



Universidade do Minho
Escola de Engenharia

Trabalho Prático - Grupo 32

Vacinação da População Portuguesa no contexto COVID-19 - 2^a Fase

Sistemas de Representação de Conhecimento e Raciocínio

Bruno Filipe de Sousa Dias A89583

Luís Enes Sousa A89597

Pedro Miguel de Soveral Pacheco Barbosa A89529



3 de maio de 2021

Conteúdo

1	Resumo	1
2	Introdução	2
3	Preliminares	2
3.1	Sistema de Inferência	2
3.2	Conhecimento Imperfeito	3
4	Descrição do trabalho e Análise de Resultados	3
4.1	Base de Conhecimento	3
4.2	Conhecimento Positivo	6
4.3	Conhecimento Negativo	6
4.3.1	Negação Forte	6
4.3.2	Conhecimento Negativo por Falhas	6
4.4	Conhecimento Imperfeito	7
4.5	Valor Nulo Tipo I - Conhecimento Incerto	7
4.6	Valor Nulo Tipo II - Conhecimento Impreciso	8
4.7	Valor Nulo Tipo III - Conhecimento Interdito	9
5	Evolução do Conhecimento	11
6	Funcionalidades Adicionais	18
6.1	Atualizar Conhecimento Imperfeito	18
6.1.1	Atualizar Conhecimento Incerto	18
6.1.2	Atualizar Conhecimento Impreciso	20
7	Conclusão	21

1 Resumo

Este relatório foi elaborado no âmbito da unidade curricular de Sistemas de Representação de Conhecimento e Raciocínio em conjunto com a resolução da segunda fase do trabalho de grupo prático proposto pelos docentes da unidade curricular. A realização deste exercício pretende motivar os alunos para a utilização da linguagem de programação PROLOG, no âmbito da representação de conhecimento e construção de mecanismos de raciocínio para a resolução de problemas.

Primeiramente foi feita uma introdução desta segunda parte do projeto, seguida da explicação do raciocínio utilizado pelo grupo tanto para a resolução das funcionalidades especificamente pedidas no enunciado como das extras adicionadas. Por último, foi feita uma apreciação crítica da globalidade do trabalho.

2 Introdução

Este relatório surge na elaboração do exercício em grupo proposto na unidade curricular de Sistemas de Representação de Conhecimento e Raciocínio. Este exercício tem como base a programação em lógica cujo objetivo é o desenvolvimento de um programa assente em factos e em predicados que, por sua vez, estão associados a factos e regras. Assim, foi utilizada a linguagem de programação PROLOG para ser possível atingir esse mesmo objetivo.

O trabalho prático consiste em desenvolver um sistema de representação de conhecimento e raciocínio com capacidade para caracterizar um universo de discurso na área da vacinação da população portuguesa no contexto COVID-19 que estamos a viver.

Nesta fase do trabalho prático o principal objetivo foi a correta representação de conhecimento positivo e negativo bem como de conhecimento imperfeito através da utilização de valores nulos de todos os diferentes tipos estudados ao longo das aulas práticas da unidade curricular. O sistema desenvolvido deverá ainda ser capaz de inserir e remover conhecimento do mesmo através de invariantes apropriados e ainda permitir a evolução e involução do conhecimento.

3 Preliminares

3.1 Sistema de Inferência

Apesar de parte do sistema de inferência vir da fase anterior do trabalho prática faz mais sentido explica-la em maior detalhe agora visto que esta se encontra mais completa e detalhada. Através da representação tanto de conhecimento completo como de conhecimento incompleto conseguimos atingir uma representação muito mais próxima daquilo que é a realidade. Deste modo, através do predicado *si* existem três tipos diferentes de interpretações de uma *Questao*, sendo eles:

- **Verdadeiro**

Quando o conhecimento existir e se provar na nossa base de conhecimento.

- **Falso**

Quando o conhecimento não existir e nem for passível de ser provado na nossa base de conhecimento.

- **Desconhecido**

Quando o conhecimento não for nem passível nem não passível de ser provado na nossa base de conhecimento.

A definição do predicado *si* é a seguinte:

```

% -----
% Predicado que verifica se o conhecimento existe na base de conhecimento.
% Extensão do predicado si: Questao, Resposta -> {V,F}.
% -----

si(Questao, verdadeiro) :- Questao.
si(Questao, falso) :- ~Questao.
si(Questao, desconhecido) :- nao(Questao), nao(~Questao).

```

Figura 1: Predicado si

3.2 Conhecimento Imperfeito

Em relação ao **Conhecimento Imperfeito** que é o conhecimento que pretendemos abordar nesta fase do projeto este pode ser de três diferentes tipos.

- **Conhecimento Impreciso**

Representa valores nulos pertencentes a um determinado conjunto.

- **Conhecimento Incerto**

Representa valores nulos que são desconhecidos, não pertencendo a nenhum conjunto de valores.

- **Conhecimento Interdito**

Representa valores nulos desconhecidos e sobre os quais não há permissão de conhecer.

4 Descrição do trabalho e Análise de Resultados

4.1 Base de Conhecimento

A base de conhecimento define a base de dados ou de conhecimento adquirido sobre um determinado assunto. Nesta fase a mesma foi aumentada de forma a não conter apenas conhecimento completo mas também conhecimento incompleto sobre a realidade. Desta forma, a nossa base de conhecimento é composta por **Utentes**, **Staffs**, **Centros de Saúde** e **Vacinações Covid** e apresenta-se da seguinte forma:

• Utentes

```
?- listing(utente).
:- dynamic utente/10.

utente(1, 12345678901, 'Joao da Costa e Campos', 1935-5-3, 'jcc@srcr.pt', 911111111, 'Rua das Adegas Felizes, 12, 1A° Cave', 'Reformado', ['Doencas Reumaticas', 'Cancro'], 5).
utente(17, 22990000000, 'Almira da Cruz Santos', 1910-5-1, 'adcs@srcr.pt', 928888888, 'Rua da Alegria, 1', 'Reformada', [], 1).
utente(10, 58210227854, 'Maria Beatriz Araujo Lacerda', 1989-7-27, 'mbal@srcr.pt', 921111111, 'Rua do Galo Barcelonete, 136', 'Cabeleireiro', ['Doenca Pulmonar Obstrutiva'], 4).
utente(14, 88888855555, 'Luis Parente Pacheco Martins', 1964-8-4, 'lppm@srcr.pt', 925555555, 'Rua dos Cabecos Lamparinos, 80', 'Eletricista', ['Doencas Cardiovasculares'], 4).
utente(4, 23345886912, 'Jesualdo Peza-Mor', 1963-11-14, 'jpm@srcr.pt', 914444444, 'Estrada do Sossego, Km10', 'Enfermeiro', ['Doencas Reumaticas', 'Colesterol'], 2).
utente(15, 10000100201, 'Bruno Felipe Enes Diaz', 2000-11-21, 'bfed@srcr.pt', 926666666, 'Rua dos Amareiros Ferrenhos, 137', 'Medico', [], 1).
utente(16, 93330201456, 'Pedro Miguel Ferreira Lemos', 1980-9-26, 'pmfl@srcr.pt', 927777777, 'Rua dos Maçaricos Portenhos, 154', 'Militar', ['Doencas Cardiovasculares', 'Visao', 'Hipertensao'], 4).
utente(2, 58203459103, 'Josefina Vivida da Paz', 1958-12-2, 'jvp@srcr.pt', 912222222, 'Avenida dos Castros Reais, 122, 3A°E', 'Professor', ['Asma', 'Diabetes'], 4).
utente(7, 78990546351, 'Carminho Cunha Bastos', 1960-1-1, 'ccb@srcr.pt', 917777777, 'Rua do Mus-Vitalis, 56, R/C', 'Catequista', ['Parkinson', 'Osteoporose'], 2).
utente(9, 9876543212, 'Maria Quintas Barros', 1950-12-31, 'mqb@srcr.pt', 919999999, 'Rua do Frago, 44', 'Piloto de Avioes', ['Diabetes', 'Hipertensao', 'Visao'], 5).
utente(11, 1928374657, 'Ana Teresa Gao Gomes', 1947-2-12, 'atgg@srcr.pt', 922222222, 'Rua da Vuvuzela, 158', 'Youtuber', ['Alzheimer', 'Asma'], 5).
utente(3, 98437219304, 'Ana Santa do Carmo', 1997-8-21, 'asc@srcr.pt', 913333333, 'Travessa do Jacob, 21', 'Estudante', [], 3).
utente(8, 55599911102, 'Francisco Correia Franco', 1999-4-3, 'fcf@srcr.pt', 918888888, 'Rua do Povo, 152', 'Estudante', [], 1).
utente(12, 34543678769, 'Joao Manuel Peixe dos Santos', 1978-6-18, 'jmps@srcr.pt', 923333333, 'Rua dos Peixotos, 61, 4A°D', 'Contabilista', ['Obesidade'], 3).
utente(5, 67483900221, 'Maria da Trindade Pascoal', 1985-5-5, 'mtp@srcr.pt', 915555555, 'Rua das Adegas da Rua, 15, 10 Esq/T', 'Historiador', ['Visao'], 1).
utente(6, 3454367887, 'Florindo Teixo Figueirinha', 1974-4-25, 'ftf@srcr.pt', 916666666, 'Autodromo das Vagas, Garagem 123', 'Lojista', ['Hipertensao'], 3).
utente(13, 21114599083, 'Paulo Nobre Sousa', 1994-10-10, 'pns@srcr.pt', 924444444, 'Rua dos Moveis Cerrados, 29', 'Engenheiro de Polimeros', [], 5).
utente(20, 19283018273, 1003345901, 'Francisco Correia Antonieto Franco', 1950-3-5, 'fcf@srcr.pt', 930000000, 'Rua das Amoreiras, 152', 'Reformado', ['Osteoporose', 'Parkinson'], 2).
utente(21, 34543434567, 'Antonio Manuel Cascao Xisto', 1968-3-1, 'amcx@srcr.pt', 931111111, 932222222, 'Largo dos Peoes, 25', 'Pedreiro', ['Asma', 'Hipertensao'], 4).
utente(18, nissDesconhecido, 'Pedro Emanuel Da Silva Oliveira', 1970-11-8, 'peso@srcr.pt', 929999999, 'Travessa das Avestruzes, 32', 'Padre', ['Diabetes', 'Cancro'], 3).
utente(19, 10022993871, 'Tiago Pinto Quintas Barros', 2001-1-5, 'tpqb@srcr.pt', telemovelDesconhecido, 'Rua de Frago, 44', 'Estudante', [], 1).
utente(22, 43567859643, 'Quintino Moreira Teixeira Crafo', dataInterdita, 'qmtg@srcr.pt', 933333333, 'Rua dos Patos Cegos, 1', 'Pescador', [], 5).
utente(23, segurancaSocialInterdita, 'William Carvalho Santos Silva', 1980-4-3, 'wcss@srcr.pt', 934444444, 'Rua do Alecrim, 2', 'Porteiro', ['Asma'], 3).

true.
```

Figura 2: Listing utentes

• Centros de Saúde

```
?- listing(centro_saude).
:- dynamic centro_saude/5.

centro_saude(1, 'Centro de Saude Mem Martins', 'Rua Pedro Bispo, 725', 951111111, 'csmm@srcr.pt').
centro_saude(2, 'Centro de Saude Chelas', 'Rua Samuel Mira, 666', 952222222, 'csc@srcr.pt').
centro_saude(3, 'Centro de Saude SÃo VÃtor', 'Rua dos Cabecudos, 80', 953333333, 'cssv@srcr.pt').
centro_saude(4, 'Centro de Saude Amares', 'Rua da Lagars, 137', 954444444, 'csa@srcr.pt').
centro_saude(5, 'Centro de Saude Viena do Forte', 'Rua Anthony Feijo, 154', 955555555, 'csvf@srcr.pt').
centro_saude(6, 'Centro de Saude Viana do Castelo', 'Rua Senhora da Agonia, 8', 956666666, {'csvc@srcr.pt', 'cvsc@srcr.pt'}).
centro_saude(7, 'Centro de Saude Sao Jorge', 'Travessa da Ilha das Flores, 229', {957777777, 958888888}, {'cssj@srcr.pt'}).
centro_saude(8, nomeDesconhecido, 'Rua das Agrelas, 8', 959999999, 'aaaa@srcr.pt').
centro_saude(9, 'Centro de Saude Olivais Sul', 'Rua dos Piriqitos, 50', 960000000, emailDesconhecido).
centro_saude(10, nomeInterdito, 'Travessa das Picaretas, 10', 961111111, 'bbbb@srcr.pt').
centro_saude(11, 'Centro de Saude Odivelas', 'Rua Sargento Coronel Frances, 4', numeroInterdito, 'csodv@srcr.pt').

true.
```

