**Informe de la E2 del MF0493\_3**

Juan Carlos Francisco Mesa.

Índice

[​](#__RefHeading___Toc200_3928665858)

[Actividad a Realizar. 1](#__RefHeading___Toc200_3928665858)

[​ Script de Instalación. 3](#__RefHeading___Toc202_3928665858)

[​ Base de datos para el registro de email. 5](#__RefHeading___Toc5003_3915280640)

[​ Script de la base de datos: 5](#__RefHeading___Toc5005_3915280640)

[​ Código de la aplicación (index.php): 6](#__RefHeading___Toc5007_3915280640)

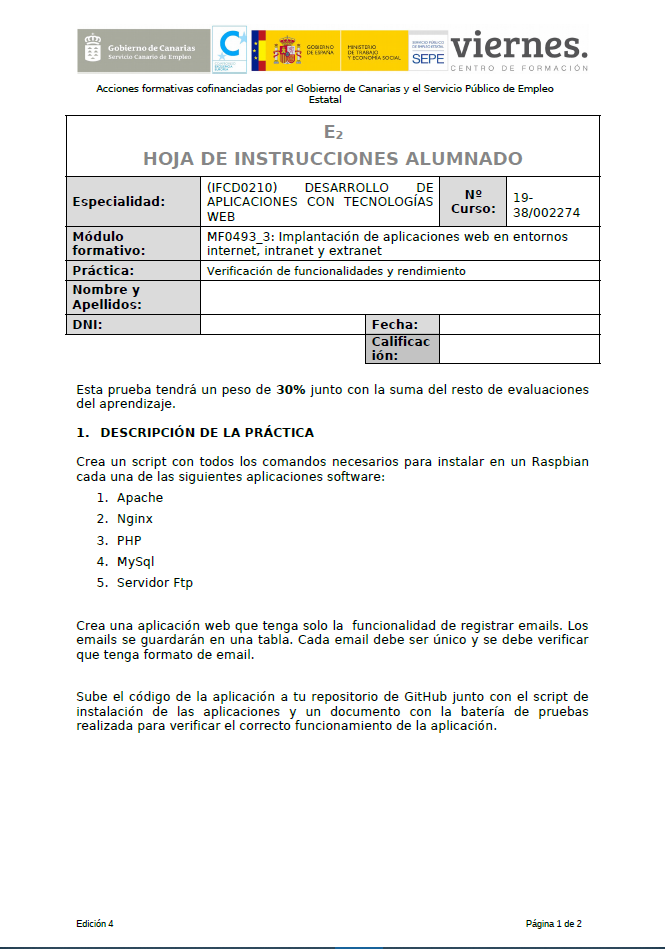
[​ Anexo A 10](#__RefHeading___Toc5034_3915280640)

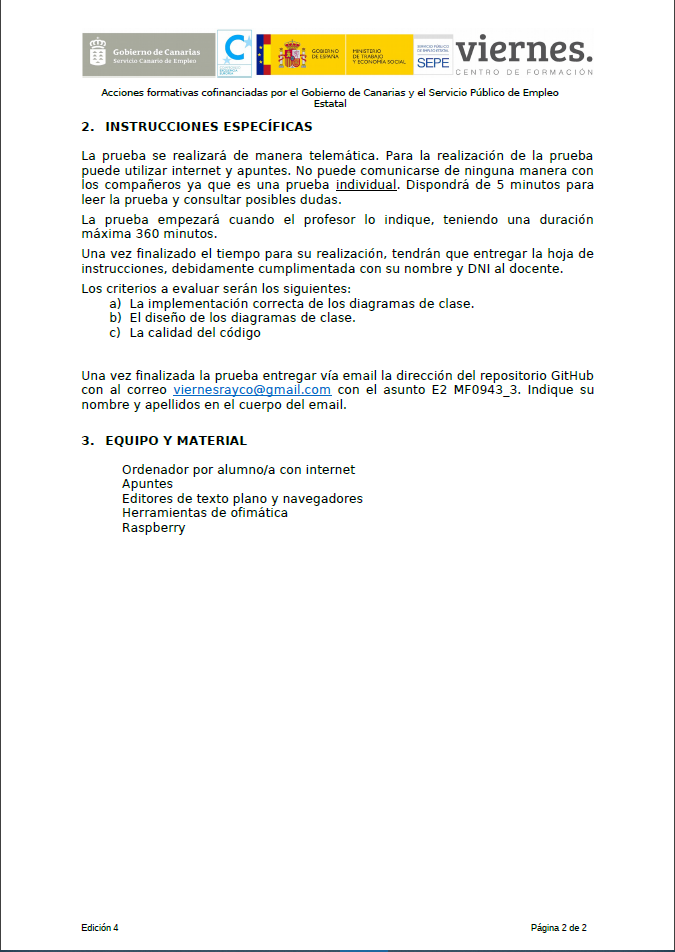
[​ Anexo B 11](#__RefHeading___Toc5036_3915280640)

