

Guía de Trabajos Prácticos

¿Dónde está el control remoto?

[Hechos, reglas y predicados]

Se trata de una familia numerosa donde viven 3 hermanos. Uno de ellos, Rodrigo, un día quiso ver la televisión, pero se encontró con un problema: El control remoto no estaba en ningún lado.

Rodrigo sabe que uno de los hermanos escondió el control remoto para tomarse revancha de conflictos anteriores, pero no sabe cuál. Pero con tanto tiempo compartido junto a sus hermanos, descubrió las siguientes pistas:

- Los hermanos son Rodrigo, Miguel y Natalia
- Un hermano le esconde a otro cosas cuando lo pelea y no es menor que él.
- Miguel se pelea con todos los hermanos que no son peleados por Rodrigo.
- Rodrigo se pelea con todos sus hermanos, salvo con Natalia.
- El hermano que no es peleado por Natalia es menor que Rodrigo.
- Natalia se pelea con las mismas personas que Rodrigo.

¿Quién escondió el control remoto?

Soñaba jugar un mundial

[Hechos, reglas, predicados, unificación]

Escribir un programa **Prolog** que responda consultas acerca de cuáles son los rivales de una determinada selección en un campeonato mundial.

Una selección tiene como rivales todos los otros equipos de su mismo grupo (¡nunca juega contra sí misma!).

Incluir en el programa la siguiente información:

- El grupo A está formado por Colombia, Camerún, Jamaica e Italia.
- El grupo B está formado por Argentina, Nigeria, Japón y Escocia.

El programa debe ser capaz de responder, p.ej., a la siguiente consulta:

- ¿cuáles son los rivales de Argentina?

otorgando como respuestas "Nigeria", "Japón" y "Escocia".

Es Cupido

[Hechos, reglas, predicados]

Escribir un programa **Prolog** que ayude a una agencia matrimonial ("*Es Cupido SRL*") a armar parejas.

Definimos a una pareja como un par (mujer, varón).

La agencia tiene esta información:

- Las mujeres melancólicas son compatibles con los varones serenos.
- Las mujeres decididas son compatibles con los varones reflexivos.
- Las mujeres soñadoras son compatibles con los varones decididos.
- Juan es sereno y decidido.
- María es melancólica.
- Ursula es decidida.
- Juana es soñadora.
- Pedro es reflexivo.
- José es melancólico.

Se pide:

a. Describir la información que maneja la agencia en **Prolog** de forma tal que se pueda preguntar qué parejas son compatibles. Por ejemplo:

- ante la pregunta sobre si la pareja (María, Juan) es compatible debe responder que sí.
- ante la pregunta sobre si la pareja (Ursula, Juan) es compatible debe responder que no.

b. Agregar al programa la siguiente información:

- Cualquier pareja formada por un decidido y un melancólico es compatible.

Según esta nueva información, la pareja (Ursula, José) es compatible, mientras que si nos remitimos al punto **a.**, no lo es.

c. Agregar al programa la posibilidad de responder a consultas sobre si una persona es deseable.

Decimos que una persona (varón o mujer) es deseable si es compatible con, por lo menos, dos personas distintas.

Según la información descripta, Juan es deseable mientras que Juana no lo es.

Asesinato en la mansión Dreadbury

[Hechos, reglas, predicados, generación, not]

Armar un programa *Prolog* que resuelva el siguiente problema lógico:



- Un asesino siempre odia a su víctima y nunca es más rico que ella. El asesino de la tía Agatha, además, vive en la mansión Dreadbury.
- Tía Agatha, el carnicero y Charles son las únicas personas que viven en la mansión Dreadbury.
- Charles odia a todas las personas de la mansión que no son odiadas por la tía Agatha.
- Agatha odia a todos los que viven en la mansión, excepto al carnicero.
- Quien no es odiado por el carnicero y vive en la mansión, es más rico que tía Agatha.
- El carnicero odia a las mismas personas que odia tía Agatha.

Al programa le tengo que poder preguntar quién es el asesino de la tía Agatha, y tiene que brindar la respuesta correcta.

