

# Tecnologias de Distribuição e Integração

Enterprise Distributed Applications: Relatório de Projeto



Universidade do Porto  
Faculdade de Engenharia  
**FEUP**

**Grupo de trabalho:**

Hugo Cardoso - 201105625 - ei11154@fe.up.pt

Vasco Gomes - 201106906 - ei11161@fe.up.pt

Faculdade de Engenharia da Universidade do Porto  
Mestrado Integrado em Engenharia Informática e Computação  
Rua Roberto Frias, sn, 4200-465 Porto, Portugal  
Responsável: António Miguel Pontes Pimenta Monteiro

3 de Junho de 2015

## Conteúdo

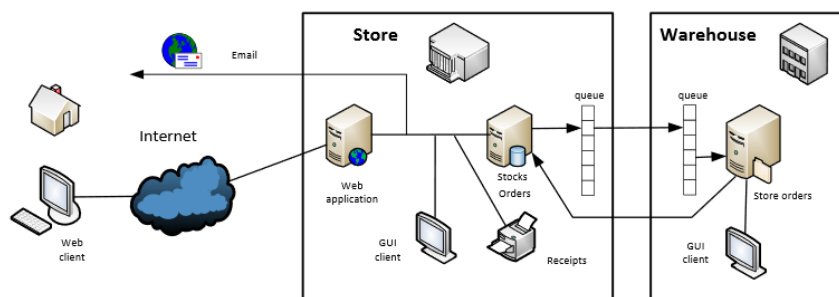
<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Arquitetura</b>	<b>3</b>
2.1	Módulos . . . . .	3
2.2	Estruturação de Classes . . . . .	4
<b>3</b>	<b>Desenvolvimento</b>	<b>6</b>
3.1	Funcionalidades Implementadas . . . . .	6
3.2	Modo de Funcionamento . . . . .	6
<b>4</b>	<b>Conclusão e Perspetivas Futuras</b>	<b>11</b>
<b>5</b>	<b>Instruções para Utilização</b>	<b>12</b>
<b>6</b>	<b>Referências</b>	<b>13</b>

# 1 Introdução

Uma editora de livros pretende desenvolver um sistema para coordenação de vendas, encomendas e gestão de stock. A editora detém duas propriedades: uma loja, com uma área pública de exposição, e um armazém. Pretende-se, com este projeto, desenvolver uma série de aplicações para a gestão de ambas as propriedades por parte dos seus funcionários, assim como uma aplicação web que permita a compra de livros online.

A loja é responsável pelo alojamento de um servidor (que controla a aplicação web) e persistência de todo o tipo de dados relativos a ambas as propriedades (livros, preços e stock atual, entre outros). Este servidor deve manter uma ligação à Internet, estando sempre disponível.

O armazém é responsável pelo alojamento de outro servidor, fisicamente diferente, e pode apenas estar disponível durante as horas de operação do edifício. O armazém deve realizar o restock de livros da loja, comunicando, para isso, com o servidor da mesma.



**Figura 1.1:** Representação gráfica da topologia proposta.

## 2 Arquitetura

Optou-se, para toda a implementação, pela utilização de C# e *Windows Communication Foundation* (WCF).

### 2.1 Módulos

De forma a seguir um conjunto de boas práticas relativas à arquitetura, a aplicação desenvolvida foi estruturada em quatro módulos, ou projetos, individuais. A saber: *StoreServer*, módulo central da loja responsável por todas as operações de compra e encomenda de livros; *StoreService*, serviço disponibilizado através do módulo *StoreServer*; *StoreApp*, aplicação local para funcionários da loja; *Warehouse*, aplicação local para funcionários do armazém.

Serão, de seguida, apresentados os módulos em maior detalhe:

- **StoreServer:** Servidor central da loja, acessível por ambos clientes e funcionários. Responsável pelo alojamento de um serviço que disponibiliza todos os métodos comuns de compras e encomendas de livros. É, também, neste módulo que a aplicação web se encontra.

Para persistir toda a informação, optou-se pela utilização de uma base de dados SQL Server Express LocalDB com as tabelas *Book* (incluindo stock atual disponível na loja), *BookOrder* (ordem de compra de um livro; mantém um estado identificador, podendo ser uma venda já consumida ou uma encomenda ainda a ser processada) e *BookRequest* (pedido de stock ao armazém; mantém, também, um identificador do seu estado atual).

