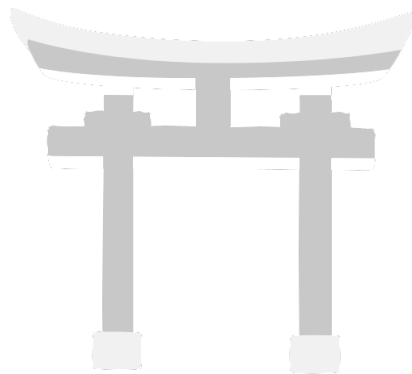


CFGS: DESARROLLO DE APLICACIONES WEB

CURSO:2021-2022



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
Descripción de la base de datos.....	4
Estructura.....	4
Tabla usuarios	5
Tabla animales	6
Tabla adopciones	6
Tabla actividades.....	7
Tabla tipoactividades	7
Cuentas de usuario de la base.....	7
Cuenta de gerente	7
Cuenta de voluntario.....	8
Cuenta de adoptante	8
Cuenta de comprobar	10
Definición y descripción de variables usadas en el sistema.....	10
Funciones MySQL.....	10
mysqli_connect.....	10
mysqli_query	12
mysqli_num_rows.....	12
mysqli_fetch_array.....	13
Funciones de PHP	14
hash_hmac.....	14
isset.....	14
header	15
implode.....	15
explode.....	16
str_shuffle.....	16
Substr	16
Funciones personalizadas	17
get_header	17
get_footer	17
Funciones para la validación de datos.....	18
validar_tel	18
validar_dni	18
validar_nombre y validar_ape.....	19
calculaedad	19

validar_hora.....	19
validar_fecha	20
Despliegue de la aplicación	21
Instalación.....	21
Configuración de la base de datos	26
Estructura de la aplicación	28
Header.....	28
Footer	28
Índex (Home).....	29
Actividades	29
Instalaciones.....	30
Login.....	30
Perfil	31
Perfil Gerente	32
Perfil Voluntario	40
Perfil adoptante.....	42
Cerrar sesión	44
Índice de figuras	45
Bibliografía.....	47

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

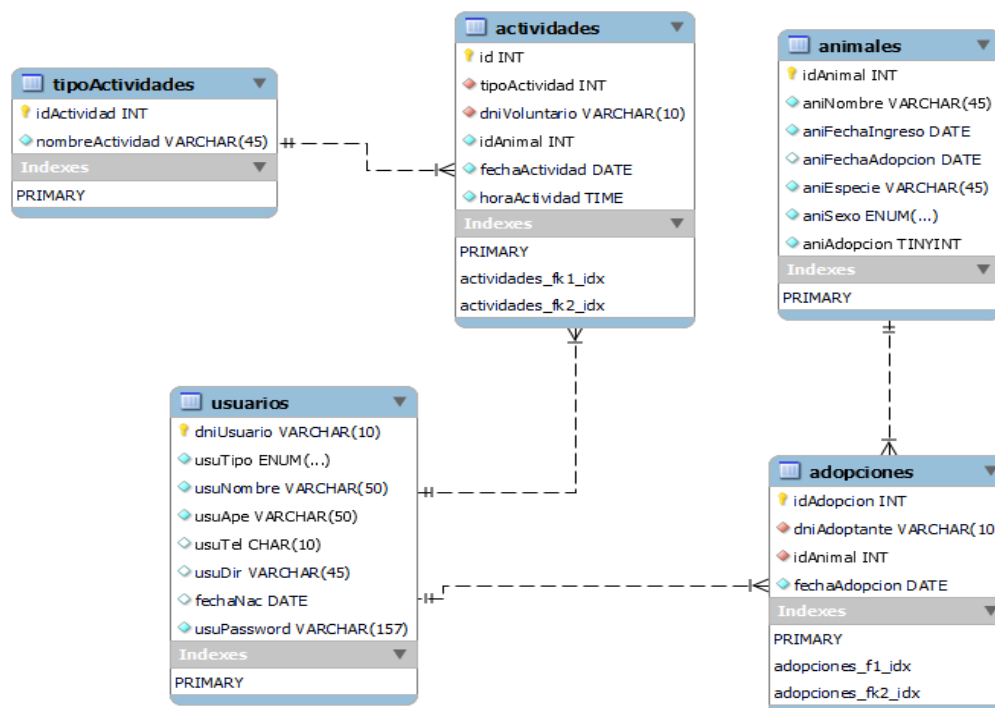


Ilustración 1:Diagrama E/R

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)

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 dniUsuario	varchar(10)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	2 usuPassword	varchar(157)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3 usuTipo	enum('Gerente', 'Voluntario', 'Adoptante')	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	4 usuNombre	varchar(50)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	5 usuApe	varchar(50)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	6 usuTel	char(10)	utf8mb4_general_ci		Sí	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	7 usuDir	varchar(50)	utf8mb4_general_ci		Sí	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	8 fechaNac	date			Sí	NULL			Cambiar Eliminar Más

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.
- aniAdopcion: valor 0 hasta que el animal es adoptado que se modificará al valor 1.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/> 1	idAnimal	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/> 2	aniNombre	varchar(20)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 3	aniFechaIngreso	date			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 4	aniFechaAdop	date			Sí	NULL			Cambiar Eliminar Más
<input type="checkbox"/> 5	aniEspecie	varchar(20)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 6	aniSexo	enum('Hembra', 'Macho')	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 7	aniAdop	tinyint(1)			No	0			Cambiar Eliminar Más

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.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/> 1	idAdopcion	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/> 2	dniAdoptante	varchar(10)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 3	idAnimal	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 4	fechaAdop	date			No	Ninguna			Cambiar Eliminar Más

Ilustración 4:Tabla adopciones

Índices										
Acción	Nombre de la clave		Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
Editar Rename Eliminar	PRIMARY		BTREE	Sí	No	idAdopcion	2	A	No	
Editar Rename Eliminar	adopciones_ibfk_1		BTREE	No	No	dniAdoptante	2	A	No	
Editar Rename Eliminar	adopciones_ibfk_2		BTREE	No	No	idAnimal	2	A	No	

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.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/>	2 tipoActividad	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3 dniVoluntario	varchar(10)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	4 idAnimal	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	5 fechaActividad	date			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	6 horaActividad	time			No	Ninguna			Cambiar Eliminar Más

Ilustración 6:Tabla actividades

Índices										
Acción	Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario	
Editar Rename Eliminar	PRIMARY	BTREE	Sí	No	id	3	A	No		
Editar Rename Eliminar	actividades_ibfk_1	BTREE	No	No	tipoActividad	3	A	No		
Editar Rename Eliminar	actividades_ibfk_2	BTREE	No	No	dniVoluntario	3	A	No		
Editar Rename Eliminar	actividades_ibfk_3	BTREE	No	No	idAnimal	3	A	No		

Ilustración 7:Índices Tabla actividades

Tabla tipoactividades

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

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 idActividad	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/>	2 nombreActividad	varchar(40)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más

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.

Editar los privilegios: Cuenta de usuario '*gerente*'@'localhost' - Base de datos *elsantuario*

Tabla	Privilegios	Conceder	Privilegios específicos para la columna	Acción
actividades	SELECT, INSERT, UPDATE	No	No	Editar privilegios Revocar
adopciones	SELECT, INSERT, UPDATE	No	No	Editar privilegios Revocar
animales	SELECT, INSERT, UPDATE	No	No	Editar privilegios Revocar
tipoactividades	SELECT	No	No	Editar privilegios Revocar
usuarios	SELECT, INSERT	No	No	Editar privilegios Revocar

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.

Editar los privilegios: Cuenta de usuario '*voluntario*'@'localhost' - Base de datos *elsantuario*

Tabla	Privilegios	Conceder	Privilegios específicos para la columna	Acción
actividades	SELECT	No	Sí	Editar privilegios Revocar
animales	SELECT	No	No	Editar privilegios Revocar
tipoactividades	SELECT	No	No	Editar privilegios Revocar

Añadir privilegios a esta tabla:

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.

Editar los privilegios: Cuenta de usuario '*adoptante*'@'*localhost*' - Base de datos *elsantuario*

Privilegios específicos para la tabla					
Tabla	Privilegios	Conceder	Privilegios específicos para la columna	Acción	
adopciones	USAGE	No	Sí	 Editar privilegios	 Revocar
animales	USAGE	No	Sí	 Editar privilegios	 Revocar
usuarios	USAGE	No	Sí	 Editar privilegios	 Revocar

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.