Figura 3: Listing centros de saúde

- Staffs

```
?- listing(staff).
:- dynamic staff/4.

staff(1, 5, 'Alvaro de Campos', 'adc@srcr.pt').
staff(2, 4, 'Joaquim Arnaldo Fernandes', 'jaf@srcr.pt').
staff(3, 2, 'Tiburcio Mantorras', 'tm@srcr.pt').
staff(4, 1, 'Jaime Oliveira', 'jo@srcr.pt').
staff(5, 3, 'Benjamin Otavio Teixeira', 'bot@srcr.pt').
staff(6, 5, 'Sergio Batedor Oliveira', 'sbo@srcr.pt').
staff(7, [2, 4], 'Joaquim Alberto Incrivel Morais', 'jaim@srcr.pt').
staff(8, 1, 'Tito Cerqueira Andrade Vieira', {'tcav@srcr.pt', 'vact@srcr.pt'}).
staff(9, centroDesconhecido, 'Pedro Manuel Marques Mota', 'pmmm@srcr.pt').
staff(10, 2, nomeStaffDesconhecido, 'cccc@srcr.pt').
staff(11, centroInterdito, 'Pedro Pereira Pauleta Aveiro', 'pppa@srcr.pt').
staff(12, 2, nomeStaffInterdito, 'dddd@srcr.pt').

true.
```

Figura 4: Listing staffs

- Vacinações Covid

```
?- listing(vacinacao_Covid).
:- dynamic vacinacao_Covid/5.

vacinacao_Covid(1, 1, 2021-1-5, 'Pfizer', 1).
vacinacao_Covid(1, 17, 2021-1-5, 'Pfizer', 1).
vacinacao_Covid(1, 10, 2021-1-5, 'Pfizer', 1).
vacinacao_Covid(1, 14, 2021-1-5, 'Pfizer', 1).
vacinacao_Covid(6, 1, 2021-1-19, 'Pfizer', 2).
vacinacao_Covid(2, 2, 2021-1-20, 'Pfizer', 1).
vacinacao_Covid(3, 7, 2021-5-24, 'Astrazeneca', 1).
vacinacao_Covid(3, 7, 2021-5-15, 'Astrazeneca', 2).
vacinacao_Covid(6, 8, 2021-8-15, 'Astrazeneca', 1).
vacinacao_Covid(2, 6, 2021-6-1, 'Pfizer', 1).
vacinacao_Covid(2, 6, 2021-6-16, 'Pfizer', 2).
vacinacao_Covid(1, 13, 2021-7-31, 'Astrazeneca', 1).
vacinacao_Covid(6, 21, {2021-3-4, 2021-3-10}, 'Pfizer', 1).
vacinacao_Covid(3, 20, 2021-5-5, {'Pfizer', 'Astrazeneca'}, 2).
vacinacao_Covid(2, utenteDesconhecido, 2021-5-7, 'Pfizer', 1).
vacinacao_Covid(3, 18, 2021-1-10, vacinaDesconhecida, 2).
vacinacao_Covid(5, utenteInterdito, 2021-3-20, 'Astrazeneca', 1).
vacinacao_Covid(3, 19, 2021-2-20, vacinaInterdita, 1).

true.
```

Figura 5: Listing vacinações covid

4.2 Conhecimento Positivo

O **Conhecimento Positivo** é representado como na povoação inicial da nossa base de conhecimento referente à primeira fase do projeto prático como exemplificado de seguida:

```
utente(5, 67483900221, 'Maria da Trindade Pascoal', 1985-05-05, 'mtp@srcr.pt', 915555555, 'Rua das Adegas da Rua, 15, 10 Esq/T', 'Historiador', ['Visao'], 1).
utente(6, 34554367887, 'Florindo Teixo Figueirinha', 1974-04-25, 'ftf@srcr.pt', 916666666, 'Autodromo das Vagas, Garagem 123', 'Lojista', ['Hipertensao'], 3).
utente(13, 21114599083, 'Paulo Nobre Sousa', 1994-10-10, 'pns@srcr.pt', 924444444, 'Rua dos Moveis Cerrados, 29', 'Engenheiro de Polimeros', [], 5).
```

Figura 6: Exemplo de Conhecimento Positivo

4.3 Conhecimento Negativo

Por sua vez o **Conhecimento Negativo** pode ser representado de duas formas distintas, através de uma **Negação Forte** ou através de um **Conhecimento Negativo por Falhas**.

4.3.1 Negação Forte

```
-utente(Id,Nss,N,D,E,T,M,P,DC,CS) :- nao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)),
                                     nao(excecao(utente(Id,Nss,N,D,E,T,M,P,DC,CS))).

-centro_saude(Id,N,M,T,E) :- nao(centro_saude(Id,N,M,T,E)),
                             nao(excecao(centro_saude(Id,N,M,T,E))).

-staff(Id,Idcs,N,E) :- nao(staff(Id,Idcs,N,E)),
                      nao(excecao(staff(Id,Idcs,N,E))).

-vacinacao_Covid(Ids,Idu,D,V,T) :- nao(vacinacao_Covid(Ids,Idu,D,V,T)),
                                   nao(excecao(vacinacao_Covid(Ids,Idu,D,V,T))).
```

Figura 7: Exemplos de Conhecimento Negativo por Falhas

4.3.2 Conhecimento Negativo por Falhas

```
utente(5, 67483900221, 'Maria da Trindade Pascoal', 1985-05-05, 'mtp@srcr.pt', 915555555, 'Rua das Adegas da Rua, 15, 10 Esq/T', 'Historiador', ['Visao'], 1).
utente(6, 34554367887, 'Florindo Teixo Figueirinha', 1974-04-25, 'ftf@srcr.pt', 916666666, 'Autodromo das Vagas, Garagem 123', 'Lojista', ['Hipertensao'], 3).
utente(13, 21114599083, 'Paulo Nobre Sousa', 1994-10-10, 'pns@srcr.pt', 924444444, 'Rua dos Moveis Cerrados, 29', 'Engenheiro de Polimeros', [], 5).
```

Figura 8: Exemplos de Negação Forte

4.4 Conhecimento Imperfeito

É através do **Conhecimento Imperfeito** que é possível representar os diferentes tipos de valores nulos que vão aproximar a nossa base de conhecimento ainda mais daquilo que é a realidade. Desta forma, temos vários tipos diferentes de valores nulos:

4.5 Valor Nulo Tipo I - Conhecimento Incerto

Para representar o **Conhecimento Incerto** começamos pelo predicado, onde declaramos uma ou mais das suas variáveis desconhecidas e após isso declaramos a exceção do mesmo. Alguns exemplos de Conhecimento Incerto utilizados no nosso trabalho prático são:

Para os *Utentes* utilizamos um número de Segurança Social e um número de Telemóvel desconhecidos.

```
% Utente com um número de Segurança Social desconhecido.
utente(8, nissDesconhecido, 'Pedro Emanuel Da Silva Oliveira', 1970-11-08, 'peso@srcr.pt', 929999999, 'Travesa das Avestruzes, 32', 'Padre', ['Diabetes', 'Cancro'], 3).
excecao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)) :- utente(Id,nissDesconhecido,N,D,E,T,M,P,DC,CS).

% Utente com um número de telemóvel desconhecido.
utente(9, 10022993871, 'Tiago Pinto Quintas Barros', 2001-01-05, 'tpqb@srcr.pt', telemovelDesconhecido, 'Rua de Fragoso, 44', 'Estudante', [], 1).
excecao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)) :- utente(Id,Nss,N,D,E,telemovelDesconhecido,M,P,DC,CS).
```

