

Documentación Tarea 2

Integrantes:

Bianca León – Rol: 202330527-8

Gabriela Trigo - Rol: 202330561-8

Simulador Gráfico del Patrón Publicador-Suscriptor

Esta tarea extiende y mejora la Tarea 1, adaptándola para simular gráficamente el patrón de diseño publicador-suscriptor utilizando JavaFX. El objetivo es modelar y visualizar la interacción entre publicadores y suscriptores en dos dominios principales: publicación de URLs de videos y seguimiento de posiciones GPS de un móvil.

Etapas del Proyecto

Etapas 1

En la primera etapa se implementa lo básico para simular el patrón publicador-suscriptor con videos. Se reutilizan las clases fundamentales de la tarea anterior y se crean las clases *VideoPublisher* y *VideoFollower*. Al crear un publicador, se pide ingresar su nombre y el tópico mediante ventanas de diálogo. El *VideoPublisher* tiene un campo de texto donde se pueden ingresar URLs de videos, que se publican al presionar ENTER y se limpia el campo para nuevos ingresos. Los suscriptores muestran botones que contienen el último URL publicado para su tópico, pero en esta etapa el botón no responde al presionarlo. Todo se maneja desde una interfaz gráfica con un menú para crear publicadores y suscriptores.

Etapas 2

Para esta etapa, el *VideoFollower* mejora para que al presionar el botón con la URL publicada, el video se reproduzca dentro del programa. Se agrega un reproductor multimedia que permite ver el video sin salir del simulador. Esto hace que la interacción entre publicadores y suscriptores sea más completa.

Etapas 3

En esta etapa se desarrolla la clase *GPSCarPublisher*, que extiende la clase *Publisher* para simular la posición GPS de un móvil. Esta clase solicita al usuario, al crear una instancia, ingresar el nombre y tópico del publicador, y luego seleccionar un archivo con las posiciones relevantes mediante un *FileChooser*. El archivo debe contener líneas con el formato <tiempo> <posX> <posY>, donde el tiempo es un entero en segundos.

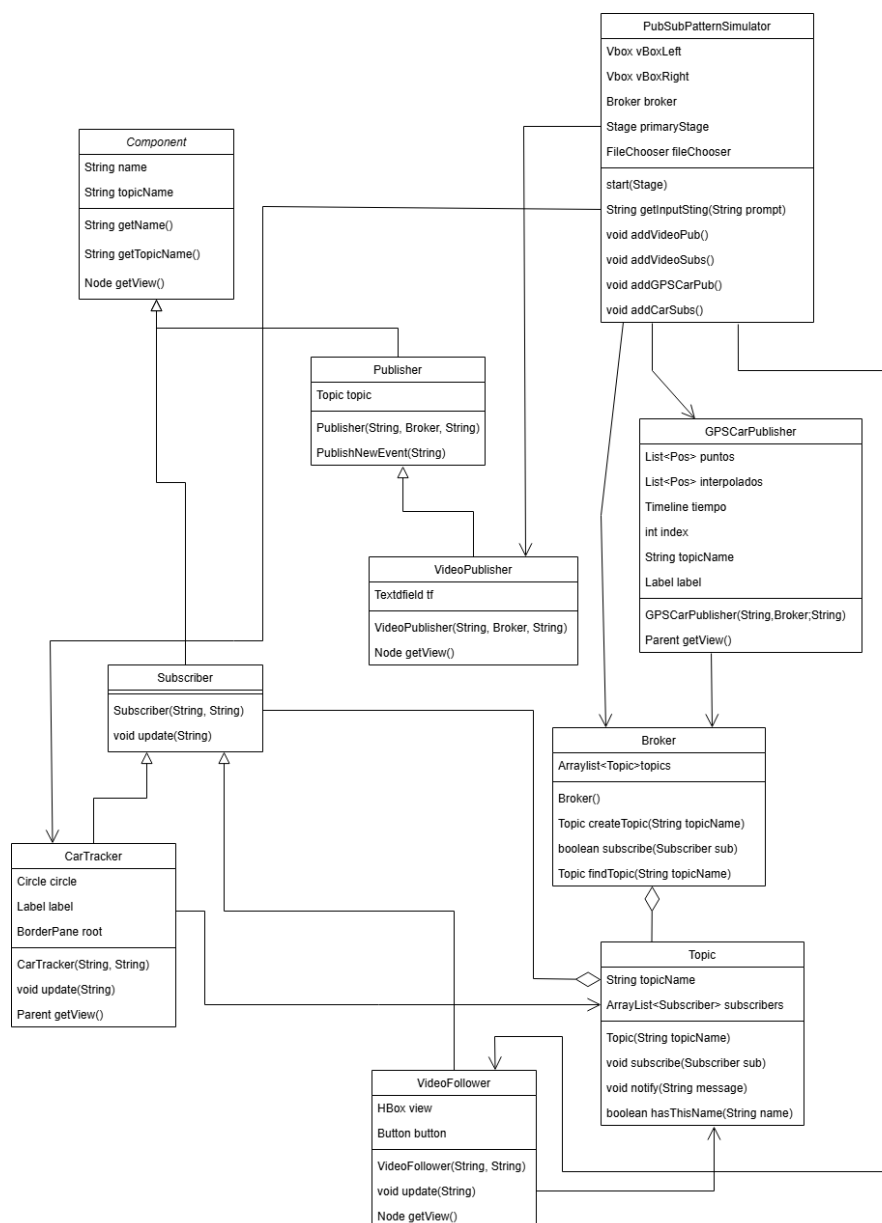
GPSCarPublisher utiliza un mecanismo basado en Timeline para enviar una actualización de posición cada segundo, incluso cuando los datos originales del archivo tienen intervalos mayores. Para ello, interpola linealmente las posiciones entre dos puntos consecutivos para generar un movimiento suave y continuo.