Editar los privilegios: Cuenta de usuario '*adoptante*'@'*localhost*' - Base de datos *elsantuario* - Tabla *adopciones*

Privilegios específicos para la tabla

Nota: Los nombres de los privilegios de MySQL están expresados en inglés.

SELECT	INSERT	UPDATE	REFERENCES	DELETE
<input type="checkbox"/> idAdopcion <input checked="" type="checkbox"/> dniAdoptante <input checked="" type="checkbox"/> idAnimal <input checked="" type="checkbox"/> fechaAdop	<input type="checkbox"/> idAdopcion <input type="checkbox"/> dniAdoptante <input type="checkbox"/> fechaAdop	<input type="checkbox"/> idAdopcion <input type="checkbox"/> dniAdoptante <input type="checkbox"/> idAnimal <input type="checkbox"/> fechaAdop	<input type="checkbox"/> idAdopcion <input type="checkbox"/> dniAdoptante <input type="checkbox"/> idAnimal <input type="checkbox"/> fechaAdop	<input type="checkbox"/> CREATE <input type="checkbox"/> DROP <input type="checkbox"/> GRANT <input type="checkbox"/> INDEX <input type="checkbox"/> ALTER <input type="checkbox"/> CREATE VIEW <input type="checkbox"/> SHOW VIEW <input type="checkbox"/> TRIGGER <input type="checkbox"/> DELETE HISTORY

☐ Ninguno
 ☐ Ninguno
 ☐ Ninguno
 ☐ Ninguno

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.

Editar los privilegios: Cuenta de usuario '*adoptante*'@'*localhost*' - Base de datos *elsantuario* - Tabla *animales*

Privilegios específicos para la tabla

Nota: Los nombres de los privilegios de MySQL están expresados en inglés.

SELECT	INSERT	UPDATE	REFERENCES	DELETE
<input type="checkbox"/> idAnimal <input checked="" type="checkbox"/> aniNombre <input checked="" type="checkbox"/> aniFechaIngres <input checked="" type="checkbox"/> aniFechaAdop <input checked="" type="checkbox"/> aniEspecie <input checked="" type="checkbox"/> aniSexo <input type="checkbox"/> aniAdop	<input type="checkbox"/> idAnimal <input type="checkbox"/> aniNombre <input type="checkbox"/> aniFechaIngres <input type="checkbox"/> aniFechaAdop <input type="checkbox"/> aniEspecie <input type="checkbox"/> aniSexo <input type="checkbox"/> aniAdop	<input type="checkbox"/> idAnimal <input type="checkbox"/> aniNombre <input type="checkbox"/> aniFechaIngres <input type="checkbox"/> aniFechaAdop <input type="checkbox"/> aniEspecie <input type="checkbox"/> aniSexo <input type="checkbox"/> aniAdop	<input type="checkbox"/> idAnimal <input type="checkbox"/> aniNombre <input type="checkbox"/> aniFechaIngres <input type="checkbox"/> aniFechaAdop <input type="checkbox"/> aniEspecie <input type="checkbox"/> aniSexo <input type="checkbox"/> aniAdop	<input type="checkbox"/> CREATE <input type="checkbox"/> DROP <input type="checkbox"/> GRANT <input type="checkbox"/> INDEX <input type="checkbox"/> ALTER <input type="checkbox"/> CREATE VIEW <input type="checkbox"/> SHOW VIEW <input type="checkbox"/> TRIGGER <input type="checkbox"/> DELETE HISTORY

☐ Ninguno
 ☐ Ninguno
 ☐ Ninguno
 ☐ Ninguno

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.

Editar los privilegios: Cuenta de usuario '*adoptante*'@'localhost' - Base de datos *elsantuario* - Tabla *usuarios*

Privilegios específicos para la tabla

Nota: Los nombres de los privilegios de MySQL están expresados en inglés.

SELECT	INSERT	UPDATE	REFERENCES	DELETE
<input checked="" type="checkbox"/> dniUsuario	<input type="checkbox"/> dniUsuario	<input type="checkbox"/> dniUsuario	<input type="checkbox"/> dniUsuario	<input type="checkbox"/> CREATE
<input type="checkbox"/> usuPassword	<input type="checkbox"/> usuPassword	<input checked="" type="checkbox"/> usuPassword	<input type="checkbox"/> usuPassword	<input type="checkbox"/> DROP
<input type="checkbox"/> usuTipo	<input type="checkbox"/> usuTipo	<input type="checkbox"/> usuTipo	<input type="checkbox"/> usuTipo	<input type="checkbox"/> GRANT
<input type="checkbox"/> usuNombre	<input type="checkbox"/> usuNombre	<input type="checkbox"/> usuNombre	<input type="checkbox"/> usuNombre	<input type="checkbox"/> INDEX
<input type="checkbox"/> usuApellido	<input type="checkbox"/> usuApellido	<input type="checkbox"/> usuApellido	<input type="checkbox"/> usuApellido	<input type="checkbox"/> ALTER
<input type="checkbox"/> usuTel	<input type="checkbox"/> usuTel	<input type="checkbox"/> usuTel	<input type="checkbox"/> usuTel	<input type="checkbox"/> CREATE VIEW
<input type="checkbox"/> usuDir	<input type="checkbox"/> usuDir	<input type="checkbox"/> usuDir	<input type="checkbox"/> usuDir	<input type="checkbox"/> SHOW VIEW
<input type="checkbox"/> fechaNac	<input type="checkbox"/> fechaNac	<input type="checkbox"/> fechaNac	<input type="checkbox"/> fechaNac	<input type="checkbox"/> TRIGGER
<input type="checkbox"/> Ninguno	<input type="checkbox"/> Ninguno	<input type="checkbox"/> Ninguno	<input type="checkbox"/> Ninguno	<input type="checkbox"/> DELETE HISTORY

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.

Editar los privilegios: Cuenta de usuario '*comprobar*'@'localhost' - Base de datos *elsantuario* - Tabla *usuarios*

Privilegios específicos para la tabla

Nota: Los nombres de los privilegios de MySQL están expresados en inglés.

SELECT	INSERT	UPDATE	REFERENCES	DELETE
<input checked="" type="checkbox"/> dniUsuario	<input type="checkbox"/> dniUsuario	<input type="checkbox"/> dniUsuario	<input type="checkbox"/> dniUsuario	<input type="checkbox"/> CREATE
<input type="checkbox"/> usuPassword	<input type="checkbox"/> usuPassword	<input checked="" type="checkbox"/> usuPassword	<input type="checkbox"/> usuPassword	<input type="checkbox"/> DROP
<input type="checkbox"/> usuTipo	<input type="checkbox"/> usuTipo	<input type="checkbox"/> usuTipo	<input type="checkbox"/> usuTipo	<input type="checkbox"/> GRANT
<input type="checkbox"/> usuNombre	<input type="checkbox"/> usuNombre	<input type="checkbox"/> usuNombre	<input type="checkbox"/> usuNombre	<input type="checkbox"/> INDEX
<input type="checkbox"/> usuApellido	<input type="checkbox"/> usuApellido	<input type="checkbox"/> usuApellido	<input type="checkbox"/> usuApellido	<input type="checkbox"/> ALTER
<input type="checkbox"/> usuTel	<input type="checkbox"/> usuTel	<input type="checkbox"/> usuTel	<input type="checkbox"/> usuTel	<input type="checkbox"/> CREATE VIEW
<input type="checkbox"/> usuDir	<input type="checkbox"/> usuDir	<input type="checkbox"/> usuDir	<input type="checkbox"/> usuDir	<input type="checkbox"/> SHOW VIEW
<input type="checkbox"/> fechaNac	<input type="checkbox"/> fechaNac	<input type="checkbox"/> fechaNac	<input type="checkbox"/> fechaNac	<input type="checkbox"/> TRIGGER
<input type="checkbox"/> Ninguno	<input type="checkbox"/> Ninguno	<input type="checkbox"/> Ninguno	<input type="checkbox"/> Ninguno	<input type="checkbox"/> DELETE HISTORY

Continuar

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.

```
$enlace = mysqli_connect("localhost", "comprobar", "comprobar", "elsantuario");
```

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.

```
$enlace = mysqli_connect("localhost", "gerente", "gerente", "elsantuario");
```

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.

```
$enlace = mysqli_connect("localhost", "voluntario", "voluntario", "elsantuario");
```

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.

```
$enlace = mysqli_connect("localhost", "adoptante", "adoptante", "elsantuario");
```

