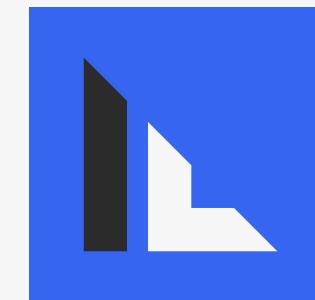


MANUAL TÉCNICO

MYGAMES

*PABLO AMORES MUÑOZ
ÁLVARO DÍAZ CASAÑO
IAN IWANCZUK*





ÍNDICE

| | | | |
|--|---------|---------------------------------------|---------|
| ■ <u>INTRODUCCIÓN</u> | 03 | ■ <u>DISEÑO DE LA SOLUCIÓN</u> | 10 - 21 |
| ■ <u>ANÁLISIS DEL PROBLEMA</u> | 04 - 07 | ■ Tecnologías elegidas | 10 |
| ■ Problemática | 04 - 06 | ■ Diagrama E/R | 11 |
| ■ Clientes potenciales | 07 | ■ Diagrama de clases | 12 - 21 |
| ■ Análisis DAFO | 08 | | |
| ■ Monetización y beneficios | 09 | | |
| ■ <u>DOCUMENTACIÓN DE LA SOLUCIÓN</u> | 22 | | |
| ■ <u>ENLACES DE INTERES</u> | 23 | | |

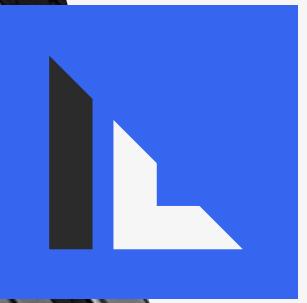
INTRODUCCIÓN



En la actualidad, la industria de los videojuegos ha crecido exponencialmente, con miles de títulos lanzados cada año en múltiples plataformas. Para los entusiastas y coleccionistas, mantener un registro organizado de los juegos jugados, pendientes o deseados puede ser un desafío. Muchos jugadores buscan herramientas que les permitan gestionar su biblioteca de manera eficiente, evitando perder de vista títulos interesantes.

Este proyecto nace con la intención de ofrecer una solución práctica y accesible a esta necesidad. La aplicación permite a los usuarios buscar títulos y agregarlos a su colección personal, ya sea para registrar los juegos que han completado o para llevar un seguimiento de aquellos que desean jugar en el futuro. De esta manera, se facilita la organización del catálogo de cada usuario, promoviendo una experiencia más estructurada y personalizada.

El software está dirigido a una amplia variedad de usuarios, desde jugadores ocasionales que desean recordar qué juegos han probado, hasta coleccionistas y críticos que necesitan un sistema confiable para gestionar su biblioteca. Gracias a su diseño intuitivo y su funcionalidad enfocada en la organización, esta aplicación se convierte en una herramienta indispensable para cualquier aficionado a los videojuegos.

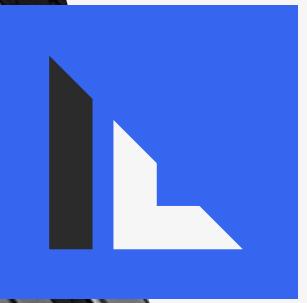


ANÁLISIS DEL PROBLEMA

PROBLEMÁTICA

Uno de los principales problemas identificados es el crecimiento acelerado de la industria de los videojuegos. Cada año se lanzan miles de títulos en diversas plataformas, lo que genera una saturación de opciones para los jugadores. Este crecimiento presenta varios desafíos:

- Dificultad para hacer un seguimiento de los juegos jugados : Con una oferta tan extensa, los jugadores pueden olvidar qué títulos han jugado, lo que podría llevar a compras o descargas repetidas.
- Problemas de gestión de juegos pendientes : Muchos usuarios tienen una lista extensa de juegos que desean jugar en el futuro, pero sin una herramienta adecuada, esta lista puede volverse inmanejable.
- Falta de visibilidad de títulos interesantes : En un mercado abarrotado de opciones, los jugadores pueden perder de vista juegos de alta calidad debido a la sobresaturación de lanzamientos.

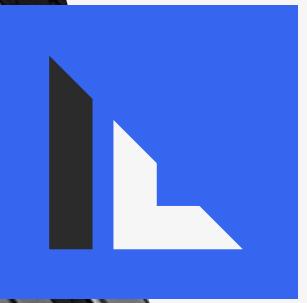


ANÁLISIS DEL PROBLEMA

PROBLEMÁTICA

Los jugadores buscan herramientas para organizar su colección de videojuegos de manera eficiente. Sin embargo, enfrentamos varios obstáculos al intentar hacerlo:

- Ausencia de herramientas integradas en las plataformas de juegos : Si bien algunas plataformas como Steam, PlayStation o Xbox ofrecen listas de juegos comprados, estas no siempre incluyen opciones avanzadas de organización ni permiten agregar títulos de otras plataformas.
- Dependencia de soluciones externas : Los usuarios a menudo recurren a hojas de cálculo o aplicaciones de notas para llevar un registro, lo cual no es una solución óptima ni escalable.
- Falta de personalización en las herramientas existentes : Algunas aplicaciones permiten gestionar listas de juegos, pero carecen de opciones avanzadas como etiquetas personalizadas, filtros o categorización detallada.

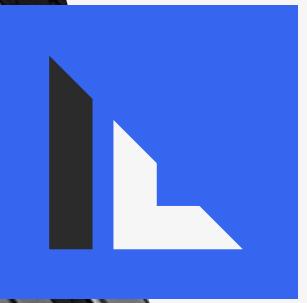


ANÁLISIS DEL PROBLEMA

PROBLEMÁTICA

Otro problema clave es la necesidad de que los jugadores registren los juegos que han completado y los que desean jugar en el futuro. Sin una herramienta adecuada, surgen diversas complicaciones:

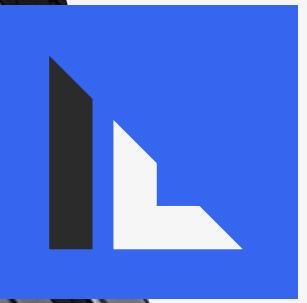
- Pérdida de información con el tiempo : Los jugadores pueden olvidar qué juegos han completado, lo que afecta la experiencia de seguimiento.
- Dificultad para establecer prioridades : Sin una lista organizada, puede ser complicado decidir qué juego jugar a continuación.
- Falta de motivación para terminar juegos : Una herramienta que registrar avances podría incentivar a los jugadores a completar títulos pendientes.



ANÁLISIS DEL PROBLEMA

CLIENTES POTENCIALES

- **Gamers entusiastas** : Usuarios que juegan con frecuencia y buscan organizar su biblioteca de videojuegos.
- **Coleccionistas** : Personas con grandes bibliotecas que necesitan gestionar su catálogo de manera eficiente.
- **Críticos y analistas** : Creadores de contenido y periodistas de videojuegos que requieren notas y reseñas.
- **Desarrolladores** : Profesionales que analizan juegos para estudio y referencia en diseño.
- **Streamers e influencers** : Creadores de contenido que organizan y planifican sus transmisiones.
- **Comunidades de videojuegos** : Grupos que comparten y recomiendan títulos con otros usuarios.
- **Jugadores ocasionales** : Personas que desean llevar un control de los juegos que han jugado o que les interesan.



