INSTITUTO DE EDUCACIÓN SECUNDARIA
I.E.S. CONSELLERIA - FAMILIA PROFESIONAL DE INFORMATICA Y COMUNICACIONES

RaspBerry Pi 3

PROYECTO DE ADMINISTRACIÓN
DE SISTEMAS EN RED
NÚMERO INF-N-12

Autor
Alberto Sánchez Ruiz
Tutor individual
Julio Contelles

Índice

Objetivo

Sustituir el ordenador que gestiona el sistema de audio por una RaspBerry pi 3 debido a su reducido coste y a sus posibilidades para configurar, crear un entorno web para gestionar las canciones que sonarán en cada descanso y configurar e instalar lo necesario para que sea accesible desde cualquier ordenador del centro por una serie de usuarios.

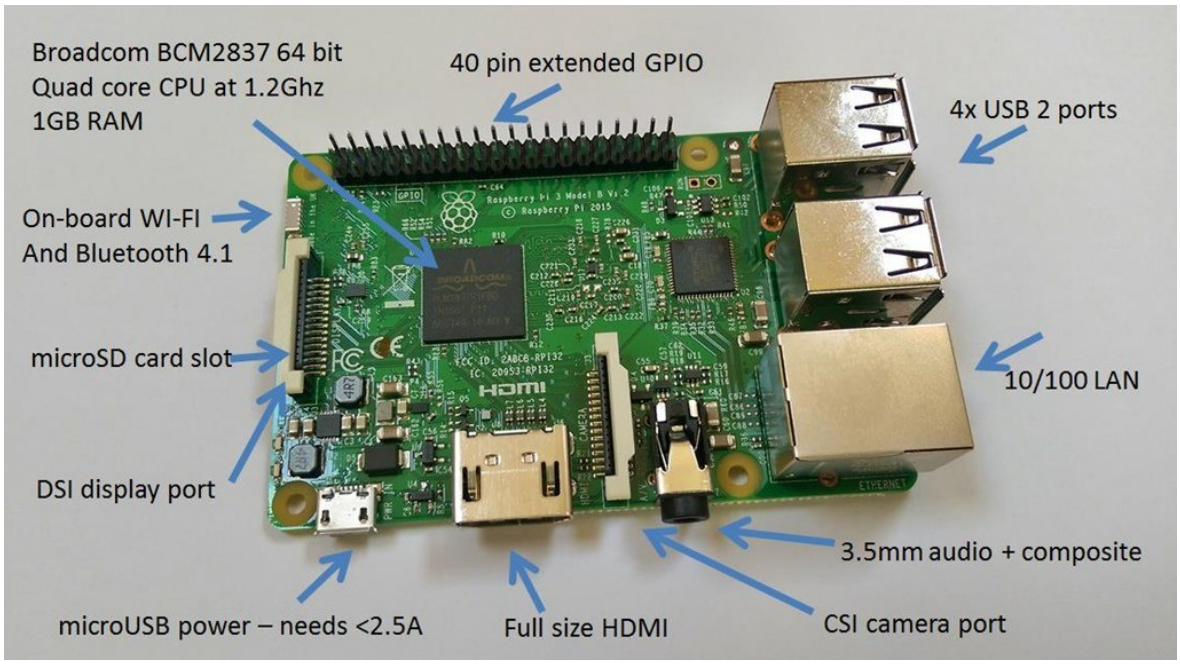
Introducción

Hardware

Se dispone de un sistema de sonido conectado a un ordenador con un sistema Debian y un fichero crontab programado para ejecutar un script, que reproducirá el audio correspondiente a la hora de los comienzos, descansos y finales de clases.

Para realizar el proyecto se dispone del siguiente hardware:

1 x RaspBerry pi 3:



Se ha elegido la raspberry pi 3 para sustituir el ordenador actual, ya que dispone de un hardware bastante capaz y a la vez dispone de una distribución Debian adaptada al tipo de procesador que monta siendo la mejor opción para realizar el proyecto puesto que es totalmente compatible con los programas que se van a usar y no tendrá ningún problema de rendimiento.

Entre sus especificaciones encontramos:

- Procesador:

  1,2 GHz de cuatro núcleos ARM Cortex-A53

- GPU

  1080p30 .

- RAM:

  1GB LPDDR2.

- Conectividad

  Ethernet socket Ethernet 10/100 BaseT

  802.11 b / g / n LAN inalámbrica y Bluetooth 4.1 (Classic Bluetooth y LE)

  Salida de vídeo HDMI y RCA compuesto (PAL y NTSC)

  Salida de audio jack de 3,5 m

  4x Puertos USB 2.0

  Conector GPIO

  Conector de la cámara de 15 pines cámara MIPI interfaz en serie (CSI-2)

  Ranura de tarjeta Micro SD

1 x Tarjeta Micro SD 64Gb

Sistema operativo

El sistema operativo seleccionado es Raspbian, que es un port basado en Debian Wheezy (Debian 7.0) y por lo tanto softeware libre(GPL), adaptado al procesar ARMv6 que monta la RaspBerry pi 3.

Para poder instalar el sistema el primer paso es montar la imagen del sistema Raspbian en la tarjeta SD, dependiendo del sistema operativo puede ser:

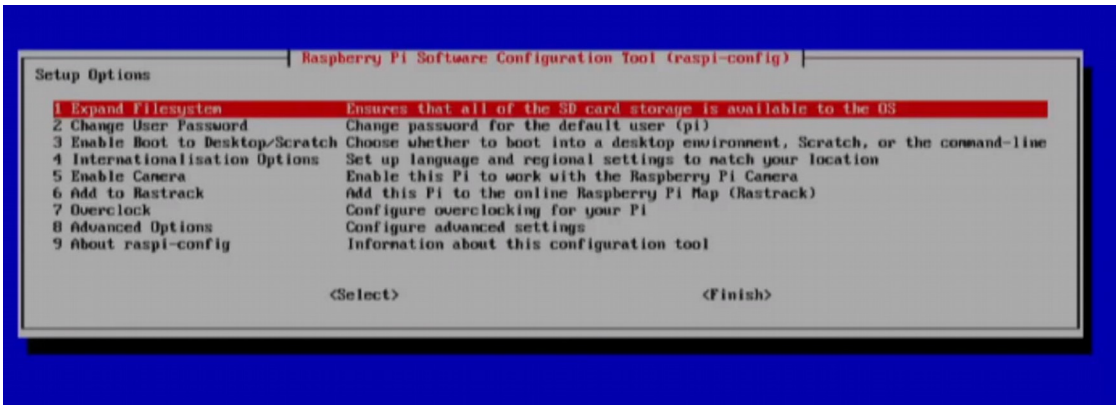
GNU/Linux= dd if=/Rasbian/ of=/destino/micro/sd

Windows= Existen varios programas como por ejemplo Rufus.

