

ENVIAR EMAIL CON PHP

ECada vez más, es necesario enviar correos electrónicos desde nuestra aplicación web, por ejemplo para:

- Autenticar cuentas
- Restaurar contraseñas
- Newsletter

OBJETIVO

Analizar los elementos que necesitamos y crear de una manera sencilla un entorno que nos permita integrar el envío de mail desde nuestra aplicación web.



¿QUÉ NECESITAMOS?

- Servidor de correo
- Saber cómo utilizar PHP para el envío de correo

SERVIDOR DE CORREO

Para poder enviar correos electrónicos necesitamos un servidor de correo. Tenemos diferentes opciones:

- Envío de correo a través de un tercero, por ejemplo Gmail
- Configurar nuestro propio servidor de correo.
- Utilizar el servicio de correo de un hosting, por ejemplo Hostinger
- Utilizar un servicio de envío masivo

CONFIGURAR GMAIL PARA EL ENVÍO DE CORREOS DESDE APLICACIÓN PHP

Nos va a permitir realizar pruebas sin necesidad de configurar ningún servidor de correo ni utilizar un servicio de hosting. Posteriormente, al hacer el despliegue es recomendable configurar nuestro propio servicio de correo o utilizar el del alojamiento ya que Gmail nos limita a 500 el número de mails por día.

PASOS A SEGUIR

1

Crearemos una cuenta gmail para realizar nuestras pruebas.

2

En el apartado de configuración -> seguridad activaremos la verificación en dos pasos.

3

En el apartado de configuración -> seguridad generaremos la contraseña de aplicaciones

Una vez hecho esto ya podemos empezar con nuestros ejemplos

PHP

Podemos hacerlo de dos maneras diferentes:

- Configurando php.ini y sendmail.ini y utilizar la función mail de php.
- Utilizar la clase PHPMailer. No es necesario modificar los ficheros de configuración.

UTILIZAR LA CLASE PHPMAILER

Nos facilita la tarea porque no es necesario modificar los ficheros de configuración y además nos permite enviar correos HTML y ficheros adjuntos de una manera sencilla.

CON LA FUNCIÓN MAIL() DE PHP

Es necesario configurar los ficheros php.ini y sendmail.ini. Una vez hecho esto inicialmente es más sencillo que PHPMailer

PASOS A SEGUIR

1

Mirar la documentación y descargar desde <https://github.com/PHPMailer/PHPMailer>. Podemos descargar la clase en formato zip desde github o utilizando composer

2

Una vez hecho esto ya podemos empezar con nuestros ejemplos

PASOS A SEGUIR

1

❑ Abrir `php.ini` que encontrarás en: `C:\xampp\php\php.ini`
Modificar los siguientes parámetros:
❑ `extension=php_openssl.dll` (quitarle el punto y coma)
❑ `SMTP=smtp.gmail.com`
`smtp_port=587`
`sendmail_from = tu_direccion_de_correo@gmail.com`
`sendmail_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"`

2

❑ Abrir el archivo `sendmail.ini`. Modificar los siguientes parámetros:
`smtp_server=smtp.gmail.com`
`smtp_port=587`
`error_logfile=error.log`
`debug_logfile=debug.log`
`auth_username=tu_direccion_de_correo_electronico@gmail.com`
`auth_password=el_password_generado`
`force_sender=tu_direccion_de_correo_electronico@gmail.com`

