CFGS: DESARROLLO DE APLICACIONES WEB

CURSO:2021-2022

Dibujo en blanco y negro

Descripción generada automáticamente con confianza baja

Manual de DESARROLLADOR

el santuario

Avilés, junio de 2022

Menéndez Muñiz, ELVIS DANIEL

Tutor/a individual: ARABIA ÁLVAREZ, JOSÉ MANUEL

Tutor/a colectivo: GARCÍA PARRA, MATEO

Índice

[Requerimientos técnicos del sistema 4](#_Toc105783695)

[Descripción de la base de datos 4](#_Toc105783696)

[Estructura 4](#_Toc105783697)

[Tabla usuarios 5](#_Toc105783698)

[Tabla animales 6](#_Toc105783699)

[Tabla adopciones 6](#_Toc105783700)

[Tabla actividades 7](#_Toc105783701)

[Tabla tipoactividades 7](#_Toc105783702)

[Cuentas de usuario de la base 7](#_Toc105783703)

[Cuenta de gerente 7](#_Toc105783704)

[Cuenta de voluntario 8](#_Toc105783705)

[Cuenta de adoptante 8](#_Toc105783706)

[Cuenta de comprobar 10](#_Toc105783707)

[Definición y descripción de variables usadas en el sistema 10](#_Toc105783708)

[Funciones MySQL 10](#_Toc105783709)

[mysqli\_connect 10](#_Toc105783710)

[mysqli\_query 12](#_Toc105783711)

[mysqli\_num\_rows 12](#_Toc105783712)

[mysqli\_fetch\_array 13](#_Toc105783713)

[Funciones de PHP 14](#_Toc105783714)

[hash\_hmac 14](#_Toc105783715)

[isset 14](#_Toc105783716)

[header 15](#_Toc105783717)

[implode 15](#_Toc105783718)

[explode 16](#_Toc105783719)

[str\_shuffle 16](#_Toc105783720)

[Substr 16](#_Toc105783721)

[Funciones personalizadas 17](#_Toc105783722)

[get\_header 17](#_Toc105783723)

[get\_footer 17](#_Toc105783724)

[Funciones para la validación de datos 18](#_Toc105783725)

[validar\_tel 18](#_Toc105783726)

[validar\_dni 18](#_Toc105783727)

[validar\_nombre y validar\_ape 19](#_Toc105783728)

[calculaedad 19](#_Toc105783729)

[validar\_hora 19](#_Toc105783730)

[validar\_fecha 20](#_Toc105783731)

[Despliegue de la aplicación 21](#_Toc105783732)

[Instalación 21](#_Toc105783733)

[Configuración de la base de datos 26](#_Toc105783734)

[Estructura de la aplicación 28](#_Toc105783735)

[Header 28](#_Toc105783736)

[Footer 28](#_Toc105783737)

[Índex (Home) 29](#_Toc105783738)

[Actividades 29](#_Toc105783739)

[Instalaciones 30](#_Toc105783740)

[Login 30](#_Toc105783741)

[Perfil 31](#_Toc105783742)

[Perfil Gerente 32](#_Toc105783743)

[Perfil Voluntario 40](#_Toc105783744)

[Perfil adoptante 42](#_Toc105783745)

[Cerrar sesión 44](#_Toc105783746)

[Índice de figuras 45](#_Toc105783747)

[Bibliografía 47](#_Toc105783748)

# Requerimientos técnicos del sistema

Para poder ejecutar la aplicación necesitaremos XAMPP y para ello necesitamos los siguientes requisitos:

* Conexión a internet para obtener el programa desde la web Proyecto XAMPP.
* 256 MB de espacio libre en Memoria RAM.
* 100 MB de espacio libre en nuestro disco duro.
* PHP 8.12.

# Descripción de la base de datos

## Estructura

La aplicación contará con una base de datos llamada elsantuario la cual tendrá 5 tablas:

* usuarios
* actividades
* animales
* adopciones
* tipoactividades

Diagrama

Descripción generada automáticamente

Ilustración 1:Diagrama E/R

En el modelo Entidad/Relación se puede ver el tipo de datos utilizado en cada campo.

Tabla usuarios

* dniUsuario: DNI que identifica a cada usuario para acceder a la aplicación web. (Clave primaria)
* usuPassword: contraseña con la que se accederá a la aplicación web.
* usuTipo: rol que tendrá el usuario en la aplicación web.
* usuNombre: nombre del usuario.
* usuApe: apellido/s del usuario.
* usuTel: número telefónico del usuario. (Puede quedar vacío el campo)
* usuDir: dirección física del usuario. (Puede quedar vacío el campo)
* fechaNac: fecha de nacimiento del usuario. (Puede quedar vacío el campo)

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración 2:Tabla usuarios

### Tabla animales

* idAnimal: Identificador del animal. (Clave primaria, Autoincrement)
* aniNombre: nombre del animal.
* aniFechaIngreso: fecha de ingreso del animal en el refugio.
* aniFechaAdopcion: fecha de salida de animal al ser adoptado. (Campo vacío hasta que se realice la adopción)
* aniEspecie: especie del animal.
* aniSexo: sexo del animal.
* aniAdopción: valor 0 hasta que el animal es adoptado que se modificará al valor 1.

