

¿QUÉ ES ADO.NET?

ADO .NET es la nueva versión del modelo de objetos ADO (ActiveX Data Objetos), es decir, la estrategia que ofrece Microsoft para el acceso a datos. ADO .NET ha sido ampliado para cubrir todas las necesidades que **ADO** no ofrecía, **ADO .NET** está diseñado para trabajar con conjuntos de datos desconectados, lo que permite reducir el tráfico de red. ADO .NET utiliza XML como formato universal de transmisión de los datos.

ADO .NET posee una serie de objetos que son los mismos que aparecen en la versión anterior de **ADO**, como pueden ser el objeto **Connection** o **Command**, e introduce nuevos objetos tales como el objeto **DataReader**, **DataSet** o **DataView**.

ADO .NET se puede definir como:

- ✎ Un conjunto de interfaces, **clases**, estructuras y enumeraciones que permiten el acceso a los datos desde la plataforma **.NET** de **Microsoft**.
- ✎ Que permite un modo de acceso **desconectado** a los datos que pueden provenir de múltiples fuentes de datos de diferente arquitectura de almacenamiento.
- ✎ Y que soporta un completo modelo de programación y adaptación basado en el estándar **XML**.

Las clases de ADO .NET

En el presente apartado vamos a enumerar brevemente el conjunto de clases que forman parte del **API** de **ADO .NET**.

Primero vamos a comentar los distintos **Namespaces** que constituyen la tecnología ADO .NET:

- ✎ **System.Data**: clases genéricas de datos de **ADO .NET**, integra la gran mayoría de clases que habilitan el acceso a los datos de la arquitectura **.NET**.

- ☞ **System.Data.SqlClient**: clases del proveedor de datos de **SQL Server**, permiten el acceso a proveedores **SQL Server** en su versión **7.0** y **Superior**.
- ☞ **System.Data.OleDb**: clases del proveedor de datos de **OleDb**, permiten el acceso a proveedores **.NET** que trabajan directamente contra controladores basados en los ActiveX de Microsoft.
- ☞ Dentro del espacio de nombres **System.Data** encontramos las clases compartidas que constituyen el eje central de **ADO: NET**, y son las siguientes:
- ☞ **DataSet**: almacén de datos por excelencia en **ADO .NET**. Representa una base de datos desconectada del proveedor de datos. Almacena tablas y sus relaciones.
- ☞ **DataTable**: un contenedor de datos. Estructurado como un conjunto de filas (**DataRow**) y columnas (**DataColumn**).
- ☞ **DataRow**: registro que almacena n valores. Representación en **ADO .NET** de una fila/tupla de una tabla de la base de datos.
- ☞ **DataColumn**: contiene la definición de una columna. Metadatos y datos asociados a su dominio.

LOS PROVEEDORES DE ACCESO A DATOS

Es la capa inferior de la parte correspondiente de acceso a datos y es la responsable de establecer la comunicación con las fuentes de datos.

En este conjunto de nombre de espacios, encontraremos casi siempre las clases **Connection**, **Command**, **DataAdapter** Y **DataReader** como las clases más generales, las cuales nos permiten establecer la conexión con la fuente de datos.

Proveedores de acceso a datos de .NET Framework

Dentro del entorno .NET Framework, encontraremos un nutrido conjunto de proveedores de acceso a datos.

- ❶ ODBC .NET Data Provider
- ❶ OLEDB .NET Data Provider
- ❶ ORACLE Client .NET Data Provider
- ❶ SQL Server .NET Data Provider

Estos proveedores de acceso a datos incluidos en Microsoft .NET Framework, los podemos encontrar en los nombres de espacio.

- ❶ Imports System.Data.SqlClient.
- ❶ Imports System.Data.OleDb.
- ❶ Imports CoreLab.MySql.
- ❶ Imports System.Data.Odbc.
- ❶ Imports System.Data.Oracle.

🔗 **El Objeto Connection:** este objeto es el encargado de establecer una conexión física con una Base de Datos determinada.

🔗 **El Objeto Command:** Este objeto es el que representa una determinada sentencia SQL o un Stored Procedure, conjuntamente con el objeto **DataAdapter**.

