Grabber

Manual técnico

Índice

Introducción

Grabber es una aplicación sencilla que permite a los usuarios descargar, administrar y ver vídeos de distintas plataformas.
El usuario puede crear distintas bibliotecas y sub-bibliotecas, y tener vídeos en cada una de estas, pudiendo categorizar cada una de sus descargas y saber de forma instintiva dónde tiene sus vídeos.

Análisis del problema

Problemática

La necesidad de una aplicación que permita guardar vídeos que encuentre interesantes o divertidos, permitiendo poder compartirlos con otros directamente sin enviarlos a la plataforma, o poder volver a verlos sin tener que entrar a la plataforma o tener conexión a Internet.

Clientes potenciales

Cualquier usuario que esté en una red social más restrictiva en cuanto a descargas de vídeos o que quiera guardar el vídeo y no recurrir a cualquier sistema de historial de la aplicación, si es que esta posee alguna en primer lugar.

Análisis DAFO

Debilidades

- Aplicación desconocida.
- Se necesitan dependencias externas a Java.
- Funcionalidades reducidas con respecto a competidores.

Amenazas

- Colaboraciones maliciosas.
- Posible futura brecha de términos de uso de páginas externas.
- Otras aplicaciones implementado un sistema parecido.

Fortalezas

- Simple de usar, todo se muestra con claridad.
- Navegación más rápida que un explorador de archivos convencional.
- Reproducción directa de vídeos, sin tener que definir un reproductor específico.
- Se puede eliminar la aplicación, que siempre que no se borren los archivos generados, el usuario podrá volver a instalarla y usarla con facilidad.

Oportunidades

- Guardar la base de datos online, sin necesidad de guardar los vídeos en sí.
- Ofrecer funcionalidades extra en local.

Monetización y beneficios

No hay ningún sistema de monetización implementado, y no se plantea implementar ninguno. Debido a la naturaleza del proyecto, se podrían pedir donaciones a una cuenta de forma voluntaria o una suscripción a una plataforma como Patreon para aquellos que deseen conocer de avances o cómo va el proyecto de forma adelantada.

Una posible monetización sería el poder subir la base de datos a una nube, permitiendo al usuario pagar por tener sus datos disponibles a través de una cuenta en cualquier equipo.

Diseño de la solución

Tecnologías elegidas

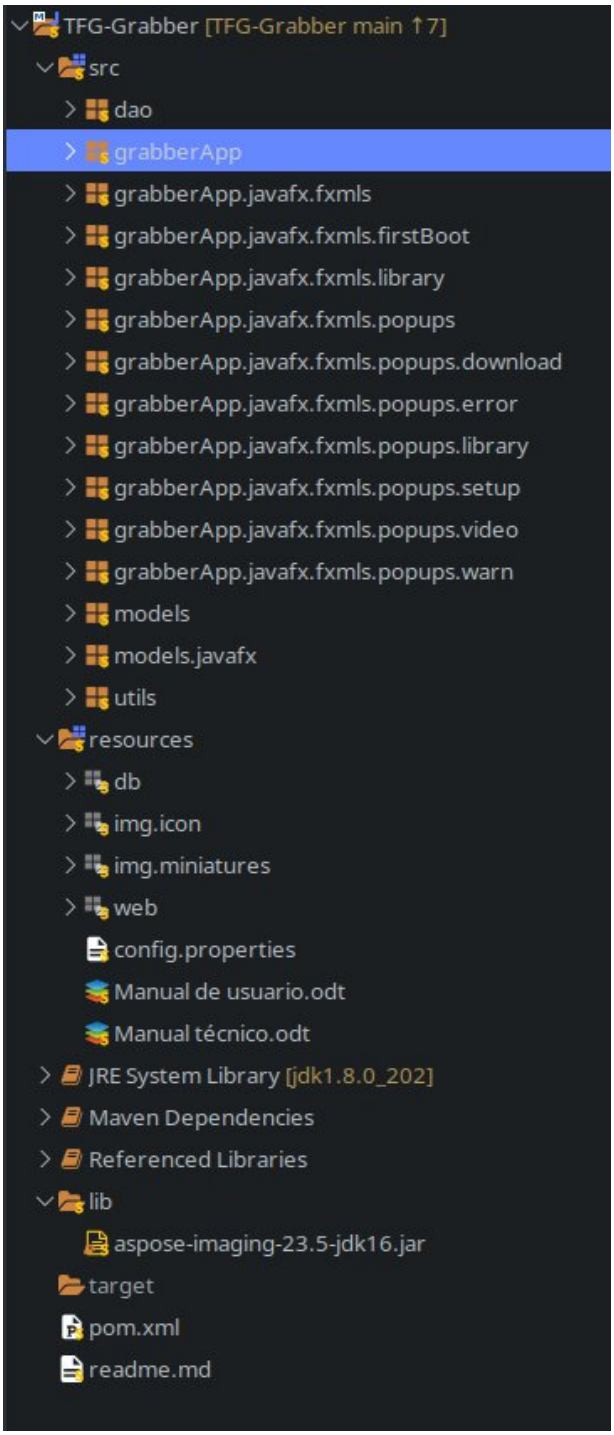
- Lenguaje: Java
- IDE: Eclipse
- Frameworks: Selenium
- Sistema de información: SQLite