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`

```
$enlace = mysqli_connect("localhost", "adoptante", "adoptante", "elsantuario");
$animales = 'SELECT aniNombre, aniEspecie, aniSexo, fechaAdop FROM animales, adopciones WHERE adopciones.dniAdoptante="' . $_SESSION["dniUsuario"] . '"
AND adopciones.idAnimal=animales.idAnimal ' ;
$dato = mysqli_query($enlace, $animales);
```

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`.

```
$enlace = mysqli_connect("localhost", "voluntario", "voluntario", "elsantuario");
$edit = "UPDATE actividades SET fechaActividad='$fecha', horaActividad='$hora' WHERE tipoActividad=' $datos[0] ' ";
$introducir = mysqli_query($enlace, $edit);
```

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.

```
$enlace = mysqli_connect("localhost", "gerente", "gerente", "elsantuario");
$consulta = "INSERT INTO animales (idAnimal, aniNombre, aniFechaIngreso, aniFechaAdop, aniEspecie, aniSexo, aniAdop)
VALUES (NULL, '{$_POST["aniNombre"]}', '{$_POST["aniFechaIngreso"]}', NULL, '{$_POST["aniEspecie"]}', '{$_POST["aniSexo"]}', '0')";
$introducir = mysqli_query($enlace, $consulta);
```

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.

```
$usuario = 'SELECT dniUsuario,usuPassword FROM usuarios WHERE dniUsuario=" . $_POST["dniUsuario"] . "'';
$dato = mysqli_query($enlace, $usuario);

if (mysqli_num_rows($dato) === 0) {

    $consulta = "INSERT INTO usuarios (dniUsuario,usuPassword,usuTipo,usuNombre,usuApellido,usuTel,usuDir,fechaNac)
    VALUES ('$_POST["dniUsuario"]','$_POST["cifrado"]','Voluntario','$_POST["nombre"]','$_POST["ape"]','$_POST["telefono"]','$_POST["direccion"]','$_POST["fechaNac"])';
    $introducir = mysqli_query($enlace, $consulta);
}
```

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

```
$animales = 'SELECT * FROM animales WHERE aniAdop=0';
$dato = mysqli_query($enlace, $animales);
$nConfig = mysqli_num_rows($dato);

if ($nConfig != 0) {
    <div class="container mt-5">
        <h3 class="login-heading mb-4">Animales actualmente en EL SANTUARIO</h3>
        <table class="table">
            <thead class="table-success">
                <td>Nombre</td>
                <td>Especie</td>
                <td>Sexo</td>
                <td>Fecha Ingreso</td>
                <td>Adoptado</td>
                <td>Fecha Adopción</td>
            </thead>
            <tbody>
                <tr>
                    <td><?php echo $fila['aniNombre'] ?></td>
                    <td><?php echo $fila['aniEspecie'] ?></td>
                    <td><?php echo $fila['aniSexo'] ?></td>
                    <td><?php echo $fila['aniFechaIngreso'] ?></td>
                    <td><?php echo "NO" ?></td>
                    <td><?php echo "-" ?></td>
                </tr>
            </tbody>
        </table>
    </div>
}
```

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.

```
$cifrada = hash_hmac('sha512', $_POST["password"], "primeraweb");
```

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.

```

if (isset($_POST["verAnimalCargo"])) {
    $_SESSION['action'] = 'verAnimalCargo';
    header("Location: voluntario.php");
} elseif (isset($_POST["verActividades"])) {
    $_SESSION['action'] = 'verActividades';
    header("Location: voluntario.php");
} elseif (isset($_POST["cambiarFecha"])) {
    $_SESSION['action'] = 'cambiarFecha';
    header("Location: voluntario.php");
}

```

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 de sesión y se redirigirá a voluntario.php.

header

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

```

} else {
    $_SESSION['logueado'] = true;
    header("Location: index.php");
}

```

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.

```

$fila = mysqli_fetch_array($dato);

<tr>
  <td><?php echo $fila['nombreActividad'] ?></td>
  <td><?php echo $fila['aniNombre'] ?></td>
  <td><?php echo $fila['fechaActividad'] ?></td>
  <td><?php echo $fila['horaActividad'] ?></td>
  <td>
    <p>
      <form action='#' method='POST'>
        <button type="submit" name="cambio" value="<?php echo implode(",", $fila); ?>"
      </form>
    </td>
</tr>

```

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.

```
$datos = explode(",", $_SESSION["cambio"]);
```

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.


```
$permitted_chars = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';  
$contra = substr(str_shuffle($permitted_chars), 0, 20);
```

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.

```
<?php  
class functions{  
    public static function get_header(){  
        include('header.php');  
    }  
  
    public static function get_footer(){  
        include('footer.php');  
    }  
}
```

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.

```
1 <?php  
2  
3 require('functions.php');  
4 functions::get_header();  
5
```

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.

```

762
763     <?php
764     functions::get_footer();
765     ?>

```

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.

```

function validar_tel($telefono)
{
    if (!preg_match("/^[0-9]{9}$/", $telefono)) {
        return false;
    } else {
        return true;
    }
}

```

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 números 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.

```

function validar_dni($dni)
{
    $letra = substr($dni, -1);
    $numeros = substr($dni, 0, -1);
    if (substr("TRWAGMYFPDXBNJZSQVHLCKE", $numeros % 23, 1) == $letra && strlen($letra) == 1 && strlen($numeros) == 8) {
        return true;
    } else {
        return false;
    }
}

```

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.

```
function validar_nombre($name)
{
    if (!preg_match("/^[a-zA-ZÀ-ÖØ-öø-ÿ]+\.(?([a-zA-ZÀ-ÖØ-öø-ÿ]+\.)*)+$/", $name)) {
        return false;
    } else {
        return true;
    }
}

function validar_ape($ape)
{
    if (!preg_match("/^[a-zA-ZÀ-ÖØ-öø-ÿ]+\.(?([a-zA-ZÀ-ÖØ-öø-ÿ]+\.)*)+$/", $ape)) {
        return false;
    } else {
        return true;
    }
}
```

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.

```
function calculaedad($nacimiento)
{
    list($ano, $mes, $dia) = explode("-", $nacimiento);
    $ano_diferencia = date("Y") - $ano;
    $mes_diferencia = date("m") - $mes;
    $dia_diferencia = date("d") - $dia;
    if ($dia_diferencia < 0 || $mes_diferencia < 0)
    {
        $ano_diferencia--;
    }
    return $ano_diferencia;
}
```

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.

```
function validar_hora($hora, $fecha)
{
    $hoy = date("Y-m-d");
    $horaAc = date('H:i:s');

    if ($hoy == $fecha) {
        if ($horaAc < $hora) {
            return false;
        } else {
            return true;
        }
    } else {
        return true;
    }
}
```

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.

```
function validar_fecha($fecha)
{
    $hoy = date("Y-m-d");
    if ($hoy > $fecha) {
        return false;
    } else {
        return true;
    }
}
```

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.

```
function validar_fecha($nacimiento)
{
    $hoy = date("Y-m-d");
    if ($hoy <= $nacimiento || 1900 > $nacimiento || 2014 < $nacimiento) {
        return false;
    } else {
        return true;
    }
}
```

Ilustración 40: Función validar_fecha voluntario

Por último, esta 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.

```
function validar_fecha($nacimiento)
{
    $edad = calculaedad($nacimiento);
    if ($edad <= 18) {
        return false;
    } else {
        return true;
    }
}
```

Ilustración 41: Función `validar_fecha` adoptante

Despliegue de la aplicación

Instalación

Lo primero que necesitamos instalar es 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](http://apachefriends.org) en la cual se encuentra el instalador. Y descargaremos la versión para Windows más reciente.



Ilustración 42: Página descarga XAMPP

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

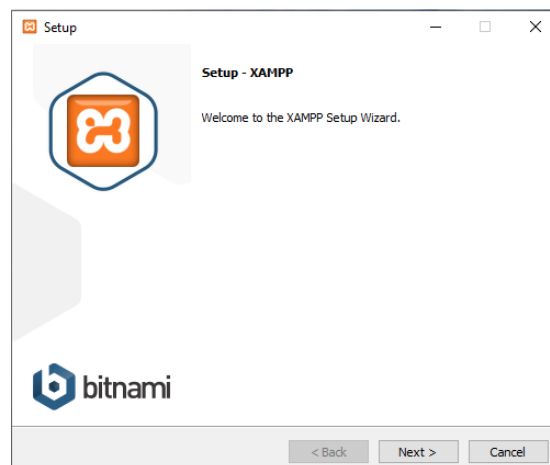


Ilustración 43: Instalación XAMPP 1

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.