[​ Pruebas realizadas al Script de instalación de componentes en la Raspberry. 11](#__RefHeading___Toc5060_3915280640)

[​ Pruebas realizadas a la aplicación 13](#__RefHeading___Toc5062_3915280640)

# **Actividad a Realizar**.





# **Script de Instalación**.

Se procedió a la creación de un Script para la instalación de los distintos componentes que se solicitan en la hoja de actividades insertada anteriormente, el script hace la instalación completamente automatizada y al finalizar, a modo de batería de pruebas, ejecuta varios comandos que nos muestran por pantalla la confirmación de la instalación y la versión instalada de cada uno de los componentes solicitados.

Para mayor información diríjase al archivo **README.MD** del repositorio indicado en el anexo “A” del presente documento y a los comentarios presentes en el mismo Script.

#!/bin/bash

*if* [ "$(whoami)" != "root" ]; *then*

    echo "Run script as ROOT please. (sudo !!)"

    exit

fi

## Instalamos apache

apt-get install apache2 -y

## Instalamos PHP

apt-get install php php-mysql libapache2-mod-php -y

## Instalar MariaDB

apt-get install mariadb-server mariadb-client -y

## Securizamos la instalación de MariDB

mysql\_secure\_installation

## Instalamos phpmyadmin

apt-get install phpmyadmin -y

## Cambiamos el propietario del directorio /html que es en donde irán nuestras páginas, se agrega el usuario pi al grupo

## y se cambian los permisos de lectura y escritura

chown -R www-data:www-data /var/www/html

usermod -g www-data pi

chmod -R 777 /var/www

echo

echo

echo

## El siguiente comando reinicia apache2 para que corra phpmyadmin

service apache2 reload

## Vamos ahora a instalar Nginx

apt-get install nginx -y

## Atención: El servicio se intentará iniciar en el puerto 80 y no podrá hacerlo por que

## ahí estará corriendo el servicio de Apache.

## Finalmente vamos a instalar Vsftp

apt-get install vsftpd -y

## Ya podremos usar el servidor ftp con las credenciales de los usuarios del sistema

##################

## Ahora vamos a realizar las comprobaciones de la instalación realizada

################

##Limpia la pantalla

clear

## Vemos la versión de apache

echo "La versión de apache es: "

apache2 -v

## Versión de Nginx

echo

echo "La versión de Nginx es: "

nginx -v

## Versión de PHP

echo

echo "Versión de PHP es: "

php -v

## Version de Vsftp

echo

echo "Versión de Vsftp es: "

vsftpd -v

## Para la comprobación de mariadb será necesario loguearse en el cli del SGBD con las

## credenciales creadas

echo "Instalación finalizada!!!!"

# Base de datos para el registro de email.

Se procedió a la creación de una base de datos llamada **email**  para el registro de email, con la condición de que cada email debería ser único (esta comprobación la llevará a cabo la base de datos) y se debe comprobar la introducción de un email con el formato correcto (esta comprobación la llevará a cabo la aplicación).

