|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Inteligência artificial | | |
| **Objectivo:** | Testar listagem de amigos sugeridos a um utilizador (com *tags* em comum entre os dois utilizadores) | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Base de conhecimento “recomendaAmigos.pl” | sugerir\_users(10). | 7[rock],20[pimba] | 7[rock],20[pimba] |
| sugerir\_users(7). | 20[terror] | 20[terror] |
| sugerir\_users(100). | --- | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Inteligência artificial | | |
| **Objectivo:** | Determinar o menor caminho entre dois utilizadores | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Base de conhecimento “menorCaminho.pl” | cam\_min(10,7,L). | L=[10,7] | L=[10,7] |
| cam\_min (10,20,L). | L=[10,7,20] | L=[10,7,20] |
| cam\_min (10,100,L). | --- | --- |
| cam\_min (10,21,L). | L=[10,7,21] | L=[10,7,21] |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Inteligência artificial | | |
| **Objectivo:** | Determinar o menor caminho entre dois utilizadores | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Base de conhecimento “tamanhoRedeUtilizador.pl” | tamanho\_rede\_user\_3(10,N). | N=4 | N=4 |
| tamanho\_rede\_user\_3(4,N). | *no* | *no* |
| tamanho\_rede\_user\_3(20,N). | N=3 | N=3 |
| tamanho\_rede\_user\_3(11,A). | A=2 | A=2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Inteligência artificial | | |
| **Objectivo:** | Listar os amigos até terceiro nível de um utilizador (imprime lista) | | |
| **Métodos de teste:** | Manual | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Base de conhecimento “lapr.pl” | rede\_user\_3 (a,N). | [b,c,d,e] | [b,c,d,e] |
| rede\_user\_3 (b,N). | [c,d,e,f] | [c,d,e,f] |
| rede\_user\_3 (z,N). | *no* | *no* |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Inteligência artificial | | |
| **Objectivo:** | Determinar Grau Médio de Separação entre dois Nós | | |
| **Métodos de teste:** | Manual | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Base de conhecimento “separacao.pl” | Grau(a,b,N). | N=1 | N=1 |
| Grau(a,x,N). | *N=2,5* | *N=2,5* |
| Grau(w,b,N). | N=2,5 | N=2,5 |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Site | | |
| **Objectivo:** | Carregar todos os nós da base de dados para uma lista. | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Ligação a base de dados através da BLL.  Chamar os métodos por servicereference localmente. | GetAllXY() | Id, x , y  de todos os nós | Id, x , y  de todos os nós |
| GetAllXY() | Id, x , y, humor, nome  de todos os nós | Id, x , y, humor, nome  de todos os nós |
| GetAllXY() | Id, x , y, humor, nome, avatar, numero de tags  de todos os nós | Id, x , y, humor, nome, avatar, numero de tags  de todos os nós |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Site | | |
| **Objectivo:** | Carregar todos os arcos da base de dados para uma lista. | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Ligação a base de dados através da BLL.  Chamar os métodos por servicereference localmente. | GetAllArcoBD() | Id da relação,  id do utilizador 1,  id do utilizador 2,  força  de todos os arcos | Id da relação,  id do utilizador 1,  id do utilizador 2,  força  de todos os arcos |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Site | | |
| **Objectivo:** | Carregar os nós até ao 3º nível de amizade de um utilizador da base de dados para uma lista. | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Ligação a base de dados através da BLL.  Chamar os métodos por servicereference localmente. | GetGrafoNivel3(intuserid) | Id, x , y, humor, nome, avatar, numero de tags  de todos os nós | Id, x , y, humor, nome, avatar, numero de tags  de todos os nós |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Site | | |
| **Objectivo:** | Carregar os nós dos amigos em comum de dois utilizadores da base de dados para uma lista. | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Ligação a base de dados através da BLL.  Chamar os métodos por servicereference localmente. | GetNosGrafoAmigos(int user1, int user2) | Id, x , y, humor, nome, avatar, numero de tags  de todos os nós | Id, x , y, humor, nome, avatar, numero de tags  de todos os nós |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Inteligência artificial | | |
| **Objectivo:** | Listar os amigos até terceiro nível de um utilizador (imprime lista) | | |
| **Métodos de teste:** | Manual | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Base de conhecimento “caminho mais forte.pl” | camMaisForte(a,d,L). | L=[a,c,d] | L=[a,c,d] |
| camMaisForte(a,ww,L). | Nenhum caminho | Nenhum caminho |
| camMaisForte(a,a,L). | Nenhum caminho | Nenhum caminho |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Inteligência artificial | | |
