



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Diginote Exchange System

Mestrado Integrado em Engenharia Informática e Computação

Tecnologias Distribuídas

Luís Filipe Rodrigues Coelho - up201304273

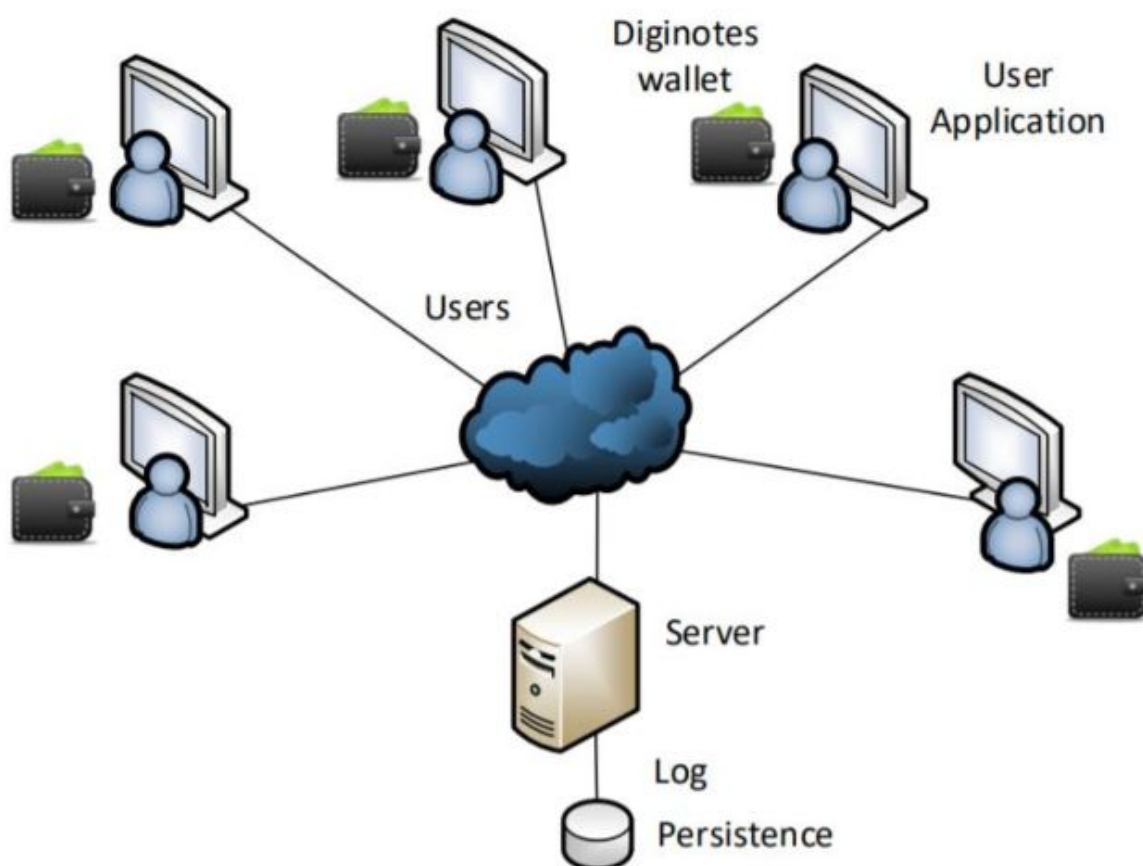
Tiago Rafael Ferreira da Silva - up201402841

Introdução	3
Arquitetura do Sistema	3
Cliente	4
Servidor	5
Base de dados	5
Funcionalidades e Testes	5
Login e Register	6
Tab Overview	8
Tab Selling Orders e Tab Purchase Orders	9
Observações	12

Introdução

Arquitetura do Sistema

A arquitetura do sistema está representada na figura seguinte. O sistema é composto por um servidor e diversos clientes. Cada aplicação cliente, possui uma interface intuitiva para que o seu utilizador possa executar facilmente todas as suas funcionalidade. O servidor por sua vez, gere e persiste a informação sobre cada transação no sistema.