## Script de la base de datos:

-- phpMyAdmin SQL Dump

-- version 5.0.2

-- https://www.phpmyadmin.net/

--

-- Servidor: 127.0.0.1

-- Tiempo de generación: 14-01-2021 a las 13:43:10

-- Versión del servidor: 10.4.14-MariaDB

-- Versión de PHP: 7.4.10

*SET* SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

*START TRANSACTION*;

*SET* time\_zone = "+00:00";

*/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/*;

*/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/*;

*/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/*;

*/\*!40101 SET NAMES utf8mb4 \*/*;

--

-- Base de datos: `email`

--

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `email`

--

*DROP* TABLE IF EXISTS `email`;

*CREATE* TABLE `email` (

  `idEmail` *int*(11) NOT NULL,

  `email` *varchar*(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Índices para tablas volcadas

--

--

-- Indices de la tabla `email`

--

*ALTER* *TABLE* `email`

  ADD *PRIMARY KEY* (`idEmail`),

*ADD UNIQUE* KEY `email` (`email`);

--

-- AUTO\_INCREMENT de las tablas volcadas

--

--

-- AUTO\_INCREMENT de la tabla `email`

--

*ALTER* *TABLE* `email`

  MODIFY `idEmail` *int*(11) NOT NULL *AUTO\_INCREMENT*;

*COMMIT*;

*/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/*;

*/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/*;

*/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/*;

## Código de la aplicación (index.php):

<!DOCTYPE *HTML*>

<html>

<head>

    <style>

        .error {

            color: #FF0000;

        }

    </style>

</head>

<body>

    <?php

*//Datos de conexión a la base de datos.*

    $db = [

        'host' => 'localhost',

        'username' => 'root',

        'password' => '',

        'db' => 'email' *//Cambiar al nombre de tu base de datos*

    ];

*//Abrir conexion a la base de datos*

*function* connect($db)

    {

*try* {

            $conn = *new* PDO("mysql:host={$db['host']};dbname={$db['db']};charset=utf8", $db['username'], $db['password']);

*// set the PDO error mode to exception*

            $conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

*return* $conn;

        } *catch* (PDOException $exception) {

*exit*($exception->getMessage());

        }

    }

*// define e inicializa variables*

    $emailErr = "";

    $email = "";

    $msg = "";

*// Hace una depuración de los datos eliminando espacios en blanco, barras,*

*//caracteres especiales.*

*function* test\_input($data)

    {

        $data = trim($data);

        $data = stripslashes($data);

        $data = htmlspecialchars($data);

*return* $data;

    }

*// Comprobamos que el email  esté informado y correctamente formado*

*if* ($\_SERVER["REQUEST\_METHOD"] == "POST") {

*if* (empty($\_POST["email"])) {

            $emailErr = "Email is required";

        } *else* {

            $email = test\_input($\_POST["email"]);

*if* (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

                $emailErr = "Invalid email format";

            } *else* {

*// Si todas las condiciones son válidas abrimos*

*// la conexión y hacemos la consulta.*

                $dbConn = connect($db);

                $sql = "*INSERT INTO* email

                        (idEmail, email)

*VALUES*

                        (null,:email)";

                $statement = $dbConn->prepare($sql);

                $statement->bindValue(':email', $email);

*// Capturamos cualquier error*

*try* {

                    $statement->execute();

                } *catch* (\Throwable $th) {

                    $error = $th;

                }

*// Recoge el valor del último registro*

                $postId = $dbConn->lastInsertId();

                $dbConn = "";

*// Definimos el mensaje de resultado*

*if* ($postId) {

                    $msg = "Email registrado exitosamente con id: {$postId}";

                } *else* {

                    $msg = "Lo siento, pero su email no ha podido ser registrado debido a: {$error}";

                }

            }

        }

    }

    ?>

*<!-- Este es el Html de la aplicación -->*

    <h2>Registro y validación de emails</h2>

    <p><span *class*="error">\* required field</span></p>

    <form *method*="post" *action*="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>">

        E-mail: <input *type*="text" *name*="email" *value*="<?php echo $email; ?>">

        <span *class*="error">\* <?php echo $emailErr; ?></span>

        <br><br>

        <input *type*="submit" *name*="submit" *value*="Submit">

    </form>

    <?php

*// Emitimos mensaje de confirmación*

    echo "<h2>Your Input:</h2>";

    echo $msg;

    echo "<br>";

