

Bases de Datos en Delphi 7 - Conexión a bases de datos MySQL

Enviado por [Rodolfo Oscar Vallejos](#) |

1. [Objetivo](#)
2. [Introducción](#)
3. [Una base de datos de ejemplo](#)
4. [ODBC](#)
5. [Un formulario ODBC](#)
6. [ZeosDBO](#)
7. [Un formulario Zeos](#)
8. [Conclusión](#)
9. [Bibliografía y recursos Web](#)

Objetivo

Proporcionar mediante ejemplos interconexión de **Bases de Datos MySQL** utilizando los componentes ODBC y ZeosDBO.

Introducción

Una de las dificultades con las que se encuentra el estudiante de **programación** al comenzar con **Delphi 7** es la de encontrar **bibliografía** en **castellano** que lo oriente en la solución de sus **problemas** mediante la utilización de esta IDE

Este **trabajo** apunta a orientar al estudiante y/o programador de Delphi en la realización de una sencilla aplicación que permita la conexión y manipulación de **datos** en una DB MySQL.

Son requisitos previos tener instalado MySQL en su **sistema**, instalar los componentes ZeosDBO, tener conocimientos de programación en Delphi y cómo manejar bases de datos **SQL**.

Una base de datos de ejemplo

Para comenzar deberá crear una base de datos MySQL, si no tiene alguna aplicación que le permita hacer esto, ingrese al **cliente** MySQL que viene con la instalación, desde una ventana DOS y mediante:

```
C:\mysql\bin\mysql.exe -h localhost -u root
```

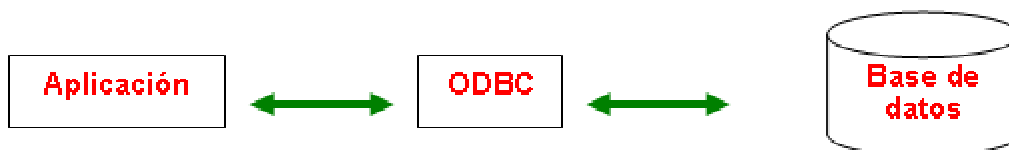
Una vez ingresado al **servidor**, utilizando **comandos SQL** cree una base de datos pertenecientes a una agenda **personal**:

```
Mysql> CREATE DATABASE agenda;  
Mysql> USE agenda;  
Mysql> CREATE TABLE myagenda (nombre  
VARCHAR(20), domicilio VARCHAR(30),  
telefono VARCHAR(10), localidad  
VARCHAR(20), codigopostal INT(4));
```

Terminada la creación de la tabla **myagenda**, salga del cliente MySQL por medio del comando **quit**.

ODBC

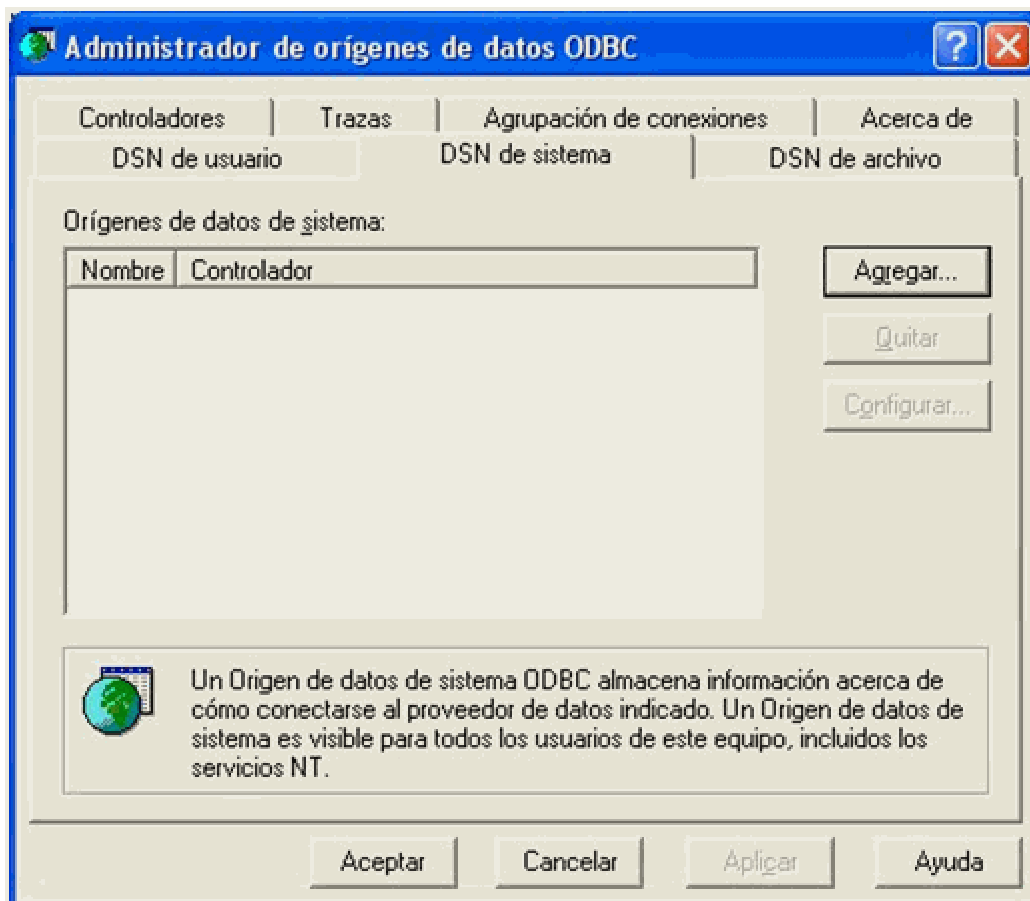
ODBC es una **tecnología** de conectividad a distintos tipos de bases de datos y se encuentra presente desde las primeras versiones de **Windows**. Básicamente es un puente que permite la conexión de cualquier aplicación con un origen de datos tal como lo **muestra** la figura:



