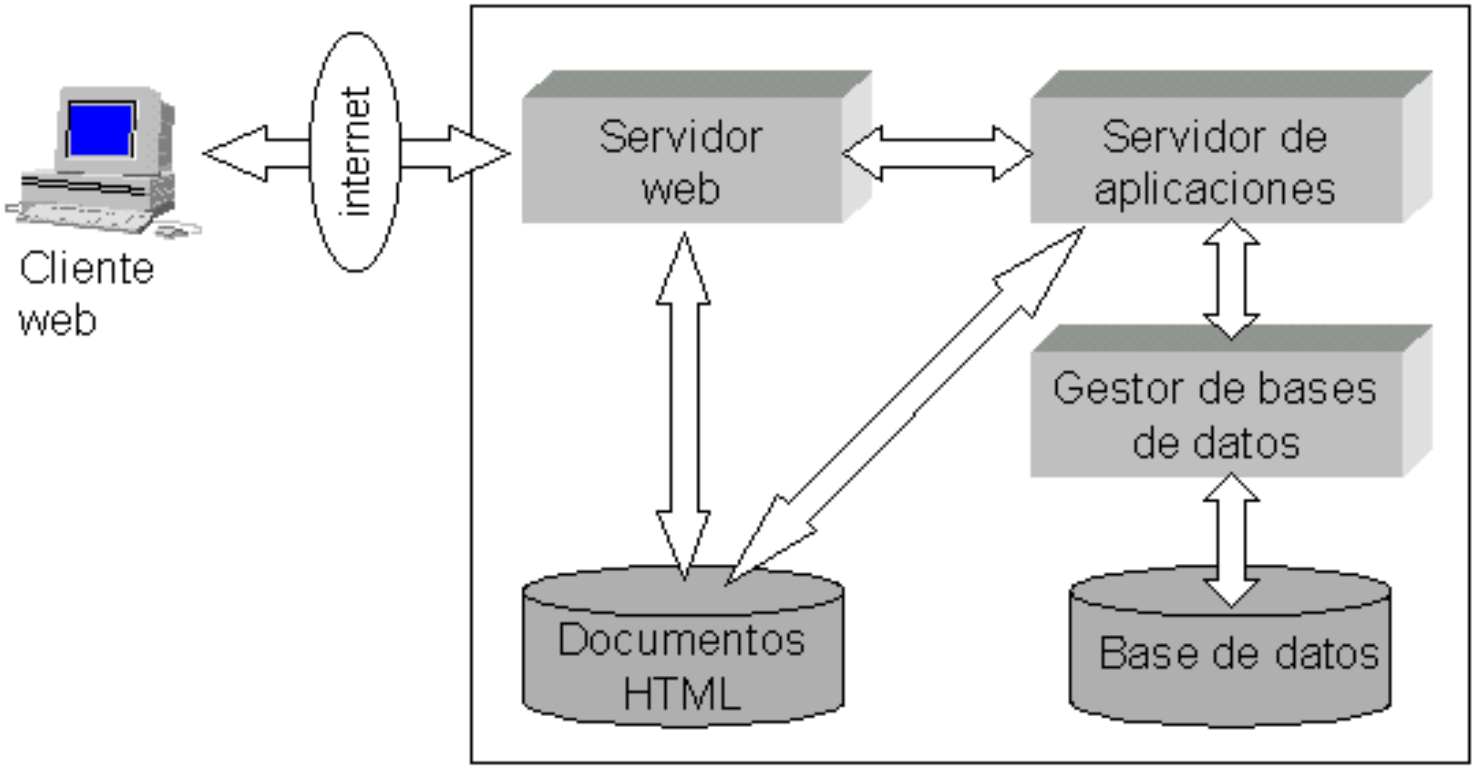
Curso de PHP

–Acceso a bases de datos

–MySQL en PHP

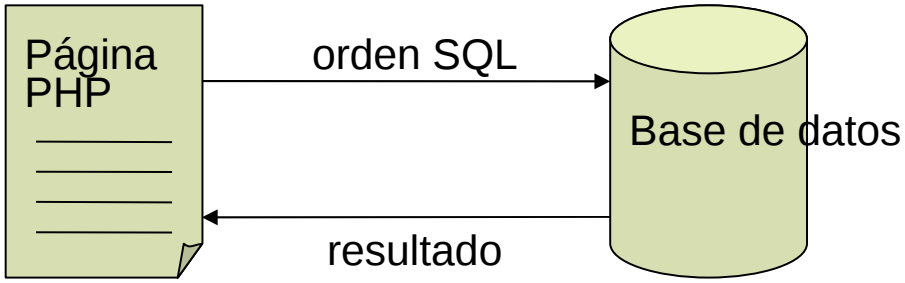
Bases de datos en la Web

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



Lenguaje SQL

- SQL (*Structured Query Language*) es el lenguaje que se utiliza para comunicarse con la base de datos
- Procedimiento de comunicación con la base de datos:



Funciones de PHP para el acceso a bases de datos MySQL

- Los pasos para acceder desde PHP a una base de datos son los siguientes:
  - Conectar con el servidor de bases de datos
  - Seleccionar una base de datos
  - Enviar la instrucción SQL a la base de datos
  - Obtener y procesar los resultados
  - Cerrar la conexión con el servidor de bases de datos

Acceso a bases de datos MySQL

- Las funciones concretas de MySQL que realizan estas operaciones son:
  - Conectar con el servidor de bases de datos:
    - **mysql_connect() o mysql_pconnect()**
  - Seleccionar una base de datos:
    - **mysql_select_db()**
  - Enviar la instrucción SQL a la base de datos:
    - **mysql_query()**
  - Obtener y procesar los resultados:
    - **mysql_num_rows() y mysql_fetch_array()**
  - Cerrar la conexión con el servidor de bases de datos:
    - **mysql_close() o mysql_pclose()**

Acceso a bases de datos MySQL

- Conectar con el servidor de bases de datos: **mysql_connect()**
  - Devuelve un identificador de la conexión en caso de éxito y false en caso contrario
  -
- Sintaxis:

```
$conexion = mysql_connect (servidor, username,
password);
```

- Ejemplo:

