

Enterprise Distributed Application

Trabalho #2



Mestrado Integrado em Engenharia Informática e Computação
Tecnologias de Distribuição e Integração
EIC0077-2S

João Carlos Teixeira de Sá - 201107925 (ei11142@fe.up.pt)
João Pedro Matos Teixeira Dias - 201106781 (ei11137@fe.up.pt)

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

24 de Maio de 2015

Conteúdo

1	Introdução	3
2	Tecnologias	3
3	Arquitetura	3
4	Front-end	6
4.1	Interface gráfica da aplicação da loja	6
4.2	Interface gráfica da aplicação do armazém	8
4.3	Interface gráfica da aplicação <i>web</i>	9
4.4	Casos de Uso	13
5	Conclusão	13
6	Recursos	14
6.1	Bibliografia	14
6.2	<i>Software</i>	14

1 Introdução

O presente documento apresenta o desenvolvimento do projeto *Enterprise Distributed Application* para a unidade curricular de Tecnologias de Distribuição e Integração.

O projeto tem como objetivo o desenvolvimento de um sistema, *Enterprise Distributed Application*, capaz de gerir compras e vendas de livros de uma editora (*diginotes*).

O sistema é composto por um servidor presente numa loja que persiste a informação relativa aos livros existentes para venda tais como o título, preço e o stock disponível tanto na loja como em armazém. Este servidor encontra-se ligado à *internet* e está sempre disponível.

As compras de um livro por parte de um consumidor podem ser feitas diretamente na loja ou através de uma aplicação *web*. Caso exista stock do livro pretendido o mesmo é imediatamente entregue ao consumidor no caso da compra ser feita na loja ou enviado pelo correio caso seja efetuado o pedido através da aplicação *web*; no caso de não existir stock a loja deve enviar uma mensagem para o armazém a pedir o envio do livro em questão numa quantidade 10 vezes superior ao pretendido pelo cliente.

2 Tecnologias

O projeto foi desenvolvido usando a *framework Windows Communication Foundation* para o desenvolvimento da API disponibilizada da loja bem como a tecnologia *Microsoft Message Queuing* para o envio e receção das mensagens de pedido de stock de livros ao armazém utilizando em ambos os casos a linguagem de programação C#. Além disso foram também utilizadas as linguagens de programação PHP, HTML e Javascript para o desenvolvimento da aplicação *web*.

Adicionalmente foi utilizado para persistência de dados a tecnologia de base de dados *MongoDB* e para o desenvolvimento da interfaces de utilizador da loja e do armazém a tecnologia *Windows Presentation Foundation* usando a linguagem *XAML*.

Por último, para gestão de dependências do projeto foi usado o *NuGet* e para gestão de versões foi utilizada a ferramenta *Git* usando a plataforma *GitHub*.

3 Arquitetura

O sistema apresenta a estrutura típica de um projeto baseado em *Windows Communication Foundation*, como representado na fig. 3, contando assim com um servidor da loja que disponibiliza um API que pode ser acedida

