



Proyecto Final de Carrera

""""""Ncs Películas

Valencia, Septiembre de 2012

''

''



Autor: Roberto Lucas Echeverría
Director: Pedro J. Valderas Aranda

INDICE DEL CONTENIDO

1. Introducción.....	6
1.1. NACIMIENTO	6
1.2. OBJETIVOS	7
1.3. MOTIVACIÓN	7
1.4. ESTRUCTURA.....	8
2. Especificación de Requisitos	8
2.1. INTRODUCCIÓN	8
2.1.1. <i>Propósito</i>	8
2.1.2. <i>Ámbito</i>	8
2.1.3. <i>Definiciones, siglas y abreviaturas</i>	9
2.1.4. <i>Visión Global</i>	9
2.2. DESCRIPCIÓN GENERAL.....	10
2.2.1. <i>Perspectiva de producto</i>	10
2.2.2. <i>Funciones del producto</i>	10
2.2.3. <i>Características del usuario</i>	11
2.2.4. <i>Restricciones</i>	11
2.2.5. <i>Supuestos y dependencias</i>	11
2.3. REQUISITOS ESPECÍFICOS.....	11
2.3.1. <i>Requisitos de interfaces externas</i>	11
2.3.2. <i>Requisitos funcionales</i>	12
2.3.3. <i>Rendimiento y atributos</i>	13
3. Análisis.....	13
3.1. CASOS DE USO.	13
3.2. DIAGRAMA DE CLASES.	14
3.3. DIAGRAMAS DE SECUENCIA.....	15
4. Diseño.....	17
4.1. INTERFAZ DE USUARIO.....	18
4.1.1. <i>Diagramas de navegabilidad</i>	18
4.1.1.1. <i>Usuario invitado</i>	18
4.1.1.2. <i>Usuario Registrado</i>	18
4.1.1.3. <i>Usuario Administrador</i>	18
4.2. NIVEL LÓGICO O DE NEGOCIO.	18
4.3. NIVEL DE PERSISTENCIA DE DATOS.	19
5. Implementación.	20
5.1. TECNOLOGÍAS.....	20
5.2. HERRAMIENTAS.	21
5.3. DETALLES DE LA IMPLEMENTACIÓN.....	22

5.3.1. Introducción.	22
5.3.2. Cabecera.	23
5.3.3. Juego.	24
5.3.4. Gestión del administrador.	30
5.3.5. Datos personales.	32
5.3.6. Niveles.	32
5.3.7. Extras.	33
5.3.8. Instrucciones, conseguir créditos y descuentos.	34
5.3.9. Sugerencias.	35
5.3.10. Contacto.	36
5.4. IMPLEMENTACIÓN DE LAS FUNCIONALIDADES.	36
5.4.1. Registro de usuarios.	36
5.4.2. Conexión a la base de datos.	38
5.4.3. Administrador.	38
5.4.4. Principal.	40
5.4.5. Sugerencias.	41
5.4.6. Comprobar respuestas.	42
5.4.7. Comprar pistas.	43
5.4.8. Ver video.	44
5.4.9. Desconectar.	44
5.4.10. AJAX.	45
5.4.11. Directorios.	48
5.4.12. Uso de código externo.	49
5.4.13. Implementación de la interfaz gráfica.	49
5.4.14. Implementación de la gestión de datos.	51

6. Evaluación y pruebas.....52

6.1. PRUEBAS	52
6.2. ANÁLISIS DE LOS RESULTADOS.	53

7. Conclusiones.54

7.1. VALORACIÓN PERSONAL	54
7.2. FUTURAS AMPLIACIONES O MEJORAS.	55

8. Bibliografía.....56

8.1. LIBROS CONSULTADOS.	56
8.2. PAGINAS WEB VISITADAS.	56

9. Anexos.....57

9.1. ANEXO A. DISEÑO LÓGICO DE LA BASE DE DATOS.	57
9.2. TOMA DE DATOS	57

Memoria proyecto final de carrera

Juego de las Películas

1. Introducción

El presente proyecto fin de carrera está relacionado en el área de la cultura, más concretamente en el cine, el séptimo arte.

El cine es la técnica de proyectar fotogramas de forma rápida y sucesiva para crear la impresión de movimiento, mostrando algún video.

La palabra cine es una palabra griega que significa movimiento, como forma de narrar historias o acontecimientos, el cine es un arte, y comúnmente, considerando las seis artes del mundo clásico, se lo denomina séptimo arte.

La historia del cine comienza el 28 de diciembre de 1895, fecha en la que los hermanos Lumière proyectaron públicamente la salida de obreros de una fábrica francesa Lyon, la demolición de un muro, la llegada de un tren y un barco saliendo del puerto.

El juego de las películas va a intentar medir el conocimiento del usuario de varias de las películas más famosas del cine repartidas por niveles de dificultad y que se puede extender también a series, cómics, cuadros etc.

La idea general del juego es mostrar un fotograma y si no se sabe la película podrá optar a diferentes tipos de pistas. El desarrollo del proyecto se ha llevado a cabo utilizando el lenguaje de programación PHP, el programa de bases de datos MySQL, javascript, Ajax, HTML, Workbench.

1.1. Nacimiento del proyecto

El proyecto nace a causa de un juego que encontré por Internet en el que se mostraba un fotograma y abajo un campo de texto que se ponía en verde si la película era correcta, pero me quedaron películas por adivinar y decidí que era buena idea introducir diferentes tipos de pistas para tener más oportunidades de adivinar la película.

También decidí que era buena idea guardar un registro de cada uno de los usuarios para que no tuvieran que empezar el juego siempre desde el principio. Poco a poco empezaron a surgir ideas como introducir niveles de dificultad para hacerlo más atractivo y una cantidad de créditos para que el usuario gestione estos créditos eficientemente y sea más divertido y ameno.

1.2. Objetivos

El objetivo principal del proyecto trata de que la gente aficionada al cine, a las series, etc. pase un buen rato intentando adivinar la totalidad de pruebas con el menor número posible de créditos utilizados. El usuario deberá pasar de nivel para poder conseguir más créditos, y para poder pasar de nivel se necesita un mínimo de películas acertadas en el nivel anterior.

Como todas las películas y otras pruebas que se tienen que adivinar en esta página están bastante contrastadas, la mayoría de los usuarios podrán descubrir películas recomendadas y que para ellos eran desconocidas, así como entrarle ganas de volver a ver algunas de ellas.

El juego se hace muy divertido cuando se realiza con un grupo de gente.

También se aceptan sugerencias para hacer la página más entretenida y más interesante.

1.3. Motivación

La motivación que me lleva a la realización de este proyecto es la de aprender a desarrollar por completo una aplicación Web que interactúa con los usuarios, y no se limita solamente a ofrecer información.

Este proyecto me ha motivado a aprender un lenguaje nuevo como PHP que hasta ahora aún no sabía utilizarlo, y ahora tengo algo de soltura, y el manejo de bases de datos MySQL. De esta forma he aprendido a unir estas dos tecnologías para realizar este proyecto.

Para coger ideas he visitado algunas páginas Web de cine y así poder elegir un conjunto de películas y intuir su dificultad y su nivel de conocimiento en la sociedad como www.filmaffinity.com

USUARIOS
Votar los tours
Entrada usuarios
Registrarse

ESPAÑA
Películas en cartelera
Cines España
Próximos estrenos
Estrenos DVD venta
Próximos DVD venta
Por fin en DVD
Ya para alquilar
Próximamente alquiler

USA - GB - FR
Estrenos USA
Estrenos Reino Unido
Estrenos Francia
DVD/Vídeo USA

SECCIONES
Taquilla
Trailers
Últimos trailers
Últimas críticas
Todas las películas
Películas por temas

TOPS
TOP Filmaffinity
TOP Estrenos
TOP estrenos DVD

El Padrino. Parte III

TÍTULO ORIGINAL The Godfather: Part III
AÑO 1990
DURACIÓN 163 min. [Trailers/Videos](#)
PAÍS [Sección visual](#)
DIRECTOR [Francis Ford Coppola](#)
GUIÓN Francis Ford Coppola & Mario Puzo (Novela: Mario Puzo)
MÚSICA Carmine Coppola
FOTOGRAFÍA Gordon Willis
REPARTO [Al Pacino](#), [Diane Keaton](#), [Andy García](#), [Joe Mantegna](#), [Talia Shire](#), [Eli Wallach](#), [Sofia Coppola](#), [George Hamilton](#), [Raf Vallone](#), [Bridget Fonda](#), [Helmut Berger](#), [John Savage](#), [Mario Donatone](#), [Franco Citti](#), [Al Martino](#), [Vito Antuofermo](#), [Richard Bright](#)
PRODUCTORA Paramount Pictures
PREMIOS 1990: 7 Nominaciones al Oscar, incluyendo película, director y actor de reparto (García)
1990: 2 Premios Razzie: Peor actriz sec. y peor nueva estrella (Sofia Coppola)
GÉNERO Drama | Mafia. Crimen. Secuela. Familia
SINOPSIS Michael Corleone, heredero del imperio de don Vito Corleone, intenta rehabilitarse socialmente y legitimar todas las posesiones de la familia negociando con el Vaticano. Después de luchar toda su vida se encuentra cansado y centra todas sus esperanzas en encontrar un sucesor que se haga cargo de los negocios. Vincent, el hijo ilegítimo de su hermano Sonny, parece ser el elegido. (FILMAFFINITY)
CRÍTICAS
"Siendo con mucho la peor de las tres, es una película soberbia. (...) La secuencia final, un espectacular clímax (...) aún es uno de los mejores momentos del cine de los noventa." (Miguel Ángel Palomo: Diario El País)
"La mejor de la saga" (Augusto M. Torres: Diccionario Espasa)
TU CRÍTICA
Puedes hacer una crítica de esta película para que el resto de los usuarios la pueda leer.
[Añade tu crítica](#)
CRÍTICAS DE LOS USUARIOS
Hay **149** críticas de los usuarios para esta película: [Ver todas](#) | [Ver por títulos](#)
Votaciones de tus Almas Gemelas (Personas con las que tienes mayor afinidad):

7,9
★★★★★
49.021 votos
149 [críticas](#) - [por títulos](#)

TU VOTO
Haz [login](#) para votar

REDES SOCIALES
[Me gusta](#) <223
[Twitter](#)
[+1](#) <2

Comprar:
DVD
[El Padrino: Parte III: La Remasterización](#)

1.4. Estructura

La memoria de este proyecto está estructurada en el orden cronológico en el que se han realizado cada una de las fases para desarrollarlo.

A continuación vemos un resumen del contenido de cada uno de los capítulos:

Introducción: se hace una pequeña presentación del proyecto donde se destacan los objetivos a cumplir, y las motivaciones.

Especificación de requisitos: en este capítulo se detallan las funcionalidades que debe tener la aplicación Web.

Análisis: aquí se indica de que forma debe comportarse la aplicación Web ante los distintos tipos de usuario.

Diseño: en este capítulo se define de que forma está estructurada la aplicación Web.

Implementación: aquí se detalla el proceso de implementación de la Web, indicando las tecnologías y herramientas usadas y comentando el código resultante.

Evaluación y pruebas: en este capítulo se describen las pruebas realizadas sobre la aplicación Web para comprobar que su funcionamiento es el correcto. Se muestran los resultados y se hace un análisis de ellos.

Conclusiones: aquí se dan las conclusiones que se han obtenido sobre la realización del proyecto, y una valoración personal sobre el trabajo realizado.

Bibliografía: aquí indicamos el material que se ha consultado (libros, artículos, páginas, Web) para obtener ayuda durante la realización del proyecto.

Anexos: en este último apartado colocamos la información que complementa a algunos de los puntos expuestos en otros capítulos.

2. Especificación de Requisitos

2.1. Introducción

2.1.1. Propósito

La finalidad de aquí es conseguir una aplicación Web intentando conseguir un juego y explicar de una forma general y sin entrar en detalles del software y objetivos de la página, ya que aún no se ha implementado.

2.1.2. Ámbito

Se desarrollara una aplicación Web para jugar on-line.

Será una página para cualquier persona, pudiendo ser visualizada por cualquier con acceso a Internet, sin ninguna restricción de edad y en el que será necesario registrarse para poder jugar, añadir sugerencias, mantener el perfil con las puntuaciones de cada usuario...etc.

Por otra parte el administrador tendrá su propio espacio y por tanto tendrá acceso para buscar cualquier película o pista introducida, poder modificar ciertos apartados, añadir videos, sonidos, pistas, modificar cada película y sus pistas relacionadas en tiempo real, así como eliminarlas.

2.1.3. Definiciones, siglas y abreviaturas

Portal Web: es una aplicación basada en html que trata algún tema en particular pudiendo hacer ser interactiva con el usuario.

Foro: lugar de discusión para cualquier usuario y dar su opinión sobre algún tema.

Registro: darse de alta en una base de datos, con datos personales para poder saber que es la persona que dice ser.

Usuario registrado: cualquier usuario que ha pasado por el registro.

Administrador: persona con privilegios que será capaz de actualizar el portal Web y manejar la base de datos, siempre que no infrinja la ley orgánica de protección de datos (LOPD).

Login: a la hora de registrarse es el nombre que la persona decide para poder entrar en el portal.

Logout: Desconectarse como usuario registrado, pasando a salir del juego.

Password: será una cadena alfanumérica, que solo el usuario conoce para poder acceder al juego.

Bd: base de datos.

Php: es un lenguaje de programación diseñado para crear páginas Web. Usado para interpretar scripts del lado del servidor. Puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

MySQL: es una base de datos desarrolladas bajo código abierto, es un lenguaje de consulta de datos estructurada.

Servidor: ordenador o conjunto de ordenadores que proporcionan servicios, como paginas Web, servicios de correo, etc. De forma centralizada.

WampServer: servidor de prueba utilizado para testear el juego como local, aquí también se instalan phpMyadmin y un Server apache entre otras cosas.

CSS: Es una hoja de estilos para hacer la pagina Web mas atractiva al usuario, se usa en programación html con Dreamweaver.

JavaScript: programa que se ejecuta al iniciar una pagina Web como algún tipo de movimiento o al presionar sobre algún enlace.

Ajax: En realidad es parte de código Javascript, pero hace conexiones al servidor de una manera asíncrona consiguiendo que aparente no refrescarse la página o parte de ella. Se comenta aparte porque está cogiendo una relevancia considerable al haber muchos desarrollos de código basados en esta tecnología como puedan ser las librerías JQuery o algunos Framework como Yii o CodeIgniter.

Dreamweaver: programa de Adobe para la creación de documentos Html (paginas Web) con la cual se podrán usar CSS y java script.

Html: Lenguaje de Marcas de Hipertexto, es el lenguaje predominante para la construcción de páginas Web, es usado para describir la estructura y el contenido en forma de texto. Pueden incluir scripts y las etiquetas están rodeadas por <>

Http: Protocolo de transferencia de hipertexto es el protocolo usado en cada transacción de la Web (WWW). HTTP fue desarrollado por el consorcio W3C.

Photoshop: Programa de Adobe con el que se pueden editar imágenes con suficientes herramientas como para conseguir cualquier cosa en edición de imágenes.

2.1.4. Visión Global

En primer lugar se hace una descripción general de los apartados que tendrá la Web, y los usuarios que intervienen en ella.

En el siguiente apartado, se describe de una forma más específica el funcionamiento de los distintos módulos que componen las distintas secciones de la Web.

2.2. Descripción general

2.2.1. Perspectiva de producto

Este producto es como cualquier otro producto software, y por tanto no forma parte de ningún proyecto más amplio que lo englobe.

El portal Web se desarrollará en HTML, con Dreamweaver, junto con el lenguaje de programación PHP para generar el contenido dinámico. Accederá a una base de datos MySQL para obtener los datos necesarios para generar dichas páginas. Es necesario por lo tanto trabajar con un sistema que nos ofrezca un servidor Web con soporte para PHP junto con una base de datos MySQL.

Para llevar a cabo el desarrollo de la aplicación de una forma más cómoda, y antes de que sea publicada en Internet, he optado por utilizar el paquete WampServer para Windows.

Que nos ofrece de forma conjunta el servidor Web, la base de datos, y el soporte de PHP.

Las funcionalidades de este servidor son las siguientes:

- Gestionar los servicios de Apache y MySQL,
- Cambiar de línea / fuera de línea (dar acceso a todos o sólo local)
- Instalar Apache, MySQL y PHP
- Gestión de la configuración de los servidores
- Acceder a los registros
- Acceder a los archivos de configuración
- Crear alias

2.2.2. Funciones del producto

Proporcionar una página Web compuesto por varias secciones donde los usuarios que accedan a él puedan realizar las siguientes tareas:

Usuario normal

- Poder registrarse como usuario para poder acceder al juego.-
- Loguearse
- Añadir sugerencias.
- Modificar sus datos personales.
- Contactar con el administrador.
- Comprar todo tipo de pistas.
- Cambiar de juegos.
- Cambiar de niveles de dificultad.
- Leer las instrucciones del juego.
- Ver y conseguir créditos.
- Ser recomendado por un amigo para conseguir créditos.
- Pasar de película.
- Reproducir las pistas compradas tantas veces como se quiera.
- Desloguearse.

