

Cómo utilizar Recaptcha de Google en un formulario



En cualquier página web que visitemos es normal encontrarnos un **formulario de contacto** con el que poder solicitar información sobre algún tipo de producto o enviar nuestras consultas. Estos formularios también se han convertido, desde hace años, en uno de los objetivos principales de los spammers a la hora de bombardearnos con sus correos basuras. Para evitar esta situación, el uso de captcha se ha convertido en algo imprescindible, aunque dependiendo del sistema utilizado, puede resultar más o menos molesto para los usuarios.



Marca la casilla *:

No soy un robot

reCAPTCHA
Privacidad - Condiciones

ENVIAR SOLICITUD

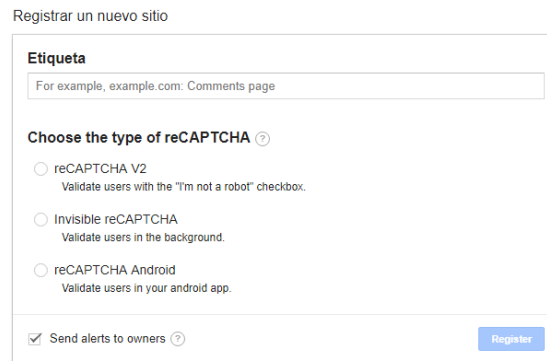
* campos obligatorios

Hasta hace bien poco, **lo más habitual era tener que introducir dos palabras** que aparecían en una imagen para demostrar que no éramos un bot. Ahora gracias a Google y a su versión 2.0 de reCaptcha, podemos implementar un sistema más sencillo y más cómodo para los usuarios, en el que únicamente habrá que indicar que no somos un robot.

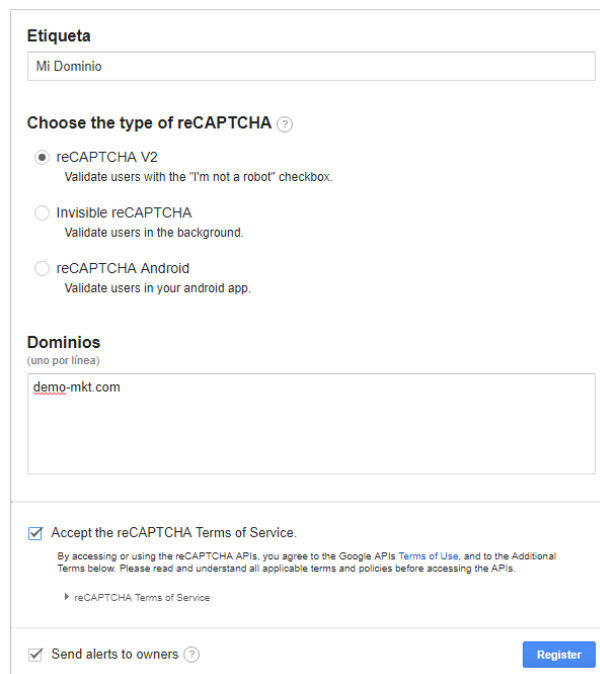
Lo que veremos a lo largo de nuestro **White Paper** serán los pasos que debemos seguir para configurar y utilizar esta nueva versión de reCaptcha con PHP.

Registrar nuestra web en la administración de reCaptcha

Para empezar a utilizar este nuevo sistema de captcha, lo primero de todo será conseguir una clave API, cosa que podremos hacer desde el [panel de administración de reCaptcha](#). Para acceder a este apartado, será necesario haber iniciado sesión en tu cuenta de Google. Una vez dentro, lo primero que nos pedirá será que registremos nuestro sitio web.



En esa primera pantalla deberemos rellenar el campo “etiqueta” donde daremos un nombre descriptivo, por ejemplo el nombre de la web. Tras eso, seleccionaremos la opción “reCaptcha V2”. Tras esto, se nos desplegará una nueva opción donde poder indicar los **dominios** donde vamos a utilizar este sistema.



A la hora de indicar los dominios, estos deben de ir sin las “www”. En el caso de querer poner más de un dominio, cada uno de ellos se indicaría en una línea diferente. Es importante indicar que al introducir un dominio también estaremos habilitando todos los subdominios que pudiera tener.

Por último, aceptaremos los términos del servicio y pulsaremos sobre el botón “Registrar”. Si todo ha ido bien, tendremos que ver por pantalla las claves que tendremos que utilizar en nuestra **web**.

Debajo de las claves verás algunos fragmentos de código que deberemos incluir en nuestro sitio para poder utilizar esta versión de reCaptcha en nuestros formularios.

Pega este fragmento antes de la etiqueta </head> de cierre en la plantilla HTML:

```
<script src='https://www.google.com/recaptcha/api.js'></script>
```

Pega este fragmento al final del elemento <form> donde quieras que aparezca el widget de reCAPTCHA:

```
<div class="g-recaptcha" data-sitekey="6LfT80cUAAAAAIz3Fd8jtISYH43bqsOamAi9E019"></div>
```

Una vez que tenemos todo esto, será el momento de implementarlo en nuestros formularios.

Cómo utilizar reCaptcha V2 en un formulario

Partiremos de la premisa de que ya tenemos creado nuestro formulario en formato HTML, un formulario que puede tener todos los campos que queramos. En nuestro ejemplo, haremos uso de uno que tendrá tres campos: Nombre, **Email** y Mensaje. El código encargado de pintar nuestro formulario es el siguiente.

```
<html>
  <head>
    <title>Formulario de envío con reCaptcha</title>
  </head>
  <body>
    <form action="formulario.php" method="post">
      <input name="name" placeholder="Nombre"/>
      <input name="email" placeholder="Email"/>
      <textarea placeholder="Tu mensaje" name="message"></textarea>
      <button type="submit" name="submit">Enviar</button>
    </form>
  </body>
</html>
```

Este código nos pintaría el siguiente formulario.

