

Relatório do 2º Projeto

Tecnologias de Distribuição e Integração

Ordens de compra/venda de ações numa Instituição Financeira

Windows Communication Foundation / Microsoft Silverlight 5



Universidade do Porto

Faculdade de Engenharia

FEUP

30 de Maio de 2012

Ordens de compra/venda de ações numa Instituição Financeira

Autores

José Pedro Silva – jose.pedro.silva@fe.up.pt

Pedro Simeão Carvalho – pedro.simeao@fe.up.pt

Trabalho realizado no âmbito da unidade curricular optativa de Tecnologias de Distribuição e Integração, do 2º semestre do 4º ano, do Mestrado Integrado em Engenharia Informática e Computação, da Faculdade de Engenharia da Universidade do Porto



Universidade do Porto
Faculdade de Engenharia
FEUP

30 de Maio de 2012

Conteúdo

Lista de Figuras	4
Introdução	5
Compilação e Execução	5
Modelação.....	7
Diagramas UML	7
Aplicações.....	7
Servidor	7
Cliente Web / Balcão	8
Serviço Bolsista.....	8
Aplicações.....	9
Servidor	9
Aplicação Web.....	10
Aplicação Balcão.....	12
Aplicação Serviço Bolsista	15
Conclusão	16
Referências.....	16
Anexo I.....	17
Anexo II.....	18

Lista de Figuras

Figura 1 – Consola do servidor	9
Figura 2 – Aplicação Web – Página inicial, com Login.....	10
Figura 3 – Aplicação Web – Página inicial após login como administrador	10
Figura 4 – Aplicação Web – Adição de nova transação como administrador	10
Figura 5 – Aplicação Web – Aplicação de filtro do nome de cliente na listagem de transações.....	11
Figura 6 – Aplicação Balcão – Opção de instalação da aplicação.....	12
Figura 7 – Aplicação Balcão – Registo de um utilizador.....	12
Figura 8 – Aplicação Balcão – Página inicial com inicio de sessão com utilizador normal.....	12
Figura 9 – Aplicação Balcão – Painel inicial com inicio de sessão com utilizador normal, após validação de uma transação.....	13
Figura 10 – Aplicação Balcão – Página de Login, com autenticação sem sucesso	13
Figura 11 – Aplicação Balcão – Página inicial com inicio de sessão com administrador	13
Figura 12 – Aplicação Balcão – Página e funcionalidade de atualização da aplicação	14
Figura 13 – Aplicação Serviço Bolsista – Página inicial com lista de transações pendentes.....	15
Figura 14 – Aplicação Serviço Bolsista – Seleção de uma transação e preenchimento do valor da mesma	15

Introdução

O presente relatório visa explicar a abordagem por nós utilizada ao trabalho proposto na unidade curricular de Tecnologias de Distribuição e Integração.

O objetivo do projeto é a implementação de um serviço bolsista, que possibilita aos utilizadores inserirem as suas transações, quer via *web browser*, quer via balcão do serviço, com aprovação por parte de um outro serviço, o serviço bolsista, dessas mesmas transações.

Pretende-se com este projeto ganhar sensibilidade para a elaboração de sistemas com diversas aplicações e subsistemas, incluindo serviços web, serviço de fila de mensagens do *Windows* e *Windows Communication Foundation*. Esta oportunidade foi também aproveitada por nós para conhecermos a tecnologia *Silverlight* e consequente ligação aos serviços já citados, interagindo entre si, criando um sistema completo.

Compilação e Execução

O projeto pode ser compilado e executado através da ferramenta *Visual Studio 2010*, para o qual disponibilizamos a solução.

A solução existente contém quatro projetos: *ClientApplicaton*, *ClientApplication.Web*, *Server* e *StockService*. Cada um dos projetos representa uma aplicação diferente, nomeadamente, a aplicação *Silverlight*, o projeto *ASP.NET* que disponibiliza a aplicação *Silverlight*, o servidor *WCF* que implementa a camada de serviços do sistema, e a *Windows Forms Application* que implementa o serviço bolsista.

Os projetos *Server* e *StockService* têm dependências para bases de dados *SQL Server 2008*, que deverão ser criadas previamente, e colocadas a correr num servidor, apropriado para o efeito, que suporte transações.

É necessária a criação de uma *queue* privada, de nome *stockservice*, com suporte a transações, no servidor *MSMQ*. Para executar os projetos, basta correr os seguintes comandos:

```
Projeto2\Server\bin\Release> Server.exe
```

```
Projeto2\StockService\bin\Release> StockService.exe
```

A aplicação Web, embora seja possível instalar manualmente, é mais simples de o efetuar a partir do *Visual Studio*. A aplicação Web pode ser corrida dentro de um servidor *IIS*, contudo o *Visual Studio 2010* oferece o *ASP.NET Development Server* que o torna mais simples de executar, bastando lançar o projeto *ClientApplication.Web* em modo *Debug*.

Para configurar as aplicações, estão disponíveis os seguintes ficheiros para cada aplicação:

- **Server:** app.config
- **ClientApplication:** ServiceReferentes.ClientConfig

Relatório do 2º Projeto – Tecnologias de Distribuição e Integração

- **ClientApplication.Web:** Web.config
- **StockService:** app.config

Para cada um dos projetos, existem configurações diferentes que se adequam aos mesmos.

Para instalar a aplicação GUI do balcão é necessário aceder à página Web da aplicação *Silverlight*, onde devemos clicar com o botão direito do rato num qualquer ponto da página. Aí basta seleccionar a opção “Instalar TDIN - Financial Institute neste computador...” e seguir o *wizard* de instalação. De notar que para que a aplicação funcione em pleno devem estar a correr, em simultâneo, as aplicações *Server* e *ClientApplication.Web*.

Os dados de acesso ao sistema, em qualquer dos projetos, como administrador são: *admin* como nome de utilizador e *qaz!"#WSX* como palavra-chave.

Modelação

Para o correto desenvolvimento do conjunto de aplicações foi necessário efetuar um levantamento dos requisitos através de uma leitura atenta do enunciado. Após estes requisitos terem sido recolhidos e analisados realizamos um trabalho de conceção e estruturação do conjunto de aplicações.

Diagramas UML

No anexo I encontra-se o diagrama físico de implementação. No anexo II encontra-se o diagrama de classes que descreve o conjunto de aplicações desenvolvido.