De forma a implementar este sistema utilizando as melhores práticas de programação criou-se três módulos principais: Cliente, Servidor e Comum. O módulo Comum possui toda a informação comum aos outros dois módulos, nomeadamente as estruturas necessárias para comunicação entre os dois, que representam um objeto formado a partir da informação guardada na base de dados e outras estruturas necessárias para a utilização da tecnologia Remoting, nomeadamente a interface que o servidor dispõe para que os clientes possam executar as funcionalidades do sistema e a interface do cliente, para que este possa subscrever-se para receber notificações do sistema i.e,

atualizações do estado do sistema. As outros dois módulos serão descritos nas próximas secções.

Cliente

O módulo Cliente é constituído por três sub-módulos: *Interface*, *Cliente*, *Notificações*. Na implementação da interface utilizou-se a classe *Form* do C#. De forma a tentar implementar uma interface apelativa e inovadora utilizou-se também o *Bunifu* que permite a criação de campos de input personalizados. A interface possui dois *Form's* : o primeiro trata-se da janela de login e sign up; o segundo trata-se da aplicação principal onde o utilizador poderá, intuitivamente, tomar partido das funcionalidades do sistema. No sub-módulo do Cliente, estão implementadas todas as chamadas ao servidor. Cada vez que o utilizador interage com a interface, esta recolhe a informação inserida pelo utilizador e comunica-a com o módulo Cliente. Este, por sua vez, tem a responsabilidade de comunicar a ação pretendida pelo utilizador com o servidor e de atualizar a interface consoante a resposta retornada pelo mesmo. Por fim, o sub-módulo das Notificações permite que o servidor tome a iniciativa de atualizar a informação na aplicação cliente. Este processo é fundamental nesta aplicação pois garante que o utilizador não sinta dúvidas sobre se determinada informação está correta e evita que a aplicação cliente esteja periodicamente a perguntar ao servidor se houve alguma mudança de estado.

Servidor

O módulo Servidor também o é dividido em sub-módulos: a base de dados (que será descrita na próxima secção) e o servidor. O sub-módulo servidor foi desenvolvido para ser stateless, isto é, este serviço não guarda qualquer informação sobre o estado do sistema, apenas dá resposta aos pedidos da aplicação cliente. Para tal, tira partido da informação persistida na da base dados e juntamente com o input proveniente do utilizador consegue exercer todas as funcionalidades do sistema. Apesar do servidor ter sido desenvolvido para ser stateless tal não foi implementado com sucesso, uma vez que o sistema de notificações oferecido pelo servidor obrigue-o a guardar informação sobre que subscreveu esta funcionalidade.

Base de dados

Para persistir a informação do sistema, criou-se uma base de dados composta por 6 tabelas:

1. *User* - Cada entrada desta tabela possui toda a informação sobre um utilizador;
2. *Diginotes* - Possui a informação sobre cada moeda e o utilizador que à possui
3. *Quote* - Guarda o valor que uma moeda possui
4. *SellingOrders* - Guarda a informação sobre as ordens de venda (utilizador, quantidade e se está suspensa ou não)

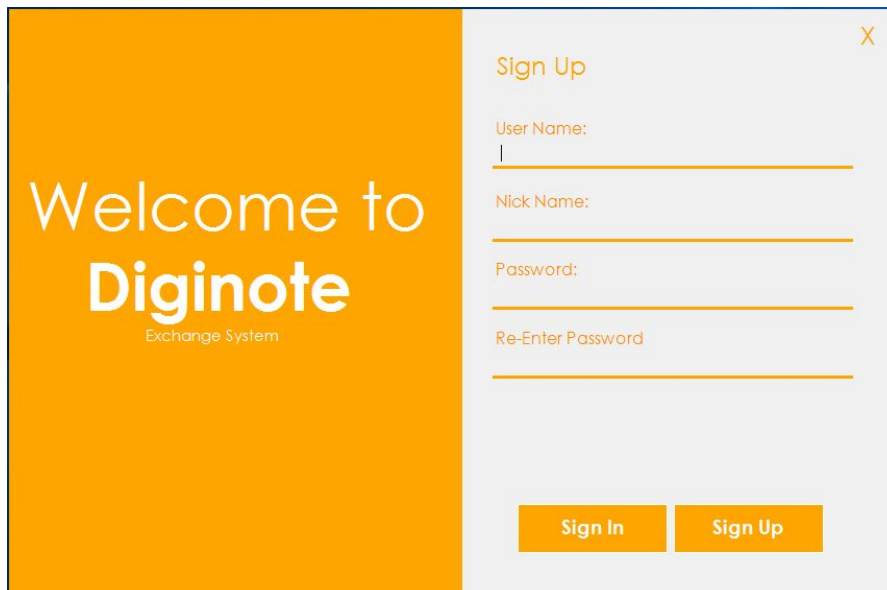
5. *PurchaseOrders* - Idêntica à anterior mas com a informação sobre os pedidos de compra
6. *CompletedOrders* - Guarda informação sobre as transações completadas

