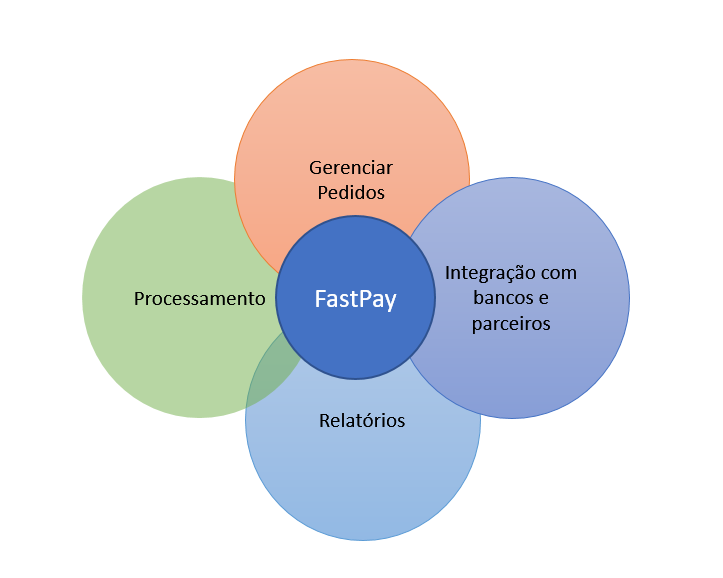
Windows Comunication Foundation

O WCF como é conhecimento foi desenvolvido pela microsoft para unificar as outras alternativas mantidas pela própria empresa para desenvolvimento de sistemas distribuídos.

**E o que é um sistema distribuído?**

E antes de responder vamos ao conteito básico de sistemas( mesmo que você saiba é bom relembrar)

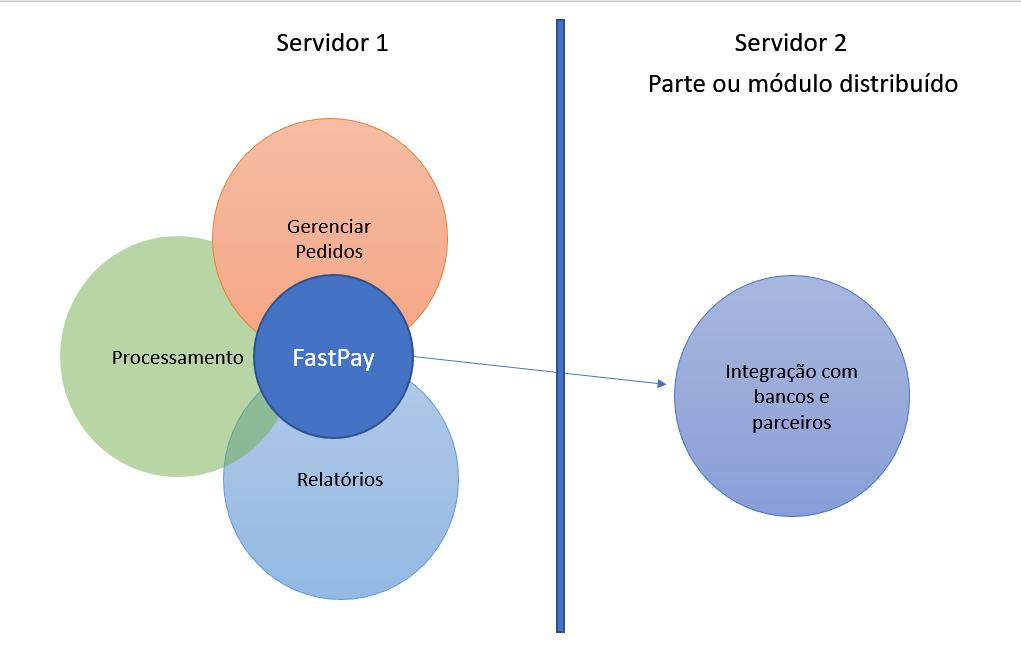
**Um sistema é formado por diferentes partes** que atuam de forma **independente**, no entanto **contribuindo** para resolver um problema em comum.



As **partes** do sistema em um ambiente não distribúido estão dentro da mesma solução e se tratando de um sistema WEB, todas as partes são publicadas juntas.

No caso de um **sistema distribuído** uma ou algumas partes podem ser distribuídas separadamente.

E como em um sistema, as partes funcionam de forma **independente**, a **parte ditribuída** **pode ser liberada e um servidor completamente diferente** das partes que permanecem juntas



No **lado direito**, local de instalação da parte do sistema que foi **distribuído** ainda é necessário ter todas as referências que existem no **lado esquerdo**.

Porque embora uma parte não dependa diretamente da outra para compilar e executar...

É importante lembrar a razão de cada parte existir dentro do sistema que é resolver um problema comum.

Então a diferença que tem na publicação do sistema como todo no lado esquerdo(**servidor1**) com o lado direito(**servidor 2 – módulo distribuído**) é que...

* No **servidor 1** o sistema é publicado com praticamente todo o sistema com cadastro, telas, relatórios etc e etc.
* No **servidor 2**, é liberado **SOMENTE** a parte do sistema que realiza integração com parceiros e bancos ( instuição financeira)

Agora que já foi apresentado o **absolutamente** básico sobre sistema distribuído, eu começo aqui a explicar sobre o WCF.

Antes do WCF existiram diferentes soluções para construção de sistemas distribuídos para a **plataforma .NET** e o que a microsoft fez foi unificar todas as tecnologias internas em uma que foi nomeada de **Windows Comunication Foundation**

**Se o que você está procurando é** um documento mais abrangante ( e não necessariamente mais fácil) você pode recorrer a documentação oficial no site da microsoft( Link no final do doc).

Resumidamente através do **WCF** é possível criar sistemas baseados em...

**Serviços** **distribuídos** utilizando comunicação baseada em mensagem e usufrindo de todo o potencial do protocolo HTTP.

Fundamentalmente o WCF é composto por:

1. Mensagem
2. EndPoint
3. Soap

1 **Mensagem** – Um serviço WCF **recebe** e **responde** as requisições no formato de um **evenlope** e dentro de cada envelope existem as **mensagens**

2 **EndPoint** – É utilizado para definir informações de acesso e como os serviços podem ser consumidos

O **EndPoint** é composto de 3 elementos importantes como:

* Address – Endereço do serviço publicado
* Binding – Define como será a conexão com o serviço
* Contract – Aqui é para informar que função do nosso sistema está sendo exposto pelo WCF

3 **SOAP** que é o mesmo que Simple Object Access Protocol e que é o protocolo baseado em **XML** que é utilizando pelo WCF para comunicação.

Serviços em WCF utilizam este protcolo mas a comunicação é realizada com o uso do protocolo HTTP

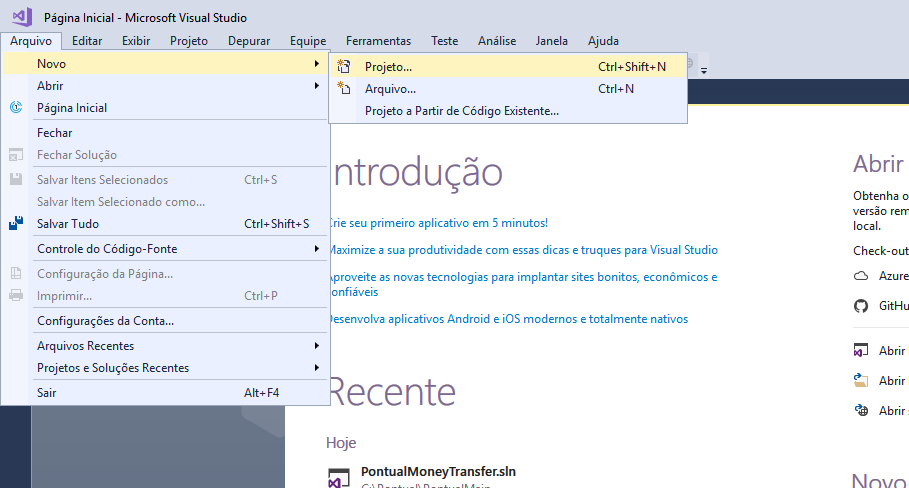
**E o que é necessário para criação de um serviço WCF ?**

