

Licenciatura em Engenharia Informática

LAPR5- Janeiro de 2012

Documentação

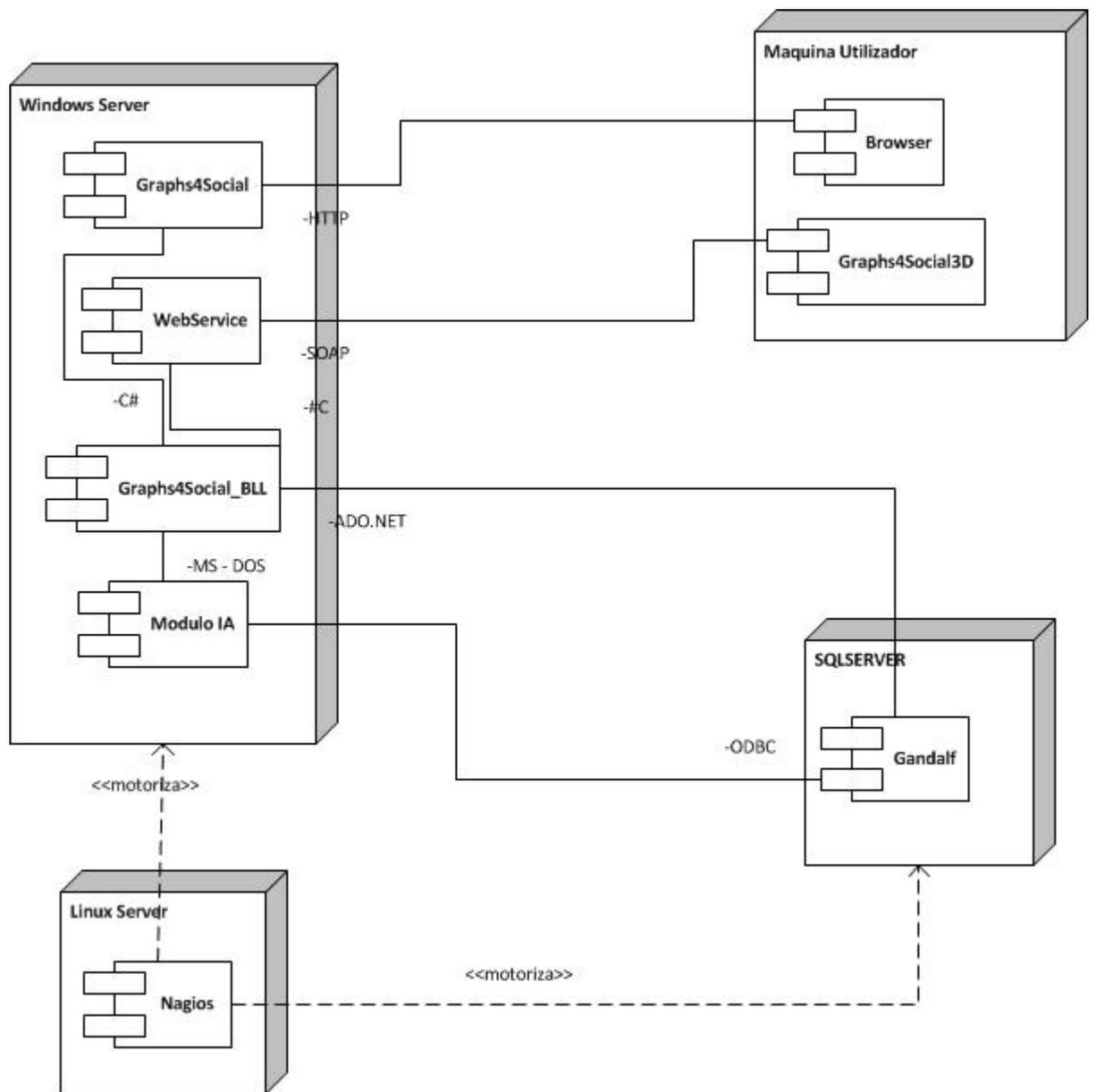
Grupo Nº: 8bitsParadox		
3DH	1100556	Bruno Ferreira
3DH	1100578	Fábio Santos
3DH	1100579	Fernando Afonso
3DH	1100594	Joana Osório
3DH	1100617	José Ribeiro
3DH	1100636	Marco Rodrigues

Índice:

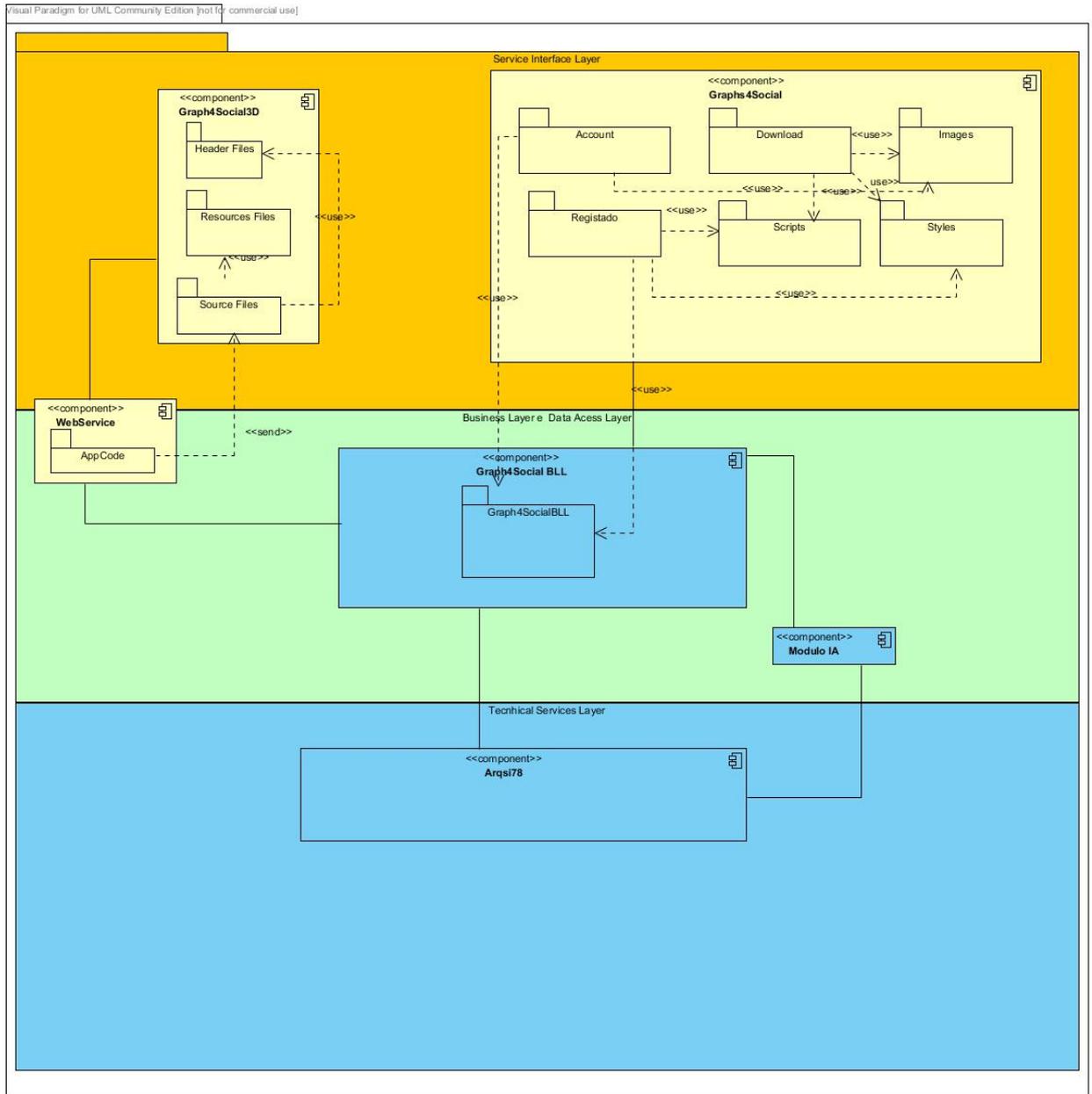
1. Arquitetura do sistema	
a. Deployment View.....	2
b. Logical View.....	3
c. Breve descrição dos componentes.....	4
d. Especificações das pré e pós condições das interfaces.....	19
i. Web service.....	19
e. Principais estruturas de dados utilizadas.....	23
2. Modelo de Dados.....	26
3. Modelo de Domínio.....	26
4. Testes.....	27
a. Grafo de testes utilizado.....	27
b. Descrição de testes unitários e outros testes efetuados.....	28
i. Testes unitários feitos à camada de negócio.....	28
ii. Projecto main de testes feitos ao IAManger.....	28
c. Planos de testes.....	29
i. Módulo de Inteligência Artificial.....	29
ii. Módulo de navegação e visualização 3D.....	31
iii. Módulo Site.....	40
iv. Módulo de Infraestrutura computacional.....	45
5. Divisão de tarefas e breve descrição do desenvolvimento do projeto utilizado pelo grupo.....	47
a. Pequena explicação da divisão das tarefas.....	47
b. Pequena explicação do desenvolvimento do projeto.....	47

1. Arquitetura do sistema:

a. Deployment View:



b. Logical View:



c. Breve descrição dos componentes:

Existem quatro estações de trabalho principais, são elas: o WindowsServer, a máquina do utilizador, o SQLServer e o LinuxServer.

O (1) *Graphs4Social*, o (2) *WebService*, a (3) *Graphs4Social.BLL* e o (4) *módulo IA*, encontram-se alojados na máquina Windows e dizem respeito ao site da rede social, ao webservice de ligação da BLL ao módulo 3D, à camada de BLL (Business Logic Layer) e DAL (Data Access Layer) e ao módulo de Inteligência Artificial, respetivamente:

(1) O site da rede social possui os seguintes principais packages:

- Account: responsável pelas páginas de registo/login do utilizador;
- Download: que contém o ficheiro.zip para download da rede social;
- Registado: responsável pelas páginas do utilizador registado;
- Scripts: que contém os javascripts necessário ao website;
- Styles: que contém as folhas de estilo utilizadas no site;
- Possui ainda páginas de controlo de erros da página, informação geral para utilizadores não registados e páginas com as tags clouds das relações e dos utilizadores.
- É possível também importar ligações, a partir do LinkedIn e do Facebook.

(2) O web service, encontra-se responsável por fazer a conexão entre o cliente C++ e a camada de BLL e DAL, que procura simultaneamente informação proveniente de pesquisas à base de dados ou de consultas ao módulo de Inteligência Artificial, as especificações da interface deste serviço podem ser detalhadamente visualizadas na secção do documento para o efeito;

(3) A BLL, tem as principais classes de negócio presentes na logical view e ainda a classe responsável pela implementação do padrão ActiveRecord, a classe responsável pelas configurações à base de dados e ainda a classe de ligação ao módulo de Inteligência Artificial.

(4) O módulo IA está representado por um único ficheiro .pl onde contem predicados desenvolvidos em prolog(db_conn.pl). O ficheiro está incorporado na BLL desenvolvida e é responsável por algumas pesquisas. Este ficheiro está diretamente conectado a base de dados e usa essa informação para desenvolver as suas respostas.

Para a utilização dos predicados construídos na BLL foi criado um conjunto de métodos para o uso dos predicados e tratamento da informação devolvida que se encontram na classe IAManager.cs.

Para o uso dos predicados foram criados 3 métodos principais (prolog_chamada, prolog_chamada_1 e prolog_chamada_2) cada um adaptado ao número de argumentos enviados por parâmetro.

Estes possuem a seguinte forma genérica:

```
public static List<Object> prolog_chamada_1(String PastaWinProlog, String metodo,
String id, String nomeFicheiro)
{
    prolog_consulta1(PastaWinProlog, metodo, id);
    //para permitir retornar qq objeto seja utilizador, relacao, caminhos, etc
    List<Object> dadosLidos = LerFicheiro(metodo, nomeFicheiro);
    return dadosLidos;
}
```

Cada um destes métodos possui ainda um outro método auxiliar(prolog_consulta, prolog_consulta1 e prolog_consulta2, respetivamente) que possui a seguinte forma genérica:

```
public static void prolog_consulta1(String PastaWinProlog, String metodo, String id)
{
    try
    {
        var pProcess = new System.Diagnostics.Process
        {
            StartInfo =
            {
                FileName = "cmd.exe",
                Arguments = "/C PRO386W.EXE /V1 consult('db_conn.pl'). init_bd_conn. " + metodo +
                "(" + id + "). end_bd_conn.",
                UseShellExecute = false,
                RedirectStandardOutput = true,
                WorkingDirectory = PastaWinProlog
            }
        };
        pProcess.Start();
        pProcess.WaitForExit();
        pProcess.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
        Console.ReadLine();
    }
}
```

A chamada de um destes métodos invoca o cmd.exe (linha de comandos – (MS-DOS)) e corre o ficheiro prolog por linha de comandos. Para tal ser possível é necessário ter o programa(WIN PROLOG 4800) e o ficheiro .pl num diretório parametrizável. Após a execução do predicado é necessário ler a informação produzida. A solução encontrada para a integração entre projetos (C# e Win-Prolog) foi, apos a execução de qualquer pesquisa, o valor da resposta ser guardado num ficheiro .txt.

