



MEMORIA PRÁCTICA 2

Desarrollo e Integración de Software



20 DE ENERO DE 2021

MATHIAS BRUNKOW MOSER
CARLOS COMPANY TORRES
ÁLVARO ORTIZ BALLESTEROS
JULIA QUINTANO MODINO

POZUELO DE ALARCÓN, MADRID

Índice

1.	Introducción	2
1.1	Datos	2
2.	DesarrolloGit.....	3
2.1	Descripción deRamas	3
2.1.1	Features.....	3
2.1.2	Releases yTags.....	5
3.	Descripción deClases	6
3.2	Clase Lector.....	6
3.2.1	Descripción de Clases.....	6
3.3	Clases Tests	7
3.3.1	LectorTest.....	7
3.3.2	MyUITest	7
3.4	Clases de Tipo deDato	7
3.4.1	Clase Atributos	7
3.4.2	Clase Videoteca	8
3.4.3	Clase Película.....	9
3.4.4	Clase Actor	10
3.5	Clase AdvancedFileDownloader	11
3.6	Clase VDException	11
3.7	Clase MyUI.....	12
3.7.1	Descripción de atributos	12
4.	Heroku	13
5.	Docker y Jenkins.....	13
6.	Manual de Usuario.....	17
7.	Conclusión.....	20
8.	Referencias.....	21

1. Introducción

En esta práctica vamos a utilizar todos los conocimientos adquiridos en la asignatura, con el fin de desarrollar una aplicación web basada en Java para gestionar la videoteca creada en nuestra primera entrega. También desplegaremos la aplicación en la plataforma Heroku mediante el proceso de integración continua configurado en Jenkins.

Aplicación desplegada Heroku → <https://videoteca-g6.herokuapp.com/>

1.1 Datos

Los datos que gestionaremos en la aplicación Java serán:

- Película
 - Título
 - Id
 - Año de estreno
 - Minutos de duración
 - Sinopsis
 - Reparto (compuesto por los actores de la película)
 - Género de la película (dato opcional)
 - Enlace IMDB.
- Actor/Actriz
 - Nombre
 - Enlace a la Wikipedia.
- Videoteca
 - Id
 - Nombre
 - Ubicación
 - Películas
 - Fecha última actualización

2. Desarrollo Git

Para trabajar en el proyecto de forma colaborativa y segura hemos utilizado el gestor en línea distribuido GitHub y a través de nuestros ordenadores realizamos la gestión del repositorio mediante el GitBash. Con el fin de poder comprender el avance del proyecto consultamos la herramienta de GitHub Insights > [Network](#).

https://github.com/UFV-INGINF/practica2-dis-dis_g6

Para mantener un flujo controlado de trabajo utilizamos la herramienta de Git Flow, que nos ayudó a crear diferentes “features” y realizar “releases”, ayudándonos a mantener un flujo continuo de información, y permitiéndonos trabajar por separado en nuestras propias ramas (features).

2.1 Descripción de Ramas

Para obtener un desarrollo optimo, continuo y por partes utilizamos una serie de ramas que nos permitieron realizar diferentes funciones por separado.

2.1.1 Features

Para el desarrollo de las funcionalidades utilizamos como base la rama de “develop”, desde la cual creamos una serie de “features”:

Descripción de “Features”		
Nombre	Clases/Documentos Implementados	Descripción
feature/tests	LectorTest.java MyUITest.java	Feature responsable por realizarse el desarrollo de las pruebas, a medida que se desarrollaba las demás funciones.
feature/guardar	Guardar()	Rama responsable de contener el desarrollo de la parte gráfica del guardar las videotecas en un fichero
feature/importar	Import()	Rama responsable de desarrollar la parte de importación de un archivo JSON en el directorio del usuario.
feature/select	Select()	Rama responsable de crear el entorno de selección y el menú del usuario. También es responsable de validar la selección y redirigir a películas.
feature/películas	Peliculas() Editar()	Rama responsable de añadir el CRUD, de crear, editar, leer y borrar las películas, así como la estructura de añadir una película nueva validando el formulario.

2.1.2 Releases y Tags

Estas son las releases que utilizamos para generar las diferentes “tags/versiones del proyecto.

