



PROYECTO - Sistema de Noticias

Santiago Botero

Francisco Guzman

Juan Camilo Alba

Jorge Fortich

Brayan Fajardo

PRESENTADO A:

Jhon Jairo Corredor

SISTEMAS OPERATIVOS

PONTIFICIA UNIVERSIDAD JAVERIANA

2024

BOGOTÁ D.C

1. Introducción

El proyecto se basa en el desarrollo de un sistema de comunicación tipo Publicador/Suscriptor. La funcionalidad principal de este sistema es permitir que ciertos procesos (publicadores) envíen noticias sin conocer los destinatarios específicos, mientras que otros procesos (suscriptores) reciben solo las noticias de interés.

2. Definiciones y Componentes del Sistema

- **Publicadores (P):**

Envían noticias al Sistema de Comunicación sin conocer a los destinatarios.

- **Suscriptores (S):**

Reciben noticias de su interés. Pueden suscribirse a temas específicos y recibir solo aquellas noticias que coincidan con sus suscripciones.

- **Sistema de Comunicación (SC):**

Intermedia entre publicadores y suscriptores, gestionando el “match” entre las suscripciones y las noticias recibidas.

3. Diseño Completo del Sistema

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

- **Publicador (P):**

No se define una estructura de datos para guardar pero, lee noticias de un archivo y las envía por un pipe al proceso central pipePSC. Utiliza un buffer para almacenar temporalmente cada noticia antes de enviarla.

- **Sistema de Comunicación (SC):**

utiliza dos estructuras, una para la configuración del sistema y otra para los suscriptores, estas fueron construidas de la siguiente manera:

```

// Estructura para la configuración del sistema
typedef struct {
    char* pipePSC;
    char* pipeSSC;
    int timeF;
} sistema_config;

// Estructura para representar a cada suscriptor
typedef struct {
    char temas[MAX_TEMAS];
    int contador_temas;
    char nombre_pipe[BUFFER];
    int fd;
} suscriptor;

```

Imagen 1. Estructura de configuración y suscriptor.

donde los suscriptores se manejan en un arreglo de suscriptores.

- **Suscriptor (S):**

No se define una estructura de datos para guardar pero, crea un pipe FIFO único para cada suscriptor, basado en el ID del proceso, para así recibir los mensajes

3.2 Pipes Adicionales para Enviar Noticias

- **pipePSC:**

Pipe nominal para enviar noticias desde los publicadores al SC.

- **pipeSSC:**

Pipe nominal para enviar suscripciones desde los suscriptores al SC.

- **Pipes Únicos:**

Para cada suscriptor, se crea un pipe único en formato /tmp/fifo_suscriptor_<pid>, utilizado por el SC para enviar noticias de temas específicos a cada suscriptor.

3.3 Formato de los mensajes

- **Para salida:**

Para la salida del programa se usa el mensaje de "salida".

- **Para suscripciones:**

Se utiliza el formato X nombre_pipe, donde X representa el tema y nombre_pipe representa el nombre del pipe único entre el sistema y el suscriptor.

- **Para noticias:**

Se utiliza el formato establecido X: noticia. donde X es el tema y se acompaña de dos puntos luego la noticia y cierra con un punto.

3.4 Lógica del “Match” de Mensajes

- **Registro de suscripciones:**

Al guardar un arreglo de suscriptores en el SC, cada suscriptor se mantiene con su propio pipe FIFO y los temas de interés vinculados. Cuando el suscriptor se suscribe a un nuevo tema, se actualiza en el SC con los temas correspondientes.

- **Recepción de mensajes:**

El SC recibe los mensajes del publicador a través del pipe nominal PSC, con cada mensaje teniendo un tema, dos puntos un contenido y un punto final.

- **Proceso de “match”:**

El SC extrae el tema del mensaje y verifica en su arreglo de suscriptores qué suscriptores están suscritos a ese tema. Luego reenvía el mensaje a los suscriptores que coinciden, usando sus pipes individuales.

- **Envío a suscriptores:**

Los suscriptores reciben los mensajes de los temas a los que están suscritos cada uno por su pipe único, asegurando una distribución eficiente y precisa de noticias.

3.5 ¿Cómo es el diseño de este proceso para recibir la comunicación de los dos tipos de procesos?

