

MySQL® se basa en el lenguaje de consultas de bases de datos SQL (Structured Query Language) y es un database Server de código abierto de Sun Microsystems.

Instalación en Windows:

Descargar: [mysql-essential-5.0.9-beta-win32.msi](#), usar el instalador automático: escribir un password fácil

En el menú de Windows, aparecerá: **MySQL ▶ MySQL Comad line**

Escribe el password anterior

Se mostrará: Welcome to the MySQL monitor. Y la línea de comandos: **mysql>**

Después de haberse conectado, puede desconectarse en cualquier momento escribiendo **QUIT** (o \q)

Opción 2: Instalar junto con un paquete tipo Xampp o Wampp

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.

- 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 remoto host.

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.

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...

Manejo desde la consola

Para conectar al servidor local

Ejecutar el símbolo del sistema o CMD

Cambiar de dirección a: CD C:\Program Files\MySQL\MySQL Server 5.0\bin\

En la línea de comandos, escribir:

```
shell> mysql --help
```

En la línea de comandos, escribir el nombre de usuario y password.

Ejemplo:

```
shell> mysql -h host -u user -p
```

```
Enter password: *****
```

Si haces las pruebas en la misma máquina te puedes saltar el host y simplemente escribir el nombre del usuario: **shell> mysql -u user -p**

Ayuda en línea en español: <http://dev.mysql.com/doc/refman/5.0/es/creating-tables.html>

Para crear tablas y base de datos

- **Crear una base de datos**: CREATE DATABASE menagerie; (Importante terminar con “;”)
Al crear una base de datos, ésta no se selecciona para su uso, debes especificar: USE menagerie
O indicar la base de datos en la línea de comandos: **shell> mysql -h host -u user -p menagerie**
- **Crear una tabla**: CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20),

-> species VARCHAR(20), sex CHAR(1), birth DATE, death DATE);

“-> “ Indica que continua en la siguiente línea

La creación de la base de datos ha sido una tarea sencilla, pero hasta ahora permanece vacía, como se muestra en: SHOW TABLES:

El tipo de dato VARCHAR almacena caracteres de longitud no uniforme;

Para verificar que la tabla ha sido creada en la forma esperada, utilice la sentencia DESCRIBE

```
mysql> DESCRIBE pet;
```

| Field | Type | Null | Key | Default | Extra |
|---------|-------------|------|-----|---------|-------|
| name | varchar(20) | YES | | NULL | |
| owner | varchar(20) | YES | | NULL | |
| species | varchar(20) | YES | | NULL | |
| sex | char(1) | YES | | NULL | |
| birth | date | YES | | NULL | |
| death | date | YES | | NULL | |

Ejercicio:

```
mysql> create database telefonos;
```

Query OK, 1 row affected (0.47 sec)

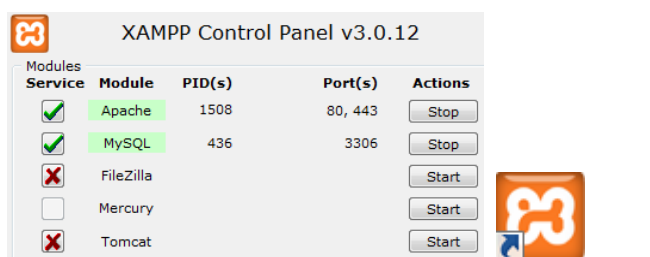
```
mysql> create table telefonos(nombre VARCHAR(20), telefono VARCHAR(20));
```

Query OK, 0 rows affected (0.54 sec)

