

MEMORIA

Estos son los módulos principales que forman esta entrega, cada módulo puede estar formado por varios ficheros que cooperan para cumplir su funcionalidad

módulos en el directorio includes(en php)

En este directorio, podemos encontrar muchos ficheros con partes de páginas html comunes y/o funciones que utilizaremos de forma recurrente en el resto de módulos.

footer.php

En este fichero tenemos el pie de página fijo en todas nuestras páginas

head.php

En este fichero encontramos todos las librerías (principalmente javascript) que usaremos en nuestros ficheros

header.php

En este fichero tenemos la barra de navegación donde encontramos los links para acceder al resto de páginas web y el saldo del usuario.

Cabe destacar que cambia dependiendo de si el usuario ha hecho login o no.

lateral.php

Aquí tenemos el panel lateral izquierdo común a todos los ficheros, en él podemos hacer queries rápidas por categoría de las películas, que se pueden hacer también en index.

Además en este panel, podemos ver la funcionalidad que muestra un número aleatorio de usuarios conectados cada 3 segundos.

Esta funcionalidad ha sido hecha mediante AJAX, que cada 3 segundos llama a un fichero generator.php, el cual genera dicho número aleatorio de usuarios.

flogin.php

En este fichero hemos encapsulado una parte de la funcionalidad del login, que detallaremos más adelante en el apartado **login**

fcesta.php

En este módulo se encuentran funciones que se encargan de manipular aspectos de la cesta de la compra. Proporciona funcionalidad como añadir un elemento a la cesta, eliminarlo de la cesta, calcular el precio total de la cesta, comprobar que la compra sea válida (evita que se añadan productos ya comprados a la cesta), corregir posibles discrepancias en la cesta (que haya productos ya comprados) y limpiar los datos de la cesta

utils.php

Este módulo proporciona funciones que se encargan de manejar distintos aspectos del sistema. Principalmente, proporciona funcionalidad sobre el historial de los usuarios (preparar el fichero xml, añadir un producto al historial, comprobar si un producto está en el historial e imprimir el historial) así como funcionalidad de gastar el saldo de un usuario (útil al realizar las compras), e imprimir la información de las películas (útil para que esto se haga de forma estandarizada). El propósito de este módulo es proporcionar funciones que eviten duplicar código para mantener mejor el sistema.

index

En esta página index.php podemos encontrar un buscador de películas en función de si es por título, categoría o director seguida del catálogo principal de películas.

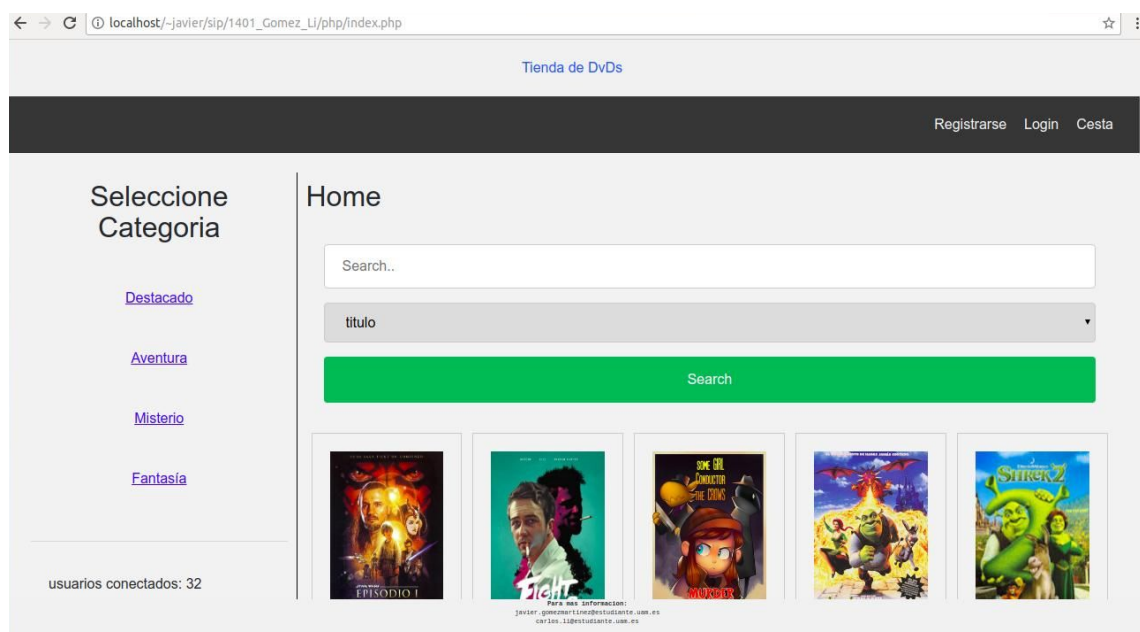
En el buscador podemos escribir las queries de forma case insensitive y con cualquier subcadena que pueda estar contenida en la película buscada bajo el filtro marcado. Tras escribirla, y darle al botón, nos aparecen debajo las películas que cumplen con dicho filtro. Si no escribimos ninguna búsqueda, nos muestra por defecto las películas destacadas.

