

Conexión Base de Datos


Lenguajes de Programación 2



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

Dr. Freddy Paz

ADO .NET

 **ADO.NET** es un conjunto de componentes del software que pueden ser usados por los programadores para acceder a datos y a servicios de datos. Es una parte de la biblioteca de clases base que están incluidas en el **Microsoft .NET Framework**. Es comúnmente usado por los programadores para acceder y para modificar los datos almacenados en un **Sistema Gestor de Bases de Datos Relacionales**.

Driver MYSQL para .NET



Versión 4.5

- **MySql.Data**
- **MySql.Data.Entity.EF5**
- **MySql.Data.Entity.EF6**
- **MySql.Fabric.Plugin**
- **MySql.Web**
- <https://dev.mysql.com/downloads/connector/net/>



Cadena de Conexión



La **cadena de conexión** provee la información necesaria para realizar la conexión a la base de datos MySQL.

Tiene la siguiente estructura:

```
"server=localhost;user=root;database=world;port=3306;  
password=*****;"
```

```
"server=localhost;user=root;database=world;port=3306;  
password=*****;SslMode=none=none;"
```

MySQLConnection

Este objeto establece la conexión a una Base de Datos MySQL a partir de una **cadena de conexión**.

```
MySQLConnection conn = new MySQLConnection(cadena);  
conn.Open();  
...  
conn.Close();
```



Ejemplo

```
try
{
    string cadena =
"server=localhost;user=prueba;database=prueba;port=3306;password=**
*****";
    MySqlConnection conn = new MySqlConnection(cadena);
    conn.Open();
    Console.WriteLine("La conexión se ha realizado con éxito");
    conn.Close();
}
catch (Exception e) {
    Console.WriteLine(e.Message)
}
Console.ReadKey();
```



MySqlCommand

Una vez que se ha establecido la conexión con la base de datos MYSQL, el siguiente paso es ejecutar operaciones (enviar sentencias SQL). Esto es posible a través de un objeto tipo MySqlCommand.

```
MySqlCommand cmd = new MySqlCommand()
```



MySqlCommand

El método **ExecuteReader(sqlString)** es para sentencias SQL de tipo SELECT.

El método **ExecuteNonQuery(sqlString)** es para sentencias SQL de tipo INSERT, UPDATE y DELETE.



INSERT

```
MySQLCommand cmd = new MySQLCommand();  
string sql = "INSERT INTO empleado " +  
    "(dni,nombres, apellido_paterno,apellido_materno) "  
+    "values " +  
    "('28112987','Juan','Perez','Ortega')";  
cmd.Connection = conn;  
cmd.CommandText = sql;  
cmd.ExecuteNonQuery();
```



UPDATE

```
MySqlCommand cmd = new MySqlCommand();  
string sql = "UPDATE empleado SET nombres = 'Maria' "  
            + "where dni = 28112987";  
  
cmd.Connection = conn;  
cmd.CommandText = sql;  
cmd.ExecuteNonQuery();  
conn.Close();
```



DELETE

```
MySQLCommand cmd = new MySQLCommand();
```

```
string sql = "DELETE FROM empleado where dni =  
28112987";
```

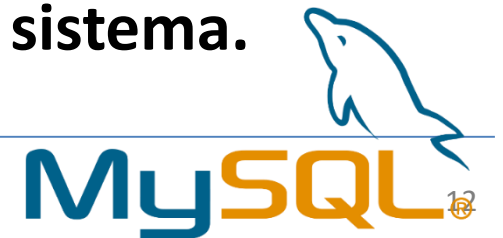
```
cmd.Connection = conn;  
cmd.CommandText = sql;  
cmd.ExecuteNonQuery();
```



MySqlDataReader

Se usa el **MySqlDataReader** para recuperar flujos de datos de solo lectura y solo avance de una base de datos. Los resultados se devuelven a medida que se ejecuta la consulta y se almacenan en el búfer del cliente hasta que se solicitan con el método **Read** del **MySqlDataReader**.

Con el **MySqlDataReader** puede aumentar el rendimiento de la aplicación al recuperar datos en cuanto están disponibles y almacenar (de forma predeterminada) una sola fila cada vez en memoria, lo que reduce la sobrecarga del sistema.



SELECT

```
MySqlCommand cmd = new MySqlCommand();
string sql = "SELECT * FROM empleado";
cmd.Connection = conn;
cmd.CommandText = sql;
MySqlDataReader reader = cmd.ExecuteReader();
while (reader.Read())
{
    string dni = reader.GetString("dni");
    string nombres = reader.GetString("nombres");
    string apellido_paterno = reader.GetString("apellido_paterno");
    string apellido_materno = reader.GetString("apellido_materno");
    Console.WriteLine(dni+" "+nombres+" "+apellido_paterno+"
        "+apellido_materno);
}
```



SELECT

```
MySQLCommand cmd = new MySQLCommand();
```

```
string sql = "DELETE FROM empleado where dni =  
28112987";
```

```
cmd.Connection = conn;  
cmd.CommandText = sql;  
cmd.ExecuteNonQuery();
```



MySqlDataAdapter

Un **MySqlDataAdapter** es un objeto que permite recuperar todos los registros (filas) que satisfacen las condiciones impuestas en una sentencia SQL de tipo SELECT.



DataSet

DataSet representa un conjunto de datos en memoria.

El método **Fill** de **MySqlDataAdapter** se usa para rellenar un objeto **DataSet** con los resultados del elemento **SelectCommand** de **MySqlDataAdapter**.



SELECT

```
string sql = "SELECT * FROM empleado";
MySqlDataAdapter adaptador = new MySqlDataAdapter(sql,conn);
DataSet ds = new DataSet();
adaptador.Fill(ds, "empleados");
foreach (DataRow row in ds.Tables["empleados"].Rows)
{
    string dni = row["dni"].ToString();
    string nombres = row["nombres"].ToString();
    string apellido_paterno = row["apellido_paterno"].ToString();
    string apellido_materno = row["apellido_materno"].ToString();
    Console.WriteLine(dni + " " + nombres + " " + apellido_paterno +
        " " + apellido_materno);
}
```



Conexión

```
try
{
    string cadena =
"server=localhost;user=prueba;database=prueba;password=*****;";
    SqlConnection conn = new SqlConnection(cadena);
    conn.Open();
    Console.WriteLine("La conexión se ha realizado con éxito");
    conn.Close();
}
catch (Exception e) {
    Console.WriteLine(e.Message);
}
Console.ReadKey();
```

