



Universidade do Porto
Faculdade de Engenharia
FEUP

Tecnologias de Distribuição e Integração

Trabalho prático #1 – Mercado de *Diginotes*

Francisco Miguel Amaro Maciel - 201100692

Hugo Miguel Ribeiro de Sousa – 201100690

Ricardo Daniel Soares da Silva - 201108043

18 de Abril de 2015

Índice

Enunciado	3
Arquitetura	4
Interfaces.....	6
Funcionalidades e Testes Executados	9
Cenário de utilização	11
Conclusão.....	18

Enunciado

Este projeto foi elaborado no âmbito da unidade curricular *Tecnologias de Distribuição e Integração* do 4º ano do Mestrado Integrado de Engenharia Informática e Computação e consiste no desenvolvimento de uma aplicação de compra e venda de itens, designados *diginotes*.

O sistema tem uma cotação atual, que corresponde ao preço a que os *diginotes* são vendidos nesse momento. A cotação inicial é de 1€. Um utilizador registado no sistema pode emitir ordens de compra ou venda dos seus *diginotes*. As ordens são satisfeitas pela ordem temporal em que foram emitidas, a partir da mais antiga. Assume-se que um utilizador não tem um limite monetário, pelo que pode comprar a quantidade que pretender. Se a oferta for superior à procura, ou vice-versa, a cotação pode ser ajustada pelos detentores das ordens pendentes (baixar caso a oferta seja superior, e aumentar caso a procura seja superior), de forma a atrair novos compradores ou vendedores.

Caso a oferta contrária não seja suficiente, as ordens de compra ou venda devem ser satisfeitas parcialmente, ficando a restante quantidade pendente. Os detentores destas ordens pendentes podem, posteriormente, alterar a cotação do sistema. Se houverem mais utilizadores com ordens semelhantes pendentes, então deverá haver 1 minuto de espera em que as ordens não são concretizadas, em que os estes deverão aceitar a nova cotação ou remover a sua ordem.

Arquitetura

O projeto desenvolvido baseia-se numa arquitetura cliente-servidor, tendo sido utilizado o *.NET Remoting*, que permite facilmente a criação de aplicações distribuídas. Para armazenar a informação, foi criada uma base de dados *sqlite*, permitindo a consistência e manutenção dos dados.

A solução é composta por 4 projetos:

- Server – servidor, que aloja o *DiginoteManager*
- Client_UI – cliente, com visualização gráfica
- Manager – projeto que contém a definição do objeto remoto - *DiginoteManager*
- Shared – definições partilhadas pelo cliente e pelo servidor

O servidor alberga um objeto remoto, que se designa *DiginoteManager*. Aqui é centralizada a informação do sistema, são feitos os acessos à base de dados e são guardados os eventos aos quais os clientes se podem registar. Existem 3 eventos distintos, que representam as principais informações que os clientes devem automaticamente modificar e notificar. Estes eventos são *broadcasted* para todos os clientes subscritos, contendo a identificação dos clientes associados ao evento. Este método tem a vantagem de poder ser extendido de forma a notificar toda a comunidade de determinado evento, mas tem como desvantagem um *overhead* de mensagens possivelmente desnecessárias para determinados clientes. Os eventos são os seguintes:

- Notificação de venda – tem como parâmetro o comprador, o vendedor e a quantidade, disparado quando um utilizador inicia uma ordem de compra.
- Notificação de compra – tem como parâmetro o comprador, o vendedor e a quantidade, disparado quando um utilizador inicia uma ordem de venda.
- Alteração de cotação – tem como parâmetro a cotação anterior e a nova cotação, disparado quando a cotação é alterada.

Os clientes registam-se aos eventos através de *repeater handlers*, evitando assim a inclusão do cliente no lado do servidor e seguindo as boas práticas do *.NET Remoting*.

Segue-se um exemplo da interação de 2 clientes com o servidor, mostrando o *broadcast* dos 3 tipos de eventos para 2 clientes subscritos. No primeiro caso, o C1 emite uma ordem de compra de 5 *diginotes*, que é parcialmente satisfeita. C1 altera posteriormente a cotação para um valor mais elevado. Por último, C2 emite uma ordem de venda que é totalmente satisfeita.