Administrador

- Listar todas las películas.
- Buscar películas por pista, título, nivel...
- Modificar pistas, títulos, dificultad...
- Eliminar un registro.
- Subir más películas con sus correspondientes pistas y campos.

2.2.3. Características del usuario

Este producto va dirigido para todo el público, aunque principalmente para los usuarios a los que les guste el séptimo arte.

Hay dos tipos de usuarios:

- **Usuario registrado:** este usuario podrá ver los contenidos del sitio, así como jugar todas las veces que quiera, cambiar sus datos y todo lo explicado en el punto anterior. Pero en ningún caso podrá realizar tareas de administración sobre el sitio.

- **Administrador:** es el usuario que tiene todos los privilegios sobre el sitio, y será capaz de ampliar el juego, gestionar el tipo de contenido, como la base de datos etc.

2.2.4. Restricciones

Para poder acceder al portal, se deberá usar un navegador Web.

La Web es de acceso restringido, se exigirá al usuario su registro si aún no lo ha hecho, y su posterior autenticación. De esta forma se consigue que un usuario mantenga sus datos cada vez que se autentifique y no volver a empezar desde el principio.

2.2.5. Supuestos y dependencias

En este caso, el sistema operativo utilizado para acceder al portal es indiferente. Por tanto podemos decir que este producto es multiplataforma.

Asimismo, el navegador Web utilizado también puede ser cualquiera de los disponibles en el mercado. Aunque este producto se ha optimizado para el navegador Mozilla Firefox versión 3 o superior.

2.3. Requisitos específicos

2.3.1. Requisitos de interfaces externas

La página Web se divide en varias secciones, claramente diferenciadas:

- **Área de registro de usuarios.** Donde cualquier usuario puede registrarse gratuitamente para poder acceder al juego y poder empezar a pasárselo bien.

- **Área de juego.** Donde el usuario puede empezar a interactuar con la página, comprar pistas, ver los créditos de los que dispone para comprar pistas, escuchar las pistas compradas, comprobar el título de la película, etc.
- **Área de menú superior.** Donde se pueden ir a las diferentes secciones que ofrece la página, ya sea cambiar datos personales, cambiar de nivel, o de clase de fotograma, entre otras.
- **Área de logo.** Donde se muestra el logo y en un futuro posibles frases célebres de películas conocidas o consejos varios.
- **Área de administración.** Donde el administrador del sitio Web pueden acceder mediante login para poder manejar la parte interna del sitio Web. El usuario administrador tendrá permisos para poder realizar cualquier operación.

2.3.2. Requisitos funcionales

- **Registro de usuarios.**

Cualquier visitante de la Web puede registrarse para poder acceder al contenido.

Los campos que serán requeridos para el registro de cualquier usuario son: nombre, apellidos, ciudad, correo electrónico y contraseña. Se comprobará que no se deje ningún campo obligatorio sin rellenar, y que el correo electrónico no existan ya en la base de datos.

El correo electrónico se repetirá dos veces para evitar confusiones utilizando la misma política para la contraseña.

También hay campos no requeridos como la ciudad y el correo electrónico de un usuario registrado.

Cualquier usuario podrá enviar un e-mail al administrador. Estos mensajes son enviados a un correo especial del administrador que será leído periódicamente.

- **Consultar información.**

Los usuarios podrán consultar todos los apartados de la Web.

- **Enviar sugerencias.**

El usuario podrá enviar las sugerencias que se le ocurran tanto para mejorar la web, como para añadir nuevas funcionalidades.

- **Usuarios registrados**

Podrán realizar todas estas acciones :

- **Editar su perfil.**

Los usuarios podrán modificar sus datos personales que introdujeron cuando se registraron en el juego, como su nombre completo, su e-mail, o la posibilidad de cambiar la contraseña. Cada vez que se requiera cambiar un dato personal se requerirá la contraseña de usuario. Si se quiere cambiar la contraseña de usuario se necesitará la actual contraseña e introducir dos veces la nueva contraseña.

- **Contacto.** Cualquier usuario registrado podrá ponerse en contacto con el administrador para preguntar cualquier tipo de duda.

2.3.3. Rendimiento y atributos

La eficiencia del portal Web, dependerá de la capacidad de proceso del servidor donde se aloje, y del ancho de banda de la red disponible para el servidor y para el cliente. El sistema operativo elegido por el servidor o por el cliente no es un factor importante para la eficiencia, ya que la página Web puede trabajar correctamente sobre cualquiera de ellos.

Para realizar las tareas de mantenimiento o actualización del sitio Web, el administrador podrá acceder desde cualquier lugar con conexión a Internet. Esto es posible gracias a que la información se encuentra guardada en la base de datos del servidor.

3. Análisis.

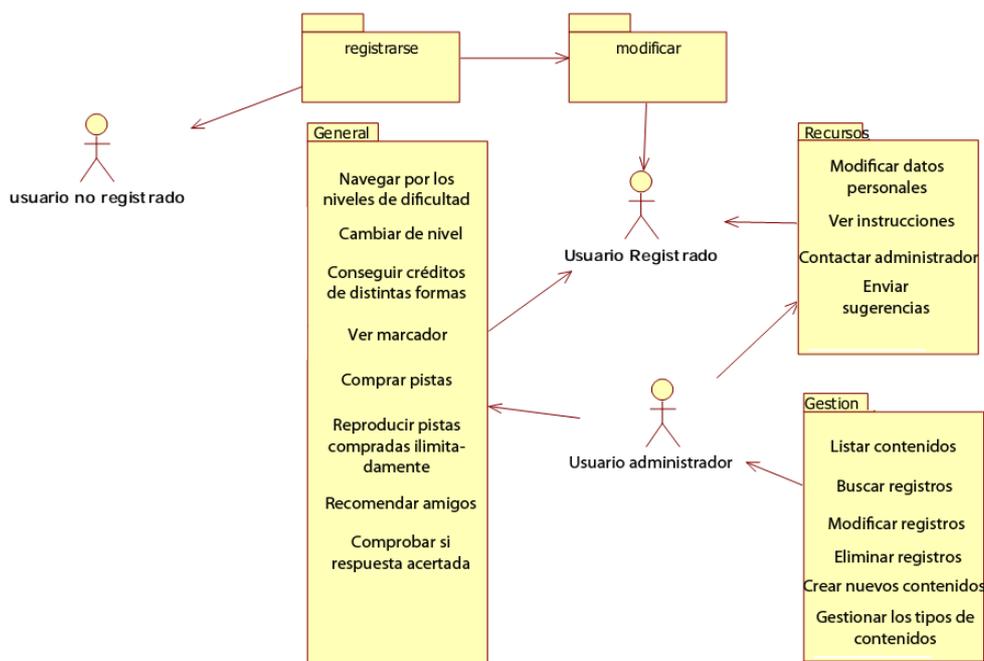
3.1. Casos de uso.

Los casos de uso es una técnica para capturar los requisitos de un nuevo sistema y las acciones que pueden realizar los usuarios con los escenarios que proporciona el sistema.

Se intenta evitar aquí las jergas técnicas para que así un usuario no iniciado en la materia pueda entender el papel de cada usuario en el sitio web y de que forma interactuar.

En este sistema intervienen dos tipos de usuarios: el usuario registrado y el usuario administrador., que tendrá permisos para modificar las bases de datos.

A continuación en la siguiente figura se muestran los casos de uso del sistema.

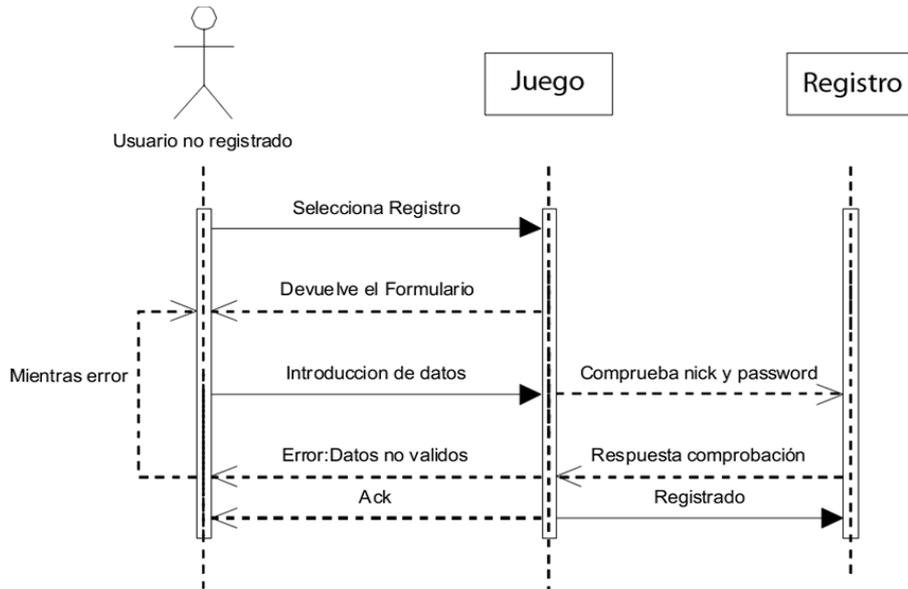


Casos de uso

3.2. Diagramas de secuencia.

Son usados para modelar la interacción entre objetos en un sistema modelado mediante UML. Se muestran a continuación un conjunto de objetos de aplicación a través del tiempo y se modela para cada método de la clase. Estos diagramas contienen detalles de implementación del escenario juego.

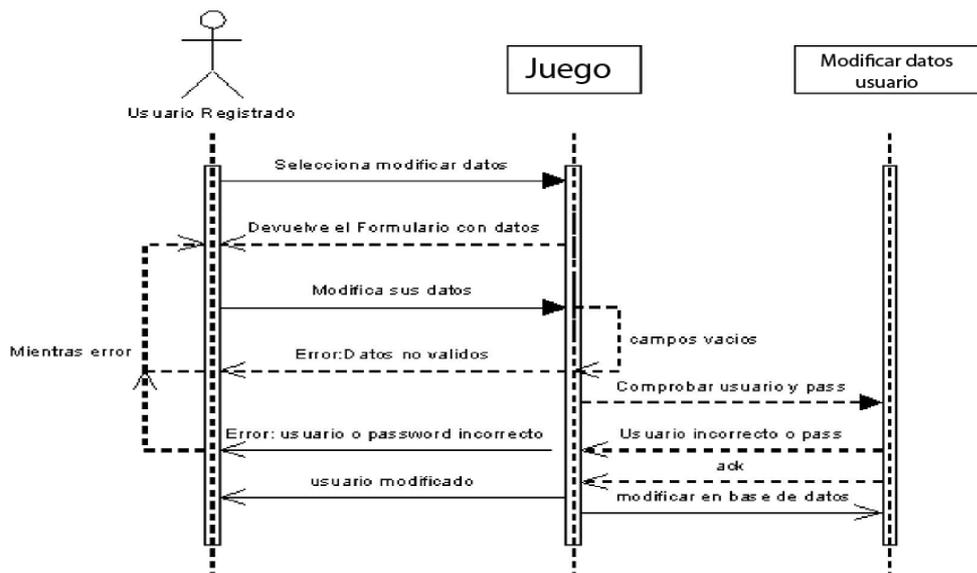
Registro



Cuando se selecciona el registro se devuelve un formulario que mientras no sean correctos se les volverá a pedir a los usuarios que introduzcan los datos hasta que sean correctos.

Modificar datos:

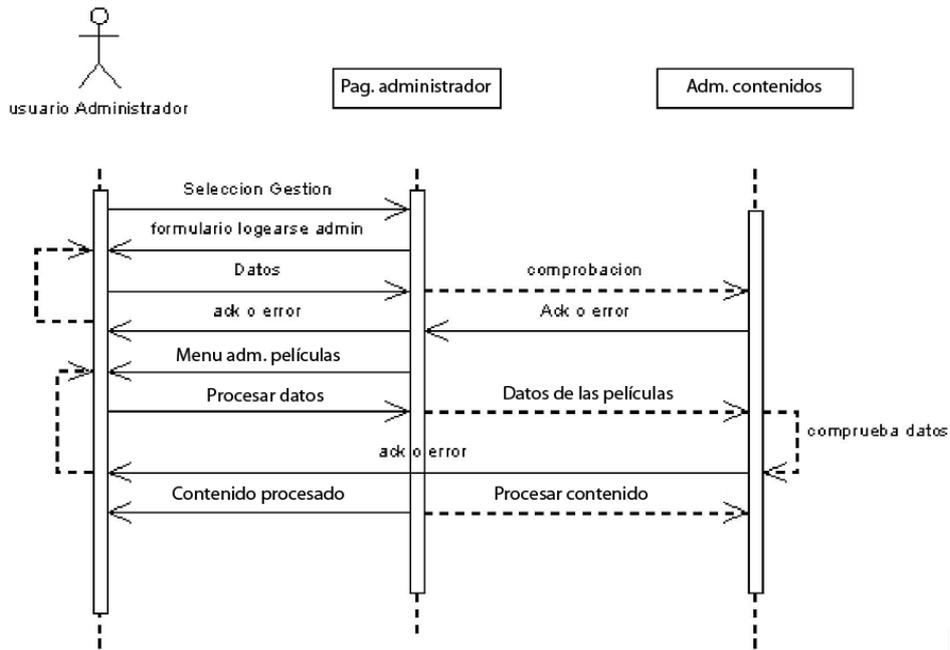
Modificar datos:



Los usuarios registrados pueden modificar sus datos, al seleccionarlo les devuelve un formulario para que lo modifiquen a su gusto mientras los datos sean correctos.

Página de gestión de pistas

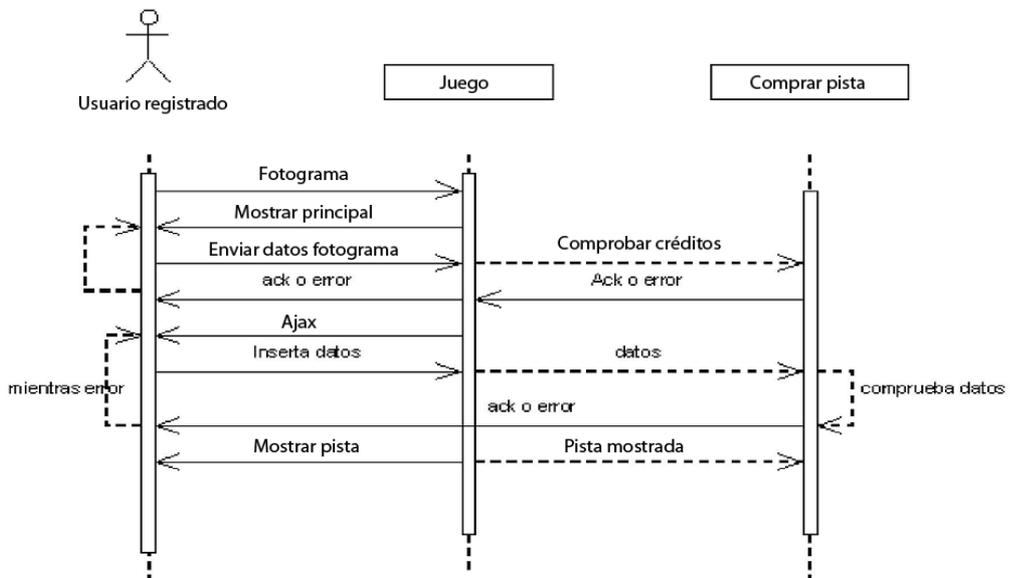
Página administrador



El administrador después de identificarse, se le devuelve un formulario con campos a rellenar para insertar/modificar/borrar cualquier campo de una película mientras los datos sean correctos.

Acceder a pistas:

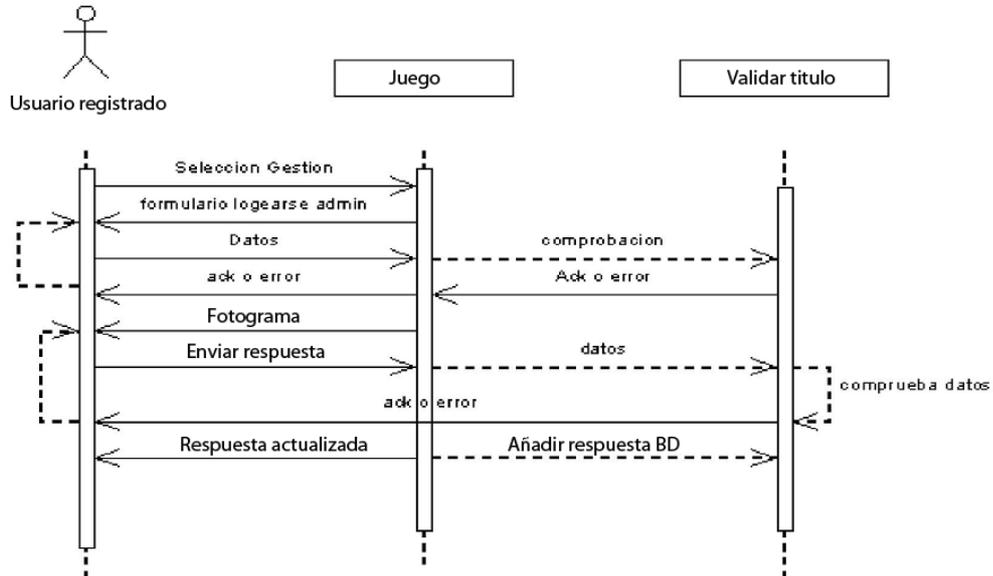
Comprar pista



Los usuarios pueden comprar pistas, previo paso la validación de créditos, después de la validación se recarga el contenido mientras los créditos sean suficientes.

