

Processamento de Linguagens

Engenharia Informática (3º ano)

Projeto Final

23 de Março de 2023

Dispõe de **7 semanas** para desenvolver este trabalho, a entrega deverá ser feita até à meia noite de **14 de Maio**.

1 Objectivos e Organização

Este trabalho prático tem como principais objectivos:

- aumentar a experiência em engenharia de linguagens e em programação generativa (gramatical), reforçando a capacidade de escrever gramáticas, quer independentes de contexto (GIC), quer tradutoras (GT);
- desenvolver processadores de linguagens segundo o método da tradução dirigida pela sintaxe, a partir de uma gramática tradutora;
- desenvolver um compilador gerando código para um objetivo específico;
- utilizar geradores de compiladores baseados em gramáticas tradutoras, concretamente o Yacc, versão PLY do Python, completado pelo gerador de analisadores léxicos Lex, também versão PLY do Python.

Na resolução dos trabalhos práticos desta UC, aprecia-se a imaginação/criatividade dos grupos em todo o processo de desenvolvimento!

Deve entregar a sua solução até Domingo dia 14 de Maio. O ficheiro com o relatório e a solução deve ter o nome 'pl2023-projeto-grNN', em que NN corresponderá ao número de grupo. O número de grupo será ou foi atribuído no registo das equipas do projeto.

Cada grupo, **é livre** para escolher qual o enunciado que pretende desenvolver.

A submissão deverá ser feita por email com os seguintes dados:

to: jcr@di.uminho.pt

subject: PL2023::grNN::Projeto::Enunciado

body: Colocar um ZIP com os ficheiros do Projeto: relatório, código desenvolvido e datasets de teste.

Em cima, "Enunciado" é um dos valores: Pandoc, Ply-Simple, RecDesc, TDTabela.

Na defesa, a realizar na semana de 16 a 20 de Maio, o programa desenvolvido será apresentado aos membros da equipa docente, totalmente pronto e a funcionar (acompanhado do respectivo relatório de desenvolvimento) e será defendido por todos os elementos do grupo, em data e hora a marcar. O relatório a elaborar, deve ser claro e, além do respectivo enunciado, da descrição do problema, das decisões que lideraram o desenho da solução e sua implementação, deverá conter exemplos de utilização (textos fontes diversos e respectivo resultado produzido). Como é de tradição, o relatório será escrito em LaTeX.

2 Enunciados

2.5 conversor de Pug- - para HTML

Construa um conversor que aceite um subconjunto da linguagem Pug e gere o HTML correspondente (defina claramente a parte coberta).

Exemplo da notação Pug:

```
html(lang="en")
  head
    title= pageTitle
    script(type='text/javascript').
      if (foo) bar(1 + 5)
  body
    h1 Pug - node template engine
    #container.col
      if youAreUsingPug
        p You are amazing
      else
        p Get on it!
    p.
      Pug is a terse and simple templating language with a
        strong focus on performance and powerful features
```

HTML correspondente:

```
<html lang="en">
  <head><title></title>
    <script type="text/javascript">
      if (foo) bar(1 + 5)</script>
  </head>
  <body>
    <h1>Pug - node template engine</h1>
    <div class="col" id="container">
      <p>Get on it!</p>
      <p>Pug is a terse and simple templating language with a
        strong focus on performance and powerful features</p>
```

```
    </div>
  </body>
</html>
```

Ver também a documentação

<https://pugjs.org/language/tags.html>

<https://pugjs.org/language/plain-text.html>

<https://pugjs.org/language/attributes.html>