



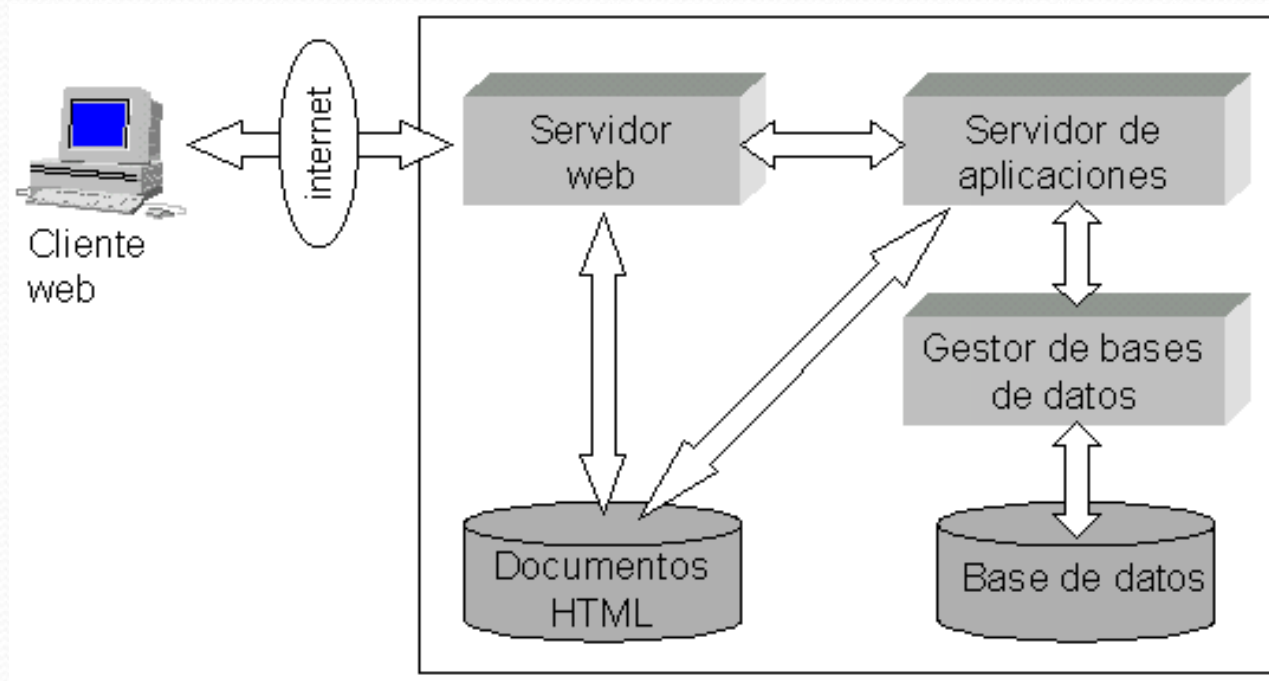
U.T.6 : Acceso a Bases de Datos MySQL desde PHP.

Bases de datos en la Web

- ❑ Las bases de datos permiten almacenar de una forma estructurada y eficiente toda la información de un sitio web.
 - ❑ Ventajas
 - ❑ Proporcionar información actualizada
 - ❑ Facilitar la realización de búsquedas
 - ❑ Disminuir los costes de mantenimiento
 - ❑ Implementar sistemas de control de acceso
 - ❑ Almacenar preferencias de los usuarios
-

Bases de datos en la Web

- Esquema básico de un sitio web soportado por bases de datos:



Instrucciones MySQL

- ❑ El usuario y contraseña estándar para conectar con el gestor de mysql es:

```
mysql -u root -p
```

```
password: root
```
 - ❑ Incluir método para restablecer la contraseña de root.
 - ❑ SHOW DATABASES;
 - ❑ SHOW TABLES;
 - ❑ USE base_datos;
 - ❑ DROP TABLE nombre_tabla;
-

Instrucciones MySQL

```
❑ CREATE TABLE EMPLE (  
    EMP_NO  INTEGER(4) PRIMARY KEY,  
    APELLIDO VARCHAR(10),  
    OFICIO  VARCHAR(10),  
    DIR     INTEGER(4),  
    FECHA_ALT DATE,  
    SALARIO INTEGER(10),  
    COMISION INTEGER(10),  
    DEPT_NO INTEGER(2) REFERENCES DEPART(DEPT_NO)  
    )engine=innodb;
```

Instrucciones MySQL

❑ INSERT INTO EMPLE VALUES(1122,
 'DIAZ','VENDEDOR',2200,'10-02-1990',1200,300,10);

PHP con acceso a BD

La interacción entre el lenguaje de programación de una aplicación web y el sistema de gestión de bases de datos es el elemento fundamental para lograr que una *aplicación web* sea realmente *dinámica*.

