

---

# SISTEMA DISTRIBUÍDO USANDO WCF

## LOJA DE LIVROS

---

27 de Julho de 2020

Beatriz Souto de Sá Baldaia, up201505633  
João Francisco Veríssimo Dias Esteves, up201505145  
Faculdade de Engenharia da Universidade do Porto

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Arquitetura</b>	<b>5</b>
2.1	Projetos da Solução . . . . .	6
<b>3</b>	<b>Funcionalidades</b>	<b>7</b>
3.1	Loja . . . . .	7
3.2	Armazém . . . . .	8
3.3	Impressora . . . . .	8
3.4	Website . . . . .	8
<b>4</b>	<b>Detalhes do código</b>	<b>8</b>
4.1	Diferentes bindings para diferentes fins . . . . .	8
4.2	<i>Callbacks</i> - Suporte de múltiplos terminais . . . . .	9
4.3	Queues . . . . .	10
4.4	WCF Client Proxy . . . . .	11
4.5	Transações da base de dados . . . . .	11
4.6	E-mails . . . . .	11
<b>5</b>	<b>Interface</b>	<b>12</b>
5.1	Loja . . . . .	12
5.2	Armazém . . . . .	15
5.3	Website . . . . .	15
5.4	Impressora . . . . .	17

## 1 INTRODUÇÃO

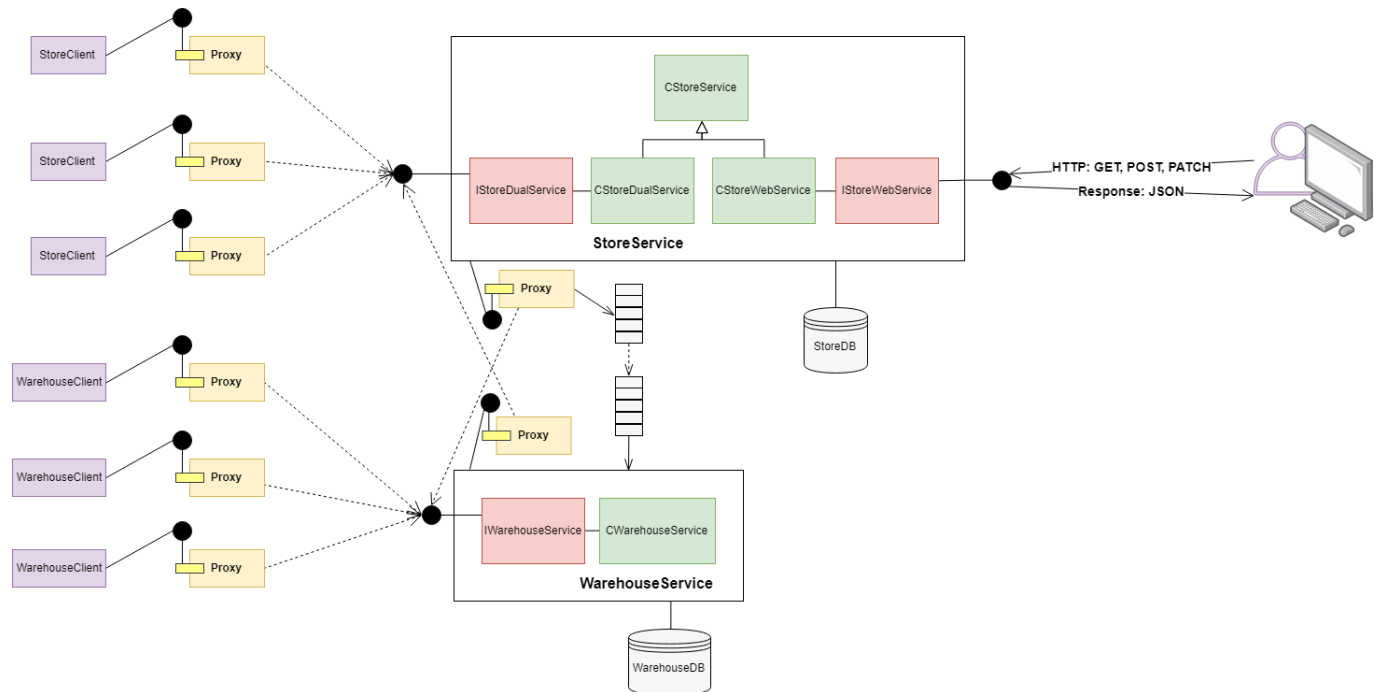
Uma Editora de livros pretende desenvolver um sistema de gestão de vendas e stock de livros. Esta possui duas áreas de trabalho distintas: a loja, onde os livros são expostos ao público, e o armazém, onde grandes quantidades de livros são guardadas. A Editora pretende ainda disponibilizar um website para que os clientes possam remotamente fazer compras e consultar o estado dos seus pedidos.