Comprobar los resultados:

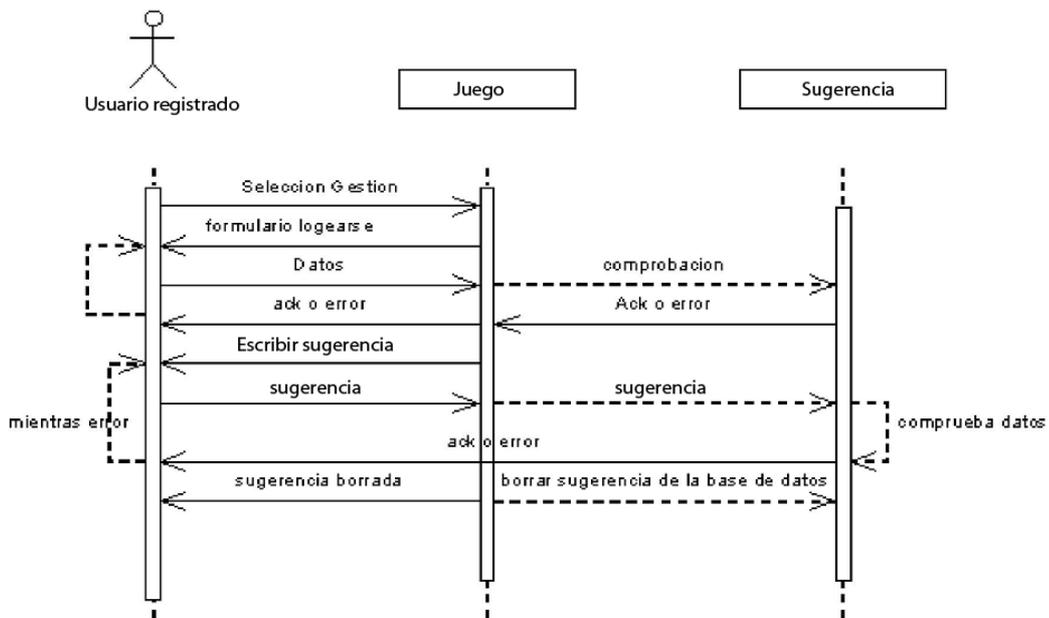
Comprobar respuesta



El usuario registrado, envía una petición con los datos correspondientes, y se actualizan el campo correspondiente, en caso de que los datos sean correctos.

Sugerencia:

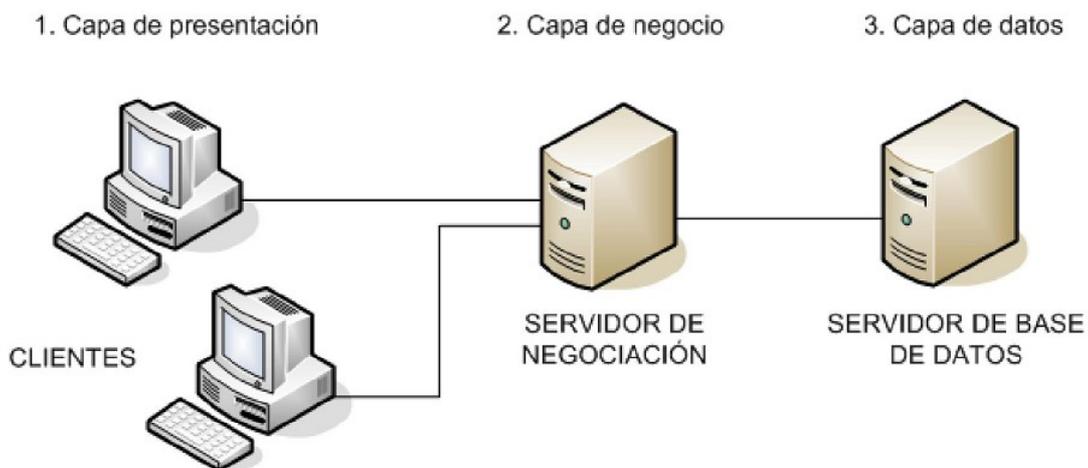
Sugerencia:



El usuario después de loguearse, y seleccionando el apartado sugerencias, podrá añadir una sugerencia siempre que esta sugerencia esté redactada correctamente. Aquí no se comprueba la inyección SQL por escribirse en un archivo de texto.

4. Diseño.

La implementación que he realizado se basa en la programación por capas, las principales ventajas que se obtiene usando este tipo de programación es que si hace falta modificar algo, se ataca directamente a la capa afectada sin necesidad que las otras se vean alteradas. En mi caso he elaborado las tres capas, pero en un entorno de gran escala las diferentes capas se pueden dividir en diferentes grupos de trabajo incluso puede ser escalable pudiendo superar los tres niveles básicos. Los tres niveles utilizados son los siguientes:



1. Capa de presentación, es la que ve el usuario o cliente, esta capa también puede denominarse capa de usuario, esta capa es la que comunica la información y recoge los datos necesarios del usuario, comprobando a su vez los datos errores de formularios etc. Esta capa debería ser entendible y fácil de usar para el usuario, debe tener una interfaz gráfica amigable. Esta capa solo se comunica con la de negocio.

2. Capa de negocio, aquí es donde se ejecutan las operaciones, los datos a operar son los enviados por el usuario en la anterior capa, y tras el computo es la que envía a la capa superior el resultado del proceso, también se denomina a esta capa lógica de negocio, porque es aquí donde están las reglas que deben cumplirse, para así comunicarse con la capa de datos para solicitar al gestor de base de datos que debe almacenarla o recuperar datos de él, e incluso eliminarlos.

3. Capa de datos, es básicamente la que se encarga de almacenar los datos y acceso a los mismos, a su vez puede estar gestionada por uno o mas gestores, que realicen el almacenamiento o acceso, esta capa solo se comunica con la capa de negocio.

Todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.

4.1. Interfaz de Usuario.

Se ha intentado realizar una página intuitiva y a la vez sencilla para que así los usuarios no se pierdan en menús y submenús y al final se pierdan como en muchas paginas. De esta forma al pinchar en alguna sección solo se modifica la zona central para el contenido pudiendo ir a cualquier parte desde cualquier posición.

La pagina se ve bien sin que aparezcan barras de desplazamiento desde una resolución de 800 x 600 la pagina dependiendo del tamaño de ventana se va modificando pero hasta cierto punto. Además esta optimizado para navegadores Mozilla y otros que usen su misma arquitectura base como opera y Chrome e incluso en el molesto Internet Explorer 8.

Zona para el contenido

Hay dos partes para interactuar con el usuario, la parte superior es sin duda la cabecera donde esta el nombre y el logo de la página y justo debajo el mero dinámico que cambiará un poco las secciones dependiendo de si estamos jugando o en cualquier otro apartado.

La segunda parte sería donde se va a mostrar todo los contenidos diferentes de la web. Por defecto después de loguearse aparece el juego directamente en la primera película del menor nivel aún no superado.

Previamente es obligatorio pasar por la página de registro en el que sí es un usuario nuevo tendrá que introducir todos sus datos para poder acceder por primera vez o introducir su usuario y contraseña para entrar a su sesión.

La página del administrador no se hace desde la página de registro de un usuario normal, es un acceso con una dirección diferente para evitar en la medida de lo posible intentos de intrusiones.

Se requerirá también usuario y contraseña y en este caso si algún dato fuera incorrecto la página está programada para que el siguiente intento no se podrá realizar hasta 10 segundos después.

Esto se hace para evitar scripts que por el método de fuerza bruta intentan capturar las contraseñas.

4.2. Nivel lógico o de negocio.

EL Nivel lógico permite relacionar la capa de interfaz de usuario con la capa de datos.

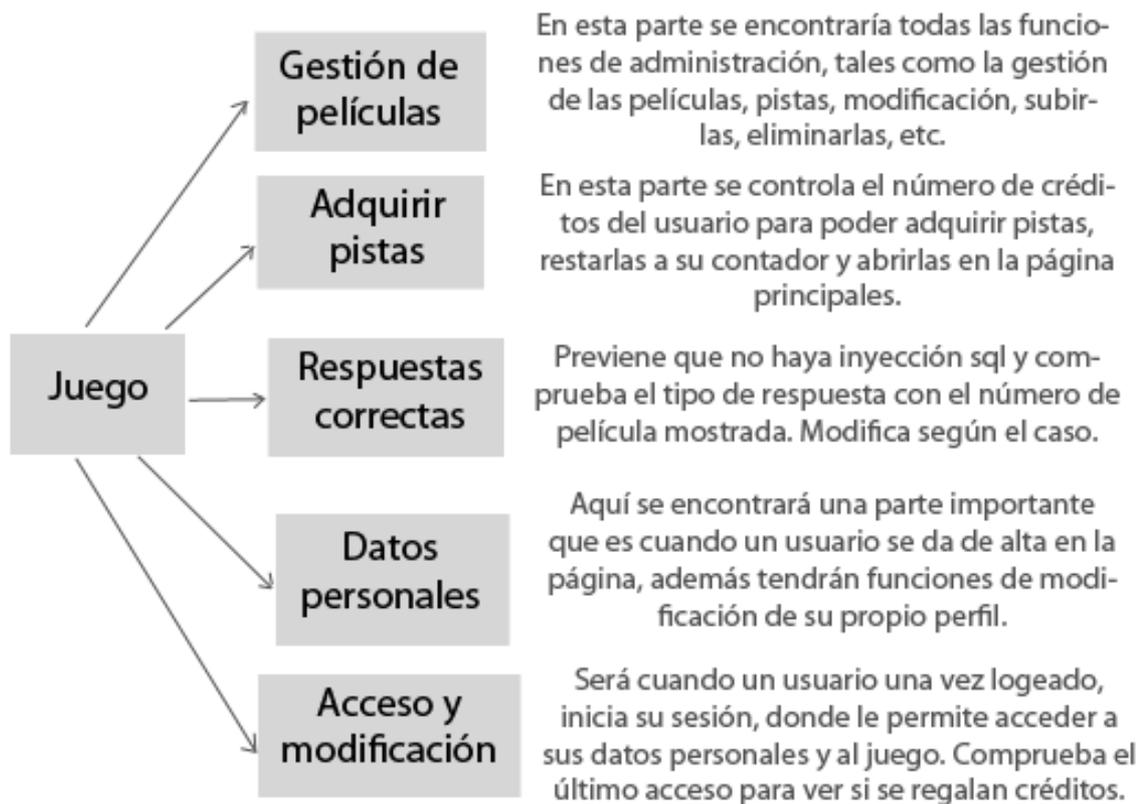
Se encuentra el código que permite llevar a cabo esta relación.

En esta capa se hacen comprobaciones para validar datos, de esta forma se evita mostrar un error inesperado al usuario que debe de pasar inalterado ante estas acciones.

En caso de modificación o cambio en el sistema gestor de base de datos, bastaría con modificar el nivel lógico para adaptarlo a la nueva gestión de datos. De esta forma el nivel de interfaz de usuario no se vería afectado.

Al haber hecho una librería con los módulos de conexión de base de datos solamente habría que cambiar el acceso a este módulo y funcionaria con cualquier tipo de base de datos.

Este diagrama muestra la estructura del nivel lógico de la aplicación:



Hay que tener en cuenta que el método utilizado para llevar a cabo esto ha sido que cada vez que hay una página interactúa con la base de datos esta página envía todo los datos a procesar a otra página invisible para el usuario que generalmente tiene el mismo nombre que la página que envían los datos acabada en proc. Por poner un ejemplo en el momento en el que un usuario se logea está en la página registrarse y esta manda los datos para ser procesados a la página registrarseProc. Esta página se encarga de procesar si son correctos los datos por medio de expresiones regulares y en caso afirmativo se haría la conexión a la base de datos para realizar los cambios o las consultas pertinentes.

4.3. Nivel de persistencia de datos.

A este nivel se encuentra la base de datos junto con el sistema que la gestiona (SGBD). Para diseñar la base de datos, empezamos con la base del diagrama de clases UML expuesto anteriormente. Y a partir del UML definiremos las entidades fuertes del modelo entidad relación. Las tablas creadas para esta página sacadas desde el diagrama UML son: 'Usuario', 'UsuarioPeli', 'Accesos', 'Ciudades', 'Tipo', 'Películas'. En algunas tablas encontraremos un identificador único autonumérico para identificar cada registro de la tabla.

Más tarde en el anexo pondremos ver en detalle la base de datos donde se especifican con más detalle los tipos de campos claves primarias, claves ajenas, auto numéricos, valores nulos o no nulos, etc.

5. Implementación.

Para llevar a cabo la implementación del proyecto, se han utilizado distintas tecnologías tanto para la programación de la web, como para almacenar los datos que nos interesen, y para poner en marcha el servidor web.

5.1. Tecnologías.

Las tecnologías que se han utilizado en este proyecto son los lenguajes HTML, CSS y PHP para la programación de la web. Para la base de datos se ha utilizado MySQL. Y para el servidor web, un servidor Apache. Se ha utilizado un servidor local WAMPserver para hacer pruebas.

WAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

Windows, como sistema operativo.

Apache, como servidor Web.

MySQL, como gestor de bases de datos;

PHP (generalmente), Perl, o Python, como lenguajes de programación.

El uso de un WAMP permite servir páginas html a Internet, además de poder gestionar datos en ellas, al mismo tiempo un WAMP, proporciona lenguajes de programación para desarrollar aplicaciones web.

LAMP es el sistema análogo que corre bajo ambiente Linux MAMP es el sistema análogo que corre bajo ambiente Mac.

Para este proyecto se ha usado la versión de Windows

HTML siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Java script), el cual puede afectar el comportamiento de navegadores Web y otros procesadores de HTML.

Normalmente este lenguaje siempre va unido a hojas de estilo en cascada los CSS.

CSS. Las hojas de estilo en cascada (en inglés Cascading Style Sheets), CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. Estos estilos pueden formar parte de un fichero y así poderlas usar en varios documentos html, o ir dentro del mismo fichero html.

Java script. JavaScript es un lenguaje de scripting basado en objetos sin tipo, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el

desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. JavaScript es un dialecto de ECMAScript y se caracteriza por ser un lenguaje basado en prototipos, con entrada dinámica y con funciones de primera clase.

JavaScript ha tenido influencia de múltiples lenguajes y se diseñó con una sintaxis similar al lenguaje de programación Java, aunque más fácil de utilizar para personas que no programan.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

El uso que le he dado a este lenguaje ha sido para retocar ciertas cosas de los formularios, validar datos, realizar advertencias, AJAX....

AJAX el término AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML".

Se define AJAX de la siguiente forma:

“ Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.”

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

PHP. Es un lenguaje de programación interpretado en la parte del servidor. Su sintaxis es similar a la de C. Hoy en día es uno de los más usados en la programación web. Este lenguaje se encuentra bajo la licencia 'PHP License' que es de carácter libre.

Su uso nos permite conectarnos a la base de datos para realizar funciones de consulta, inserción, actualización, o borrado. También nos permite hacer una tarea muy útil, que es la de generar código HTML de forma dinámica.

MySQL. Es el sistema gestor de bases de datos que hemos utilizado en este proyecto.

Se distribuye bajo la licencia del software libre, y además es multiplataforma. Esas son algunas de las razones por las que es uno de los más usados.

La utilidad de la base de datos en este proyecto ha sido la poder almacenar información y datos, como pueden ser los de los usuarios que se registran en la página, las relaciones entre los usuarios y las diferentes películas, etc...

5.2. Herramientas.

PhpMyadmin. Es una herramienta escrita en PHP para ser usada en un navegador web, y poder administrar una base de datos MySQL. En este proyecto ha sido útil para crear las tablas de la base de datos. Para el usuario administrador al ser único es el que se me dio al principio de proyecto que es el único que puede añadir, modificar y borrar en las tablas.

Dreamweaver. Programa con el que se ha hecho gran parte del proyecto en el se ha programado en html, javascript, css y php.