PHP con acceso a MySQL

- ❑ MySQL es el servidor de bases de datos preferido para el entorno PHP-Apache.
 - ❑ Trataremos los principios elementales de la programación PHP con acceso a bases de datos MySQL.
-

PHP con acceso a MySQL:

librería mysqli

- ❑ Usar una base de datos MySQL mediante PHP es muy fácil:
 - ❑ Establecer la conexión con el servidor de bases de datos:
mysqli_connect().
 - ❑ Seleccionar la base de datos:
mysqli_select_db(base_de_datos)
 - ❑ Realizar una consulta: *mysqli_query(consulta)*
 - ❑ Procesar los resultados de la consulta
 - ❑ Cerrar la conexión: `mysqli_close(conexión)`
-

PHP con acceso a MySQL: librería mysqli

- ❑ Desde la versión PHP5, MySQL no está activado de modo predeterminado. Se debe incluir como una extensión la biblioteca `php_mysql.dll` (Windows) o `php_mysql.so` (Linux).
-

API de PHP para MySQL

- ❑ Se recomienda utilizar la API mejorada *mysqli* a partir de PHP5. (La API de PHP mysql no funciona bien con MySQL posterior a la versión 4.1.2)
 - ❑ En general, cada función *mysql* tiene su función paralela en *mysqli*, aunque ésta incorpora muchas más funcionalidades.
-

API de PHP para MySQL

- ❑ La API mejorada *mysqli* incorpora también soporte para el modelo de objetos.
- ❑ Cada función de la extensión mysqli tiene su paralela como método de la **clase mysqli**:

```
$con = new mysqli("localhost","root","root");
$con->select_db("periodico");
$strsql="SELECT * FROM Noticias";
if ($resu=$con->query($strsql)){
    $fila=$resu->fetch_row();
    foreach($fila as $valor)
        echo $valor;
    $resu->close();
    $con->close();
}
```

Operaciones básicas con MySQL

- ☐ Apertura de una conexión
 - ☐ Creación
 - ☐ Selección
 - ☐ Creación de una tabla
 - ☐ Inserción de filas
 - ☐ Lectura de filas
 - ☐ Eliminación de filas
 - ☐ Actualización de filas
 - ☐ Cierre de conexión
-

Apertura de una conexión: librería mysql

- ❑ PHP suministra las funciones incorporadas para establecer la conexión con la base de datos:

`obj mysqli_connect(pservidor, pusuario, pcontraseña)`

- ❑ Los tres parámetros son opcionales. Si no se informan se supone 'localhost', el propietario del proceso y contraseña vacía.
-

Apertura de una conexión: librería mysql

- ❑ Otra función de la extensión mysql para establecer la conexión con la base de datos estableciendo parámetros adicionales en la conexión:

`bool mysql_real_connect($conexion, pservidor, pusuario, pcontraseña, bd, puerto, socket, flags)`

- ❑ Todos los parámetros son opcionales salvo el indicador de la conexión.
 - ❑ Esta función no está implementada en la extensión *mysql*.
-

Cierre de una conexión: librería mysql

- ❑ Devuelve el código de error de la última conexión intentada a un servidor MySQL

`int mysqli_connect_errno()`

`int mysqli_connect_error()`

- ❑ La conexión se cierra con la función

`bool mysqli_close($conexion)`

- ❑ Libera la memoria asociada con un resultado

`void mysqli_free_result($resultado)`

- ❑ Reciben como parámetro el identificador de conexión previamente abierta
-

Selección de una bd

- ❑ PHP suministra una función que permite seleccionar una bd creada para poder utilizarla:

`bool mysqli_select_db($conexion, pnombreBD)`

[Ejemplo 142(2)]

Ejecución de consultas

- ❑ PHP suministra una función que permite ejecutar sentencias de SQL de cualquier tipo:

`mixed mysqli_query($conexión, $consulta, modo_resultado)`

- ❑ El parámetro `$conexión` es opcional en la primera. Toma la última conexión abierta.
- ❑ `modo_resultado` es opcional y puede contener las constantes `MYSQLI_USE_RESULT` o `MYSQLI_STORE_RESULT` (predeterminado)
- ❑ Por motivos de seguridad, esta instrucción envía, en cada ejecución, una única sentencia SQL al gestor de BD.
- ❑ La cadena de la consulta no debería terminar con un punto y coma. Los datos insertados en la consulta deberían estar correctamente escapados.