Aplicações

Como especificado no enunciado, foram desenvolvidos três conjuntos aplicativos que modelam o problema. Estes três conjuntos são o Servidor/ClientApplication.Web, que contém os serviços indispensáveis ao funcionamento do sistema, a aplicação Web/GUI para o balcão e a aplicação GUI para o serviço bolsista.

Segue-se uma descrição detalhada dos conjuntos aplicativos.

Servidor

O servidor desenvolvido é um *self-hosted WCF Service*, com base numa aplicação de consola, que fornece os serviços básicos a todas as aplicações através da disponibilização de um serviço *WCF* sobre *NetTcpBinding*. A escolha deste *binding*¹ deveu-se, não só à sua compatibilidade com a tecnologia *Microsoft Silverlight 5*, mas sobretudo ao seu desempenho² quando comparado com outros *bindings* também suportados pelo *Microsoft Silverlight 5*, como o *basicHttpBinding*.

Detalhes de Implementação

Qualquer aplicação *Silverlight* que aceda a um *WCF Service* que não seja disponibilizado por ela própria considera que esse *WCF Service* está a correr num domínio diferente, até porque a porta onde o serviço é disponibilizado é diferente da sua. Assim sendo, e para que fosse possível aceder a um *self-hosted WCF Service*, foi necessário que o servidor disponibilizasse o ficheiro *clientaccesspolicy.xml* na porta 80 do seu endereço. Desta forma, para que tudo funcionasse, foi necessário adicionar um *webHttpBinding* que fornece, na porta 80, o ficheiro já mencionado à aplicação *Silverlight*.

Para que a autenticação se procedesse com a aplicação *Silverlight*, visto que este não está preparado para funcionar com *CustomAuthentication* através de serviços *WCF*, foi necessário adicionar um *OperationBehavior* a todas as funções que requerem autenticação. Este *OperationBehavior* encarrega-se de analisar o *header* do pacote SOAP recebido e verifica se os dados fornecidos pelo cliente estão corretos. Desta forma o processo de autenticação torna-se transparente para o programador.

¹ [Windows Communication Foundation Bindings](#)

² Afirmação feita com base num [estudo](#) comparativo publicado por Terrance A. Snyder

Cliente Web / Balcão

Uma das razões para a adoção da tecnologia *Silverlight* foi a possibilidade de utilizar a mesma base para desenvolver as aplicações Web e local. Assim, o cliente Web e de Balcão são, no fundo, a mesma aplicação, a correr ora num qualquer *browser* ora instalada no computador.

Destacam-se as seguintes funcionalidades implementadas que não estavam especificadas no enunciado do projeto:

- Registo e autenticação no sistema;
- Atualização automática da aplicação instalada localmente através da página “About”;
- Tratamento erros e sugestões visuais em formulários;

O projeto *ClientApplication.Web* é gerado automaticamente pelo Visual Studio e tem como único objetivo a disponibilização da aplicação *Silverlight* via web através de ASP.NET. Foi apenas ligeiramente modificado para que suportasse a autenticação implementada.

Detalhes de Implementação

Para a autenticação se proceder da forma que todos estamos habituados numa outra qualquer aplicação Web, recorreremos *WCF RIA Services*, que fazem a interligação entre ASP.Net e Silverlight, tornando simples processos como a validação de dados ou a autenticação.

Para procedermos à autenticação de todas as chamadas ao servidor foi necessário adicionar um *Endpoint Behavior* que se encarregou de inserir no cabeçalho das chamadas os campos necessários, para que tudo se desenrolasse de forma transparente.

Ainda dentro da aplicação *Silverlight* recorreremos ao controlo *DataForm*, uma inovação introduzida pela framework, que nos permitiu construir formulários muito completos e implementar a validação de campos mesmo antes de enviarmos os dados para o servidor, permitindo ao utilizador uma interação mais agradável com a interface.

Serviço Bolsista

O serviço bolsista é uma aplicação *Windows Forms Application* onde incorporamos os requisitos estipulados no enunciado. Para além disto, à imagem daquilo que foi nas restantes aplicações cliente, também introduzimos *custom headers* nos pedidos feitos ao servidor para validar as mensagens trocadas entre ambos.

Esta aplicação tem acesso às filas de mensagens do sistema (*Microsoft Message Queue*) para ler os pedidos que são enviados, e comunica-se ao servidor com um pedido normal de acesso a *WCF*.

Aplicações

Seguem-se algumas imagens de execução das aplicações desenvolvidas. Foram escolhidas as partes mais interessantes de cada aplicação, que visam demonstrar aspetos do funcionamento das mesmas.

