

## Sistema de Noticias (patrón Publicador/Suscriptor)

### Definiciones:

El patrón o modelo de comunicación publicador/suscriptor se compone de un conjunto de aplicaciones o procesos que envían y reciben mensajes (noticias, eventos, elementos de datos, etc.) utilizando algún sistema de comunicación, de una forma especial. En este proyecto los elementos de datos o eventos a enviar serán noticias.

De forma más específica, los programas a desarrollar deberán hacer lo siguiente:

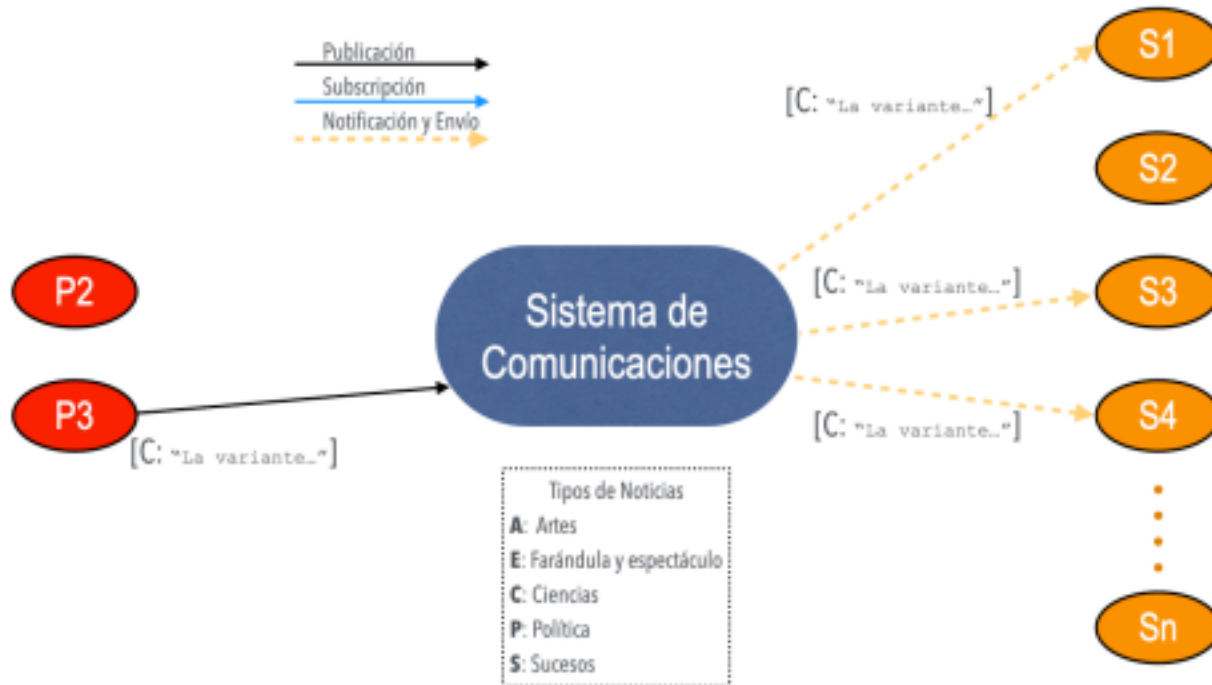
- **Los publicadores (P):** publican noticias, aunque no conocen quienes son los destinatarios de dichas noticias.
- **Los suscriptores (S):** El o los consumidores de los eventos (en este caso noticias) reciben el nombre de suscriptor(es). Mediante una subscripción, el suscriptor informa al sistema de comunicación las noticias que son de su interés. Los suscriptores pueden recibir noticias de publicadores diferentes, pero lo hacen es a través del proceso sistema de comunicación.
- **Sistema de Comunicación (SC):** La tarea del sistema de comunicación es hacer coincidir las suscripciones con los eventos publicados; notifica a los suscriptores de la llegada de eventos de su interés y asegura la entrega correcta de todas las noticias que los publicadores han publicado.

En la **Figura 1** describe el patrón *Publicador/Suscriptor*: los círculos rojos son los procesos publicadores, que pueden publicar diferentes tipos de noticias (Arte, Política, etc.). Los suscriptores, círculos de color naranja, se suscriben a un tipo de noticia específico. Por ejemplo, **S2** (suscriptor 2) solo quiere recibir noticias de **Política**, mientras que **S3** (suscriptor 3) decide recibir noticias de **Ciencia** y **Espectáculos**.