información técnica:

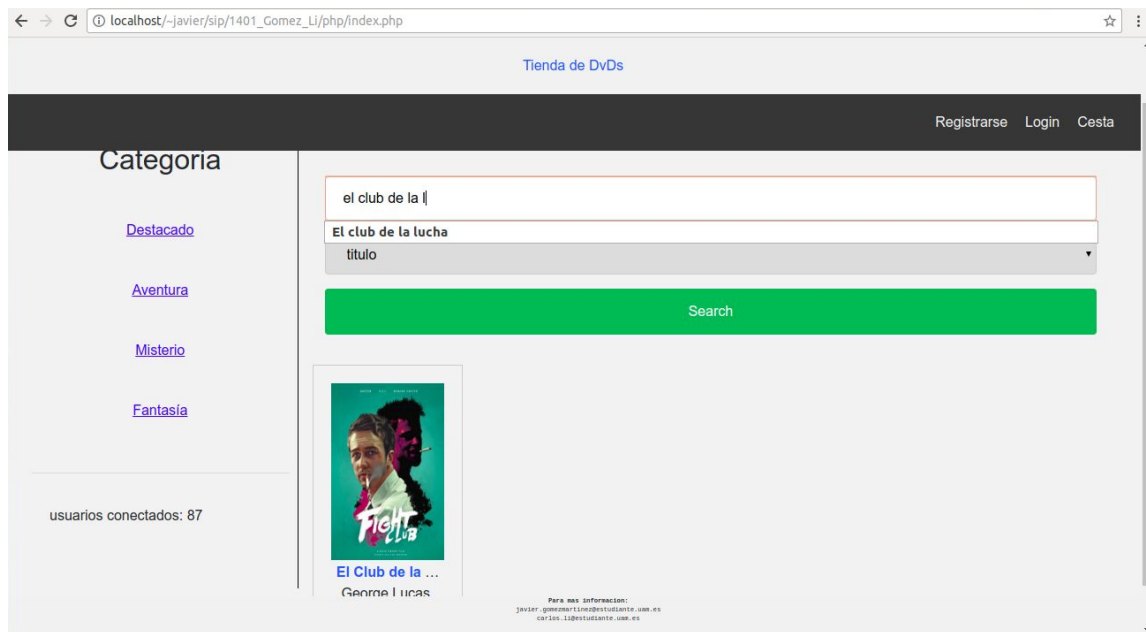
La funcionalidad principal de esta página radica en las queries de películas, es decir, mediante un form en la barra de búsqueda, mandamos una petición http post a sí misma para que busque en el fichero ../xml/catalogo.xml mediante la función xpath, las películas que cumplen la query pedida.

Como añadido, hemos hecho que la query con xpath sea más inteligente mediante el uso de funciones de xpath. La primera que usamos es translate() para convertir todo lo que lee del fichero catalogo.xml en minúsculas y, que de esta forma, junto a la query del usuario convertida también en minúsculas, podamos darle una funcionalidad case insensitive a la query.

También hemos hecho uso de la función contains, para que no tengamos que escribir el campo deseado completamente, sino que con escribir un fragmento del texto sea capaz de encontrarlo.



En esta imagen vemos el index si se entra sin hacer queries



En esta vemos el funcionamiento de la query.

Si escribimos un título a medias y en minúsculas o mayúsculas por igual, encuentra la película

Producto

Esta página muestra la información de un producto.

Comenzando con el título del producto y siguiendo con una imagen del mismo, una descripción, la fecha de publicación, su género, su precio en la tienda, el director de la película y los actores del reparto

Desde esta página el usuario, habiéndose identificado o no, puede añadir el producto a su cesta de la compra. En caso de que intente añadir un elemento que ya se encuentra en la cesta de la compra, el sistema lo detectará y, sin añadirlo a la cesta, notificará al usuario de esto.

En caso de que el añadir al carrito fuera bien, el usuario continuará en esa misma página y se le notificará de que se añadió el producto correctamente al carrito.

Adicionalmente, si el usuario ha sido registrado, el sistema puede comprobar si el producto que intenta comprar ya lo compró con anterioridad, en caso afirmativo, tampoco lo añadirá a la cesta de la compra y notificará al usuario de este evento.

