

Paradigma lógico

Profesión y pareja

Para los primeros pasos en el paradigma lógico, con Prolog, les proponemos agrupar una serie de definiciones (hechos y reglas lógicas) sobre lo que cada estudiante considera respecto a dos aspectos importantes de la vida.

¡Mirá lo que hicieron tus compañeros, agregá tu solución, copiatelo y probalo!

1) ¿Qué es un buen ingeniero?

Ejemplo de consulta:

buenIngeniero(Alguien).

Soluciones

%% La que vimos en clase

```
estudia(marcos, wollok).
estudia(julian, wollok).
estudia(maria, prolog,70).
estudia(juan, haskell).

horasDe(haskell, 5).
horasDe(wollok, 30).
horasDe(prolog, 100).

lenguaje(haskell).
lenguaje(wollok).
lenguaje(prolog).

buenIngeniero(Alguien):-
    estudia(Alguien, Lenguaje),
    lenguaje(Lenguaje),
    horasDe(Lenguaje, Horas), Horas > 10.
```

%% Otra que vimos en clase

```
estudioso(marta).
estudioso(marcelo).
```

```
seCopia(marta).
seCopia(ernesto).

buenIngeniero(Alguien):-
    estudioso(Alguien),
    not(seCopia(Alguien)).
```

```
% ¿Es buen/a ingeniero/a?
%Ficha de Estudios:
%fichaEstudios(nombre,profesion,nivelDeConocimientos,habilidad,accionesMalas).

fichaEstudios(abril,ingeniero,90,baja).
fichaEstudios(diego,ingeniero,100,alta).
fichaEstudios(hector,ingeniero,99,media).
fichaEstudios(lucia,medico,20,alta).
fichaEstudios(marcelo,ingeniero,10,media).
fichaEstudios(santiago,ingeniero,89,alta).
fichaEstudios(yamila,maestra,50,alta).
fichaEstudios(zoe,ingeniero,70,alta).

%Tolerancia de las acciones malas:

esBuenIngeniero(Alguien):-
    fichaEstudios(Alguien,ingeniero,NivelDeConocimientos,Habilidad),
    amplioConocimiento(NivelDeConocimientos),
    eshabil(Habilidad).

amplioConocimiento(Conocimiento):-
    Conocimiento>80.

eshabil(alta).
eshabil(muyalta).
```

```
estudioso(maria).
estudioso(marcos).

noSeCopia(marcos).
noSeCopia(jose).

buenIngeniero(Alguien):-
```

```
estudioso(Alguien),  
noSeCopia(Alguien).
```

```
esEstudioso(fabi).  
esEstudioso(joni).
```

```
sabe(fabi,matematica).  
sabe(joni,fisica).  
sabe(fabi,algebra).  
sabe(joni,algebra).
```

```
edad(fabi,25).
```

```
esMateria(algebra).  
esMateria(fisica).
```

```
buenIngeniero(Persona):- esEstudioso(Persona),  
                           sabe(Persona,Materia),  
                           esMateria(Materia),  
                           edad(Persona,Edad),  
                           Edad > 24.
```

```
%% ejemplo ingenieros
```

```
esIngeniero(maria).  
esIngeniero(jose).  
esIngeniero(juan).
```

```
programaEn(wollok,maria).  
programaEn(python,juan).
```

```
lenguaje(python).  
lenguaje(wollok).
```

```
buenIngeniero(Alguien):-  
    programaEn(Lenguaje, Alguien),  
    lenguaje(Lenguaje),  
    esIngeniero(Alguien).
```

2) Condiciones o circunstancias en las cuales cada uno o cada una invita a salir a alguien.

Ejemplo de consulta:

invitaASalir(Persona,Invitado).