[Ejemplo 141]

Escapar todas las consultas

- ❑ Escapar todas las cadenas de caracteres antes de insertarlas en la bd:

String mysqli_real_escape_string(obj conexión, string cadena)

- ❑ Los caracteres codificados son NUL (ASCII 0), \n, \r, \, ', ", y Control-Z.
 - ❑ El conjunto de caracteres debe ser establecido a nivel del servidor, o con la función mysqli_set_charset() de la API para que afecte a **mysqli_real_escape_string()**
-

Escapar todas las consultas

```
<?php
    $enlace = mysqli_connect("localhost", "mi_usuario", "mi_contraseña", "world");
    if (mysqli_connect_errno()) {
        printf("Falló la conexión: %s\n", mysqli_connect_error());
        exit();
    }
    mysqli_query($enlace, "CREATE TEMPORARY TABLE miCiudad LIKE City");

    $ciudad = "s Hertogenbosch";

    /* esta consulta fallará debido a que no escapa $ciudad */
    if (!mysqli_query($enlace, "INSERT into miCiudad (Name) VALUES ('$ciudad')")) {
        printf("Error: %s\n", mysqli_sqlstate($enlace));
    }
    $ciudad = mysqli_real_escape_string($enlace, $ciudad);

    /* esta consulta con $ciudad escapada funcionará */
    if (mysqli_query($enlace, "INSERT into miCiudad (Name) VALUES ('$ciudad')")) {
        printf("%d fila insertada.\n", mysqli_affected_rows($enlace));
    }
    mysqli_close($enlace);
```


Acceso al conjunto de datos de una consulta

- ❑ La sentencia SELECT efectúa la consulta pero no devuelve directamente los resultados.
- ❑ Debemos recurrir a otras funciones para navegar por ese conjunto de resultados:

- ❑ Devuelve la cantidad de filas en el conjunto de resultados:

int mysqli_num_rows(\$resultado)

- ❑ Devuelve un array con los valores de una fila del conjunto de resultados o falso.

mixed mysqli_fetch_array(\$resultado, tipo_resultado)

- ❑ El tipo de resultado indica cómo se entrega la matriz: MYSQL_ASSOC, MYSQL_NUM o MYSQL_BOTH (defecto)

Acceso al conjunto de datos de una consulta

- ❑ Devuelve un array numérico con una fila del resultado de la consulta, o FALSE si ya no quedan filas por extraer.

mixed mysqli_fetch_row(\$resultado)

- ❑ Devuelve un array asociativo con una fila del resultado de la consulta, o FALSE si ya no quedan filas por extraer.

mixed mysqli_fetch_assoc(\$resultado)

- ❑ La cantidad de columnas se obtiene con la función:

int mysqli_field_count(\$conexion)

- ❑ Retorna el próximo campo del resultSet, o FALSE si ya no quedan columnas por extraer.

object mysqli_fetch_field(\$resultado)

Ejecución de consultas

- ❑ Otra función de la extensión mysqli para la ejecución de múltiples sentencias:

bool mysqli_multi_query(\$conexión, \$consulta)

- ❑ \$consulta es una o más sentencias SQL separadas por punto y coma.
- ❑ La obtención del primer conjunto de resultados se consigue con la función

result mysqli_store_result() / result mysqli_use_result()

y los siguientes conjuntos de resultados con las funciones

bool mysql_more_result() y mysqli_next_result() .

- ❑ Se debe liberar la memoria reservada para el conjunto de resultados con la función **void mysqli_free_result(\$resultado)**
-

Ejecución de consultas

```
<?php
$con = mysqli_connect("localhost", "root", "root", "jardineria");
if (mysqli_connect_errno()) {
    printf("Conexión fallida: %s\n", mysqli_connect_error());
    exit();
}
$query = "SELECT CURRENT_USER();";
$query .= "SELECT distinct NombreCliente from Clientes natural Join Pedidos
          WHERE FechaEntrega > FechaEsperada";
if (mysqli_multi_query($con, $query)) {
    do {
        /* almacenar primer juego de resultados */
        if ($result = mysqli_store_result($con)) {
            // recorre los diferentes conjuntos Result
            while ($row = mysqli_fetch_row($result)) {
                // recorre las filas de cada conjunto Result
                printf("%s\n", $row[0]);
            }
            mysqli_free_result($result);
        } /* mostrar divisor */
        if (mysqli_more_results($con) {
            printf("-----\n");
        }
    } while (mysqli_next_result($con));
}
mysqli_close($con);
?>
```


Creación de una tabla

- ❑ Para crear una tabla en MySQL desde PHP se utiliza la cadena SQL con la definición de la tabla y la función que envía la sentencia a la bd activa o indicada.