Esto lo hemos hecho pensando en que nuestra tienda de películas proporciona copias digitales a los usuarios y por tanto no tiene sentido que el usuario pague dos veces por el mismo producto (de manera similar a como lo hacen plataformas ya existentes como Steam) por tanto, es el servidor el que se encarga de asegurarse que un usuario no tenga más de una copia de un mismo objeto.

información técnica:

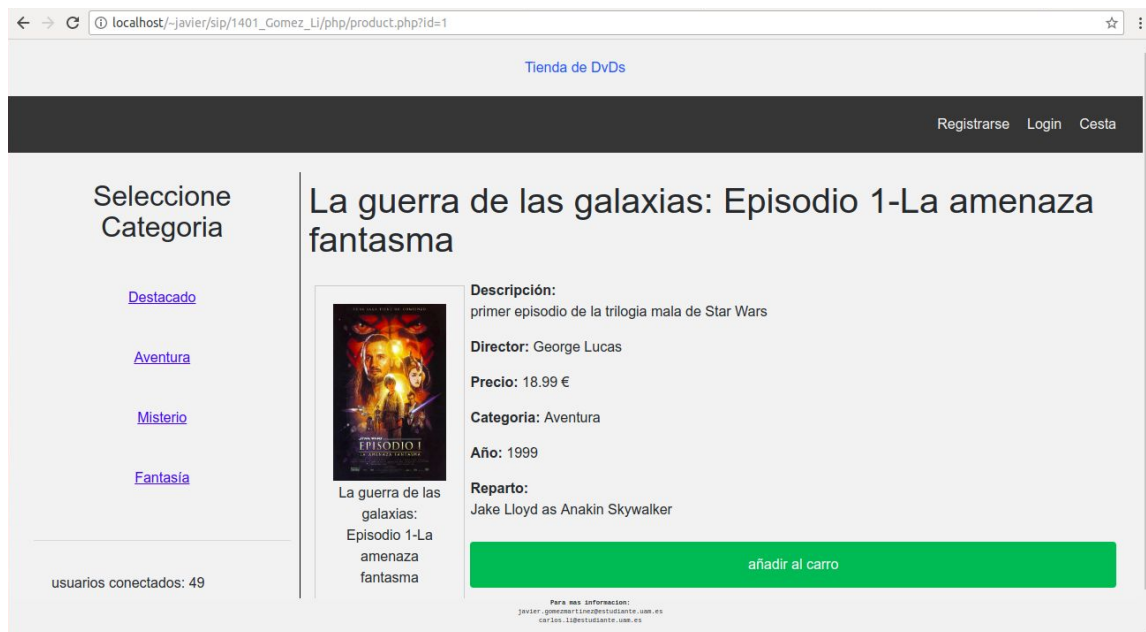
Esta página es generada por un script php que espera recibir, mediante el método GET, el id de una película. Este id es único y por tanto sirve para identificar unequivocamente a las películas que se encuentran almacenadas en el fichero xml “catálogo”.

Lo primero que se hace es comprobar que efectivamente se ha llegado a esta página con un id válido en el método get. Si no es así, mostrará un error pues el producto que se buscó no existía.

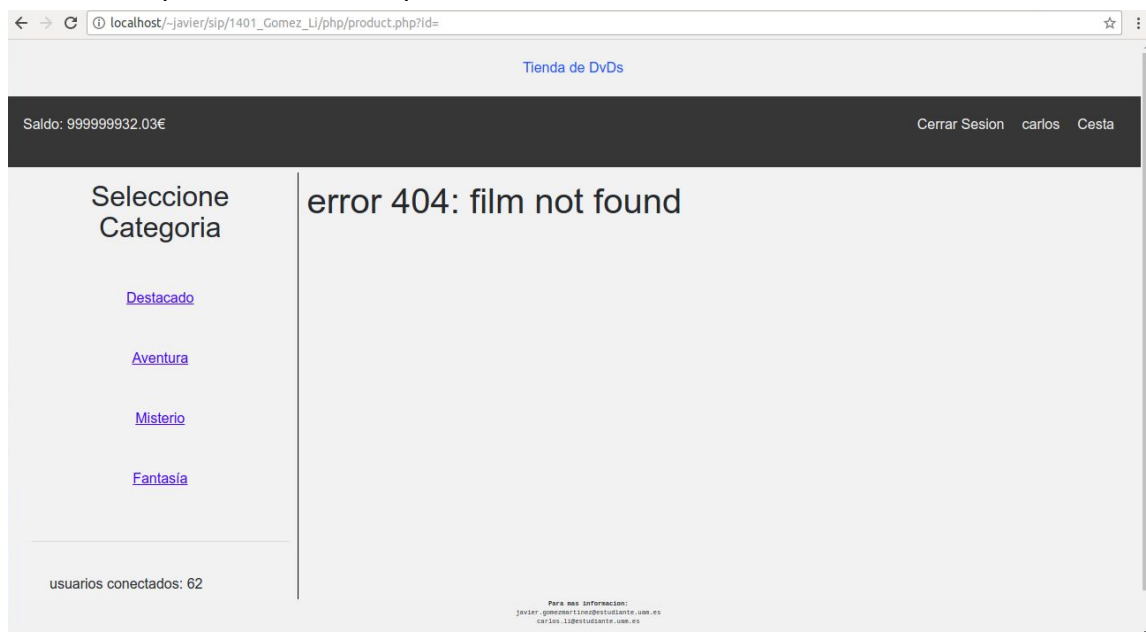
Si el id proporcionado es válido, entonces mostrará los contenidos de la película tal y como vienen especificados en el xml. Al final de esto, aparece el botón “añadir al carrito” que de ser pulsado envía un formulario al servidor indicando que se pretende comprar este producto. Este formulario se enviará mediante el método POST un campo oculto indicando que se pretende añadir al carrito el producto. El servidor entonces tratará de añadir el producto al array de SESSION[‘cesta’] (siempre y cuando se cumpla que no está ya en el carrito y que el usuario no ha comprado el producto con anterioridad).

