

Instituto Nacional de Telecomunicações - INATEL

**Pós-graduação em Desenvolvimento de Aplicações
para Dispositivos Móveis e Cloud Computing**

DM113

Desenvolvimento de serviço SOAP com WCF em C#

Projeto Avaliativo

Autor: Prof. M.Sc. Felipe Andery Reis

Sumário

1) Introdução.....	3
2) Projeto Avaliativo	3
2.1) Banco de dados (10 pontos)	4
2.2) Serviço (30 pontos).....	5
2.3) Aplicação hospedeira (<i>self-hosting</i>) (30 pontos)	7
2.4) Cliente (30 pontos).....	8
3) Entrega	10

1) Introdução

O projeto de um sistema completo orientado a serviços está sendo desenvolvido durante o curso de pós-graduação. O objetivo da implementação do projeto da disciplina DM113 é avaliar a aprendizagem dos conteúdos apresentados nas aulas na criação do provedor de serviços de estoque, conforme mostrado a seguir.

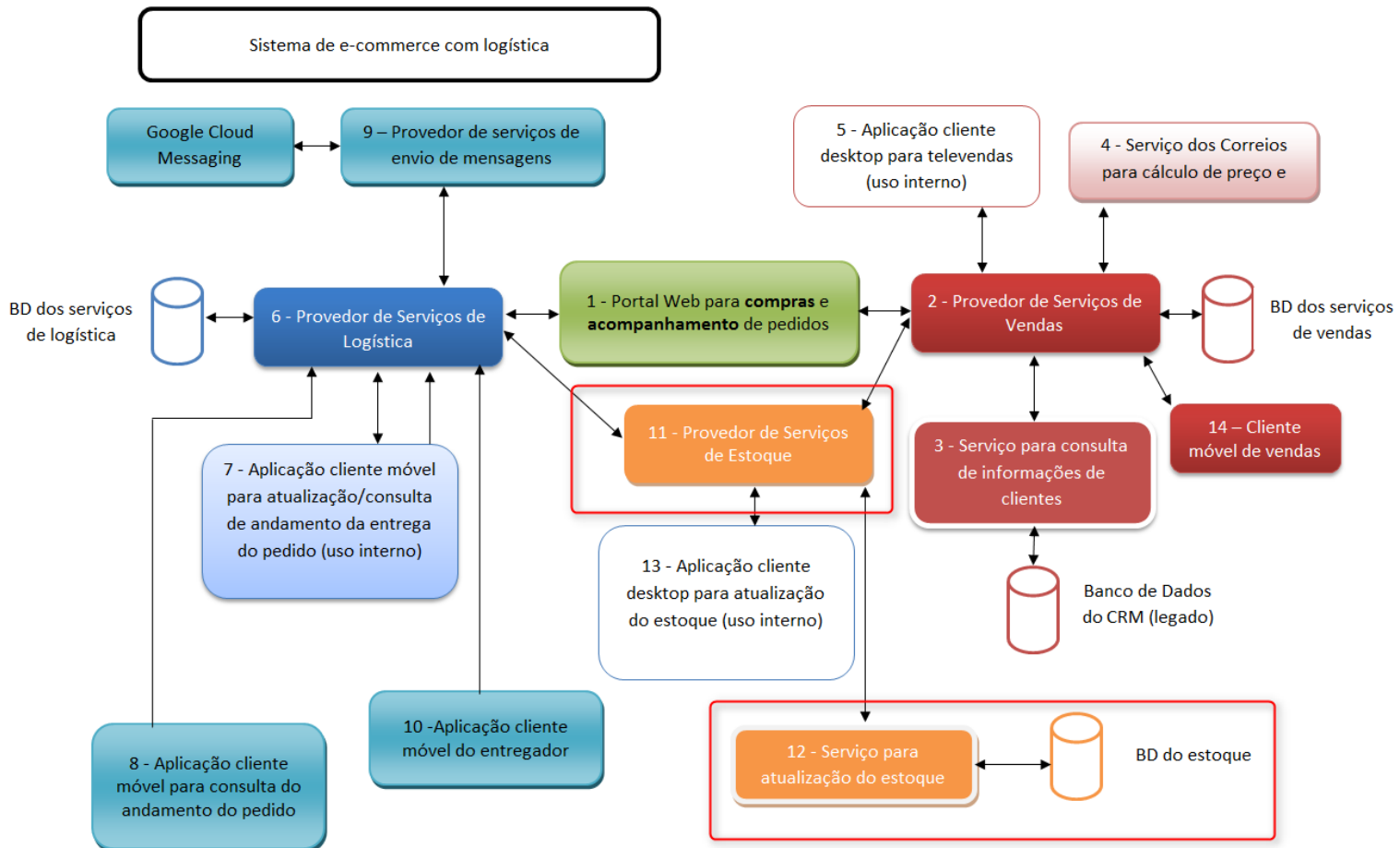


Figura 1 – Diagrama do sistema de e-commerce com logística

2) Projeto Avaliativo

