

# Perfil de EL - Engenharia de Linguagens (1º ano do MEI)

## Trabalho Prático 2 (TP2) de EG – Engenharia Gramatical

Ano Letivo 2024/25

### 1 Analisador de Código Fonte

Como estudou no 1º Trabalho Prático, existem várias *ferramentas avançadas de análise de código* (ou seja, de *programas-fonte em Linguagens de Programação de alto-nível*) com vista a ajudar em tarefas diferentes: embelezar textualmente a escrita do programa; detetar situações que infringem as boas-práticas de codificação na linguagem em causa ou que podem ser vulneráveis durante a execução; sugerir *melhores* formas de codificar sem alterar o significado do programa-fonte; avaliar a performance do programa estática ou dinamicamente.

Agora, neste 2º Trabalho Prático, pretende-se que desenvolva um Analisador de Código para a Linguagem de Programação Imperativa (LPI) que desenhou no início do semestre.

Para que a análise que se pretende fazer neste trabalho, note que a sua linguagem deve permitir, conforme foi pedido inicialmente, declarar variáveis atómicas e estruturadas (incluindo como no `Python` as estruturas: *conjunto*, *lista*, *tuplo*, *dicionário*), instruções de selecção (condicionais) e pelo menos 3 variantes de ciclos.

Concretamente, neste TP2, deve escrever em `Python`, usando o `Parser` e os `Visitors` do módulo para geração de processadores de linguagens `Lark`, uma ferramenta que analise programas escritos na sua linguagem LPI e gere em HTML um relatório com os resultados dessa análise, nomeadamente:

1. Lista de todas as variáveis do programa indicando os casos de: *redeclaração*<sup>1</sup> ou *não-declaração*<sup>2</sup>; variáveis usadas<sup>3</sup> mas *não inicializadas*<sup>4</sup>; variáveis declaradas e *nunca mencionadas*.
2. Total de variáveis declaradas por cada Tipo de dados usados.
3. Total de instruções que formam o corpo do programa, indicando o número de instruções de cada tipo (*atribuições*, *leitura e escrita*, *condicionais* e *cíclicas*).
4. Total de situações em que estruturas de controlo surgem aninhadas em outras estruturas de controlo do mesmo ou de tipos diferentes.
5. Lista de situações em que existam `ifs` aninhados que possam ser substituídos por um só `if`.

De maneira a permitir um bom acompanhamento do trabalho vão existir tarefas a serem cumpridas durante as aulas práticas, nomeadamente :

- Criação da tabela de símbolos para albergar todos os identificadores (de variáveis, tipos, funções, ou outro) que existam no programa. (Obs: comece por identificar todos os campos que precisamos de ter na referida tabela);
- Travessia para recolher as informações e construir a tabela de símbolos;

---

<sup>1</sup>Uma variável declarada 2 vezes.

<sup>2</sup>Uma variável mencionada no código sem ter sido inicialmente declarada.

<sup>3</sup>Uma variável que está a ser mencionada como operando de uma operação ou como parametro na chamada de uma função.

<sup>4</sup>Uma variável à qual nunca foi atribuído um valor.

- Criação de estrutura para albergar informação acerca das estruturas de controlo.
- Travessia para recolher as informações sobre as estruturas de controlo;

Como é habitual, o TP será entregue na blackboard, por apenas um elemento do grupo, até dia **28 de Abril**. Devem fazê-lo na forma de um relatório desenvolvido em  $\text{\LaTeX}$ .