

# IFRN

## PROGRAMAÇÃO ORIENTADA A OBJETOS EM C#

---

### LINQ para Objetos

Prof. Gilbert Azevedo

# Objetivos

- Conhecer a Linguagem de Consulta LINQ
  - LINQ para Objetos
  - Métodos da LINQ: Select, Where, OrderBy, ThenBy, Count, Distinct, Join

# Visão Geral

- LINQ – Language Integrated Query (Consulta integrada à linguagem)
- Introduzido no VS 2008, .Net Framework 3.5
- Preenche a lacuna entre o mundo do objeto e o mundo dos dados
- Realiza consulta de dados de fontes distintas (coleções, arquivos XML, bancos de dados) com mesma sintaxe
- É verificada pelo compilador
- É utilizada em coleções enumeráveis – IEnumerable e consultáveis – IQueryable

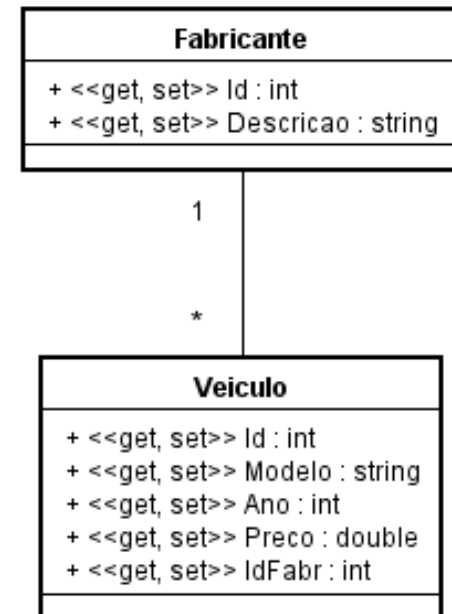
# Tecnologias do LINQ

- LINQ para Objetos
  - Para coleções que implementam `IEnumerable` ou `IEnumerable<T>`
  - Ex: Array, List, Dictionary, ...
- LINQ para XML
  - Manipulação de arquivos XML
- LINQ para ADO.NET
  - LINQ para DataSets
  - LINQ para SQL
  - LINQ para Entities

# Classes de Exemplo

```
class Fabricante {  
    public int Id { get; set; }  
    public string Descricao { get; set; }  
}
```

```
class Veiculo {  
    public int Id { get; set; }  
    public string Modelo { get; set; }  
    public int Ano { get; set; }  
    public double Preco { get; set; }  
    public int IdFabr { get; set; }  
}
```



# Listas de Fabricantes e Veículos

```
var fs = new List<Fabricante> {  
    new Fabricante { Id = 1, Descricao = "Honda" },  
    new Fabricante { Id = 2, Descricao = "VW" },  
    new Fabricante { Id = 3, Descricao = "Fiat" },  
    new Fabricante { Id = 4, Descricao = "GM" }  
};  
  
var vs = new List<Veiculo> {  
    new Veiculo { Id = 1, Modelo = "Punto", Ano = 2010,  
                  Preco = 55000, IdFabr = 3 },  
    new Veiculo { Id = 2, Modelo = "Gol", Ano = 2015,  
                  Preco = 40000, IdFabr = 2 },  
    new Veiculo { Id = 3, Modelo = "Corsa", Ano = 2015,  
                  Preco = 38000, IdFabr = 4 },  
    new Veiculo { Id = 4, Modelo = "Civic", Ano = 2019,  
                  Preco = 100000, IdFabr = 1 }  
};
```

# Enumerable

- Classe do System.Linq que fornece um conjunto de métodos estáticos para consultar objetos que implementam IEnumerable<T>.
  - Select
  - Where
  - OrderBy, ThenBy
  - Count, Distinct
  - Join
- [Enumerable Classe \(System.Linq\) | Microsoft Docs](#)

# Select

- Select obtém uma coleção de objetos (TResult) a partir de uma coleção de objetos (T), utilizando um método seletor
  - `IEnumerable<TResult> IEnumerable<T>.Select(Func<T, TResult> selector)`
- Exemplo
  - `IEnumerable<string> x = fs.Select(f => f.Descricao);`
  - `IEnumerable<string> y = from f in fs select f.Descricao;`
- Resultado
  - Coleção de strings com a descrição dos fabricantes
  - `foreach (string s in x) Console.WriteLine(s);`



