



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Trabalho Prático 1

Mestrado Integrado em Engenharia Informática

SISTEMAS DE REPRESENTAÇÃO
DE CONHECIMENTO E RACIOCÍNIO
(2º semestre, 2020/21)

a83700	António Santos
a84912	Joana A. Gomes
a84240	Jorge Vieira
a85852	José Magalhães
a86475	Paulo R. Pereira

Braga
Abril 2021

Resumo

O presente relatório é referente ao primeiro trabalho prático de grupo da unidade curricular de Sistemas de Representação de Conhecimento e Raciocínio.

Este trabalho prático tem como objetivo melhorar as competências na utilização da linguagem de programação em lógica PROLOG, no âmbito da representação de conhecimento e construção de mecanismos de raciocínio para a resolução de problemas.

Será então apresentado o problema proposto, a sua contextualização e a proposta de resolução adotada pelo grupo.

Toda a base de conhecimento, bem como os predicados que a constituem e os invariantes estruturais e referenciais, serão devidamente referidos e explicados.

Será também analisada a adição e remoção de conhecimento e algumas funcionalidades extras implementadas para uma abordagem mais complexa do problema.

CONTEÚDO

1	Introdução	3
2	Preliminares	5
3	Descrição do Trabalho e Análise de Resultados	6
3.1	Contextualização	6
3.1.1	Decisões iniciais	7
3.2	Base de Conhecimento	8
3.2.1	Predicados Auxiliares	8
3.3	Invariantes	10
3.3.1	Invariantes de Utente	10
3.3.2	Invariantes de Centro	11
3.3.3	Invariantes de Staff	12
3.3.4	Invariantes de Vacinação	12
3.3.5	Invariantes de Fase	13
3.4	Adicionar conhecimento	13
3.4.1	Utente	13
3.4.2	Staff	14
3.4.3	Centro de saúde	14
3.4.4	Vacinação	14
3.5	Remover conhecimento	15
3.5.1	Utente	15
3.5.2	Staff	15
3.5.3	Centro de Saúde	16
3.5.4	Vacinação	16
3.6	Identificações	16

3.6.1	Identificar utentes não vacinados (por fases)	16
3.6.2	Identificar utentes vacinados	21
3.6.3	Vacinação Indevida	21
4	Conclusões e Sugestões	25
A	Código fonte	27
A.1	auxiliares.pl	27
A.2	dados.pl	30
A.3	tp1.pl	41

INTRODUÇÃO

No âmbito da Unidade Curricular de **Sistemas de Representação de Conhecimento e Raciocínio** foi-nos proposto 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 COVID que estamos a viver.

A construção do caso prático deve ser capaz de demonstrar as seguintes funcionalidades, subjacentes à utilização da linguagem de programação em lógica PROLOG:

- permitir a definição de fases de vacinação, definindo critérios de inclusão de utentes nas diferentes fases (e.g. doenças crónicas, idade, profissão);
- identificar pessoas não vacinadas;
- identificar pessoas vacinadas;
- identificar pessoas vacinadas indevidamente;
- identificar pessoas não vacinadas e que são candidatas a vacinação;
- identificar pessoas a quem falta a segunda toma da vacina;
- Desenvolver um sistema de inferência capaz de implementar os mecanismos de raciocínio inerentes a estes sistemas.

Como seria de esperar, o último tópico é quase como uma injunção do trabalho prático, tendo em conta que a linguagem de programação PROLOG é um motor de inferência lógica cujo objetivo é responder a questões, questões essas que serão respondidas com base nos predicados cuja base é a implementação das funcionalidades do trabalho.

PRELIMINARES

De forma a implementar uma boa resolução, foi necessário analisar o caso real das vacinações para a pandemia COVID, isto é, analisar as regras de vacinação, desde as diferentes fases até às implicações de cada fase, por exemplo, doenças e profissões associadas.

Foi também necessário estudar em detalhe o motor de inferência PROLOG de forma a explorar mais funcionalidades desta linguagem de programação em lógica.

Reconhecendo a importância das aulas como a base da aprendizagem, o grupo achou por bem ir um pouco além do que foi aprendido nas mesmas, não só para benefício académico e profissional, mas também para facilitar algumas implementações, conforme será explicado no capítulo seguinte.

3

DESCRIÇÃO DO TRABALHO E ANÁLISE DE RESULTADOS

Este capítulo está dividido de forma a que o leitor entenda bem a forma como o trabalho está organizado. Inicialmente é feita uma pequena contextualização do caso real, e depois, na secção **Base de Conhecimento**, abordamos os predicados que constituem a base de conhecimento do nosso sistema, na secção **Invariantes**, abordamos o controlo da inserção e remoção de conhecimento e nas secções **Adicionar Conhecimento** e **Remover Conhecimento**, abordamos os procedimentos para tal. Por fim, no capítulo **Identificações** é abordada a implementação das funcionalidades do nosso sistema, cujos resultados e a sua análise são descritos na última secção, **Análise de Resultados**.

3.1 Contextualização

Nos dias que correm, a sociedade vive uma fase difícil provocada pela pandemia COVID-19. Para travar esta pandemia, Portugal teve que desenvolver um plano de vacinação com as vacinas disponíveis até ao momento, distribuindo a população por fases de vacinação segundo várias restrições.

3.1.1 Decisões iniciais

Fases

O grupo decidiu distribuir os utentes pelas diferentes fases baseando-se em conceitos que essencialmente espelham a realidade da situação atual do plano de vacinação, adaptando apenas alguns dados que ainda não se encontram totalmente delineados.

Assim, foram criadas 3 fases de vacinação com os intervalos apresentados de seguida, sendo que o sistema está preparado para possíveis alterações ao nível deste conhecimento.

- **Fase 1:** (de 01-12-2020 a 31-03-2021)
 - Profissionais de Saúde;
 - Pessoas com idade igual ou superior a 80 anos;
 - Pessoas com idade igual ou superior a 50 anos e uma das seguintes doenças: *{Insuficiência Cardíaca, Doença Coronária, Insuficiência Renal, DPOC}*
- **Fase 2:** (de 01-04-2021 a 31-08-2021)
 - Pessoas com idade igual ou superior a 65 anos;
 - Pessoas com idade compreendida entre 50 e 64 anos e uma das seguintes doenças: *{Diabetes, Neoplasia maligna ativa, Doença Renal Crónica, Insuficiência Hepática, Hipertensão arterial, Obesidade}*
- **Fase 3:** (a partir de 01-09-2021)
 - Restantes utentes.

Toma indevida da Vacina

O grupo definiu ainda critérios que distinguem utentes que foram vacinados indevidamente.

1. Tomou numa fase que não lhe correspondia, ou seja, não encaixa nos critérios para a toma nessa fase (Ex: *Um utente com menos de 50 anos tomou a vacina na Fase 2*);

2. Tomou duas vacinas distintas (Ex: *Um utente tomou a vacina da Pfizer na primeira fase e da Moderna na segunda fase*);
3. Tomou mais que duas vacinas.

A toma da vacina por um utente considera-se indevida se um ou mais destes critérios se verificar.

3.2 Base de Conhecimento

O panorama será caracterizado por conhecimento dado na seguinte forma:

- **utente:** #IdUtente, Número_Segurança_Social, Nome, Género, Data_Nascimento, Email, Telefone, Morada, Profissão, [Doenças_Crónicas], #IdCentroSaúde $\leadsto \{\mathbb{V}, \mathbb{F}\}$
- **centro:** #IdCentroSaúde, Nome, Morada, Telefone, Email $\leadsto \{\mathbb{V}, \mathbb{F}\}$
- **staff:** #IdStaff, IdCentroSaúde, Nome, email $\leadsto \{\mathbb{V}, \mathbb{F}\}$
- **vacinação:** #staf, #utente, Data, Vacina, Toma $\leadsto \{\mathbb{V}, \mathbb{F}\}$
- **fase:** #IdFase, DataInicio, DataFim $\leadsto \{\mathbb{V}, \mathbb{F}\}$

O conhecimento foi estendido (adição de novos predicados). Além de incluir o *género* no utente, foi também adicionado o predicado **fase**, de forma a facilitar o cumprimento dos objetivos propostos.

Desta forma, o nosso sistema foi “povoado” pela adição dos seguintes factos, conforme é possível ver no anexo A.2.

3.2.1 Predicados Auxiliares

O grupo criou alguns predicados que, não tendo diretamente a ver com a área do projeto, foram úteis para o suporte de desenvolvimento de outras funcionalidades.

De entre estas predicados auxiliares, destacamos o **calcularIdade**, que implementa o cálculo da idade de um determinado utente, dado como argumento o seu ID. Este é especialmente útil na atribuição da fase de vacinação correspondente ao utente, sendo os critérios muito baseados na idade dos utentes.

```
calcularIdade(IdU,R) :- utente(IdU,_,_,_,DNasc,_,_,_,_,_),
    split_string(DNasc, "-", "", Sp),
    nth0(0,Sp,Ano), nth0(1,Sp,Mes), nth0(2,Sp,Dia),
    get_date_time_value(year, AnoN),
    ↪ get_date_time_value(month, MesN),
    get_date_time_value(day, DiaN),
    atom_number(Ano, Year), atom_number(Mes, Month),
    ↪ atom_number(Dia, Day),
    atom_string(AnoN, YearN), atom_number(YearN,
    ↪ YearNN),
    atom_string(MesN, MonthN), atom_number(MonthN,
    ↪ MonthNN),
    atom_string(DiaN, DayN), atom_number(DayN, DayNN),
    Age is (YearNN - Year),
    (Month > MonthNN -> AgeN is Age-1 ;
    Month < MonthNN -> AgeN is Age ;
    (Day > DayNN -> AgeN is Age-1 ;
    AgeN is Age)),
    R = AgeN.
```

% Data atual

```
get_date_time_value(Key, Value) :-
    get_time(Stamp),
    stamp_date_time(Stamp, DateTime, local),
    date_time_value(Key, DateTime, Value).
```

Salienta-se ainda o predicado **comparaDatasStr** que compara duas datas no formato escolhido para a base de conhecimento, comparando-as de modo a indicar se uma é antes da outra.

```

comparaDatasStr(X1,X2,R):- parseData(X1,A1,M1,D1),
    ↪ parseData(X2,A2,M2,D2), comparaDatas(A1,M1,D1,A2,M2,D2,R).

% Parse dos dados
parseData(K,A,M,D) :- split_string(K, "-", "", Sp),
    nth0(0,Sp,Ano), nth0(1,Sp,Mes), nth0(2,Sp,Dia),
    atom_number(Ano, Year), atom_number(Mes, Month),
    ↪ atom_number(Dia, Day),
    (A is Year, M is Month, D is Day).

% Verifica se a data 1 é antes de data 2
comparaDatas(A1,_,_,A2,_,_,0):- A1 > A2.
comparaDatas(A1,M1,_,A2,M2,_,0):- M1 > M2, A1 >= A2.
comparaDatas(A1,M1,D1,A2,M2,D2,0):- D1 > D2, A1 >= A2, M1 >= M2.
comparaDatas(_,_,_,_,_,_,1).

```

3.3 Invariantes

Para que os predicados de inserção e remoção não criem inconsistências na nossa base de conhecimento, foi necessário criar invariantes, estruturais e referenciais, que mantenham a base de conhecimento coerente e consistente.

3.3.1 Invariantes de Utente

- Garantir que o ID de cada utente é único:

```

+utente(Id,A,B,C,D,E,F,G,H,I,J) ::
    ↪ (solucoes(Id,utente(Id,_,_,_,_,_,_,_,_,_,_), R),
        length(R, 1)).

