

Componentes Restulizáveis

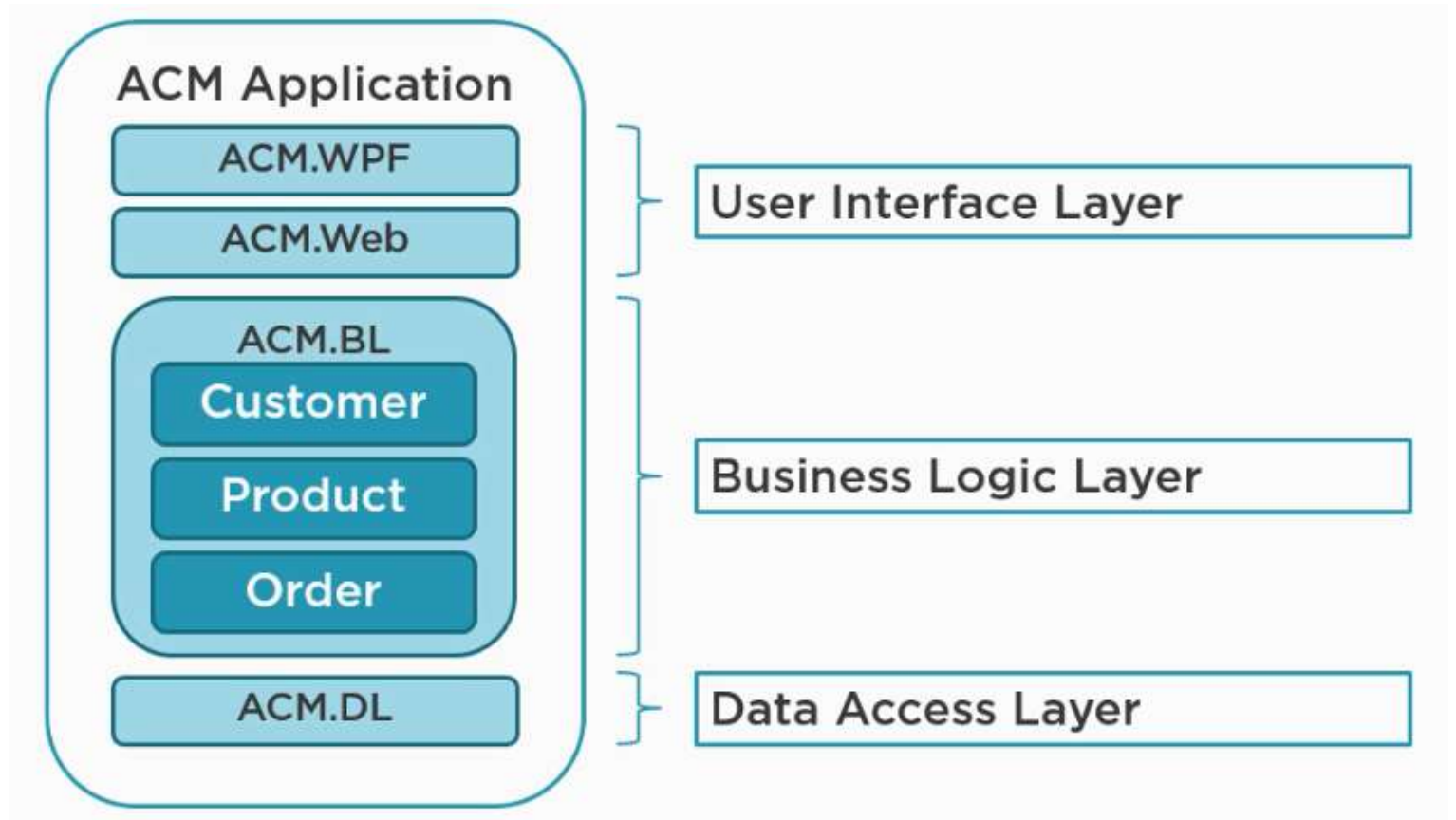
UNOESC

Prof. Mauricio Roberto Gonzatto

Componentes

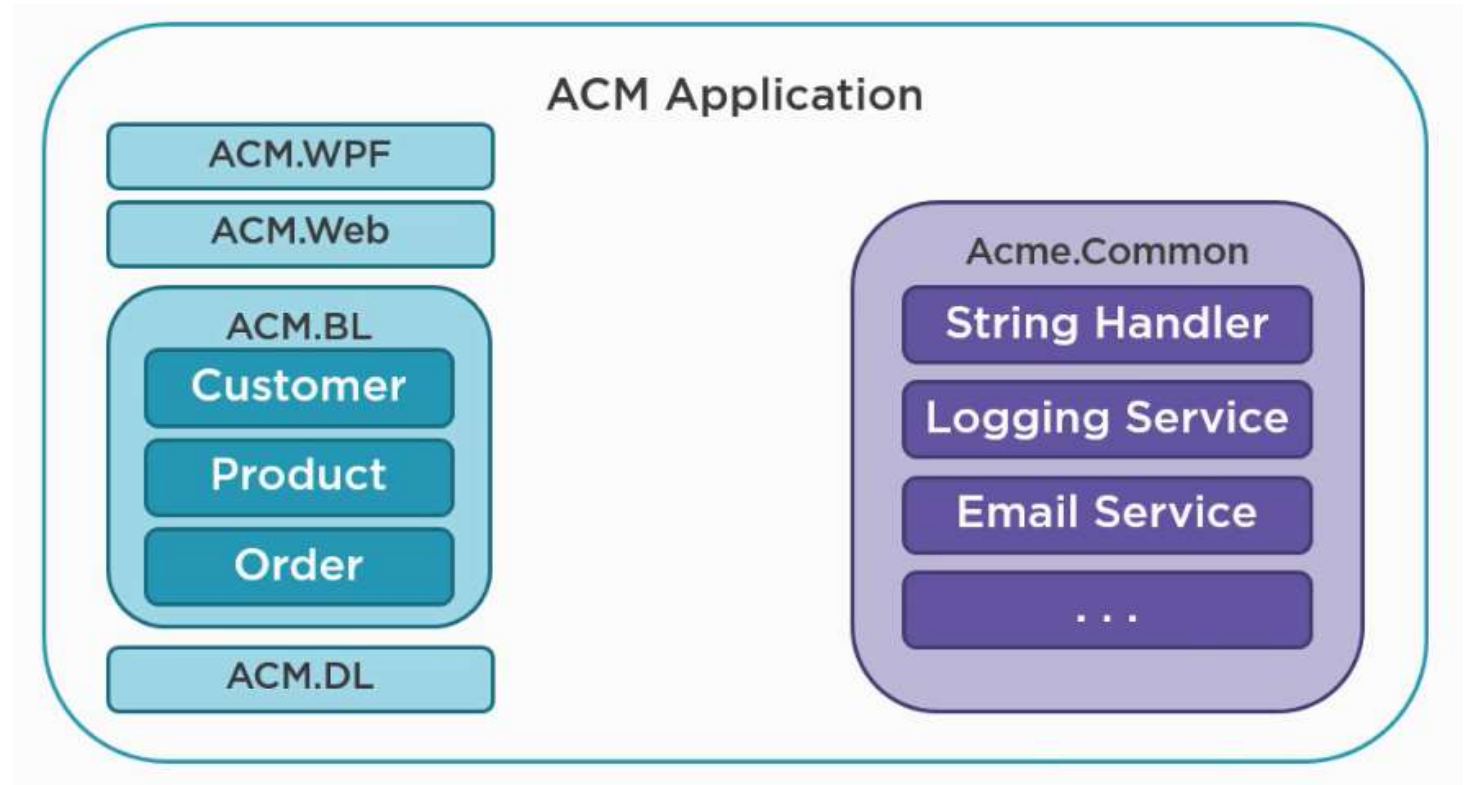
- Aplicações C# são construídas com um ou mais componentes e a POO define e constrói as classes de cada componente;
- Podemos construir componentes para Interfaces de Usuário (UI) como Desktop, Web ou Mobile;
- Componentes da Camada de Negócios (.BL);
- Componentes de Camada de Acesso a Dados (Repository, ORMs, DataBase)
- Cada camada da arquitetura da aplicação deve funcionar como um componente reutilizável

