

# 4. Implementación

Ph.D Priscilla Jiménez P.

# Programación en la capa de datos

Capa de datos

ADO.NET

# Capa de Datos

- Proporciona los métodos necesarios para insertar, actualizar, borrar y consultar información (métodos CRUD - create, read, update, delete).
- La implementación puede acceder a diferentes fuentes de datos (base de datos relacionales, No SQL, XML, archivos planos, servicios web, entre otros).

# Capa de Datos en .NET

Representa una capa de acceso a datos que oculta la fuente y los detalles técnicos para recuperar los datos.



ADO.NET no depende de conexiones continuamente activas. Los datos se almacenan en memoria cache lo que permite trabajar sin conexión.



ADO.NET: Conjunto de clases pertenecientes a System.Data para acceso a los datos de un origen de datos.

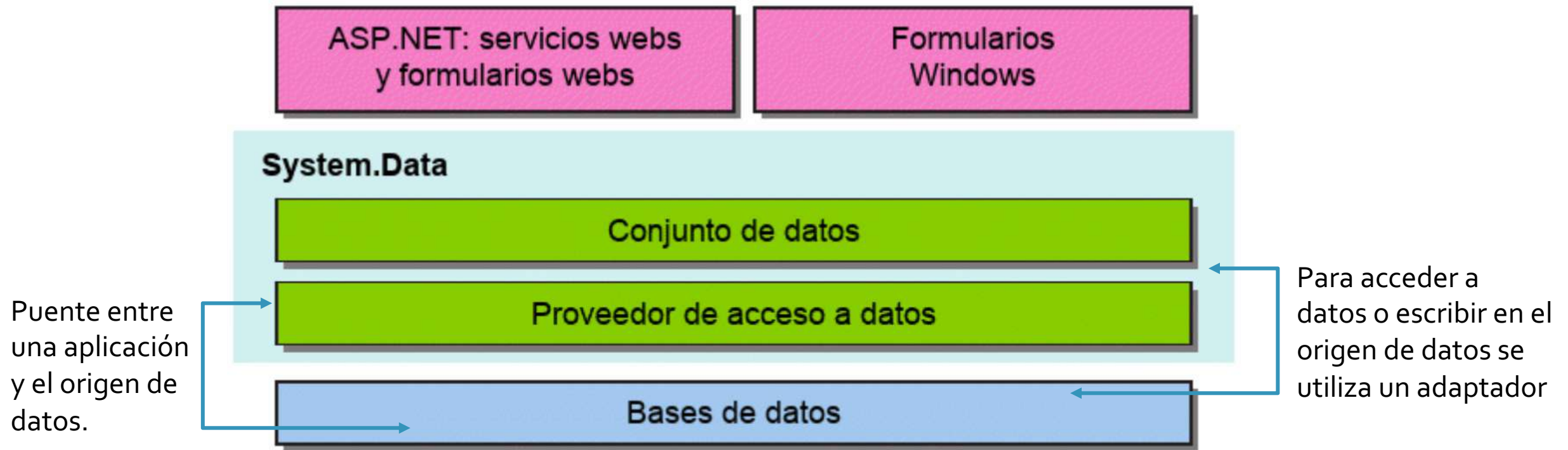
DataSet

Connection

Command

DataReader

DataAdapter



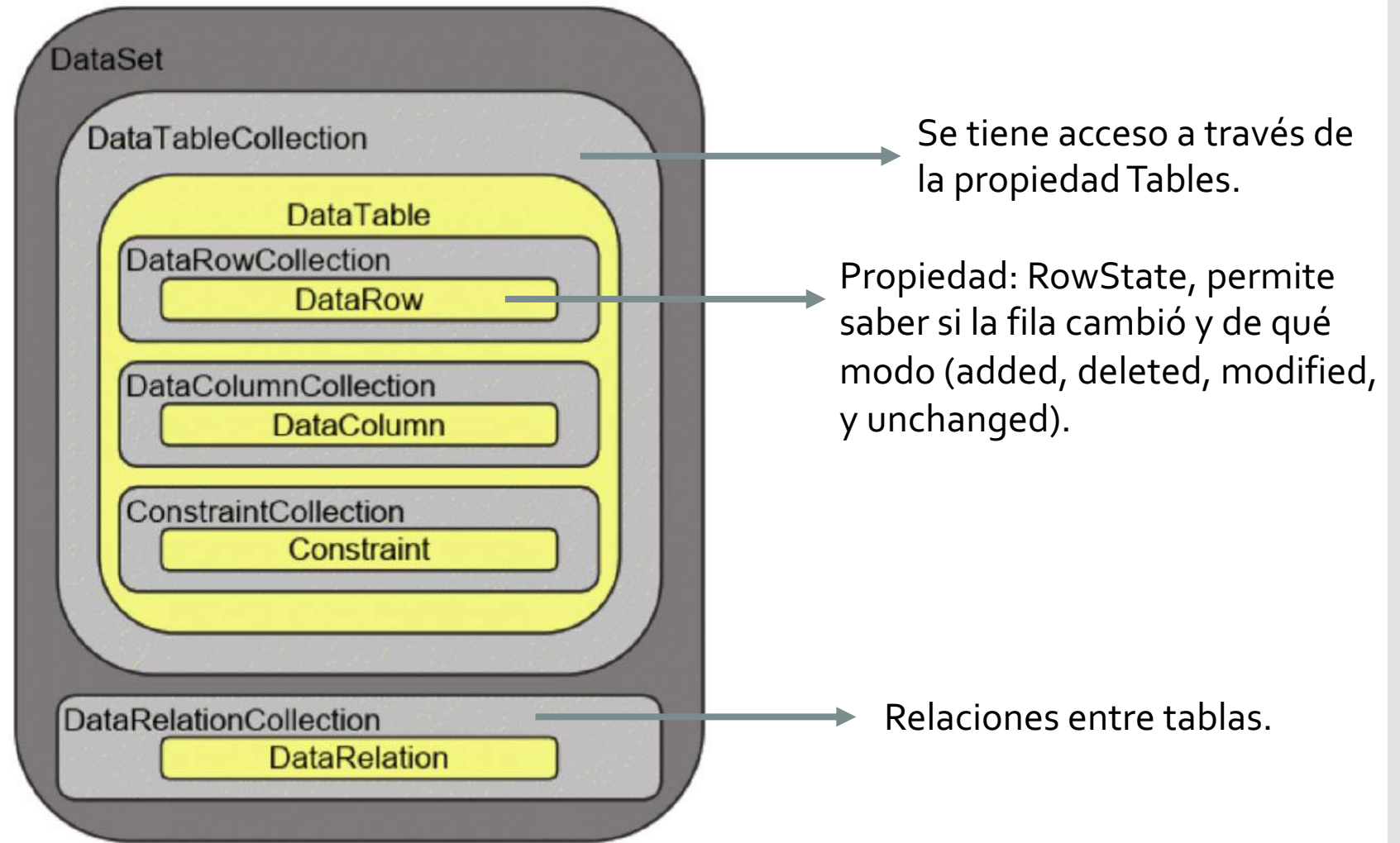
# ADO.NET

Las interacciones con la base de datos se realizan mediante:

- Un objeto de órdenes (Command) que encapsulan sentencias de SQL.
- Referencia a un procedimiento almacenado.