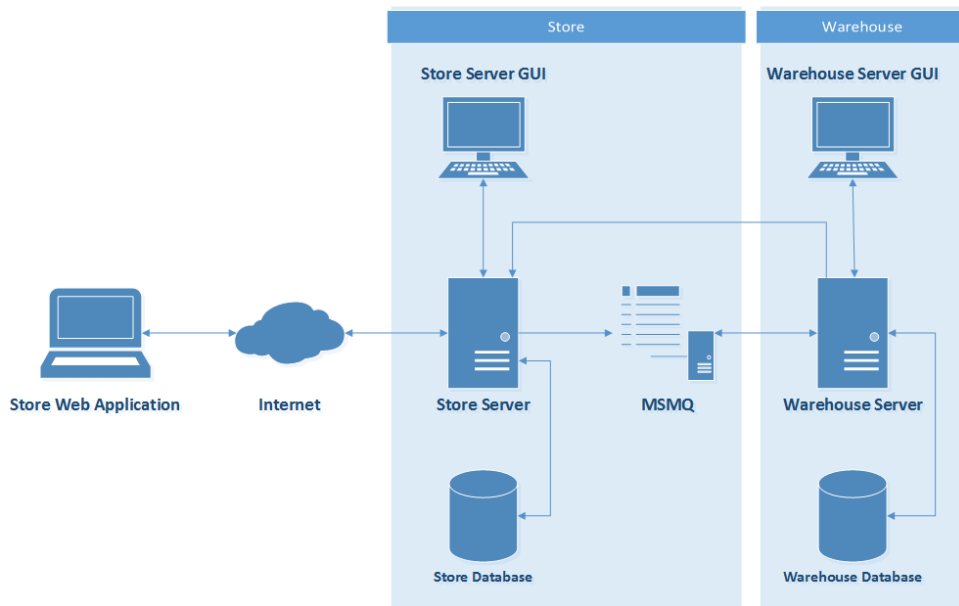


Figura 1: Arquitetura do Sistema.

por aplicações clientes, neste caso uma aplicação com interface gráfica presente na loja e uma aplicação disponível na *web*. Além disso existe também um servidor no armazém que recebe e trata mensagens de pedidos de stock provenientes da loja. Acrescenta-se ainda uma persistência de dados tanto na loja como no armazém existindo uma base de dados MongoDB associada a cada um dos servidores.

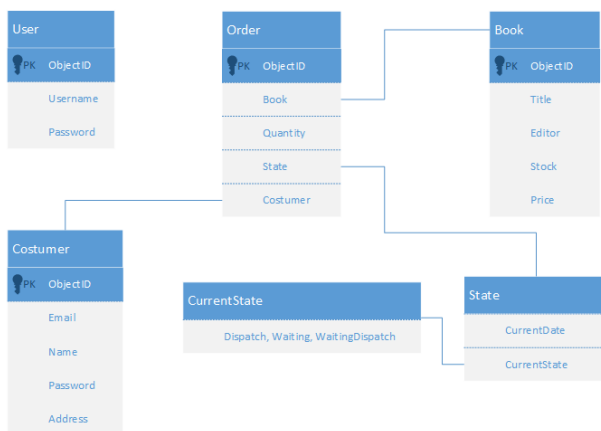


Figura 2: Esquema da base de dados da loja.

??
??

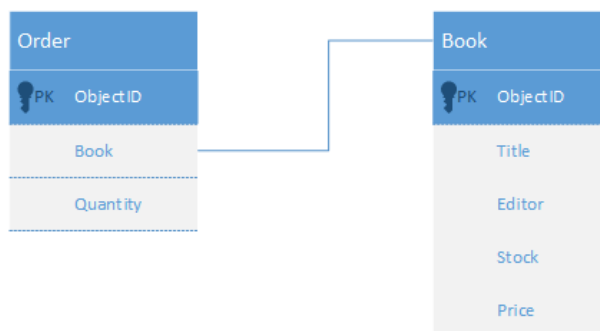


Figura 3: Esquema da base de dados do armazém.

Apresenta-se também com uma divisão modular apresentando:

- Módulo comum
 - Definição das estruturas de dados partilhadas entre cliente e servidor.
- Módulo cliente
 - Aplicação com interface gráfica para a loja.
 - Aplicação *web*.
- Módulo servidor (da loja)
 - Aplicação com lógica do sistema, transações e persistência de dados.
 - Disponibilização de uma API para consulta pelas aplicações cliente.
- Módulo servidor (do armazém)
 - Aplicação para receção e tratamento de mensagens de pedidos de stock por parte da loja.
 - Aplicação com interface gráfica para os funcionários do armazém.

O sistema ainda utiliza objetos para representação das transações (*order*), dos livros (*book*) e para a informação dos clientes (*costumer*) bem como dos utilizadores das aplicações com interface gráfica da loja e do armazém (*user*).

