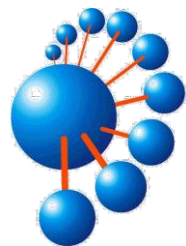


# Inteligencia Artificial

## Resolución de problemas mediante búsqueda

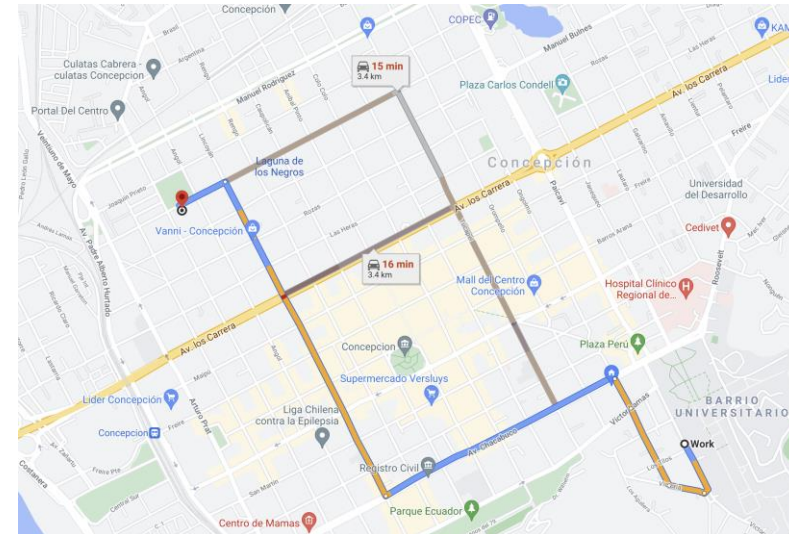
Profesor: Julio Godoy  
Ayudante: Felipe Cerda  
DIICC





# EJEMPLOS DE PROBLEMAS

- Problemas de juguete
  - Puzzle-8
  - 8-reinas
  - Misioneros y caníbales
- Problemas del mundo real
  - Determinación de una ruta
  - Problema del vendedor viajero
  - Navegación de un robot
  - Secuencia de ensamblaje





# PUZZLE 8

4	8	1
5	6	2
3	7	



1	2	3
4	5	6
7	8	

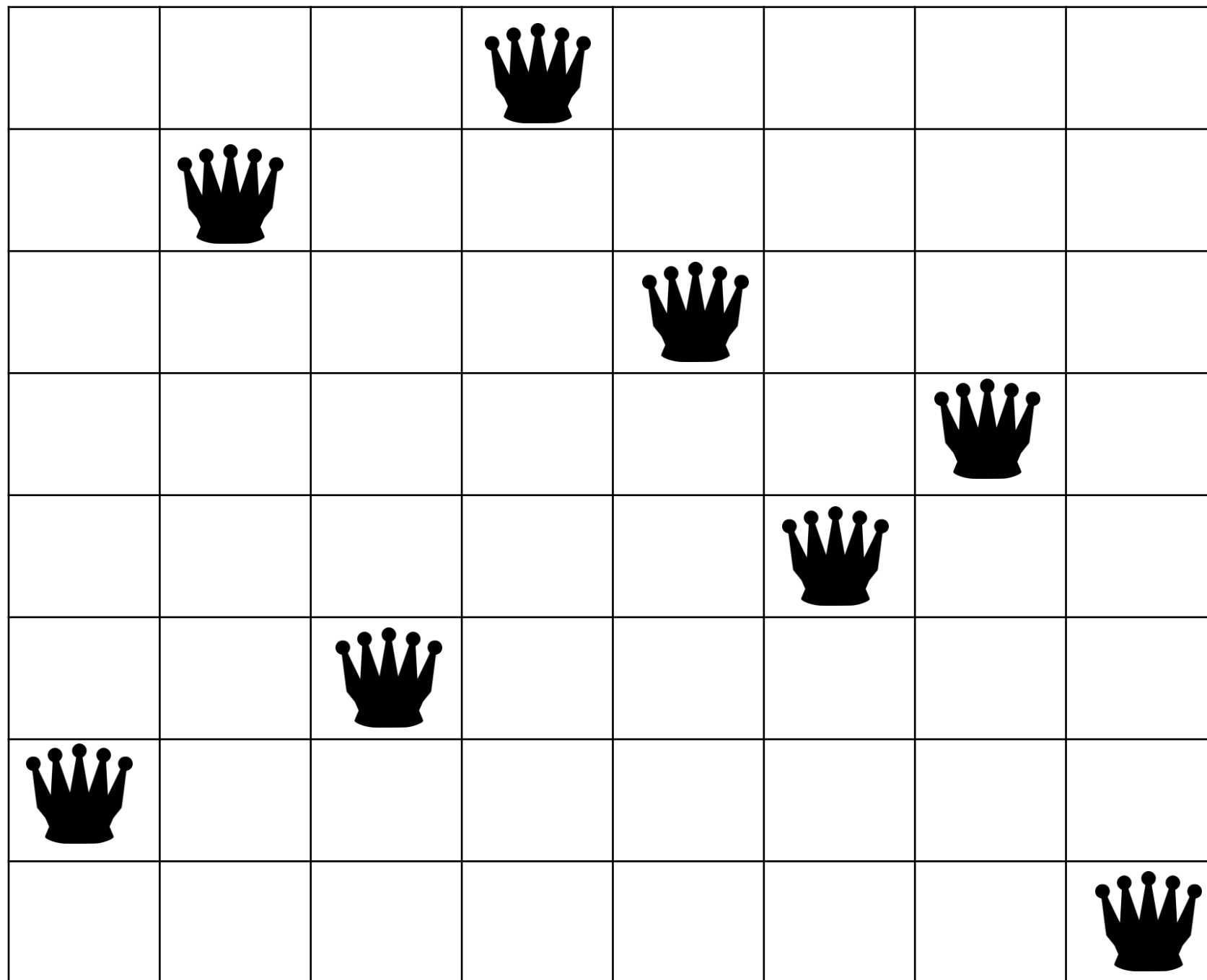


# FORMULACIÓN DE PROBLEMA – PUZZLE 8

- Estados
  - ubicación de piezas (incluyendo el “blanco”)
- Operadores
  - mover pieza
  - alternatively: mover el “blanco”
- Test de objetivo
  - configuración predefinida de piezas
- Costo del camino
  - una unidad por movimiento

4	8	1
5	6	2
3	7	

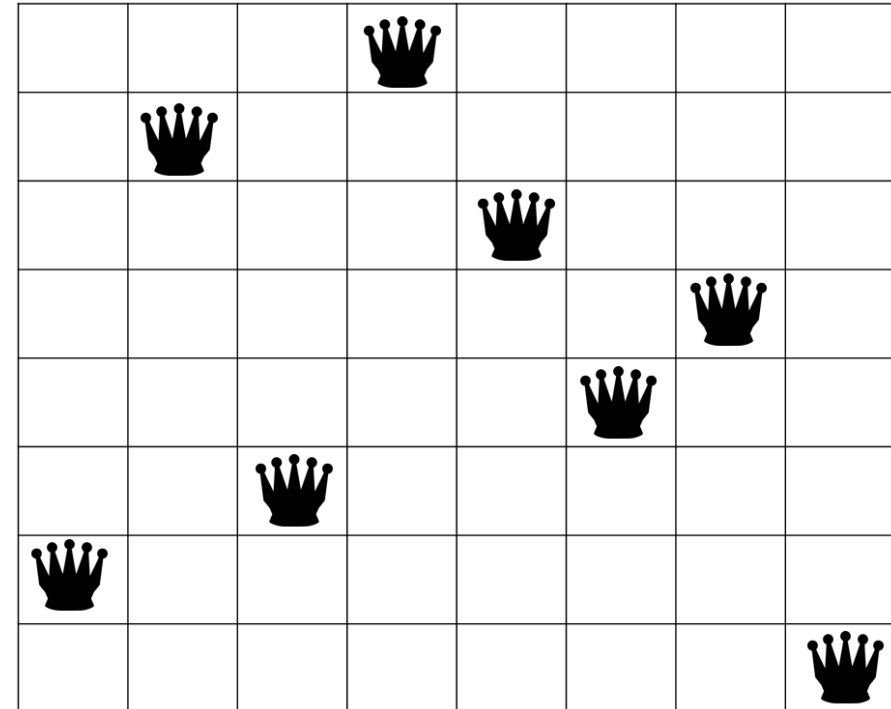
# 8-REINAS





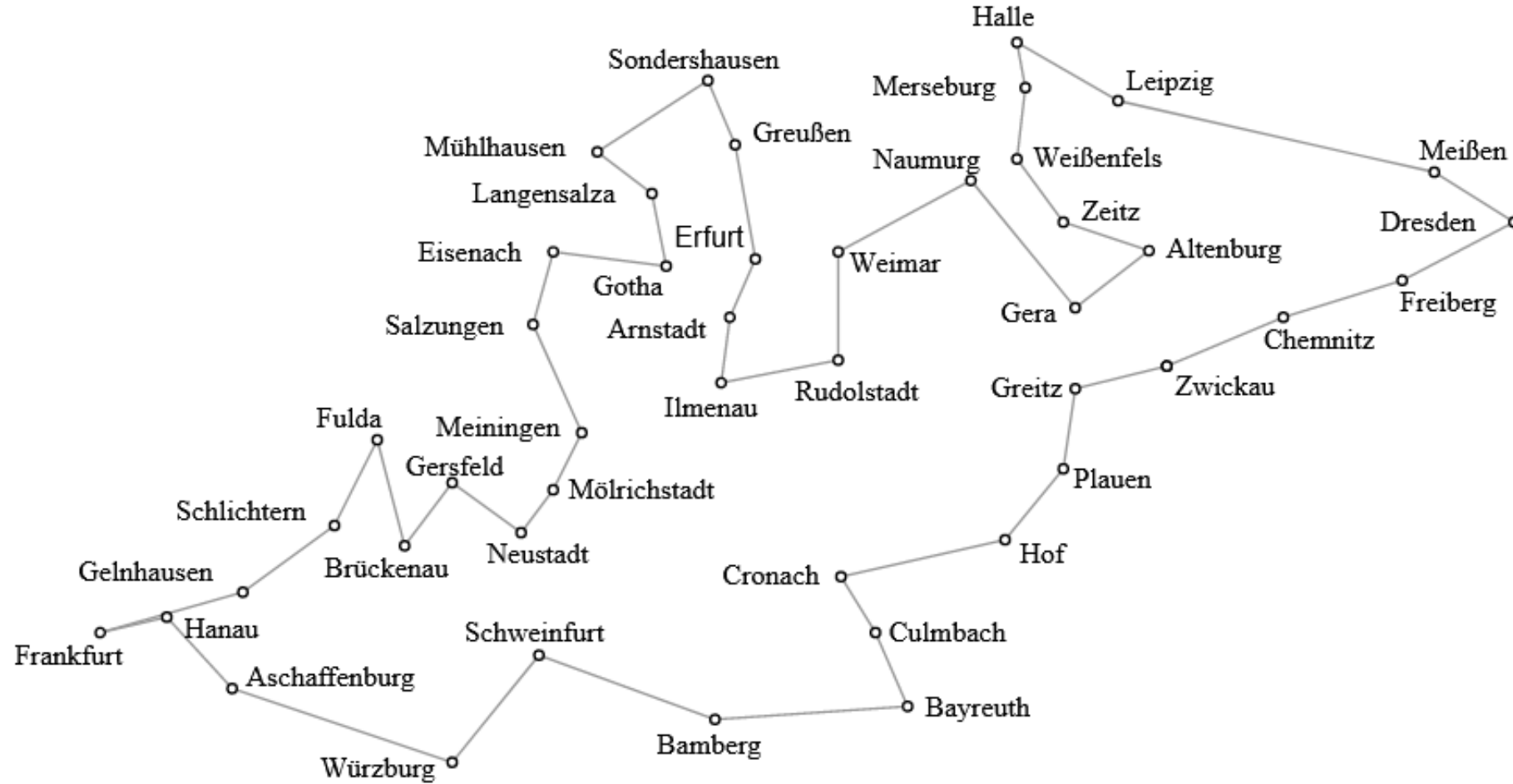
# FORMULACIÓN DE PROBLEMA - 8 REINAS

- Estados
  - Ubicación de reinas
- Operadores
  - Mover reina a otra posición
- Test de objetivo
  - Cantidad de reinas que se atacan
- Costo del camino
  - Una unidad por movimiento





# PROBLEMA DEL VENDEDOR VIAJERO

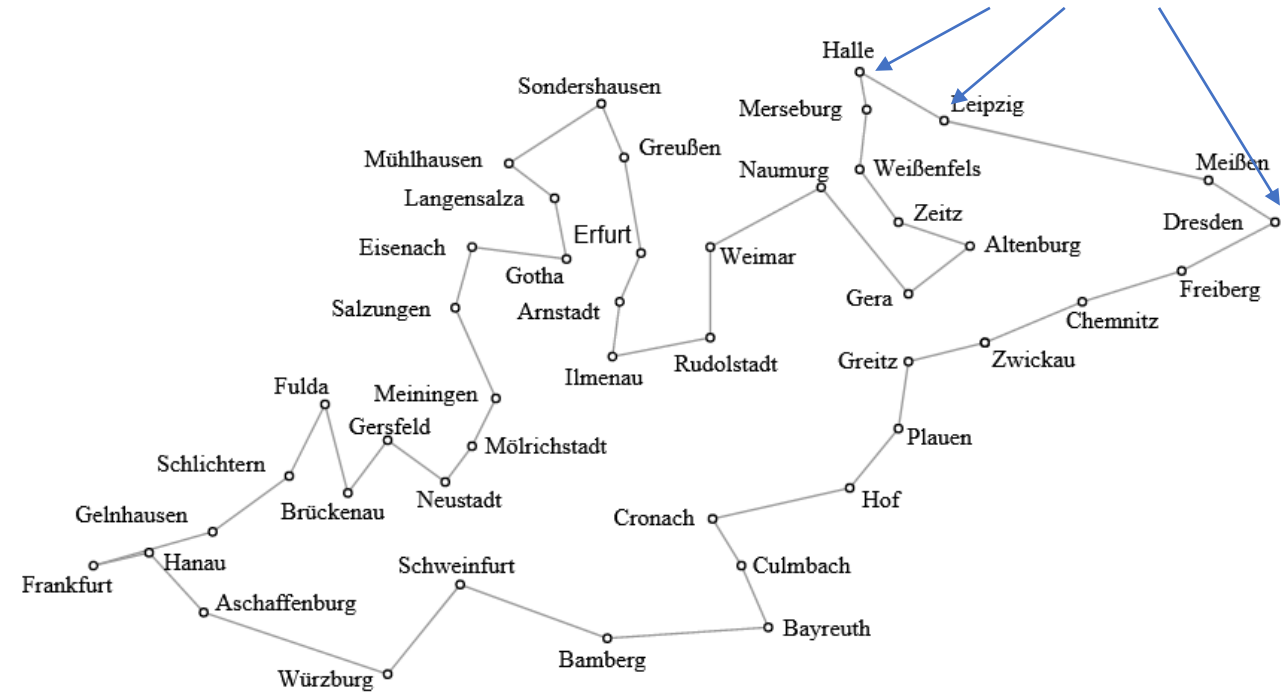




# PROBLEMA DEL VENDEDOR VIAJERO



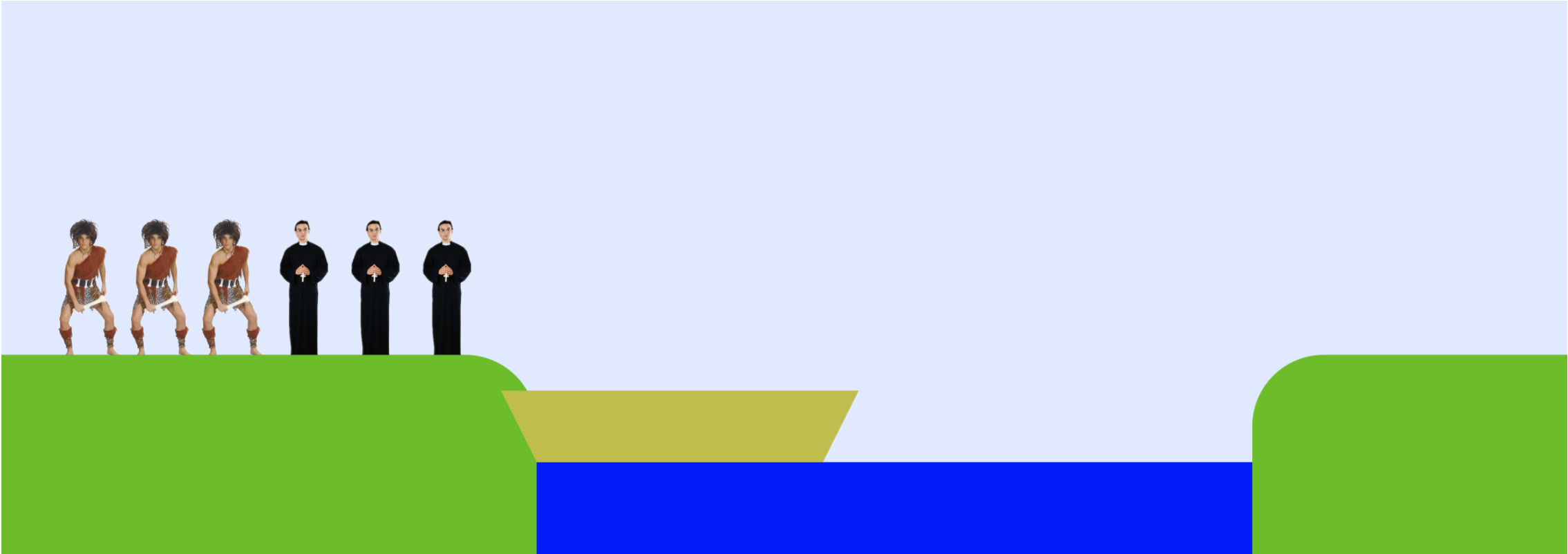
- Estados
  - Ubicaciones / ciudades
  - Estados ilegales
    - cada ciudad debe ser visitada sólo una vez
- Operadores
  - Mover desde una ubicación a otra
- Test de objetivo
  - Todas las ubicaciones visitadas
  - Agente en la ubicación inicial
- Costo del camino
  - Distancia entre ubicaciones







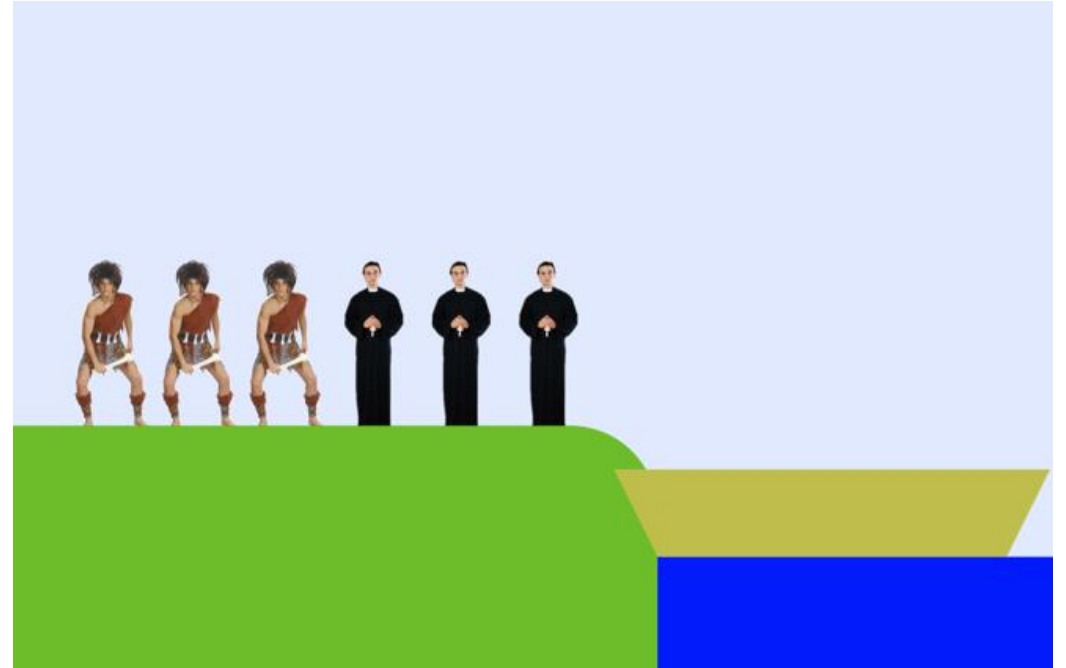
# MISIONEROS Y CANÍBALES





# MISIONEROS Y CANÍBALES

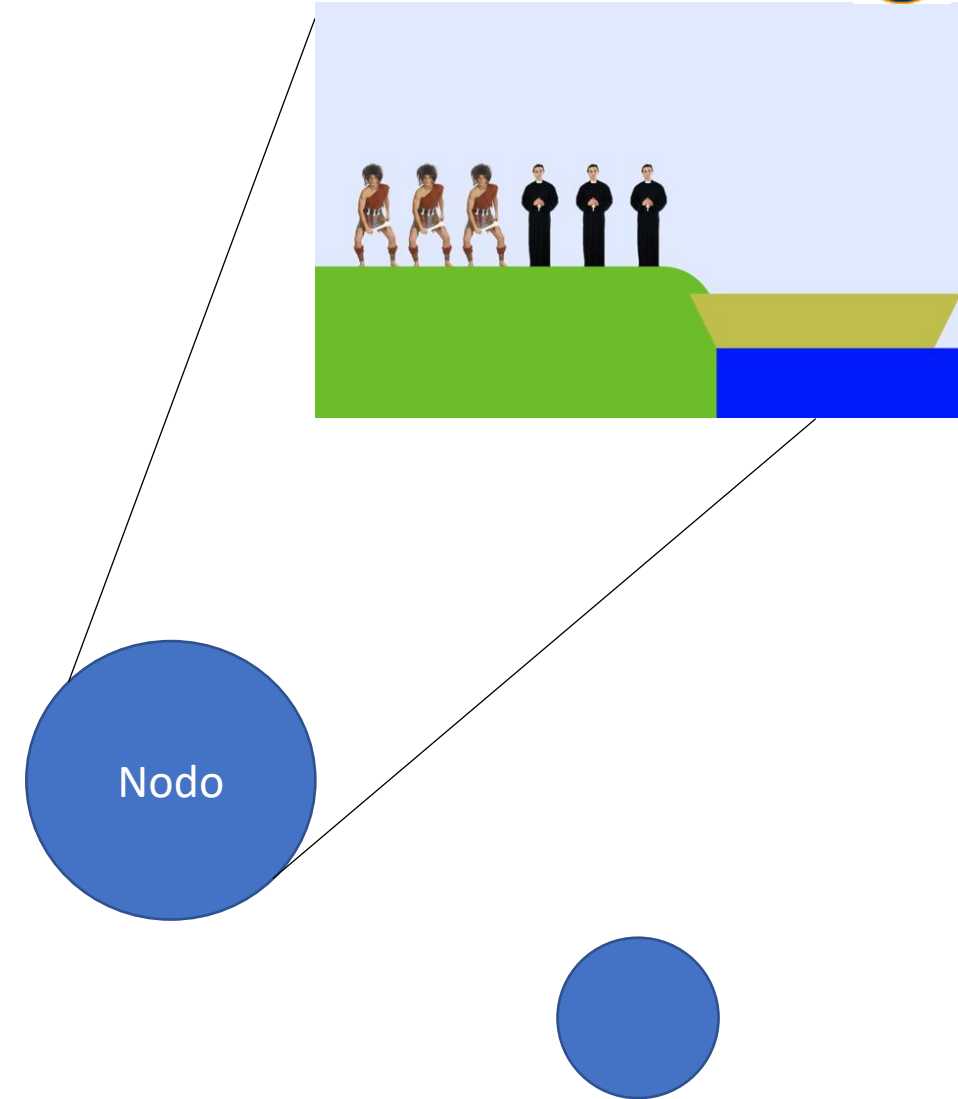
- Estados
  - # caníbales en una orilla
  - # misioneros en una orilla
  - Ubicación del bote (E/O)
- Operadores
  - Mover a uno o dos misioneros y/o caníbales de una orilla a la otra
- Test de objetivo
  - Todos los misioneros y caníbales en la orilla opuesta
- Costo del camino
  - Cantidad de viajes del bote





