

Dado el nuevo furor (y negocio?) de las redes sociales en Internet, pensamos un sistema en prolog que me resuelva la lógica que se necesita para esta actividad.



Se tiene la siguiente base de conocimientos perteneciente a una red social que tiene los siguientes datos

```
usuario(leo).
viveEn(leo, direccion(argentina, liniers)).
nacimiento(leo, años(27)).
tieneAmigo(leo, gaston).
tieneAmigo(leo, gabi).

usuario(gabi).
viveEn(gabi, direccion(argentina, liniers,1407)).
nacimiento(gabi, fecha(30,09,1982)).
tieneAmigo(gabi, leo).
tieneAmigo(gabi, fer).

usuario(fer).
viveEn(fer, direccion(argentina, liniers)).
nacimiento(fer, años(27)).
tieneAmigo(fer, gabi).

hoy(fecha(2,10,2008)).

grupo(ayudantesParadigmas).
integra(ayudantesParadigmas, leo).
integra(ayudantesParadigmas, gabi).
integra(ayudantesParadigmas, fer).
```

Se pide

- 1) Realice las consultas que permitan determinar
 - Quienes son los todos amigos de leo
 - Si leo tiene 27 años
 - Si hay por lo menos una persona que tenga CP 1407
- 2) Realice el predicado **sonAmigos/2** que relacionan a dos personas si son amigas. Dos personas son amigas si una persona esta en la lista de amigos de la otra y viceversa.

- 3) Definir el predicado **amigosQueCumplenEsteMes/2** que me relaciona un usuario con la lista de sus amigos que cumplen años el mes actual.

Luego responder: Por la forma en que está la información, el individuo fer: ¿podría estar contenido en alguna respuesta? ¿Por qué?

- 4) Resuelva el predicado **talVezConozcas/2**, que relaciona una persona con otra que puede llegar a conocer, dos personas se pueden llegar a conocer si:
- Viven en la misma ciudad
 - Tienen **amigos** en común
 - Los dos son parte de un mismo grupo

En la resolución de este punto, ¿donde aparece el concepto de pattern matching?

- 5) a) Codifique el predicado **esPopular/1** que verifica si una persona es popular, donde una persona es popular si es amiga de todos los miembros de los grupos a los que pertenece.
Realizar el predicado utilizando predicados de orden superior. Indicar si el predicado es inversible, Justificar.

b) Indicar la consulta que hay que realizar para obtener en una lista todos los usuarios populares.

- 6) Existe una **teoría** que dice que todas las personas **están** conectadas entre si y puedo llegar a conocer a cualquier persona en le mundo a través de mis contactos, se pide realizar el predicado **estanConectados/2** que relaciona una persona con otra persona que esta conectada a **través** de sus contactos

```
?-estanConectados(leo, fer).  
Yes.
```

Porque leo conoce a gabi y gabi conoce a fer

Resolución

%1

```
%findall(Amigo,tieneAmigo(leo,Amigo),Amigos).  
%nacimiento(leo,anios(27)).  
%viveEn(_,direccion(_,_,1407)).
```

%2

```
sonAmigos(Per1,Per2):- tieneAmigo(Per1,Per2), tieneAmigo(Per2,Per1).
```

%3

```
amigosQueCumplenEsteMes(Persona,Amigos):- usuario(Persona),  
                                             findall( Amigo, (sonAmigos(Persona,Amigo),  
                                             cumpleEsteMes(Amigo)), Amigos).
```

```
cumpleEsteMes(Amigo):-nacimiento(Amigo, fecha(_,Mes,_)), hoy(fecha(_,Mes,_)).
```

%4

```
talVezConozcas(Per1,Per2):-estanEnLaMismaCiudad(Per1,Per2),  
                             not(sonAmigos(Per1,Per2)),  
                             Per1\=Per2.
```

```
talVezConozcas(Per1,Per2):- tieneAmigo(Per1,Amigo), tieneAmigo(Per2,Amigo),  
                             not(sonAmigos(Per1,Per2)),  
                             Per1\=Per2.
```

```
talVezConozcas(Per1,Per2):-  integra(Grupo,Per1),  
                             integra(Grupo,Per2),  
                             not(sonAmigos(Per1,Per2)),  
                             Per1\=Per2.
```

```
estanEnLaMismaCiudad(Pers, Otro):- ciudad(Pers, Ciudad), ciudad(Otro, Ciudad).
```

```
ciudad(Pers, Ciudad):- persona(Pers, _, direccion(_,Ciudad, _), _).
```

```
ciudad(Pers, Ciudad):- persona(Pers, _, direccion(_,Ciudad, _), _).
```

