X$  *is* -12.

$X = -12$

## 2. Operadores aritméticos

- Introducción
- Operadores prefijos
- Operadores infijos
- Ejemplos de predicados con operadores aritméticos

## 2. Operadores aritméticos

- Operadores infijos

Operador	Significado	Precedencia	Asociatividad	Ejemplo
$\wedge$	Potencia	200	Derecha	?- $X$ is $2^3^2$ . $X = 512$ .  ?- $X$ is $(2^3)^2$ . $X = 64$
$**$	Potencia	200	No	?- $X$ is $2**3$ . $X = 8$ .  ?- $X$ is $2**3** 2$ . <b>Error</b>  ?- $X$ is $(2**3)**2$ . $X = 64$

## 2. Operadores aritméticos

- Operadores infijos

Operador	Significado	Precedencia	Asociatividad	Ejemplo
*	Producto	400	Izquierda	?- $X$ is $2*3*4$ . $X = 24$ .
/	División	400	Izquierda	?- $X$ is $60 / 2 / 3$ . $X = 10$
<b><i>rdiv</i></b>	Cociente de la división entera	400	Izquierda	?- $X$ is $12$ <i>rdiv</i> $4$ . $X = 3$ .
<b><i>//</i></b>	Cociente de la división entera	400	Izquierda	?- $X$ is $15$ <i>//</i> $4$ . $X = 3$ .
<b><i>mod</i></b>	Resto de la división entera	400	Izquierda	?- $X$ is $12$ <i>mod</i> $3$ . $X = 0$ .
<b><i>rem</i></b>	Resto de la división entera	400	Izquierda	?- $X$ is $12$ <i>rem</i> $3$ . $X = 0$ .

## 2. Operadores aritméticos

- Operadores infijos

Operador	Significado	Precedencia	Asociatividad	Ejemplo
$+$	Suma	500	Izquierda	?- $X$ is $2+3+4$ . $X = 9$
$-$	Resta	500	Izquierda	?- $X$ is $8 - 2 - 3$ . $X = 3$

## 2. Operadores aritméticos

- Operadores infijos

- Precedencia

- Un operador tiene **mayor** precedencia si su valor numérico de precedencia es menor.
- Orden de precedencia
  - Mayor: ()
  - $\wedge$ , \*\*
  - \*, /, *rdiv*, //, *mod*, *rem*
  - Menor: +, -



## 2. Operadores aritméticos

- **Operadores infijos**
  - Ejemplos

?- *X is 2 \* 3 + 4 rdiv 2+ 5 mod 2.*

*X = 9.*

?- *X is 2^3 - 12 / 4 \* 3.*

*X = -1.*

## 2. Operadores aritméticos

- Introducción
- Operadores prefijos
- Operadores infijos
- Ejemplos de predicados con operadores aritméticos

## 2. Operadores aritméticos

- Ejemplos de predicados con operadores aritméticos

- Cuadrado

*cuadrado(X,R):- R is X \* X.*

*?- cuadrado(2,R).*

*R = 4.*

*?- cuadrado(N,4).*

**ERROR:** *is/2: Arguments are not sufficiently instantiated*

## 2. Operadores aritméticos

- **Ejemplos de predicados con operadores aritméticos**

- Media

*media(A,B,M):-*

*M is (A+B) / 2.*

?- *media(2,3,M).*

*M = 2.5*

## 2. Operadores aritméticos

- **Ejemplos de predicados con operadores aritméticos**

- Factorial

*factorial(0, 1).*

*factorial(N, R):-*

*N1 is N - 1,*

*factorial(N1, R1),*

*R is N \* R1.*

*?- factorial (3, R).*

*R = 6.*

## 2. Operadores aritméticos

- **Ejemplos de predicados con operadores aritméticos**

- Fibonacci

*fibonacci(0, 1).*

*fibonacci(1, 1).*

*fibonacci(N, R):-*

*N1 is N-1,*

*N2 is N-2,*

*fibonacci(N1, R1),*

*fibonacci(N2, R2),*

*R is R1 + R2.*

*?- fibonacci (5, R).*

*R = 8*

## 2. Operadores aritméticos

- **Ejemplos de predicados con operadores aritméticos**
  - Máximo común divisor (pseudocódigo)