1. .NET Framework 3.0 ou superior
2. Visual Studio( a versão mais recente por favor 😊 )

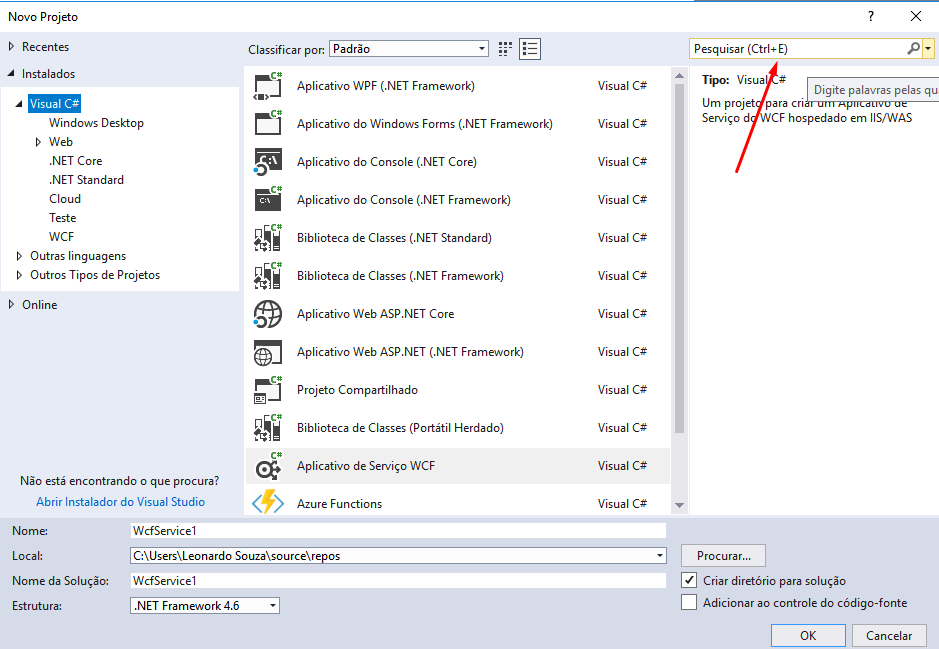
**E como faço para criar meu primeiro serviço WCF?**

Para mostrar como... eu vou utilizar o visual Studio 2017.

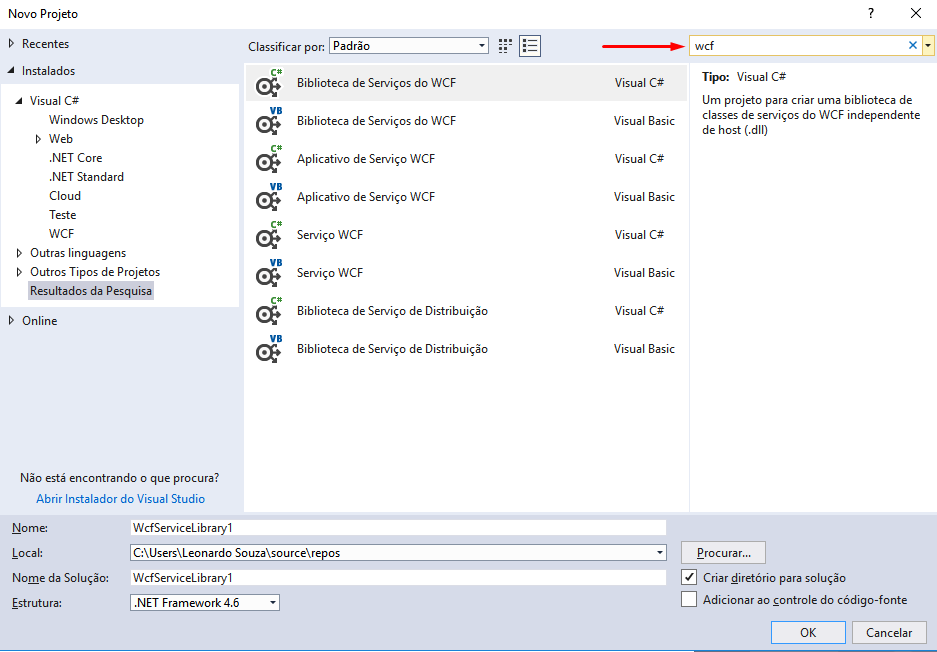
Começo primeiramente criando um projeto de **Service Application** e para começar, clique em Arquivo - > Novo - Projeto



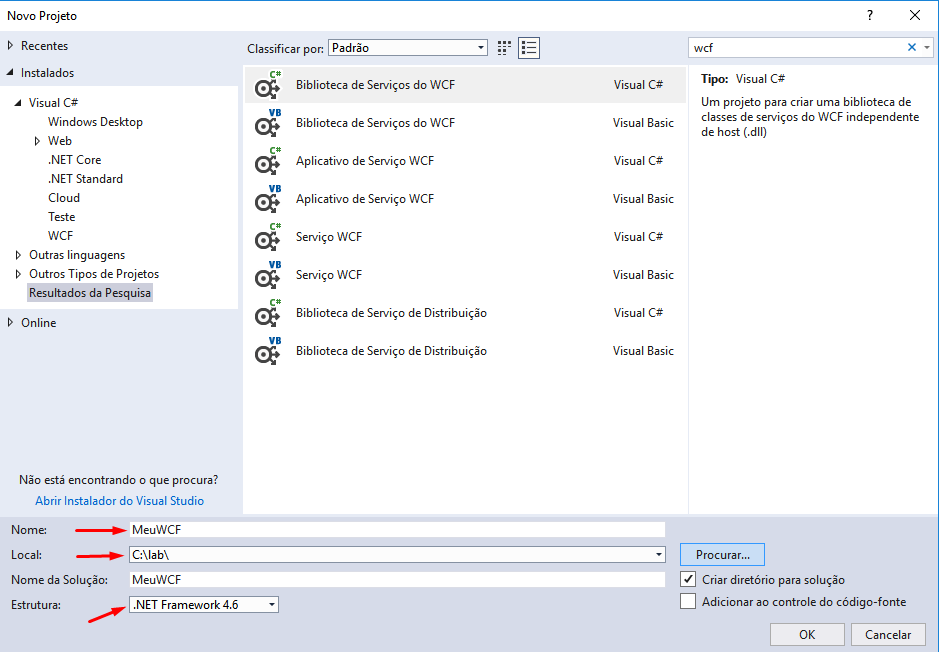
Escolha o template para criação de Aplicativo de Serviço Web ou WCF Service Application



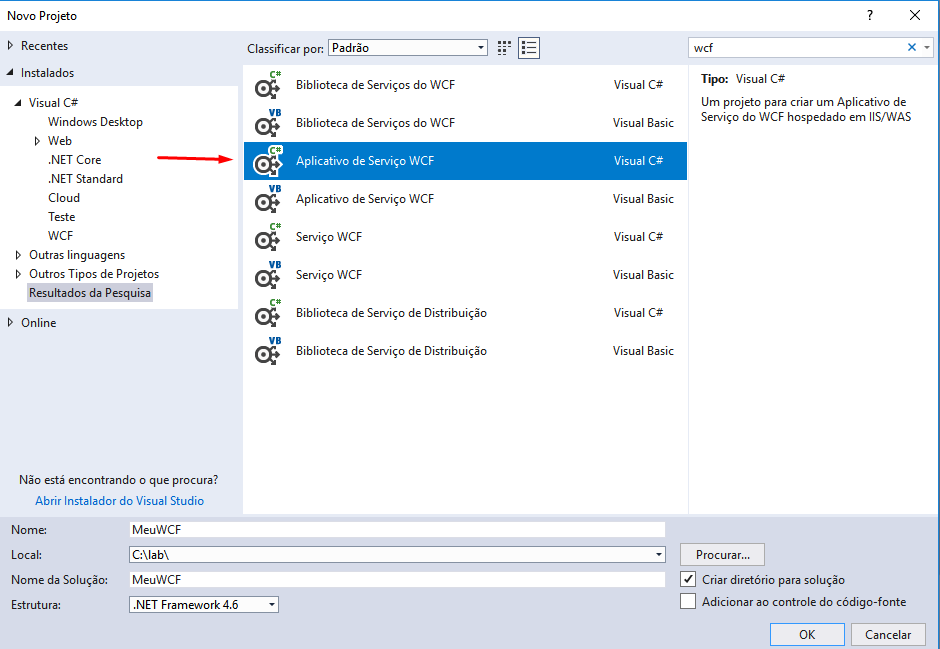
Caso tenha dificuldade de encontrar, clique( conforme tela acima) na área de pesquisa de templates e digite **wcf** conforme o print abaixo



