

Implantación de foro en AWS

Proyecto de fin de grado de Administración de Sistemas Informáticos en Red

Andrea Osma Rafael

CFP Juan XXIII



Ciclo 2020/2022 CFGS Administración
de Sistemas Informáticos en Red

Índice

Índice	1
Introducción.....	3
Programas e infraestructura utilizados.....	4
XAMPP.....	4
GitHub.....	6
Ubuntu 22.04.....	6
Amazon Web Services.....	6
Foro.....	7
Base de datos.....	7
Relación de tablas	7
Tabla de usuarios	7
Tabla de temas	8
Tabla de posts	9
Añadiendo las foreign keys.....	9
Código PHP.....	11
Cabecera y pie de página.....	11
Conexión con la base de datos	13
Página de registro	14
Inicio de sesión	23
Cierre de sesión.....	27
Crear una categoría	28
Crear un tema.....	31
Inicio.....	37
Vista de categoría.....	40
Vista de tema	44
Responder a temas	47
Creación del servidor web y la base de datos en AWS	50
Despliegue de la instancia de EC2	50
Despliegue del servidor de bases de datos.....	56
Instalación del servidor web Apache	63
Añadiendo contenido a la base de datos	68
Prueba de funcionamiento	71
Activación de HTTPS.....	72

Conclusión del proyecto	73
Glosario	74
Licencia	76

Introducción

En este proyecto he querido mostrar los beneficios que el almacenamiento en la nube puede tener, tanto para una persona individual como para una empresa, como espacio para desplegar nuestros proyectos. Lo he querido mostrar mediante un foro que he programado mediante PHP y mySQL, debido a mi interés en la programación de páginas web.

El proyecto consistirá en un foro en el que los miembros podrán compartir sus pensamientos con el resto de los usuarios, por lo tanto podrán:

- Crear su propia cuenta.
- Iniciar o cerrar sesión.
- Ver el contenido, pudiendo desplegar cada una de las categorías y temas para ver el contenido ya publicado.
- Crear sus propias categorías (solamente si son administradores).
- Crear sus propios temas.
- Responder a los temas que ya haya publicados.

A continuación, usaré AWS (Amazon Web Services) para desplegar esa página web, y que cualquier persona con su IP o DNS pueda acceder sin problemas. Le daré seguridad mediante el protocolo HTTPS y mostraré opciones como **Route 53** para darle un nombre de dominio a la página web.

La memoria de este proyecto también se encuentra en GitHub:

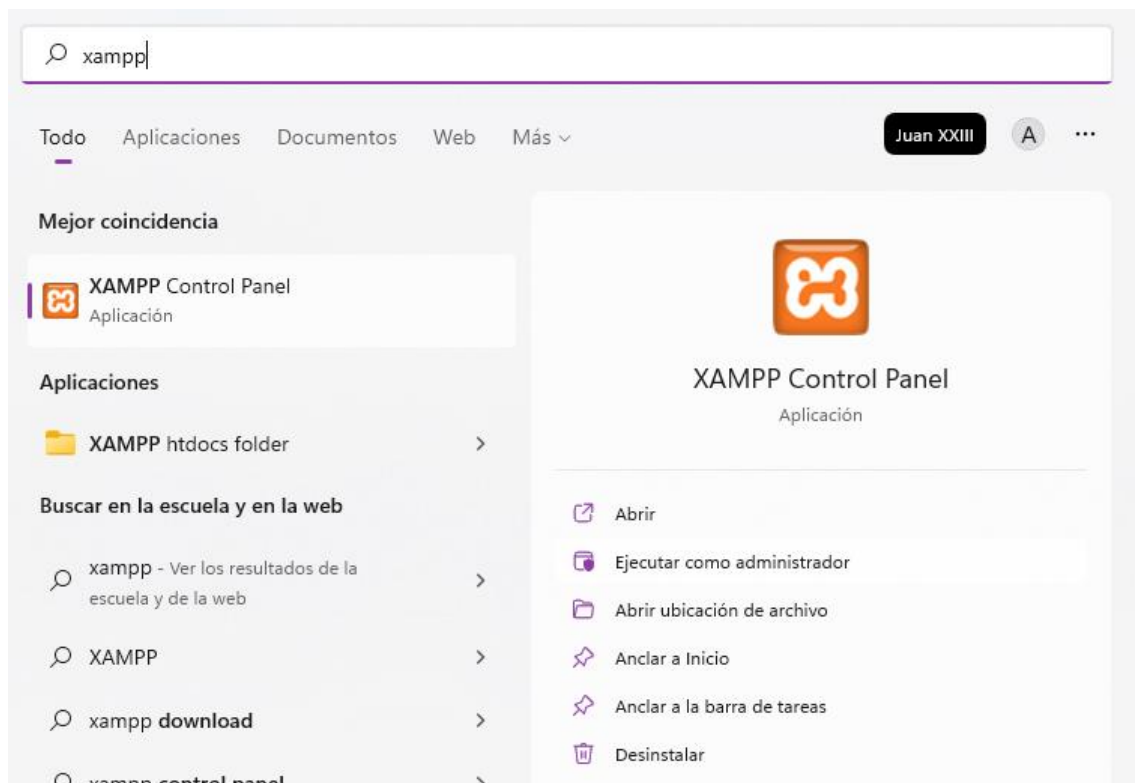
<https://github.com/AndreaOsma/ProyectoASIR/>

Programas e infraestructura utilizados

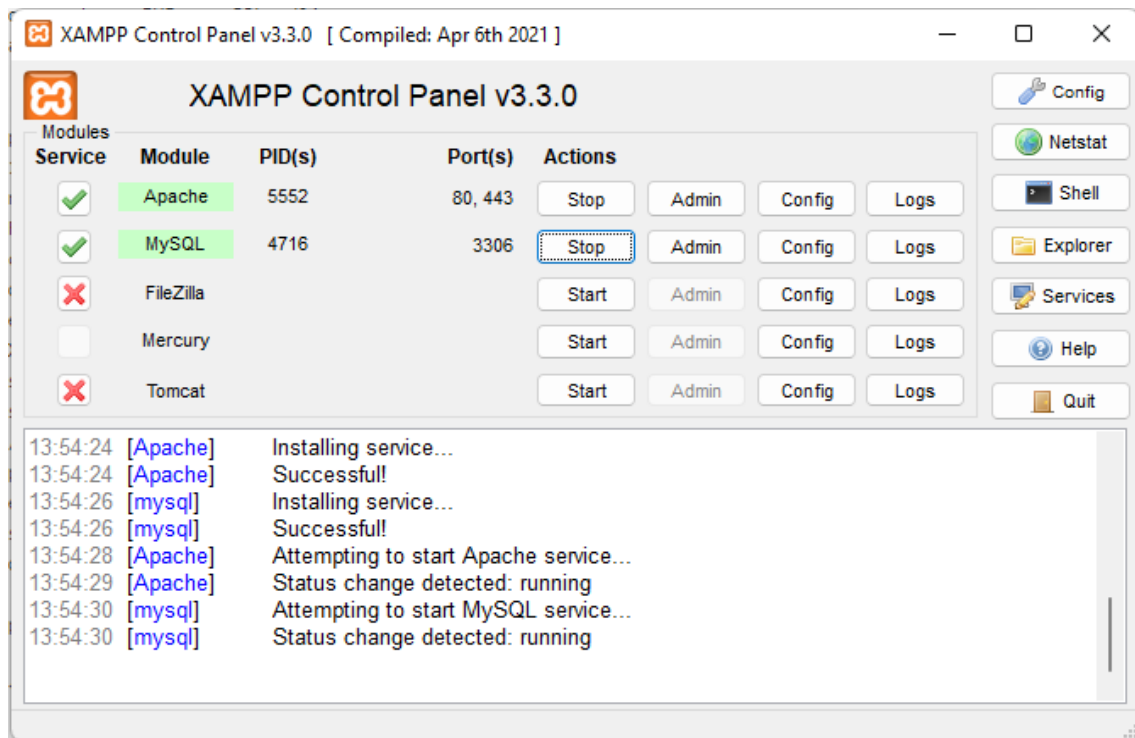
XAMPP

XAMPP es un entorno de desarrollo web. La distribución incluye el servidor web Apache, un servidor de bases de datos mySQL, un servidor FTP Filezilla, el servidor de correo Mercury y el contenedor Tomcat para utilizar el lenguaje de programación Java. Puesto que este proyecto estará programado en PHP, JavaScript y mySQL, solo necesitaremos los dos primeros (Apache y mySQL). XAMPP funciona de manera que crea estos servidores en la máquina donde se instalen, pudiendo luego acceder a ellos o bien desde la IP del dispositivo y el puerto del servidor (por ejemplo: <http://192.168.1.60:80> para acceder al servidor Apache, es decir, a la web principal) o bien accediendo desde la propia máquina poniendo <http://localhost> en el navegador. Este programa solamente lo usaré como entorno de pruebas mientras programo el foro, ya que después en AWS instalaré tanto Apache como mySQL, pero no el resto de servidores.

Para que estos servidores funcionen, simplemente una vez instalado XAMPP, que se puede descargar desde [aquí](#), entraremos a su panel de control como administrador.



Posteriormente, le daremos a "Start" a los servicios que queramos utilizar. Como se puede ver en la imagen que hay a continuación, aparecerán el PID (código de proceso) del servicio activado y el puerto o puertos que tiene asignados dicho servicio. Los que no utilizamos simplemente se quedarán desactivados, para no consumir recursos de manera innecesaria.



También hay varios botones de acción en cada servicio. El botón "admin" es un enlace directo a la ubicación del servidor, el botón "config" permite seleccionar entre los diferentes archivos de configuración del servidor para poder editarlos en caso de que fuese necesario, y el botón "logs" lleva a los registros del servidor, como por ejemplo para ver los accesos o para ver los errores.



Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft, que tiene aplicaciones para Windows, Linux, macOS y web. Tiene control integrado de Git y permite la instalación de extensiones para depurar el código de cualquier lenguaje y encontrar errores. He decidido utilizar este programa ya que, primero, ayuda con la depuración de errores y la sintaxis, y segundo, permite iniciar sesión con la cuenta de Microsoft o la de Github para poder tener una copia de seguridad del código o en el caso de Github poder clonar repositorios con facilidad.

GitHub

GitHub es una web para alojar proyectos utilizando el sistema de control de versiones Git, que es un software que diseñó Linus Torvalds, la misma persona que creó Linux, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código. GitHub se utiliza principalmente para la creación de código fuente de programas de ordenador, y lo he elegido para poder alojar, además del contenido de mi memoria, que es este archivo **README.md**, el código fuente de mi foro, para luego poder por ejemplo clonarlo en la instancia donde alojaré el servidor web.

Ubuntu 22.04

Ubuntu es una distribución Linux basada en Debian GNU/Linux, que incluye principalmente software libre y de código abierto. Lo utilizaré mediante el subsistema WSL de Windows, que permite implantar un sistema Linux en una línea de código de PowerShell, como si de una máquina virtual se tratara, y usarlo mediante la interfaz de Terminal. Utilizaré Ubuntu para acceder en remoto a la instancia creada en AWS.

Amazon Web Services

Amazon Web Services, abreviado como AWS, es una colección de servicios web que en conjunto forman una plataforma de computación en la nube. La mayoría de estos servicios no están expuestos directamente a los usuarios finales, sino que ofrecen una funcionalidad que otros desarrolladores puedan utilizar en sus aplicaciones. Se accede a través de HTTP, pudiendo realizar todas las gestiones a través de una consola web, pero una vez ya realizado el registro también se puede acceder mediante línea de comandos instalando el paquete que lo permite. AWS está situado en 18 regiones geográficas, y solamente en Europa hay servidores en cinco diferentes ciudades, cada una con tres zonas diferentes de disponibilidad. Las zonas de seguridad son los centros de datos que proporcionan sus servicios, y están aisladas unas de otras para evitar la propagación de cortes entre las zonas. Esto nos da la seguridad de que es prácticamente imposible que nuestros servicios dejen de estar disponibles en algún periodo de tiempo, ya que sería muy complicado que todas las zonas de disponibilidad cayesen a la vez, y si cayese solamente una no importaría ya que nuestros datos tienen una réplica en otra. Hay otros servicios como Azure, de Microsoft, o Gcloud, de Google, siendo estos son competidores más directos, pero yo he elegido AWS debido a que es donde tengo cierta experiencia.

Foro

Base de datos

El objetivo de esta sección es realizar una página web, un foro creado en PHP con una base de datos mySQL para que los usuarios puedan compartir sus pensamientos, ideas o quejas. Está enfocado a una red de empresa, por lo que alguien que no sea usuario no podría ver el contenido, pero se podría adaptar a ser un foro normal quitando solamente un par de líneas de código. Podríamos utilizar otros servicios como Slack o Discord, donde crear un foro con diferentes subforos con esta misma misión, pero haciendo nuestro propio foro nos aseguramos de tener una web propia, sin necesidad de depender de los servidores de terceros.

Relación de tablas

Para crear el modelo de datos he utilizado Case Studio. Los datos se gestionarán así:

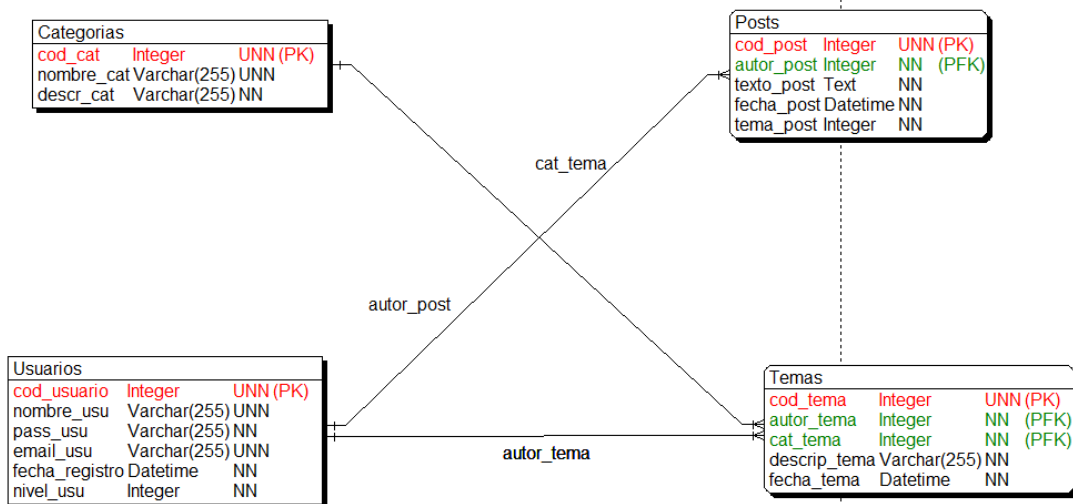


Tabla de usuarios

En primer lugar, es necesario crear una tabla dentro de la base de datos que recoja los usuarios que habrá en el foro. La primera columna será `cod_usuario`, que será el id del usuario, de tipo integer, que no podrá estar vacía y que se irá incrementando cada vez que se meta una nueva fila. Llevará también las columnas `nombre_usu` (para el nombre de usuario), `pass_usu` (para la contraseña), `email_usu` (para el correo electrónico), `fecha_registro` (que contendrá la fecha en la que se hizo el registro), y `nivel_usu`, ya que el usuario podrá ser administrador o estándar, por lo que si tiene un 1 será administrador y si tiene un 0 será estándar. La primary key será `cod_usuario` y `nombre_usu` debe ser único.


```
CREATE TABLE usuarios (  
cod_usuario    INT(8) NOT NULL AUTO_INCREMENT,  
nombre_usu     VARCHAR(30) NOT NULL,  
pass_usu       VARCHAR(255) NOT NULL,  
email_usu      VARCHAR(255) NOT NULL,  
fecha_registro DATETIME NOT NULL,  
nivel_usu      INT(8) NOT NULL,  
UNIQUE INDEX nombre_usuario_unico (nombre_usu),  
PRIMARY KEY (cod_usuario)  
);
```