ANÁLISIS DEL PROBLEMA

ANÁLISIS DAFO

Debilidades

- Necesidad de sincronización con bases de datos externos.
- Competencia con plataformas establecidas.
- Dependencia de conexión a internet para ciertas funciones.

Amenazas

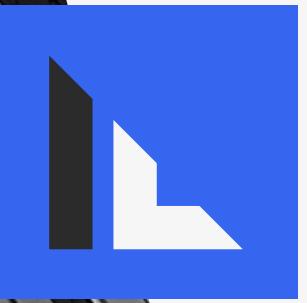
- Saturación del mercado con soluciones similares.
- Posible resistencia de los usuarios a cambiar de herramienta.
- Riesgo de cambios en políticas de plataformas externas.

Fortalezas

- Interfaz intuitiva y personalizable.
- Organización eficiente de bibliotecas de juegos.
- Compatibilidad con múltiples plataformas y dispositivos.

Oportunidades

- Crecimiento del mercado de videojuegos y colecciónismo.
- Integración con redes sociales y plataformas de streaming.
- Ampliación con funciones como análisis de tiempo de juego y recomendaciones personalizadas.



ANÁLISIS DEL PROBLEMA

MONETIZACIÓN Y BENEFICIOS

Monetización

- **Monetización Modelo Freemium :** Acceso gratuito con funciones básicas y suscripción premium con herramientas avanzadas.
- **Publicidad no intrusiva :** Anuncios en la versión gratuita sin afectar la experiencia del usuario.
- **Integración con tiendas :** Comisiones por referidas a plataformas de compra de videojuegos.
- **Personalización Premium :** Compra de temas, íconos y opciones de personalización avanzada.

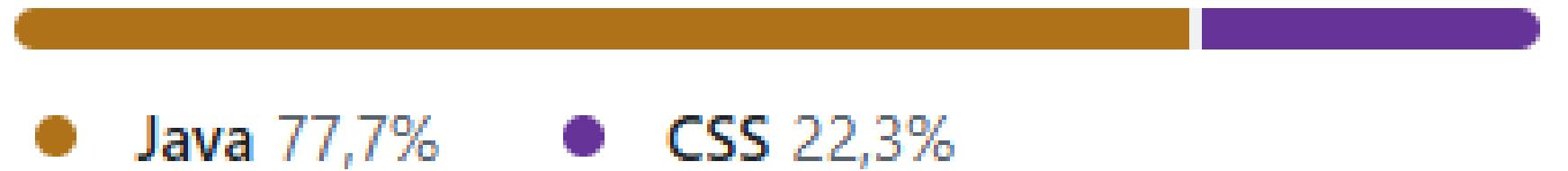
Beneficios

- **Experiencia optimizada :** Interfaz intuitiva y adaptable a cada usuario.
- **Gestión eficiente :** Organización rápida y personalizada de la biblioteca de juegos.
- **Accesibilidad multiplataforma :** Sincronización entre dispositivos para una experiencia fluida.
- **Expansión de funcionalidades :** Posibilidad de futuras integraciones con plataformas de streaming y redes sociales.

DISEÑO DE LA SOLUCIÓN

TECNOLOGÍAS ELEGIDAS

Idiomas



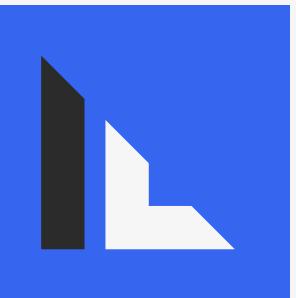
- Entorno de desarrollo: ECLIPSE IDE
- Persistencia: MySQL
- Framework: Hibernate

ARQUITECTURA

- La app se divide en principalmente src/main y src/resources que poseen los paquetes principales

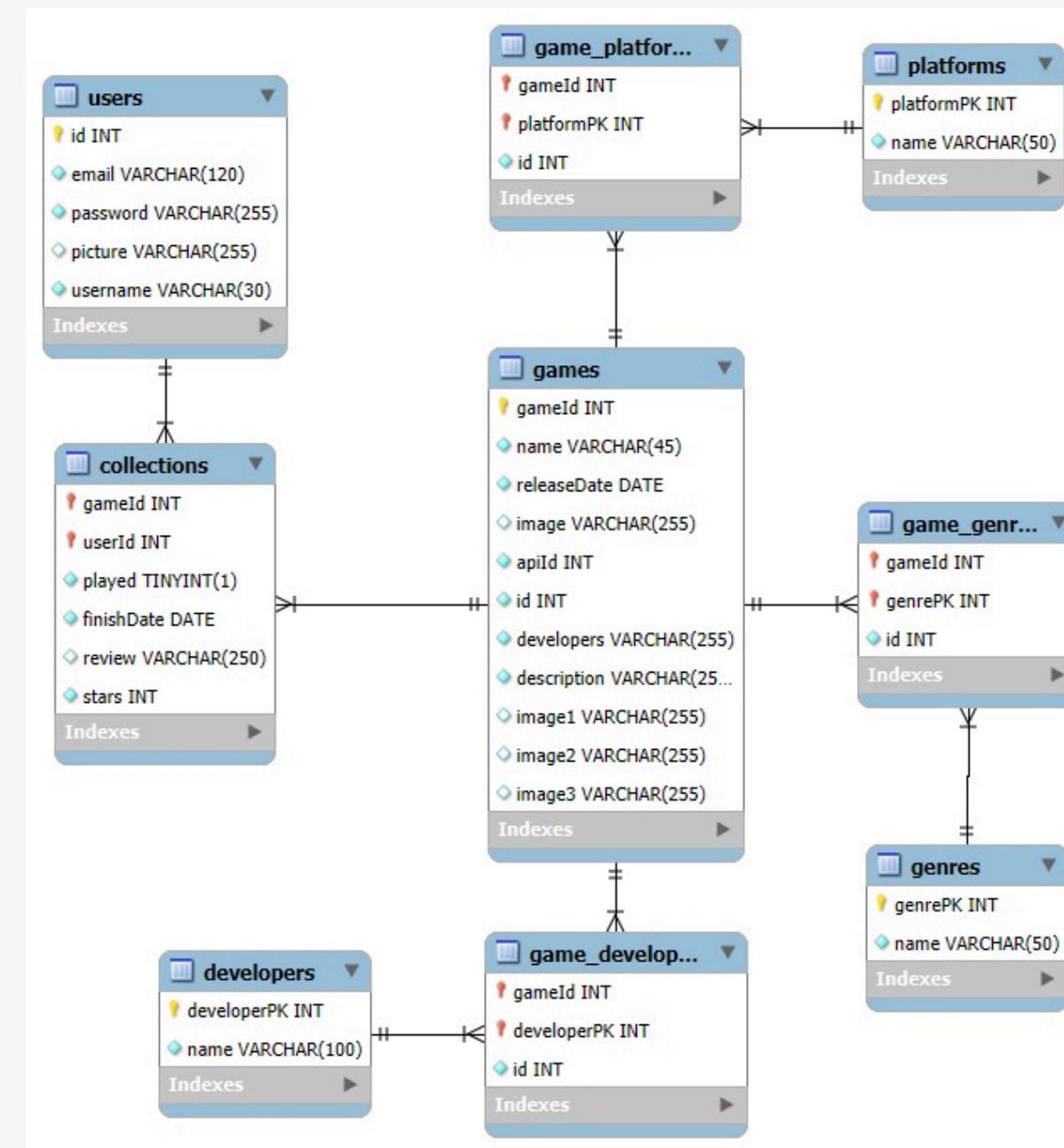
CONSIDERACIONES TÉCNICAS

- La app actualmente se podría denominar como "prototipo", ya que esta actualmente no ha salido al mercado, actualmente no tiene ni instalador, es decir la única forma de probarla es teniendo acceso al propio proyecto. Esta se puede usar tanto en Linux, Windows y MacOs para su correcto funcionamiento es necesario poseer de una conexión a internet.