Una vez hecho esto ya podemos empezar con nuestros ejemplos



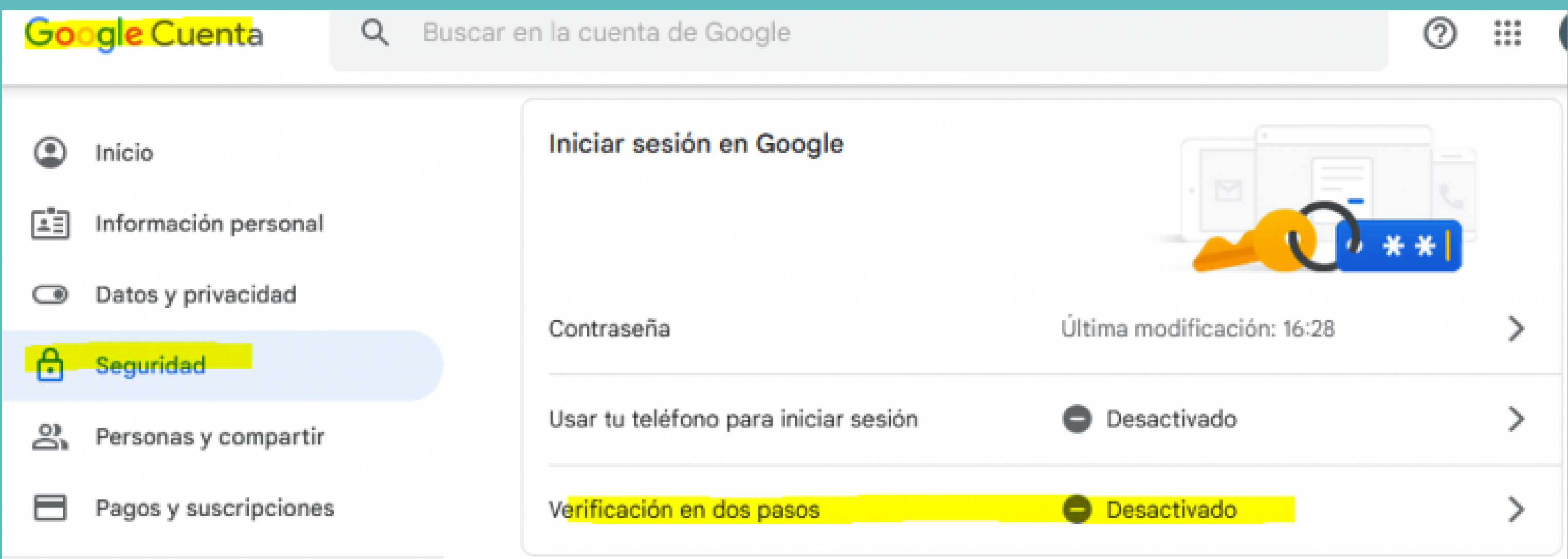
Pasos para configurar nuestra cuenta gmail para enviar correos desde aplicaciones



1

Después de crear nuestra cuenta de prueba, lo primero que necesitaremos es habilitar la verificación en dos pasos y lo haremos clicando en :

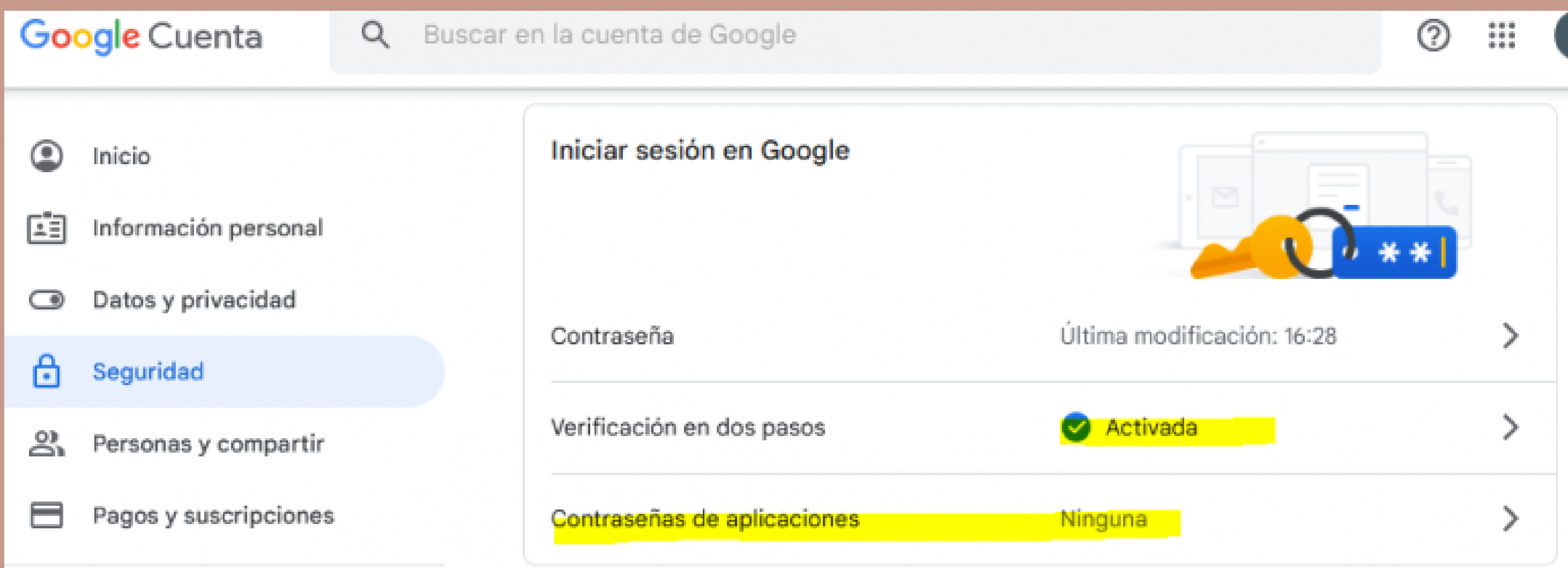
Gestionar tu cuenta de Google -> Seguridad -> Verificación en dos pasos