Para o projeto foi usado **Windows Communication Foundation (WCF)**. WCF é uma framework para construir aplicações orientadas a serviços. Com WCF é possível enviar mensagens assincronamente de um endpoint (o serviço) a outro (o cliente). De entre as funcionalidades do WCF podemos salientar:

- Orientado a serviços - *Loosely-coupled services* permitem que qualquer cliente criado, em qualquer plataforma, se conecte a qualquer serviço, desde que o contrato estabelecido se verifique.
- Múltiplos padrões de mensagens - O mais conhecido é o *request/reply*, onde um endpoint pede dados (cliente) e o outro responde ao pedido. Também existe o padrão *one-way* em que um único endpoint envia a mensagem sem esperar por uma resposta, e o *duplex exchange* em que dois endpoints estabelecem um conexão, sendo enviadas mensagens nos dois sentidos.
- Metadados do serviço - São partilhados os metadados do serviço que são usados para automaticamente gerar e configurar clientes para que estes possam aceder a serviços WCF.
- Contratos de dados - A forma mais fácil de processar e partilhar dados é criando classes que representam entidades. O WCF é capaz de, depois de criadas essas classes, gerar metadados que permitem que os clientes usem os tipos de dados que queremos partilhar.
- Múltiplos transportes e codificações - As mensagens podem ser enviadas em diversos tipos de protocolos (como HTTP, TCP, named pipes, MSMQ) e codificações (texto, formatos binários otimizados). Também há a opção de criar o próprio e personalizado protocolo de transporte e codificação.
- Filas de mensagens - Filas permitem a transferência confiável de mensagens. Aqui a comunicação por filas é construída sobre MSMQ.

- Mensagens persistentes - As mensagens nunca são perdidas em comunicações rompidas. As mensagens neste padrão são sempre guardadas na base de dados. Assim, se ocorrer um falha de comunicação, a base de dados permite concluir a troca de mensagens quando a conexão é restabelecida.
- AJAX e REST - O WCF pode ser configurado para processar puro XML que não está no formato SOAP. Suporta diferentes formatos XML e formatos que não XML, tais como JSON.

