

# ESPECIALIZACIÓN ASP.NET 5.0 DEVELOPER





# Fundamentos de Programación

Trabajando a  
Base de Datos

Instructor: Erick Aróstegui  
[earostegui@galaxy.edu.pe](mailto:earostegui@galaxy.edu.pe)



# TEMAS

## Mi primer programa

01

ADO.NET y conexiones a nivel de IDE

02

Principales servicios de acceso a datos

03

Creación de la base y tablas

04

Creando una aplicación de consola básica

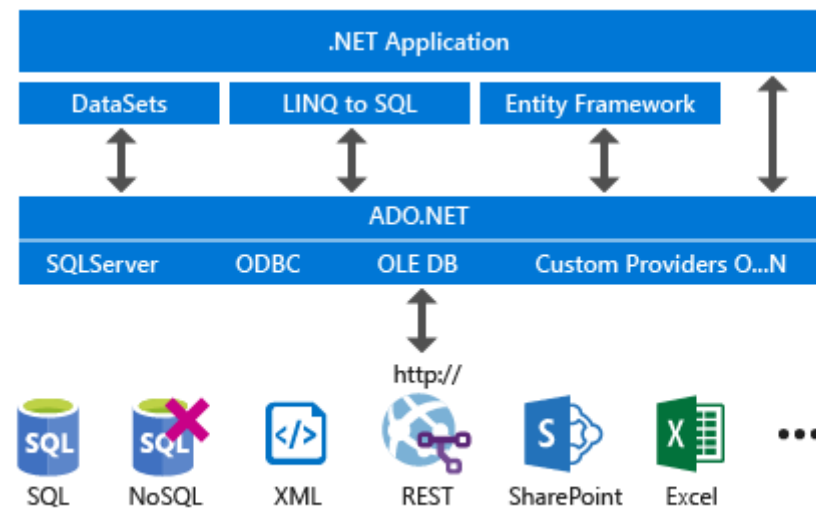
05

Creando una aplicación web básica



# ADO.NET

Es un conjunto de clases que exponen servicios de acceso a datos para programadores de .NET Framework. ADO.NET ofrece abundancia de componentes para la creación de aplicaciones de uso compartido de datos distribuidas. Constituye una parte integral de .NET Framework y proporciona acceso a datos relacionales, XML y de aplicaciones. ADO.NET satisface diversas necesidades de desarrollo, como la creación de clientes de base de datos front-end y objetos empresariales de nivel medio que utilizan aplicaciones, herramientas, lenguajes o exploradores de Internet.



## —• Conexiones a nivel de IDE

C# y Visual Basic son lenguajes de programación diseñados para crear una gran variedad de aplicaciones que se ejecutan en .NET Framework. Estos lenguajes con seguridad de tipos son eficaces y están orientados a objetos. Se basan en .NET Compiler Platform “Roslyn”, que proporciona API de análisis de código sofisticadas y es de código abierto en GitHub.

C# es un lenguaje de programación moderno, basado en objetos y con seguridad de tipos. que permite a los desarrolladores crear muchos tipos de aplicaciones seguras y sólidas que se ejecutan en el ecosistema de .NET.

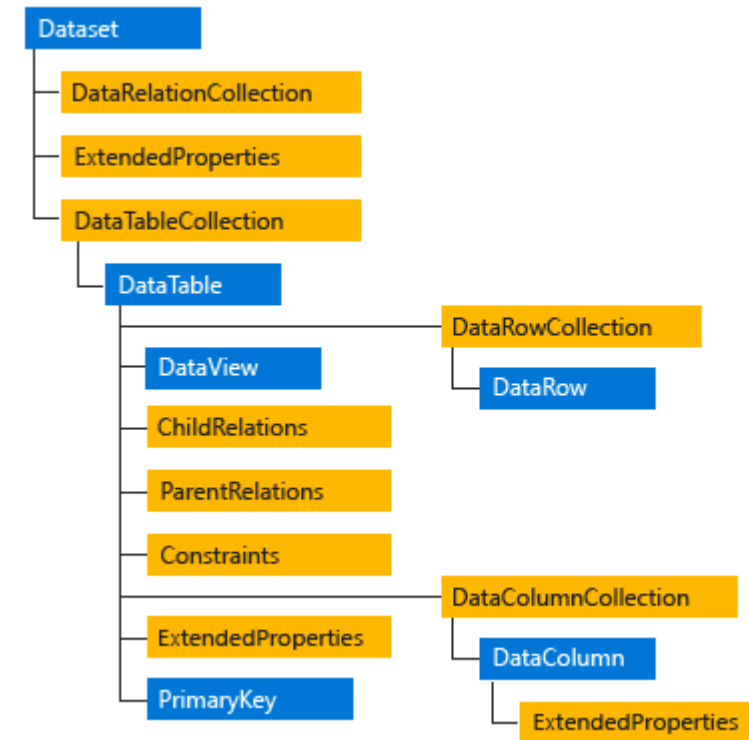


## — Principales servicios de acceso a datos(ADO.NET)

Proveedores de datos .NET Framework: Los proveedores de datos .NET Framework son componentes diseñados explícitamente para la manipulación de datos y el acceso rápido a datos de solo lectura y solo avance. El objeto **Connetion** proporciona conectividad a un origen de datos. El objeto **Command** permite tener acceso a comandos de base de datos para devolver datos. **DataReader** proporciona un flujo de datos de alto rendimiento desde el origen de datos. **DataAdapter** proporciona el puente entre el objeto DataSet y el origen de datos.



