

TÍTULO DE GRADO EN

INGENIERÍA INFORMÁTICA

**Tecnologías de Desarrollo de Software**

Práctica final

**Documentación**

CURSO 2018 / 2019

CONVOCATORIA DE ENERO

Grechyshkin Vladyslav – vladyslav.grechyshkin@um.es

Valero Leal Enrique – enrique.valerol@um.es

Grupo 2.2

**Índice general**

[I. Sistema Operativo xv6 1](#_Toc532923324)

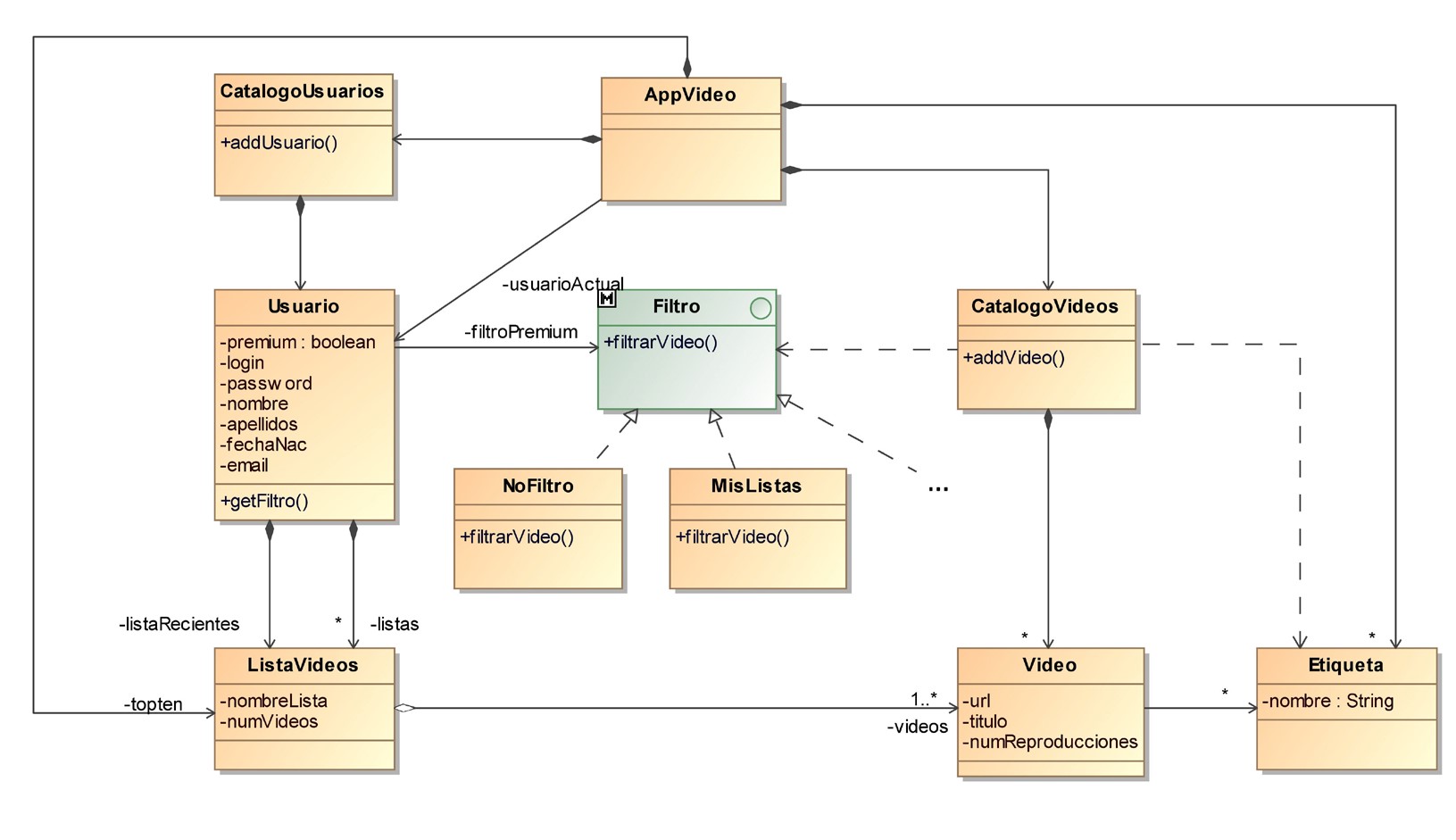
[**1.1 – Boletín 7 1**](#_Toc532923325)

[**1.2 – Boletín 8 1**](#_Toc532923326)

[**1.2 – Boletín 9 2**](#_Toc532923327)

# I. Diagrama de clases del dominio

Adjuntamos el diagrama de clases propuesto por el profesorado y utilizado en el desarrollo de nuestra práctica.



