



UNIVERSIDADE
LUSÓFONA

Sistema de Autenticação para a Bateria Sistémica de Lisboa

Trabalho Final de curso

Nome do Aluno: André Santos 21705309
Nome do Aluno: Guilherme Borges 21705312
Nome do Orientador: José Faísca

Trabalho Final de Curso | LEI | 06/12/2019

www.lusofona.pt

Direitos de cópia

Sistema de Autenticação para a Bateria Sistémica de Lisboa, Copyright de André Santos e Guilherme Borges, ULHT.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Índice

1. Índices de Quadros, Figuras e Tabelas	4
2. Introdução.....	5
3. Resumo	6
4. Abstract (em Inglês)	7
5. Identificação do Problema	8
6. Levantamento e análise dos Requisitos	9
6.1. Requisitos Funcionais.....	9
6.2. Requisitos Não-Funcionais.....	10
6.3. Novos Requisitos Adicionados.....	10
7. Viabilidade e Pertinência.....	12
8. Solução Desenvolvida	13
9. Benchmarking.....	23
10. Método e planeamento.....	24
11. Resultados.....	25
12. Conclusão e trabalhos futuros.....	26
13. Bibliografia.....	27
14. Anexos	28
15. Manual técnico da aplicação.....	29
16. Manual de utilizador da aplicação	30

1. Índices de Quadros, Figuras e Tabelas

Figura 1 - Servidor LDAP	13
Figura 2 - Verificação das portas.....	14
Figura 3 - Ifconfig.....	14
Figura 4 - Configuração LDAP	15
Figura 5 - Hierarquia conceptual	15
Figura 6 - Configuração da estrutura no LDAP	16
Figura 7 - Biblioteca LDAPJS.....	16
Figura 8 - Login com Administrador.....	17
Figura 9 - Função para autenticar o Administrador.....	17
Figura 10 - Página de gestão do Administrador	18
Figura 11 - Página de Registo.....	18
Figura 12 - Função de adicionar um paciente.....	19
Figura 13 - Página de remoção de um paciente.....	19
Figura 14 - função para eliminar um paciente	20
Figura 15 - Página de listar Pacientes.....	20
Figura 16 - Função de Pesquisar Pacientes nas Organizações	21
Figura 17 - Página de Log dos Pacientes.....	21
Figura 18 - Login em unity para o paciente.....	22
Figura 19 - Função de autenticação do paciente.....	22

2. Introdução

Nesta última entrega do projeto de TFC sobre a Bateria Sistemica de Lisboa, melhorámos a nossa implementação do LDAP, implementámos inserção de pacientes, a pesquisa de vários pacientes nas várias organizações e registo de logins (data e paciente) dos pacientes após os mesmos se terem autenticado através do jogo desenvolvido em unity.

Para o administrador conseguir controlar a sua organização, criamos um login em web(HTML) que após este ter feito a sua autenticação no LDAP, terá acesso as respetivas funcionalidades referidas anteriormente.

Para o paciente, criamos um login utilizando a plataforma unity em que o mesmo é autenticado através do LDAP.

3. Resumo

A Bateria Sistemática de Lisboa é uma plataforma computadorizada, que tem sido utilizada em vários estudos para avaliação e estimulação cognitiva. Esta plataforma consiste numa cidade virtual, na qual é possível estimular/treinar a realização de tarefas virtuais, as quais, descrevem atividades de vida diária. A premissa inerente a este procedimento é a estimulação ou treino cognitivo, contextualizado em tarefas da vida diária, permite maximizar os ganhos da intervenção cognitiva, na medida em que promove a generalização das competências adquiridas durante a intervenção.

Durante estas tarefas, os pacientes podem mover-se livremente no ambiente virtual com a utilização de dispositivos periféricos, como o rato, o teclado ou o joystick. O comportamento e a forma de interação com os objetos são rastreados automaticamente pelo programa, o que permite igualmente aferir sobre o desempenho dos indivíduos ao longo da prova.

Esta prova tem sido continuamente atualizada e neste momento estão disponíveis as seguintes tarefas virtuais:

- a) tarefa de higiene matinal, com utilização da escova de dentes e duche;
- b) teste do roupeiro, para escolha de peças de roupa;
- c) teste da sapateira, para emparelhamento de sapatos;

Estas tarefas visam avaliar e estimular/treinar funções cognitivas, como memória, atenção, e funcionamento executivo, numa abordagem funcional, que envolve a contextualização de cada um destes domínios em tarefas virtuais da vida diária.

Neste momento a aplicação atual da BSL é uma aplicação Stand-alone, ou seja, a aplicação é capaz de funcionar de forma independente de qualquer tipo de hardware ou software e para além desta característica a aplicação funciona ainda em modo mono-utilizador, isto é, pode ser acedida ou usada apenas por um utilizador. Esta implementação torna impossível a capacidade de verificação de acessos e controlo dos resultados derivados das tarefas virtuais realizadas na aplicação.

4. Abstract (em Inglês)

The Lisbon Systemic Battery is a computerized platform that has been used in various studies for assessment and cognitive stimulation. This platform consists of a virtual city, where it is possible to stimulate / train the accomplishment of virtual tasks that describe activities of daily life. The inherent assumption in this procedure is that cognitive stimulation or training, contextualized in daily life tasks, will maximize the gains of cognitive intervention, as it promotes the generalization of skills acquired during the intervention.

