



Universidad de Alcalá

Escuela Politécnica Superior

Universidad de Alcalá

Práctica Final

Arquitectura y Diseño de Sistemas Web y C/S

Página web de gestión de un cine

Laboratorio Jueves 10:00 – 12:00

Grado en Ingeniería Informática & Ingeniería en Sistemas de
Información – Curso 2019/2020

Marcos Barranquero Fernández – 51129104N

Daniel Manzano Estébanez – 03220212M

ÍNDICE

Instalación del proyecto y de la Base de Datos	3
Análisis del problema	4
Implementación	5
Modelo de datos	5
Tecnologías utilizadas	6
El control de sesiones	7
Manual de usuario	9
Parte del usuario.....	9
<i>inicio.jsp</i>	9
<i>eligeCine.jsp</i>	10
<i>cartelera.jsp</i>	10
<i>pelicula.jsp</i>	11
<i>elegirButaca.jsp</i>	12
<i>quieresRegistrarte.jsp</i>	13
<i>pagar.jsp</i>	14
<i>exitoPagando.jsp</i>	14
<i>registro.jsp</i>	15
<i>iniciandoSesion.jsp</i>	15
<i>cerrarSesion.jsp</i>	15
Parte del administrador	16
<i>administracion.jsp</i>	16
<i>Añadir información a la base de datos</i>	17
<i>Borrar información de la base de datos</i>	20
<i>Informes</i>	21
<i>Probar la seguridad de los Servlets (crearPeliculaSinSeguridad.jsp)</i>	22

INSTALACIÓN DEL PROYECTO Y DE LA BASE DE DATOS

Antes de comenzar, hay que instalar el proyecto y su base de datos correctamente. Para añadir la base de datos, hay dos opciones:

1. Copiando la base de datos directamente al directorio de NetBeans.

Para ello, hay que copiar la carpeta que se encuentra dentro de "1. Base de datos", llamada "BBDDPECLF", en la siguiente ruta:

```
C:\Users\danim\AppData\Roaming\NetBeans\Derby
```

Es la ruta donde se encuentran todas las bases de datos de Apache Derby. Tras copiarla e iniciar posteriormente NetBeans, debería aparecer en la lista de bases de datos.

2. Creando la base de datos como nueva.

En este caso, se debe crear una nueva base de datos en NetBeans con el nombre "BBDDPECLF", dejando los campos de usuario y contraseña en blanco. Para crear y rellenar las tablas, hay que ejecutar el contenido del archivo "1. Base de datos.sql" sobre una ventana de comandos.

Es recomendable seguir el primer método, ya que con el segundo el contenido de la base de datos es inferior, pues el archivo SQL solo insertará cines, salas, películas y actores, mientras que la base de datos proporcionada contiene también proyecciones y sitios (butacas) en los cines "Madrid" (tiene todas las películas) y "Alcalá de Henares", que han sido registradas desde la propia web, ya que las butacas se generan de manera automática al crear proyecciones. Hacer esto manualmente supondría una gran cantidad de sentencias SQL.

También se incluyen algunos clientes (cuyos datos de acceso aparecen en la tabla de esta página), comentarios y ventas de entradas para las películas proyectadas en el cine "Alcalá de Henares". La creación de proyecciones, la compra de entradas y el funcionamiento de los comentarios se explicarán detalladamente en el [manual de usuario](#) de este documento.

Hecho esto, se puede ejecutar el proyecto, que está en "2. Proyecto de NetBeans". Es posible que sea necesario ejecutar un "Clean and build" sobre el mismo si no se ejecutara correctamente la primera vez.

Correo	Contraseña
daniel.manzanoe@edu.uah.es	Contraseña01
marcos.barranquero@edu.uah.es	Contraseña02
fernando.perez@hotmail.com	Contraseña03
elena.suarez@gmail.com	Contraseña04
jesus.torres@yahoo.es	Contraseña05
teresa.rodriguez@gmail.com	Contraseña06
maría.parra@outlook.es	Contraseña07
jose.olmos@hotmail.es	Contraseña08
sara.garcia@gmail.com	Contraseña09
david.gomez@yahoo.es	Contraseña10

ANÁLISIS DEL PROBLEMA

Se desea crear un sistema web que permita la gestión de venta de entradas online de una cadena que posee 50 cines. Esta web contará con una cartelera donde se muestran las películas disponibles, que enlazarán a otra página, en la que se podrá ver su información detallada.

Dentro de dicha página, se podrá ver la información de la película y comprar entradas. Eligiendo una sesión disponible, se accederá a un mapa de la sala donde se podrán escoger los sitios que haya disponibles. Al seleccionar una butaca, quedará temporalmente reservada para garantizar al cliente que nadie se quede con ella mientras tramite la compra. Cuando confirme las butacas, se le pedirán sus datos personales y su tarjeta para completar el pago, y se le asignará un identificador de entrada con el que podrá acceder al cine. Si el cliente aborta la compra, o si tarda demasiado, las butacas que haya seleccionado volverán a estar disponibles al cabo de un tiempo.

Otra funcionalidad es la de los comentarios. Si un cliente se identifica con su correo, podrá ver y añadir comentarios a la película que ha visto. Estas sesiones de los clientes deben ser únicas, es decir, cada cliente se identifica de forma independiente en el navegador que esté usando.

