

RESUMEN PARADIGMA LÓGICO  
FACULTAD REGIONAL BUENOS AIRES  
UNIVERSIDAD TECNOLÓGICA NACIONAL

PARADIGMAS DE PROGRAMACIÓN – *K2032*

Trabajos Prácticos

PAZ PORTILLA, José Miguel      2028244  
jpazportilla@frba.utn.edu.ar

1 de diciembre de 2023

# Índice

<b>1. Introducción a Lógico -&gt;Video 10 Youtube</b>	<b>1</b>
1.1. <i>maestro_alumno.pl</i> . . . . .	1
1.2. <i>aritmetica.pl</i> . . . . .	2
1.3. <i>familia.pl</i> . . . . .	2
<b>2. Orden Superior -&gt;Video 11 Youtube</b>	<b>4</b>
2.1. forall vs not ->caso 1 . . . . .	4
2.2. forall vs not ->caso 2 . . . . .	4
2.3. forall vs not ->caso 3 . . . . .	4
2.4. <i>animales.pl</i> . . . . .	4
<b>3. TEG -&gt;Video 12 Youtube</b>	<b>7</b>
3.1. <i>teg.pl</i> . . . . .	7
<b>4. Functores y Polimorfismo -&gt;Video 13 Youtube</b>	<b>10</b>
4.1. <i>libreria.pl</i> . . . . .	10
<b>5. Listas -&gt;Video 14 Youtube</b>	<b>12</b>

---

## 1. Introducción a Lógico -> Video 10 Youtube

En prolog se deben escribir predicados en un archivo con extensión .pl, luego para ejecutarlo desde terminal se debe escribir `swipl nombre_archivo.pl`, para salir `halt.`, para actualizar `make.`, para correr test `run_test`.

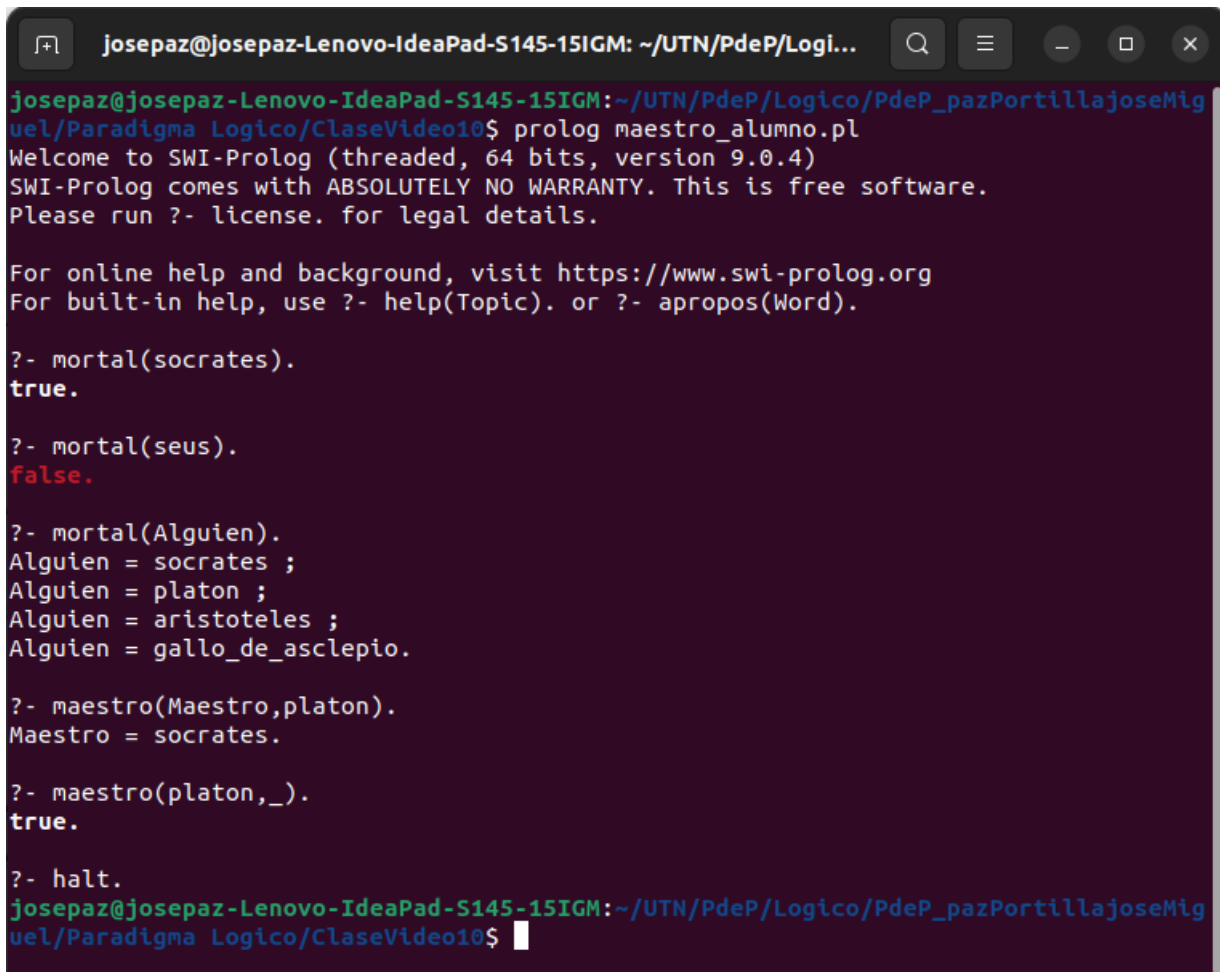
La cantidad de parametros que recibe un predicado se llama aridad. Un predicado puede estar constituido por un hecho o bien por una regla. Donde los hechos no tienen pre requerimiento sino que es algo definido como verdadero, las reglas separan el nombre del predicado con un `:-` y a su derecha los pre requerimientos que debe cumplirse.

