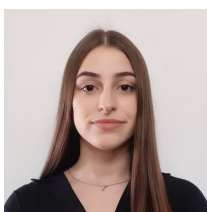
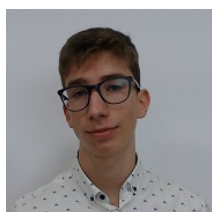




Universidade do Minho
Mestrado Integrado em Engenharia Informática
4ºano - 1º Semestre

Desenvolvimento de Aplicações Web

Plataforma de Gestão e Disponibilização de Recursos Educativos



Capa/42814.png

a84197 – João Pedro Araújo Parente
pg44412 - Ana Margarida da Rocha Ferreira
PG42814 – Raimundo Nonato Barros NEto

15 de Janeiro de 2021

Conteúdo

1	Introdução	3
2	Conceitos	4
3	Persistência de dados	4
4	Utilizadores	5
4.1	Níveis	6
4.1.1	Consumidor - nível 0	6
4.1.2	Produtor - nível 1	6
4.1.3	Administrador - nível 2	6
5	Tipos	7
6	Recursos	8
6.1	Meta-data	8
6.2	Política de armazenamento	9
7	Pacote de SIP e DIP	9
7.1	Manifesto	10
8	Exportação/Importação	11
8.1	Utilizadores e Tipos	11
8.2	Recursos	11
9	Autenticação	12
10	Conclusão	13
11	Referências	14

Lista de Figuras

1	Exemplo de um utilizador.	5
2	Exemplo de um tipo.	7
3	Exemplo de um recurso na base de dados.	9
4	Exemplo do conteúdo dentro de um SIP/DIP.	9
5	Exemplo do conteúdo dentro de um manifesto.	10
6	Exemplo de um utilizador em formato 'csv'.	11
7	Exemplo de um tipo em formato 'csv'	11
8	Exemplo do ficheiro zip que constitue um export de recursos. .	11
9	Exemplo do 'csv' de recursos	11

1 Introdução

O presente trabalho prático foi realizado no âmbito da unidade curricular de Desenvolvimento de Aplicações Web e consistiu no desenvolvimento de uma plataforma de gestão e disponibilização de recursos educativos, ou seja um servidor aplicacional que recebe pedidos em determinadas rotas e fornece um html adequado, permitindo um utilizador registar-se/fazer login/uploads/downloads/publicações tudo a partir do seu browser.

2 Conceitos

Todos os conceitos presentes no trabalho:

- **Ficheiro:** arquivo de computador que tem associado a si um tipo.
- **Tipo de ficheiro:** extensão de um ficheiro e meta-data adicional que pode conter
- **Utilizador:** representação de um cliente no sistema.
- **Recurso:** conjunto de ficheiros.
- **Pacote SIP:** formato que um recurso deve seguir para ser inserido no sistema.
- **Pacote DIP:** formato que o sistema usa para partilhar a informação com os clientes.
- **Publicação:** referente a um recurso do sistema que contém um comentário feito pelo criador da publicação e aceita outros comentários dos clientes.

3 Persistência de dados

A persistência de dados de toda a meta-informação foi feita tirando partido de um sistema de base dados MongoDB e os ficheiros foram guardados numa pasta tendo o sua referência na base de dados. No MongoDB foram criadas três coleções: Recurso,Tipos,Utilizadores.

4 Utilizadores

Um utilizador tem os seguintes campos, que vão ser guardados todos na base de dados:

- **_id:** username do utilizador.
- **password:** password do utilizador.
- **dataRegisto:** data em que o utilizador foi registado.
- **dataUltimoAcesso:** ultima vez que o utilizador esteve conectado ao sistema.
- **nivel:** nivel de acesso do utilizador no sistema.
- **email:** email do utilizador.
- **nome:** nome do utilizador.
- **filiacao.estudante:** se o utilizador é estudante ou não.
- **filiacao.docente:** se o utilizador é docente ou não.
- **filiacao.departamento:** departamento associado ao utilizador.
- **filiacao.curso:** curso associado ao utilizador.

```
{
  "_id": "joao123",
  "dataRegisto": {
    "$date": "2021-01-26T09:22:00Z"
  },
  "dataUltimoAcesso": {
    "$date": "2021-01-27T13:40:00Z"
  },
  "nivel": 0,
  "nome": "João Pedro Afonso",
  "email": "jpbp.pfdrp@gmail.com",
  "filiacao": {
    "estudante": true,
    "docente": false,
    "curso": "Miei",
    "departamento": "Informática"
  },
  "password": "123"
}
```

Figura 1: Exemplo de um utilizador.

4.1 Níveis

Um utilizador tem um nível associado a ele e cada nível atribui diferentes permissões.