Una captura de pantalla de una red social

Descripción generada automáticamente

Ilustración 3:Tabla animales

### Tabla adopciones

* idAdopcion: identificación de la adopción realizada. (Clave primaria, Autoincrement)
* dniAdoptante: identificación del usuario que ha adoptado.(adopciones\_fk1 (Clave Foránea))
* idAnimal: identificación del animal adoptado.(adopciones\_fk2 (Clave Foránea))
* fechaAdopcion: fecha en la que se realizó la adopción.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Ilustración 4:Tabla adopciones

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 5:Índices Tabla adopciones

### Tabla actividades

* id: identificación de la actividad que se va a realizar. (Clave primaria, Autoincrement)
* tipoActividad: identificación del tipo de actividad.(actividades\_fk1 (Clave Foránea))
* dniVoluntario: identificación del usuario con rol voluntario.(actividades\_fk2 (Clave Foránea))
* fechaActividad: fecha en la que se va a realizar la actividad.
* horaActividad: hora en la que se va a realizar la actividad.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Ilustración 6:Tabla actividades

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración 7:Índices Tabla actividades

### Tabla tipoactividades

* idActividad: identificación del tipo de actividad. (Clave primaria, Autoincrement)
* nombreActividad: nombre descriptivo de la actividad.

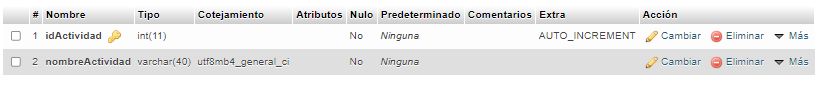


Ilustración 8:Tabla tipoActividades

## Cuentas de usuario de la base

### Cuenta de gerente

El usuario con rol de gerente solo tendrá permisos a nivel de tablas para que no pueda manipular otras tablas o bases de datos.

* Privilegios de SELECT en todas las tablas.
* Privilegios de INSERT en las tablas de actividades, adopciones, usuarios y animales para añadir nuevos datos a través de los formularios.
* Privilegios de UPDATE en las tablas actividades, adopciones y animales para poder modificar los datos.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Ilustración 9:Privilegios cuenta gerente

### Cuenta de voluntario

El usuario con rol de voluntario tendrá permisos a nivel de tablas y campo para restringir sus funciones en la base de datos.

* Privilegios de SELECT en las tablas actividades, animales y tipoActividad.
* Privilegios de UPDATE en las tabla actividades únicamente en los campos fechaActividad y horaActividad para poder modificar los datos.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Ilustración 10:Privilegios cuenta voluntario

### Cuenta de adoptante

El usuario con rol de voluntario solo tendrá permisos a nivel de campos de determinadas tablas para restringir sus funciones en la base de datos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 11:Privilegios cuenta adoptante

#### Tabla adopciones

Solo podrá realizar SELECTS de los campos dniAdoptante ,idAnimal y fechaAdopcion para poder sacar los datos de sus adopciones.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ilustración 12:Permisos adoptante en la tabla adopciones

#### Tabla animales

Solo podrá realizar SELECTS de los campos idAnimal ,aniNombre, aniEspecie y aniSexo para poder sacar los datos de sus adopciones.Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 13:Permisos adoptante en la tabla animales

#### Tabla usuarios

Podrá realizar SELECT en el campo dniUsuario y UPDATE en el campo usuPassword para poder cambiar su contraseña.Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 14:Permisos adoptante en la tabla usuarios

### Cuenta de comprobar

Este usuario solo podrá hacer SELECT en la tabla de usuarios para poder comprobar si un usuario existe en la base de datos a la hora de hacer el login. Si existe guardará toda la información del usuario en una sesión salvo la contraseña para poder ser usada en el futuro.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 15:Privilegios cuenta comprobar

# Definición y descripción de variables usadas en el sistema

## Funciones MySQL

### mysqli\_connect

La función mysqli\_connect crea una conexión al servidor MySQL (servidor de la base de datos) al recibir ciertos parámetros.

* **Host**: especifica el host o la IP. (En nuestro caso será siempre localhost)
* **Username**: especifica el nombre de usuario de MySQL con el que nos queremos conectar. (En nuestro caso irá variando, dependiendo del módulo en el que nos encontremos)
* **Password**: especifica la contraseña del usuario de MySQL
* **Nombre de la base de datos:** especifica el nombre de la base de datos a la que nos queremos conectar. (Nuestro caso a la base de datos elsantuario)

**Conexiones a la base de datos utilizadas:**

Conexión a la base de datos con el usuario comprobar el cual usaremos en el módulo de login para comprobar si el usuario que intenta iniciar sesión se encuentra en la base de datos.

Texto

Descripción generada automáticamente

Ilustración 16:Conexión a la base de datos como comprobar

Conexión a la base de datos con el usuario gerente el cual usaremos en el módulo de gerente para realizar las operaciones necesarias en la base de datos.Texto

Descripción generada automáticamente

Ilustración 17:Conexión a la base de datos como gerente

Conexión a la base de datos con el usuario voluntario el cual usaremos en el módulo de gerente para realizar las operaciones necesarias en la base de datos.Texto

Descripción generada automáticamente

Ilustración 18:Conexión a la base de datos como voluntario

Conexión a la base de datos con el usuario voluntario el cual usaremos en el módulo de gerente para realizar las operaciones necesarias en la base de datos. Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Ilustración 19:Conexión a la base de datos como adoptante

### mysqli\_query

La función mysqli\_query ejecuta consultas sobre la base de datos. Para ello necesita dos parámetros; el primero será una variable con la conexión a MySQL y el segundo parámetro será la consulta que deseemos realizar.