Los parametros pueden ser individuos/átomos que empieza con minúscula o bien pueden ser una Variable que empieza con mayúscula.

### 1.1. *maestro\_alumno.pl*

```
1 % humano(Humano)
2 humano(socrates).
3 humano(platon).
4 humano(aristoteles).
5
6 % mortal(Alguien)
7 % Alguien es mortal si es humano o bien si es el gallo de asclepio
8 mortal(Alguien):-
9     humano(Alguien).
10 mortal(gallo_de_asclepio).
11
12 % maestro(Maestro,Alumno)
13 maestro(socrates,platon).
14 maestro(platon,aristoteles).
15 maestro(socrates,jose).
16
17 % groso(Alguien)
18 % Alguien es groso si es maestro de al menos 2 alumnos
19 groso(Alguien):-
20     maestro(Alguien,UnAlumno),
21     maestro(Alguien,OtroAlumno),
22     UnAlumno \= OtroAlumno.
```

Se observa que el predicado humano esta contituido por 3 hechos de aridad 1, el predicado mortal por 1 regla y 1 hecho de aridad 1, el predicado maestro por 2 hechos de aridad 2, el predicado groso por 1 regla de aridad 1.



```

josepaz@josepaz-Lenovo-IdeaPad-S145-15IGM: ~/UTN/PdeP/Logi...
josepaz@josepaz-Lenovo-IdeaPad-S145-15IGM:~/UTN/PdeP/Logico/PdeP_pazPortillajoseMig
uel/Paradigma Logico/ClaseVideo10$ prolog maestro_alumno.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- mortal(socrates).
true.

?- mortal(seus).
false.

?- mortal(Alguien).
Alguien = socrates ;
Alguien = platon ;
Alguien = aristoteles ;
Alguien = gallo_de_asclepio.

?- maestro(Maestro,platon).
Maestro = socrates.

?- maestro(platon,_).
true.

?- halt.
josepaz@josepaz-Lenovo-IdeaPad-S145-15IGM:~/UTN/PdeP/Logico/PdeP_pazPortillajoseMig
uel/Paradigma Logico/ClaseVideo10$

