

Laboratórios de Informática V

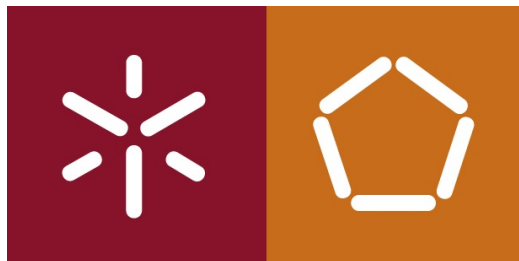
# **Projeto em Tecnologia Blockchain**

## **Transferência de dinheiro**

José Manuel Gonçalves Leitão da Cunha  
(A74702)

Pedro Daniel Gomes Fonseca      César Cachulo  
(A74166)

26 de Junho de 2018  
Universidade do Minho



# 1 Introdução

A tecnologia *blockchain*, com origem nas estruturas criptográficas, tem evoluído rapidamente nos últimos anos e já não é só utilizado no âmbito da cripto-moeda, estando a ser aplicada em muitas outras áreas.

Com este projeto, pretendemos desenvolver um protótipo que permita transferências monetárias, consenso e balanceamento de contas entre utilizadores pertencentes á rede no âmbito da tecnologia *blockchain*. Para tal usufruimos de uma framework chamada de *Hyperledger Fabric*<sup>1</sup> que nos fornece todas as ferramentas necessárias para a nossa implementação. Em concreto, o nosso projeto teve por base uma das suas *samples* chamada de *Balance Transfer*, que implementa muitas das ideias que pretendemos atingir para este trabalho.

Portanto durante este trabalho, iremos apresentar sucintamente o front-end do sistema através de uma *Swing GUI* e o back-end onde se realizaram todas as operações de *blockchain* implementadas para atingir o objetivo pretendido.

---

<sup>1</sup>O Hyperledger Fabric é uma implementação de estrutura blockchain e um dos projetos Hyperledger hospedados pela The Linux Foundation. Destinado como uma base para o desenvolvimento de aplicativos ou soluções com uma arquitetura modular, o Hyperledger Fabric permite que componentes, como consenso e serviços de associação, sejam plug-and-play.

## 2 Interface

Nesta secção, iremos explicar sucintamente todas as funcionalidades que a Interface oferece desenvolvidas em *Swing* no IDE *Netbeans*. O objetivo principal desta aplicação, consiste numa fácil utilização do sistema desenvolvido onde, o utilizador poderá realizar as seguintes operações:

- **Register** - Operação na qual o utilizador poderá registar na aplicação e futuramente ser integrado na rede de *blockchain*.
- **Login** - Operação na qual o utilizador será capaz de realizar o seu login se previamente registado na aplicação.
- **Account Balance** - Operação onde o utilizador é capaz de verificar o saldo da sua conta após ter realizado o login.
- **Transfer Money** - Operação na qual o utilizador realiza uma transferência monetária á sua escolha para uma conta existente após ter realizado o seu login.
- **Deposit Money** - Operação que permite ao utilizador depositar uma quantia monetária na sua conta. Semelhante ao que acontece na operação Register no campo "Account Initial Deposit".

### 2.1 Menu Principal

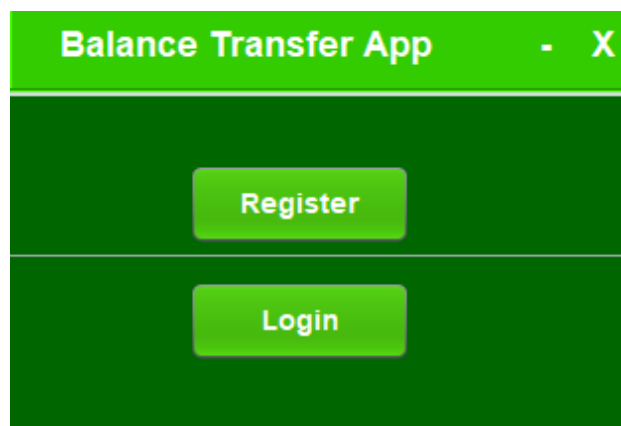


Figura 1: Interface do Menu Principal.