Búsqueda laboral

[Hechos, reglas, predicados]

Se cuenta con la siguiente base de conocimientos:

```
contador(roque).
joven(roque).
trabajoEn(roque,acme).
habla(roque,frances).
honesto(roque).
ingeniero(ana).
habla(ana,ingles).
habla(ana,frances).
trabajoEn(ana,omni).
habla(lucia,ingles).
habla(lucia,frances).
trabajoEn(lucia,omni).
abogado(cecilia).
ambicioso(cecilia).
habla(cecilia,frances).
```

Y se sabe que:

- Los contadores jóvenes son ambiciosos.
- Se considera que alguien tiene experiencia si trabajó en alguna empresa.
- Alguien es profesional si es contador, o abogado, o ingeniero.
- Si alguien es ambicioso, entonces puede andar en comercio exterior.
- En contaduría sólo pueden andar los contadores honestos.
- En ventas pueden andar los que sean ambiciosos y tengan experiencia.
- Lucía siempre puede andar en ventas.

1. Implementar el predicado **puedeAndar/2** (y todos los predicados auxiliares que se consideren necesarios) que relaciona una profesión y una persona que pueda cumplirla.

2. Agregar la información para las búsquedas de estas secciones:

- proyectos: pueden andar ingenieros con experiencia y también abogados jóvenes.
- logística: pueden andar profesionales que o bien sean jóvenes o bien hayan trabajado en Omni.

3. Agregar postulantes tales que:

- uno pueda andar para proyectos pero no para logística
- otro pueda andar para ventas pero no para contaduría

Gossip

[Unificación, predicados]

Dado el siguiente programa:

```
gustaDe(luis,nora).
gustaDe(roque,nora).
gustaDe(roque,ana).
gustaDe(marcos,zulema).
gustaDe(X,zulema):- gustaDe(X,ana).
gustaDe(juan,X):- gustaDe(roque,X).
gustaDe(X,Y):- gustaDe(X,ana).
gustaDe(juan,nuria).
compiten(X,Y):- gustaDe(X,Z), gustaDe(Y,Z).
debeDinero(juan,marcos).
debeDinero(juan,roque).
```

se pide:

a. determinar con cuáles de las cláusulas unifica cada una de las consultas de acá abajo. En cada caso que una cláusula unifica con una consulta, indicar qué variables se ligan y con qué átomos.

```
?- gustaDe(juan,A).
?- gustaDe(A,zulema).
?- gustaDe(marcos,ana).
?- gustaDe(juan,zulema).
?- gustaDe(A,B).
```

b. Describir en castellano a quiénes relaciona el predicado compiten, con una descripción por comprensión.

¡Fiesta!

[Unificación, hechos, reglas, predicados]

Se está organizando una fiesta para una cátedra. Para saber quién puede ir a la fiesta y quién no, se establece quién es alumno de quién. También se tiene información de otras personas, que no son de la facultad.

La información es esta:

- Luisa y Juan son alumnos de Daniel.
- Ana es alumna de Luisa.
- Diana y Nahuel son alumnos de Nico.
- Tamara es alumna de Nahuel.
- Claudio y José son alumnos de Rubén.
- Alvaro es alumno de José y de Luisa.
- Brad, Leo y Johnny son carilindos.
- Luciano y Lautaro son simpáticos.

A la fiesta pueden ir: Nico, Daniel, y los alumnos de alguien que puede ir.

Como las chicas insistieron, también pueden ir los carilindos.

Armar un programa **Prolog** al que se le pueda preguntar quiénes pueden ir a la fiesta

Vomistar

[Hechos, reglas, predicados, inversibilidad]

La siguiente es la nómina de personal de *Vomistar SA*:

- **Departamento de ventas:** empleada María, cadetes Juan y Roque
- **Departamento de compras:** empleada Nora, cadete Pedro
- **Departamento de administración:** empleados Felipe y Hugo, cadete Ana.

Escribir un programa **Prolog** que modele a *Vomistar SA*, tal que puedan responderse las siguientes consultas :

- ¿quiénes trabajan en el departamento de compras? ¿y en el de ventas?
- dadas dos personas, ¿trabajan en el mismo departamento?
- dadas dos personas **a** y **b**, ¿puede **a** darle órdenes a **b**?

Decimos que **a** puede darle órdenes a **b** si y sólo si trabajan en el mismo departamento y **a** tiene un cargo superior al de **b**. Se considera que “*empleado*” es un cargo superior a “*cadete*”.

Para gustos, colores

[Unificación, predicados, hechos, reglas]

a. Escribir un programa **Prolog** que responda consultas acerca de qué colores atraen a una determinada persona, de acuerdo con la siguiente información:

- A Mabel y a Ana les atrae el rosa.
- A Mara le atraen el celeste y el lila.
- A Juan le atraen todos los colores pastel.
- A todas las mujeres y a Pablo les atrae el azul.
- A los varones mayores y a Mabel les atrae el rojo.
- A todos los porteños (sin importar el sexo) y a Adrián les atrae el amarillo.
- A Ana y a Juan les atrae el naranja.
- Mabel, Mara y Pablo son porteños.