**Ilustración 1. Diagrama de clases del dominio utilizado**

Nótese que ignoramos el diagrama de clases diseñado para la tarea parcial propuesta sobre este mismo, puesto que no es el utilizado en la práctica.

# II. Diagrama de colaboración para añadir un vídeo

Adjuntamos a continuación el diagrama realizado en *MagicDraw*, mostrando la secuencia de invocación de cada una de las clases involucradas para añadir vídeos a una lista.

# III. Arquitectura de la aplicación

La arquitectura general de la aplicación está basada directamente en la arquitectura de tres capas estudiada en la asignatura *(Modelo-Vista-Controlador)* y el principio de «separación modelo-vista». Excluimos cualquier explicación de dicho patrón arquitectural dado que no es el propósito de esta documentación. No obstante, resaltamos que, en la jerarquía de directorios y paquetes de la aplicación, existen, además de cada una de las tres capas, un paquete para las clases relacionadas con la persistencia para proporcionar mayor legibilidad y separación. Asimismo, incluimos un directorio con todos los recursos visuales utilizados para el desarrollo de la interfaz gráfica.

Sobre las decisiones de diseño, destacamos fundamentalmente el uso de las librerías JavaFX y JFoenix *(explicada en el apartado de componentes de este mismo documento)* de Java para la GUI. Asimismo, a diferencia de Swing, donde los elementos de la interfaz creados son añadidos estrictamente como código Java, el uso de la librería JavaFX y la herramienta visual JavaFX Scene Buildder nos permite generar ficheros «FXML»que contienen la jerarquía de ventanas y elementos de la aplicación en un lenguaje de marcas similar a XML, separando de manera efectiva el control de la vista de su creación.

Este fichero FXML *(en nuestro caso, «Root.fxml»)* se carga en la inicialización de la aplicación junto a la hoja de estilo utilizada, en formato .css, y el controlador de la vista *(especificado dentro del fichero FXML)*.

En cuanto a la estructura de nuestra GUI, destacamos el uso de un único fichero FXML *(bajo recomendación del profesorado)* para representar todas las ventanas indicadas en la especificación de la práctica. Esto facilita su implementación, reduce la complejidad y número de archivos, pero aumenta drásticamente el tamaño del controlador de la vista y no resulta de gran escalabilidad.

De este modo, la aplicación consta, en su nodo raíz *(a nivel de ventanas)*, de un *BorderPane* que, en su región central, contiene un *StackPane* con cada una de las ventanas *(Explorar, Mis Listas, Recientes, etc…)* de la aplicación, inicialmente invisibles. Para “abrir” cualquiera de ellas, simplemente volvemos invisible la ventana que estuviese mostrada, traemos la deseada al frente y la volvemos visible.

Como matices adicionales de elección de diseño, subrayamos la inclusión de la ventana «Nueva Lista», indicada en la especificación, dentro de la ventana de «Mis listas». Consideramos oportuno este cambio puesto que los elementos de cada una de las ventanas son casi idénticos *(en cuanto a la visualización de listas y vídeos)*. En vez de duplicarlos, por la falta de escalabilidad que hemos indicado antes, los fusionamos. Esta modificación queda reflejada en la capacidad de manejo total de las listas del usuario en una única ventana, en vez de separar la creación, edición y borrado de la reproducción.

Asimismo, incluimos una ventana adicional llamada «Mi perfil» que permite la modificación y consulta de la información del usuario actual de manera más extensa que la pedida en la especificación.

# IV. Patrones de diseño utilizados

Texto

## **1.1 – Títulos**

Texto

# V. Componentes utilizados

Texto

## **1.1 – Títulos**

Texto

# VI. Tests unitarios

Texto

## **1.1 – Títulos**

Texto

# VI. Manual de usuario

Texto

## **1.1 – Títulos**

Texto

# VII. Observaciones finales

Texto

## **1.1 – Títulos**

Texto