

# **Desarrollador de Aplicaciones Web**

## **Programación Web III**

**1er Cuatrimestre (2013)**



**Departamento de Ingeniería e Investigaciones Tecnológicas**

## **ADO.NET**

**Ing. Gerardo Barbosa**  
**Ing. Mariano Juiz**  
**Ing. Matias Paz Wasiuchnik**  
**Pablo Nicolás Sanchez**

<https://piazza.com/configure-classes/spring2013/unlampw31c2013>

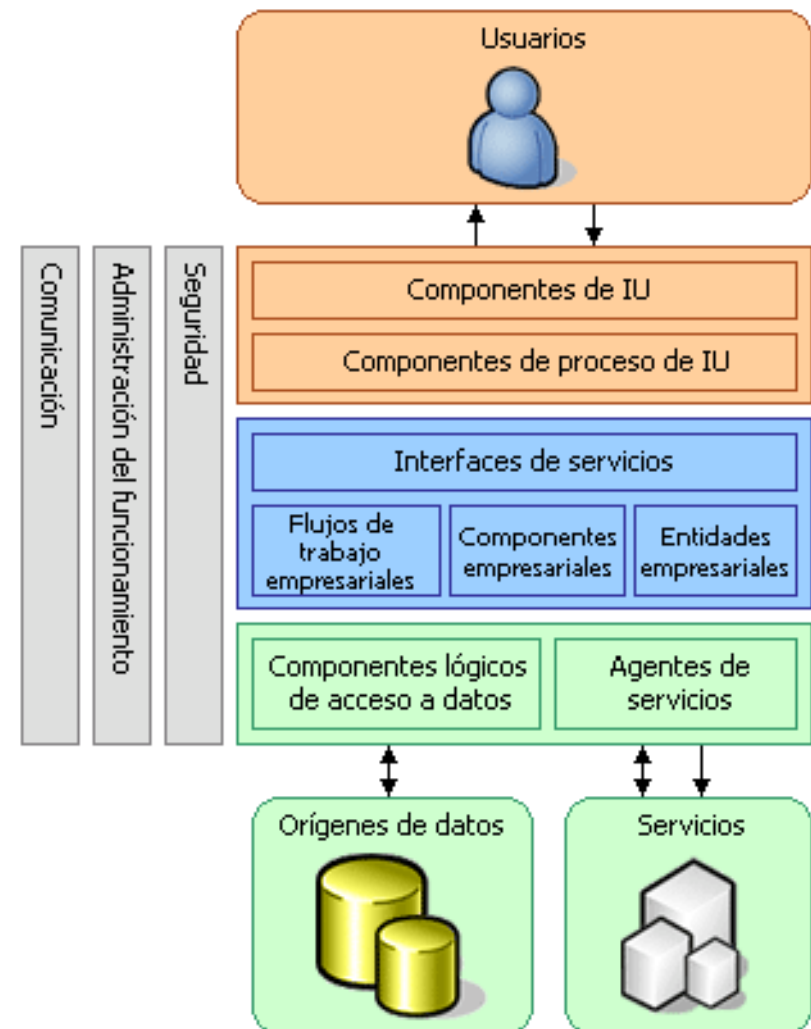
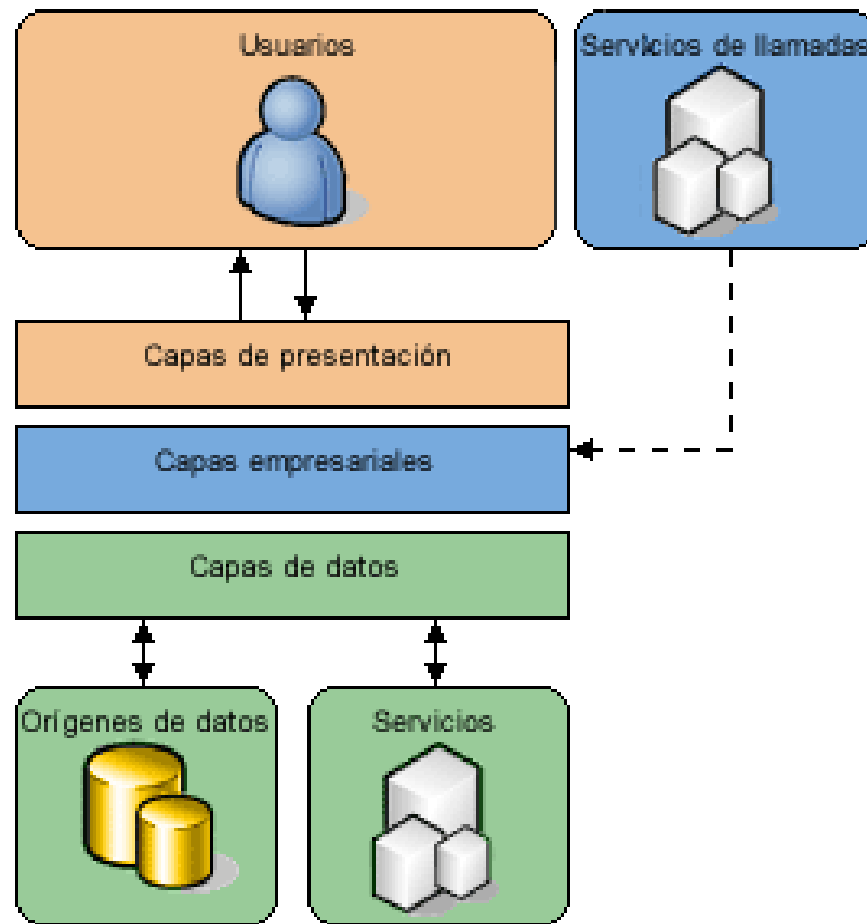
# Agenda

