



UNIVERSIDAD
DE GRANADA



Tecnologías Web

Grado en Ingeniería Informática

Tema 3 – Programación en el lado del servidor PHP y conexión con Bases de Datos

Este documento está protegido por la Ley
de Propiedad Intelectual ([Real Decreto Ley
1/1996 de 12 de abril](#)).
Queda expresamente prohibido su uso o
distribución sin autorización del autor.

© Javier Martínez Baena
jbaena@ugr.es

Departamento de Ciencias de la
Computación e Inteligencia Artificial
<http://decsai.ugr.es>



UNIVERSIDAD
DE GRANADA

Tecnologías Web

Grado en Ingeniería Informática

Programación en el lado del servidor

1. El lenguaje PHP
2. PHP y aplicaciones web
3. PHP y conexión con BBDD



1. Introducción
2. Conexión, consultas y organización
3. Página de resultados
4. Edición y borrado de registros
5. Saneamiento de datos de formularios
6. Inserción y búsqueda de registros
7. Inyección de código SQL
8. Consultas preparadas

4. Consideraciones finales



PHP y acceso a Bases de Datos
Introducción

Objetivo

Acceso a BBDD:

- Obtener información
- Almacenar información

Kevin Yank. "PHP & MySQL. From novice to Ninja (5ed)". SitePoint. 2012

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 3

PHP y acceso a Bases de Datos
Popularidad de diferentes sistemas de BBDD

DB-ENGINES

	Rank			DBMS	Database Model	Score		
	Apr 2018	Mar 2018	Apr 2017			Apr 2018	Mar 2018	Apr 2017
	1.	1.	1.			Oracle +	Relational DBMS	1289.79
2.	2.	2.	MySQL +	Relational DBMS	1226.40	-2.46	-138.22	
3.	3.	3.	Microsoft SQL Server +	Relational DBMS	1095.51	-9.28	-109.26	
4.	4.	4.	PostgreSQL +	Relational DBMS	395.47	-3.88	+33.69	
5.	5.	5.	MongoDB +	Document store	341.41	+0.89	+15.98	
6.	6.	6.	DB2 +	Relational DBMS	188.95	+2.28	+2.29	
7.	7.	7.	Microsoft Access	Relational DBMS	132.22	+0.27	+4.04	
8.	↑ 9.	↑ 11.	Elasticsearch +	Search engine	131.36	+2.81	+25.69	
9.	↓ 8.	9.	Redis +	Key-value store	130.11	-1.12	+15.75	
10.	10.	8.	Cassandra +	Wide column store	119.09	-4.40	-7.10	

342 systems in ranking, April 2018

Nota: Ranking basado en menciones en buscadores, Google Trends, discusiones en Stack Overflow o DBA Stack Exchange, ofertas de trabajo, perfiles en LinkedIn, menciones en Twitter, ... (No está basado en instalaciones en sistemas web)

© April 2018, DB-Engines.com

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 4

**Extensiones específicas****API específica para cada tipo de DBMS soportado**

dBase	IBM DB2	Informix	Ingres
MongoDB	MySQL	Oracle	Paradox
PostgreSQL	SQLite	... +20	

- Dependientes del DBMS
- Más eficientes

Extensiones abstractas**API genérica que abstrae los detalles de cada DBMS**

ODBC	Open DataBase Connectivity
PDO	PHP Data Objects
...	

- Independientes del DBMS
- Menos eficientes
- Necesitan drivers de cada DBMS
- Facilitan la programación

**Comparación de las opciones de la API de MySQL para PHP**

	Extensión mysqli de PHP	PDO (Usando el driver PDO MySQL y el Driver Nativo MySQL)	Extensión MySQL de PHP
Versión de PHP en que se introdujo	5.0	5.0	Antes de 3.0
Incluido con PHP 5.x	Sí	Sí	Sí
Estado de desarrollo de MySQL	Desarrollo activo	Desarrollo activo, desde PHP 5.3	Sólo se le mantiene
Recomendado por MySQL para nuevos proyectos	Sí - opción recomendada	Sí	No
Soporte para juegos de caracteres	Sí	Sí	No
Soporte para Declaraciones Preparadas en el lado del servidor	Sí	Sí	No
Soporte para Declaraciones Preparadas en el lado del cliente	No	Sí	No
Soporte para Procedimientos Almacenados	Sí	Sí	No
Soporte para Declaraciones Múltiples	Sí	Mayormente	No
Soporte para todas las funcionalidades de MySQL 4.1+	Sí	Mayormente	No
	Orientado a objetos Procedural	Orientado a objetos	

<http://php.net/manual/es/mysqli.overview.php>



UNIVERSIDAD
DE GRANADA

Tecnologías Web

Grado en Ingeniería Informática

Programación en el lado del servidor

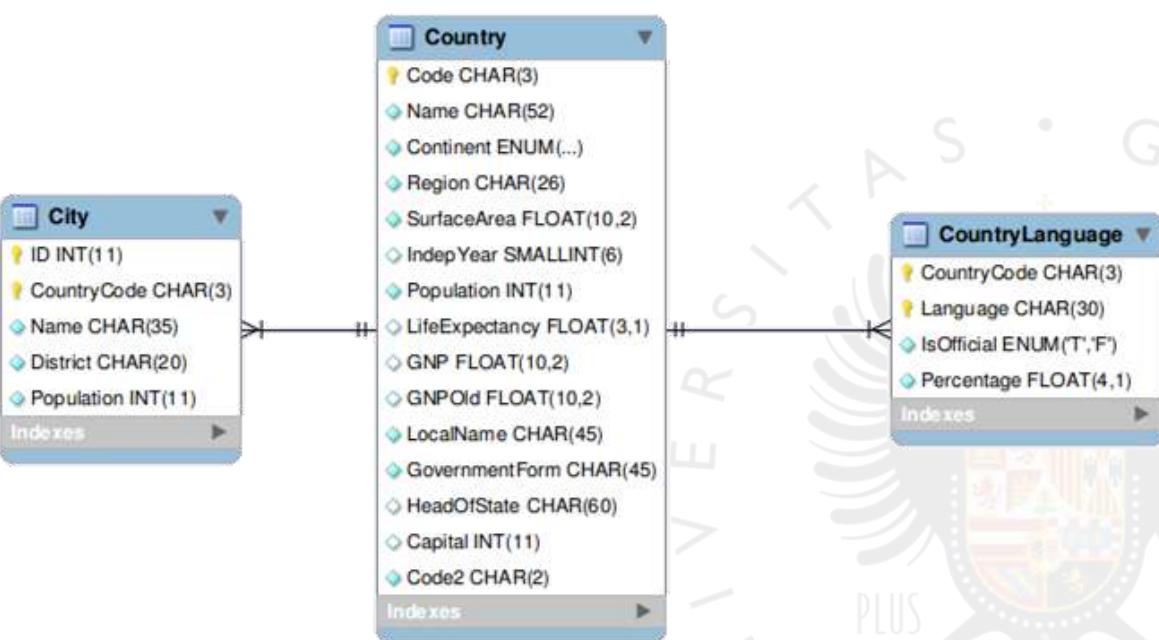
- 1. El lenguaje PHP
- 2. PHP y aplicaciones web
- 3. PHP y conexión con BBDD
 - 1. Introducción
 - 2. Conexión, consultas y organización
 - 3. Paginación de resultados
 - 4. Edición y borrado de registros
 - 5. Saneamiento de datos de formularios
 - 6. Inserción y búsqueda de registros
 - 7. Inyección de código SQL
 - 8. Consultas preparadas
- 4. Consideraciones finales

PHP y acceso a Bases de Datos

BBDD de ejemplo

BBDD de ejemplo de MySQL: "world"

<https://dev.mysql.com/doc/index-other.html>



```

    erDiagram
        Country ||--o|| City : " "
        Country ||--o|| CountryLanguage : " "
    }
  
```

The ER diagram illustrates the relationships between three tables:

- Country** table:
 - Attributes: Code CHAR(3), Name CHAR(52), Continent ENUM(...), Region CHAR(26), SurfaceArea FLOAT(10,2), IndepYear SMALLINT(6), Population INT(11), LifeExpectancy FLOAT(3,1), GNP FLOAT(10,2), GNPOld FLOAT(10,2), LocalName CHAR(45), GovernmentForm CHAR(45), HeadOfState CHAR(60), Capital INT(11), Code2 CHAR(2).
 - Relationships: Has many **City** (multiplicity 1..n) via a foreign key.
 - Relationships: Has many **CountryLanguage** (multiplicity 1..n) via a foreign key.
- City** table:
 - Attributes: ID INT(11), CountryCode CHAR(3), Name CHAR(35), District CHAR(20), Population INT(11).
 - Relationships: Belongs to one **Country** (multiplicity 1..n) via a foreign key.
- CountryLanguage** table:
 - Attributes: CountryCode CHAR(3), Language CHAR(30), IsOfficial ENUM('T','F'), Percentage FLOAT(4,1).
 - Relationships: Belongs to one **Country** (multiplicity 1..n) via a foreign key.

PHP y acceso a Bases de Datos
Ciclo de trabajo con BBDD

```

graph LR
    A[Conexión] --> B[Consulta]
    B --> C[Desconexión]
  
```

Conexión con la BBDD

```
$db = mysqli_connect(host, usuario, clave, bbdd)
```

```

// Conexión a la BBDD
$db = mysqli_connect("localhost", "tweb", "dejameentrar", "world");
if ($db) {
    echo "<p>Conexión con éxito</p>";
} else {
    echo "<p>Error de conexión</p>";
    echo "<p>Código: ".mysqli_connect_errno()."</p>";
    echo "<p>Mensaje: ".mysqli_connect_error()."</p>";
    die("Adiós");
}
// Establecer la codificación de los datos almacenados ("collation")
mysqli_set_charset($db, "utf8");
  
```

Desconexión de la BBDD

```
mysqli_close($db)
```

(si se omite, la desconexión es automática al finalizar el script PHP)

PHP y acceso a Bases de Datos
Consulta a la BBDD y obtención de resultados

Conexión y desconexión de la BBDD

Cada conexión a la BBDD es un proceso lento
Opción 1.- Conectar y desconectar en cada consulta
Opción 2.- Conectar al principio del script y desconectar al final

Apache puede tener uno o más procesos para interpretar código PHP

Cada proceso PHP puede mantener su propio "connection pool"

No hay garantía de que una petición HTTP sea atendida por el mismo proceso PHP

PHP y acceso a Bases de Datos
Ciclo de trabajo con BBDD

```

graph LR
    A[Conexión] --> B[Consulta]
    B --> C[Desconexión]
    C --- X((X))
  
```

Forzar la conexión persistente desde PHP
Al finalizar el script la conexión se mantiene en caché y no se cierra

```
$db = mysqli_connect("p:localhost", "tweb", "dejameentrar", "world");
```

- PHP detecta si la conexión proviene de un mismo user, password, host, port, socket y base de datos y aprovecha la conexión abierta
- mysqli_close no tiene efecto

Inconveniente: si una aplicación termina inesperadamente ...

- ... puede dejar tablas bloqueadas
- ... puede estar en medio de una transacción
- ... puede tener consultas preparadas
- ... puede tener tablas temporales

Puede ocurrir que una aplicación no termine en un tiempo razonable

Al reutilizar la conexión:

- El nuevo cliente conserva ese estado
- MySQLi puede realizar tareas de limpieza pero ¡cuidado!

