

## Práctica No. 6 Ejercicios en Prolog.

### Introducción

El lenguaje Prolog (Programación Lógica) puede ser visto como un lenguaje prescriptivo así como un lenguaje descriptivo. El enfoque de Prolog es describir un conjunto de hechos conocidos y un conjunto de relaciones entre esos hechos respecto al problema a solucionar, más que describir la secuencia de pasos que debe seguir la computadora para resolver el problema. Prolog tiene implementada cierta "inteligencia" para resolver el problema (esta "inteligencia" es un mecanismo de razonamiento automático llamado "motor de inferencia" basado en un algoritmo que se conoce como resolución).

Un programa en Prolog consiste de un conjunto de cláusulas, donde cada cláusula es, o un hecho o una regla acerca de como puede inferirse alguna solución para los hechos dados.

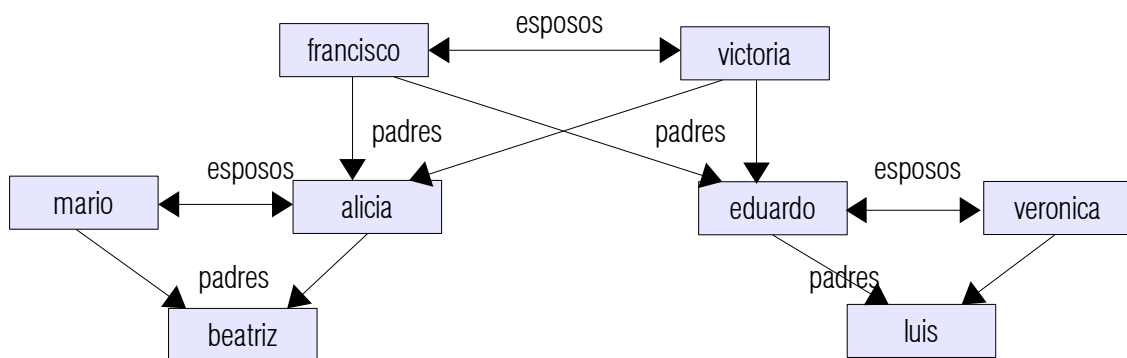


Figura 1.- Una bonita familia.

### Actividades

1.- Captura las siguientes cláusulas y guarda el archivo con el nombre **familiares.pl** (que representa las relaciones familiares mostradas en la Figura 1).

```
/*  
  Archivo: Familiares.pl  
  Clausulas sobre una familia y reglas para determinar las relaciones  
  familiares  
*/  
varon(eduardo).  
varon(francisco).  
varon(luis).  
varon(mario).  
  
mujer(alicia).  
mujer(veronica).  
mujer(victoria).  
mujer(beatriz).  
  
padres(eduardo,francisco,victoria).  
padres(alicia,francisco,victoria).  
padres(luis,eduardo,veronica).  
padres(beatriz,alicia,mario).  
  
esposos(eduardo,veronica).
```

```
esposos(mario,alicia).
esposos(francisco,victoria).
```

```
hermana(Ella,X) :- mujer(Ella),padres(Ella,M,P),padres(X,M,P).
hermano(El,X) :- varon(El),padres(El,M,P),padres(X,M,P).
```

```
hijo(El,X) :- varon(El),padres(El,X,_).
hijo(El,X) :- varon(El),padres(El,_,X).
```

```
hija(Ella,X) :- mujer(Ella),padres(Ella,X,_).
hija(Ella,X) :- mujer(Ella),padres(Ella,_,X).
```

2.- Ejecuta el intérprete de Prolog dando click sobre su ícono (tecleando `p1`, desde una terminal X en Linux)

2.1.- Para cargar el programa contenido en el archivo **familiares.pl**, tecleamos:

```
consult('familiares.pl').
```

O también puede cargarse tecleando:

```
['familiares.pl']
```

2.2.- “Convierte” a Prolog las siguientes preguntas, ejecútalas y reporta los resultados:

a) ¿Eduardo y Alicia son hermanos?

Consulta en Prolog: \_\_\_\_\_

Resultados: \_\_\_\_\_

b) ¿Quiénes son los padres de Beatriz?

Consulta en Prolog: \_\_\_\_\_

Resultados: \_\_\_\_\_

c) ¿Eduardo es hijo de Mario?

Consulta en Prolog: \_\_\_\_\_

Resultados: \_\_\_\_\_

d) ¿Luis es hijo de Verónica?

Consulta en Prolog: \_\_\_\_\_

Resultados: \_\_\_\_\_

e) ¿De quién es hija Beatriz?

Consulta en Prolog: \_\_\_\_\_

Resultados: \_\_\_\_\_

2.3.- Define reglas para las relaciones **nieto** y **nieta**.

Regla para nieto: \_\_\_\_\_

Regla para nieta: \_\_\_\_\_

2.4.- Define reglas para la relación **abuelo** y **abuela**

Regla para abuelo: \_\_\_\_\_

Regla para abuela: \_\_\_\_\_

2.5.- Edita el archivo **familiares.pl** para agregar las reglas definidas en los pasos anteriores, guarda el archivo, cárgalo nuevamente en el intérprete de Prolog y realiza las siguientes consultas:

a) ¿Eduardo es abuelo?

Consulta en Prolog: \_\_\_\_\_

Resultados: \_\_\_\_\_

b) ¿Victoria es abuela?

Consulta en Prolog: \_\_\_\_\_

Resultados: \_\_\_\_\_

c) ¿De quien es nieto Luis?

Consulta en Prolog: \_\_\_\_\_

Resultados: \_\_\_\_\_

d) ¿Francisco es abuelo de Beatriz?

Consulta en Prolog: \_\_\_\_\_

Resultados: \_\_\_\_\_

2.6.- Define al menos 2 de las siguientes relaciones: sobrino, sobrina, cuñado o cuñada. Codifícala en Prolog, modifica el archivo **familiares.pl** y prueba que la relación está bien definida por medio de varias preguntas.

Relación: \_\_\_\_\_

Codificación: \_\_\_\_\_

Preguntas y resultados:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

3.- Captura el siguiente programa que ilustra cómo aprobar un examen:

```
% Archivo: aprexamen.pl
% Predicados
aprueba_examen(X) :- preparado_para_examen(X).
aprueba_examen(X) :- persona(X),examen_facil.
preparado_para_examen(X) :- sabe_todo(X).
preparado_para_examen(X) :- asiste_a_clases(X), realiza_ejercicios(X),
lee_libro(X).
preparado_para_examen(X) :- preparado_para_examen(Y),tutor(Y,X).

% Proposiciones
persona(maria).
persona(alfredo).
persona(tomas).
persona(susana).
persona(juan).

lee_libro(alfredo).
lee_libro(maria).
```



