



Tarea DWES02

Christian Miguel Giménez Puerma - 2ºDAW

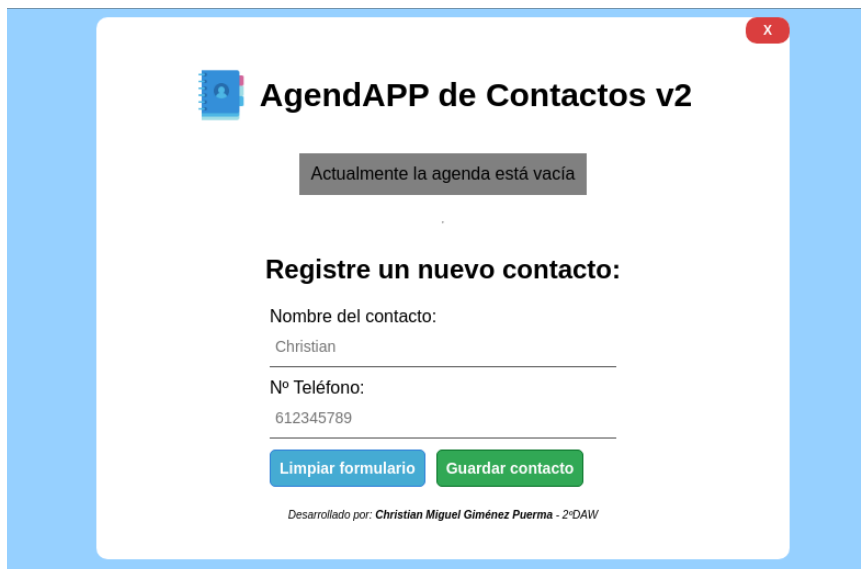


Sumario

1. Descripción de la solución:.....	3
1.1. Descripción lógica/funcional de cada componente.....	3
1.1.1. Si el usuario introdujo el teléfono pero no un nombre.....	4
1.1.2. Si el usuario introdujo el nombre y el teléfono, y este sí existe en la agenda, se elimina el contacto.....	4
1.1.3. Si el usuario introdujo el nombre y el teléfono, y este no existe en la agenda.....	5
1.1.5. Botón para restablecer/borrar los datos de la agenda.....	5
1.2. Descripción técnica de cada componente.....	6
1.2.1. Fichero agenda_v2.php.....	6
1.2.1.1. Cabecera web, metadatos e importación de <i>funciones.php</i>	6
1.2.1.2. Inicialización de la aplicación y validación de datos de entrada.....	7
1.2.1.2. <i>Botón para reinicializar la aplicación (testing)</i>	7
1.2.1.3. Tabla de la agenda generada dinámicamente.....	8
1.2.1.4. Formulario para agregar/eliminar contactos de la agenda.....	8
1.2.2. Fichero <i>funciones.php</i>	9
1.2.2.1. Función <i>botonCerrarSesion()</i>	9
1.2.2.2. Función <i>mostrarAgendaConCamposOcultos(\$agenda, \$titulo = "Mi agenda personal")</i>	10
1.2.2.3. Función <i>validarDatos(&\$agenda, &\$telefonoBorrado, &\$contactoAgregado)</i>	10
1.2.2.4. Función <i>comprobarNombre()</i>	11
1.2.2.5. Función <i>comprobarTelefono(&\$agenda)</i>	11
2. Manual de despliegue para Apache (LAMP).....	12
2.1. Requisitos previos.....	12
2.2. Desplegar aplicación web.....	12

1. Descripción de la solución:

La aplicación consta de diversas funciones que proporcionarán al usuario la funcionalidad de agenda de contactos, de forma que el usuario pueda almacenar un nombre asociado a un número de teléfono.



The screenshot shows the 'AgendAPP de Contactos v2' interface. At the top, there's a title bar with a close button (X). Below the title, a message states 'Actualmente la agenda está vacía'. A section titled 'Registre un nuevo contacto:' contains two input fields: 'Nombre del contacto:' with the value 'Christian' and 'Nº Teléfono:' with the value '612345789'. Below these fields are two buttons: 'Limpiar formulario' (blue) and 'Guardar contacto' (green). At the bottom, a small text line reads 'Desarrollado por: Christian Miguel Giménez Puerma - 2ºDAW'.