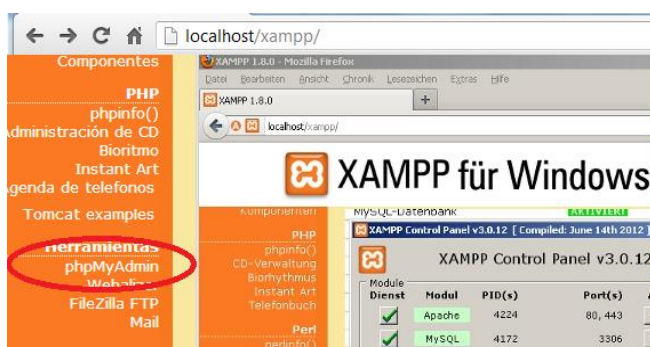
```
mysql> show tables;
```

```
+-----+
| Tables_in_telefonos |
+-----+
| telefonos           |
+-----+
1 row in set (0.44 sec)
```

Gestión de la base de datos desde un IDE (Interface de usuario)

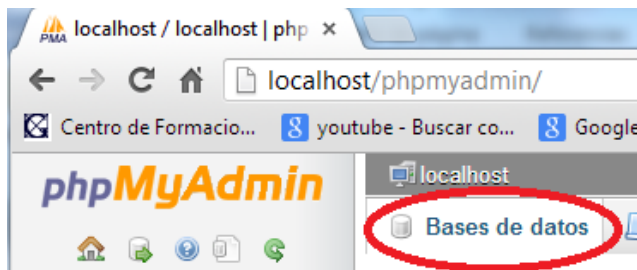


1. Una vez instalado **Xampp** o **Wampp**, ejecuta el acceso directo al panel de control para comprobar que tienes activado los servicios **Apache** y **MySQL**. En caso contrario actívalos pulsando el botón Start.



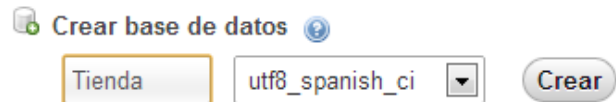
2. Ejecuta tu navegador web (Chrome, Explorer, Mozilla...) y escribe en el cuadro de la dirección: localhost
Accederás a la página de presentación de **Xampp**. **Localhost** equivale a usar nuestra propia máquina local como servidor de nuestra página web. Es decir, emula la dirección de nuestra web 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

- 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.



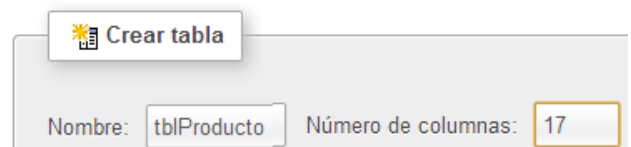
Hemos optado por el nombre Tienda porque es un nombre bastante genérico que nos permitirá adaptar este ejemplo fácilmente a comercios de diversos sectores.

2º Crear las tablas.

Crear la tabla Productos.

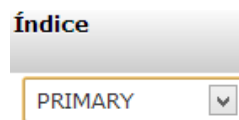
- Haz clic sobre la base de datos creada Tienda y en el recuadro para Crear tabla escribe:
tblProducto Nº de columnas 7.
- Pulsa en **Continuar**


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 | <u>idProducto</u> | int(11) | No | Ninguna | AUTO_INCREMENT |
| 2 | strNombre | varchar(100) | Sí | NULL | |
| 3 | dbIPrecio | 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 | |



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.

- 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

Una vez rellenados todos los campos,

pulsa el botón: **Guardar**

| # | 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 | |

El lenguaje SQL

Utilizado con frecuencia 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`

Integración del lenguaje SQL en PHP con Dreamweaver:

Conectar una página web con la base de datos:

- Ten activado el servicio apache y MySQL de Xamp o Wamp.