Cada vez que se publica una nueva posición, los suscriptores de ese tópico reciben la información y actualizan su vista. Los suscriptores muestran una ventana independiente con un círculo que representa la posición actual del móvil, además de mostrar el tiempo y las coordenadas, permitiendo así un seguimiento en tiempo real.

Etapa 4

En esta etapa se junta todo lo anterior para que el simulador funcione completo. Ahora se pueden crear muchos publicadores, como varios que mandan videos o posiciones GPS, y varios tipos de suscriptores diferentes. También se agrega un suscriptor nuevo llamado Monitor, que solo guarda posiciones que estén lejos del origen. El programa se configura leyendo un archivo de texto y para terminar solo hay que presionar ENTER.

Diagrama UML



Detalles de Uso y Ejecución

Antes de ejecutar el proyecto se debe considerar tener clonado el repositorio de GitHub con el siguiente comando:

git clone <https://github.com/grupo-8-leon.bianca-trigo.gabriela-Tarea2.2025.1.git>

Una vez clonado, se debe navegar a la carpeta raíz del proyecto para continuar.

Luego para ejecutar el proyecto en las diferentes etapas, se debe usar los siguientes comandos en la terminal:

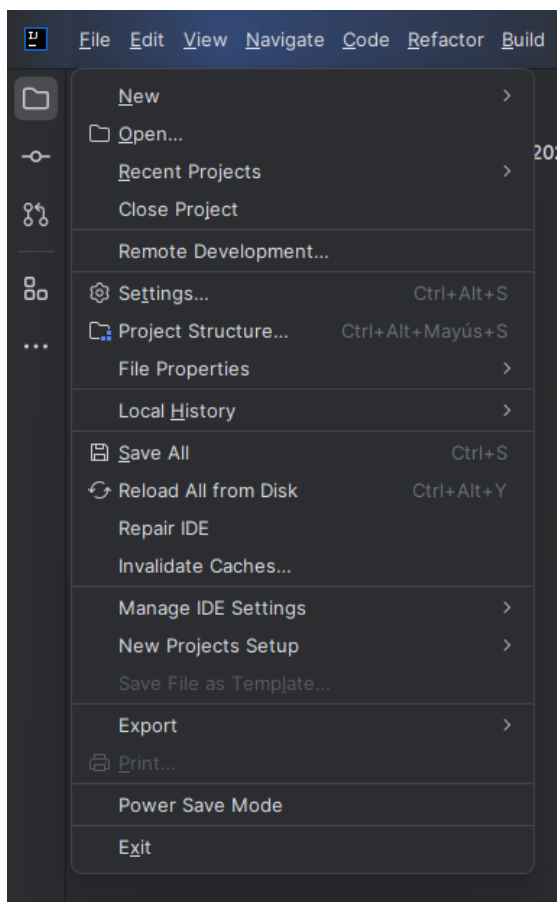


Figura 1

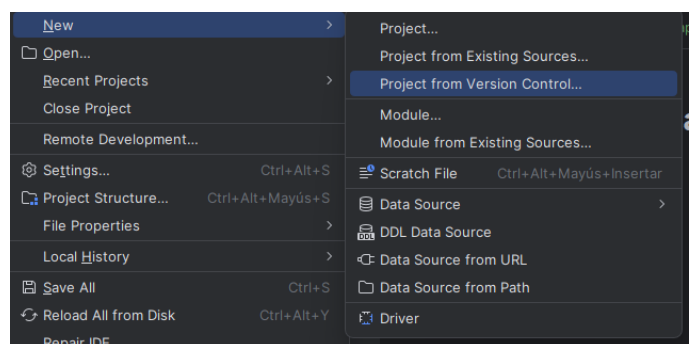


Figura 2

Para clonar el repositorio dirigirse a:
File>New>Project from Version Control...

