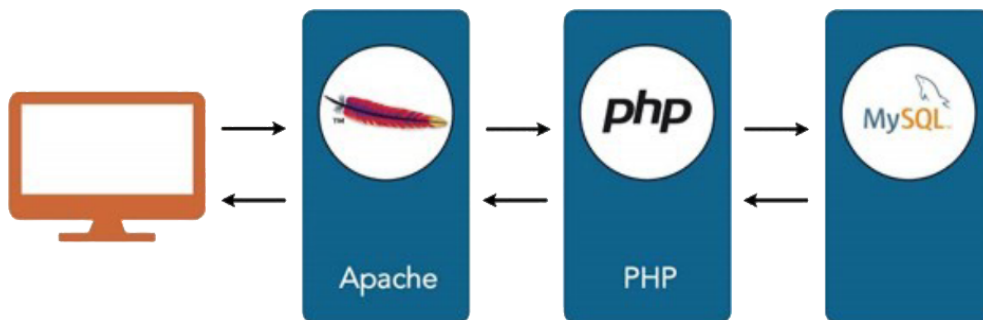


## unidad didáctica 6

# Acceso a datos



1. **duración y criterios de evaluación**
2. **instalación**
3. **estructura de una base de datos**
4. **CholloSevero**
5. **SQL**
6. **phpMyAdmin**
  6. 1. **Creando una base de datos dentro de phpMyAdmin**
  6. 2. **opciones en phpMyAdmin**
7. **MySQLi**
  7. 1. **recuperando datos de una BD**
8. **PHP Data Objects :: PDO**
9. **referencias**

# 1. duración y criterios de evaluación

---

En esta unidad vamos a aprender a acceder a datos que se encuentran en un servidor; recuperando, editando y creando dichos datos a través de una base de datos.

A través de las distintas capas o niveles, de las cuales 2 de ellas ya conocemos (*Apache, PHP*) y *MySQL* la que vamos a estudiar en este tema.

**Duración estimada:** - sesiones

---

**Resultado de aprendizaje y criterios de evaluación:**

X. ---

## 2. instalación

A través de **XAMPP** es muy sencillo, simplemente nos descargaríamos el programa y lo activaríamos. Para descargar XAMPP [pulsa aquí](#).

Con **Docker** nos descargaremos [esta imagen de docker](#) y lanzamos

```
1 | docker-compose up -d
```

Si todo ha salido bien y el contenedor está en marcha, podremos visitar la página de phpMyAdmin de la siguiente manera

```
1 | http://localhost:8000
```



Para acceder debemos utilizar las siguientes credenciales que vienen configuradas en el archivo docker-compose.yml

```
1 | usuario: root
2 | contraseña: 1234
```

### 3. estructura de una base de datos

Sabemos que una base de datos tiene muchos campos con sus nombres y valores, pero además sabemos que la base de datos debe tener un nombre. por tanto tendríamos la siguiente estructura para una base de datos:

```

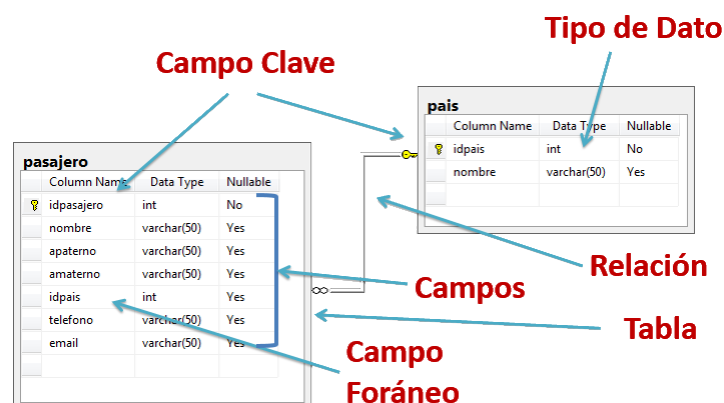
1  NombreBaseDeDatos
2      |__Tabla-#1
3      |          |__DatosTabla-#1
4      |
5      |__Tabla-#2
6      |          |__DatosTabla-#2
7      |
8      |__Tabla-#3
9      |          |__DatosTabla-#3
10     [...]
    
```