No menu principal, são apresentadas duas opções ao utilizador: Registrar-se e consecutivamente criar uma conta bancária, ou realizar o Login. Além disso, este

poderá minimizar a janela no símbolo "-" ou encerrar o programa no símbolo "X". Esta janela será sempre o ponto de partida na aplicação.

## 2.2 Menu Registro

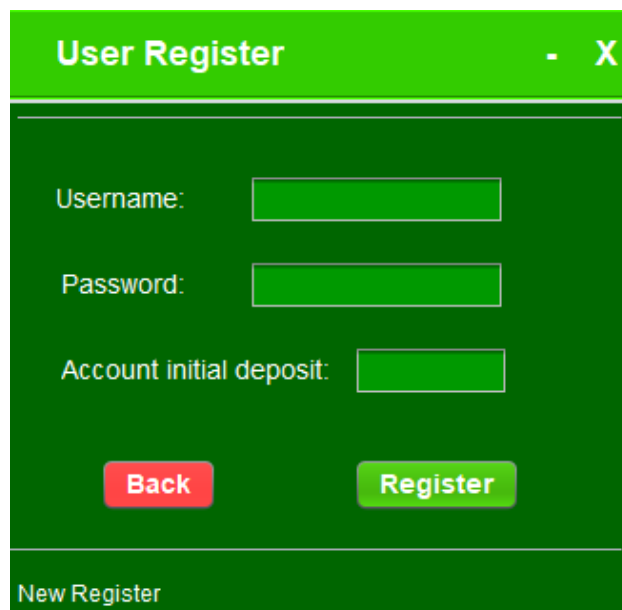
A screenshot of a 'User Register' window. The window has a green title bar with the text 'User Register' and standard window control buttons (minimize, maximize, close). The main area has a dark green background. It contains three text input fields labeled 'Username:', 'Password:', and 'Account initial deposit:'. Below these fields are two buttons: a red 'Back' button and a green 'Register' button. At the bottom of the window, there is a small text label 'New Register'.

Figura 2: Interface do Menu do Registro.

No menu de registro, o utilizador terá que preencher 3 campos. O primeiro campo é o de "Username" onde escreve o nome da sua conta. Em segundo lugar vem o campo "Password", escrevendo a password para a sua conta por forma de obter segurança. E por fim, inicia a sua conta com um depósito de X euros.

Após o preenchimento de todos os campos, o utilizador poderá clicar no botão "Register" obtendo a seguinte mensagem:

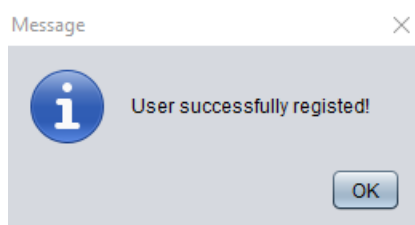


Figura 3: Mensagem de sucesso no registro.

Caso tente realizar um registro com o mesmo nome da conta bancária obterá a seguinte mensagem de erro:

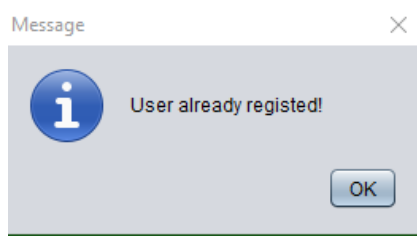


Figura 4: Mensagem de erro no registro.

O utilizador poderá criar uma nova conta bancária no botão "New Register" atualizando a janela. Caso queira retroceder ao Menu principal basta pressionar o botão "Back".