Soluciones

```
seCopia(marcos).
seCopia(diego).
seCopia(juan).
seCopia(lucia).

prestaAtencion(marcos).
prestaAtencion(maria).
prestaAtencion(diego).
prestaAtencion(belen).
prestaAtencion(lucia).

conocimientos(marcos,fisica).
conocimientos(marcos,matematica).
conocimientos(maria,wolLok).
conocimientos(maria,fisica).
conocimientos(diego,haskell).
conocimientos(diego,proLog).
conocimientos(lucia,css).
conocimientos(lucia,wolLok).

esLenguajeProg(wolLok).
esLenguajeProg(haskell).
esLenguajeProg(proLog).

sabeLenguajeProg(Alguien):-
    conocimientos(Alguien,Conocimiento),
    esLenguajeProg(Conocimiento).

buenIngeniero(Alguien):-
    not(seCopia(Alguien)),
    prestaAtencion(Alguien),
    sabeLenguajeProg(Alguien).

LeGustaIrALCine(juan).
```

```
leGustaIrALCine(romina).
leGustaIrALCine(maria).
leGustaIrALCine(diego).
leGustaIrALCine(marta).

gustaDe(juan,romina).
gustaDe(maria, diego).
gustaDe(maria,juan).
gustaDe(romina,maria).
gustaDe(romina,juan).
gustaDe(diego,maria).
gustaDe(jorge,marta).
gustaDe(jorge,marta).
gustaDe(marta,jorge).
gustaDe(marta,diego).

tienePareja(maria).
tienePareja(diego).
tienePareja(romina).
tienePareja(jorge).
tienePareja(marta).

invitaASalirA(Alguien,Otro):-
    leGustaIrALCine(Alguien),
    leGustaIrALCine(Otro),
    gustaDe(Alguien,Otro),
    not(tienePareja(Alguien)),
    not(tienePareja(Otro)).
```

```
%% Ejemplo parejas
```

```
seGustan(maria,jose).
seGustan(sol,pablo).

esDivertida(sol).
esDivertida(maria).
```

```
esGracioso(jose).
esGracioso(pablo).

tieneBarbijo(sol).
tieneBarbijo(maria).
tieneBarbijo(pablo).

invitaASalir(Alguien, Otro):-
    seGustan(Alguien, Otro),
    esDivertida(Alguien),
    esGracioso(Otro),
    tieneBarbijo(Alguien),
    tieneBarbijo(Otro).
```

```
%% materia(Materia,NotaParcialMinima,NotaParcialMaxima)<-
%% es buen Ingeniero si esa materia se considera las notas necesarias aprobadas
con 6 o más
materia(fisica).
materia(quimica).
materia(arquitectura).
materia( analisis,5,10).
materia( paradigmas,5,10).
materia( sintaxis,5,10).
materia( analisisDeSistemas,5,10).
materia( algebra,5,10).
materia( discreta,5,10).
% materia( ecologia,7,20).

%notaMinimaAprobacion(5).
%notaMaximaValida(10).

notasLimite(5,10).

buenIngeniero(Alguien):-
    estudia(Alguien,Nota1,Nota2,Materia),
    % materia(Materia),
    notasLimite(NMin,NMax),
    notaValida(Nota1,NMin,NMax),
    notaValida(Nota2,NMin,NMax).

notaValida(Nota,Min,Max):-
```

```

Nota>Min,
Nota=<Max.

% notaValida(Nota,Min,Max):- between(Nota,Min,Max).

estudia(marcos,1,6,fisica).
estudia(maria,5,6,analisis).
estudia(marcos,4,2,paradigmas).
estudia(maria,10,10,sintaxis).
% estudia(marcos,6,6,corteYConfeccion).

```

--- --- ---

```

%PROFESION Y PAREJA:

% PAREJAS <3.
invitaASalir(UnaPersona,OtraPersona):-
    fichaEstudios(UnaPersona,_,_,_,AccionesMalas),
    fichaEstudios(OtraPersona,_,_,_,AccionesMalasOtraPersona),
    compatibilidad(AccionesMalas,AccionesMalasOtraPersona).

compatibilidad(Acciones,AccionesOtraPersona):-
    tolerancia(Tolerancia,Acciones),
    tolerancia(Tolerancia,AccionesOtraPersona).

```

--- --- ---

```

sabeProlog(juan).
sabeProlog(maria).

tienePensamientoCritico(juan).
tienePensamientoCritico(maria).

tieneExperiencia(juan).
tieneExperiencia(marcos).

buenIngeniero(Alguien):-
    sabeProlog(Alguien),
    tienePensamientoCritico(Alguien),
    tieneExperiencia(Alguien).

