

– *MIEIC Sistemas Distribuídos* –

Aplicação para Bancas Comunitárias

(Relatório Final)

29 de Maio de 2016

Grupo:
David Azevedo - ei12026@fe.up.pt
Luís Reis - ei11009@fe.up.pt
Maria Miranda - ei12046@fe.up.pt

Índice

[1 – Introdução](#)

[2 – Arquitetura](#)

[2.1 – Servidor](#)

[2.2 – Cliente](#)

[3 – Implementação](#)

[3.1 – Componentes](#)

[3.2 – Concorrência](#)

[4 – Informações Relevantes](#)

[4.1 – Segurança](#)

[4.2 – Tolerância e Falhas](#)

[5 – Considerações Finais](#)

[5.1 – Conclusão](#)

[5.2 – Dificuldades](#)

[5.3 – Esforço Dedicado](#)

[5.2 – Melhorias](#)

1 – Introdução

Este relatório está associado e pretende demonstrar o desenvolvimento do projeto Aplicação para Bancas Comunitárias para a unidade curricular de Sistemas Distribuídos onde será caracterizado, definido e analisado, de forma detalhada, os aspetos referentes ao mesmo.

Esta aplicação consiste no agrupamento de vários utilizadores numa sala exclusiva de modo a que consigam partilhar ficheiros entre si, de forma relativamente simples e rápida. No âmbito do projeto estes ficheiros serão idealmente “apostas” para que os elementos constituintes da “banca” possam avaliar o risco e dessa forma aprovarem ou rejeitarem as variadas “apostas”.

Por fim, este relatório está dividido nas seguintes secções:

- **Arquitetura** – descrição das componentes principais da aplicação e como interagem;
- **Implementação** – descrição das bibliotecas ou frameworks e a sua utilização no desenvolvimento da aplicação, como a aplicação gere a concorrência e as respectivas mensagens;
- **Detalhes Relevantes** – descrição de certos aspectos, tais como: segurança e tolerância a falhas;
- **Conclusões** – sugestões e melhoramentos a realizar ao projeto, esforço dedicado por cada elemento do grupo, assim como as dificuldades encontradas ao longo do desenvolvimento;
- **Referências** – indicação das referencias bibliográficas.

2 – Arquitetura

A arquitetura da nossa aplicação tem 2 tipos de componentes básicos: um servidor e vários clientes, sendo estes últimos computadores pessoais.

2.1 – Servidor

A função principal do nosso servidor é gerir os pedidos de conexão dos clientes para se juntarem à aplicação e posteriormente a uma sala. Recebe também os pedidos por parte dos clientes de upload, download e delete dos ficheiros de uma sala.

Os pedidos a que o servidor responde são os seguintes: inicialmente o servidor está a correr e recebe um pedido do utilizador, este reencaminha o utilizador para a página de login, onde o Cliente se pode registar ou, se já estiver registado, fazer log in na aplicação com os seus dados, email e password, sendo depois reencaminhado para a página das salas, estes dois pedidos, log in e registo do cliente são ambos pedidos Post ao Servidor; Se o utilizador já estiver associado a uma sala, então é reencaminhado para essa mesma sala, se não tem a opção de criar uma ou juntar-se a uma através de um código específico, tanto quando o utilizador se junta a uma sala como quando cria, é feito um pedido Post ao Servidor para fazer o join à sala ou a criação da mesma; Na página da sala o Cliente tem as opções de Delete, Upload e Download, quando um cliente faz um Upload é um pedido Post ao servidor, e outro pedido Get para ir buscar o ficheiro que o utilizador quer fazer o Upload, para o Delete, Download e Listagem de ficheiros o Cliente faz pedidos Get ao Servidor.

2.2 – Cliente

O ecrã inicial do Cliente tem apenas os campos para o mesmo poder fazer log in e assim aceder à sua sala ou criar uma, bem como a parte de registo de um utilizador. Após esta página o Cliente tem a possibilidade de criar uma sala nova ou juntar-se a outra, já existente, através de um código referente a uma sala específica. Depois disto, os Clientes são reencaminhados para a página da sala onde é apresentada uma tabela com a lista dos dados da mesma, a lista dos ficheiros que estão associados à sala e um botão para fazer o Upload de ficheiros.



