



UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

Gestor e Disponibilização de Recursos

João Gomes
A82238

João Gomes
A82428

Luís Ferreira
A76936

Sofia Teixeira
A80624

(7 de fevereiro de 2021)

Conteúdo

1	Introdução	3
2	Arquitetura da solução	4
2.1	Micro Serviços	4
2.2	Autenticação	5
2.2.1	LocalStrategy	5
2.2.2	FacebookStrategy	5
2.2.3	GoogleStrategy	5
2.3	Persistência de Dados	5
2.3.1	Users	6
2.3.2	Recursos	6
2.3.3	Comentários	7
3	Funcionalidades	7
3.1	Consumidor	7
3.2	Produtor	8
3.3	Admin	8
4	Conclusão	9

1 Introdução

Neste projeto pretende-se desenvolver uma WebApp com o intuito de fazer a Gestão e Disponibilização de Recursos Educativos.

Tem-se por objetivo que a versão final da App forneça as seguintes funcionalidades:

- consultar recursos de vários tipos (por exemplo livros, artigos, trabalhos de alunos, relatórios)
- fazer download e upload de recursos
- pré-visualizar os vários recursos
- navegar pelos recursos através de filtros, ordenações e procuras
- classificar e comentar recursos

Para realizar este projeto, irá-se usar as tecnologias lecionadas este semestre na UC de Processamento e Representação de Informação. A aplicação Web vai ser feita recorrendo à *framework* **Express** e utilizando as técnicas lecionadas de autenticação e gestão de ficheiros e de rotas .

2 Arquitetura da solução

2.1 Micro Serviços

De forma a atender ao projeto pedido, o nosso projeto foi arquitetado tendo em conta uma solução baseado em micro-serviços:

- **App Server:**

- É o micro-serviço responsável pela interação com o cliente. Esta interação é feita através de diversas interfaces gráfica criada recorrendo ao **View Engine** Pug. Após receber pedidos do cliente, comunica com o *Auth Server* e/ou *Api-server* para receber os dados necessários para conseguir responder ao cliente.

- **Auth Server:**

- É o micro-serviço responsável pela gestão dos dados dos utilizadores.

- **API Server:**

- É o micro-serviço pela gestão dos dados referentes aos recursos e aos comentários. É por aqui que todos os pedidos referentes a estes dados passam.

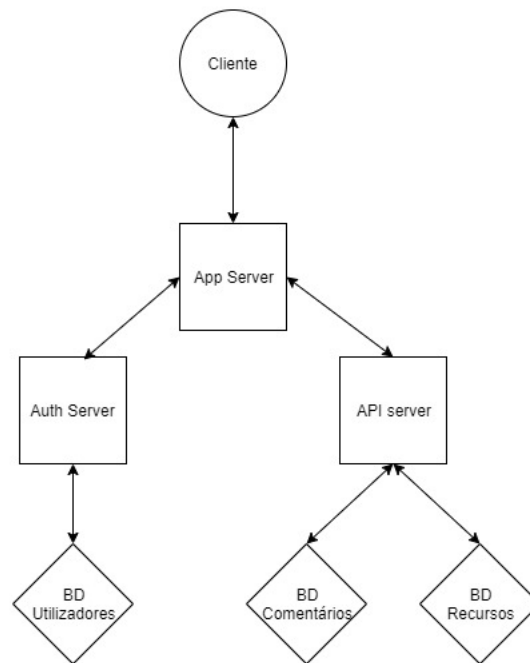


Figura 1: Arquitetura do sistema.

2.2 Autenticação

A autenticação foi implementada fazendo uso do middleware **Passport**, recorrendo à **LocalStrategy** e **FacebookStrategy** para autenticar o utilizador. O *setup* do Passport foi feito no App Server, mas como já foi referido na secção anterior, os dados referentes aos users encontram-se no Auth Server.

2.2.1 LocalStrategy

Esta estratégia verifica se o username e a password introduzidos correspondem aos dados presentes na base de dados.

Assim sendo, quando um utilizador insere as suas credenciais de acesso (Username e Password), a Estratégia que se definiu pelo Passport verifica que os dados do utilizador estão de acordo com os dados presentes na nossa base de dados de Users. Caso estes coincidam, autentica o utilizador, caso não, impede o acesso.