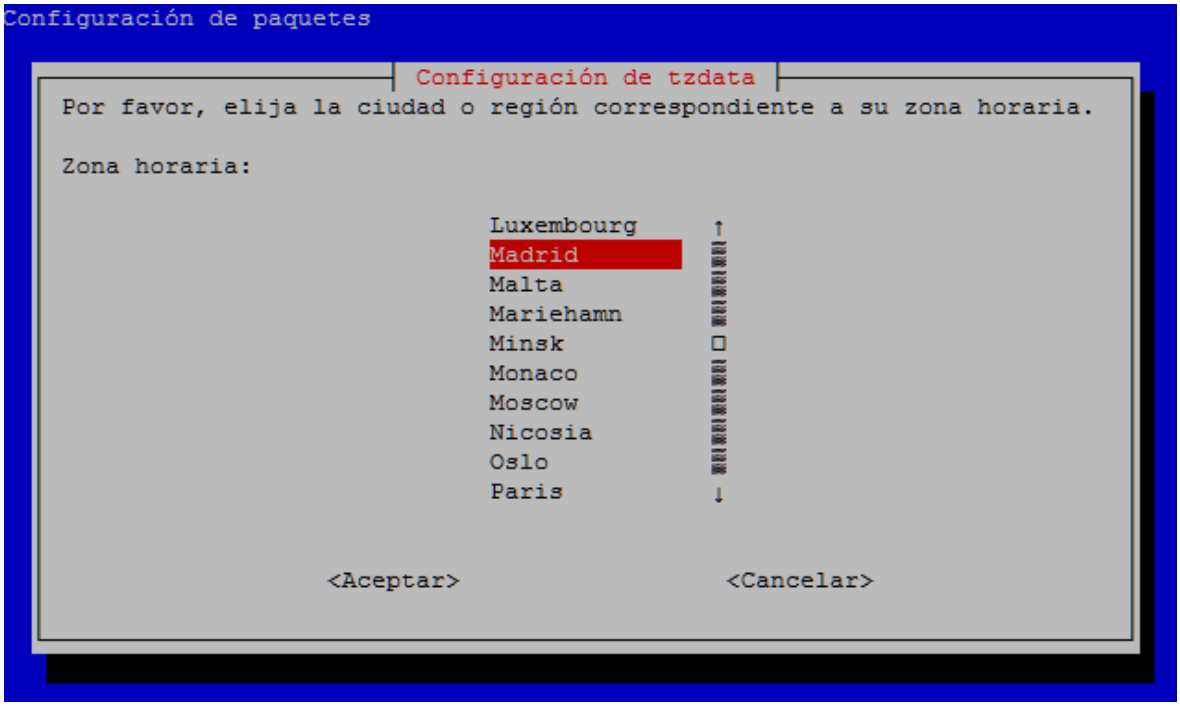
Instalación

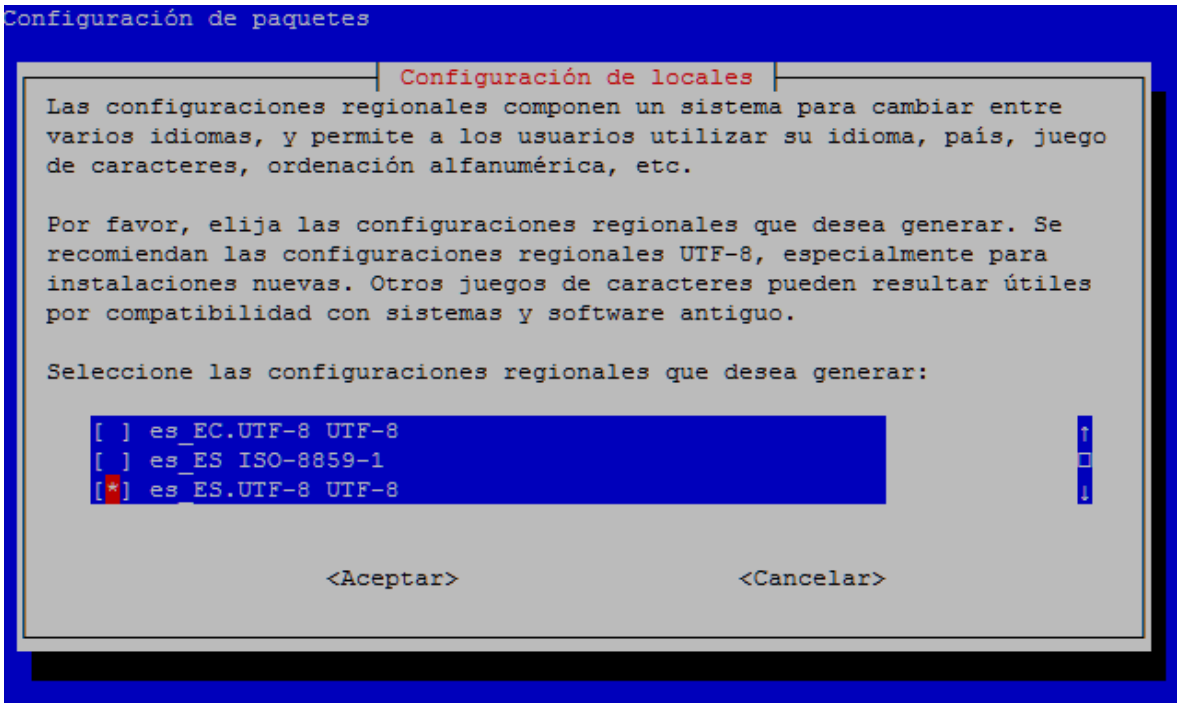
En la primera pantalla de configuración que nos aparece tenemos que seleccionar:

1.Expand Filesystem; la cual configurará la partición de root para poder aprovechar la tarjeta SD debidamente.



4.Internationalisation options: para cambiar la configuración del teclado que se vaya a conectar, el idioma y la hora del sistema.