**DataSet:** DataSet de ADO.NET está expresamente diseñado para el acceso a datos independientemente del origen de datos. Como resultado, se puede utilizar con múltiples y distintos orígenes de datos, con datos XML o para administrar datos locales de la aplicación. DataSet contiene una colección de uno o más objetos DataTable formados por filas y columnas de datos, así como información sobre claves principales, claves externas, restricciones y de relación relacionada con los datos incluidos en los objetos DataTable



## — Principales servicios de acceso a datos(LINQ)

LINQ admite consultas en un modelo de objetos asignado a las estructuras de datos de una base de datos relacional sin utilizar un modelo conceptual intermedio. Cada tabla se representa mediante una clase distinta, acoplando de manera precisa el modelo de objetos al esquema de la base de datos relacional. LINQ to SQL convierte las consultas de Language-Integrated Query del modelo de objetos a Transact-SQL y las envía a la base de datos para su ejecución. Cuando la base de datos devuelve los resultados, LINQ to SQL los vuelve a traducir a objetos. Para más información





## — Creación de la base de datos

Crear la base de datos implica indicar los archivos y ubicaciones que se utilizarán para la misma, además de otras indicaciones técnicas y administrativas que no se comentarán en este tema. Lógicamente solo es posible crear una base de datos si se tienen privilegios de DBA(DataBase Administrador)

```
CREATE DATABASE FARMACIA--CREACION DE LA BASE DE DATOS
```

# — Creación de Tablas

Cada columna es un campo y cada fila un registro, cada campo debe tener su respectivo tipo de dato en base a los datos que se le registrarán. Una base de datos almacena su información en tablas.

Las tablas se componen de dos estructuras:

- Campo: Corresponde al nombre de la columna. Debe ser único y además de tener un tipo de dato asociado a los registros que se insertaran.
- Registro: Corresponde a cada fila que compone la tabla. Allí se componen los datos y los registros. Eventualmente pueden ser nulos en su almacenamiento.

```
CREATE DATABASE FARMACIA--CREACION DE LA BASE DE DATOS
```

```
--CREACION DE TABLAS
```

```
CREATE table Clientes
```

```
(  
  cod_cli varchar(8) primary key not null,  
  nom_cli varchar(40)not null,  
  dir_cli varchar(40) null,  
  cod_dis varchar(8)null,  
  sexo varchar(1)not null,  
  DNI INT NULL,  
  RUC int null,  
  Telefono int null,  
  Celular int null  
)  
go
```

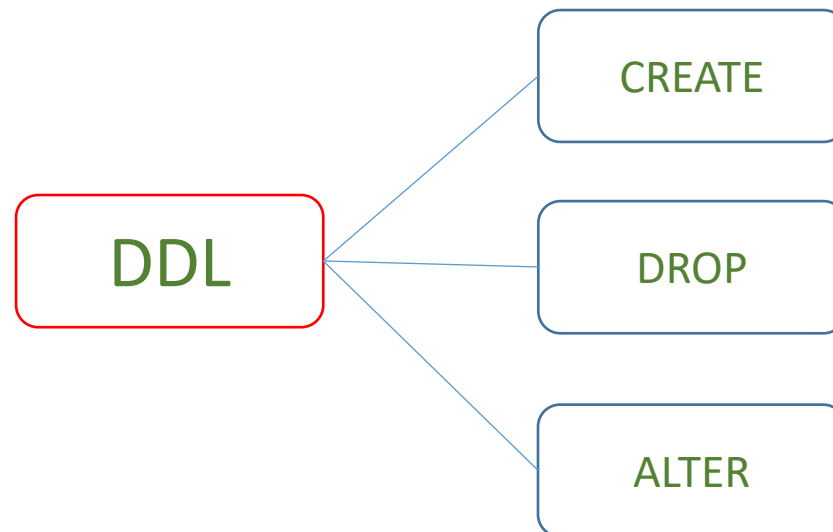
```
create table OrdenPedido
```

```
(  
  num_ordenPedido varchar(8) primary key not null,  
  fecha datetime not null,  
  cod_cli varchar (8) null,  
  nom_cli varchar(40)null,  
  cod_emp varchar(8) null,  
  cod_tipoPago varchar(8) null,  
  total int null  
)
```



