

ACEITES MÁLAGA

blog.aceitesmalaga.com

INICIO

1.1 Bloque central

Saco la información de la última noticia metida a través del panel de administración (que explicaré más adelante).

Para ello lo que hago es utilizar el método `getUltimaNoticia()` de la clase **noticia**, en el que hago la consulta "SELECT idNoticia, idUsuario, titulo, fecha, descripcion, imagen, fechaModificacion, video, visita FROM noticias order by idNoticia desc LIMIT 1" y en el index, lo llamo de la manera que se muestra en la foto.

```
$data['noticias'] = Noticia::getUltimaNoticia();
foreach ($data['noticias'] as $noticia) {
    ?>

    <header>
        <h2><?= $noticia->getTitulo() ?></h2>
        <?php
            if ($noticia->getFechaModificacion() == null){
                ?>
                <span><b>Fecha de creación:</b><?= $noticia->getFecha() ?></span>
            }else{
                ?>
                <span><b>Fecha de creación:</b><?= $noticia->getFecha() ?></span>
                <br>
                <span id="ultimaEdicion">Última edición: <?= $noticia->getFechaModificacion() ?></span>
            }
        <?php
    ?>
</header>
<?php
if ($noticia->getImagen() != null) {
    ?>
        <a href="#" class="image full"></a>
    <?php
}
?>
<p><?= substr($noticia->getDescripcion(), 0, 1000) ?>...</p>
<a href="leerMas/index.php?id=<?= $noticia->getIdNoticia() ?>" class="readMore"><span>Leer más</span></a>
<?php
}
?>
```

Tengo en cuenta que si no existe una imagen en una noticia, que no se ponga, ya que si no se controla esto, sale el icono de una imagen vacía.

Otra cosa que controlo es que en esta pantalla, la descripción sea corta, es decir, uso la función `substr(x,x)`, para mostrar hasta qué carácter quiero mostrar.

```
<p><?= substr($noticia->getDescripcion(), 0, 1000) ?>...</p>
```

1.2 Columna Derecha

En esta columna, lo que hago es mostrar las dos noticias más leídas, para ello lo que hago es almacenar en la BBDD un campo **visita**, este campo se incrementa cada vez que se entra en una noticia.

Para ello, guardo en una variable **\$idNoticia** el ID que me llega por GET y por el método `getNoticiaByIdNoticia($idNoticia)`, obtengo el número de visitas.

Este número de visitas, lo paso junto al ID de la noticia al método `incrementarVisita()`.

```

if (isset($_GET['id'])) { // Si pasamos ID
    $contadorGaleriaImagenes = 0;
    $idNoticia = $_GET['id'];
    $data['noticias'] = Noticia::getNoticiaByIdNoticia($idNoticia);
    $numeroVisita = 0;
    foreach ($data['noticias'] as $noticia) { //Obtiene num visitias de
        //la noticia actual
        $numeroVisita = $noticia->getVisita();
    }
    Noticia::incrementarVisita($numeroVisita, $_GET['id']);
?>

```

El método de **incrementarVisita()** lo que hace es incrementar una visita en el campo visita y actualizarlo en la BBDD.

```

public static function incrementarVisita($numeroVisita, $idNoticia) {
    $conexion = BlogDB::connectDB();
    $numeroVisita++;
    $modificado = "UPDATE noticias SET visita = $numeroVisita WHERE idNoticia = $idNoticia";
    $conexion->exec($modificado);
}

```

1.3 Paginación

La paginación la hago, metiendo en el div con **ID contenido** un **include**. Este div, es el contenedor en el que se encuentra el paginador y llama al **paginadorIndex.php**.

En él que lo que hago es comprobar si existe una página, si existe restamos 1 al número de página y lo multiplicamos por el número de \$RegistrosAMostrar que en este caso son 4 (es un valor que le he asignado yo), para que cada vez empiece por los siguientes 4 registros.

Y sino, que el registro a empezar sea el 0 y la página actual sea la 1.

```

//Sstos valores los recibo por GET
if (isset($_GET['pag'])) {
    $RegistrosAEmpezar = ($_GET['pag'] - 1) * $RegistrosAMostrar;
    $PagAct = $_GET['pag'];
} //caso contrario los iniciamos
else {
    $RegistrosAEmpezar = 0;
    $PagAct = 1;
}

```

Después hago dos consultas, una consulta de todas las noticias y con un fetchObject, saco todos los registros que me interesan y una consulta para obtener el número de registros.

```

$Resultado= $connection->query("SELECT * FROM noticias ORDER BY fecha DESC, idNoticia DESC LIMIT $RegistrosAEmpezar, $RegistrosAMostrar");
echo "<table border='1px'>";
while($MostrarFila=$Resultado->fetchObject()){
    echo "<tr>";
    echo "<td class='date'>".$MostrarFila->fecha."</td>";
    echo "<td class='titulo'>".<a href='leerMas/index.php?id=$MostrarFila->idNoticia'>".$MostrarFila->titulo."</a>.</td>";
    echo "</tr>";
}
echo "</table>";
//*****-----determinar las páginas-----*****
$consultaNumRegistro=$connection->query("SELECT count(*)AS Registros FROM noticias");
while($MostrarFila=$consultaNumRegistro->fetchObject()){
    $NroRegistros= $MostrarFila->Registros;
}

```

Una vez que hago esto, guardo en variables lo que serán los botones de primero, anterior, siguiente o último.

También utilizo la función **floor** para redondear y que no haya decimales.

```
$PagAnt=$PagAct-1;
$PagSig=$PagAct+1;
$PagUlt=$NroRegistros/$RegistrosAMostrar;

//verificamos residuo para ver si llevará decimales
$Res=$NroRegistros%$RegistrosAMostrar;
// si hay residuo usamos funcion floor para que me
// devuelva la parte entera, SIN REDONDEAR, y le sumamos
// una unidad para obtener la ultima pagina
if($Res>0) $PagUlt=floor($PagUlt)+1;

//desplazamiento
echo "<button onclick=\"Página('1')\">Primero</button> ";
if($PagAct>1) echo "<button onclick=\"Página('$PagAnt')\">Anterior</button> ";
echo "<strong>Página ".$PagAct."/". $PagUlt."</strong>";
if($PagAct<$PagUlt) echo "<button onclick=\"Página('$PagSig')\">Siguiente</button> ";

if ($PagUlt==0){
    echo "<button>Último</button>";
} else{
    echo "<button onclick=\"Página('$PagUlt')\">Último </button>";
}

?>
```

Previamente llamamos al archivo **paginadorIndex.js** que es el que le pasa por GET el número de página al archivo **paginadorIndex.php**.

```
function Página(nropagina){
    //donde se mostrarán los registros
    divContenido = document.getElementById('contenido');

    ajax=objetoAjax();
    //uso del metodo GET
    //indicamos el archivo que realizará el proceso de paginar
    //junto con un valor que representa el nro de pagina
    ajax.open("GET", "paginadorIndex.php?pag="+nropagina);

    ajax.onreadystatechange=function() {
        if (ajax.readyState==4) {
            //mostrar resultados en esta capa
            divContenido.innerHTML = ajax.responseText
        }
    }
    //como hacemos uso del metodo GET
    //colocamos null ya que enviamos
    //el valor por la url ?pag=nropagina
    ajax.send(null)
}
```

Una de las opciones que nos da la paginación, es el poder pinchar encima de cualquier noticia y que nos lleve a ella. Esto lo he hecho pasando por el href el ID de la noticia.

```
echo "<td class='titulo'>".<a href='leerMas/index.php?id=$MostrarFila->idNoticia'>".$MostrarFila->título."</a>".</td>";
```

1.4 Pie de página

El pie de página se divide en tres DIV de 33% y el central a su vez se divide en otro tres, están como float left y como display block con un tamaño de 33% también.

1.5 Botones de leer más..

Es similar a lo que hago con el href del título de las noticias, pero con el botón de leer más.

```
<a href="leerMas/index.php?id=<?= $noticia->getIdNoticia() ?>" class="readMore"><span>Leer más</span></a>
```

LEER MÁS

Lo primero que hace la página en leer más es el contador de visitas que anteriormente he explicado.

