

# **TRABAJO FINAL**

## **2ª EVALUACIÓN**

### **DAW**

**AUTOR: JOSÉ MANUEL PÉREZ MÁRQUEZ**

**I.E.S. RUIZ GIJÓN (UTRERA)**

# ÍNDICE

<b>PARTE A:</b> .....	<b>3</b>
<b>nginx</b> .....	<b>3</b>
<b>mysql</b> .....	<b>3</b>
<b>php</b> .....	<b>4 - 6</b>
<b>conexión mysql</b> .....	<b>6- 9</b>

**PARTE B:**

## PARTE A:

```
usuario@hobbit:/etc/nginx/sites-available$ sudo ufw allow 'Nginx HTTP'  
Skipping adding existing rule  
Skipping adding existing rule (v6)
```

Verificando que el cortafuegos ha añadido la regla:

```
sudo apt install mysql-server
```

Instalación de mysqlserver.

```
sudo apt install php-fpm php-mysql
```

Instalación de php-fpm.

```
sudo mkdir /var/www/josemanuel.net
```

Creo un dominio con mi nombre.

```
sudo chown -R $USER:$USER /var/www/josemanuel.net
```

Cambio los permisos de propietario sobre el archivo josemanuel.net.

```
usuario@hobbit: /etc/nginx/sites-available
GNU nano 5.2 josemanuel.net
server {
    listen 80;
    server_name josemanuel.net www.josemanuel.net;
    root /var/www/josemanuel.net;

    index index.html index.htm index.php;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
    }

    location ~ /\.ht {
        deny all;
    }
}
```

Modifico el archivo creado anteriormente para que quede de la siguiente forma. Añado el puerto, en este caso el puerto 80.

```
sudo ln -s /etc/nginx/sites-available/josemanuel.net
/etc/nginx/sites-enabled/josemanuel.net
```

```
usuario@hobbit:/etc/nginx/sites-enabled$ ls -la
total 8
drwxr-xr-x 2 root root 4096 Feb 19 19:12 .
drwxr-xr-x 8 root root 4096 Feb 19 17:50 ..
lrwxrwxrwx 1 root root 41 Feb 19 19:12 josemanuel.net -> /etc/nginx/sites-available/josemanuel.net
usuario@hobbit:/etc/nginx/sites-enabled$
```

Creamos un enlace simbólico al archivo **josemanuel.net** en la carpeta **/etc/nginx/sites-enabled/** para la gestión de los futuros archivos.

Para comprobar que no ha sucedido ningún error al crear el enlace simbólico se usa el comando:

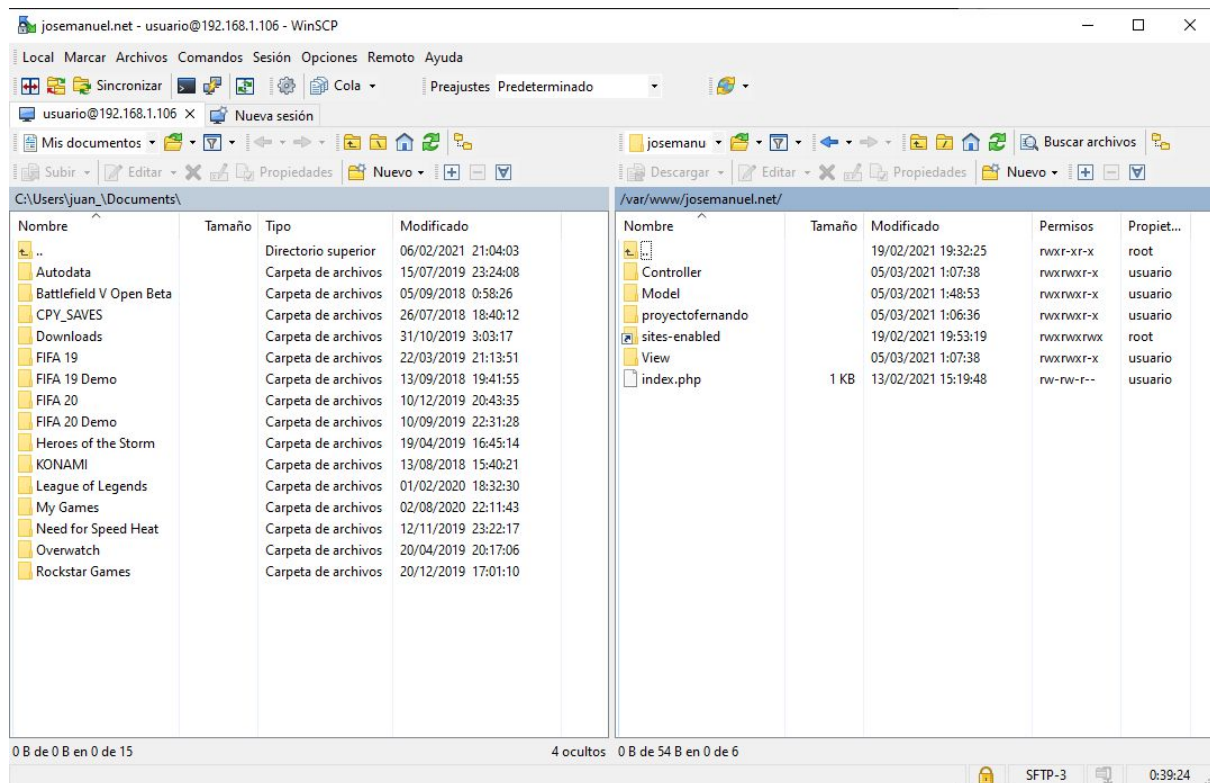
```
sudo nginx -t
```

Para aplicar los cambios realizados, se utiliza el siguiente comando:

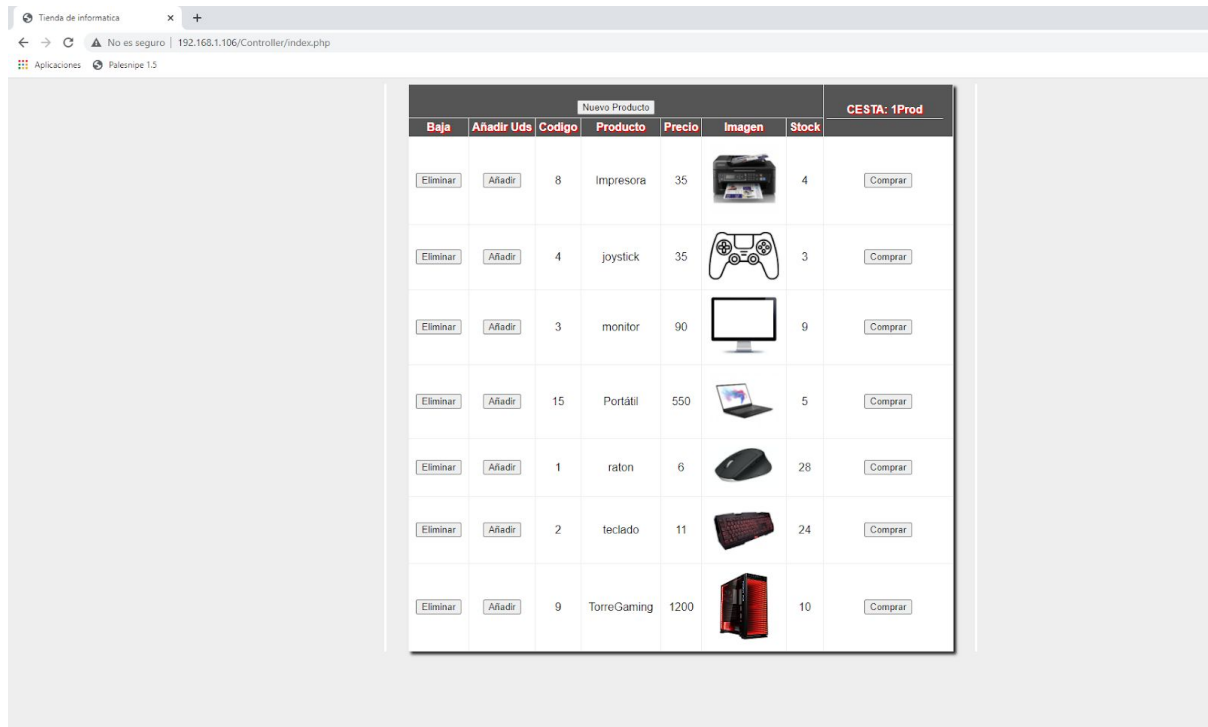
```
sudo systemctl reload nginx
```

```
usuario@hobbit:/var/www/josemanuel.net$ git clone https://gitlab.com/planetLinux/proyectofernando.git
Cloning into 'proyectofernando'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (32/32), done.
remote:
mysql> ng objects: 100% (32/32), 735.81 KiB | 286.00 KiB/s, done.
mysql> @hobbit:/var/www/josemanuel.net$ ls
mysql> ler Model View index.php proyectofernando sites-enabled
```

