

## Faça como eu fiz

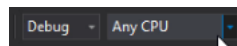
Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

1) O projeto inicial desde treinamento é o projeto implementado no treinamento anterior. Então, caso você tenha feito o treinamento anterior, pode prosseguir com o mesmo projeto. Caso você não tenha feito o treinamento anterior ou não tenha mais o projeto, você pode baixá-lo [aqui \(https://caelum-online-public.s3.amazonaws.com/1549-windows-forms-c-sharp-parte-2/01/CursoWindowsForms.zip\)](https://caelum-online-public.s3.amazonaws.com/1549-windows-forms-c-sharp-parte-2/01/CursoWindowsForms.zip).

2) Crie um diretório na sua máquina (por exemplo: `C:\CursoWindowsForms\Curso`), e descompacte o arquivo ZIP nele, baixado no passo anterior.

3) Vá para o sub-diretório `\CursoWindowsForms` e abra o arquivo com extensão `.sln`.

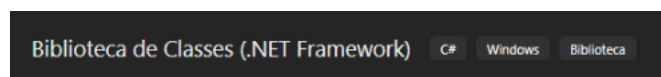
4) No barra de ferramentas do **Visual Studio**, modifique as propriedades do projeto para **Debug / Any CPU**:



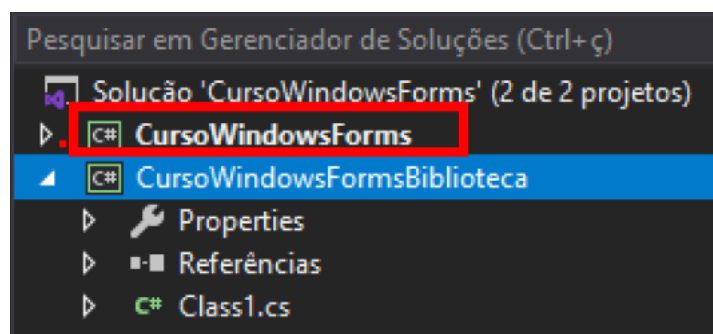
5) Execute o projeto e verifique se a aplicação do final do curso anterior está sendo executada.

6) Volte ao **Visual Studio** e adicione um novo projeto à sua solução. Para isso, no **Gerenciador de Soluções**, clique com o botão direito do mouse sobre o nome da solução e selecione **Adicionar --> Novo Projeto**.

7) Escolha um projeto do tipo **Biblioteca de Classes (.NET Framework)**:



8) Chame este projeto de `CursoWindowsFormsBiblioteca`:



9) No **Gerenciador de Soluções**, renomeie a classe padrão `Class1.cs` para `Cls_Uteis.cs`.

10) Abra a classe `Cls_Uteis.cs` e inclua as referências nesta classe:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

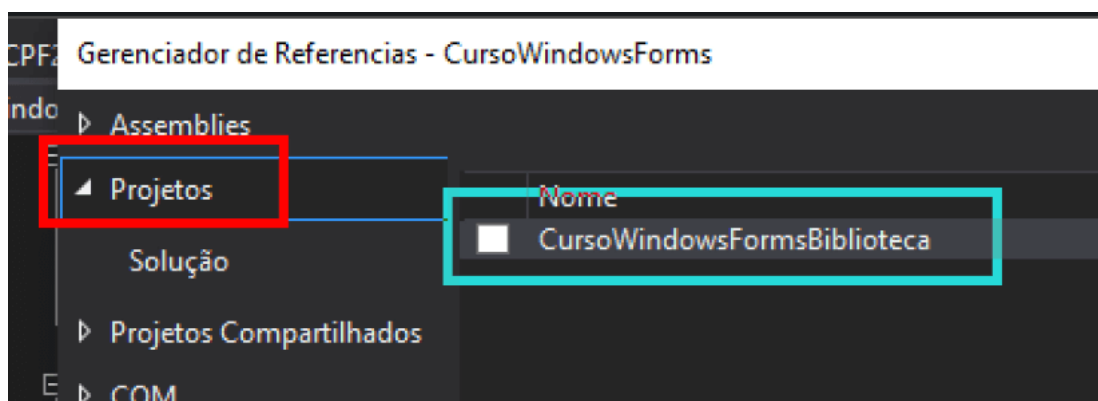
11) Copie o código que valida CPF, da classe original, do projeto principal, para a nova classe:

```
public static bool Valida(string cpf)
{
    int[] multiplicador1 = new int[9] { 10, 9, 8, 7, 6, 5, 4, 3, 2 };
    int[] multiplicador2 = new int[10] { 11, 10, 9, 8, 7, 6, 5, 4, 3, 2 };
    string tempCpf;
    string digito;
    int soma;
    int resto;
    cpf = cpf.Trim();
    cpf = cpf.Replace(".", "").Replace("-", "");
    if (cpf.Length != 11)
        return false;
    tempCpf = cpf.Substring(0, 9);
    soma = 0;
    for (int i = 0; i < 9; i++)
        soma += int.Parse(tempCpf[i].ToString()) * multiplicador1[i];
    resto = soma % 11;
    if (resto < 2)
        resto = 0;
    else
        resto = 11 - resto;
    digito = resto.ToString();
    tempCpf = tempCpf + digito;
    soma = 0;
    for (int i = 0; i < 10; i++)
        soma += int.Parse(tempCpf[i].ToString()) * multiplicador2[i];
    resto = soma % 11;
    if (resto < 2)
        resto = 0;
    else
        resto = 11 - resto;
    digito = digito + resto.ToString();
    return cpf.EndsWith(digito);
}
```

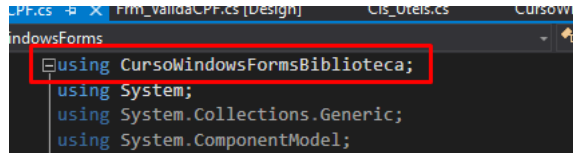
12) Remova a classe original **Cls\_Uteis.cs** do projeto principal.

