

# Solución de la tarea para DWES02.



A continuación se presenta una posible solución a la tarea. Aspectos a destacar de la solución son los siguientes:

- Saneamiento y validación de los valores introducidos en el formulario por medio de la función `filter_input`.
- Separación de la lógica del ejercicio del código de la creación de la vista. Se contemplan los distintos casos en cada parte del script.
- Construcción de la vista mediante bloques embebidos de código PHP, para evaluar expresiones (`<?= ?>`) o para tomar decisiones sobre el contenido de la página a enviar al cliente (condicionales `<?php if (condicion): ?>` y repetitiva `<?php foreach (...): ?>`).
- No se utilizan sentencias "echo" para escribir la página sino que se escribe HTML directamente.
- Uso de un array bidimensional para almacenar las tareas de la agenda y su estado de completado.
- Compactación de los datos de la agenda en un array construido con campos ocultos del formulario.
- Uso de varios botones de envío con peticiones diferentes en el formulario.
- Uso de la función `empty` para tomar decisiones sobre el contenido que debe figurar en la vista.

Archivo `index.php`:

```
<?php
$tasks = (filter_input(INPUT_POST, 'tasks', FILTER_UNSAFE_RAW, FILTER_REQUIRE_ARRAY)) ?? []; // Contiene la lista de tareas
if (filter_has_var(INPUT_POST, 'crear_tarea')) { // Si se solicita la creación de una tarea
    $task = trim(filter_input(INPUT_POST, 'tarea', FILTER_UNSAFE_RAW)); // Lee la tarea del formulario
    if (empty($task)) { // Si la tarea no es la cadena vacía
        $errorNombreTarea = true; // Error en el valor introducido como nombre de tarea
    } else {
        $tasks[] = ['nombre' => $task, 'estado' => 0]; // Añado la tarea a la lista
    }
} else if (filter_has_var(INPUT_POST, 'borrar_tarea')) { // Si se solicita que se borre la tarea
    $taskId = filter_input(INPUT_POST, 'tarea_id', FILTER_VALIDATE_INT); // Se lee el número de tarea (uno más que el índice real)
    unset($tasks[$taskId - 1]); // Se borra la tarea de la lista
    $tasks = array_values($tasks); // Se reindexa la lista de tareas para que los índices sean consecutivos
} else if (filter_has_var(INPUT_POST, 'completar_tarea')) { // Si se solicita que se complete una tarea
    $taskId = filter_input(INPUT_POST, 'tarea_id', FILTER_VALIDATE_INT); // Se lee el número de tarea (uno más que el índice real)
    $tasks[$taskId - 1]['estado'] = 1; // Se cambia el estado de completado de la tarea
}
?>
<!DOCTYPE html>
```

```

<html lang="es">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="stylesheet.css">
    <title>Agenda de Tareas</title>
  </head>
  <body>
    <main class="principal">
      <h1>Agenda de Tareas</h1>
      <form class="agenda" action="?= $_SERVER['PHP_SELF'] ?>" method="POST"> <!-- Creo un formulario que envía los datos de
        <fieldset class="section"> <!-- Sección de nueva tarea -->
          <legend>Nueva Tarea:</legend>
          <div class="form-section">
            <label for="tarea">Tarea:</label>
            <input id="tarea" type="text" name="tarea">
            <?php if (isset($errorNombreTarea)): ?> <!-- Si la tarea está vacía es muestra un mensaje de error -->
              <p class="error">(*) Nombre obligatorio</p>
            <?php endif ?>
          </div>
          <div class="form-section">
            <input class="submit blue" type="submit" value="Añadir Tarea" name="crear_tarea"/> <!-- Envío de petición de
            <input class="submit green" type="reset" value="Limpiar Campos"/>
          </div>
        </fieldset>
        <?php if (!empty($tareas)): ?>
          <fieldset class="section"> <!-- Sección de creación de la tabla de tareas -->
            <legend>Lista de Tareas:</legend>
            <table>
              <thead>
                <tr>
                  <th>Número Tarea</th>
                  <th>Tarea</th>
                  <th>Completado</th>
                </tr>
              </thead>
              <tbody>
                <?php foreach ($tareas as $numTarea => $tarea): ?> <!-- Bucle de creación de las filas de la tabla -
                  <tr>

```

```

        <td><?= $numTarea + 1 ?></td> <!-- Añado uno al índice para que la lista se inicie en 1 -->
        <td><?= htmlentities($tarea['nombre']) ?></td>
        <td><?= ($tarea['estado']) ? "Si" : "No" ?></td>
    </tr>
    <input type='hidden' name="<?= "tareas[$numTarea][nombre]" ?>" value="<?= $tarea['nombre'] ?>"> <!--
    <input type='hidden' name="<?= "tareas[$numTarea][estado]" ?>" value="<?= $tarea['estado'] ?>"> <!--
    <?php endforeach ?>
</tbody>
</table>
</fieldset>
<fieldset class="section"> <!-- Sección de operaciones sobre la lista de tareas -->
    <div class="form-section">
        <label class="blue" for="tarea">Num Tarea:</label>
        <input id="tarea" type="number" min="1" max="<?= count($tareas) ?>" value="1" name="tarea_id">
        <input class="submit blue" type="submit" value="Tarea Completada" name='completar_tarea' />
        <input class="submit blue" type="submit" value="Tarea Borrada" name='borrar_tarea' />
        <input class="submit red" type="submit" formmethod = "GET" formaction="<?= "{$_SERVER['PHP_SELF']}"?limpiar_tareas" />
        <!-- Otra manera de enviar la petición al servidor con un mensaje GET -->
        <!-- Esta forma no requiere que el botón sea de tipo submit -->
        <!-- <a href="<?= "{$_SERVER['PHP_SELF']}"?limpiar_tareas" ?>"><input class="submit red" value="Vaciar Agend
    </div>
</fieldset>
<?php else: ?>
    <fieldset class="section">
        <legend>Lista de Tareas:</legend>
        <p class="blue">No hay tareas</p>
    </fieldset>
<?php endif ?>
</form>
</main>
</body>
</html>

```

Archivo stylesheet.css

```

* {
    margin: 0;
}

```

```
padding: 0;
box-sizing: border-box;
}

.principal {
display: flex;
flex-flow: column nowrap;
margin: 10px;
align-items: center;
}

.agenda {
display: flex;
flex-flow: column nowrap;
align-items: center;
width: 80%;
}

.section {
display: flex;
flex-flow: column nowrap;
flex: 1 0;
margin: 10px;
padding: 10px;
align-content: center;
width: 60%;
}

blue {
color: blue;
}

.form-section {
margin: 5px;
display: flex;
flex-wrap: wrap;
align-items: center;
}
```



```
body {
  background:#FA8072;
}

label {
  flex: 1 0;
  max-width: 100px;
  color:blue;
  font-weight: bold;
}

td, th {
  text-align: center;
}

input[type=text]{
  width: 300px;
}

input{
  border-style: none;
  border-radius: 5px;
  margin: 5px;
  height: 25px;
}

.error {
  color: white;
  margin-left: 10px;
}

.submit {
  height: 30px;
  padding: 5px;
}

.blue {
  color: blue;
}
```



```
}
```

```
.green {  
  color: green;  
}
```

```
.red {  
  color: red;  
}
```