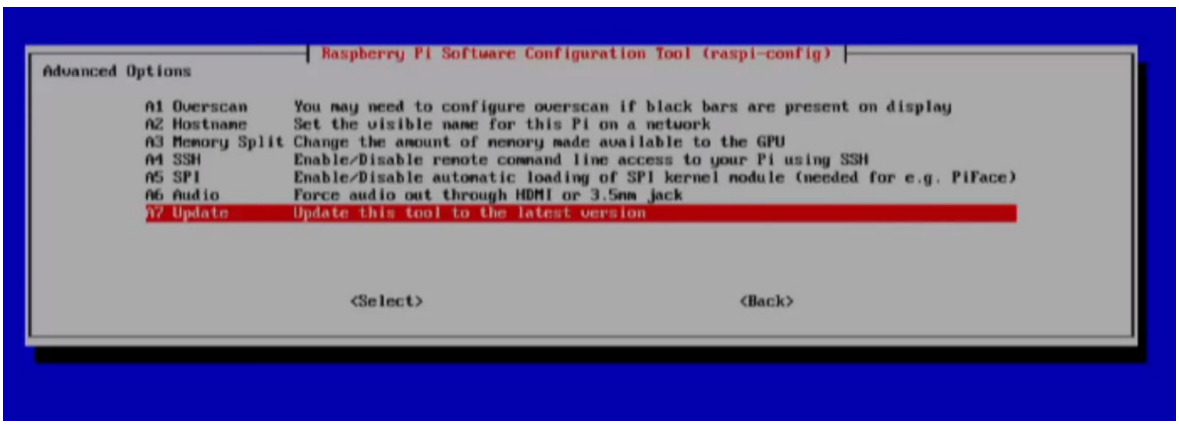


Y opcionalmente:

3.Enable boot to Desktop/Scratch: para que el sistema inicie directamente a un entorno de consola de comandos y no cargue en entorno gráfico.

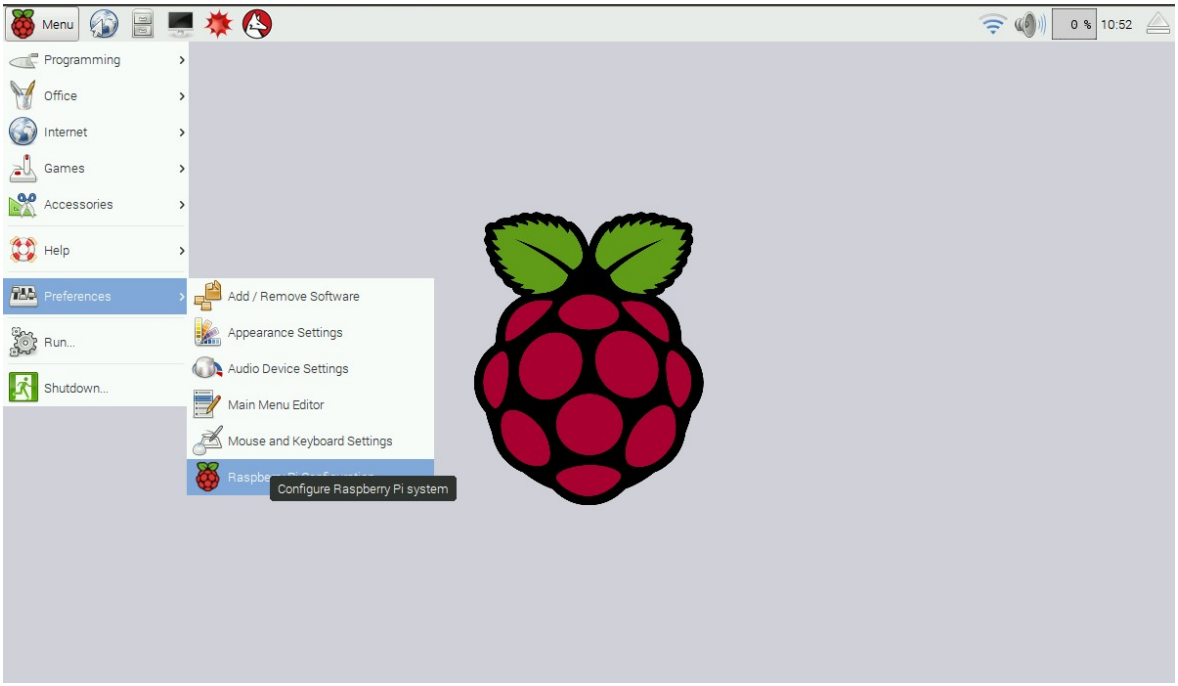
SSH

Para realizar las pruebas se configuró el acceso remoto a la RaspBerry pi por SSH desde la misma pantalla de configuración anterior, en el apartado de *Advanced Options(A4 SSH)*:



Tras realizar la configuración el sistema se reiniciará, una vez en el escritorio se usaron los comando ***sudo apt-get update* y *sudo apt-get upgrade*** para actualizar el sistema a la última versión y evitar fallos de compatibilidad.

La configuración de red está configurada por defecto para coger una dirección Ip por **DHCP** pero tambien se nos permite conectarnos por WiFi, en este caso y para facilitar las pruebas dejaremos la configuración por defecto.

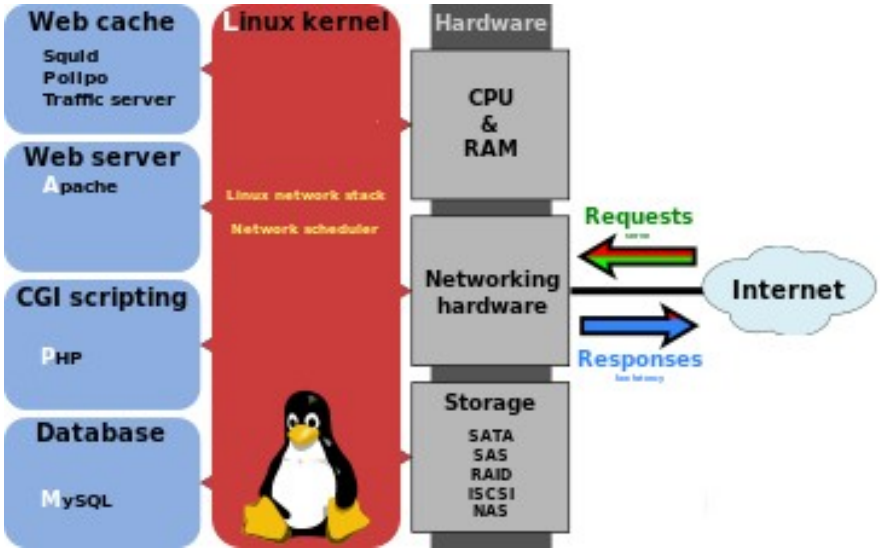


La última instalación requerida será el paquete con el programa necesario para reproducir la música, el paquete a instalar se llama ***SOX,*** su instalación junto a las librerias necesarias para reproducir los archivos ***MP3, MP2 o WAV (Que son los más usados, aunque el comando permite más tipos de archivos de audio)*** seria el siguiente:

$ sudo apt-get install sox libsox-fmt-all

Software y configuración (entorno LAMP)

Se trata de una agrupación de servicios en los cuales están conformados en sus siglas Apache servidor web, MYSQL base de datos y PHP lenguaje de programación el cual se ha instalado para alojar nuestra aplicación web.