Nombre	Descripción
release/1.0	Release responsable de desplegar a master la primera parte de la aplicación vaadin
release/1.5	Release responsable de actualizar y añadir las clases de test.
release/2.0	Implementa el contenido final desplegado en la plataforma de heroku

3. Descripción de Clases

Para la gestión de la Videoteca, de sus películas y de los actores de estas decidimos separar en varias clases los métodos y variables asociadas. Las clases del proyecto son:

3.2 Clase Lector

Clase que contiene los métodos para parsear el fichero json a objeto de java para su tratamiento en el aplicativo. También añade la función para guardar objetos java en un JSON

3.2.1 Descripción de Clases

Descripción de Funciones de la Clase Lector			
Nombre	Entradas	Salidas	Descripción
find()	Videoteca Existente, Nombre de Fichero	Videoteca Encontrada	Encuentra una videoteca dentro de una lista de videotecas por su nombre y ubicación, ya que no puede haber una misma videoteca con su mismo nombre y ubicación (sirve para no añadir una videoteca existente)
importar()	Videoteca Existente, Nombre de Fichero	Videoteca con videotecas añadidas	Carga de un fichero.json seleccionado por el usuario si existe, las videotecas que contiene el mismo fichero, asignando nuevas id para que todas las películas tengan una id diferente basada en su contenido.
guardar()	Videoteca Existente, Nombre de Fichero	Videoteca Guardada	Guarda las videotecas que han sido añadidas en nuestra aplicación, en un fichero existe o en un fichero nuevo.

3.3 Clases Tests

3.3.1 LectorTest

Funciones de test implementadas

- importarVideotecaConMasVideotecasCreadas()
 - Verifica si se puede importar videotecas con una añadida
- guardarUnFicheroNoJSON()
 - Verifica si se realiza la excepción de fichero no JSON
- findUnaVideotecalqual()
 - Verifica que podemos encontrar una videoteca
- importarUnFicheroNoJSON()
 - Verifica si no podemos importar un fichero no JSON.

3.3.2 MyUITest

Funciones de test implementadas

- hasDigits()
 - Verifica si la función detecta dígitos
- isNumeric()
 - Verifica si la función detecta números
- selectpeliculanoexistente()
 - Verifica no se puede seleccionar un a película no existente

3.4 Clases de Tipo de Dato

3.4.1 Clase Atributos

La función de esta clase es definir el tipo de dato de los atributos los atributos de minutos de duración y años añadiendo las funciones necesarias para su posterior gestión en otras clases.

3.4.1.1 Descripción de Atributos

Descripción de Atributos de la clase Atributos		
Nombre	Tipo	Descripción
min_dur	Entero	Responsable de almacenar los minutos de duración de una película
ano_estreno	Entero	Responsable de almacenar el año de estreno de la película

3.4.2 Clase Videoteca

La función de finir los tipos de datos de las videotecas añadiendo las funciones necesarias para su posterior gestión en otras clases. Nombre, Ubicación, Películas...

3.4.2.1 Descripción de Atributos/Métodos

Descripción de Atributos de la clase Videoteca		
Nombre	Tipo	Descripción
id	int	Responsable de contener el id de la videoteca
nombre	String	Responsable de contener el nombre de la videoteca
ubicacion	String	Responsable de contener la ubicación de la videoteca
peliculas	Lista de Peliculas	Contiene una lista de todas las peliculas disponibles en la videoteca
fecha_ultima_actualizacion	String	Contiene la fecha de última modificación de la videoteca
getGeneros()	Método	Devuelve una lista de géneros de las peliculas que contiene, si una película no tiene genero se añade [No calificada]

3.4.3 Clase Película

El objetivo de esta clase es definir el tipo de dato de cada película añadiendo las funciones necesarias para su posterior gestión en otras clases. Título, Sinopsis, Reparto...

3.4.3.1 Descripción de Atributos

Descripción de Atributo de la clase Pelicula		
Nombre	Tipo	Descripción
id	Entero	Id única de una película, se utiliza la clase hashCode para añadir
titulo	String	Contiene el título de una película
sinopsis	String	Contiene el texto de la sinopsis de la película
reparto	Lista de Actores	Contiene todos los artistas pertenecientes a una película
atributos	Atributo	Contiene los dos atributos de la peli en el tipo de dato atributos
genero	String	Contiene el género de película si existe, si no existe estará vacía
enlaceIMDB	String	Contiene el enlace de IMDB de la película.
getActores()	Método	Devuelve una lista de los nombres de los actores que participan en una película y los guarda en un string

3.4.4 Clase Actor

El propósito de esta clase es definir el tipo dato de cada actor añadiendo las funciones necesarias para su posterior gestión en otras clases. Nombre del Actor y URL Wikipedia.

