

HISTORIAS DE TERROR

JOSÉ MIGUEL MARTÍN HIDALGO

C.F.GS. DESARROLLO DE APLICACIONES WEB

PROYECTO FINAL

IES CAMPANILAS

MÁLAGA

CURSO 17/18

Cuenta administrador:

Usuario: admin

Contraseña: admin123@

1. Manual del usuario

1.1 Descripción de la aplicación

Se trata de una página web en la que los usuarios que se registren pueden escribir historias de terror y que cualquier persona que entre en la página puede leerlas, cada historia tiene su categoría (puede ser de fantasmas, de rituales, de pesadillas...).

Los usuarios que se registrados pueden puntuar de 0 a 10 estas historias, y se mostrará la puntuación media.

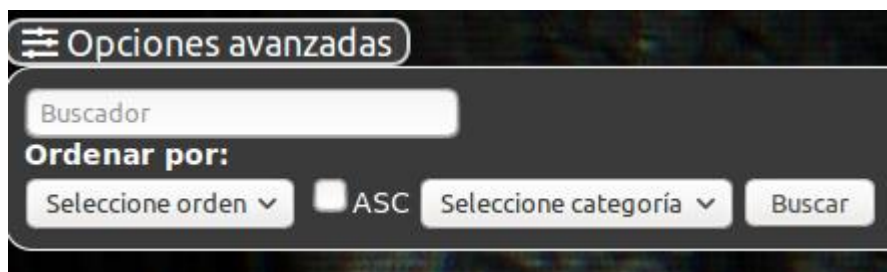
1.2 Funcionamiento

Nada más abrir la página, se mostrará la lista de todas las historias escritas por los usuarios:



Conforme vas bajando por la página, se van mostrando el resto de historias.

Al pulsar el botón de opciones avanzadas puedes buscar, ordenar, y clasificar las historias para que la persona encuentre más fácilmente el tipo de historia que quiere leer:



Al hacer click en “Ver historia completa” se mostrará la historia completa, su puntuación y el usuario que la ha escrito:


Ver historia

← Volver

Quisque ut erat curabitur

Etiam vel augue vestibulum rutrum rutrum neque aenean auctor gravida sem praesent id massa id nisl venenatis lacinia aenean sit amet justo morbi ut odio cras mi pede malesuada in imperdiet et commodo vulputate justo in blandit ultrices enim lorem ipsum dolor sit amet consectetur adipiscing elit proin interdum mauris non ligula pellentesque ultrices phasellus id sapien

- **Categoría:** Juegos
- **Fecha:** 2018-01-26
- **Notamedia:** 7.00
- **Escrito por:** josemiguel

