

Práctica 6

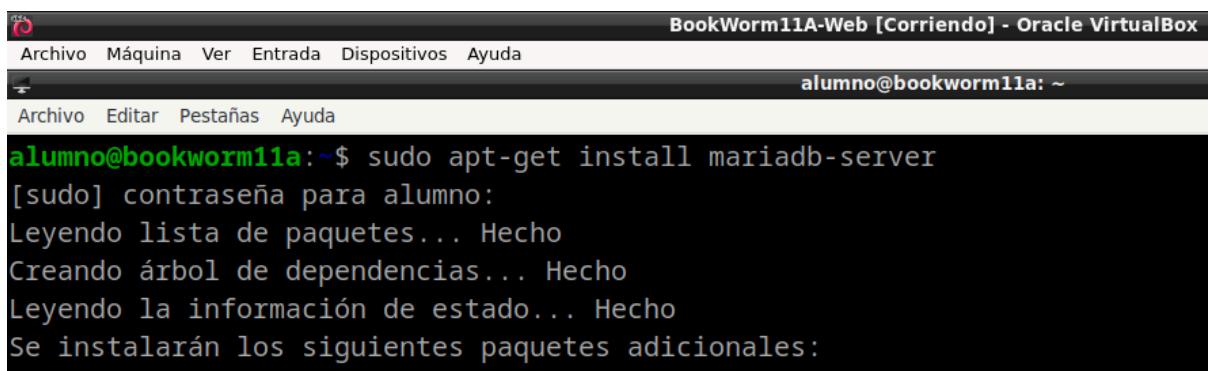
Antes de comenzar con el trabajo:

- Es conveniente haber realizado previamente el resto de prácticas
- Dispones de toda la documentación sobre apache en:
<http://httpd.apache.org/docs/current/>
- La práctica es autocontenido, es decir, todo lo nuevo que es necesario está explicado en la propia práctica pero tendrás que usar aspectos que ya hemos visto en otras prácticas
- En el caso de php y su conexión con bases de datos se utilizará 'mysqli' cuya documentación podéis consultar en:
<https://www.php.net/manual/es/class.mysql.php>

La práctica tiene como objetivo familiarizarnos con la conexión de Apache con bases de datos de MySQL. En los siguientes puntos se detallan los pasos que será necesario dar:

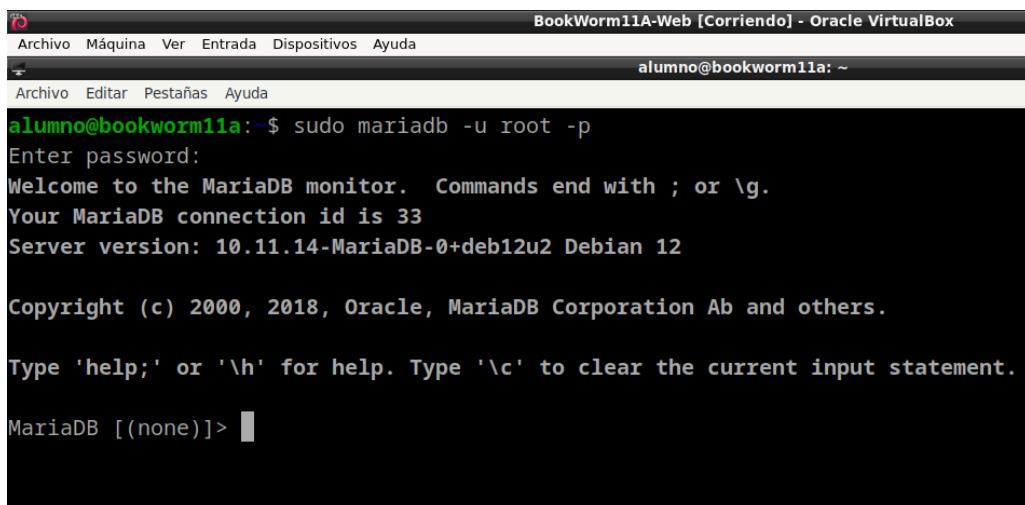
1. Instalación del Servidor de Bases de datos y prueba

