



UNIVERSIDADE DE ÉVORA

# Trabalho Prático de Programação III

## 2021/2022

Realizado por: Gonçalo Correia (43735) e Jorge Salsinha (45447)

## **1. Introdução**

Pretende-se com este trabalho a implementação em Prolog de um programa que seja capaz de resolver um puzzle Strimko a partir de um ficheiro de input dado pelo utilizador com o número de colunas/linhas, a disposição das streams no tabuleiro e as posições pré-preenchidas no mesmo.

## 2. Implementação

Na implementação deste trabalho foi usado apenas o interpretador SWI-Prolog e foram implementadas algumas diferenças em relação ao enunciado, nomeadamente o facto de as streams e o nome do ficheiro .txt necessitarem de ser inseridos no próprio código e só funcionar para matrizes 4x4, sendo estes os únicos problemas no funcionamento deste trabalho.

### 2.1. Ficheiros e Bibliotecas

Inicialmente recorreremos ao exercício da aula prática para colocar em funcionamento a leitura do ficheiro de input, estando presente do código principal através de um import na segunda linha do mesmo. Foi também recorrente ao longo deste trabalho o uso de *Constraint Logic Programming* e por isso, na primeira linha do código foi importada a library clpfd para se poder manusear predicados tal como all\_distinct/1 e labelling/2 .

### 2.2. Predicados

Além das descrições que se encontram na seguinte lista de predicados, estas podem também ser complementadas pelos comentários ao longo do código strimko.pl:

- read1/1: pega no predicado read\_lines criado em files.pl e insere-se no próprio código o nome do ficheiro de input entre plicas.
- converter0/2 e converter/2: este dois predicados trabalham em conjunto de modo a converter um array que contém arrays de strings em valores inteiros.
- converterInput/1: pega nos predicados anteriores e aplica converter/2 ao read1/1.
- fill\_strimko/2: recebe como primeiro argumento um tabuleiro e preenche-o com os dados do array que está presente no segundo argumento.
- readValores/1: guarda apenas os valores das posições pré-preenchidas do puzzle presentes em converterInput/1.
- dosomething/2 e dosomething2/2: estes dois predicados recebem os valores já convertidos e aplicam fill\_strimko/2 a um tabuleiro.
- go/1: serve para executar o programa, e funciona a partir de uma função auxiliar goaux/1 com os valores de dosomething2/2.

- `diff/1`: faz com que os valores dentro de cada array sejam diferentes uns dos outros com o objetivo de ser aplicado para diferir os valores das linhas/colunas/streams através do predicado `all_distinct/1`.
- `goaux/1`: recebe apenas o tabuleiro do jogo aplicando o predicado `diff/1` aos elementos de cada linha, coluna e stream, de modo a serem diferentes uns dos outros, usa o predicado `labelling/1` para que cada posição do array tenha um valor específico caso o tabuleiro esteja vazio e finalmente uma série de `writes` para produzir o output do programa.

### **3. Execução**

Para utilizar esta aplicação basta ir ao código de `strimko.pl`, mudar o conteúdo entre aspas da linha 5 para o ficheiro de input desejado, alterar o conteúdo das streams na linha 61 de acordo com o que está no input e executar `go(_)`.

## **4. Conclusão**

Este trabalho foi um verdadeiro desafio pois esta linguagem apresenta-se bastante distinta de todas as outras lecionadas no curso, causando bastante confusão na altura de raciocinar e implementar certas funcionalidades, daí terem ficado a faltar certos aspetos para o programa funcionar 100% como era pedido. Apesar disso foi excelente para ficar mais a par de como trabalhar com Prolog que é uma linguagem bastante característica.