1.1. Descripción lógica/funcional de cada componente.

Inicialmente, la aplicación no contará con una agenda, esta estará vacía y se le mostrará al mismo que está vacía. Para rellenarla, el usuario contará con un formulario que permitirá introducir el nombre y el teléfono para posteriormente guardarla, mostrándola si se ha realizado la operación de inserción correctamente.



The screenshot shows the 'AgendAPP de Contactos v2' interface with a populated agenda. The title bar and registration form are identical to the previous screenshot. The agenda section, titled 'Mi agenda personal', displays a table with one contact:

Nombre	Nº Teléfono
Pepe	693829480

Below the table, the registration form is visible. At the bottom, a green message box states 'Se ha agregado/modificado el contacto en la agenda.' The footer text 'Desarrollado por: Christian Miguel Giménez Puerma - 2ºDAW' is also present.

Los datos no se insertarán sin previamente ser validados en el servidor y, si se encuentran las siguientes casuísticas, se le informará al usuario del error/advertencia para que el mismo sea consciente.

1.1.1. Si el usuario introdujo el teléfono pero no un nombre

Se le mostrará al usuario que el campo del nombre no puede estar vacío para que lo sepa y lo corrija.



The screenshot shows the 'AgendAPP de Contactos v2' interface. At the top, there's a title bar with a close button (X). Below it, a section titled 'Mi agenda personal' contains a table with two columns: 'Nombre' and 'Nº Teléfono'. The table has one row with 'Pepe' and '693829480'. Below the table, there's a section titled 'Registre un nuevo contacto:'. It has two input fields: 'Nombre del contacto:' with the value 'Christian' and 'Nº Teléfono:' with the value '123123'. A red error message box is displayed below the name field, stating 'El campo nombre no puede estar vacío.' At the bottom of the form, there are two buttons: 'Limpiar formulario' and 'Guardar contacto'. The footer text reads 'Desarrollado por: Christian Miguel Giménez Puerma - 2ºDAW'.

Nombre	Nº Teléfono
Pepe	693829480

Registre un nuevo contacto:

Nombre del contacto:
Christian

El campo nombre no puede estar vacío.

Nº Teléfono:
123123

Limpiar formulario Guardar contacto

Desarrollado por: Christian Miguel Giménez Puerma - 2ºDAW

1.1.2. Si el usuario introdujo el nombre y el teléfono, y este sí existe en la agenda, se elimina el contacto.

A petición en los requerimientos, cuando un contacto existe en la agenda y no se indica su número de teléfono, este se eliminará de la agenda y se mostrará la advertencia al usuario.



The screenshot shows the 'AgendAPP de Contactos v2' interface. At the top, there's a title bar with a close button (X). Below it, a section titled 'Mi agenda personal' contains a table with two columns: 'Nombre' and 'Nº Teléfono'. The table has two rows: 'Hercules' with '600202023' and 'Christian' with '666333444'. Below the table, there's a section titled 'Registre un nuevo contacto:'. It has two input fields: 'Nombre del contacto:' with the value 'Pepe' and 'Nº Teléfono:' with the value '612345789'. At the bottom of the form, there are two buttons: 'Limpiar formulario' and 'Guardar contacto'. A yellow confirmation message box is displayed below the buttons, stating 'Se ha borrado el teléfono de la agenda.' The footer text reads 'Desarrollado por: Christian Miguel Giménez Puerma - 2ºDAW'.

Nombre	Nº Teléfono
Hercules	600202023
Christian	666333444

Registre un nuevo contacto:

Nombre del contacto:
Pepe

Nº Teléfono:
612345789

Limpiar formulario Guardar contacto

Se ha borrado el teléfono de la agenda.

Desarrollado por: Christian Miguel Giménez Puerma - 2ºDAW

1.1.3. Si el usuario introdujo el nombre y el teléfono, y este no existe en la agenda.

Se le indicará al usuario de que el nombre indicado no está en la agenda, ya que la agenda no acepta que se introduzcan nombres sin teléfonos. Además, si el nombre sí estuviera, se eliminaría el mismo como se especificó en el punto anterior a petición de los requerimientos.