- **StoreService:** Serviço responsável por toda a lógica de compras e encomendas de livros. É neste módulo que são efetuadas todas as conexões à base de dados do sistema. Deste modo, um único serviço disponibiliza todos os métodos de processamento de novas compras, evitando implementações e definições de estruturas iguais em módulos diferentes.
- **StoreApp:** Aplicação utilizada por funcionários da loja para o registo de novas compras ou encomendas, e aceitação de reposições de stock por parte do armazém. Deve ser notado que a aplicação detém apenas uma interface gráfica para fácil disponibilização de todas as funcionalidades, sendo que todos as ações são comunicadas (e processadas) no serviço do módulo *StoreServer*.

É, de igual modo, nesta aplicação que é disponibilizado um serviço de simulação de impressão de recibos, chamado sempre que uma ordem de compra é registada no sistema.

- **Warehouse:** Aplicação utilizada por funcionários do armazém para o processo de renovação de stock. Dispõe uma lista de todos os pedidos de reposição de stock ainda não completos, de forma a que possam ser facilmente processados.

O módulo *Warehouse* é o único que comunica, de forma direta, com o serviço do armazém, registando todas as ações para que possam ser processadas assim que o serviço é iniciado (através de uma fila de mensagens).

## 2.2 Estruturação de Classes

Cada um dos módulos criados é, por sua vez, estruturado em várias classes. Tentou-se, ao longo do desenvolvimento, manter uma estruturação funcional de modo a que cada classe seja responsável por uma funcionalidade específica e possa facilmente ser modificada sem afetar de modo significativo os outros componentes do sistema.

Apresenta-se, de seguida, uma visão mais detalhada das classes implementadas.

### Módulo StoreServer

- *Index*: Aplicação web disponibilizada a clientes. Mantém o objeto de conexão ao serviço *StoreService*.

### Módulo StoreService

- *IStoreServ*: Interface implementada pela classe *StoreServ*, de forma a omitir a implementação de todos os métodos da mesma a clientes externos.
- *StoreServ*: Serviço responsável pelo processamento de toda a lógica do módulo.
- *DatabaseLayer*: Representa um *data access layer* (DAL). Responsável por todos os acessos diretos à base de dados, adicionando um maior nível de abstração à classe *StoreServ*.
- *MailHelper*: Utilizado para auxiliar o envio de e-mails, de modo a que todas as configurações do cliente SMTP possam ser abstraídas do serviço.
- *StoreBook*: Representa um *data access object* (DAO) para acesso a objetos do tipo *Book*.
- *StoreOrder*: Representa um DAO para acesso a objetos do tipo *BookOrder*.
- *StoreRequest*: Representa um DAO para acesso a objetos do tipo *BookRequest*.
- *IStoreCallback*: Interface implementada por todos os clientes do serviço que pretendam receber informação de eventos acionados pelo mesmo.

### Módulo StoreApp

- *Main*: Interface gráfica da aplicação, com a qual um funcionário da loja interage e realiza ações (processadas, depois, no serviço do módulo *StoreServer*).
- *StoreServCallback*: Utilizado para a subscrição a eventos acionados pelo serviço da loja.

## Módulo Warehouse

- *Main*: Interface gráfica da aplicação, com a qual um funcionário do armazém interage e realiza ações (processadas, depois, no serviço do armazém após leitura da fila de mensagens). É, também, nesta classe que é feita a subscrição a eventos acionados pelo serviço da loja.

## 3 Desenvolvimento

### 3.1 Funcionalidades Implementadas

Todas as funcionalidades propostas no enunciado foram implementadas. De referir:

- Aplicação web - consulta de todos os livros e respetivos stocks atualmente disponíveis na loja.
- Aplicação web - registo de uma nova encomenda. Se uma encomenda pode ser completa no momento em que é registada (o stock atualmente disponível na loja é suficiente para cobrir a nova encomenda), deve ser enviado um email ao seu respetivo cliente.
- Aplicação web - consulta do estado atual de uma encomenda. Os clientes devem, também, ter acesso à consulta de todas as suas vendas previamente registadas no sistema.
- Aplicação loja - consulta de todos os livros e respetivos stocks atualmente disponíveis na loja.
- Aplicação loja - registo de uma nova venda ou encomenda.
- Aplicação loja - impressão de um recibo após o registo de uma nova venda. Quando uma venda é completa no momento em que é registada (o stock atualmente disponível na loja é suficiente para cobrir a nova compra), deve ser realizada a simulação de impressão de um recibo.
- Aplicação loja - processo de reposição de stock por parte do armazém. Após a reposição do stock ter sido efetuada por um funcionário do armazém e antes da nova quantidade ser adicionada ao stock atual de livros, deve, primeiro, ser aceite por um funcionário da loja. Se a quantidade adicionado for suficiente para completar encomendas previamente em espera, devem ser enviados emails aos seus respetivos clientes.
- Aplicação armazém - consulta de todos os pedidos de reposição de livros e respetivos stocks necessários. O armazém deve receber, em *real-time*, todos os novos pedidos de reposição de stock assim que estes são lidos pela fila de mensagens do serviço do armazém.
- Aplicação armazém - reposição do stock associado a um pedido.

### 3.2 Modo de Funcionamento

De seguida são apresentadas algumas imagens que demonstram os principais fluxos de funcionamento da aplicação:

## Aplicação Web

The screenshot shows the home page of a web application titled "Distributed Store". At the top, there is a navigation bar with "TDIN 14/15" on the left, "About" and "Help" links in the center, and "Built by Hugo Cardoso and Vasco Gomes" on the right. Below the navigation bar, the main heading "Distributed Store" is displayed, followed by the text "Created in C# using ASP.NET". The page is divided into three main sections. The first section, "New Order", contains a form with the instruction "Complete all the fields, and have your order ready in a matter of seconds". The form fields are "Book ID", "Quantity", "Name", "Address", and "Email", each with a corresponding input box. A blue "SUBMIT" button is at the bottom of the form. The second section, "Order status", contains the instruction "NEW: check where your order is and when it is expected to arrive". It has input boxes for "Email" and "ID", and a "CHECK STATUS" button. The third section, "Available books", contains the instruction "We have a large collection available, take a look around" and a table with the following data:

id	title	quantity	price
1	Harry Potter	2	10
2	50 Shades of Grey	62	10
3	Lord of the Rings	30	10

**Figura 3.2.1:** Ecrã inicialmente apresentado aos clientes aquando da sua chegada à aplicação.

This is a close-up screenshot of the "New Order" form. It features the heading "New Order" and the instruction "Complete all the fields, and have your order ready in a matter of seconds". The form contains five input fields: "Book ID", "Quantity", "Name", "Address", and "Email". Each input field has a blue underline, indicating that all fields are mandatory. A blue "SUBMIT" button is located at the bottom of the form.

**Figura 3.2.2:** Secção de registo de compra. Deve ser notado que todos os campos são de carácter obrigatório.



Available books			
We have a large collection available, take a look around			
id	title	quantity	price
1	Harry Potter	2	10
2	50 Shades of Grey	62	10
3	Lord of the Rings	30	10

**Figura 3.2.3:** Listagens de todos os livros e respetivos stocks atualmente disponíveis na loja.

### Order status

NEW: check where your order is and when it is expected to arrive

Email

ID

CHECK STATUS

**Figura 3.2.4:** Secção de consulta do estado atual de uma encomenda. Optou-se, para este cenário, por emular o ambiente encontrado em sistemas reais de transporte de encomendas (através de um identificador único).

## Aplicação Loja

Store

Add Order

id	title	price	quantity
1	Harry Potter	10	5181
2	50 Shades of Grey	10	10527
3	Lord of the Rings	10	1020

Book Information

Id:

Quantity:

Client Information

Name:

Address:

Email:

Book Requests

id	book_id	quantity	fulfilled
87	3	500	0
89	3	500	0

**Figura 3.2.5:** Ecrã inicialmente apresentado aos funcionários da loja aquando da sua chegada à aplicação. Permite o registo de novas compras ou encomendas, e consulta de todo o tipo de stocks e pedidos.

```
C:\WINDOWS\system32\cmd.exe

=====
===== Distributed Store Printer Service =====
=====
- Initializing Printer...
- Status: Online

Order of Id 125 correctly processed.
Thank you for your preference.
Please consult our website for more details.

=====
```

**Figura 3.2.6:** Simulação de impressão de recibos pela aplicação da loja.

## Aplicação Armazém



The screenshot shows a window titled "Warehouse" with a table containing two rows of data. The table has four columns: id, book\_id, quantity, and fulfilled. The first row has values 87, 3, 500, and 0. The second row has values 89, 3, 500, and 0. The table is styled with a blue header and a light blue body.

id	book_id	quantity	fulfilled
87	3	500	0
89	3	500	0

**Figura 3.2.7:** Ecrã inicialmente apresentados aos funcionários da loja aquando da sua chegada à aplicação. Permite a consulta e processo de pedidos de reposição de stock.