Servidor

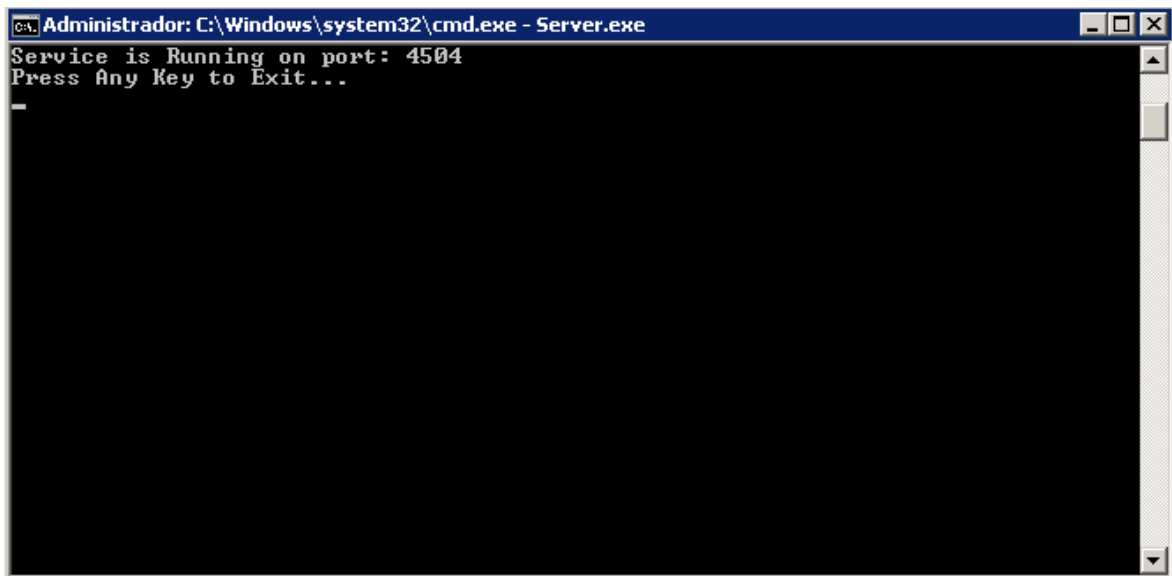


Figura 1 – Consola do servidor

Aplicação Web

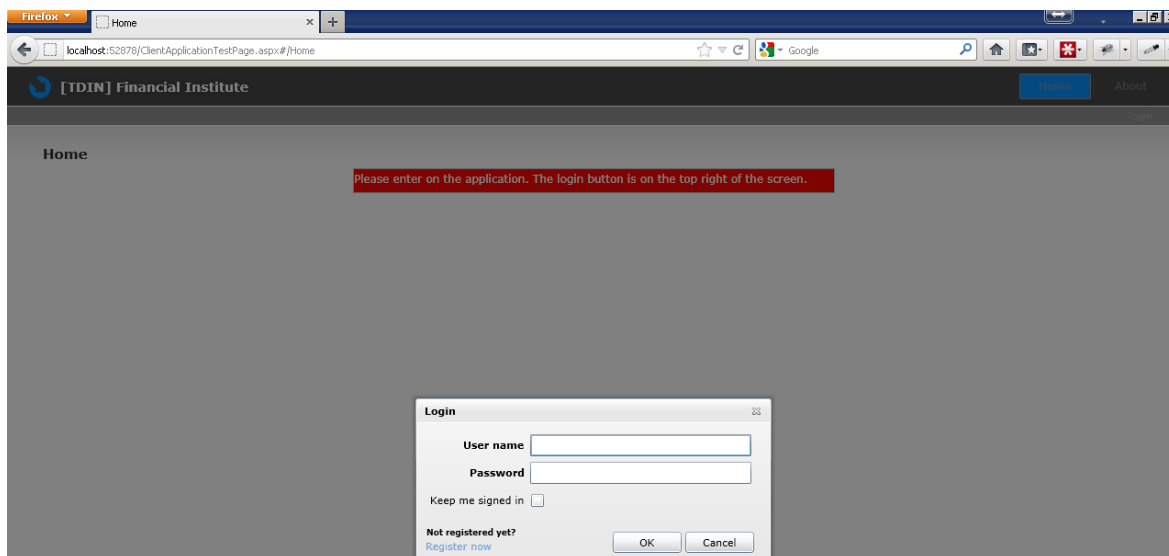


Figura 2 – Aplicação Web – Página inicial, com Login

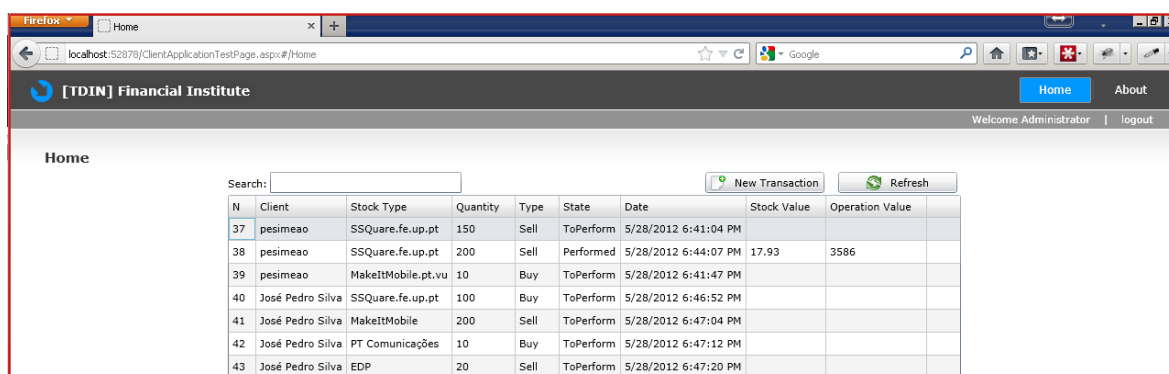


Figura 3 – Aplicação Web – Página inicial após login como administrador

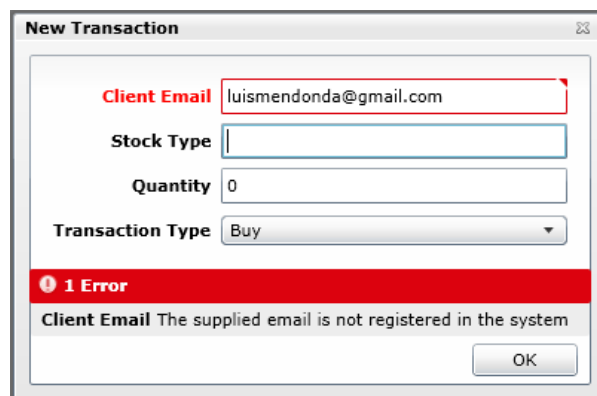
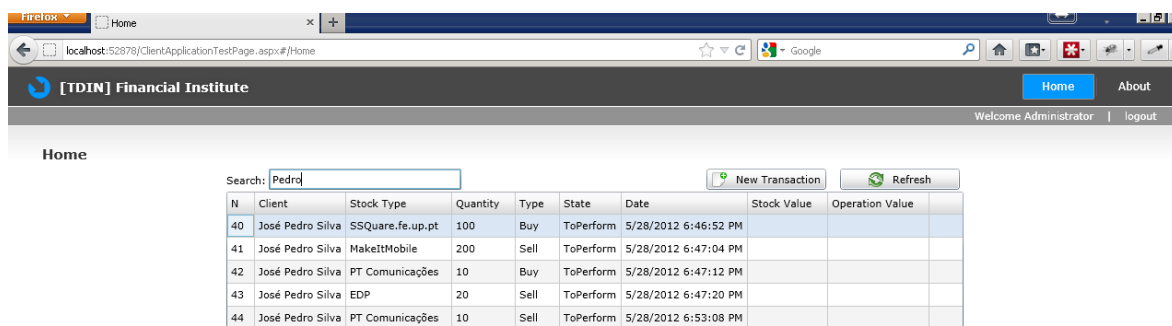


