



Departamento de Informática

UNIVERSIDADE DO MINHO

Programação em Lógica

Mestrado Integrado em Engenharia Informática

Sistemas de Representação de Conhecimento e Raciocínio
(2^o semestre 2020/2021)

81105	Tiago Manuel Queirós Barata
85517	Duarte Manuel Vilar de Oliveira
82098	Melânia Rafaela Sousa Pereira
67645	António Sérgio da Costa Gomes
83693	Rui Filipe Chaves

Braga,
abril 2021

Resumo

O presente documento apresenta um relatório detalhado do desenvolvimento de um sistema de representação de conhecimento e raciocínio com capacidade para caracterizar um universo de discurso na área da vacinação global da população portuguesa no contexto pandémico em que o país se encontra.

Este sistema foi desenvolvido com recurso à linguagem de programação Prolog e com um foco mais incidente na programação em lógica, englobando também sistemas de inferência e mecanismos de controlo de consistência e evolução de conhecimento.

O principal objetivo deste projeto foi a consolidação de conhecimentos na área da programação em lógica, objetivo esse que o grupo que desenvolveu o projeto viu cumprido. Para além deste, há também os objetivos que se pretendia para o sistema, sendo alguns deles a identificação de pessoas vacinadas, candidatas a vacinação, definição de critérios de inclusão a fases de vacinação, entre outros.

Ao longo deste relatório serão apresentados com pormenor todos os raciocínios e técnicas que levaram o grupo a desenvolver este sistema da forma que o fez, juntando, a cada explicação, um anexo em que é possível consultar o código desenvolvido.

Índice

1	Introdução	1
2	Preliminares	2
3	Descrição do Trabalho e Análise de Resultados	4
3.1	Apresentação da Solução	4
3.1.1	Base de Conhecimento [1]	4
3.1.2	Funcionalidades Básicas [2]	4
3.1.3	Funcionalidades Extra [3]	7
3.1.4	Predicados auxiliares [6]	9
3.2	Análise dos Resultados Obtidos	11
3.2.1	Registo de dados	11
3.2.2	Remoção de dados	12
3.2.3	Consultas	13
3.2.4	Listagens	13
4	Conclusões e Sugestões	15
	Bibliografia	16
	Lista de Siglas e Acrónimos	17
	Anexos	18
Anexo 1	18
Anexo 2	19
Anexo 3	20
Anexo 4	21
Anexo 5	24
Anexo 6	28

1 Introdução

O presente relatório descreve o desenvolvimento de uma solução para o problema apresentado pelos docentes da unidade curricular Sistemas de Representação de Conhecimento e Raciocínio do terceiro ano do Mestrado Integrado em Engenharia Informática.

O trabalho proposto tem como principal objetivo a aplicação dos conhecimentos adquiridos nesta UC, nomeadamente sobre programação lógica, mas não excluindo outros, para o desenvolvimento de um sistema de representação de conhecimento e raciocínio com capacidade para caracterizar um universo de discurso na área da vacinação global da população portuguesa no contexto pandémico em que o país se encontra.

Com este relatório pretende-se explicar e dar a conhecer todo o raciocínio e técnicas usadas pelo grupo para a realização deste projeto.

Toda a estrutura deste relatório foi baseada nas sugestões presentes no relatório técnico fornecido pelos docentes [ANALIDE et al., 2001].

2 Preliminares

Para a realização deste trabalho foi necessário ter em mente todos os conceitos teóricos aprendidos e explorados nas aulas desta UC e também conceitos práticos sobre a linguagem Prolog, conceitos estes que vão desde a criação de termos que permitem adicionar dados à base de conhecimento até ao desenvolvimento de predicados mais complexos que permitam fornecer respostas em relação ao conhecimento já adquirido, e ainda, predicados que permitem a evolução desse conhecimento de forma consistente.

Além disto, foi ainda necessário analisar o enunciado do problema de forma a enquadrar o tema e perceber quais as fontes de conhecimento que iriam servir de base para o sistema a desenvolver:

- **utente** - caracteriza os utentes do sistema de saúde, que têm associados a si um identificador de utente, o seu NSS, nome, data de nascimento, email, número de telemóvel, morada de residência, profissão, uma lista de doenças crónicas e o identificador do seu centro de saúde.

utente: $\#Utente, N^o \text{Seguran\c{c}a_Social, Nome, Data_Nasc, Email, Telefone, Morada, Profiss\c{a}o, [Doen\c{c}as_Cr\c{o}nicas], \#CentroSa\c{u}de, \#M\c{e}dicoDeFamilia \rightarrow \{V,F\}$

- **centro de saúde** - caracteriza os centros de saúde existentes no país, e tem associado a si um identificador único, o nome, a morada onde está localizado, o número de telefone e email de contacto.

centro_saude: $\#Centro, Nome, Morada, Telefone, Email \rightarrow \{V,F\}$

- **staff** - caracteriza o *staff* de toda a vacinação e cada pessoa tem associada a si um identificador único, o identificador do centro de saúde onde está a trabalhar e o seu nome e email.

staff: $\#Staff, \#Centro, Nome, email \rightarrow \{V,F\}$

- **vacinação covid** - caracteriza as vacinações efetuadas e tem associado o identificador da pessoa do *staff* que realizou a vacinação, o identificador do utente que tomou a vacina, a data em que a vacina foi administrada, o nome da vacina e o número da toma.

vacinacao_covid: $\#Staff, \#Utente, Data, Vacina, Toma \rightarrow \{V,F\}$

Assim, foram estas as bases necessárias ao grupo para iniciar o desenvolvimento deste sistema.

No entanto, mais tarde o grupo percebeu que iria precisar de mais fontes de conhecimento para desenvolver funcionalidades extra àquelas que foram sugeridas, e decidiu adicionar as seguintes:

- **médico de família** - caracteriza um médico de família pelo seu identificador numeral, o nome, o *e-mail* e o valor indicativo do centro de saúde a que este médico de família pertence.

`medico_familia: #Medico, Nome, Email, #Centro → {V,F}`

- **consulta** - caracteriza uma consulta médica entre um médico de família e um utente. Tem associado um identificador da consulta, do médico e do utente, a descrição da consulta, o seu custo e a data da realização da mesma.

`consulta: #Consulta, #Utente, #Medico, Descricao, Custo, Data → {V,F}`

Para implementação de todas as funcionalidades pretendidas, o grupo viu o seu trabalho bastante simplificado com o uso de um *package* do *prolog* com o nome de `date_time` [NOGATZ, 2020]. Este *package* é responsável por todas as operações com datas, desde cálculos de tempo, saber a data atual, entre outras funções. Deste modo, para o bom funcionamento da solução apresentada, será necessária a instalação do *package* para executar estas operações:

```
?- pack_install(date_time).
```

Listing 2.1: Instalação do *package date_time*

Além disto, o *package* em questão tem uma dependência de um outro, que é necessário instalar também:

```
?- pack_install(tap).
```

Listing 2.2: Instalação da dependência

Após estas instalações, o programa desenvolvido tem todas as funcionalidades disponíveis.

3 Descrição do Trabalho e Análise de Resultados

3.1 Apresentação da Solução

Para o desenvolvimento do sistema o grupo começou por criar uma base de conhecimento aleatória com os dados necessários ao sistema para poder testar todas as suas funcionalidades.

De seguida passou-se a desenvolver predicados que permitam ao sistema ter as funcionalidades básicas pedidas no enunciado do problema e, depois, de todas essas estarem funcionais e a dar o resultado esperado, procedeu-se à implementação de algumas funcionalidades extra que o grupo achou que fariam sentido num sistema baseado no tema em causa.

Apresenta-se de seguida uma breve explicação do raciocínio e técnicas usados para desenvolvimento de cada um dos passos descritos acima.

3.1.1 Base de Conhecimento [1]

Para criar uma base de conhecimento razoável para testar o sistema foi necessário adicionar um conjunto de dados das fontes de conhecimento, dados esses que foram gerados arbitrariamente pelo grupo, sendo por isso aleatórios e não baseados na realidade.