- Ana y Pablo son mayores.
- El rosa, el celeste y el lila son colores pastel.

El programa debe ser capaz de responder, p.ej., a la siguiente consulta: ¿qué colores atraen a Mabel? Obteniendo como respuestas "rosa", "azul", "rojo" y "amarillo".

b. Modificar el programa anterior para que responda consultas acerca de con qué colores puede iluminarse una determinada reunión.

Se dice que una reunión puede iluminarse con un color si entre los asistentes a la reunión hay una pareja (dos personas de distinto sexo) a quienes les atrae el color.

Incluir en el programa la siguiente información:

- Mabel, Ana, Adrián y Pablo asistirán a la reunión del viernes.
- Mara, Mabel, Adrián y Juan asistirán a la reunión del sábado.
- Las personas mayores y Juan asistirán a la reunión del domingo.

El programa debe ser capaz de responder, p.ej., a la siguiente consulta

¿con qué colores puede iluminarse la reunión del sábado? Obteniendo como respuestas "rosa", "celeste", "lila" y "amarillo".

Los topos

[functores, polimorfismo, listas, findall, inversibilidad, generación]

Se cuenta con la siguiente base de conocimientos:

```
entierra(juanCarlos, [originalidad]).
%% Old McDonald had a farm iaiahoooooooooy
entierra(oldMcDonald, [semillaDeLino, semillaDeGirasol]).
entierra(ignacio, [televisor, computadora, fotosFamiliares, mocos, ignacio]).
entierra(pobreton, []).
entierra(turistaDeLaPlaya, [turistaDeLaPlaya]).
%% Se sabe de un pirata su nombre y los mares por los que navegó.
entierra(pirata(barbaRoja, [atlantico, pacifico, indico, artico, antartico]),
[joyas, oro]).
entierra(pirata(jackSparrow, [atlantico]), [oro, mapa]).
entierra(pirata(piraton, []), [anilloDeCompromiso, relacionMatrimonial]).
entierra(pirata(alCapone, []), [declaracionJurada, oro, dolares, armas, victimas,
drogas]).
%% De los perros se sabe el nombre, la raza y la edad en años.
entierra(perro(perroDinamita, ovejero, 3), [hueso]).
entierra(perro(lazy, collie, 8), [victimas]).
```

Se pide:

1. Saber si alguien tiene síndrome de diógenes (o síndrome de acumulación), es decir, si entierra más de 5 objetos.
2. Implementar el predicado **enterro/2** que relaciona a una persona (o perro) con una cosa que enterró.
3. Implementar el predicado **superficial/1** que dice si alguien no enterró nada.
4. Implementar **seQuiereIr/1** que indica si alguien se enterró a sí mismo.

5. Saber si algo es un tesoro (si lo enterró un pirata).
6. Saber quién es rico (alguien es rico si entierra oro).
7. Los piratas que no navegan por ningún mar son nauseosos. Saber quiénes son los nauseosos.
8. Saber si alguien es peligroso (si entierra víctimas).
9. Implementar **estaViejito/1** que se cumple para los perros de siete años o más.
10. Implementar **esPirataArgento/1** que da verdadero para los piratas cuyo primer mar navegado es Mar Del Plata.
11. Implementar el predicado **granPirata/1** que obtiene una lista con todos los tesoros enterrados (recordar que algo es un tesoro si fue enterrado por un pirata, como lo establece el punto 5).
12. Saber si estamos al horno. Estamos al horno si hay más piratas peligrosos que piratas nauseosos.
13. Para evitar que otros piratas encuentren los tesoros enterrados, los piratas suelen dividir sus tesoros en dos partes. Implementar el predicado **dividirTesoro/3** que relaciona un pirata con dos listas con sus tesoros, de modo que la unión de esas listas sea la totalidad del tesoro y que ninguna de las listas sea vacía.

Nota: todos los predicados “principales” (no los auxiliares) deben ser completamente inversibles. Además, cuando se deba unificar contra una persona o perro debe hacerse contra toda la persona o perro, no sólo contra el nombre.

