

Formularios:

1. Crea un formulario para enviar los datos de registro de un libro: título, autor, editorial y páginas.
2. Crea un formulario para enviar el campo nombre.
 - a) Si el nombre existe se da un saludo.
 - b) Si no existe se vuelve atrás indicando que el campo es obligatorio.
3. Envío del script al mismo script. Crea un formulario para enviar el campo nombre.
 - a) El nombre debe existir y tener una longitud mínima de 3 caracteres.
 - b) Si es válido se da un saludo.
 - c) Si no lo es vuelve a mostrar el formulario indicando que el campo es obligatorio y mostrando en el "input" el valor anterior no válido.
4. Envío del script al mismo script. Crea un formulario que funcione como una calculadora. Debe contener dos input como operandos y un select para elegir operador.
 - a) Si se reciben los datos muestra el resultado.
 - b) Si no son válidos o no existen debe mostrar de nuevo el formulario de la calculadora.
5. Crea un formulario que envíe un array de 3 nombres. Para hacerlo debes usar el mismo nombre en todos los input (name = "nombres[]").
6. Realiza un minicuestionario con 5 preguntas tipo test sobre las asignaturas que se imparten en el curso. Cada pregunta acertada sumará un punto. El programa mostrará al final la calificación obtenida.

Aplicaciones Web:

7. Crea una aplicación web con una clase App y varios métodos.
 - a) En todos los casos se trata de obtener una serie numérica.
 - b) El método debe calcular la serie y guardarla en un array, después hay que incluir una vista que muestre la serie.
 - c) Puede ser que necesites crear métodos auxiliares (private) para el cálculo del array, por ejemplo: esPrimo().
 - d) Los métodos necesarios son:
 - i. Index (**index**). Presentación de la App y enlaces.
 - ii. Fibonacci (**fibonacci**). Muestra la serie de Fibonacci. Debe mostrar todos los términos menores a un millón.
 - iii. Potencias de 2 (**potencias2**). Debe mostrar los valores de las potencias de 2 hasta 2 elevado a 24.
 - iv. Factorial (**factoriales**). Debe mostrar los factoriales desde 1 hasta n de tal manera que el último término sea el más próximo cercano al millón.
 - v. N°. primos (**primos**). Debe encontrar los números primos entre 1 y 10.000.

Cookies:

8. Vamos a crear una clase App con estos métodos:
 - a) login: muestra un formulario de login (usuario y contraseña).
 - b) auth: guarda el usuario y su contraseña en una cookie. Después reenvía la petición a home.
 - c) home: Muestra un saludo y un enlace para cerrar sesión.
 - d) logout: elimina las cookies (setcookie(..., time() - 1)) y reenvía a login.
 - e) Depura tu código. En login, comprueba que no hay ya un usuario. Si lo hay reenvía a home.

9. Se trata de crear una lista de deseos. Usaremos la clase App con los siguientes métodos:
- login** método que muestra el formulario de entrada.
 - auth** método que toma el nombre de usuario tras el login.
 - Tras hacer esto reenvía a **home**.
 - home** método que muestra la lista de deseos. Además muestra un formulario de nuevos deseos.
 - El formulario envía al método **new**.
 - new** toma el nuevo deseo y lo incluye en la lista.
 - delete** borra un deseo de la lista. Debe recibir el índice del deseo.
 - empty** vacía la lista de deseos.
 - close** cierra sesión: borra la cookie.
10. Colorear una página con ayuda de una cookie. Usaremos la clase App con los siguientes métodos:
- home** muestra un mensaje de bienvenida.
 - Comprueba si hay una cookie llamada "color".
 - Si existe la usa para darle color al fondo de la página.
 - colores** muestra una lista de colores. Cada color tiene un enlace del estilo **?method=cambio&color=red**. Al hacer clic cambiará el color del home.
 - cambio** recibe el color de la página anterior, crea la cookie y reenvía ('header...') al método home.

NOTA: dos vistas (home y colores) y un reenvío (cambio).

Sesiones:

11. Repite el ejercicio 9 usando sesiones en lugar de cookies.

12. Repite el ejercicio 10 usando sesiones en vez de cookies.

Ficheros:

13. Crea una aplicación para almacenar una lista de deseos. Lista de métodos:
- home**, método por defecto.
 - Muestra la lista de deseos y un formulario para añadir nuevos.
 - Index**, lee el contenido de la lista de ficheros
 - new**, toma la información del nuevo deseo y lo añade al fichero. Después reenvía a home.
 - clear**, borra toda la lista de deseos y vuelve al home.
 - delete**, borra selectivamente elementos de la lista de deseos.
 - NOTA: debes, cargar el array, borrar el elemento y volver a escribir el fichero completo.

HERRAMIENTAS DE DESARROLLADOR

1) Ver las cookies

Opciones (F12) -> Storage:

Name	Value	Domain	Path	Expires / Max-Age	Size
visitas	6	localhost	/dwes/UD3/ejemplos/6_Co...	Tue, 05 Oct 2021 09:23:00 GMT	8
visitas	4	localhost	/dwes/UD3/ejemplos/9_Co...	Tue, 19 Oct 2021 09:39:10 GMT	8

2) Depuración de errores

- Instalar Xdebug.

- sudo apt-get install php7.4-xdebug
- sudo apt-get update
- sudo apt-get install php-dev autoconf automake

- Añade la siguiente línea al fichero php.ini


```
sudo nano /etc/php/7.4/apache2/php.ini
```

```
zend_extension=xdebug
xdebug.mode = debug
xdebug.start_with_request = yes
xdebug.client_port = 9000
xdebug.remote_enable = 1
xdebug.remote_autostart = 1
```

- Reiniciar apache

```
sudo apache2 restart
```

- Dentro de VSCODE

- Abrir el debugger: 
- Configurar un entorno para PHP

Run -> Open Configuration -> PHP

```
"version": "0.2.0",
"configurations": [
```

```
    {
        "name": "Listen for XDebug",
        "type": "php",
        "request": "launch",
        "port": 9000
```

```
    },
    {
        "name": "Launch currently open script",
        "type": "php",
        "request": "launch",
        "program": "${file}",
        "cwd": "${fileDirname}",
        ...
    }
]
```