

```

public class Empresa
{
    public int Id { get; set; }
    public string Nome { get; set; }
}

public class Pessoa
{
    public int Id { get; set; }
    public string Nome { get; set; }
    public string Sobrenome { get; set; }
    public DateTime DataNascimento { get; set; }
    public Empresa Empresa { get; set; }
}

```

O rascunho do código fonte de todos os exercícios estão no projeto da pasta teste\_backend

1. Escreva o código em Linq para executar ordenação descendente por Sobrenome de uma lista da classe acima (`List<Pessoa>`)

```

List<Empresa> Empresas = new List<Empresa>()
{
    new Empresa { Id=1, Nome="Encripta" },
    new Empresa { Id=2, Nome="Nao Encripta" },
};

List<Pessoa> Pessoas = new List<Pessoa>()
{
    new Pessoa { Id=2, Nome="Bdriano", Sobrenome="Daverson",
DataNascimento=new DateTime(1976, 04, 02), Empresa=Empresas[1]},
    new Pessoa { Id=3, Nome="Cdriano", Sobrenome="Vaverson",
DataNascimento=new DateTime(1976, 02, 02), Empresa=Empresas[1]},
    new Pessoa { Id=4, Nome="Ddriano", Sobrenome="Naverson",
DataNascimento=new DateTime(1976, 01, 02), Empresa=Empresas[1]},
    new Pessoa { Id=1, Nome="Adriano", Sobrenome="Caverson",
DataNascimento=new DateTime(1976, 07, 02), Empresa=Empresas[0]},
};

var orderedPessoas = Pessoas.OrderByDescending(Pessoa => Pessoa.Sobrenome).ToList();

foreach (var Pessoa in orderedPessoas)
{
    Console.WriteLine("{0} {1} {2} {3}", Pessoa.Nome, Pessoa.Sobrenome, String.Format("{0:dd/MM/yyyy}", Pessoa.DataNascimento), Pessoa.Empresa.Nome);
}

```

Resultado:

Cdriano Vaversan 02/02/1976 Nao Encripta  
Ddriano Naversan 02/01/1976 Nao Encripta  
Bdriano Daversan 02/04/1976 Nao Encripta  
Adriano Caversan 02/07/1976 Encripta

2. Escreva um loop na mesma classe usando Parallel.ForEach (ou Parallel.For) que faça a iteração em todos os itens da lista.

```
Parallel.ForEach(orderedPessoas, Pessoa => {  
    Console.WriteLine("{0} {1} {2} {3}", Pessoa.Nome, Pessoa.  
Sobrenome, String.Format("{0:dd/MM/yyyy}", Pessoa.DataNascimento), Pessoa  
.Empresa.Nome);  
});
```

Resultado: **(Ordem inesperada)** Exemplo

Ddriano Naversan 02/01/1976 Nao Encripta  
Adriano Caversan 02/07/1976 Encripta  
Bdriano Daversan 02/04/1976 Nao Encripta  
Cdriano Vaversan 02/02/1976 Nao Encripta

3. Escreva em c# dois design patterns de sua escolha e diga para que servem

Iterator: Permite que eu consiga percorrer uma coleção de itens dentro de uma lista sem que eu precise conhecer todos seus valores ou sua extensão desta lista o que facilita muito na exibição de uma amostra de valores dentro de uma lista muito extensa (exemplo: uma página com o resultado de uma pesquisa com apenas uma dezena de filmes dentro de um catálogo de milhares)

```
using System;  
class Program  
{  
    public class Weeks //Aggregate object  
    {  
        private string[] weeks = new string[]{  
            "Monday",  
            "Tuesday",  
            "Wednesday",  
            "Thursday",  
            "Friday",  
            "Saturday",  
            "Sunday"  
        };  
    };  
}
```

```

    public IWeeksIterator GetWeeksIterator()
    {
        //creates an Iterator object
        return new WeeksIterator(weeks);
    }
}

public interface IWeeksIterator //Iterator interface
{
    string Current { get; }

    bool MoveNext();
}

public class WeeksIterator : IWeeksIterator //Iterator object
{
    private readonly string[] weeks;
    private int position;

    public WeeksIterator(string[] weeks)
    {
        this.weeks = weeks;
        this.position = -1;
    }

    public string Current => weeks[position];

    public bool MoveNext()
    {
        if (++position == weeks.Length) return false;
        return true;
    }
}

static void Main(string[] args)
{
    var weeks = new Weeks();
    var iterator = weeks.GetWeeksIterator();
    while (iterator.MoveNext())
    {
        Console.WriteLine(iterator.Current);
    }
    Console.ReadLine();
}
}