Sooooomos los piratas

[Explosión combinatoria, inversibilidad, not, listas, findall, recursividad]

El capitán *Barbazul* tiene muchos tripulantes posibles para partir en su barco pirata y nos pidió que lo ayudáramos para ver las posibles tripulaciones que podría llevar. Se conocen los siguientes posibles tripulantes, modelados en base a los siguientes hechos:

```
mujer(betina).
mujer(laura).
mujer(carola).
bonita(laura).
bonita(betina).
cocinera(laura).
pirata(felipe, 27). /*nombre, edad*/
pirata(marcos, 39).
pirata(facundo, 45).
pirata(tomas, 20).
pirata(gonzalo, 22).
bravo(tomas).
bravo(felipe).
bravo(marcos).
```

La primera premisa que tiene el capitán es que la ferocidad total de sus tripulantes sea de al menos 10, sabiendo que:

- Las mujeres no cocineras tienen una ferocidad de 2.
- Las mujeres cocineras tienen una ferocidad de 4 (porque manejan cuchillos).
- Los piratas bravos tienen una ferocidad de 5.
- Los piratas cobardes, que son aquellos que no son bravos, tienen una ferocidad de 0.

Sólo puede haber mujeres o piratas: los hombres comunes no son del interés del capitán. Pero no es sólo lo anterior lo que se requiere, ya que a *Barbazul* solo le interesan las mujeres bonitas o cocineras, los piratas bravos y los piratas de más de 40 años. **¿Qué tripulaciones podría formar *Barbazul*?**

TEG

[forall, generación]

Se tiene la siguiente base de conocimientos que modela el estado actual de un tablero de TEG (sí, faltan 34 países...)

```
continente(americaDelSur). continente(americaDelNorte).
continente(asia). continente(oceania).
```

```
paisContinente(americaDelSur, argentina). paisContinente(americaDelSur,
brasil).
paisContinente(americaDelSur, chile). paisContinente(americaDelSur, uruguay).
paisContinente(americaDelNorte, alaska). paisContinente(americaDelNorte,
yukon).
paisContinente(americaDelNorte, canada). paisContinente(americaDelNorte,
oregon). paisContinente(asia, kamtchatka). paisContinente(asia, china).
paisContinente(asia, siberia). paisContinente(asia, japon).
paisContinente(oceania, australia). paisContinente(oceania, sumatra).
paisContinente(oceania, java). paisContinente(oceania, borneo).
```

```
limitrofes(argentina, brasil). limitrofes(argentina, chile).
limitrofes(argentina, uruguay). limitrofes(uruguay, brasil).
limitrofes(alaska, kamtchatka). limitrofes(alaska, yukon).
limitrofes(canada, yukon). limitrofes(alaska, oregon).
limitrofes(canada, oregon). limitrofes(siberia, kamtchatka).
limitrofes(siberia, china). limitrofes(china, kamtchatka).
limitrofes(japon, china). limitrofes(japon, kamtchatka).
limitrofes(australia, sumatra). limitrofes(australia, java).
limitrofes(australia, borneo). limitrofes(australia, chile).
```

```
% Usar este para saber si son limitrofes ya que es una relacion simetrica
sonLimitrofes(X, Y) :- limitrofes(X, Y).
sonLimitrofes(X, Y) :- limitrofes(Y, X).
```

```
jugador(amarillo).
jugador(magenta).
```


jugador(negro) .

ocupa(argentina, magenta) . ocupa(chile, negro) . ocupa(brasil, amarillo) .
ocupa(uruguay, magenta) . ocupa(alaska, amarillo) . ocupa(yukon, amarillo) .
ocupa(canada, amarillo) . ocupa(oregon, amarillo) . ocupa(kamtchatka, negro) .
ocupa(china, amarillo) . ocupa(siberia, amarillo) . ocupa(japon, amarillo) .
ocupa(australia, negro) . ocupa(sumatra, negro) . ocupa(java, negro) .
ocupa(borneo, negro) .

Se pide definir los siguientes predicados de modo que sean completamente inversibles:

estaPeleado/1 que se cumple para los continentes en los cuales todos los jugadores ocupan algún país.

ocupaContinente/2 que relaciona un jugador y un continente si el jugador ocupa todos los países del mismo.

seAtrinchero/1 que se cumple para los jugadores que ocupan países en un único continente.

estaEnElHorno/1 que se cumple para los países cuyos limítrofes están todos ocupados por otros jugadores.