3.1.2 Funcionalidades Básicas [2]

Permitir a definição de fases de vacinação

Para esta funcionalidade começou por se decidir quais os critérios de inclusão de utentes em determinada fase, decidiu-se que seriam os seguintes:

- Fase 1 - Pessoas com idade superior ou igual a 80 anos;
- Fase 2 - Pessoas com doenças crónicas;
- Fase 3 - Pessoas com profissão de risco;
- Fase 4 - A restante população.

Para definir o predicado que permita determinar que utentes estão incluídos na fase 1, foi necessário desenvolver um predicado auxiliar que determina a idade [3.1.4] de um certo utente. Tendo esse predicado, foi fácil perceber que um utente está incluído na fase 1 se a sua idade for maior ou igual a 80 anos..

Para a fase 2, foi também necessário desenvolver um predicado auxiliar que determina se uma lista é vazia, de maneira a poder saber se a lista de doenças crónicas de um utente está, ou não, vazia[3.1.4], para que, no caso de não estar, saber que essa pessoa está incluída na fase 2..

Para a fase 3, mais uma vez, foi necessário desenvolver um predicado auxiliar que seja verdadeiro no caso de um elemento pertencer a uma lista. Neste caso, o objetivo é perceber se a profissão de um determinado utente está na lista[3.1.4] de profissões de risco e, se tal for verdadeiro, então esse utente está incluído nesta fase.

Finalmente, para definir a fase 4, apenas se usou um predicado auxiliar que é verdadeiro se uma determinada questão em análise for falsa, e vice versa[3.1.4]. Com este predicado auxiliar é, então, possível determinar se um utente não está em nenhuma das fases anteriores, o que, no caso de isso ser verdade, significa que ele está incluído nesta fase.

Para completar, e porque as pessoas apenas pertencem a uma única fase de vacinação (isto é, a primeira vez que serão vacinadas), o grupo implementou também um predicado que identifica a fase a que uma pessoa pertence.

Para além disto, desenvolveu-se também, para cada fase, um predicado que encontra todos os utentes incluídos na fase em questão e os dá na forma de uma lista.

Identificar pessoas não vacinadas

Uma pessoa não vacinada é uma pessoa que está registada como utente na base de conhecimento mas que não tem nenhuma vacinação registada associada ao seu *id* de utente.

Para passar este raciocínio para a linguagem Prolog, usou-se, mais uma vez, o predicado auxiliar `nao/1`, para verificar se não existe nenhuma vacinação associada ao utente, pois, caso não exista, ou seja, o predicado `nao` ser verdadeiro, significa que o utente não está vacinado.

O grupo decidiu ainda que seria bom ter um predicado que informasse qual ou quais tomas em que uma pessoa não foi vacinada, este predicado apenas acede aos registos de vacinação e confirma a existência da toma 1 e toma 2 para o utente em causa.

Identificar pessoas vacinadas

Em oposição à anterior, uma pessoa vacinada é uma pessoa que está registada como utente e que tem uma vacinação associada ao seu *id* de utente.

Este raciocínio é mais fácil de implementar porque apenas é necessário verificar o registo do utente e o registo de uma vacinação que tenha o seu *id* de utente associado.

Tal como no anterior, decidiu-se implementar um predicado que informa qual ou quais as tomas da vacina em que uma pessoa já foi vacinada.

Identificar pessoas indevidamente vacinadas

Uma pessoa vacinada indevidamente, na interpretação do grupo, é alguém que devia ser vacinada numa fase posterior à qual foi vacinada, ou seja, “furou” a lista de espera de alguma fase.

Para desenvolver este predicado, optou-se por verificar se a pessoa foi vacinada com a 1^a toma e em que data. Depois avalia-se a fase em que a pessoa devia ser vacinada e a data de início dessa fase. Por fim, comparam-se as datas, verificando se a data da fase é posterior à data na qual a pessoa foi vacinada. Se se verificar tudo isto, então conclui-se que a pessoa foi vacinada indevidamente.

Identificar pessoas candidatas a vacinação

Um candidato a vacinação, é alguém que será vacinado na fase que está a decorrer.

Para implementação deste predicado, verifica-se em primeira instância se a pessoa está registada na base de conhecimento, através do predicado *utente/1* e se a pessoa não foi já vacinada. De seguida, verificamos que fase está a decorrer, através da data atual e da data de início da fase, sendo que a data atual será posterior à data de início da fase.

Estão assim reunidas todas as condições para verificar que uma pessoa é candidata a vacinação.

Identificar pessoas a quem falta a segunda toma da vacina

Na interpretação e implementação do grupo, alguém a quem falta a segunda toda da vacina, é alguém que já tomou a primeira, mas não a segunda.

Para desenvolvimento deste predicado, faz-se extamente essas verificações, obtendo assim as pessoas vacinadas apenas com 1 dose.

Desenvolver um sistema de inferência

O sistema de inferência permite que dado um predicado, se obtenha o valor de ‘verdadeiro’, ‘falso’ ou ‘desconhecido’.

Abre-se assim a possibilidade de representação do conhecimento imperfeito, que apesar de não haver nada implementado nesta fase, ficam os alicerces para um posterior desenvolvimento mais complexo do trabalho.

3.1.3 Funcionalidades Extra [3]

Consultar o custo total de um utente em consultas médicas

De modo a explorar uma das fontes de conhecimento que foi desenvolvida - as consultas médicas - decidiu-se criar um predicado que verificasse os custos associados a consultas por parte de um utente.

Para formular este predicado começou por se verificar a existência do utente pelo seu identificador. Após isso, tomando como base o seu *id* e utilizando o predicado *solucoes/3*, reuniu-se todos os custos de consultas do utente em questão. Finalmente, a partir do predicado *sumlist/2* calcula-se o somatório de valores constituintes dessa lista, que dará o valor total gasto em consultas pelo utente.

Listagens

Uma funcionalidade que se implementou foi criar listas sobre algo, o que facilitava uma consulta, de um modo mais genérico. Por exemplo, em vez de verificar se uma pessoa já foi vacinada, é possível listar todas as pessoas que já foram vacinadas.

Listagens que o grupo implementou:

- Lista de médicos por centro de saúde
- Lista dos utentes por centro de saúde
- Lista do staff por centro de saúde
- Lista das pessoas não vacinadas
- Lista das pessoas vacinadas
- Lista das pessoas vacinadas indevidamente
- Lista das pessoas candidatas a vacinação
- Lista das pessoas a quem falta a segunda toma

Mecanismos de controlo de consistência e evolução [4]

Para manter a consistência da base de conhecimento do sistema foi necessário implementar mecanismos de controlo da mesma, sendo estes constituídos por invariantes que definem regras às quais o sistema deve obedecer aquando de uma evolução ou involução de conhecimento.

Os invariantes desenvolvidos pelo grupo garantem que não há inserção de conhecimento repetido de nenhuma fonte de conhecimento e que todos os identificadores - de utente, de staff e de centro de saúde - são únicos, para além disso, garantem ainda que:

- não existem utentes nem *staff* registados em centros de saúde que não sejam

conhecidos pelo sistema;

- não há vacinações feitas nem por uma pessoa do staff que não esteja registada no sistema nem a um utente que não esteja registado no sistema;
- não há vacinações em que a toma seja a segunda sem que exista já a primeira toma registada;
- não pode ser retirado da base de conhecimento um utente que ainda não tenha a vacinação completa (duas doses/tomas);
- não pode ser retirado da base de conhecimento um centro de saúde que tenha utentes registados.

Para poder existir esta evolução (e também involução) de conhecimento controlada pelos invariantes, foi necessário desenvolver predicados (*evolucao/3* e *involucao/3*) que verificam que, depois da inserção (ou remoção) de novo conhecimento, o invariante continua a verificar-se.