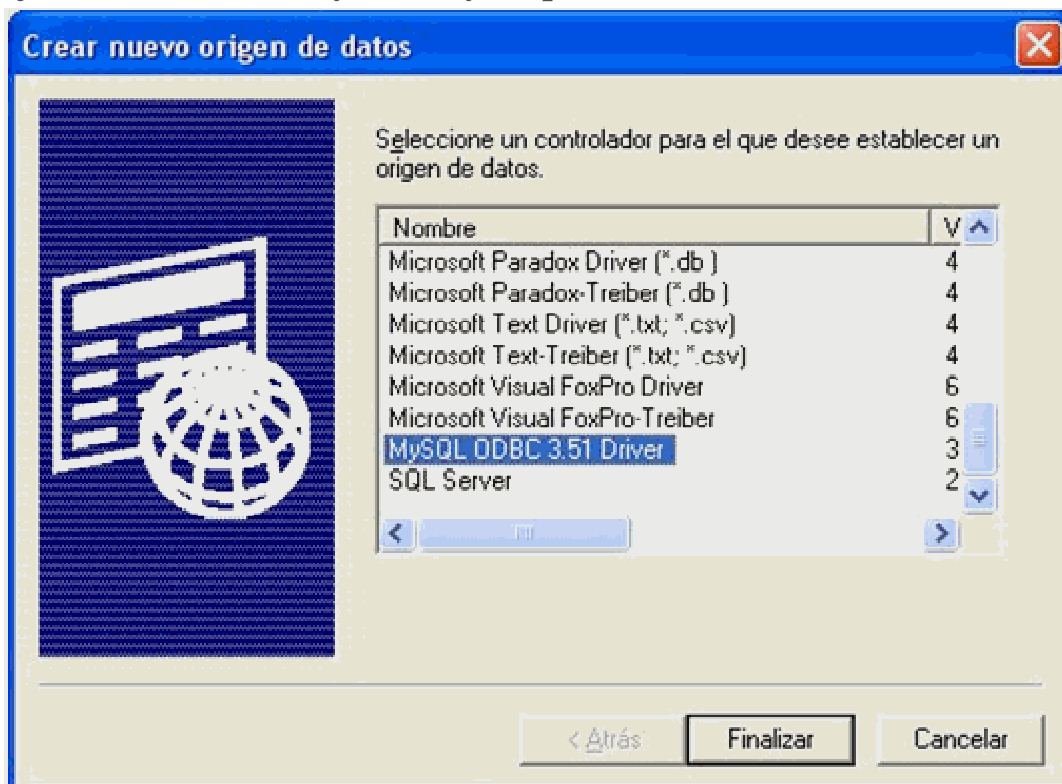
El origen de datos puede apuntar a una base de datos local o a una remota utilizando **servicios de red**.

MySQL posee un puente ODBC (MyODBC) que puede ser bajado del sitio en forma gratuita y viene con un instalador de fácil ejecución.