Gustos cinéfilos

[conjunción, disyunción, negación, universo cerrado, consultas]

Se quiere modelar los gustos cinéfilos de María, Juan y José. Se sabe que a Juan le copan las películas de acción y en las que actúa Jean Renó, que a María le copan los dramas de DiCaprio y Forrest Gump pero no El Profesional, y a José le copan los dramas en los que no trabaja DiCaprio.

actua(diCaprio,titanic) .
actua(diCaprio,atrapameSiPuedes) .
actua(diCaprio,gilbertGrape) .
actua(jeanReno,elProfesional) .

genero(titanic,drama) .
genero(gilbertGrape,drama) .
genero(atrapameSiPuedes,comedia) .
genero(ironMan,accion) .
genero(rapidoYFurioso,accion) .
genero(elProfesional,drama) .

1. Agregar las cláusulas que correspondan para representar los gustos de María, Juan y José.

2. Qué consultas se deben realizar para saber:

¿Qué películas le copan a Juan? ¿Existe alguien a quien le cope Gilbert Grape? ¿Existe alguna película que les cope a María y a José?

El último teorema de Fermat

[negación, forall, generación, aritmética]

“Demostrar” el último teorema de Fermat:

" Para todo entero $n > 2$ no existen A, B y C enteros tal que $A^n + B^n = C^n$ "

Tip: Considerar los enteros acotados, por ejemplo entre -1000 y 1000.

Tip: El predicado tiene aridad 0.

```
?- fermatTeniaRazon.  
true.
```

Pdep Smartphones

[Ejercicio integrador]

Viendo como crece en la actualidad el mercado de las aplicaciones de celulares, a un grupo de iluminados se les ocurrió que podrían aprovechar la moda. Para esto nos encargaron un sistema que los ayude en sus investigaciones de mercado.

Se conoce por cada smartphone el sistema operativo con el cual es compatible. Cada smartphone esta representado por un functor `smartphone(marca, modelo)`.

```
%compatible(smartphone(marca, modelo), sistemaOperativo)  
compatible(smartphone(nokia, lumia710), windowsPhone).  
compatible(smartphone(nokia, n8), nokiaBelle).  
compatible(smartphone(motorola, xT615), android(2.37,[market, flash,  
multitask])).  
compatible(smartphone(motorola, backflip), android(1.6, [market])).  
compatible(smartphone(motorola, razr), android(4.12, [market, flash,  
multitask])).  
compatible(smartphone(samsung, galaxyS4), android(4.22, [market, flash,  
multitask])).  
compatible(smartphone(samsung, galaxySAdvance), android(4.22, [market, flash,  
multitask])).  
compatible(smartphone(sonyEricsson, vivaz), symbian(S60)).  
compatible(smartphone(sonyEricsson, x8), android(2.1, [market, multitask])).  
compatible(smartphone(mac, iphone4), iOS([market, multitask, itunes]))
```

Y se conocen los precios de cada uno de los modelos así:

```
%precio(modelo, precio).  
precio(galaxyS4, 3100).
```

```

precio(galaxySAdvance,2500).
precio(lumia710,1500).
precio(n8,1600).
precio(XT615,2000).
precio(backflip,900).
precio(vivaz,1100).
precio(x8,1200).
precio(iphone4,3000).
precio(razr, 2500).

```

Sabemos que hay más smartphones en el mercado, pero con esta muestra nos alcanza para saber.

Entrada en calor

1) Definir los predicados que permitan conocer:

- a) Qué empresa es **innovadora**, esto es si tiene celulares de distintos sistemas operativos
- b) Qué empresa es **insistente** esto es si tiene por lo menos 2 celulares con el mismo sistema operativo.
- c) Qué empresa es **marketinera** esto es si es innovadora y además es insistente.

Las aplicaciones y sus requerimientos..

2) Definir el predicado cumpleRequisito/2, que relaciona una aplicación con el sistema operativo en donde funciona. Sabiendo que:

- El **mapaLoco**, funciona en nokiaBelle, en symbian S60 y en todos los iOS.
- El **aceleradorDeDescargas**, funciona en celulares que tengan market, y para windowPhone..
- El **haskellIMobile** requiere que el celular tenga multitask.

Definir los siguientes predicados totalmente inversibles. (Excepto que diga lo contrario).

3) **correAplicacion/2** relaciona un smartphone con una aplicación, si cumple con lo que esa aplicación requiere.

Ejemplo

```

?- correAplicacion(Smartphone, Aplicacion).
Smartphone = smartphone(nokia, n8);
Aplicacion = mapaLoco;

```

4) **funcionaEn/2**, relaciona una aplicación con un conjunto de smartphones en la que la puede correr.