Tabla de categorías

La segunda tabla será la de categorías, que será donde vayan englobados todos los temas. La primera columna es cod_cat, estructurada de la misma manera que en la tabla de usuarios, la segunda nombre_cat, donde irá el nombre de la categoría, y la tercera descr_cat, donde irá la descripción de la categoría. La columna nombre_cat será única y la primary key será cod_cat.

```
CREATE TABLE categorias (  
cod_cat          INT(8) NOT NULL AUTO_INCREMENT,  
nombre_cat       VARCHAR(255) NOT NULL,  
descr_cat        VARCHAR(255) NOT NULL,  
UNIQUE INDEX nombre_cat_unico (nombre_cat),  
PRIMARY KEY (cod_cat)  
);
```

Tabla de temas

La tercera tabla será la de temas. La primera columna es cod_tema, estructurada de la misma manera que en el resto de las tablas, la segunda descrip_tema, donde irá la descripción o asunto del tema, la tercera fecha_tema, que indicará la fecha en la que fue insertado el tema, la cuarta cat_tema, que será una foreign key donde aparezca el código de la categoría a la que pertenece el tema, y la quinta será autor_tema, donde irá una foreign key hacia el código del usuario que ha creado el tema. La primary key será cod_tema.

```
CREATE TABLE temas (
cod_tema          INT(8) NOT NULL AUTO_INCREMENT,
descrip_tema      VARCHAR(255) NOT NULL,
fecha_tema        DATETIME NOT NULL,
cat_tema          INT(8) NOT NULL,
autor_tema        INT(8) NOT NULL,
PRIMARY KEY (cod_tema)
);
```

Tabla de posts

La cuarta tabla será la de posts, es decir, la de las respuestas a los temas. La primera columna es cod_post, estructurada de la misma manera que en el resto de las tablas, la segunda texto_post, donde irá el texto que el usuario meta en él, la tercera fecha_post, que indicará la fecha en la que fue insertado el post, la cuarta tema_post, que será una foreign key donde aparezca el código del tema al que responde el post, y la quinta será autor_post, donde irá una foreign key hacia el código del usuario que ha creado el post. La primary key será cod_post.

```
CREATE TABLE posts (
cod_post          INT(8) NOT NULL AUTO_INCREMENT,
texto_post        TEXT NOT NULL,
fecha_post        DATETIME NOT NULL,
tema_post         INT(8) NOT NULL,
autor_post        INT(8) NOT NULL,
PRIMARY KEY (cod_post)
);
```

Añadiendo las foreign keys

Por último, ya que no lo hemos hecho en el código insertado anteriormente, insertaremos tres comandos alter table para crear las foreign keys en cada una de las tablas, con un "on delete on restrict update cascade", que no nos dejará borrar una categoría si hay un tema, un usuario si tiene temas o un tema si tiene posts.

```
ALTER TABLE temas ADD FOREIGN KEY(cat_tema)
REFERENCES categorias(cod_cat) ON DELETE CASCADE ON
UPDATE CASCADE;
```

```
ALTER TABLE temas ADD FOREIGN KEY(autor_tema)  
REFERENCES usuarios(cod_usuario) ON DELETE RESTRICT  
ON UPDATE CASCADE;
```

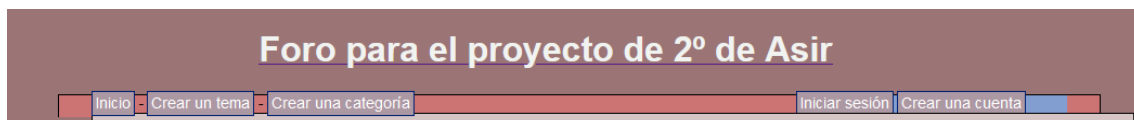
```
ALTER TABLE posts ADD FOREIGN KEY(tema_post)  
REFERENCES temas(cod_tema) ON DELETE RESTRICT ON  
UPDATE CASCADE;
```

Código PHP

Cabecera y pie de página

En primer lugar, crearemos un archivo **header.php**. Este archivo contendrá el menú, y estará incluido al principio del código PHP de cada archivo con la línea:

```
include "header.php";
```



El código fuente es el siguiente:

```
<?php
    session_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="nl" lang="nl">
<head>
    <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
    <title>Foro</title>
    <link rel="stylesheet" href="css/estilo.css"
type="text/css">
    <link rel="icon" type="image/x-icon"
href="img/favicon.ico">
</head>
<body>
<a href="index.php"><h1>Foro para el proyecto de 2º
de ASIR</h1></a>
    <div id="wrapper">
```

```

        <div id="menu">
            <a class="item" href="index.php">Inicio</a>
-
            <a class="item" href="crear_tema.php">Crear
un tema</a> -
            <a class="item" href="crear_cat.php">Crear
una categoría</a>

            <div id="userbar">
<div id="userbar">
    <?php
$sesion_iniciada=$_SESSION['sesion_iniciada']    ??
null;

    if($sesion_iniciada)
    {
        echo 'Hola, ' . $_SESSION['nombre_usu'] .
'. ¿No eres tú? <a class="item"
href="cierresesion.php">Cerrar sesión</a>';
    }
    else
    {
        echo '<a class="item"
href="login.php">Iniciar sesión</a> <a class="item"
href="registro.php">Crear una cuenta</a>';
    }
?>
</div>
    </div>
        <div id="contenido">

```

Crearemos también un archivo **footer.php** que en este caso contendrá mi nombre, pero podría contener cualquier otra información como el copyright o enlaces de interés, e irá incluido al final del código PHP de cada archivo con la línea:

```
include footer.php;
```

El código fuente es el siguiente:

```
</div>
</div>
<div id="footer">Andrea Osma Rafael, 2º ASIR</div>
</body>
</html>
```

Como podemos ver en el código, el **header.php** parece incompleto, ya que no tiene una etiqueta `</body>` que finalice el código, y tiene un `<div>` abierto que tampoco está finalizado. Este código se finaliza en el **footer.php**, ya que al dedicarnos a incluirlos en cada archivo, estos actuarán como si el código estuviese escrito en cada archivo.

Conexión con la base de datos

Para conectar con la base de datos y poder realizar cualquier acción dentro de ella, necesitaremos unas líneas de código dedicadas a ello. Para no escribirlas en cada archivo, crearemos un archivo con ellas que será el que después incluyamos en cada archivo. Este archivo contendrá una variable llamada `$conn` (aunque la podremos llamar como queramos) que dentro contendrá la función `mysqli_connect`, donde dentro irán en orden la ubicación de la base de datos (en nuestro caso `localhost` o `127.0.0.1`), el usuario con el que queramos acceder a la base de datos, su contraseña y el nombre de la base de datos a la que queramos acceder. Después crearemos una condición `if` que declara que si no se puede lograr la conexión por un error en la base de datos, el archivo mostrará un texto de error y el código de error.

El código fuente es el siguiente:

```
<?php
$conn = mysqli_connect("localhost", "root", "",
"foro");

if (!$conn) {
    die("ERROR: no se ha podido establecer una
conexión con la base de datos. " .
mysqli_connect_errno());
}
```



Página de registro

Este foro será privado, por lo que para ver o crear contenido será necesario registrarse mediante la página de registro, que contendrá un formulario. El formulario tendrá cuatro campos, que serán el nombre de usuario, la contraseña, una repetición de la contraseña y el email del usuario.

El código funcionará de manera que, primero, tendrá incluidos los archivos **connect.php** y **header.php**, el primero para mostrar la base de datos y el segundo para mostrar el menú de navegación. Si no hay ya ningunos datos guardados con el método post, mostrará el formulario, que el usuario rellenará para poder iniciar sesión. Si ya hay datos, comprobará que estos estén bien rellenados:

- Si el usuario contiene algún símbolo, mostrará una alerta en JavaScript de que no puede añadirlo ya que el nombre de usuario solo puede contener letras o números, y añadirá el mismo texto de error al array \$errors, que contendrá todos los errores para así mostrar todos los errores en una lista.

[Inicio](#) - [Crear un tema](#) - [Crear una categoría](#)

Registro

Algunos campos son incorrectos o están vacíos.

- El nombre de usuario solo puede contener números y letras.

[Volver al registro](#)

- Si el nombre de usuario contiene más de 30 caracteres, mostrará otra alerta y añadirá ese error al array.

[Inicio](#) - [Crear un tema](#) - [Crear una categoría](#)

Registro

Nombre de usuario:

Contraseña:

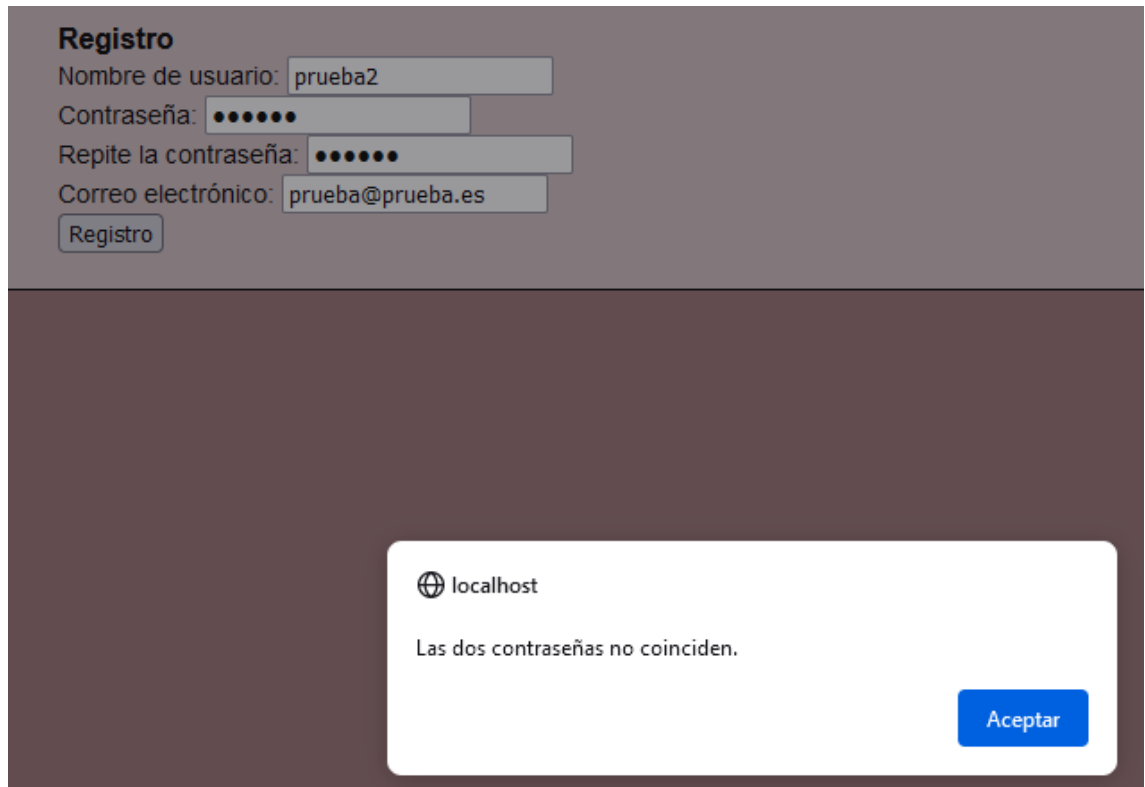
Repite la contraseña:

Correo electrónico:

localhost

El nombre de usuario no puede tener más de 30 caracteres.

- Otro error ocurrirá si el campo de contraseña y el de repetir contraseña no coinciden.



Registro

Nombre de usuario: prueba2

Contraseña: ●●●●●●

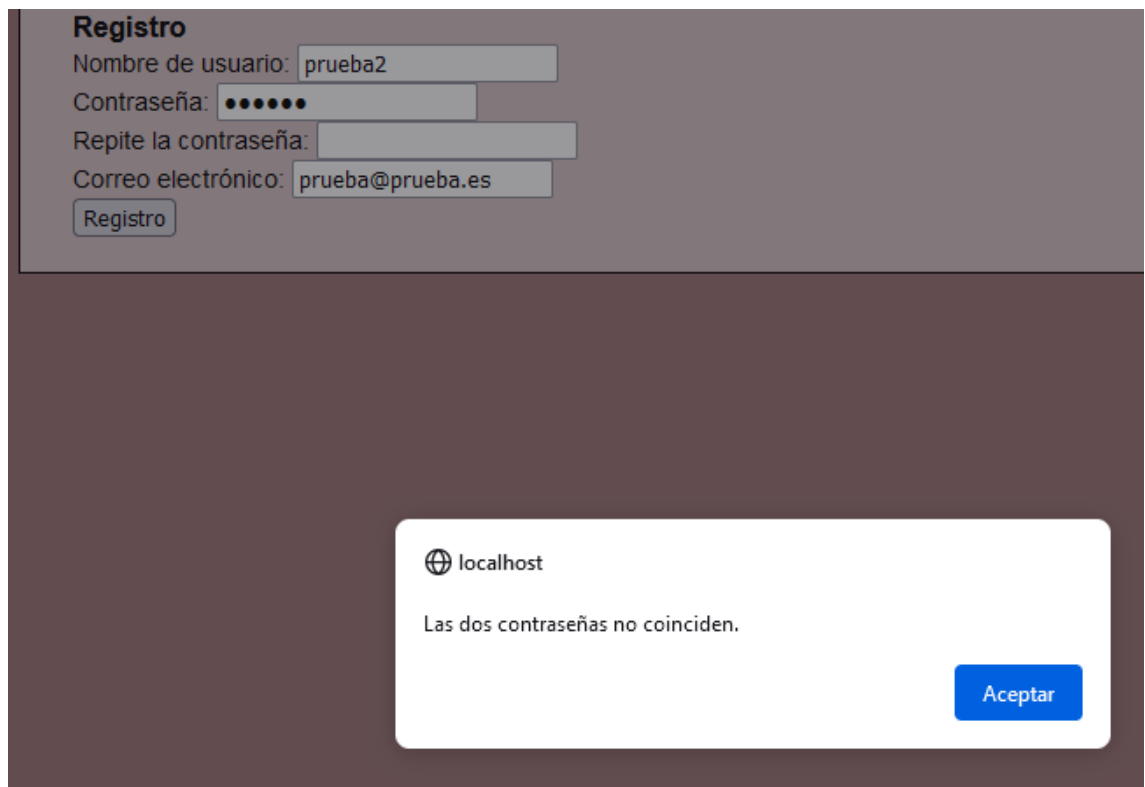
Repite la contraseña: ●●●●●●

Correo electrónico: prueba@prueba.es

localhost

Las dos contraseñas no coinciden.

- El último error se dará si cualquiera de los campos de contraseña están vacíos.



Registro

Nombre de usuario: prueba2

Contraseña: ●●●●●●

Repite la contraseña:

Correo electrónico: prueba@prueba.es

localhost

Las dos contraseñas no coinciden.

- El correo electrónico es opcional, por lo que no se mostrará ningún error.

The image shows two screenshots of a web application interface. The top screenshot displays a registration form titled 'Registro'. It includes input fields for 'Nombre de usuario:' (containing 'prueba2'), 'Contraseña:' (masked with dots), 'Repite la contraseña:' (masked with dots), and 'Correo electrónico:'. Below these fields is a 'Registro' button. The bottom screenshot shows the same interface after a successful registration, displaying the message 'Registro realizado con éxito. ¡Ahora puedes [iniciar sesión](#) y empezar a postear!'.