The screenshot shows the 'AgendAPP de Contactos v2' interface. At the top right, there is a red close button with a white 'X'. Below the title, there is a section titled 'Mi agenda personal' containing a table with two columns: 'Nombre' and 'Nº Teléfono'. The table has one row with the values 'Pepe' and '693829480'. Below the table, there is a section titled 'Registre un nuevo contacto:' with two input fields: 'Nombre del contacto:' (containing 'Paco') and 'Nº Teléfono:' (containing '612345789'). A red error message box is displayed below the input fields, stating: 'El nombre indicado no está en la agenda. Por favor, introduzca el nºTeléfono asociado.' Below the error message, there are two buttons: 'Limpiar formulario' (blue) and 'Guardar contacto' (green). At the bottom, it says 'Desarrollado por: Christian Miguel Giménez Puerma - 2ºDAW'.

1.1.5. Botón para restablecer/borrar los datos de la agenda

Además de lo ya comentado, disponemos de un botón rojo de cierre en la esquina superior derecha de la aplicación, la cual restaurará la misma a su punto inicial, borrando todos los datos almacenados en la agenda.



The screenshot shows the 'AgendAPP de Contactos v2' interface. At the top right, there is a red close button with a white 'X', which is highlighted by a magnifying glass. Below the title, there is a section titled 'Actualmente la agenda está vacía'. Below this, there is a section titled 'Registre un nuevo contacto:' with two input fields: 'Nombre del contacto:' (containing 'Christian') and 'Nº Teléfono:' (containing '612345789'). Below the input fields, there are two buttons: 'Limpiar formulario' (blue) and 'Guardar contacto' (green). At the bottom, it says 'Desarrollado por: Christian Miguel Giménez Puerma - 2ºDAW'.

1.2. Descripción técnica de cada componente.

1.2.1. Fichero agenda_v2.php

El fichero **agenda_v2.php** es la presentación principal de la aplicación, conteniendo el código HTML y llamando a las funciones alojadas en **funciones.php** para realizar las funcionalidades principales del programa. A continuación se detallará técnicamente la implementación de cada sección del fichero **agenda_v2.php**

1.2.1.1. Cabecera web, metadatos e importación de funciones.php

La aplicación divide la funcionalidad en dos ficheros, por ello se debe importar el fichero **funciones.php** para el correcto funcionamiento. Además se han dejado comentadas unas líneas de depuración de errores para futuras incidencias.

En la cabecera del documento HTML se encuentran importados los ficheros .css y .png necesarios con rutas relativas.

```
1  <?php
2  //Importación de las funciones
3  include_once "funciones.php";
4  //Descomentar para reportar errores en la aplicación
5  // ini_set('display_errors', 1);
6  // ini_set('display_startup_errors', 1);
7  // error_reporting(E_ALL);
8
9  ?>
10 <!DOCTYPE html>
11 <html lang="es">
12
13 <head>
14     <meta charset="UTF-8">
15     <meta name="viewport" content="width=device-width, initial-scale=1.0">
16     <title>AgendAPP - Christian Gimenez 2DAW</title>
17     <link rel="icon" href="/imgs/ico.png" type="image/png">
18     <link rel="stylesheet" href="/css/estilos.css" type="text/css">
19 </head>
20
```

1.2.1.2. Inicialización de la aplicación y validación de datos de entrada

El programa inicializará el array asociativo **\$agenda** el cual almacenará los datos que lleguen desde el cliente con el formato ["nombre_contacto" => "num_telefono"]. Además se declaran las variables booleanas (flag) para poder mostrar al usuario si el nombre está vacío (**\$errorNombre**), si falta el teléfono y no hay un nombre existente (**\$errorTelefono**), si el teléfono/contacto fue eliminado (**\$telefonoBorrado**) y si el contacto fue agregado exitosamente (**\$contactoAgregado**).