Funcionalidades e Testes

Nesta secção serão apresentadas as funcionalidades da aplicação cliente e os testes realizados à mesma.

Login e Register

Quando o utilizador inicia a aplicação depara-se com a janela seguinte, onde é requisitado que preencha os campos para se registar. Após se registar (ao clicar no botão *Sign Up*, o login será feito instantaneamente. Caso o utilizador já possua uma conta, poderá carregar no botão *Sign In* e será atualizada a página para que este possa inserir os campos para fazer login. Na segunda imagem está exemplificado o estado da janela em login.



The image shows a user interface for the 'Diginote Exchange System'. On the left, there is a large orange rectangle with the text 'Welcome to Diginote' in white, with 'Exchange System' in smaller white text below it. On the right, there is a light gray panel with a blue border and a close button 'X' in the top right corner. The panel is titled 'Sign Up' in orange. It contains four input fields with orange labels: 'User Name:', 'Nick Name:', 'Password:', and 'Re-Enter Password:'. At the bottom of the panel, there are two orange buttons: 'Sign In' and 'Sign Up'.

The screenshot shows the Diginote login interface. On the left, an orange panel contains the text "Welcome to Diginote" and "Exchange System" below it. On the right, a light gray panel has a title bar with "Login" and a close button "X". Below the title bar are two input fields: "Nick Name:" and "Password:". At the bottom of the gray panel are two orange buttons labeled "Sign In" and "Sign Up".

[Teste] Esta funcionalidade é bastante simples, a única forma de o utilizador fazer com que a aplicação falhe será aquando a má introdução da informação nos campos. Vários erros poderão acontecer: o utilizador tenta fazer login com username/password não correspondentes; o utilizador tenta se registar com um username já existente ou as passwords não coincidem. Em qualquer um destes teste o utilizador será notificado como demonstra a figura seguinte:

This screenshot shows the same Diginote login interface as the previous one, but with an error state. The "Nick Name:" field now contains the text "zcbstsfm" and the "Password:" field contains six dots. Below the input fields, a red warning triangle icon is displayed next to the text: "Ups! Something Went Wrong Please, Check All Fields and Try Again". The "Sign In" and "Sign Up" buttons remain at the bottom.

Tab Overview

Invés de explicar como é possível efetuar cada funcionalidade individualmente iremos explicar como está estruturada a janela principal e que funcionalidades o utilizador poderá encontrar. Como é possível verificar na imagem seguinte estamos a utilizar uma sidebar para agrupar as funcionalidades.

Começando pela informação comum a todas as tabs, no canto superior direito o utilizador tem duas opções, terminar a aplicação ou terminar sessão. Ambos os botões terminam sessão no entanto, apenas o botão *Logout* retorna às janelas anteriores. O início de sessão e término da mesma correspondem, respetivamente, à subscrição das notificações do servidor e ao cancelamento da mesmas.

No canto superior esquerdo está identificado o nome do utilizador, neste caso, “Teste” e por baixo a barra de navegação pela aplicação.