**Ejemplos:**

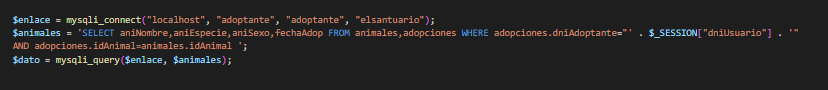
En este ejemplo de consulta SELECT guardamos la conexión como usuario adoptante en la variable $enlace y en la variable $animales una consulta SELECT sobre las tablas animales y adopciones

Ilustración 20:Ejemplo mysqli\_query de consulta SELECT

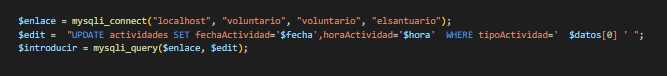
Este ejemplo es similar al anterior, pero en este caso realizaremos una consulta de tipo UPDATE para modificar la fecha y hora de la tabla actividades.

Ilustración 21:Ejemplo mysqli\_query de consulta UPDATE

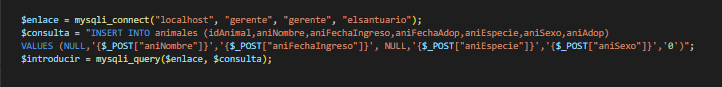
Ejemplo de consulta para realizar un INSERT en la tabla animales, para ello hay que especificar todos los campos de la tabla y añadir los valores que se deseen insertar.

Ilustración 22:Ejemplo mysqli\_query de consulta INSERT

### mysqli\_num\_rows

La función mysqli\_num\_rows devuelve el número de filas de la consulta que se le pase como parámetro. Esta función es útil para comprobar si ciertos datos existen en una tabla o no.

En este ejemplo realizamos un SELECT para comprobar que no haya otro usuario con el DNI que se ha introducido en el formulario de registro. Si mysqli\_num\_rows devuelve el valor 0 se insertarán los datos del nuevo usuario, si devuelve el valor 1 dará un mensaje de error.

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración 23:Ejemplo mysqli\_num\_rows

### mysqli\_fetch\_array

La función mysqli\_fetch\_array retorna un array que corresponde a la fila obtenida o nulo si es que no hay más filas. Esta función es útil para visualizar datos de las tablas de la base de datos.

En este ejemplo por cada fila que exista en la base de datos se guardará toda la información en la variable $fila y se irá visualizando en la tabla pintada en HTML

Texto

Descripción generada automáticamente

Ilustración 24:Ejemplo utilización mysqli\_fetch\_array

Como se ve en el ejemplo se necesitan usar todas las funciones explicadas anteriormente para visualizar los datos en una simple tabla. La función mysqli\_query para crear la consulta, mysqli\_num\_rows para comprobar que existan datos en la tabla y mysqli\_fetch\_array para obtener los datos fila a fila.

## Funciones de PHP

### hash\_hmac

La función hash\_hmac será utilizada en la aplicación para cifrar las contraseñas introducidas por los usuarios a la hora de iniciar sesión y registrarse ya que genera un valor cifrado mediante una clave específica que le pasaremos como parámetro.

Parámetros necesarios:

* El primer parámetro será el tipo de cifrado que usaremos. En nuestro caso sha512
* Segundo parámetro será el mensaje que queremos cifrar.
* Tercer parámetro será una clave secreta compartida que se usará para crear el cifrado.

Texto

Descripción generada automáticamente con confianza media

Ilustración 25:Función hash\_hmac

### isset

La función isset() determina si una variable está definida y no es nula. Principalmente la utilizaremos para saber que botón pulsa el usuario y así realizar una acción u otra.

Texto

Descripción generada automáticamente

Ilustración 26:Función isset

Al pulsar el botón con determinado name se ejecutará una parte del código o no. Por ejemplo, si el usuario pulsa el botón con el nombre verAnimalCargo se guardará una cadena de texto en una variable se sesión y se redirigirá a voluntario.php.

### header

La función header es utilizada para enviar encabezados HTTP y así redireccionar hacia la dirección pasaba por parámetro en formato cadena.

Texto

Descripción generada automáticamente

Ilustración 27:Función header

### implode

La función implode() une los elementos de un array en una cadena de texto. Tiene dos parámetros

* Separador: es opcional y si se deja vacío los elementos de la cadena estarán separados por espacios en blanco
* Array: array el cual queremos transformar en una cadena.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Ilustración 28:Función implode

La utilizaremos para guardar los datos en el value de un botón para luego enviarlos en una session a otra parte de la página y modificarlos.

### explode

La función explode() al contrario que la función implode() transforma una cadena de texto en un array. Tiene los mismos parametros que include excepto que en vez de pasarle un array le pasaremos una cadena de texto.