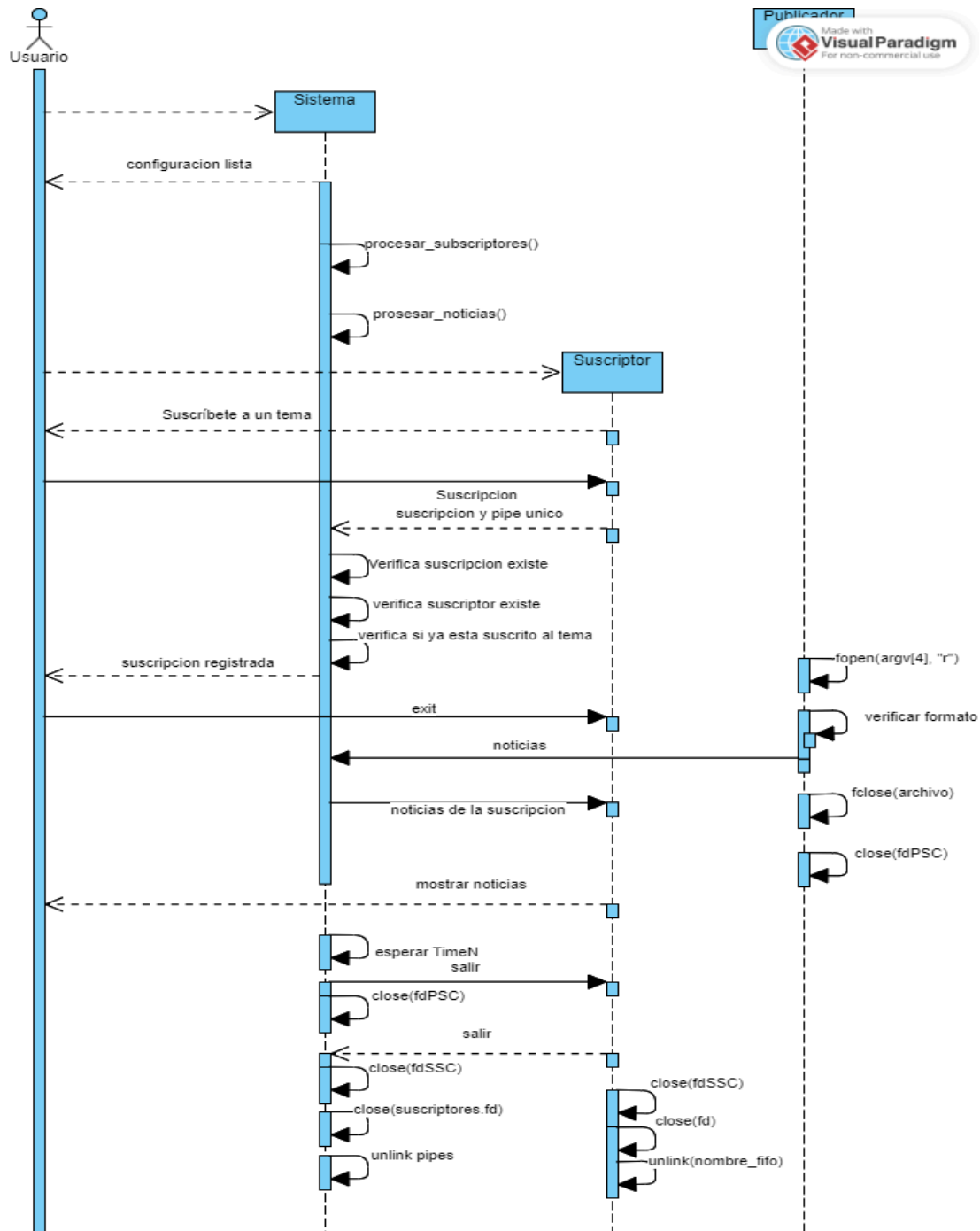


Imagen 2. Diagrama de secuencia UML.

3.6 ¿Cómo deja de funcionar todo el sistema? ¿Cómo lo implementará, con mensajes señales, etc.?

- Proceso de Finalización Controlada:

El SC asegura que todas las noticias en curso sean entregadas y espera el tiempo especificado antes de enviar el mensaje de finalización a los suscriptores y cerrar el pipe nominal PSC.

- Mensajes de Finalización

Cuando el mensaje “salida” es recibido por el suscriptor este envía el mensaje al procesar_suscripciones y luego cierra su pipe unico y el pipe nominal SSC. El procesar_suscripciones al recibir el mensaje de salida procede a cerrar el pipe nominal SSC.

- Liberación de Recursos

Al finalizar el programa el SC cierra sus conexiones con los pipes unicos de los suscriptores y los borra para luego borrar los pipes nominales PSC y SSC.

4. Ejecución y Ejemplos de Uso

- **Ejemplo de Publicador:** `$ publicador -p pipePSC -f noticias.txt -t 5`

Este comando ejecuta un publicador que envía noticias del archivo `noticias.txt` a intervalos de 5 segundos entre noticia y noticia.

- **Ejemplo de Suscriptor:** `$ suscriptor -s pipeSSC`

El suscriptor solicita al usuario que ingrese los temas de interés y se suscribe, esperando recibir noticias solo de esos temas.

- **Ejemplo de Sistema de Comunicación:** `$ sistema -p pipePSC -s pipeSSC -t 10`

Este comando ejecuta el SC, el cual gestionará las suscripciones y noticias por un tiempo adicional de 10 segundos una vez que los publicadores terminen.

- **Orden de ejecución:**

El orden correcto de ejecución es primero iniciar el sistema que es el que abre los pipes nominales, luego abrir el suscriptor y suscribirse a los temas de interés y por último el publicador, si mantenemos este orden nos aseguramos de que el sistema de comunicación funcione correctamente.

5. Casos de prueba

Caso de prueba 1: Comunicación entre dos Publicadores y un Suscriptor.

- Precondiciones:
 - ❖ El código de los publicadores y del suscriptor debe estar compilado y listo para ejecutarse.
 - ❖ Los programas deben estar configurados para funcionar correctamente en el mismo entorno y bajo el mismo mecanismo de comunicación (sockets, pipes, etc.).
 - ❖ El suscriptor debe ser capaz de recibir mensajes desde múltiples publicadores de manera simultánea o en secuencia, dependiendo de la lógica implementada.

- Descripción:
 - ❖ Este caso de prueba verifica si el suscriptor puede recibir datos de dos publicadores, ya sea de manera secuencial o simultánea, y si procesa los datos correctamente.