## 2 ARQUITETURA



**Figura 1:** Arquitetura orientada a serviços.

Serviço	StoreService			WarehouseService		
Interfaces	IStoreDualService	IStoreWebService	IOrdersChanged	IWarehouseQueueService	IWarehouseService	IRequestsChanged
Service Contract	ServiceContract(CallbackContract = typeof(IOrdersChanged))	ServiceContract	-	ServiceContract	ServiceContract(CallbackContract = typeof(IRequestsChanged))	-
Operation Contract	Subscribe, Unsubscribe, GetBooks, GetClients, GetClientOrders, GetBook, GetOrders, CreateOrder, CreateClient, NotifyFutureArrival, SatisfyOrders, GetRequests.	<b>GET:</b> GetBooks, GetClients, GetClientOrders, GetBook, GetOrders. <b>POST:</b> CreateOrder, CreateClient. <b>PATCH:</b> NotifyFutureArrival.	<b>One way:</b> OrderCreated, OrderStateUpdated, AddRequest, DeleteRequest	<b>One way:</b> RequestBooks.	Subscribe, Unsubscribe, GetRequests, SendBooks.	<b>One way:</b> RequestCreated, RequestStateUpdated
Data Contract	Order, Request			Request		
Base Addresses	<a href="http://localhost:8733/Design_Time_Addresses/StoreService/ws/">http://localhost:8733/Design_Time_Addresses/StoreService/ws/</a>	<a href="http://localhost:8733/Design_Time_Addresses/StoreService/web/">http://localhost:8733/Design_Time_Addresses/StoreService/web/</a>	<a href="http://localhost:8733/Design_Time_Addresses/StoreService/ws/">http://localhost:8733/Design_Time_Addresses/StoreService/ws/</a> <a href="http://localhost:8733/Design_Time_Addresses/StoreService/web/">http://localhost:8733/Design_Time_Addresses/StoreService/web/</a>	net.msmq://localhost/private/	<a href="http://localhost:8733/Design_Time_Addresses/WarehouseService/">http://localhost:8733/Design_Time_Addresses/WarehouseService/</a>	
Endpoint Addresses	-	-	mex	BookShopQueue	ws	mex
Binding	wsDualHttpBinding	webHttpBinding	mexHttpBinding	netMsmqBinding	wsDualHttpBinding	mexHttpBinding
Contract	IStoreDualService	IStoreWebService	IMetadataExchange	IWarehouseQueueService	IWarehouseService	IMetadataExchange

## 2.1 Projetos da Solução

*WCF Service Library:*

- StoreService - Representa o serviço de venda de livros, gestão de pedidos de clientes e geração de pedidos de reposição do stock.
- WarehouseService - Representa o serviço de receção de pedidos da loja e envio de livros para a mesma.

*Console applications:*

- Printer - Aplicação que apresenta a fatura de cada pagamento realizado.

*Windows GUI applications ("Windows Forms App"):*

- StoreClient - Aplicação usada pelos empregados da loja para realizarem vendas e pedidos ao armazem, notificar a receção de livros e aceder ao estado geral dos pedidos dos clientes e do stock dos livros.
- WarehouseClient - Aplicação usada pelos empregados do armazem acederem à lista de pedidos feitos pela loja física ao armazem e envio de livros para a loja.

*Web application:*

- BookShopWebsite - Aplicação usada remotamente pelos clientes para que estes possam realizar compras e aceder ao estado destas.

### **3 FUNCIONALIDADES**

Este sistema consiste em integrar as diferentes zonas da empresa - loja e armazém - no processo de venda de livros. Cada uma das zonas contém um servidor e suporta vários terminais idênticos entre si, para poder paralelizar o trabalho de diferentes empregados nas mesmas zonas. Para além dos terminais, existe também um website que comunica com o servidor da loja para que as pessoas possam encomendar livros.

Os pedidos dos livros são primeiro realizados ou pelo website ou pelo terminal da loja, selecionando-se a quantidade e cliente. Se houver stock na loja, a venda será realizada imediatamente, sendo o estado do pedido marcado de acordo, senão é feito um pedido ao armazém para que envie mais livros. No armazém um empregado pode ver num dos seus terminais os pedidos atuais e, quando for a enviar os livros respetivos, marcar os pedidos como enviados. O empregado da loja pode agora ver que os livros foram enviados e confirmar a sua receção, o que irá marcar os pedidos respetivos como "a chegar" para uma certa data, e confirmar a sua receção, o que irá concluir o processo de venda de todos os pedidos possíveis consoante o novo stock.

#### **3.1 Loja**

Seguem-se as funcionalidades aqui disponibilizadas ao utilizador:

- Criação de um pedido
- Criação de um cliente
- Consulta de stock por livro
- Consulta de pedidos dos clientes
- Consulta de encomendas de livros do armazém
- Confirmação da receção de encomendas de livros do armazém

## 3.2 Armazém

Seguem-se as funcionalidades aqui disponibilizadas ao utilizador:

- Marcação do envio de livros encomendadas pela loja
- Consulta das encomendas da loja

## 3.3 Impressora

Este terminal é uma simples consola que apresenta as faturas que são geradas com o pagamento realizado por cada livro. Uma fatura é constituída pelos detalhes do cliente, detalhes do livro comprado (título, quantidade e preço unitário) e preço total.

