Informe de prácticas de ISDCM

Entrega 1

Alumno/a: Javier Abella

Alumno/a: Ana Poveda



Contenido

[Introducción 1](#_Toc192373505)

[Decisiones de diseño 1](#_Toc192373506)

[Arquitectura final implentada 1](#_Toc192373507)

[Vistas 1](#_Toc192373508)

[Estilos 1](#_Toc192373509)

[Control de errores: 1](#_Toc192373510)

[2. Control de errores en los métodos doGet y doPost 1](#_Toc192373511)

[3. Redirección con mensajes de errores persistentes 1](#_Toc192373512)

[4. Controlar la expiración de la sesión 1](#_Toc192373513)

[5. Validación de los campos al registrar un usuario: 2](#_Toc192373514)

[6. Validación de los campos al hacer login: 2](#_Toc192373515)

[Repositorios de código consultados 2](#_Toc192373516)

[Bibliografía consultada 2](#_Toc192373517)

# Introducción

La aplicación web creada en esta práctica consiste en iniciar sesión, registrar nuevos usuarios, registrar videos y visualizar una lista de los vídeos disponibles.

# Decisiones de diseño

## Arquitectura final implentada

## Vistas

Las vistas que hemos creado son las mismas que se especifican en la figura 3 del documento **Descripción Proyecto y Entrega 1***, login.jsp, registroUsu.jso, registroVid.jsp y listado.jsp*. Además, hemos añadido la vista *error.jsp*, a la que dirigimos en caso de que haya un error en base de datos o se intente realizar una acción no válida.

## Estilos

Hemos creado una carpeta css en la que hemos añadido ficheros Cascading Style Sheet. Hemos creado un .css por cada vista. De esta forma nos resulta mucho más sencillo localizar los estilos que queremos editar en uan vista en concreto.

Control de errores:  
1. Gestionar errores de la Base de datos:

En los métodos para hacer login y para registrar a un nuevo usuario usamos la clase SQLException para gestionar excepciones de la base de datos. Esta clase nos permite obtener información acerca de los errores y avisos que se producen mientras accedemos a la base de datos. En caso de que se produzca un error, redirigimos al usuario s a una vista especifica, error.jsp.

### 2. Control de errores en los métodos doGet y doPost

Para evitar peticione son válidas, gestionamos tanto las acciones definidas por nosotros, por ejemplo, *register, login y logout*, y otras posibles acciones. De esta forma, i el usuario intenta acceder a servletUsuarios sin la acción correcta le redirijimos a la vista error.jsp.

### 3. Redirección con mensajes de errores persistentes

Para mantener los atributos de error y no perderlos con el método sendRedirect(), usamos la clase HttpSession en lugar de request.setAttibute() en el método loginUser() del servletUsuarios. A continuación, en la vista login.jsp recuperamos el error.

### 4. Controlar la expiración de la sesión

Verificamos si la sesión sigue activa antes de redirigir. Si un usuario deja su sesión abierta y vuelve después de un tiempo, la sesión puede expirar. En ese caso, se le redirigirá a la página de login.

### 5. Validación de los campos al registrar un usuario:

Para implementar la funcionalidad de registrar un usuario se han realizado las siguientes validaciones:

* Todos los campos del formulario deben ser informados.
* Los valores indicados en los campos de contraseña, *Contraseña* y *Repetir Contraseña*, deben coincidir.
* No puede existir un usuario con el mismo *Username*.
* La contraseña debe tener mínimo 8 caracteres, incluir letras, números y símbolos.

### 6. Validación de los campos al hacer login:

Para implementar la funcionalidad de iniciar sesión se han realizado las siguientes validaciones:

* Todos los campos del formulario deben ser informados.
* Las credenciales indicadas en el formulario deben coincidir con las de algún usuario registrado en la base de datos.

# Repositorios de código consultados

Lista de ejemplos de código consultados en github, gitlab, etc.

# Bibliografía consultada

<https://www.ciberaula.com/cursos/java/manejo_errores_excepciones_java.php>

<https://keepcoding.io/blog/errores-de-entrada-del-usuario/>

<https://developer.mozilla.org/es/docs/Learn_web_development/Extensions/Forms/Styling_web_forms>