- Pasos a Seguir:
 - ❖ Ejecutar el primer publicador.
 - ❖ Ejecutar el segundo publicador.
 - ❖ Ejecutar el programa del suscriptor.
 - ❖ El primer publicador envía datos.
 - ❖ El segundo publicador envía datos.
 - ❖ Validar que el suscriptor recibe los datos de ambos publicadores de manera correcta.

- Datos de Entrada:
 - ❖ Datos enviados por el primer publicador.
 - ❖ Datos enviados por el segundo publicador. (Los tipos de datos dependen de la lógica de tu sistema, ya sea números, cadenas de texto, o estructuras más complejas).

- Criterios de Aceptación:
 - ❖ El suscriptor debe recibir correctamente los datos de ambos publicadores, sin pérdida de información.
 - ❖ No deben producirse errores en el proceso de envío ni en la recepción de datos.
 - ❖ El suscriptor debe diferenciar o procesar adecuadamente los datos provenientes de los dos publicadores, si así lo requiere el sistema.

- Resultados Esperados:
 - ❖ El suscriptor recibe los datos enviados por el primer y segundo publicador de forma correcta.
 - ❖ El sistema no arroja errores ni durante la ejecución de los publicadores ni del suscriptor.
- Resultado:

```

~/proyecto$ ./sistema -p pipePSC -s pipeSSC -t 10
Configuración del sistema:
Pipe de publicadores: pipePSC
Pipe de suscriptores: pipeSSC
Tiempo de espera: 10
Suscripción recibida: A /tmp/fifo_suscriptor_1158
Nuevo suscriptor registrado: /tmp/fifo_suscriptor_1158 con tema A
Suscripción recibida: E /tmp/fifo_suscriptor_1158
Nuevo tema agregado: E para el suscriptor /tmp/fifo_suscriptor_1158
Recibida la noticia 'A: El 25 se presenta la orquesta de cámara.
'
' al suscriptor del tema 'A'e presenta la orquesta de cámara.
Recibida la noticia 'P: Se aproximan las elecciones.
'
Recibida la noticia 'P: Hay un nuevo candidato en Alianza Popular.
'
Recibida la noticia 'E: Fue exitosa la presentación de Maluma.'.
Enviando noticia 'E: Fue exitosa la presentación de Maluma.' al suscriptor del tem
a 'E'
Recibida la noticia 'A: El 25 se presenta la orquesta de cámara.
'
' al suscriptor del tema 'A'e presenta la orquesta de cámara.
Recibida la noticia 'P: Se aproximan las elecciones.
'
Recibida la noticia 'P: Hay un nuevo candidato en Alianza Popular.
'
Recibida la noticia 'E: Fue exitosa la presentación de Maluma.'.
Enviando noticia 'E: Fue exitosa la presentación de Maluma.' al suscriptor del tem
a 'E'
La emisión ha terminado
~/proyecto$

```

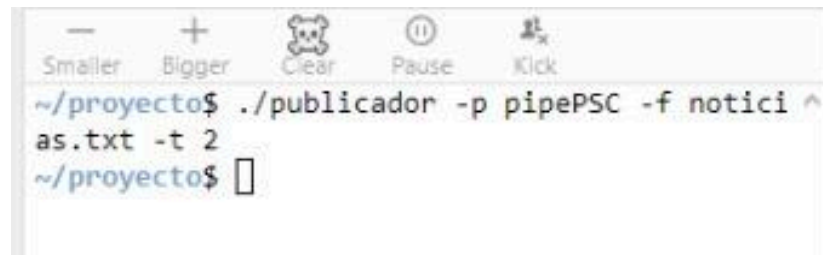
Imagen 3. Resultado sistema para caso de prueba 1.

```

~/proyecto$ ./suscriptor -s pipeSSC
Suscríbete a un tema (A, E, C, P, S) o 'exit'
para terminar: A
Suscripción enviada: A /tmp/fifo_suscriptor_11
58
Suscríbete a un tema (A, E, C, P, S) o 'exit'
para terminar: E
Suscripción enviada: E /tmp/fifo_suscriptor_11
58
Suscríbete a un tema (A, E, C, P, S) o 'exit'
para terminar: exit
Saliendo....
Noticia recibida: A: El 25 se presenta la orqu
esta de cámara.
Noticia recibida: E: Fue exitosa la presentaci
ón de Maluma.
Noticia recibida: A: El 25 se presenta la orqu
esta de cámara.
Noticia recibida: E: Fue exitosa la presentaci
ón de Maluma.
Notificación de salida recibida.
~/proyecto$

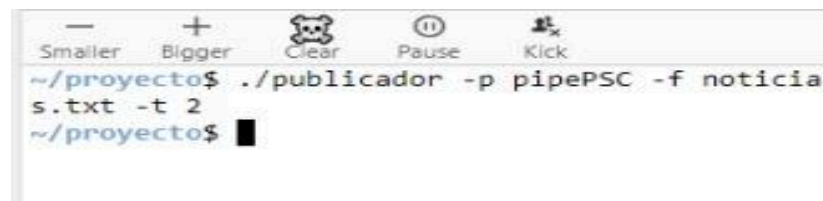
```


Imagen 4. Resultado suscriptor para caso de prueba 1.



```
~/proyecto$ ./publicador -p pipePSC -f notici  
as.txt -t 2  
~/proyecto$
```

Imagen 5. Resultado publicador 1 para caso de prueba 1.