```

**Figura 1.** Consultas a prolog con base de conocimiento maestro\_alumno.pl

## 1.2. *aritmetica.pl*

```

1  % siguiente(Anterior,Siguiente)
2  % El is es inversible a izquierda, es decir siguiente,
3  % pero is no es inversible a derecha.
4  % El + no es inversible, por ende Anterior debe venir ligado,
5  % El 1 ya es un numero, por ende el + lo reconoce.
6  siguiente(Anterior,Siguiente):-
7      % Como el predicado numero es inversible, liga Anterior para hacer al
8      % predicado siguiente inversible
9      numero(Anterior),
10     Siguiente is Anterior + 1.
11
12 % El predicado numero es inversible ya que between lo es.
13 numero(Numero):-
14     between(0,100,Numero).

```

## 1.3. *familia.pl*

```
1  % padre(Padre,Hijo)
2  padre(homero,bart).
3  padre(homero,lisa).
4  padre(homero,maggi).
5  padre(abraham,homero).
6
7  % abuelo(Abuelo,Nieto)
8  abuelo(Abuelo,Nieto):-
9      padre(Padre,Nieto),
10     padre(Abuelo,Padre).
11 % Fry es su propio abuelo.
12 abuelo(fry,fry).
13
14 % hermano(UnHermano,OtroHermano)
15 hermano(UnHermano,OtroHermano):-
16     padre(Padre,UnHermano),
17     padre(Padre,OtroHermano),
18     UnHermano \= OtroHermano.
19
20 % Alguien es ancestro de un descendiente si es su padre, abuelo, bisabuelo
21   , ...
22 % ancestro(Ancestro,Descendiente)
23 ancestro(Ancestro,Descendiente):-
24     padre(Ancestro,Descendiente).
25 ancestro(Ancestro,Descendiente):-
26     padre(Ancestro,Alguien),
27     ancestro(Alguien,Descendiente).
```

---

## 2. Orden Superior -> Video 11 Youtube

Son predicados que como parametro tienen otro predicado.

### 2.1. forall vs not -> caso 1

$$\forall x \in X : P(x) \iff \neg \exists x \in X : \neg P(x) \quad (1)$$

Por ende decir todos los animales son friolentos, equivale a decir que no existe un animal que no sea friolento. En prolog serian:

*forall(animal(Animal), friolento(Animal)).*

y

*not((animal(Animal), not(friolento(Animal)))).*

### 2.2. forall vs not -> caso 2

$$\neg \forall x \in X : P(x) \iff \exists x \in X : \neg P(x) \quad (2)$$

Por ende decir no todos los animales son friolentos, equivale a decir que existen animales que no son friolentos. En prolog serian:

*not(forall(animal(Animal), friolento(Animal))).*

y

*animal(Animal), not(friolento(Animal)).*

### 2.3. forall vs not -> caso 3

$$\forall x \in X : \neg P(x) \iff \neg \exists x \in X : P(x) \quad (3)$$

Por ende decir de todos los animales ninguno es friolento, equivale a decir que no existen animales que sean friolentos. En prolog serian:

*forall(animal(Animal), not(friolento(Animal))).*

y

*not((animal(Animal), friolento(Animal))).*

### 2.4. animales.pl

```
1 | % habitat(Animal,Bioma)
2 | % El predicado habitat es inversible porque permite hacer consultas
   |   existenciales.
3 | habitat(jirafa,sabana).
4 | habitat(tigre,sabana).
5 | habitat(tigre,bosque).
```

```

6  habitat(tiburon,mar).
7
8  acuatico(Animal):-
9      habitat(Animal,mar).
10
11 terrestre(Animal):-
12     % Para volverlo inversible
13     habitat(Animal,_),
14     not(habitat(Animal,mar)).
15
16
17 templado(Bioma):-
18     habitat(_,Bioma),
19     Bioma == sabana.
20 templado(Bioma):-
21     habitat(_,Bioma),
22     Bioma == bosque.
23
24 friolento(Animal):-
25     habitat(Animal,_),
26     forall(habitat(Animal,Bioma),templado(Bioma)).
27
28 % come (Depredador,Presa)
29 come(tigre,jirafa).
30 come(tigre,tiburon).
31
32 % hostil/2 relaciona un animal con un bioma, si todos los animales que
33     viven ahi se lo comen.
34 hostile(Animal,Bioma):-
35     habitat(Animal,_),
36     habitat(_,Bioma),
37     forall(habitat(OtroAnimal,Bioma),come(OtroAnimal,Animal)).
38
39 % terrible/2 relaciona un animal con un bioma si todos los animales que se
40     lo comen, habitan en el bioma.
41 terrible(Animal,Bioma):-
42     habitat(Animal,_),
43     habitat(_,Bioma),
44     forall(come(OtroAnimal,Animal),habitat(OtroAnimal,Bioma)).
45
46 % compatibles/2 relaciona dos animales si ninguno de los dos come al otro
47 compatibles(UnAnimal,OtroAnimal):-
48     habitat(UnAnimal,_),
49     habitat(OtroAnimal,_),
50     not(come(UnAnimal,OtroAnimal)),
51     not(come(OtroAnimal,UnAnimal)).
52
53 % adaptable/1 se cumple para los animales que habitan todos los biomas
54 adaptable(Animal):-
55     habitat(Animal,_),
56     forall(habitat(_,Bioma), habitat(Animal,Bioma)).
57
58 % raro/1 se cumple para todos los animales que habitan un unico bioma
59 raro(Animal):-
60     habitat(Animal,Bioma),
61     not((habitat(Animal,OtroBioma), Bioma \= OtroBioma)).

```

```
60 | % dominante/1 se cumple para los animales que se comen a todos los otros
    |   animales del bioma en el que viven
61 | dominante(Animal):-
62 |   habitat(Animal,Bioma),
63 |   forall((habitat(OtroAnimal,Bioma), Animal \= OtroAnimal),come(Animal,
    |     OtroAnimal)).
```



---

### 3. TEG ->Video 12 Youtube

#### 3.1. *teg.pl*

```
1 jugador(rojo).
2 jugador(azul).
3 jugador(amarillo).
4
5 ubicadoEn(argentina, america_del_sur).
6
7 aliados(rojo, amarillo).
8
9 ocupa(rojo, argentina).
10
11 limitrofes(argentina, brasil).
12
13 % tienePresenciaEn/2 relaciona un jugador con un continente del cual ocupa
14   al menos un pais.
15 tienePresenciaEn(Jugador, Continente):-
16     ocupa(Jugador, Pais),
17     ubicadoEn(Pais, Continente).
18
19 % puedenAtacarse/2 relaciona 2 jugadores si uno ocupa al menos 1 pais
20   limitrofe a algun otro pais ocupado por el otro jugadores
21 puedenAtacarse(UnJugador, OtroJugador):-
22     ocupa(UnJugador, UnPais),
23     ocupa(OtroJugador, OtroPais),
24     limitrofes(UnPais, OtroPais).
25
26 % sinTensiones/2 relaciona 2 jugadores que o bien no pueden atacarse o son
27   aliados
28 sinTensiones(UnJugador, OtroJugador):-
29     aliados(UnJugador, OtroJugador).
30 sinTensiones(UnJugador, OtroJugador):-
31     jugador(UnJugador),
32     jugador(OtroJugador),
33     not(puedenAtacarse(UnJugador, OtroJugador)).
34
35 % perdio/1 se cumple para un jugador que no ocupa ningun pais
36 perdio(Jugador):-
37     jugador(Jugador),
38     % el jugador no ocupa algun pais
39     not(ocupa(Jugador, _)).
40
41 % controla/2 relaciona un jugador con un continente si ocupa todos los
42   paises del continente
43 controla(Jugador, Continente):-
44     jugador(Jugador),
45     ubicadoEn(_, Continente),
46     forall(ubicadoEn(Pais, Continente), ocupa(Jugador, Pais)).
47
48 controla_(Jugador, Continente):-
49     jugador(Jugador),
50     ubicadoEn(_, Continente),
51     not((ubicadoEn(Pais, Continente), not(ocupa(Jugador, Pais))))).
```

```

48
49 % reñido/1 se cumple para los continentes donde todos los jugadores ocupan
      algun pais
50 reñido(Continente):-
51     ubicadoEn(_,Continente),%Para hacerlo inversible
52     forall(
53         jugador(Jugador),%Dominio
54         ( ocupa(Jugador,Pais) , ubicadoEn(Pais,Continente) ) %Tiene 2
              condiciones
55     ).
56
57 %No hay un jugador que no ocupe ningun pais del continente
58 reñido_(Continente):-
59     ubicadoEn(_,Continente),
60     not(
61         ( jugador(Jugador),
62           not( ( ocupa(Jugador,Pais),
63                ubicadoEn(Pais,Continente) ) )
64         )
65     ).
66
67 reñido__(Continente):-
68     ubicadoEn(_,Continente),%Para hacerlo inversible
69     forall(
70         jugador(Jugador),%Dominio
71         tienePresenciaEn(Jugador,Continente) %Tiene 1 condicion
72     ).
73
74 reñido___(Continente):-
75     ubicadoEn(_,Continente),
76     not(
77         (
78             jugador(Jugador),
79             not( tienePresenciaEn(Jugador,Continente) )
80         )
81     ).
82
83 % atrincherado/1 se cumple para jugadores que ocupan paises en un unico
      continente
84 atrincherado(Jugador):-
85     ubicadoEn(_,Continente),
86     ocupa(Jugador,_),
87     forall(ocupa(Jugador,Pais),ubicadoEn(Pais,Continente)).
88
89 atrincherado_(Jugador):-
90     ubicadoEn(_,Continente),
91     ocupa(Jugador,_),
92     not(
93         (
94             ocupa(Jugador,Pais) ,
95             not( ubicadoEn(Pais,Continente) )
96         )
97     ).
98
99 % puedeConquistar/2 Relaciona un jugador con un continente si no lo
      controla, pero todos los paises del continente