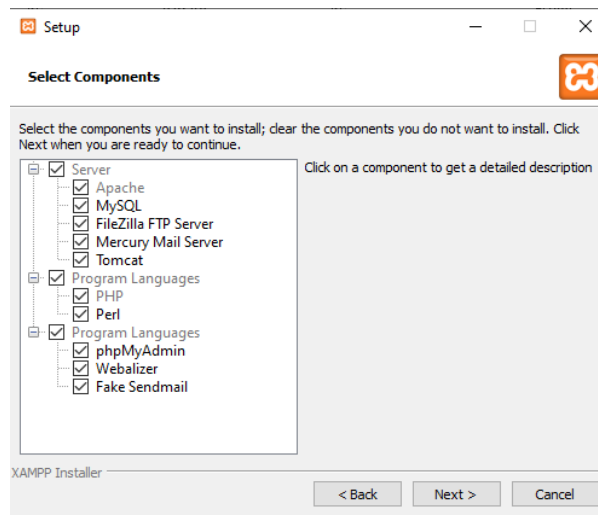


Ilustración 44: Instalación XAMPP 2

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.

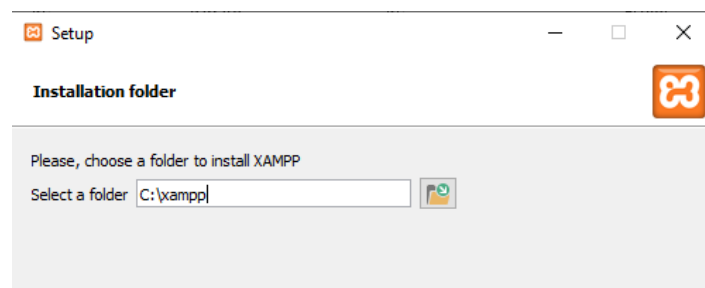


Ilustración 45: Instalación XAMPP 3

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



Ilustración 46: Instalación XAMPP 4

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

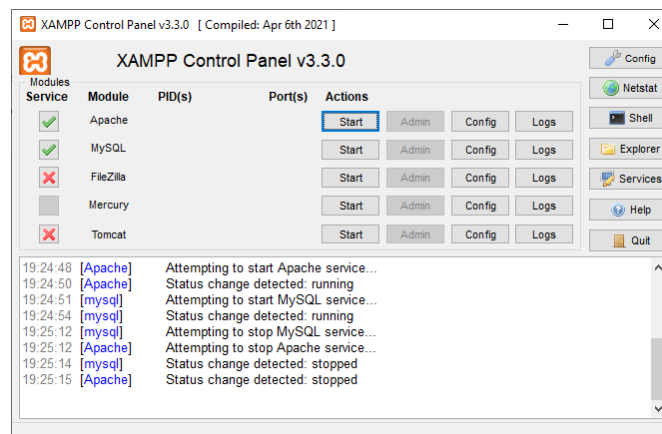


Ilustración 47: Instalación XAMPP 5

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


```

*hosts: Bloc de notas
Archivo Edición Formato Ver Ayuda
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com          # source server
#       38.25.63.10       x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1         localhost
#       ::1               localhost

127.0.0.1 elsantuario.es
|

```

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

```

<VirtualHost *:80>
    DocumentRoot "C:\xampp\htdocs\ElSantuario"
    ServerName elsantuario.org
    <Directory "C:\xampp\htdocs\ElSantuario">
        Require all granted
    </Directory>
</VirtualHost>

```

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/>

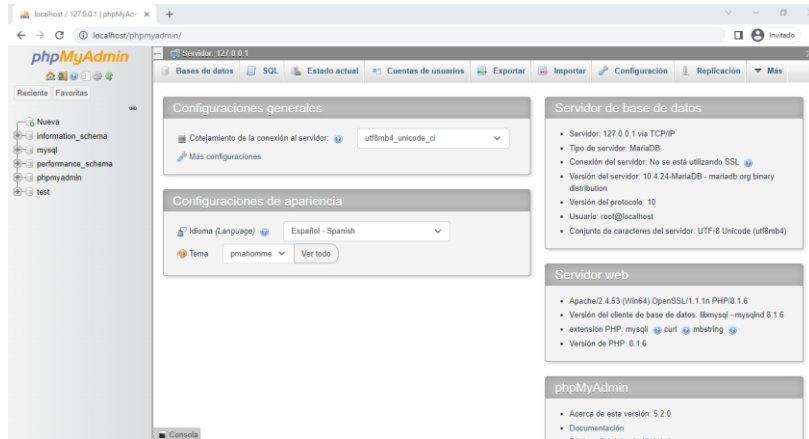


Ilustración 50:Índice phpMyAdmin

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.

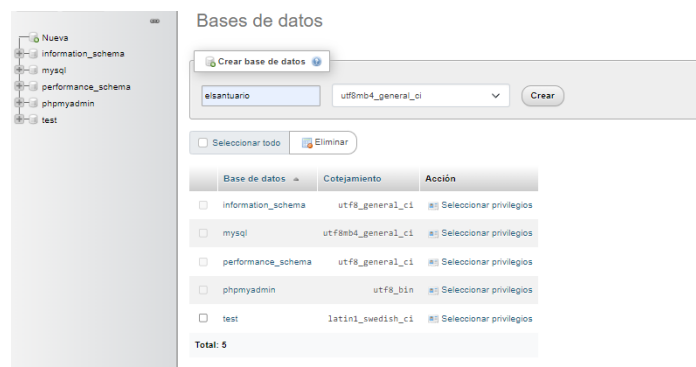


Ilustración 51:Crear base de datos

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.

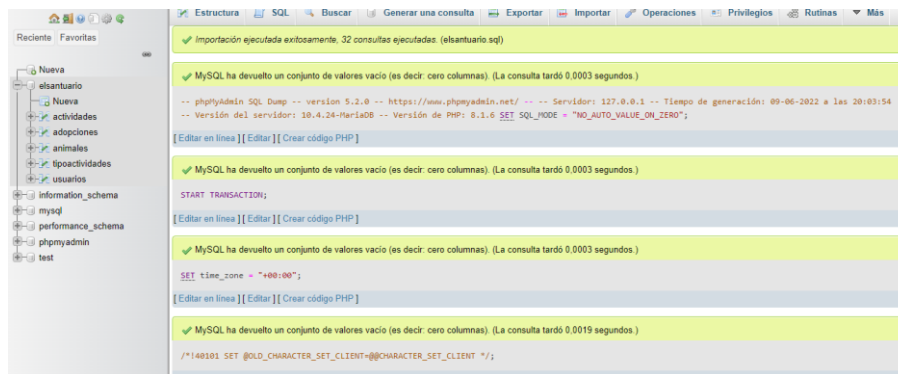


Ilustración 52: Importación base de datos

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.

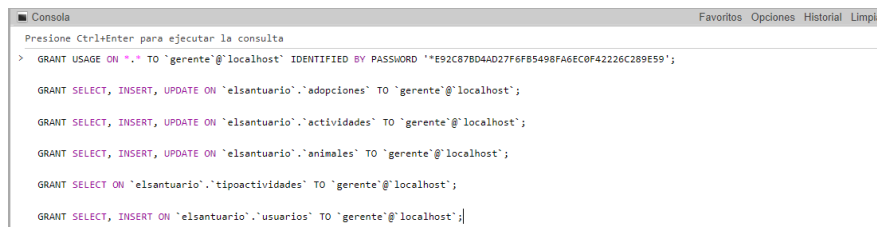


Ilustración 53: Creación de usuarios en la base de datos

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.

Vista global de las cuentas de usuario

	Nombre de usuario	Nombre del servidor	Contraseña	Privilegios globales	Grupo de usuario	Conceder	Acción
<input type="checkbox"/>	cualquiera	%	No	USAGE		No	Editar privilegios Exportar Bloquear
<input type="checkbox"/>	adoptante	localhost	Sí	USAGE		No	Editar privilegios Exportar Bloquear
<input type="checkbox"/>	comprobar	localhost	Sí	USAGE		No	Editar privilegios Exportar Bloquear
<input type="checkbox"/>	gerente	localhost	Sí	USAGE		No	Editar privilegios Exportar Bloquear
<input type="checkbox"/>	pma	localhost	No	USAGE		No	Editar privilegios Exportar Bloquear
<input type="checkbox"/>	root	127.0.0.1	No	ALL PRIVILEGES		Sí	Editar privilegios Exportar Bloquear
<input type="checkbox"/>	root	:::1	No	ALL PRIVILEGES		Sí	Editar privilegios Exportar Bloquear
<input type="checkbox"/>	root	localhost	No	ALL PRIVILEGES		Sí	Editar privilegios Exportar Bloquear
<input type="checkbox"/>	voluntario	localhost	Sí	USAGE		No	Editar privilegios Exportar Bloquear

☐ Seleccionar todo
 Para los elementos que están marcados: [Exportar](#)

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.

```
<div class="collapse navbar-collapse" id="mynavbar">
  <ul class="navbar-nav me-auto">
    <li class="nav-item">
      <a class="nav-link" href="index.php">Inicio</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="actividades.php">Actividades</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="instalaciones.php">Instalaciones</a>
    </li>
  </ul>
  <?php
  if (!isset($_SESSION['logueado'])) {
  >
    <li class="nav-item">
      <a class="nav-link" href="login.php">Login</a>
    </li>
  <?php
  }
  if (isset($_SESSION['logueado'])) {
  >
    <li class="nav-item">
      <a class="nav-link" href="perfil.php">Perfil</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="cerrar.php">Cerrar sesión</a>
    </li>
  <?php
  }
  >
  </ul>
</div>
```

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.



