



UNIVERSIDAD DE CÓRDOBA

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE  
INFORMÁTICA Y ANÁLISIS NUMÉRICO



**LENGUAJES**

**DE INTELIGENCIA ARTIFICIAL**

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

SEGUNDO CURSO

PRIMER CUATRIMESTRE

**CURSO ACADÉMICO 2009 - 2010**



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

- Generación de múltiples soluciones
- Uso del corte

- **Generación de múltiples soluciones**
  - Uso del punto coma
  - bagof
  - setof
  - findall

Ejemplo 1: meses

mes(enero,31).

mes(febrero,28).

mes(marzo,31).

mes(abril,30).

mes(mayo,31).

mes(junio,30).

mes(julio,31).

mes(agosto,31).

mes(septiembre,30).

mes(octubre,31).

mes(noviembre,30).

mes(diciembre,31).

### Ejemplo 1: meses

Meses que tienen 31 días: se utiliza el operador punto y coma ;

?- mes(M,31).

M = enero ;

M = marzo ;

M = mayo ;

M = julio ;

M = agosto ;

M = octubre ;

M = diciembre ;

No

### Ejemplo 1: meses

Lista de meses que tienen 31 días:

?- bagof(M,mes(M,31),L).

M = \_G624

L = [enero, marzo, mayo, julio, agosto, septiembre, octubre, diciembre] ;

Lista ordenada de meses que tienen 31 días:

?- setof(M,mes(M,31),L).

M = \_G624

L = [enero, marzo, mayo, julio, agosto, septiembre, octubre, diciembre] ;



### Ejemplo 1: meses

Lista de meses sin tener en cuenta los días que tienen

?- bagof(M,D^mes(M,D),L).

M = \_G1701

D = \_G1702

L = [enero, marzo, mayo, julio, agosto, septiembre | ...]

Lista ordenada de los meses sin tener en cuenta los días que tienen

?- setof(M,D^mes(M,D),L).

M = \_G2548

D = \_G2549

L = [abril, agosto, diciembre, enero, febrero, julio, junio, marzo, mayo | ...]

## Ejemplo 1: meses

Lista de meses sin tener en cuenta los días que tienen

?- findall(M,mes(M,D),L).

M = \_G5549

D = \_G5550

L = [enero, febrero, marzo, abril, mayo, junio, julio, agosto, septiembre|...] ;

### Ejemplo 1: meses

#### Lista de meses y días

?- bagof(M/D,mes(M,D),L).

M = \_G3431

D = \_G3432

L = [enero/31, febrero/28, marzo/31, abril/30, mayo/31, junio/30,  
julio/31, agosto/31, ... /... |...]

#### Lista ordenada de los meses y días

?- setof(M/D,D^mes(M,D),L).

M = \_G4633

D = \_G4634

L = [abril/30, agosto/31, diciembre/31, enero/31, febrero/28, julio/31,  
junio/30, marzo/31, ... /... |...]

## Ejemplo 2: personas

persona(patricia,9,femenino).

persona(laura,9,femenino).

persona(juan,9,masculino).

persona(teresa,8,femenino).

persona(pedro,8,masculino).

persona(laura,8,femenino).

## Ejemplo 2: personas

buscar\_por\_edad(E,L):-

bagof(P,S^persona(P,E,S),L).

buscar\_ordenado\_por\_edad(E,L):-

setof(P,S^persona(P,E,S),L).

buscar\_por\_edad\_2(E,L):-

findall(P,persona(P,E,\_),L).

## Ejemplo 2: personas

buscar\_por\_sexo(S,L):-

bagof(P,E^persona(P,E,S),L).

buscar\_ordenado\_por\_sexo(S,L):-

setof(P,E^persona(P,E,S),L).

buscar\_por\_sexo\_2(S,L):-

findall(P,persona(P,\_,S),L).

## Ejemplo 2: personas

buscar\_personas\_1(L):-

bagof(P,E^S^persona(P,E,S),L).

buscar\_personas\_2(L):-

findall(P,persona(P,\_,\_),L).

## Ejemplo 2: personas

```
numero_personas_por_edad(E,N):-  
    findall(X,persona(X,E,_),L),  
    contar(L,N).
```

```
contar([],0).
```

```
contar([_ | Cola],N):-  
    contar(Cola,N1),  
    N is N1+1.
```



Generación de múltiples soluciones usando el punto y coma ;

? persona(P,9,femenino).

P = patricia ;

P = laura ;

No

? bagof(P, persona(P, 9, femenino), L).

P = \_G386

L = [patricia, laura]

? setof(P, persona(P, 9, femenino), L).

P = \_G386

L = [laura, patricia]

?- buscar\_por\_edad(9,L).

L = [patricia, laura, juan]

?- buscar\_ordenado\_por\_edad(9,L).

