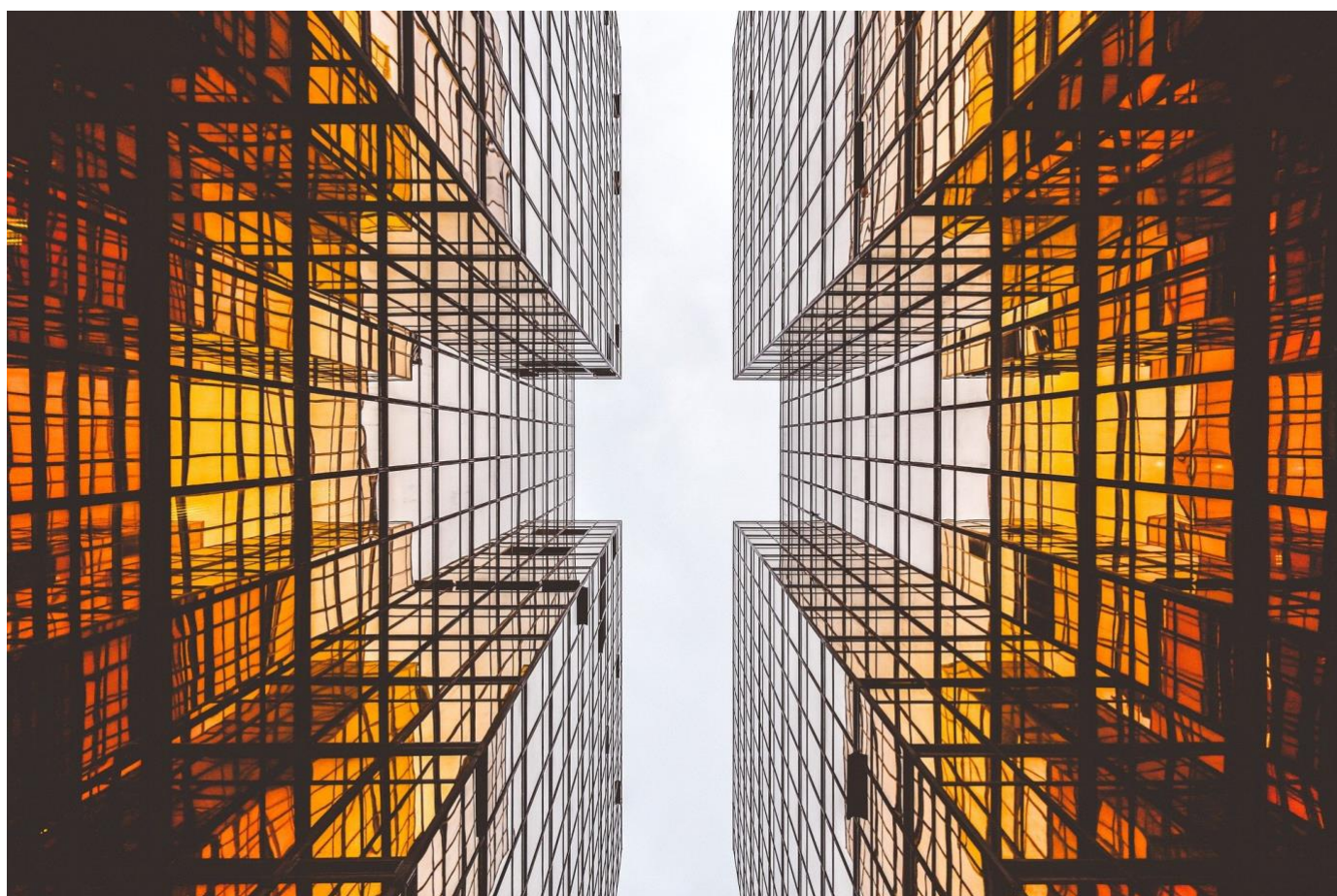


SPRINT B

LPROG

18 de junho, 2023



Elaborado por:

1170264 – Nuno Franco
1210816 – João Castro
1211430 – Inês Silva

1211432 – Maria Pinto
1211433 – Pedro Carvalho

Índice

Gramática do Exame..... 3

Introdução à gramática3

Estrutura do exame3

Estrutura da secção4

Retorno da validação5

Gramática das Perguntas 8

Introdução à gramática8

Estrutura das questões8

Gramática do Aluno Realizar o Exame 13

Introdução à gramática13

Estrutura das questões13

Gramática do Exame

Introdução à gramática

Um exame é composto por um título, que deve ser único na base de dados, uma descrição, uma data de início e fim, uma linguagem e um header. Este é composto por uma descrição, tipo de feedback e de nota (havendo três possibilidades para cada), e por um conjunto de secções, que contém uma descrição e um conjunto de perguntas, cuja gramática será abordada posteriormente.

