

# Engenharia Gramatical

## MEI

### Guião Prático 2

Pedro Rangel Henriques   Tiago Baptista

Pretende-se que comece por definir uma linguagem de programação imperativa simples, a seu gosto. Apenas deve ter em consideração que essa linguagem terá de permitir:

- *declarar* variáveis atômicas do tipo *inteiro*, com os quais se podem realizar as habituais operações aritméticas, relacionais e lógicas.
- *efetuar* instruções algorítmicas básicas como a *atribuição do valor de expressões numéricas a variáveis*.
- *efetuar* instruções *condicionais* para controlo do fluxo de execução.
- *efetuar* instruções *cíclicas* para controlo do fluxo de execução, permitindo o seu aninhamento.

Após definida a linguagem pretende-se desenvolver um analisador de código, utilizando o lark, que permita :

- A declaração de variáveis para aceitar variáveis atômicas (inteiras e booleanas) e estruturadas (arrays, listas e conjuntos).
- Comece por Detetar e contar a presença de ifs.
- Depois complete a análise e assinale os ifs aninhados e indique também o nível de aninhamento
- Retornar alternativa para os ifs aninhados detetados (se for possível).

**Input :**

```
if(x){  
    if(y){  
        x=0;  
    }  
}
```

**Output :**

```
if(x && y){  
    x=1  
}
```

- Detetar e encontrar variáveis declaradas mas não inicializadas no código.

**Input :**

```
int x;  
int y;  
if(x>1){  
}
```

**Output :** x e y não inicializados.

**Nota:** Deve e pode utilizar como base a gramática desenvolvida no trabalho prático da cadeira de PL.