

Microsoft
Dev Group
{Cobán}

Comparte | Aprende | Desarrolla



.NET Core

Entity Framework Core y LINQ

Acceso a Datos Moderno





Hola!

Geovani de León (Yova)

Ingeniero de Software en PensoTec

Docente Universitario –CUNOR–

<https://github.com/yovafree>

<https://yovadeleon.dev>

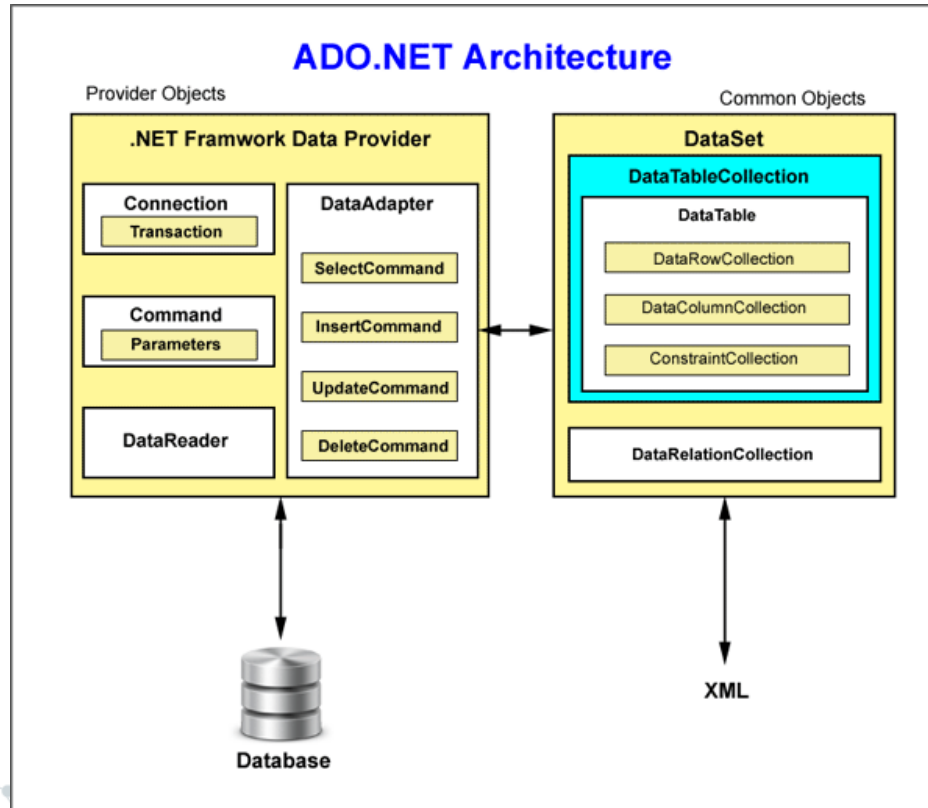
Agenda

- ◎ ADO.NET
- ◎ ORM
- ◎ Entity Framework Core
- ◎ Demo 1 - Instalación de Entity Framework
- ◎ Código Generado – Scaffolding
- ◎ LINQ
- ◎ Demo 2 - CRUD

ADO.NET

- ◎ Conjunto de componentes de software para acceder a datos y a servicios de datos.
- ◎ Es utilizado comúnmente por desarrolladores .NET para acceder y manipular datos en Sistemas de Bases de Datos Relacionales y fuentes no relacionadas.

Arquitectura de ADO.NET



ORM – Object Relational Mapping

- © Un *ORM* es un modelo de programación que permite mapear las estructuras de una base de datos relacional (*SQL Server*, *Oracle*, *MySQL*, etc.), sobre una estructura lógica de entidades con el objeto de simplificar y acelerar el desarrollo de nuestras aplicaciones.

