

Laboratório 10 – LINQ to Object

Este laboratório introduz a manipulação de consultas LINQ sobre coleções de objetos.

1 Declarar e executar consultas LINQ

1. Crie um novo projeto “Console Application” com nome “Laboratorio10”.
2. Adicione uma nova classe com o nome “Pessoa” contendo código a seguir:

```
class Pessoa
{
    public bool Casada
    {
        get;
        set;
    }

    public String Nome
    {
        get;
        set;
    }

    public DateTime DataNascimento
    {
        get;
        set;
    }

    public override string ToString()
    {
        return "[Nome=" + nome + ", Casada=" + casada + ", DataNascimento=" +
dataNascimento.ToShortDateString() + "]\n";
    }
}
```

3. No método Main da classe “Program” adicione o seguinte código para inicializar uma lista de objetos Pessoa:

```
List<Pessoa> pessoas = new List<Pessoa>
{
    new Pessoa{Nome="Ana",DataNascimento=new DateTime(1980,3,14), Casada=true},
    new Pessoa{Nome="Paulo",DataNascimento=new DateTime(1978,10,23), Casada=true},
    new Pessoa{Nome="Maria",DataNascimento=new DateTime(2000,1,10), Casada=false},
    new Pessoa{Nome="Carlos",DataNascimento=new DateTime(1999,12,12), Casada=false}
};
```

4. Adicione a seguinte consulta LINQ:

```
var linq1 =
    from p in pessoas
    where p.Casada && p.DataNascimento >= new DateTime(1980, 1, 1)
    select p;

foreach (var pessoa in linq1)
{
    Console.WriteLine(pessoa);
}
```

```
}
```

5. Teste a aplicação.

6. Adicione a seguinte consulta LINQ:

```
var linq2 = pessoas.Where(p => p.Casada && p.DataNascimento >= new DateTime(1980, 1, 1));  
  
foreach (var pessoa in linq2)  
{  
    Console.WriteLine(pessoa);  
}
```

7. Teste a aplicação e observe que o resultado é o mesmo.

3 Exercícios

1. Construa uma consulta que retorne as pessoas agrupadas em casadas e solteiras e também o número de pessoas casadas e solteiras.
2. Construa uma consulta que retorne a pessoa mais velha.
3. Construa uma consulta que retorne a pessoa solteira mais nova.