```

```
leGustaA(juan,leticia).
leGustaA(leticia,francisco).
leGustaA(francisco,juan).

tieneDinero(juan).
tieneDinero(leticia).

invitaASalirA(Alguien,Otro):-
    leGustaA(Alguien,Otro),
    tieneDinero(Alguien).
```

```
leGusta(agustin,carla).
leGusta(sofia,julian).
```

```
vanAlCine(agustin,carla).
vanAlCine(sofia,julian).
```

```
tieneDinero(agustin).
tieneDinero(carla).
tieneDinero(sofia).
```

```
edad(agustin,20).
edad(carla,19).
edad(sofia,20).
edad(julian,21).
```

```
invitaASalir(Persona,OtraPersona):- leGusta(Persona,OtraPersona),
                                     vanAlCine(Persona,OtraPersona),
                                     tieneDinero(Persona),
                                     tieneDinero(OtraPersona),
                                     edad(Persona,EdadPer),
                                     edad(OtraPersona,EdadOtra),
                                     EdadPer > EdadOtra.
```

```
%-----
sale(agustin,carla).
sale(sofia,julian).
```

```
sale(carla,sofia).
sale(agustin,valentina).
```

```
saleConAlMenosDosPersonas(Persona):- sale(Persona,OtraX),
                                         sale(Persona,Segunda),
                                         OtraX\=Segunda.
```

```
-----

%% Invita a salir alguien a otro.
esFlaquito(maria).
esFlaquito(juan).
```

```
tieneAuto(pedro).
tieneAuto(josefina).
```

```
edad(juan , 18).
edad(maria, 25).
edad(josefina, 45).
edad(pedro, 32).
```

```
invitaASalirA(Alguien, Otro):-
    esFlaquito(Alguien),
    esMayor(Alguien),
    tieneAuto(Otro).

esMayor(Alguien):-
    edad(Alguien, Edad), Edad > 18.
```

```
fachero(marcos).
fachero(jose).

mujer(maria).

tienePlata(jose).

invitaASalirA(Alguien,Otro):-
    fachero(Otro),
    tienePlata(Otro),
    mujer(Alguien).
```

3) Mansion Dreadbury

Copia aca tu solucion!!

```
%%%%%%%%% PARTE 1

viveEnLaMansion(tiaAgatha).
viveEnLaMansion(carnicero).
viveEnLaMansion(charles).

odia(tiaAgatha,P):-
    viveEnLaMansion(P),
    P \= carnicero.

odia(carnicero,P):- odia(tiaAgatha,P).

odia(charles,P):- not(odia(tiaAgatha,P)).

esMasRico(P,tiaAgatha):-
    not(odia(carnicero,P)),
    viveEnLaMansion(P).

esElAsesino(Asesino,Victima):-
    viveEnLaMansion(Asesino),
    odia(Asesino,Victima),
    not(esMasRico(Asesino,Victima)).

%?- esElAsesino(X,tiaAgatha).
%X = tiaAgatha .

%%%%%%%%% PARTE 2

odia(P,milhouse):- viveEnLaMansion(P).

% + Prueba en consola

%?- odia(X,milhouse).      %?- odia(charles,X).
%X = carnicero ;          %X = milhouse.
%X = tiaAgatha ;          %?- odia(X,tiaAgatha).
%X = carnicero ;          %X = tiaAgatha ;
```

```

%X = charles.           %X = carnicero ;
                        %false.

%?- odia(X,Y).           %?- odia(carnicero,X).
%X = Y, Y = tiaAgatha ; %X = tiaAgatha ;
%X = tiaAgatha,         %X = charles ;
%Y = charles ;          %X = milhouse ;
%X = carnicero,         %X = milhouse.
%Y = tiaAgatha ;
%X = carnicero,
%Y = charles ;
%X = carnicero,
%Y = milhouse ;
%X = tiaAgatha,
%Y = milhouse ;
%X = carnicero,
%Y = milhouse ;
%X = charles,
%Y = milhouse.