Una vez inicializadas las variables globales, se comprobará si se introdujeron datos por el método POST a través del formulario, si fue así, estos se decodifican con **json_decode()** (método más seguro para des-serializar que **unserialize()**) y se almacenan en **\$agenda**. Después se comprobará si se introdujo el nombre llamando a **comprobarNombre()** ← *seguir enlaces para ver la implementación*, se comprobará también el teléfono con **comprobarTelefono(\$agenda)** ← , y por último, se validarán todos los datos juntos, realizando la inserción o borrado de los datos a la agenda con **validarDatosConCamposOcultos(\$agenda, \$telefonoBorrado, \$contactoAgregado)** ←

```

21 <body>
22 <?php
23 //Si No existe la agenda con campos ocultos, la creamos vacía
24 $agenda = [];
25 //Variables 'flag' para mostrar errores/mensajes/advertencias al usuario
26 $errorNombre = false;
27 $errorTelefono = false;
28 $telefonoBorrado = false;
29 $contactoAgregado = false;
30
31 //Si existe la agenda, volcamos los datos en la misma
32 if (isset($_POST["agenda"])) {
33 //Decodificamos la agenda del input hidden (usamos la función de json porque es más seguro que serialize())
34 $agenda = json_decode($_POST["agenda"], true);
35 //Validamos nada más recargar la página por si hubieran entrado datos por $_POST
36 $errorNombre = comprobarNombre();
37 $errorTelefono = comprobarTelefono($agenda);
38 //Los datos se introducen en la agenda a la vez que se validan y se indica lo ocurrido a las variables 'flag'
39 validarDatosConCamposOcultos($agenda, $telefonoBorrado, $contactoAgregado);
40 }
41 ?>

```

1.2.1.2. Botón para reiniciar la aplicación (testing)


Botón que cerrará la sesión actual, borrando todos los datos almacenados en el *input hidden* así como los valores de **\$agenda**, este botón puede ser quitado si el usuario no está conforma, pero para la fase de testeo facilita el no tener que borrar las *cookies* de sesión para re-inicializar la aplicación y volver a testearla con otros casos de prueba. Implementación: → **botonCerrarSesion()**

```

43 <main>
44 <?php
45 //Botón para cerrar la sesión y así borrar todos los datos almacenados de la agenda
46 botonCerrarSesion();
47 ?>

```

1.2.1.3. Tabla de la agenda generada dinámicamente

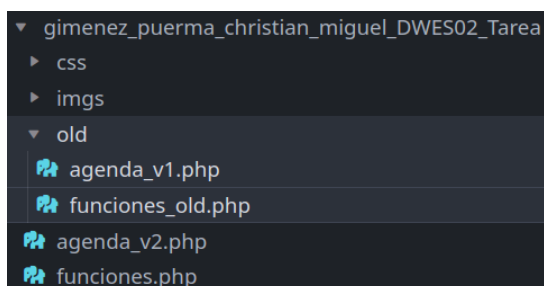
Esta sección se creará de forma automática con los datos que vaya almacenando el usuario en **\$agenda** con la función [mostrarAgendaConCamposOcultos\(\\$agenda\)](#)  , realizándose la acción sólo si la misma dispone ya de datos, si no, se mostrará al usuario de que la agenda actualmente está vacía.

```
48 <header>
49   
50   <h1>AgendAPP de Contactos v2</h1>
51 </header>
52
53 <?php //Si la agenda está vacía se lo indica al usuario
54 if (!empty($agenda)) {
55     mostrarAgendaConCamposOcultos($agenda);
56 } else {
57     ?> <p class="vacía">Actualmente la agenda está vacía</p>
58 <?php
59 }
60 ?>
61 <hr />
```

1.2.1.4. Formulario para agregar/eliminar contactos de la agenda

Por último estará la implementación del formulario securizado (evitando en sus *inputs* y en el *action* del mismo la inserción de código o *Cross Site Scripting* formateando los campos introducidos a caracteres especiales y de escape de HTML con **htmlspecialchars()** el cual mandará al mismo fichero (\$_SERVER["PHP_SELF"]) a través del método POST, el nombre introducido, el n.º teléfono (sólo permite caracteres numéricos) y los datos anteriormente introducidos, los cuales se almacenarán en el *value* del `<input type="hidden">`, serializandose de forma segura con **json_encode()** para guardar los datos y que no se borren al refrescarse la página cada vez que se pulsa el botón de guardar contacto, ya que uno de los requisitos fue que no hubiera persistencia de datos ni en ficheros ni en BBDD.