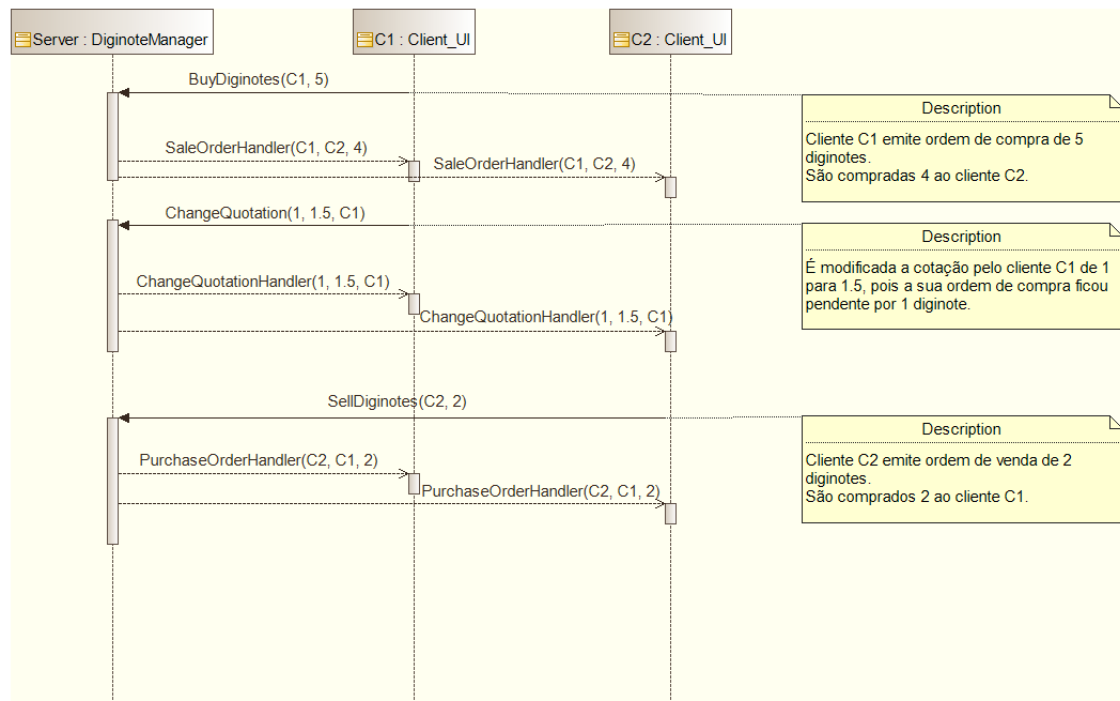
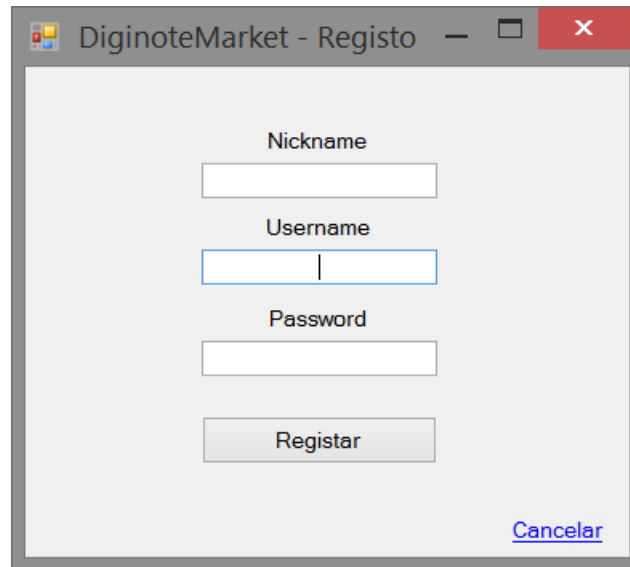


Fig. 1 – exemplo de interação de 2 clientes e servidor

No módulo *Shared*, são apenas identificados os *delegates*, as definições dos argumentos e dos *repeaters* dos eventos, que têm de ser partilhados com o cliente e o servidor.

