

Relojes de Lamport Procesos

GOMEZ EDGAR, Universidad Central de Venezuela, Venezuela

Los relojes lógicos de Lamport constituyen un mecanismo fundamental para ordenar eventos en sistemas distribuidos asíncronos sin depender de una sincronización física precisa. Este trabajo presenta el algoritmo, basado en contadores que se incrementan localmente y se ajustan al intercambiar mensajes, y lo ilustra mediante un ejercicio práctico con tres procesos. La solución detallada demuestra cómo se asigna un orden causal consistente a los eventos, validando la utilidad del esquema como base para la coordinación en entornos distribuidos.

CCS Concepts: • **Organización de sistemas informáticos** → **Sistemas Distribuidos**; *Relojes de Lamport*.

Additional Key Words and Phrases: Sistemas Distribuidos, Asincronía

ACM Reference Format:

GOMEZ EDGAR. 2026. Relojes de Lamport Procesos. *Proc. ACM Hum.-Comput. Interact.* 1, 1, Article 1 (January 2026), 3 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1. Introduction

Los relojes o tiempos lógicos de Lamport, propuestos por Leslie Lamport en 1978, son un mecanismo utilizado en sistemas distribuidos para ordenar eventos causalmente sin requerir sincronización física exacta. Utilizan contadores que se incrementan en cada evento y se actualizan al recibir mensajes, estableciendo una relación de sucesos.

El algoritmo sigue las siguientes reglas:

- Un proceso incrementa su contador antes de cada evento que ocurra en ese proceso.
- Cuando un proceso envía un mensaje, este incluye su contador en el envío.
- Al recibir un mensaje, se actualiza el contador del receptor si es necesario, al mayor entre su propio contador y la marca de tiempo recibida en dicho mensaje.

En pseudocódigo para proceso interno

```
reloj = reloj + 1;  
marca_temporal_mensaje = reloj;
```

En pseudocódigo el algoritmo para enviar un mensaje es:

```
reloj = reloj + 1;  
marca_temporal_mensaje = reloj;  
enviar(mensaje, marca_temporal_mensaje);
```

Algoritmo para la recepción del mensaje:

```
recibir() = (mensaje, marca_temporal_mensaje);  
reloj = max(marca_temporal_mensaje, reloj) + 1;
```

Author's Contact Information: GOMEZ EDGAR, Universidad Central de Venezuela, Caracas, Venezuela.

2026. ACM 2573-0142/2026/1-ART1
<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

Proc. ACM Hum.-Comput. Interact., Vol. 1, No. 1, Article 1. Publication date: January 2026.

2. Ejercicio de Funcionamiento de relojes de Lamport en procesos

En un sistema distribuido, existen tres procesos: **P1**, **P2** y **P3**. Estos procesos se comunican mediante el envío y recepción de mensajes. A continuación se muestra una secuencia parcial de eventos (etiquetados con letras) que ocurren en cada proceso:

- **P1**: $a \rightarrow b \rightarrow c \rightarrow d$
- **P2**: $e \rightarrow f \rightarrow g$
- **P3**: $h \rightarrow i \rightarrow j \rightarrow k$

Se sabe que se producen los siguientes eventos de comunicación entre procesos:

- El evento **b** envía un mensaje que es recibido por el evento **f**.
- El evento **h** envía un mensaje que es recibido por el evento **c**.
- El evento **g** envía un mensaje que es recibido por el evento **i**.
- El evento **d** envía un mensaje que es recibido por el evento **k**.

Ejercicio:

1. Asigna marcas de tiempo usando el algoritmo de **relojes lógicos de Lamport**, suponiendo que:
 - Cada proceso incrementa su reloj en 1 unidad entre eventos locales.
 - El reloj del evento receptor debe ajustarse según el algoritmo de Lamport.
2. Muestra la tabla con los eventos y sus marcas de tiempo para cada proceso.

2.1. Solución

- Cada proceso comienza con reloj = 0.
- Para cada evento local: $\text{reloj} = \text{reloj} + 1$.
- Al enviar un mensaje: se envía la marca de tiempo actual.
- Al recibir un mensaje: $\text{reloj} = \max(\text{reloj local, marca del mensaje}) + 1$.

Proceso P1 (reloj inicial = 0)

Evento	Tipo	Cálculo	Marca
a	local	$0 + 1 = 1$	1
b	envía a f	$1 + 1 = 2$ (envía 2)	2
c	recibe de h (marca 1)	$\max(2, 1) + 1 = 3$	3
d	envía a k	$3 + 1 = 4$ (envía 4)	4

Proceso P2 (reloj inicial = 0)

Evento	Tipo	Cálculo	Marca
e	local	$0 + 1 = 1$	1
f	recibe de b (marca 2)	$\max(1, 2) + 1 = 3$	3
g	envía a i	$3 + 1 = 4$ (envía 4)	4

Evento	Tipo	Cálculo	Marca
h	envía a c	$0 + 1 = 1$ (envía 1)	1
i	recibe de g (marca 4)	$\text{máx}(1, 4) + 1 = 5$	5
j	local	$5 + 1 = 6$	6
k	recibe de d (marca 4)	$\text{máx}(6, 4) + 1 = 7$	7

Proceso P3 (reloj inicial = 0)

Tabla final consolidada

Proceso	Evento	Marca Lamport
P1	a	1
P1	b	2
P1	c	3
P1	d	4
P2	e	1
P2	f	3
P2	g	4
P3	h	1
P3	i	5
P3	j	6
P3	k	7

3. Conclusión

La implementación del algoritmo de Lamport, ejemplificada en el ejercicio con tres procesos, confirma su eficacia para establecer un orden causal lógico robusto en sistemas distribuidos. El mecanismo, que combina incrementos locales y la función de maximización al recibir mensajes, garantiza la coherencia del tiempo lógico ante las asincronías de la comunicación. Los resultados obtenidos validan que este enfoque proporciona una base sólida y sencilla para rastrear relaciones de precedencia, siendo esencial para técnicas avanzadas de sincronización y coordinación en este tipo de entornos.

Referencias

- [1] Wikipedia. *Tiempos lógicos de Lamport*. Recuperado de https://es.wikipedia.org/wiki/Tiempos_l%C3%B3gicos_de_Lamport. Último acceso: 25 de enero de 2026.