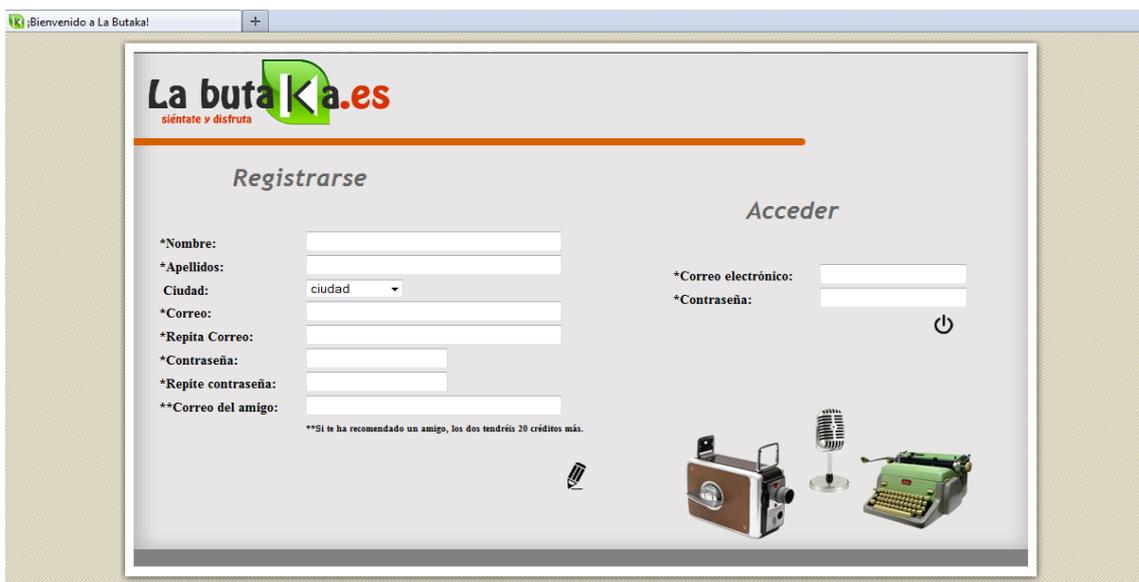
PhotoShop. Es un programa de retoque y edición de imágenes de gran potencia, ha sido usado para modificar ciertos apartados del portal referente a imágenes.

5.3. Detalles de la implementación.

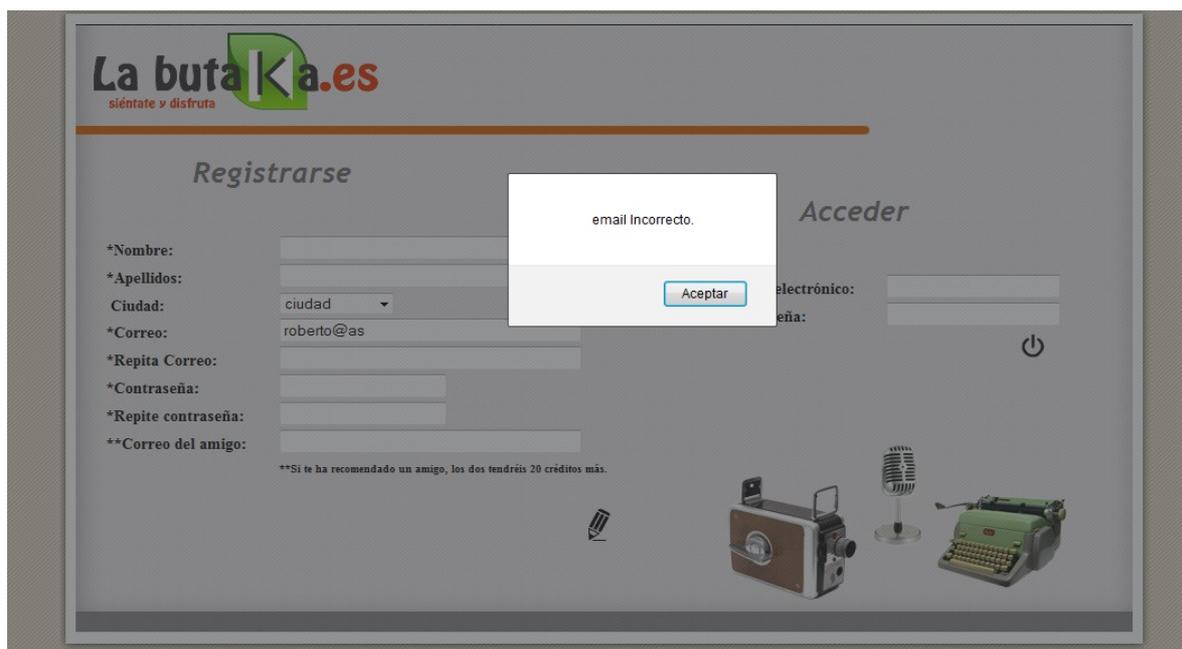
5.3.1. Introducción.

Cuando entramos a la butaKa nos sale la página de registro que se divide en dos partes, la izquierda que es para nuevos usuarios y la derecha es para usuarios registrados.

En la imagen siguiente se muestra la página de inicio:



Aquí hay campos que no son obligatorios, pero todos los campos se comprueban mediante expresiones regulares tanto con javascript en la parte del cliente, como el PHP en la parte servidor. Se muestra un mensaje de error si alguno o varios de los datos son incorrectos, y te avisa qué tipo de dato.



También se comprueba si los correos son iguales y lo mismo con las contraseñas. Se comprueba que el correo sea único y tenga formato de correo.

Aquí podemos ver un poco de código de expresiones regulares que son las que comprueban el correcto formato de los datos:

```
$patronCorreo="/[\w-\.] {3,}@([\w-]{2,}\.)*([\w-]{2,}\.){2,4}/";
$patronContraseña="/([a-zA-Z0-9]{6,18})$/";
$patronTexto="/([a-zA-Z0-9]{3,18})$/";

if (!preg_match($patronCorreo, $correo)) {
    echo " Debes registrarte correctamente ";
    sleep(3);
    header("Location: index.php");
}

if (!preg_match($patronContraseña, $contraseña)) {
    echo " Debes registrarte correctamente ";
    sleep(3);
    header("Location: index.php");
}
```

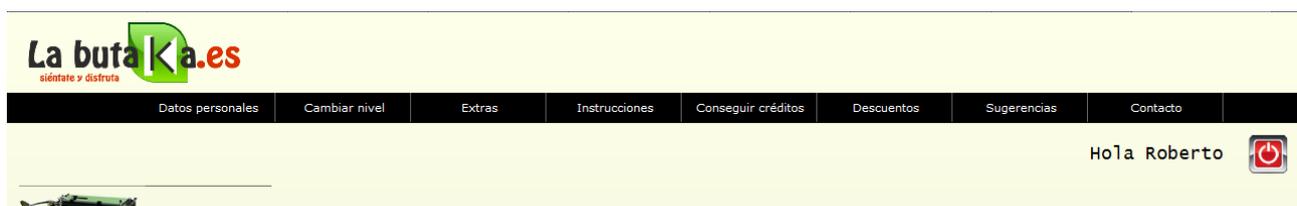
Introducimos la función Sleep() para que si se quisiera hacer un script para sobrecargar el servidor, esta función limitaría su influencia.

Todos los datos se envían por \$POST y si todo es correcto el usuario se registra. A continuación entra directamente a jugar. Se guarda el identificador del usuario en una variable de sesión. Un paso previo para fomentar que se añadan otros usuarios ha sido introducir un campo llamado correo del amigo, en el que si se introduce un correo de un usuario ya registrado, automáticamente a los dos se les añade 20 créditos. La parte de la derecha es para usuarios de registrados, también se comprueban los datos por expresiones regulares y aparte se consulta a la tabla accesos, comprobando si el último acceso es como mínimo anterior al día actual. Si eso ocurre, al usuario se le regala un crédito más y se graba en la tabla accesos el día actual. Con esto se consigue que cada día diferente que un usuario entre tenga un crédito de regalo.

5.3.2 Cabecera

Aquí se pueden distinguir 3 partes. La primera es la zona del logo de la página. La segunda parte es ocupada por el menú principal donde están todas las secciones. En esta parte podemos resaltar que al elegir el nivel de muestra las películas que faltan por acabarlo y muestra solamente los niveles que están abiertos.

Se pueden observar las partes de la cabecera en la siguiente imagen:



Para pasar de nivel, se tienen que completar al menos todas las películas menos 5. Cada vez que se pasa de nivel, los créditos para comprar pistas se incrementan en 20. El menú tiene parte de javascript para que al desplegar algunas secciones se produzca la sensación de movimiento.

La tercera parte consta del nombre del usuario y un botón para salir de la página. Al presionar este botón la página destruye toda las variables de sesión que ha creado, todos los cookies y redirige a la página de registro.

```
//necesario para que puedas cerrar la sesion decirle que la
session_name("sesionActiva");
session_start();

//vacía las cookies asociadas a la sesion
unset($_SESSION['codigo']);
unset($_SESSION['contrasenya']);
unset($_SESSION['posicion']);

//cierra la sesion
session_destroy();

//borrar las cookies que hemos utilizado
//setcookie("posicion", "aqui va un valor", time()-60);
setcookie("nombre", "aqui va un valor", time()-60);

//para redirigir a la pagina de inicio
header("Location: index.php");
```

5.3.3 Juego

Esta sección consta de cuatro partes. Hay que decir que toda esta parte está implementada en AJAX, consiguiendo así una sensación de no cambiar de página a cada clic y a cada comprobación del título de la película.

```
//Variable global que almacena el identificador de la cuenta atras
var cuentaAtras = null;
var tiempoMaximo = 10000; //5000= 5 segundos

var peticion_http;

function cargaContenido(url, metodo, parametros) {

    peticion_http = inicializa_xhr();

    if (peticion_http) {

        //establecer la cuenta atrás al realizar la peticion HTTP
        cuentaAtras = setTimeout(expirada, tiempoMaximo);
        peticion_http.onreadystatechange = muestraContenido;
        peticion_http.open(metodo, url, true);

        //la siguiente linea es necesaria para poder enviar parametros
        peticion_http.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        //el array arguments tiene todos los elementos que se le pasan a la funcion por orden
        peticion_http.send (parametros);

    } //fin de if
} //fin de cargaContenido

//encapsula la creacion del objeto XMLHttpRequest
function inicializa_xhr() {
    if (window.XMLHttpRequest) {
        return new XMLHttpRequest()
    }
    else if(window.ActiveXObject) {
        return new ActiveXObject("Microsoft.XMLHTTP");
    }
}
```

Básicamente lo que se hace aquí es enviar una petición Javascript, se procesa, y se vuelve a cargar en el contenedor principal. Comentar que en todo momento el puntero correspondiente al fotograma se guarda en una cookie para que a la hora de refrescar la página, se mantenga en el mismo fotograma.

A la izquierda se distinguen los diferentes tipos de pistas, que son, pista escrita que se puede comprar por un crédito, y que contendrá o una frase famosa de la película o alguna relación general de la película.

Pista de audio, que se puede comprar por tres créditos, puede contener un trozo de banda sonora o un pequeño diálogo bastante conocido de la película. Para poder reproducir audio he utilizado un reproductor MP3 que está hecho para poder modificarse. Este reproductor se llama “dewplayer” y se puede encontrar en la siguiente página: www.alsacreations.fr/dewplayer-en.html

la última pista es la pista de video, que cuesta cinco créditos y al pulsarla se transforma en un link para ver el video. Se abre una nueva pestaña o ventana y con otro reproductor, igual que el anterior pero reproduciendo video, se puede ver unos pocos segundos representativos de la película. Este reproductor sigue la misma lógica que el anterior, ya que se puede manipular y reproducir contenidos que varían con la página. Este reproductor se llama jwplayer y se puede descargar desde la web <http://www.longtailvideo.com/players/jw-flv-player/>

Ejemplos de la vista cuando se van abriendo las pistas:



Una vista general de cómo sería en el juego una pista abierta:

La butaKa.es
siéntate y disfruta

Datos personales | Cambiar nivel | Extras | Instrucciones | Conseguir créditos | Descuentos | Sugerencias | Contacto

Ho!a Roberto 

Pista escrita 

Pista video 

Total acertadas: 15
Créditos para pistas: 97
Bonos descuento: 15





[comprobar](#)

Y por último, se muestra la nueva pestaña que se abre al elegir la pista de video:

La butaKa.es
siéntate y disfruta

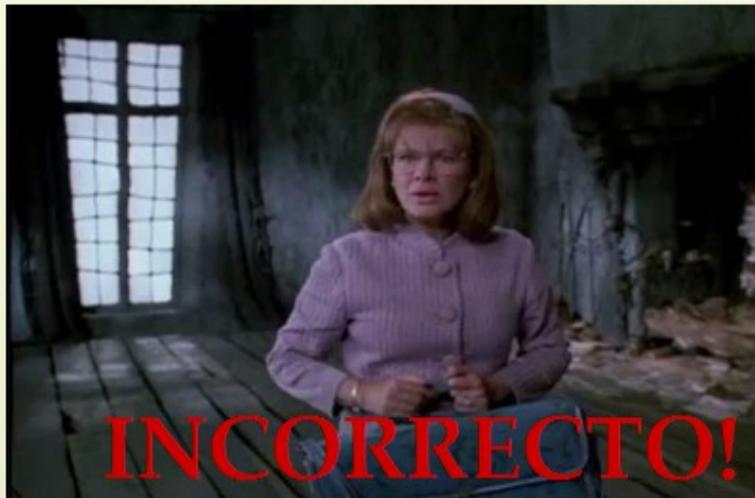


00:01 | 00:12

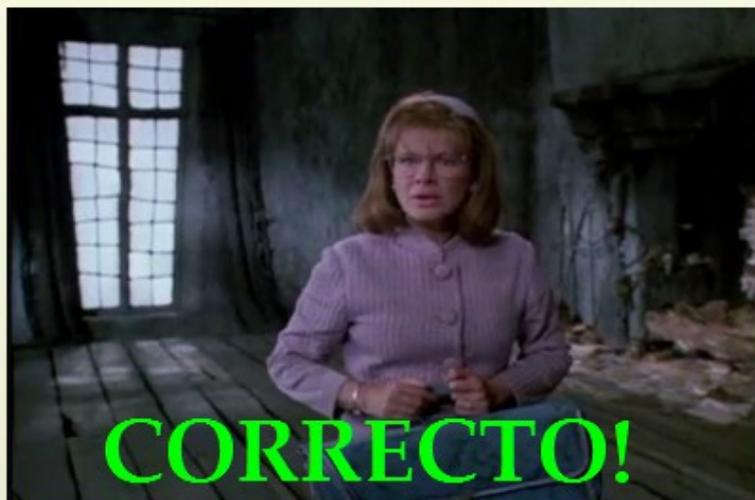
La parte central que es donde se muestra el fotograma. A ambos lados hay un botón que pasa al anterior fotograma si se pulsa el de la izquierda, y al siguiente el de la derecha. aquí está implementada una función para la que cuando se esté en el último fotograma y se pulsa siguiente vaya al primero y viceversa.



Una funcionalidad añadida es que pulsando sobre el fotograma se pasaría al siguiente. Obviamente si una pista está comprada no se muestra la imagen, sino la pista correspondiente. También hay que decir que HTML nos da la posibilidad de esconder un contenedor, que se llenará cada vez con la respuesta a si una película es correcta o incorrecta.



comprobar



eduardo manostijeras

comprobar

La parte de la derecha es la encargada de mostrar todo los marcadores, que son los créditos que quedan para comprar pistas, las películas acertadas y los créditos que se tendrían para la tienda de descuentos. Todo esto se comprueba automáticamente cada vez que se acierta una película o se compró una pista.