```

- Garantir que o género só pode ser M ou F:

```
+utente(_,_,_,G,_,_,_,_,_,_,_) :: generoValido(G).
```

```
generoValido('M').
```

```
generoValido('F').
```

- Garantir que não se pode eliminar um utente se tiver vacinações marcadas:

```
-utente(Id,_,_,_,_,_,_,_,_,_,_) :: (findall(Id,
  ↳ vacinacao(_,Id,_,_,_), R),
  ↳ \+length(R, 0)).
```

- Garantir que utentes com IDs diferentes têm diferente informação (número de segurança social, email, telefone):

```
+utente(Id,N,NSS,G,DN,E,TLF,M,P,D,IDC) ::
  ↳ (solucoes((Id,NSS,E,TLF),
  ↳ utente(_,_,NSS,_,_,E,TLF,_,_,_,_), R),
  ↳ length(R,1)).
```

- Os utentes têm de ter um ID de centro existente:

```
+utente(_,_,_,_,_,_,_,_,_,_,IdC) :: centro(IdC,_,_,_,_,_).
```

3.3.2 Invariantes de Centro

- Garantir que o ID de cada centro é único:

```
+centro(Id,A,B,C,D) :: (solucoes(Id, centro(Id,_,_,_,_), R),
  ↳ length(R, 1)).
```

- Garantir que não se pode eliminar um centro se este tiver uma vacinação associada:

```
-centro(Id,_,_,_,_) :: (findall(sId,staff(sId,_,_,_,Id),R),
  ↳ \+length(R, 0)).
```

- Garantir que centros com IDs diferentes têm diferente informação (nome, morada, telefone, email):

```
+centro(Id,N,_,T,E) :: (solucoes((Id,N,T,E),
  ↪  centro(_,N,_,T,E), R),
                                     length(R,1)).
```

3.3.3 Invariantes de Staff

- Garantir que o ID de cada staff é único:

```
+staff(Id,A,B,C) :: (solucoes(Id, staff(Id,_,_,_), R),
                                     length(R, 1)).
```

- Garantir que não se pode eliminar um staff se este tiver uma vacinação associada

```
-staff(Id,_,_,_,_) :: (findall(Id, vaccinacao(Id,_,_,_,_), R),
                      \+length(R, 0)).
```

- Garantir que staff com IDs diferentes têm diferente informação (ID do centro de saúde, email):

```
+staff(Id,_,SS,T,C) :: (solucoes((SS,T,C), staff(_,_,SS,T,C),
  ↪  R),
                                     length(R,1)).
```

3.3.4 Invariantes de Vacinação

- Garantir que não há vacinações repetidas:

```
+vaccinacao(A,B,C,D,E) :: (solucoes(Id, vaccinacao(A,B,C,D,E),
  ↪  R),
                                     length(R, 1)).
```

- Adicionar uma vacinação da toma 2 requer que a toma 1 já esteja na base de conhecimento:

```
+vacinacao(S,U,_,_,2) :: vacinacao(S,U,_,_,1).
```

3.3.5 Invariantes de Fase

- Garantir que o ID de cada fase é único:

```
+fase(Id,A,B) :: (solucoes(Id, fase(Id,_,_), R),  
                  length(R, 1)).
```

- Garantir que a data de início tem de ser antes da data de fim:

```
+fase(_,DI,DF) :: comparaDatasStr(DI,DF).
```

3.4 Adicionar conhecimento

Adicionar conhecimento deve-se ao meta predicado *novoConhecimento* definido da seguinte forma:

```
novoConhecimento(T) :- solucoes(I, +T::I, L),  
                      insercao(T),  
                      testaPredicados(L).
```

```
insercao(Q) :- assert(Q).  
insercao(Q) :- retract(Q), !, fail.
```

e também aos invariantes estruturais referidos em cima. Observe-se os vários predicados que complementam este meta predicado.

3.4.1 Utente

Para adicionar um novo utente à base de conhecimento, é necessário invocar o predicado **novoUtente** com os devidos parâmetros:

```
nov Utente(Id,NISS,N,G,Dn,E,Tlf,M,P,Dc,IdCs) :-  
    novoConhecimento(utente(Id,NISS,N,G,Dn,E,Tlf,M,P,Dc,IdCs)).
```

Para termos a certeza que os dados que estamos a inserir na base de conhecimento são válidos, fazemos uso de vários invariantes, que têm que continuar a ser válidos depois da inserção de um novo utente, estes invariantes já foram explicitados na secção prévia.

3.4.2 Staff

No registo de um elemento do *Staff*, o predicado a utilizar é apresentado de seguida:

```
nov Staff(Id,IdCs,N,E) :-  
    novoConhecimento(staff(Id,IdCs,N,E)).
```

Para o caso dos dados do staff também temos que garantir a validade dos dados na base de conhecimento por isso utilizamos invariantes para o fazer, que já foram abordados na ultima secção.

3.4.3 Centro de saúde

Para o registo de um novo centro de saúde, o predicado usado é o seguinte:

```
nov Centro(Id,IdCs,N,M,Tlf,E) :-  
    novoConhecimento(centro(Id,IdCs,N,M,Tlf,E)).
```

Tal como nos dois outros tipos de conhecimento, nos centros de saúde também temos que garantir a validade destes dados e utilizamos invariantes para o fazer, estes invariantes já foram descritos acima.

3.4.4 Vacinação

Relativamente ao registo de uma nova vacinação a um determinado utente, o predicado utilizado é:


```
novaVacinacao(IdS,IdU,D,V,T) :-
novoConhecimento(vacinacao(IdS,IdU,D,V,T)).
```

Neste tipo de dados, como nos outros, também temos que garantir que estes são válidos e para isso são usados invariantes, que já foram descritos acima.

3.5 Remover conhecimento

Remover conhecimento deve-se ao meta predicado *removeConhecimento* definido da seguinte forma:

```
removeConhecimento(T) :- solucoes(I, -T::I, L),
                        remocao(T),
                        testaPredicados(L).
```

```
remocao(Q) :- retract(Q).  
remocao(Q) :- assert(Q), !, fail.
```

3.5.1 Utente

Para remover um utente da base de conhecimento, é necessário invocar **removerUtente**.

```
removeUtente(Id) :- removeConhecimento(utente(Id,_,_,_,_,_,_,_,_,_)).
```

3.5.2 Staff

De modo a remover um elemento do *staff* da base de conhecimento, é necessário invocar **removerStaff**.

```
removeStaff(Id) :- removeConhecimento(staff(Id,_,_,_,_)).
```

3.5.3 Centro de Saúde

Com vista a remover um centro de saúde da base de conhecimento, é necessário invocar **removeCentro**.

```
removeCentro(Id) :- removeConhecimento(centro(Id,_,_,_,_,_)).
```

3.5.4 Vacinação

Por fim, para remover um elemento do *staff* da base de conhecimento, é necessário invocar **removerUtente**.

```
removeVacinacao(IdS,T) :- removeConhecimento(vacinacao(IdS,_,_,_,T)).
```

3.6 Identificações

De forma a cobrir as funcionalidades subjacentes deste caso prático, foram criado diversos predicados que têm como objetivo a identificação de utentes e de outras entidades com características específicas que possam ser relevantes para a zona de estudo em questão.

3.6.1 Identificar utentes não vacinados (por fases)

O grupo adaptou este predicado, que implementa a identificação de utentes candidatos a vacinação que ainda não foram vacinados, **relativamente a uma das três fases possíveis**, dando como argumento o número da fase da qual se pretende obter essa informação (1, 2, ou 3). Fizemos ainda a divisão em dois tipos de identificação distintos:

- Utesntes não vacinados **totalmente**, ou seja, que ainda não tomaram nenhuma dose da vacina;
- Utesntes não vacinados **parcialmente**, ou seja, que ainda só tomaram a primeira dose da vacina.

De destacar, como parte essencial destes predicados, os predicados **profissao1Fase**, **doencas1Fase** e **doencas2Fase**, que se encontram no ficheiro **auxiliares.pl** e que seleccionam os utentes com as determinadas características que os distinguirão no que toca à fase de vacinação atribuída, juntamente com os critérios que envolvem a idade, que é calculada com recurso ao predicado **calcularIdade**, que já foi apresentada na secção 3.2.1.

% Profissões de utentes correspondentes à fase 1

```
profissao1Fase(IdU) :- utente(IdU,_,_,_,_,_,_,_,P,_,_),
                        P = 'Profissional de Saude'.
```

% Doenças correspondentes à fase 1 (50 ou mais anos)

```
doencas1Fase(IdU) :- utente(IdU,_,_,_,_,_,_,_,D,_,_),
                        A = ['Insuficiencia renal','Doenca renal
                             ↳ cronica','Doenca coronaria','DPOC'],
                        temElementosComum(A,D).
```

% Doenças correspondentes à fase 2 (50 a 64 anos)

```
doencas2Fase(IdU) :- utente(IdU,_,_,_,_,_,_,_,D,_,_),
                        A = ['Diabetes','Neoplasia maligna ativa','Doenca
                             ↳ renal cronica','Insuficiencia
                             ↳ hepatica','Hipertensao arterial','Obesidade'],
                        temElementosComum(A,D).
```

Totalmente

Para o primeiro ponto apresentado, recorreremos a um predicado auxiliar **naoVacinados** que identifica utentes candidatos à vacinação que ainda não tomaram nenhuma dose da vacina.

```
naoVacinados(R) :- listaIdUtentes(X),
                    naoVacinadosAux(X,[],R).
```

```
naoVacinadosAux([],Acc,Acc).
```

```

naoVacinadosAux([H|T],Acc,R) :- (checkVacinado(H,Acc, 1);
    → checkVacinado(H,Acc, 2)), naoVacinadosAux(T,Acc,R).
naoVacinadosAux([H|T],Acc,R) :- naoVacinadosAux(T,[H|Acc],R).

```

Foi de seguida criado o predicado **candidatosVacinacaoT** que, tendo como base apenas esses utentes que ainda não tomaram vacinas, identifica os que correspondem a uma fase específica, baseando-se nos critérios definidos para cada uma.

No caso da primeira fase de vacinação, desenvolvemos da forma descrita no seguinte bloco de código, seleccionando utentes com mais de 80 anos, utentes que sejam Profissionais de Saúde ou utentes que tenham mais de 50 anos e alguma das doenças que têm acesso à primeira fase. No final são juntados todos os resultados e removidos possíveis pessoas que se encaixem em mais do que um dos critérios.

```

candidatosVacinacaoT(1,R) :-
    % Utentes que ainda não tomaram nenhuma dose da vacina
    naoVacinadosAux(S),
    % Utentes candidatos + de 80 anos
    findall(X,(member(X,S),calcularIdade(X,C), C>=80), I),
    % Utentes candidatos que são Profissionais de Saúde
    findall(X,(member(X,S),profissao1Fase(X)), Pr),
    % Utentes candidatos + de 50 anos e com doenças da fase 1
    findall(X,(member(X,S),doencas1Fase(X),calcularIdade(X,C),C>=50), Do),
    % Juntar as listas
    append(I,Pr,App), append(App,Do,App2),
    % Eliminar ids repetidos
    repetidos(App2,R).

```

Com a mesma linha de pensamento, excetuando os critérios corresponderem à segunda fase, o predicado seguinte selecciona utentes candidatos à segunda fase que ainda não tomaram nenhuma vacina. No final, é feito um **findall** para eliminar utentes que se encaixam neste critério mas que seriam da primeira fase e não da segunda, recorrendo ao predicado **member** e à negação do mesmo.

```

candidatosVacinacaoT(2,R) :-
    % Utentes que ainda não tomaram nenhuma vacina
    naoVacinadosAux(S),
    % Utentes candidatos com + de 65 anos
    findall(X,(member(X,S),calcularIdade(X,C),C>=65), I),
    % Utentes candidatos de 50 a 64 anos com doenças da fase 2
    findall(X,(member(X,S),doencas2Fase(X),calcularIdade(X,C),C>=50,C<=64),Do),
    % Juntar as listas e eliminar repetidos
    append(I,Do,App), repetidos(App,Rep),
    % Eliminar os que são da fase 1
    candidatosVacinacaoT(1,F1),
    findall(X,(member(X,Rep),\+member(X,F1)), R).

