



Fundamentos de Desenvolvimento Web/BD

Aula 01: LINQ

Professor: Rinaldo Ferreira Junior

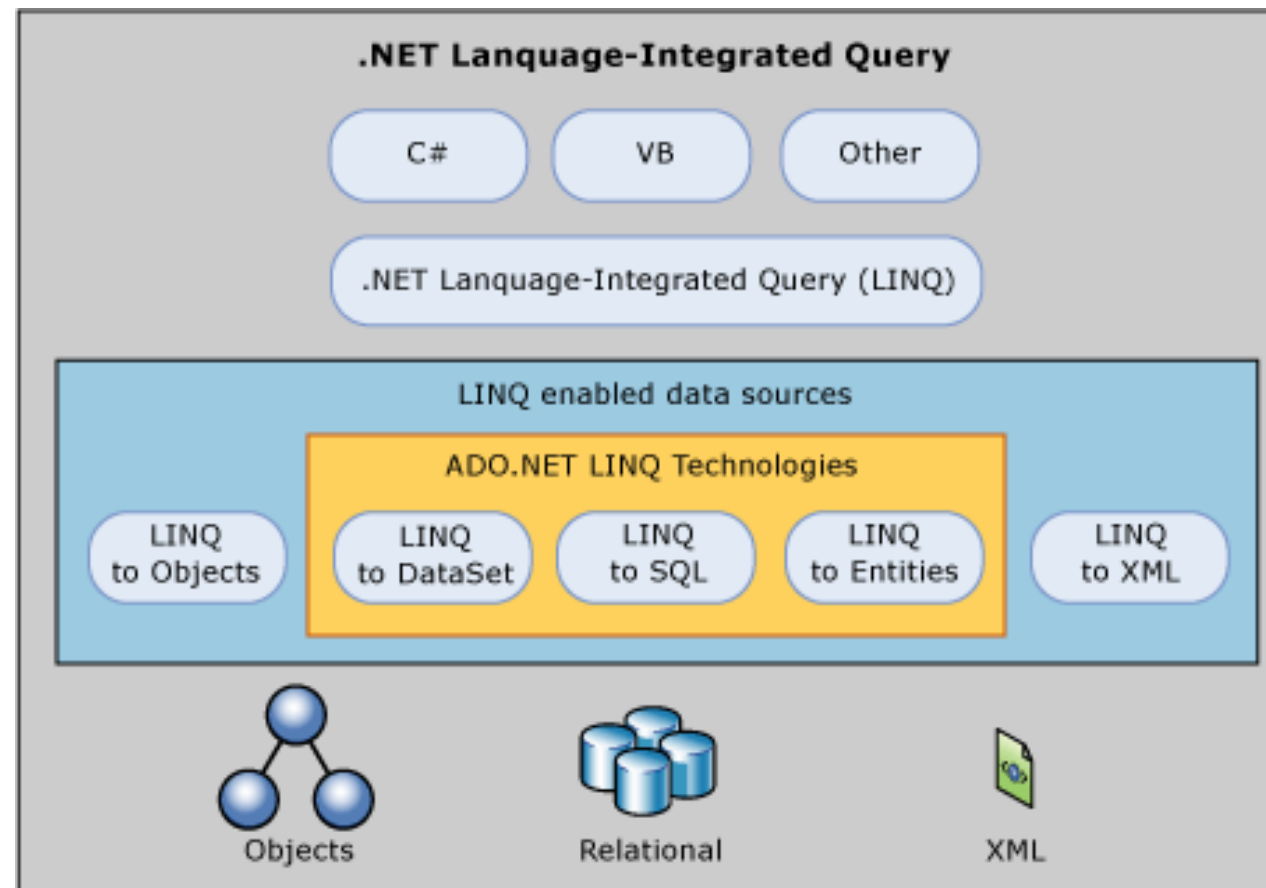
E-mail: rinaldo.fjunior@prof.infnet.edu.br



- **Professor:** Rinaldo Ferreira Junior
- **Graduação:** Pós-graduado em Arquitetura de Softwares
- **Atuação:** .Net | C# | SQL | NoSQL | Engenheiro de Software
- **E-mail:** rinaldo.fjunior@prof.infnet.edu.br
- **Linkedin:** <https://www.linkedin.com/in/rinaldo-ferreira-junior-787326a>

- LINQ

- **LINQ** é o acrônimo para **Language Integrated Query**, uma parte do framework que permite codificar com um estilo baseado em queries.
- **LINQ** abstrai o acesso à fontes de dados, o que reduz a curva de aprendizado, pois não é necessário foca em linguagens específicas como SQL, XPath etc.



