



Universidad de Jaén

Facultad de Ciencias Sociales y Jurídicas

Trabajo Fin de Grado

CREACIÓN Y GESTIÓN DE UNA BASE DE DATOS CON MYSQL Y PHPMYADMIN

Alumno: José María Lozano Banqueri

Junio, 2018

ÍNDICE DE CONTENIDOS

RESUMEN	5
ABSTRACT	5
1. INTRODUCCIÓN.....	6
2. UNA BREVE HISTORIA DE LAS BASES DE DATOS	6
3. QUÉ GESTOR DE BASES DE DATOS UTILIZAR	8
3.1. MySQL	9
3.2 Microsoft SQL Server.....	10
3.3 Oracle	11
3.5 DB2	12
4. INSTALACIÓN DEL SERVIDOR APACHE, MYSQL Y EL GESTOR PHPMYADMIN	13
4.1. ¿QUÉ ES PHP?.....	13
4.2. ¿QUÉ ES APACHE?	13
4.3. ¿QUÉ ES phpMyAdmin?	14
4.4. INSTALACIÓN DEL PAQUETE WAMP	14
5. INTRODUCCIÓN AL LENGUAJE SQL	16
5.1 COMANDOS SQL BÁSICOS: DEFINIENDO COMO SE ALMACENAN LOS DATOS.	17
5.2 COMANDOS SQL BÁSICOS: MANIPULACIÓN DE LOS DATOS.	17
6. CREACIÓN DE UNA BASE DE DATOS	18
6.1. CREACIÓN DE LAS TABLAS DENTRO DE LA BASE DE DATOS	19
7. GESTIÓN DE DATOS EN LAS TABLAS	23
7.1. INTRODUCCIÓN A LA GESTIÓN DE DATOS MEDIANTE CONSULTAS.....	23
7.1.1. INTRODUCCIÓN DE DATOS	23
7.1.2. MODIFICACIÓN DE DATOS	25

7.1.3 VISUALIZACIÓN Y SELECCIÓN DE DATOS.....	26
7.1.4. ELIMINACIÓN DE DATOS	27
7.1.5. COMBINANDO COMANDOS EN LA MISMA CONSULTA PARA REGISTRAR LAS COMPRAS	29
7.2 INTRODUCCIÓN A LA GESTIÓN DE DATOS MEDIANTE LA INTERFAZ DE phpMyAdmin.....	29
7.2.1. INTRODUCCIÓN DE DATOS	31
7.2.2. MODIFICACIÓN DE DATOS	31
7.2.3. BÚSQUEDA DE DATOS.....	32
7.2.4. ALGUNAS OPERACIONES DE GRAN UTILIDAD EN phpMyAdmin.....	33
8. PLANES DE MANTENIMIENTO EN MYSQL	34
8.1 Copias de seguridad	34
8.2 Limpieza de logs, optimización de las tablas y vaciamiento del cache de las consultas	36
9. CONCLUSIONES.....	37
10. BIBLIOGRAFÍA.....	38

ÍNDICE DE FIGURAS

Figura 1: Sistemas de gestión de bases de datos más populares y utilizados.....	8
Figura 2: Instalación del servidor Apache, MySQL y el gestor phpMyAdmin (Paso 1).....	14
Figura 3: Instalación del servidor Apache, MySQL y el gestor phpMyAdmin (Paso 2).....	15
Figura 4: Instalación del servidor Apache, MySQL y el gestor phpMyAdmin (Paso 3).....	16
Figura 5: Partes de phpMyAdmin en el directorio raíz del servidor.....	18
Figura 6: Éxito en la creación de la base de datos.	19
Figura 7: Diseño de la tabla ‘clientes’ en SQL	21
Figura 8: Diseño de la tabla ‘productos’ en SQL.....	22
Figura 9: Diseño de la tabla ‘historial_compras’ en SQL.....	23
Figura 10: Introducción de datos mediante una consulta.....	24
Figura 11: Introducción de datos mediante una consulta múltiple.....	24

Figura 12: Modificación de datos mediante una consulta SQL	26
Figura 13: Visualización y selección de datos mediante una consulta	26
Figura 14: Resultados de la consulta con el comando SELECT	27
Figura 15: Eliminación de datos mediante una consulta.....	28
Figura 16: Combinación de comandos en una consulta.....	29
Figura 17: Distintos elementos de phpMyAdmin	30
Figura 18: Opción insertar en phpMyAdmin	31
Figura 19: Modificación de datos utilizando la interfaz phpMyAdmin.....	32
Figura 20: Operadores SQL dentro de la interfaz phpMyAdmin.....	33
Figura 21: Operaciones útiles de phpMyAdmin	34
Figura 22: Opción exportar en phpMyAdmin.....	35
Figura 23: Opción importar en phpMyAdmin	35
Figura 24: Opciones de mantenimiento en phpMyAdmin	36

ÍNDICE DE TABLAS

Tabla 1: Datos de los productos	25
Tabla 2: Operadores SQL.....	32

RESUMEN

En este trabajo nos centraremos en la creación de una base de datos utilizando un servidor MySQL, su gestión mediante phpMyAdmin y sus tareas de mantenimiento.

Por otro lado, veremos las ventajas que nos ofrece MySQL frente a otros tipos de bases de datos, el lenguaje SQL y los comandos más útiles enfocados a una tienda virtual.

El objetivo final del trabajo es que el lector aprenda los conceptos básicos de las bases de datos MySQL y del propio lenguaje SQL, así como que aprenda a crear y gestionar una base de datos en un sitio virtual.

Palabras clave: base de datos, tabla, MySQL, SQL, phpMyAdmin, consulta, servidor.

ABSTRACT

In this project we will focus on creating a database using a MySQL server, its management by using PhpMyAdmin and the maintenance tasks.

On the other side, we will see the advantages that MySQL offers compared other kind of databases, the SQL language and the most useful commands focused on a virtual site.

The final objective of this project is that the reader learn the basic concepts of MySQL databases and the SQL language so as learn to create and manage a database on a virtual site.

Keywords: database, table, MySQL, SQL, phpMyAdmin, query, server.

1. INTRODUCCIÓN

Como primer contacto al concepto de base de datos se podría decir que es un conjunto de datos que se encuentran estructurados en diferentes categorías o clases con un orden preestablecido.

En informática cuando nos referimos a bases de datos nos referimos a información almacenada en discos físicos y podemos acceder a ellos de forma local en nuestro ordenador o de forma local mediante una red local o internet con algún gestor de bases de datos.

Es aquí donde podemos introducir el término sistema de gestión de bases de datos, que lo podemos definir como un conjunto de software que permiten la manipulación de las bases de datos de forma que el usuario pueda agregar, eliminar, ordenar o modificar la información de la misma.

2. UNA BREVE HISTORIA DE LAS BASES DE DATOS

Se escuchó por primera vez el término base de datos en una reunión de expertos que se celebró en 1963 en la ciudad de California.

Pero para entender realmente de dónde provienen las bases de datos debemos mirar su historia, las bases de datos se pueden encontrar desde la antigüedad donde ya existían bibliotecas y toda clase de registros sin embargo había que invertir mucho tiempo en la búsqueda de la información mayormente cuando se trataba de grandes cúmulos de datos puesto que no contaban con máquinas que pudieran reemplazar el trabajo humano

En 1889 Herman Hollerith creó la máquina tabuladora la cual fue utilizada en el censo de los Estados Unidos de 1890 con unos buenos resultados.

En 1950 se comenzó a utilizar las cintas magnéticas para automatizar la información y hacer respaldos, la principal desventaja de este sistema era que sólo se podía leer de manera secuencial.

En 1960 la bajada de precios de los ordenadores y con ello la adquisición de los mismos por las compañías privadas hizo que los discos duros reemplazaran el uso de las cintas magnéticas

permitiendo que la información se pudiera consultar directamente sin saber la ubicación exacta de los datos, en esta época se introducen las bases de datos jerárquicas y las bases de datos de red ya que ahora era posible guardar información en listas y árboles, uno de los principales logros de esta década fue la alianza de IBM y American Airlines para crear SABRE, un sistema operativo que manipulaba la información de los pasajeros de la compañía aérea American Airlines.

Más tarde, Charles Bachman desarrolló el IDS (Integrated Data Store) que introdujo el modelo en red de base de datos que permitió la estandarización de un lenguaje de programación.

En 1970, Edgar Frank Codd, un científico informático de origen inglés gracias a su artículo “Un modelo relacional de datos para grandes bancos de datos compartidos” se dio paso a la creación de lo que sería la segunda generación de los sistemas de gestión de bases de datos.

En esta década también se creó ORACLE, uno de los sistemas más competentes que existen actualmente junto a SQL Server o MySQL.

En 1980 SQL comenzó estandarizarse por el auge de los sistemas relaciones de bases de datos con su sistema de tablas que estaban compuestas de filas y columnas derrotando a las bases de datos jerárquicas y de red por simplificar el trabajo de programación y aumentar la claridad al manipularlas.

Desde 1990 hasta la actualidad la investigación en bases de datos se centró en las bases de datos orientadas a objetos que dieron paso a la creación de programas como Excel y Access.