- Se va a usar el servidor MySQL (MariaDB), por lo que necesitaremos instalarlo en nuestro servidor (apt-get install mariadb-server)



```
alumno@bookworm11a:~$ sudo apt-get install mariadb-server
[sudo] contraseña para alumno:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
```

- Comprobar que podemos acceder al servidor como usuario root del servidor de bases de datos (mariadb -u root -p)



```
alumno@bookworm11a:~$ sudo mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.11.14-MariaDB-0+deb12u2 Debian 12

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

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

MariaDB [(none)]> 
```

- Crea una base de datos en el servidor que contenga una tabla con atributos 'idusuario, nombreusuario, urlusuario', que se corresponderá con un número de usuario entero, un nombre de usuario, por ejemplo 'alumno' y una URL, por ejemplo 'https://virtual1.asirXX.asir/%usuario%'. Es necesario poblar la base de datos con algunos registros, entre ellos el usuario creado en la práctica anterior para la autenticación 'Basic'.

```
BookWorm11A-Web [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
alumno@bookworm11a: ~
Archivo Editar Pestañas Ayuda
MariaDB [(none)]> CREATE DATABASE bd_usuarios;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> USE bd_usuarios;
Database changed
MariaDB [bd_usuarios]> CREATE TABLE TABLA1 (
    ->     idusuario INT AUTO_INCREMENT PRIMARY KEY,
    ->     nombreusuario VARCHAR(50),
    ->     urlusuario VARCHAR(100)
    -> );
Query OK, 0 rows affected (0,014 sec)

MariaDB [bd_usuarios]> INSERT INTO TABLA1 (nombreusuario, urlusuario) VALUES ('alumno', 'https://virtual1.asir11.asir/alumno');
Query OK, 1 row affected (0,002 sec)

MariaDB [bd_usuarios]> INSERT INTO TABLA1 (nombreusuario, urlusuario) VALUES ('dorian', 'https://virtual1.asir11.asir/dorian');
Query OK, 1 row affected (0,003 sec)

MariaDB [bd_usuarios]>
```

```
BookWorm11A-Web [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
alumno@bookworm11a: ~
Archivo Editar Pestañas Ayuda
MariaDB [bd_usuarios]> SELECT * FROM TABLA1;
+-----+-----+-----+
| idusuario | nombreusuario | urlusuario |
+-----+-----+-----+
| 1 | alumno | https://virtual1.asir11.asir/alumno |
| 2 | dorian | https://virtual1.asir11.asir/dorian |
+-----+-----+-----+
2 rows in set (0,000 sec)

MariaDB [bd_usuarios]>
```

- Para que Apache pueda acceder al servidor de bases de datos y a la base de datos vamos a crear un usuario específico para tal fin (por control y seguridad), el usuario se denominará 'apache' y la contraseña será '123456'. Además, a la vez que lo creamos le concederemos permisos explícitos de lectura (y solo lectura) sobre la base de datos de usuarios que acabamos de crear. Si la base de datos se denominase 'BD_USUARIOS': 'GRANT SELECT ON BD_USUARIOS.* TO apache@localhost IDENTIFIED BY '123456';

```

BookWorm11A-Web [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
alumno@bookworm11a: ~
Archivo Editar Pestañas Ayuda
MariaDB [bd_usuarios]> GRANT SELECT ON bd_usuarios.* TO 'apache'@'localhost' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0,002 sec)

MariaDB [bd_usuarios]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB [bd_usuarios]> 
```

