



Apresentação

Trabalho de Inteligência Artificial em Engenharia Biomédica Parte 1

Alexandra Mendes A101472

Gabriela Barbosa A100463

Carmo Guimarães A100474

Marina Miranda A101474

Índice

Esta apresentação terá como intuito testar todos os predicados criados para o correto funcionamento desta primeira parte do trabalho da UC de Inteligência Artificial em Engenharia Biomédica.

- 01 Testes ao Sistema de Inferência (SI)
- 02 Testes ao Sistema de Inferência de Conjunção (SIC)
- 03 Testes ao Sistema de Inferência de Disjunção (SID)
- 04 Testes ao Calcular IMC
- 05 Teste à Evolução da Base de Conhecimento:
 - Utente
 - Registo
- 06 Testes à Involução da Base de Conhecimento
- 07 Testes aos Tipos de Conhecimento Imperfeito:
 - Conhecimento Imperfeito Incerto (TIPO I)
 - Conhecimento Imperfeito Impreciso (TIPO II)
 - Conhecimento Imperfeito Interdito (TIPO III)
- 08 Testes à Passagem de Conhecimento Perfeito Positivo para Conhecimento Perfeito Negativo e vice-versa
- 09 Testes à Alteração de Dados
- 10 Questões Finais

01 Testes ao Sistema de Inferência (SI)

1º: Testar o Conhecimento Perfeito Positivo:

```
?- si(utente(123456780, antonio, (23,10,2002), sim, guimaraes),R).  
R = verdadeiro .
```

2º: Testar o Conhecimento Perfeito Negativo:

```
?- si(registo(gmr356, (25,12,2022), 237987543, 2, 23, 1.67, 63.0),R).  
R = falso .
```

3º: Testar o Conhecimento Imperfeito:

```
?- si(utente(567235912, alexandre, (08,07,1985), sim, braga),R).  
R = desconhecido.
```

02 Testes ao Sistema de Inferência de Conjunção (SIC)

1º: Conhecimento Perfeito Positivo + Conhecimento Perfeito Positivo:

```
?- sic(utente(272731457, gabriela, (9,10,2002), nao, vale_de_cambra),  
registro(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = verdadeiro .
```

2º: Conhecimento Perfeito Positivo + Conhecimento Perfeito Negativo:

```
?- sic(utente(754137272, marina, (11,9,1986), nao, barcelos),  
registro(gmr532, (25,10,2022), 425887123, 1, 36, 1.76, 72.0),R).  
R = falso .
```

3º: Conhecimento Perfeito Positivo + Conhecimento Imperfeito:

```
?- sic(registo(gmr597, (23,9,2023), 122664691, 2, 24, 1.60, 57.3),  
utente(546734892, francisco, (23,10,1959), sim, sao_tome),R).  
R = desconhecido .
```

02 Testes ao Sistema de Inferência de Conjunção (SIC)

4º: Conhecimento Perfeito Negativo + Conhecimento Perfeito Negativo:

```
?- sic(utente(325544694, fausto, (5,5,2005), sim, faro),  
registro(gmr356, (25,12,2022), 237987543, 2, 23, 1.67, 63.0),R).  
R = falso .
```

5º: Conhecimento Perfeito Negativo + Conhecimento Imperfeito:

```
?- sic(registo(gmr532, (25,10,2022), 425887123, 1, 36, 1.76, 72.0),  
utente(472904678, lucrecia, (1,12,1985), nao, moimenta),R).  
R = falso .
```

6º: Conhecimento Imperfeito + Conhecimento Imperfeito:

```
?- sic(registo(gmr988, (29,9,2023), 947574675, 1, 45, 1.73, 70),  
registro(gmr988, (29,9,2023), 947574675, 1, 45, 1.73, 75),R).  
R = desconhecido.
```

03 Testes ao Sistema de Inferência de Disjunção (SID)

1º: Conhecimento Perfeito Positivo + Conhecimento Perfeito Positivo:

```
?- sid(utente(272731457, gabriela, (9,10,2002), nao, vale_de_cambra),  
registro(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = verdadeiro .
```