During these tasks, the patients can move freely in the virtual environment using peripheral devices such as the mouse, keyboard or joystick. The behavior and the way the patient interacts with the objects are automatically tracked by the program, which also allows us to verify the performance of individuals throughout the test.

This test has been continuously updated and at this time the following virtual tasks are available:

- a) morning hygiene task, using the toothbrush and shower;
- b) wardrobe test, for choosing clothes;
- c) shoe rack test for pairing shoes;

These tasks aim to evaluate and stimulate / train cognitive functions, such as memory, attention, and executive functioning, in a functional approach, which involves the contextualization of each of these domains in virtual tasks of daily life.

Right now the current LSB application is a Stand-alone app, which means that the app is capable of working independently of any hardware or software, it also works in a single user mode, this means it can only be accessed or used by only one user. This implementation makes it impossible to verify accesses and to control the results of the virtual tasks performed in the application.

5. Identificação do Problema

O nosso trabalho consiste na implementação de um sistema de autenticação de utilizadores na Bateria Sistémica de Lisboa.

O problema principal encontra-se ao nível da natureza independente da referida aplicação. Neste momento a solução do departamento é standalone o que traz vários problemas ao departamento ao nível de manutenção da aplicação e escalabilidade para os restantes departamentos.

É assim necessário transformar a aplicação numa solução distribuída/ web based, onde seja possível funcionar em modo multi-utilizador com autenticação e controlo de acessos

Nesta última entrega, o nosso projeto já consegue autenticar os pacientes/administradores e os administradores conseguem controlar os acessos dos pacientes. Assim qualquer utilizador que utilize esta aplicação terá então de se conectar ao LDAP que será o único ponto de autenticação da mesma.

6. Levantamento e análise dos Requisitos

De acordo com algumas informações que nos foram sendo fornecidas ao longo do semestre sobre o projeto a realizarmos, identificámos alguns requisitos que nos pareceram mais críticos. Estes requisitos foram já verificados e aceites pelo professor Jorge Oliveira do departamento de Psicologia.

6.1. Requisitos Funcionais

UC	Nome UC	Descrição UC	Estado
UC-01	Capacidade de alojamento	O diretório deverá conseguir alojar e gerir a informação de 7 organizações, neste caso terá no mínimo 1 organização e no máximo 15 organizações. Se o diretório tiver apenas uma organização não terá de ter um administrador local.	
UC-02	Número de utilizadores	Cada organização pode conter no máximo 30 pessoas. É necessário pelo menos um utilizador por organização e pessoas de uma organização podem ir a outras organizações se não exceder as 30 pessoas por organização.	
UC-03	Gestão de Administrador	Em cada organização existirá uma pessoa que será administradora e esta não pode ser paciente	
UC-04	Registo (Alterado)	Cada registo será composto por um nome, apelido, email, organização e password apenas será guardado um histórico de tarefas efetuadas. Todas as informações estarão bloqueadas a utilizadores com um baixo nível de autorização.	
UC-05	Gestão do Administrador Geral	Existirá um administrador geral que terá acesso às informações de todas as organizações. O administrador geral tem permissões para aceder a todas as organizações;	

6.2. Requisitos Não-Funcionais

UC	Nome UC	Descrição UC	Estado
UC-06	Restrição de Administrador	Em cada organização existirá um administrador que consegue ver todos os registos/ informações da sua organização. Os Administradores não terão acesso a outras organizações, deste modo não poderão ver as informações de participantes de outras organizações.	
UC-07	Validação de credências	Para um paciente ter acesso às tarefas terão que ter um registo feito com credências validadas. Apenas pacientes terão as contas validadas para não sobrecarregarem o sistema. Apenas após um login ter sido efetuado será mostrado ao paciente as tarefas que este poderá realizar.	
UC-08	Controlo de histórico	O administrador terá acesso ao histórico dos pacientes.	
UC-09	Restrição do Administrador Geral	O administrador geral tem a possibilidade de poder verificar o histórico de todos os pacientes.	
UC-10	Limitação local de Login	Um paciente não pode ter login feito em dois computadores em simultâneo.	
UC-11	Limitação de Login de organizações	Um paciente não pode ter o login feito em organizações diferentes em simultâneo.	

6.3. Novos Requisitos Adicionados

UC	Nome UC	Descrição UC	Estado
UC-12	Gerir plataforma	O administrador deverá ter uma plataforma para gerir a organização.	
UC-13	Remover paciente	O administrador poderá remover um paciente.	
UC-14	Adicionar paciente	O administrador poderá adicionar um novo paciente.	
UC-15	Verificar pacientes	O administrador poderá verificar as pessoas que estão registadas.	
UC-16	Logs de pacientes	O administrador poderá verificar que pacientes fazem login no “jogo”. Guardar informação em ficheiro.	