La “L” de las siglas LAMP se refiere al sistema donde va ha estar instalado, en este caso un sistema GNU/LINUX (RaspBian), en el esquema se resume el funcionamiento de este, en el cual los bloques azules corresponderían al sistema LAMP el cual funciona gracias al sistema donde está instalado (Bloque rojo) y por ultimo el kernel del sistema se comunica con el hardware (Bloque gris) del dispositivo para responder a las solicitudes de los clientes que intenten acceder a nuestra aplicación.

Apache

El servidor web apache es el que se ha instalado para alojar la página web ya que es un servidor web HTTP de código abierto, para plataformas Unix.

Tras instalar el servidor Apache*(sudo apt-get install apache2)*, el directorio /var/www/html/ es donde están ubicados los archivos *PHP* del sitio web, ya que es el directorio que Apache tiene configurado por defecto y no necesita configuración adiciona.

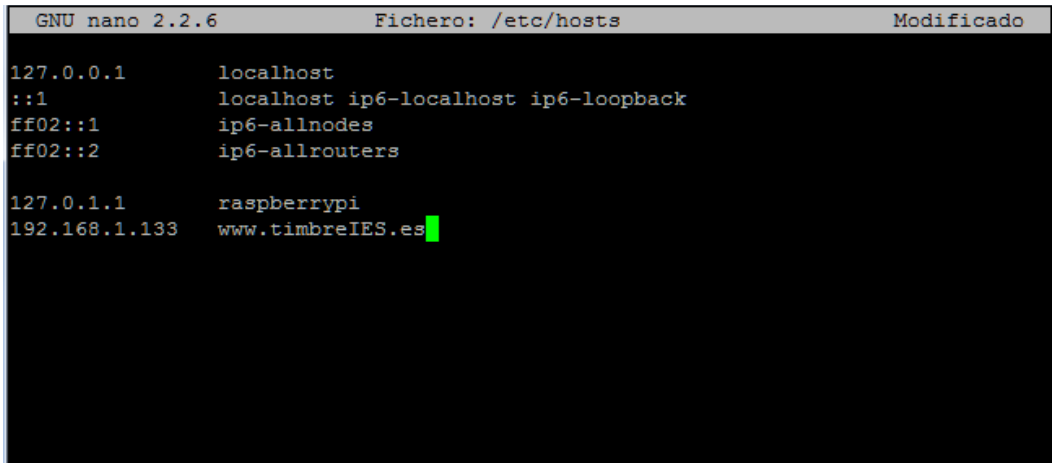
El archivo con nombre ***index.php*** será el primer archivo en mostrarse al entrar a la página si tiene permisos de ejecución *(Sudo chmod o+x index.php)* y ya que la pagina web va a trabajar moviendo y copiando ciertos archivos, el usuario propietario de los archivos y directorios que vamos a usar debe de ser ***WWW-DATA*** *(Sudo chown -R www-data /var/www/html/)*que que apache tiene configurado para gestionar estas funciones.

```
pi@raspberrypi:~ $ ls -l /var/www/html/
total 32
drwxr-xr-x 3 www-data pi       4096 may 11 18:41 aspecto
drwxr-xr-x 2 www-data pi       4096 may 11 18:41 descartes
-rwxr-xr-x 1 www-data www-data 4676 may 13 10:37 funciones.php
-rw-r--r-x 1 www-data pi       5331 may 13 12:38 index.php
-rw-r--r-- 1 www-data pi         21 may 11 18:41 phpinfo.php
-rwxrwx--- 1 www-data www-data  795 may 13 11:28 scriptmusica.sh
pi@raspberrypi:~ $
```

Adicionalmente y para que no haya problemas de permisos el usuario ***WWW-DATA*** anteriormente nombrado debe de estar en los grupos de audio y video ya que desde la aplicación web se va a ejecutar un comando para reproducir multimedia (En este caso *play*) y el sistema instalado comprobará antes de reproducir los archivos si el usuario que lo esta ejecutando está en dichos grupos.

```
pi@raspberrypi:~ $ sudo adduser www-data audio
Añadiendo al usuario `www-data' al grupo `audio' ...
Añadiendo al usuario www-data al grupo audio
Hecho.
pi@raspberrypi:~ $ sudo adduser www-data video
Añadiendo al usuario `www-data' al grupo `video' ...
Añadiendo al usuario www-data al grupo video
groupsHecho.
pi@raspberrypi:~ $ groups www-data
www-data : www-data audio video
pi@raspberrypi:~ $ groups pi
pi : pi adm dialout cdrom sudo audio www-data video plugdev games users input ne
tdev spi i2c gpio
pi@raspberrypi:~ $ sudo /etc/init.d/apache2 restart
[....] Restarting apache2 (via systemctl): apache2.serviceWarning: Unit file of
apache2.service changed on disk, 'systemctl daemon-reload' recommended.
. ok
```

Para acceder bastaría con introducir en el navegador la **dirección ip** de la RaspBerry y aun que no disponemos de servidor DNS, podemos editar el archivo /etc/hosts y añadir la dirección IP del servidor apache (En este caso la de nuestra RaspBerry)e indicar el nombre, tal que así:



MySQL

MySQL (sudo apt-get install mysql-server-php5 mysql) es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial el cual se ha utilizado para gestionar el acceso a la pagina web, configurando una base de datos para que únicamente los usuarios registrados con un nombre de usuario y una contraseña puedan acceder, ya que de otra manera cualquiera que conociese la dirección *IP* de la RaspBerry podría acceder.

Adicionalmente hay otra base de datos donde se guardaran los nombres de las ultimas canciones establecidas en el entorno web. Quedando las tablas de la base de datos de la siguiente manera:

```
+-------------+-----------------------------------+-------------+
| Primera     | Patio                             | Ultima      |
+-------------+-----------------------------------+-------------+
| LaGente.mp3 | K-OtixWorldRenown-Instrumental.mp3 | LaGente.mp3 |
+-------------+-----------------------------------+-------------+
1 row in set (0.00 sec)

mysql> select * from usuarios
    -> ;
+--------+------+
| nombre | pwd  |
+--------+------+
| admin  | 1234 |
+--------+------+
1 row in set (0.00 sec)
```

PHP

PHP es el lenguaje de programación en el que va a estar escrita nuestra aplicación web ya que forma parte de nuestro sistema LAMP..