```

```

viveEnLaMansion(tiaAgatha).
viveEnLaMansion(carnicero).
viveEnLaMansion(charles).

odia(charles,Alguien):-
    not(odia(tiaAgatha,Alguien)).

odia(tiaAgatha,Alguien):-
    viveEnLaMansion(Alguien),
    Alguien \= carnicero.

odia(carnicero,Alguien):-
    odia(tiaAgatha,Alguien).

esMasRico(Alguien, tiaAgatha):-
    not(odia(carnicero,Alguien)),
    viveEnLaMansion(Alguien).

mata(Asesino,Victima):-
    odia(Asesino,Victima),
    not(esMasRico(Asesino,Victima)),

```

```
viveEnLaMansion(Asesino).
```

```
%%%%%%%%%%
```

CONSULTAS

```
%%%%%%%%%%
```

```
%Consulta: mata(Alguien,tiaAgatha).
```

```
%Respuesta: tiaAgatha
```

```
%Consulta: odia(Alguien,milhouse).
```

```
%Respuesta: Alguien = Charles
```

```
%Consulta: odia(charles,Alguien).
```

```
%Respuesta: False
```

```
%Consulta: odia(tiaAgatha,Alguien).
```

```
%Respuesta: Alguien = tiaAgatha ;
```

```
%           Alguien = charles.
```

```
%Consulta: odia(Alguien,Otro).
```

```
%Respuesta: Alguien = Otro, Otro = tiaAgatha ;
```

```
%           Alguien = tiaAgatha,
```

```
%           Otro = charles ;
```

```
%           Alguien = carnicero,
```

```
%           Otro = tiaAgatha ;
```

```
%           Alguien = carnicero,
```

```
%           Otro = charles.
```

```
%Consulta: odia(carnicero,_).
```

```
%Respuesta: True
```

```
% Parte 1
```

```
viveEnLaMansion(agatha).  
viveEnLaMansion(charles).  
viveEnLaMansion(carnicero).
```

```
odiaA(charles,Persona):-  
    viveEnLaMansion(Persona),  
    not(odiaA(agatha,Persona)).
```

```
odiaA(agatha,Persona):-  
    viveEnLaMansion(Persona),  
    Persona\=carnicero.
```

```
odiaA(carnicero,Persona):-  
    odiaA(agatha, Persona).
```

```
masRicoQue(Persona,agatha):-  
    viveEnLaMansion(Persona),  
    not(odiaA(carnicero,Persona)).
```

```
asesino(Asesino,Persona):-  
    odiaA(Asesino,Persona),  
    viveEnLaMansion(Asesino),  
    not(masRicoQue(Asesino,Persona)).
```

```
%% asesino(Asesino,agatha).  
%% Asesino = agatha
```

```
% Parte 2
```

```
% anadie quiere a milhouse
```

```
% odiaA(Persona,milhouse).  
% false
```

```
% a quien odia charles
```

```
%odiaA(charles,Persona).  
%Persona = carnicero.
```

```
% quien odia a agatha

% odiaA(Persona,agatha).
%Persona = agatha ;
%Persona = carnicero.

%a quien odia el carnicero

%odiaA(carnicero,Persona).
%Persona = agatha ;
%Persona = charles ;

%odiarores y odiados

%odiaA(Odiador,Odiado).
%Odiador = charles,
%Odiado = carnicero ;
%Odiador = Odiado, Odiado = agatha ;
%Odiador = agatha,
%Odiado = charles ;
%Odiador = carnicero,
%Odiado = agatha ;
%Odiador = carnicero,
%Odiado = charles ;

%% hice que la persona no se pueda odiar a ella misma así que no va a funcionar
(ya lo arregle y funciona).

persona(tiaAgatha).
persona(elCarnicero).
persona(charles).
persona(milhouse).

vivieEnDreadbury(tiaAgatha).
vivieEnDreadbury(elCarnicero).
vivieEnDreadbury(charles).

odia(tiaAgatha, Alguien):-
    vivieEnDreadbury(Alguien),
    not(Alguien = elCarnicero),
```

```

    not(Alguien = tiaAgatha).

odia(charles, Alguien):-
    vivieEnDreadbury(Alguien),
    not(odia(tiaAgatha, Alguien)),
    not(Alguien = charles).

odia(elCarnicero, Alguien):-
    odia(tiaAgatha, Alguien).

esMasRicoQue(Alguien, tiaAgatha):-
    not(odia(elCarnicero, Alguien)),
    vivieEnDreadbury(Alguien).

esAscechino(Alguien, Victima):-
    odia(Alguien, Victima),
    not(esMasRicoQue(Alguien, tiaAgatha)),
    vivieEnDreadbury(Victima).

%% 1.
%% esAscechino(Alguien, Victima).
%%     esAscechino(Alguien, tiaAgatha).
%%     Alguien = charles ;
%%     false.

%% 2.b
%% odia(Alguien, milhouse).
%%     false.

%% odia(charles, Alguien).
%%     Alguien = tiaAgatha ;
%%     Alguien = elCarnicero ;
%%     false.

%% odia(Alguien, tiaAgatha).
%%     Alguien = charles ;
%%     false.

%% odia(Alguin, Odiado).
%%     Alguien = tiaAgatha,
%%     Odiado = charles ;
%%     Alguien = charles,