2.1 Datos de la noticia

2.1.1 El título

Después lo que se ve, es el título de la noticia que hemos elegido, ya sea mediante leer más o bien, pinchando en el título de la noticia en el paginador y junto con el título, todos los datos de la noticia, que los obtengo con el método getNoticiaByIdNoticia(\$idNoticia);

```
$data['noticias'] = Noticia::getNoticiaByIdNoticia($idNoticia);
```

y este método saca todos los datos de la **noticia** con el **\$idNoticia** que se le ha pasado.

2.1.2 El usuario

Luego vemos el usuario que ha publicado la noticia, que lo obtengo con otro método, pero esta vez de la clase **usuario**, pues en la BBDD tengo una tabla **usuarios** relacionada con la tabla **noticias**.

El método que uso para obtener el usuario es getNombreUsuarioByIdUsuario(\$noticia->getIdUsuario()); sin cerrar previamente el método getNoticiaByIdNoticia(\$idNoticia);.

```
$data['usuarios'] = Usuario::getNombreUsuarioByIdUsuario($noticia->getIdUsuario());
foreach ($data['usuarios'] as $usuario) {
    ?
    by <a href="#" id="author"><?= $usuario->getNombre() ?></a>
    ?php
}
```

2.1.3 Fecha de publicación y modificación

Ambas fechas, son fechas que se obtienen con el método getNoticiaByIdNoticia(\$idNoticia); , con la diferencia que tengo controlado que si no existe fecha de modificación ese campo no lo muestre.

```
<?php
if ($noticia->getFechaModificacion() == null) {
    ?
    <span><b>Fecha de creación:</b><?= $noticia->getFecha() ?></span>
    ?php
} else {
    ?
    <span><b>Fecha de creación:</b><?= $noticia->getFecha() ?></span>
    <br>
    <span id="ultimaEdicion">Última edición:<?= $noticia->getFechaModificacion() ?></span>
    ?php
}
?>
```

2.1.4 Botones para compartir

Lo siguiente que podéis ver los botones de compartir en Facebook, twitter y google plus.

Fecha de creación: 25/04/2016

Última edición: 25/04/2016



Estos botones los hago utilizando la librería de **bootstrap-social.css** y la librería de **font-awesome**, la primera me permite tener el estilo de cada botón y con segunda, le pongo los iconos a cada botón.

Otra peculiaridad de estos botones es que te permiten compartir en cada red social, la noticia que estás leyendo, para ello, le pasas el **ID** de la **noticia** a través del href.

Esta parte la hice, buscando el google, como compartir contenido en Facebook mediante un botón en tu página web y como hacer para que aparezca en una página diferente y este código fue el resultado.

(Este es un ejemplo de uno de los botones)

```
<a onclick="javascript:window.open(this.href, '', 'menubar=no,toolbar=no,resizable=yes,scrollbars=yes,height=600,width=600');return false;" href="https://plus.google.com/share?url=http://localhost/Noticias/leerMas/index.php?<?= $idNoticia ?>" class="btn btn-social-icon btn-google"><span class="fa fa-google-plus"></span></a>
```

2.1.5 Galería

Más abajo está la galería dentro de un div con **ID slider**, en el que obtengo las fotos con el método **Imagen::getNombreImagenByIdNoticia(\$noticia->getIdNoticia())**.

Una vez obtengo las imágenes, compruebo que haya imágenes para que no se quede un cuadro vacío en la página (como anteriormente hacia con la imagen de la noticia en la página principal).

Después obtengo la imagen que se almacena en noticia que es la imagen principal y para terminar, obtengo las imágenes de la clase **imágenes** y compruebo que no se repita la imagen principal al mostrar la galería, ya que en la BBDD en la tabla imágenes se almacenan todas las imágenes y hay que comprobar que no se repitan dos veces la misma imagen.

```

<div id="slider" >
    <ul>
        <?php
            $data['imagenes'] = Imagen::getNombreImagenByIdNoticia($noticia->getIdNoticia());
            if ($noticia->getImagen() != null) {
                //if ($contadorGaleriaImagenes == 0) {
                ?>
                    <li></li>
                <?php
                //}
            }
            foreach ($data['imagenes'] as $imagen) {
                if ($imagen->getNombre() != $noticia->getImagen()) {
                    ?>
                    <li></li>
                    <?php
                }
            }
        }
    ?>
</ul>

</div>

```

Después está el script para la animación de la galería.

```

<script type="text/javascript">
    $(document).ready(function () {
        var sudoSlider = $("#slider").sudoSlider({
            numeric: true,
            beforeAnimation: function (slide) {
                $('#div.descrip-text #anistate').text('Animating to slide ' + slide).show(600);
            },
            afterAnimation: function (slide) {
                $('#div.descrip-text #anistate').hide(400);
                $('#div.descrip-text #slidenumber').text(slide);
                var text = $(this).children().attr('src');
                $('#div.descrip-text #slidehtml').text(text);
            },
            initCallback: function () {
                var slide = this.getValue("currentSlide");
                $('#div.descrip-text #slidenumber').text(slide);

                var text = this.getSlide(slide).children().attr('src');
                $('#div.descrip-text #slidehtml').text(text);
            }
        });
    });
</script>

```

Esta galería la he usado con la librería **sudoSlider.min.js**, es una librería que te permite hacer galerías. En esta en concreto tube que cambiar el ancho de las imágenes al 100% en el **CSS** que trae hecho para esta galería para que actuara de forma responsive.

```
<script src="js/jquery.sudoSlider.min.js" type="text/javascript"></script>
```

La galería funciona correctamente con una o varias fotos.

2.1.6 Descripción

Por último como datos de la noticia, tenemos la descripción entera de la noticia, que la obtenemos con el método que anteriormente utilizábamos para el resto de los datos de la noticia, `getNoticiaByIdNoticia($idNoticia);`

```

$data['noticias'] = Noticia::getNoticiaByIdNoticia($idNoticia);
foreach ($data['noticias'] as $noticia) {

```

```
<!-- Post Content -->
<p class="lead"><?= $noticia->getDescripcion() ?></p>
```

2.2 Los comentarios

Para la parte de comentarios, primeramente hago un formulario para recoger los comentarios de los usuarios.

```
<div class="well">
<h4>Deje un comentario:</h4>
<form role="form" action="../Controller/insertarComentario.php" method="POST" id="formularioComentario">
  <div class="form-group">
    <textarea class="form-control" rows="3" name="comentario"></textarea>
  </div>
  <input type="hidden" value="<?= $idNoticia ?>" name="idNoticia">
  <a href="#" id="login" class="btn btn-block btn-social btn-facebook">
    <span class="fa fa-facebook"></span>
    Iniciar sesión
  </a>
  <button type="submit" class="btn btn-primary" id="botonEnviar">Enviar</button>
</form>
</div>
```

Después muestro todo los comentarios, pero tengo en cuenta una cosa, para realizar un comentario es necesario estar logueado en Facebook y una vez logueado podrás escribir comentarios y editar y borrar tus comentarios.

Sin loguearte, no te da la posibilidad de enviar un comentario (al iniciar sesión, el botón de iniciar sesión desaparece y aparece el botón de enviar).



Una vez logueado, , en la esquina superior derecha aparece la imagen de perfil de Facebook que el usuario logueado tenga puesta.

[Volver](#) [Nosotros](#) [Tienda](#)



Con el logueo, guardo el **ID del usuario de Facebook**, así como el **email** y **nombre**.