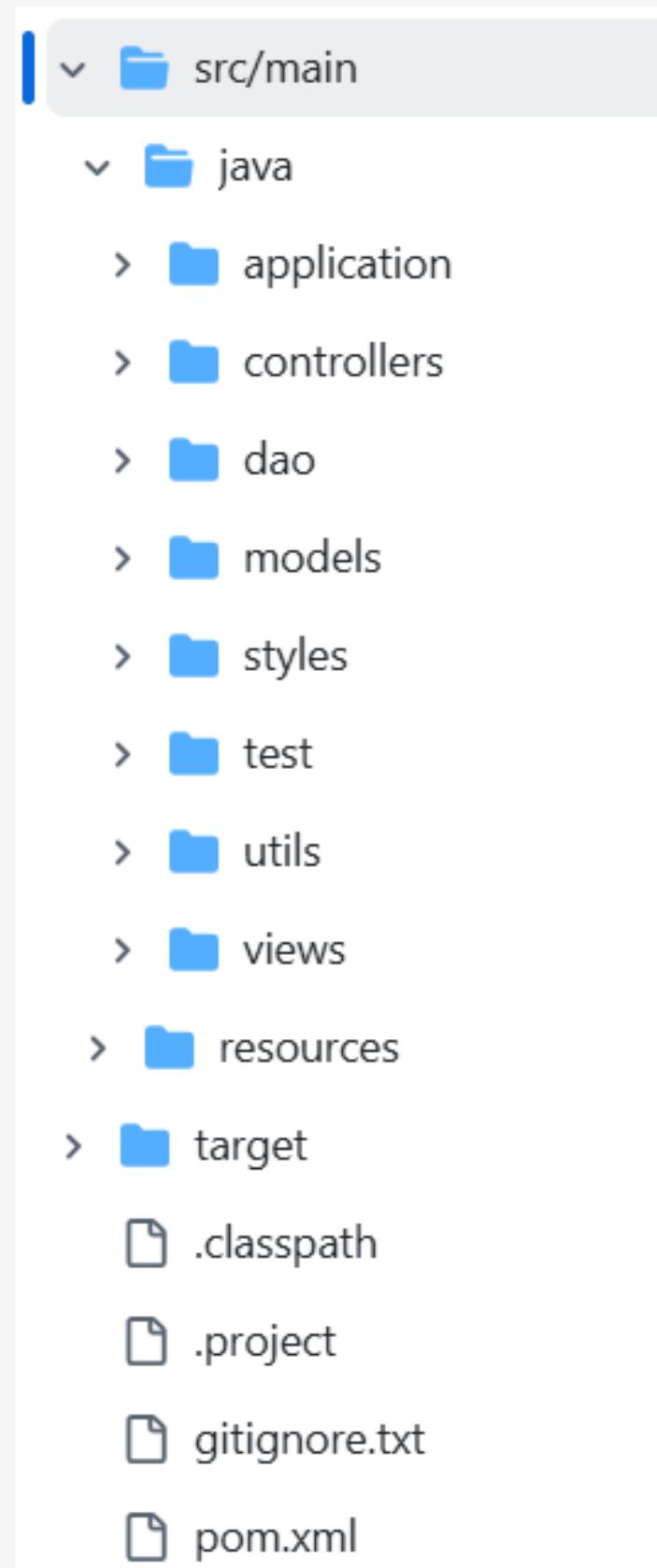
ANÁLISIS DEL PROBLEMA

DIAGRAMA E/R



ANÁLISIS DEL PROBLEMA

DIAGRAMA DE CLASES



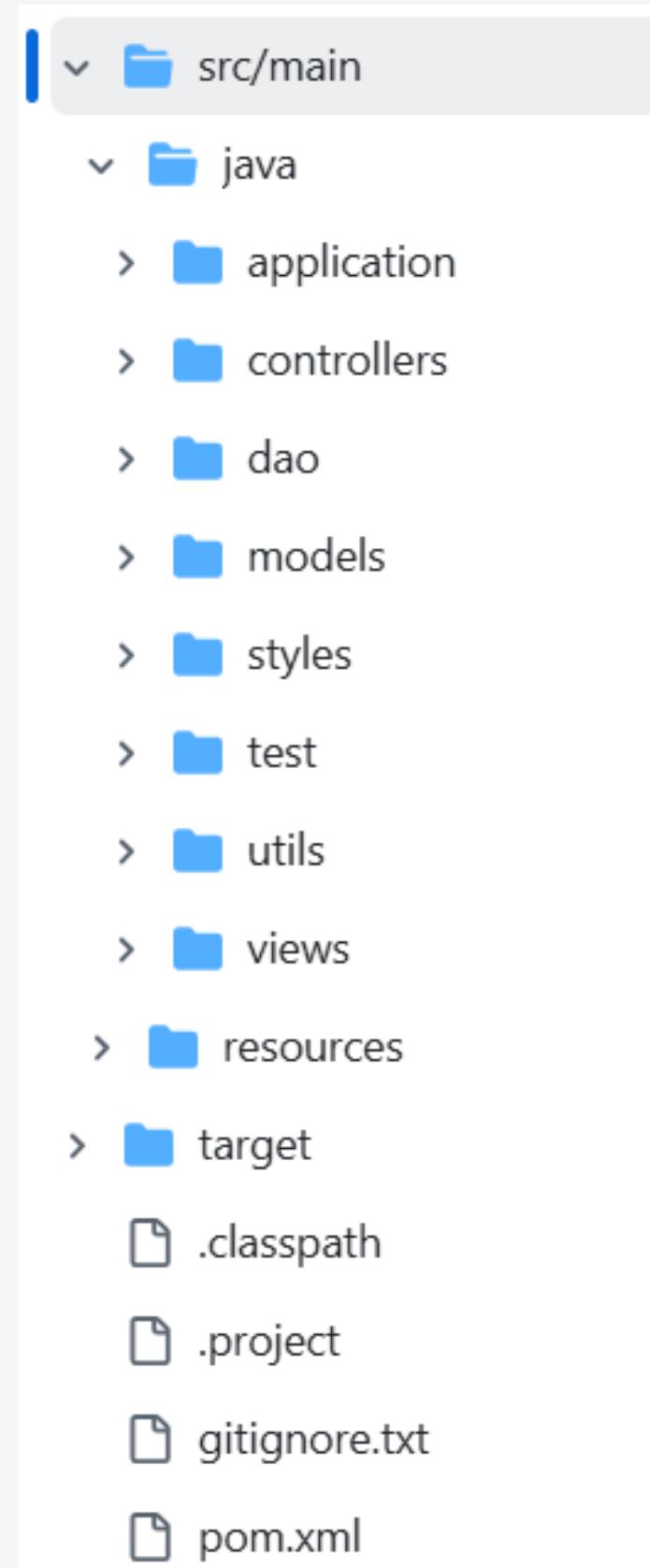
1.-src/main/java

1.1-application

- **main.java:** Contiene el main del proyecto.
- **launcher:** Contiene el launcher de la app.