2º: Conhecimento Perfeito Positivo + Conhecimento Perfeito Negativo:

```
?- sid(utente(754137272, marina, (11,9,1986), nao, barcelos),  
registro(gmr532, (25,10,2022), 425887123, 1, 36, 1.76, 72.0),R).  
R = verdadeiro .
```

3º: Conhecimento Perfeito Positivo + Conhecimento Imperfeito:

```
?- sid(registro(gmr597, (23,9,2023), 122664691, 2, 24, 1.60, 57.3),  
utente(546734892, francisco, (23,10,1959), sim, sao_tome),R).  
R = verdadeiro .
```

03 Testes ao Sistema de Inferência de Disjunção (SID)

4º: Conhecimento Perfeito Negativo + Conhecimento Perfeito Negativo:

```
?- sid(utente(325544694, fausto, (5,5,2005), sim, faro),  
registro(gmr356, (25,12,2022), 237987543, 2, 23, 1.67, 63.0),R).  
R = falso .
```

5º: Conhecimento Perfeito Negativo + Conhecimento Imperfeito:

```
?- sid(registo(gmr532, (25,10,2022), 425887123, 1, 36, 1.76, 72.0),  
utente(472904678, lucrecia, (1,12,1985), nao, moimenta),R).  
R = desconhecido .
```

6º: Conhecimento Imperfeito + Conhecimento Imperfeito:

```
?- sid(registo(gmr988, (29,9,2023), 947574675, 1, 45, 1.73, 70),  
registro(gmr988, (29,9,2023), 947574675, 1, 45, 1.73, 75),R).  
R = desconhecido .
```

04 Testes ao Calcular IMC

1º: Testar o cálculo do IMC:

```
?- calcular_imc(473219744, IMC, Classificacao).  
IMC = 17.34046769718589,  
Classificacao = magreza_leve .
```

04 Testes ao Calcular IMC

2º: Testar o cálculo do IMC a partir do utente:

```
?- sic(utente(IdUt, alexandre, (8,7,1985), sim, lisboa),  
      calcular_imc(IdUt, IMC, Classificacao), R).  
IdUt = 567235912,  
IMC = 33.579471415683784,  
Classificacao = obesidade_grau1,  
R = verdadeiro .
```

04 Testes ao Calcular IMC

3º: Testar o cálculo do IMC a partir do registo:

```
?- sic(registo(gmr252, (19,8,2023), IdUt, 1, 38, 1.73, 100.5),  
calcular_imc(IdUt, IMC, Classificacao),R).  
IdUt = 567235912,  
IMC = 33.579471415683784,  
Classificacao = obesidade_grau1,  
R = verdadeiro .
```

05 Teste à Evolução da Base de Conhecimento: UTENTE

1º: Testar a inserção de um utente que não está na base de conhecimento:

```
?- si(utente(893052362, beatriz, (23,10,2002), sim, lisboa),R).  
R = falso .
```

```
?- evolucao(utente(893052362, beatriz, (23,10,2002), sim, lisboa)).  
true .
```

```
?- si(utente(893052362, beatriz, (23,10,2002), sim, lisboa),R).  
R = verdadeiro .
```

2º: Testar a inserção de um utente que já está na base de conhecimento:

```
?- si(utente(123123123, carminha, (13,5,1998), nao, porto),R).  
R = verdadeiro .
```

```
?- evolucao(utente(123123123, carminha, (13,5,1998), nao, porto)).  
false.
```

05 Teste à Evolução da Base de Conhecimento: UTENTE

3º: Testar a inserção de um utente com um ano de nascimento incorreto:

```
?- evolucao(utente(982174950, lucinda, (25,7,2024), sim, lisboa)).  
false.
```

4º: Testar a inserção de um utente com um mês de nascimento incorreto:

```
?- evolucao(utente(923456540, paula, (1,13,1977), sim, barcelos)).  
false.
```