🔗 **El Objeto DataAdapter:** Este objeto es quizás el objeto más complejo y a la vez complicado de todos los que forman parte de un proveedor de acceso a datos en NET. Cuando se establece una comunicación entre una fuente de datos y un **DataSet**, se utiliza como intermediario a un objeto DataAdapter. A su vez, un DataAdapter contiene 4 objetos que son.

1. **selectCommand:** Realiza las acciones a de selección de datos.
2. **DeleteCommand:** Realiza las acciones de borrados de datos.
3. **InsertCommand:** Realiza las acciones de inserción de datos.
4. **UpdateCommand:** Realiza las acciones de actualización de datos.

- 🔗 **El objeto DataReader:** Es utilizado en una sola dirección de datos. Se trata de un objeto muy rápido. Puede usar a su vez el objeto **Command** o el método **ExecuteReader**.

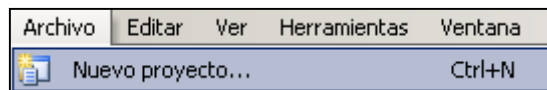
CONCEPTO DATABINDING

Databinding es una expresión de enlace de datos. Es una forma rápida y sencilla de manejar la fuente de datos mediante su enlace con controles (TextBox) o clases.

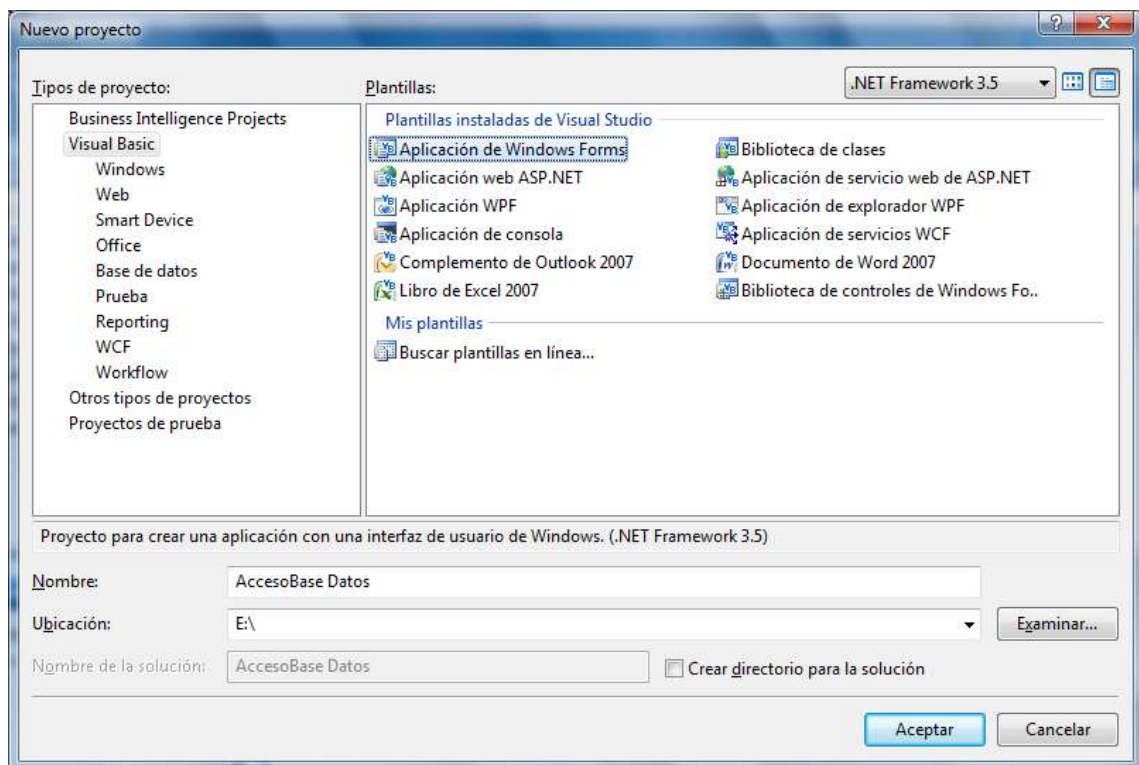
ACCESO A BASE DE DATOS

Crear un proyecto

1. Hacer clic en el Menú **Archivo > Nuevo proyecto**.



2. Aparecerá un cuadro de Dialogo llamado **Nuevo Proyecto**.



3. El diseño del formulario es el siguiente gráfico.



Importar los siguientes namespace.

```
Imports System.Data  
Imports System.Data.SqlClient
```