- Activa la *Ventana ▶ Base de datos* de *DreamWeaver*

En la ventana de base de datos, activa la pestaña: **Vinculaciones (Bindings)** [Ctrl+F10]

- ✓ 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: ▼

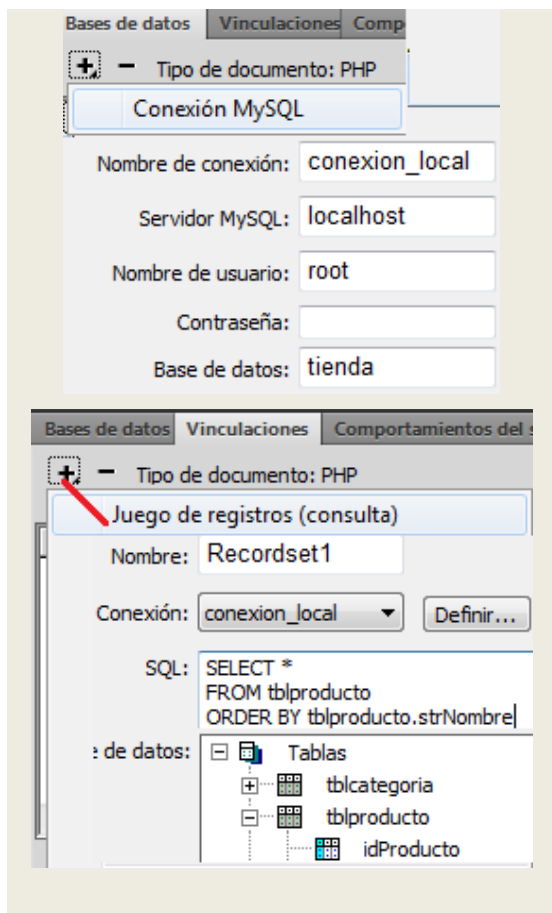
Carpeta del servidor: ✓

URL Web:

| XAMPP Control Panel v3.2.1 | | | | |
|----------------------------|--------|--------------|---------|---------|
| Modules | | | | |
| Service | Module | PID(s) | Port(s) | Actions |
| <input type="checkbox"/> | Apache | 6092 2272 | 80, 443 | Start |
| <input type="checkbox"/> | MySQL | 4968 | 3306 | Start |

- Comprueba que en los tres primeros elementos aparece el símbolo: ✓
- Pulsa en el segundo elemento: **Tipo de documento:** Escoge PHP
- Pulsa 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` (debe incluir el protocolo http://)

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**,
- Pulsando en **SELECT** para crear crea la consulta en SQL. Lo que nos escribirá:

*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*.

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

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

Dentro de las etiquetas: `<?php ?>` indicadoras 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”*

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_I)

Para ello puedes acceder a la estructura de cada tabla.

Selecciona el campo y pulsa en:  **Primaria**

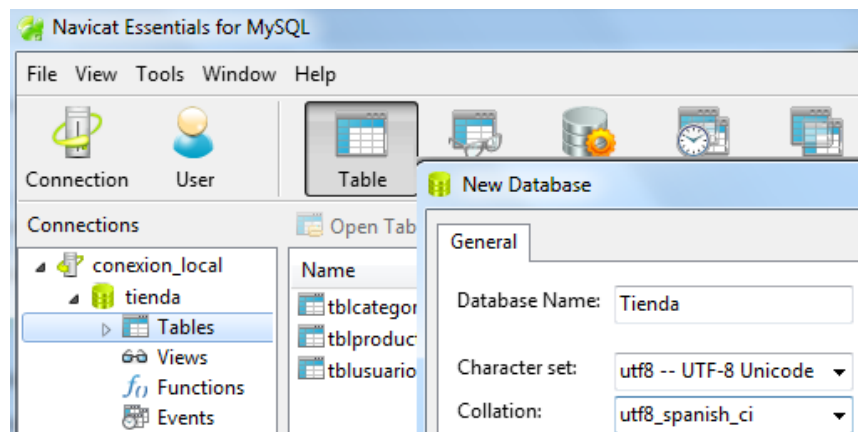


- Las tablas tendrán el tipo de almacenamiento *InnoDB*, esto nos permitirá más adelante crear claves foráneas.

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:

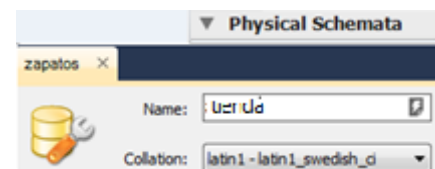
- Sobre la conexión creada, pulsa botón derecho del mouse
- Escoge **New database** para crea una nueva base de datos.
- En la ventana, escribe el nombre de la base de datos: **Tienda**
- Sobre la base de datos *Tienda*, pulsa en: **New table**

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.