Arquitectura

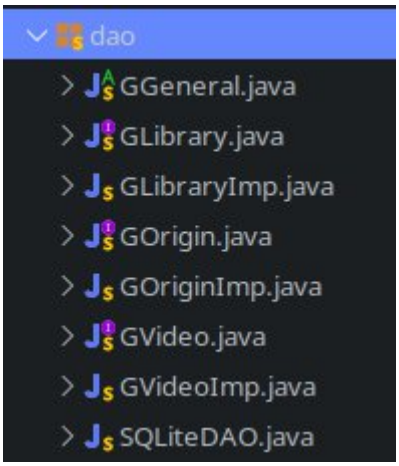
Cliente – Servidor

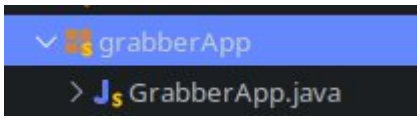
Diagrama de clases

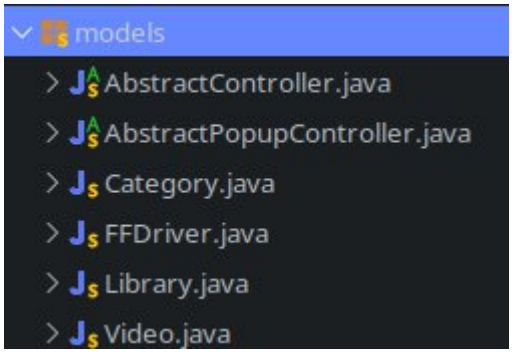
Las clases están organizadas en los siguientes paquetes.

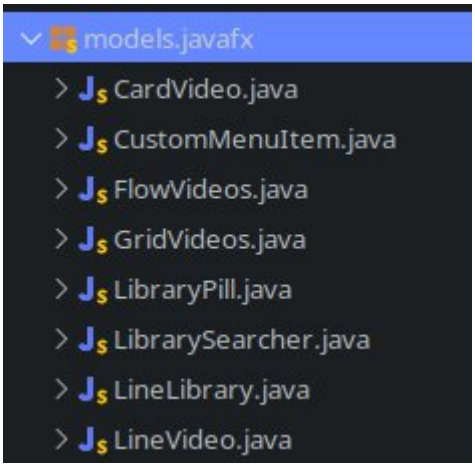


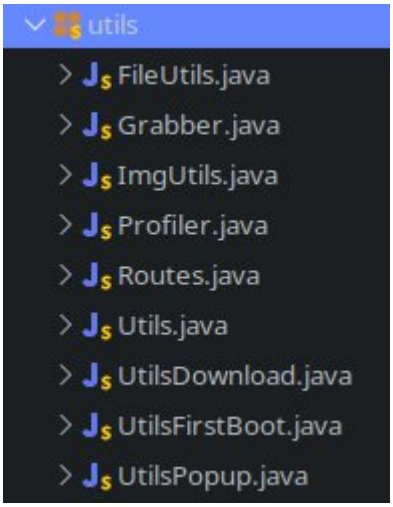
En la carpeta “src” nos encontramos con:

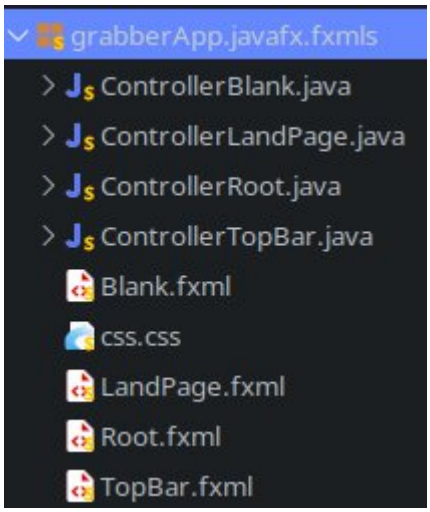
- dao
  - Se encuentran las clases necesarias para conectarse a la base de datos y ejecutar cambios.
  - 

