



# **Tema 2. La Sintaxis**

---

**1. El vocabulario**

**2. Términos**

**2.1. Constantes**

**2.2. Variables**

**2.3. Estructuras**

**3. Operadores**

**4. Igualdad y Desigualdad**

**5. Aritmética**

**Ejercicio**



## 2. Términos

### 2.3. Estructuras (Functor, Functores)

- Identifica un objeto compuesto (similar a un dato estructurado):  
colección de objetos o componentes
- Los componentes pueden ser a su vez estructuras
- Permite tratar como un **único objeto** a una **colección de informaciones relacionadas**
- También se usa para nombrar de forma diferente a varios objetos de un mismo tipo



## 2.3. Estructuras

- Misma sintaxis que la de los hechos:  
*functor*(comp#1, comp#2,...).
- En general, *functor* es un nombre que designa:
  - un hecho,
  - una relación
- Internamente se guardan en forma arborescente.
- Ejemplo: fichas de libros de tu biblioteca
  - tiene(belarmino, libro).
  - tiene(belarmino, hobbit).
  - tiene(belarmino, el\_senor\_de\_los\_anillos).
  - tiene(belarmino, yo\_robot).
  - tiene(belarmino, fundacion).



## 2.3. Estructuras (cont.)

Frente a:

- `tiene(belarmino, libro(hobbit, tolkien)).`
- `tiene(belarmino, libro(el_senor_de_los_anillos, tolkien)).`
- `tiene(belarmino, libro(yo_robot, asimov)).`
- `tiene(belarmino, libro(fundacion, asimov)).`
  
- `tiene(belarmino, libro(hobbit, autor(jrr, tolkien))).`
- `tiene(belarmino, libro(yo_robot, autor(isaac, asimov))).`

**Ejercicio** (fuera de clase): Crea un fichero PROLOG, `mi_biblioteca.pl`, con una definición de tus libros, diferenciados por géneros, y después realiza consultas concretas, con variables y con variables anónimas. Comenta cuál sería la pregunta original.



### 3. Operadores

- Algunas estructuras/funtores se pueden escribir como operadores:
  - $x + y + z$  frente a  $+(z, +(x,y))$
- Escribir un operador no implica evaluarlo.
  - $?- 4 = 2 + 2.$
  - $?- 4 = +(2,2).$
- Los operadores vendrán especificados por:
  - su posición (infija, prefija o postfija),
  - por su precedencia, y
  - por la asociatividad (por la izquierda o por la derecha).



### 3. Operadores (cont.)

---

- Trataremos los operadores aritméticos más habituales:  $+$ ,  $-$ ,  $*$ , y  $/$ .
- Los operadores aritméticos serán todos infijos.
- Las precedencias están asociadas a clases de operadores y varían según el compilador (entre 1 y 255):
  - los operadores  $*$  y  $/$  tendrán mayor precedencia siempre que  $+$  y  $-$

### 3. Operadores (cont.)

- Los operadores aritméticos serán asociativos por la izquierda:
  - debe tener a su izquierda operaciones de precedencia menor o igual; a la derecha tendrá operadores de precedencia mayor
  - Ejemplos:
    - $8/2/2$  será 2 u 8.
    - asociativo por la izquierda  $8/2/2 = (8/2)/2$
- Importante: los operadores aritméticos son como cualquier otra estructura:
  - pero pueden evaluarse con **is**
    - ?- X is 2 + 2.
    - ?- X is +(2,2).
    - ?- 4 is +(2,2).

## 4. Igualdad y Desigualdad

- El predicado *igualdad*,  $=$ , está predefinido:

- $?- X = Y.$

PROLOG para satisfacer el objetivo *comprueba si ambas coinciden*.

Asociado (internamente) a satisfacer:

$X = X.$

- Reglas para decidir si  $X$  e  $Y$  son “iguales”:
  - Si  $X$  no está instanciada e  $Y$  sí, entonces son iguales y  $X$  toma como valor el término al que estuviese instanciada  $Y$ .
  - Los enteros y los átomos siempre son iguales entre sí.
  - Dos estructuras son iguales si tienen el mismo functor y el mismo número de argumentos, y cada uno de esos argumentos son iguales.





## 4. Igualdad y Desigualdad (cont.)

---

?- mesa = mesa.

?- silla = silla.

?- mesa = silla.

?- 2005 = 2004.

?- tiene(belarmino, X) = tiene(belarmino,  
libro(fundacion, asimov)).

?- tiene(belarmino, libro(\_, X)) = tiene(belarmino,  
libro(yo\_robot, autor(isaac, asimov))).

## 4. Igualdad y Desigualdad (cont.)

- Existe otro predicado predefinido: *no es igual*,  $\backslash=$ 
  - $?- X \backslash= Y.$   
Se satisface si no se cumple el objetivo  $X = Y.$
- Al igual que  $=$ , al estar predefinido **no** podemos modificarlos:  
queso = yogurt.  
mesa = silla.

Ejercicio: Comprueba si se cumplen:

```
libro(titulo, autor) = libro(yo_robot, asimov).  
libro(Titulo, Autor) = libro(yo_robot, asimov).  
'constante' = constante.  
fun1(a, b) = fun1(A, B).  
fun1(a, b, c) = fun1(A, B).
```

## 5. Aritmética

- Los programas PROLOG no están pensados para requerir una gran manipulación numérica
- Operadores comparación (predefinidos):
  - $X = Y$
  - $X \neq Y$
  - $X < Y$
  - $X > Y$
  - $X \geq Y$
  - $X \leq Y$
- Operadores aritméticos:
  - $X + Y$
  - $X - Y$
  - $X * Y$
  - $X / Y$
  - $X \bmod Y$



## 5. Aritmética (cont.)

- Los operadores aritméticos pueden usarse para hacer cálculos si se combinan con el operador **is**:
  - operador infijo,
  - el argumento a la derecha debe asociarse a un operador aritmético,
  - obliga a evaluar el argumento de la derecha y contrasta el resultado con el argumento a la izquierda:
    - $X \text{ is } 2 + 2$ .
    - $\text{resto is } X \text{ mod } Y$ .
  - No todos los operadores pueden evaluarse como expresiones aritméticas.



## Ejercicio (curso pasado)

---

- Base de conocimiento: Reyes de la casa de Austria en España
- Hecho:
  - `reina(nombre_del_rey, año_de_inicio, año_final_del_reinado).`
  - Ejemplo:
    - `reina(felipe_iii, 1598, 1621).`
- Construir la propiedad  
`rey_casa_austria(X,Y),`

Para averiguar qué rey (X) era el que estaba en el trono en el año (Y).



## Ejercicio

- Calcular un número combinatorio:

$$\binom{m}{n} = \frac{m!}{(m-n)! n!} ; \quad \text{donde } m > n$$

- Más eficiente:

$$\binom{m}{n} = \frac{m(m-1)(m-2)\dots(n+1)}{(m-n)!} ; \quad \text{donde } m > n$$