L = [juan, laura, patricia]

?- buscar\_por\_edad\_2(9,L).

L = [patricia, laura, juan]

?- buscar\_por\_sexo(femenino,L).

L = [patricia, laura, teresa, laura]

?- buscar\_ordenado\_por\_sexo(femenino,L).

L = [laura, patricia, teresa]

?- buscar\_por\_sexo\_2(femenino,L).

L = [patricia, laura, teresa, laura]

?- buscar\_personas\_1(L).

L = [patricia, laura, juan, teresa, pedro, laura]

?- buscar\_personas\_2(L).

L = [patricia, laura, juan, teresa, pedro, laura]

?numero\_personas\_por\_edad(9,N).

N = 3

- **Uso del corte**
  - Evitar la generación de múltiples soluciones
  - Confirmación de una regla
  - Provocar un fallo inmediato: corte y fail

- Uso del corte
  - Evitar la generación de múltiples soluciones
  - Confirmación de una regla
  - Provocar un fallo inmediato: corte y fail

## Evitar la generación de soluciones múltiples

### Ejemplo 1: función definida por partes

/\*

\* Función de saltos definida por partes

\*

\*/

$f(X,0) :- X < 3.$

$f(X,2) :- 3 \leq X, X < 6.$

$f(X,4) :- 6 \leq X.$

La pregunta

?  $f(1,R), 2 < R.$

hacen intentos de búsqueda superfluos



## Evitar la generación de soluciones múltiples

### Ejemplo 1: función definida por partes

/\*

\* Segunda versión

\* Función de saltos definida por partes

\* Se utiliza el corte

\*

\*/

$g(X,0) :- X < 3, !.$

$g(X,2) :- 3 \leq X, X < 6, !.$

$g(X,4) :- 6 \leq X.$

## Evitar la generación de soluciones múltiples

### Ejemplo 1: función definida por partes

```
/*  
* Tercera versión  
* Función de saltos definida por partes  
* Se utiliza el corte  
*  
*/
```

$h(X,0) :- X < 3, !.$

$h(X,2) :- X < 6, !.$

$h(_,4).$

## Comparación entre el "not" y el "corte"

/\* Uso del predicado "not": claridad semántica \*/

A: - B, C

A: - not(B), D

/\* Uso del corte: eficiencia \*/

A: -B, !, C

A: - D

## Evitar la generación de soluciones múltiples

### Ejemplo 2: nota alfanumérica

/\*

\* nota determina la calificación alfanumérica

\* a partir de la nota numérica

\*

\*/

nota(X,suspenso):- X < 5.

nota(X,aprobado):- 5 =< X, X < 7.

nota(X,notable):- 7 =< X, X < 9.

nota(X,sobresaliente):- 9 =< X, X < 10.

nota(X,matricula\_honor):- X = 10.

## Evitar la generación de soluciones múltiples

### Ejemplo 2: nota alfanumérica

```
/*  
 * nota_bis determina la calificación alfanumérica a partir de la nota  
numérica  
 * Se utiliza el corte  
 *  
 */
```

```
nota_bis(X,suspenso):- X < 5, !.  
nota_bis(X,aprobado):- X < 7, !.  
nota_bis(X,notable):- X < 9, !.  
nota_bis(X,sobresaliente):- X < 10, !.  
nota_bis(10, matricula_honor).
```

## Evitar la generación de soluciones múltiples

### Ejemplo 2: nota alfanumérica

#### OBSERVACIÓN

? nota(3,sobresaliente).

No

? nota\_bis(3,sobresaliente).

Yes

## Evitar la generación de soluciones múltiples

### Ejemplo 3: factorial de un número

/\*

\* Primera versión

\*

\* factorial\_1(N,R): R es el factorial de N, es decir,  $R = N!$

\* Se usa la negación not

\* No controla los valores negativos de N

\*/

factorial\_1(0, 1).

factorial\_1(N,R):-

not(N = 0),

N1 is N - 1,

factorial\_1(N1,R1),

R is N \* R1.

## Evitar la generación de soluciones múltiples

### Ejemplo 3: factorial de un número

```
/*  
* Segunda versión  
*  
* factorial_2(N,R): R es el factorial de N, es decir, R = N!  
* Si N es no positivo entonces R vale 0  
* Se usa la negación not  
*/
```

```
factorial_2(N, 1):- N =< 0.
```

```
factorial_2(N,R):-  
    not(N =< 0), /* N > 0 */  
    N1 is N - 1,  
    factorial_2(N1,R1),  
    R is N * R1.
```



## Evitar la generación de soluciones múltiples

### Ejemplo 3: factorial de un número

/\*

\* Tercera versión

\*

\* factorial\_3(N,R): R es el factorial de N, es decir,  $R = N!$

\* Esta versión es equivalente a la primera, pero utiliza el corte

\* No controla los valores negativos de N

\*

\*/

factorial\_3(0,1):-!.

factorial\_3(N,R):-

    N1 is N - 1,

    factorial\_3(N1,R1),

    R is N \* R1.