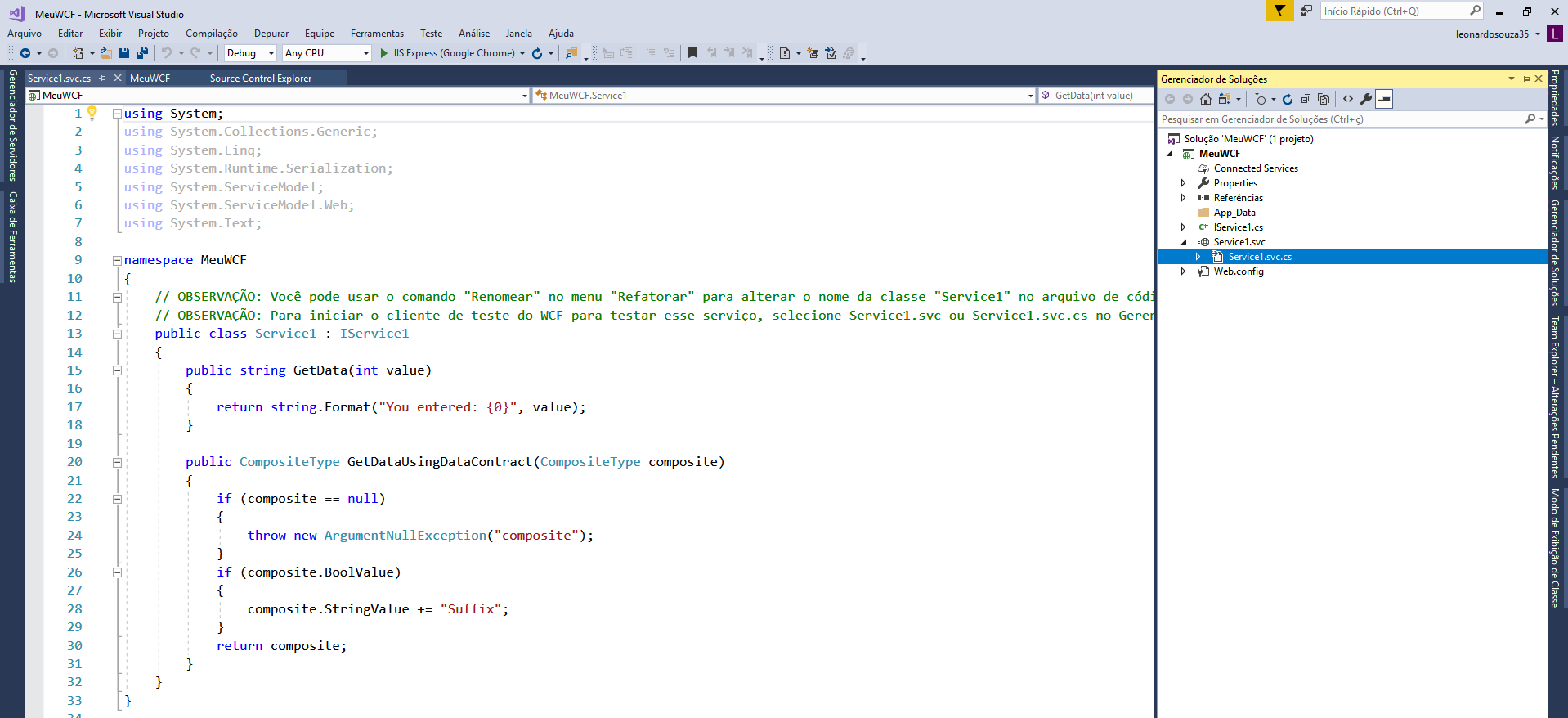
Mude o nome do projeto para **MeuWCF,** defina uma nova pasta para salvar seu projeto e mantenha o .net framework 4.6 conforme o print abaixo



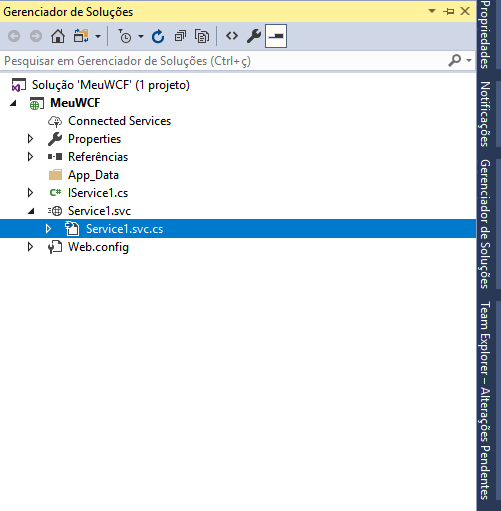
E antes de clicar em OK, verifique se o template selecionado está apontando para um template do C#. Só conferir se o ícone é do C# em vez do VB.



E agora sim.. clique no botão OK para criação do projeto de WCF

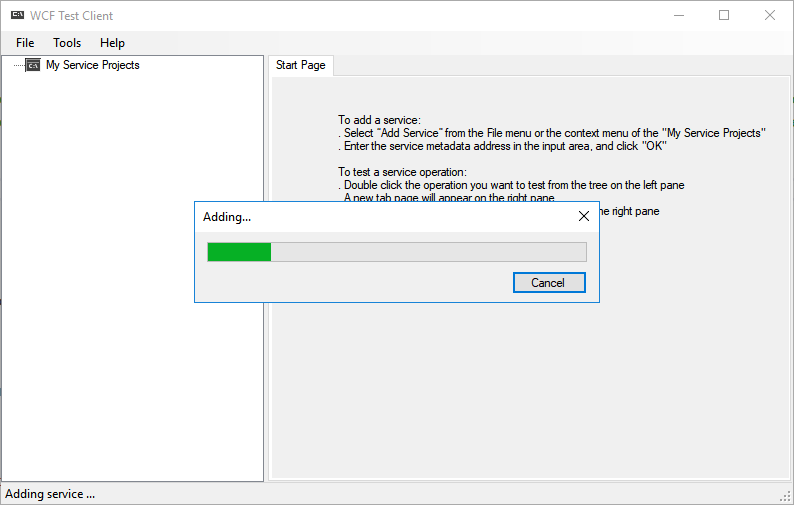


E aqui é o gerenciador de soluções aberto no projeto de WCF

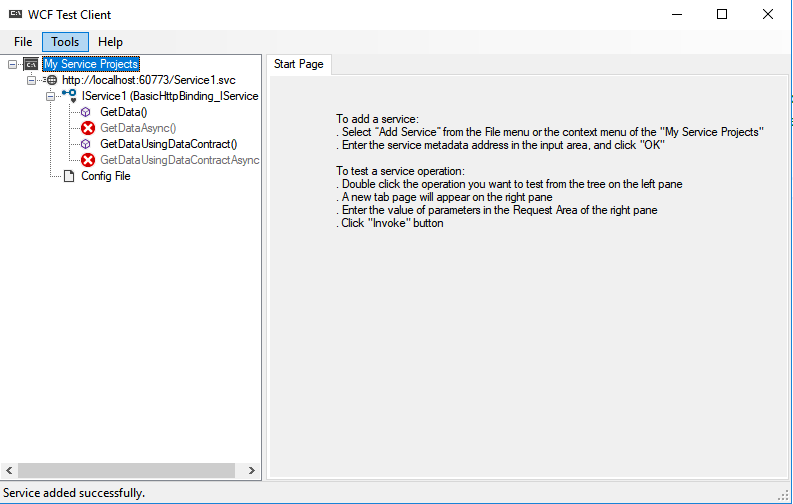


E exatamente da forma como está digite a tecla F5 para compilar e executar o serviço

