

El juego de las sillas

Como especialistas en el armado de mesas para eventos (fiestas de Casamiento, Cumpleaños de 15, Comuniones, etc.) se quiere automatizar las relaciones que siempre se deben de considerar en la distribución de las personas en mesas.

Para esto se dispone de la siguiente información en forma de base de conocimiento Prolog:

- de cada persona se conocen sus amigos.

```
amigo(juan, alberto).  
amigo(juan, pedro).  
amigo(pedro, mirta).  
amigo(alberto, tomas).  
amigo(tomas, mirta).
```

- ... y sus enemigos.

```
enemigo(mirta, ana).  
enemigo(juan, nestor).  
enemigo(juan, mirta).
```

- Para cada fiesta tenemos las mesas conformadas

- Para modelar una **mesa** vamos a usar un functor que tiene número de mesa y una lista de comensales.

```
mesaArmada(navidad2010, mesa(1, [juan, mirta, ana, nestor])).  
mesaArmada(navidad2010, mesa(5, [andres, german, ludmila, elias])).  
mesaArmada(navidad2010, mesa(8, [nestor, pedro])).
```

Se pide:

1. `estaSentadaEn/2` que relaciona a cada persona que esta sentada en una mesa. Este predicado debe de ser **inversible**.

```
?- estaSentadaEn(Quien, Mesa).  
Quien = juan,  
Mesa = mesa(1, [juan, mirta, ana, nestor]) ;  
Quien = mirta,  
Mesa = mesa(1, [juan, mirta, ana, nestor]) ;
```

2. `sePuedeSentar/2` que relaciona a una persona y una mesa si la persona puede sentarse en esa mesa. Una persona se sienta en una mesa si hay por lo menos un amigo y no hay enemigos.

```
?- sePuedeSentar(pedro, mesa(1, [juan, mirta, ana, nestor])).  
true .
```

3. `mesaDeCumpleaño/2` donde se crea una mesa ideal para una persona (el cumpleaños). La mesa ideal para alguien el día de su cumpleaños es la mesa número 1 donde está él y todos sus amigos.

```
?- mesaDeCumpleaño(juan, M).  
M = mesa(1, [juan, alberto, pedro]).
```

Nota: Recuerde que existe el predicado `append(Lista1, Lista2, Lista12)`.

4. `incompatible/2` relacione a dos personas por ser incompatibles, esto pasa cuando tienen una persona en común pero para uno es amigo y para el otro es enemigo.

```
?- incompatible(X, Y).  
X = pedro,  
Y = juan ;
```

5. `laPeorOpcion/2` relaciona a una persona y una mesa posible si todos los que están

sentados en ella son sus enemigos.

6. `mesasPlanificadas/2`: relaciona a una fiesta con todas las mesas que fueron planificadas para ella. Este predicado debe de ser **inversible**.

```
?- mesasPlanificadas(F,Mesas).
```

```
F = navidad2010,
```

```
Mesas = [mesa(1, [juan, mirta, ana, nestor]), mesa(5, [andres, german,  
ludmila, elias]), mesa(8, [nestor, pedro])]
```

7. `esViable/1`: se verifica para una lista de mesas si:

- Las mesas son consecutivas (es decir viene la mesa 1, después la 2, etc)
- Hay 4 personas en cada mesa (si exactamente 4 ni mas ni menos).
- En ninguna mesa hay dos personas que sean enemigas entre sí.

Una lista vacía de mesas no se considera viable.

```
?- esViable([mesa(1,[juan,pedro,ana,diego]),mesa(2,[ludmi,javi,elias,hernan])]).  
true .
```

```
?- mesasPlanificadas(navidad2010,Mesas),esViable(Mesas).  
false.
```