Para obter essa informação foi criado um método, que se segue:

```
//-----Leitura do ficheiro
public static List<Object> LerFicheiro( string metodo, string nomeFicheiro)
{
    List<Object> dadosLidos = new List<Object>();
    try
    {
        System.IO.StreamReader arquivo = new System.IO.StreamReader(@nomeFicheiro);
        string linha = "";
        while (true)
        {
            linha = arquivo.ReadLine();
            if (linha != null)
            {
                //metodos que retornam utilizadores
                if (metodo == "getSuggestions" || metodo == "getCommonFriends" || metodo ==
                    "getSmallNetwork" || metodo == "getFriendsByTag" || metodo == "getMaven" || metodo
                    == "getCMinimo" || metodo == "getCForte" || metodo == "getEco" || metodo ==
                    "getMaven" || metodo == "getMedia")
                    dadosLidos = TrataInformacaoListaUtilizadores(dadosLidos, linha);
                //metodo das estatisticas que retornam ints?
            }
            else
                break;
        }
        arquivo.Close();
        File.Delete(nomeFicheiro);
        return dadosLidos;
    }
    catch (System.IO.FileNotFoundException)
    {
        Console.WriteLine("Não foi encontrado o ficheiro output de ligação ao WINPROLOG!");
        return null;
    }
}
```

O método está preparado para tratar a informação devolvida das pesquisas que se encontra, na maior parte do tempo, num formato "[id1,id2,...,idx]" em que cada id representa o id de um utilizador. A exceção a regra será o grau médio de separação da rede que devolve apenas um valor float e o método para auxílio a navegação timeline do módulo 3D, que retorna apenas string. Após a leitura do ficheiro txt este é apagado.

A conexão do ficheiro de pesquisas com a base de dados é realizada pelos seguintes predicados:

```
:- ensure_loaded( system(dblink) ).
:- dynamic datasource/1.
init_bd_conn:-
db_connect( 'gandalf;Uid=ARQSI78;Pwd=78grupo;MARS_Connection=yes;', HDBC ),
(...)
end_bd_conn:-
retract( datasource( HDBC ) ),
db_disconnect( HDBC ).
```

O predicado "init_bd_conn" tem que ser executado antes de qualquer predicado para que estes possam aceder ao conteúdo da mesma. Ainda neste predicado são mapeadas as tabelas da base de dados para possibilitar a sua consulta. Após a utilização é executado o "end_bd_conn" para fechar a conexão.

As principais pesquisas realizadas foram:

- Obter a rede até ao 3º nível (`getSmallNetwork\1`);
 - Obter os amigos em comum entre dois utilizadores (`getCommonFriends\2`);
 - Obter sugestões de amigos (`getSuggestions\1`);
 - Obter amigos por tag (`getFriendsByTag\2`);
 - Obter o maven de uma tag (`getMaven\1`);
 - Obter o caminho mínimo entre dois utilizadores (`getCMinimo\2`);
 - Obter o caminho forte entre dois utilizadores (`getCForte\2`);
 - Obter o grau médio de separação da rede (`getMedia\0`);
 - Obter o grau médio de separação entre dois utilizadores (`getMedia2U\2`);
 - Obter o eco de uma mensagem (`getEco\1`).
-
- `getSmallNetwork`: Recebe 1 id e retorna uma lista de utilizadores do tipo [idx, idy, ..., idn], que são os amigos e amigos de amigos do utilizador até ao 3º nível;
 - `getCommonFriends`: Recebe 2 ids e retorna uma lista de utilizadores do tipo [idx, idy, ..., idn], que são os os amigos em comum entre dois utilizadores;
 - `getSuggestions`: Recebe 1 id e retorna uma lista de utilizadores do tipo [idx, idy, ..., idn], que são sugestões de amigos, ou seja, amigos de amigos que tenham pelo menos uma tag em comum. As tags passam por uma análise de semântica, ou seja, as tags comuns podem não ser idênticas mas terem o mesmo valor semântica.
 - `getFriendsByTag`: Recebe 1 id e uma lista de tags do tipo [idx, idy, ..., idn] e retorna os amigos que tenham uma, ou mais, tags em comum. Como no predicado anterior, as tags são processadas pelo seu valor semântico.
 - `GetMaven`: Dada uma tag, retorna o ou os mavens dessa tag. Para ser maven de uma tag o utilizador necessita de ter 60% (ou mais) das relações entre os utilizadores que partilham essa tag e é necessário que a tag tenha mais do que 5 utilizadores que a usem. Como anteriormente, a tag é processado pelo valor semântico.
 - `GetCMinimo`: Recebe o id do utilizador origem e do utilizador destino e retorna o caminho mínimo (menos saltos) entre dois utilizadores. Para este cálculo foi utilizada uma visita em largura.
 - `GetCForte`: Recebe o id do utilizador origem e do utilizador destino e retorna o caminho forte (com mais força média) entre dois utilizadores. Para este cálculo foi utilizado o método Branch and Bound adaptado às necessidades específicas da rede social. A adaptação feita foi para em vez de dar o caminho mais fraco, dar o mais forte (após todas as soluções encontradas estas são ordenadas e


```

System.IO.File.WriteAllLines(@"\" + Properties.Settings.Default["Location"].ToString() +
"test_relacao.pl", test_relacao);
System.IO.File.WriteAllLines(@"\" + Properties.Settings.Default["Location"].ToString() +
"test_tagrelacao.pl", test_tagrelacao);
}
}
}

```

Este código foi compilado e permite que, já como executável, seja alterado o número de utilizadores/relações geradas tal como o destino desses ficheiros. Após a análise concluiu-se que se teria que analisar uma amostragem do grafo caso houvesse mais do que 20 utilizadores. Essa amostragem seria a seleção de 20 utilizadores para análise. A média é calculada fazendo uma seleção aleatória de utilizadores e encontrando a média do grau de separação entre si, semelhante ao predicado `getMedia2U\2`.

O (5) *Graph4Social3D* e o browser encontram-se alojados na máquina do utilizador, sendo que representam o módulo de navegação e visualização 3D e o browser do utilizador, respetivamente.

O módulo de navegação e visualização 3D tem os seguintes principais packages:

- **Header Files:** contém todos os *headers files* utilizados, entre eles destacam-se: *AplicarTexturas.h* (responsável pela configuração e manipulação das texturas, materiais e luzes), *EstruturasDados.h* (responsável pela configuração e manipulação das principais estruturas de dados utilizadas), *fmodManager.h* (responsável pela configuração da música e efeitos sonoros no projecto), *grafos.h* (responsável pela manipulação e criação do grafo), *SkyDome.h* (responsável pela manipulação do SkyDome) e *WebService.h* (responsável pela manipulação dos principais métodos de ligação ao webservice, representa o cabeçalho do cliente C++ utilizado) e *mdlviewer.h* (representa o cabeçalho dos métodos utilizados para o carregamento dos avatares).
- **Resource Files:** contém os *resources files* utilizados para a ligação ao webservice e para a leitura de imagens;
- **Sources Files:** contém todos os *source files*, entre eles destacam-se: *aplicarTexturas.cpp* (contém o corpo dos métodos utilizados para a manipulação das luzes, materiais e texturas), *EstruturasDados.cpp* (contém o corpo dos métodos utilizados para a manipulação das principais estruturas de dados utilizadas), *fmodManager.cpp* (contém o corpo dos métodos utilizados para configuração da música e efeitos sonoros), *SkyDome* (contém o corpo dos métodos utilizados para o SkyDome), ***SocialGraph3D.cpp*** (ficheiro main do projecto, contém as principais funcionalidades de arranque: menu, display, controlos de câmara, rato, teclado, etc) e *WebService.cpp* (contém o corpo dos métodos utilizados para a ligação ao WebService, é o cliente C++ do web service), *mdlviewer.cpp* (representa o corpo dos métodos utilizados para o carregamento dos avatares)

Foram utilizadas diversas bibliotecas para o suporte das funcionalidades, entre as quais se destacam: *fmod* (para o carregamento de músicas e efeitos sonoros), *corona* (para a exportação de imagens png), *glTGA* (para o carregamento de texturas do billboard) e *libjpeg* (para o

carregamento de textura do céu, chão e menu de login). Foram ainda utilizadas outras bibliotecas de carácter geral, como é o caso da mathlib, do glut e do glaux.

Para ligação do cliente c++, foram utilizadas as ferramentas: SvcUtil e WSUtil, para o qual foi criado o respetivo .bat de compilação, estes ficheiros encontram-se disponíveis na pasta Graphs4Social/ComunicaModulo3D, tendo que ser adaptados à máquina local em que estão a ser executados por não possuírem caminhos relativos.

No protótipo da aplicação desktop de navegação e visualização 3D foram implementadas as seguintes funcionalidades:

- Visualização geral de toda rede social de um utilizador: após ser efetuado o processo de login o utilizador consegue visualizar toda a sua rede até ao 3º nível, caso seja um utilizador com conta normal ou toda a rede social, no caso de ser um utilizador Premium. Para uma visualização mais clara do grafo de um utilizador, foi criado um pequeno algoritmo de posicionamento no grafo, com base no grau das relações de um determinado utilizador. O utilizador que efetua o log in encontra-se desenhado com o seu nó desenhado a vermelho.

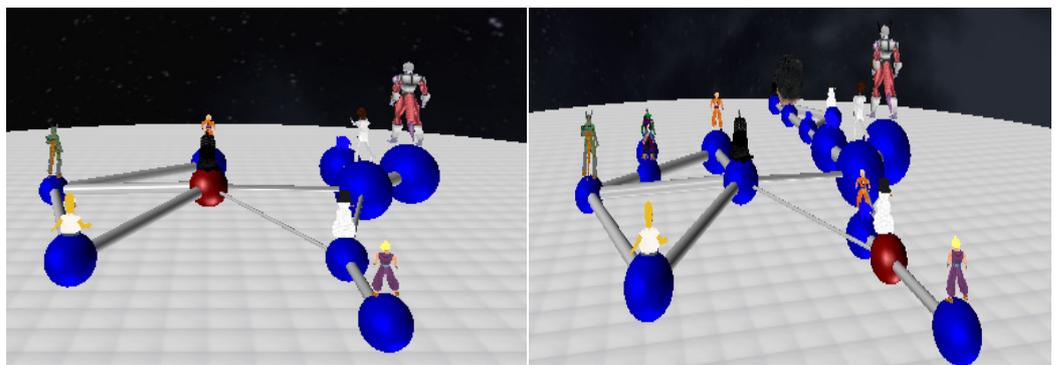


Figura 1 - Rede Social do Utilizador Afonso Henriques Figura 2 - Rede Social do Utilizador Rainha (Premium)

- Mini-mapa de todo o grafo da rede social: o mini-mapa acompanha o movimento da câmara, ou seja, do utilizador, este encontra-se representado por um triângulo azul.

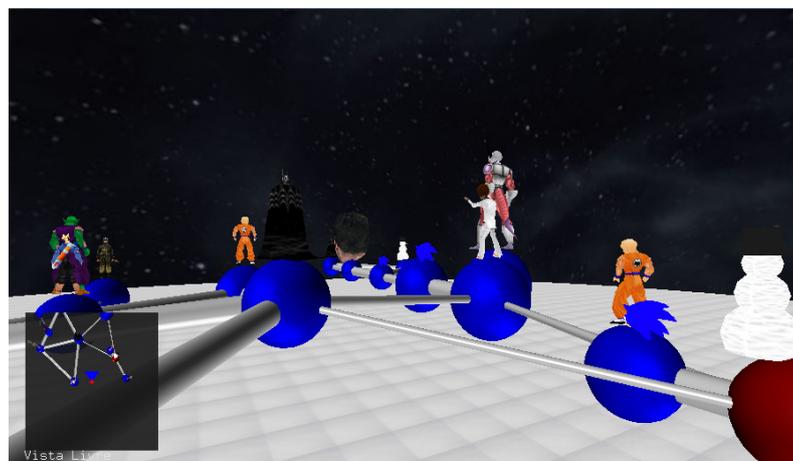


Figura 3 - Visualização do mini mapa em vista livre

- Exportação da visualização do grafo para um ficheiro num formato raster: a exportação da visualização do grafo é feita apenas para o formato raster (.png), através da leitura de

uma matriz de pixels, no entanto a cena do mundo não consegue ser corretamente capturada, aparecendo os avatares, nós e respetivos ramos a branco. Esta funcionalidade é despoletada quando é clicada a tecla F12, sendo que é mostrada uma mensagem de sucesso/insucesso da exportação da imagem e esta é guardada na pasta exportações do projeto.

- Cálculo da dimensão do nó: o cálculo da dimensão de nó depende do número de tags de cada utilizador.
- Cálculo da espessura dos arcos que ligam os nós: o cálculo da espessura do elemento de ligação que liga os respetivos nós, depende da força de ligação.