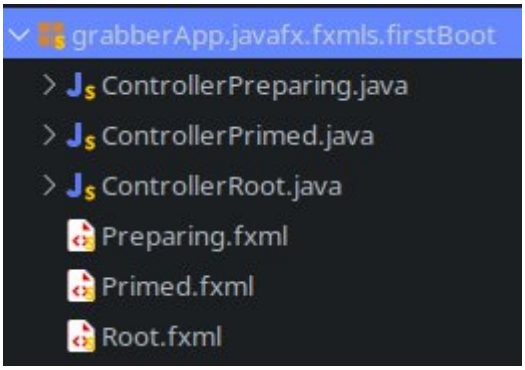
- grabberApp
  - Se encuentra la aplicación principal
  - 

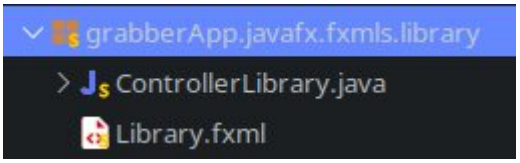
- models
  - Se encuentran los modelos utilizados en la aplicación
  - 

- models.javafx
  - Modelos específicos que extienden de componentes de JavaFX
  - 

- utils
  - Se encuentran clases que tienen utilidades
    

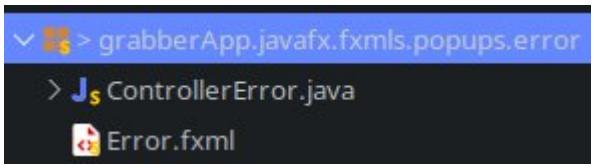
- grabberApp.javafx.fxmls (gjf)
  - La raíz de las vistas del programa
    

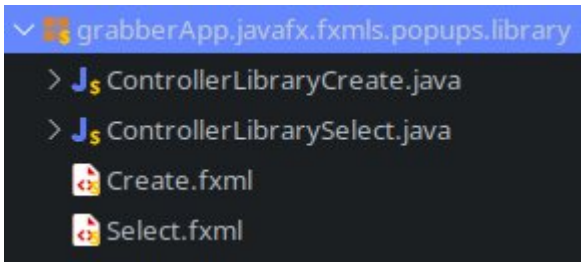
- gjf.firstBoot
  - Pantallas utilizadas cuando el usuario utiliza por primera vez el programa. Tómese como primera vez que el usuario no haya ejecutado nunca la aplicación, que el usuario haya ejecutado la aplicación pero que no haya configurado una carpeta raíz, o que un usuario haya eliminado/renombrado la carpeta raíz.
    

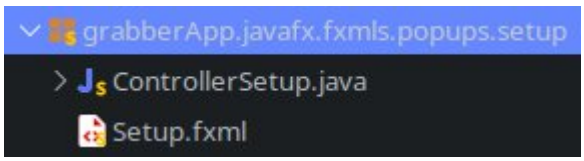
- gjf.library
  - Pantalla utilizada para mostrar las bibliotecas
  - 

