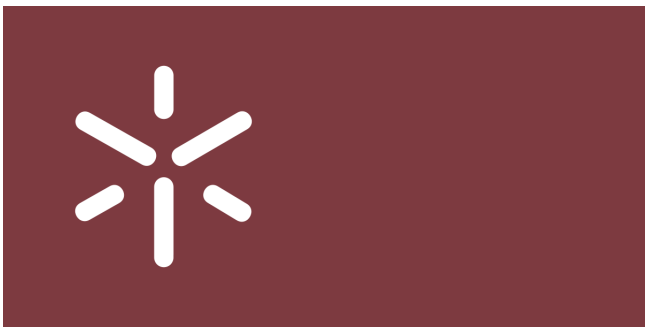

Universidade do Minho

Mestrado em Engenharia Informática

Processamento e Representação de Informação

iBanda

Trabalho prático

Autores:
Frederico Pinto - 73639
Rui Vieira - 74658

1 de Fevereiro de 2019

Conteúdo

1. *Introdução*

Foi-nos proposto na Unidade Curricular, de Processamento de Representação de Informação, o desenvolvimento de uma aplicação Web. Aplicação essa que deve funcionar como um repositório digital de obras musicais com suas respetivas partituras, referentes a uma banda filarmónica. Dispondo também de uma calendário de eventos assim como uma páginas de notícias. A implementação desta aplicação foi baseada no modelo OAIS.

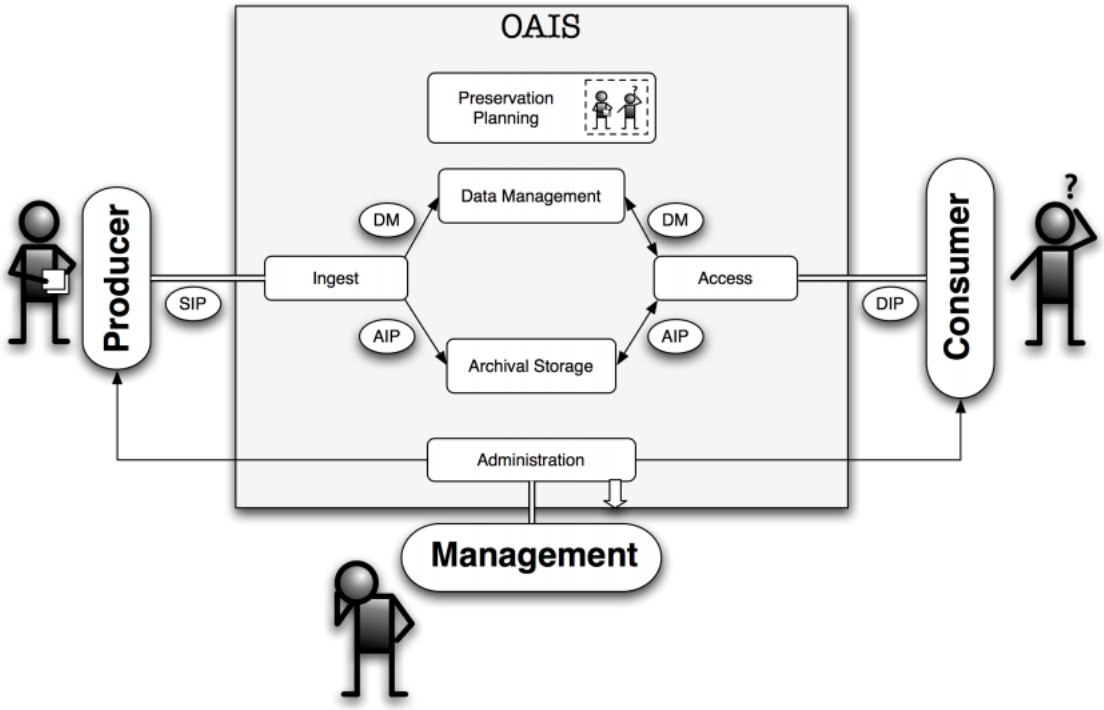

Figura 1.1: Modelo de referência OAIS.

2. *Opções de Desenvolvimento Inicial*

Na fase inicial de desenvolvimento foram tomadas várias decisões importantes referentes à implementação, tendo o grupo optado pelo desenvolvimento do *front-end* em *Bootstrap* possibilitando a aprendizagem de ferramentas não lecionadas. Outra das decisões importantes a definição dos diferentes tipos de utilizadores assim como as suas permissões.