3. QUÉ GESTOR DE BASES DE DATOS UTILIZAR

La elección del sistema de gestión de bases de datos depende de las necesidades del proyecto y, además, de factores subjetivos como la propia experiencia del administrador o programador.

Desde mi propia experiencia siempre he trabajado con MySQL por su facilidad de instalación y configuración, promueve el software libre puesto que tiene licencia GNU GPL y además es poseedor de una gran potencia para manipular datos a cambio de pocos recursos.

“La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente sus siglas del inglés GNU GPL, es una licencia creada por la Free Software Foundation en 1989”¹

No obstante, a continuación, veremos los sistemas de gestión de bases de datos SQL más utilizados (Figura 1) y varias de sus características más destacables, así como sus principales ventajas y desventajas.

Figura 1: Sistemas de gestión de bases de datos más populares y utilizados.

Rank			DBMS	Database Model	Score		
Jul 2018	Jun 2018	Jul 2017			Jul 2018	Jun 2018	Jul 2017
1.	1.	1.	Oracle +	Relational DBMS	1277.79	-33.47	-97.09
2.	2.	2.	MySQL +	Relational DBMS	1196.07	-37.62	-153.04
3.	3.	3.	Microsoft SQL Server +	Relational DBMS	1053.41	-34.32	-172.59
4.	4.	4.	PostgreSQL +	Relational DBMS	405.81	-4.86	+36.37
5.	5.	5.	MongoDB +	Document store	350.33	+6.54	+17.56
6.	6.	6.	DB2 +	Relational DBMS	186.20	+0.56	-5.05

Fuente: DB-Engines ranking (2018), Disponible on-line: <https://db-engines.com/en/ranking/>.

¹ ¿Quiénes somos? WampServer, Disponible on-line: <http://www.wampserver.es/#about/>.

3.1. MySQL



MySQL es uno de los sistemas de gestión de bases de datos más populares para aplicaciones basadas en la web. Es gratuito y además está constantemente actualizado con nuevas características y medidas de seguridad. Existen una variedad de ediciones de pago destinadas al uso comercial pero la versión gratuita ofrece más velocidad y seguridad al no llevar incluidas una gran cantidad de características, esto puede ser positivo o negativo depende de las necesidades del administrador.

Este sistema de bases de datos permite seleccionar y manejar datos de una gran cantidad diferente de tipos de tablas, con una seguridad excelente y una gran fama de no desperdiciar los recursos del servidor.

Sus principales ventajas son:

- Es gratuito.
- Ofrece una gran cantidad de características incluso en su versión gratuita.
- Existen multitud de interfaces de usuario y una gran comunidad detrás.
- Es compatible con otras bases de datos como Oracle.

Sus principales desventajas son:

- No existe soporte gratuito oficial para la versión gratuita.
- Puedes necesitar tiempo para automatizar funciones que otros sistemas llevan ya incluidos.

Ideal para desarrolladores independientes que necesitan un sistema robusto y un presupuesto muy limitado.

3.2 Microsoft SQL Server



Como con cualquier otro sistema de gestión de bases de datos popular, puedes elegir entre multitud de versiones de este sistema. Puede funcionar tanto de forma local como en servidores basados en la nube e inclusive de ambas formas a la vez. Está también disponible en sistemas Linux desde el lanzamiento de la versión de Microsoft SQL Server 2016.

Algunas de las características que incluye este sistema es el soporte de la información temporal, que hace posible rastrear los cambios que se han efectuado a los datos a lo largo del tiempo. La última versión de Microsoft SQL Server también incluye medidas de seguridad como los permisos de usuario, donde solo determinados usuarios con privilegios podrán acceder a la información más importante.

Sus principales ventajas son:

- Es rápido y estable.
- Ofrece la posibilidad de ver gráficas y datos del rendimiento de los recursos utilizados.
- Incluye soporte para los dispositivos móviles.
- Es totalmente compatible con los paquetes y programas de Microsoft.

Sus principales desventajas son:

- El precio puede ser elevado para muchas organizaciones.
- Puede llegar a consumir demasiados recursos, necesitando máquinas muy potentes.
- Muchos usuarios han notificado problemas a la hora de hacer tareas de mantenimiento.

Ideal para organizaciones grandes que utilizan muchos productos de la compañía Microsoft.

3.3 Oracle



No es una sorpresa que Oracle ostente la medalla de gestor de bases de datos más utilizados debido a la historia de las mismas. Las últimas versiones de Oracle incluso permiten la gestión de bases de datos en servidores múltiples permitiendo fácilmente sostener miles de millones de datos al mismo tiempo.

Sus principales ventajas son:

- Innovación, generalmente Oracle va un paso más allá de todos los demás sistemas.
- Es increíblemente seguro, robusto y posee una gigantesca lista de características disponibles.

Sus principales desventajas son:

- Tiene un coste prohibitivo para pequeñas empresas y usuarios individuales.
- Este sistema requiere grandes cantidades de recursos para funcionar correctamente.

Ideal para organizaciones muy grandes que necesitan manejar gigantescas bases de datos y una gran cantidad de características.

3.4 PostgreSQL



Comúnmente usado para las bases de datos en sitios web. Fue uno de los primeros sistemas de gestión de bases de datos en ser desarrollado y permite a los usuarios gestionar tanto datos estructurados como no estructurados. Puede ser usado en la mayoría de plataformas incluyendo los sistemas basados en Linux y, además, es muy sencillo importar bases de datos de otros sistemas.

Puede ser utilizado en sistemas basados en la nube y en local y tiene una gran comunidad detrás que crece día a día.

Sus principales ventajas son:

- Puede soportar terabytes de información.
- Soporta JSON.
- Incluye una gran cantidad de funciones predefinidas.
- Tiene multitud de interfaces disponibles.

Las principales desventajas son:

- La documentación e información de este sistema no está muy clara por lo que para ejecutar funciones avanzadas el usuario promedio puede tener grandes dificultades.
- La configuración puede hacerse tediosa y complicada.
- No es muy veloz a la hora de hacer consultas muy grandes.

Es ideal para organizaciones con un presupuesto limitado que necesitan interfaces de usuario simples y usar JSON.

3.5 DB2



Creado por la compañía IBM, no utiliza las funciones SQL y puede leer los archivos XML y JSON. Fue diseñado para funcionar en los servidores iSeries de IBM, pero existen versiones que funcionan en Windows, Linux y Unix.

La última versión de DB2 incluye una larga lista de nuevas características, pero sobre todo destaca la mejora en su velocidad y en el uso de memoria.

Sus principales ventajas son:

- Uso eficaz de los recursos del sistema.
- Puede correr varias tareas al mismo tiempo.
- Los códigos de error son muy descriptivos y permiten ahorrar tiempo a la hora de solucionar problemas.

Sus principales desventajas son:

- Alto coste monetario.

- Se requiere de software de terceros para ejecutar algunas funciones.
- El soporte básico solo está disponible durante tres años, después, hay que pagar.

Es ideal para organizaciones grandes que no tienen muchos recursos disponibles y necesitan manejar grandes bases de datos.

4. INSTALACIÓN DEL SERVIDOR APACHE, MySQL Y EL GESTOR phpMyAdmin

Para crear nuestra base de datos vamos a utilizar un paquete de instalación con las librerías de las aplicaciones ya compiladas para Windows 10 en arquitectura 64 bits llamado WampServer que incluye PHP 5.4.3, Apache 2.4.2, MySQL 5.5.24 y el gestor phpMyAdmin que serán las herramientas que utilizaremos para crear y gestionar nuestra base de datos.

4.1. ¿QUÉ ES PHP?

PHP es un lenguaje de programación interpretado usado en la mayoría de webs hoy día originalmente creado para webs dinámicas, una página web dinámica permite programar aplicaciones que permitirán al usuario final de esa página un fácil mantenimiento de la misma, ya sea añadiendo entradas a un blog, introduciendo productos en una tienda virtual o cambiando los estilos de la página web.

4.2. ¿QUÉ ES APACHE?

Apache es un servidor web de protocolo HTTP que permite la transmisión de información en la World Wide Web, es de código abierto y nos permitirá la utilización de phpMyAdmin a través de nuestro navegador web.

4.3. ¿QUÉ ES phpMyAdmin?

Es una herramienta escrita en el lenguaje PHP al que se accede mediante páginas web que garantiza el control de nuestras bases de datos con una interfaz sencilla e intuitiva a la vez que potente.

También es completamente gratuito y nos ofrece una vasta cantidad de características y opciones para manejar nuestra base de datos.

4.4. INSTALACIÓN DEL PAQUETE WAMP

Debemos entrar en www.wampserver.com, descargaremos, ejecutaremos e instalaremos el paquete de instalación de la página oficial.