5) Conocer si una marca **esCopada/1**, esto se verifica si es marketinera y además en todos sus celulares corren al menos 2 aplicaciones

6) Si un sistema operativo es **elDesterrado/1**, se verifica si no puede correr ninguna aplicación (sin usar forall/2)

7) **mayorCantidadDeModelos/1**, conocer la/s marca/s de smartphone que tengan la mayor cantidad de modelos en el mercado. (**Definirla con y sin recursividad**).

```
?-mayorCantidadDeModelos(Marca).
```

```
Marca = motorola;
```

8) **nombresSistemasOperativos/1**, conocer un conjunto de nombres de los sistemas operativos que están en el mercado. (En el conjunto de nombre de marcas no debe haber repetidos).

```
?-nombresSistemasOperativos(Nombres).
```

```
Nombres = [windowsPhone, nokiaBelle, android, symbian, iOS]
```

9) **correEn/2**, relaciona un nombre de sistema operativo, con un conjunto de marcas de celulares donde puede correr. (En el conjunto de marcas no debe haber repetidos).

10) **compatibleConTodosExceptoUno/1**, conocer la marca de smartphone que tiene solo un teléfono que no es compatible con android. El resto de los teléfonos son todos compatibles con android.

Pueden estar conectados..

11) **puedenConectarse/1**, una persona puede estar conectada, si tiene un smartphone o es conocido de alguna persona que tiene smartphone o algún conocido de sus conocidos tiene acceso.

```
%tiene(Persona, Smartphone).  
tiene(mariano, smartphone(mac, iphone4)).  
tiene(rodrigo, smartphone(samsung, galaxyS4)).  
conocidos(hernan,mariano).  
conocidos(juan, hernan).  
conocidos(martina,pepe).
```

En el ejemplo mariano, y rodrigo pueden estar conectados, porque tienen smartphone también hernán puede estar conectado porque es conocido de mariano que esta conectado, además juan también puede estar conectado porque es conocido de hernán que puede estar conectado.

Todas las combinaciones de smartphones posibles

12) **smartphonesPosibles/2**, relaciona una cantidad de dinero disponible, con las distintas alternativas de smartPhones que se pueden comprar con dicha cantidad. En una misma compra se puede comprar más de un celular. (No es inversible por el primer argumento).

13) ¿ En qué predicados uso polimorfismo? ¿Para qué fue útil?

Dioses Griegos

[Ejercicio integrador]

En un lugar muy remoto (muy), en la Galaxia Griggian, existen dioses que reinan y someten a los mortales mirando las acciones que realizan. Nos contrataron para hacer un programa que estudie el comportamiento de este lugar. Como le vimos relaciones y condiciones por todos lados, nos jugamos a hacerlo en el paradigma lógico.

La base de conocimiento inicial incluye esta información acerca de los grigianos o grigios. Tenemos el censo de mortales, y sabemos qué mortal adora a qué dios:

```
% mortal(Mortal) .  
% adora(Mortal, Dios) .
```

También contamos con la información básica acerca de cada dios; existen dioses buenos y malos; y los modelamos con los respectivos predicados:

```
% diosBueno(Nombre, MortalFavorito, NivelDivinidad) .  
% diosMalo(Nombre, DiosesOdiados, NivelDivinidad) .  
% hijoDivino(DiosPadre, DiosHijo) .
```

Ambas clases de dioses tienen un Nivel de Divinidad, a mayor nivel mas groso es el dios. Cada uno de los dioses buenos tiene un mortal favorito, y cada uno de los dioses malos tiene una lista de Dioses a los que odia.

De muchos dioses sabemos quién es su dios padre, para eso tenemos el predicado *hijoDivino/2*.

Los mortales realizan acciones que pueden ser de interés para los dioses, las incluimos mediante el predicado *hizo(Mortal, Accion)*.

Los dioses premian (con regalos) o castigan distintas acciones, lo vamos a representar así:

```
% don(Dios, regalo(Accion,Regalo)) .  
% don(Dios, castigo(Accion,Castigo)) .
```

La siguiente es una base de conocimiento inicial de ejemplo

```
diosMalo(urano, [apolo], 60) .  
diosMalo(kronos, [zeus, urano, afrodita], 90) .  
diosMalo(hades, [kronos, apolo], 400) .  
  
diosBueno(zeus, hercules, 500) .  
diosBueno(afrodita, enneas, 50) .  
diosBueno(apolo, creso, 80) .  
  
hijoDivino(afrodita, apolo) .
```

hijoDivino(zeus,afrodita).
hijoDivino(kronos,zeus).
hijoDivino(kronos,urano).