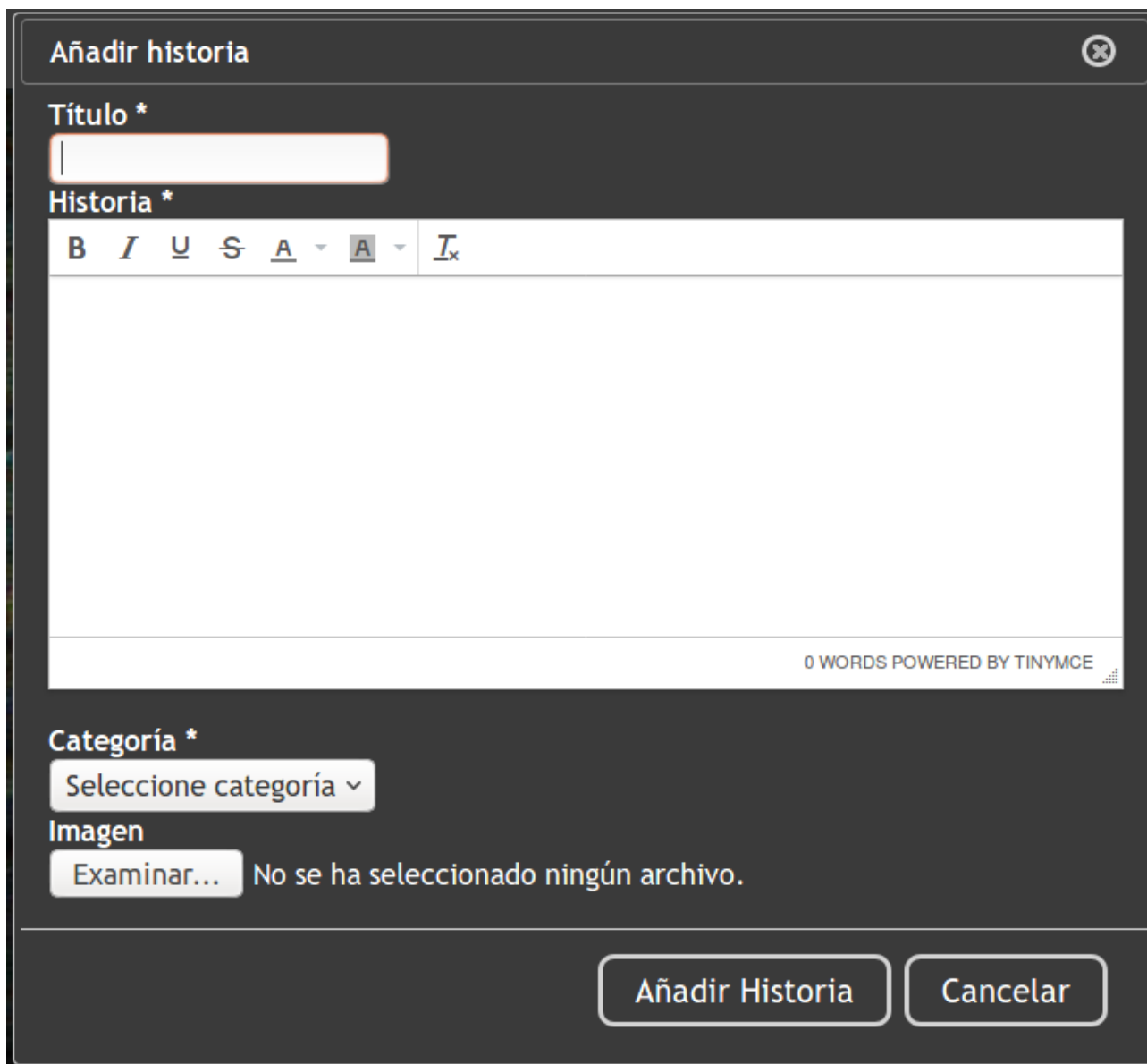


Para escribir tu propia historia, lo primero que hay que hacer es registrarse, al pulsar el botón de registrarse, se mostrará un formulario en el que hay que introducir un nombre de usuario, la contraseña, los nombres y apellidos, el email y opcionalmente una imagen que identifique al usuario.

The image shows a web application interface with a registration modal form. The background is a dark-themed page with a sidebar on the left containing a menu icon and the text "Opciones avanzadas". The main content area displays several text snippets in a serif font, including "Tincfidunt in l...", "Suspendisse potenti in...", "Ver historia completa -->", "In leo maecen...", "Vel lectus in quam fring...", "Ver historia completa -->", "Pede lobortis", and "Ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae quis tauribus accumsan odio c...". At the top right, there are two buttons: "Iniciar sesión" with a right-pointing arrow icon and "Registrarse" with a user icon. The registration modal is a light gray box with a close button (X) in the top right corner. It contains the following fields: "Nombre:" and "Apellidos:" (two text input fields), "Nombre de usuario:" and "E-mail:" (two text input fields), "Avatar:" with an "Examinar..." button and the text "No se ha seleccionado ningún archivo.", "Contraseña:" and "Confirmar contraseña:" (two text input fields). At the bottom of the modal are two buttons: "Registrarse" and "Cancelar".

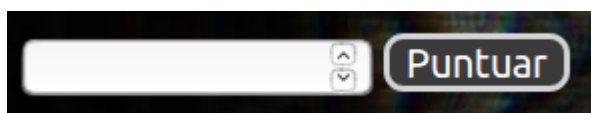
Las contraseñas se almacenarán en la base de datos en cifrado md5

Una vez registrado, se iniciará sesión automáticamente y podrás escribir tu propia historia pulsando el botón “Añadir” de la barra de arriba, aparecerá una ventana en la que tienes que escribir el título, la historia, categoría y opcionalmente asignarle una imagen que identifique a la historia:



Utilizo el editor de texto TinyMCE para poner formato al texto automáticamente.

Si estás logueado, puedes puntuar las historias, al pulsar en “ver historia completa” aparecerá una caja de texto en la que puedes introducir un número de 0 a 10 que será tu puntuación de esa historia:



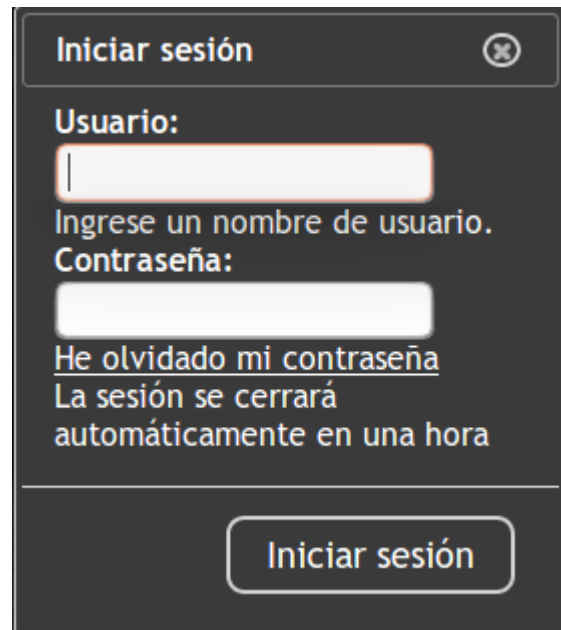
Lógicamente no puedes puntuar tu propia historia.

Si accedes a una historia que has creado tú, aparecerá en la barra de arriba los botones de eliminar y añadir historia:



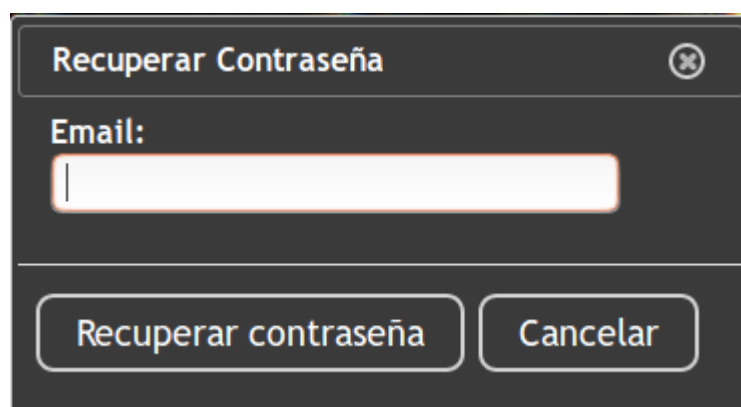
1.2.1 Recuperación de contraseña

Si el usuario ha olvidado la contraseña, puede recibir una nueva pulsando en “He olvidado mi contraseña”.



