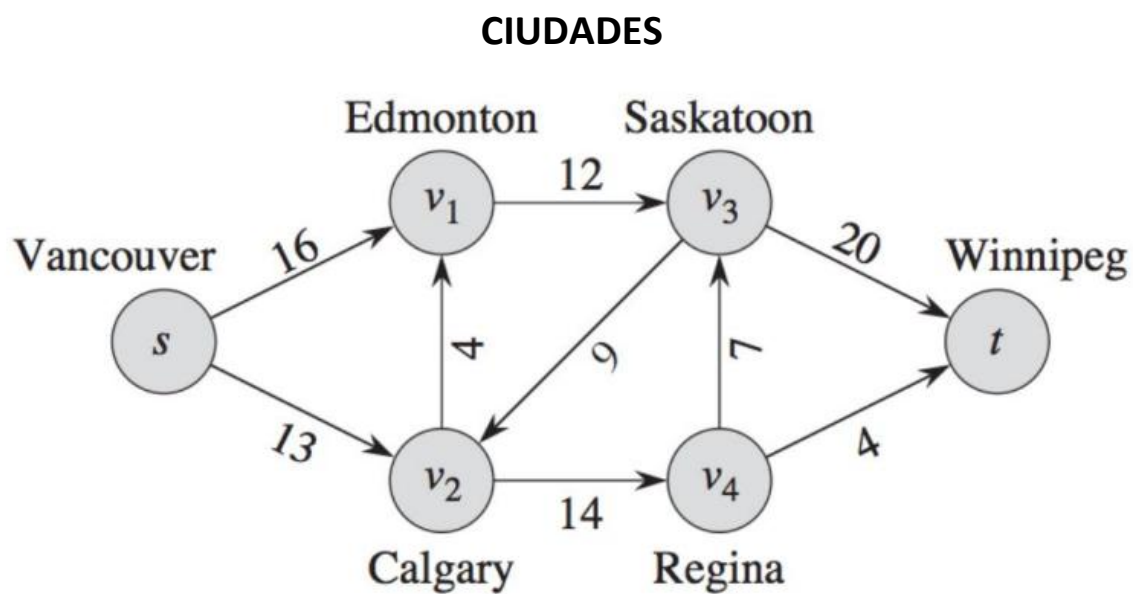


Actividad 2: Backtracking y grafo dirigido

Juan Camilo Guevara Osorio – 1105365091 – Grupo 1












CÓDIGO FUENTE






```
1  %----- DEFINICION DE RELACIONES -----%
2  conexion(vancouver, edmonton, 16).
3  conexion(vancouver, calgary, 13).
4  conexion(edmonton, saskatoon, 12).
5  conexion(calgary, edmonton, 4).
6  conexion(calgary, regina, 14).
7  conexion(saskatoon, calgary, 9).
8  conexion(saskatoon, winnipeg, 20).
9  conexion(regina, saskatoon, 7).
10 conexion(regina, winnipeg, 4).
11
12 %----- DEFINICION DE REGLAS -----%
13
14 tiene_aristas(X) :-
15     conexion(X, _, _), !.
16
17 costo_nodoXYZ(X, Z, C) :-
18     conexion(X, Y, C1), conexion(Y, Z, C2),
19     C is C1 + C2.
20
21 %muestra todos los nodos con los que tiene conexion (aristas)
22 aristas(X, Y) :-
23     findall((Z, C), conexion(X, Z, C), Y).
24
25 %caso base
26 camino(I, I, 0, [(I, 0)], _) :- !.
27
28 %inicio, final, coste, camino, ciudades visitadas
29 %consulta: (ciudad de inicio, ciudad final, C, Y, [ciudad de inicio])
30 camino(I, F, C, [(I, C1)|Camino], Visitados) :-
31     conexion(I, M, C1),
32     \+ member(M, Visitados),
33     camino(M, F, C2, Camino, [M|Visitados]),
34     C is C1 + C2,
35     !.
36
37
```

COMPROBACION DE LAS FUNCIONES

Tiene aristas:

 <code>tiene_aristas(saskatoon)</code>	  
<code>true</code>	<code>1</code>
 <code>tiene_aristas(regina)</code>	  
<code>true</code>	<code>1</code>
 <code>tiene_aristas(winnipeg)</code>	  
<code>false</code>	



Costo nodo XYZ:

 <code>costo_nodoXYZ(vancouver, edmonton, C)</code>	  
<code>C = 17</code>	
 <code>costo_nodoXYZ(vancouver, saskatoon, C)</code>	  
<code>C = 28</code>	
 <code>costo_nodoXYZ(saskatoon, regina, C)</code>	  
<code>C = 23</code>	
<code>false</code>	

Aristas:

 <code>aristas(vancouver, Y)</code>	  
<code>Y = [(edmonton,16), (calgary,13)]</code>	
 <code>aristas(calgary, Y)</code>	  
<code>Y = [(edmonton,4), (regina,14)]</code>	
 <code>aristas(winnipeg, Y)</code>	  
<code>Y = []</code>	
 <code>aristas(edmonton, Y)</code>	  
<code>Y = [(saskatoon,12)]</code>	

Camino recursivo:

 <code>camino(edmonton, regina, C, Y, [edmonton])</code>	  
<code>C = 35,</code> <code>Y = [(edmonton,12), (saskatoon,9), (calgary,14), (regina,0)]</code>	
 <code>camino(vancouver, regina, C, Y, [vancouver])</code>	  
<code>C = 51,</code> <code>Y = [(vancouver,16), (edmonton,12), (saskatoon,9), (calgary,14), (regina,0)]</code>	
 <code>camino(regina, edmonton, C, Y, [vancouver])</code>	  
<code>C = 20,</code> <code>Y = [(regina,7), (saskatoon,9), (calgary,4), (edmonton,0)]</code>	
 <code>camino(vancouver, winnipeg, C, Y, [vancouver])</code>	  
<code>C = 55,</code> <code>Y = [(vancouver,16), (edmonton,12), (saskatoon,9), (calgary,14), (regina,4), (winnipeg,0)]</code>	