Documentación línea por línea
Autor: Héctor Arciniega Valencia

1 <?php
Apertura del intérprete PHP. Obliga a que todo lo siguiente se ejecute como PHP hasta cerrar o cambiar a HTML.

2 /**
Inicio de un bloque de documentación (DocBlock). No afecta la ejecución. Solo describe el script.

3 * Nivel B – Ejercicio 1 (Condicional)
Título del ejercicio. Texto informativo dentro del DocBlock.

4 * Tarifa eléctrica simple por tramos
Descripción breve del problema. Sin efecto en tiempo de ejecución.

5 *
Línea en blanco dentro del DocBlock para separar secciones.

6 * Entrada: kwh (entero ≥ 0)
Especificación de entrada esperada. No valida por sí misma.

7 * Tarifas:
Encabezado de la sección de tramos de precio.

8 *  - 0..150  → $0.80/kWh
Define el primer tramo tarifario. Es documentación.

9 *  - 151..300→ $1.00/kWh
Define el segundo tramo tarifario. Es documentación.

10 *  - >300  → $1.20/kWh
Define el tercer tramo tarifario. Es documentación.

11 * Salida: kwh, tarifa_aplicada y costo_total (2 decimales).
Especificación de salidas esperadas.

12 */
Cierre del DocBlock.

13 *(línea en blanco)*
Separador visual. Sin efecto.

14 declare(strict_types=1);
Habilita tipado estricto. Las declaraciones de tipo en funciones no permitirán coerción implícita.

15 *(línea en blanco)*
Separador visual. Sin efecto.

16 function validar_kwh(?string $raw): array {
Declaración de función pública validar_kwh.
Entrada: string|null. Salida: array. Inicia el bloque de función.

17 // 1) Verificar que haya dato
Comentario. Indica la intención del bloque siguiente.

18 if ($raw === null || $raw === '') {
Condición. Detecta ausencia de dato: null o cadena vacía.

19 return ['ok'=>false, 'msg'=>'Dato vacío.'];
Retorno temprano ante ausencia. Estructura con bandera de error y mensaje.

20 }
Cierre del if de ausencia de dato.

21 // 2) Debe ser entero no negativo (sin decimales)
Comentario. Indica validación de formato numérico entero.

22 if (!preg_match('/^\d+$/', trim($raw))) {
Valida con regex que, tras trim, la entrada contenga solo dígitos. Rechaza signos, puntos y espacios internos.

23 return ['ok'=>false, 'msg'=>'Debe ser entero no negativo.'];
Retorno de error por formato inválido.

24 }
Cierre del if de formato.

25 $val = (int)$raw;
Conversión explícita a entero. Asume que la regex ya garantizó solo dígitos.

26 // 3) Rango válido
Comentario. Indica validación de rango.

27 if ($val < 0) {
Condición redundante en la práctica por la regex, pero protege ante futuros cambios.

28 return ['ok'=>false, 'msg'=>'No puede ser negativo.'];
Retorno de error por valor negativo.

29 }
Cierre del if de rango.

30 return ['ok'=>true, 'val'=>$val];
Retorno de éxito. Entrega el entero validado bajo clave val.

31 }
Cierre de la función validar_kwh.

32 *(línea en blanco)*
Separador. Sin efecto.

33 $kwh = null;
Inicializa variable de consumo a null. Tipo esperado posterior: int.

34 $tarifa = null;
Inicializa variable de tarifa por kWh a null. Tipo esperado posterior: float.

35 $costo = null;
Inicializa costo total a null. Tipo esperado posterior: float.

36 $error = null;
Inicializa mensaje de error a null. Tipo posterior: string|null.

37 *(línea en blanco)*
Separador. Sin efecto.

38 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
Control de flujo. Solo procesa cálculo cuando el método HTTP es POST.

39 $res = validar_kwh($_POST['kwh'] ?? null);
Llama a validar_kwh con el campo kwh recibido. Usa null coalescing si no existe.

40 if (!$res['ok']) {
Revisa bandera de validación.

41 $error = $res['msg'];
Guarda el mensaje de error para mostrarlo en la vista.

42 } else {

Alternativa cuando la validación es correcta.

43 $kwh = (int)$res['val'];
Asigna el entero validado a $kwh. Asegura tipo int.

44 *(línea en blanco)*
Separador. Sin efecto.

45 // Determinar tarifa según tramo (bordes 150 y 300 incluidos donde corresponde).
Comentario. Explica límites inclusivos.

46 if ($kwh <= 150) {
Condición primer tramo. Incluye 0 a 150.

47 $tarifa = 0.80;
Tarifa por kWh para el primer tramo. Tipo float.

48 } elseif ($kwh <= 300) {
Segundo tramo. Aplica a 151 a 300 inclusive.

49 $tarifa = 1.00;
Tarifa segundo tramo.

50 } else {
Tercer tramo. Aplica a valores > 300.

51 $tarifa = 1.20;
Tarifa tercer tramo.

52 }
Cierre de la estructura condicional de tramos.

53 *(línea en blanco)*
Separador. Sin efecto.

54 // Costo total = kwh * tarifa
Comentario. Define la fórmula.