    ?>

</body>

</html>

# Anexo A

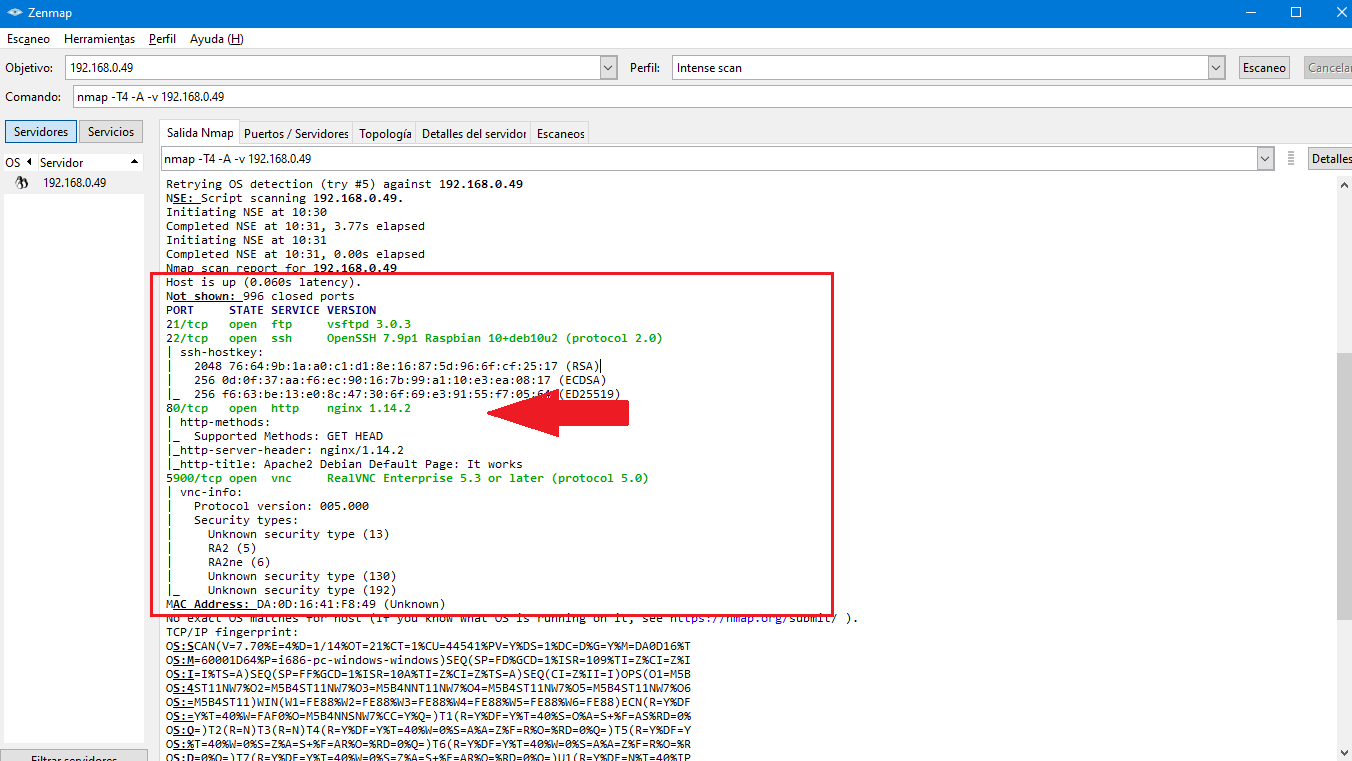
Todo el Código Fuente se puede descargar del repositorio en:

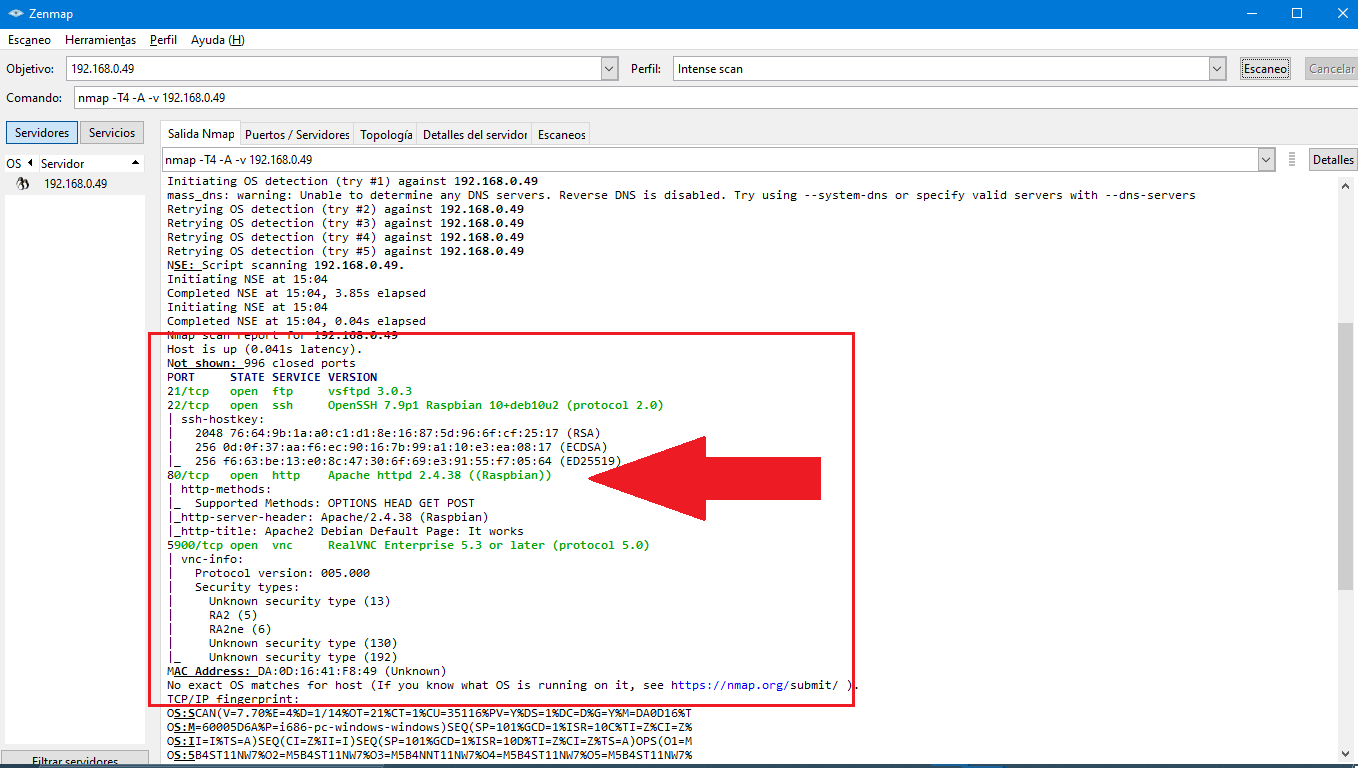
https://github.com/jcfr4n/E2\_MF0493\_3.git

# Anexo B

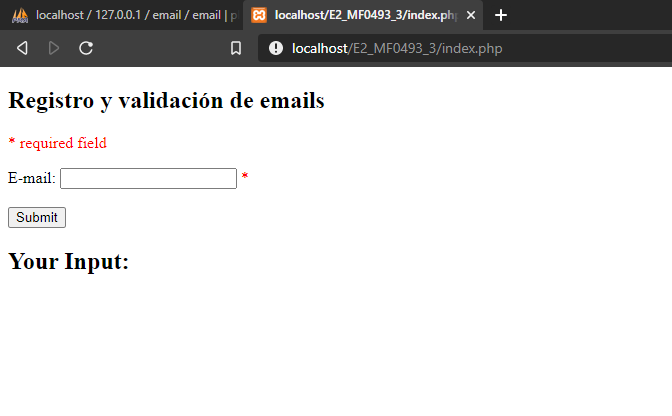
## Pruebas realizadas al Script de instalación de componentes en la Raspberry.