Si todos los datos son correctos y el usuario no está duplicado, se procederá a insertar mediante MySQL los datos en la base de datos. Por defecto, al usuario se le asignará el nivel de usuario 0, que significa que este usuario no es administrador. Si se quisiera cambiar, lo haría una persona con acceso a la base de datos (el administrador) modificando el campo **nivel_usu** de 0 a 1. La contraseña se hashearé cuando se incluya en la base de datos mediante la función sha1, que le hace un encriptado sha1 para que ni siquiera una persona con acceso a la base de datos pueda saber la contraseña de cada usuario. Además, se utilizará la función **mysqli_real_escape_string** al introducir texto plano para dificultar los ataques por inyección SQL. Si la inserción no se ha podido realizar con éxito, mostrará una alerta, y si se ha podido mostrará un mensaje de éxito y un enlace al inicio de sesión.

El código fuente es el siguiente:

```
<?php
include 'connect.php';
include 'header.php';

echo '<h3>Registro</h3>';

if($_SERVER['REQUEST_METHOD'] != 'POST')
{
    echo '<form method="post" action="">
```

```

        Nombre de usuario: <input type="text"
name="nombre_usu" /> <br/>
        Contraseña: <input type="password"
name="pass_usu"> <br/>
        Repite la contraseña: <input
type="password" name="pass_usu_check"> <br/>
        Correo electrónico: <input type="email"
name="email_usu"> <br/>
        <input type="submit" value="Registro" />
    </form>';

```

```

$usuario=$_POST['nombre_usu']    ?? null;
$nombre_usu = strip_tags($usuario);

}
else {
    $sql = "SELECT
                nombre_usu,email_usu
            FROM
                usuarios
            WHERE
                nombre_usu = '" . $conn-
>real_escape_string($_POST['nombre_usu']) . "'";

    $result = $conn->query($sql);
    if(!$result)
    {
        echo "<script>alert('Algo ha fallado.
Por favor, prueba más tarde.')</script>";
    }
    else
    {
        $errors = array();

```

```

        if(isset($_POST['nombre_usu']))
        {
            while ($row =
mysqli_fetch_assoc($result)) {
                if ($_POST['nombre_usu'] =
$row['nombre_usu']) {
                    echo "<script>alert('El
nombre de usuario ya existe. Por favor, prueba otro
diferente.')

```

```
        else
        {
            echo "<script>alert('El campo
nombre de usuario no puede estar
vacío.')
```

```

        foreach($errors as $key =>
$value)
        {
            echo '<li>' . $value .
'</li>';
        }
        echo '</ul>';
        echo '<a
href="registro.php">Volver al registro</a>';
    }
    else
    {
        $sql = "INSERT INTO
                        usuarios(nombre_usu
, pass_usu, email_usu ,fecha_registro, nivel_usu)
                        VALUES('\" . $conn-
>real_escape_string($_POST['nombre_usu']) . '\",
                        '\" .
sha1($_POST['pass_usu']) . '\",
                        '\" . $conn-
>real_escape_string($_POST['email_usu']) . '\",
                        NOW(),
                        0);";

        $result = $conn->query($sql);
        if(!$result)
        {
            echo "<script>alert('Algo
ha fallado en el registro. Por favor, prueba más
tarde.')</script>";
        }
        else
        {

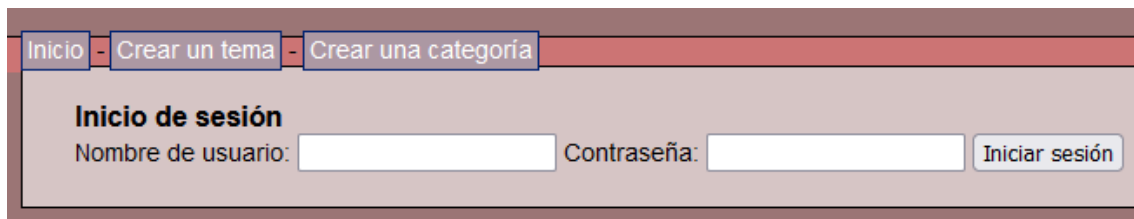
```



```
        echo 'Registro realizado  
con éxito. ¡Ahora puedes <a  
href="login.php">iniciar sesión</a> y empezar a  
postear!';  
    }  
}  
}  
}  
}  
  
include 'footer.php';  
?>
```

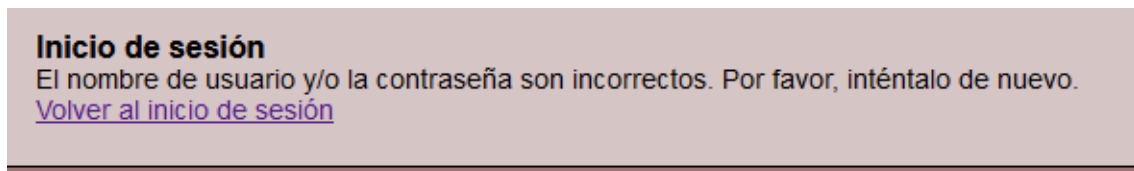
Inicio de sesión

Una vez creado un usuario, querremos iniciar sesión para ver el contenido del foro. Para esto tenemos un archivo **login.php** que realiza una selección de la base de datos.

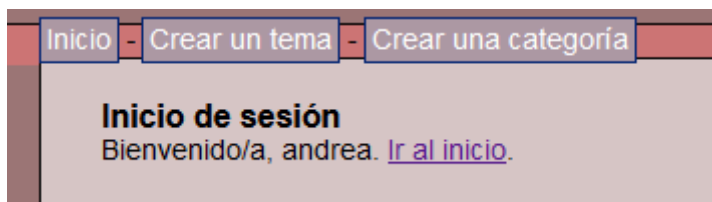


Al igual que con la página de registro, primero incluiremos **connect.php** y **header.php**. Si se detecta que ya hay una sesión iniciada, se imprimirá un mensaje diciendo que ya se ha iniciado sesión y que puedes cerrar sesión si quieres (con un enlace al archivo de cierre de sesión). Si la sesión no está iniciada y no hay ya datos enviados mediante post, saldrá un formulario que pedirá el usuario y la contraseña. Si el campo de nombre de usuario está vacío, se añadirá un texto de error al array `$errors` como anteriormente, y lo mismo ocurrirá si no hay nada en la contraseña. Si ocurre cualquiera de estas situaciones, al enviar el formulario saldrá una lista con los errores. Si no ocurre ninguno de estos errores, se procederá a seleccionar el código de usuario, el nombre y el nivel, donde el nombre de usuario sea igual que el enviado y la contraseña sea igual que la contraseña que se ha enviado hasheada. Si no se ha podido realizar el select, saldrá un error, y si se ha podido, si el nombre de usuario y la contraseña dan resultados en la base de datos se iniciará sesión y se mostrará un mensaje de bienvenida, y si no se ha podido se mostrará un error de que el nombre de usuario y la contraseña no coinciden con el contenido de la base de datos.

Error



Inicio de sesión con éxito



El código fuente es el siguiente:

```
<?php
include 'connect.php';
include 'header.php';

echo '<h3>Inicio de sesión</h3>';

if(isset($_SESSION['sesion_iniciada']) &&
$_SESSION['sesion_iniciada'] == true)
{
    echo 'Ya has iniciado sesión, puedes <a
href="cierresesion.php" onclick="return
confirm(¿Seguro que quieres cerrar
sesión?);">cerrar sesión</a> si quieres.';
}
else
{
    if($_SERVER['REQUEST_METHOD'] != 'POST')
    {
        echo '<form method="post" action="">
            Nombre de usuario: <input type="text"
name="nombre_usu" />
            Contraseña: <input type="password"
name="pass_usu">
            <input type="submit" value="Iniciar
sesión" />
        </form>';
    }
    else
    {
        $errors = array();

        if(!isset($_POST['nombre_usu']))
```

```
{
    $errors[] = 'El campo nombre de usuario
no puede estar vacío.';
}

if(!isset($_POST['pass_usu']))
{
    $errors[] = 'El campo contraseña no
puede estar vacío.';
}

if(!empty($errors))
{
    echo 'Un par de campos no han sido
rellenados correctamente.';
    echo '<ul>';
    foreach($errors as $key => $value)
    {
        echo '<li>' . $value . '</li>';
    }
    echo '</ul>';
}
else
{
    $sql = "SELECT
            cod_usuario,
            nombre_usu,
            nivel_usu
        FROM
            usuarios
        WHERE
            nombre_usu = '" . $conn-
>real_escape_string($_POST['nombre_usu']) . "'
        AND
```

```

                                pass_usu = '"' .
sha1($_POST['pass_usu'])) . '"';

    $result = $conn->query($sql);
    if(!$result)
    {
        die('Algo ha ido mal en el inicio
de sesión. Por favor, inténtalo de nuevo más
tarde.' . mysqli_connect_errno());
    }
    else
    {
        if(mysqli_num_rows($result) == 0)
        {
            echo 'El nombre de usuario y/o
la contraseña son incorrectos. Por favor, inténtalo
de nuevo.<br/>';

            echo '<a
href="login.php">Volver al inicio de sesión</a>';
        }
        else
        {
            $_SESSION['sesion_iniciada'] =
true;

            while($row =
mysqli_fetch_assoc($result))
            {
                $_SESSION['cod_usuario']
= $row['cod_usuario'];
                $_SESSION['nombre_usu'] =
$row['nombre_usu'];
                $_SESSION['nivel_usu'] =
$row['nivel_usu'];

```

```

    }

    echo 'Bienvenido/a, ' .
$_SESSION['nombre_usu'] . ' . <a href="index.php">Ir
al inicio</a>.';
    }
}
}
}
}

include 'footer.php';
?>

```

Cierre de sesión

Una vez iniciada la sesión, en el menú de navegación aparecerá nuestro nombre de usuario y al lado un enlace para cerrar sesión que irá al archivo **cierresesion.php**. Este archivo simplemente borrará la sesión y hará una redirección a la página de inicio de sesión.

El código fuente es el siguiente:

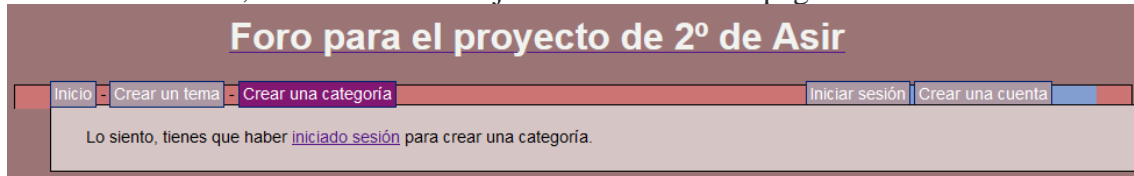
```

<?php
session_start();
session_unset();
session_destroy();
header ('Location: login.php');
?>

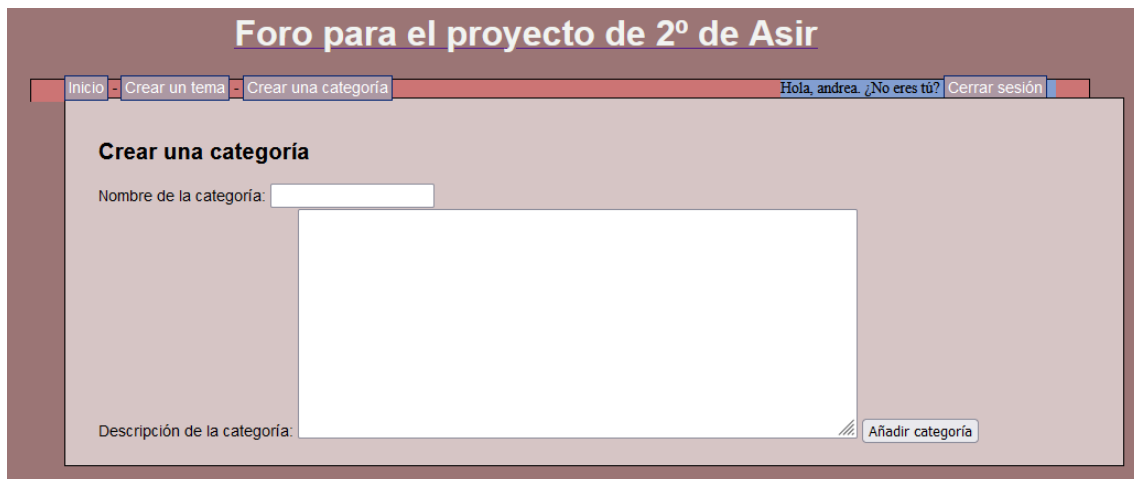
```

Crear una categoría

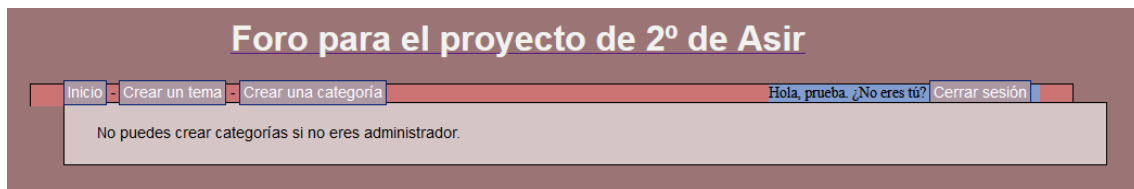
Como se explicó antes, el foro tiene tres niveles de contenidos. En primer lugar, están las categorías, donde se englobarán cada uno de los temas, y dentro de los temas habrá posts, que son respuestas a estos temas. Para crear categorías necesitaremos ser administradores, es decir, tener un 1 en la columna nivel_usu de nuestro usuario en la base de datos. Los usuarios estándar no podrán crear categorías, solamente temas y respuestas a esos temas. Para empezar, la página comprobará que la sesión está iniciada. Si no está iniciada, nos dará un mensaje con un enlace a la página de inicio de sesión.



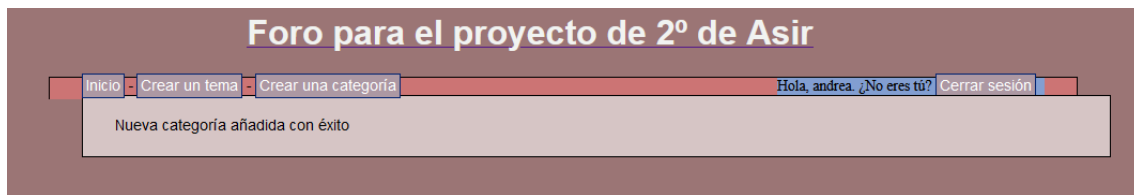
Una vez iniciada, como dije antes, necesitaremos ser administradores. Para comprobarlo, la página cogerá el valor nivel_usu de la sesión, que declaramos al hacer el inicio de sesión junto al nombre de usuario y el código del usuario. Si este valor es 1, es decir, administrador, se imprimirá un formulario donde podrán rellenarse el nombre de la categoría y su descripción.



De lo contrario, al ser un usuario estándar saldrá un mensaje de que no es posible crear categorías sin ser administrador.



Al enviar la categoría con éxito, saldrá un mensaje de éxito al crear la categoría, si no si se produjese un error en la base de datos saldría el código de error.



