

# Programación Declarativa

## Sesión de laboratorio 6

Curso 2020/21

- Realiza los siguientes ejercicios individualmente en un mismo fichero con extensión .pl.
- Para el ejercicio 1 debes incluir tus respuestas como comentarios.
- Escribe tu nombre al comienzo del fichero como líneas comentadas.
- Sube el fichero al Campus Virtual antes de que acabe la clase.

1. Considera la siguiente especificación:

$\text{elimina}(L, X, NL) \leftrightarrow NL$  es la lista resultante de eliminar de la lista  $L$  todas las apariciones de  $X$ .

Estudia las diferencias entre las siguientes definiciones de este predicado escritas en Prolog.

(a) Usando igualdad sintáctica:

```
elimina1([ ],X,[ ]).
elimina1([X|R],Y,NR) :- Y == X, elimina1(R,Y, NR).
elimina1([X|R],Y,[X|NR]) :- Y \== X, elimina1(R,Y,NR).
```

(b) Usando unificación:

```
elimina2([ ],X,[ ]).
elimina2([X|R],Y,NR) :- Y = X, elimina2(R,Y, NR).
elimina2([X|R],Y,[X|NR]) :- Y \= X, elimina2(R,Y,NR).
```

(c) Combinando las dos anteriores:

```
elimina3([ ],X,[ ]).
elimina3([X|R],X,NR) :- elimina3(R,X,NR).
elimina3([X|R],Y,[X|NR]) :- Y \== X, elimina3(R,Y,NR).
```

Ejecuta los siguientes objetivos en cada una de las tres versiones, pidiendo todas las respuestas posibles.

```
?- elimina1([a,b,a,c],a,L). (i= 1,2,3)
```

```
?- elimina1([a,b,a,c],X,L). (i= 1,2,3)
```

Compara los resultados. ¿Qué puedes concluir?

2. Utilizando la estructura Prolog de árbol binario definida en clase, programa los siguientes predicados:

(a)  $\text{sumatree}(A, N) \leftrightarrow A$  es un árbol binario con números enteros en sus nodos y  $N$  es la suma de sus elementos.

(b)  $\text{maximo}(A, X) \leftrightarrow A$  es un árbol binario de números enteros positivos dado y  $X$  es el elemento máximo de sus nodos.  $X$  vale 0 si el árbol es vacío

3. Define un predicado Prolog `sublistas/2` para conseguir la lista de todas las sublistas de una lista dada.

4. Programa en Prolog una versión recursiva para resolver el problema de las torres de Hanoi. Para ello, define un predicado `hanoi(N, A, B, C, M)` donde  $N$  es el número de fichas que hay que mover de la torre inicial,  $A$  es el nombre de la torre inicial,  $B$  es el nombre de la torre final,  $C$  es el nombre de la torre auxiliar y  $M$  es la secuencia de movimientos de una torre a otra hasta conseguir la traslación de todas las fichas. Por ejemplo, para trasladar dos fichas de las torres con nombres, `ini`, `fin`, `aux`, respectivamente, la lista de movimientos sería:

```
[ini, aux, ini, fin, aux, fin].
```

Modifica el programa anterior para que la secuencia de movimientos se exprese mediante una expresión que usa flechas entre torres para indicar un movimiento. En el ejemplo anterior, la secuencia sería:

```
ini-->aux yy ini-->fin yy aux-->fin.
```

Define para ello dos operadores, `yy` y `-->`, adecuadamente.