

Gramáticas na Compreensão de Software

Relatório GAMu Grupo 9

Luís Vieira
nºa62819

Oleskii Gylytskyy
nºpg30469

18 de Janeiro 2016

1 Introdução

Este trabalho pretende facilitar o planeamento de uma audição através da criação de uma gramática que permita validar toda a informação fornecida pelo utilizador. São feitas validações quer da sintaxe quer da semântica das frases fornecidas, e é devolvida ao utilizador uma mensagem informativa da natureza do erro.

2 Gramática de Atributos

De seguida mostramos a Gramática de Atributos desenvolvida, onde foram omitidas regras de contexto, regras de tradução e Condições de contexto.

GA = < T, N, S, P, A >

T = {ID_ALUNO, ID_PROFESSOR, ID_OBRA, ID_INSTRUMENTO,
INT, STRING, COMMENT, ',', '#', '/', ':', 'ano-letivo:',
'titulo:', 'subtitulo:', 'tema:', 'data:', 'hora:', 'local:',
'duracao-maxima:', 'atuacoes:', 'grupo:', 'elementos:',
'obras:', 'solo'}

N = {audicao, metaAud, atuacoes, data, hora, duracao, atuacao,
grupo, solo, obras, musicos, musico, idObra, idInstrumento,
idProf, idAluno}

S = {audicao}

A = {id, linha, coluna, xmlIn, xmlOut, tempoIn, tempo,
hora_inicio, titulo_aud}

P = {
 audicao : metaAud atuacoes
 ;
 metaAud : 'organizador:' idProf
 'titulo:' STRING
 'subtitulo:' STRING
 'tema:' STRING
 'data:' data
 'hora:' hora
 'local:' STRING
 'duracao-maxima:' hora
 ;
 data : INT'-'INT'-'INT
 ;
 hora : horas=INT':'minutos=INT
 ;

```

atuacoes      :   'atuacoes:' atuacao '#' (atuacao '#')*
                ;

atuacao        :   grupo
                |   solo
                ;

grupo          :   'grupo:' STRING 'elementos:' musicos
                |   'obras:' obras
                ;

solo           :   'solo:' musico 'obras:' obra
                ;

obras          :   idObra (',' idObra)*
                ;

musicos        :   musico (',' musico)*
                ;

musico         :   idAluno ',' idInstrumento
                |   idProf ',' idInstrumento
                ;

idObra :    ID_OBRA;
idProf :    ID_PROFESSOR;
idAluno:    ID_ALUNO;
idInstrumento:    ID_INSTRUMENTO;
}

```

2.1 Sintaxe

Os tokens definidos para os identificadores têm nomes bastante elucidativos do seu domínio, com a finalidade de tornar as mensagens de erro mais claras quanto à sua natureza.

2.2 Semântica

Para validar o planeamento de uma audição foram verificadas as seguintes condições semânticas:

- identificador de aluno está registado
- identificador de professor está registado
- identificador de obra está registado
- identificador de instrumento está registado
- duração total não ultrapassa a duração máxima estimada para a audição
- aluno/professor tem habilitações para tocar instrumento

Estas validações consultam uma base de dados MySQL, onde estas entidade estão relacionadas. Em anexo segue o esquema dessa mesma base de dados.

3 Interface

O planeamento de uma audição é feito através de um editor integrado numa página web. Este editor auxilia o processo de criação e planeamento de uma audição através de uma descrição verbal e um indicador gráfico no editor. Podemos ver da imagem que se segue um exemplo de um planeamento com erros.

Criar

Validar Audição

carregar audição exemplo

Nova Audição

plano:

2ano-letivo: 2015/2016

3titulo: "Titulo"

4subtitulo: "subtitulo"

5tema: "tema"

6data: 20/12/2016

7hora: 13:30

8local: "local da audicao"

9duracao-maxima: 13:00

10atuacoes:

11grupo: "nome da banda"

12elementos: A1510,I07

13P11,I15

14A20021,I7

15obras: 01,

1602,

1703,

1804,

19032

20#

21solo: A1510,I6

22obras: 022,

23023

24#

25

26

27

28

29

30

31

32

33

34

35

submeter Audição

Mensagens do Servidor

O Aluno: A1510 tem habilitacoes em:

- I6 Flauta

- I9 Harpa

- I3 Corne Inglês

tempo total audicao: 10:02:18

duracao-maxima: 13:00:00

Planeamento da Audicao contem erros, alteracoes ignoradas.

line 12 instrumento: I07 nao existe

Figura 1: criação de uma audição

As tecnologias utilizadas para fazer a ponte entre o executável Java e plataforma web, foram JSON para as comunicações entre cliente-servidor, jQuery e AJAX do lado do cliente, e por fim PHP do lado do servidor que fica responsável por receber o texto do cliente e executar a gramática sobre esse texto.