En este array, se van guardando los ids de las películas que se vayan añadiendo al carrito.

Este php utiliza funciones externas add_to_cesta (includes/fcesta.php) y is_valid_compra (includes/utis.php) que respectivamente añaden la funcionalidad de añadir un id a la cesta (asegurándose de que no está ya en la cesta) y de comprobar que el usuario no haya comprado ya un producto.



captura de pantalla de una página de producto estándar. los datos son rellenados con la información particular de cada película



captura en caso de que se trate de acceder a la página de una película que no existe.

Registro

En este módulo, que forma principalmente el fichero register.php junto a un fichero ../javascript/register.js, lo que hacemos es registrar los datos de un usuario nuevo y crearlo, a menos que se haya detectado algún tipo de error.

información técnica:

Siendo más precisos, lo que hace el register.php es mostrar el formulario a rellenar, donde se hacen varios controles.

El primer control, se encuentra en el cliente, es decir, en html hemos incluido el campo required en todos los campos, de esta forma, el usuario no puede dejar ninguno vacío. Por otra parte, en javascript hemos creado 2 funciones principales para controlar el resto de funcionalidades:

check():

Es la función que usamos para comprobar en vivo si la contraseña y su repetición coinciden, antes de enviarlos al servidor.

strength():

Es la función que usamos para medir, mediante expresiones regulares, la fortaleza de una contraseña.

Por otra parte, en php hemos implementado las comprobaciones que haría el servidor para validar la creación de dicho usuario, que consiste principalmente en ver si el usuario existe, y en caso de que no, crear la carpeta para dicho usuario, donde almacenará en el fichero datos.dat los campos rellenados por el usuario escritos línea a línea (la contraseña está encriptada en md5)

The screenshot shows a web browser window with the address bar displaying 'localhost/~javier/sip/1401_Gomez_Li/php/register.php'. The page title is 'Tienda de DVDs'. The main content area is divided into two sections. On the left, under the heading 'Seleccione Categoría', there are four links: 'Destacado', 'Aventura', 'Misterio', and 'Fantasia'. On the right, under the heading 'Register', there is a form with the following fields: 'Username' (containing 'Javier'), 'E-mail' (containing 'javier.gomezmartinez@estudiante.uam.es'), and 'Contraseña' (masked with dots). Below the password field, there is a password strength indicator showing a blue bar and five empty boxes. The text 'Repita tu contraseña' is visible below the strength indicator. At the bottom of the page, there is a status bar that says 'usuarios conectados: 59'.

En esta imagen vemos cómo se rellenan los campos del registro y principalmente ver el funcionamiento del medidor de fuerza de la contraseña.

← → ↻ localhost/~javier/sip/1401_Gomez_Li/php/register.php

Tienda de DVDs

Registrarse Login Cesta

[Aventura](#)

[Misterio](#)

[Fantasía](#)

usuarios conectados: 33

E-mail

javiergomez_1996@hotmail.com

Contraseña

.....

Repite tu contraseña

.....

contraseñas no coinciden

Cuenta Bancaria

Tu cuenta bancaria...

Registrarse

Para mas informacion:
javier.gomez@inecibn.com.es
carlos.li@estudiante.um.es

En esta imagen podemos ver otra funcionalidad como la de detectar en vivo si la contraseña y su repetición coinciden o no.

Login

En este módulo, compuesto principalmente por login.php y includes/flogin.php , y se encarga de iniciar la sesión del usuario, comprobando que la cuenta existe, y que la contraseña es correcta.

