|  |  |  |
| --- | --- | --- |
| Módulo | Inteligência Artificial | |
| Objetivo | Testar cálculo do número de ligações de um utilizador (rede nível 2) | |
| Método Teste | Manual | |
| Cenário | Teste | Resultado Esperado |
| Base de conhecimento “bd.pl” | tamanho2(pedro,N). | 1 |
| tamanho2(bruno,N). | 3 |
| tamanho2(pfq,N). | Erro - Utilizador inexistente |

|  |  |  |
| --- | --- | --- |
| Módulo | Inteligência Artificial | |
| Objetivo | Testar cálculo do número de ligações de um utilizador (rede nível 3) | |
| Método Teste | Manual | |
| Cenário | Teste | Resultado Esperado |
| Base de conhecimento “bd.pl” | tamanho3(pedro,N). | N=3 |
| tamanho3(bruno,N). | N=6 |
| tamanho3(pfq,N). | Erro - Utilizador inexistente |

|  |  |  |
| --- | --- | --- |
| Módulo | Inteligência Artificial | |
| Objetivo | Sugerir amigos para um utilizador | |
| Método Teste | Manual | |
| Cenário | Teste | Resultado Esperado |
| Base de conhecimento no gandalf (Sql server) com estrutura da “Bd.pl” | sugereAmigos(sara,L). | L=[bruno] |
| sugereAmigos(asd,L). | Erro - Utilizador inexistente |
| sugereAmigos(carlos,L). | L=[ ] |
| Módulo | Inteligência Artificial | |
| Objetivo | Determinar quem são os “maven” de um determinado assunto (tag). | |
| Método Teste | Manual | |
| Cenário | Teste | Resultado Esperado |
| Base de conhecimento “bd.pl” | maven(comida,U). | U=mario |
| maven(chelsea,U). | Erro –min utilizadores não atingido |
| maven(unix,U). | Erro – Tag não existente |

|  |  |  |
| --- | --- | --- |
| Módulo | Inteligência Artificial | |
| Objetivo | Amigos em comum entre 2 utilizadores | |
| Método Teste | Manual | |
| Cenário | Teste | Resultado Esperado |
| Base de conhecimento “bd.pl” | grafoComum(pedro,catia,G). | G=[bruno] |
| grafoComum(bruno,catia,G). | G=[ ] |
| grafoComum(bruno,tania,G). | Erro - Utilizador inexistente |

|  |  |  |
| --- | --- | --- |
| Módulo | Inteligência Artificial | |
| Objetivo | Calcular Grau médio de separação | |
| Método Teste | Manual | |
| Cenário | Teste | Resultado Esperado |
| Base de conhecimento no gandalf (Sql server) com estrutura da “Bd.pl” | grauMedio(R). | R= ~3.1 |

|  |  |  |
| --- | --- | --- |
| Módulo | Inteligência Artificial | |
| Objetivo | Determinar o caminho mais forte e o mais curto entre 2 utilizadores | |
| Método Teste | Manual | |
| Cenário | Teste | Resultado Esperado |
| Base de conhecimento no gandalf (Sql server) com estrutura da “Bd.pl” | camCurto(hugo,sara,R). | R = [hugo,catia,sara] |
| camForte(hugo,sara,R). | R = [hugo,mario,bruno,catia,sara] |
| camCurto(bruno,asd,R). | Erro - Utilizador inexistente |

|  |  |  |
| --- | --- | --- |
| Módulo | Inteligência Artificial | |
| Objetivo | Obter os amigos que tenham em comum *X tags (c/ semântica tags)* | |
| Método Teste | Manual | |
| Cenário | Teste | Resultado Esperado |
| Base de conhecimento “bd.pl” | amigosTag(bruno,[chelsea],L). | L = [catia] |
| amigosTag(bruno,[blues],L). | L = [catia] |
| amigosTag(bruno,[carros],L). | Erro - tag inexistente |

|  |  |  |
| --- | --- | --- |
| Módulo | Active Record | |
| Objetivo | Adicionar ligação entre 2 utilizadores com força de ligação e a tag de relação. | |
| Método Teste | Manual | |
| Cenário | Teste | Resultado Esperado |
| Usando DLL | AddLigacao(“Mario”, “Tiago”, “Conhecido”, 2); | Ligação Efectuada |
| AddLigacao(“Mario”, “Bruno”, “Primo”, 3); | Ligação Efectuada |
| AddLigacao(“Bruno”, “Hugo”, “Irmão”, 1); | Erro – Ligação já efectuada |

|  |  |  |
| --- | --- | --- |
| Módulo | Active Record | |
| Objetivo | Enviar Pedido de Amizade | |
| Método Teste | Manual | |
| Cenário | Teste | Resultado Esperado |
| Website | PedidoAmizade(“Hugo”, “Catia”, 5, “Namorada”); | Pedido Enviado |
| PedidoAmizade (“Leniker”, “Hugo”, “Primo”, 3); | Pedido Enviado |
| PedidoAmizade (“Tiago”, “Mario”, “Conhecido”, 1); | Erro – Amizade já efectuada |

|  |  |  |
| --- | --- | --- |
| Módulo | Active Record | |
| Objetivo | Aceitar Pedido de Amizade | |
| Método Teste | Manual | |
| Cenário | Teste | Resultado Esperado |
| Website | AceitarPedido(“Catia”, “Hugo”, 5, “Namorada”); | Pedido Aceite |
| AceitarPedido (“Mario”, “Tiago”, “Primo”, 3); | Pedido Aceite |
| AceitarPedido (“Mario”, “Tiago”, “Conhecido”, 1); | Erro – Pedido já aceite |

|  |  |  |
| --- | --- | --- |
| Módulo | Active Record | |
| Objetivo | Rejeitar Pedido de Amizade | |
| Método Teste | Manual | |
| Cenário | Teste | Resultado Esperado |
| Website | RejeitarPedido(“Catia”, “Hugo”); | Pedido Rejeitado |
| RejeitarPedido (“Mario”, “Tiago”); | Pedido Rejeitado |
| RejeitarPedido (“Mario”, “Tiago”); | Erro – Pedido já rejeitado |