4

Criar

Validar Audição

carregar audição exemplo

Nova Audição

plano:

2

ano-letivo: 2015/2016

3

titulo: "Titulo"

4

subtitulo: "subtitulo"

5

tema: "tema"

6

data: 20/12/2016

7

hora: 13:30

8

local: "local da audicao"

9

duracao-maxima: 13:00

10

atuacoes:

11

grupo: "nome da banda"

12

elementos: A1510, 66

13

P11, I15

14

A20021, I7

15

obras: Q1,

16

Q2,

17

Q3,

18

Q4,

19

Q32

20

#

21

solo: A1510, I6

22

obras: Q22,

23

Q23

24

#

25

26

27

28

29

30

31

32

33

34

35

submeter Audicao

Mensagens do Servidor

O Aluno: A1510 tem habilitacoes em:

- I6 Flauta

- I9 Harpa

- I3 Corne Inglês

tempo total audicao: 10:02:18

duracao-maxima: 13:00:00

Planeamento da Audicao contem erros, alteracoes ignoradas.

Existem erros de **Syntax!** Audicao nao inserida

line 12:19 mismatched input '66' expecting

ID_INSTRUMENTO

line 0 instrumento: null nao existe

Figura 2: exemplo erro sintaxe

O output do executável JAVA (gramática) é filtrado pelo javascript do lado do cliente, de forma a extrair o número da linha onde ocorreu o erro, isto pressupõe que as mensagens de erro tenham um formato bem definido.

As imagens que se seguem mostram este tipo de comportamento para erros semânticos.

Criar

Validar Audição

carregar audição exemplo

Nova Audição

plano:

2

ano-letivo: 2015/2016

3

titulo: "Titulo"

4

subtitulo: "subtitulo"

5

tema: "tema"

6

data: 20/12/2016

7

hora: 13:30

8

local: "local da audicao"

9

duracao-maxima: 13:00

10

atuacoes:

11

grupo: "nome da banda"

12

elementos: A1510, I7

13

P11, I15

14

A20021, I7

15

obras: Q1,

16

Q2,

17

Q3,

18

Q4,

19

Q32

20

#

21

solo: A1510, I6

22

obras: Q22,

23

Q23

24

#

25

26

27

28

29

30

31

32

33

34

35

submeter Audicao

Mensagens do Servidor

O Aluno: A1510 tem habilitacoes em:

- I6 Flauta

- I9 Harpa

- I3 Corne Inglês

tempo total audicao: 10:02:18

duracao-maxima: 13:00:00

Planeamento da Audicao contem erros, alteracoes ignoradas.

line 12 (aluno: A1510 - Ademar Fontes De Magalhaes Goncalves) nao tem habilitacoes para tocar (I7 - Fliscorne)

Figura 3: exemplo erro semântico

De seguida podemos ver mais um exemplo de um erro semântico e a respetiva mensagem a ele associado. A linha assinalada indica qual a obra cuja duração excede o tempo máximo previsto para a audição.

Por fim caso não exista nenhum tipo de erro, a audição é inserida no ficheiro do correspondente ano letivo, e o utilizador é informado através da mensagem que podemos ver na imagem que se segue.

Criar

Validar Audição

carregar audição exemplo

Nova Audição

plano:

2

ano-letivo: 2015/2016

3

titulo: "Titulo"

4

subtitulo: "subtitulo"

5

tema: "tema"

6

data: 20/12/2016

7

hora: 13:30

8

local: "local da audicao"

9

duracao-maxima: 9:00

10

atuacoes:

11

grupo: "nome da banda"

12

elementos: A1510,I6

13

P11,I15

14

A20021,I7

15

obras: 01,

16

02,

17

03,

18

04,

19

032

20

#

21

solo: A1510,I6

22

obras: 022,

23

023

24

#

25

26

27

28

29

30

31

32

33

34

35

submeter Audicao

Mensagens do Servidor

tempo total audicao: 10:02:18

duracao-maxima: 09:00:00

tempo total ultrapassa duracao estimada: +01:02:18

Planeamento da Audicao contem erros, alteracoes ignoradas.

line 23 - tempo_max ultrapassado O23 tem duracao: 10:02:18

Figura 4: exemplo erro semântico 2

Mensagens do Servidor

tempo total audicao: 09:05:16

duracao-maxima: 10:03:00

Audicao inserida com sucesso!

