

## ***Jugadores de fútbol***

Se tiene una base de conocimientos con los partidos de un campeonato de fútbol:

partido(1, defensoresPDP, 3, [lombardi, spigariol, lombardi], algoritmosUnidos, 1, [bruno], baldassi).  
 partido(1, sistemasOperativos, 0, [], atleticoTADP, 3, [gassman, salanova, peck], lamolina).  
 ...

Donde el predicado partido relaciona:

- Fecha del campeonato
- Equipo local
- Goles del equipo local
- Goleadores del equipo local (en el orden en que convirtieron, pueden repetirse)
- Equipo visitante
- Goles del equipo visitante
- Goleadores del equipo visitante (misma consideración que los goleadores del local)
- Árbitro del encuentro.

a) Se pide realizar las consultas/cláusulas para determinar:

- 1) si ganó un equipo en una fecha.
- 2) los goles que convirtió un jugador para un equipo en un campeonato.
- 3) cuántos puntos tiene un equipo en el campeonato (3 por partido ganado y 1 por partido empatado).
- 4) si un equipo está invicto de local.
- 5) si un árbitro es sospechoso (cuando cada vez que dirigió a un equipo ese equipo ganó siempre o perdió siempre).

b) Tomar un predicado de los definidos en el a) y hacer dos versiones, una que resulte inversible y la otra que no, indicando por qué una no es inversible y la otra sí.

c) Se agrega ahora información referente a un equipo:

equipo(defensoresPDP, [arquero(cesario), defensor(dodino), volante(lombardi), delantero(passerini), ...]).

Con el formato equipo(nombre, lista de jugadores con su posición).

Definir el predicado

jugadoresBuenos(Equipo, ListaJugadoresBuenos)

que relaciona los jugadores buenos de un equipo. Un jugador es bueno si:

- es arquero (se supone que todos los arqueros son buenos)
- siendo volante convirtió más de 3 goles en el campeonato
- siendo delantero hizo 2 ó más goles en algún partido

## ***Una solución posible***

```
partido(1, defensoresPDP, 3, [lombardi, spigariol, lombardi],
      algoritmosUnidos, 1, [bruno], baldassi).
partido(1, sistemasOperativos, 0, [], atleticoTADP, 3, [gassman, salanova, peck], lamolina).
partido(2, sistemasOperativos, 0, [], sintaxis, 0, [], lamolina).
...
```

```
/** 1 **/
```

```
gano(Equipo, Fecha):-partido(Fecha, Equipo, GolesAFavor, _, _, GolesEnContra, _, _),
  GolesAFavor > GolesEnContra.
gano(Equipo, Fecha):-partido(Fecha, _, GolesEnContra, _, Equipo, GolesAFavor, _, _),
  GolesAFavor > GolesEnContra.
```

```
/** 2 **/
```

```
goles(Jugador, Equipo, CantidadGoles):-
  findall(Goleadores, goleadoresXFecha(Equipo,_, Goleadores), GoleadoresXFecha),
  cantidadGoles(Jugador, GoleadoresXFecha, CantidadGoles).
```

```
goleadoresXFecha(Equipo, Fecha, Goleadores):-
  partido(Fecha, Equipo, _, Goleadores, _,_,_).
goleadoresXFecha(Equipo, Fecha, Goleadores):-
  partido(Fecha, _,_,_, Equipo,_,Goleadores, _).
```

```
cantidadGoles(_, [], 0).
cantidadGoles(Jugador, [Goleadores|Gs], CantidadGoles):-
  golesXFecha(Jugador, Goleadores, GolesFecha),
  cantidadGoles(Jugador, Gs, CantidadGolesOtrasFechas),
  CantidadGoles is GolesFecha + CantidadGolesOtrasFechas.
```

```
golesXFecha(Jugador, Goleadores, GolesFecha):-
  findall(Jugador, member(Jugador, Goleadores), ListaVecesQueHizoGoles),
  length(ListaVecesQueHizoGoles, GolesFecha).
```

```
/** 3 **/
```

```
puntos(Equipo, Puntos):-
  findall(Equipo, gano(Equipo, Fecha), ListaVecesQueGano),
  findall(Equipo, empate(Equipo, Fecha), ListaVecesQueEmpato),
  length(ListaVecesQueGano, VecesQueGano),
  length(ListaVecesQueEmpato, VecesQueEmpato),
  Puntos is VecesQueGano * 3 + VecesQueEmpato.
```

```
empato(Equipo, Fecha):-
  partido(Fecha, Equipo, GolesAFavor, _, _, GolesEnContra, _, _),
  GolesAFavor is GolesEnContra.
empato(Equipo, Fecha):-
  partido(Fecha, _, GolesEnContra, _, Equipo, GolesAFavor, _, _),
  GolesAFavor is GolesEnContra.
```

```
/** 4 **/
```

```
invictoDeLocal(Equipo):-equipo(Equipo),
  forall(jugo(Equipo, Fecha),
    not(perdio(Equipo, Fecha))).
```

```
jugo(Equipo, Fecha):-partido(Fecha, Equipo, _,_,_,_,_).
jugo(Equipo, Fecha):-partido(Fecha,_,_,_,Equipo,_,_,_).
```

```
perdio(Equipo, Fecha):-not(gano(Equipo, Fecha)), not(empato(Equipo, Fecha)).
```

**/\*\* 5 \*\*/**

sospechoso(Arbitro, Equipo):-forall(dirigio(Arbitro, Equipo, Fecha),  
gano(Equipo, Fecha)).

sospechoso(Arbitro, Equipo):-forall(dirigio(Arbitro, Equipo, Fecha),  
perdio(Equipo, Fecha)).

dirigio(Arbitro, Equipo, Fecha):-partido(Fecha,Equipo,\_,\_,\_,\_,Arbitro).

dirigio(Arbitro, Equipo, Fecha):-partido(Fecha,\_,\_,\_,Equipo,\_,\_,Arbitro).

**/\*\* b \*\*/**

/\* Tienen que usar predicados generadores para el punto

5 con el Arbitro y el Equipo

4 con el Equipo

3 con el Equipo

2 con el Jugador y el Equipo \*/

**/\*\* c 1 \*\*/**

equipo(defensoresPDP, [arquero(cesario), defensor(dodino), volante(lombardi),  
delantero(passerini)]).

jugadoresBuenos(Equipo, JugadoresBuenos):-

findall(Jugador, (equipo(Equipo, Jugadores), member(Jugador, Jugadores),  
jugadorBueno(Equipo, Jugador)), JugadoresBuenos).

jugadorBueno(\_, arquero(\_)).

jugadorBueno(Equipo, volante(Jug)):-goles(Jug, Equipo, CantGoles),  
CantGoles >= 3.

jugadorBueno(Equipo, delantero(Jug)):-

goleadoresXFecha(Equipo, \_, Goleadores),  
golesXFecha(Jug, Goleadores, CantGolesXFecha), CantGolesXFecha >= 2.