Hola Roberto 

Total acertadas: 15 _____

Créditos para pistas: 97 _____

Bonos descuento: 15 _____



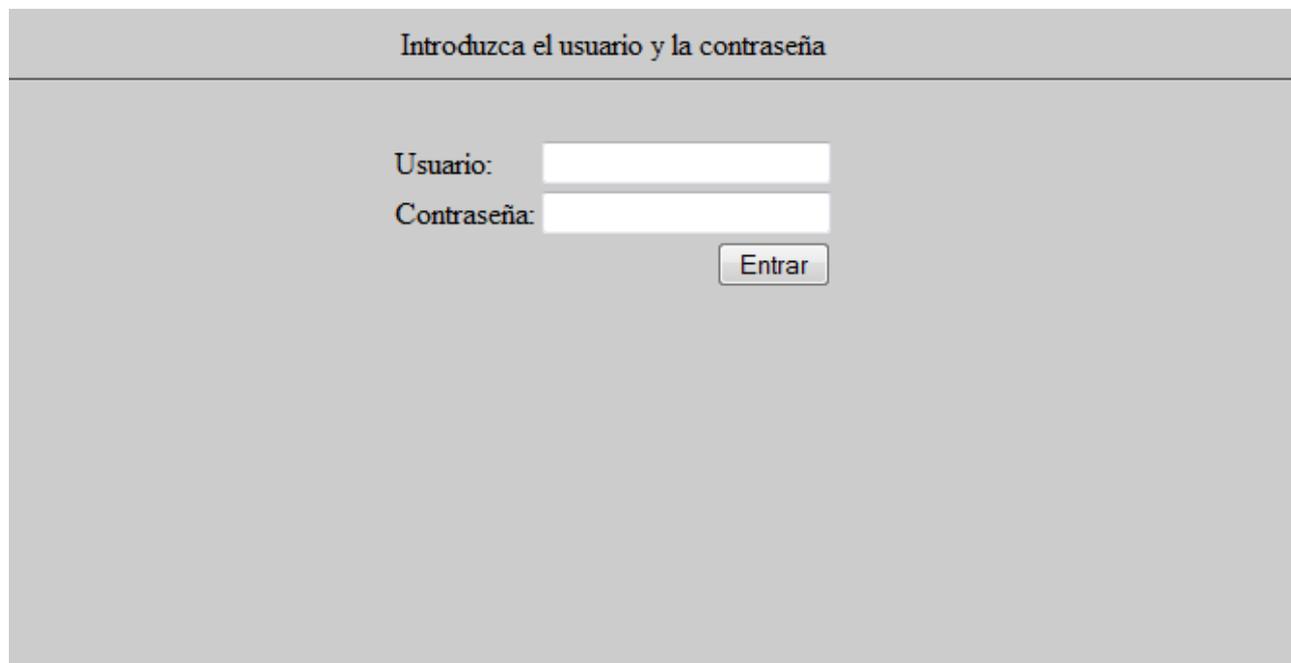
Y por último, la parte de abajo que cuenta con el espacio para introducir la respuesta. En caso de que la respuesta sea incorrecta, al pulsar el botón se escribirá en el contenedor vacío nombrado anteriormente, la palabra incorrecto. Y el espacio de la respuesta se borra automáticamente. Si es correcto, lo mostrará por pantalla, y al siguiente movimiento de la página se actualizará todo. El fotograma de la película correcta ya no se vuelve a mostrar.



eduardo manostijeras 

5.3.4 Gestión de administrador

La página de administrador no se puede acceder a no ser que se sepa la ruta exacta. Se abre un formulario HTML donde pide usuario y contraseña. Este privilegio está marcado en la base de datos. Si se introduce mal cualquiera de los campos, la página tardará 10 segundos en responder y volver''a al formulario de entrada.



Introduzca el usuario y la contraseña

Usuario:

Contraseña:

Una vez dentro, se puede elegir el tipo de registro y se pueden buscar registros. tanto por nivel como por parte del título. Todo este código lo he desarrollado yo sin ningún tipo de código externo. Arriba del todo se puede elegir el género de todos los fotogramas y automáticamente, todo tipo de búsqueda será de ese tipo de fotograma.



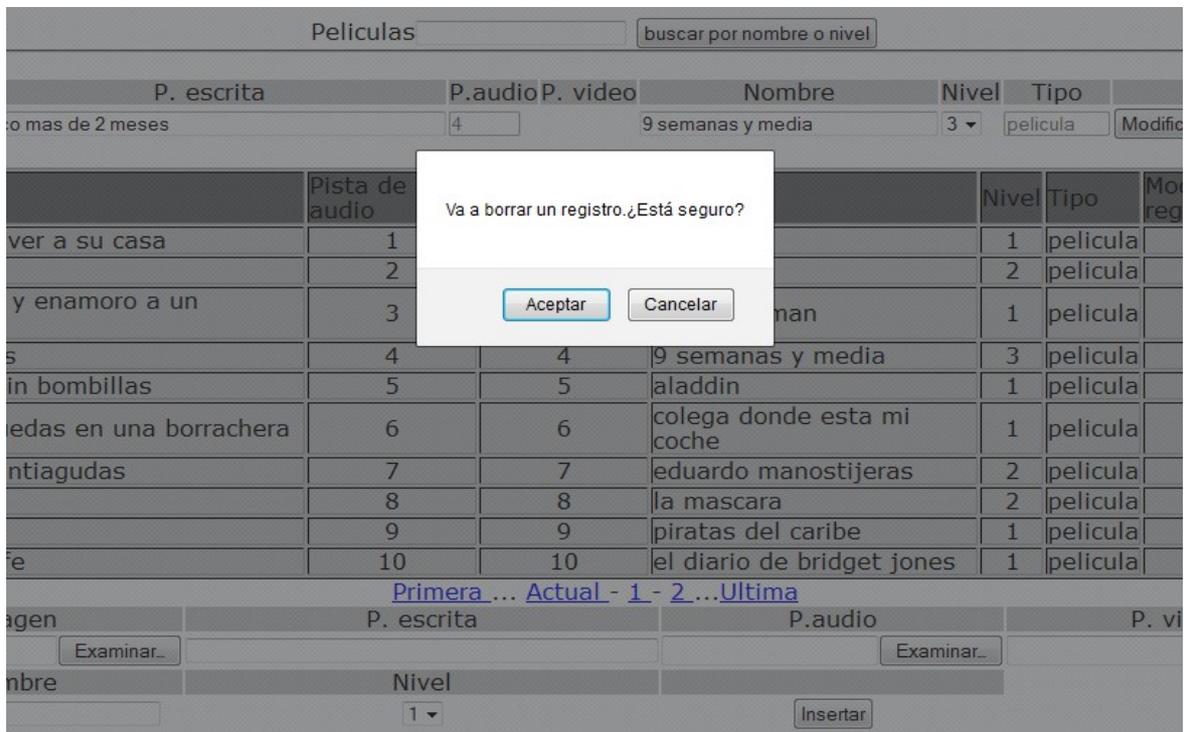
La página muestra un número máximo de registros, y en caso de que hayan más, debajo de la tabla se muestran enlaces de paginación a las dos páginas anteriores y posteriores, y a la primera y la última. Si se quiere modificar una película, hay que pulsar al lápiz, y automáticamente con una función de javascript hecha por mí se cargan los datos en la parte de arriba para que puedan ser modificados.

Código	P. escrita	P.audio	P. video	Nombre	Nivel	Tipo	
4	poco mas de 2 meses	4		9 semanas y media	3	pelicula	Modificar el registro

Código	Pista escrita	Pista de audio	Pista de video	Nombre	Nivel	Tipo	Modificar registro	Borrar registro
1	solamente queria volver a su casa	1	1	et	1	pelicula		
2	queria crecer	2	2	big	2	pelicula		
3	trabajaba en la calle y enamoro a un millonario	3	3	pretty woman	1	pelicula		
4	poco mas de 2 meses	4	4	9 semanas y media	3	pelicula		
5	usaba una lampara sin bombillas	5	5	aladdin	1	pelicula		
6	perdieron algo con ruedas en una borrachera	6	6	colega donde esta mi coche	1	pelicula		
7	tenia unas manos puntiagudas	7	7	eduardo manostijeras	2	pelicula		
8	verde	8	8	la mascara	2	pelicula		
9	buscaban una perla	9	9	piratas del caribe	1	pelicula		
10	se enamoro de su jefe	10	10	el diario de bridget jones	1	pelicula		

[Primera ... Actual - 1 - 2 ...Ultima](#)

Se puede eliminar una película pero previamente la página se advertirá y le preguntara si estás seguro.



El apartado más costoso ha sido subir y comprobar los diferentes tipos de archivo aceptados. Todo esto se comprueba en la parte del servidor para cada tipo de archivo y siempre sale un nivel de dificultad más de los existentes para qué se puedan crear nuevos niveles de dificultad.

6	perdieron algo con ruedas en una borrachera	6	6	coche	1	pelicula		
7	tenia unas manos puntiagudas	7	7	eduardo manostijeras	2	pelicula		
8	verde	8	8	la mascara	2	pelicula		
9	buscaban una perla	9	9	piratas del caribe	1	pelicula		
10	se enamoro de su jefe	10	10	el diario de bridget jones	1	pelicula		

[Primera ... Actual - 1 - 2 ...Ultima](#)

Imagen	P. escrita	P.audio	P. video
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Examinar...		Examinar...	Examinar...
Nombre	Nivel		
<input type="text"/>	<input type="text"/>		
	1		
		Insertar	

Una vez efectuada la inserción de la nueva película se procede automáticamente a la inserción de esa película en todo los usuarios que tengan abierto ese nivel, incluso aunque el nivel estuviera completo.

5.3.5 Datos personales

Este apartado es bastante parecido al de la página de inicio correspondiente al registro, de hecho se utiliza las mismas librerías y el mismo código javascript que en registro, con la salvedad de que cada vez que se quiere hacer un cambio se tiene que introducir la contraseña actual.



Hola

Cambiar datos personales

Nombre:
Apellidos:
Ciudad:
Correo:
Repita Correo:

Rellena la contraseña solamente en caso de querer cambiarla

Contraseña:
Repite contraseña:

**Introduce tu actual contraseña para realizar los cambios:*



En caso contrario, mostrará una advertencia. Si se quiere cambiar la contraseña, se deberá introducir la nueva dos veces, y además, también la anterior.

5.3.6 Niveles

En la siguiente imagen se muestra el desplegable con todos los niveles abiertos hasta el momento. Te informa de las películas que faltan para completar cada nivel que está abierto.



5.3.7 Extras

Aunque la idea original era un juego sobre películas, posteriormente para llenarlo de más contenido se decidió crear otras cosas a adivinar. Se decidieron los siguientes tipos, series, ciudades, cuadros, cómics, anuncios. En el momento en el que se pulsa sobre uno de estos contenidos, se empiezan a mostrar los fotogramas relacionados. Solamente se han añadido algunas series para comprobar el funcionamiento, pero se puede ir llenando de contenido posteriormente.



5.3.8 Instrucciones, conseguir créditos y descuentos

Instrucciones explica la funcionalidad del juego mostrado y formateado con código HTML solamente.



Volver a juegos	Datos personales	Instrucciones	Conseguir créditos	Descuentos	Sugerencias	Contacto
-----------------	------------------	---------------	--------------------	------------	-------------	----------

Cómo jugar

La forma de jugar es muy sencilla. Básicamente se trata de relacionar la imagen asociada al tipo de juego. **Por defecto son películas**, entonces las imágenes que se mostrarán serán todas de películas. El juego, en la pestaña de extras, contiene los otros tipos de objetos a adivinar como series, ciudades, cuadros... Tanto al acertar como al fallar, la página te lo advertirá.

iiiiMUY IMPORTANTE!!! En la respuesta no hay que poner ni acentos, ni comas, exclamaciones, puntos o cualquier símbolo que no sean letras. De lo contrario dará incorrecto.

Con los botones con flechitas, avanzas y retrocedes sobre todas las películas no acertadas que contiene el nivel actual de dificultad. Cuando se acierta un número concreto de películas en cada nivel, se desbloqueará el siguiente nivel de dificultad. Puedes cambiar el nivel de dificultad que hayas desbloqueado desde la pestaña "**cambiar nivel**" donde te muestra las imágenes que faltan para completar el nivel. Puedes cambiar de niveles e imágenes cuantas veces quieras.

Al acertar una película, el marcador se incrementará, tanto en acertadas como en bonos descuento. Los bonos descuento se explican en la pestaña "**descuentos**". Tendrás que escribir correctamente el nombre del objeto relacionado con la imagen, de lo contrario lo tomará como incorrecto.

A la izquierda de la pantalla encontrarás tres tipos diferentes de **pistas** que se explican en el siguiente apartado.

Solamente se puede acceder a la pestaña de "extras" y "niveles" desde la pestaña de juegos.

Cómo funcionan las pistas

Hay tres tipos diferentes de pistas y cada una tiene un valor distinto de créditos.

Primero está la **pista escrita** que cuesta adquirirla **1 crédito**. Al pulsar sobre la pista aparecerá una frase relacionada con la imagen. Si has comprado una pista escrita de una imagen, se te mostrará cada vez que estés en esa imagen, de modo que no tienes que volver a comprarla.

Después está la **pista de audio**, que cuesta **3 créditos**. Tiene las mismas características que la pista escrita, pero en vez de un texto se mostrará un reproductor de audio para que la escuches cuantas veces quieras.

Y por último está la **pista de video**, que cuesta **5 créditos**. Igual que las dos anteriores pero en este caso aparecerá un enlace que la pulsarlo se abrirá en una nueva pestaña/ventana un reproductor de video para visualizarlo cuantas veces quieras.

Hola Roberto

Conseguir créditos explica las distintas maneras de conseguir créditos, que ya hemos explicado anteriormente y descuentos será el sitio donde se empiecen a mostrar los distintos descuentos con su precio.

Datos personales	Instrucciones	Conseguir créditos	Descuentos	Sugerencias
------------------	---------------	--------------------	------------	-------------

Cómo conseguir créditos

Los **créditos** sirven para poder desbloquear los diferentes tipos de pistas de los que dispone el juego. Como los créditos son limitados, hay 4 maneras diferentes de conseguir créditos para comprar pistas.

- **La primera** forma es automática y se añade un crédito cada día diferente que el usuario entra en la página.
- **La segunda** forma es al recomendar la página a un amigo, y éste lo confirme al registrarse, introduciendo tu correo en el campo "correo del amigo". Esta acción incrementa en 20 créditos tanto al que se registra como al que recomienda.
- **La tercera** forma es cuando se consigue desbloquear un nivel de dificultad. Se incrementa en 50 créditos automáticamente el marcador.

5.3.9 Sugerencias



Sugerencias

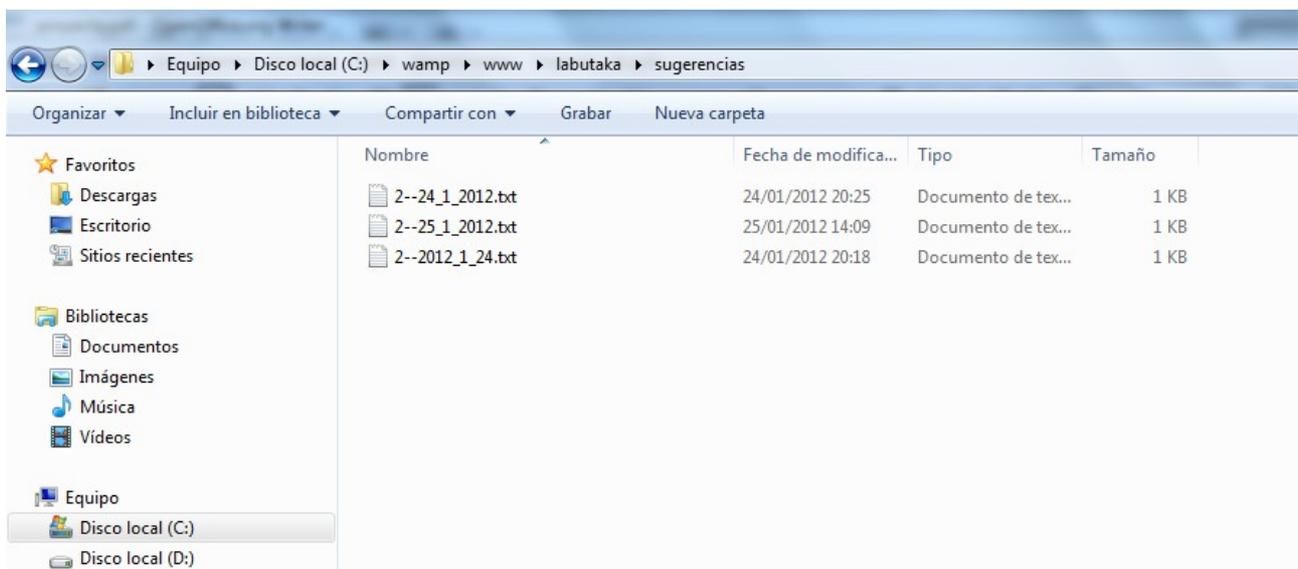
Si tienes alguna sugerencia de cómo mejorar el juego para hacerlo más emocionante, para hacer mejoras o incluso para crear otro tipo de juegos, no te lo pienses y **DINOSLO!!!**

Las **buenas ideas serán recompensadas** por los administradores.

O incluso para hacer más amigable la página, más funcional o cualquier tipo de mejora constructiva, será vista con buenos ojos.

enviar

La particularidad de esta función es que la sugerencia al ser enviada, es ejecutada por el servidor y crea en la carpeta sugerencias un nuevo archivo de texto cuyo título es el número de usuario y la fecha. Dentro del archivo de texto se muestra el nombre completo del usuario, la fecha y la sugerencia.



5.3.10 Contacto

El contacto muestra un correo de contacto esta formateado con un código HTML para que al pulsarlo se abra el servidor de correo predefinido.



Contacto

Si quiere ponerse en contacto con nosotros para cualquier gestión, envíenos un correo a esta dirección:

contacto@labutaka.es

En este apartado se explicara el contenido de las partes más importantes de la implementación, como es el registro de usuario, la gestión del menú del administrador y los procedimientos para comprobar todas las partes del juego.

La implementación se realiza con el programa Dreamweaver, con los lenguajes de programación php, html, y en algunas partes JavaScript y AJAX por añadidura.

La parte principal se llama juego.php donde se encuentran a un click, todas las partes de la página.

Hay código que se repite y es usado por varios apartados modificando solo ciertas partes, se explicara aquí ejemplos de cada uno.