Legenda:

Implementado
Parcialmente implementado
Não Implementado

7. Viabilidade e Pertinência

Em relação à viabilidade e em relação ao problema identificado existem várias soluções para a sua resolução, no entanto, à primeira vista será necessário ter alguns parâmetros delimitados, como por exemplo, a existência de um servidor da Universidade Lusófona de Humanidades e Tecnologias que iria assegurar uma maior privacidade dos utilizadores, ao mesmo tempo iria também providenciar uma maior segurança, visto que, apenas os responsáveis sobre a Bateria Sistemática de Lisboa iriam ter acesso às informações dos utilizadores.

Tendo em conta que vamos utilizar o protocolo LDAP, no contexto deste projeto podemos tirar partido dos seguintes recursos:

- Autenticação da máquina
- Autenticação de usuário
- Grupos de Usuários / Sistemas
- Representação da Organização
- Rastreamento de ativos
- Gerenciamento de recursos do usuário
- Pesquisas de endereço de email

Quanto à pertinência é um projeto importante, uma vez que a aplicação atual da BSL não possui um controlo de acessos à aplicação nas várias organizações, estando desta forma guardados todos os resultados em cada máquina, com o desenvolvimento em web based conseguimos mitigar esse problema pois a aplicação vai estar conectada ao servidor conseguindo assim criar uma ligação entre o login do utilizador e os seus resultados.

Com esta alteração os médicos têm a possibilidade de ter uma base de dados com todos os pacientes e a sua performance nas suas tarefas conseguindo assim avaliar mais facilmente a evolução das suas capacidades cognitivas.

Tendo em conta a solução produzida e os resultados obtidos através dos testes conseguimos tomar partido de alguns dos melhores pontos do LDAP tal como, autenticação de usuário e a representação da organização.

Em relação à pertinência do projeto, tal como foi dito nos relatórios anteriores, é muito importante para a BSL, visto que a mesma não fazia controlo dos seus utilizadores, podendo isto ser importante para estudar a evolução dos pacientes ao longo do tempo.

8. Solução Desenvolvida

A tecnologia principal que vamos utilizar no nosso projeto será o LDAP.

O LDAP (Lightweight Directory Access Protocol), como o nome sugere é um protocolo leve para aceder serviços de diretórios. O LDAP corre através de uma ligação TCP/IP. Estes diretórios são geralmente relativos a uma ou múltiplas organizações.

O modelo de informações do LDAP é baseado em entradas, em que cada uma é uma coleção de atributos que possui um atributo globalmente único (chamado DN (Distinguished Name)), e cada um dos atributos possui um tipo e um ou mais valores.

No LDAP as entradas do diretório são organizadas numa estrutura hierárquica em forma de árvore, normalmente estas estruturas refletem os limites organizacionais.

O protocolo LDAP define os métodos para aceder aos dados no servidor da aplicação, permitindo assim à aplicação fazer a conexão e desconexão, comparação de informações, alteração das mesmas (inserção, alteração ou exclusão) e também uma pesquisa sobre informações no diretório.

O LDAP fornece um mecanismo para que um cliente se autentifique ou prove a sua identidade perante um servidor de diretórios. Desta forma é permitindo um controlo de acessos avançado para proteger as informações que o servidor contém, visto que alguns dos serviços de diretório não oferecem qualquer tipo de proteção.

O LDAP também suporta serviços de segurança de dados (integridade e confidencialidade).

Esta tecnologia utiliza um modelo cliente-servidor, em que um ou mais servidores LDAP contêm os dados que compõem a árvore de informações do diretório (DIT (Directory Information Tree)). Independentemente do servidor LDAP ao qual o utilizador se conecta, ele vê o mesmo diretório.

Concluindo, no contexto do projeto a tecnologia do LDAP irá facilitar o controlo e gestão de informações relativas aos utilizadores de diferentes organizações e permitirá a sua autenticação.

Na configuração do ambiente de desenvolvimento do projeto (BSL), criamos uma máquina virtual em Linux (Ubuntu versão 18.04.4 LTS) e decidimos optar por utilizar o LDAP Apache Directory Studio uma vez que este possui uma interface gráfica que nos facilitou e acelerou a configuração do ambiente teste. O processo de configuração do LDAP foi muito simples, o primeiro passo depois da sua instalação foi a configuração do servidor que ficou com o nome de “ApacheDS 2.0.0”.

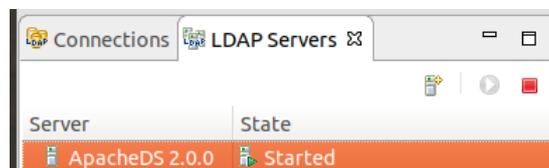


Figura 1 - Servidor LDAP

Após este processo estar completo apenas foi necessário criar uma conexão ao servidor, escolhemos para o nome da conexão “TFC-LDAP” e com a ajuda do terminal verificamos o Hostname com o comando “ifconfig”

```
tfc@tfc-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::acd:b8ab:5b0a:4eca prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:be:80:f7 txqueuelen 1000 (Ethernet)
    RX packets 259364 bytes 310420508 (310.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 51851 bytes 3231713 (3.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Loopback Local)
    RX packets 373 bytes 34676 (34.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 373 bytes 34676 (34.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figura 3 - Ifconfig

