

## Ejercicios Prolog 1

1. Considerando el siguiente programa, dibuja el árbol de ejecución de Prolog de la consulta

?- top(X,Y)

mostrando las respectivas soluciones para cada rama del árbol.

```
q(a).
q(b).
r(c).
r(d).
s(e).
top(X,Y) :- p(X,Y).
top(X,X) :- s(X).
p(X,Y) :- q(X), r(Y).
p(X,Y) :- s(X), r(Y).
```

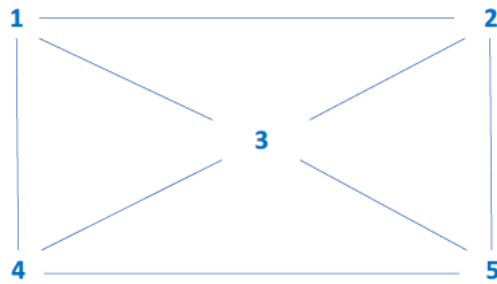
2. Define una base de conocimiento con series gallegas o rodadas en Galicia. Diseña la sintaxis de los hechos, indicando su significado y teniendo en cuenta las consultas que se desea realizar a la Base de Conocimiento. Consigue una colección de hechos lo suficientemente grande (de forma colaborativa con tus compañeros de clase). Añade los hechos necesarios recopilados en el paso previo, y al menos las siguientes reglas a la base de conocimiento:

```
estrenada_despues(P, A) <- la serie P se estrenó después del año A.
estrenada_antes(P, A) <- la serie P fue estrenada antes del año A.
mismo_año(P1, P2) <- las dos películas P1 y P2 se estrenan en el mismo año.
participan(A1, A2) <- los actores/actrices A1 y A2 participan en la misma serie.
```

Escribe las preguntas para responder a las siguientes preguntas.

- a. ¿En qué año se estrenó la serie Mareas Vivas?
  - b. ¿Qué series gallegas se estrenaron en el año 2010?
  - c. ¿Qué series se estrenaron antes del 2010?
  - d. ¿Qué series se estrenaron después de 2010?
  - e. Encuentre, al menos, dos series que se estrenaran en el mismo año.
  - f. Encuentre un actor que haya aparecido en más de una serie.
  - g. Encuentre la serie en la que Luis Tosar y Willy Toledo participan.
3. Un grafo conectado como el de la figura puede representarse mediante predicados que representan sus conexiones (aristas). Por ejemplo, el enlace entre el nodo 1 y el nodo 2 se puede representar por el hecho:

arista(1,2).



- Define todos los hechos necesarios para representar el grafo de la figura.
- Para representar enlaces bidireccionales podríamos añadir ocho hechos más de tipo arista (por ejemplo, arista(2,1)) o podríamos intentar una regla como:

`arista(X,Y) :- arista(Y,X).`

Prueba esta regla para consultar la arista entre 5 y 1:

?- arista(5,1).

¿Qué problema tiene esta regla? ¿Por qué?

- Prueba las siguientes dos reglas:

`conexion(X,Y) :- arista(X,Y).`

`conexion(X,Y) :- arista(Y,X).`

- Sustituye las dos reglas anteriores por la única regla:

`conexion(X,Y) :- arista(X,Y) ; arista(Y,X).`

Indica la diferencia entre esta regla y las reglas del apartado c.

- Realiza un programa en Prolog que devuelva el camino (o los caminos, si hay más de uno) entre dos nodos del grafo.

```

Via = [1, 4]
Via = [1, 2, 3, 4]
Via = [1, 2, 3, 5, 4]
Via = [1, 2, 5, 4]
Via = [1, 2, 5, 3, 4]
Via = [1, 3, 4]
Via = [1, 3, 5, 4]
Via = [1, 3, 2, 5, 4]
false
?- camino(1,4,Via).

```

Se alcanza un camino de A a B si A y B están conectados. Defina una (o más) reglas recursivas intermedias llamadas **encuentra\_camino** que obtengan un camino de A a B siempre que A esté conectado a un nodo C (diferente de B), que no esté en la parte ya visitada del camino, y siga encontrando un camino de C a B. Al evitar los nodos repetidos aseguramos que el programa no se meterá en un bucle infinito. La función **camino** invocará a esta función intermedia y dará la vuelta al camino devuelto, para visualizar el camino de A a B (y no de B a A que nos devolvería *encuentra\_camino*).