Interfaces

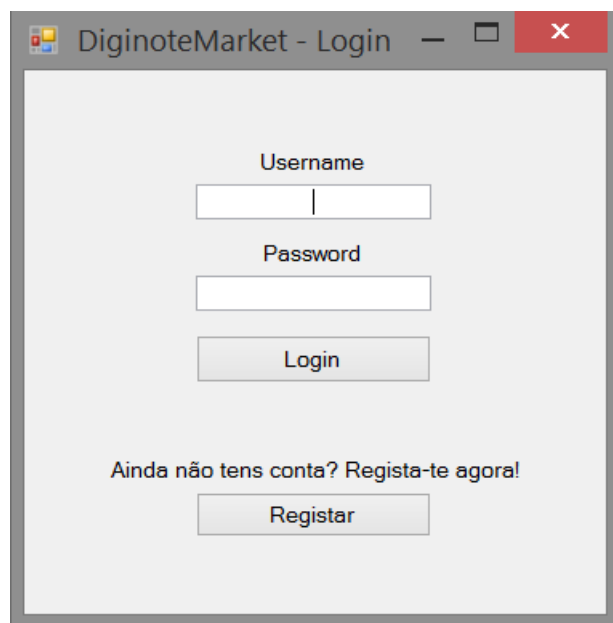
A aplicação apresenta uma interface gráfica simples e intuitiva, sendo composta por 3 *forms*. Na janela de registo, é possível registar uma nova conta, para posteriormente entrar no sistema. Uma conta necessita de um *nickname*, *username* e *password*, sendo que os dois primeiros necessitam de ser únicos no sistema. Clicando em “Cancelar”, o utilizador é remetido para a janela de Login novamente.



The image shows a software window titled "DiginoteMarket - Registo". It features a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is light gray and contains three text input fields stacked vertically, labeled "Nickname", "Username", and "Password". Below the "Password" field is a button labeled "Registar". In the bottom right corner of the window, there is a blue underlined text link labeled "Cancelar".

Fig. 1 – Janela de registo

Na janela de Login é possível introduzir as credenciais de acesso ao sistema. É a partir daqui, que se pode também aceder ao Registo.



The image shows a window titled "DiginoteMarket - Login". Inside the window, there are two text input fields. The first field is labeled "Username" and the second is labeled "Password". Below the "Password" field is a button labeled "Login". At the bottom of the window, there is a link that says "Ainda não tens conta? Regista-te agora!" and a button labeled "Registar". The window has a standard macOS-style title bar with a red close button on the right.

Fig. 2 – Janela de login

Na barra superior da janela principal, podemos ver o *nickname* do utilizador que se encontra no sistema. É possível verificar em real-time a quantidade de *diginotes* que o utilizador possui e a cotação atual do mercado. Clicando no botão do lado esquerdo de *My Diginotes*, é possível expandir ou retrair a visualização dos números de série das *diginotes* do utilizador, o que pode ser útil para validação visual das transações. Ainda do lado esquerdo da janela, é possível adicionar ou modificar ordens de compra ou venda, alterando os valores numéricos associados. Abaixo, são registadas as notificações do utilizador. Do lado direito da janela, são registadas as ordens de compra e venda pendentes, onde é indicada a data da ordem e a quantidade pendente. Aqui, é ainda possível alterar a cotação a qualquer momento.

No exemplo demonstrado, o utilizador *abc* tem 2 *diginotes* na sua carteira pessoal, tendo uma ordem de compra de 1 unidade pendente. A cotação atual é de 1.00€.

The screenshot displays the 'DiginoteMarket @ abc' application window. The interface is organized into several sections:

- Top Bar:** Displays the user's nickname 'abc'.
- My Diginotes:** Shows the user's current balance of 2 diginotes. A minus button is located to the left.
- Cotação:** Displays the current market quote of 1,00.
- Order Management:**
 - Ordem de Compra:** Shows a pending buy order for 1 unit, dated 16/04/2015 at 13:56:10. The current price is 1.00. A button labeled 'Alterar/Manter Cotação' is present.
 - Ordem de Venda:** Shows a pending sell order for 0 units.
- Notifications:** A section labeled 'Notificações' with a button 'Adicionar Ordem de Compra' and a button 'Adicionar Ordem de Venda'.
- Serial Numbers:** A list of diginote serial numbers, with '12345678' highlighted in blue and '123456789' listed below it.