# BÚSQUEDA DE SOLUCIONES

- Consiste en recorrer el espacio de búsqueda
  - Desde el estado inicial al estado final
  - Secuencia permitida de acciones según operadores





# BÚSQUEDA DE SOLUCIONES

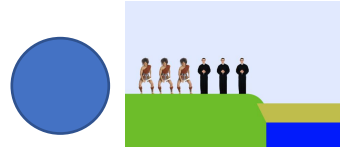
- Consiste en recorrer el espacio de búsqueda
  - Desde el estado inicial al estado final
  - Secuencia permitida de acciones según operadores





# BÚSQUEDA DE SOLUCIONES

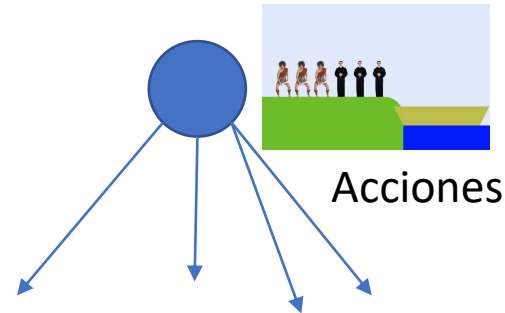
- Consiste en recorrer el espacio de búsqueda
  - Desde el estado inicial al estado final
  - Secuencia permitida de acciones según operadores





# BÚSQUEDA DE SOLUCIONES

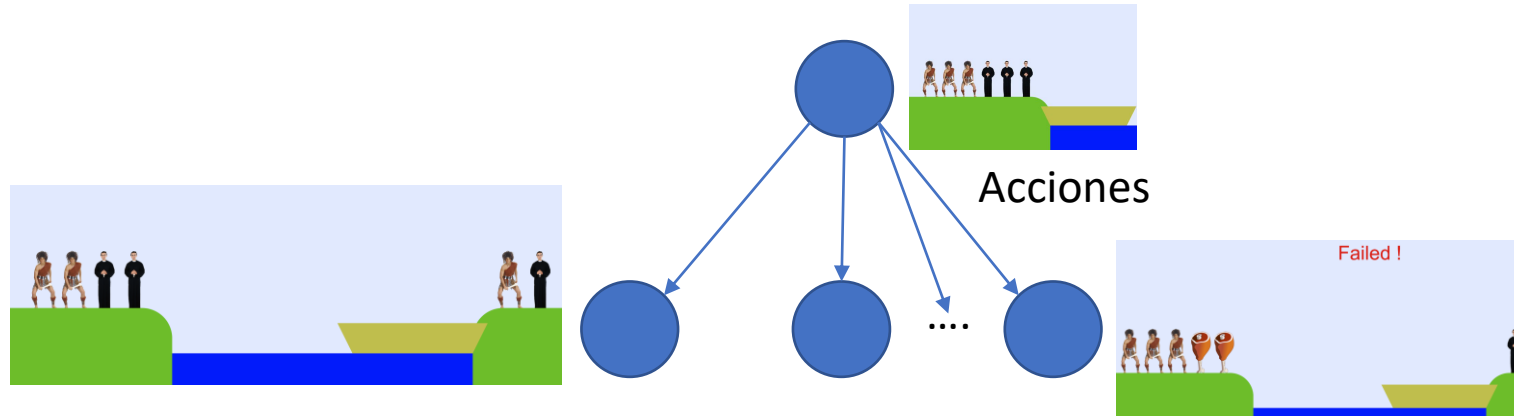
- Consiste en recorrer el espacio de búsqueda
  - Desde el estado inicial al estado final
  - Secuencia permitida de acciones según operadores





# BÚSQUEDA DE SOLUCIONES

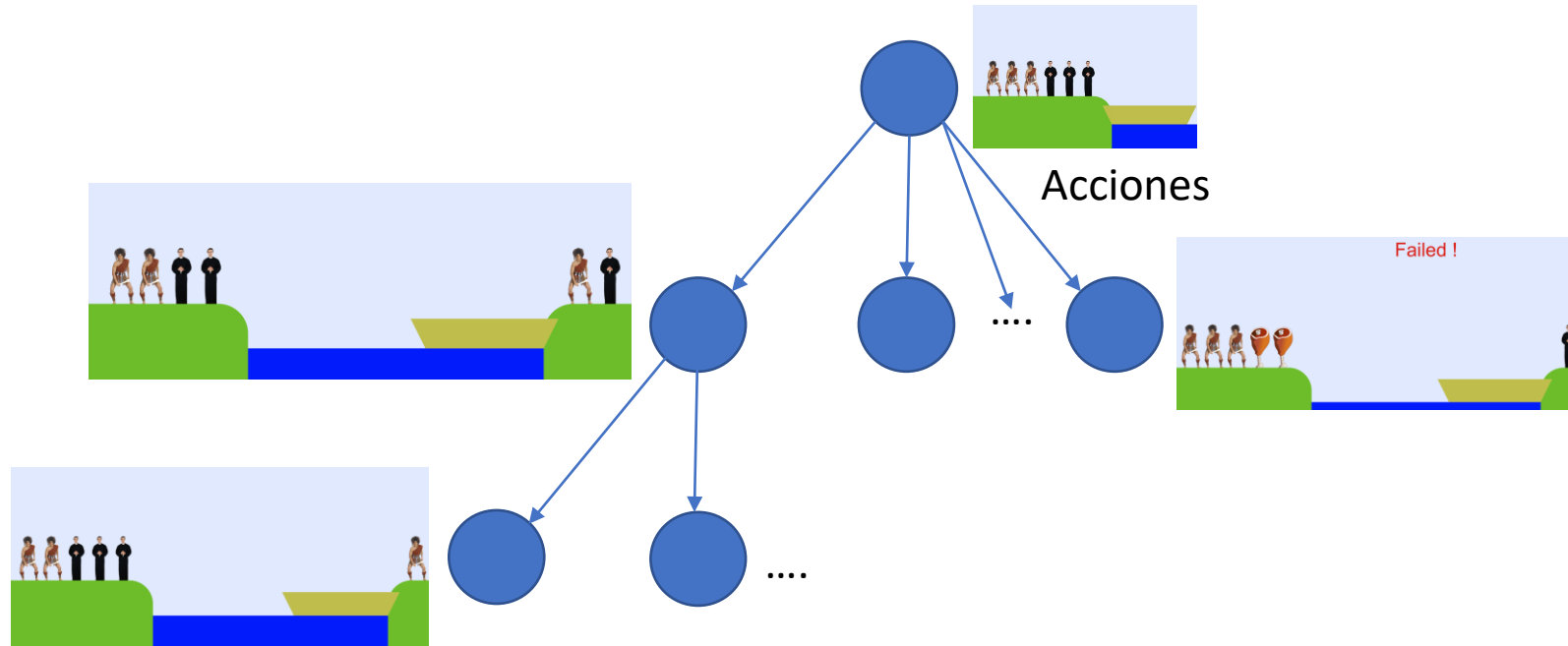
- Consiste en recorrer el espacio de búsqueda
  - Desde el estado inicial al estado final
  - Secuencia permitida de acciones según operadores





# BÚSQUEDA DE SOLUCIONES

- Consiste en recorrer el espacio de búsqueda
  - Desde el estado inicial al estado final
  - Secuencia permitida de acciones según operadores

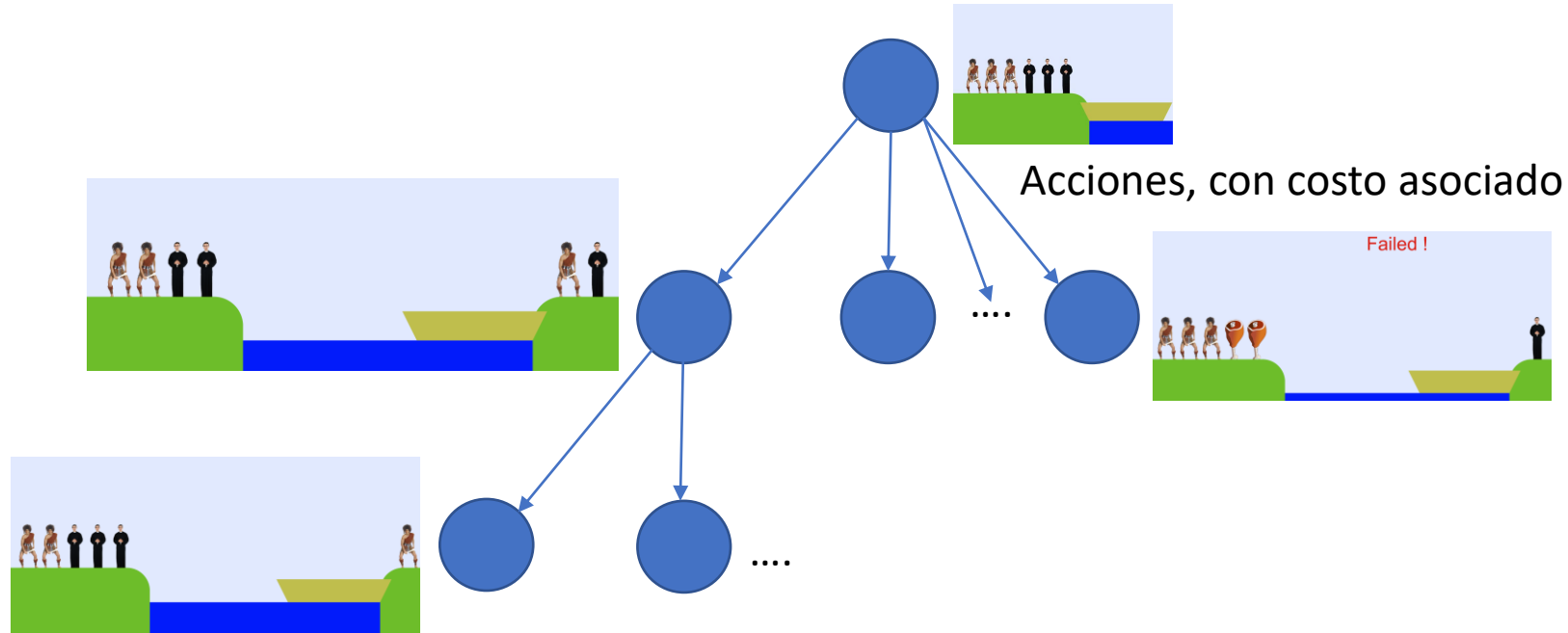






# BÚSQUEDA DE SOLUCIONES

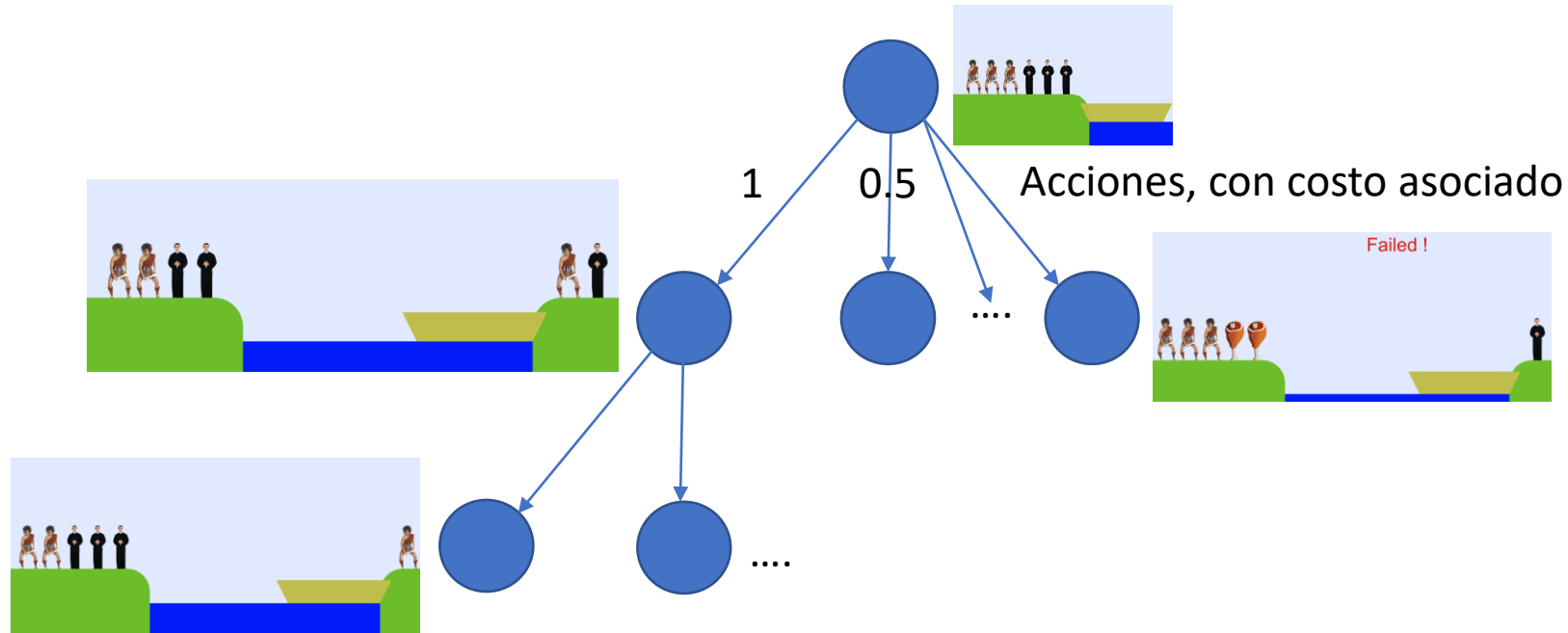
- Consiste en recorrer el espacio de búsqueda
  - Desde el estado inicial al estado final
  - Secuencia permitida de acciones según operadores





# BÚSQUEDA DE SOLUCIONES

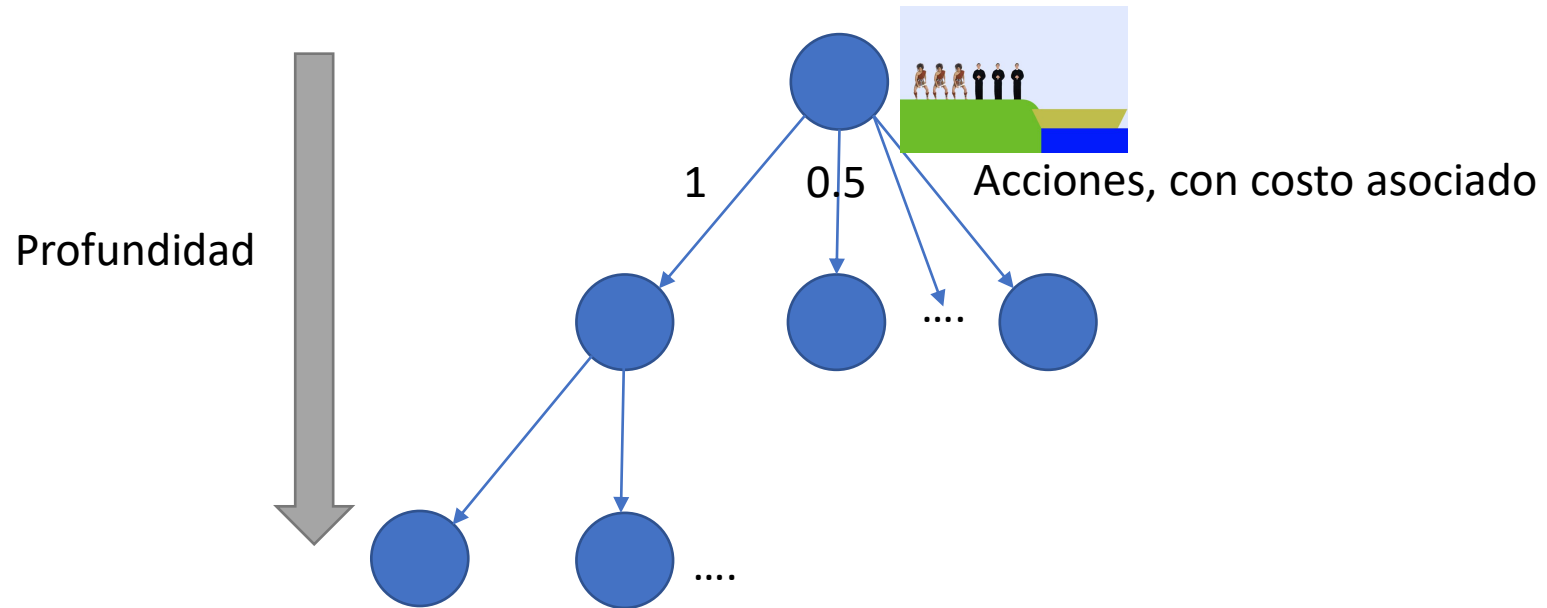
- Consiste en recorrer el espacio de búsqueda
  - Desde el estado inicial al estado final
  - Secuencia permitida de acciones según operadores





# BÚSQUEDA DE SOLUCIONES

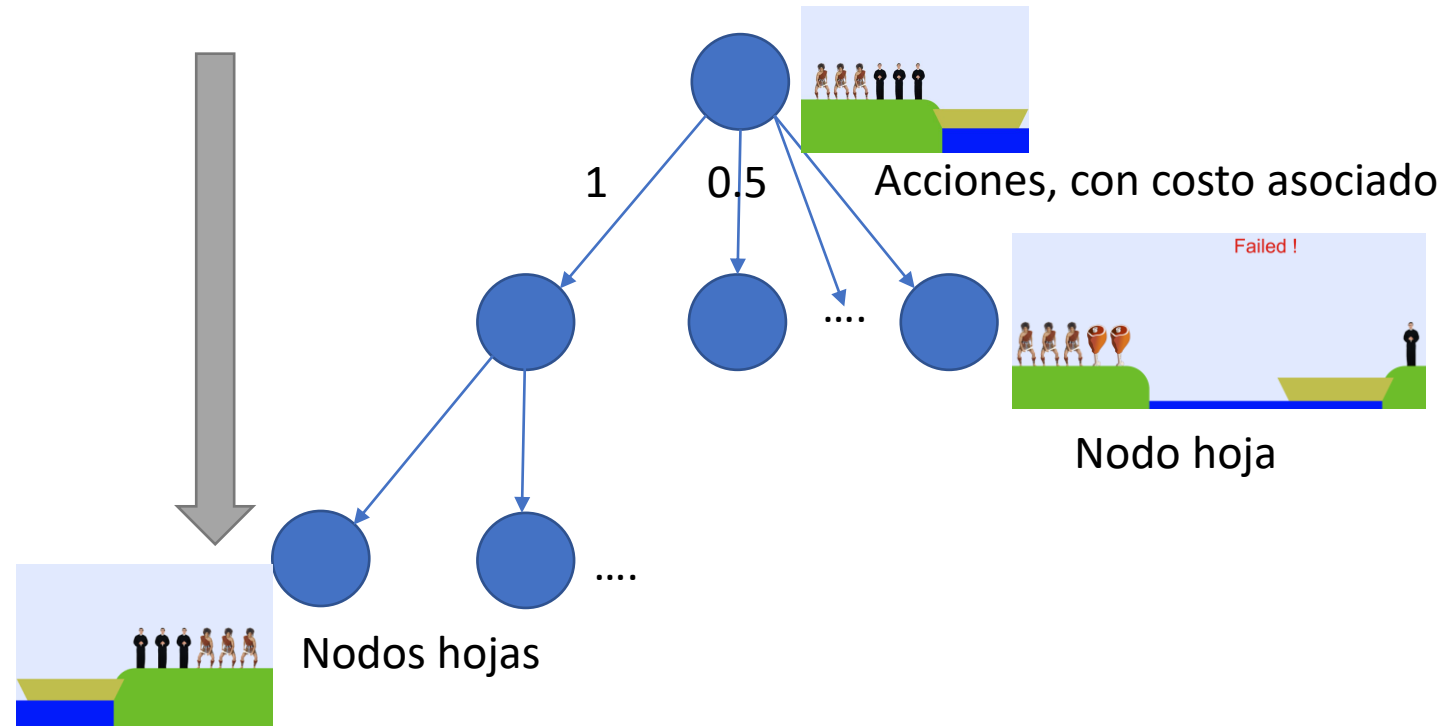
- Consiste en recorrer el espacio de búsqueda
  - Desde el estado inicial al estado final
  - Secuencia permitida de acciones según operadores





# BÚSQUEDA DE SOLUCIONES

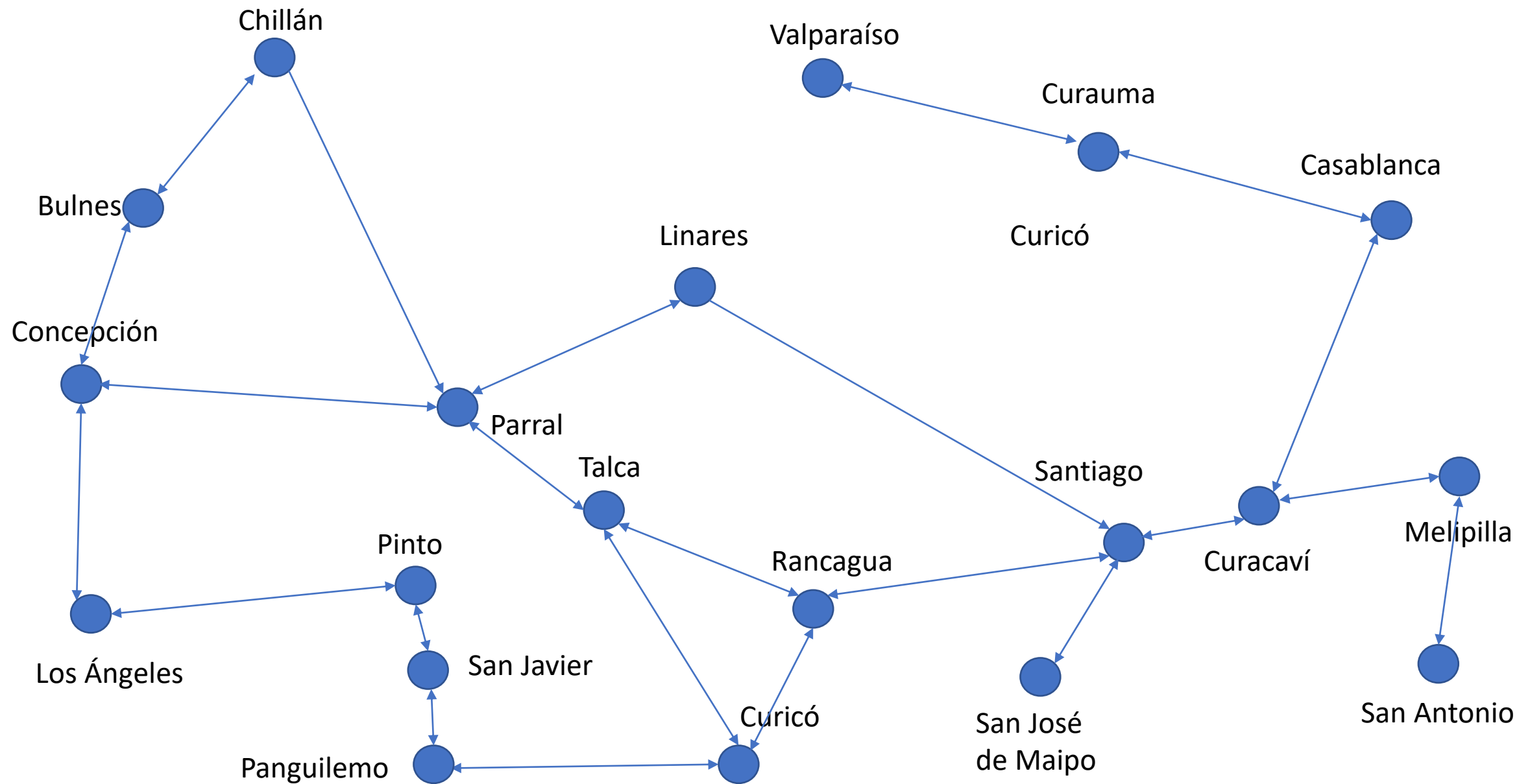
- Consiste en recorrer el espacio de búsqueda
  - Desde el estado inicial al estado final
  - Secuencia permitida de acciones según operadores





