

EJERCICIOS NUEVOS (nivel creciente)

Parte 1 – Generador de colores aleatorios (color.php)

Objetivo: practicar `GET`, funciones aleatorias y actualización de página.

Flujo:

1. Muestra un botón que diga “**Generar color**”.
2. Al pulsarlo, recarga la misma página enviando una variable `color` con un valor aleatorio (por ejemplo, `#A1B2C3`).
3. Muestra un div con fondo del color generado y el código del color en texto.
4. Si no se ha generado ninguno todavía, muestra un mensaje tipo “Pulsa el botón para generar un color”.

Pistas:

```
$color = sprintf('#%06X', rand(0, 0xFFFFFF));
```

Extra: Guarda un histórico de los últimos 5 colores y muéstralos como cuadrados pequeños debajo.

Parte 2 – Calculadora de edad canina (edadCanina.php)

Objetivo: practicar formulario + PHP con `GET`.

1. Muestra un formulario con un input de texto para el nombre del perro y otro para su edad humana.
 2. Al enviar, calcula la edad “canina” (por ejemplo, edad humana \times 7).
 3. Muestra: “Tu perro [nombre] tiene [resultado] años caninos 🐶”.
 4. El formulario sigue visible para calcular otro.
-

Parte 3 – Generador de excusas (excusas.php)

Inspirado en el de frases cuñadas.

1. Tres `<select>`:
 - Sujeto: “Mi gato”, “El jefe”, “Mi vecino”, “El profe de DAW”, “Mi ordenador”
 - Acción: “se comió”, “perdió”, “rompió”, “olvidó”, “borró”
 - Objeto: “mi trabajo”, “las llaves”, “la base de datos”, “los apuntes”, “el servidor”
2. Al enviar, PHP genera una excusa:
“Mi gato rompió el servidor.”
3. Muestra la excusa y el formulario otra vez.

Extra JS: botón “Aleatorio” que rellene aleatoriamente los selects antes de enviar.

Parte 4 – Contador de visitas (contador.php)

Objetivo: usar GET + variables + cookies o parámetros simples.

1. Cada vez que visitas `contador.php`, aumenta un contador (`?visitas=3`, `?visitas=4`, etc.).
 2. Muestra: “Has visitado esta página X veces.”
 3. Si no hay parámetro, empieza en 1.
 4. Opción B: usa `setcookie("visitas", ...)` para que no se borre al cerrar.
-

Parte 5 – Calculadora de operaciones mixtas (mixcalc.php)

Objetivo: formularios, condicionales y operadores.

Formulario con:

- Dos inputs numéricos

- Un select con opciones:

- +
- -
- *
- /
- mayor
- menor
- igual

- Botón “Calcular”

PHP:

- Recoge los datos con `$_GET`.
- Usa `if / elseif` o `switch` para decidir la operación.
- Muestra el resultado en pantalla.

Ejemplo:

5 mayor 3 → Resultado: true

Parte 6 – Contador de palabras (contadorPalabras.php)

Objetivo: trabajar con strings y arrays (`explode`, `count`, `implode`).

1. Muestra un `<textarea>` para escribir texto.
2. Al enviar, el PHP:
 - Convierte el texto en array de palabras (`explode(" ", $texto)`).

- Muestra cuántas palabras hay.
- Muestra la frase al revés (`array_reverse()` + `implode()`).

Extra: elimina palabras repetidas (`array_unique()`).

Parte 7 – Trivia simple con JavaScript y PHP (trivia.php)

Objetivo: combinar lógica JS y PHP en un mismo flujo.

1. En HTML muestra una pregunta:
“¿Cuál es el lenguaje que se ejecuta en el servidor?”
Con tres radios: PHP, JavaScript, Python.
 2. Al hacer clic en “Comprobar”, un script JS:
 - Captura la respuesta marcada.
 - Muestra por JS si es correcta.
 - También envía por GET al PHP la respuesta seleccionada.
 3. PHP recibe la respuesta y guarda cuántos aciertos lleva (`?aciertos=2`).
-

Parte 8 – Selector de temas (tema.php)

Objetivo: combinar PHP y JS para personalizar estilos.**

1. Formulario con un `<select>`: “Claro”, “Oscuro”, “Arcoíris”.
 2. Al enviar (GET), el PHP cambia el color de fondo del `<body>` según el tema elegido.
 3. Usa un pequeño script JS para cambiar el tema sin recargar (añade una clase CSS al `body` dinámicamente).
-

Parte 9 – Sorteador de nombres (sorteo.php)

Objetivo: usar arrays dinámicos y `rand()`.**

1. Un `<textarea>` donde el usuario mete una lista de nombres (uno por línea).
 2. Al enviar, PHP convierte eso en array (`explode("\n", $texto)`).
 3. Elige un nombre al azar (`$ganador = $nombres[rand(0, count($nombres)-1)]`).
 4. Muestra el ganador en pantalla 🎉.
-

Parte 10 – Simulador de temperatura (temperatura.php)

Objetivo: combinar JS + PHP + DOM + GET.**

1. Input numérico (temperatura) y select con unidades: °C o °F.
2. Al enviar:
 - PHP convierte la temperatura a la otra unidad.
 - JS cambia el color del fondo según el resultado:
 - Azul si < 10
 - Naranja si 10–25
 - Rojo si > 25
3. Muestra el valor convertido y un mensaje:
“22°C equivalen a 71.6°F”.