

2-11-2023

Características del lenguaje PHP.

Tarea 02

ROBERTO RODRÍGUEZ JIMÉNEZ
roberto.rodjim.1@educa.jcyl.es

Contenido

Tarea online DWES01	2
Enunciado.....	2
Caso práctico	2
.....	2
Caso práctico	2
¿Qué te pedimos que hagas?	2
Recursos necesarios y recomendaciones.....	4
Recursos necesarios	4
Recomendaciones	4
Evaluación de la tarea	4
Criterios de evaluación implicados	4
¿Cómo valoramos y puntuamos tu tarea?	4
RESOLUCIÓN	5
Creación del formulario	5
Si el nombre está vacío, se mostrará una advertencia.....	6
Si el nombre que se introdujo no existe en la agenda, y el número de teléfono no está vacío, se añadirá a la agenda.....	7
Si el nombre que se introdujo ya existe en la agenda y se indica un número de teléfono, se sustituirá el número de teléfono anterior	7
Si el nombre que se introdujo ya existe en la agenda y no se indica número de teléfono, se eliminará de la agenda la entrada correspondiente a ese nombre.	8
En el momento que la agenda tenga un nombre nos aparecerá un nuevo formulario que nos permitirá vaciar todos los contactos.	8

Tarea online DWES01

Título de la tarea: Preferencias de Usuario con Sesiones

Unidad 2: Características del lenguaje PHP.

Ciclo formativo y módulo: Ciclo superior de DAW. Módulo de DWES.

Curso académico: ...

Enunciado

Caso práctico

Caso práctico

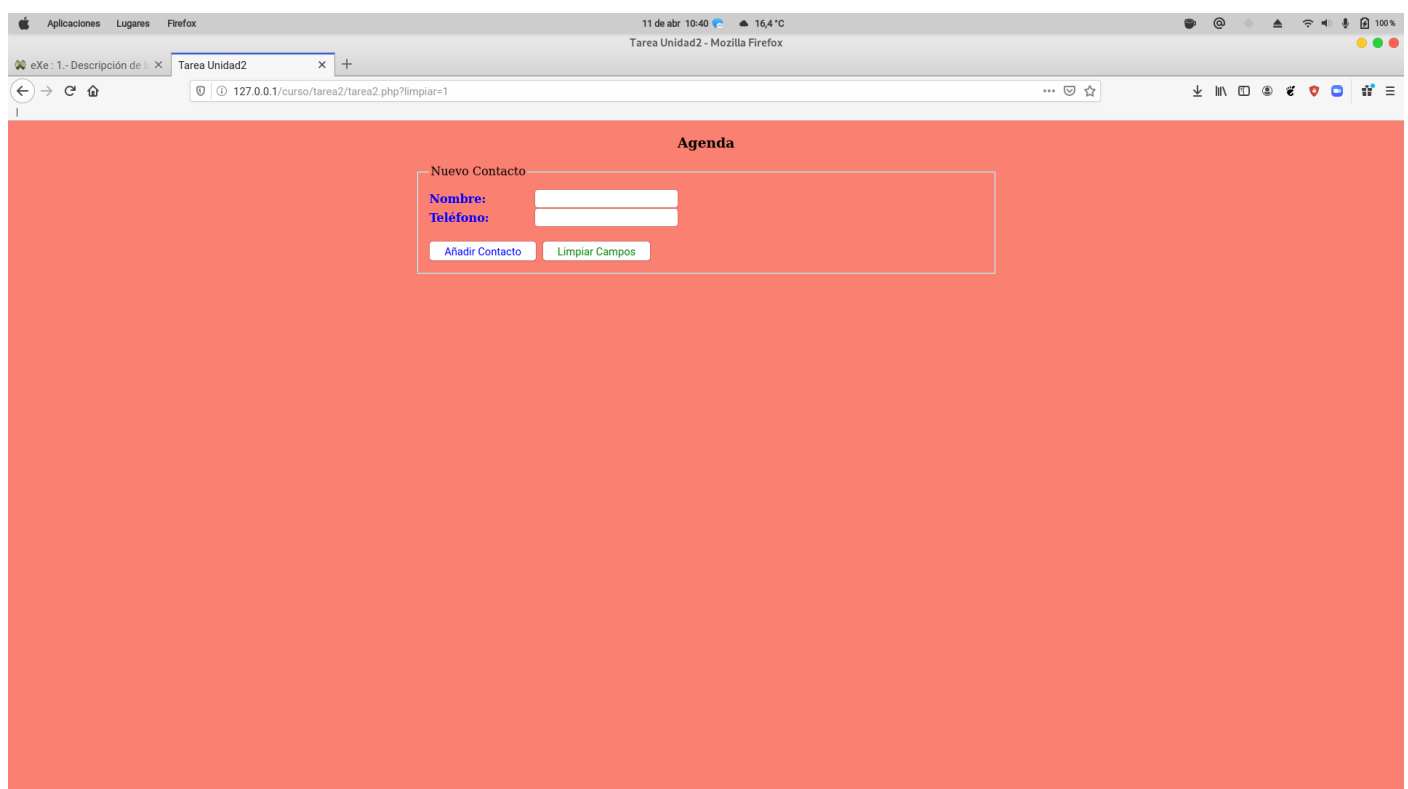
El presidente de la Comunidad de vecinos donde vive Carlos, le comenta la posibilidad de poder acceder al nombre y teléfono de los vecinos de una forma cómoda y sencilla, Carlos piensa que con lo que ya sabe de formularios podría echarle una mano y se pone a pensar en una posible implementación.