Email

Password

Name

John Doe

Username

Email

john1009@le.up.pt

Password



Express

Welcome to Community Banks

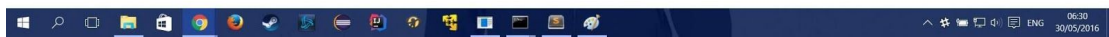
Room Info

Room Data
Room Name: Proj2
Room Code: GcwmEr

File List

File Name Download Delete

No file chosen





Create Room

Room Name:

Create

Join Existent Room

Code:

Join



Express

Welcome to Community Banks

Room Info

Room Data
Room Name: Proj2
Room Code: Gcwm5Er

File List

File Name Download Delete

Choose File No file chosen

Upload

Logout



3 – Implementação

Nesta secção é indicado a utilização de libraries ou frameworks (caso tenham sido usadas) e é descrito como a aplicação gere a concorrência e as respectivas mensagens.

3.1 – Componentes

Para o nosso projeto usamos node.js e módulos de node para fazer as rotas, para fazer a ligação à base de dados, para gerir a sessão dos utilizadores, para gerir os ficheiros, bem como para o envio das notificações (quando é feito o download, upload ou delete de um ficheiro) e para a encriptação dos dados, mais especificamente, para a encriptação das passwords.

3.2 – Concorrência

Por forma a minimizar os problemas de concorrência o nosso Servidor trabalha com pedidos assíncronos podendo assim responder aos pedidos feitos pelos Clientes sem espera circular.

4 – Informações Relevantes

4.1 – Segurança

Relativamente à Segurança podemos dizer que foi usado o módulo BCrypt de node.js para encriptar as passwords dos Clientes aumentando assim a segurança da nossa Aplicação.

4.2 – Tolerância e Falhas

Quanto a questões de falhas e tolerância podemos dizer que quando o Cliente falha é necessário que o mesmo volte a fazer log in, não perdendo nunca os seus dados, salas e ficheiros associados à mesma.

Relativamente ao Servidor, este não falha quando recebe os pedidos mas no caso de haver um problema com a base de dados ou com a rede, o Servidor não fica operacional, já se for falha por parte do Cliente, como disse anteriormente, não há qualquer problema, o mesmo só tem de voltar a fazer log in.

5 – Considerações Finais

5.1 – Conclusão

Após a realização deste projeto o nosso grupo concluiu que este trabalho foi uma mais valia para o aumento do conhecimento do mesmo no que diz respeito ao conceito de Sistemas Distribuídos assim como para aprendermos a trabalhar melhor com node.js e aplicações rest.

Relativamente ao produto final, pensamos que no conjunto global da Aplicação os objectivos base foram cumpridos, isto porque conseguimos implementar o projeto que idealizamos e que abordamos na proposta inicial alterando apenas os tópicos que o Professor Pedro Souto referiu. Esses tópicos eram a criação de canais privados de comunicação, que passaram a ser as salas privadas onde os utilizadores têm os seus ficheiros disponíveis para download e onde podem também fazer o upload dos mesmos.

5.2 – Dificuldades

As nossas principais dificuldades centraram-se principalmente no desenvolvimento da arquitetura da base de dados , e por vezes também tivemos alguns problemas a nível de rotas .

5.3 – Esforço Dedicado

- David Emanuel dos Santos Silva - 33%
- Luís Brochado Pinto Reis - 33%
- Maria João Pombinho Miranda - 33%

No total foram dedicadas aproximadamente 32 horas para se concluir o trabalho.

5.4 – Melhorias

Algumas melhorias que o nosso grupo pensou que poderiam ser feitas à nossa Aplicação foram as seguintes:

- Criação de um chat para termos conversação na sala;
- Poderíamos também melhorar a parte visual da Aplicação;
- E permitir que um utilizador faça parte mais de mais do que uma sala.

6 – Referências

- StackOverflow - <http://stackoverflow.com/> ;
- <https://scotch.io/tutorials/using-mongoosejs-in-node-js-and-mongodb-applications>