



## Proyecto Final

Facultad de Ingeniería

Sistemas Operativos

### **Integrantes:**

Sebastián Angulo Vergara  
David Estevan Rodriguez Jurado  
Juan David Ordoñez Jiménez  
Juan Martin Trejo Vanegas

## Introducción

Este proyecto implementa un sistema de noticias basado en el modelo Publicador/Suscriptor. En este modelo, los procesos que generan noticias, llamados publicadores, están separados de los que las reciben, los suscriptores. Los publicadores envían noticias de diversas categorías (arte, política, ciencia, farándula y sucesos) al Sistema de Comunicación (SC), que actúa como intermediario, distribuyendo cada noticia a los suscriptores interesados en esa categoría.

El propósito del proyecto es simular un sistema de distribución de noticias en tiempo real, donde múltiples publicadores pueden enviar información y múltiples suscriptores pueden recibir solo las noticias de las categorías de su interés. Este sistema de comunicación permite a los suscriptores y publicadores conectarse y desconectarse sin interrumpir el flujo de información, asegurando flexibilidad y escalabilidad en la gestión de noticias.

## Descripción del Sistema

- **Publicador:** Envía noticias al sistema de comunicación. Cada noticia incluye una categoría específica y un texto, cumpliendo con un formato predefinido.
- **Suscriptor:** Recibe las noticias enviadas por el sistema de comunicación de acuerdo con los temas seleccionados. Cada suscriptor se conecta al sistema, indica sus categorías de interés y espera recibir noticias relevantes.
- **Sistema de Comunicación (SC):** Gestiona la entrega de noticias, asegurando que cada noticia enviada por los publicadores llegue solo a los suscriptores interesados. Este proceso central también gestiona la terminación del sistema, informando a los suscriptores cuando no habrá más noticias.

Esta arquitectura modular permite probar y analizar el sistema con diferentes combinaciones de publicadores y suscriptores, proporcionando una estructura robusta y eficiente para la gestión de información en tiempo real.

## Desarrollo

La implementación del sistema se divide en tres módulos principales: Publicador, Suscriptor y Sistema de Comunicación (SC). Cada módulo cumple un rol específico en el sistema de noticias, utilizando el patrón de arquitectura Publicador/Suscriptor. En este

patrón, el SC actúa como intermediario, distribuyendo noticias de acuerdo con las preferencias de suscripción de los usuarios.

**Publicador:** Este proceso lee noticias de un archivo y las envía al SC a través de un pipe compartido. Cada noticia incluye una categoría (Arte, Política, Ciencia, Farándula o Sucesos) que identifica su tipo. El publicador envía las noticias al SC respetando un intervalo de tiempo, permitiendo que múltiples publicadores publiquen en simultáneo sin sobresaturar el sistema.

**Suscriptor:** Este proceso se conecta al SC, elige las categorías de noticias que desea recibir y espera las noticias de esos temas. Cada suscriptor utiliza un pipe compartido con el SC para recibir solo las noticias de las categorías seleccionadas. Los suscriptores pueden unirse y desconectarse en cualquier momento sin interrumpir el sistema.

**Sistema de Comunicación (SC):** Este proceso central gestiona la recepción de noticias de los publicadores y las distribuye a los suscriptores según sus preferencias. También asegura que el sistema finalice adecuadamente, enviando un mensaje de cierre a todos los suscriptores cuando ya no hay noticias pendientes.