Um exame automático é idêntico a um exame normal, mas este não tem data de início nem de fim.

Estrutura do exame

O exame é constituído por vários *tokens*, como é representado na figura abaixo:

```
stat: INIT title SEPARATOR description SEPARATOR openDate SEPARATOR  
closeDate SEPARATOR language SEPARATOR header SEPARATOR structure  
INIT;
```

O exame automático é semelhante ao anterior, mas sem “openDate” e “closeDate”:

```
stat: INIT title SEPARATOR description SEPARATOR language SEPARATOR  
header SEPARATOR structure INIT;
```

Desta forma, os *tokens* presentes na gramática acima, estão descritos na tabela:

Identificação	Descrição
title	Título do exame (validado quando o exame é persistido na base de dados, deve ser único)
description	Texto referente à descrição do exame
openDate	Data referente à data de início do exame. Formato: dd/mm/yyyy
closeDate	Data referente à data de fim do exame. Formato: dd/mm/yyyy
language	Texto referente à linguagem de cada exame
header	Conjunto de tokens referentes à informação do cabeçalho do exame
structure	Conjunto de tokens referentes à estrutura do exame, isto é, conjunto de secções e perguntas

O 'header' segue a seguinte estrutura:

```
header: headerDescription SEPARATOR feedbackType SEPARATOR gradeType;
```

A descrição do header é uma string. O 'feedbacktype' e 'gradetype' indica a forma como o feedback e a nota são retornadas ao aluno e podem ser de três tipos: 'NONE', 'ON_SUBMISSION' ou 'AFTER_CLOSING'.

Um exame automático apenas terá o feedbacktype' e 'gradetype' de um tipo, neste caso, 'ON_SUBMISSION'.

Identificação	Descrição
headerDescription	Texto referente à descrição do cabeçalho
feedbackType	String referente ao tipo de feedback do exame. Pode ser: 'NONE', 'ON_SUBMISSION' ou 'AFTER_CLOSING'
gradeType	String referente ao tipo de feedback do exame. Pode ser: 'NONE', 'ON_SUBMISSION'

Por fim, a estrutura (structure) refere-se a um conjunto de secções, cuja estrutura será abordada de seguida.

Estrutura da secção

O exame é constituído por vários *tokens*, como é representado na figura abaixo:

```
section: SECTIONOPEN TEXT SEPARATOR sectionStructure SECTIONCLOSE;

sectionStructure: question+

question: INITQUESTION questiontype INITQUESTION;

questiontype: NUM SEPARATOR matching
             | NUM SEPARATOR missingWord
             | NUM SEPARATOR multipleChoice
             | NUM SEPARATOR numeric
             | NUM SEPARATOR short
             | NUM SEPARATOR trueOrFalse
             ;
```

Desta forma, os *tokens* presentes na gramática acima, estão descritos na tabela:

Identificação	Descrição
TEXT	Texto referente à descrição da secção
sectionStructure	Texto referente a toda a estrutura do exame. Conjunto de perguntas
question	Pergunta, que segue a mesma estrutura da criação de perguntas soltas (dentro da estrutura de cada tipo, está definido também um token correspondente à cotação)

As perguntas seguem a mesma estrutura aplicada na criação de perguntas soltas, adicionando-se apenas um número, antes de especificar as mesmas, referente à numeração das mesmas no exame.