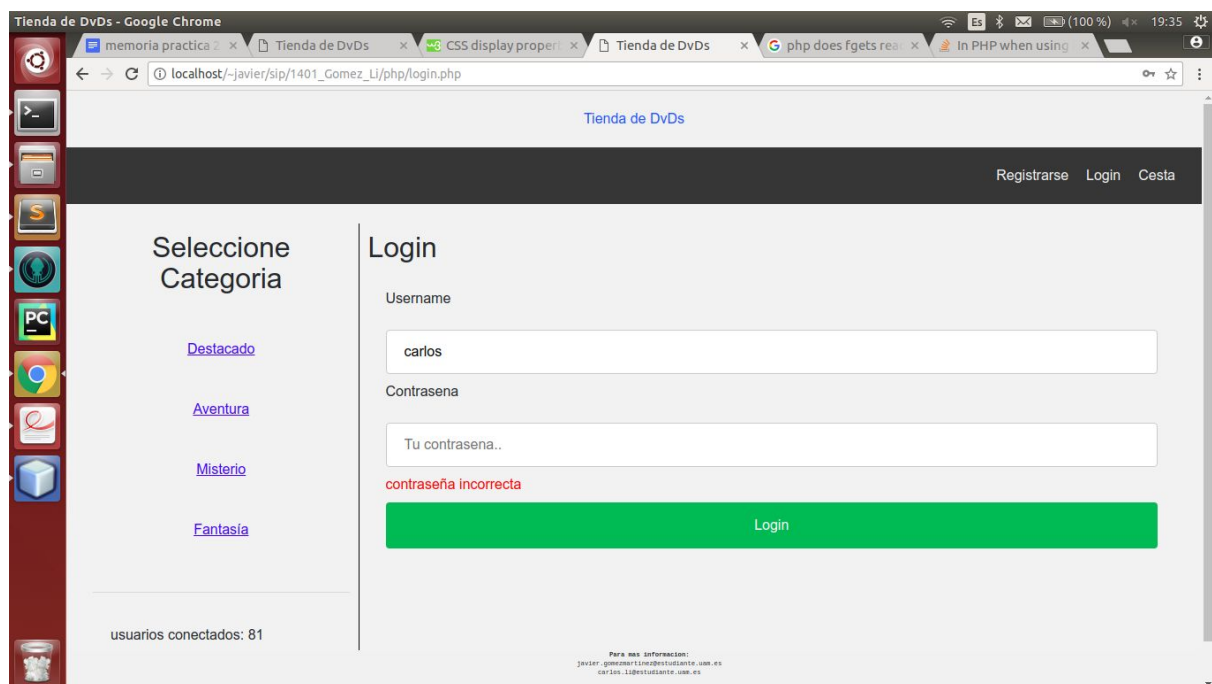
información técnica:

El fichero login.php incluye el formulario que pide el usuario y la contraseña.

En el usuario, por defecto usamos una cookie que creamos para el último usuario que ha iniciado sesión para rellenar el campo, de esta forma, si quiere volver a hacer login, le ahorramos escribir.

Por otra parte, el resto de comprobaciones las hacemos en el fichero flogin.php, donde comprobamos que el usuario escrito existe (existe su carpeta de usuario) y que la contraseña coincide con la cifrada en md5 dentro del fichero de usuario datos.dat.

En el caso en que haya algún error, imprimimos debajo del formulario, el error de validación que ha sucedido. En caso de que todo sea correcto, se crea una sesión para el usuario con su nombre de usuario y su saldo y se le redirige a index.php, donde aparece como loggeado.



En esta imagen vemos actuando la cookie (que escribe por defecto en el campo username el último usuario que hizo login correctamente), y como, si se escribe la contraseña incorrectamente notifica del error. En caso de que se haga login correctamente redirigiría la url al index.php con la sesión iniciada.

Cesta

Esta página, accesible tanto si el usuario ha hecho Login como si no, muestra todos los productos que tiene almacenados en el carrito de la compra.

Para empezar, la página muestra la suma total de los precios individuales de los elementos del carrito. A continuación, muestra un botón de "comprar" y bajo este botón, una lista con todos los elementos del carrito junto con una opción para eliminarlos.

El usuario puede hacer click en el botón de comprar. Si el usuario no había hecho Login, se le pedirá que lo haga para poder comprar esos productos (en este proceso no se perderán los contenidos del carrito). Una vez el usuario, con el Login hecho, pulse en comprar, si se dispone de saldo suficiente, el carrito se vaciará y se producirá la compra (añadiendo a su historial qué productos compró y en qué fecha lo hizo).

En el caso de que el usuario hubiera añadido productos al carrito sin haber hecho Login antes, es posible que añadiera un producto que ya compró con anterioridad, sin embargo, para evitar que el usuario compre dos veces el mismo producto, se eliminan del carrito automáticamente aquellos productos que ya fueron adquiridos. Se le notifica al usuario de esto.