¿Qué te pedimos que hagas?

Debes programar una aplicación para mantener una pequeña agenda en una única página web programada en PHP.

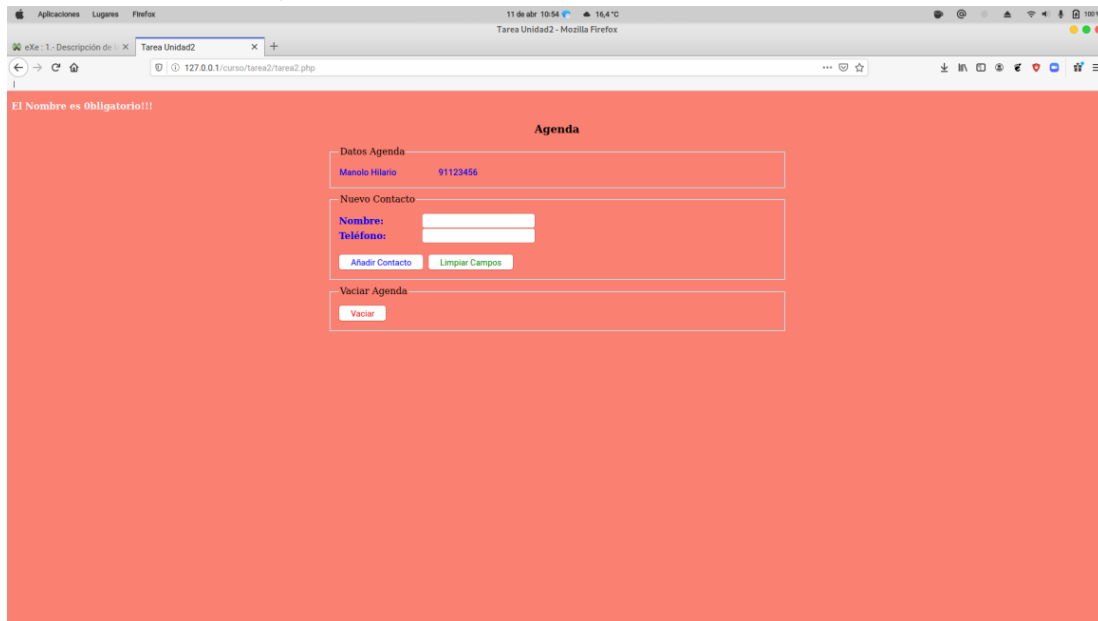
La agenda almacenará únicamente dos datos de cada persona: su nombre y un número de teléfono. Además, no podrá haber nombres repetidos en la agenda.

