# Documentación: b1.php
# Autor: Héctor Arciniega Valencia

*Nivel B – Ejercicio 1*

**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 1 (Condicional)**
• Rol: comentario; sin efecto en tiempo de ejecución.

**4 * Tarifa eléctrica simple por tramos**
• Rol: comentario; sin efecto en tiempo de ejecución.

**5 ***
• Rol: comentario; sin efecto en tiempo de ejecución.

**6 * Entrada: kwh (entero ≥ 0)**
• Rol: comentario; sin efecto en tiempo de ejecución.

**7 * Tarifas:**
• Rol: comentario; sin efecto en tiempo de ejecución.

**8 * - 0..150 → $0.80/kWh**
• Rol: comentario; sin efecto en tiempo de ejecución.

**9 * - 151..300→ $1.00/kWh**
• Rol: comentario; sin efecto en tiempo de ejecución.

**10 * - >300 → $1.20/kWh**
• Rol: comentario; sin efecto en tiempo de ejecución.

**11 * Salida: kwh, tarifa_aplicada y costo_total (2 decimales).**
• Rol: comentario; sin efecto en tiempo de ejecución.

**12 */**
• Rol: fin de DocBlock.

**13**

**14 declare(strict_types=1);**
• Rol: directiva strict_types.
• Efecto: activa tipado estricto en este archivo; prohíbe coerciones implícitas en firmas.

**15**

**16 function validar_kwh(?string $raw): array {**
• Rol: declaración de función.
• Nombre: validar_kwh
• Parámetro $raw: tipo declarado ?string.
• Retorno: array.
• Contrato: retorna array: {ok:bool, val?:T, msg?:string}.

**17 // 1) Verificar que haya dato**
• Rol: comentario; sin efecto en tiempo de ejecución.

**18 if ($raw === null || $raw === '') {**
• Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.
• Comparación con cadena vacía para validar presencia de datos.

**19 return ['ok'=>false, 'msg'=>'Dato vacío.'];**
• Rol: retorno; finaliza la función devolviendo el valor indicado.

**20 }**

**21 // 2) Debe ser entero no negativo (sin decimales)**
• Rol: comentario; sin efecto en tiempo de ejecución.

**22 if (!preg_match('/^\d+$/', trim($raw))) {**
• Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.
• Función: preg_match(patrón, sujeto, [coincidencias], [flags], [offset]). Devuelve 1/0 o false.
• Patrón: /^\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.

**23 return ['ok'=>false, 'msg'=>'Debe ser entero no negativo.'];**
• Rol: retorno; finaliza la función devolviendo el valor indicado.

**24 }**

**25 $val = (int)$raw;**

**26 // 3) Rango válido**
• Rol: comentario; sin efecto en tiempo de ejecución.

**27 if ($val < 0) {**
• Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.

**28 return ['ok'=>false, 'msg'=>'No puede ser negativo.'];**
• Rol: retorno; finaliza la función devolviendo el valor indicado.

**29 }**

**30 return ['ok'=>true, 'val'=>$val];**
• Rol: retorno; finaliza la función devolviendo el valor indicado.

**31 }**

**32**

**33 $kwh = null;**
• Rol: inicialización de variable a null.

**34 $tarifa = null;**
• Rol: inicialización de variable a null.

**35 $costo = null;**
• Rol: inicialización de variable a null.

**36 $error = null;**
• Rol: inicialización de variable a null.

**37**

**38 if ($_SERVER['REQUEST_METHOD'] === 'POST') {**
• Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.
• Condición: $_SERVER['REQUEST_METHOD'] === 'POST' (método HTTP).

**39 $res = validar_kwh($_POST['kwh'] ?? null);**

**40 if (!$res['ok']) {**
• Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.

**41 $error = $res['msg'];**

**42 } else {**

**43 $kwh = (int)$res['val'];**

**44**

**45 // Determinar tarifa según tramo (bordes 150 y 300 incluidos donde corresponde).**
• Rol: comentario; sin efecto en tiempo de ejecución.

**46 if ($kwh <= 150) {**
• Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.

**47 $tarifa = 0.80;**

**48 } elseif ($kwh <= 300) {**
• Rol: rama elseif; evalúa si el if anterior fue false.

**49 $tarifa = 1.00;**

**50 } else {**

**51 $tarifa = 1.20;**

**52 }**

**53**

**54 // Costo total = kwh * tarifa**
• Rol: comentario; sin efecto en tiempo de ejecución.

**55 $costo = $kwh * $tarifa;**

**56 }**

**57 }**

**58**

**59 function fmt($n){ return number_format((float)$n, 2, '.', ','); }**
• Rol: declaración de función.
• Nombre: fmt
• Parámetro $n: tipo declarado mixed.
• Función: number_format(número, decimales, separador_decimal, separador_miles). Retorna string formateado.

**60 ?>**
• Rol: cierre del bloque PHP; desde aquí se emite HTML sin evaluar.

**61 <!doctype html>**
• Rol: declaración HTML5 (modo estándar).

**62 <html lang="es">**
• Atributos: lang="es": idioma del documento.

**63 <head>**

**64 <meta charset="utf-8" />**
• Atributos: charset="utf-8": codificación de caracteres.

**65 <title>NB-B1 Tarifa eléctrica</title>**

**66 <style>**

**67 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.

**68 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.

**69 label{display:block;margin:8px 0 4px}**
• CSS: display:block: ocupa toda la línea disponible. margin: 8px 0 4px; margen externo del elemento.

**70 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.

**71 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.

**72 .error{color:#b00020;font-weight:700}**
• CSS: color: #b00020; color del texto.

**73 .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.

**74 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.

**75 </style>**

**76 </head>**

**77 <body>**

**78 <h1>NB-B1 – Tarifa eléctrica simple</h1>**

**79 <form method="post" novalidate>**
• Atributos: novalidate: desactiva validación HTML5; validación queda del lado servidor.

**80 <fieldset>**

**81 <legend>Entrada</legend>**

**82 <label for="kwh">kWh consumidos (entero ≥ 0)</label>**
• Atributos: for="kwh": asocia la etiqueta con el control id="kwh".

**83 <input type="number" id="kwh" name="kwh" min="0" step="1" value="<?php echo htmlspecialchars($_POST['kwh'] ?? ''); ?>" />**
• 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="kwh": identificador único en el DOM. name="kwh": clave enviada en la petición (GET). min="0": valor mínimo permitido por el navegador. step="1": incremento permitido (enteros). value="<php…>": relleno dinámico; se escapa con htmlspecialchars para evitar XSS.

**84 <button type="submit">Calcular</button>**
• Atributos: type="submit": tipo de control.

**85 <?php if ($error): ?> <?php echo htmlspecialchars($error); ?> <?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.

**86 </fieldset>**

**87 </form>**

**88**

**89 <?php if ($kwh !== null && $tarifa !== null): ?>**
• Rol: apertura del intérprete PHP; el servidor evalúa el bloque.

**90 **

**91 Salida:**

**92 kWh: <code><?php echo $kwh; ?></code> **

**93 Tarifa aplicada: $<code><?php echo fmt($tarifa); ?></code> por kWh **

**94 Costo total: $<code><?php echo fmt($costo); ?></code> **

**95 **

**96 <?php endif; ?>**
• Rol: apertura del intérprete PHP; el servidor evalúa el bloque.

**97 </body>**

**98 </html>**