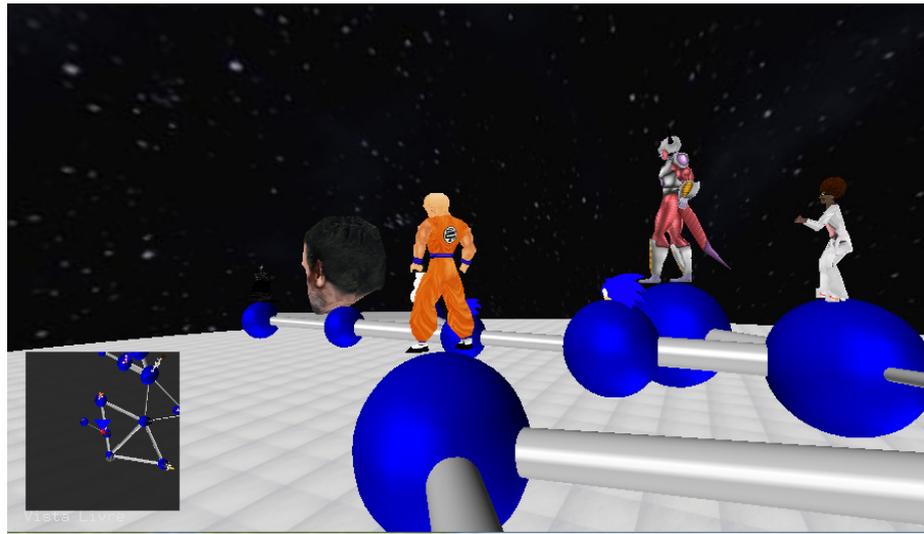


Figura 4 - Visualização diferenciada dos nós e espessura dos arcos

- Sistema de login diferenciado: O sistema de login utiliza uma imagem (.bmp), que foi criada pelo grupo no photoshop, sendo que a textura foi personalizada de acordo com os logotipos da "empresa". Esta textura encontra-se na pasta das texturas do projeto. O menu de login funciona quer por sistema de picking nas caixas, quer através do premir da tecla enter. Foi criado um sistema de log diferenciado para o utilizador Premium e o utilizador normal, tal como pode ser comprovado nas imagens seguintes:



Figura 5 - Login Utilizador Normal

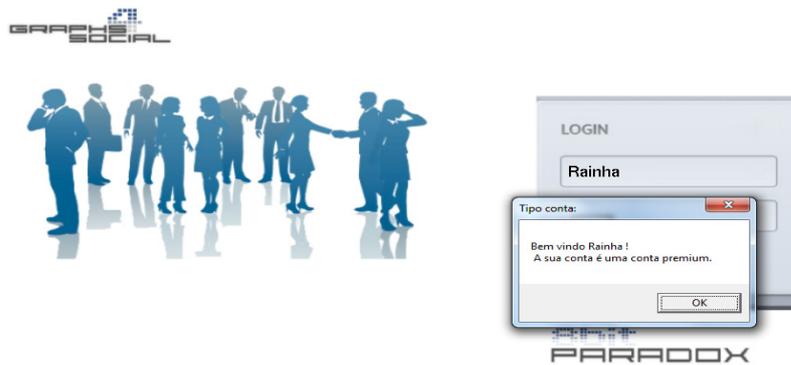


Figura 6 - Login de um utilizador Premium

- Visualização do caminho de ligação entre o utilizador autenticado e outro utilizador: pode ser feitas duas visualizações entre o caminho do utilizador autenticado e o outro utilizador, sendo essas visualizações respetivamente o caminho mínimo e o caminho forte, esta visualização é feita, através do picking nos nós e ativadas pelas opções de consola indicada na informação de menu.

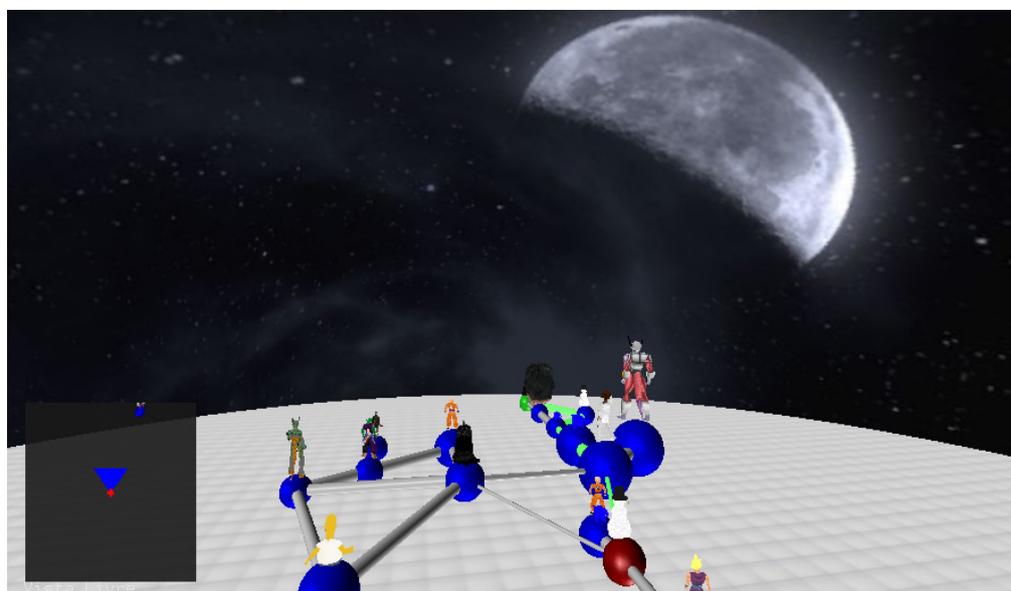


Figura 7 - Caminho forte entre a Rainha e o Dinis

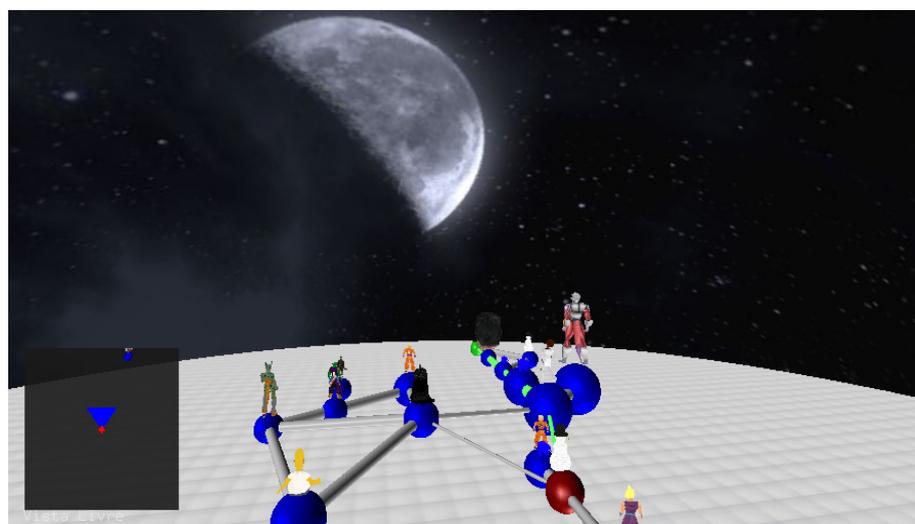


Figura 8 - Caminho mais curto entre a Rainha e o Dinis

- Animação automática: a animação automática do caminho permite que o utilizador veja em voo rasante os caminhos indicados acima.
- Vista livre: representa o efeito navegação no espaço, em que o grafo representa uma estrutura sólida e a câmara se pode movimentar livremente, o efeito já pode ser comprovado pelas figuras acima anexadas.
- Voo rasante: quando é aplicado um efeito de atração a um nó, a câmara passa a ser para um modo de voo rasante, que permite seguir a estrutura do grafo como se fosse um rede viária, quando o utilizador sobe, o voo rasante é desativado.

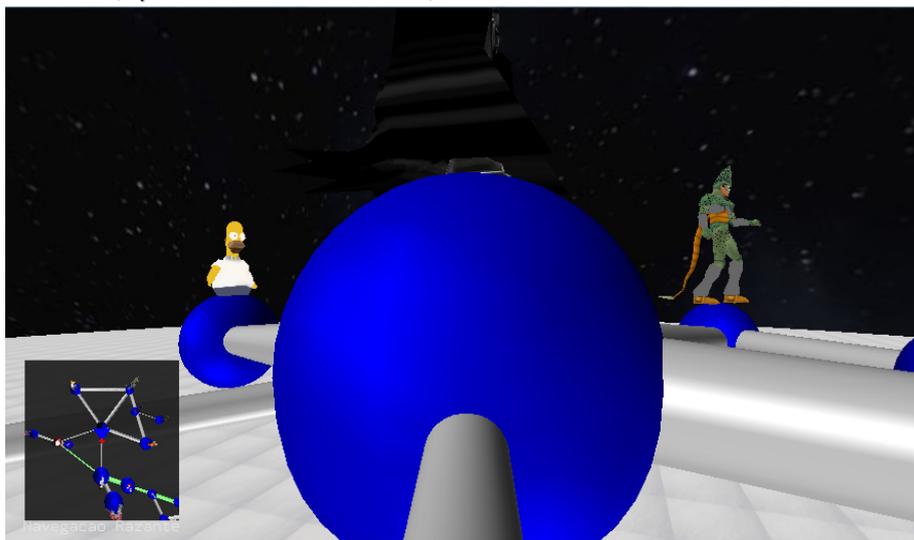


Figura 9 - Voo rasante

- Controlo de colisões e fontes de iluminação: É feito o controlo de colisões e fontes de iluminação para que o utilizador saiba em que posições que se encontra. No entanto, para permitir efeito de atração, sempre que um utilizador choca com um nó, sobe para posição desse utilizador.
- Tip flutuante/Indicação visual do "estado de humor" de um utilizador: Sempre que um nó é clicado, é despoletada uma tip flutuante com o nome do utilizador e o seu estado de humor, sendo que no estado de humor é utilizado um mecanismo de billboard. As texturas dos estados de utilizadores encontram-se na pasta de texturas.

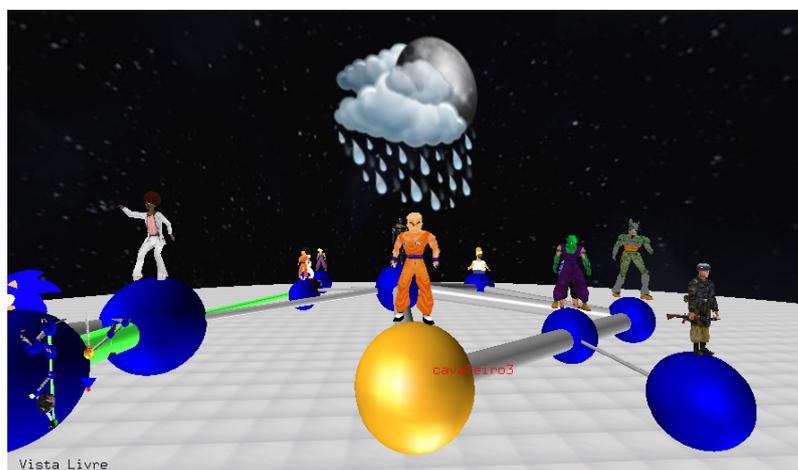


Figura 10- Tip do nome do utilizador e respetivo estado

- Efeitos especiais e música de fundo: Assim que o utilizador faz login é carregada uma playlist de músicas que se encontra na pasta músicas do projeto. Esta playlist pode ser controlada por consola, sendo permitidas opções para controlo do som, pause e para a escolha da música seguinte. A playlist é reproduzida automaticamente ao longo de toda a "viagem" do utilizador, voltando ao início quando é terminada. Quando um nó clicada, a música que toca na playlist é interrompida, sendo que são despoletados efeitos sonoros conforme o estado do utilizador, e.g: cantar dos pássaros, assim que o efeito sonoro acaba é carrega a próxima música da playlist. Os efeitos sonoros encontram-se na pasta "Sons" do projeto.
- Avatar: Como se pode ver ao longo das diversas imagens do módulo de navegação e visualização 3D, é permitido ao utilizador a escolha de um avatar, essa escolha pode ser feita no módulo de site e é mais tarde carregada a partir da base de dados.
- Carregamento de texturas para o céu e chão: A textura do céu é carregada através de skydome, esta é mudada consoante a hora do sistema, sendo que a partir das sete da tarde fica de noite. As textura do chão mudam consoante a data do sistema, de acordo com as estações do ano. Todas estas texturas se encontram na pasta de texturas do projeto.