En la parte superior de la página web se mostrará el contenido de la agenda. En la parte inferior debe figurar un sencillo formulario con dos cuadros de texto, uno para el nombre y otro para el número de teléfono.

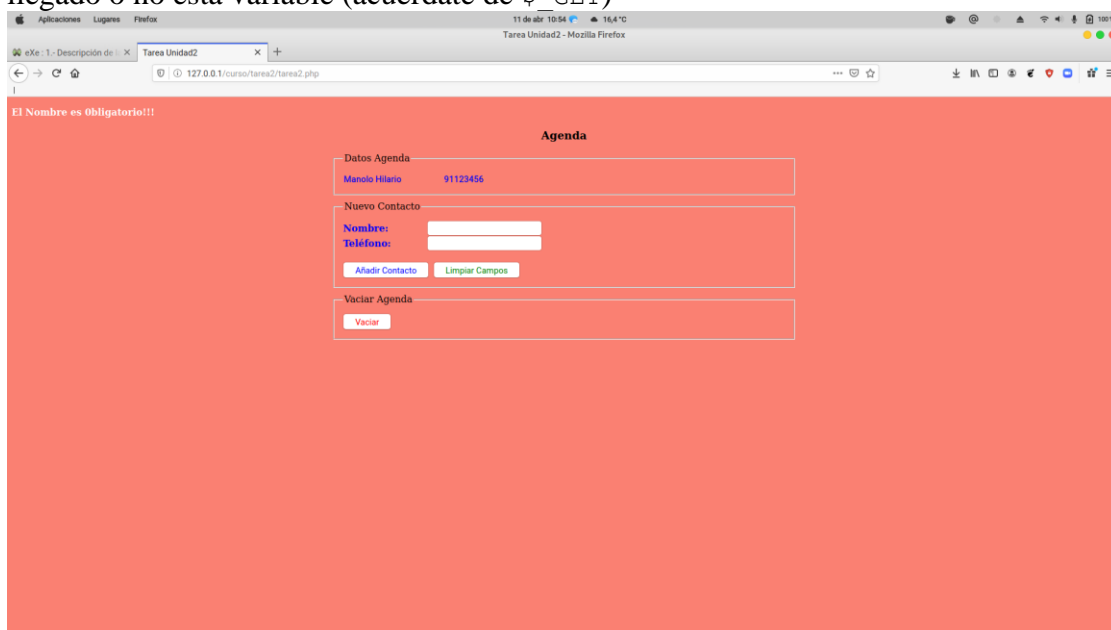


Cada vez que se envíe el formulario:

- ✓ Si el nombre está vacío, se mostrará una advertencia.



- ✓ Si el nombre que se introdujo no existe en la agenda, y el número de teléfono no está vacío, se añadirá a la agenda
- ✓ Si el nombre que se introdujo ya existe en la agenda y se indica un número de teléfono, se sustituirá el número de teléfono anterior.
- ✓ Si el nombre que se introdujo ya existe en la agenda y no se indica número de teléfono, se eliminará de la agenda la entrada correspondiente a ese nombre.
- ✓ En el momento que la agenda tenga un nombre nos aparecerá un nuevo formulario que nos permitirá vaciar todos los contactos. Para ello mandaremos en la url una variable a la misma página de la agenda (fíjate en la url de la primera imagen). Al procesar la página comprobaremos si nos ha llegado o no esta variable (acuérdate de \$ _GET)



Recursos necesarios y recomendaciones

Recursos necesarios

Ordenador con un entorno AMP correctamente instalado y configurado y un IDE, como Visual Studio Code, con las extensiones adecuadas según lo visto en el tema anterior.