1. Introducción
2. Capas
3. Proveedores de Acceso a Base de Datos
4. Acceso a Base de Datos
5. Controles de Navegaci
6. XML

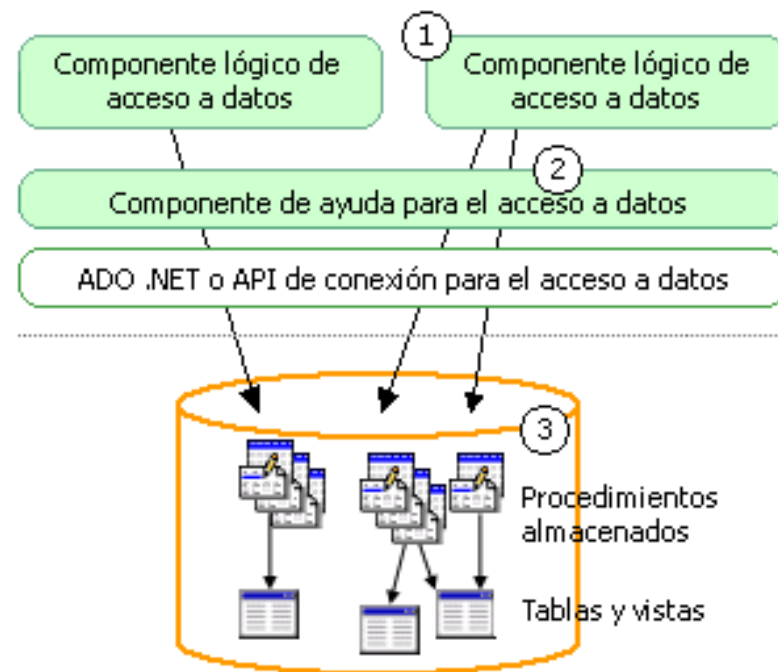
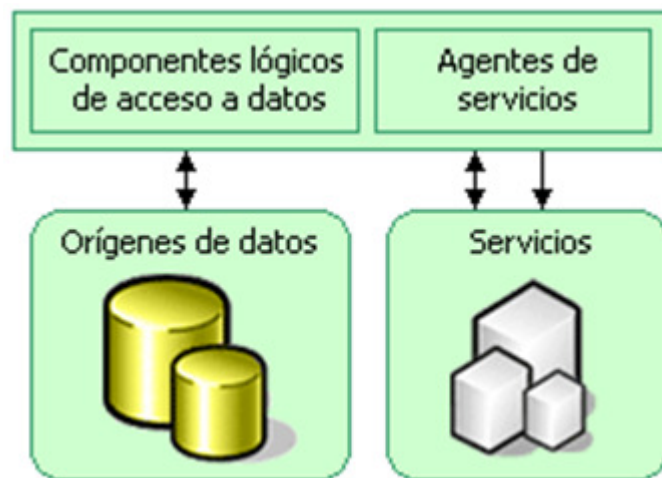
# ADO.NET - Introducción

- ADO.NET es un subconjunto de la .NET Framework Class Library, que contiene todas las funcionalidades necesarias para conectarse e interactuar con dos tipos de repositorios permanentes de información:
  - ❑ Bases de Datos, como Microsoft SQL Server (clases del namespace System.Data, que se encuentran compiladas en System.data.dll)
  - ❑ Archivos XML (clases del namespace System.XML, que se encuentran compiladas en System.Xml.dll)

