

Desarrollo de Web Api's con .NET

LinQ y EF



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

**Centro de
e-Learning**

Un poco de repaso...

- ¿Qué responsabilidades tienen el model, view y controlador en MVC?
- ¿Qué es y para qué sirve Razor?

¿Qué vamos a ver hoy?

- ORM
- LinQ
- Entity Framework
- Inyección de dependencia



¿Qué es un
ORM?

Object Relational Mapping

Mapeo objeto relacional.

Es un modelo de programación que consiste en la transformación de las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los datos para el programador.

Object Relational Mapping

Es una técnica de programación para convertir datos entre sistemas de tipos incompatibles utilizando lenguajes de programación orientados a objetos.

Esto crea, en efecto, una "base de datos de objetos virtuales" que se puede utilizar desde dentro del lenguaje de programación.

Algunas consideraciones

El uso de ORM nos abstrae de tener que “lidiar” con temas relacionados a la base de datos.

Siempre pensamos en objetos.

Ya no tenemos que hacer las queries de INSERT, SELECT, UPDATE y DELETE.

Puede mermar el rendimiento en entornos de gran carga.



¿Qué es un
LinQ?

Language **IN**tegrated Query

Es un conjunto de tecnologías basadas en la integración de funciones de consulta directamente en el lenguaje C#.

Tradicionalmente, las consultas sobre datos se expresan como cadenas simples sin verificación de tipo en tiempo de compilación o compatibilidad con IntelliSense.

Las consultas se escriben en colecciones de objetos fuertemente tipados mediante el uso de palabras clave de lenguaje y operadores.

[Documentación oficial](#)

Obteniendo el origen de datos

C#

```
var queryLondonCustomers = from cust in customers  
                             where cust.City == "London"  
                             select cust;
```

Where

C#

```
var queryLondonCustomers = from cust in customers  
                             where cust.City == "London"  
                             select cust;
```

C#

```
where cust.City == "London" || cust.City == "Paris"
```

OrderBy

C#

```
var queryLondonCustomers3 =  
    from cust in customers  
    where cust.City == "London"  
    orderby cust.Name ascending  
    select cust;
```

Group

C#

```
// queryCustomersByCity is an IEnumerable<IGrouping<string, Customer>>
var queryCustomersByCity =
    from cust in customers
    group cust by cust.City;

// customerGroup is an IGrouping<string, Customer>
foreach (var customerGroup in queryCustomersByCity)
{
    Console.WriteLine(customerGroup.Key);
    foreach (Customer customer in customerGroup)
    {
        Console.WriteLine("    {0}", customer.Name);
    }
}
```

Join

C#

```
var innerJoinQuery =  
    from cust in customers  
    join dist in distributors on cust.City equals dist.City  
    select new { CustomerName = cust.Name, DistributorName = dist.Name };
```

Language **IN**tegrated **Q**uery

```
class LINQQueryExpressions
{
    static void Main()
    {
        // Specify the data source.
        int[] scores = new int[] { 97, 92, 81, 60 };

        // Define the query expression.
        IEnumerable<int> scoreQuery =
            from score in scores
            where score > 80
            select score;

        // Execute the query.
        foreach (int i in scoreQuery)
        {
            Console.Write(i + " ");
        }
    }
}
// Output: 97 92 81
```

Entity Framework

Es un mapeador de bases de datos de objetos para .NET.

Admite consultas **LINQ**, seguimiento de cambios, actualizaciones y migraciones de esquemas.

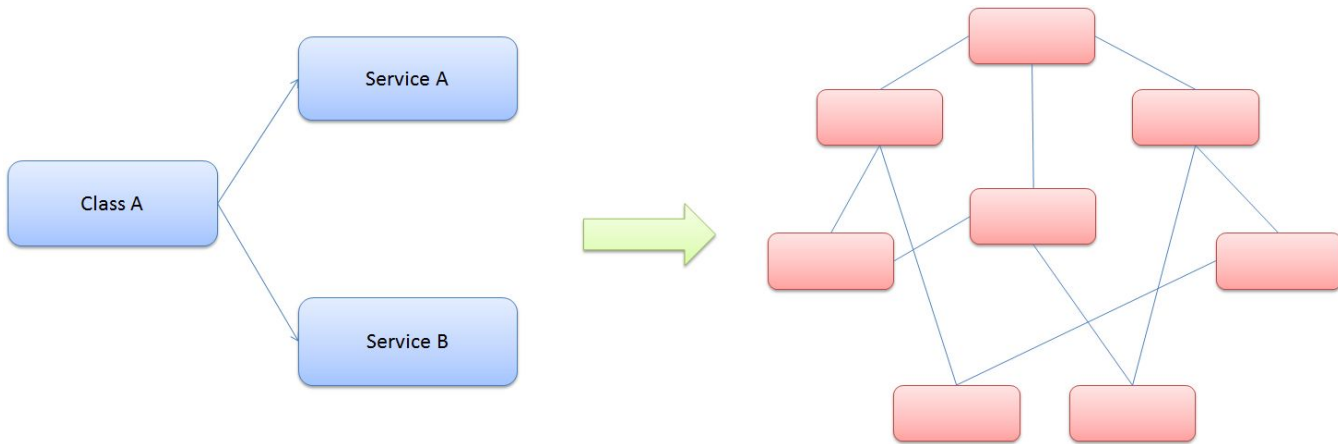
EF Core funciona con muchas bases de datos, incluida SQL Database (local y Azure), SQLite, MySQL, PostgreSQL y Azure Cosmos DB.

[Documentación oficial](#)

Inyección de dependencia

En el paradigma de POO, los objetos trabajan juntos en un modelo de colaboración donde hay contribuyentes y consumidores.

Naturalmente, este modelo de comunicación genera **dependencias** entre objetos y componentes, siendo difícil de gestionar cuando aumenta la complejidad.



Inyección de dependencia

Es una implementación particular de la inversión de control. La inversión de control (IoC) significa que los objetos no crean otros objetos en los que se basan para hacer su trabajo. En su lugar, obtienen los objetos que necesitan de una fuente externa (por ejemplo, un archivo de configuración xml).

Inyección de dependencia (DI) significa que esto se hace sin la intervención del objeto, generalmente por un componente del marco que pasa los parámetros del constructor y establece las propiedades.

Ventajas de usa DI

- Reduce el acoplamiento entre clases
- Aumenta la reutilización del código
- Mejora la mantención del código
- Mejora la aplicación de pruebas automáticas

¡¡A trabajar!!