5º: Testar a inserção de um utente com um dia de nascimento incorreto:

```
?- evolucao(utente(256743876, jorge, (32,1,1973), nao, barcelos)).  
false.
```

05 Teste à Evolução da Base de Conhecimento: UTENTE

6º: Testar a inserção de um utente com um historico familiar incorreto:

```
?- evolucao(utente(224537768, sara, (23,2,1973), ola, barcelos)).  
false.
```

7º: Testar a inserção de um utente com id igual ao de um utente já existente:

```
?- si(utente(123123123, carminha, (13,5,1998), nao, porto),R).  
R = verdadeiro .
```

```
?- evolucao(utente(123123123, marta, (23,2,1973), sim, barcelos)).  
false.
```

05 Teste à Evolução da Base de Conhecimento: UTENTE

8º: Testar a inserção de conhecimento positivo quando já existe conhecimento negativo sobre o utente:

```
?- si(utente(636237854, gloria, (10,2,1969), sim, cascais),R).  
R = falso .
```

```
?- evolucao(utente(636237854, gloria, (10,2,1969), sim, cascais)).  
false.
```

```
?- si(utente(636237854, gloria, (10,2,1969), sim, cascais),R).  
R = falso .
```

9º: Testar a inserção de conhecimento negativo quando já existe conhecimento positivo sobre o utente:

```
?- si(registo(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = verdadeiro .
```

```
?- evolucao(-registo(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0)).  
false.
```

```
?- si(registo(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = verdadeiro .
```

05 Teste à Evolução da Base de Conhecimento: REGISTO

1º: Testar a inserção de um registo que não está na base de conhecimento:

```
?- evolucao(registo(gmr748, (30,10,2023), 793058762, 1, 21, 1.61, 75.7)).  
true .
```

```
?- si(registo(gmr748, (30,10,2023), 793058762, 1, 21, 1.61, 75.7),R).  
R = verdadeiro .
```

2º: Testar a inserção de um registo que já está na base de conhecimento:

```
?- si(registo(gmr123, (30,9,2023), 123456780, 1, 21, 1.80, 75.7),R).  
R = verdadeiro .
```

```
?- evolucao(registo(gmr123, (30,9,2023), 123456780, 1, 21, 1.80, 75.7)).  
false.
```

05 Teste à Evolução da Base de Conhecimento: REGISTO

3º: Testar a inserção de um registo com um sexo incorreto:

```
?- evolucao(registo(gmr516, (30,10,2022), 267898870, 3, 23, 1.80, 75.7)).  
false.
```

4º: Testar a inserção de um registo com uma idade incorreta:

```
?- evolucao(registo(gmr256, (30,10,2022), 189265780, 1, -1, 1.80, 75.7)).  
false.
```

5º: Testar a inserção de um utente com um ano de registo incorreto:

```
?- evolucao(registo(gmr356, (30,10,2010), 356898780, 2, 23, 1.80, 75.7)).  
false.
```

05 Teste à Evolução da Base de Conhecimento: REGISTO

6º: Testar a inserção de um registo com mês de registo incorreto:

```
?- evolucao(registo(gmr014, (30,13,2021), 671838980, 2, 23, 1.80, 75.7)).  
false.
```

7º: Testar a inserção de um registo com dia de registo incorreta:

```
?- evolucao(registo(gmr230, (35,1,2021), 645838980, 2, 23, 1.80, 67.8)).  
false.
```

8º: Testar a inserção de um utente com uma altura incorreta:

```
?- evolucao(registo(gmr500, (12,1,2021), 567876580, 2, 23, 0.1, 67.8)).  
false.
```

05 Teste à Evolução da Base de Conhecimento: REGISTO

9º: Testar a inserção de um registo com um o peso incorreto:

```
?- evolucao(registo(gmr435, (12,1,2021), 567878980, 2, 23, 1.43, 5)).  
false.
```