Para esta verificação são necessários predicados auxiliares que são *built-in*, como o *assert/1* e o *retract/1*, que adicionam e retiram, respetivamente, conhecimento ao sistema, estes predicados são utilizados neste sistema para fazer exatamente isso, mas seguidos dos devidos testes de verificação da consistência com a ajuda dos invariantes e, no caso dessa consistência não se verificar, o processo de inserção ou remoção que foi feito, é desfeito.

Registo e remoção de dados

Para facilitar a adição e remoção de novo conhecimento ao sistema pelo utilizador, o grupo desenvolveu um conjunto de predicados que recorrem aos predicados de evolução e involução já mencionados acima, acabando por "esconder" do utilizador os mecanismos usados para tal. Assim, o utilizador pode:

- Registar/remover um utente
- Registar/remover um centro de saúde
- Registar/remover um elemento do staff
- Registar/remover um registo de vacinação
- Registar/remover um médico de família
- Registar/remover uma consulta médica

Menu [5]

Com o objetivo de criar uma interface intuitiva para o utilizador, o grupo implementou um menu, com diversas opções. Este menu foi criado em texto, com opções numéricas, e após a apresentação de resultados, aparece novamente o menu.

Estas opções dividem-se em 4 grandes grupos:

- Inserir [3.1.3]
- Remover [3.1.3]
- Consultar [3.1.2]
- Listar [3.1.3]

Dentro de cada grupo, o utilizador pode consultar a base de conhecimento, e todas as suas funcionalidades. De modo a tornar esta interface um pouco mais apelativa, adotou-se um código de cores, que permite o utilizador identificar os resultados, no meio de tanto texto.

3.1.4 Predicados auxiliares [6]

idade

Para definir este predicado (*idade/2*), foram usados dois predicados *built-in* no *SWI-Prolog*, são eles o *date/1*, que dá a data atual na forma de *date(Ano,Mes,Dia)*, e o *date_interval/3* que dá o intervalo entre duas datas na forma que for especificada - *years, months, days*, entre outros.

Assim, para saber a idade de um utente, acede-se aos seus dados para saber a sua data de nascimento, através do predicado *date* sabe-se a data do momento e, depois, calcula-se o intervalo entre as duas em anos, sabendo, assim, a idade do utente.

vazia

Este predicado (*vazia/1*) é muito simples e resume-se a um termo. Ou seja, é um predicado que é verdadeiro porque se encontra na "base de conhecimento" do sistema, o sistema sabe que vazia de uma lista vazia será verdadeiro, e, portanto, se a esse predicado for dada uma lista que não seja vazia, como o sistema não tem conhecimento de nenhum termo com essa forma, será falso.

pertence

Este predicado (**pertence**/2) já foi desenvolvido em aulas desta UC e é um predicado que usa recursividade. Começa por avaliar se o elemento à cabeça da lista é igual ao elemento que se pretende analisar e, se for igual, então o predicado é verdadeiro porque se sabe que o elemento está na lista; caso contrário, faz-se uma chamada recursiva do predicado mas, desta vez, com apenas a cauda da lista.

nao

Este predicado (**nao**/1) foi também desenvolvido nas aulas da UC e fornecido pelos docentes. Trata-se de um predicado que nega uma questão, para isso, recorre ao predicado **cut**/0, que impede o retrocesso, seguido do predicado **fail**/0 que é sempre falso, para que, se uma questão for verdadeira, chegar ao predicado **fail** e, como dá falso, retroceder para tentar novamente mas, como encontra o **cut**, isso não acontece e, assim, o resultado para uma questão que seja verdadeira, dará falso.

print

Este predicado **print**/1 foi desenvolvido no contexto do menu, de modo a apresentar uma lista de modo mais claro para o utilizador. Para o seu funcionamento recorre ao predicado **write**/1 de maneira a formatar o output. Faz isto para cada elemento da lista recebida como argumento, fazendo uma chamada recursiva de si próprio.

getNames

Este predicado **getNames**/2 foi desenvolvido no contexto do menu, de modo a que de uma lista de Ids de utentes, fosse obtida uma lista com os nomes desses mesmos utentes. Para isso, foi feita uma chamada recursiva desse predicado, criando a lista pretendida.

3.2 Análise dos Resultados Obtidos

Dada a base de conhecimento acima descrita é possível saber *a priori* quais os resultados que seriam esperados aquando da execução de cada uma das funcionalidades pretendidas do sistema e desta forma é possível corroborar os resultados obtidos pela solução desenvolvida permitindo a validação dos mesmos.

De entre as funcionalidades disponíveis ao utilizador através do menu podemos então comparar o resultado obtido com o resultado que seria esperado dada a base de conhecimento, de forma a corroborar os resultados.

```
-----MENU-----
-----Inserir-----
1.Registar Utente
2.Registar Centro de Saúde
3.Registar Elemento do Staff
4.Registar Vacinação
5.Registar Médico de Família
6.Registar Consulta
-----Remover-----
7.Remover Utente
8.Remover Centro de Saúde
9.Remover Elemento do Staff
10.Remover Vacinação
11.Remover Médico de Família
12.Remover Consulta
-----Consultar-----
13.Custo total de consultas por utente
14.Consultar se um utente já foi vacinado
15.Consultar se um utente ainda não foi vacinado
16.Consultar em que fase vai ser vacinado
17.Consultar candidata em vacinação na fase a decorrer
18.Consultar se a pessoa ainda não tem a 2ª dose
19.Consultar se a pessoa foi vacinada indevidamente
-----Listar-----
20.Listar os médicos por centro de saúde
21.Listar os utentes por centro de saúde
22.Listar o staff por centro de saúde
23.Listar as pessoas não vacinadas
24.Listar as pessoas vacinadas
25.Listar as pessoas vacinadas indevidamente
26.Listar as pessoas candidatas a vacinação
27.Listar as pessoas a quem falta a segunda toma
0.Sair
>>
```

3.2.1 Registo de dados

Registo de utentes

Registo de um novo utente realizado com sucesso através do menu disponível ao utilizador. A inserção de um utente está também sujeita a um conjunto de invariantes de evolução que regem a inserção de novos utentes de modo a manter a consistência da base de conhecimento.

```
>> 1.
Id Utente: |: 10.
NSS: |: 12124545787.
Nome (entre plicas): |: 'Manel Santos'.
Data de Nascimento: (no formato date(Ano,Mes,Dia))|: (1999,10,16).
Email (entre plicas): |: 'manelsantos@gmail.com'.
Telefone: |: 919598946.
Morada (entre plicas): |: 'Rua José Antunes Manel 117, Gualtar Braga'.
Profissao (entre plicas): |: 'Eletricista'.
Lista de doenÇas crÃ³nicas (no formato [doenÃ§a1,doenÃ§a2,...] ou [] se nenhum[]).
Id Centro de SaÃ¼de: |: 9.
Id MÃ©dico de FamÃ-lia: |: 6.
Sucesso!
```

Registo de centros de saúde

Registo de um novo centro de saúde com sucesso através do menu disponível ao utilizador. A inserção de um centro de saúde está também sujeita a um conjunto de invariantes de evolução que regem a inserção de novos centros de saúde de modo a manter a consistência da base de conhecimento.

```
-----
>> |: 2.
Id Centro de Saúde: |: 11.
Nome (entre plicas): |: 'Centro de Saúde de Santa Maria da Feira'.
Morada (entre plicas): |: 'Rua do Vergario nº77 SMF'.
Telefone: |: 256879880.
E-mail (entre plicas): |: 'smfcentrosaude@gomail.pt'.
Sucesso!
```

3.2.2 Remoção de dados

Remoção de utentes

Remoção de um utente com sucesso dado o ID do utente a remover através do menu disponível ao utilizador. A remoção de um utente está também sujeita a um conjunto de invariantes de involução que regem a remoção de utentes de modo a manter a consistência da base de conhecimento.

```
>> 7.
Id Utente: |: 3.
Sucesso!
```

Remoção de consultas médicas

Remoção de consultas médicas com sucesso através do menu disponível ao utilizador. A remoção de uma consulta está também sujeita a um conjunto de invariantes de involução que regem a remoção de consultas de modo a manter a consistência da base de conhecimento

```
>> 12.
Id Consulta: |: 10.
Sucesso!
```