```

```

100 % que le falta ocupar son limitrofes a alguno de los paises que si ocupa y
      pertenece a alguien que no es su aliado.
101 puedeConquistar(Jugador,Continente):-
102     jugador(Jugador),
103     ubicadoEn(_,Continente),
104     not(controla(Jugador,Continente)),
105     forall(
106         ( ubicadoEn(Pais,Continente) , not(ocupa(Jugador,Pais)) ),%
              Universo
107         puedeAtacar(Jugador,Pais) % Condicion
108     ).
109
110 puedeConquistar(Jugador,Continente):-
111     jugador(Jugador),
112     ubicadoEn(_,Continente),
113     not(controla(Jugador,Continente)),
114     not(
115         (( ubicadoEn(Pais,Continente) , not(ocupa(Jugador,Pais)) ),%
              Universo
116         not(puedeAtacar(Jugador,Pais))) % Condicion
117     ).
118
119 % puedeAtacar/2 el pais atacado debe ser limitrofe a alguno de los paises
      que el jugador ocupa
120 % y pertenece a alguien que no es su aliado.
121 puedeAtacar(Jugador,PaisAtacado):-
122     ocupa(Jugador,PaisPropio),
123     limitrofes(PaisAtacado,PaisPropio),
124     not( (aliados(Jugador,Aliado),ocupa(Aliado,PaisAtacado)) ).

