Entity Framework Core

Acceso a Datos

Entity Framework

- Tecnología de acceso a datos en .NET Core.
  - ADO.NET.
  - Entity Framework 6.x.
- ORM.
- Modelo.
  - Code First.
    - DbContext, DbSet.
  - Scaffold.
  - Anotaciones.
  - Fluent API.

Entity Framework (EF) Core es una versión ligera y extensible de la popular tecnología de acceso a datos de Entity Framework.

EF Core es un mapeador objeto-relacional (ORM) que permite a los desarrolladores de .NET trabajar con una base de datos utilizando objetos .NET. Se elimina la necesidad de que la mayor parte del código de acceso a datos que los desarrolladores en general tienen que escribir.

EF Core soporta varios motores de bases de datos:

- Microsoft SQL Server
- SQLite
- Npgsql (PostgreSQL)
- Microsoft SQL Server Compact Edition
- IBM Data Servers
- Devart (MySQL, Oracle, PostgreSQL, SQLite, DB2, SQL Server, and more)
- En proceso: InMemory. MySQL, Oracle.

Se propone como tecnología de acceso a datos no relacionales como las conocidas como NoSQL.

Es una nueva versión del Entity Framework por lo que no soporta actualmente todas la facilidades que existen en el Entity Framework 6.x. En el siguiente enlace se puede encontrar una comparación de las características del EF 6.x y el EF Core 1.0, https://docs.efproject.net/en/latest/efcore-vs-ef6/features.html#side-by-side-comparison.

Entity Framework utiliza un conjunto de convenciones para construir un modelo basado en la forma de clases de entidad (modelos, estructuras de datos). Se puede especificar una configuración adicional para complementar y anular lo que fue descubierto por convención en el proceso de Scaffold.

Con el Entity Framework Core 1.0 el acceso a los datos se efectúa por medio de un modelo. Un modelo se compone de clases de entidad y una clase de contexto (derivada de DbContext) que representa una sesión con la base de datos, lo que permite consultar y guardar los datos.

Se puede generar un modelo utilizando el Scaffolding considerando los siguientes pasos generales:

- Crear una aplicación de consola cuyo target sea el .NET Core 1.0.
- Instalar los paquetes en el proyecto de la aplicación recién creada:

Microsoft.EntityFrameworkCore.Tools -Pre
Microsoft.EntityFrameworkCore.Design -Pre
Microsoft.EntityFrameworkCore.SqlServer
Microsoft.EntityFrameworkCore.SqlServer.Design

*: Corresponde al paquete en versión prerelease.

. Agregar la sección tools tal como se presenta a continuación al archivo project.json.

"tools": {   "Microsoft.EntityFrameworkCore.Tools": "1.0.0-preview2-final" }

- Ejecutar dentro de la carpeta que contiene el código fuente de la aplicación y utilizando la línea de comandos, el siguiente comando, haciendo los cambios correspondientes a la conexión con la base de datos.

dotnet ef dbcontext scaffold "Data Source=.\sqlexpress;Database=Northwind;Integrated Sec
urity=SSPI;" Microsoft.EntityFrameworkCore.SqlServer

En el siguiente enlace se puede encontrar un detalle de las diferentes opciones ofrecidas por el comando ef, https://docs.efproject.net/en/latest/miscellaneous/cli/dotnet.html#dotnet-

ef-dbcontext .

Entity Framework

- Consulta de datos.
- LINQ.

