



SERIELIZABLE

Índice

Introducción	2
Propósito y explicación.....	2
Análisis del problema.....	3
Problemática.....	3
Público objetivo.....	3
Análisis DAFO	4
Monetización y beneficios	4
Diseño de la solución	5
Tecnologías utilizadas	5
Arquitectura	5
Modelo de datos	6
Diagrama de clases	7
Consideraciones técnicas	8
Posibles mejoras	8
Bibliografía.....	9

Introducción

Propósito y explicación

El propósito de la aplicación es poder buscar, gestionar y persistir tanto películas como series de manera independiente así como información relativa a ellas.

El usuario podrá registrarse, buscar y añadir tanto series como películas y consultarlas cuando desee, editar la información de las mismas, filtrarlas, ordenarlas, e incluso tener acceso a una ventana estadísticas propias del usuario, tanto de sus películas como de sus series.

Análisis del problema

Problemática

La aplicación pretende dar la posibilidad al usuario para tener un sitio donde mantener un registro de sus películas y/o series preferidas, disponiendo de información de las mismas, además de poder editar su estado, añadir una reseña o una valoración personal sobre ellas.

El usuario podrá filtrar por las series y películas de su lista, y añadirlas cómodamente desde el apartado de buscar.

También dispone de un apartado de estadísticas propias del usuario que podrá consultar cuando desee, y que se actualiza en función de las propias listas del usuario.

Público objetivo

- La aplicación está pensada para gente que poseen la necesidad de mantener un registro sobre todo lo que han visto, están viendo o incluso planean ver en un futuro
- También está orientada a personas que simplemente quieran acceso rápido a la información más relevante de una serie o película
- Para personas despistadas que no recuerdan por qué capítulo o temporada iban de la serie que se estaban viendo

Análisis DAFO

DEBILIDADES <ul style="list-style-type: none"> - La base de datos es local. - Bastante dependiente de la conexión a internet. 	AMENAZAS <ul style="list-style-type: none"> - Dependiente de la API de themoviedb. - Existencia de muchas aplicaciones similares.
FORTALEZAS <ul style="list-style-type: none"> - Intuitiva y fácil de usar. - Posibilidad de registrar varios usuarios. - Estadísticas útiles sobre el usuario. 	OPORTUNIDADES <ul style="list-style-type: none"> - Dirigida a todo tipo de público. - Posibilidad de subir la base de datos a un servidor remoto.

Monetización y beneficios

La aplicación pretende ser un software de libre uso accesible para cualquiera.

Aunque la idea original no fuera obtener beneficios, añadir anuncios no invasivos o crear un apartado de donaciones podría funcionar como alternativas para monetizar la aplicación.

Diseño de la solución

Tecnologías utilizadas

- Lenguaje de programación: *Java 8*
- Base de datos: *MySQL*
- IDE: *Eclipse IDE, MySQL Workbench*
- Framework: *JavaFX, Hibernate*
- Gestor de dependencias: *Maven*
- Diseño: *JavaFX (Scene builder), CSS*

Arquitectura

Patrón [modelo-vista-controlador \(MVC\)](#).