3.2.3 Consultas

Consulta dos custos em consultas médicas de um utente

Para o utente dado como exemplo é sabido *a priori* que os custos somados de consultas que o mesmo efectuou é de 90 U.M. O que permite validar os dados apresentados na figura abaixo.

```
>> |: 13.  
Id Utente: |: 4.  
Total: 90.78
```

Interrogar se um utente é candidato a vacinação na corrente fase de vacinação

Tomando o utente 4 como exemplo, podemos ver que este não pertence ao grupo de indivíduos que são elegíveis para a presente fase de vacinação. Tal informação valida os dados obtidos na respectiva interrogação à base de conhecimento.

```
-----  
>> |: 17.  
Id Utente: |: 4.  
Falso
```

3.2.4 Listagens

Listagem de utentes por centro de saúde

Para o centro de saúde *USFVida+* correspondente à opção 1 do menu de centros de saúde podemos verificar que o mesmo possui duas utentes *Ana Murciellago* e *Maria do Olival* tal como pode ser verificado.

```
-----  
>> |: 21.  
1-USFVida+  
2-UCSP Quarteira  
3-USF infesta  
4-USF MOLICEIRO  
5-CS Penha de França  
6-Centro de Saude Caldas da Rainha  
7-Centro Saúde Lousã  
8-Centro de Saúde S.Pedro do Sul  
9-Centro Saúde Ovar  
10-Centro de Saude Almodovar  
Id Centro: |: 1.  
[Ana Murciellago,Maria do Olival]
```


Listagem de pessoas vacinadas indevidamente

Os indivíduos *Maria do Olival* e *João Aristóteles* são os únicos indivíduos presentes na base de conhecimento que se encontram indevidamente vacinados. Tal pode ser também verificado na respectiva interrogação.

>> 25.

[*Maria do Olival, José Aristóteles*]

4 Conclusões e Sugestões

Finalmente, inferimos que este projeto mostrou-se bastante importante para proliferar e consolidar o que fomos aprendendo ao longo deste semestre. Sentimos que aplicamos e encontramos outras formas de desenvolver o léxico associado à linguagem *Prolog* aplicado a esta *UC* e que soubemos de forma inteligente utiliza-lo neste trabalho prático. Em retrospectiva achamos que temos um projeto bem organizado, com bastantes funcionalidades, cumprindo não só os objetivos mínimos mas também implementando diversas novas funcionalidades, tal como sugerido no enunciado deste TP. Como tal consideramos que o nosso projeto está bastante bem implementado, fundamentado e definido.

Bibliografia

- [ANALIDE et al., 2001] ANALIDE, C., NOVAIS, P., & NEVES, J. (2001). *Sugestões para a Elaboração de Relatórios*. Technical report, Departamento de Informática, Universidade do Minho, Portugal.
- [NOGATZ, 2020] NOGATZ, F. (2020). Logical arithmetic on dates and times. https://www.swi-prolog.org/pack/list?p=date_time.

Lista de Siglas e Acrónimos

TP Trabalho Prático

UC(s) Unidades(s) Curricular(es)

Anexos

Anexo 1

```
1 utente(1,21111111112,'Ana Murciellago',date(1930,1,1),'ana@gmail.com',911234567,'Braga',
2 'Médica',[],1,1).
3 utente(2,22111111112,'Maria do Olival',date(2001,11,5), 'maria@gmail.com',936484263, '
4 Viana do Castelo', 'Professora', ['asma'],1,1).
5 utente(3,01234567890,'Igor Marcos',date(1989,8,20),'marcos.mendes@gmail.com',
6 ,252252252,'R. Amaral 8499-852 São Mamede de Infesta','Agricultor',['hiv','
7 hepatite'],3,2).
8 utente(4,11111111111,'Rebeca Tavares Marques',date(2009,8,10),'maia.lara@gmail.com',
9 ,930643063,'Avenida St. Valentim Fonseca, nº 3, 94º Dir. 9794-701 Quarteira','
10 Estudante',['tuberculose'],2,3).
11 utente(5,09876543210,'Yasmin Sara Leite de Borges',date(1989,4,19),'dinis91@pereira.
12 info',236082002,'Avenida Leonardo Rodrigues, nº 2 3819-133 Aveiro','CEO Empresa de
13 Marketing',['brucellose','gripe canina'],4,4).
14 utente(6,66666666666,'José Aristoteles',date(1967,1,29),'exprimeiro@gov.pt',090909090,
15 'Alameda','Engenheiro',[],9,9).
16 utente(7,12345652858,'Custódio Borges',date(1987,4,15),'custodio@sapo.pt',251252254,'
17 Rua Esteves Vasto 23 Santo Tirso','Calceiteiro',['Hipertensão','Colesterol'],7,7).
18 utente(8,15615956260,'Francisco Carriço',date(1977,4,17),'carriçochico@yahoo.com',
19 ,255256258,'Rua Professor Batata Aracena','Advogado',['Diabetes','Artrite'],8,8).
20 utente(9,30861910266,'Jacinta Barboza',date(1982,2,13),'jacibar@gmail.com',256257269,'
21 Rua do Rei de Fafe, Fafe','Empresário',['Alzheimer','Doença crónica dos rins'
22 ],10,10).
23 centro_saude(1,'USFVida+', 'Vila Verde',253123456,'usfvida+@hotmail.com').
24 centro_saude(2,'UCSP Quarteira', 'Quarteira',244322111,'ucspquart@info.gmailcom').
25 centro_saude(3,'USF infesta', 'São Mamede Infesta', 254254252,'centrosaude@gmail.com').
26 centro_saude(4,'USF MOLICEIRO', 'Aveiro', 2512522523, 'aveirocsaude@saude.com').
27 centro_saude(5,'CS Penha de França', 'Rua Luís Pinto Moitinho,5 Lisboa', 224322123, '
28 cspenhafranca@gmail.com').
29 centro_saude(6,'Centro de Saude Caldas da Rainha', 'Rua do Centro de Saude, Caldas da
30 Rainha', 2412422423,'caldasofthequeen@gmail.com').
31 centro_saude(7,'Centro Saúde Lousã', 'Alameda Juiz Conselheiro Nunes Ribeiro,
32 Vilarinho',232123123,'vilarinholsa@minsaude.com').
33 centro_saude(8,'Centro de Saúde S.Pedro do Sul', 'Avenida da Ponte, São Pedro do Sul'
34 ,224123412, 'saintpetersouth@minsaude.com').
35 centro_saude(9,'Centro Saúde Ovar', 'Rua Dr. Francisco Zagalo s/n, OVAR', 245245245,'
36 ovarhashealth@minsaude.com').
37 centro_saude(10,'Centro de Saude Almodovar', 'Rua Professor Dr Fernando Padua,
38 ALMODOVAR', 263264265,'almodovaralmodovar@minsaude.com').
39
40 staff(1,1,'Maria Silva','maria@gmail.com').
41 staff(2,1,'Joana Guerra','joana@gmail.com').
42 staff(3,1,'Miguel Santos', 'santosmiguel@gmail.com').
43 staff(4,2,'Mario Sergio', 'sergiomario@gmail.com').
44 staff(5,2,'Ricardo Salgado', 'salgadoricardo@bancoes.com').
45 staff(6,3,'Joana Vasconcelos', 'arte@gmail.com').
46 staff(7,4,'Pedro Granger', 'apresentador@rtp.pt').
47 staff(8,5,'Luis Magalhaes', 'luismagalean@saudept.pt').
48 staff(9,6,' Mario Carvalhal', 'carvalhal@braga.pt').
49 staff(10,7,'Sergio Pedro', 'pedrosergio@csso.pt').
50 staff(11,8,'Ana Silva', 'anasilva123@sapo.pt').
51 staff(12,8,'Ricardo Tomé', 'riccardootome@yahoo.com').
52 staff(13,9,'Tiago Xavier', 'xavy99gem@riot.com').
53 staff(14,7,'Miguel Tavares', 'miguelsousatavares@tvi.pt').
54 staff(15,10,'Sophia Anderson', 'sophiamellobryner@denmark.de').
```

```

38 vacinacao_covid(1,1,date(2021,2,1),'Pfizer',1).
39 vacinacao_covid(2,1,date(2021,2,16),'Pfizer',2).
40 vacinacao_covid(1,2,date(2021,2,28),'Pfizer',1).
41 vacinacao_covid(6,3,date(2021,3,20),'AstraZeneca',1).
42 vacinacao_covid(4,4,date(2021,3,17),'Pfizer',1).
43 vacinacao_covid(5,4,date(2021,4,1),'Pfizer',2).
44 vacinacao_covid(6,3,date(2021,4,10),'AstraZeneca',2).
45 vacinacao_covid(7,6,date(2021,2,5),'AstraZeneca',1).
46
47 medico_familia(1,'Carlos Estevão','carlitos@gmail.com',1).
48 medico_familia(2,'Maria Silvana','silvana@gmail.com',3).
49 medico_familia(3,'Rogério Magalhães','magalhaes@gmail.com',2).
50 medico_familia(4,'Luis Ricciardi','ricciardi76@gmail.com',4).
51 medico_familia(5,'Joana Cunha','joanaCunha@gmail.com',5).
52 medico_familia(6,'Maria Suzete','suzetemaria@gmail.com',6).
53 medico_familia(7,'José Carlos Saraiva','zecarlos@gmail.com',7).
54 medico_familia(8,'Manuel Saramago','sarmanu@gmail.com',8).
55 medico_familia(9,'Ana Costa Silva','anacostafilomena@hotmail.com',9).
56 medico_familia(10,'Luisa Sobral','sobraleurovisi@almodovarcs.pt',10).
57 medico_familia(11,'Salvador Sentido','sentsentid@gmail.com',9).
58 medico_familia(12,'Manuel Rodrigues','sarmanu@sapo.pt',8).
59
60 consulta(1,1,2,'Consulta de rotina',15.00,date(2021,4,21)).
61 consulta(2,2,1,'Dores pós vacinação',7.50,date(2021,4,11)).
62 consulta(3,2,3,'Relatorio medicação HIV',15.00,date(2021,3,7)).
63 consulta(4,4,5,'Formulação de baixa laboral',70.00,date(2021,7,1)).
64 consulta(5,4,4,'Avaliacao de teste psicotécnico',20.70,date(2021,5,11)).
65 consulta(6,5,7,'Consulta de rotina',15.00,date(2021,11,3)).
66 consulta(7,7,8,'Consulta de rotina',15.00,date(2021,7,6)).
67 consulta(8,8,6,'Verificação exames diabéticos',15.00,date(2021,2,13)).
68 consulta(9,9,11,'Consulta de rotina',15.00,date(2021,1,11)).
69 consulta(10,9,11,'Marcação de exames rotineiros',5.00,date(2021,5,10)).

