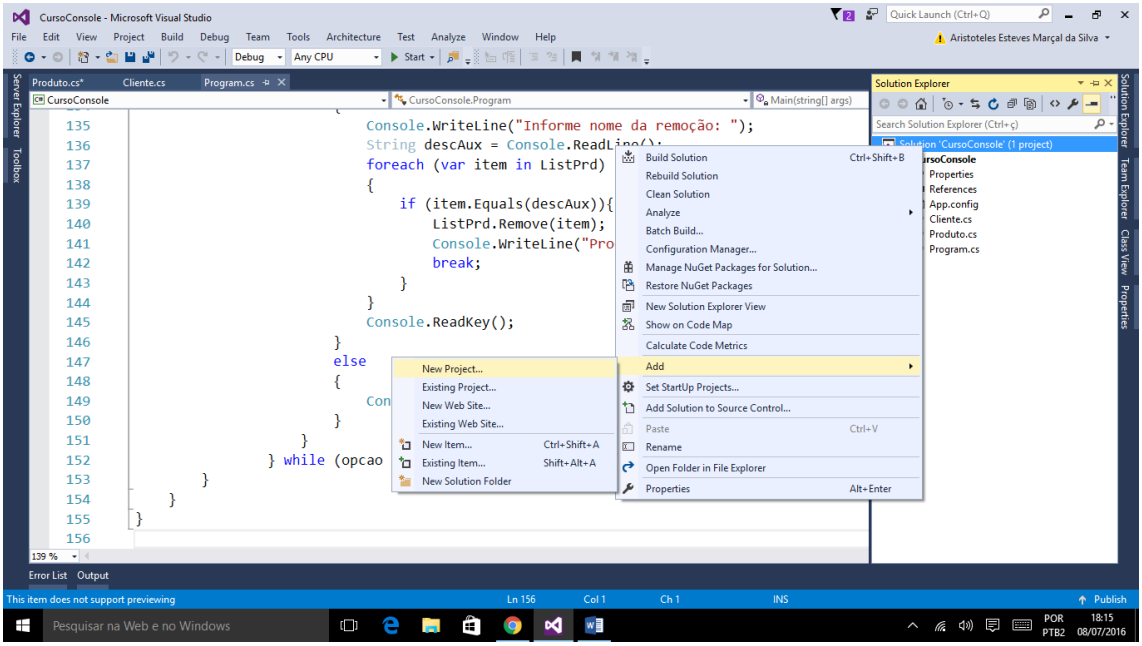
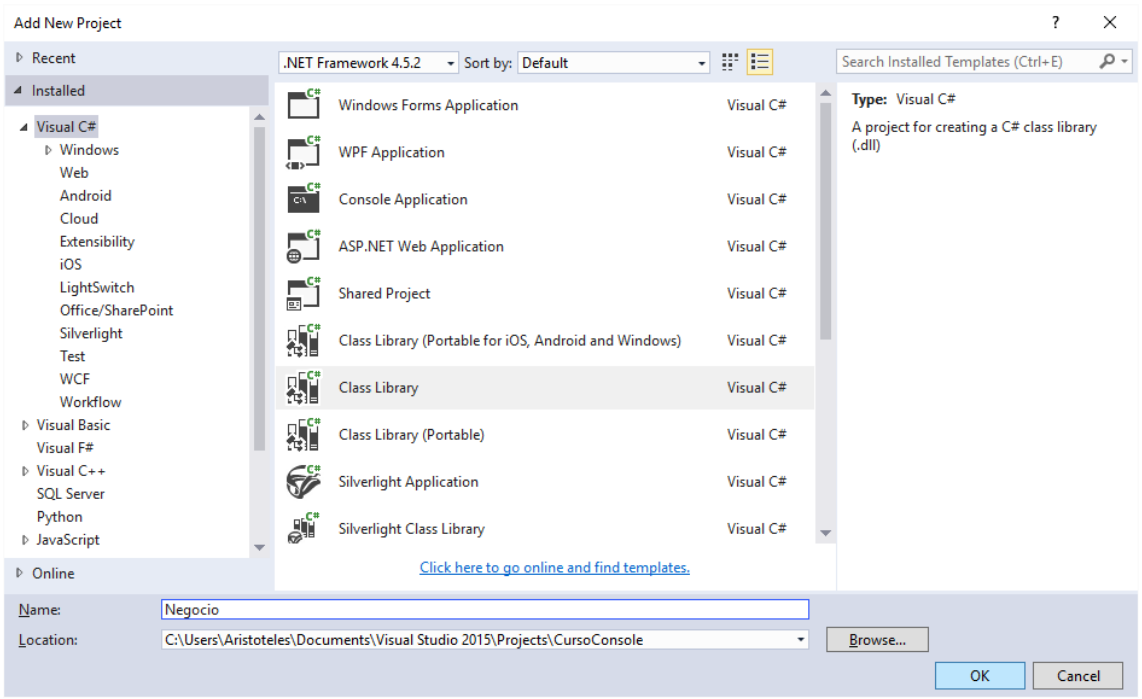
Prática 5

