

## Taller de Prolog 8

### Listas y Recursividad

#### Iteración sobre listas

Dada una lista, interesa recorrer sus elementos y tomar decisiones de acuerdo con el valor de cada elemento. Por ejemplo el siguiente procedimiento acepta una lista de números enteros e imprime aquellos elementos que son divisibles por tres:

```
divisiblesPorTres/1 : Imprime los divisibles por 3 en una lista de números.
divisiblesPorTres([]) :- !.
divisiblesPorTres([H|T]) :- H mod 3 =:= 0, !,
                             write(H), tab(2),
                             divisiblesPorTres(T).
divisiblesPorTres([_|T]) :- divisiblesPorTres(T).
```

#### Creación de listas

En muchas situaciones prácticas interesa construir procedimientos que retornen una lista creada de acuerdo con unos requerimientos. En estos casos la idea general es que el caso base construye la lista inicial (e.g. la lista vacía) y los casos recursivos agregan los elementos adicionales utilizando la operación de concatenar un elemento a la cabeza de la lista (operación '|').

Por ejemplo, en la práctica anterior se definió el procedimiento `factorial/2` para obtener el factorial de N. Se puede hacer un ciclo anidado para obtener el factorial de todos los naturales hasta M así:

```
% listaFactoriales/3 : Construir la lista de números factoriales hasta N!
% arg1 : Contador ascendente 0..N
% arg2 : N, mayor natural a considerar
% arg3 : Lista de factoriales
listaFactoriales(I,N,[]) :- I>N, !.
listaFactoriales(I,N,[Fi|Lj]) :- factorial(I,Fi), J is I+1, listaFactoriales(J,N,Lj).
% factorial/2: Calcula el factorial de N
% arg1 : N
% arg2 : resultado N!
factorial(0,1) :- !.
```

`factorial(N,X) :- N>0, NN is N-1, factorial(NN,Y), X is Y*N.`

## Ejercicios a desarrollar

1. Los divisores de un natural N son todos los naturales que lo dividen exactamente (incluyendo 1 y N). Por ejemplo los divisores de 12 son 1, 2, 3, 4, 6, 12.
  - a) Hacer un procedimiento `divisores/2` que imprima en pantalla todos los divisores del argumento N.
  - b) Ilustrar el correcto funcionamiento con 3 números compuestos.
  - c) Ilustrar el correcto funcionamiento con 3 números primos.
  - d) Que ocurre si se invoca con un entero negativo? En caso de error corregir el programa para que no genere errores. Ilustrar el resultado con un negativo.
  
2. Se desea desarrollar un procedimiento para obtener la lista de divisores de un número. Por ejemplo `listaDivisores(12,L,1)` debería retornar la lista de divisores de 12 empezando desde el 1, es decir, `L=[1,2,3,4,6,12]`.
  - a) Implementar el procedimiento `listaDivisores/3`.
  - b) Ilustrar el correcto funcionamiento de `listaDivisores` con 3 números compuestos.
  - c) Ilustrar el correcto funcionamiento de `listaDivisores` con 3 números primos.
  - d) Ilustrar el correcto funcionamiento cuando el entero de entrada es negativo.
  
3. Un número es primo si solo es divisible por si mismo y por la unidad.
  - a) Implementar el procedimiento `esPrimo/1` que compruebe si el argumento es número primo.
  - b) Ilustrar el correcto funcionamiento con 3 números primos y 3 números compuestos.
  - c) Ilustrar el correcto funcionamiento con los valores 0, 1 y -1.

## Informe

Implementar sus procedimientos en una base de datos `<NombreApellido>-<id>-bd8.pl` y las consultas respectivas en un archivo de texto `<NombreApellido>-<id>-consultas8.txt`. Documentar apropiadamente su programa.