# ADO.NET

## Conjunto de Datos

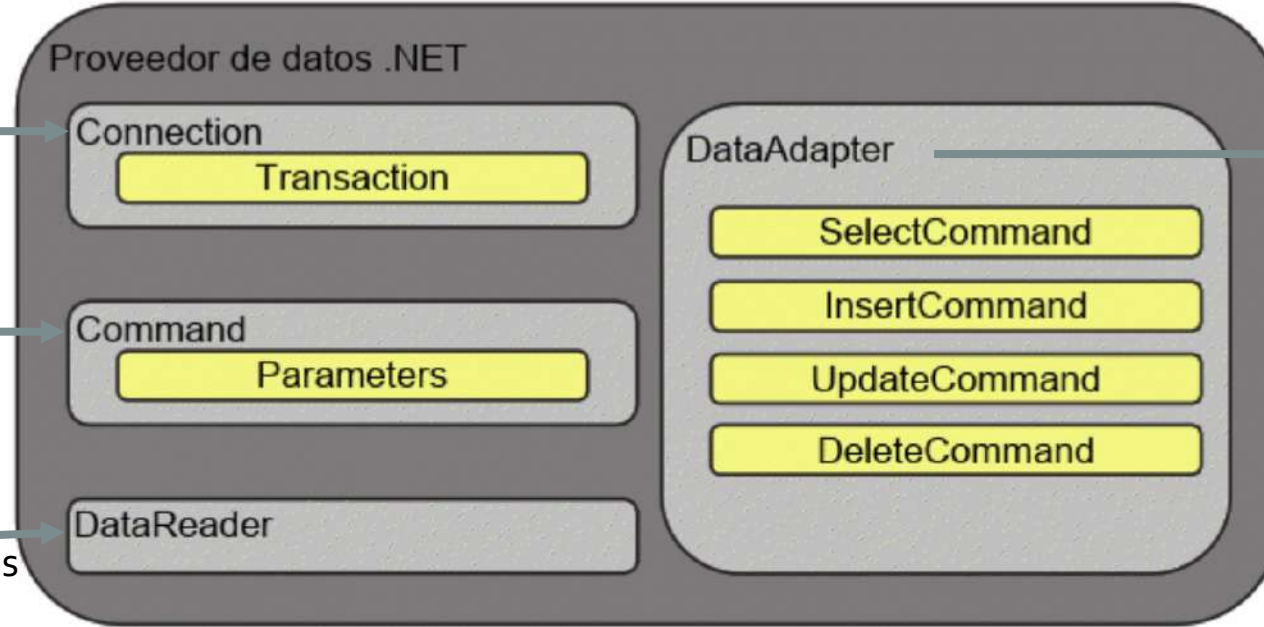


El componente central de la arquitectura sin conexión es la clase de objetos DataSet

Establece una conexión a un origen de datos determinado.

Ejecuta una orden SQL o un procedimiento almacenado en un origen de datos.

Proporciona una forma rápida de acceder a los datos recuperados después de una consulta a la BD.



Llena un DataSet y realiza las actualizaciones necesarias en el origen de datos.

## ADO.NET Proveedor de datos

Se utiliza para recuperar datos del origen como para actualizarlos.

.NET incluye los siguientes proveedores de datos: ODBC, OLE DB, Oracle, SQL Server. Se pueden encontrar en System.Data.Odbc, System.Data.OleDb, System.Data.OracleClient, System.Data.SqlClient.

Cada proveedor de datos contiene una implementación concreta de las clases Connection, Command, DataReader y DataAdapter.

- Para una conexión a Microsoft SQL Server se utiliza el objeto SqlConnection.
- Se necesita el espacio de nombres System.Data.SqlClient y System.Data.

`SqlConnection` conexion = `new SqlConnection(cadena-de-conexión);`

- La conexión debe abrirse lo más tarde posible.
- La conexión debe cerrarse lo antes posible.

## El Objeto Conexión - Connection

Otros objetos de conexión: OleDbConnection, OdbcConnection, OracleConnection.



Servidor	Servidor donde se encuentra la base de datos. Puede ser nombre_del_equipo, nombre_DNS, dirección IP, dirección IP y puerto (predeterminado 1433), ServidorSQL\Instancia
Base de datos	Nombre de la base de datos
Autenticación	Tipo de autenticación o credenciales como usuario y contraseña.

## La cadena de conexión

Especifica la información necesaria para crear una conexión.

Nombre = Valor separados por ;

"Data Source = .\sqlexpress; Initial Catalog = NOMBRE\_BD;  
Integrated Security = True"

"Server = NOMBRE\_DEL\_SERVIDOR; Database = NOMBRE\_BD;  
User name =\*\*\*\*; Password:\*\*\*\*\*"

# Objeto orden - Command

Se utiliza el objeto **Command** para ejecutar sentencias SQL y devolver resultados desde el origen de datos.