## 3.4 Website

Seguem-se as funcionalidades aqui disponibilizadas ao utilizador:

- Escolha do utilizador logado atualmente
- Criação de pedidos de livros para o utilizador atual
- Consulta dos pedidos realizados pelo utilizador atual
- Consulta dos livros disponibilizados pela loja
- Registo de novos utilizadores

# 4 DETALHES DO CÓDIGO

## 4.1 Diferentes bindings para diferentes fins

Um serviço WCF é definido, instalado e consumido por endpoints. Estes têm um *address* (localização do serviço), um *binding* (protocolo do serviço) e um *contract* (operações e estrutura de dados que o serviço disponibiliza).

No nosso projeto foi usado:

- *netMsmqBinding* para o serviço de envio de pedidos ao armazem usando filas para as mensagens.



- *wsDualHttpBinding* para os serviços usados pelas aplicações da loja e do armazem, estabelecendo um contrato duplex, permitindo que ambos serviço e cliente enviem e recebam mensagens.
- *webHttpBinding*, que configura endpoints expostos por pedidos HTTP em vez de mensagens SOAP. Para serviços que usam este *binding*, as suas operações devem ser anotadas com os atributos *WebGetAttribute* ou *WebInvokeAttribute*. Nós usamos o *webHttpBinding* para a aplicação web, o site da loja.
- *mexHttpBinding* que especifica as definições para o *binding* usado para a troca de mensagens WS-MetadataExchange (WS-MEX), por HTTP. Esta metadata é necessária para a construção do proxy no lado do cliente.

## 4.2 Callbacks - Suporte de múltiplos terminais

Poderá ser do interesse da empresa ter vários terminais tanto na loja como no armazém para que diferentes empregados executem as mesmas funções em paralelo. Para isto, os terminais terão de apresentar informação idêntica. Surge então o modelo de *callbacks*, no qual os servidores invocam funções nos terminais para que estes atualizem o seu estado. Isto é contrastado com o modelo típico de servidor/cliente em que é o terminal (o cliente) a fazer um pedido ao servidor e este lhe responde algo, enquanto que aqui a resposta tem de ir para todos os terminais e não só para um.

Neste modelo, os terminais subscrevem o servidor. Assim, este irá invocar certos *callbacks* para todos os seus clientes quando algum deles executar alguma ação. Por exemplo, quando um terminal da loja criar um pedido, este executa um método do servidor que, após o respetivo processamento, irá executar os *callbacks* de todos os terminais subscritos para que estes adicionem o novo pedido às suas listas de pedidos.

Para suportar esta comunicação *duplex* são necessários certos requerimentos:

- Cada *endpoint* dos serviços da loja e do armazém (exceto o do serviço REST, para o website) usa o binding *wsDualHttpBinding*
- Os contratos das respetivas interfaces definem os seus *callback contracts*, que são interfaces que contêm os *callbacks* dos clientes
- Cada *callback* tem o atributo *IsOneWay* na respetiva interface, pois o *callback* executa apenas no sentido servidor->cliente

- As classes que implementam as interfaces dos serviços têm o atributo *InstanceContextMode* definido como *PerCall*
- As classes dos terminais implementam uma interface de *callback* que contém os *callbacks* que cada terminal deve implementar
- As classes dos terminais têm o atributo *[CallbackBehavior(UseSynchronizationContext = false)]*

Este último é necessário para que os *callbacks* possam ser executados numa thread diferente da que iniciou o pedido ao servidor. Sem este atributo, o terminal entraria em deadlock: estaria para sempre à espera que o servidor lhe respondesse, mas o servidor não conseguiria terminar a conexão porque não conseguia executar o *callback*.

### 4.3 Queues

O uso de filas permite uma transmissão confiável de mensagens, persistência das mesmas, tanto na origem como no destino do envio, o envio de mensagens mesmo quando o destinatário não está disponível/ligado e a inclusão de envios e receções em transações. Em WCF *bindings* encapsulam detalhes da troca de mensagens. *Queuing* é tratado como qualquer outro *transport binding* em WCF.