`mixed mysqli_query($conexión, $consulta, modo_resultado)`

[Ejemplo 143]

La importancia del *charset* y el *cotejamiento*

- ❑ La mayoría de los datos que se almacenan en las tablas son textos.
 - ❑ En la nomenclatura de MySQL se denomina *charset* al método de codificación y *cotejamiento* (*collation*) al criterio utilizado para realizar comparaciones entre textos.
 - ❑ MySQL permite establecer el juego de caracteres a nivel de servidor, de base de datos, de tabla y de campo.
-

Elegir el conjunto de caracteres

- ❑ La siguiente función establece qué codificación utilizará PHP para los datos que envíe a través de la conexión con la bd.

bool mysqli_set_charset (\$conexión , \$charset)

- ❑ Obtener el conjunto de caracteres predeterminado:

string mysqli_character_set_name (\$conexión)

- ❑ Puedes consultar los juegos de caracteres que reconoce MySQL en:

<http://dev.mysql.com/doc/refman/5.6/en/charsetcharsets.html>

Elegir el conjunto de caracteres

- ❑ Codifica un string ISO-8859-1 a UTF-8

string utf8_encode (string \$data)

- ❑ Convierte una cadena con los caracteres codificados ISO-8859-1 con UTF-8 a un sencillo byte ISO-8859-1

string utf8_decode (string \$data)

Inserción de filas en la tabla

- ❑ Utilizamos la función `mysqli_query()` que envía la sentencia INSERT a la bd activa o indicada.
- ❑ Escapar todas las cadenas de caracteres antes de insertarlas en la bd:

String mysqli_real_escape_string(obj conexión, string cadena)

Ejecución de transacciones

- ❑ Ejecuta la transacción actual

bool mysqli_commit(\$con)

- ❑ Deshace la transacción actual

bool mysqli_rollback(\$con)

- ❑ Activa o desactiva el modo 'auto-commit' en consultas para la conexión a la base de datos.

bool mysqli_autocommit(\$con, bool \$modo)

Funciones de utilidad

- ❑ Recupera información sobre la última consulta ejecutada

string mysqli_info(\$con)

- ❑ Devuelve el error SQLSTATE de la operación previa de MySQL

string mysqli_sqlstate(\$con)

- ❑ Ejecución de procedimientos almacenados
-

Funciones de utilidad

- ❑ Las bases de datos de MySQL soportan procedimientos almacenados.
- ❑ Las aplicaciones pueden llamar y ejecutar el procedimiento almacenado
- ❑ La sentencia de SQL **CALL** se usa para ejecutar un procedimiento almacenado.

```
if (!$mysqli->multi_query("CALL p())) {  
    echo "Falló CALL: (" . $mysqli->errno . ") " . $mysqli->error;  
}
```

Seguridad

- ❑ La seguridad en las bases de datos es un asunto crucial. Deberíamos aplicar siempre los siguientes principios generales:
 - ❑ Validar *los datos* para asegurarnos de que antes de salir de PHP son *del tipo* que se espera en *MySQL*.
 - ❑ Sanear *los datos* para asegurarnos de que antes de salir de PHP tienen el *formato* que espera *MySQL* (por ejemplo, las fechas en formato YYYY-MM-DD).
 - ❑ Limitar la longitud de los datos. No permitas que se introduzcan textos demasiado largos en su base de datos si no es estrictamente necesario; podrían ser códigos maliciosos.
-

Seguridad

- ❑ Escapar todas las cadenas de caracteres con `mysqli_real_escape_string()`.
 - ❑ Usar un usuario para la conexión de PHP que posea los privilegios mínimos para realizar las operaciones que necesitemos.
 - ❑ No almacenar datos que realmente no sea necesario. Por ejemplo, no almacenes un número de tarjeta de crédito si no vas a usarlo en el futuro.
 - ❑ No almacenes los datos redundantemente. Si en el futuro es necesario modificarlos te costará más mantener tu base de datos actualizada y no provocar una corrupción de información.
-

Seguridad

- ❑ Si sólo necesitas saber si un dato se ha escrito correctamente (por ejemplo una contraseña), pero no necesitas saber exactamente cuál es ese dato, almacénalo hashado en la base de datos (por ejemplo con MD5).
- ❑ Si tienes que almacenar datos sensibles, considera la posibilidad de encriptarlos previamente en PHP mediante la librería Mcrypt.