Para crear un **Origen de datos** desde el menú **Inicio** seleccione **Configuración** y desde el **Panel de Control**, en **Herramientas administrativas** elija **Orígenes de datos (ODBC)** y posicione en la pestaña **DSN de sistema**, haga clic en agregar



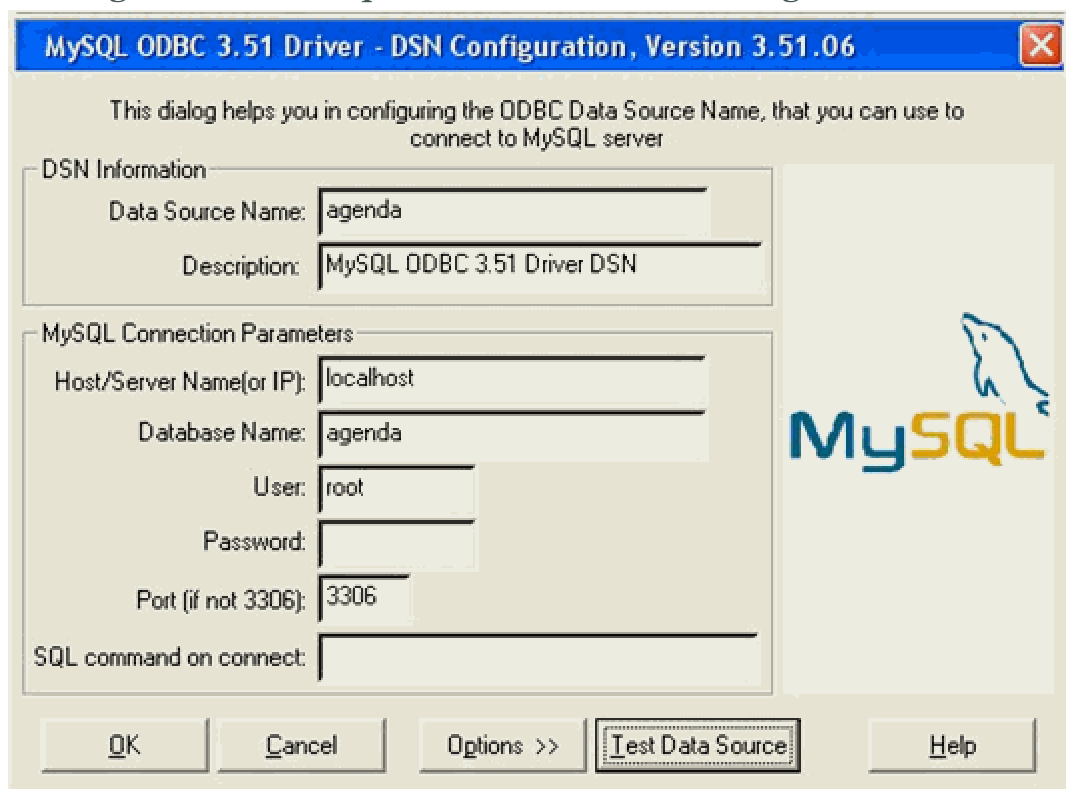
Elija el controlador MyODBC y cliquee el botón **Finalizar**.



A continuación llene los **datos** correspondientes a la agenda en la pantalla de configuración MyODBC:

- Nombre de nuestra conexión ODBC (Data Source Name): agenda
- Nombre del **servidor** (Host/Server name (IP)): se puede colocar el nombre de la PC local (localhost), el nombre de un equipo remoto que contenga el **administrador MySQL** o la **dirección IP**. Para nuestro caso será: localhost
- Nombre de la **base de datos** (Database name): agenda
- Nombre del usuario (User): root
- Password: en el caso de que el usuario la tenga (muy recomendable)

La configuración debe quedar como **muestra** la figura



Para asegurarse que todo se realizó correctamente presione el botón **Test Data Source** y el puente ODBC le informará si la conexión con la base de datos es satisfactoria.