Utilizo el ID para controlar que ese usuario edite y borre solo los comentarios con ese **ID**. Los datos los obtengo con la API de Facebook y para guardarlos, los paso por campos ocultos al formulario mediante jQuery para que cuando se pinche en enviar, a través del controlador insertarComentario.php, se almacene en la BBDD.

```
var getFacebookData = function() {
  FB.api('/me', 'GET', {fields: 'first_name, name,id,email'}, function(response) {
    //Añadir al formulario datos (Nombre y email)
    //Se añaden al final del formulario estos tres campos ocultos
    $("#formularioComentario").append("<input type='hidden' name='usuario' value='"+response.name+"'>");
    $("#formularioComentario").append("<input type='hidden' name='idUsuarioFacebook' value='"+response.id+"'>");
    $("#formularioComentario").append("<input type='hidden' name='email' value='"+response.email+"'>");
```

Una vez guardado los datos que quiero, un usuario puede escribir comentarios y debajo de la opción de escribir comentarios, los muestro todos.

```
$data['comentarios'] = Comentario::getComentariosByID($idNoticia);
foreach ($data['comentarios'] as $comentario) {
    >
    <div class="media" >?> $comentario->getIdUsuarioFacebook() ?>"> <!-- Aquí empieza cada comentario !-->
    <!-- Si los usuarios escriben dos comentario, le salen dos div con el misma clase (ID de facebook) y son los que puede editar/borrar !-->
    
    <div class="media-body">
        <h4 class="media-heading"><?> $comentario->getUsuario() ?>
        <small><?> $comentario->getFecha() ?></small>
    </h4>
    <p name="<?> $comentario->getIdComentario() ?>"><?> $comentario->getComentario() ?></p>
```

Deje un comentario:

Enviar

 Desarrollador Aceites Málaga 2016-04-27 18:52:25
Esto es una prueba para el proyecto
[Editar](#) [Borrar](#)

 Desarrollador Aceites Málaga 2016-04-27 12:59:19
Es otra prueba
[Editar](#) [Borrar](#)

Si nos fijamos aparece el nombre del usuarios de Facebook, la foto de perfil y la posibilidad de **editar y borrar**. (Ambas posibilidades sólo son posible si los comentarios tienen el mismo ID).

Si pinchamos sobre borrar, el comentario se borra automáticamente (fue un requisito de la empresa) y para ello lo que hago es pasarle al controlador borrarComentario.php mediante **GET el ID del comentario y el ID de la noticia** a la que pertenece y este a su vez usa el método deleteComentario() de la clase **Comentario** para borrar el comentario y volver a la página donde nos encontrábamos.

```
<?php
require_once '../Model/Comentario.php';
$idNoticia = $_GET['idNoticia'];
$noticiaAux = new Comentario($_GET['id'], "", "", "", "", "", "");
$noticiaAux->deleteComentario();
header("Location: ../leerMas/index.php?id=$idNoticia");
```

Si pinchamos en editar, lo primero que hago mediante jQuery es cambiar la palabra **editar** por **cancelar** (de manera que si se pincha en otro editar, el cancelar anterior se pone en editar y el que acabamos de pinchar en editar), después, oculto el formulario que está visible y muestro otro con el contenido del comentario dentro de un texarea, pasando por campo oculto todos los datos del comentario y el usuario, de manera que el usuario rectifica su comentario y yo paso todos los datos junto con el nuevo comentario al contralador modificarComentario.php, de manera que lo modifica y llama al método update() de la clase **Comentario** y vuelve a la página donde nos encontrábamos.

```
require_once '../Model/Comentario.php';
$idNoticia = $_POST['idNoticia'];
$usuarioAux = new Comentario($_POST['idComentario'], $_POST['usuario'], $_POST['fecha'], $_POST['comentario'], $_POST['idNoticia'], $_POST['idUsuarioFacebook'], $_POST['email']);
$usuarioAux->update();
header("Location: ../leerMas/index.php?id=$idNoticia");
```

Si por el contrario, pinchamos en **cancelar**, cancelamos la modificación y el texto vuelve a ponerse como **editar** y se oculta el formulario de editar.

Si queremos desloguearnos de Facebook, pincharemos en la imagen de la esquina superior derecha y nos aparecerá un botón de cerrar sesión, pinchamos sobre él y nos salimos de Facebook.

Par ello uso nuevamente jQuery y la librería de **bootstrap-social.css** y la librería de **font-awesome** para el botón.

```
$('document').on('click', '#logout', function(e) {  
    e.preventDefault();  
  
    if (confirm("¿Está seguro?")){  
        $("#cajaCerrarSesion").hide();  
        $("#botonCerrarSesion").remove();  
        $("#botonEnviar").hide();  
        $('#imagenSesion').remove();  
        facebookLogout();  
    }  
  
    else  
        return false;  
}
```



PANEL DE LOGIN

1. Acceso

Para poder acceder al panel de administrador, es necesario loguearte con un email y contraseña, para ello, recojo ambos datos mediante un formulario.

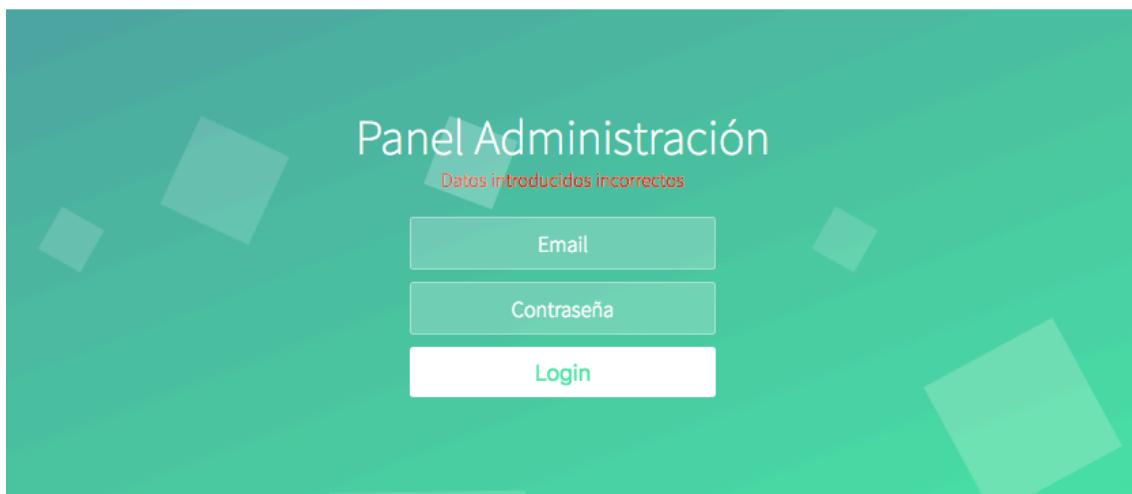
A screenshot of a login form titled "Panel Administración". The form is set against a green gradient background with white text. It contains two input fields: one labeled "Email" and another labeled "Contraseña", both with placeholder text. Below these is a large, prominent "Login" button.

```
<form action="login.php" method="post">
    <input type="text" placeholder="Email" name="email">
    <input type="password" placeholder="Contraseña" name="password">
    <button type="submit" id="login-button">Login</button>
</form>
```

Una vez introducidos los datos, compruebo si se ha pasado el email y si se ha pasado, obtengo por el método `getUsuarioByEmail($_POST['email'])`; de la clase `login.php` el usuario que corresponda con ese email y compruebo que la contraseña que se ha metido por formulario sea la misma que corresponde al usuario de ese email, si es así, pongo la sesión login a true y guardo en una sesión el ID del usuario, el email y el permiso.

```
if(isset($_POST['email'])){
    $data['usuarios'] = Login::getUsuarioByEmail($_POST['email']);
    foreach ($data['usuarios'] as $usuario){
        if(md5($_POST['password']) == $usuario->getPassword()){
            $_SESSION['login']['login'] = true;
            $_SESSION['login']['idUser'] = $usuario->getIdUsuario();
            $_SESSION['login']['email'] = $usuario->getEmail();
            $_SESSION['login']['permiso'] = $usuario->getPermiso();
        }else{
            $_SESSION['login']['login'] = false;
        }
    }
}
```

Si por el contrario la contraseña no es correcta, muestro otro formulario indicando que los datos no son correctos.



En cuanto a los permisos de los usuarios, los pongo porque hay usuarios con permiso de editor y usuarios con permiso de administrador. Los usuarios editores solo pueden editar, modificar o borrar sus propias noticias o comentarios, mientras que los usuarios con permiso de administrador tienen acceso a todo el panel.