Um serviço deverá ser desenvolvido e este será responsável por atualizar o estoque de determinado produto, cujas interfaces serão mostradas em uma seção posterior. Ele deverá acessar um banco de dados que será constituído por uma única tabela. E, por fim, o funcionamento do sistema deverá ser comprovado através de clientes de testes. Dessa forma, deve-se utilizar a seguinte abordagem:

Banco de dados, serviço e cliente locais

- O acesso ao banco de dados poderá ser feito localmente através da instância (localdb) do Visual Studio.
- O serviço deverá ser criado como uma biblioteca para ser hospedado em um projeto *host*
- Os clientes deverão ser criados utilizando uma aplicação do tipo *Console*.

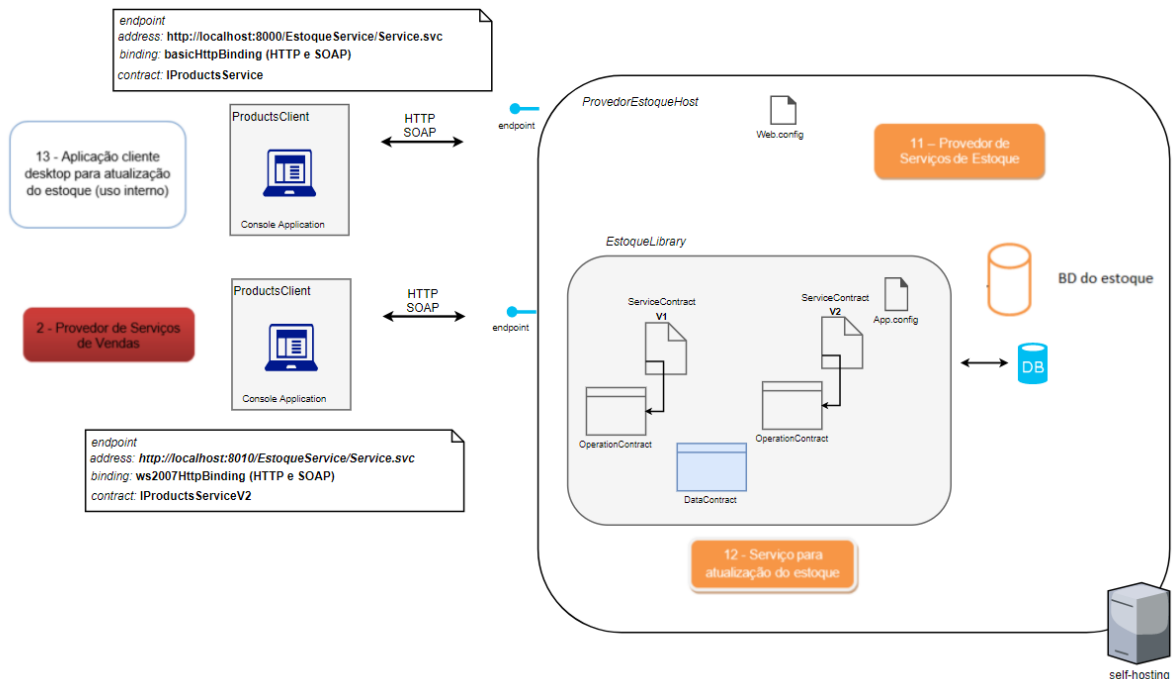
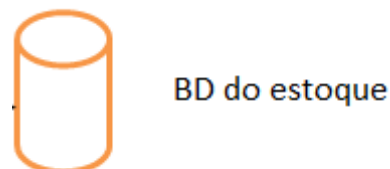


Figura 2 – Diagrama do projeto avaliativo

2.1) Banco de dados (10 pontos)

Neste item deverá ser criada uma base de dados através da utilização do Entity Framework Code First e do Migrations. Esta base de dados será responsável por armazenar as informações de estoque dos produtos do sistema de e-commerce, conforme o módulo BD do estoque mostrado na Figura 1.