3.4.4.1 Definición de Atributos

Descripción de Atributos de la clase Actor		
Nombre	Tipo	Descripción
nombre	String	Contiene el nombre del actor
URL_Wiki	String	Contiene la url de Wikipedia del actor

3.5 Clase AdvancedFileDownloader

Clase que nos permite generar un conector que permite la descarga de un archivo en nuestro servidor de jetty, para que el usuario pueda acceder al mismo desde vaadin:

Referencia:

<https://vaadin.com/attachment/106dfd4f-0c82-4fda-b345-a9c1daf58a7d/AdvancedFileDownloader.java>

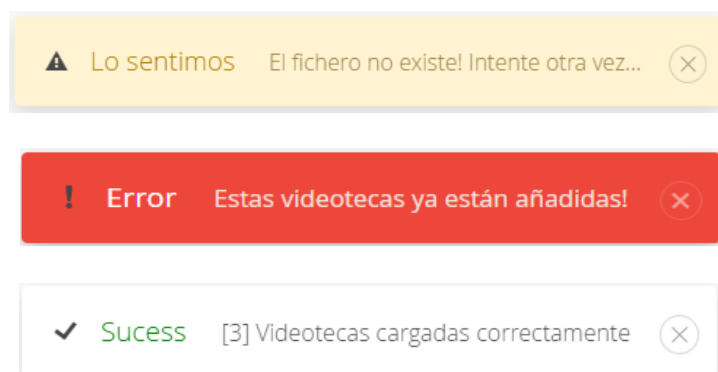
<https://vaadin.com/forum/thread/3329519/download-generated-file>

3.6 Clase VDException

Hemos creado la clase VDException (ViDeotecaException) que nos permite capturar las excepciones lanzadas por lectura y escritura de las videotecas

Gracias a ella cada vez que hay una excepción podemos enviar:

- Un mensaje
- Un mensaje y una causa
- Una causa



3.7 Clase MyUI

Esta clase es principal pues es donde se abre Vaadin. La clase de entrada es init.

3.7.1 Descripción de atributos

Descripción de Atributos de la clase Actor		
Nombre	Tipo	Descripción
editar	Void	Función que permite editar una película y borrarla
guardar	Void	Función que permite guardar o actualizar un fichero (b = true [Crear], b = false [Actualizar])
hasDigits	Boolean	Función que detecta si una cadena tiene caracteres numéricos
importar	Void	Función que inicia la función que importa videotecas y establece un entorno seguro para captar la excepciones de importar e imprimirlas en Vaadin
init	Void	Función de entrada
isNumeric	Boolean	Función que detecta si los caracteres no son numéricos o no se pueden parsear
peliculas	Void	Función que añade el layout de películas para poder verlas, editarlas y leerlas
select	Void	Función encargada de seleccionar una videoteca y abrir su menú

4. Heroku

En primer lugar, nos hemos creado una cuenta en Heroku, y hemos creado desde la página de Heroku una nueva aplicación, a la cual hemos llamado videoteca_g6.

Después en el propio bash de git hemos inicializado el Heroku y, después de logearnos, a través del comando `heroku git:clone -a videoteca_g6`.

Para depositar nuestro proyecto en el repositorio Heroku, utilizamos el comando `git push heroku master` (importante estar en master para que al pushear nos haga el build también)

Ahora, configuraremos nuestro proyecto para que funcione en Heroku. Para ello primero añadiremos un plugin de maven que nos permite incluir la dependencia de jetty en el fichero jar generado como dependencia para que nuestra aplicación levante correctamente en Heroku.

Segundo, añadiremos a la raíz de nuestro proyecto el archivo Procfile que nos permite ejecutar nuestra aplicación Vaadin en Heroku.

5. Docker y Jenkins

En segundo lugar, vamos a montar una imagen Jenkins en el contenedor Docker. Iniciamos Docker a través de `docker run -d -p 80:80 docker/getting-started`, y hacemos un pull de la imagen de Jenkins con `docker pull jenkins/Jenkins`

Creación volumen docker

```
carlo@DESKTOP-7I29G7H MINGW64 ~/OneDrive/UFV/3 CURSO/1º Cuatrimestre/DIS/Practica 2/practica2-dis-dis_g6 (master)
$ docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
alpine/git          latest      04dbb58d2cea  14 hours ago  25.1MB
jenkins/jenkins     latest     10fafa8484ea  7 days ago   721MB
jenkins/jenkins     lts        e5599f396096  7 days ago   717MB
docker/getting-started latest     021a1b85e641  6 weeks ago  27.6MB

carlo@DESKTOP-7I29G7H MINGW64 ~/OneDrive/UFV/3 CURSO/1º Cuatrimestre/DIS/Practica 2/practica2-dis-dis_g6 (master)
$ docker volume ls
DRIVER      VOLUME NAME
local       b0bbcf000b96f299e28a458b22e6ac60204e5c208aa9b85a3f0d04e88520cffa

carlo@DESKTOP-7I29G7H MINGW64 ~/OneDrive/UFV/3 CURSO/1º Cuatrimestre/DIS/Practica 2/practica2-dis-dis_g6 (master)
$ docker volume create jenkindis
jenkindis
```