No se recomienda <http://php.net/manual/en/mysqli.persistconns.php>

PHP y acceso a Bases de Datos
Consultas a la BBDD

Consulta a la BBDD

```
$res = mysqli_query($db, consulta)
```

La consulta puede ser de cualquier tipo:
SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ...

```
// Consulta a la BBDD
$res = mysqli_query($db,
    "SELECT name,district,population FROM city WHERE countrycode='ESP'");
if ($res) { // Si no hay error
    if (mysqli_num_rows($res)>0) { // Si hay alguna tupla de respuesta
        echo "<table><tr><th>Nombre</th><th>Comunidad</th><th>Población</th></tr>";
        while ($tupla=mysqli_fetch_array($res)) // Para cada tupla ...
            echo "<tr><td>{$tupla['name']}

```



Conexión con éxito

Nombre	Comunidad	Población
Madrid	Madrid	2879052
Barcelona	Katalonia	1503451
Valencia	Valencia	739412
Sevilla	Andalusia	701927
Zaragoza	Aragonía	603367
Málaga	Andalusia	530553
Bilbao	Baskimaa	357589
Las Palmas de Gran Canaria	Canary Islands	354757
Murcia	Murcia	353504
Palma de Mallorca	Balears	326993
Valladolid	Castilla and León	319998
Córdoba	Andalusia	311708



Recorriendo la consulta en orden arbitrario

```
mysqli_data_seek($res,$posicion)
```

```
// Consulta a la BBDD
$res = mysqli_query($db,
    "SELECT name,district,population FROM city WHERE countrycode='ESP'");
if ($res) { // Si no hay error
    if (mysqli_num_rows($res)>0) { // Si hay alguna tupla de respuesta
        echo "<table><tr><th>Nombre</th><th>Comunidad</th><th>Población</th></tr>";
        for ($i=0; $i<mysqli_num_rows($res); $i++) { // Para cada tupla ...
            mysqli_data_seek($res,$i);
            $tupla=mysqli_fetch_array($res);
            echo "<tr><td>{$tupla['name']}

```



Credenciales de acceso a la BBDD

Se definen en un fichero independiente para facilitar la modularización

```
<?php // Fichero dbcredenciales.php
DEFINE('DB_HOST', 'localhost');
DEFINE('DB_USER', 'tweb');
DEFINE('DB_PASSWD', 'dejameentrar');
DEFINE('DB_DATABASE', 'world');
?>

// Otros scripts PHP
require_once('dbcredenciales.php');
$db = mysqli_connect(DB_HOST, DB_USER, DB_PASSWD, DB_DATABASE);
...
```

Seguridad

- El fichero de credenciales debe estar:
 - Fuera del sistema de ficheros exportado por el servidor web
 - ... o, en todo caso, protegido para que no sea accesible vía web
- La conexión con la BBDD no está cifrada
 - Es habitual alojar en un mismo host tanto el servidor web como el DBMS (se evita que clave y datos viajen en plano por la red)



Conexiones cifradas

Toda la información (credenciales y consultas) se transmite cifrada

1. `mysqli_init()`
Inicializa MySQLi y prepara conexión (no establece conexión)
2. `mysqli_ssl_set()`
Establece opciones de cifrado SSL
3. `mysqli_real_connect()`
Abre una conexión preparada previamente con `mysqli_init()`

Configuración del servidor MySQL

Para dar acceso solo a localhost, añadir a my.cnf

```
bind-address=127.0.0.1
```



Organización del código

Separar en distintos ficheros funciones con cometidos diferentes

Código auxiliar	db.php	// Funciones de conexión y consultas
	dbcredenciales.php	// Credenciales de acceso a la BBDD
	db_backup.php	// Funciones para gestión de backup
Páginas del sitio	html.php	// Funciones genéricas de maquetación
	htmlforms.php	// Generación HTML de formularios, listados, etc
	estilo.css	// Reglas de estilo del sitio
Páginas del sitio	listado.php	// Listado
	listado_paginado.php	// Listado con paginación
	listado_paginadoBotones.php	// Listado con paginación y botones
	addCiudad.php	// Añadir nueva ciudad
	buscarCiudad.php	// Búsqueda de datos
	editarCiudad.php	// Edición de datos
	backup.php	// Crear un backup de la BBDD
	restore.php	// Restaurar una BBDD



Organización del código

Aspecto de una página de la aplicación (Listado de datos)

```

Común {
    <?php
    require('html.php');           // Maquetado de página
    require('db.php');             // Operaciones con BBDD

    // ***** Inicio de la página
    htmlStart('Listado completo','Listado');

    if (!is_string($db=DB_conexion())) { // Si la conexión es correcta
        $ciudades=DB_getCiudades($db);   // Obtener listado de ciudades
        VIEW_listadoCiudades($ciudades); // Mostrar listado
        DB_desconexion($db);            // Desconectar de la BBDD
    } else                           // Si hay error en conexión
        msgError($db);

    // ***** Fin de la página
    htmlEnd();
?>

```

PHP y acceso a Bases de Datos
Aspecto general de la web

GeoWeb: geografía política
Una web con datos geográficos

Listado Listado Página Listado Página (botones) Búsqueda Nueva ciudad Backup Restore

Ciudad	Comunidad	Población
A Coruña (La Coruña)	Galicia	243402
Albacete	Kastilia-La Mancha	147527
Alcalá de Henares	Madrid	164463
Alcorcón	Madrid	142048
Algeciras	Andalusia	103106
Alicante [Alacant]	Valencia	272432
Almería	Andalusia	169027

Valencia	Valencia	139412
Valladolid	Castilla and León	319998
Vigo	Galicia	283670
Vitoria-Gasteiz	Baskimaa	217154
Zaragoza	Aragonia	603367
[San Cristóbal de] la Laguna	Canary Islands	127945

© Tecnologías Web

PHP y acceso a Bases de Datos
Aspecto general de la web

GeoWeb: geografía política
Una web con datos geográficos

Listado Página Listado Página (botones) Búsqueda Nueva ciudad Backup

Ciudad	Comunidad	Población
A Coruña (La Coruña)	Galicia	243402
Albacete	Kastilia-La Mancha	147527
Alcalá de Henares	Madrid	164463
Alcorcón	Madrid	142048
Algeciras	Andalusia	103106
Alicante [Alacant]	Valencia	272432
Almería	Andalusia	169027
Badajoz	Extremadura	136613
Badalona	Katalonia	209635
Barakaldo	Baskimaa	98212
Barcelona	Katalonia	1503451
Bilbao	Baskimaa	357589
Burgos	Castilla and León	162802
Cádiz	Andalusia	142449
Cartagena	Murcia	177709
Castellón de la Plana [Castell	Valencia	139712

```
.listado table tr:nth-child(even) {
    background-color: #B7D7E8;
}

.listado table tr:nth-child(odd) {
    background-color: #CFE0E8;
}

/* Encabezado */
.listado table tr:nth-child(1) {
    background-color: #87BDD8;
    border-bottom: solid 2px #667292;
}
```



Organización del código: html.php

```

<?php
require_once('htmlforms.php');      // Inclusión de formularios

// $titulo: título de la página      // $activo: enlace activo del menú
function htmlStart($titulo,$activo='') {
    __htmlInicio($titulo);
    __htmlEncabezado($activo);
    __htmlContenidosIni();
}

function htmlEnd() {
    __htmlContenidosFin();
    __htmlPiepagina();
    __htmlFin();
}

function htmlNav($clase,$menu,$activo='') {
    echo "<nav class='$clase'>";
    foreach ($menu as $elem)
        echo "<a ".($activo==$elem['texto'])?"class='activo' ":"")."
                           "href='{$elem['url']}'>".$elem['texto']."

```



Organización del código: html.php

```

// Cabecera de página web
function __htmlInicio($titulo) {
echo <<< HTML
<!DOCTYPE html> <html> <head>
<meta charset="utf-8" />
<link rel="stylesheet" href="estilo.css" />
<title>{$titulo}</title> </head> <body>
HTML;
}

function __htmlEncabezado($activo) {      // Encabezado
echo <<< HTML
<div class='encabezado'>
    <h1>GeoWeb: geografía política</h1>
    <h2>Una web con datos geográficos</h2>
</div>
HTML;
htmlNav('menu',[['texto'=>'Listado', 'url'=>'listado.php'],
              ['texto'=>'Listado Paginado', 'url'=>'listado_paginator.php'],
              ['texto'=>'Listado Paginado (botones)', 'url'=>'listado_paginatorBotones.php'],
              ['texto'=>'Búsqueda', 'url'=>'buscarCiudad.php'],
              ['texto'=>'Nueva ciudad', 'url'=>'addCiudad.php'],
              ['texto'=>'Backup', 'url'=>'backup.php'],
              ['texto'=>'Restore', 'url'=>'restore.php']],$activo);
}

```



Organización del código: html.php

```
// Contenidos FIN
function __htmlContenidosFin() {
echo '</div>';
}

// Pie de página
function __htmlPiepagina() {
echo <<< HTML
<div class='piepagina'>
    <h1>&copy; Tecnologías Web</h1>
</div>
HTML;
}

// Cierre de página web
function __htmlFin() {
echo '</body></html>';
}
```



Organización del código: htmlforms.php

```
// Mostrar tabla con listado de ciudades
// $datos: array asociativo
// cada elemento es un registro (name,district,population)
function VIEW_listadoCiudades($datos) {
echo <<< HTML
<div class='listado'> <table> <tr>
<th>Ciudad</th> <th>Comunidad</th> <th>Población</th> </tr>
HTML;

foreach ($datos as $v) {
echo '<tr>';
echo "<td class='ciu_nombre'>{$v['name']}

```



Organización del código: db.php

```

require_once('dbcredenciales.php');

// Conexión a la BBDD
// Devuelve:
// Si conexión ok → resource asociado a la BBDD
// Si error de conexión → un string con mensaje de error
function DB_conexion() {
    $db = mysqli_connect(DB_HOST,DB_USER,DB_PASSWD,DB_DATABASE);
    if (!$db)
        return "Error de conexión a la base de datos (".mysqli_connect_errno()."
                ") : ".mysqli_connect_error();
    // Establecer la codificación de los datos almacenados ("collation")
    mysqli_set_charset($db,"utf8");
    return $db;
}

// Desconexión de la BBDD
function DB_desconexion($db) {
    mysqli_close($db);
}

```



Organización del código: db.php

```

// Consulta para obtener listado de ciudades
function DB_getCiudades($db) {
    $res = mysqli_query($db, "SELECT id,name,district,population FROM city
                                WHERE countrycode='ESP' ORDER BY name");
    if ($res) {
        // Si no hay error
        if (mysqli_num_rows($res)>0) { // Si hay alguna tupla de respuesta
            $tabla = mysqli_fetch_all($res,MYSQLI_ASSOC);
        } else { // No hay resultados para la consulta
            $tabla = [];
        }
        mysqli_free_result($res); // Liberar memoria de la consulta
    } else { // Error en la consulta
        $tabla = false;
    }
    return $tabla;
}

```



PHP y acceso a Bases de Datos

Consulta a la BBDD y comprobación del resultado

Consulta a la BBDD

\$result = mysqli_query(\$db,\$query);

Realiza la consulta \$query y devuelve un objeto con la información de la consulta o false si no se ha podido hacer

\$err = mysqli_errno(\$db)

Devuelve el código (int) de error de la última consulta hecha 0 si no hay error

\$msg = mysqli_error(\$db)

Devuelve la descripción (string) del error de la última consulta hecha o cadena vacía si no hay error

\$msg = mysqli_sqlstate(\$db)

Devuelve un string con el código de error SQLSTATE de la última consulta (<http://dev.mysql.com/doc/mysql/en/error-handling.html>)

\$n = mysqli_affected_rows(\$db)

Devuelve el número de filas afectadas o recuperadas en la última consulta, 0 si no hubo resultados, -1 si hubo error

PHP y acceso a Bases de Datos

Consulta a la BBDD y obtención de resultados



Consulta a la BBDD: operaciones sobre la consulta

\$n = mysqli_num_fields(\$result);

Devuelve el número de campos del resultado

\$n = mysqli_num_rows(\$result)

Devuelve el número de tuplas/filas del resultado

\$arr = mysqli_fetch_all(\$result,\$tipo)

Devuelve, en forma de array, todas las tuplas

\$tipo==MYSQL_NUM Devuelve array enumerado (*por defecto*)

\$tipo==MYSQL_ASSOC Devuelve array asociativo

\$tipo==MYSQL_BOTH Devuelve array asociativo y enumerado

\$msg = mysqli_fetch_array(\$result,\$tipo)

Devuelve array asociativo

\$msg = mysqli_fetch_assoc(\$result)

Devuelve array enumerado

\$msg = mysqli_fetch_row(\$result)

Devuelve, en forma de array, la siguiente tupla de la consulta o NULL si no hay más filas.