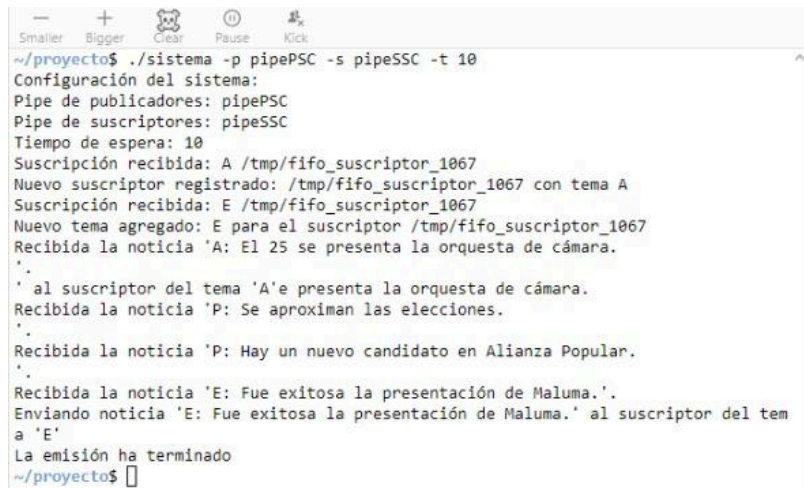
```
~/proyecto$ ./publicador -p pipePSC -f noticia  
s.txt -t 2  
~/proyecto$
```

Imagen 6. Resultado publicador 2 para caso de prueba 1.

Caso de prueba 2: Comunicación entre un Publicador y un Suscriptor.

- Precondiciones:
 - ❖ El código del publicador y del suscriptor debe estar compilado y listo para ser ejecutado.
 - ❖ Ambos programas deben estar configurados para funcionar correctamente en el mismo entorno (ya sea con sockets, pipes, o cualquier mecanismo de comunicación que utilicen).
 - ❖ Descripción:
 - ❖ Este caso de prueba verifica si el publicador puede enviar datos correctamente y si el suscriptor es capaz de recibir esos datos.
- Pasos a Seguir:
 - ❖ Ejecutar el programa del publicador.
 - ❖ Ejecutar el programa del suscriptor.
 - ❖ El publicador envía datos según la lógica implementada.
 - ❖ El suscriptor recibe los datos.
 - ❖ Validar si el suscriptor recibió los datos correctos.
 - ❖ Datos de Entrada:

- ❖ Dependerá del tipo de datos que manejan tus programas. Puede ser un número, una cadena de texto, u otra estructura según el protocolo que utilices.
- Criterios de Aceptación:
 - ❖ El suscriptor debe recibir y procesar correctamente los datos enviados por el publicador.
 - ❖ No deben producirse errores durante el envío o la recepción de datos.
 - ❖ Resultados Esperados:
 - ❖ El suscriptor recibe los datos enviados por el publicador correctamente.
 - ❖ El sistema no debe reportar errores durante la ejecución.
- Resultado:

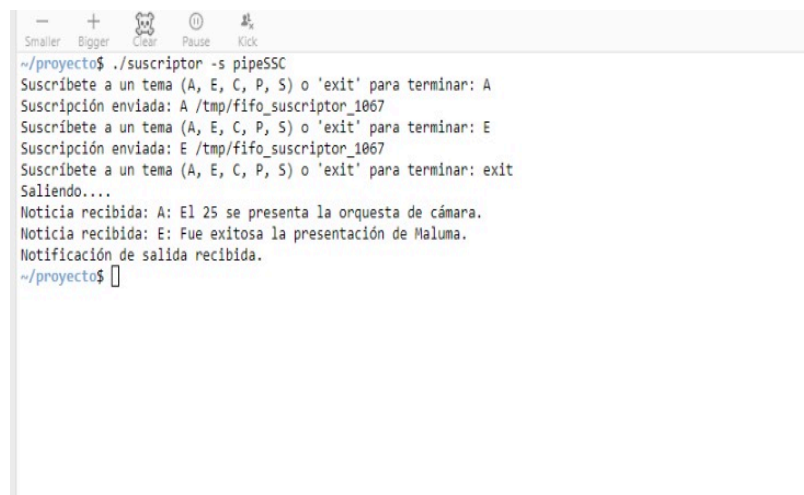


```

~/proyecto$ ./sistema -p pipePSC -s pipeSSC -t 10
Configuración del sistema:
Pipe de publicadores: pipePSC
Pipe de suscriptores: pipeSSC
Tiempo de espera: 10
Suscripción recibida: A /tmp/fifo_suscriptor_1067
Nuevo suscriptor registrado: /tmp/fifo_suscriptor_1067 con tema A
Suscripción recibida: E /tmp/fifo_suscriptor_1067
Nuevo tema agregado: E para el suscriptor /tmp/fifo_suscriptor_1067
Recibida la noticia 'A: El 25 se presenta la orquesta de cámara.
'
' al suscriptor del tema 'A' se presenta la orquesta de cámara.
Recibida la noticia 'P: Se aproximan las elecciones.
'
Recibida la noticia 'P: Hay un nuevo candidato en Alianza Popular.
'
Recibida la noticia 'E: Fue exitosa la presentación de Maluma.'
Enviando noticia 'E: Fue exitosa la presentación de Maluma.' al suscriptor del tem
a 'E'
La emisión ha terminado
~/proyecto$

```

Imagen 7. Resultado sistema para caso de prueba 2.