13) No **Gerenciador de Soluções**, clique com o botão direito sobre **CursoWindowsForms** / **Referências** e selecione **Adicionar Referência**.

14) Escolha a seção **Projetos** e adicione como referência o projeto **CursoWindowsFormsBiblioteca**:



15) No código fonte dos formulários **Frm\_ValidaCPF** e **Frm\_ValidaCPF2**, adicione a referência o projeto **CursoWindowsFormsBiblioteca**:



16) Recompile a solução e teste a aplicação.

17) Vá no código fonte do formulário **Frm\_ValidaSenha** e mova a classe **ChecaForcaSenha** para a classe **Cls\_Uteis.cs**, no projeto **CursoWindowsFormsBiblioteca**:

```
public class ChecaForcaSenha
{
    public enum ForcaDaSenha
    {
        Inaceitavel,
        Fraca,
        Aceitavel,
        Forte,
        Segura
    }

    public int geraPontosSenha(string senha)
    {
        if (senha == null) return 0;
        int pontosPorTamanho = GetPontoPorTamanho(senha);
        int pontosPorMinusculas = GetPontoPorMinusculas(senha);
        int pontosPorMaiusculas = GetPontoPorMaiusculas(senha);
        int pontosPorDigitos = GetPontoPorDigitos(senha);
        int pontosPorSimbolos = GetPontoPorSimbolos(senha);
        int pontosPorRepeticao = GetPontoPorRepeticao(senha);
        return pontosPorTamanho + pontosPorMinusculas + pontosPorMaiusculas + pontosPorDigitos + pontosPorSimbolos + pontosPorRepeticao;
    }

    private int GetPontoPorTamanho(string senha)
    {
        return Math.Min(10, senha.Length) * 7;
    }

    private int GetPontoPorMinusculas(string senha)
    {
        int rawplacar = senha.Length - Regex.Replace(senha, "[a-z]", "").Length;
        return Math.Min(2, rawplacar) * 5;
    }

    private int GetPontoPorMaiusculas(string senha)
    {
        int rawplacar = senha.Length - Regex.Replace(senha, "[A-Z]", "").Length;
        return Math.Min(2, rawplacar) * 5;
    }

    private int GetPontoPorDigitos(string senha)
    {

```

```

    int rawplacar = senha.Length - Regex.Replace(senha, "[0-9]", "").Length;
    return Math.Min(2, rawplacar) * 6;
}

private int GetPontoPorSimbolos(string senha)
{
    int rawplacar = Regex.Replace(senha, "[a-zA-Z0-9]", "").Length;
    return Math.Min(2, rawplacar) * 5;
}

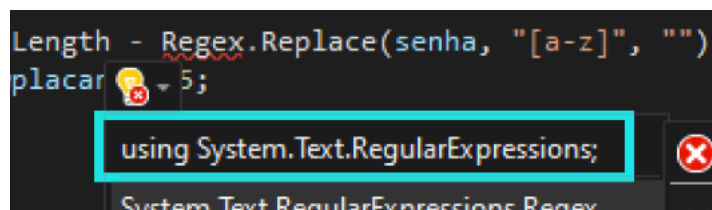
private int GetPontoPorRepeticao(string senha)
{
    System.Text.RegularExpressions.Regex regex = new System.Text.RegularExpressions.Regex('@');
    bool repete = regex.IsMatch(senha);
    if (repete)
    {
        return 30;
    }
    else
    {
        return 0;
    }
}

public ForcaDaSenha GetForcaDaSenha(string senha)
{
    int placar = geraPontosSenha(senha);

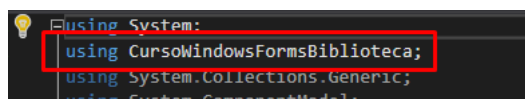
    if (placar < 50)
        return ForcaDaSenha.Inaceitavel;
    else if (placar < 60)
        return ForcaDaSenha.Fraca;
    else if (placar < 80)
        return ForcaDaSenha.Aceitavel;
    else if (placar < 100)
        return ForcaDaSenha.Forte;
    else
        return ForcaDaSenha.Segura;
}
}

```

18) Ainda na classe **Cls\_Uteis.cs**, adicione a referência à classe que manipula expressões regulares:



19) No formulário **Frm\_ValidaSenha**, adicione a referência à **CursoWindowsFormsBiblioteca**:



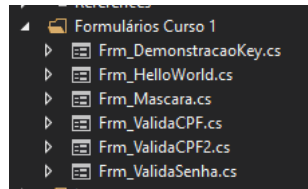
20) Mude a forma com que a classe de validação de senha é chamada, acrescentando o nome da classe da biblioteca:

```
Cls_Uteis.ChecaForcaSenha verifica = new Cls_Uteis.ChecaForcaSenha();  
Cls_Uteis.ChecaForcaSenha ForcaDaSenha forca;  
forca = verifica.GetForcaDaSenha(Txt_Senha.Text);  
Lbl_Resultado.Text = forca.ToString();
```

21) Salve os arquivos, recompile a solução e teste a aplicação.

22) No projeto **CursoWindowsForms** adicione uma nova pasta. Chame esta pasta de **Formulários Curso 1**.

23) Copie os formulários feitos no curso anterior para dentro desta pasta, conforme listados abaixo:



24) Salve os arquivos, recompile a solução e teste a aplicação.