A dark-themed login modal window titled "Iniciar sesión" with a close button in the top right corner. It contains two input fields: "Usuario:" and "Contraseña:". Below the password field is a link that reads "He olvidado mi contraseña". A message below the link states "La sesión se cerrará automáticamente en una hora". At the bottom is a large button labeled "Iniciar sesión".

Aparecerá un formulario en el que tienes que introducir tu email, tras introducirlo, recibirás un correo con la nueva contraseña de acceso.

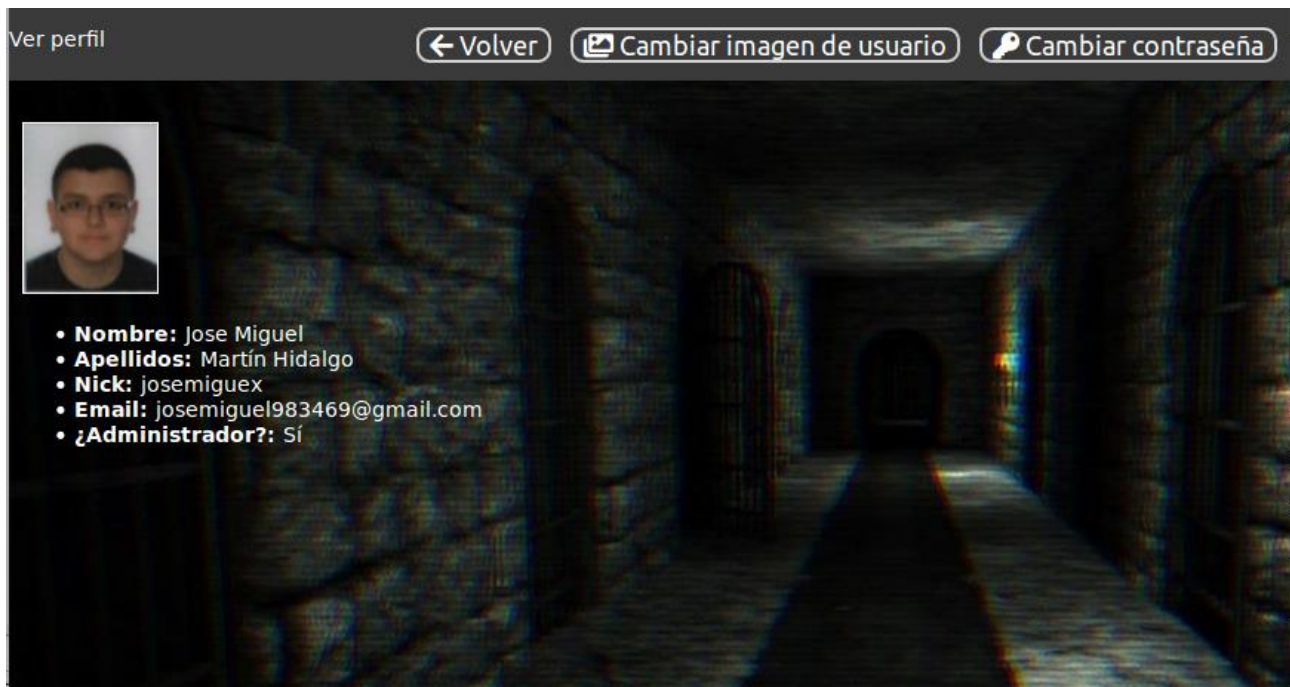


A dark-themed password recovery modal window titled "Recuperar Contraseña" with a close button in the top right corner. It features a single input field labeled "Email:". At the bottom, there are two buttons: "Recuperar contraseña" and "Cancelar".

Para el envío de correo utilizo la librería PHPmailer con el servidor de Gmail.

1.2.2 Gestión del perfil

Puedes acceder a tu perfil pulsando en el botón “ver perfil”, al acceder, verás lo siguiente:



Aquí se ven todos los datos de tu usuario y además puedes cambiar tu contraseña o tu imagen de usuario.

1.2.3 Administración

Si eres un usuario administrador, en el header de la página te aparecerá un botón más que pone “Administración”, si pulsas en ese botón accederás a otra página en la que aparece una tabla con todas las historias publicadas:

Administración [← Volver](#) [+ Añadir historia](#) [Usuarios](#) [Categorías](#) [Cerrar sesión](#)