- gjf.popups (gjfp)
  - Pantalla raíz para los pop-ups que puedan aparecer
  - 

- gjfp.download
  - Pantalla que muestra el sistema de descarga
  - 

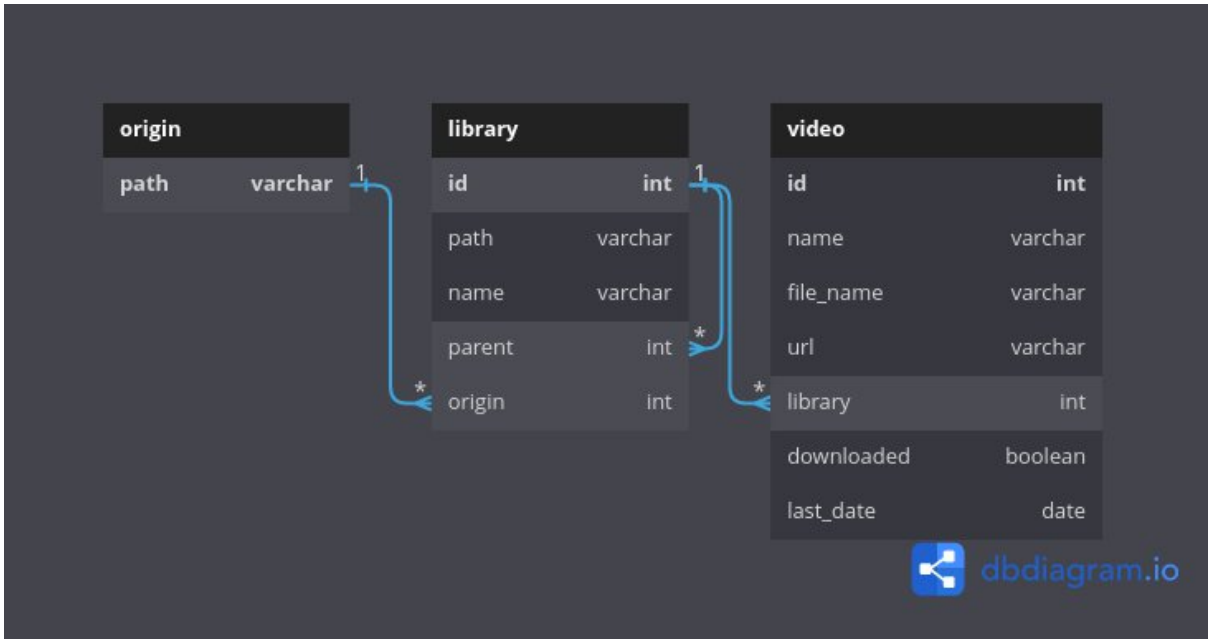
- gjfp.error
  - Pantalla que muestra el error, dependiendo del tipo de error que sea, mostrará un mensaje u otro, con unas opciones u otras.
  - 

- gjfp.library
  - Pantallas relacionadas con las bibliotecas:
    - Crear biblioteca.
    - Seleccionar biblioteca.
  - 

- gjfp.setup
  - Pantalla que sólo aparece para configurar una carpeta raíz
  - 

- gjfp.warn
    - Pantalla que nos advierte de que nuestro vídeo ya existe, y si queremos moverlo a esa biblioteca
    - 

Diagrama E/R



Diagrama

Consideraciones técnicas

Para poder desarrollar en la aplicación hay que cumplir con los siguientes requerimientos.

- Tener instalado Eclipse para Java 2022-12 o anterior. La 2022-09 es la que se ha utilizado en este proyecto.
- Tener instalada la extensión de Eclipse e(fx)clipse. En el momento del proyecto, esta extensión no es compatible con versiones de Eclipse superiores a la 2022-12.
- Tener instalado SceneBuilder.
- Tener instalado un programa de base de datos. Se recomienda DB Browser.
- Tener descargado el JDK 1.8 de Java.
- Tener descargado un SKD de JavaFX. La versión de JavaFX del programa es la 19.

Nota técnica: se puede programar el proyecto con una versión de Java reciente. A través de Maven, se pueden importar las dependencias necesarias para utilizar JavaFX sin estar limitados a Java 8.

Se necesita VLC para poder reproducir los vídeos, saltando un error si hay algún problema en la ejecución.

Primero, ejecutamos Eclipse. Nos pedirá que seleccionemos una carpeta como entorno de trabajo. Ahora, para obtener el proyecto podemos:

- Descargar el proyecto y descomprimirlo.
- Clonar el proyecto. Para esto, es necesario Git.

Ahora que tenemos el proyecto, configuramos Eclipse para poder trabajar en el proyecto.

- Instalamos la extensión e(fx)clipse. Para esto, hacemos:
  - En el menú horizontal superior, nos dirigimos a "Ayuda/Help" y hacemos clic en "Eclipse Marketplace".
  - Buscamos "javafx".
  - Cuando encontremos la extensión, le damos a "Install", y en el proceso de instalación, aceptamos las licencias y esperamos a que Eclipse nos diga que tiene que reiniciarse.
- Una vez que la instalamos, configuramos la extensión:
  - En el menú horizontal superior, nos dirigimos a "Ventana/Window" y hacemos clic en "Preferencias/Preferences".
  - Nos movemos al menú "JavaFX", e indicamos dónde está el ejecutable de SceneBuilder y el SDK de JavaFX.
- Ahora que tenemos la extensión configurada, añadimos el JDK a los JRE de Eclipse. En el propio menú de preferencias:
  - Expandimos árbol de "Java" y nos movemos a "Installed JREs".
  - Hacemos clic en "Add...".
  - Seleccionamos "Standard VM".
  - En "JRE home", indicamos la carpeta del JDK 1.8.
  - Le damos a "Finalizar/Finish".
- Continuamos importando el proyecto. Para ello:
  - En el menú horizontal superior, nos dirigimos a "Archivo/File" y hacemos clic en "Import...".
  - Seleccionamos "File System".
  - En "From directory", seleccionamos la carpeta del proyecto y le damos a "Finish".