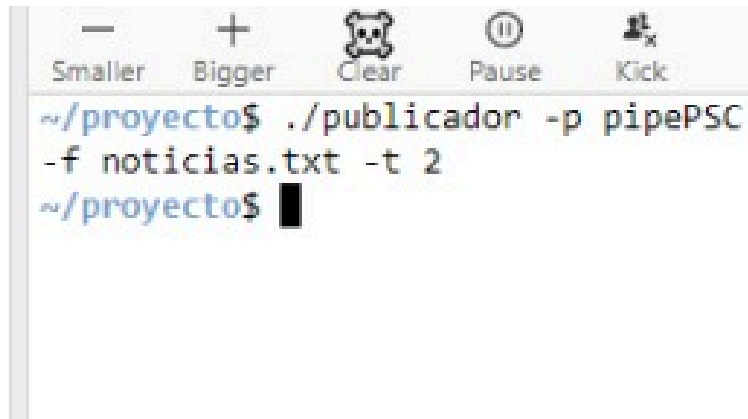


```

~/proyecto$ ./suscriptor -s pipeSSC
Suscríbete a un tema (A, E, C, P, S) o 'exit' para terminar: A
Suscripción enviada: A /tmp/fifo_suscriptor_1067
Suscríbete a un tema (A, E, C, P, S) o 'exit' para terminar: E
Suscripción enviada: E /tmp/fifo_suscriptor_1067
Suscríbete a un tema (A, E, C, P, S) o 'exit' para terminar: exit
Saliendo....
Noticia recibida: A: El 25 se presenta la orquesta de cámara.
Noticia recibida: E: Fue exitosa la presentación de Maluma.
Notificación de salida recibida.
~/proyecto$

```

Imagen 8. Resultado suscriptor para caso de prueba 2.

A screenshot of a terminal window with a light gray background. At the top, there is a toolbar with five icons: a minus sign (Smaller), a plus sign (Bigger), a skull (Clear), a pause symbol (Pause), and a person with an 'x' (Kick). Below the toolbar, the terminal shows a command prompt at the root of a directory named 'proyecto'. The user enters the command './publicador -p pipePSC -f noticias.txt -t 2'. The prompt returns to '~ /proyecto\$' with a black cursor at the end.

```
~/proyecto$ ./publicador -p pipePSC  
-f noticias.txt -t 2  
~/proyecto$ █
```

Imagen 9. Resultado publicador para caso de prueba 2.

Caso de prueba 3: Comunicación entre un Publicador y dos Suscriptores.

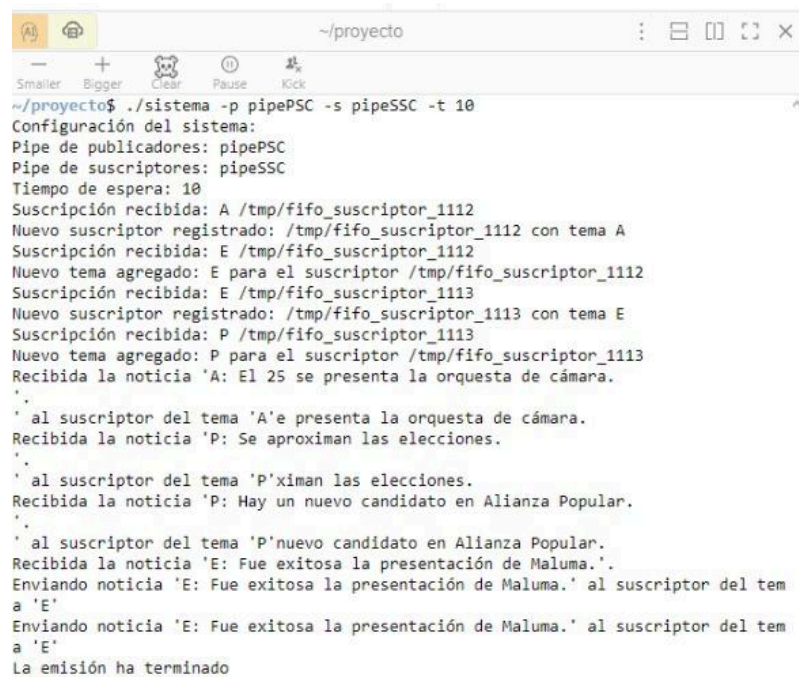
- Precondiciones:
 - ❖ El código del publicador y de ambos suscriptores debe estar compilado y listo para ejecutarse.
 - ❖ Los programas deben estar configurados para funcionar correctamente en el mismo entorno y bajo el mismo mecanismo de comunicación (sockets, pipes, etc.).
 - ❖ Los suscriptores deben poder conectarse al publicador y estar listos para recibir mensajes simultáneamente.
- Descripción:
 - ❖ Este caso de prueba verifica si el publicador puede enviar datos correctamente a dos suscriptores y si ambos suscriptores reciben los datos sin problemas.
- Pasos a Seguir:
 - ❖ Ejecutar el programa del publicador.
 - ❖ Ejecutar el primer suscriptor.
 - ❖ Ejecutar el segundo suscriptor.
 - ❖ El publicador envía datos.
 - ❖ Validar que ambos suscriptores recibieron los datos correctamente.
- Datos de Entrada:
- Criterios de Aceptación:
 - ❖ Ambos suscriptores deben recibir los datos enviados por el publicador sin pérdida de información.
 - ❖ No deben producirse errores en el proceso de envío ni en la recepción de datos.