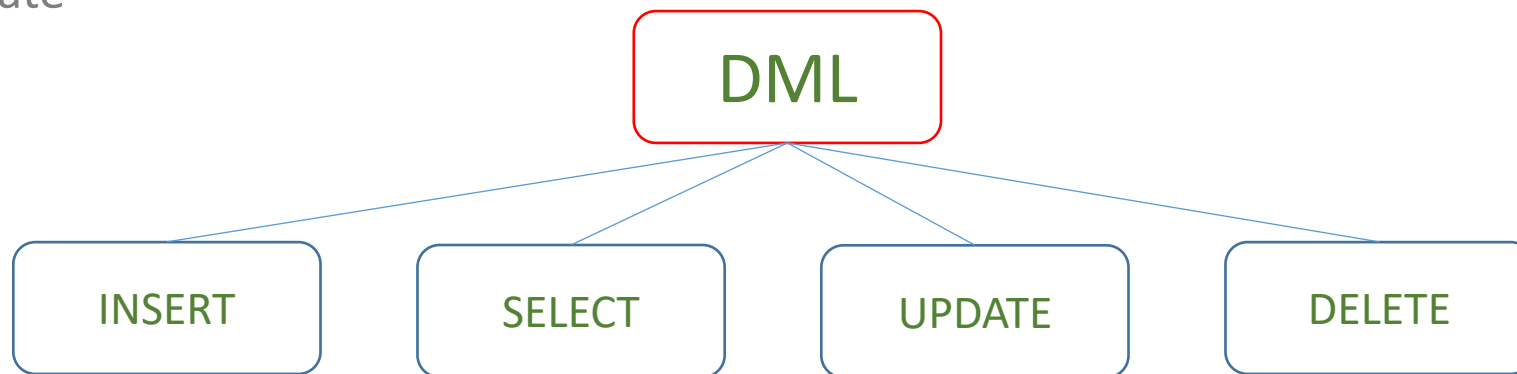
# DDL

El DDL es la parte del lenguaje SQL que realiza la función de definición de datos del SGBD. Fundamentalmente, se encarga de la creación, modificación y eliminación de los objetos de la base de datos. Por supuesto es el encargado de la creación de las tablas. Los elementos, llamados objetos, de la base de datos: tablas, vistas, columnas, índices,... se almacenan en el diccionario de datos. Por otro lado, muchos Sistemas Gestores de Bases de Datos aportan elementos para organizar estos objetos



# —• DML

es un idioma proporcionado por los sistemas gestores de bases de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o modificación de los datos contenidos en las Bases de Datos del Sistema Gestor de Bases de Datos. El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional .Sus elementos de manipulación son: Select, Insert, Delete y Update



INSERT :Una sentencia INSERT de SQL agrega uno o más registros a una (y sólo una) tabla en una base de datos relacional.

```
insert into Cientes values ('34984835','CLEMENTE','LIMA S/N','062','M',72884742,1072884742,3288428,997547215),  
('45346576','HARRY','LIMA S/N','065','M',84884342,1084884342,3266428,997547655),
```

SELECT :permite realizar consultas sobre los datos almacenados en la base de datos.

```
] SELECT ProductID, ProductName, CategoryID, UnitPrice  
FROM dbo.Products
```

DELETE :Una sentencia DELETE de SQL borra uno o más registros existentes en una tabla.

```
delete from EMPLEADO where codigo='002'
```

UPDATE :Una sentencia UPDATE de SQL es utilizada para modificar los valores de un conjunto de registros existentes en una tabla.

```
update CURSOS  
    set ID_PROFE = 2  
    where ID_CURSO = 5
```

## —• Crear aplicación de consola

El método `WriteLine()`: Este método es el que se usa para mostrar texto en la consola, el método escribe en la pantalla el valor que le pasemos como parámetro.

```
using System;

namespace ConsoleApplication2
{
    class Class1
    {
        [STAThread]
        static void Main(string[] args)
        {
            //Escribimos una cadena de caracteres.
            Console.WriteLine("Escribiendo una línea en la consola");
            //Escribimos un numero entero
            Console.WriteLine(23);
            //Escribimos una comparación lógica
            Console.WriteLine(3 > 1);
            Console.ReadLine();
        }
    }
}
```



El método `ReadLine()` Este método se usa para recoger la información que el usuario introduce cuando la aplicación así lo requiera. Cuando invocamos al método `Console.ReadLine()` el sistema queda en espera hasta que el usuario pulsa la tecla Intro.

```
//Declaramos una variable de tipo cadena de caracteres
string cadena;
//Mostramos un mensaje al usuario
Console.WriteLine("Por favor, introduzca su nombre:");
//Capturamos el dato introducido por el usuario
cadena = Console.ReadLine();
//Operamos con el dato
cadena = "El nombre introducido es: " + cadena;
//Mostramos la cadena
Console.WriteLine(cadena);
Console.ReadLine();
```





PREPÁRATE  
PARA SER EL  
**MEJOR**



+ **ENTREMIENTO  
EXPERIENCIA**



**BIENVENIDOS.**



***GRACIAS***  
***POR TU PARTICIPACIÓN***





Por favor, bríndanos tus comentarios  
y sugerencias para mejorar nuestros servicios.