```
tfc@tfc-VirtualBox:~$ netstat -l
Ligações de Internet Activas (só servidores)
Proto Recv-Q Send-Q Endereço Local          Endereço Remoto          Estado
tcp        0      0 localhost:domain        0.0.0.0:*                ESCUTA
tcp        0      0 localhost:ipp           0.0.0.0:*                ESCUTA
tcp6       0      0 [::]:10636             [::]:*                  ESCUTA
tcp6       0      0 [::]:10389             [::]:*                  ESCUTA
tcp6       0      0 ip6-localhost:ipp     [::]:*                  ESCUTA
tcp6       0      0 localhost:40929        [::]:*                  ESCUTA
udp        0      0 0.0.0.0:ipp           0.0.0.0:*                *
udp        0      0 0.0.0.0:33904         0.0.0.0:*                *
udp        0      0 0.0.0.0:mdns          0.0.0.0:*                *
udp        0      0 localhost:domain       0.0.0.0:*                *
udp        0      0 0.0.0.0:bootpc        0.0.0.0:*                *
udp6      0      0 [::]:mdns             [::]:*                  *
udp6      0      0 [::]:57464            [::]:*                  *
raw6      0      0 [::]:ipv6-icmp        [::]:*                  7

Sockets de domínio UNIX activos (só servidores)
Proto RefCnt Flags      Type       State      I-Node   Rota
unix    2      [ ACC ] SEQPACOTE  ESCUTANDO 1313     /run/udev/control
unix    2      [ ACC ] STREAM    ESCUTANDO 25073    /run/user/1000/systemd/private
unix    2      [ ACC ] STREAM    ESCUTANDO 25077    /run/user/1000/bus
```

Figura 2 - Verificação das portas

e utilizamos o comando “netstat -e” (mostra sockets de servidor à escuta) para saber em que porta é que comunicava com o servidor uma vez que sabemos que a conexão do LDAP é através do protocolo TCP foi apenas necessário procurar na lista do output a respetiva comunicação.

Para finalizar as propriedades da conexão estão descritas na figura 4.

Network Parameter Authentication Browser Options Edit Options

Connection name: TFC-LDAP

Network Parameter

Hostname: 127.0.0.1

Port: 10389

Connection timeout (s): 30

Encryption method: No encryption

Server certificates for LDAP connections can be managed in the ['Certificate Validation'](#) preference page.

Provider: Apache Directory LDAP Client API

Check Network Parameter

Read-Only (prevents any add, delete, modify or rename operation)

Figura 4 - Configuração LDAP

Uma vez concluída a configuração realizámos um (desenho) da árvore de diretório para perceber conceptualmente as hierarquias, ramificações e organizações do LDAP da Lusofona como demonstrado na Figura 1.

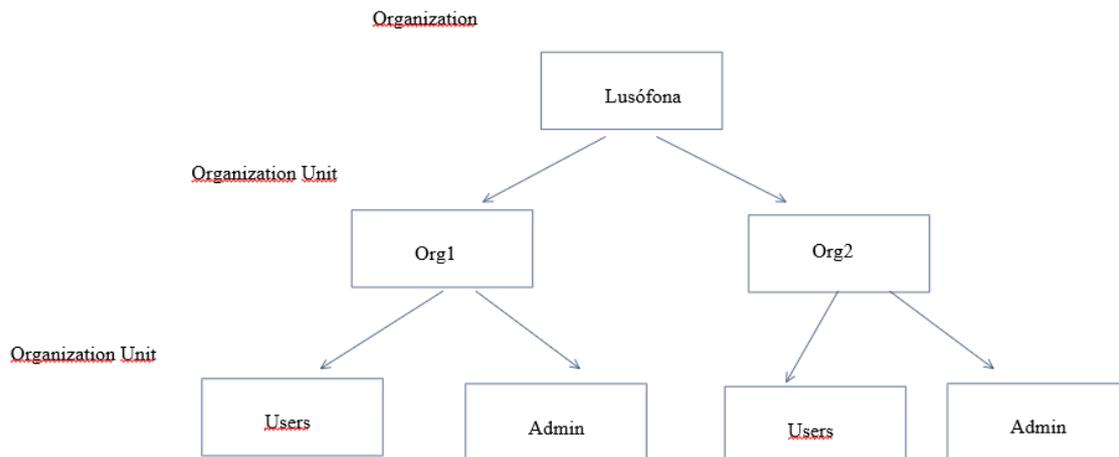


Figura 5 - Hierarquia conceptual

Como demonstrado na Figura 5 dividimos a hierarquia em 3 níveis, o primeiro é a organização Lusofona que terá acesso a todas as informações existentes em todas as organizações. O segundo nível é o das organizações, o terceiro e último são os constituintes das organizações sendo estes pacientes e apenas um administrador que apenas terá acesso à sua respetiva organização.

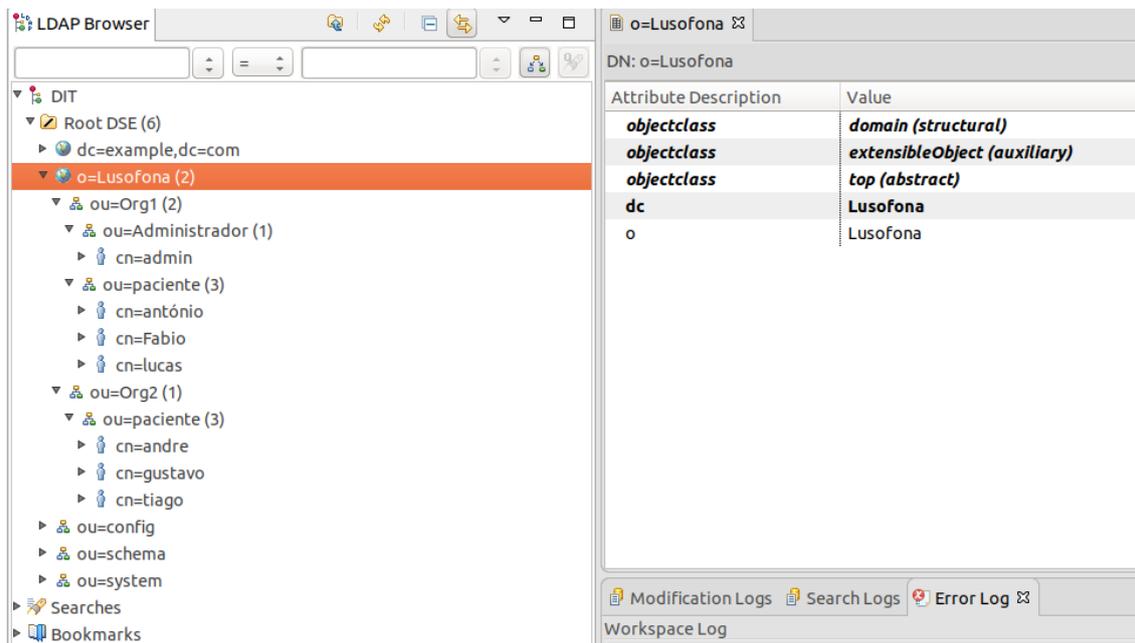


Figura 6 - Configuração da estrutura no LDAP

Ao longo do projeto optamos por dividir as autenticações dos pacientes e dos administradores, uma vez que não faria sentido o administrador ter de entrar no jogo para conseguir controlar os registos e a deleção de pacientes.

Por isso a nível dos administradores criamos uma solução em web(HTML) e para os pacientes criamos uma solução em unity(C#), tanto os pacientes como os administradores se conectam ao mesmo servidor(node.js) que contém as funções necessárias para ser possível completar as autenticações, registos e eliminações de utilizadores.

Começando pelo servidor (que é utilizado tanto pelos administradores como pelos pacientes), para fazer a ligação ao LDAP utilizamos uma biblioteca chamada "ldapjs".

O ldapjs é uma biblioteca de javascript pura que é utilizada para implementar clientes LDAP e servidores em node.js.

