**Utente:**

Na nossa base de conhecimento o predicado utente representa um indivíduo ao qual é prestado um determinado cuidado, podemos então deduzir, que no mundo real o utente será um cidadão comum. Portanto tem de possuir um nome, mas como um nome não identifica inequivocamente uma pessoa numa atividade é necessário um identificador. Precisamos ainda de um campo com a idade dos cidadãos, DIZER PORQUE PRECISAMOS DA IDADE e também da sua morada para que possamos identificar os utentes de diferentes localidades. Para caraterizar a morada de um utente decidimos criar um predicado morada, sendo que esta contém um campo relativo à rua da morada do mesmo, outra relativo à localidade e outro relativo à cidade.

Pelas razões referidas anteriormente, decidimos criar o predicado:

**utente( ID,Nome,Idade,morada( Rua,Localidade,Cidade ) )**

De seguida apresentamos alguns exemplos relativos a utentes presentes na base de conhecimento:

EXEMPLOS

**Invariantes relativos ao utente:**

Como referido anteriormente, o ID (campo) identifica inequivocamente um utente. Por isto não pode ser possível, aquando da inserção de novos utentes, inserir um utente com um identificador que já esteja presente na base de conhecimento. Decidimos criar um invariante para garantir esta questão. (Tanto é válido para o utente como para o prestador)

A manipulação do invariante foi feita da seguinte forma:

+utente( IdU,\_,\_,\_ ) :: (solucoes( IdU,utente( IdU,\_,\_,\_ ),L ),

comprimento( L,X ),

X =< 1).

Apenas temos de utilizar o predicado soluções (que nos dá todas as soluções para um predicado presentes na base de conhecimento), sendo que o predicado que procuramos são o utente com o identificador igual ao do utente adicionado. Depois apenas temos de garantir que a lista de soluções que obtivemos tem um comprimento inferior ou igual a 1 (poderia ser só igual).

Não é possível remover um utente da base de conhecimento para os quais ainda existem cuidados, visto que iria retirar consistência à mesma. Para garantir o referido, criamos um invariante. . (Tanto é válido para o utente como para o prestador)

A manipulação do invariante foi feita da seguinte forma:

-utente( IdU,\_,\_,\_ ) :: (solucoes( D,cuidado(D,IdU,\_,\_,\_,\_ ),L ),

comprimento( L,X ),

X == 0).

Utilizamos o predicado soluções para obter todas as datas dos cuidados para o utente com o identificador do utente que queremos retirar. Depois apenas podemos remover se o comprimento dessa lista for zero ou seja, caso não existam cuidados.