Tras instalar php y sus módulos para apache*(sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql)*, hay que configurar el archivo *dir.conf* (/etc/apache2/mods-enabled/dir.conf) para que apache busque en primer lugar el archivo *index.php* en lugar de *index.html* del directorio del sitio web por defecto donde se encuentran los archivos de la página web:

```
<IfModule mod_dir.c>
        DirectoryIndex index.php index.html index.xhtml
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Por último y para facilitar las pruebas y ver en pantalla los posibles fallos que surjan con el código de php se puede modificar las sentencias del archivo */etc/php5/apache2/php.ini*:

1. error_reporting = E_ALL

2. display_errors = On

Por ultimo la sentencia *disable_function* del archivo anterior desactiva una serie de funciones que se pueden usar en PHP y para poder utilizar la función *exec()* habia que borrarla de la lista.

*Tras modificar cualquier archivo de configuración de Apache es necesario un reinicio del servicio para que los cambios se hagan efectivos `sudo service apache2 restart`

Solución web

Script automatizado

Para poder reproducir de manera automática la música que sonará en cada hora se ha modificado el script original que ejecutaba el *Crontab* al cual se le pasan dos parámetros uno para indicar la canción a reproducirlos comienzos de clase reproducirá el archivo ***PRIMERA.mp3,*** en los descansos y patios el archivo ***PATIOS.mp3*** y al finalizar las clases el archivo ***ULTIMA.mp3.***

```
# Ruta al ficheros de sonido del timbre
PATIO=/var/www/musicaenuso/PATIOS.mp3 #Descansos
PRIMERA_HORA=/var/www/musicaenuso/PRIMERA.mp3
ULTIMA_HORA=/var/www/musicaenuso/ULTIMA.mp3
```

Por motivos de seguridad la carpeta de ***MusicaEnUso*** se encuentran fuera del directorio donde la página web se aloja para ya que tanto el script como la carpeta tienen que tener permisos de ejecución.

```
drwxrwx--- 2 www-data www-data 4096 may 11 18:35 musicaenuso
drwxrwx--- 2 www-data www-data 4096 may 13 10:26 musicasubida
-rwxrwx--- 1 www-data www-data  810 may 15 19:05 scriptmusica.sh
pi@raspberrypi:/var/www $
```

En primer lugar mediante la función ***suena_timbre()*** comprobaremos si el segundo parámetro que le pasemos al script es “0” o “1” para agregarle una amplificación de 10db o no, activar el sistema de sonido y usando el comando ***play***:

```
function suena_timbre{
        # Activa la tarjeta de sonido
        modprobe snd_hda_intel

        # Espera unos segundos
        sleep 5

        # Pone el volumen al máximo
        amixer sset Master 100%

        # Reproduce el sonido comenzando desde el segundo 0 hasta el 90
        if [ $2 = "1" ]; then
                play $1 vol 10dB trim 0 90
        else
                play $1 trim 0 90
        fi

        # Baja el volumen
        amixer sset Master 0%


        # Desactiva la tarjeta de sonido
        modprobe -r snd_hda_intel
}
```

A continuación mediante un ***case*** haremos una llamada a la función anterior comprobando el valor del primer parámetro y agregándole el directorio a la canción correspondiente:

```
case $1 in
        "primera")
                #A primera hora
                suena_timbre $PRIMERA_HORA $2
        ;;
        "ultima")
                #A última hora
                suena_timbre $ULTIMA_HORA $2
        ;;
        "patio")
                #Patios
                suena_timbre $PATIO $2
        ;;
        *)
                #Reproducir ahora
                play $1 $2
        ;;
esac
```

Por último estas son algunas sentencias para ejecutar el script mediante el *crontab*

```
55 07 * 1,2,3,4,5,6,9,10,11,12 1-5 bash -l /var/www/scriptmusica.sh primera 0
00 08 * 1,2,3,4,5,6,9,10,11,12 1-5 bash -l /var/www/scriptmusica.sh patio 1
55 08 * 1,2,3,4,5,6,9,10,11,12 1-5 bash -l /var/www/scriptmusica.sh patio 1
35 19 * 1,2,3,4,5,6,9,10,11,12 1-4 bash -l /var/www/scriptmusica.sh patio 1
10 20 * 1,2,3,4,5,6,9,10,11,12 5 bash -l /var/www/scriptmusica.sh ultima 0
30 20 * 1,2,3,4,5,6,9,10,11,12 1-4 bash -l /var/www/scriptmusica.sh patio 1
20 21 * 1,2,3,4,5,6,9,10,11,12 1-4 bash -l /var/www/scriptmusica.sh ultima 0
35 21 * * * poweroff
```

Descripción de la aplicación

La página web se estructura en dos archivos principales, uno llamado *index.php* que alberga la totalidad de la página web mostrando los formularios correspondientes si se cumples ciertas condiciones además gestiona la copia y subida de los archivos de música con los que se va a trabajar.

Las funciones que se van a usar en el archivo *index.php están* recogidas en el archivo *funciones.php* e incluidas mediante la sentencia *include('funciones.php');* al principio del documento, de esta manera podremos organizar mejor el código para que pueda ser mas legible en caso de necesitar alguna modificación, además de esta forma se podrá aprovechar el código que se usa en las funciones para otras aplicaciones o incluso cambiar el dispositivo donde se aloja la web con más facilidad

Inicio de sesión:

Lo primero con lo que nos encontramos al entrar a la página es el formulario de inicio de sesión en el cual se comprobará si el nombre de usuario y la contraseña introducidas están en la base de datos y declarar dos variables de sesión con esos datos para su posterior uso. El código PHP es el siguiente:

```php
function login()
    {
    if (!isset($_REQUEST['Enviar']))
        {
        $pagina = "<form action='index.php' method='post'>
Usuario: <input type='text' name='Usuario'><br>
Contraseña: <input type='password' name='pswd'>
<input type='submit' name='Enviar' value='Entrar'>
</form>";
        echo $pagina;
        }
    else
        {
        if (isset($_REQUEST['Enviar']))
            {
            $Usuario = $_REQUEST['Usuario'];
            $pswd = $_REQUEST['pswd'];

            $cond = "SELECT * FROM usuarios where nombre='" . $Usuario . "' and pwd='" . $pswd . "';";
            $result = consulta($cond);

            if ($row = mysqli_fetch_array($result))
                {
                $_SESSION['usuario_valido'] = $Usuario;
                $_SESSION['pswd'] = $pswd;
                header('Location:index.php');
                }
            else
                {
                echo "usuario incorrecto";
                echo "<a href='index.php'> Volver a intentar";
                }
            mysqli_free_result($result);
            mysqli_close($link);
            }
        }
    }
```

En primer lugar comprobamos con un *if* si está declarada la variable de sesión "usuadio_valido" (Significa que esa variable está disponible en cualquier parte del script mientras la sesión este activa) para que en caso contrario *(else)* se ejecute la función *login()* para mostrar el formulario de inicio de sesión:

```php
if (isset($_SESSION['usuario_valido'])){

    if (isset($_POST['Enviar']))
        {
        subida();
        }

    if (isset($_POST['Subir']))
        {
        seleccion();
        }
```

Adicionalmente si el usuario que se ha introducido en el formulario coincide con el usuario *Admin* y con su contraseña se mostrará el botón de agregar un nuevo usuario a la base de datos.