Características de un ORM

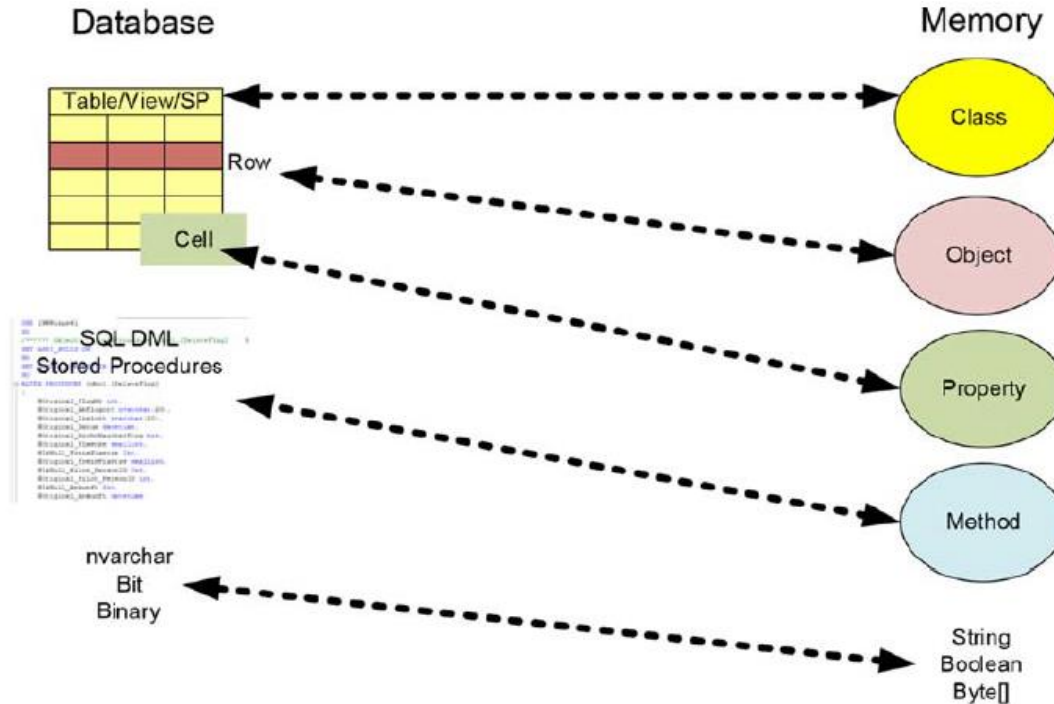
- ◎ Mapeo de la estructura virtual definida de una base de datos u origen de datos.
- ◎ Permite realizar operaciones CRUD (Create, Read, Update, Delete).
- ◎ Evita utilizar directamente instrucciones SQL
- ◎ Manejo automático transaccional (Commit, Rollback)
- ◎ Manejo automático transaccional distribuido

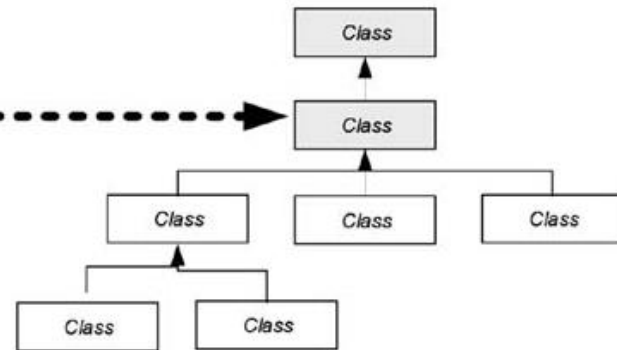
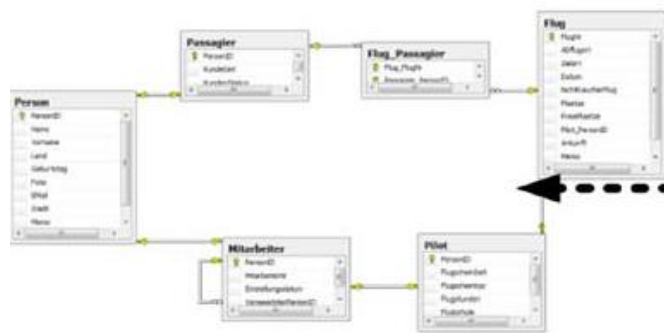
ORM en el Mundo de .NET

- ◎ NHibernate
- ◎ Telerik Data Access
- ◎ Genome
- ◎ Wilson
- ◎ SubSonic
- ◎ OBJ.NET
- ◎ Dapper
- ◎ Entity Framework

Entity Framework Core (2017, *La mayoría de productos ORM se centraron en esta implementación*)

¿Cómo sucede el mapeo?