Opciones avanzadas

Pulse en el título de la historia para leer la historia

ID	Título	Categoría	Usuario	Fecha	Notamedia	Imagen		
50	Dolor vel est donec	Pesadillas	josemiguelx	2018-01-17		Ver	Eliminar	Modificar
49	In leo maecenas pulvinar lobortis	Fantasmas	josemiguelx	2018-02-14		Ver	Eliminar	Modificar
47	Auctor sed tristique	Internet	josemiguelx	2018-01-12		Ver	Eliminar	Modificar
46	Tincidunt ante vel	Juegos	josemiguelx	2018-02-03			Eliminar	Modificar
45	Iaculis congue vivamus metus arcu	Juegos	josemiguelx	2018-01-28			Eliminar	Modificar
44	Tincidunt in leo maecenas pulvinar2	Juegos	josemiguelx	2018-01-02			Eliminar	Modificar
43	Mauris ullamcorper purus sit amet	Juegos	josemiguelx	2018-01-14			Eliminar	Modificar
41	Turpis adipiscing lorem vitae mattis	Juegos	josemiguelx	2018-01-08			Eliminar	Modificar

1 2 3 4 5 >> Ultimo

Puedes añadir historias y eliminar o modificar las historias de todos los usuarios, además si le das a “Mostrar usuarios” se mostrarán una tabla con todos los usuarios, y puedes elegir que usuarios quieres que sean administradores:

Administración [← Volver](#) [Historias](#) [Categorías](#) [Cerrar sesión](#)

ID	Nombre	Apellidos	Nick	Email	Tipo de usuario
1	Jose Miguel	Martin Hidalgo	josemiguelx	josemiguel983469@gmail.com	Ya es administrador
18	0	0	admin		Ya es administrador
23	Jose Carlos	Sanjuan Cardenas	josec	jcsc@gmail.com	Administrador
36	David	Campos Moreno	davidDCM	david@gmail.com	Administrador
40	Pablo	Jimenez Ruiz	newuser	pjr@gmail.com	Administrador

1

También puede gestionar las categorías que puede tener una historia:

Administración [← Volver](#) [+ Añadir categoría](#) [Historias](#) [Usuarios](#) [Cerrar sesión](#)

ID	Nombre	Acciones	
1	Rituales	Eliminar	Modificar
2	Juegos	Eliminar	Modificar
3	Noche	Eliminar	Modificar
4	Muerte	Eliminar	Modificar
5	Fantasmas	Eliminar	Modificar
6	Pesadillas	Eliminar	Modificar
7	Internet	Eliminar	Modificar

2. Manual técnico

2.1 Modelo de datos

La base de datos del proyecto se ha basado en la extensión PDO porque es más sencillo para hacer consultas.

El sistema consta de cuatro tablas: **usuarios**, **historiasdeterror**, **categorías** y **puntuaciones**.

Usuarios:

Contiene la lista de los usuarios que se han registrado en la página.

id	nombre	apellidos	email	usuario	contraseña	avatar	admin
1	Jose Miguel	Martín Hidalgo	josemiguel983469@gmail.com	josemiguex	fpdaw123@	M19JY5MLVZ.jpg	1
18	0	0		admin	~;lj}*DEZCThD}S9U"XF		1
20	sdada	asdsa	sadsa@gmail.com	asasd	asda		0
21	dsfds	sdfsdf	sdfs@sdsaad.com	sadad	asdasd		0
22	sdfs	sdfs	sdfs@sds.com	sds	sd		0
23	Jose Carlos	Sanjuan Cardenas	jcsc@gmail.com	josec	josec		0
24	asdas	asdsadas	asdas@asasdas.com	asdsad	sdadsa		0
25	asdsa	asdsad	asdas@ddsdf.com	sdfs	sdfsdf		0
26	sdfs	sdfsdf	sdfs@dsfs.com	asdas	asdasd		0
28	dgdgdsfds	dfgdf	dfgdf@dfds.com	asdasdf	asdsa		0
30	sdfs	sfsd	fdfs123d@fd.com	sdfsdsfd	sdfs		0

Esta tabla tiene las siguientes columnas:

- ⑩ **Id:** Clave primaria que es un número que identifica al usuario
- ⑩ **Nombre:** Nombre de la persona
- ⑩ **Apellidos:** Apellidos de la persona
- ⑩ **Email:** Email del usuario
- ⑩ **Usuario y contraseña:** Los datos necesarios para iniciar sesión
- ⑩ **Avatar:** Nombre de la imagen del usuario
- ⑩ **Admin:** Indica si un usuario es un admin o no

Historias de terror:

Contiene la lista de las historias escritas por los usuarios.

id	Título	Historia	usuario_id	categoria_id	fecha	img
9	Faucibus orci luctus et eultrices	Pharetra magna ac consequat metus sapien ut nunc v...	1	2	2018-02-09	
10	erat id mauris	nisi nam ultrices libero non mattis pulvinar nulla...	1	2	2018-02-08	
13	pede lobortis ligula	ipsum primis in faucibus orci luctus et ultrices p...	1	2	2018-01-16	
14	lobortis vel dapibus at diam	donec diam neque vestibulum eget vulputate ut ultr...	1	2	2018-01-13	
15	semper sapien a	tellus nisi eu orci mauris lacinia sapien quis lib...	1	2	2018-01-02	
16	quis tortor id nulla ultrices	cubilia curae mauris viverra diam vitae quam suspe...	1	2	2018-01-28	
17	tempus vivamus in	tortor sollicitudin mi sit amet lobortis sapien sa...	1	2	2018-01-04	
18	ligula sit amet eleifend pede	ultrices eu nibh quisque id justo sit amet sapien...	1	2	2018-01-06	
19	at ipsum ac	feugiat non pretium quis lectus suspendisse potent...	1	2	2018-01-24	
20	Ante vestibulum ante ipsum primis 34	Porta volutpat erat quisque erat eros viverra eget...	1	2	2018-02-06	
21	semper est quam pharetra magna	auctor gravida sem praesent id massa id nisl venen...	1	2	2018-02-05	
22	sed magna at	nam nulla integer pede justo lacinia eget tincidunt...	1	2	2018-02-18	