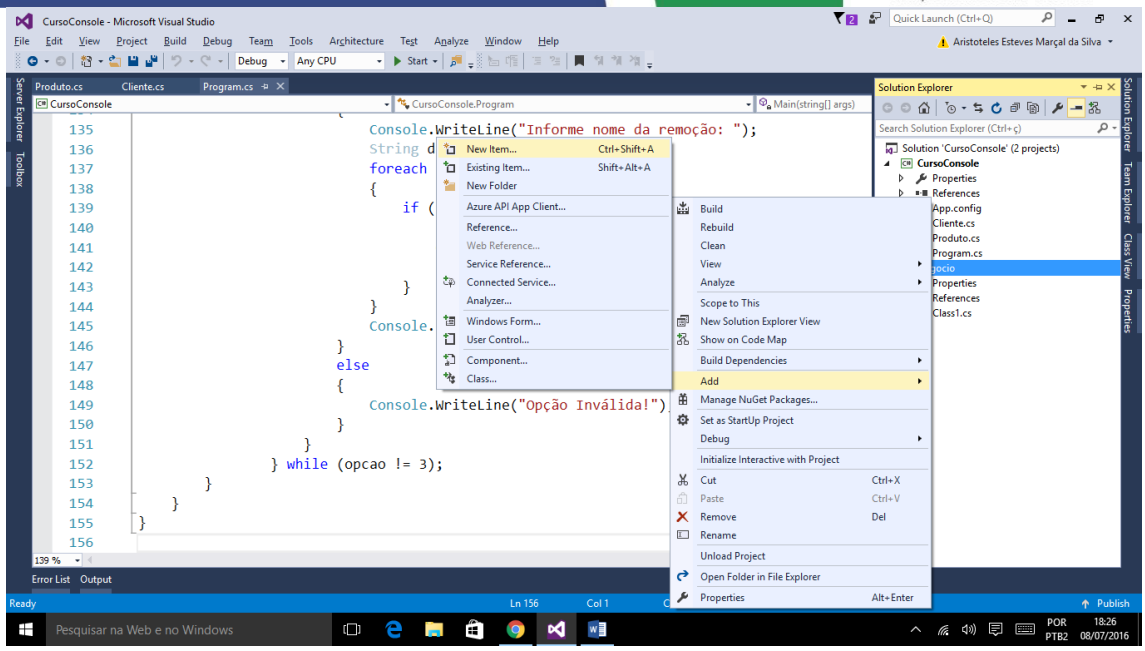
- Criar um projeto novo de biblioteca de classes chamado Negocio