10º: Testar a inserção de um registo com id igual ao de um registo já existente:

```
?- si(registo(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = verdadeiro .
```

```
?- evolucao(registo(gmr321, (12,1,2021), 567878980, 2, 23, 1.43, 57)).  
false.
```

05 Teste à Evolução da Base de Conhecimento: REGISTO

8º: Testar a inserção de conhecimento positivo quando já existe conhecimento negativo sobre o registo:

```
?- si(registo(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = verdadeiro .
```

```
?- evolucao(-registro(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0)).  
false.
```

```
?- si(registo(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = verdadeiro .
```

9º: Testar a inserção de conhecimento negativo quando já existe conhecimento positivo sobre o registo:

```
?- si(registo(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = verdadeiro .
```

```
?- evolucao(-registro(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0)).  
false.
```

```
?- si(registo(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = verdadeiro .
```

06 Teste à Involução da Base de Conhecimento

1º: Testar a involução de um utente que está na base de conhecimento:

```
?- si(utente(473219744, lara, (20,3,1960), sim, vila_do_conde),R).  
R = verdadeiro .  
  
?- involucao(utente(473219744, lara, (20,3,1960), sim, vila_do_conde)).  
true .  
  
?- si(utente(473219744, lara, (20,3,1960), sim, vila_do_conde),R).  
R = falso .
```

2º: Testar a involução de um utente que não está na base de conhecimento:

```
?- si(utente(245165729, luisa, (8,2,2003), sim, lisboa),R).  
R = falso .  
  
?- involucao(utente(245165729, luisa, (8,2,2003), sim, lisboa)).  
false.
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Incerto (Tipo I)

1º: Testar um exemplo do conhecimento imperfeito incerto:

```
?- si(utente(567235912, alexandre, (08,07,1985), sim, barcelos),R).  
R = desconhecido.
```

```
?- si(utente(567235912, alexandre, (08,07,1985), sim, esposende),R).  
R = desconhecido.
```

```
?- si(utente(567235912, alexandre, (08,07,1985), sim, braga),R).  
R = desconhecido.
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Incerto (Tipo I)

2º: Testar a passagem de conhecimento imperfeito incerto para conhecimento perfeito positivo (utente):

```
?- si(utente(567235912, alexandre, (08,07,1985), sim, braga),R).  
R = desconhecido.
```

```
?- alterarmorada(567235912,braga).  
true .
```

```
?- si(utente(567235912, alexandre, (08,07,1985), sim, braga),R).  
R = verdadeiro .
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Incerto (Tipo I)

3º: Testar a passagem de conhecimento imperfeito incerto para conhecimento perfeito positivo (registo):

```
?- si(registo(gmr654, (26,9,2023), 156987245, 1, 30, 1.70, 58), R).  
R = desconhecido.
```

```
?- alterarsexo(gmr654, 1).  
true .
```

```
?- si(registo(gmr654, (26,9,2023), 156987245, 1, 30, 1.70, 58), R).  
R = verdadeiro .
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Incerto (Tipo I)

4º: Testar a passagem de conhecimento perfeito positivo para conhecimento imperfeito incerto (utente):

```
?- si(utente(754137272, marina, (11,9,1986), nao, barcelos),R).  
R = verdadeiro .
```

```
?- perfeito_incerto_utente(754137272, marina, (11,9,1986), nao, barcelos).  
true .
```

```
?- si(utente(754137272, beatriz, (11,9,1986), nao, barcelos),R).  
R = desconhecido.
```

```
?- si(utente(754137272, sofia, (11,9,1986), nao, barcelos),R).  
R = desconhecido.
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Incerto (Tipo I)

5º: Testar a passagem de conhecimento perfeito positivo para conhecimento imperfeito incerto (registo):

```
?- si(registo(gmr175, (12,2,2023), 754137272, 2, 37, 1.64, 85.4),R).  
R = verdadeiro .
```

```
?- perfeito_incerto_registro(gmr175, (12,2,2023), 754137272, 2, 37, 1.64, 85.4).  
true .
```