Figura 4 – Aplicação Web – Adição de nova transação como administrador

Relatório do 2º Projeto – Tecnologias de Distribuição e Integração



N	Client	Stock Type	Quantity	Type	State	Date	Stock Value	Operation Value
40	José Pedro Silva	SSquare.fe.up.pt	100	Buy	ToPerform	5/28/2012 6:46:52 PM		
41	José Pedro Silva	MakeItMobile	200	Sell	ToPerform	5/28/2012 6:47:04 PM		
42	José Pedro Silva	PT Comunicações	10	Buy	ToPerform	5/28/2012 6:47:12 PM		
43	José Pedro Silva	EDP	20	Sell	ToPerform	5/28/2012 6:47:20 PM		
44	José Pedro Silva	PT Comunicações	10	Sell	ToPerform	5/28/2012 6:53:08 PM		

Figura 5 – Aplicação Web – Aplicação de filtro do nome de cliente na listagem de transações

Aplicação Balcão

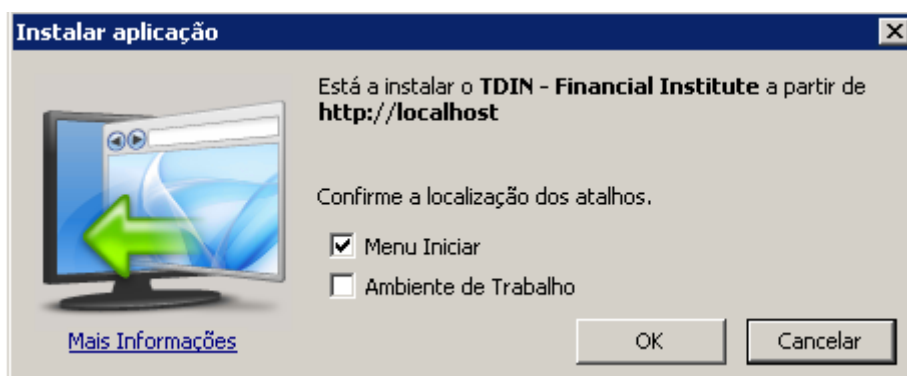


Figura 6 – Aplicação Balcão – Opção de instalação da aplicação

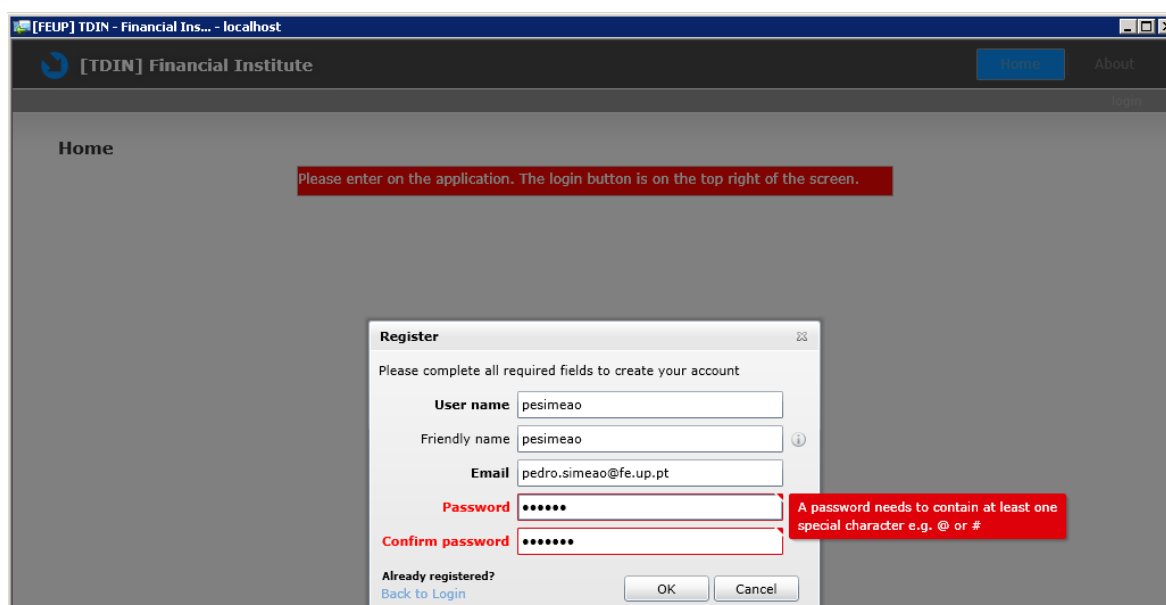


Figura 7 – Aplicação Balcão – Registo de um utilizador

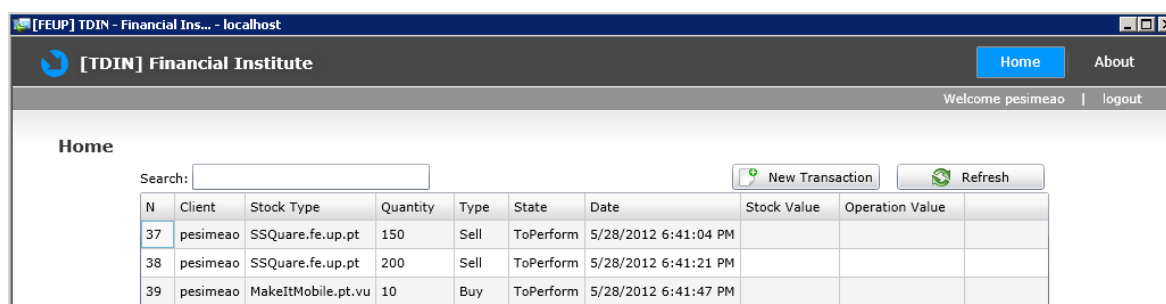


Figura 8 – Aplicação Balcão – Página inicial com início de sessão com utilizador normal