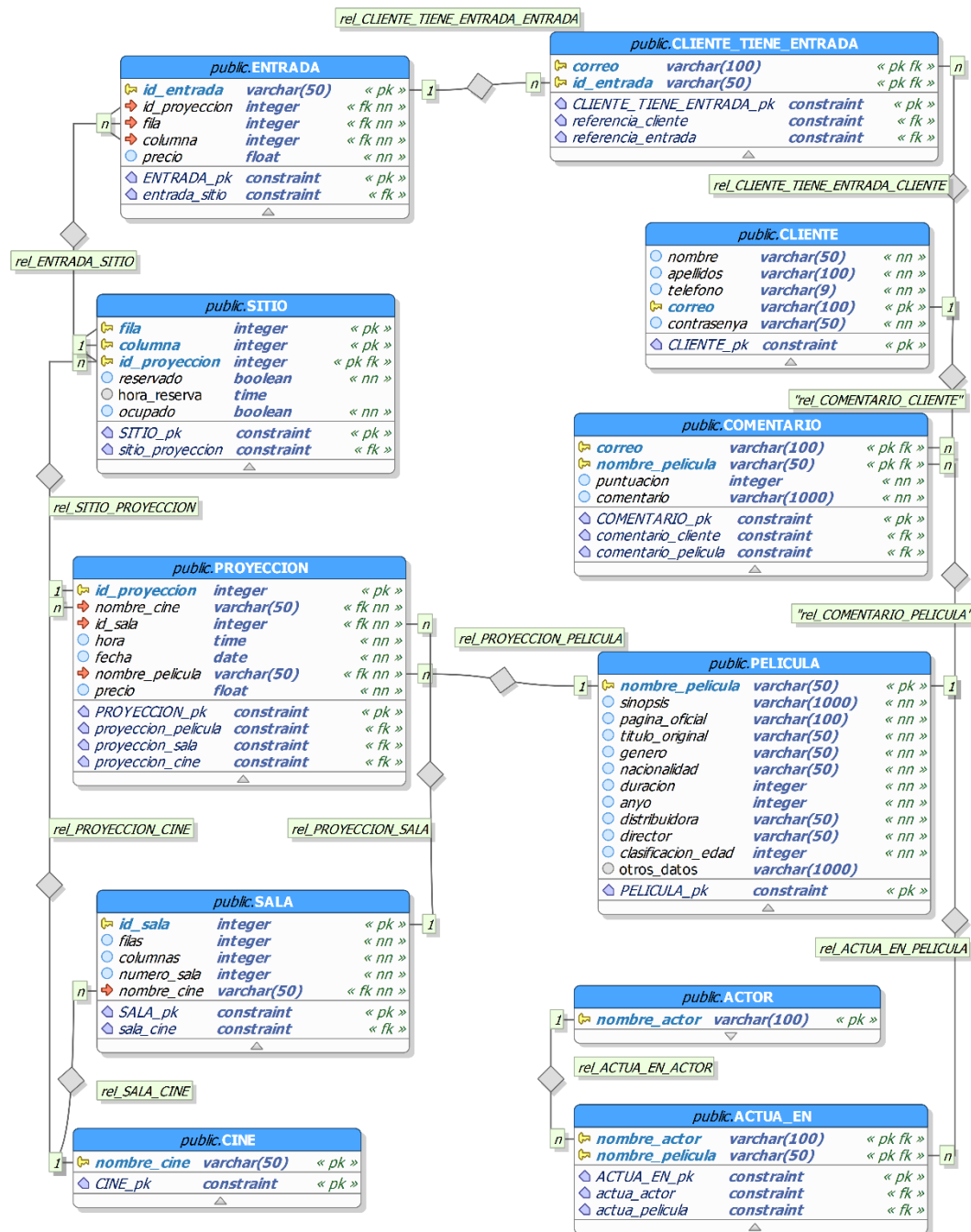
También deberá haber una parte para la administración que permitirá gestionar diferentes aspectos de la base de datos. Podrá añadir información, o borrar elementos, así como acceder a varios informes.

Todo esto debe ser implementado con el patrón Modelo-Vista-Controlador. En otras palabras, tienen que estar diferenciadas la vista que tienen los clientes del sistema del modelo de datos y de la parte lógica que actúa como intermediaria.

IMPLEMENTACIÓN

MODELO DE DATOS

Para crear una solución a este problema, se ha diseñado un modelo de datos con el que se dará cobertura a todos los aspectos del cine. Cuenta con los elementos básicos (cines, salas, películas...), pero también tiene una estructura de datos que permite gestionar la venta de entradas y la reserva de butacas. Este es el diagrama, adjunto también como "Diagrama Entidad-Relación.png".



Nota: En el diagrama, se muestran etiquetas como "public" que se corresponden a PostgreSQL. Al generar el código SQL, se han adaptado las sentencias al formato de Apache Derby.

Los **cines** y sus **salas** se registrarán en la base de datos, así como las **películas** y los diferentes **actores**. Estos dos últimos quedan relacionados con "**actua_en**", de forma que los actores podrán ser registrados una única vez en la base de datos y relacionados en aquellas películas donde actúen.

Se ha creado una tabla que representará las **proyecciones**. Estas estarán asociadas a una **sala** de un **cine** y a una **película**, y contará con una fecha y hora, que será cuando se proyecte la película. También se le ha añadido un campo con el precio, de tipo float para poder incluir decimales, para que los clientes sepan cuánto cuesta la entrada antes de comprarla.

Sobre esa tabla de proyección se crean, a su vez, **sitios**, que serán los asientos de la sala con información exclusiva de cada proyección. Estos se corresponden con una tabla que guardará la ubicación de la butaca (fila y columna), y si está reservada u ocupada. Cuando una butaca está marcada como reservada pero no ocupada, significará que alguien está tramitando su compra. Pero esa compra se podría abortar, por lo que, para desbloquearla, se guarda también la hora a la que ha sido reservada. Así, cuando se compruebe el estado de las butacas, si ha pasado una determinada cantidad de tiempo y no se ha vendido, quedará disponible de nuevo.

Cuando se tramite la compra, se generarán registros en la tabla **entrada**, donde se indicará un identificador único que será proporcionado al cliente para que pueda acceder al cine. Las compras podrán ser realizadas por usuarios registrados, aunque no es obligatorio. Estos usuarios serán almacenados en la tabla **cliente**, con su nombre, apellidos, teléfono, correo y contraseña. Si realiza la compra con su sesión activa, se reflejará también en una relación, "**cliente_tiene_entrada**", que asociará a los clientes con las entradas que compre. Así, se le permitirá dejar **comentarios** en aquellas **películas** que haya visto, los cuales quedarán registrados también en la base de datos. Si no ha querido registrarse, sólo quedará guardada la entrada, pero no su información.

TECNOLOGÍAS UTILIZADAS

Debido a la complejidad de la funcionalidad de la web, es evidente que no se puede desarrollar con páginas HTML. Por ello se han utilizado las tecnologías que proporciona Java para el ámbito web, que son los Servlets y las páginas JSP.

Para la parte visual de la web, se han utilizado páginas JSP. Estas páginas adaptan el contenido en función de los atributos que se encuentren en la sesión del cliente, o de algunos parámetros enviados por la página anterior. Por ejemplo, denegará el acceso a las páginas de administración si el usuario no se ha identificado como tal, o se ocultará la opción de comentar una película si ya hay un comentario de la misma persona. Este es un ejemplo.

```
<% if (session.getAttribute("usuario") == null ||  
        !session.getAttribute("usuario").equals("admin")) { %>  
    <!-- Se muestra un aviso de acceso denegado --%>  
<% } else { %>  
    <!-- Se muestra el contenido de la página --%>  
<% }%>
```

Para lograr el estilo de la página web, se ha empleado la biblioteca Bootstrap.

