EJERCICIOS DE FORMULARIOS CON PHP

EJERCICIO 1: Prueba de formularios html

Primera parte:

Escribe un programa PHP llamado **prueba_formularios.php**, que muestre los parámetros enviados por un request, tanto si son GET como si son POST.

- Si se ha enviado el Request con GET, se mostrará una lista con todos los parámetros enviados:

- En el caso de que se haya enviado con POST, se mostrará lo mismo, cambiando el título a "Se recibe request con POST".
- Si no se han enviado parámetros, mostrará el mensaje

  "No se ha enviado información"

Segunda parte:

Para probar tu programa, elabora un formulario en HTML, llamado **formulario.html,** indicando en el action el enlace al programa prueba_formularios.php

a. Prueba el formulario enviando información con GET
b. Prueba el formulario enviando información con POST
c. Accede directamente a prueba_formularios.php sin parámetros

EJERCICIO 2: Cálculo de hash salado

Vas a realizar un programa PHP que calcule el "**salted hash**" de una password (Ver apartado de definición de salted hash para más información).

Especificaciones:

El programa mostrará un formulario con dos campos:

1. **Campo password**: Será tipo **texto** y es la password a codificar. En *servidor* validarás que la password tiene al menos 8 caracteres, y que contiene al menos un número, una letra minúscula, una letra mayúscula, y un carácter especial entre los siguientes: "_", "-", "."
2. **Campo sal:** Será la sal que se añadirá a la password. No se podrá modificar. Por defecto (la primera vez), aparecerá un número aleatorio entre 10000 y 99999 (un número de cinco cifras).
3. **Un botón de submit**

Cuando el usuario pulse el botón submit, se enviarán los datos del formulario: password y sal.

El servidor validará que la password es correcta, si no lo es mostrará el mensaje: " La password debe tener al menos 8 caracteres, y contener alguna minúscula, alguna mayúscula, algún número, y algún carácter especial entre '_', '-', o '. '"

Si la password no es válida, el programa sólo mostrará el mensaje de error.

Si la password es válida, el programa concatenará la sal con la password (sal+password) y calculará el código hash con el algoritmo sha256. Luego mostrará el mensaje:

"El sha256 de la password XXX salada con XXXXX es: XXXXXXX" debajo del formulario, que mantendrá la persistencia tanto del campo "password" como del campo "sal".

Requisitos:

- El código del **CONTROLADOR** estará separado del código **VISTA**
- **Persistencia** de la información, con **desinfección** para prevenir inyección de malware
- Programar la **validación** de la password en una función. Puedes usar **preg_match**.

```
function passOK(string $password): bool
```

Uso de preg_match:

La función preg_match de php comprueba si una cadena cumple con subcadenas que se adapten a una expresión regular. Retornará 1 si encuentra al menos una subcadena que cumpla la expresión, 0 si no hay ninguna, y false si hubo un error.

Recuerda, la expresión regular debe ir entre los símbolos "/" y "/".

Ejemplo: Contiene una subcadena que empiece por "a" seguida de números

preg_match("/a[0-9]*/","k333a222") => retorna true porque "a222" lo cumple

Definición de "salted hash":

Como sabes, un código hash es la codificación de una password mediante algoritmos indescifrables. Se guarda el código en vez de la password, y así es más segura.

Con el tiempo, y por seguridad, se mejoró esta codificación concatenando una cadena aleatoria (sal) a la password, antes de codificarla: Se concatena esta "sal" con la password, y luego se codifica. Lo codificado no es el hash de la password, sino el hash de la password "salada", una cadena más larga.

Codificación	Password	Código guardado
hash sha256	1234	`hash('sha256','1234')`
"salted" hash sha256	1234	`hash ('sha256', $cadenaAleatoria . '1234')`