Modelo de datos

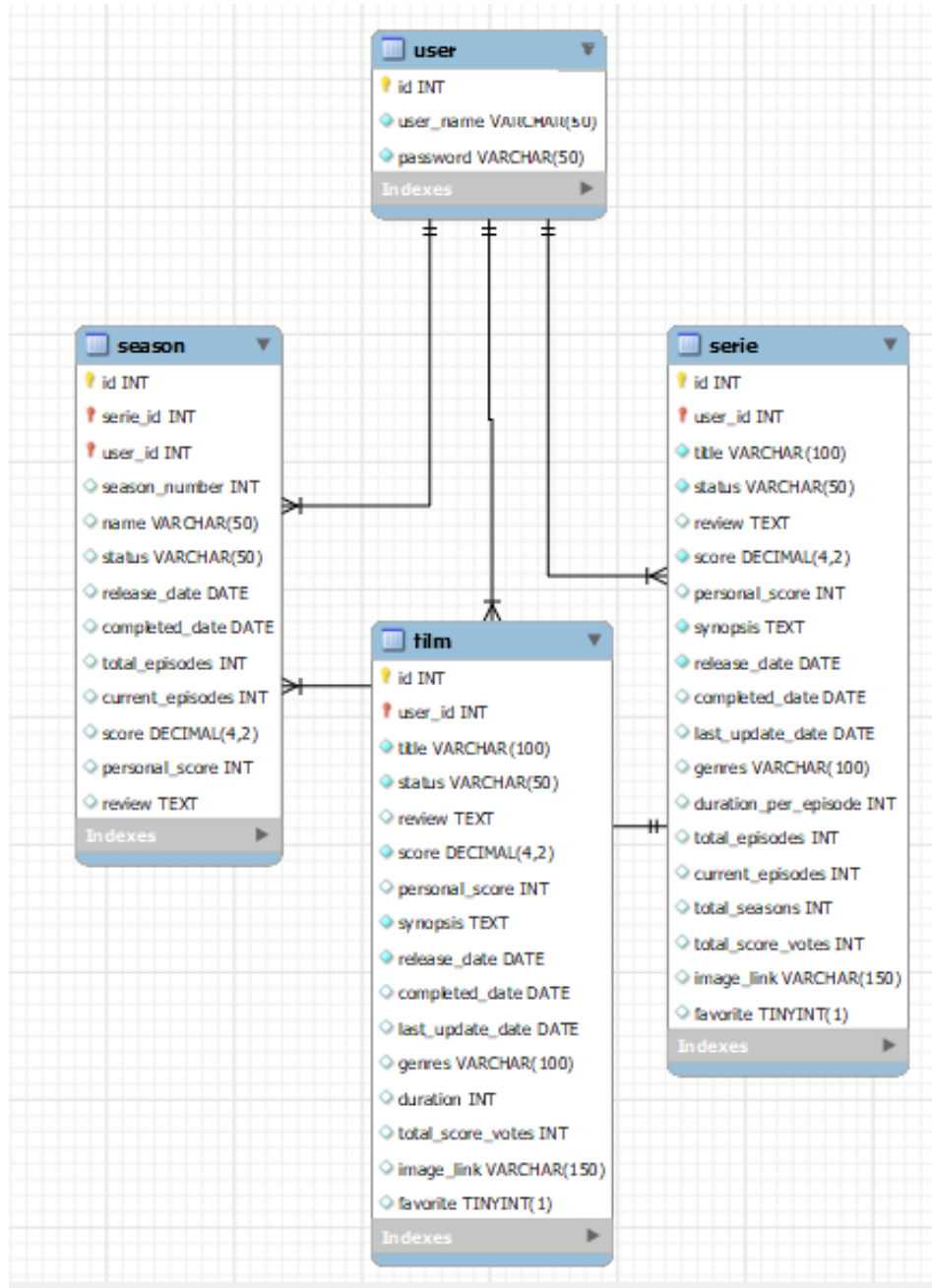
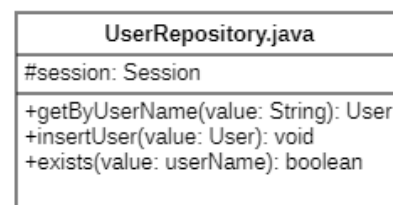
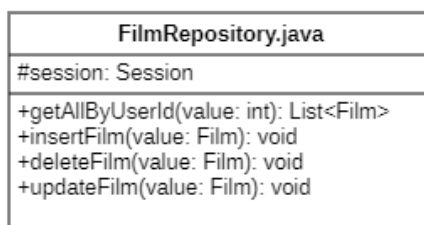
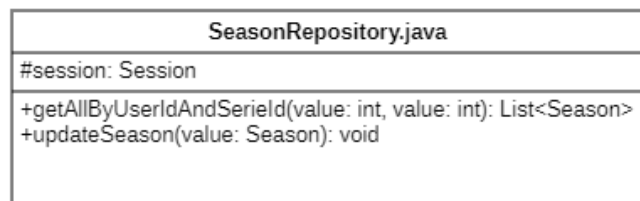
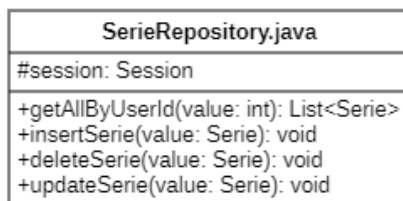
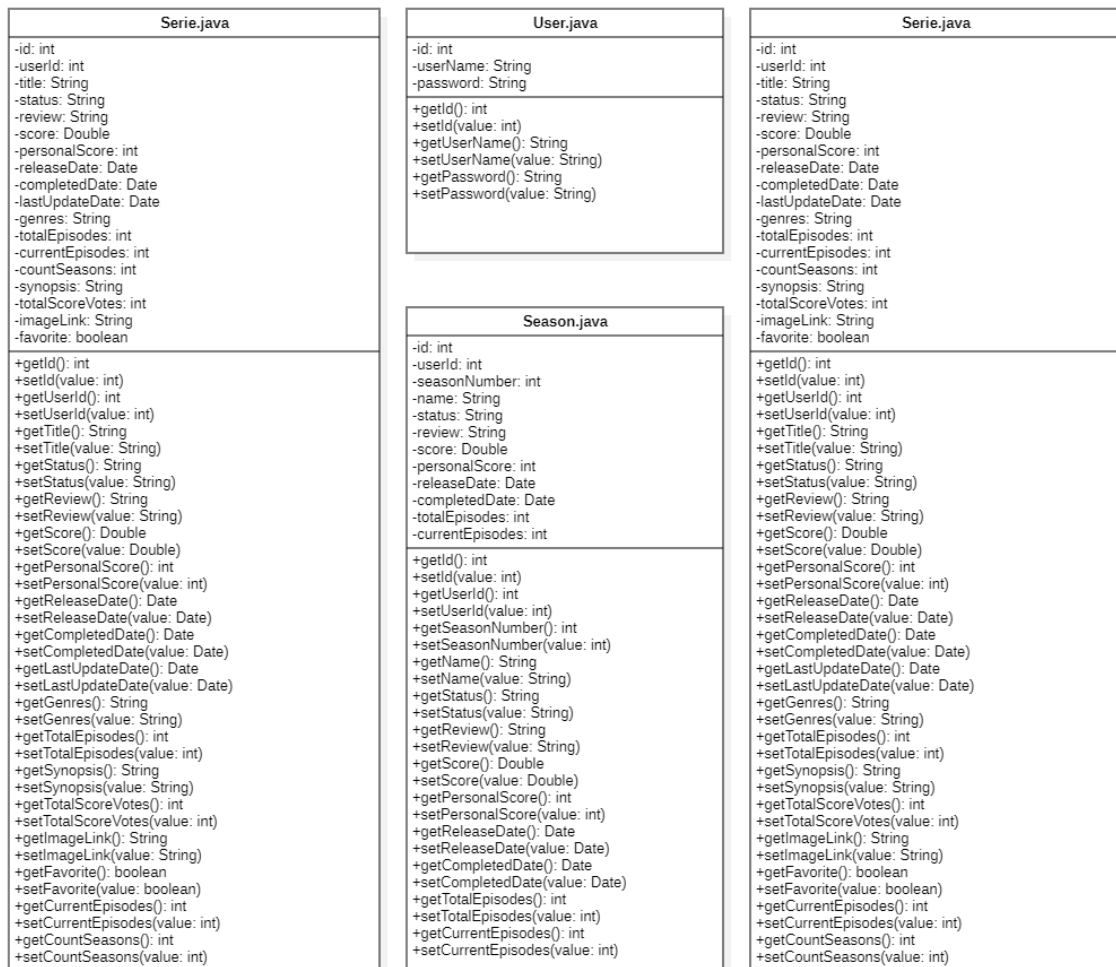