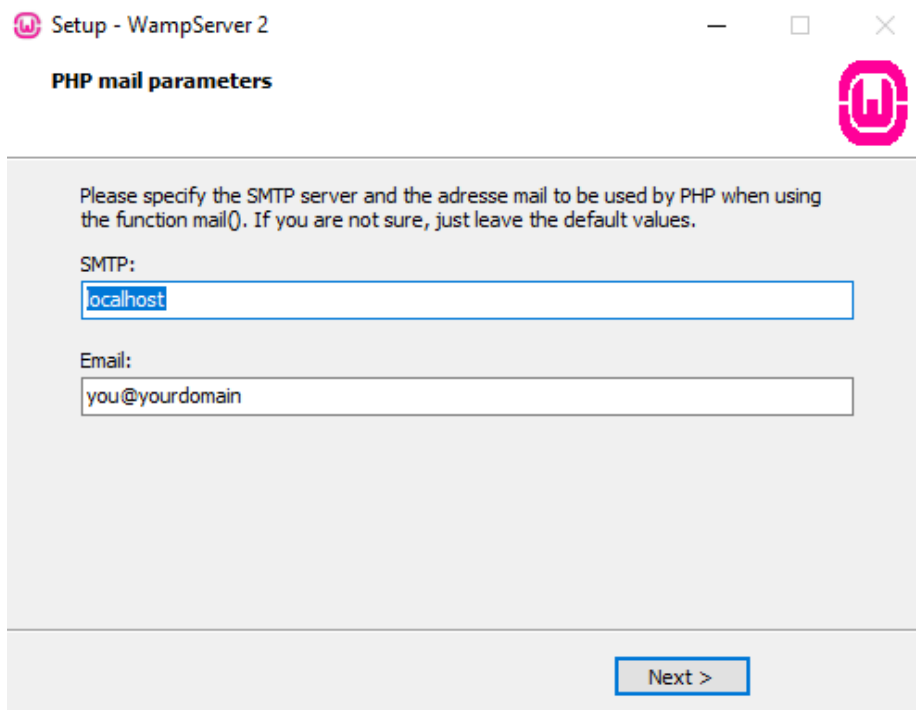
Figura 2: Instalación del servidor Apache, MySQL y el gestor phpMyAdmin (Paso 1)



Fuente: WampServer (2018) disponible on-line: <http://www.wampserver.es/>

Llegados a este punto de la instalación, nos pedirá que configuremos los parámetros del servicio de correo de PHP, vamos a usar localhost y el dominio de correo por defecto ya que no daremos uso a esta función.

Figura 3: Instalación del servidor Apache, MySQL y el gestor phpMyAdmin (Paso 2)

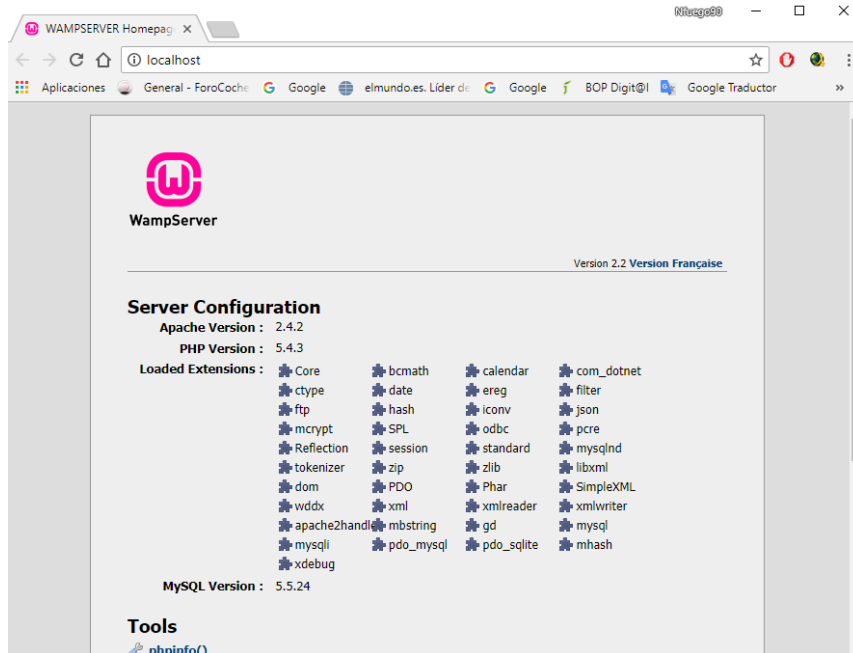


Fuente: Elaboración propia.

También se nos preguntará si queremos añadir una excepción a nuestro cortafuegos en caso de tenerlo activo, permitimos su acceso y finalizamos la instalación.

Una vez terminada la instalación comprobaremos que ha sido satisfactoria abriendo nuestro navegador web y escribiendo en la barra de direcciones lo siguiente: <http://localhost> o bien <http://127.0.0.1> que es por defecto nuestra dirección local, deberíamos poder ver lo dispuesto en la figura 4.

Figura 4: Instalación del servidor Apache, MySQL y el gestor phpMyAdmin (Paso 3)



Fuente: Elaboración propia.

5. INTRODUCCIÓN AL LENGUAJE SQL

El SQL es un lenguaje de programación que trabaja con datos relacionados entre ellos por lo tanto la mayoría de programas de gestión de bases de datos relacionales como MySQL o Microsoft Office Access usan SQL para trabajar con los datos.

La ventaja principal del lenguaje SQL es su facilidad de compresión y lectura incluso para usuarios inexpertos.

Como cualquier lenguaje de programación, tiene un conjunto de reglas que debe respetarse, a esto se le llama sintaxis. La sintaxis SQL se basa en la sintaxis del idioma inglés y del lenguaje de programación VBA.

Los comandos de SQL se dividen en muchos tipos, entre ellos tenemos el lenguaje de manipulación de datos (DML por sus siglas en inglés, Data Manipulation Language), el lenguaje de definición de datos (DDL, Data Definition Language), controles de transacción y

medidas de seguridad. Los comandos DML se usan para manipular y recuperar datos mientras que los comandos DDL se utilizan para definir y modificar las estructuras de la base de datos, los controles de transacción ayudan con el procesamiento de datos y las medidas de seguridad manejan los permisos y roles de cada usuario.

5.1 COMANDOS SQL BÁSICOS: DEFINIENDO COMO SE ALMACENAN LOS DATOS.

`CREATE DATABASE `nombre de la base de datos`:` Comando utilizado para crear una base de datos vacía.

`DROP DATABASE `nombre de la base de datos`:` Comando utilizado para eliminar totalmente una base de datos.

`CREATE TABLE `nombre de la tabla dentro de la base de datos`:` Comando utilizado para crear las tablas, que es verdaderamente donde almacenaremos nuestra información.

`ALTER TABLE `nombre de la tabla dentro de la base de datos`:` Comando utilizado para modificar los parámetros y campos de nuestra tabla.

`DROP TABLE `nombre de la tabla dentro de la base de datos`:` Comando utilizado para eliminar por completo una tabla dentro de nuestra base de datos.

5.2 COMANDOS SQL BÁSICOS: MANIPULACIÓN DE LOS DATOS.

`SELECT:` Comando utilizado cuando deseas leer o seleccionar los datos.

`INSERT INTO:` Comando utilizado para introducir nuevos datos.

`UPDATE:` Comando utilizado para modificar datos ya existentes.

`DELETE:` Comando utilizado para borrar datos.

`REPLACE:` Comando utilizado cuando quieres comprobar si un valor existe, si existe, eliminarlo e insertar nueva información y en caso de no existir, introducirlo.

TRUNCATE: Comando utilizado para eliminar todos los datos de una tabla dejándola vacía.