```
var ldap = require('ldapjs');
```

Figura 7 - Biblioteca LDAPJS

Na implementação do servidor utilizamos 3 bibliotecas, a express, a http e a socket.io(utilizada para criar a comunicação).

Para verificar se o utilizador está conectado ao servidor utilizamos o “io.sockets.on('connection', function (socket)” que vai verificar se o cliente está ligado, com o “socket.on('disconnect',function (data)” verificamos quando o utilizador se desconecta.

Para tratar a informação que vem dos utilizadores temos 2 funções, uma apenas para os utilizadores que se ligam através do unity “socket.on('send message unity', async function (data) “ e a outra para o administrador que se liga através da web “socket.on('send message', async function (data)”.

A nível do administrador, este pode-se autenticar introduzindo na página de login, a sua organização, username e password, que serão utilizados como argumentos na função “authenticateDNAdmin”.

Login como Administrador

Organização

Username

Password

Figura 8 - Login com Administrador

```
async function authenticateDNAdmin(user, pass) {  
  
  await client.bind(user, pass, controls: function (err) {  
    if (err) {  
      console.log("Error in new connection" + err);  
      adminOuPacien = 5;  
    } else {  
      console.log("\nBem vindo Administrador");  
      adminOuPacien = 0;  
    }  
  });  
}
```

Figura 9 - Função para autenticar o Administrador

Após a autenticação do administrador ser confirmada, este terá acesso à seguinte página como demonstrado na “figura 10”.

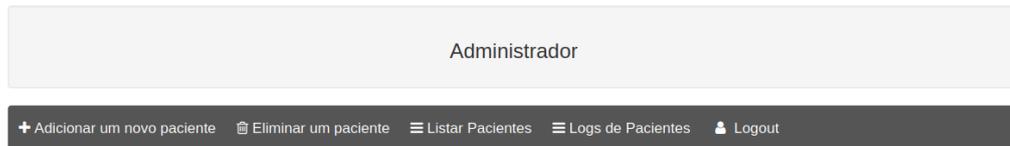


Figura 10 - Página de gestão do Administrador

Ao clicar no botão “Adicionar um novo paciente”, este mesmo administrador pode adicionar pacientes introduzindo assim a organização (que os pacientes vão pertencer), username, nome, email e password, utilizamos estas credenciais na função “addUser” que irá adicionar ao LDAP o respetivo paciente.

Registo

Username

Nome

Email

Password

Organização

Figura 11 - Página de Registo