5.4 Implementación de las funcionalidades.

5.4.1 Registro de usuarios

Para el registro de usuarios se emplea un código en javascript para validar los campos y luego añadirlo dentro de la base de datos.

Todas las páginas que contengan un formulario, será redirigidas a otra página con el mismo nombre, pero acabadas en “proc” donde se hará toda la validación necesaria de los datos de entrada.

Una vez validados los datos, se conectará la base de datos mediante una función y añadirá los datos.

En este código se diferencia entre un usuario existente de uno que se está creando y se hacen diversas funcionalidades dependiendo de la acción, pero acaban yendo a la página del juego.

Si es un usuario nuevo se guarda la fecha de creación, y si es un usuario registrado se comprueba el último día de acceso para ver si se le incrementan los créditos y se le actualiza la fecha de último día jugado.

```

.
//sumar creditos cuando se entra en dias distintos
$sumaCreditos="UPDATE usuario SET creditoPistas=creditoPistas+1 where codigo=$codigo";

$diaActual=date('j');
$mesActual=date('n');
$anyoActual=date('Y');
$actualizaFecha="UPDATE accesos SET dia=$diaActual, mes=$mesActual, anyo=$anyoActual where usuario=$codigo";
if($anyo<$anyoActual) mysql_query($sumaCreditos);
else if($anyo==$anyoActual) {
    if ($mes<$mesActual) mysql_query($sumaCreditos);
    else if ($mes==$mesActual) if ($dia<$diaActual) mysql_query($sumaCreditos);
}
//actualiza a la fecha de la ultima conexion
mysql_query($actualizaFecha);
Desconectarse($resultado);
//vamos a ponerle un nombre a la sesion para que se desconecte por inactividad
session_name("sesionActiva");
session_start();
//if($_POST["correo"] == "estoessunnombredesusuario" && $_POST["contrasenia"] == "estoessunacontrasena123"){
if ($habilitado==1 && $contrasenia == $comprobacion) {
    $_SESSION['codigo'] = $codigo ;
    $_SESSION['contrasenia'] = $contrasenia ;
    //esta cookie es necesaria para la primera vez que se entra

    //echo "Bienvenido ".$_SESSION['codigo'];
    $_SESSION['posicion']=$posicion;
    setcookie("nombre", $nombre);

    header("Location: juego.php");
}
else {

```

```

if (!preg_match($patronContrasenia, $contrasenia2)) {
    echo " Debes registrarte correctamente ";
    sleep(3);
    header("Location: index.php");
}

if (!preg_match($patronCorreo, $correo2)) {
    echo " Debes registrarte correctamente ";
    sleep(3);
    header("Location: index.php");
}

if (!preg_match($patronTexto, $nombre) || !preg_match($patronTexto, $apellidos)) {
    echo " Debes registrarte correctamente ";
    sleep(3);
    header("Location: index.php");
}

}

Conectarse();

$confirmar=false;
$final=false;

$insertion=" INSERT INTO usuario (habilitado, nombre, apellidos, creditos, correo, contrasenia, ciudad, acertad
nivel ) ";
$insertion=$insertion ." VALUES (1, '$nombre', '$apellidos', 0, '$correo2', '$contrasenia2', '$ciudad', 0, $amig
mysql_query($insertion);
mysql_close();

```

Cuando son correctos todos los datos se hace un submit y se redirecciona a una pagina diferente que procesa todos los datos. Sino se cumplen los patrones de los campos, dirige al index.

5.4.2 Conexión a la base de datos.

Se muestra la conexión a la base de datos y se hace una función para conectarse y mostrar errores si los hubiera, y otra para desconectarse y liberar recursos.

```
function Conectarse() {  
  
if (!($conexion=mysql_connect("localhost", "labutaka", "XXXXXXXXXX")))  
{  
    echo "Error al conectarse a la Base de Datos";  
    exit();  
}  
if (!mysql_select_db("labutaka", $conexion))  
  
{  
    echo "Error al seleccionar la base de datos."<br>;  
    echo "Número de error: ".mysql_errno($conexion)."<br>;  
    echo "Mensaje de error:".mysql_error($conexion)."<br>;  
    exit();  
}  
  
return $conexion;  
}  
  
function Desconectarse($resultado) {  
mysql_free_result($resultado);  
mysql_close();  
}
```

5.4.3 Administrador

Pasamos al código administrador en el que podemos resaltar en este código la subida de archivos y la comprobación de los tipos, si se inserta o se actualiza una película, o mostrar un error si hubiera ocurrido un percance.

Para facilitar la búsqueda, los archivos se renombran al final con el mismo número ingresado en la película.

```

conectarse ();
$consulta="update peliculas set texto='$escrita' , nombre='$nombre' , nivel=$nivel where codigo=$codigoPeli";
echo $consulta;
mysql_query($consulta);
echo mysql_error();

mysql_close();

} //fin de accion modificar

else if($accion=="insertar") {
    $nivel=$_POST['nivel'];
    $nombre=$_POST['nombre'];
    //para poner en minusculas
    $nombre=strtolower($nombre);
    $escrita=$_POST['escrita'];

    $uploadImagen = "imagenes/imagen$maximo.png";
    $uploadAudio = "sonidos/sonido$maximo.mp3";
    $uploadVideo = "videos/video$maximo.flv";

    //$uploadfile = $uploaddir . "1";
    $errorImagen = $_FILES['imagen']['error'];
    $errorAudio = $_FILES['audio']['error'];
    $errorVideo = $_FILES['video']['error'];

    //if($_FILES['imagen']['type']=="image/jpeg" && $_FILES['audio']['type']!="video/mpeg" && $errorImagen==UPLOAD_ERR_OK &&
    $errorAudio==UPLOAD_ERR_OK && $errorVideo==UPLOAD_ERR_OK) {
        if( $errorImagen==UPLOAD_ERR_OK && $errorAudio==UPLOAD_ERR_OK && $errorVideo==UPLOAD_ERR_OK) {

            $subidoImagen = false;
            $subidoAudio = false;
            $subidoVideo = false;

            $subidoImagen = copy($_FILES['imagen']['tmp_name'], $uploadImagen);
            $subidoAudio = copy($_FILES['audio']['tmp_name'], $uploadAudio);
            $subidoVideo = copy($_FILES['video']['tmp_name'], $uploadVideo);

            if($subidoImagen && $subidoAudio && $subidoVideo) {

                conectarse ();
                $consulta="insert into peliculas (codigo, texto, audio, video, nombre, tipo, nivel) ";
                $consulta=$consulta ." VALUES ($maximo, '$escrita', $maximo, $maximo, '$nombre', $tipo, $nivel) ";
                echo $consulta;
                mysql_query($consulta);
                echo mysql_error();

                $consulta=" insert into usuariopeli (usuario, pelicula, acertada, pistaEscrita, pistaAudio, pistaVideo) ";
                $consulta=$consulta ."SELECT distinct usuario, $maximo, 0, 0, 0 from usuariopeli where pelicula in (select codigo
from peliculas where nivel=$nivel)";
                mysql_query($consulta);
                echo mysql_error();
                mysql_close();

            } else {
                echo "Se ha producido un error: ".$errorImagen;
                echo "Se ha producido un error: ".$errorAudio;
                echo "Se ha producido un error: ".$errorVideo;
            }
        } // fin de comprobacion de tipos
    } else
        echo "los archivos no eran los correctos o no se han subido bien";
}

```

5.4.4 Principal

Este es el contenedor principal donde transcurre el juego. Básicamente es una página que por medio de AJAX, está introducida en un contenedor de la página principal.

Todo lo que ocurre aquí, gracias a la conexión asíncrona al servidor que nos ofrece AJAX, hace que se consiga un efecto de que nunca se recarga la página.

En el código se puede ver cómo se muestran o esconden las pistas para cada fotograma. El código no está entero, pero después vendría todo lo relacionado a la comprobación de las respuestas, a los bucles para mostrar los fotogramas y la s gestiones de los diferentes tipos de créditos.

```
- -
if ($contador>=1){
//esto es para comprobar previamente si se tiene dinero para comprar pistas, quitando el enlace si no las tiene
//pista escrita
echo "<table><tr><td height=\"120px\" valign=\"middle\" width=\"200px\" align=\"center\" >";
if ($pistaEscrita){
echo "\&#x201c;\$texto\&#x201c;";
}
//LA VARIABLE GET VALOR ES LA QUE DECIDE EN COMPRARPISTA LA PISTA QUE SE HA COMPRADO
else if ($credito>=1){
echo "<a href=\"comprarPista.php?dato=$posicion&valor=1\" class=\"enlace\"><img src=\"../images/imagenEscrito.png\"
name=\"pistaEscrita\"></a>";
//echo "<img src=\"../images/imagenEscrito.jpg\" onclick=\"actualizarMarcador(1)\" width=\"120\" height=\"50\" name=\"pistaEscrita\"";
";
}
else
echo "<img src=\"../images/imagenEscritoNo.png\" name=\"pistaEscrita\"";

echo "</td></tr>";
echo "<tr ><td valign=\"middle\" nowrap height=\"120px\" align=\"center\" >";
//pista de audio. habria que confirmar en BBDD
if ($pistaAudio){
//se ponen las barras invertidas para que php no las procese
echo "<object type=\"application/x-shockwave-flash\" data=\"/reproductores/dewplayer-vol.swf?mp3=../sonidos/sonido.\$posicion.mp3\"
width=\"240\" height=\"20\" id=\"dewplayer-vol\"><param name=\"wmode\" value=\"transparent\" /><param name=\"movie\"
value=\"/reproductores/dewplayer-vol.swf?mp3=../sonidos/sonido.\$posicion.mp3\" /></object>";
}
else if ($credito>=3) {
echo "<a href=\"comprarPista.php?dato=$posicion&valor=2\" class=\"enlace\"><img src=\"../images/imagenAudio.png\"
name=\"pistaAudio\"></a>";
}
else
echo "<img src=\"../images/imagenAudioNo.png\" name=\"pistaAudio\"";
echo "</td></tr>";

echo "<tr><td width=\"200px\" valign=\"middle\" height=\"120px\" align=\"center\" >";

//pista de video. habria que confirmar en BBDD
if ($pistaVideo) {

echo "<br>Pulse <a href=\"../pista_video.php?video=.\$posicion.\" target=\"_blank\">aquí; </a> para ver la pista de video. Se
abre en nueva ventana/pestañtilde;a";
} //fin de if de video
else if ($credito>=5) {
echo "<a href=\"comprarPista.php?dato=$posicion&valor=3\" class=\"enlace\"><img src=\"../images/imagenVideo.png\"
name=\"pistaVideo\"></a>";
}
else
echo "<img src=\"../images/imagenVideoNo.png\" name=\"pistaVideo\"";

echo "</td></tr></table>";
} //fin de If contador
//aloo echo "esta nalicula está ocortada".
```

5.4.5 Sugerencias

En este trozo de código se puede observar cómo se cargan todas las librerías hechas por mí, y a continuación, el procesamiento que lleva a cabo la inserción de la sugerencia en un archivo de texto en el servidor.

```
<?php
sleep(10);
include_once("top.php");
//el menu de las paginas sin ajax
include_once("menuOtros.php");
include_once("conexionBD.php");

$sugerencia=$_POST['sugerencia'];

?>
<div class="cajaCentral">
  <p><br />
  <br />
  <br />
  
</p>
<?php

$diaActual=date('j');
$mesActual=date('n');
$añoActual=date('Y');
$fecha=$diaActual.'_'.$mesActual.'_'.$añoActual;
$nombreArchivo=$codigo."--".$fecha;
$tituloArchivo=" el dia ".$fecha." ha escrito: ";
$sugerencia=$tituloArchivo.$sugerencia;

$archivo = fopen ("sugerencias/$nombreArchivo.txt", "w+");
fwrite($archivo, $sugerencia);
fclose($archivo);
```

5.4.6 Comprobar respuestas

Se recoge el texto enviado por el usuario y se comprueba con el número de fotograma actual. Si es correcto se actualizan todos los datos necesarios.

```
session_name("sesionActiva");
session_start();
$codigo=$_SESSION['codigo'];

include_once("conexionBD.php");
Conectarse();
$consulta="SELECT nombre, acertada FROM peliculas p, usuariopeli u WHERE p.codigo=$foto and p.codigo=u.pelicula and u.usuario=$codigo"
;
$resultado=mysql_query($consulta);
$comprobacion=0;
$acertada=0;
while ($fila=mysql_fetch_array($resultado)) {
$comprobacion= $fila['nombre'];
$acertada= $fila['acertada'];
}
//Desconectarse($resultado);

if ($valor==$comprobacion) {

if ( $acertada==0 ) {
//añado la pelicula acertada a la base de datos
$consulta="UPDATE usuariopeli SET acertada=1 WHERE usuario=$codigo && pelicula=$foto ";
//añado 1 a peliculas acertadas
$consulta2="UPDATE usuario SET acertadas=acertadas+1 WHERE codigo=$codigo ";
//añado 1 a los creditos para comprar descuentos
$consulta3="UPDATE usuario SET creditos=creditos+1 WHERE codigo=$codigo ";

mysql_query($consulta);
mysql_query($consulta2);
mysql_query($consulta3);
```

5.4.7 Comprar pistas

Se recibe una petición de compra de una pista, y dependiendo de la clase de pista se descuenta un diferente número de créditos.

Decir que previamente se comprueba que se tienen suficientes créditos, ya que en la página principal se realiza un cálculo y si no se tienen suficientes créditos para comprar una pista, en vez de un enlace a comprar pistas, se muestra una imagen advirtiéndolo.

```
//el valor es el tipo de pista que compramos
//1. para escrito
//2.para audio
//3.para video
$valor= $_GET['valor'];
include_once("conexionBD.php");
// me conecto...
Conectarse();
//HABRIA QUE HACER OTRA COMPROBACION CON CREDITOPISTAS POR SI TIENE SUFICIENTES CREDITOS YA QUE PUEDE MODIFICAR EL GET
if($valor==1){
    //$creditoPistas=$creditoPistas-1;
    //echo $creditoPistas;
    $consulta="UPDATE usuariopeli SET pistaEscrita=1 WHERE usuario=$codigo and pelicula=$peli ";
    $marcador="UPDATE usuario SET creditoPistas=creditoPistas-1 WHERE codigo=$codigo";
}
else if ($valor==2){
    //$creditoPistas=$creditoPistas-3;
    //echo $creditoPistas;
    $consulta="UPDATE usuariopeli SET pistaAudio=1 WHERE usuario=$codigo and pelicula=$peli ";
    $marcador="UPDATE usuario SET creditoPistas=creditoPistas-3 WHERE codigo=$codigo";
}
else { //$valor==3
    //$creditoPistas=$creditoPistas-5;
    //echo $creditoPistas;
    $consulta="UPDATE usuariopeli SET pistaVideo=1 WHERE usuario=$codigo and pelicula=$peli ";
    $marcador="UPDATE usuario SET creditoPistas=creditoPistas-5 WHERE codigo=$codigo";
}
mysql_query($consulta);
mysql_query($marcador);
//echo mysql_error();
mysql_close();
//Desconectarse($resultado);
```

5.4.8 Ver video

Cuando se pulsa en ver un video, como este dato se pasa por GET, un usuario podría cambiar el número de la cabecera y ver todas las pistas de video que quisiera. Este código evita que esto suceda porque siempre se comprueba si el usuario tiene esa pista comprada. El final del código es código externo, necesario para poder ver los videos .

```
include_once("conexionBD.php"); //linea 20
// me conecto...
Conectarse();
//para comprobar si de verdad la pista está comprada, porque como se pasa por GET, pueden cambiar la URI y acceder a los otros videos
$consulta="SELECT pistaVideo FROM usuariopeli WHERE usuario=$codigo && pelicula=$video ";
$resultado=mysql_query($consulta);

$comprada=0;
while ($fila=mysql_fetch_array($resultado)) {
    $comprada= $fila['pistaVideo'];
}
Desconectarse($resultado);

//las siguientes lineas donde se almacenan los valores para que el video los recupere con javascript
if ($comprada){
    echo "<param id=\"valor\" value=\"".$video."\">";
}
else {
    echo "<param id=\"valor\" value=\"0\">";
}
?>
<script type='text/javascript'>
var video=document.getElementById("valor").value;
var so = new SWFObject('../reproductores/player.swf','mpl','470','320','9');

so.addParam('allowfullscreen','true');
so.addParam('allowscriptaccess','always');
so.addParam('wmode','opaque');
so.addVariable('file','../videos/video'+video+'.flv');
so.addVariable('frontcolor','000066');
so.write('mediaspace');
</script>
```

5.4.9 Desconectarse

Básicamente, se destruyen todas las variables de sesión existentes y también las cookies. Después de esto te redirige a la página de inicio.

```
<?php
//necesario para que puedas cerrar la sesion decirle que la continuas con este script
session_name("sesionActiva");
session_start();

//vacía las cookies asociadas a la sesion
unset($_SESSION['codigo']);
unset($_SESSION['contrasena']);
unset($_SESSION['posicion']);

//cierra la sesion

session_destroy();

//borrar las cookies que hemos utilizado
//setcookie("posicion", "aquí va un valor", time()-60);
setcookie("nombre", "aquí va un valor", time()-60);

//para redirigir a la pagina de inicio

header("Location: index.php");
?>
```

5.4.10 Ajax

Por último, pongo el código entero de la parte AJAX para que se pueda observar que todo el código ha sido escrito por mi, sin ningún tipo de librería.