Figura 5: mensagem de sucesso

4 Conclusão

Apesar do elevado número de símbolos sintáticos definidos na nossa linguagem, acreditamos que o planeamento de uma audição se tornou mais simples graças à capacidade de verbalização da gramática, deixando ainda espaço em aberto para melhorias no que diz respeito ao tipo de mensagens que podem ser construídas com esta disposição das produções.

5 ANEXOS

5.1 Base de dados

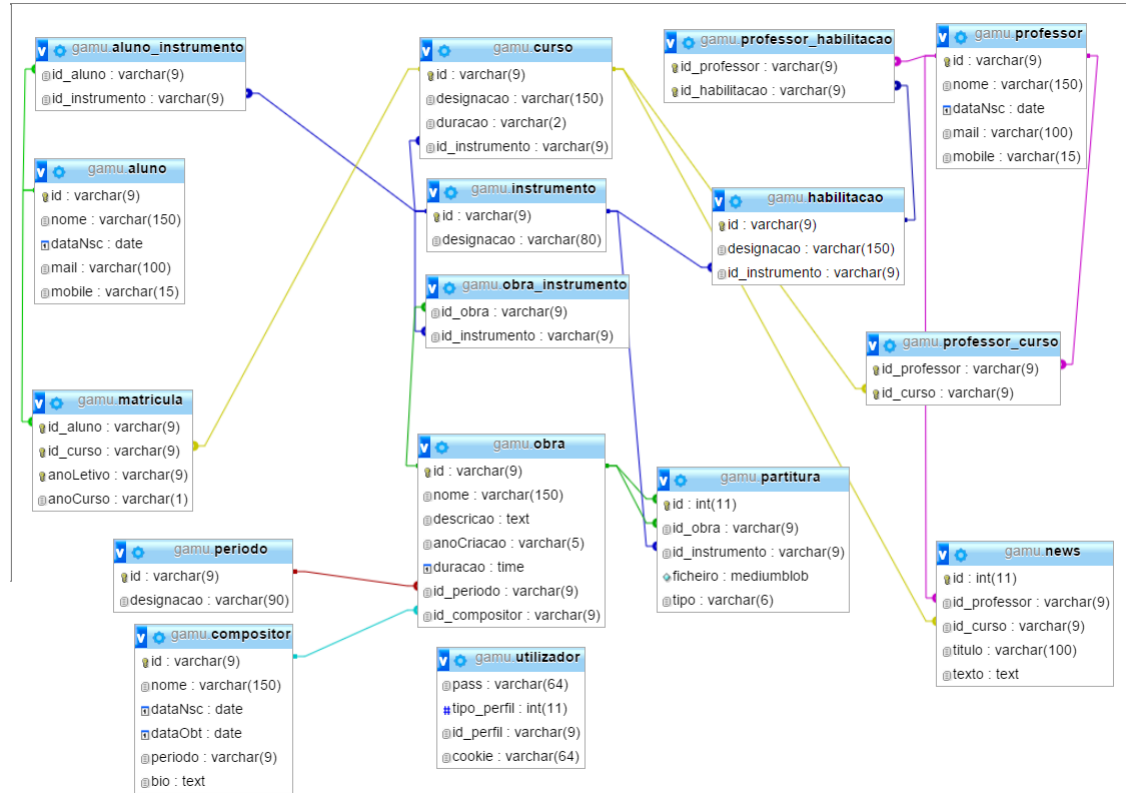


Figura 6: Modelo Lógico MySQL

6 Gramática

grammar GAMu;

```
@header{
import java.util.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.ResultSet;
```



```

import java.sql.Statement;
import java.sql.Time;
import java.time.Instant;
import java.io.PrintWriter;
import java.io.FileNotFoundException;
import java.util.logging.Level;
import java.util.logging.Logger;

import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.IOException;
import java.io.PrintStream;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpression;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;
import javax.xml.transform.OutputKeys;
}
@members{
    // JDBC driver name and database URL
    String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    String DB_URL = "jdbc:mysql://127.0.0.1:3306/gamu??
        useUnicode=true&characterEncoding=UTF-8";
    // Database credentials
    String USER = "root";
    String PASS = "qweqwe";
    // Database connection
    Connection conn = null;
    Statement stmt = null;

    // Grammar variables