Componentes



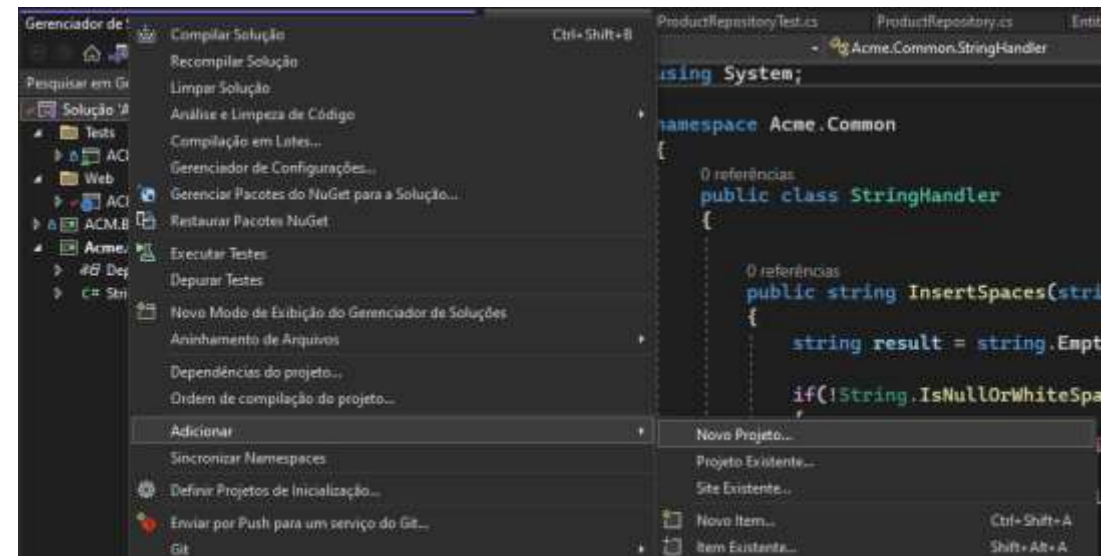
Biblioteca de Componentes Reutilizáveis

- Vamos construir uma nova biblioteca de componentes que será utilizada como serviços por nossas camadas da aplicação que poderia ser reutilizada em qualquer outra aplicação.