2. Panel de administrador

2.1 Noticias

Una vez que accedemos al panel de administrador (que nos redirecciona login.php a panelAdmin.php), tenemos en cuenta lo de los permisos anteriormente mencionados.

```
<!-- Header -->
<div id="header" class="headerAdmin">
<div class="container">
    <?php
    if ($_SESSION['login']['permiso'] == 'administrador') {
        ?>
        <nav id="nav">
            <ul>
                <li><a href="panelAdmin.php">NOTICIAS</a></li>
                <li><a href="nuevoUsuario/nuevoUsuarioAdmin.php">USUARIOS</a></li>
                <li><a href="logout.php">SALIR</a></li>
            </ul>
        </nav>
        <?php
    } else {
        ?>
        <nav id="navPanelAdmin">
            <ul>
                <li><a href="logout.php">SALIR</a></li>
            </ul>
        </nav>
        <?php
    }
}
```

La diferencia visualmente se aprecia tanto en el nav como en las noticias que se muestran. En mi caso al ser administradora, tengo acceso a todo y así es como se vería para cualquier usuario administrador.



Gestión de noticias

		Añadir +	Búsqueda Q		
25/04/2016	Recetas para nuestros aceites aromatizados de La Flor de Málaga...				
18/04/2016	Aceites Málaga en las I Jornadas Gastronómicas Comarcales de la Sierra de las Nieves....				
14/04/2016	Innovaciones en almazara: Aceite al chocolate, trufa o con oro de 22 quilates...				
14/04/2016	Edición Limitada...				

Primeros Página 1/2 Siguiente Último

Para mostrar las noticias, lo que hago es hacer un include de una paginación en Ajax.

```
<div class="container">
    <h2 class="tituloAdministracion">Gestión de noticias</h2>
    <div class="row" id="divPanel">
        <a href="View/formularioInsertarNoticia.php" id="nuevaNoticia">Añadir +</a>
        <a id="busqueda" href="#">Búsqueda <i class="fa fa-search" aria-hidden="true"></i></a>
        <div id="contenedorTabla">
            <?php include('paginadorAdmin.php') ?>
        </div>
    </div>
</div>
```

En **paginadorAdmin.php**, lo que hago es hacer algo similar a lo que en el **paginadorIndex.php**, con la diferencia que aquí incluyo la búsqueda del panel de administrador.

Esta búsqueda se puede hacer por **fecha** o por **título** y en función de si se ha realizado una búsqueda o no, se muestran unas noticias u otras.

Primeramente guardo en variables los datos de fecha o título si existen.

```
//Datos del filtrado
if (isset($_GET["fecha"])) {
    $fecha = $_GET["fecha"];
}

if (isset($_GET["titulo"])) {
    $titulo = $_GET["titulo"];
}
```

Después controlo que exista **permiso** de administrador o editor y en ambos compruebo que exista fecha, título o se muestren sin filtrado.

```
if ($_SESSION['login']['permiso'] == "administrador") {
    if (isset($fecha)) {
        $Resultado = $connection->query("SELECT * FROM noticias WHERE fecha = '$fecha' ORDER BY idNoticia DESC LIMIT $RegistrosAEpezar, $RegistrosAMostrar");
    } elseif (isset ($titulo)){
        $Resultado = $connection->query("SELECT * FROM noticias WHERE titulo LIKE '$titulo%' ORDER BY idNoticia DESC LIMIT $RegistrosAEpezar, $RegistrosAMostrar");
    } else {
        $Resultado = $connection->query("SELECT * FROM noticias ORDER BY idNoticia DESC, idNoticia DESC LIMIT $RegistrosAEpezar, $RegistrosAMostrar");
    }
} elseif (($_SESSION['login']['permiso'] == "editor")) { //Consulta para el filtrado de [Editor]
    if (isset($fecha)) {
        $Resultado = $connection->query("SELECT * FROM noticias WHERE fecha = '$fecha' AND idUsuario = $idUsuario ORDER BY idNoticia DESC LIMIT $RegistrosAEpezar, $RegistrosAMostrar");
    } elseif(isset ($titulo)){
        $Resultado = $connection->query("SELECT * FROM noticias WHERE titulo LIKE '$titulo%' AND idUsuario = $idUsuario ORDER BY idNoticia DESC LIMIT $RegistrosAEpezar, $RegistrosAMostrar");
    } else {
        $Resultado = $connection->query("SELECT * FROM noticias WHERE idUsuario = $idUsuario ORDER BY idNoticia DESC, idNoticia DESC LIMIT $RegistrosAEpezar, $RegistrosAMostrar");
    }
}
```

Una vez comprobado esto, la pantalla principal del panel de administrador, son las **noticias**, por lo que muestro las noticias correspondientes a lo seleccionado (esta parte se hace igual que la paginación del **index.php**).

```
while ($MostrarFila = $Resultado->fetchObject()) {
    >
        <tr id="noticiasPanelAdmin">=> $MostrarFila->idNoticia ?>">
            <td id="fechaAdmin">=> $MostrarFila->fecha ?></td>
            <td id="tituloAdmin">=> substr($MostrarFila->titulo, 0, 100) ?>...</td>
            <td class="iconosFa"><a href="View/FormularioModificarNoticia.php?id=<?= $MostrarFila->idNoticia ?>"><i class="fa fa-pencil"></i></a></td>
            <td class="iconosFa"><i id="borrar" class="fa fa-close" name=<?= $MostrarFila->idNoticia ?>"></i></td>
            <td class="iconosFa"><a href="administrarComentarios.php?id=<?= $MostrarFila->idNoticia ?>"><i class="fa fa-commenting-o" id="iconComents"></i></a></td>
            <td class="iconosFa"><a href="administrarGaleria.php?id=<?= $MostrarFila->idNoticia ?>"><i class="fa fa-picture-o" aria-hidden="true" id="iconImagen"></i></a></td>
        </tr>
    </php>
}
```

Ahora para la hora de la paginación de los botones, lo que hago es, al **paginadorAdmin.js**, para contar las páginas, en la función de búsqueda que es para controlar que ordene los datos, le paso desde **paginadorAdmin.php** la palabra **todo** para que lo muestre todo, la palabra **fecha** para que cuente las páginas de fecha o **titulo** para que muestre el título.

```

function Pagina(nropagina,fecha,busqueda,titulo){
    //busqueda es para controlar que ordene los datos, le pasamos desde
    //paginadorAdmin.php la palabra TODO para que lo muestre todo o mo le
    //escribimos nada para que muestre la fecha.
    //donde se mostrar los registros
    divContenido = document.getElementById('contenedorTabla');

    ajax=objetoAjax();
    //uso del metodo GET
    //indicamos el archivo que realizará el proceso de paginar
    //junto con un valor que representa el nro de pagina
    if (busqueda == "todo"){
        ajax.open("GET", "paginadorAdmin.php?pag="+nropagina);

    }
    if (busqueda == "fecha"){
        ajax.open("GET", "paginadorAdmin.php?pag="+nropagina+"&fecha="+fecha);
    }

    if (busqueda == "titulo"){
        ajax.open("GET", "paginadorAdmin.php?pag="+nropagina+"&titulo="+titulo);
    }
    ajax.onreadystatechange=function() {
        if (ajax.readyState==4) {
            //mostrar resultados en esta capa
            divContenido.innerHTML = ajax.responseText
        }
    }
    //como hacemos uso del metodo GET
    //colocamos null ya que enviamos
    //el valor por la url ?pag=nropagina
    ajax.send(null)
}

```

De esta manera, en el **paginadorAdmin.php**, voy controlando cual es el parámetro que le llega por la función y hago la ordenación.

2.1.1 La búsqueda

El botón de búsqueda funciona en el panel de administrador de la misma forma para el usuario con permiso de administrador que para el usuario con permisos de editor.