E após a compilação, o **WCF Test Client** será executado para realização dos testes do seu serviço

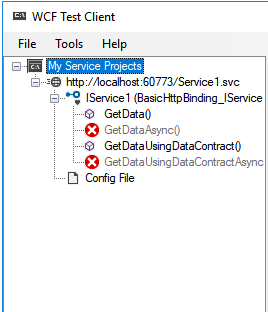


E aqui abaixo um print com **detalhes do serviço** e com a opção de chamar alguns dos métodos que foram publicados



Dentro da pasta **My Service Projects** aparece o nosso serviço que foi publicado no IIS Express que por sua vez foi iniciado pelo Visual Studio 2017

Observe um pouco mais de perto a pasta **My Service Projects**

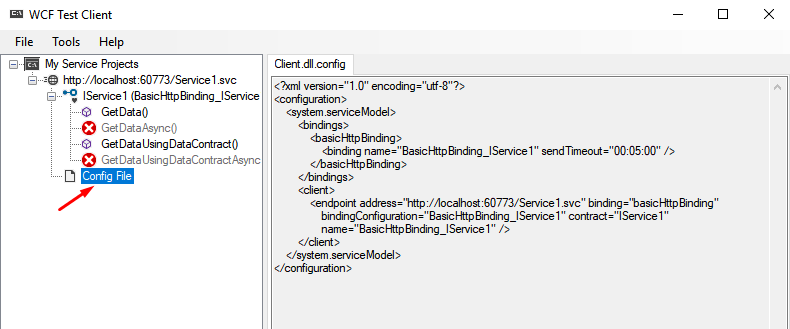


E veja que nosso serviço está publicado no endereço <http://localhost:60773/Service1.svc>

E os métodos estão organizados dentro de **IService(BasicHttpBinding\_IService)**

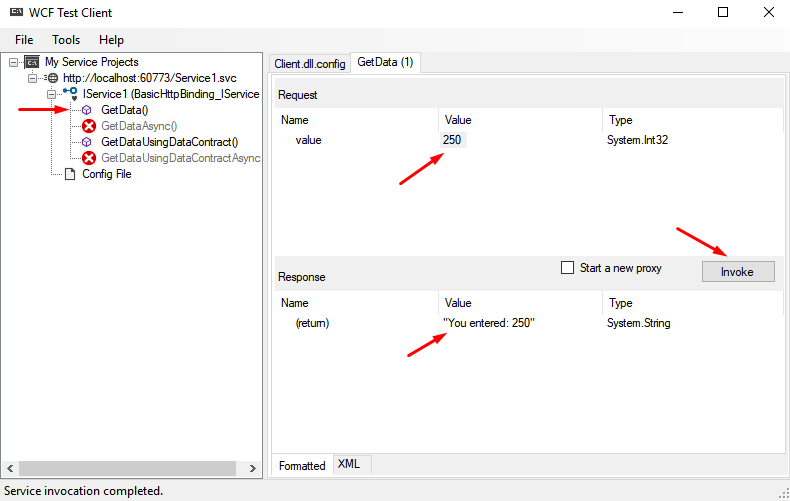
Onde **IService** é o contrato e o **BasicHttpBinding** representa a forma como o serviço será acessado.

E por último, clicando em Config File é possível ver o **EndPoint** utilizado pelo próprio **WCF Test Client** para acessar o serviço publicado em <http://localhost:60773/Service1.svc>

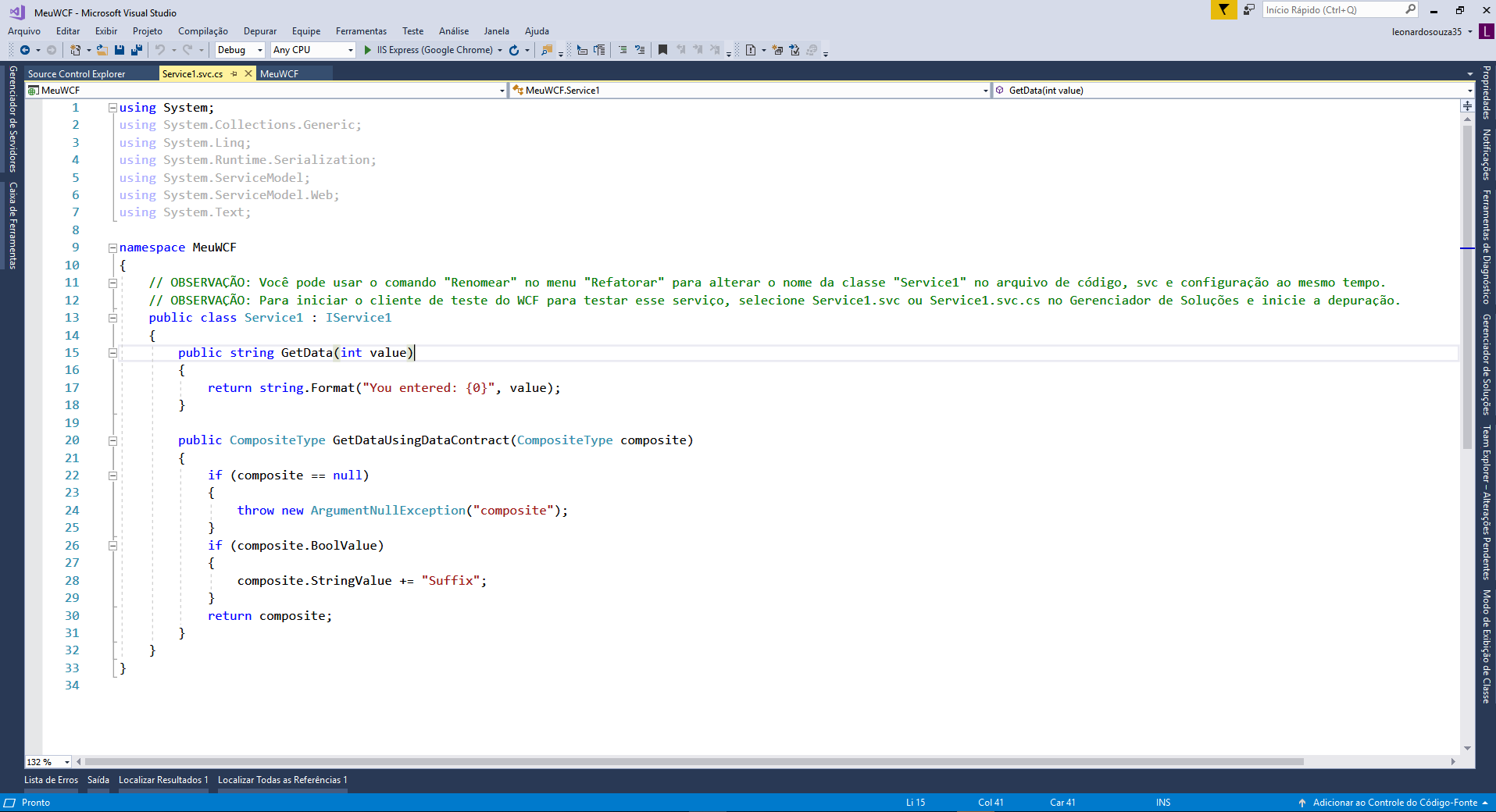


Para testar se de fato o serviço está funcionando, clique duas vezes no método **GetData(),** digite um valor no campo Value que esta na aba **Request(**no exemplo está 250)

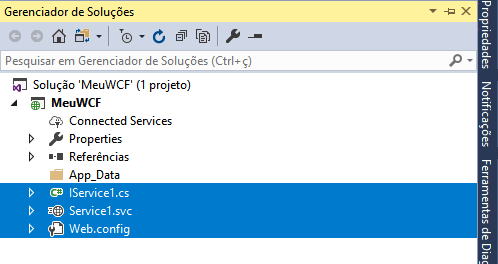
Clique no botão **Invoke**  e o resultado vai aparecer na aba **Response** que no exemplo foi exibida a mensagem “You entered 250”



Agora que ja foi apresentado como é o serviço rodando, é hora de fechar o **WCT Test Client** para voltarmos para o Visual Studio e observar o código que foi gerado para construir o serviço WCF.