El código fuente es el siguiente:

```
<?php
include 'header.php';
include 'connect.php';

$sesion_iniciada=$_SESSION['sesion_iniciada']    ??
null;

if($sesion_iniciada == false)
{
    echo 'Lo siento, tienes que haber <a
href="login.php">iniciado sesión</a> para crear una
categoría.';
}
else {
    if($_SERVER['REQUEST_METHOD'] != 'POST')
    {
        if($_SESSION['nivel_usu'] == 1)
        {
            echo '<h2>Crear una
categoría</h2>';
            echo '<form method="post"
action="">
                Nombre de la categoría:
<input type="text" name="nombre_cat" /> <br/>
                Descripción de la
categoría: <textarea name="descr_cat" /></textarea>
```

```

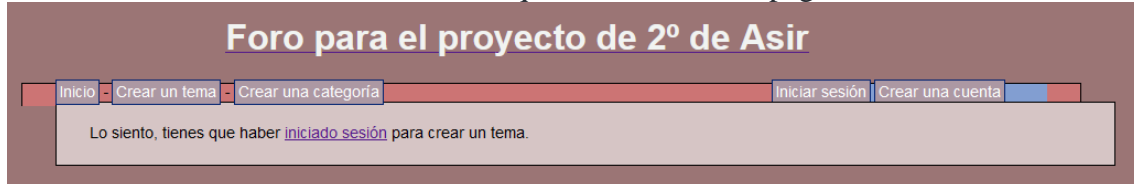
                                <input type="submit"
value="Añadir categoría" />
                                </form>';
                                }
                                else
                                {
                                    echo 'No puedes crear
categorías si no eres administrador.';
                                }
                            }
                            else
                            {
                                $sql = "INSERT INTO categorias(nombre_cat,
descr_cat)
                                VALUES('" . $conn-
>real_escape_string($_POST['nombre_cat']) . "',
                                    '" . $conn-
>real_escape_string($_POST['descr_cat']) . "');"

                                $result = $conn->query($sql);
                                if(!$result)
                                {
                                    die('ERROR: ' .
mysqli_connect_errno());
                                }
                                else
                                {
                                    echo 'Nueva categoría añadida con
éxito';
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
include 'footer.php';?>

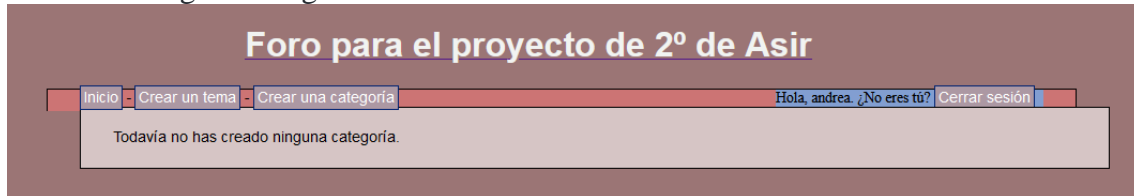
```

Crear un tema

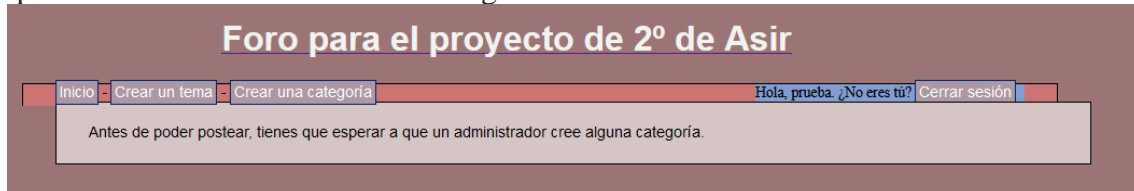
Teniendo ya categorías, podremos ir a crear un tema, tanto siendo usuario administrador como siendo usuario estándar. Primero comprobará que hemos iniciado sesión, si no lo hemos hecho saldrá un error indicando que debemos ir a la página de inicio de sesión.



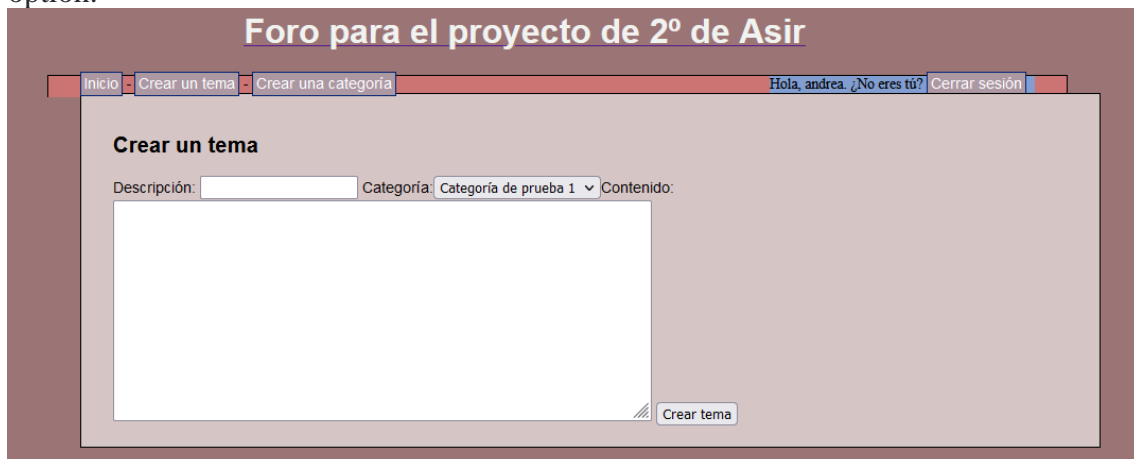
Si tenemos la sesión iniciada, seleccionaremos todas las categorías que haya en la base de datos. Si diese un error en la conexión saldría un mensaje de error y su código de error, y si no hay un error pero no hubiese ninguna categoría habría dos casos. El primer caso sería que le saliese a un usuario administrador, por lo que le saldría que todavía no ha creado ninguna categoría.



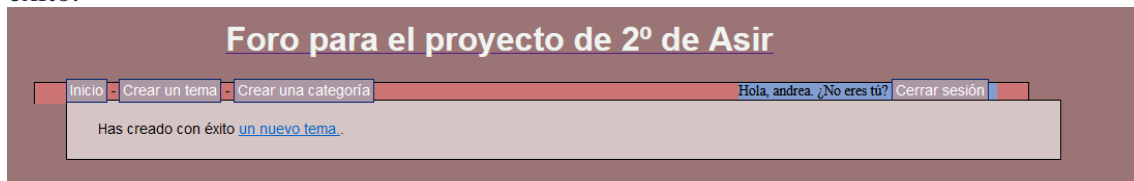
El segundo caso sería el de un usuario estándar, al que le saldría que tiene que esperar a que un administrador crease una categoría.



Habiendo categorías, saldrá un formulario para crear un tema, donde habrá un campo para la descripción o asunto del tema, otro para el texto del tema, y uno tipo select que contendrá un bucle while que diga que despliegue todas las categorías cada una como un option.



Cuando ya haya datos enviados, haremos una consulta SQL con un left join, donde vamos a seleccionar las columnas cod_tema, descrip_tema, fecha_tema y cat_tema de la tabla temas y las columnas cod_cat, nombre_cat y descr_cat de la tabla categorías, pero solo donde coincida la columna categorías.cod_cat con la columna temas.cat_tema, para así recoger todos los temas que tenga la categoría seleccionada. Si no ha habido ningún error, se procederá a insertar el contenido del formulario en la tabla de temas, con el código de la categoría elegida en la columna cat_tema y con el código del usuario que hay en la sesión en la columna autor_tema. A la vez, la descripción del tema se insertará también en la tabla posts en la columna texto_post, junto a la fecha de inserción, el código del tema en tema_post y el código de usuario en autor_post. Si no hay ningún error, se mostrará un mensaje de éxito.



El código fuente es el siguiente:

```
<?php
include 'connect.php';
include 'header.php';

$sesion_iniciada=$_SESSION['sesion_iniciada']    ??
null;

if($sesion_iniciada == false)
{
    echo 'Lo siento, tienes que haber <a
href="login.php">iniciado sesión</a> para crear un
tema.';
}
else
{
    if($_SERVER['REQUEST_METHOD'] != 'POST')
    {
        $sql = "SELECT
                    cod_cat,
                    nombre_cat,
```

```
        descr_cat
    FROM
        categorias";

$result = $conn->query($sql);

if(!$result)
{
    die('Error seleccionando de la base de
datos. Por favor, inténtalo de nuevo más tarde.' .
mysqli_connect_errno());
}
else
{
    if(mysqli_num_rows($result) == 0)
    {
        if($_SESSION['nivel_usu'] == 1)
        {
            echo 'Todavía no has creado
ninguna categoría.';
        }
        else
        {
            echo 'Antes de poder postear,
tienes que esperar a que un administrador cree
alguna categoría.';
        }
    }
    else
    {
        echo '<h2>Crear un tema</h2>';
        echo '<form method="post"
action="">
```

```

                Descripción: <input type="text"
name="descrip_tema" />
                Categoría:';

                echo '<select name="nombre_cat">';
                while($row =
mysqli_fetch_assoc($result))
                {
                        echo '<option value="' .
$row['cod_cat'] . '">' . $row['nombre_cat'] .
'</option>';
                }
                echo '</select>';

                echo 'Contenido: <textarea
name="texto_post" /></textarea>
                <input type="submit"
value="Crear tema" />
                </form>';
        }
    }
}
else
{
    $sql = "SELECT
            temas.cod_tema,
            temas.descrip_tema,
            temas.fecha_tema,
            temas.cat_tema,
            categorias.cod_cat,
            categorias.nombre_cat,
            categorias.descr_cat
        FROM
            temas
    
```

```

        LEFT JOIN
            categorias
        ON
            temas.cat_tema =
categorias.cod_cat";

$query = "BEGIN WORK;";
$result = $conn->query($sql);

if(!$result)
{
    die('Ha ocurrido un error creando tu
tema. Por favor, inténtalo de nuevo más tarde.' .
mysqli_connect_errno());
}
else
{
    $sql = "INSERT INTO
            temas(descrip_tema,
                fecha_tema,
                cat_tema,
                autor_tema)
            VALUES('" . $conn-
>real_escape_string($_POST['descrip_tema']) . "',
                NOW(),
                " . $conn-
>real_escape_string($_POST['nombre_cat']) . ",
                " .
$_SESSION['cod_usuario'] . "
            )";

    $result = $conn->query($sql);
    if(!$result)
    {

```



```

        die('Ha ocurrido un error
insertando tu tema. Por favor, inténtalo de nuevo
más tarde.' . mysqli_connect_errno());
        mysqli_rollback($conn);
        $result = $conn->query($sql);
    }
    else
    {
        $cod_tema =
mysqli_insert_id($conn);

        $sql = "INSERT INTO
                    posts(texto_post,
                        fecha_post,
                        tema_post,
                        autor_post)
                    VALUES
                        ('" . $conn-
>real_escape_string($_POST['descrip_tema']) . "',
                        NOW(),
                        " . $cod_tema .
",
                        " .
$_SESSION['cod_usuario'] . "
                    )";
        $result = $conn->query($sql);

        if(!$result)
        {
            die('Ha ocurrido un error
insertando tu post. Por favor, inténtalo de nuevo
más tarde.' . mysqli_connect_errno());
            mysqli_rollback($conn);
            $result = $conn->query($sql);

```

```

    }
    else
    {
        $sql = "COMMIT;";
        $result = $conn->query($sql);

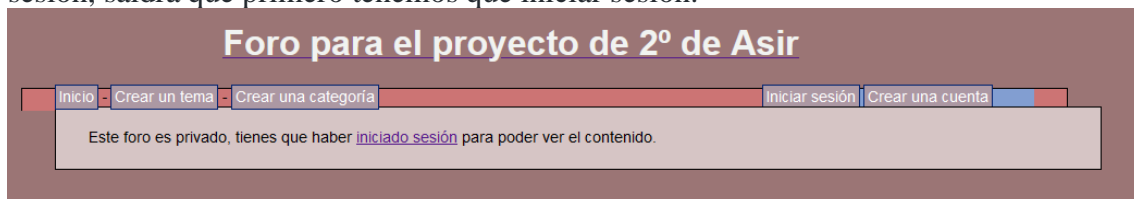
        echo 'Has creado con éxito <a
href="tema.php?cod_tema='. $cod_tema . '">un nuevo
tema.</a>.';
    }
}
}
}
}

include 'footer.php';
?>

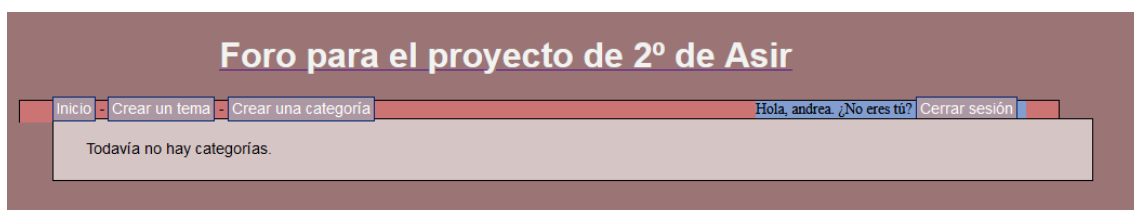
```

Inicio

Una vez ya haya temas, querremos verlos. Para eso tenemos el archivo **index.php**, que es la página de inicio del foro. Como el foro es privado, si no tenemos iniciada la sesión, saldrá que primero tenemos que iniciar sesión.



Una vez iniciada la sesión, si todavía no hay contenido saldrá este mensaje:



En cambio, si hay contenido saldrá listado, pero solamente saldrán las categorías que tengan un tema dentro. Como se puede ver, aparecerá una tabla donde a la izquierda saldrá el nombre de cada una de las categorías y a la derecha el último tema creado. Esto se ha hecho mediante un select con left join igual al de la página **crear_tema.php**.