\$tipo==MYSQL_NUM Devuelve array enumerado

\$tipo==MYSQL_ASSOC Devuelve array asociativo

\$tipo==MYSQL_BOTH Devuelve array asociativo y enumerado (*defecto*)

mysqli_free_result(\$result)

Libera la memoria ocupada por la consulta



UNIVERSIDAD
DE GRANADA

Tecnologías Web

Grado en Ingeniería Informática

Programación en el lado del servidor

»

- 1. El lenguaje PHP**
- 2. PHP y aplicaciones web**
- 3. PHP y conexión con BBDD**
 - 1. Introducción**
 - 2. Conexión, consultas y organización**
 - 3. Página de resultados**
 - 4. Edición y borrado de registros**
 - 5. Saneamiento de datos de formularios**
 - 6. Inserción y búsqueda de registros**
 - 7. Inyección de código SQL**
 - 8. Consultas preparadas**
- 4. Consideraciones finales**

PHP y acceso a Bases de Datos

Página de resultados

Página de resultados

Uso de parámetros `$_GET` para indicar el rango (permite personalizar al usuario, permite almacenar las búsquedas)

localhost/tw/bbdd/listado_paginado.php?primero=0&items=10

GeoWeb: geografía política
Una web con datos geográficos

localhost/tw/bbdd/listado_paginado.php?primero=4&items=7

Ciudad	Comunidad	Población
Badajoz	Extremadura	136613
Barcelona	Katalonia	1503451
Castellón de la Plana [Castell	Valencia	139712
Córdoba	Andalucía	311708
Elche [Elx]	Valencia	193174
Fuenlabrada	Madrid	171173
Getafe	Madrid	145371



Paginación de los resultados: listado_paginado.php

Uso de parámetros \$_GET para indicar el rango

```

// **** Argumentos GET de la página
// primero: Primer item a visualizar
// items : cuantos items incluir (<=0 para ver todos)
if (!isset($_GET['items']))
    $numitems = 10; // Valor por defecto
else if (!is_numeric($_GET['items']) || $_GET['items']<1)
    $numitems = 0; // Para mostrar todos los items
else
    $numitems = $_GET['items'];

if ($numitems==0)
    $primero=0; // Ver todos los items
else {
    $primero = isset($_GET['primero']) ? $_GET['primero'] : 0;
    if (!is_numeric($primero) || $primero<0)
        $primero=0;
}
...

```

Paginación de los resultados: listado_paginado.php

Uso de parámetros \$_GET para indicar el rango

```

...
// **** Contenido // Obtener listado de ciudades
if (!is_string($db=DB_conexion())) {

    // Obtener listado de ciudades
    $numciudades=DB_getNumCiudades($db);
    $ciudades=DB_getCiudades($db,$primero,$numitems);

    // Mostrar listado
    if ($ciudades!==false)
        VIEW_listadoCiudades($ciudades);
    else
        $info[] = 'Ha habido un error en la consulta a la BBDD';

    // Barra de paginación
...

```



PHP y acceso a Bases de Datos

Paginación de resultados

Paginación de los resultados: dp.php

Cláusulas LIMIT y OFFSET de SQL

```
// Consulta para obtener listado de ciudades
function DB_getCiudades($db,$primero=0,$numitems=0) {
    if ($numitems<=0) // Listarlos todos
        $rango='';
    else
        $rango = 'LIMIT '.(int)($numitems).' OFFSET '.abs($primero);

    // Consulta a la BBDD
    $res = mysqli_query($db, "SELECT id,name,district,population FROM city
                                WHERE countrycode='ESP' ORDER BY name $rango");
    if ($res) {
        if (mysqli_num_rows($res)>0) // Si hay alguna tupla de respuesta
            $tabla = mysqli_fetch_all($res,MYSQLI_ASSOC);
        else // No hay resultados para la consulta
            $tabla = [];
        mysqli_free_result($res); // Liberar memoria de la consulta
    } else
        $tabla = false;
    return $tabla;
}
```

PHP y acceso a Bases de Datos

Paginación de resultados: barra de navegación

Paginación de los resultados

Añadir una barra de navegación (anterior, siguiente, ...)

localhost/tw/bbdd/listado_paginado.php?primero=0&items=5

Ciudad	Comunidad	Población
A Coruña (La Coruña)	Galicia	243402
Albacete	Kastilia-La Mancha	147527
Alcalá de Henares	Madrid	164463
Alcorcón	Madrid	142048
Algeciras	Andalusia	103106

Primera Anterior Siguiente Última

localhost/tw/bbdd/listado_paginado.php?primero=5&items=5

Ciudad	Comunidad	Población
Alicante [Alacant]	Valencia	272432
Almería	Andalusia	169027
Badajoz	Extremadura	136613
Badalona	Katalonia	209635
Barakaldo	Baskimaa	98212

Primera Anterior Siguiente Última

PHP y acceso a Bases de Datos
Paginación de resultados: barra de navegación

Paginación de los resultados: listado_paginado.php
Añadir una barra de navegación (anterior, siguiente, ...)

```

...
$numciudades=DB_getNumCiudades($db);

...
// Barra de paginación
if ($numitems>0) {
    $ultima = $numciudades - ($numciudades%$numitems);
    $anterior = $numitems>$primero ? 0 : ($primero-$numitems);
    $siguiente = ($primero+$numitems)>$numciudades ? $ultima : ($primero+$numitems);
    htmlNav('paginador',[ 
        ['texto'=>'Primera', 'url'=>'?primero=0&items='.$numitems],
        ['texto'=>'Anterior', 'url'=>'?primero='.$anterior.'&items='.$numitems],
        ['texto'=>'Siguiente', 'url'=>'?primero='.$siguiente.'&items='.$numitems],
        ['texto'=>'Última', 'url'=>'?primero='.$ultima.'&items='.$numitems]]); 
}

```

PHP y acceso a Bases de Datos
Paginación de resultados: barra de navegación

Paginación de los resultados: db.php
Añadir una barra de navegación (anterior, siguiente, ...)

```

// Consulta para obtener el número de ciudades
function DB_getNumCiudades($db) {
    $res = mysqli_query($db, "SELECT COUNT(*) FROM city WHERE countrycode='ESP'");
    $num = mysqli_fetch_row($res)[0];
    mysqli_free_result($res);
    return $num;
}

```



UNIVERSIDAD
DE GRANADA

Tecnologías Web

Grado en Ingeniería Informática

Programación en el lado del servidor

- 1. El lenguaje PHP
- 2. PHP y aplicaciones web
- 3. PHP y conexión con BBDD
 - 1. Introducción
 - 2. Conexión, consultas y organización
 - 3. Página de resultados
 - 4. Edición y borrado de registros
 - 5. Saneamiento de datos de formularios
 - 6. Inserción y búsqueda de registros
 - 7. Inyección de código SQL
 - 8. Consultas preparadas
- 4. Consideraciones finales

»

PHP y acceso a Bases de Datos

Botones de edición

Inclusión de botones en el listado
Facilitan la edición, borrado, etc.

Ciudad	Comunidad	Población	Acción
A Coruña (La Coruña)	Galicia	243402	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Albacete	Kastilia-La Mancha	147527	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Alcalá de Henares	Madrid	164463	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Alcorcón	Madrid	142048	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Algeciras	Andalusia	103106	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Alicante [Alacant]	Valencia	272432	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Almería			
Badajoz			
Badalona			
Barakaldo			

Edite los datos:

Nombre:

Comunidad:

Población:

Confirme borrado de esta ciudad:

Nombre:

Comunidad:

Población:

Primeras Anterior Siguientes Últimas

PHP y acceso a Bases de Datos
Botones de edición

Inclusión de botones en el listado
Añadir formulario en columna de botones (en htmlforms.php)

```
VIEW_listadoCiudadesBotones($ciudades, 'editarCiudad.php');

function VIEW_listadoCiudadesBotones($datos, $accion) {
echo <<< HTML
<div class='listado'> <table> <tr>
<th>Ciudad</th> <th>Comunidad</th> <th>Población</th> <th>Acción</th></tr>
HTML;
foreach ($datos as $v) {
    echo "<tr><td class='ciu_nombre'>{$v['name']}


Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 39


```

PHP y acceso a Bases de Datos
Botones de edición

Inclusión de botones en el listado
Página de edición/borrado de ciudades (editarCiudad.php)

The diagram illustrates the workflow for editing and deleting cities. It starts with a list of cities (A Coruña, Galicia, 243402) with 'Editar' and 'Borrar' buttons. Clicking 'Editar' leads to an 'Edit' form with fields for Nombre (Gijón), Comunidad (Asturias), and Población (267980). Clicking 'Borrar' leads to a 'Delete Confirmation' form with fields for Nombre (Mataró), Comunidad (Katalonia), and Población (104095). Finally, two success messages are shown: 'La ciudad Gijón ha sido actualizada' and 'La ciudad Mataró ha sido borrada'.

1. Inicializar página
2. Comprobar si hay datos \$_POST con \$id de ciudad y acción a realizar
3. Realizar acción
4. Mostrar información sobre operación realizada o error
5. Finalizar página

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 40

PHP y acceso a Bases de Datos
Botones de edición

Inclusión de botones en el listado
Comprobar si existe un \$id y qué acción hay que realizar

```
// ***** Argumentos POST de la página
if (isset($_POST['accion']) && isset($_POST['id'])) {
    switch ($_POST['accion']) {
        case 'Borrar': // Presentar formulario y pedir confirmación
            $accion = 'Borrar';
            $id = $_POST['id'];
            break;
        case 'Editar': // Presentar formulario y pedir confirmación
            $accion = 'Editar';
            $id = $_POST['id'];
            break;
        case 'Confirmar Borrado': // Borrado confirmado
            $accion = 'BorrarOK';
            $id = $_POST['id'];
            break;
        case 'Modificar Datos': // Modificación confirmada
            $accion = 'Modificar';
            $id = $_POST['id'];
            break;
        case 'Cancelar': break;
    }
}
```

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 41

PHP y acceso a Bases de Datos
Botones de edición. Borrado de registros

Inclusión de botones en el listado
editarCiudad.php: borrado de un registro

```
if (isset($id)) {
    if (!is_string($db=DB_conexion())) {
        switch ($accion) {
            case 'Borrar': $ciudad = DB_getCiudad($db,$id);
                $ciudad['editable']=false;
                FORM_editCiudad('Confirme borrado de esta ciudad..',
                                $ciudad,'Confirmar Borrado');
                break;
            case 'BorrarOK':
                if (DB_delCiudad($db,$id))
                    $info[] = 'La ciudad '.$_POST['ciu_nombre'].' ha sido borrada';
                else
                    $info[] = 'No se ha podido borrar '.$_POST['ciu_nombre'];
                break;
            ... (>> siguiente transparencia <<)
        }
        DB_desconexion($db);
    }
} else { // Si los parámetros no son correctos: volver al listado
    header('Location: listado_paginadoBotones.php');
}
```

Confirme borrado de esta ciudad:

Nombre:	<input type="text" value="Mataró"/>
Comunidad:	<input type="text" value="Katalonia"/>
Población:	<input type="text" value="104095"/>
<input type="button" value="Confirmar Borrado"/> <input type="button" value="Cancelar"/>	

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 42

Inclusión de botones en el listado

editarCiudad.php: edición de un registro

```

switch ($accion) {
    ...
    case 'Editar':
        $ciudad = DB_getCiudad($db,$id);
        FORM_editCiudad('Edite los datos:',$ciudad,'Modificar Datos');
        break;
    case 'Modificar':
        $msg = DB_actCiudad($db,[ 'id'=>$_POST['id'],
                                'nombre'=>$_POST['ciu_nombre'],
                                'comunidad'=>$_POST['ciu_comunidad'],
                                'poblacion'=>$_POST['ciu_poblacion']]);
        if ($msg==true)
            $info[] = 'La ciudad '.$_POST['ciu_nombre'].' ha sido actualizada';
        else {
            $info[] = 'No se ha podido actualizar '.$_POST['ciu_nombre'];
            $info[] = $msg;
        }
        //header('refresh: 5; url=listado_paginadoBotones.php');
        break;
    ...
}

```

Este cuadro de diálogo muestra los campos Nombre (Gijón), Comunidad (Asturias) y Población (267980). Los botones 'Modificar Datos' y 'Cancelar' están en la parte inferior.

