

Estructura y repaso del examen

Los formularios de contacto

En el caso de prueba.php añadiemos un HTML dentro de un PHP.
Necesitamos XAMPP con servidor apache. Tengamos PHP corriendo y que funciona.

En el método GET aparecería arriba los métodos que habíamos añadido.

Se vieron las diferencias entre GET y POST

Podíamos guardarlas en variable globales como GET y POST.

El **action** es a donde voy. Si no pongo nada, lo enviará a el mismo. Por eso en el GET se pone el mismo. Si fuera POST, no saldría arriba, pero quedaría también guardado en si mismo.

Cuando lo enviamos por POST la variable que está llena es la de POST. Lo que estamos haciendo.

- ➔ En la clave array teníamos el valor que ha gente había introducido en el Input.

Filtros para formularios

Cuando teníamos un formulario **también poníamos filtro**. El principal filtro es en el action.

```
<form action="<? Php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>"
```

- ➔ El PHP self es la forma más segura de decirle que vaya a este archivo.
- ➔ No es una constante real pero guarda el PHP_SELF guardaría que está donde está. Y se lo da la variable \$_SERVER.

Una de las cosas que vimos es htmlspecialchars, también el trim, y luego vimos el FILTER_SANITIZE_STRING. Esto ya no sirve. Necesitamos un filtro a la hora de recibir los datos. En el nombre y en el email, hay dos en concretos.

- ➔ FILTER_SANITIZE_EMAIL
- ➔ FILTER_VALIDATE_EMAIL

Uno lo sanea y otro lo filtra para validarlo. Validar envío es si se ha hecho por alguno de los dos métodos. (arxiu valida envio)

1. Revisar la forma en la que se han enviado los datos
\$_SERVER['REQUEST.....

Finalmente se hizo una validación general.

Generación de mensaje

Se hizo que salieran mensajes cuando se intentó enviar. Posteriormente conectaremos el aplicativo con la base de datos. Poca cosa más. Da igual que no tengas un css, le puedes poner el mismo que generamos nosotros.

Las sessions

Se empezaban con un sesión_start

Archivo 13.index, todo iba a parar a la variable global \$_SESSION

Si no había session destroy, esa sesión se mantenía y no finalizaba. [array] → Había un array con los datos que se le daban. Si nosotros utilizamos la sesión start en otro sitio. Cuando se clicca cerrar sesión y no se actualiza la pestaña es que se mantiene la sesión.

La diferencia fundamental con las cookies es que las cookies se guardan en el navegador. Y la sesión se guarda en la variable durante el uso.

Cookies

Hay que utilizar el set cookie.

- ➔ Se le pone un nombre
- ➔ Se le pone un valor
- ➔ Se le pone una expiración
 - Una fecha negativa o pasada eliminaba la vigencia de esa cookie.
- ➔ Se le pone el dominio. Se le pone un path: La ruta dentro del servidor en la que la cookie estará disponible.
 - Si poníamos / se refería a todo el dominio.
 - El domain podría ser la explicitación de ese dominio.

Bases de datos

Entender principalmente que es: un conjunto de datos relacionados.

- ➔ Álgebra relacional: página 38.

Tienen que ser compatibles. Mismo valor. Los mismos campos. Similares.

- Unión
 - Mismo grado (mismo número de campos).
 - Mismos valores, similares.
- Intersección
 - Son comunes en ambas tablas. Lo que es común en ambas.
- Diferencia.
 - Solamente los que coinciden son los que se restan.
- ➔ Normalización: página 75.
 - Primera forma normal: Establece que no puede haber más de un valor en un registro.
 - Segunda forma normal: si tenemos claves primarias juntas que cualquiera podía ser única, hay que separarla. Siempre tendrán

relación directa con una de las claves primarias que ya existen en esa tabla. Se repetirá porque tiene diferentes cursos pero no se repetirá igual.

- Tercera firma: Cuando tenemos una relación de otra relación, también tiene que ir fuera. El código de provincia y provincia siempre va a ir juntos.
- Boyce Scott.: Podías tener diferentes Combinaciones. Establece más divisiones.
 - Hay que estrechar al máximo. No hay que repetir lo que es otra relación.

➔ No van a entrar los modelos avanzados.