Relatório do 2º Projeto – Tecnologias de Distribuição e Integração

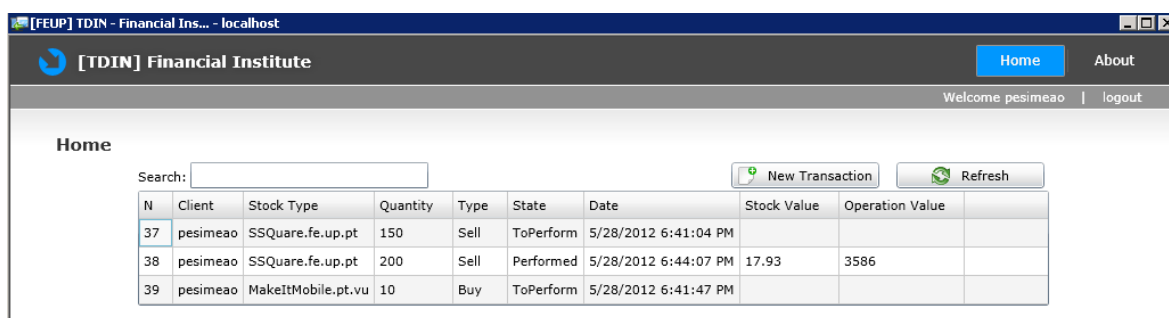


Figura 9 – Aplicação Balcão – Painel inicial com início de sessão com utilizador normal, após validação de uma transação

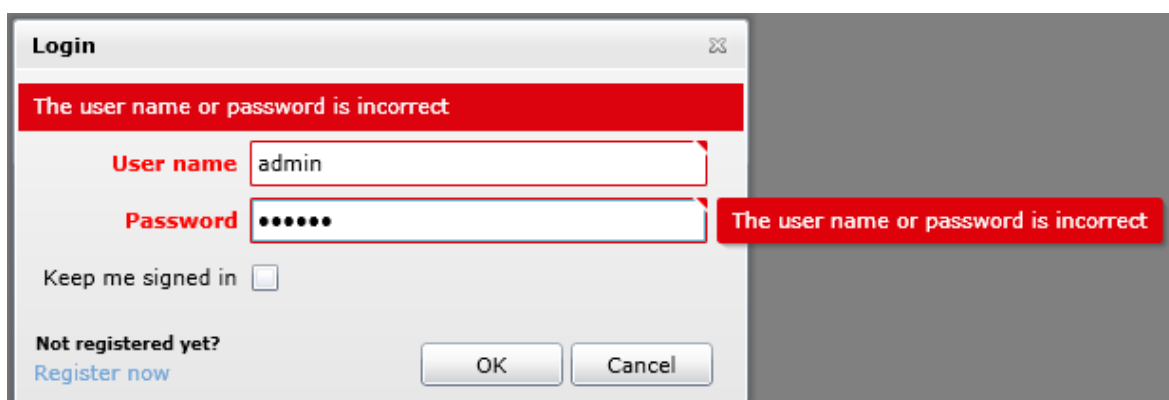


Figura 10 – Aplicação Balcão – Página de Login, com autenticação sem sucesso

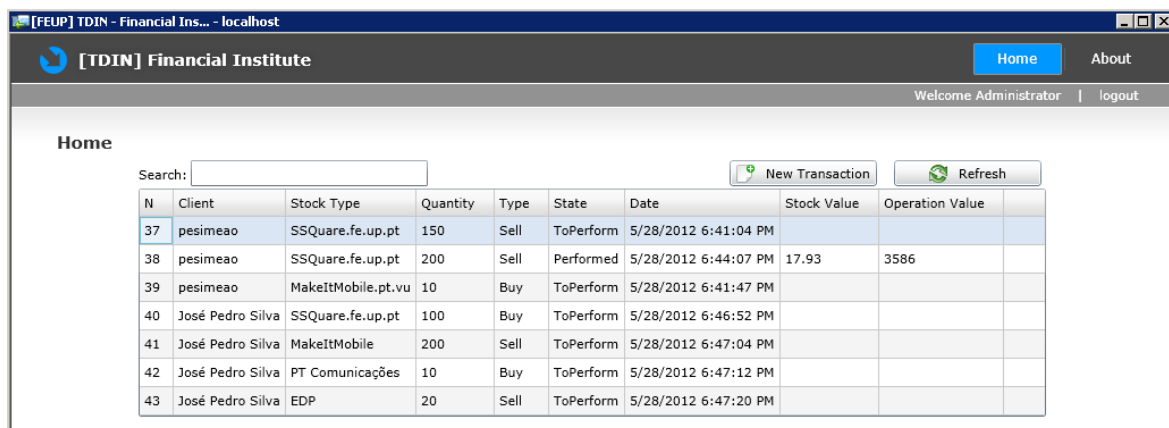


Figura 11 – Aplicação Balcão – Página inicial com início de sessão com administrador

Relatório do 2º Projeto – Tecnologias de Distribuição e Integração

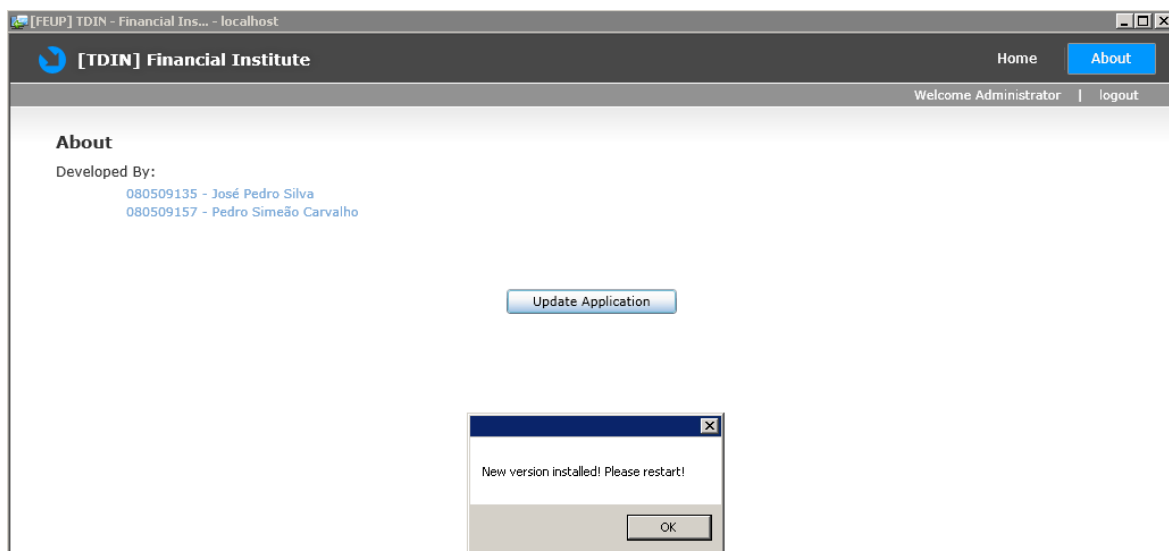


Figura 12 – Aplicação Balcão – Página e funcionalidade de atualização da aplicação