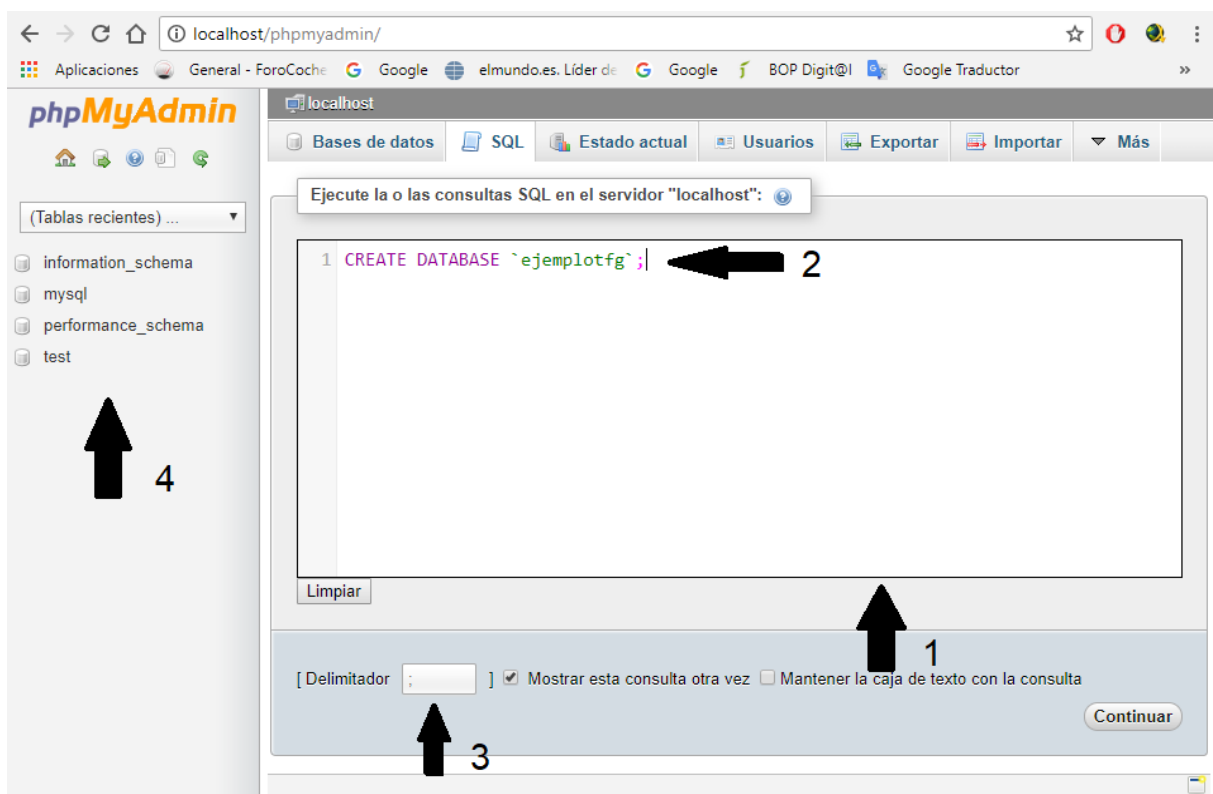
6. CREACIÓN DE UNA BASE DE DATOS

Para la creación de la base de datos vamos a utilizar una consulta, una consulta es una forma de comunicarnos con la base de datos o el servidor MySQL a través de una serie de instrucciones y condiciones que nosotros especificamos a través del propio lenguaje SQL.

Para introducir las consultas mediante phpMyAdmin necesitamos acceder a nuestro servidor ya creado anteriormente mediante la dirección web: <http://localhost/phpmyadmin/>

Una vez estemos dentro debemos ir a la pestaña SQL:

Figura 5: Partes de phpMyAdmin en el directorio raíz del servidor



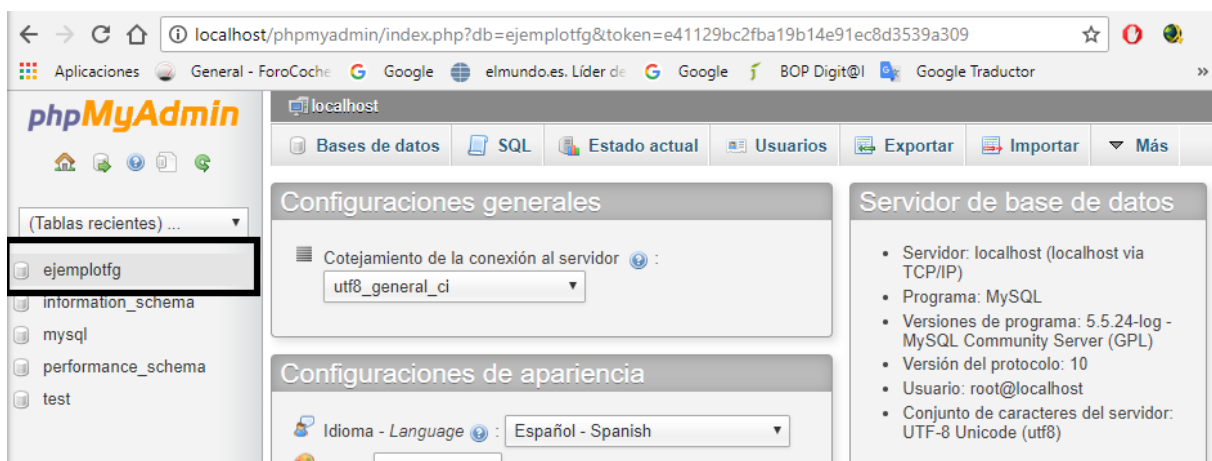
Fuente: Elaboración propia.

En la figura 5 los números corresponden a:

1. Consola donde podremos mandar información en forma de consulta al servidor MySQL.
2. Consulta que utilizaremos para crear nuestra base de datos: `CREATE DATABASE `ejemplotfg`;`
3. Delimitador de línea, es decir, el carácter que utilizaremos para indicar el final de línea dentro de nuestro código, en este caso será el punto y coma “;”.
4. Listado de todas las bases de datos existentes que tenemos en nuestro servidor.

Ejecutamos nuestra consulta, actualizamos la página y podremos ver que a la izquierda (Figura 6) se nos ha creado una base de datos nueva llamada ejemplotfg, esta será la base de datos donde trabajemos a partir de ahora.

Figura 6: Éxito en la creación de la base de datos.



Fuente: Elaboración propia.

6.1. CREACIÓN DE LAS TABLAS DENTRO DE LA BASE DE DATOS

Para la creación de las tablas, debemos introducirnos en la base de datos haciendo click en el nombre de nuestra base de datos y volviendo a hacer click en la pestaña SQL para introducir nuestra consulta de creación de la tabla.

Como ejemplo he diseñado tres consultas para crear tres tablas.

En la siguiente consulta (figura 7) vamos a crear la tabla donde registraremos tanto los usuarios como su información personal.

Los elementos que vamos a usar en la siguiente consulta son:

- ID_clientes: Nombre de la columna en la tabla que asigna un número único gracias al parámetro AUTO_INCREMENT, por cada entrada que se introduzca en la tabla este parámetro asignará automáticamente un número en una secuencia sumatoria de 1.
- Nombre: Nombre de la columna en la tabla donde guardaremos el nombre de nuestro cliente.
- Apellido1: Nombre de la columna en la tabla donde guardaremos el primer apellido de nuestro cliente.
- Apellido2: Nombre de la columna en la tabla donde guardaremos el segundo apellido de nuestro cliente.
- Edad: Nombre de la columna en la tabla donde guardaremos la edad de nuestro cliente.
- DNI: Nombre de la columna en la tabla donde guardaremos el DNI de nuestro cliente.
- Teléfono: Nombre de la columna en la tabla donde guardaremos el teléfono de nuestro cliente.
- País: Nombre de la columna en la tabla donde guardaremos el país al que pertenece nuestro cliente.
- Dirección: Nombre de la columna en la tabla donde guardaremos la dirección actual de nuestro cliente.
- Población: Nombre de la columna en la tabla donde guardaremos la población a la que pertenece nuestro cliente.
- CodPostal: Nombre de la columna en la tabla donde guardaremos el código postal perteneciente a la población en la que reside nuestro cliente.

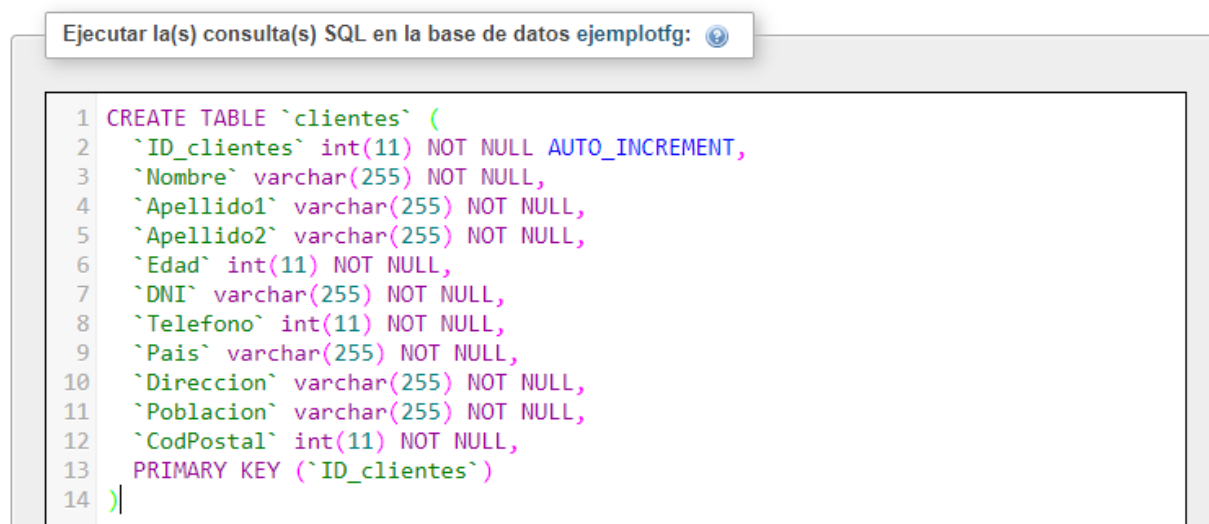
El parámetro NOT NULL asignado en las columnas impide que el valor del campo sea nulo o esté vacío.

El parámetro int indica el tamaño en bites que podrá almacenar el campo en números.

El parámetro varchar indica el tamaño en bites que podrá almacenar el campo en números, letras o la combinación de los mismos.

El parámetro PRIMARY KEY indica cuál es la clave primaria de los datos, este campo no puede repetirse en toda la tabla y nunca puede ser nulo, por lo tanto, se utiliza como identificador de la fila.