```

---

## 4. Functores y Polimorfismo ->Video 13 Youtube

### 4.1. *libreria.pl*

```
1 % vende/2 recibe un articulo y su precio
2 % Functor libro(Titulo,Autor,Genero,Editorial)
3 % Functor cd(Titulo,Autor,Genero,CantidadDeDiscos,CantidadDeTemas)
4 vende(libro(elResplandor,stephen_king,terror,de_bolsillo),2300).
5 vende(libro(mort,terry_pratchett,aventura,plaza_janez),1300).
6 vende(libro(harry_potter_3,jk_roling,ficcion,salamandra),2500).
7 vende(cd(different_class,pulp,pop,2,24),1450).
8 vende(cd(blood_on_the_tracks,bob_dylan,1,12),2500).
9 vende(peli(it,terror,wallace),1600).
10
11 % tematico/1 se cumple para un autor si todo lo que se vende es del autor
12 tematico(Autor):-
13     autor(_,Autor),
14     forall( vende(Articulo,_), autor(Articulo,Autor) ).
15
16 % autor/2 relaciona un articulo con su autor ->Polimorfismo
17 autor(libro(_,Autor,_,_),Autor):-
18     vende(libro(_,Autor,_,_),_).
19 autor(cd(_,Autor,_,_,_),Autor):-
20     vende(cd(_,Autor,_,_,_),_).
21
22 % libroMasCaro/1 se cumple para un articulo si es el libro de mayor precio
23 libroMasCaro(libro(Titulo,Autor,Genero,Editorial)):-
24     vende(libro(Titulo,Autor,Genero,Editorial),Precio),
25     forall(vende(libro(_,_,_,_),OtroPrecio) , OtroPrecio <= Precio).
26
27 % curiosidad/1 se cumple para un articulo si es el unico que hay a la
28     venta de su autor
29 curiosidad(Articulo):-
30     vende(Articulo,_),
31     autor(Articulo,Autor),
32     not( (vende(OtroArticulo,_),autor(OtroArticulo,Autor),OtroArticulo\=
33         Articulo) ).
34
35 % sePrestaAConfusion/1 se cumple para un titulo si pertenece a mas de un
36     articulo
37 sePrestaAConfusion(Titulo):-
38     titulo(UnArticulo,Titulo),
39     titulo(OtroArticulo,Titulo),
40     UnArticulo\=OtroArticulo.
41
42 % titulo(Articulo,Titulo)
43 titulo(libro(Titulo,_,_,_),Titulo):-
44     vende(libro(Titulo,_,_,_),_).
45 titulo(cd(Titulo,_,_,_,_),Titulo):-
46     vende(cd(Titulo,_,_,_,_),_).
47 titulo(peli(Titulo,_,_),Titulo):-
48     vende(peli(Titulo,_,_),_).
49
50 % mixto/1 se cumple para los autores de mas de 1 tipo de articulo
51 mixto(Autor):-
```

```
49     autor(libro(_,_,_,_),Autor),
50     autor(cd(_,_,_,_,_),Autor).
51 mixto(Autor):-
52     autor(peli(_,_,_),Autor),
53     autor(libro(_,_,_,_),Autor).
54 mixto(Autor):-
55     autor(peli(_,_,_),Autor),
56     autor(cd(_,_,_,_,_),Autor).
```