Inicio el Jenkins con Docker en local

```
carlo@DESKTOP-7129G7H MINGW64 ~/OneDrive/UFV/3 CURSO/1º Cuatrimestre/DIS/Practica 2/practica2-dis-dis_g6 (master)
$ docker run -p 8080:8080 -p 50000:50000 -v jenkinsdis:/var/jenkins_home jenkins/jenkins:its
Running from: /usr/share/jenkins/jenkins.war
webroot: EnvVars.masterEnvVars.get("JENKINS_HOME")
2021-01-20 19:07:33.149+0000 [id=1] INFO org.eclipse.jetty.util.log.Log#initialized: Logging initialized @728ms to org.
2021-01-20 19:07:33.317+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2021-01-20 19:07:36.379+0000 [id=1] WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2021-01-20 19:07:36.472+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-9.4.33.v20201020; built: 2020-1
2ef1c2ebbc99e25420; jvm 1.8.0_242-b08
2021-01-20 19:07:36.833+0000 [id=1] INFO o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did no
2021-01-20 19:07:36.934+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: DefaultSessionIdManager workerName=
2021-01-20 19:07:36.935+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: No SessionScavenger set, using defa
2021-01-20 19:07:36.938+0000 [id=1] INFO o.e.j.server.session.HouseKeeper#startScavenging: node0 Scavenging every 60000
2021-01-20 19:07:37.476+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: /var/jenkins_ho
_JENKINS_HOME")
2021-01-20 19:07:37.666+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started w.@1b58ff9e[Jenkins v2.263.2,/
/var/jenkins_home/war}
2021-01-20 19:07:37.694+0000 [id=1] INFO o.e.j.server.AbstractConnector#doStart: Started ServerConnector@126253fd[HTTP/
2021-01-20 19:07:37.694+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started @5273ms
2021-01-20 19:07:37.700+0000 [id=22] INFO winstone.Logger#logInternal: winstone Servlet Engine running: controlPort=disa
2021-01-20 19:07:40.071+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Started initialization
2021-01-20 19:07:40.124+0000 [id=34] INFO jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2021-01-20 19:07:42.017+0000 [id=28] INFO jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2021-01-20 19:07:42.026+0000 [id=28] INFO jenkins.InitReactorRunner$1#onAttained: Started all plugins
```

Fijo las credenciales

Panel de Control > mavenjenkins >

General Configurar el origen del código fuente Disparadores de ejecuciones Entorno de ejecución Ejecutar

Add Credentials

Domain: Global credentials (unrestricted)

Kind: SSH Username with private key

Scope: Global (Jenkins, nodes, items, all child items, etc)

ID: dis3

Description: credentials to access github

Username: carletes6

Private Key: ☒ Enter directly
Key:

Enter New Secret Below

Tarea creada

Jenkins

Carlos Company

Desconectar

Panel de Control

Nueva Tarea

Personas

Historial de trabajos

Relacion entre proyectos

Comprobar firma de archivos

Administrar Jenkins

Mis vistas

Lockable Resources

New View

Trabajos en la cola

No hay trabajos en la cola

añadir descripción

Todo

+

S	W	Nombre	Último Éxito	Último Fallo	Última Duración
		jenkins	5 Min 43 Seg - #1	N/D	7 Seg

Icono: S M L

Guía de iconos

Atom feed para todos

Atom feed para fallas

Atom feed para los más recientes

Test realizados correctamente

localhost:8080/job/jenkins/lastBuild/console

Panel de Control > jenkins > #13

Progress (1): 4.1/37 kB
Progress (1): 8.2/37 kB
Progress (1): 12/37 kB
Progress (1): 16/37 kB
Progress (1): 20/37 kB
Progress (1): 25/37 kB
Progress (1): 29/37 kB
Progress (1): 33/37 kB
Progress (1): 37/37 kB
Progress (1): 37 kB

Downloaded from central: <https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/2.12.4/surefire-junit4-2.12.4.jar> (37 kB at 1.4 MB/s)

T E S T S

Running org.dis.practica2.grupo6.backend.LectorTest
ImportarVideotecaConMasVideotecasCreadas: Expected [3]
| NumVideotecasAntes: 1
| NumVideotecasDespues: 3
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.327 sec
Running org.dis.practica2.grupo6.frontend.MyUITest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.261 sec

