



## UD 4.3 Envío de correos electrónicos con PHP y PHPMailer

DESARROLLO WEB EN ENTORNO SERVIDOR

CFGs DAW

Daniel Álvarez Peral

2024/2025

El envío de correos electrónicos es una funcionalidad muy común en las aplicaciones web. Ya sea para enviar confirmaciones de registro, notificaciones o boletines informativos, contar con una forma sencilla y eficiente de enviar correos electrónicos es esencial para cualquier desarrollador web.

Veamos cómo utilizar PHP y **PHPMailer** para enviar correos electrónicos desde tu aplicación web. **PHPMailer** es una biblioteca de código abierto que simplifica el proceso de envío de correos electrónicos utilizando PHP.

## Instalar PHPMailer en tu proyecto PHP

Para enviar correos electrónicos desde un proyecto PHP, necesitarás utilizar una biblioteca externa llamada **PHPMailer**.

### 1. Descargar PHPMailer

- Ve al repositorio oficial de **PHPMailer** en **GitHub**: [PHPMailer en GitHub](#).
- Haz clic en **Code** y selecciona **Download ZIP** para obtener la última versión de **PHPMailer**.

### 2. Extraer los archivos

- Una vez descargado el archivo **ZIP**, descomprímelo.
- Dentro de la carpeta descomprimida, encontrarás una carpeta llamada **src**, que contiene los archivos principales de **PHPMailer**.

### 3. Copiar los archivos a tu proyecto

- Copia la carpeta **src** a la carpeta de tu proyecto en **XAMPP**. Puedes ponerla en cualquier lugar de tu proyecto, pero una ubicación común sería dentro de la carpeta **includes** o **libs**, por ejemplo: **C:\xampp\htdocs\mi\_proyecto\libs\phpmailer**.

### 4. Incluir los archivos necesarios en tu proyecto

- En el archivo PHP donde quieras usar **PHPMailer**, debes incluir los archivos de **clase** necesarios. Para hacer esto, usa el siguiente código:

```
require '../envioEmail/phpmailer/src/PHPMailer.php';  
require '../envioEmail/phpmailer/src/SMTP.php';  
require '../envioEmail/phpmailer/src/Exception.php';
```

Asegúrate de que las rutas sean correctas según donde hayas colocado la carpeta **src** en tu proyecto.

Antes de enviar un correo electrónico, necesitarás configurar la conexión **SMTP**. **PHPMailer** te permite enviar correos electrónicos a través de un servidor **SMTP**. Puedes utilizar el servidor **SMTP** de tu proveedor de correo electrónico o configurar uno propio.

## 5. Configuración básica de PHPMailer

Ahora que has incluido **PHPMailer** en tu proyecto, puedes empezar a configurarlo y usarlo para enviar correos electrónicos. A continuación, veremos un ejemplo básico de cómo hacerlo:

```
// Incluye las clases necesarias de PHPMailer para el envío de correos y el manejo de
// excepciones
require '../envioEmail/phpmailer/src/PHPMailer.php'; // Clase principal de PHPMailer
require '../envioEmail/phpmailer/src/SMTP.php'; // Clase para manejar la conexión SMTP
require '../envioEmail/phpmailer/src/Exception.php'; // Clase para manejar errores de
// PHPMailer

use PHPMailer\PHPMailer\PHPMailer; // Importa la clase PHPMailer
use PHPMailer\PHPMailer\Exception; // Importa la clase Exception para manejo de errores

// Crear una instancia de PHPMailer
$mail = new PHPMailer(true);

try {
    // Configuración del servidor SMTP
    $mail->isSMTP(); // Usar SMTP
    $mail->Host = 'smtp.gmail.com'; // Servidor SMTP de Gmail
    $mail->SMTPAuth = true; // Autenticación SMTP
    $mail->Username = 'miCorreo@gmail.com'; // Tu correo de Gmail
    $mail->Password = '1234 5678 1234 abcd'; // Tu contraseña de Gmail
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS; // Cifrado TLS
    $mail->Port = 587; // Puerto SMTP de Gmail

    // Destinatarios
    $mail->setFrom('miCorreo@gmail.com', 'Tu Nombre');
    $mail->addAddress('destino@hotmail.com', 'Destinatario'); // El correo de destino

    // Contenido del mensaje
    $mail->isHTML(true); // Correo en formato HTML
    $mail->Subject = $asunto;
    $mail->Body = "Mensaje de: <b>$nombre</b><br><br>$mensaje";
    $mail->AltBody = $mensaje;

    // Enviar el correo
    $mail->send();
    echo 'Correo enviado con éxito';
} catch (Exception $e) {
    echo "El correo no pudo enviarse. Error de Mailer: {$mail->ErrorInfo}";
}
}
```