## 2.3 Menu Login

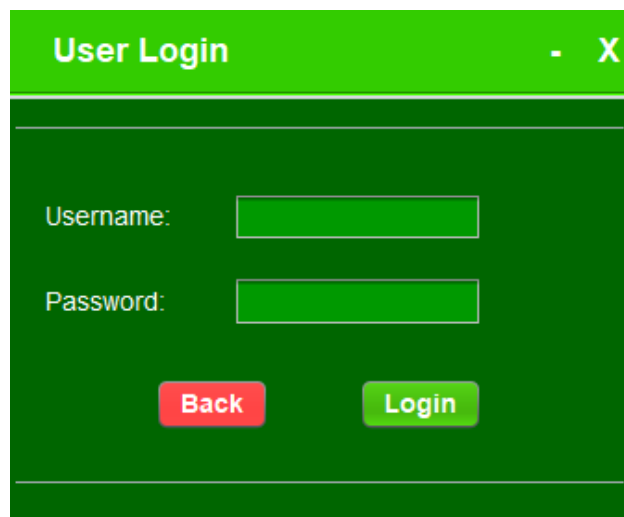
A screenshot of a "User Login" window. The title bar is green and contains the text "User Login" on the left and a close button (X) on the right. The main area has a dark green background. It features two input fields: "Username:" followed by a white rectangular box, and "Password:" followed by a white rectangular box. Below these fields are two buttons: a red button labeled "Back" and a green button labeled "Login".

Figura 5: Interface do Login.

No menu do Login, o utilizador terá de preencher 2 campos. O campo do "Username" e o da "Password".

Caso o utilizador prima o "Login" e os valores nos campos corresponderem a uma conta previamente registada aparecerá a seguinte mensagem a confirmar o Login:

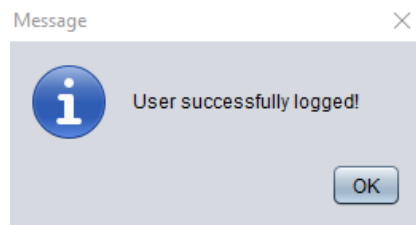


Figura 6: Mensagem de sucesso no login.

Caso o utilizador tenha preenchido um Username/Password errado e tentado fazer login, será apresentada a seguinte mensagem de erro:

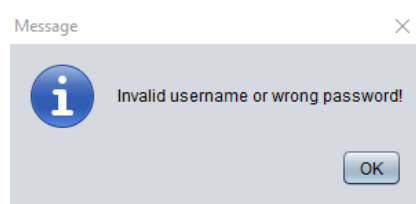


Figura 7: Mensagem de insucesso no login.

O utilizador poderá ainda clicar no botão "Back" para regressar ao Menu Principal.

## 2.4 Menu Options

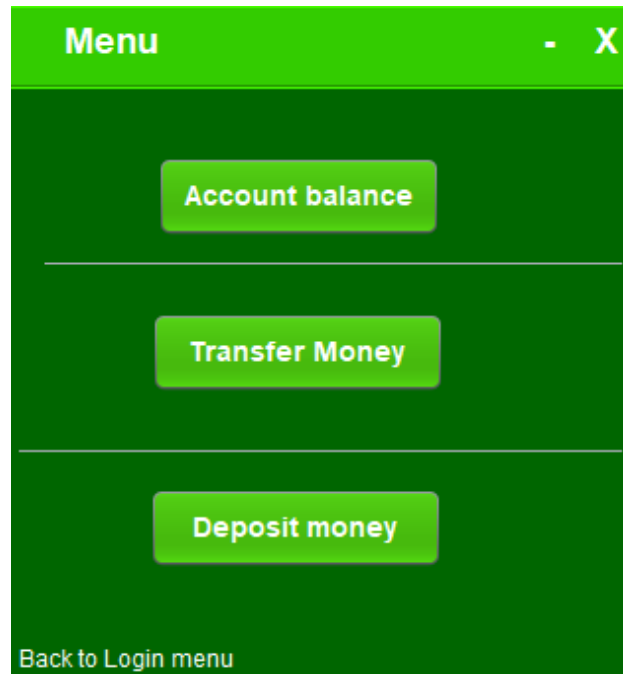


Figura 8: Interface do menu das opções.