<http://php.net/manual/es/ref.mcrypt.php>

- ❑ Cifradores Mcrypt soportados por PHP

<http://php.net/manual/es/mcrypt.ciphers.php>

- ❑ Modos de cifrado Mcrypt

<http://php.net/manual/es/mcrypt.constants.php>

Seguridad: ejemplo

```
<?php
class Encrypter {

    private static $Key = "dublin";

    public static function encrypt ($input) {
        $output = base64_encode(mcrypt_encrypt(MCRYPT_RIJNDAEL_256, md5(Encrypter::$Key), $input,
MCRYPT_MODE_CBC, md5(md5(Encrypter::$Key))));
        return $output;
    }
    public static function decrypt ($input) {
        $output = rtrim(mcrypt_decrypt(MCRYPT_RIJNDAEL_256, md5(Encrypter::$Key), base64_decode($input),
MCRYPT_MODE_CBC, md5(md5(Encrypter::$Key))), "\0");
        return $output;
    }
}

$texto = "Esto es una prueba de encriptacion de diferente longitud";

// Encriptamos el texto
$texto_encriptado = Encrypter::encrypt($texto);
echo "Texto encriptado: ".$texto_encriptado." longitud: ".strlen($texto_encriptado)."<br>";

// Desencriptamos el texto
$texto_original = Encrypter::decrypt($texto_encriptado);
echo "Texto desencriptado: ".$texto_original."<br>";

if ($texto == $texto_original) echo 'Encriptación / Desencriptación realizada correctamente.';
?>
```


Seguridad

- ❑ Otras funciones PHP para encriptado:

 - `string crypt (string $str [, string $salt])`

 - `string md5 (string $str [, bool $raw_output = false])`

 - `string sha1 (string $str [, bool $raw_output = false])`

- ❑ Enlace a temas de seguridad en PHP

 - <http://phpsec.org/>

Lista de funciones MySQL para usar con PHP6

- ❑ Funciones que se pueden utilizar desde PHP para comunicarse con un servidor MySQL.
 - ❑ Se recomienda utilizar la extensión mysqli dado que ofrece más funcionalidades.
-

Funciones de conexión

- ❑ `mysql_change_user()` / `mysqli_change_user()`
 - ❑ `mysql_close()` / `mysqli_close()`
 - ❑ `mysql_connect()` / `mysqli_connect()`
 - ❑ `mysql_connect_errno()` / `mysqli_connect_errno()`
 - ❑ `mysqli_init()`
 - ❑ `mysqli_init()`
 - ❑ `mysqli_options()`
 - ❑ `mysqli_ping()`
 - ❑ `mysqli_real_connect()`
 - ❑ `mysql_select_db()` / `mysqli_select_db()`
 - ❑ `mysqli_set_option()`
 - ❑ `mysqli_ssl_set()`
-

Funciones de consulta

- ❑ `mysqli_execute()`
 - ❑ `mysqli_multi_query()`
 - ❑ `mysql_query()` / `mysqli_query()`
 - ❑ `mysqli_real_query()`
-

Funciones de conjuntos de resultados

- ❑ `mysql_data_seek()` / `mysqli_data_seek()`
 - ❑ `mysql_fetch_array()` / `mysqli_fetch_array()`
 - ❑ `mysql_fetch_assoc()` / `mysqli_fetch_assoc()`
 - ❑ `mysql_fetch_field()` / `mysqli_fetch_field()`
 - ❑ `mysqli_fetch_fields()`
 - ❑ `mysqli_fetch_field_direct()`
 - ❑ `mysql_fetch_lengths()` / `mysqli_fetch_lengths()`
 - ❑ `mysql_fetch_object()` / `mysqli_fetch_object()`
 - ❑ `mysql_fetch_row()` / `mysqli_fetch_row()`
-

Funciones de conjuntos de resultados

- ❑ `mysqli_field_count()`
 - ❑ `mysqli_field_len()`
 - ❑ `mysqli_field_name()`
 - ❑ `mysql_field_seek()` / `mysqli_field_seek()`
 - ❑ `mysql_field_table()`
 - ❑ `mysql_field_type()`
 - ❑ `mysqli_field_tell()`
 - ❑ `mysql_free_result()` / `mysqli_free_result()`
 - ❑ `mysql_info()` / `mysqli_info()`
-

Funciones de conjuntos de resultados

- ❑ `mysql_insert_id()` / `mysqli_insert_id()`
 - ❑ `mysqli_more_results()`
 - ❑ `mysqli_next_results()`
 - ❑ `mysql_num_fields()` / `mysqli_num_fields()`
 - ❑ `mysql_num_rows()` / `mysqli_num_rows()`
 - ❑ `mysqli_store_result()`
 - ❑ `mysqli_use_result()`
-