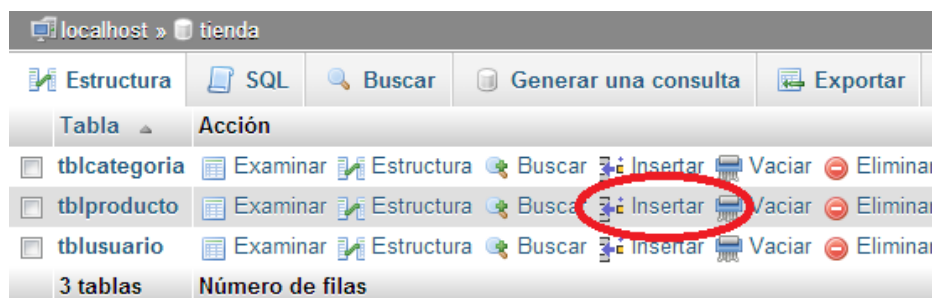
Alternativa opcional 3: Desde la web GenerateData

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 situado en la Parte Alta |

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

| idPrc | 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 leather L-0934 | 89 | 1 | Deportivas | | Deportivas Mike piel nº 42 |

- Guarda los datos y cierra la página.

Acceso a la base de datos MySQL desde aplicaciones Windows

El acceso a los datos desde otras aplicaciones permite ampliar su control y manipulación.

El conector ODBC:

Open DataBase Connectivity (ODBC) es un estándar de acceso a las bases de datos en Windows cuyo objetivo es acceder a cualquier dato desde cualquier aplicación, sin importar el sistema de bases de datos. ODBC permite utilizar la misma conexión por muchas aplicaciones que disponen de este método de conexión como **Access**, Visual C++ o **Visual Basic** de Microsoft© o **Delphi**, C++ Builder, de Borland/Embarcadero©.

Vamos a mostrar algunos ejemplos de acceso desde *Microsoft Office* o desde un software de programación como *Delphi* o *C++ Builder*. Ambos utilizan los mismos componentes y son lenguajes de programación orientado a objetos visuales, fáciles para diseñar y programar.

Agregar el driver ODBC para MySQL

En primer lugar hay que descargar este driver para MySQL desde la página web: Accede a la página de mysql.com y en Products – Conectores, descargar el driver ODBC para Windows, para vuestra versión de 32 o 64 bits. Una vez descargado, instalarlo en el equipo.

En segundo lugar, acceder a las herramientas administrativas del panel de control de Windows y, en orígenes de datos, *Agregar* una nueva conexión seleccionando el controlador MySQL e introducir los siguientes valores:

- Data Source Name: conexión_tienda
- Server: localhost
- User: root
- Database: tienda

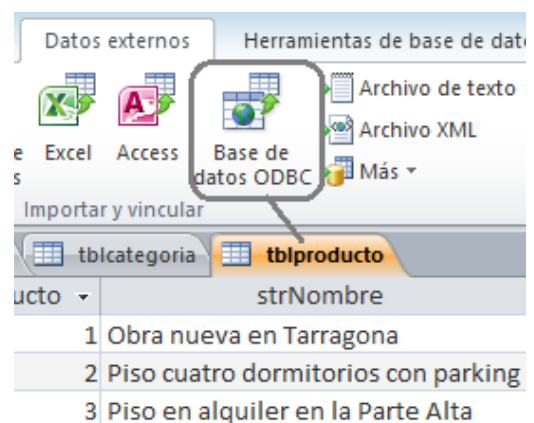
Pulsa en *Test* para comprobar que la conexión local es satisfactoria.



Opcional: Crea otra conexión llamada *conexión_remota* donde estableces los valores de conexión a la base de datos de tu servidor remoto si ya tienes alojada la base de datos en tu servidor web.

