

# Programación de Servicios y Procesos

## Ejercicio opcional 13:

Imagina que trabajas en el desarrollo de una aplicación de mensajería en la que un **Servicio de Notificaciones** genera mensajes para ser enviados a usuarios, y un **Servicio de Entrega** es el encargado de enviar esos mensajes. Los mensajes se generan a diferentes intervalos de tiempo y también se envían con ciertos retrasos.

El objetivo es implementar un sistema en Python que simule este escenario utilizando hilos y semáforos para la sincronización entre el generador de mensajes (productor) y el servicio de entrega (consumidor). Además, usa la clase `Timer` de `threading` para añadir retrasos en las operaciones.

## Requisitos:

1. **Servicio de Notificaciones (Productor):** Genera un total de 10 mensajes para ser enviados, cada uno con un contenido aleatorio y un identificador único. Usa `Timer` para simular el tiempo de generación de cada mensaje (entre 1 y 3 segundos de retraso).
2. **Servicio de Entrega (Consumidor):** Procesa los mensajes y los "envía" (imprime en la consola que el mensaje ha sido enviado). También simula un retraso en el envío de entre 2 y 4 segundos.
3. Usa semáforos para sincronizar el acceso a una cola de mensajes compartida.
4. Mide el tiempo total que tarda el sistema en procesar y enviar los mensajes. Para ello puedes utilizar la librería `time`, puedes almacenar el tiempo de inicio, de final y restar para obtener la duración.