mortal(hercules).
mortal(eneas).
mortal(teseo).
mortal(creso).
mortal(aquiles).

adora(hercules,zeus).
adora(eneas,zeus).
adora(eneas,hades).
adora(teseo,apolo).
adora(teseo,afrodita).
adora(aquiles,urano).
adora(aquiles,kronos).

don(zeus,regalo(hacerOfrenda,placer)).
don(zeus,regalo(pensar,vino)).
don(zeus,castigo(maldecir,rayo)).
don(afrodita,regalo(enamorarse,vino)).
don(afrodita,regalo(pensar,oro)).
don(urano,castigo(ignorar,sequa)).
don(apolo,regalo(pensar,vino)).
don(apolo,regalo(construir,salud)).

accion(eneas,pensar).
accion(hercules,maldecir).
accion(teseo,maldecir).
accion(aquiles,ignorar).
accion(eneas,ignorar).
accion(hercules,construir).

Se pide definir los siguientes predicados:

1. Status religioso de un mortal, mediante tres predicados que deben ser inversibles.
 - a. **monoteista/1** : Un mortal es monoteista si adora exactamente a un dios.
 - b. **ateo/1** : Un mortal es ateo si no adora a ningun dios
 - c. **indeciso/1** : Un mortal es indeciso si adora al menos a un dios bueno y también adora al menos a un dios malo.

2. Castigos y regalos

- a. **diosLoco/1**: un dios se considera loco si siendo bueno tiene castigos entre sus dones, o si siendo malo tiene regalos entre sus dones.
- b. **regaloDivino/3** : relaciona un dios, un mortal y cada regalo que el dios le hace al mortal, de acuerdo a estas reglas:
 - Si un mortal que adora a un dios realiza una acción que ese dios premia con un regalo, el mortal se ganó ese regalo.
 - Cada dios bueno le da a su mortal favorito todos los regalos que él sabe dar, o sea, que tiene asociados a alguna acción, sin importar qué acciones haya hecho o dejado de hacer el favorito.
- c. **castigoDivino/3**: relaciona un dios, un mortal y cada castigo con que el dios somete al mortal, de acuerdo a estas reglas:
 - Si un mortal realiza una acción que para un dios es merecedora de un castigo, y el mortal no adora a ese dios, entonces se ganó el castigo.
 - Cada dios malo castiga a cada mortal que adora a un dios que él odia con la culpa, que es algo así como el peor de los castigos.

3. Análisis de los mortales

- a. **mimado/2**: relaciona dios y mortal. Cada dios mima a su mortal favorito, y también a los mortales favoritos de sus ancestros (padre, abuelo, etc.).
- b. **tiendeALoMaligno/1**: un mortal tiende a lo maligno si se cumple alguna de estas condiciones: es merecedor de al menos 4 castigos divinos, adora a un dios malo que tiene un nivel de divinidad mayor a 100, o adora únicamente a dioses malos sin ser ni ateo ni monoteísta (lo de ateo o monoteísta afecta sólo a la última condición). Este predicado debe ser inversible.

4. Bendiciones y santidad

- a. **puedeBendecir/2**: Un mortal puede bendecir a otro si la suma del nivel de divinidad de los dioses buenos que adora el primero es mayor a la suma del nivel de divinidad de los dioses malos que adora el segundo. Este predicado debe ser inversible
- b. **esUnSanto/1**: Los santos son los mortales que pueden bendecir a cualquier otro mortal, no es necesario que puedan bendecirse a sí mismos.

5. Análisis de acciones

- a. **accionConveniente/1**: Una acción es conveniente si todos los dioses buenos la premian con regalos y ningún dios malo la castiga.
- b. **accionMistica/1**: Una acción es mística si es premiada con regalos por exactamente 3 dioses, o por al menos un dios de nivel de divinidad 80.

6. **diosQuisquilloso/1**: un dios es quisquilloso si es malo y todos los dioses que ocupan una posición impar en su lista de dioses odiados (el primero, el tercero, el quinto, etc.) regalan vino (no importa por qué acción.regala vino cada dios odiado impar).

Viscosos pero Sabrosos

[Ejercicio integrador]

En la jungla tan imponente el león rey duerme ya...

Y Timón y Pumba salieron a lastrar bichos.