Fig. 3 - MainForm

Funcionalidades e Testes Executados

Todas as funcionalidades básicas inicialmente pensadas foram implementadas. Segue-se uma lista com a descrição destas e respetivo método de teste da funcionalidade.

Descrição da Funcionalidade	Teste
1. Registo	Registar uma conta com um <i>nickname</i> e <i>username</i> únicos, e uma <i>password</i> , na janela de registo.
2. Login	Entrar no sistema com uma conta previamente registada, usando o <i>username</i> e a <i>password</i> nos campos respetivos.
3. Log	Os eventos mais relevantes são gravados num ficheiro de texto "log.txt", do lado do servidor.
4. Persistência dos dados	Os dados são gravados numa base de dados sql. Ao fazer login com uma conta, verifica-se que contém toda a informação que continha antes de sair da conta.
5. Visualizar números de série das <i>diginotes</i>	Clicando no botão "+" da <i>MainForm</i> , surge uma lista das <i>diginotes</i> que o cliente possui. Para fechar esta visualização deve clicar-se no mesmo botão usado para abrir, que entretanto se modifica para "-".
6. Visualizar cotação atual e quantidade de <i>diginotes</i>	Após o login, é automaticamente mostrado o número de <i>diginotes</i> do cliente e a cotação atual, obtido através da base de dados. Quando existe uma atualização destes valores, são também automaticamente atualizados na <i>MainForm</i> .
7. Notificar cliente	Os clientes são notificados acerca das suas transações, ou seja, <i>diginotes</i> que sejam compradas ou vendidas por si, assim como pelas alterações de cotação.
8. Emitir ordem de compra	Para emitir uma ordem de compra, deve seleccionar-se uma quantidade no campo respetivo e clicar no botão "Adicionar ordem de compra". Caso a ordem seja totalmente satisfeita, será notificado no campo das notificações abaixo. Caso seja apenas parcialmente satisfeito, o cliente é notificado da quantidade que foi comprada, e do lado direito, na seção "Ordem de compra", surgirá a quantidade que fica pendente.

9. *Update* de ordem de compra

Para fazer *update* a uma ordem de compra, deve seguir-se os mesmos passos de emissão da mesma, ficando esta com a data do último *update*. Por exemplo, para remover uma ordem de compra, deve clicar-se em “Adicionar ordem de compra” com a quantidade de “0”. A qualquer momento, pode também aumentar-se a cotação, através de uma ordem de compra pendente.

10. Aceitação de nova cotação ou remoção de ordem por clientes com ordens de compra pendentes

Quando um cliente com uma ordem de compra pendente aumenta a cotação, as compras são interrompidas durante 1 minuto, onde os restantes clientes com ordens de compra devem aceitar a nova cotação ou remover a sua ordem de compra. Na *MainForm*, surgirá um contador decrescente e um botão “Remover ordem de compra”. Para aceitar a nova cotação, deve clicar em “Alterar/Manter Cotação”, deixando o novo valor sugerido, ou um valor superior. No caso se ser superior, o mesmo processo começará de novo para os restantes clientes com ordens de venda pendentes, começando o contador de novo. Se o tempo limite chegar ao fim sem nenhuma decisão por parte do utilizador, assume-se que a ordem se mantém.

11. Emitir ordem de venda

Para emitir uma ordem de venda, deve seleccionar-se uma quantidade no campo respetivo e clicar no botão “Adicionar ordem de venda”. Caso a ordem seja totalmente satisfeita, será notificado no campo das notificações abaixo. Caso seja apenas parcialmente satisfeito, o cliente é notificado da quantidade que foi vendida, e do lado direito, na seção “Ordem de venda”, surgirá a quantidade que fica pendente.

12. *Update* de ordem de venda

Para fazer *update* a uma ordem de venda, deve seguir-se os mesmos passos de emissão da mesma, ficando esta com a data do último *update*. Por exemplo, para remover esta ordem de venda, deve clicar-se em “Adicionar ordem de venda” com a quantidade de “0”. A qualquer momento, pode também diminuir-se a cotação, através de uma ordem de venda pendente.