-Mediante el comando **git clone** copiamos el repositorio de gitlab del ejercicio de la tienda con base de datos de Fernando.



-También podríamos conectarnos con el programa WinSCP metiendo nuestra ip, nombre y contraseña de la máquina virtual y copiar el repositorio arrastrando los archivos en **/var/www/josemanuel.net/**



Para comprobar que todo funciona correctamente, introduce la ip de tu máquina **ip a** en el navegador y aparecerá el ejercicio que hemos clonado del repositorio.

```
sudo mysql
CREATE DATABASE usuarioDB;
CREATE USER 'example_user'@'%' IDENTIFIED WITH mysql_native_password BY
'password';
```

Comprobamos el funcionamiento del Sistema Gestor de Base de Datos (SGBD). Creamos un usuario llamado usuarioDB, que utiliza mysql\_native\_password como método de autenticación predeterminado.

Para dar permisos al usuario (usuarioDB) a la base de datos se utiliza:

```
GRANT ALL ON example_database.* TO 'usuarioDB'@'%';
```

El siguiente paso sería loguearse correctamente con nuestro usuario a mysql:

```
mysql -u example_user -p
```

```
mysql> CREATE DATABASE IF NOT EXISTS `tienda` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
Query OK, 1 row affected (0.07 sec)

mysql> USE `tienda`;
Database changed
mysql> CREATE TABLE `catalogo` (
  -> `codigo` varchar(20) COLLATE utf8_bin NOT NULL,
  -> `nombre` varchar(100) COLLATE utf8_bin NOT NULL,
  -> `precio` decimal(10,2) NOT NULL,
  -> `imagen` varchar(100) COLLATE utf8_bin NOT NULL,
  -> `detalle` varchar(500) COLLATE utf8_bin NOT NULL
  -> ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
Query OK, 0 rows affected, 6 warnings (0.32 sec)

mysql> INSERT INTO `catalogo` (`codigo`, `nombre`, `precio`, `imagen`, `detalle`) VALUES
  -> ('parkduo', 'Parker Duofold International', '406.00', 'parkerduo.jpg', 'Plumín de oro 18K. Fabricada en Reino Unido. Cuerpo en resina de alta resistencia.'),
  -> ('pelii1000', 'Pelikan Souvëran M-1000', '545.00', 'pelikan.jpg', 'Plumín de oro 18K. Émbolo de bronce. Fabricación alemana. Excelentes acabados.'),
  -> ('viscvan', 'Visconti Van Gogh', '180.00', 'visconti.jpg', 'Diseño y acabados de lujo. Cuerpo fabricado en Italia. Plumín alemán.'),
  -> ('waterexp', 'Waterman Expert', '103.55', 'waterman.jpg', 'Excelente pluma de uso diario. Fabricada en Francia. Plumín de acero.').
Query OK, 4 rows affected (0.08 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> CREATE TABLE `clientes` (
  -> `nombre` varchar(30) NOT NULL,
  -> `token` varchar(10) NOT NULL,
  -> `peticiones` int(11) NOT NULL
  -> ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
Query OK, 0 rows affected, 1 warning (0.12 sec)

mysql> INSERT INTO `clientes` (`nombre`, `token`, `peticiones`) VALUES
  -> ('Ana', '321cba', 0),
  -> ('Pepe', '123abc', 0);
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> CREATE TABLE `productos` (
  -> `codigo` int(11) NOT NULL,
  -> `nombre` varchar(30) NOT NULL,
  -> `precio` int(11) NOT NULL,
  -> `imagen` varchar(30) NOT NULL,
  -> `stock` int(11) NOT NULL,
  -> `descripcion` varchar(50) NOT NULL
  -> ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
Query OK, 0 rows affected, 3 warnings (14.12 sec)
```

```
mysql>
