

# Documentación: b2.php

## Autor: Héctor Arciniega Valencia

Nivel B – Ejercicio 2

**1 <?php**

- Rol: apertura del intérprete PHP; el servidor evalúa el bloque.

**2 /\*\***

- Rol: inicio de DocBlock; documentación para personas y herramientas.

**3 \* Nivel B – Ejercicio 2 (Ciclo)**

- Rol: comentario; sin efecto en tiempo de ejecución.

**4 \* Suma de múltiplos de 3 o 5 entre 1 y N (inclusive).**

- Rol: comentario; sin efecto en tiempo de ejecución.

**5 \***

- Rol: comentario; sin efecto en tiempo de ejecución.

**6 \* Entrada: N (entero  $\geq 1$ )**

- Rol: comentario; sin efecto en tiempo de ejecución.

**7 \* Salida: lista 1..N marcando múltiplos con '\*', y Suma final.**

- Rol: comentario; sin efecto en tiempo de ejecución.

**8 \*/**

- Rol: fin de DocBlock.

**9**

**10 declare(strict\_types=1);**

- Rol: directiva strict\_types.
- Efecto: activa tipado estricto en este archivo; prohíbe coerciones implícitas en firmas.

**11**

**12 function validar\_N(?string \$raw): array {**

- Rol: declaración de función.
- Nombre: validar\_N
- Parámetro \$raw: tipo declarado ?string.
- Retorno: array.
- Contrato: retorna array: {ok:bool, val?:T, msg?:string}.

**13 if (\$raw === null || \$raw === '') return ['ok'=>false, 'msg'=>'Dato vacío.'];**

- Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.
- Comparación con cadena vacía para validar presencia de datos.

**14 if (!preg\_match('/^\d+\$/ ', trim(\$raw))) return ['ok'=>false, 'msg'=>'Debe ser entero  $\geq 1$ .'];**

- Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.
- Función: preg\_match(patcón, sujeto, [coincidencias], [flags], [offset]). Devuelve 1/0 o false.
- Patcón: `/^\d+$/` → `^` ancla inicio; `\d` dígito [0-9]; `+` uno o más; `$` ancla fin. Permite '0', '10', '123'. Rechaza '-1', '1.2', '3', '3 '.
- Función: trim(cadena). Elimina espacios/blancos en extremos.

**15 \$val = (int)\$raw;**

**16 if (\$val < 1) return ['ok'=>false, 'msg'=>'Debe ser  $\geq 1$ .'];**

- Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.

**17 return ['ok'=>true, 'val'=>\$val];**

- Rol: retorno; finaliza la función devolviendo el valor indicado.

**18 }**

**19**

**20 \$N = null;**

- Rol: inicialización de variable a null.

**21 \$lista = [];**

- Rol: inicialización de arreglo vacío [].

**22 \$suma = 0;**

**23 \$error = null;**

- Rol: inicialización de variable a null.

**24**

**25 if (\$\_SERVER['REQUEST\_METHOD'] === 'POST') {**

- Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.

- Condición: \$\_SERVER['REQUEST\_METHOD'] === 'POST' (método HTTP).

**26 \$res = validar\_N(\$\_POST['N'] ?? null);**

**27 if (!\$res['ok']) {**

- Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.

**28 \$error = \$res['msg'];**

**29 } else {**

**30 \$N = (int)\$res['val'];**

**31 // Recorrer de 1 a N y acumular múltiplos de 3 o 5**

- Rol: comentario; sin efecto en tiempo de ejecución.

**32 for (\$i=1; \$i <= \$N; \$i++) {**

- Rol: bucle for (inicialización; condición; actualización).

**33 \$esMultiplo = (\$i % 3 === 0) || (\$i % 5 === 0);**

**34 \$lista[] = [\$i, \$esMultiplo];**

**35 if (\$esMultiplo) {**

- Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.

**36 \$suma += \$i;**

**37 }**

**38 }**

**39 }**

**40 }**

**41 ?>**

- Rol: cierre del bloque PHP; desde aquí se emite HTML sin evaluar.

**42 <!doctype html>**

- Rol: declaración HTML5 (modo estándar).

**43 <html lang="es">**

- Atributos: lang="es": idioma del documento.

**44 <head>**

**45 <meta charset="utf-8" />**

- Atributos: charset="utf-8": codificación de caracteres.

**46 <title>NB-B2 Suma de múltiplos 3 o 5</title>**

**47 <style>**

**48 body{font-family:system-ui, -apple-system, Segoe UI, Roboto, Arial, sans-serif; margin:24px}**

- CSS: font-family: pila de fuentes del sistema para legibilidad y rendimiento. margin: 24px; margen externo del elemento.

**49 fieldset{border:1px solid #ccc;padding:12px;border-radius:8px;max-width:520px}**

- CSS: border: 1px solid #ccc; estilo y color del borde. padding: 12px; relleno interno del elemento. border-radius: 8px; radios de las esquinas. max-width: 520px; ancho máximo para diseño fluido.