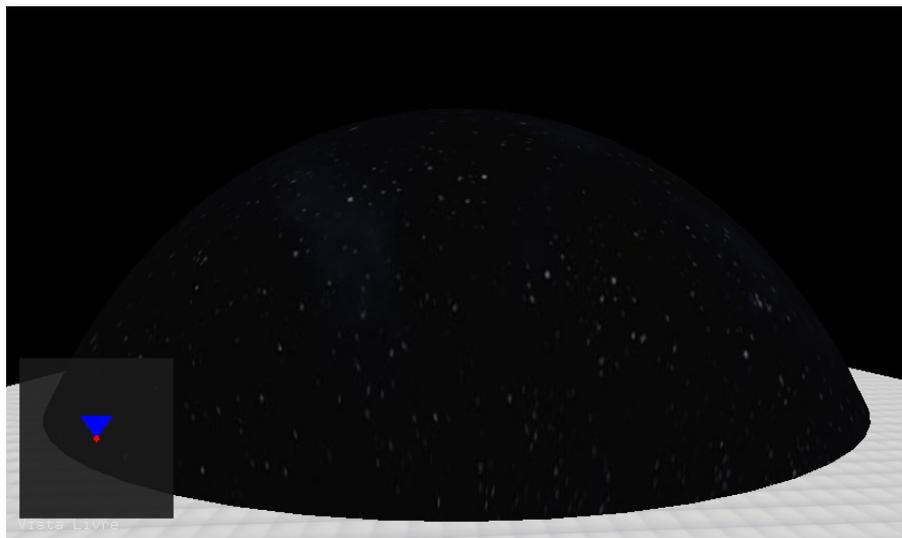


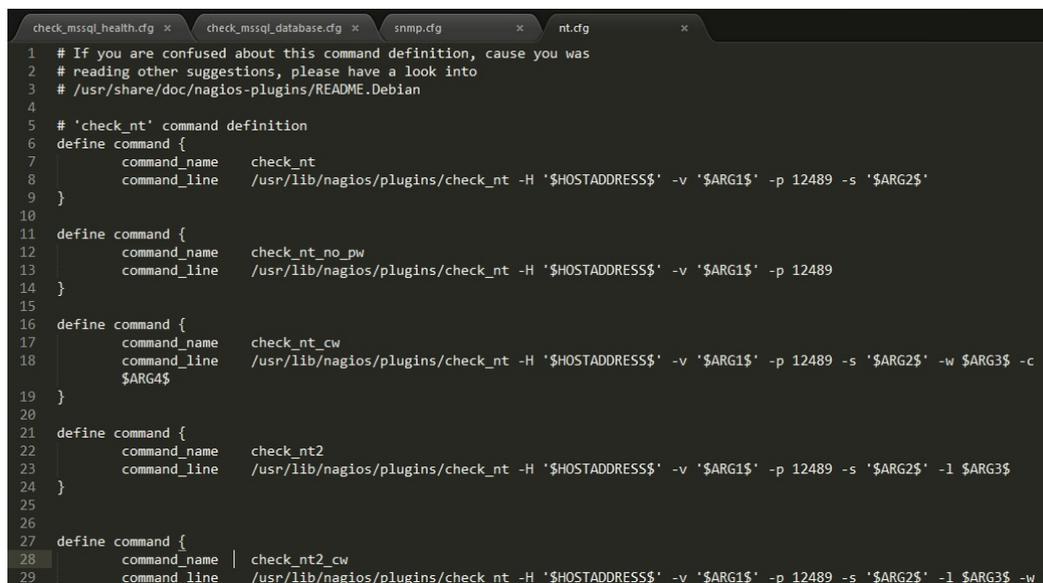
Figura 11 - SkyDome do céu

- Amigos em comum entre dois utilizadores: Através do sistema de picking e das opções de consola, podem ser indicados os amigos em comum entre dois utilizadores.
- Motor de partículas: são feitos alguns efeitos com o motor de partículas, sendo que é carregado o fogo para zangado, o fumo alegre e chuva triste.

O (6)nagios, responsável pela manutenção da infraestrutura computacional encontra-se alojado no LinuxServer e representa o software de monitorização da rede.

(6)Entre as várias pastas de configuração de ficheiros destacam-se as seguintes:

- nagios-pluggins: contém os ficheiros de plugins utilizados pelo nagios para monitorizar o servidor de dados e servidor web. Como principais plugins desenvolvidos ou modificados temos os ficheiros `check_mssql_health.cfg`,`check_mssql_database.cfg`,`snmp.cfg`,`nt.cfg`;



```
check_mssql_health.cfg x check_mssql_database.cfg x snmp.cfg x nt.cfg x
1 # If you are confused about this command definition, cause you was
2 # reading other suggestions, please have a look into
3 # /usr/share/doc/nagios-plugins/README.Debian
4
5 # 'check_nt' command definition
6 define command {
7     command_name    check_nt
8     command_line    /usr/lib/nagios/plugins/check_nt -H '$HOSTADDRESS$' -v '$ARG1$' -p 12489 -s '$ARG2$'
9 }
10
11 define command {
12     command_name    check_nt_no_pw
13     command_line    /usr/lib/nagios/plugins/check_nt -H '$HOSTADDRESS$' -v '$ARG1$' -p 12489
14 }
15
16 define command {
17     command_name    check_nt_cw
18     command_line    /usr/lib/nagios/plugins/check_nt -H '$HOSTADDRESS$' -v '$ARG1$' -p 12489 -s '$ARG2$' -w $ARG3$ -c
19     $ARG4$
20 }
21
22 define command {
23     command_name    check_nt2
24     command_line    /usr/lib/nagios/plugins/check_nt -H '$HOSTADDRESS$' -v '$ARG1$' -p 12489 -s '$ARG2$' -l $ARG3$
25 }
26
27 define command {
28     command_name    check_nt2_cw
29     command_line    /usr/lib/nagios/plugins/check_nt -H '$HOSTADDRESS$' -v '$ARG1$' -p 12489 -s '$ARG2$' -l $ARG3$ -w
```

Figura 12 - Print Screen dos principais scripts de pluggins utilizados

- Ficheiro tweet.php: plugin que permite enviar um tweet com a notificação do nagios



Figura 13 - Print Screen de notificação via twitter

- nagiosgraph: contém os ficheiros de configuração do plugin nagiosgraph, aplicação que permite construir gráficos do estado de monitorizações ao servidor de dados e do servidor web;

Nagiosgraph

Data for host [sw2626a](#), service [Port 25 Bandwidth Usage traffic](#)

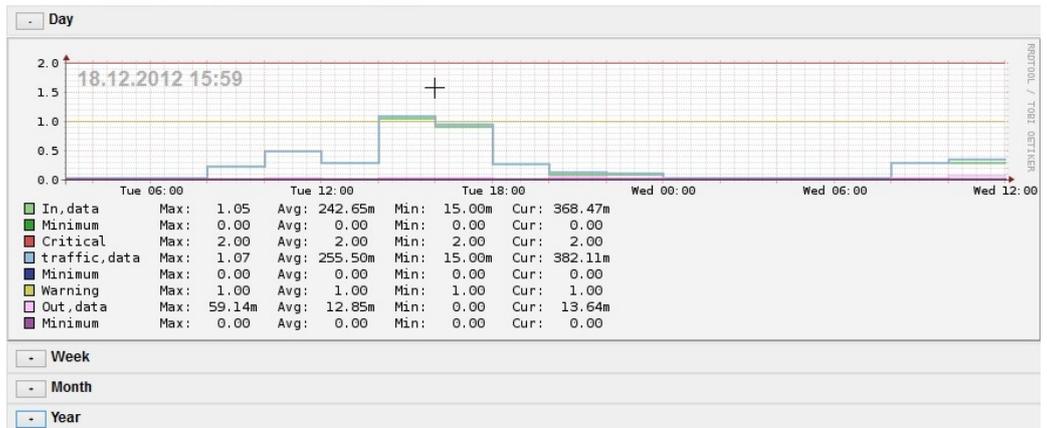


Figura 14 - Print Screen do gráfico do uso de largura de banda do computador “HP ProCurve Switch 2626 (J4900B)”

- nagios3: contém os ficheiros de configuração utilizados para monitorizar o servidor de dados e servidor web e os restantes “hosts” exigidos pelo enunciado;

Host	Status	Services	Actions
Portal Isep	UP	2 OK	[Icons]
gandalf	UP	11 OK	[Icons]
gateway	UP	1 OK	[Icons]
google	UP	2 OK	[Icons]
localhost	UP	6 OK	[Icons]
ssh.dei.isep.ipp.pt	UP	1 OK	[Icons]
sw2626a	UP	4 OK	[Icons]
wvm078	UP	8 OK 1 WARNING 1 CRITICAL	[Icons]
www.dei.isep.ipp.pt	UP	2 OK	[Icons]

Figura 15 - Print Screen de motorização dos serviços

- Pasta conf.d: contém todos os ficheiros de configuração que definem os hosts, serviços e contactos bem como respetivos grupos. Os principais ficheiros configurados que se destacam são:
 - `contacts_nagios2.cfg`: ficheiro que define os contactos ou grupos de contactos a serem notificados pelo nagios.
 - `deihttp.cfg`: ficheiro que define os serviços a monitorizar para o host `dei.isep.ipp.pt`

- gandalf.cfg: ficheiro que define os serviços a monitorizar para o servidor de base de dados gandalf.
 - generic-host_nagios2.cfg: ficheiro que define um conjunto de definições que serão herdadas por hosts que implementem o modelo definido neste ficheiro.
 - generic-services_nagios2.cfg: ficheiro que define um conjunto de definições que serão herdadas por serviços que implementem o modelo definido neste ficheiro.
 - google.cfg: ficheiro que define os serviços a monitorizar para o host google.pt
 - isephttp.cfg: ficheiro que define os serviços a monitorizar para o host portal.isep.ipp.pt
 - ssh.cfg: ficheiro que define os serviços a monitorizar para o host ssh.dei.isep.ipp.pt
 - switch.cfg: ficheiro que define os serviços a monitorizar para o switch sw2626a.dei.isep.ipp.pt
 - timeperiods_nagios2.cfg: ficheiro que onde estão definidos horários de notificações pelo nagios.
 - wvm078.cfg: ficheiro que define os serviços a monitorizar do servidor Windows onde está alojado o site Graphs4Social.
- Ficheiro commands.cfg: contém as definições dos comandos a utilizar nas notificações do nagios. Os comandos definidos são:
 - notify-host-by-email
 - notify-service-by-email
 - notify-service-by-facebook
 - host-notify-by-twitter
 - notify-by-twitter
 - facebook: contém as aplicações necessárias e os ficheiros de configuração para notificação do estado do servidor de dados e servidor web por facebook;



Figura 16- Print Screen - da atualização de um estado de facebook a partir de linha de comandos

- email: contém os ficheiros de configuração do smtp (aplicação pela qual é possível enviar o email) para notificação do estado do servidor de dados e servidor web por email.

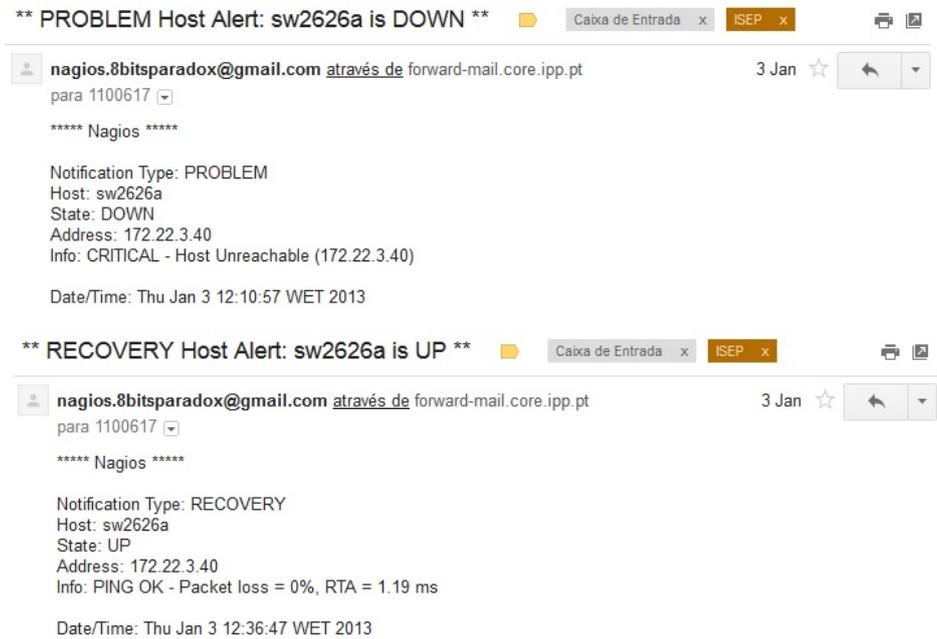


Figura 17 - Print Screen da notificação por e-mail

Em termos de servidores e estruturas de base de dados foi utilizado o seguinte:



Monitoring Server
Linux Server – uvm078

Server 172.31.100.78 /
uvm078.dei.issep.ipp.pt

Services:
- Nagios
- SSH
- PostFix(SMTP)

Descrição:
- O servidor linux vai efectuar toda a monitorização exigida do Database Server e do Web Server através do serviço Nagios. Para notificação por email vai também ser instalado o Postfix.



Database Server

Server : gandalf.dei.issep.ipp.pt

DB :ARQSI78

Descrição:
- O servidor gandalf vai alojar a base de dados da aplicação.



Web Server
Windows Server 2008 –
wvm078

Server:172.31.101.78 /
wvm078.dei.issep.ipp.pt

Services:
- NSClient++ (check_nt)
- SNMP Service (check_snmp)
- SMTP

Descrição:
- Este servidor windows vai alojar a aplicação Web desenvolvida.

d. Especificações das pré e pós condições das interfaces:

i. Web service:

- IService: esta interface faz a comunicação entre a BLL e cliente C++ do módulo de navegação e visualização 3D.
 - Métodos:

Assinatura do método	Pré condições	Pós condições
Utilizador checkAutentication(string username, string password)	O utilizador deve introduzir o seu username e password na aplicação local do módulo 3D.	Ligação à base de dados, às tabelas do Utilizador, Estado e as tabelas do Membership.
IList<Utilizador>getGrafoUtilizador(string user)	Inserir um username de um utilizador sobre o qual se pretenda saber os amigos e amigos de amigos ate ao 3º grau do grafo.	Existir o ficheiro db_conn.pl e a ligação à base de dados.
IList<RelacaoUtilizador>getRelacoesUtilizadores(string user)	Inserir um username de um utilizador sobre o qual se pretenda saber as relações de amigos ao 3º grau do grafo.	Existir ligação à base de dados, às tabelas de Utilizador, Estado, Relação e RelacaoTagRelacao
IList<Utilizador> getCaminhoForte(string nomeUser1, string nomeUser2)	Inserir um username de um utilizador origem e de destino sobre o qual se pretenda saber o caminho forte.	Existir o ficheiro db_conn.pl e a ligação à base de dados.

