



Universidad Nacional Autónoma de México  
Facultad de Ciencias  
Práctica 4 ReadMe

Cervantes Duarte Jose Fernando 422100827

Morales Chaparro Gael Antonio 320076972

Rivera Lara Sandra Valeria 320039823

23 de octubre de 2024



## 1 Constructor de la clase `NodoConsenso`

- `V`: Vector que mantiene el conjunto de nodos descubiertos.
- `New`: Conjunto que contiene los nodos recién descubiertos en una ronda.
- `rec_from`: Vector que almacena los mensajes recibidos de otros nodos.
- `fallare` y `fallo`: Determinan si el nodo fallará y cuándo lo hará.
- `lider`: Nodo líder al final del proceso de consenso.

## 2 Algoritmo de Consenso

El método `consenso` es el núcleo del algoritmo. A continuación, se describen sus pasos principales:

1. **Determinación del fallo:** Los nodos cuyo identificador sea menor a la cantidad de fallos tendrán su atributo `fallare` como verdadero. Además cada uno selecciona aleatoriamente una ronda en la que fallará, dentro de las primeras  $f + 1$  rondas, numeradas de 0 a  $f$ .
2. **Inicio de las rondas:** Cada ronda tiene una duración equivalente a la constante `TICK`. Los nodos que no han fallado y que tienen información nueva en su conjunto `New`, envían esta información a sus vecinos a través del `canal_salida`.
3. **Recepción de mensajes:** Los nodos reciben los mensajes de sus vecinos a través del `canal_entrada`. La información recibida se guarda en `rec_from` y haremos que el conjunto `New` sea vacío, para que después los nodos descubiertos se agreguen a `V` y `New` si aún no han sido descubiertos.
4. **Elección del líder:** Una vez que han transcurrido todas las rondas (hasta  $f$ ), el nodo selecciona el líder como el primer nodo válido (no nulo) que encuentra en su vector `V`. Este nodo se considera el líder de la red.
5. **Finalización:** El algoritmo termina una vez que el líder ha sido seleccionado.

Este algoritmo de consenso permite a los nodos de una red distribuida ponerse de acuerdo sobre un líder, incluso cuando algunos de los nodos pueden fallar de manera no predecible. Utiliza un enfoque simple basado en la transmisión de mensajes y un tiempo de espera para simular el fallo de nodos. El algoritmo es robusto y garantiza la elección de un líder mientras haya un número suficiente de nodos no fallidos en la red.