Voltando para o gerenciador de soluções, vamos analisar os arquivos criados no momento em que você escolheu criar um **aplicativo de serviço WCF**

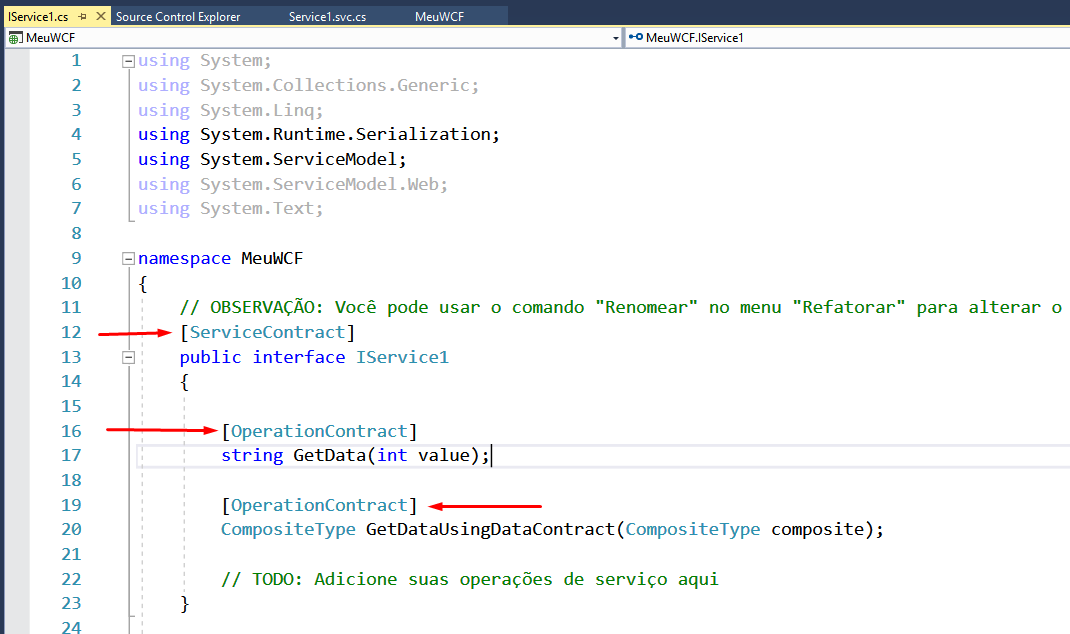


Quando criamos um projeto de WCF, os seguintes arquivos são criados:

* IService.cs
* Service1.svc
* Web.config

**IService.cs**

Dentro deste arquivo de C#, temos a peça importante na construção de um WCF que é a interface com os atributos utilizados pelo próprio WCF.

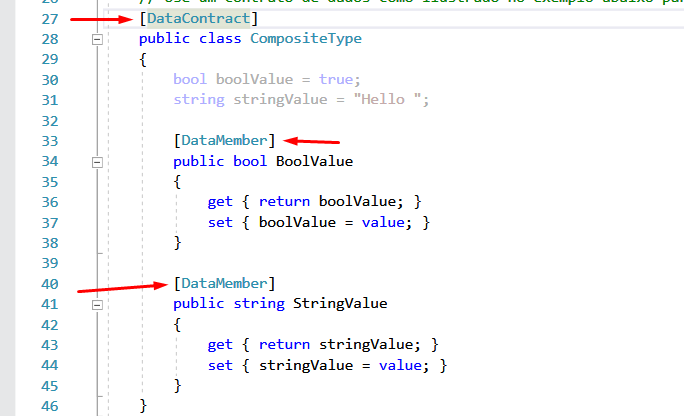


**E que atributos são estes ?**

Os que estão selecionados na imagem acima: **ServiceContract** atributo para colocar acima do nome da interface e **OperationContract** que deve ser colocado logo acima de cada assinatura(método) da interface

E cada atributo deve ser adicionado entre colchetes []

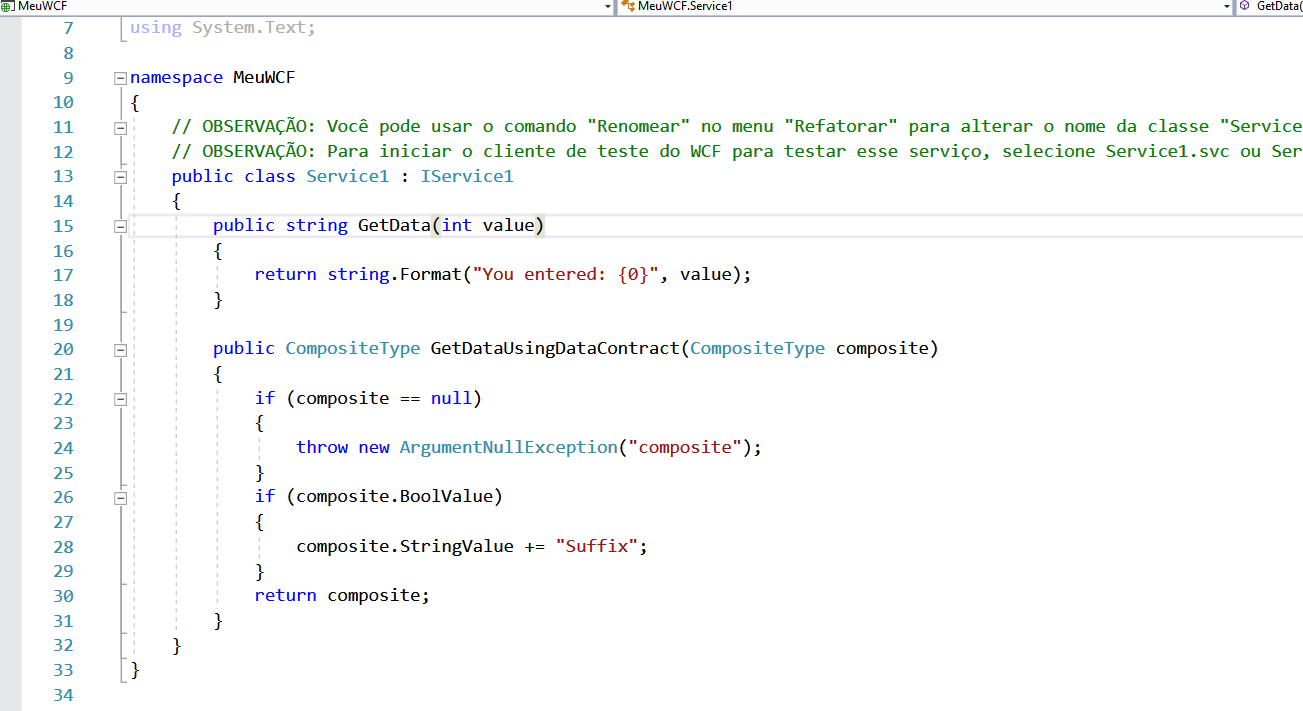
Observe que no método **GetDataUsingDataContract..** é retornado um **CompositeType** que poderia ser sua própria classe desde que tenha atributos que serão utilizados pelo WCF



Então se você deseja que sua classe seja retornada pelo WCF, ela deve conter o atributos de **DataContract** e **DataMember** da mesma forma que a classe **CompositeType**

**Service1.svc**

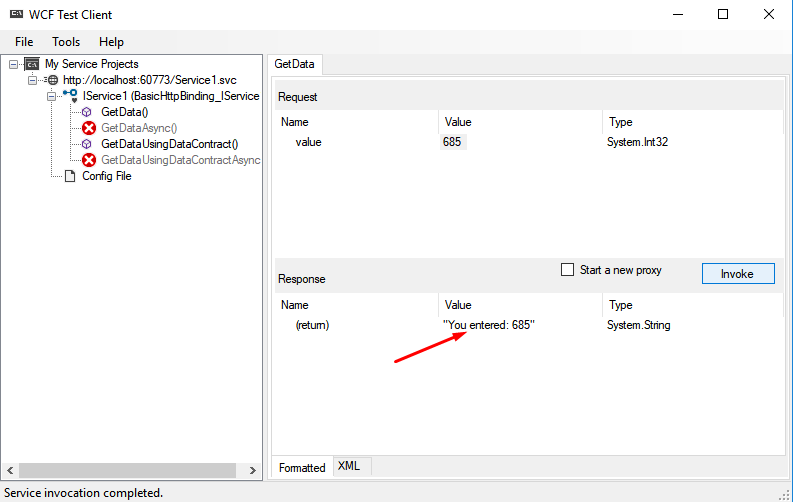
Essa classe implementa o contrato **IService1 e** nela contém a implementação de fato do serviço (Print do código abaixo)



Observe que o retorno do método GetData(int value), que é a mensagem...