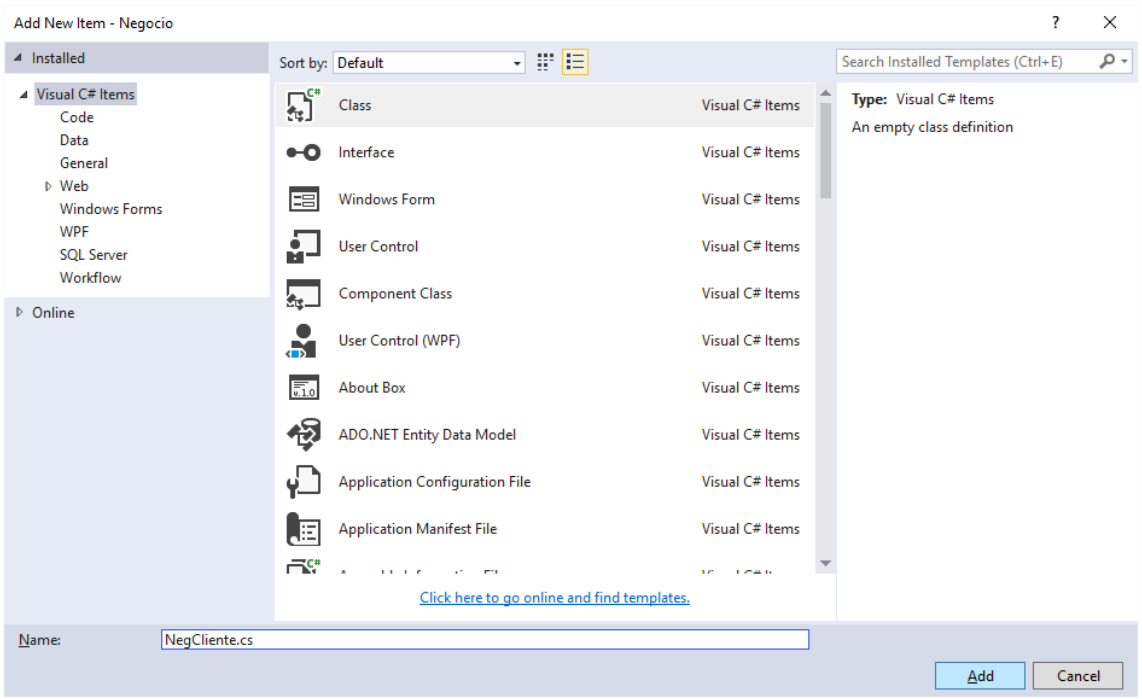
- Informe o nome do projeto como “Negocio”



- Criar uma nova classe



❖ Informe o nome da classe como “NegCliente”



❖ Informe que a classe é pública para que possa ser usada em outro namespace

```
namespace Negocio
{
    public class NegCliente
    {
    }
}
```