Nos solicitará un número de teléfono para enviar un código de verificación

2

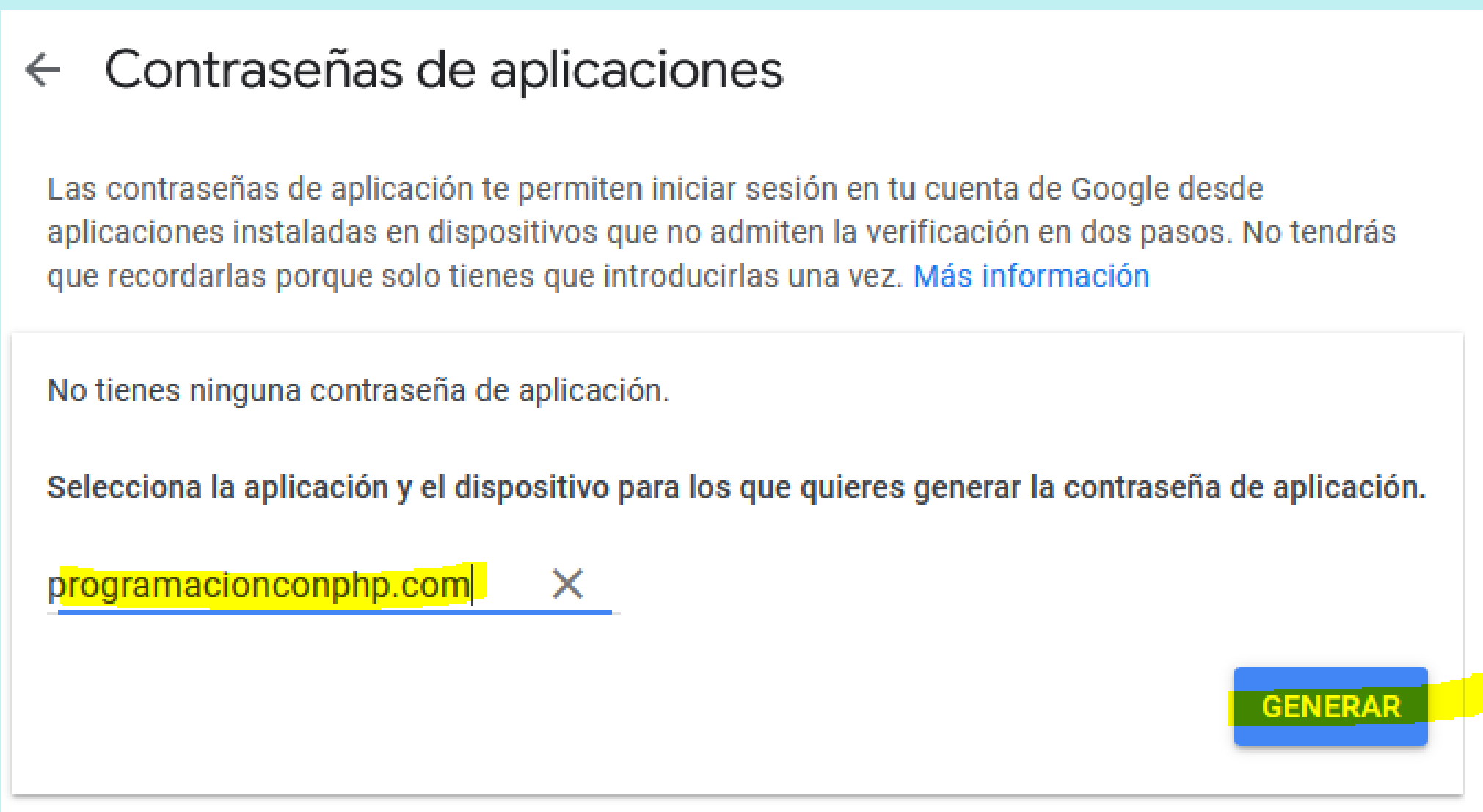
Ahora generaremos la contraseña de aplicaciones
Gestionar tu cuenta de Google -> Seguridad -> Contraseña de aplicaciones