**Figura 1.** Sistema Publicador/Suscriptor a implementar en su fase de inicialización.

La **Figura 2** presenta una imagen del sistema en un instante de tiempo. En ese instante solo **P3** (publicador 3) está publicando una noticia de ciencia que el sistema de comunicación envía a **S1, S3 y S4** (suscriptores 1, 3 y 4). Se puede observar que en ese instante de tiempo, solo hay dos publicadores en el *Sistema de Comunicaciones*. El esquema de *publicador/subscriptor* es muy similar a la forma de operar de X/Twitter.



**Figura 2.** Imagen del Sistema de Comunicación en un instante determinado de su operación.

## Implementación del Proyecto

A continuación se describe la funcionalidad de cada proceso, entradas y salidas (cuando aplique) y los mecanismos de comunicación que se van a usar. Hay aspectos que no están explícitos, porque forman parte de lo que se debe diseñar e implementar los estudiantes, como parte del proyecto. **Recuerde que no se puede modificar ninguna de las llamadas de los procesos (publicador, suscriptor y sistema) que aparecen en este enunciado, sin antes contar con el aval de su profesor.**

### Publicador

Son los procesos que envían las noticias al sistema de comunicaciones. Los procesos se invocarán desde el **shell** (terminal/consola) de la siguiente forma:

**\$ publicador -p pipePSC -f file -t timeN**

**Donde:**

Los **flags**, **-p**, **-f** y **-t** indican el tipo de argumento que viene a continuación, esto permite que se cambie el orden de los argumentos, siempre que este se anteceda con el **flag** correspondiente.

**pipePSC**: es el pipe nominal por donde los publicadores envían información al Sistema de Comunicación. *Este corresponde al pipe de color **violeta** en la figura 3*

**file**: es el nombre de un archivo donde están las diferentes noticias que el publicador enviará al Sistema de Comunicaciones. Este archivo tiene un formato específico. Cada una de las líneas del archivo contendrá una noticia. La primera letra, que debe estar en mayúscula, debe representar alguno de los 5 tipos de noticia que se especifican en este enunciado y luego debe estar un símbolo de dos puntos (:) y el resto de la línea corresponderá al texto de la noticia, terminando en un punto. Las líneas podrán tener espacios adicionales en blanco siempre que respeten los detalles que se dan aquí del formato. Si alguna noticia no está en el formato adecuado se debe indicar por la pantalla al iniciar el programa publicador.

Por ejemplo, algunos ejemplos del formato de estos archivos pueden ser los siguientes:

P1	P2
A: El 25 se presenta la orquesta de cámara. P: Se aproximan las elecciones. P: Hay un nuevo candidato en Alianza Popular. E: Fue exitosa la presentación de Maluma.	A: XXXXXXXXXXXXXXXX. P: El presidente dio declaraciones. S: Candyman fue robado por un motorizado. S: YYYYYYYYYYYYYY. C: El telescopio captó un nuevo exoplaneta.

**Tabla 1.** Ejemplo de 2 archivos de publicación.

La **Tabla 1** presenta solo dos publicadores como ejemplo, en las sustentaciones se podrá crear un número mayor de publicadores. La letra indica el tipo de noticia (solo las explícitas en la **Figura 1**) y a continuación el texto.

**timeN**: es el tiempo en segundos entre el envío de una noticia y el envío de la siguiente al *Sistema de Comunicación* (esto se usa en la sustentación para que se entremezclen las publicaciones de varios publicadores)

Adicionalmente se debe tener en cuenta:

- En cualquier momento, antes de que finalice el funcionamiento del sistema completo, se puede crear un proceso publicador.
- Los publicadores terminan luego de que el archivo que contiene las noticias llega a su fin y que transcurra el tiempo **timeN**.

## Suscriptor

Puede imaginar que son procesos que se van a ejecutar en las pantallas de una cafetería. El dueño de la cafetería, por supuesto, sabe los intereses de sus clientes, por tanto, se suscribe a varios tópicos. Cada proceso suscriptor se invocará de la siguiente manera: **\$ suscriptor -s pipeSSC**

**Donde:**

**pipeSSC:** es el pipe de comunicación de los suscriptores con el Sistema de Comunicación. Es el pipe **verde** en la **Figura 3**. Ese **flag** permite enviar los temas o tópicos a los cuales se suscribe un determinado suscriptor.