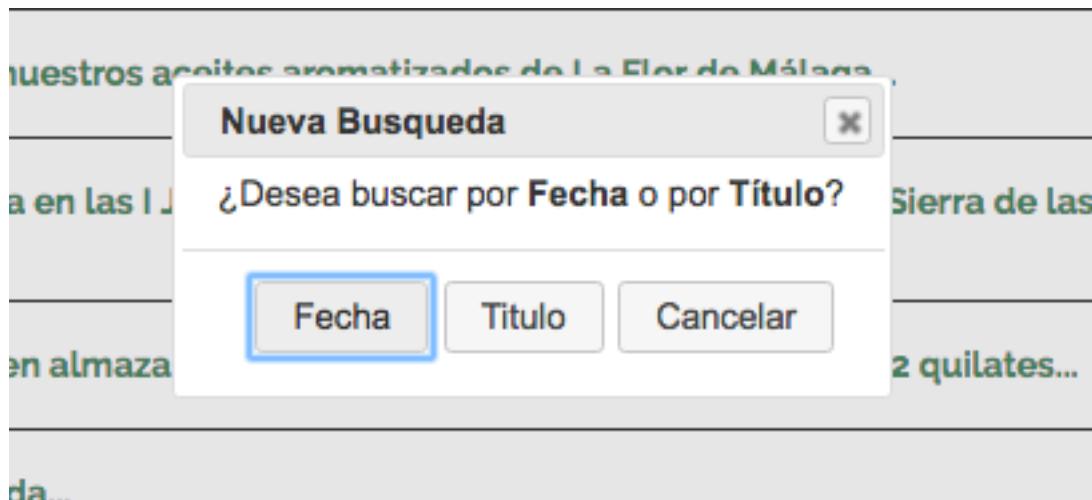
Para realizar la búsqueda, pincho sobre el botón buscar llamado **#btnbuscar** y mediando jQuery, me aparece un diálogo con tres opciones para elegir, **fecha**, **título** o **cancelar**.

```

$( "#btnBuscar" ).dialog({
    autoOpen: false,
    resizable: false,
    modal: true,
    buttons: {
        "Fecha": function () {
            $("#formBuscarFecha").dialog("open");
            $(this).dialog("close");
        },
        "Titulo": function () {
            $("#formBuscarTitulo").dialog("open");
            $(this).dialog("close");
        },
        "Cancelar": function () {
            $(this).dialog("close");
        }
    }
});

$(document).on("click", "#busqueda", function () {
    $("#btnBuscar").dialog("open");
})

```



Si pincho sobre fecha, nos aparece un calendario para elegir la fecha, pero en realidad es un formulario oculto que aparece cuando lo llamo por jQuery.

```

<form id="formBuscarFecha">
    <input id="inputFecha" type="text" name="fecha">
</form>

```

```

$("#formBuscarFecha").dialog({
    autoOpen: false,
    resizable: false,
    modal: true,
    buttons: {
        "enviar": function () {
            fecha = document.getElementById("inputFecha").value;
            console.log(fecha);

            divContenido = document.getElementById('contenedorTabla');
            ajax = objetoAjax();

            ajax.open("GET", "paginadorAdmin.php?fecha=" + fecha);

            ajax.onreadystatechange = function () {
                if (ajax.readyState == 4) {
                    divContenido.innerHTML = ajax.responseText;
                }
            }
            ajax.send(null);
            $(this).dialog("close");
        }
    }
})

```

ESTO ES PARA PRESENTACIÓN POWER POINT

`resizable()` es otra de las funciones básicas de Jquery ui, esta función permite cambiar el tamaño de los elementos amoldando sus dimensiones con el ratón, por ejemplo un cuadro de texto, un textarea ...

Algunos de los métodos que se pueden aplicar a la función `resizable()` son los siguientes ...

- `animate: true | false` -> permite dar un efecto de retardo a las nuevas dimensiones del elemento.

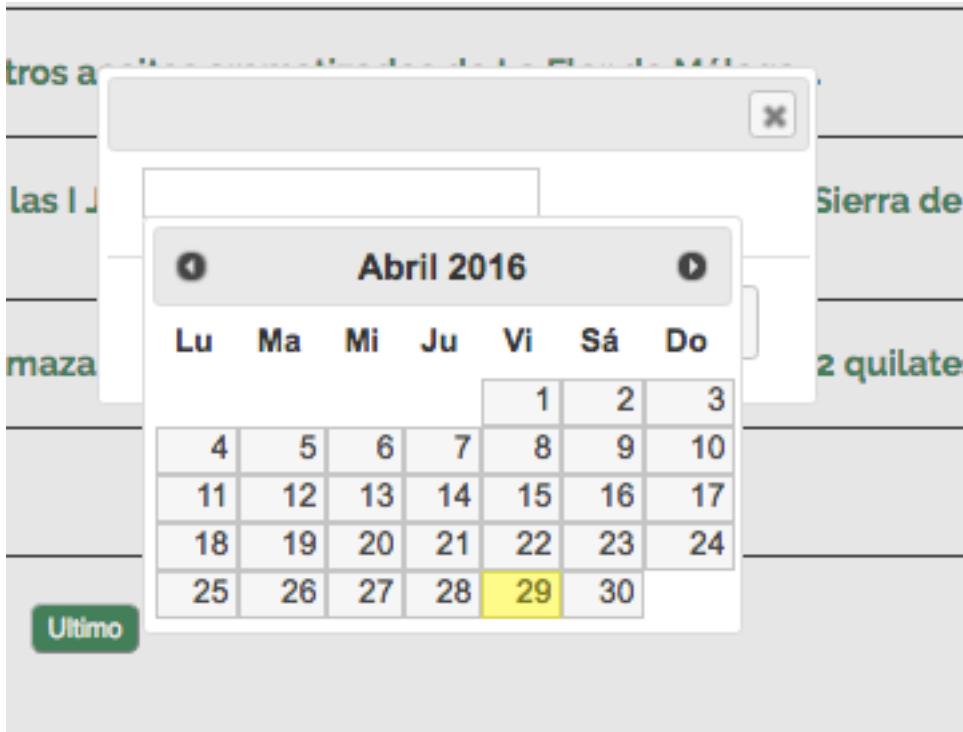
De la manera que se muestra anteriormente en el código, no muestra un calendario, pero para que lo haga, he utilizado la librería `jquery.ui.datepicker-es.js` y un script adecuado que me permite elegir un calendario en español y elegir el día en el que quiero que empiece el calendario.

```

<script src="js/jquery-ui-1.11.4.custom/jquery.ui.datepicker-es.js" type="text/javascript"></script>
<script>
$(function () {
    $.datepicker.setDefaults($.datepicker.regional["es"]);
    $("input[name=fecha]").datepicker({
        firstDay: 1
    });
});
</script>

```

Con esto, la apariencia es la siguiente.



Cuando elegimos la fecha, desde el jQuery se pasa por GET al **paginadorAdmin.php** y de la manera anteriormente explicada, ordena las noticias.

```
ajax.open("GET", "paginadorAdmin.php?fecha=" + fecha);
```

Para buscar por **título**, el proceso es prácticamente igual al seguido en la búsqueda por fecha, la diferencia que en el formulario oculto en lugar de haber un input type text modificado por el script de la librería **jquery.ui.datepicker-es.js**, es un input type text normal que recoge el texto introducido y en la búsqueda que se hace en el **paginadorAdmin.php**, se hace un **select** indicando que aparezcan todos los títulos que empiecen con ese nombre (diferencia entre mayúsculas y minúsculas).

```
} elseif(isset($titulo)){
    $Resultado = $connection->query("SELECT * FROM noticias WHERE titulo LIKE '$titulo%' ORDER BY idNoticia DESC LIMIT $RegistrosAEmpezar, $RegistrosAMostrar");
```

2.1.2 Editar noticias

Visualmente nos encontramos una tabla con las diferentes opciones que tenemos para aplicar a una noticia y aquí me voy a centrar en explicar la edición.



Gestión de noticias	
Añadir +	Búsqueda Q
25/04/2016	Recetas para nuestros aceites aromatizados de La Flor de Málaga...
18/04/2016	Aceites Málaga en las I Jornadas Gastronómicas Comarcales de la Sierra de las Nieves....
14/04/2016	Innovaciones en almazara: Aceite al chocolate, trufa o con oro de 22 quilates...
14/04/2016	Edición Limitada...

