

Diogo Braga

Diogo Silva

João Silva

Ricardo Caçador

Ricardo Veloso

**2º Exercício do Trabalho de Grupo:**

**Programação em Lógica Estendida e Conhecimento Imperfeito**

**Universidade do Minho**

Departamento de Informática

**2º Exercício do Trabalho de Grupo:**

**Programação em Lógica Estendida e Conhecimento Imperfeito**

Diogo Braga

Diogo Silva

João Silva

Ricardo Caçador

Ricardo Veloso

**Universidade do Minho**

Departamento de Informática



# Resumo

# ------- POR FAZER -------

O trabalho representado neste relatório foi desenvolvido no âmbito da UC de Sistemas de Representação de Conhecimento e Raciocínio por forma a desenvolver competências na utilização da linguagem de programação em lógica - PROLOG.

Este exercício consistiu no desenvolvimento de uma base de conhecimento e raciocínio para caracterizar um universo de discurso na área da prestação de cuidados de saúde.

Este relatório irá explicar todo o processo que envolveu a criação dessa base até ao resultado final.

**Tabela de Conteúdos**

[Resumo 3](#_Toc6753857)

[------- POR FAZER ------- 3](#_Toc6753858)

[1 – Introdução 6](#_Toc6753859)

[------- POR FAZER ------- 6](#_Toc6753860)

[2 - Descrição do Trabalho 7](#_Toc6753861)

[2.1- Representação de Conhecimento Positivo 7](#_Toc6753862)

[3 – Conclusão 8](#_Toc6753863)

[------- POR FAZER ------- 8](#_Toc6753864)

**Tabela de Figuras**

# 1 – Introdução

# ------- POR FAZER -------

Esta primeira fase tem como objetivo a criação de um sistema representação de conhecimento e raciocínio que caracterize a estrutura de uma área de prestação de cuidados de saúde. A criação deste sistema é feita através da utilização da linguagem de programação PROLOG.

Para o efeito, foi-nos apresentado um panorama possível para caracterizar o conhecimento bem como um conjunto de funcionalidades que o sistema deve respeitar.

De seguida, iremos apresentar todas as soluções realizadas pelo grupo para a consumação do exercício proposto bem como as extensões de conhecimento implementadas no sistema.

# 2 - Descrição do Trabalho

Devido à ligação que existe com a primeira fase do projeto, é importante relembrar os predicados que foram criados:

* utente: IdUt, Nome, Idade, Cidade, Seguro -> {V,F,D}
* serviço: IdServ, Descrição, Instituição, Cidade -> {V,F,D}
* consulta: Data, IdUt, IdServ, Custo, IdMed -> {V,F,D}
* data: Dia, Mes, Ano -> {V,F}
* medico: IdMed, Nome, Idade, IdServ -> {V,F,D}
* seguro: IdSeg, Descrição, Taxa -> {V,F,D}

Tendo em conta a Programação em Lógica Estendida abordada nesta fase, é importante diferenciar que, em complemento com um sistema de inferência, agora os predicados podem resultar num valor de Verdadeiro, Falso ou Desconhecido.

## 2.1- Representação de Conhecimento Positivo

A representação de conhecimento positivo foi replicada da primeira fase, tendo em conta que já nesse momento tinha existido este tipo de representação de conhecimento.

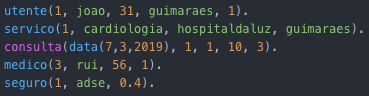
Desta forma, a seguir apresentamos um exemplo de conhecimento positivo, para cada um dos predicados estabelecidos no projeto:

Figura 1 – Exemplos de Conhecimento Positivo

## 2.2 – Representação de Conhecimento Negativo

A representação de conhecimento negativo foi algo que, devido aos novos requisitos provenientes da Lógica Estendida, fez sentido criar no trabalho, pois desta forma tornamos possível representar algo que é falso. Tal é possível representar através da negação explícita, como através da negação forte.

**Negação Explícita**

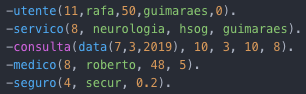
De seguida, são apresentados alguns exemplos de negação explícita dos predicados envolvidos no trabalho, associados à explicação dos mesmos.

Figura 2 – Exemplos de Negação Explícita

Exemplos:

1. Não existe um utente de 50 anos com id 11 e nome rafa, que viva em guimaraes sem seguro.
2. Não existe um serviço de neurologia, com id 8, no hsog em guimaraes.
3. Não existe uma consulta do dia 07/03/2019, do utente com o id 10 para o serviço com o id 3, no qual o custo monetário é 10 e o id do médico é 8.
4. Não existe um médico de ginecologia com o id 8, chamado roberto, e com 48 anos de idade.
5. Não existe um seguro com id 4 e nome secur que possuía uma taxa de retorno de 0.2.

**Negação Forte**

Este género negação assenta na aplicação do Pressuposto do Mundo Fechado na programação em Lógica Estendida. Com tal aplicação garantimos que algo que não tem prova de ser verdadeiro, é implicitamente falso.

De seguida, é apresentada a extensão do predicado que define a negação forte do predicado utente. As extensões dos restantes predicados são realizadas da mesma forma, só que com o predicado respetivo.



Figura 3- Negação Forte do predicado utente

Esta negação do utente parametrizado na figura é validada caso não exista prova que o utente seja conhecimento verdadeiro, e caso não exista prova duma exceção associada a esse mesmo utente.

## 2.3 – Representação Conhecimento Imperfeito

# 3 – Conclusão

# ------- POR FAZER -------

O grupo considera que realizou um trabalho bastante aprofundado tendo respondido às questões propostas no enunciado do exercício de uma forma simples e correta. O facto de termos criado um número significativo de funcionalidades extra faz-nos acreditar que temos um trabalho bastante completo e conciso.

Como trabalho futuro alguns aspetos no que conta à implementação do sistema de seguros poderiam ser melhorados criando por exemplo situações de utentes isentos de pagamento.

De uma forma geral, sentimos que os conceitos de sistemas de representação de conhecimento e raciocínio pedidos neste primeiro exercício ficaram bem consolidados e solidificamos a nossa capacidade de utilizar a linguagem PROLOG.