Como se muestra en la figura 1 y 2.

En la figura 3 se muestra cómo se seleccionar GitHub con su cuenta ya registrada en IntelliJ y entrada al repositorio previamente y seleccionar **bleonkaroi/grupo-8-leon.bianca-trigo.gabriela-Tarea2.2025.1**

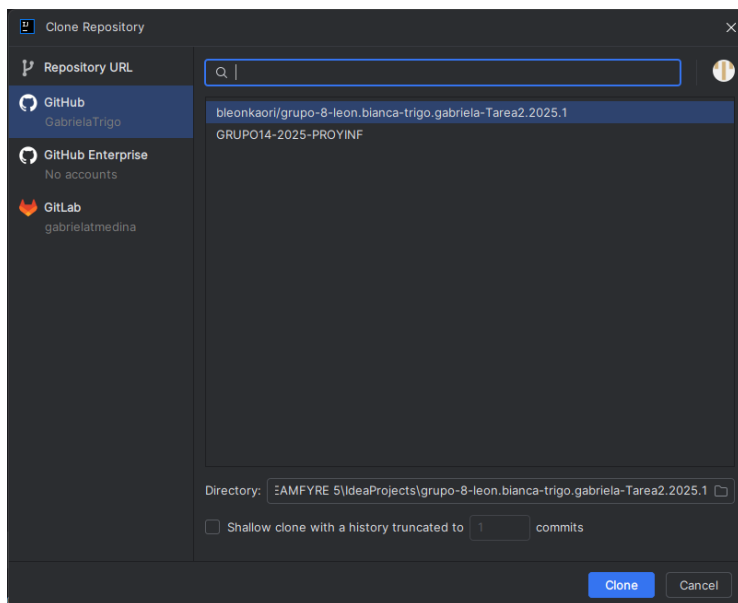
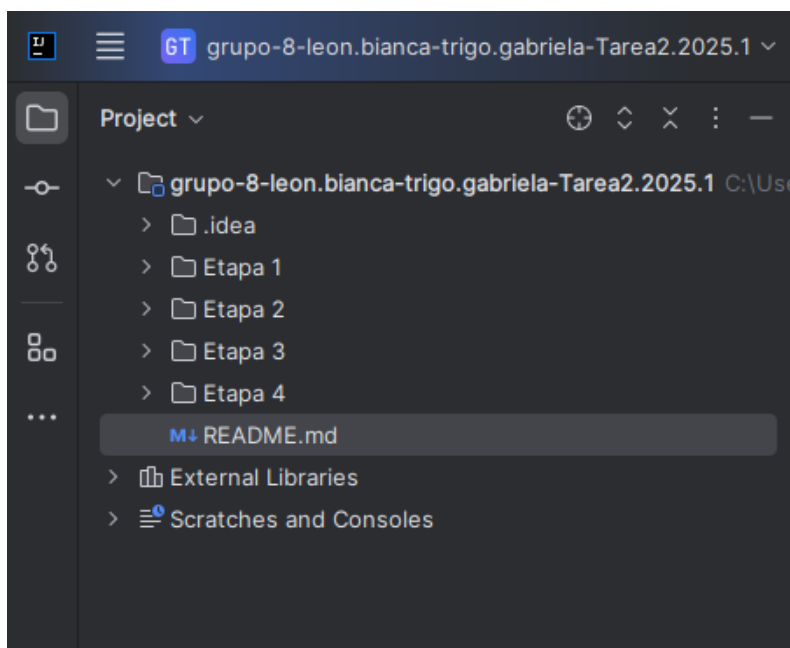
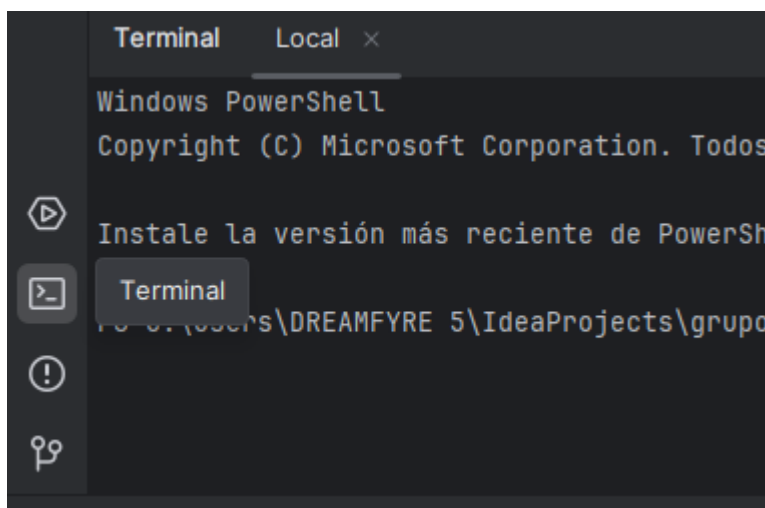