Primeros Página 1/2 Siguiente Último

Para editar las noticias, lo hago de la siguiente manera, en el paginadorAdmin.php, en la parte donde se muestran las noticias, hay varios enlaces que mediante la librería **font-awesome.min.css** y **font-awesome.css** le pongo las imágenes y le aplico un color a cada uno.

```
<td class="iconosFa" ><a href="View/formularioModificarNoticia.php?id=<?= $MostrarFila->idNoticia ?>"><i class="fa fa-pencil"></i></a></td>
```

Al pinchar sobre el enlace, nos lleva a un formulario para modificar la noticia, mediante el método **getNoticiaById(\$_GET['id'])** obtengo todos los datos que tiene la noticia y modiflico lo que quiera.

```
$data['noticias'] = Noticia::getNoticiaById($_GET['id']);
foreach ($data['noticias'] as $noticia) {
    ?
    <div id="centro">
        <form action="../Controller/modificarNoticia.php" enctype="multipart/form-data" method="POST" class="formNuevoUsuario">
            <h3>Título</h3>
            <input class="anchoInput" type="text" size="40" name="titulo" value=<?=$noticia->getTitulo() ?>>
            <h3>Descripción:</h3>
            <textarea id="myTextArea" name="descripcion"><?=$noticia->getDescripcion() ?></textarea>
            <h3>Fecha Modificación:</h3>
            <input class="anchoInput" type="date" name="fechaModificacion">
            <h3>Vídeo:</h3>
            <input class="anchoInput" type="text" id="video" name="video" value='<?=$noticia->getVideo() ?>'>
            <input type="hidden" name="fechaPublicacion" value='<?=$noticia->getFecha() ?>'>
            <input type="hidden" name="idNoticia" value='<?=$_GET['id']?>'>
            <input type="hidden" name="idUser" value='<?=$noticia->getIdUsuario() ?>'>
            <input class="aceptar" type="submit" value="Aceptar">
            <a class="cancelar" href="../panelAdmin.php">Cancelar</a>
            <script type="text/javascript">
                CKEDITOR.replace ("editor10");
            </script>
        </form>
    </div>
```

Lo más característico de este formulario es el uso de un editor de textos, que he utilizado para la descripción.

Para implementar, he usado la librería **tinymce.min.js** y un script para modificar el textarea con ID **myTextArea**.

```
<script>
    tinymce.init({
        selector: '#myTextArea',
        plugin: 'a_tinymce_plugin',
        a_plugin_option: true,
        a_configuration_option: 400
    });
</script>
```

La forma visual de este formulario, está hecha en CSS por mí, sin el uso de plantilla.

Título	Recetas para nuestros aceites aromatizados de La Flor de Málaga
Descripción	<p>File ▾ Edit ▾ View ▾ Format ▾</p> <p>Formats ▾ B I</p> <p>Cómo ya sabéis, hace un tiempo que lanzamos al mercado nuestros aceites aromatizados con la marca La Flor de Málaga. Hoy queremos haceros llegar una serie de recetas para que disfrutéis muchísimo más del sabor de estos aceites. Aquí las tenéis: Para nuestro SABOR NARANJA: Ensalada de naranja y anchoas.</p> <p>p</p>
Fecha Modificación	dd/mm/aaaa
Vídeo	<input type="text"/>
<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>	

2.1.3 Borrar noticias



Gestión de noticias

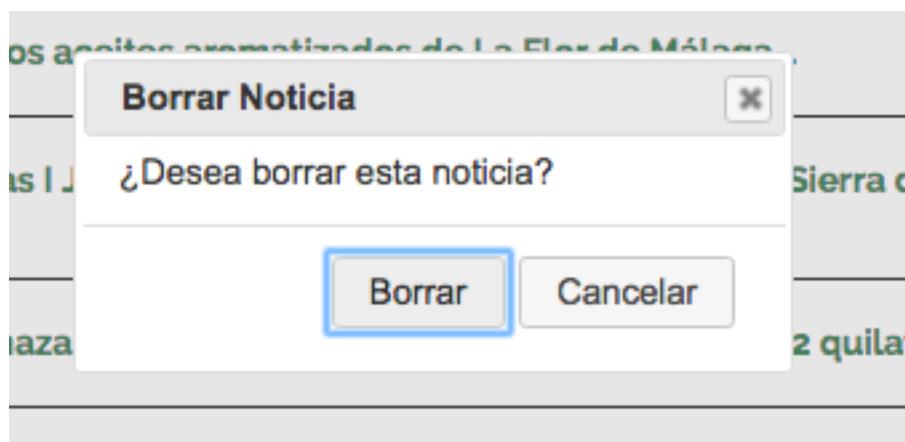
Añadir + Búsqueda Q

Fecha	Título	Acciones
25/04/2016	Recetas para nuestros aceites aromatizados de La Flor de Málaga...	
18/04/2016	Aceites Málaga en las I Jornadas Gastronómicas Comarcales de la Sierra de las Nieves....	
14/04/2016	Innovaciones en almazara: Aceite al chocolate, trufa o con oro de 22 quilates...	
14/04/2016	Edición Limitada...	

Primeros Página 1/2 Siguiente Último



Para borrar las noticias, es otro enlace, que al pinchar sobre él, aparece un cuadro de diálogo indicando si quiero borrar la noticia o cancelar.



Si pincho en Borrar, desde jQuery mando el id de la noticia al controlador **borrarNoticia.php** y borro la noticia, también utilizo un **fadeOut** para simular que desaparece.

```

$(document).ready(function(){
    var idNoticia;
    //VENTANA DIALOGO BORRAR
    $("#dialogoBorrar").dialog({
        autoOpen :false,
        resizable :false,
        modal : true,
        buttons:{ 
            "Borrar": function(){
                $.get("Controller/borrarNoticia.php?id=" + idNoticia,function(){
                    $("#noticiasPanelAdmin" + idNoticia).fadeOut(1000);
                });
                $(this).dialog( "close" );
            },
            "Cancelar": function(){
                $(this).dialog( "close" );
            }
        }
    });
    //BORRAR
    $(document).on("click","#borrar",function(){
        idNoticia = $(this).attr("name");
        $("#dialogoBorrar").dialog("open");
        console.log("El id es: " + idNoticia);
    })
});

```

2.1.4 Ver comentarios



Gestión de noticias

		Añadir +	Búsqueda Q				
25/04/2016	Recetas para nuestros aceites aromatizados de La Flor de Málaga...						
18/04/2016	Aceites Málaga en las I Jornadas Gastronómicas Comarcales de la Sierra de las Nieves....						
14/04/2016	Innovaciones en almazara: Aceite al chocolate, trufa o con oro de 22 quilates...						
14/04/2016	Edición Limitada...						

Primeros Página 1/2 Siguiente Último

Tenemos la opción de ver los comentarios de la noticia en la que pinchamos. (Hay que tener en cuenta que los usuarios administradores verán todas las noticias con sus comentarios correspondientes y los usuarios editores, verán sólo sus noticias con los comentarios de sus noticias.).

Al pinchar sobre el enlace de comentarios, nos lleva a **administrarComentarios.php** y aquí tenemos en cuenta si el usuario es administrador o editor para que se vea en el nav, sólo lo que él tenga acceso. Además hacemos un include al **paginadorAdminComentarios.php**.

```

<?php
if ($_SESSION['login']['permiso'] == 'administrador') {
    ?>
    <nav id="navPanelAdmin">
        <ul>
            <li><a href="panelAdmin.php">NOTICIAS</a></li>
            <li><a href="nuevoUsuario/nuevoUsuarioAdmin.php">USUARIOS</a></li>
        </ul>
    </nav>
    <?php
}
?>
<div id="tituloPanel">
    <h1>PANEL DE ADMINISTRACIÓN</h1>
</div>

</div>
<!-- Header -->

<div id="featured">
    <div class="container">
        <h2 class="tituloAdministracion">Gestión de comentarios</h2>
        <div class="row" id="divPanel">
            <a href="panelAdmin.php" id="nuevaNoticia">Volver</a>
            <div id="contenedorTabla">
                <?php include('paginadorAdminComentario.php') ?>
            </div>
        </div>
    </div>
</div>

```

El **paginadorAdminComentarios.php**, funciona como los paginadores anteriormente explicados.

