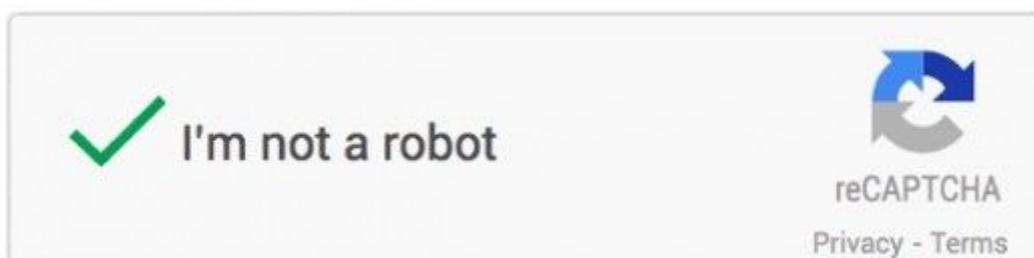


## DAWeb práctica 10, día 25 de abril de 2018

### Configuración de reCAPTCHA 2.0 con Demo/Tutorial PHP

Actualmente reCAPTCHA es extremadamente importante para verificar que los bots no están tratando de tomar ventaja de un sitio en internet. reCAPTCHA 2.0 es un poco diferente al antiguo reCAPTCHA. Ya no funciona escribiendo texto en un cuadro, es simplemente el clic de un botón.



El término CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*) es una marca registrada de la Universidad Carnegie-Mellon (EE.UU.) que designa una familia de pruebas Turing que automáticamente diferencian a un usuario humano de un equipo informático.

Esta es una prueba de desafío-respuesta utilizada en informática, para asegurar que una respuesta no sea generada por un ordenador. Siendo realizado por un ordenador, a diferencia de las pruebas estándar de Turing en humanos, un

CAPTCHA es a menudo descrito como una prueba de Turing inversa. Sin embargo, el término es ambiguo porque también podría significar que los participantes están tratando de demostrar que son ordenadores.

## reCAPTCHA con PHP

El método de verificación de usuarios es un poco diferente a versiones previas. "g-recaptcha-response" se envía a través de una petición POST a un *script* del lado del servidor, y junto con la clave reCAPTCHA que se obtiene cuando se realiza el registro en Google, se pasa a Google para saber si el usuario es un *bot* o no. La respuesta es una respuesta JSON (notación de objetos JavaScript es un formato de datos de texto derivado de la notación de objetos de lenguaje JavaScript. Permite representar información estructurada como lo permite por ejemplo XML) así que se usa `file_get_contents()` y `json_decode()` para obtener la respuesta y decidir qué hacer desde allí con el *script*.

Lo que sigue es un simple formulario HTML que lo más probable es que se vea como un formulario de contacto habitual. El div con la clase de "g-recaptcha" es donde se coloca el captcha. Se debe sustituir la clave de la ubicación de datos por la clave que se obtiene de Google cuando se registra (<https://www.google.com/recaptcha/intro/index.html>).

▼ Claves

<b>Clave del sitio</b> Úsala en el código HTML que muestra tu sitio a los usuarios.	<b>Clave secreta</b> Úsala para las comunicaciones entre tu sitio y Google. Ten la precaución de no revelársela a nadie.
6LfpXVYUAAAAM2xFK4e5nCKZsW4yc_3Mw7gu4tV	[Redacted]

▼ Step 1: Client side integration

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

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

HA Ejemplo1 (www.um.es)

▼ Paso 2: integración en el lado del servidor

When your users submit the form where you integrated reCAPTCHA, you'll get as part of the payload a string with the name "g-recaptcha-response". In order to check whether Google has verified that user, send a POST request with these parameters:

URL: `https://www.google.com/recaptcha/api/siteverify`

<b>secret</b> (required)	[Redacted]
<b>response</b> (required)	El valor de "g-recaptcha-response".
<b>remoteip</b>	The end user's ip address.

## Ejemplo:

### Código HTML

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>Ejemplo recaptcha</title>
<script src='https://www.google.com/recaptcha/api.js'></script>
</head>
<body>
<form          action="recaptcha-ejemplo.php"          method="post"
enctype="multipart/form-data">
  <input name="envia_nombre" placeholder="Nombre..."/>
  <input name="envia_email" placeholder="email..."/>
  <input name="envia_mensaje" placeholder="Mensaje..." >

  <div class="g-recaptcha"
data-sitekey="6LfpXVYUAAAAAM2xFK4e5nCKZsW4yc_3MW7gu4tV"></div>
  <button type="submit" id="envia_mensaje">Enviar</button>
</form>
</body>
</html>
```

### Código PHP

```
<?php
$envia_nombre = stripslashes($_POST["envia_nombre"]);
$envia_email = stripslashes($_POST["envia_email"]);
$envia_mensaje = stripslashes($_POST["envia_mensaje"]);
$response = $_POST["g-recaptcha-response"];
$url = 'https://www.google.com/recaptcha/api/siteverify';
$data = array(
    'secret' => '*****',
    'response' => $_POST["g-recaptcha-response"]
);
$options = array(
    'http' => array (
        'method' => 'POST',
        'content' => http_build_query($data)
```

```
)  
);  
$context = stream_context_create($options);  
$verify = file_get_contents($url, false, $context);  
$captcha_success=json_decode($verify);  
if ($captcha_success->success==false) {  
    echo "<p>Eres un bot</p>";  
} else if ($captcha_success->success==true) {  
    echo "<p>No eres un bot</p>";  
}  
?>
```

Pulsa [aquí](#) para acceder al ejemplo.

El programa de Google que gestiona reCAPTCHA en PHP, se encuentra en [GitHub](#).