También se han empleado Servlets en la implementación. Estos se encargarán de ejecutar funcionalidades con más cantidad de código que, en el caso de esta práctica, no requieren mostrar nada de manera visual, como cuando se quieren insertar elementos en la base de datos.

Para gestionar algunos elementos, como los comentarios o las películas, se han creado clases de Java que contienen los mismos atributos que sus respectivas tablas en la base de datos. Esto permite encapsular toda su información en un único objeto, para facilitar su paso y manipulación entre las distintas páginas JSP y Servlets del proyecto.

Por último, existe una clase, BaseDatos, que contiene una gran cantidad de métodos que ejecutan distintas operaciones sobre la base de datos, que son utilizados por la gran mayoría de elementos de la web, para que puedan funcionar adecuadamente.

Todo esto, en conjunto, representa el patrón Modelo-Vista-Controlador. Se muestra una parte visual al cliente que interactúa con una base de datos, utilizando siempre como intermediarias otras clases que se encuentran en el lado del servidor, encargadas de ejecutar diferentes consultas sobre los datos y de proporcionar un resultado al cliente.

EL CONTROL DE SESIONES

Otro requisito del proyecto era el control de sesiones. Hay que identificar a los usuarios, pero de forma independiente, para que su información no se vea comprometida por otras conexiones en paralelo a la web.

Para lograr esto se han utilizado las sesiones que proporcionan las páginas JSP. Cuando un cliente se identifica, se establecerá el atributo "usuario" de la sesión con su correo. Así, cuando un contenido requiera que un cliente esté identificado, utilizará ese parámetro como validación.

También se guardan otros elementos en la sesión, como el cine elegido o la película seleccionada. Esto permitirá que un cliente pueda ver la cartelera de un cine en concreto o la información de una película sin tener que estar

indicando constantemente esa información, ya que las páginas de cartelera y de información de película son dinámicas: están preparadas para mostrar la información de cualquier cine o película que haya en la base de datos.

La página de la cartelera recibe como parámetro en la petición el nombre del cine, y la de película, a su vez, el nombre de la película. Si no encontraran los parámetros en la petición, comprobarían los atributos de la sesión, pues algún cliente podría acceder a ellas desde otro punto de la web mediante la URL o el historial. Si aun así no encontrara nada, avisará del error y permitirá volver a inicio. Si por el contrario recibe los parámetros en la petición, los establecerá automáticamente en la sesión, para ocasiones futuras.

Con esto, se logra ofrecer una experiencia personalizada al usuario y evitar las restricciones del protocolo HTTP, que no es orientado a estado.

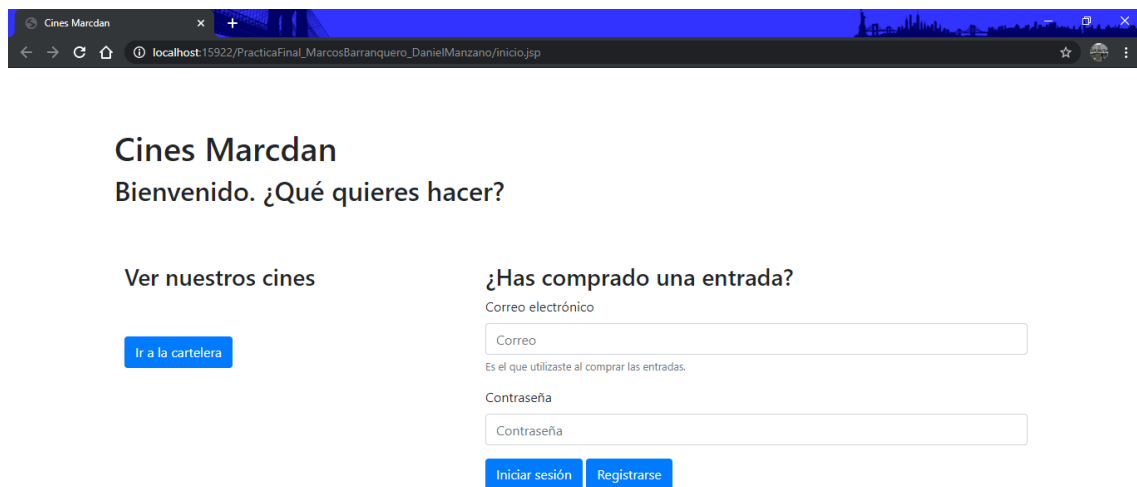
Otro aspecto que se consigue con las sesiones es la identificación de un administrador. Si en vez de especificar un correo y contraseña de un usuario normal se indican otros valores (**usuario y contraseña "admin"**), se establecerá la sesión como "admin", permitiendo el acceso a un menú de administración que queda bloqueado para cualquier otro usuario. En él se encuentran varias herramientas que serán explicadas en el [manual de usuario](#), al igual que todo el funcionamiento de la página web.

MANUAL DE USUARIO

Este manual ha sido creado teniendo en cuenta los datos que figuran en la base de datos adjunta con el proyecto, ya que cuenta con proyecciones ya registradas, así como algunos clientes, comentarios y entradas, como se ha explicado en el punto [instalación del proyecto y de la Base de Datos](#).

PARTE DEL USUARIO

inicio.jsp



Cines Marcdan

Bienvenido. ¿Qué quieres hacer?

Ver nuestros cines

[Ir a la cartelera](#)

¿Has comprado una entrada?

Correo electrónico

Es el que utilizaste al comprar las entradas.

Contraseña

[Iniciar sesión](#) [Registrarse](#)