## Evitar la generación de soluciones múltiples

### Ejemplo 3: factorial de un número

```
/*  
* Cuarta versión  
*  
* factorial_4(N,R): R es el factorial de N, es decir, R = N!  
* Esta versión es equivalente a la segunda, pero utiliza el corte  
*  
*/
```

```
factorial_4(N, 1):- N =< 0, !.
```

```
factorial_4(N,R):-  
    N1 is N - 1,  
    factorial_4(N1,R1),  
    R is N * R1.
```

- Uso del corte
  - Evitar la generación de múltiples soluciones
  - Confirmación de una regla
  - Provocar un fallo inmediato: corte y fail

## Confirmación del uso de una regla

### Ejemplo 4: préstamo de libros

*/\* Servicios \*/*

servicio\_basico(consulta).  
servicio\_basico(referencia).

servicio\_adicional(prestamo).  
servicio\_adicional(prestamo\_interbibliotecario).

servicio\_general(X):- servicio\_basico(X).  
servicio\_general(X):- servicio\_adicional(X).

## Confirmación del uso de una regla

### Ejemplo 4: préstamo de libros

*/\* Servicios \*/*

servicio(Persona, Servicio):-

lector(Persona),  
prestamo(Persona, \_),  
!,  
servicio\_basico(Servicio).

servicio(Persona, Servicio):-

lector(Persona),  
servicio\_general(Servicio).

## Confirmación del uso de una regla

### Ejemplo 4: préstamo de libros

*/\* Lectores \*/*

lector('Juan Campos Aguilera').

lector('Ana Silva Arroyo').

lector('Pedro Luque Salas').

*/\* Préstamos \*/*

prestamo('Juan Campos Aguilera','Nazarín').

prestamo('Juan Campos Aguilera','Misericordia').

prestamo('Ana Silva Arroyo','La cruz del Sur').

*/\* La pregunta \*/*

?- servicio(P,S).

## Confirmación del uso de una regla

### Ejemplo 5: deportistas

vence(luis, diego).  
vence(ana, luis).  
vence(ana, diego).

## Confirmación del uso de una regla

### Ejemplo 5: deportistas

/\*

\* Se determina la clase de una persona

\* luchador: gana y pierde

\* ganador: siempre gana, es decir, gana y no pierde

\* deportista: siempre pierde, es decir, pierde y no gana.

\*/

```
clase(P,luchador):- vence(P,_),  
                   vence(_,P).
```

```
clase(P,ganador):- vence(P,_),  
                  not(vence(_,P)).
```

```
clase(P,deportista):- vence(_,P),  
                     not(vence(P,_)).
```



## Confirmación del uso de una regla

### Ejemplo 5: deportistas

```
/*  
 * Segunda versión: se utiliza el corte  
 *  
 */  
clase_bis(P,luchador):- vence(P,_),  
                       vence(_,P),  
                       !.  
  
clase_bis(P,ganador):- vence(P,_),  
                       !.  
  
clase_bis(P,deportista):- vence(_,P),  
                          !.
```

- Uso del corte
  - Evitar la generación de múltiples soluciones
  - Confirmación de una regla
  - Provocar un fallo inmediato: corte y fail

## Combinación de “corte” y “fail”

### Ejemplo 6: Vegetarianos

```
come(Persona,X):-  
    vegetariano(Persona),  
    not(verdura(X)),  
    !,  
    fail.
```

```
come(_,X):- comida(X).
```

```
comida(X):- carne(X); verdura(X); pescado(X).
```

## Combinación de “corte” y “fail”

### Ejemplo 6: Vegetarianos.

verdura(espinacas).  
verdura(acelgas).  
carne(cerdo).  
carne(ternera).  
pescado(bacalao).  
pescado(merluza).

vegetariano(anselmo).  
vegetariano(alicia).

La pregunta

? come(anselmo, cerdo).

No

## Combinación de “corte” y “fail”

### Ejemplo 6: animales

le\_gusta(maria, X):- serpiente(X), !, fail.

le\_gusta(maria, X):- animal(X).

ave(loro).

ave(buitre).

serpiente(boa).

serpiente(anaconda).

animal(X):- serpiente(X); ave(X).



UNIVERSIDAD DE CÓRDOBA

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE  
INFORMÁTICA Y ANÁLISIS NUMÉRICO



**LENGUAJES**

**DE INTELIGENCIA ARTIFICIAL**

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

SEGUNDO CURSO

PRIMER CUATRIMESTRE

**CURSO ACADÉMICO 2009 - 2010**