En el ejemplo anterior, hemos configurado los datos de conexión **SMTP** para utilizar el servidor de correo saliente de **Gmail**. Sin embargo, estos datos pueden variar dependiendo del proveedor de correo que utilicemos. Es importante consultar la documentación proporcionada por el proveedor para obtener los datos correctos.

Una vez configurados los datos de conexión **SMTP**, podemos utilizar los métodos y propiedades de la clase **PHPMailer** para personalizar el correo electrónico antes de enviarlo. Podemos

establecer el remitente, el destinatario, el asunto, el cuerpo del mensaje, adjuntar archivos, entre otras opciones.

## Gmail

El error **SMTP Error: Could not authenticate** suele ocurrir cuando hay un problema con las credenciales de acceso a tu cuenta de Gmail. Esto puede deberse a varias razones, pero aquí te indico algunos pasos y soluciones comunes para resolverlo:

### 1. Verificación en dos pasos (2FA) y contraseña de aplicación:

Si tienes habilitada la verificación en dos pasos (autenticación de dos factores) en tu cuenta de Google, necesitarás generar una contraseña de aplicación en lugar de usar tu contraseña normal.

Pasos para generar una contraseña de aplicación:

- Accede a tu cuenta de Google: Cuenta de Google.
- Ve a la sección Seguridad.
- En Iniciar sesión en Google, asegúrate de tener activada la Verificación en dos pasos.
- Después, en la misma sección de seguridad, busca Contraseñas de aplicación.
- Selecciona el tipo de aplicación que estás utilizando (por ejemplo, "Correo" y "Windows").
- Google te proporcionará una contraseña de aplicación de 16 caracteres que debes usar en el código en lugar de tu contraseña habitual.

<https://myaccount.google.com/apppasswords>

**Ejercicio crear un formulario para enviar correos**

## Ejecución Automática de Scripts PHP con el Programador de Tareas de Windows

Vamos a configurar tareas programadas para ejecutar un script **PHP** de manera automática. Usaremos el **Programador de Tareas de Windows** para automatizar tareas como el envío de correos electrónicos, procesamiento de datos, o ejecución de scripts PHP en intervalos regulares.

**Ejemplo:** Enviar correos electrónicos automáticamente con PHP

Imaginemos que deseas enviar un correo electrónico automáticamente todos los días a las 9 AM. Para ello, utilizaremos PHP con la librería **PHPMailer**.

## ¿Cómo configurar un Programador de Tareas en Windows?

### Paso 1: Crear el Script PHP

Asegúrate de tener el archivo **PHP** listo, por ejemplo **enviar\_correo.php**, y que funcione correctamente en tu servidor local o entorno de desarrollo.

