

## PHP 3.8

### PHP MAILER

#### 1. INTRODUCCIÓN

*PHPMailer* es una de las bibliotecas más populares y robustas para enviar correos electrónicos desde aplicaciones PHP.

PHPMailer es ampliamente utilizada en proyectos web debido a su facilidad de uso, su flexibilidad y su capacidad para enviar correos electrónicos seguros y personalizados.

PHPMailer es particularmente útil cuando se necesita interactuar con servidores de correo como **Gmail**, **Outlook** o servidores personalizados que requieren **autenticación**.

Puede manejar correos electrónicos con configuraciones avanzadas como la codificación de caracteres, la autenticación mediante SSL/TLS, la autenticación SMTP, etc

#### 2. MANEJO DE PHPMAILER

PHPMailer permite enviar correos electrónicos a través de varios métodos, siendo el más común el uso del protocolo *SMTP*, *Simple Mail Transfer Protocol*, en lugar de la función *mail()* de PHP, que no siempre es confiable o flexible).

**Instalación y configuración:** Para usar PHPMailer, primero es necesario instalar la biblioteca. Esto se puede hacer fácilmente a través de **Composer**, un gestor de dependencias para PHP. Una vez instalada, se debe incluir el archivo ***autoload.php*** generado por Composer, que carga automáticamente las clases necesarias. Si no estás usando Composer, puedes incluir manualmente los archivos PHP de PHPMailer.

```
require 'path/to/PHPMailer/vendor/autoload.php';
```

**Autenticación SMTP:** Una de las principales ventajas de PHPMailer es que puedes configurar la autenticación SMTP (*Simple Mail Transfer Protocol*) para enviar correos de manera más segura y confiable. Esto significa que podrás enviar correos electrónicos a través de servidores externos (como Gmail, Mailgun, etc.) que requieren autenticación.

PHPMailer permite la configuración de las credenciales necesarias para autenticarte con un servidor SMTP:

- nombre de usuario
- contraseña
- puerto
- método de encriptación

### Ejemplo

```
require 'vendor/autoload.php';

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

$mail = new PHPMailer();

$mail->isSMTP();
$mail->Host = 'smtp.gmail.com';
$mail->SMTPAuth = true;
$mail->Username = 'tuemail@gmail.com';
$mail->Password = 'tucontraseñaaplicación';
$mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
$mail->Port = 587;
```

**Envío de correos HTML y texto plano:** PHPMailer te permite enviar correos electrónicos tanto en formato texto plano como HTML. Esto es útil cuando necesitas enviar correos con formato enriquecido (por ejemplo, usando etiquetas HTML para dar formato al texto o agregar imágenes).

```
$mail->isHTML(true);
$mail->Body = '<h1>Este es un correo HTML</h1>';
```

**Archivos adjuntos:** PHPMailer también permite agregar archivos adjuntos a los correos electrónicos. Estos documentos pueden ser documentos, imágenes o cualquier otro tipo de archivo.

Se usa el método `addAttachment` para adjuntar uno o varios archivos.

```
$mail->addAttachment('ruta/a/archivo.pdf');
```

**Manejo de errores y depuración:** PHPMailer incluye un sistema de depuración que ayuda a identificar problemas con el envío de correos. El modo de depuración se activa mediante el método *SMTPDebug*. Esto muestra un registro detallado de los pasos realizados durante la conexión al servidor SMTP y el envío del correo, lo cual es útil para identificar errores.

```
$mail->SMTPDebug = 2;
```

**Configuración de direcciones de correo:** Es fácil establecer el remitente y destinatarios del correo. Con PHPMailer, puedes agregar múltiples destinatarios, copias (CC y BCC), y personalizar el remitente. La función *setFrom()* te permite establecer el correo electrónico y el nombre del remitente.

```
$mail->setFrom('tuemail@gmail.com', 'Tu Nombre');  
$mail->addAddress('destinatario@dominio.com');
```

**Seguridad con SSL/TLS:** PHPMailer permite configurar la encriptación SSL o TLS para garantizar que el correo se envíe de manera segura. TLS es una opción recomendada por ser más segura que el método de encriptación SSL más antiguo.

```
$mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
```

**Manejo de errores:** Si el correo no se puede enviar, PHPMailer captura los errores y proporciona mensajes descriptivos que facilitan la identificación del problema. Por ejemplo, si las credenciales SMTP son incorrectas o si hay un problema con la red, PHPMailer te dará una descripción detallada del error.

```
if(!$mail->send()) {  
    echo 'Error en el envío: ' . $mail->ErrorInfo;  
} else {  
    echo 'Correo enviado con éxito';  
}
```

### 3. INSTALACIÓN DE PHPMAILER

Si tenemos instalado el gestor de dependencias **composer**

Ir a la carpeta del proyecto



Abrir terminal de línea de comandos y escribir

```
composer require phpmailer/phpmailer
```

Veremos un montón de cosas:

```
./composer.json has been updated  
Running composer update phpmailer/phpmailer  
Loading composer repositories with package information  
Updating dependencies
```

Al final del proceso se me habrá instalado y habrá generado en la carpeta del proyecto muchas cosas, las más importantes:

- Archivo  **composer.json**
- Carpeta  **vendor**
- Archivo *autoload.php*

## 4. EJEMPLO ENVÍO DE EMAIL CON PHPMAILER

*correo\_test\_ALU.php*

```
<?php

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

// Asegúrate de tener autoload.php de Composer
require dirname(__FILE__) . "/vendor/autoload.php";

// Inicializar el objeto PHPMailer
$mail = new PHPMailer();
$mail->isSMTP();
$mail->SMTPDebug = 2; // Cambiar a 0 para no mostrar mensajes de depuración
$mail->SMTPAuth = true;
$mail->SMTPSecure = 'tls';
$mail->Host = 'smtp.gmail.com'; // Usar el servidor SMTP de Gmail
$mail->Port = 587; // Puerto para TLS
// Ingresar tus credenciales de Google
$mail->Username = "tuemail@gmail.com"; // Tu dirección de correo electrónico
$mail->Password = "tucontraseña de aplicación"; // Contraseña de la aplicación,
//si tienes activado 2FA

// Establecer el remitente y el asunto
$mail->setFrom('tuemail@gmail.com', 'Test');
$mail->Subject = 'Correo de prueba';

// Definir el cuerpo del mensaje
$mail->msgHTML('Este es un correo de prueba enviado usando PHPMailer.');
```

```
// Agregar archivo adjunto
$mail->addAttachment('adjunto_test.xsd');
```

```
// Destinatario
$address = 'destino@servidor.com';
$mail->addAddress($address, 'Test');
```

```
// Enviar el correo
$resul = $mail->send();

// Verificar si hubo éxito en el envío
if (!$resul) {
    echo '<br>ERROR DE ENVÍO: ' . $mail->ErrorInfo;
} else {
    echo 'Correo enviado exitosamente';
}
```

## Ejecución

Desde línea de comandos de la carpeta del proyecto:

*php correo\_test\_ALU.php*

Veremos muchas cosas:

```
2025-01-16 17:43:29    SERVER -> CLIENT: 250-smtp.gmail.com at your service, [2a02:9130:fd1c:49ca:1c95:38a0:57b2:367c]
                        250-SIZE 35882577
                        250-8BITMIME
```

....

```
2025-01-16 17:43:30    CLIENT -> SERVER: Este es un correo de prueba enviado usando PHPMailer.
2025-01-16 17:43:30    CLIENT -> SERVER:
```

.....

```
2025-01-16 17:43:31    SERVER -> CLIENT: 221 2.0.0 closing connection
Correo enviado exitosamente
```

## 5. EJEMPLO ENVÍO DE EMAIL CON PHPMAILER. VERSIÓN CON ENCRIPCIÓN DE CONTRASEÑA

Archivos necesarios:

- `encripta_password.php` → encargado de generar la contraseña encriptada a partir de la real
- `correo_test_encriptado.php` → encargado de desencriptar la contraseña recibida, generar objeto email y enviarlo
- `encripta_devuelve.php` → encargado de devolver la contraseña encriptada

*encripta\_password\_ALU.php*

```
// Define la contraseña que deseas encriptar
$password = '';

// Usa base64_encode para encriptar la contraseña
$encrypted_password = base64_encode($password);

// Muestra la contraseña encriptada
echo "Contraseña encriptada: " . $encrypted_password;
```

*encripta\_devuelve\_ALU.php*

```
return [
    'EMAIL_PASSWORD' => 'cGFnxXcAghjdkd'
];
```

*correo\_test\_encriptado\_ALU.php*

```

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

// Asegúrate de tener autoload.php de Composer
require dirname(__FILE__)."/vendor/autoload.php";

// Cargar configuración desde un archivo externo
$config = include 'encripta_devuelve_ALU.php';
$decrypted_password = base64_decode($config['EMAIL_PASSWORD']);

// Inicializar el objeto PHPMailer
$mail = new PHPMailer();
$mail->isSMTP();
$mail->SMTPDebug = 2; // Cambiar a 0 para no mostrar mensajes de depuración
$mail->SMTPAuth = true;
$mail->SMTPSecure = 'tls';
$mail->Host = 'smtp.gmail.com';
$mail->Port = 587;

// Ingresar tus credenciales de Google
$mail->Username = "tuemail@gmail.com";
$mail->Password = $decrypted_password;

// Establecer el remitente y el asunto
$mail->setFrom('tuemail@gmail.com', 'Test');
$mail->Subject = 'Correo de prueba';

// Definir el cuerpo del mensaje
$mail->msgHTML('Este es un correo de prueba enviado usando PHPMailer.');
```

```

// Agregar archivo adjunto
$mail->addAttachment('adjunto_test.xsd');
```

```

// Destinatario
$address = 'destino@servidor.com';
$mail->addAddress($address, 'Test');
```

```

// Enviar el correo
$resul = $mail->send();

// Verificar si hubo éxito en el envío
if (!$resul) {
    echo '<br>ERROR DE ENVÍO: ' . $mail->ErrorInfo;
} else {
    echo 'Correo enviado exitosamente';
}

```



## Ejecución

Desde línea de comandos de la carpeta del proyecto:

- *encripta\_password\_ALU.php*
- introducir la contraseña real que queremos encriptar: pepe
- nos devuelve la contraseña encriptada correspondiente y la anotamos → ejemplo: 'cGFnxxXcAghjdkd'
- editamos el archivo *encripta\_devuelve\_ALU.php* y cargamos el valor obtenido en el return → 'EMAIL\_PASSWORD' => 'cGFnxxXcAghjdkd'
- *correo\_test\_encriptado\_ALU.php*

Veremos muchas cosas:

.....

```
2025-01-16 17:43:31    SERVER -> CLIENT: 221 2.0.0 closing connection
Correo enviado exitosamente
```