Seguem-se requisitos para o uso de *queued binding* em WCF:

- Todas as operações do serviço têm de ser num só sentido porque, por defeito, WCF não suporta comunicações *duplex* com o uso de filas.
- Para gerar um cliente WCF usando os metadatos é preciso mais um endpoint HTTP no serviço.
- A fila MSMQ a usar já deve existir no sistema.

*netMsmqBinding* permite que dois endpoints comuniquem usando MSMQ de uma forma transparente. No nosso projeto o WarehouseService foi configurado de forma a fornecer um serviço com um endpoint que usa este *binding*. Tal serviço é usado pelo StoreService para que possa enviar pedidos de reposição de stock ao armazem. A fila aqui é necessária porque, ao contrário do servidor da loja, o servidor do armazem não está sempre ativo. Desta forma, mesmo que os pedidos da loja sejam enviados quando o servidor do armazem não está ligado, o armazem poderá mais tarde, ao ligar-se novamente, ler a mensagem deixada na fila.

## 4.4 WCF Client Proxy

A aplicação de um cliente usa um proxy para comunicar com o serviço. Tipicamente, a aplicação do cliente importa os metadados do serviço para gerar o código que pode ser usado para invocar as operações do serviço.

No nosso trabalho bastou-nos recorrer à ferramenta *Add Service Reference* do Visual Studio para que a classe cliente ficasse com uma referência do serviço. Desta forma também é gerado automaticamente o ficheiro de configuração do proxy.

## 4.5 Transações da base de dados

Há certas operações à base de dados da loja que são executadas sequencialmente. Tais operações apenas fazem sentido se forem todas executadas com sucesso. Então, estas são incluídas em transações para que, caso uma falhe, todas as operações da transação respetiva são revertidas.

Por exemplo, a criação de um pedido na loja requer a inserção do tal na base de dados e que o stock do livro respetivo tenha o seu valor atualizado. Não faria sentido que o pedido fosse criado mas que o stock, devido a uma falha, se mantivesse igual, então estas 2 operações são incluídas numa transação para que não hajam inconsistências na base de dados.

## 4.6 E-mails

Quando o pedido é criado e quando o pedido é enviado são enviados e-mails ao cliente a informar do seu pedido. Estes e-mails são enviados via SMTP pelo Gmail, com um corpo em HTML.

## 5 INTERFACE

### 5.1 Loja

1. Realizar uma compra (para um cliente já registado ou para um novo cliente).
2. Aceder à secção de livros onde se encontram os livros da loja e os respetiva quantidade.
3. Aceder à lista de pedidos do cliente onde está disponível, para cada pedido já feito, o identificador, título do livro, a quantidade e estado do pedido.
4. Aceder à secção de notificações do armazem (lista de pedidos feitos ao armazem que já foram satisfeitos pelo mesmo).
5. Sinalizar a receção de livros provenientes do armazem.

The screenshot shows a window titled "Store Terminal" with standard Windows window controls (minimize, maximize, close). Inside the window, there are four orange buttons at the top: "New Order", "Stock", "Orders", and "Warehouse Requests". Below these, the "Book" label is followed by a dropdown menu showing "O Diário de um Banana - Vai Tudo Alt". The "Quantity" label is followed by a text input field containing the number "1". Below these fields are two buttons: "Registered Client" and "New Client". The "Name" label is followed by a dropdown menu showing "Joao Esteves". In the bottom right corner, the text "Total Price 15,29" is displayed, and below it is a "Pay" button.