Foro para el proyecto de 2º de Asir	
Inicio	Hola, andrea. ¿No eres tú?
Crear un tema	Cerrar sesión
Crear una categoría	
Categoría	Último tema
Categoría de prueba 1 Esta es la descripción de la categoría de prueba 1.	Tema de prueba 1 en la fecha 2022-12-07 16:43:47

El código fuente es el siguiente:

```
<?php
include 'connect.php';
include 'header.php';
if($sesion_iniciada == false)
{
    echo 'Este foro es privado, tienes que haber <a
href="login.php">iniciado sesión</a> para poder ver
el contenido.';
}
else {
    $sql = "SELECT
                temas.cod_tema,
                temas.descrip_tema,
                temas.fecha_tema,
                temas.cat_tema,
                categorias.cod_cat,
                categorias.nombre_cat,
                categorias.descr_cat
            FROM
                temas
            LEFT JOIN
                categorias
            ON
```

```

        temas.cat_tema =
categorias.cod_cat";

$result = $conn->query($sql);

if(!$result)
{
    echo 'No se han podido desplegar las
categorías. Por favor, inténtalo de nuevo.';
}
else
{
    if(mysqli_num_rows($result) == 0)
    {
        echo 'Todavía no hay categorías.';
    }
    else
    {
        echo '<table border="1">
            <tr>
                <th>Categoría</th>
                <th>Último tema</th>
            </tr>';

        while($row =
mysqli_fetch_assoc($result))
        {
            echo '<tr>';
            echo '<td class="leftpart">';
            echo '<h3><a
href="categoria.php?cod_cat='.$row['cod_cat'].'">
. $row['nombre_cat'] . '</a></h3>' .
$row['descr_cat'];
            echo '</td>';

```

```

        echo '<td class="rightpart">';
        echo '<a
href="tema.php?cod_tema='.$row['cod_tema'].'">'.$ro
w['descrip_tema'].'</a> en la fecha
'.$row['fecha_tema'].'';
        echo '</td>';
        echo '</tr>';
    }
}
}
}
include 'footer.php';
?>

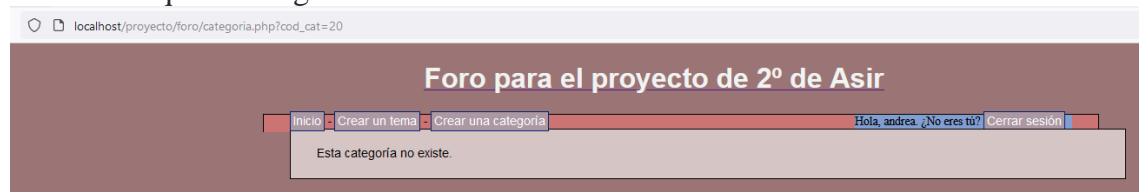
```

Vista de categoría

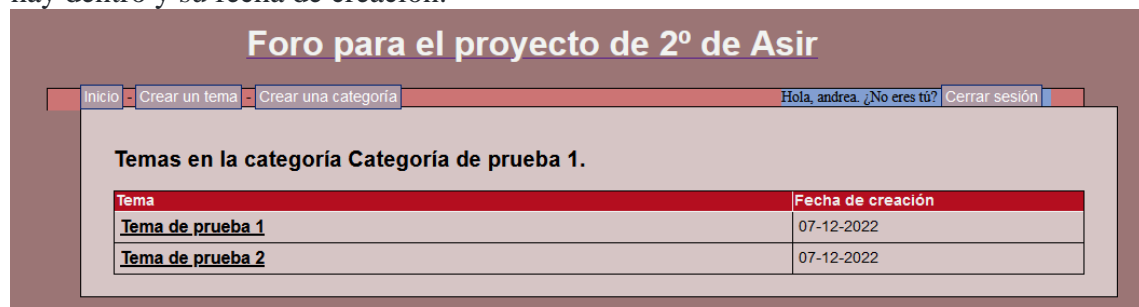
Ahora queremos que al pinchar en el nombre de una categoría nos muestre todos los temas que hay dentro. Para ello vamos a crear un archivo **categoria.php**, pero no accederemos directamente a este archivo sino a la categoría en cuestión, que habremos mandado antes mediante el protocolo GET en el **index.php** al pulsar el enlace, por lo que en el enlace saldrá el código de la categoría.

`localhost/proyecto/foro/categoria.php?cod_cat=9`

Si intentásemos acceder directamente poniendo el código de una categoría cualquiera, nos saldría que la categoría no existe.



Al acceder a una categoría que sí que exista, saldrá una tabla con todos los temas que hay dentro y su fecha de creación.



El código fuente es el siguiente:

```
<?php
include 'connect.php';
include 'header.php';

$sql = "SELECT
        cod_cat,
        nombre_cat,
        descr_cat
    FROM
        categorias
    WHERE
        cod_cat =
'".mysqli_real_escape_string($conn,$_GET["cod_cat"])
'."";

$result = $conn->query($sql);

if(!$result)
{
    die('ERROR: No se ha podido desplegar la
categoría, por favor, inténtalo de nuevo más
tarde.' . mysqli_connect_errno());
}
else
{
    if(mysqli_num_rows($result) == 0)
    {
        echo 'Esta categoría no existe.';
    }
    else
    {
        while($row = mysqli_fetch_assoc($result))
```

```
{
    echo '<h2>Temas en la categoría ' .
$row['nombre_cat'] . '</h2>';
}

$sql = "SELECT
        cod_tema,
        descrip_tema,
        fecha_tema,
        cat_tema
    FROM
        temas
    WHERE
        cat_tema =
'".mysqli_real_escape_string($conn,$_GET["cod_cat"])
'."';";

$result = $conn->query($sql);

if(!$result)
{
    die('ERROR: Los temas no han podido ser
desplegados, por favor, inténtalo de nuevo más
tarde.' . mysqli_connect_errno());
}
else
{
    if(mysqli_num_rows($result) == 0)
    {
        echo 'Todavía no hay temas en esta
categoría.';
    }
}
```

```

        else
        {
            echo '<table border="1">
                <tr>
                    <th>Tema</th>
                    <th>Fecha de creación</th>
                </tr>';

            while($row =
mysqli_fetch_assoc($result))
            {
                echo '<tr>';
                echo '<td
class="leftpart">';

                    echo '<h3><a
href="tema.php?cod_tema=' . $row['cod_tema'] . '">'
. $row['descrip_tema'] . '</a><h3>';
                echo '</td>';
                echo '<td
class="rightpart">';

                    echo date('d-m-Y',
strtotime($row['fecha_tema']));
                echo '</td>';
                echo '</tr>';
            }
        }
    }
}

include 'footer.php';
?>

```


Vista de tema

A los temas accederemos de una forma similar a las categorías mediante el enlace el protocolo GET. Dentro de cada tema saldrá desplegado tanto el tema como todas sus respuestas o posts, estando a un lado de la tabla el contenido de ese post y al otro el usuario y la fecha de publicación. Habrá un formulario para enviar respuestas nuevas.

Contenido	Fecha y autor
Tema de prueba 1	07-12-2022 andrea

El código fuente es el siguiente:

```
<?php
include 'connect.php';
include 'header.php';

$sql= "SELECT
    posts.tema_post,
    posts.texto_post,
    posts.fecha_post,
    posts.autor_post,
    usuarios.cod_usuario,
    usuarios.nombre_usu
FROM
    posts
LEFT JOIN
    usuarios
ON
    posts.autor_post = usuarios.cod_usuario
```

```

WHERE
    posts.tema_post =
    '".mysqli_real_escape_string($conn,$_GET["cod_tema"
    ]).'";

$result = $conn->query($sql);

if(!$result)
{
    die('La categoría no ha podido ser desplegada.
    Por favor, inténtalo de nuevo más tarde.' .
    mysqli_connect_errno());
}
else
{
    echo '<div id="tablatema">
    <table border="1">
        <tr>
            <th>Contenido</th><th>Fecha
y autor</th>
        </tr>';

    while($row =
mysqli_fetch_assoc($result))
    {
        echo '<tr>';
        echo '<td
class="leftpart">';
        echo
$row['texto_post'];
        echo '</td>';
        echo '<td
class="rightpart">';

```

```

                                echo '<p>' . date('d-m-
Y', strtotime($row['fecha_post'])) . '</p><p>' .
$row['nombre_usu'] . '</p>';
                                echo '</td>';
                                echo '</tr>';
                                }

                                echo '</div>';
                                }

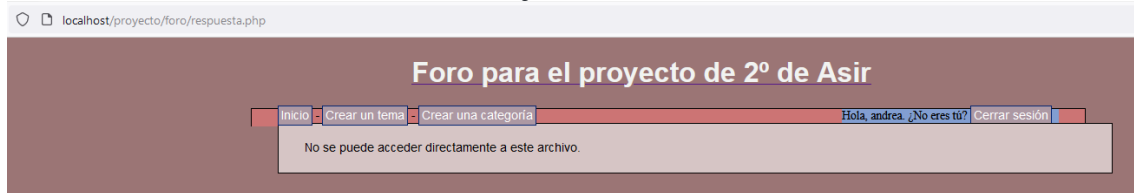
                                echo '
                                <div id="respuesta">
                                <h3>Escribe aquí tu respuesta</h3>
                                <form method="post"
                                action="respuesta.php?cod_tema=' . $_GET["cod_tema"] .
                                '">
                                <textarea name="texto-post"></textarea>
                                <input type="submit" value="Responder" />
                                </form>
                                </div>
                                '
                                ;

                                include 'footer.php';
?>

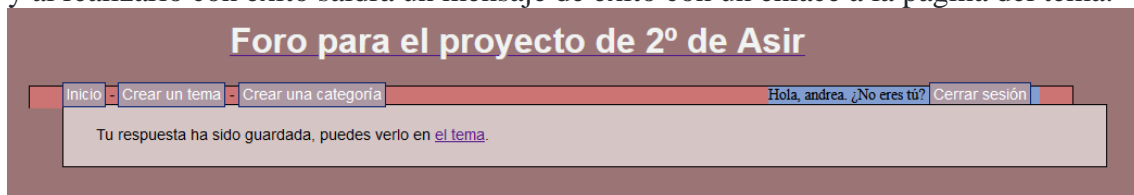
```

Responder a temas

Para procesar lo que se introduzca en el formulario mencionado anteriormente para responder, necesitaremos el fichero **respuesta.php**. No se podrá acceder a este archivo directamente, es decir, cuando no haya ningún dato guardado mediante el protocolo POST. Si lo intentamos saldrá un mensaje.



Si hay datos, estos se procesarán y se guardarán en la base de datos en la tabla de posts, y al realizarlo con éxito saldrá un mensaje de éxito con un enlace a la página del tema.



Finalmente, al acceder a la vista del tema veremos que la respuesta se ha guardado correctamente.



El código fuente es el siguiente:

```
<?php
include 'connect.php';
include 'header.php';

if($_SERVER['REQUEST_METHOD'] != 'POST')
```

```

{
    echo 'No se puede acceder directamente a este
archivo.';
}
else
{
    if(!$_SESSION['sesion_iniciada'])
    {
        echo 'Tienes que haber iniciado sesión para
poder responder.';
    } else {
        $sql = "INSERT INTO
                posts(texto_post,
                    fecha_post,
                    tema_post,
                    autor_post)
                VALUES ('" . $_POST['texto-post'] .
'',
                    NOW(),
                    '" .
mysqli_real_escape_string($conn, $_GET["cod_tema"]))
. "',
                    " .
$_SESSION['cod_usuario'] . "));";

        $result = $conn->query($sql);

        if (!$result) {
            die('Tu respuesta no ha podido ser
guardada. Por favor, inténtalo de nuevo más tarde.
Error: ' . mysqli_connect_errno());
        } else {
            echo 'Tu respuesta ha sido
guardada, puedes verlo en <a

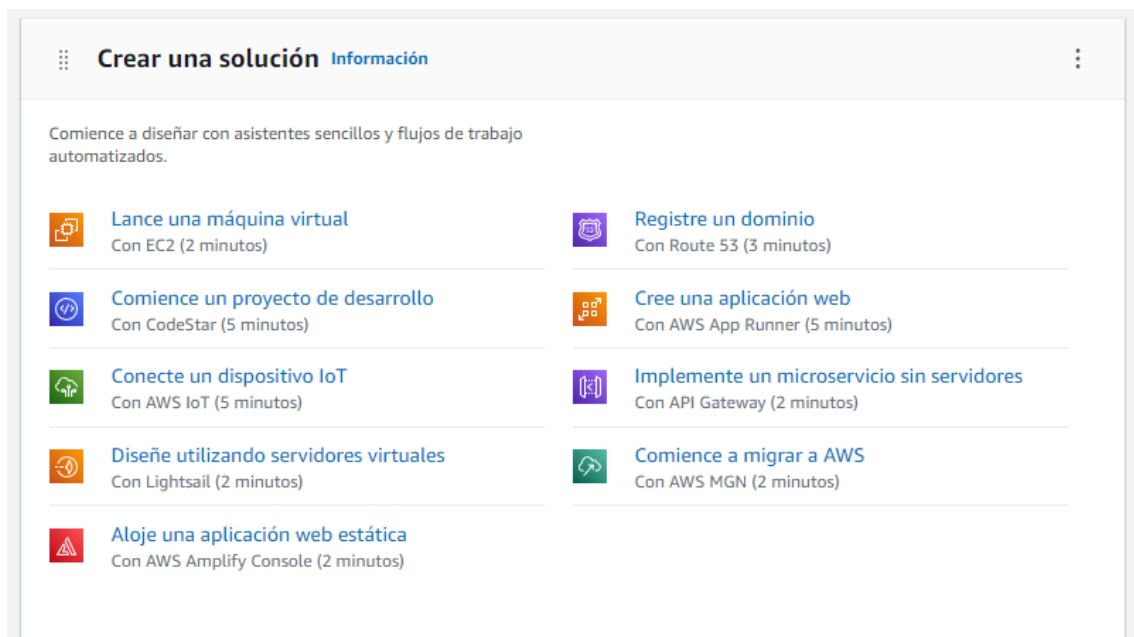
```

```
href="tema.php?cod_tema=' .  
htmlentities($_GET['cod_tema']) . '">el tema</a>.';  
    }  
}  
}  
  
include 'footer.php';  
?>
```

Creación del servidor web y la base de datos en AWS

Despliegue de la instancia de EC2

En primer lugar, hay que ir a <https://aws.amazon.com/es/>, donde crearemos una cuenta. Una vez creada, iremos a la sección dentro del panel llamada "Crear una solución". Allí elegiremos "Lance una página máquina virtual", que nos llevará a la consola de creación de EC2.



Una vez allí, en primer lugar le daremos un nombre a la máquina y en segundo lugar elegiremos un sistema operativo, pudiendo elegir Amazon Linux, Amazon Linux 2, Ubuntu, Windows Server, Red Hat, etcétera. En este caso elegiré Amazon Linux 2, con la arquitectura de 64 bits. Al lado de la arquitectura veremos que sale el ID de la AMI. Esta es la imagen utilizada para crear la máquina.

Nombre y etiquetas

Información

Nombre

Agregar etiquetas adicionales

▼ Imágenes de aplicaciones y sistemas operativos (Amazon Machine Image)

Información

Una AMI es una plantilla que contiene la configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) necesaria para lanzar la instancia. Busque o examine las AMI si no ve lo que busca a continuación.

Inicio rápido

Amazon Linux

Ubuntu

Windows

Red Hat

SUSE Linux

Buscar más AMI

Incluidas las AMI de AWS, Marketplace y la comunidad

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

ami-0114de0ad847da090 (64 bits (x86)) / ami-0743c3f3f38fdc794 (64 bits (Arm))

Virtualización: hvm Habilitado para ENA: true Tipo de dispositivo raíz: ebs

Apto para la capa gratuita

Descripción

Amazon Linux 2 Kernel 5.10 AMI 2.0.20221103.3 x86_64 HVM gp2

Arquitectura

64 bits (x86)

ID de AMI

ami-0114de0ad847da090

Proveedor verificado

Posteriormente se nos dará a elegir el tipo de instancia. Esto define el número de núcleos de CPU y la cantidad de memoria RAM que utilizará. Con la micro para esta página es suficiente ya que no se espera un tráfico muy elevado y además es apta para la capa gratuita. Para otro tipo de recursos que se sepa que van a tener un tráfico mayor, sería necesario coger un tipo de instancia mayor, los precios de instancia salen desplegados debajo para eso. Una vez elegido el tipo, crearemos un par de claves para poder acceder por SSH.

▼
Tipo de instancia
Información

Tipo de instancia

t3.micro

Familia: t3
2 vCPU
1 GiB Memoria

Bajo demanda Linux precios: 0.0114 USD por hora

Bajo demanda Windows precios: 0.0206 USD por hora

Apto para la capa gratuita

▼

[Comparar tipos de instancias](#)

▼
Par de claves (inicio de sesión)
Información

Puede utilizar un par de claves para conectarse de forma segura a la instancia. Asegúrese de que tiene acceso al par de claves seleccionado antes de lanzar la instancia.

Nombre del par de claves - obligatorio

Seleccionar

▼

↺

[Crear un nuevo par de claves](#)

Le daremos un nombre a ese par de claves y un tipo, para lo cual yo he usado RSA, que es un sistema criptográfico que me va a generar una clave de 2048 bits, por lo cual es bastante seguro. Elegiré como formato .pem ya que voy a usar OpenSSH en Ubuntu para acceder, pero si quisiera utilizar la aplicación Putty para Windows elegiría .ppk. Una vez le demos a crear, se nos descargará automáticamente el archivo en el ordenador, y deberemos guardarlo muy bien para no perder el acceso por SSH a la máquina.