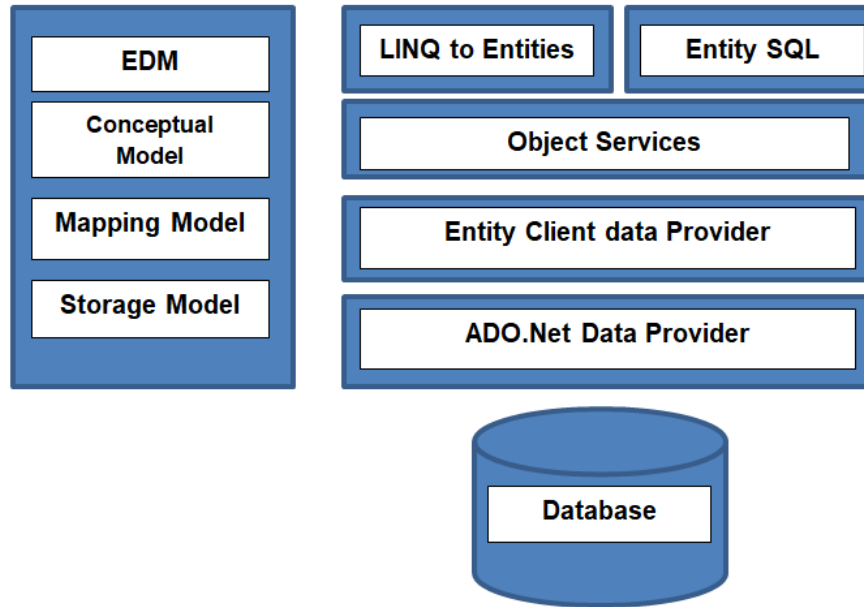




Entity Framework

- ◎ Es un ORM que permite a los desarrolladores de .NET trabajar con datos relacionales utilizando objetos específicos del dominio.
- ◎ Elimina la necesidad de la mayoría del código de acceso a datos que los desarrolladores generalmente necesitan escribir.
- ◎ Posee un modelo conceptual Entity Data Model (modelo Entidad-Relación) es mapeado a un modelo de esquema de datastore.

Componentes de Entity Framework



Entity Framework Architecture

Entity Framework Core

- ◎ Es la nueva versión de Entity Framework luego de la 6.x.
- ◎ Es una versión opensource ligera y multiplataforma de la tecnología de Entity Framework.

Soporte de Entity Framework Core

Application Types	<u>ASP.NET Core Applications</u> Web, API, Console, etc.	<u>.NET 4.5+ Applications</u> Console, WinForm, WPF, ASP.NET	Devices + IoT, Mobile, PC, Xbox, Surface Hub	<u>Mobile Application</u> Android, iOS, Windows
EF Core	EF Core	EF Core	EF Core	EF Core
Framework	.NET Core	.NET 4.5+	UWP	Xamarin
OS	Windows, Mac, Linux	Windows	Windows 10	Mobile

© EntityFrameworkTutorial.net

Bases de Datos Soportadas

Database	Company/Price	URL
Microsoft SQL Server	Microsoft/free	www.nuget.org/packages/Microsoft.EntityFrameworkCore.SqlServer
Microsoft SQL Server Compact 3.5	Microsoft/free	www.nuget.org/packages/Microsoft.EntityFrameworkCore.SqlServerCompact35
Microsoft SQL Server Compact 4.0	Microsoft/free	www.nuget.org/packages/Microsoft.EntityFrameworkCore.SqlServerCompact40
SQLite	Microsoft/free	www.nuget.org/packages/Microsoft.EntityFrameworkCore.sqlite
In Memory	Microsoft/free	www.nuget.org/packages/Microsoft.EntityFrameworkCore.InMemory
MySQL	Oracle/free	www.nuget.org/packages/MySQL.Data.EntityFrameworkCore
PostgreSQL	Open source team npgsql.org/free	www.nuget.org/packages/Npgsql.EntityFrameworkCore.PostgreSQL
DB2	IBM/free	www.nuget.org/packages/EntityFrameworkCore.IBMDaServer