# ADO.NET - Capas



# ADO.NET - Capas



# ADO.NET - Capas



## System.Data

**Common**

**OracleClient**

**Odbc**

**SqlClient**

**OleDb**

**SqlTypes**

# ADO.NET – Capas

## API Independiente

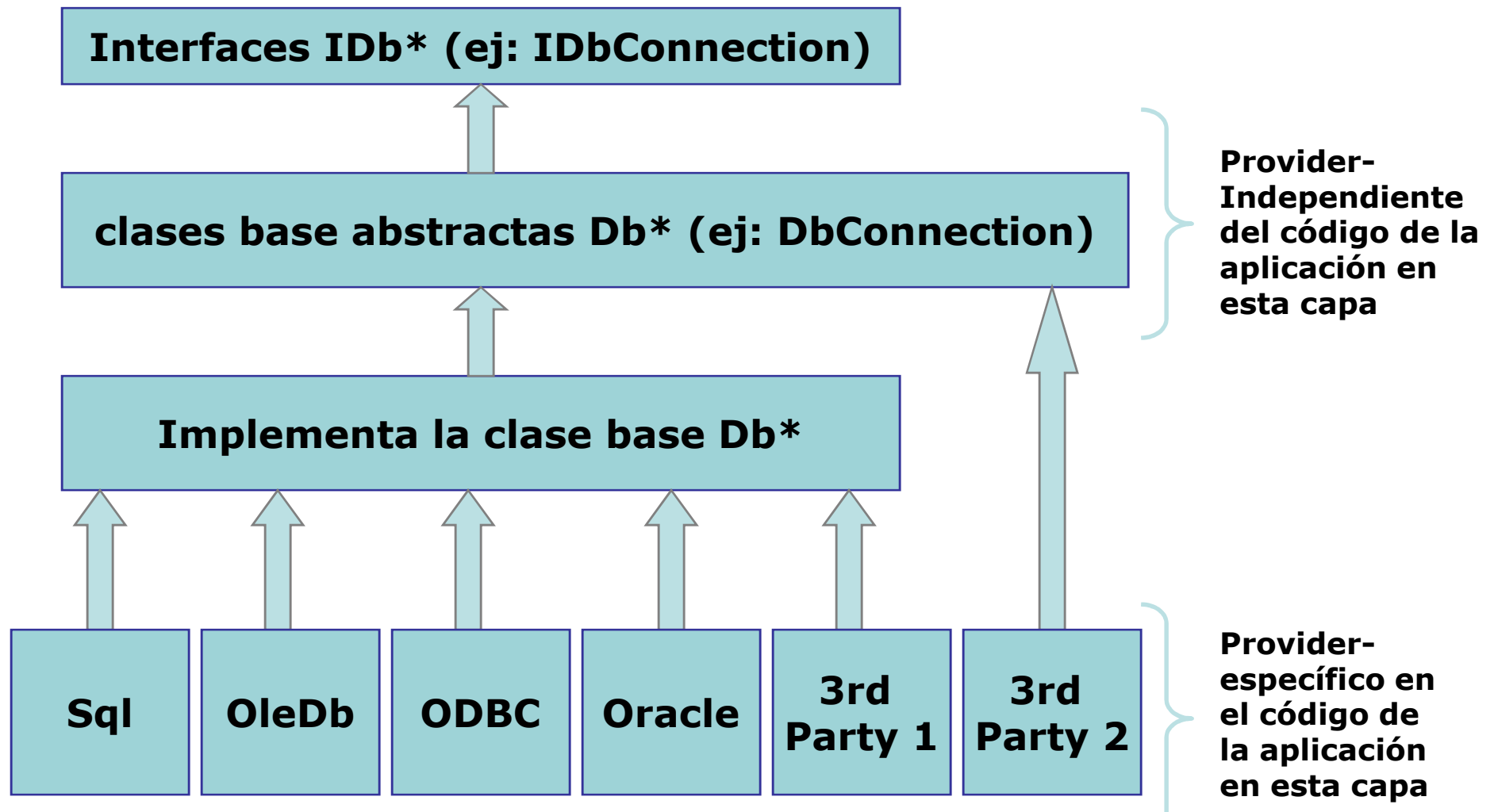


## Namespace System.Data.Common

<b>DbCommand</b>	<b>DbCommandBuilder</b>	<b>DbConnection</b>
<b>DataAdapter</b>	<b>DbDataAdapter</b>	<b>DbDataReader</b>
<b>DbParameter</b>	<b>DbParameterCollection</b>	<b>DbTransaction</b>
<b>DbProviderFactory</b>	<b>DbProviderFactories</b>	<b>DbException</b>

# ADO.NET – Capas

## API Independiente





# ADO.NET – Proveedores de Acceso a Datos

Representan conjuntos específicos de clases que permiten conectarse e interactuar con una base de datos, cada uno utilizando un protocolo particular.

El .NET Framework incluye cuatro proveedores de acceso a datos:

- Data Provider For SQL Server (7 o Superior) y MS Access: Al conectarse vía protocolos nativos de bajo nivel, provee la alternativa más performante para conexiones contra estos motores de bases de datos. Sus clases se encuentran en el namespace `System.Data.SqlClient`.
- Data Provider For OLE DB: es el proveedor de acceso a datos que permite interactuar via el protocolo estándar OLE DB con cualquier repositorio de datos que lo soporte. Sus clases se encuentran en el namespace `System.Data.OleDb`.

# ADO.NET – Proveedores de Acceso a Datos

- Data Provider For ODBC: es el proveedor de acceso a datos que permite interactuar via el protocolo estándar ODBC con cualquier repositorio de datos que lo soporte. Sus clases se encuentran en el namespace `System.Data.Odbc`.
- Data Porvider For Oracle: es el proveedor de acceso nativo a bases de datos Oracle, desarrollado por Microsoft utilizando las herramientas de conectividad de Oracle. Sus clases se encuentran en el namespace `System.Data.OracleClient`, y están compiladas en un assembly diferente al resto: `System.Data.OracleClient.dll`.

