

Páginas web dinámicas con base de datos MySQL®

Diseño web con Dreamweaver® y PHP® - Parte 1

By Edu – Ofimega acadèmies

Parte 1. Creación de la base de datos y páginas de administración

1. [Sesión 0. Introducción y requisitos](#)
2. [Sesión 1. Creación de la base de datos y las tablas MySQL](#)
3. [Sesión 2. Sitio y plantilla de administración](#)
4. [Sesión 3. Páginas de administración de productos](#)
5. [Sesión 4. Páginas de administración de categorías](#)
6. [Sesión 5. Mejoras en páginas de administración](#)
7. [Sesión 6. Páginas de administración de usuarios](#)

En la segunda parte veremos: Páginas frontales, filtros y subir la web

8. Sesión 7: Página frontal. Cara al público
9. Sesión 8: Filtros o consultas. Clasificación de productos
10. Sesión 9: Envío de formularios y pedidos
11. Sesión 10: Gestión de pedidos y carrito
12. Sesión 11: Copia y exportación de la base de datos
13. Sesión 12: Subir la web al servidor remoto
14. Ampliación: Acceso a los datos desde aplicaciones Windows

Sesión 0: Introducción y requisitos

Mostrar datos en la web con Dreamweaver® y PHP®. Introducción

Objetivo:

El objetivo de este curso es que, con los mínimos conocimientos, puedas diseñar una sencilla página web dinámica, es decir, interactiva. En la que puedas presentar y gestionar los productos de tu empresa de manera rápida, por numerosos que sean, porque estos estarán almacenados en unas tablas de datos. También podremos enviar pedidos y registrar clientes.

En los ejercicios de simulación hemos supuesto una empresa *tipo* que necesite de la exposición de sus productos. Puedes tratarse de una tienda de calzado denominada: “**Mike**” o de una inmobiliaria llamada: “**Oficasa**”. A la hora de presentar los productos el proceso es equivalente. Puedes ver un ejemplo real similar realizado por un alumno en la siguiente dirección: www.tayre.cat

Objetivo didáctico:

El objetivo didáctico de este curso es aprender a gestionar bases de datos con MySQL desde páginas web dinámicas. Incidiremos especialmente en la práctica con el lenguaje de consulta de datos **SQL** reduciendo a lo más básico el uso del lenguaje PHP y al mínimo los Scripts en las páginas.

Requisitos:

Conocimientos previos:



Para realizar este curso sin problemas, es necesario tener unos conocimientos básicos de diseño de páginas web en HTML, o haber realizado previamente el primer curso de diseño web con *Dreamweaver*® Adobe.

Aplicaciones o programas que utilizaremos:

Los recursos de software que necesitamos son:

1. **El lenguaje PHP:** es un lenguaje para el desarrollo de páginas web y, sobre todo, manipulación de datos ya que permite ejecutar parte del programa en el servidor web. Se integra perfectamente con *Dreamweaver*
2. **MySQL:** Es un gestor de base de datos **BDR** (Base de Datos Relacional) de software libre (licencia gratuita) que permite *manipular* los datos desde un servidor web con el famoso lenguaje estándar de bases de datos: **SQL** (Estándar Query Language). El cual aprenderemos un poco.
3. **Apache** es el programa que procesa la aplicación PHP del lado del servidor.
Xampp o *Wampp* Simulan el funcionamiento desde el servidor pero permiten hacer las pruebas dentro de tu equipo, sin subirlo a la web. Además estos incluyen a los tres anteriores.

Aplicaciones mínimas que debemos tener instaladas en el PC para realizar los ejercicios del curso:

-  **Dreamweaver** versiones CS 4, 5 ó 6 © Adobe. No versión CC. Puedes obtener una copia de 30 días de prueba en la Web de Adobe.
-  **Xampp** (o **Wampp**) ya que estos paquetes instalan Apache, PHP5, MySQL y PHPmyadmin de forma automática. (**Lampp** desde Linux)

Opcional: Un manejador de bases de datos MySQL como *Navicat* o *MySQLWorkBench* (aunque utilizaremos en el curso *PhpMyAdmin* ya que se proporciona con el *Wampp* o *Xampp*) y un diseñador de imágenes como *Photoshop* © Adobe.

Nota: Los ejercicios del curso se han realizado bajo el sistema operativo Windows 7, Xampp y Dreamweaver 5.5.

Instalaciones previas:

- Si necesitas instalar Adobe **Dreamweaver** en el equipo:
Accede a la web: www.adobe.com/es/products/dreamweaver.html y pulsa en: **Descargar** – Versión de prueba. CS 6. Una vez descargado en el equipo, ejecuta el paquete de instalación. *Nota:* Hasta la fecha, la versión de DW CC (Creative Cloud) no dispone de los asistentes para la gestión de base de datos.
- Si necesitas instalar **Xampp** en el equipo:
Accede a la web: www.apachefriends.org/es/xampp.html Descarga y ejecuta el archivo de instalación para Windows. Se recomienda seleccionar como carpeta de instalación: **c:\xampp** o similar, para evitar entrar en la carpeta Archivos de programas y tener que crear permisos.

Requisito del servidor.

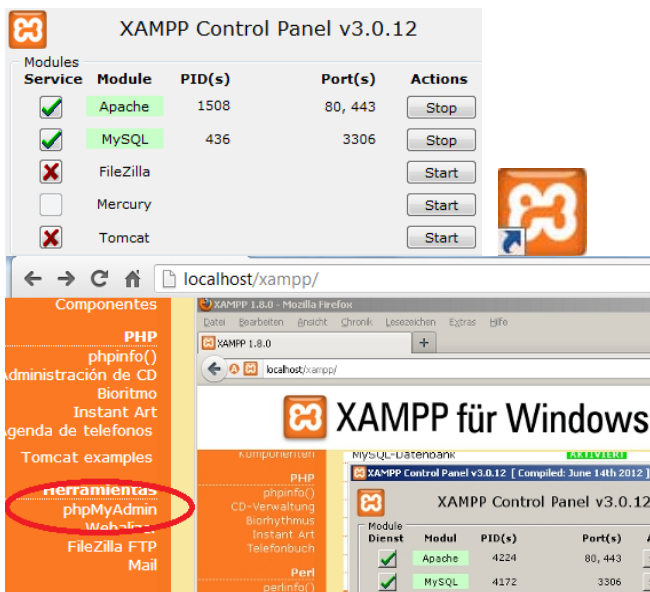
Para subir la web a un servidor, es necesario que la empresa servidora de la web, te ofrezca un espacio de almacenamiento con, al menos, una base de datos MySQL y poder ejecutar PHP.

Ejemplos de servidores de almacenamiento web con PHP, MySQL:

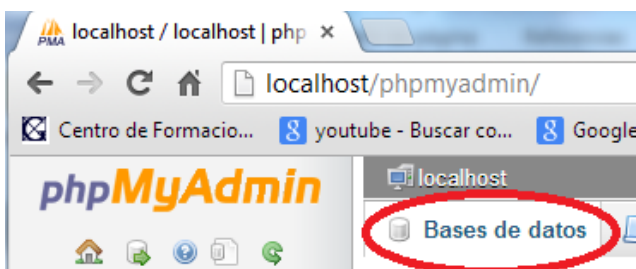
- Servidores gratuitos: Hostinger.es, liquid2k.com, brinkster.com, demented.org, cfm-resources.com, digitalrice.com
- Servidores de pago, incluyen dominio alojado con PHP, MySQL a bajo precio: Stratoweb, Nominalia, Acens, Arsys, EvidaliaHosting, Hostsuar.com, HostPapa, GoDaddy, One.com, etc...

Sesión 1: Creación de la base de datos y las tablas con MySQL

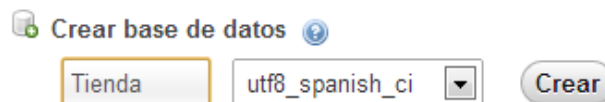
1º Crear de la base de datos.



1. Una vez instalado *Xampp* o *Wampp*, ejecuta el acceso directo al panel de control para comprobar que tienes activado los servicios *Apache* y *MySQL*. En caso contrario actívalos pulsando el botón **Start**.
2. Ejecuta tu navegador web (*Chrome*, *Explorer*, *Mozilla*...) y escribe en el cuadro de la dirección: **localhost**. Accederás a la página de presentación de *Xampp*. *Localhost* equivale a usar nuestra propia máquina local como servidor de nuestra página web. Es decir, emula la dirección de nuestra web desde nuestro ordenador.
3. En el panel de navegación izquierdo de la página, pulsa sobre: **PhpMyAdmin**. Otra manera de acceder, es escribir directamente, la dirección: *localhost/phpmyadmin*



- Nota: la carpeta virtual *localhost* quedará localizado normalmente en: C:\xampp\htdocs
4. Pulsa en **Base de datos**. Luego escribe en el cuadro de **Crear base de datos** el nombre: **Tienda**. Escoge un *cotejamiento* para el juego de caracteres tipo Unicode *utf8_spanish_ci* o similar (para evitar luego problemas con los acentos o la ñ) y pulsa el botón **Crear**.



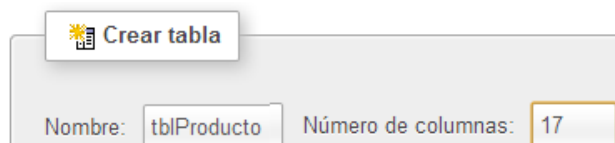
Hemos optado por el nombre **Tienda** porque es un nombre bastante genérico que nos permitirá adaptar este ejemplo fácilmente a comercios de diversos sectores.

2º Crear las tablas.

Crear la tabla Productos.

- Haz clic sobre la base de datos creada **Tienda** y en el recuadro para **Crear tabla** escribe: **tblProducto** N° de columnas 7.
- Pulsa en **Continuar**

Se mostrará la lista donde rellenaremos los campos que formarán la estructura de la tabla.




Rellenar los campos de la tabla

#	Nombre	Tipo	Nulo	Predeterminado	Extra
1	idProducto	int(11)	No	Ninguna	AUTO_INCREMENT
2	strNombre	varchar(100)	Sí	NULL	
3	dblPrecio	double	Sí	NULL	
4	intEstado	int(11)	Sí	NULL	
5	strCategoria	varchar(50)	No		
6	strImagen	varchar(50)	Sí	NULL	
7	strDescripcion	varchar(200)	Sí	NULL	

Índice

PRIMARY

El primer campo **idProducto**, es el más importante ya que se identificará cada producto por este campo. Por lo tanto debe ser único y exclusivo. No se puede repetir. Para ello, activa la casilla **A_I** que indica que será *AutoIncrementable* y **No** actives la casilla **Nulo** porque no podrá estar vacío. También será un campo clave .

Nombre	Tipo	Longitud/Valores	Predeterminado	Nulo	A_I
idProducto	INT	11	Ninguno	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Su ancho en dígitos lo hemos puesto a 11, lo que significa que, de momento, sólo podremos tener 99.999.999.999 productos.

Deberás añadir el resto de los campos que se muestran en la imagen: strRef - strNombre – dbPrecio – intEstado – strCategoria , etc...


Una vez rellenados todos los campos, pulsa el botón: **Guardar**

Notas: Existe la costumbre de añadir al nombre del campo los prefijos: **str** si este se trata de un campo de texto también llamado string o cadena de caracteres - **int** si se trata de un campo numérico entero (integer) o **db** si es numérico decimal doble precisión.

Un campo *Autoincrementable*, también llamado *contador* o *autonumérico*, tiene la propiedad de ir aumentado su valor cada vez añadimos un registro nuevo a la tabla.

Creación del resto de las tablas.

Crear la tabla para las categorías del producto: tblCategoria


- En el recuadro para *Crear tabla* escribe: **tblCategoria** – 2 columnas.
- Pulsa en **Continuar**
- Añade los campos: **idCategoria**  *AutoInc* y **strDescripcion**, con sus características, como se muestran en la imagen.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	idCategoria	int(11)			No	Ninguna	AUTO_INCREMENT
2	strDescripcion	varchar(50)	latin1_swedish_ci		Sí	NULL	

- Una vez rellenados todos los campos, pulsa el botón: **Guardar**

Crear otra tabla para los usuarios: tblUsuario

- En el recuadro para *Crear tabla* escribe: **tblUsuario** 6 columnas - **Continuar**
- Añade los campos de la figura:

idUsuario será del tipo integer, *AutoIncrementable* y campo clave  (primary key);
StrNombre tipo varchar 50 ;
StrEmail tipo varchar 100
intActivo tipo integer

#	Nombre	Tipo	Nulo	Predeterminado	Extra
1	idUsuario	int(11)	No	Ninguna	AUTO_INCREMENT
2	strNombre	varchar(50)	Sí	NULL	
3	strEmail	varchar(100)	Sí	NULL	
4	intActivo	int(11)	Sí	NULL	
5	strPassword	varchar(50)	Sí	NULL	
6	intNivel	smallint(6)	No	0	


Una vez rellenados todos los campos, pulsa el botón: **Guardar**

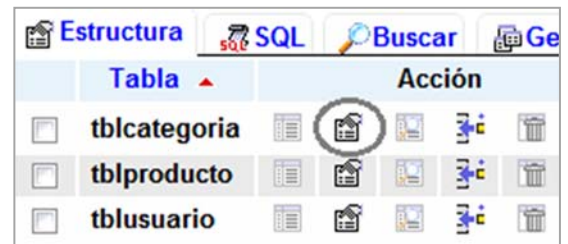
Comprobaciones adicionales.

- Comprobar las claves primarias.

La *clave primaria* es aquella que identificará el elemento de la tabla.