$mcd(a, b)$

**si**  $b = 0$

**entonces**  $a$

**si no**  $mcd(b, a \bmod b)$

**fin\_si**

	<b>Paso</b>		
	1	2	3
$a$	18	12	6
$b$	12	6	0
$r$	6	0	

## 2. Operadores aritméticos

- **Ejemplos de predicados con operadores aritméticos**

- Máximo común divisor

$mcd(X, 0, X).$

$mcd(X, Y, M):-$

$X1 \text{ is } X \text{ mod } Y,$

$mcd(Y, X1, M).$

?-  $mcd(12, 18, R).$

$R= 6.$

?-  $mcd(18, 12, R).$

$R= 6.$



## 2. Operadores aritméticos

- **Ejemplos de predicados con operadores aritméticos**

- Densidad

*poblacion('Francia', 60000000).*

*poblacion('España', 45000000).*

*area('Francia', 640000).*

*area('España', 505000).*

*densidad(Pais, D):-*

*poblacion(Pais, P),*

*area(Pais, A),*

*D is P / A.*

## 2. Operadores aritméticos

- **Ejemplos de predicados con operadores aritméticos**

- Densidad

?- *densidad('Francia',D).*

*D = 93.75.*

?- *densidad('España',D).*

*D = 89.1089.*

## Índice

1. Términos
2. Operadores aritméticos
3. Operadores relacionales
4. Definición de nuevos operadores

### 3. Operadores relacionales

- Operadores de igualdad
- Operadores de desigualdad
- Operadores de "mayor y menor que"
- Ejemplos de predicados con operadores relacionales

### 3. Operadores relacionales

- **Operadores de igualdad**
- Operadores de desigualdad
- Operadores de "mayor y menor que"
- Ejemplos de predicados con operadores relacionales

### 3. Operadores relacionales

- Operadores de igualdad

- =, ==, :=

### 3. Operadores relacionales

- Operadores de igualdad

- =

- Reglas de uso

- Dos constantes

- Una constante y una variable

- Dos variables

- Estructuras y variables

### 3. Operadores relacionales

- Operadores de igualdad

- =

- Reglas de uso

- Dos constantes son iguales si poseen el mismo valor

?- 10 = 10.

true

?- 10 = 9.

false

?- a = a.

true

?- a = b.

false

?- lugar(3,3) = lugar(3, 3).

true.

?- lugar(3,3) = lugar(3, 0).

false.

✓ Nota: *lugar* es una estructura



### 3. Operadores relacionales

- Operadores de igualdad

- =

- Reglas de uso

- Una constante y una variable

- ✓ Si la variable tiene un valor (variable "**instanciada**") entonces se comprueba si dicho valor es igual a la constante.

*?- X is 10, X = 10.*

*X = 10*

*?- X is 10, X = 9.*

*false*

### 3. Operadores relacionales

- Operadores de igualdad

- =

- Reglas de uso

- Una constante y una variable

- ✓ Si la variable **no** tiene un valor (variable “no instanciada”) entonces

- se asigna el valor de la constante a la variable

- y la igualdad se cumple

$?- X = 10.$ $X = 10$
--------------------------

$?- X = a.$ $X = a$
------------------------

$?- X = lugar(3, 3).$ $X = lugar(3, 3).$
---

### 3. Operadores relacionales

- Operadores de igualdad

- =

- Reglas de uso

- Una constante y una variable

- ✓ No importa el orden

*?- X = 10.*

*X = 10.*

*?-10 = X.*

*X = 10.*

### 3. Operadores relacionales

- Operadores de igualdad

- =

- Reglas de uso

- Dos variables

- ✓ Si las variables están "**instanciadas**", se comprueba si sus valores son iguales.
- ✓ Si **una** variable está "**instanciada**" y la **otra no**, la variable "instanciada" le asigna su valor a la otra variable.
- ✓ Si las dos variables **no** están "instanciadas" entonces las variables pasan a **compartir** "memoria".

### 3. Operadores relacionales

- Operadores de igualdad

- =

- Reglas de uso

- Dos variables

- ✓ Si las variables están “instanciadas”, se comprueba si sus valores son iguales.

*?- X is 2, Y is 2, X = Y.*

*X = 2,*

*Y = 2.*

*?- X is 2, Y is 3, X = Y.*

*false.*

### 3. Operadores relacionales

- Operadores de igualdad

- =

- Reglas de uso

- Dos variables

- ✓ Si una variable está “instanciada” y la otra no, la variable “instanciada” le asigna su valor a la otra variable.

*?- X is 2, X = Y.*

*X = 2,*

*Y = 2.*

### 3. Operadores relacionales

- Operadores de igualdad

- =

- Reglas de uso

- Dos variables

- ✓ Si las dos variables no están “instanciadas” entonces las variables pasan a **compartir** “memoria”.

?-  $X = Y$ ,  $X$  is 2.

$X = 2$ ,

$Y = 2$ .

### 3. Operadores relacionales

- Operadores de igualdad

- =

- Reglas de uso

- Estructuras y variables

- ✓ Se comprueba la igualdad **atributo** a **atributo**.

- ✓ Se tiene en cuenta si las variables están o no "**instanciadas**".