Ilustración 58: Footer

Índex (Home)

Página principal de la aplicación web.

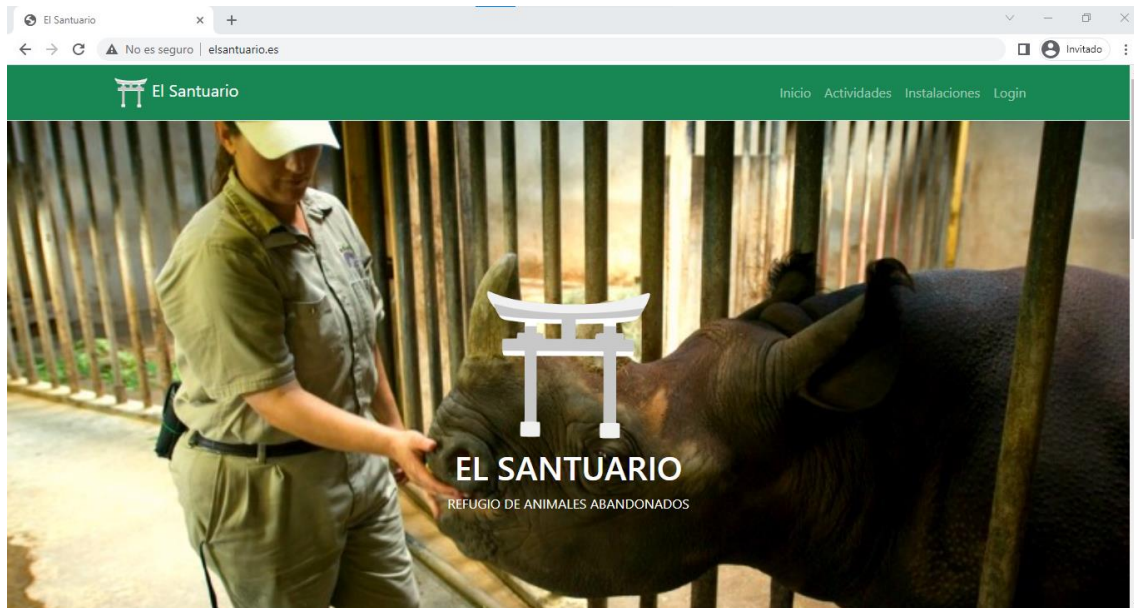


Ilustración 59:Índex

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.



Ilustración 60:Actividades

Instalaciones

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



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.

Bienvenido

DNI

Password

ENTRAR

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.

```
$dniError = "";
if (isset($_POST["entrar"])) {
    $cifrada = hash_hmac('sha512', $_POST["password"], "primeraweb");
    // echo $cifrada;
    $enlace = mysqli_connect("localhost", "comprobar", "comprobar", "elsantuario");
    $usuario = 'SELECT dniUsuario,usuPassword,usuTipo,usuNombre,usuApell,usuTel,usuDir,fechaNac FROM
    usuarios WHERE dniUsuario="' . $_POST["dni"] . '" AND usuPassword="' . $cifrada . '"';
    $dato = mysqli_query($enlace, $usuario);

    $dni = $_POST["dni"];

    function validar_dni($dni)
    {
        $letra = substr($dni, -1);
        $numeros = substr($dni, 0, -1);
        if (substr("TRWAGMYFPDXBNJZSQVHLCKE", $numeros % 23, 1) == $letra && strlen($letra) == 1 && strlen($numeros) == 8) {
            return true;
        } else {
            return false;
        }
    }

    $fila = mysqli_fetch_assoc($dato);
    if (isset($fila)) {
        $_SESSION['r'] = $fila['usuTipo'];
        $_SESSION['dniUsuario'] = $fila['dniUsuario'];
        $_SESSION['usuNombre'] = $fila['usuNombre'];
        $_SESSION['usuApell'] = $fila['usuApell'];
        $_SESSION['usuTel'] = $fila['usuTel'];
        $_SESSION['usuDir'] = $fila['usuDir'];
        $_SESSION['fechaNac'] = $fila['fechaNac'];
    }

    if (validar_dni($dni) == false) {
        $dniError = "El DNI no es válido";
    } elseif (validar_dni($dni) == true) {
        $dniError = "DNI válido";
        if (mysqli_num_rows($dato) == 0) {
            $dniError = "El DNI no existe en la base de datos";
        } else {
            $_SESSION['logueado'] = true;
            header("Location:perfil.php");
        }
    }
}
```

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.

Perfil

Información General

Nombre: Nuria

Apellidos: Rodríguez Álvarez

DNI: 71907370D

Fecha Nacimiento: 2001-07-21

Número de teléfono:

Dirección:

Rol: Gerente

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.

Selecciona una operacion



ALTA ANIMAL
ALTA VOLUNTARIO
AÑADIR ACTIVIDAD
NUEVA ADOPCIÓN
VER ANIMALES ACTUALES
VER TODOS ANIMALES


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

Inserte los datos del nuevo animal

dd/mm/aaaa 



AÑADIR

Ilustración 65: Insertar animal

Para que los datos sean introducidos en la tabla animales antes deben ser pasados por las validaciones. Si todos los datos los 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 de la validación.

```
if (validar_fecha($fecha) == true && validar_nombre($aniNombre) == true && validar_especie($aniEspecie)) {  
    $consulta = "INSERT INTO animales (idAnimal,aniNombre,aniFechaIngreso,aniFechaAdop,aniEspecie,aniSexo,aniAdop)  
VALUES (NULL,'{$_POST["aniNombre"]}','{$_POST["aniFechaIngreso"]} ', NULL,'{$_POST["aniEspecie"]}','{$_POST["aniSexo"]}','0')";  
    $introducir = mysqli_query($enlace, $consulta);  
  
    if (!$introducir) {  
        $introducidos = "Error: No se pudieron insertar los datos";  
    } else {  
        $introducidos = "Datos introducidos con éxito";  
    }  
} else {  
    if (validar_fecha($fecha) == false) {  
        $fechaError = "La fecha no puede ser posterior a hoy";  
        $fecha = "";  
    }  
    if (validar_nombre($aniNombre) == false) {  
        $nombreError = "Solo letras y espacios";  
        $aniNombre = "";  
    }  
    if (validar_especie($aniEspecie) == false) {  
        $espeError = "Solo letras y espacios";  
        $aniEspecie = "";  
    }  
}
```

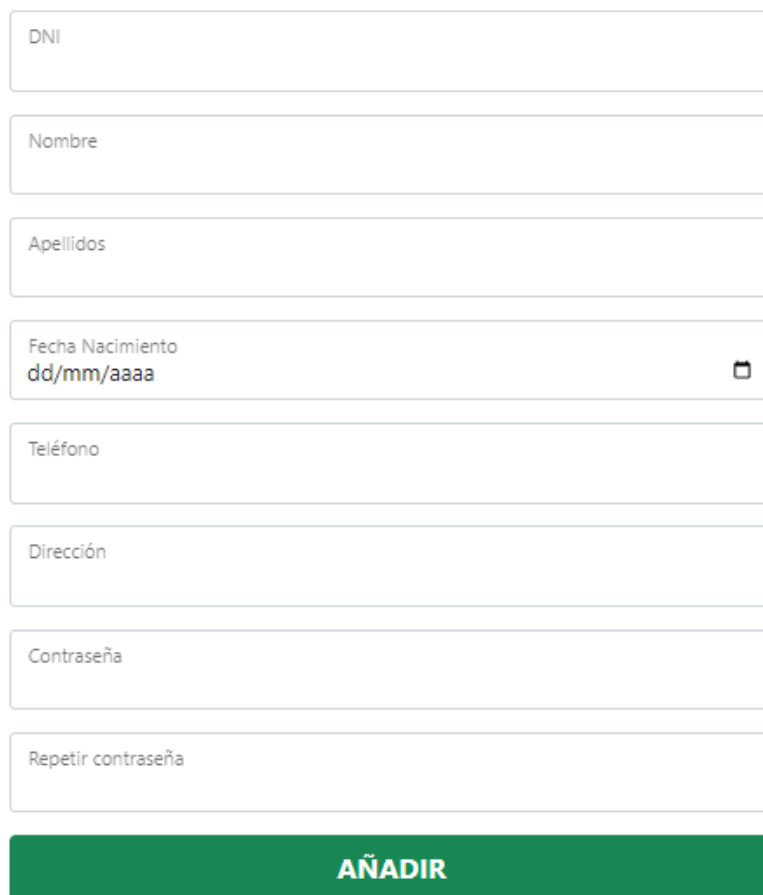
Ilustración 66: Código introducir animal