Recomendaciones

Se recomienda emplear como apoyo en el desarrollo del examen un navegador con acceso a Internet, para poder consultar el manual online de PHP .

Evaluación de la tarea

Criterios de evaluación implicados

- ✓ Se han utilizado mecanismos de decisión en la creación de bloques de sentencias.
- ✓ Se han utilizado bucles y se ha verificado su funcionamiento
- ✓ Se han utilizado arrays para almacenar y recuperar conjuntos de datos.
- ✓ Se han creado y utilizado funciones.
- ✓ Se han utilizado formularios Web para interactuar con el usuario del navegador Web.
- ✓ Se han empleado métodos para recuperar la información introducida en el formulario.
- ✓ Se han añadido comentarios al código.
- ✓ Se han identificado los mecanismos disponibles para el mantenimiento de la información que concierne a un cliente Web concreto y se han señalado sus ventajas.
- ✓ Se han realizado adaptaciones a aplicaciones Web existentes como gestores de contenidos u otras.

¿Cómo valoramos y puntuamos tu tarea?

Rúbrica de la tarea (10 Puntos)

Generar la estructura de la página PHP correctamente.	1
Mostrar los contactos existentes en la agenda de manera adecuada.	1
Generar el formulario de introducción de nuevo contacto.	0.5
Introducir los datos de la agenda como campos ocultos en el formulario.	1
Comprobar los datos enviados por el formulario, mostrando una advertencia cuando no se manda el nombre.	0.5
Introducir en la agenda los datos de un nuevo contacto.	1
Modificar el teléfono de un contacto ya existente.	0.5
Eliminar de la agenda un contacto.	0.5
Muestra correctamente el enlace a vaciar agenda en el momento que la agenda tiene datos	1
Eliminar todos los contactos de la agenda utilizando correctamente los enlaces para mandar datos por GET.	1
Utilizar un array asociativo.	1
Introducir comentarios en el código y cuidar el diseño de la página.	1

Creación del formulario

El formulario tiene los campos necesarios.

Los tipos de los *inputs* se especifican para cada campo: *text* para el nombre y *tel* para el teléfono.

Para el teléfono se indica expresión que debe cumplirse: `[0-9]{3}`.

El formulario carga la misma página *index.php* para poder mostrar la información tras procesar los datos.

Para limpiar los campos del formulario se usa un botón de tipo *reset*.

Formulario de registro con el título 'Agenda'. Contiene una sección 'Nuevo contacto' con dos campos de entrada: 'Nombre:' y 'Telefono:'. El campo 'Telefono:' contiene el valor '123456789'. Debajo de los campos hay dos botones: 'Añadir contacto' y 'Limpiar campos'.

Formulario de registro

Formulario y registros con el título 'Agenda'. Contiene una tabla con los registros de la agenda:

Nombre	Teléfono
Benito Boniato	123456789
Roberto Rodríguez	898989898
Junior	987654321

Debajo de la tabla hay una sección 'Nuevo contacto' con los mismos campos de entrada que el formulario de registro, pero el campo 'Nombre:' tiene un mensaje de error 'Min. 3 caracteres'. Debajo de los campos hay dos botones: 'Añadir contacto' y 'Limpiar campos'. En la parte inferior hay una sección 'Vaciar agenda' con un botón 'Vaciar'.

Formulario y registros

El contenido de la agenda se muestra si hay registros

```
<?php if (count($agenda) >= 1): ?>
    <fieldset>
        <legend>Agenda</legend>
        <table>
            <tr>
                <th>Nombre</th>
                <th>Teléfono</th>
            </tr>
            <?php foreach ($agenda as $key => $value): ?>
                <tr>
                    <td>
                        <?= $key ?>
                    </td>
                    <td class="telefono">
                        <?= $value ?>
                    </td>
                </tr>
            <?php endforeach ?>
        </table>
    </fieldset>
<?php endif ?>
```

Si el nombre está vacío, se mostrará una advertencia.