# ADO.NET – Proveedores de Acceso a Datos

ADO.NET provee una arquitectura extensible, posibilitando que terceras partes creen sus propios proveedores de acceso nativo para aplicaciones .NET.

Algunos ejemplos de esto son:

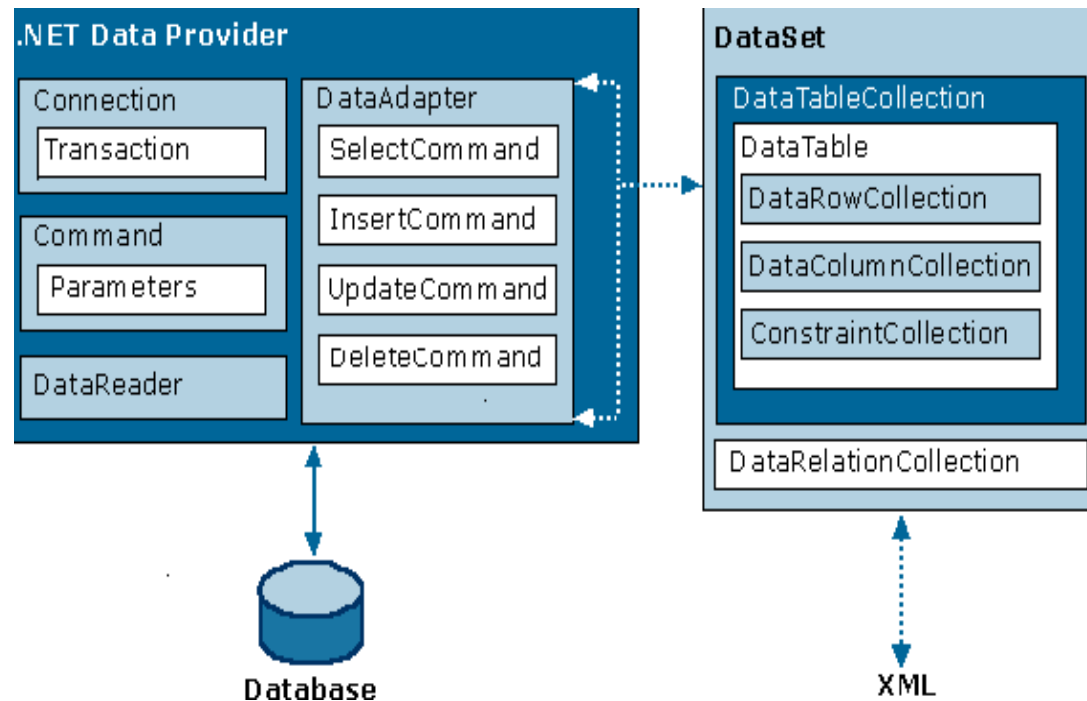
- Data Provider For DB2, desarrollado por IBM
- Oracle Data Provider For .NET, desarrollado por Oracle
- Providers de acceso nativo a bases de datos OpenSource, como MySQL y PostgreSQL

**Es importante volver a destacar que utilizando estos proveedores de acceso a datos cualquier aplicación .NET puede utilizar casi cualquier base de datos relacional existente en la actualidad como repositorio de información persistente** (esto incluye, además de MS SQL Server, a IBM DB2, Oracle, Sybase, Informix, TeraData, MySQL y PostgreSQL, entre otras).