Una vez invocado el proceso, se le preguntará al usuario en la pantalla por los tópicos a los que desea suscribirse. Debe suscribirse al menos a un tópico. El suscriptor envía esta información al *Sistema de Comunicaciones* y luego se queda esperando por las noticias de interés. Cada vez que llegan una o varias noticias deben aparecer por la pantalla.

En cualquier momento de la ejecución del sistema se puede crear o levantar un suscriptor. Los suscriptores terminan cuando el Sistema de Comunicación les avisa.

### Sistema de Comunicaciones (SC)

Es el proceso encargado de: aceptar las noticias de los publicadores, suscripciones de los clientes o suscriptores, hacer el *match* y enviar la información a quien corresponda. El proceso se va a invocar desde la consola de la siguiente forma:

```
$ sistema -p pipePSC -s pipeSSC -t timeF
```

**Donde:**

**pipePSC:** nombre del *pipe nominal* de comunicación entre los publicadores y el SC. Este nombre debe coincidir con el nombre dado al invocar a los publicadores, ya que se trata del mismo pipe nominal.

**pipeSSC:** nombre del *pipe nominal* de comunicación entre los suscriptores y el SC. Este nombre debe coincidir con el nombre dado al invocar a los suscriptores, ya que se trata del mismo *pipe nominal*.

**timeF:** Una vez que han terminado todos los procesos publicadores el SC esperará este **timeF** en segundos (*por si se crea otro publicador*). Una vez concluido este tiempo, envía un mensaje o notificación a los suscriptores indicando el fin de la emisión de noticias y muestra un mensaje por su pantalla, indicando que ha terminado su trabajo.

La **Figura 3** presenta los procesos *publicadores* y *suscriptores*. Los procesos envían la información relacionada con noticias a las suscripciones a través de 2 *pipes nominales* (uno para cada tipo de proceso).

Se debe diseñar el resto de las funcionalidades del Sistema de Comunicaciones: estructuras de datos para guardar la información de publicadores y suscriptores, pipes adicionales para enviar las noticias, hilos o señales (si los considera necesario). Este diseño se debe exponer en la sustentación del proyecto (también con ilustraciones, como se muestran en este enunciado).



**Figura 3.** Comunicación con el *Sistemas de Comunicaciones*

## Noticias

Solo se manejarán 5 categorías de noticias:

A: Noticia de Arte

E: Noticia de Farándula y Espectáculos

C: Noticia de Ciencia

P: Noticia de Política

S: Noticia de Sucesos

Cada noticia publicada tendrá un máximo de 80 caracteres.

## Entrega del proyecto

La entrega final será en la semana 17 (12 de noviembre para todos los grupos). La sustentación será los días 13 y 14 según el día de clase que toque a cada sección. Sin excepción, los proyectos que no se entreguen el día 12 de noviembre no serán calificados (si tienen algún problema deben hablar previamente con el o la profesora responsable del curso).

Recuerde que los profesores entregarán los casos de prueba para el día de la sustentación.

## Sustentación

### Sustentación:

1. El diseño completo del sistema:

- Estructuras de datos para guardar la información en cada uno de los tres tipos de procesos.

- Pipes adicionales a los especificados en el enunciado para enviar las noticias.
- Formato de todos los mensajes que se envían entre todos los procesos
- ¿Cómo se hace el “match” de mensajes publicados con los suscriptores, internamente en el **SC**? ¿Cómo es el diseño de este proceso para recibir la comunicación de los dos tipos de procesos? Se debe entregar un diagrama de secuencia de los que se usan en UML.
- ¿Cómo deja de funcionar todo el sistema? ¿Cómo lo implementarán, con mensajes señales, etc.?
- Implementación de los tres programas, en la fase de suscripción de sus funciones, la publicación de noticias se evaluará en la entrega.

Todo esto se explicará en una exposición de máximo 10 minutos, luego el profesor realizará las preguntas y observaciones que considere pertinentes.