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

Inserte los datos del nuevo voluntario

El formulario de registro de nuevo voluntario está compuesto por ocho campos de entrada de texto apilados verticalmente. Los campos están etiquetados como 'DNI', 'Nombre', 'Apellidos', 'Fecha Nacimiento' (con un ícono de calendario y el formato 'dd/mm/aaaa'), 'Teléfono', 'Dirección', 'Contraseña' y 'Repetir contraseña'. Debajo de los campos hay un botón rectangular de color verde con el texto 'AÑADIR' en blanco.

DNI
Nombre
Apellidos
Fecha Nacimiento dd/mm/aaaa
Teléfono
Dirección
Contraseña
Repetir contraseña
AÑADIR

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 retornado false.

```

if (validar_tel($telefono) == true && validar_fecha($nacimiento) == true && validar_contra($conta, $repeContra) == true
&& validar_dni($dni) == true && validar_ape($ape) == true && validar_nombre($name) == true) {
    $cifrado = hash_hmac('sha512', $_POST['contrase'], "primeraweb");
    $usuario = "SELECT dniUsuario,usuPassword FROM usuarios WHERE dniUsuario=" . $_POST["dniUsuario"] . " ";
    $dato = mysqli_query($enlace, $usuario);
    if (mysqli_num_rows($dato) == 0) {
        $consulta = "INSERT INTO usuarios (dniUsuario,usuPassword,usuTipo,usuNombre,usuApellido,usuTel,usuDir,fechaNac)
VALUES ('$_POST["dniUsuario"]','$_POST["cifrado"]','Voluntario','$_POST["nombre"]','$_POST["ape"]','$_POST["telefono"]','$_POST["direccion"]','$_POST["fechaNac"]')";
        $introducir = mysqli_query($enlace, $consulta);

        if (!$introducir) {
            $introducidos = "Error: No se pudieron insertar los datos";
        } else {
            $introducidos = "Datos introducidos con éxito";
        }
    } else {
        $dniError = "El DNI ya existe en la base de datos";
    }
} else {
    if (validar_fecha($nacimiento) == false) {
        $nacError = "Fecha incorrecta";
        $nacimiento = "";
    }
    if (validar_dni($dni) == false) {
        $dniError = "El DNI no es válido";
        $dni = "";
    }
    if (validar_ape($ape) == false) {
        $apeError = "Solo letras y espacios ";
        $ape = "";
    }
    if (validar_nombre($name) == false) {
        $nombreError = "Solo letras y espacios ";
        $name = "";
    }
    if (validar_tel($telefono) == false) {
        $telefonoError = "Movil Invalido";
        $telefono = "";
    }
    if (validar_contra($conta, $repeContra) == false) {
        $contraError = " Las contraseñas no son iguales,vuelve a introducirlas";
    }
}
}

```

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.

Asigne la actividad al voluntario

Egea Moya,Berta

Mino(Gato)

Fecha

dd/mm/aaaa

Hora

--:--

Limpieza

AÑADIR

Ilustración 69:Formulario nueva actividad

```

<?php
$volun = 'SELECT dniUsuario,usuNombre,usuApellido FROM usuarios WHERE usuTipo="Voluntario" ORDER BY usuApellido';
$dato = mysqli_query($enlace, $volun);
$nConfig = mysqli_num_rows($dato);
if ($nConfig != 0) {
?>
    <select name="voluntario" class="form-select form-select-lg mb-3" aria-label=".form-select-lg example">
        <?php
        for ($i = 0; $i < $nConfig; $i++) {
            $fila = mysqli_fetch_array($dato);
            ?>
            <option value="<?php echo $fila['dniUsuario']; ?>"><?php echo $fila['usuApellido'] . ", " . $fila['usuNombre']; ?>
            <?php
            ?>
        }
    </select>
<?php
}

```

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.

```

if (validar_fecha($fecha) == true && validar_hora($hora, $fecha) == true) {
    $comprobarHora = 'SELECT horaActividad,fechaActividad FROM actividades WHERE dniVoluntario=" . $_POST["voluntario"] . "
    AND fechaActividad=" . $_POST["fecha"] . " AND horaActividad=" . $_POST["hora"] . " ";
    $dato = mysqli_query($enlace, $comprobarHora);

    if (mysqli_num_rows($dato) == 0) {
        $insertAct = 'INSERT INTO actividades (id,tipoActividad,dniVoluntario,idAnimal,fechaActividad,horaActividad)
        VALUES (NULL,\'$_POST["actividad"]\',\'$_POST["voluntario"]\',\'$_POST["animal"]\',\'$_POST["fecha"]\',\'$_POST["hora"]\')';
        $introducir = mysqli_query($enlace, $insertAct);
        if (!$introducir) {
            $introducidos = "Error: No se pudieron insertar los datos";
        } else {
            $introducidos = "Datos introducidos con éxito";
        }
    } else {
        $introducidos = "Error: El voluntario con dni " . $_POST["voluntario"] . " ya tiene una actividad a la " . $_POST["hora"];
    }
} else {
    if (validar_fecha($fecha) == false) {
        $fechaError = "La fecha no puede ser anterior a hoy";
        $fecha = "";
    }
    if (validar_hora($hora, $fecha) == false) {
        $horaError = "La hora no puede ser anterior a la actual";
        $hora = "";
    }
}
}

```

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.

Inserte los datos de la adopción

DNI

Nombre

Apellidos

Fecha Nacimiento

dd/mm/aaaa

Teléfono

Dirección

Mino(Gato)

AÑADIR

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.

```

if (validar_tel($telefono) == true && validar_fecha($nacimiento) == true && validar_dni($dni) == true && validar_ape($ape) == true && validar_nombre($name) == true) {
    $permitted_chars = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
    $contra = substr(str_shuffle($permitted_chars), 0, 20);
    $cifrado = hash_hmac('sha512', $contra, "primeraweb");
    $usuario = "SELECT dniUsuario,usuPassword FROM usuarios WHERE dniUsuario=" . $_POST["dniUsuario"] . " ";
    $dato = mysqli_query($enlace, $usuario);
    if (mysqli_num_rows($dato) == 0) {
        $fechaAct = date("Y-m-d");
        $consulta = "INSERT INTO usuarios (dniUsuario,usuPassword,usuTipo,usuNombre,usuApellido,usuTel,usuDir,fechaNac)
VALUES ('$_POST["dniUsuario"]','$_POST["cifrado"]','Adoptante','$_POST["nombre"]','$_POST["ape"]','$_POST["telefono"]','$_POST["direccion"]','$_POST["fechaNac"]')";
        $introducir1 = mysqli_query($enlace, $consulta);
        $insertAnimal = "INSERT INTO adopciones(idAdopcion,dniAdoptante,idAnimal,fechaAdop)
VALUES (NULL,'$_POST["dniUsuario"]','$_POST["animal"]','$_POST["fechaAct"]')";
        $introducir2 = mysqli_query($enlace, $insertAnimal);
        $editar = "UPDATE animales SET aniFechaAdop='$_POST["fechaAct"]',aniAdop='1' WHERE idAnimal='$_POST["animal"]' ";
        $editar = mysqli_query($enlace, $editar);

        if ($introducir1 || $introducir2 || $editar) {
            $introducidos = "Error: No se pudieron insertar los datos";
        } else {
            $introducidos = "Datos introducidos con éxito";
            $mensaje = "Guarde la contraseña para el primero inicio de sesión.<br>Podrá cambiarla después de iniciar sesión: $contra";
        }
    } else {
        $dniError = "El DNI ya existe en la base de datos";
    }
} else {
    if (validar_fecha($nacimiento) == false) {
        if (calculaedad($nacimiento) < 18) {
            $nacError = "Debe de ser mayor de 18 para poder adoptar";
            $nacimiento = "";
        }
    }
}

```

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).