```
$conexion = mysql_connect (“localhost”, “cursophp”, “”)
   or die (“No se puede conectar con el servidor”);
$conexion = mysql_connect (“localhost”, “cursophp-ad”,
“php.hph”)
   or die (“No se puede conectar con el servidor”);
```

Acceso a bases de datos MySQL

- Seleccionar una base de datos: **mysql_select_db()**
  - Devuelve true en caso de éxito y false en caso contrario
- Sintaxis:

```
mysql_select_db (database);
```

- Ejemplo:

```
mysql_select_db (“lindavista”)
   or die (“No se puede seleccionar la base de datos”);
```

Acceso a bases de datos MySQL

- Enviar la instrucción SQL a la base de datos: **mysql_query()**
  - Devuelve un identificador o true (dependiendo de la instrucción) si la instrucción se ejecuta correctamente y false en caso contrario
- Sintaxis:

```
$consulta = mysql_query (instrucción, $conexion);
```

- Ejemplo:

```
$consulta = mysql_query (“select * from noticias”, $conexion)
   or die (“Fallo en la consulta”);
```

Acceso a bases de datos MySQL

- Obtener y procesar los resultados: **mysql_num_rows(), mysql_fetch_array()**
  - En el caso de que la instrucción enviada produzca unos resultados, mysql_query() devuelve las filas de la tabla afectadas por la instrucción
  - mysql_num_rows() devuelve el número de filas afectadas
  - Para obtener las distintas filas del resultado se utiliza la función mysql_fetch_array(), que obtiene una fila del resultado en un array asociativo cada vez que se invoca
- Sintaxis:

```
$nfilas = mysql_num_rows ($consulta);
$fila = mysql_fetch_array ($consulta);
```