Figura 9: Exemplos de Conhecimento Incerto Utente

Para os *Centros de Saúde* utilizamos um Nome e um Email desconhecidos.

```
% Centro de Saúde com um nome desconhecido.
centro_saude(8, nomeDesconhecido, 'Rua das Agrelas, 8', 959999999, 'aaaa@srcr.pt').
excecao(centro_saude(Id,N,M,T,E)) :- centro_saude(Id,nomeDesconhecido,M,T,E).

% Centro de Saúde com um email desconhecido.
centro_saude(9, 'Centro de Saude Olivais Sul', 'Rua dos Piriquitos, 50', 960000000, emailDesconhecido).
excecao(centro_saude(Id,N,M,T,E)) :- centro_saude(Id,N,M,T,emailDesconhecido).
```

Figura 10: Exemplos de Conhecimento Incerto Centro de Saúde

Para os *Staffs* utilizamos um Id de Centro de Saúde e Nome desconhecidos.

```
% Staff com id Centro de Saúde desconhecido.
staff(9, centroDesconhecido, 'Pedro Manuel Marques Mota', 'pmmm@srcr.pt').
excecao(staff(Id,Idcs,N,E)) :- staff(Id,centroDesconhecido,N,E).

% Staff com nome desconhecido.
staff(10, 2, nomeStaffDesconhecido, 'cccc@srcr.pt').
excecao(staff(Id,Idcs,N,E)) :- staff(Id,Idcs,nomeStaffDesconhecido,E).
```

Figura 11: Exemplos de Conhecimento Incerto Staff

Para as *Vacinações ao Covid* utilizamos um Id Utente e uma Vacina desconhecidos.

```
% Vacinação Covid com id Utente desconhecido.
vacinacao_Covid(2, utenteDesconhecido, 2021-05-07, 'Pfizer', 1).
execcao(vacinacao_Covid(Ids,Idu,D,V,T)) :- vacinacao_Covid(Ids,utenteDesconhecido,D,V,T).

% Vacinação Covid com vacina desconhecida.
vacinacao_Covid(3, 18, 2021-01-10, vacinaDesconhecida, 2).
execcao(vacinacao_Covid(Ids,Idu,D,V,T)) :- vacinacao_Covid(Ids,Idu,D,vacinaDesconhecida,T).
```

Figura 12: Exemplos de Conhecimento Incerto Vacinação Covid

4.6 Valor Nulo Tipo II - Conhecimento Impreciso

No caso do **Conhecimento Impreciso** ao declarar um predicado fazemos com que uma das suas variáveis tenha várias opções, de onde deduzimos que uma delas é verdadeira mas não sabemos qual, ou que se encontra dentro de uma certa gama de valores. Após o predicado, declaramos as respetivas exceções das variáveis referidas anteriormente. Alguns exemplos de Conhecimento Impreciso existentes no projeto são:

Para os *Utentes* utilizamos dois números de Segurança Social e dois números de Telemóvel.

```
% Utente com dois números de Segurança Social.
utente(20, {19283018273, 10033345901}, 'Francisco Correia Antonieto Franco', 1950-03-05, 'fcaf@srcr.pt', 930000000, 'Rua das Amoreiras, 152', 'Reformado', ['Osteoporose', 'Parkinson'], 2).
execcao(utente(20, 19283018273, 'Francisco Correia Antonieto Franco', 1950-03-05, 'fcaf@srcr.pt', 930000000, 'Rua das Amoreiras, 152', 'Reformado', ['Osteoporose', 'Parkinson'], 2)).
execcao(utente(20, 10033345901, 'Francisco Correia Antonieto Franco', 1950-03-05, 'fcaf@srcr.pt', 930000000, 'Rua das Amoreiras, 152', 'Reformado', ['Osteoporose', 'Parkinson'], 2)).

% Utente com dois números de telemóvel.
utente(21, 34554334567, 'Antonio Manuel Cascao Xisto', 1968-03-01, 'amcx@srcr.pt', {931111111, 932222222}, 'Largo dos Peões, 25', 'Pedreiro', ['Asma', 'Hipertensao'], 4).
execcao(utente(21, 34554334567, 'Antonio Manuel Cascao Xisto', 1968-03-01, 'amcx@srcr.pt', 931111111, 'Largo dos Peões, 25', 'Pedreiro', ['Asma', 'Hipertensao'], 4)).
execcao(utente(21, 34554334567, 'Antonio Manuel Cascao Xisto', 1968-03-01, 'amcx@srcr.pt', 932222222, 'Largo dos Peões, 25', 'Pedreiro', ['Asma', 'Hipertensao'], 4)).
```

Figura 13: Exemplos de Conhecimento Impreciso Utente

Para os *Centros de Saúde* utilizamos dois Emails e dois números de Telemóvel distintos.

```
% Centro de Saúde com dois emails.
centro_saude(6, 'Centro de Saude Viana do Castelo', 'Rua Senhora da Agonia, 8', 956666666, {'csvg@srcr.pt', 'cvsc@srcr.pt'}).
execcao(centro_saude(6, 'Centro de Saude Viana do Castelo', 'Rua Senhora da Agonia, 8', 956666666, 'csvg@srcr.pt')).
execcao(centro_saude(6, 'Centro de Saude Viana do Castelo', 'Rua Senhora da Agonia, 8', 956666666, 'cvsc@srcr.pt')).

% Centro de Saúde com dois números de telemóvel.
centro_saude(7, 'Centro de Saude Sao Jorge', 'Travessa da Ilha das Flores, 229', {957777777, 958888888}, 'cssj@srcr.pt').
execcao(centro_saude(7, 'Centro de Saude Sao Jorge', 'Travessa da Ilha das Flores, 229', 957777777, 'cssj@srcr.pt')).
execcao(centro_saude(7, 'Centro de Saude Sao Jorge', 'Travessa da Ilha das Flores, 229', 958888888, 'cssj@srcr.pt')).
```