Animales actualmente en EL SANTUARIO

Nombre	Especie	Sexo	Fecha Ingreso	Adoptado	Fecha Adopción
Mino	Gato	Macho	2022-02-08	NO	-
Pepe	Gato	Macho	2022-05-13	NO	-
Susu	Perro	Macho	2021-04-30	NO	-
Nea	Perro	Hembra	2022-02-10	NO	-
Kadie	Gato	Hembra	2022-05-27	NO	-
Nea	Gato	Hembra	2022-01-03	NO	-
Coco	Perro	Macho	2022-05-13	NO	-
Hades	Perro	Macho	2020-08-22	NO	-
Dory	Gato	Hembra	2021-08-26	NO	-

Ilustración 74: Tabla animales actuales

```

$animales = 'SELECT * FROM animales WHERE aniAdop=0';
$dato = mysqli_query($enlace, $animales);
$nConfig = mysqli_num_rows($dato);

if ($nConfig != 0) {
    <div class="container mt-5">
        <h3 class="login-heading mb-4">Animales actualmente en EL SANTUARIO</h3>
        <table class="table">
            <thead class="table-success">
                <td>Nombre</td>
                <td>Especie</td>
                <td>Sexo</td>
                <td>Fecha Ingreso</td>
                <td>Adoptado</td>
                <td>Fecha Adopción</td>
            </thead>
            <tbody>
                <tr>
                    <td><?php echo $fila['aniNombre'] ?></td>
                    <td><?php echo $fila['aniEspecie'] ?></td>
                    <td><?php echo $fila['aniSexo'] ?></td>
                    <td><?php echo $fila['aniFechaIngreso'] ?></td>
                    <td><?php echo "NO" ?></td>
                    <td><?php echo "-" ?></td>
                </tr>
            </tbody>
        </table>
    </div>
    <br><br>
}

```

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.

Todos los animales EL SANTUARIO

Nombre	Especie	Sexo	Fecha ingreso	Adoptado	Fecha Adopción
Mino	Gato	Macho	2022-02-08	NO	-
Laski	Perro	Macho	2020-02-20	SI	2021-04-07
Desy	Perro	Hembra	2021-09-10	SI	2021-10-07
Riota	Gato	Hembra	2022-04-03	SI	2022-04-27
Chester	Serpiente	Macho	2022-03-14	SI	2022-05-24
Pepe	Gato	Macho	2022-05-13	NO	-
Nieve	Gato	Hembra	2022-05-05	SI	2022-05-11
Leo	Perro	Macho	2014-10-16	SI	2015-10-06
Susu	Perro	Macho	2021-04-30	NO	-
Nea	Perro	Hembra	2022-02-10	NO	-
Kadie	Gato	Hembra	2022-05-27	NO	-
Nea	Gato	Hembra	2022-01-03	NO	-
Coco	Perro	Macho	2022-05-13	NO	-
Hades	Perro	Macho	2020-08-22	NO	-
Dory	Gato	Hembra	2021-08-26	NO	-

Ilustración 76: Tabla ver todos los animales

```

$animales = 'SELECT * FROM animales';
$dato = mysqli_query($enlace, $animales);
$nConfig = mysqli_num_rows($dato);

if ($nConfig != 0) {

?>
    <div class="container mt-5">
        <h3 class="login-heading mb-4">Todos los animales EL SANTUARIO</h3>
        <table class="table">
            <thead class="table-success">
                <td>Nombre</td>
                <td>Especie</td>
                <td>Sexo</td>
                <td>Fecha Ingreso</td>
                <td>Adoptado</td>
                <td>Fecha Adopción</td>
            </thead>
            <?php
            for ($i = 0; $i < $nConfig; $i++) {
                $fila = mysqli_fetch_array($dato);

                ?>
                <tr>
                    <td><?php echo $fila['aniNombre'] ?></td>
                    <td><?php echo $fila['aniEspecie'] ?></td>
                    <td><?php echo $fila['aniSexo'] ?></td>
                    <td><?php echo $fila['aniFechaIngreso'] ?></td>
                    <?php
                    if ($fila['aniAdop'] === "0") { ?>
                        <td><?php echo "NO" ?></td>
                        <td><?php echo "-" ?></td>
                    <?php
                    } else {
                        ?>
                        <td><?php echo "SI" ?></td>
                        <td><?php echo $fila['aniFechaAdop'] ?></td>
                    <?php
                    }
                </tr>
            <?php
            }
        </table>
    </div>
}

```

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.

Selecciona una operacion

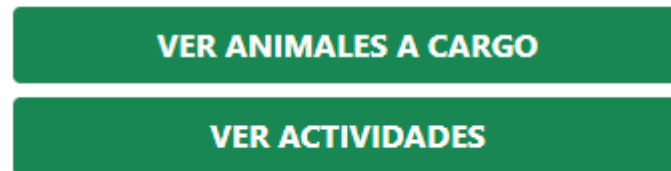


Ilustración 78: Perfil voluntario

Ver animales a cargo

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

Animales a cargo

Nombre	Especie	Sexo	Fecha Ingreso
Pepe	Gato	Macho	2022-05-13
Leo	Perro	Macho	2014-10-16

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.

```
$animales = 'SELECT aniNombre, aniEspecie, aniSexo, aniFechaIngreso FROM animales, actividades
WHERE animales.idAnimal=actividades.idAnimal AND actividades.dniVoluntario=" . $_SESSION['dniUsuario'] . "'';
$dato = mysqli_query($enlace, $animales);
$nConfig = mysqli_num_rows($dato);

if ($nConfig != 0) {
    <div class="container mb-5" >
    <h3 class="login-heading mt-5">Animales a cargo</h3>
    <table class="table mb-4">
        <thead class="table-success">
            <tr>
                <td>Nombre</td>
                <td>Especie</td>
                <td>Sexo</td>
                <td>Fecha Ingreso</td>
            </tr>
        </thead>
        <tbody>
            <?php
            for ($i = 0; $i < $nConfig; $i++) {
                $fila = mysqli_fetch_array($dato);
                <tr>
                    <td>?php echo $fila['aniNombre'] ?></td>
                    <td>?php echo $fila['aniEspecie'] ?></td>
                    <td>?php echo $fila['aniSexo'] ?></td>
                    <td>?php echo $fila['aniFechaIngreso'] ?></td>
                </tr>
            </tbody>
        </table>
    }
```

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.

Actividades

Actividad	Animal	Fecha Actividad	Hora Actividad	Modificar Fecha/Hora
Lavado	Leo	2022-05-27	13:45:00	<button>CAMBIAR</button>
Paseo	Pepe	2022-05-27	19:45:00	<button>CAMBIAR</button>

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.

```
$select = 'SELECT id,nombreActividad,aniNombre,fechaActividad,horaActividad FROM tipoactividades,actividades,animales
WHERE actividades.tipoActividad= tipoactividades.idActividad AND actividades.dniVoluntario="' . $_SESSION["dniUsuario"] . '" AND animales.idAnimal=actividades.idAnimal';
$dato = mysqli_query($enlace, $select);
$nConfig = mysqli_num_rows($dato);

if ($nConfig != 0) {
    <div class="container mt-5">
    <h3 class="login-heading mb-4">Actividades</h3>
    <table class="table">
        <thead class="table-success">
            <tr>
                <th>Actividad</th>
                <th>Animal</th>
                <th>Fecha Actividad</th>
                <th>Hora Actividad</th>
                <th>Modificar Fecha/Hora</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td><?php echo $fila['nombreActividad'] ?></td>
                <td><?php echo $fila['aniNombre'] ?></td>
                <td><?php echo $fila['fechaActividad'] ?></td>
                <td><?php echo $fila['horaActividad'] ?></td>
                <td>
                    <form action="" method="POST">
                        <button type="submit" name="cambio" value="<?php echo implode(",", $fila); ?>"
                            class="btn btn-lg btn-success btn-login text-uppercase fw-bold mb-2">Cambiar</button></form>
                </td>
            </tr>
        </tbody>
    </table>
    </div>
}
```

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)

```

if (validar_fecha($fecha) == true && validar_hora($hora, $fecha) == true) {
    $comprobarHora = 'SELECT horaActividad, fechaActividad FROM actividades
    WHERE dniVoluntario="' . $_SESSION["dniUsuario"] . '" AND fechaActividad="' . $fecha . '" AND horaActividad="' . $hora . '"';
    $dato = mysqli_query($enlace, $comprobarHora);

    if (mysqli_num_rows($dato) == 0) {
        $edit = "UPDATE actividades SET fechaActividad='$fecha', horaActividad='$hora' WHERE tipoActividad=' $datos[0] ' ";
        $introducir = mysqli_query($enlace, $edit);
        if (!$introducir) {
            $introducidos = "Error: No se pudieron insertar los datos";
        } else {
            $introducidos = "Datos introducidos con exito";
            header("Location:voluntario.php");
        }
    } else {
        $introducidos = "Error: El voluntario con dni " . $_SESSION["dniUsuario"] . " ya tiene una actividad a la " . $hora;
    }
} else {
    if (validar_fecha($fecha) == false) {
        $fechaError = "La fecha no puede ser anterior a hoy";
        $fecha = "";
    }
    if (validar_hora($hora, $fecha) == false) {
        $horaError = "La hora no puede ser anterior a la actual";
        $hora = "";
    }
}
}

```

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.