MySQL, Oracle,
PostgreSQL, SQLite, DB2,
Salesforce, Dynamics CRM,
SugarCRM, Zoho CRM,
QuickBooks, FreshBooks,
MailChimp, ExactTarget,
Bigcommerce, Magento

Devart/\$99 to \$299
per driver type www.devart.com/purchase.html#dotConnect

Consideraciones

- ◎ Se eliminó el proceso de mapeo mediante Database First (Solo en CLI) y Model First. Es posible realizarlo a través de ingeniería reversa y utilidades de la comunidad.
- ◎ Se eliminó soporte a Entity Data Model (EDM) y la representación en XML (EDMX).
- ◎ Se eliminó la claseObjectContext y se deja como única clase base DbContext.
- ◎ Se eliminó la clase base EntityObject. Las clases de entidad son desde ahora objetos CLR (POCOs)

Consideraciones

- ◎ Se tiene soporte para acceso a tablas no a vistas de base de datos. Se puede acceder a datos por medio de vistas si se utiliza SQLQuery.
- ◎ Se debe tener cuidado en algunas implementaciones ya que pueden ejecutarse en RAM. Eje: Operador Group By.

Proceso de mapeo – Code First

- ⦿ Code First nos situamos en definir nuestras clases mediante código, y EF se encarga de generar la base de datos y todo lo necesario para encajar las clases en ellas.
- ⦿ Se trata de un enfoque muy orientado al programador. Es para gente a la que le gusta escribir código y tener control total sobre cómo se comportan sus clases.

Ventajas

- ⦿ Máximo control sobre el código del modelo de datos en C#, ya que son clases que construyes desde cero.
- ⦿ Ofrece control sobre las colecciones de objetos que quieres que se carguen de modo "perezoso" o lazy loading (es decir, a medida que se vayan necesitando).
- ⦿ El código es el que manda y el "tooling" se encarga de generar la base de datos en función de tu código.

Desventajas o inconvenientes

- ⦿ Se debe dominar bastante más EF, conceptualización de las etiquetas y como mapear tipos de datos.
- ⦿ Cualquier cosa que se necesite persistir u obtener de la base de datos la tiene que implementar en código C# y LINQ.
- ⦿ Si la base de datos es muy grande y con muchas tablas, la gestión de la base de datos puede complicarse.

Demo 1



Instalación de Entity Framework



◎ dotnet new console -n demo1

◎ dotnet run

◎ dotnet add package Microsoft.EntityFrameworkCore.SqlServer

◎ dotnet add package Microsoft.EntityFrameworkCore.Design

◎ dotnet add package Microsoft.VisualStudio.Web.CodeGeneration.Design



Código Generado - Scaffolding

- © Podemos generar algunas de las partes de estos proyectos, como controladores y vistas. A esto le llamamos scaffolding: a la generación de areas, controladores, vistas y páginas con código predefinido.

Demo 2



CRUD - Scaffolding

Utilidad para Código Generado

© dotnet tool install --global dotnet-aspnet-codegenerator

LINQ

- ◎ Operadores de conversión
 - ToList(), ToArray(), ToLookup(), ToDictionary(), Single(), SingleOrDefault(), First(), FirstOrDefault()
- ◎ Operadores de Agregación
- ◎ Condiciones

Siguientes de sesiones

- ◎ Sesión #4: Sync, Async y await – Manejo de Excepciones
- ◎ Sesión #5: .NET Core Web API
- ◎ Sesión #6: SignalR en .NET Core – Parte 1 – Web Chat
- ◎ Sesión #7: SignalR en .NET Core – Parte 2 – Web Chat



Gracias!

¿Preguntas?

Puedes contactarme mediante:

@yovafree

erwingeo1021@gmail.com

<https://github.com/yovafree>

