# LAB 6 Interfaces do Framework

Este laboratório introduz o conceito de implementação de ***interfaces*** presentes no .NET Framework.

## Exercício 1 – Utilizando método de ordenação padrão

1. Crie um novo projeto do tipo ***ASP.NET Empty Web Application***.
2. Adicione ao projeto um formulário do tipo *Web Form*.
3. Escreva o seguinte código no método ***Page\_Load*** para testar a *ordenação* de *arrays*:

string[] lista = { "Alexandre", "Julia", "Daniel", "Gabriel" };

Response.Write("Array antes da ordenação<br/>");

foreach (string nome in lista)

{

Response.Write(nome + " ");

}

Response.Write("<br/>");

Array.Sort(lista);

Response.Write("Array depois da ordenação<br/>");

foreach (string nome in lista)

{

Response.Write(nome + " ");

}

1. Compile e execute o programa.
2. Adicione a ***classe Pessoa*** ao projeto:

public class Pessoa

{

private string meuNome;

private int minhaIdade;

public Pessoa(string n, int i)

{

meuNome = n;

minhaIdade = i;

}

public string Nome

{

get { return meuNome; }

}

public int Idade

{

get { return minhaIdade; }

set { minhaIdade = value; }

}

}

1. Adicionar o código abaixo ao método ***Page\_Load***.

...

Response.Write("<br/>");

Pessoa[] lista2 = {

new Pessoa("Amanda", 25),

new Pessoa("Ana", 28),

new Pessoa("Paulo", 20)

};

Array.Sort(lista2);

Response.Write("Array depois da ordenação<br/>");

foreach (Pessoa p in lista2)

{

Response.Write(p.Nome + " ");

}

1. Executar o programa e verificar o seu comportamento.

## Exercício 2 – Implementando interfaces presentes no .NET Framework (IComparable)

1. Alterar a definição da classe “*Pessoa*” para indicar a implementação da ***interface IComparable***.

public class Pessoa : IComparable<Pessoa>

1. Adicionar o seguinte código à classe “*Pessoa*” para implementar a ***interface IComparable***.

public int CompareTo(Pessoa outro)

{

return meuNome.CompareTo(outro.meuNome);

}

1. Compilar e executar o programa novamente.

**Exercícios**

1. Altere o método ***CompareTo()*** da *classe* *Pessoa* para que a comparação seja realizada pela *idade* da pessoa em *ordem crescente*.
2. Você conseguiria criar uma classe *Pessoa* que permitisse *ordenar* por *nome* ou *idade*, utilizando a ***interface IComparable***?

## Exercício 3 – Implementando interfaces presentes no .NET Framework (IComparer)

1. Com o botão direito do mouse clique sobre o projeto e selecione o menu ***Add****->****New Item***.
2. Na lista selecione ***CodeFile*** e coloque o nome “Comparadores.cs”.
3. Adicione o seguinte código ao arquivo para implementar uma classe “ComparadorNomePessoa” e uma classe “ComparadorIdadePessoa”:

using System;

using System.Collections.Generic;

using System.Collections;

namespace LAB6

{

public class ComparadorNomePessoa : IComparer

{

public int Compare(object x, object y)

{

Pessoa p1 = (Pessoa)x;

Pessoa p2 = (Pessoa)y;

return p1.Nome.CompareTo(p2.Nome);

}

}

public class ComparadorIdadePessoa : IComparer

{

public int Compare(object x, object y)

{

Pessoa p1 = (Pessoa)x;

Pessoa p2 = (Pessoa)y;

return p1.Idade - p2.Idade;

}

}

}

1. Adicione ao método ***Page\_Load*** o seguinte código:

...

Response.Write("<br/>");

Array.Sort(lista2, new ComparadorIdadePessoa());

Response.Write("Array depois da ordenação<br/>");

foreach (Pessoa p in lista2)

{

Response.Write(p.Nome + " ");

}

1. Compile e execute o programa.