Recuerda asignarla a los campos (idUsuario, idProducto, idCategoria) los atributos de **clave primaria** y *autoincrementable* (A_1)

Para ello puedes acceder a la estructura de cada tabla. Selecciona el campo y pulsa en:  **Primaria**

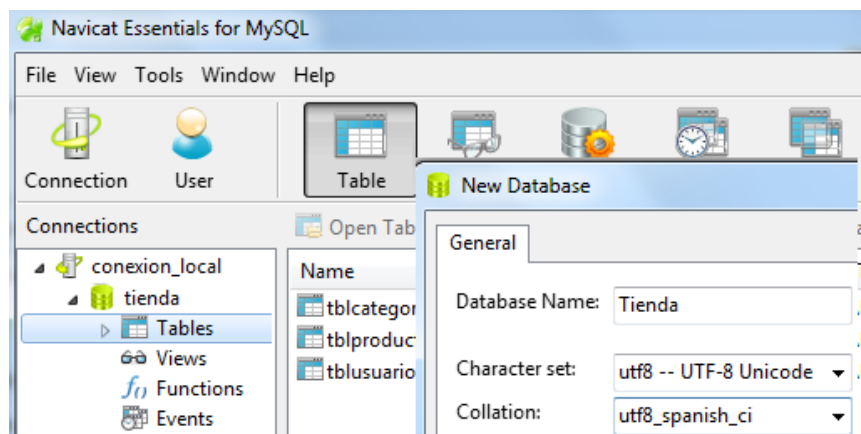


- Las tablas tendrán el tipo de almacenamiento *InnoDB*, esto nos permitirá más adelante crear claves foráneas.

Alternativa opcional 1: Crear la base de datos con el programa Navicat

Navicat es un sencillo programa para crear bases de datos.

- Ejecuta la interfaz para MySQL: *Navicat* (<http://www.navicat.com/>)
- Crea una nueva conexión llamada *conexión_local* con valores predeterminados de fábrica:
 - server: localhost*
 - user: root*
- Doble clic para conectar



Para crear la base de datos:

- Sobre la conexión creada, pulsa botón derecho del mouse
- Escoge **New database** para crea una nueva base de datos.
- En la ventana, escribe el nombre de la base de datos: **Tienda**
- Sobre la base de datos *Tienda*, pulsa en: **New table**

Procede creando las tablas como las anteriores.

Alternativa opcional 2: Crear la base de datos con MySQLWorkBench

MySQLWorkBench es la aplicación de gestión de base de datos MySQL distribuida por **Oracle**, empresa que da soporte a MySQL. Una de las mayores compañías de software del mundo. Su producto original fue la famosa base de datos **Oracle** pero luego compró a **SunSystems**, la creadora del lenguaje *Java*, que también adquirió.

- Para descargar MySQLWorkBench accede a la URL: <http://www.mysql.com/products/workbench/>
- Una vez instalado, entramos y abrimos primero una *conexión local* pulsando doble clic sobre *LocalInstance MySQL55* o pulsa en crear una conexión local:
 - User: **root** Host: **localhost**.
- Escoge del menú: Crear un *New Model*.
- Pulsa doble clic sobre *MyDB* para cambiar el nombre de la base de datos a *Tienda*.
- Pulsa en *Add Table* y añade las tablas con los campos antes mencionados.

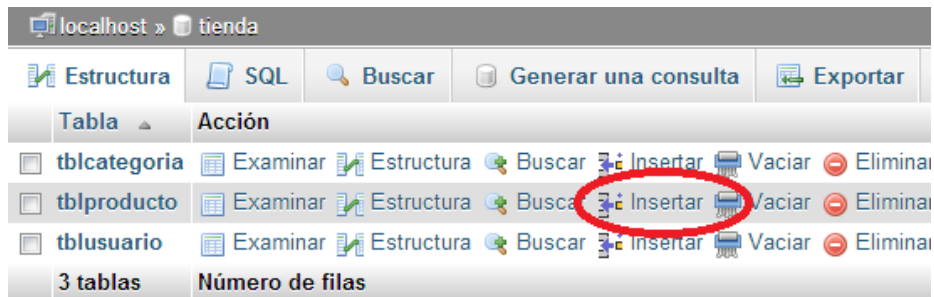


Alternativa opcional 3: Desde la web GenerateData

Para los expertos: esta web permite crear las tablas por código y añadir 100 registros de muestra. Accede a la URL: <http://www.generatedata.com/>

Introducir datos de prueba desde PhpMyadmin

- Con los servicios de *Xampp Apache* y *MySQL* activados, accede a tu navegador web y ejecuta: <localhost/phpmyadmin> para acceder a *PHPMYAdmin*.
- Abre la base de datos *Tienda*.
- Pulsa en **Insertar** en la tabla **tblProducto** como se muestra en la imagen.



Se mostrará la sección para añadir un registro nuevo:

Columna	Tipo	Valor
idProducto	int(11)	<input type="text" value="1"/>
strNombre	varchar(100)	<input type="text" value="Obra nueva en Tarragona"/>
dblPrecio	double	<input type="text" value="157900"/>
intEstado	int(11)	<input type="text" value="1"/>
strCategoria	varchar(50)	<input type="text" value="Vivienda"/>
strImagen	varchar(50)	<input type="text"/>
strDescripcion	varchar(200)	<input type="text" value="Pisos y duplex de obra nueva, con plaza de parking"/>

- Introduce estos tres registros en la tabla de productos *tblProducto*:

idPrc	strNombre	dblPrecio	intEstado	strCategoria	strImagen	strDescripcion
1	Obra nueva en Tarragona	157900	1	Vivienda		Pisos y duplex de obra nueva, con plaza de parkin...
2	Piso cuatro dormitorios con parking	270000	1	Vivienda		Piso en venta, en la zona centro
3	Piso en alquiler en la Parte Alta	350	1	Vivienda		Piso en alquiler siutado en la Parte Alta

Si se tratase de una web de calzado deportivo, los registros serían:

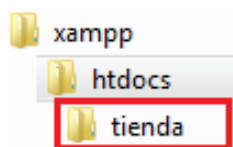
Prc	strNombre	dblPrecio	intEstado	strCategoria	strImagen	strDescripcion
1	Mike aire M-2134	120	1	Deportivas		Deportivas Mike aire azules nº 41
2	Mike tela T-1234	250	1	Deportivas		Deportivas Mike tela verdes nº 39
3	Mike lether L-0934	89	1	Deportivas		Deportivas Mike piel nº 42

- Guarda los datos y cierra la página.

-- Fin de la 1ª sesión --

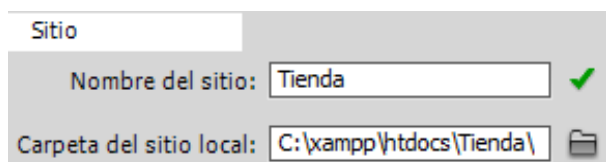
Sesión 2: Dreamweaver - Sitio y plantilla administración

Preparación de las carpetas y el sitio en Dreamweaver.



Creación de la carpeta de trabajo.

- Activados los servicios *Apache* y *MySQL* (con el *Xampp* o *Wamp*) y desde el *explorador de archivos*, abre la carpeta donde tienes asignado el *localhost*, (...xampp\htdocs o ...wamp\www)
- Crea dentro una nueva carpeta llamada: **tienda**







Definición del sitio y carpeta de trabajo local

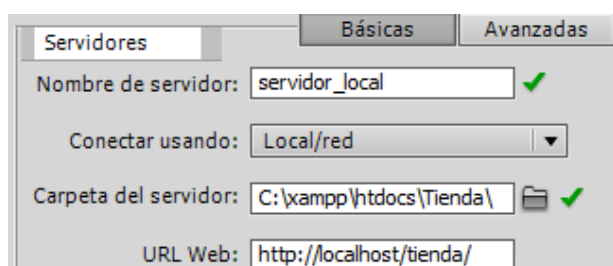
- Ejecuta *DreamWeaver* (DW), y escoge del menú: *Sitio* ▶ **Nuevo sitio...**

Nombre del sitio: **Tienda**

En carpeta del sitio local, asigna la carpeta anterior de **Tienda** (no cierres la ventana)

Asignación de la carpeta del servidor de prueba y su dirección web local

- Pulsa en el apartado *Servidores* y luego sobre el signo **+** Añadir nuevo servidor    . Introduce los valores de la imagen. Importante que la URL web incluya: `http://`. Activa el servidor como De Prueba. Guarda el sitio.



Archivos locales



Creación de las subcarpetas.

- En el panel de *Archivos* (F8), pulsa el botón derecho del mouse sobre la carpeta *Tienda* y escoge del menú contextual: *Nueva carpeta*.
- Crea en este sitio las siguientes subcarpetas: **Admin – Images – Productos – Estilos**

Área de administración (back-end)

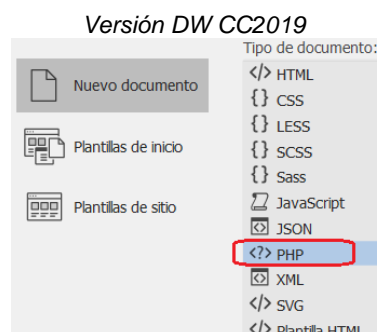
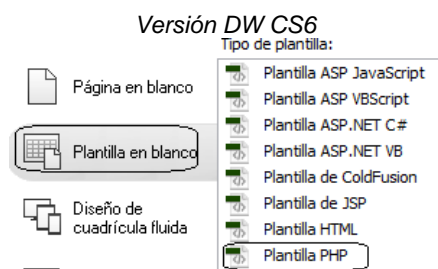
En esta sección realizaremos una serie de páginas que se encargarán de la gestión y mantenimiento de los datos de la página, (parte trasera).

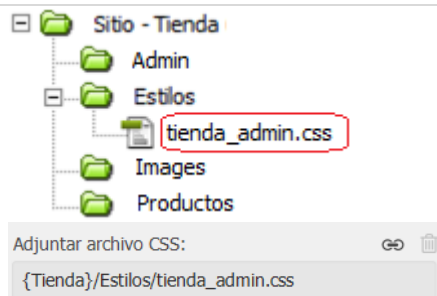
No serán visibles por el usuario normal pero sí serán accesibles mediante una autenticación previa de los usuarios con privilegios de administrador, que crearemos posteriormente.



Crear la plantilla base para las páginas de administración

Las plantillas permiten utilizar elementos comunes para todas las páginas web.

- Escoge del menú: **Nuevo - Plantilla en blanco – Tipo de plantilla: Plantilla PHP**
O Nuevo documento - PHP










- **En la versión CS:** Escoge 1 columna fija encabezado y pie de página. Activa el diseño CSS en: **Crear nuevo archivo, - Crear**. Nombre: **tienda_admin.css**
Pulsa en el panel de Archivos el botón **Actualizar**  y comprueba la creación de la carpeta de estilos
- **En la versión CC:** escoge  adjuntar archivo CSS
- Indica la carpeta para los estilos llamada **Estilos** en el sitio, y dentro de esta, deberás guardar el archivo del estilo el nombre: **tienda_admin.css**.



Otro modo de crear plantillas: Desde el panel **Activos**.

En el panel **Activos** dispones del botón **Plantillas**  y los botones inferiores     que permiten crear, editar o eliminar plantillas.

El archivo de estilo tienda-admin.css

```
@charset "utf-8";
body {
    font:
    background-color: #42413C;
    margin: 0;
    padding: 0;
    color: #666;
    font-family: Tahoma, Geneva, sans-serif;
    font-size: 12px;
}

h1, h2, h3, h4, h5, h6, p {
    margin-top: 0; /* quita el margen superior */
    padding-right: 15px;
    padding-left: 15px;
    text-align: center;
    font-size: 10pt;
}

a img { /* elimina el borde azul */
    border: none;
}

a:link {
    color: #42413C;
    text-decoration: underline; /* subrayado */
}

a:visited {
    color: #6E6C64;
    text-decoration: underline;
}

a:hover, a:active, a:focus { /* mismo efecto con
teclado al paso por encima del ratón. */
    text-decoration: none;
}

.container { /* contenedor fijo rodea a las demás*/
    width: 960px;
    background-color: #FFF;
    margin: 0 auto; /* el valor automático de los
lados, unido a la anchura, centra el diseño */
}

.header {
    background-color: #ADB96E;
}

.content {
    padding: 10px 0;
    text-align: center;
}

.footer { /* clase pie de página */
    padding: 10px 0;
    background-color: #CCC49F;
    font-family: Tahoma, Geneva, sans-serif;
    font-size: 10px;
    color: #FFF;
}

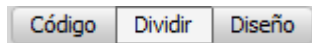
/* clases flotantes */
.flrt { /* para que flote a la derecha */
    float: right;
    margin-left: 8px;
}

.flflt { /* para que flote a la izquierda */
    float: left;
    margin-right: 8px;
}

.container .content table tr td {
    font-weight: normal;
    text-align: left;
    font-family: Tahoma, Geneva, sans-serif;
    font-size: 10pt;
}

.Centrado {
    text-align: center;
}
```

Editar la plantilla de administración.



```
<!doctype html>
<html>
<head>
<meta charset=" ISO-8859-1">
<!-- TemplateBeginEditable name="doctype" -->
<title> Administración </title>
<!-- TemplateEndEditable -->
<link href="../Estilo/tienda_admin.css"
rel="stylesheet" type="text/css" />
</head>
```

- **Versión CS:** Con la página de plantilla creada, pulsa en el modo de vista: **Dividir**. Observa, en el panel izquierdo, el código generado.
- **Versión CC:** Escribe el código del panel izquierdo.

Charset.

- Bajo la zona del `<head>`, se recomienda cambiar en el código charset UTF-8 a ISO-8859-1 (para uso de ñ).

Título de documento

- Observa ver cómo creamos una **región editable** para poder modificar el título del documento.
- Pon en el título: **Administración**

Observa el link externo a la página de estilos

Guardar la plantilla de administración

- Escoge del menú: **Archivo – Guardar como plantilla...**
- **Vers. CS:** Rellena los datos de la 1ª imagen contigua.
- **Vers. CC:** Guarda esta plantilla en la carpeta **Templates** con el nombre: **Admin.dwt.php**

Esta plantilla sirve para hacer diversas páginas web de la parte de administración, con estilos y estructuras comunes.

Regiones editables en plantillas

Las regiones editables en plantillas (*TemplateEditable*) son las zonas de la plantilla (*template*) donde se podrá escribir en las diversas páginas que se creen a partir de esta plantilla. Mientras en el resto la plantilla, no se podrá modificar nada desde las páginas, porque serán zonas *comunes* a todas ellas.

La ventaja es que, con sólo modificar un área común de la plantilla, se actualizarán todas las páginas dependientes de ella.

