



**Universidad Nacional  
Autónoma de México**

**Facultad de Ciencias**



**Licenciatura en Ciencias de la Computación**

**Computación Distribuida**

## **Práctica 2**

**(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

19 de Septiembre del 2025

El principal objetivo de la práctica fue modelar algoritmos distribuidos usando SimPy. Para ello se implementaron canales de comunicación tipo broadcast y distintos tipos de nodos, de modo que cada tipo de nodo ejecuta el protocolo correspondiente sobre la topología prevista por las pruebas. El desarrollo se mantuvo dentro de las restricciones del curso, evitando dependencias externas y usando únicamente los módulos proporcionados y SimPy.

El algoritmo de construcción de árbol generador se implementó en la clase `NodoGenerador`. El nodo distinguido inicia enviando un mensaje de inicio. Cada nodo que recibe por primera vez ese mensaje adopta al remitente como padre, propaga el mensaje a sus vecinos salvo al padre y responde con mensajes de retorno para que el padre registre a sus hijos. El protocolo asegura que cada nodo tenga asignado un padre y que la raíz obtenga la lista de hijos correspondiente.

El algoritmo de convergecast se implementó en `NodoConvergecast`. La raíz envía un INIT para construir la dirección padre-hijo. Las hojas comienzan el convergecast enviando BACK con sus valores. Cada nodo acumula el conjunto recibido de sus hijos, lo actualiza con su propio valor y reenvía el conjunto al padre. Finalmente la raíz aplica la función agregadora sobre el conjunto global para obtener el valor final.

El ordenamiento distribuido está en `NodoSort`. El coordinador divide el arreglo con la función cuadrícula y envía segmentos a los trabajadores mediante mensajes GO. Los trabajadores ordenan su segmento y responden con mensajes de retorno. El coordinador recolecta todos los segmentos ordenados y realiza un merge k-way local sin importar librerías externas, produciendo el arreglo ordenado final en `self.arr`.

La búsqueda distribuida se implementó en `NodoBusqueda` siguiendo el patrón estrella. El coordinador incluye su porción local al dividir el arreglo, busca localmente y envía GO a los trabajadores con sus segmentos y el elemento objetivo. Los trabajadores responden FOUND o NOT\_FOUND. El coordinador

espera las respuestas de todos los vecinos y combina los resultados para fijar `self.contenido` con el booleano final.

Se prestó especial atención a la sincronización y al formato de mensajes. Todas las rutinas que se ejecutan con `env.process(...)` son generadores que usan `yield` en lecturas de canales y en retardos de procesamiento. Los mensajes siguen un formato consistente para que remitente y receptor interpreten los campos en el mismo orden. Se corrigieron fallos intermitentes haciendo que el coordinador de búsqueda revise su segmento local y espere exactamente las respuestas esperadas de los vecinos.

En conclusión, la práctica quedó implementada y ajustada para pasar las pruebas automáticas tras corregir problemas de sincronización, mensajes y particionado de datos.