Figura 3



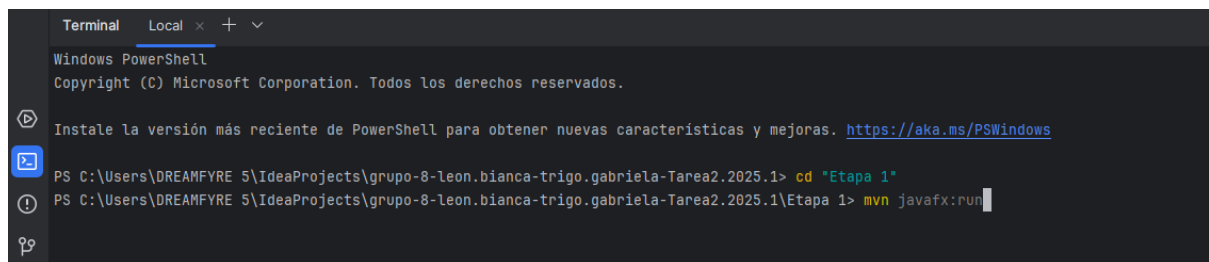
Cuando se clone correctamente debe estar en la siguiente forma las carpetas como en la figura 4.

Figura 4



Para abrir la terminal se pueden de dos maneras: con el ícono que aparece en la figura 5 o con alt+f12.

Figura 5



```
Terminal Local x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\DREAMFYRE 5\IdeaProjects\grupo-8-leon.bianca-trigo.gabriela-Tarea2.2025.1> cd "Etapa 1"
PS C:\Users\DREAMFYRE 5\IdeaProjects\grupo-8-leon.bianca-trigo.gabriela-Tarea2.2025.1\Etapa 1> mvn javafx:run
```

Aquí se introduce los comandos para hacer funcionar las etapas 1,2,3 y 4.

Figura 6

Comandos para las Etapas 1,2,3 y 4

- | | |
|----------------|----------------------------------------------------------------------------------------|
| cd "Etapa 1" | ➔ Para entrar a la carpeta "Etapa 1" |
| mvn javafx:run | ➔ mvn javafx:run compila y ejecuta la aplicación JavaFX para la etapa correspondiente. |
| mvn clean | ➔ mvn clean limpia la carpeta target generada para evitar conflictos. |
| cd .. | ➔ cd .. para volver a la carpeta raíz |

-
- | | |
|----------------|----------------------------------------------------------------------------------------|
| cd "Etapa 2" | ➔ Para entrar a la carpeta "Etapa 2" |
| mvn javafx:run | ➔ mvn javafx:run compila y ejecuta la aplicación JavaFX para la etapa correspondiente. |
| mvn clean | ➔ mvn clean limpia la carpeta target generada para evitar conflictos. |
| cd .. | ➔ cd .. para volver a la carpeta raíz |

-
- | | |
|----------------|----------------------------------------------------------------------------------------|
| cd "Etapa 3" | ➔ Para entrar a la carpeta "Etapa 3" |
| mvn javafx:run | ➔ mvn javafx:run compila y ejecuta la aplicación JavaFX para la etapa correspondiente. |
| mvn clean | ➔ mvn clean limpia la carpeta target generada para evitar conflictos. |
| cd .. | ➔ cd .. para volver a la carpeta raíz |
-

cd "Etapa 4"	➔ Para entrar a la carpeta "Etapa 4"
mvn javadoc:javadoc	➔ mvn javafx:javadoc con la cual se crea una carpeta target/site/apidocs/ dentro de la etapa. (Para ver la documentación, abre el archivo index.html haciendo doble clic sobre él.)
mvn clean	
cd ..	➔ mvn clean limpia la carpeta target generada para evitar conflictos. ➔ cd .. para volver a la carpeta raíz

NOTA: La documentación generada por mvn javadoc:javadoc queda en la carpeta target/site/apidocs/ dentro de la etapa 4. El archivo principal para abrir es index.html.

Referencias:

Wikipedia: Patrón Publicador-Suscriptor (Publish–subscribe pattern)

Documentación oficial JavaFX

Material de apoyo asignatura ELO329 (UTFSM)

Wikipedia: Modelo Vista Controlador (MVC)