Crear una región editable

```
<body>
<div class="container">
  <div class="header"><a href="#"></a>
  <!-- end .header --></div>
  <div class="content"><!--
TemplateBeginEditable name="Cuerpo" --
>
  <h1>Administración</h1>
  <p>Esta es la plantilla base para la
creación de las páginas de
administración</p>
  <!-- TemplateEndEditable --><!-- end
.content --></div>
  <div class="footer">
  <p>Pie de página</p>
  <!-- end .footer --></div>
  <!-- end .container --></div>
</body>
</html>
```

- Cambia o añade el texto del panel izquierdo.
- Para añadir la región editable, puedes escoger del menú de **Dreamweaver** el elemento de plantilla: **Región editable (TemplateEditable)** mediante: **Insertar ▶ Plantilla ▶ Región editable**, con el nombre: **Cuerpo**. ↵
- En esta región editable, puedes escribir:

“Esta es la plantilla base para la creación de las páginas de administración”.
- Pulsa la tecla [F12] para ver la página en tu navegador. Guarda la plantilla **Admin** antes.

Insert_logo (180 x 90)

Administración

Contenido

Esta es la plantilla base para la creación de las páginas de administración.

Pie de página

Crear el archivo Index.php para la zona de administración (back-end o parte trasera de la tienda)

- Escoge: Archivo - nuevo – **Página de plantilla**.
- Usamos la plantilla del sitio *Tienda* llamada **Admin.dwt.php** que hemos creado antes. Observa que, si activas el modo de vista **dividir**, la mayoría del código se muestra en gris. Este código no se puede cambiar porque pertenece a la plantilla base, común a todas las páginas, y que sólo nos deja escribir dentro de la *región editable* llamada *contenido*.
- En la región editable escribe: **Bienvenido a la zona de Administración. Seleccione una opción.**
- Guarda la página (Archivo ▶ Guardar como...) en la carpeta **Admin** con el nombre: **index.php**.
- Para ver cómo queda, abre tu navegador y escribe la siguiente dirección: <http://localhost/tienda/admin>
- Comprueba que se muestra correctamente la página web [F12]. Recuerda tener activado el servidor (Wamp **W** o Xampp) y que se ejecute como *administrador*.
- En el caso de que la página *Index* no haya actualizado bien la hoja de estilos *tienda.css* o se haya desvinculado de la plantilla *Admin*, escoge del menú: Modificar ▶ Plantillas ▶ **Aplicar plantilla a página...** Selecciona la plantilla *Admin* para volver a vincularla.

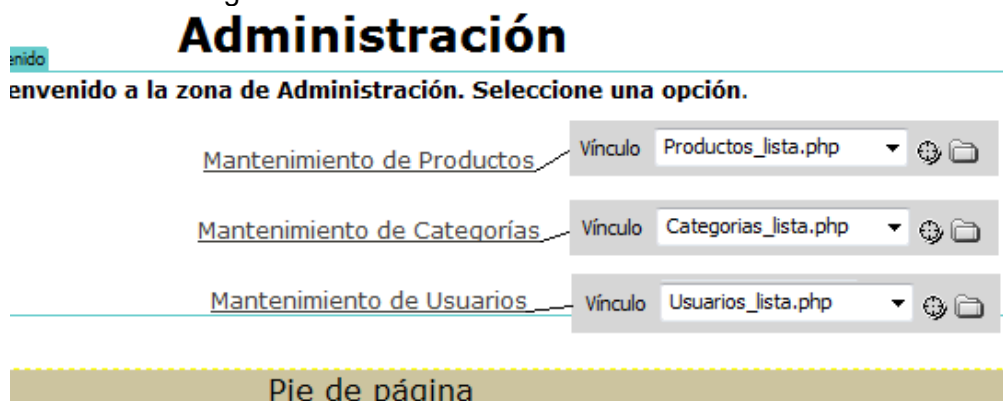
Modificar el archivo Index.php. Asignar los vínculos.

- Abre la página *index.php*,
- Bajo el texto de bienvenida, añade las líneas:
- Asigna a cada uno los vínculos:

Mantenimiento de Productos
Mantenimiento de Categorías
Mantenimiento de Usuarios

Productos_lista.php
Categorias_lista.php
Usuarios_lista.php

Como se muestra en la imagen...



- Guarda de nuevo la página [Ctrl+S]
- Comprueba su aspecto pulsando F12 en *DreamWeaver*. Ten en cuenta que los vínculos aún no funcionan. Faltan las páginas vinculadas.

Nota: Si no se muestra la página desde la vista previa de DW es porque en la configuración local no pusiste la dirección URL web correcta: <http://localhost/tienda/>, aunque sí la podrías ver desde tu navegador, escribiendo la dirección correcta: <http://localhost/tienda/admin/>

Sesión 3: Páginas de administración de productos

Crear la página de lista de productos



- Para ir más rápido, puedes duplicar la página *index.php* y renombrar por **productos_lista.php**
- Si es necesario, actualiza los vínculos con la plantilla *Admin*
- Abre esta página de productos y cambia el Texto por: **Lista de productos**

Conectar la página con la base de datos:

- Ten activado el servicio apache y MySql de Xamp,
- Comprueba que tienes un *Servidor de prueba* activo

Versión CS: Activa la *Ventana ▶ Base de datos* y la pestaña:

Vinculaciones (Bindings)

- En el segundo elemento: **Tipo de documento:** Escoge PHP
- En el tercer elemento: **Servidor de prueba** y añade el servidor con los datos de la figura:
 - Nombre del servidor: `servidor_local`
 - Conectar usando: `Local/red`
 - Carpeta del servidor: `c:\xampp\htdocs\tienda\` (sin espacios)
 - URL web: `http://localhost/tienda`

La dirección del *servidor de prueba* debe coincidir con la que hayas puesto en el sitio.

- En el panel *Base de datos* pulsa en **+** y escoge: **Conexión MySQL**

Te aparecerá la ventana donde deberás Definir una *Nueva conexión*.

- Pon los datos:
 - Servidor MySQL: `localhost`
 - Usuario: `root`
 - Contraseña: `en blanco.`
 - Base de datos: `tienda` Pulsa *Aceptar*

Si la conexión es correcta, se mostrarán las tablas en la parte inferior de la ventana de base de datos.

- En el panel *Vinculaciones (Bindings)* pulsa en **+** y escoge: **Juego de registros**
- *En la ventana*, activa el modo *Avanzada...*
- Despliega la base de datos y selecciona la tabla **tblproducto**,
- Pulsa en **SELECT** crea la consulta en SQL: `SELECT * FROM tblProducto.`
 Selecciona el campo de la tabla: *StrNombre* y pulsa en **ORDER BY** lo que nos añadirá:
`ORDER BY tblProducto.strNombre` para ordenarlo por el nombre.
- Pulsa *Aceptar*.

Versión CC:

En Dreamweaver CC, los comportamientos del servidor están disponibles como extensión, *Deprecated: Server Behavior & Database*

Desde: Adobe.com/addons o desde exchange.adobe.com Buscar el complemento: *Server Behavior & Database* después de instalar la extensión, la función de comportamientos del servidor se activa y las extensiones que dependen de ella funcionan como lo hacían anteriormente. En caso contrario se deberá crear manualmente por código.

- ✓ 1. Cree un [sitio](#) para este archivo.
- ✓ 2. Elija un [tipo de documento](#).
- ✓ 3. Configure el [servidor de prueba](#) del sitio.

Nombre de servidor: ✓

Conectar usando:

Carpeta del servidor: ✓

URL Web:

Bases de datos Vinculaciones Comp

+

Tipo de documento: PHP

Conexión MySQL

Nombre de conexión:

Servidor MySQL:

Nombre de usuario:

Contraseña:

Base de datos:

Bases de datos Vinculaciones Comportamientos del s

+

Tipo de documento: PHP

Juego de registros (consulta)

Nombre:

Conexión: Definir...

SQL: `SELECT * FROM tblproducto ORDER BY tblproducto.strNombre`

de datos: Tablas

- tblcategoria
- tblproducto
- idProducto

Crea el archivo de conexión: *Conexión_local.php* dentro de la carpeta *Connections*:

```
<?php
$conexion_local = mysqli_connect("localhost", "root", ""); //servidor – usuario - contraseña
if (!$conexion_local) { echo "Error: No se pudo conectar a MySQL." . PHP_EOL;
    exit; }
echo "Conexion correcta a MySQL" . PHP_EOL;
?>
```

En el código fuente del archivo *Productos_lista.PHP* escribe:

```
//mysqli_select_db($database_conexion_local, $conexion_local);
mysqli_select_db($link, "Conexion_Local");
$query_Recordset1 = sprintf("SELECT * FROM tblproducto WHERE strCategoria = %s ORDER BY
strNombre ASC", GetSQLValueString($colname_Recordset1, "text"));
```

Ejemplo de conexión y consulta de datos a la base de datos Tienda en un solo archivo PHP.

```
<?php
$servername = "localhost";
$dbname = "tienda";
$username = "root";
$password = "";

// creamos la conexión...
$conexion = mysqli_connect($servername, $username, $password, $database);

// comprobamos la conexión...
if (!$conexion) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";

//seleccionamos la conexión.....

$db = mysqli_select_db( $conexion, $database ) or die ( "No se ha podido conectar" );

//consultamos una tabla....
$query = "SELECT * FROM tblProducto";
$resultado = mysqli_query( $conexion, $query ) or die ( "Algo ha ido mal en la consulta");

//Mostrar los datos....

echo "<table borde='2'>";
echo "<tr>";
echo "<th>Nombre</th>";
echo "<th>Precio</th>";
echo "</tr>";
while ($columna = mysqli_fetch_array( $resultado ))
{
    echo "<tr>";
    echo "<td>" . $columna['strNombre'] . "</td><td>" . $columna['dblPrecio'] . "</td>";
    echo "</tr>";
    echo "</table>";
}

// cerramos la conexión...
mysqli_close( $conexion );
?>
```

Un poco de teoría. El lenguaje SQL

Dado que utilizaremos con frecuencia este lenguaje para consultas y manipulación de datos, es importante conocer las 10 órdenes básicas de este lenguaje para entender mejor las consultas.

1. **SELECT**: Extrae datos de una o más tablas indicando los campos (o columnas) separados por comas.
El asterisco (*) sustituye a todos los campos
Ejemplo: `SELECT * FROM CLIENTES WHERE CIUDAD = "SALOU" AND PROVINCIA = "TARRAGONA"`
2. **FROM**: Especifica la tabla o tablas (separadas por comas) de la cual se extraen los datos.
3. **JOIN**: Junta varias tablas a través del campo índice, que existe en las dos tablas.
4. **WHERE**: Filtra los registros que cumplan el criterio de condición.
Estos pueden utilizar los operadores de comparación (=, <, > ...) o predicados adicionales IN, ANY, ALL, EXISTS.
5. **ORDER BY**: Especifica el orden de extracción. *Ejemplo*: `SELECT * FROM CLIENTES WHERE EDAD >18 ORDER BY NOMBRE ASC`
6. **GROUP BY**: Especifica cómo se agrupan las filas o registros.
7. **INSERT**: Añade un registro a la tabla.
Ejemplo: `INSERT INTO FACTURAS (FACNO, NOMBRECLIENTE, CIUDAD) VALUES (99, 'Mariano Rajoy', Madrid)`
8. **APPEND** Inserta un registro al final de la tabla:
9. **UPDATE**: Modifica/actualiza los datos de la tabla.
10. **DELETE**: Borra datos de la tabla. Ej: `DELETE FROM tblproducto WHERE idProducto = 23`

Consulta de datos en Dreamweaver:

“Las consultas de datos, en Dreamweaver, se llaman Juego de registros o recordSet”

Observa cómo en el código de la web nos ha agregado la conexión a la base de datos

Dentro de las etiquetas: `<?php ?>` indicadores que en su interior se ejecuta el código PHP

Tenemos la consulta SQL: `$query_Recordset1 = "SELECT * FROM tblproducto ORDER BY tblproducto.strNombre";`

Que se traduce por: “Selecciona todos los registros (comodín *) de la tabla de productos ordenados por el campo del nombre”

Añadir los campos en la página

- Inserta bajo el texto una nueva tabla (Insertar ▶ Tabla) de 2 filas x 4 columnas (como en la imagen)
- Escribe en la primera fila los encabezados: *Producto, Nombre, Estado y acciones.*
- Arrastra desde el panel de vinculaciones a las celdas de la segunda fila, los campos *idProducto, StrNombre e IntEstado,*
- En la tercer escribe: *Nuevo – Modificar - Borrar*

Administración

Contenido

Lista de productos

Producto	Nombre	Estado	Acciones
{Recordset1.idProducto}	{Recordset1.strNombre}	{Recordset1.intEstado}	Nuevo - Modificar - Borrar



Bases de datos Vinculac

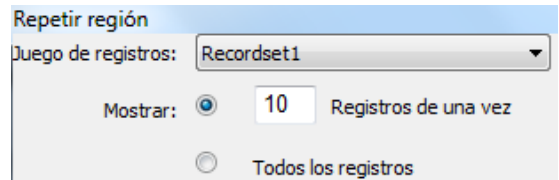
+ - Tipo de docum

Fuente


- ⊞ Juego de registros
- ⚡ idProducto
- ⚡ strNombre
- ⚡ dbPrecio
- ⚡ intEstado
- ⚡ strCategoria
- ⚡ strImagen

Para mostrar los registros de 10 en 10:

- Selecciona la 2ª fila de la tabla (mejor en código o en la barra de estado pulsa en `<tr>`):

- En el panel de *Base de datos*, pestaña **Comportamientos del servidor** [Ctrl+F9], (*Server Behaviors*)
- Pulsa sobre  y escoge: **Repetir región**
- En la ventana, selecciona Mostrar 10 registros de una vez. Pulsa Aceptar
- Si observas el código, verás cómo se ha insertado un bucle: **do ...while** con un `MaxRows_recordset1` a 10.
- Guarda la página **productos_lista.php**.