- Para comprobar la corrección de lo realizado nos conectaremos con el usuario creado (mariadb -u apache -p) y verificaremos que podemos seleccionar la base de datos y que podemos hacer operaciones de lectura (SELECT) pero no de escritura (DELETE, UPDATE, INSERT...)

```
BookWorm11A-Web [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
alumno@bookworm11a: ~
Archivo Editar Pestañas Ayuda
alumno@bookworm11a:~$ mariadb -u apache -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 34
Server version: 10.11.14-MariaDB-0+deb12u2 Debian 12

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

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

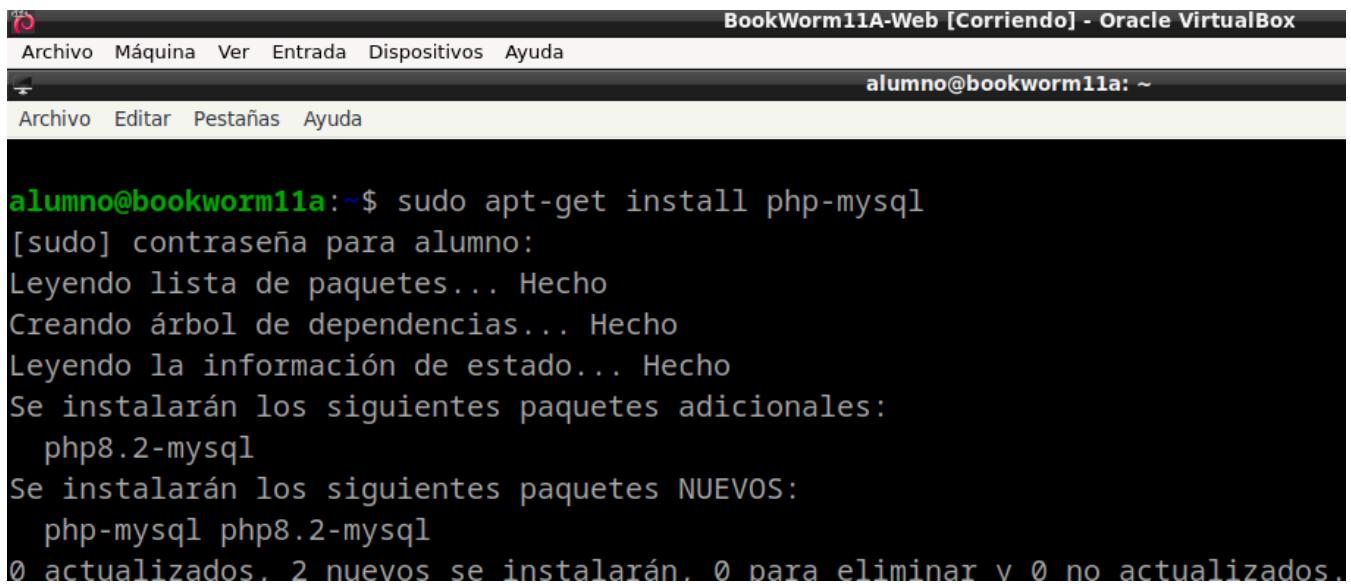
MariaDB [(none)]> USE bd_usuarios;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [bd_usuarios]> SELECT * FROM TABLA1;
+-----+-----+-----+
| idusuario | nombreusuario | urlusuario |
+-----+-----+-----+
| 1 | alumno | https://virtual1.asir11.asir/alumno |
| 2 | dorian | https://virtual1.asir11.asir/dorian |
+-----+-----+-----+
```

```
BookWorm11A-Web [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
alumno@bookworm11a: ~
Archivo Editar Pestañas Ayuda
MariaDB [bd_usuarios]> INSERT INTO TABLA1 (nombreusuario, urlusuario) VALUES ('prueba', 'https://virtual1.asir11.asir/prueba');
ERROR 1142 (42000): INSERT command denied to user 'apache'@'localhost' for table `bd_usuarios`.`TABLA1`
MariaDB [bd_usuarios]>
```