Inclusión de botones en el listado

Mostrar información sobre la operación realizada o error (htmlforms.php)

```

if (isset($info) && msgCount($info)>0)
    msgError($info);

```

La ciudad Gijón ha sido actualizada

```

function msgCount($msg) {
    if (is_array($msg))
        if (count($msg)==0)
            return 0;
        else
            return msgCount($msg[0])+
                msgCount(array_slice($msg, 1));
    else if (!is_bool($msg))
        return 1;
    else
        return 0;
}

```

```

function msgError($msg,$tipo='msgerror')
{
    echo "<div class='$tipo'>";
    _msgErrorR($msg);
    echo '</div>';
}

function _msgErrorR($msg) {
    if (is_array($msg))
        foreach ($msg as $v)
            _msgErrorR($v);
    else
        echo "<p>$msg</p>";
}

```



Inclusión de botones en el listado

Formulario de edición de ciudad (en htmlforms.php)

```

function FORM_editCiudad($titulo,$datos,$accion) {
    if (isset($datos['editable'])) && $datos['editable']==false)
        $disabled='readonly="readonly"';
    else
        $disabled='';
    echo <<< HTML
<div class='frm_ciudad'> <form action='$_SERVER["PHP_SELF"]' method='POST'>
<h3>$titulo</h3>
<input type='hidden' name='id' value='{$datos["id"]}'/>
<div class='frm_ciudad_input'><label for='ciu_nombre'>Nombre:</label>
    <input type='text' name='ciu_nombre' value='{$datos["name"]}' $disabled/></div>
<div class='frm_ciudad_input'> <label for='ciu_comunidad'>Comunidad:</label>
    <input type='text' name='ciu_comunidad' value='{$datos["district"]}' $disabled/></div>
<div class='frm_ciudad_input'> <label for='ciu_poblacion'>Población:</label>
    <input type='text' name='ciu_poblacion' value='{$datos["population"]}' $disabled/></div>
<div class='frm_ciudad_submit'> <input type='submit' name='accion' value='$accion' />
    <input type='submit' name='accion' value='Cancelar' /></div>
</form> </div>
HTML;
}

```



Inclusión de botones en el listado

db.php: edición de un registro

```

// Borrar una ciudad
function DB_delCiudad($db,$id) {
    mysqli_query($db, "DELETE FROM city WHERE id='$id'");
    if (mysqli_affected_rows($db)==1)
        return true;
    else
        return false;
}

```

PHP y acceso a Bases de Datos
Botones de edición. Edición de registros

Inclusión de botones en el listado
db.php: edición de un registro

```
// Actualizar una ciudad
function DB_actCiudad($db,$datos) {
    // Comprobar si ya hay una ciudad con el mismo nombre
    $res = mysqli_query($db,
        "SELECT id,name FROM city WHERE name='{$datos['nombre']}'"
        AND countrycode='ESP'");
    $ciudad = mysqli_fetch_assoc($res);
    mysqli_free_result($res);
    if ($ciudad['name']==$datos['nombre'] && $ciudad['id']!=$datos['id'])
        $info[] = 'Ya hay otra ciudad con ese nombre';
    else {
        $res = mysqli_query($db, "UPDATE city SET name='{$datos['nombre']}',
            district='{$datos['comunidad']}',
            population='{$datos['poblacion']}'"
            WHERE id='{$datos['id']}');

        if (!$res) {
            $info[] = 'Error al actualizar';
            $info[] = mysqli_error($db);
        }
    }
    if (isset($info))
        return $info;
    else
        return true; // OK
}
```

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 47

Tecnologías Web
Grado en Ingeniería Informática

Programación en el lado del servidor

UNIVERSIDAD DE GRANADA

DECSAI

- 1. El lenguaje PHP
- 2. PHP y aplicaciones web
- 3. PHP y conexión con BBDD
 - 1. Introducción
 - 2. Conexión, consultas y organización
 - 3. Página de resultados
 - 4. Edición y borrado de registros
 - 5. Saneamiento de datos de formularios
 - 6. Inserción y búsqueda de registros
 - 7. Inyección de código SQL
 - 8. Consultas preparadas
- 4. Consideraciones finales

»

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena

PHP y acceso a Bases de Datos
Saneamiento de datos de formularios

Saneamiento de datos de formularios
Inclusión de tags HTML en las entradas

Ciudad	Comunidad	Población	Acción
A Coruña (La Coruña)	Galicia	243402	Editar Borrar
Albacete	Kastilia-La Mancha	147527	Editar Borrar
Alcorcón	Madrid	142048	Editar Borrar
Almería	Andalucía	169027	Editar Borrar

Edite los datos:

Nombre: <h1>A Coruña (La Coruña)</h1>

Comunidad: Galicia

Población: 243402

Modificar Datos Cancelar

La ciudad

A Coruña (La Coruña) ha sido actualizada

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 49

PHP y acceso a Bases de Datos
Saneamiento de datos de formularios

Saneamiento de datos de formularios
XSS (Cross Site Scripting): Ejecución de código malicioso en las entradas

Ciudad	Comunidad	Población	Acción
A Coruña (La Coruña)	Galicia	243402	Editar Borrar
Albacete	Kastilia-La Mancha	147527	Editar Borrar
Alcorcón	Madrid	142048	Editar Borrar
Almería	Andalucía	169027	Editar Borrar

Edite los datos:

Nombre: A Coruña (La Coruña)<script>alert("hola");</script>

Comunidad: Galicia

GeoWeb: geografía política
Una web con datos geográficos

No se ha podido actualizar A Coruña (La Coruña)
Error al actualizar
Data too long for column 'Name' at row 1

No se ha podido actualizar A Coruña (La Coruña)

hola

Aceptar

Firefox

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 50



Saneamiento de datos de formularios

Alternativas:

1. Limpiar las entradas antes de almacenar en BBDD. Menos recomendado.
No sabemos si la información incluida puede tener un uso distinto en el que tengan sentido ciertos caracteres especiales.
2. Limpiar las entradas antes de enviar como salida de PHP (recomendado)
 - htmlentities
 - htmlspecialchars

```
function VIEW_listadoCiudadesBotones($datos,$accion) {
echo "<div class='listado'> <table> <tr> <th>Ciudad</th> <th>Comunidad</th> <th>Población</th> <th>Acción</th> </tr>";
foreach ($datos as $v) {
    echo '<tr>'; echo '<td class="ciu_nombre">' . htmlentities($v['name']) . '</td>';
    echo '<td class="ciu_comunidad">' . htmlentities($v['district']) . '</td>';
    echo '<td class="ciu_poblacion">' . htmlentities($v['population']) . '</td>';
    echo "<td class='ciu_botones'><form action='".$accion' method='POST'>
        <input type='hidden' name='id' value='".$v['id']."' />
        <input type='submit' name='accion' value='Editar' />
        <input type='submit' name='accion' value='Borrar' />
    </form></td>";
    echo '</tr>';
}
echo "</table> </div>";
}
```



Saneamiento de datos de formularios

Además, conveniente codificación de query strings con urlencode

editarCiudad.php

```
$info[] = 'La ciudad '.htmlentities($_POST['ciu_nombre']).' ha sido borrada';
$info[] = 'No se ha podido borrar '.htmlentities($_POST['ciu_nombre']);
$info[] = 'La ciudad '.htmlentities($_POST['ciu_nombre']).' ha sido actualizada';
$info[] = 'No se ha podido actualizar '.htmlentities($_POST['ciu_nombre']);
```

listadoPaginado.php

```
htmlNav('paginador',[ 
    ['texto'=>'Primera', 'url'=>'?primero=0&items=' . urlencode($numitems)],
    ['texto'=>'Anterior', 'url'=>'?primero=' . urlencode($anterior).
        '&items=' . urlencode($numitems)],
    ['texto'=>'Siguiente', 'url'=>'?primero=' . urlencode($siguiente).
        '&items=' . urlencode($numitems)],
    ['texto'=>'Última', 'url'=>'?primero=' . urlencode($ultima).
        '&items=' . urlencode($numitems)]]);
```


UNIVERSIDAD
DE GRANADA

Tecnologías Web

Grado en Ingeniería Informática

Programación en el lado del servidor

- 1. El lenguaje PHP
- 2. PHP y aplicaciones web
- 3. PHP y conexión con BBDD
 - 1. Introducción
 - 2. Conexión, consultas y organización
 - 3. Paginación de resultados
 - 4. Edición y borrado de registros
 - 5. Saneamiento de datos de formularios
 - 6. Inserción y búsqueda de registros
 - 7. Inyección de código SQL
 - 8. Consultas preparadas
- 4. Consideraciones finales

»»»



PHP y acceso a Bases de Datos

Inserción de nuevos registros

Añadir nuevos registros (addCiudad.php)

Formulario de edición de ciudad

GeoWeb: geografía política
Una web con datos geográficos

Listado Listado Página Listado Página (botones) Búsqueda Nueva ciudad

Indique los datos:

Nombre:

Comunidad:

Población:

Universidad de Granada Logo Watermark



Añadir nuevos registros (addCiudad.php)

Formulario de edición de ciudad: comprobación de \$_POST

// ***** Argumentos POST de la página

```
$datos=false;
$accion='';
if (isset($_POST['accion']) && $_POST['accion']=='Añadir Ciudad') {
    $datos['id'] = '';
    $datos['name'] = isset($_POST['ciu_nombre']) ? $_POST['ciu_nombre'] : '';
    $datos['district'] = isset($_POST['ciu_comunidad']) ? $_POST['ciu_comunidad'] : '';
    $datos['population'] = isset($_POST['ciu_poblacion']) ? $_POST['ciu_poblacion'] : '';

    if ($datos['name']=='' || $datos['district']=='' || $datos['population']=='')
        $info[]='No puede dejar campos vacíos';
    if (!is_numeric($datos['population']) || $datos['population']<=0)
        $info[]='El valor de población debe ser numérico y superior a cero';

    if (!isset($info))
        $accion='Añadir';
}
```



Añadir nuevos registros (addCiudad.php)

Formulario de edición de ciudad: ejecutar acción o mostrar formulario

```
// ***** Contenido
if (!is_string($db=DB_conexion())) {
    if ($accion=='Añadir') {
        $res = DB_addCiudad($db,$datos);
        if ($res==true)
            $info[] = 'Se ha añadido la ciudad con éxito';
        else
            $info[] = $res;
    } else
        FORM_editCiudad('Indique los datos:',$datos,'Añadir Ciudad');

    if (isset($info) && msgCount($info)>0)
        msgError($info);

    // Desconectar de la BBDD (se puede omitir)
    DB_desconexion($db);
} else
    msgError($db);
```

PHP y acceso a Bases de Datos
Inserción de nuevos registros

Añadir nuevos registros (db.php)

Formulario de edición de ciudad: ejecutar acción o mostrar formulario

```

function DB_addCiudad($db,$datos) {
    // Comprobar si ya hay una ciudad con el mismo nombre
    $res = mysqli_query($db, "SELECT COUNT(*) FROM city
                                WHERE name='{$datos['name']}' AND countrycode='ESP'");
    $num = mysqli_fetch_row($res)[0];
    mysqli_free_result($res);

    if ($num>0)
        $info[] = 'Ya existe una ciudad con ese nombre';
    else {
        $res = mysqli_query($db, "INSERT INTO city (name,district,population,countrycode)
                                    VALUES ('{$datos['name']}','{$datos['district']}',
                                            '{$datos['population']}','ESP')");
        if (!$res) {
            $info[] = 'Error en la consulta '.__FUNCTION__;
            $info[] = mysqli_error($db);
        }
    }
    if (isset($info))
        return $info;
    else
        return true; // OK
}

```

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 57

PHP y acceso a Bases de Datos
Búsqueda de registros

Formulario de búsqueda (buscarCiudad.php)

- Coincidencia de parte del nombre / comunidad
- Con población dentro de un rango

Datos de la búsqueda:

Nombre:

Comunidad:

Población mínima:

Población máxima:

Buscar

Ciudad	Comunidad	Población	Acción
Jaén	Andalusia	109247	Editar Borrar
Lleida (Lérida)	Katalonia	112207	Editar Borrar
Marbella	Andalusia	101144	Editar Borrar
Ourense (Orense)	Galicia	109120	Editar Borrar
Tarragona	Katalonia	113016	Editar Borrar

Primera **Anterior** **Siguiente** **Última**

localhost/tw/bbdd/buscarCiudad.php?primero=0&items=10&bpobmin=100000&bpobmax=120000

PHP y acceso a Bases de Datos

Búsqueda de registros

Formulario de búsqueda (buscarCiudad.php)