```
?- si(registo(gmr175, (12,2,2023), 754137272, 2, 37, 1.64, 85.4),R).  
R = desconhecido.
```

```
?- si(registo(gmr175, (12,2,2023), 754137272, 1, 37, 1.64, 85.4),R).  
R = desconhecido.
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Incerto (Tipo I)

6º: Testar a inserção de conhecimento imperfeito incerto (utente):

```
?- inserir_incerto_utente(121567245, gabriela, (9,10,2002), nao, porto).  
true .
```

```
?- si(utente(121567245, gabriela, (9,10,2002), nao, lisboa),R).  
R = desconhecido.
```

```
?- si(utente(121567245, gabriela, (9,10,2002), nao, esposende),R).  
R = desconhecido.
```

```
?- si(utente(121567245, gabriela, (9,10,2002), nao, vale_de_cambra),R).  
R = desconhecido.
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Incerto (Tipo I)

7º: Testar a inserção de conhecimento imperfeito incerto (registro):

```
?- inserir_incerto_registro(gmr175, (12,2,2023), 754137272, 1, 37, 1.64, 85.4).  
true .
```

```
?- si(registo(gmr175, (12,2,2023), 754137272, 1, 37, 1.64, 90.0),R).  
R = desconhecido.
```

```
?- si(registo(gmr175, (12,2,2023), 754137272, 1, 37, 1.64, 95.0),R).  
R = desconhecido.
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Impreciso (Tipo II)

1º: Testar um exemplo do conhecimento imperfeito impreciso:

```
?- si(utente(546734892, francisco, (23,10,1959), sim, sao_tome),R).  
R = desconhecido.
```

```
?- si(utente(546734892, francisco, (23,10,1960), sim, sao_tome),R).  
R = desconhecido.
```

```
?- si(utente(546734892, francisco, (23,10,1961), sim, sao_tome),R).  
R = falso .
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Impreciso (Tipo II)

2º: Testar a passagem de conhecimento imperfeito impreciso para conhecimento perfeito positivo:

```
?- si(utente(546734892, francisco, (23,10,1960), sim, sao_tome),R).  
R = desconhecido.
```

```
?- impreciso_perfeito_utente(utente(546734892, francisco, (23,10,1960), sim, sao_tome)).  
true .
```

```
?- si(utente(546734892, francisco, (23,10,1960), sim, sao_tome),R).  
R = verdadeiro .
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Impreciso (Tipo II)

4º: Testar a passagem de conhecimento perfeito positivo para conhecimento imperfeito impreciso (utente):

```
?- si(utente(123456780, antonio, (23,10,2002), sim, guimaraes),R).  
R = verdadeiro .
```

```
?- perfeito_impreciso([utente(123456780, antonio, (23,10,2002), sim, guimaraes),utente(123456780, antonio, (23,10,2002), sim, lisboa),utente(123456780, antonio, (23,10,2002), sim, porto)]).  
true .
```

```
?- si(utente(123456780, antonio, (23,10,2002), sim, guimaraes),R).  
R = desconhecido.
```

```
?- si(utente(123456780, antonio, (23,10,2002), sim, lisboa),R).  
R = desconhecido.
```

```
?- si(utente(123456780, antonio, (23,10,2002), sim, porto),R).  
R = desconhecido.
```

```
?- si(utente(123456780, antonio, (23,10,2002), sim, braga),R).  
R = falso
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Impreciso (Tipo II)

5º: Testar a passagem de conhecimento perfeito positivo para conhecimento imperfeito impreciso (registo):

```
?- si(registo(gmr232, (5,11,2022), 473219744, 2, 63, 1.74, 52.5),R).  
R = verdadeiro .
```

```
?- perfeito_impreciso([registro(gmr232, (5,11,2022), 473219744, 2, 63, 1.74, 52.5), registro(gmr232, (5,11,2022),  
473219744, 1, 63, 1.74, 52.5)]).  
true .
```