Práctica Final - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2019

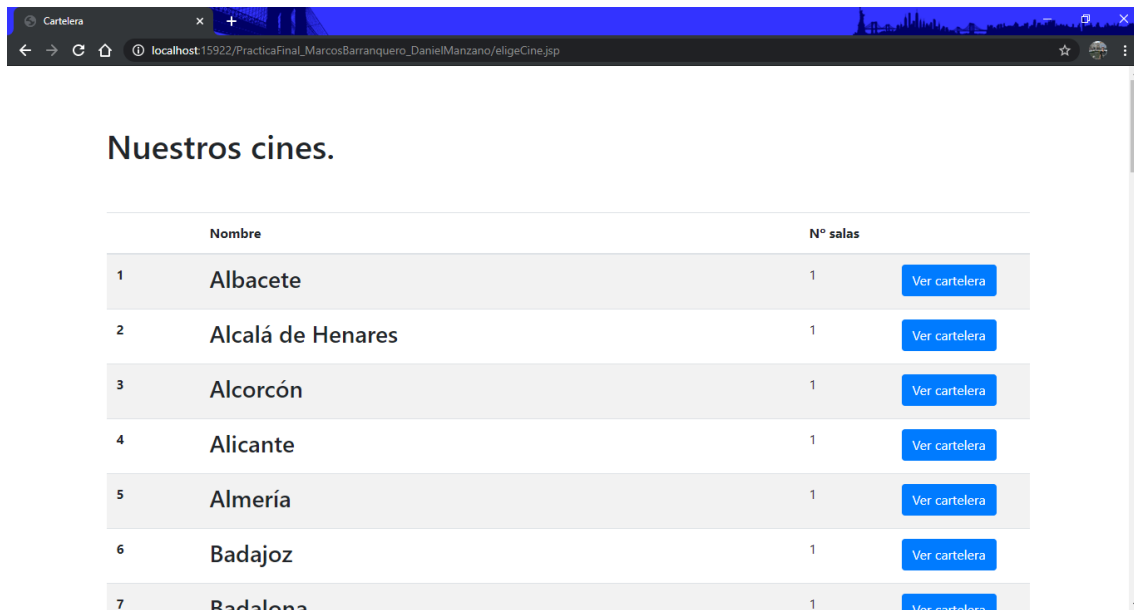
La página principal muestra un mensaje de bienvenida y permite varias opciones: elegir un cine (para ver su cartelera), iniciar sesión si ya tienes una cuenta o registrarte si quieres crearla.

La primera opción enlaza con la página `eligeCine.jsp`.

El inicio de sesión se corresponde con `iniciandoSesion.jsp`.

El registro enlaza con `registro.jsp`.

eligeCine.jsp

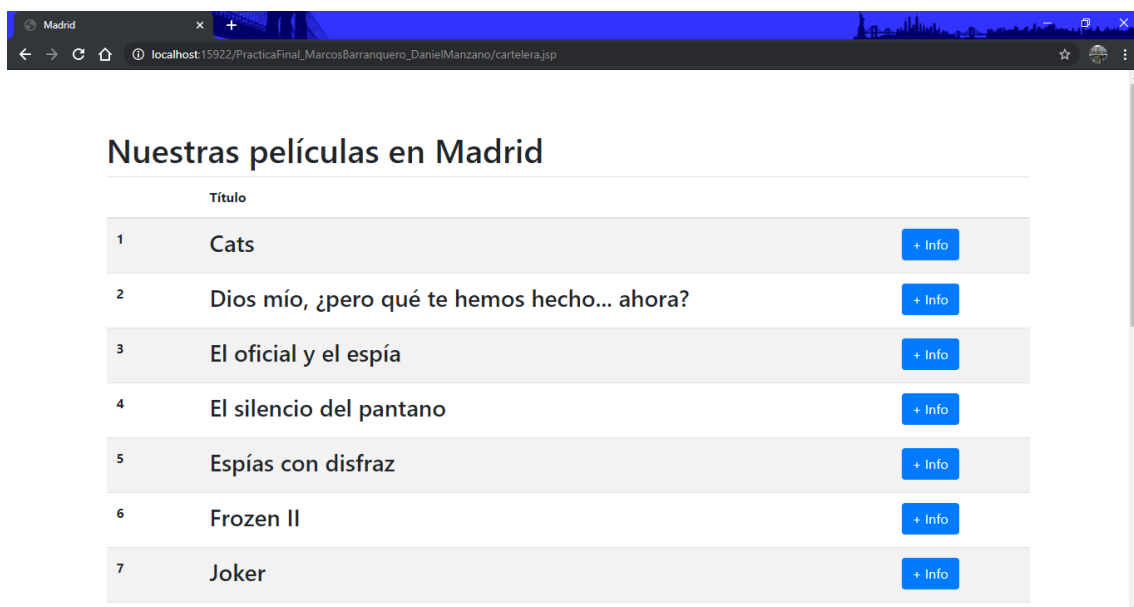


	Nombre	Nº salas	
1	Albacete	1	Ver cartelera
2	Alcalá de Henares	1	Ver cartelera
3	Alcorcón	1	Ver cartelera
4	Alicante	1	Ver cartelera
5	Almería	1	Ver cartelera
6	Badajoz	1	Ver cartelera
7	Badalona	1	Ver cartelera

En esta página se muestra una lista con los 50 cines y su cantidad de salas, junto a un enlace que mostrará las películas disponibles (cartelera.jsp).

Nota: Los cines que tienen proyecciones registradas son Alcalá de Henares y Madrid.

cartelera.jsp



	Título	
1	Cats	+ Info
2	Dios mío, ¿pero qué te hemos hecho... ahora?	+ Info
3	El oficial y el espía	+ Info
4	El silencio del pantano	+ Info
5	Espías con disfraz	+ Info
6	Frozen II	+ Info
7	Joker	+ Info

Aquí se mostrarán todas las películas que se proyectan en un cine. Si una película no se emite en un cine, no se mostrará como opción. Al elegir una película, se mostrará su información (película.jsp).

pelicula.jsp

Si yo fuera rico (2019)
No recomendada para menores de 16 años

Título original	Si yo fuera rico	País	España
Género	Comedia	Duración	98 minutos
Director	Álvaro Fernández Armero	Distribuidora	Telecinco Cinema
Sitio web	https://www.imdb.com/title/tt9010802/		

Sinopsis
Santi es un joven en apuros que, de la noche a la mañana, se vuelve rico. Muy rico. El problema es que lo "mejor" que podrá hacer es no contárselo a sus amigos. Y mucho menos a su pareja, claro.

Actores

- Alex García Fernández
- Alexandra Jiménez
- Antonio Resines

Compra tus entradas

	Fecha	Hora	Comprar
1	2020-01-16	20:30:00	Elegir butaca

No puedes ver ni publicar comentarios.
Debes iniciar sesión en la página principal.