Agenda

El nombre es obligatorio

Nombre	Teléfono
Benito Boniato	898989898
Lorenza	545123234

Nuevo contacto

Nombre:

Telefono:

Vaciar agenda

Se envía el formulario y se carga la misma página `index.php`. Al recibirse datos por post, lo primero que se hace es validar el formulario llamando a `validarFormulario()`, que recibe un array con el índice de los errores encontrados.

```
if (count($_POST) > 0) {
    $errores = validarFormulario();
}
```

`validarFormulario()` se ha implementado en `funciones.inc.php`

```
function validarFormulario()
{
    $errores = [];
    if (!isset($_POST["nombre"]) || empty($_POST['nombre'])) {
        $errores[] = 0;
    } else if (!validarNombre($_POST['nombre'])) {
        $errores[] = 1;
    }

    . . .

    return $errores;
}
```

Se evalúa tanto que no esté vacío como que tenga un formato adecuado, llamando a la función `validarNombre()` la cual devuelve `true` o `false` dependiendo de la validez.

```
function validarNombre($nombre)
{
    $temp = trim($nombre);

    $temp = str_replace(['á', 'é', 'í', 'ó', 'ú'], ['a', 'e', 'i', 'o', 'u'], $temp);
    $pattern = "/^[a-zA-Z ]+$/";
    return ((preg_match($pattern, $temp) == 1) && (strlen($temp) >= 3)) ? true : false;
}
```

Si el nombre que se introdujo no existe en la agenda, y el número de teléfono no está vacío, se añadirá a la agenda

Se reciben los datos del formulario en la misma página index.php y, después de validar el formulario y si este no contiene errores, le llama a `procesarAgenda()`.

```
if (count($_POST) > 0) {  
  
    . . .  
  
    if (!$hayError) {  
  
        $agenda = procesarAgenda(  
            isset($_POST['agenda']) ? $_POST['agenda'] : $agenda,  
            $_POST['nombre'],  
            $_POST['telefono']  
        );  
    }  
}
```

Procesar agenda recibe los datos y actúa en relación con ellos, no debe evaluar ni realizar otras operaciones que no sean exclusivamente modificar la agenda.

Tras comprobar los datos recibidos, ejecuta la acción que le corresponde.

```
function procesarAgenda($datos, $nombre, $telefono)  
{  
    $agenda = $datos;  
  
    if (!empty($nombre)) {  
  
        if (!empty($telefono)) {  
            $agenda[$nombre] = $telefono;  
        } else {  
            unset($agenda[$nombre]);  
        }  
    }  
  
    return $agenda;  
}
```

Si el nombre que se introdujo ya existe en la agenda y se indica un número de teléfono, se sustituirá el número de teléfono anterior

El procedimiento es el mismo que el anterior, usando el nombre como índice el cual, en este caso existe.

Si el nombre que se introdujo ya existe en la agenda y no se indica número de teléfono, se eliminará de la agenda la entrada correspondiente a ese nombre.

Se vuelve a usar la función anterior, `procesarAgenda()`.

Desde `procesarAgenda()` se evalúa si el teléfono está vacío y, si lo está, se elimina el registro del array.

En el momento que la agenda tenga un nombre nos aparecerá un nuevo formulario que nos permitirá vaciar todos los contactos.

En `index.php` declaramos un nuevo formulario que solo se muestra si la agenda tiene registros.

```
<?php if (count($agenda) >= 1): ?>
    <form action="" id="form-vaciar" method="get" name="form-vaciar">
        <fieldset>
            <legend>Vaciar agenda</legend>
            <button type="submit" value="1" name="vaciar" id="1">Vaciar</button>
        </fieldset>
    </form>
<?php endif ?>
```

El formulario carga la misma página `index.php` pero pasa los datos por URL mediante el método GET. Cuando se recibe un valor por GET, se comprueba si existe 'vaciar' y, si existe, se resetea el array.

```
if (count($_GET) > 0) {
    if (isset($_GET['vaciar']) && ($_GET['vaciar'] == 1)) {
        reset($agenda);
    }
}
```