Aplicação Serviço Bolsista

The screenshot shows a web application window titled "TDIN - Financial Institute - Stock Service". It features a table with the following data:

ID	Client Name	Stock Type	Type	Quantity
38	pesimeao	SSquare.fe.up.pt	Sell	200
37	pesimeao	SSquare.fe.up.pt	Sell	150
39	pesimeao	MakeltMobile.pt.vu	Buy	10

To the right of the table are input fields for a new transaction:

- Trans. ID:
- Client Name:
- Stock Type:
- Quantity:
- Stock Value:
-

Figura 13 – Aplicação Serviço Bolsista – Página inicial com lista de transações pendentes

This screenshot shows the same application window as Figure 13, but with the following changes:

- The table data remains the same.
- The "Trans. ID" field now contains the value "38".
- The "Client Name" field now contains the value "pesimeao".
- The "Stock Type" field now contains the value "SSquare.fe.up.pt".
- The "Quantity" field now contains the value "Sell".
- The "Stock Value" field now contains the value "17.93".
- The "Save" button remains visible.

Figura 14 – Aplicação Serviço Bolsista – Seleção de uma transação e preenchimento do valor da mesma

Conclusão

Com o desenvolvimento deste projeto foi possível termos um contacto mais profundo com a *framework* .Net nomeadamente com o *Windows Communication Foundation* e os *bindings* *netTcpBinding* e *netMsmqBinding*. Esta tecnologia mostrou uma curva de aprendizagem muito acessível, conseguindo obter resultados em pouco tempo.

Por outro lado, a utilização de *Microsoft Silverlight* foi uma nova experiência que nos permitiu conhecer melhor esta tecnologia, e a sua interligação ao WCF trouxe à aplicação Web uma experiência de utilização muito agradável, na nossa opinião.

A modelação do problema foi, ao início, morosa mas revelou ser determinante para o sucesso do projeto. A partilha de estruturas entre os diversos serviços permitiu a construção de uma base de código de simples manutenção e correção.

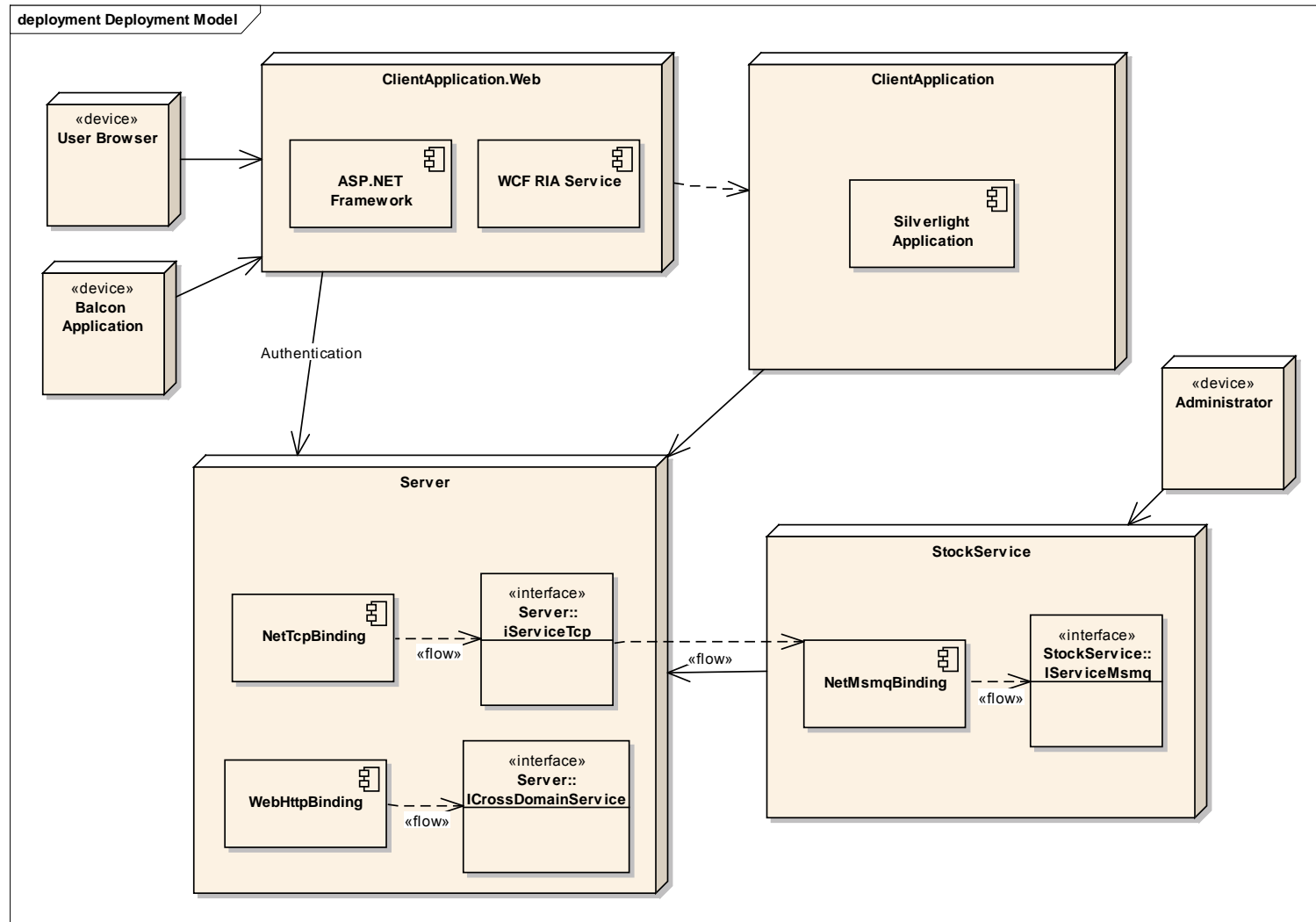
A correta estruturação e modelação do conjunto aplicacional permitiu a implementação de todos os requisitos especificados no enunciado e ainda algumas outras funcionalidades especificadas ao longo deste relatório.