2. Instalación de la extensión PHP para MySQL

- Es necesario instalar la extensión que permite que apache se pueda conectar a bases de datos de MySQL si no está ya instalada. En caso de no estarlo ejecutaremos 'apt-get install php-mysql'



```
alumno@bookworm11a:~$ sudo apt-get install php-mysql
[sudo] contraseña para alumno:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  php8.2-mysql
Se instalarán los siguientes paquetes NUEVOS:
  php-mysql php8.2-mysql
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

3. Para comprobar que Apache se puede conectar a la base de datos y recuperar información para mostrarla al usuario tendremos que dar varios pasos. Lo recomendable es ir dando los pasos de uno en uno y chequeando el correcto funcionamiento. Para ello vamos a introducir en el archivo 'permitido.html' (que se convertirá en 'permitido.php') un conjunto de instrucciones que nos permitan comprobar que podemos conectarnos con el servidor de bases de datos, que nos podemos conectar con la base de datos en cuestión, que podemos extraer datos de dicha base de datos y por último, que podemos mostrar al usuario los registros recuperados.

- En primer lugar podemos definir en el archivo permitido.php las variables PHP que nos interesan (se usa de modo genérico y como ejemplo la base de datos 'BD_USUARIOS', la tabla 'TABLA1' y los atributos 'atributo1' y 'atributo2', de modo que solamente tendréis que cambiarlos por los vuestros)
 - \$host= 'localhost';
 - \$usuario= 'apache';
 - \$password= '123456';
 - \$bd= 'BD_USUARIOS';
 - \$consulta= 'SELECT atributo1, atributo2 FROM TABLA1';
- Conexión con el servidor: Se lanza la conexión y se comprueba con un 'if' si la conexión ha sido o no establecida y en caso de no serlo que nos muestre el error concreto que se haya producido. Aunque en un entorno de producción no se haría de este modo porque hay otros modos de hacerlo, en una práctica como ésta, de aprendizaje, nos permite hacer las pruebas sin conocimientos adicionales:

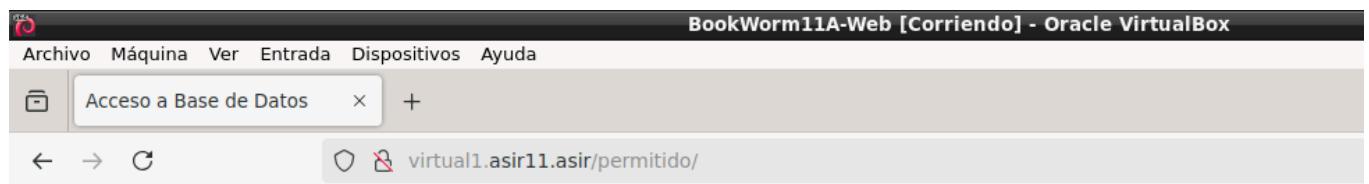
- \$conexion=
`mysqli_connect($host,$usuario,$password);`
 - if(!\$conexion) {echo'Conexión fallida con el servidor: '.mysqli_connect_error().'
';}
 - else {echo 'Conexión establecida con el servidor.
';}
 - Conexión con la base de datos: Se lanza la conexión y se comprueba con un 'if' si la conexión ha sido o no establecida y en caso de no serlo que nos muestre el error concreto que se haya producido.:
 - \$conexionBD=
`mysqli_select_db($conexion,$bd);`
 - if(!\$conexionBD) {echo'Conexión fallida a la base de datos.
';}
 - else {echo 'Conexión establecida con la base de datos.
';}
 - Consulta sobre la base de datos: Se lanza la consulta y se comprueba con un 'if' si ha tenido éxito o no y en caso de no tenerlo que nos muestre el error concreto que se haya producido.:
 - \$resultado=
`mysqli_query($conexion,$consulta);`
 - if(!\$resultado) {echo'No se ha podido obtener la consulta a la base de datos: '.mysqli_sqlstate().'
';}
 - else {echo 'Consulta ejecutada correctamente.
';}
 - Para mostrar en el navegador los resultados de la consulta se puede hacer en una tabla. Para ello podemos hacer:
 - echo 'La consulta anterior devolvió el siguiente número de registros:
`'.mysqli_num_rows($resultado).'
';`
 - while
`($fila=mysqli_fetch_array($resultado,MYSQLI_NUM)){printf("Atributo1: %d Atributo2: %s",$fila[0],$fila[1]);}`
 - echo '
';
 - Liberamos el resultado de la consulta con:
`'mysql_free_results()';` (por ejemplo).
 - Cerramos la conexión con: `mysqli_close($conexion);`
4. Lo ideal es ir anidando los if-else, de modo que si se produce un error de conexión ya no siga ejecutándose la consulta y demás, es decir, sería algo como:
- if
 - else
 - if

- else
 - if
 - else