Una vez importado el proyecto, is nos da error, tendremos que cambiar el JDK con el que se ejecuta. Para ello:

- En el explorador de la izquierda, hacemos clic derecho y le damos a "Properties".
- Nos vamos a "Java Build Path".
- Seleccionamos la opción de JRE y seleccionamos "Edit...".
- En "Alternate JRE", seleccionamos el JDK 1.8 y pulsamos "Finish".
- Nos movemos a "Java Compiler".
- Activamos "Enable project specific settings" y cambiamos "Compiler compliance level" a 1.8.

Con esto, ya podríamos empezar a desarrollar.

Consideraciones:

- La base de datos está incluida en el proyecto.
- La ejecución del programa crea una serie de archivos en la carpeta de configuraciones del usuario. Entre ellos, una copia de la base de datos limpia, por lo que si se hace algún cambio a la base de datos en el proyecto, se ha de eliminar este archivo, que se encuentra en:
  - Windows: C:\Users\[tu_usuario]\AppData\Local\JGrabber
  - Linux: /home/[tu_usuario]/.config/jgrabber
- Recuerda siempre refrescar el proyecto una vez hecho cambios en los FXML o en la base de datos, o tras haber añadido archivos desde fuera de Eclipse.

Documentación de la solución

El código se puede encontrar en GitHub, siendo este el repositorio.
Es un programa de código abierto, por lo que cualquiera puede tener acceso a este, analizar el código y contribuir.

Posibles mejoras

- Para que la aplicación reproduzca vídeos a través de JavaFX, hacen falta una serie de CODEX específicos en la máquina, lo que limita su implementación. Actualmente, la aplicación reproduce vídeos a través de VLC a fuerza. Existe VLCJ, un proyecto que permite utilizar VLC dentro de una ventana de JavaFX o Swing. Es necesario actualizar la versión del JDK a, como mínimo, la 11.
- Mejor sistema de descargas. Actualmente, si el hilo donde se está descargando un vídeo no pertenece a JavaFX, no se pueden modificar componentes de JavaFX.

- Mejor sistema de historial de descargas. Actualmente, existe un bug que no permite ver un vídeo en el historial, ya que ese vídeo es una instancia única y no se modifica una vez insertado.
- Búsqueda global en la pantalla de inicio.
- Búsqueda de vídeo en cada biblioteca.
- Permitir importar vídeos.
- Obtener miniaturas del vídeo directamente.
- Implementación de servidor completo.
    - Se puede crear un backend puro, que se comunique con la base de datos y se descargue los vídeos.
    - Se puede crear un frontend puro, que hable al servidor y utilice los datos recibidos. Permite crear aplicaciones en otros entornos, como por ejemplo una aplicación móvil en Flutter.

Problemas conocidos

- JavaFX no comienza el proceso de forma síncrona en Windows, obligando al usuario a cerrar la aplicación para poder ejecutar el vídeo. Arreglos posibles:
    - Encontrar una forma de usar un reproductor nativo.
    - Incrustar VLC en JavaFX usando VLCJ.
- Las descargas se han de hacer en primer plano si se quieren evitar problemas. Arreglos posibles:
    - Insertar el vídeo en cuanto se acepta la petición, y cambiar el manejo de descargas para evitar posibles problemas.
    - Realizar la descarga en segundo plano, ejecutando el sistema bajo un hilo de aplicación JavaFX.

Enlaces de interés

Documentación

Oficiales

- Java
- JavaFX: OpenFX, FXDocs, Oracle

Recursos

Imágenes

- Iconos: Material Icons
- Icono de la aplicación: Papirus + Material

Extensiones Maven

- Selenium
- WebDriver Manager
- SQLite JDBC

Páginas externas

Para implementar la descarga de vídeos en el proyecto, se hace uso de la página SaveTheVideo. Esta página viene con sus limitaciones, y se plantea cambiar la automatización de Selenium con esta página por el programa youtube-dl, con la posibilidad de introducir yt-dlp, un fork con mejores funcionalidades.