---

## 5. Listas -> Video 14 Youtube

### 5.1. *listas.pl*

```
1 % receta(Nombre,Ingredientes)
2 %   Ingrediente: [ingrediente(NombreIngrediente,Cantidad)]
3 receta( caramelo , [ingrediente(agua,100),ingrediente(azucar,100)]).
4
5 receta_(caramelo,ingrediente(agua,100)).
6 receta_(caramelo,ingrediente(azucar,100)).
7
8 % member(ElementoDeLaLista,Lista) relaciona un elemento de la lista a la
   cual pertenece
9 % length(Lista,longitudDeLaLista) relaciona una lista con su
   longitudDeLaLista
10 % sumlist(Lista, TotalDeLaLista) relaciona una lista con el total de su
    suma de sus elementos
11
12 % rapida(Receta) si tiene menos de 4 ingredientes
13 rapida(Receta):-
14     receta(Receta,Ingredientes),
15     length(Ingredientes,Total),
16     Total < 4.
17
18 % postre(Receta) si hay un ingrediente azucar que pesa mas de 250g
19 postre(Receta):-
20     receta(Receta,Ingredientes),
21     member(ingrediente(azucar,Cantidad),Ingredientes),
22     Cantidad > 250.
23
24 % findall(SELECTOR,CONSULTA,Lista). sirve para relacionar respuestas a
   consulta seleccionada en una lista
25
26 % cantidadDePostres(CantidadDePostres) se cumple para la cantidad de
   recetas de postres en la base de conocimiento
27 cantidadDePostres(CantidadDePostres):-
28     receta(Receta,_),
29     findall(1,postre(Receta),ListaDeUnos),
30     sumlist(ListaDeUnos,CantidadDePostres).
31
32 calorias(ingrediente(agua,100),500).
33 calorias(ingrediente(azucar,100),1500).
34
35 % trivial/1 se cumple para recetas con un unico ingredientes
36 trivial(Receta):-
37     receta(Receta,[_]).%Lista con 1 solo ingrediente
38
39 % elPeor/2 relaciona una receta con su ingrediente mas calorico
40 elPeor(Ingredientes,Peor):-
41     member(Peor,Ingredientes),
42     calorias(Peor,CaloriasDelPeor),
43     forall(
44         member(Ingrediente,Ingredientes),
45         (calorias(Ingrediente,Calorias),CaloriasDelPeor >= Calorias)
46     ).
```

```
47
48
49
50 % caloriasTotales/2 relaciona una receta y su total de calorias
51 caloriasTotales(Receta,Total):-
52     receta(Receta,Ingredientes),
53     findall(Kcal,(member(Ing,Ingredientes),calorias(Ing,Kcal)),ListaDeKCal)
54     ,
55     sumlist(ListaDeKCal,Total).
56
57 % versionLight/2 relaciona una receta con sus ingredientes, sin el peor
58 versionLight(Receta,IngredientesLight):-
59     receta(Receta,Ingredientes),
60     elPeor(Ingredientes,Peor),
61     findall(Ing,(member(Ing,Ingredientes), Ing \= Peor ),IngredientesLight)
62     .
63
64 % guasada/1 re cumple para una receta con algun ingrediente de mas de 1000
65 kcal
66 guasada(Receta):-
67     receta(Receta,Ingredientes),
68     member(IngredienteEngordador,Ingredientes),
69     calorias(IngredienteEngordador,Kcal),
70     Kcal > 1000.
```

# T.E.G.

## Paradigmas de Programación - Jueves Noche - Ejercitación Paradigma Lógico

Nos piden modelar una herramienta para analizar el tablero de un juego de [Táctica y Estrategia de Guerra](#). Para eso ya contamos con los siguientes predicados completamente inversibles en nuestra base de conocimiento:

```
% Se cumple para Los jugadores.
```

```
jugador(Jugador)
```

```
% Ejemplo:
```

```
% jugador(rojo).
```

```
% Relaciona un país con el continente en el que está ubicado,
```

```
ubicadoEn(País, Continente)
```

```
% Ejemplo:
```

```
% ubicadoEn(argentina, américaDelSur).
```

```
% Relaciona dos jugadores si son aliados.
```

```
aliados(UnJugador, OtroJugador)
```

```
% Ejemplo:
```

```
% aliados(rojo, amarillo).
```

```
% Relaciona un jugador con un país en el que tiene ejércitos.
```

```
ocupa(Jugador, País)
```

```
% Ejemplo:
```

```
% ocupa(rojo, argentina).
```

```
% Relaciona dos países si son limítrofes.
```

```
limítrofes(UnPaís, OtroPaís)
```

```
% Ejemplo:
```

```
% limítrofes(argentina, brasil).
```

Se pide modelar los siguientes predicados, de forma tal que sean **completamente inversibles**:

1. **tienePresenciaEn/2**: Relaciona un jugador con un continente del cual ocupa, al menos, un país.
2. **puedenAtacarse/2**: Relaciona dos jugadores si uno ocupa al menos un país limítrofe a algún país ocupado por el otro.
3. **sinTensiones/2**: Relaciona dos jugadores que, o bien no pueden atacarse, o son aliados.
4. **perdió/1**: Se cumple para un jugador que no ocupa ningún país.
5. **controla/2**: Relaciona un jugador con un continente si ocupa todos los países del mismo.
6. **reñido/1**: Se cumple para los continentes donde todos los jugadores ocupan algún país.
7. **atrincherado/1**: Se cumple para los jugadores que ocupan países en un único continente.
8. **puedeConquistar/2**: Relaciona un jugador con un continente si no lo controla, pero todos los países del continente que le falta ocupar son limítrofes a alguno que sí ocupa y pertenecen a alguien que no es su aliado.