```

Finalmente, no caso de se pretender seleccionar os candidatos à terceira fase de vacinação que ainda não tomaram nenhuma vacina, o seguinte predicado exclui do resultado da **naoVacinadosAux** aqueles a que são atribuídas a primeira ou a segunda fase, retornando os restantes.

```

candidatosVacinacaoT(3,R) :-
    % Utentes que ainda não tomaram nenhuma vacina
    naoVacinadosAux(S),
    % Utentes candidatos não vacinados da F1 e F2
    candidatosVacinacaoT(1,F1),
    candidatosVacinacaoT(2,F2),
    append(F1,F2,App), repetidos(App,Rep),
    % Identificar os restantes
    findall(X,(member(X,S),\+member(X,Rep)), R).

```

Incluímos ainda um caso de exceção, para o caso de os argumentos do predicado não corresponderem a nenhuma fase válida.

```

candidatosVacinacaoT(A,R) :- \+ pertence(A,[1,2,3]) -> write('Fase invalida.
↳ As fases podem ser 1, 2 ou 3. '),
    fail.

```

Parcialmente

Exatamente a mesma lógica foi seguida para encontrar os utentes que apenas tomaram a primeira dose da vacina, ou seja, que ainda não estão totalmente vacinados, faltando a toma da segunda dose.

Porém, como queremos agora utilizar esse novo universo de pessoas, o predicado auxiliar passa a ser **vacinadosSoUmaDose**.

```
vacinadosSoUmaDose(R) :- % Identificar os IDs de utentes candidatos
    solucoes(IdU,utente(IdU,_,_,_,DNasc,_,_,_,P,D,_),V),
    % Identificar o ID dos que fizeram a toma 1
    solucoes(IdU, vacinacao(_,IdU,_,_,1), T1),
    % Identificar o ID dos que fizeram a toma 2
    solucoes(IdU, vacinacao(_,IdU,_,_,2), T2),
    % Identificar os que pertencem a uma e não à outra
    findall(X,(member(X,V),member(X,T1),\+member(X,T2)),R).
```

A título de exemplo, dado que o raciocínio se repete, apresenta-se de seguida o predicado que implementa a identificação dos utentes que ainda só tomaram uma dose e cuja fase correspondente é a primeira.

```
candidatosVacinacaoP(1,R) :- % Uteses que só tomaram a dose 1
    vacinadosSoUmaDose(S),
    % Uteses candidatos + de 80 anos
    findall(X,(member(X,S),calcularIdade(X,C), C>=80), I),
    % Uteses candidatos que são Profissionais de Saúde
    findall(X,(member(X,S),profissao1Fase(X)), Pr),
    % Uteses candidatos + de 50 anos e com doenças da fase 1
    findall(X,(member(X,S),doencas1Fase(X),calcularIdade(X,C),C>=50), Do),
    % Juntar as listas
    append(I,Pr,App), append(App,Do,App2),
    % Eliminar os que pertencem à fase 1
    repetidos(App2,R).
```

3.6.2 Identificar utentes vacinados

O grupo desenvolveu também predicados para identificar, de forma global (não referente a fases específicas), os utentes que já tomaram uma ou duas doses da vacina. Com este fim foram criados os predicados **vacinadosSoUmaDose** e **vacinadosDuasDoses**.

O **vacinadosSoUmaDose** identifica os utentes que tomaram a primeira dose e os utentes que tomaram a segunda dose e lista os que pertencem ao primeiro caso e não ao segundo.

Já o **vacinadosDuasDoses** identifica todos os que tomaram tanto a primeira como a segunda dose, intersectando as duas listas.

```
vacinadosSoUmaDose(R) :- % Encontrar os IDs de utentes candidatos
    solucoes(IdU,utente(IdU,_,_,_,DNasc,_,_,_,P,D,_),V),
    % Encontrar o ID dos que fizeram a toma 1
    solucoes(IdU, vacinacao(_,IdU,_,_,1), T1),
    % Encontrar o ID dos que fizeram a toma 2
    solucoes(IdU, vacinacao(_,IdU,_,_,2), T2),
    % Ver os que pertencem a uma e nao ao outro
    findall(X,(member(X,V),member(X,T1),\+member(X,T2)),R).

vacinadosDuasDoses(R) :- solucoes(UId,vacinacao(A,UId,B,C,1),L1),
    solucoes(UId,vacinacao(A,UId,B,C,2),L2),
    intersecao(L1,L2,R).
```

3.6.3 Vacinação Indevida

Baseando-nos nas decisões feitas e apresentadas na secção 3.1.1, o grupo criou predicados que implementam funcionalidades de identificação de utentes vacinados de forma indevida.

Fase não correspondente

Para identificar utentes que tomaram a vacina numa fase que não lhes era destinada, foi criado o predicado **vacinacaoFaseErrada**, com recurso ao auxiliar **vacinacaoFaseErradaAux**, que por sua vez tem como auxiliar o **vacinacaoFaseErradaUtente**.

O predicado **vacinacaoFaseErradaUtente** recorre às funções **checkFaseUtente** e **pertenceAFase**, verificando a que fase o utente era suposto ser vacinado e comparando com a fase em que foi.

% Verifica em que fase é que um utente foi vacinado

```
checkFaseUtente(UId,R):- vaccinacao(_,UId,D,_,1), checkFase(D,R).
```

% Verifica a que fase pertence um certo dia

```
checkFase(D,R):- fase1DataI(DI1), comparaDatasStr(D,DI1,A1),  
                 fase1DataF(DF1), comparaDatasStr(D,DF1,A2),  
                 fase2DataI(DI2), comparaDatasStr(D,DI2,A3),  
                 fase2DataF(DF2), comparaDatasStr(D,DF2,A4),  
                 fase3DataI(DI3), comparaDatasStr(D,DI3,A5),  
                 fase3DataF(DF3), comparaDatasStr(D,DF3,A6),  
                 ((A1==0,A2==1) -> R is 1 ;  
                 (A3==0,A4==1) -> R is 2 ;  
                 (A5==0,A6==1) -> R is 3 ;  
                 R is 0).
```

```
pertenceAFase(IDU,R) :- profissao(IDU,Prof),  
                        calcularIdade(IDU,Idade),  
                        (((doencas1Fase(IDU), Idade >= 50); Idade >= 80; Prof  
                        == 'Profissional de Saude') -> R is 1;  
                        ((doencas2Fase(IDU), Idade >= 50, Idade <= 64); Idade  
                        >= 65) -> R is 2;  
                        R is 3).
```


Deste modo, o seu resultado difere caso o utente tenha sido vacinado na fase que lhe pertence, ou não.

Este predicado é utilizado na **vacinacaoFaseErradaAux**, que cria uma lista com os elementos de uma outra lista que encaixem nos critérios de fase de vacinação indevida. A **vacinacaoFaseErrada** aplica esse mesmo predicado à lista de utentes (na sua generalidade).

```
vacinacaoFaseErrada(R) :- listaIdUtentes(X),
                           vaccinacaoFaseErradaAux(X, [], R).

vacinacaoFaseErradaAux([], Acc, Acc).
vacinacaoFaseErradaAux([H|T], Acc, R) :-
    → (vacinacaoFaseErradaUtente(H, X), X = 1),
    → vaccinacaoFaseErradaAux(T, Acc, R).
vacinacaoFaseErradaAux([H|T], Acc, R) :-
    → vaccinacaoFaseErradaAux(T, [H|Acc], R).

vacinacaoFaseErradaUtente(UId, X) :- (checkFaseUtente(UId, R1),
                                       pertenceAFase(UId, R2)),
                                       (R1 \= R2 -> X = 0; X = 1).
```

Toma de duas vacinas diferentes

Com a funcionalidade de identificar utentes que tomaram doses de diferentes tipos de vacinas foram criados os predicados **vacinaDiferente** (que identifica vacinações indevidas neste sentido) e **diferente** (que lista os utentes em que esta situação se verifica).

```
diferente(R) :-
    solucoes(U_ID, vacinaDiferente(U_ID), R).

vacinaDiferente(U_ID) :-
    vaccinacao(_, U_ID, _, V1, 1),
```

```
vacinacao(_,U_ID,_,V2,2),  
V1 \= V2.
```

Toma de mais do que duas vacinas

Para determinar a identificação de casos de toma de vacina indevidos, são detetados utentes que por alguma razão registam a toma de mais do que duas doses da vacina com o predicado **tomouMaisDe2DosesAux** e listados com o **tomouMaisDe2Doses**.

```
tomouMaisDe2Doses(R) :- listaIdUtentes(X),  
                           tomouMaisDe2DosesAux(X, [], R).  
  
tomouMaisDe2DosesAux([], Acc, Acc).  
tomouMaisDe2DosesAux([H|T], Acc, R) :- (quantasVacinas(H, N), N<3),  
    ↪ tomouMaisDe2DosesAux(T, Acc, R).  
tomouMaisDe2DosesAux([H|T], Acc, R) :- tomouMaisDe2DosesAux(T, [H|Acc], R).
```

CONCLUSÕES E SUGESTÕES

No que toca à parte pedagógica, concluímos que esta fase do projeto foi muito enriquecedora para o grupo, pois permitiu que melhorássemos as nossas competências a nível prático, relacionadas com a Unidade Curricular de Sistemas de Representação de Conhecimento e Raciocínio.

O grupo considera bastante produtiva a exploração das ferramentas da disciplina, tais como a linguagem PROLOG e o programa *SWI-Prolog*, usadas para construir bases de conhecimentos dinâmicas e sistemas de inferência complexos, de modo a tratar conhecimento proveniente de situações reais.

O sistema de representação de conhecimento criado permite caracterizar de forma eficiente e completa o tema da vacinação global da população portuguesa, o que vai ao encontro dos objetivos traçados inicialmente e, por isso, o grupo está satisfeito com o resultado final, tendo, também, sido cumpridos todos os requisitos do enunciado.

Em suma, o esforço foi grande com o intuito de garantir boas soluções para o enunciado proposto deixando, assim, uma janela aberta e curiosa para a próxima fase.