4.1.1 Consumidor - nível 0

Um consumidor apenas consegue visualizar e download de conteúdo, editar a informação referente a ele mesmo, criar publicações sobre um recurso, editar/remover as publicações feitas pelo mesmo e comentar publicações.

4.1.2 Produtor - nível 1

Um produtor têm as permissões de um consumidor e consegue ainda dar upload a conteúdo e editar o conteúdo criado pelo mesmo.

4.1.3 Administrador - nível 2

Um administrador têm as permissões dos níveis anteriores e consegue ainda editar e remover todo o conteúdo além de poder editar/remover utilizadores e fazer import/export dos recursos.

5 Tipos

Um tipo tem os seguintes campos, que vão ser guardados todos na base de dados:

- **_id:** que vai ser o nome e identificador deste tipo.
- **parametros:** que é um array de meta-data referentes a este tipo onde cada elemento do array tem duas propriedades: "nome_param" que vai especificar o nome do parâmetro e o "tipo_param" que vai dizer se este parâmetro aceita "String" ou "Number"

```
{
  "_id": "txt",
  "parametros": [
    {
      "nome_param": "numero_linhas",
      "tipo_param": "Number"
    },
    {
      "nome_param": "sobre",
      "tipo_param": "String"
    }
  ]
}
```

Figura 2: Exemplo de um tipo.

6 Recursos

Um recurso é constituído por 2 partes: a meta-data referente ao recurso que vai ser guardada na base de dados e por um conjunto de ficheiros que vai ser guardados numa pasta do servidor.

6.1 Meta-data

Um recurso tem os seguintes campos:

- **_id:** identificador do recurso.
- **titulo:** titulo do recurso.
- **subtitulo:** subtitulo do recurso.
- **dataCriação:** data em que o recurso foi criado.
- **dataRegisto:** data em que o recurso foi registado no sistema.
- **visibilidade:** se o recurso é público(toda a gente o consegue ver) ou privado(apenas quem deu upload)
- **likes:** lista do identificador de utilizadores que gostaram do recurso.
- **hashtags:** lista de hashtags associados ao recurso.
- **dataCriação:** data em que o recurso foi criado.
- **posts:** publicações referentes ao recurso.
- **manifesto:** estrutura do recurso(mais abaixo entramos mais em detalhe).
- **path:** onde os ficheiros do recurso estão a ser guardados.

```

{
  "id": "2021-01-27T10:38:23Z-Projectos",
  "subtitulo": "3ae_Ano-1ae_Semestre-Sistemas_Distribuidos-Projectos",
  "dataCriacao": {
    "date": "2021-01-27T10:38:23Z"
  },
  "dataRegisto": {
    "date": "2021-01-27T10:38:23Z"
  },
  "visibilidade": "publico",
  "likes": [],
  "hashtags": ["Sistemas_Distribuidos"],
  "titulo": "Projectos",
  "produtor": "admin",
  "posts": [],
  "manifesto": "{\\\"ficheiros\\\":[],\\\"pasta_rec\\\":[{\\\"nome\\\":\\\"2015-2016\\\",\\\"pasta\\\":\\\"fileStore/Sistemas_Distribuidos/2021-01-27T10:38:23Z-Projectos\\\"}]}",
  "path": "fileStore/Sistemas_Distribuidos/2021-01-27T10:38:23Z-Projectos"
}

```

Figura 3: Exemplo de um recurso na base de dados.

6.2 Política de armazenamento

A política de armazenamento é o que dado um recurso nos diz aonde ele deve ser guardado. Optamos por uma política de armazenamento simples que foi dividir os recursos por pastas, onde cada pasta é o primeiro hashtag do recurso, evitando assim em maior parte dos casos alcançar o limite de ficheiros por pasta, acreditando que os recursos têm hashtags mais ao menos uniformemente distribuído.

7 Pacote de SIP e DIP

Como simplificação consideramos o pacote de submissão de informação(SIP) igual ao pacote de disseminação de informação(DIP). A formato escolhido para estes pacotes foi um zip, constituído por um ficheiro que têm se de chamar manifesto.json onde vai ter a estrutura do do recurso e uma pasta chamada "data" que vai ter o recurso em si.

```

2021-01-27T10:35:08Z-Testes_e_exames_de_LESI/
├── data
│   ├── Exames_LESI.pdf
│   └── Resolucoes
│       ├── Exame_Primeira_Chamada_2006-07_LESI_Resolucao.pdf
│       └── Exame_Segunda_Chamada_2006-07_LESI_Resolucao.pdf
└── manifesto.json

```

Figura 4: Exemplo do conteúdo dentro de um SIP/DIP.