Depois de feito com sucesso o Login, o utilizador poderá agora realizar 3 operações. Clicar no botão "Account balance" para inspecionar quanto saldo a sua conta atualmente apresenta ou, clicar no botão "Transfer Money" para ser direcionado para o Menu onde se realiza a transferência monetária de uma conta para outra e "Deposit money" para ir ao Menu onde se deposita uma quantia monetária na conta do utilizador.

Se o utilizador clicar no botão "Account balance" eis a mensagem que aparece:

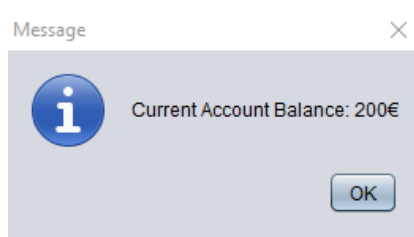


Figura 9: Mensagem de informação do valor de uma conta como exemplo, o seu saldo apresenta 200€.

Agora se o utilizador clicar no botão "Transfer Money" será redirecionado para o Menu Money Transfer que irá ser apresentado na seguinte subsecção.

Além disso, o utilizador poderá sair do Menu Options e regressar ao menu de Login através do click no texto "Back to Login Menu".

## 2.5 Menu Money Transfer

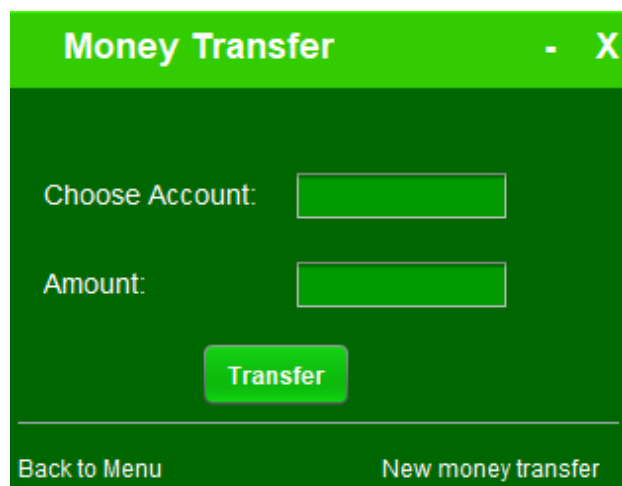
The image shows a window titled "Money Transfer" with a green header bar. Inside the window, there are two input fields: "Choose Account:" and "Amount:". Below these fields is a green button labeled "Transfer". At the bottom of the window, there are two links: "Back to Menu" on the left and "New money transfer" on the right.

Figura 10: Interface do Menu Money Transfer.

Após clicar no botão "Transfer Money" o utilizador será direcionado para este menu. Aqui, é necessário preencher 2 campos. O campo "Choose Account" permite a este, selecionar a conta para qual será transferido o dinheiro introduzido no campo "Amount".

Caso o utilizador pressione o botão "Transfer" poderá ser apresentada duas mensagens. Se o utilizador atual no login tiver na sua conta um saldo superior ao introduzido no campo "Amount" será apresentada esta mensagem:

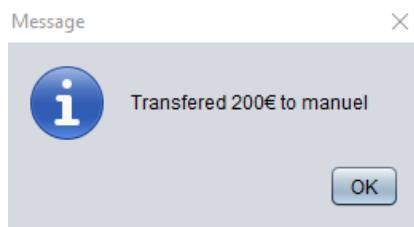


Figura 11: Mensagem de sucesso na transação monetária de 200€ para a conta chamada de "manuel".



No caso contrário, será apresentada a seguinte mensagem de erro:

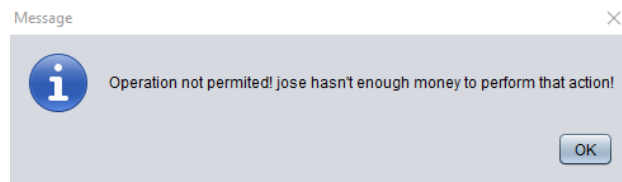


Figura 12: Mensagem de erro na transação monetária a partir da conta "jose".