```

Flyweight: Este padrão permite que eu compartilhe partes do estado de um objeto armazenado na memória RAM entre vários objetos de características iguais ou parecidas trazendo grande economia de memória RAM.

```
using System;
using System.Collections.Generic;

namespace teste_encripta
{
    class Program
    {
        //Flyweight

        interface IShape
        {
            void Print();
        }
        class Rectangle : IShape
        {
            {
                public void Print()
                {
                    Console.WriteLine("Printing Rectangle");
                }
            }
        }

        class Circle : IShape
        {
            {
                public void Print()
                {
                    Console.WriteLine("Printing Circle");
                }
            }
        }
        class ShapeObjectFactory
        {
            Dictionary<string, IShape> shapes = new Dictionary<string, IShape>();

            public int TotalObjectsCreated
            {
                {
                    get { return shapes.Count; }
                }
            }

            public IShape GetShape(string ShapeName)
            {
                IShape shape = null;
                if (shapes.ContainsKey(ShapeName))
                {
                    shape = shapes[ShapeName];
                }
            }
        }
    }
}
```

```

        }
        else
        {
            switch (ShapeName)
            {
                case "Rectangle":
                    shape = new Rectangle();
                    shapes.Add("Rectangle", shape);
                    break;
                case "Circle":
                    shape = new Circle();
                    shapes.Add("Circle", shape);
                    break;
                default:
                    throw new Exception("Factory cannot create the ob
ject specified");
            }
        }
        return shape;
    }
}

static void Main()
{
    //Flyweight
    ShapeObjectFactory sof = new ShapeObjectFactory();

    IShape shape = sof.GetShape("Rectangle");
    shape.Print();
    shape = sof.GetShape("Rectangle");
    shape.Print();
    shape = sof.GetShape("Rectangle");
    shape.Print();

    shape = sof.GetShape("Circle");
    shape.Print();
    shape = sof.GetShape("Circle");
    shape.Print();
    shape = sof.GetShape("Circle");
    shape.Print();

    int NumObjs = sof.TotalObjectsCreated;
    Console.WriteLine("\nTotal No of Objects created = {0}", NumO
bjs);
}
}

```

```
}
```

4. Escreva um extension method qualquer para uma string

```
using System;
using System.Linq;
using System.Collections.Generic;
using CustomExtensions;

namespace CustomExtensions
{
    // Extension methods must be defined in a static class.
    public static class StringExtension
    {
        // This is the extension method.
        // The first parameter takes the "this" modifier
        // and specifies the type for which the method is defined.
        public static int WordCount(this String str)
        {
            return str.Split(new char[] { ' ', '.', '?' }, StringSplitOptions
s.RemoveEmptyEntries).Length;
        }
    }
}

namespace teste_encrypta
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Questão 4 Extension Methods");
            // Questão 4 Extension Methods
            // Import the extension method namespace.
            string s = "The quick brown fox jumped over the lazy dog.";
            // Call the method as if it were an
            // instance method on the type. Note that the first
            // parameter is not specified by the calling code.
            int i = s.WordCount();
            System.Console.WriteLine("Word count of s is {0}", i);
        }
    }
}
```

5. No Entity Framework, o que faz o método Include?

O método include() é usado para carregar conteúdo dinamicamente na página, pode ser utilizado em estratégias de carregamento adiantado (pré-carregamento da página) ou de carregamento progressivo ou sob demanda (Lazy Load).

6. Qual a diferença entre FirstOrDefault(); e First(); do Linq?

O First() retorna o primeiro elemento de uma sequência já o FirstOrDefault() retorna o primeiro item de uma sequência OU um valor padrão caso ele não exista.

7. No Controller do MVC cite dois possíveis retornos como ActionResult e explique

JsonResult: Retorna um dado no formato json para ser utilizado na página, suponhamos que temos uma interface em React, podemos interagir via Json com um backend em C# para modificar dinamicamente o resultado visto em uma página.

JavaScriptResult: Com este tipo de ActionResult podemos implementar ou manipular código JavaScript a partir de uma ação o que é muito útil em aplicações complexas que necessitam de carregamento ou execução progressiva de código JavaScript

8. Usando Jquery, selecione todos os elementos <DIV> de uma página que contenham a classe CSS "okButton"

```
$('.okButton')
```

9. Usando Jquery, adicione e remova a classe CSS de um elemento

```
$('.okButton').removeClass('okButton').addClass('notOkButton');
```

10. Escreva um algoritmo de fatorial usando C#.

```
int fatorial = 1;
for (int n = 1; n <= 10; n++)
{
    fatorial*=n;
    Console.WriteLine(n+ " fatorial= "+fatorial);
}
```

Resultado:

1 fatorial= 1

2 fatorial= 2

3 fatorial= 6

4 fatorial= 24  
5 fatorial= 120  
6 fatorial= 720  
7 fatorial= 5040  
8 fatorial= 40320  
9 fatorial= 362880  
10 fatorial= 3628800

11. Qual o código SQL resultando do comando

```
(from t in db.TB_User where t.name.StartsWith(variável) orderby t.name select t)
```

```
SELECT t FROM d.TB_User WHERE t.name LIKE 'variável' ORDER BY t.name
```

12. Choose a sorting algorithm and explain it.

Quicksort: Is a sorting algorithm where the algorithm divide the list by two. The chosen element is called pivot. All elements smaller than the pivot are moved before it and all greater elements are moved after it dividing this list in two unsorted lists an a new pivot is chosen recursively.

13. What is the difference between Hashing and Encrypting.

The hashing is an one way encryption, when we use an encryption algorithm in a file or data like a MD5 or SHA for example, we cannot able to decrypt the encrypted data with a decrypted key the decryption is possible only by brute force. However the encrypting permits an encrypt or decrypt keys creation.

14. What is the difference between symmetric and asymmetric keys?

In symmetric encryption keys the encryption and decryption keys are same keys. However in asymmetric keys we can encrypt our data with a private key, that key is secret, and the data only can be decrypted with a public key created from the private key, that key can be distributed.

15. Explain the man-in-the-middle attack.

In the man-in-the-middle attack the hacker need intercept an insecure communication to get private data normally creating fake pages hosted worldwide or in fake wi-fi networks. This attack consists in attract the victim to a fake environment where the victim send your private data to the hacker without the victim knows that it happened.

16. What is the difference between a process and a thread?

Threads are independent parts of a running process. Process can have many threads sharing a same computing resource simultaneously, but process don't share resources between process. Process is like a specific program running in computer and the threads are parts of this programs.

17. What is a mutex?

Mutex (mutual exclusion) is a property of concurrency control, is a binary flag used to signalize another threads about your execution.