Figura 7: Diseño de la tabla ‘clientes’ en SQL

A screenshot of a SQL query editor window. The title bar reads "Ejecutar la(s) consulta(s) SQL en la base de datos ejemplotfg:". The editor contains a SQL script to create a table named 'clientes'. The script is as follows:

```
1 CREATE TABLE `clientes` (  
2   `ID_clientes` int(11) NOT NULL AUTO_INCREMENT,  
3   `Nombre` varchar(255) NOT NULL,  
4   `Apellido1` varchar(255) NOT NULL,  
5   `Apellido2` varchar(255) NOT NULL,  
6   `Edad` int(11) NOT NULL,  
7   `DNI` varchar(255) NOT NULL,  
8   `Telefono` int(11) NOT NULL,  
9   `Pais` varchar(255) NOT NULL,  
10  `Direccion` varchar(255) NOT NULL,  
11  `Poblacion` varchar(255) NOT NULL,  
12  `CodPostal` int(11) NOT NULL,  
13  PRIMARY KEY (`ID_clientes`)  
14 )|
```

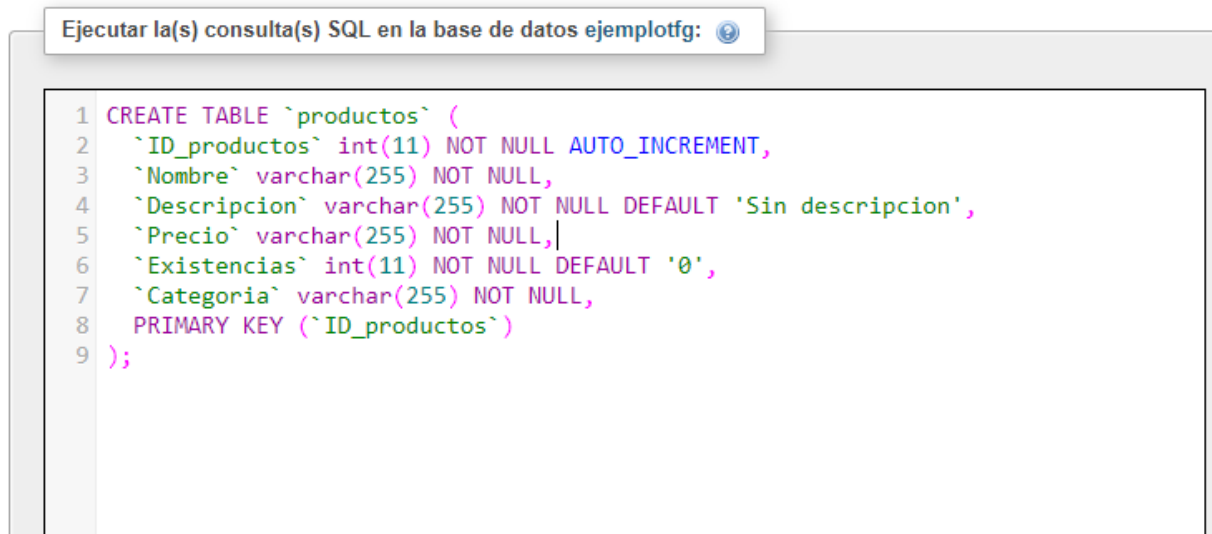
Fuente: Elaboración propia.

En la siguiente consulta (figura 8) vamos a crear la tabla donde registraremos nuestros productos y los datos de los mismos.

- ID_productos: Nombre de la columna en la tabla que asigna un número único al producto gracias al parámetro AUTO_INCREMENT, por cada entrada que se introduzca en la tabla este parámetro asignará automáticamente un número en una secuencia sumatoria de 1.
- Nombre: Nombre de la columna en la tabla donde asignamos el nombre del producto.
- Descripción: Nombre de la columna en la tabla donde describimos al producto.
- Precio: Nombre de la columna en la tabla donde fijaremos el precio del producto.

- Existencias: Nombre de la columna en la tabla donde registraremos el número de existencias del producto.
- Categoría: Nombre de la columna en la tabla donde asignaremos la categoría de nuestro producto.

Figura 8: Diseño de la tabla ‘productos’ en SQL



Ejecutar la(s) consulta(s) SQL en la base de datos ejemplotfg: 

```

1 CREATE TABLE `productos` (
2   `ID_productos` int(11) NOT NULL AUTO_INCREMENT,
3   `Nombre` varchar(255) NOT NULL,
4   `Descripcion` varchar(255) NOT NULL DEFAULT 'Sin descripcion',
5   `Precio` varchar(255) NOT NULL,
6   `Existencias` int(11) NOT NULL DEFAULT '0',
7   `Categoria` varchar(255) NOT NULL,
8   PRIMARY KEY (`ID_productos`)
9 );

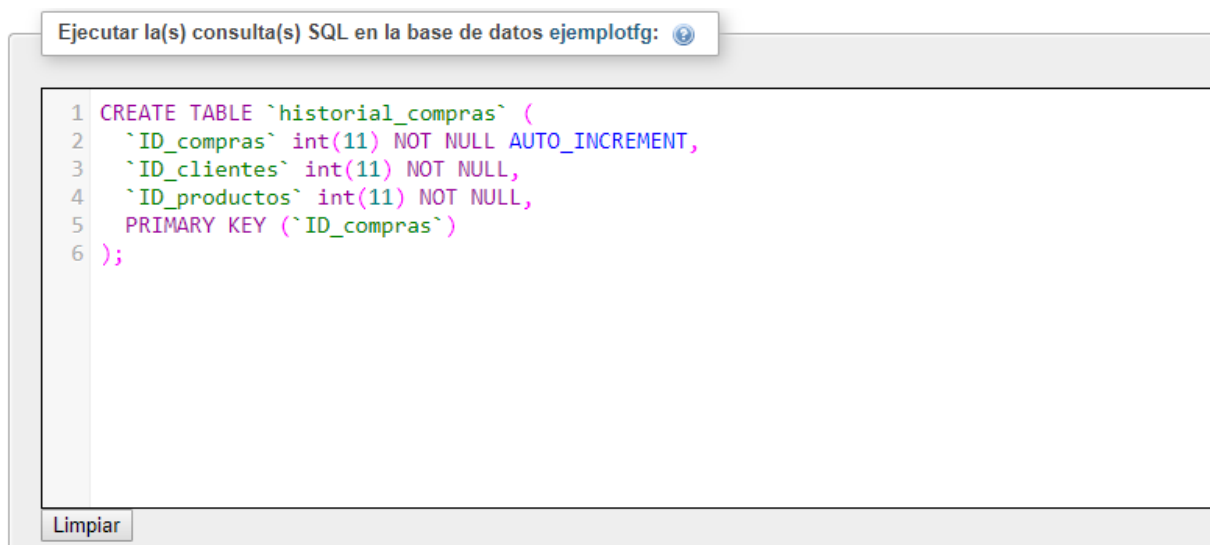
```

Fuente: Elaboración propia.

En la última consulta (figura 9) se creará una tabla destinada a registrar todas las compras que se hayan podido hacer en nuestra tienda.

- ID_compras: Nombre de la columna en la tabla que asigna un número único al producto gracias al parámetro AUTO_INCREMENT, por cada entrada que se introduzca en la tabla este parámetro asignará automáticamente un número en una secuencia sumatoria de 1.
- ID_clientes: Nombre de la columna en la tabla donde quedará registrado el ID del cliente que hizo la compra.
- ID_productos: Nombre de la columna en la tabla donde quedará registrado el ID del producto que compró el cliente.

Figura 9: Diseño de la tabla ‘historial_compras’ en SQL



The image shows a screenshot of a SQL query editor interface. At the top, there is a button labeled "Ejecutar la(s) consulta(s) SQL en la base de datos ejemplotfg:". Below this, a text area contains the following SQL code:

```
1 CREATE TABLE `historial_compras` (  
2   `ID_compras` int(11) NOT NULL AUTO_INCREMENT,  
3   `ID_clientes` int(11) NOT NULL,  
4   `ID_productos` int(11) NOT NULL,  
5   PRIMARY KEY (`ID_compras`)  
6 );
```

At the bottom left of the editor, there is a button labeled "Limpiar".

Fuente: Elaboración propia.

7. GESTIÓN DE DATOS EN LAS TABLAS

Para la introducción y gestión de los datos en phpMyAdmin podemos usar la consola, pero es más sencillo usar la interfaz que nos ofrece la aplicación web, no obstante, ambas formas tienen sus ventajas e inconvenientes.

7.1. INTRODUCCIÓN A LA GESTIÓN DE DATOS MEDIANTE CONSULTAS

Desde la pestaña SQL de nuestro gestor phpMyAdmin habiendo seleccionado una base de datos donde trabajar, podremos ejecutar las consultas de ejemplo diseñadas a continuación.

7.1.1. INTRODUCCIÓN DE DATOS

Para introducir datos nuevos usaremos el comando INSERT INTO dentro la tabla clientes y en los campos ‘Nombre’, ‘Apellido1’, ‘Apellido2’, ‘Edad’, ‘DNI’, ‘Telefono’, ‘Pais’, ‘Direccion’,

‘Poblacion’, ‘CodPostal’ tras el operador VALUES con sus datos respectivamente y usando la sintaxis SQL. De esta forma registraremos nuestros clientes (figura 9).