Ejemplo de conexión con Access 2010.

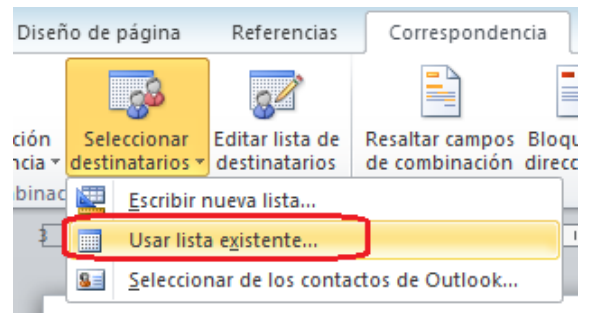
- Ejecuta Access y accede a la pestaña: *Datos externos*
- Pulsa en el botón: *Base de datos ODBC*
- Escoge: *Vincular al origen de datos*, de este modo actualizaremos los datos desde Access.
- Selecciona el origen de datos: *Conexión_tienda*
- Selecciona la tabla: *TblProducto* y pulsa Aceptar.



Ejemplo de combinar correspondencia con Word 2010

Si dispusiéramos de los datos de correo de los usuarios en la tabla *tblUsuarios*, podríamos crear *mailings* en Word de la siguiente forma:

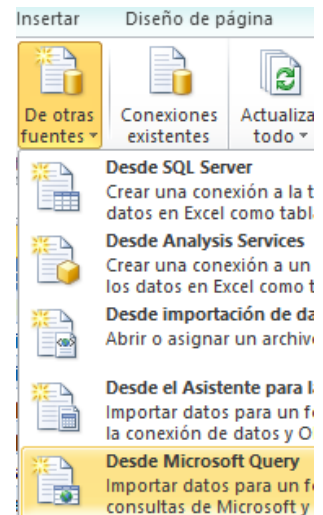
- Ejecuta *Word* y accede a la pestaña: *Correspondencia*
- Pulsa en *Seleccionar destinatarios*
- Escoge: *Usar lista existente*.
- Pulsa en: *Nuevo origen...*
- Selecciona: *DNS origen de datos ODBC*
- Selecciona el origen de datos: *Conexión_tienda*
- Selecciona la tabla: *TblUsuario* y pulsa en *Finalizar*.
- Pulsa sobre *Editar lista de destinatarios* de la cinta, para escoger a los destinatarios.
- Pulsa sobre *Seleccionar lista de destinatarios* de la cinta, para escoger a los destinatarios.
- Pulsa sobre: *Insertar campo combinado* para añadir los campos a la página.
- Pulsa en *Vista Previa de resultados* antes de Combinar correspondencia.



Ejemplo importar datos a Excel 2010.

Del mismo modo, para obtener una lista de usuarios o productos en Excel, podemos utilizar el servicio de Microsoft Query:

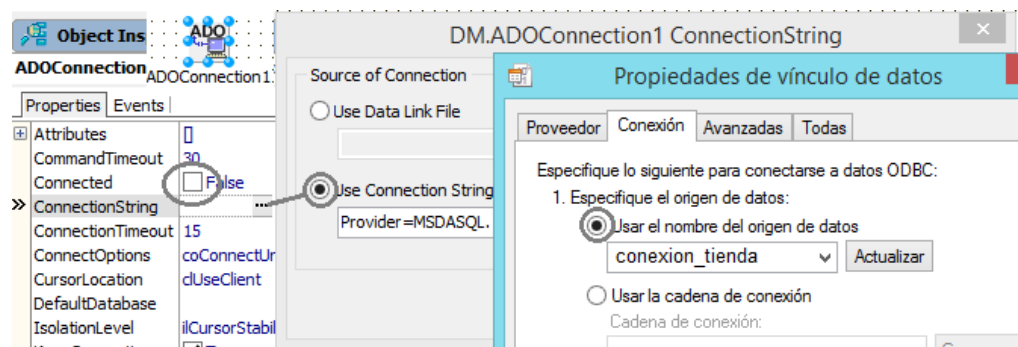
- En la pestaña datos, escoge de otras fuentes... *Microsoft Query*
- Selecciona el origen de datos: *Conexión_tienda*
- Selecciona la tabla: *TblUsuario* y pulsa Finalizar.