### **Diagrama de secuencia**

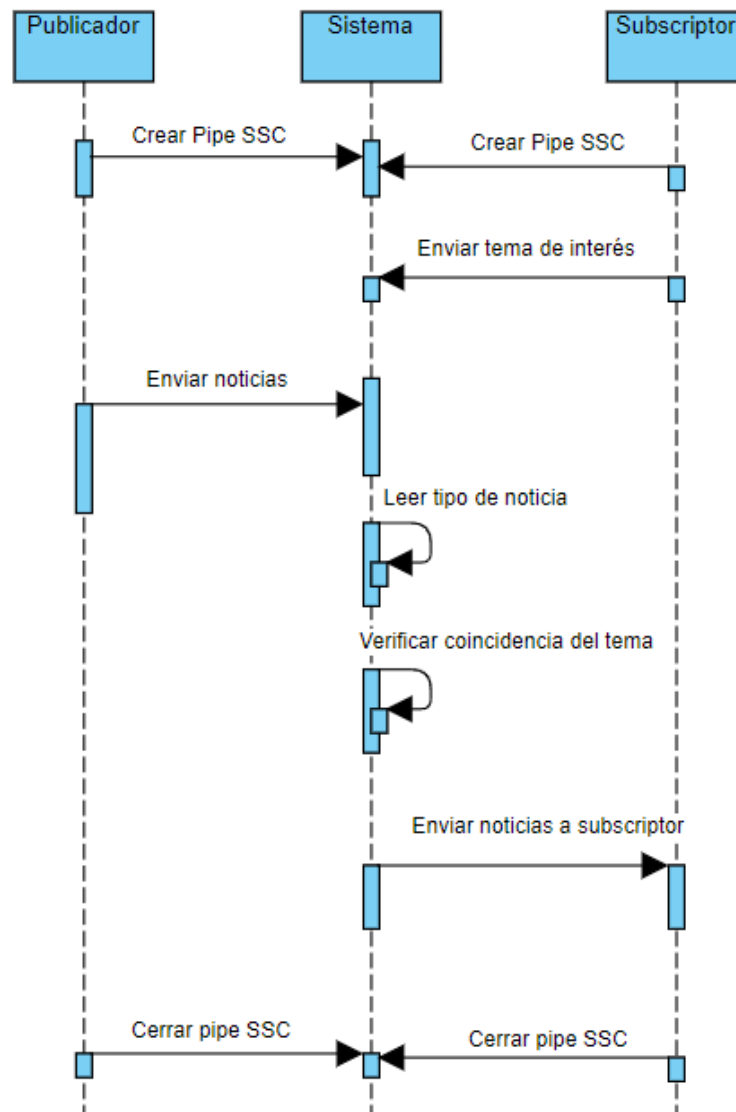


Figura 1. Diagrama de secuencia.

### Funcionalidades Principales

**Comunicación mediante Pipes:** Se utilizan pipes como medio de comunicación entre los procesos. Los publicadores escriben en un pipe compartido con el SC, mientras que los suscriptores leen de otro pipe compartido con el SC para recibir sus noticias.

**Filtrado de Noticias:** El SC compara la categoría de cada noticia con las suscripciones de los usuarios y solo envía la noticia a los suscriptores interesados, garantizando que cada usuario reciba solo la información relevante.

## Contenido

**publicador.c:** Implementa el proceso Publicador, encargándose de leer las noticias de un archivo y enviarlas al SC.

**suscriptor.c:** Implementa el proceso Suscriptor, que se conecta al SC, se suscribe a temas específicos y recibe noticias filtradas.

**sistema.c:** Implementa el SC, gestionando la comunicación entre publicadores y suscriptores y distribuyendo las noticias según las suscripciones.

**Archivos de Noticias:** Archivos de texto que contienen las noticias en un formato específico. Cada publicador lee su archivo de noticias, permitiendo pruebas con distintos tipos y frecuencias de noticias.

**Makefile:** Un archivo de automatización para compilar todos los módulos y generar los ejecutables (publicador, suscriptor, sistema).

## Plan de Pruebas

### Caso de Prueba 1: Publicación Básica de Noticias

**Objetivo:** Verificar que un publicador pueda enviar una noticia al sistema de comunicación.

#### Pasos:

Iniciar el proceso sistema.

Iniciar un proceso publicador que envíe una noticia de prueba con formato correcto.

### Caso de Prueba 2: Suscripción a Temas y Recepción de Noticias

**Objetivo:** Comprobar que un suscriptor recibe solo las noticias de las categorías a las que está suscrito.

#### Pasos:

Iniciar el proceso sistema.

Iniciar un proceso suscriptor y suscribirse a las categorías "

Iniciar un publicador y enviar noticias de varias categorías.

### Caso de Prueba 3: Múltiples Suscriptores con Diferentes Intereses

**Objetivo:** Verificar que el sistema maneje varios suscriptores y distribuya correctamente las noticias según sus intereses.

**Pasos:**

Iniciar el proceso sistema.

Iniciar dos procesos suscriptor:

Suscriptor 1: Suscripción a "A" (Arte) y "S" (Sucesos).

Suscriptor 2: Suscripción a "C" (Ciencia) y "P" (Política).

Iniciar un publicador y enviar noticias de las categorías "A", "S", "C" y "P".

**Caso de Prueba 4: Finalización del Sistema**

**Objetivo:** Comprobar que el SC finaliza correctamente cuando no hay más publicadores enviando noticias y transcurre el tiempo de espera (timeF).

**Pasos:**

Iniciar el proceso sistema con un timeF definido.

Iniciar un publicador y enviar una noticia.

Verificar que, una vez que el publicador termina, el SC espera timeF y luego envía un mensaje de finalización a todos los suscriptores.

**Referencias**

Equipo editorial de IONOS. (2022, November 23). *Pipes de Linux explicados*. IONOS Digital

Guide. <https://www.ionos.com/es-us/digitalguide/servidores/configuracion/pipes-linux/>

GeeksforGeeks. (2024, October 11). *Named Pipe or FIFO with example C program*.

GeeksforGeeks. <https://www.geeksforgeeks.org/named-pipe-fifo-example-c-program/>

Arsys. (n.d.). *Todo sobre la arquitectura cliente-servidor*. [https://www.arsys.es/blog/todo-sobre-](https://www.arsys.es/blog/todo-sobre-la-arquitectura-cliente-servidor)

[la-arquitectura-cliente-servidor](https://www.arsys.es/blog/todo-sobre-la-arquitectura-cliente-servidor)