- Crie um projeto do tipo *Class Library* com o nome “*EstoqueEntityModel*” que representará uma biblioteca com o modelo de *Entity* que será utilizado para a comunicação com o banco de dados. Este modelo deverá utilizar o ADO.NET Entity Data Model com o nome “*ProvedorEstoque*” e deverá ser criado a partir do Empty Code First Model.

- A partir deste modelo, crie a entidade *ProdutoEstoque* que representará a tabela do banco de dados. A tabela *ProdutoEstoque* deverá conter os campos mostrados na Tabela I.

Tabela I - Estrutura da tabela *ProdutoEstoque*

Nome do Campo	Tipo	Permite Nulos?
NumeroProduto (chave primária)	nchar(10)	Não
NomeProduto	nchar(20)	Sim
DescricaoProduto	nchar(50)	Sim
EstoqueProduto	Int	Sim

- Adicione o Migrations ao projeto
- Utilizando o *seed* do Migrations, inicialize a tabela *ProdutoEstoque* com os seguintes dados.

(NumeroProduto, NomeProduto, DescricaoProduto, EstoqueProduto)

```
( '1000', 'Produto 1', 'Este é o produto 1', 100 );
( '2000', 'Produto 2', 'Este é o produto 2', 10 );
( '3000', 'Produto 3', 'Este é o produto 3', 200 );
( '4000', 'Produto 4', 'Este é o produto 4', 300 );
( '5000', 'Produto 5', 'Este é o produto 5', 400 );
( '6000', 'Produto 6', 'Este é o produto 6', 500 );
( '7000', 'Produto 7', 'Este é o produto 7', 30 );
( '8000', 'Produto 8', 'Este é o produto 8', 30 );
( '9000', 'Produto 9', 'Este é o produto 9', 400 );
( '10000', 'Produto 10', 'Este é o produto 10', 2 );
```

- Utilize o Migrations para gerar a base de dados
- Utilizando a instância do (*localdb*) do *Visual Studio*, verifique a criação da base de dados chamada de *ProvedorEstoque*.

2.2) Serviço (30 pontos)

Este item representará o módulo *Serviço para atualização do estoque* do diagrama da Figura 1. Nele deverá ser criado o serviço “*ServicoEstoque*” como parte de uma biblioteca, a *EstoqueLibrary*, para ser hospedado localmente (*host*) na máquina.

12 - Serviço para
atualização do estoque

Os contratos de dados e serviço deverão ser definidos conforme a seguir.

- Contrato de Dados

O contrato de dados do serviço deverá ser criado conforme o diagrama da Figura 3. Lembre-se de utilizar as *tags* `[DataContract]` e `[DataMember]`.

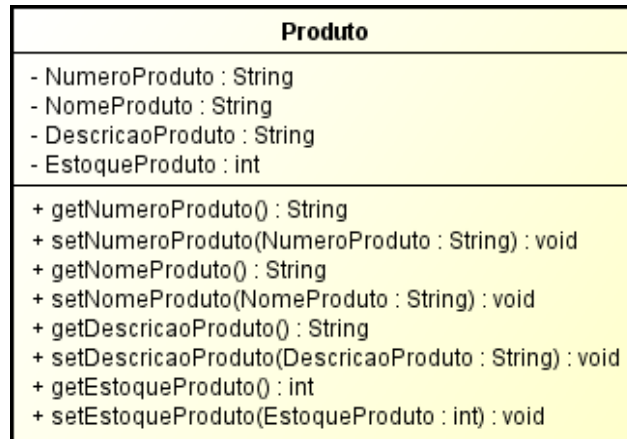


Figura 3 - Diagrama do Contrato de Dados

- Contrato de Serviço

O serviço deverá possuir dois contratos. A primeira versão do contrato deverá ser criada conforme o diagrama da Figura 4. Utilize o seguinte *namespace* para esta versão:

Namespace = "<http://projetoavaliativo.dm113/01>"

A segunda versão deverá ser criada conforme o diagrama da Figura 5. Utilize o seguinte *namespace* para esta versão:

Namespace = "<http://projetoavaliativo.dm113/02>"

Lembre-se de utilizar as *tags* `[ServiceContract]` e `[OperationContract]`.

