

Práctica 4 – Consenso

Computación Distribuida

Profesor: Mauricio Riva Palacio Orozco

Ayudantes: Adrián Felipe Fernández Romero y Alan Alexis Martínez López

Ayudantías Distribuida — Facultad de Ciencias, UNAM

Facultad de Ciencias, UNAM
Semestre 2025-1

1. Descripción de la práctica

En esta práctica se implementará el algoritmo de consenso (sin terminación temprana). El objetivo es comprender cómo los procesos en un sistema distribuido pueden alcanzar un acuerdo común incluso ante fallos controlados.

2. Desarrollo

La práctica está conformada por tres archivos principales (con sus interfaces) y un archivo de prueba:

- `NodoConsenso.py`
- `CanalRecorridos.py` (se reutiliza el canal de la práctica anterior)
- `Test.py`

Deberás implementar la clase `NodoConsenso`, encargada de ejecutar el algoritmo y coordinar los mensajes entre procesos.

3. Prueba (`Test.py`)

El archivo de prueba (`Test.py`) incluye verificaciones automáticas para validar el comportamiento de las implementaciones. Estas pruebas permiten confirmar que los nodos logran consenso bajo las condiciones establecidas.

Nota: que las pruebas unitarias pasen con éxito no garantiza una calificación máxima; tómenlas como apoyo para el diseño.

3.1 Prerrequisitos

Para ejecutar las pruebas, corre el siguiente comando en la terminal:

```
pytest -q Test.py
```

Asegúrate de tener instalado `pytest` y las dependencias necesarias antes de ejecutar la práctica.

4. Observaciones

- Respetar los constructores proporcionados en el código base.
- La función consenso recibirá como parámetro el número de fallos f .
- Convención: $V = \{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$. Si $f = 4$, los nodos que fallarán serán $\{v_0, v_1, v_2, v_3\}$.
- Los fallos se simulan haciendo que los nodos dejen de enviar y procesar mensajes a partir de cierta ronda.
- Si un proceso falla en la ronda r , permanecerá fallado en todas las rondas siguientes.
- El algoritmo es sensible a la ronda actual; en SimPy se puede obtener con `env.now`. Si no usas SimPy, puedes simular este comportamiento limitando el número fijo de rondas.

Atributos importantes de un nodo

- V : arreglo de identificadores (IDs) de cada nodo.
- New : conjunto de mensajes de cada nodo (usa IDs, no referencias).
- $lider$: ID del nodo líder determinado.
- $fallare$: booleano que indica si el nodo fallará en algún punto del algoritmo.

5. Lineamientos de entrega

El esqueleto de la práctica ya está proporcionado; se espera únicamente que completes las secciones faltantes. Incluye un archivo ReadMe con:

- Número de práctica (Práctica 4).
- Nombres completos y números de cuenta de los integrantes.
- Breve explicación de la implementación realizada.

Solo uno de los integrantes debe subir la práctica; el otro debe marcarla como entregada antes de la fecha límite.

El archivo entregado debe ser una carpeta comprimida con el nombre: **Practica4.zip**

Para cualquier duda o aclaración, comuníquense con el ayudante a la brevedad.

6. Conclusión

Con esta práctica se refuerzan los conceptos de acuerdo distribuido y manejo de fallos en sistemas distribuidos. El algoritmo de consenso es base para la coordinación y confiabilidad en redes de procesos.