4 Front-end

4.1 Interface gráfica da aplicação da loja

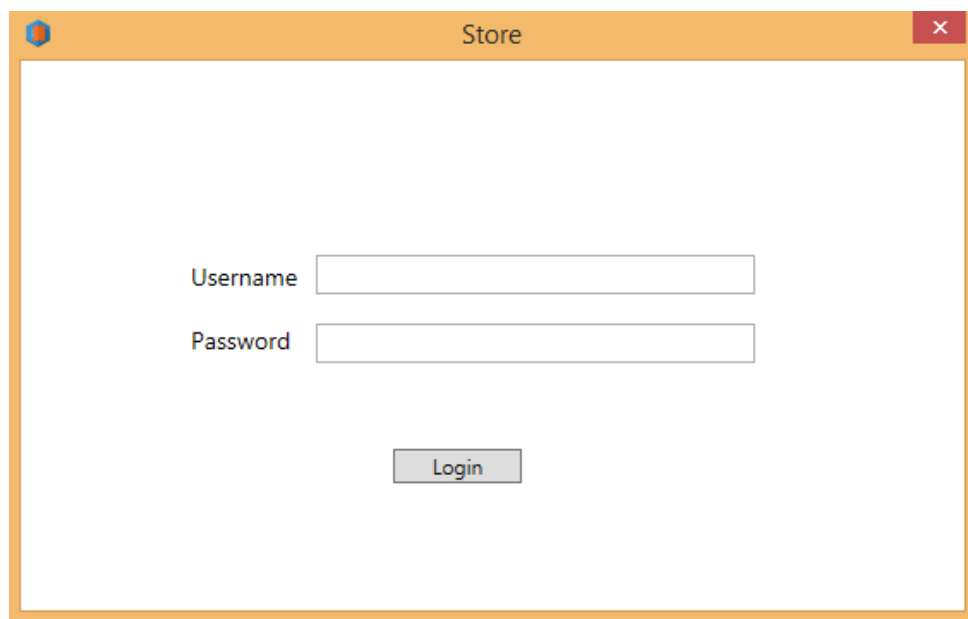


Figura 4: Interface gráfica da aplicação da loja - *Login*.

Ao nível da interface gráfica desenvolvida para a aplicação presente na loja é disponibilizada numa primeira fase um ecrã de *login*, fig. 4, onde o funcionário da loja pode entrar no sistema.

Após o funcionário fazer *login* no sistema são disponibilizadas as opções de criar uma venda e de verificar quais as encomendas efetuadas através da aplicação *web* podendo estas serem marcadas como enviadas após o envio para o cliente, fig. 5 e 6.

Por último, no momento em que é criada uma venda o sistema questiona o funcionário se pretende imprimir uma fatura, fig. 7, sendo esta exportada para um ficheiro pdf em caso afirmativo.

The screenshot shows a window titled "Store" with a standard Windows-style title bar (orange background, minimize, maximize, and close buttons). The window is divided into two main sections. On the left, there are three input fields: "Book Name", "Quantity", and "Client". The "Client" field has a checkbox labeled "New" next to it. Below these fields are two buttons: "Create Order" and "Check Pending Orders". On the right side of the window, there is a large, empty rectangular area, likely intended for displaying a list of books or orders.

Figura 5: Interface gráfica da aplicação da loja - Ecrã principal.

This screenshot is similar to the previous one, but the large rectangular area on the right now contains text. It displays "Book: Harry Potter and the Chamber of Secrets" and "Client: Joao" on two lines. The rest of the interface, including the input fields and buttons, remains the same.

Figura 6: Interface gráfica da aplicação da loja - Ecrã principal com encomenda pendente.

