

# Trabalho 3 de Inteligência Artificial

## Trabalho realizado por:

-Cristiana Morais da Silva up201505454

-Sara Daniela Ferreira de Sousa up201504217

## 1. Introdução

O paradigma declarativo (ou linguagem declarativa) baseado em lógica como é o caso do prolog recorre a uma base de dados com regras lógicas e cria uma resolução passo a passo para responder a queries colocadas pelos utilizadores do programa.

O paradigma imperativo (ou linguagem imperativa) como o java destaca-se pela sua simplicidade, uma vez que qualquer pessoa ao fazer o programa, o implementa de forma imperativa, baseando-se na utilização de ações e estados, quase como um programa de computador.

A principal desvantagem deste paradigma é que se as estruturas de dados forem modificadas, o código de todas as funções no programa tem de ser modificado.

## 2. Trip\_Plan

A primeira parte do programa trata-se de um planeador de viagens que recorrendo a uma base de dados de voos entre várias cidades, tem de ser capaz de conseguir responder a questões do tipo:

- Em que dias é que existem ligações diretas entre dois aeroportos;
- Quais são os voos que temos de apanhar num determinado dia para conseguir ir de um aeroporto para outro;
- Planear uma rota de viagem com início e fim na mesma cidade sendo necessário que este percurso passe por determinados locais.

### 2.1. Implementação em Prolog

Esta implementação baseia-se na utilização dos predicados dados no enunciado do trabalho. Os voos estão guardados numa timetable(Place1,Place2,List\_of\_flights), em que List\_of\_flights é uma lista de elementos do tipo (Dep\_time/Arr\_time/Flight\_num/Ndays, List\_of\_flights), em que Flight\_num é o número do voo e Ndays são os dias em que existem voos que pode ser uma lista de dias ou uma constante alldays (que contém todos os dias da semana).

O predicado `flight/6` tem por único objetivo representar os dados de uma base em dados que podemos manipular no programa.

O predicado `route/4` representa o caminho direto entre duas cidades, ou o caminho entre duas cidades usando outras cidades como ligação.

O predicado `trip_plan/5` permite-nos planear um percurso ao longo de várias cidades de forma a que seja feito um único voo por dia.

## **2.2. Implementação em Java**

A implementação em java foi um pouco mais extensa uma vez que foi necessária a criação de várias estruturas de dados. Depois de interpretados os dados, estes são guardados numa árvore, onde se encontram todas as ligações entre aeroportos e os seus respetivos horários.

## **3. DCG**

A segunda parte do trabalho é pedida a implementação de um analisador sintático de língua portuguesa(ou DCG) que verifica se uma frase esta corretamente estruturada em português, ou seja se é composta por uma frase nominal seguida de uma frase verbal sendo utilizada uma base de dados com regras de concordância de frase.

### **3.1. Implementação em Prolog**

A implementação em prolog deste problema tornou-se mais simples uma vez que o DCG é nativo do prolog e sendo assim foi mais fácil analisar uma frase e verificar se ela se encontra correta a nível da sintaxe.

### **3.2. Implementação em Java**

A implementação deste problema em java foi mais complexa pois foi necessária a utilização de estruturas de dados (Listas Ligadas) para guardar frase dada como input, e ainda de bastantes linhas de código para guardar as várias palavras dadas no enunciado do problema.

## **4. Conclusão e comentário**

Com o decorrer do trabalho ficou claro que a implementação dos problemas em prolog era mais simples mas em contrapartida foi mais trabalhoso uma vez que não era uma linguagem tão familiar .

Em suma nenhuma das linguagens é pior que a outra, apenas se aplicam diferentes paradigmas, dando origem a diferentes abordagens do mesmo problema.

## **5. Bibliografia**

[https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o\\_imperativa](https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_imperativa)

[https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o\\_declarativa](https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_declarativa)

<https://en.wikipedia.org/wiki/Prolog>

<https://www.metalevel.at/prolog/dcg>

<https://pt.stackoverflow.com/questions/81854/paradigma-imperativo-e-declarativo>

<http://www.pathwayslms.com/swipltuts/dcg/>