- ❖ Los dos suscriptores deben recibir exactamente los mismos datos si el sistema está configurado para difusión.

- Resultados Esperados:

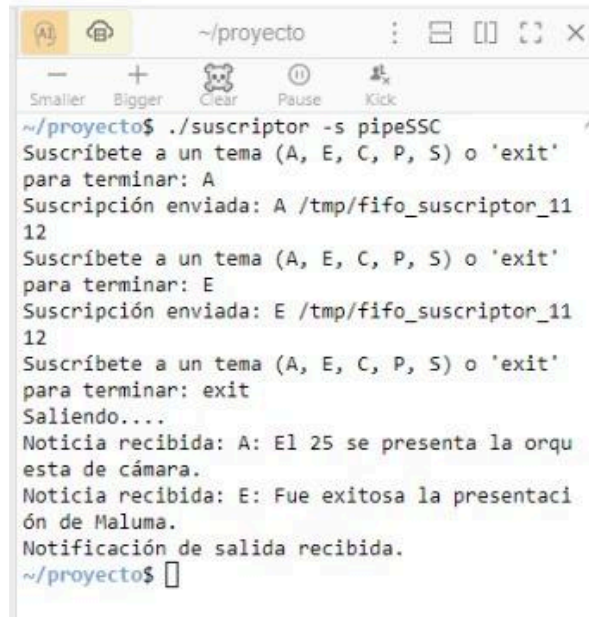
- ❖ El publicador envía los datos.
- ❖ Ambos suscriptores reciben los datos correctamente y los procesan.
- ❖ El sistema no debe arrojar errores durante la ejecución.

- Resultado:



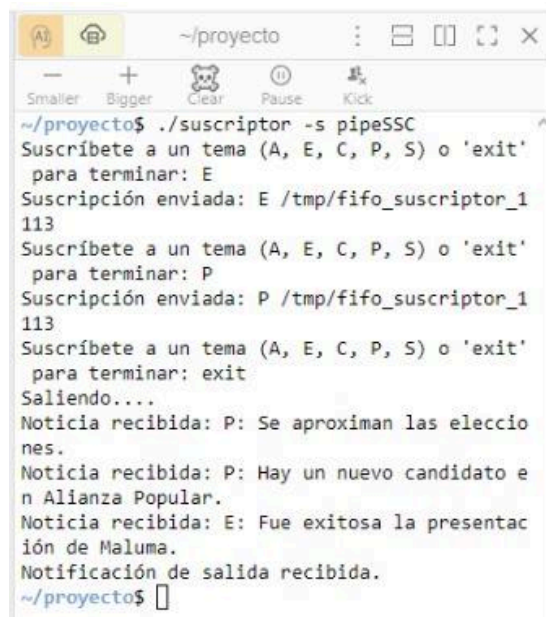
```
~/proyecto
~
Smaller Bigger Clear Pause Kick
~/proyecto$ ./sistema -p pipePSC -s pipeSSC -t 10
Configuración del sistema:
Pipe de publicadores: pipePSC
Pipe de suscriptores: pipeSSC
Tiempo de espera: 10
Suscripción recibida: A /tmp/fifo_suscriptor_1112
Nuevo suscriptor registrado: /tmp/fifo_suscriptor_1112 con tema A
Suscripción recibida: E /tmp/fifo_suscriptor_1112
Nuevo tema agregado: E para el suscriptor /tmp/fifo_suscriptor_1112
Suscripción recibida: E /tmp/fifo_suscriptor_1113
Nuevo suscriptor registrado: /tmp/fifo_suscriptor_1113 con tema E
Suscripción recibida: P /tmp/fifo_suscriptor_1113
Nuevo tema agregado: P para el suscriptor /tmp/fifo_suscriptor_1113
Recibida la noticia 'A: El 25 se presenta la orquesta de cámara.
'
' al suscriptor del tema 'A'e presenta la orquesta de cámara.
Recibida la noticia 'P: Se aproximan las elecciones.
'
' al suscriptor del tema 'P'ximan las elecciones.
Recibida la noticia 'P: Hay un nuevo candidato en Alianza Popular.
'
' al suscriptor del tema 'P'nuevo candidato en Alianza Popular.
Recibida la noticia 'E: Fue exitosa la presentación de Maluma.'.
Enviando noticia 'E: Fue exitosa la presentación de Maluma.' al suscriptor del tem
a 'E'
Enviando noticia 'E: Fue exitosa la presentación de Maluma.' al suscriptor del tem
a 'E'
La emisión ha terminado
```

Imagen 10. Resultado sistema para caso de prueba 3.



