



# Universidad Nacional Autónoma de México

## Facultad de Ciencias

Computación Distribuida

Práctica 05

### Sincronización y relojes

#### Profesores:

Fernando Michel Tavera  
Luis Mario Escobar Rosales  
Brenda Ayala Flores  
David Ortega Medina

Fecha de entrega

Jueves , 06 de noviembre del 2025.

## Lineamientos

Los alumnos deberán hacer y entregar las prácticas siguiendo todos los puntos mencionados, si alguno llegase a ser incumplido la penalización puede variar desde puntos menos a su calificación hasta la nulificación total de éstas.

1. Las prácticas deben ser realizadas en parejas, no se calificarán prácticas individuales, así como prácticas de equipos con más de 2 miembros.
2. Las prácticas se deberán entregar a través de classroom, en un archivo .zip. Solo un miembro del equipo debe entregar la tarea con los archivos, pero ambos deben marcar como entregado.
3. El nombre del archivo .zip, debe empezar con el número de la práctica, seguido de los nombres de los integrantes.

Practica1\_FernandoMichel\_DavidOrtega.zip

4. Se deberán usar únicamente las bibliotecas permitidas para dicha práctica. Queda prohibido el uso de cualquier biblioteca externa en el código.
5. Se deberá realizar un reporte en un archivo pdf que debe llevar lo siguiente :
  - Una descripción general de como se desarrolló la práctica.
  - Se debe hacer un análisis detallado en como se implementaron los algoritmos solicitados. Mencionando todas las consideraciones, etc.
  - Cualquier otro comentario o aclaración que consideren pertinente.

Este pdf debe ir al mismo nivel que la carpeta principal de sus programas.

6. Se debe agregar un README.txt que contenga :

- Número de la práctica
- Nombre y número de cuenta de los integrantes

Debe ir al mismo nivel (en la jerarquía de carpetas que el reporte).

7. Si ni el README ni el reporte llevan los nombres de los integrantes, habrá una penalización de la calificación con un punto menos.
8. Queda estrictamente prohibido el uso de cualquier código generado por Inteligencia Artificial, así como código copiado de Internet y copias entre equipos. De haber sospecha por alguna de estas situaciones, los integrantes del equipo deberán tener una entrevista con el ayudante de laboratorio. En esta entrevista se les cuestionará aspectos de su implementaciones, algoritmos y código en general. En caso de incumplirse esta norma, la calificación de dicha práctica será automáticamente 0.
9. Si se tiene alguna complicación con la fecha y horario de entrega de la práctica, se debe avisar al ayudante para buscar una solución.
10. Si se entrega código que no compile y/o muera nada más ejecutarlo, la calificación será 0.

## Computación distribuida - Relojes

**Nota :** En su implementación, no deberán importar bibliotecas o usar otros archivos a menos que la práctica o el profesor lo indique. Para esta práctica bastará con usar `Simpy`, `time` y `pytest`.

### 1. Introducción

Para esta práctica convertiremos el sistema síncrono que hemos estado construyendo en uno parcialmente asíncrono. Esto lo lograremos esperando tiempos aleatorios entre cada envío y recepción de mensajes (Como estos valores estarán 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 deberán 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. Para esto tendrán que modificar su constructor para decirle al nodo cuantos nodos hay en la gráfica o bien hacer que cada nodo (antes de ejecutar el algoritmo principal) conozca la gráfica de comunicación.

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).

Por tanto deberán que re-implementar y ajustar `NodoBroadcast` y `NodoDFS`.

### 2. Uso Para el uso de las pruebas sigue los siguientes pasos:

- Localiza en la misma carpeta que tus códigos fuentes el archivo `test.py`
- Ejecuta el siguiente comando:

```
myUser:$ pytest -q test.py
```