Figura 14: Exemplos de Conhecimento Impreciso Centro de Saúde

Para os *Staffs* utilizamos dois Ids de Centros de Saúde e dois Emails.

```
% Staff que está associado a dois centros de saúde.
staff(7, [2,4], 'Joaquim Alberto Incrível Morais', 'jaim@srrcr.pt').
excecao(staff(7, 2, 'Joaquim Alberto Incrível Morais', 'jaim@srrcr.pt')).
excecao(staff(7, 3, 'Joaquim Alberto Incrível Morais', 'jaim@srrcr.pt')).
excecao(staff(7, 4, 'Joaquim Alberto Incrível Morais', 'jaim@srrcr.pt')).

% Staff com dois emails.
staff(8, 1, 'Tito Cerqueira Andrade Vieira', {'tcav@srrcr.pt', 'vact@srrcr.pt'}).
excecao(staff(8, 1, 'Tito Cerqueira Andrade Vieira', 'tcav@srrcr.pt')).
excecao(staff(8, 1, 'Tito Cerqueira Andrade Vieira', 'vact@srrcr.pt')).
```

Figura 15: Exemplos de Conhecimento Impreciso Staff

Para as *Vacinações ao Covid* utilizamos duas Datas e duas Vacinas diferentes.

```
% Vacinação Covid com duas datas.
vacinacao_Covid(6, 21, {2021-03-04, 2021-03-10}, 'Pfizer', 1).
excecao(vacinacao_Covid(6, 21, 2021-03-04, 'Pfizer', 1)).
excecao(vacinacao_Covid(6, 21, 2021-03-10, 'Pfizer', 1)).

% Vacinação Covid com duas vacinas.

vacinacao_Covid(3, 20, 2021-05-05, {'Pfizer', 'Astrazeneca'}, 2).
excecao(vacinacao_Covid(3, 20, 2021-05-05, 'Pfizer', 2)).
excecao(vacinacao_Covid(3, 20, 2021-05-05, 'Astrazeneca', 2)).
```

Figura 16: Exemplos de Conhecimento Impreciso Vacinação Covid

4.7 Valor Nulo Tipo III - Conhecimento Interdito

Para o **Conhecimento Interdito** declaramos o predicado com uma ou mais das suas variáveis interditas, ou seja, variáveis sobre as quais não será possível conhecer nada, e, após isso, as exceções a elas associadas. Alguns exemplos deste tipo de conhecimento são os seguintes:

Para os *Utentes* utilizamos uma Data de Nascimento e um número de Segurança Social interditos.

```
% Utente com uma data de nascimento interdita.
utente(22, 43567859643, 'Quintino Moreira Teixeira Grafo', dataInterdita, 'qmtg@srrcr.pt', 933333333, 'Rua dos Patos Cegos, 1', 'Pescador', [], 5).
excecao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)) :- utente(Id,Nss,N,dataInterdita,E,T,M,P,DC,CS).
nulo(dataInterdita).

% Utente com um número de Segurança Social interdito.
utente(23, segurancaSocialInterdita, 'William Carvalho Santos Silva', 1980-04-03, 'wcss@srrcr.pt', 934444444, 'Rua do Alecrim, 2', 'Porteiro', ['Asma'], 3).
excecao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)) :- utente(Id,segurancaSocialInterdita,N,D,E,T,M,P,DC,CS).
nulo(segurancaSocialInterdita).
```

Figura 17: Exemplos de Conhecimento Interdito Utente

Para os *Centros de Saúde* utilizamos um Nome e um número de Telemóvel interditos.

```
% Centro de Saúde com nome interdito.
centro_saude(10, nomeInterdito, 'Travessa das Picaretas, 10', 961111111, 'bbbb@srcr.pt').
execcao(centro_saude(Id,N,M,T,E)) :- centro_saude(Id,nomeInterdito,M,T,E).
nulo(nomeInterdito).

% Centro de Saúde com número de telemóvel interdito.
centro_saude(11, 'Centro de Saude Odívelas', 'Rua Sargento Coronel Frances, 4', numeroInterdito, 'csodv@srcr.pt').
execcao(centro_saude(Id,N,M,T,E)) :- centro_saude(Id,N,M,numeroInterdito,E).
nulo(numeroInterdito).
```

Figura 18: Exemplos de Conhecimento Interdito Centro

Para os *Staffs* utilizamos um Id Centro de Saúde e um Nome interditos.

```
% Staff com id Centro de Saúde interdito.
staff(11, centroInterdito, 'Pedro Pereira Pauleta Aveiro', 'pppa@srcr.pt').
execcao(staff(Id,Idcs,N,E)) :- staff(Id,centroInterdito,N,E).
nulo(centroInterdito).

% Staff com nome interdito.
staff(12, 2, nomeStaffInterdito, 'dddd@srcr.pt').
execcao(staff(Id,Idcs,N,E)) :- staff(Id,Idcs,nomeStaffInterdito,E).
nulo(nomeStaffInterdito).
```

Figura 19: Exemplos de Conhecimento Interdito Staff

Para as *Vacinações ao Covid* utilizamos um Id Utente e uma Vacina interditos.

```
% Vacinação Covid com id Utente interdito.
vacinacao_Covid(5, utenteInterdito, 2021-03-20, 'Astrazeneca', 1).
execcao(vacinacao_Covid(Ids,Idu,D,V,T)) :- vacinacao_Covid(Ids,utenteInterdito,D,V,T).
nulo(utenteInterdito).

% Vacinação Covid com nome vacina interdito.
vacinacao_Covid(3, 19, 2021-02-20, vacinaInterdita, 1).
execcao(vacinacao_Covid(Ids,Idu,D,V,T)) :- vacinacao_Covid(Ids,Idu,D,vacinaInterdita,T).
nulo(vacinaInterdita).
```

Figura 20: Exemplos de Conhecimento Interdito Vacinação Covid

5 Evolução do Conhecimento

Para uma correta inserção e remoção de conhecimento da nossa **Base de Conhecimento** foi necessário definir alguns invariantes. Os **Invariantes** têm como objetivo garantir que o conhecimento é consistente e não repetido, sendo por estas razões essenciais para este trabalho. Desta forma, implementamos dois tipos de invariantes: invariantes para a inserção e invariantes para a remoção de conhecimento.

Para definir os invariantes necessários foi necessário definir o predicado procura que executa o predicado pré definido do PROLOG *findall* e que encontra todas as soluções que satisfazem um ou mais predicados. Este predicado foi extremamente necessário para todos os predicados elaborados ao longo do projeto.

De seguida apresentamos o predicado **procura**.

```
%
% Predicado que permite a procura de Conhecimento.
% Extensão do predicado procura: Termo,Predicado,Lista -> {V,F}.
%
procura(T,P,L) :-
    findall(T,P,L).
```

Figura 21: Predicado Procura

Assim, recorrendo a este predicado foi possível criar todos os invariantes necessários quer para a inserção quer para a remoção de conhecimento.

Para o conjunto de **invariantes associados à inserção**, encontramos invariantes que impedem a repetição de conhecimento (utentes, staffs, centros de saúde ou vacinações). Deste modo, após a inserção de um determinado conhecimento, o predicado verifica se o seu ID é único, caso contrário, retira-o da base de conhecimento.