# BÚSQUEDA DE SOLUCIONES

- Procedimiento general:
  1. Realizar el test de objetivo al estado actual
  2. Expandir el estado actual
    1. Determinar el conjunto de estados accesibles
    2. Retornar “fracaso” si el conjunto está vacío
  3. Seleccionar un estado desde el conjunto de estados accesibles
  4. Moverse al estado seleccionado
- Un árbol de búsqueda es generado





# ESTRATEGIAS DE BÚSQUEDA

- Depende de la información sobre el problema
  - Búsqueda no informada (búsqueda a ciegas)
    - Número de pasos, costo del camino desconocido
    - El agente sabe cuando alcanza su meta
  - Búsqueda informada (búsqueda heurística)
    - El agente tiene información de respaldo sobre el problema
      - En específico, tiene una estimación de la distancia al objetivo



# CRITERIOS DE EVALUACIÓN

- ¿Encuentra una solución?
  - Si hay una solución, ¿se encontrará?
- ¿Encuentra la mejor solución?
- Costo computacional (tiempo y espacio)







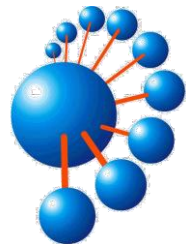
# ESTRATEGIAS DE BÚSQUEDA

- Búsqueda no informada
  - búsqueda en anchura
  - búsqueda en profundidad
  - búsqueda en profundidad limitada
  - profundización iterativa
  - búsqueda bidireccional
  - satisfacción de restricciones
- Búsqueda Informada
  - búsqueda best-first
  - búsqueda con heurísticas
  - búsqueda con memoria limitada
  - búsqueda con mejoramiento iterativo

# Inteligencia Artificial

## Métodos de Búsqueda no informada

Profesor: Julio Godoy  
Ayudante: Felipe Cerda  
DIICC



# BÚSQUEDA NO INFORMADA (A CIEGAS)

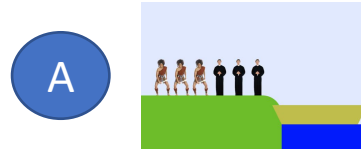
- Búsqueda en anchura
- Búsqueda en profundidad
  - variaciones



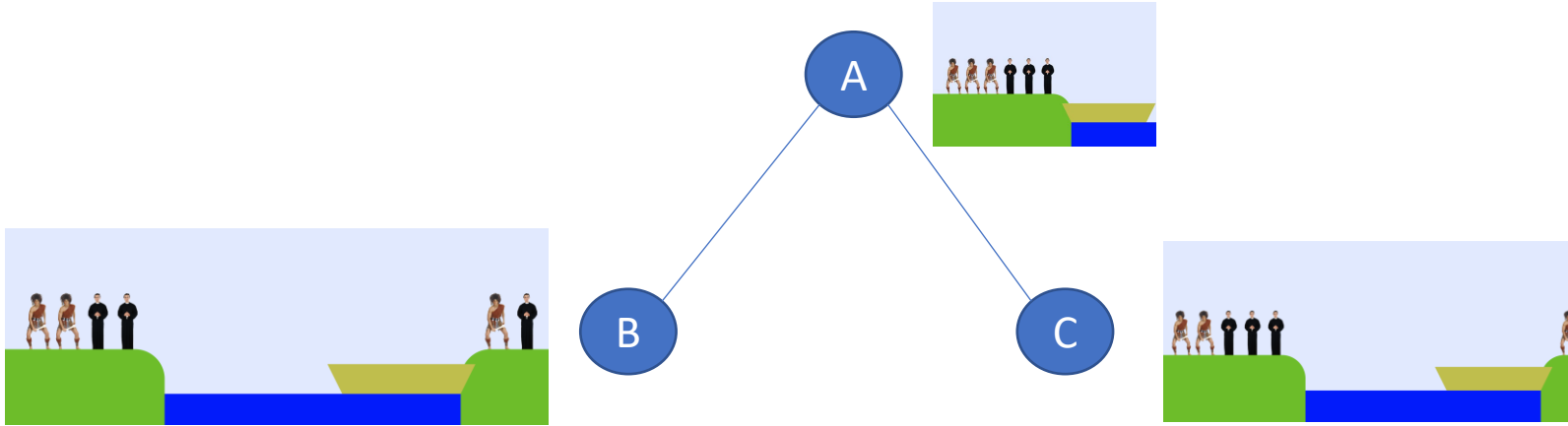
# BÚSQUEDA EN ANCHURA

- Todos los nodos alcanzables a partir del nodo actual son explorados primero.
  - Test objetivo se realiza al ***generar*** un nodo
  - Realiza una búsqueda sistemática del espacio de estados, un “nivel” a la vez