BIBLIOGRAFIA

- [1] Cesar Analide, Paulo Novais, José Neves. *Sugestões para a Redacção de Relatórios Técnicos*. 2011.
- [2] PROLOG,
<https://pt.wikibooks.org/wiki/Prolog>
- [3] FASES DE VACINAÇÃO,
<https://www.sns24.gov.pt/guia/programa-nacional-vacinacao>
- [4] SWI-Prolog,
<https://www.swi-prolog.org>



CÓDIGO FONTE

A.1 auxiliares.pl

```
:- use_module(library(lists)).
:- use_module(library(date)).

% -----
% FUNCOES AUXILIARES
% -----
% Apresenta todas as solucoes
solucoes(T,Q,S) :- findall(T,Q,S).

% Inserção de conhecimento
insercao(Q) :- assert(Q).
insercao(Q) :- retract(Q), !, fail.

% Remoção de conhecimento
remocao(Q) :- retract(Q).
remocao(Q) :- assert(Q), !, fail.

% Testa se todos os predicados são verdadeiros
testaPredicados([]).
testaPredicados([I|L]) :- I, testaPredicados(L).

% Insere novo conhecimento na base de conhecimento
novoConhecimento(T) :- solucoes(I, +T::I, L),
                        insercao(T),
                        testaPredicados(L).
```

```

% Retira conhecimento da base de conhecimento
removeConhecimento(T) :- solucoes(I, -T::I, L),
    remocao(T),
    testaPredicados(L).

% Vê se um genero é válido
genderValido('M').
genderValido('F').

% Elimina elementos repetidos de uma lista
repetidos([], []).
repetidos([X|L], [X|NL]) :- removerElem(L, X, TL), repetidos(TL, NL).

removerElem([], _, []).
removerElem([X|L], X, NL) :- removerElem(L, X, NL).
removerElem([X|L], Y, [X|NL]) :- X \== Y, removerElem(L, Y, NL).

% Verifica se um elemento pertence a uma lista
pertence(A, [A|_]).
pertence(A, [_|XS]) :- pertence(A, XS).

% Verifica se duas listas têm elementos em comum
temElementosComum([H|_], L2) :- pertence(H, L2).
temElementosComum([_|T], L2) :- temElementosComum(T, L2).

% Calcular a idade
calcularIdade(IdU, R) :- utente(IdU, _, _, DNasc, _, _, _, _),
    split_string(DNasc, "-", "", Sp),
    nth0(0, Sp, Ano), nth0(1, Sp, Mes), nth0(2, Sp, Dia),
    get_date_time_value(year, AnoN), get_date_time_value(month, MesN),
    → get_date_time_value(day, DiaN),
    atom_number(Ano, Year), atom_number(Mes, Month), atom_number(Dia, Day),
    atom_string(AnoN, YearN), atom_number(YearN, YearNN),
    atom_string(MesN, MonthN), atom_number(MonthN, MonthNN),
    atom_string(DiaN, DayN), atom_number(DayN, DayNN),
    Age is (YearNN - Year),
    (Month > MonthNN -> AgeN is Age-1 ;
    Month < MonthNN -> AgeN is Age ;
    (Day > DayNN -> AgeN is Age-1 ;
    AgeN is Age)),
    R = AgeN.

% Obter a data atual
get_date_time_value(Key, Value) :-
    get_time(Stamp),
    stamp_date_time(Stamp, DateTime, local),
    date_time_value(Key, DateTime, Value).

```



```

fase1DataI(DI) :- fase('F1',DI,_).
fase1DataF(DF) :- fase('F1',_,DF).
fase2DataI(DI) :- fase('F2',DI,_).
fase2DataF(DF) :- fase('F2',_,DF).
fase3DataI(DI) :- fase('F3',DI,_).
fase3DataF(DF) :- fase('F3',_,DF).

% Verifica a que fase pertence um certo dia
checkFase(D,R):- fase1DataI(DI1), comparaDatasStr(D,DI1,A1),
                 fase1DataF(DF1), comparaDatasStr(D,DF1,A2),
                 fase2DataI(DI2), comparaDatasStr(D,DI2,A3),
                 fase2DataF(DF2), comparaDatasStr(D,DF2,A4),
                 fase3DataI(DI3), comparaDatasStr(D,DI3,A5),
                 fase3DataF(DF3), comparaDatasStr(D,DF3,A6),
                 ((A1==0,A2==1) -> R is 1 ;
                 (A3==0,A4==1) -> R is 2 ;
                 (A5==0,A6==1) -> R is 3 ;
                 R is 0).

%Verifica em fase é que um Utente foi vacinado
checkFaseUtente(UId,R):- vaccinacao(_,UId,D,_,1), checkFase(D,R).

%valida o género
generoValido('M').
generoValido('F').

```

A.2 dados.pl

```

utente('U0','Romulo Mota','22337840788','M','1936-04-22','RomuloMota@outlook.com',962935227,'Rua
→ Francisco Augusto Alvim 4700-004 Braga','Biologo',[],'C1').
utente('U1','Goncalo
→ Melo','65105592957','M','1931-09-18','GoncaloMelo1991@hotmail.com',938823950,'Travessa 1 de Maio
→ 4700-008 Braga','Geriatra',[],'C2').
utente('U2','Bianca Moura','84874182284','F','1958-10-30','BiancaMoura244@gmail.com',921591637,'Rua de
→ Sao Rosendo ( Bispo de Dume ) Dume 4700-008 Braga','Personal trainer',[],'C3').
utente('U3','Vilar Nunes','05052988794','F','1976-02-17','VilarNunes1976@gmail.com',968452327,'Rua de
→ Remelhe Dume 4700-008 Braga','Vigilante',[],'C3').
utente('U4','Cinderela
→ Nogueira','50775228700','F','1950-12-08','CinderelaNogueira@outlook.com',924781129,'Rua Paulo Vi
→ 4700-004 Braga','Auxiliar de limpeza',[],'C2').
utente('U5','Apolo Amaral','39446661300','M','1980-05-06','ApoloAmaral@hotmail.com',926158761,'Avenida
→ dos Descobrimentos Antas 4760-001 Vila Nova de Famalicao','Reporter',['Insuficiencia
→ hepatica'],'C3').

```


utente('U6','Zelia Abreu','94334540382','F','1995-06-19','ZeliaAbreu@hotmail.com',966727220,'Rua do
→ Brasil 4775-001 Cambeses','Eletricista',[],'C2').

utente('U7','Afonsino Torres','82144830813','M','1992-12-24','AfonsinoTorres@gmail.com',911799897,'Rua
→ Jose Albino Costa e Silva Azurem 4800-004 Guimaraes','Profissional de Saude',[],'C2').

utente('U8','Aldo Castro','89885876143','M','1980-04-16','AldoCastro388@hotmail.com',962967073,'Rua
→ Doutor Jose Regio Dume 4700-004 Braga','Programador',[],'C2').

utente('U9','Tiburcio
→ Guerreiro','08519097436','M','1978-03-16','TiburcioGuerreiro342@outlook.com',936810784,'Rua do
→ Cortinhal 4775-001 Cambeses','Militar',['Obesidade'],'C2').

utente('U10','Jansenio
→ Figueiredo','02069412938','M','1983-02-02','JansenioFigueiredo218@outlook.com',914112112,'Avenida
→ do Cavado Dume 4700-008 Braga','Comissario de bordo',[],'C2').

utente('U11','Florisbela
→ Carvalho','38557360298','F','1969-02-01','FlorisbelaCarvalho1969@gmail.com',910843987,'Rua Dona
→ Teresa Aldao 4800-004 Guimaraes','Tecnico em edificacoes',[],'C4').

utente('U12','Biana Andrade','03294496759','F','1982-05-02','BianaAndrade@hotmail.com',932477556,'Rua
→ de Sao Martinho Dume 4700-008 Braga','Padeiro',[],'C1').

utente('U13','Adelia Amaral','41527730111','F','1944-09-26','AdeliaAmaral@hotmail.com',910419566,'Rua
→ Doutor Joao Afonso Almeida Azurem 4800-004 Guimaraes','Bioquimico',[],'C2').

utente('U14','Tatiano
→ Carvalho','39247286162','M','1960-12-30','TatianoCarvalho@outlook.com',921622230,'Praceta do
→ Viajante Azurem 4800-004 Guimaraes','Biologo',[],'C3').

utente('U15','Indro Leal','45346361911','M','1949-07-08','IndroLeal1949@gmail.com',929206797,'Rua de
→ Sao Lourenco da Ordem 4700-004 Braga','Webmaster',[],'C1').

utente('U16','Gino Campos','21515788464','M','2004-03-23','GinoCampos@gmail.com',920820869,'Lugar da
→ Ordem 4700-008 Braga','Piloto de aviao',[],'C4').

utente('U17','Dinarda
→ Matias','29224473940','F','1974-10-09','DinardaMatias1974@outlook.com',915914158,'Largo do Rego
→ Dume 4700-008 Braga','DevOps',[],'C4').

utente('U18','Marcilene
→ Goncalves','77919813012','M','1989-12-04','MarcileneGoncalves305@hotmail.com',924527670,'Rua do
→ Carvalhal Dume 4700-005 Braga','Perito criminal',['Doenca coronaria'],'C1').

utente('U19','Milena
→ Vicente','39280061650','F','2004-06-04','MilenaVicente@gmail.com',969590100,'Travessa General
→ Humberto Delgado Antas 4760-001 Vila Nova de Famalicao','Jovem Aprendiz',[],'C3').

utente('U20','Delmiro
→ Lopes','36606379205','M','1956-05-23','DelmiroLopes103@outlook.com',927819307,'Rua Monte de Baixo
→ 4705-001 Arentim','Cerimonialista',[],'C3').

utente('U21','Rodrigo Matos','90131842525','M','1996-06-04','RodrigoMatos@gmail.com',930344386,'Largo
→ da Confeiteira Dume 4700-005 Braga','Cientista de dados',[],'C3').

utente('U22','Heliodora
→ Marques','08984526417','F','1963-02-10','HeliodoraMarques393@gmail.com',931974714,'Calcada de
→ Gondomar 4705-001 Arentim','Radialista',[],'C2').

utente('U23','Alexia Oliveira','71895649370','F','1978-05-23','AlexiaOliveira@gmail.com',919122593,'Rua
→ da Igreja Velha 4705-001 Arentim','Embaixador',[],'C1').

utente('U24','Milena Lima','44309762734','F','2006-12-31','MilenaLima145@hotmail.com',911200424,'Rua de
→ Cabanas Dume 4700-004 Braga','Operador de telemarketing',[],'C1').

utente('U25','Taisa Tavares','48015001236','F','2007-01-26','TaisaTavares@gmail.com',923954996,'Rua da
→ Gloria Azurem 4800-004 Guimaraes','Mecanico de carro',[],'C3').

utente('U26','Gildasio
→ Martins','96690354667','M','1998-04-29','GildasioMartins@hotmail.com',911314055,'Rua Jose Antonio
→ Gomes Santos 4775-001 Cambeses','Agronomo',[],'C4').

utente('U27','Titania Pinho','47582688120','F','1968-08-03','TitaniaPinho208@gmail.com',934177609,'Rua
→ de Sao Rosendo (Bispo de Dume) Dume 4700-008 Braga','Tecnico em Seguranca do Trabalho',[],'C2').

utente('U28','Cilia Coelho','64133680592','F','1984-12-14','CiliaCoelho1984@outlook.com',932543818,'Rua
→ Jose Albino Costa e Silva Azurem 4800-004 Guimaraes','Geriatra',[],'C2').

utente('U29','Inocencia
→ Sousa','70241114578','F','1946-01-20','InocenciaSousa1946@hotmail.com',931864703,'Calçada de
→ Gondomar 4705-001 Arentim','Cenografo',[],'C2').

utente('U30','Fania Antunes','47451047894','F','1950-03-06','FaniaAntunes@gmail.com',960581729,'Rua da
→ Liberdade 4705-001 Arentim','Promotor de justica',[],'C2').

utente('U31','Martinho
→ Baptista','30102331640','M','1951-11-10','MartinhoBaptista1951@outlook.com',960643533,'Travessa
→ Cimo de Vila Dume 4700-005 Braga','Massoterapeuta',[],'C4').

utente('U32','Ederia Sa','67479105813','F','1973-08-23','EderiaSa@outlook.com',963318093,'Rua Doutor
→ Manuel Jose de Oliveira Machado Dume 4700-005 Braga','Embaixador',[],'C4').

utente('U33','Tiberio Gomes','43790724848','M','1941-07-13','TiberioGomes@gmail.com',915913364,'Rua da
→ Carcamije Dume 4700-004 Braga','Web designer',[],'C3').