```

async function deleteUser(nome, org) {
  await client.del( name: 'cn=' + nome + ',ou=paciente,ou=' + org + ',o=Lusofona', controls: async function (err) {
    if (err) {
      pacienteEliminado = "erro delete";
      console.log("err in delete user" + err);
      console.log("a: " + pacienteEliminado);
      return false;
    } else {
      pacienteEliminado = "successfully deleted";
      console.log("deleted user\n");
      console.log("b: " + pacienteEliminado);
      return true;
    }
  });
}

```

Figura 14 - função para eliminar um paciente

Caso o administrador queira verificar os pacientes inscritos no LDAP, poderá utilizar o botão “Listar pacientes”, que irá utilizar a função “searchUser” e será assim apresentado todos os utilizadores que se encontram no LDAP.

Listar Pacientes

```
{"dn":"cn=gustavo,ou=paciente,ou=Org2,o=Lusofona","controls":[],"cn":"gustavo"}
```

```
,{"dn":"cn=admin,ou=Administrador,ou=Org1,o=Lusofona","controls":[],"cn":"admin"}
```

```
,{"dn":"cn=antónio,ou=paciente,ou=Org1,o=Lusofona","controls":[],"cn":"antónio"}
```

```
,{"dn":"cn=andre,ou=paciente,ou=Org2,o=Lusofona","controls":[],"cn":"andre"}
```

```
,{"dn":"cn=Fabio,ou=paciente,ou=Org1,o=Lusofona","controls":[],"cn":"Fabio"}
```

```
,{"dn":"cn=tiago,ou=paciente,ou=Org2,o=Lusofona","controls":[],"cn":"tiago"}
```

```
,{"dn":"cn=lucas,ou=paciente,ou=Org1,o=Lusofona","controls":[],"cn":"lucas"}
```

Figura 15 - Página de listar Pacientes

```

async function searchUser() {

  var opts = {
    filter: '(cn=*)',
    scope: 'sub',
    attributes: ['cn']
  };

  await client.search( base: 'o=Lusofona', opts, controls: async function (err, res) {
    if (err) {
      console.log("Error in new connetion" + err)
    } else {
      res.on('searchEntry', function (entry) {
        console.log('entry: ' + JSON.stringify(entry.object));
        y.push(JSON.stringify(entry.object) + "///");
      });
      res.on('searchReference', function (referral) {
        console.log('referral: ' + referral.uris.join());
      });
      res.on('error', function (err) {
        console.error('error: ' + err.message);
      });
      res.on('end', function (result) {
        console.log('status: ' + result.status);
      });
    }
  });
}

```

Figura 16 - Função de Pesquisar Pacientes nas Organizações

Quando o administrador quiser observar o histórico dos pacientes que se conectaram ao jogo conseguirá observar esta mesma informação ao carregar no botão “Logs de Pacientes”. Esta informação será guardada num ficheiro local que será atualizado sempre que um utilizador se autentique no jogo, sendo esta informação guardada num ficheiro nunca será perdida mesmo que o servidor esteja online ou offline.



Figura 17 - Página de Log dos Pacientes

Para o paciente, existe apenas um ecrã de login com base em unity (C#) que envia para o servidor node.js através da biblioteca socket.io, o login tem três campos de input e um botão de confirmação, após o paciente inserir as suas credenciais estas são verificadas através da função `authenticateDNPaciente` que autentica com o LDAP este mesmo paciente, mostrando de seguida um pequeno texto que indica se a autenticação foi realizada com sucesso ("login com sucesso no Unity") ou sem sucesso ("login inválido no Unity"). Caso os campos de input estejam vazios é apresentada a mensagem seguinte "Campos Vazios".

Org1

lucas

*

Login

login com sucesso no Unity

Figura 18 - Login em unity para o paciente