Crie uma Biblioteca de Componentes

- Abra o Gerenciador de Soluções
- Clique com o botão direito na sua Solução
- Menu Adicionar -> Novo Projeto
- Escolha Biblioteca de Classes
- Defina o nome como Acme.Common
- Escolha a versão do Framework que está utilizando (Core 3.1 ou .NET 6)



```

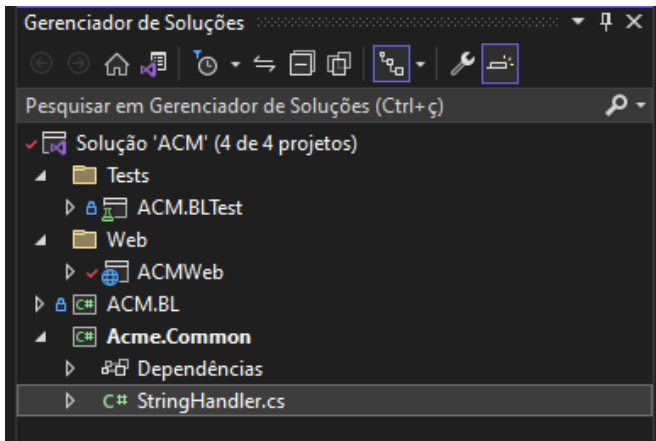
using System;

namespace Acme.Common
{
    0 referências
    public class StringHandler
    {
        0 referências
        public string InsertSpaces(string source)
        {
            string result = string.Empty;
            if(!String.IsNullOrEmpty(source))
                foreach ( char letter in source )
                {
                    if ( char.IsUpper(letter) )
                        result += " ";
                    result += letter;
                }
            return result;
        }
    }
}

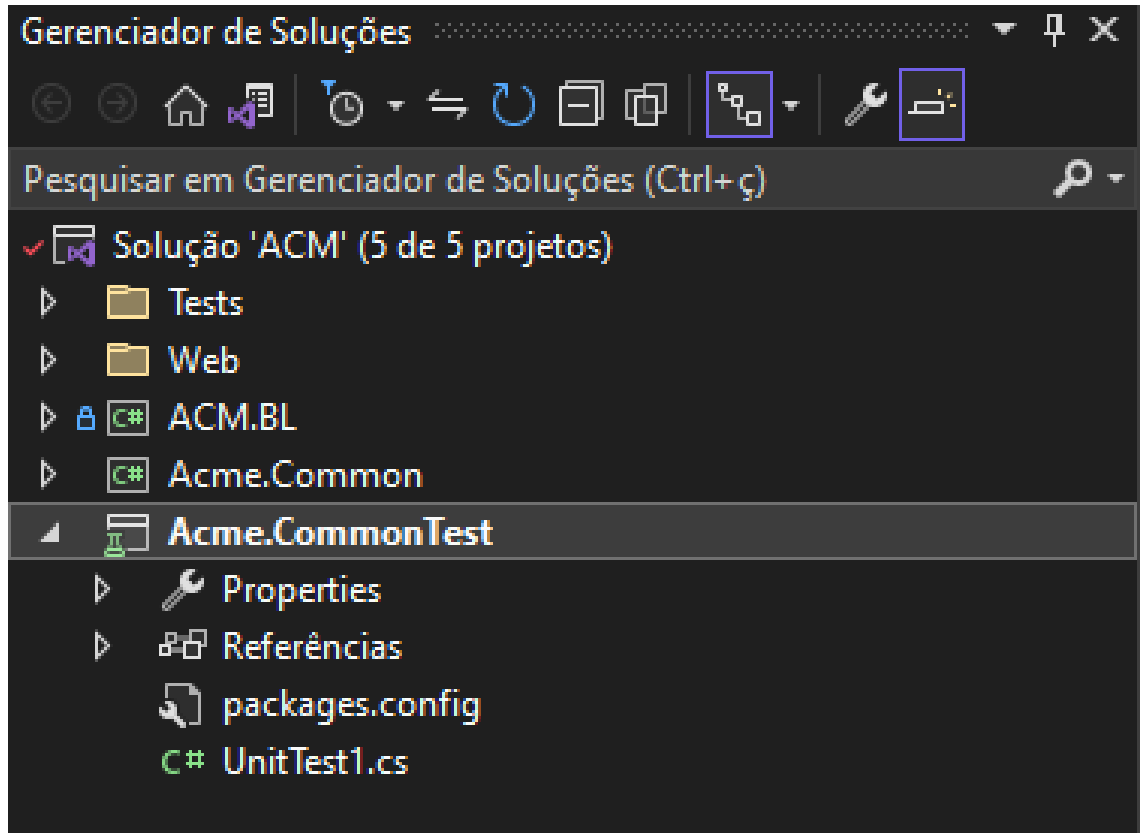
```

Crie uma Biblioteca de Componentes

- Renomeie a o arquivo Class1.cs padrão para StringHandler.cs
- Defina a nova classe como public e implemente o método InsertSpaces()