La forma visual de esta parte es así.

Gestión de comentarios

Gestión de comentarios		
Volver		
2016-04-27 18:52:25	Esto es una prueba para el proyecto...	
2016-04-27 12:59:19	Es otra prueba ...	

[Primero](#) Página 1/1 [Último](#)

Y nos da la opción de borrar, que está hecha de la misma forma que el borrar noticias, pero con comentarios.

2.1.4 Editar imágenes

NOTICIAS USUARIOS SALIR

PANEL DE ADMINISTRACIÓN

Gestión de noticias

	Añadir +	Búsqueda Q			
25/04/2016	Recetas para nuestros aceites aromatizados de La Flor de Málaga...				
18/04/2016	Aceites Málaga en las I Jornadas Gastronómicas Comarcales de la Sierra de las Nieves....				
14/04/2016	Innovaciones en almazara: Aceite al chocolate, trufa o con oro de 22 quilates...				
14/04/2016	Edición Limitada...				

Primeros Página 1/2 Siguiente Último

Si pincho sobre el enlace de editar las imágenes, nos lleva a la página para la edición de imágenes ([administrarGaleria.php](#)), en la que nos aparecen todas las imágenes que tiene la noticia, con opción de añadir, borrar o hacer principal una imagen. Visualmente se ve así.

Gestión de Galería

« Volver	Añadir +	Borrar	Hacer Principal

En cuanto al código, se puede ver, que si se pincha en **volver**, vuelve al **panelAdmin.php**, si se pincha en **añadir**, se redirige a **formularioInsertarImagen.php**, pasándole el ID de la noticia, si se pincha en **borrar**, borra la imagen mediante jQuery y si se pincha en **hacer principal**, sigue unos pasos mediante jQuery,

que explicaré más adelante para hacer principal la imagen.

```
<div id="featured">
<div class="container">
    <h2 class="tituloAdministracion">Gestión de Galería</h2>
    <div class="row" id="divPanel">
        <a href="panelAdmin.php" id="nuevaNoticia"/><i class="fa fa-arrow-circle-o-left" aria-hidden="true"></i> Volver</a>
        <a style="margin-left: 2px;" href="View/formularioInsertarImagen.php?idNoticia=<?= $_GET['id']; ?>" id="nuevaNoticia"/>Añadir&ampnbsp+</a>
        <a style="margin-left: 2px;" id="borrarImagen"/>Borrar <i class="fa fa-trash" aria-hidden="true"></i></a>
        <a style="margin-left: 2px;" id="hacerPrincipal"/>Hacer Principal <i class="fa fa-picture-o" aria-hidden="true"></i></a>
    </div>
    <div id="contenedorTabla">
        <?php
        $data['imagenes'] = Imagen::getImagenesByIdNoticia($_GET['id']);
        foreach ($data['imagenes'] as $imagen) {
            if ($imagen->getNombre() != "") {
                ?
                getIdImagen() ?>" file="<?= $imagen->getNombre() ?>" idNoticia="<?= $imagen->getNoticia() ?>">
            }
        }
        ?
    </div>
    </div>
</div>
```

2.1.4.1 Añadir Imagen

Para añadir una una imagen, lo que hago es mediante un formulario con un campo type file, subir las imágenes.

```
<div id="centro">
    <form action="../Controller/insertarImagen.php" id="formularioInsertarImagen" enctype="multipart/form-data" method="POST">
        <h3>Imagenes galería</h3>
        <input class="anchoInput" type="file" id="imagen" name="imagen[]" multiple="">
        <input type="hidden" name="idNoticia" value="<?= $_GET['idNoticia']; ?>">
        <br>
        <div id="vistaPrevia">
            </div><input class="aceptar" type="submit" value="Aceptar">
            <a class="cancelar" href="../administradorGaleria.php?id=<?= $_GET['idNoticia']; ?>">Cancelar</a>
        </form>
    </div>
```

Imagenes galería

Elegir archivos Ningún archivo seleccionado

Aceptar

Cancelar

Si bien, el campo type file, tiene como name un **Array**, esto es para la subida múltiple de imágenes, que hago mediante el siguiente script.

```
<script>
$(function(){
    //Limpiar vista previa
    $("#imagen").on("change",function(){
        $("#vistaPrevia").html("");
        var archivos = document.getElementById('imagen').files;
        var navegador = window.URL || window.webkitURL;
        //Recorrer los archivos
        for(i = 0; i < archivos.length; i++){
            console.log("holo");
            var type = archivos[i].type;

            if(type !== "image/jpeg" && type !== "image/png" && type !== "image/gif"){
                $("#vistaPrevia").append("El fichero introducido no es del tipo de imagen permitido");
            }else{
                var objetoUrl = navegador.createObjectURL(archivos[i]);
                $("#vistaPrevia").append("<img src='"+objetoUrl+"' width='100'>");
            }
        }
    });
</script>
```

Se puede observar que en el script, controlo que solo se puedan subir unos formatos de imágenes concretos, tales como, **jpeg**, **jpg**, **png** o **gif**. En caso de no ser un formato de estos, aparecerá un mensaje indicando “El fichero introducido no es del tipo de imagen permitido”, por el contrario, si es un archivo de los indicados, aparecerá una vista previa de las imágenes.

Una vez subidas las imágenes, al pinchar en aceptar, se redirige al controlador **insertarImagen.php** en el que se realiza la inserción.

2.1.4.2 Borrar Imagen

Para borrar una imagen, es necesario tenerla seleccionada previamente, para ello, mediante jQuery, controlo si se hace click en una imagen. Cuando se hace click sobre ella, se le añade el atributo **selected**, sino, se le añade el atributo **unselected**.

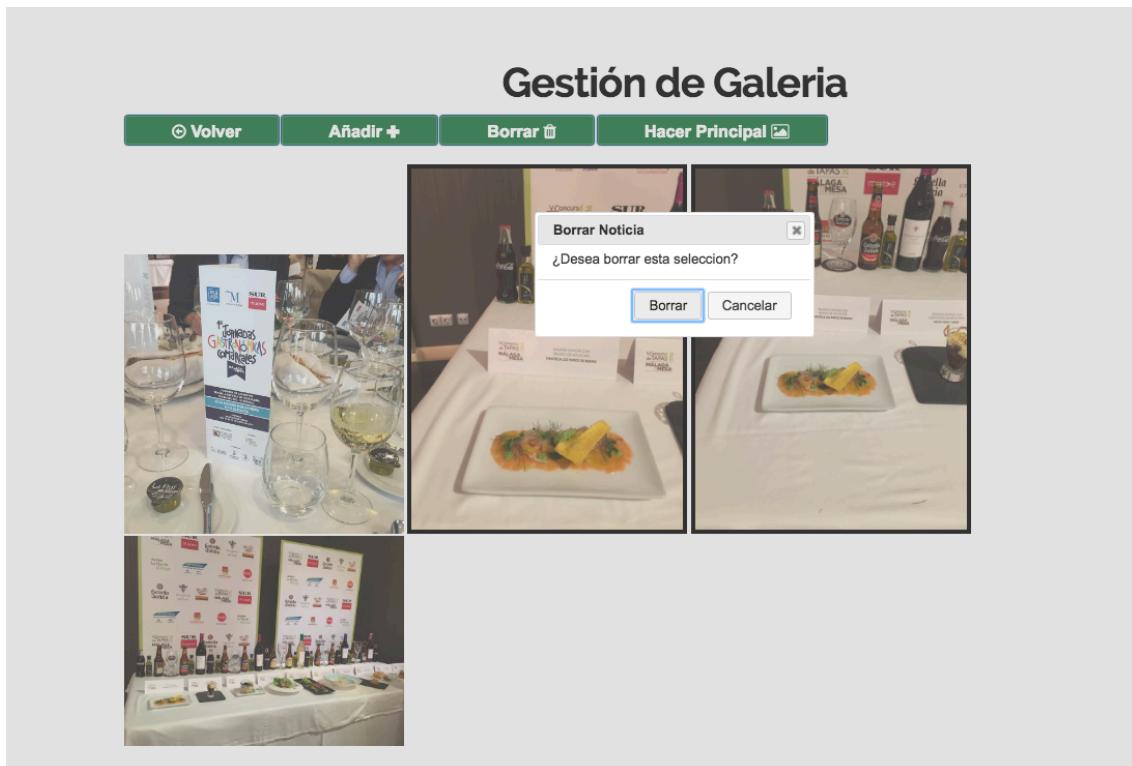
Una vez que se le añade el atributo selected, se le pinta un cuadro negro simulando la selección.

```
$(document).on("click", "#contenedorTabla img", function () {
    if ($(this).attr("name") !== "selected") {
        $(this).attr("name", "selected");
        $(this).css("border", "5px solid black");
    } else {
        $(this).attr("name", "unselected");
        $(this).removeAttr("style");
    }
});
```