Apresentamos assim os diferentes invariantes explicados em cima.

```
%-----Invariantes para a inserção-----
%
%
+utente(Id,_,_,_,_,_,_,_,_): (procura(Id,utente(Id,_,_,_,_,_,_,_,_),L),
                                comprimento(L,R),
                                R = 1).

+centro_saude(Id,_,_,_): (procura(Id,centro_saude(Id,_,_,_,_),L),
                           comprimento(L,R),
                           R = 1).

+staff(Id,_,_,_): (procura(Id,staff(Id,_,_,_,_),L),
                   comprimento(L,R),
                   R = 1).

+vacinacao_Covid(Ids,Idu,D,_,_): (procura((Ids,Idu,D),vacinacao_Covid(Ids,Idu,D,_,_,_),L),
                                   comprimento(L,R),
                                   R = 1).
```

Figura 22: Invariantes para a inserção


```

% -----
% -----Invariantes para a remoção-----
% -----
-utente(Id,Nss,N,D,E,T,M,P,DC,CS):: (procura(Id,utente(Id,Nss,N,D,E,T,M,P,DC,CS),L),
                                     comprimento(L,R),
                                     R == 1).

-centro_saude(Id,N,M,T,E):: (procura(Id,centro_saude(Id,N,M,T,E),L),
                             comprimento(L,R),
                             R == 1).

-staff(Id,Idcs,N,E):: (procura(Id,staff(Id,Idcs,N,E),L),
                       comprimento(L,R),
                       R == 1).

-vacinacao_Covid(Ids,Idu,D,V,T):: (procura(Idu,vacinacao_Covid(Ids,Idu,D,V,T),L),
                                    comprimento(L,R),
                                    R <= 2).

```

Figura 25: Invariantes para a remoção

```

% Não permitir a remoção de utentes e de staff se existirem vacinações de covid a eles associadas.
-utente(Id,Nss,N,D,E,T,M,P,DC,CS):: (procura(Id,vacinacao_Covid(Ids,Id,D,V,T),L),
                                     comprimento(L,R),
                                     R == 0).

-staff(Id,Idcs,N,E):: (procura(Id,vacinacao_Covid(Id,Idu,D,V,T),L),
                       comprimento(L,R),
                       R == 0).

```

Figura 26: Invariantes para a remoção de um utente ou um staff

Por fim, e para um correto funcionamento do **Conhecimento Interdito** foi necessário criar invariantes que o garantissem. Deste modo, definimos os seguintes invariantes.

```
%
%-----
% Invariantes para o Conhecimento Interdito
%-----
%
+utente(Id,Nss,N,D,E,T,M,P,DC,CS) :: (procura(D,utente(22, 43567859643, 'Quintino Moreira Teixeira Grafo', D, 'qmtg@srcr.pt', 933333333, 'Rua dos Patos Cegos, 1',
'Pescador', 1), 3),
nao(nulo(D))),R),
comprimento(R,N1),
N1 = 0).

+utente(Id,Nss,N,D,E,T,M,P,DC,CS) :: (procura(Nss,utente(23, Nss, 'William Carvalho Santos Silva', 1988-04-03, 'wcss@srcr.pt', 934444444, 'Rua do Alecrim, 2', 'Porteiro',
['Asma'], 3),
nao(nulo(Nss))),R),
comprimento(R,N1),
N1 = 0).

+centro_saude(Id,N,M,T,E) :: (procura(N,centro_saude(10, N, 'Travessa das Picaretas, 10', 961111111, 'bbb@srcr.pt'),nao(nulo(N))),R),
comprimento(R,N1),
N1 = 0).

+centro_saude(Id,N,M,T,E) :: (procura(T,centro_saude(11, 'Centro de Saude Odivelas', 'Rua Sargento Coronel Frances, 4', T, 'csodv@srcr.pt'),nao(nulo(T))),R),
comprimento(R,N1),
N1 = 0).

+staff(Id,Idcs,N,E) :: (procura(Idcs,(staff(11, Idcs, 'Pedro Pereira Pauleta Aveiro', 'pppa@srcr.pt'),nao(nulo(Idcs))),R),
comprimento(R,N1),
N1 = 0).

+staff(Id,Idcs,N,E) :: (procura(N,(staff(12, 'Centro Saude Sao Pedro Do Sul', N, 'ddddd@srcr.pt'),nao(nulo(N))),R),
comprimento(R,N1),
N1 = 0).

+vacinacao_Covid(Idcs,Idu,D,V,T) :: (procura(Idu,(vacinacao_Covid(5, Idu, 2021-03-20, 'Astrazeneca', 1),nao(nulo(Idu))),R),
comprimento(R,N1),
N1 = 0).

+vacinacao_Covid(Idcs,Idu,D,V,T) :: (procura(V,(vacinacao_Covid(3, 19, 2021-02-20, V, 1),nao(nulo(V))),R),
comprimento(R,N1),
N1 = 0).
```

Figura 27: Invariantes para o Conhecimento Interdito

Uma das funcionalidades do nosso trabalho prático é conseguirmos **adicionar** e **remover** termos da nossa base de conhecimento. Para estas funcionalidades funcionarem da melhor forma, estas encontram-se intrinsecamente ligadas aos invariantes de inserção e de remoção explicados anteriormente no relatório.

Desta forma, para **adicionarmos** termos à nossa base de conhecimento utilizamos o predicado auxiliar `evolucao`. Este predicado permite adicionar um termo, respeitando todos os invariantes definidos, à base de conhecimento. Desta forma, pega numa lista onde já tem o termo verificado com os respetivos invariantes, insere-o na base de conhecimento e testa se o mesmo ficou inserido na base de conhecimento. Para isto, também são utilizados dois predicados auxiliares um que permite a inserção de um termos e outro que permite o teste de invariante de uma lista.

```
%
%-----
% Predicado que permite a evolução do Conhecimento.
% Extensão do predicado evolucao: Termo -> {V,F}.
%-----

evolucao( Termo ) :- procura(Invariante, +Termo::Invariante, Lista),
insercao( Termo ),
teste( Lista ).
```

Figura 28: Predicado evolucao


```
% -----  
% Predicado que permite o teste de invariante de uma lista.  
% Extensão do predicado teste: Lista -> {V,F}.  
% -----  
  
teste( [] ).  
teste( [R|LR] ) :- R, teste( LR ).  
  
% -----  
% Predicado que permite a inserção de um termo.  
% Extensão do predicado insercao: Termo -> {V,F}.  
% -----  
  
insercao(T) :- assert(T).  
insercao(T) :- retract(T), !, fail.
```