- Parámetros \$_POST para solicitar una nueva búsqueda
- Parámetros \$_GET para controlar el paginado y la búsqueda

```
$accion='';
// **** Argumentos POST de la página
if (isset($_POST['accion'])) {
    if (isset($_POST['ciu_nombre']) && $_POST['ciu_nombre']!='')
        $cadenab['bnombre']=$_POST['ciu_nombre'];
    if (isset($_POST['ciu_comunidad']) && $_POST['ciu_comunidad']!='')
        $cadenab['bcomunidad']=$_POST['ciu_comunidad'];
    if (isset($_POST['ciu_poblacion_min']) && $_POST['ciu_poblacion_min']!='' &&
        is_numeric($_POST['ciu_poblacion_min']) && $_POST['ciu_poblacion_min']>=0)
        $cadenab['bpobmin']=$_POST['ciu_poblacion_min'];
    if (isset($_POST['ciu_poblacion_max']) && $_POST['ciu_poblacion_max']!='' &&
        is_numeric($_POST['ciu_poblacion_max']) && $_POST['ciu_poblacion_max']>=0)
        $cadenab['bpobmax']=$_POST['ciu_poblacion_max'];
    if (isset($cadenab) && count($cadenab)>0) {
        $accion='Buscar';
        $primero=0;
        $numitems=10;
    } else
        $info[] = 'No ha indicado ningún campo de búsqueda';
} else {
    // **** Argumentos GET de la página
}

```

PHP y acceso a Bases de Datos

Búsqueda de registros

Formulario de búsqueda (buscarCiudad.php)

- Parámetros \$_POST para solicitar una nueva búsqueda
- Parámetros \$_GET para controlar el paginado y la búsqueda

```
$accion='';
// **** Argumentos POST de la página
if (isset($_POST['accion'])) {
    ...
} else { // **** Argumentos GET de la página
    // primero: Primer item a visualizar
    // items : cuantos items incluir (<=0 para ver todos)
    if (!isset($_GET['items']))
        $numitems = 10; // Valor por defecto
    else if (!is_numeric($_GET['items']) || $_GET['items']<1)
        $numitems = 0; // Para mostrar todos los items
    else
        $numitems = $_GET['items'];
    if ($numitems==0)
        $primero=0; // Ver todos los items
    else {
        $primero = isset($_GET['primero']) ? $_GET['primero'] : 0;
        if (!is_numeric($primero) || $primero<0)
            $primero=0;
    }
}

```

PHP y acceso a Bases de Datos

Búsqueda de registros

Formulario de búsqueda (buscarCiudad.php)

- Parámetros `$_POST` para solicitar una nueva búsqueda
- Parámetros `$_GET` para controlar el paginado y la búsqueda

```
$accion='';
// **** Argumentos POST de la página
if (isset($_POST['accion'])) {
    ...
} else { // **** Argumentos GET de la página
    ...
$cadenab = [];
if (isset($_GET['bnombre']))
    $cadenab['bnombre']=$_GET['bnombre'];
if (isset($_GET['bcomunidad']))
    $cadenab['bcomunidad']=$_GET['bcomunidad'];
if (isset($_GET['bpobmin']) && is_numeric($_GET['bpobmin']) &&
    $_GET['bpobmin']>=0)
    $cadenab['bpobmin']=$_GET['bpobmin'];
if (isset($_GET['bpobmax']) && is_numeric($_GET['bpobmax']) &&
    $_GET['bpobmax']>=0 )
    $cadenab['bpobmax']=$_GET['bpobmax'];
if (count($cadenab)>0)
    $accion="Buscar";
}
```

PHP y acceso a Bases de Datos

Búsqueda de registros

Formulario de búsqueda (buscarCiudad.php)

- Mostrar formulario (sticky form)
- Ejecutar búsqueda y mostrar resultados

```
// **** Contenido
if (!is_string($db=DB_conexion())) {

    if (isset($cadenab))
        FORM_buscarCiudad('Datos de la búsqueda:',$cadenab);
    else
        FORM_buscarCiudad('Datos de la búsqueda:');

    if ($accion=='Buscar') {
        // Buscar y mostrar resultados
        // Barra de paginación
    }

    if (isset($info) && msgCount($info)>0)
        msgError($info);

    DB_desconexion($db);
} else
    msgError($db);
```

PHP y acceso a Bases de Datos

Búsqueda de registros

Formulario de búsqueda (buscarCiudad.php)

- Parámetros `$_POST` para solicitar una nueva búsqueda
- Parámetros `$_GET` para controlar el paginado y la búsqueda

```

if ($accion=='Buscar') {
    $busc = DB_array2SQL($cadenab);
    $numciudades=DB_getNumCiudades($db,$busc);
    if ($numciudades>0) {
        $ciudades=DB_getCiudades($db,$primero,$numitems,$busc);
        // Mostrar listado
        if ($ciudades!==false)
            VIEW_listadoCiudadesBotones($ciudades, 'editarCiudad.php');
        else {
            $info[] = 'Ha habido un error en la consulta a la BBDD';
            $info[] = mysqli_error($db);
        }
    } else
        $info[] ='No hay resultados de la búsqueda';
    ...
    // Barra de paginación
}

```

PHP y acceso a Bases de Datos

Búsqueda de registros

Formulario de búsqueda (buscarCiudad.php)

Barra de paginación (incluye `$_GET` con la búsqueda solicitada)

```

// Barra de paginación
if ($numciudades>0 && $numitems>0) {
    $ultima = $numciudades - ($numciudades%$numitems);
    $anterior = $numitems>$primero ? 0 : ($primero-$numitems);
    $siguiente = ($primero+$numitems)>$numciudades ? $ultima : ($primero+$numitems);
    htmlNav('paginator',[

        ['texto'=>'Primera',
         'url'=>'?primero=0&items=' . urlencode($numitems) . '&' . http_build_query($cadenab)],

        ['texto'=>'Anterior',
         'url'=>'?primero=' . urlencode($anterior) . '&items=' . urlencode($numitems) . '&' .
                     . http_build_query($cadenab)],

        ['texto'=>'Siguiente',
         'url'=>'?primero=' . urlencode($siguiente) . '&items=' . urlencode($numitems) . '&' .
                     . http_build_query($cadenab)],

        ['texto'=>'Última',
         'url'=>'?primero=' . urlencode($ultima) . '&items=' . urlencode($numitems) . '&' .
                     . http_build_query($cadenab)]]);
}

```



Formulario de búsqueda (buscarCiudad.php)

Mostrar formulario (en htmlforms.php)

```
function FORM_buscarCiudad($titulo,$datos=false) {
    $bnombre = isset($datos['bnombre']) ? " value='{$datos['bnombre']}' " : '';
    $bcomunidad = isset($datos['bcomunidad']) ? " value='{$datos['bcomunidad']}' " : '';
    $bpobmin = isset($datos['bpobmin']) ? " value='{$datos['bpobmin']}' " : '';
    $bpobmax = isset($datos['bpobmax']) ? " value='{$datos['bpobmax']}' " : '';

    echo <<< HTML
<div class='frm_ciudad'> <form action='' method='POST'> <h3>$titulo</h3>
<div class='frm_ciudad_input'> <label for='ciu_nombre'>Nombre:</label>
    <input type='text' name='ciu_nombre' $bnombre/> </div>
<div class='frm_ciudad_input'> <label for='ciu_comunidad'>Comunidad:</label>
    <input type='text' name='ciu_comunidad' $bcomunidad/> </div>
<div class='frm_ciudad_input'> <label for='ciu_poblacion_min'>Población mínima:</label>
    <input type='text' name='ciu_poblacion_min' $bpobmin/> </div>
<div class='frm_ciudad_input'> <label for='ciu_poblacion_max'>Población máxima:</label>
    <input type='text' name='ciu_poblacion_max' $bpobmax/> </div>
<div class='frm_ciudad_submit'>
    <input type='submit' name='accion' value='Buscar' /> </div> </form> </div>
HTML;
}
```



Formulario de búsqueda (buscarCiudad.php)

Obtención de datos de la BBDD (en db.php)

```
// Consulta para obtener listado de ciudades
function DB_getCiudades($db,$primero=0,$numitems=0,$cadenab='') {
    if ($numitems<=0) { // Listarlos todos
        $rango='';
    } else {
        $rango = 'LIMIT '.(int)($numitems).' OFFSET '.abs($primero);
    }
    // Consulta a la BBDD
    if (strlen($cadenab)!=0)
        $cadenab.=' AND ';
    $res = mysqli_query($db, "SELECT id,name,district,population FROM city
                                WHERE $cadenab countrycode='ESP' ORDER BY name $rango");
    ...

function DB_getNumCiudades($db,$cadenab='') {
    if ($cadenab!='')
        $cadenab .= ' AND ';
    $res = mysqli_query($db, "SELECT COUNT(*) FROM city
                                WHERE $cadenab countrycode='ESP'");
    ...
}
```

 **PHP y acceso a Bases de Datos**
Búsqueda de registros

Formulario de búsqueda (buscarCiudad.php)
Crear la cadena de búsqueda SQL (en db.php)

```

function DB_array2SQL($query) {
    $cadenab= '';
    if (array_key_exists('bnombre', $query))
        $cadenab .= " name LIKE '%{$query['bnombre']}%' AND";
    if (array_key_exists('bcomunidad', $query))
        $cadenab .= " district LIKE '%{$query['bcomunidad']}%' AND";
    if (array_key_exists('bpobmin', $query))
        $cadenab .= " population>='{$query['bpobmin']}' AND";
    if (array_key_exists('bpobmax', $query))
        $cadenab .= " population<='{$query['bpobmax']}' AND";
    if (strlen($cadenab)>0)
        $cadenab = substr_replace($cadenab, ' ', strlen($cadenab)-4, 4);
    return $cadenab;
}

```

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 67

 **Tecnologías Web**
Grado en Ingeniería Informática

Programación en el lado del servidor

UNIVERSIDAD DE GRANADA

1. El lenguaje PHP
2. PHP y aplicaciones web
3. PHP y conexión con BBDD

- 1. Introducción**
- 2. Conexión, consultas y organización**
- 3. Paginación de resultados**
- 4. Edición y borrado de registros**
- 5. Saneamiento de datos de formularios**
- 6. Inserción y búsqueda de registros**
- 7. Inyección de código SQL**
- 8. Consultas preparadas**

4. Consideraciones finales

»

 DECSAI

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena

PHP y acceso a Bases de Datos
Inyección de código SQL

SQL Injection

Inclusión de datos de formularios que invalidan la consulta SQL

Edite los datos:

Nombre:	A ' Coruña
Comunidad:	Galicia
Población:	243402

Modificar Datos

Warning: mysqli_fetch_assoc() expects parameter 1 to be mysqli_result, boolean given in /home/jbaena/ownCloud/Documents/Docencia/TecnologiasWeb/1617/transparencias/src/bbdd2/db.php on line 40

Warning: mysqli_free_result() expects parameter 1 to be mysqli_result, boolean given in /home/jbaena/ownCloud/Documents/Docencia/TecnologiasWeb/1617/transparencias/src/bbdd2/db.php on line 41

No se ha podido actualizar A '
Coruña

Error al actualizar

You have an error in your SQL
syntax; check the manual that
corresponds to your MySQL server
version for the right syntax to use
near 'Coruña', district='Galicia',
population='243402' WHERE
id='670" at line 1

PHP y acceso a Bases de Datos
Inyección de código SQL

SQL Injection

Inclusión de datos de formularios que invalidan la consulta SQL

Edite los datos:

Nombre:	A ' Coruña
Comunidad:	Galicia
Población:	243402

Modificar Datos **Cancelar**

```

function DB_actCiudad($db,$datos) {
    // Comprobar si ya hay una ciudad con el mismo nombre
    $res = mysqli_query($db, "SELECT id,name FROM city WHERE
        name='".$datos['nombre']."' AND countrycode='ESP'");
    $ciudad = mysqli_fetch_assoc($res);
    mysqli_free_result($res);

    SELECT id,name FROM city WHERE name='A ' Coruña' AND countrycode='ESP'
  
```



Puede ser no intencionado

```
$cad = "O'Reilly";
$sql = "SELECT * FROM libros WHERE editorial = '" . $cad . "'"; 
echo $sql, PHP_EOL; // SELECT * FROM libros WHERE editorial = 'O'Reilly'
```