Al pinchar en el botón **Borrar Imagen**, recorre las imágenes con el atributo selected y abre el diálogo de borrar, en el que si se pincha en borrar, redirige el ID de la imagen al controlador **borrarImagen.php** y la/s borra. Si se pincha en cancelar, se cancela la acción

```
//BORRAR
$(document).on("click", "#borrarImagen", function () {
    $("#contenedorTabla img[name='" + "selected" + "']").each(function () {
        if(contador == 0){
            idImagen = $(this).attr("value")+",";
            contador++;
        }else{
            idImagen += $(this).attr("value")+",";
        }
    });
    contador = 0;
    $("#dialogoBorrar").dialog("open");
});

$(document).ready(function () {
    var idImagen;
    var contador = 0;
    //VENTANA DIALOGO BORRAR
    $("#dialogoBorrar").dialog({
        autoOpen: false,
        resizable: false,
        modal: true,
        buttons: {
            "Borrar": function () {
                //console.log(idImagen);
                $.get("Controller/borrarImagen.php?idImagen=" + idImagen, function () {
                    $("#contenedorTabla img[name='" + "selected" + "']").fadeOut(1000,function(){
                        $(this).remove();
                    });
                });
            },
            "Cancelar": function () {
                $(this).dialog("close");
            }
        }
    });
});
```



2.1.4.3 Hacer imagen principal

Para hacer una imagen principal, primeramente tienes que hacer click sobre ella y como anteriormente he explicado, se le pondrá el atributo selected, que en este caso aprovecho en el jQuery de hacer la imagen principal.

Al pinchar sobre le botón de **Hacer principal**, recorre las imágenes con el atributo selected y las cuenta, de manera que si hay más de una imagen seleccionada, nos aparecerá un mensaje de error indicando “**Para hacer principal la imagen solo puedes seleccionar una**”, si no hay ninguna seleccionada, aparecerá un mensaje de error indicando “**Debes seleccionar un elemento como mínimo**” y si por el contrario, sólo hay una imagen seleccionada, se abrirá el diálogo de hacer principal.

```
$(document).on("click", "#hacerPrincipal", function () {
    $("#contenedorTabla img[name=" + "selected" + "]").each(function () {
        nombreImagen = $(this).attr("file");
        idNoticia = $(this).attr("idNoticia");
        contador++;

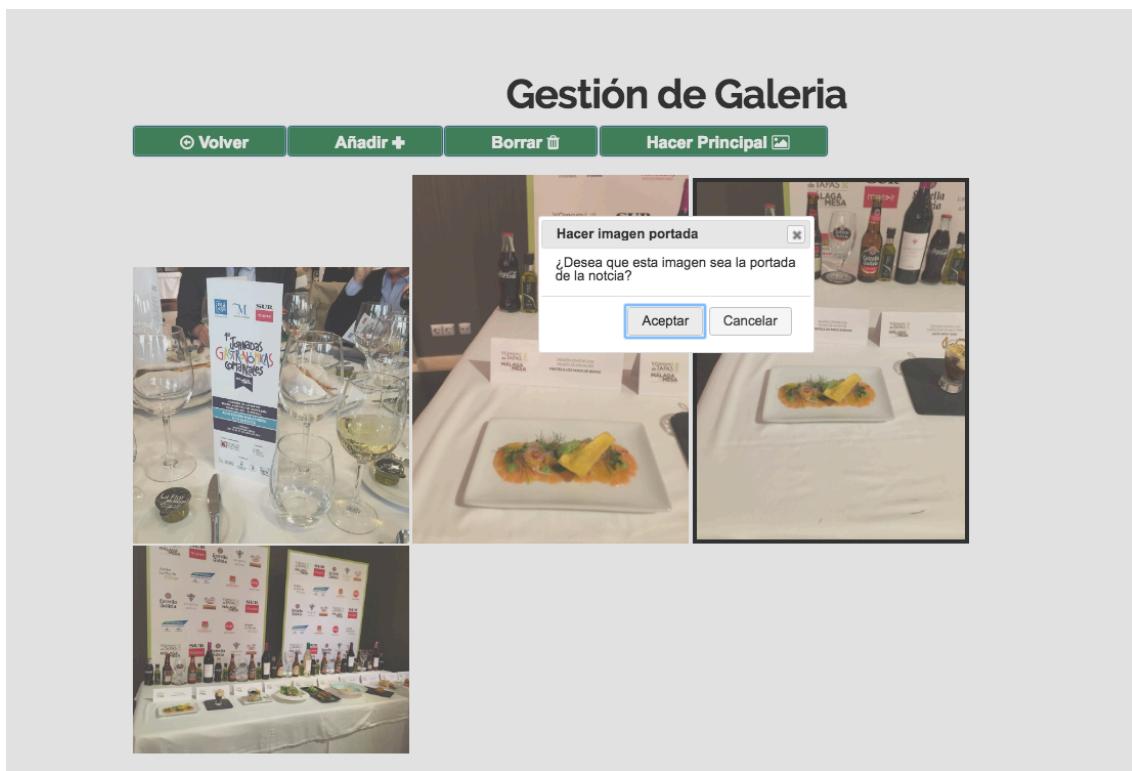
    });
    if(contador > 1){
        $.notify("Para hacer principal la imagen solo puedes seleccionar una ");
        contador = 0;
    }else{
        if(contador == 1){
            $("#dialogoHacerPrincipial").dialog("open");
        }else{
            if(contador == 0){
                $.notify("Debes seleccionar un elemento como minimo");
                contador = 0;
            }
        }
    }
});
```

Cuando se abre el diálogo de hacer principal, da dos opciones, la de **Aceptar** y **Cancelar**. Si se pincha sobre aceptar, se redirige al controlador **hacerImagenPrincipal.php** pasando el ID de la noticia y mostrando un mensaje de “**La imagen seleccionada ha sido establecida como principal**” en caso de que todo se haya hecho correctamente. Si por el contrario se pincha en cancelar, se cancela la acción.

```
$("#dialogoHacerPrincipial").dialog({
    autoOpen: false,
    resizable: false,
    modal: true,
    buttons: {
        "Aceptar": function () {
            //console.log(idImagen);
            $.get("Controller/hacerImagenPrincipal.php?idNoticia=" + idNoticia+"&nombreImagen=" + nombreImagen, function () {
                $.notify("La imagen seleccionada ha sido establecida como principal","success");
            });
            contador = 0;
            $(this).dialog("close");
        },
        "Cancelar": function () {
            $(this).dialog("close");
        }
    }
});
```

```
<?php
require_once '../Model/Noticia.php';

$noticiaAux = new Noticia($_GET['idNoticia'], "", "", "", "", "", $_GET['nombreImagen'], "", "", "");
//var_dump($noticiaAux);
//die();
$noticiaAux->updateImagen();
header("Location: ../panelAdmin.php");
```



Los mensajes de error y el de éxito los he puesto con **notify**, usando la librería **notify.min.js**.

```
<script src="js/notify.min.js" type="text/javascript"></script>
```