Por último, bajo cada una de las películas del carrito, hay un botón "eliminar" que permite, en caso de que así lo desee el usuario, extraer un producto del carrito de la compra (de manera que al realizar la compra, dicho producto no intervenga).

Información técnica:

Lo primero que hará el script php al generar esta página es eliminar de la cesta todos los productos que hubiera en la cesta y que el usuario ya hubiera comprado en el pasado. Esto es porque el usuario puede añadir productos al carrito habiéndose identificado o no. En el caso de que lo estuviera, el sistema se asegura de que en la cesta no se introduzcan este tipo de productos (comprobando que no hayan sido ya comprados por el usuario) pero en el caso de que no estuviera identificado, por defecto el sistema lo mete en la cesta, pudiendo suceder que al hacer login más tarde, en la cesta hubiera elementos que el usuario ya compró anteriormente. Por tanto, al generar esta página lo primero que se hace es tomar una medida para arreglar esto.

Esto se hace llamando a la función `fix_cesta` del fichero `includes/utills.php`, que además retorna true en caso de que se tuvieran que efectuar modificaciones a la cesta (se comprueba el retorno para generar el texto que notifica al usuario de este evento)

También se utiliza la función `calculate_total` de `includes/fcesta.php` para obtener la suma total de los precios de las películas de la cesta.

La parte de mostrar los elementos de la cesta se hace mediante un bucle que recorre el array de ids `$_SESSION['cesta']` y encuentra en el `catalogo.xml` los elementos con cada uno de los ids de la cesta, generando el código html necesario para mostrarlos en la página. Tanto el botón de "comprar" como los botones de "eliminar" son formularios que al hacer Submit retornarán a esta misma página pasando campos ocultos en el método POST.

En el caso de los botones de eliminar, los campos ocultos contendrán información sobre la acción que ha realizado el usuario (pulsar un botón de eliminar) y sobre el id de la película

que se desea eliminar. Si esto sucede, mediante la llamada a la función `remove_from_cesta`, se elimina de la cesta el elemento asociado a dicho id. Luego se recalcula el total para mostrarlo bien actualizado.

En el caso del botón de comprar, el campo oculto informa sobre que el usuario pretendía comprar los elementos de la cesta y con la fecha (en la zona horaria del usuario) en la que se realizó la compra. El sistema procederá a realizar las siguientes acciones:

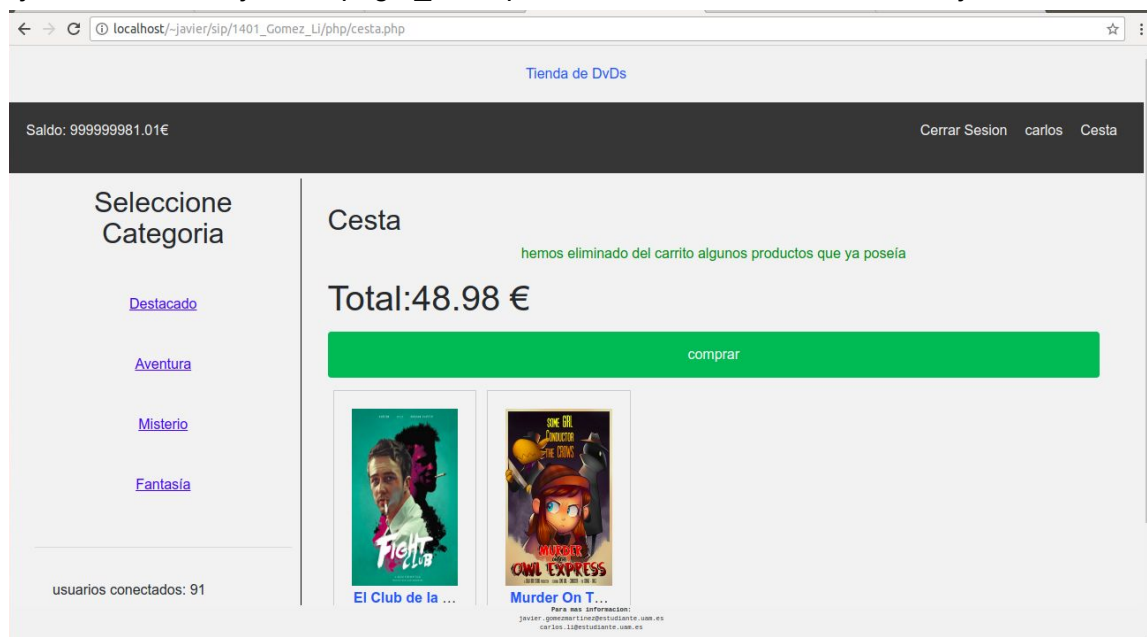
Primero comprobará en la variable de sesión si el usuario se ha identificado o no. En caso negativo, le redireccionará a la página de Login. En caso afirmativo, se procede con la compra