ANÁLISIS DEL PROBLEMA

DIAGRAMA DE CLASES



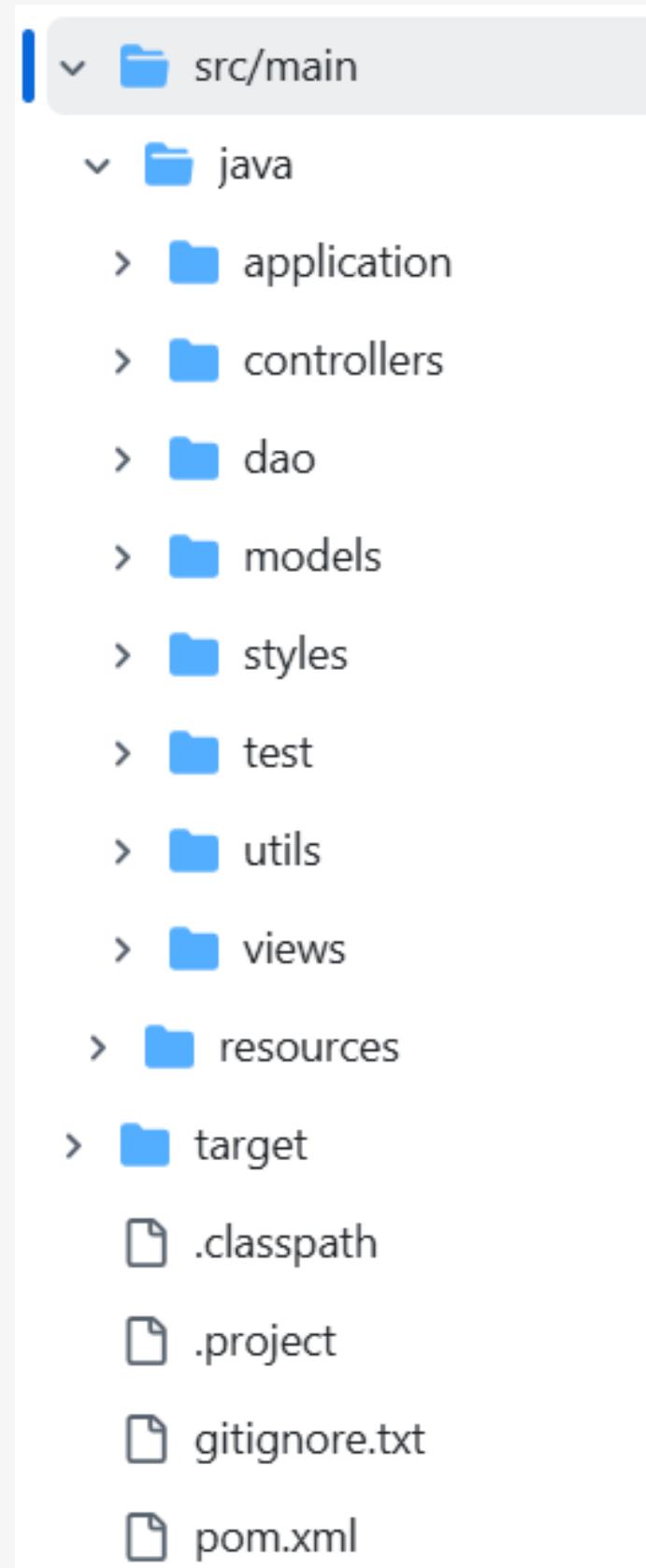
1.-src/main/java

1.2-controllers

- ColecciónGameInfoPaneController.java: Gestiona la interfaz de información de juegos en la colección.
- Controlador CollectionPane.java: Controla la vista de la colección de juegos del usuario.
- Controlador GameContainer.java: Maneja la organización y visualización de los juegos en la aplicación.
- Controlador GamesPane.java: Gestiona el panel donde se muestran todos los juegos.
- Controlador de panel de encabezado.java: Controla la barra superior de la aplicación.
- Controlador del panel de inicio.java: Maneja la vista principal de inicio.
- Controlador de inicio de sesión.java: Gestiona la lógica del formulario de login.
- Controlador de datos de inicio de sesión.java: Controla los datos y validaciones del inicio de sesión.
- Controlador del panel principal.java: Gestiona la vista principal de la aplicación.
- Controlador de datos de registro.java: Controla la entrada de datos en el formulario de registro.
- Controlador UserDataPane.java: Administra la vista para la gestión de datos del usuario.