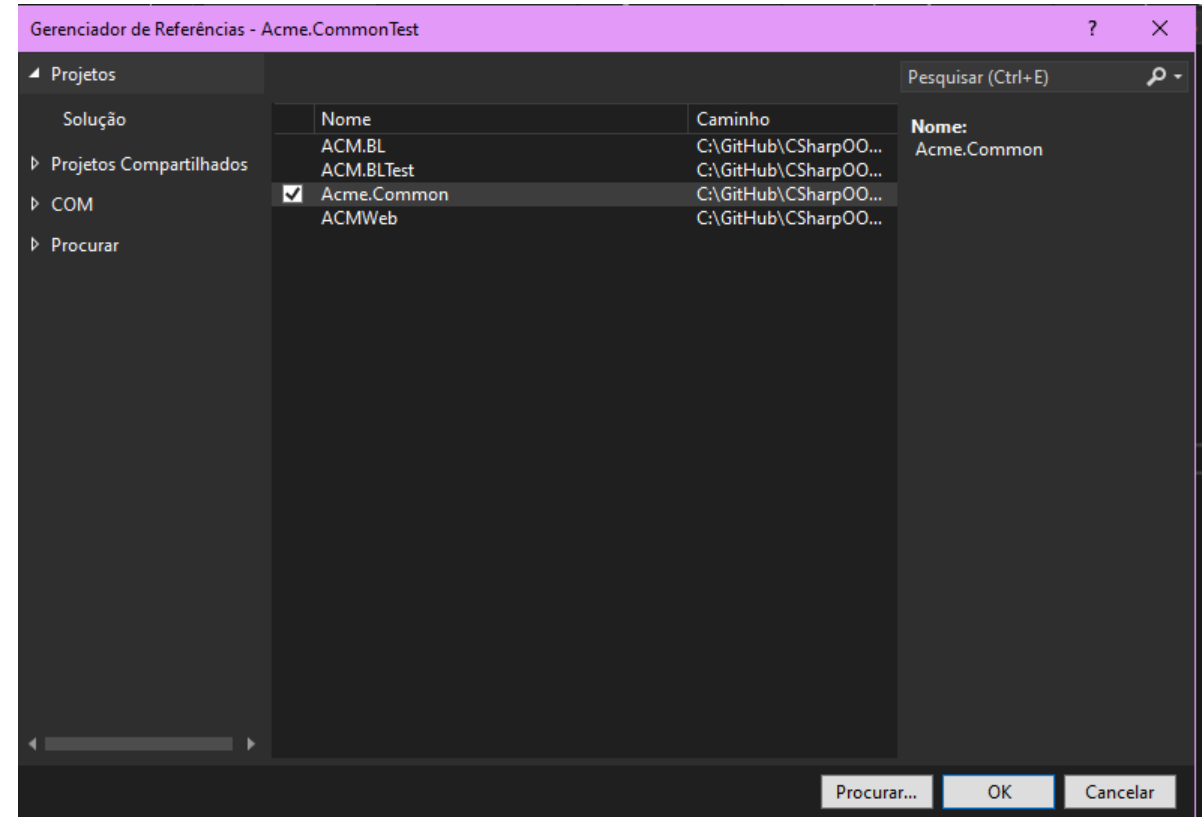
Testando Componentes Reutilizáveis



- Agora vamos criar um novo Projeto de Teste para esta Biblioteca de Componentes;
- Abra o Gerenciador de Soluções
- Clique com o botão direito no diretório Tests
- Menu Adicionar -> Novo Projeto
- Selecione Projeto de Teste de Unidade, se estiver utilizando o .NET Core selecione Projeto de Teste MSTest;
- Nomeie como Acme.CommonTest
- Defina a versão do framework que está utilizando

Testando Componentes Reutilizáveis

- Adicione uma referência a biblioteca de componentes Acme.Common em seu novo projeto de testes;
- Abra o Gerenciador de Soluções
- Tests -> Acme.CommonTest -> Dependências
- Clique com o botão direito
- Selecione Adicionar Referência de Projeto
- Selecione Acme.Common e clique OK