```php
if ($_SESSION['usuario_valido']=="admin" && $_SESSION['pswd']=="1234"){//Comprobar usuario para mostrar botón de agregar
    echo "<hr>";
    echo "<legend>Añadir usuario</legend>";
    echo '<button id="boton_nuevo_usuario" onclick="mostrar();">Pulsa para añadir</button>';
}
```

Para que el botón tuviera la función de ocultarse en caso de ser pulsado, se ha usado la sentencia *Button y onclick* que pertenecen al lenguaje ***Java Script*** que ejecutaría la función ***mostrar()*** situada en la etiqueta ***<script>*** dentro del <!DOCTYPE html>

```html
<!DOCTYPE html>
<html>
<head>
<link rel='stylesheet' href='aspecto/estilo.css' type='text/css'>
<script>
    function mostrar()
    {
        document.getElementById("nuevo_usuario").className="visible";//Ocultar boton "Agregar usuario"
        document.getElementById("boton_nuevo_usuario").className="oculto";//Hacer visible formulario
    }
</script>
```

Mediante la sentencia ***document.getElementById("Nombre del elemento")*** recuperamos el objeto que en este caso queremos cambiar, y con ***.className="visible u oculto"*** establecemos el parámetro ***class*** al nuevo valor para que se muestre (Formulario) y se oculte (Botón "agregar usuario")

***Usuario admin:** El usuario *Admin* tiene la función extra en la página web permitiendo a este agregar nuevos usuarios a la base de datos para facilitar su gestión. La contraseña establecida y el nombre de este usuario son genéricos para facilitar las pruebas, pero por motivos de seguridad se recomienda cambiarlo.

Formularios y funciones

A continuación se va explicar los puntos más relevantes de cada formulario además de como realizan sus funciones.

Subir archivos

En el formulario de subir archivos hay que incluir dos lineas que nos permitirán, en una indicar que en este formulario se van a subir archivos (*enctype='multipart/form-data'*) y en la segunda linea la cual estará oculta (*hidden*) indicaremos el tamaño máximo de los archivos que se vayan a subir en bytes, en este caso seria como máximo 10Mb

```
echo "<form action='index.php' method='post' enctype='multipart/form-data'>";//SUBIR MÚSICA
echo "<input type='hidden' name='max_file_size' value'1024000'>";//Tamaño máximo permitido
```

Una vez seleccionado el archivo que se va a subir y se ha pulsado el botón de subir este hará una llama a la ***funcion seleccion():***

En primer lugar se declara una variable que contendrá un array de los tipos de archivo de audio permitidos para después mediante la funcion ***in_array()*** comparar si la extensión del archivo ([‘***type’***])subido coincide con algún parámetro del array.

```
$file_mimes = ['audio/mpeg', 'audio/wav', 'audio/x-wav', 'audio/mp3', 'audio/mpeg3', 'audio/x-mpeg-3'];
$resultado = "";
if (in_array($_FILES['musica']['type'], $file_mimes))//Comprobación de la extensión del archivo a subir
```

En caso de que la extensión del archivo sea permitido, comprobará si el archivo de música se ha subido (***is_upload_file***) en el caso de que sea asi establecemos una variable con la función ***finfo_open*** esto extraerá el tipo de extensión ***Mime type*** del archivo para comprobar si realmente es un archivo de audio independientemente de la extensión que tenga, así evitaremos que se pueda engañar a la aplicación cambiado la extensión del archivo a subir.

Después comprobamos si el ***Mime type*** tiene el formato correcto comparando la información extraída ***(finfo_file)*** de los archivos a subir con nuestro ***array***

```php
if (is_uploaded_file($_FILES['musica']['tmp_name']))
        {

        $finfo = finfo_open(FILEINFO_MIME_TYPE); // devuelve el tipo mime de su extensión

        if (in_array(finfo_file($finfo, $_FILES['musica']['tmp_name']), $file_mimes))
                {
```

Por ultimo y en caso que tanto la extensión como el tipo ***mime*** de los archivos sean los correctos, asignamos una variable con el nombre real del archivo mediante la función de PHP ***basename()*** ya que hasta ahora hemos trabajado con el nombre temporal del archivo, al mover el archivo se hará una ultima comprobación para asegurarnos que el archivo se ha movido correctamente, de la carpeta temporal (tmp_name) al directorio **MusicaSubida** con el nombre original del archivo y en caso contrario que nos muestre un error.

```php
$Nombremp3 = basename($_FILES['musica']['name']);
if (move_uploaded_file($_FILES['musica']['tmp_name'], "$dir$t$Nombremp3"))
        {
        $resultado = "<p>Música añadida a la lista</p>";
        }
else
        {
        $resultado = "<p>El archivo " . $_FILES['musica']['name'] . " no se ha subido</p>";
        echo "<a href='index.php'> Volver";
        }
```

Seleccionar música

En primer lugar realizamos un escaneo de la carpeta "MusicaSubida"(***Scandir()***), a continuación con *"array_dif"* omitimos del resultado el "." y ".." así podremos asignar una variable con un *array* unicamente de los nombre de archivo del directorio.

```
$sdir  = array_diff(scandir($directorio), array('..', '.'));
```

Por ultimo los incluimos en una barra desplegable(*select*) dentro del formulario.

Dentro del bucle(foreach) este comprobará si alguna canción del directorio coincide con la que hay en la base de datos guardad, en ese caso la mostrará en primer lugar y en el resto de casos las mostrará como una opción más.

```
foreach ($sdir as $key => $nom) //obtenemos un archivo y luego
otro sucesivamente
    {
        $resaltar = "";
        if ($row['Primera'] == $nom)//Si la canción de la base
        de datos coincide con con el nombre de la canción del
        directorio
        {
            $resaltar = ' selected="selected"';//Establece esa
            canción como seleccionada
        }
        echo "<option value=\"$nom\"$resaltar>$nom</option>";//
        Agrega un valor para seleccionar al formulario

        }

echo   "</select></td></tr>";
```

Este formulario se repite en tres ocasiones, una por cada archivo de música que se va procesar, por último una vez se haya pulsado el botón de enviar se hará una llamada a la función ***subida():***

En primer lugar se comprueba si el nombre del archivo seleccionada en el formulario es distinto del guardado en la base de datos, en caso ser cierto copiará dicho archivo del directorio **"MusicaSubida"** a **"MusicaEnUso"** con el nombre correspondiente a la hora en la que tiene que sonar.

