Tarea 1 POO

Benjamin Arias Vega 202230007-8
Joaquin Belmar Leal 202330003-9

April 29, 2025

Explicación Breve de la Solución

El sistema implementa un modelo de publicaciones y suscripciones donde un **Broker** centraliza los mensajes enviados por los **Publicadores**, que son recibidos por distintos tipos de **Suscriptores** (*Follower* y *Monitor*). La configuración inicial se carga desde un archivo `config.txt` y permite crear múltiples publicadores y suscriptores asociados a tópicos específicos. Durante la ejecución, los usuarios ingresan eventos manualmente, los cuales son procesados según los tópicos y almacenados en archivos de salida.

Dificultades Encontradas y Soluciones

1. **Dificultad:** Manejar distintos tipos de suscriptores en la función `setupSimulator`.
   Solución: Se utilizó una estructura condicional `else if` para crear correctamente *Follower* o *Monitor* según corresponda.

2. **Dificultad:** Organizacion del codigo debido a su entrega por etapas
   Solución: Fue complicado imaginarse el orden de cada etapa y en que parte realizarla, pero a traves de GitHub logramos facilitar el avance.

3. **Dificultad:** Hacer que la ejecución termine de forma controlada con una entrada vacía.
   Solución: Se reemplazó el `continue` por un `break` en el `runSimulation()` para detectar entradas vacías y finalizar el programa adecuadamente.

Información Adicional

- No se implementó la etapa adicional con bonificación. - Se utilizó un Makefile para compilar y ejecutar automáticamente. - El sistema fue probado localmente y en el servidor Aragorn de la universidad.

Diagrama UML

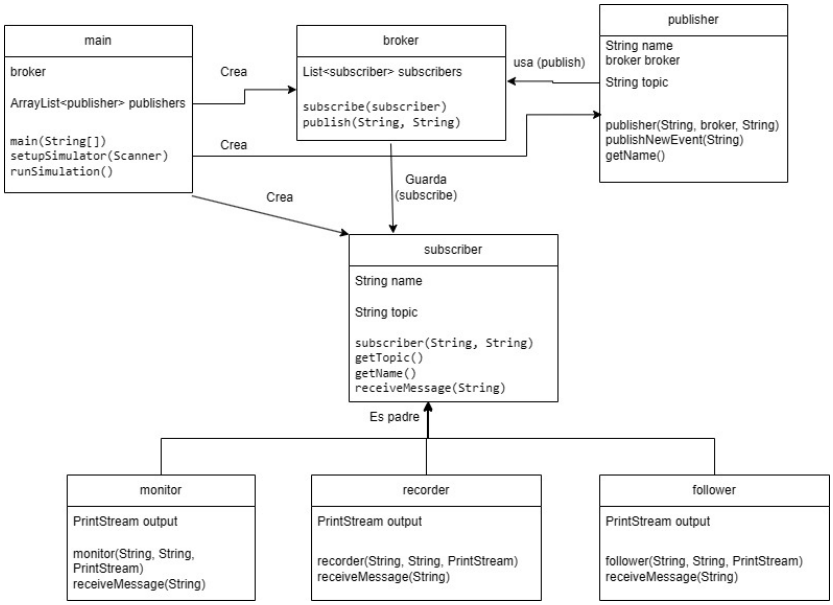

Figure 1: Enter Caption