mysql> INSERT INTO `productos` (`codigo`, `nombre`, `precio`, `imagen`, `stock`, `descripcion`) VALUES
  -> (1, 'raton', 6, 'raton.png', 28, 'Raton de la marca logitech de muy buena calidad'),
  -> (2, 'teclado', 11, 'teclado.png', 25, 'Teclado gaming de tacto mecánico muy resistente'),
  -> (3, 'monitor', 90, 'monitor.png', 9, 'Monitor de 24 pulgadas 1ms y 75hz con panel LCD'),
  -> (4, 'joystick', 35, 'joystick.png', 3, 'Joystick compatible con ps4 y PC con vibración'),
  -> (8, 'impresora', 35, 'impresora.jpg', 4, 'Impresora de inyección tinta'),
  -> (9, 'TorreGaming', 1200, 'torregaming.jpg', 10, 'Torre adaptada para videojuegos de alto rendimiento'),
  -> (15, 'Portátil', 550, 'portatil.jpg', 5, '');
Query OK, 7 rows affected (0.21 sec)
Records: 7 Duplicates: 0 Warnings: 0

mysql>
mysql> ALTER TABLE `catalogo`
  -> ADD PRIMARY KEY (`codigo`);
Query OK, 0 rows affected (0.17 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE `clientes`
  -> ADD PRIMARY KEY (`nombre`);
Query OK, 0 rows affected (0.22 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE `
  -> ADD PRIMARY KEY (`codigo`);
  -> K, 0 rows affected (0.28 sec)
  -> : 0 Duplicates: 0 Warnings: 0
  ->
  -> ALTER TABLE `productos`
  -> MODIFY `codigo` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=18;
  ->
  -> K, 7 rows affected, 1 warning (28.24 sec)
  -> : 7 Duplicates: 0 Warnings: 1
  ->
  -> COMMIT;
  -> K, 0 rows affected (0.00 sec)
```

En las últimas dos imágenes estoy realizando la creación de la base de datos y de las tablas y la inserción de los campos de cada tabla con su respectivo contenido que voy a utilizar en el ejercicio.

```
GNU nano 5.2 TiendaDB.php
<?php
abstract class TiendaDB {
private static $server = 'localhost';
private static $db = 'tienda';
private static $user = 'usuario';
private static $password = 'Usuario_1';
public static function connectDB() {
try {
$conection = new PDO("mysql:host=".self::$server.";dbname=".self::$db.";charset=utf8", self::$user, self::$password);
} catch (PDOException $e) {
echo "No se ha podido establecer conexión con el servidor de bases de datos.<br>";
die ("Error: " . $e->getMessage());
}
return $conection;
}
}
```

Para que el ejercicio funcione correctamente con la base de datos, necesito modificar el archivo que conecta ambas cosas, como la contraseña, el usuario o el nombre de la base de datos.

```
mysql> show databases;
+-----+
| Database |
+-----+
| example_database |
| information_schema |
| mibasededatos |
| mysql |
| performance_schema |
| sys |
| tienda |
+-----+
7 rows in set (0.00 sec)
```

Utilizando el comando `sudo mysql` podemos conectarnos a la base de datos y realizar consultas como la de la foto de arriba que muestra todas las bases de datos creadas, y la tienda que es la que vamos a utilizar en el ejercicio.



```
mysql> show full tables FROM tienda;
+-----+-----+
| Tables_in_tienda | Table_type |
+-----+-----+
| catalogo          | BASE TABLE |
| clientes           | BASE TABLE |
| productos          | BASE TABLE |
+-----+-----+
3 rows in set (0.00 sec)
```

También podemos mostrar el contenido de la base de datos (tienda) para ver sus tablas y el contenido de dichas tablas.

## **PARTE B:**