```

Listing 1: Base de conhecimento

Anexo 2

```

1 %-----
2 % Permitir a definição de fases de vacinação, definindo critérios de inclusão de
   utentes nas diferentes fases (e.g., doenças crónicas, idade, profissão);
3
4 fase(1,Id,date(2021,2,1)) :- idade(Id,I), I>=80.
5
6 fase(2,Id,date(2021,3,16)) :- utente(Id,_,_,_,_,_,_,DoencasCronicas,_,_),
   nao(vazia(DoencasCronicas)).
7
8
9 fase(3,Id,date(2021,4,29)) :-
10     utente(Id,_,_,_,_,_,_,Profissao,_,_,_),
11     pertence(Profissao,['Médica','Médico','Enfermeira','Enfermeiro','Auxiliar de Saúde
   ','Professora','Professor']).
12
13 fase(4,Id,date(2021,6,12)) :- utente(Id), nao(fase(1,Id,_)),
14     nao(fase(2,Id,_)), nao(fase(3,Id,_)).
15
16 %-----
17 % Identificar pessoas nao vacinadas
18 -vacinada(Id) :- utente(Id), nao(vacinacao_covid(_,Id)).
19
20 % extra
21 %-vacinada(Id) :- nao(vacinada(Id)), nao(excecao(vacinada(Id))).
22
23 %-----
24 % Extensão do predicado que identifica pessoas vacinadas
25
26 vacinada(Id) :- utente(Id), vacinacao_covid(_,Id).
27
28 %-----
29 % Extensão do predicado que identifica pessoas vacinadas indevidamente

```

```

30
31 -vacinada_indevidamente(Id) :- nao(vacinada_indevidamente(Id)).
32 vacinada_indevidamente(Id) :- vacinacao_covid(_,Id,DataVac,_,1), verificaFase(_,Id,
    DataFase), date_compare(DataFase,>,DataVac).
33
34 %-----
35 % Extensão do predicado que identifica pessoas não vacinadas e que são candidatas a
    vacinação
36
37 -candidata(Id) :- nao(candidata(Id)).
38 candidata(Id) :- utente(Id), nao(vacinada(Id)), date(DataAtual), verificaFase(_,Id,
    DataFase), date_compare(DataAtual,>=,DataFase).
39
40 %-----
41 %Extensão do predicado que identifica pessoas a quem falta a segunda toma da vacina
42
43 -segunda_toma(Id) :- nao(segunda_toma(Id)).
44 segunda_toma(Id) :- utente(Id), vacinada(Id,1), nao(vacinada(Id,2)).
45
46 %-----
47 % Desenvolver um sistema de inferência capaz de implementar os mecanismos de raciocí
    nio inerentes a estes sistemas.
48
49 si(Questao,verdadeiro) :- Questao.
50 si(Questao,falso) :- -Questao.
51 si(Questao,desconhecido) :- nao(Questao), nao(-Questao).

```

Listing 2: Funcionalidades básicas

Anexo 3

```

1 %-----
2 % Lista de pessoas numa fase de vacinação
3
4 faseLista(1,L,date(2021,2,1)) :- solucoes(Id,fase(1,Id,date(2021,2,1)),L).
5
6 faseLista(2,L,date(2021,3,16)) :- solucoes(Id,fase(2,Id,date(2021,3,16)),L).
7
8 faseLista(3,L,date(2021,4,29)) :- solucoes(Id,fase(3,Id,date(2021,4,29)),L).
9
10 faseLista(4,L,date(2021,6,12)) :- solucoes(Id,fase(4,Id,date(2021,6,12)),L).
11
12 %-----
13 % Identificar pessoas nao vacinadas numa certa toma
14 -vacinada(Id,1) :- utente(Id), nao(vacinacao_covid(_,Id,_,_,1)).
15 -vacinada(Id,2) :- utente(Id), nao(vacinacao_covid(_,Id,_,_,2)).
16
17 % extra
18 %-vacinada(Id,T) :- nao(vacinada(Id,T)), nao(excecao(vacinada(Id,T))).
19
20 % Lista das pessoas não vacinadas
21 naoVacinadas(S) :- solucoes(Id,-vacinada(Id),S).
22
23 %-----
24 % Identificar pessoas vacinadas numa certa toma
25 vacinada(Id,2) :- utente(Id), vacinacao_covid(_,Id,_,_,2).
26 vacinada(Id,1) :- utente(Id), vacinacao_covid(_,Id,_,_,1).
27
28 % Lista das pessoas vacinadas
29 vacinadas(S) :- solucoes(Id,vacinada(Id),S).
30
31 %-----
32 % Lista das pessoas que foram vacinadas indevidamente
33
34 vacinadas_indevidamente(S) :- solucoes(Id,vacinada_indevidamente(Id),S).
35
36 %-----
37 % Lista das pessoas que não foram vacinadas e são candidatas a vacinação

