

INSTRUÇÕES

- A não entrega deste trabalho prático implica a reprovação à unidade curricular no ano letivo 2021/2022, **não sendo possível a sua realização em nenhuma época de avaliação.**
- O trabalho prático será realizado em Grupo com um **máximo d 2 alunos**;
- A data limite para a entrega do primeiro trabalho prático é o **dia 15 de Janeiro de 2022**. A submissão de trabalhos será feita apenas usando o formulário correspondente disponível no Moodle. Não serão aceites trabalhos práticos via e-mail;
- A apresentação do trabalho é feita por **todos os elementos do grupo**, de forma **individual**.
- Para além da implementação em Python + Ply (lex e yacc) do projeto, deverá ser preparado um **pequeno relatório** que explique de que forma o enunciado foi interpretado, e quais as decisões tomadas na sua implementação.

A. Portugal

O Portugal é uma linguagem algorítmica descrita em português. O seu objetivo é ser, por um lado, uma forma de os estudantes de programação começarem a esquematizar os seus programas, mas também garantir que os algoritmos definidos não se limitam ao papel, e podem ser executados.

No Moodle encontra-se uma apresentação sobre a sintaxe do Portugal. Existem muitos mais recursos pela Internet, que poderão ser consultados.

O objetivo deste trabalho é a implementação de um reconhecimento léxico e sintático da linguagem, que permite que, com base num ficheiro de texto com um programa em Portugal seja possível:

- interpretar o código, colocando deste modo o algoritmo a ser executado;
- gerar o código C correspondente ao algoritmo processado.

O utilizador deverá poder escolher, ao utilizar a ferramenta desenvolvida, qual das duas opções pretende.

A avaliação deste trabalho terá em conta:

- A qualidade do reconhecedor léxico e da gramática implementados;

- A organização e qualidade do código Python desenvolvido;
- A estrutura da Árvore Abstrata de Sintaxe;
- A quantidade de operadores do Portugol implementados;
- A possibilidade de geração de código C.

B. TQL: Text Query Language

Pretende-se a implementação de uma variante à SQL (structured query language), mais simples, e que funcione sobre ficheiros CSV (Comma Separated Value) armazenados numa pasta. Cada ficheiro CSV é guardado em formato tabular, tal como descrito no enunciado B do primeiro trabalho prático.

Considere-se que existem os seguintes ficheiros em determinada pasta:

- `produtos.csv`

```
ID,Descricao,Preco
B1,Balão Decorativo (10),1.00
B2,Balão da Minnie (5),1.50
Ax,Pipocas,1.00
```
- `vendas.csv`

```
ID,Qt
B1,100
B2,10
```

Note que cada ficheiro tem, como primeira linha, os nomes dos vários campos.

A linguagem TQL (Text Query Language) suportará os seguintes comandos:

Manuseamento de Tabelas

- `LOAD TABLE produtos FROM "produtos.csv";` Carrega o ficheiro CSV indicado para memória, armazenando-o com o nome `produtos`.
- `DISCARD TABLE produtos;` Remove os dados da tabela `produtos` da memória.
- `SAVE TABLE produtos AS "produtos2.csv";` Guarda uma tabela de memória para ficheiro.
- `SHOW TABLE produtos;` Lista uma tabela no ecrã.

Execução de *Queries*

- `SELECT * FROM produtos;` Lista todas as linhas e colunas da tabela indicada.
- `SELECT ID,Descricao FROM produtos;` Lista todas as linhas, mas apenas as colunas indicadas.
- `SELECT * FROM vendas WHERE Qt > 50;` Lista todas as linhas que respondam a determinada condição. Sugere-se a possibilidade de permitir mais do que uma condição, separadas por `AND`.

- Poderão ser realizados diferentes tipos de comparação: igualdade (=), desigualdade (<>), bem como as quatro comparações (<,>, <= and >=).
- Aos exemplos anteriores poderá ainda ser adicionado o modificador `LIMIT n` para limitar o número de resultados apresentados ao valor *n* indicado.

Criação de novas Tabelas

É possível criar novas tabelas em memória a partir de *queries* e de outras tabelas:

- `CREATE TABLE mais_vendidos FROM SELECT * FROM vendas WHERE Qt > 50;` Armazena numa nova tabela o resultado do *query*, permitindo para que se possa mais tarde armazenar o resultado num ficheiro.
- `CREATE TABLE tudo FROM vendas JOIN produtos USING(ID);` Para juntar tabelas pode-se criar uma nova tabela. Ao contrário do SQL, a junção de tabelas não irá permitir a seleção de colunas ou linhas, obrigando sempre à união completa de duas tabelas.

Procedimentos

É possível criar procedimentos. Os procedimentos são declarados usando o comando `PROCEDURE`, e terminam com o comando `END`:

```
PROCEDURE atualizar_vendas DO
  CREATE TABLE mais_vendidos FROM SELECT * FROM vendas WHERE Qt > 50;
  CREATE TABLE tudo FROM vendas JOIN produtos USING(ID);
END
```

Os procedimentos podem ser executados usando o comando `CALL`, por exemplo, `CALL atualizar_vendas;`.

A avaliação deste trabalho terá em conta:

- A qualidade do reconhecedor léxico e da gramática implementados;
- A organização e qualidade do código Python desenvolvido;
- A estrutura da Árvore Abstrata de Sintaxe;
- A quantidade de operadores do TQL implementados;

Note-se que não serão considerados como avaliação positiva soluções que apenas aceitem os exemplos descritos neste enunciado. A solução deve ser pensada como genérica, com capacidade de adaptação.