**Figura 2:** Criação de uma nova compra para um cliente já existente

Store Terminal

New Order Stock Orders Warehouse Requests

Book O Diário de um Banana - Vai Tudo Abaixo

Quantity 1

Registered Client New Client

Name Vitor Magalhaes

Address Paranhos, Porto

Email vitor.m@gmail.com

Total Price 15,29

Pay

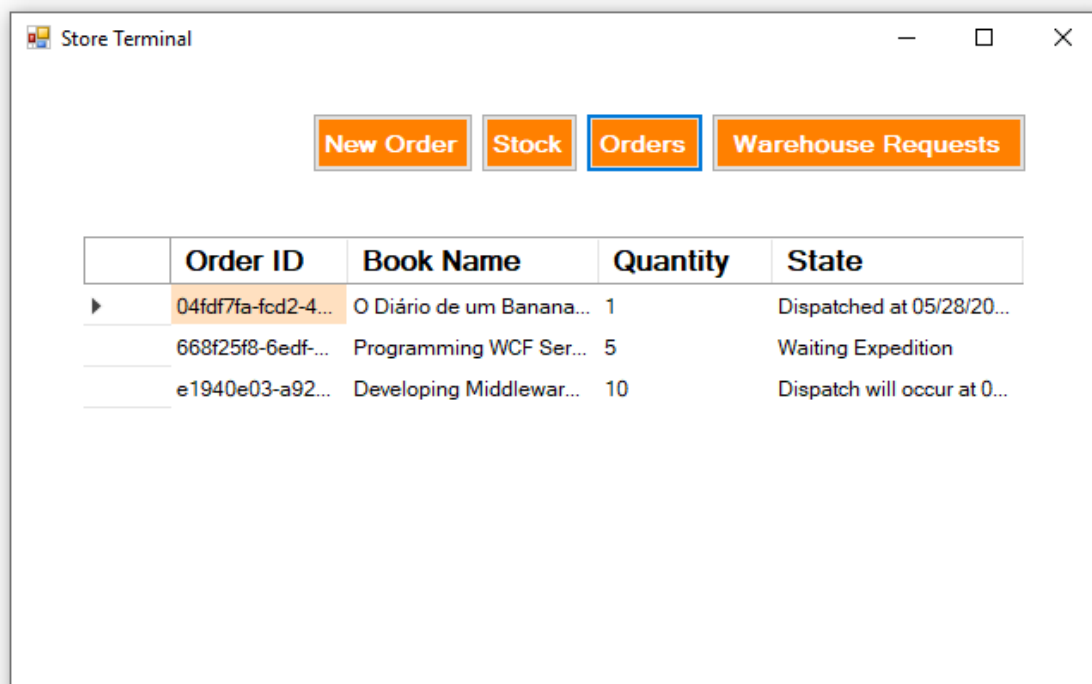
**Figura 3:** Criação de uma nova compra para um cliente novo

Store Terminal

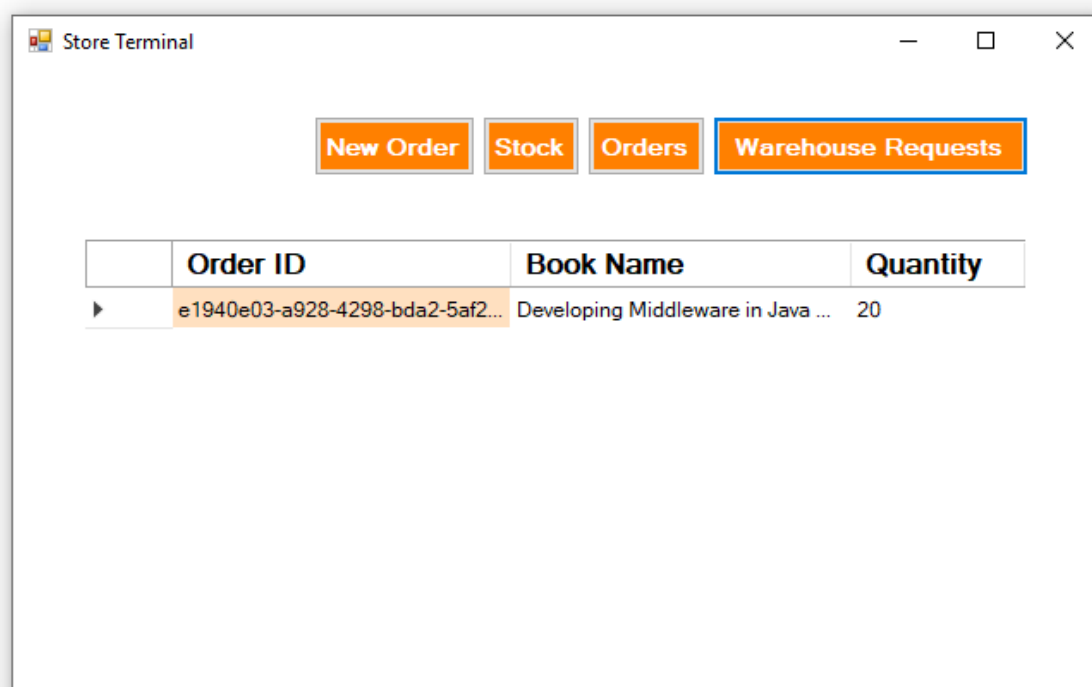
New Order Stock Orders Warehouse Requests