### Paso 2: Verificar la Ruta del Ejecutable de PHP

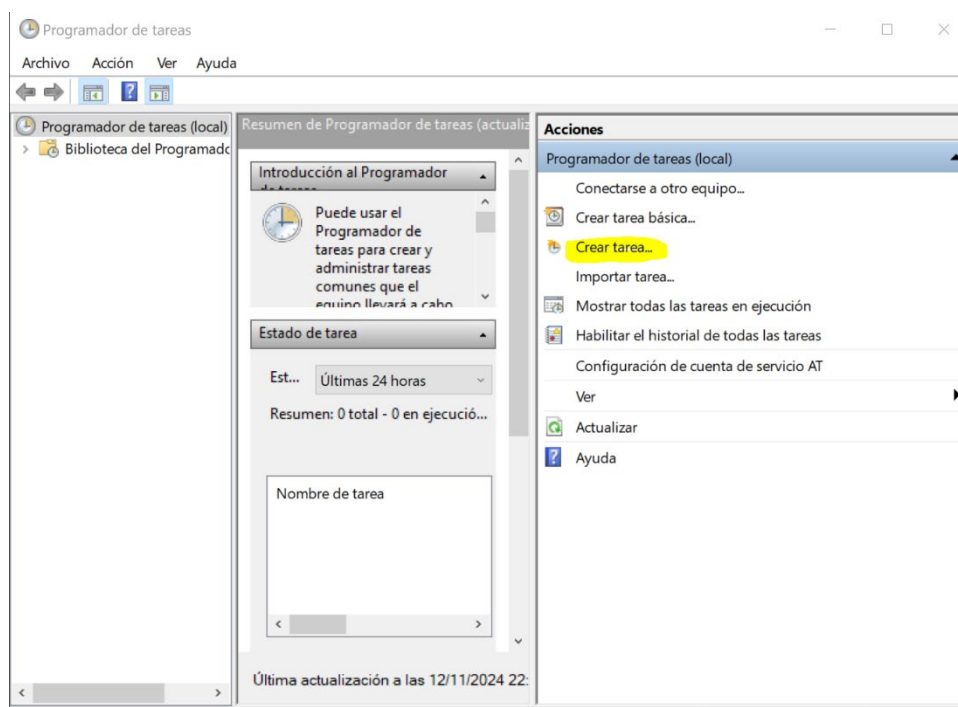
Si estás usando XAMPP o WAMP, localiza el archivo **php.exe**. Normalmente se encuentra en la siguiente ruta:

**XAMPP:** C:\xampp\php\php.exe

### Paso 3: Abrir el Programador de Tareas

Ve a Panel de Control > Herramientas Administrativas > Programador de Tareas o busca "**Task Scheduler**" en el menú de inicio.

Haz clic en Crear tarea en la parte derecha.



#### Paso 4: Configurar el Trigger (Desencadenador)

En la pestaña Desencadenadores, haz clic en Nuevo.... Configura la tarea para que se ejecute todos los días a las 9:00 AM, o el intervalo que prefieras.

#### Paso 5: Configurar la Acción

En la pestaña Acciones, haz clic en Nuevo.... En Acción, selecciona Iniciar un programa. Luego:

En Programa/script, escribe la ruta completa del ejecutable PHP:

C:\xampp\php\php.exe

En **Agregar argumentos (opcional)**, escribe la ruta completa de tu archivo PHP:

"C:\ruta\a\mi\script\enviar\_correo.php"

En **Iniciar en (opcional)**: escribe la carpeta de trabajo desde la cual se ejecutará el programa o script.

"C:\ruta\a\mi\script"

#### Paso 6: Configurar Condiciones (Opcional)

En la pestaña **Condiciones**, puedes seleccionar si la tarea solo debe ejecutarse cuando el equipo esté inactivo, o si debe ejecutarse, aunque no haya usuarios conectados.

#### Paso 7: Configurar la Configuración (Opcional)

En la pestaña **Configuración**, puedes elegir permitir que la tarea se ejecute incluso si falla o detenerla después de un tiempo específico.

#### Paso 8: Guardar la Tarea

Haz clic en **Aceptar** para guardar la tarea.

### Paso 9: Verificar la Ejecución

Para verificar que la tarea se ejecutó correctamente, puedes agregar una línea de código en el script **PHP** para registrar la ejecución en un archivo de log:

```
file_put_contents('log.txt', "El script fue ejecutado a: " . date('Y-m-d H:i:s') . "\n",  
FILE_APPEND);
```

Si usas una ruta absoluta como C:\logs\log.txt, el log se guardará en esa ubicación.

Si no especificas una ruta completa, el archivo de log se guardará en el directorio donde se ejecuta el script PHP.