Funciones de control de errores

- ❑ `mysqli_connect_errno()`
 - ❑ `mysqli_connect_error()`
 - ❑ `mysql_errno()` / `mysqli_errno()`
 - ❑ `mysql_error()` / `mysqli_error()`
 - ❑ `mysqli_sqlstate()`
-

Funciones informativas

- ❑ `mysql_affected_rows()` / `mysqli_affected_rows()`
 - ❑ `mysql_get_client_info()` / `mysqli_get_client_info()`
 - ❑ `mysqli_get_client_version()`
 - ❑ `mysql_get_host_info()` / `mysqli_get_host_info()`
 - ❑ `mysql_get_proto_info()` / `mysqli_get_proto_info()`
 - ❑ `mysql_get_server_info()` / `mysqli_get_server_info()`
 - ❑ `mysqli_get_server_version()`
 - ❑ `mysql_info()` / `mysqli_info()`
 - ❑ `mysql_insert_id()` / `mysqli_insert_id()`
 - ❑ `mysqli_warning_count()`
-

Funciones de control de transacciones

- ❑ `mysqli_autocommit()`
 - ❑ `mysqli_commit()`
 - ❑ `mysqli_rollback()`
-

Funciones de sentencias preparadas

- ❑ `mysqli_param_count()`
 - ❑ `mysqli_prepare()`
 - ❑ `mysqli_stmt_affected_rows()`
 - ❑ `mysqli_stmt_bind_param()`
 - ❑ `mysqli_stmt_bind_result()`
 - ❑ `mysqli_stmt_close()`
 - ❑ `mysqli_stmt_data_seek()`
 - ❑ `mysqli_stmt_errno()`
 - ❑ `mysqli_stmt_error()`
-

Funciones de sentencias preparadas

- ☐ `mysqli_stmt_execute()`
 - ☐ `mysqli_stmt_fetch()`
 - ☐ `mysqli_stmt_free_result()`
 - ☐ `mysqli_stmt_init()`
 - ☐ `mysqli_stmt_num_rows()`
 - ☐ `mysqli_stmt_param_count()`
 - ☐ `mysqli_stmt_prepare()`
 - ☐ `mysqli_stmt_result_metadata()`
 - ☐ `mysqli_stmt_sqlstate()`
 - ☐ `mysqli_stmt_store_result()`
-

Funciones auxiliares

- ❑ `mysqli_character_set_name()`
 - ❑ `mysql_create_db()`
 - ❑ `mysql_drop_db()`
 - ❑ `mysqli_dump_debug_info()`
 - ❑ `mysqli_kill()`
 - ❑ `mysql_list_dbs()`
 - ❑ `mysql_list_fields()`
 - ❑ `mysql_list_processes()`
 - ❑ `mysql_list_tables()`
-

Funciones auxiliares

- ❑ `mysqli_ping()`
 - ❑ `mysql_real_escape_string()` / `mysqli_real_escape_string()`
 - ❑ `mysqli_report()`
 - ❑ `mysql_stat()` / `mysqli_stat()`
 - ❑ `mysql_thread_id()` / `mysqli_thread_id()`
 - ❑ `mysqli_thread_safe()`
-

Clase mysqli: métodos

- ❑ Apertura de una conexión con el gestor MySQL:

```
$mysqli = new mysqli($host, $username, $password, $db,  
                    $port, $socket)
```

- ❑ Selección de una Base de Datos

```
bool $mysqli -> select_db (string $name)
```

- ❑ Cierre de una conexión:

```
bool $mysqli -> close ()
```

- ❑ Liberar la memoria asociada a un resultado:

```
void $result -> free_result (void)
```

```
void $result -> close (void)
```

Clase mysqli: métodos

- ❑ Obtención del código de error de la última conexión:

```
String $mysqli -> connect_errno;
```

```
String $mysqli -> connect_error;
```

```
Mysqli->affected_rows()
```

Clase mysqli: métodos

- ❑ Ejecución de consultas:

`mixed $mysqli -> query (string $query, int $modo_resultado)`

- ❑ Escapar todas las consultas:

`string $mysqli -> escape_string (string $query)`

`string $mysqli -> real_escape_string (string $query)`

- ❑ Obtener una array con cada fila resultado de la consulta:

`mixed $result -> fetch_array (int $tipo_resultado)`

- ❑ Cantidad de filas en el conjunto de resultados de una consulta:

`int $result -> num_rows;`

Clase mysqli: métodos

- ❑ Obtener una array numérico con cada resultado de la consulta:

`mixed $result -> fetch_row (void)`

- ❑ Obtener una array asociativo con cada resultado de la consulta:

`array $result -> fetch_assoc (void)`

- ❑ Cantidad de columnas en el conjunto de resultados de una consulta:

`int $mysqli -> field_count;`

- ❑ Devuelve el próximo campo del resultSet:

`object $result -> fetch_field(void);`

Clase mysqli: ejemplo

```
<?php
$con = new mysqli("mysql.hostinger.es","root","root");
$con->select_db("pizzeria");
$strsql="SELECT * FROM ingredientes";
if ($resu=$con->query($strsql)){
    while ($fila=$resu->fetch_row()){
        foreach($fila as $valor)
            echo $valor." ";
        echo "<br>";
    }
    $resu->close();
    $con->close();
}
```

Clase mysqli: métodos

- ❑ Ejecución de consultas múltiples:

`bool $mysqli -> multi_query (string $query)`

- ❑ Obtención del primer conjunto de resultados:

`mysqli_result $mysqli -> store_result(void)`

`mysqli_result $mysqli -> use_result(void)`

- ❑ Verificar la existencia de más conjuntos de resultados:

`bool $mysqli -> more_result(void)`

`bool $mysqli -> next_result(void)`

Clase mysqli: métodos

- ❑ Establecer el conjunto de caracteres en la conexión con la bd:

`bool $mysqli -> set_charset (string $charset);`

- ❑ Obtener el conjunto de caracteres predeterminado:

`string $mysqli -> character_set_name (void);`

- ❑ Ejecución de la transacción actual:

`bool $mysqli -> commit(int $flags, string $name);`

- ❑ Deshacer la transacción actual:

`bool $mysqli -> rollback(int $flags, string $name);`

Clase mysqli: métodos

- ❑ Desactivar el modo autocommit:

`bool $mysqli -> autocommit(bool $modo);`

- ❑ Recuperar información sobre la última consulta ejecutada:

`string $mysqli -> info;`

- ❑ Devolver el código de error de MySQL:

`string $mysqli -> sqlstate;`

- ❑ Ejecución de procedimientos almacenados:

`bool $mysqli -> query("CALL (p1)");`

Función mail()

bool **mail** (string \$destino ,string \$asunto , string \$message
[,string \$additional_headers
[, string \$additional_parameters]])

- ❑ Función útil para enviar correos electrónicos sin necesidad de disponer de un gestor de correo.
 - ❑ Los primeros tres parámetros son obligatorios y representan el *destinatario*, el *asunto* y el *cuerpo* del mensaje que componen el correo.
 - ❑ *additional headers* se usa para añadir cabeceras extra (From, Cc y Bcc). Las cabeceras múltiples adicionales deberían separarse con un CRLF(\r\n).
-

Función mail()

- ❑ Las cabeceras (*headers*) son los elementos que aparecen en la parte superior de un correo. Normalmente:
 - ❑ **To** : Lista de destinatarios separados por coma
 - ❑ **From** : Dirección desde la que enviamos
 - ❑ **Reply-To** : Dirección de reply
 - ❑ **Return-Path** : Como Reply-To. Requerido por algunos clientes de correo
 - ❑ **Subject** : Asunto del correo
 - ❑ **CC** (Carbon Copy): Lista de destinatarios separados por comas
 - ❑ **BCC** (Blind Carbon Copy) : Lista de destinatarios ocultos
 - ❑ **MIME-Version**: 1.0
 - ❑ **Content-Type**: text/plain
 - ❑ **Content-Type**: multipart/mixed; boundary="frontera"
 - ❑ **Content-Transfer-Encoding**: 7bit
-

Función mail()

- ❑ TO y SUBJECT se indican en los dos primeros parámetros
- ❑ El resto de cabeceras se indican en el cuarto parámetro (opcional)
- ❑ Cómo leer la cabecera completa de un correo:

<https://support.google.com/mail/answer/29436?hl=es>

<https://siliconhosting.com/kb/questions/312/Entender+una+cabecera+de+correo>

- ❑ MIME:

[http://es.wikipedia.org/wiki/Multipurpose Internet Mail Extensions](http://es.wikipedia.org/wiki/Multipurpose_Internet_Mail_Extensions)

Función mail()

- ❑ Configuración para utilizar el servidor de correo de gmail en los envíos de correo con mail() (Windows)
 - ❑ Ficheros a modificar en el servidor (xampp):
`php.ini` , `sendmail.ini`
 - ❑ En el fichero **`sendmail.ini`** actualizamos las siguientes directivas:

```
smtp_server = smtp.gmail.com  
smtp_port = 587  
Error_logfile = error.log  
debug_logfile = debug.log  
auth_username = tucuenta@gmail.com  
auth_password= tu contraseña de gmail  
force_sender = tucuenta@gmail.com
```

Función mail()