Un formulario ODBC

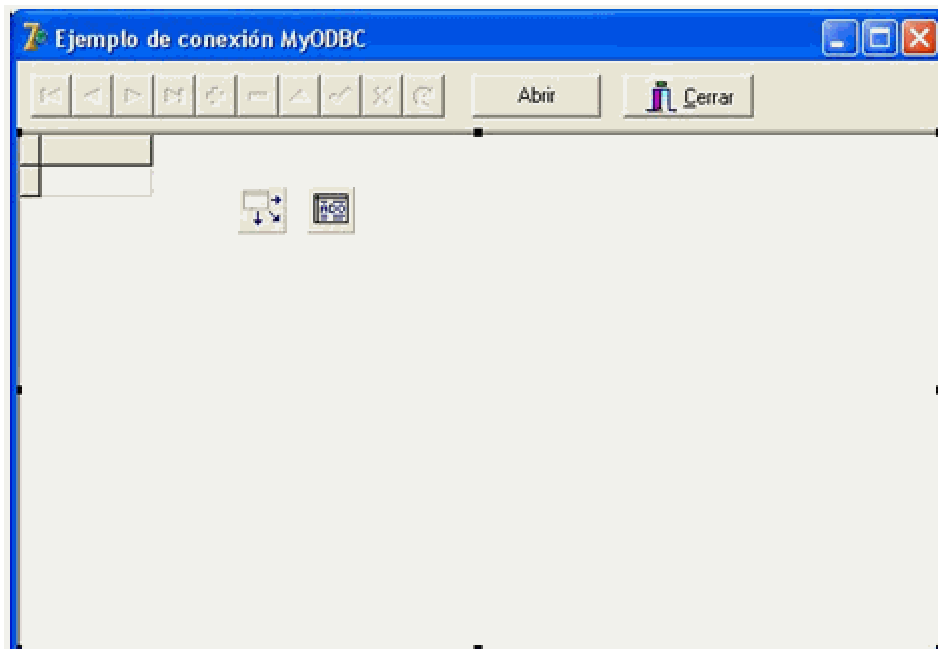
Ejecute **Delphi 7**. En un formulario vacío coloque los siguientes componentes:

- Un panel con la alineación (Align) en **Top**.
- Dentro del panel un TDBNavigator
- Un botón: Name: **Control**, Caption: Abrir

- Un botón TbitBtn: Name:Cerrar, Kind: bkClose, Caption: Cerrar
- Un TDBGrid con alineación (Align) AllClient
- De la pestaña ADO seleccione un TTable y modifique las siguientes propiedades
 - Name: Tabla
 - Connectionstring:
Provider=MSDASQL.1;Persist
Security Info=False;User
ID=root;Data Source=agenda
 - TableName: myagenda
- De la pestaña DataAccess seleccione el componente TDataSource. Asocie el DataSet con la Tabla
- Actualice los DataSource del DBNavigator y DBGrid con el TDataSource.

Para comprobar que todo está correcto, modifique la **propiedad Active** del TTable a True y verá en el DBGrid la **estructura** de la agenda creada anteriormente.

El formulario terminado deberá verse como en la figura:



Compile el formulario y ejecute su aplicación

Para que la misma sea funcional debe escribir algo de **código**:

Cliquee en el formulario el botón **Abrir** y en el editor de código escriba lo siguiente:

```
procedure TForm1.ControlClick(Sender: TObject);
begin
  If Control.Caption='Abrir' then
  begin
    Control.Caption := 'Cerrar';
    Tabla.Open;
  end
  Else
  begin
    Control.Caption := 'Abrir';
    Tabla.Close;
  end;
end;
```

Escriba el **método** del botón Cerrar

```
procedure TForm1.CerrarClick(Sender: TObject);
begin
  Tabla.Close;
  Halt(o);
end;
```

Compile el formulario y ejecute la aplicación.

Verá como con un mismo control puede abrir y cerrar la base de datos. Ingrese **registros** oprimiendo el botón con el símbolo más (+) del TDBNavigator e inserte los registros en la base de datos por medio del botón con el símbolo de tilde (✓)

ZeosDBO

ZeosDBO es un conjunto de componentes creados por ZeosLib Development Group, de **distribución** gratuita y posibilitan la conectividad con distintas **bases de datos**: MySQL, PostgreSQL, etc. Descargue el paquete desde el sitio de ZeosLib e instálelos en su **sistema** mediante los siguientes **procedimientos**:

1. Descomprima el **archivo** .zip dentro de la carpeta *Archivos de **programas**\Borland\Source\Zeos*

2. Copie la DLL correspondiente a su versión de MySQL que se encuentra en la carpeta *Archivos de programas\Borland\Source\Zeos\lib\mysql* a la carpeta *WINNT\SYSTEM32* o *WINDOWS\SYSTEM32*
3. Agregue el directorio *Zeos\packages\delphi7\build* a la opción Library Path de Delphi
4. De la carpeta *Zeos\packages\delphi7* compile ZCore.bpl, ZParseSql.bpl, ZPlain.bpl y ZDbc.bpl
5. Si los paquetes anteriores se compilan sin **problemas**, compile y luego instale ZComponent.bpl. En la IDE aparecerá una nueva paleta de componentes como en la siguiente figura:



De izquierda a derecha los componentes son los siguientes:

ZConnection: Conexión a la Base de Datos

ZReadOnlyQuery: Igual a TQuery pero de sólo lectura

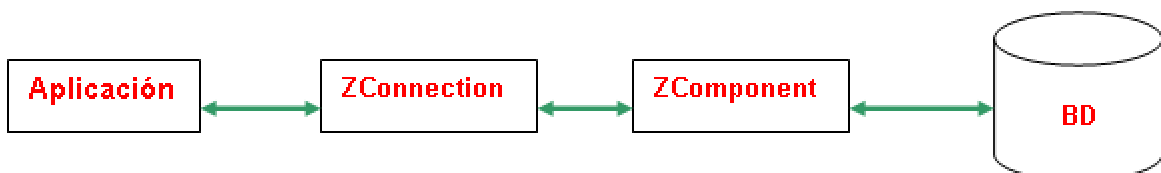
ZQuery: Igual a TQuery

ZTable: Igual a TTable

ZUpdateSQL: Componente transaccional

ZStoredProc: Llamadas a procedimientos almacenados (MySQL 5.0 y superior)

Al contrario de ODBC, Zeos se maneja con una estructura distinta: todos los procedimientos de **lectura/escritura** se realizan a través de una conexión a la base de datos, ya sea local o en cualquier sitio de **una red** donde se encuentre instalado el servidor MySQL



Un formulario ZEOS

De la misma manera que en el ejemplo anterior, inicie un nuevo **proyecto** en Delphi agregando al formulario todos los componentes del ejemplo ODBC menos la Ttable.

Agregue un Zconnection modificando las siguientes propiedades:

Database: agenda

HostName: localhost o el nombre del equipo/dirección IP

User: root (en caso de poner otro nombre de usuario proveer la clave de acceso en la propiedad Password)

Protocol: la versión del servidor MySQL

Nota: Si su aplicación correrá en varias **computadoras** de una **red** **debe colocar en la carpeta System de cada una de ellas el DLL libmysqlXXX.dll correspondiente a la versión del servidor**

Agregue un ZTable y modifique las siguientes propiedades:

Connection: ZConnection1

TableName: myagenda

El código del botón que maneja la apertura/cierre de la base es el siguiente:

```
procedure TForm1.ControlClick(Sender: TObject);
```

```
begin
```

```
If Control.Caption='Abrir' then
```

```
begin
```

```
Control.Caption := 'Cerrar';
```

```
ZConnection1.Connect;
```

```
ZTable.Open;
```

```
end
```

```
Else
```

```
begin
```

```
Control.Caption := 'Abrir';
```

```
ZTable.Close;
```

```
ZConnection1.Disconnect;
```

```
end;
```

```
end;
```

El cierre del formulario

```
procedure TForm1.CerrarClick(Sender: TObject);
```

```
begin  
ZTable.Close;  
ZConnection.Disconnect;  
Halt(o);  
end;
```

Conclusión

Estos ejemplos bastan para ilustrar brevemente dos mecanismos para la conexión y manipulación de datos utilizando MySQL, tanto en forma local como remota.

El manejo de datos utilizando los componentes xQuery, tanto para ADO como para Zeos, son similares a el uso de componentes BDE referido a consultas, maestro-detalle, etc.

Vaya aquí mi agradecimiento a los integrantes de la lista IDelphi del **Grupo Albor** quienes ayudan permanentemente a solucionar problemas de **programación** en esta IDE

Bibliografía y **recursos** WEB

Dubois, Paul. *MySQL*. Prentice Hall; Barcelona, 2000

Programming whit Delphi. Ayuda en línea de la IDE

Sitio de programación dedicado a Delphi (en **inglés**):

<http://delphi.about.com>

Sitio de ZeosLib (en inglés): **<http://www.zeoslib.net>**

Rodolfo Oscar Vallejos

[rodolfovallejos\[arroba\]yahoo.com](mailto:rodolfovallejos@yahoo.com)

Rosario (Santa Fe) - Argentina