Dentro de ese código, deberemos añadir la librería que proporciona Google para hacer uso de este sistema. Para ello, entre las etiquetas “<head>” y “</head>”, añadiremos esta línea.

```
<script src='https://www.google.com/recaptcha/api.js'></script>
```

También deberemos añadir el código que pintará ese campo y que podremos colocar donde queramos, aunque lo más habitual es hacerlo al final del formulario, antes del botón de “Enviar”. El código que se encarga de pintar esto es el siguiente.

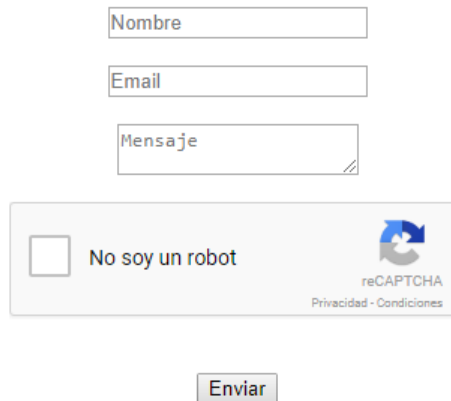
```
<div class="g-recaptcha" data-sitekey="MIAPI"></div>
```

En este caso, deberemos sustituir “API” por la clave que nos ha proporcionado Google al registrar nuestro sitio.

Hecho todo esto, el código final quedaría de la siguiente forma.

```
<html>
  <head>
    <title>Formulario de envío con reCaptcha</title>
    <script src='https://www.google.com/recaptcha/api.js'></script>
  </head>
  <body>
    <center>
      <form action="formulario.php" method="post">
        <input name="name" placeholder="Nombre"/><br/><br/>
        <input name="email" placeholder="Email"/><br/><br/>
        <textarea placeholder="Mensaje" name="message"></textarea><br/><br/>
        <div class="g-recaptcha" data-sitekey="MIAPI"></div><br/><br/>
        <button type="submit" name="submit">Enviar</button>
      </form>
    </center>
  </body>
</html>
```

Y por pantalla ya deberíamos ver el nuevo campo “No soy un robot”.



The image shows a web form with three input fields: 'Nombre', 'Email', and 'Mensaje'. Below these fields is a reCAPTCHA widget containing a checkbox labeled 'No soy un robot' and the reCAPTCHA logo with the text 'reCAPTCHA Privacidad - Condiciones'. At the bottom of the form is an 'Enviar' button.

Con todo lo visto en este paso, hemos conseguido que se pinte el campo, pero ahora falta controlar que no se envíe el formulario mientras que no se marque la casilla de verificación. Para ello, deberemos controlarlo a nivel de **servidor**.

Verificando que no eres un robot

Para verificar que realmente se ha seleccionado la casilla, lo primero de todo será recuperar el valor que es enviado mediante POST. Para ello, actuamos de la misma forma que lo hacemos con el resto de campos del formulario. Aquí hay que destacar que el nombre del campo reCaptcha es “**g-recaptcha-response**”. Así que para recuperarlo lo haríamos de la siguiente manera.

```
$recaptcha = $_POST["g-recaptcha-response"];
```

Ya que tenemos ese valor, ahora hay que validar si la verificación ha sido satisfactoria. Esto se debe hacer haciendo uso de la url <https://www.google.com/recaptcha/api/siteverify>. Esta url la almacenaremos en una variable para utilizarla posteriormente.

```
$url = 'https://www.google.com/recaptcha/api/siteverify';
```

Ahora creamos un array donde almacenaremos nuestro “**API-SECRET**” que obtuvimos cuando hicimos el registro, y el valor recibido del reCaptcha mediante POST.

```
$data = array(  
    'secret' => 'API-SECRET',  
    'response' => $recaptcha  
);
```

A continuación crearemos otro array donde almacenaremos el array creado anteriormente y el método utilizado para realizar la verificación.

```
$options = array(
    'http' => array (
        'method' => 'POST',
        'content' => http_build_query($data)
    )
);
```

Para finalizar la verificación, lo único que nos hace falta es conocer si se ha realizado la selección de la casilla, valor que vendrá devuelto en formato JSON.

```
$context = stream_context_create($options);
$verify = file_get_contents($url, false, $context);
$captcha_success = json_decode($verify);
```

`$captcha_success` será un objeto que tendrá un campo “success”. Si este campo tiene el valor 1 entonces la verificación ha sido correcta. Si no es así, no se ha pasado la verificación y por lo tanto no se debería realizar el envío del formulario.

```
if ($captcha_success->success) {
    echo 'Se envía el formulario';
} else {
    echo 'No se envía el formulario';
}
```

El código completo quedaría de la siguiente forma.

```
$recaptcha = $_POST["g-recaptcha-response"];

$url = 'https://www.google.com/recaptcha/api/siteverify';
$data = array(
    'secret' => 'API-SECRET',
    'response' => $recaptcha
);

$options = array(
    'http' => array (
        'method' => 'POST',
        'content' => http_build_query($data)
    )
);

$context = stream_context_create($options);
$verify = file_get_contents($url, false, $context);
$captcha_success = json_decode($verify);

if ($captcha_success->success) {
    echo 'Se envía el formulario';
} else {
    echo 'No se envía el formulario';
}
```