```

```

/** flag!=0, ocorreu erro semantico.**/
private int flag = 0;

public int getFlag(){
return this.flag;
}

}

audicao      @init{
try{
//STEP 2: Register JDBC driver
Class.forName("com.mysql.jdbc.Driver");
//STEP 3: Open a connection
conn = (Connection) DriverManager.getConnection(DB_URL,
        USER,PASS);

//STEP 4: Execute a query
stmt = (Statement) conn.createStatement();

}catch(SQLException se){
//Handle errors for JDBC
//    se.printStackTrace();
}catch(Exception e){
//Handle errors for Class.forName
//e.printStackTrace();
}

//iniciar a construçao do XML
StringBuilder audicao_xml = new StringBuilder();
audicao_xml.append("<?xml_version=\"1.0\"_encoding=\"UTF
        -8\"?>\n");
}

@after{
try{
if(stmt!=null)
    stmt.close();
if(conn!=null)
    conn.close();
}catch(SQLException se){
//Handle errors for JDBC
//se.printStackTrace();
}

}

:      metaAud [audicao_xml]

```

```

{
//titulo = $metaAud.titulo_aud;
String anoLetivo = $metaAud.ano_letivo;
audicao_xml = $metaAud.xmlOut;
if(audicao_xml != null){
audicao_xml.append("</metainfo>\n<atuacoes>\n");
}else{
audicao_xml = new StringBuilder();
}

}
atuacoes [audicao_xml, $metaAud.hora_inicio, $metaAud.
tempo_max_aud]
{
Time audicao = new Time($atuacoes.tempo * 1000);
System.out.println("tempo_total_audicao:_"+ audicao.
toString());
Time max = new Time($metaAud.tempo_max_aud*1000);
System.out.println("duracao-maxima:_"+ max.toString());

if($metaAud.tempo_max_aud < $atuacoes.tempo){
long dif = $atuacoes.tempo - $metaAud.tempo_max_aud -
3600;
Time time = new Time(dif * 1000);
System.out.println("tempo_total_ultrapasssa_duracao_
estimada:<font_color=\"red\">_"+time.toString()+"</
font>");
}
audicao_xml = $atuacoes.xmlOut;
if(audicao_xml != null){
audicao_xml.append("</atuacoes>\n</audicao>");
}else{
audicao_xml = new StringBuilder();
}

//-----XML-----
//audicao valida?

if(anoLetivo!=null && flag!=1){
try {
String filepath = "C:\\xampp\\htdocs\\site\\files\\
auditions\\"+anoLetivo+".xml";
DocumentBuilderFactory docFactory =
DocumentBuilderFactory.newInstance();
DocumentBuilder docBuilder = docFactory.
newDocumentBuilder();

```

```

Document doc = docBuilder.parse(filepath);

String xml = audicao_xml.toString();
Document audi_xml = docBuilder.parse( new InputSource(new
    ByteArrayInputStream(xml.getBytes("utf-8"))));

Element basket = (Element) doc.getFirstChild();
Element fruit = (Element) audi_xml.getFirstChild();

// remover Nodo antigo
XPathFactory xpf = XPathFactory.newInstance();
XPath xpath = xpf.newXPath();
XPathExpression expression = xpath.compile("//audicao[@id
    =\"+$metaAud.titulo_aud+\"]");
Node audicao_antiga = (Node) expression.evaluate(doc,
    XPathConstants.NODE);
if(audicao_antiga != null){
    audicao_antiga.getParentNode().removeChild(
        audicao_antiga);
}

// Adicionar nova audicao
Node imported = doc.importNode(fruit, true);
basket.appendChild(imported);

// write the content into xml file
TransformerFactory transformerFactory =
    TransformerFactory.newInstance();
Transformer transformer = transformerFactory.
    newTransformer();
transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8
    ");
DOMSource source = new DOMSource(doc);
StreamResult result = new StreamResult(new File(filepath)
    );
transformer.transform(source, result);

} catch (ParserConfigurationException |
    TransformerException | IOException |
    XPathExpressionException | SAXException pce) {
    //pce.printStackTrace();
    System.out.println("erro_-nao_foi_possivel_guardar_
        audicao");
}
} else{