Testando Componentes Reutilizáveis

- Abra o Gerenciador de Soluções
- Navegue até seu novo projeto de testes
- Renomeie o arquivo padrão UnitTest1.cs do seu novo projeto de teste para StringHandlerTest.cs
- E implemente o teste!

```
using Acme.Common;
using Microsoft.VisualStudio.TestTools.UnitTesting;

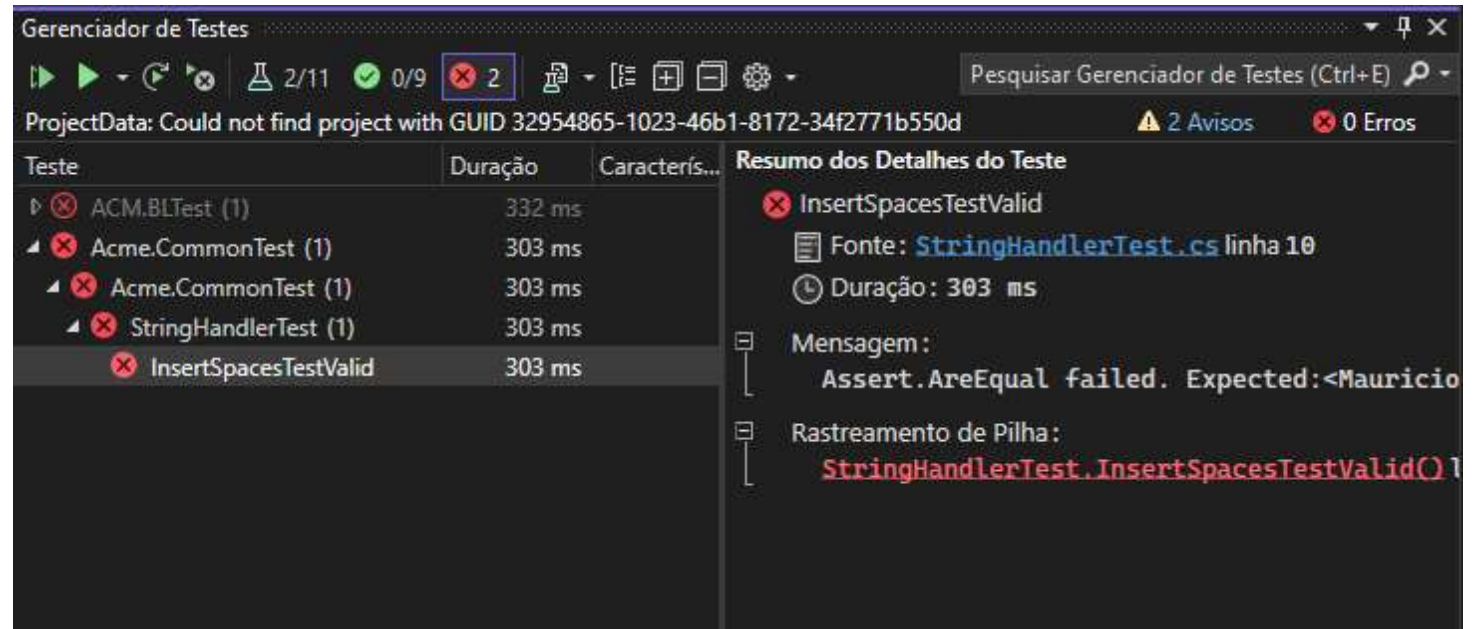
namespace Acme.CommonTest
{
    [TestClass]
    0 referências
    public class StringHandlerTest
    {
        [TestMethod]
        0 referências
        public void InsertSpacesTestValid()
        {
            // Arrange
            var source = "MauricioGonzatto";
            var expected = "Mauricio Gonzatto";
            var handler = new StringHandler();

            // Act
            var actual = handler.InsertSpaces(source);

            // Assert
            Assert.AreEqual(expected, actual);
        }
    }
}
```

Testando Componentes Reutilizáveis

- Agora execute o teste e observe que ele irá falhar!
- Isto por que o método InsertSpaces não está operando corretamente.



Testando Componentes Reutilizáveis

- Altere o método InsertSpaces() no StringHandler para ele tratar os espaços em branco e depois execute o teste novamente.

```
1 referência
public class StringHandler
{
    1 referência | 0/1 passando
    public string InsertSpaces(string source)
    {
        string result = string.Empty;
        if(!String.IsNullOrEmpty(source))
            foreach ( char letter in source )
            {
                if ( char.IsUpper(letter) )
                {
                    result += " ";
                    result += letter;
                }
            }
        result = result.Trim();
        return result;
    }
}
```

Gerenciador de Testes

10/11 10 0/1 Pesquisar Gerenciador de Testes (Ctrl+E)

Execução de teste concluída: 1 Testes (1 Aprovados, 0 Com falha, 0 Ignorados) executados em 262 ms 0 Avisos 0 Erros

Teste	Duração	Caracterís...
ACM.BLTest (9)	134 ms	
Acme.CommonTest (1)	62 ms	
Acme.CommonTest (1)	62 ms	
StringHandlerTest (1)	62 ms	
InsertSpacesTestValid	62 ms	

Resumo dos Detalhes do Teste

- InsertSpacesTestValid
- Fonte: [StringHandlerTest.cs](#) linha 10
- Duração: 62 ms