# Where

- Where filtra uma coleção de objetos (T), utilizando um predicado para testar se cada um dos objetos deve ser incluído ou não no resultado
  - `IEnumerable<T> IEnumerable<T>.Where(Func<T, bool> predicate)`
- Exemplo
  - `var x = vs.Where(v => v.Preco < 50000);`
  - `var y = from v in vs where v.Preco < 50000 select v;`
- Resultado
  - Coleção de veículos com preço menor que 50000

# OrderBy

- OrderBy ordena uma coleção de objetos (T), utilizando um seletor de chave no objeto T para definir a ordenação
  - `IOrderedEnumerable<T> IEnumerable<T>.OrderBy(Func<T, TKey> keySelector)`
- Exemplo
  - `var x = vs.OrderBy(v => v.Preco);`
  - `var y = from v in vs orderby v.Preco select v;`
- Resultado
  - Coleção de veículos ordenada pelo preço

# ThenBy

- ThenBy ordena uma coleção de objetos (T) após uma 1ª ordenação e utiliza também um seletor no objeto T para definir a nova ordem
  - `IOrderedEnumerable<T> IOrderedEnumerable<T>.ThenBy(Func<T, TKey> keySelector)`
- Exemplo
  - `var x = vs.OrderByDescending(v=>v.Ano).ThenBy(v=>v.Preco);`
  - `var y = from v in vs orderby v.Ano descending, v.Preco`
  - `select v;`
- Resultado
  - Coleção de veículos ordenada pelo ano em ordem descendente e depois pelo preço ascendente

# Count

- Count retorna o número de elementos em uma coleção
  - `int IEnumerable<T>.Count()`
  - `int IEnumerable<T>.Count(Func<T, bool> predicate)`
- Exemplos
  - `int c1 = vs.Count();`
  - `int c2 = vs.Count(v => v.Preco < 50000);`
- Resultados
  - Quantidade de veículos e quantidade de veículos com preço menor que 50000

# Distinct

- Distinct retorna os elementos distintos em uma coleção
  - `IEnumerable<T> IEnumerable<T>.Distinct()`
- Exemplos
  - `int[] w = { 1, 1, 2, 2, 3, 4, 5, 5, 6 };`
  - `IEnumerable<int> x = w.Distinct();`
  - `IEnumerable<int> y = vs.Select(v => v.Ano).Distinct();`
- Resultados
  - Valores inteiro diferentes no vetor w
  - Diferentes anos dos veículos na lista

# Join

- Join realiza a junção de duas coleções (T1, T2) utilizando um seletor de chave comum aos objetos para definir essa junção
  - `IEnumerable<TResult> IEnumerable<T1>.Join(`
    - `IEnumerable<T2> inner, Func<T1, TKey> outerKeySelector,`
    - `Func<T2, TKey> innerKeySelector,`
    - `Func<T1, T2, TResult> resultSelector)`
- Exemplo
  - `var x1 = vs.Join(fs, v => v.IdFabr, f => f.Id,`
  - `(v, f) => new { f.Descricao, v.Modelo, v.Ano, v.Preco });`
  - `var x2 = vs.Join(fs, v => v.IdFabr, f => f.Id,`
  - `(v, f) => new {v, f});`

# Join

- `var y1 =`
- `from v in vs join f in fs`
- `on v.IdFabr equals f.Id`
- `select new { f.Descricao, v.Modelo, v.Ano, v.Preco };`
  
- `var y2 =`
- `from v in vs`
- `join f in fs`
- `on v.IdFabr equals f.Id`
- `select new { v, f };`

# Referências

- LINQ - Consulta Integrada à Linguagem
  - <https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/concepts/linq/>
- LINQ e o ADO.NET
  - <https://docs.microsoft.com/pt-br/dotnet/framework/data/adonet/linq-and-ado-net>
- LINQ to Objects
  - <https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/concepts/linq/linq-to-objects>
- LINQ to Entities
  - <https://docs.microsoft.com/pt-br/dotnet/framework/data/adonet/ef/language-reference/linq-to-entities>



# Referências

- LINQ to SQL
  - <https://docs.microsoft.com/pt-br/dotnet/framework/data/adonet/sql/linq/>
- Documentação do Entity Framework
  - <https://docs.microsoft.com/pt-br/ef/>
- Classe Enumerable
  - <https://docs.microsoft.com/pt-br/dotnet/api/system.linq.enumerable?view=net-5.0>