```

```

System.out.println("Planeamento_da_Audicao_contem_erros , _
    alteracoes_ignoradas.");
}
}
;

metaAud [StringBuilder xmlIn]
returns[String titulo_aud, String ano_letivo,
    StringBuilder xmlOut, long hora_inicio, int
    tempo_max_aud]
:
    'organizador:' id_prof=idProf
'ano-letivo:' a1=INT/'a2=INT {$ano_letivo = $a1.text+"_"
    +$a2.text;}
'titulo:' STRING {
    $titulo_aud = $STRING.text;
    $xmlIn.append("\n<audicao_id="+$STRING.text+"
        >");
    $xmlIn.append("\n<metainfo>");
    try{
        String sql = "SELECT_*_FROM_professor
            _WHERE_id='"+$id_prof.id+"'";
        ResultSet rs = (ResultSet) stmt.
            executeQuery(sql);
        if(rs.next()){
            $xmlIn.append("<organizador_id=\
                "+$id_prof.id+"\">"+rs.
                    getString("nome")+"</
                    organizador>\n");
        }else{
            System.out.print("line_"+$id_prof
                .linha+"_coluna:_" + $id_prof.
                    coluna);
            System.out.println("_professor:_
                "+$id_prof.id+"_nao_existe");
            flag=1;
        }
        rs.close();
    }catch(SQLException se){
        //se.printStackTrace();
    }
}
('subtitulo:' STRING{$xmlIn.append("<subtitulo>"+$STRING.
    text+"</subtitulo>\n");})?
('tema:' STRING{$xmlIn.append("<tema>"+$STRING.text+"</
    tema>\n");})?
'data:' data[$xmlIn]

```

```

' hora: ' hora[$xmlIn] { $hora_inicio = $hora.seconds
    -3600;}
' local: ' STRING { $xmlIn.append("<local>"+$STRING.text+"</
    local>\n");}
' duracao-maxima: ' duracao[$xmlIn] {
    $tempo_max_aud = $duracao
        .seconds-3600;
    $xmlOut = $xmlIn;
    }
;

data [StringBulder xmlIn]
returns[StringBulder xmlOut]:
dia=INT'/'mes=INT'/'ano=INT {
    $xmlIn.append("<data>"+$dia.int+"/"+"
        $mes.int+"/"+"$ano.int+"</data>\n")
    ;
    $xmlOut = $xmlIn;
}

;

hora [StringBulder xmlIn]
returns[int seconds, StringBulder xmlOut]:
horas=INT': 'minutos=INT {
    $seconds = ($horas.int*60*60 + $minutos.
        int*60);
    $xmlIn.append("<hora>"+$horas.int+": "+
        $minutos.int+"</hora>\n");
    $xmlOut = $xmlIn;
}

;

duracao [StringBulder xmlIn]
returns[int seconds, StringBulder xmlOut]:
horas=INT': 'minutos=INT {
    $seconds = ($horas.int*60*60 + $minutos.
        int*60);
    $xmlIn.append("<duracao>"+$horas.int+": "+
        $minutos.int+"</duracao>\n");
    $xmlOut = $xmlIn;
}

;

atuacoes [StringBulder xmlIn, long hora_inicio, int
    tempo_max_aud]
returns[long tempo, StringBulder xmlOut]

```

```

:      'atuacoes:' atuacao[( $tempo-3600), $xmlIn,
      $hora_inicio, $tempo_max_aud] {
                                $tempo = $atuacao
                                .tempo;
                                $xmlOut =
                                $atuacao.
                                xmlOut;
                                } '#',
(atuacao[$tempo, $xmlOut, $hora_inicio, $tempo_max_aud]
{
                                $tempo = $atuacao.tempor;
                                $xmlOut = $atuacao.xmlOut;
                                } '#')*
;

atuacao      [long tempoIn, StringBuilder xmlIn, long
      hora_inicio, int tempo_max_aud]
returns[long tempo, StringBuilder xmlOut]
:      { $xmlIn.append("\n<atuacao_tipo=\"grupo\">\n");}
grupo[$tempoIn, $xmlIn, $hora_inicio, $tempo_max_aud] {
                                $tempo=$grupo.tempor;
                                $xmlOut = $grupo.xmlOut;
                                $xmlOut.append("</atuacao>");
                                }
|      { $xmlIn.append("\n<atuacao_tipo=\"solo\">");}
solo[$tempoIn, $xmlIn, $hora_inicio, $tempo_max_aud] {
                                $tempo=$solo.tempor;
                                $xmlOut = $solo.xmlOut;
                                $xmlOut.append("</atuacao>");
                                }
;

grupo      [long tempoIn, StringBuilder xmlIn, long
      hora_inicio, int tempo_max_aud]
returns[long tempo, StringBuilder xmlOut]
:      'grupo:' STRING {
                                $xmlIn.append("\n<nome_grupo>"+$STRING.text+"
                                </nome_grupo>");
                                Time hora_ini = new Time( ($hora_inicio +
                                $tempoIn ) * 1000);
                                $xmlIn.append("\n<hora_inicio>"+hora_ini.
                                toString()+"</hora_inicio>");
                                }
'elementos:' {
                                $xmlIn.append("\n<musicos>\n");
                                }