- Queries tradicionais, como queries SQL, são instruções em forma de texto que não possuem Intellisense ou checagem de tipo.
- **LINQ** suporta escrita de código em duas versões:
 - Query Syntax
 - Method Syntax
- Em ambas as formas, há suporte de Intellisense e checagem de tipo, em design-time.
- O namespace responsável por manter as classes do LINQ é o **System.Linq**.

Query Syntax

```
static void Main(string[] args)
{
    int[] values = [18, 97, 92, 81, 60];

    IEnumerable<int> resultQuery =
        from value in values
        where value > 80
        select value;

    foreach (int i in resultQuery)
    {
        Console.Write(i + " ");
    }

    Console.ReadLine();
}
```

97 92 81

Method Syntax

```
static void Main(string[] args)
{
    int[] values = [18, 97, 92, 81, 60];

    IEnumerable<int> resultQuery = values
        .Where(v => v > 80)
        .Select(v => v);

    foreach (int i in resultQuery)
    {
        Console.Write(i + " ");
    }

    Console.ReadLine();
}
```

97 92 81

- A instrução **Where** executa o filtro informado em uma coleção. Essa coleção pode estar em memória ou em uma base de dados.

```
static void Main(string[] args)
{
    List<Pessoa> pessoas = new()
    {
        new() { Nome = "Juca", SobreNome = "Bala", Cpf = "001.001.001-99", Idade = 39},
        new() { Nome = "Tião", SobreNome = "Veneno", Cpf = "007.007.007-91", Idade = 43},
        new() { Nome = "Jussara", SobreNome = "Chumbinho", Cpf = "004.004.004-88", Idade = 37}
    };

    var result = pessoas.Where(p => p.Nome.StartsWith("j", StringComparison.OrdinalIgnoreCase)
                                ||
                                p.Nome.EndsWith("o", StringComparison.OrdinalIgnoreCase));

    foreach (Pessoa i in result)
    {
        Console.WriteLine($"Nome: {i.Nome}");
    }

    Console.ReadLine();
}
```

- A instrução **Select** define o que deve ser retornado após o filtro.

```
static void Main(string[] args)
{
    List<Pessoa> pessoas = new()
    {
        new() { Nome = "Juca", SobreNome = "Bala", Cpf = "001.001.001-99", Idade = 39},
        new() { Nome = "Tião", SobreNome = "Veneno", Cpf = "007.007.007-91", Idade = 43},
        new() { Nome = "Jussara", SobreNome = "Chumbinho", Cpf = "004.004.004-88", Idade = 37}
    };

    var result = pessoas.Where(p => p.Idade > 40)
                        .Select(s => s)
                        .ToList();

    foreach (Pessoa i in result)
    {
        Console.WriteLine($"Nome: {i.Nome} / Idade: {i.Idade}");
    }

    Console.ReadLine();
}
```


- A instrução **OrderBy** classifica o resultado por um ou mais dos fields de retorno.

```
static void Main(string[] args)
{
    List<Pessoa> pessoas = new()
    {
        new() { Nome = "Juca", SobreNome = "Bala", Cpf = "001.001.001-99", Idade = 39},
        new() { Nome = "Tião", SobreNome = "Veneno", Cpf = "007.007.007-91", Idade = 43},
        new() { Nome = "Jussara", SobreNome = "Chumbinho", Cpf = "004.004.004-88", Idade = 37},
        new() { Nome = "Fulgêncio", SobreNome = "Fogueira", Cpf = "009.008.007-81", Idade = 31}
    };

    var result = pessoas.Where(p => p.Idade > 35)
                        .Select(s => s)
                        .OrderBy(p => p.Idade)
                        .ToList();

    foreach (Pessoa i in result)
    {
        Console.WriteLine($"Nome: {i.Nome} / Idade: {i.Idade}");
    }

    Console.ReadLine();
}
```

- O mesmo trecho em Query Syntax.

```
static void Main(string[] args)
{
    List<Pessoa> pessoas = new()
    {
        new() { Nome = "Juca", SobreNome = "Bala", Cpf = "001.001.001-99", Idade = 39},
        new() { Nome = "Tião", SobreNome = "Veneno", Cpf = "007.007.007-91", Idade = 43},
        new() { Nome = "Jussara", SobreNome = "Chumbinho", Cpf = "004.004.004-88", Idade = 37},
        new() { Nome = "Fulgêncio", SobreNome = "Fogueira", Cpf = "009.008.007-81", Idade = 37}
    };

    var result = from p in pessoas
                  where p.Idade > 35
                  orderby p.Idade, p.Nome
                  select p;

    foreach (Pessoa i in result)
    {
        Console.WriteLine($"Nome: {i.Nome} / Idade: {i.Idade}");
    }

    Console.ReadLine();
}
```