**50 label{display:block;margin:8px 0 4px}**

- CSS: display:block: ocupa toda la línea disponible. margin: 8px 0 4px; margen externo del elemento.

**51 input[type=number]{width:100%;padding:8px;border:1px solid #bbb;border-radius:6px}**

- CSS: width: 100%; ancho del control. padding: 8px; relleno interno del elemento. border: 1px solid #bbb; estilo y color del borde. border-radius: 6px; radios de las esquinas.

**52 button{padding:8px**

**14px;border:0;border-radius:8px;background:#07a;color:#fff;cursor:pointer;margin-top:10px}**

- CSS: padding: 8px 14px; relleno interno del elemento. border: 0; estilo y color del borde. border-radius: 8px; radios de las esquinas. background: #07a; color de fondo. color: #fff; color del texto. cursor:pointer: apariencia de mano al pasar el mouse.

**53 .error{color:#b00020;font-weight:700}**

- CSS: color: #b00020; color del texto.

**54 .result{margin-top:12px;padding:8px;border-left:4px solid #07a;background:#eef7ff}**

- CSS: padding: 8px; relleno interno del elemento. background: #eef7ff; color de fondo.

**55 .nums{display:flex;flex-wrap:wrap;gap:8px;margin-top:8px}**

- CSS: gap: 8px; separación entre elementos en grid/flex.

**56 .box{padding:4px 8px;border:1px solid #ddd;border-radius:6px}**

- CSS: padding: 4px 8px; relleno interno del elemento. border: 1px solid #ddd; estilo y color del borde. border-radius: 6px; radios de las esquinas.

**57 .hit{background:#fffbcc;border-color:#ffe58f}**

- CSS: background: #fffbcc; color de fondo.

**58 code{background:#f7f7f7;padding:2px 4px;border-radius:4px}**

- CSS: background: #f7f7f7; color de fondo. padding: 2px 4px; relleno interno del elemento. border-radius: 4px; radios de las esquinas.

**59 </style>**

**60 </head>**

**61 <body>**

**62 <h1>NB-B2 – Suma de múltiplos de 3 o 5</h1>**

**63 <form method="post" novalidate>**

- Atributos: novalidate: desactiva validación HTML5; validación queda del lado servidor.

**64 <fieldset>**

**65 <legend>Entrada</legend>**

**66 <label for="N">N (entero  $\geq 1$ )</label>**

- Atributos: for="N": asocia la etiqueta con el control id="N".

**67 <input type="number" id="N" name="N" min="1" step="1" value="<?php echo htmlspecialchars(\$\_POST['N'] ?? ""); ?>" />**

- Función: htmlspecialchars(texto[, ENT\_QUOTES, 'UTF-8']). Escapa &, <, > y comillas.

- Seguridad: usar ENT\_QUOTES y 'UTF-8' evita XSS por comillas simples y dobles.

- Atributos: type="number": tipo de control. id="N": identificador único en el DOM. name="N": clave enviada en la petición (GET). min="1": valor mínimo permitido por el navegador. step="1": incremento permitido (enteros). value="&lt;?php...&gt;": relleno dinámico; se escapa con htmlspecialchars para evitar XSS.

**68 <button type="submit">Procesar</button>**

- Atributos: type="submit": tipo de control.

**69 <?php if (\$error): ?><div class="error"><?php echo htmlspecialchars(\$error); ?></div><?php endif; ?>**

- Rol: apertura del intérprete PHP; el servidor evalúa el bloque.

- Función: htmlspecialchars(texto[, ENT\_QUOTES, 'UTF-8']). Escapa &, <, > y comillas.

- Seguridad: usar ENT\_QUOTES y 'UTF-8' evita XSS por comillas simples y dobles.

**70 </fieldset>**

**71 </form>**

**72**

**73 <?php if (\$N !== null): ?>**

- Rol: apertura del intérprete PHP; el servidor evalúa el bloque.

**74 <div class="result">**

**75 <strong>Salida:</strong>**

**76 <div class="nums">**

**77 <?php foreach (\$lista as [\$num, \$hit]): ?>**

- Rol: apertura del intérprete PHP; el servidor evalúa el bloque.

**78 <div class="box <?php echo \$hit ? 'hit' : ''; ?>">**

**79 <?php echo \$num; ?><?php echo \$hit ? '\*' : ''; ?>**

- Rol: apertura del intérprete PHP; el servidor evalúa el bloque.

**80 </div>**

**81 <?php endforeach; ?>**

- Rol: apertura del intérprete PHP; el servidor evalúa el bloque.

**82 </div>**

**83 <div style="margin-top:10px;">Suma = <code><?php echo \$suma; ?></code></div>**

**84 </div>**

**85 <?php endif; ?>**

- Rol: apertura del intérprete PHP; el servidor evalúa el bloque.

**86 </body>**

**87 </html>**