```
if ($primerahora != $musica_elegida['Primera'])//Si la música
seleccionada es diferente de la última guardada
    {

    if (copy("$dir$t$primerahora", "$rid/PRIMERA.mp3"))//Copiar la
    canción al directorio de la musica en uso con el nombre
    apropiado
        {
```

Al mismo tiempo que realiza la copia, comprueba si se ha copiado con éxito el archivo, en el caso de ser así, realizará un ***UPDATE*** en la base de datos para saber cual es la ultima canción que se ha copiado a "MusicaEnUso". En caso contrario mostrará un mensaje de error. Estas comprobaciones al igual que los formularios se repiten el mismo número de veces

```
if (copy("$dir$t$primerahora", "$rid/PRIMERA.mp3"))//Copiar la
canción al directorio de la musica en uso con el nombre
apropiado
    {

    $result = consulta("UPDATE musica set Primera='" . $
        primerahora . "'");
    $resultado .= "<p>Primera hora: SE HA CAMBIADO</p>";
    }
else
    {
    echo "Ha ocurrido un error";//Al no haberse copiado
    correctamente
    }
```

Reproducir ahora

Este formulario nos permitirá reproducir los primeros 90 segundos de los archivos de música subidos previamente a la carpeta **MusicaSubida,** para mostrar los archivos disponibles para su reproducción repetiremos el formulario en el que mostraos el nombre de los archivos con ***scandir()*** *y* omitimos los resultados "." y ".." que nos saca está función de PHP.

```
 echo "<form action='index.php' method='post'>";
 echo   "<legend>Reproducir ahora</legend>";
echo "Seleccionar música:";

               echo   "<select name='ahora1'>";
               echo       "<option value='NONE' selected='selected'>Selección</option>";
        foreach ($sdir as $numero => $nom) //obtenemos un archivo y luego otro $
                        {
                                echo "<option value=$nom>$nom</option>";

                                }
 echo   "</select>";
echo "<br><input type='submit' value='Reproducir' name='ahora'/><br>";
echo "</form>";
```

Una vez se pulse en el botón reproducir del formulario se mostrará un cuadro con un mensaje de "Reproduciendo..." y el nombre del archivo, ejecutando el script que reproducirá el archivo seleccionado además de activar el sistema de audio gracias a la función de PHP ***Exex()***que nos permite ejecutar un comando externo de nuestro sistema y en este caso el script que hemos creado para reproducir la música.

```
//REPRODUCIR AHORA
if (isset($_REQUEST['ahora'])){
echo "Reproduciendo...";
$ahoramus=$_REQUEST['ahora1'];
$musicaA="$directorio/$ahoramus";
$comando = 'sh /var/www/scriptmusica.sh '. $musicaA . ' 0';
exec($comando);
```

Link Salir

Una vez iniciada la sesión la primera funcionalidad que nos encontramos en la página es una URL subrayada junto a un texto indicando con que usuario hemos iniciado, esto ultimo es posible gracias a que se ha declarado una ***variable de sesión*** que guarda el valor introducido en el formulario de inicio de sesión.

```
echo "<div id='usuario'>";
echo "Cuenta de ".$_SESSION['usuario_valido'];
echo ' | <a href="?salir">Salir</a>';
echo "</div>";
```

Una vez se hace click sobre el texto *Salir,* con la etiqueta *<a href=>* declaramos una variable llamada *salir.*

```
if (isset($_GET['salir']))//Comprobar si existe variable
    {
    salir();//Función salir
    header("Location:index.php");
    }
```

Despues comprobamos si esta variable está declara (es decir han pulsado en el link) y en caso de cumplirse haremos una llamada a la función ***salir()*** *y* mediante la función de php ***header()***nos recargará la página y mostrará de nuevo el formulario de inicio de sesión.

```
function salir()
    {
    session_destroy();
    unset($_SESSION['usuario_valido']);//Borrar variables
    unset($_SESSION['pswd']);//Borrar variables
    }
```

Con la función ***salir*** en primer lugar destruimos las variables de sesión que haya en ese momento y por motivos de seguridad vaciamos las variables de sesión con el nombre y la contraseña del usuario que se han usado anteriormente.

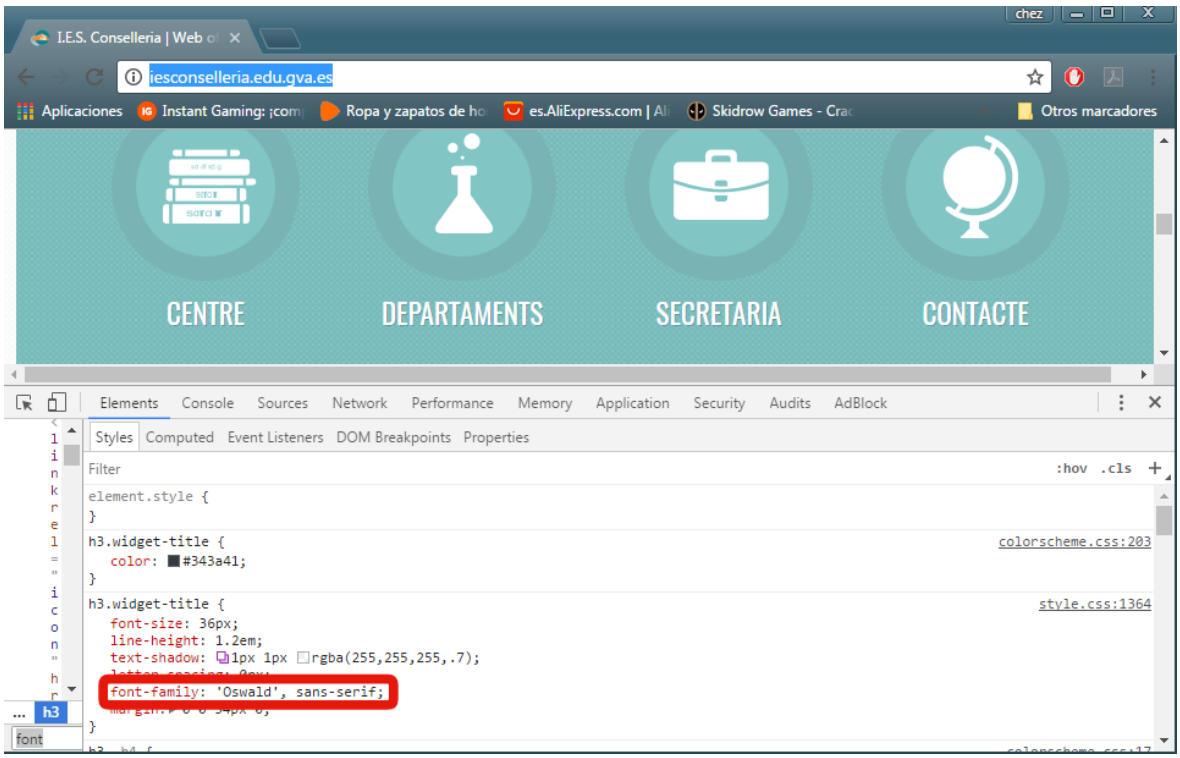
CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación.

Por lo tanto los estilos están declarados en el comienzo de la pagina principal (*Index.php*) en el conjunto de etiquetas pertenecientes a HTML concretamente en la etiqueta ***<header>*** donde van a estar recogidos todos los elementos a los que habrá que aplicarle los estilos.