# ADO.NET – Acceso a Datos

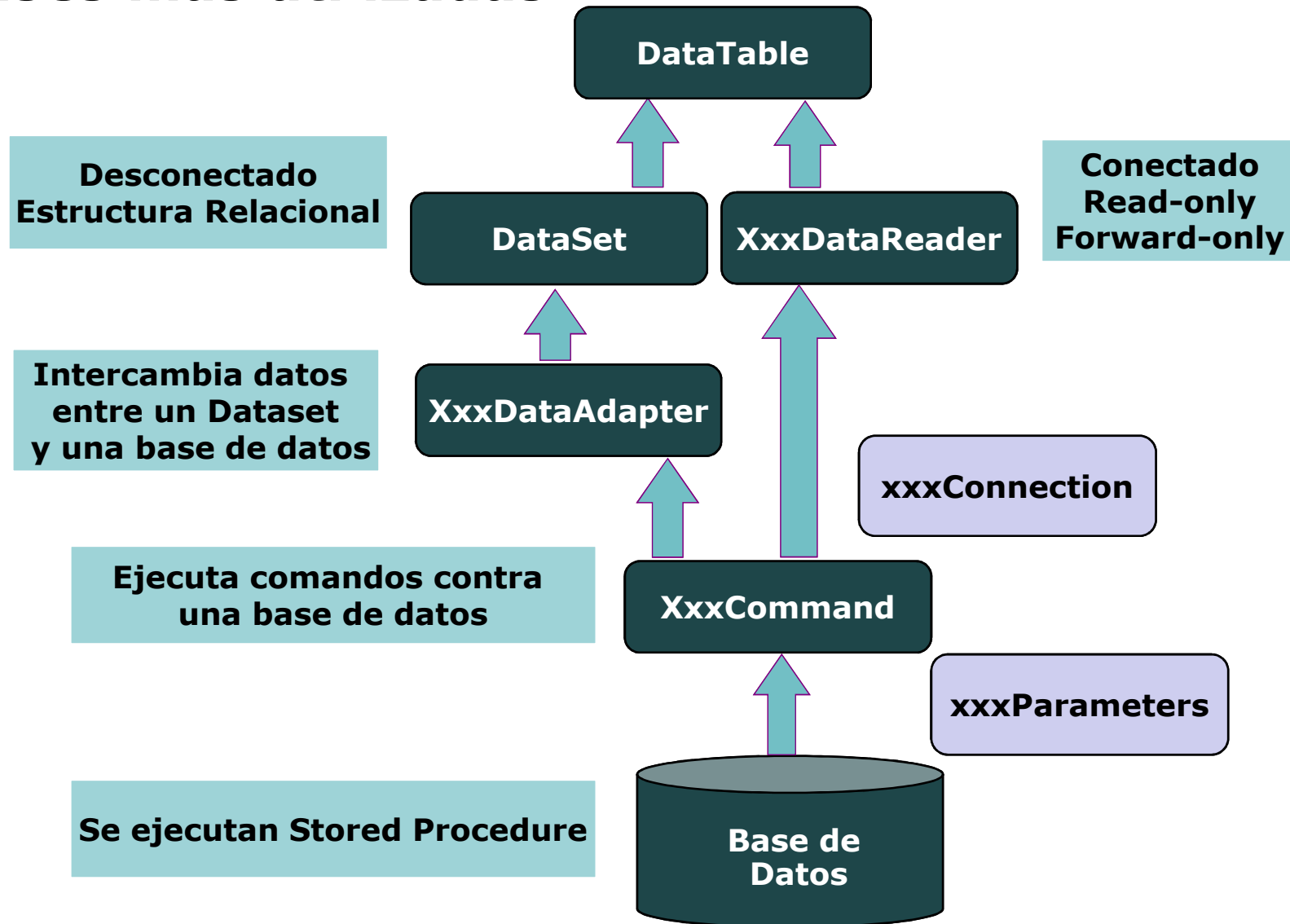
## Arquitectura

Está basada en el concepto de proveedores de acceso a datos, siendo un proveedor un conjunto de clases que permiten conectarse a una base de datos, ejecutar un comando sobre ella y tener acceso a los resultados de su ejecución, tanto de forma conectada como desconectada.



