

Documentacion del Proyecto: Gestion de Ficheros de Videojuegos

Índice

- [Introducción](#)
 - [Datos Guardados](#)
 - [Elemento principal](#)
 - [Campos guardados](#)
 - [Formatos](#)
 - [Programas](#)
 - [1. CrearFicheroLuismi.java](#)
 - [2. ModificarFicheroLuismi.java](#)
 - [3. CrearFicheroXMLLuismi.java](#)
 - [4. LeerFicheroXMLLuismi.java](#)
 - [5. TransformarXSLLuismi.java](#)
 - [6. AppLuismi.java](#)
 - [Ejemplo de Uso](#)
 - [Comentarios](#)
 - [Requisitos y Compilación](#)
 - [Conclusión](#)
-

Introduccion

Este proyecto permite la gestion y manipulacion de un archivo binario de videojuegos usando Java. Ademias, se puede convertir de archivo binario a XML y HTML, lectura con DOM y SAX y el uso de ([JOptionPane](#)) para que se vea mas visual y se pueda probar todo junto. El proyecto tambien esta subido a mi GitHub [GitHub - Luiloprom/ficheros_luismi](#).

Datos Guardados

- **Elemento principal:** Videojuego
- **Campos guardados:**
 - [codigo](#) (int): identificador del videojuego
 - [nombre](#) (String): nombre del videojuego
 - [plataforma](#) (String): plataforma principal
 - [precio](#) (float): precio del videojuego
 - [año](#) (int): año de lanzamiento

Formatos:

- Binario ([videojuegos.dat](#))

- XML (`videojuegos.xml`)
 - HTML (`videojuegos.html`) - tabla generada mediante XSLT
 - Todos situados en la carpeta `src/main/resources`
-

Programas

1. CrearFicheroLuismi.java

Funcionalidad:

Crea el archivo binario `videojuegos.dat` mediante la escritura secuencial de una lista de objetos `VideojuegoLuismi`.

Comentarios:

- Se crea una lista de videojuegos
 - Por cada videojuego, se escribe en el fichero con sus cinco campos
 - El fichero se guarda en resources con `DataOutputStream`
-

2. ModificarFicheroLuismi.java

Funcionalidad:

Busca un videojuego por código y modifica sus datos.

Comentarios:

- Se lee todo el fichero y se almacena en una lista de objetos `VideojuegoLuismi`
 - Si el código coincide, se modifican sus campos y se guarda en la lista.
 - Se reescribe todo el fichero con los datos actualizados y escribiendo la lista completa de nuevo.
-

3. CrearFicheroXMLLuismi.java

Funcionalidad:

Convierte el archivo binario en un archivo XML usando DOM.

Comentarios:

- Se recorre el fichero binario y para cada videojuego se crea un nodo XML
 - Se utiliza un método `crearElemento` para crear y añadir elementos hijos, evitando duplicar código
-

4. LeerFicheroXMLLuismi.java

Funcionalidad:

Muestra el contenido del XML tanto con DOM como con SAX.

Comentarios:

- DOM: Recorre nodos y muestra campos por consola

- SAX: Utiliza un flag string y switch-case para identificar y mostrar cada campo dinamicamente sin booleanos repetidos

Para el apartado de SAX me he ayudado de la IA.

5. TransformarXSLLuismi.java

Funcionalidad:

Genera una tabla HTML a partir del XML utilizando una hoja de estilos XSLT (`videojuegos.xsl`).

Comentarios:

- Usa la API Transformer para aplicar la hoja XSLT y genera videojuegos.html
 - La tabla incluye los campos nombre, plataforma y año
 - El fichero xsl me ayude con IA para generarlo
-

6. AppLuismi.java

Funcionalidad:

Menu grafico con `JOptionPane` para ejecutar todas las funcionalidades anteriores facilmente.

Comentarios:

- Antes de mostrar el menu borra todos los ficheros de resources excepto el .xsl
 - El menu permite crear, modificar, exportar, leer (DOM/SAX), transformar y salir
-

Ejemplo de Uso

1. El usuario selecciona "Crear fichero videojuegos.dat". Se genera un archivo dat con varios videojuegos.
 2. El usuario puede modificar un videojuego introduciendo su codigo y los nuevos datos.
 3. Se puede exportar el binario a XML y visualizar el resultado con DOM o SAX.
 4. Puede transformar el XML a HTML mediante el XSLT.
 5. La interfaz grafica permite que se vea todo mas visual y se pueda hacer todo junto por decirlo asi.
-

Comentarios

- Todo el codigo esta debidamente comentado explicando el proposito de cada clase y metodo.
 - Se emplean metodos auxiliares para evitar duplicacion (especialmente en el manejo de nodos DOM).
 - La estructura del repositorio sigue las convenciones de Maven.
 - Se toman precauciones para evitar corrupcion del archivo binario reescribiendo toda la lista tras modificar un registro.
 - La limpieza de archivos evita borrar la hoja XSL.
-

Requisitos y Compilacion

- Java 8+
 - Maven instalado
 - Extensiones Java y Lombok activas en Visual Studio Code
 - En visual abrir la carpeta raiz Ficheros_Luismi para ejecutar
 - Si no funciona el Lombok que es lo que uso para que me genere el constructor y los getter y setter borrar la dependencia del archivo pom
-

Conclusion

- He realizado este ejercicio ayudandome de la teoria, los ejercicios hechos en clase y en algunos casos me he ayudado de la IA para entender el codigo o que me ayudase con errores que ya no era capaz de solventar.
- Tambien he usado lombok, un proyecto maven, git y GitHub, y la documentacion esta hecha en markdown. Todo esto para practicar contenido de otras asignaturas e irme acostumbrando.

Perdon por las faltas de ortografia ya que al usar markdown no hay auto corrector y muchas no las he puesto