<p>IList<Utilizador> getCaminhoMinimo(string nomeUser1, string nomeUser2)</p>	<p>Inserir um username de um utilizador origem e de destino sobre o qual se pretenda saber o caminho forte.</p>	<p>Existir o ficheiro db_conn.pl e a ligação à base de dados.</p>
<p>IList<RelacaoUtilizador> getAllRelacoes()</p>	<p>Ser utilizador premium</p>	<p>Existir ligação à base de dados às tabelas: Utilizador, Estado, Relacao e RelacaoTagRelacao</p>
<p>IList<Utilizador> getAllUtilizadores()</p>	<p>Ser utilizador premium</p>	<p>Existir ligação à base de dados às tabelas: Utilizador, Estado.</p>
<p>IList<Utilizador> getAmigosComuns(string nomeUser1, string nomeUser2)</p>	<p>Ter o username de dois utilizadores sobre os quais se pretenda saber os amigos em comum</p>	<p>Existir o ficheiro: db_conn.pl e ligação à base de dados</p>
<p>IList<string> getSmallNetworkDate(string nomeUser1)</p>	<p>Ter o username de utilizador sobre o qual se pretenda saber as datas dos pedidos de aceite dos seus amigos.</p>	<p>Existir o ficheiro: db_conn.pl e ligação à base de dados</p>

IList<string> getAllRelacoesDates()	Ser utilizador Premium	Existir ligação à base de dados e as tabelas: Relacao e Utilizador
IList<RelacaoUtilizador> relacoesBydate(string data)	Ser utilizador Premium	Existir ligação à base de dados e as tabelas: Relacao e Utilizador

- Principais estruturas utilizadas no web service:

Utilizador	
ID	
membershipID	
nome	
idade	
sexo	
dataNasc	
email	
contacto	
imagem	
estado	
perfFacebook	
perfLinkedIn	
tipoConta	
coordenadaX	
coordenadaY	

Relacao	
id	
idUtilizadorOrigem	
idUtilizadorDestino	
dataPedido	
dataAceite	

Nome estrutura	Atributos principais	Descrição
Utilizador	ID	Chave primária do id do utilizador
	membership	Chave estrangeira para a tabela de membership
	Nome	Nome único do utilizador
	Idade	Idade do utilizador

	Sexo	Sexo do utilizador
	dataNasc	Data de Nascimento do utilizador
	Email	Email do utilizador
	contacto	Contacto telefónico do utilizador
	imagem	Nome do avatar do utilizador
	Estado	Estado de humor do utilizador
	Tipo de conta	Inteiro que indica o tipo de conta (0-normal, 1-premium).
	coordenada_X	Coordenada incrementada automaticamente de -15 a 15 (de 5 em 5).
	coordenada_Y	Coordenada incrementada automaticamente de 0 a $+\infty$ (de 5 em 5).
Nome estrutura	Atributos	Descrição
Relacao	Id	Chave primária da relação
	idUtilizadorOrigem	Id do utilizador origem da relação.
	idUtilizadorDestino	Id do utilizador que se encontra diretamente ligado ao anterior.
	dataPedido	Data do pedido da relação
	dataAceite	Data em que foi aceite o pedido

2. Principais estruturas de dados utilizadas:

Para além das estruturas já atrás referenciadas foram inicialmente utilizados os seguintes factos do módulo de Inteligência Artificial:

```
%user_profile(Id,Tags,DataNasc,Tel,Email,Linkedin,Facebook,Estado).
```

```
%user_conn(Id,ConnList).
```

```
%ConnList=[(UserIdA,TagA,Strength1),(UserIdB,TagB,Strength2),...]
```

% seman0c_eq([tag1, tag2,...,tagn]), que foram mais tarde reestruturadas para as estruturas que irão referenciadas na secção do modelo de dados:

```
db_attach( utilizador, 'Utilizador' ),
```

```
db_attach( tagUtilizador, 'TagUtilizador' ),
```

```
db_attach( tag, 'Tag' ),
```

```
db_attach( relacaoTagRelacao, 'RelacaoTagRelacao' ),
```

```
db_attach( tagRelacao, 'TagRelacao' ),
```

```
db_attach( relacao, 'Relacao' ),
```

```
db_attach( estado, 'Estado' ),
```

```
db_attach( semantica, 'Semantica').
```

No módulo de navegação e visualização 3D para o suporte do grafo, foi utilizado o *header*: grafos.h e o respetivo *source file*: grafo.cpp, onde foram definidas as seguintes constantes e estruturas de dados:

```
#define _MAX_NOS_GRAFO 100
#define _MAX_ARCOS_GRAFO 200
#define NORTE_SUL 0
#define ESTE_OESTE 1
#define PLANO 2
#define _MAX_TAMANHO_NO 2.0
#define _MIN_TAMANHO_NO 0.9
#define _MAX_TAMANHO_ARCO 0.9
#define _MIN_TAMANHO_ARCO 0.1
#define _MAX_COTA 14
#define _MIN_COTA 2
```

```
typedef struct Coordenadas{
    float x,y,z;
}Coordenadas;
```

```
typedef struct Avatar{
    StudioModel ModeloAvatar;
    double altura;
    string NomeAvatar;
    bool carregado;
    float tamanhoAvatar;
}Avatar;
```

```

typedef struct No{
    Coordenadas coordenadas;
    float largura,tamanhoAvatar;
    Avatar avatarUtilizador;
    string NomeUtilizador;
    int EstadoUtilizador;
    bool selecionado;
}No;

typedef struct Arco{
    No* noi;No *nof;
    float peso,largura;
    bool iluminado;
}Arco;
extern No nos[];
extern Arco arcos[];
extern int numNos, numArcos;

```

Para o suporte das restantes estruturas de dados foi utilizado o *header file* "EstruturasDados.h" e o respetivo *source file* "EstruturasDados.cpp". De entre as diferentes estruturas e constantes utilizadas nesses ficheiros, destacam-se as seguintes:

```

#define OBJECTO_ALTURA 4
#define OBJECTO_VELOCIDADE 0.5
#define OBJECTO_ROTACAO 5
#define OBJECTO_RAIO 2
#define SCALE_Avatar 0.005
#define EYE_ROTACAO 1
#define NUM_TEXTURAS 3
#define NUM_TEXTURAS_ESTADOS 4
#define OBJECTO_ALTURA_ESTADO 2.0
#define HEIGHT 900
#define WIDTH 500
#define BUFSIZE 100
#define HEIGHT_TIP 100
#define WIDTH_TIP 100

```

```

typedef struct pos_t{
    GLfloat x,y,z;
}pos_t;

```

```

typedef struct objecto_t{
    pos_t pos;
    GLfloat dir;
    GLfloat vel;
}objecto_t;

```

```

typedef struct teclas_t{
    GLboolean up,down,left,right;
}teclas_t;

```

```

typedef struct Camera{
    GLfloat fov;
    GLdouble dir_lat;
    GLdouble dir_long;
    GLfloat dist;
    Vertice center;

}Camera;

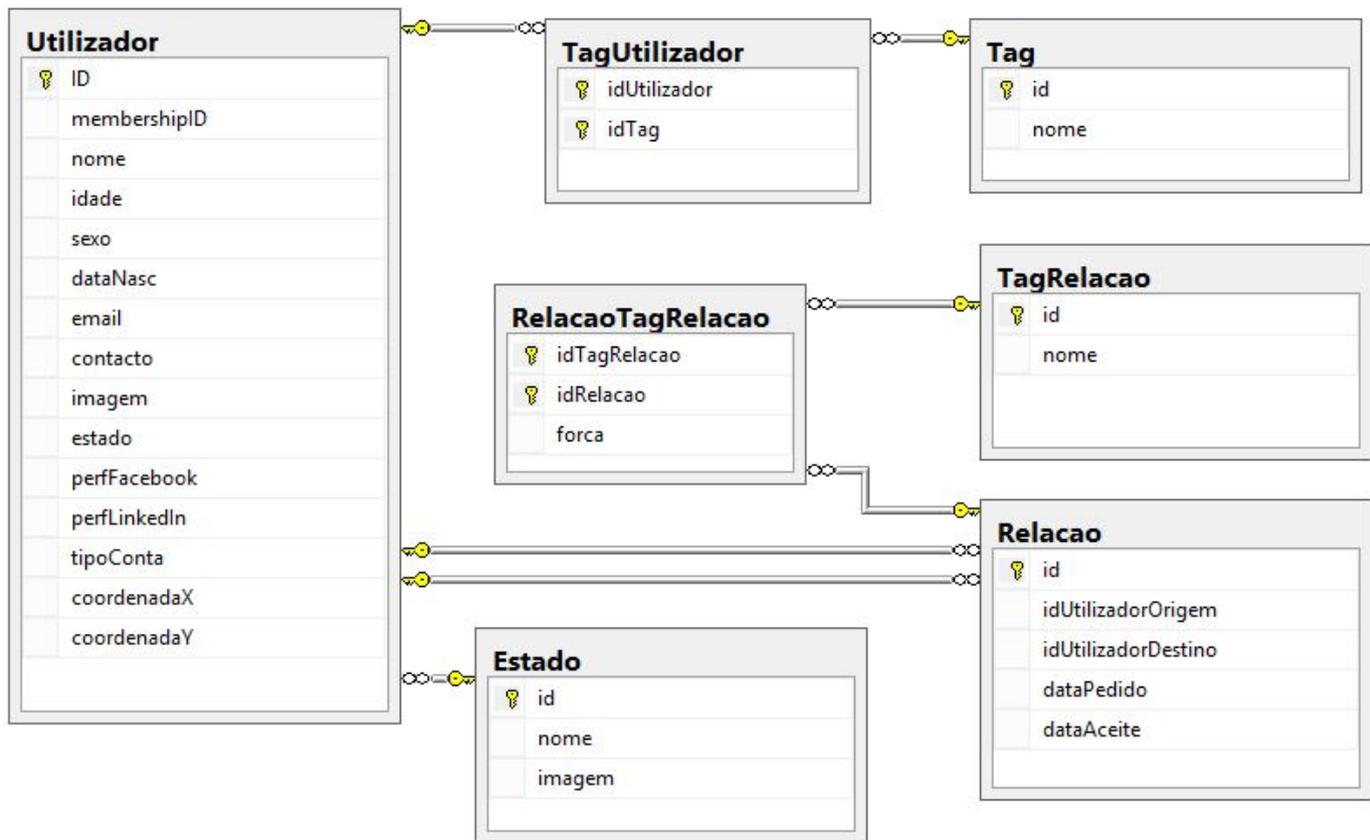
typedef struct Estado{
    Camera          camera;
    int
    xMouse,yMouse,locating,mouse_state,menu_inuse,noSeleccionado;
    GLboolean      light,reproducaoEfeito;
    GLboolean      apresentaNormais;
    GLint          lightViewer,inicioEfeito;
    GLint          eixoTranslacao;
    GLdouble       eixo[3];
    GLint          vista;//0->livre      1->razante
    GLint          timer;
    teclas_t      teclas;
}Estado;

typedef struct Modelo {
    GLfloat g_pos_luz1[4];
    GLfloat g_pos_luz2[4];
    GLfloat escala;
    GLUquadric *quad;
    GLuint textura[NUM_TEXTURES];
    GLuint texturas_estados[NUM_TEXTURES_ESTADOS];
    objeto_t objeto;
    objeto_t obj_aux;
    GLuint      prev,theObject,UtilizadorPos;
}Modelo;

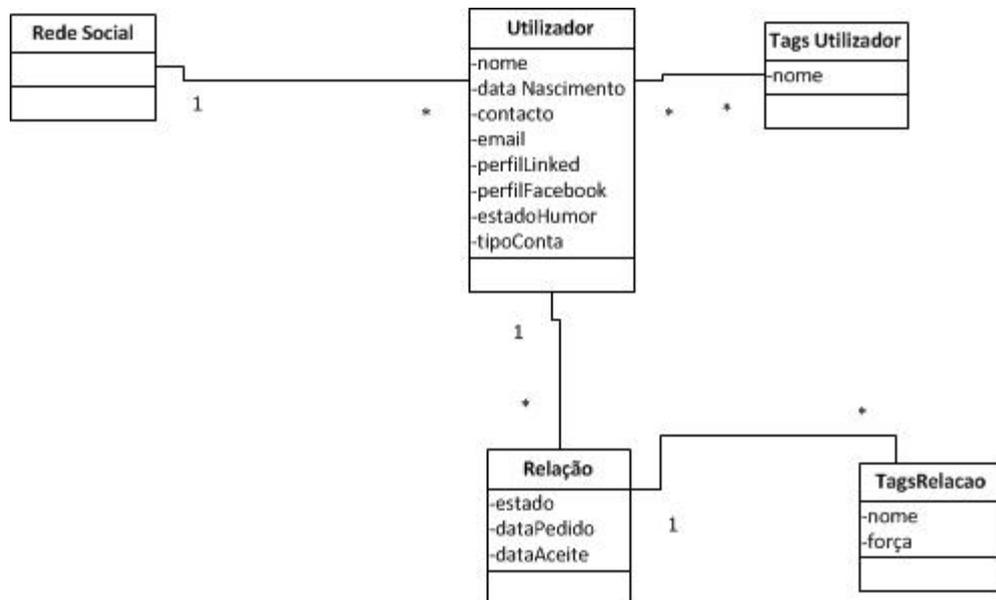
typedef struct {
    float x,y,z;
    unsigned int color;
    float u, v;
} VERTEX;
extern VERTEX *Vertices;
extern int NumVertices;
extern Utilizador*  utilizador;
extern Estado estado;
extern Modelo modelo;