| **Objectivo:** | Obter o utilizador estrela de uma tag | | |
| **Métodos de teste:** | Manual | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Base de conhecimento “maven.pl” | maven('rock',U). | U = e | U = e |
| maven('roc',U). | Nenhum Maven | Nenhum Maven... |
| maven('pimba',U). | Nenhum Maven | Nenhum Maven... |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Inteligência artificial / C # | | |
| **Objectivo:** | Obter os nós para o ficheiro pl | | |
| **Métodos de teste:** | Manual | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Simular o webservice a ir buscar os nós a base de dados para gerar o ficheiro pl | obterNosPublic() | Nós da base de dados  no(noID,[Tags],X,Y). | no(noID,[Tags],X,Y)./n |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Inteligência artificial / C # | | |
| **Objectivo:** | Obter os ramos para o ficheiro pl | | |
| **Métodos de teste:** | Manual | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Simular o webservice a ir buscar os ramos a base de dados para gerar o ficheiro pl | obterRamosPublic() | Nós da base de dados  Ramo  (IDA,IDB,[Tags],força) | Ramo  (IDA,IDB,[Tags],força)./n |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Inteligência artificial / C # | | |
| **Objectivo:** | Obter o conteúdo do ficheiro prolog conforme o pedido | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Simular o webservice a ir buscar o ficheiro de texto para gerar o ficheiro pl.  2º teste erro na ligação ao base de dados  3º não encontra prolog pedido | obterLines(target) | Nos  Ramos  MetodoProlog  Run | Nos  Ramos  MetodoProlog  Run |
| obterLines(target) | Null | Null |
| obterLines(target) | Null | Null |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Inteligência artificial / C # | | |
| **Objectivo:** | Executar um comando prolog autonomamente. | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Simular o webservice a executar um comando prolog, fornecendo o id do utilizador que faz o pedido e o nome do comando (ie. menorCaminho, grafoUser) | PrologExec p = new PrologExec("user", "maven");  p.executaComandoProlog("musica"); | 7 | 7 |
| PrologExec p = new PrologExec("user", "maven");  p.executaComandoProlog("rock"); | 1 | 1 |
| PrologExec p = new PrologExec("user", "maven");  p.executaComandoProlog("m"); | Nenhum maven | Nenhum maven |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Webservice / C++ | | |
| **Objectivo:** | Retornar o profileid do utilizador que faz login | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Webservice vai ao membership do site para validar o username | validateLogin  ("RafaUser",”qwerty”) | 1 | 1 |
| validateLogin  ("RafaUser",qwert) | 0 | 0 |
| validateLogin  ("",””) | 0 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Site | | |
| **Objectivo:** | Procurar utilizadores registados | | |
| **Métodos de teste:** | Manual | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Relacoes.aspx  Relacoes.aspx.cs  -Button1\_Click(object sender, EventArgs e) | Procurar user com “r” no nome | Exemplo: Rafael | Carlos, Rafael, Bruno. |
| Procurar user com nome “Rafael” | Rafael | Rafael |
| Procurar user com “h” no nome | User com h no nome | Vazio (não existe) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Site | | |
| **Objectivo:** | Listar sugestões de pessoas que o utilizador possa conhecer (tags em comum) | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Relacoes.aspx  -Sugestoes()  -sugere\_amigos(string);  Service.cs  -sugereAmigos(int) | User: Carlos  Tags: rock, musica, teatro, pt | Users com mesmas tags | User: Rafael  Tagscomum:rock, musica teatro.  User:Bruno  Tags comum: rock, musica, pt |
| User:Fabio  Tags:Rock, Pop, cinema, terrorismo, portugal | Users com mesmas tags | User: Bruno  Tags comum: rock, portugal |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Site | | |
| **Objectivo:** | Registar Users | | |
| **Métodos de teste:** | Manual/Automatico | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Register.aspx  -Username  -Email  -Password  -Confirm Password  Register.aspx.cs  -RegisterUser\_CreatedUser(objectsender, EventArgs e) | Username:CarlosUser  Password:qwerty  Email:xxxxxxx@isep.ipp.pt | User registado e inserido na BD | User registado e inserido na BD |
| Username:FabioUser  Password:qwerty  Email:xxxxxxx@isep.ipp.pt | User registado e inserido na BD | User registado e inserido na BD |
| Username:RafaUser  Password:qwerty  Email:xxxxxxx@isep.ipp.pt | User registado e inserido na BD | User registado e inserido na BD |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Site | | |