	Book Name	Stock
	O Diário de um Banana - Vai Tudo Abaixo	8
	O Diário de um Banana - Um Romance com Cartoons	20
▶	Advanced .NET Remoting, 2nd Edition	10
	Programming WCF Services, 4th Edition	3
	Developing Middleware in Java EE 8	7
	Activiti in Action: Executable business processes in BPMN 2.0	10
	O Diário de um Banana - O Rodrick é Terrível	30
	O Diário de um Banana - Um Dia de Cão	10
	O Diário de um Banana - A Verdade Nua e Crua	5

**Figura 4:** Stock disponível na loja



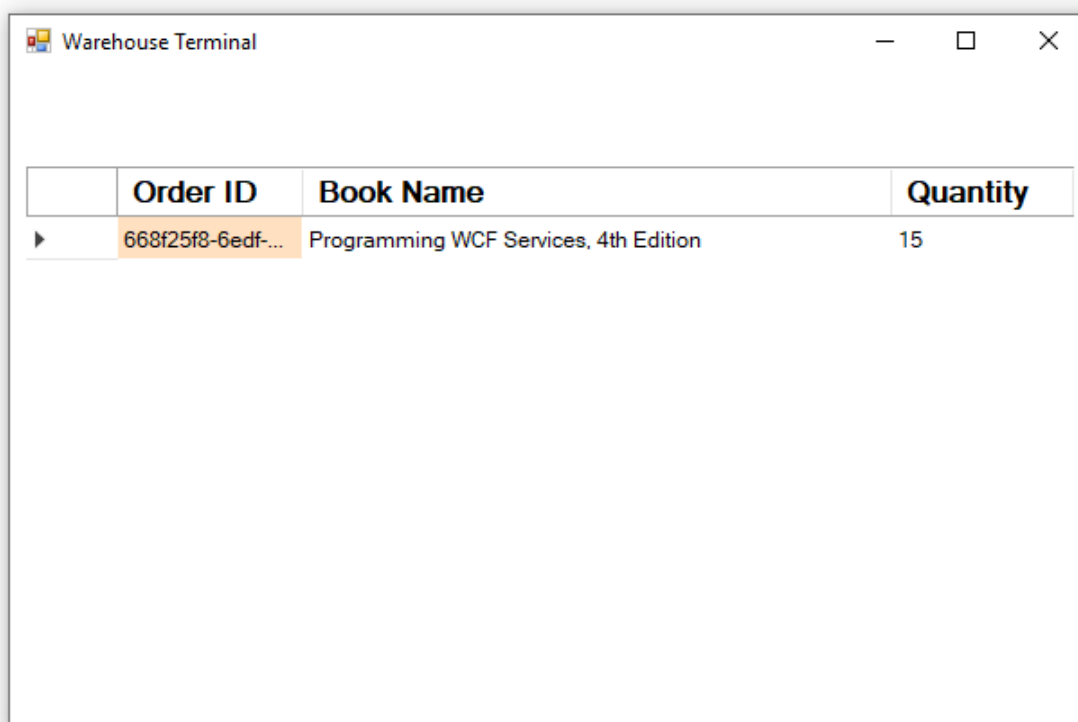
**Figura 5:** Pedidos dos clientes nos diversos estados



**Figura 6:** Encomendas feitas ao armazém

## 5.2 Armazém

1. Aceder à lista de encomendas da loja.
2. Selecionar as encomendas que já foram satisfeitos pelo armazém (caso em que o armazem está pronto para enviar livros para a loja).



