



## Programación Declarativa

Ingeniería Informática  
Especialidad de Computación  
Cuarto curso. Primer cuatrimestre



Escuela Politécnica Superior de Córdoba  
Universidad de Córdoba

Curso académico: 2015 - 2016

---

### Práctica número 6.- Introducción al lenguaje Prolog

#### 1. Amantes

- Escribe un fichero denominado "**amantes.pl**" que contenga los siguientes hechos
  - *ama(juan, ana).*
  - *ama(ana, miguel).*
  - *ama(luis, isabel)*
  - *ama(miguel, ana).*
  - *ama(laura, juan).*
  - *ama(isabel, luis).*

donde el predicado *ama(X, Y)* indica que *X ama a Y*.

- Escribe en **prolog** las siguientes preguntas
  - *¿A quién ama "Juan"?*
  - *¿Quién ama a "Ana"?*
  - *¿Quién ama a alguien?*
  - *¿Quién es amado por alguien?*
  - *¿Quiénes se aman mutuamente?*
  - *¿Quién ama sin ser correspondido?*
- Añade al fichero *amantes.pl* una regla que permita describir a los "**amantes**", es decir, aquellas personas que se aman mutuamente.

#### 2. Familia

- Escribe un fichero denominado "*familia.pl*" que contenga los siguientes hechos:
  - *hombre(antonio).*
  - *hombre(juan).*
  - *hombre(luis).*
  - *hombre(rodrigo).*
  - *hombre(ricardo).*
  - *mujer(isabel).*
  - *mujer(ana).*
  - *mujer(marta).*
  - *mujer(carmen).*
  - *mujer(laura).*

- *mujer(alicia)*.
  - Define hechos en los que se afirmen los siguientes enunciados:
    - *Antonio y Ana son matrimonio*
    - *Juan y Carmen son matrimonio.*
    - *Luis e Isabel son matrimonio*
    - *Rodrigo y Laura son matrimonio.*
    - *Juan, Rodrigo y Marta son hijos de Antonio y Ana.*
    - *Carmen es hija de Luis e Isabel.*
    - *Ricardo es hijo de Juan y Carmen.*
    - *Alicia es hija de Rodrigo e Isabel.*
  - Define una regla que indique que "*matrimonio*" es reflexivo, es decir, si *X* e *Y* forma un matrimonio entonces *Y* y *X* también lo forman.
  - Define reglas para obtener:
    - *los nietos de una persona*
    - *los abuelos de una persona*
    - *los hermanos de una persona*
    - *los tíos de una persona*
    - *las tías de una persona*
    - *los primos de una persona*
    - *las primas de una persona*
    - *los suegros de una persona*
3. Declara los siguientes hechos relativos a trabajadores de una empresa.
- Utiliza el predicado ***encargado\_de\_tarea***(trabajador,tarea)
    - *Miguel está encargado de las tareas de admisión, control y vigilancia.*
    - *Ricardo está encargado de las tareas de planificación y asesoramiento.*
    - *Alicia está encargada de la dirección y control.*
  - Define reglas que permitan comprobar los siguientes hechos:
    - Si una tarea ha sido encargada a alguna persona. Utiliza el predicado ***encargada***(Tarea).
    - Si dos personas comparten alguna tarea, es decir, ***comparten\_tarea***(Persona1,Persona2).
4. Predicados sobre deportistas:
- Utiliza el predicado ***juega***(Persona,Deporte) para declarar lo siguientes hechos relativos a deportistas:
    - *Héctor juega al baloncesto.*
    - *Miguel juega al balonmano y rugby*
    - *Alicia juega al tenis, baloncesto y ajedrez.*
  - Define un predicado que permita comprobar si dos personas juegan al mismo deporte.
5. Escribe un programa que permita realizar las siguientes operaciones aritméticas:

- Mínimo de dos números.
- Área de un círculo.
- Área de un trapecio.
- Producto de los números comprendidos entre dos dados.

## 6. Lectores

- Escribe un fichero denominado "lectores.pl" que contenga los siguientes hechos que utilizan la estructura **nombre** y el predicado **lector**:
  - *lector(nombre("Ana", "Garrido", "Aguirre"),mujer,31).*
  - *lector(nombre("Marta", "Cantero", "Lasa"),mujer,20).*
  - *lector(nombre("Rodrigo", "Duque", "Soto"),hombre,30).*
  - Etc.
- Escribe en Prolog las siguientes preguntas:
  - *¿Hay lectores?*
  - *¿Quiénes son lectores?*
  - *¿Qué lectores son mujeres? y ¿hombres?*
  - *¿Hay lectores con el mismo nombre y diferentes apellidos?*
- Escribe una regla para comprobar si unos apellidos están repetidos.
  - Nota: utiliza el predicado **bagof** y un predicado auxiliar para **contar** los elementos de una lista.