```

3. Modelo de Dados:

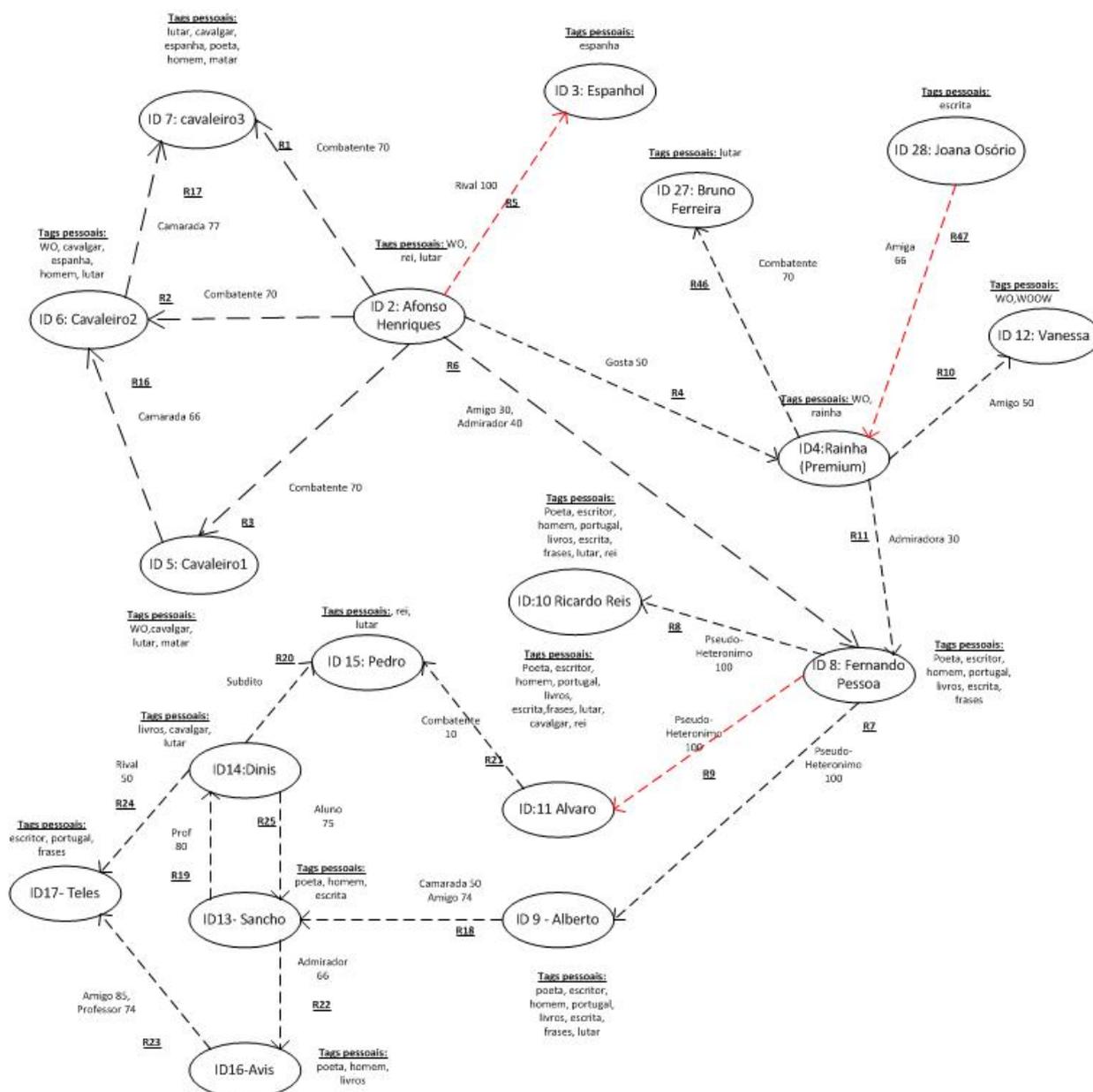


4. Modelo de Domínio:



5. Testes:

a. Grafo de testes utilizado:



Nota: As relações a vermelho, encontra-se pendentes, pelo que são só consideradas relações no prolog, site e módulo de visualização e navegação 3D, depois de serem aceites.

b. Descrição de testes unitários:

i. Testes unitários à camada de negócio:

Foram feitos testes unitários às principais classes de negócios, que são as seguintes:

- Relação
- Utilizador
- Tag
- TagRelação
- Estado

A cada uma destas classes, foram feitos métodos para testar a sua leitura e escrita. Com isto certificamo-nos que não existem falhas nesta principal e importante componente. Na classe Relação, foi feito em particular o teste ao método, “mediaForcaRelacao” que retorna a média da força da relação em questão. Na classe Utilizador foram testados os seguintes métodos: (1)*getIDByMembershipID*, (2)*GetNTagsPessoais*, (3)*getNRelacoes*: (1)neste primeiro teste é verificada a leitura de um utilizador com base no seu identificador de “membership “;(2) no segundo é testado o número de tags pessoais de um utilizador;(3) no último teste é verificado o retorno do numero de relações do mesmo.

Para as classes Tag e TagRelação, é verificado o número de vezes que essa Tag é usada com o método “getNvezesUsada”.

ii. Projeto main de testes feitos ao IA Manger:

Por não ser um projeto de testes unitários, mas sim parte integrante do plano de testes do módulo de Inteligência Artificial, este encontra-se em anexo na pasta Documentação\Testes e Dados de teste, não estando presente na solução final.

Neste foram testados os seguintes predicados do ficheiro db_conn.pl:

- getSmallNetwork
- getCommonFrieds
- getSuggestions
- getFriendsByTag
- getMaven
- getCMinimo
- getCForte
- getMedia
- getEco
- getMedia

Este pequeno projeto para além de permitir testar o resultado de cada um método, segundo o plano de testes delineados, permite ainda testar a execução da chamada dos métodos pela shell do MS-DOS, que por sua vez invoca o programa: win-prolog e executa o predicado desejado, fechando a janela automaticamente.

Para além disto, testa se a transformação e a leitura de ficheiros foi a desejada, quer o retorno seja uma lista de utilizadores ou valores decimais.

c. Plano de testes:

i. Módulo de Inteligência Artificial:

Módulo:	Inteligência Artificial	
Objetivo:	Testar o cálculo da rede um utilizador até ao 3º nível	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Dados da base de dados	getSmalNetwork(12).	no ficheiro output.txt deve ser apresentado []
Dados da base de dados	getSmalNetwork(11).	no ficheiro output.txt deve ser apresentado [15]
Dados da base de dados	getSmalNetwork(2).	no ficheiro output.txt deve ser apresentado [12,27,9,10,7,6,5,4,8]
Módulo:	Inteligência Artificial	
Objetivo:	Testar sugestões de amigos que tenham pelo menos uma tag em comum	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Dados da base de dados	getSuggestions(2).	no ficheiro output.txt deve ser apresentado [12,27,9,10]
Módulo:	Inteligência Artificial	
Objetivo:	Testar se amigos em comum entre dois utilizadores retornados estão corretos	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Dados da base de dados	getCommonFriends (4,2).	no ficheiro output.txt deve ser apresentado [8]
Módulo:	Inteligência Artificial	
Objetivo:	Testar se amigos que tenham uma, ou mais, tags em comum retornados estão corretos	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Dados da base de dados	getFriendsByTag (4,[1]).	no ficheiro output.txt deve ser apresentado [12]
Módulo:	Inteligência Artificial	
Objetivo:	Testar se dada uma tag, retorna o ou os mavens dessa tag	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Dados da base de dados	getMaven ([1]).	No ficheiro output.txt deve ser apresentado [2]

Módulo:	Inteligência Artificial	
Objetivo:	Testar se dado um utilizador inicial e final retorna o caminho mais forte	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Dados da base de dados	getCForte (2,9).	[2,4,8,9] no output.txt
Módulo:	Inteligência Artificial	
Objetivo:	Testar se dado um utilizador inicial e final retorna o caminho mínimo	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Dados da base de dados	getCMin (2,9).	[2,8,9] no output.txt
Módulo:	Inteligência Artificial	
Objetivo:	Testar o eco de uma mensagem de um determinado utilizador	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Dados da base de dados	getEco(2).	[9,10,13,14,16,7,6,5,4,8]no output.txt

Nota: Foi ainda criado outro predicado auxiliar com finalidade de teste, que permite reproduzir todo o plano do módulo IA. Este predicado executa uma serie de predicados com valores de teste e são analisadas os valores obtidos. Este predicado serviu para testar os predicados a medida que estes iam sendo alterados.

ii. Módulo de visualização e navegação 3D:

Módulo:	Visualização 3D	
Objetivo:	Entrada no sistema	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Dados do utilizador da base de dados	Nome de um utilizador inexistente	Acesso negado ao sistema
Dados do utilizador da base de dados	Nome de utilizador correto e password errada	Acesso negado ao sistema
Dados do utilizador da base de dados	Nome de utilizador e password corretos	Mensagem de boas vindas, tipo de conta, e acesso ao sistema
Dados do utilizador da base de dados	Entrar no sistema como os dados de "Afonso Henriques"	Mensagem de boas vindas com o nome e tipo de conta: conta normal

Módulo:	Visualização 3D	
Objetivo:	Visualização da rede 3D	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Entrar no sistema com o utilizador "Afonso Henriques"	Ver rede do utilizador	Entrada no sistema, e visualizar os amigos e ligações de "Afonso Henriques"
Entrar no sistema com o utilizador "Afonso Henriques"	Verificar nós	O raio de cada nó está em relação com o número de Tags de cada utilizador no grafo
Entrar no sistema com o utilizador "Afonso Henriques"	Verificar nós	A altura de cada nó está em relação com o número de relações, do utilizador em questão, dos utilizadores no grafo
Entrar no sistema com o utilizador "Afonso Henriques"	Verificar nós	Cada nó aparece segundo o algoritmo de posicionamento otimizado implementado no sistema
Entrar no sistema com o utilizador "Afonso Henriques"	Verificar Arcos	O raio de cada arco está em relação com a força de cada relação no grafo
Entrar no sistema com o utilizador "Afonso Henriques"	Verificar Avatares	Todos os avatares estão corretos, com os que cada utilizador escolheu no website
Entrar no sistema com o utilizador "Afonso Henriques"	Verificar Estados	Todos os estados estão corretos, com os que cada utilizador escolheu no website
Entrar no sistema com o utilizador "Afonso Henriques"	Cores dos nós	O nó do utilizador autenticado aparece a vermelho.

Entrar no sistema com o utilizador "Afonso Henriques"	Cores dos nós	O nó do utilizador selecionado aparece a verde.
Entrar no sistema com o utilizador "Afonso Henriques"	Cores dos nós	O nó do utilizador cujo rato está por cima, aparece a dourado

Módulo:	Visualização 3D	
Objetivo:	Testar mini-mapa	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Entrar no sistema com o utilizador "Afonso Henriques"	Analisar nós no mini-mapa	Todos os nós são apresentados corretamente no mini-mapa
Entrar no sistema com o utilizador "Afonso Henriques"	Analisar arcos no mini-mapa	Todos os arcos são apresentados corretamente no mini-mapa
Entrar no sistema com o utilizador "Afonso Henriques"	Analisar estados no mini-mapa	Todos os estados são apresentados corretamente no mini-mapa
Entrar no sistema com o utilizador "Afonso Henriques"	Analisar avatares no mini-mapa	Todos os avatares são apresentados corretamente no mini-mapa
Entrar no sistema com o utilizador "Afonso Henriques"	Analisar cores dos nós no mini-mapa	O nó do utilizador autenticado aparece a vermelho no mini-mapa
Entrar no sistema com o utilizador "Afonso Henriques"	Analisar cores dos nós no mini-mapa	O nó do utilizador selecionado aparece a verde no mini-mapa
Entrar no sistema com o utilizador "Afonso Henriques"	Analisar cores dos nós no mini-mapa	O nó do utilizador cujo rato está por cima, aparece a dourado no mini-mapa

Módulo:	Visualização 3D	
Objetivo:	Verificar motor de partículas	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Carregamento de motor de partículas	Estado zangado	É carregado fogo
Carregamento de motor de partículas	Estado triste	É carregada chuva
Carregamento de motor de partículas	Estado alegre	É carregado fumo

Módulo:	Visualização 3D	
Objetivo:	Música de fundo	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Entrar no sistema com o utilizador “Afonso Henriques”	Ouvir música de fundo ao iniciar o ambiente 3D	Música começa a ser reproduzida quando se faz a autenticação na aplicação 3D
Entrar no sistema com o utilizador “Afonso Henriques”	Mudar de música	Ao clicar no x ou X a música em produção muda
Entrar no sistema com o utilizador “Afonso Henriques”	Pausar a música	Ao clicar no espaço a música em produção fica em pausa
Entrar no sistema com o utilizador “Afonso Henriques”	Aumentar o volume da música	Ao clicar em + o volume da música em produção fica mais alta
Entrar no sistema com o utilizador “Afonso Henriques”	Diminuir o volume da música	Ao clicar em - o volume da música em produção fica mais baixo
Entrar no sistema com o utilizador “Afonso Henriques”	Todas as músicas	Quando todas as fachas terminam, volta a reproduzir a primeira

Módulo:	Visualização 3D	
Objetivo:	SkyBox, Skydome ou Skyplane	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Entrar no sistema com o utilizador “Afonso Henriques”	SkyBox ao iniciar a aplicação 3D	A SkyBox é apresentada logo quando após a validação da autenticação
Entrar no sistema com o utilizador “Afonso Henriques”	Navegar livremente, comportamento da SkyBox	A SkyBox mantém-se estável e é apresentada corretamente, navegando em qualquer sentido e direção.
Entrar no sistema com o utilizador “Afonso Henriques”	Ver diferentes ambientes da SkyBox	A SkyBox muda corretamente de céu, caso seja de noite ou de dia

Módulo:	Visualização 3D	
Objetivo:	Chão	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Entrar no sistema com o utilizador "Afonso Henriques"	Chão ao iniciar a aplicação 3D	O chão é apresentado logo quando após a validação da autenticação
Entrar no sistema com o utilizador "Afonso Henriques"	Navegar livremente, comportamento do chão	O chão mantém-se estável e é apresentado corretamente, navegando em qualquer sentido e direção.
Entrar no sistema com o utilizador "Afonso Henriques"	Ver diferentes ambientes do chão	O chão muda corretamente de ambiente, mediante a estação do ano corrente.