```

```

%%      Odiado = tiaAgatha ;
%%      Alguien = charles,
%%      Odiado = elCarnicero ;
%%      Alguien = elCarnicero,
%%      Odiado = charles.

%%  odia(elCarnicero, Alguien).
%%      Alguien = charles.

```

```

%Hechos, reglas, predicados, generación, not.

%1

%Quien mata es porque odia a su víctima y no es más rico que ella.
%Además, quien mata debe vivir en la mansión Dreadbury.

mata(Asesino,Victima):-
    odia(Asesino,Victima),
    not(masRico(Asesino,Victima)),
    vive(Asesino,Lugar),
    Lugar = mansionDreasbury.

%Tía Agatha, el carnicero y Charles son las únicas personas que viven en la
mansión Dreadbury.

vive(agatha,mansionDreasbury).
vive(carnicero,mansionDreasbury).
vive(charles,mansionDreasbury).
vive(milhouse,mansionDreasbury).

%Agatha odia a todos los que viven en la mansión, excepto al carnicero.
odia(agatha,Alguien):-
    vive(Alguien,mansionDreasbury),
    not(Alguien = carnicero).

%El carnicero odia a las mismas personas que odia tía Agatha.
odia(carnicero,Alguien):-
    odia(agatha,Alguien).

%Charles odia a todas las personas de la mansión que no son odiadas por la tía
Agatha.

```

```

odia(charles,Alguien):-
    vive(Alguien,mansionDreasbury),
    not(odia(agatha,Alguien)).

%Quien no es odiado por el carnicero y vive en la mansión, es más rico que tía
Agatha.
masRico(Asesino,agatha):-
    not(odia(carnicero,Asesino)),
    vive(Asesino,mansionDreasbury).

%El programa debe resolver el problema de quién mató a la tía Agatha.
%Mostrar la consulta utilizada y la respuesta obtenida.

%?- mata(Asesino,agatha).
% Asesino = agatha

%2

%Agregar los mínimos hechos y reglas necesarios para poder consultar:

%Si existe alguien que odie a milhouse.
%?- odia(Alguien,milhouse).
%Alguien = agatha ;
%Alguien = carnicero ;
%false.

%A quién odia charles.
%odia(charles,Alguien).
%Alguien = carnicero ;
%false.

%El nombre de quien odia a agatha.
%?- odia(Alguien,agatha).
%Alguien = agatha ;
%Alguien = carnicero ;
%false.

%Todos los odiadores y sus odiados.
/* ?- odia(Odiador,Odiado).
Odiador = Odiado, Odiado = agatha ;
Odiador = agatha,
Odiado = charles ;

```

```

Odiador = agatha,
Odiado = milhouse ;
Odiador = carnicero,
Odiado = agatha ;
Odiador = carnicero,
Odiado = charles ;
Odiador = carnicero,
Odiado = milhouse ;
Odiador = charles,
Odiado = carnicero ;
false. */

%Si es cierto que el carnicero odia a alguien.
/* ?- odia(carnicero,_).
true */

%Mostrar las consultas utilizadas para conseguir lo anterior, junto con las
respuestas obtenidas.

```

```

/*Tía Agatha, el carnicero y Charles son
las únicas personas que viven en la mansión Dreadbury.*/
%viveEnMansion(tiaAgatha).
viveEnMansion(carnicero).
viveEnMansion(charles).

viveEnMansion(flanders).

odia(tiaAgatha,lisa).
odia(tiaAgatha,bart).
%odia(tiaAgatha,carnicero).

odia(carnicero,milhouse).

odia(homero,flanders).

%odia(charles,milhouse).
odia(carnicero,lisa).

rico(tiaAgatha).
rico(charles).

