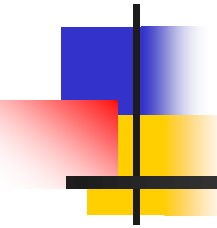
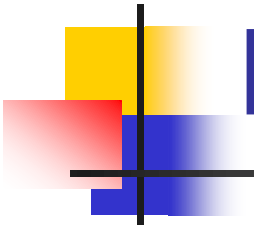


7. PHP - Acceso a BBDD

Profesor: Alejandro Amat Reina
alex@iessanvicente.com



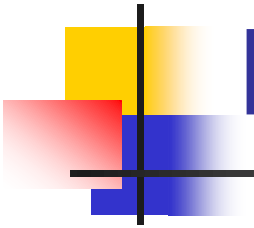
MariaDB

- Nosotros usaremos MariaDB (sucesor de MySql)
- SGBD relacional con licencia GPL
- El más empleado con el lenguaje PHP
- El driver nativo de PHP para MariaDB se llama mysqli (sucesor del driver mysql)

Características de MariaDB

- Incorpora múltiples motores de almacenamiento
- Elegiremos el motor en función de las características propias de la aplicación
- Por defecto utiliza MyISAM:
  - Muy rápido
  - No contempla integridad referencial ni tablas transaccionales
- El motor InnoDB es un poco más lento, pero sí soporta estas dos características
- Casi siempre usaremos InnoDB
- Documentación:
  - https://mariadb.com/kb/en/library/documentation/

Instalación y configuración de MariaDB

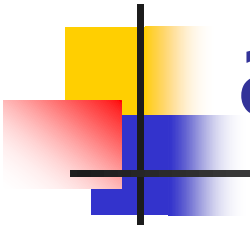


- No es necesario que lo instalemos puesto que lo tenemos integrado en nuestro XAMPP
- Se ejecuta por defecto en el puerto 3306
- El fichero de configuración se llama /opt/lampp/etc/my.cnf
- Su contenido se divide en secciones
- Las directivas relacionadas con el servidor están en la sección [mysqld]

Parámetros de configuración

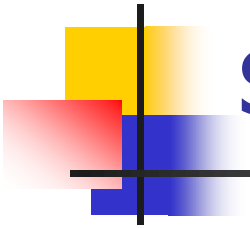
- Entre los parámetros que podemos configurar en el fichero my.cnf tenemos:
  - port. Indica el puerto TCP en el que escuchará el servidor y con el que se establecerán las conexiones
  - user. Nombre del usuario que se utilizará para ejecutar el servidor
  - datadir. Directorio del servidor en el que se almacenarán las bases de datos
  - Language. Directorio donde se encuentran los mensajes de error en el idioma deseado (para español /opt/lampp/share/mysql/spanish)
- Para más información:
  - https://mariadb.com/kb/en/library/mysqld-configuration-files-and-groups/

Herramientas de administración



- Existen muchas herramientas que permiten establecer una conexión con un servidor MariaDB para realizar tareas de administración
- Algunas se incluyen con el propio servidor
- Otras es necesario instalarlas de forma independiente

Herramientas incluidas con el servidor



- mysql. Permite conectarse a un servidor MySQL para ejecutar sentencias SQL
- mysqladmin. Es un cliente específico para tareas de administración
- mysqlshow. Muestra información sobre bases de datos y tablas
- Mysqldump. Herramienta para creación de backups
- Para un listado más completo:
  - https://mariadb.com/kb/en/library/clients-utilities/

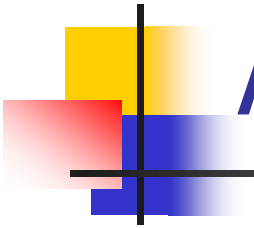
Herramientas de administración independientes

- MySQL Workbench: Herramienta genérica con interface gráfico nativo que permite administrar tanto el servidor como las bases de datos que éste gestiona
  - http://dev.mysql.com/doc/workbench/en/index.html
- PHPMyAdmin: es una aplicación web muy popular para la administración de servidores MySQL.
  - http://www.phpmyadmin.net/
- Nosotros usaremos PHPMyAdmin

