

# Informe de prácticas de ISDCM

## Entrega 1

Alumno/a: Albert Bausili Fernández

Alumno/a: Noa Yu Ventura Vila

**UNIVERSITAT POLITÈCNICA DE CATALUNYA**  
**BARCELONATECH**

---

**Facultat d'Informàtica de Barcelona**



## Contenido

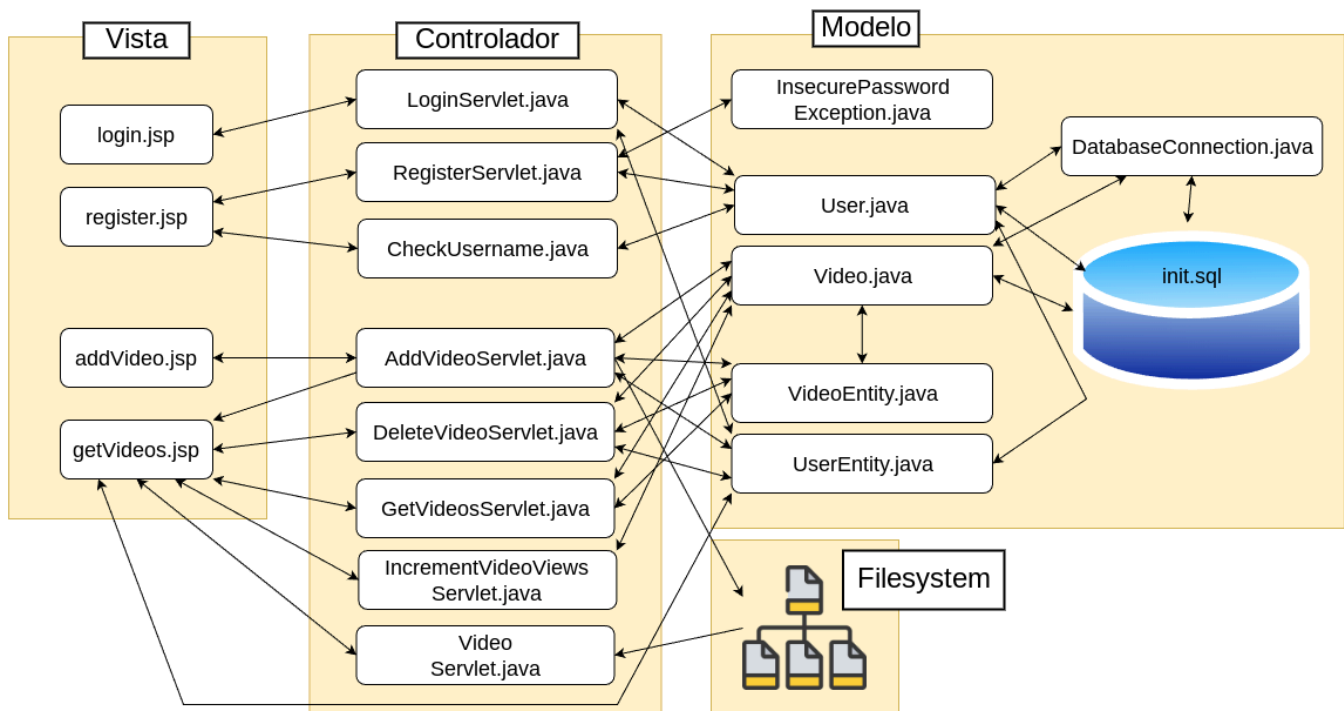
Introducción.....	1
Decisiones de diseño.....	1
Extras realizados.....	3
Repositorios de código consultados.....	3
Bibliografía consultada.....	3

## Introducción

Este proyecto es una aplicación web en Java usando Servlets y JSP que permite a cada usuario gestionar sus vídeos.

Este informe contiene la arquitectura básica de la aplicación web con una breve descripción de sus componentes y los recursos utilizados de Internet. También hay un apartado donde se describen todos los extras que hemos realizado en esta entrega.

## Decisiones de diseño



Hemos implementado mecanismos de control de errores por todas partes. Donde es necesario se capturan las excepciones correspondientes (como por ejemplo el `SQLException` para las llamadas a la base de datos) y se proporciona un mensaje de error claro. También hay otros errores que no requieren lanzar una excepción, como por ejemplo si no existe un usuario con el nombre de usuario entrado o si la contraseña entrada es incorrecta. En este caso se redirige al usuario a la página correspondiente para que pueda volver a intentar la acción (si se ha equivocado con la contraseña se le redirige al `login.jsp` nuevamente). En caso de errores inesperados los controlamos y damos un mensaje indicando que ha sucedido un error catastrófico.