13. Aceitação de nova cotação ou remoção de ordem por clientes com ordens de venda pendentes

Quando um cliente com uma ordem de venda pendente diminui a cotação, as vendas são interrompidas durante 1 minuto, onde os restantes clientes com ordens de venda devem aceitar a nova cotação ou remover a sua ordem de venda. Na *MainForm*, surgirá um contador decrescente e um botão “Remover ordem de venda”. Para aceitar a nova cotação, deve clicar em “Alterar/Manter Cotação”, deixando o novo valor sugerido, ou um valor inferior. No caso se ser inferior, o mesmo processo começará de novo para os restantes clientes com ordens de venda pendentes, começando o contador de novo. Se o tempo limite chegar ao fim sem nenhuma decisão por parte do utilizador, assume-se que a ordem se mantém.

Cenário de utilização

Demonstraremos aqui um cenário de utilização da aplicação, com recurso a 2 clientes. Considere-se 2 clientes registados no sistema, de *nicknames* *abc* e *abcd*, com as seguintes configurações iniciais. Não existem ordens de compra ou venda pendentes no sistema, por outros utilizadores.

The screenshot shows the 'DiginoteMarket @ abc' application window. The interface is divided into several sections:

- My Diginotes:** A box containing the number '2'.
- Cotação:** A box containing the value '1,00'.
- Order Entry:** Two columns, each with a numeric input field (both set to '0') and a button labeled 'Adicionar Ordem de Compra' and 'Adicionar Ordem de Venda' respectively.
- Ordem de Compra:** A box containing the value '0'.
- Ordem de Venda:** A box containing the value '0'.
- Notificações:** A large empty rectangular box.
- Footer:** A box containing the numbers '12345678' and '123456789' stacked vertically.

Fig. 4 – Configuração inicial do utilizador *abc*.

DiginoteMarket @ abcd

-

My Diginotes

2

Cotação

1,00

Ordem de Compra

0

0

Adicionar Ordem de Compra

0

Adicionar Ordem de Venda

Ordem de Venda

0

Notificações

123456

1234567

Fig. 5 – Configuração inicial do utilizador *abcd*.

Ao adicionar uma ordem de venda de 1 *diginote* no utilizador *abcd*, esta fica pendente, pois não existem ordens de compra pendentes.

The screenshot displays the 'DiginoteMarket @ abcd' application window. The interface is divided into several sections:

- My Diginotes:** A box showing the user's current balance of 2 diginotes.
- Cotação:** A box showing the current market rate of 1,00.
- Ordem de Compra:** A box for adding a buy order, currently showing a quantity of 0.
- Ordem de Venda:** A box for adding a sell order. It shows the date and time '16/04/2015 14:18:53', a quantity of 1, and a price of 1.00. A button labeled 'Alterar/Manter Cotação' is located below the price.
- Notificações:** A box for displaying notifications.
- Order History:** A large box at the bottom containing a list of orders with IDs 123456 and 1234567.

The 'Adicionar Ordem de Venda' button is highlighted with a blue border, indicating it is the active action being performed.

Fig. 6 – Emissão de ordem de venda pelo utilizador *abcd*.

O utilizador *abcd* pode então diminuir a cotação atual. A cotação é atualizada automaticamente em ambos os clientes, sendo também notificados desta alteração.