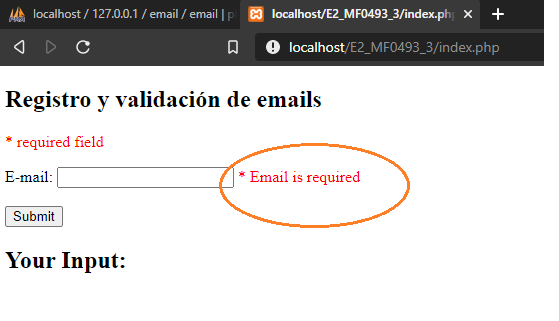
En la Raspberry como ya pudimos observar el mismo Script contiene la comprobación de las versiones de los componentes instalados, además se analizó la ip con Zenmap (Interfaz gráfica de windows para la herramienta Nmap) en donde se puede observar los servicios que están corriendo en ella.

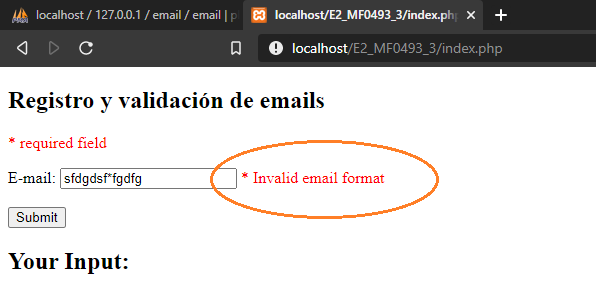
Figura 1: Servicio Nginx Activado

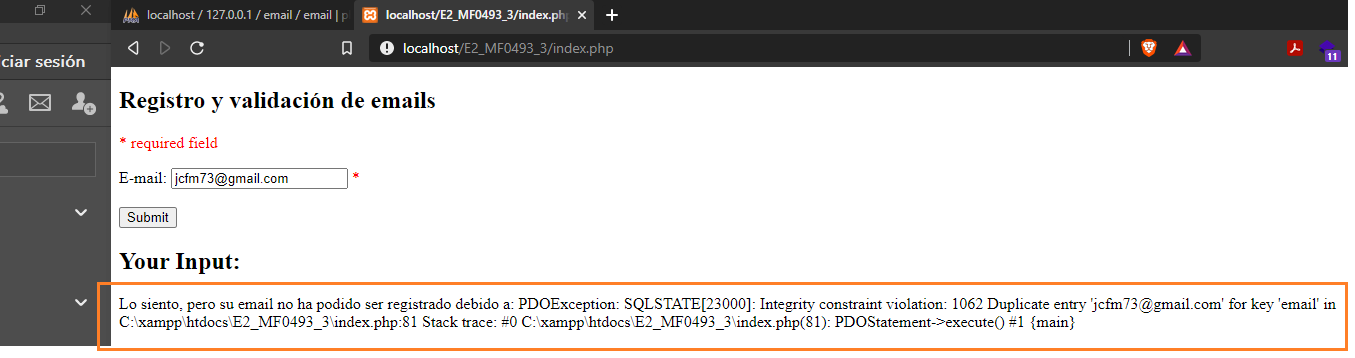
Figura 2: Corriendo el servicio Apache

## Pruebas realizadas a la aplicación

Figura 3: Pantalla inicial de aplicación

Figura 4: Submit sin rellenar el campo

Figura 5: Formato inválido

Figura 6: Email duplicado

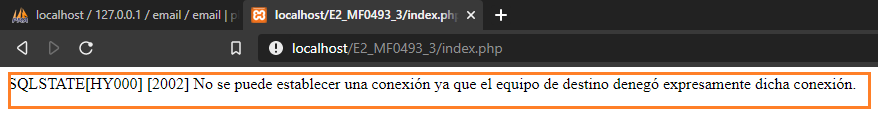
Figura 7: Base de datos inaccesible

Figura 8: Email registrado correctamente