Crear par de claves

Los pares de claves le permiten conectarse a la instancia de forma segura.

Escriba el nombre del par de claves a continuación. Cuando se lo pida, almacene la clave privada en una ubicación segura y accesible de su equipo. **Lo necesitará más adelante para conectarse a la instancia.** [Más información](#)

Nombre del par de claves

El nombre puede incluir hasta 255 caracteres ASCII. No puede incluir espacios al principio ni al final.

Tipo de par de claves

☒ RSA
Par de claves públicas y privadas cifradas por RSA

☐ ED25519
Los pares de claves privadas y públicas cifradas ED25519 (no se admite para instancias de Windows)

Formato de archivo de clave privada

☒ .pem
Para usar con OpenSSH

☐ .ppk
Para usar con PuTTY

Cancelar Crear par de claves

Después entraremos en la configuración de red, donde nos saldrá el nombre de red que se va a utilizar. Crearemos un grupo de seguridad, lo cual es un firewall, en el cual permitamos el tráfico SSH solamente desde mi IP para mayor seguridad, y el tráfico HTTP (puerto 80) y HTTPS (puerto 443) desde cualquier punto de Internet. Esta configuración se podrá modificar después si es necesario.

▼ Configuraciones de red

Editar

Red

vpc-0b5063ed0a21e1c47

Subred

Sin preferencia (subred predeterminada en cualquier zona de disponibilidad)

Asignar automáticamente la IP pública

Habilitar

Firewall (grupos de seguridad) Información

Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

☒ Crear grupo de seguridad

☐ Seleccionar un grupo de seguridad existente

Crearemos un nuevo grupo de seguridad denominado "launch-wizard-1" con las siguientes reglas:

☒ Permitir el tráfico de SSH desde

Ayuda a establecer conexión con la instancia

Mi IP

██████████/32

☒ Permitir el tráfico de HTTPS desde Internet

Para configurar un punto de enlace, por ejemplo, al crear un servidor web

☒ Permitir el tráfico de HTTP desde Internet

Para configurar un punto de enlace, por ejemplo, al crear un servidor web

Finalmente, configuraremos el almacenamiento. Aquí elegiremos la cantidad de almacenamiento que queremos tener en la máquina, y el tipo de unidad de almacenamiento. Por ejemplo, gp2 es un SSD de uso general.

▼ Configurar almacenamiento Información

Avanzado

1x

8

GiB

gp2

Volumen raíz (Sin cifrar)

Agregar un nuevo volumen

La AMI seleccionada contiene más volúmenes de almacén de instancias de los que permite la instancia. Solo se podrá obtener acceso desde la instancia a los primeros 0 volúmenes de almacén de instancias de la AMI

0 x sistemas de archivos

Editar

Cancelar

Lanzar instancia

54

Al elegir el tipo podremos elegir si queremos que el volumen se elimine cuando eliminemos la máquina virtual y si queremos que el disco esté cifrado.

Tamaño (GiB) Información <input type="text" value="8"/>	Tipo de volumen Información <input type="text" value="gp2"/>	IOPS Información 100 / 3000
Eliminar cuando termine Información <input type="text" value="Sí"/>	Cifrado Información <input type="text" value="No cifrado"/>	Clave de KMS Información <input type="text" value="Seleccionar"/>

Las claves de KMS solo se aplican cuando se establece el cifrado en este volumen.

Por último, nos iremos a la sección "Resumen", a la derecha, y nos aseguraremos de que toda la configuración está bien. Cuando terminemos, le daremos a "Lanzar instancia" y la máquina se empezaría a crear.

▼ Resumen

Número de instancias [Información](#)

Imagen de software (AMI)

Amazon Linux 2 Kernel 5.10 AMI...[más información](#)

ami-0114de0ad847da090

Tipo de servidor virtual (tipo de instancia)

t3.micro

Firewall (grupo de seguridad)

Nuevo grupo de seguridad

Almacenamiento (volúmenes)

1 volúmen(es): 8 GiB

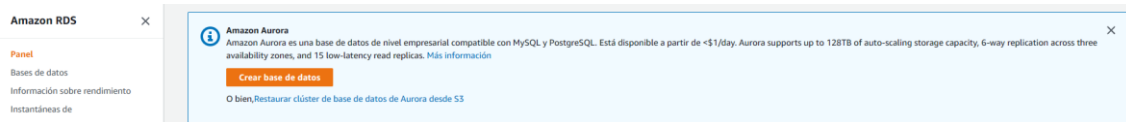
ⓘ Nivel gratuito: El primer año incluye 750 horas de uso de instancias t2.micro (o t3.micro en las regiones en las que t2.micro no esté disponible) en las AMI del nivel gratuito al mes, 30 GiB de almacenamiento de EBS, 2 millones de E/S, 1 GB de instantáneas y 100 GB de ancho de banda a Internet. **✕**

Cancelar

Lanzar instancia

Despliegue del servidor de bases de datos

Una vez creada la instancia, crearemos la base de datos. Para ello iremos a la consola de creación mediante el enlace <https://eu-south-2.console.aws.amazon.com/rds>. Le daremos a "Crear base de datos".



En la consola de creación de base de datos, elegiremos "Creación estándar", para poder elegir todos los parámetros y el motor MySQL ya que es el que se ha usado para crear el foro. En otros motores se podría elegir además otra edición, pero en este la única es "Comunidad de MySQL".



En plantillas le daremos a "Capa gratuita", no podremos elegir las opciones de "Disponibilidad y durabilidad" ya que por defecto esta plantilla nos hace una instancia de base de datos única, que para el foro es suficiente.

Plantillas

Elija una plantilla de ejemplo para adaptarla a su caso de uso.

☐ **Producción**
Utilice los valores predeterminados para disfrutar de una alta disponibilidad y de un rendimiento rápido y constante.

☐ **Desarrollo y pruebas**
Esta instancia se ha diseñado para su uso en desarrollo, fuera de un entorno de producción.

☒ **Capa gratuita**
Utilice la capa gratuita de RDS para desarrollar nuevas aplicaciones, probar aplicaciones existentes o adquirir experiencia práctica con Amazon RDS.
[Información](#)

Disponibilidad y durabilidad

Opciones de implementación [Información](#)

Las siguientes opciones de implementación están limitadas a las compatibles con el motor que ha seleccionado anteriormente.

☐ **Nuevo clúster de base de datos Multi-AZ**
Crea un clúster de base de datos con una instancia de base de datos primaria y dos instancias de base de datos en espera con capacidad de lectura, con cada instancia de base de datos en una zona de disponibilidad (AZ) diferente. Proporciona alta disponibilidad, redundancia de datos y aumenta la capacidad de incluir cargas de trabajo de lectura.

☐ **Instancia de base de datos Multi-AZ (no compatible con la instantánea de clúster de base de datos Multi-AZ)**
Crea una instancia de base de datos primaria y una instancia de base de datos en espera en una zona de disponibilidad diferente. Proporciona alta disponibilidad y redundancia de datos, pero la instancia de base de datos en espera no admite conexiones para cargas de trabajo de lectura.

☐ **Instancia de base de datos única (no compatible con la instantánea de clúster de base de datos Multi-AZ)**
Crea una sola instancia de base de datos sin instancias de base de datos en espera.

Le daremos un nombre al clúster (este nombre no es el de la base de datos, si no el de la instancia donde está). Crearemos un nombre de usuario maestro que podrá acceder a todo el contenido de la base de datos y una contraseña maestra.

Configuración

Identificador del clúster de base de datos [Información](#)
Escriba un nombre para el clúster de base de datos. El nombre debe ser único en relación con todos los clústeres de base de datos pertenecientes a su cuenta de AWS en la región de AWS actual.

El identificador del clúster de base de datos no distingue entre mayúsculas y minúsculas, pero se almacena todo en minúsculas (por ejemplo, "miclústerdebasededatos"). Restricciones: de 1 a 60 caracteres alfanuméricos o guiones. El primer carácter debe ser una letra. No puede contener dos guiones consecutivos. No puede terminar con un guion.

▼ Configuración de credenciales

Nombre de usuario maestro [Información](#)
Escriba un ID de inicio de sesión para el usuario maestro de la instancia de base de datos.

De 1 a 32 caracteres alfanuméricos. El primer carácter debe ser una letra.
☐ **Generación automática de contraseña**
Amazon RDS puede generar una contraseña en su nombre, o bien puede especificar su propia contraseña.

Contraseña maestra [Información](#)

Restricciones: debe tener al menos 8 caracteres ASCII imprimibles. No puede contener ninguno de los siguientes caracteres: / (barra diagonal), ' (comillas simples), " (dobles comillas) y @ (signo de arroba).

Confirmar contraseña [Información](#)

Al igual que con la instancia de EC2, elegiremos un tipo de instancia y configuraremos su almacenamiento. Pincharemos "Habilitar escalado automático de almacenamiento" para que si se alcanza el almacenamiento que se le ha asignado al principio se le asigne más hasta un máximo de 1000GB. En este caso no se llegará a usar ni siquiera los 20GB, pero es una opción por si en un futuro la base de datos va a escalar mucho y tener cada vez más datos.

Configuración de la instancia

Las opciones de configuración de la instancia de base de datos que aparecen a continuación están limitadas a las que admite el motor que ha seleccionado anteriormente.

Clase de instancia de base de datos [Información](#)

- ☐ Clases estándar (incluye clases m)
- ☐ Clases optimizadas para memoria (incluye clases r y x)
- ☒ Clases con ráfagas (incluye clases t)

db.t3.micro
2 vCPUs 1 GiB RAM Red: 2085 Mbps

☐ Incluir clases de generación anterior

Almacenamiento

Tipo de almacenamiento [Información](#)

SSD de uso general (gp2)
Rendimiento de referencia determinado por el tamaño del volumen

Almacenamiento asignado

20
GiB

El valor mínimo es 20 GiB y el valor máximo es 6144 GiB

Escalado automático de almacenamiento [Información](#)

Proporciona compatibilidad con el escalado dinámico para el almacenamiento de la base de datos en función de las necesidades de la aplicación.

☒ Habilitar escalado automático de almacenamiento

Si se habilita esta característica, el almacenamiento podrá aumentar después de que se supere el umbral especificado.

Umbral de almacenamiento máximo [Información](#)

Los cargos se aplicarán cuando la base de datos escale automáticamente el umbral especificado.

1000
GiB

El valor mínimo es 22 GiB y el valor máximo es 6144 GiB

Posteriormente le indicaremos que se conecte a un recurso de EC2, para conectarlo con la instancia realizada antes, y elegiremos el nombre de dicha instancia. Elegiremos el tipo de red que va a usar la conexión y la VPC.

Conectividad
Información

Recurso informático
Elija si desea configurar una conexión a un recurso informático para esta base de datos. La configuración de una conexión cambiará automáticamente la configuración de conectividad para que el recurso informático pueda conectarse a esta base de datos.

☐
No se conecte a un recurso informático EC2
No configure una conexión a un recurso informático para esta base de datos. Puede configurar manualmente una conexión a un recurso informático más adelante.

☒
Conectarse a un recurso informático de EC2
Configure una conexión a un recurso informático EC2 para esta base de datos.

Instancia EC2
Información
Elija la instancia EC2 que desea añadir como recurso informático para esta base de datos. Se añade un grupo de seguridad de VPC a esta instancia EC2. También se añade un grupo de seguridad de VPC a la base de datos con una regla de entrada que permite a la instancia EC2 obtener acceso a la base de datos.

i-0b966f458c7f43e88
Servidor web Proyecto

Algunas configuraciones de VPC no se pueden cambiar cuando se añade un recurso informático.
Al añadir un recurso informático EC2, se selecciona automáticamente la VPC, el grupo de subredes de DB y la configuración de acceso público de esta base de datos. Para permitir que la instancia EC2 obtenga acceso a la base de datos, se añade un grupo de seguridad de VPC rds-ec2-X a la base de datos y otro denominado ec2-rds-X a la instancia EC2. Puede eliminar el nuevo grupo de seguridad de la base de datos únicamente eliminando el recurso informático.

Tipo de red
Información
Para utilizar el modo de pila doble, asegúrese de asociar un bloque de CIDR IPv6 a una subred en la VPC que especifique.

☒
IPv4
Sus recursos solo pueden comunicarse a través del protocolo de direcciones IPv4.

☐
Modo de pila doble
Sus recursos pueden comunicarse a través de IPv4, IPv6 o ambos.

Virtual Private Cloud (VPC)
Información
Elija la VPC. La VPC define el entorno de red virtual para esta instancia de DB.

Default VPC (vpc-09f258635bb7e870b)

Solo se muestran las VPC con grupos de subredes de base de datos correspondientes.

60

Posteriormente indicaremos si queremos que la instancia de base de datos tenga una dirección IP pública o no. En este caso no la tendrá ya que solamente se accederá desde la instancia de EC2. La configuración de firewall la dejaremos como por defecto ya que ella sola creará los parámetros necesarios para poder conectar la instancia EC2 con esta.

Acceso público [Información](#)

☐ Sí
RDS asigna una dirección IP pública a la base de datos. Las instancias de Amazon EC2 y otros recursos fuera de la VPC pueden conectarse a la base de datos. Los recursos de la VPC también pueden conectarse a la base de datos. Elija uno o varios grupos de seguridad de VPC que especifiquen qué recursos pueden conectarse a la base de datos.

☒ No
RDS no asigna una dirección IP pública a la base de datos. Solo las instancias de Amazon EC2 y otros recursos dentro de la VPC pueden conectarse a la base de datos. Elija uno o varios grupos de seguridad de VPC que especifiquen qué recursos pueden conectarse a la base de datos.

Grupo de seguridad de VPC (firewall) [Información](#)
Elija uno o varios grupos de seguridad de VPC para permitir el acceso a su base de datos. Asegúrese de que las reglas del grupo de seguridad permitan el tráfico entrante adecuado.

☒ **Elegir existente**
Elegir grupos de seguridad de VPC existentes

☐ **Crear nuevo**
Crear un grupo de seguridad nuevo de VPC

Grupo de seguridad de VPC adicional

Choose one or more options ▼

default ✕

i Se agregará `rds-ec2-1` un nuevo grupo de seguridad de VPC para habilitar la conectividad con el recurso informático.

Zona de disponibilidad [Información](#)

eu-west-1b ▼

► **Configuración adicional**

Elegiremos el puerto, que por defecto será 3306 que es el puerto que usa MySQL para las conexiones.

▼ **Configuración adicional**

Puerto de la base de datos [Información](#)
Puerto TCP/IP que la base de datos usará para las conexiones de las aplicaciones.

3306

Elegiremos cómo queremos autenticarnos en esa base de datos, yo pondré que solo se requiera la contraseña.

Autenticación de bases de datos

Opciones de autenticación de bases de datos [Información](#)

- ☒ **Autenticación con contraseña**
Se autentica con las contraseñas de las bases de datos.
- ☐ **Autenticación de bases de datos con contraseña e IAM**
Se autentica con las credenciales de usuario y la contraseña de las bases de datos a través de usuarios y roles de AWS IAM.
- ☐ **Autenticación Kerberos y con contraseña**
Elija un directorio en el que desee permitir que los usuarios autorizados se autenticuen en esta instancia de base de datos a través de la autenticación Kerberos.

Por último, le echaremos un vistazo a la sección de "Costos mensuales estimados" y si estamos conformes crearemos la base de datos.