Diagrama de clases



Consideraciones técnicas

La base de datos fue inicialmente diseñada y planteada para su uso en MongoDB, pero por cuestiones de falta de conocimiento y tiempo no ha sido finalmente abordable. Esto por esto que la estructura del diagrama E/R puede no ser la esperada de un modelo relacional como lo es MySQL.

Posibles mejoras

- Subir la base de datos a un servidor remoto y realizarla en MongoDB.
- Añadir posibilidad de interacción con otros usuarios; poder agregar amigos, ver sus listas, etc.
- Añadir un apartado de películas o series populares.
- Añadir perfil y personalización de usuario (esto solo tendría sentido si paralelamente se añade la interacción con otros usuarios).

Bibliografía

Haga clic [aquí](#) para acceder al repositorio en github.

Adjunto algunos de los enlaces que he usado como referencia:

<https://docs.oracle.com/javafx/2/charts/pie-chart.htm>

https://www.tutorialspoint.com/javafx/pie_chart.htm

<https://docs.oracle.com/javafx/2/charts/bar-chart.htm>

https://www.tutorialspoint.com/javafx/bar_chart.htm

https://docs.oracle.com/javafx/2/ui_controls/table-view.htm

https://docs.oracle.com/javafx/2/ui_controls/list-view.htm

<https://www.themoviedb.org/>

<https://developer.themoviedb.org/docs/search-and-query-for-details>

<https://developer.themoviedb.org/reference/tv-series-details>

<https://www.themoviedb.org/talk/644a4b69f794ad04fe3cf1b9>

https://docs.jboss.org/hibernate/annotations/3.5/reference/en/html_single/

<https://es.stackoverflow.com/questions/111168/como-mostrar-un-valor-de-un-combobox-en-javafx-pasando-un-valor>

<https://stackoverflow.com/questions/51452785/java-fx-changing-imageview-image-or-image-url>

<https://coderanch.com/t/682889/java/read-image-URL>

<http://www.java2s.com/Code/Java/JavaFX/LoadImagefromURL.htm>

<https://stackoverflow.com/questions/19343531/javafx-combobox-styling>

<https://www.outsystems.com/forums/discussion/24808/combo-box-arrow-color/>

<https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/MenuBar.html>