

Cómo poner una marca de agua a las imágenes con PHP



Hay ocasiones en las que necesitamos poder poner algún tipo de marca de agua a las imágenes que subimos a nuestra [página web](#). Con esto las estamos protegiendo y evitando que otras las puedan utilizar libremente. Si utilizamos algún tipo de CMS, como puedes ser [WordPress](#), podemos encontrarnos plugins que se encargan de hacer este proceso. Pero si no es así, y tenemos un desarrollo a medida, deberemos crearnos nuestra propia función que se encargue de añadir esa marca de agua.

Esto es precisamente lo que veremos a lo largo de nuestro [White Paper](#) de este mes. Crearemos el código necesario para poder añadir una “watermark” a las imágenes que vayamos subiendo. Para el ejemplo que hoy llevaremos a cabo, utilizaremos dos archivos:

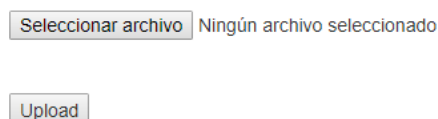
- **index.php:** Aquí será donde implementemos el formulario que nos permitirá realizar la subida de la imagen a la que añadiremos la marca de agua.
- **watermark.php:** Será donde tengamos todas las instrucciones encargadas de añadir la marca de agua.

Veamos a continuación cada uno de estos archivos.

Código index.php

Lo primero de todo será crearnos un pequeño formulario que nos permitirá subir la imagen a la que queremos añadir la marca de agua. El aspecto de este formulario será el siguiente.

Ejemplo añadir marca de agua con PHP



El código encargado de pintar esto por pantalla es el siguiente:

HTML

```
<h2>Ejemplo añadir marca de agua con PHP</h2>
<br>
<form action="" method="post" enctype="multipart/form-data">
  <input type="file" name="image" value="">
  <br><br>
  <input type="submit" value="Upload">
</form>
```

Como se puede observar, se trata de código HTML sencillo con un campo “file” que será el encargado de permitir seleccionar las imágenes.

En este **index.php**, además de este código, también tendremos toda la programación que se encarga de la subida del archivo al servidor, además de realizar algunas comprobaciones previas para asegurarnos que todo funcione bien. El código encargado de realizar esto es el siguiente:

PHP

```
if(isset($_FILES['image']['name'])){
    if(getimagesize($_FILES['image']['tmp_name']) < (1024*1024*1024*1024)){
        echo 'La imagen supera los 2 MB de tamaño. Elije otra más pequeña';
    }else{
        list($txt, $ext) = explode(".", $_FILES['image']['name']);
        $file_name = "images/watermark".$ext;
        $upload = copy($_FILES['image']['tmp_name'], $file_name);
        if($upload == true){
            $WaterMark = 'marca-agua.png';
            addImageWatermark ($file_name, $WaterMark, $file_name, 50);
            echo '<br><br>';
        }else{
            echo 'Error subiendo imagen';
        }
    }
}
```

En el código anterior, lo primero que hacemos es comprobar que se ha enviado la imagen desde el formulario. Esto lo podremos saber si la variable “\$_FILES” de PHP, que tiene la información de la imagen subida, existe:

PHP

```
if(isset($_FILES['image']['name'])){
    .....
}
```

A continuación, lo que hacemos es recuperar el tamaño de la imagen para ver que no sobrepase el límite que hayamos establecido. En nuestro ejemplo, si supera los 2 MB lanzaremos un error y el proceso no seguirá.

PHP

```
if(getimagesize($_FILES['image']['tmp_name']) < (1024*1024*1024*1024)){
    echo 'La imagen super los 2 MB de tamaño. Elije otra más pequeña';
}
```

En el caso de que la imagen subida al servidor sea menor del límite indicado, empezaremos el proceso de insertarle nuestra marca de agua. Para ello, lo primero que haremos será dividir el nombre de la imagen subida para tener por un lado el nombre y por otro la extensión. Para ello utilizaremos la función “explode” de PHP que se encarga de partir una cadena de texto según el limitador indicado. En nuestro caso será el punto (.) que es lo que separa el nombre de la extensión. Los trozos de esa partición se almacenarán en las variables “\$txt” y “\$ext”.

PHP

```
list($txt, $ext) = explode(".", $_FILES['image']['name']);
```

Lo siguiente que haremos será indicar el nombre de la imagen que vamos a crear junto con la marca de agua.

PHP

```
$file_name = "images/watermark." . $ext;
```

Tras esto, utilizamos la función **“copy”** de PHP para copiar la imagen al **servidor**. Si la subida es correcta, entonces se ejecutaría la llamada de la función **“addImageWatermark”** que se encargará de añadir la marca de agua. En caso de no poderse subir la imagen por cualquier motivo, mostraremos un error por pantalla.