```
async function authenticateDNPaciente(user, pass) {  
  await client.bind(user, pass, controls: async function (err) {  
    if (err) {  
      console.log("Error in new connection" + err);  
      adminOuPacien = 6;  
      loginUnity = "login inválido no Unity";  
    } else {  
      console.log("Success");  
      adminOuPacien = 1;  
      loginUnity = "login com sucesso no Unity";  
      var d = new Date();  
      var mes = parseInt( (d.getMonth()+1) );  
      var date = d.getHours() + ":" + d.getMinutes() + " - " + d.getDate() + "/" + mes.toString() + "/" + d.getFullYear();  
      logs.push(date + " - " + user + "////");  
      orig = orig + date + " - " + user + "////";  
      fs.writeFile( path: 'logs.txt', orig, callback: function (err : ErrnoException | null ) {  
        if (err){  
          return console.log(err);  
        }  
        console.log(logs.toString());  
      });  
    }  
  });  
}
```

Figura 19 - Função de autenticação do paciente

9. Benchmarking

LDAP vs RDBMS

O LDAP é um protocolo usado especificamente para consultar e modificar dados. Tudo isto é realizado usando serviços de diretório (sistema de software que armazena, organiza e fornece acesso às informações que estão em um diretório) com uma ligação TCP / IP. A principal função de qualquer diretório é atuar como um conjunto de objetos com atributos organizados hierarquicamente.

Uma base de dados é simplesmente uma coleção de dados que tem um ou mais usos. O modelo de base de dados mais comum é o do modelo relacional (que é baseado na lógica de predicado de primeira ordem).

Tendo em conta que no projeto é necessário controlar o acesso e a gestão dos utilizadores de diferentes organizações, surge o LDAP, que fornece um método padronizado de acesso a dados remotos e locais através do qual os dados podem ser movidos para múltiplas localizações sem afetar qualquer acesso externo desses dados. Usando métodos de referência os dados do LDAP podem ser movidos para servidores alternativos LDAP alterando apenas os parâmetros operacionais.

Assim sendo, com a utilização do LDAP é possível construir sistemas distribuídos, utilizando dados provenientes de organizações autónomas separadas ao mesmo tempo, que fornece uma visão única e consistente dos dados de todos os utilizadores.

Tendo em conta o que explicamos desde o início do desenvolvimento do projeto continuamos a pensar que a utilização do LDAP seria mais benéfica para este tipo de projeto, uma vez que nos dá uma maior perceção do nível ou do tamanho das organizações e a sua estrutura. Com isto em mente continuamos o desenvolvimento do projeto através do LDAP.

10.Método e planeamento

No desenvolvimento do projeto e durante a conceção dos calendários, estes foram feitos e pensados já a contar com a pesquisa que teria de ser feita, uma vez que todo este projeto se baseia em tecnologias nunca ou muito pouco utilizadas por nós, algumas como o LDAP, C# e Unity que foi a primeira vez que utilizamos uma vez que não foram lecionadas no curso de Engenharia Informática.

Em relação as maiores complicações encontradas no desenvolvimento foram descobertas quando tentávamos encontrar uma biblioteca que melhor satisfizesse as nossas necessidades relativas aos requisitos do projeto uma vez que existem várias bibliotecas de LDAP disponíveis para node.js com funcionalidades muitos similares.

Com a sugestão do orientador decidimos desenvolver um sistema de autenticação já integrado com base em unity uma vez que já é o motor utilizado na BSL. Estando esta funcionalidade fora dos nossos objetivos iniciais levou a um pequeno atraso uma vez que nos levou a pesquisar e aprender mais sobre estas novas tecnologias.

Em geral correu tudo de acordo com o planeado nos calendários apresentados nas entregas anteriores uma vez que estes foram concebidos para termos um certo tempo/margem de manobra em caso de encontrar erros bloqueantes que necessitassem de uma pesquisa mais intensa.

11.Resultados

Embora grande parte dos requisitos tenham sido implementados ou parcialmente implementados, encontramos algumas adversidades que nos impossibilitou de implementar de forma completa todos os requisitos iniciais, sendo estes referidos os requisitos UC-10 e UC-11. Nesta solução foi deixado um bocado de parte durante o desenvolvimento uma vez que seria uma situação rara (os utilizadores estarem autenticados em dois computadores/organizações diferentes ao mesmo tempo) uma vez que quando os pacientes estão a jogar o jogo estão a ser monitorizados por especialistas.

Encontramos alguns requisitos que conseguimos desenvolver apenas parcialmente, sendo estes os requisitos: UC-05, UC-06 e UC-09. Sendo todos estes relativos às limitações dos administradores. Não sendo requisitos que interferissem demasiado no funcionamento da autenticação em LDAP.

Em relação aos requisitos completos e completamente implementados, temos os seguintes: UC-01, UC-02, UC-03, UC-04, UC-07, UC-08, UC-12, UC-13, UC-14, UC-15 e UC-16, que são também os mais necessários para o funcionamento pedido desde o início da apresentação do TFC.

12. Conclusão e trabalhos futuros

Em relação à hipótese de a continuação deste trabalho se tornar num outro TFC, achamos que seria interessante ter um maior foco no lado de segurança e na integração de uma base de dados que permitisse guardar ainda mais informação. No entanto achamos que como um TFC, deveria ter requisitos mais específicos e mais complexos, com maior feedback do departamento de psicologia.

Concluindo, este projeto foi uma experiência única porque tivemos de utilizar várias tecnologias nunca antes trabalhadas por nós, o que levou à necessidade de uma vasta pesquisa para conseguirmos interligar as diversas tecnologias, o que se provou um desafio em certas alturas. No entanto, pensamos que através do desenvolvimento deste trabalho podemos adquirir novas competências que muito provavelmente nos poderão ajudar na continuação dos nossos estudos e também ao longo das nossas carreiras.