```

```

        musicos [$xmlIn]
    {
        $xmlIn = $musicos.xmlOut;
        $xmlIn.append("\n</musicos>\n");
    }
'obras:' {
    $xmlIn.append("\n<obras>\n");
}
obras [$tempoIn, $xmlIn, $tempo_max_aud]
{
    $xmlIn = $obras.xmlOut;
    $xmlIn.append("\n</obras>\n");
    $tempo = $obras.tempo;
    $xmlOut = $xmlIn;
}
;
solo [long tempoIn, StringBuilder xmlIn, long
    hora_inicio, int tempo_max_aud]
returns [long tempo, StringBuilder xmlOut]
: {
    Time hora_ini = new Time( ($hora_inicio + $tempoIn) *
        1000);
    $xmlIn.append("\n<hora_inicio>"+hora_ini.toString()+"</
        hora_inicio>");
}
'solo:' {
    $xmlIn.append("\n<musicos>\n");
}
    musico [$xmlIn]
{
    $xmlIn=$musico.xmlOut;
    $xmlIn.append("\n</musicos>\n");
}
'obras:' {
    $xmlIn.append("\n<obras>\n");
}
obras [$tempoIn, $xmlIn, $tempo_max_aud]
{
    $xmlIn=$obras.xmlOut;
    $xmlIn.append("\n</obras>\n");
    $tempo = $obras.tempo;
    $xmlOut=$xmlIn;
}
;
obras [long tempoIn, StringBuilder xmlIn, int
    tempo_max_aud]

```



```

returns [long tempo, StringBuilder xmlOut]
: idObra {
    try{
        String sql = "SELECT_*_FROM_obra_WHERE_id='"+
            $idObra.id+" '";
        ResultSet rs = (ResultSet) stmt.executeQuery(sql)
        ;
        if (rs.next()) {
            $tempo += rs.getTime("duracao").toLocalTime()
                .toSecondOfDay();
            $tempo += $tempoIn;
            if ($tempo > $tempo_max_aud) {
                Time time = new Time($tempo * 1000);
                System.out.print("line_"+$idObra.
                    linha+"_-tempo_max_ultrapassado"
                );
                System.out.println("_"+$idObra.id+"_
                    tem_duracao:_" + time.toString());
                flag=1;
            } else {
                $xmlIn.append("\n<obra_id=\""+$idObra.id+
                    "\">");
                $xmlIn.append("<nome>"+rs.getString("nome
                    ")+"</nome>");
                $xmlIn.append("<duracao>"+rs.getString("
                    duracao")+"</duracao>");
                $xmlIn.append("</obra>");
            }
        } else {
            System.out.print("line_"+$idObra.linha);
            System.out.println("_obra:_" + $idObra.id + "_nao
                _existe");
            $xmlIn.append("<obra></obra>");
            flag=1;
        }
        rs.close();
    } catch (SQLException se) {
        //se.printStackTrace();
    }
    $xmlOut = $xmlIn;
}
(, 'idObra {
    try{
        String sql = "SELECT_*_FROM_obra_WHERE_id='"+
            $idObra.id+" '";