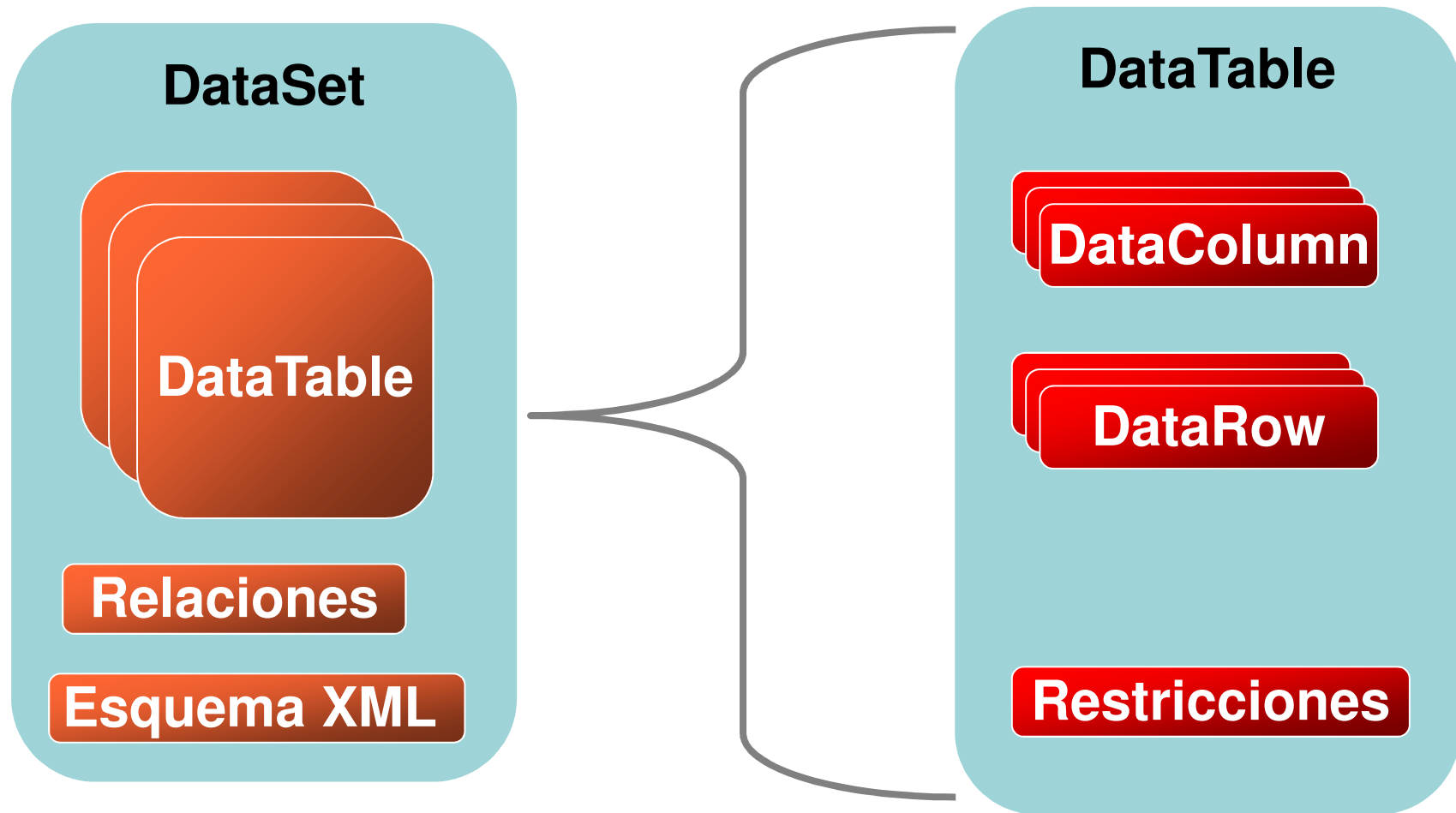
# ADO.NET – Acceso a Datos

## Clases más utilizadas



# ADO.NET - Acceso a Datos

## DataSet y DataTable



# ADO.NET- Acceso a Datos

## Escenario Conectado

Un entorno conectado es uno en el cual los usuarios están constantemente conectados a la fuente de datos

### Algunas Ventajas

- Al haber una única conexión por usuario, o incluso a veces por aplicación, establecida permanentemente, puede llegar a resultar más sencillo administrar la seguridad y el acceso al servidor de datos
- Lo mismo ocurre con el control de concurrencia: en un escenario donde múltiples usuarios se estuvieran conectando y desconectando permanentemente para realizar distintas acciones, este control sería más difícil de llevar
- Siempre la aplicación tiene acceso a los datos actualizados

### Algunas Desventajas

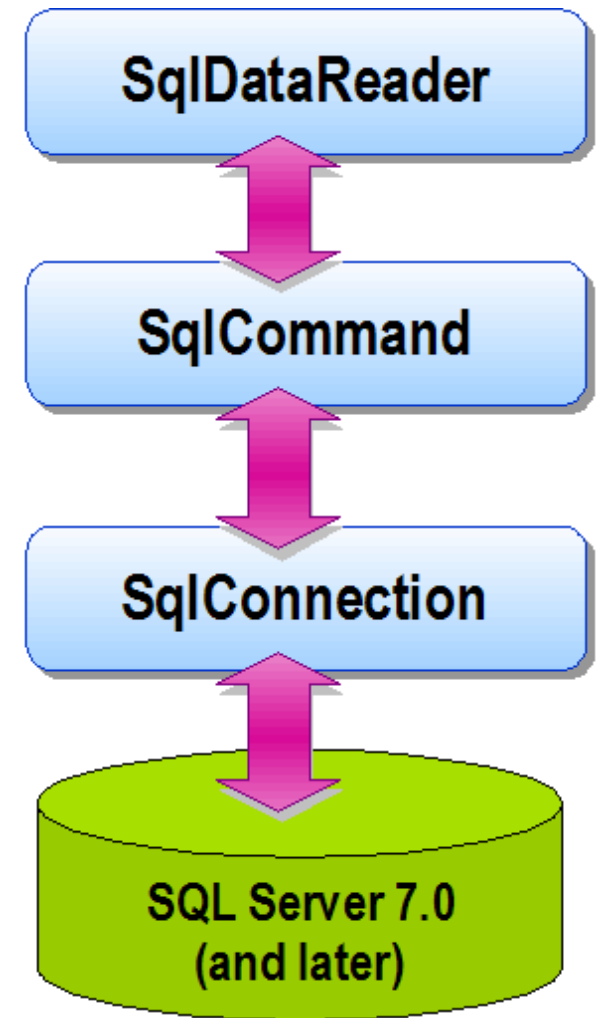
- Se requiere una conexión abierta todo el tiempo con el servidor de base de datos, lo cual consume recursos innecesariamente si no se la está utilizando
- La escalabilidad del acceso a los datos se ve limitada por la cantidad de conexiones establecidas simultáneamente contra el servidor de base de datos