Selecciona una operacion

VER ANIMALES ADOPTADOS

CAMBIAR CONTRASEÑA

Ilustración 84:Perfil adoptante

Ver animales adoptados

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

Adopciones

Nombre	Especie	Sexo	Fecha Adopción
Susu	Perro	Macho	2015-10-06

Ilustración 85:Tabla ver animales adopciones

```

$animales = 'SELECT aniNombre,aniEspecie,aniSexo,fechaAdop FROM animales,adopciones WHERE adopciones.dniAdoptante="" . $_SESSION["dniUsuario"] . ""
AND adopciones.idAnimal=animales.idAnimal ';
$dato = mysqli_query($enlace, $animales);
$nConfig = mysqli_num_rows($dato);
if ($nConfig != 0) {
?>
<div class="container mt-5">
<h4>Adopciones</h4>
<table class="table">
<thead class="table-success">
<tr>
<td>Nombre</td>
<td>Especie</td>
<td>Sexo</td>
<td>Fecha Adopción</td>
</tr>
</thead>
<tbody>
<?php
for ($i = 0; $i < $nConfig; $i++) {
$fila = mysqli_fetch_array($dato);
?>
<tr>
<td><?php echo $fila["aniNombre"] ?></td>
<td><?php echo $fila["aniEspecie"] ?></td>
<td><?php echo $fila["aniSexo"] ?></td>
<td><?php echo $fila["fechaAdop"] ?></td>
</tr>
<?php
}
?>
</tbody>
</table>

```

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.

Cambio de contraseña

AÑADIR

Ilustración 87: Formulario cambio de contraseña

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

```

$contra = $_POST["contrase"];
function validar_contra($contra, $repeContra)
{
    if ($contra != $repeContra) {
        return false;
    } else {
        return true;
    }
}
if (validar_contra($contra, $repeContra) == true) {
    $cifrada = hash_hmac('sha512', $_POST["contrase"], "primeraweb");
    $edit = "UPDATE usuarios SET usuPassword='$cifrada' WHERE dniUsuario=" . $_SESSION["dniUsuario"] . " ";
    $introducir = mysqli_query($enlace, $edit);
    if (!$introducir) {
        $introducidos = "Error: No se pudieron insertar los datos";
    } else {
        $introducidos = "Datos introducidos con éxito";
    }
} else {
    if (validar_contra($contra, $repeContra) == false) {
        $contraError = "Las contraseñas no son iguales, vuelve a introducirlas";
    }
}
}

```

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 session_destroy y nos redirigirá a la pantalla de home.

```
<?php
    session_start();
    session_destroy();
    header('location: ./index.php');
?>
```

Ilustración 89: Código cerrar sesión

Índice de figuras

Ilustración 1:Diagrama E/R	5
Ilustración 2:Tabla usuarios	5
Ilustración 3:Tabla animales	6
Ilustración 4:Tabla adopciones.....	6
Ilustración 5:Índices Tabla adopciones.....	6
Ilustración 6:Tabla actividades	7
Ilustración 7:Índices Tabla actividades	7
Ilustración 8:Tabla tipoActividades	7
Ilustración 9:Privilegios cuenta gerente	8
Ilustración 10:Privilegios cuenta voluntario	8
Ilustración 11:Privilegios cuenta adoptante.....	9
Ilustración 12:Permisos adoptante en la tabla adopciones	9
Ilustración 13:Permisos adoptante en la tabla animales.....	9
Ilustración 14:Permisos adoptante en la tabla usuarios.....	10
Ilustración 15:Privilegios cuenta comprobar	10
Ilustración 16:Conexión a la base de datos como comprobar	11
Ilustración 17:Conexión a la base de datos como gerente	11
Ilustración 18:Conexión a la base de datos como voluntario	11
Ilustración 19:Conexión a la base de datos como adoptante.....	11
Ilustración 20:Ejemplo mysqli_query de consulta SELECT.....	12
Ilustración 21:Ejemplo mysqli_query de consulta UPDATE	12
Ilustración 22:Ejemplo mysqli_query de consulta INSERT.....	12
Ilustración 23:Ejemplo mysqli_num_rows	13
Ilustración 24:Ejemplo utilización mysqli_fetch_array	13
Ilustración 25:Función hash_hmac.....	14
Ilustración 26:Función isset	15
Ilustración 27:Función header	15
Ilustración 28:Función implode.....	16
Ilustración 29:Función explode.....	16
Ilustración 30:Funciones str_shuffle y substr	17
Ilustración 31:Archivo functions.php.....	17
Ilustración 32:Función get_header	17
Ilustración 33:Función get_footer.....	18
Ilustración 34:Función validar_tel	18
Ilustración 35:Función validar_dni	18
Ilustración 36:Funciones validar_nombre y validar_ape	19
Ilustración 37:Función calculaedad	19
Ilustración 38:Función validar_hora.....	20
Ilustración 39:Función validar_fecha actividades.....	20
Ilustración 40:Función validar_fecha voluntario.....	20
Ilustración 41:Función validar_fecha adoptante	21
Ilustración 42:Página descarga XAMPP	22
Ilustración 43:Instalación XAMPP 1	22
Ilustración 44:Instalación XAMPP 2	23
Ilustración 45:Instalación XAMPP 3	23
Ilustración 46:Instalación XAMPP 4	24

Ilustración 47:Instalación XAMPP 5	24
Ilustración 48:Configuración archivo hosts.....	25
Ilustración 49:Configuración archivo httpd-vhost.....	25
Ilustración 50:Índex phpMyAdmin.....	26
Ilustración 51:Crear base de datos.....	26
Ilustración 52:Importación base de datos	27
Ilustración 53:Creación de usuarios en la base de datos	27
Ilustración 54:Cuentas de usuarios	27
Ilustración 55:Header sin iniciar sesión	28
Ilustración 56:Header con sesión iniciada	28
Ilustración 57:Código header.....	28
Ilustración 58:Footer	28
Ilustración 59:Índex	29
Ilustración 60:Actividades	29
Ilustración 61:Login	30
Ilustración 62:Código login.....	31
Ilustración 63:Información general.....	32
Ilustración 64:Menú gerente	32
Ilustración 65:Insertar animal.....	33
Ilustración 66:Código introducir animal	33
Ilustración 67:Formulario registro nuevo voluntario	34
Ilustración 68:Código formulario nuevo voluntario	35
Ilustración 69:Formulario nueva actividad	35
Ilustración 70:Ejemplo de <select>.....	36
Ilustración 71:Código asignar actividad	36
Ilustración 72:Formulario nueva adopción.....	37
Ilustración 73:Código nueva adopción	37
Ilustración 74:Tabla animales actuales	38
Ilustración 75:Código ver animales actuales.....	38
Ilustración 76:Tabla ver todos los animales	39
Ilustración 77:Código ver todos los animales	39
Ilustración 78:Perfil voluntario	40
Ilustración 79:Tabla animales a cargo	40
Ilustración 80:Código ver animales a cargo	40
Ilustración 81:Tabla ver actividades voluntario	41
Ilustración 82:Código ver actividades voluntario	41
Ilustración 83:Código modificar fecha/hora	42
Ilustración 84:Perfil adoptante	42
Ilustración 85:Tabla ver animales adopciones.....	42
Ilustración 86:Código ver animales adoptados	43
Ilustración 87:Formulario cambio de contraseña.....	43
Ilustración 88:Código formulario cambiar contraseña	43
Ilustración 89:Código cerrar sesión	44

Bibliografía

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