13. Bibliografia

- 1] Fundação OpenLDAP, “openldap.org,” Fundação OpenLDAP, [Online]. Available: <http://www.openldap.org/doc/admin24/intro.html>. [Acedido em 06 Dezembro 2019].
- 2] Linuxtopia, “Linuxtopia.org,” [Online]. Available: https://www.linuxtopia.org/online_books/network_administration_guides/ldap_administration/intro_LDAP_vs_RDBMS.html. [Acedido em 12 12 2019].
- 3] Zytrax, “Zytrax.com,” [Online]. Available: <https://www.zytrax.com/books/ldap/ch2/>. [Acedido em 12 12 2019].
- 4] S. P. R. Carvalho, “Bateria Sistemica de Lisboa,” [Online]. Available: <http://recil.grupolusofona.pt/jspui/handle/10437/9476>. [Acedido em 10 12 2019].
- 5] Oracle, “docs.oracle.com,” [Online]. Available: <https://docs.oracle.com/cd/E19182-01/820-7855/6nj391me2/index.html>. [Acedido em 15 Março 2020].
- 6] A. A. Directory, “directory.apache.org,” Apache, [Online]. Available: <https://directory.apache.org/api/user-guide/6.12-entry.html>. [Acedido em 22 Março 2020].
- 7] A. D. Directory, “directory.apache.org,” Basic authorization, [Online]. Available: <https://docs.oracle.com/cd/E19182-01/820-7855/6nj391me2/index.html>.
- 8] P. M. Martin Löfberg, “Web vs. Standalone Application,” SE – 372 25 Ronneby Sweden, 2005.
- 9] Joyent, “LDAPJS Documentation,” [Online]. Available: <http://ldapjs.org/>. [Acedido em 10 Junho 2020].
- 10] “Node.js v14.5.0 Documentation,” [Online]. Available: <https://nodejs.org/api/http.html>. [Acedido em 15 Junho 2020].
- 11] “Socket.io,” [Online]. Available: <https://socket.io/docs/>. [Acedido em 15 Junho 2020].

14.Anexos

Em anexo enviamos:

- Um projeto em IntelliJ (TFC_servidor_html), que corresponde aos ficheiros node.js e HTML (servidor e web para administrador);
- Um projeto em Unity (UnityNode_LDAP), que corresponde ao ficheiro C# e à scene de Unity (Autenticação do utilizador).

15. Manual técnico da aplicação

O projeto é constituído por oito ficheiros desenvolvidos por nós, sendo estes:

- Um node.js (teste.js) que atua como servidor e ligação ao LDAP, possui todas as funções de autenticação (authenticateDNAdmin e authenticateDNPaciente), inserção (addUser), eliminação (deleteUser), pesquisa (searchUser) e histórico de pacientes (implementado dentro da função authenticateDNPaciente).
- Seis ficheiros HTML que são apenas direcionados ao administrador, cada ficheiro mostra um ecrã com uma funcionalidade diferente que apenas podem ser acedidos após o administrador se ter autenticado. O ficheiro index.html possui o login com três campos de input (respetivamente organização, username e password), o adminPage.html que possui uma navbar com botões que permite aceder as restantes funcionalidades desenvolvidas, o registo.html com cinco campos de input (respetivamente username, nome, email, password e organização) que permitem fazer o registo, o eliminarPaciente.html com apenas dois campos de input (username e organização) permite apagar um paciente do LDAP, o listarPacientes.html retorna uma lista de pacientes que pertencem às organizações e o logPacientes.html retorna uma lista de autenticações dos pacientes bem como as horas e dias em que ocorreram as mesmas, os dados dos logs dos pacientes são guardados em ficheiro para se ocorrer algum problema no servidor os dados estarem sempre guardados. Nos ecrãs onde necessitam de input os campos são concatenados numa string e enviados para o servidor onde apenas é necessário dar split e retornar o resultado para o html,
- Um ficheiro em C# no qual criamos um login que recebe três inputs (organização, username e password), estes valores são enviados para o servidor através de JSON. Para fazer a comunicação com o servidor utilizamos a classe SocketIOComponent que nos permitiu fazer a ligação cliente-servidor. Enviamos através do servidor o resultado da autenticação e apresentamos o resultado no ecrã do jogo, o valor retornado do servidor poderá ser utilizado para verificar se a autenticação foi validada e se for avançar para o jogo.

Para a execução do projeto primeiramente será necessário ter o servidor LDAP a correr, de seguida, terá de ser inicializado o servidor em node.js para que este consiga receber os pedidos quer sejam estes autenticação, registo, remoção ou pesquisa. Todos os ficheiros HTML e o ficheiro logs.txt necessitam de estar na mesma diretoria que este ficheiro em node.js.

Para o sistema de autenticação dos pacientes ser funcional o nosso projeto em unity tem que ser associado ao projeto já existente do jogo.

16.Manual de utilizador da aplicação

Para os administradores: Terão de aceder ao servidor em node.js em que poderão fazer a sua autenticação e onde terão depois acesso a todas as funcionalidades exclusivas. Estas funcionalidades são:

- Adicionar Pacientes, onde após clicar nesse botão terá acesso a um ecrã onde poderá fazer um registo de um novo paciente precisando então de fornecer alguns valores (organização, username, nome, email e password).

- Eliminar Pacientes, onde após clicar nesse botão terá acesso a um ecrã onde poderá fazer a eliminação de um paciente precisando então de fornecer alguns valores (organização, username).

- Pesquisar Pacientes, onde após clicar nesse botão terá acesso a um ecrã onde lhe será mostrado todos os utilizadores existentes no LDAP.

- Mostrar logs dos Pacientes, onde após clicar nesse botão terá acesso a um ecrã onde lhe será mostrado todos os logins e as suas respectivas datas efetuados por todos os pacientes.

Para os pacientes: Ao entrarem no jogo da BSL serão apresentados com um pequeno formulário onde lhes será pedido a que organização pertencem, o seu username e a sua password, após esta autenticação ser verificada com o LDAP o paciente terá acesso ao resto dos conteúdos do jogo.