```

```
/*Quien mata es porque odia a su víctima y no es más rico que ella.  
Además, quien mata debe vivir en la mansión Dreadbury.*/
```

```
mataA(Asesino,Victima):-odia(Asesino,Victima), not(rico(Asesino)),  
viveEnMansion(Asesino).
```

```
/*Charles odia a todas las personas de la mansión que no son  
odiadas por la tía Agatha.*/
```

```
odiadaPorCharles(Persona):-viveEnMansion(Persona),not(odia(tiaAgatha,Pe  
rsona)).
```

```
/*Agatha odia a todos los que viven en la mansión,excepto al  
carnicero.*/
```

```
odiadoPorAgatha(Persona):-viveEnMansion(Persona),not(odia(tiaAgatha,Per  
sona)).
```

```
/*Quien no es odiado por el carnicero y vive en la mansión,  
es más rico que tía Agatha.*/
```

```
esMasRicoQueAgatha(Persona):-not(odia(carnicero,Persona)),viveEnMansion  
(Persona).
```

```
%El carnicero odia a las mismas personas que odia tía Agatha.
```

```
odiadoPorCarnicero(Persona):-odia(carnicero,Persona),odia(tiaAgatha,Otr  
aPersona),Persona==OtraPersona.
```

```
%Consulta1
```

```
?-mataA(Asesino,Victima).  
Asesino = carnicero,  
Victima = milhouse ;  
Asesino = carnicero,  
Victima = lisa.
```

```
%Consulta2
```

```
?-odiadoPorCharles(_).  
true .
```

```
?- odiadoPorCharles(Alguien).  
Alguien = carnicero ;  
Alguien = charles ;  
Alguien = flanders.
```

%Consulta3

```
?- odiadoPorAgatha(_).  
true .
```

```
?- odiadoPorAgatha(Alguien).  
Alguien = carnicero ;  
Alguien = charles ;  
Alguien = flanders.
```

%Consulta4

```
?- odiadoPorCarnicero(_).  
true .
```

```
?- odiadoPorCarnicero(Alguien).  
Alguien = lisa ;  
false.
```

%Consulta5

```
?- esMasRicoQueAgatha(_).  
false.
```

```
?- esMasRicoQueAgatha(Alguien).  
false.
```

```
% a)  
viveEnLaMansion(tiaAgatha).  
viveEnLaMansion(elCarnicero).  
viveEnLaMansion(charles).  
  
esRico(charles).
```

```
odiaA(elCarnicero, charles).
odiaA(charles, tiaAgatha).
odiaA(charles, elCarnicero).
odiaA(tiaAgatha, tiaAgatha).
```

```
quienMato(Alguien):-
    odiaA(tiaAgatha, Alguien),
    not(esRico(Alguien)),
    viveEnLaMansion(Alguien).
```

```
% quienMato(tiaAgatha).
%     true.
% quienMato(elCarnicero).
%     false.
% quienMato(charles).
%     false.
```

```
asesino(Victima,Persona):-
    odiaA(Persona,Victima),
    viveEnMansion(Persona),
    not(esMasRicoQueAgatha(Persona)).
```

```
%%asesino(agatha,X).
```

```
%%X = agatha
```

```
viveEnMansion(agatha).
viveEnMansion(charles).
viveEnMansion(carnicero).
```

```
odiaA(charles,Persona):-
    viveEnMansion(Persona),
    not(odiaA(agatha,Persona)).
```

```
odiaA(agatha,Persona):-
    viveEnMansion(Persona),
    Persona\=carnicero.
```

```
esMasRicoQueAgatha(Persona):-
    viveEnMansion(Persona),
    not(odiaA(carnicero,Persona)).
```

```
odiaA(carnicero,Persona):-odiaA(agatha,Persona).
```

```
viveEnMansion(milhouse).
existeAlguienQueOdieAMilhouse(milhouse):-
    odiaA(Persona,milhouse),
    odiaA(OtraPersona,milhouse),
    Persona\=OtraPersona.
%% odiaA(charles,carnicero).
%%odiaA(agatha,charles).
%%odiaA(agatha,milhouse).
%%odiaA(charles,X).
%%X = carnicero
%%odiaA(agatha,X).
%%X = agatha ;
%%X = charles ;
%%X = milhouse.
%%odiaA(carnicero,X).
%%X = agatha ;
%%X = charles ;
%%X = milhouse.
%%odiaA(carnicero,milhouse).
%%odiaA(carnicero,agatha).
%%odiaA(carnicero,charles).
```