Luego comprobará que dispone de saldo suficiente para realizar la operación. En caso de negativo, lo notificará, en caso afirmativo, procederá con la compra (sustrayendo del saldo del usuario el precio total del carrito)

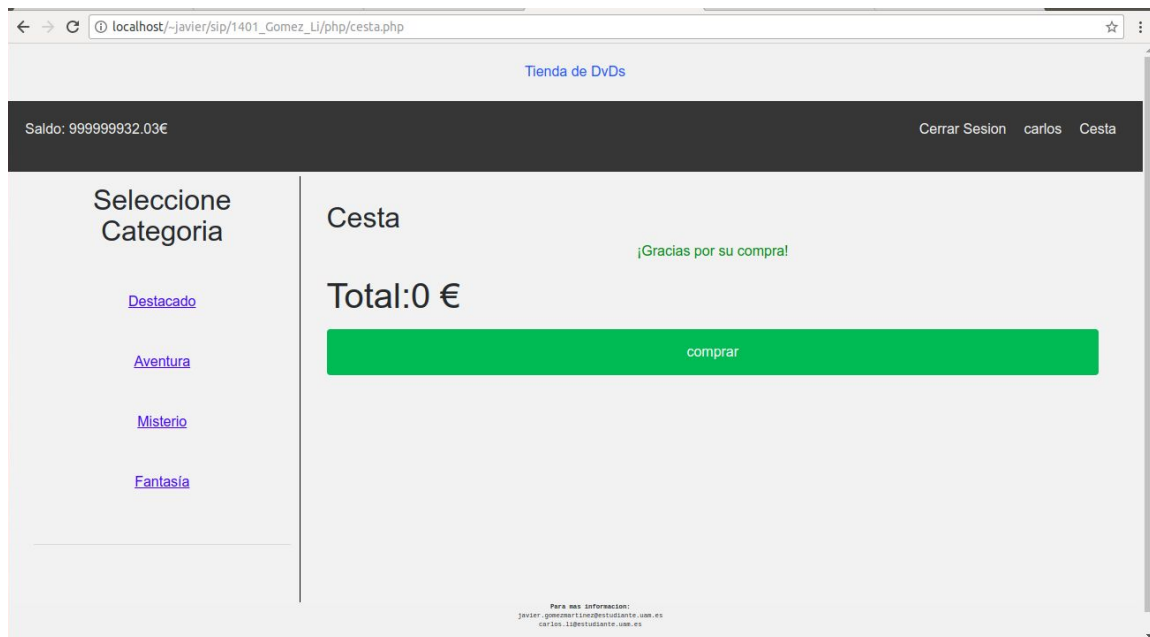
Después agregará cada uno de los elementos de la cesta al historial del usuario con la fecha correspondiente a cuando se realizó la compra.

Finalmente, se limpiarán los datos almacenados en la variable de sesión que contiene los productos del carrito de la compra, para que el usuario pueda continuar navegando y realizando compras en la página web.

Para que el cliente pueda indicarle al servidor la fecha a la que se realizó la compra, se ejecuta la función javascript `get_date` que se encuentra en el fichero `utils.js`



captura de pantalla de la página de la cesta. Aquí podemos ver la configuración de la página. También mostramos un texto que notifica al usuario de cierto evento relevante con respecto a la cesta de la compra.



La página de la cesta de la compra tras realizar una compra (la de la anterior captura) como podemos observar, todos los datos se actualizan correctamente y se le notifica al usuario que su compra se realizó con éxito

Historial

En esta página el usuario puede consultar su historial de compras, viendo así qué películas compró.