### 3. Operadores relacionales

- Operadores de igualdad

- =

- Reglas de uso

- Estructuras y variables

*?- lugar(X, Y) = lugar(2, 3).*

*X = 2,*

*Y = 3.*

*?- lugar(X, 3) = lugar(2, Y).*

*X = 2,*

*Y = 3.*

### 3. Operadores relacionales

- Operadores de igualdad

- =

- Reglas de uso

- Listas y variables

- ✓ Se comprueba la igualdad **elemento** a **elemento**.
        - ✓ Se tiene en cuenta si las variables están o no "**instanciadas**".

### 3. Operadores relacionales

- Operadores de igualdad

- =

- Reglas de uso

- Listas y variables

?-  $[X, Y] = [a, b]$ .

$X = a,$

$Y = b.$

?-  $[a, Y] = [X, b]$ .

$Y = b,$

$X = a.$

### 3. Operadores relacionales

- Operadores de igualdad

- ==

- Reglas de uso

- ❑ El funcionamiento de "=" y "==" es igual sobre las constantes o si las variables están **"instanciadas"**

- ❑ Si una variable no está "instanciada" entonces el resultado siempre es **"false"**.

### 3. Operadores relacionales

- Operadores de igualdad

- ==

- Ejemplos

?- X == 1.

*false.*

?- X is 1, X == 1.

X = 1.

?- X is 1, X == Y.

*false.*

?- X == Y.

*false.*

### 3. Operadores relacionales

- Operadores de igualdad

- $:=$

- Reglas de uso

- Solamente se puede usar con expresiones **aritméticas**.

- Las variables siempre deben estar "instanciadas".

### 3. Operadores relacionales

- Operadores de igualdad

- $::=$

- Ejemplos

?-  $X ::= 1$ .

**ERROR:**  $::= /2$ : Arguments are not sufficiently instantiated

?-  $X$  is 2,  $3 * X ::= 6$ .

$X = 2$ .

?-  $X$  is 2,  $X ::= a$ .

**ERROR:**  $::= /2$ : Arithmetic:  $'a/0'$  is not a function

### 3. Operadores relacionales

- Operadores de igualdad
- Operadores de desigualdad
- Operadores de “mayor y menor que”
- Ejemplos de predicados con operadores relacionales



### 3. Operadores relacionales

- Operadores de desigualdad

- $\neq$ ,  $\neq\neq$ ,  $=\neq$

### 3. Operadores relacionales

- Operadores de desigualdad

- $\neq$

- Reglas de uso

- Es verdadero si el primer argumento no es igual al segundo argumento.

- Ejemplos

?-  $X = b, X \neq a.$

$X = b.$

?-  $X \neq a.$

*false.*

?-  $X = a, X \neq a.$

*false.*

### 3. Operadores relacionales

- Operadores de desigualdad

- `\==`

- Reglas de uso

- ❑ Es verdadero si el primer argumento no es igual al segundo argumento.
- ❑ No importa que las variables no estén instanciadas.

### 3. Operadores relacionales

- Operadores de desigualdad

- $\backslash==$

- Ejemplos

?-  $X = b, X \backslash== a.$

$X = b.$

?-  $X \backslash== a.$

*true.*

?-  $X = a, X \backslash== a.$

*false.*

### 3. Operadores relacionales

- Operadores de desigualdad

- $\neq$

- Reglas de uso

- Solamente se puede usar con expresiones **aritméticas**.

- Las variables siempre deben estar "instanciadas".

### 3. Operadores relacionales

- Operadores de desigualdad

- $\neq$

- Ejemplos

?-  $X \neq a$ .

**ERROR:**  $\neq/2$ : Arguments are not sufficiently instantiated

?-  $X$  is 1,  $X \neq a$ .

**ERROR:**  $\neq/2$ : Arithmetic: `a/0' is not a function

?-  $X$  is 1,  $X \neq 2$ .

$X = 1$ .

### 3. Operadores relacionales

- Operadores de igualdad
- Operadores de desigualdad
- Operadores de “mayor y menor que”
- Ejemplos de predicados con operadores relacionales

### 3. Operadores relacionales

- Operadores de “mayor y menor que”

- $<$ ,  $=<$ ,  $>$ ,  $>=$

- Reglas de uso

- ❑ Solamente se puede usar con expresiones **aritméticas**.

- ❑ Las variables siempre deben estar “instanciadas”.

- **Observación**

- ❑ El predicado “menor o igual que” es  $=<$  y no  $<=$



### 3. Operadores relacionales

- Operadores de “mayor y menor que”

- $<$ ,  $=<$ ,  $>$ ,  $>=$

- Ejemplos

?-  $X >= 0$ .

*ERROR: >=/2: Arguments are not sufficiently instantiated*

?-  $X \text{ is } 2, X < 10$ .

$X = 2$ .