Esta tabla tiene las siguientes columnas:

- ⑩ **Id:** Clave primaria que es un número que identifica a la historia
- ⑩ **Título:** Título de la historia
- ⑩ **Historia:** Contenido de la historia
- ⑩ **Usuario_id:** Id del usuario que ha escrito la historia
- ⑩ **Categoria_id:** Id de la categoría a la que pertenece la historia
- ⑩ **Fecha:** Fecha en la que se ha escrito la historia
- ⑩ **Img:** Nombre de la imagen de la historia (si se ha elegido una a la hora de escribir la historia)

Categorías:

Contiene la lista de categorías que puede tener una historia.

id	nombre
1	Rituales
2	Juegos
3	Noche
4	Muerte
5	Fantasmas

Tiene las siguientes categorías:

- ⑩ **Id**: Clave primaria que es un número que identifica a la categoría
- ⑩ **Nombre**: Nombre de la categoría

Puntuaciones:

Contiene las puntuaciones de los usuarios que tiene cada historia.

id	historia_id	usuario_id	puntuacion
1	79	36	4
45	31	1	7
62	15	1	3
77	79	1	4
79	86	1	9
80	24	1	1

Tiene las siguientes columnas:

- ⑩ **Id**: Clave primaria que es un número que identifica a la puntuación
- ⑩ **Historia_id**: Id de la historia
- ⑩ **Usuario_id**: Usuario que ha puntuado esa historia
- ⑩ **Puntuación**

Al acceder a una historia, se calculará automáticamente la nota media.

Relaciones: La relación de dependencia entre las tablas historiasdeterror y puntuaciones es de 1 a N ya que cada historia puede tener una o muchas puntuaciones y de entre las tablas categorías y historiasdeterror es de 1 a 1 ya que cada historia sólo puede pertenecer a una categoría.

2.2 Funcionamiento

2.2.1 Lista de historias

Al acceder a la página, lo primero que se hace es un **SELECT** de las 5 primeras historias, también se comprueba si hemos introducido en el buscador algún termino o hemos seleccionado alguna categoría, luego los datos obtenidos se pasan a un array y se hace un **for each** sobre ese array.

```

//Según los parámetros que reciba se cambia la sentencia que se ejecutará
if (isset($_POST['busquedahistoria']) && $_POST['busquedahistoria'] != "" || isset($_POST['categoria']) && $_POST['categoria'] != "") {
    $termino = $_POST['busquedahistoria'];
    $historias .= " WHERE";
}

if (isset($_POST['busquedahistoria']) && $_POST['busquedahistoria'] != "") {
    $historias .= " 'Titulo' LIKE '%" . $_POST['busquedahistoria'] . "%' ";
}

if (isset($_POST['busquedahistoria']) && $_POST['busquedahistoria'] != "" && isset($_POST['categoria']) && $_POST['categoria'] != "") {
    $historias .= " AND";
}

if (isset($_POST['categoria']) && $_POST['categoria'] != "") {
    $historias .= " categoria_id='" . $_POST['categoria'] . "' ";
}

if (isset($_POST['ordenapor']) && $_POST['ordenapor'] != "") {
    $historias .= " ORDER BY " . $_POST['ordenapor'];

    if (isset($_POST['asc'])) {
        $historias .= " ASC ";
    } else {
        $historias .= " DESC ";
    }
}

$limit = " limit 0, " . $rowperpage;
$resultado2 = $lnk->query($historias.$limit);
$fla = $resultado2->fetchAll(PDO::FETCH_ASSOC);

```

```

foreach ($fla as $historia) {
    echo "<div class='historia'>";
    echo "<h2 class='titulo' id='" . $historia['id'] . "' onclick='mostrar(\"historia-" . $historia['id'] . "\")'>" . ucfirst($historia['Titulo']) . "</h2> ";

    // Si la historia tiene una imagen, se mostrará
    if ($historia['img'] != "") {
        echo "<img src='subidas/img_historias/" . $historia['img'] . "' style='height:256px'></img>";
    }
    //echo "<div class='historia' id='historia-" . $historia['id'] . "' data-idhistoria='" . $historia['id'] . "' style='display:none'> ";

    if (strlen($historia['Historia']) > 100) {
        $puntos = "..."; // Se muestra puntos suspensivos en la historia si tiene más de 100 caracteres
    } else {
        $puntos = "";
    }

    echo "<p>" . substr(str_replace("\n", "<br>", ucfirst($historia['Historia'])), 0, 100) . $puntos . "</p> ";
    // Para acceder a la historia completa
    echo "<a class='enlacehistoria' href='historia?id=" . $historia['id'] . "'>Ver historia completa --></a><br>";
    echo "</div>";
}
echo "</div>";

```

Para el mostrar el resto de las historias, mediante jQuery se detecta si estamos en la parte de abajo de la pantalla, si es así, y además el número de historias mostradas es menor al número de historias totales, se hace una llamada AJAX a un archivo php (recoger_historias.php) que mediante limit recoge las 3 siguientes historias y ajax co-ge esos datos y los pinta.