## 2.6 Menu Deposit Money

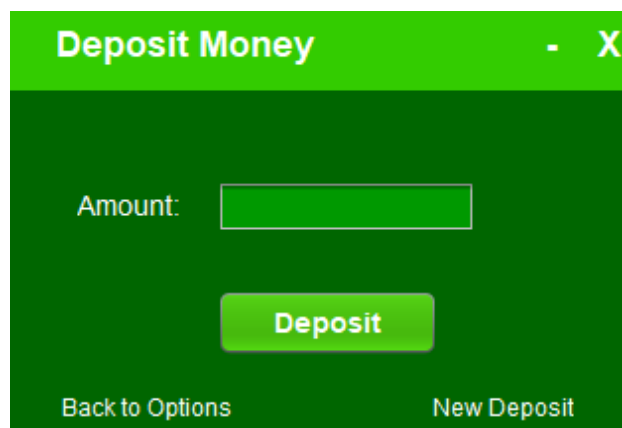


Figura 13: Interface do Menu Money Transfer.

Após clicar no botão "Deposit Money", o utilizador é redirecionado para este menu. Será necessário preencher o campo "Amount" com a quantia a transferir para a sua conta. Depois de preenchido, o utilizador pressiona o botão "Deposit" para realizar a ação pretendida. Em caso de sucesso, será apresentada a mensagem "Balance added!" e no caso contrário, a mensagem "Can't add balance. Error!" é apresentada no ecrã. Além disso, o utilizador pode refrescar a página pressionando no texto "New Deposit" ou regressar ao Menu das opções clicando para tal, o texto que diz "Back to Options".

### 3 Cliente

A interface faz uso de funções disponíveis numa classe desenvolvida chamada "Cliente". Esta classe oferece funcionalidades que, para serem atingidas, comunicam com uma classe chamada "Servidor" que, por sua vez comunica com a rede de forma a disponibiliza-las.

Essas funções são apresentadas a seguir:

- **Registrar utilizador** - Esta função recebe como argumentos o nome do utilizador, a password desejada e o balanço inicial do utilizador e devolve o token do utilizador ou uma string "INSUCESSO". A função também regista o utilizador no servidor e na rede blockchain.
- **Login utilizador** - Esta função recebe como argumentos o nome do utilizador, a password com que o utilizador pretende fazer login o token e produz uma string "SUCESSO" ou "INSUCESSO". Esta função verifica se o utilizador está registado e, se essa condição se verificar, permite que o utilizador usufrua da rede.
- **Obter balanço** - Esta função recebe como argumentos o nome do utilizador e devolve o balanço do mesmo ou uma string "INSUCESSO".
- **Adicionar balanço** - Esta função recebe como argumentos o utilizador e o balanço que se pretende adicionar e devolve uma string "SUCESSO" ou "INSUCESSO".
- **Transferir** - Esta função recebe como argumentos o utilizador que pretende transferir dinheiro, o utilizador destinatário, a quantidade desejada para a transferência e o token do utilizador que pretende transferir dinheiro. Esta função devolve o balanço dos utilizadores após a transferência ou uma string "INSUCESSO".
- **Obter token** - Esta função recebe como argumento o utilizador do qual se quer obter o token e devolve o token do utilizador ou uma string "INSUCESSO".

## 4 Servidor

A classe "Cliente" faz uso de funções disponíveis numa classe desenvolvida chamada "Servidor", tal como referenciado anteriormente. Esta classe comunica com a rede blockchain de forma a usufruir das funcionalidades que a mesma oferece. Essa comunicação é realizada com ajuda da classe "PedidoCURL" que formula os pedidos e os envia para a rede, devolvendo a resposta da mesma.