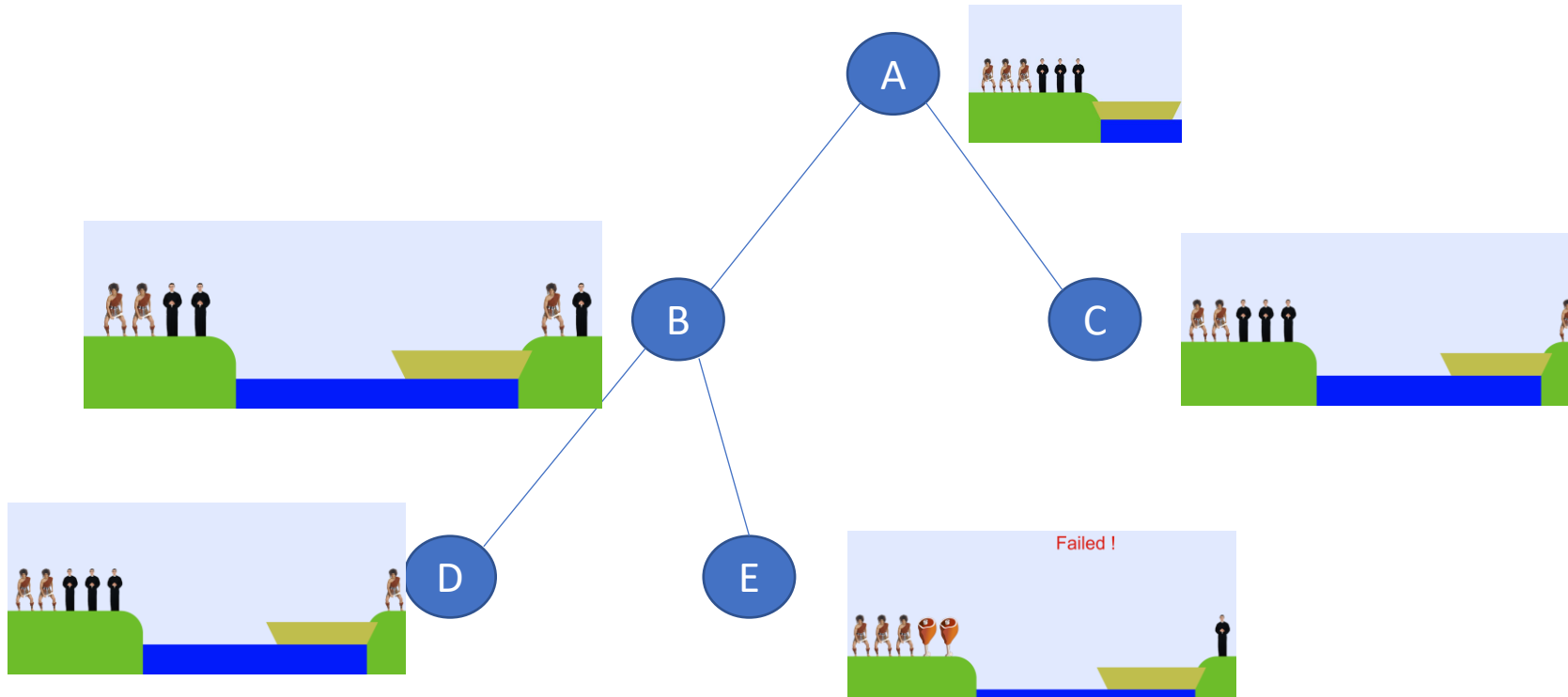
# BÚSQUEDA EN ANCHURA



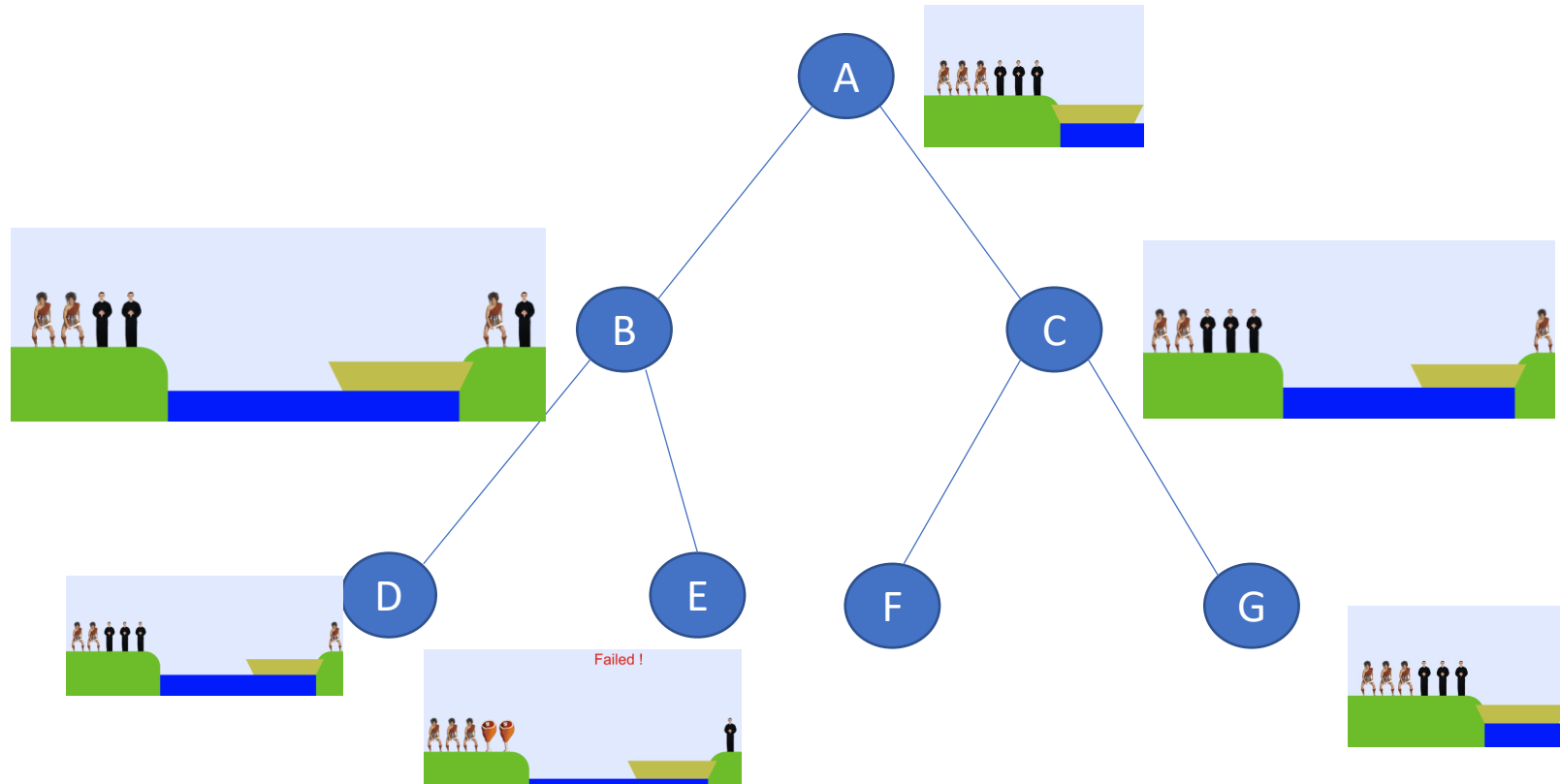
# BÚSQUEDA EN ANCHURA



# BÚSQUEDA EN ANCHURA

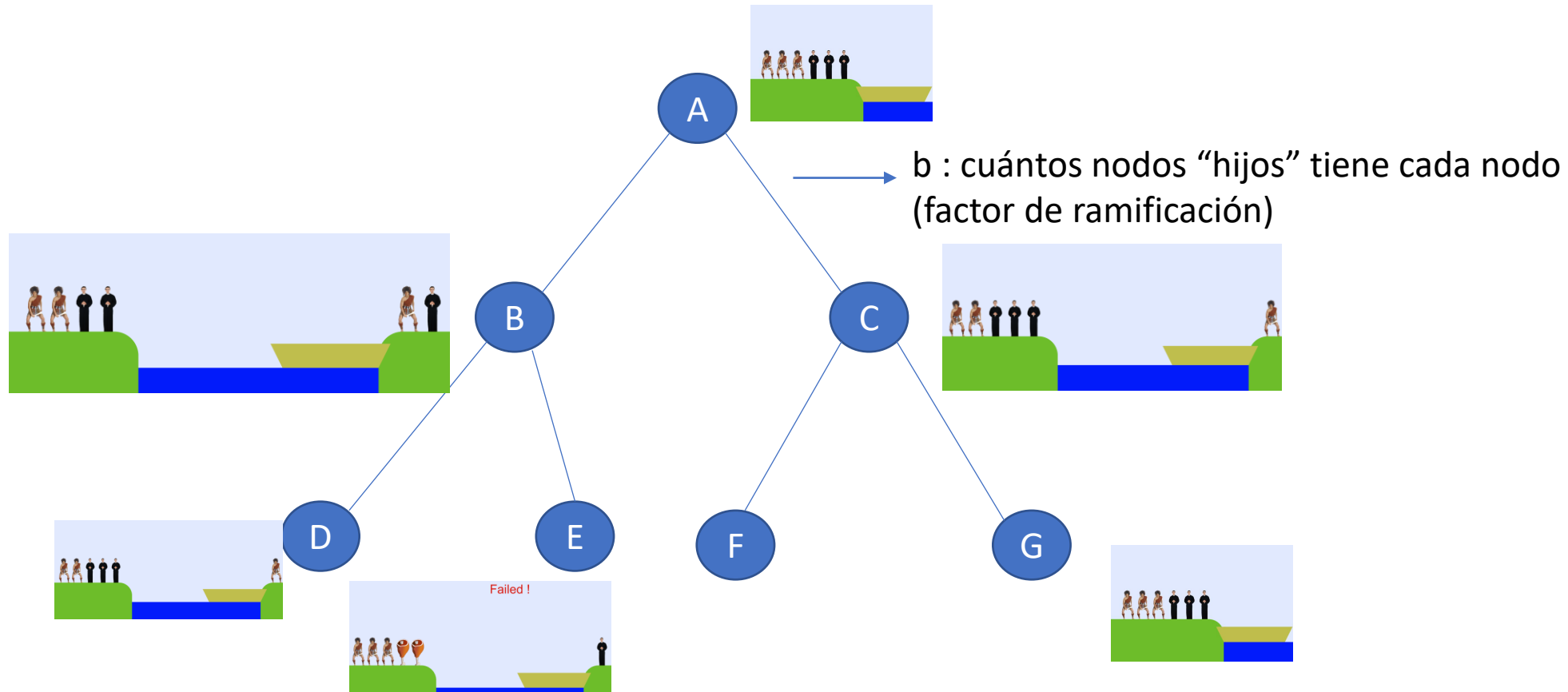


# BÚSQUEDA EN ANCHURA

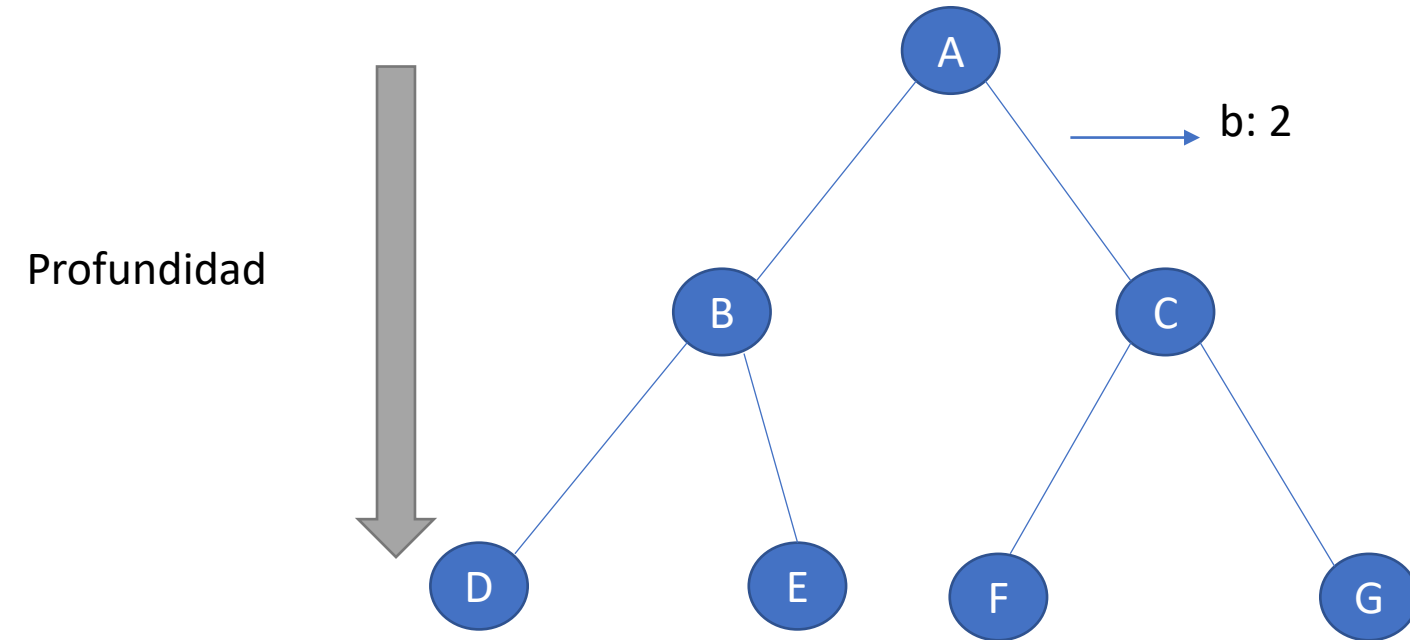




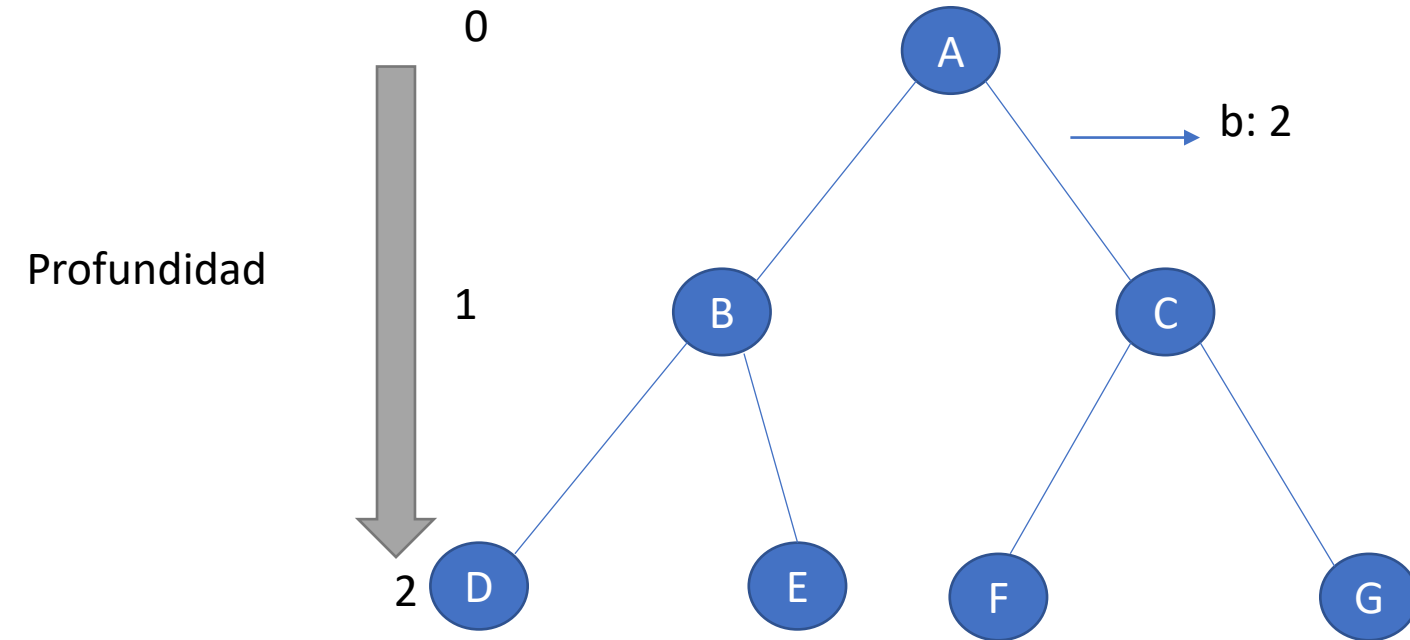
# BÚSQUEDA EN ANCHURA

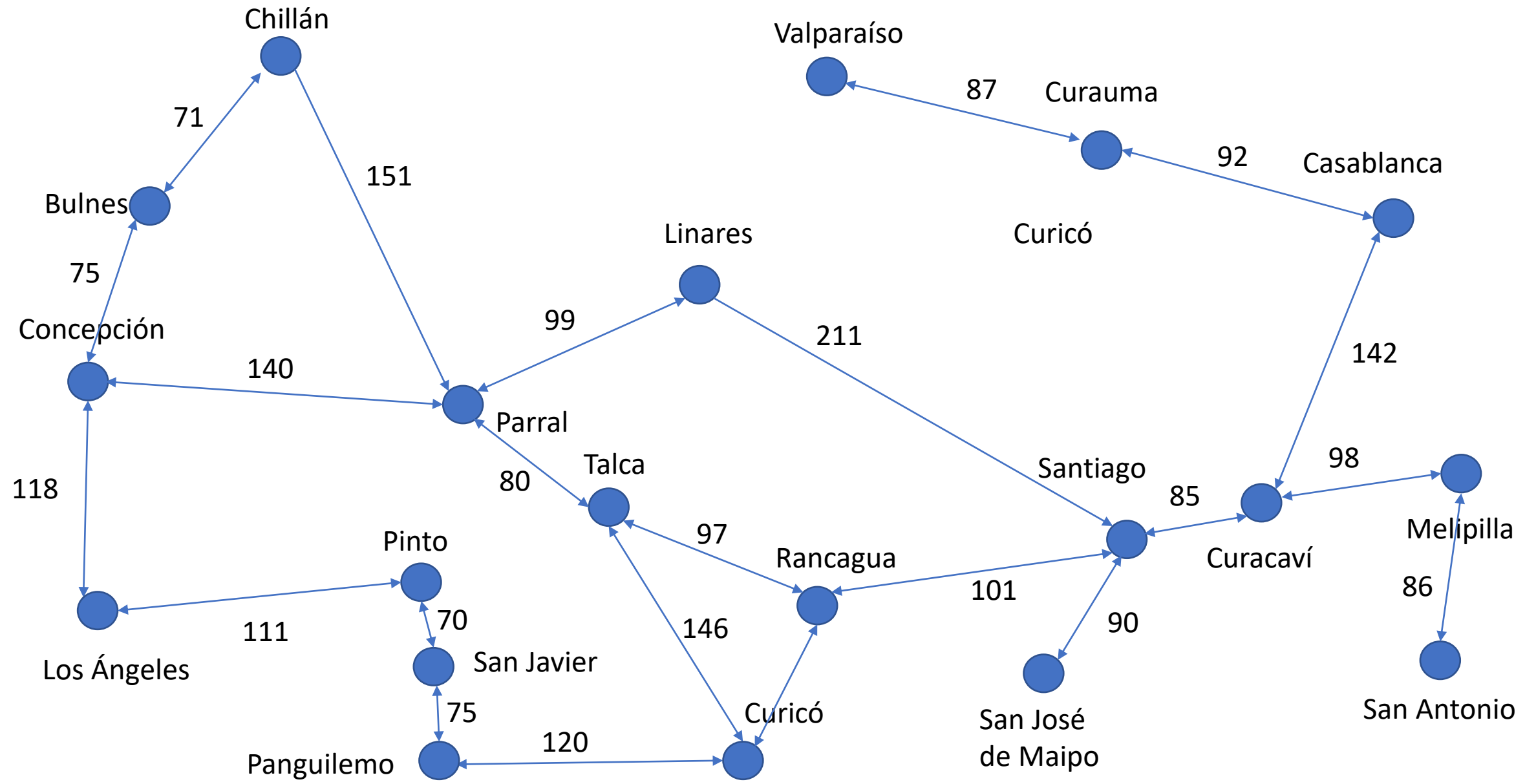


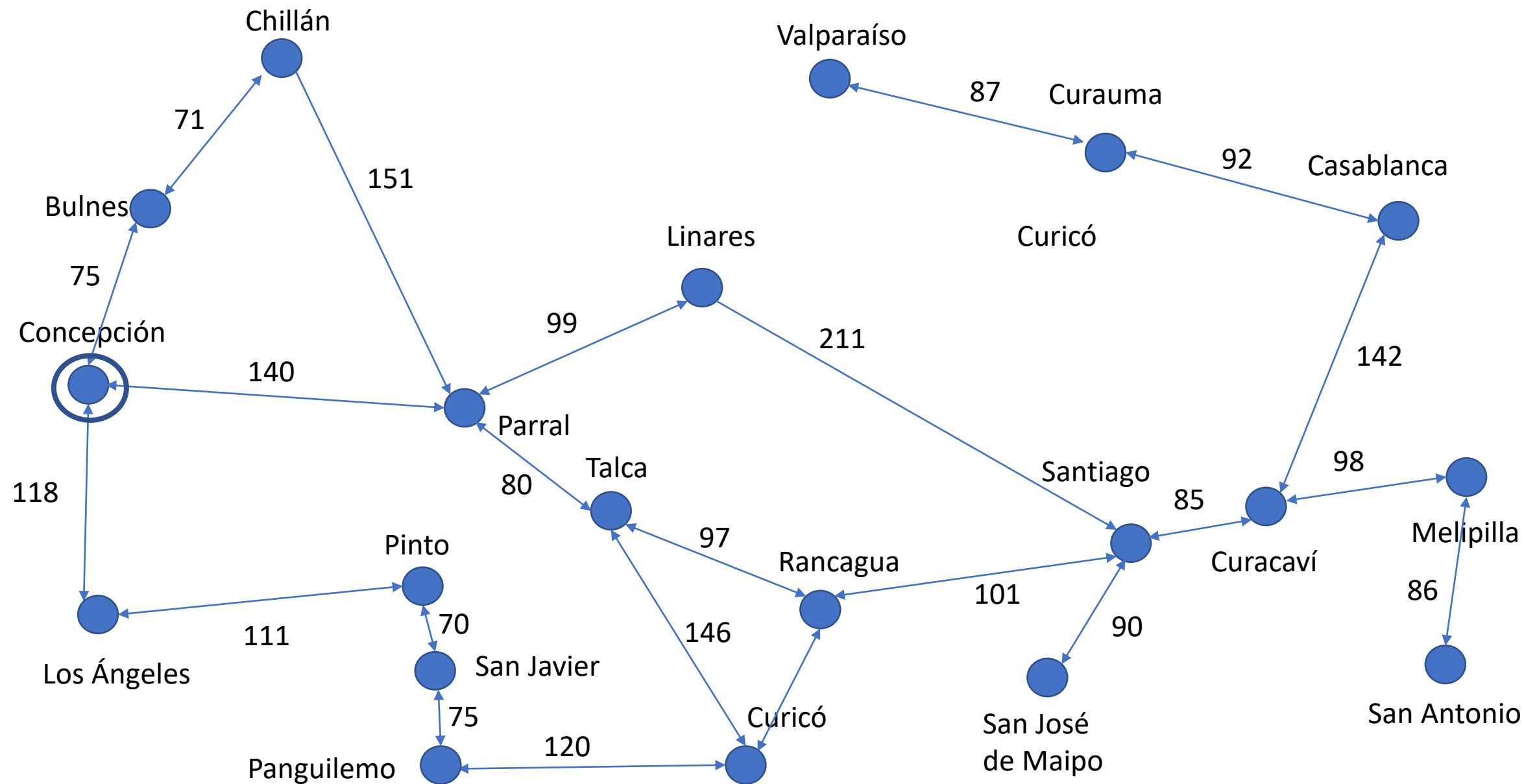
# BÚSQUEDA EN ANCHURA

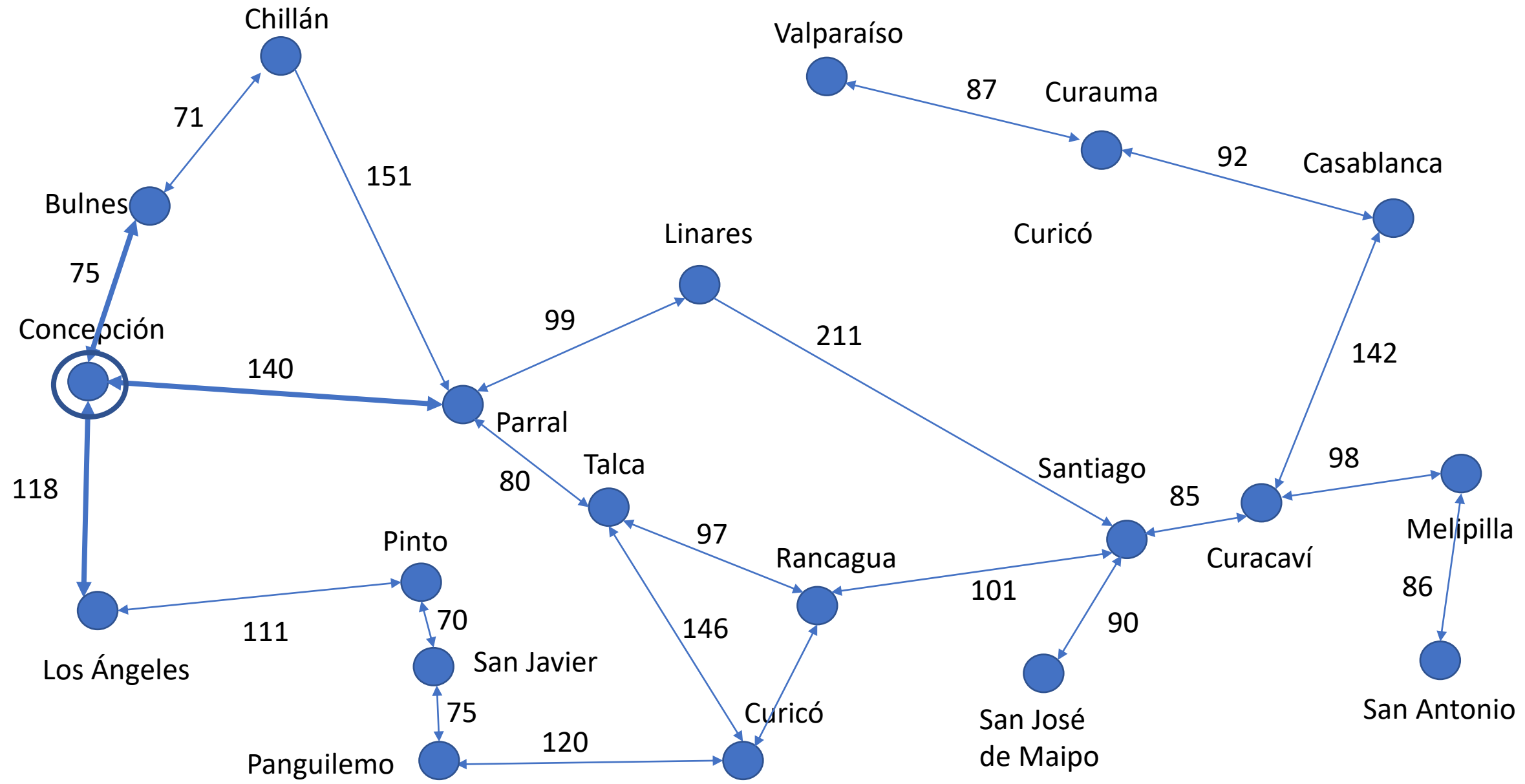


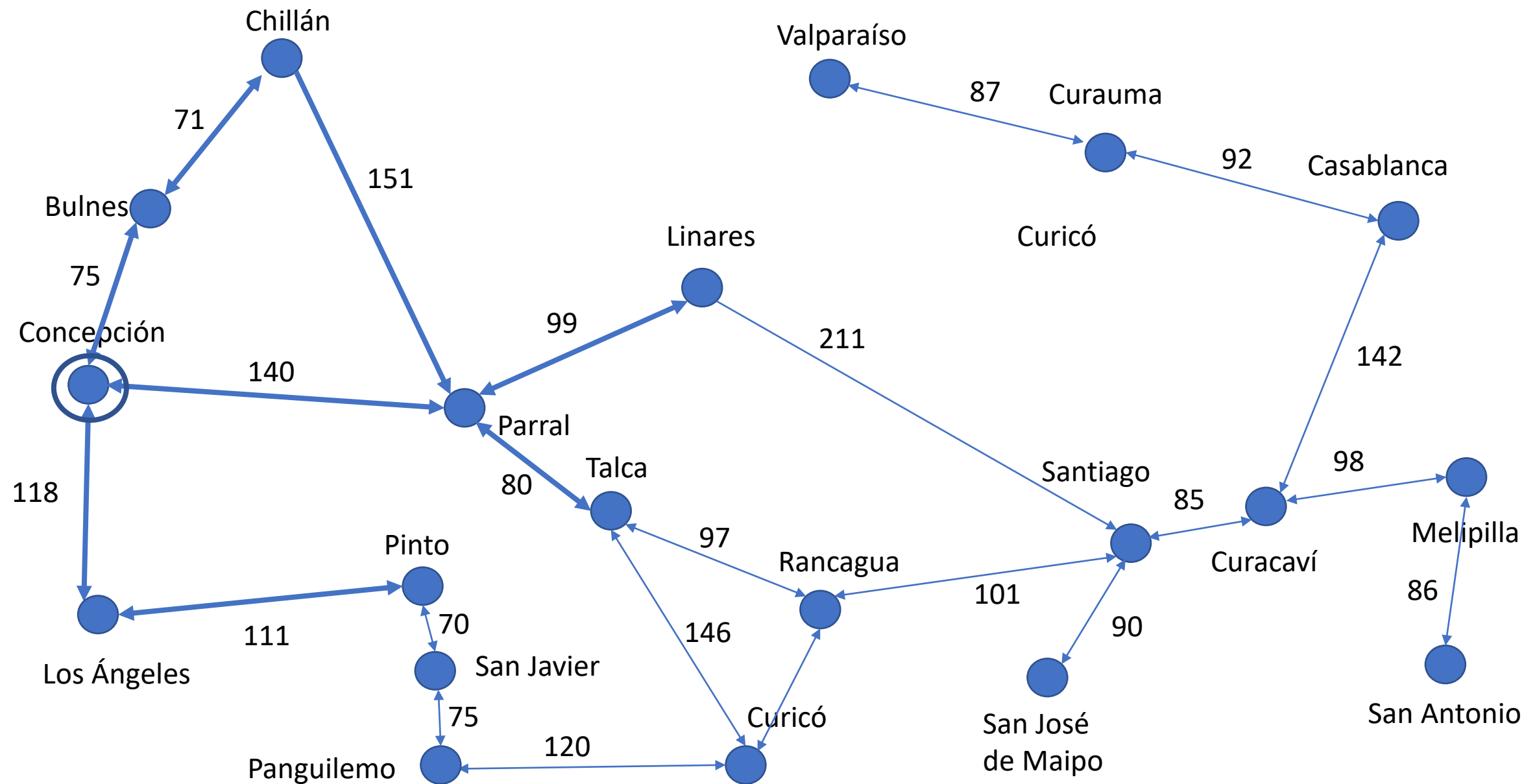
# BÚSQUEDA EN ANCHURA

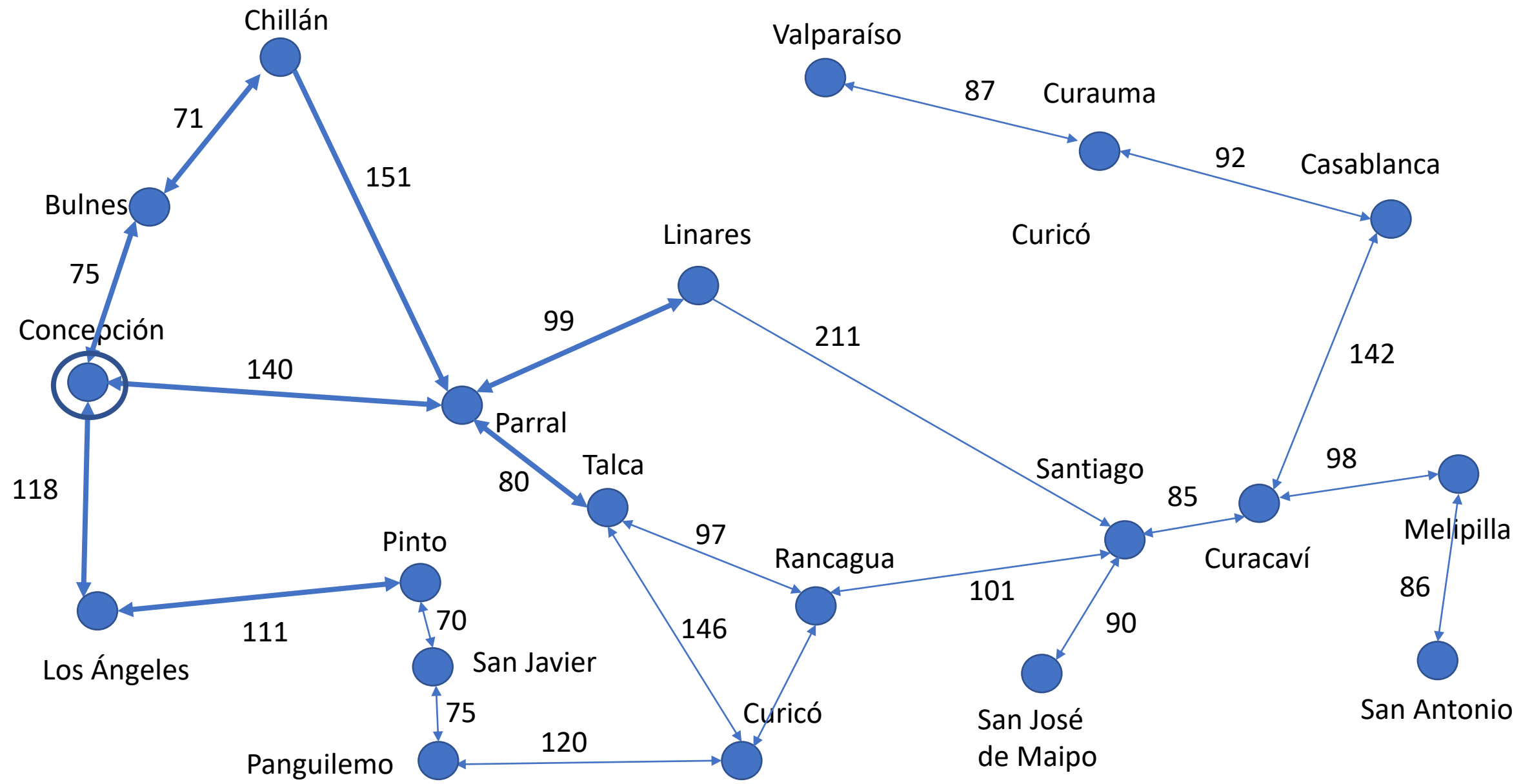




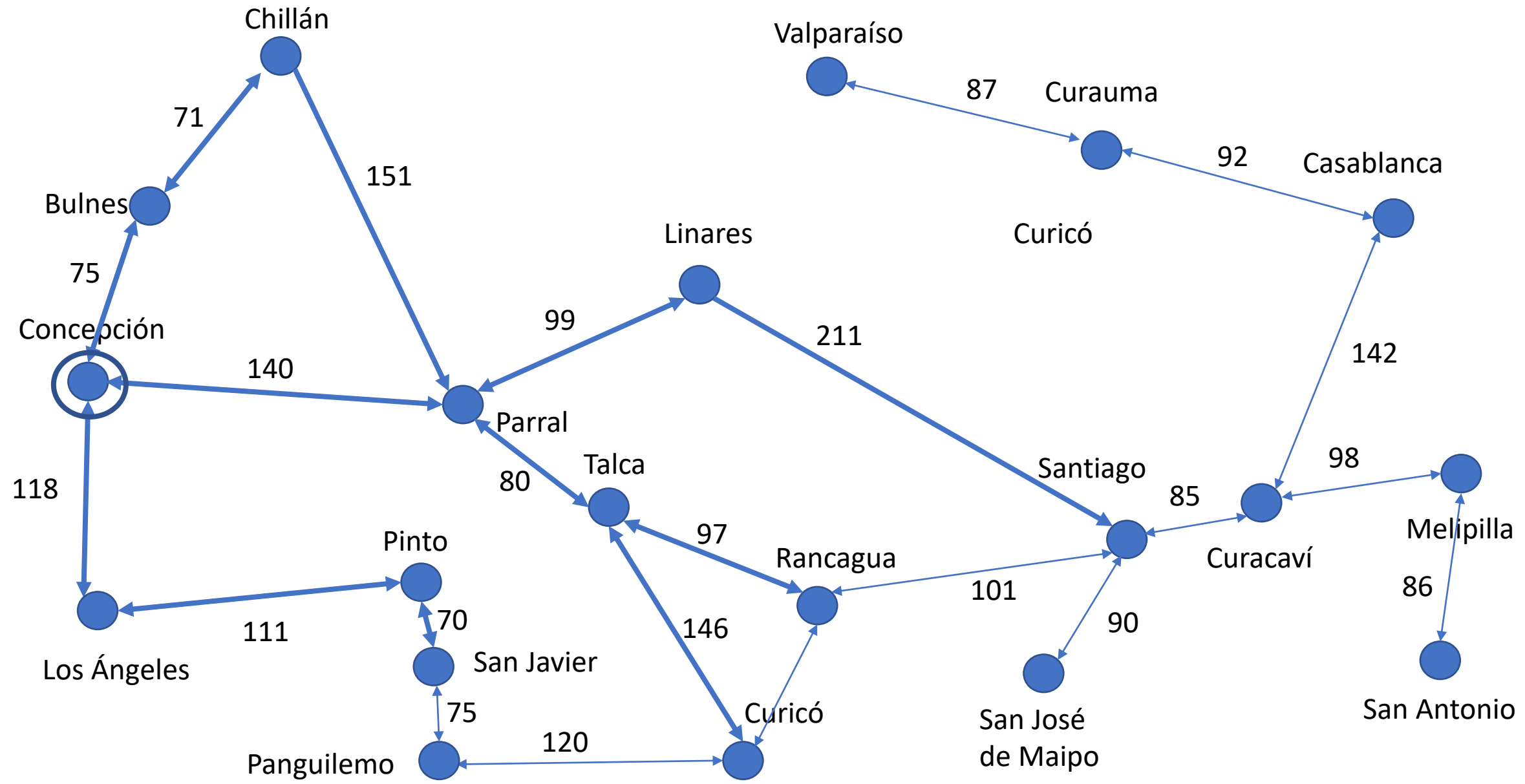


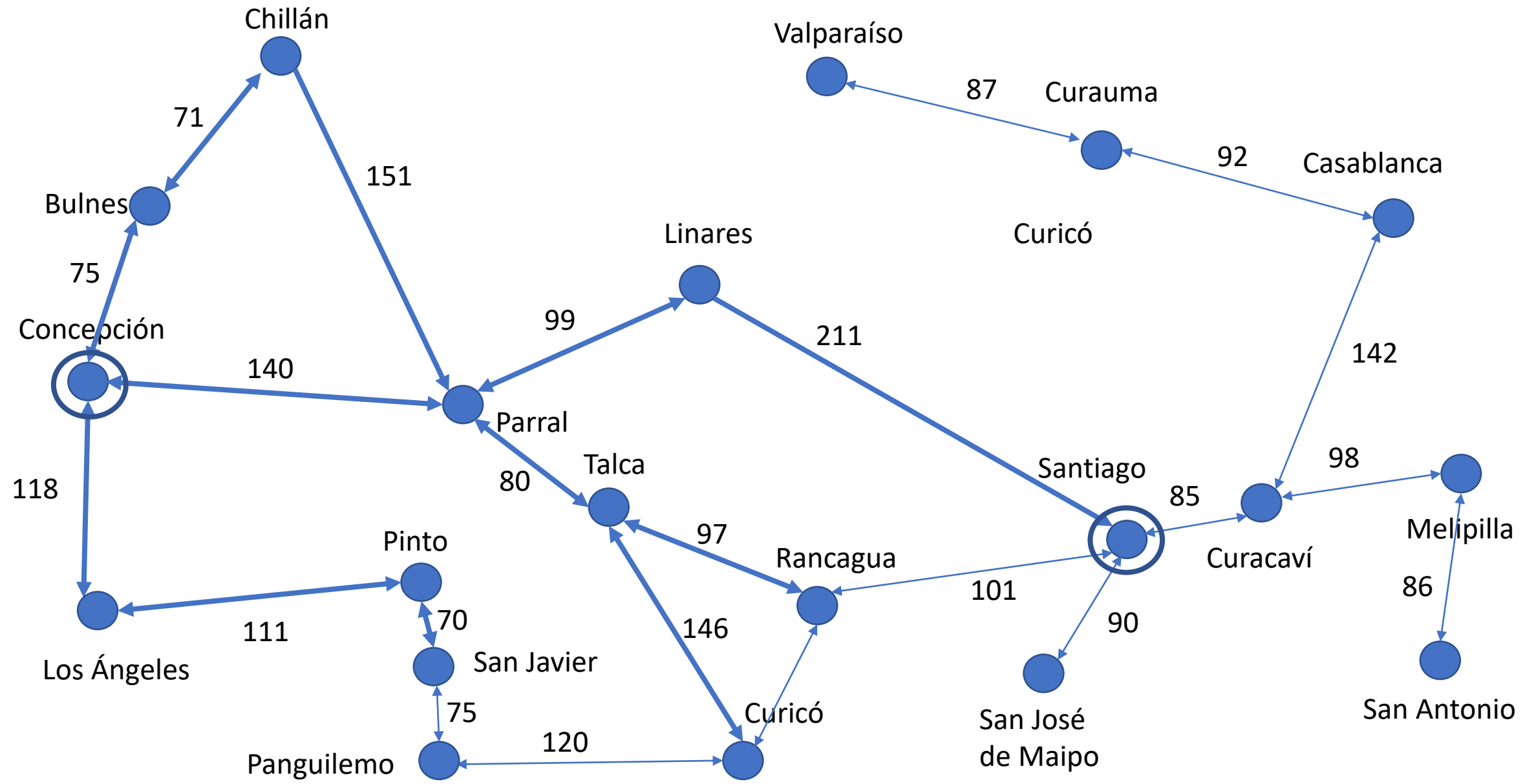








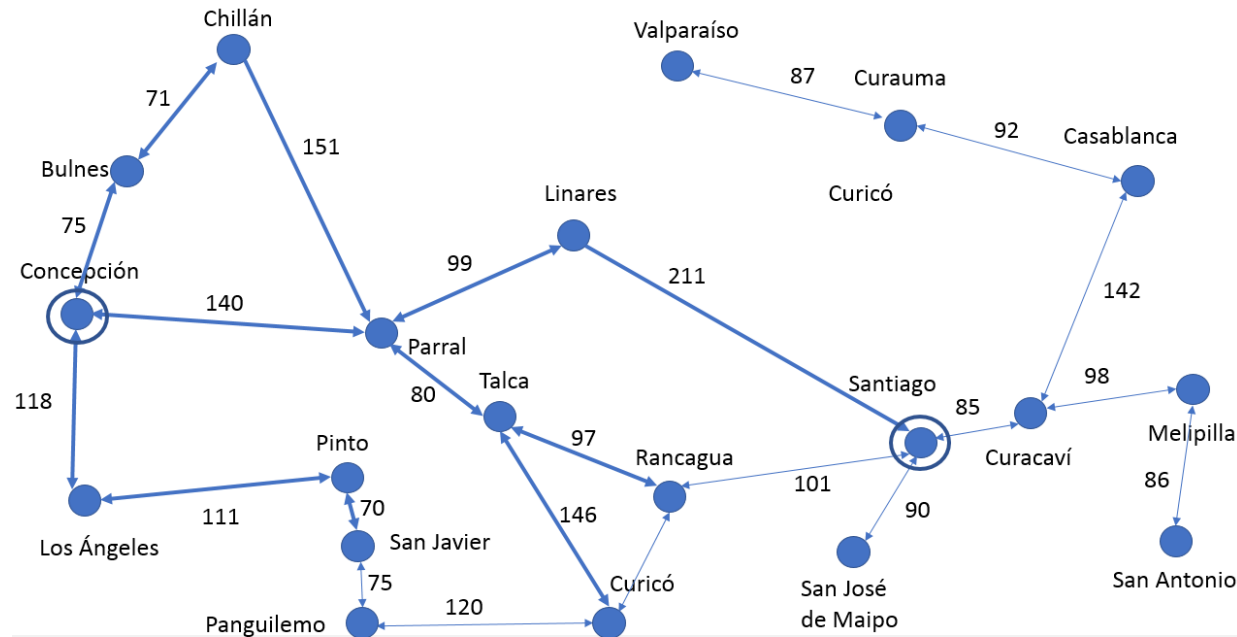




# BÚSQUEDA EN ANCHURA

- Solución:

Concepción – Parral - Linares - Santiago



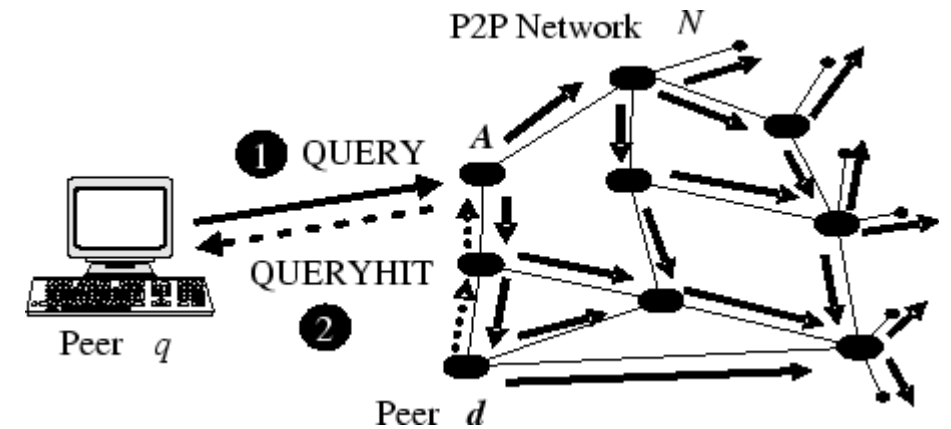
# BÚSQUEDA EN ANCHURA

Profundidad (d)	Nodos	Tiempo	Memoria
2	110	.11 miliseg.	107 KB
4	11110	11 miliseg.	10.6 MB
10	$10^{10}$	3 horas	10 TB
12	$10^{12}$	13 días	1 Petabyte
14	$10^{14}$	3.5 años	99 petabytes
16	$10^{16}$	350 años	10 exabytes

Suponiendo que  $b=10$ , 1 millón de nodos/seg. , 1000 bytes/nodo.

# BÚSQUEDA EN ANCHURA - APLICACIONES

- Redes P2P – Bit torrent
- Social networks
- Crawlers en motores de búsqueda
- Navegación GPS



# BÚSQUEDA EN ANCHURA - PROPIEDADES

- ¿Encuentra una solución?

# BÚSQUEDA EN ANCHURA - PROPIEDADES

- ¿Encuentra una solución?
  - Sí

# BÚSQUEDA EN ANCHURA - PROPIEDADES

- ¿Encuentra una solución?
  - Sí
- ¿Encuentra la mejor solución?



# BÚSQUEDA EN ANCHURA - PROPIEDADES

- ¿Encuentra una solución?
