Bibliografía:

www.w3schools.com

https://www.php.net/

SEGURIDAD EN FORMULARIOS HTML CON PHP

Nunca usar GET para enviar contraseñas o cualquier otra información confidencial. Además este método tiene límites en la cantidad de datos a enviar.

Usar POST ya que los datos no son visibles, no hay límite en cuanto al número de datos y también admite información en formato binario.

Recoger los datos con $_POST y no con $_REQUEST, para evitar recoger código que nos inyecten en la url.

Existe un array asociativo global $_SERVER, que tiene información que nos puede ser de utilidad:

$_SERVER[“PHP_SELF”] : Guarda el nombre del archivo que se está ejecutando, lo podemos utilizar para que un formulario se autoenvíe los datos.

Hay que tener cuidado con esa variable, ya que si no lo evitamos nos pueden hackear a través de una inyection en esa variable.

Ej. Seguridad1.php

Para evitarlo tenemos que usar la función **htmlspecialchars,** esta función convierte los caracteres especiales en entidades HTML. Esto significa que reemplazará caracteres HTML como < y > con < y >. Esto evita que los atacantes exploten el código inyectando código HTML o Javascript (Cross-site Scripting) en los formularios.

Probar como en el ejemplo anterior usando esa función impedimos el ataque.

En general, tenemos que limpiar todos los datos que nos lleguen desde un formulario.

Podemos incluir la siguiente función que limpia los datos:

```
function test_input($data) {
  $data = trim($data); // quita los espacios en blanco
  $data = stripslashes($data); // quita los slash \
  $data = htmlspecialchars($data);
  return $data;
}
```

Ej. Seguridad2.php

INYECCIONES SQL

La inyección SQL es la colocación de código malicioso en las instrucciones SQL, a través de un formulario de una página web.

Inyección SQL (w3schools.com)

Para evitarlo:

- Limpiar los datos de entrada como hemos indicado antes.
- Usar prepared Staments en las instrucciones MySQL.(No lo vemos) Ejemplo
  PHP MYSQLi Procedural Prepared Statements for beginners - WDB24

```php
<?php
$id = 93;
$firstName = "Conner";
$lastName = "Krajcik";

$qry = 'select * from customers where id = ? and first_name = ? and last_name = ? ';

$userStatement = mysqli_prepare($conn, $qry);

mysqli_stmt_bind_param($userStatement, 'iss',$id,$firstName,$lastName);

mysqli_stmt_execute($userStatement);

$result = mysqli_stmt_get_result($userStatement);
$getData = mysqli_fetch_assoc($result);

print_r($getData);
?>
```