ANÁLISIS DEL PROBLEMA

DIAGRAMA DE CLASES



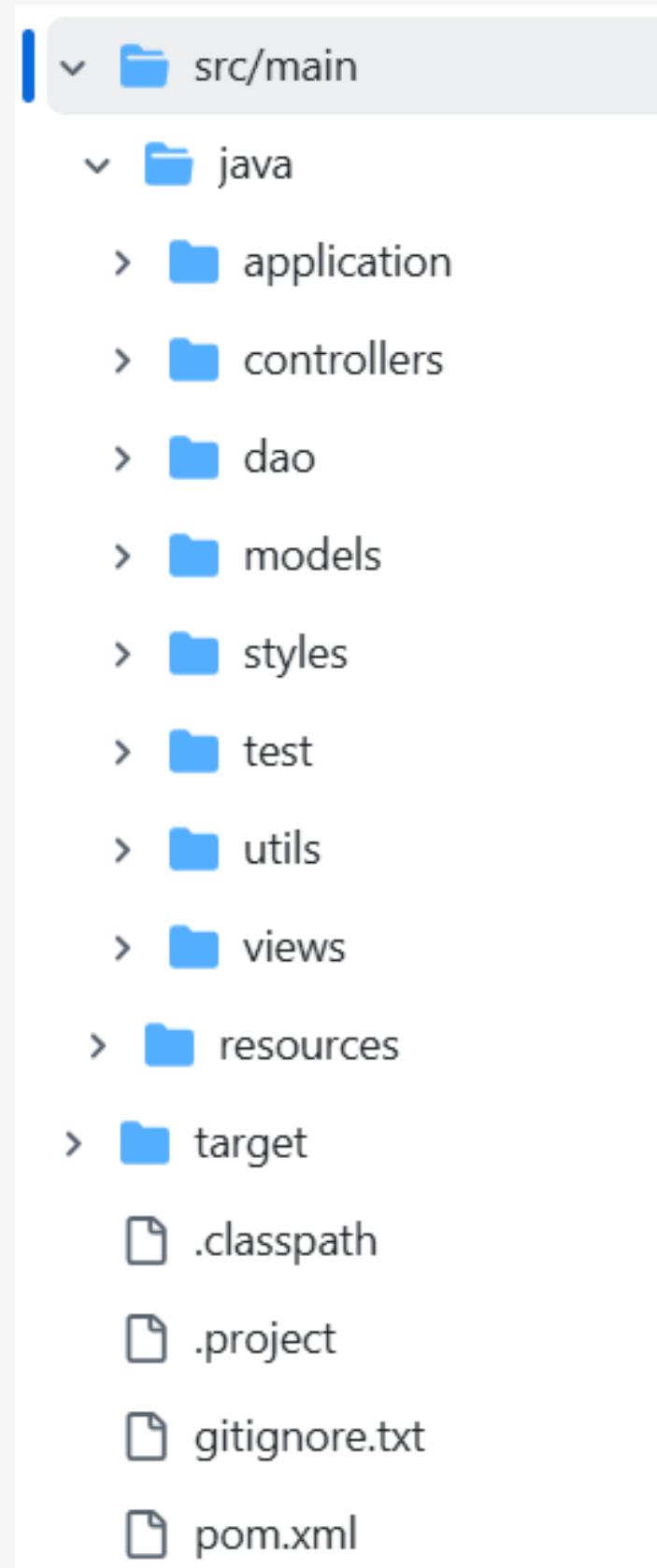
1.-src/main/java

1.3-dao

- **CollectionDaoInt.java:** Interfaz que define las operaciones de acceso a datos para colecciones.
- **CollectionDaoImpl.java:** Implementación de la interfaz CollectionDaoInt para gestionar colecciones en la base de datos.
- **CommonDaoInt.java:** Interfaz genérica que define operaciones comunes para la base de datos.
- **CommonDaoImpl.java:** Implementación de CommonDaoInt con métodos generales para acceso a datos.
- **GameDaoInt.java:** Interfaz que gestiona las operaciones relacionadas con videojuegos.
- **GameDaoImpl.java:** Implementación de GameDaoInt para acceder y manipular datos de videojuegos.
- **UserDaoInt.java:** Interfaz que define las operaciones de acceso a datos para usuarios.
- **UserDaoImpl.java:** Implementación de UserDaoInt para la gestión de usuarios en la base de datos.

ANÁLISIS DEL PROBLEMA

DIAGRAMA DE CLASES



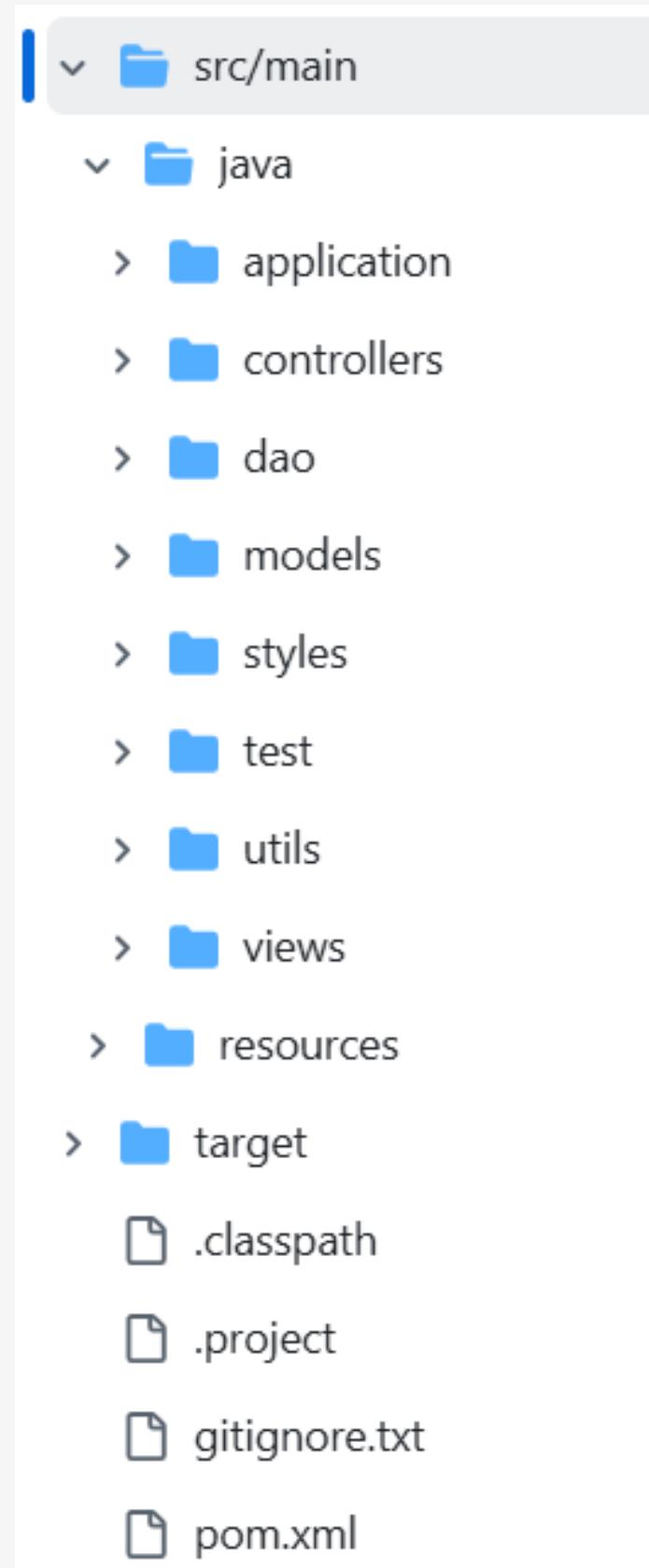
1.-src/main/java

1.4-models

- Colección.java: Representa una colección de videojuegos asociada a un usuario.
- ColecciónId.java: Define la clave primaria compuesta para la entidad de colección.
- Desarrollador.java: Modelo que almacena la información de los desarrolladores de videojuegos.
- Juego.java: Representa un videojuego con sus atributos principales (nombre, género, plataforma, etc.).
- Género.java: Modelo que define los distintos géneros de videojuegos (acción, aventura, estrategia, etc.).
- Imágenes.java: Gestiona las imágenes asociadas a videojuegos.
- Plataformas.java: Representa las plataformas en las que están disponibles los videojuegos (PC, PlayStation, Xbox, etc.).
- Usuario.java: Modelo de usuario que almacena la información del jugador registrado.