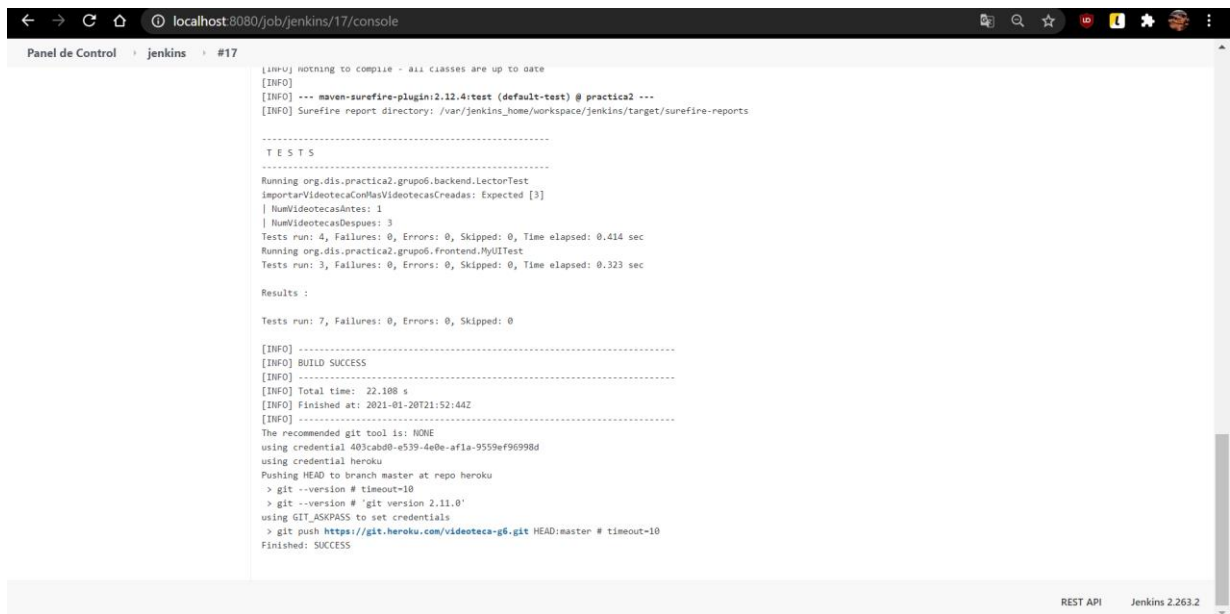
Results :

Tests run: 7, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-war-plugin:3.2.0:war (default-war) @ practica2 ---
Downloading from central: <https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-component-annotations/1.6/plexus-component-annotations-1.6.pom>
Progress (1): 748 B

Downloaded from central: <https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-component-annotations/1.6/plexus-component-annotations-1.6.pom> (748 B at 33 kB/s)
Downloading from central: <https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-containers/1.6/plexus-containers-1.6.pom>
Progress (1): 3.8 kB

Test realizados + enlace con heroku



The screenshot shows a Jenkins console output for a build. The output includes information about the Maven Surefire plugin, test results for two test classes, and the successful deployment to Heroku. The test results show that all tests passed. The deployment steps include pushing the HEAD to the master branch of the Heroku repository.

```
localhost:8080/job/jenkins/17/console
Panel de Control > jenkins > #17

[Linux] nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ practica2 ---
[INFO] Surefire report directory: /var/jenkins_home/workspace/jenkins/target/surefire-reports

-----
T E S T S
-----
Running org.dis.practica2.grupo6.backend.LectorTest
ImportarVideotecaConMasVideotecasCreadas: Expected [3]
| NumVideotecasAntes: 1
| NumVideotecasDespues: 3
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.414 sec
Running org.dis.practica2.grupo6.frontend.HyuiTest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.323 sec

Results :

Tests run: 7, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 22.108 s
[INFO] Finished at: 2021-01-20T21:52:44Z
[INFO] -----
The recommended git tool is: NONE
using credential 403cabd0-e539-4e0e-afla-9559ef96998d
using credential heroku
Pushing HEAD to branch master at repo heroku
> git --version # timeout=10
> git --version # 'git version 2.11.0'
using GIT_ASKPASS to set credentials
> git push https://git.heroku.com/videoteca-g6.git HEAD:master # timeout=10
Finished: SUCCESS

REST API Jenkins 2.263.2
```

6. Manual de Usuario

Seleccionar una Película



Id	Nombre	Generos	Ubicación	Modified
1	Mas que Cine	[Ciencia Ficción, Fantasía, Aventura]	Pozuelo, Madrid, España	20/01/2021
2	Mil y una Películas	[Comedia, Superheroes, No calificada]	Triana, Sevilla, España	16/11/2020
3	Cinema World	[Histórica, Oeste, No calificada]	Gandia, Valencia, España	20/11/2020

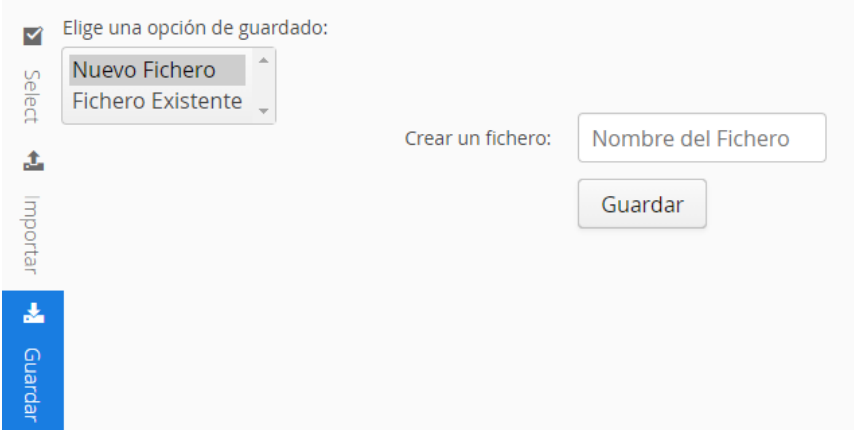
*Haz click sobre cualquier campo de una videoteca para ver y editar sus películas