Veámoslo en un ejemplo real:

```


1  Ryanair
2      |__pasajero
3      |      |__id[*]
4      |      |__nombre
5      |      |__apellidos
6      |      |__edad
7      |      |__id_vuelo[^]
8      |
9      |__vuelo
10     |      |__id[*]
11     |      |__n_plazas
12     |      |__disponible
13     |      |__id_pais[^]
14     |
15     |__pais
16         |__id[*]
17         |__nombre
    
```


[\*] Clave primaria    [^] Clave Foránea






## 4. CholloSevero


Como muy bien habéis elegido, a lo largo de esta unidad vamos a trabajar con una base de datos que iremos confeccionando conforme avancemos, donde almacenaremos la información relacionada con ofertas que publiquen los usuarios y los listaremos en función de varios filtros; nuevos, más votados, más vistos, más comentados entre otros, al más puro estilo [Chollometro](#).



**Portátil LENOVO IdeaPad 3 15.6" - Intel Core i7-1165G7 - 8 GB RAM - 512 GB SSD PCIe - Intel Iris Xe Graphics**  
**579€** 699€ (-17%)  Envío gratuito | Ofertas MediaMarkt




Buen precio en Mediamarkt, precio final al tramitar el pedido con el descuento directo de 100€. Portátil - Lenovo IdeaPad 3 15ITL6, 15.6" FHD, Intel® Core™ i7-1165G7, 8 GB RAM, 512 GB SSD PCIe, Windows 11... [Leer más](#)


 Destrangis   23 [Ir al chollo](#)




**Peluches solidarios de Carrefour (para enfermedades infantiles) por solo 3'50€**  
**3,50€** | Ofertas Carrefour




Peluches solidarios de Carrefour, para combatir enfermedades raras infantiles. Recordar el chollo es ser solidario <3


 Samarkando   36 [Ir al chollo](#)



**Barra de sonido LG SN7Y, Subwoofer inalámbrico, 380 W, Dolby Atmos, Meridian Sound, ASC, Bass Blast+**  
**234€** 399€ (-41%)  Envío gratuito | Ofertas MediaMarkt

Descuento de 25€ al añadir al carrito. Con la barra de sonido LG SN7Y de color negro escucharás tus canciones favoritas en una calidad de audio impresionante gracias a sus tecnologías. [Leer más](#)

 TheGamerKap   26 [Ir al chollo](#)



**¡CELEBRA LA NAVIDAD CON HASTA UN 50% DTO. EN HUAWEI!\***

hace 2 h, 19 m

Categorías populares

- Portátiles
- Consola PS5
- Consola Xbox S...
- Televisores
- Smartphones y ...

hace 2 h, 51 m

Tiendas populares

## 5. SQL

---

Este lenguaje de consulta estructurada (Structured Query Language) es el que vamos a utilizar para realizar las consultas a nuestras bases de datos para mostrar el contenido en las distintas interfaces web que creemos a lo largo de la unidad. Si quieres saber más detalles visita [Wiki SQL](#).

Ejemplo de una sentencia SQL donde seleccionamos todas las filas y columnas de nuestra tabla llamada '**pais**'

```
1 | SELECT * FROM pais
```

Estas sentencias pueden invocarse desde la consola de comandos mediante el intérprete *mysql* (previamente instalado en el sistema) o a través de la herramienta phpMyAdmin.

Las sentencias SQL también las podemos usar dentro de nuestro código php, de tal manera que cuando se cargue nuestra interfaz web, lance una sentencia SQL para mostrar los datos que queramos.

```
1 | <?php
2 |     // Listado de clientes, ordenados por DNI de manera ASCendente
3 |     $clientesOrdenadosPorDNI = "SELECT * FROM `pasajero` ORDER BY `dni`" ASC;
4 | ?>
```

## 6. phpMyAdmin

Este software funciona bajo Apache y PHP y es más que nada una interfaz web para gestionar las bases de datos que tengamos disponibles en nuestro servidor local. Muchos *hostings* ofrecen esta herramienta por defecto para poder gestionar las BBDD que tengamos configuradas bajo nuestra cuenta.



### 6.1. Creando una base de datos dentro de phpMyAdmin

1. Para crear una nueva base de datos debemos entrar en phpMyAdmin como **usuario root** y pinchar en la opción **Nueva** del menú de la izquierda.
2. En la nueva ventana de creación pondremos un **nombre** a nuestra bbdd.
3. También estableceremos el **cotejamiento** `utf8m4_unicode_ci` para que nuestra bbdd soporte todo tipo de caracteres (como los asiáticos) e incluso emojis ;)
4. Le damos al botón de **Crear** para crear la bbdd y empezar a escribir las distintas tablas que vayamos a introducir en ella.