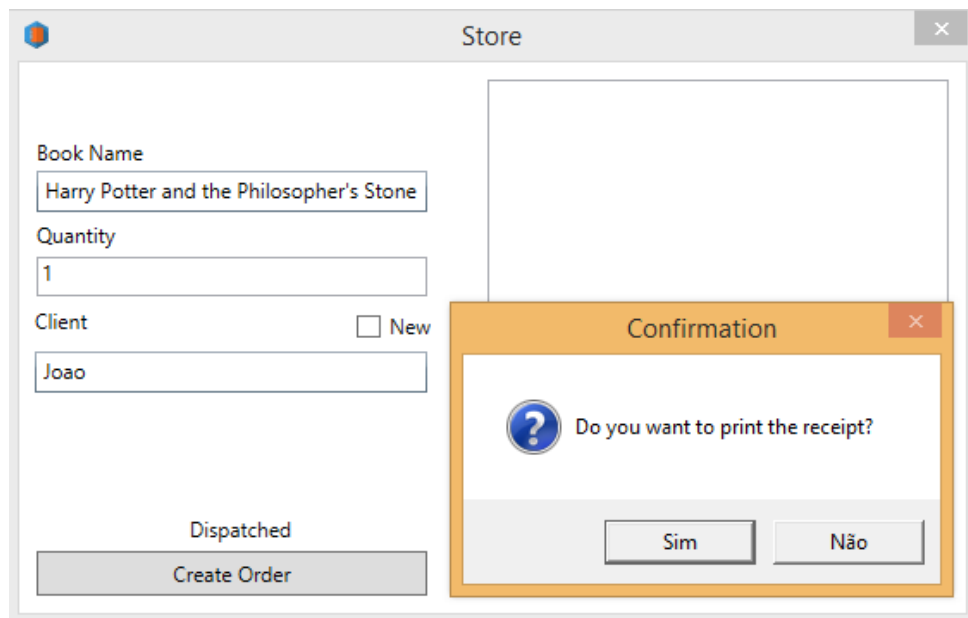


Figura 7: Interface gráfica da aplicação da loja - Confirmação de impressão de fatura de uma venda.

4.2 Interface gráfica da aplicação do armazém

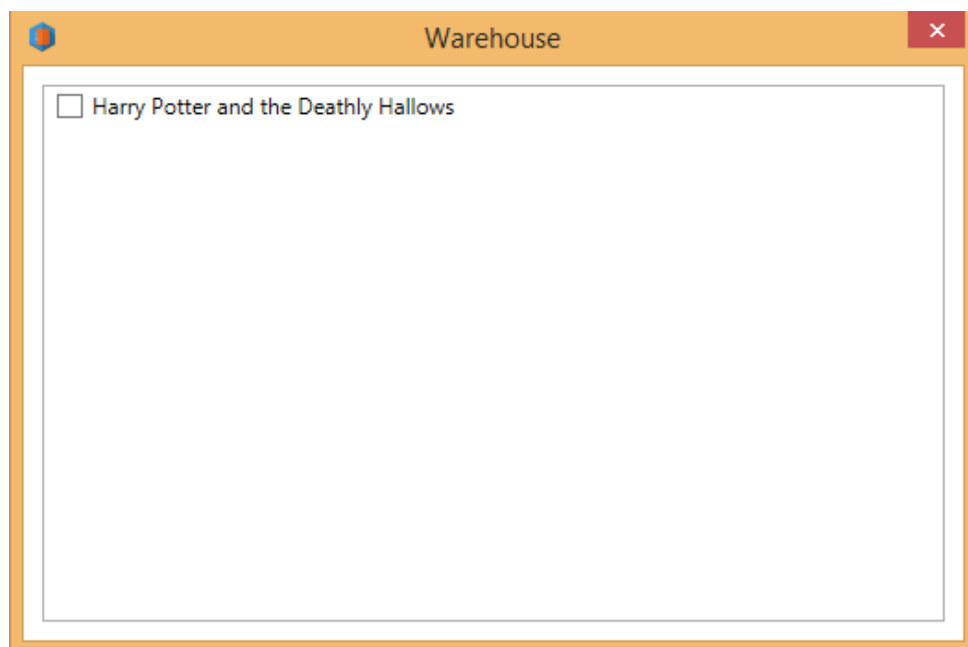


Figura 8: Interface gráfica da aplicação do armazém - Ecrã principal.