utente('U34','Bartolomeu
→ Moreira','29902966799','M','1949-01-11','BartolomeuMoreira304@gmail.com',933737673,'Rua Bernardim
→ Ribeiro Dume 4700-004 Braga','Agricultor',[],'C4').

utente('U35','Roberta Jesus','95330506511','F','1976-09-08','RobertaJesus@hotmail.com',963001104,'Rua
→ da Veiga de Fora Fermentoes 4800-004 Guimaraes','Radialista',[],'C3').

utente('U36','Abraao Maia','01772262173','M','2009-03-18','AbraaoMaia2009@gmail.com',923291250,'Caminho
→ de Fontelas 4775-001 Cambeses','Auxiliar de cozinha',[],'C2').

utente('U37','Altina Freitas','93533906519','F','1941-08-04','AltinaFreitas@hotmail.com',937004801,'Rua
→ Ferreira de Castro Azurem 4800-004 Guimaraes','Encanador',['Obesidade'],'C4').

utente('U38','Delia
→ Nascimento','97154142206','F','1990-12-22','DeliaNascimento81@hotmail.com',966453438,'Avenida
→ Rebelo Mesquita Calendario 4760-001 Vila Nova de Famalicao','Auxiliar de expedicao',[],'C3').

utente('U39','Agnelo Leite','13667173790','M','1986-01-12','AgneloLeite1986@gmail.com',937121916,'Rua
→ Comendador Padre David de Oliveira Martins 4705-001 Arentim','Delegado',[],'C2').

utente('U40','Alivar
→ Pinho','06353715131','F','1948-07-09','AlivarPinho@hotmail.com',939959983,'Travessa da Carreira
→ 4775-001 Cambeses','Auditor fiscal',[],'C2').

utente('U41','Placido
→ Henriques','62071335155','M','1991-03-14','PlacidoHenriques1991@gmail.com',926159990,'Rua Jose
→ Antonio Gomes Santos 4775-001 Cambeses','Designer',[],'C2').

utente('U42','Bricia
→ Nascimento','67676986761','F','2011-08-11','BriciaNascimento@hotmail.com',922598775,'Calçada de
→ Gondomar 4705-001 Arentim','Encanador',['Obesidade'],'C1').

utente('U43','Leocadia
→ Freitas','51604631364','F','1954-10-08','LeocadiaFreitas@hotmail.com',929465172,'Cangosta do Ruivo
→ Dume 4700-005 Braga','Auditor fiscal',[],'C4').

```

utente('U44','Cesaltina
→ Gaspar','74221775731','F','1973-10-20','CesaltinaGaspar@outlook.com',918321370,'Rua do Brasil
→ 4775-001 Cambeses','Advogado',[],'C3').

utente('U45','Delmiro
→ Jesus','80502189538','M','2011-07-08','DelmiroJesus2011@outlook.com',931578562,'Rua Conego Insuelas
→ Dume 4700-005 Braga','Desenvolvedor',['Doenca coronaria'],'C1').

utente('U46','Biana Moreira','48946886509','F','1990-09-20','BianaMoreira@outlook.com',961221225,'Rua
→ do Brasil 4775-001 Cambeses','Bancario',[],'C1').

utente('U47','Eliana
→ Antunes','94319794560','F','1951-01-14','ElianaAntunes130@hotmail.com',923081249,'Rua de Sao
→ Lourenco da Ordem 4700-004 Braga','Mestre cervejeiro',[],'C4').

utente('U48','Fiona
→ Teixeira','72522541652','F','1999-11-11','FionaTeixeira@hotmail.com',923206228,'Travessa 1 de Maio
→ Antas 4760-001 Vila Nova de Famalicao','Trader',[],'C4').

utente('U49','Eda
→ Fernandes','47005694161','F','1946-09-22','EdaFernandes54@gmail.com',911220373,'Avenida Rebelo
→ Mesquita Antas 4760-001 Vila Nova de Famalicao','Editor de texto',[],'C2').

utente('U50','Tulipa
→ Soares','76322680317','F','1995-06-03','TulipaSoares368@hotmail.com',926265902,'Rua da Liberdade
→ 4705-001 Arentim','Piloto de aviao',[],'C3').

utente('U51','Anabela Vaz','56064967830','F','2011-05-25','AnabelaVaz2011@hotmail.com',933462896,'Rua
→ Antonio Aleixo Dume 4700-004 Braga','Radialista',[],'C1').

utente('U52','Analdina
→ Andrade','36861200568','F','1962-06-28','AnaldinaAndrade1962@gmail.com',920087770,'Rua do Cordeiro
→ Dume 4700-005 Braga','Oficial de Justica',[],'C2').

utente('U53','Arquimínio
→ Amorim','09865962646','M','1944-05-14','ArquimínioAmorim@outlook.com',916808532,'Avenida do Brasil
→ Gaviao 4760-001 Vila Nova de Famalicao','Engenheiro civil',[],'C3').

utente('U54','Lineia Andrade','78235131463','F','1969-11-29','LineiaAndrade@hotmail.com',918868429,'Rua
→ de Entre-Portas Dume 4700-005 Braga','Tecnico em Seguranca do Trabalho',[],'C2').

utente('U55','Melania
→ Andrade','46092080561','F','1952-04-14','MelaniaAndrade57@gmail.com',925763198,'Rua de Pecelar
→ 4705-001 Arentim','Produtor de video',[],'C3').

utente('U56','Amalia
→ Ribeiro','76328401404','F','1975-05-06','AmaliaRibeiro@gmail.com',918939924,'Praceta Padre Luis
→ Gonzaga Azurem 4800-004 Guimaraes','Profissional de Relacoes Publicas',[],'C1').

utente('U57','Fausto
→ Assuncao','12752615643','M','1962-10-07','FaustoAssuncao1962@outlook.com',914431564,'Caminho de
→ Fontelas 4775-001 Cambeses','Professor',[],'C3').

utente('U58','Frederico
→ Baptista','87205579488','M','1949-03-24','FredericoBaptista@gmail.com',918218889,'Rua Fernando
→ Pessoa Dume 4700-004 Braga','Advogado',['Doenca renal cronica'],'C4').

utente('U59','Gavio
→ Cruz','59914154683','M','1973-01-06','GavioCruz1973@hotmail.com',934488621,'Travessa do Rego
→ 4700-008 Braga','Turismologo',[],'C4').

utente('U60','Afonso
→ Ramos','17806159011','M','1973-04-17','AfonsoRamos1973@gmail.com',919273556,'Rua de Francos
→ Azurem 4800-004 Guimaraes','Auxiliar de cozinha',[],'C2').

```

```

utente('U61','Nidia
→ Assuncao','55485443638','F','1946-06-12','NidiaAssuncao280@outlook.com',922990127,'Rua do Souto
→ 4775-001 Cambeses','Estoquista',[],'C4').
utente('U62','Altina Paiva','10576158284','F','1979-09-30','AltinaPaiva349@hotmail.com',922806446,'Rua
→ Comendador Padre David de Oliveira Martins 4705-001 Arentim','Estoquista',[],'C3').
utente('U63','Delcio
→ Tavares','50921922602','M','1959-12-26','DelcioTavares@hotmail.com',968252844,'Largo de Sao Paulo
→ 4700-004 Braga','Operador de telemarketing',[],'C1').
utente('U64','Katarina
→ Domingues','66831923508','F','1940-11-12','KatarinaDomingues364@hotmail.com',916774272,'Rua do Real
→ 4705-001 Arentim','Delegado',[],'C1').
utente('U65','Ziza Vieira','40247354095','F','1961-09-02','ZizaVieira1961@outlook.com',926272084,'Rua
→ do Rego Dume 4700-008 Braga','Assistente social',[],'C3').
utente('U66','Domitilia
→ Maia','13376306360','F','1994-07-05','DomitiliaMaia@outlook.com',924422638,'Travessa do Cruzeiro
→ Dume 4700-005 Braga','Assistente social',[],'C1').
utente('U67','Isidoro
→ Pacheco','63787512555','M','1979-10-31','IsidoroPacheco@hotmail.com',968049748,'Rua do Real
→ 4705-001 Arentim','Coach',[],'C1').
utente('U68','Turgo Brito','99513767922','M','1969-03-16','TurgoBrito@outlook.com',933455173,'Rua
→ Ferreira de Castro Azurem 4800-004 Guimaraes','Profissional de Saude',['Insuficiencia
→ cardiaca'],'C4').
utente('U69','Helio Coelho','85336251307','M','1977-11-14','HelioCoelho@outlook.com',925146413,'Largo
→ de Sao Paulo 4700-004 Braga','Auxiliar de limpeza',[],'C4').
utente('U70','Reginaldo
→ Lima','27916959506','M','2000-05-25','ReginaldoLima2000@hotmail.com',925397987,'Rua Francisco
→ Teixeira Azurem 4800-004 Guimaraes','Motorista',[],'C1').
utente('U71','Cristovao
→ Torres','71980305446','M','2007-06-29','CristovaoTorres@outlook.com',929194515,'Praceta do Viajante
→ Azurem 4800-004 Guimaraes','Assistente financeiro',[],'C1').
utente('U72','Berto
→ Fernandes','54572112374','M','1976-10-30','BertoFernandes@hotmail.com',920192141,'Rua Bernardim
→ Ribeiro Dume 4700-004 Braga','Engenheiro de alimentos',[],'C2').
utente('U73','Mileide
→ Moreira','89739835122','F','2008-07-13','MileideMoreira@gmail.com',917821843,'Caminho do Carvalhedo
→ 4700-005 Braga','Confeiteiro',[],'C1').
utente('U74','Aquira
→ Matos','43401087910','F','1997-06-23','AquiraMatos1997@outlook.com',924857933,'Lugar do Pinhel
→ 4700-005 Braga','Desenhista',[],'C3').
utente('U75','Pandora
→ Moreira','91397307632','F','1970-03-30','PandoraMoreira@hotmail.com',915740811,'Rua do Senhor da
→ Cruz 4775-001 Cambeses','Promotor de justica',[],'C3').
utente('U76','Fernando
→ Lourenco','68859363325','M','1943-02-14','FernandoLourenco1943@outlook.com',925089624,'Rua de Sao
→ Rosendo ( Bispo de Dume ) Dume 4700-008 Braga','Promotor de vendas',[],'C2').
utente('U77','Simao Nunes','42103103421','M','1990-12-19','SimaoNunes77@outlook.com',964476432,'Rua do
→ Senhor da Cruz 4775-001 Cambeses','Publicitario',[],'C4').