SqlCommand

OleDbCommand

OdbcCommand

OracleCommand



La propiedad **CommandText** contendrá la instrucción SQL. Para ejecutar la orden se utilizan los métodos:

**ExecuteNonQuery:** para update, insert y delete. Retorna # filas afectadas.

**ExecuteReader:** para select. Devuelve un objeto **DataReader**.



```
SqlCommand ordenSQL = new SqlCommand("SELECT * FROM contacto",conexion);
```

## Objeto Lector de Datos - SqlDataReader

- Es un objeto conectado, trabaja en segundo plano con un conjunto de datos.

```
conexion.Open();
```

```
SqlDataReader lector = ordenSQL.ExecuteReader()
```

```
while (lector.Read())
```

```
{
```

```
    Console.WriteLine(lector.GetString(0) + ",  
                        "+lector.GetString(1));
```

```
}
```

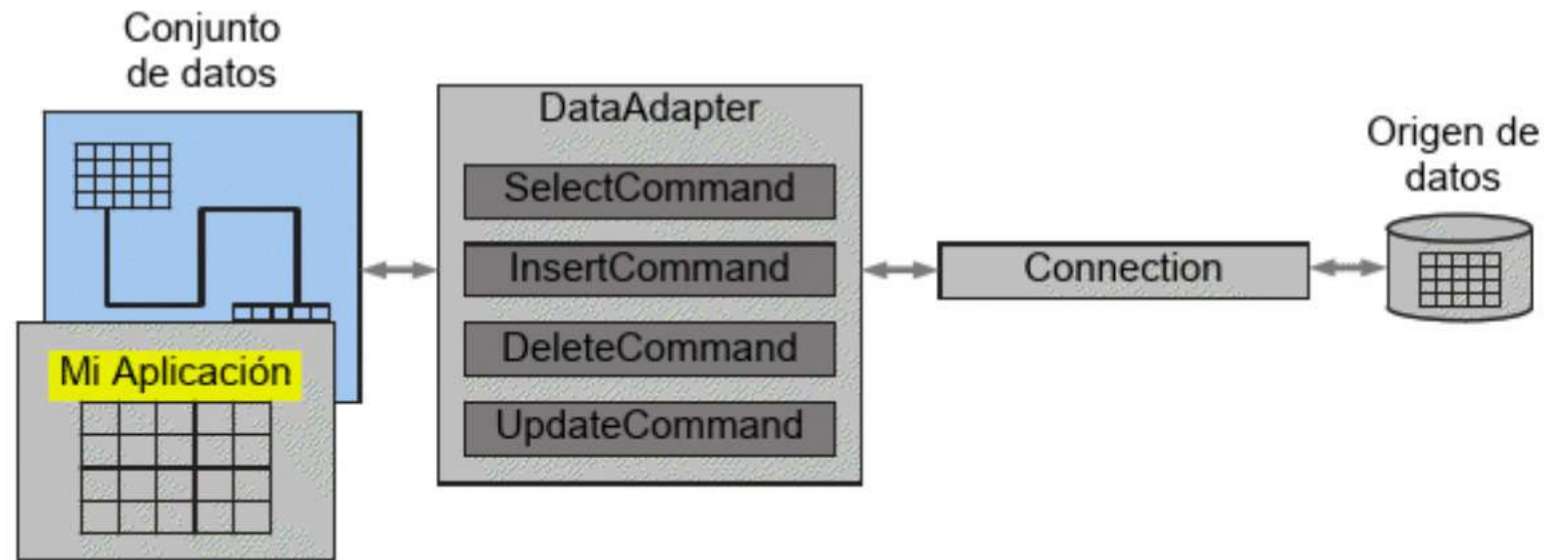
```
lector.Close();
```

```
conexion.Close();
```