Escapado de comillas

addslashes	Escapa ' , " , \ , el byte nulo
stripslashes	Quita el escapado de addslashes

```
$sql = "SELECT * FROM libros WHERE editorial = '" . addslashes($cad) . "'"; 
echo $sql, PHP_EOL; // SELECT * FROM libros WHERE editorial = 'O\Reilly'
```

Escapado de cadenas para uso en BBDD

Usar función específica del DMBS que se esté usando
`mysqli_real_escape_string` Escapa ' , " , \ , \n , \r , Ctrl-Z, byte nulo

```
$sql = "SELECT * FROM libros WHERE editorial = '" . 
        mysqli_real_escape_string($db,$cad) . "'"; 
echo $sql, PHP_EOL; // SELECT * FROM libros WHERE editorial = 'O\Reilly'
```



SQL Injection

Inyección de código SQL en la consulta a través del formulario

Indique los datos:

Nombre:	<input type="text" value="Alfacar');DELETE FROM city WHERE name='Albacete';#"/>
Comunidad:	<input type="text" value="Andalucía"/>
Población:	<input type="text" value="4424"/>

Ciudad	Comunidad	Población	Acción
A Coruña (La Coruña)	Galicia	243402	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Alcorcón	Madrid	142048	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Alfacar	Andalucía	4424	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Almería	Andalusia	169027	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Badajoz	Extremadura	136613	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>

Se ha borrado Albacete



Saneamiento de datos de formularios

SQL Injection: destrucción de datos

```
$datos['name'] = "Alfacar");DELETE FROM city WHERE name='Albacete';#"
```



```
"INSERT INTO city (district,countrycode,population,name) VALUES
('{$datos['district']}','ESP','{$datos['population']}','{$datos['name']}')"
```



```
INSERT INTO city (district,countrycode,population,name) VALUES
('Andalucía','ESP','4424','Alfacar');DELETE FROM city WHERE name='Albacete';#')
```

- PHP mysqli_query solo permite una instrucción SQL
- Para hacer la prueba de inyección se ha cambiado por mysqli_multi_query
- En otros DBMS puede variar el comportamiento de la API



Saneamiento de datos de formularios

SQL Injection: obtención de datos sin permiso

#	id	nombre	apellidos	clave
1	2	Javier	Martinez	col
2	5	Antonio	Garrido	tomate
3	7	Joaquin	Fernandez	lechuga

```
echo "Consulta con inyección: ", PHP_EOL;
$campo = "select nombre from usuarios where id='".$campo."'";
$result = mysqli_query($db,$sql);
while ($f = mysqli_fetch_assoc($result))
    echo $f['nombre'], PHP_EOL;
```

```
$campo = '' union select concat(nombre,' ',clave) from usuarios where nombre<>'';
```

```
select nombre from usuarios where id='' union
select concat(nombre,' ',clave) from usuarios
where nombre<>''
```

Consulta con inyección:
Javier col
Antonio tomate
Joaquín lechuga



PHP y acceso a Bases de Datos

Inyección de código SQL

```

function DB_delCiudad($db,$id) {
    mysqli_query($db, "DELETE FROM city WHERE id='".mysqli_real_escape_string($db,$id)."'");
    ...
}

function DB_actCiudad($db,$datos) {
    $res = mysqli_query($db, "SELECT id,name FROM city WHERE name='".
        mysqli_real_escape_string($db,$datos['nombre'])."' AND countrycode='ESP'");
    ...
    $res = mysqli_query($db,
        "UPDATE city SET name='". mysqli_real_escape_string($db,$datos['nombre'])."' ,
        district='". mysqli_real_escape_string($db,$datos['comunidad'])."' ,
        population='". mysqli_real_escape_string($db,$datos['poblacion'])."' ,
        WHERE id='". mysqli_real_escape_string($db,$datos['id'])."'");
    ...
}

function DB_array2SQL($db,$query) {
    $cadenab='';
    if (array_key_exists('bnombre', $query))
        $cadenab .= " name LIKE '%".mysqli_real_escape_string($db,$query['bnombre'])."%'" AND";
    ...

function DB_addCiudad($db,$datos) {
    $res = mysqli_query($db, "SELECT COUNT(*) FROM city WHERE name='".
        mysqli_real_escape_string($db,$datos['name'])."' AND countrycode='ESP'");
    $res = mysqli_query($db, "INSERT INTO city (name,district,population,countrycode) VALUES ('".
        mysqli_real_escape_string($db,$datos['name'])."',",
        mysqli_real_escape_string($db,$datos['district'])."',",
        mysqli_real_escape_string($db,$datos['population'])."', 'ESP')");
    ...
}

```


UNIVERSIDAD
DE GRANADA

Tecnologías Web

Grado en Ingeniería Informática

Programación en el lado del servidor

1. El lenguaje PHP

2. PHP y aplicaciones web

3. PHP y conexión con BBDD

- 1. Introducción**
- 2. Conexión, consultas y organización**
- 3. Página de resultados**
- 4. Edición y borrado de registros**
- 5. Saneamiento de datos de formularios**
- 6. Inserción y búsqueda de registros**
- 7. Inyección de código SQL**
- 8. Consultas preparadas**

4. Consideraciones finales



Consultas preparadas

Qué son

Consultas preparadas

Mecanismo para mejorar:

- Eficiencia de las consultas
- Seguridad de las consultas

Consulta normal:

```
$consulta = "SELECT name FROM city WHERE population>'50000'";
mysqli_query($db, $consulta);
```

Consulta preparada:

```
$consulta = "SELECT name FROM city WHERE population>?"; } Preparación
$c = mysqli_prepare($db,$consulta);
```

1.- Se crea un patrón y se usan marcadores de posición (placeholders) para indicar donde se colocará un valor concreto

```
$val = 50000;
mysqli_stmt_bind_param($c, 's', $val);
mysqli_stmt_execute($c); } Ejecución:
bind + execute
```

2.- Se sustituyen los marcadores de posición por valores concretos y se ejecuta la consulta

Consultas preparadas

Seguridad

Consultas preparadas: seguridad

Añade comillas y escapa las variables ligadas automáticamente

Consulta normal

```
function DB_delCiudad($db,$id) {
    mysqli_query($db, "DELETE FROM city WHERE id='".
        mysqli_real_escape_string($db,$id)."');
    if (mysqli_affected_rows($db)==1)
        return true;
    else
        return false;
}
```

Consulta preparada

```
function DB_delCiudad($db,$id) {
$prep = mysqli_prepare($db,"DELETE FROM city WHERE id=?");
$val = $id;
mysqli_stmt_bind_param($prep, 's', $val);
mysqli_stmt_execute($prep);
if (mysqli_stmt_affected_rows($prep)==1)
    $ret = true;
else
    $ret = false;
mysqli_stmt_close($prep);
return $ret;
}
```

Consultas preparadas

Parámetros ligados

Consultas preparadas

Ligando parámetros para la consulta

```
mysqli_stmt_bind_param($consulta, TIPOS, variables, ...)

$consulta: consulta preparada
TIPOS: cadena con una letra por cada variable ligada indicando su tipo
    s: cadena
    i: entero
    d: real
    b: binario (blob), por ejemplo para imágenes, pdf, etc
Variables: tantas variables como marcadores de posición
```

Los marcadores de posición:

- Solo se pueden usar para valores de columnas
- No se pueden usar en otras partes de la consulta

Consultas preparadas

Acceso a datos de la consulta

Consultas preparadas: seguridad

Obteniendo datos con SELECT

Consulta normal

```
function DB_actCiudad($db,$datos) {
    // Comprobar si ya hay una ciudad con el mismo nombre
    $res = mysqli_query($db, "SELECT id,name FROM city WHERE name='".
        mysqli_real_escape_string($db,$datos['nombre'])."' AND countrycode='ESP'");
    $ciudad = mysqli_fetch_assoc($res);
    mysqli_free_result($res);
    if ($ciudad['name']==$datos['nombre'] && $ciudad['id']!=$datos['id'])
        $info[] = 'Ya hay otra ciudad con ese nombre';
    else {
        ...
    }
}
```

Consulta preparada

```
function DB_actCiudad($db,$datos) {
    $prep = mysqli_prepare($db, "SELECT id,name FROM city WHERE name=? AND countrycode='ESP'");
    $val = $datos['nombre'];
    mysqli_stmt_bind_param($prep, 's', $val);
    mysqli_stmt_execute($prep);
    $res = mysqli_stmt_get_result($prep);
    $ciudad = mysqli_fetch_assoc($res);
    mysqli_free_result($res);
    mysqli_stmt_close($prep);
    if ($ciudad['name']==$datos['nombre'] && $ciudad['id']!=$datos['id'])
        $info[] = 'Ya hay otra ciudad con ese nombre';
    else
        ...
}
```

El resultado obtenido con mysqli_stmt_get_result se procesa igual que en las consultas normales

Consultas preparadas

Acceso a datos de la consulta

Consultas preparadas: seguridad

Actualizando datos con UPDATE

```
function DB_actCiudad($db,$datos) {
    ...
    else {
        $res = mysqli_query($db, "UPDATE city SET name='".
            mysqli_real_escape_string($db,$datos['nombre'])."', district='".
            mysqli_real_escape_string($db,$datos['comunidad'])."', population='".
            mysqli_real_escape_string($db,$datos['poblacion'])."' WHERE id='".
            mysqli_real_escape_string($db,$datos['id'])."'");
        if (!$res) {
            ...
}

function DB_actCiudad($db,$datos) {
    ...
    else {
        $prep = mysqli_prepare($db, "UPDATE city SET name=?, district=?, population=? WHERE id=?");
        $val1 = $datos['nombre'];
        $val2 = $datos['comunidad'];
        $val3 = $datos['poblacion'];
        $val4 = $datos['id'];
        mysqli_stmt_bind_param($prep, 'sssi', $val1, $val2, $val3, $val4);
        $result = mysqli_stmt_execute($prep);
        if (!$result) { $info[] = 'Error al actualizar'; $info[] = mysqli_stmt_error($db); }
        mysqli_stmt_close($prep);
    ...
}
```

Consultas preparadas

Acceso a datos de la consulta: alternativas

Consultas preparadas

- El resultado de las preguntas preparadas, por defecto:
 - Se mantiene en el DBMS (requiere menos memoria en el cliente)
 - No se mantiene en buffer: al obtener una tupla, esta se pierde → No se puede iterar más de una vez por el resultado.

Alternativas para procesar los resultados obtenidos

1. Transferir al cliente con `mysqli_stmt_get_result()` y tratar el resultado como una consulta normal (ejemplo previo).
2. Obtener desde el servidor uno a uno con `mysqli_stmt_fetch()`. No es posible iterar más de una vez por el resultado puesto que con cada `fetch` se pierde la tupla del DBMS.
3. Almacenar en un buffer del cliente el resultado con `mysqli_stmt_store_result()` de forma que podemos iterar múltiples veces con `mysqli_stmt_fetch()`.