## 4 Conclusão e Perspetivas Futuras

Analisando o trabalho desenvolvido, chega-se à conclusão que se cumpriram todos os objetivos proposto para este ponto de avaliação, sendo que todas as funcionalidades pedidas foram implementadas com sucesso.

Conclui-se que o conceito de *Service Oriented Architecture* como arquitetura para sistemas distribuídos prova-se não só relevante, como uma aposta a considerar em sistemas modelares e de fácil integração. Devem, contudo, ser referidas algumas dificuldades nas configurações dos projetos que acabaram por atrasar as implementações das lógicas dos serviços. Julga-se, no entanto, que essas dificuldades se prenderam à falta de experiência com a framework WCF, e acredita-se que, no final do projeto, tenham sido completamente superadas.

Considerando o balanço entre planeamento, projeções, objetivos realizados e objetivos por alcançar, estima-se que 100% do trabalho a cumprir tenha sido concluído, e o projeto tenha sido terminado num ponto positivo.

## 5 Instruções para Utilização

Para correr a aplicação, o módulo *StoreServer* deve ser iniciado em primeiro lugar, disponibilizando o serviço e aplicação web da loja. De seguida, podem ser iniciadas múltiplas instâncias das aplicações *StoreApp* (StoreApp.exe) e/ou *Warehouse* (Warehouse.exe).

Como referência e manual de instruções, são explicados os variados elementos da interface gráfica para cada aplicação:

- **Aplicação Web:** Tentou-se, para esta aplicação, simular um ambiente real onde todo o processo de compra e consulta possa ser intuitivo para clientes externos. Deste modo, não existe nada a realçar para a sua utilização.
- **Aplicação Loja:** O ecrã superior da aplicação é utilizado para registo de novas compras ou encomendas. O Id do livro a ser comprado deve ser selecionado através de um clique sobre a respetiva entrada na tabela da direita.

id	title	price	quantity
1	Harry Potter	10	5181
2	50 Shades of Grey	10	10527
3	Lord of the Rings	10	1020

O ecrã inferior é utilizado para listagem e processamento de pedidos de reposição de stock ao armazém. Após processamento por funcionários do armazém, o seu estado muda (coluna *fulfilled* toma o valor de 1), podendo então serem adicionados ao stock atual da loja através de um duplo clique sobre as respetivas entradas na tabela.

id	book_id	quantity	fulfilled
87	3	500	0
89	3	500	0

- **Aplicação Armazém:** A tabela apresentada na aplicação do armazém é utilizada para listagem e processamento de pedidos de reposição de stock ao armazém. Pedidos ainda não processados (coluna *fulfilled* com valor atual de 0) podem ser completos com um duplo clique sobre as respetivas entradas na tabela, enquanto que pedidos já completos (coluna *fulfilled* com valor atual de 1) podem ser removidos da tabelas do mesmo modo (utilizado apenas organização gráfica).

## 6 Referências

- [1] Microsoft Developer Network: "Windows Communication Foundation", 2015 [Online]. Disponível em <https://msdn.microsoft.com/en-us/library/dd456779%28v=vs.110%29.aspx> (Junho, 2015)
- [2] Microsoft Developer Network: "Working with Sample Databases", 2015 [Online]. Disponível em <https://msdn.microsoft.com/en-us/library/dd264901.aspx> (Junho, 2015)
- [3] Microsoft Developer Network: "SmtpClient.Send Method (MailMessage)", 2015 [Online]. Disponível em <https://msdn.microsoft.com/en-us/library/swas0fwc%28v=vs.110%29.aspx> (Junho, 2015)
- [4] António Miguel Pontes Pimenta Monteiro: "Distribution and Integration Technologies", 2015 [Online]. Disponível em <http://paginas.fe.up.pt/apm/T-DIN/> (Junho, 2015)