Ao nível da interface gráfica desenvolvida para a aplicação presente no armazém esta contém apenas uma janela em que é disponibilizada uma lista de pedidos de livros por parte da loja devendo o funcionário do armazém colocar um visto num pedido quando os livros correspondentes a esse pedido forem enviados para a loja, fig. 8.

4.3 Interface gráfica da aplicação *web*

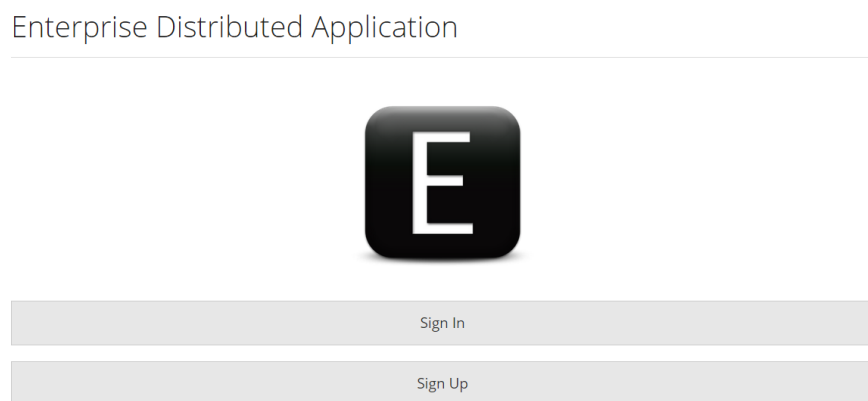


Figura 9: Interface gráfica da aplicação *web* - Ecrã inicial.

Enterprise Distributed Application [Sign In](#)

Email	<input type="text" value="Email"/>
Password	<input type="password" value="Password"/>
<input type="button" value="Sign In"/>	

Figura 10: Interface gráfica da aplicação *web* - *Login*.

Enterprise Distributed Application [New Customer](#)

Name	<input type="text" value="Name"/>
Address	<input type="text" value="Address"/>
E-Mail	<input type="text" value="E-mail"/>
Password	<input type="password" value="Password"/>
<input type="button" value="Sign Up"/>	

Figura 11: Interface gráfica da aplicação *web* - Registo.

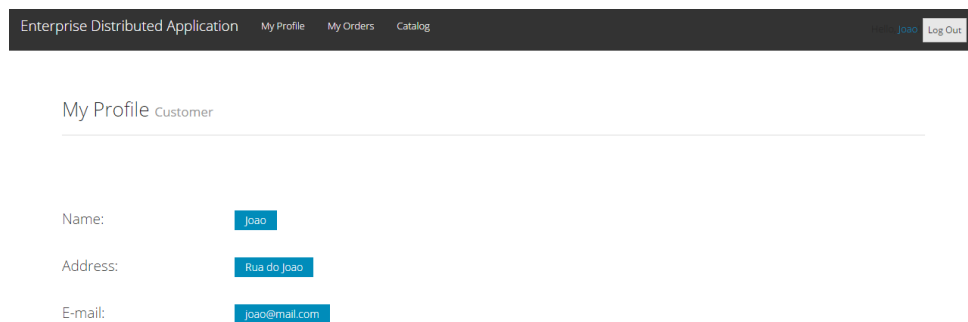


Figura 12: Interface gráfica da aplicação *web* - Perfil.