“You entered: {0}...

É exibido no resultado da chamada ao serviço pelo **WCF Test Client** ( Print da tela abaixo)



**Web.Config**

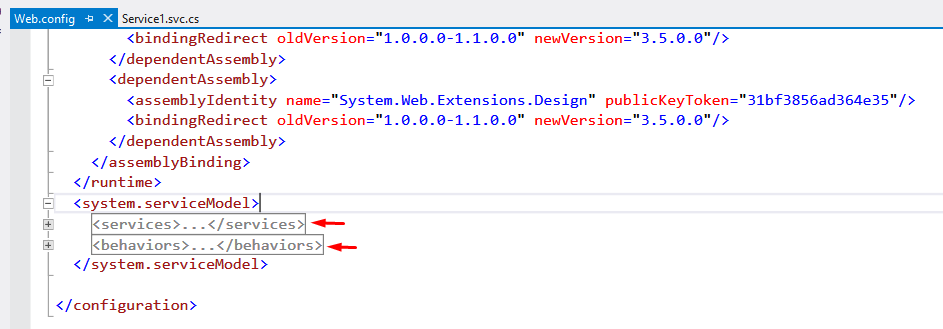
Quando um novo WCF é adicionado ao projeto, o Visual Studio( **dependendo da versão**) configura o EndPoint que é necessário para acessar o serviço

No Visual Studio 2012, a seguinte tag é adicionada ao web config

<system.serviceModel>

</system.serviceModel>

E dentro da tag é adicionada as tags **<services>** e **<behaviors>**



Dentro de services é definido o nome do serviço, referência e a configuração de behavior ( behavior siginifa comportamente)

E dentro de **services** tem informações sobre o ponto de acesso ao nosso serviço que o End Point que é configurado informando...

* **Address** – Endereço do seu serviço ( se estiver vazio sera definido automaticamente)
* **Binding** – Qual protocolo é necessário para acessar o serviço
* **Contract** – O nome completo incluindo o namespace da interface que esta sendo exposta.. ex.: MeuWCF.IService1

E dentro da tag **behaviors** existem informações relacionadas a forma como o seu serviço vai responder.

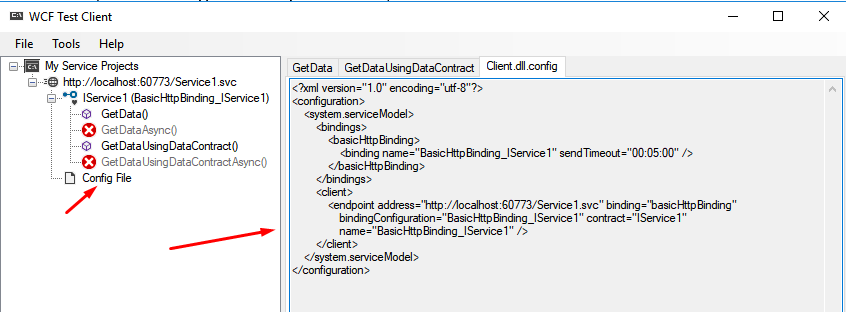
**Não é tarefa fácil** definir como montar uma configuração para o seu serviço e em muitos casos é necessário visitar as páginas oficiais da microsoft para dominar o assunto.

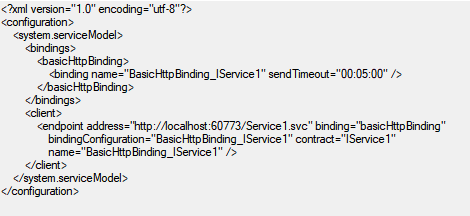
**Atualmente não chega a ser uma dor de cabeça** porque existem formas de acessar serviços WCF utilizando configuração gerada de forma automática ou melhor ainda...

É possível acessar um serviço de WCF sem utilizar seu EndPoint.

**Lembra da tela do WCF Test Client ??**

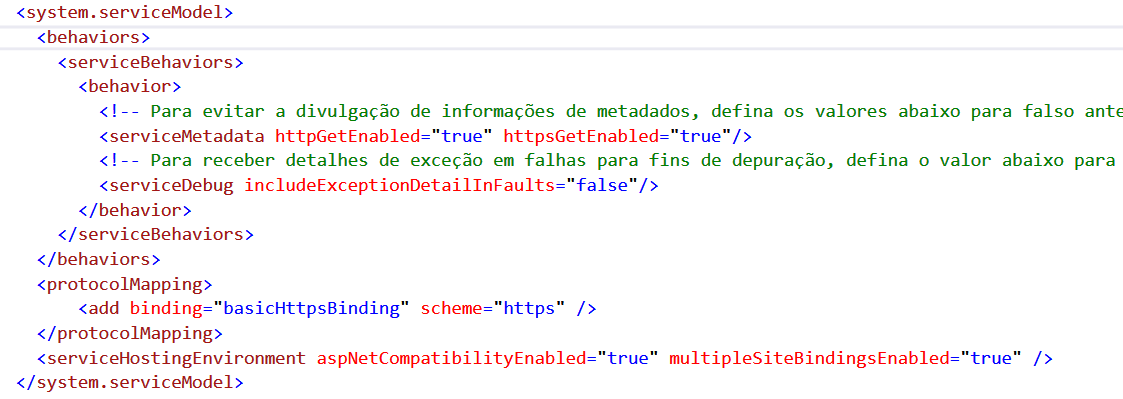
Então... existe na tela o item Config File que exibe as configurações de end point do nosso serviço





O **WCF Test Client** através do caminho que foi informado pelo próprio Visual Studio na hora da compilação e execução (<http://localhost:60773/Service1.svc>), fez download das informações do serviço e gerou o arquivo de configuração necessário para realizar o acesso.

No **Visual Studi 2017** a parte do web.config não entra muita informação.. Somente algumas informações essenciais conforme o print abaixo



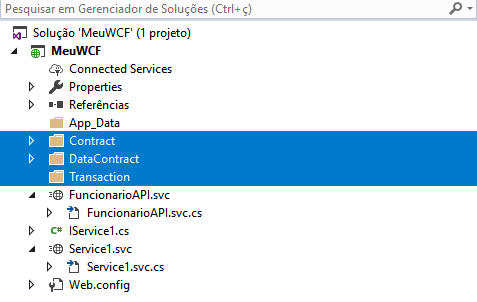
Não tente memorizar essas informações para evitar o risco de uma overdose por excesso de informação até porque sabe lá quando essas configurações mudam.