As funções que a classe "Servidor" disponibilizada são as seguintes:

- **Registrar utilizador** - Esta função recebe como argumento uma string com o formato "registrar nome password balanco" e envia para o utilizador "INSUCESSO" ou o token recebido da rede. A função também regista o utilizador no servidor (guardando o seu nome, password e token em arrays) e na rede blockchain (através do uso de funções da classe "PedidoCURL").
- **Login utilizador** - Esta função recebe como argumentos uma string com o formato "login nome password token" e produz uma string "SUCESSO" ou "INSUCESSO". Esta função verifica se o utilizador está registado no servidor e usando funções da classe "PedidoCURL" permite que o utilizador faça pedidos de transferência.
- **Transferir** - Esta função recebe como argumentos uma string com o formato "transferir utilizador1 utilizador2 quantidade token" e devolve ao cliente o balanço dos utilizadores após a transferência ou uma string "INSUCESSO". A string devolvida com os balanços dos utilizadores é resultado do output produzido pela rede após o pedido de transferência feita com ajuda de funções da classe "PedidoCURL".
- **Obter balanço** - Esta função recebe como argumento uma string com o formato "balanco user" e devolve o balanço do utilizador ao cliente ou uma string "INSUCESSO".
- **Adicionar balanço** - Esta função recebe como argumento uma string com o formato "adicionar user balancoadicionar" e adiciona balanço à conta de um utilizador e envia uma string "SUCESSO" ou "INSUCESSO" ao cliente.
- **Obter token** - Esta função recebe como argumento uma string com o formato "token user" e envia o token do utilizador ou uma string "INSUCESSO" ao cliente.

## 5 Pedidos

A classe "PedidosCURL" torna possível ao servidor fazer pedidos à rede blockchain e obter as respostas que a rede devolve.

As funções que esta classe disponibiliza são as seguintes:

- **Inserir utilizador na organização** - Esta função recebe como argumentos o nome do utilizador e o nome da organização a que se pretende adicionar o utilizador. A rede responde se a operação foi um sucesso ou não e o token do utilizador.
- **Criar canal** - Esta função recebe como argumentos o token do utilizador que pretende criar o canal e o nome do canal que pretende criar. A rede responde se a operação foi um sucesso ou não.
- **Juntar a canal** - Esta função recebe como argumentos o token do utilizador que pretende juntar-se ao canal, o nome do canal a que o utilizador se pretende juntar e uma lista de peers que são responsáveis pela integridade da rede. A rede responde se a operação foi um sucesso ou não.
- **Instalar chaincode** - Esta função recebe como argumentos o token do utilizador de um canal, uma lista de peers e a versão da chaincode que se vai instalar. A rede instala a chaincode nos peers que recebeu como argumento e responde com sucesso ou insucesso.
- **Instanciar chaincode** - Esta função recebe como argumentos o token, argumentos e a versão da chaincode. Os argumentos vão conter os utilizadores e os seus balanços no formato, por exemplo, ["jorge","650","ana","100"]. A rede responde com sucesso ou insucesso.
- **Mover (ou transferência)** - Esta função recebe como argumentos o token do utilizador que pretende fazer a transferência, os utilizadores origem e destino e a quantidade de dinheiro para transferir. A rede responde com sucesso ou insucesso.
- **Query chaincode** - Esta função recebe como argumentos o token do utilizador que deseja fazer a query, o peer a qual deseja fazer a query, a função a que deseja obter resposta e argumentos. O único uso que foi dado a esta função foi o de obter o balanço de um utilizador enviando como argumentos apenas uma string no formato "jorge" onde jorge é o utilizador cujo saldo pretendemos conhecer. A rede responde com a resposta à query.

Outras funções foram implementadas mas não tiveram utilidade para o que era pretendido uma vez que consistem em funções que nos davam informações sobre a organização e funcionamento da rede.

## 6 Conclusão

Com este trabalho, obtivemos um melhor conhecimento de como funciona as redes *blockchain* permissionadas e toda a estrutura de segurança que é utilizada bem como alguns detalhes de funcionamento relacionados com a implementação da mesma. Foram encontrados bastantes problemas e grande parte deles foram ultrapassados. Grande parte deles tiveram origem na falta de informação detalhada existente fazendo com que a principal forma de os ultrapassar fosse a de "tentativa e erro". A ferramenta tem muitas funcionalidades interessantes que não foram exploradas com detalhe no projeto mas que para projetos mais complexos ou de outra natureza são muito úteis, possibilitando o desenvolvimento de projetos muito criativos.