Figura 10: Introducción de datos mediante una consulta

The screenshot shows a window titled "Ejecutar la(s) consulta(s) SQL en la base de datos ejemplotfg:". The main text area contains the following SQL code:

```
1 INSERT INTO clientes ( Nombre, Apellido1, Apellido2,
2 Edad, DNI, Telefono, Pais, Direccion, Poblacion,
3 CodPostal )
VALUES ( "José", "García", "Lopez", "26",
"26256361Z", "683123568", "España", "C/Espronceda Nº
3 Piso 4", "Linares", "23700" );|
```

Below the text area are buttons for "SELECT *", "SELECT", "INSERT", "UPDATE", "DELETE", and "Limpiar". To the right, a list of columns is displayed: ID_clientes, Nombre, Apellido1, Apellido2, Edad, DNI, Telefono, Pais, Direccion, Poblacion, and CodPostal. At the bottom, there is a "Delimitador" field with a semicolon, a checkbox "Mostrar esta consulta otra vez" (checked), and a checkbox "Mantener la caja de texto con la consulta" (unchecked). A "Continuar" button is in the bottom right corner.

Fuente: Elaboración propia.

También vamos a introducir cuatro productos aplicando el mismo método, pero en la tabla “productos” (figura 10).

Figura 11: Introducción de datos mediante una consulta múltiple

The screenshot shows a window titled "Ejecutar la(s) consulta(s) SQL en la base de datos ejemplotfg:". The main text area contains the following SQL code:

```
1 INSERT INTO productos ( Nombre, Descripcion, Precio, Existencias, Categoria )
2 VALUES ( "Logitech g502", "Ratón gaming de alta precisión ideal para jugadores competitivos.", "70€", "5", "Ratones" );
3
4 INSERT INTO productos ( Nombre, Descripcion, Precio, Existencias, Categoria )
5 VALUES ( "HP200", "Ratón de oficina de alta calidad.", "17€", "10", "Ratones" );
6
7 INSERT INTO productos ( Nombre, Descripcion, Precio, Existencias, Categoria )
8 VALUES ( "Newskill Hanshi Spectrum", "Teclado gaming con luces RGB LED de gama alta.", "100€", "3", "Teclados" );
9
10 INSERT INTO productos ( Nombre, Descripcion, Precio, Existencias, Categoria )
11 VALUES ( "Tacens Mars Gaming MCP0", "Teclado gaming con luces de gama baja.", "15€", "20", "Teclados" );
```

Que corresponden a los datos de la tabla 1.

Tabla 1: Datos de los productos

Nombre	Descripción	Precio	Existencias o unidades disponibles	Categoría
Logitech g502	Ratón gaming de alta precisión ideal para jugadores competitivos.	70€	5	Ratones
HP200	Ratón de oficina de alta calidad.	17€	10	Ratones
Newskill Hanshi Spectrum	Teclado gaming con luces RGB LED de gama alta	100€	3	Teclados
Tacens Mars Gaming MCP0	Teclado gaming con luces de gama baja	15€	20	Teclados

Fuente: Elaboración propia.

7.1.2. MODIFICACIÓN DE DATOS

Para modificar datos ya creados usaremos el comando UPDATE `tabla` que queremos modificar SET valores y un filtro, WHERE. Si quisiéramos modificar todos los campos de una tabla quitaríamos el filtro.

En la figura 11 podemos ver que en la consulta de ejemplo vamos a filtrar los datos buscando que el campo Nombre sea igual a José de manera que sólo los campos donde esta condición se cumple sean modificados en cuanto a: 'Edad', 'DNI' y 'Telefono'.

Figura 12: Modificación de datos mediante una consulta SQL

The screenshot shows a web-based SQL query editor. At the top, a title bar reads "Ejecutar la(s) consulta(s) SQL en la base de datos ejemplotfg:". Below this is a large text area containing the SQL query: `1 UPDATE clientes SET Edad = "25", DNI = "16246020B", Telefono = "685473129" WHERE Nombre = "José";`. To the right of the text area is a list of columns from the 'clientes' table: ID_clientes, Nombre, Apellido1, Apellido2, Edad, DNI, Telefono, Pais, Direccion, Poblacion, and CodPostal. Below the text area are buttons for "SELECT *", "SELECT", "INSERT", "UPDATE", "DELETE", and "Limpiar". At the bottom, there is a section for query options: a "Delimitador" dropdown set to ";", a checked checkbox for "Mostrar esta consulta otra vez", and an unchecked checkbox for "Mantener la caja de texto con la consulta". A "Continuar" button is located in the bottom right corner.

Fuente: Fuente propia.

7.1.3 VISUALIZACIÓN Y SELECCIÓN DE DATOS

Si queremos ver la información que hemos introducido usaremos el comando SELECT, dependiendo de si queremos ver toda la información o sólo alguna utilizaríamos filtros como WHERE o IF.

Figura 13: Visualización y selección de datos mediante una consulta

The screenshot shows the same SQL query editor interface as Figure 12. The title bar is "Ejecutar la(s) consulta(s) SQL en la base de datos ejemplotfg:". The text area now contains the SQL query: `1 SELECT * FROM `clientes`;`. The column list on the right remains the same. The buttons below the text area are "SELECT *", "SELECT", "INSERT", "UPDATE", "DELETE", and "Limpiar". The bottom section shows the "Delimitador" dropdown set to ";", the "Mostrar esta consulta otra vez" checkbox checked, and the "Mantener la caja de texto con la consulta" checkbox unchecked. The "Continuar" button is in the bottom right corner.

Fuente: Elaboración propia.

Y el resultado que nos debería de dar sería el que tenemos en la figura 13 si todos los pasos son correctos mostrando todos los registros de la tabla clientes:

Figura 14: Resultados de la consulta con el comando SELECT

✓ Mostrando registros 0 - 0 (1 total, La consulta tardó 0.0023 seg)

```
SELECT *  
FROM `clientes`  
LIMIT 0 , 30
```

☐ Perfilando [En línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar : Fila de inicio: 0 Número de filas: 30 Cabeceras cada 100 filas

+ Opciones

	ID_clientes	Nombre	Apellido1	Apellido2	Edad	DNI	Telefono	Pais	Direccion	Pol
☐ Editar ☐ Copiar ☐ Borrar	1	José	García	Lopez	25	16246020B	685473129	España	C/Espronceda Nº 3 Piso 4	Lin

⬆ Marcar todos / Desmarcar todos Para los elementos que están marcados: ☐ Cambiar ☐ Borrar ☐ Exportar

Mostrar : Fila de inicio: 0 Número de filas: 30 Cabeceras cada 100 filas

Operaciones sobre los resultados de la consulta

☐ Vista de impresión ☐ Previsualización para imprimir (documento completo) ☐ Exportar ☐ Mostrar gráfico ☐ Crear vista

Fuente: Elaboración propia.

Si quisiéramos ordenar por algún valor los datos que nos devuelve la consulta SELECT deberíamos utilizar ORDER BY.

7.1.4. ELIMINACIÓN DE DATOS

Para eliminar información existente en la tabla se utiliza el comando DELETE FROM `tabla` de la que queremos eliminar la información` más un filtro, WHERE, donde delimitaremos lo que queremos borrar, en el caso de no especificar un filtro se borraría toda la información registrada en nuestra tabla.

En la siguiente consulta que podemos ver en la figura 14 eliminaremos todas las entradas que contienen 'José' en la columna llamada 'Nombre'.

Figura 15: Eliminación de datos mediante una consulta

The screenshot shows a web-based SQL query execution tool. At the top, a button says "Ejecutar la(s) consulta(s) SQL en la base de datos ejemplotfg:". Below this is a large text area containing the SQL query: `1 DELETE FROM `clientes` WHERE Nombre = "José";`. To the right of the query area is a list of columns for the 'clientes' table: ID_clientes, Nombre, Apellido1, Apellido2, Edad, DNI, Telefono, Pais, Direccion, Poblacion, and CodPostal. Below the query area are buttons for "SELECT *", "SELECT", "INSERT", "UPDATE", "DELETE", and "Limpiar". At the bottom, there is a section for query options: a "Delimitador" dropdown set to ";", a checked checkbox for "Mostrar esta consulta otra vez", an unchecked checkbox for "Mantener la caja de texto con la consulta", and a "Continuar" button.

Fuente: Elaboración propia.

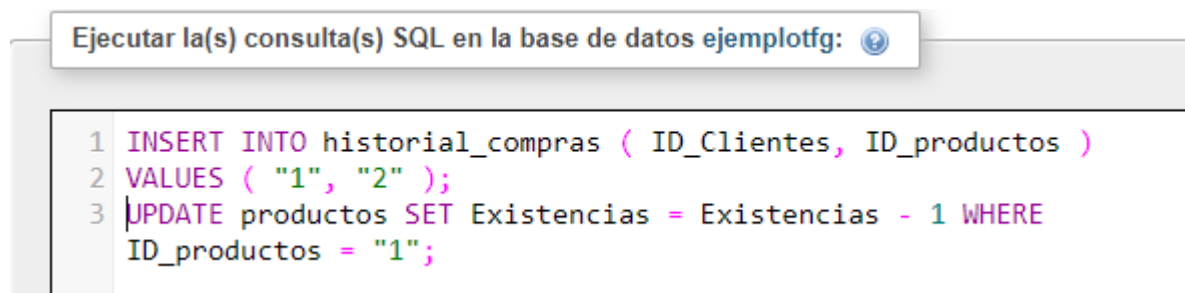
Si quisiéramos eliminar todos los datos de una tabla manteniendo la estructura de la misma lo ideal sería utilizar el comando TRUNCATE `nombre de la tabla que dejaremos vacía` puesto que utiliza menos recursos que DELETE FROM y por lo tanto es más eficiente cuando manejamos grandes cantidades de información.