```

```

utente('U78','Ursicino
→ Gonçalves','38952401676','M','1983-10-29','UrsicinoGoncalves1983@outlook.com',965471826,'Rua de Sao
→ Rosendo ( Bispo de Dume ) Dume 4700-008 Braga','Esteticista',[],'C4').
utente('U79','Adamantino
→ Pinheiro','99783015761','M','1969-05-18','AdamantinoPinheiro@outlook.com',963879712,'Rua Monte de
→ Baixo 4705-001 Arentim','Engenheiro de alimentos',[],'C3').
utente('U80','Enio
→ Loureiro','05563703555','M','2001-08-19','EnioLoureiro185@outlook.com',918068382,'Rua Carlos Paiao
→ 4700-004 Braga','Mestre cervejeiro',['Doenca coronaria'],'C2').
utente('U81','Arao Paiva','73831300882','M','1955-05-06','AraoPaiva@gmail.com',966896680,'Rotunda
→ Bernardino Machado Antas 4760-001 Vila Nova de Famalicao','Taquigrafo',[],'C4').
utente('U82','Demetrio
→ Carvalho','37263260880','M','1941-12-23','DemetrioCarvalho@outlook.com',935964013,'Rua Fernando
→ Pessoa Dume 4700-004 Braga','Biologo',[],'C4').
utente('U83','Elza Miranda','61337738817','F','1939-03-30','ElzaMiranda341@outlook.com',919290967,'Rua
→ Antonio Dias de Azevedo Antas 4760-001 Vila Nova de Famalicao','Advogado criminalista',['Doenca
→ renal cronica'],'C4').
utente('U84','Alcindo
→ Santos','66999734239','M','1985-07-10','AlcindoSantos93@hotmail.com',963620531,'Calcada de Sao
→ Martinho Real 4700-004 Braga','Cientista de dados',['Insuficiencia renal'],'C2').
utente('U85','Maximiliano
→ Cardoso','34704659251','M','1976-04-11','MaximilianoCardoso@hotmail.com',933602880,'Largo de
→ Carcavelos Dume 4700-005 Braga','Veterinario',[],'C3').
utente('U86','Acursio
→ Gonçalves','32596244006','M','1970-04-15','AcursioGoncalves1970@hotmail.com',926999449,'Travessa
→ Veiga de Fora Fermentoes 4800-004 Guimaraes','Assistente administrativo',[],'C3').
utente('U87','Fabiao
→ Ribeiro','16596319906','M','1950-07-11','FabiaoRibeiro1950@outlook.com',935689581,'Travessa Cimo de
→ Vila Dume 4700-005 Braga','Engenheiro de alimentos',[],'C1').
utente('U88','Gislana
→ Fernandes','50010259292','F','1979-09-30','GislanaFernandes73@hotmail.com',928960733,'Rua Antonio
→ Alves Rei Dume 4700-004 Braga','Bartender',[],'C4').
utente('U89','Isa Loureiro','84417052024','F','1987-03-23','IsaLoureiro6@gmail.com',938760652,'Estrada
→ Nacional 101 Dume 4700-005 Braga','Bioquimico',['Obesidade'],'C1').
utente('U90','Ferdinando
→ Freitas','05466312375','M','1953-10-08','FerdinandoFreitas@gmail.com',925101473,'Calcada de
→ Gondomar 4705-001 Arentim','Agronomo',[],'C3').
utente('U91','Emaus Borges','33469863754','F','2000-05-11','EmausBorges@gmail.com',910199316,'Calcada
→ de Sao Martinho Real 4700-004 Braga','Farmaceutico',[],'C4').
utente('U92','Lutero
→ Amorim','83025402612','M','1974-04-03','LuteroAmorim152@hotmail.com',910704567,'Largo da
→ Confeiteira Dume 4700-005 Braga','Engenheiro civil',['Diabetes'],'C1').
utente('U93','Amarildo
→ Vicente','91868122305','M','1948-11-28','AmarildoVicente122@hotmail.com',939782863,'Rua de Sao
→ Paulo 4700-004 Braga','Tecnico em edificacoes',[],'C3').
utente('U94','Isolina Brito','30911560716','F','2006-02-15','IsolinaBrito71@hotmail.com',915491202,'Rua
→ Jose da Carolina 4705-001 Arentim','Assistente de marketing',[],'C1').

```

```

utente('U95','Fabricia Abreu','06670427362','F','1954-10-05','FabriciaAbreu@gmail.com',914571513,'Rua
→ de Remelhe Dume 4700-008 Braga','Engenheiro de alimentos',['Hipertensao arterial'],'C3').
utente('U96','Delmiro Castro','14908941761','M','1969-03-05','DelmiroCastro@hotmail.com',926706821,'Rua
→ do Regedor 4775-001 Cambeses','Militar',[],'C2').
utente('U97','Carmerio
→ Araujo','65989136423','M','1980-11-03','CarmerioAraujo@outlook.com',966564116,'Largo de Sao
→ Sebastiao 4700-008 Braga','Assistente pessoal',[],'C3').
utente('U98','Horacio
→ Pereira','13101300241','M','1970-02-11','HoracioPereira89@gmail.com',928548906,'Praca da Autarquia
→ 4775-001 Cambeses','Engenheiro de materiais',[],'C1').
utente('U99','Debora
→ Mendes','39536881148','F','1963-11-06','DeboraMendes250@gmail.com',968445773,'Caminho do Carvalho
→ 4700-005 Braga','Profissional de Facilities',[],'C4').
utente('U100','Antonio
→ Santos','39536881144','M','1999-07-22','a83700@alunos.uminho.com',968445772,'Caminho do Carvalho
→ 4700-005 Braga','Vagabundo',[],'C4').
utente('U101','Jorge
→ Vieira','09536881144','M','1999-06-23','a84240@alunos.uminho.com',928445772,'Caminho do Carvalho
→ 4700-005 Braga','Engenheiro Informatico',[],'C3').
utente('U102','Joana
→ Gomes','39533881144','F','1999-01-01','a84912@alunos.uminho.com',922445772,'Caminho do Carvalho
→ 4700-005 Braga','Engenheiro Informatico',[],'C2').
utente('U103','Ze Carlos
→ Magalhaes','19536881144','M','1999-01-01','a85852@alunos.uminho.com',922645772,'Caminho do
→ Carvalho 4700-005 Braga','Engenheiro Informatico',[],'C4').
utente('U104','Paulo
→ Pereira','59236281144','M','1999-01-01','a86475@alunos.uminho.com',922445222,'Caminho do Carvalho
→ 4700-005 Braga','Professor de CP',[],'C2').

staff('S0','Leonardina Anjos','12287559265','LeonardinaAnjos49@outlook.com','C2').
staff('S1','Fernando Guerreiro','28567061757','FernandoGuerreiro@outlook.com','C2').
staff('S2','Idelia Paiva','61816783608','IdeliaPaiva@hotmail.com','C1').
staff('S3','Bianca Baptista','55616211371','BiancaBaptista@hotmail.com','C3').
staff('S4','Conceicao Lopes','21645651780','ConceicaoLopes2@outlook.com','C2').
staff('S5','Isandro Azevedo','41601362411','IsandroAzevedo107@gmail.com','C3').
staff('S6','Almiro Soares','65199252818','AlmiroSoares1991@gmail.com','C2').
staff('S7','Juliao Abreu','54101439180','JuliaoAbreu@gmail.com','C4').
staff('S8','Dominico Goncalves','65543192708','DominicoGoncalves@hotmail.com','C3').
staff('S9','Tatiana Faria','59173595792','TatianaFaria1958@outlook.com','C4').
staff('S10','Juliano Campos','62975266565','JulianoCampos1991@hotmail.com','C2').
staff('S11','Vitalia Lopes','82542391840','VitaliaLopes@gmail.com','C1').
staff('S12','Domitilo Guerreiro','29735739606','DomitiloGuerreiro195@outlook.com','C1').
staff('S13','Clarinda Ramos','61761802470','ClarindaRamos2007@hotmail.com','C4').
staff('S14','Ferdinando Faria','22252030175','FerdinandoFaria@gmail.com','C3').
staff('S15','Elpidio Campos','25846359652','ElpidioCampos1972@outlook.com','C3').
staff('S16','Xico Macedo','41779629692','XicoMacedo1973@gmail.com','C4').
staff('S17','Adonilo Coelho','57502712689','AdoniloCoelho@gmail.com','C1').
staff('S18','Erika Assuncao','64622347121','ErikaAssuncao@hotmail.com','C3').

```

```

staff('S19','Jabim Sa','86532268329','JabimSa@gmail.com','C1').

centro('C1','Centro de saude de Braga','Rua Paulo Vi 4700-004
→ Braga','219229219','suporte@debraga.com').
centro('C2','Centro de saude do Caranda','Rua de Goa Azurem 4800-004
→ Guimaraes','246512282','suporte@docaranda.com').
centro('C3','Centro de saude do Cavado','Rua de Sao Paulo 4700-004
→ Braga','233330348','suporte@docavado.com').
centro('C4','Centro de saude Dr.Paulo Novais','Rua Francisco Ribeiro de Castro Azurem 4800-004
→ Guimaraes','219282688','suporte@drpaulonovais.com').

vacinacao('S10','U1','2021-06-08','Moderna',1).
vacinacao('S10','U1','2021-06-30','Moderna',2).
vacinacao('S15','U2','2021-07-30','Pfizer',1).
vacinacao('S15','U2','2021-08-19','Pfizer',2).
vacinacao('S5','U3','2021-07-07','Pfizer',1).
vacinacao('S5','U3','2021-07-25','Pfizer',2).
vacinacao('S0','U4','2021-04-03','Moderna',1).
vacinacao('S0','U4','2021-04-24','Moderna',2).
vacinacao('S3','U5','2021-06-08','Sputnik V',1).
vacinacao('S3','U5','2021-06-27','Sputnik V',2).
vacinacao('S4','U6','2021-07-05','Astrazeneca',1).
vacinacao('S4','U6','2021-07-22','Astrazeneca',2).
vacinacao('S0','U7','2021-06-15','Astrazeneca',1).
vacinacao('S0','U7','2021-07-02','Astrazeneca',2).
vacinacao('S10','U8','2021-11-21','Sputnik V',1).
vacinacao('S10','U8','2021-12-08','Sputnik V',2).
vacinacao('S10','U9','2021-07-05','Pfizer',1).
vacinacao('S10','U9','2021-07-27','Pfizer',2).
vacinacao('S0','U10','2021-07-21','Sputnik V',1).
vacinacao('S0','U10','2021-08-13','Sputnik V',2).
vacinacao('S7','U11','2021-06-19','Pfizer',1).
vacinacao('S7','U11','2021-07-07','Pfizer',2).
vacinacao('S2','U12','2021-07-16','Moderna',1).
vacinacao('S2','U12','2021-08-05','Moderna',2).
vacinacao('S4','U13','2021-05-11','Astrazeneca',1).
vacinacao('S4','U13','2021-05-30','Astrazeneca',2).
vacinacao('S5','U14','2021-06-30','Sputnik V',1).
vacinacao('S5','U14','2021-07-23','Sputnik V',2).
vacinacao('S17','U15','2021-04-11','Pfizer',1).
vacinacao('S17','U15','2021-05-04','Pfizer',2).
vacinacao('S9','U16','2021-07-12','Astrazeneca',1).
vacinacao('S9','U16','2021-08-01','Astrazeneca',2).
vacinacao('S13','U17','2021-07-27','Pfizer',1).
vacinacao('S13','U17','2021-08-17','Pfizer',2).
vacinacao('S12','U18','2021-07-23','Sputnik V',1).
vacinacao('S12','U18','2021-08-15','Sputnik V',2).
vacinacao('S8','U19','2021-07-13','Pfizer',1).