❖ Importe o projeto Negocio na classe `Program`

```
using Negocio;
```

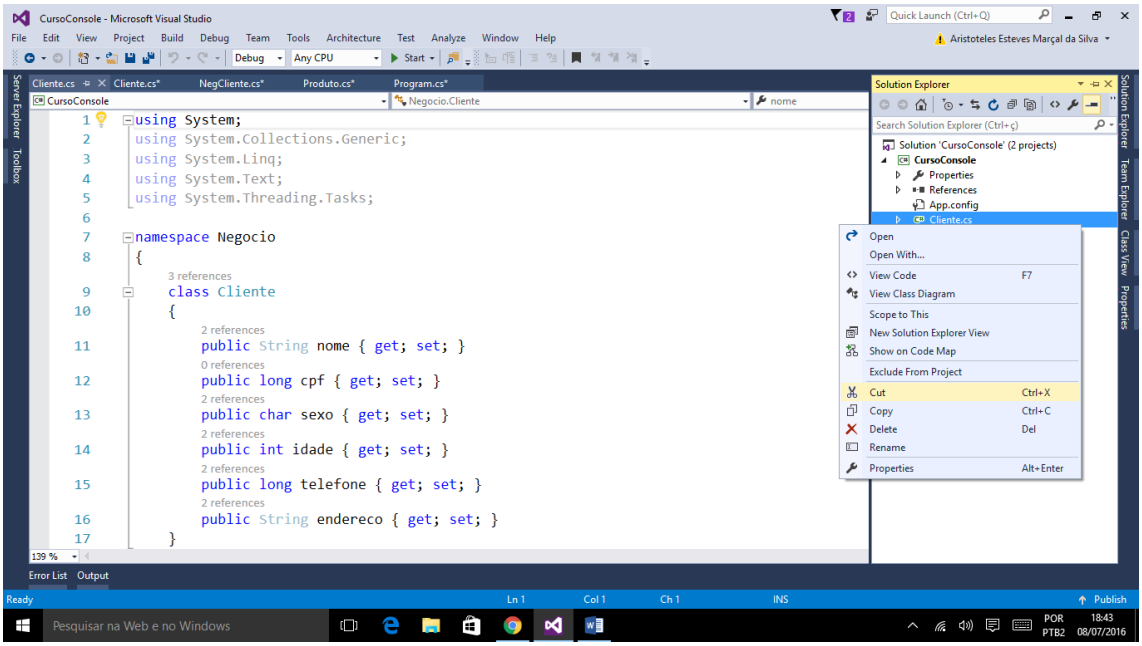
❖ Instancie um objeto da classe `NegCliente` no método main

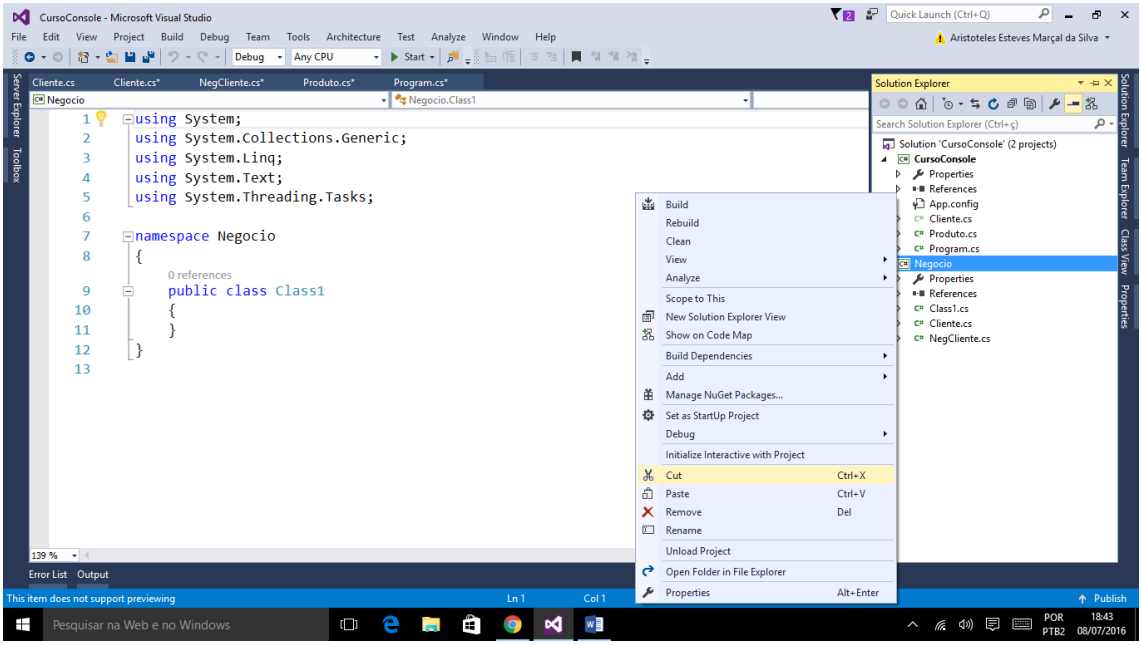
```
NegCliente negCliente = new NegCliente();
```