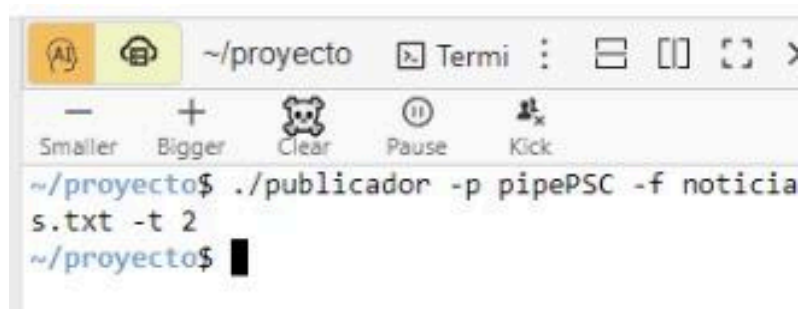
```
~/proyecto
Smaller Bigger Clear Pause Kick
~/proyecto$ ./suscriptor -s pipeSSC
Suscríbete a un tema (A, E, C, P, S) o 'exit'
para terminar: A
Suscripción enviada: A /tmp/fifo_suscriptor_11
12
Suscríbete a un tema (A, E, C, P, S) o 'exit'
para terminar: E
Suscripción enviada: E /tmp/fifo_suscriptor_11
12
Suscríbete a un tema (A, E, C, P, S) o 'exit'
para terminar: exit
Saliendo...
Noticia recibida: A: El 25 se presenta la orqu
esta de cámara.
Noticia recibida: E: Fue exitosa la presentaci
ón de Maluma.
Notificación de salida recibida.
~/proyecto$
```

Imagen 11. Resultado suscriptor 1 para caso de prueba 3.



```
~/proyecto
Smaller Bigger Clear Pause Kick
~/proyecto$ ./suscriptor -s pipeSSC
Suscríbete a un tema (A, E, C, P, S) o 'exit'
para terminar: E
Suscripción enviada: E /tmp/fifo_suscriptor_1
113
Suscríbete a un tema (A, E, C, P, S) o 'exit'
para terminar: P
Suscripción enviada: P /tmp/fifo_suscriptor_1
113
Suscríbete a un tema (A, E, C, P, S) o 'exit'
para terminar: exit
Saliendo...
Noticia recibida: P: Se aproximan las eleccio
nes.
Noticia recibida: P: Hay un nuevo candidato e
n Alianza Popular.
Noticia recibida: E: Fue exitosa la presentac
ión de Maluma.
Notificación de salida recibida.
~/proyecto$
```

Imagen 12. Resultado suscriptor 2 para caso de prueba 3.



```
~/proyecto
Smaller Bigger Clear Pause Kick
~/proyecto$ ./publicador -p pipePSC -f noticia
s.txt -t 2
~/proyecto$
```

Imagen 13. Resultado publicador para caso de prueba 3.

Caso de prueba 4: Comunicación entre dos Publicadores y dos Suscriptores.

- Precondiciones:
 - ❖ El código de los publicadores y de los suscriptores debe estar compilado y listo para ejecutarse.
 - ❖ Todos los programas deben estar configurados para funcionar correctamente en el mismo entorno y utilizando el mismo mecanismo de comunicación (sockets, pipes, etc.).
 - ❖ Los suscriptores deben estar preparados para recibir datos de múltiples publicadores simultáneamente o en secuencia, según la implementación.

- Descripción:
 - ❖ Este caso de prueba verifica si los dos publicadores pueden enviar datos a dos suscriptores, y si ambos suscriptores reciben y procesan correctamente los datos provenientes de ambos publicadores.

- Pasos a Seguir:
 - ❖ Ejecutar el primer publicador.
 - ❖ Ejecutar el segundo publicador.
 - ❖ Ejecutar el primer suscriptor.
 - ❖ Ejecutar el segundo suscriptor.
 - ❖ El primer publicador envía datos.
 - ❖ El segundo publicador envía datos.
 - ❖ Validar que ambos suscriptores reciben correctamente los datos de ambos publicadores.

- Datos de Entrada:

- Criterios de Aceptación:
 - ❖ Ambos suscriptores deben recibir correctamente los datos enviados por ambos publicadores.
 - ❖ No debe haber pérdida de datos ni errores en el proceso de envío o recepción.
 - ❖ Si el sistema está diseñado para procesar los datos de forma concurrente, ambos suscriptores deben recibir los datos de ambos publicadores simultáneamente o secuencialmente, sin problemas de sincronización.

- Resultados Esperados:

- ❖ El primer y segundo publicador envían los datos sin errores.
- ❖ El primer y segundo suscriptor reciben y procesan correctamente los datos de ambos publicadores.
- ❖ No se presentan errores durante la ejecución.

Resultado Actual:

```

~/proyecto$ ./sistema -p pipePSC -s pipeSSC -t 10
Configuración del sistema:
Pipe de publicadores: pipePSC
Pipe de suscriptores: pipeSSC
Tiempo de espera: 10
Suscripción recibida: A /tmp/fifo_suscriptor_1289
Nuevo suscriptor registrado: /tmp/fifo_suscriptor_1289 con tema A
Suscripción recibida: E /tmp/fifo_suscriptor_1289
Nuevo tema agregado: E para el suscriptor /tmp/fifo_suscriptor_1289
Suscripción recibida: P /tmp/fifo_suscriptor_1290
Nuevo suscriptor registrado: /tmp/fifo_suscriptor_1290 con tema P
Recibida la noticia 'A: El 25 se presenta la orquesta de cámara.
'
' al suscriptor del tema 'A'e presenta la orquesta de cámara.
Recibida la noticia 'P: Se aproximan las elecciones.
'
' al suscriptor del tema 'P'ximan las elecciones.
Recibida la noticia 'A: El 25 se presenta la orquesta de cámara.
'
' al suscriptor del tema 'A'e presenta la orquesta de cámara.
Recibida la noticia 'P: Hay un nuevo candidato en Alianza Popular.
'
' al suscriptor del tema 'P'nuevo candidato en Alianza Popular.
Recibida la noticia 'P: Se aproximan las elecciones.
'
' al suscriptor del tema 'P'ximan las elecciones.
Recibida la noticia 'E: Fue exitosa la presentación de Maluma.'.
Enviando noticia 'E: Fue exitosa la presentación de Maluma.' al suscriptor del tema 'E'
Recibida la noticia 'P: Hay un nuevo candidato en Alianza Popular.
'
' al suscriptor del tema 'P'nuevo candidato en Alianza Popular.
Recibida la noticia 'E: Fue exitosa la presentación de Maluma.'.
Enviando noticia 'E: Fue exitosa la presentación de Maluma.' al suscriptor del tema 'E'
La emisión ha terminado
~/proyecto$