Añadir un contador de registros

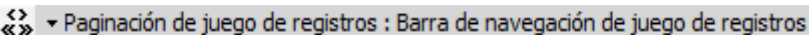
- En el diseño de la web, *productos_lista.php*, sitúa el cursor debajo de la tabla de datos.
- Escribe el texto: **Total registros:**
- En la venta de **Comportamientos del servidor** [Ctrl+F9], pulsa sobre  y escoge: **Mostrar recuento de registros** ▶ **Mostrar el total de registros**. (Mostrará la variable: `$totalRows_Recordset1`)

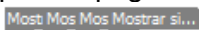
Nombre	Estado
<code>cordset1.strNombre</code>	<code>{Recordset1.intEstad</code>

Total registros: `{Recordset1.TotalRecords}`

Añadir botones de navegación

Si el número de registros es superior a 10, deberíamos poder cambiar a la siguiente página. Por ello sería interesante que para comprobar este ejercicio, tuvieras, al menos 11 registros. Recuerda que para añadir registros, aún no hemos hecho la página para añadir registros. Por ello deberás acudir a un gestor de datos como *PhPMyAdmin* o similar y añadir más registros como se explicó en la 1ª sesión.

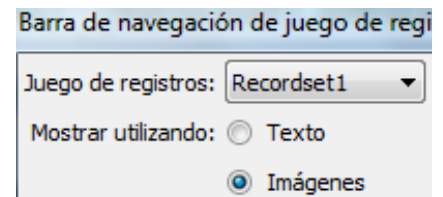
- En el diseño de la web, *productos_lista.php*, sitúa el cursor debajo de la tabla de datos a la derecha del total de registros..
- En la venta **Insertar** escoge las herramientas **Datos** ▼. Pulsa sobre:
 **Paginación de juego de registros : Barra de navegación de juego de registros**
Se mostrará una tabla conteniendo 4 iconos para desplazarse por las páginas de datos





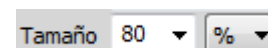
Si lo prefieres, selecciona la tabla de navegación y escoge: Alineación centro.

- Si el texto de la tabla te parece algo grande, puedes reducir el tamaño de la fuente en porcentaje:
- Comprueba [F12] y guarda la página: **productos_lista.php**
En la carpeta *Admin* del sitio Tienda, te aparecerán las 4 imágenes gif de los botones de navegación.



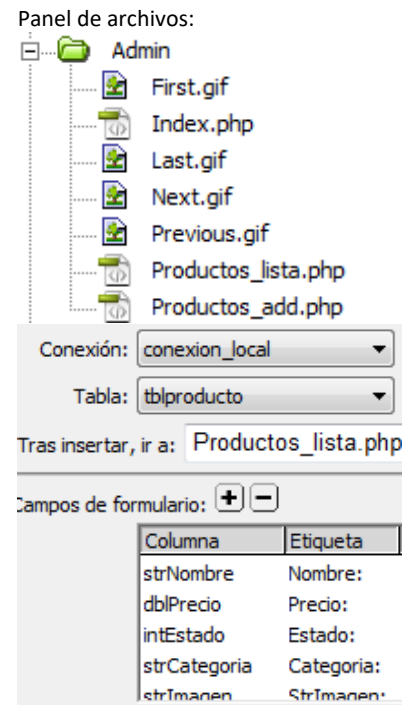
parking	1	N
de Alta	1	N
ciones	1	N

Total registros: 11



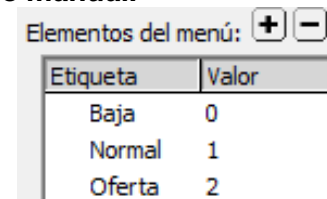
Crear página para añadir producto.

- Para ir más rápido, puedes copiar la página *index.php* desde el panel de *Archivos*, carpeta *Admin*, [ctrl+C – Ctrl+V] y cambiarla por **productos_add.php**
 - Si es necesario, *actualiza* los vínculos con la plantilla *Admin*
 - Abre esta página de productos y cambia el Texto por: **Añadir nuevo producto**
- Pulsa [Intro] ↵ para situar el cursor en la fila bajo el texto.
 - Escoge del menú:
 - Insertar ▶ Objeto de datos ▶ Insertar registro ▶ **Asistente de formulario de inserción de registros.**
 - En la ventana del asistente, selecciona la tabla de productos *tblproducto* y añade todos los campos de la tabla productos y quita (-) el *IdProducto* que es del tipo contador.
 - Cambia las etiquetas de los campos por nombres que se muestran en la imagen.
 - En el cuadro: *Trans insertar ir a:* Examina y busca la ubicación de la página: *Productos_lista.php* que será la página de retorno, desde donde se llamó a añadir producto.



Creación de un cuadro combinado con una lista de valores, tipo manual.

- En el campo *intEstado* escoge *Mostrar como menú* y pulsa en *Propiedades de menú.*
- Añade los elementos:
 - Baja (0) – Normal (1) – Oferta (2) como en la imagen.
- Pulsa *Aceptar* y aparecerá una tabla de datos dentro del formulario "form1" en la página:



```
<form action="<?php echo $editFormAction; ?>"
method="post" name="form1" id="form1">
```

- En la *propiedades* de botón del formulario, cambia el texto del botón: *Insertar registro* por: **Añadir Producto.**
- Por último, previsualiza [F12] y guarda la página *Productos_add.php*

Si observas el código, verás la línea: **INSERT INTO** *tblproducto* (*strNombre*, *dblPrecio*, *intEstado*, *strCategoria*, *strImagen*, *strDescripcion*) **VALUES** (*%s*, *%s*, *%s*, *%s*, *%s*, *%s*)" donde añade un registro a la tabla con los valores de los campos que se han recogido en el formulario *form1*

Administración

Contenido

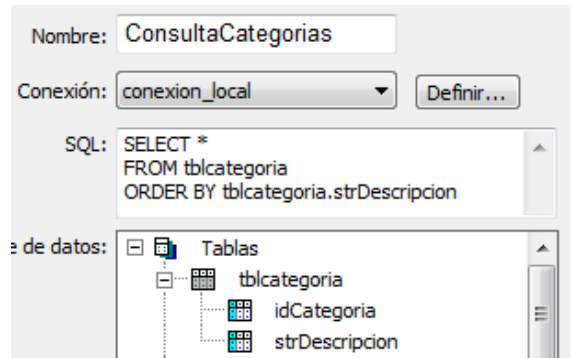
Añadir nuevo producto

Nombre:	<input type="text"/>
Precio:	<input type="text"/>
Estado:	Normal <input type="button" value="v"/>
Categoría:	<input type="text"/>
Imagen:	<input type="text"/>
Descripción:	<input type="text"/>
<input type="button" value="Añadir producto"/>	

Creación de un cuadro combinado con una lista de valores, tipo dinámica.

Primero crearemos un *juego de registros* (consulta) para poder extraer las categorías de la tabla categorías:

- En el panel *Vinculaciones* (*Bindings*) pulsa en **+** y escoge: **Juego de registros**
- En la ventana, activa el modo *Avanzada...*
- Despliega la base de datos y selecciona la tabla **tblcategoria**,
- Pulsa en **SELECT** para crear crea la consulta en SQL.
- Luego haz clic sobre el campo *strDescripcion* y pulsa el botón **ORDER BY** para ordenar por la descripción

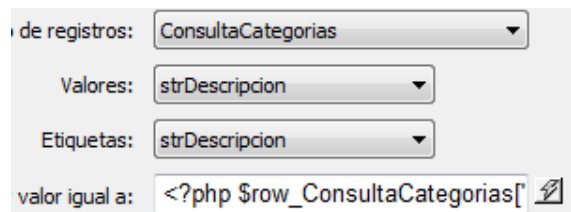


Se mostrará la siguiente consulta SQL:

*SELECT * FROM tblcategoria ORDER BY tblcategoria.strDescripcion*

Cambiar el campo categoría

- Primero borra el cuadro de *Categoría* del formulario.
- Luego, añade en el hueco de la categoría, un menú desplegable, escogiendo del menú: *Insertar* ▶ *Formulario* ▶ **Seleccionar lista de menú**. Se mostrará un cuadro de lista desplegable. Selecciónalo haciendo clic encima. En las *propiedades* de la *lista menú* cambia el nombre por: *strCategoría*
- En las *propiedades* de la *lista menú*, pulsa en **Dinámico...** y escoge el juego de registros *ConsultaCategorias*. Valores: *strDescripcion*.
- En *Selecciona valor igual a:* pulsa en **⚡** y escoge el campo: *strDescripcion*. Si lo pruebas, es normal que no se muestre ninguna categoría, porque aún no tenemos ningún registro de categorías.
- Para comprobar su funcionamiento, sería interesante añadir unos valores a la tabla de categorías. Para ello, desde tu navegador web, ejecuta: *localhost/phpmyadmin* para acceder a *PHPMYAdmin*. Abre la base de datos *Tienda*, selecciona la tabla *tblCategoría* y pulsa en *Insertar* para añadir algunos datos a la tabla. Por ejemplo: *Vivienda*, *Chalet*, *Piso*, *Finca*, *Ático*, etc...



Crear los enlaces o vínculos

Vínculos en la página *lista de productos*:

- Abre la página *productos_lista.php*
- Añade bajo los botones de navegación, el texto: **Volver**
- Enlaza el texto con la página *index.php*.
- Asigna los enlaces a las acciones como se muestra en la imagen:



Nuevo: Productos_add.php
 Modificar: Productos_edit.php
 Borrar: Productos_delete.php

- Guarda de nuevo la página [Ctrl+S]
- Comprueba el funcionamiento de los vínculos pulsando F12 en *DreamWeaver*.
- Ten en cuenta que las páginas para editar y borrar aún no las hemos hecho. No funcionan temporalmente (en construcción).

Vínculos en la página *añadir producto*:

- Abre la página ***productos_add.php***,
- Bajo el formulario añade el texto:
Lista de productos
Asígnale el vínculo: *productos_lista.php*

Atajo: También puedes copiar el vínculo *Lista de productos* que creaste en la página *Index.php*



Crear página para modificar producto.

- Desde el panel de *Archivos* [F8], dentro de la carpeta *Admin*, crea una copia de ***Index.php*** (Ctrl+C - Ctrl+V).
- Cambia el nombre de la copia de index a ***Productos_edit.php***.
- Abre la página y cambia el título por: ***Modificar producto***, borrando el resto del texto de la página.

Modificar el vínculo:

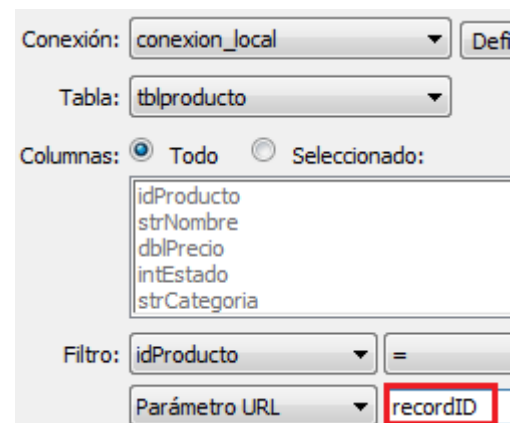
- Abre también la página ***Productos_lista.php*** de la carpeta *Admin*.
- Selecciona el texto: ***Modificar*** y busca el vínculo a la página antes creada: *productos_edit*. Para modificar, debemos especificar de qué producto se trata, por ello debemos añadirle al vínculo el parámetro: ***?recordID=*** y a continuación arrastramos el campo del *idProducto* desde el panel de vínculos de la base de datos, quedando así el vínculo (el texto todo junto entre comillas es el vínculo):

```
<a href="Productos_edit.php?recordID= <?php echo $row_recordset1['idProducto']; ?>">Modificar</a> -
```

- Guarda la página ***Productos_lista.php*** [Ctrl+S]

Crear las consultas o juegos de registros

- Vuelve a la página de ***Productos_edit.php***
- En la ventana de *Base de datos*, pestaña ***Vinculaciones*** [Ctrl+F10], pulsa en ***+*** y crea un nuevo ***juego de registros*** (consulta) *Recordset1*
- Esta vez, utilizaremos el modo de vista *Simple*.
- En la ventana, selecciona la conexión: *conexión_local* y la tabla: *tblProducto*.
- Si pulsas en *Avanzada* observarás que genera la siguiente consulta SQL:
SELECT *FROM tblproductoWHERE idProducto = colname
Donde **colname** es la variable que recoge el valor de **recordID**
Este filtro recordID es imprescindible para que localice el registro a modificar.
- Pulsa *Aceptar* para cerrar la ventana.



También usaremos otra consulta para categorías:

- En la ventana de *Base de datos*, pestaña **Vinculaciones** [Ctrl+F10], pulsa en **+** y crea un nuevo **juego de registros** (consulta) *Recordset2*
- En la ventana, selecciona la conexión: *conexión_local* y la tabla: *tblCategoria*.
- Ordenar por *strDescripcion* en Ascendente. Como en la imagen.
- Pulsa Aceptar. Esto genera la siguiente consulta SQL:
`SELECT * FROM tblCategoria ORDER BY tblCategoria.strDescripcion`

Tabla: **tblCategoria**

Columnas: Todo Seleccionado:

idCategoria
strDescripcion

Filtro: Ninguna =

Parámetro URL

Ordenar: strDescripcion Ascendente

Crear el formulario para modificar registros

- Escoge del menú:
Insertar ▶ *Objeto de datos* ▶ *Actualizar registro* ▶ **Asistente de formulario de actualización de registros**
- Rellena los datos como en la imagen

Tabla a actualizar: **tblproducto**

Seleccionar registro de: **Recordset1**

Campo de clave exclusiva: **idProducto** numérico

Después de la actualización, ir a: **Productos_lista.php** **Examinar...**

Columna	Etiqueta	Mostrar como	Enviar como
strNombre	Nombre:	Campo de texto	Texto
dblPrecio	Precio:	Campo de texto	Doble
intEstado	Estado:	Menú	numérico
strCategoria	Categoría:	Menú	numérico

Etiqueta: **Categoría:**

Mostrar como: **Menú**

Propiedades de menú

Rellenar elementos de menú: Manualmente De base de datos

Juego de registros: **Recordset2**

Obtener etiquetas de: **strDescripcion**

Obtener valores de: **strDescripcion**

Seleccionar valor igual a: **<?php echo htmlentities(\$rc**

- En la etiqueta puedes quitar las letras el prefijo tipo de campo.
- Elimina **-** el campo: *IDProducto*.
- En el campo *intEstado* escoge Mostrar como **menú** y pulsa en *Propiedades de menú*.
- Añade los elementos:
Baja (0) – Normal (1) – Oferta (2) como en el formulario de alta de producto..
- Poner en el campo *categoria* el modo *Mostrar como: Menú* y en sus *Propiedades* especificar como en la imagen:
- Pulsa *Aceptar* y aparecerá el formulario en la página
- Guarda **Productos_edit.php** y comprueba desde la lista de productos.