- ❑ Configuración para utilizar el servidor de correo de gmail en los envíos de correo con mail() en Windows

- ❑ Ficheros a modificar en el servidor (xampp):

`php.ini` , `sendmail.ini`

- ❑ En **php.ini** modificamos las siguientes directivas:

```
smtp = smtp.gmail.com  
smtp_port = 587  
sendmail_from = tucuenta@gmail.com  
; sendmail_path = "C:\xampp\mailtodisk\mailtodisk.exe"
```

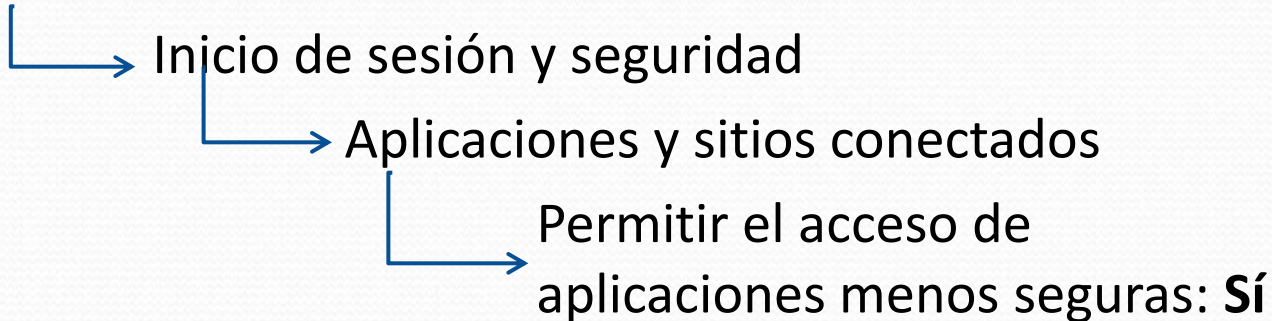
- ❑ Enlace:

<http://php.codeindepth.com/php-sending-mail/>

Función mail()

- ❑ Cambia la configuración de tu cuenta de gmail para permitir el acceso de aplicaciones menos seguras:
 - ❑ Dentro de tu cuenta de gmail:

Mi cuenta



Ejemplo función mail()

```
mail("carmen@domenico.es",  
    "Una prueba definitiva", "Bienvenid@ a PHP",  
    "From: Administrador <admin@domenico.es>  
    Reply-To: carmen@domenico.es  
    Cc: antonio@domenico.es  
    Bcc: jorge@domenico.es" . phpversion())
```

Ejemplo función mail()

```
<?php
```

```
$to = "destinatario@somewhere.com " . ",";
```

```
$to.="destinatario2@somewhere.com";
```

```
$subject = "Asunto";
```

```
$message = "Hello, how are you?";
```

```
$headers = "From: micorreo@here.com\r\n";
```

```
$headers .= "Reply-To: miotrocorreo@here.com\r\n";
```

```
$headers .= "Return-Path: micorreo@here.com\r\n";
```

```
$headers .= "CC: otrodestinatario@noplac.com\r\n";
```

```
$headers .= "BCC: destinatarioculto@special.com\r\n";
```

```
if ( mail($to, $subject, $message, $headers) ) {
```

```
    echo "Correo enviado!";
```

```
    } else
```

```
    echo "Correo no enviado!";
```

```
?>
```


Función mail()

- ❑ Es posible también enviar correos HTML con **mail()**.
- ❑ Para enviar un correo HTML mail, la cabecera Content-type debe fijarse así:

```
$cabeceras = 'MIME-Version: 1.0' . "\r\n";
```

```
$cabeceras .= 'Content-type: text/html; charset=iso-88591' . "\r\n";
```

[Ejemplo correo]

Importar datos en MySQL desde ficheros.CSV

1. Convertir el formato PDF al formato Excel utilizando herramientas de conversión como:
 - ❑ WonderShare PDF Converter
 - ❑ WonderShare Data Recovery
2. Obtener el fichero.csv desde Microsoft Excel:
Archivo -> Guardar como .csv
3. Ejecutar el código de importación en MySQL:

```
load data infile 'alumnos.csv'  
into table alumnos ←  
fields terminated by ';' ←  
lines terminated by '\n'  
ignore 1 rows;
```

La tabla debe existir
con la misma estructura
que las líneas del
fichero.csv

Generar documentos .pdf

❑ 5 librerías para generar PDF en PHP:

- ❑ FPDF
- ❑ TCPDF
- ❑ R&OS
- ❑ FPDFI
- ❑ DOMPDF

<http://blog.tednologia.com/5-librerias-para-generar-pdf-con-php/>