  - Sí
- ¿Encuentra la mejor solución?
  - Sí, con una condición

# BÚSQUEDA EN ANCHURA - PROPIEDADES

- ¿Encuentra una solución?
  - Sí
- ¿Encuentra la mejor solución?
  - Sí, con una condición
- Costo computacional

# BÚSQUEDA EN ANCHURA - PROPIEDADES

- ¿Encuentra una solución?
  - Sí
- ¿Encuentra la mejor solución?
  - Sí, con una condición
- Costo computacional
  - Muy alto 😞



qiao.github.io/PathFinding.js/visual/



Games



Cómo se está integr...



PathFinding.js



Tutorial- Robot loca...



COMP 152: Prob...

## Instructions

hide

Click within the white grid and drag your mouse to draw obstacles.

Drag the **green** node to set the start position.

Drag the **red** node to set the end position.

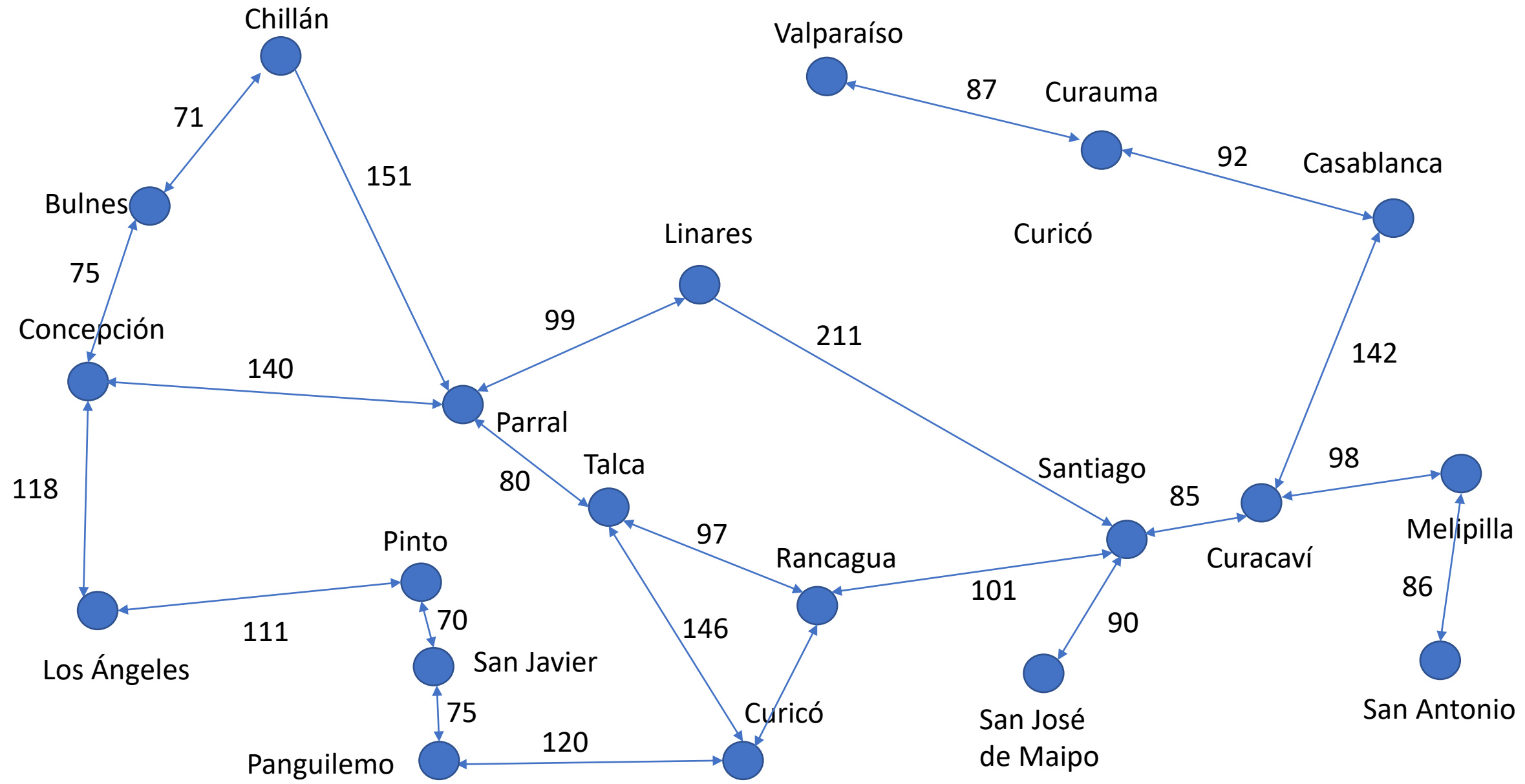
Choose an algorithm from the right-hand panel.

Click Start Search in the lower-right corner to start the animation.



# BÚSQUEDA CON COSTO UNIFORME

- Similar a la búsqueda por anchura
- En lugar de escoger el nodo más superficial para expansión → nodo con menor costo de camino
- Test objetivo se realiza al ***expandir*** un nodo



# BÚSQUEDA CON COSTO UNIFORME

- Completitud? (  $\epsilon > 0$  )
- Complejidad temporal? (b: cantidad de acciones,  $C^*$ : costo sol. óptima)
- Complejidad espacial?
- Optimalidad?

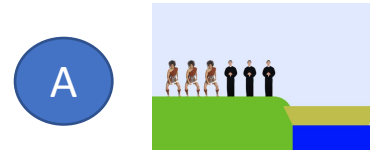
# BÚSQUEDA CON COSTO UNIFORME

- Completitud? (  $\epsilon > 0$  )
- Complejidad temporal? (b: cantidad de acciones,  $C^*$ : costo sol. óptima)
  - $O(b^{1+(C^*/\epsilon)})$
- Complejidad espacial?
- Optimalidad?