Modo Autenticación de Windows

```
Dim CN As New SqlConnection("integrated security=true;data source=(local);  
catalog=Neptuno")  
CN.Open()  
MsgBox("me conecte modo windows")  
CN.Close()
```

Modo SQL-server

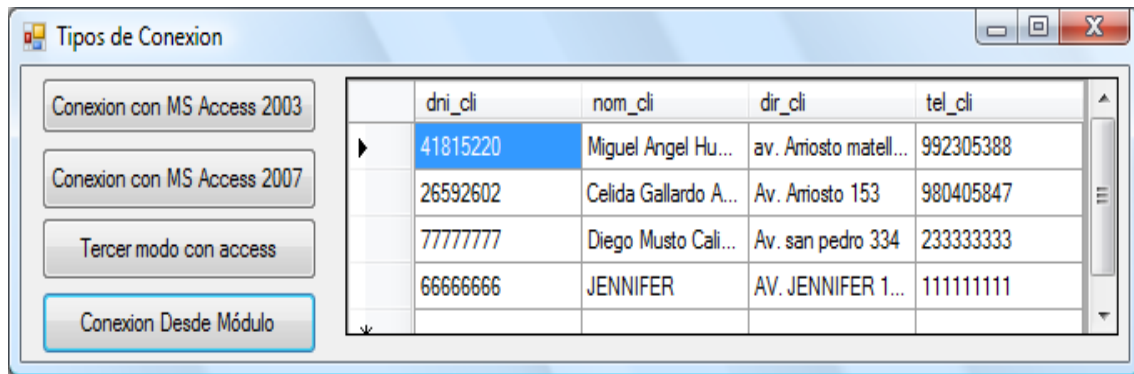
```
Dim cn As New SqlConnection("server=(local);database=Neptuno;user  
id=sa;password=VISTARE2")  
  
cn.Open()  
  
MsgBox("me conecte modo sql")  
  
cn.Close()
```

Tercera forma

```
Dim cn As New  
SqlConnection("server=(local);database=Neptuno;integrated security=true")  
  
cn.Open()  
  
MsgBox("tercemodo")  
  
cn.Close()
```

Conexión con Proveedor OLEDB (Ms Access 2003, 2007)

Añadir un nuevo Windows forms al proyecto.



Importar los siguientes namespace.

```
Imports System.Data
Imports System.Data.OleDb
```

Botón: MS Access 2003

```
Dim CN As New OleDbConnection("Provider=Microsoft.Jet.oledb.4.0;Data
Source=D:\almacen.mdb") 'conexion a BD ver 2003
CN.Open()
MsgBox("ok")
```

Botón: MS Access 2007

```
Dim cnx As New OleDbConnection("Provider=Microsoft.ace.Oledb.12.0;Data
Source=D:\Eduotec.accdb")
cnx.Open()
MsgBox("Conexión conexitó")
```

Tercera forma con Ms Access 2003

```
Dim Ruta As String = Application.StartupPath
Dim strCn As String = "Provider=Microsoft Jet 4.0 OLE DB Provider;Data
Source=" & Ruta & "\almacen.MDB" 'la bd esta ubicado en la carpeta bin del proyecto
Dim sql As String = "select*from productos"
Dim da As New OleDbDataAdapter(Sql, strCn)
Dim dt As New DataTable
da.Fill(dt)
Me.DataGridView1.DataSource = dt
```

Conexión desde un Módulo

```
Public CN As New OleDbConnection
Public Sub CONECTAR()
    If CN.State = ConnectionState.Open Then
        CN.Close()
    End If
    Dim ruta As String ruta = Mid(Application.StartupPath, 1,
Len(Application.StartupPath) - 9) & "EDUTEC.accdb;"
    'Ubicar la base de datos en la carpeta bin del proyecto
    CN.ConnectionString = "provider=microsoft.ace.oledb.12.0;data source=" & ruta
    CN.Open()
End Sub

'en el Boton
Call CONECTAR()
MsgBox("OK")
```

Practica N # 1

1. Desarrollar una aplicación que nos permita seleccionar una base de datos **Microsoft Excel 2007**.