ANÁLISIS DEL PROBLEMA

DIAGRAMA DE CLASES



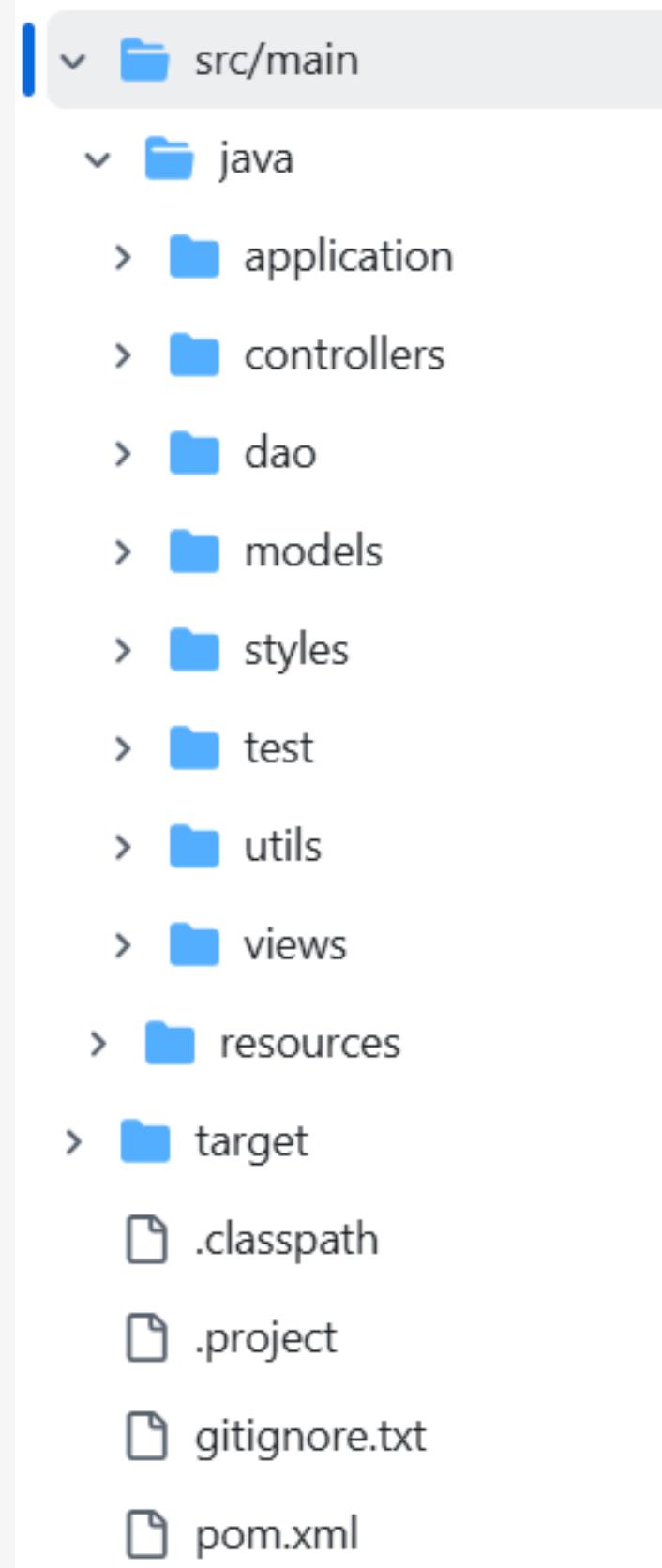
1.-src/main/java

1.5-Styles

- **CollectionGameInfoEditPane.css:** Define los estilos para la edición de información de juegos en la colección.
- **CollectionGameInfoPaneStyles.css:** Estilos para la visualización de información de juegos en la colección.
- **GamesPaneStyles.css:** Define la apariencia del panel donde se muestran todos los juegos.
- **HeaderPaneStyles.css:** Personaliza el diseño de la cabecera de la aplicación.
- **HomePaneStyles.css:** Estilos para la pantalla principal o de inicio de la aplicación.
- **LoginDataPaneStyles.css:** Define los estilos del formulario de entrada de datos en el login.
- **LoginStyles.css:** Personaliza el diseño de la pantalla de inicio de sesión.
- **MainPaneStyles.css:** Estilos generales para la vista principal de la aplicación.
- **RegisterDataPaneStyles.css:** Personaliza la apariencia del formulario de registro de usuarios.
- **UserDataPaneStyles.css:** Estilos para la pantalla de visualización y edición de datos del usuario.

ANÁLISIS DEL PROBLEMA

DIAGRAMA DE CLASES



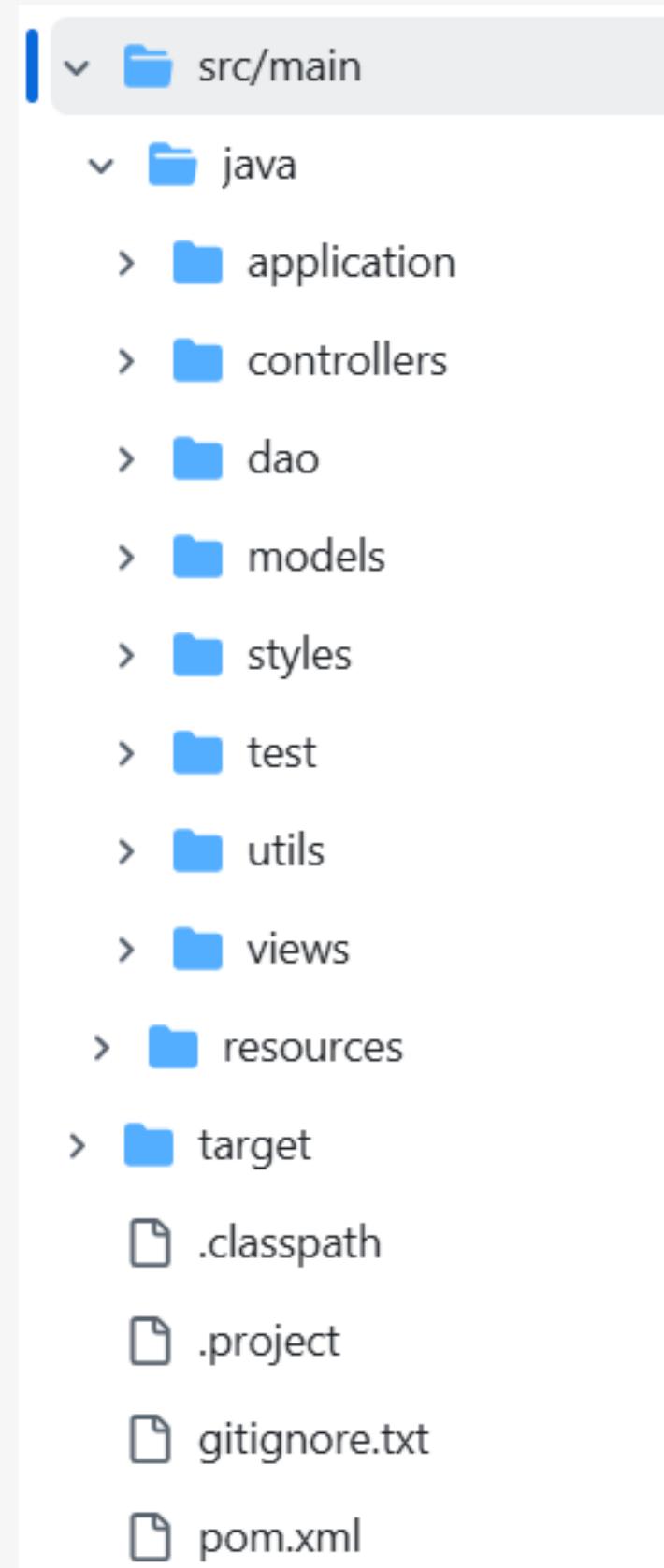
1.-src/main/java

1.6-test

- **LoginTest.java:** Contiene pruebas para validar el proceso de inicio de sesión.
- **RegisterTest.java:** Verifica el correcto funcionamiento del registro de nuevos usuarios.