2.1 Tipos de Utilizadores

- **Tipo 3**- Utilizador registado, capaz apenas de consultar conteúdo. Corresponde ao Consumer no modelo OAIS.
- **Tipo 2**- Utilizador registado, capaz de consultar, adicionar e eliminar conteúdo. Corresponde ao Producer no modelo OAIS.
- **Tipo 1**- Corresponde ao administrador que partilha todas as funcionalidades do utilizador de tipo 2, gerindo ainda os utilizadores e podendo consultar estatísticas sobre as obras mais visualizadas e as partituras com mais downloads. Corresponde ao Managment no modelo OAIS.

3. *Base de dados*

Nesta aplicação é usada uma base de dados em *MongoDB*, definindo modelos diferentes guardados coleções separadas. Cada uma com o seu respetivo controlador, mantendo a integridade dos dados.

3.1 Caraterização dos Modelos

- USER - Representa o utilizador, composto por:
  - name : Nome do utilizador.
  - email : E-mail do utilizador.
  - password : Password do utilizador.
  - userType : Tipo de utilizador (abordado no capitulo anterior).

- NEWS - Representa uma noticia e é composto por:
  - meta
    - data : Data da noticia
    - autor : Autor da Noticia
  - titulo : Titulo
  - corpo : Corpo
- CALENDAR - Representa o evento e é composto por:
  - nome : Nome do evento.
  - data : Data do evento.
  - tipo : Tipo do evento
  - local : Local do evento.
  - bilhetes : Bilhetes do evento.
- OBRAS - Representa as obras musicais e é composto por:
  - id : Identificador da obra.
  - nVisualizacao : Numero de visualizações da obra.
  - titulo : Titulo da obras.
  - tipo : Tipo da obra.
  - compositor : Compositor da obra.
  - arranjo: Arranjo da obra.
  - instrumentos: Lista de instrumentos da obras.
    - nome : Nome do instrumento.
    - nDownloads : Numero de downloads do instrumento.
    - partitura : Partitura referente ao instrumento.
      - path : Caminho para o ficheiro da partitura.
      - voz : Voz da partitura.
      - clave : Clave da partitura.
      - afinacao : Afinação da partitura.

4. *Autenticação*

Para a autenticação utilizamos a estratégia de sessões, usando o *Passport*, lecionada na Unidade Curricular. Depois de iniciada a sessão sempre que é feito um pedido verificamos se está autenticado com o método *isAuthenticated()*. Contudo, como temos diferentes tipos de utilizador, também não podemos permitir que, por exemplo, o utilizador do tipo 3 aceda às capacidades do administrador que também é um utilizador autenticado. Ou seja, para resolver tal problema verificamos qual o tipo de utilizador ao receber o pedido com no *req.user*.

5. *Rotas*

Nesta secção decidimos criar um roteador da API de dados para cada um dos modelos anteriormente referidos. E de forma a que os diferentes utilizadores tenham apenas acesso às funcionalidades que lhes dizem respeito, optamos pela criação de roteadores específicos para cada tipo de utilizador, que utiliza o *front-end* adequado. Limitando desta forma as funcionalidades disponíveis a cada tipo de utilizador.

6. *Features adicionais*

De forma a simplificar a manutenção da aplicação optamos por guardar todos os *logs* num ficheiro. Possibilitando que em caso de falha seja possível ter um registo do que ocorreu.

Para além disso também usufruímos do *JQuery* para trabalhar do lado cliente e assim poupar esforço por parte do servidor.

7. *Gramática*

Como combinado, na nossa implementação a parte de inserção de noticias na aplicação, iria utilizar uma gramática para fazer o reconhecimento do *input* e com isto gerar *json* que posteriormente seria inserido na base de dados e tratado para mostrar ao consumidor. Contudo, nem tudo correu como planeado e não conseguimos implementar a gramática na nossa aplicação em *JavaScript*. Apesar disso, apresentamos de seguida duas gramáticas e as respetivas linguagens que criamos na tentativa de conseguir fazer a ligação.