Texto

Descripción generada automáticamente con confianza media

Ilustración 29:Función explode

### str\_shuffle

La función str\_shuffle baraja una cadena recibida como parámetro. Esta función será utilizada para crear una contraseña aleatoria para el usuario adoptante en el momento que el gerente le registra en la base de datos.

### Substr

La función substr devuelve una parte de la cadena de texto recibida como parámetro. Esta función tiene 3 parametros:

* string: es la cadena entera recibida.
* start: el valor en el que se desea empezar a cortar la cadena.
* length: el valor de la longitud de la cadena.



Ilustración 30:Funciones str\_shuffle y substr

Para crear la contraseña aleatoria guardaremos en una variable todos los números del 0 al 9 y todas las letras en minúsculas y mayúsculas. Con la función str\_shuffle se barajará toda la cadena y la función substr recortará la cadena a los 20 primeros valores.

## Funciones personalizadas

Para no tener que estar repitiendo tanto código HTML he creado 2 funciones para que llamen a los archivos en los que están guardados el header y footer de la aplicación web.

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

Ilustración 31:Archivo functions.php

Para hacer la llamada a estas dos funciones tendremos que añadir el siguiente código en cada una de las paginas en las cuales queremos que se visualicen.

### get\_header

La función get\_header() la llamaremos en la primera línea de código de cada archivo. Para ello necesitamos llamar al archivo en el que tenemos la función con un require.

Texto

Descripción generada automáticamente

Ilustración 32:Función get\_header

### get\_footer

Con la función get\_footer es el mismo procedimiento que con el get\_header pero en vez de llamarlo al inicio del archivo, haremos la llamada en la última línea del código.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Ilustración 33:Función get\_footer

## Funciones para la validación de datos

### validar\_tel

La función validar\_tel comprobará que el valor introducido en el campo teléfono sea una cadena de 9 números(return true), sino es así devolverá un valor false junto a un mensaje de error.

Captura de pantalla de un celular

Descripción generada automáticamente

Ilustración 34:Función validar\_tel

### validar\_dni

La función validar\_dni comprueba que el DNI que se introduce tenga una longitud de 9, los cuales son 8 números y 1 letra mayúscula y los 8 numeros sean divisibles entre 23 como ocurre en los DNI reales. Si una de las anteriores comprobaciones no se cumple se mostrará un mensaje de DNI inválido.Pantalla de computadora

Descripción generada automáticamente con confianza baja

Ilustración 35:Función validar\_dni

### validar\_nombre y validar\_ape

Estas dos funciones comprueban lo mismo, que no se introduzcan valores que no sean letras y el primer valor sea mayúscula. Sino se cumple se enviará un mensaje de error.

Texto

Descripción generada automáticamente

Ilustración 36:Funciones validar\_nombre y validar\_ape

### calculaedad

La función calculaedad recibe como parámetro la fecha de nacimiento introducida en el formulario y con el explode transforma el array en un string y se guarda en un listado.

Con los datos de ese listado y la fecha actual se consigue la fecha de diferencia en días, meses y años a la fecha actual.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Ilustración 37:Función calculaedad

### validar\_hora

Esta función comprueba que la hora introducida no sea una hora de una fecha ya pasada. Para ello con la función date conseguimos la hora y la fecha actual. Primero se comprueba si el día es el mismo recibido por parámetro,(sino es la misma retorna true) si es el mismo se comprueba que la hora actual sea menor que la hora recibida. Si la hora actual es una hora ya pasada retornará false y un mensaje de error, y si la hora recibida aún no ha llegado retornará true.

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración 38:Función validar\_hora

### validar\_fecha

He creado 3 versiones diferentes de esta función para cada necesidad.

La primera versión comprueba que la fecha actual no sea menor a la fecha recibida por parámetro. Esta se utilizará para comprobar que no se puedan introducir actividades en un día ya pasado.

Una captura de pantalla de un celular

Descripción generada automáticamente

Ilustración 39:Función validar\_fecha actividades

La segunda función validar\_fecha comprueba que un usuario registrado no haya nacido el día actual y entre los años 1900 y 2014. Sino retornará false y un mensaje de fecha incorrecta.

Captura de pantalla de un celular

Descripción generada automáticamente

Ilustración 40:Función validar\_fecha voluntario

Por último, está función se utilizará al crear una nueva adopción. Para ello se comprobará que el usuario sea mayor de 18 años, para ello llamaremos a la función calculaedad pasándole como parámetro la fecha de nacimiento introducida en el formulario.

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

Ilustración 41:Función validar\_fecha adoptante

# Despliegue de la aplicación

## Instalación

Lo primero que necesitamos instalar el servidor de aplicaciones XAMPP.

Es una distribución de Apache que **incluye varios softwares libres**. El nombre es un acrónimo compuesto por las iniciales de los programas que lo constituyen. La inicial **X** se usa para representar a los sistemas operativos Linux, Windows y Mac OS.

* **Apache**: el servidor web de código abierto es la aplicación más usada globalmente para la entrega de contenidos web.
* **MariaDB:**Es uno de los sistemas relacionales de gestión de bases de datos más populares del mundo. En combinación con el servidor web Apache y el lenguaje PHP, MySQL sirve para el almacenamiento de datos para servicios web.
* **PHP:**es un lenguaje de programación de código de lado del servidor que permite crear páginas web o aplicaciones dinámicas.
* **Perl**: este lenguaje de programación se usa en la administración del sistema, en el desarrollo web y en la programación de red.