Figura 29: Predicados auxiliares do predicado evolucao

Para **removermos** termos da nossa base de conhecimento utilizamos a involução, num predicado ao qual chamamos de retrocesso. Este predicado permite remover um termo, respeitando todos os invariantes de remoção definidos, da nossa base de conhecimento. Deste modo, a partir de uma lista com o termos onde já foram verificados os invariantes respetivos, testa se o termo existe na base de conhecimento e, caso isto se verifique, retira-o da base de conhecimento.

```
% -----  
% Predicado que permite o retrocesso do Conhecimento.  
% Extensão do predicado retrocesso: Termo -> {V,F}.  
% -----  
  
retrocesso( Termo ) :- procura(Invariante, -Termo::Invariante, Lista),  
                        teste( Lista ),  
                        remocao( Termo ).
```

Figura 30: Predicado retrocesso

```
% -----  
% Predicado que permite a remoção de um termo.  
% Extensão do predicado remocao: Termo -> {V,F}.  
% -----  
  
remocao(T) :- retract(T).  
remocao(T) :- assert(T),!,fail.
```

Figura 31: Predicado auxiliar do predicado retrocesso

Após estes predicados, temos todas as condições reunidas para conseguir registrar e remover conhecimento da nossa base de conhecimento. Passando os parâmetros necessários a cada um dos respetivos conhecimentos, utentes, staffs, centros de saúde e vacinações, utilizamos os predicados evolução e retrocesso para o conseguir.

```
%
% Predicado que regista Utes.
% Extensão do predicado registrarUte: Id_Ute, Número de Segurança Social, Nome, Data Nascimento, Email, Telefone, Morada, Profissão, [Doenças_Crónicas], CentroSaúde {V,F}.
%
%
% registrarUte(Id,Nss,N,D,E,T,M,P,DC,CS) :-
%   evolucao(ute(Id,Nss,N,D,E,T,M,P,DC,CS)).
%
% Predicado que regista Centros de Saúde.
% Extensão do predicado registrarCentro: Id_Centro, Nome, Morada, Telefone, Email -> {V,F}.
%
% registrarCentro(Id,N,M,T,E) :-
%   evolucao(centro_saude(Id,N,M,T,E)).
%
% Predicado que regista Staffs.
% Extensão do predicado registrarStaff: Id_Staff, Id_Centro, Nome, Email -> {V,F}.
%
% registrarStaff(Id,Idcs,N,E) :-
%   evolucao(staff(Id,Idcs,N,E)).
%
% Predicado que regista Vacinações de Covid.
% Extensão do predicado registrarVacinao: Id_Staff, Id_Ute, Data, Vacina, Toma -> {V,F}.
%
% registrarVacinao(Ids,Idu,D,V,T) :-
%   evolucao(vacinao_covid(Ids,Idu,D,V,T)).
```

Figura 32: Predicados de registo de conhecimento

```
%
% Predicado que remove Utes.
% Extensão do predicado removerUte: Id_Ute, Número de Segurança Social, Nome, Data Nascimento, Email, Telefone, Morada, Profissão, [Doenças_Crónicas], CentroSaúde {V,F}.
%
%
% removerUtes(Id,Nss,N,D,E,T,M,P,DC,CS) :-
%   retrocesso(ute(Id,Nss,N,D,E,T,M,P,DC,CS)).
%
% Predicado que remove Centros de Saúde.
% Extensão do predicado removerCentros: Id_Centro, Nome, Morada, Telefone, Email -> {V,F}.
%
% removerCentros(Id) :-
%   retrocesso(centro_saude(Id,N,M,T,E)).
%
% Predicado que remove Staff.
% Extensão do predicado removerStaff: Id_Staff, Id_Centro, Nome, Email -> {V,F}.
%
% removerStaff(Id) :-
%   retrocesso(staff(Id,Idcs,N,E)).
%
% Predicado que remove Vacinações de Covid.
% Extensão do predicado removerVacinao: Id_Staff, Id_Ute, Data, Vacina, Toma -> {V,F}.
%
% removerVacinao(Ids,Idu,D) :-
%   retrocesso(vacinao_covid(Ids,Idu,D,V,T)).
```

Figura 33: Predicado de remoção de conhecimento

6 Funcionalidades Adicionais

6.1 Atualizar Conhecimento Imperfeito

Uma das funcionalidades adicionais do nosso Trabalho Prático consiste na capacidade de ser possível atualizar **Conhecimento Imperfeito**, mais especificamente **Conhecimento Incerto** e **Conhecimento Impreciso**. Tudo isto é apenas possível devido aos invariantes acima descritos e é feito através da substituição de valores desconhecidos por valores que queremos que sejam conhecidos. Os seguintes predicados exemplificam estas funcionalidades adicionais.

6.1.1 Atualizar Conhecimento Incerto

- Utentes

```
% ----- Utente -----
% -----
% -----

% Atualizar um Utente com Número de Segurança Social Desconhecido.
atualizarUtenteImp(Id,Nss,N,D,E,T,M,P,DC,CS) :- excecacao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)),
retrocesso(utente(Id,nissDesconhecido,N,D,E,T,M,P,DC,CS)),
evolucao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)).

% Atualizar um Utente com Nome Desconhecido.
atualizarUtenteImp(Id,Nss,N,D,E,T,M,P,DC,CS) :- excecacao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)),
retrocesso(utente(Id,Nss,nomeUtenteDesconhecido,D,E,T,M,P,DC,CS)),
evolucao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)).

% Atualizar um Utente com Data de Nascimento Desconhecido.
atualizarUtenteImp(Id,Nss,N,D,E,T,M,P,DC,CS) :- excecacao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)),
retrocesso(utente(Id,Nss,N,dataUtenteDesconhecido,E,T,M,P,DC,CS)),
evolucao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)).

% Atualizar um Utente com Email Desconhecido.
atualizarUtenteImp(Id,Nss,N,D,E,T,M,P,DC,CS) :- excecacao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)),
retrocesso(utente(Id,Nss,N,D,emailUtenteDesconhecido,T,M,P,DC,CS)),
evolucao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)).

% Atualizar um Utente com Número de Telemóvel Desconhecido.
atualizarUtenteImp(Id,Nss,N,D,E,T,M,P,DC,CS) :- excecacao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)),
retrocesso(utente(Id,Nss,N,D,E,telemovelDesconhecido,M,P,DC,CS)),
evolucao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)).

% Atualizar um Utente com Morada Desconhecido.
atualizarUtenteImp(Id,Nss,N,D,E,T,M,P,DC,CS) :- excecacao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)),
retrocesso(utente(Id,Nss,N,D,E,T,moradaUtenteDesconhecido,P,DC,CS)),
evolucao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)).

% Atualizar um Utente com Profissão Desconhecido.
atualizarUtenteImp(Id,Nss,N,D,E,T,M,P,DC,CS) :- excecacao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)),
retrocesso(utente(Id,Nss,N,D,E,T,M,profissaoUtenteDesconhecido,DC,CS)),
evolucao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)).

% Atualizar um Utente com Doenças Desconhecido.
atualizarUtenteImp(Id,Nss,N,D,E,T,M,P,DC,CS) :- excecacao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)),
retrocesso(utente(Id,Nss,N,D,E,T,M,P,doencasUtenteDesconhecido,CS)),
evolucao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)).

% Atualizar um Utente com Centro de Saúde Desconhecido.
atualizarUtenteImp(Id,Nss,N,D,E,T,M,P,DC,CS) :- excecacao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)),
retrocesso(utente(Id,Nss,N,D,E,T,M,P,DC,centroSaudeUtenteDesconhecido)),
evolucao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)).
```