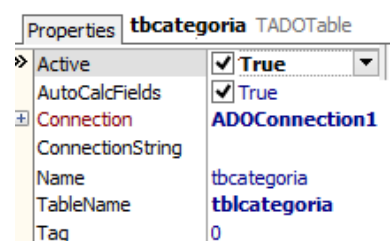
Ejemplo de conexión desde Delphi o C++ Builder.

- Accede al IDE visual del programa, ya sea Delphi o C++ Builder de Embarcadero/Borland©
- Buscar la paleta DbGO. Añade un componente un *ADOConnection* al formulario.
- Accede a la pestaña de *propiedades* de este componente *ADOConnection1*

- Escoge:
 - ConnectionString
 - Build...
 - Conexión
 - Usar el origen de datos



- Escoge el origen de datos de la *conexion_tienda* creada anteriormente. Desactiva el *LoginPromt* para que no nos pida el user cada vez. Activa la propiedad *Conected* = *True*. Si no conecta es que algo va mal.
- Añade un componente *ADOTable* de la paleta dbGo al formulario de datos.
- Introduce las propiedad *Name:TbProducto* y *Tablename: tblproducto*
- De la paleta *DataAccess*, añade un *Datasource* con nombre: *DSProducto* apuntando su *Dataset* a su respectiva tabla. En todos los componentes de datos que pongamos en nuestra ventana, estableceremos la propiedad *Datsource*: *DSProducto*



Nota: Este es simplemente un ejemplo base de cómo conectar una aplicación de escritorio accediendo a nuestros datos de la web.

LENGUAJE SQL EN DELPHI

Manipulación de Datos mediante lenguaje SQL: SELECT, INSERT, UPDATE, DELETE

SELECT: Extrae datos de una o más tablas indicando los campos (o columnas) separados por comas.

El asterisco (*) sustituye a todos los campos

Sintaxis (fórmula):

```
SELECT {Campo1 AS
NombreCampo1, Campo2, ...CampoN
}
FROM {Tabla/Vista}
[ WHERE {Condición} ]
[ LEFT JOIN OtraTabla1
```

Ejemplo:

```
SELECT * FROM CLIENTES
WHERE CIUDAD = "MEXICO" AND ESTADO = "DF"
```

FROM: Especifica la tabla o tablas (separadas por comas) de la cual se extraen los datos.

JOIN: Junta varias tablas a través del campo índice, que existe en las dos tablas.

WHERE: Filtra los registros que cumplan el criterio de condición. Estos pueden utilizar los operadores de comparación (=, <, > ...) o predicados adicionales IN, ANY, ALL, EXISTS.

```
SELECT * FROM PARTS WHERE PART NO >
```

ORDER BY: La cláusula Order by especifica el orden de extracción.

```
SELECT * FROM PARTS ORDER BY PART NAME ASC
```

GROUP BY: Especifica cómo se agrupan las filas o registros.

UNION: Combina el resultado de dos o más SELECTS para convertirla en una sola tabla. (JOIN)

INSERT: Añade un registro a la tabla APPEND Inserta el registro al final de la tabla:

Insert permite añadir datos al mismo tiempo mediante dos métodos:

1.- Añade una fila (registro) a la tabla asignando valores a dos columnas:

```
INSERT INTO FACTURAS
( FACNO, NOMBRECLIENTE, CIUDAD, ESTADO )
VALUES ( 99, 'Winston Churchill', 'Londres', 'La Mera
Capital' )
```