Para descargar XAMPP lo primero que debemos hacer es dirigirnos a la página de [apachefriends](https://www.apachefriends.org/es/index.html) en la cual se encuentra el instalador. Y descargaremos la versión para Windows más reciente.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 42:Página descarga XAMPP

Una vez descargado, abriremos el ejecutable y comenzaremos con la configuración.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 43:Instalación XAMPP 1

Le daremos al botón de Next y nos aparecerá una pantalla con todos los componentes de XAMPP. Dejaremos todo como viene por defecto y continuaremos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 44:Instalación XAMPP 2

Nos pedirá la ruta en la cual crear la carpeta de XAMPP donde instalar todos sus componentes. También la dejaremos por defecto en C:\xampp.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 45:Instalación XAMPP 3

Nos pedirá seleccionar el idioma y dejaremos el que selecciona por defecto; inglés.

Nos aparecerán dos pestañas más antes del inicio de la instalación. Una de leer sobre Bitnami

Y otra advirtiendo que la instalación va a comenzar. Las omitiremos ambas y comenzaremos la instalación. Esto tardará unos minutos.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 46:Instalación XAMPP 4

Una vez finalizada la instalación abriremos el panel de control de XAMPP

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración 47:Instalación XAMPP 5

Para desplegar la aplicación en local necesitaremos modificar 2 archivos: el archivo hosts y el archivo httpd-vhost.

El archivo hosts lo encontraremos en C:\Windows\System32\drivers\etc .Para poder modificarlo correctamente necesitaremos abrir el blog de notas como Administrador, sino no se guardarán los cambios correctamente .

Añadiremos la siguiente línea al final del archivo: 127.0.0.1 elsantuario.es

Texto

Descripción generada automáticamente

Ilustración 48:Configuración archivo hosts

El archivo httpd-vhost lo encontraremos en la ruta: C:\xampp\apache\conf\extra

Y escribiremos las siguientes líneas de configuración al final del archivo

Texto

Descripción generada automáticamente

Ilustración 49:Configuración archivo httpd-vhost

Indicaremos que usaremos el puerto 80 (por defecto) para atender las solicitudes

**ServerName** indicamos el nombre del host. (Configurado en el archivo hosts)

**DocumentRoot** para indicar el directorio en el que se encuentra la aplicación que deseamos mostrar

**Require all granted** permite el acceso incondicionalmente al directorio especificado.

Para que los cambios sean configurados correctamente deberemos reiniciar el servidor de Apache.

## Configuración de la base de datos

Para instalar la base de datos necesitaremos arrancar el servicio de MySQL y en un navegador ir a la siguiente dirección: <http://localhost/phpmyadmin/>

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 50:Índex phpMyAdmin

Lo primero que hay que hacer es en el margen izquierdo crear una nueva base de datos. La llamaremos elsantuario y le daremos a crear.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 51:Crear base de datos

Una vez creada la base de datos daremos a importar y seleccionaremos el archivo .sql facilitado con todas las tablas creadas. Una vez importada correctamente nos aparecerán varios mensajes de base de datos importada correctamente y en el menú de la izquierda nos aparecerá la base de datos con las tablas. Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 52:Importación base de datos

El siguiente paso es crear los usuarios de la base. Para ello también se ha facilitado un archivo con todos los usuarios y permisos (usuariosBase.txt), solo hay que copiarlos y pegarlos en la consola de phpMyAdmin y ejecutarlos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 53:Creación de usuarios en la base de datos

Al crearlos exitosamente como al importar la base nos aparecerá un mensaje de usuario creado exitosamente. Para comprobar que sean creado podemos ir a la página de Cuentas de usuario. Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración 54:Cuentas de usuarios

## Estructura de la aplicación

### Header

Aparecerá en la parte superior de cada página que visitemos. Variará dependiendo de si el usuario haya iniciado sesión o no.



Ilustración 55:Header sin iniciar sesión



Ilustración 56:Header con sesión iniciada

Para hacer que dependiendo si un usuario este logeado o no nos salga diferente header se ha creado una variable de sesión que se mantendrá en valor false hasta que el usuario inicie sesión.

Texto

Descripción generada automáticamente

Ilustración 57:Código header

### Footer

Aparecerá en la parte inferior de todas las páginas que visitemos. Está compuesto por dos columnas, una con las redes sociales y otra con el contacto. Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración 58:Footer

### Índex (Home)

Página principal de la aplicación web.

Imagen que contiene cerca, persona, hombre, frente

Descripción generada automáticamente

Ilustración 59:Índex

### Actividades

Página en la que se pueden ver varias de las actividades que se realizan en el refugio. Cada una de estas actividades tienen un modal que al pulsar el botón te abre una pestaña con más información de la actividad.

Texto

Descripción generada automáticamente

Ilustración 60:Actividades

### Instalaciones

Página con un carousel con varias imágenes de las instalaciones del refugio.