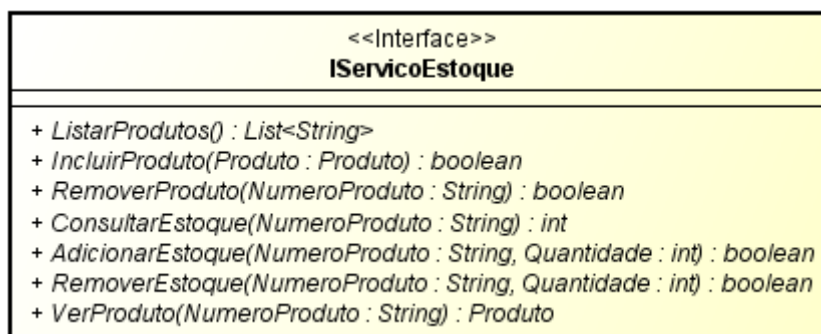


Figura 4 - Diagrama da primeira versão do Contrato de Serviço

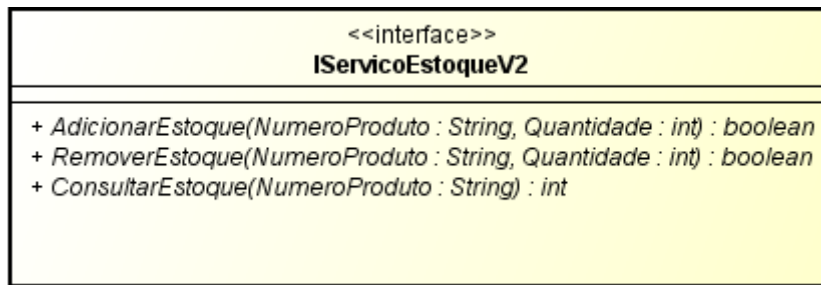


Figura 5 - Diagrama da segunda versão do Contrato de Serviço

As operações deverão realizar as funções descritas na Tabela V.

Tabela V - Funções das Operações do Contrato de Serviço

Operação	Parâmetros	Retorno
List<string> ListarProdutos()	-	Lista de <i>strings</i> contendo os nomes dos produtos no banco.
bool IncluirProduto(Produto)	Produto a ser incluído com seus parâmetros preenchidos.	Bool indicando o sucesso ou falha na operação.
bool RemoverProduto(string)	String contendo o número do produto a ser removido.	Bool indicando o sucesso ou falha na operação.
int ConsultarEstoque(string)	<i>String</i> contendo o número do produto do qual se requer o estoque.	<i>int</i> com o número de itens em estoque.
bool AdicionarEstoque(string, int)	<i>String</i> contendo o número do produto a ter o estoque aumentado. <i>int</i> contendo o valor a ser acrescentado ao estoque.	<i>Bool</i> indicando o sucesso ou falha na operação.
bool RemoverEstoque(string, int)	<i>String</i> contendo o número do produto a ter o estoque reduzido. <i>int</i> contendo o valor a ser deduzido do estoque.	<i>Bool</i> indicando o sucesso ou falha na operação.
Produto VerProduto(string)	<i>String</i> contendo o número do produto a ser consultado.	<i>Produto</i> contendo os parâmetros recuperados do banco.

2.3) Aplicação hospedeira (*self-hosting*) (30 pontos)

Este item representará o módulo *Provedor de Serviços de Estoque* do diagrama da Figura 1. Nele deverá ser criada uma aplicação para hospedar o serviço *ServicoEstoque*.

11 – Provedor de Serviços de Estoque

- Para hospedar o serviço, crie um aplicativo utilizando o modelo *Console Application*, com o nome *ProvedorEstoqueHost*. O aplicativo deve ser similar ao criado em sala de aula.
- As operações que serão expostas pelo serviço deverão ser disponibilizadas para as aplicações clientes através de dois *endpoints*. Cada aplicação *Cliente*

deverá acessar o serviço através de um *endpoint* específico. As seguintes configurações devem ser feitas na aplicação de hospedagem:

- *Endpoint 1*

Contrato: *EstoqueLibrary.IServicoEstoque*

Modo de comunicação: HTTP

Endereço: *http://localhost:8000/EstoqueService/Service.svc*

Binding: *basicHttpBinding*

Nome: *BasicHttpBinding_IServicoEstoque*

- *Endpoint 2*

Contrato: *EstoqueLibrary.IServicoEstoqueV2*

Modo de comunicação: HTTP

Endereço: *http://localhost:8010/EstoqueService/Service.svc*

Binding: *ws2007HttpBinding*

Nome: *WS2007HttpBinding_IServicoEstoque*

Lembre-se de reservar as portas utilizando o comando *netsh*.