A destacar puedo decir que se maneja el tiempo que tarda el servidor en responder y muestra un error si se sobrepasa el límite.

Este código funciona con todos los navegadores, incluso con el “independiente” Internet Explorer.

Otro dato a comentar es que aparte de todas las funcionalidades comentadas anteriormente, también se encarga de vaciar el contenido de las respuestas cuando una respuesta es incorrecta, así como de mostrarlo por pantalla.

```

var READY_STATE_UNINITIALIZED = 0;
var READY_STATE_LOADING = 1;
var READY_STATE_LOADED = 2;
var READY_STATE_INTERACTIVE = 3;
var READY_STATE_COMPLETE = 4;

//var direccion="http://localhost/";
var direccion="";
var contiene;

var pagina = "juego.php";
var segundos = 2000; //en realidad son milisegundos

//Variable global que almacena el identificador de la cuenta atras
var cuentaAtras = null;
var tiempoMaximo = 10000; //5000= 5 segundos

var peticion_http;

function cargaContenido(url, metodo, parametros) {

    peticion_http = inicializa_xhr();

    if (peticion_http) {

        //establecer la cuenta atrás al realizar la peticion HTTP
        cuentaAtras = setTimeout(expirada, tiempoMaximo);
        peticion_http.onreadystatechange = muestraContenido;
        peticion_http.open(metodo, url, true);

        //la siguiente linea es necesaria para poder enviar parametros
        peticion_http.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        //el array arguments tiene todos los elementos que se le pasan a la funcion por orden
        peticion_http.send (parametros);

    } //fin de if
} //encapsula la creación del objeto XMLHttpRequest
function inicializa_xhr() {
    if (window.XMLHttpRequest) {
        return new XMLHttpRequest()
    }
    else if(window.ActiveXObject) {
        return new ActiveXObject("Microsoft.XMLHTTP");
    }
} //fin de inicializa

function muestraContenido() {

    if(peticion_http.readyState == READY_STATE_COMPLETE) {
        if(peticion_http.status ==200) {
            //si se ha recibido la respuesta del servidor, eliminar la cuenta atras
            clearTimeout(cuentaAtras);
            //llamo a la funcion que rellenará en según que contenedor

            llenaContenido();
        }
    }
} //fin de muestra contenido

function llenaContenido() {
    if(contiene=="contenido") {
        document.getElementById("contenido").innerHTML = peticion_http.responseText;
        document.getElementById("respuesta_procesada").innerHTML= "";
    }
    else if (contiene=="respuesta") {
        if(peticion_http.responseText=="si") {
            document.getElementById("respuesta_procesada2").innerHTML= "CORRECTO!";
            //para redirigir la pagina y que se recargue el marcador. Anulado porque sino habi:
            la cookie
            //setTimeout("redireccion()",segundos);
        }else {
            document.getElementById("respuesta_procesada").innerHTML= "INCORRECTO!";
            document.getElementById("respuesta_procesada2").innerHTML= "";
        }
    }
}

```

```

        document.respuestaForm.respuesta.value="";
    } //fin de la comprobacion de la pelicula
} //fin de else if
else if(contiene=="marcador") {
    document.getElementById("marcador").innerHTML = peticion_http.responseText;
}
else {
    document.getElementById("respuesta_procesada2").innerHTML= "CORRECTO!";
}
} //fin de llenaContenido

function expirada() {
    //la cuenta atras se ha cumplido, detener la peticion HTTP pendiente
    peticion_http.abort();
    alert("El servidor tarda en responder. Inténtalo un poco más tarde.");
}

function descargaArchivo(url, metodo, parametros, contenedor) {
    contiene=contenedor;
    if (arguments.length==2 || arguments[2]==null){
        cargaContenido(direccion+url, metodo, "null");
    }
    else if(contenedor=="contenido") {
        //el no cache del final se usa para evitar problemas con la caché de los navegadores. coge u
use la cache
        cargaContenido(direccion+url, metodo, parametros+"&nocache="+Math.random());
    }
    else if(contenedor=="marcador") {
        cargaContenido(direccion+url, metodo, parametros+"&nocache="+Math.random());
    }
    else {
        cargaContenido(direccion+url, metodo, parametros+"&nocache="+Math.random());
    }
} //fin de descargaArchivo

//es la funcion que hace cambiar las peliculas
function lanzadera(valor) {
//es la funcion que hace cambiar las peliculas
function lanzadera(valor) {
descargaArchivo("principal.php", "POST", "posicion="+valor, "contenido");
}

/*function lanzaderaSiguiente(siguiente){
descargaArchivo("principal.php", "POST", "posicion="+siguiente, "contenido");
}*/
//vacía el campo del formulario de la comprobacion de la respuesta
function vaciar() {
//document.respuestaForm.respuesta.value="";
}

function comprobarRespuesta(posicion) {
if (document.respuestaForm.respuesta.value.length!=0) {
descargaArchivo("comprobar_respuesta.php", "POST", "dato="+document.respuestaForm.respuesta.value+"&
}
//el return se pone para que no se envíe nunca el formulario
return false;
}

function redireccion() {
    document.location.href=direccion+pagina;
}

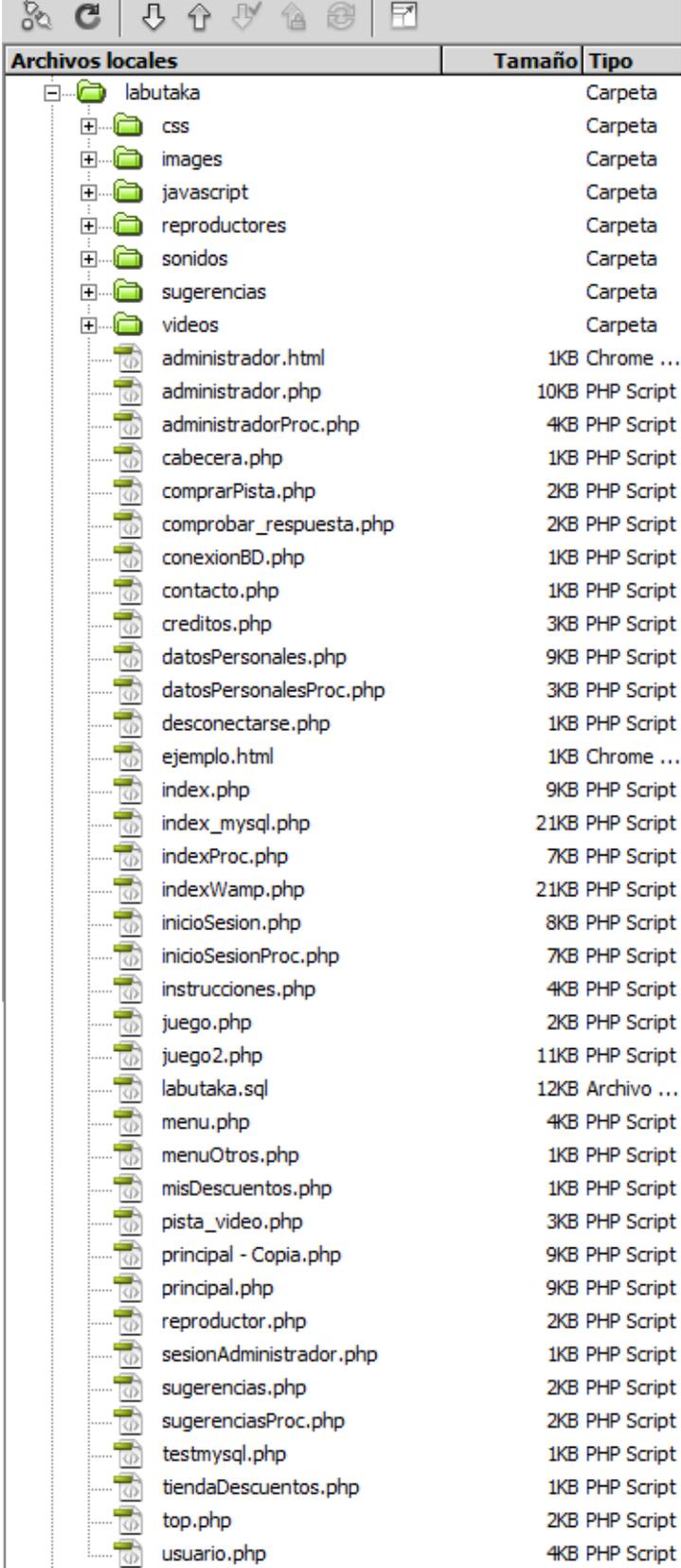
function restaurar () {
    document.getElementById("respuesta_procesada").innerHTML= "";
    //alert ("hola");
}

//cambia la vista del boton de enviar

```

5.4.11 Directorios

Por último, muestro toda la colección de archivos y carpetas que he creado para poder llevar a cabo este proyecto. En mi opinión siempre se pueden mejorar y añadir cosas, pero para ser el primer proyecto de este tipo, se han tocado bastantes temas y ha sido bastante completa la realización.



Archivos locales	Tamaño	Tipo
labutaka		Carpeta
css		Carpeta
images		Carpeta
javascript		Carpeta
reproductores		Carpeta
sonidos		Carpeta
sugerencias		Carpeta
videos		Carpeta
administrador.html	1KB	Chrome ...
administrador.php	10KB	PHP Script
administradorProc.php	4KB	PHP Script
cabecera.php	1KB	PHP Script
comprarPista.php	2KB	PHP Script
comprobar_respuesta.php	2KB	PHP Script
conexionBD.php	1KB	PHP Script
contacto.php	1KB	PHP Script
creditos.php	3KB	PHP Script
datosPersonales.php	9KB	PHP Script
datosPersonalesProc.php	3KB	PHP Script
desconectarse.php	1KB	PHP Script
ejemplo.html	1KB	Chrome ...
index.php	9KB	PHP Script
index_mysql.php	21KB	PHP Script
indexProc.php	7KB	PHP Script
indexWamp.php	21KB	PHP Script
inicioSesion.php	8KB	PHP Script
inicioSesionProc.php	7KB	PHP Script
instrucciones.php	4KB	PHP Script
juego.php	2KB	PHP Script
juego2.php	11KB	PHP Script
labutaka.sql	12KB	Archivo ...
menu.php	4KB	PHP Script
menuOtros.php	1KB	PHP Script
misDescuentos.php	1KB	PHP Script
pista_video.php	3KB	PHP Script
principal - Copia.php	9KB	PHP Script
principal.php	9KB	PHP Script
reproductor.php	2KB	PHP Script
sesionAdministrador.php	1KB	PHP Script
sugerencias.php	2KB	PHP Script
sugerenciasProc.php	2KB	PHP Script
testmysql.php	1KB	PHP Script
tiendaDescuentos.php	1KB	PHP Script
top.php	2KB	PHP Script
usuario.php	4KB	PHP Script

5.4.12 Uso de código externo.

Se ha utilizado únicamente código externo para poder reproducir tanto las pistas de audio con un reproductor basado en HTML y Javascript y las pistas de video, con otro reproductor hecho con las mismas herramientas, pero de diferentes compañías.

5.4.13 Implementación de la Interfaz gráfica.

Se ha implementado hojas de estilo en cascada, (CSS) para dar una apariencia similar en todos los apartados, este tipo de hoja se a utilizado en todos los apartados de forma dinámica haciendo referencia al archivo en concreto “style.css”, excepto en algunos donde se han implementado estáticamente el diseño grafico ya que se han modificado pequeñas cosas que no merecía la pena crearlas en el archivo general.

Como el archivo donde guardo todos los estilos era bastante extenso, en la imagen de abajo solamente muestro el código relacionado con el menú desplegable que está en la cabecera, donde se tienen todas las partes del juego en un sólo click.

```

#menu { text-align: center;
font-size: 0.7em;
width: 100%;
height: 30px;
background-color: #000;
//background: url(../images/barraCon.jpg);
}
#menu ul { list-style-type: none; }
#menu ul li.nivell1 { float: left;
width: 132px;
margin-right: 2px;
}
#menu ul li a {display: block;
text-decoration: none;
color: #fff;
background-color: #000;
//background: url(../images/barraCon.jpg);
border-right: solid 1px #666;
//border-left: solid #666;
padding: 8px;
position: relative;
}
#menu ul li: hover {position: relative;
}
#menu ul li a: hover, #menu ul li: hover a.nivell1 {
background-color: #0C0;
//background: url(../images/barraSin.jpg);
color: #FFF;
position: relative;
}
#menu ul li a.nivell1 {display: block !important; display: none;
position: relative;
}
#menu ul li ul {display: none;
}
#menu ul li a: hover ul, #menu ul li: hover ul {display: block;
position: absolute; left: 0px;
}
#menu ul li ul li a {width: 132px;
padding: 6px 0px 8px 0px;
border-top-color: #FFF;
}
#menu ul li ul li a: hover {border-top-color: #FFF;
position: relative;
}
table.falsa {border-collapse: collapse;
border: 0px;
float: left;
position: relative;
}

```

5.4.14 Implementación de la gestión de datos.

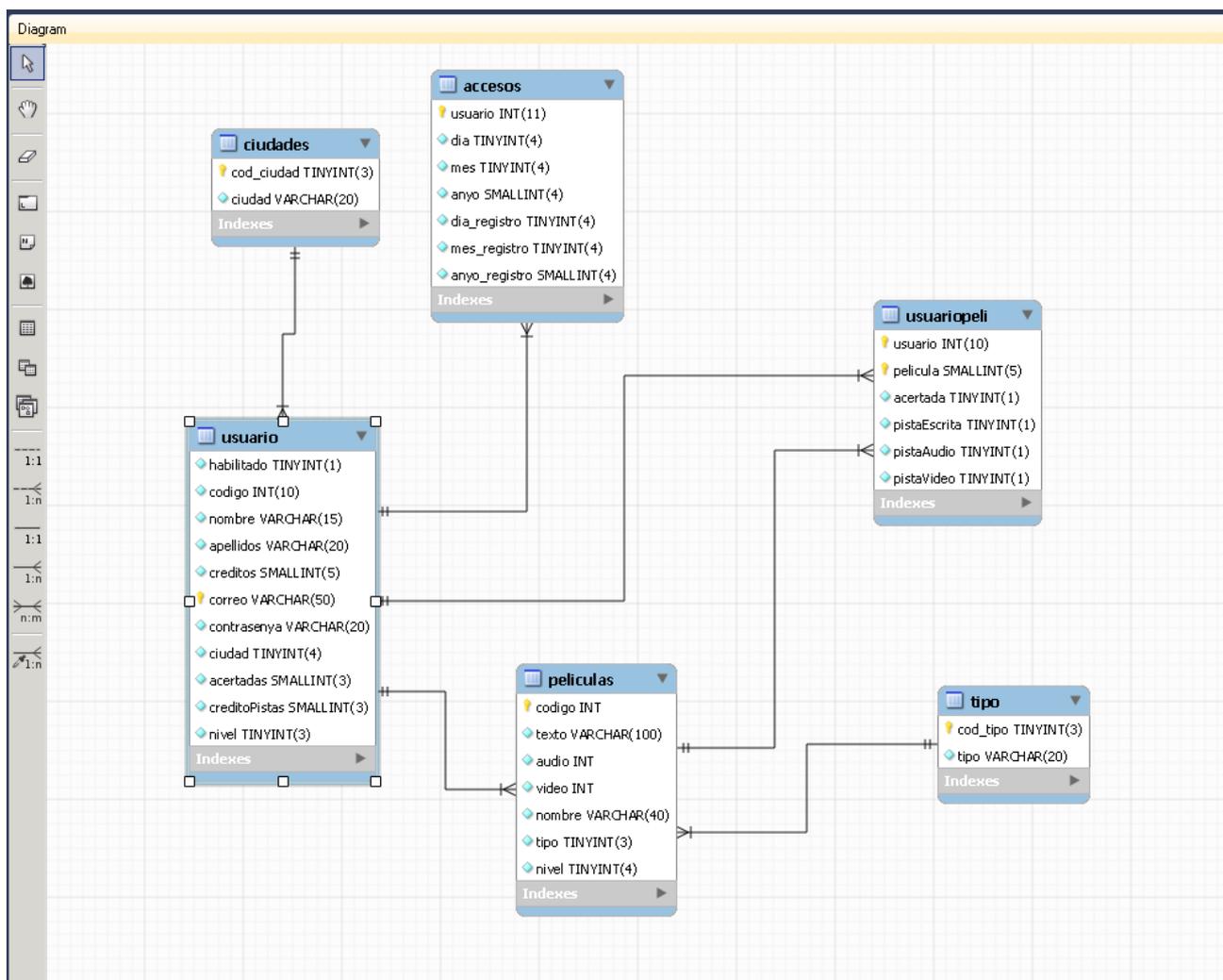
Para almacenar los datos de la aplicación he utilizado la una base de datos en MySQL donde se quedaran guardados permanentemente.