```
?- si(registo(gmr232, (5,11,2022), 473219744, 2, 63, 1.74, 52.5),R).  
R = desconhecido.
```

```
?- si(registo(gmr232, (5,11,2022), 473219744, 1, 63, 1.74, 52.5),R).  
R = desconhecido.
```

```
?- si(registo(gmr232, (5,11,2022), 473219744, 3, 63, 1.74, 52.5),R).  
R = falso
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Impreciso (Tipo II)

6º: Testar a inserção de conhecimento imperfeito impreciso (utente):

```
?- inserir_conhecimento_impreciso([utente(175281882, lucia, (1,12,1985), nao, viseu),utente(175281882, lucia, (1,12,1985), sim, viseu)]).  
true .
```

```
?- si(utente(175281882, lucia, (1,12,1985), nao, viseu),R).  
R = desconhecido.
```

```
?- si(utente(175281882, lucia, (1,12,1985), sim, viseu),R).  
R = desconhecido.
```

```
?- si(utente(175281882, lucia, (1,12,1985), ola, viseu),R).  
R = falso .
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Impreciso (Tipo II)

7º: Testar a inserção de conhecimento imperfeito impreciso (registro):

```
?- inserir_conhecimento_impreciso([registro(gmr019, (23,1,2021), 187290376, 2, 30, 1.73, 68.8), registro(gmr019, (23,1,2021), 187290376, 2, 31, 1.73, 68.8), registro(gmr019, (23,1,2021), 187290376, 2, 32, 1.73, 68.8)]).  
true .  
  
?- si(registro(gmr019, (23,1,2021), 187290376, 2, 30, 1.73, 68.8),R).  
R = desconhecido.  
  
?- si(registro(gmr019, (23,1,2021), 187290376, 2, 31, 1.73, 68.8),R).  
R = desconhecido.  
  
?- si(registro(gmr019, (23,1,2021), 187290376, 2, 32, 1.73, 68.8),R).  
R = desconhecido.  
  
?- si(registro(gmr019, (23,1,2021), 187290376, 2, 33, 1.73, 68.8),R).  
R = falso .
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Interdito (Tipo III)

1º: Testar um exemplo do conhecimento imperfeito interdito:

```
?- si(utente(277631457, mario, (09,10,1967), nao, perelhal),R).  
R = desconhecido.
```

```
?- si(utente(277631457, mario, (09,10,1967), nao, lisboa),R).  
R = desconhecido.
```

07 Testes aos tipos de Conhecimento Imperfeito

- Conhecimento Imperfeito Interdito (Tipo III)

2º: Testar a passagem de conhecimento imperfeito interdito para conhecimento perfeito positivo:

```
?- si(utente(277631457, mario, (09,10,1967), nao, perelhal),R).  
R = desconhecido.
```

```
?- si(utente(277631457, mario, (09,10,1967), nao, lisboa),R).  
R = desconhecido.
```

```
?- alterarmorada(277631457,porto).  
false.
```

```
?- si(utente(277631457, mario, (09,10,1967), nao, porto),R).  
R = desconhecido.
```

O conhecimento imperfeito impreciso é desconhecido e sempre assim será, logo não é possível passá-lo para conhecimento perfeito positivo.

08

Testes à Passagem de Conhecimento Perfeito Positivo para Conhecimento Perfeito Negativo

```
?- si(utente(272731457, gabriela, (9,10,2002), nao, vale_de_cambra),R).  
R = verdadeiro .
```

```
?- si(registo(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = verdadeiro .
```

```
?- pos_neg(272731457,gmr321).  
true .
```

```
?- si(utente(272731457, gabriela, (9,10,2002), nao, vale_de_cambra),R).  
R = falso .
```

```
?- si(registo(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = falso .
```