Si además quisiéramos borrar también la estructura utilizaríamos la orden DROP `nombre de la tabla que eliminaremos por completo`.

7.1.5. COMBINANDO COMANDOS EN LA MISMA CONSULTA PARA REGISTRAR LAS COMPRAS

Para registrar nuestras compras vamos a necesitar la combinación de dos comandos: INSERT INTO y UPDATE. El primero será el encargado de registrar el número de identificación del cliente y del producto en la tabla historial_compras y el segundo modificará las existencias actuales del producto que el cliente ha comprado en la tabla productos.

Figura 16: Combinación de comandos en una consulta



The image shows a screenshot of a SQL query editor. At the top, there is a button labeled "Ejecutar la(s) consulta(s) SQL en la base de datos ejemplotfg:" with a blue circular icon. Below the button, the SQL query is displayed in a text area with line numbers 1, 2, and 3. The query is as follows:

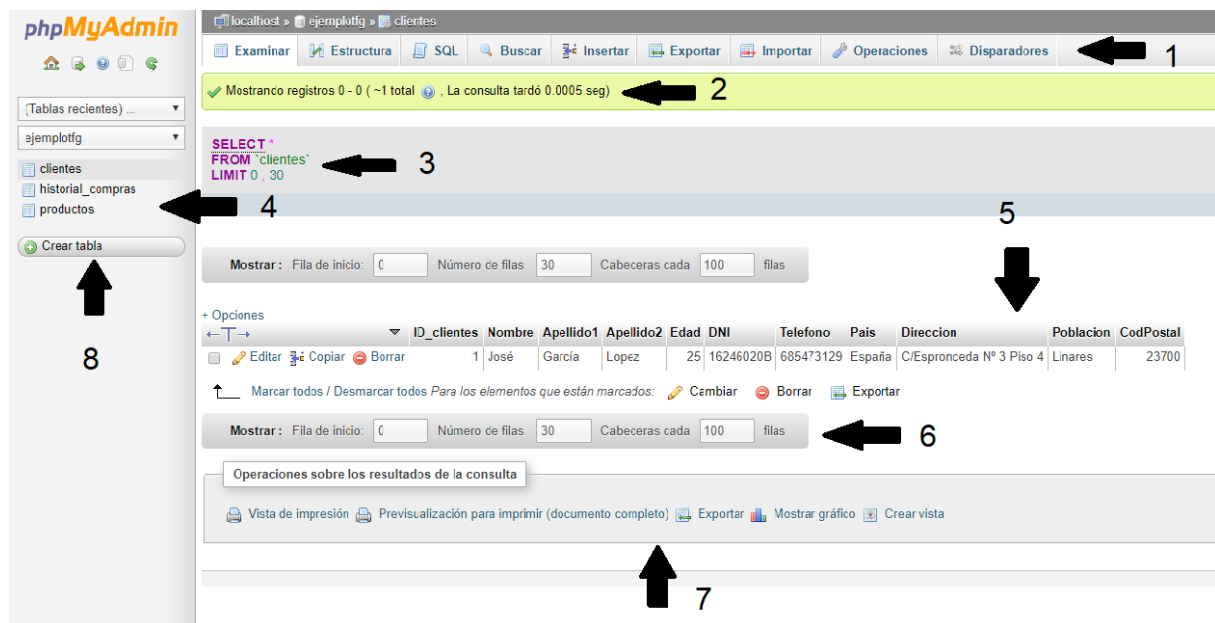
```
1 INSERT INTO historial_compras ( ID_Clientes, ID_productos )
2 VALUES ( "1", "2" );
3 UPDATE productos SET Existencias = Existencias - 1 WHERE
  ID_productos = "1";
```

Fuente: Elaboración propia.

7.2 INTRODUCCIÓN A LA GESTIÓN DE DATOS MEDIANTE LA INTERFAZ DE phpMyAdmin

A pesar de todas las posibilidades que nos ofrece el lenguaje SQL no es necesario conocerlo en profundidad teniendo un gestor como phpMyAdmin el cual gracias a su interfaz nos permite manejar una base de datos de forma sencilla.

Figura 17: Distintos elementos de phpMyAdmin



Fuente: Elaboración propia.

En la figura 16 vemos los distintos elementos que lo componen.

1. Barra de herramientas.
2. Barra que nos da información sobre el número de resultados obtenidos y la velocidad de nuestras operaciones.
3. Caja que nos muestra la última consulta que hemos ejecutado.
4. Tablas dentro de nuestra base de datos.
5. Tabla para previsualizar la información que hay en nuestra tabla.
6. Barra donde podemos configurar la vista de nuestra tabla de previsualización de datos.
7. Recuadro de herramientas sobre nuestra consulta.
8. Botón para la creación de tablas nuevas.

7.2.1. INTRODUCCIÓN DE DATOS

En la barra de herramientas de phpMyAdmin encontramos la pestaña Insertar (figura 17) que permite la introducción de datos a través de un formulario, de esta forma también se pueden agregar usuarios de una forma más cómoda sin la necesidad del lenguaje SQL.

Figura 18: Opción insertar en phpMyAdmin

Nombre	varchar(255)	
Apellido1	varchar(255)	
Apellido2	varchar(255)	
Edad	int(11)	
DNI	varchar(255)	
Telefono	int(11)	
País	varchar(255)	
Dirección	varchar(255)	
Población	varchar(255)	
CodPostal	int(11)	

Continuar

Fuente: Elaboración propia.

7.2.2. MODIFICACIÓN DE DATOS

Para modificar los datos mediante la interfaz de nuestro gestor de bases de datos tan sólo tenemos que hacer clic en cualquiera de los campos de la tabla de previsualización para poder editarlos (figura 18), una vez realizados los cambios con presionar el botón intro de nuestro teclado quedará guardado.

Figura 19: Modificación de datos utilizando la interfaz phpMyAdmin

	ID_clientes	Nombre	Apellido1	Apellido2	Edad	DNI	Telefono	Pais	Direccion	Poblacion	CodPostal
 Borrar	1	José	García	Lopez	25	16246020B	685473129	España	C/Espronceda N° 3 Piso 4	Linares	23700
 Borrar	2	Antonio	Pérez	Lozano	32	27286361Z	613683521	España	C/Jaén N° 24	Torreblascopedro	23510

Fuente: Elaboración propia.

7.2.3. BÚSQUEDA DE DATOS

En la barra de herramientas encontramos la función Buscar, que permite buscar datos desde cualquier campo y nos facilita una gran cantidad de operadores. Los comandos más usados los podemos ver en la tabla 2 y en la figura 19.

Tabla 2: Operadores SQL

Operador	Descripción
LIKE	Se utiliza para buscar cadenas de texto por patrones junto con % o también llamado comodín, si ponemos A% nos devolvería los campos que empezarán por "A".
NOT LIKE	Es el opuesto a LIKE, se utiliza para filtrar cadenas de texto. Si utilizamos J% no nos devolverá ningún campo con ese patrón.
=	Utilizado para buscar valores que coincidan con lo que especifiquemos.
!=	Se utiliza para descartar en la búsqueda los valores que coincidan con lo que especifiquemos.
IN	Se utiliza para devolver los valores que especifiquemos en una lista separada de comas.
NOT IN	Utilizado para descartar de la búsqueda los valores que especifiquemos en una lista separada de comas.
BETWEEN	Se utiliza para buscar datos que estén entre un determinado intervalo.
NOT BETWEEN	Se utiliza para descartar datos de la búsqueda aquellos valores que estén entre un determinado intervalo.
IS NULL	Busca todos los valores del campo que estén vacíos, no tendrá en cuenta los valores que especifiquemos.
IS NOT NULL	Busca todos los valores del campo que no estén vacíos, igualmente no tendrá en cuenta los valores que especifiquemos.

Fuente: Elaboración propia.

Figura 20: Operadores SQL dentro de la interfaz phpMyAdmin

The screenshot shows the phpMyAdmin SQL interface. At the top, there are tabs for 'Examinar', 'Estructura', 'SQL', 'Buscar', 'Insertar', 'Exportar', 'Importar', and 'Más'. Below these, there is a table structure with columns and their types. The 'Operador' dropdown menu is open, showing various SQL operators. The 'Valor' column contains the value 'Antonio' for the 'Nombre' column.