```

```

38
39 candidatas(S) :- solucoes(Id,candidata(Id),S).
40
41 %-----
42 % Lista de pessoas a quem falta a segunda toma da vacina
43
44 lista_segunda_toma(S) :- solucoes(Id,segunda_toma(Id),S).
45
46 %-----
47 % Custo total de consultas por utente
48
49 custo_consultas(Idutente,Total) :- utente(Idutente), solucoes(Custo,consulta(_,Id
    utente,_,_,Custo,_),S), sum_list(S,Total).
50
51 %-----
52 % Lista de centros de saude
53
54 centrosSaude(C) :- solucoes(Id-Centro,centro_saude(Id,Centro,_,_,_),S),
    sort(S,C).
55
56 %-----
57 %lista de utentes
58
59 utentes(C) :- solucoes(Utente,utente(_,_,Utente,_,_,_,_,_,_,_),S),
    sort(S,C).
60
61 %-----
62 % Lista de staff
63
64 staffs(C) :- solucoes(Staff,staff(_,_,Staff,_),S),
    sort(S,C).
65
66 %-----
67 % Lista de medicos
68
69 medicos(C) :- solucoes(Medico,medico_familia(_,Medico,_,_),S),
    sort(S,C).
70
71 %-----
72 % Lista de médicos de um determinado centro de saúde
73
74 medicosPorCentro(IdC,C) :- centro_saude(IdC), solucoes(Nome,medico_familia(_,Nome,_,
    IdC),S),
    sort(S,C).
75
76 %-----
77 % Lista de utentes de um determinado centro
78
79 utentesPorCentro(IdC,C) :- centro_saude(IdC), solucoes(Nome,utente(_,_,Nome,_,_,_,_,_,
    _,_,IdC),S),
    sort(S,C).
80
81 %-----
82 % Lista de pessoas do staff de um determinado centro
83
84 staffPorCentro(IdC,C) :- centro_saude(IdC), solucoes(Nome,staff(_,IdC,Nome,_),S),
    sort(S,C).
85
86 %-----
87
88
89
90

```

Listing 3: Predicados extra

Anexo 4

```

1 %-----
2 % Extensao do predicado que permite a evolucao do conhecimento
3
4 evolucao(Termo) :- solucoes(Invariante,+Termo::Invariante,Lista), insercao(Termo),
    teste(Lista).
5
6 insercao(Termo) :- assert(Termo).

```



```

7  insercao(Termo) :- retract(Termo),!,fail.
8
9  teste([]).
10 teste([R|LR]) :- R, teste(LR).
11
12 %-----
13 % Extensao do predicado que permite a involucao do conhecimento
14
15 involucao(Termo) :- solucoes(Invariante,-Termo::Invariante,Lista), remocao(Termo),
    teste(Lista).
16
17 remocao(Termo) :- retract(Termo).
18 remocao(Termo) :- assert(Termo),!,fail.
19
20 %-----
21 %---- Invariantes
22 %-----
23 % Não pode haver mais que um utente com o mesmo identificador
24 +utente(Idutente,_,_,_,_,_,_,_,_) ::
25     (solucoes(Idutente,utente(Idutente),S),
26      comprimento(S,N),
27      N==1).
28
29 %-----
30 % Não pode haver um utente registado num centro de saúde que não seja conhecido pelo
    sistema
31 +utente(.,.,.,.,.,.,.,.,Idcentro,_) ::
32     (solucoes(Idcentro,centro_saude(Idcentro),S),
33      comprimento(S,N),
34      N>0).
35
36 %-----
37 % Não pode haver um utente com um médico de família que não seja do seu centro de saúde
    de
38 +utente(.,.,.,.,.,.,.,.,Idcentro,Idmedico) ::
39     medico_familia(Idmedico,.,.,IdCentro).
40
41 %-----
42 % Não pode haver uma vacinação feita por uma pessoa do staff a um utente que não sejam
    conhecidos pelo sistema
43 +vacinacao_covid(Idstaff,Idutente,.,.,_) ::
44     (solucoes((Idstaff,Idutente),(staff(Idstaff),utente(Idutente)),S),
45      comprimento(S,N),
46      N==1).
47
48 %-----
49 % Não pode haver mais que uma vacinação com os mesmos dados
50 +vacinacao_covid(Idstaff,Idutente,Data,Vacina,Toma) ::
51     (solucoes((Idstaff,Idutente),vacinacao_covid(Idstaff,Idutente,Data,
52      Vacina,Toma),S),
53      comprimento(S,N),
54      N==1).
55
56 %-----
57 % Não pode haver uma segunda toma da vacina sem haver a primeira
58 +vacinacao_covid(Idstaff,Idutente,Data,Vacina,2) ::
59     (solucoes((Idstaff,Idutente),vacinacao_covid(Idstaff,Idutente,Data,
60      Vacina,1),S),
61      comprimento(S,N),
62      N==1).
63
64 %-----
65 % Não pode haver mais que um centro de saúde com o mesmo identificador
66 +centro_saude(Idcentro,.,.,_,_) ::
67     (solucoes(Idcentro,centro_saude(Idcentro),S),
68      comprimento(S,N),
69      N==1).
70
71 %-----
72 % Não pode haver mais que um centro de saúde com os mesmos dados
73 +centro_saude(Idcentro,Nome,Morada,Telefone,Email) ::

```

```

72         (solucoes(Idcentro,centro_saude(Idcentro,Nome,Morada,Telefone,Email),S
73         ),
74         comprimento(S,N),
75         N==1).
76 %-----
77 % Não pode haver mais que um staff com o mesmo identificador
78 +staff(Idstaff,_,_,_) ::
79         (solucoes(Idstaff,staff(Idstaff),S),
80         comprimento(S,N),
81         N==1).
82 %-----
83 % Não pode haver staff de um centro de saúde que não é conhecido pelo sistema
84 +staff(_,Idcentro,_,_) ::
85         (solucoes(Idcentro,centro_saude(Idcentro),S),
86         comprimento(S,N),
87         N>0).
88 %-----
89 % Não pode haver mais que um medico com o mesmo identificador
90 +medico_familia(Idmedico,_,_,_) ::
91         (solucoes(Idmedico,medico_familia(Idmedico),S),
92         comprimento(S,N),
93         N==1).
94 %-----
95 % Não pode haver mais que uma consulta com o mesmo identificador
96 +consulta(Idconsulta,_,_,_,_,_) ::
97         (solucoes(Idconsulta,consulta(Idconsulta),S),
98         comprimento(S,N),
99         N==1).
100 %-----
101 % Não pode haver uma consulta feita por um médico a um utente que não sejam conhecidos
102 % pelo sistema
103 +consulta(_,Idutente,Idmedico,_,_,_) ::
104         (solucoes((Idmedico,Idutente),(medico_familia(Idmedico),utente(Id
105         utente)),S),
106         comprimento(S,N),
107         N==1).
108 %-----
109 % Não pode ser retirado um utente que ainda não tenha a vacinação completa
110 -utente(Idutente,_,_,_,_,_,_,_,_) ::
111         (vacinacao_covid(_,Idutente,_,_,2)).
112 %-----
113 % Não pode ser retirado um centro de saúde que tenha utentes registados
114 -centro_saude(Idcentro,_,_,_,_) ::
115         (solucoes(Idutente,(utente(Idutente,_,_,_,_,_,_,_,Idcentro,_)),S),
116         comprimento(S,N),
117         N==0).
118 %-----
119 % Não pode ser retirado um médico de família que seja médico de um utente registado
120 -medico_familia(Idmedico,_,_,_) ::
121         (solucoes(Idutente,utente(Idutente,_,_,_,_,_,_,_,Idmedico),S),
122         comprimento(S,N),
123         N==0).
124 %-----
125 %----- Registos
126 %-----
127 % utente
128 registrarUtente(Idutente,NISS,Nome,Data_Nasc,Email,Telefone,Morada,Profissao,
129         DoencasCronicas,Idcentro,Idmedico) :-
130         evolucao(utente(Idutente,NISS,Nome,Data_Nasc,Email,Telefone,Morada,Profissao,
131         DoencasCronicas,Idcentro,Idmedico)).
132 %-----
133 % centro_saude
134 registrarCentro(Id,Nome,Morada,Telefone,Email) :-