Deberán implementar y mostrar para la entrega:

- Al menos un publicador enviando varios tipos de noticias.
- Al menos dos suscriptores “suscribiéndose” a un tipo de noticia
- El SC imprime lo publicado y decide qué noticia va para qué suscriptor y lo avisa en su pantalla.

Los profesores enviarán previamente las noticias que van a publicar los publicadores y cuáles serán las suscripciones realizadas por el suscriptor.

La entrega que deben realizar en la plataforma del aula virtual consiste en: códigos fuente de las funcionalidades implementadas, un archivo **makefile** para generar los procesos ejecutables, un archivo pdf con la presentación utilizada en la sustentación. Todo debe venir en un archivo en formato **tar.gz**.

## Observaciones adicionales

- El proyecto se deben realizar en grupos de como máximo 3 estudiantes.
- Los estudiantes pueden intercambiar ideas sobre el proyecto, pero al momento de codificar cada grupo debe trabajar de forma autónoma, sin intercambiar códigos con otros grupos. Cualquier sospecha de copia o fraude académico será elevada a las instancias correspondientes dentro de la Facultad de Ingeniería.
- El proyecto se debe hacer en lenguaje ANSI C, es decir que funcione en el lenguaje C disponible en **cocalc.com**, ya que allí será corregido el proyecto.

CATEGORÍA	<i>Excelente</i>	<i>Regular</i>	<i>Deficiente</i>
<u>A.</u> COMPREN DER EL FUNCIONA MIENTO DEL PROBLEMA PLANTEADO Y LA IMPLEMEN TACIÓN CUMPLE LAS ESPECIFICA CIONES PLANTEAD AS.	Se realiza la solución del problema según el enunciado, respetando requerimientos de entrada, salida y TODAS las especificaciones para la implementación.	El estudiante realiza la solución al problema siguiendo la mayoría de las especificaciones indicadas en el enunciado pero tiene algunas fallas menores.	El estudiante no sigue la mayoría de las especificaciones que se dieron en el enunciado con respecto a los argumentos de cada tipo de proceso, las operaciones de los publicadores, suscriptores y el SC, comunicación síncrona o asíncrona entre procesos, funcionamiento del proyecto
<u>B.</u> FUNCIONAMI ENTO DEL PROGRAMA	Según el funcionamiento de los casos de prueba, correcta implementación y comunicación entre los procesos.		
<u>C.</u> CÓDIGO	Se siguen todas las indicaciones del enunciado: Uso de Guía de Programación C y archivos de encabezado. Se validan llamadas al sistema. El código está bien documentado.	El código presenta deficiencias menores en no más de dos de los siguientes aspectos: documentación, estructura o validación de las llamadas al sistema.	Se presenta alguna falla importante en 2 o más de los siguientes aspectos: <ul style="list-style-type: none"> <li>• Documentación</li> <li>• Validación de las llamadas al sistema</li> <li>• Estilo de programación (Estructura)</li> </ul>

<u>D.</u> SUSTENTACIÓN	El estudiante responde correctamente a la(s) pregunta(s) formuladas. En el caso de que se solicite modificación, ésta se realiza correctamente en el tiempo establecido. El estudiante demuestra buen manejo de la interfaz de línea de comando y de las herramientas necesarias para ejecutar su proyecto.	El estudiante realiza la modificación solicitada correctamente pero toma más del tiempo establecido o la modificación tiene problemas menores. En el caso de preguntas, el estudiante responde bien a la mayoría de las preguntas formuladas. En general, se observa que el estudiante maneja la interfaz de línea de comandos del SOP, no obstante el equipo tuvo que ayudarlo en algunos momentos de la sustentación.	El estudiante no responde la o (las) preguntas formuladas o no logra el funcionamiento de la modificación solicitada, sólo muestra algunas líneas de código. El manejo de la interfaz de línea de comandos del SOP es deficiente.
CATEGORÍA RÚBRICA	<i>Excelente</i>	<i>Regular</i>	<i>Deficiente</i>
<u>A.</u>	3	[2 , 3)	[0 , 2)
<u>B.</u>	7	[3 , 7)	[0 , 3)
<u>C.</u>	4	[2 , 4)	[0 , 2)
<u>D.</u>	6	[2 , 4)	[0 , 2)
<b>Total</b>	20		