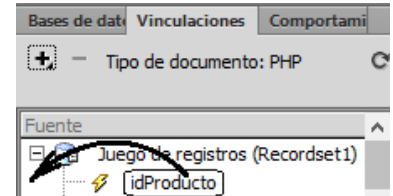
Crear la página para eliminar productos

- Desde el panel de *Archivos* [F8], dentro de la carpeta *Admin*, crea una copia de *Index.php* (Ctrl+C - Ctrl+V).
- Cambia el nombre de la copia de index a *Productos_delete.php*.
- Abre la página y cambia el título por: **Eliminado producto**, borrando el resto del texto de la página.

Modificar el vínculo:

- Abre también la página *Productos_lista.php* de la carpeta *Admin*.
- Selecciona el texto: **Borrar** y busca el vínculo a la página antes creada: *Productos_delete*.

Antes de eliminar, debemos especificar de qué producto se trata, por ello debemos añadirle al vínculo el parámetro: **?recordID=** y a continuación arrastra el campo del *idProducto* desde el panel de *vinculaciones* de la base de datos y suéltalo en el código, quedando así el vínculo (el texto todo junto entre comillas es el vínculo):



```
<a href="Productos_delete.php?recordID=<?php echo $row_recordset1['idProducto']; ?>">Borrar</a>
```

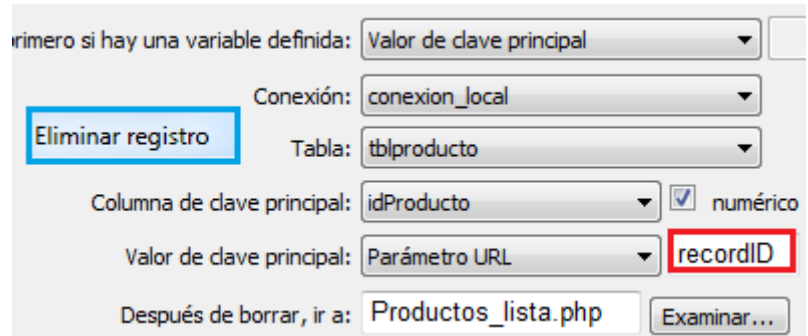
- Guarda la página *Productos_lista.php* [Ctrl+S]

Eliminar registro

- Volvemos a la página de *Productos_delete.php*
- Escoge del menú: *Insertar* ▶ *Objeto de datos* ▶ **Eliminar registro**

Rellena los datos como en la imagen adjunta

No hemos necesitado crear una consulta o juego de registros en esta página. El asistente para eliminar registro se encarga de ello, e introduce el código SQL:



"DELETE FROM tblproducto WHERE idProducto=%s"

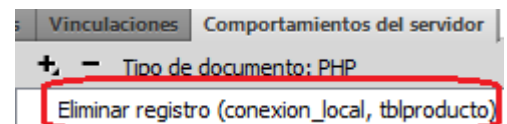
Donde el parámetro que le pasamos **%s** que es el parámetro URL: recordID

- Guarda la página *Productos_delete.php* y comprueba su funcionamiento. Desde la página de lista de productos elimina un producto de la lista.

Cambiar comportamiento *Eliminar registro*.

Si la página retorna a *Productos_lista.php* después de eliminar el registro, apenas si se muestra la página de borrado del producto.

- Para cambiar este comportamiento, en el panel de *Comportamientos del servidor* (Server Behaviors), pulsa doble clic sobre el comportamiento: *Eliminar registro*
- Eliminar el texto del vínculo *Después de borrar, ir a:* *Productos_lista.php*




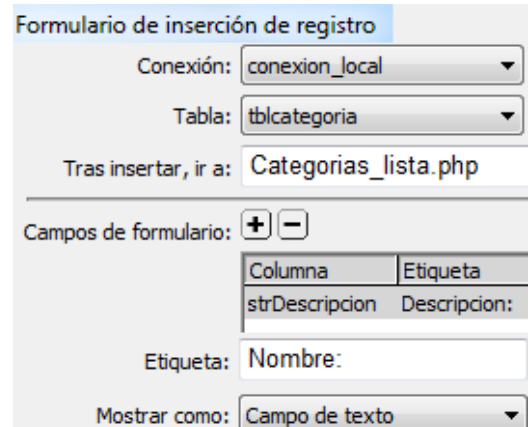
- Pulsa *Aceptar*, para guardar los cambios.
- Añade al final de la página el texto **Volver** y asigne el vínculo: *Productos_lista.php*. Guarda la página *Productos_delete.php* y comprueba su funcionamiento.

Sesión 4 – Páginas de administración de categorías

Crear página para añadir categorías

Desde el *panel de Archivos* [F8], en la carpeta *Admin*, crea una copia de *index.php* (Ctrl+C - Ctrl+V).

- Cambia el nombre de la copia de index a:
categorias_add.php.
- Abre esta nueva página y cambia el título por: *Crear categoría*
- Escoge del menú:
Insertar ▶ Objeto de datos ▶ Insertar registro ▶ Asistente de formulario de inserción de registros
- Rellena los datos como en la imagen adjunta
 - Quita, pulsando en  el campo *IdCategoría* (no se necesita y es autonumérico)
 - Cambia la etiqueta para el campo *strDescripcion* por: *Nombre*:
 - *Tras insertar* mostraremos a la página *Categorias_lista* (que aún no está hecha)
- Pulsa *Aceptar* para insertar el formulario.



Observa que no hemos necesitado crear ninguna consulta o juego de registros para esta página web ya que la conexión con la base de datos ya la proporciona el asistente: **INSERT INTO tblcategoria (strDescripcion) VALUES (%s)**

Modificaciones en la página:

- Añade bajo el formulario el vínculo: *Volver*, para que retorne a la página *Categorias_lista* (que aún no está hecha) para facilitar el retorno en el caso de que el usuario cambie de opinión y no desee añadir ninguna categoría
- Cambia el texto del botón *Insertar Registro* por *Añadir Categoría*

Observación del código:

- Para mejorar su aspecto puedes combinar las celdas del botón y cambiar la alineación centrada en el código del botón.

Observa que no es más que un botón de formulario tipo submit:

```
<td colspan="2" align="center" nowrap="nowrap"><input type="submit" value="Añadir categoría" /></td>
```

Sin embargo la acción del formulario es enviada a la función escrita en PHP *editFormAction*:

```
<form action="<?php echo $editFormAction; ?>"
```

Un poquito más de código.

Observa la función en PHP:

```
$editFormAction = $_SERVER['PHP_SELF'];  
if (isset($_SERVER['QUERY_STRING'])) { $editFormAction .= "?" .  
htmlentities($_SERVER['QUERY_STRING']);}  
if ((isset($_POST["MM_insert"])) && ($_POST["MM_insert"] == "form1")) {$insertSQL =  
sprintf("INSERT INTO tblcategoria (strDescripcion) VALUES (%s)",  
GetSQLValueString($_POST['strDescripcion'], "text"));
```

La función evalúa el campo oculto (*hiddend*) en el formulario *MM_insert* para insertar en la tabla *tblcategoria* el texto del campo de texto del formulario: *strDescripcion* y añadir este valor en el campo *strDescripcion* de la tabla *tblcategoria*.

Para comprobar su funcionamiento:

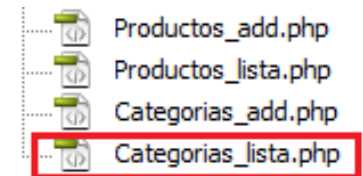
- Guarda y previsualiza la página [F12]
- Desde la vista previa de la web, añade una categoría:
Por ejemplo: *Apartamento* o *Deportivas carrera* (según si tu web es *Oficasa* o *Mike*)



Observa cómo después de pulsar en *Añadir Categoría*, debería ir a la página de *Lista de Categorías*, (que si no la tenemos aún hecha nos mostrará un error)

Crear la página lista de categorías

- Desde el panel de *archivos* crea una copia de *index.php*
- Cambia el nombre de la copia de index a *categorias_lista.php*.
- Abre la página y cambia el título por: *Listado de categorías*



Crear la consulta:

- En la ventana de *Base de datos*, pestaña **Vinculaciones** [Ctrl+F10], pulsa en **+** y crea un nuevo **juego de registros** (consulta)
- En la ventana, selecciona la conexión: *conexion_local* y la tabla: *tblCategorias*.
- Genera la siguiente consulta SQL:

SELECT * FROM tblcategoria ORDER BY tblcategoria.strDescripcion ASC

- Pulsa *Aceptar* para cerrar la ventana.
- Inserta una tabla en la página: Escoge (*Insertar* ▶ **Tabla**): 2 filas x 2 columnas.
- En la primera fila escribe: **Nombre de la categoría** y **Acciones**
- En la segunda fila:

Arrastra desde la pestaña de *Vinculaciones*, el campo *strDescripcion* hasta la celda.

En la siguiente celda escribe centrado: **Añadir – Modificar – Eliminar**

Para que se repitan las filas de datos:

- Selecciona toda la segunda fila de la tabla (pulsando en <tr> de la barra de estado)
- Escoge de *Comportamiento del servidor* **+**: **Repetir región**.
- Selecciona **⊙ Todos los registros**, pues no serán muchas las categorías a mostrar.
- Pula *Aceptar*.

Preparar los vínculos:

- Selecciona cada acción de la 2ª fila y asigna los siguientes vínculos:

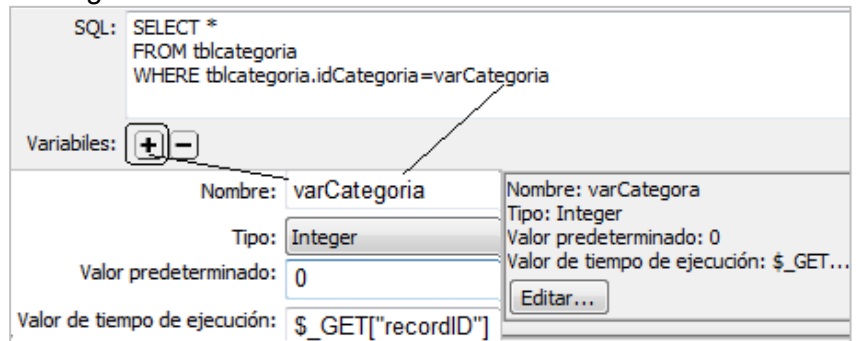
Añadir: Categorias_add.php
 Modificar: Categorias_edit.php
 Eliminar: Categorias_delete.php

- Guarda la página con el nombre: **Categorias_lista.php**
- Comprueba en la vista previa [F12] el vínculo a Añadir (el resto aún están en construcción)

Crear la página para editar categorías:

- Desde el *panel de Archivos* [F8], abre a la carpeta *Admin*, crea una copia de *index.php* (Ctrl+C - Ctrl+V).
- Cambia el nombre de la copia de index a ***categorias_edit.php***.
- Abre la página y cambia el título por: ***Modificar categoría***
- En la ventana de *Base de datos*, pestaña ***Vinculaciones*** [Ctrl+F10], pulsa en **+** y crea un nuevo ***juego de registros*** (consulta)
- En la ventana de *Juego de registros* activa la vista *Avanzada*
- Selecciona la tabla de *tblCategorias* y pulsa en ***SELECT***
- Pulsa sobre el campo *IdCategoria* y luego el botón ***WHERE***
- Completa la línea SQL como en la imagen

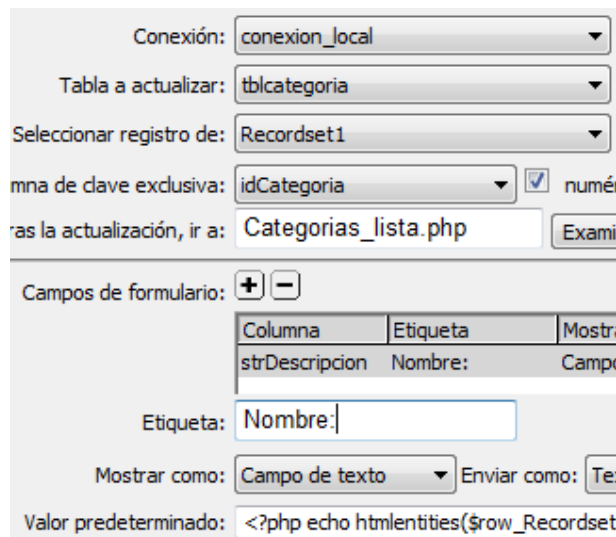
Pulsa en **[+]** para crear la variable ***varCategoria***
Será del tipo: *Integer* – número entero
Valor de tiempo de ejecución: ***\$_GET["recordID"]***



Esta consulta SQL indica que seleccione únicamente el registro que cumpla que su campo *idCategoria* coincida con la variable *varCategoria* que a su vez ha recogido el dato (GET) del parámetro ***recordID*** que le hemos mandado desde la página de lista de categorías.

Esta consulta también se puede hacer *utilizando el Filtro del modo Simple...*

Crear el formulario editar categorías:



- Escoge del menú: *Insertar* ▶ *Objeto de datos* ▶ *Actualizar registro* ▶ *Asistente de formulario de actualización de registros*
- Rellena los datos como en la imagen adjunta. Al pulsar *Aceptar* se mostrará un formulario con una tabla.
Nombre: {Recordset1.strDescripcion} Actualizar registro
- Cierra y guarda la página *categorias_edit.php*
- Comprueba desde la página *categorias_lista.php* que puedes abrir la página modificar la categoría al pulsar en el enlace *Modificar*.

Vincular los datos de la lista de categorías con editar categorías.