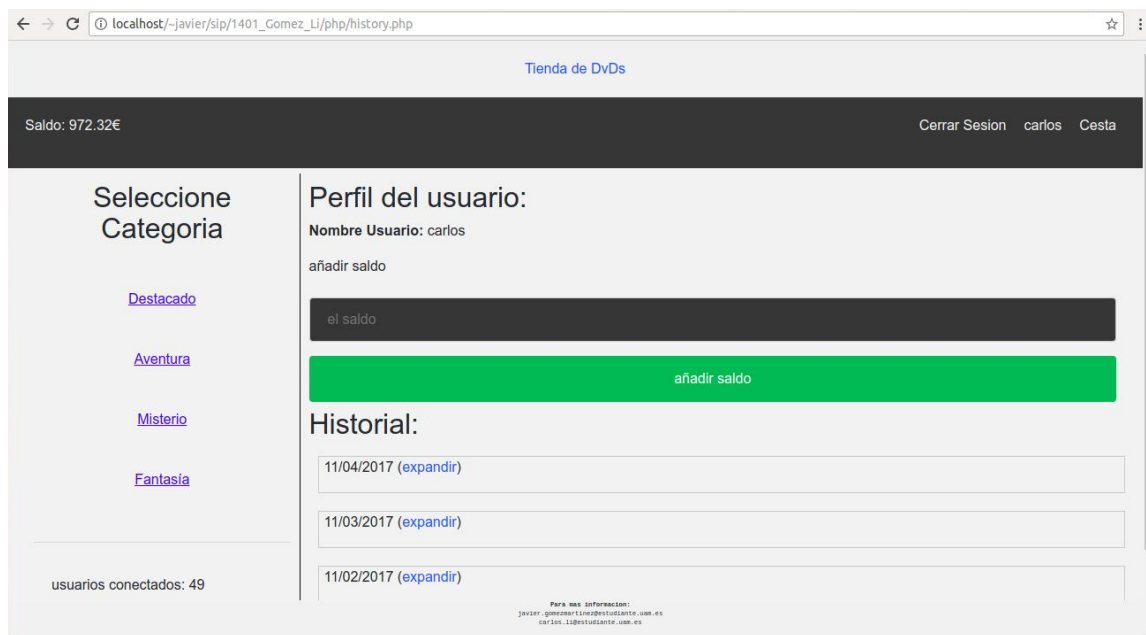
El historial está organizado por fechas y, para no ocupar demasiado espacio a priori) haciendo click en “extender”, para cada una de las fechas se despliega la lista de las películas que compró. Se pueden tener varias fechas desplegadas a la vez y se pueden volver a plegar.

Información técnica:

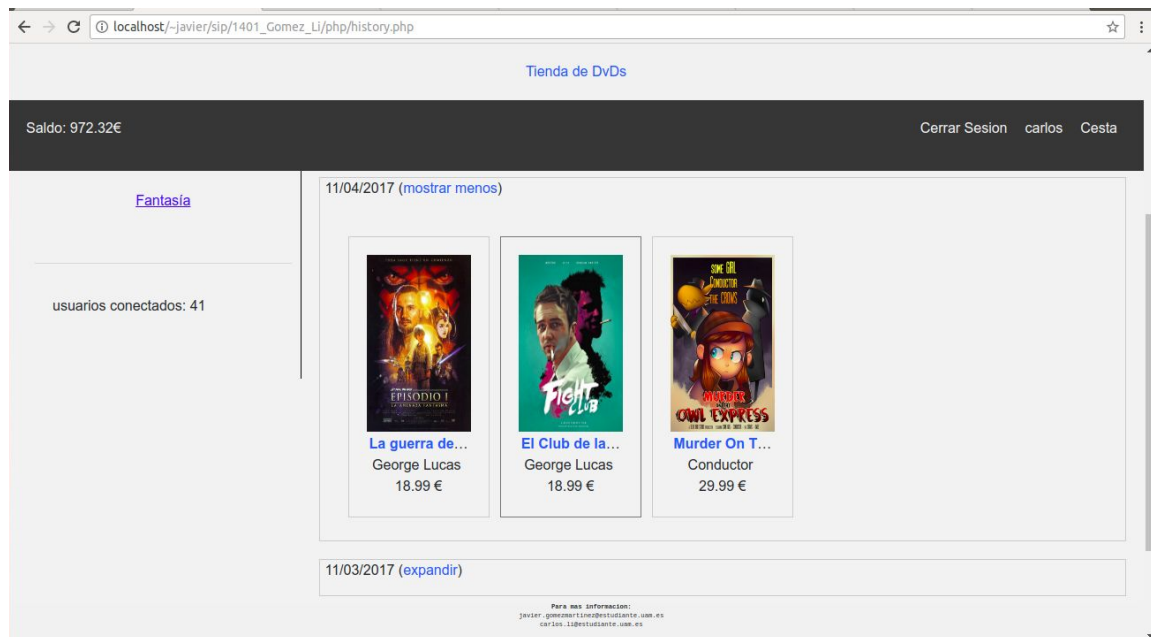
Desde el punto de vista técnico esta página es más bien simple. Mediante una serie de bucles y queries sobre un handler del fichero historial.xml del usuario en cuestión, se obtienen los datos necesarios para rellenar los contenidos de la página.

Las funciones utilizadas se encuentran todas en el módulo includes/utills.php.

Para realizar el efecto de los desplegables, se ha codificado un script en el lenguaje javascript que se encuentra en el fichero toggle_script.js empleando la tecnología de jQuery.



Captura de la página del historial con todas las fechas sin desplegar



En esta captura podemos ver cómo se muestra la lista de productos del historial una vez desplegamos una fecha.