Tenemos tres tipos de bichos, representados por functores: las vaquitas de San Antonio (de quienes nos interesa un peso), las cucarachas (de quienes nos

interesa un tamaño y un peso) y las hormigas, que pesan

siempre lo mismo. De los personajes también se conoce el peso, mediante hechos.



La base de conocimiento es la que sigue:

```
%comio(Personaje, Bicho)
comio(pumba, vaquitaSanAntonio(gervasia,3)).
comio(pumba, hormiga(federica)).
comio(pumba, hormiga(tuNoEresLaReina)).
comio(pumba, cucaracha(ginger,15,6)).
comio(pumba, cucaracha(erikElRojo,25,70)).
comio(timon, vaquitaSanAntonio(romualda,4)).
comio(timon, cucaracha(gimeno,12,8)).
comio(timon, cucaracha(cucurucha,12,5)).
comio(simba, vaquitaSanAntonio(remeditos,4)).
comio(simba, hormiga(schwartzenegger)).
comio(simba, hormiga(niato)).
comio(simba, hormiga(lula)).
```

```
pesoHormiga(2).
```

```
%peso(Personaje, Peso)
peso(pumba, 100).
peso(timon, 50).
peso(simba, 200).
```



1) A falta de pochoclos...

Definir los predicados que permitan saber:

a) Qué cucaracha es jugosita: ó sea, hay otra con su mismo tamaño pero ella es más gordita.

```
?- jugosita(cucaracha(gimeno,12,8)).
```

Yes

b) Si un personaje es hormigofílico... (Comió al menos dos hormigas).

?- hormigofilico(X) .

X = pumba;

X = simba.

c) Si un personaje es cucarachofóbico (no comió cucarachas).

?- cucarachofobico(X) .

X = simba

d) Conocer al conjunto de los picarones. Un personaje es picarón si comió una cucaracha jugosita ó si se come a Reditos la vaquita. Además, pumba es picarón de por sí.

?- picarones(L) .

L = [pumba, timon, simba]

2) Pero yo quiero carne...

Aparece en escena el malvado Scar, que persigue a algunos de nuestros amigos. Y a su vez, las hienas Shenzi y Banzai también se divierten...

```
persigue(scar, timon).  
persigue(scar, pumba).  
persigue(shenzi, simba).  
persigue(shenzi, scar).  
persigue(banzai, timon)
```



Por ejemplo, un día había una hiena distraída y con mucho hambre y amplió su dieta
`comio(shenzi,hormiga(conCaraDeSimba)) .`

Completando la base...

`peso(scar, 300) .`

`peso(shenzi, 400) .`

`peso(banzai, 500) .`

a) Se quiere saber cuánto engorda un personaje (sabiendo que engorda una cantidad igual a la suma de los pesos de todos los bichos en su menú). Los bichos no engordan.

?- cuantoEngorda(Personaje, Peso) .

Personaje= pumba

Peso = 83;

Personaje= timon

```
Peso = 17;
Personaje= simba
Peso = 10
```

- b) Pero como indica la ley de la selva, cuando un personaje persigue a otro, se lo termina comiendo, y por lo tanto también engorda. Realizar una nueva version del predicado cuantoEngorda.

```
?- cuantoEngorda(scar,Peso).
    Peso = 150
    (es la suma de lo que pesan pumba y timon)
```

```
?- cuantoEngorda(shenzi,Peso).
    Peso = 502
    (es la suma del peso de scar y simba, mas 2 que pesa la
hormiga)
```

- c) Ahora se complica el asunto, porque en realidad cada animal antes de comerse a sus víctimas espera a que estas se alimenten. De esta manera, lo que engorda un animal no es sólo el peso original de sus víctimas, sino también hay que tener en cuenta lo que éstas comieron y por lo tanto engordaron. Hacer una última version del predicado.

```
?- cuantoEngorda(scar,Peso).
    Peso = 250
    (150, que era la suma de lo que pesan pumba y timon, más
83 que se come pumba y 17 que come timon )
```

```
?- cuantoEngorda(shenzi,Peso).
    Peso = 762
    (502 era la suma del peso de scar y simba, mas 2 de la
hormiga. A eso se le suman los 250 de todo lo que engorda scar y
10 que engorda simba)
```

3) Buscando el rey...

Sabiendo que todo animal adora a todo lo que no se lo come o no lo persigue, encontrar al rey. El rey es el animal a quien sólo hay un animal que lo persigue y todos adoran.

Si se agrega el hecho:

```
persigue(scar, mufasa).
```

```
?- rey(R).
R = mufasa.
(sólo lo persigue scar y todos los adoran)
```