Este projeto veio consolidar a ideia de que a distribuição, e consequente integração, das aplicações é cada vez mais importante e sem dúvida que as tecnologias aqui abordadas permitem que esse passo seja dado de forma segura e simples.

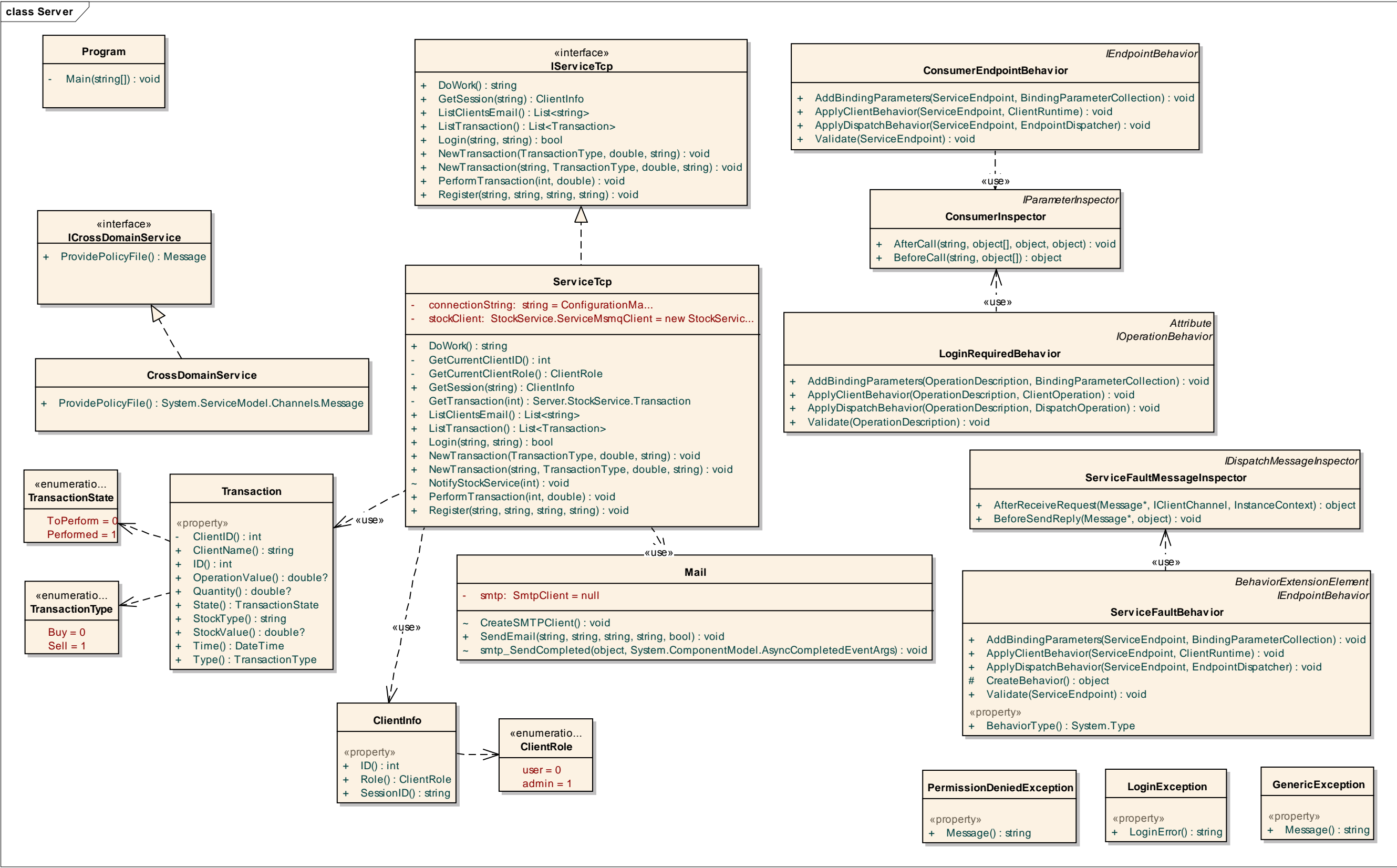
Referências

1. Windows Communication Foundation Bindings Overview, <http://msdn.microsoft.com/en-us/library/ms734662.aspx>, 09/05/2012
2. Estudo comparativo entre WCF Bindings, <http://blog.shutupandcode.net/?p=1085>, 09/05/2012
3. WCF Queued Messaging, <http://www.codeproject.com/Articles/34168/WCF-Queued-Messaging>, 18/05/2012
4. Sending Email C#, <http://social.msdn.microsoft.com/forums/en/netfxnetcom/thread/a75533eb-131b-4ff3-a3b2-b6df87c25cc8/>, 20/05/2012
5. WCF Parameters Inspector, <http://cgeers.com/2008/11/09/wcf-extensibility-parameter-inspectors/>, 12/05/2012
6. Using Silverlight to access a WCF Self-Hosted Application, <http://www.dotnetcurry.com/ShowArticle.aspx?ID=208>, 10/05/2012
7. Silverlight Custom Headers, <http://www.adilmughal.com/2011/10/wcf-custom-header-with-silverlight.html>, 10/05/2012
8. Silverlight as a Desktop Application, <http://www.silverlightshow.net/items/Silverlight-3-as-a-Desktop-Application-Out-of-Browser-Applications.aspx>, 08/05/2012
9. Silverlight – Rich Data Forms, <http://www.silverlightshow.net/items/Creating-Rich-Data-Forms-in-Silverlight-3-Introduction.aspx>, 12/05/2012