```



```

vacinacao('S8','U19','2021-07-30','Pfizer',2).
vacinacao('S15','U20','2021-07-30','Sputnik V',1).
vacinacao('S15','U20','2021-08-16','Sputnik V',2).
vacinacao('S3','U21','2021-07-07','Pfizer',1).
vacinacao('S3','U21','2021-07-27','Pfizer',2).
vacinacao('S1','U22','2021-06-04','Sputnik V',1).
vacinacao('S1','U22','2021-06-25','Sputnik V',2).
vacinacao('S12','U23','2021-06-07','Pfizer',1).
vacinacao('S12','U23','2021-06-28','Pfizer',2).
vacinacao('S17','U24','2021-07-03','Astrazeneca',1).
vacinacao('S17','U24','2021-07-26','Astrazeneca',2).
vacinacao('S15','U25','2021-06-16','Moderna',1).
vacinacao('S15','U25','2021-07-04','Moderna',2).
vacinacao('S16','U26','2021-07-12','Sputnik V',1).
vacinacao('S16','U26','2021-08-05','Sputnik V',2).
vacinacao('S0','U27','2021-06-06','Sputnik V',1).
vacinacao('S0','U27','2021-06-29','Sputnik V',2).
vacinacao('S1','U28','2021-07-05','Sputnik V',1).
vacinacao('S1','U28','2021-07-26','Sputnik V',2).
vacinacao('S1','U29','2021-04-03','Astrazeneca',1).
vacinacao('S1','U29','2021-04-26','Astrazeneca',2).
vacinacao('S6','U30','2021-05-11','Pfizer',1).
vacinacao('S6','U30','2021-05-30','Pfizer',2).
vacinacao('S13','U31','2021-05-06','Pfizer',1).
vacinacao('S13','U31','2021-05-23','Pfizer',2).
vacinacao('S9','U32','2021-07-30','Sputnik V',1).
vacinacao('S9','U32','2021-08-20','Sputnik V',2).
vacinacao('S5','U33','2021-04-20','Moderna',1).
vacinacao('S5','U33','2021-05-09','Moderna',2).
vacinacao('S9','U34','2021-05-28','Moderna',1).
vacinacao('S9','U34','2021-06-18','Moderna',2).
vacinacao('S8','U35','2021-07-26','Astrazeneca',1).
vacinacao('S8','U35','2021-08-13','Astrazeneca',2).
vacinacao('S6','U36','2021-07-04','Pfizer',1).
vacinacao('S6','U36','2021-07-24','Pfizer',2).
vacinacao('S16','U37','2021-05-03','Pfizer',1).
vacinacao('S16','U37','2021-05-23','Pfizer',2).
vacinacao('S18','U38','2021-06-09','Pfizer',1).
vacinacao('S18','U38','2021-07-01','Pfizer',2).
vacinacao('S6','U39','2021-07-18','Pfizer',1).
vacinacao('S6','U39','2021-08-07','Pfizer',2).
vacinacao('S0','U40','2021-05-07','Astrazeneca',1).
vacinacao('S0','U40','2021-05-26','Astrazeneca',2).
vacinacao('S0','U41','2021-07-20','Astrazeneca',1).
vacinacao('S0','U41','2021-08-13','Astrazeneca',2).
vacinacao('S11','U42','2021-07-04','Sputnik V',1).
vacinacao('S11','U42','2021-07-21','Sputnik V',2).
vacinacao('S13','U43','2021-05-10','Pfizer',1).

```



```

vacinacao('S13','U43','2021-06-03','Pfizer',2).
vacinacao('S8','U44','2021-07-21','Sputnik V',1).
vacinacao('S8','U44','2021-08-14','Sputnik V',2).
vacinacao('S11','U45','2021-06-17','Pfizer',1).
vacinacao('S11','U45','2021-07-09','Pfizer',2).
vacinacao('S19','U46','2021-06-20','Astrazeneca',1).
vacinacao('S19','U46','2021-07-14','Astrazeneca',2).
vacinacao('S7','U47','2021-04-22','Moderna',1).
vacinacao('S7','U47','2021-05-15','Moderna',2).
vacinacao('S16','U48','2021-06-13','Pfizer',1).
vacinacao('S16','U48','2021-07-05','Pfizer',2).
vacinacao('S4','U49','2021-05-26','Astrazeneca',1).
vacinacao('S4','U49','2021-06-15','Astrazeneca',2).
vacinacao('S14','U50','2021-07-22','Astrazeneca',1).
vacinacao('S14','U50','2021-08-09','Astrazeneca',2).
vacinacao('S12','U51','2021-06-11','Pfizer',1).
vacinacao('S12','U51','2021-07-04','Pfizer',2).
vacinacao('S0','U52','2021-06-03','Sputnik V',1).
vacinacao('S0','U52','2021-06-23','Sputnik V',2).
vacinacao('S5','U53','2021-05-27','Pfizer',1).
vacinacao('S5','U53','2021-06-13','Pfizer',2).
vacinacao('S6','U54','2021-06-09','Moderna',1).
vacinacao('S6','U54','2021-06-29','Moderna',2).
vacinacao('S18','U55','2021-04-11','Sputnik V',1).
vacinacao('S18','U55','2021-05-02','Sputnik V',2).
vacinacao('S12','U56','2021-06-26','Pfizer',1).
vacinacao('S12','U56','2021-07-19','Pfizer',2).
vacinacao('S3','U57','2021-06-30','Pfizer',1).
vacinacao('S3','U57','2021-07-20','Pfizer',2).
vacinacao('S13','U58','2021-05-18','Astrazeneca',1).
vacinacao('S13','U58','2021-06-08','Astrazeneca',2).
vacinacao('S13','U59','2021-07-10','Moderna',1).
vacinacao('S13','U59','2021-08-01','Moderna',2).
vacinacao('S0','U60','2021-07-20','Pfizer',1).
vacinacao('S0','U60','2021-08-07','Pfizer',2).
vacinacao('S16','U61','2021-04-21','Pfizer',1).
vacinacao('S16','U61','2021-05-11','Pfizer',2).
vacinacao('S5','U62','2021-07-02','Sputnik V',1).
vacinacao('S5','U62','2021-07-22','Sputnik V',2).
vacinacao('S12','U63','2021-06-27','Moderna',1).
vacinacao('S12','U63','2021-07-15','Moderna',2).
vacinacao('S2','U64','2021-02-22','Pfizer',1).
vacinacao('S2','U64','2021-03-18','Pfizer',2).
vacinacao('S18','U65','2021-07-02','Sputnik V',1).
vacinacao('S18','U65','2021-07-21','Sputnik V',2).
vacinacao('S2','U66','2021-07-24','Sputnik V',1).
vacinacao('S2','U66','2021-08-14','Sputnik V',2).
vacinacao('S12','U67','2021-06-05','Sputnik V',1).

```

```

vacinacao('S12','U67','2021-06-26','Sputnik V',2).
vacinacao('S16','U68','2021-07-26','Astrazeneca',1).
vacinacao('S16','U68','2021-08-17','Astrazeneca',2).
vacinacao('S13','U69','2021-06-07','Pfizer',1).
vacinacao('S13','U69','2021-06-28','Pfizer',2).
vacinacao('S19','U70','2021-07-18','Pfizer',1).
vacinacao('S19','U70','2021-08-08','Pfizer',2).
vacinacao('S2','U71','2021-07-11','Pfizer',1).
vacinacao('S2','U71','2021-07-29','Pfizer',2).
vacinacao('S0','U72','2021-07-06','Sputnik V',1).
vacinacao('S0','U72','2021-07-23','Sputnik V',2).
vacinacao('S19','U73','2021-07-26','Sputnik V',1).
vacinacao('S19','U73','2021-08-16','Sputnik V',2).
vacinacao('S3','U74','2021-06-30','Astrazeneca',1).
vacinacao('S3','U74','2021-07-17','Astrazeneca',2).
vacinacao('S14','U75','2021-07-03','Astrazeneca',1).
vacinacao('S14','U75','2021-07-23','Astrazeneca',2).
vacinacao('S1','U76','2021-05-28','Moderna',1).
vacinacao('S1','U76','2021-06-15','Moderna',2).
vacinacao('S9','U77','2021-07-30','Moderna',1).
vacinacao('S9','U77','2021-08-19','Moderna',2).
vacinacao('S16','U78','2021-06-27','Sputnik V',1).
vacinacao('S16','U78','2021-07-16','Sputnik V',2).
vacinacao('S18','U79','2021-06-29','Moderna',1).
vacinacao('S18','U79','2021-07-21','Moderna',2).
vacinacao('S4','U80','2021-06-11','Sputnik V',1).
vacinacao('S4','U80','2021-07-01','Sputnik V',2).
vacinacao('S7','U81','2021-05-28','Pfizer',1).
vacinacao('S7','U81','2021-06-20','Pfizer',2).
vacinacao('S13','U82','2021-04-26','Moderna',1).
vacinacao('S13','U82','2021-05-18','Moderna',2).
vacinacao('S9','U83','2021-02-23','Moderna',1).
vacinacao('S9','U83','2021-03-12','Moderna',2).
vacinacao('S6','U84','2021-06-06','Pfizer',1).
vacinacao('S6','U84','2021-06-28','Pfizer',2).
vacinacao('S3','U85','2021-06-21','Moderna',1).
vacinacao('S3','U85','2021-07-11','Moderna',2).
vacinacao('S8','U86','2021-06-18','Pfizer',1).
vacinacao('S8','U86','2021-07-05','Pfizer',2).
vacinacao('S17','U87','2021-04-12','Astrazeneca',1).
vacinacao('S17','U87','2021-05-03','Astrazeneca',2).
vacinacao('S13','U88','2021-06-02','Pfizer',1).
vacinacao('S13','U88','2021-06-21','Pfizer',2).
vacinacao('S2','U89','2021-06-29','Pfizer',1).
vacinacao('S2','U89','2021-07-17','Pfizer',2).
vacinacao('S15','U90','2021-04-08','Sputnik V',1).
vacinacao('S15','U90','2021-04-25','Sputnik V',2).
vacinacao('S7','U91','2021-06-12','Moderna',1).

```

```

vacinacao('S7','U91','2021-06-30','Moderna',2).
vacinacao('S12','U92','2021-07-06','Sputnik V',1).
vacinacao('S12','U92','2021-07-26','Sputnik V',2).
vacinacao('S14','U93','2021-05-25','Astrazeneca',1).
vacinacao('S14','U93','2021-06-11','Astrazeneca',2).
vacinacao('S17','U94','2021-06-02','Pfizer',1).
vacinacao('S17','U94','2021-06-20','Pfizer',2).
vacinacao('S3','U95','2021-04-25','Pfizer',1).
vacinacao('S3','U95','2021-05-17','Pfizer',2).
vacinacao('S0','U96','2021-06-25','Sputnik V',1).
vacinacao('S0','U96','2021-07-19','Sputnik V',2).
vacinacao('S18','U97','2021-07-02','Pfizer',1).
vacinacao('S18','U97','2021-07-24','Pfizer',2).
vacinacao('S12','U98','2021-07-21','Pfizer',1).
vacinacao('S12','U98','2021-08-11','Pfizer',2).
vacinacao('S13','U99','2021-06-29','Sputnik V',1).
vacinacao('S13','U99','2021-07-18','Sputnik V',2).
vacinacao('S13','U100','2021-01-01','Pfizer',1).
vacinacao('S13','U100','2021-01-23','Pfizer',2).
vacinacao('S2','U101','2021-10-01','Pfizer',1).
vacinacao('S2','U101','2021-10-23','Pfizer',2).
vacinacao('S4','U102','2021-10-01','Moderna',1).
vacinacao('S4','U102','2021-10-23','Astrazeneca',2).
vacinacao('S13','U103','2021-10-01','Sputnik V',1).
vacinacao('S13','U103','2021-10-23','Sputnik V',2).
vacinacao('S13','U103','2021-10-23','Sputnik V',2).
vacinacao('S4','U104','2021-10-01','Moderna',1).
vacinacao('S4','U104','2021-10-02','Moderna',2).

```

```

fase('F1','2020-12-01','2021-03-31').
fase('F2','2021-04-01','2021-08-31').
fase('F3','2021-09-01','9999-12-31').

```

A.3 tp1.pl

```

% Este ficheiro vai conter: as declarações iniciais / os invariantes / as funcionalidades.
% -----
% SRCR TP1
% -----

% -- INCLUDES --
:- include('auxiliares.pl').