```

```

ResultSet rs = (ResultSet) stmt.executeQuery(
    sql);
if (rs.next()) {
    $tempo += rs.getTime("duracao").
        toLocalTime().toSecondOfDay();
    if ($tempo > $tempo_max_aud) {
        Time time = new Time($tempo * 1000);
        System.out.print("line_" + $idObra.
            linha + "_" + tempo_max_ultrapassado "
        );
        System.out.println("_" + $idObra.id + "_"
            tem_duracao:_" + time.toString());
        flag = 1;
    } else {
        $xmlIn.append("\n<obra_id=\"" + $idObra
            .id + "\">");
        $xmlIn.append("<nome>" + rs.getString("
            nome") + "</nome>");
        $xmlIn.append("<duracao>" + rs.
            getString("duracao") + "</duracao>")
            ;
        $xmlIn.append("</obra>");
    }
} else {
    System.out.print("line_" + $idObra.linha);
    System.out.println("_obra:_" + $idObra.id + "
        _nao_existe");
    $xmlIn.append("<obra></obra>");
    flag = 1;
}
rs.close();
} catch (SQLException se) {
    //se.printStackTrace();
}
$xmlOut = $xmlIn;
}
)*
;

musicos [StringBuilder xmlIn]
returns [StringBuilder xmlOut]
:   musico [$xmlIn] { $xmlOut = $musico.xmlOut; }
(musico [$xmlOut] { $xmlOut = $musico.xmlOut; }) *
;

musico [StringBuilder xmlIn]

```

```

returns[StringBuilder xmlOut]
:      idAluno ', ' idInstrumento {
$xmlIn.append("\n<musico_tipo=\\"aluno\\"_id=\\""+$idAluno.
      id+"\\">");
try{
    String sql = "SELECT_*_FROM_aluno_WHERE_id='"+
        $idAluno.id+"'";
    ResultSet rs = (ResultSet) stmt.executeQuery(sql);
    if(rs.next()){
        String nome_aluno = rs.getString("nome");
        sql = "SELECT_*_FROM_instrumento_WHERE_id='"+
            $idInstrumento.id+"'";
        rs = (ResultSet) stmt.executeQuery(sql);
        if(rs.next()){
            //aluno sabe tocar instrumento?
            String designacao_instrumento = rs.getString(
                "designacao");
            sql = "SELECT_*_FROM_aluno_instrumento_WHERE_
                id_aluno='"+$idAluno.id+"'_AND_
                id_instrumento='"+$idInstrumento.id+"'";
            rs = (ResultSet) stmt.executeQuery(sql);
            if(rs.next()){
                //OK
                $xmlIn.append("<nome>"+nome_aluno+"</nome>");
                $xmlIn.append("<instrumento_id=\\""+
                    $idInstrumento.id+"\\">"+
                    designacao_instrumento+"</instrumento>");
            }else{
                System.out.print("line_"+$idAluno.linha);
                System.out.print("_("aluno:_"+$idAluno.id+
                    "_-_"+ nome_aluno+"")");
                System.out.println("_nao_tem_habilidades_
                    _para_tocar_("+ $idInstrumento.id+"_-_"
                    + designacao_instrumento+""));

                sql = "SELECT_*_FROM_aluno_instrumento_
                    INNER_JOIN_instrumento_ON_
                    aluno_instrumento.id_instrumento=
                    instrumento.id_WHERE_aluno_instrumento_
                    .id_aluno='"+$idAluno.id+"'";
                rs = (ResultSet) stmt.executeQuery(sql);
                System.out.print("O_Aluno:_ "+ $idAluno.id
                    );
            }
        }
    }
}

```

```

        System.out.println("_tem_habilitacoes_em:");
    };
    while(rs.next()){
        System.out.println("_-_" + rs.
            getString("id")+"_"+rs.getString("
            designacao"));
    }
    flag=1;
}
} else{
    System.out.print("line_"+$idInstrumento.linha
    );
    System.out.println("_instrumento:_" +
        $idInstrumento.id+"_nao_existe");
    sql = "SELECT_*_FROM_aluno_instrumento_INNER_
        JOIN_instrumento_ON_aluno_instrumento.
        id_instrumento=instrumento.id_WHERE_
        aluno_instrumento.id_aluno='"+$idAluno.id+
        "'";
    rs = (ResultSet) stmt.executeQuery(sql);
    System.out.print("O_Aluno:_" + $idAluno.id);
    System.out.println("_tem_habilitacoes_em:");
    while(rs.next()){
        System.out.println("_-_" + rs.getString("
            id")+"_"+rs.getString("designacao"));
    }
    flag=1;
}
} else{
    System.out.print("line_"+$idAluno.linha);
    System.out.println("_aluno:_" + $idAluno.id+"_nao_
        existe");
    flag=1;
}
rs.close();
} catch(SQLException se){
    //se.printStackTrace();
}
$xmlIn.append("</musico>");
$xmlOut = $xmlIn;
}
| idProf ', ' idInstrumento {
$xmlIn.append("\n<musico_tipo=\"professor\"_id=\""+
    $idProf.id+"\">");
try{