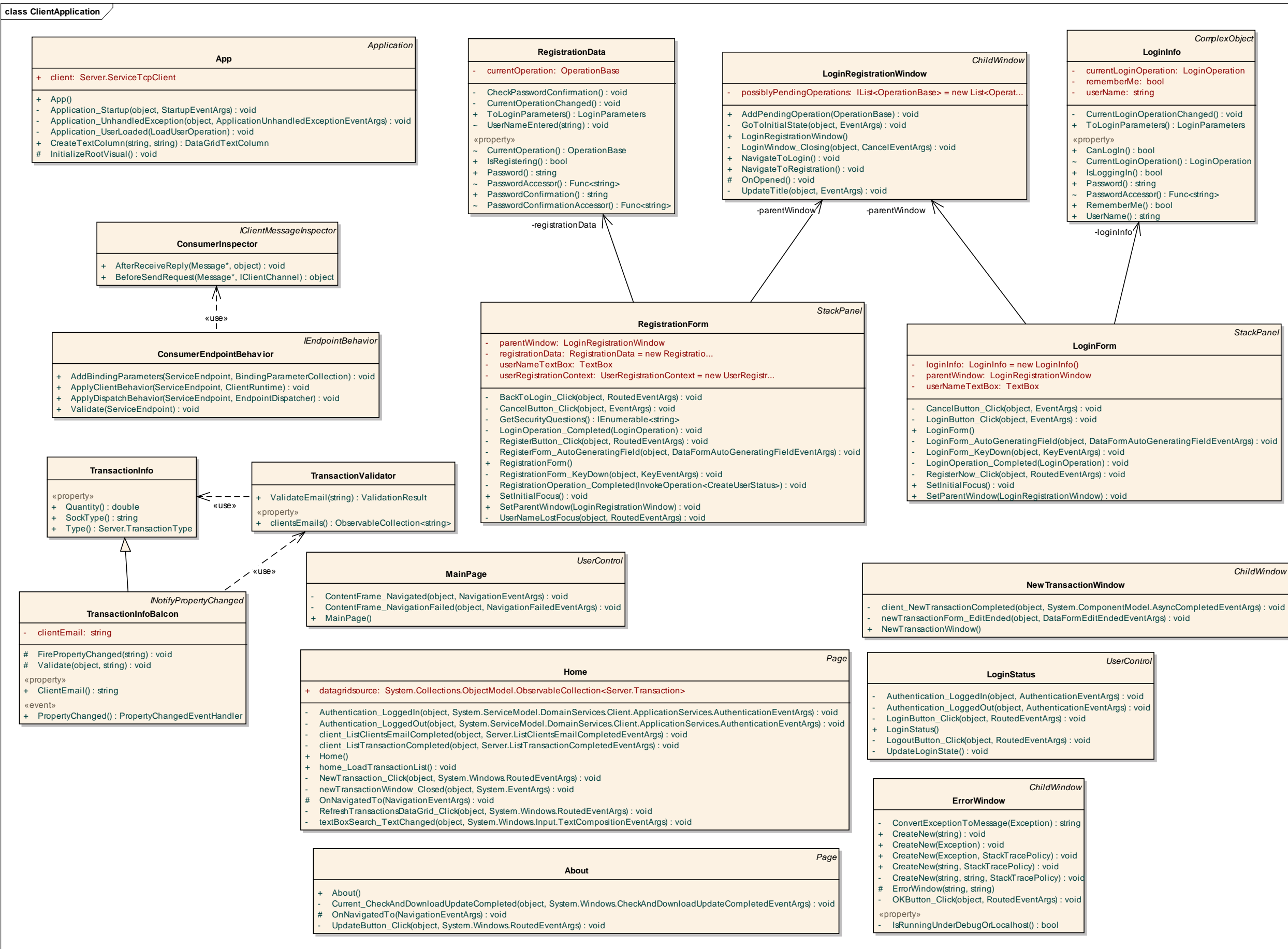
Anexo I



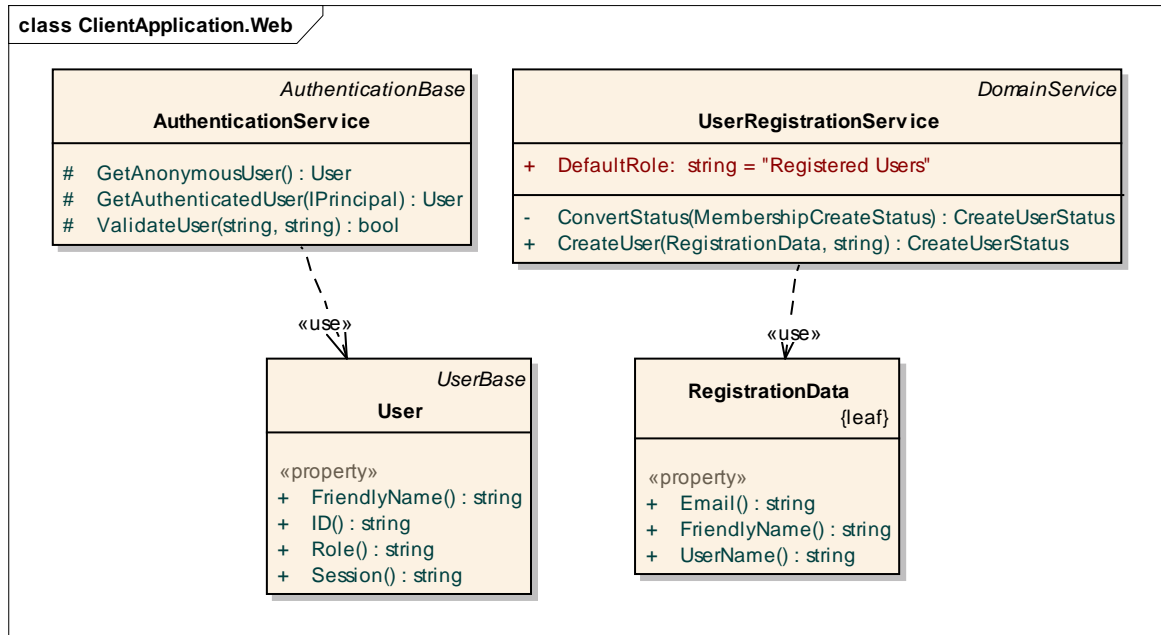
Anexo II



Relatório do 2º Projeto – Tecnologias de Distribuição e Integração



Relatório do 2º Projeto – Tecnologias de Distribuição e Integração



Relatório do 2º Projeto – Tecnologias de Distribuição e Integração