Costos mensuales estimados

La capa gratuita de Amazon RDS se encuentra disponible durante 12 meses. Cada mes natural, la capa gratuita le permitirá utilizar los recursos de Amazon RDS que se indican a continuación de forma gratuita:

- 750 h de Amazon RDS en una instancia Single-AZ db.t2.micro, db.t3.micro o db.t4g.micro.
- 20 GB de almacenamiento de uso general (SSD).
- 20 GB de capacidad para el almacenamiento de backups automatizados y para las instantáneas de bases de datos realizadas por el usuario.

[Más información sobre la capa gratuita de AWS.](#)

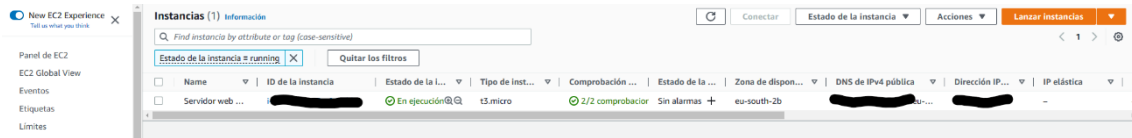
Cuando venza el periodo de uso gratuito, o si el uso de la aplicación supera las capas de uso gratuito, solo tendrá que pagar las tarifas estándar de pago por uso de los servicios que se describen en la [página de precios de Amazon RDS](#).

 Usted es responsable de asegurarse de que dispone de todos los derechos necesarios para cualquier producto o servicio de terceros que utilice con los servicios de AWS.

Cancelar Crear base de datos

Instalación del servidor web Apache

Iremos al panel de EC2 y echaremos un vistazo a las instancias que están activas. Elegimos la que queremos, que en mi caso es la única que hay, y clicamos arriba en "Conectar".



Saldrá la información para acceder por SSH y sus instrucciones. Copiaremos el comando del final y lo usaremos en Ubuntu, en una carpeta donde tengamos metida la clave .pem que descargamos antes. Este comando indica que queremos acceder por SSH con el fichero "servidor-web-proyecto.pem" mediante el usuario root a la dirección asignada a continuación. Esta dirección puede ser IP o DNS. En lugar de acceder al root, es recomendable acceder al ec2-user por seguridad, que es el usuario con poderes sudo que hay en la instancia.

EC2 > Instancias > i-██████████ > Conectarse a la instancia

Conectarse a la instancia Información

Conéctese a la instancia i-██████████ (Servidor web proyecto) mediante cualquiera de estas opciones

Administrador de sesiones

Cliente SSH

ID de la instancia
i-██████████ (Servidor web proyecto)

1. Abra un cliente SSH.
2. Localice el archivo de clave privada. La clave utilizada para lanzar esta instancia es servidor-web-proyecto.pem
3. Ejecute este comando, si es necesario, para garantizar que la clave no se pueda ver públicamente.
`chmod 400 servidor-web-proyecto.pem`
4. Conéctese a la instancia mediante su DNS público:
`ec2-██████████.compute.amazonaws.com`

Ejemplo:
`ssh -i "servidor-web-proyecto.pem" root@ec2-██████████.compute.amazonaws.com`

Nota: En la mayoría de los casos, el nombre de usuario adivinado es correcto. Sin embargo, lea las instrucciones de uso de la AMI para comprobar si el propietario de la AMI ha cambiado el nombre de usuario predeterminado de la AMI.

Cancelar

Una vez ejecutado ese comando, veremos cómo ya estamos dentro de la instancia EC2.

```
ec2-user@ip-172-31-29-122:~$ sudo ssh -i "servidor-web-proyecto.pem" ec2-user@ec2-... .eu-south-2.compute.amazo
aws.com

  __|  __|_  )
 _| (  _/  Amazon Linux 2 AMI
---|\\---|---|

https://aws.amazon.com/amazon-linux-2/
19 package(s) needed for security, out of 38 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-29-122:~]$
```

A continuación procederemos a instalar Apache y sus dependencias. En primer lugar, actualizaremos los repositorios.

```
sudo yum update -y
```

Instalaremos los paquetes php8.0 y mariadb10.5.

```
sudo amazon-linux-extras install php8.0 mariadb10.5
```

Instalaremos el paquete apache, el cuál aquí se llamará **httpd**.

```
sudo yum install -y httpd
```

Iniciaremos el servidor web.

```
sudo systemctl start httpd
```

Haremos que el servidor web se inicie cada vez que se encienda la instancia.

```
sudo systemctl enable httpd
```

Posteriormente, y crearemos el grupo **apache** y meteremos al usuario **ec2-user**.

```
sudo usermod -a -G apache ec2-user
```

Nos aseguraremos de que este grupo ahora existe.

```
[ec2-user@ip-172-31-29-122:~]$ groups
ec2-user adm wheel apache systemd-journal
```

Ahora, haremos que la carpeta **/var/www**, que es donde se guarda el contenido a mostrar en el servidor web, sea propiedad del usuario **ec2-user** y del grupo **apache**.

```
sudo chown -R ec2-user:apache /var/www
```

Le cambiaremos los permisos a **/var/www** para añadir permisos de escritura de grupo y establecer el ID de grupo en los subdirectorios que se creen en el futuro.

```
sudo chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} \;
```

Este comando cambiará recursivamente los permisos de los archivos del directorio **/var/www** y sus subdirectorios para añadir permisos de escritura de grupo.

```
find /var/www -type f -exec sudo chmod 0664 {} \;
```

Con los comandos anteriores, hemos hecho que solamente los miembros del grupo **apache**, como **ec2-user**, puedan añadir, eliminar y editar archivos en la raíz de documentos de Apache para poder agregar contenido como esta página web.

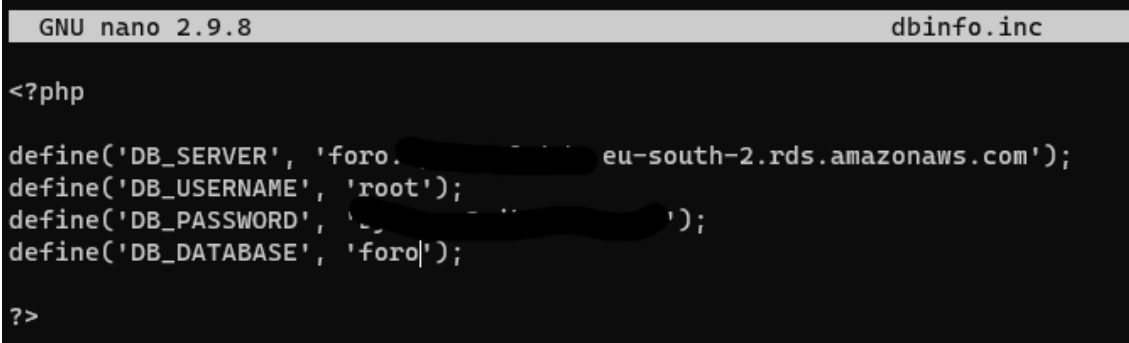
Ahora nos moveremos a **/var/www**, crearemos un directorio llamado **inc** y nos moveremos a él.

```
cd /var/www
mkdir inc
cd inc
```

Dentro de esta carpeta crearé un archivo llamado **dbinfo.inc**

```
nano dbinfo.inc
```

Aquí irá el siguiente contenido, que indica el punto de enlace de la base de datos (su IP o DNS), el nombre de usuario, su contraseña y la base de datos en cuestión.



```
GNU nano 2.9.8 dbinfo.inc

<?php

define('DB_SERVER', 'foro.██████████.eu-south-2.rds.amazonaws.com');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '██████████');
define('DB_DATABASE', 'foro');

?>
```

Para saber el punto de enlace, iremos al panel de RDS, seleccionaremos la instancia de base de datos, y allí podremos verlo.

The screenshot shows the AWS RDS console for an instance named 'foro'. The 'Conectividad y seguridad' (Connectivity and security) tab is active. Under 'Punto de enlace y puerto' (Endpoint and port), the endpoint is 'foro-2.eu-south-2.rds.amazonaws.com' and the port is '3306'. Other tabs include 'Resumen', 'Supervisión', 'Registros y eventos', 'Configuración', 'Mantenimiento y copias de seguridad', and 'Etiquetas'.

Para poder tener la página web que ya tengo programada en la instancia, voy a clonar el contenido de este mismo repositorio. Para eso en primer lugar instalaré el comando git, que me permitirá clonar repositorios pero también hacer cambios en repositorios a los que tenga acceso.

```
sudo yum install git
```

Me moveré a `/var/www/html`, donde irá el contenido que va a desplegar el servidor web.

```
cd /var/www/html
```

Una vez esté allí, clonaré el repositorio.

```
git clone https://github.com/AndreaOsma/ProyectoASIR
```

Como podemos comprobar en la siguiente imagen, al listar el contenido del directorio veremos que ahora está clonado el repositorio en nuestra instancia.

```
[ec2-user@ip-10-0-1-100 ~]$ ll
total 0
drwxrwsr-x 4 ec2-user apache 51 Dec 13 23:06 ProyectoASIR
[ec2-user@ip-10-0-1-100 ~]$ |
```

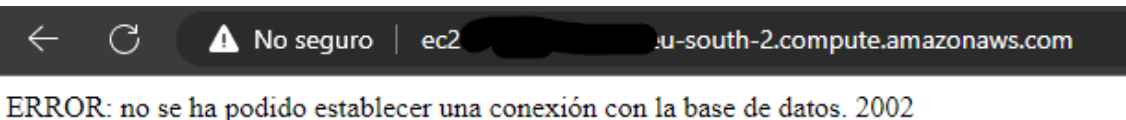
Como quiero que el contenido esté en la carpeta **/var/www/html**, moveré todo el contenido de **ProyectoASIR/proyecto/foro** al directorio actual, que es el **/var/www/html**, y una vez hecho borraré la carpeta ya que ya no la necesito más.

```
mv ProyectoASIR/proyecto/foro/* ./
rm -r ProyectoASIR/
```

Si listamos veremos que ahora estará todo el contenido en **/var/www/html**

```
[ec2-user@ip-10-0-1-100 html]$ ll
total 60
-rw-rw-r-- 1 ec2-user apache 2367 Dec 13 23:06 categoria.php
-rw-rw-r-- 1 ec2-user apache 101 Dec 13 23:06 cierre_sesion.php
-rw-rw-r-- 1 ec2-user apache 201 Dec 13 23:06 connect.php
-rw-rw-r-- 1 ec2-user apache 1475 Dec 13 23:06 crear_cat.php
-rw-rw-r-- 1 ec2-user apache 4967 Dec 13 23:06 crear_tema.php
drwxrwsr-x 2 ec2-user apache 24 Dec 13 23:06 css
-rw-rw-r-- 1 ec2-user apache 108 Dec 13 23:06 error.php
-rw-rw-r-- 1 ec2-user apache 85 Dec 13 23:06 footer.php
-rw-rw-r-- 1 ec2-user apache 1185 Dec 13 23:06 header.php
-rw-rw-r-- 1 ec2-user apache 1850 Dec 13 23:06 index.php
-rw-rw-r-- 1 ec2-user apache 3028 Dec 13 23:06 login.php
-rw-rw-r-- 1 ec2-user apache 1195 Dec 13 23:06 mensaje.php
-rw-rw-r-- 1 ec2-user apache 3085 Dec 13 23:06 registro.php
-rw-rw-r-- 1 ec2-user apache 1193 Dec 13 23:06 respuesta.php
-rw-rw-r-- 1 ec2-user apache 1606 Dec 13 23:06 tema.php
[ec2-user@ip-10-0-1-100 html]$ |
```

Si ahora vamos a la página web mediante su DNS o IP veremos que sale un error de que no se ha podido establecer una conexión con la base de datos. Ese error es el que yo he metido en el fichero **connect.php** para que muestre si no ha podido establecer una conexión con la base de datos.



The screenshot shows a web browser address bar with the URL `ec2-10-0-1-100-south-2.compute.amazonaws.com`. Below the address bar, a red error message is displayed: **ERROR: no se ha podido establecer una conexión con la base de datos. 2002**.

Para solucionarlo, dentro de **connect.php** hay que poner los mismos parámetros que en **dbinfo.inc**, es decir: el punto de enlace, el usuario, la contraseña y la base de datos.

```
<?php
$conn = mysqli_connect("foro.10-0-1-100-south-2.rds.amazonaws.com", "root", "password", "foro");

if (!$conn) {
    die("ERROR: no se ha podido establecer una conexión con la base de datos. " . mysqli_connect_errno());
}

?>
```


Añadiendo contenido a la base de datos

Ahora hay que crear la base de datos en el servidor MySQL y las tablas de usuarios, categorías, temas y posts. Como ya hemos instalado antes MariaDB, vamos a acceder al servidor mediante el siguiente comando, donde indicaremos detrás del -h el punto de enlace, detrás del -P el puerto y detrás del -u el usuario. El -p es para que nos pida la contraseña al ejecutarlo.

```
mysql -h <puntodeenlace> -P 3306 -u <usuario> -p
```

Como podemos ver en la siguiente imagen, la conexión se ha realizado con éxito.

```
[ec2-user@ip-... html]$ mysql -h foro. ... .eu-south-2.rds.amazonaws.com -P 3306 -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 23
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> |
```

No estamos dentro de ninguna base de datos, por lo que mostraremos las bases de datos que actualmente existen.

```
MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.001 sec)
```

Crearemos la base de datos foro, y mostraremos de nuevo las bases de datos existentes para comprobar que se ha creado con éxito.

```
MySQL [(none)]> create database foro
-> ;
Query OK, 1 row affected (0.005 sec)

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| foro |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.001 sec)
```

Le indicaremos a MySQL que queremos usar la base de datos **foro**.

```
MySQL [(none)]> use foro
Database changed
MySQL [foro]> |
```

Una vez dentro de la base de datos, crearemos las tablas de usuarios, categorías, temas y posts y haremos los alter table.

```
MySQL [foro]> CREATE TABLE usuarios (
->  usuariocod_usuario INT(8) NOT NULL AUTO_INCREMENT,
->  nombre_usu VARCHAR(30) NOT NULL,
->  pass_usu VARCHAR(255) NOT NULL,
->  email_usu VARCHAR(255) NOT NULL,
->  fecha_registro DATETIME NOT NULL,
->  nivel_usu INT(8) NOT NULL,
->  UNIQUE INDEX nombre_usuario_unico (nombre_usu),
->  PRIMARY KEY (cod_usuario)
-> );
Query OK, 0 rows affected, 2 warnings (0.024 sec)

MySQL [foro]> CREATE TABLE categorias (
->  cod_cat INT(8) NOT NULL AUTO_INCREMENT,
->  nombre_cat VARCHAR(255) NOT NULL,
->  descr_cat VARCHAR(255) NOT NULL,
->  UNIQUE INDEX nombre_cat_unico (nombre_cat),
->  PRIMARY KEY (cod_cat)
-> );
Query OK, 0 rows affected, 1 warning (0.029 sec)

MySQL [foro]> CREATE TABLE temas (
->  cod_tema INT(8) NOT NULL AUTO_INCREMENT,
->  descrip_tema VARCHAR(255) NOT NULL,
->  fecha_tema DATETIME NOT NULL,
->  cat_tema INT(8) NOT NULL,
->  autor_tema INT(8) NOT NULL,
->  PRIMARY KEY (cod_tema)
-> );
Query OK, 0 rows affected, 3 warnings (0.017 sec)

MySQL [foro]> CREATE TABLE posts (
->  cod_post INT(8) NOT NULL AUTO_INCREMENT,
->  texto_post TEXT NOT NULL,
->  fecha_post DATETIME NOT NULL,
->  tema_post INT(8) NOT NULL,
->  autor_post INT(8) NOT NULL,
->  PRIMARY KEY (cod_post)
-> );
Query OK, 0 rows affected, 3 warnings (0.022 sec)

ALTER TABLE posts ADD FOREIGN KEY(tema_post) REFERENCES temas(cod_tema) ON DELETE RESTRICT ON UPDATE CASCADE;Query OK, 0 rows
affected (0.045 sec)
Records: 0 Duplicates: 0 Warnings: 0

MySQL [foro]>
MySQL [foro]> ALTER TABLE temas ADD FOREIGN KEY(autor_tema) REFERENCES usuarios(cod_usuario) ON DELETE RESTRICT ON UPDATE CASC
ADE;
Query OK, 0 rows affected (0.050 sec)
Records: 0 Duplicates: 0 Warnings: 0

MySQL [foro]>
MySQL [foro]> ALTER TABLE posts ADD FOREIGN KEY(tema_post) REFERENCES temas(cod_tema) ON DELETE RESTRICT ON UPDATE CASCADE;
Query OK, 0 rows affected (0.053 sec)
Records: 0 Duplicates: 0 Warnings: 0

MySQL [foro]> |
```

Por último, mostraremos las tablas existentes para comprobar que estén creadas.

```
MySQL [foro]> show tables;
+-----+
| Tables_in_foro |
+-----+
| categorias      |
| posts           |
| temas          |
| usuarios        |
+-----+
4 rows in set (0.001 sec)
```

Cuando ya podamos acceder a la página web y hacer un registro, si queremos que un usuario tenga permisos de administrador para poder crear categorías, entraremos en la base de datos del foro y haremos un update, donde cambiaremos el nivel de usuario a 1 al usuario cuyo nombre de usuario sea el que queremos cambiar. También se puede usar el código de usuario ya que es una condición también única.

