



**Universidad Nacional
Autónoma de México**

Facultad de Ciencias



Licenciatura en Ciencias de la Computación

Computación Distribuida

Práctica 3
(Reporte)

Equipo:

Elizalde Maza Jesús Eduardo 321031686

Peredo López Citlalli Abigail 321161022

Fernando Michel Tavera
Luis Mario Escobar Rosales
Brenda Ayala Flores
David Ortega Medina

4 de octubre del 2025

Objetivo

El principal objetivo de la práctica fue implementar algoritmos de búsqueda distribuida usando **SimPy**. El desarrollo se mantuvo dentro de las restricciones del curso, evitando dependencias externas y utilizando únicamente los módulos proporcionados y SimPy.

Desarrollo

Se implementó un **BFS descentralizado** para generar árboles de propagación de información y un **DFS** con orden determinista según las consideraciones dadas. En ambos algoritmos se asegura que los nodos mantengan correctamente sus atributos: padre, hijos y, en el caso de BFS, distancia; para DFS, los vecinos. La correcta implementación se verificó mediante las pruebas unitarias propuestas en la práctica.

Implementaciones

BFS

Cada nodo cuenta con los atributos: padre, hijos, distancia y mensajes_esperados. El nodo raíz envía el mensaje "IR" a sus vecinos. Los nodos que reciben dicho mensaje actualizan su padre y distancia si es la primera vez que se visitan o si encuentran un camino más corto. Una vez que todos los hijos responden, envían un mensaje "VOLVER" a su padre. Para manejar los casos en que los nodos reciben mensajes de retorno, se usan los valores "SI" y "NO" y se actualiza la lista de hijos en consecuencia.

DFS

Cada nodo tiene los atributos: padre, hijos, visitado, finalizado y completed_children. El nodo raíz inicia el recorrido marcándose como visitado y enviando un mensaje "GO" al vecino con menor ID. Los nodos no visitados se marcan como visitados, registran a su padre y continúan la propagación del mensaje "GO". Cuando un nodo no tiene más vecinos disponibles, envía un mensaje "BACK" a su padre.

Durante la implementación surgió el problema de coordinar correctamente el envío de mensajes "BACK", ya que en algunos casos un nodo regresaba a su padre antes de que todos sus hijos hubieran finalizado, lo que causaba que el árbol no se construyera de forma correcta. La solución fue asegurarse de que cada nodo sólo enviará el

"BACK" cuando todos sus hijos hubieran completado su exploración, usando la lista de `completed_children` para llevar el control.

Conclusión

La práctica permitió comprender mejor la implementación de algoritmos de búsqueda en un entorno distribuido, así como las diferencias entre **BFS** y **DFS** en la propagación de información y la determinación de los hijos y la distancia a la raíz.