Una captura de pantalla de una red social

Descripción generada automáticamente

### Login

Página con un formulario para realizar el inicio de sesión. Si el usuario está en la base de datos se le redirigirá al perfil.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 61:Login

Para comprobar si el usuario está en la base de datos se usará el usuario comprobar de phpMyAdmin para realizar un SELECT en la tabla usuarios donde el dniUsuario y password tienen que ser los mismos que el DNI y contraseña introducido en el formulario.

Texto

Descripción generada automáticamente

Ilustración 62:Código login

Si el usuario mete sus credenciales correctamente se guardará toda su información en una sesión para posteriormente ser usada en otras página, la sesión logueado cambiará su valor a true y finalmente se le redireccionará al perfil.

Si el usuario mete un DNI no valido se le mostrará un mensaje de error, lo mismo pasa si el DNI no se encuentra en la base de datos.

### Perfil

Dependiendo del rol de usuario que haya iniciado sesión se le mostrarán unas opciones o otras. Esta página tiene una parte común para todos los usuarios en la cual se muestran los datos personales de cada uno.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

Ilustración 63:Información general

### Perfil Gerente

El usuario con rol gerente es el que más opciones tiene ya que tiene que administrar la aplicación web.

Una captura de pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Ilustración 64:Menú gerente

#### Alta animal

Formulario en el que el gerente introducirá los datos de un nuevo animal que entre en El Santuario. Campos del formulario:

* Nombre del animal
* Fecha del ingreso en el refugio
* Especie
* Sexo

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 65:Insertar animal

Para que los datos sean introducidos en la tabla animales antes deben se pasar por las validaciones. Si todos los datos las cumplen se introducen los datos y se recibe un mensaje de respuesta de datos introducidos correctamente. Si uno de estos datos no cumple los requisitos se devolverá un mensaje de error dependiendo la validación.

Texto

Descripción generada automáticamente

Ilustración 66:Código introducir animal

#### Alta voluntario

Otro formulario para introducir los datos de un usuario el cual quiere ayudar a las funciones de El Santuario. Campos del formulario:

* DNI
* Nombre
* Apellidos
* Fecha de Nacimiento
* Teléfono
* Dirección (Opcional)
* Contraseña
* Repetir contraseña

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ilustración 67:Formulario registro nuevo voluntario

Para que se inserten los datos se deben de cumplir todas las validaciones. Sino recibiremos un mensaje de error dependiendo la validación que ha returnado false.

Texto

Descripción generada automáticamente

Ilustración 68:Código formulario nuevo voluntario

#### Nueva actividad

Nueva actividad es un formulario compuesto por varias etiquetas <select> las cuales muestran los nombres de los voluntarios, otro para los nombres de animales que están en el centro y otro para las actividades que se pueden realizar.



Ilustración 69:Formulario nueva actividad

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración 70:Ejemplo de <select>

Para que se inserten los datos se han de cumplir las validaciones de fecha y hora. Estas no pueden ser anteriores a la fecha y hora actual. Sino se cumple una de ellas devolverá un mensaje de error.

Texto

Descripción generada automáticamente

Ilustración 71:Código asignar actividad

#### Nueva adopción

Formulario para la creación de una nueva adopción de un animal. Para ello se deben de introducir los datos de un nuevo usuario al cual se le asignará el rol de adoptante. Los adoptantes sólo podrán tener 1 animal adoptado por usuario.

Interfaz de usuario gráfica, Aplicación, Correo electrónico

Descripción generada automáticamente

Ilustración 72:Formulario nueva adopción

En este formulario aparte de insertar datos en las tablas usuarios y adopciones. Se hará un UPDATE en la tabla animales para agregarle al animal la fecha de adopción y cambiarle el valor de aniAdop a 1 para saber que ha sido adoptado.

Texto

Descripción generada automáticamente

Ilustración 73:Código nueva adopción

#### Ver animales actuales

Al pulsar en este botón se nos mostrará una tabla con todos los animales que no han sido aún adoptados(valor de aniAdop en 0).

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración 74:Tabla animales actuales

Texto

Descripción generada automáticamente

Ilustración 75:Código ver animales actuales

#### Ver todos los animales

Nos mostrará una tabla con todos los animales que han estado en algún momento en el refugio, los adoptados y los que aún siguen sin ser adoptados.

Tabla

Descripción generada automáticamente

Ilustración 76:Tabla ver todos los animales

Texto

Descripción generada automáticamente

Ilustración 77:Código ver todos los animales

### Perfil Voluntario

El usuario con rol de voluntario tendrá dos opciones en la aplicación; ver las actividades y ver los animales de los cuales se encarga.

Imagen de la pantalla de un celular de un mensaje en letras blancas

Descripción generada automáticamente con confianza baja

Ilustración 78:Perfil voluntario

#### Ver animales a cargo

Mostrará una tabla con todos los animales de los cuales es responsable y tiene que hacer actividades con ellos.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 79:Tabla animales a cargo

Para ello se debe hacer un SELECT en el que el id del animal en la tabla animales sea igual al id del animal en la tabla actividades y el dniVoluntario en la tabla actividades sea igual al DNI del usuario logeado.