Consultas preparadas

Acceso a datos de la consulta desde DBMS

Consultas preparadas

Obtener datos desde DBMS sin buffer

```

function DB_getCiudad($db, $id) {
    $res = mysqli_query($db, "SELECT id,name,district,population FROM city WHERE id='".
                           mysqli_real_escape_string($db,$id)."'"');
    if ($res && mysqli_num_rows($res)==1)    $ciudad = mysqli_fetch_assoc($res);
    else   $ciudad = false;
    mysqli_free_result($res);
    return $ciudad;
}

function DB_getCiudad($db, $id) {
    $prep = mysqli_prepare($db,"SELECT id,name,district,population FROM city WHERE id=?");
    $val = $id;
    mysqli_stmt_bind_param($prep,'s',$val);
    if (mysqli_stmt_execute($prep)) {
        mysqli_stmt_bind_result($prep,$rid,$rname,$rdistrict,$rpopulation);
        if (mysqli_stmt_fetch($prep)) {
            $ciudad['id'] = $rid;
            $ciudad['name'] = $rname;
            $ciudad['district'] = $rdistrict;
            $ciudad['population'] = $rpopulation;
        } else
            $ciudad = false; // No hay resultados
    } else
        $ciudad = false; // Error en consulta
    mysqli_stmt_close($prep);
    return $ciudad;
}

```

mysql_stmt_bind_result() liga las variables de la consulta a variables de PHP, cada fetch() actualiza las variables PHP

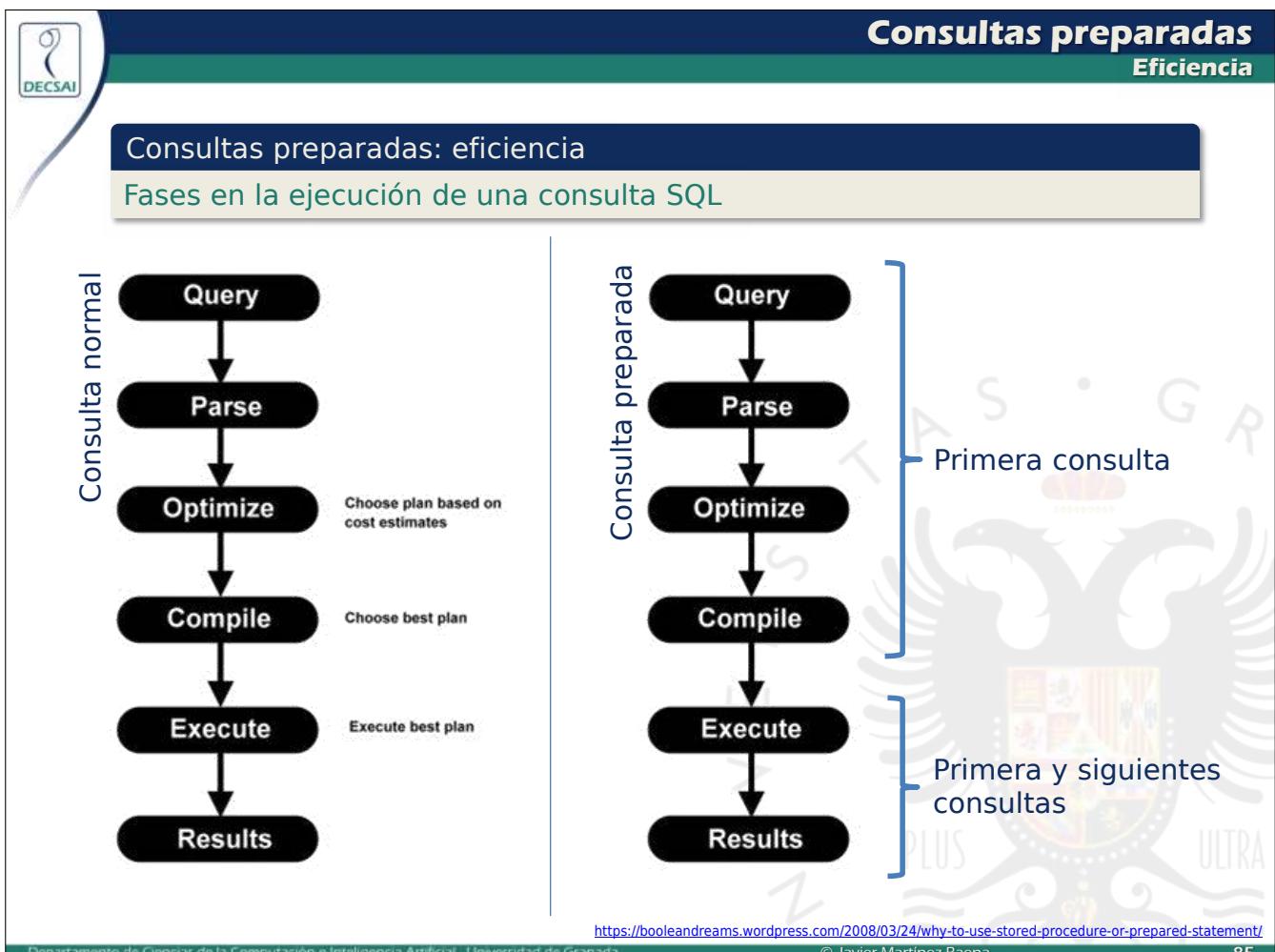
Consultas preparadas

Resumen de API

Consultas preparadas: algunas funciones de la API

<code>mysqli_prepare</code>	Crear una consulta preparada
<code>mysqli_stmt_bind_param</code>	Ligar datos a la consulta
<code>mysqli_stmt_execute</code>	Ejecutar consulta
<code>mysqli_stmt_close</code>	Cerrar consulta
<code>mysqli_stmt_get_result</code>	Obtener resultados completos
<code>mysqli_stmt_bind_result</code>	Ligar variables de salida
<code>mysqli_stmt_fetch</code>	Obtener siguiente tupla
<code>mysqli_stmt_store_result</code>	Almacenar en buffer de cliente todas las tuplas
<code>mysqli_stmt_errno</code>	Código de error de última operación
<code>mysqli_stmt_error</code>	Mensaje de error de última operación
<code>...</code>	

<http://php.net/manual/es/mysqli.summary.php>



Consultas preparadas: eficiencia

- Requieren 2 conexiones al DBMS en la primera consulta
- Las siguientes consultas requieren solo una conexión
- En las siguientes consultas solo se transmiten los datos de los marcadores de posición (y no la consulta completa)

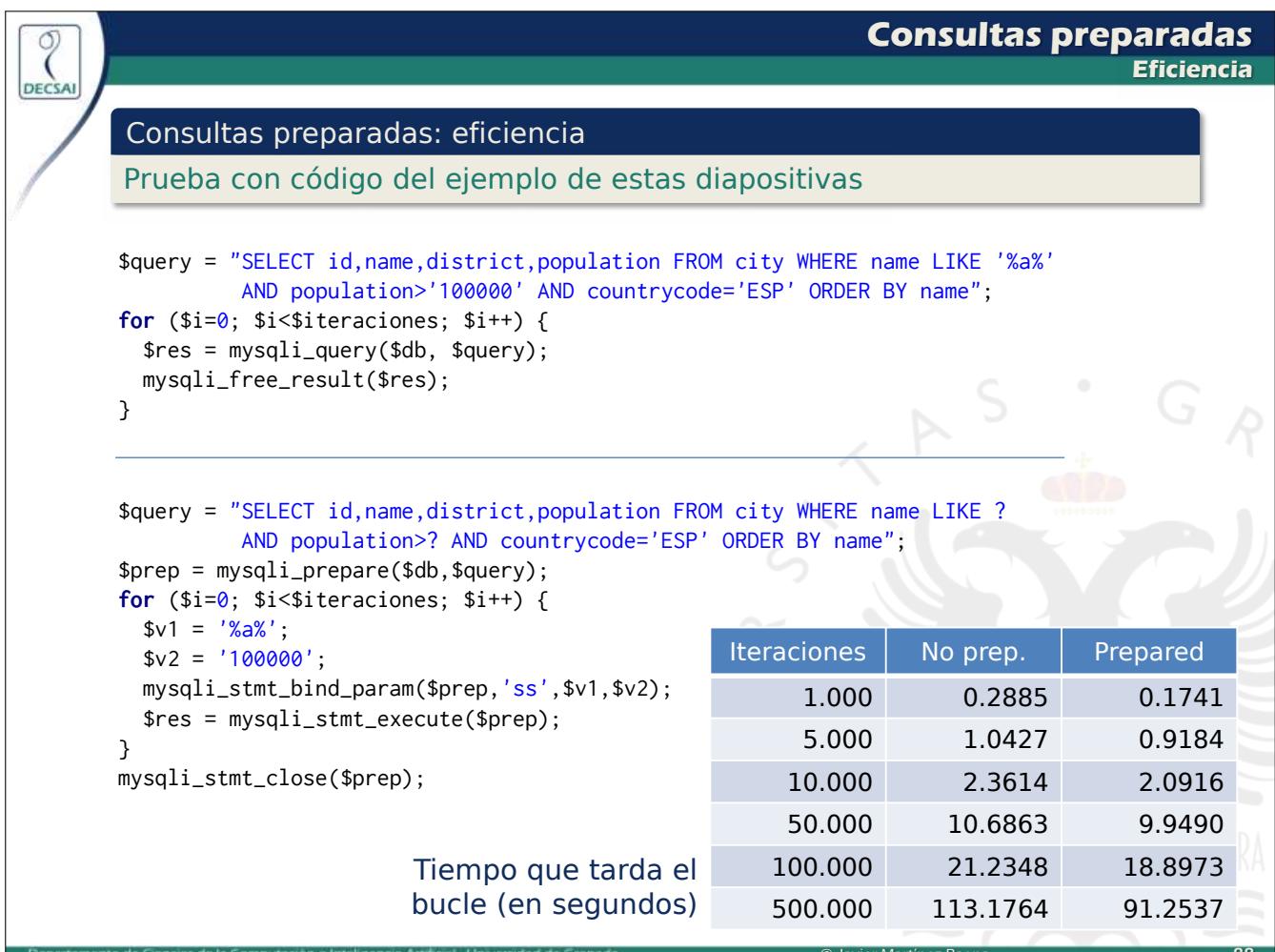
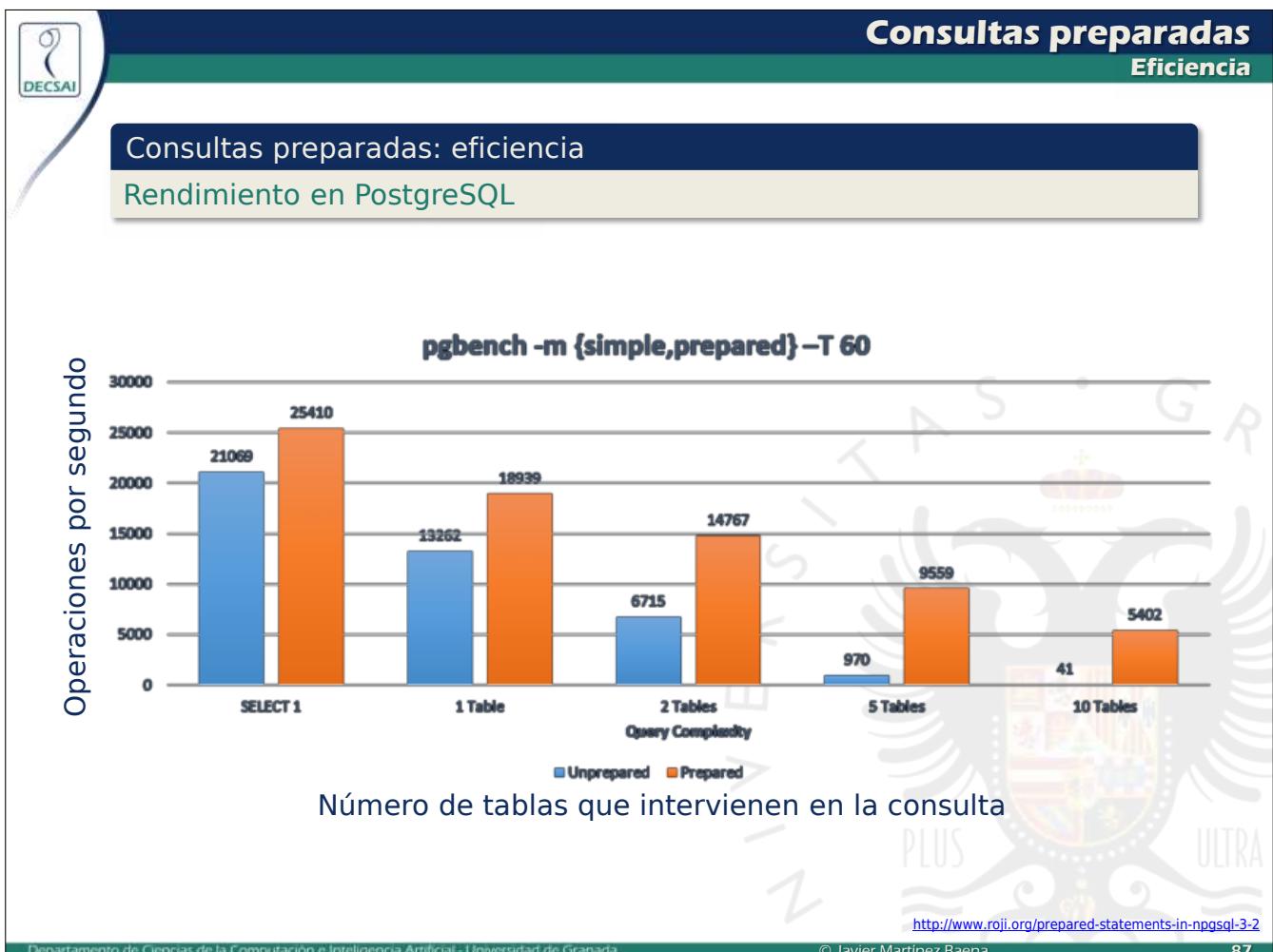
Sobre la eficiencia de la consulta:

- Puede depender el DBMS y de su configuración. Ejemplo: MySQL admite cacheo de tablas para este tipo de consultas solo en nuevas versiones

```
SELECT t.id,t.name,t.district,t.population FROM (SELECT * FROM city) t
      WHERE name LIKE '%a%' AND population>'100000' AND countrycode='ESP'
      ORDER BY name
```

En una consulta como esta cachear el resultado de la subconsulta puede acelerar mucho los cálculos

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 86





UNIVERSIDAD
DE GRANADA

Tecnologías Web

Grado en Ingeniería Informática

Programación en el lado del servidor

1. El lenguaje PHP
2. PHP y aplicaciones web
3. PHP y conexión con BBDD
 1. Introducción
 2. Conexión, consultas y organización
 3. Paginación de resultados
 4. Edición y borrado de registros
 5. Saneamiento de datos de formularios
 6. Inserción y búsqueda de registros
 7. Inyección de código SQL
 8. Consultas preparadas
- » 4. Consideraciones finales



PHP y aplicaciones web

Backups de seguridad

Backups de seguridad

Si tenemos acceso a Shell:

- MySQL CLI: mysqldump

Si tenemos acceso a cron:

- Cron: backups periódicos automáticos

```
mysqldump -u user -p password database > database.sql
```

DROP + CREATE + INSERT

```
mysql -u user -ppassword -D database < database.sql
```



PHP y aplicaciones web
Backups de seguridad

```
// Obtener listado de tablas
$tablas = array();
$result = mysqli_query($db, 'SHOW TABLES');
while ($row = mysqli_fetch_row($result))
    $tablas[] = $row[0];

// Salvar cada tabla
$salida = '';
foreach ($tablas as $tab) {
    $result = mysqli_query($db, 'SELECT * FROM '.$tab);
    $num = mysqli_num_fields($result);
    $salida .= 'DROP TABLE '.$tab.';';
    $row2 = mysqli_fetch_row(mysqli_query($db, 'SHOW CREATE TABLE '.$tab));
    $salida .= "\n\n".$row2[1].";\n\n"; // row2[0]=nombre de tabla
    while ($row = mysqli_fetch_row($result)) {
        $salida .= 'INSERT INTO '.$tab.' VALUES(';
        for ($j=0; $j<$num; $j++) {
            $row[$j] = addslashes($row[$j]);
            $row[$j] = preg_replace("/\n/", "\n", $row[$j]);
            if (isset($row[$j]))
                $salida .= "'".$row[$j]."'";
            else
                $salida .= "''";
            if ($j < ($num-1)) $salida .= ',';
        }
        $salida .= ");\n";
    }
    $salida .= "\n\n\n";
}

```

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 91

PHP y aplicaciones web
Backups de seguridad

Backups

Backups de la BBDD (y restauración) desde la aplicación web

```
mysqli_query($db, 'SET FOREIGN_KEY_CHECKS=0');

$result = mysqli_query($db, 'SHOW TABLES');
while ($row = mysqli_fetch_row($result))
    mysqli_query($db, 'DELETE * FROM '.$row[0]);

$error = '';
$sql = file_get_contents($f);
$queries = explode(';', $sql);
foreach ($queries as $q) {
    if (!mysqli_query($db, $q))
        $error .= mysqli_error($db);
}

mysqli_query($db, 'SET FOREIGN_KEY_CHECKS=1');
```

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 92

PHP y aplicaciones web

Backups de seguridad

Backups (backup.php)

Backups de la BBDD (y restauración) desde la aplicación web

```
<?php
require('html.php');           // Maquetado de página
require('db.php');             // Operaciones con BBDD
require('db_backup.php');      // Operaciones de backup de la BBDD

if (isset($_GET['download'])) {
    if (!is_string($db=DB_conexion())) {
        header('Content-Type: application/octet-stream');
        header('Content-Disposition: attachment; filename="db_backup.sql"');
        echo DB_backup($db);
        DB_desconexion($db);
    }
} else {
    htmlStart('Copia de seguridad', 'Backup');
    msgError("<a href='".$SERVER['SCRIPT_NAME']."?download'>Pulse aquí</a> para
descargar un fichero con los datos de la copia de seguridad",'msginfo');
    htmlEnd();
}
?>
```

PHP y aplicaciones web

Backups de seguridad

Backups (restore.php)

Backups de la BBDD (y restauración) desde la aplicación web

```
<?php
require('html.php');           // Maquetado de página
require('db.php');             // Operaciones con BBDD
require('db_backup.php');      // Operaciones de backup de la BBDD
htmlStart('Restaurar Base de Datos', 'Restore');
if (isset($_POST['submit'])) {
    /* Comprobar que se ha subido algún fichero */
    if ((sizeof($_FILES)==0) || !array_key_exists("fichero",$_FILES))
        $error = "No se ha podido subir el fichero";
    else if (!is_uploaded_file($_FILES['fichero']['tmp_name']))
        $error = "Fichero no subido. Código de error: ".$_FILES['fichero']['error'];
    else {
        $db=DB_conexion();
        $error = DB_restore($db,$_FILES['fichero']['tmp_name']);
        DB_desconexion($db);
    }
    if (isset($error) && msgCount($error)>0)
        msgError($error);
    else
        msgError("Base de datos restaurada correctamente",'msginfo');
} else
FORM_restore();
htmlEnd();
?>
```



Instalación de la aplicación

Instalación de la aplicación

- Desde la aplicación web
- Desde el sistema (command line u otras aplicaciones)

Pasos:

- Copia de ficheros en servidor
- Creación de fichero de credenciales
- Creación de la BBDD



Actualización de la aplicación

¿Cómo instalar una nueva versión de una aplicación web en producción?

Importante: probar antes de cambiar

Detener el servidor y ponerlo en modo mantenimiento

- Se modifica index.html para mostrar información de actualización
- Puede mantener el servicio detenido demasiado tiempo

Subir nueva versión encima de la antigua:

- No se ha probado antes de cambiar la versión
- El servidor en producción puede ser diferente al servidor de desarrollo
- Se pueden dejar ficheros de la antigua perdidos
- Durante la subida de ficheros podría haber usuarios navegando y obtener resultados incorrectos/extráños

Subir nueva versión en otra carpeta:

- Permite probar la aplicación antes de ponerla en producción
- Una vez probada: mover directorios
- Cabe la posibilidad de que un usuario pida una página mientras se está moviendo y obtenga un error 404
- Si un usuario está navegando y se mueven los directorios: pasa de estar en versión antigua a nueva y puede haber disfunciones



Actualización de la aplicación

¿Cómo instalar una nueva versión de una aplicación web en producción?

Instalar nuevas versiones en una carpeta de pruebas

- Permite probar la aplicación
- Se mantiene la versión en producción

Una vez probada:

- Se renombra la carpeta de pruebas a una carpeta definitiva (distinta de otras versiones en producción)

En el directorio raíz se pone un index.php que redirige automáticamente a la última versión en producción para nuevas sesiones

- Si un usuario está usando una versión antigua, seguirá en ella aunque pongamos una nueva
- Puede haber varias versiones en producción a la vez

Cuidado si cambia la estructura de la BBDD

Marc Rochkind, "Expert PHP and MySQL". APress. 2013

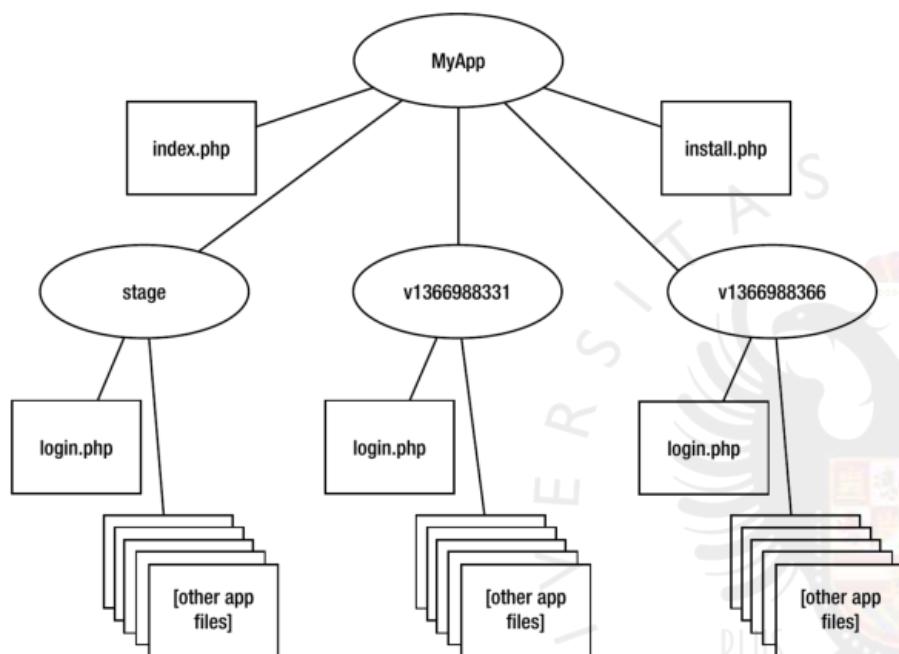
© Javier Martínez Baena

97



Actualización de la aplicación

¿Cómo instalar una nueva versión de una aplicación web en producción?



Marc Rochkind, "Expert PHP and MySQL". APress. 2013

© Javier Martínez Baena

98

 **PHP y aplicaciones web**
Actualización de la aplicación

Actualización de la aplicación

¿Cómo instalar una nueva versión de una aplicación web en producción?

```

index.php
if (empty($_GET['stage']))
    $directorio = ultima_version();
else
    $directorio = 'stage';
header("Location: $directorio/login.php");

// Obtener la última versión instalada
function ultima_version() {
    $directorio = '';
    $dirs = scandir('.', SCANDIR_SORT_DESCENDING);
    $c = 0;
    $encontrado = false;
    while (!encontrado && $c < count($dirs)) {
        if (preg_match('/^v[0-9]{10}$/', $dirs[$c])) {
            $directorio = $dirs[$c];
            $encontrado = true;
        }
        $c++;
    }
    return $directorio;
}

```

Marc Rochkind. "Expert PHP and MySQL". APress. 2013
 © Javier Martínez Baena 99

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada

 **PHP y aplicaciones web**
Actualización de la aplicación

Actualización de la aplicación

¿Cómo instalar una nueva versión de una aplicación web en producción?

```

install.php
echo '<p>Instalando versión en producción</p>';
$dir = 'v' . str_pad(ultima_version() + 1, 10, '0', STR_PAD_LEFT);
if (!rename('stage', $dir))
    die('<p>Fallo renombrando directorio</p>');
echo "<p>instalada la versión ($dir)</p>";

// Localizar última versión instalada
function ultima_version() {
    $directorio = 0;
    $dirs = scandir('.', SCANDIR_SORT_DESCENDING);
    $c = 0;
    $encontrado = false;
    while (!encontrado && $c < count($dirs)) {
        if (preg_match('/^v([0-9]{10})$/', $dirs[$c], $loc)) {
            $directorio = $loc[1];
            $encontrado = true;
        }
        $c++;
    }
    return $directorio;
}

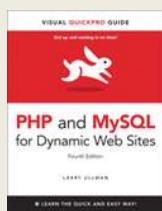
```

Marc Rochkind. "Expert PHP and MySQL". APress. 2013
 © Javier Martínez Baena 100

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada

Programación en el lado del servidor: PHP y MySQL

Bibliografía

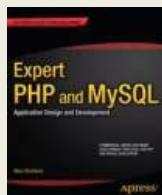


Larry Ullman

PHP and MySQL for Dynamic Web Sites (4ed)

Peachpit Press. 2012

<http://www.larryullman.com/books/php-and-mysql-for-dynamic-web-sites-visual-quickpro-guide-4th-edition/>



Marc Rochkind

Expert PHP and MySQL.

Application Design and Development

APress. 2014



David Powers

PHP Solutions (3ed)

Dynamic Web Design Made Easy

APress. 2015