| **Objectivo:** | Criar Perfis de utilizadores registados | | |
| **Métodos de teste:** | Inserção de dados manual, save Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Perfil.aspx  -Nome, Nick, Morada, DataNascimento, Telemovel, Facebook, Linkedin, Avatar3D, Avatar, Humor  Perfil.cs  -save() | Guardar perfil de Fabio na BD e carregar para o site | Perfil guardado e carregado em Perfil.aspx | Perfil guardado e carregado em Perfil.aspx |
| Guardar perfil de Rafael na BD e carregar para o site | Perfil guardado e carregado em Perfil.aspx | Perfil guardado e carregado em Perfil.aspx |
| Guardar perfil de Bruno na BD e carregar para o site | Perfil guardado e carregado em Perfil.aspx | Perfil guardado e carregado em Perfil.aspx |
| Guardar perfil de Carlos na BD e carregar para o site | Perfil guardado e carregado em Perfil.aspx | Perfil guardado e carregado em Perfil.aspx |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Site | | |
| **Objectivo:** | Mostrar nº Users Registados | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Default.aspx  Default.aspx.cs  -Dimensao\_rede()  Perfil.cs  -get\_UsersRegistados() | Mostrar na homepage os utilizadores registados no sistema | 8 (guardados na bd) | Utilizadores Registados:    8 |
| Mostrar na homepage os utilizadores registados no sistema | 8 (guardados na bd) | Utilizadores Registados:    8 |
| Mostrar na homepage os utilizadores registados no sistema | 8 (guardados na bd) | Utilizadores Registados:    8 |
| Mostrar na homepage os utilizadores registados no sistema | 8 (guardados na bd) | Utilizadores Registados:    8 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | Site | | |
| **Objectivo:** | Mostrar tags (cloud) de relação e de perfil com fonte de texto de tamanho proporcional ao número de ocorrências | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Default.aspx  Default.aspx.cs  -Cloud\_taguser()  - Cloud\_tagrelacao()  Tag.cs  -TagUser()  - TagCloudRel() | Mostrar na homepage as tags de perfil no formato cloud | Tagscloud de perfil com tamanhos proporcionais as ocorrências | **vizinho terrorismo teatro rock pt** |
| Mostrar na homepage as tags de relaçãono formato cloud | Tagscloud de relação com tamanhos proporcionais as ocorrências | amigo colega lololol namorado pai ra rock Terrorismo\_Assistido vizinho |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | C++/WebService | | |
| **Objectivo:** | Efectuar o login na aplicação gráfica | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| 1 - Introduzir username e password  2- Clicar em entrar  3- A aplicaçao c++ comunica com o servidor para obtenção do id do utilizador que fez login (em caso de sucesso) ou 0 (no caso das credenciais estarem erradas). | Introduzir username ou password incorrectos | Mensagem de erro "Dados incorrectos" | Mensagem de erro "Dados incorrectos" |
| Introduzir username ou password correctos | Carregamento e visualização do grafo | Carregamento e visualização do grafo |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | C++/webservice | | |
| **Objectivo:** | Verificar se os nós e ligações são obtidos correctamente do servidor | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **obtido** |
| Depois de ser efectuado o login com sucesso na aplicação c++, é mostrado o grafo que é obtido através do webservice. | Nós (1,2,3)  Ligacoes (1 a 2, 2 a 3) | 3 Nós e 2 ligações entre eles | 3 Nós e 2 ligações entre eles |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | C++/webservice | | |
| **Objectivo:** | Obtenção do estado de humor dos utilizadores e desenhar a billboard correspondente ao estado de humor obtido. | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **Obtido** |
| Ao serem obtidos os nós e as ligações do servidor, tambem é obtido o estado de humor dos utilizadores correspondentes aos nós. Esse estado de humor obtido deve ser usado para desenhar uma billboard representativa dessa humor. | Verificar se as billboards desenhadas correspondem ao estado de humor dos utilizadores. | As billboards desenhadas correspondem ao estado de humor dos utilizadores. | As billboards desenhadas correspondem ao estado de humor dos utilizadores. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo:** | C++/webservice | | |
| **Objectivo:** | Testar a câmara (voo livre). | | |
| **Métodos de teste:** | Automático | | |
| **Cenário** | **Teste** | **Resultado esperado** | **Resultado**  **Obtido** |
| Pressionando as teclas para controlo da câmara, ela deve seguir a direcção correcta. | Pressionar as teclas para controlo da câmara, para verificar se ela segue a direcção correcta. | A câmara segue as direcções correctas. | A câmara segue as direcções correctas. |