El sistema generará el código SQL para crear todo lo que le hemos puesto y creará la base de datos con las tablas que le hayamos metido.

```
1 CREATE TABLE `persona`. ( `id` INT NOT NULL AUTO_INCREMENT , `nombre` TINYTEXT NOT NULL , `apellidos` TEXT NOT NULL , `telefono` TINYTEXT NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB;
```

### 6.2. opciones en phpMyAdmin

Cuando seleccionamos una base de datos de la lista, el sistema nos muestra varias pestañas con las cuales interactuar con la base de datos en cuestión:

- **Estructura** : Podemos ver las distintas tablas que consolidan nuestra base de datos.



- **SQL** : Por si queremos inyectar código SQL para que el sistema lo interprete.
- **Buscar** : Sirve para buscar por términos, en nuestra base de datos, aplicando distintos filtros de búsqueda.
- **Generar consulta** : parecido a SQL pero de una manera más gráfica, sin tener que saber nada del lenguaje.
- **Exportar** e **importar** : Como su nombre indica, para hacer cualquiera de las 2 operaciones sobre la base de datos.
- **Operaciones** : Distintas opciones avanzadas para realizar en nuestra base de datos, de la cual destacaremos la opción **Cotejamiento** donde podremos cambiar el cotejamiento de nuestra tabla pero **OJO CON ÉSTO porque podemos eliminar datos sin querer, ya que al cambiar el cotejamiento podemos suprimir caracteres no soportados por el nuevo cotejamiento.**

No vamos a profundizar en el resto de opciones pero, en la pestaña **Más** existe la opción **Diseñador** para poder editar las relaciones entre tablas de una manera gráfica (pinchando y arrastrando) que veremos más adelante.

## 7. MySQLi

PHP hace uso de esta extensión mejorada para poder comunicarse con las bases de datos, ya sean MySQL (4.1 o superior) o MariaDB.

Cualquier consulta que queramos hacer a una base de datos necesitaremos hacer uso de la extensión `mysqli()`.

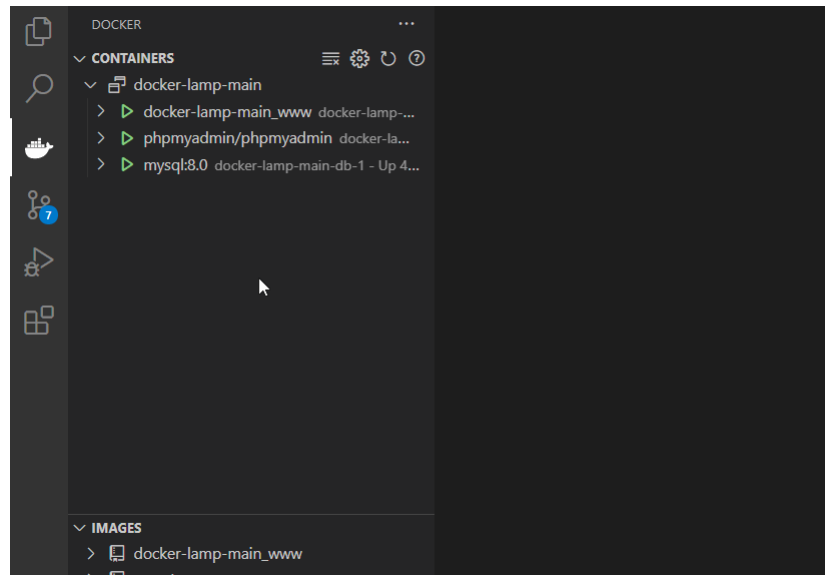
Veamos como conectarnos con una base de datos a través del código PHP:

```

1  <?php
2      // pruebas.php
3
4      // "SERVIDOR", "USUARIO", "CONTRASEÑA", "BASE DE DATOS"
5      $conexion = mysqli_connect("d939ebf6a741","tuUsuario","1234","pruebas");
6
7      // COMPROBAMOS LA CONEXIÓN
8      if(mysqli_connect_errno()) {
9          echo "Failed to connect to MySQL: " . mysqli_connect_error();
10         exit();
11     }
12
13     echo "<h1>Bienvenid@ a MySQL !!</h1>";
14 ?>

```

- `servidor`: El servidor donde se encuentra la base de datos que queremos usar suele ser **localhost**, pero en nuestro caso, al utilizar Docker será el nombre de la imagen **mysql:8.0** que aparece al dejar el ratón encima en el Visual Studio Code.



