

1 <?php

Apertura del intérprete PHP. Indica que lo que sigue será procesado por el motor PHP.

2 /\*\*

Inicio de bloque de documentación (DocBlock).

3 \* Nivel B – Ejercicio 3 (Función)

Título del ejercicio.

4 \* ¿Número capicúa?

Descripción del objetivo: verificar si un número es palíndromo.

5 \*

Separador dentro del DocBlock.

6 \* Entrada: num (entero  $\geq 0$ )

Especifica el dato que debe ingresar el usuario.

7 \* Función: esCapicua(int \$n): bool

Firma esperada de la función. Entrada: entero. Salida: booleano.

8 \* Salida: "SI" si es capicúa, "NO" en caso contrario.

Define el formato de salida.

9 \*/

Cierra el DocBlock.

10 *(línea en blanco)*

Separador visual.

11 declare(strict\_types=1);

Habilita tipado estricto: los parámetros deben cumplir su tipo exacto, sin coerción automática.

12 *(línea en blanco)*

Separador visual.

13 function validar\_num(?string \$raw): array {

Función de validación.

Entrada: string|null (\$raw).

Salida: array con claves ok, msg o val.

14 if (\$raw === null || \$raw === '') return ['ok'=>false,'msg'=>'Dato vacío.'];  
Valida ausencia o vacío. Devuelve array de error.

15 if (!preg\_match('/^\d+\$/ ', trim(\$raw))) return ['ok'=>false,'msg'=>'Debe ser entero ≥ 0.'];  
Valida con regex que solo haya dígitos (0–9). trim() elimina espacios en extremos.  
Rechaza negativos, decimales o símbolos.

16 // El patrón anterior ya garantiza no-negativo  
Comentario aclaratorio.

17 return ['ok'=>true,'val'=>(int)\$raw];  
Conversión a entero y retorno de éxito. Array con ok=true y el valor numérico.

18 }  
Cierre de la función.

19 *(línea en blanco)*  
Separador.

20 /\*\*  
Inicio de documentación para la función principal.

21 \* Determina si un entero no negativo es capicúa (palíndromo).  
Explicación.

22 \* Implementación usando string: compara con su reverso.  
Indica el método de verificación.

23 \* @param int \$n  
Anotación del parámetro.

24 \* @return bool  
Anotación del tipo de retorno.

25 \*/  
Fin del DocBlock.

26 function esCapicua(int \$n): bool {  
Función que recibe un entero y devuelve booleano.

27 \$s = (string)\$n;  
Convierte el número en cadena.

28 \$rev = strrev(\$s);  
Aplica strrev() para obtener la cadena invertida.

29 return \$s === \$rev;  
Compara ambas y retorna true si son idénticas.

30 }  
Cierre de la función.

31 *(línea en blanco)*  
Separador.

32 \$num = null;  
Variable para guardar el número validado.

33 \$resp = null;  
Variable para guardar la respuesta final “SI” o “NO”.

34 \$error = null;  
Variable para mensaje de error.

35 *(línea en blanco)*  
Separador.

36 if (\$\_SERVER['REQUEST\_METHOD'] === 'POST') {  
Evalúa si la petición HTTP es POST.

37 \$res = validar\_num(\$\_POST['num'] ?? null);  
Obtiene y valida el valor ingresado en el formulario. Usa operador ?? para manejar ausencia.

38 if (!\$res['ok']) {  
Rama de error de validación.

39 \$error = \$res['msg'];  
Asigna mensaje de error.

40 } else {  
Rama de validación correcta.

41 \$num = (int)\$res['val'];  
Asigna el número validado.

42 \$resp = esCapicua(\$num) ? 'SI' : 'NO';  
Llama a esCapicua y guarda “SI” o “NO”.

43 }  
Cierra else.

44 }  
Cierra if de POST.

45 *(línea en blanco)*  
Separador.

46 ?>  
Cierra bloque PHP.

47 <!doctype html>  
Define HTML5.

48 <html lang="es">  
Etiqueta raíz con idioma español.

49 <head>  
Cabecera.

50 <meta charset="utf-8" />  
Codificación UTF-8.

51 <title>NB-B3 Número capicúa</title>  
Título de la página.

52 <style>  
Inicio de estilos en línea.

53 body{font-family:system-ui, -apple-system, Segoe UI, Roboto, Arial, sans-serif;  
margin:24px}  
Fuente y márgenes del documento.

54 fieldset{border:1px solid #ccc;padding:12px;border-radius:8px;max-width:520px}  
Estilos del fieldset.

55 label{display:block;margin:8px 0 4px}  
Estilos para etiquetas.

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

Estilos para campo numérico.

```
57 button{padding:8px 14px;border:0;border-radius:8px;background:#07a;color:#fff;cursor:pointer;margin-top:10px}
```

Estilos para botón.

```
58 .error{color:#b00020;font-weight:700}
```

Estilo para mensajes de error.

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

Estilo contenedor de resultados.

```
60 code{background:#f7f7f7;padding:2px 4px;border-radius:4px}
```

Estilo para etiquetas `<code>`.

```
61 </style>
```

Cierre estilos.

```
62 </head>
```

Cierra cabecera.

```
63 <body>
```

Abre cuerpo del documento.

```
64 <h1>NB-B3 – ¿Número capicúa?</h1>
```

Encabezado principal.

```
65 <form method="post" novalidate>
```

Formulario. Envío por POST. novalidate evita validación automática del navegador.

```
66 <fieldset>
```

Grupo de controles.

```
67 <legend>Entrada</legend>
```

Título del grupo.

```
68 <label for="num">Número entero ≥ 0</label>
```

Etiqueta asociada al input.

```
69 <input type="number" id="num" name="num" min="0" step="1" value="php echo htmlspecialchars($_POST['num'] ?? ''); ?&gt;" /&gt;</pre
```

Campo numérico.

type="number": entrada numérica.

id="num": identificador para asociar la etiqueta.

name="num": clave que viaja en POST.

min="0": no admite negativos.

step="1": solo enteros.

value="...": repone valor previo. Se escapa con htmlspecialchars para seguridad.

70 <button type="submit">Evaluar</button>

Botón de envío.

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

Si existe \$error, se muestra dentro de <div class="error">. Se escapa el mensaje.

72 </fieldset>

Cierra fieldset.

73 </form>

Cierra formulario.

74 (*línea en blanco*)

Separación.

75 <?php if (\$resp !== null): ?>

Si existe resultado, lo muestra.

76 <div class="result">

Contenedor para resultados.

77 <strong>Salida:</strong>

Etiqueta.

78 <div>¿Capicúa? <code><?php echo \$resp; ?></code></div>

Muestra la respuesta dentro de <code> ("SI" o "NO").

79 </div>

Cierra contenedor de resultados.

80 <?php endif; ?>

Cierra condicional.

81 </body>

Cierra el cuerpo del documento.

82 </html>

Cierra el documento HTML.