Texto

Descripción generada automáticamente

Ilustración 80:Código ver animales a cargo

#### Ver actividades

Mostrará una tabla con todas las actividades que tiene asignadas el voluntario logeado. Este puede modificar la hora y la fecha de cada actividad pulsando en el botón de la actividad que desee modificar.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 81:Tabla ver actividades voluntario

Para diferenciar una actividad de otra al pulsar el botón se usará la función implode() explicada en las funciones personalizadas. Texto

Descripción generada automáticamente

Ilustración 82:Código ver actividades voluntario

Para que los datos sean modificados correctamente en la base de datos se han de cumplir las funciones de validación de la hora y fecha (no ser anteriores a la fecha y hora actual)

Texto

Descripción generada automáticamente

Ilustración 83:Código modificar fecha/hora

### Perfil adoptante

Podrá realizar dos funciones; ver la información del animal que ha adoptado y cambiar su contraseña ya que el gerente le dará una contraseña aleatoria.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 84:Perfil adoptante

#### Ver animales adoptados

Mostrará una tabla con toda la información del animal que ha adoptado. Para ello se realizar un SELECT de las tablas animales y adopciones donde el dniAdoptante de la tabla adopciones debe ser igual al DNI del usuario logeado.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 85:Tabla ver animales adopciones

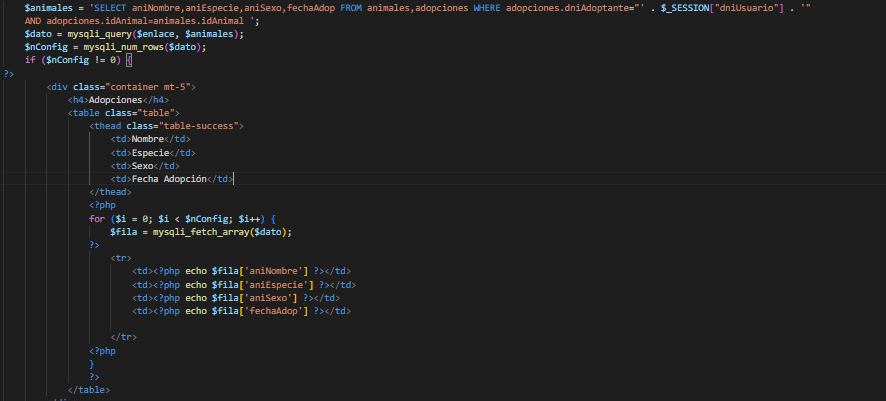


Ilustración 86:Código ver animales adoptados

#### Cambiar contraseña

Formulario con dos campos, el primero para introducir la contraseña y el segundo para confirmar que la contraseña introducida sea la misma del primer campo.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 87:Formulario cambio de contraseña

Si las contraseñas no coinciden se vaciará el formulario y recibiremos un mensaje de error.

Texto

Descripción generada automáticamente

Ilustración 88:Código formulario cambiar contraseña

### Cerrar sesión

Al pulsar el botón del header de cerrar sesión hará una llamada al archivo cerrar.php en el cual hay un sesssion\_destroy y nos redirigirá a la pantalla de home.

Texto

Descripción generada automáticamente

Ilustración 89:Código cerrar sesión

# Índice de figuras

[Ilustración 1:Diagrama E/R 5](#_Toc105783553)

[Ilustración 2:Tabla usuarios 5](#_Toc105783554)

[Ilustración 3:Tabla animales 6](#_Toc105783555)

[Ilustración 4:Tabla adopciones 6](#_Toc105783556)

[Ilustración 5:Índices Tabla adopciones 6](#_Toc105783557)

[Ilustración 6:Tabla actividades 7](#_Toc105783558)

[Ilustración 7:Índices Tabla actividades 7](#_Toc105783559)

[Ilustración 8:Tabla tipoActividades 7](#_Toc105783560)

[Ilustración 9:Privilegios cuenta gerente 8](#_Toc105783561)

[Ilustración 10:Privilegios cuenta voluntario 8](#_Toc105783562)

[Ilustración 11:Privilegios cuenta adoptante 9](#_Toc105783563)

[Ilustración 12:Permisos adoptante en la tabla adopciones 9](#_Toc105783564)

[Ilustración 13:Permisos adoptante en la tabla animales 9](#_Toc105783565)

[Ilustración 14:Permisos adoptante en la tabla usuarios 10](#_Toc105783566)

[Ilustración 15:Privilegios cuenta comprobar 10](#_Toc105783567)

[Ilustración 16:Conexión a la base de datos como comprobar 11](#_Toc105783568)

[Ilustración 17:Conexión a la base de datos como gerente 11](#_Toc105783569)

[Ilustración 18:Conexión a la base de datos como voluntario 11](#_Toc105783570)

[Ilustración 19:Conexión a la base de datos como adoptante 11](#_Toc105783571)

[Ilustración 20:Ejemplo mysqli\_query de consulta SELECT 12](#_Toc105783572)

[Ilustración 21:Ejemplo mysqli\_query de consulta UPDATE 12](#_Toc105783573)

[Ilustración 22:Ejemplo mysqli\_query de consulta INSERT 12](#_Toc105783574)

[Ilustración 23:Ejemplo mysqli\_num\_rows 13](#_Toc105783575)

[Ilustración 24:Ejemplo utilización mysqli\_fetch\_array 13](#_Toc105783576)