Figura 34: Predicado de atualização de Conhecimento Incerto Utentes

- Centros de Saúde

```
% ----- Centro de Saúde -----
%
%
% Atualizar um Centro de Saúde com Nome Desconhecido.
atualizarCentroSaudeImp(Id,N,M,T,E) :- excecacao(centro_saude(Id,N,M,T,E)),
                                         retrocesso(centro_saude(Id,nomeDesconhecido,M,T,E)),
                                         evolucao(centro_saude(Id,N,M,T,E)).

% Atualizar um Centro de Saúde com Morada Desconhecido.
atualizarCentroSaudeImp(Id,N,M,T,E) :- excecacao(centro_saude(Id,N,M,T,E)),
                                         retrocesso(centro_saude(Id,N,moradaCentroSaudeDesconhecido,T,E)),
                                         evolucao(centro_saude(Id,N,M,T,E)).

% Atualizar um Centro de Saúde com Número de Telemóvel Desconhecido.
atualizarCentroSaudeImp(Id,N,M,T,E) :- excecacao(centro_saude(Id,N,M,T,E)),
                                         retrocesso(centro_saude(Id,N,M,telemovelCentroSaudeDesconhecido,E)),
                                         evolucao(centro_saude(Id,N,M,T,E)).

% Atualizar um Centro de Saúde com Email Desconhecido.
atualizarCentroSaudeImp(Id,N,M,T,E) :- excecacao(centro_saude(Id,N,M,T,E)),
                                         retrocesso(centro_saude(Id,N,M,T,emailDesconhecido)),
                                         evolucao(centro_saude(Id,N,M,T,E)).
```

Figura 35: Predicado de atualização de Conhecimento Incerto Centro de Saúde

- Utentes

```
% ----- Staff -----
%
%
% Atualizar um Staff com Id Centro de Saúde Desconhecido.
atualizarStaffImp(Id,Idcs,N,E) :- excecacao(staff(Id,Idcs,N,E)),
                                   retrocesso(staff(Id,centroDesconhecido,N,E)),
                                   evolucao(staff(Id,Idcs,N,E)).

% Atualizar um Staff com Nome Desconhecido.
atualizarStaffImp(Id,Idcs,N,E) :- excecacao(staff(Id,Idcs,N,E)),
                                   retrocesso(staff(Id,Idcs,nomeStaffDesconhecido,E)),
                                   evolucao(staff(Id,Idcs,N,E)).

% Atualizar um Staff com Email Desconhecido.
atualizarStaffImp(Id,Idcs,N,E) :- excecacao(staff(Id,Idcs,N,E)),
                                   retrocesso(staff(Id,Idcs,N,emailStaffDesconhecido)),
                                   evolucao(staff(Id,Idcs,N,E)).
```

Figura 36: Predicado de atualização de Conhecimento Incerto Staffs

6.1.2 Atualizar Conhecimento Impreciso

- Utentes

```
% ----- Utente -----  
%  
%  
% Atualizar o Conhecimento Impreciso de um Utente Imperfeito.  
atualizarUtenteImp(Id,Nss,N,D,E,T,M,P,DC,CS) :- si(utente(Id,Nss,N,D,E,T,M,P,DC,CS),desconhecido),  
retrocesso(utente(Id,_,_,_,_,_,_,_,_)),  
procura(excecao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)),excecao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)),R),  
comprimento(R,N1),  
N1 > 0,  
removerExcecoes(R),  
evolucao(utente(Id,Nss,N,D,E,T,M,P,DC,CS)).
```

Figura 37: Predicado de atualização de Conhecimento Impreciso Utentes

- Centros de Saúde

```
% ----- Centro de Saúde -----  
%  
%  
% Atualizar o Conhecimento Impreciso de um Centro de Saúde Imperfeito.  
atualizarCentroSaudeImp(Id,N,M,T,E) :- si(centro_saude(Id,N,M,T,E),desconhecido),  
retrocesso(centro_saude(Id,_,_,_,_)),  
procura(excecao(centro_saude(Id,N,M,T,E)),(excecao(centro_saude(Id,N,M,T,E))),R),  
comprimento(R,N1),  
N1 > 0,  
removerExcecoes(R),  
evolucao(centro_saude(Id,N,M,T,E)).
```

Figura 38: Predicado de atualização de Conhecimento Impreciso Centros de Saúde

- Utentes

```
% ----- Staff -----  
%  
%  
% Atualizar o Conhecimento Impreciso de um Staff Imperfeito.  
atualizarStaffImp(Id,Idcs,N,E) :- si(staff(Id,Idcs,N,E),desconhecido),  
retrocesso(staff(Id,_,_,_)),  
procura(excecao(staff(Id,Idcs,N,E)),(excecao(staff(Id,Idcs,N,E))),R),  
comprimento(R,N1),  
N1 > 0,  
removerExcecoes(R),  
evolucao(staff(Id,Idcs,N,E)).
```

Figura 39: Predicado de atualização de Conhecimento Impreciso Staffs

7 Conclusão

Em suma, a realização deste trabalho prático permitiu a consolidação de diversos conhecimentos obtidos nas aulas práticas e teóricas da unidade curricular de Sistemas de Representação de Conhecimento e Raciocínio e também relativamente à linguagem de programação PROLOG.

O nosso projeto permite representar todo o conhecimento pedido no enunciado e todas as funcionalidades pedidas estão funcionais e eficientes. Acrescentamos algumas funcionalidades extra que achamos convenientes dado o propósito do projeto e a ideia base em que foi feito, ou seja, a vacinação da população portuguesa ao COVID-19 e, nesta fase, a adição de Conhecimento Imperfeito que aproxima todo o trabalho daquilo que é a realidade.

Concluindo, todos os objetivos propostos pelos docentes da unidade curricular foram conseguidos e pensamos que, no geral, todo o trabalho prático foi bem conseguido pelo grupo.