Para identificar qué registro se va a editar, es necesario indicar en el vínculo el número de registro:

- Recupera la página ***Categorias_lista.php*** si no la tienes cargada.
- Con el texto *Editar* seleccionado, divide la ventana para acceder al *Código*, y busca la sección:
<td>>Editar</td>
- Añade el texto: ***?recordID=*** luego arrastra desde el panel de vínculos el campo *idCategoria* y suéltalo, de modo que al final el código quede así:
<td><a href="categorias_edit.php?recordID=<?php echo \$row_Recordset1['idCategoria']; ?>">Editar</td>

Con esto vinculamos *Editar* con la página *categorias_edit.php* pero le adjuntamos también el parámetro con la variable *recordID* que contendrá el número de registro del campo *idCategoria*.

Nota: Para adjuntar un parámetro a un vínculo, se indica poniendo delante el signo ?

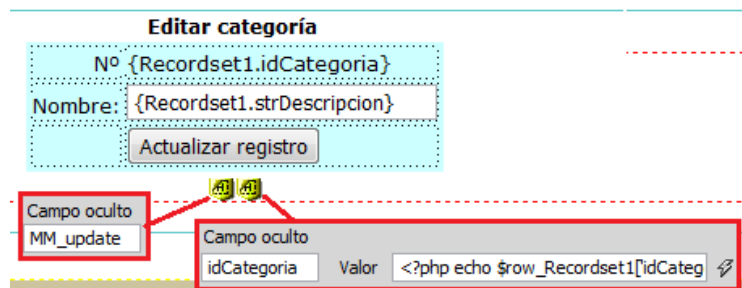
Ojo: el parámetro distingue mayúsculas y minúsculas.

- Guarda la página y comprueba en el navegador si se muestra la categoría y que puedas añadir más.

Añadir elementos al formulario de edición.

Añadir el campo N° de categoría

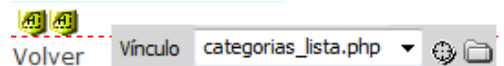
- Como comprobante de que el número de registro que estamos editando es el correcto, (pero no editable), hemos decidido añadir una fila a la tabla con el campo *idCategoria* en modo etiqueta – label
- Para ello sitúa el cursor en la primera fila de la tabla del formulario y escoge del menú: *Insertar* ▶ *Objetos de tabla* ▶ *Fila arriba*
- En esta nueva fila añade el texto: **N°** en la primera columna y arrastra hasta la segunda columna, el campo *idCategoria* de la ventana de *Vinculaciones*.
La tabla debe quedar como en la imagen.



Observa los campos ocultos *MM_update* para la actualización del registro y el *idCategoria* que recoge el n° de registro que tiene que modificar.

Añadir el vínculo volver

Añade bajo el formulario el vínculo: **Volver**, para que retorne a la página *Categorias_lista* para el caso de que el usuario cambie de opinión.



Comprobación de la relación entre la página lista de categorías y la edición de categorías:

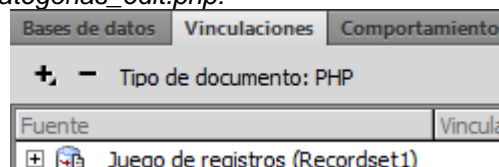
- Guarda ambas páginas: *Categorias_lista.php* y *categorias_edit.php*
- Previsualiza desde el navegador la página *Categorias_lista.php*.
- Sitúa el cursor sobre el vínculo de una categoría de la lista (*Apartamento* o *Deportivas carrera*)
- Observa si se muestra el N° de registro **recordID** en el enlace de tu navegador
Si no se muestra, no has creado bien el enlace en la página *Categorias_lista.php*
- Haz clic en el enlace y comprueba que el número de categoría que vas a modificar se corresponde con el que has hecho clic.
Si no se muestra, has recogido mal el valor en la variable *varCategoria* en el juego de registros.
En tal caso, pulsa en doble clic sobre el juego de registros Recordset1 de la ventana de vinculaciones.
Comprueba también el valor `$_GET["recordID"]` de la variable *varCategoria*. Debe ser exactamente el mismo nombre que pusiste en el enlace.

En la página *Categorias_lista.php*:

1	Vivienda	Añadir - Modificar - Elim
2	Apartamento	Añadir - Modificar - Elim

localhost/tienda/Admin/categorias_edit.php?recordID=1 e de página

En la página *categorias_edit.php*:



Crear la página para eliminar categorías

- Desde el panel de *Archivos* [F8], dentro de la carpeta *Admin*, crea una copia de *index.php* (ctrl+c - ctrl+v).
- Cambia el nombre de la copia a ***categorias_delete.php***.
- Abre la página y cambia el título por: ***Eliminar categoría***
- Escoge del menú: *Insertar* ▶ *Objeto de datos* ▶ ***Eliminar registro***
Rellena los datos como en la imagen adjunta

No hemos necesitado crear una consulta o juego de registros en esta página.

El asistente para eliminar registro se encarga de ello, e introduce el código:

```
"DELETE FROM tblcategoria WHERE idCategoria=%s"
```

Donde el parámetro que le pasamos **%s** es el parámetro URL: recordID.

si hay una variable definida: Valor de clave principal

Conexión: conexion_local

Tabla: tblcategoria

Columna de clave principal: idCategoria numéric

Valor de clave principal: Parámetro URL recordID

Después de borrar, ir a: Categorías_lista.php

- Vuelve a la página ***Categorías_lista.php*** de la carpeta Admin.
- Selecciona el texto: *Borrar* y modifica el vínculo a la página creada: *Categorias_delete.php*
Para especificar de qué elemento se trata, añade al vínculo el parámetro: **?recordID=** y a continuación arrastramos el campo del *idCategoria* desde el panel de vínculos de la base de datos. También puedes copiar el mismo identificador que hiciste para *Editar* productos, quedando así el vínculo:

```
<a href="Categorías_delete.php?recordID= <?php echo $row_Recordset1['idCategoria']; ?>">Borrar</a>
```

- Guarda ambas páginas y comprueba que desde la página de *Categorías_lista* que puedes borrar una categoría.

Recuerda que si al previsualizar desde el navegador la página *Categorías_lista.php*, no se muestra el N° de registro **recordID** en el enlace de tu navegador, no has creado bien el enlace en la página *Categorías_lista.php*

Si no se te actualiza la página *Categorías_lista*, después de volver de la página *Categorias_delete*, añade en la sección Head, antes del Title, la siguiente etiqueta html: `<meta http-equiv="Last-Modified" content="0">` que indica que la página se ha modificado y necesita ser recargada.

Sesión 5 – Mejoras en páginas de administración

Hasta aquí hemos procurado emplear campos de carácter general para poder ser funcionales a los objetivos de una amplia gama de páginas web.

Sin embargo, a partir de aquí, debemos añadir una serie de campos específicos a cada objetivo web.

Antes realizaremos un duplicado de la tabla *tblProductos*.

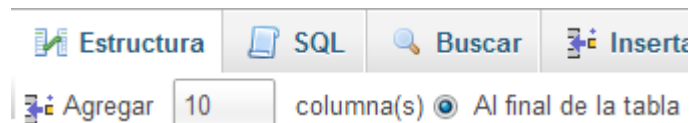
Copia de seguridad de la tabla original en la misma base de datos.

1. Con el servicio Apache iniciado, escribe en tu navegador web la URL: **localhost/phpmyadmin** para acceder directamente al gestor **PhpMyAdmin**.
2. Haz clic sobre la base de datos creada **tienda** y selecciona la tabla de productos. **tblProductos**
3. Haz clic sobre **Estructura**.
4. En la sección. Operaciones escoge **Copiar la tabla a**
Esta opción también te permite traspasar tablas de una base de datos a otra.



Añadir campos específicos a productos

- Con el servicio Apache iniciado, escribe en tu navegador web la URL: **localhost/phpmyadmin** para acceder directamente al gestor **PhpMyAdmin**.
- Haz clic sobre la base de datos creada **tienda** y sobre la tabla de productos. **tblProductos**
- Haz clic sobre **Estructura** y agrega 10 campos al final de la tabla



En el caso de una web inmobiliaria como *Oficasa* añadiremos los campos de la imagen izquierda. Para una web tipo comercio, como *Mike*, añadiremos los campos de la imagen derecha.

#	Nombre	Tipo	Nulo	Predeterminado	Extra
1	<u>idProducto</u>	int(11)	No	Ninguna	AUTO_INCREMENT
2	strRef	varchar(20)	No	Ninguna	
3	strTipo	varchar(25)	No	Ninguna	
4	strNombre	varchar(100)	Sí	NULL	
5	strPoblacion	varchar(100)	Sí	NULL	
6	strZona	varchar(50)	Sí	NULL	
7	dblPrecio	double	Sí	NULL	
8	intEstado	int(11)	Sí	NULL	
9	strCategoria	varchar(50)	No	Vivienda	
10	strImagen	varchar(50)	Sí	NULL	
11	strDescripcion	varchar(200)	Sí	NULL	
12	strProvincia	varchar(30)	Sí	Tarragona	
13	strSuperficie	int(10)	Sí	NULL	
14	strHabit	int(2)	Sí	NULL	
15	strPropietario	varchar(100)	Sí	NULL	
16	strContacto	varchar(100)	Sí	NULL	
17	strCondiciones	varchar(200)	Sí	NULL	

#	Nombre	Tipo	Nulo	Predeterminado	Ex
1	<u>idProducto</u>	int(11)	No	Ninguna	AUTO_INCREMENT
2	strRef	varchar(20)	No	Ninguna	
3	strTipo	varchar(25)	No	Ninguna	
4	strNombre	varchar(100)	Sí	NULL	
5	strDireccion	varchar(100)	Sí	NULL	
6	strColor	varchar(50)	Sí	NULL	
7	dblPrecio	double	Sí	NULL	
8	intEstado	int(11)	Sí	NULL	
9	strCategoria	varchar(50)	No	Vivienda	
10	strImagen	varchar(50)	Sí	NULL	
11	strDescripcion	varchar(200)	Sí	NULL	
12	intTalla	int(3)	Sí	NULL	
13	strUnidades	varchar(10)	Sí	NULL	
14	intCantidad	int(2)	Sí	NULL	
15	strProveedor	varchar(100)	Sí	NULL	
16	strContacto	varchar(100)	Sí	NULL	
17	strCondiciones	varchar(200)	Sí	NULL	

Cómo añadir los campos nuevos de la tabla en el formulario web

Vamos a añadir a la página el campo de referencia **strRef**, necesario para identificar el producto, sin asistente:

- En *Dreamweaver*, abre la página de añadir productos: **productos_add.php**.
- Pon el cursor en la primera fila de la tabla datos, que contiene el texto *Nombre*
- Escoge del menú: *Insertar - Objeto de tabla* ▶ **Insertar fila arriba**. Para añadir un nuevo renglón
- En la primera celda, escribe el texto: **Refer**.
- En la segunda celda, escoge del menú: *Insertar – Formulario* ▶ **Campo de texto** con el ID: *strRef*

Aunque hemos añadido el campo *strRef* al formulario, debemos también incluirlo en la consulta SQL:

- Abre el código y busca la etiqueta SQL: INSERT, en ella añade el campo **strRef** del tipo texto y el parámetro %s y añade a los valores recogidos por el formulario el mostrado en negrita:

```
$insertSQL = sprintf("INSERT INTO tblproducto (strRef, strNombre, dbIPrecio, intEstado, strCategoria, strImagen, strDescripcion) VALUES (%s, %s, %s, %s, %s, %s, %s)",  
GetSQLValueString($_POST['strRef'], "text"), GetSQLValueString($_POST['strNombre'], "text"),  
GetSQLValueString($_POST['dbIPrecio'], "double"), GetSQLValueString($_POST['intEstado'], "int"),  
GetSQLValueString($_POST['strCategoria'], "text"), GetSQLValueString($_POST['strImagen'], "text"),  
GetSQLValueString($_POST['strDescripcion'], "text"));
```

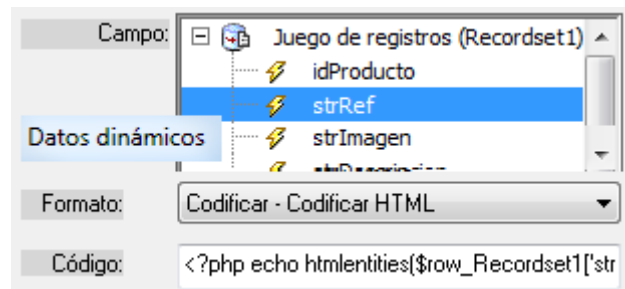
En PHP, las variables siempre empiezan con '\$'

Fíjate cómo la variable que contiene la cadena SQL anterior, **\$insertSQL** la utilizará en la consulta: `mysql_query($insertSQL, $conexion_local)` o sino hará un `die(mysql_error());`

*En PHP, la función: **die**(Mensaje) permite "cortar" la ejecución del programa PHP -mostrando el mensaje de error.*

Vamos a añadir el campo **strRef** al formulario de editar productos: *productos_edit.php*

- En *Dreamweaver*, abre la página de editar o modificar productos: **productos_edit.php**.
- Insertar una fila nueva y añade un campo de texto igual como has hecho en la página *productos_add.php*.
- En la ventana de vinculaciones, pulsa doble clic sobre el Juego de registros (Recordset1) y pulsa *Aceptar* para actualizar los campos. Se mostrará el nuevo campo de la tabla: *strRef*
- Selecciona el nuevo campo del formulario y en su propiedad **Valor inicial** pulsa en *Vincular con fuente dinámica*.
- Escoge el campo y formato de la imagen y pulsa *Aceptar*
- Busca en el código la cadena sql: `UPDATE` y añade los valores en negrita:
`$updateSQL = sprintf("UPDATE tblproducto SET strRef=%s, strNombre=%s, dbIPrecio=%s, intEstado=%s, strCategoria=%s, strImagen=%s, strDescripcion=%s WHERE idProducto=%s",
GetSQLValueString($_POST['strRef'], "text"), GetSQLValueString($_POST['strNombre'], "text"), ...`
- Guarda la página. Comprueba la página accediendo desde *productos_lista.php*



Añadir el resto de los campos.

Para añadir el resto de los campos en ambos formularios, puedes tomar la decisión de repetir estos pasos anteriores o generar de nuevo el asistente de formularios. Ambos modos están ya explicados.