Até esse ponto foi demonstrado somente o que o Visual Studio gera automaticamente e na próxima seção, você vai aprender como criar seu próprio serviço WCF

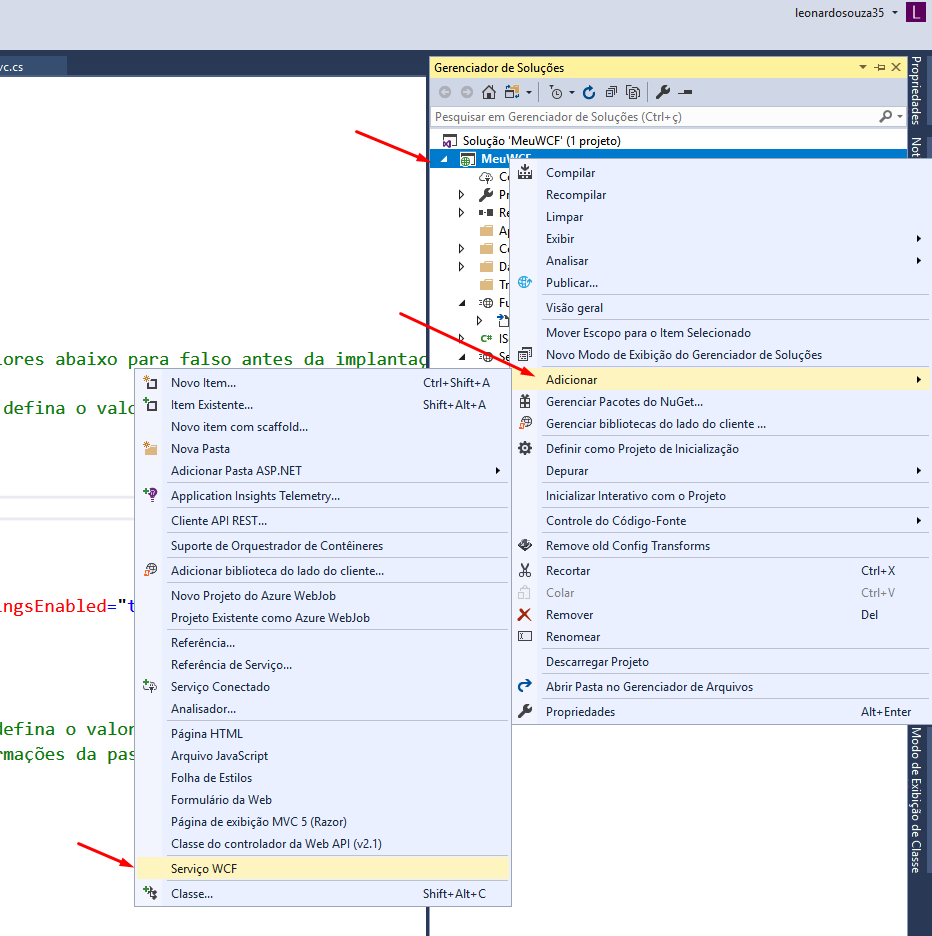
Criando um Serviço WCF

Antes de criar seu primeiro serviço, vamos adicionar algumas pastas na nossa solução

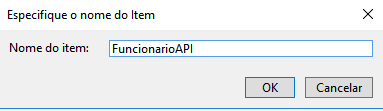
* Contract
* DataContract
* Transaction



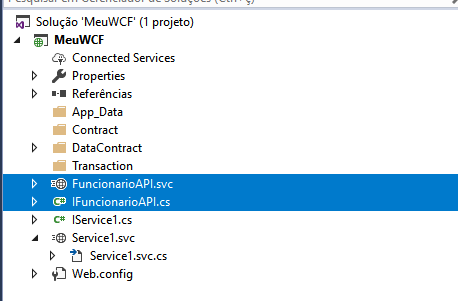
Agora adicione na raiz do projeto um novo serviço de WCF clicando com botão direito no projeto, adicionar e escolha **Serviço WCF** ou **WCF Service**



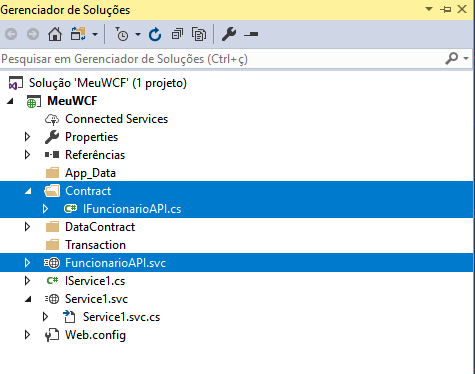
Escolha o nome para o serviço de **FuncionarioAPI**



Agora existem mais doi novos arquivos... que é a classe e o contrato(interface)

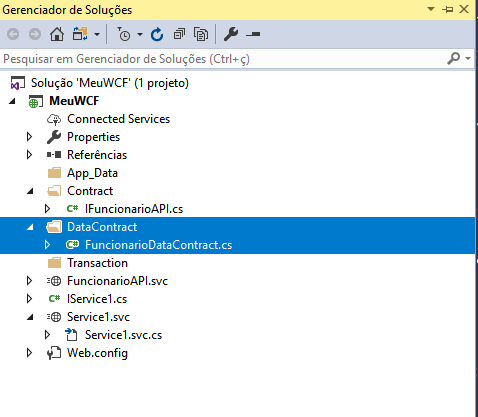


E agora mova a interface IFuncinarioAPI para a pasta Contract para deixas o projeto um pouco mais organizado

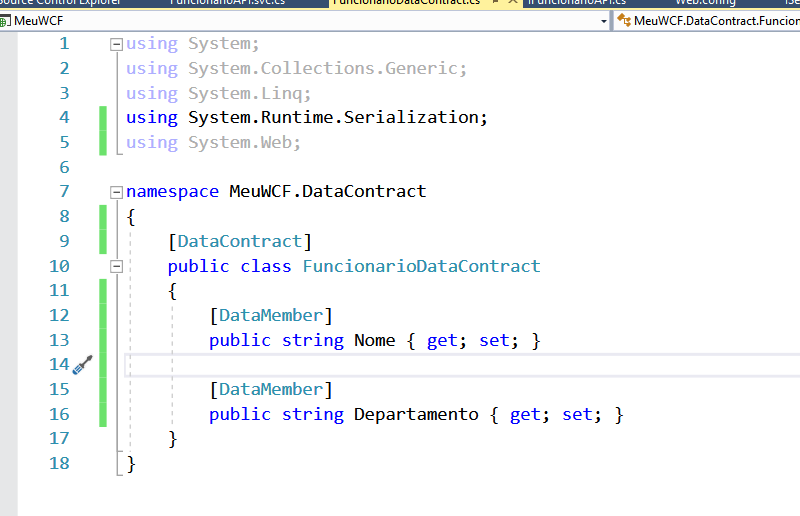


Agora crie a classe FuncionarioDataContract dentro da pasta DataContract

Esta pasta tem as classes que serão serializadas e devolvidas pela rede como o WCF

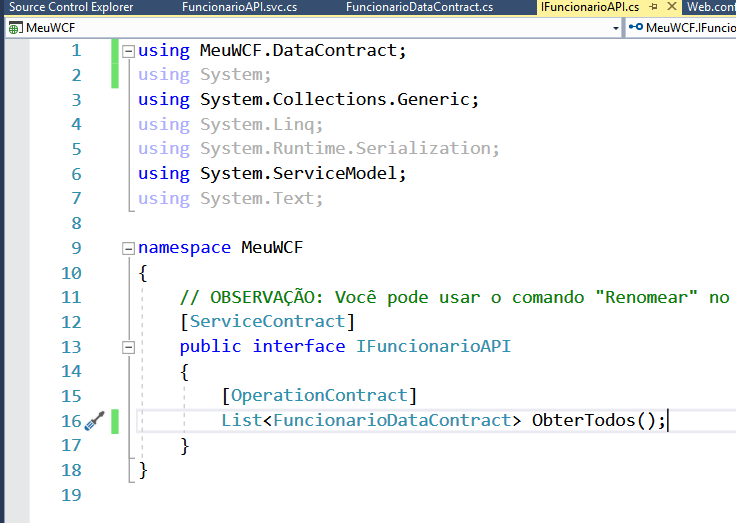


E deixe a classe especificada dessa forma:

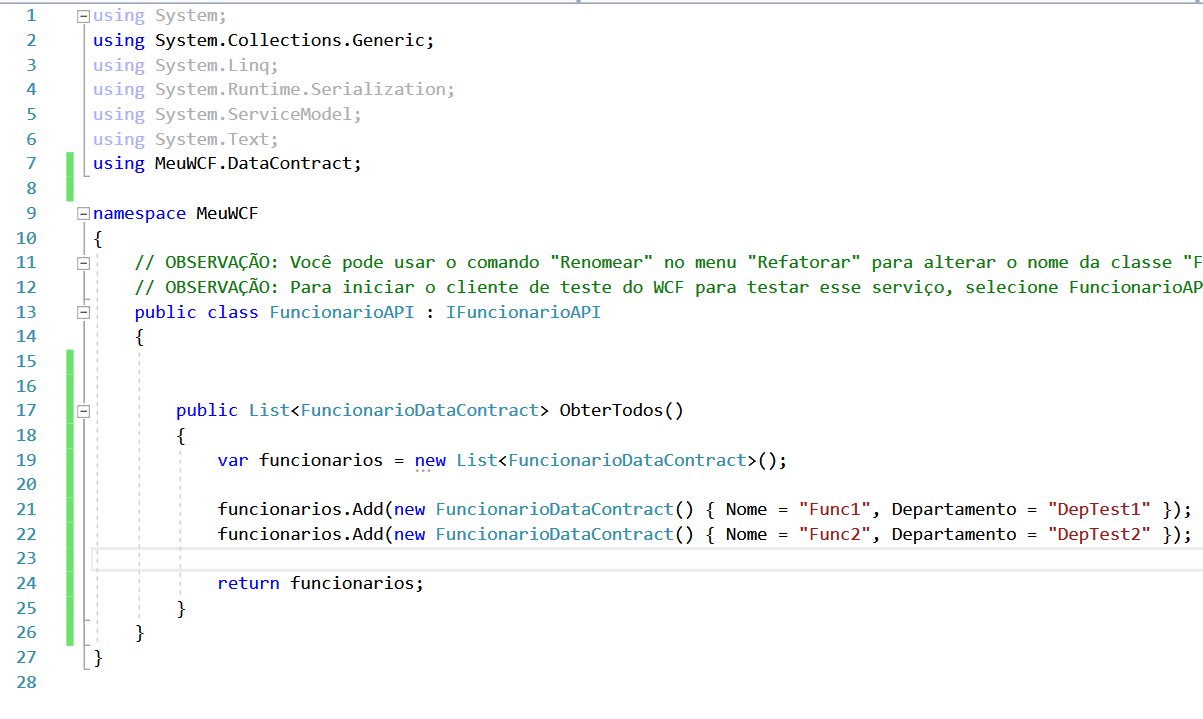


Observe o uso dos atributos **DataContract** e **DataMember**

Voltando a interface IFuncionarioAPI... deixe especificada desta forma

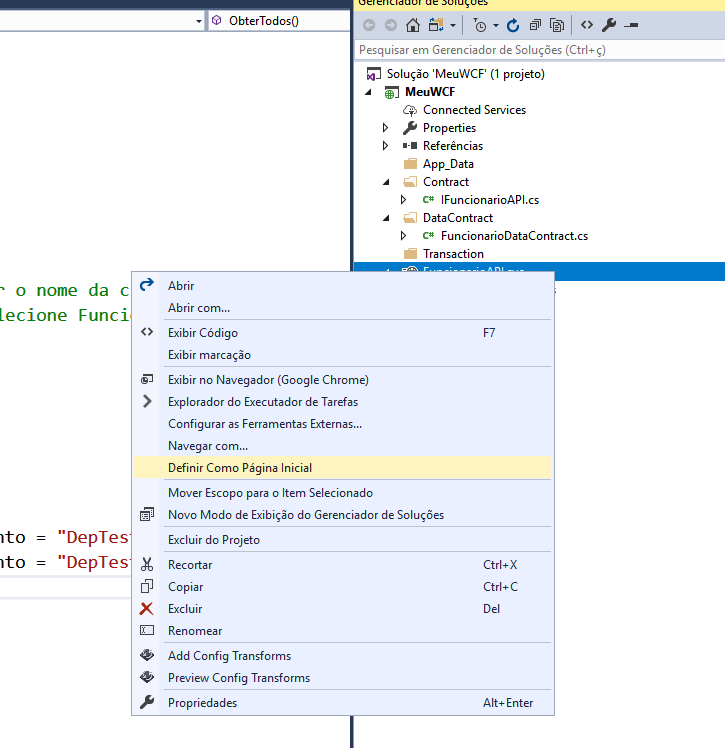


E a classe concreta FuncionarioAPI implementando a interface da seguinte maneira:

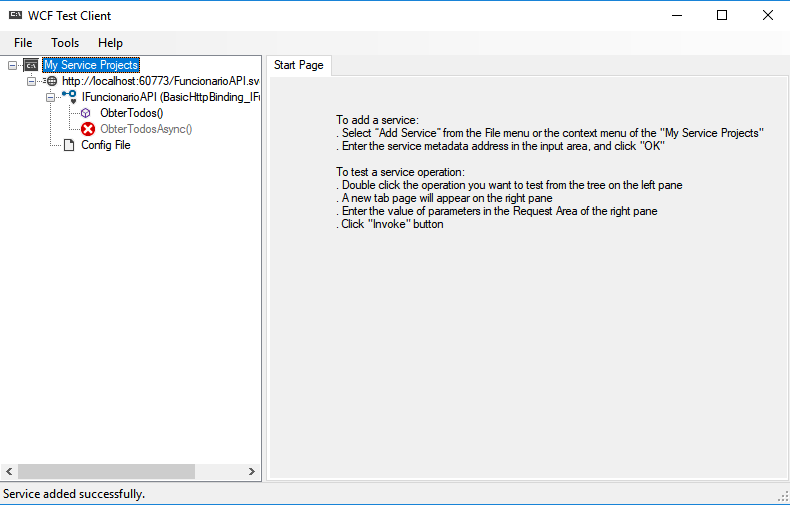


Seguindo esses últimos passos temos um novo serviço WCF e para testar...

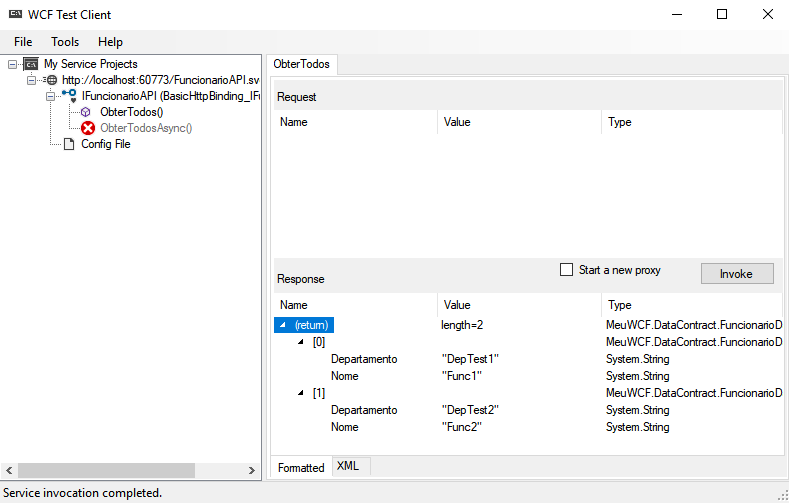
Clique com o botão direito em cima do arquivo **FuncionarioAPI.svc** , marque **Definir como Página Inicial**  e execute o projeto.



WCF Test Client na tela…



E realize alguns testes com o serviço clicando no método ObterTodos() e no botão Invoke



Finalizando...

Se você chegou até o final deste documento( é claro.. que lendo todo o conteúdo), você aprendeu de forma básica que..

* Todo sistema é contituído de partes e que juntos resolvem um problema em comum
* E que você pode distribuir somente parte do seu sistema
* WCF é uma forma de criar um sistema com arquitetura distribuída
* A parte fundamenal do WCF é composto por Mensagem, EndPoint e SOAP
* End Point é composto por.. **Address**, **Binding** e **Contract**
* **Address** informa o endereço do seu serviço
* **Binding** é para informar como seu serviço será acessado( procotolo no WCF)
* **Contract** é para informar qual contrato( interface) que será exposto no serviço
* E que também nem sempre é necessário informar toda a configuração de acesso ao seu serviço e que pode variar de acordo com a versão do .net framework e do visual studio
* Aprendeu também como criar seu próprio serviço
* E aprendeu também como realizar testes básicos utilizando o WCF Test Client

No próximo artigo, você vai aprender como consumir o seu serviço a partir de outro projeto em .net

Referências:

Sobre o WCF Oficial: <https://docs.microsoft.com/pt-br/dotnet/framework/wcf/whats-wcf>