Módulo:	Visualização 3D	
Objetivo:	Voo Rasante	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Entrar no sistema com o utilizador "Afonso Henriques"	Estabilidade em cima dos arcos	Quando a câmara está em cima de um arco, só é possível transitar por cima do arco, mudar para outro arco ou para um nó, seguindo um caminho
Entrar no sistema com o utilizador "Afonso Henriques"	Estabilidade em cima dos nós	Quando a câmara está em cima de um nó, só é possível transitar por cima do nó, mudar para um arco ou para um outro nó, seguindo um caminho
Entrar no sistema com o utilizador "Afonso Henriques"	Transição de arco para nó	Durante a transição de nó para arco, a câmara sobe o desnível para cima do nó.
Entrar no sistema com o utilizador "Afonso Henriques"	Transição de vista livre para voo rasante	Quando se está em vista livre, e se embate num nó, a vista passa automaticamente para voo rasante

Módulo:	Visualização 3D	
Objetivo:	Vista livre	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Entrar no sistema com o utilizador "Afonso Henriques"	Navegação	A câmara navega livremente pelo ambiente 3D sem defeitos, mantém-se estável e sendo possível, navegar em qualquer sentido e direcção.
Entrar no sistema com o utilizador "Afonso Henriques"	Colisões com nós	Quando a câmara colide com um nó, a câmara para, não sendo possível atravessar o nó
Entrar no sistema com o utilizador "Afonso Henriques"	Colisões com arcos	Quando a câmara colide com um arco, a câmara para, não sendo possível atravessar o arco

Módulo:	Visualização 3D	
Objetivo:	Tip flutuante	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Entrar no sistema com o utilizador "Afonso Henriques"	Mostrar tip	Quando o rato entra dentro da área do avatar a tip aparece, mostrando o nome do utilizador
Entrar no sistema com o utilizador "Afonso Henriques"	Mostrar tip	Quando o rato sai da área do avatar a tip desaparece, permanecendo tudo como estava inicialmente

Módulo:	Visualização 3D	
Objetivo:	Efeitos sonoros (estados)	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Validar efeito sonoro	O estado do utilizador é triste	Ouve-se chuva quando se clica
Validar efeito sonoro	O estado do utilizador é alegre	Ouve-se passarinhos quando se clica
Validar efeito sonoro	O estado do utilizador é zangado	Ouve-se trovoadas quando se clica

Módulo:	Visualização 3D	
Objetivo:	Exportação	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Testar exportação para ficheiro raster	Clicar f12	Print screen de toda a cena na pasta das exportações

Módulo:	Visualização 3D	
Objetivo:	Iluminação caminho forte	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Dados da base de dados da tabela utilizadores	<p>O utilizador "Afonso Henriques" faz log in e clica no utilizador "Alberto"</p> <p>O utilizador "Rainha" faz log in e clica no utilizador "Alberto" O utilizador "Rainha" faz log in e seleciona o utilizador "Afonso Henriques". Após a seleção, clica no utilizador "Alberto"</p>	<p>Os arcos entre "Afonso Henriques" - "Rainha", "Rainha" - "Fernando Pessoa" - "Fernando Pessoa" - "Alberto" são iluminados</p> <p>Os arcos entre "Rainha" - "Fernando Pessoa" - "Fernando Pessoa" - "Alberto" são iluminados</p> <p>Os arcos entre "Afonso Henriques" - "Rainha", "Rainha" - "Fernando Pessoa" - "Fernando Pessoa" - "Alberto" são iluminados</p>

Módulo:	Visualização 3D	
Objetivo:	Carregamento de estados	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Validar o carregamento de estados	O utilizador que fez log in tem o estado 1	É carregada a textura de um sol
Validar o carregamento de estados	O utilizador que fez log in tem o estado 2	É carregada a textura de um sol com nuvens
Validar o carregamento de estados	O utilizador que fez log in tem o estado 3	É carregada a textura de uma lua com nuvens
Validar o carregamento de estados	O utilizador que fez log in tem o estado 4	É carregada a textura de um trovão e uma nuvem

Módulo:	Visualização 3D	
Objetivo:	Carregamento de avatares	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Validar o carregamento de avatares	O utilizador "Afonso Henriques" faz log in	É carregado o avatar "Batman"
Validar o carregamento de avatares	O utilizador "Espanhol" faz log in	É carregado o avatar "Trunks"
Validar o carregamento de avatares	O utilizador "Rainha" faz log in	É carregado o avatar "Boneco de Neve"
Validar o carregamento de avatares	O utilizador "Cavaleiro1" faz log in	É carregado o avatar "Homer"

Módulo:	Visualização 3D	
Objetivo:	Utilização de picking	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Validar estado (durante o picking)	O utilizador não é selecionado	O estado não aparece
Validar estado (durante o picking)	O utilizador é selecionado	O estado aparece e o nó muda para amarelo quando selecionado

Módulo:	Visualização 3D	
Objetivo:	Carregamento de textura	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Carregamento de texturas	Existem texturas na pasta	As texturas são carregados
Carregamento de textura	Alteração da hora (antes 19h)	É carregada a textura “sky”
Carregamento de textura	Alteração da hora (depois das 19h)	É carregada a textura “night sky”
Carregamento de textura	Alteração da estação do ano para primavera(mudança da data do sistema)	É carrega a textura “relva”
Carregamento de textura	Alteração da estação do ano para verão(mudança da data do sistema)	É carrega a textura “areia”
Carregamento de textura	Alteração da estação do ano para outono(mudança da data do sistema)	É carrega a textura “outono”
Carregamento de textura	Alteração da estação do ano para inverno(mudança da data do sistema)	É carrega a textura “inverno”

Módulo:	Visualização 3D	
Objetivo:	Visualização da rede como utilizador Premium	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Rede de um utilizador Premium	Utilizador=rainha password= rainha	O utilizador vê toda a rede toda

Módulo:	Visualização 3D	
Objetivo:	Amigos em comum	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Dados dos utilizadores da base de dados	Selecionar Rainha e Afonso Henriques	O amigo em comum deve ser Fernando Pessoa

Módulo:	Visualização 3D	
Objetivo:	Verificar a rotação do Billboard	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Movimento da câmara	Alteração das coordenadas da câmara	Mediante a posição da câmara, o billboard roda, de modo a ficar de frente para esta

Módulo:	Visualização 3D	
Objetivo:	Verificar a rotação do mini-mapa	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Movimento da câmara	Alteração das coordenadas da câmara	Mediante a posição da câmara, o conteúdo do mini-mapa movimenta-se de forma a que as coordenadas da câmara estejam no centro do mini-mapa e acompanhem o movimento

Módulo:	Visualização 3D	
Objetivo:	Verificar a atração a câmara	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Movimento da câmara	Alteração das coordenadas da câmara de um ponto em que não está perto de um no ou um arco para um ponto onde está perto de um no ou arco	O modo da câmara passa de voo livre para o modo rasante

Módulo:	Visualização 3D	
Objetivo:	Posicionamento do grafo	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Carregamento dos utilizadores	O utilizador "Afonso Henriques" faz log in	Os amigos do utilizador são posicionados mediante o calculo do raio entre eles $(360/\text{numero_de_amigos})$ ficando os utilizadores "Cavaleiro3" "Cavaleiro2" "Cavaleiro1" "Rainha" "Espanhol" "Fernando Pessoa" são colocados de forma a que o angulo de cada um seja 60º

iii. Módulo do Site:

Módulo:	Site	
Objetivo:	Editar relações	
Método de teste:	Manual	
Cenário:	Teste	Cenário:
Website aberto e a correr com o utilizador “Afonso Henriques”	Verificar se na página de relações estão todas as relações do “Afonso Henriques”	Website aberto e a correr com o utilizador “Afonso Henriques”
Website aberto e a correr com o utilizador “Afonso Henriques”	Verificar as Tags da relação entre “Afonso Henriques” e “cavaleiro3”	Website aberto e a correr com o utilizador “Afonso Henriques”
Website aberto e a correr com o utilizador “Afonso Henriques”	Verificar a Tags da relação entre “Afonso Henriques” e “cavaleiro3”	Website aberto e a correr com o utilizador “Afonso Henriques”
Website aberto e a correr com o utilizador “Afonso Henriques”	Verificar a Tags da relação entre “Afonso Henriques” e “cavaleiro3”	Website aberto e a correr com o utilizador “Afonso Henriques”
Website aberto e a correr com o utilizador “Afonso Henriques”	Verificar a Tags da relação entre “Afonso Henriques” e “cavaleiro3”	Website aberto e a correr com o utilizador “Afonso Henriques”

Módulo:	Site	
Objetivo:	Editar Tag pessoais	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Website aberto e a correr com o utilizador “Afonso Henriques”	Verificar se na página de Tags estão todas Tags pessoais do “Afonso Henriques”	Segundo os dados da base dados, todas as Tags pessoais do “Afonso Henriques” encontram se na sua página
Website aberto e a correr com o utilizador “Afonso Henriques”	Verificar a edição de Tags pessoais do “Afonso Henriques”	Editada a Tag “Lutar” para “frases” com sucesso.
Website aberto e a correr com o utilizador “Afonso Henriques”	Verificar a edição de Tags pessoais do “Afonso Henriques”	Editada a Tag “frases” para “Lutar” com sucesso.
Website aberto e a correr com o utilizador “Afonso Henriques”	Verificar a Eliminação de Tags pessoais do “Afonso Henriques”	Eliminada a Tag “Escudos” com sucesso
Website aberto e a correr com o utilizador “Afonso Henriques”	Adicionar uma Tag pessoal a “Afonso Henriques”	Adicionada a Tag “Escudos” com sucesso .

Módulo:	Site	
Objetivo:	Páginas de Tag Pool	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Website aberto e a correr na página “Informações”	Verificar se as Tag pools são carregadas	Ambas as Tag pools são carregadas
Website aberto e a correr na página “Informações”	Verificar se todas as Tags pessoais se encontram no ficheiro XML de Tags pessoais	Todas as Tags pessoais se estão no ficheiro “TagCloud.xml”
Website aberto e a correr na página “Informações”	Verificar se todas as Tags de relação se encontram no ficheiro XML de Tags de relação	Todas as Tags de relação se estão no ficheiro “TagCloud.xml”
Website aberto e a correr na página “Informações”	Verificar o tamanho de letra das Tags pessoais, que estão no ficheiro XML de Tags pessoais	O tamanho de letra das Tags pessoais, estão de acordo com a percentagem definida
Website aberto e a correr na página “Informações”	Verificar o tamanho de letra das Tags de relação, que estão no ficheiro XML de Tags pessoais	O tamanho de letra das Tags de relação, estão de acordo com a percentagem definida

Módulo:	Site	
Objetivo:	Mudança de estado	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Website aberto e a correr	Verificar a mudança de estado	Clique em cima da imagem do estado, e é apresentado o próximo estado
Website aberto e a correr	Verificar a mudança de estado	Feito refresh ao site, e o estado que previamente foi alterado, é apresentado correctamente

Módulo:	Site	
Objetivo:	Transição entre páginas, gerais e privadas. Registo de utilizador	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Website aberto e a correr na página homepage	Tentar aceder a uma página que exige autenticação prévia sem ter autenticação	Página de erro 404
Website aberto e a correr na página homepage	Tentar aceder a uma página que exige autenticação prévia tendo autenticação feita	Apresenta a página corretamente
Website aberto e a correr na página homepage	Percorrer todas as páginas com e sem autenticação, para testar consistência do website	Todas as páginas apresentadas com sucesso
Website aberto e a correr na página homepage	Tentar autenticação com um utilizador inexistente	Apresentado erro de autenticação
Website aberto e a correr na página homepage	Tentar autenticação com um utilizador existente mas com a password errada	Apresentado erro de autenticação
Website aberto e a correr na página homepage	Tentar autenticação com dados corretos	Reencaminhamento para página de acesso privado
Website aberto e a correr na página de registo (1/3)	Tentar criar um utilizador com um nome de utilizador já existente	Erro de aviso que o utilizador já existe
Website aberto e a correr na página de registo (1/3)	Tentar criar um utilizador em que a password de verificação não coincide com a 1ª	Erro de as passwords não serem iguais
Website aberto e a correr na página de registo (1/3)	Tentar criar um utilizador com um nome de utilizador inexistente e ambas as passwords corretas	Reencaminhamento para a página de registo (2/3)
Website aberto e a correr na página de registo (2/3)	Tentar criar um utilizador sem idade	Erro de validação – a idade é obrigatória
Website aberto e a correr na página de registo (2/3)	Tentar criar um utilizador com uma data de nascimento invalida	Erro de validação – a idade é obrigatória
Website aberto e a correr na página de registo (2/3)	Tentar criar um utilizador com um contacto inválido	Erro de validação – Insira um contacto valido
Website aberto e a correr na página de registo (2/3)	Tentar criar um utilizador com todos os dados corretos	Reencaminhamento para a página de registo (3/3)
Website aberto e a correr na página de registo (3/3)	Tentar criar uma Tag	Tag criada foi adicionada à gridview de Tags
Website aberto e a correr na página de registo (3/3)	Tentar criar uma Tag já criada	É apresentada uma mensagem que diz “Já tem a Tag que acabou de inserir, esta será ignorada”