Importar una Videoteca



Seleccionar archivo Ningún archivo seleccionado Importar

Guardar en un nuevo fichero



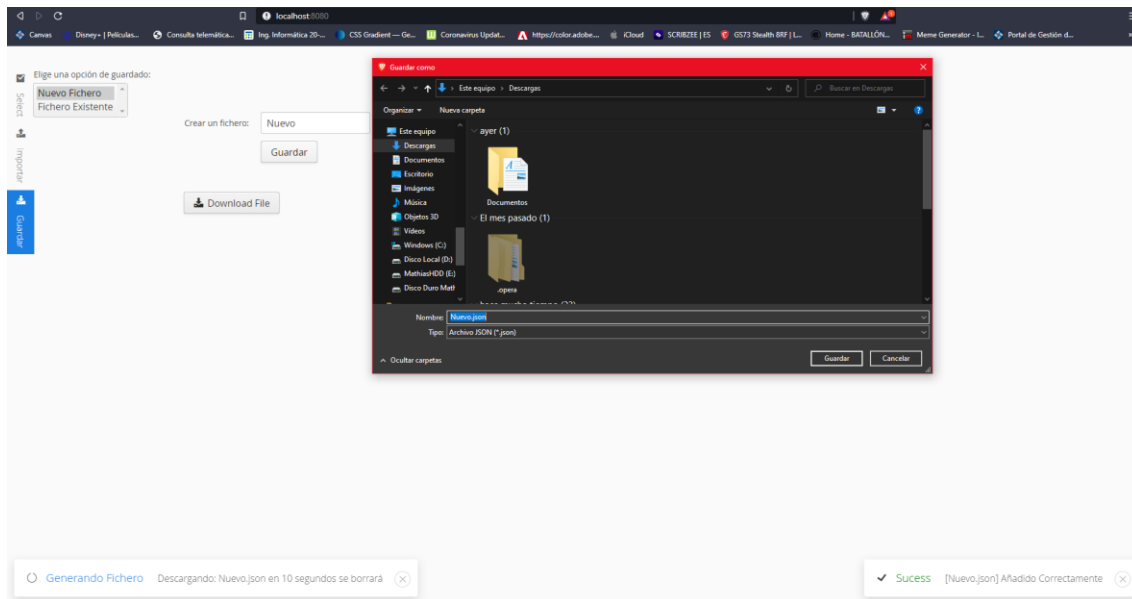
Elige una opción de guardado:

Nuevo Fichero
Fichero Existente

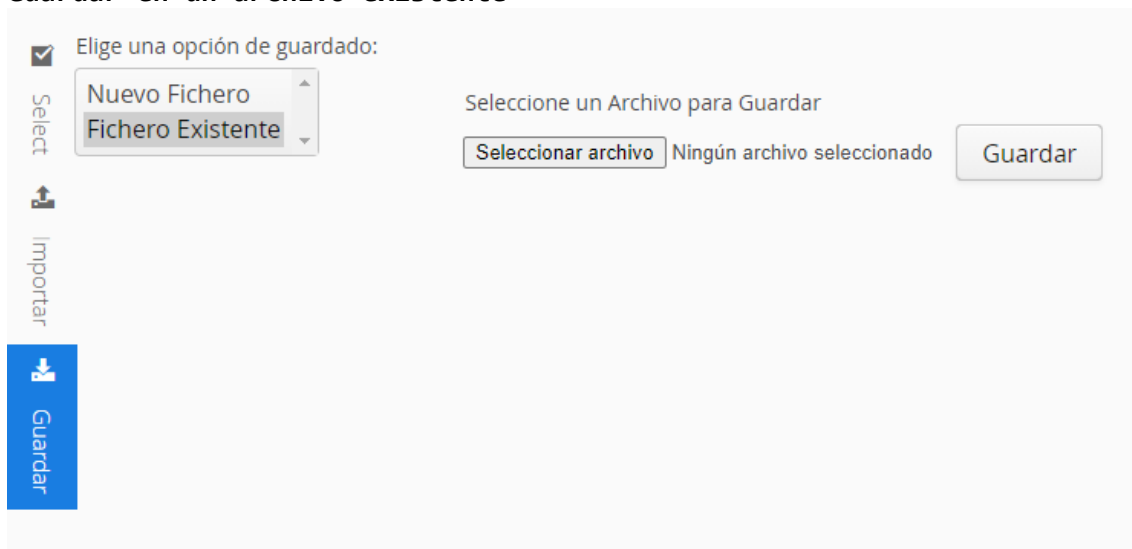
Crear un fichero: Nombre del Fichero

Guardar

Descargar un archivo nuevo creado



Guardar en un archivo existente

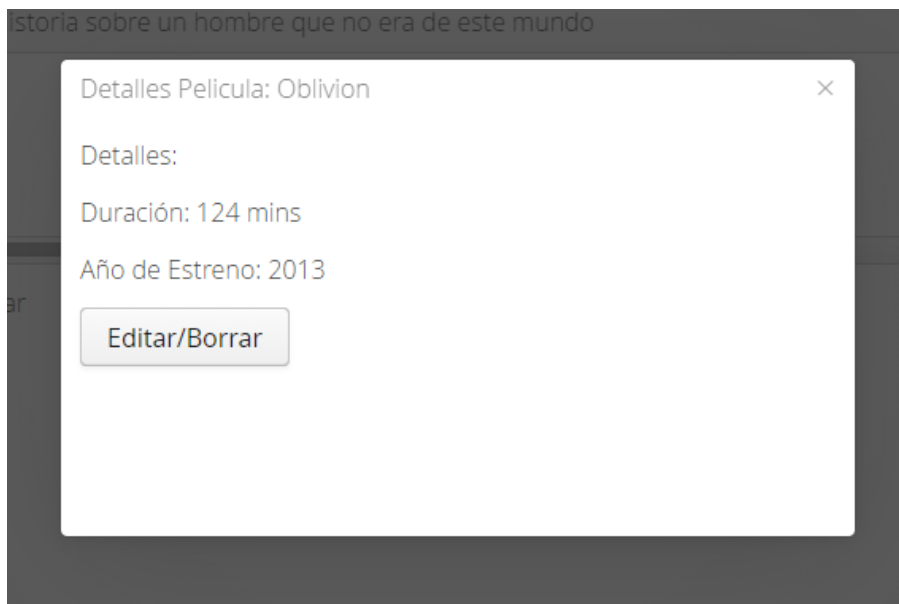


Ver detalles y editar

Ver	Id	Título	Sinopsis
	1049525321	Oblivion	Un veterano asignado para extraer los recursos restantes de la Tierra comienza a cuestionar lo que sabe sobre su misión y sobre sí mismo.
	1598649655	Star Wars: El Retorno del Jedi	Después de una atrevida misión para rescatar a Han Solo de Jabba the Hutt, los rebeldes van a Endor para destruir la segunda Estrella de la Muerte. Mientras tanto, Luke lucha por ayudar
	872594334	El Señor de los Anillos: El Retorno del Rey	Gandalf y Aragorn lideran el Mundo de los Hombres contra el ejército de Sauron para desviar su mirada de Frodo y Sam mientras se acercan al Monte del Destino con el Anillo Único.
	369177550	Harry Potter y las Reliquias de la Muerte	Harry, Ron y Hermione buscan los Horrocruxes restantes de Voldemort en su esfuerzo por destruir al Señor Oscuro mientras la batalla final continúa en Hogwarts.
	1530120357	Underwater	Un equipo de investigadores oceánicos que trabaja para una empresa de perforación en aguas profundas intenta ponerse a salvo después de que un misterioso terremoto devastara sus
	1330118201	World of Men	Una emocionante historia sobre un hombre que no era de este mundo

**Haz click sobre cualquier campo de una película para ver los detalles u editar o borrar

Modal detalles película y editar



Editar películas

Editar

Borrar

Volver

Editar película: [Oblivion]

Un veterano asignado para extraer los recursos restantes de la

Atributos:

[1-20] Actores

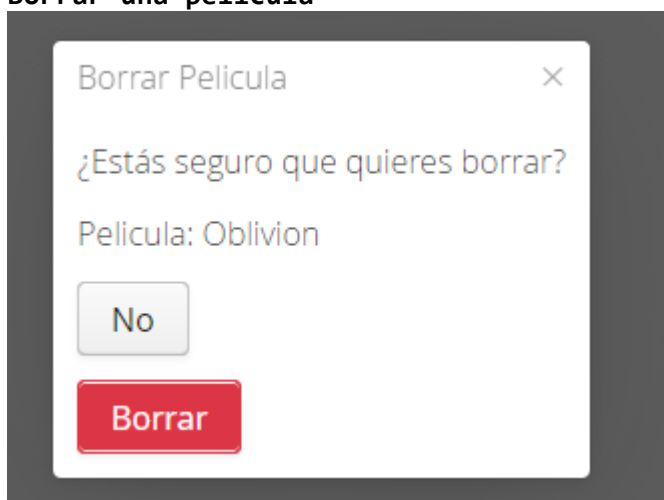
Editar Pelicula

Actor 1

Actor 2

Actor 3

Borrar una pelicula



7. Conclusión

La realización de esta práctica nos ha servido para afianzar los conocimientos adquiridos en la asignatura. En la primera parte del trabajo desarrollamos la aplicación en Java para poder gestionar la videoteca creada en la anterior práctica. Mediante el framework de Vaadin realizamos un desarrollo TDD, y mediante tests unitarios pudimos comprobar la funcionalidad de la aplicación.