2.4) Cliente (30 pontos)

- O cliente deverá ser desenvolvido como se o código do serviço não estivesse disponível. Dessa forma, a ferramenta *svcutil* deverá ser utilizada para criar a classe *proxy* que proverá acesso ao serviço. Como o serviço é local, a ferramenta *svcutil* deverá ser utilizada na biblioteca (*dll*) do serviço. Lembre-se que cada cliente deverá utilizar um contrato de serviço específico, ou seja, deve ser gerada uma nova classe *proxy* para cada versão do contrato.
- Para realizar a comunicação com o serviço, deverão ser criadas duas aplicações clientes, do tipo *Console Application*, onde cada cliente terá a seguinte função:

Cliente 1

- Representará um aplicativo desktop (para uso interno) que será responsável pela inclusão de itens no estoque e terá informações dos produtos cadastrados.

13 - Aplicação cliente
desktop para atualização
do estoque (uso interno)

- O *Endpoint* para este cliente deve ser configurado com as informações do *Endpoint 1 da seção Aplicação Hospedeira*. Lembrando que este cliente terá acesso às operações fornecidas pela primeira versão do contrato de serviços.
- Deverá requisitar ao serviço *ServicoEstoque* todas as operações. Os seguintes testes poderão ser realizados:
 - 1) Adicionar um produto (por exemplo, Produto 11)
 - 2) Remover o produto 10
 - 3) Listar todos os produtos
 - 4) Verificar todas as informações do Produto 2
 - 5) *Adicionar 10 unidades para este produto*
 - 6) *Verificar o estoque do Produto 2*
 - 7) *Verificar o estoque atual do Produto 1*
 - 8) *Remover 20 unidades para este produto*
 - 9) *Verificar o estoque do Produto 1 novamente*
 - 10) Verificar todas as informações do Produto 1

Cliente 2

- Representará o módulo *Provedor de Serviços de Vendas*, que irá informar a quantidade vendida de cada item para que o banco de dados de estoque seja atualizado.

2 - Provedor de Serviços de Vendas

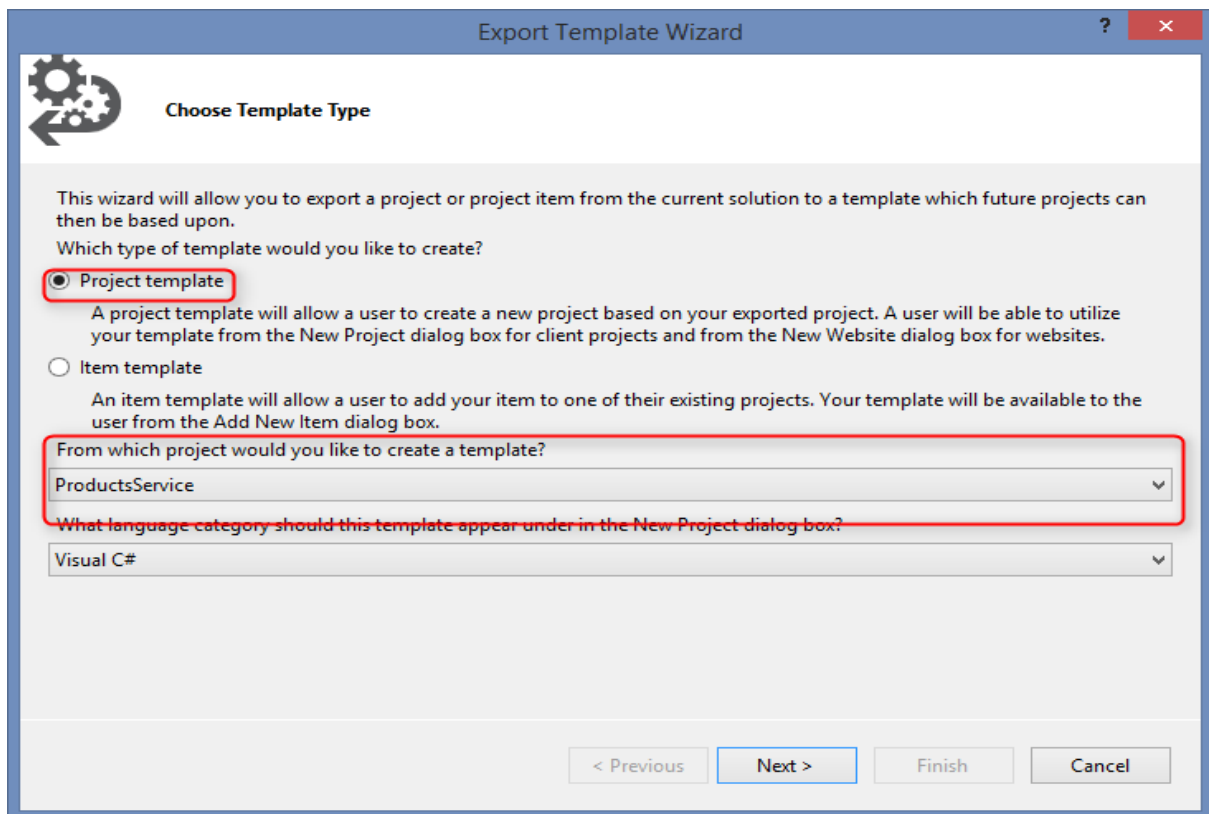
- O *Endpoint* para este cliente deve ser configurado com as informações do *Endpoint 2 da seção Aplicação Hospedeira*. Lembrando que este cliente terá acesso às operações fornecidas pela segunda versão do contrato de serviços.
- Deverá requisitar ao serviço *ServicoEstoque* as operações de *AdicionarEstoque*, *RemoverEstoque* e *ConsultarEstoque*. Os seguintes testes poderão ser realizados:
 - 1) *Verificar o estoque atual do Produto 1*
 - 2) *Adicionar 20 unidades para este produto*
 - 3) *Verificar o estoque do Produto 1 novamente*
 - 4) *Verificar o estoque atual do Produto 5*
 - 5) *Remover 10 unidades para este produto*
 - 6) *Verificar o estoque do Produto 5 novamente*