Teste	Logout	X
Overview	Current Quote	Current Balance
Selling Orders	2,2	638,3
Purchase Orders	Diginotes Owned	Available Balance
	11	603,3
	Add Money	Retrieve Money
	Amount: <input type="text"/>	Amount: <input type="text"/>
	Submit Value	Submit Value

Neste estado da janela, o utilizador poderá consultar informações pertinentes sobre a sua carteira e a quota da moeda:

- No Campo “*Current Quote*” o utilizador poderá consultar o valor atual da moeda, que está atualizado em tempo real. [Teste] Para testar a veracidade deste campo basta iniciar sessão com outro utilizador e ao modificar o valor da diginote voltar a esta janela e verificar que o mesmo foi atualizado sem “carregar em qualquer botão”
- No Campo “*Current Balance*” está a informação sobre a quantidade de dinheiro que o utilizador tem para investir. [Teste] Para testar a veracidade deste campo basta simular a introdução de dinheiro nos textbox’s abaixo e verificar o valor introduzido foi acrescentado ao saldo do utilizador

- No campo “Diginotes Owned” está identificado a quantidade de diginotes que o utilizador possui. [Teste] Para testar a veracidade deste campo basta criar uma nova Purchase order e, noutro utilizador logado que possua diginotes, criar uma Sell order e verificar que o valor é alterado devido à compra de diginotes.
- O campo “Available Balance” é uma “segurança” que o sistema possui para evitar que o utilizador crie infinitas “purchase orders”. É um pouco idêntico ao saldo contabilístico numa conta bancária.

Tab Selling Orders e Tab Purchase Orders

Nas figuras seguintes está representado o layout da tab Purchase Orders e Selling Orders. Ambas partilham o mesmo layout apenas muda o propósito de cada uma.

- No painel do canto superior direito é demonstrada parte da informação presente na tab overview para que o utilizador não tenha de navegar até a overview para confirmar, por exemplo, o dinheiro que possui ou, no caso das selling orders, o número de moedas que possui. Novamente, este valor é atualizado através de notificações do servidor mas também sempre que o utilizador troca de tab como tal, para testar se o servidor envia notificações será necessário manter-se na mesma tab e esperar que o valor em questão se altere.
- No painel à esquerda “*New Purchase Order*” é onde o utilizador poderá definir uma nova ordem. Para tal, basta indicar a quantidade de moedas que deseja comprar/vender. Caso consiga comprar/vender a quantidade indicada, aparecerá uma mensagem no painel “Message” a indicar que a ordem está completada caso contrário, como mostra na segunda figura, aparece uma mensagem a indicar que a ordem não foi completada e é necessário indicar uma nova quota (visível no painel “*Select a new quote...*”, segunda figura desta secção). Após o utilizador inserir a nova quote, e se comprir os requisitos (se está a comprar tem de ser maior ou igual à atual, se a vender menor ou igual à atual, caso contrário é ignorada e aparece uma nova mensagem de erro).
- No Painel “*Change Quote*” o utilizador poderá alterar o valor da quota para as suas ordens. Este painel só é visível se o utilizador tiver ordens pendentes.
- Na Grelha do canto inferior esquerdo, aparecem todas as ordens pendentes. Em cada ordem está demonstrado a data de criação e a quantidade em falta para completar a ordem. Ao comparar as duas figuras seguintes (primeira e segunda figura desta secção), podemos ver que a segunda contém mais uma ordem. Essa ordem foi criada para enunciar o ponto acima, ou seja, o utilizador inseriu uma nova ordem para comprar 100 diginotes, no entanto, como podemos ver, não foi concretizada. Outra particularidade desta listagem de ordens pendentes é que permite ao utilizador apagar uma e confirmar se a quer manter (quando a quote é alterada o utilizador precisa de confirmar se ainda quer manter a quote das transações pendentes). Na figura terceira figura abaixo, aumentou-se o valor da quota, como tal, todas as ordens estão suspensas e o utilizador terá de confirmar se quer manter a ordem (ao clicar em “confirm”) ou apagar a ordem. Se o utilizador tentar apagar uma ordem, uma message box é iniciada para que o utilizador confirme se essa é mesmo a sua vontade. [Teste] Um teste pertinente realizado foi alterar a quota de forma a que todas as buy orders ficassem suspensas e, no

intervalo de 1 minuto, criar uma seller order. Pode-se verificar que a sell order fica pendente e apenas quando as ordens de compra deixam de estar suspensas dá-se a transação.