Seleccionamos **Otra**

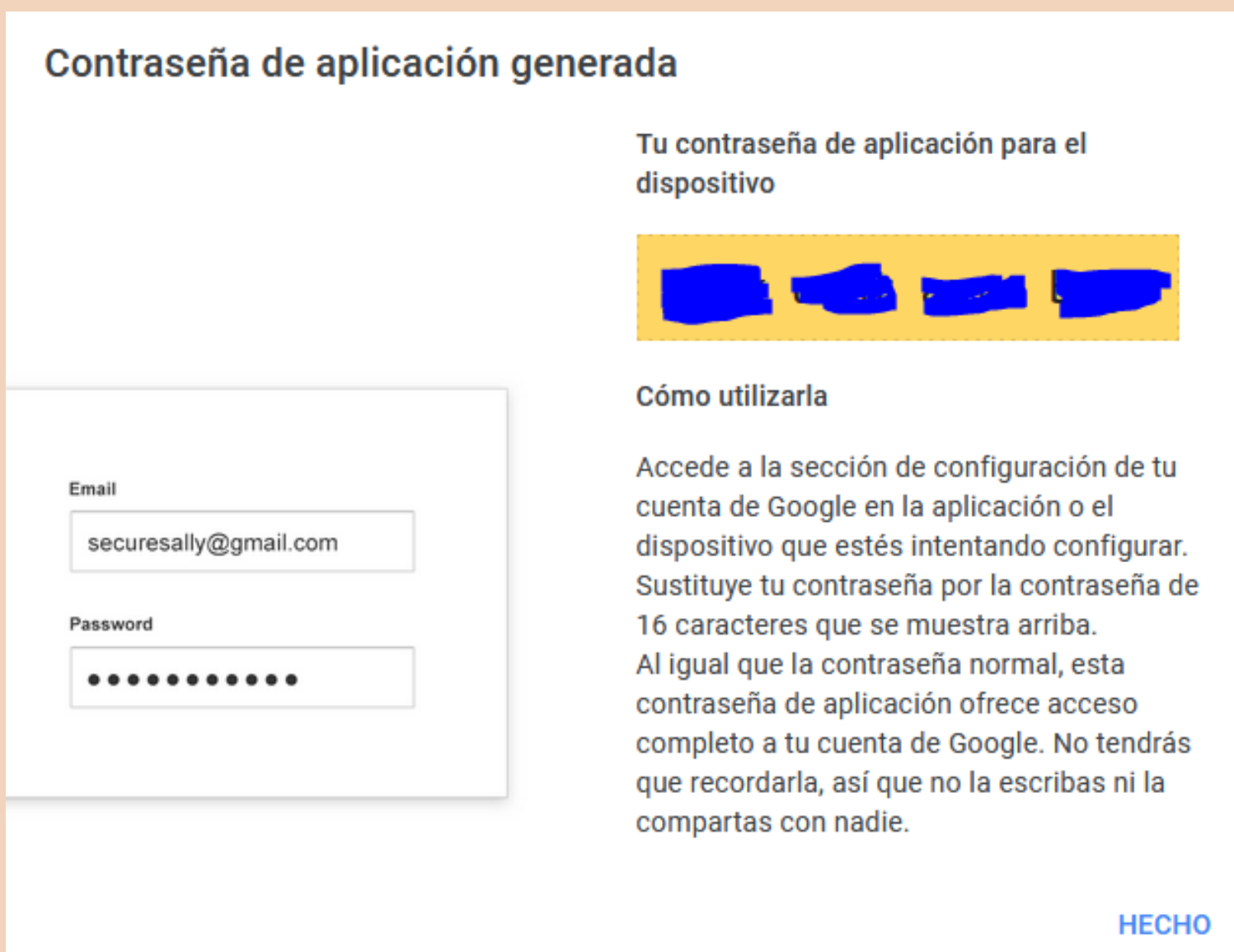
3

Le daremos el nombre que queramos



4

Para terminar pulsaremos **Generar** y nos mostrará la contraseña que nos ha geerado y que tendremos que utilizar en nuestras aplicaciones



Copia la contraseña para poder utilizarla posteriormente



```
<?php
//Carga de las clases necesarias
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
require 'PHPMailer/src/Exception.php';
require 'PHPMailer/src/PHPMailer.php';
require 'PHPMailer/src/SMTP.php';
```

Suponemos que hemos descomprimido el fichero descargado de
PHP Mailer en una carpeta llamada PHPMailer. (revisar el ejemplo)

//Crear una instancia. Con true permitimos excepciones

```
$mail = new PHPMailer(true);
```

```
try {
```

//Valores dependientes del servidor que utilizamos

```
$mail->isSMTP(); //Para usar SMTP
$mail->Host = 'smtp.gmail.com'; //Nuestro servidor SMTP smtp.gmail.com en caso de usar gmail
$mail->SMTPAuth = true;
/*
 * SMTP username y password Poned los vuestros. La contraseña es la que nos generó GMAIL
 */
$mail->Username = 'tuusuario@gmail.com';
$mail->Password = 'tu contraseña generada';
/*
 * Encriptación a usar ssl o tls, dependiendo cual usemos hay que utilizar uno u otro puerto
 */
$mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;
$mail->Port = "465";
/**TCP port to connect to; use 587 if you have set `SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS`
 * $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
 * $mail->Port = 587;
 */
```

Crearemos la instancia de PHPMailer pasándole true para poder