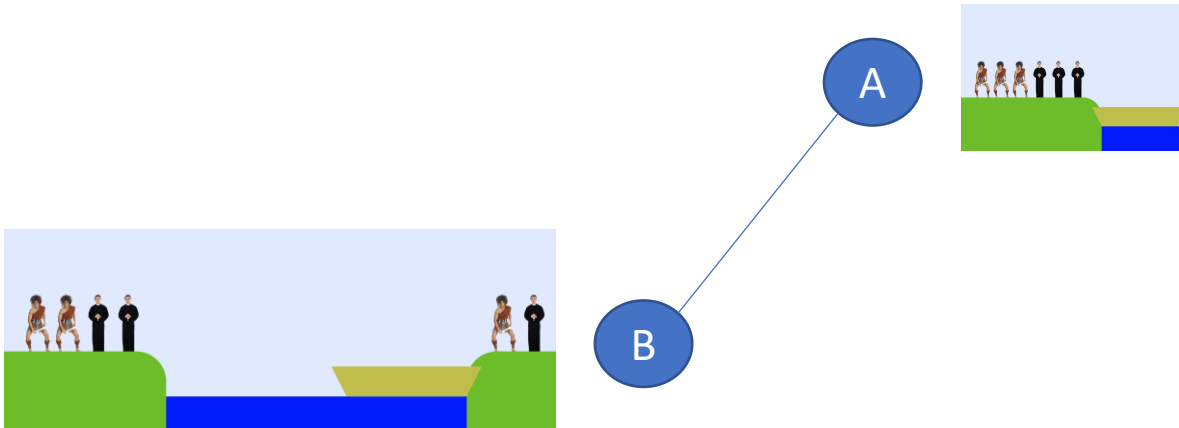
# BÚSQUEDA EN PROFUNDIDAD

- Expande el primer nodo generado, de forma iterativa



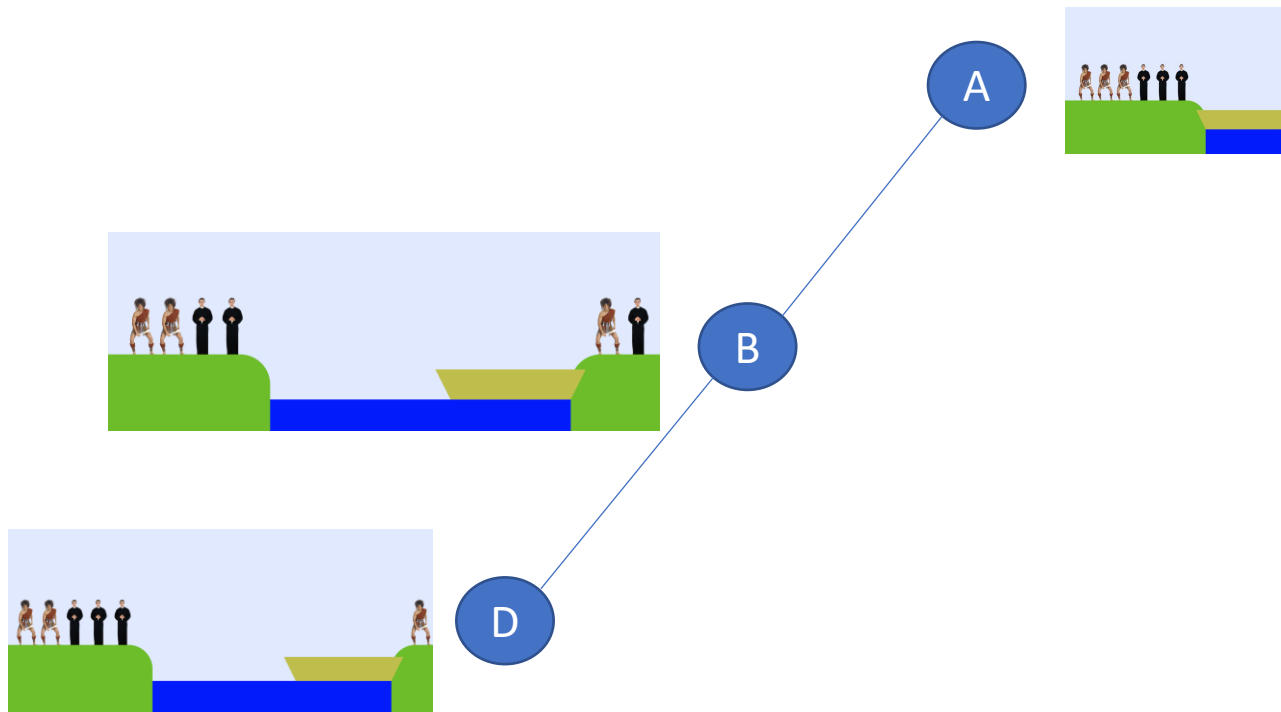
# BÚSQUEDA EN PROFUNDIDAD

- Expande el primer nodo generado, de forma iterativa



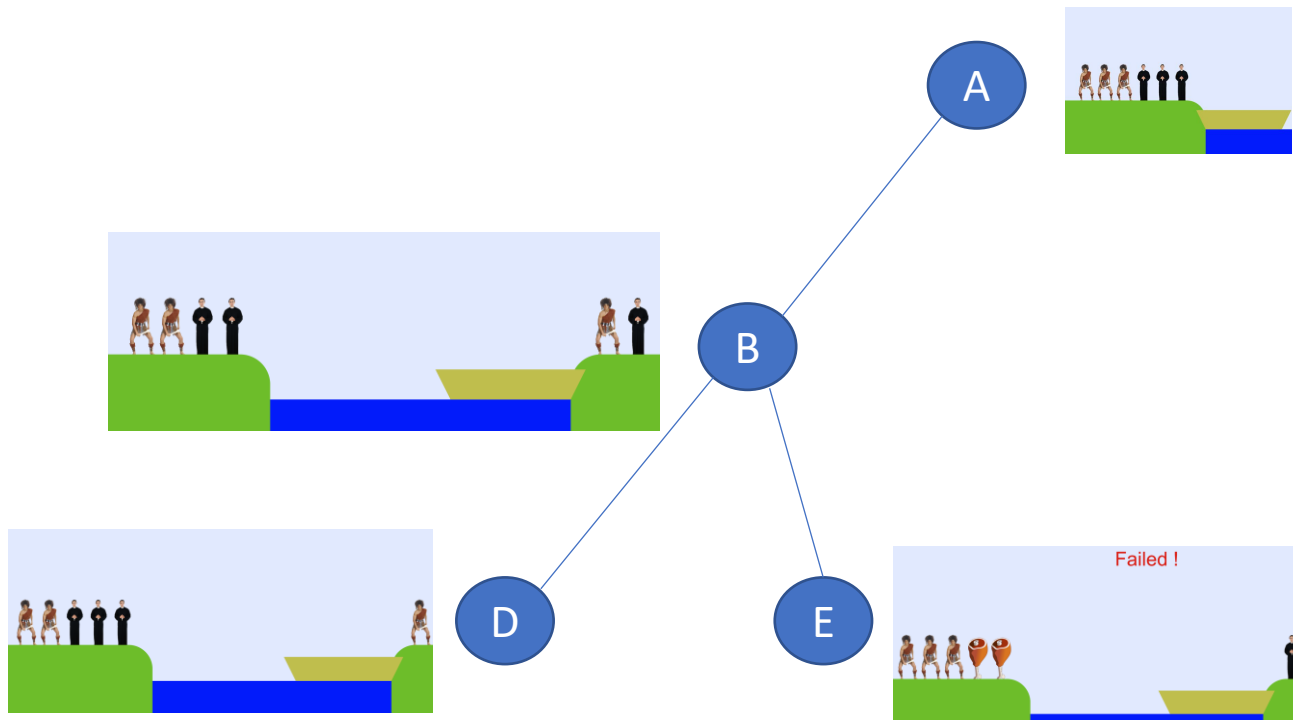
# BÚSQUEDA EN PROFUNDIDAD

- Expande el primer nodo generado, de forma iterativa



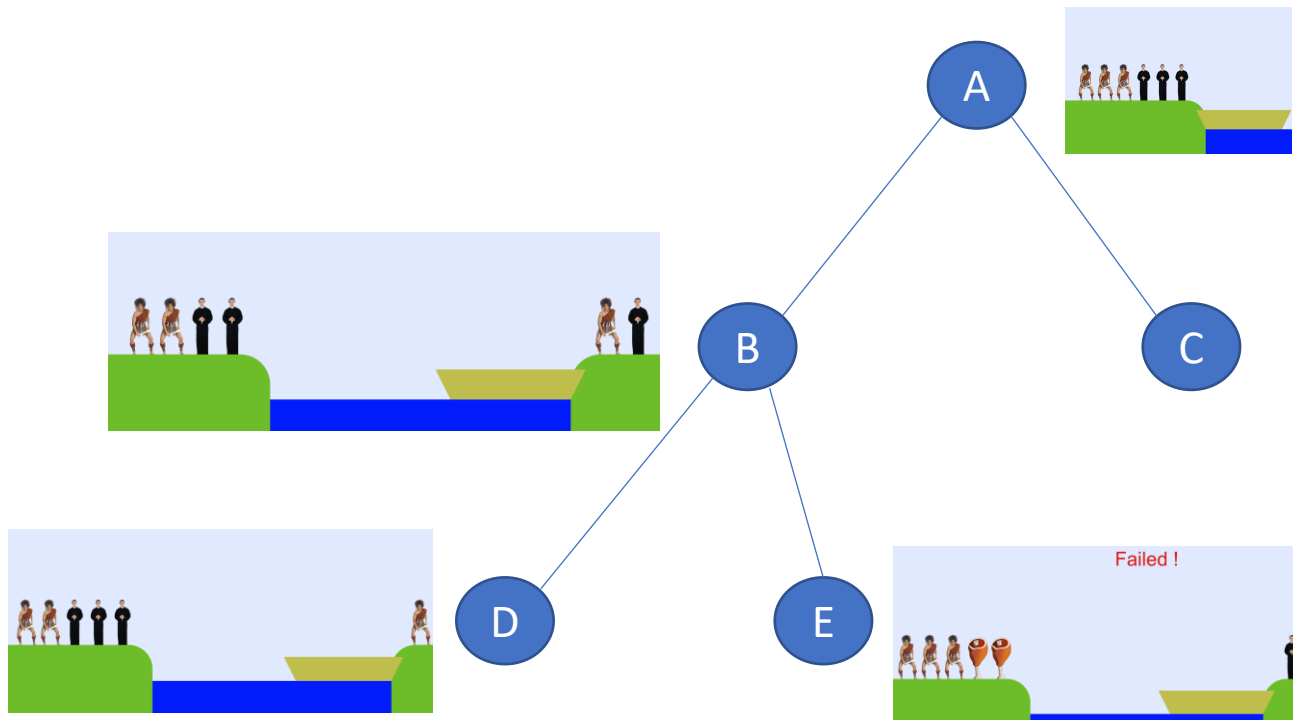
# BÚSQUEDA EN PROFUNDIDAD

- Expande el primer nodo generado, de forma iterativa



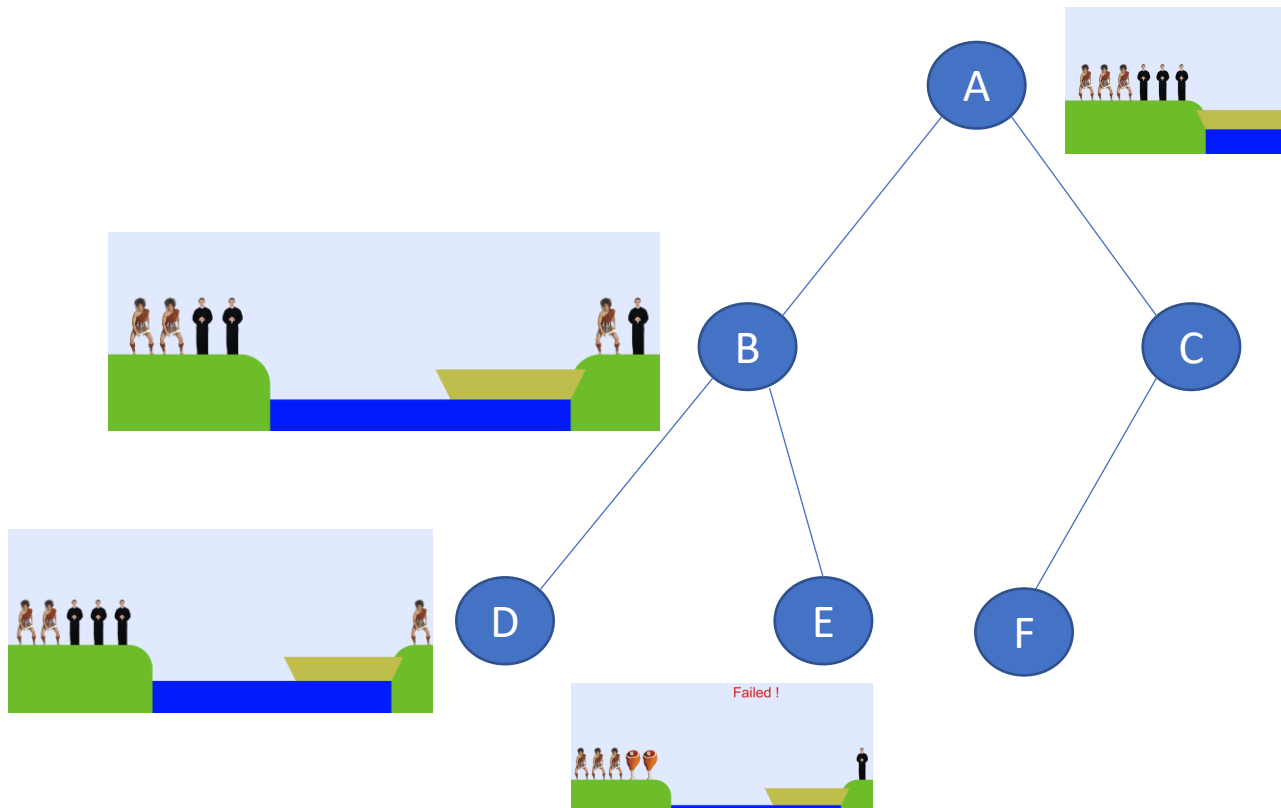
# BÚSQUEDA EN PROFUNDIDAD

- Expande el primer nodo generado, de forma iterativa



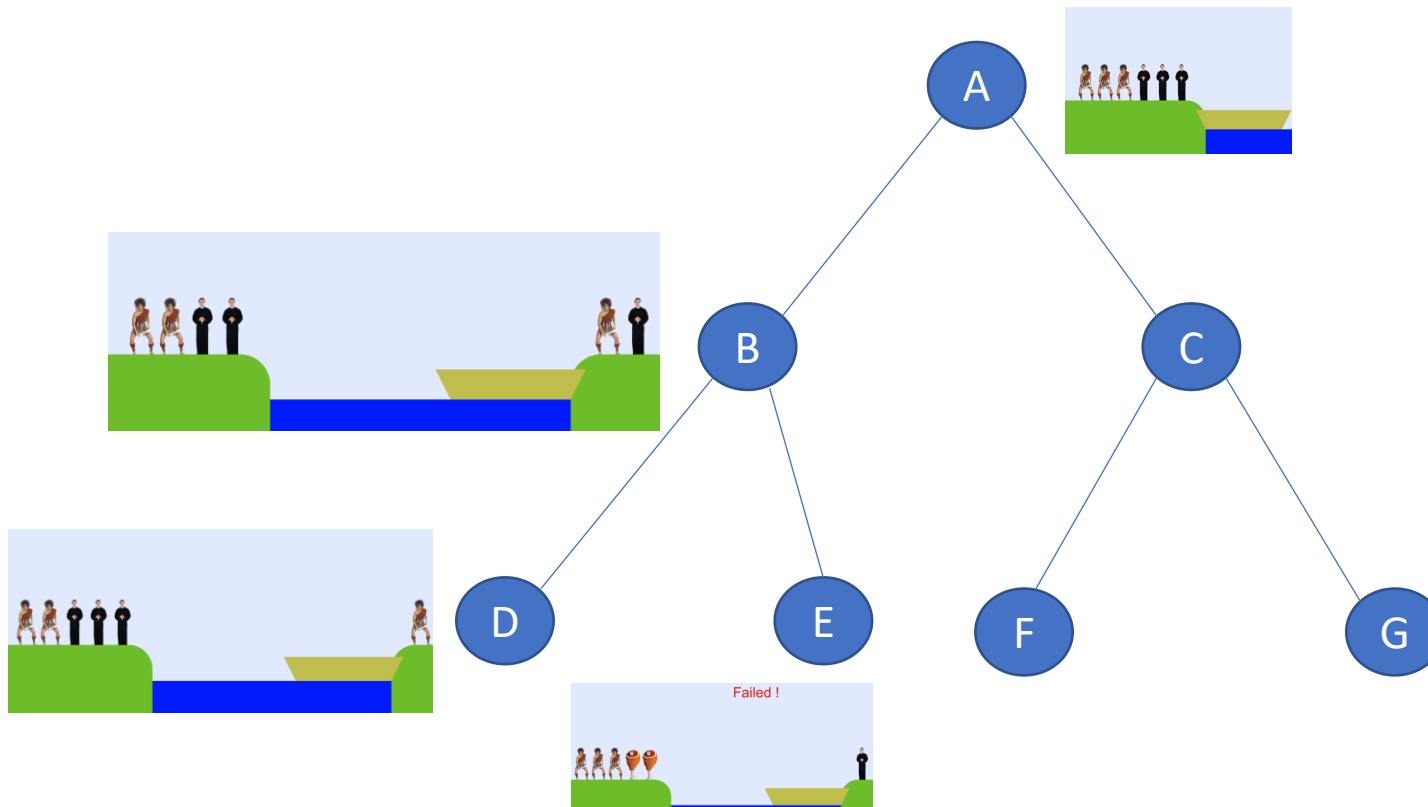
# BÚSQUEDA EN PROFUNDIDAD

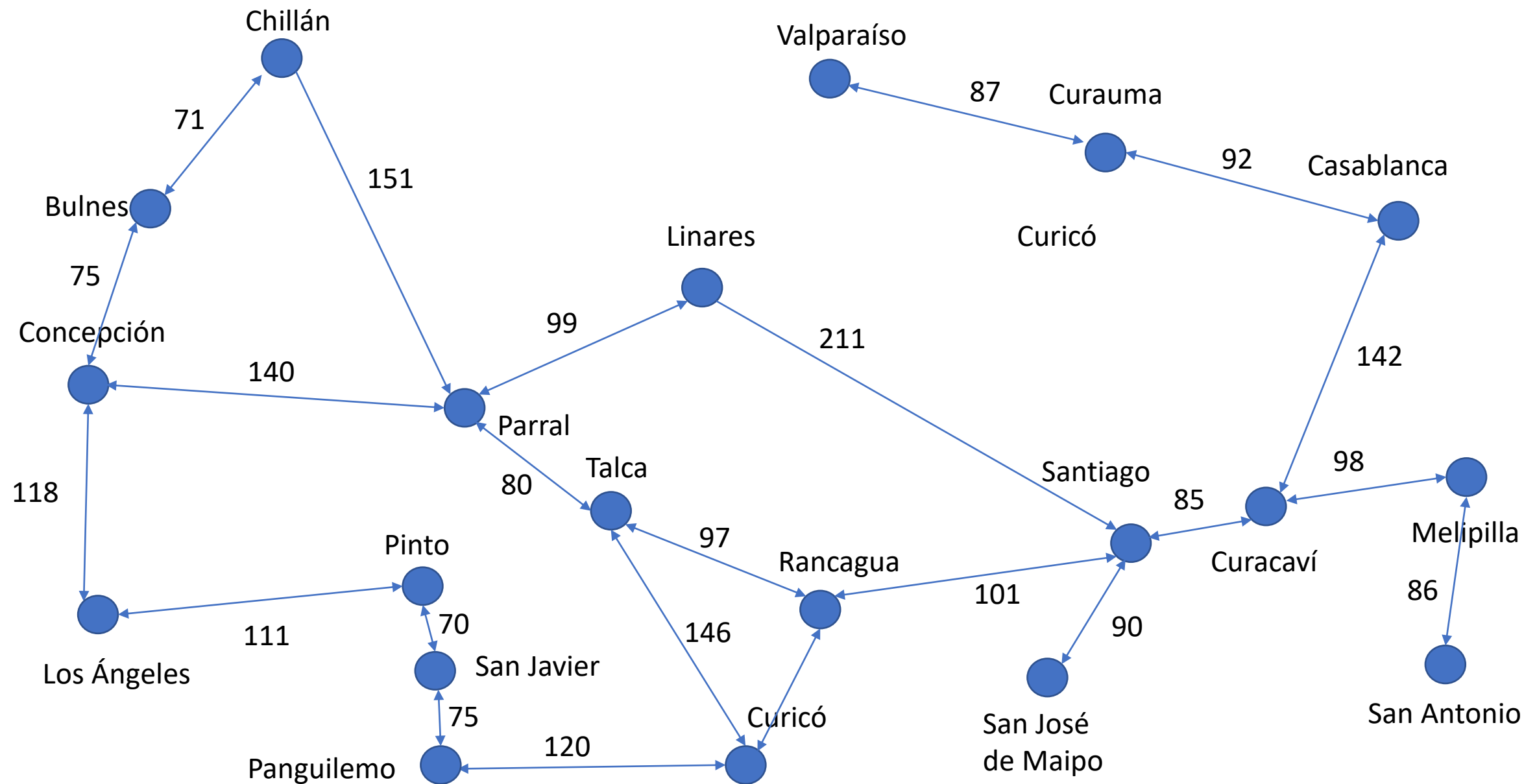
- Expande el primer nodo generado, de forma iterativa



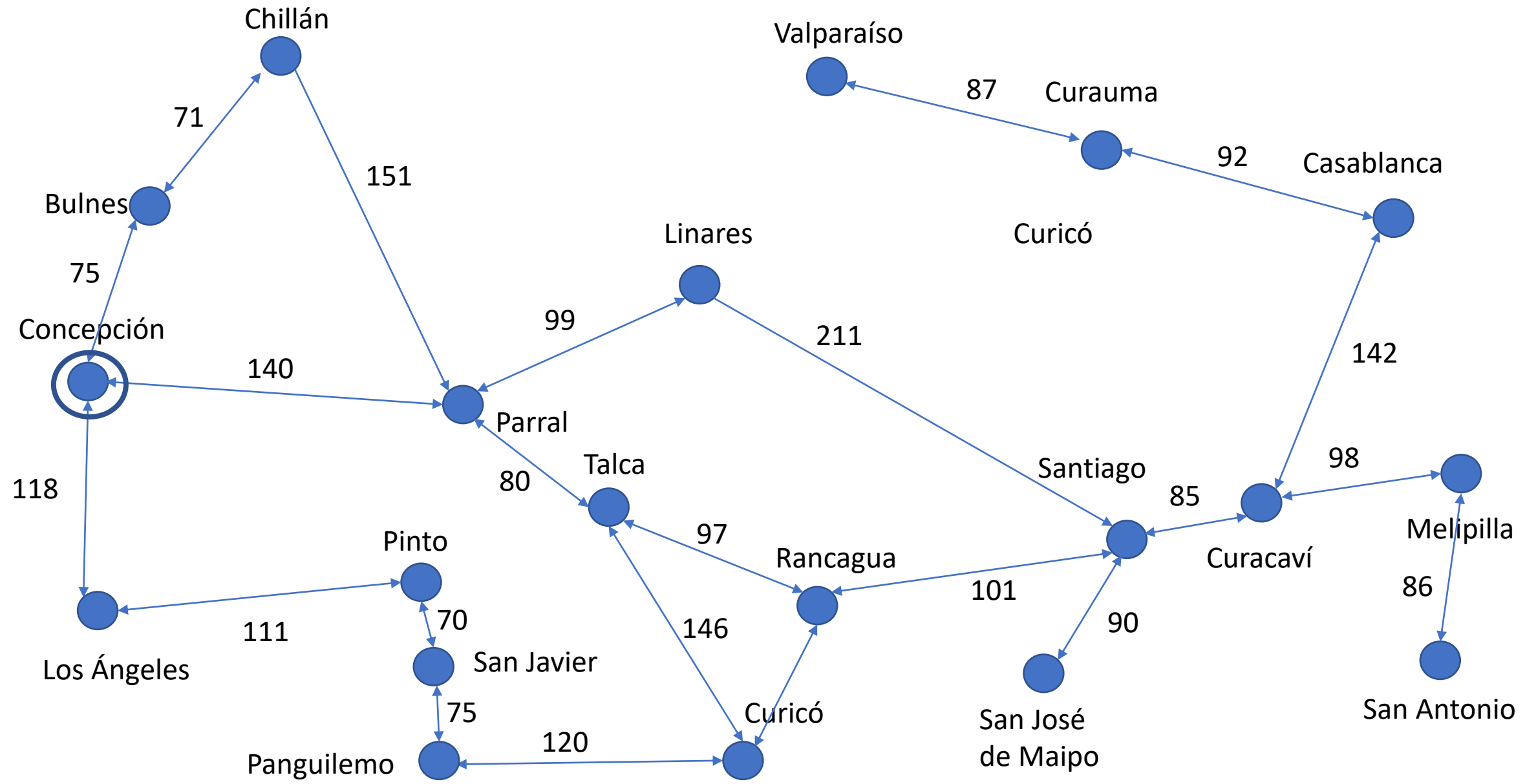
# BÚSQUEDA EN PROFUNDIDAD

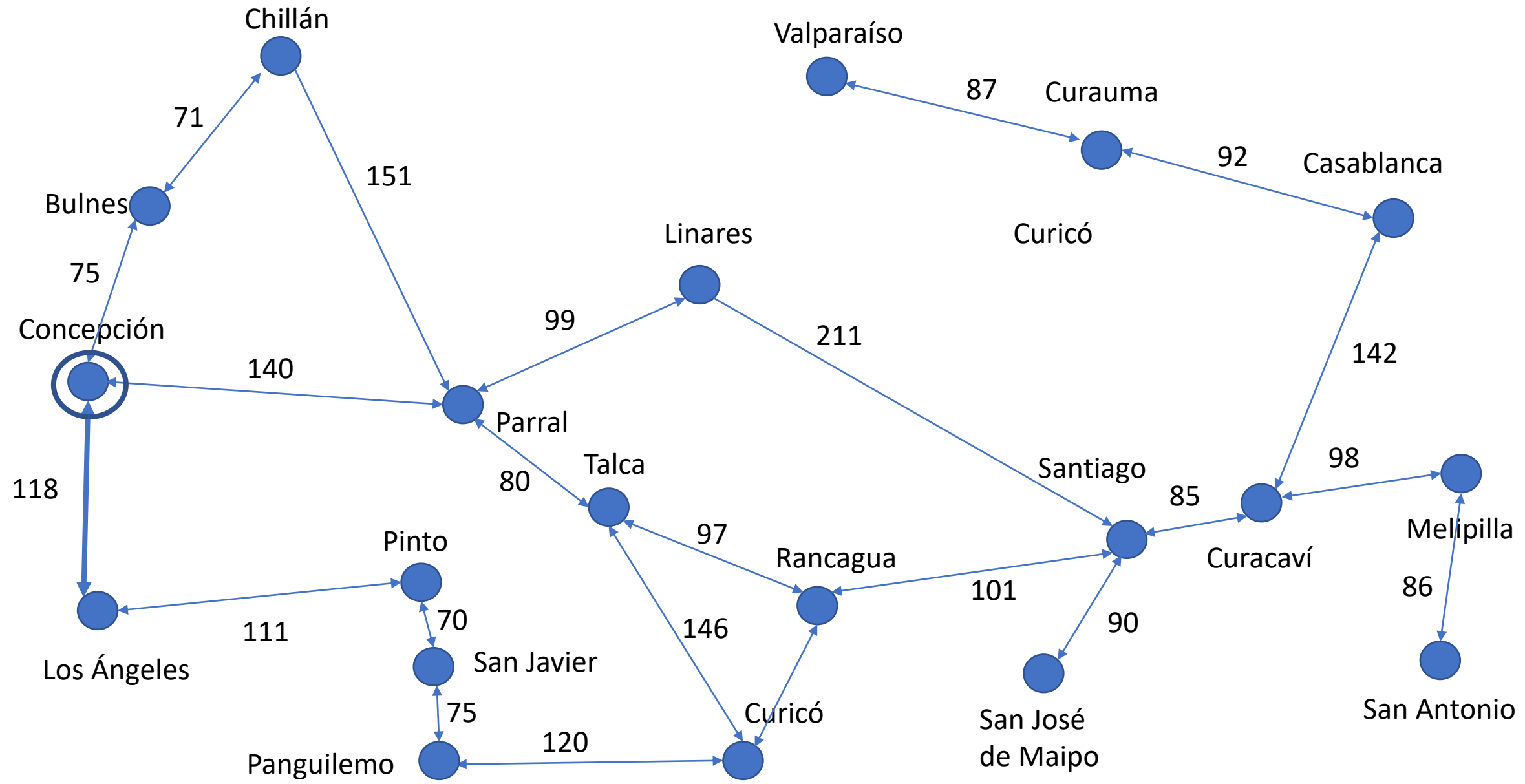
- Expande el primer nodo generado, de forma iterativa