08 Testes à Passagem de Conhecimento Perfeito Negativo para Conhecimento Perfeito Positivo

```
?- si(utente(272731457, gabriela, (9,10,2002), nao, vale_de_cambra),R).  
R = falso .
```

```
?- si(registo(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = falso .
```

```
?- neg_pos(272731457,gmr321).  
true .
```

```
?- si(utente(272731457, gabriela, (9,10,2002), nao, vale_de_cambra),R).  
R = verdadeiro .
```

```
?- si(registo(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = verdadeiro .
```

09 Testes à Alteração de Dados

1º: Testar a alteração do id de um utente:

```
?- si(utente(135246978, carlos, (12,3,1980), nao, vila_verde),R).  
R = verdadeiro .
```

```
?- alteraridut(135246978,198267356).  
true .
```

```
?- si(utente(IdUt, carlos, (12,3,1980), nao, vila_verde),R).  
IdUt = 198267356,  
R = verdadeiro .
```

09 Testes à Alteração de Dados

2º: Testar a alteração do histórico familiar de um utente:

```
?- si(utente(567235912, alexandre, (8,7,1985), sim, lisboa),R).  
R = verdadeiro .
```

```
?- alterarhistorico(567235912,nao).  
true .
```

```
?- si(utente(567235912, alexandre, (8,7,1985), Historico_familiar, lisboa),R).  
Historico_familiar = nao,  
R = verdadeiro .
```

09 Testes à Alteração de Dados

3º: Testar a alteração da morada de um utente:

```
?- si(utente(346502568, manuel, (1,6,1993), sim, leiria),R).  
R = verdadeiro .
```

```
?- alterarmorada(346502568, lisboa).  
true .
```

```
?- si(utente(346502568, manuel, (1,6,1993), sim, Morada),R).  
Morada = lisboa,  
R = verdadeiro .
```

09 Testes à Alteração de Dados

4º: Testar a alteração do id de um registo:

```
?- si(registo(gmr567, (30,9,2023), 135246978, 1, 43, 1.76, 93.5),R).  
R = verdadeiro .
```

```
?- alteraridregistro(gmr567,gmr981).  
true .
```

```
?- si(registo(Idreg, (30,9,2023), 135246978, 1, 43, 1.76, 93.5),R).  
Idreg = gmr981,  
R = verdadeiro .
```

09 Testes à Alteração de Dados

5º: Testar a alteração da idade de um registo:

```
?- si(registo(gmr321, (2,10,2022), 272731457, 2, 21, 1.59, 60.0),R).  
R = verdadeiro .  
  
?- alteraridade(gmr321,25).  
true .  
  
?- si(registo(gmr321, (2,10,2022), 272731457, 2, Idade, 1.59, 60.0),R).  
Idade = 25,  
R = verdadeiro .
```

09 Testes à Alteração de Dados

6º: Testar a alteração da altura de um registo:

```
?- si(registo(gmr396, (26,9,2023), 346502568, 1, 30, 1.90, 150.4),R).  
R = verdadeiro .  
  
?- alteraraltura(gmr396,1.80).  
true .  
  
?- si(registo(gmr396, (26,9,2023), 346502568, 1, 30, Altura, 150.4),R).  
Altura = 1.8,  
R = verdadeiro .
```

09 Testes à Alteração de Dados

7º: Testar a alteração do peso de um registo:

```
?- si(registo(gmr734, (31,1,2022), 123123123, 2, 25, 1.70, 77.8),R).  
R = verdadeiro .  
  
?- alterarpeso(gmr734,72.3).  
true .  
  
?- si(registo(gmr734, (31,1,2022), 123123123, 2, 25, 1.70, Peso),R).  
Peso = 72.3,  
R = verdadeiro .
```

10 Questões Finais

1º: Quais os paradigmas de aprendizagem a utilizar para resolver o problema?

Machine e supervised learning.

2º: O tipo problema apresentado pelos dados suscita uma resolução contínua, discreta, ou outra?

Discreta.



FIM

Trabalho de Inteligência Artificial em Engenharia Biomédica Parte 1

Alexandra Mendes A101472

Gabriela Barbosa A100463

Carmo Guimarães A100474

Marina Miranda A101474