7.1 ANTLR

7.1.1 Gramática

```
noticias: (noticia)*
        ;

noticia returns[JSONObject obj]
         @init{
             $noticia.obj = new JSONObject();
         }
         @after{}
         :meta[$noticia.obj] titulo[$noticia.obj]  corpo[$noticia.obj]
          ;

titulo [JSONObject obj]
     :'TITULO:' '{'TEXTO'}' { $titulo.obj.put("titulo",$TEXTO.text);}
     ;

corpo [JSONObject obj]
     : 'CORPO:' '{'TEXTO'}' { $corpo.obj.put("corpo",$TEXTO.text);}
     ;

meta [JSONObject obj] returns [JSONObject metaObj]
     @init{
         $meta.metaObj = new JSONObject();
     }
     : data[$meta.metaObj] autor[$meta.metaObj] (tema[$meta.metaObj])? {obj.put("meta", metaObj); }
     ;

autor [JSONObject metaObj]
     :'AUTOR:' TEXTO {$autor.metaObj.put("autor",$TEXTO.text);}
     ;

tema [JSONObject metaObj]
     :'TEMA:' TEXTO {$tema.metaObj.put("tema",$TEXTO.text);}
     ;

data [JSONObject metaObj]
     :'DATA:' DATA {$data.metaObj.put("data",$DATA.text);}
     ;
```

Figura 7.1: Gramática ANTLR.

7.1.2 Linguagem

A gramática apresentada anteriormente tem a seguinte linguagem:

DATA: 29/01/2019
AUTOR: "Rui Vieira"
TEMA: "Benfica"
TITULO: {"ÁGUIAS DESTACAM FORMAÇÃO: O FUTURO DO BENFICA ESTÁ ASSEGURADO"}
CORPO: {"Através da plataforma News Benfica, o clube da Luz destacou a baixa média de idades da convocatória feita por Bruno Lage para a receção ao Boavista, na noite desta terça-feira.

Treze destes jogadores têm 25 ou menos anos e há apenas um caso acima dos 30 (o capitão Jardel). Não é apenas em Portugal que o exemplo do Benfica se distingue. Até nos principais campeonatos europeus é difícil encontrar hoje uma equipa com raízes tão internas e que, ao mesmo tempo, permita projetar um futuro tão risonho."}

7.2 PEG JS

Após testarmos com o ANTLR e termos falhado, decidimos tentar utilizar ferramentos do *NodeJS* para nos ajudar a obter o nosso propósito, então optamos pelo *PEG JS*. De seguida apresentamos a gramática criada em *Javascript* utilizando essa ferramenta.

7.2.1 Gramática

```
noticia = m:meta t:titulo c:corpo {return {meta:m, titulo:t, corpo:c}}

meta = data:data autor:autor {return {data, autor}}

data = "DATA:" _ d:word _ ':' _ {return d}

autor = "AUTOR:" _ a:(word _)+ ':' _ {return a.toString().replace(/,/g, "")}

titulo = "TITULO:" _ t:(word _)+ ':' _ {return t.toString().replace(/,/g, "")}

corpo = "CORPO:" _ c:(word _)+ ':' _ {return c.toString().replace(/,/g, "")}

word = l:letter+ {return l.join("")}

letter = [a-zA-ZáéíóúÁÉÍÓÚÃãÕõâêôÂÊÔÀÈÌÒÙàèìòùÇç0-9_/]

_ "whitespace"
  = [ \t\n\r]*
```

Figura 7.2: Gramática PEG JS.

7.2.2 Linguagem

A gramática apresentada anteriormente tem a seguinte linguagem:

DATA: 21/01/1996:
AUTOR: Rui Reino:
TITULO: Fuga do Zoo:
CORPO: Uma vez uma raposa fugiu do zoo:

8. *Conclusão*

A implementação deste projeto foi de extrema importância para a nossa formação como futuros engenheiros, pois deu-nos bases sólidas para podermos evoluir.

Relativamente à aplicação *web*, achamos que cumprimos os requisitos propostos no enunciado e estamos felizes com a nossa prestação. Contudo, nem tudo foi bom, atravessamos sérias dificuldades com a implementação da gramática na aplicação, e o fato de não termos conseguido fazer essa implementação desmotivou-nos um pouco mas todo o esforço que colocamos na tentativa de resolução, valeu a pena esse esforço trouxe-nos ainda mais conhecimento.