ANÁLISIS DEL PROBLEMA

DIAGRAMA DE CLASES



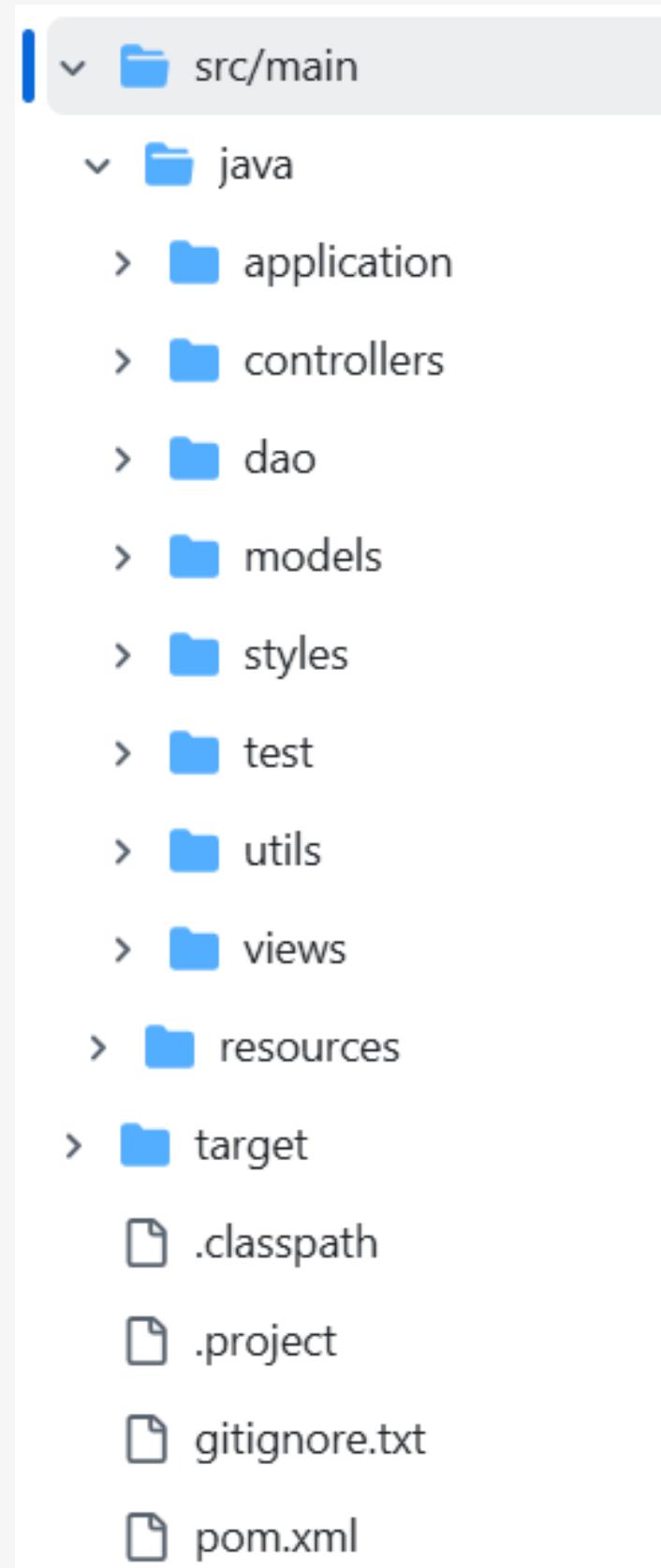
1.-src/main/java

1.7-utils

- **ApiUtils.java:** Facilita la conexión y comunicación con APIs externas.
- **ConversionUtils.java:** Contiene métodos para convertir datos entre distintos formatos.
- **HibernateUtil.java:** Gestiona la configuración y sesión de Hibernate para la base de datos.
- **PasswordUtil.java:** Proporciona funciones para el manejo seguro de contraseñas, como hash y validación.

ANÁLISIS DEL PROBLEMA

DIAGRAMA DE CLASES



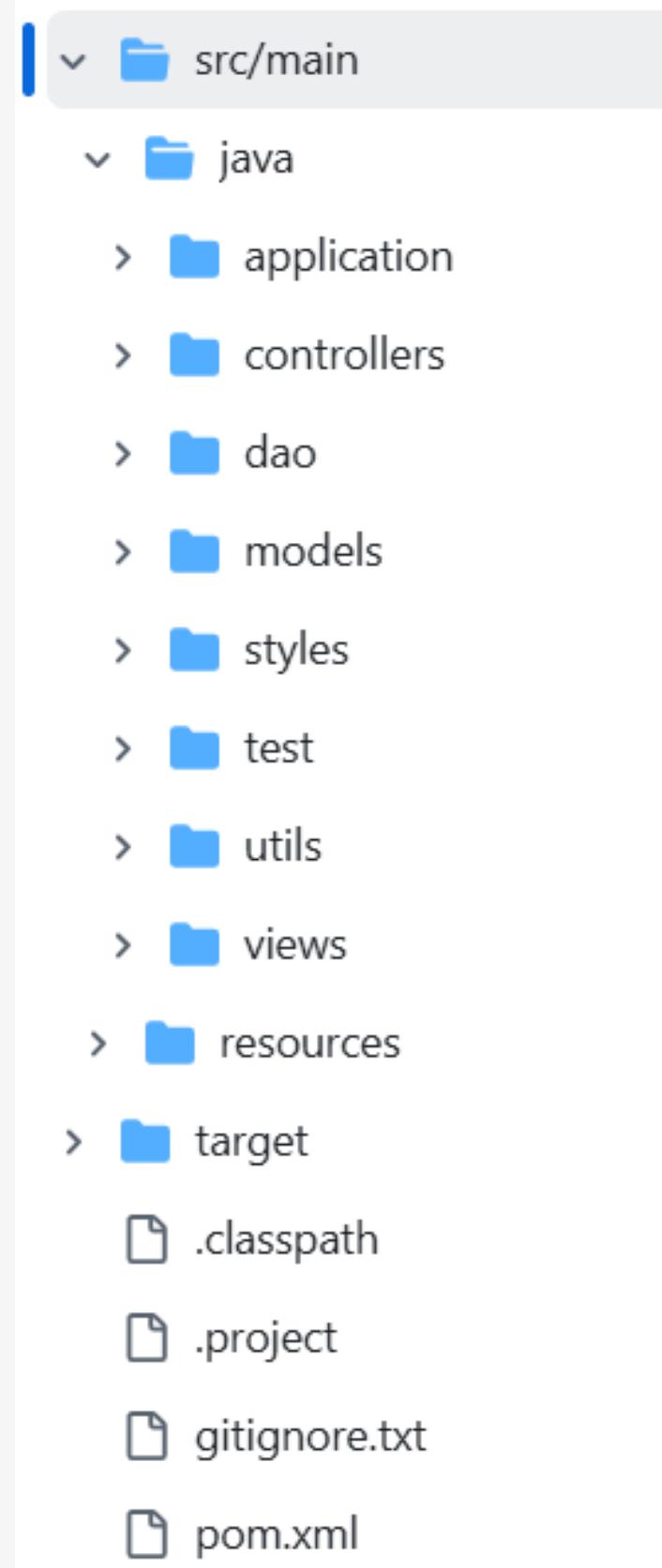
1.-src/main/java

1.8-views

- **CollectionPane.fxml**: Vista para mostrar la colección de juegos del usuario.
- **DescriptionPane.fxml**: Contiene la interfaz con la descripción detallada de un juego.
- **GameContainer.fxml**: Organiza y muestra la lista de juegos en la aplicación.
- **Gameltem.fxml**: Representa la plantilla visual para un solo juego dentro de la lista.
- **GamesPane.fxml**: Vista general donde se muestran todos los juegos disponibles.
- **HeaderPane.fxml**: Sección superior de la interfaz, normalmente para navegación o información.
- **HomePane.fxml**: Vista principal de inicio de la aplicación.
- **Login.fxml**: Interfaz para el inicio de sesión de los usuarios.
- **LoginDataPane.fxml**: Maneja la entrada de datos en el formulario de inicio de sesión.
- **MainPane.fxml**: Contenedor principal que organiza las vistas de la aplicación.
- **RegisterDataPane.fxml**: Vista para la entrada de datos en el formulario de registro.
- **userDataPane.fxml**: Vista para mostrar o editar la información del usuario.

