Apuntes/Teoría 9

LINQ (Language-INtegrated Query)

Es un conjunto de tecnologías utilizadas para hacer operaciones sobre datos. Esto se utiliza para consultar bases de datos SQL, conjuntos de datos ADO.NET, secuencias y documentos XML y colecciones .NET. Provee un montón de métodos para datos de tipo IEnumerable<T>

Características y convenciones

- Es común usar un tipo var para usar LINQ por si se retorna un dato de tipo anónimo
- **Interfaz fluida (fluent API) →** sus métodos son capaces de retransmitir el contexto de la instrucción de una llamada subsecuente. Esto significa que se le pueden invocar métodos encadenados por puntos a un solo dato.
- **Nombre de las tablas** → se determina a partir del nombre de las propiedades DbSet<> de nuestra clase Context
- **Id** → Se establece como la claves de las tablas. Si esta es entero, se establece como *clave autoincremental*
- **Propiedades de navegación** → EF Core facilita la navegación entre entidades relacionadas por medio de estas propiedades.

(Algunos) métodos

.First()	.All()	.SingleOrDefault()	.Sum()	.Intersect()	.Join()
.Last()	.Any()	.Reverse()	.GroupJoin()	.Zip()	.Concat()
.Max()	.Select()	.OrderBy()	.Average()	.GroupBy()	.Distinct()
.Min()	.Where()	.ToList()	.Union()	.OrderByDescending()	

Persistencia de datos (con SQLite)

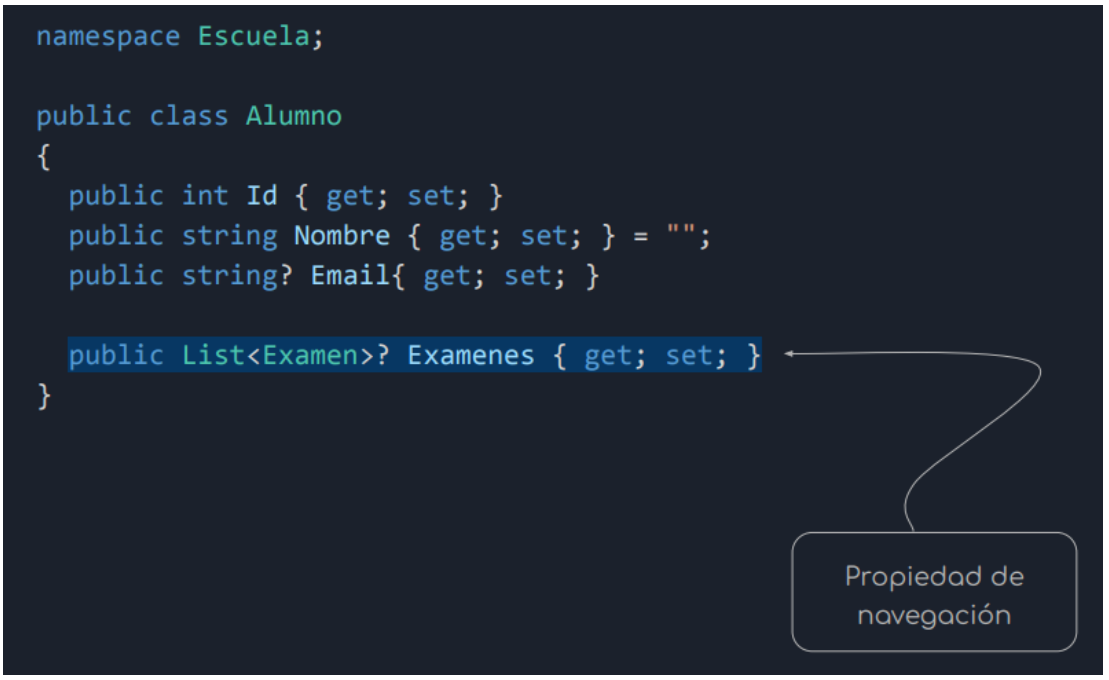
SQLite es una biblioteca que implementa un motor de base de datos SQL “en proceso”, autónomo, sin servidor

- **DB Browser for SQLite**
- **Entity Framework Core** → ORM (Object-Relational Mapper) → Modelo de programación que permite mapear las estructuras de una base de datos relacional
  - (dotnet add package Microsoft.EntityFrameworkCore.SQLite)
  - Con **EF Core**, el acceso a datos se realiza mediante un **modelo**, compuesto de **clases de entidad** y un **objeto de contexto** → representa una *sesión con la base de datos*
- **OnModelCreating()** → Especifica distintos aspectos sobre como se debería realizar el mapeo entre el modelo y la base de datos
- context.Add() / context.CLASE.Add(OBJETO)

- context.Remove()
- context.SaveChanges()
- **Code First →** Estrategia para crear bases de datos. Consiste en que el dev le meta datos iniciales para probarla
- **Migraciones** → ?

Propiedad de navegación

Permite relacionar un objeto con otro objeto en la base de datos. Con esto podremos asegurarnos de que un objeto borrado borre otros objetos de otro tipo con los cuales estaba vinculado



Si queremos agregar un alumno a la DB junto a un par de examenes, podemos hacerlo de esta forma