# ADO.NET- Acceso a Datos

## Escenario Conectado

Los recursos se mantienen en el servidor hasta que la conexión se cierra

- 1) Abrir Conexión
- 2) Ejecutar Comando
- 3) Procesar Filas en DataReader
- 4) Cerrar Reader
- 5) Cerrar Conexión





# ADO.NET- Acceso a Datos

## Escenario Desconectado

En un entorno desconectado, una parte de los datos del repositorio central se copia y modifica en forma local, para luego sincronizarse con éste (ideal para dispositivos móviles)

### Algunas Ventajas

- La posibilidad de trabajar sobre los datos independientemente del resto de los usuarios de la aplicación
- Mayor escalabilidad en el acceso a datos y utilización más óptima de recursos del servidor, ya que se mantiene en un mínimo indispensable la cantidad y duración de conexiones abiertas
- Mayor performance, al trabajar con una copia local de los datos

### Algunas Desventajas

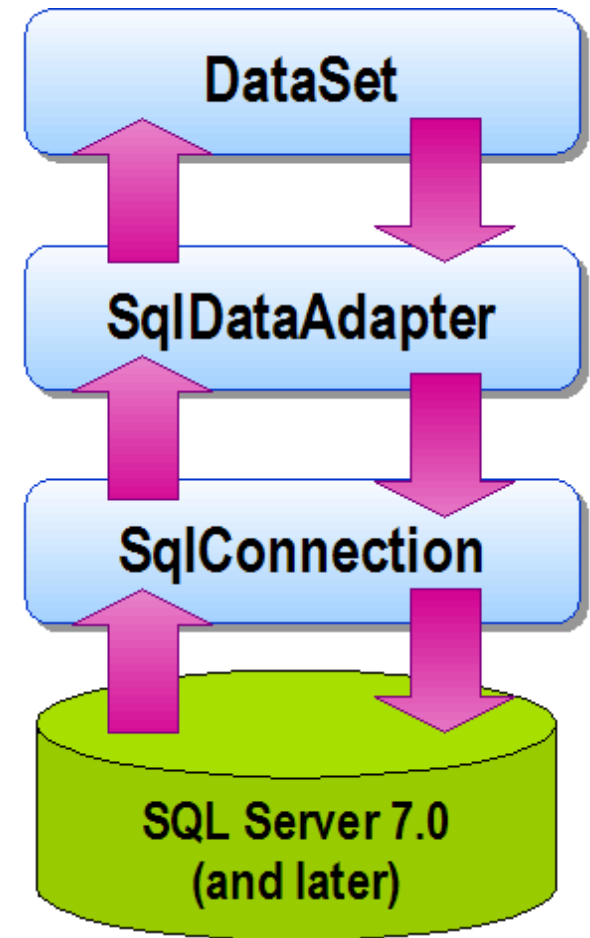
- Puede ocurrir que en un momento dado un usuario no esté accediendo a los datos más actualizados del repositorio central
- Al momento de sincronizar los cambios efectuados localmente contra el repositorio central pueden surgir conflictos, los cuales deben ser resueltos manualmente

# ADO.NET – Acceso a Datos

## Escenario Desconectado

Los recursos no se mantienen en el servidor mientras los datos se procesan

- 1) Abrir Conexión
- 2) Llenar DataSet mediante DataAdapter
- 3) Cerrar Conexión
- 4) Procesar DataSet
- 5) Abrir Conexión
- 6) Actualizar fuente de datos mediante DataAdapter
- 7) Cerrar Conexión