Instalación de PHPMyAdmin

- No se instala con el servidor MariaDB
- Instalación en Debian:

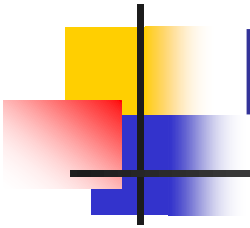
  sudo apt-get install phpmyadmin
- Viene integrada en XAMPP, no necesitamos instalarla
- Documentación:
  - https://docs.phpmyadmin.net/en/latest/



Acceso a BBDD desde PHP

- Para acceder a BBDD MariaDB tenemos dos opciones:
  - El driver nativo mysqli
  - El driver genérico PDO
- En este curso utilizaremos mysqli, aunque el uso de PDO es muy similar

Acciones a realizar sobre la BBDD



- Establecer conexiones
- Ejecutar sentencias SQL
- Obtener los registros afectados o devueltos por una sentencia SQL
- Gestionar los errores que se produzcan durante la conexión o en el establecimiento de la misma

Establecimiento de la conexión

- Tendremos que instanciar un objeto de la clase mysqi pasándole los siguientes parámetros:
  - Dirección del servidor: En nuestro caso el servidor será localhost, ya que el SGBD se encuentra en la misma máquina que el servidor web
  - Nombre de usuario con permisos para establecer la conexión
  - Contraseña del usuario
- Ejemplo:
  - `$con = new mysqli('localhost', 'root', '');`

Control de errores en la conexión

- Para verificar si se ha producido algún error en la usaremos connect_error.
- Esta propiedad de la conexión que hemos creado anteriormente, nos indicará si se ha producido alg´n error.
- El uso será el siguiente:

```
$con = @ new mysqli('localhost', 'root', '');
if ($con->connect_error) {
    die('Error de conexión: ' . $con->connect_error);
}
```

- El operador @ silencia los errores que se produzcan en la instrucción en el que lo indiquemos.
- La función die muestra un mensaje y detiene la ejecución del script

Selección de la BBDD

- Para seleccionar la BBDD con la que vamos a trabajar usaremos el método select_db pasándole el nombre de la BBDD.
- Lógicamente, el usuario con el que iniciamos la sesión tiene que tener acceso a esta BBDD.
- Ejemplo:

```
$con->select_db('carrito');
if ($con->errno !== 0)
    die('Error al seleccionar la BBDD: ' . $con->error);
```

  - errno guardará el nº de error que se ha producido en la última operación de BBDD
  - error guardará el texto descriptivo del error

Ejemplo de conexión

```
$con = @ new mysqli('localhost', 'root', '');
if ($con->connect_error) {
    die('Error de conexión: ' . $con->connect_error);
}

$con->select_db('carrito');
if ($con->errno !== 0)
    die('Error al seleccionar la BBDD: ' . $con->error);
```

Cierre de la conexión

- La conexión permanecerá activa durante el tiempo de vida del objeto $con
- Para cerrar la conexión es necesario llamar al método close de este objeto

```
$con->close();
```

Ejecución de consultas que devuelven datos

- Si la consulta genera un conjunto de datos (SELECT), se puede utilizar el método query
- Ejecuta una sentencia SQL
- Devuelve un conjunto de resultados como un objeto mysqli_result
  - `$resultado = $con->query("SELECT id, nombre FROM articulos");`
- **Esta sentencia es vulnerable a inyecciones SQL**
- En su lugar se recomienda utilizar consultas preparadas, pero no las veremos en este curso

Obtención y utilización de conjuntos de resultados

- Tenemos varias posibilidades para tratar el conjunto de resultados devuelto por el método query
- Nosotros usaremos el método fetch_array
- Devuelve un registro del conjunto de resultados, como un array asociativo, donde las claves serán los nombres de las columnas

```
$resultado = $pdo->query("SELECT id, unidades FROM articulos");
while ($registro = $resultado->fetch_array())
{
    echo "Artículo ".$registro['id'].": ";
    echo $registro['unidades']."<br>";
}
```