3) Entrega

- Projeto Individual
- Data de entrega: **29/10/2017**
- Todos os projetos realizados deverão ser enviados para o email fandery@inatel.br
- Por favor, identificar o assunto do email com a seguinte nomenclatura:
DM113_Turma_Nome_aluno
- Qualquer dúvida ou dificuldade no entendimento do trabalho proposto entrar em contato por email.
- Para cada projeto criado no *Solution Explorer* do Visual Studio, deverá ser exportado um template. Este template irá gerar um arquivo compactado no formato .zip.
- Todos os templates exportados deverão ser colocados em uma única pasta, e esta deverá ser compactada para o envio ao professor.

Para exportar cada projeto como um template, siga os seguintes passos:

- 1) No Visual Studio, clicar em *Project* e depois em *Export Template*.
- 2) A tela de *Export Template Wizard* será aberta, como mostrado na Figura 6. Marque a opção *Project Template* e escolha um projeto a ser exportado na opção *From which project would you like to create a template?*
- 3) Clique em *Next*.



The image shows the 'Export Template Wizard' dialog box. It has a title bar with a question mark and a close button. The main area is titled 'Choose Template Type' and contains a gear icon. The text explains the wizard's purpose and asks for the template type. Two options are shown: 'Project template' (selected) and 'Item template'. Below these, there are two dropdown menus: 'From which project would you like to create a template?' (set to 'ProductService') and 'What language category should this template appear under in the New Project dialog box?' (set to 'Visual C#'). At the bottom are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

Export Template Wizard

Choose Template Type

This wizard will allow you to export a project or project item from the current solution to a template which future projects can then be based upon.

Which type of template would you like to create?

☒ Project template

A project template will allow a user to create a new project based on your exported project. A user will be able to utilize your template from the New Project dialog box for client projects and from the New Website dialog box for websites.

☐ Item template

An item template will allow a user to add your item to one of their existing projects. Your template will be available to the user from the Add New Item dialog box.

From which project would you like to create a template?

ProductService

What language category should this template appear under in the New Project dialog box?

Visual C#

< Previous Next > Finish Cancel

Figura 6 - Export Template Wizard

- 4) Na próxima tela clique em *Finish* e verifique se o arquivo foi gerado no diretório especificado em *Output Location*.
- 5) Repita os passos anteriores para todos os projetos do *Solution Explorer*.