[Ir a la página de inicio](#)

Práctica final - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2020

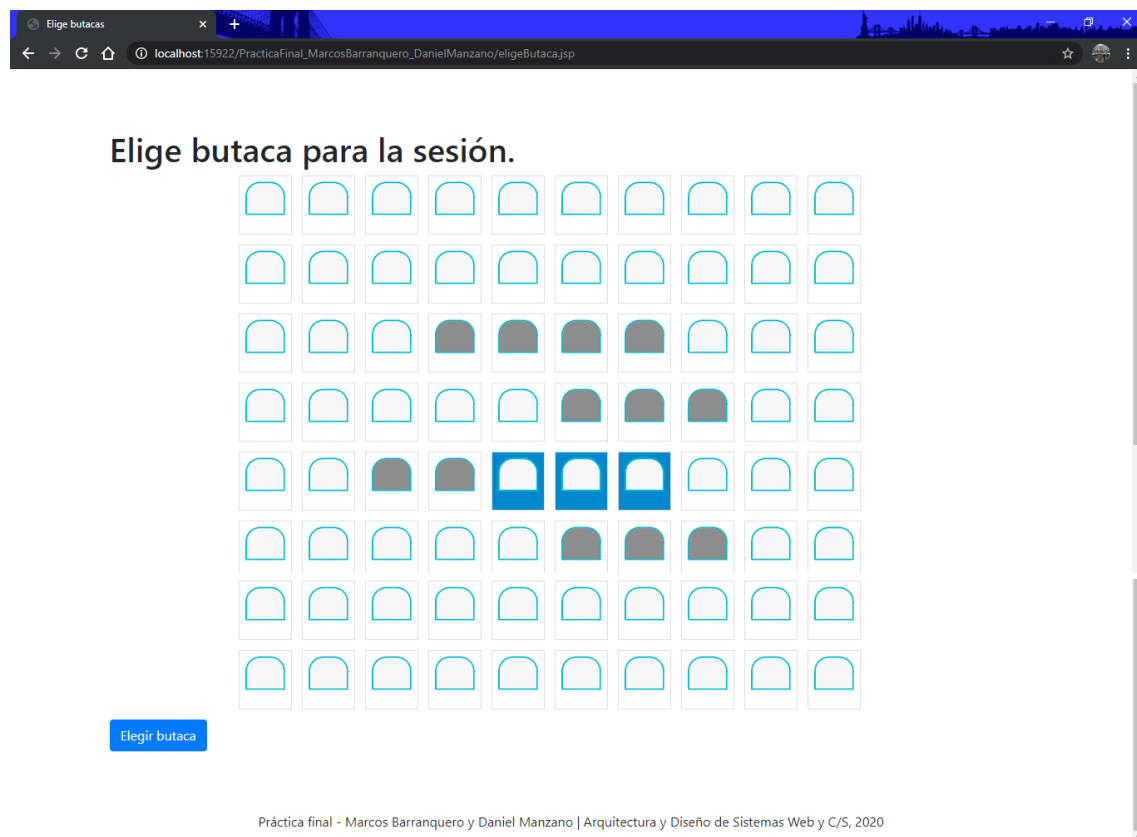
Aquí se puede ver la información de la película: su título, año, recomendación de edad, una tabla con varios datos relevantes, la sinopsis y el reparto de actores.

Tras ello, se mostrarán las sesiones disponibles (fecha y hora) de la película, donde se podrá comprar entradas (eligeButaca.jsp).

También hay un apartado de comentarios, que variará de la siguiente forma:

- Si no te has registrado, no se permitirá ver ni escribir comentarios.
- Si te has registrado, podrás ver los comentarios que haya publicados para esa película, que serán comunes a todos los cines. Por otro lado, habrá otras comprobaciones.
 - Si no has visto la película (no tienes entradas), no podrás comentar.
 - Si tienes entradas y no has comentado ya, podrás dejar un comentario con una puntuación del 1 al 5 y un texto.
 - Si ya has comentado, no podrás añadir otro comentario, pero podrás eliminarlo.

elegirButaca.jsp



Muestra un mapa con las butacas disponibles (tabla "sitio" en la base de datos). Las butacas disponibles se representan en blanco, y pueden ser seleccionadas, quedando resaltadas en azul. Las que se muestran en negro no son seleccionables, ya que alguien las ha comprado (o está en proceso de comprarlas).

Al pulsar en Elegir butaca, se redirigirá a quieresRegistrarte.jsp si no has iniciado sesión, o a pagar.jsp si ya hay una sesión activa.

Al hacer esto, las butacas elegidas se marcarán como reservadas, pero no vendidas, y se guardará la hora en la que esto sucede. Así, si finalmente no se marcan como vendidas y pasan 10 minutos, volverán a quedar libres en cuanto esta página se abra de nuevo, ya que es la encargada de realizar esa comprobación.

quieresRegistrarte.jsp

Antes de continuar...

Regístrate.

Nombre

Apellidos

Teléfono

Correo

Contraseña

[Registrarse y continuar](#)

[Volver al inicio](#)

Inicia sesión.

Correo electrónico

Contraseña

[Iniciar sesión](#)

O bien, continua sin registrarte.
No podrás dejar comentarios en las películas que veas.

[Continuar sin registrarse](#)

Práctica final - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2020

Si no estás registrado o con la sesión iniciada, se mostrará un formulario con un registro (enlaza al Servlet "AltaClienteYContinuar.java", que ejecuta el registro y establece la sesión y redirige a pagar.jsp) y un login (utiliza el Servlet "LoginYContinuar.java", y también establece la sesión y redirige a pagar.jsp).

Si se decide continuar sin registrarse, se envía directamente a pagar.jsp, pero no se creará un usuario, por lo que no será posible dejar comentarios en la película.

pagar.jsp

Fila	Columna	Precio
5	5	6.3
5	6	6.3
5	7	6.3

Método de pago

Número de tarjeta

Fecha de caducidad (MM-AA)

CVV

[Pagar](#)

Práctica final - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2020

Para finalizar el pago, se muestra una tabla con el resumen de las entradas seleccionadas, con su precio. Se pedirán los datos de una tarjeta, utilizando expresiones regulares para verificar el formato de los datos. Al pulsar en pagar, se accede a `exitoPagando.jsp`.

exitoPagando.jsp

Fila	Columna	Precio	Id
5	5	6.3	E1655
5	6	6.3	E1656
5	7	6.3	E1657

[Volver al inicio](#)

Práctica final - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2020

Muestra las entradas con su ID, para poder acceder al cine. Con esta página termina el proceso de compra, y se ofrece volver a inicio (`inicio.jsp`).

registro.jsp

Registro.
Introduce tus datos.

Nombre

Apellidos

Teléfono

Correo electrónico

Contraseña

[Insertar](#) [Reestablecer el formulario](#)

[Volver al inicio](#)

Práctica final - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2020

Contiene un formulario, con expresiones regulares para verificar el contenido, que permite a los clientes darse de alta en el sistema web.

iniciandoSesion.jsp

Se encarga de hacer la comprobación de los datos del usuario y establecer la sesión. Si los datos son correctos, redirige automáticamente a la página de inicio, por lo que apenas se puede percibir esta página. Si estuvieran mal, se muestra un mensaje de advertencia junto con un botón que permite volver a inicio.