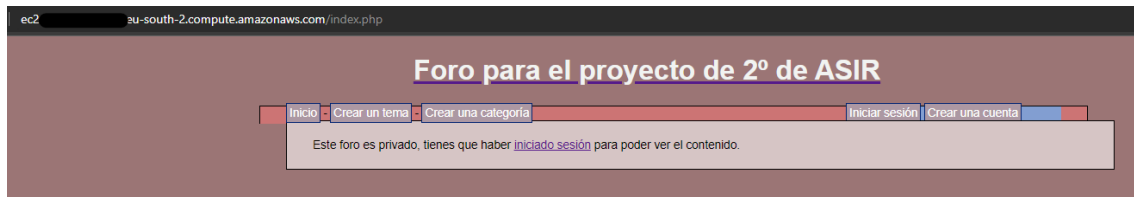
```
MySQL [foro]> select * from usuarios;
+-----+-----+-----+-----+-----+
| cod_usuario | nombre_usu | pass_usu | email_usu | fecha_registro | nivel_usu |
+-----+-----+-----+-----+-----+
| 1           | andrea     |          | andrea@andrea.es | 2022-12-13 23:37:45 | 0         |
+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)

MySQL [foro]> update usuarios set nivel_usu=1 where nombre_usu='andrea';
Query OK, 1 row affected (0.004 sec)
Rows matched: 1  Changed: 1  Warnings: 0

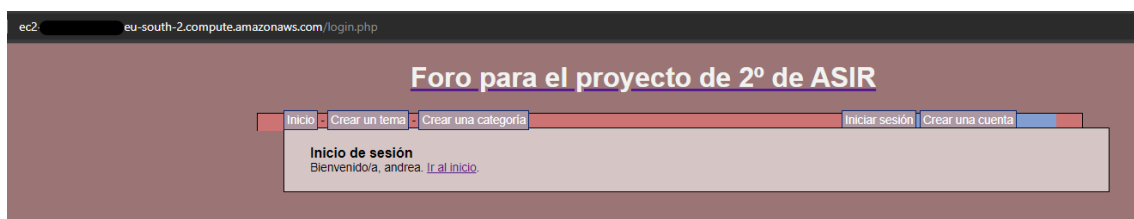
MySQL [foro]> select * from usuarios;
+-----+-----+-----+-----+-----+
| cod_usuario | nombre_usu | pass_usu | email_usu | fecha_registro | nivel_usu |
+-----+-----+-----+-----+-----+
| 1           | andrea     |          | andrea@andrea.es | 2022-12-13 23:37:45 | 1         |
+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

Prueba de funcionamiento

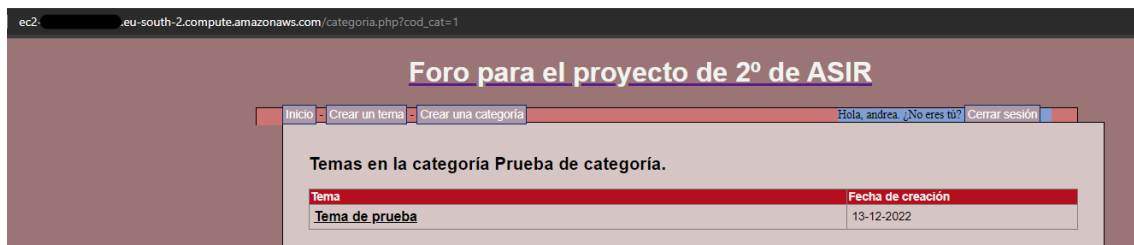
Ahora podremos acceder a la página web y ejecutarla con normalidad, como accedíamos en local.



Podremos registrar usuarios e iniciar sesión con ellos, al igual que en local.



Podremos crear categorías, temas, respuestas... y verlos desplegados al igual que en local.



Activación de HTTPS

En este punto la página web es perfectamente funcional, pero solamente se puede acceder mediante el protocolo HTTP, si intentamos acceder por HTTPS nos saldrá un error de que no se ha podido establecer la conexión. Para poder acceder también por HTTPS, vamos a activar SSL/TLS, que es un protocolo que crea un canal encriptado entre el servidor web y el cliente web para proteger los datos enviados por de poder ser interceptados por terceros. En primer lugar, nos conectaremos a la instancia y confirmaremos que Apache está activado.

```
sudo systemctl is-enabled httpd
```

Actualizaremos los repositorios:

```
sudo yum update -y
```

Instalaremos ahora el paquete **mod_ssl**, que es un módulo de Apache que añade soporte a TLS.

```
sudo yum install -y mod_ssl
```

Nos moveremos a **/etc/pki/tls/certs** y ejecutaré el fichero **make-dummy-cert** con el parámetro del archivo donde quiero que me cree el certificado SSL. El contenido del archivo será un certificado autofirmado y la clave privada del certificado.

```
cd /etc/pki/tls/certs  
sudo ./make-dummy-cert localhost.crt
```

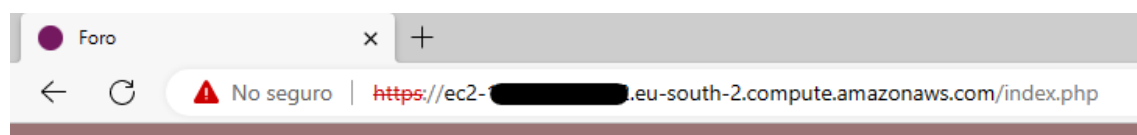
Entraremos en el archivo **/etc/httpd/conf.d/ssl.conf** con un editor de texto y nos aseguraremos de que la siguiente línea está descomentada.

```
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
```

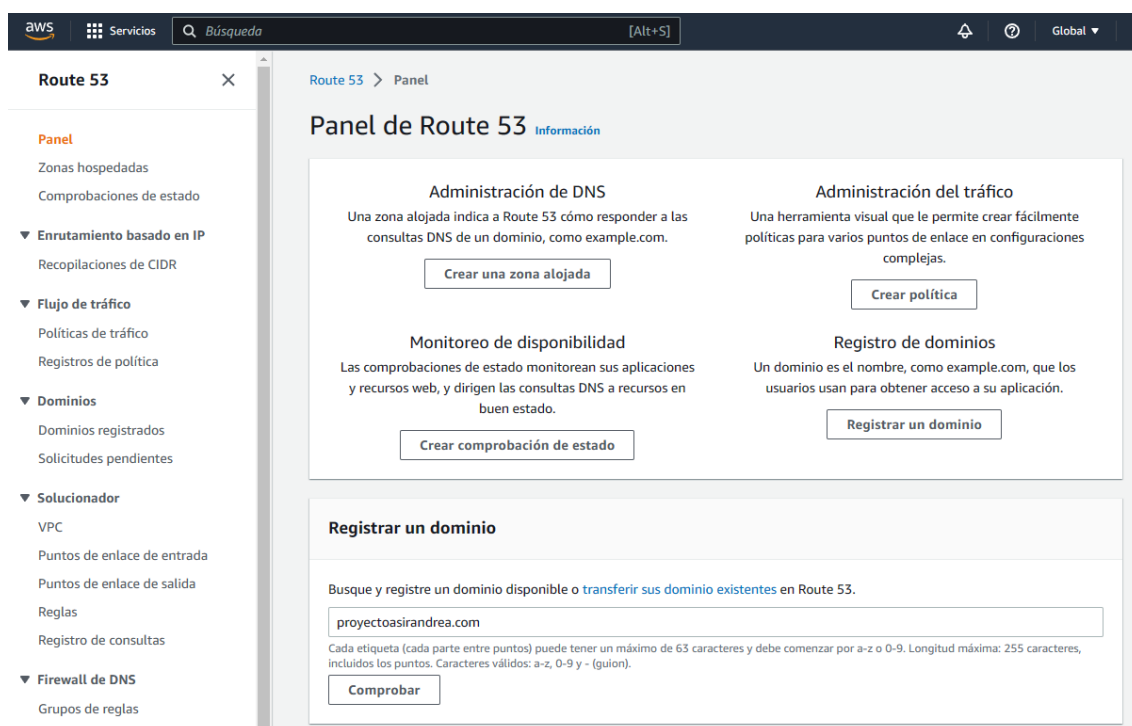
Por último, reiniciaremos Apache.

```
sudo systemctl restart httpd
```

Finalmente, podremos acceder a la página mediante el protocolo HTTPS, saldrá que no es seguro, pero se podrá acceder.



Para evitar que saliese el mensaje "No seguro", habría que crear una petición de certificación y enviarle ese archivo a una autoridad de certificación, que son empresas que comprueban ese archivo, si le dan el visto bueno te enviarían un certificado firmado y finalmente se configuraría Apache para usar ese certificado. También existe la opción de configurar un DNS, para en lugar de acceder mediante la IP del servidor o la DNS "https://ec2-*-*-*.*.eu-south-2.compute.amazonaws.com", acceder mediante una DNS legible y fácil de recordar como podría ser "proyectoasirandrea.com". Para eso tendríamos que comprar un dominio, que mismamente en AWS se puede comprar en el panel de **Route 53**, o en otras empresas que ofrecen estos servicios. Podríamos comprar el DNS que quisiéramos y estuviera disponible, con un precio que depende del subdominio (por ejemplo **.com** son 12\$ al año) y crear una zona alojada por ese dominio, la cual sería la DNS de nuestra instancia.



Yo no voy a hacer ninguna de las dos opciones, ya que los objetivos del proyecto son puramente académicos, pero existen las dos opciones. En la bibliografía dejaré más información sobre ello.

Conclusión del proyecto

Como conclusión, queda claro que mediante PHP y MySQL no es difícil hacer un foro básico, y mediante otras herramientas como AWS o cualquier otro hosting en la nube ya no es necesario hostearlo en una máquina física. Gracias a esto, aseguro que la página web siempre esté accesible, sin peligros como cortes de luz o peligro de que se comprometa el hardware.

Glosario

PHP

PHP es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web. El código PHP suele ser procesado en un servidor web por un intérprete PHP implementado como un módulo, un daemon o como un ejecutable de interfaz de entrada común (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado —que puede ser cualquier tipo de datos, como el HTML generado o datos de imágenes binarias— formaría la totalidad o parte de una respuesta HTTP.

MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.

WSL

Subsistema de Windows para Linux (WSL) es una capa de compatibilidad desarrollada por Microsoft para correr ejecutables de Linux (en formato ELF) nativamente en Windows 10 y Windows Server 2019. A partir de junio de 2019 está disponible WSL versión 2, el cual incorpora cambios importantes, como el uso de un núcleo Linux real.

Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual según la normativa RFC 2616.

MariaDB

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License). Es desarrollado por Michael (Monty) Widenius —fundador de MySQL—, la fundación MariaDB y la comunidad de desarrolladores de software libre. Introduce dos motores de almacenamiento nuevos, uno llamado Aria —que reemplaza a MyISAM— y otro llamado XtraDB —en sustitución de InnoDB—. Tiene una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, API y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente.

EC2

EC2 (Elastic Compute Cloud) es un servicio de computación bajo demanda en la plataforma de nube de AWS. Proporciona todos los servicios que ofrecería un ordenador junto a la flexibilidad de un entorno virtual. También permite a los usuarios configurar instancias y sus requisitos. Por ejemplo, la RAM o el almacenamiento que se requieran para un proyecto.

RDS

El Servicio de Bases de Datos Relacionales de Amazon (o Amazon RDS) es un servicio de base de datos relacional distribuido de Amazon Web Services (AWS). Es un servicio web que se ejecuta en la nube diseñado para simplificar la configuración, el funcionamiento y el escalado de una base de datos relacional para su uso en aplicaciones.

Bibliografía

Crear una instancia de base de datos. (s.f.). Obtenido de AWS:

https://docs.aws.amazon.com/es_es/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateDBInstance.html

Explicación: crear un servidor web y una instancia de base de datos de Amazon RDS. (s.f.).

Obtenido de AWS:

https://docs.aws.amazon.com/es_es/AmazonRDS/latest/UserGuide/TUT_WebAppWithRDS.html

Instalación de un servidor web en la instancia de EC2. (s.f.). Obtenido de AWS:

https://docs.aws.amazon.com/es_es/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateWebServer.html

Lanzamiento de una instancia de EC2. (s.f.). Obtenido de AWS:

https://docs.aws.amazon.com/es_es/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.LaunchEC2.html

MariaDB. (s.f.). Obtenido de Wikipedia: <https://es.wikipedia.org/wiki/MariaDB>

MySQL. (s.f.). Obtenido de Wikipedia: <https://es.wikipedia.org/wiki/MySQL>

Padje, E. (17 de Marzo de 2010). *How to Create a PHP/MySQL Powered Forum From Scratch.*

Obtenido de Envato Tuts+: <https://code.tutsplus.com/tutorials/how-to-create-a-phpmysql-powered-forum-from-scratch--net-10188>

PHP. (s.f.). Obtenido de Wikipedia: <https://es.wikipedia.org/wiki/PHP>

Servicio de Bases de Datos Relaciones de Amazon. (s.f.). Obtenido de Wikipedia:

https://es.wikipedia.org/wiki/Servicio_de_Bases_de_Datos_Relaciones_de_Amazon

Servidor HTTP Apache. (s.f.). Obtenido de Wikipedia:

https://es.wikipedia.org/wiki/Servidor_HTTP_Apache

Tutorial: Configure SSL/TLS on Amazon Linux 2. (s.f.). Obtenido de AWS:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/SSL-on-amazon-linux-2.html>

What is Amazon Route 53? (s.f.). Obtenido de AWS:

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html>

What is Elastic Compute Cloud (EC2)? (s.f.). Obtenido de Geeks for Geeks:

<https://www.geeksforgeeks.org/what-is-elastic-compute-cloud-ec2/>

XAMPP. (s.f.). Obtenido de Apache Friends: <https://www.apachefriends.org/es/index.html>

Licencia



Implantación de foro en AWS por [Andrea Osma Rafael](#) se distribuye bajo una [Licencia Creative Commons Atribución-NoComercial 4.0 Internacional](#).

Basada en una obra en <https://github.com/AndreaOsma/ProyectoASIR>