[Ilustración 25:Función hash\_hmac 14](#_Toc105783577)

[Ilustración 26:Función isset 15](#_Toc105783578)

[Ilustración 27:Función header 15](#_Toc105783579)

[Ilustración 28:Función implode 16](#_Toc105783580)

[Ilustración 29:Función explode 16](#_Toc105783581)

[Ilustración 30:Funciones str\_shuffle y substr 17](#_Toc105783582)

[Ilustración 31:Archivo functions.php 17](#_Toc105783583)

[Ilustración 32:Función get\_header 17](#_Toc105783584)

[Ilustración 33:Función get\_footer 18](#_Toc105783585)

[Ilustración 34:Función validar\_tel 18](#_Toc105783586)

[Ilustración 35:Función validar\_dni 18](#_Toc105783587)

[Ilustración 36:Funciones validar\_nombre y validar\_ape 19](#_Toc105783588)

[Ilustración 37:Función calculaedad 19](#_Toc105783589)

[Ilustración 38:Función validar\_hora 20](#_Toc105783590)

[Ilustración 39:Función validar\_fecha actividades 20](#_Toc105783591)

[Ilustración 40:Función validar\_fecha voluntario 20](#_Toc105783592)

[Ilustración 41:Función validar\_fecha adoptante 21](#_Toc105783593)

[Ilustración 42:Página descarga XAMPP 22](#_Toc105783594)

[Ilustración 43:Instalación XAMPP 1 22](#_Toc105783595)

[Ilustración 44:Instalación XAMPP 2 23](#_Toc105783596)

[Ilustración 45:Instalación XAMPP 3 23](#_Toc105783597)

[Ilustración 46:Instalación XAMPP 4 24](#_Toc105783598)

[Ilustración 47:Instalación XAMPP 5 24](#_Toc105783599)

[Ilustración 48:Configuración archivo hosts 25](#_Toc105783600)

[Ilustración 49:Configuración archivo httpd-vhost 25](#_Toc105783601)

[Ilustración 50:Índex phpMyAdmin 26](#_Toc105783602)

[Ilustración 51:Crear base de datos 26](#_Toc105783603)

[Ilustración 52:Importación base de datos 27](#_Toc105783604)

[Ilustración 53:Creación de usuarios en la base de datos 27](#_Toc105783605)

[Ilustración 54:Cuentas de usuarios 27](#_Toc105783606)

[Ilustración 55:Header sin iniciar sesión 28](#_Toc105783607)

[Ilustración 56:Header con sesión iniciada 28](#_Toc105783608)

[Ilustración 57:Código header 28](#_Toc105783609)

[Ilustración 58:Footer 28](#_Toc105783610)

[Ilustración 59:Índex 29](#_Toc105783611)

[Ilustración 60:Actividades 29](#_Toc105783612)

[Ilustración 61:Login 30](#_Toc105783613)

[Ilustración 62:Código login 31](#_Toc105783614)

[Ilustración 63:Información general 32](#_Toc105783615)

[Ilustración 64:Menú gerente 32](#_Toc105783616)

[Ilustración 65:Insertar animal 33](#_Toc105783617)

[Ilustración 66:Código introducir animal 33](#_Toc105783618)

[Ilustración 67:Formulario registro nuevo voluntario 34](#_Toc105783619)

[Ilustración 68:Código formulario nuevo voluntario 35](#_Toc105783620)

[Ilustración 69:Formulario nueva actividad 35](#_Toc105783621)

[Ilustración 70:Ejemplo de <select> 36](#_Toc105783622)

[Ilustración 71:Código asignar actividad 36](#_Toc105783623)

[Ilustración 72:Formulario nueva adopción 37](#_Toc105783624)

[Ilustración 73:Código nueva adopción 37](#_Toc105783625)

[Ilustración 74:Tabla animales actuales 38](#_Toc105783626)

[Ilustración 75:Código ver animales actuales 38](#_Toc105783627)

[Ilustración 76:Tabla ver todos los animales 39](#_Toc105783628)

[Ilustración 77:Código ver todos los animales 39](#_Toc105783629)

[Ilustración 78:Perfil voluntario 40](#_Toc105783630)

[Ilustración 79:Tabla animales a cargo 40](#_Toc105783631)

[Ilustración 80:Código ver animales a cargo 40](#_Toc105783632)

[Ilustración 81:Tabla ver actividades voluntario 41](#_Toc105783633)

[Ilustración 82:Código ver actividades voluntario 41](#_Toc105783634)

[Ilustración 83:Código modificar fecha/hora 42](#_Toc105783635)

[Ilustración 84:Perfil adoptante 42](#_Toc105783636)

[Ilustración 85:Tabla ver animales adopciones 42](#_Toc105783637)

[Ilustración 86:Código ver animales adoptados 43](#_Toc105783638)

[Ilustración 87:Formulario cambio de contraseña 43](#_Toc105783639)

[Ilustración 88:Código formulario cambiar contraseña 43](#_Toc105783640)

[Ilustración 89:Código cerrar sesión 44](#_Toc105783641)

# Bibliografía

PHP.net manual de php en el cual encontrar toda la documentación sobre las funciones.(<https://www.php.net/manual/es/function>)