Llamada ajax:

```
var d = document.documentElement;
var offset = d.scrollTop + window.innerHeight;
var height = d.offsetHeight;
|
// Si estamos en la parte de abajo de la pantalla
if((height - offset) < 5) {

    var row = Number($('#row').val());
    var allcount = Number($('#all').val());
    var rowperpage = 3;
    row = row + rowperpage;

    // Si el número de historias mostradas es menor al número total de historias
    if(row <= allcount){
        $('#row').val(row);
        $.ajax({
            url: 'recoger_historias.php',
            type: 'post',
            data: {row:row, categoria: categoria,busquedahistoria: termino, ordenapor: ordenapor,asc:asc},
            success: function(response){
                $(".historia:last").after(response).show().fadeIn("slow");
            }
        });
    }
}
}
```

Si hemos hecho realizado búsqueda, ordenado o seleccionado alguna categoría, se enviará también los datos de esa búsqueda, para que se mantenga el orden.

Recoger_historias.php:

1 – Se recogen las 3 siguientes historias:

```
// Recogemos los datos de las historias
$sql = "SELECT notamedia,historiasDeTerror.*,usuarios.usuario, categorias.nombre FROM historiasDeTerror LEFT JOIN categorias ON categorias.id=
historiasDeTerror.categoria id LEFT JOIN usuarios ON usuarios.id=historiasDeTerror.usuario id LEFT JOIN ( SELECT historia_id,AVG(puntuacion) as
notamedia FROM puntuaciones GROUP BY historia_id) as p ON historiasDeTerror.id=p.historia_id";
//Según los parámetros que reciba se cambia la sentencia que se ejecutará
if (isset($_POST['busquedahistoria']) || isset($_POST['categoria'])) {
    $sql .= " WHERE";
}

if (isset($_POST['busquedahistoria'])) {
    $sql .= " 'Título' LIKE '%" . $_POST['busquedahistoria'] . "%' ";
}

if (isset($_POST['busquedahistoria']) && isset($_POST['categoria'])) {
    $sql .= " AND";
}

if (isset($_POST['categoria'])) {
    $sql .= " categoria_id='" . $_POST['categoria'] . "' ";
}

if (isset($_POST['ordenapor'])) {
    $sql .= " ORDER BY " . $_POST['ordenapor'];

    if (isset($_POST['asc'])) {
        $sql .= " ASC ";
    } else {
        $sql .= " DESC ";
    }
}

// Si no se obtiene el ordenapor, por defecto se ordenará por ID
if (!isset($_POST['ordenapor'])) {
    $sql .= " ORDER BY ID DESC";
}

try {
    $limit = " limit " . $_row . "," . $_rowperpage;
    $resultado = $lnk -> query($sql.$limit) ;
}
}
```


2 – Se imprimen mediante un *for each*

```
if (isset($fla['0'])){
foreach ($fla as $historia) {
echo "<div class='historia'>";
echo "<h2 class='titulo' id='". $historia['id']."' onclick='mostrar(\"historia-". $historia['id']."\")'>".ucfirst($historia['Titulo'])."</h2>" ;

// Si la historia tiene una imagen, se mostrará
if ($historia['img'] != ""){
echo "<img src='subidas/img_historias/'. $historia['img']."' style='height:256px'></img>";
}
//echo "<div class='historia' id='historia-". $historia['id']."' data-idhistoria='". $historia['id']."' style='display:none'>" ;

if (strlen($historia['Historia']) > 100) {
    $puntos = "..."; // Se muestra puntos suspensivos en la historia si tiene más de 100 caracteres
} else {
    $puntos = "";
}

echo "<p>".substr(str_replace("\n", "<br>", ucfirst($historia['Historia'])),0,100).$puntos."</p>" ;
// Para acceder a la historia completa
echo "<a class='enlacehistoria' href='historia?id='". $historia['id']."'>Ver historia completa --></a><br>";
echo "<br>";
echo "</div>";
}
}
```

2.2.2 Registro

Al pulsar en el botón de registrarse y rellenar los datos, se comprueba mediante jQuery validate si todos los campos obligatorios están rellenos:

```
// Validación formulario registro
$( '#registerform' ).validate({
    rules: {
        name: { required: true},
        surname: { required: true},
        email: { required: true},
        usr: { required: true},
        pass: { required: true},
        passconfirm: { required: true}
    },
    messages: {
        usr: "Ingrese un nombre de usuario.",
        name: "Ingrese su nombre.",
        surname: "Ingrese sus apellidos.",
        email: "Ingrese un email correcto.",
        pass: "Ingrese una contraseña.",
        passconfirm: "Confirme la contraseña."
    },
    submitHandler: function (form) {
```

El registro se realiza haciendo una llamada al archivo registrar.php, que recibe todos los datos del formulario y realiza el INSERT, pero sólo se realizará el insert si no existe en la base de datos un usuario con el nombre de usuario introducido:

```
//Añadimos usuario a la base de datos (Si no existe un usuario que tenga el mismo nombre o email)
$sql = "INSERT INTO usuarios (`nombre`,`apellidos`,`email`,`usuario`,`contraseña`,`rol_id`,`avatar`) values
('".$_POST['nombre']."','".$_POST['apellidos']."','".$_POST['email']."','".$_POST['usr']."','".$_POST['pass']."','1','".$_filename."')";
$sentencia = $lnk->query($sql);

if (!$sentencia) {
    //Si hay error 1062 (usuario duplicado) no se añade el usuario
    if ($lnk->errorInfo()[1] == 1062) {
        echo false;
    }
    die();
}
}
```