```

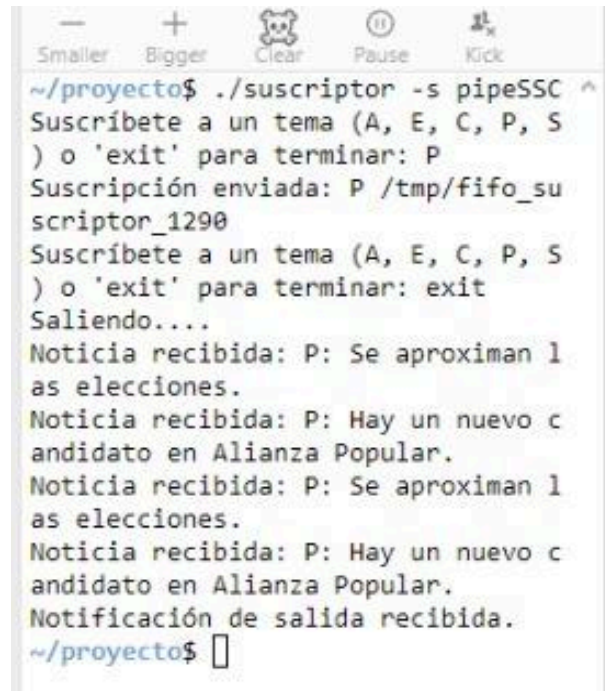
Imagen 14. Resultado sistema caso de prueba 4.

```

~/proyecto$ ./suscriptor -s pipeSSC
Suscríbete a un tema (A, E, C, P, S) o '
exit' para terminar: A
Suscripción enviada: A /tmp/fifo_suscrip
tor_1289
Suscríbete a un tema (A, E, C, P, S) o '
exit' para terminar: E
Suscripción enviada: E /tmp/fifo_suscrip
tor_1289
Suscríbete a un tema (A, E, C, P, S) o '
exit' para terminar: exit
Saliendo....
Noticia recibida: A: El 25 se presenta l
a orquesta de cámara.
Noticia recibida: A: El 25 se presenta l
a orquesta de cámara.
Noticia recibida: E: Fue exitosa la pres
entación de Maluma.
Noticia recibida: E: Fue exitosa la pres
entación de Maluma.
Notificación de salida recibida.
~/proyecto$


```

Imagen 15.Resultado suscriptor 1 caso de prueba 4.

A terminal window with a light gray background and a toolbar at the top containing icons for 'Smaller', 'Bigger', 'Clear', 'Pause', and 'Kick'. The terminal text shows the execution of the './suscriptor -s pipeSSC ^' command. The program prompts the user to subscribe to a topic (A, E, C, P, S) or 'exit'. The user enters 'P', and the program sends a subscription to '/tmp/fifo_suscriptor_1290'. The user then enters 'exit', and the program prints 'Saliendo....'. Subsequently, the program receives four notifications from the pipe, each starting with 'Noticia recibida: P:'. The first and third notifications are 'Se aproximan las elecciones.', while the second and fourth are 'Hay un nuevo candidato en Alianza Popular.'. The terminal ends with the prompt '~/proyecto\$' and a cursor.

```
~/proyecto$ ./suscriptor -s pipeSSC ^
Suscríbete a un tema (A, E, C, P, S ) o 'exit' para terminar: P
Suscripción enviada: P /tmp/fifo_suscriptor_1290
Suscríbete a un tema (A, E, C, P, S ) o 'exit' para terminar: exit
Saliendo....
Noticia recibida: P: Se aproximan las elecciones.
Noticia recibida: P: Hay un nuevo candidato en Alianza Popular.
Noticia recibida: P: Se aproximan las elecciones.
Noticia recibida: P: Hay un nuevo candidato en Alianza Popular.
Notificación de salida recibida.
~/proyecto$
```

Imagen 16. Resultado suscriptor 2 caso de prueba 4.

Two side-by-side terminal windows, each with a light gray background and a toolbar at the top with icons for 'Smaller', 'Bigger', 'Clear', 'Pause', and 'Kick'. Both windows show the execution of the './publicador -p pipePSC -f noticias.txt -t 2' command. The left window shows the prompt '~/proyecto\$' and a cursor. The right window shows the prompt '~/proyecto\$' followed by a solid black block.

```
~/proyecto$ ./publicador -p pipePSC -f noticias.txt -t 2
~/proyecto$
```

```
~/proyecto$ ./publicador -p pipePSC -f noticias.txt -t 2
~/proyecto$
```

Imagen 17. Resultado publicador 1 y 2 caso de prueba 4

6. Repositorio

- Link al repositorio:
<https://github.com/JFortich-Ordosgoitia/Jorge-Andr-s-Fortich-Ordosgoitia/tree/main/Proyecto>