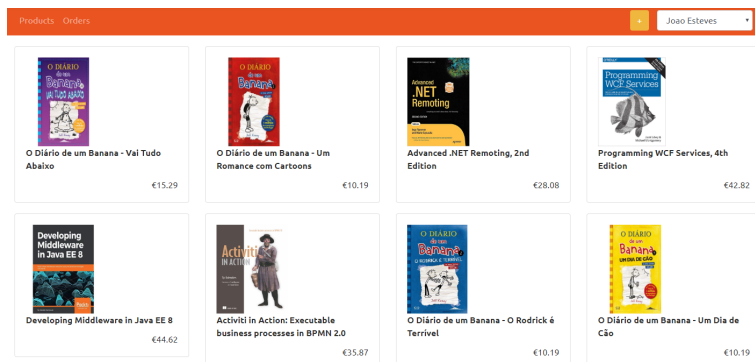
The image shows a window titled "Warehouse Terminal" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there is a table with the following structure:

	Order ID	Book Name	Quantity
▶	668f25f8-6edf-...	Programming WCF Services, 4th Edition	15

**Figura 7:** Terminal do armazém com 1 encomenda por satisfazer

## 5.3 Website

1. Aceder à lista de livros à venda
2. Mudar de utilizador
3. Registar novo utilizador
4. Realizar compra de livros
5. Aceder ao estados dos pedidos de um utilizador



**Figura 8:** Página com a lista de livros que estão à venda



**Figura 9:** Página para efetuar a compra de um livro, especificando-se a quantidade desejada

Products Orders				
Joao Esteves				
GUID	Quantity	Title	Price	State
8ca8bfee-a9e7-438c-ad26-0c640c830ddd	1	O Diário de um Banana - Vai Tudo Abaixo	15.29	Dispatch will occur at May 28, 2019

**Figura 10:** Página com a lista de pedidos/compras já realizados(as) por um cliente.

Products Orders
Joao Esteves

Name

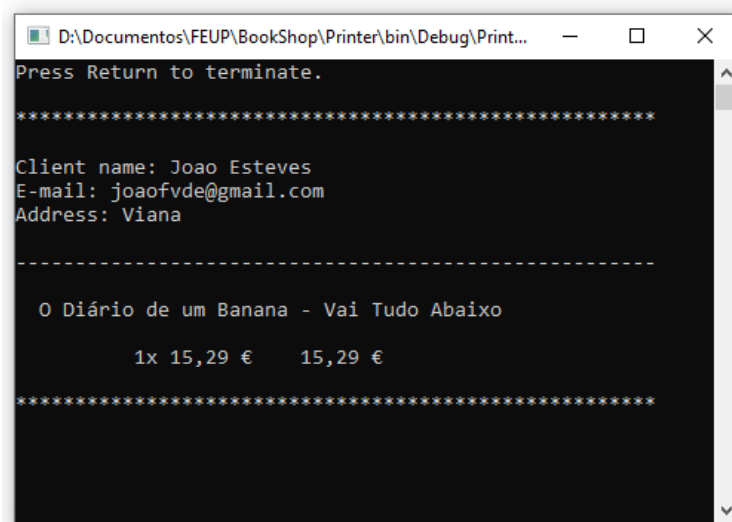
E-mail

Address

**Figura 11:** Página para efetuar o registo de um novo utilizador



## 5.4 Impressora



```
D:\Documentos\FEUP\BookShop\Printer\bin\Debug\Print...
Press Return to terminate.

*****

Client name: Joao Esteves
E-mail: joaofvde@gmail.com
Address: Viana

-----

O Diário de um Banana - Vai Tudo Abaixo

1x 15,29 €    15,29 €

*****
```

**Figura 12:** "Impressora" simulada por uma consola, com uma factura de uma compra realizada por um cliente.