55 $costo = $kwh * $tarifa;
Calcula el costo total. Resultado float.

56 }
Cierre del else de validación correcta.

57 }
Cierre del if de método POST.

58 *(línea en blanco)*
Separador. Sin efecto.

59 function fmt($n){ return number_format((float)$n, 2, '.', ','); }
Función de utilidad. Recibe numérico. Fuerza float. Devuelve string con 2 decimales, punto decimal y coma de miles.

60 ?>
Cierra el bloque PHP. A partir de aquí se emite HTML literal salvo bloques PHP de apertura corta.

61 <!doctype html>
Declaración del tipo de documento HTML5.

62 <html lang="es">
Raíz del documento. Idioma español para accesibilidad y SEO.

63 <head>
Inicio del encabezado HTML.

64 <meta charset="utf-8" />
Declara codificación UTF-8. Evita problemas de acentos y símbolos.

65 <title>NB-B1 Tarifa eléctrica</title>
Título de la página. Aparece en la pestaña del navegador.

66 <style>
Inicio de estilos en línea CSS. Aplica a este documento.

67 body{font-family:system-ui, -apple-system, Segoe UI, Roboto, Arial, sans-serif; margin:24px}
Regla CSS. Define tipografías de sistema y un margen general de 24px.

68 fieldset{border:1px solid #ccc;padding:12px;border-radius:8px;max-width:520px}
Regla CSS. Estilo del contenedor de campos: borde, padding, esquinas redondeadas y ancho máximo.

69 label{display:block;margin:8px 0 4px}
Regla CSS. Muestra etiquetas como bloque con márgenes superior e inferior.

70 input[type=number]{width:100%;padding:8px;border:1px solid #bbb;border-radius:6px}
Regla CSS. Estilo de input numérico: ancho completo, padding, borde y esquinas.

71 button{padding:8px 14px;border:0;border-radius:8px;background:#07a;color:#fff;cursor:pointer;margin-top:10px}
Regla CSS. Estilo del botón: relleno, sin borde, radio, color de fondo, color de texto, cursor y margen superior.

72 .error{color:#b00020;font-weight:700}
Regla CSS. Estilo de mensajes de error: rojo y negritas.

73 .result{margin-top:12px;padding:8px;border-left:4px solid #07a;background:#eef7ff}
Regla CSS. Estilo del bloque de resultados: margen, padding, borde izquierdo y fondo.

74 code{background:#f7f7f7;padding:2px 4px;border-radius:4px}
Regla CSS. Estilo para contenido dentro de <code>.

75 </style>
Cierra el bloque de estilos.

76 </head>
Cierra el encabezado HTML.

77 <body>
Abre el cuerpo del documento.

78 <h1>NB-B1 – Tarifa eléctrica simple</h1>
Encabezado visible principal de la página.

79 <form method="post" novalidate>
Formulario HTML. Envía por POST. novalidate desactiva validaciones nativas del navegador.

80 <fieldset>
Agrupa controles del formulario.

81 <legend>Entrada</legend>
Título del grupo de campos.

82 <label for="kwh">kWh consumidos (entero ≥ 0)</label>
Etiqueta asociada al control con id kwh. Texto guía para el usuario.

83 <input type="number" id="kwh" name="kwh" min="0" step="1" value="<?php echo htmlspecialchars($_POST['kwh'] ?? ''); ?>" />
Campo numérico. Acepta enteros ≥ 0 (min="0", step="1"). Prellena con el último valor enviado, escapado con htmlspecialchars para prevenir XSS.

84 <button type="submit">Calcular</button>
Botón para enviar el formulario.

85 <?php if ($error): ?> <?php echo htmlspecialchars($error); ?> <?php endif; ?>
Bloque PHP corto. Si existe $error no vacío, renderiza un  con la clase error y el mensaje escapado.

86 </fieldset>
Cierra el grupo de campos.

87 </form>
Cierra el formulario.

88 *(línea en blanco)*
Separador visual.

89 <?php if ($kwh !== null && $tarifa !== null): ?>
Condición de salida. Muestra resultados solo si ya se computaron $kwh y $tarifa.

90 
Contenedor visual de resultados.

91 Salida:
Etiqueta textual para la sección de salida.

92 kWh: <code><?php echo $kwh; ?></code>
Muestra el consumo final como número dentro de <code>.

93 Tarifa aplicada: $<code><?php echo fmt($tarifa); ?></code> por kWh
Muestra la tarifa por kWh. Formateada con fmt a dos decimales con separador de miles.

94 Costo total: $<code><?php echo fmt($costo); ?></code>
Muestra el costo total formateado con fmt.

95 
Cierra el contenedor de resultados.

96 <?php endif; ?>
Cierra la condición que gobierna la sección de resultados.

97 </body>
Cierra el cuerpo del documento.

98 </html>
Cierra el documento HTML.

---

## Notas de validación y seguridad

- preg_match('/^\d+$/', trim($raw)) evita inyección de símbolos y decimales. Correcto para enteros no negativos.
- htmlspecialchars(...) en valores reinyectados y en errores bloquea XSS.
- strict_types=1 asegura que firmas de funciones se respeten.
- number_format(..., 2, '.', ',') fija formato consistente de números mostrados.