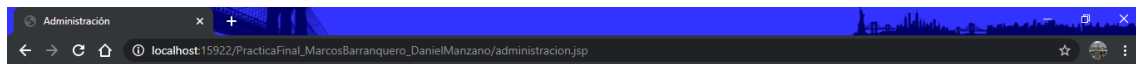
cerrarSesion.jsp

Su funcionamiento es parecido al JSP anterior. Elimina el atributo "usuario" de la sesión y redirige al inicio. Para cerrar sesión, hay que utilizar el botón que aparece en la página de inicio cuando un usuario está identificado.

PARTE DEL ADMINISTRADOR

Para identificarse como administrador, hay que introducir la palabra "admin" tanto en el campo de usuario como en el de contraseña de la página de inicio. Al pasar por `iniciandoSesion.jsp`, se establecerá la sesión y se redireccionará a `administracion.jsp`.

Si se intenta acceder a cualquier página de administración sin estar identificado, se mostrará un aviso de acceso denegado.

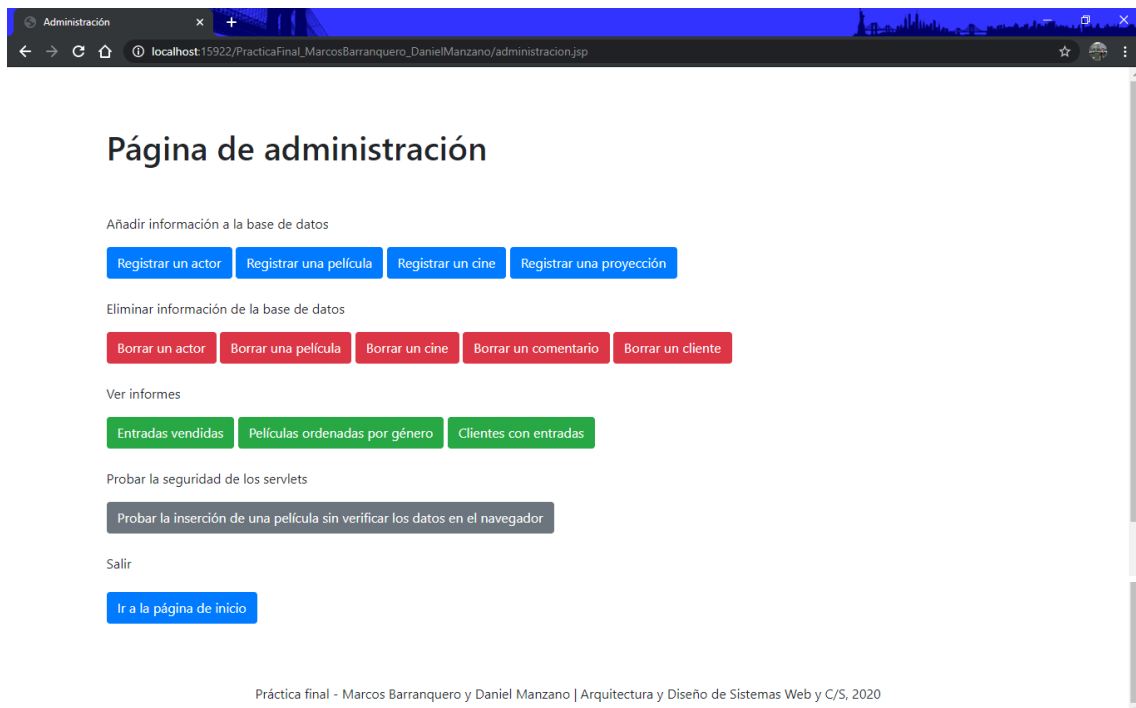


Acceso denegado.
No eres un administrador.

[Volver a inicio](#)

Práctica final - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2020

administracion.jsp

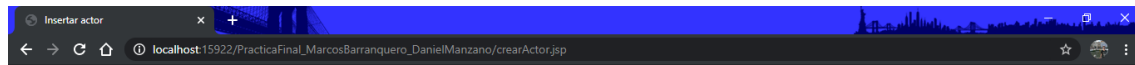


Práctica final - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2020

Muestra un menú con todas las opciones especiales que puede realizar un administrador: inserciones de datos, eliminaciones de registros, visualizar informes y una formulario que no tiene expresiones regulares para probar la seguridad de los Servlets, ya que el resto de páginas verifica los datos en el lado del cliente (expresiones regulares) y del servidor (comprobaciones de los datos recibidos en los Servlets).

Añadir información a la base de datos

crearActor.jsp



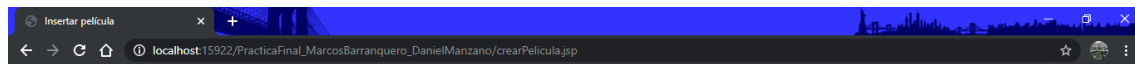
Introduce los datos del actor.

Nombre del actor

Práctica final - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2020

Permite registrar un actor en la base de datos, para luego poder añadirlo a las películas a la hora de crearlas. Utiliza el Servlet CargarActor.java.

crearPelicula.jsp



Introduce los datos de la película.

Nombre de la película:

Síntesis:

Página web oficial:

Título original:

Género:

Nacionalidad:

Duración: minutos

Año:

Distribuidora:

Director:

Clasificación por edad: (0 = todos los públicos) No recomendada para menores de años

Otros datos:

Actores de la película:

Elija, de los actores de la base de datos, aquellos que participen en la película.

Nota: debe haberlos insertado en la BBDD previamente.

Actores de la BBDD

Alex Garcia Fernandez
Alexandra Jimenez
Ana de Armas
Antonio Resines

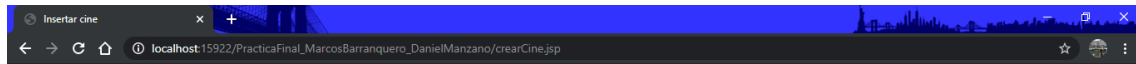
Actores seleccionados

--

Práctica final - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2020

Contiene un formulario que permite registrar películas en la base de datos. Pide todos los parámetros necesarios, y muestra una caja con todos los actores que hay en la base de datos, que se añaden a otra al hacer doble clic sobre ellos, para quedar también registrados en la relación "actua_en" de la base de datos. La carga de la película se realiza mediante el Servlet "CargarPelicula.java".

crearCine.java



Introduce los datos del cine.