Website aberto e a correr na página de registo (3/3)	Tentar editar uma Tag	A Tag foi editada
Website aberto e a correr na página de registo (3/3)	Tentar editar uma Tag com o nome de uma já existente	É apresentada uma mensagem que diz “Já tem a Tag que acabou de editar, esta será ignorada”

Módulo:	Site	
Objetivo:	Download, Galeria de Avatares, Editar dados pessoais e Procura de relação	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Website aberto e a correr na página download	Tentar fazer o <i>download</i> da aplicação do módulo 3D	É feito de imediato, o <i>download</i> da aplicação
Website aberto e a correr na página de dados pessoais	Tentar editar um utilizador sem idade	Erro de validação – a idade é obrigatória
Website aberto e a correr na página de dados pessoais	Tentar editar um utilizador com uma data de nascimento invalida	Erro de validação – a idade é obrigatória
Website aberto e a correr na página de dados pessoais	Tentar editar um utilizador com um contacto inválido	Erro de validação – Insira um contacto valido
Website aberto e a correr na página de dados pessoais	Tentar editar um utilizador com todos os dados corretos	É apresentada uma mensagem que diz “Dados Atualizados ”
Website aberto e a correr na página de ligações	Visualizar relações do utilizador autenticado com outros utilizadores da rede.	São apresentados corretamente todas as relações do utilizador autenticado com outros utilizadores da rede.
Website aberto e a correr na página de ligações	Visualizar tags de uma relação do utilizador autenticado com outro utilizador da rede.	Após o pedido de visualização de tags de vários pedidos foram sempre apresentados corretamente todas as tags de cada relação
Website aberto e a correr na página de ligações	Apagar relações do utilizador autenticado com outros utilizadores da rede.	Após o pedido, a relação foi apagada da BD e é atualizada a tabela de relações.
Website aberto e a correr na página de ligações	Editar tag de uma relação.	É permitido ao utilizador editar o nome e a força da tag de uma relação.

Módulo:	Site	
Objetivo:	Download, Galeria de Avatares, Editar dados pessoais e Procura de relação	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Website aberto	Mudar estado através da funcionalidade de mudança de estado rápido.	Muda para a imagem de uma nuvem e um raio e não dá para alterar.
Website aberto e a correr na página de pedidos de ligações	Visualizar pedidos de ligação pendentes de um utilizador que não tenha pedidos pendentes	É apresentada uma mensagem que informa que não existem pedidos para aceitar.
Website aberto e a correr na página de pedidos de ligações	Visualizar pedidos de ligação pendentes de um utilizador que tenha pedidos pendentes	São apresentados corretamente todos os estados pendentes do utilizador
Website aberto e a correr na página de pedidos de ligações	Aceitar pedido de ligação pendente de um utilizador.	A relação foi corretamente adicionada às relações do utilizador que originou o pedido.
Website aberto e a correr na página de ligações	Apagar tag de uma relação.	É permitido ao utilizador apagar a tag de uma relação sem restrição ao número de tags de uma relação, ou seja a ligação pode ficar com zero tags. Visto que assumimos que a relação deve ter pelo menos 1 tag na sua criação, penso que devemos validar a eliminação de tags para não permitir eliminar quando o numero de tags for 1.
Website aberto e a correr na página de informações	Verificar se cloud de tags apresenta todas as tags pessoais e relacionais.	São apresentadas todas as tags corretamente.
Website aberto e a correr na página de pedidos de ligações	Visualizar pedidos de ligação pendentes de um utilizador que tenha pedidos pendentes	São apresentados corretamente todos os estados pendentes do utilizador
Website aberto e a correr na página de pedidos de ligações	Aceitar pedido de ligação pendente de um utilizador.	A relação foi corretamente adicionada às relações do utilizador que originou o pedido.

iv. Módulo de Infraestruturas computacionais:

Módulo:	Infraestrutura/Nagios	
Objetivo:	Testar monitorizações e notificações do nagios.	
Método de teste:	Manual	
Cenário:	Teste	Resultado esperado
Monitorização Nagios ao servidor windows em tempo real.	Monitorizar os processos críticos (status (on/off) dos serviços web) do jogo desenvolvido.	Apresenta valores e estado de acordo com o estado atual do item monitorizado.
Monitorização Nagios ao servidor Windows em tempo real.	Monitorizar no servidor no servidor Windows a Utilização do CPU	CPU Load 4% (5 min average)
Monitorização Nagios ao servidor windows em tempo real.	Monitorizar no servidor Windows a Taxa de utilização da placa de rede	Intel(R) PRO/1000 MT Network Connection #2-QoS Packet Scheduler-0000:UP (0.3KBps/0.0KBps), Intel(R) PRO/1000 MT Network Connection-QoS Packet Scheduler-0000:UP (5.0KBps/0.1KBps), Intel(R) PRO/1000 MT Network Connection:UP (5.0KBps/0.1KBps), Intel(R) PRO/1000 MT Network Connection #2:UP (0.3KBps/0.0KBps):4 UP: OK
Monitorização Nagios ao servidor windows em tempo real.	Monitorizar no servidor Windows a Utilização de discos	c: - total: 20.00 Gb - used: 17.02 Gb (85%) - free 2.98 Gb (15%)
Monitorização Nagios ao servidor windows em tempo real.	Monitorizar no servidor Windows a Utilização de memória	Memory usage: total:2311.95 Mb - used: 708.84 Mb (31%) - free: 1603.12 Mb (69%)
Monitorização Nagios ao servidor windows em tempo real.	Monitorizar no servidor Windows a Utilização de serviços	OK: All processes are running.
Monitorização Nagios ao servidor gandalf em tempo real.	Monitorizar no servidor Gandalf (BD) a Utilização do CPU	CPU Load 7% (5 min average) 6% (10 min average)
Monitorização Nagios ao servidor gandalf em tempo real.	Monitorizar no servidor Gandalf (BD) a Taxa de utilização da placa de rede	VMware Accelerated AMD PCNet Adapter:UP (WARN 172.6KBps/116.0KBps):(1 UP): WARNING
Monitorização Nagios ao servidor gandalf em tempo real.	Monitorizar no servidor Gandalf (BD) a Utilização de discos	c: - total: 140.00 Gb - used: 71.52 Gb (51%) - free 68.48 Gb (49%)
Monitorização Nagios ao servidor gandalf em tempo real.	Monitorizar no servidor Gandalf (BD) a Utilização de memória	Memory usage: total:5214.19 Mb - used: 2417.56 Mb (46%) - free:

		2796.62 Mb (54%)
Monitorização Nagios ao servidor gandalf em tempo real.	Monitorizar no servidor de dados e no servidor Gandalf (BD) a Utilização de serviços	OK: All processes are running.
Monitorização Nagios ao servidor www.dei.isep.ipp.pt em tempo real.	Monitorizar o serviço HTTP no servidor www.dei.isep.ipp.pt	HTTP OK: HTTP/1.1 200 OK - 8963 bytes in 0.018 second response time
Monitorização Nagios ao servidor www.dei.isep.ipp.pt em tempo real.	Monitorizar o serviço HTTPS no servidor www.dei.isep.ipp.pt	HTTP OK: HTTP/1.1 200 OK - 8963 bytes in 0.077 second response time
Monitorização Nagios ao servidor portal.isep.ipp.pt em tempo real.	Monitorizar o serviço HTTP no servidor portal.isep.ipp.pt	HTTP OK: HTTP/1.1 302 Object moved - 459 bytes in 0.007 second response time
Monitorização Nagios ao servidor portal.isep.ipp.pt em tempo real.	Monitorizar o serviço HTTPS no servidor portal.isep.ipp.pt	HTTP OK: HTTP/1.1 302 Object moved - 467 bytes in 0.030 second response time
Monitorização Nagios ao servidor www.google.pt em tempo real.	Monitorizar o serviço HTTP no servido www.google.pt	HTTP OK: HTTP/1.1 200 OK - 11624 bytes in 0.088 second response time
Monitorização Nagios ao servidor ssh.dei.isep.ipp.pt em tempo real.	Monitorizar o serviço SSH de um dos servidores do DEI	SSH OK - OpenSSH_5.3p1 Debian-3ubuntu7 (protocol 2.0)
Monitorização Nagios ao servidor sw2626a.dei.isep.ipp.pt em tempo real.	Monitorizar no dispositivo sw2626a.dei.isep.ipp.pt o uso de largura de banda da porta 25	Traffic OK - .21 Mb/s in Sum
Monitorização Nagios ao servidor sw2626a.dei.isep.ipp.pt em tempo real.	Monitorizar o tráfego de rede (via SNMP) na porta UPLINK (porta 25) (usar o plugin check_snmp_int.pl)	25:UP (25.4KBps/4.7KBps):1 UP: OK
Monitorização Nagios em tempo real.	envio de email de notificação de estados (warning critical)	Recebido via email a notificação.
Monitorização Nagios em tempo real.	envio de tweet de notificação de estados (warning critical)	Recebido via twitter a notificação.
Monitorização Nagios em tempo real.	envio de "post" para facebook de estados (warning critical)	Funciona na linha de comandos, no entanto não funcionou no NAGIOS.
Notificação Nagios em tempo real.	Efectuar um pedido de notificação de um serviço que esteja a ser monitorizado	Recebido via email e twitter a notificação.

6. Divisão de tarefas

a. Descrição da divisão de tarefas:

A divisão de tarefas é dividida em três sprints, sendo que um acabou dia 18 de Dezembro(entrega da primeira demo intermédia), o segundo acabou dia 9 de Janeiro(entrega da segunda demo intermédia) e o terceiro acabou no dia da demo final 16 de Janeiro(entrega da demo final), pelo que algumas tarefas como os slides de apresentação ou tarefas de valorização se não encontram não feitos, ou ficaram no estado "em curso".

A primeira iteração tem a planificação geral inicialmente idealizada pelo grupo, apenas as tarefas em "em curso" ou "não feitas" transitam para a iteração seguinte, mas ao longo de cada iteração surgem novas tarefas que inicialmente não estavam previstas.

Este tipo de planificação permite uma ideia geral da evolução do grupo e como as tarefas de cada um, se encontram sublinhadas no respetivo Excel, é projetada uma ideia geral do envolvimento do elemento no projeto.

Para além disso cada Excel de divisão de tarefas é complementado com o log individual, que descreve pormenorizadamente as tarefas de cada elemento.

b. Descrição do desenvolvimento do projeto:

Foi utilizado um processo de desenvolvimento ágil, que incluía: reuniões semanais do grupo com o cliente, reuniões diárias do grupo, registo diário do log no grupo e respetivos registos no repositório de controlo de versões.

Foi feita uma gestão distribuída de versões, no qual cada elemento tinha o repositório individual que era por sua vez, integrado no repositório central do github.

Para controlar o repositório privado do github foi utilizada a respetiva ferramenta: "*github for windows*". A respetiva ferramenta tem apenas um histórico de commits não contendo, qualquer tipo de grafo de revisões de cada subprojecto, sendo que a situação foi exposta no fórum ao respetivo regente da unidade curricular e nos foi dito para mostrar apenas o histórico dos commits efetuados por cada elemento do grupo. No entanto, na página do github é possível a visualização de vários tipos gráficos, sendo que o mais parecido com o pedido, é gráfico dos commits feitos por cada elemento, excluindo os merge efetuados.

Foram feitos testes unitários ao C#, à camada de negócio, que são descritos na respetiva secção e foram ainda feitos planos de testes e construídos "base de dados" para reproduzir os mesmos, toda a documentação sobre isto encontra-se na respetiva secção.

A tarefas e responsabilidades ao longo do projeto foram divididas de acordo com seguinte hierarquia de cargos, sendo que alguns dos cargos são apenas fictícios, para uma gestão mais eficaz do grupo: (1) coordenador do projeto: foi pensada a ideia de uma figura de coordenador do projeto que é o regente da unidade curricular, esta figura é simultaneamente "cliente" nas reuniões semanais e coordenador, no sentido que apoia os alunos no seu planeamento; (2) chefe de equipa: foi criada a figura de chefe de equipa, esta figura é desempenhada pelo elemento de equipa, Joana

Osório e tem como principais funções: garantir que a equipa preenche o log semanal, garantir o estado da divisão de tarefas, submeter o repositório da equipa, garantir a integração das diferentes áreas e da documentação; (3) coordenadores de áreas: existem quatro coordenadores, um por cada módulo principal, sendo que o Bruno Ferreira se encontra responsável por coordenar o módulo de site, que o Fábio Santos se encontra responsável por coordenar o módulo de visualização e navegação 3D, que o Fernando Afonso se encontra responsável por coordenar o módulo de Inteligência Artificial e o José Pedro Ribeiro se encontra responsável por coordenar o módulo de Infraestrutura computacional, estes coordenadores encontram-se responsáveis pela atribuição das tarefas do seu módulo a cada elemento da equipa, encontram-se também responsáveis pela integração da sua parte e pela reportação do estado da área ao chefe de equipa; (3) consultores/orientadores técnicos: esta função é desempenhada pelos docentes de OT, que têm como principal função prestar apoio aos elementos de grupos nas áreas técnicas; (4) elemento da equipa: este papel é assumido por todos os elementos da equipa, têm como principais responsabilidades: atualizar o log diário, colaborar na integração do trabalho e colaborar na elaboração do relatório técnico.