ANÁLISIS DEL PROBLEMA

DIAGRAMA DE CLASES



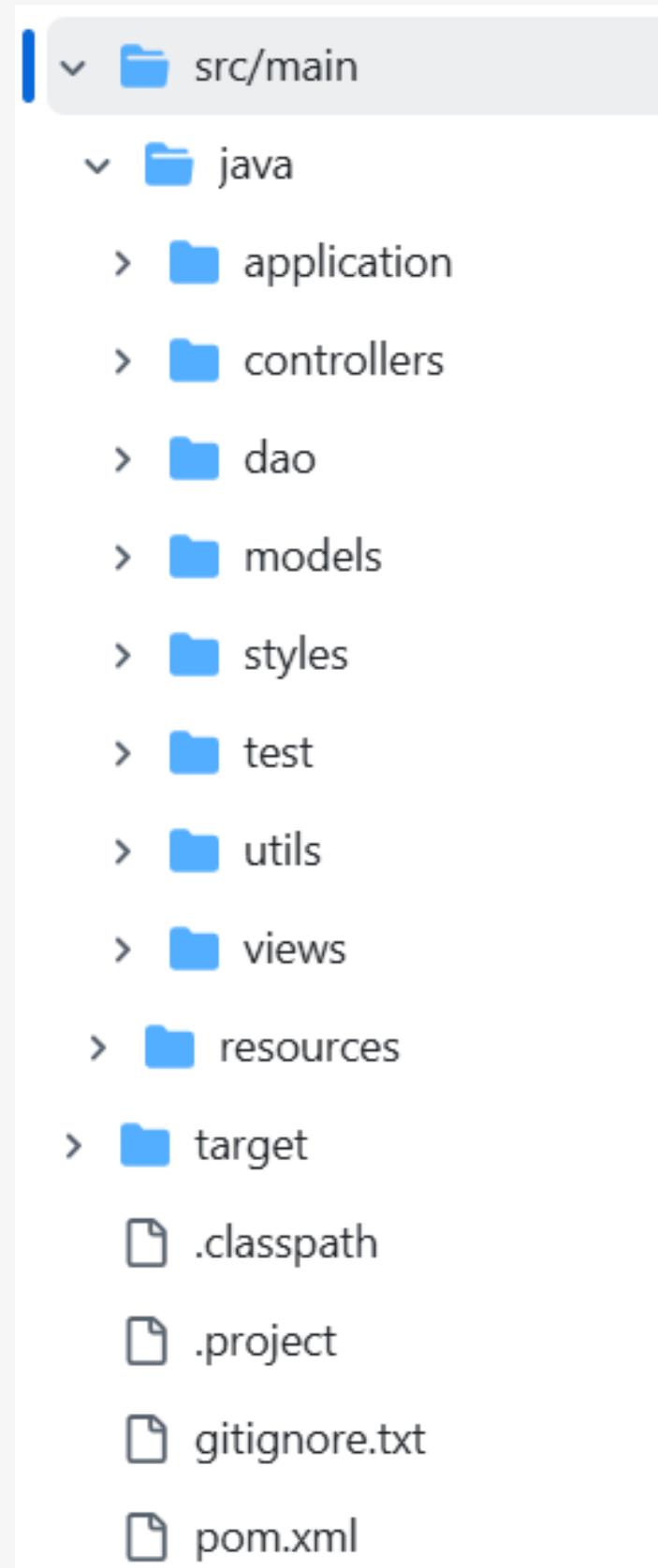
2.-src/main/resources

2.1-Hibernate

- **hibernate.cfg.xml**: Archivo de configuración de Hibernate que define la conexión a la base de datos, el dialecto SQL, las credenciales de acceso y las entidades mapeadas.

ANÁLISIS DEL PROBLEMA

DIAGRAMA DE CLASES



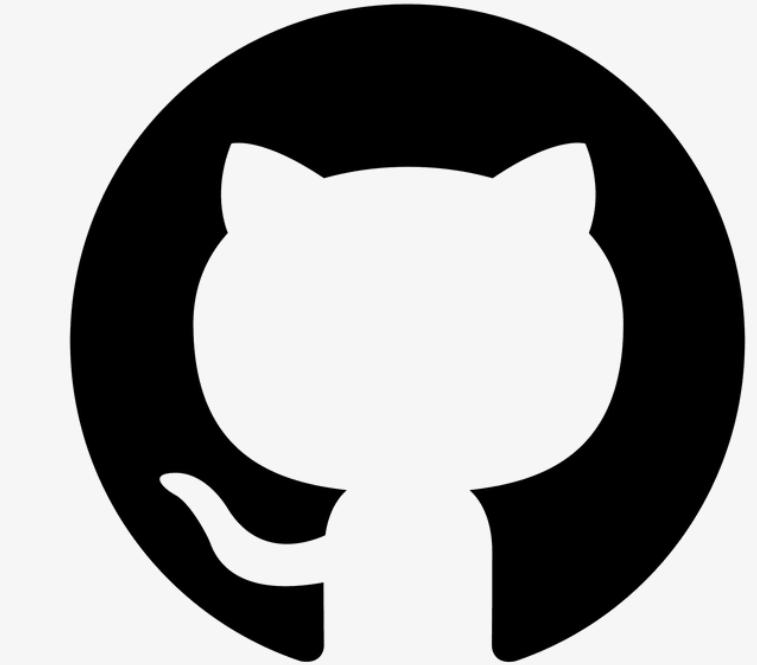
3.-POM

- pom.xml: Archivo de configuración de Maven que gestiona las dependencias, el ciclo de vida del proyecto, los plugins y la compilación del código en Java..

DOCUMENTACIÓN DE LA SOLUCIÓN

ENLACE A GITHUB

PULSE EN LA IMAGEN



GitHub

ENLACES DE INTERES

PULSE EN LOS ENLACES PARA IR A LA DOCUMENTACIÓN

- ◆ *API utilizada en el proyecto*

RAWG Video Game Database API (Para obtener datos de videojuegos)

🔗 <https://rawg.io/apidocs>

- ◆ *Recursos útiles para el desarrollo*

Documentación de JavaFX (Interfaz gráfica en Java)

🔗 <https://openjfx.io/>

Guía de Maven (Para la gestión del proyecto y dependencias)

🔗 <https://maven.apache.org/guides/>

Gluon Mobile (Para portar JavaFX a dispositivos móviles)

🔗 <https://gluonhq.com/products/mobile/>