Página	Breve descripción
login.jsp	Contiene la tag <i>style</i> y bootstrap para el diseño de la página. Comprueba que el nombre de usuario y la contraseña son correctos. Se puede acceder sin sesión.
register.jsp	Contiene la tag <i>style</i> y bootstrap para el diseño de la página. Tiene un script

	que en tiempo real llama a CheckUsername.java para ver si el nombre de usuario ya existe (mensaje de color verde indicando que el nombre de usuario está libre, o rojo con un error en caso contrario). Marca los campos que son requeridos y los que no, e impide registrarse si hay campos requeridos sin rellenar. Se puede acceder sin sesión.
addVideo.jsp	Contiene la tag <i>style</i> y bootstrap para el diseño de la página. Vista para colgar un nuevo vídeo después de darle al botón "Upload File" en la vista getVideos.jsp. Contiene un script que envía el vídeo y la información introducida por el usuario (Título y descripción) al servlet y automáticamente obtiene la duración del vídeo. Requiere sesión.
getVideos.jsp	Contiene la tag <i>style</i> y bootstrap para el diseño de la página. Comprueba que haya un usuario que se haya logueado para mostrar el icono de borrar video solo en los que el usuario haya subido. Cada video tiene un botón para reproducir mediante streaming de un servlet y al reproducirlo llama a otro servlet que aumenta el número de vistas en 1. Requiere sesión.
LoginServlet.java	Tiene las mismas comprobaciones de errores que en el JSP y llama a User.java para comprobar que el nombre de usuario y la contraseña son correctos. Si son correctos inicia la sesión del usuario.
RegisterServlet.java	Tiene las mismas comprobaciones de errores que en el JSP para ver si la contraseña es segura (en caso negativo lanza un InsecurePasswordException). Si los datos son correctos los envía a la capa modelo para guardarlos en la DB y redirige al usuario al login. No requiere sesión.
CheckUsername.java	Llama a User.java para comprobar que el nombre de usuario no existe aún (y si no impide proseguir con el registro). Se usa cada vez que el usuario deja de teclear durante un tiempo específico en el campo de Usuario en el formulario de registro. No requiere sesión.
AddVideoServlet.java	Crea un nuevo vídeo con las características que recibe con la request y llama a Video.java para que lo añada a la base de datos. Guarda el video también en el filesystem para que después se pueda recuperar. Requiere sesión.
DeleteVideoServlet.java	Cuando recibe la petición comprueba que el vídeo exista, que quien quiere borrarlo sea el autor de ese vídeo y que la operación haya sido exitosa. Requiere sesión.
GetVideosServlet.java	Devuelve todos los vídeos que están subidos en la plataforma. Requiere sesión.
IncrementVideo ViewsServlet.java	Cuando recibe un POST incrementa el número de visitas del vídeo que dice el parámetro en 1. Se llama desde getVideos.jsp. Requiere sesión.
VideoServlet.java	Hace el streaming del vídeo cuando se quiere visualizar. Requiere sesión.
UserEntity.java	Clase usuario con sus atributos y métodos. Requiere sesión.

VideoEntity.java	Clase vídeo con sus atributos y métodos. Requiere sesión.
User.java	Gestiona todas las llamadas a la base de datos relacionadas con el usuario: registro, autenticación, hashear contraseña y comprobar si el nombre de usuario existe. Requiere sesión.
Video.java	Gestiona todas las llamadas a la base de datos relacionadas con los vídeos: añadir uno nuevo, obtener todos los vídeos, incrementar las visitas en 1 de un vídeo en concreto, obtener un vídeo por ID y borrar vídeos. Requiere sesión.
DatabaseConnection.java	Crea la conexión con la base de datos. Requiere sesión.
InsecurePassword Exception.java	La comprobación de que la contraseña que quiere crear el usuario en la vista de registro es segura solamente está implementada en el servlet de registrar usuario (no ha dado tiempo a hacerlo en el JSP, pero en un futuro lo implementaremos). Cuando la contraseña no es segura se lanza esta excepción explicándole al usuario los requerimientos para que la contraseña sea segura. Requiere sesión.

## Extras realizados

Propuestos por el enunciado:

- Guardar los vídeos en el filesystem y guardar el path donde se encuentran en la DB.
- Aplicar estilos a las vistas para que sean más bonitas.

Iniciativa propia:

- Streaming de los vídeos.
- Borrado de los videos (cada usuario solo puede borrar los que haya subido)
- Mejora en la UX del registro del usuario vía comprobación de nombre usuario a tiempo real.
- Mejora en la UX de registro y subida de videos implementando los campos requeridos.
- Incremento de views cada vez que se reproduce un video.
- Comprobación de la seguridad de las passwords para cumplir con los requisitos de OWASP. De momento para esta release se encuentra solo en el servlet, en la siguiente release lo añadiremos también en el frontend para mejorar la UX del usuario.

## Repositorios de código consultados

Nuestros repositorios de código propios de la asignatura de GEI-AD.

## Bibliografía consultada

<https://www.geeksforgeeks.org/user-defined-custom-exception-in-java/>

<https://www.studytonight.com/java-type-conversion/how-to-convert-java-timestamp-to-date>

[https://cheatsheetseries.owasp.org/cheatsheets/Authentication\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html)

<https://forum.bubble.io/t/find-video-duration-using-javascript/258505>

<https://stackoverflow.com/questions/8557490/redirect-to-a-different-url>

<https://howtodoinjava.com/java/io/inputstream-to-outputstream/>