

Entity Framework

ISI Quevedo Fabricio

¿Qué es un ORM?

- ▶ Object-Relational mapping, **mapeo de objeto-relacional**, es un modelo de programación que consiste en la transformación de las tablas de una base de datos (Modelo Relacional), en una serie de entidades u objetos (Programación orientada a objetos) que simplifican la persistencia de datos.
- ▶ El principal problema es que la mayor parte de las aplicaciones están diseñadas para usar la Orientación a Objetos (POO), mientras que las bases de datos más extendidas son del tipo relacional.

Ventajas

- ▶ Facilidad y velocidad de uso
- ▶ Abstracción de la base de datos usada.
- ▶ Seguridad de la capa de acceso a datos contra ataques.

Desventajas

- ▶ En entornos con gran carga poner una capa más en el proceso puede mermar el rendimiento.
- ▶ Aprender el nuevo lenguaje del ORM.

Entity Framework

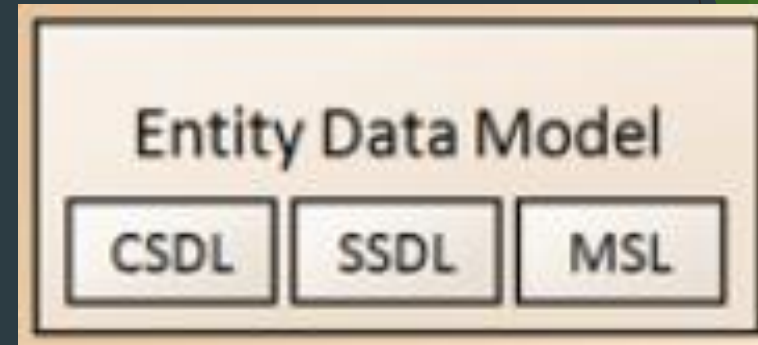
- ▶ ADO.NET Entity Framework es un framework ORM para la plataforma Microsoft .NET.
- ▶ EF puede generar los comandos en la BD para leer o escribir datos en la misma y ejecutarlo.
- ▶ Puede expresar consultas del dominio de objetos usando LINQ to entities
- ▶ EF puede ejecutar consultas en la base de datos y materializar los resultados en instancias de objetos del dominio de la aplicación

Arquitectura

- ▶ The ADO.NET Entity Framework tiene capas que abstraen el esquema relacional de la base de datos y presenta un modelo conceptual.
- ▶ **Data Source** : la capa inferior son los datos que pueden ser persistidos en una o varias bases de datos.
- ▶ **Data Providers** : los datos pueden ser accedidos por ADO.NET data provider. Data providers soportados Oracle, MySQL, DB2, Firebird, Sybase, VistaDB, SQLite, ...



Entity Data Model (EDM)



- ▶ Consists of 3 parts:
- ▶ **Conceptual schema definition language (CSDL)** : Declara y define entidades, asociaciones, herencia... Entity classes son generadas por este schema.
- ▶ **Store schema definition language (SSDL)** : Metadata describe el storage container (=database) que persiste los datos.
- ▶ **Mapping specification language (MSL)** : Mapea las entidades en el modelo CSDL a tablas descritas en el SSDL.

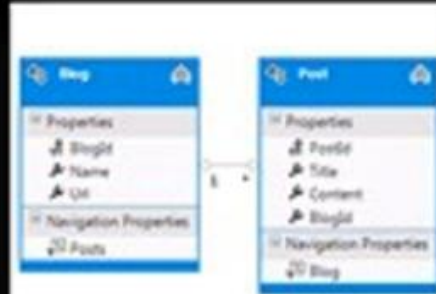
Arquitectura

- ▶ **Entity Client** : EntityClient is an ADO.NET managed provider that supports accessing data described in an Entity Data Model. It is similar to SQLClient, OracleClient and others. It provides several components like EntityCommand, EntityConnection and EntityTransaction.
- ▶ **Object Services** : This component enables you to query, insert, update, and delete data, expressed as strongly-typed CLR objects that are instances of entity types. Object Services supports both Entity SQL and LINQ to Entities queries.
- ▶ **Entity SQL (ESQL)** : Entity SQL is a derivative of Transact-SQL, designed to query and manipulate entities defined in the Entity Data Model. It supports inheritance and associations. Both Object Services components and Entity Client components can execute Entity SQL statements.
- ▶ **LINQ to Entities** : This is a strong-typed query language for querying against entities defined in the Entity Data Model.



Workflows de Desarrollo

Entity Framework Development Workflows



```
public class Blog
{
    public int BlogId { get; set; }
    public string Name { get; set; }

    public IList<Post> Posts { get; set; }
}

public class Post
{
    public int PostId { get; set; }
    public string Title { get; set; }
    public string Content { get; set; }

    public int BlogId { get; set; }
    public Blog Blog { get; set; }
}
```



New
Database

Model First

- Create model in designer
- Database created from model
- Classes auto-generated from model

Code First (New Database)

- Define classes & mapping in code
- Database created from model
- Use Migrations to evolve database



Existing
Database

Database First

- Reverse engineer model in designer
- Classes auto-generated from model

Code First (Existing Database)

- Define classes & mapping in code
- Reverse engineer tools available