Gestión para el campo de imagen:

El campo llamado **strImagen** guardará la ruta de la imagen que después subiremos.

- En *Dreamweaver*, abre la página de *añadir productos: productos_add.php*.
- Sitúa el cursor tras el campo de imagen
- Escoge del menú: Insertar – Formulario ▶ **Botón** [Aceptar]
- En sus propiedades cambia el Valor por *Subir Imagen* y la Acción a *Ninguno*
- En su código añade la llamada al script *subirimagen*, que está en negrita:

Refer:	<input type="text" value="{Recordset1.strRef}"/>
Nombre:	<input type="text" value="{Recordset1.strNombre}"/>
Precio:	<input type="text" value="{Recordset1.dblPrecio}"/>
Estado:	Normal ▼
Categoría:	▼
Imagen:	<input type="text" value="{Recordset1.strImagen}"/> <input type="button" value="Subir imagen"/>
Descripción:	<input type="text" value="{Recordset1.strDescripcion}"/>
<input type="button" value="Actualizar registro"/>	

```
<input type="button" name="button" id="button" value="Subir imagen"
```

```
onclick="javascript:subirimagen();" /></td>
```

- En la región editable del *Body*, al principio, añadimos la función Script del recuadro.
- Esta función nos abrirá una ventana llamada *remote* de medidas 400 x 150, sin barras ni menú, y cargará en esta la página *gestionimagen.html*

```
<script> //dentro del Body
function subirimagen()
{
self.name = 'opener'; // nos llamamos opener
remote = open('gestionimagen.php', 'remote', //abrimos ventana
'width=400, height=150, location=no, scrollbars=yes, menubars=no, toolbars=no, resizable=yes, fullscreen=no, status=yes');
remote.focus(); //ventana remote pasa a primer plano
}
</script>
```

Crear el formulario para subir la imagen:

- Crea una página nueva PHP en blanco.
- Añade a la página el código que se muestra, en el recuadro.
- Guarda la página en la carpeta, con el nombre: **gestionimagen.php**.

Guarda ambos archivos y comprueba desde navegador que se muestra una ventana al pulsar el botón: *Subir imagen*.

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Subir imagen</title>
</head>
<body>
<?php if ((isset($_POST["enviado"])) && ($_POST["enviado"] == "form1")) {
    $nombre_archivo = $_FILES['userfile']['name'];
    move_uploaded_file($_FILES['userfile']['tmp_name'], ".. /Productos/".$nombre_archivo);

    ?>
    <script>
        opener.document.form1.strImagen.value="<?php echo $nombre_archivo; ?>";
        self.close();
    </script>
    <?php
}
else
{?>
<form action="gestionimagen.php" method="post" enctype="multipart/form-data" id="form1">
    <p>
        <input name="userfile" type="file" />
    </p>
    <p>
        <input type="submit" name="button" id="button" value="Subir Imagen" />
    </p>
    <input type="hidden" name="enviado" value="form1" />
</form>
<?php }?>
</body>
```

Explicación breve: En esta ventana se incluye un formulario del tipo multipart-form-data para poder enviar datos en binario.

El formulario contiene un solo campo llamado Userfile del tipo file.

Al pulsar el botón del formulario, es enviado a la misma ventana y isset, comprobará si, al enviarse, ha sido enviado por el botón de subir imagen y entonces la variable: `$nombre_archivo` almacenará la ruta del archivo escogido, lo copia al servidor, dentro de la carpeta **Productos**. Luego, un script guarda el valor del nombre del archivo en el campo `strImagen` del formulario de la ventana llamada `openar` y luego autocierra la ventana (self close).

Para ello hemos añadido un campo oculto (hidden) en el formulario llamado `enviado` y que contiene el nombre del formulario `form1`.

El cuadro de texto de archivo de imagen con el botón de *Examinar* puede insertarse desde el menú de DW5.5: *Insertar* ▶ *Formulario* ▶ **Campo de Archivo** o en DW5: Campo de texto - tipo archivo.

Como el archivo de imagen lo mandamos guardar en el servidor dentro de la carpeta *Productos*, será necesario tener esta carpeta en el sitio.

- Guarda y comprueba desde navegador que la ventana carga el nombre de la imagen en el campo: `StrImagen`

Añadir el botón subir imagen en la página Productos:edit.

- Del mismo modo que en la página de *añadir productos*, abre la página: `productos_edit.php`.

- Sitúa el cursor tras el campo de imagen

- Escoge del menú: *Insertar* – *Formulario* ▶ **Botón** [Aceptar]

- En sus propiedades cambia el Valor por `Subir Imagen` y la Acción a `Ninguno`

- En su código añade la llamada al script `subirimagen`, que está en negrita:

```
<input type="button" name="button" id="button" value="Subir imagen"
```

```
onclick="javascript:subirimagen();" /></td>
```

- En la región editable, en el inicio del `body`, añade la misma función script `subirimagen`.

Mejora en la plantilla. Estilos CSS

Añadir una imagen o logo:

- Como prueba, puedes dibujar un sencillo logo en *Paint* y guardarlo en:

`..xampp\htdocs\tienda\imagenes` con el nombre: **Logo.png** o `logo.jpg`.

- Vuelve a Dreamweaver y abre la plantilla de la carpeta *Templates*: **Admin.dwt.php**

- *Selecciona el marcador de posición de imagen* que existe en la esquina superior para insertar el logo y cambia el origen por el logo que has guardado antes

- Si no tienes marcador de posición de imagen, puedes escoger del menú: *Insertar* ▶ *Imagen*.

- Escoge del menú: *Archivo* ▶ **Guardar**.

Al guardar la plantilla se mostrará una ventana para actualizar todas las páginas que dependen de esta plantilla.

Pulsa en **Actualizar** los ficheros dependientes.

Campo requerido. Obligar a introducir un valor.

- Abre la página para añadir productos:

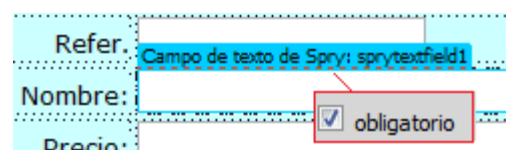
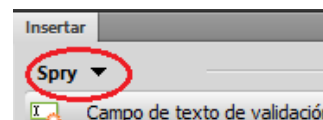
`Productos_add.php`.

- Selecciona el campo del *Nombre* en el formulario.

- En la paleta de *Spry* pulsa en el botón: **Campo de texto de validación de Spry**.

- Se superpondrá al recuadro del nombre este recuadro de spry.

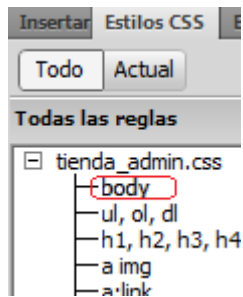
- Marca la casilla *Obligatorio*.



- Guarda la página. Es posible que solicite actualizar o añadir los objetos de Spry al sitio. Se creará una carpeta en el sitio **SpryAssets** donde se guardan los scripts .js

Ajustar la fuente en la hoja de estilos:

body {	footer {
font-	font-
family:	color:#FFF;
Tahoma,	padding:
Geneva,	10px 0;
sans-	background:
serif;	#369;
font-	text-align:
size:	center;
12px;	font-size:
color:	80%;. }
#333; }	



- Abre el archivo de estilos de la carpeta *Estilos* del sitio: **tienda_admin.css**
- En el apartado **Body** borra la fuente
- En la ventana – panel lateral de estilos CSS [May + F11], activa el botón **Todo** y haz **doble clic** en **Body** para abrir la ventana de definición de regla.
- Dentro de la ventana, selecciona la fuente Tahoma – tamaño 12 color gris oscuro.
- En el estilo **footer**, introduce los valores del recuadro y en la plantilla, el texto inferior:

[Página principal](#) | [Aviso legal](#) | [Protección datos](#) | [Contacto](#)

El logo de *Facebook* y *Tweter* los puedes guardar en la carpeta *Images*.

- Guarda la página de estilos y la plantilla observa el cambio en la visualización de todas las páginas.

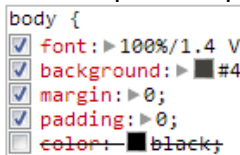
Resaltar las celdas en la lista de productos:

- En la hoja de estilos **tienda_admin.css** añade al final de la página una nueva clase llamada **resaltar** que se activa al pasar el ratón por encima (hover):

```
.resaltar:hover{background-color:#FFFF99}
```
- Guarda la hoja de estilos y abre la página **Productos_lista**
- En la fila de los datos de los productos (2ª fila), añadimos: `<tr class="resaltar">`

Utilizar herramientas de desarrollo de los navegadores:

Para reconocer la posición de un código en la página podemos utilizar como herramienta de ayuda el visor que incorporan los navegadores:



- Abre la página con el navegador (Explorer, Chrome o Firefox) y pulsa la tecla [F12]
- Selecciona un objeto en la página y observa que puedes desactivar atributos:
- En *Header* cambia : background-color: #FFFFFF;

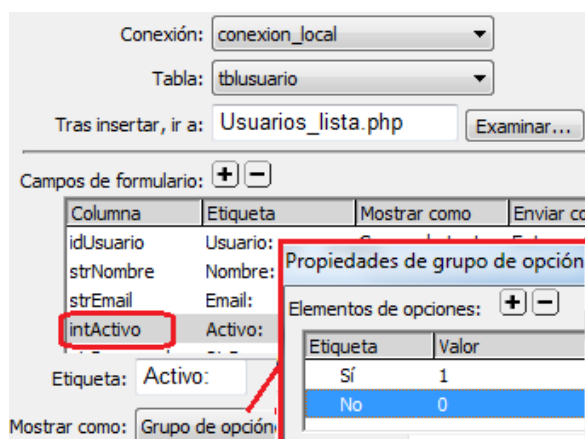
Sesión 6: Páginas de administración de usuarios

En esta sección crearemos las páginas de gestión de usuarios.


Ten en cuenta que a estas páginas estarán restringidas y sólo podrá acceder un usuario con privilegios de administrador.

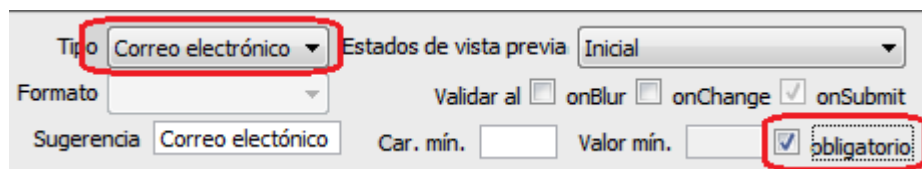
Página para añadir usuarios

- Desde el *panel de Archivos* [F8], en la carpeta *Admin*, crea una copia de *index.php* (Ctrl+C - Ctrl+V)
- Cambia el nombre de la copia de index a: **usuarios_add.php**
- Abre esta nueva página y cambia el título por: *Crear usuario*
- Escoge del menú:
Insertar ▶ *Objeto de datos* ▶ *Insertar registro* ▶ **Asistente de formulario de inserción de registros**
- Rellena los datos como en la imagen adjunta
 - Quita el campo *IdUsuario*
 - Corrige las etiquetas para los campos.
 - *Tras insertar, ir a...* mostraremos a la página *usuarios_lista* (que aún no está hecha)
 - En el campo **IntActivo** escoge *Mostrar como Grupo de opción*
Elementos: Sí – No Valores: 1 y 0
 - En el campo **IntNivel** escoge *Mostrar como Menú*
Elementos: Usuario: 1 – Administrador: 0
- Pulsa *Aceptar* para insertar el formulario.
- Cambia el nombre del botón por: *Añadir usuario*.
- Cambia la propiedad del *Estado inicial* del botón de opción Sí por *Activado*



Validación de campos con Spry

- Desde la barra de *Insertar Spry* puedes convertir los campos nombre y email en obligatorios. Pulsando en:  *Campo de texto de validación* y activar el estado de vista previa: **Obligatorio**.
- El de *Mail* también de tipo correo electrónico.
- Aunque el *Password*, podría tener la propiedad tipo Contraseña no es aconsejable ya que accederá un usuario administrador.



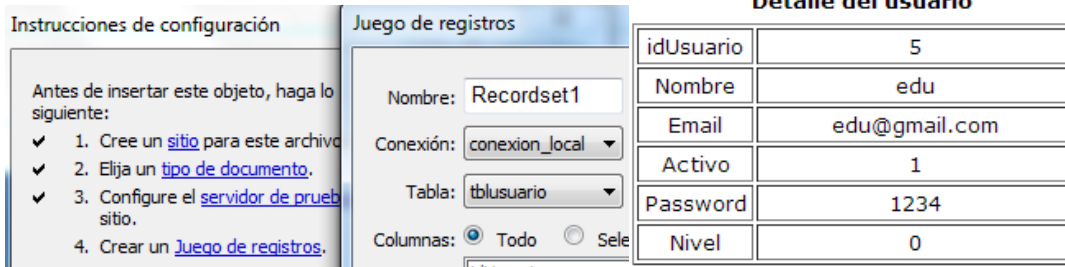
- Guarda y comprueba la página añadiendo un par de usuarios. Uno de cada nivel. No hemos necesitados crear ninguna consulta o juego de registros para esta página ya que la conexión con la base de datos ya la proporciona el asistente:
**INSERT INTO tblusuario (idUsuario, strNombre, strEmail, intActivo, strPassword, intNivel)
VALUES (%s, %s, %s, %s, %s, %s)",**

Páginas formulario de lista y detalle

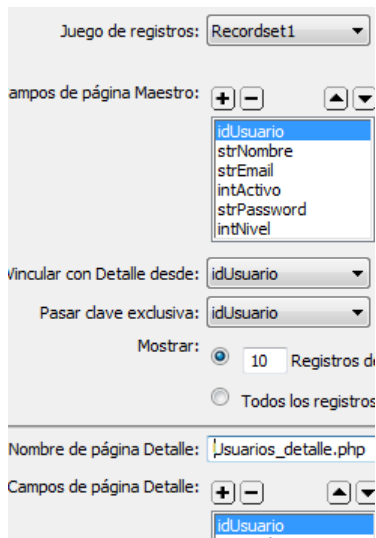
Como vamos a crear un formulario maestro-detalle, para ello, prepararemos dos páginas.

- Desde el *panel de archivos* crea una copia de *index.php* (ctrl+c - ctrl+v)
- Cambia el nombre a **usuarios_lista.php**

- Desde el panel de *archivos* otra una copia de *index.php* (ctrl+c - ctrl+v)
- Cambia el nombre a *usuarios_detalle.php*
- Abre la página *usuarios_lista.php* y cambia el título por: **Lista de usuarios**



Para crear el formulario maestro-detalle:



- Escoge del menú de DW: *Insertar – Objetos de datos ▶ Juego de páginas maestro-detalle.*

El asistente nos pedirá un juego de registros (tener una consulta). Pulsa en crear juego de registros.

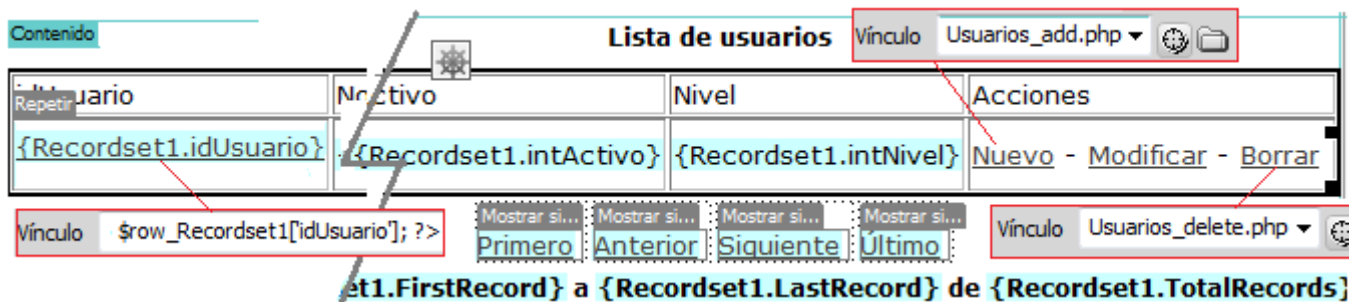
```
SELECT * FROM tblusuario ORDER BY tblusuario.strNombre
```

Al pulsar Aceptar, DW nos creará o actualizará la página subordinada *usuarios_detalle.php* que dependerá de la página *usuarios_lista.php*

Guarda ambas página y comprueba el vínculo al pulsar en el ID del usuario.

Añadir los vínculos

- Vuelve a abrir la página *usuarios_lista.php*.
- Por cuestiones de seguridad, borra los datos de la columna del password.
- Desplaza los datos de la columna del estado para que ocupe la columna libre de la izquierda, dejando una columna libre al final de la tabla. Como en la imagen.



En esa última columna, escribe los vínculos:

Nuevo → *Usuarios_add.php*

Modificar → *Usuarios_edit.php*

Borrar → *Usuarios_delete.php*

- Copia el enlace o vínculo que se muestra en el ID de usuario y se lo pegas al texto Modificar y Borrar. Luego, cambia los vínculos a sus páginas respectivas:

```
usuarios_edit.php?recordID=<?php echo $row_Recordset1['idUsuario']; ?>
```

```
usuarios_delete.php?recordID=<?php echo $row_Recordset1['idUsuario']; ?>
```

- Guarda de nuevo la página de *usuarios_lista.php*.

Editar y eliminar usuarios.

Básicamente equivale a repetir las mismas acciones que para editar y eliminar categorías o productos, por eso, emplearemos una variante, a título de aprendizaje.

Para editar usuarios:

Aprovecharemos los datos de la página de *usuarios_detalle.php*

Si abres esta página ***usuarios_detalle.php***, puedes ver el juego de registros que generó el asistente (**DetailRS1**) donde ya está pasado el parámetro de nº de registro a editar (**recordID**).

- Selecciona la tabla de datos y elimínala y en su lugar escoge del menú:

Insertar ▶ *Objeto de datos* ▶ *Actualizar registro* ▶ **Asistente de formulario de actualización de registros**

- Rellena los datos como en la imagen adjunta.
- En campo *Activo* puede ser un *Menú* con los valores de lista: 1 (Sí) y 0 (No).

Guarda la página (Archivo – Guardar como...) con el nombre: ***usuarios_edit.php***

Para borrar usuarios:

Aprovecharemos la página de usuarios datos donde ya está pasado el parámetro de nº de registro a editar (**DetailRS1**)

- Abre la página: ***usuarios_detalle.php***.
- Selecciona la tabla de datos y elimínala y en su lugar escribe el texto: “**Usuario eliminado**”
- Escoge del menú: *Insertar* ▶ *Objeto de datos* ▶ **Eliminar registro**
- Guarda la página con el nombre: ***usuarios_delete.php***

Comprueba la gestión de usuarios completa.

Comprobación de email repetido

En la página de alta de usuarios vamos a comprobar que el E-mail no esté repetido. Antes crearemos una página de error:

- Desde el *panel de Archivos* [F8], en la carpeta *Admin*, crea una copia de ***index.php*** (Ctrl+C - Ctrl+V)
- Cambia el nombre de la copia de index a: ***usuario_repetido.php***
- Abre esta nueva página y cambia el título por: “**El e-mail ya existe.**” “Comprueba que ya está dado de alta en el sistema o recupera la contraseña.”
- Guarda la página ***usuario_repetido.php***
- Abre la página de alta: ***usuarios_add.php***
- En el panel de base de datos, apartado *Comportamientos*, Autenticación de usuario, escoge: **Comprobar nuevo nombre de usuario.**

Al pulsar aceptar, DW genera un código de comprobación

En la sección: // *** Redirect if username exists
\$MM_dupKeyRedirect=" *usuario_repetido.php* ";

- Comprueba su funcionamiento volviendo a dar de alta de nuevo con el mismo email.
- Guarda la página ***usuarios_add.php***.

Autenticación de acceso usuario (login)

Creación de la página de acceso de usuario que le identificará como administrador para poder acceder a las páginas de administración:

- Desde el *panel de Archivos* [F8], en la carpeta *Admin*, crea una copia de *index.php*
- Cambia el nombre de la copia de index a: **usuario_acceso.php**
- Abre esta nueva página y cambia el título por: "**Formulario de acceso - Rellene el formulario con su ID y contraseña**"
 - Sitúa el cursor en el cuerpo y elige del menú: Insertar – Formularios – **Formulario**.
 - Dentro de la sección punteada roja del formulario, añade 2 campos de texto como en la imagen (Menú: *Insertar - Formulario - Campo de texto*).
Campo Email: ID: E_mail - Etiqueta: E_mail
Campo Contraseña: ID: Contra – Etiqueta: Contraseña.

- En las propiedades del campo *Contra*, activa el tipo:
 - ⊙ *Contraseña*
- Para poner los 2 botones: Elige del menú: Insertar – Formulario – **Botón**
 - Para el botón *Aceptar* asigna la Acción: ⊙ *Enviar formulario*
 - Para el botón *Cancelar* o *Restablecer* asigna la Acción: ⊙ *Reestablecer formulario*
- Con el cursor dentro del formulario escoge del menú: *Insertar – Objeto de datos – Autenticación de usuarios* ▶ **Conectar usuario**.

También lo puedes hacer desde el panel de **Comportamientos del servidor** (*Server Behaviors*) [Ctrl+F9], pulsa sobre **+** y escoge: *Autenticación de usuarios* ▶ **Conectar usuario**

Rellena el formulario como el de la imagen.
Pulsa *Aceptar* al finalizar.

El campo *intNivel* determinará el tipo de acceso para administradores (nivel 1) o para usuarios normales (nivel 0).

Explicación de la acción:

Al pulsar el botón **Enviar** se valida el formulario (*submit*) *Form1*.

Es el *Form1* el que se encarga de llamar a la acción **loginFormAction** desde el código:

```
<form id="form1" name="form1" method="POST" action="<?php echo $loginFormAction; ?>">
```

Esta acción está escrita al principio de la página en la sección:

```
$loginFormAction = $_SERVER['PHP_SELF'];
```

Y luego DW genera las siguiente variables de sesión:

- \$loginUsername=\$_POST['E_mail']; -> El ID de usuario será el campo 'E_mail'
- \$password=\$_POST['Contra']; -> el password será el campo 'Contra'
- \$MM_fldUserAuthorization = ""intNivel "; -> Nivel de autorización
- \$MM_redirectLoginSuccess -> Vínculo en caso correcto
- \$MM_redirectLoginFailed -> Vínculo en caso de error
- \$MM_UserGroup -> tipo de usuario

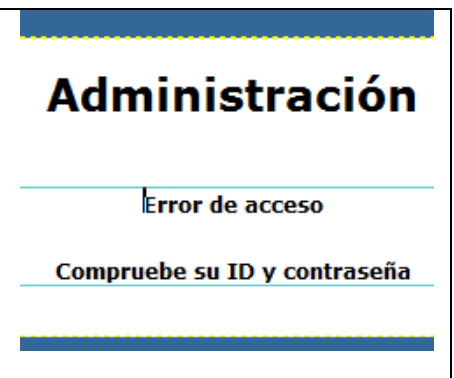
Estas variables de sesión estarán activas para todas páginas del sitio mientras dure la sesión.

- Guarda la página [Ctrl+S]

Crea la página de error de acceso:

- Desde el *panel de Archivos* [F8], abre la carpeta *Admin*,
- Crea una copia de ***index.php*** (Ctrl+C - Ctrl+V)
- Cambia el nombre de la copia de index a:
usuario_acceso_error.php
- Abre esta nueva página y cambia el título por:

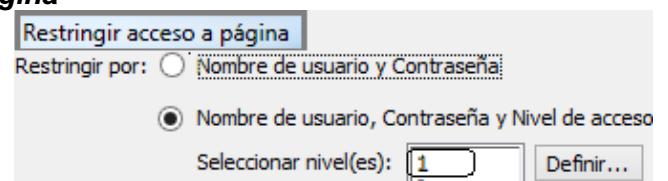
“Error de acceso
Compruebe su ID y contraseña”
Volver → *usuario_acceso.php*



- Guarda la página [Ctrl+S]

Modificación de las páginas de administración para restringir acceso a usuarios

- Vuelve a abrir la página ***usuarios_lista.php***.
- Desde el panel de ***Comportamientos del servidor*** [Ctrl+F9], pulsa sobre **+** y escoge: Autenticación de usuarios – ***Restringir acceso a página***
 - Introduce los valores de la imagen.
 - Guarda la página [Ctrl+S]
 - Comprueba su funcionamiento.



En caso de error intenta cambiar de las páginas la primera línea de carga virtual:

```
<?php virtual('/tienda/connections/conexion_local.php'); ?>
```

```
Por: <?php require_once('../connections/conexion_local.php'); ?>
```

Deberás poner, o no, los .. si llamas desde dentro de la carpeta *Admin* o desde la raíz del sitio)

Repite el mismo proceso para añadir el comportamiento de restringir página a todas las páginas de la Administración: *index*, categorías y productos.

El nivel 1 está definido en la página de acceso a los usuarios de administración por el campo de la tabla de usuarios: *intNivel*.

Esto permitirá acceder a las páginas de administración únicamente a los usuarios registrados en la tabla que tengan el nivel 1 (administradores) y no a los que estén registrados con el nivel 0 (usuario normal). Estos podrán acceder a las páginas de consulta y pedidos de productos, pero no a su modificación.

Identificación de usuario conectado

Desde el interior de la página, podemos mostrar el usuario que está “loqueado” simplemente aprovechando las variables que ha generado DW.

- Accede a la página **usuarios_lista.php** pulsa en dividir para ver el código:
Si observas el principio del código, DW crea, entre otras, las variables:
 - \$MM_authorizedUsers = " " -> de usuario autorizado en blanco
 - \$MM_donotCheckaccess = true -> es una variable lógica del tipo “flag” permiso_denegado, por defecto activado (true)
 - \$_SESSION['MM_Username'] -> variable que recoge la identificación del usuario del formulario
 - \$_SESSION['MM_UserGroup'] -> variable que recoge tipo de usuario

Inicia sesión de usuario (*session_start*) si la variable \$MM_authorizedUsers no está vacía y \$MM_donotCheckaccess está en false

En ese momento, la variable \$_SESSION['MM_Username'] contiene el email, en nuestro caso. (o nombre de usuario en otro caso)

- Sitúa, en diseño, el cursor bajo el texto: *Lista de usuarios*.
- En el código escribe el siguiente trozo de PHP: `<?php echo "Admnistrado por $_SESSION['MM_Username']; ?>`
(*echo* repite la cadena de texto que le sigue en el dispositivo preseleccionado. Es decir, en nuestro caso, muestra el mensaje en la pantalla)
- Guarda la página [Ctrl+S] y comprueba su funcionamiento.

Si la página no estuviese restringida, como es ahora el caso, se debería antes de comprobar si alguien ha iniciado sesión.

Por lo que el código anterior se podría cambiar por:

```
<?php
if (isset($_SESSION['MM_Username']) && ($_SESSION['MM_Username'] != ""))
// --> significa si está definido (is set) y (&&) el nombre de usuario no está en blanco... (! = " ")
{
echo "Hola. Has iniciado sesión como " . $_SESSION['MM_Username'];
// --> echo mostrará en pantalla el nombre o email
}
Else // --> en caso contrario...
{
echo "Usuario anónimo ";
}
?>
```

- Si pruebas este nuevo código, apenas notarás la diferencia. Puedes añadirlo en las página que quieras mostrar el identificador del usuario.

Recuerda añadir el comportamiento de restringir página a usuarios autorizados en todas las páginas de la Administración.

Al terminar, guarda todas las páginas y cierra la sesión.

Opcional: Cerrar sesión

Para cerrar la sesión del usuario, en la página de *usuarios_lista.php* puedes crear un vínculo a la página: **usuario_cerrar.php**

En la página: *usuario_cerrar.php* añadiremos con el asistente de DW:

Insertar – Objeto de datos – Autenticación de usuarios ▶ **Desconectar usuario**

Al *terminar* la sesión debería cargar la página pública principal que aún no tenemos hecha

Esta acción vacía el contenido de las variables: \$_SESSION['MM_Username'] = NULL;