LINQ

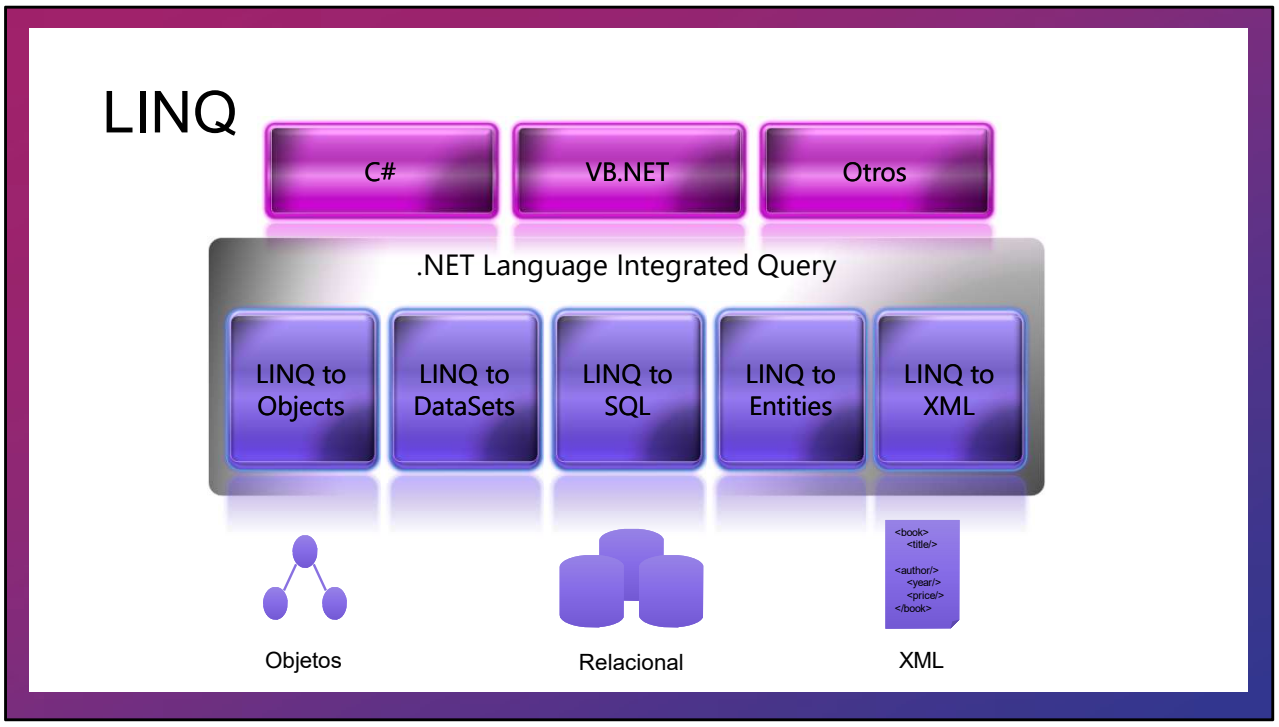
- **L**anguage **In**tegrated **Q**uery.
- Tecnología de búsqueda.
- Agnóstico a la fuente de datos.
- Sintaxis común.

LINQ son las iniciales en inglés para Language Integrated Query.

LINQ es una tecnología de búsqueda sobre repositorio de datos incorporada por Microsoft en la versión 3.5 del .NET Framework.

Este tecnología permite consultar información sobre distintos tipos de repositorios utilizando una sintaxis común en todos los casos

La sintáxis para consultas utilizada por LINQ es muy parecida a una consulta SQL



LINQ

- Expresiones Query
  - La sintáxis tipo SQL.
- Expresiones Lambda
  - El modelo de objetos de LINQ.

Existen 2 tipos de expresiones de consulta en LINQ:

Expresiones Query: La sintáxis tipo SQL.
Expresiones Lambda: El modelo de objetos de Linq.

Ambas sintáxis producen los mismos resultados.

La sintáxis tipo SQL es más legible para aquellas personas acostumbradas a trabajar con bases de datos.

La sintáxis con el modelo de objetos genera menos código y es más legible para los programadores acostumbrados a trabajar con modelos de objetos.

LINQ, Operadores

- Restricción
  - Where
- Proyección
  - Select, SelectMany
- Orden
  - OrderBy, OrderByDescending
  - ThenBy, ThenByDescending
  - Reverse

LINQ, Operadores

- Agrupamiento
  - GroupBy
- Enlace
  - Join, GroupJoin
- Conjunto
  - Distinct, Union, Intersect, Except
- Agregados
  - Count, LongCount, Sum, Min, Max, Average, Aggregate

LINQ, Operadores

- Cuantificadores
  - Any, All, Contains
- Partición
  - Take, TakeWhile, Skip, SkipWhile
- Elemento
  - First, FirstOrDefault, Last, LastOrDefault, Single, SingleOrDefault, ElementAt, ElementAtDefault, DefaultIfEmpty

LINQ, Operadores

- Conversión
  - AsEnumerable, Cast, OfType, ToArray, ToDictionary, ToList
- Otros
  - Concat

Entity Framework

- Manipulación de datos (CRUD).
- Transacciones.
- Anotaciones de datos.
  - Validación.
  - Presentación.
  - Estructura de Datos.
- Migrations.

System.ComponentMode.DataAnnotations
System.ComponentMode.DataAnnotations.Schema


universidad
cenfotec_
tecnologías digitales