Retorno da validação

Desta forma, segue-se um exemplo possível da criação de um exame:

```
"
exame matcp;
descricao do exame de matcp;
29/10/2023;
01/11/2023;
linguagem;
descricao do header;
NONE;
ON_SUBMISSION;
<<
    seccaol;
    ??
        1;
        MatchingQuestion;

        match the translation;
        cao,gato;
        cat,dog;

        cao->dog, gato->cat;
    2
    ??
```

```

??
    2;
    MatchingQuestion;

    match the translation;
    porco,baleia;
    pig,whale;

    porco->pig, baleia->whale;

    1
??
>>
<<
seccao2;
??
    1;
    MultipleChoiceQuestion;

    how tall is the everest?;
    100metros,
    2000metros,
    7000metros;

    7000metros;
    5
??
??
    2;
    MissingWordQuestion;
    FILL *** GAP FOR *** PHRASE;
    the - that - these,the - that - this - these;
    the, this;
    2
??
??
    3;
    NumericQuestion;
    How tall is the everest?;
    7000;
    5
??
??
    4;
    ShortQuestion;
    What is the everest?;
    mountain;
    2
??
??
    5;
    TrueFalseQuestion;
    Is the everest tall?;
    True;
    3
??
>>
"
```

Agora vamos dar o exemplo de uma possível criação de um exame automático:

```
1  "n
2  autoexam;
3  this is an autoexam;
4  lang;
5  header;
6  ON_SUBMISSION;
7  ON_SUBMISSION;
8  <<
9      section1;
10     ??
11         1;
12         ShortQuestion
13     ??
14     ??
15         2;
16         MatchingQuestion
17     ??
18     ??
19         3;
20         ShortQuestion
21     ??
22 >>
23 <<
24     section2;
25     ??
26         1;
27         MultipleChoiceQuestion
28     ??
29     ??
30         2;
31         MissingWordQuestion
32     ??
33     ??
34         3;
35         NumericQuestion
36     ??
37     ??
38         4;
39         TrueFalseQuestion
40     ??
41 >>
42 "
```

Como podemos analisar neste ficheiro, os exames automáticos serão criados com base em tipos de questões e não nas questões em si, com o tipo de questões presente podemos ir á base de dados procurar por questões referentes a esse tipo, tornando o processo mais simples e automatizado.

Gramática das Perguntas

Introdução à gramática

As perguntas podem ser criadas de forma individual para serem persistidas na base de dados e serem posteriormente usadas na criação de exames automáticos. Existem seis tipos de pergunta, que adotam uma estrutura diferente mediante a necessidade de ter uma estrutura diferente para cada, isto é, a pergunta em si, mais, possivelmente, uma série de opções, e também uma resposta apresentada de forma diferente mediante o tipo de pergunta, como será abordado de seguida.

Estrutura das questões

O exame é constituído por vários *tokens*, como é representado na figura abaixo:

```
question: INITQUESTION questionType SEPARATOR quotation INITQUESTION;
quotation: NUM;

questionType: matching
             | missingWord
             | multipleChoice
             | numeric
             | short
             | trueOrFalse
             ;
```

Desta forma, os *tokens* presentes na gramática acima, estão descritos na tabela:

Identificação	Descrição
questionType	Esta string remete para o tipo de pergunta
quotation	Refere-se à cotação da pergunta

Quanto ao tipo da questão, pode assumir os seguintes valores:

Matching – A questão tem uma lista sub-perguntas, juntamente com uma lista de respostas. Espera uma resposta com o identificador da lista correta:

- **Estrutura:**

```
matching: 'MatchingQuestion' SEPARATOR matchingText
SEPARATOR matchingAnswers SEPARATOR quotation;

matchingText: TEXT SEPARATOR matchingOption SEPARATOR
matchingOption;

matchingOption: TEXT ',' matchingOption
| TEXT

;
```

- **Exemplo:**

```
??
(numeração da pergunta, se for a gramática do exame:"1;")

MatchingQuestion;

match the translation;
cao, gato;
cat, dog;

cao->dog,
gato->cat;

2
??
```

Select Missing Words - A questão é constituída por um conjunto de opções (Figura 5). Espera uma resposta com o identificador de uma das opções:

- **Estrutura:**

```
missingWord: 'MissingWordQuestion' SEPARATOR
missingWordText SEPARATOR missingWordAnswer;

missingWordText: mwText SEPARATOR missingWordOptions;

mwText: GAPSPACE TEXT mwText
| TEXT GAPSPACE mwText
| TEXT GAPSPACE
| GAPSPACE TEXT
| TEXT GAPSPACE TEXT

;
```