```
alumno@bookworm11a:~$ sudo mv /var/www/virtual1/permitido/permitido.html /var/www/virtual1/permitido/permitido.php
alumno@bookworm11a:~$
```

```
GNU nano 7.2 /var/www/virtual1/permitido/permitido.php *
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Acceso a Base de Datos</title>
</head>
<body>
    <h1>Prueba de conexión MySQL (MariaDB)</h1>
    <ul>
        <?php
            // 1. Definición de variables [cite: 112-116]
            $host = 'localhost';
            $usuario = 'apache';
            $password = '123456';
            $bd = 'bd_usuarios';
            $consulta_sql = 'SELECT nombreusuario, urlusuario FROM TABLA1';
```

```
GNU nano 7.2 /etc/apache2/sites-available/virtual1.conf
    Require all denied
</Directory>
<Directory /var/www/virtual1/permitido>
    #Options -Indexes
    DirectoryIndex permitido.php
</Directory>
```



Prueba de conexión MySQL (MariaDB)

- Conexión establecida con el servidor correctamente.
- Base de datos seleccionada correctamente.
- Consulta ejecutada correctamente.
- La consulta devolvió 2 registros:

Usuario	URL
alumno	https://virtual1.asir11.asir/alumno
dorian	https://virtual1.asir11.asir/dorian

5. Disponéis de un documento PDF en que se pueden ver algunos resultados que podríais obtener incluyendo algunos errores

Recuerda hacer todas las capturas que consideres necesarias

Comando ver logs:

- Ver las últimas 20 líneas:

```
sudo tail -n 20 /var/log/apache2/error.log
```

- Ver los errores EN VIVO

```
sudo tail -f /var/log/apache2/error.log
```

- Para ver visitas (Accesos)

```
sudo tail -f /var/log/apache2/access.log
```

6.2. Añade a la tabla que estás usando para extraer los datos, un atributo adicional de tipo texto denominado huella en que aparezca la password del usuario codificada (te puedes inventar la codificación o usar alguno que conozcas). Conéctate desde el cliente LinuxXX y captura el resultado:

1. **Entra en MariaDB como root (desde tu servidor bookworm11a):**

```
sudo mariadb -u root -p
```

2. Selecciona tu base de datos:

```
USE BD_USUARIOS;
```

3. Añade la nueva columna llamada huella:

```
ALTER TABLE TABLA1 ADD huella VARCHAR(100);
```

4. Actualiza los usuarios

```
UPDATE TABLA1 SET huella = SHA1('contraseña_alumno') WHERE nombreusuario = 'alumno';
```

```
UPDATE TABLA1 SET huella = SHA1('contraseña_maje') WHERE nombreusuario = 'maje';
```

Parte 2: Modificar el Script PHP (`permitido.php`)

1. Edita el archivo:

```
sudo nano /var/www/virtual1/permitido/permitido.php
```

2. Modifica la consulta SQL:

```
$consulta_sql = 'SELECT nombreusuario, urlusuario, huella FROM TABLA1';
```

3. Modifica el encabezado de la tabla HTML: Busca donde haces el `echo` de la tabla `<tr><th>...` y añade una columna más.

```
echo '<table border="1"><tr><th>Usuario</th><th>URL</th><th>Huella (Pass Codificada)</th></tr>';
```

4. Modifica el bucle que pinta los datos: Busca la línea del `printf` dentro del `while` y añade la variable nueva.

```
while ($fila = mysqli_fetch_array($resultado)) {  
    printf("<tr><td>%s</td><td>%s</td><td>%s</td></tr>", $fila['nombreusuario'],  
    $fila['urlusuario'], $fila['huella']);  
}
```