

Trabalho Prático Nº2 - Compiladores

Marco Sousa^{1,2[62608]}, José Malheiro^{1,2[93271]}, and Miguel Fernandes^{1,2[94269]}

¹ Universidade do Minho

² Licenciatura em Engenharia Informática, Braga, Portugal

Abstract. Keywords: Expressões regulares · Python · Templates · Gramática Independente de Contexto · Compiladores.

1 Introdução

1.1 Contextualização

O presente relatório foi escrito no âmbito da Unidade Curricular (UC) de Processamento de Linguagens (PL), apresentando como objetivo a descrição da solução desenvolvida para o segundo trabalho prático, assim como as decisões tomadas para a sua conceção.

A solução é relativa ao primeiro enunciado: **Linguagens de Templates** (inspirada nos templates *Pandoc*).

Neste sentido, é proposto o desenvolvimento de um compilador de *templates*, de modo a criar ficheiros *markup*, a partir do processamento de *templates* definidos. Para tal usou-se a linguagem *YAML* para guardar toda a informação a ser usada.

Assim, engloba o desenvolvimento do **analizador léxico** e o **parser**, bem como a gramática e a estrutura da árvore sintática criadas para poder gerar o texto final.

1.2 Breve Descrição do Trabalho Proposto

O enunciado escolhido, **Linguagem de templates** (inspirada nos templates *Pandoc*), propõe a criação de um compilador de *templates*. Deste modo, utiliza-se:

1. um *Template*.
2. um Dicionário.

...para construir o texto final, um **ficheiro *markup***, com todos os dados submetidos pelo utilizador.

Neste sentido, é pertinente definir a gramática que todos os *templates*

2 HandleDoc

- Gramática abstrata
 - máquina de estados
 - estratégia utilizada – árvore de parsing concreta – classes – hierarquia – descentralização das responsabilidades – diagrama de classes
 - Funcionalidades – if – elseif – else
 - for
 - pipes
 - subtemplating
 - range
 - it
 - sep
- CONSTRUIR MANUAL!!

2 Marco Sousa, José Malheiro, and Miguel Fernandes

3 Conclusões

4 Anexos