

GitHub: <https://github.com/DanielCuellar7/Programacion-utp/blob/main/prolog/ProgIIIIG101-Act02-DanielCuellar-YohanChala.pdf>

Backtracking y SDL

1. ¿Existe una conexión entre Saskatoon y Vancouver?
2. ¿Con qué nodos está conectado Regina y cuál es el costo de cada conexión?
3. Construir una regla para determinar si un nodo
4. tiene aristas
5. Construir una regla para determinar cuál es el
6. costo para ir de un nodo X a un Z pasando por Y

SOLUCIÓN:

HECHOS

%Conexiones de las ciudades
conexion(vancouver, edmonton, 16).
conexion(vancouver, calgary, 13).
conexion(edmonton, saskatoon, 12).
conexion(saskatoon, winnipeg, 20).
conexion(calgary, edmonton, 4).
conexion(calgary, regina, 14).
conexion(regina, saskatoon, 7).
conexion(regina, winnipeg, 4).

REGLA

nodoConArista(X) :- conexion(X, _, _).
costo(X,Y,C):-conexion(X,Y,C).
costo(X,Y,C):-conexion(X,Z,C1), conexion(Z,Y,C2), C is C1+C2.

PREGUNTAS

1. ?- conexion(saskatoon, vancouver, _).
2. ?- conexion(regina, X, Costo).
3. ?-nodoConArista(winnipeg). / FALSE
4. ?nodoConArista(regina). / TRUE
5. ?-costo(vancouver, saskatoon, Costo). / 28

The screenshot shows the SWISH Prolog IDE interface. The left pane displays the Prolog code, and the right pane shows the execution results.

Program Code:

```
1 %Conexiones de las ciudades
2 conexion(vancouver, edmonton, 16).
3 conexion(vancouver, calgary, 13).
4 conexion(edmonton, saskatoon, 12).
5 conexion(saskatoon, winnipeg, 20).
6 conexion(calgary, edmonton, 4).
7 conexion(calgary, regina, 14).
8 conexion(regina, saskatoon, 7).
9 conexion(regina, winnipeg, 4).
10
11 %Reglas para determinar si hay una arista
12 arista(X):-conexion(X,_,_).
13
14 %Reglas para dar el costo total de un X
15 %a un nodo Y pasando por Z.
16 costo(X,Y,C):-conexion(X,Y,C).
17 costo(X,Y,C):-conexion(X,Z,C1), conexion(Z,Y,C2), C is C1+C2.
18
```

Execution Results:

- `conexion(saskatoon,vancouver,c).` **false**
- `conexion(regina,X,Costo).`
Costo = 7,
X = saskatoon
Costo = 4,
X = winnipeg
- `arista(SASKATOON)`
SASKATOON = vancouver
Next 10 | 100 | 1,000 | Stop
- `costo(vancouver,saskatoon,Costo).`
Costo = 28
Next 10 | 100 | 1,000 | Stop
- `conexion(edmonton,calgary,c)` **false**
- `?- conexion(edmonton,calgary,c)`