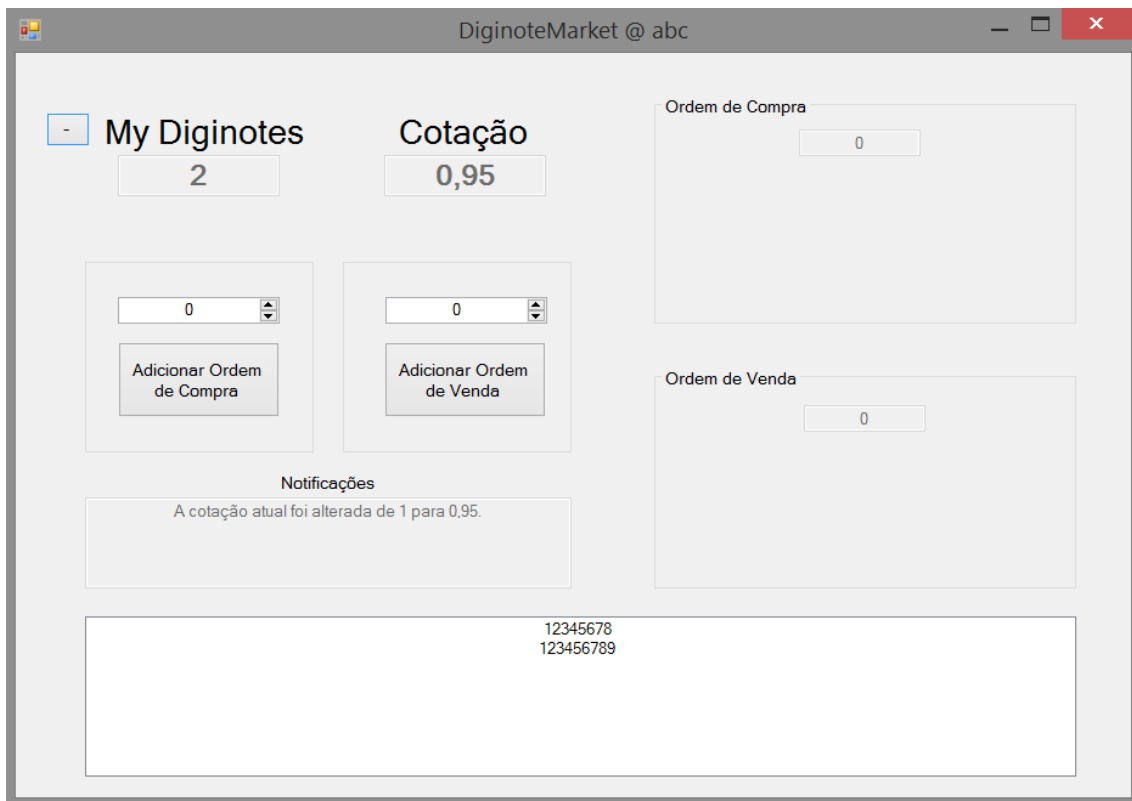


Fig. 7 – alteração da cotação na interface do utilizador *abc*.



Fig. 8 – alteração da cotação na interface do utilizador *abcd*.

O utilizador *abc*, observando a diminuição da cotação, pretende comprar 1 *diginote*, emitindo uma ordem de compra. Ambos os utilizadores são notificados da venda bem sucedida. Pode verificar-se que a *diginote* de serial 1234567 foi transferida de *abcd* para *abc*.



Fig. 9 – interface do utilizador *abc*, após emissão de ordem de compra.



Fig. 10 – interface do utilizador *abcd*, após venda de 1 *diginote*.

Ambos os utilizador adicionaram, posteriormente, uma ordem de compra de 1 unidade, tendo ficando ambas pendentes. O utilizador *abcd* decide então subir a cotação para 1.10€. O utilizador *abc* é então notificado da alteração, tendo 60 segundos para decidir manter/subir a cotação ou remover a sua ordem de compra.



Fig. 11 – Utilizador abc tem 57 segundos restantes para confirmar se pretende manter ou remover a ordem de compra.



Fig. 12 – A cotação é alterada na interface do utilizador abcd.

Um utilizador pode, a qualquer momento, modificar a sua ordem de venda ou compra, executando o mesmo processo de emissão da mesma. A ordem já emitida é então atualizada para a nova quantidade, assim como a sua data. Pode verificar-se nas seguintes imagens, que o utilizador *abc* atualizou a sua ordem de compra pendente de 2 unidade para 5 unidades, tendo sido também a data atualizada.



Fig. 13 – Utilizador abc tem uma ordem de compra pendente de 2 diginotes.



Fig. 14 – Ordem de compra é atualizada para 5 diginotes, clicando em “Adicionar Ordem de Compra” com o respetivo campo numérico a “5”. A data da ordem é atualizada.

Conclusão

Os objetivos deste trabalho prático foram cumpridos com sucesso. A tecnologia *.NET Remoting* pareceu-nos interessante e relativamente simples de usar, apesar de por vezes, o código parecer relativamente complexo e confuso para a real complexidade do problema. Atualmente, esta tecnologia encontra-se num estado de *legacy*, tendo sido substituída pelo *WCF (Windows Communication Foundation)*, também abordado nas aulas da unidade curricular, que engloba esta e muitas outras funcionalidades.