Si el usuario ha insertado una imagen, esa imagen se renombrará a un código alfanumérico generado mediante la siguiente función:

```
function randomcode() {
    $an = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    $su = strlen($an) - 1;
    return substr($an, rand(0, $su), 1) .
        substr($an, rand(0, $su), 1) .
        substr($an, rand(0, $su), 1) .
        substr($an, rand(0, $su), 1) .
        substr($an, rand(0, $su), 1) .
        substr($an, rand(0, $su), 1) .
        substr($an, rand(0, $su), 1) .
        substr($an, rand(0, $su), 1) .
        substr($an, rand(0, $su), 1) .
        substr($an, rand(0, $su), 1);
}
```

Se renombra ese archivo y se sube a una carpeta mediante `move_uploaded_file`:

```
// Obtengo la extensión del archivo
$info = new SplFileInfo(basename($_FILES['avatar']['name']));

// Elegimos el directorio de subida y cambiamos el nombre del archivo
$dir_subida = '../avatares/';
$filename = randomcode()." ".$info->getExtension();
// Subimos el avatar (la imagen del usuario)
$fichero_subido = $dir_subida . $filename;
move_uploaded_file($_FILES['avatar']['tmp_name'], $fichero_subido);
```

Una vez registrado se hace el inicio de sesión, que simplemente se hace un `SELECT` del id del usuario que acabas de crear (que será el id del último usuario registrado más uno) y creo las variables de sesión necesarias.

```
//Hacemos login
$sql = "SELECT * FROM usuarios WHERE usuario='".$$_POST['usr']."' AND contraseña='".$$_POST['pass']."' LIMIT 1" ;
$sentencia = $lnk -> query($sql) ;

$datosusuario = $sentencia->fetchAll();

// Crear las variables de sesión necesarias
$_SESSION["id"] = session_id();
$_SESSION["usr"] = $_POST['usr'];
$_SESSION["usuario_id"] = $datosusuario['0']['id'];
$_SESSION['tiempo'] = time();

//Para que ajax reciba el id del usuario
```

2.2.4 Inicio de sesión

Al rellenar los campos y pulsar en iniciar sesión, se comprueba con jquery autocomplete si se han rellenado todos los campos, en el caso de que sí se hace una llamada mediante AJAX al archivo login.php, antes se comprueba si existe un usuario con dicho nombre de usuario y contraseña, en el caso de que sí se obtiene su id y se crean las variables de sesión necesarias.

```
$sql = "SELECT * FROM usuarios WHERE usuario= BINARY :user AND contraseña=:password" ;
$sentencia = $lnk -> prepare($sql) ;

$sentencia->execute(array(':user' => $_POST['usr'] ,':password'=> $_POST['pass'] )) ;
$resultado = $sentencia->fetchAll();

$data;
//Si he obtenido un resultado correcto
if ($resultado != false) {

    // Crear las variables de sesión necesarias

    $_SESSION["id"] = session_id();
    $_SESSION["usr"] = $_POST['usr'];
    $_SESSION["usuario_id"] = $resultado['0']['id'];
    $_SESSION['tiempo'] = time();
    $_SESSION['admin'] = $resultado['0']['_admin'];

    // Para que ajax reciba el id del usuario
    echo $resultado['0']['id'];
}
```

2.2.5 Funciones de la página (añadir, modificar, eliminar, puntuar historia...)

Todas las funciones que realiza la página relacionadas con las historias (añadir, modificar, eliminar) o de gestión del perfil (cambiar contraseña o imagen de usuario) se hacen también mediante llamadas AJAX a un archivo PHP que recibe los datos que se han enviado y ejecuta la sentencia SQL determinada, por ejemplo, para añadir una historia:

Llamada a AJAX que envía los datos del formulario al archivo php:

```
// Recojo los datos del formulario
var form_data = new FormData();
form_data.append('img', $("#img").prop("files")[0])
form_data.append('titulo', $("#titulo").val())
form_data.append('historia', $("#historia").val())
form_data.append('usuario_id', usuario_id)
form_data.append('fecha', $("#fecha").val())
form_data.append('categoria_id', $("#categoriaañadir").val())

$.ajax({
  url : "operaciones/historia_añadir.php",
  type: "POST",
  data : form_data,
  processData: false,
  contentType: false,
  success: function(data){
    if (data == "duplicated") {
      $("#info").html("La historia introducida ya existe en la base de datos");
      $("#infomodal").css('display', 'block');
    } else {
      //Pinta de nuevo la tabla
      rellenar(data);
      $("#añadirform")[0].reset();
      $("#info").html("Se ha añadido la historia correctamente");
      $("#infomodal").css('display', 'block');
    }
  }
})
```

