

# Computación Distribuida 2023-2

## - Practica 5 Relojes-

Profesor: Fernando Michel Tavera

Ayudantes: Yael Antonio Calzada Martín y Mauricio Riva Palacio Orozco

Fecha de entrega: Miércoles 3 de mayo 2023

### 1. Descripción Practica

Esta práctica servirá para que comprendan el concepto de reloj en un sistema distribuido asíncrono. Implementarán dos relojes (Lamport y vectorial) encima de algoritmos anteriormente vistos.

### 2. Desarrollo

Para esta práctica convertiremos nuestro sistema síncrono en parcialmente asíncronos. Esto lo lograremos esperando tiempos aleatorios entre cada envío y recepción de mensajes (observen que como estos valores están acotados, nuestro sistema no será completamente asíncrono).

Para implementar el reloj Lamport usaremos como base el algoritmo de BroadCast. Tendrán que usar el mismo archivo que entregaron para esa práctica, pero adaptarlo para que el sistema sea asíncrono y que cada nodo tenga una variable reloj.

Para el vectorial usarán el algoritmo DFS como base. Harán lo mismo para el de Lamport, pero en vez de que su reloj sea un entero, será un arreglo de estos (Nota: Tendrán que modificar su constructor para decirle al nodo cuantos nodos hay en la gráfica).

En ambos algoritmos, aparte de eso, tendrán que guardar, en una lista, todos los eventos que hayan ocurrido con ese nodo, envío y recepción de mensajes). Cada elemento de esta lista de eventos deberá tener el valor del reloj que tenía el nodo al enviar (o recibir) el mensaje, el tipo de evento (Usen la cadena E cuando se trate de envío, R cuando se trate de recibir) el mensaje, el nodo que envió y el nodo que recibió (en ese orden)

Tendrás que implementar la clase del NodoConsenso para la ejecución de este algoritmo.

### 3. Prerrequisitos

Para esta práctica no se les dará un esqueleto, pues usarán los algoritmos ya hechos. Solo se les dará el archivo `test.py` y lo tendrán que poner en la carpeta donde están los archivos de sus nodos. Posteriormente ejecutar las pruebas basta con ejecutar el siguiente comando en la terminal:

```
pytest -q test.py
```

Estas pruebas son distintas, ya que se revisa que los eventos pasaron en la ejecución respeten el orden causal.

### 4. Observaciones

- Recuerden que lo importante de los relojes es verificar que éste le asigna valores temporales correctos a cada evento.
- Para los números aleatorios pueden usar la biblioteca *random* de python.

### 5. Lineamientos de entrega

Dado que reutilizaran código ya hecho, se asume que esta bien documentado, sin embargo, deben documentar las cosas que se hagan en esta práctica para los relojes,, además dentro de la práctica se deberá agregar un ReadMe con el número de la practica, en este caso Practica 5, nombre de los integrantes con su número de cuenta y una explicación de su implementación para cada algoritmo.

Solo un integrante debe subir la practica, el otro integrante con que la marque como entregada antes de la fecha límite es suficiente, además, se deberá subir la carpeta comprimida como un .zip con el nombre "Practica5"

Cualquier duda escríbanme a la brevedad.