

**Práctica 5:****LABERINTO****OBJETIVO**

- Representación de grafos en Prolog.
- Recorrido de un grafo.

**PLANTEAMIENTO DEL PROBLEMA**

a) En esta practica se debe implementar un predicado que nos permita encontrar un camino en un grafo dirigido.

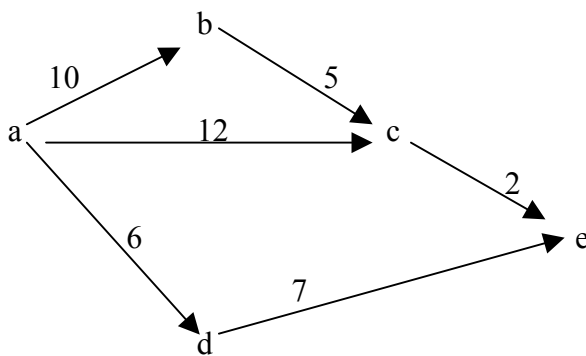
Un grafo se representa en Prolog utilizando hechos, donde cada hecho representa un arco del grafo. Cada hecho tendrá tres parámetros: nodo origen del arco, nodo destino del arco y longitud del arco.

Ej.

```
arco(a, b, 10).
arco(b, c, 5).
arco(a, c, 12).
arco(a, d, 6).
arco(d, e, 7).
arco(c, e, 2).
```

El predicado a implementar es camino/4. A este predicado se le pasan como parámetros de entrada el nodo de inicio y el nodo final, y tendrá como resultados el camino y su longitud. El camino será una lista con todos los nodos del camino, incluyendo el nodo inicial y el final.

El predicado camino se deberá probar sobre el siguiente grafo:



b) En un laberinto real, cada tramo del laberinto se puede recorrer en las dos direcciones: hacia delante y hacia atrás. El grafo correcto para representar este laberinto será, por tanto, un grafo no dirigido. Para conseguir que nuestro laberinto sea un grafo no dirigido se puede definir un nuevo tipo de arco:

```
arcond(X, Y, Dist):-
    arco(X, Y, Dist);
    arco(Y, X, Dist).
```

Escribir un nuevo predicado caminond/5 similar al predicado camino/4 para que sea capaz de recorrer un grafo no dirigido.

Para evitar que el algoritmo se quede en ciclos infinitos, hay que llevar la cuenta de los nodos que hemos recorrido para no pasar dos veces por el mismo sitio. Los nodos recorridos se guardarán en una lista que corresponderá con el quinto parámetro. El algoritmo, cada vez que pase por un nodo nuevo, deberá comprobar que no está en la lista. Para ello se utilizará la función *miembro* de la práctica 4.