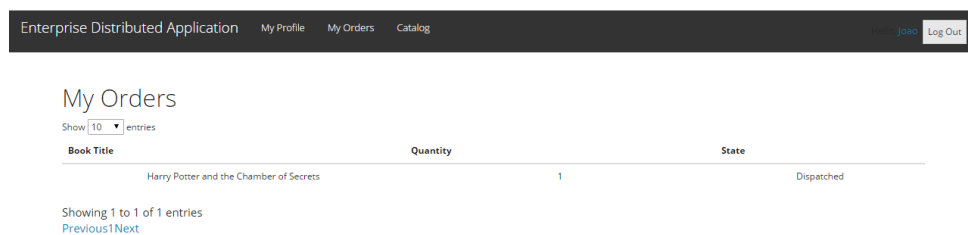


Figura 13: Interface gráfica da aplicação *web* - Encomendas efetuadas.

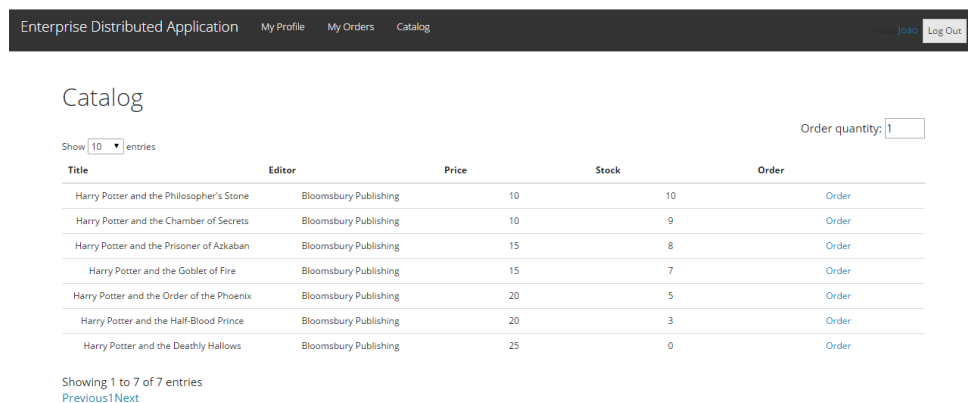


Figura 14: Interface gráfica da aplicação *web* - Consulta dos livros disponíveis em catálogo e possibilidade de fazer encomenda.

Ao nível da interface gráfica desenvolvida para a aplicação *web* numa primeira fase um ecrã onde é possível escolher entre fazer *login* ou registar caso o utilizador ainda não seja cliente da loja, fig. 9.

Após o cliente fazer *login* no sistema são disponibilizadas as opções de ver o seu perfil onde são apresentados os seus dados pessoais (fig 11), consultar as encomendas efetuadas (fig 13) e consultar o catálogo dos livros disponíveis para venda com a possibilidade fazer uma encomenda (fig 14).