```

```

139     evolucao(centro_saude(Id, Nome, Morada, Telefone, Email)).
140
141 % staff
142 registrarStaff(Idstaff, Idcentro, Nome, Email) :-
143     evolucao(staff(Idstaff, Idcentro, Nome, Email)).
144
145 % vacinacao_covid
146 registrarVacinacao(Idstaff, Idutente, Data, Vacina, Toma) :-
147     evolucao(vacinacao_covid(Idstaff, Idutente, Data, Vacina, Toma)).
148
149 % medico_familia
150 registrarMedico(IdMedico, Nome, Email, IdCentro) :-
151     evolucao(medico_familia(Idstaff, Nome, Email, IdCentro)).
152
153 % consulta
154 registrarConsulta(IdConsulta, IdUtente, IdMedico, Descricao, Custo, Data) :-
155     evolucao(consulta(IdConsulta, IdUtente, IdMedico, Descricao, Custo, Data)).
156
157 %-----
158 %---- Remoções
159 %-----
160 % utente
161 removerUtente(Idutente) :-
162     utente(Idutente, NISS, Nome, Data_Nasc, Email, Telefone, Morada, Profissao,
163         DoencasCronicas, Idcentro, Idmedico),
164     involucao(utente(Idutente, NISS, Nome, Data_Nasc, Email, Telefone, Morada, Profissao,
165         DoencasCronicas, Idcentro, Idmedico)).
166
167 % centro_saude
168 removerCentro(Idcentro) :-
169     centro_saude(Idcentro, Nome, Morada, Telefone, Email),
170     involucao(centro_saude(Idcentro, Nome, Morada, Telefone, Email)).
171
172 % staff
173 removerStaff(Idstaff) :-
174     staff(Idstaff, Idcentro, Nome, Email),
175     involucao(staff(Idstaff, Idcentro, Nome, Email)).
176
177 % vacinacao_covid
178 removerVacinacao(Idstaff, Idutente, Data, Vacina, Toma) :-
179     involucao(vacinacao_covid(Idstaff, Idutente, Data, Vacina, Toma)).
180
181 % medico_familia
182 removerMedico(Idmedico) :-
183     medico_familia(Idmedico, Nome, Email, IdCentro),
184     involucao(medico_familia(Idmedico, Nome, Email, IdCentro)).
185
186 % consulta
187 removerConsulta(Idconsulta) :-
188     consulta(Idconsulta, Idutente, Idmedico, Descricao, Custo, Data),
189     involucao(consulta(Idconsulta, Idutente, Idmedico, Descricao, Custo, Data)).

```

Listing 4: Mecanismos de controlo de consistência e evolução de conhecimento

Anexo 5

```

1 % ----- Menu -----
2 menu:- write('\n'),
3         write('-----MENU-----\n'),
4         write('\n'),
5         write('-----Inserir-----\n'),
6         write('1.Registar Utente \n'),
7         write('2.Registar Centro de Saúde \n'),
8         write('3.Registar Elemento do Staff \n'),
9         write('4.Registar Vacinação \n'),
10        write('5.Registar Médico de Família \n'),
11        write('6.Registar Consulta \n'),
12        write('\n'),

```

```

13     write('-----Remover-----\n'),
14     write('7.Remover Utente \n'),
15     write('8.Remover Centro de Saúde \n'),
16     write('9.Remover Elemento do Staff \n'),
17     write('10.Remover Vacinação \n'),
18     write('11.Remover Médico de Família\n'),
19     write('12.Remover Consulta\n'),
20     write('\n'),
21     write('-----Consultar-----\n'),
22     write('13.Custo total de consultas por utente\n'),
23     write('14.Consultar se um utente já foi vacinado\n'),
24     write('15.Consultar se um utente ainda não foi vacinado\n'),
25     write('16.Consultar em que fase vai ser vacinado\n'),
26     write('17.Consultar candidata em vacinação na fase a decorrer\n'),
27     write('18.Consultar se a pessoa ainda não tem a 2ª dose\n'),
28     write('19.Consultar se a pessoa foi vacinada indevidamente\n'),
29     write('-----Listar-----\n'),
30     write('20.Listar os médicos por centro de saúde\n'),
31     write('21.Listar os utentes por centro de saúde\n'),
32     write('22.Listar o staff por centro de saúde\n'),
33     write('23.Listar as pessoas não vacinadas\n'),
34     write('24.Listar as pessoas vacinadas\n'),
35     write('25.Listar as pessoas vacinadas indevidamente\n'),
36     write('26.Listar as pessoas candidatas a vacinação\n'),
37     write('27.Listar as pessoas a quem falta a segunda toma\n'),
38     write('0.Sair \n'),
39     write('>> '),
40     read(Option),
41     executar(Option).
42
43 executar(Option):-Option==1,addUtente,menu;
44                  Option==2,addCentro,menu;
45                  Option==3,addStaff,menu;
46                  Option==4,addVacinacao,menu;
47                  Option==5,addMedico,menu;
48                  Option==6,addConsulta,menu;
49                  Option==7,deleteUtente,menu;
50                  Option==8,deleteCentro,menu;
51                  Option==9,deleteStaff,menu;
52                  Option==10,deleteVacinacao,menu;
53                  Option==11,deleteMedico,menu;
54                  Option==12,deleteConsulta,menu;
55                  Option==13,consultCustoConsulta,menu;
56                  Option==14,consultPessoaVacinada,menu;
57                  Option==15,consultPessoaNaoVacinada,menu;
58                  Option==16,consultFaseVacinacao,menu;
59                  Option==17,consultCandidataVacinacao,menu;
60                  Option==18,consultSegundaToma,menu;
61                  Option==19,consultVacinadaInfevidamente,menu;
62                  Option==20,listagemMedicosPorCentro,menu;
63                  Option==21,listagemUtentesPorCentro,menu;
64                  Option==22,listagemStaffPorCentro,menu;
65                  Option==23,listagemPessoasNaoVacinadas,menu;
66                  Option==24,listagemVacinadas,menu;
67                  Option==25,listagemVacinadasIndevidamente,menu;
68                  Option==26,listagemPessoasCandidatas,menu;
69                  Option==27,listagemPessoasSemSegundaToma,menu;
70                  Option==99,true,make,menu;
71                  Option==0,true,write('Goodbye.'),halt.
72
73 /* ----- INSERÇÕES ----- */
74 addUtente :- write('Id Utente: '),read(Idutente), (utente(Idutente),
75              write('\nId de utente já existente!\nTente novamente.\n'));
76              nao(utente(Idutente)),
77              write('NSS: '),read(NISS),
78              write('Nome (entre plicas): '),read(Nome),
79              write('Data de Nascimento: (no formato date (Ano,Mes,Dia))'),read(Data_Nasc
80              ),
81              write('Email (entre plicas): '),read(Email),
82              write('Telefone: '),read(Telefone),
83              write('Morada (entre plicas): '),read(Morada),
              write('Profissao (entre plicas): '),read(Profissao),