Acceso a bases de datos MySQL

- Ejemplo:

noticias				
1	Título 1	Texto 1	ofertas	05/02/2004
2	Título 2	Texto 2	promociones	05/02/2004
3	Título 3	Texto 3	promociones	04/02/2004
4	Título 4	Texto 4	costas	01/02/2004
5	Título 5	Texto 5	promociones	31/01/2004

•Instrucción:

```
•select * from noticias where
categoria=“promociones”
```

Acceso a bases de datos MySQL

- Ejemplo:

noticias				
1	Título 1	Texto 1	ofertas	05/02/2004
2	Título 2	Texto 2	promociones	05/02/2004
3	Título 3	Texto 3	promociones	04/02/2004
4	Título 4	Texto 4	costas	01/02/2004
5	Título 5	Texto 5	promociones	31/01/2004

`$consulta`

`$nfilas=3`

•Instrucción:

•`select * from noticias where categoria=“promociones”`

Acceso a bases de datos MySQL

- Obtención de las filas:

```
$nfilas = mysql_num_rows ($consulta);
if ($nfilas > 0)
{
   for ($i=0; $i<$nfilas; $i++)
   {
      $fila = mysql_fetch_array ($consulta);
      procesar fila i-ésima de los resultados
   }
}
```

Acceso a bases de datos MySQL

- Obtener los resultados: **mysql_num_rows(), mysql_fetch_array()**
  - Para acceder a un campo determinado de una fila se usa la siguiente sintaxis:
  - `$fila[“nombre_campo”]// por ser un array asociativo`
  - `$fila[i] // i=índice del campo desde 0`
  - •
- Ejemplo:

```
for ($i=0; $i<$nfilas; $i++)
{
   $fila = mysql_fetch_array ($consulta);
   echo “Título: “ . $fila[“titulo”];
   echo “Fecha: “ . $fila[”fecha”];
}
```

Acceso a bases de datos MySQL

- Cerrar la conexión con el servidor de bases de datos: **mysql_close()**
- Sintaxis:

```
mysql_close ($conexion);
```

- Ejemplo

```
mysql_close ($conexion);
```

Acceso a BD unificado
Clase ConexionGenerica

```
class ConexionGenerica {
    var $link;

    function connect($servidor, $usuario, $clave) {
        echo "<h1>El m&eacute;todo <i>connect</i> ",
             "no est&aacute; implementado en la ",
             "clase <i>".get_class($this)."</i></h1>";
       return FALSE;
    }

    function close() {}
    function select_db($base_datos) {}
    function query($query) {}
    function fetch_array($resultado) {}
    function free_result($resultado) {}
}
```

Acceso a BD unificado
Clase ConexionMySQL

```
require ("ConexionGenerica.inc");

class ConexionMySQL extends ConexionGenerica {

    function connect($servidor="localhost",
                     $usuario="nobody", $clave="") {
        return $this->link = mysql_connect($servidor,
                                           $usuario, $clave);
    }

    function close() {}
    function select_db($base_datos) {}
    function query($query) {}
    function fetch_array($resultado) {}
    function free_result($resultado) {}
}
```

Acceso a BD unificado
Uso

```
// Fichero CreaConexion.inc
require ("ConexionMySQL.inc");
$conexion = new ConexionMySQL();
```

```
require ("CreaConexion.inc");

$conexion->connect("localhost", "nobody");
$conexion->select_db("prueba", $link);

$asignaturas = $conexion->query("SELECT codigo, nombre,
                                        descripcion,
                                        creditos, tipo
                                  FROM asignatura");
```

Ejemplos avanzados

Consulta avanzada de tablas

- Objetivo:
  - Mostrar los resultados de la consulta **divididos en bloques de un número determinado de elementos** (por ejemplo, de 5 en 5)
- Requisitos:
  - Recuperar un número limitado de elementos de la tabla
  - Implementar un mecanismo de navegación que permita avanzar al siguiente o volver al anterior bloque de elementos