Columna	Tipo	Cotejamiento	Operador	Valor
ID_clientes	int(11)		=	
Nombre	varchar(255) latin1_swedish_ci		LIKE	Antonio
Apellido1	varchar(255) latin1_swedish_ci		LIKE %...%	
Apellido2	varchar(255) latin1_swedish_ci		=	
Edad	int(11)		REGEXP	
DNI	varchar(255) latin1_swedish_ci		REGEXP ^...\$	
Telefono	int(11)		NOT REGEXP	
Pais	varchar(255) latin1_swedish_ci		= "	
Direccion	varchar(255) latin1_swedish_ci		LIKE	
Poblacion	varchar(255) latin1_swedish_ci		LIKE	
CodPostal	int(11)		=	

Fuente: Elaboración propia.

7.2.4. ALGUNAS OPERACIONES DE GRAN UTILIDAD EN phpMyAdmin

Una de las grandes ventajas de phpMyAdmin es que pone a fácil disposición características avanzadas que requerirían de un profundo conocimiento del lenguaje SQL.

En la figura 20 podemos ver las siguientes funciones:

- Modificar el ORDER BY: Con esta herramienta podemos ordenar la cuadrícula de visualización de la forma que queramos por el valor que especifiquemos.
- Mover tablas: Herramienta que permite mover tablas entre bases de datos.
- Opciones de la tabla: Recuadro de configuración donde podemos cambiar el nombre de la tabla, añadir un comentario descriptivo, cambiar el motor de almacenamiento de datos, predefinir el conjunto de caracteres que queremos usar o fijar el valor del AUTO_INCREMENT.

- Copiar la tabla: De igual forma que la opción de mover la tabla también la podemos copiar en la misma base de datos o en otra diferente, también nos ofrece una serie de parámetros.

Figura 21: Operaciones útiles de phpMyAdmin

The screenshot displays the phpMyAdmin interface with four panels for table operations:

- Modificar el ORDER BY de la tabla:** Shows a dropdown for 'ID_clientes' (solamente) and a dropdown for 'Ascendente'. A 'Continuar' button is at the bottom.
- Mover tabla a (Base de datos.tabla):** Shows a dropdown for 'ejemplotfg' and a text input for 'clientes'. A checkbox 'Añadir el valor AUTO_INCREMENT' is checked. A 'Continuar' button is at the bottom.
- Opciones de la tabla:** Includes fields for 'Cambiar el nombre de la tabla a' (clientes), 'Comentarios de la tabla', 'Motor de almacenamiento' (InnoDB), 'Cotejamiento' (latin1_swedish_ci), 'AUTO_INCREMENT' (3), and 'ROW_FORMAT' (COMPACT). A 'Continuar' button is at the bottom.
- Copiar la tabla a (base de datos.tabla):** Shows a dropdown for 'ejemplotfg' and a text input for 'clientes'. Radio buttons for 'Únicamente la estructura', 'Estructura y datos' (selected), and 'Solamente datos' are present. Checkboxes for 'Añada DROP TABLE', 'Añadir el valor AUTO_INCREMENT', and 'Cambiar a la tabla copiada' are also shown. A 'Continuar' button is at the bottom.

Fuente: Elaboración propia.

8. PLANES DE MANTENIMIENTO EN MySQL

Los planes de mantenimiento dentro de MySQL típicos y que normalmente están automatizados en bases de datos grandes son los siguientes:

8.1 Copias de seguridad

Se trata del guardado de la información y la estructura de las tablas dentro de un archivo por si la base de datos fuese víctima de algún pirata informático o error en el sistema, este archivo será capaz de recrear toda la información que hayamos guardado en cualquier servidor basado en SQL.

PhpMyAdmin nos ofrece en su interfaz la capacidad de exportar nuestras tablas ofreciendo algunas opciones como qué tablas queremos exportar y en qué formato de archivo (Figura 21).

Figura 22: Opción exportar en phpMyAdmin

The screenshot shows the 'Export' tab selected in the phpMyAdmin navigation bar. The main heading is 'Exportando filas de la tabla "clientes"'. Under the 'Método de exportación:' section, the 'Rápido' option is selected. Under the 'Formato:' section, 'SQL' is selected in the dropdown menu. A 'Continuar' button is at the bottom.

Fuente: Elaboración propia.

En el caso de que quisiéramos importar un archivo guardado anteriormente podemos ir a la herramienta Importar, seleccionar el archivo, configurar las opciones y ejecutarlo (Figura 22).

Figura 23: Opción importar en phpMyAdmin

The screenshot shows the 'Import' tab selected in the phpMyAdmin navigation bar. The main heading is 'Importando en la tabla "clientes"'. Under the 'Archivo a importar:' section, there is a text area for the file path and a 'Seleccionar archivo' button. Below this, the 'Conjunto de caracteres del archivo:' is set to 'utf-8'. Under the 'Importación parcial:' section, the 'Permitir la interrupción...' checkbox is checked, and the 'Número de filas a omitir' is set to 0. Under the 'Formato:' section, 'SQL' is selected in the dropdown menu.

Fuente: Elaboración propia.

8.2 Limpieza de logs, optimización de las tablas y vaciamiento del cache de las consultas

Desde el gestor de bases de datos que estamos utilizando podemos hacer estas tareas de una manera sencilla y en pocos segundos.

En la figura 23 podemos ver las opciones dentro del recuadro de Mantenimiento de la tabla, es recomendable realizar todas estas opciones al menos una vez por semana para mantener la salud de nuestra base de datos en buen estado, el mantenimiento no sólo borrará archivos basura sino mejorará la rapidez con la que se ejecutan las consultas en nuestra base de datos.

Figura 24: Opciones de mantenimiento en phpMyAdmin

The screenshot displays the phpMyAdmin interface for table maintenance. At the top, there is a navigation bar with tabs: Examinar, Estructura, SQL, Buscar, Insertar, Exportar, Importar, and Más. Below this, the 'Estructura' tab is active, showing the 'Mantenimiento de la tabla' section. This section includes fields for 'Comentarios de la tabla', 'Motor de almacenamiento' (set to InnoDB), 'Cotejamiento' (set to latin1_swedish_ci), 'AUTO_INCREMENT' (set to 3), and 'ROW_FORMAT' (set to COMPACT). To the right, there are radio buttons for 'Únicamente la estructura', 'Estructura y datos' (selected), and 'Solamente datos', along with checkboxes for 'Añada DROP TABLE', 'Añadir el valor AUTO_INCREMENT', and 'Cambiar a la tabla copiada'. A 'Continuar' button is at the bottom right of this section. Below the main form, there are two side-by-side panels. The left panel, titled 'Mantenimiento de la tabla', contains a list of actions: 'Revisar la tabla', 'Defragmentar la tabla', 'Optimizar la tabla', and 'Vaciar el caché de la tabla (FLUSH)'. The right panel, titled 'Borrar datos o tabla', contains a list of actions: 'Vaciar la tabla (TRUNCATE)' and 'Borrar la tabla (DROP)'. Both panels have a 'Continuar' button at the bottom.

Fuente: Elaboración propia.

9. CONCLUSIONES

Para empezar, me gustaría explicar por qué elegí este Trabajo Fin de Grado, el principal motivo es porque tengo experiencia anterior manejando bases de datos con el gestor MySQL y quería hacer una guía para que un usuario sin conocimientos de este campo pudiese aprender los conceptos básicos de bases de datos, nociones de MySQL en un entorno sencillo y que favorece el aprendizaje como es phpMyAdmin.

También he elegido este tema porque creo que la importancia de las bases de datos hoy en día es increíblemente grande y para cualquier sitio web o negocio ya es casi de obligado conocimiento.

Puesto que yo aprendí de forma autodidacta en mis inicios como desarrollador web para comunidades de juegos y posteriormente en las asignaturas de Informática quería hacer un tutorial y una documentación básica que pudiera serle útil a otras personas a la hora de montar el sistema de gestión de bases de datos de su propio sitio web o de su negocio o proyecto de forma local.

Creo firmemente que MySQL y phpMyAdmin son unas de las herramientas que pueden impulsar un negocio o idea de cualquier tipo y aumentar todas las posibilidades que pueda tener.

10. BIBLIOGRAFÍA

“Elementos del lenguaje (Transact-SQL)” (10 de junio de 2016) disponible on-line: <https://docs.microsoft.com/es-es/sql/t-sql/language-elements/>

“DB-Engines Ranking” (Julio de 2018) disponible on-line: <https://db-engines.com/en/ranking>

Iruela, J. “Los gestores de bases de datos más usados” (19 de enero de 2016) disponible on-line: <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>

“MySQL Reference Manual” (Sin fecha) disponible on-line: <https://dev.mysql.com/doc/refman/5.5/en/>

“SQL Syntax” (Sin fecha) disponible on-line: https://www.w3schools.com/sql/sql_syntax.asp

“Basic SQL Statements” (Sin fecha) disponible on-line: <https://mariadb.com/kb/en/library/basic-sql-statements/>

“¡Bienvenido a la documentación de phpMyAdmin!” (Sin fecha) disponible on-line: <https://docs.phpmyadmin.net/es/latest/>

“Historia de las bases de datos” (4 de enero de 2011) disponible on-line: <http://histinf.blogs.upv.es/2011/01/04/historia-de-las-bases-de-datos/>