NOTA: Para la resolución de este punto, realicé otra implementación guardando los datos en la variable de sesión `$_SESSION` y recuperándolos en cada envío, como no estaba seguro de si este paso se consideraría o no persistente, realicé esta segunda versión, pero si se desea comprobar la implementación, se puede revisar el código o desplegar la aplicación v1 que se encuentra en la sub-carpeta *old*:



El formulario además cuenta con código imbricado el cual mostrará al usuario, con distintos colores, los mensajes/advertencias/errores anteriormente citados con respecto a **si falta el campo nombre**, **no existe el nombre en la agenda el cual se desea eliminar**, **si se ha eliminado el contacto indicado** y **si se ha introducido correctamente el contacto en la agenda**.

```

63 <!--Formulario para introducir nombre y teléfono-->
64 <h2>Registre un nuevo contacto:</h2>
65 <form action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>" method="POST">
66   <input type="hidden" value="<?php echo htmlspecialchars(json_encode($agenda)); ?>" name="agenda" />
67   <label for="nombre">Nombre del contacto:</label>
68   <input type="text" name="nombre" placeholder="Christian" />
69   <?php
70   //Si le dieron al botón de guardar y el nombre está vacío, mostramos el error
71   if ($errorNombre) {
72     echo "<span class='error'>El campo nombre no puede estar vacío.</span>";
73   }
74   <?>
75   <label for="telefono">Nº Teléfono:</label>
76   <input type="number" min="0" name="telefono" placeholder="612345789" />
77   <?php
78   //Si le dieron al botón de guardar, el nombre no existe en la agenda y el teléfono está vacío
79   if ($errorTelefono) {
80     //Le pedimos al usuario introducirlo
81     echo "<span class='error'>El nombre indicado no está en la agenda.<br>Por favor, introduzca el nºTeléfono asociado.</span>";
82     $telefonoBorrado = false;
83   }
84   <?>
85   <div class="botones">
86     <input type="reset" value="Limpiar formulario" name="limpiar" />
87     <input type="submit" value="Guardar contacto" name="guardar" />
88     <?php
89     //Si se añadió/modificó correctamente el contacto, lo indicamos
90     if ($contactoAgregado) {
91       echo "<span class='exito'>Se ha agregado/modificado el contacto en la agenda.</span>";
92     }
93     //Si le dieron al botón de guardar, y se borró un teléfono, lo indicamos
94     } else if ($telefonoBorrado) {
95       echo "<span class='warning'>Se ha borrado el teléfono de la agenda.</span>";
96     }
97     <?>
98   </div>
99 </form>

```

1.2.2. Fichero *funciones.php*

El fichero **funciones.php** contiene las funcionalidades principales de la aplicación, para entender mejor el contexto de dónde se implementa cada una de ellas, recomiendo leer los apartados anteriores y hacer **click** en los enlaces donde se llama a estas funciones.

1.2.2.1. Función *botonCerrarSesion()*

Esta función consta de un pequeño formulario con un sólo input, el cual si se hace click, el fichero lo reconocerá y llamará a **session_destroy()** para limpiar los datos de **\$_SESSION**, reenviará al usuario a la página de nuevo con **header()** y se llama a **die()** para salir correctamente de esta función , ya que en el proceso de desarrollo, la aplicación se quedaba colgada y no se reenviaba correctamente la página, por lo que investigando di con esta función que resolvió esa incidencia.

```

3 //Crea el botón para cerrar la sesión y borrar los datos
4 function botonCerrarSesion()
5 {
6   <?>
7   <!--Botón para destruir la sesión actual y eliminar todos los datos almacenados -->
8   <form action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>" method="POST">
9     <input type="submit" name="cerrar" title="Cierra la sesión actual y reinicializa los datos (Borra la agenda)" value="X"></input>
10  </form>
11  <?php
12  if (isset($_POST["cerrar"])) {
13    session_destroy();
14    header("Location: " . $_SERVER["PHP_SELF"]);
15    die();
16  }
17 }