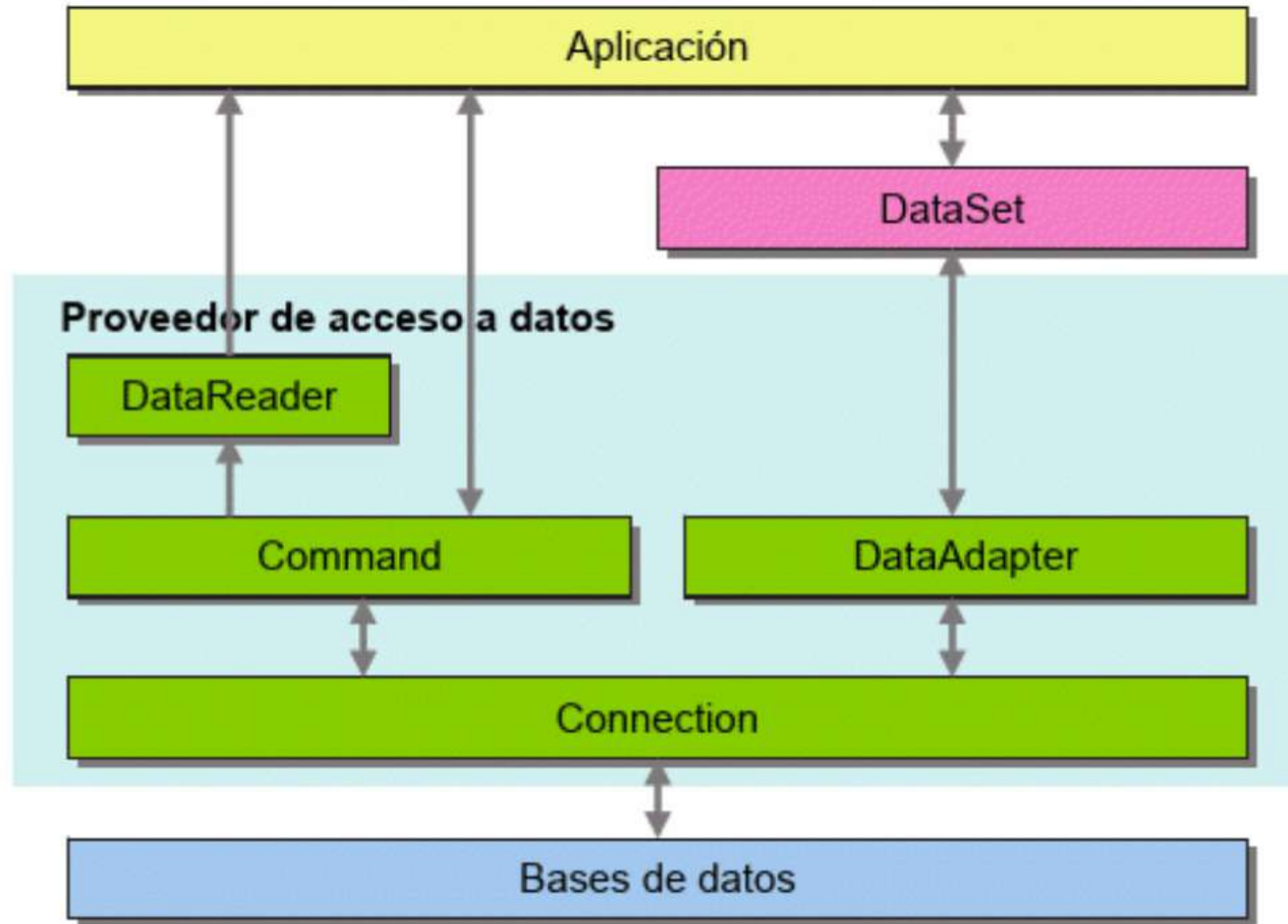
GetValue, obtiene el valor de la columna especificada en su formato nativo. GetInt32 obtiene el valor como entero.

# Objeto Adaptador Datos - SqlDataAdapter

Es un conjunto de objetos utilizado para intercambiar datos entre un origen de datos y un conjunto de datos (DataSet).



Método Fill y Update.

