Cuidado:

Na nossa base de conhecimento o predicado cuidado representa um serviço médico prestado a um determinado utente. Para o caracterizarmos de uma forma completa precisamos de saber a quem foi prestado o serviço, quem foi o prestador do serviço, qual foi o tipo do serviço, quando foi prestado, em que instituição acorreu e qual foi o custo total que esse serviço implicou.

Pelos motivos referidos decidimos criar o seguinte predicado:

**cuidado(Data, IdUtente, IdPrestador, Descrição, Custo, Instituição )**

Exemplo de alguns cuidados registados na nossa base de conhecimento:

**cuidado( data( 2018,3,14,20 ),1,1,'curativo',23,'Hospital de Braga' ).**

**cuidado( data( 2018,3,10,20 ),2,3,'rotina',23,'Hospital de Guimaraes' ).**

**cuidado( data( 2018,3,10,20 ),3,3,'rotina',23,'Hospital de Guimaraes' ).**

**Invariantes:**

De maneira a que a nossa base de conhecimento fosse consistente tanto a um nível estrutural como referencial decidimos criar alguns invariantes que nos ajudassem a manter essa consistência.

* **Data válida:** Não é possível inserir a informação relativa a um cuidado se a data desse cuidado não foi válida.

**Invariante:** *+cuidado( Data,U,P,Descricao,C,I ) :: (dataValida(Data)).*

Neste caso apenas temos de utilizar o predicado dataValida que irá verificar se todos os valores presentes na data são válidos.

* **Profissional apenas atende a 3 pedidos por hora:** Um prestador de serviços apenas está admitido a prestar 3 serviços por hora, logo temos de verificar senão existem já 3 cuidados daquele profissional à hora indicada.

**Invariante*:*** *+cuidado(D,U,P,Des,C,I) :: (solucoes((D,P),cuidado(D,Ut,P,Descr,Cus,Ins),L),*

*comprimento(L,X), X<4).*

Utilizando o predicado soluções colocamos numa lista pares que contem informações sobre o prestador em questão e a data. Após isso verificamos se essa lista contem menos que 3 pares.

* **Utente apenas recebe 3 serviços por hora:** Um utente está limitado a receber no máximo 3 serviços por hora, logo temos de verificar o numero de serviços que o utente recebeu na hora em questão.

**Invariante***:+cuidado(D,U,P,Des,C,I) ::(solucoes((D,U),cuidado(D,U,Pr,Descr,Cus,Ins),L),*

*comprimento(L,X),X<4).*

Modo de funcionamento é semelhante ao anterior, a única diferença é que a verificação é feita para o utente.

* **Cuidado tem utente válido:**  Não podemos adicionar cuidados para os quais não existam utentes registados, logo temos de verificar se o utente presente no cuidado é válido.

**Invariante:** *+cuidado( \_,IdU,\_,\_,\_,\_ ) :: (solucoes( IdU,utente( IdU,\_,\_,\_ ),L ),comprimento( L,X ),X == 1).*

Utilizando o predicado soluções construímos uma lista com o ID do utente presente no cuidado e verificamos se essa lista contem lá um ID.

* **Cuidado tem prestador válido:** Não podemos adicionar cuidados para os quais não existam prestadores registados, logo temos de verificar se o prestador presente no cuidado é válido.

**Invariante:** *+cuidado( \_,\_,IdP,\_,\_,\_ ) :: (solucoes( IdP,prestador( IdP,\_,\_,\_ ),L ), comprimento( L,X ), X == 1).*

Modo de funcionamento semelhante ao anterior, a única diferença é que a verificação é feita para o prestador.

* **Prestador tem de fazer da parte da instituição onde é feito o curativo:** Como os prestadores não trabalham em todas as instituições temos de confirmar que ele faz parta da instituição indicada no cuidado.

**Invariante: +cuidado( \_,\_,IdP,\_,\_,I ) :: (solucoes( LI,( prestador( IdP,\_,\_,LI ),pertence( I,LI ) ),L ),**

**comprimento( L,X ),**

**X == 1).**

Conjugando o predicado pertence e soluções primeiramente colocamos numa lista as instituições a que o prestador em questão trabalha e de seguida verificamos se a instituição indicada no cuidado pertence a essa mesma lista.