La herramienta utilizada es phpMyAdmin, creando una base de datos con un usuario administrador, siendo este usuario el único que puede acceder a esta base de datos para realizar cualquier consulta.

Así como para gestionar la base de datos se hace muy sencillo manejar PHPMyAdmin, para crearla he utilizado una herramienta, también muy útil, llamada MySQL Workbench. Se puede hacer de todo, crear claves ajenas, decir todos los tipos de campos y sus longitudes, definir la clave primaria, hacer tablas con un sólo click poniendo muchos parámetros por defecto.

En conclusión, una herramienta muy intuitiva y que recomiendo a todo el mundo.

A continuación se muestran todas las tablas del proyecto hechas con esta herramienta.

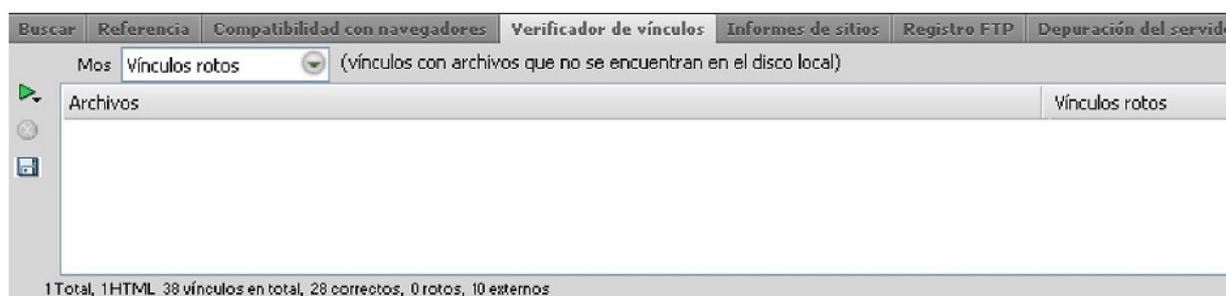


6. Evaluación y pruebas.

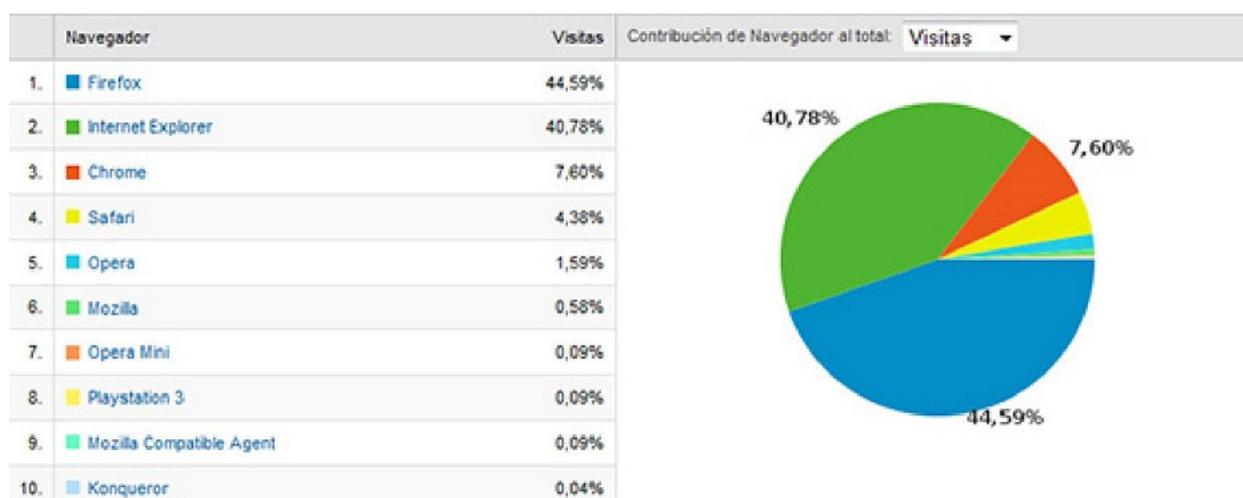
Ya terminada la implementación de código y de la interfaz gráfica, es el momento de evaluar y comprobar su funcionamiento, que sea el correcto y cumpla los estándares de la World Wide Web.

6.1. Pruebas

Se ha pasado por el depurador de Dreamweaver para encontrar posibles links rotos alguna etiqueta mal cerrada, etc, este depurador no detecta errores de programación ya que depende del propio programador.



En cuanto al modo de funcionalidad en diferentes navegadores Web, se ha centrado el portal en Firefox desde la 3.2 hasta su ultima versión 3.6, también se a probado en navegadores opera desde su versión 9, hasta la 10.10, chrome desde la versión 6 hasta la 7, Internet Explorer versiones 7 y 8.



Otra forma de poder observar el tipo de visualización en los navegadores actuales es en la pagina Web www.browsershots.org, donde te devuelve en forma de capturas de imagen las diferentes formas en que los navegadores muestran en inicio de la pagina.

Funcionamiento en navegadores más comunes

en todos los navegadores se ha conseguido que se vea prácticamente igual, al estar la página bien estructurada y haber seguido los estándares más comunes usados por estos navegadores. Los navegadores en los que se ha comprobado el funcionamiento han sido Mozilla Firefox, Internet Explorer 8 y Google Chrome

En Internet Explorer 8 tuve un pequeño problema con las imágenes que funcionaban como enlaces. Mientras en otros navegadores las imágenes se veían perfectamente en Internet Explorer aparecía un marco descubriendo que era un enlace. Esto se ha solucionado con una característica de las hojas de estilo aplicadas a todas las imágenes que enlazaban a cualquier otro apartado.

Resoluciones.

En cuanto a las resoluciones, se ha usado una herramienta Web gratuita <http://viewlike.us/> donde se le puede indicar que tipo de resolución queremos probar.

He probado varias resoluciones mas comunes de los usuarios siendo la mas optima a partir de 1024x768, en el supuesto de usar una resolución mayor no habría problemas pero en caso de usarla por de bajo aparecerían las flechas de desplazamiento.

Resolución 800x600

En esta resolución haría falta usar las barras de desplazamiento seria mas incomodo navegar por la pagina aunque esta resolución apenas se esta usando en la actualidad, ya que seria para monitores de 14 o 15 pulgadas.

Resolución 1024x768

Con esta resolución se puede apreciar toda la pagina Web sin necesidad de barras de desplazamiento por tanto a partir de aquí se podría ver de forma optima.

Resolución 1440x900

Esta es la resolución mas común en monitores actuales, a partir de un monitor de 17 pulgadas esta seria la resolución optima.

6.2. Análisis de los resultados.

Después de todas las pruebas haremos un breve resumen de cada prueba obtenida.

Navegadores

Desde el principio se ha estado diseñando para el navegador Firefox, todas las pruebas y comprobaciones de toda la programación se a hecho desde este navegador. Una vez finalizado, comprobé que en todos los demás navegadores funcionaban de la misma forma exceptuando el Internet Explorer 8 teniendo problemas en los enlaces de las imágenes.

En un principio para crear las sesiones de los usuarios se utilizó las funciones sesión, (`$_SESSION["variable"]`), donde funcionaba en todos los navegadores.

Otro de los aspectos era el tema del CSS, las visualizaciones no eran correctas en estos dos tipos de navegadores y se tuvo que redefinir varias clases del CSS añadiendo algún parámetro para que la visualización fuera correcta en los dos tipos de navegadores.

Resoluciones de pantalla.

En un principio no tuve en cuenta la resolución de pantalla usando la que tengo normalmente en mi pantalla 1440x900, el diseño estaba muy avanzado para volver a redefinirlo, pero tras las pruebas ví que desde una resolución de 1024x768 eran óptimas y mostraban todo el contenido, siendo estas las más usadas por los usuarios. Después de buscar encontré que para los atributos de los `<DIV>`, de las `<TABLE>` de las imágenes `` se podían usar en lugar de tamaños relativos, el uso de porcentajes de esta forma en cualquier navegador con cualquier resolución se vería de forma correcta. Sin ninguna duda en futuros proyectos será una de las cosas que tendré en cuenta.

Validación de HTML y CSS.

Aquí también se obtuvieron errores de validación, como el cierre de algunas etiquetas, la incompatibilidad de ciertas funciones que estaban deprecated (en fase de transición a la desaparición), y por el html versión 4 transaccional.

7. Conclusiones.

En cuanto a la conclusión del proyecto, a nivel personal son satisfactorias. Se ha invertido mucho tiempo e investigación para conseguir hacer ciertas cosas de la funcionalidad de la web, ha sido un esfuerzo el cual a merecido la pena, ya que he aprendido bastantes cosas de los lenguajes de programación que he utilizado aquí. Bien es cierto que actualmente podría realizar una página Web más vistosa y con más funcionalidades.

Sin embargo es una página que tendría mejor rendimiento cambiando pocas cosas o mejorando otras, e incluso sin cambiar nada de la programación y solo los datos.

7.1. Valoración personal

El proyecto consistía en realizar una página Web desde cero, aplicando conocimientos que se han ido adquiriendo durante la carrera e innovar en algo que no se ha utilizado en esta como es el php, JavaScript o AJAX.

Como es lógico he encontrado ciertas dificultades en ciertos puntos más que en otros por ejemplo mantener la película actual cuando se refrescaba la página que es solucionado con cookies o usando AJAX, ya que hay muchas cosas a tener en cuenta y lo que te devuelve no se comporta a veces como código normal. En cuanto al desarrollo gráfico, el lenguaje html, ya contaba con ciertos conocimientos como en la asignatura de CDH el cual me ayudó.

En otras situaciones como la programación php o javascript a sido de investigación propia a base de leer manuales por Internet, sin embargo estos lenguajes tienen cierta semejanza con java, c++ o c, siendo más fácil su comprensión y puesta en funcionamiento.

En cuanto a los apartados teóricos como son los casos de uso, diagramas de secuencia, diagramas

uml, o diseño de la base de datos, e tenido que hacer memoria y recuperar los apuntes de las no tan lejanas asignaturas de ingeniería del software (ISS) o Bases de datos (BDA).

También he aprendido a como estructurar un proyecto y dividirlo en partes y cumplir ciertas fechas, a pesar de las demoras de mi proyecto, soy consciente de la importancia que tiene el tiempo en cuanto a la entrega de una pagina web en una empresa.

7.2. Futuras ampliaciones o mejoras.

Finalizada la web con todas sus opciones siempre es bueno notar que se puede mejorar, añadir o incluso cambiar para así conseguir que sea una web acorde con los tiempos que corren en Internet.

Estos son algunas opciones personales que se podría mejorar en la Web.

Me gustaría crear un ranking de usuarios para qué hubiera una competencia sana y así darle más emoción al juego. En el ranking se tendría en cuenta tanto las películas acertadas como la cantidad de pistas sin comprar y la cantidad de créditos que aún se tienen. Esto no se ha realizado porque a pesar de ser bastante sencillo en principio el tiempo se echaba encima y no había una cantidad de películas suficiente como para poder diferenciar a los mejores jugadores.

Cada vez que se acertara una película, que apareciera un link a una página de cine con la sinopsis de la película y o un enlace para comprarla.

También se puede realizar en un futuro una minitienda con descuentos en comercios y que cada descuento se puede adquirir con un número variable de respuestas acertadas. El marcador actual contempla la cantidad de descuentos que se poseen pero no sea implementado la tienda por falta de recursos y comercios.

En un principio se había diseñado una imagen de una televisión para ponerla encima de la imagen pero al utilizar esta capa se perdía la propiedad de pulsar en la imagen para pasar a la siguiente imagen, y se decidió por eliminar este diseño.

Siempre se pueden hacer mejoras en el diseño para qué la página parezca más atractiva, pero acepto mis limitaciones como diseñador.

El apartado de contacto se podría haber implementado con la clase PHPMailer pero ha faltado tiempo y con la clase mail(), los correos iban directamente a la bandeja de correo no deseado. Por eso se ha optado por no usar esta clase y esperar a la implementación de la clase PHPMailer.

8. Bibliografía

8.1. Libros consultados.

- Una guía para la realización y supervisión de proyectos web (PFC)
Buendía García, Félix
Editorial UPV
- CSS, hojas de estilo en cascada para el diseño Web
Christopher Schmitt
Editorial Anaya Multimedia
- Creación de sitios web con PHP5

8.2. Páginas Web visitadas

PHP:

<http://www.manualdephp.com/>

<http://www.php-es.com/>

<http://php.net/>

HTML:

<http://www.desarrolloweb.com/htm>

<http://www.webestilo.com/html/>

CSS:

<http://www.desarrolloweb.com/css>

MYSQL

<http://dev.mysql.com>

www.mysql.com/products/workbench/

AJAX:

www.librosweb.es/AJAX

Otros

<http://jalbum.net/>

<http://www.webtaller.com/tallerscripts/scripts/34/>

Asignatura CDS

http://personales.upv.es/jagil/CDH/Transpas/tema1_html_archivos/frame.htm

http://personales.upv.es/jagil/CDH/Transpas/tema2_css_archivos/frame.htm

http://personales.upv.es/jagil/CDH/Transpas/tema3_javascript_archivos/frame.htm

9. Anexos

9.1. Anexo A. diseño lógico de la base de datos

```
-- Estructura de tabla para la tabla `accesos`  
--
```

```
CREATE TABLE IF NOT EXISTS `accesos` (  
  `usuario` int(11) NOT NULL,  
  `dia` tinyint(4) NOT NULL,  
  `mes` tinyint(4) NOT NULL,  
  `anyo` smallint(4) NOT NULL,  
  `dia_registro` tinyint(4) NOT NULL,  
  `mes_registro` tinyint(4) NOT NULL,  
  `anyo_registro` smallint(4) NOT NULL,  
  PRIMARY KEY (`usuario`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
-- Estructura de tabla para la tabla `ciudades`  
--
```

```
CREATE TABLE IF NOT EXISTS `ciudades` (  
  `cod_ciudad` tinyint(3) unsigned NOT NULL AUTO_INCREMENT,  
  `ciudad` varchar(20) COLLATE utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`cod_ciudad`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=4 ;
```

```
-- Estructura de tabla para la tabla `peliculas`  
--
```

```
CREATE TABLE IF NOT EXISTS `peliculas` (  
  `codigo` mediumint(8) unsigned NOT NULL,  
  `texto` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,  
  `audio` mediumint(8) unsigned NOT NULL,  
  `video` mediumint(8) unsigned NOT NULL,  
  `nombre` varchar(40) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,  
  `tipo` tinyint(4) NOT NULL,  
  `nivel` tinyint(4) NOT NULL COMMENT 'el nivel al que corresponde la pelicula',  
  PRIMARY KEY (`codigo`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf16 COLLATE=utf16_spanish_ci;
```

```
-- Estructura de tabla para la tabla `tipo`  
--
```

```
CREATE TABLE IF NOT EXISTS `tipo` (  
  `cod_tipo` tinyint(3) unsigned NOT NULL,  
  `tipo` varchar(20) COLLATE utf16_spanish_ci NOT NULL,  
  PRIMARY KEY (`cod_tipo`),  
  UNIQUE KEY `tipo` (`tipo`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf16 COLLATE=utf16_spanish_ci;
```

```
-- Estructura de tabla para la tabla `usuario`  
--
```

```
CREATE TABLE IF NOT EXISTS `usuario` (  
  `habilitado` tinyint(1) NOT NULL COMMENT 'para banear si hace algo malo',  
  `codigo` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(15) COLLATE utf8_unicode_ci NOT NULL,  
  `apellidos` varchar(20) COLLATE utf8_unicode_ci NOT NULL,  
  `creditos` smallint(5) unsigned NOT NULL,  
  `correo` varchar(50) COLLATE utf8_unicode_ci NOT NULL,  
  `contrasena` varchar(20) COLLATE utf8_unicode_ci NOT NULL,  
  `ciudad` tinyint(4) NOT NULL,  
  `acertadas` smallint(3) unsigned NOT NULL,  
  `creditoPistas` smallint(3) unsigned NOT NULL,  
  `nivel` tinyint(3) unsigned NOT NULL,  
  PRIMARY KEY (`correo`),  
  KEY `codigo` (`codigo`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=64 ;
```

```
-- Estructura de tabla para la tabla `usuariopeli`  
--
```

```
CREATE TABLE IF NOT EXISTS `usuariopeli` (  
  `usuario` int(10) unsigned NOT NULL,  
  `pelicula` smallint(5) unsigned NOT NULL,  
  `acertada` tinyint(1) NOT NULL,  
  `pistaEscrita` tinyint(1) NOT NULL,  
  `pistaAudio` tinyint(1) NOT NULL,  
  `pistaVideo` tinyint(1) NOT NULL,  
  PRIMARY KEY (`usuario`,`pelicula`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf16 COLLATE=utf16_spanish_ci COMMENT='Relacion  
de los usuarios con cada pelicula';
```