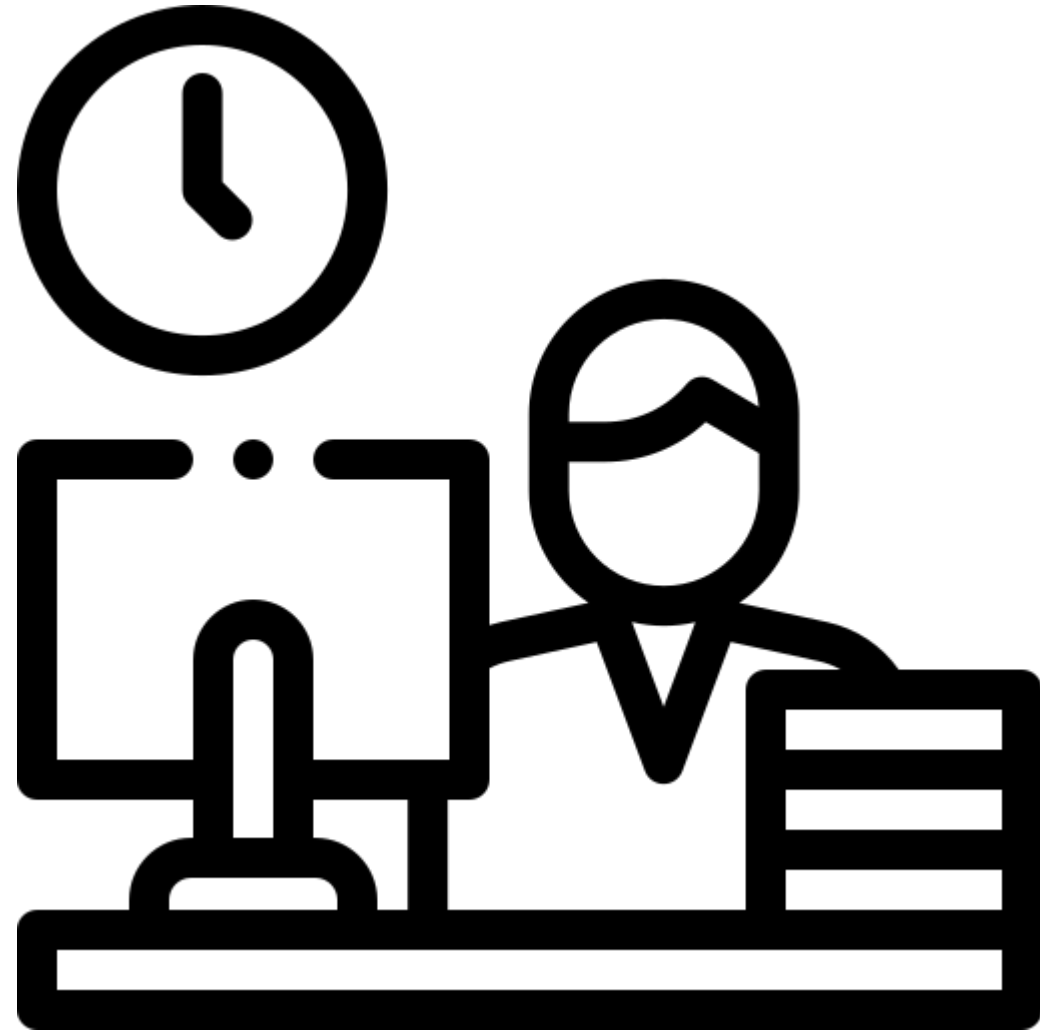


## Pasos fundamentales al trabajar con una BD (Microsoft SQL Server)

1. Crear la cadena de conexión: Servidor, Base de datos, Credenciales.
2. Se crea la conexión `SqlConnection` especificando la cadena de conexión.
3. Para establecer/abrir la conexión `.Open()`
4. Se realiza la operación/transacción. Escritura/Lectura a BD ->  
`SqlCommand & SQLExecuteReader / SqlDataAdapter / SQLExecuteNonQuery`
5. Después de una operación se deben cerrar los objetos relacionados a la BD con `.Dispose()` o `.Close()`

# Práctica 17

Programación en la capa de datos





## Bibliografía y Recursos útiles

1. Colmenar, Santos, Antonio. *Visual C#: interfaces gráficas y aplicaciones para Internet con WPF, WCF y Silverlight*, RA-MA Editorial, 2014. ProQuest Ebook Central, (Cap 6)  
<https://bvirtual.epn.edu.ec:2117/lib/epnsp/detail.action?docID=5759069>.
2. <https://docs.microsoft.com/es-es/dotnet/framework/data/adonet/connection-string-syntax>
3. <https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.sqldataadapter?view=netframework-4.8>
4. <https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.sqlcommand?view=netframework-4.8>





## Bibliografía y Recursos útiles

### 5. Videos:

- <https://www.youtube.com/watch?v=-It7ZCbRRfE>
- <https://www.youtube.com/watch?v=Uf3FpOd5oYA> min 15-52
- <https://www.youtube.com/watch?v=DbgeRW1jaOo&list=PLnWAzeXpgV4kjwtKztMiBYPo2ncAjl2QqD>
- <https://www.youtube.com/watch?v=IKG8JGchydQ>