Consulta avanzada de tablas

- Para recuperar un número fijo de elementos de una tabla se utiliza la opción **LIMIT** de la orden SELECT. Así, por ejemplo,
- `SELECT * from noticias LIMIT 0, 5`
-
- recupera los 5 primeros elementos de la tabla. Y en general,
-
- `SELECT * from noticias LIMIT $comienzo, $num`
-
- recupera $num elementos a partir de la posición $comienzo

Consulta avanzada de tablas

- La variable $num tendrá un valor constante (en este caso 5), mientras que la variable $comienzo se incrementará o decrementará en 5 unidades al pasar a la página siguiente o anterior
- Para ello se pasará la variable como parámetro en el enlace asociado al botón correspondiente
- Por ejemplo, el código para el botón siguiente será:
-
- ```
  "<A HREF='$PHP_SELF?comienzo=" . ($comienzo + $num) . "'>Siguiente</A>“
  ```
-
- Habrá que comprobar previamente que el nuevo valor de comienzo es válido, es decir, que se encuentra dentro de los límites de la tabla devuelta por la consulta
  ```

Consulta avanzada de tablas (2)

- Objetivo:
  - Mostrar los resultados de una consulta de manera que se puedan **filtrar en función del valor de una determinada columna** de la tabla
- Requisitos:
  - Recuperar de una tabla los elementos que cumplan una determinada condición
  - Permitir seleccionar un valor de entre los valores posibles de una columna

Consulta avanzada de tablas (2)

- Para recuperar los elementos de una tabla que cumplen una condición se utiliza la opción **WHERE** de la orden SELECT. Por ejemplo,
-
-
```
SELECT * from noticias WHERE categoria='ofertas'
```
-
- recupera las noticias cuya categoría tiene el valor “ofertas”. Y en general,
-
-
```
SELECT * from noticias WHERE categoria
='$categoria'
```
-
- recupera las noticias cuya categoría tiene el valor dado por la variable $categoria
-
- La variable $categoria se obtendrá de un elemento SELECT

Consulta avanzada de tablas (3)

- Objetivo:
  - Modificar el ejercicio anterior para que los resultados se actualicen de forma automática al seleccionar un nuevo valor para la columna
- Requisitos:
  - Obtener automáticamente los valores de una columna de tipo enumerado a partir de la tabla
  - Utilizar código JavaScript para detectar un cambio en la opción seleccionada de un elemento de tipo SELECT

Consulta avanzada de tablas (3)

- Crearemos el código HTML dinámicamente generando las diferentes categorías de noticias.
- Para ello es preciso generar las opciones del elemento SELECT de forma automática
- También es conveniente que la opción seleccionada aparezca marcada por defecto al actualizar la página
- Ejecutaremos una JavaScript asociada al elemento SELECT a través del evento ONCHANGE, que se activa cuando se modifica la opción seleccionada:

```
<SELECT NAME=“categoria"
onchange=”document.forms.form1.submit()”>
   <OPTION VALUE=“costas">costas
   <OPTION VALUE=“ofertas" SELECTED>ofertas
   <OPTION VALUE=“promociones">promociones
</SELECT>
```

Consulta avanzada de tablas (3)

- Para obtener la lista de diferentes categorías de noticias utilizaremos:
-
- `SELECT DISTINCT categoria FROM noticias ORDER BY categoria`
-
-

Consulta avanzada de tablas (3)

- El código que nos muestra la lista de noticias según la categoria escogida tendría els siguiente aspecto:
-
- `$categoria = $_REQUEST[“categoria”];`
- `// Realizariamos la siguiente consulta en la base de datos:`
- `// SELECT * FROM noticias WHERE categoria='$categoria'`
-
- `// Mostraríamos los datos`
-
-