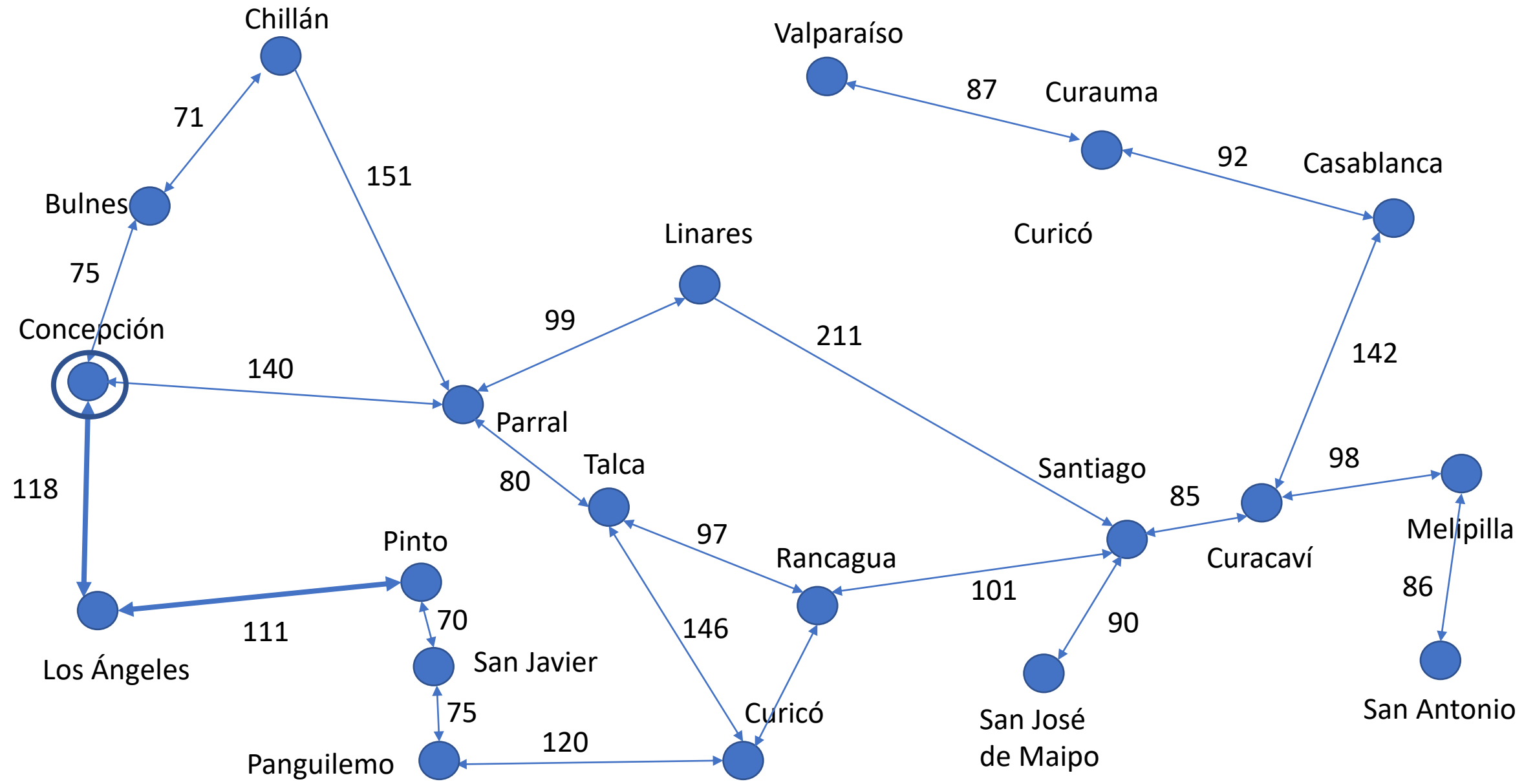


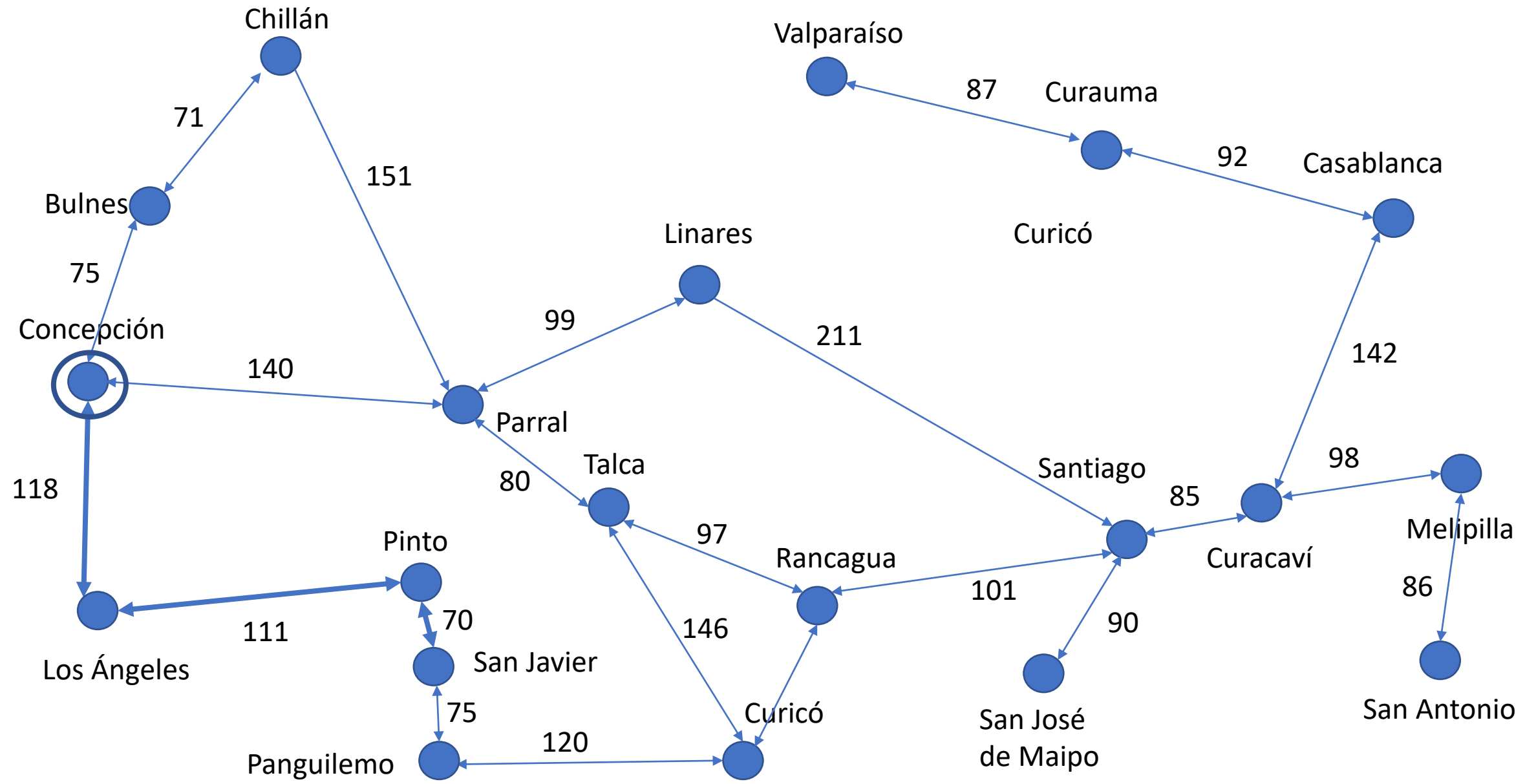


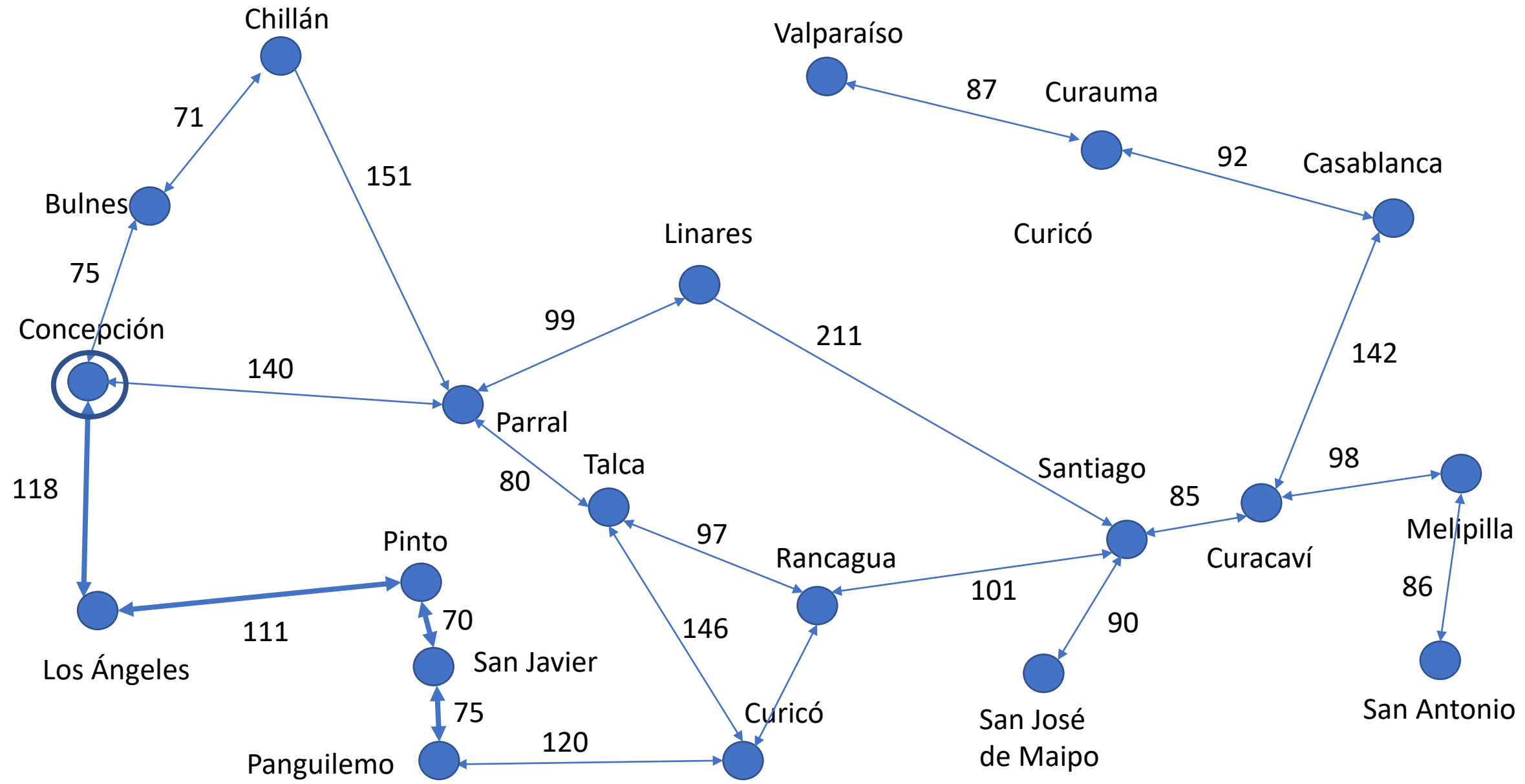


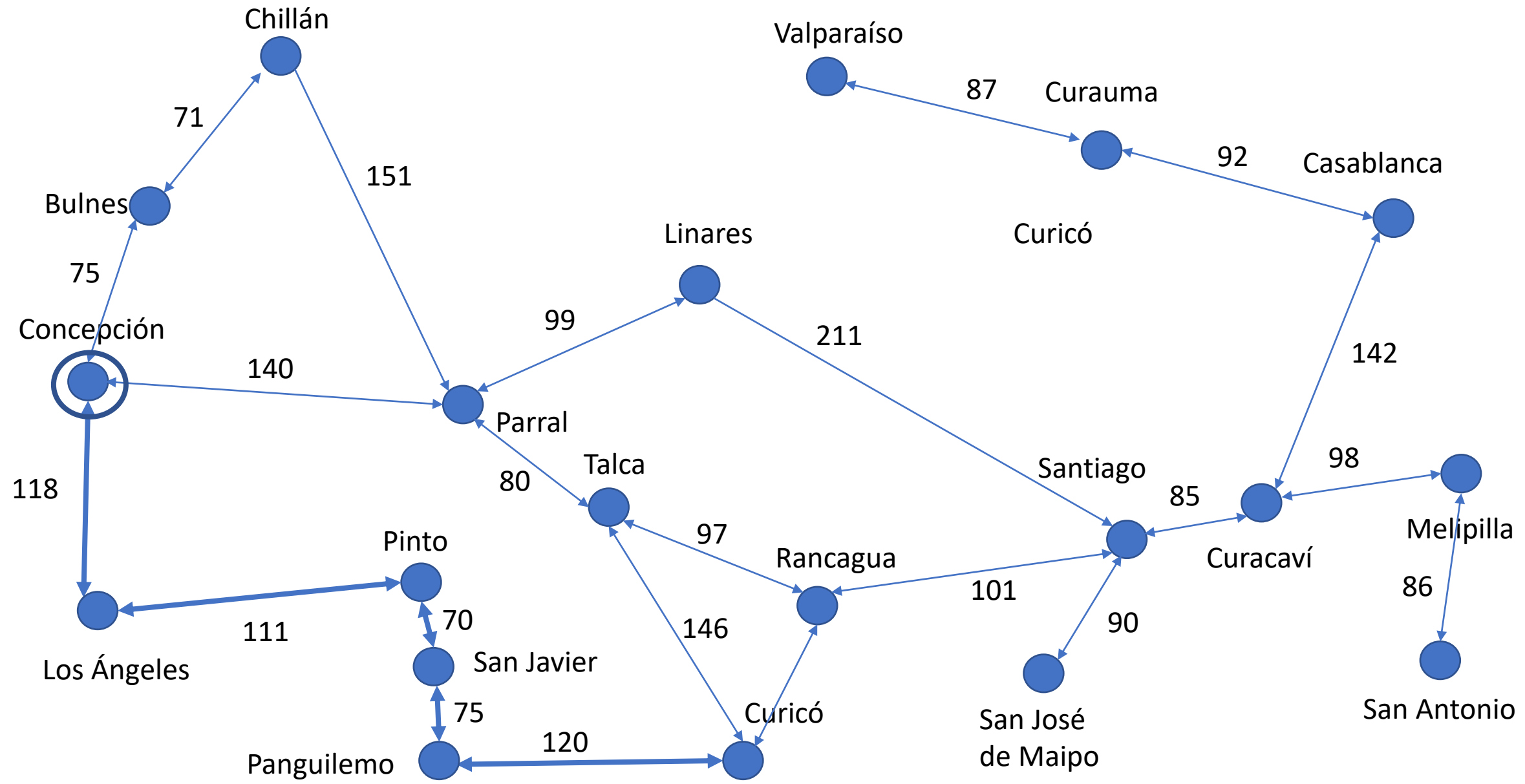


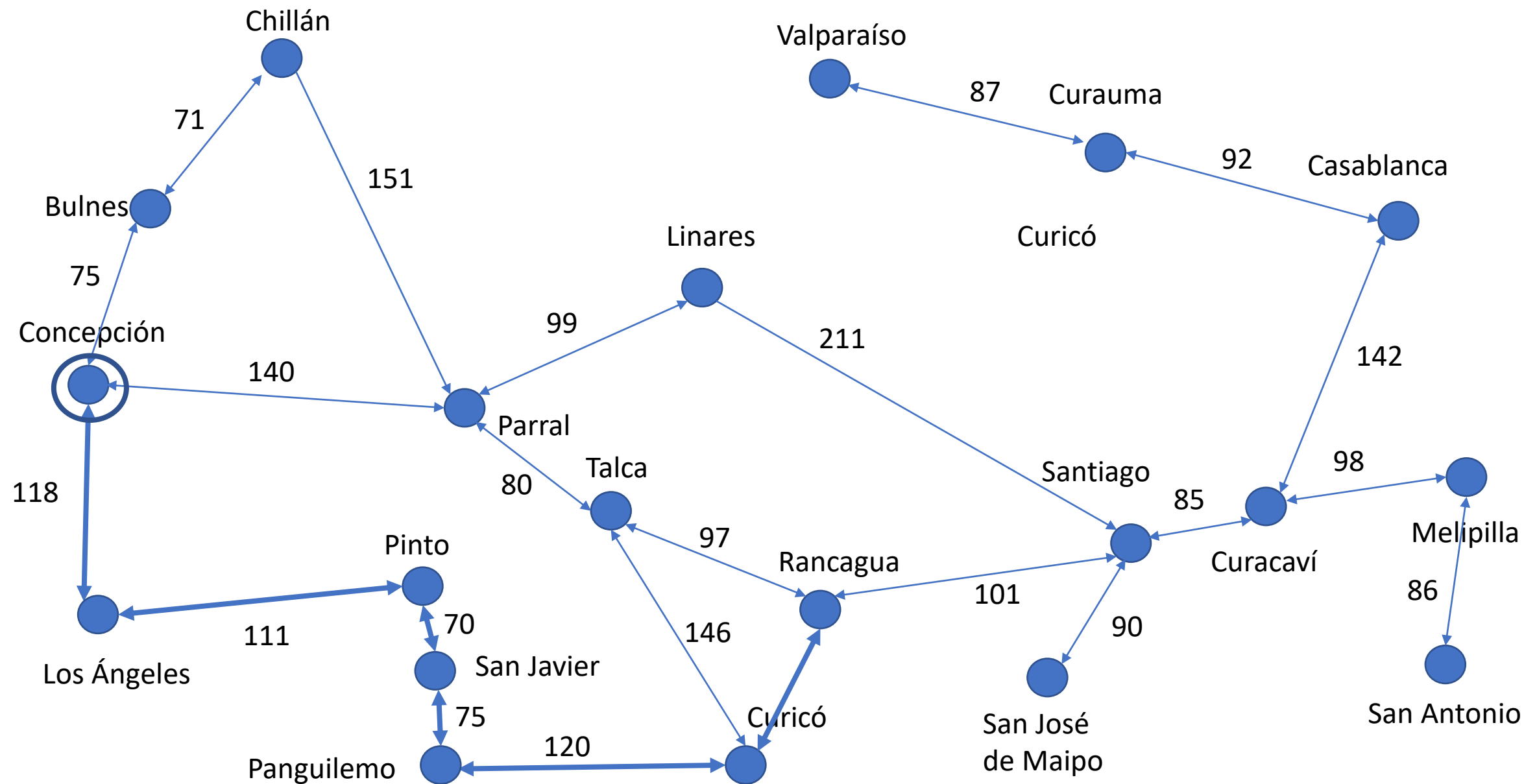


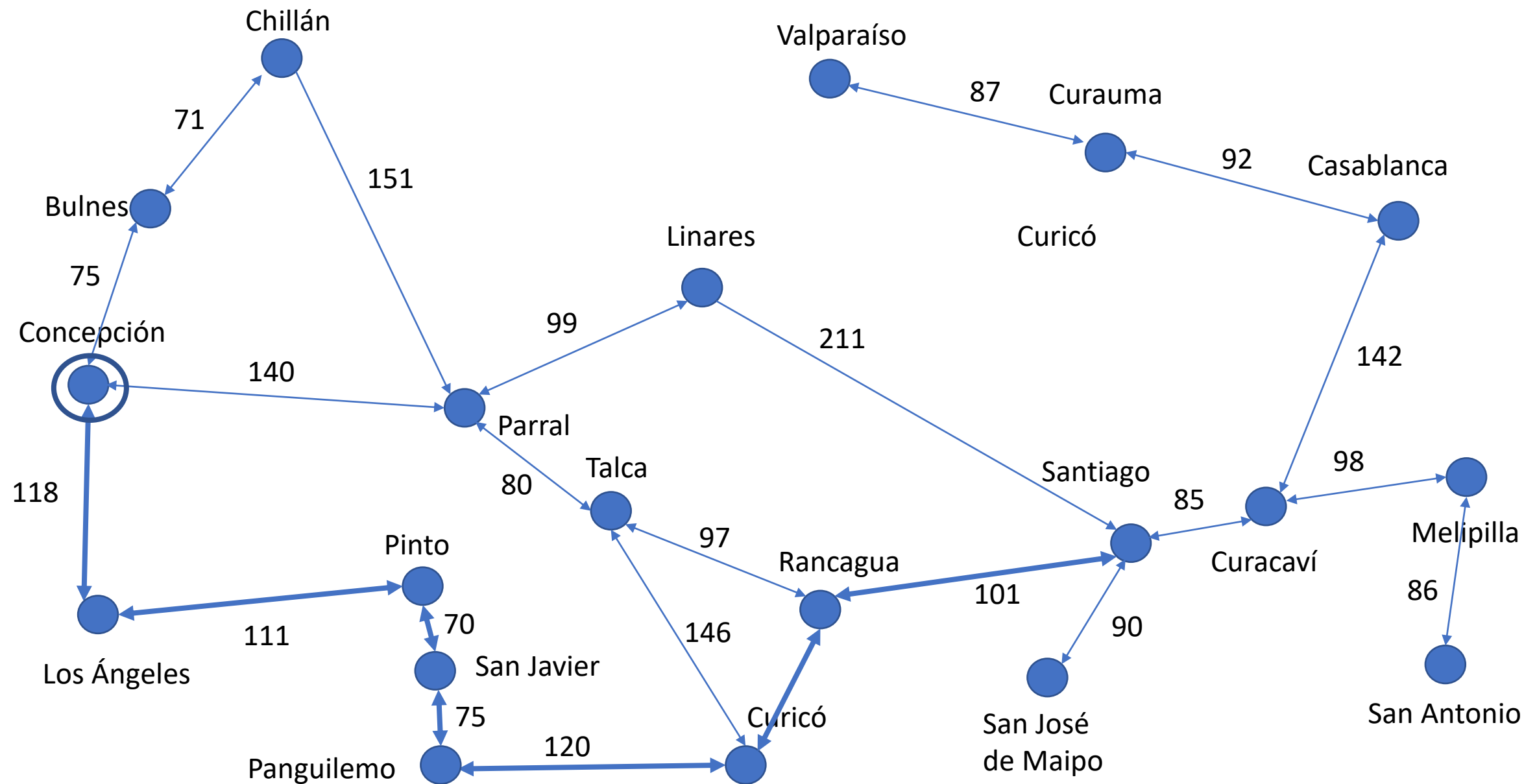




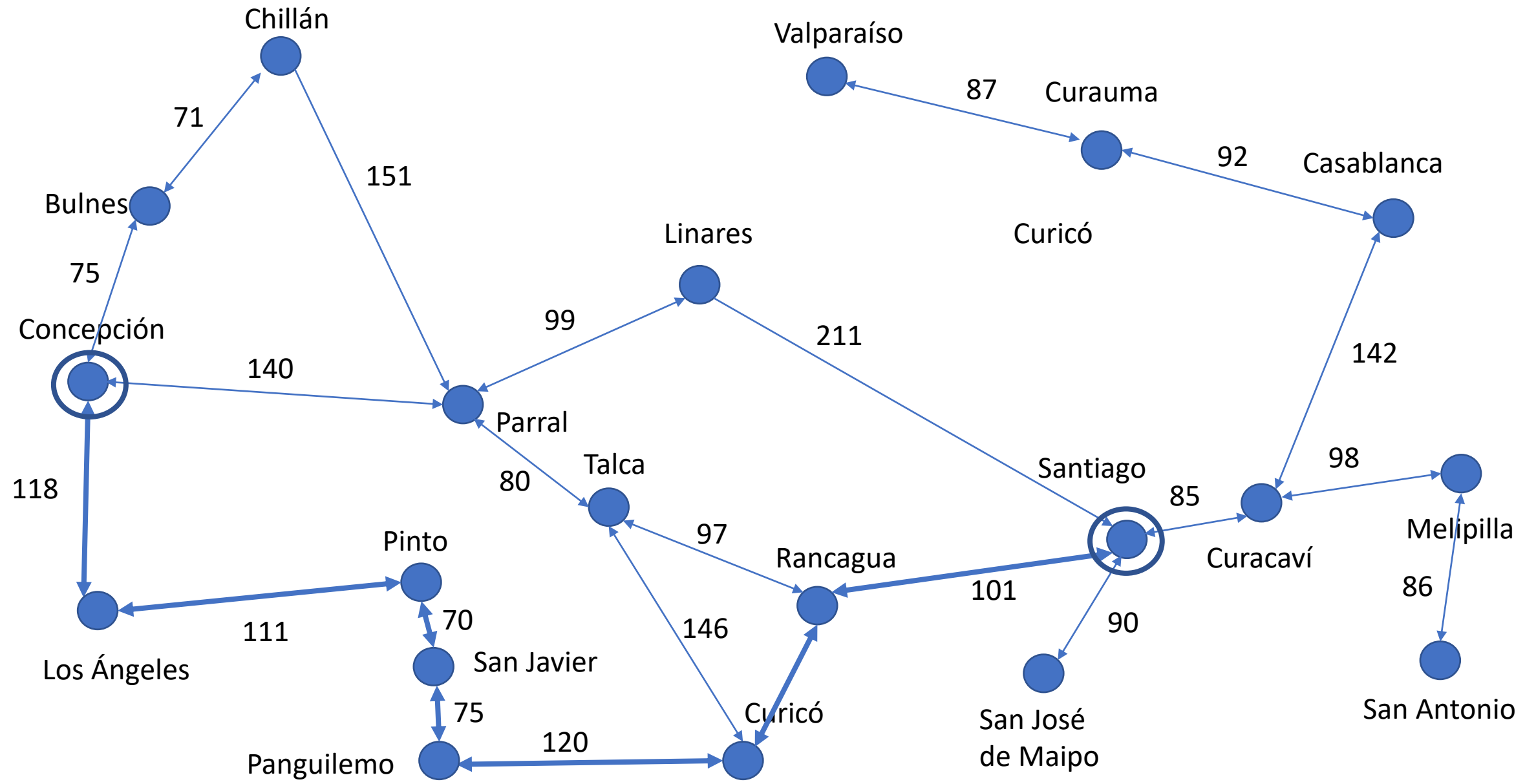








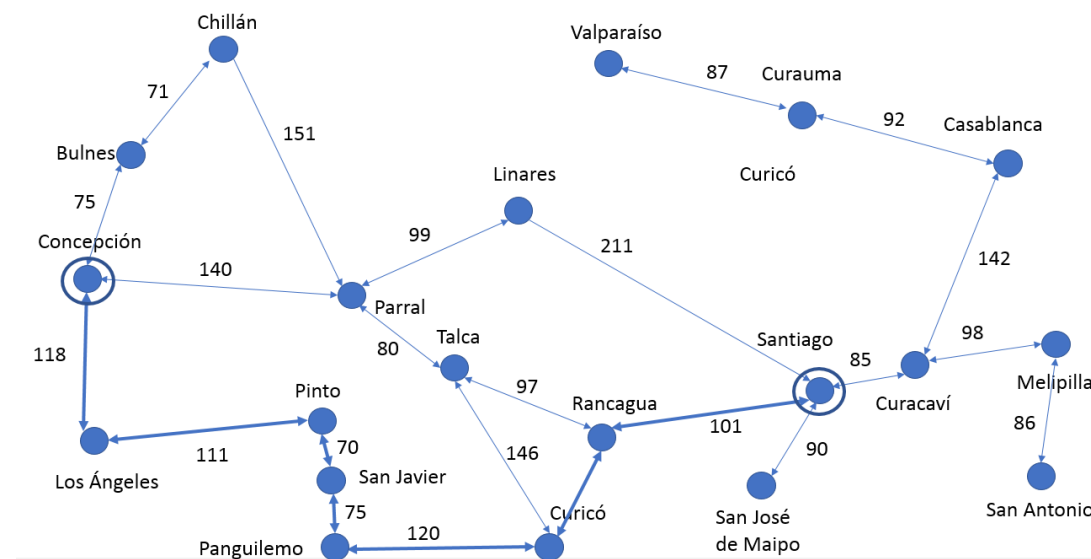


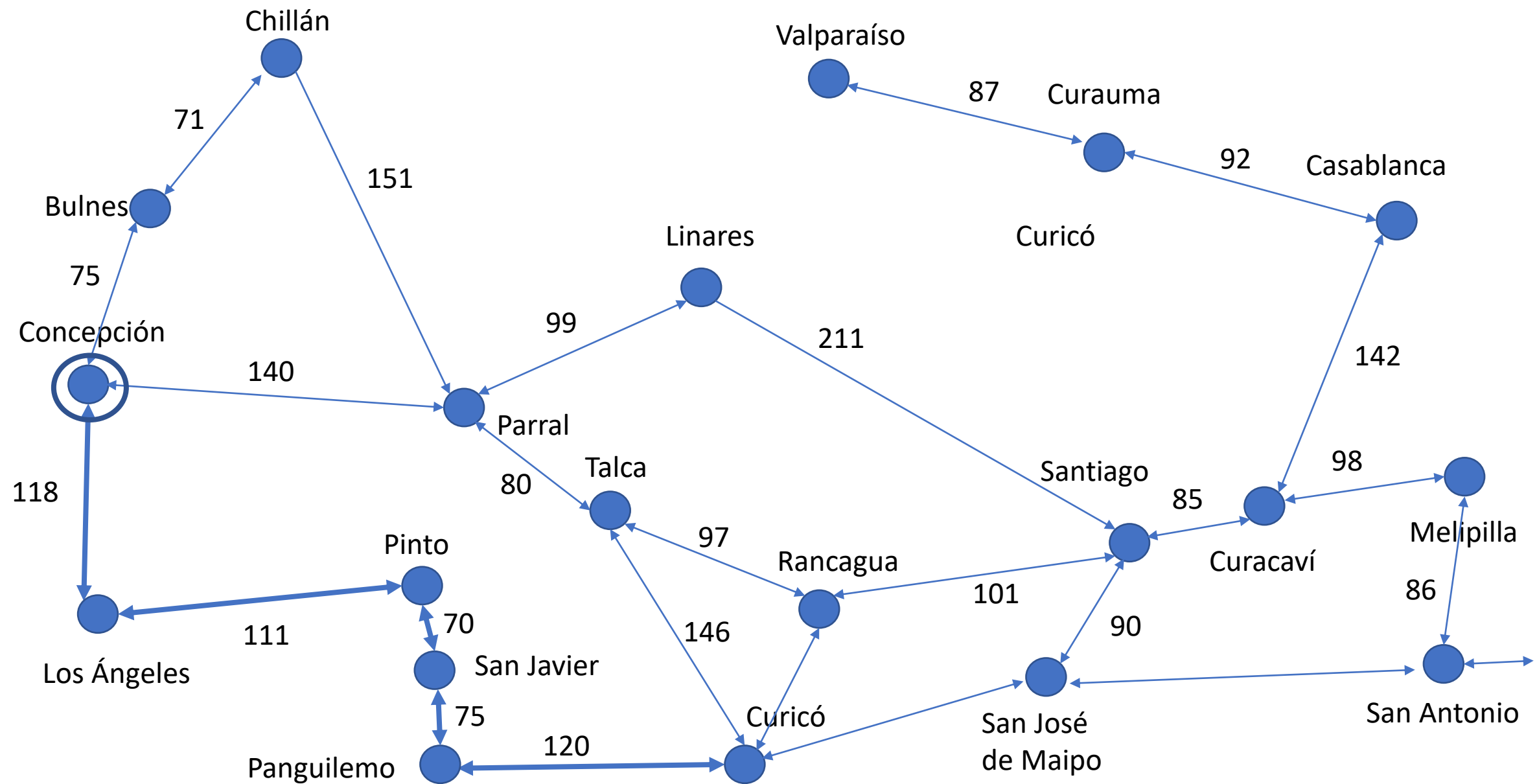


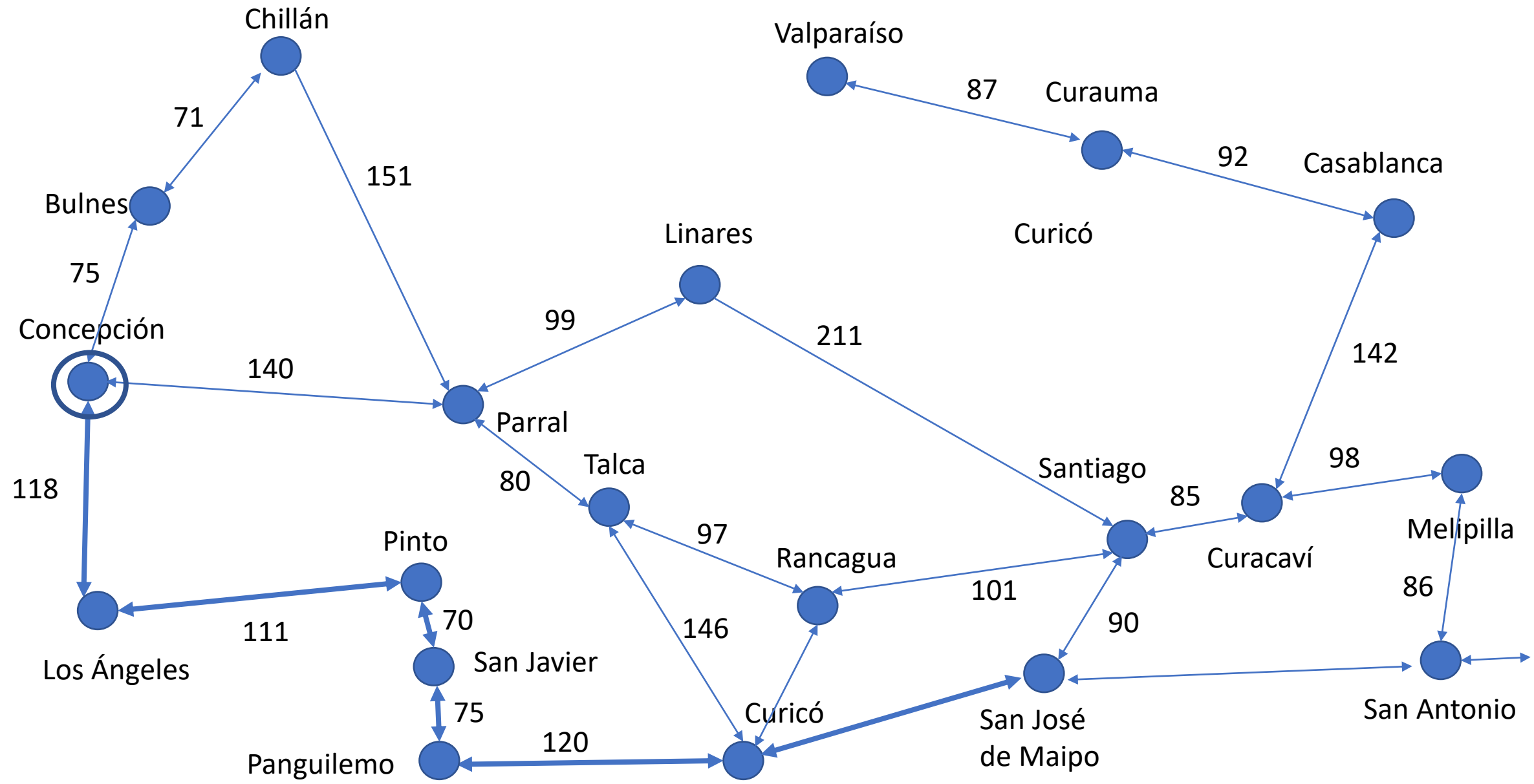
# BÚSQUEDA EN PROFUNDIDAD

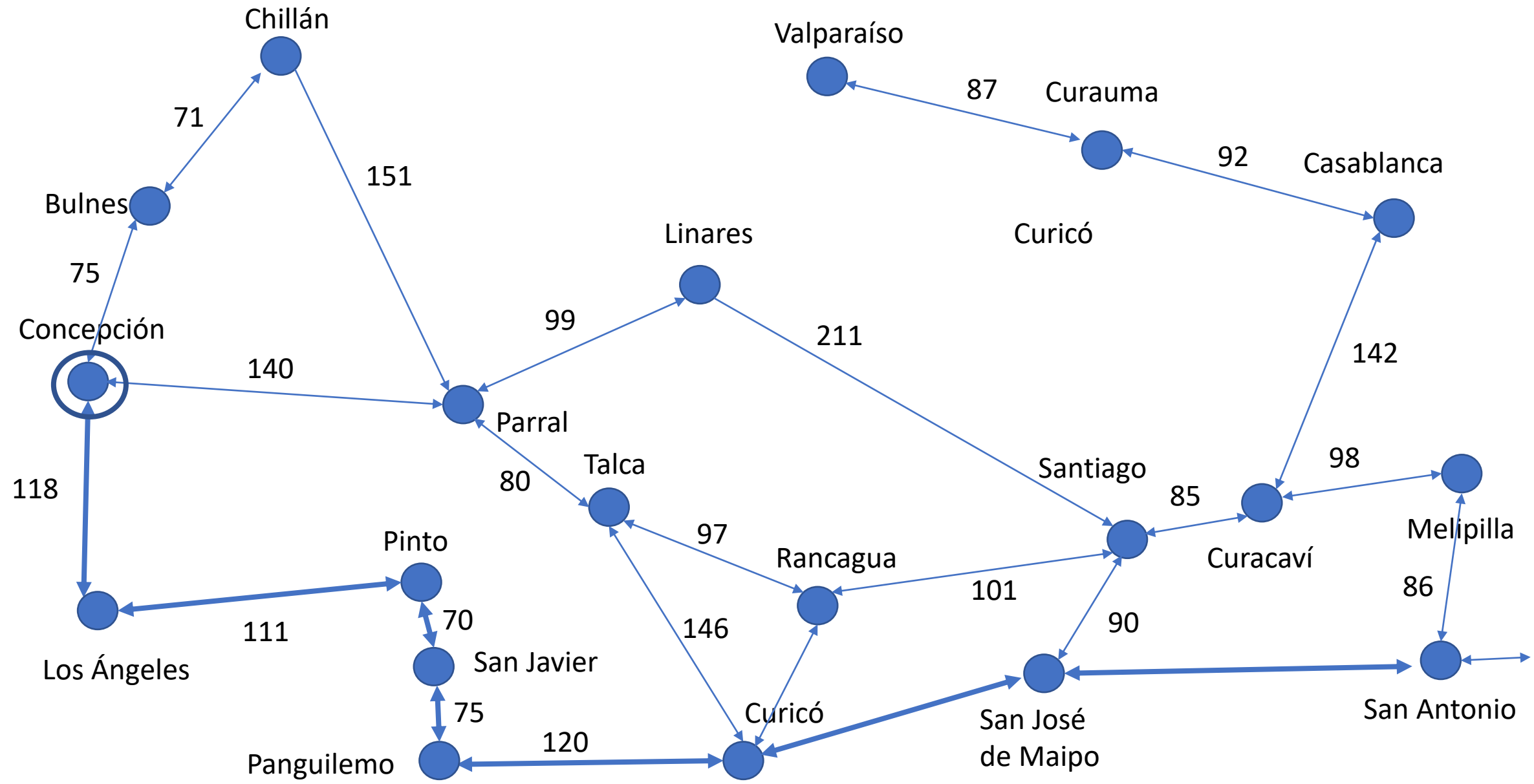
- Solución:

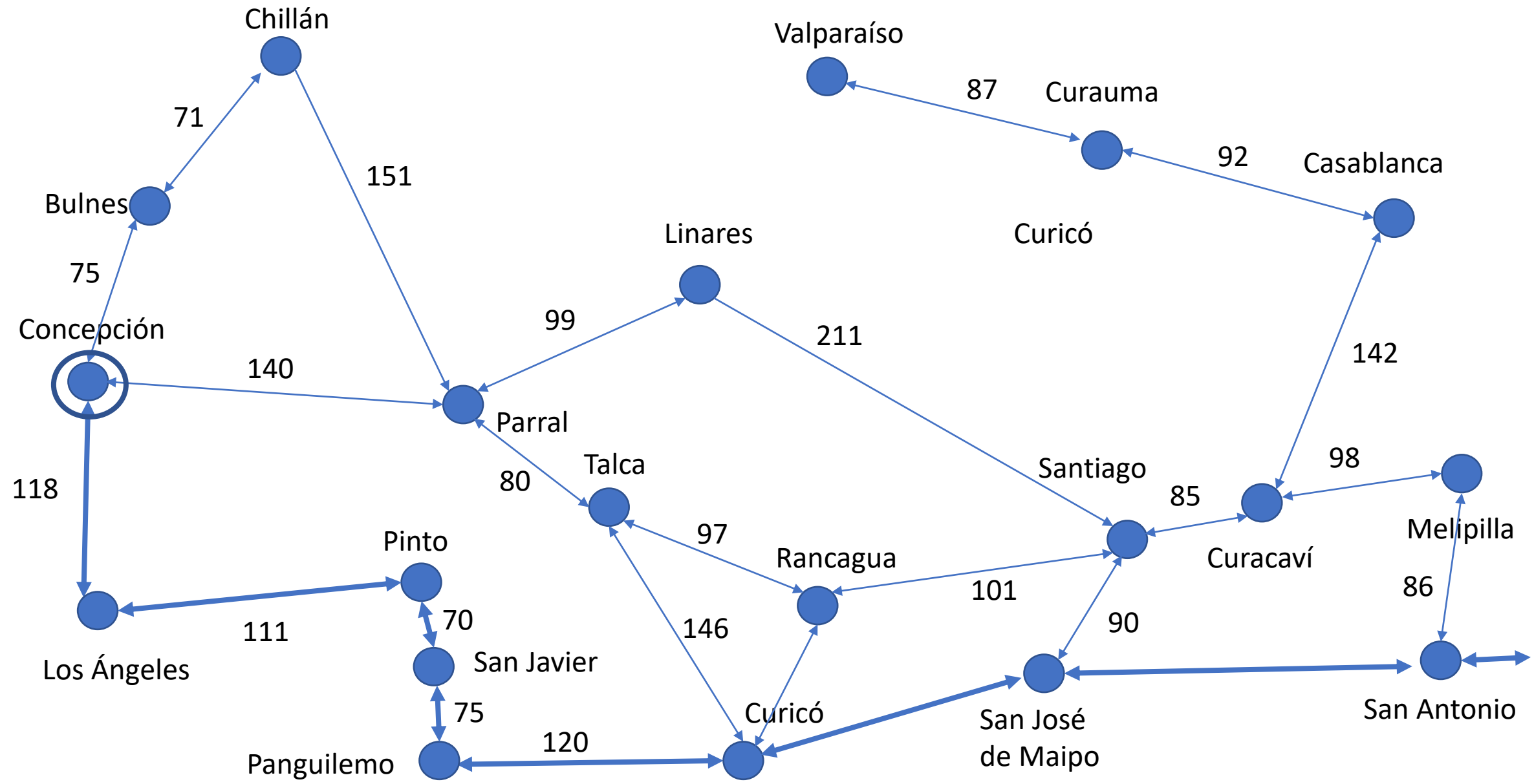
Concepción – Los Ángeles – Pinto – San Javier – Panguilemo – Curicó – Rancagua - Santiago











# BÚSQUEDA EN PROFUNDIDAD- PROPIEDADES

- ¿Encuentra una solución?
  - No siempre
- ¿Encuentra la mejor solución?
  - No necesariamente
- Costo computacional
  - Bajo 😊

# BÚSQUEDA EN PROFUNDIDAD- VARIACIONES

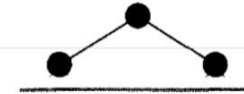
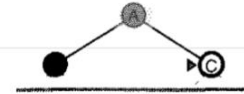
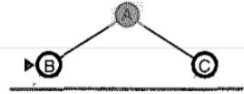
- Búsqueda en profundidad **limitada**
  - Similar a búsqueda en profundidad, pero con un límite
  - Útil cuando hay un conocimiento del problema para establecer el límite de la profundidad
- Búsqueda en profundidad **iterativa**:
  - Similar a búsqueda en profundidad limitada, pero con un límite que se incrementa gradualmente
  - Útil cuando no hay una noción predefinida del límite correcto