4.4 Casos de Uso

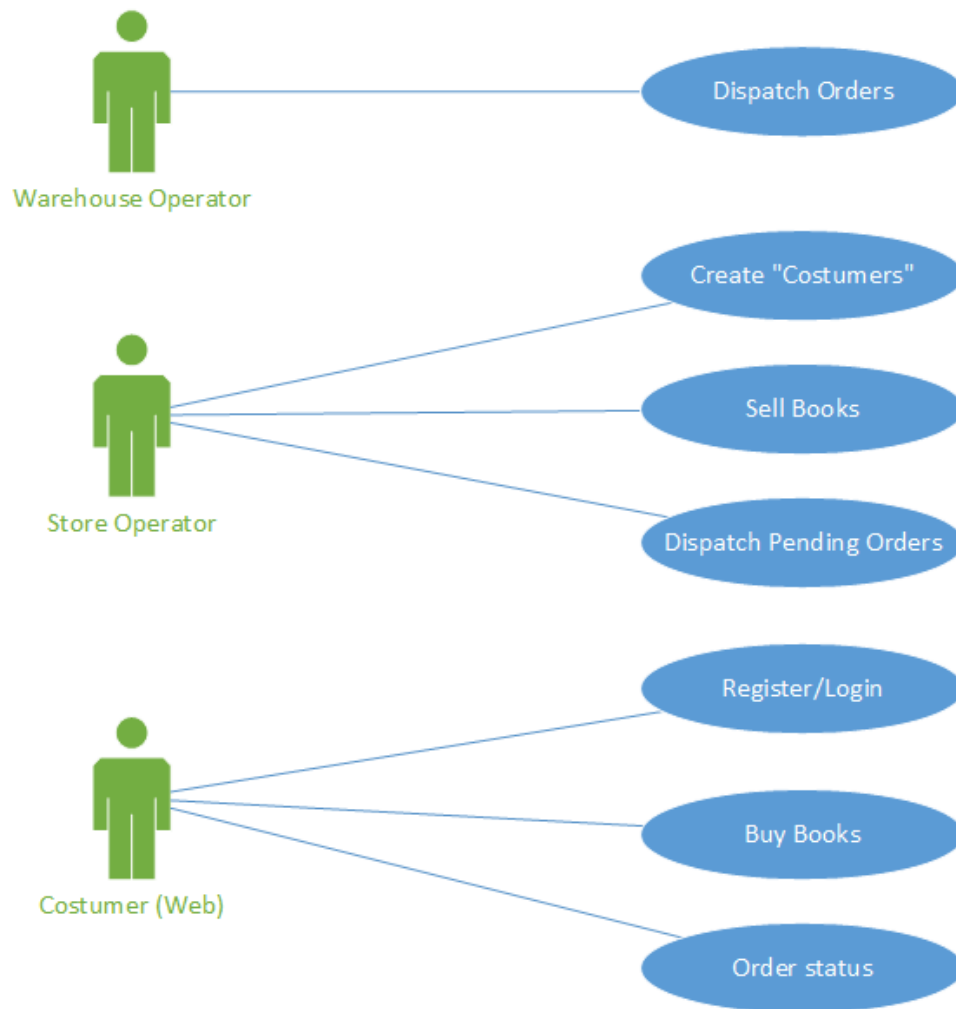


Figura 15: Diagrama de casos de uso.

5 Conclusão

O sistema encontra-se desenvolvido com todas as funcionalidades solicitadas no enunciado do projeto, possibilitando ao utilizador uma utilização total do sistema.

Testes

Para testar o correto funcionamento do sistema foram efetuadas várias experiências com diferentes clientes ligados a aplicação servidor, assim como, foram testados vários casos de falha ou do cliente e/ou servidor e garantida a persistência dos dados.

Deploy

O sistema pode ser utilizado colocando os servidores da loja e do armazém a correr e abrindo a aplicação cliente da loja e/ou a aplicação cliente *web* disponível através do endereço /WebApp/index.php. Pode ser também aberto o projeto de *Visual Studio* (*EnterpriseDistributedApplication.sln*) e fazer *Start* ao mesmo na interface do *IDE*, no entanto, este não inclui a aplicação *web*.

6 Recursos

6.1 Bibliografia

Windows Communication Foundation .NET Framework 4.6 and 4.5, <https://msdn.microsoft.com/en-us/library/dd456779%28v=vs.110%29.aspx>.

Message Queuing (MSMQ), <https://msdn.microsoft.com/en-us/library/ms711472%28v=vs.85%29.aspx>.

Distribution and Integration Technologies, Miguel Monteiro, Faculdade de Engenharia da Universidade do Porto, paginas.fe.up.pt/~apm/TDIN/.

6.2 Software

Visual Studio 2013 Ultimate, Microsoft, <http://www.visualstudio.com/>.

MongoDB, <https://www.mongodb.org/>.

SQLiteClient, Community, <http://www.nuget.org/packages/Community.CsharpSqlite.SQLiteClient/>.

NuGet, <http://www.nuget.org/>.

GitHub, <http://github.com/>.

MongoLab, <https://mongolab.com/>.