En la segunda parte integramos la aplicación en Heroku, ya que es una buena plataforma para crear una aplicación donde ejecutar nuestro proyecto. Ha sido la parte más “sencilla” del proyecto, ya que hemos sabido adaptarnos al software más o menos bien. Previamente está explicada la manera en la que trabajamos con este software. Además, hemos llegado a la conclusión de que, con tiempo y constancia, se pueden conseguir cosas bastante efectivas, ya que proporciona algunas ventajas respecto a github (por ejemplos, nos hace el build del proyecto directamente desde el git bash).

En la tercera parte, montamos una imagen Jenkins en Docker, que nos ha servido para desplegar nuestra aplicación en Heroku, y gracias a ello Heroku nos ofrece un repositorio y una plataforma web para poder probar nuestra aplicación en un entorno real.

8. Referencias

Proyecto de Git: https://github.com/UFV-INGINF/practica2-dis-dis_g6

Proyecto en Heroku: <https://dashboard.heroku.com/apps/videoteca-g6>

Aplicacion Heroku: <https://videoteca-g6.herokuapp.com/>

Apuntes de la asignatura Desarrollo e Integración del Software de la UFV

- Librería Junit versión 4.11
- Librería Gson versión 2.8.6
- <https://mkyong.com/java/how-do-convert-java-object-to-from-json-format-gson-api/>
- Plugin Maven Jar + Dependencias
- https://es.wikipedia.org/wiki/Tom_Cruise
- https://es.wikipedia.org/wiki/Morgan_Freeman
- https://es.wikipedia.org/wiki/Andrea_Riseborough
- <https://www.imdb.com/title/tt1483013/>
- https://es.wikipedia.org/wiki/Mark_Hamill
- https://es.wikipedia.org/wiki/Harrison_Ford
- https://es.wikipedia.org/wiki/Carrie_Fisher
- https://www.imdb.com/title/tt0086190/?ref=nm_sr_srsrg_0
- https://es.wikipedia.org/wiki/Elijah_Wood
- https://es.wikipedia.org/wiki/Viggo_Mortensen
- https://es.wikipedia.org/wiki/Ian_McKellen
- https://www.imdb.com/title/tt0167260/?ref=nm_sr_srsrg_1
- https://es.wikipedia.org/wiki/Daniel_Radcliffe
- https://es.wikipedia.org/wiki/Emma_Watson
- https://es.wikipedia.org/wiki/Rupert_Grint
- https://www.imdb.com/title/tt1201607/?ref=nm_sr_srsrg_0
- https://es.wikipedia.org/wiki/Kristen_Stewart
- https://es.wikipedia.org/wiki/Vincent_Cassel
- https://es.wikipedia.org/wiki/Mamoudou_Athie
- <https://www.imdb.com/title/tt5774060/>
- https://es.wikipedia.org/wiki/Clara_Lago
- https://es.wikipedia.org/wiki/Dani_Rovira
- https://es.wikipedia.org/wiki/Carmen_Machi
- https://www.imdb.com/title/tt2955316/?ref=nm_sr_srsrg_0
- https://es.wikipedia.org/wiki/Macaulay_Culkin
- https://es.wikipedia.org/wiki/Joe_Pesci
- https://es.wikipedia.org/wiki/Daniel_Stern
- https://www.imdb.com/title/tt0104431/?ref=nm_sr_srsrg_0
- https://es.wikipedia.org/wiki/Robert_Downey_Jr
- https://es.wikipedia.org/wiki/Chris_Evans
- https://es.wikipedia.org/wiki/Chris_Hemsworth
- https://www.imdb.com/title/tt4154756/?ref=nm_sr_srsrg_0
- https://es.wikipedia.org/wiki/Leonardo_DiCaprio

- <https://www.imdb.com/title/tt0120338/>
- [https://es.wikipedia.org/wiki/Russell Crowe](https://es.wikipedia.org/wiki/Russell_Crowe)
- [https://es.wikipedia.org/wiki/Connie Nielsen](https://es.wikipedia.org/wiki/Connie_Nielsen)
- [https://es.wikipedia.org/wiki/Eric Bana](https://es.wikipedia.org/wiki/Eric_Bana)
- <https://www.imdb.com/title/tt1853728/>
- https://www.imdb.com/title/tt0172495/?ref=nm_sr_srsrg_0