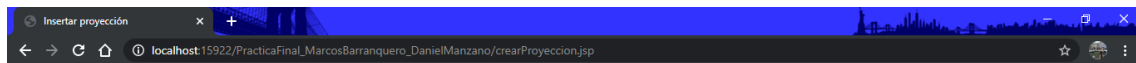
Nombre del cine:

Nº de salas:

Práctica final - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2020

Registra un cine con un nombre. También hay que indicar el número de salas, de forma que al crearlo se insertan también sus salas en la base de datos automáticamente. En este caso, el Servlet utilizado es "CargarCine.java".

crearProyeccion.jsp y crearProyeccionHorario.jsp



Introduce los datos de la proyección.

Cine:

Práctica final - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2020

Con las páginas anteriores se podían crear películas y cines, pero esas películas deben tener una proyección con fecha, hora, precio, etcétera. Por ello es necesario registrar proyecciones. En la primera página, "crearProyeccion.jsp", podemos elegir el cine donde se quiere registrar la proyección. Al pulsar Insertar, seremos enviados a "crearProyeccionHorario.jsp".

Introduce los datos de la proyección.

Cine: Madrid

Película elegida:
Dios mío, ¿pero qué te hemos hecho... ahora?
El oficial y el espía

Sala elegida:

Fecha:

Hora:

Cuarto de hora:

Precio:

Práctica final - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2020

Aquí podremos ver la confirmación del cine donde queremos registrar la proyección, seguido de una lista con las películas disponibles en la base de datos, una lista desplegable con las salas del cine y campos para indicar la fecha y hora, así como el precio que costarán esas entradas.

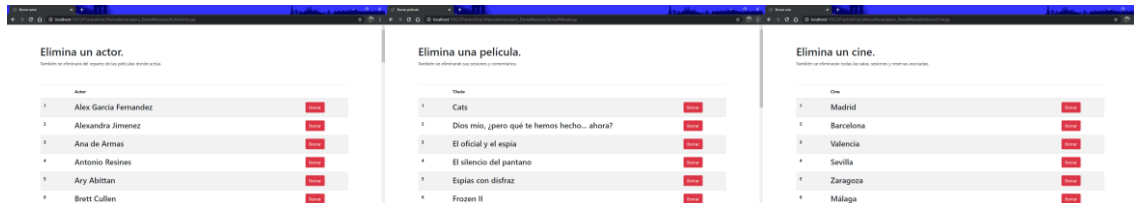
Al pulsar en insertar, mediante el Servlet "CargarProyeccion.java", al igual que con todas las otras páginas de inserciones, se validarán los datos y se ejecutará un método de la clase "BaseDatos.java" que insertará el registro. La diferencia es que este método, "insertarProyeccion", ejecuta a su vez otro, "crearSitio", que se encargará de insertar tantos sitios como butacas tenga la sala de la proyección. Es decir, no solo se crea una proyección, sino que también todos los registros necesarios en la tabla "sitio", que son necesarios para los clientes a la hora de comprar entradas.

Una vez insertado una proyección, se puede comprobar que aparece la película en la cartelera de ese cine si antes no estaba, y que dentro de la información de la película aparecen los datos de esa sesión.

Borrar información de la base de datos

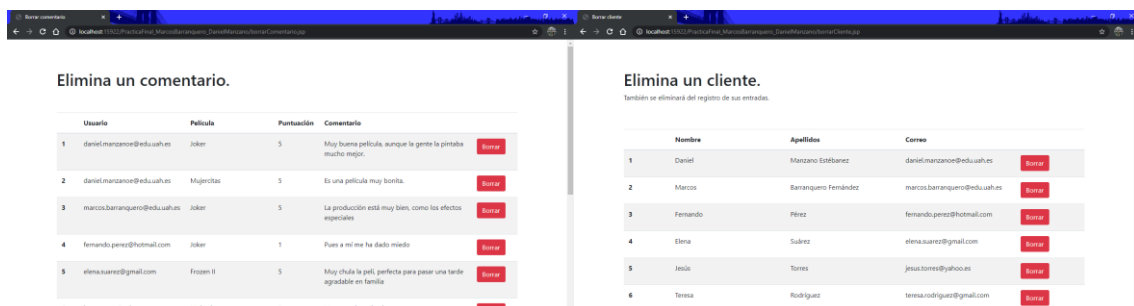
Para evitar problemas con la eliminación de registros en la base de datos, las claves foráneas han sido creadas con la sentencia "ON DELETE CASCADE".

[borrarActor.jsp](#), [borrarPelícula.jsp](#) y [borrarCine.jsp](#)



Su contenido es similar. Muestra una lista con el nombre del actor, película o cine junto a un botón de borrar. Al pulsar un botón, se borrará de la base de datos. Emplean los Servlets "BorrarActor.java", "BorrarPelícula.java" y "BorrarCine.java", respectivamente.

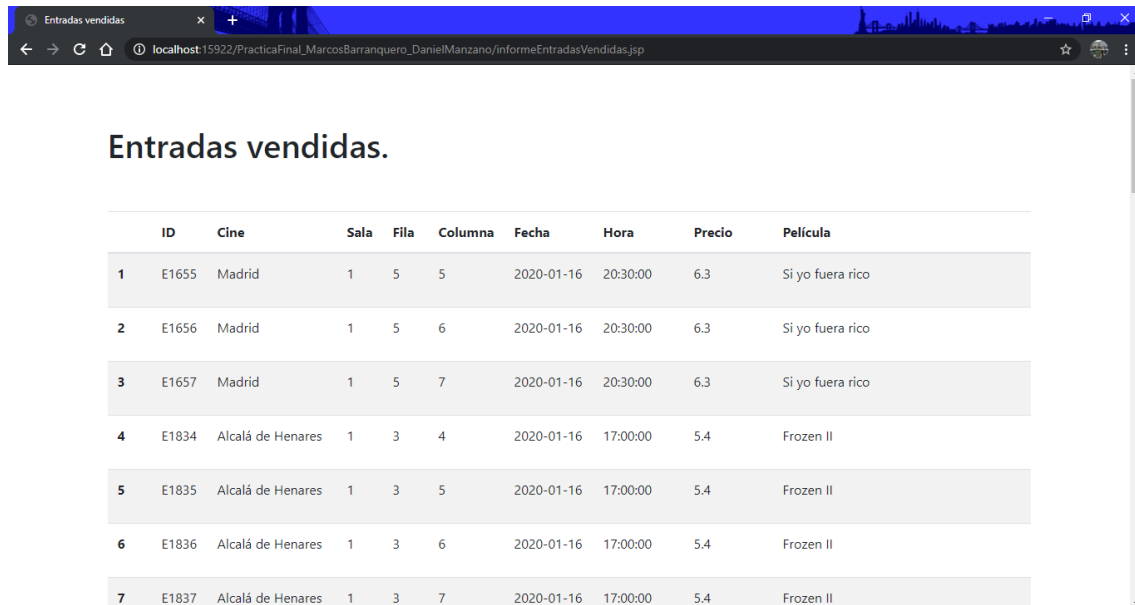
[borrarComentario.jsp](#) y [borrarCliente.jsp](#)



Su funcionamiento es el mismo que el de las tres páginas anteriores, pero en este caso se muestran más datos. Se podrían considerar también como informes de comentarios y de clientes por los detalles que dan. Utilizan los Servlets "BorrarComentario.java" y "BorrarCliente.java".