Dentro del archivo php se reciben los datos y se ejecuta la sentencia SQL:

```
// Si obtengo un archivo
if (isset($_FILES['img']['name'])) {
  // Obtengo la extensión del archivo
  $info = new SplFileInfo(basename($_FILES['img']['name']));

  // Elegimos el directorio de subida y cambiamos el nombre del archivo
  $dir_subida = '../subidas/img_historias/';
  $filename = randomcode().".$info->getExtension();
} else {
  $filename = $_POST['imganterior'];
}

//Añadimos la historia a la base de datos
$sql = "INSERT INTO historiasDeTerror ('Titulo', 'Historia', 'usuario_id', 'categoria_id', 'fecha', 'img') VALUES ('".$_POST['titulo'].", '".$_POST['historia'].", '".$_POST['usuario_id'].", '".$_POST['categoria_id'].", '".$_POST['fecha'].", '".$_POST['img']."'";
echo $sql;
$reg = $lnk -> query($sql) ;

if (!$reg) {
  //Si hay error 1062 (historia duplicada) no se añade la historia
  if ($lnk->errorInfo()[1] == 1062) {
    echo "duplicated";
  }
  die();
}

if (isset($_FILES['img']['name'])) {
  // Subimos la imagen de la historia
  $fichero_subido = $dir_subida . $filename;
  move_uploaded_file($_FILES['img']['tmp_name'], $fichero_subido);
}
```

O para eliminar historias, se envía mediante AJAX a un archivo php el id de la historia y ejecuta un DELETE historiadeterror donde el id de la historia sea el que recibe:

```
// Eliminamos la historia de la base de datos
$sql = "DELETE FROM historiasDeTerror WHERE id=".$_POST["idhistoria"] ;
$reg = $lnk -> query($sql) ;

// Eliminamos sus puntuaciones
$sql = "DELETE FROM puntuaciones WHERE historia_id=".$_POST["idhistoria"] ;
$reg = $lnk -> query($sql) ;
```

También se eliminarán sus puntuaciones.

2.2.3 Envío de email para recuperar contraseña

Los pasos que se siguen tras pulsar el botón de recuperar contraseña son los siguientes:

- 1 – Se hace la llamada ajax a un archivo php enviándole el email
- 2 – Se comprueba si existe en la base de datos un usuario con ese email

```
// Primero miramos si el email introducido existe en la base de datos
$sql = "SELECT usuario FROM usuarios WHERE email='".$_POST['email']."' ;
$resultado = $lnk -> query($sql) ;
$fla = $resultado->fetchAll(PDO::FETCH_ASSOC);
```

3 – Si existe, se genera una nueva contraseña con la misma función que utilizo para renombrar los archivos

```
function randomcode() {
    $an = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    $su = strlen($an) - 1;
    return substr($an, rand(0, $su), 1) .
           substr($an, rand(0, $su), 1) .
           substr($an, rand(0, $su), 1) .
           substr($an, rand(0, $su), 1) .
           substr($an, rand(0, $su), 1) .
           substr($an, rand(0, $su), 1) .
           substr($an, rand(0, $su), 1) .
           substr($an, rand(0, $su), 1) .
           substr($an, rand(0, $su), 1) .
           substr($an, rand(0, $su), 1);
}
```

4 – Cambio la contraseña y envío el email con la nueva contraseña, mediante la utilización de PHPmailer, una librería que permite elegir el servidor de correo y es más completo que la función mail() de PHP:


```
// Cambiamos la contraseña en la base de datos
$sql2 = "UPDATE usuarios SET contraseña='".$newpass.'" WHERE email='".$$_POST['email']."'";
$resultado = $link -> query($sql2);

//Envío del email con la nueva contraseña
$email_user = "josemiguel983469@gmail.com";
$email_password = "Fpdaw123@";
$the_subject = "Recuperación de contraseña";
$address_to = "josemiguel983469@gmail.com";
$from_name = "Historias de terror";
$phpmailer = new PHPMailer();

// ----- datos de la cuenta de Gmail -----
$phpmailer->Username = $email_user;
$phpmailer->Password = $email_password;

// -----
$phpmailer->SMTPDebug = 1;
$phpmailer->CharSet = 'UTF-8';
$phpmailer->SMTPSecure = 'ssl';
$phpmailer->Host = "smtp.gmail.com"; // Gmail
$phpmailer->Port = 465;
$phpmailer->isSMTP(); // use SMTP
$phpmailer->SMTPAuth = true;
$phpmailer->setFrom($phpmailer->Username,$from_name);
$phpmailer->AddAddress($address_to); // recipients email
$phpmailer->Subject = $the_subject;
$phpmailer->Body = "<p>Esta es su nueva contraseña de acceso: ".$newpass."</p><a href='http://".$_SERVER['HTTP_HOST']."/trabajophpajax/historias'>
Haga click aquí para acceder a la página</a>";

$phpmailer->isHTML(true);
$phpmailer->Send();
```

2.3 Tecnologías usadas

JQUERY

Se utiliza para muchas funciones en la página, sobre todo para que cada botón de la página realice su función, como mostrar los jquery dialog, redirigir a otra página, hacer llamadas AJAX, etc.

AJAX

Todas las llamadas que se realizan en esta web se hacen mediante AJAX, como iniciar sesión, registrarse, añadir, eliminar, modificar o puntuar historia.

CCS3

Se utiliza en toda la página sin ningún framework ninguno, excepto que utilicé FontAwesome para los iconos de los botones y se adapta al tamaño de la pantalla mediante los media queries.

REDES

Este proyecto está alojado en un dominio llamado 000webhostapp, se puede acceder a él mediante FTP y las URL son más amigables mediante la modificación del fichero .htaccess

SMTP

Para el envío de correos con una nueva contraseña, utilicé la librería PHPMailer que te permite elegir el servidor de correo, yo he elegido Gmail.