PHP

```
if($upload == true) {
    $WaterMark = 'marca-agua.png';
    addImageWatermark ($file_name, $WaterMark, $file_name, 50);
    echo '<br><br>';
} else {
    echo 'Error subiendo imagen';
}
```

En el código anterior también se pinta la imagen generada con la marca de agua.

```
'<br><br>';
```

Código watermark.php

En este segundo archivo lo único que tendremos será el código de la función **“addImageWatermark”**, una función a la que le pasaremos cuatro parámetros.

- La imagen subida
- La marca de agua
- Fichero de destino
- Opacidad

El código de la función completa sería el siguiente:

PHP

```
function addImageWatermark($SourceFile, $WaterMark, $DestinationFile=NULL,
$opacity) {
    $main_img = $SourceFile;
    $watermark_img = $WaterMark;
    $padding = 5;
    $opacity = $opacity;

    $watermark = imagecreatefrompng($watermark_img);
    $image = imagecreatefromjpeg($main_img);

    if(!$image || !$watermark) die("Error: La imagen o la marca de agua no puede
ser cargada");
```

```

$watermark_size = getimagesize($watermark_img);
$watermark_width = $watermark_size[0];
$watermark_height = $watermark_size[1];
$image_size = getimagesize($main_img);
$dest_x = $image_size[0] - $watermark_width - $padding;
$dest_y = $image_size[1] - $watermark_height - $padding;

imagecopymerge($image, $watermark, $dest_x, $dest_y, 0, 0, $watermark_width,
$watermark_height, $opacity);

if ($DestinationFile<>'') {
    imagejpeg($image, $DestinationFile, 100);
} else {
    header('Content-Type: image/jpeg');
    imagejpeg($image);
}

imagedestroy($image);
imagedestroy($watermark);
}

```

En el código anterior, lo primero que hacemos es asignar una serie de valores a unas variables que utilizaremos dentro de esta función.

PHP

```

$main_img = $SourceFile;
$watermark_img = $WaterMark;
$padding = 5;
$opacity = $opacity;

```

Los dos primeros hacen referencia a la imagen subida y a la imagen que utilizaremos como marca de agua. El “**\$padding**” lo utilizaremos para colocar la marca de agua. Por último recogemos la opacidad de la imagen.

El siguiente paso será crear nuevas imágenes partiendo de la marca de agua y la imagen que hayamos subido.

PHP

```

$watermark = imagecreatefrompng($watermark_img);
$image = imagecreatefromjpeg($main_img);

```

Comprobamos que estas se hayan creado correctamente. Si no es así, lanzamos un error avisando de la situación.

PHP

```

if(!$image || !$watermark) die("Error: La imagen o la marca de agua no puede ser cargada");

```

Lo siguiente que haremos será sacar el tamaño de la marca de agua. Para ello hacemos uso de la función de PHP “**getimagesize**”, que devuelve un array con el ancho y el alto. Cada uno de estos valores lo almacenaremos en una variable independiente.

PHP

```
$watermark_size = getimagesize($watermark_img);  
$watermark_width = $watermark_size[0];  
$watermark_height = $watermark_size[1];
```

Hacemos lo mismo con la imagen principal.

PHP

```
$image_size = getimagesize($main_img);
```

Calculamos las coordenadas X e Y donde colocaremos la marca de agua. Esta información la sacaremos de restar al tamaño de la imagen principal el tamaño de la marca de agua y el padding.

PHP

```
$dest_x = $image_size[0] - $watermark_width - $padding;  
$dest_y = $image_size[1] - $watermark_height - $padding;
```

Una vez que tengamos toda esta información mezclamos las dos imágenes. Lo que hacemos es copiar la marca de agua en la imagen principal partiendo de las coordenadas indicadas.

PHP

```
imagecopymerge($image, $watermark, $dest_x, $dest_y, 0, 0, $watermark_width,  
$watermark_height, $opacity);
```

Lo que nos falta es indicar dónde queremos guardar la imagen creada. Esto dependerá de la variable “\$DestinationFile” que se le pasa a la función. Si esta tiene valor, se guarda en el archivo indicado. Si por el contrario está vacío, entonces se muestra por el navegador.

PHP

```
if ($DestinationFile<>'') {  
    imagejpeg($image, $DestinationFile, 100);  
} else {  
    header('Content-Type: image/jpeg');  
    imagejpeg($image);  
}
```

Para finalizar, eliminamos las imágenes creadas en pasos anteriores.

PHP

```
imagedestroy($image);  
imagedestroy($watermark);
```

Gracias a este código es posible poner una marca de agua a vuestras imágenes, para evitar que la puedan utilizar sin vuestro consentimiento.