```

```

String sql = "SELECT_*_FROM_professor_WHERE_id='"+
    $idProf.id+"'";
ResultSet rs = (ResultSet) stmt.executeQuery(sql);
if(rs.next()){
    String nome_prof = rs.getString("nome");
    sql = "SELECT_*_FROM_instrumento_WHERE_id='"+
        $idInstrumento.id+"'";
    rs = (ResultSet) stmt.executeQuery(sql);
    if(rs.next()){
        //professor sabe tocar instrumento?
        String designacao_instrumento = rs.getString(
            "designacao");
        sql = "SELECT_*_FROM_professor_INNER_JOIN_
            professor_habilitacao_ON_professor.id=
            professor_habilitacao.id_professor_INNER_
            JOIN_habilitacao_ON_professor_habilitacao.
            id_habilitacao=habilitacao.id_WHERE_
            professor.id_='"+$idProf.id+"'AND_
            id_instrumento='"+$idInstrumento.id+"'";
        rs = (ResultSet) stmt.executeQuery(sql);
        if(rs.next()){
            //OK
            $xmlIn.append("<nome>"+nome_prof+"</nome>
                ");
            $xmlIn.append("<instrumento_id=\\\""+
                $idInstrumento.id+"\\\">"+
                designacao_instrumento+"</instrumento>
                ");
        }else{
            System.out.print("line_"+$idProf.linha);
            System.out.print("_ (professor:_" +$idProf.
                id+"_-_" + nome_prof+" )" );
            System.out.println("_nao_tem_habilitacoes
                _para_tocar("+ $idInstrumento.id+"_-_" +
                designacao_instrumento+" )" );
            flag=1;
        }
    }
}
}else{
    System.out.print("line_"+$idInstrumento.linha
        );
    System.out.println("_instrumento:_" +
        $idInstrumento.id+"_nao_existe");
    flag=1;
}
}
}else{
    System.out.print("line_"+$idProf.linha);

```

```

        System.out.println("_professor:"+$idProf.id+"_
        nao_existe");
        flag=1;
    }
    rs.close();
} catch (SQLException se) {
    //se.printStackTrace();
}
$xmlIn.append("</musico>");
$xmlOut = $xmlIn;
}
;

```

```

/*————— Lexer —————*/

```

COMMENT

```

: ( '//' ~[\r\n]* '\r'? '\n'
  | '/*' .*? '*/'
  | '\n'
  ) -> skip
;

```

```

ID_ALUNO : ('A') ('0'..'9')*
;

```

```

ID_PROFESSOR : ('P') ('0'..'9')*
;

```

```

ID_OBRA : ('O') ('0'..'9')*
;

```

```

ID_INSTRUMENTO : ('I') ('0'..'9')*
;

```

```

ID : ('a'..'z'|'A'..'Z'|'_') ('a'..'z'|'A'..'Z'|'0'..'9'|'_')*
;

```

```

INT : '0'..'9'+('.' ('0'..'9'))+)?
;

```

```

WS : [ \t\r\n ] -> skip
;

```

```

STRING
:   ' " ' ( ESC_SEQ | ~( ' " ' ) * ' " '
;

fragment
ESC_SEQ
:   '\\ ' ( 'b' | 't' | 'n' | 'f' | 'r' | '\\ " ' | '\\ ' | '\\ \\ ' )
|   UNICODE_ESC
|   OCTAL_ESC
;

fragment
OCTAL_ESC
:   '\\ ' ( '0' .. '3' ) ( '0' .. '7' ) ( '0' .. '7' )
|   '\\ ' ( '0' .. '7' ) ( '0' .. '7' )
|   '\\ ' ( '0' .. '7' )
;

fragment
UNICODE_ESC
:   '\\ ' 'u' HEX_DIGIT HEX_DIGIT HEX_DIGIT HEX_DIGIT
;

fragment
HEX_DIGIT : ( '0' .. '9' | 'a' .. 'f' | 'A' .. 'F' )
;

```