```
<!DOCTYPE html>
<html>
<head>
<link rel='stylesheet' href='aspecto/estilo.css' type='text/css'>
<script>
```

El estilo de la página esta inspirado en la página web del centro IES Conselleria (*www.iesconselleria.edu.gva.es/*) tanto para los colores como para el tipo de fuente escogido (**Oswald font family**). Para saber que tipo de fuente se ha usado en dicha web ha bastado con hacer *Click derecho* sobre cualquier lugar de la web y seleccionar *Inspeccionar elemento* o pulsar la combinación de teclas *CTRL+MAYUS+I* y buscar dentro del desplegable alguna etiqueta ***Font*** que nos diga el nombre de la familia de fuente en uso.



Después bastaría con declarar las fuentes que se vayan a usar de esa familia en nuestro archivo css.

```css
@font-face {
  font-family: Osvaldo;
  src: url('Oswald/Oswald-Regular.ttf');
}

@font-face {
  font-family: OsvaldoStrong;
  src: url('Oswald/Oswald-SemiBold.ttf');
}
```

Para los títulos que encabezan cada formulario se ha dejado un espacio de 34px hacia abajo con la etiqueta *margin:*

```css
legend {
    font-family: Osvaldo, sans-serif;
    font-size: 2rem;
    color: #343a41;
    margin: 0 0 34px 0;
}
```

Todo lo que conforma la página web esta dentro de la etiqueta ***<body>*** del HTML así poder incluirlo todo dentro de un mismo recuadro de acciones el cual cambiará de tamaño dependiendo del tamaño de la ventana del navegador mediante la sentencia ***position: relative,*** ademas de establecer un máximo y mínimo de ancho (*max-width y min-widh)* al recuadro para que en caso de reducir demasiado la venta el texto de dentro no se sobreponga.

```css
body {
    width: 80%;
    max-width: 800px;
    min-width: 660px;
    margin: 1rem auto;
    padding: 1rem;
    box-shadow: 0 0 10px #333333;
    background-color: white;
    position: relative;
}
```

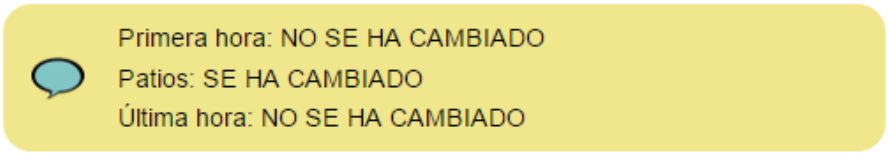
La ID usuario se ha asignado al texto que conforma el botón de salir y el nombre de usuario que ha iniciado sesión, este tiene la etiqueta *position :absolute* la cual haría que las dos etiquetas siguientes que indican la posición del texto estuviera en relación al recuadro anterior donde se encuentra.

```css
#usuario {
    position: absolute;
    right: 1rem;
    top: 2.4rem;
}
```

La ID *resultado* está asignada a la variable $resultado de la función subida(), esta variable muestra una cadena de texto dependiendo de si el archivo de música se ha tenido que cambiar o si no se ha cambiado:

```
$resultado .= "<p>Primera hora: SE HA CAMBIADO</p>";
```

Una vez se hayan enviado el formulario de selección de música se mostrará el mensaje dentro de un cuadro al comienzo de la aplicación web.



El texto se ha centrado para dejar hueco a la imagen con el *padding* y con *flex-direction: column* se ha colocado el texto en lineas separadas, el tamaño del recuadro ha sido limitado a 500px para que no ocupará la totalidad de la pagina y con *border-radius* se han redondeado los bordes, la imagen del bocadillo se ha agregado con la etiqueta *background-image: url(bocadillo.png)* y establecido para que no se repita más que una vez.

```css
#resultado {
    margin: 1rem 1rem 1rem 1rem;
    padding: 0.5rem 0.5rem 0.5rem 65px;
    width: 500px;
    border-radius: 1rem;
    background-color: khaki;
    background-image: url(bocadillo.png);
    background-position: 15px center;
    background-repeat: no-repeat;
    min-height: 3rem;
    display: flex;
    justify-content: center;
    flex-direction: column;
}
```

Conclusiones y mejoras

Se dispone de una aplicación web dentro de un sistema RaspBerry pi 3B con RaspBian instalado que realiza las mismas funciones que el script escrito en bash del ordenador del centro con el sistema de sonido conectado.

El entorno web ha sido enfocado a mostrar de manera grafica lo que se está haciendo de manera manual con los archivos de audio que se desean reproducir

Para que el usuario que se conecte a la web solo tenga que subir los archivos que desee utilizar y seleccionar la música que quiere que se reproduzca en los descansos u horas de comienzo y final correspondientes, suprimiendo así la necesidad de que el usuario tenga que preocuparse de la duración de los archivos ni del tipo de extensión que tengan dichos archivos, puesto que la misma aplicación realiza estas comprobación de seguridad.

La aplicación esta en su totalidad escrita en PHP con un pequeño detalle gestionado con *Java script* y el diseño y aspecto configurado con *CSS*.

En el proyecto se incluirán tanto los archivos PHP de la web como el script que ejecutará esta como el archivo *crontab* configurado para que reproduzca el archivo de audio correcto en el horario establecido en el centro.

El sistema operativo y el servidor web Apache totalmente configurados, excepto la configuración de red ya que en el centro la IP se da a los equipos por DHCP por lo que convendría reservar una IP estática para la RaspBerry y asi poder acceder a la web siempre con la misma dirección y adicionalmente agregar en el archivo host de los equipos donde se vaya a usar la aplicación una sentencia para poder acceder por un nombre de dominio y no por una dirección IP.