

1.- El objetivo es crear un formulario de contacto que nos envíe un correo electrónico.

Los pasos que seguiremos son:

- Crearemos la siguiente estructura de carpetas:
- Hoja03_composer_02
 - public
 - Index.php
 - Procesar.php
 - app
 - Clases
 - ServicioCorreo.php
 - ProveedorMailtrap.php
 - Interfaces
 - InterfazProveedorCorreo.php
- Instalar PHPMailer y desarrollar una interfaz de correos
 - La dependencia la instalaremos con Composer:
 - Composer require phpmailer/phpmailer
 - También utilizaremos los espacios de nombre para las clases e interfaces
 - Realizaremos una interfaz para que independiente del proveedor del servicio nos funcione:
 - Creamos una clase InterfazProveedorCorreo.php
 - La interfaz tendrá un método enviarCorreo con tres parámetros de entrada de tipo string y devolverá un booleano (si ha enviado o no):
 - paraQuien
 - asunto
 - cuerpoMensaje
 - Desarrollamos el servicio de envío de correos permitiendo diferentes proveedores fichero ServicioCorreo.php
 - Definimos la clase ServicioCorreo:
 - Constructor, pasamos como parámetro el interfaz creado como privado y de solo lectura
 - Y el método enviarCorreo
 - Desarrollamos la clase concreta del proveedor de servicio en este caso para las pruebas se propone utilizar el proveedor de servicios Mailtrap
 - Crear cuenta en mailtrap <https://mailtrap.io/> utilizando el correo de Eduacantabria. La ventaja es que los correos no se envían al destinatario final se envía a la bandeja de entrada de mailtrap y nos sirve para realizar las pruebas.
 - Tenemos un apartado “Email Testing” con una bandeja de entrada y donde podemos recuperar las setting de SMTP para PHPmailer

Hoja03_COMPOSER_02

My Inbox

Total messages sent:

SMTP Settings	Email Address	Auto Forward	Manual Forward	Access Rights
Port: 25 or 465 or 587 or 2525 Username: 02cf71a3731489 Password: 6b02661606bfaa Auth: PLAIN, LOGIN and CRAM-MD5 TLS: Optional (STARTTLS on all ports)				Port: 1100 or 9950 Username: 02cf71a3731489 Password: 6b02661606bfaa Auth: USER/PASS, PLAIN, LOGIN, APOP and CRAM-MD5 TLS: Optional (STARTTLS on all ports)

Integrations ⓘ

PHPMailer
 ⌵

Use the following setting to configure PHPMailer:

```
$phpmailer = new PHPMailer();
$phpmailer->isSMTP();
$phpmailer->Host = 'sandbox.smtp.mailtrap.io';
$phpmailer->SMTPAuth = true;
$phpmailer->Port = 2525;
$phpmailer->Username = '02cf71a3731489';
$phpmailer->Password = '*****bfaa';
```

Copy

- Nos ofrece toda la configuración que vamos a necesitar
- Crearemos nuestro proveedor de servicio, creando el fichero ProveedorMailtrap.php, con la clase ProveedorMailtrap utiliza espacio de nombres que implemente la interfaz definida, observa antes el fichero adjunto a la tarea de enviar_correo.php
 - Define los espacios de nombres PHPMailer, SMTP y Exception, para utilizar las clases
 - Requiere el fichero de la interfaz
 - Definimos la clase implementando la interfaz
 - Y desarrollamos el método enviarCorreo
- Para probarlo creamos un formulario denominado index.php que nos pida los siguientes datos:
 - Nombre
 - Correo electrónico, el tipo de input vamos a poner que sea texto y a través de una función validaremos que sea correcto.
 - Mensaje
 - Al formulario a través de GET, le pueden llegar los siguientes atributos:
 - Si error=1 – debemos mostrar el mensaje” Por favor, rellena todos los campos.”
 - Si error=2 - debemos mostrar el mensaje” Por favor, introduce un email válido.”
 - Si error=3 - debemos mostrar el mensaje” Ha ocurrido un error al enviar el email.”
 - Si success - debemos mostrar el mensaje” El email se ha enviado correctamente.”
- El formulario una vez enviado ejecutará el fichero procesar.php
- El fichero procesar.php, es donde se realizará las siguientes validaciones:

Hoja03_COMPOSER_02

-
- Utilizando el autoloader de composer en PSR4 ServicioCorreo.php y ProveedorMailtrap.php
 - Comprobaremos que los tres campos no estén vacíos, si no lo cumple enviaremos al formulario a través de GET el error =1
 - Ejemplo: header('Location: index.php?error=1')
 - Comprobaremos a través de la función filter_var que el correo electrónico es válido, en caso contrario enviar al formulario por GET error=2
 - Crearemos una instancia de ServicioCorreo a la que pasamos por parámetro una instancia de ProveedorMailtrap
 - Llamamos al método enviarCorreo pasando los parámetros de para quién, asunto y el mensaje
 - Comprobamos si se ha podido enviar en cuyo caso realizamos un get con success=1 y salimos
 - Realizamos una redirección con error=3

Nota : se adjunta a la actividad un ejemplo “enviar_correo.php” de envío de correo PHPMailer y Mailtrap