```

```

:- set_prolog_flag( disjoint_warnings,off ).
:- set_prolog_flag( single_var_warnings,off ).
:- set_prolog_flag( unknown,fail ).

:- op(900,xfy,'::').
:- dynamic utente/11.
:- dynamic staff/5.
:- dynamic centro/5.
:- dynamic vacinacao/6.

%-----
% INVARIANTES
%-----

% Garantir que o ID de cada utente é único:
+utente(Id,_,_,_,_,_,_,_,_,_,_,_) :: (solucoes(Id, utente(Id,_,_,_,_,_,_,_,_,_,_,_), R),
                                     length(R, 1)).

% Garantir que o ID de cada centro é único:
+centro(Id,_,_,_,_) :: (solucoes(Id, centro(Id,_,_,_,_), R),
                       length(R, 1)).

% Garantir que o ID de cada staff é único:
+staff(Id,_,_,_) :: (solucoes(Id, staff(Id,_,_,_), R),
                    length(R, 1)).

% Garantir que não há vacinações repetidas:
+vacinacao(Id,B,C,D,E) :: (solucoes(Id, vacinacao(Id,B,C,D,E), R),
                           length(R, 1)).

% Garantir que o ID de cada fase é único:
+fase(Id,_,_) :: (solucoes(Id, fase(Id,_,_), R),
                 length(R, 1)).

% Garantir que não posso eliminar um elemento se tiver vacinacoes marcadas
% Só pode haver 1 utente por id TODO: Mudar para solucoes quando for implementado nas auxiliares
+utente(Id,_,_,_,_,_,_,_,_,_,_,_) :: (findall(Id, utente(Id,_,_,_,_,_,_,_,_,_,_,_), R),
                                     length(R, 1)).

% Os utentes só podem ser do sexo masculino ou feminino
+utente(_,_,_,G,_,_,_,_,_,_,_,_) :: genderValido(G).

% Os utentes têm de ter um ID de centro existente
+utente(_,_,_,_,_,_,_,_,IdC) :: centro(IdC,_,_,_,_).

% Garantir que utentes com IDs diferentes têm diferente informacao
+utente(Id,N,NSS,G,DN,E,TLF,M,P,D,IDC) :: (solucoes((Id,NSS,E,TLF), utente(_,_,_,NSS,_,_,_,_,_,_,_,_),
↪ R),

```

```

length(R,1)).

%Garantir que centros com IDs diferentes t^em diferente informacao
+centro(Id,N,_,T,E) :: (solucoes((Id,N,T,E), centro(_,N,_,T,E), R),
length(R,1)).

%Garantir que centros com IDs diferentes t^em diferente informacao
+staff(Id,_,SS,T,C) :: (solucoes((SS,T,C), staff(_,_,SS,T,C), R),
length(R,1)).

%Garantir que não posso eliminar um elemento se tiver vacinacoes marcadas
-utente(Id,_,_,_,_,_,_,_,_) :: (findall(Id, vacinacao(_,Id,_,_,_), R),
\+length(R, 0)).

%Adicionar uma vacina c ao da toma 2 requer que a toma 1 j a esteja na base de conhecimento:
+vacinacao(S,U,_,_,2) :: vacinacao(S,U,_,_,1).

% Garantir que o género do utente é 'M' ou 'F'
+utente(_,_,_,G,_,_,_,_,_) :: generoValido(G).

-staff(Id,_,_,_,_) :: (findall(Id, vacinacao(Id,_,_,_,_), R),
\+length(R, 0)).

-centro(Id,_,_,_,_) :: (findall(sId,staff(sId,_,_,_,Id),R),
\+length(R, 0)).

%Garantir que o inicio de uma fase seja antes do fim da mesma
+fase(_,DI,DF) :: comparaDatasStr(DI,DF).

%-----
% REGISTRAR utentes, staff, centros de saude e vacinações
%-----

% UTENTE: Idutente, Nº Segurança_Social, Nome, Genero, Data_Nasc, Email, Telefone, Morada, Profissão,
→ [Doenças_Crónicas], #CentroSaúde
novoUtente(Id,N,NISS,G,Dn,E,Tlf,M,P,Dc,IdCs) :-
→ novoConhecimento(utente(Id,N,NISS,G,Dn,E,Tlf,M,P,Dc,IdCs)).
removeUtente(Id) :- removeConhecimento(utente(Id,_,_,_,_,_,_,_,_,_)).

% STAFF: IdStaff, IdCentro, Nome, Email
novoStaff(Id,NSS,IdCs,N,E) :- novoConhecimento(staff(Id,NSS,IdCs,N,E)).
removeStaff(Id) :- removeConhecimento(staff(Id,_,_,_,_)).

% CENTROS DE SAUDE: IdCentro, Nome, Morada, Telefone, Email
novoCentro(Id,IdCs,N,M,Tlf,E) :- novoConhecimento(centro(Id,IdCs,N,M,Tlf,E)).
removeCentro(Id) :- removeConhecimento(centro(Id,_,_,_,_,_)).

% VACINAÇÕES: IdStaff, IdUtente, Data, Vacina, Toma

```

```

novaVacinao(IdS,IdU,D,V,T) :- novoConhecimento(vacinao(IdS,IdU,D,V,T)).
removeVacinao(IdS,T) :- removeConhecimento(vacinao(IdS,_,_,T)).

%-----
% FUNCIONALIDADES
%-----

% Identificar pessoas nao vacinadas que sao candidatas em vacinao (para uma fase em especifico)

% Não tomaram nenhuma dose

% Utentes candidatos a vacinao da fase 1 que ainda nao tomaram nenhuma vacina
candidatosVacinaoT(1,R) :- % utentes que ainda nao tomaram nenhuma vacina
    naoVacinados(S),
    % utentes candidatos + de 80 anos
    findall(X,(member(X,S),calcularIdade(X,C), C>=80), I),
    % utentes candidatos que sao profissionais de saude
    findall(X,(member(X,S),profissao1Fase(X)), Pr),
    % utentes candidatos + de 50 anos e com doencas da fase 1
    findall(X,(member(X,S),doencas1Fase(X),calcularIdade(X,C),C>=50), Do),
    % juntar as listas
    append(I,Pr,App), append(App,Do,App2),
    % eliminar ids repetidos
    repetidos(App2,R).

% Utentes candidatos a vacinao da fase 2 que ainda nao tomaram nenhuma vacina
candidatosVacinaoT(2,R) :- % utentes que ainda nao tomaram nenhuma vacina
    naoVacinados(S),
    % utentes candidatos com + de 65 anos
    findall(X,(member(X,S),calcularIdade(X,C),C>=65), I),
    % utentes candidatos de 50 a 64 anos com doencas da fase 2
    findall(X,(member(X,S),doencas2Fase(X),calcularIdade(X,C),C>=50,C<64), Do),
    % juntar as listas e elim repetidos
    append(I,Do,App), repetidos(App,Rep),
    % eliminar os que sao da fase 1
    candidatosVacinaoT(1,F1),
    findall(X,(member(X,Rep),\+member(X,F1)), R).

% Utentes candidatos a vacinao da fase 3 que ainda nao tomaram nenhuma vacina
candidatosVacinaoT(3,R) :- % utentes que ainda nao tomaram nenhuma vacina
    naoVacinados(S),
    % utentes candidatos nao vacinados da f1 e f2
    candidatosVacinaoT(1,F1),
    candidatosVacinaoT(2,F2),
    append(F1,F2,App), repetidos(App,Rep),
    % identificar os restantes
    findall(X,(member(X,S),\+member(X,Rep)), R).

```

```

% Input de fase indevido
candidatosVacinacaoT(A,_) :- \+ pertence(A,[1,2,3]) -> write('Fase invalida. As fases podem ser 1, 2 ou
→ 3.'),

fail.

% Só tomaram a primeira dose

% Utentes candidatos a vacinacao da fase 1 que ainda so tomaram a primeira dose da vacina
candidatosVacinacaoP(1,R) :- % utentes que so tomaram a dose 1
    vacinadosSoUmaDose(S),
    % utentes candidatos + de 80 anos
    findall(X,(member(X,S),calcularIdade(X,C), C>=80), I),
    % utentes candidatos que sao profissionais de saude
    findall(X,(member(X,S),profissao1Fase(X)), Pr),
    % utentes candidatos + de 50 anos e com doencas da fase 1
    findall(X,(member(X,S),doencas1Fase(X),calcularIdade(X,C),C>=50), Do),
    % juntar as listas
    append(I,Pr,App), append(App,Do,App2),
    % eliminar os que sao da fase 1
    repetidos(App2,R).

% Utentes candidatos a vacinacao da fase 2 que ainda so tomaram a primeira dose da vacina
candidatosVacinacaoP(2,R) :- % utentes que so tomaram a dose 1
    vacinadosSoUmaDose(S),
    % utentes candidatos com + de 65 anos
    findall(X,(member(X,S),calcularIdade(X,C),C>=65), I),
    % utentes candidatos de 50 a 64 anos com doencas da fase 2
    findall(X,(member(X,S),doencas2Fase(X),calcularIdade(X,C),C>=50,C<=64), Do),
    % juntar as listas e elim repetidos
    append(I,Do,App), repetidos(App,Rep),
    % eliminar os que sao da fase 1
    candidatosVacinacaoP(1,F1),
    findall(X,(member(X,Rep),\+member(X,F1)), R).

% Utentes candidatos a vacinacao da fase 3 que ainda so tomaram a primeira dose da vacina
candidatosVacinacaoP(3,R) :- % utentes que so tomaram a dose 1
    vacinadosSoUmaDose(S),
    % ident utentes candidatos nao vacinados da f1 e f2
    candidatosVacinacaoP(1,F1),
    candidatosVacinacaoP(2,F2),
    append(F1,F2,App), repetidos(App,Rep),
    % identificar os restantes
    findall(X,(member(X,S),\+member(X,Rep)), R).

% Input de fase indevido
candidatosVacinacaoP(A,_) :- \+ pertence(A,[1,2,3]) -> write('Fase invalida. As fases podem ser 1, 2 ou
→ 3.'),

fail.

```



```

vacinacaoFaseErradaAux([],Acc,Acc).
vacinacaoFaseErradaAux([H|T],Acc,R) :- (vacinacaoFaseErradaUtente(H,X),X = 1),
    ↪ vacinacaoFaseErradaAux(T,Acc,R).
vacinacaoFaseErradaAux([H|T],Acc,R) :- vacinacaoFaseErradaAux(T,[H|Acc],R).

%Ser vacinado numa fase que não lhe corresponde
vacinacaoFaseErradaUtente(UId,X) :- (checkFaseUtente(UId,R1),
    pertenceAFase(UId,R2)),
    (R1 \= R2 -> X = 0; X = 1).

%Ser vacinado

%Haver uma diferença de vacinas da toma 1 para a 2 (duas vacinas diferentes)
diferente(R):-
    solucoes(U_ID,vacinaDiferente(U_ID),R).

vacinaDiferente(U_ID):-
    vacinacao(_,U_ID,_,V1,1),
    vacinacao(_,U_ID,_,V2,2),
    V1 \= V2.

%Tomar mais do que duas doses
tomouMaisDe2Doses(R) :- listaIdUtentes(X),
    tomouMaisDe2DosesAux(X,[],R).

tomouMaisDe2DosesAux([],Acc,Acc).
tomouMaisDe2DosesAux([H|T],Acc,R) :- (quantasVacinas(H, N), N<3), tomouMaisDe2DosesAux(T,Acc,R).
tomouMaisDe2DosesAux([H|T],Acc,R) :- tomouMaisDe2DosesAux(T,[H|Acc],R).

%Ver utentes q estão num grupo de risco e não foram vacinados, ou seja, utente idu existe não mas existe
↪ vacinação(idu)
utentesRiscoSemVacinacao(R) :- candidatosVacinacaoT(1,L1), candidatosVacinacaoT(2,L2),
    append(L1,L2,L3), repetidos(L3,R).

%Import dos dados
:- include('dados.pl').

```