?-  $X \text{ is } 2, 10 * X =< 7$ .

*false.*

### 3. Operadores relacionales

- Operadores de igualdad
- Operadores de desigualdad
- Operadores de "mayor y menor que"
- Ejemplos de predicados con operadores relacionales

### 3. Operadores relacionales

- Ejemplos de predicados con operadores relacionales

- Máximo

$max(X, Y, X):- X \geq Y.$

$max(X, Y, Y):- Y \geq X.$

?-  $max(2, 3, R).$

$R = 3.$

?-  $max(3, 2, R).$

$R = 3 .$

?-  $max(3, R, 3).$

**ERROR:**  $\geq/2$ : Arguments are not sufficiently instantiated

### 3. Operadores relacionales

- **Ejemplos de predicados con operadores relacionales**

- Reinado

*reinado('Carlos II', 1665, 1700).*

*reinado('Felipe V', 1700, 1724).*

*reinado('Luis I', 1724, 1724).*

*reinado('Felipe V', 1724, 1746).*

*rige(Persona, N):-*

*reinado(Persona, A, B),*

*A =< N,*

*N =< B.*

### 3. Operadores relacionales

- Ejemplos de predicados con operadores relacionales
  - Reinado

?- *rige*(R, 1724).

R = 'Felipe V' ;

R = 'Luis I' ;

R = 'Felipe V'.

Se teclea ";"

## Índice

1. Términos
2. Operadores aritméticos
3. Operadores relacionales
4. Definición de nuevos operadores

## 4. Definición de nuevos operadores

- **Sintaxis**

***op**(Precedencia, Tipo, Nombre)*

- **Significado**

- Declara un operador con el nombre, precedencia y tipos indicados.

## 4. Definición de nuevos operadores

- **Sintaxis**

*op*(**Precedencia**, **Tipo**, **Nombre**)

- **Precedencia**

- Varía desde 0 (máxima precedencia) hasta 1200 (menor precedencia).
- El valor 0 borra la declaración.



## 4. Definición de nuevos operadores

- **Sintaxis**

***op***(Precedencia, **Tipo**, Nombre)

- **Tipo**

- $xf, yf, xfx, xfy, yfx, fy, fx.$

- **f**: indica la posición del operador

- **x, y**: indican la posición de los argumentos

- ✓ **y**: debe ser interpretado en esa posición con precedencia menor o igual que la precedencia del operador

- ✓ **x**: la precedencia del operador debe ser estrictamente menor

## 4. Definición de nuevos operadores

- **Sintaxis**

*op*(Precedencia, **Tipo**, Nombre)

	<b>Tipo</b>	<b>Asociatividad</b>
<b>Prefijo</b>	$fx$	No
	$fy$	Derecha
<b>Infijo</b>	$xfx$	No
	$xfy$	Derecha
	$yfx$	Izquierda
<b>Postfijo</b>	$xf$	No
	$yf$	Izquierda

## 4. Definición de nuevos operadores

- **Sintaxis**

*op*(Precedencia, Tipo, **Nombre**)

- **Nombre:**

- Puede ser una lista de nombres del operador, en cuyo caso todos son considerados como operadores con características similares.

## 4. Definición de nuevos operadores

- **Ejemplo**

?- [factorial].

true.

? 3 ! R.

R = 6

### Fichero factorial.pl

```
factorial(0, 1).
```

```
factorial(N, R):-
```

```
    N1 is N - 1,
```

```
    factorial(N1, R1),
```

```
    R is N * R1.
```

```
!(X, R):-factorial(X, R).
```

```
?- op(150, xfy, !).
```

## 4. Definición de nuevos operadores

- **Ejemplo**

- **Observación**

- Si se desea declarar el operador en el fichero

entonces es

**obligatorio** escribir

los símbolos ?-



### Fichero factorial.pl

```
factorial(0, 1).  
factorial(N, R):-  
    N1 is N - 1,  
    factorial(N1, R1),  
    R is N * R1.  
  
!(X, R):-factorial(X, R).  
?- op(150, xfy, !).
```

## 4. Definición de nuevos operadores

- **Ejemplo**

*tiene(juan, coche).*

*tiene(juan, bici).*

*has(john, car).*

*has(john, bike).*

?- *op(150, xfy, [tiene,has]).*

*true.*

?- *juan tiene R.*

*R = coche*

*R = bici.*

?- *john has R.*

*R = car*

*R = bike.*



UNIVERSIDAD DE CÓRDOBA

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE  
INFORMÁTICA Y ANÁLISIS NUMÉRICO

# PROGRAMACIÓN DECLARATIVA

INGENIERÍA INFORMÁTICA

CUARTO CURSO

PRIMER CUATRIMESTRE



Tema 9.- Elementos básicos de Prolog