Teste

Overview

Selling Orders

Purchase Orders

LogoutX

New Purchase Order

Amount:

Submit Value

Currently Buying:

15

Available Balance:

605,3

Quote:

2,2

ID	Data	Quantidade	Confirm	Apagar
1	19/04/2018 03...	4		Delete
2	19/04/2018 03...	5		Delete
3	19/04/2018 03...	1		Delete
20	19/04/2018 07...	5		Delete
*				

Change Quote

Quote:

Submit Value

Message:

Teste

Overview

Selling Orders

Purchase Orders

LogoutX

Select a New Quote equal or smaller than current quote

Quote:

Submit Value

Currently Buying:

115

Available Balance:

385,3

Quote:

2,2


ID	Data	Quantidade	Confirm	Apagar
1	19/04/2018 03...	4		Delete
2	19/04/2018 03...	5		Delete
3	19/04/2018 03...	1		Delete
20	19/04/2018 07...	5		Delete
23	19/04/2018 11...	100		Delete
*				

Change Quote

Quote:

Submit Value

Message:



Not All Diginotes were bought.
You must define a new quote
(higher or equal) than the current
quote (2,2)

Teste

Overview

Selling Orders

Purchase Orders

LogoutX

New Purchase Order

Amount: 100

Submit Value

Currently Buying:

215

Available Balance:

143,8

Quote:

2,3


ID	Data	Quantidade	Confirm	Apagar
1	19/04/2018 03...	4	Confirm	Delete
2	19/04/2018 03...	5	Confirm	Delete
3	19/04/2018 03...	1	Confirm	Delete
20	19/04/2018 07...	5	Confirm	Delete
23	19/04/2018 11...	100	Confirm	Delete
24	19/04/2018 12...	100	Confirm	Delete
*				

Change Quote

Quote: 2,3

Submit Value

Message:

 Quote Successfully updated

Teste

Overview

Selling Orders

Purchase Orders

LogoutX

New Purchase Order

Amount: 100

Submit Value

Currently Buying:

115

Available Balance:

373,8

Quote:

2,3


24	19/04/2018 12...	100	Confirm	Delete
25	19/04/2018 12...	100	Confirm	Delete
26	19/04/2018 12...	100	Confirm	Delete
27	19/04/2018 12...	100	Confirm	Delete
28	19/04/2018 12...	100	Confirm	Delete
*				

Change Quote


Quote: 2,3

Submit Value

Message:

 Quote Successfully updated

Are you sure?

 Are you sure you want to delete this purchase order?
If you choose to delete this purchase order you can undo it by creating another one

SimNãoCancelar

Observações

Apesar de o sistema estar operacional, algumas funcionalidades precisam de atenção para melhor a experiência do utilizador.

Uma das funcionalidades em concreto são as ordens suspensas. Quando a quota é alterada todas as ordens do mesmo tipo ficam suspensas, independentemente de ter sido o utilizador a definir a nova quota (se o utilizador definiu a quota é porque está ciente das ordens que possui). Isto sucede-se devido às propriedades do servidor. Como referido acima, o objetivo era criar um servidor *Stateless* como tal, para o servidor, não interessa quem alterou a quota, o que interessa é que foi alterada. Outro problema é a possibilidade de problemas derivados de concorrência no acesso à base de dados. Quando a quota é alterada o servidor suspende todas as ordens do mesmo tipo e para cada ordem suspenda, faz *schedule* de uma tarefa para após um minuto verificar se a tarefa foi apagada caso contrário é mantida no entanto, se durante esse intervalo a quota é novamente alterada há um novo *schedule*. E temos vários threads a lidar com a mesma ordem. Nos vários testes corridos não houve problemas de concorrência mas caso haja, a ordem poderá não se tornar válida passado 1 minuto, mas sim, quando o utilizador voltar online.

Outra melhoria no projeto é lidar de melhor forma com o “crash” do servidor. A partir do momento que o servidor deixa de funcionar, os utilizadores que estavam subscritos a notificações de servidor deixam de os estar (têm de voltar a fazer login).

O tema do projeto é interessante para testar os conhecimentos em Remoting e apesar dos problemas referidos acima, o grupo está satisfeito com o projeto final.