```

```

84         write('Lista de doenças crônicas (no formato [doença1,doença2,...] ou []
           se nenhuma): '),read(DoençasCronicas),
85         write('Id Centro de Saúde: '),read(Idcentro),
86         write('Id Médico de Família: '),read(Idmedico),
87         registrarUtente(Idutente,NISS,Nome,Data_Nasc,Email,Telefone,Morada,
           Profissao,DoencasCronicas,Idcentro,Idmedico),
88         ansi_format([bold,fg(green)], 'Sucesso! ~w', ['\n'])).
89
90 addCentro :- write('Id Centro de Saúde: '), read(Idcentro), (centro_saude(Idcentro),
91 write('\nId do Centro de Saúde já existente!\nTente novamente.\n');
92 nao(centro_saude(Idcentro)),
93 write('Nome (entre plicas): '),read(Nome),
94 write('Morada (entre plicas): '),read(Morada),
95 write('Telefone: '),read(Telefone),
96 write('E-mail (entre plicas): '),read(Email),
97 registrarCentro(Idcentro,Nome,Morada,Telefone,Email),
98 ansi_format([bold,fg(green)], 'Sucesso! ~w', ['\n'])).
99
100 addStaff :- write('Id Staff: '),read(Idstaff), (staff(Idstaff),
101 write('\nId de Staff já existente!\nTente novamente.\n');
102 nao(staff(Idstaff)),
103 write('Id Centro: '),read(Idcentro),
104 write('Nome (entre plicas): '),read(Nome),
105 write('Email (entre plicas): '),read(Email),
106 registrarStaff(Idstaff,Idcentro,Nome,Email),
107 ansi_format([bold,fg(green)], 'Sucesso! ~w', ['\n'])).
108
109 addVacinaçao :- write('Id Staff: '),read(Idstaff),
110 write('Id Utente: '),read(Idutente),
111 write('Data (no formato date(Ano,Mes,Dia)): '),read(Data),
112 write('Vacina (entre plicas): '),read(Vacina),
113 write('Toma: '),read(Toma),
114 registrarVacinaçao(Idstaff,Idutente,Data,Vacina,Toma),
115 ansi_format([bold,fg(green)], 'Sucesso! ~w', ['\n'])).
116
117 addMedico :- write('Id Médico: '), read(Idmedico), (medico_familia(Idmedico),
118 write('\nId de médico já existente!\nTente novamente.\n');
119 nao(medico_familia(Idmedico)),
120 write('Nome (entre plicas): '),read(Nome),
121 write('Email (entre plicas): '),read(Email),
122 write('Id Centro: '),read(Idcentro),
123 registrarMedico(Idmedico,Nome,Email,Idcentro),
124 ansi_format([bold,fg(green)], 'Sucesso! ~w', ['\n'])).
125
126 addConsulta :- write('Id Consulta: '), read(Idconsulta), (consulta(Idconsulta),
127 write('\nId de médico já existente!\nTente novamente.\n');
128 nao(consulta(Idconsulta)),
129 write('Id Utente: '),read(Idutente),
130 write('Id Médico: '),read(Idmedico),
131 write('Descrição do motivo da consulta (entre plicas): '), read(
           Descricao),
132 write('Custo da consulta: '), read(Custo),
133 write('Data da consulta (no formato date(Ano,Mes,Dia)): '), read(Data),
134 registrarConsulta(Idconsulta,Idutente,Idmedico,Descricao,Custo,Data),
135 ansi_format([bold,fg(green)], 'Sucesso! ~w', ['\n'])).
136
137 /* ----- REMOÇÕES ----- */
138 deleteUtente :- write('Id Utente: '),read(Idutente),
139 removeUtente(Idutente),
140 ansi_format([bold,fg(green)], 'Sucesso! ~w', ['\n'])).
141
142 deleteCentro :- write('Id Centro de Saúde: '),read(Idcentro),
143 removeCentro(Idcentro),
144 ansi_format([bold,fg(green)], 'Sucesso! ~w', ['\n'])).
145
146 deleteStaff :- write('Id Staff: '),read(Idstaff),
147 removeStaff(Idstaff),
148 ansi_format([bold,fg(green)], 'Sucesso! ~w', ['\n'])).
149
150 deleteVacinaçao :- write('Id Staff: '),read(Idstaff),
151 write('Id Utente: '),read(Idutente),
152 write('Data (no formato date(Ano,Mes,Dia)): '),read(Data),

```

```

153         write('Vacina (entre plicas): '),read(Vacina),
154         write('Toma: '),read(Toma),
155         removerVacinacao(Idstaff,Idutente,Data,Vacina,Toma),
156         ansi_format([bold,fg(green)], 'Sucesso! ~w', ['\n']).
157
158 deleteMedico:-write('Id médico: '), read(Idmedico),
159             removerMedico(Idmedico),
160             ansi_format([bold,fg(green)], 'Sucesso! ~w', ['\n']).
161
162 deleteConsulta:-write('Id Consulta: '),read(Idconsulta),
163             removerConsulta(Idconsulta),
164             ansi_format([bold,fg(green)], 'Sucesso! ~w', ['\n']).
165
166 /* ----- CONSULTAS ----- */
167 consultCustoConsulta:-write('Id Utente: '),read(Idutente),
168             custo_consultas(Idutente, Total),
169             ansi_format([bold,fg(cyan)], 'Total: ~w€\n', [Total]).
170
171 consultPessoaVacinada:-write('Id Utente: '),read(Idutente),
172             ((si(vacinada(Idutente), verdadeiro),
173             ansi_format([bold,fg(green)], 'Verdadeiro ~w', ['\n'])));
174             (si(vacinada(Idutente), falso),
175             ansi_format([bold,fg(red)], 'Falso ~w', ['\n'])))).
176
177 consultPessoaNaoVacinada:-write('Id Utente: '),read(Idutente),
178             ((si(vacinada(Idutente), falso),
179             ansi_format([bold,fg(green)], 'Verdadeiro ~w', ['\n'])));
180             (si(vacinada(Idutente), verdadeiro),
181             ansi_format([bold,fg(red)], 'Falso ~w', ['\n'])))).
182
183 consultFaseVacinacao:-write('Id Utente: '),read(Idutente),
184             verificaFase(F, Idutente, _),
185             ansi_format([bold,fg(cyan)], 'Será vacinado na fase ~w\n', [F]).
186
187 consultCandidataVacinacao:-write('Id Utente: '),read(Idutente),
188             ((si(candidata(Idutente), verdadeiro),
189             ansi_format([bold,fg(green)], 'Verdadeiro ~w', ['\n'])));
190             (si(candidata(Idutente), falso),
191             ansi_format([bold,fg(red)], 'Falso ~w', ['\n'])))).
192
193 consultSegundaToma:-write('Id Utente: '),read(Idutente),
194             ((si(segunda_toma(Idutente), verdadeiro),
195             ansi_format([bold,fg(green)], 'Verdadeiro ~w', ['\n'])));
196             (si(segunda_toma(Idutente), falso),
197             ansi_format([bold,fg(red)], 'Falso ~w', ['\n'])))).
198
199 consultVacinadaInfevidamente:-write('Id Utente: '),read(Idutente),
200             ((si(vacinada_infevidamente(Idutente), verdadeiro),
201             ansi_format([bold,fg(green)], 'Verdadeiro ~w', ['\n'])));
202             (si(vacinada_infevidamente(Idutente), falso),
203             ansi_format([bold,fg(red)], 'Falso ~w', ['\n'])))).
204
205 /* ----- LISTAGEM ----- */
206 listagemMedicosPorCentro:-centrosSaude(Centros), print(Centros),
207             write('Id Centro: '),read(CentroID),
208             medicosPorCentro(CentroID, Medicos),
209             ansi_format([bold,fg(cyan)], '~w\n', [Medicos]).
210
211 listagemUtentesPorCentro:-centrosSaude(Centros), print(Centros),
212             write('Id Centro: '),read(CentroID),
213             utentesPorCentro(CentroID, Utentes),
214             ansi_format([bold,fg(cyan)], '~w\n', [Utentes]).
215
216 listagemStaffPorCentro:-centrosSaude(Centros), print(Centros),
217             write('Id Centro: '),read(CentroID),
218             staffPorCentro(CentroID, Staff),
219             ansi_format([bold,fg(cyan)], '~w\n', [Staff]).
220
221 listagemPessoasNaoVacinadas:-naoVacinadas(Pessoas),
222             getNames(Pessoas,R),
223             ansi_format([bold,fg(cyan)], '~w\n', [R]).
224

```



```

47 getNames([E], [Nome]):- utente(E, Nome).
48 getNames([H|T], [Nome|R]):- utente(H, Nome), getNames(T,R).
49
50 %-----
51 % Verifica qual a fase em que um utente está incluído
52
53 verificaFase(F, Id, Data) :- fase(F, Id, Data), !.

```

Listing 6: Predicados auxiliares