Informes

informeEntradasVendidas.jsp

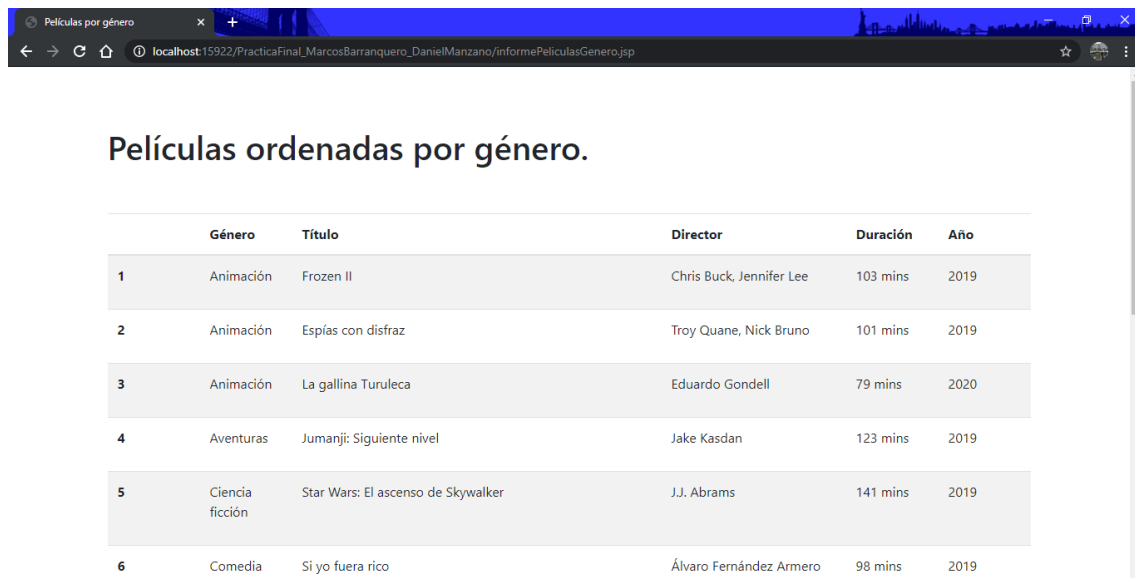


Entradas vendidas.

	ID	Cine	Sala	Fila	Columna	Fecha	Hora	Precio	Película
1	E1655	Madrid	1	5	5	2020-01-16	20:30:00	6.3	Si yo fuera rico
2	E1656	Madrid	1	5	6	2020-01-16	20:30:00	6.3	Si yo fuera rico
3	E1657	Madrid	1	5	7	2020-01-16	20:30:00	6.3	Si yo fuera rico
4	E1834	Alcalá de Henares	1	3	4	2020-01-16	17:00:00	5.4	Frozen II
5	E1835	Alcalá de Henares	1	3	5	2020-01-16	17:00:00	5.4	Frozen II
6	E1836	Alcalá de Henares	1	3	6	2020-01-16	17:00:00	5.4	Frozen II
7	E1837	Alcalá de Henares	1	3	7	2020-01-16	17:00:00	5.4	Frozen II

Muestra todos los registros de la tabla de las entradas y los combina con información de otras, como el nombre de la película, el cine y el número de sala. El informe es creado con el método "getEntradasVendidas", de "BaseDatos.java". Los datos de cada registro están encapsulados mediante una clase de Java, llamada "InformeEntradas.java".

informePelículasGenero.jsp



Películas ordenadas por género.

	Género	Título	Director	Duración	Año
1	Animación	Frozen II	Chris Buck, Jennifer Lee	103 mins	2019
2	Animación	Espías con disfraz	Troy Quane, Nick Bruno	101 mins	2019
3	Animación	La gallina Turuleca	Eduardo Gondell	79 mins	2020
4	Aventuras	Jumanji: Siguiente nivel	Jake Kasdan	123 mins	2019
5	Ciencia ficción	Star Wars: El ascenso de Skywalker	J.J. Abrams	141 mins	2019
6	Comedia	Si yo fuera rico	Álvaro Fernández Armero	98 mins	2019

Contiene una tabla donde aparecen los datos más importantes de todas las películas, que se encuentran ordenadas alfabéticamente por género. Utiliza el método "getPelículasGenero" de "BaseDatos.java", que devuelve un ArrayList de objetos del tipo "Película".

informeClientesEntrada.jsp



	Nombre	Apellidos	Correo	Total de entradas
1	Daniel	Manzano Estébanez	daniel.manzanoe@edu.uah.es	6
2	Elena	Suárez	elena.suarez@gmail.com	5
3	Fernando	Pérez	fernando.perez@hotmail.com	3
4	Jesús	Torres	jesus.torres@yahoo.es	5
5	José	Olmos	jose.olmos@hotmail.es	4
6	Marcos	Barranquero Fernández	marcos.barranquero@edu.uah.es	4
7	María	Parra	maria.parra@outlook.es	4

Mediante el método "getClientesConEntrada" de "BaseDatos.java", se obtiene una lista con los datos de los clientes que tienen entrada junto con el número total de entradas compradas. Cada registro encapsula sus datos en un objeto de la clase "InformeClientes.java".

Probar la seguridad de los Servlets (crearPeliculaSinSeguridad.jsp)

Se trata de una página similar a la de crear películas, pero se han eliminado las expresiones regulares y las verificaciones de los campos del formulario. El objetivo es comprobar que, si se introducen datos equivocados, el Servlet los revisará y redireccionará a una página con un error ("errorDatosMal.html"), que tiene un botón que envía al navegador a la página anterior, por lo que sirve para cualquier error y los campos del formulario se conservan escritos (dependiendo del navegador).

Esta comprobación podría hacerse con cualquier página de inserción, ya que todos los Servlets verifican los datos que reciben.