2.- Especifica valores a insertar mediante el estado SELECT:

este INSERT se combina con otro SELECT para copiar los renglones de la factura número dos y ponerlos en la factura número 99.

```
INSERT INTO FACTURARENGLON
( FACNO, NOMBRECLIENTE, CIUDAD, ESTADO )
SELECT 99, ITEMNO, DESCRIPTION, PRECIO_USD
FROM
FACTURARENGLON WHERE FACNO = 2
```

UPDATE: Modifica datos de la tabla. Ej: UPDATE GOODS SET CITY = 'SANTA CRUZ' WHERE GOODS.CITY = 'SCOTTS VALLEY'

DELETE: Borra datos de la tabla. Ej: DELETE FROM GOODS WHERE PART_NO = 'AA0093'

Funciones agregadas a SQL

SUM(), Total

AVG(), Promedio de valores no nulos

MIN(), mínimo valor

MAX(), máximo valor

COUNT(), cuenta el nº de valores según un criterio.

De fecha: Permite extraer de una fecha: YEAR, MONTH, DAY, HOUR, MINUTE y SECOND

```
SUM( Field * 10 )
SUM( Field ) * 10
SUM( Field1 + Field2 )
```

Substitución de variables:

En Delphi, cuando se añade una palabra precedida por un signo de dos puntos (:), la palabra es automáticamente convertida en un parámetro, el cual es agregado a la propiedad: Params

```
SELECT * FROM "CLIENTS.DBF" CLIENTES WHERE Last_Name LIKE :Apellido
```

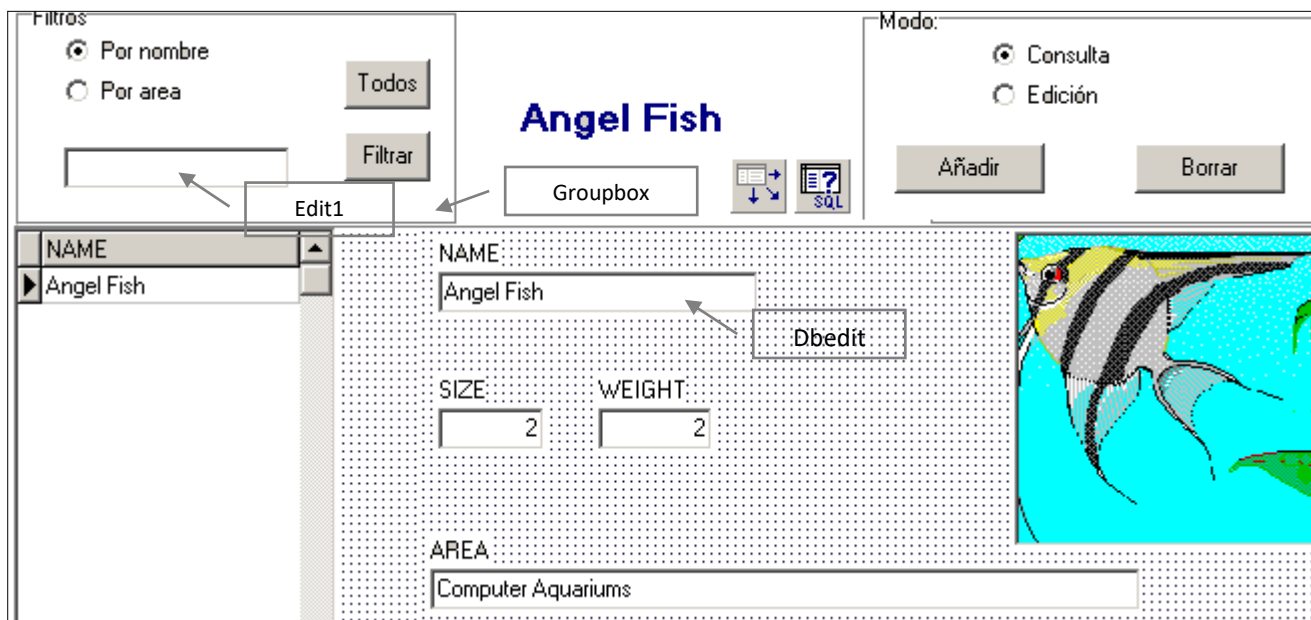
LIKE: devuelve todos los registros que se parezcan a la cadena. La línea que asigna el parámetro añade un signo de porcentaje (%) al final:

```
qryClientes.ParamByName('Apellido').AsString := dlgConsulta.edtApellido.Text+'%';
```