```

1.2.2.2. Función *mostrarAgendaConCamposOcultos(\$agenda, \$titulo = "Mi agenda personal")*

Esta función crea la vista de la tabla de la agenda personal, recorriendo el array asociativo **\$agenda**, de modo que el usuario pueda ver de forma gráfica que se han introducido correctamente los contactos. Como puede observarse, en la implementación las filas y columnas de las tablas son generadas dinámicamente. El segundo argumento por defecto no se ha implementado en la resolución final por falta de tiempo, pero inicialmente serviría para que el usuario al iniciar a la aplicación, introdujese un nombre personalizado a la agenda.

```

20 //Muestra la agenda del input hidden al usuario
21 function mostrarAgendaConCamposOcultos($agenda, $titulo = "Mi agenda personal")
22 {
23     ?>
24     <table>
25         <caption><?php echo $titulo; ?></caption>
26         <tr>
27             <th>Nombre</th>
28             <th>Nº Teléfono</th>
29         </tr>
30     <?php
31     foreach ($agenda as $nombre => $telefono) {
32         echo "<tr>";
33         echo "<td>$nombre</td><td>$telefono</td>";
34         echo "</tr>";
35     }
36     echo "</table>";
37 }

```

1.2.2.3. Función *validarDatos(&\$agenda, &\$telefonoBorrado, &\$contactoAgregado)*

Todos los argumentos de esta función se pasan por referencia para modificar, respectivamente, el array asociativo **\$agenda** y las variables ‘flag’ **\$telefonoBorrado** y **\$contactoAgregado**, todo ello se realizó para poder modificar el array que almacena los datos de los contactos, y tratar de no repetir condicionales para poder mostrar al usuario las advertencias pero pudiendo imbricar el código de las mismas en el mismo formulario, debajo de los input. Como los datos deben ser validados al principio del documento .php y esto serán tratados al momento de ser recibidos (insertando/modificando/borrando los contactos), tampoco se podía volver a evaluar los condicionales en el formulario, ya que los datos ya habrían sido tratados y no se mostraban las advertencias adecuadamente.

El flujo de condicionales trata de cumplir los requerimientos pedidos y los comentarios de código tratan de aclarar lo que se realiza en cada paso de los mismos.

Como se puede observar, se limpian los datos eliminando los espacios sobrantes con **trim()** y securizando la entrada de nuevo con **htmlspecialchars()**.

```

40 //Valida los datos introducidos por el usuario, se le pasa agenda y las variables 'flag' por referencia
41 function validarDatosConCamposOcultos(&$agenda, &$telefonoBorrado, &$contactoAgregado)
42 {
43     //Si se le dió al botón de Guardar
44     if (isset($_POST["guardar"])) {
45         //Si se introdujo nombre por el formulario
46         if (isset($_POST["nombre"]) && !empty($_POST["nombre"])) {
47             $nombre = trim(htmlspecialchars($_POST["nombre"]));
48             //Si se introdujo n°tlfn en el formulario
49             if (isset($_POST["telefono"]) && !empty($_POST["telefono"])) {
50                 $telefono = trim(htmlspecialchars($_POST["telefono"]));
51                 //Evitamos que entren cadenas vacías después de limpiar los datos que entren
52                 if ($nombre !== "") {
53                     /*Si el nombre no existe en la agenda,
54                     y el n°tlfn no está vacío, se añade a la agenda
55                     Pero si ya existe el nombre, simplemente se sustituye su n°tlfn*/
56                     $agenda[$nombre] = $telefono;
57                     $contactoAgregado = true;
58                 }
59             } else {
60                 /*Si el nombre ya existe en la agenda,
61                 y el n°tlfn está vacío, se eliminará de la agenda ese contacto*/
62                 if (!array_search($nombre, $agenda)) {
63                     unset($agenda[$nombre]);
64                     $telefonoBorrado = true;
65                 }
66             }
67         }
68     }
69 }

```

1.2.2.4. Función comprobarNombre()

Esta función comprobará si se introdujo bien el nombre para modificar la variable booleana \$errorNombre al principio del programa.

```

72 //Comprueba si se introdujo el campo nombre en el formulario
73 function comprobarNombre()
74 {
75     if (isset($_POST["guardar"]) && empty($_POST["nombre"])) {
76         return true;
77     }
78     return false;
79 }