recoger como excepción en caso de que haya un error en el
envío del correo

Este grupo de valores dependen del servidor que utilicemos y de
nuestro usuario y contraseña generada.

En nuestro código siempre será igual por lo que podemos separarlo

```
/*
Receptores y remitente
*/
//Remitente
$mail->setFrom('tuUsuario@gmail.com', 'Tu nombre');
//Receptores. Podemos añadir más de uno. El segundo argumento es opcional, es el nombre
$mail->addAddress('destinatario', 'Nombre destinatario opcional'); //Add a recipient
$mail->addAddress('ejemplo@example.com');

//Copia
$mail->addCC('cc@example.com');
//Copia Oculta
$mail->addBCC('bcc@example.com');

//Archivos adjuntos
$mail->addAttachment('/var/tmp/file.tar.gz'); //Add attachments
$mail->addAttachment('/tmp/image.jpg', 'new.jpg'); //Optional name
```

En esta parte del código configuramos el nombre que aparecerá asociado
a nuestra cuenta en el correo y el receptor o receptores del correo

```
//Contenido
//Si enviamos HTML
$mail->isHTML(true);
$mail->CharSet = "UTF8";
//Asunto
$mail->Subject = 'Here is the subject';
//Contenido HTML
$mail->Body = 'This is the HTML message body <b>in bold!</b>';
//Contenido alternativo en texto simple
$mail->AltBody = 'This is the body in plain text for non-HTML mail clients';
//Enviar correo
$mail->send();
echo 'El mensaje se ha enviado con éxito';
} catch (Exception $e) {
    echo "El mensaje no se ha enviado: {$mail->ErrorInfo}";
}
?>
```

Actualmente en la mayoría de los casos se envían correos HTML pero
es aconsejable enviar un texto alternativo para lectores que no admitan
HTML