Ejercicio de consulta de datos en Delphi con SQL

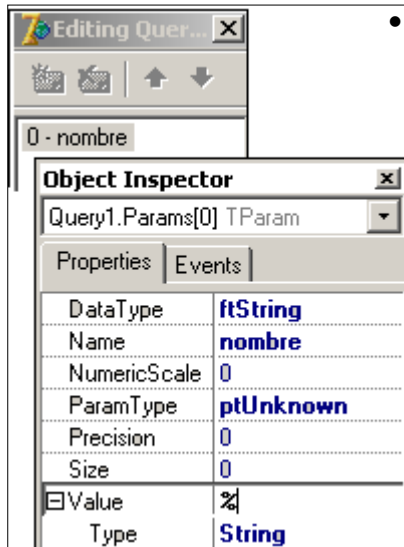
- Crea una aplicación nueva en Delphi. File ► New ► *Aplicación*.
- De la pestaña Data Access y DBE, inserta los componentes: **Query** y **DataSource** que representan las partes "no visuales" del acceso a base de datos.
- Ahora, vaya al TQuery y cambie la propiedad "Databasename" a Dbdemos, que es un **alias** local que apunta al directorio de Demos de Delphi.
- Selecciona el **Datasource**. En la propiedad Dataset, selecciona: **Query1**
- Selecciona el Query1, haz doble-click en su propiedad: **SQL** Escribe lo siguiente:


```
SELECT * from animals where upper(NAME) like :nombre ORDER BY NAME
```
- Obseva que :nombre es un parámetro. En la propiedad **Params** pulsa doble clic y pon su Datatype a string
- A continuación, pon la propiedad "**RequestLive**" de su Query en "**True**". Esto es importante porque SQL siempre devuelve resultados "Read-Only", o "muertos" (dead) a menos que especifiquemos que nuestro query debe ser "en vivo" (live)
- Haz doble-click en el query1. A continuación verás una ventana vacía. Haz clic con el boton derecho del mouse y selecciona: "**Add Fields**". Nos aparece la lista de las "columnas" (campos) disponibles. Así que simplemente presiona {Enter} para aceptar el añadir todos los campos.
- "Arrastra" desde la lista de campos del query1, hasta el formulario. Cuando "sueltes" cada campo, Delphi creará un componente TLabel y un DbEdit, listos para recibir los datos adecuados. Conecta todos los DBEdits con la propiedad Datasource al Datasource1. A asignar su datafield al campo correspondiente.



- Para ver los datos vamos a abrir el query1 (active = true). Completamos el formulario con los objetos que se muestran en la figura. En la propiedad *Params* del *Query 1* le añadimos el valor: % para que por defecto aparezcan todos.
- En el botón **Filtrar**, añadimos el código:

```
begin
  query1.active:=false;
  Query1.Params[0].Value :=
  UpperCase(edit1.Text)+'%';
  query1.active:=true;
end;
```



En el botón **Todos**, escribimos:

```
query1.active:=false;
Query1.Params[0].Value := '%';
query1.active:=true;
```

Notas:

- ▶ **Upper** en el SQL y **Uppercase** en el procedure, convierten ambos a mayúsculas para que no haya distinción entre mayúsculas y minúsculas.
 - ▶ El botón de radio: Edición, cambia la propiedad del query1: **RequestLive** a true.
 - ▶ Los botones añadir y borrar añaden el código: **Query1.Append**; y **Query1.Delete**; respectivamente.
- Para consultar por área, añadir otro query2 con este campo como parámetro en el SQL. Con el mismo datasource, cambiar de uno a otro, según se active uno u otro botón de radio.