## 7. Libros

- Utiliza el predicado "**prestado**" para definir algunos hechos en los que se indique que un *libro* (con título y autor) ha sido prestado a una persona (con nombre y apellidos).
- Por ejemplo:
 

```
prestado(libro("Misericordia", autor("Benito", "Pérez", "Galdós"),
              persona("Almudena", "Alegría", "Sol))).
```

 donde **libro** y **persona** son "estructuras".
- Construye preguntas en las que se indique:
  - *Si un lector tiene prestado algún libro.*
  - *Si un libro está prestado a alguien.*
  - *Si una persona es un escritor.*
  - *Si un escritor es leído.*
  - *Si existen autores leídos.*
- Define una regla para comprobar que un escritor es **leído** si alguno de sus libros está prestado.

## 8. Codifica los siguientes predicados sobre listas numéricas.

- Media de una lista
- Máximo de una lista.
- Mínimo de una lista
- crear(N,L): crea una lista a partir de un número.
  - Por ejemplo:
 

```
?crear(N,L)
L = [0,1,...,N].
```

9. Codifica los siguientes predicados sobre listas.
- **lugar\_impar(L,R)**: crea una lista que contiene sólo los elementos que ocupan un lugar impar.
    - Por ejemplo:
      - ?- *lugar\_impar*([a,b,c,d,e],R).
      - R = [a,c,e]
  - **comunes(L1,L2,R)**: R es la lista compuesta los elementos comunes a las listas L1 y L2
    - Por ejemplo:
      - ?- *comunes* ([a,b,c,d],[b,e,d,f],R).
      - R = [b,d]
10. Codifica un predicado denominado "**invertir**" para invertir todos los elementos de una lista que puede contener **sublistas**:
- Por ejemplo
    - ?- **invertir**([1,2,3,4,5],R).
    - R = [5, 4, 3, 2, 1].
  
    - ?- **invertir**([1,[2,3],[4,5]],R).
    - R = [[5, 4], [3, 2], 1].
  - Observación: codifica los siguientes predicados auxiliares
    - **es\_lista(X)**: comprueba si X es una lista
    - **concatenar(L1,L2,L)**: L es el resultado de concatenar L1 y L2.
11. Números primos
- Define el predicado **primo(N)** para comprobar si el número N es primo o no
    - Nota: un número es primo si no tiene divisores propios menores o iguales que su raíz cuadrada.
  - Define el predicado **crear\_primos(N,L)** para crear una lista compuesta por los números primos menores o iguales que el número N.
    - Por ejemplo:
      - ?- *crear\_primos*(10,L).
      - L = [2,3,5,7]
12. Monumentos
- Utiliza el predicado **monumento(Nombre,Localidad,Estilo)** para definir hechos asociados a los siguientes monumentos:
    - *Mezquita, Córdoba, Árabe*
    - *Medina Azahara, Córdoba, Árabe*
    - *Catedral, Santiago de Compostela, Románico*
  - Define el predicado **contar\_monumentos(Localidad,N)** para contar los monumentos que hay en una localidad.
    - Por ejemplo
      - ?- **contar\_monumentos**("Córdoba",N).

$N = 2$ .

- Observación:
  - Utiliza el predicado **bagof** o **findall**
  - Define un predicado auxiliar para **contar** para contar los elementos de una lista.

### 13. Donantes de sangre

- Declara los hechos relativos a una base de datos de donantes que contiene la siguiente información:
  - *donante(persona(juan, campos, ruiz), a, positivo)*.
  - *donante(persona(ana, lara, silva), ab, negativo)*.
  - *donante(persona(luis, luna, pachecho), ab, negativo)*.
  - Nota: **persona** es una estructura
- Escribe los hechos y las reglas que permitan comprobar si una persona **puede donar** sangre a otra teniendo en cuenta el grupo sanguíneo y el factor RH.
- Define reglas para el predicado **contar\_por\_grupo\_y\_factor** que permita contar todos los donantes de un grupo sanguíneo y factor rh específicos.
  - Por ejemplo:  
*?- contar\_por\_grupo\_y\_factor (ab, negativo, N).*  
 $N = 2$
  - Nota: utilizar el predicado **bagof** y un predicado auxiliar para **contar** los elementos de una lista.
- Escribe una regla que permita hacer las siguientes acciones consecutivas
  1. Pedir por pantalla un grupo sanguíneo y un factor rh,
  2. Pedir por pantalla el nombre de un fichero,
  3. Y escribir en dicho fichero los nombres de todos los donantes que tengan el grupo sanguíneo y el factor rh indicados.

### 14. Un árbol binario es representado por una lista de la forma

**[raíz, hijo izquierdo, hijo derecho]**

donde **raíz** es un átomo e **hijo izquierdo** e **hijo derecho** son árboles binarios.

- Define predicados para:
  - Escribir la lista en orden prefijo, sufijo e infijo.
  - Determinar la profundidad del árbol.
  - Comprobar si un elemento está en el árbol.
  - Determinar el número de nodos del árbol.
  - Determinar el número de hojas del árbol.
    - Un nodo es una hoja si sus hijos izquierdo y derecho son listas vacías.
- ¿Cómo se pueden redirigir las salidas de los predicados anteriores hacia un fichero de escritura?

15. Escribe un programa que permita contar los términos contenidos en un fichero y que escriba el resultado en otro fichero.
16. Escribe un programa que sume los números de un fichero y que escriba el resultado en otro fichero.