- **Exemplo:**

```
??
(numeração da pergunta, se for a gramática do exame:"1;")

MissingWordQuestion;

Um *** e uma estrutura de controlo que permite a ***
repetida de um bloco de código enquanto uma condição
especificada for verdadeira;
objeto - laço - grafo - serviço,
execução - gestão - repetição - organização;

laço, execução;
2
??
```

Multiple-choice - A questão é constituída por um conjunto de opções (*Figura 5*). Espera uma resposta com pelo menos um identificador de uma das opções:

- **Estrutura:**

```
missingWord: 'MissingWordQuestion' SEPARATOR
missingWordText SEPARATOR missingWordAnswer;

missingWordText: mwText SEPARATOR missingWordOptions;

mwText: GAPSPACE TEXT mwText
      | TEXT GAPSPACE mwText
      | TEXT GAPSPACE
      | GAPSPACE TEXT
      | TEXT GAPSPACE TEXT
      ;

missingWordOptions: missingWordOption ','
missingWordOption
      | missingWordOption;

missingWordOption: TEXT '-' missingWordOption
      | TEXT
      ;

missingWordAnswer: TEXT
      | TEXT ',' missingWordAnswer;
```

- **Exemplo:**

```
??
(numeração da pergunta, se for a gramática do exame:"1;")

MultipleChoiceQuestion;

how tall is the everest?;
100metros,
2000metros,
7000metros;

7000metros;
5
??
```

Numerical – Espera uma resposta numérica:

- **Estrutura:**

```
numeric: 'NumericQuestion' SEPARATOR numericText SEPARATOR  
numericAnswer;  
  
numericText: TEXT QUESTION? ;  
  
numericAnswer: NUM ;
```

- **Exemplo:**

```
??  
(numeração da pergunta, se for a gramática do exame:"1;")  
  
NumericQuestion;  
  
how tall is everest?;  
7000;  
  
5  
??  
  
??
```

Short Answer – Espera uma palavra ou um conjunto de palavras:

- **Estrutura:**

```
short: 'ShortQuestion' SEPARATOR shortText SEPARATOR  
shortAnswer;  
  
shortText: TEXT QUESTION? ;  
  
shortAnswer: TEXT;
```

- **Exemplo:**

```
??  
(numeração da pergunta, se for a gramática do exame:"1;")  
  
ShortQuestion;  
  
how tall is the everest?;  
  
7000 metros;  
5  
??
```

True/False - A questão é constituída por duas opções (*Figura 5*). Espera uma resposta com o identificador de uma das duas opções:

- **Estrutura:**

```
trueOrFalse: 'TrueFalseQuestion' SEPARATOR trueOrFalseText  
QUESTION? SEPARATOR trueOrFalseAnswer;  
  
trueOrFalseText: TEXT QUESTION? ;  
  
trueOrFalseAnswer: 'True'  
                   | 'False'  
                   ;
```

- **Exemplo:**

```
??  
  
TrueFalseQuestion;  
  
Is the everest the tallest mountain in the world?;  
  
True;  
5  
??
```

Gramática do Aluno Realizar o Exame

Introdução à gramática

O aluno pode realizar um exame pelos menus ou por um ficheiro, com as respostas ao exame escolhido, que passa por uma gramática, que valida a estrutura. A correção do exame é também feita nos métodos dos visitors.

Estrutura das questões

O exame é constituído por vários *tokens*, como é representado na figura abaixo:

```
stat: INIT structure INIT;
structure: section+;
section: TEXT TWO_DOTS sectionStructure;
sectionStructure: question+;
question: NUM DOT questionAnswer SEPARATOR;
```

Desta forma, os *tokens* presentes na gramática acima, estão descritos na tabela:

Identificação	Descrição
structure	Esta string remete para a estrutura do exame, que consiste num conjunto de secções e respetivas perguntas
section	A secção é definida pelo seu título/descrição, seguindo-se “:” e um conjunto de respostas.
sectionStructure	Esta string remeta para a estrutura das secções
question	Uma pergunta é definida por a sua numeração (“1.”) e a resposta a essa pergunta