7.1 Manifesto

O manifesto é um ficheiro chamado "manifesto.json" que dentro tem uma string em json que tem de ter duas propriedades:

- **ficheiros:** lista de ficheiros (nome,tipo,meta)
- **pasta_rec:** que vai representar uma pasta, vai ter um nome e uma propriedade "pasta" que vai ser a recursividade.

```
{
  "ficheiros": [{
    "nome": "Exames_LESI.pdf",
    "tipo": "pdf",
    "meta": []
  }],
  "pasta_rec": [{
    "nome": "Resolucoes",
    "pasta": {
      "ficheiros": [{
        "nome": "Exame_Primeira_Chamada_2006-07_LESI_Resolucao.pdf",
        "tipo": "pdf",
        "meta": []
      }, {
        "nome": "Exame_Segunda_Chamada_2006-07_LESI_Resolucao.pdf",
        "tipo": "pdf",
        "meta": []
      }],
      "pasta_rec": []
    }
  ]
}
```

Figura 5: Exemplo do conteúdo dentro de um manifesto.

8 Exportação/Importação

A exportação e importação vão ser operações que poderão ser feitas pelos administradores e ambas recebem/devolvem o mesmo formato e optamos por ficheiros '.csv'

8.1 Utilizadores e Tipos

Como a informação dos utilizadores ou dos tipos esta toda na base de dados apenas convertemos ou desfazemos a conversão do modelo de dados para 'csv'.

```
[_id,nome,email,[estudante,docente,curso,departamento],dataRegisto,dataUltimoAcesso,password,nivel  
joao123,João Pedro Afonso,jpbp.pfdrp@gmail.com,[true;false;Miei;Informática],2021-01-26T09:22:00.000Z,2021-01-26T09:22:00.000Z,123,0
```

Figura 6: Exemplo de um utilizador em formato 'csv'.

```
_id,[param]  
js,[numero_linhas#Number;bibliotecas_adicionais#String]
```

Figura 7: Exemplo de um tipo em formato 'csv'

8.2 Recursos

Como os recursos têm parte da informação na base de dados e numa pasta, a parte da base de dados é igual aos tipos e ficheiros e com isso cria-se um 'csv' com a outra parte é zipada para um zip com o '_id' do recurso após termos as duas partes zipamos as duas formando assim um zip com um 'csv' com toda a meta-data e um zip por recurso onde estão lá os ficheiros de cada recurso.

```
export recursos0.5980619374700369 2021-01-27T15 14  
└─ 2021-01-27T1514-0.673176074498731.zip  
export:recursos0.5980619374700369:2021-01-27T15:14.csv
```

Figura 8: Exemplo do ficheiro zip que constitui um export de recursos.

```
_id,titulo,subtitulo,dataCriacao,dataRegisto,produtor,visibilidade,[likes],[hashtags],[post]  
2021-01-27T1514-0.673176074498731,Lesi exame,,2021-01-27T15:14:00.000Z,2021-01-27T15:08:00.000Z,admin,publico,[],[Algoritmos_e_Complexidade],[]
```

Figura 9: Exemplo do 'csv' de recursos

9 Autenticação

Autenticação é algo bastante importante neste projeto para distinguir utilizadores pois dependendo do nível dos utilizadores as acções que permitimos fazer são diferentes, vamos usar sessões no servidor aplicacional e cookies nos clientes, como apenas temos um servidor que vai tratar de autenticação e responder pedidos preferimos esta abordagem a usar JWT tokens pois ser mais simples e realizar todas as funcionalidades que queremos.

10 Conclusão

O presente relatório descreveu de forma sucinta, o trabalho realizado sendo que importa referir os seguintes aspetos.

Foi usada para unzip de ficheiros a biblioteca unzip-stream que como não traz todos os ficheiros para memória de uma só vez consegue lidar com zips de grande tamanho, o que foi uma das nossas preocupações pois existiam o objetivo de criar a funcionalidade de import/export, não foi possível encontrar nenhuma biblioteca para fazer o zip de ficheiros em stream, apenas encontramos bibliotecas com limitações de tamanho, como foi o caso da admin-zip que foi uma biblioteca que usamos em grande parte do desenvolvimento do trabalho mas imponha uma limitação de 2gb, a solução que encontramos foi usar partido da ferramenta "zip" que está instalada normalmente em todos os sistemas unix que nos permite lidar com ficheiros de tamanho maior, impondo assim uma dependência externa ao node, mas que achamos pertinente pois considerarmos, como referido anteriormente as operações de import e export bastante importantes.

Sentimos que a realização deste trabalho prático permitiu desenvolver competências com a ferramenta *express* e com *nodejs*, competências que poderão ser cruciais nas nossas futuras carreiras.

11 Referências

- [1] BagIt <https://tools.ietf.org/id/draft-kunze-bagit-16.html>