❖ Retira a lista de cliente da classe `Program` e coloque na classe `NegCliente`

```
namespace Negocio
{
     public class NegCliente
    {
            List<Cliente> ListCli = new List<Cliente>();
    }
}
```

❖ Retire a classe `Cliente` do projeto CursoConsole e passe para o projeto `Negocio`





❖ Altere o namespace da classe Cliente de `CursoConsole` para Negocio

```
namespace Negocio
{
```

```
public class Cliente
{
```

- Crie o método `CadastrarCliente` que receba como parâmetro uma variável `Cliente` e adicione na lista de clientes

```
public void CadastrarCliente(Cliente cli)
{
    ListCli.Add(cliente);
}
```

- Altere o cadastro de cliente na classe `Program` para que invoque o método `CadastrarCliente` da classe `NegCliente`

~~`ListCli.Add(cliente);`~~

```
negCliente.CadastrarCliente(cliente);
```

- Crie o método `ConsultarCliente` que retorne a lista de cliente

```
public List<Cliente> ConsultarCliente()
{
    return ListCli;
}
```

- Altere a consulta de cliente na classe `Program` para que utilize o método `CosultarCliente` da classe `NegCliente`

~~`foreach (Cliente cli in ListCli)`~~

```
foreach (Cliente cli in negCliente.ConsultarCliente())
```

- Crie o método `ConsultarClientePorNome` que retorne um cliente e receba como parâmetro o nome do cliente

```
public Cliente ConsultarClientePorNome(String nome)
{
    foreach (var item in ListCli)
    {
            if (item.nome.Equals(nome))
            return item;
    }
    return null;
}
```

- Altere a consulta de cliente na classe `Program` para que utilize o método `CosultarClientePorNome` da classe `NegCliente`

~~`foreach (var item in ListCli)`~~
~~`{`~~
~~`    if (item.nome.Equals(nomeAux))`~~
~~`        Console.WriteLine("Achei {0}!", item.nome);`~~
~~`}`~~

```
Cliente cli = negCliente.ConsultarClientePorNome(nomeAux);
Console.WriteLine("Achei {0}!", cli.nome);
```

- Crie o método `RemoverCliente` que retorne um cliente e receba como parâmetro o nome do cliente

```
public void RemoverCliente(String nome)
{
```

```
    foreach (var item in ListCli)
    {
            if (item.nome.Equals(nome))
            {
                    ListCli.Remove(item);
            }
    }
}
```

❖ Altere a consulta de cliente na classe `Program` para que utilize o método `CosultarClientePorNome` da classe `NegCliente`

```
foreach (var item in ListCli)
{
    if(item.nome.Equals(nomeAux)){
            ListCli.Remove(item);
            Console.WriteLine("Cliente removido com sucesso!");
    }
}
```

```
 negCliente.RemoverCliente(nomeAux);
Console.WriteLine("Cliente removido com sucesso!");
```

❖ Compile e teste

❖ Crie a classe `Pessoa` com os atributos nome e idade

```
public class Pessoa
{
    public String nome { get; set; }
    public int idade { get; set; }
}
```

❖ Retire os atributos nome e idade da classe `Cliente` e compile

```
public String nome { get; set; }
public int idade { get; set; }
```

❖ Altere Cliente para herdar de Pessoa

```
public class Cliente : Pessoa
```

❖ Compile e teste a aplicação

❖ Adicione um `try`/`catch` no método pesquisar por Nome da classe `NegCliente`

```
public Cliente ConsultarClientePorNome(String nome)
{
    try
    {
            foreach (var item in ListCli)
            {
                    if (item.nome.Equals(nome))
                    return item;
            }
    }
    catch (Exception)
    {
            throw;
    }
    return null;
}
```

- Compile e teste a aplicação
- Criar uma nova classe com o nome de `NegProduto`
- Crie os métodos `CadastrarProduto, ConsultarProduto, ConsultarProdutoPorDescrição e RemoverProduto` para a classe `NegProduto`
- Faças as alterações na classe `Program` para utilizar a classe `NegProduto`