

LANGUAGE-INTEGRATED QUERY (LINQ)

Seminario T6-3 - Desarrollo de Software en
Visual Studio 2022

Ingeniería del Software

DSIC-UPV

2024-2025

This work is licensed under [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)



Objetivos

- Aprender las características fundamentales del lenguaje LINQ para acceso a datos de distintas fuentes en la plataforma .NET (C#)

Language-Integrated Query (LINQ)

- LINQ es un conjunto de funcionalidades del framework .NET de Microsoft que extiende la sintaxis de los lenguajes de programación aceptados (e.g. C# o Visual Basic) con expresiones o patrones que facilitan la recuperación de información de fuentes de datos distintas (e.g. base de datos, colecciones u objetos ADO)
- Las expresiones o patrones aceptados se asemejan a las funciones y *lambda* expresiones de lenguajes de programación funcionales como Haskell, OCaml o F#
 - Hoy día las expresiones *lambda* están presentes en la mayoría de lenguajes importantes, incluido Java desde la versión 8, y son muy comunes en lenguajes basados en eventos como Apple Swift o Google Kotlin

De LINQ a SQL

- Vamos a utilizar LINQ para acceder de forma sencilla a bases de datos.
- Las expresiones LINQ se traducen internamente en expresiones SQL.
- Permiten olvidarnos de los detalles internos de cada arquitectura de base de datos y dejar los detalles de conexión, acceso y consulta al sistema de traducción de LINQ a SQL.

Sintaxis de LINQ

- Una expresión LINQ consiste en tres partes diferenciadas:

1. Obtener la fuente de datos

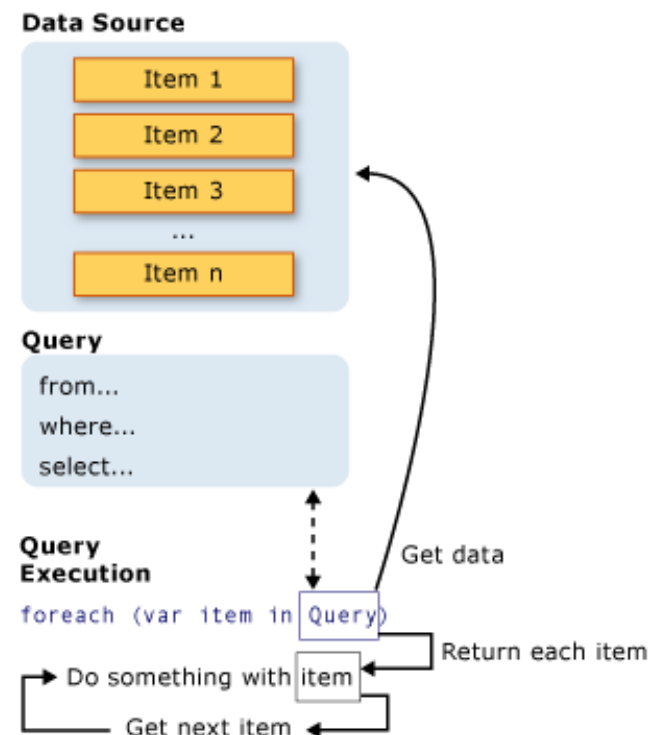
```
var bo = dbcontext.offices
```

2. Crear la consulta

```
var offices_query =  
    from office in bo  
    where office.Id == "Valencia"  
    select office;
```

3. Ejecutar la consulta

```
foreach (BranchOffice office in offices_query)  
    Console.WriteLine(office.Id,office.address);
```



Lambda expresiones en LINQ

- LINQ permite una forma abreviada para introducir expresiones, conocida como *method-based*. Esta forma usa lambda expresiones como los lenguajes funcionales.
 - Los dos fragmentos de código siguientes hacen lo mismo.

- Código 1:

```
var offices_query =  
    from office in dbcontext.offices  
    where office.Id == "Valencia"  
    select office;  
foreach (BranchOffice office in offices_query)  
    Console.WriteLine(office.Id,office.address);
```

- Código 2

```
var offices_query = dbcontext.offices.Where(office => office.Id == "Valencia")  
foreach (BranchOffice office in offices_query)  
    Console.WriteLine(office.Id,office.address);
```

Where en LINQ

- El método Where en una expresión LINQ devuelve un objeto de tipo `IEnumerable`.
- Este objeto admite muchos otros métodos que pueden resultar útiles, p.ej. `OrderBy`, `Distinct`, `FirstOrDefault`, etc.

LINQ en C#

- LINQ está integrado en distintas funcionales de C#:

- Consultas, como se ha mostrado anteriormente
- Variables sin tipo declarado.

```
var number = 5;  
var name = "Virginia";  
var query = from str in stringArray  
             where str[0] == 'm'  
             select str;
```

- Inicializadores de objetos y colecciones.

```
Customer cust = new Customer { Name = "Mike", Phone = "555-1212" };
```

- Tipos anónimos

```
select new {name = cust.Name, phone = cust.Phone};
```

- Extensión de métodos
- Expresiones Lambda

- Propiedades auto implementadas

```
public string Name {get; set;}
```


Conclusiones

- Funcionalidades de lenguajes de programación funcionales integradas en lenguajes de programación, incluyendo Microsoft .NET y Java
- Diversas extensiones y aplicaciones: SQL, objetos ADO, interfaces gráficas, etc.