|  |
| --- |
| Asignatura: |
| Desarrollo e Integración de Software |

|  |
| --- |
| Título del documento: |
| **Práctica Final DIS** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Preparado por: |  |  |  | |  |
| Nombre |  | Sergio Luna, Daniel Sánchez, Daniel Ojeda, Andrés Arcones | |  | | 31/12/20 |
|  |  |
|  |  |  |  | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre de fichero: |  | Fecha: |  | Edición: |  | Página: |
| **Memoria Final DIS.docx** |  | **31/12/20** |  | **2** |  | **1/12** |

Registro de cambios

|  |  |  |
| --- | --- | --- |
| **Ed.** | **Fecha** | **Cambio**  **(Incluya el capítulo / subcapítulo y una corta descripción)** |
| 1 | 22/11/20 | V.1 |
| 2 | 31/12/20 | V.2 |

Índice

[1 Introducción 4](#_Toc57104713)

[1.1 Objetivos 4](#_Toc57104714)

[1.2 Descripción del Proyecto 4](#_Toc57104715)

# Introducción

## Objetivos

Aprender a manejar los comandos de Git

Aprender a controlar un repositorio en equipo

Aprender a leer un fichero con Java

Aprender XML y DTD

Aprender a trabajar con lectura de ficheros estructurados

Aprender a usar Docker, Jenkins y Vaadin

Aprender a hacer backend y frontend de una aplicación

## Descripción del Proyecto

En esta práctica, debemos crear una aplicación Java, que lea un fichero JSON, creado a partir de un fichero XML y que nos muestre diversos contenidos.

Lo primero que hacemos es crear nuestro fichero XML, por ejemplo

Texto

Descripción generada automáticamente

Una vez creado nuestro XML, vamos a crear nuestro proyecto usando Maven

Ayudandonos de Maven, creamos nuestro fichero POM.XML, donde vamos a configurar las dependencias y plugins que tendra nuestro proyecto.

![Imagen que contiene Patrón de fondo

Descripción generada automáticamente]()

Señalamos en azul, las dos dependencias que hemos añadido, en este caso, la que nos permite trabajar con GSON y la que nos permite leer XML con Java.

Una vez hecho, nuestro POM.XML, podemos abrir el proyecto, a partir, de este fichero

En nuestro caso, vamos a usar IntelliJ, como entorno de desarrollo.

Captura de pantalla de computadora

Descripción generada automáticamente

Ahora con nuestro entorno abierto, importamos nuestro proyecto, a partir del POM.XML

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ahora, vamos a describir, cómo está formado nuestra aplicación

Siguiendo las indicaciones del documento de la práctica, creamos varias clases, para estructurar mejor la aplicación y después en el Main, invocamos a la clase correspondiente.

![Diagrama

Descripción generada automáticamente]()

En rojo tenemos los distintos ficheros fuente de nuestra aplicación, en este caso, tenemos nuestro fichero XML(videotecas.xml) que ya conocemos, tenemos nuestro fichero JSON(videotecas.json) que se genera en un proceso de nuestra aplicación, que después comentaremos y por último el fichero DTD(videotecas.dtd) que genera nuestro entorno de desarrollo.

En azul tenemos todas nuestras clases Java, con sus métodos, que comentaremos.

En verde tenemos nuestro fichero pom.xml.

Por último, tenemos en morado tenemos la sección de librerías externas, donde se importan aquellas librerías que necesitamos para llevar acabo las tareas que queremos hacer en esta aplicación.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Nuestra primera clase, se llama CrearJson

En ella, como su nombre indica, generamos un fichero JSON, a partir del fichero XML que le pasamos, en nuestro caso, videotecas.xml

Además, también tenemos el método leerFicheroJson() que trae la información del fichero .json a las clases.

Continuar explicación…

La siguiente clase, que podemos apreciar es la clase Actor, que básicamente contiene dos strings (nombre y enlace\_wiki), inicializamos los dos strings a NULL.

Después, creamos dos métodos(void) setNombre y setEnlace\_wiki que los usamos para asignar a esas variables, los valores que encontremos en nuestro fichero mientras lo vamos leyendo.

Por último, formamos la expresión de cómo queremos que el programa saque la información de los actores, que encontremos en el fichero.

Texto

Descripción generada automáticamente

A continuación, vemos otras dos clases muy parecidas, la clase Película y la clase Videoteca

Que se construyen de la misma forma que la de actor, pero adaptándola a los campos que posee Película y Videoteca.

Texto

Descripción generada automáticamente

Imagen que contiene Tabla

Descripción generada automáticamente

Falta comentar las clases LeerXML y Main.

Al final, cuando hacemos una búsqueda en el Main, obtenemos los resultados que buscamos, por ejemplo…

![Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente]()

Aquí vemos, que queremos obtener los actores de la película Star Wars El Retorno del Jedi

Vemos que la aplicación, efectivamente, nos da dichos actores por pantalla.

LeerXML

Hacemos uso de la libraría StAx, usando un xmlstreamreader para leer el xml al completo, hacemos uso de un switch y varios if que buscan el localname del xml (etiqueta) y si esta añade dicho elemento a la videoteca, se ha hecho asi para que no importe el orden, además se ha usado arraylist para que el sistema sea dinámico y se puedan actualizar infinidad de videotecas. Todas las variables se inicializan a null de tal madera que, si no hay género, ese se te quede a null y sea fácil y rápido encontrar aquellas películas sin género.

CrearJson:

Crear json hace uso de la biblioteca de Google, gson. Hacemos uso del método tooJson junto con un writer de java para escribir objetos instanciados al fichero json. Para leer, este hacemos uso del método fromjson que también proporciona la biblioteca para recoger todas las videotecas en un array de videotecas para su uso.

**En los comentarios de las clases java aparece comentado todo en mucho mas detalle.**