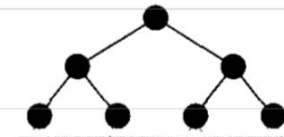
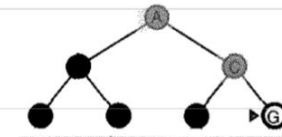
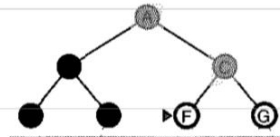
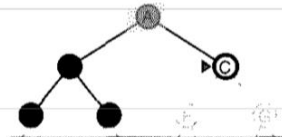
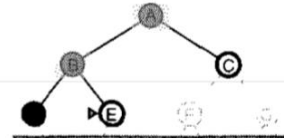
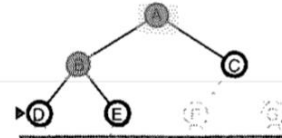
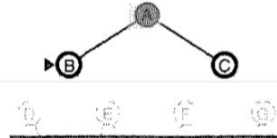
Limit = 0



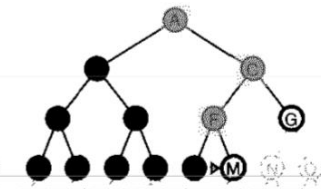
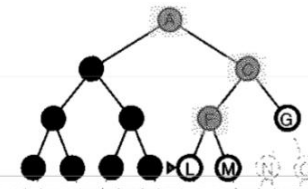
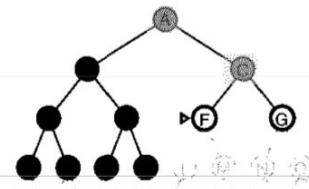
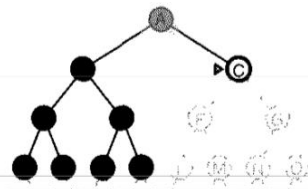
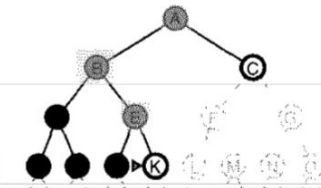
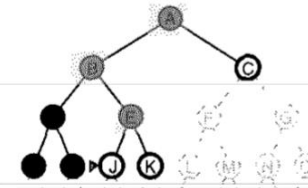
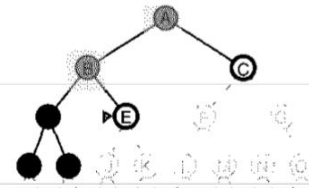
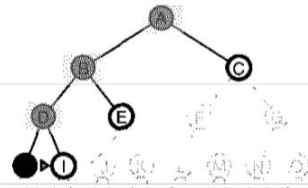
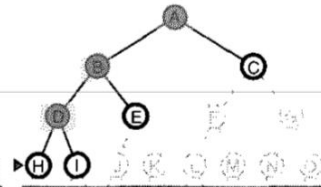
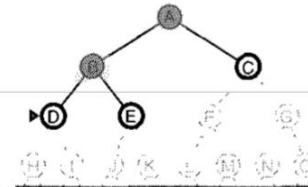
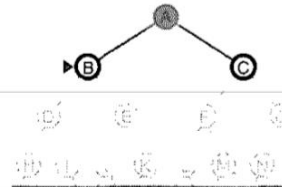
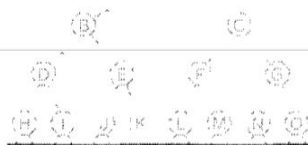
Limit = 1



Limit = 2

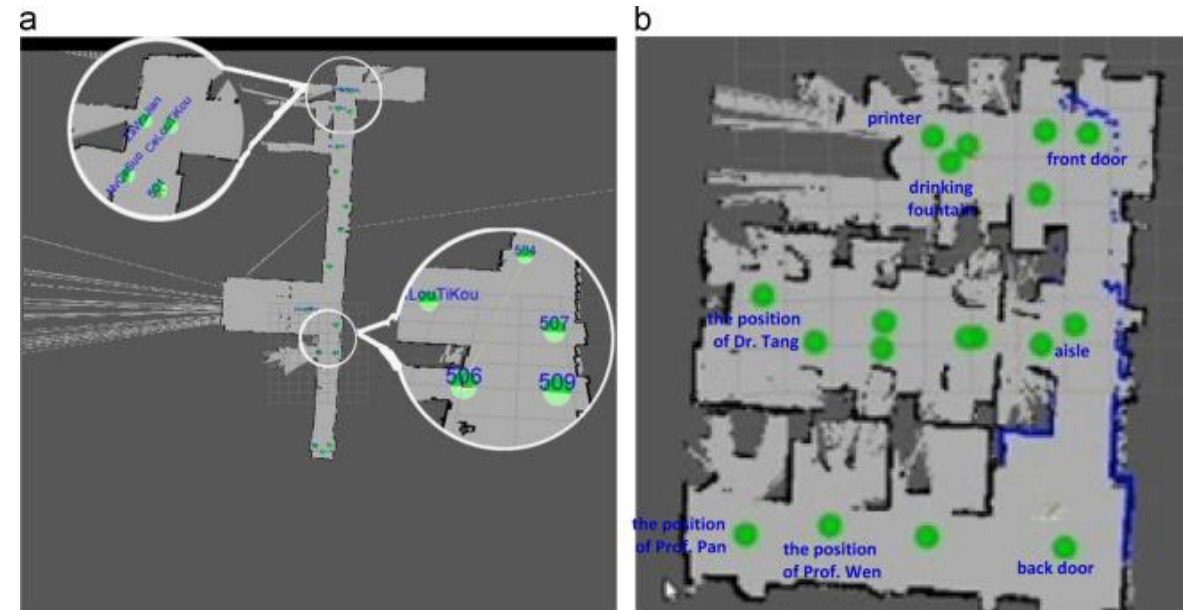


Limit = 3

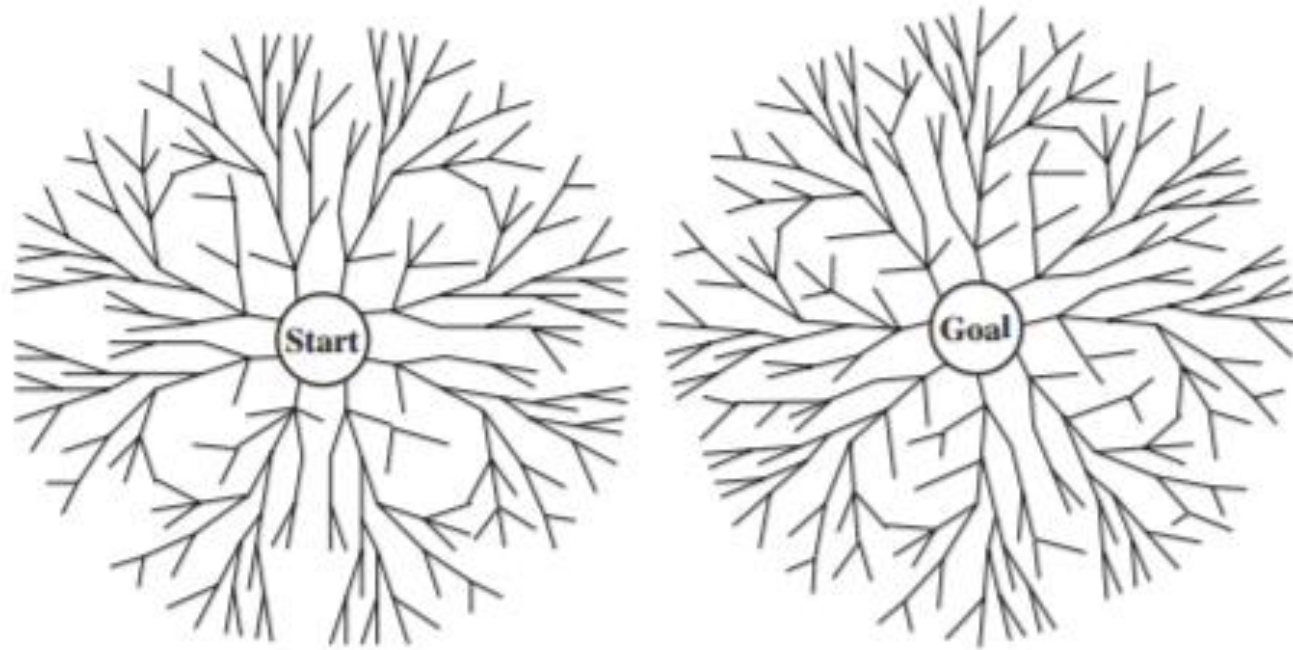


# BÚSQUEDA EN PROFUNDIDAD- APLICACIONES

- Calendarización de tareas
- Navegación de robots en ambientes desconocidos
- Mapeo de ambientes desconocidos



# BÚSQUEDA BIDIRECCIONAL



# Comparación de Estrategias no Informadas

	Anchura	Profundidad	Profundidad limitada
Tiempo	$b^d$	$b^m$	$b^l$
Espacio	$b^d$	$bm$	$bl$
¿Es óptima?	Sí	No	No
¿Es completa?	Sí	No	Sí, cuando $l \geq d$

$b$ : factor de ramificación

$d$ : profundidad de la solución

$m$ : profundidad máxima del árbol

$l$ : límite de profundidad