```

1.2.2.5. Función comprobarTelefono(&\$agenda)

Esta función pasa por referencia el array **\$agenda** para poder comprobar si se introdujo un nombre sin teléfono con la intencionalidad de borrar uno existente, o si se introdujo por error uno no existente sin teléfono, evitando que se añadan nombres a la agenda que no tengan un teléfono asociado, modificando así el valor booleano de \$errorTelefono.

```

82 //Comprueba si se introdujo el teléfono con o sin la intención de borrar un nombre
83 function comprobarTelefono(&$agenda)
84 {
85     if (
86         isset($_POST["guardar"]) && !empty($_POST["nombre"]) &&
87         empty($_POST["telefono"]) && !array_key_exists($_POST["nombre"], $agenda)
88     ) {
89         return true;
90     }
91     return false;
92 }

```

2. Manual de despliegue para Apache (LAMP).

2.1. Requisitos previos

Para desplegar la aplicación es necesario contar con el Stack LAMP (Linux, Apache, MySQL/MariaDB y PHP), siendo necesario tener instalado en un S.O. Linux el servidor web apache2 con la lista de repositorios previamente actualizadas.

```
sudo apt update && sudo apt install apache2
```

El gestor de bases de datos mysql-server o mariadb-server (aunque para la versión actual de la aplicación no sería necesario un SGBD, ya que la aplicación no tiene actualmente persistencia de datos):

```
sudo apt install mariadb-server
```

o:

```
sudo apt install mysql-server
```

Y el intérprete de **PHP versión 8.3** y el módulo mod_php de php (**libapache2-mod-php**):

```
sudo apt install php-8.3 libapache2-mod-php
```

Si están estos requisitos ya cumplidos, se recomienda habilitar y comprobar el estado de los mismos con los siguientes comandos:

```
sudo systemctl enable apache2
```

```
sudo systemctl status apache2
```

```
sudo systemctl enable mysql o sudo systemctl enable mariadb
```

```
sudo systemctl status mysql o sudo systemctl status mariadb
```

```
php --version
```

Así como recomendablemente tener configurado el sitio de Apache personalizado donde se desee servir la aplicación.

2.2. Desplegar aplicación web

Si se cumplen los requisitos anteriormente citados, para desplegar la aplicación debe descomprimir el fichero **.zip**

```
sudo unzip gimenez_puerma_christian_miguel_DWES02_Tarea.zip
```

Copiar los ficheros desde la carpeta actual en la que haya descomprimido los ficheros, a la carpeta del site de Apache (en la variables de bash `${SITE}`), inserte el nombre que se le ha dado a la carpeta del DocumentRoot creada en Apache previamente, por ejemplo: `/var/www/agendapp`)

```
sudo cp -r ./gimenez_puerma_christian_miguel_DWES02_Tarea/*  
/var/www/${SITE}
```

Después, reinicie y compruebe el estado del servicio de Apache:

```
sudo systemctl restart apache2
```

```
sudo systemctl status apache2
```

Si todo fue correctamente, ahora debería poder visualizar la aplicación en el dominio configurado en el servidor de apache, el cual será el ServerName configurado en el fichero de configuración del sitio personalizado de Apache (por ej: www.agendapp.com). Si no se realizó todo ese proceso y se desea testear la aplicación en un entorno de desarrollo y no desplegarle en un entorno de ejecución, bastaría con desplegar la aplicación en el sitio por defecto de apache /var/www/html:

```
sudo cp -r ./gimenez_puerma_christian_miguel_DWES02_Tarea/*  
/var/www/html
```

Borrar el index.html por defecto de Apache:

```
sudo rm /var/www/html/index.html
```

Comprobar que en el fichero de configuración por defecto de Apache se encuentre la directiva Indexes:

```
sudo cat /etc/apache2/apache.conf
```

```
<Directory /var/www/>  
    Options Indexes FollowSymLinks  
    AllowOverride None  
    Require all granted  
</Directory>
```

Y acceder a través del navegador web a **localhost/agenda_v2.php**

