

## Paradigma Lógico

### Trabajo Práctico

## Los Topos



Se cuenta con la siguiente base de conocimientos sobre distintos individuos que entierran una gran variedad de cosas (y por eso el mote de “topos”):

```
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, [marDelPlata, 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(lassie, collie, 8), [victimas]).
```

```
entierra(perro(pongo, dalmata, 4), [oro, joyas]).
```



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 alguien es peligroso (si entierra víctimas).
6. Saber si algo es un tesoro (si lo enterró un pirata).
7. Saber quién es rico (alguien es rico si entierra oro).
8. Los piratas que no navegan por ningún mar son nauseosos. Saber quiénes son los nauseosos.
9. Implementar **estaViejito/1** que se cumple para los perros de siete años o más.
10. Implementar **perrodeal/2** que relaciona el nombre de un perro y el nombre de una persona (sea o no pirata, se pide sólo el nombre), cuando todo lo enterrado por el perro, también lo entierra la persona, siempre y cuando el perro no sea viejito.
11. Implementar **esPirataArgento/1** que da verdadero para los piratas cuyo primer mar navegado es Mar Del Plata.
12. 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).
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. Aviso... **¡no debe restringirse el orden de los tesoros!** Resolverlo sin usar el símbolo “\=”.
14. Saber si, dado un elemento, todos los que enterraron ese elemento son patéticos. Un pirata es patético si es nauseoso y no peligroso; un perro es patético si no enterró un hueso; “Old Mac Donald” siempre es patético. El resto nunca es patético.

## **Extras** (no hace falta hacerlos, pero les sirve de práctica)

15. Dado un conjunto de piratas, conocer el conjunto de mares en común por los que navegaron. *El predicado no necesita ser inversible para el conjunto de piratas.*
16. Saber si un pirata es pichon de otro. Esto sucede cuando un pirata navega un subconjunto de mares del segundo.
17. Implementar el predicado **sonCompetencia/2** que relaciona a dos topos si entierran más de una cosa en común.

**Nota:** todos los predicados “principales” deben ser completamente inversibles a menos que se indique lo contrario. No hace falta que los predicados auxiliares lo sean. Además, cuando se deba unificar contra un individuo, ya sea persona o perro, debe hacerse contra todo el functor que lo representa, no sólo contra el nombre.

Puede serles de utilidad el predicado **select/3**, el cual relaciona un elemento, una lista que contiene a ese elemento, y otra lista que es igual a la primera sin el elemento. Ejemplo:

```
?- select(2, [1,2,3,4], [1,3,4]).  
true .
```