

Actividad 3.1

Módulo	Desarrollo web en entorno servidor
Nombre y Apellidos ¹	Igor Karabinovskyy Kulyna
La entrega de esta tarea deberá efectuarse en PDF. La redacción del documento con faltas de ortografía podrá ser motivo de penalización.	

PRÁCTICA PARA RESOLVER:

Observe la siguiente imagen, debe realizarla en el formato indicado con PHP + XAMPP. Por pantalla deberá salir un Formulario y al pulsar envío recibiremos el mensaje.

Este es el Formulario para DAW2:

INTRODUZCA SU NOMBRE

DIRECCIÓN DE CORREO

COMENTARIOS

Mensaje

Enviar

Limpiar

Welcome david
Your email address is: xxxxxxxxxxxxxxxx@gmail.com

¹ En el caso de trabajo en grupo Apellidos y nombre de todos los componentes.

TEORIA PARA RESOLVER LA ACTIVIDAD

PHP: un formulario HTML simple

El siguiente ejemplo muestra un formulario HTML simple con dos campos de entrada y un botón de envío:

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

Cuando el usuario completa el formulario anterior y hace clic en el botón Enviar, los datos del formulario se envían para su procesamiento a un archivo PHP llamado "welcome.php". Los datos del formulario se envían con el método HTTP POST.

```
<html>
<body>

Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>

</body>
</html>
```

La salida podría ser algo como esto:

Welcome david
Your email address is: XXXXXXXXXXXX@gmail.com

El mismo resultado también podría lograrse utilizando el método HTTP GET:

```
<html>
<body>

<form action="welcome_get.php" method="get">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>
```

y "welcome_get.php" se ve así:

```
<html>  
<body>
```

```
Welcome <?php echo $_GET["name"]; ?><br>  
Your email address is: <?php echo $_GET["email"]; ?>
```

```
</body>  
</html>
```

El código anterior es bastante simple. Sin embargo, falta lo más importante. Debe validar los datos del formulario para proteger su secuencia de comandos del código malicioso.

¡Piense en **SEGURIDAD** al procesar formularios PHP!

Esta página no contiene ninguna validación de formulario, solo muestra cómo puede enviar y recuperar datos de formulario.

Sin embargo, las próximas páginas mostrarán cómo procesar formularios PHP teniendo en cuenta la seguridad. ¡La validación adecuada de los datos del formulario es importante para proteger su formulario de piratas informáticos y spammers!

¿Cuándo usar **GET**?

La información enviada desde un formulario con el método GET es visible para todos (todos los nombres y valores de las variables se muestran en la URL). GET también tiene límites en la cantidad de información a enviar. La limitación es de unos 2000 caracteres. Sin embargo, debido a que las variables se muestran en la URL, es posible marcar la página. Esto puede ser útil en algunos casos.

GET puede usarse para enviar datos no confidenciales.

Nota: NUNCA se debe usar GET para enviar contraseñas u otra información confidencial.

¿Cuándo usar **POST**?

La información enviada desde un formulario con el método POST es invisible para los demás (todos los nombres/valores están incrustados en el cuerpo de la solicitud HTTP) y no tiene límites en la cantidad de información a enviar.

Además, POST admite funciones avanzadas, como compatibilidad con entrada binaria de varias partes al cargar archivos en el servidor.



Entrega de Actividad

Sin embargo, debido a que las variables no se muestran en la URL, no es posible marcar la página.

Los desarrolladores prefieren POST para enviar datos de formulario.