Práctica 04 - Computacion Distribuida

Integrantes:

- Sandoval Mendoza Antonio
- Sinencio Granados Dante J.

DESCRIPCION DEL CODIGO:

Dentro del archivo contiene 2 clases y 1 carpeta con dos clases más, las primeras 2 son:

1. **NODO.py:** Este contiene el constructor de los nodos y también contiene getters (que no ocupe), el método get_eventos() el cual sirve para obtener los eventos al enviar los mensajes y el método mínimo que nos ayuda a obtener al vecino menor y es usada en nuestro algoritmo bfs ,esta clase es usada por NODOBROADCAST.py y NODODFS.py

2. **NODOBROADCAST.py:** Este contiene el método de broadcast, se comprueba su funcionamiento con la clase Test.py (aunque no pasa varios test y no sabemos el porqué), este algoritmo funciona básicamente igual que el broadcast pasado pero agregando las funcionalidad para que se guarden los eventos en una lista y tener un contador reloj que va aumentando por cada envio de mensaje.

3. **NODODFS.py:** Este contiene el método de dfs, se comprueba su funcionamiento con la clase Test.py (aunque no pasa varios test por que la lista de eventos que utilizamos no la puede hashear el test.py), este algoritmo funciona básicamente igual que el dfs pasado pero agregando las funcionalidad para que se guarden los eventos en una lista y tener un contador reloj que va aumentando por cada envio de mensaje y también tiene el método máximo que nos da el reloj máximo vectorial.

Los siguientes están dentro de la carpeta canales:

4. **Canal.py:** Interfaz utilizada en CanalRecorridos.py

5. **CanalRecorridos.py:** Este consta de los metodos envia() (el que envía el mensaje a sus vecinos) y crea_canal_de_entrada() (como su nombre indica, crea el canal de entrada)

Nota: Tuvimos muchos problemas con los test y el código que decidimos enviar no los pasaba pero aun así pensando que esta correcto, si fuera posible ser flexible con la cuestión de que debe pasar los test pero si tu puedes revisar su estructura te lo agradeceríamos.