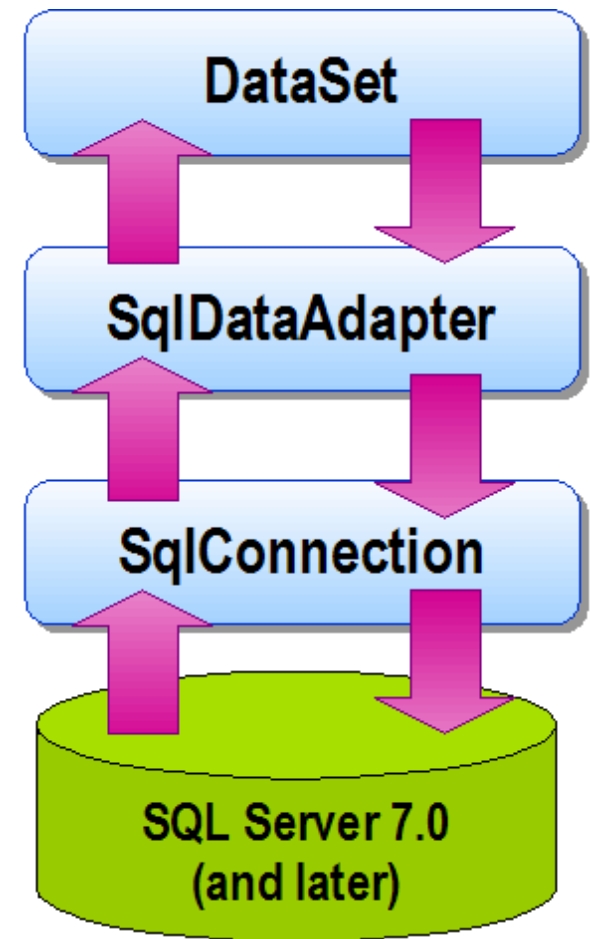
# ADO.NET – Acceso a Datos

## Escenario Desconectado

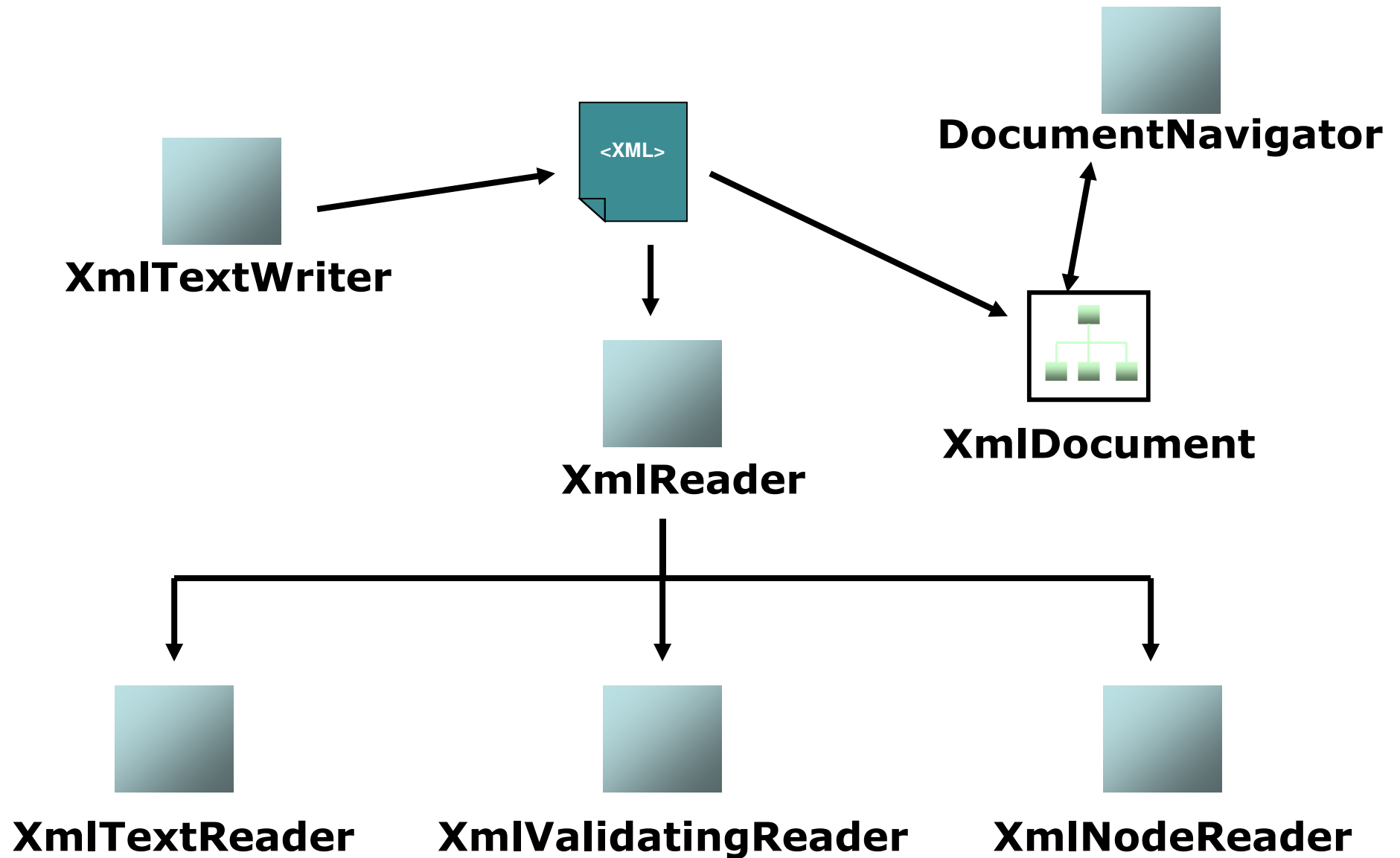
### SQLDataAdapter

Tiene estas cuatro propiedades:

- SelectCommand
- UpdateCommand
- InsertCommand
- DeleteCommand



# ADO.NET – XML



# **Desarrollador de Aplicaciones Web**

## **Programación Web III**

**1er Cuatrimestre (2013)**



**Departamento de Ingeniería e Investigaciones Tecnológicas**

# **Muchas gracias**

**Ing. Gerardo Barbosa  
Ing. Mariano Juiz  
Ing. Matias Paz Wasiuchnik  
Pablo Nicolás Sanchez**

**<https://piazza.com/configure-classes/spring2013/unlampw31c2013>**