- `usuarioDB`: El usuario de la base de datos.
- `passwordDB`: La contraseña para ese usuario de la base de datos.
- `baseDeDatos`: Nombre de la base de datos que queramos usar.

Si todo ha salido bien habréis visto un mensaje diciendo **Bienvenid@ a MySQL !!**

## 7.1. recuperando datos de una BD

Ahora que ya sabemos cómo conectarnos a una base de datos alojada en nuestro servidor, lo que necesitamos saber es cómo recuperar datos almacenados en la base de datos.

Durante la instalación de la imagen de Docker, se ha creado una tabla llamada **Pruebas** que contiene varios registros de distintas personas.

Vamos a recuperar esos datos para ver de qué forma se hace con PHP.

```

1  <?php
2      // pruebas.php
3
4      $conexion = mysqli_connect("d939ebf6a741", "lupa", "1234", "pruebas");
5
6      // COMPROBAMOS LA CONEXIÓN
7      if (mysqli_connect_errno()) {
8          echo "Failed to connect to MySQL: " . mysqli_connect_error();
9          exit();
10     }
11
12     // CONSULTA A LA BASE DE DATOS
13     $consulta = "SELECT * FROM `Person`";
14     $listaUsuarios = mysqli_query($conexion, $consulta);
15
16     // COMPROBAMOS SI EL SERVIDOR NOS HA DEVUELTO RESULTADOS
17     if($listaUsuarios) {
18
19         // RECORREMOS CADA RESULTADO QUE NOS DEVUELVE EL SERVIDOR
20         foreach ($listaUsuarios as $usuario) {
21             echo "
22                 $usuario[id]
23                 $usuario[name]
24                 <br>
25             ";
26         }
27     }
28     ?>

```

Si todo ha salido bien, por pantalla verás el siguiente listado

```

1  http://localhost/pruebas.php
2
3  1 William
4  2 Marc
5  3 John
6  4 Antonio Moreno

```

## 8. PHP Data Objects :: PDO

De la misma manera que hemos visto con mysqli, PHP Data Objects (o PDO) es un driver de PHP que se utiliza para trabajar bajo una interfaz de objetos con la base de datos. A día de hoy es lo que más se utiliza para manejar información desde una base de datos, ya sea relacional o no relacional.

De igual manera que pasaba con los objetos en PHP nativos, en la interfaz de MySQL la cosa cambia la hora de conectarnos con una base de datos.

```
1 <?php
2 $conexion = new PDO('mysql:host=localhost; dbname=dwes', 'dwes', 'abc123');
```

Además, con PDO podemos usar las excepciones con try catch para gestionar los errores que se produzcan en nuestra aplicación, para ello, como hacíamos antes, debemos encapsular el código entre bloques try / catch.

```
1 <?php
2 $dsn = 'mysql:dbname=prueba;host=127.0.0.1';
3 $usuario = 'usuario';
4 $contraseña = 'contraseña';
5
6 try {
7     $mbd = new PDO($dsn, $usuario, $contraseña);
8     $mbd->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
9 } catch (PDOException $e) {
10     echo 'Falló la conexión: ' . $e->getMessage();
11 }
```

En primer lugar, creamos la conexión con la base de datos a través del constructor PDO pasándole la información de la base de datos.

En segundo lugar, establecemos los parámetros para manejar las excepciones, en este caso hemos utilizado:

- `PDO::ATTR_ERRMODE` indicándole a PHP que queremos un reporte de errores.
- `PDO::ERRMODE_EXCEPTION` con este atributo obligamos a que lance excepciones, además de ser la opción más humana y legible que hay a la hora de controlar errores.

Cualquier error que se lance a través de PDO, el sistema lanzará una **PDOException**.

## 9. referencias

---

- [Tutorial de Composer](#)
- [Web Scraping with PHP – How to Crawl Web Pages Using Open Source Tools](#)
- [PHP Monolog](#)
- [Unit Testing con PHPUnit — Parte 1.](#), de Emiliano Zublena.