2.2.2 FacebookStrategy

Esta estratégia tira proveito das credenciais de uma conta Facebook para autenticar um utilizador. Ao ser reencaminhado para uma página do Facebook são pedidas as credenciais da conta. Caso essas credenciais sejam válidas o utilizador é reencaminhado de volta para a nossa aplicação e é-lhe concedido o acesso.

2.2.3 GoogleStrategy

A dinâmica desta estratégia é muito idêntica à anterior, seguindo exatamente a mesma lógica mas recorrendo a autenticação da Google. Para ambas as estratégias é necessário o utilizador estar registado como *tester* da aplicação na respetiva plataforma, visto não se encontrar no domínio público.

2.3 Persistência de Dados

A persistência de dados foi realizada recorrendo ao **MongoDB**, fazendo uso de três *collections* diferentes: **Recursos** e **Comentários**, acessíveis através do Api Server, e **Users**, acessível através do Auth Server.

A separação dos dados dos Users foi uma decisão de *design* realizada tendo em conta questões de segurança, uma vez que os dados dos utilizadores e da sua privacidade são o mais importante. Assim, caso alguém consiga obter acesso direto à API de dados, isto não implicaria que obtivesse também acesso aos dados dos utilizadores.

2.3.1 Users

As informações guardadas referentes aos users são as seguintes:

- **username:** Nome de utilizador.
- **password:** Palavra-passe.
- **level:** Tipo de conta. Pode ser consumidor, produtor ou admin.
- **email:** Endereço de Email associado à conta.
- **filiação:** Se é uma conta de um estudante ou professor, de que departamento e escola.
- **dataRegisto:** Data em que o registo da conta foi efetuado.
- **dataLastAcess:** Data em que foi efetuado log in pela última vez.
- **dataLastLastAcess:** Parâmetro que guarda a penúltima data de *login*. Usado para fornecer ao utilizador uma página de notícias que contém todos os recursos que foram *uploaded* desde o seu último acesso.
- **pedido:** Parâmetro que contém informação sobre se o utilizador realizou um pedido para fazer *upgrade* da conta.

2.3.2 Recursos

As informações guardadas referentes aos Recursos são as seguintes:

- **tipo:** Tipo de recurso (teste, relatório, ficha,...).
- **titulo:** Título do recurso.
- **subtitulo:** Subtítulo do recurso (opcional).
- **dataRegisto:** Data em que o registo foi uploaded na app.
- **visibilidade:** Recurso pode ser público ou privado.
- **dataCriação:** Data de criação do recurso.
- **autor:** Nome do autor do recurso.
- **owner:** Username da conta que fez upload do recurso.
- **hashtags:** Pequenas palavras que enquadram o recurso numa determinada área (ML, SD, compiladores...).
- **fileName:** Nome do ficheiro uploaded.
- **preview:** Campo que indica que o ficheiro pode ser previewed ou não.
- **classificaco:** Classificação(*rating*) associado ao recurso. Calculado a partir da média das classificações todas associadas ao recurso.
- **classificacoes:** Array que guarda a informação de todas as classificações feitas a este recurso, associando a conta e a classificação.

2.3.3 Comentários

As informações guardadas referentes aos Recursos são as seguintes:

- **recursoID**: ID do recurso ao qual este comentário esta associado.
- **userID**: ID do user que realizou este comentário.
- **recursoTitulo**: Titulo do recurso associado ao recursoID.
- **text**: Conteúdo do comentário.
- **data**: Data em que o comentário foi feito.

3 Funcionalidades

Neste projeto, os vários níveis dos utilizadores refletem nas funcionalidades a que este podem ter acesso. Os níveis mais altos podem aceder a todas as funcionalidades permitidas aos mais baixos. Assim, nesta secção, declara-se os acessos de cada nível, começando do mais baixo para o mais alto.

3.1 Consumidor

Acesso às páginas:

- Lista de Recursos
- Recurso
- Novidades
- Procurar
 - Encontrados por título
 - Encontrados por hashtag
- Recursos Hashtag
- Meu perfil
 - Recursos pessoais
- Perfil de Utilizador
 - Recursos do utilizador

Ações permitidas (apenas tem acesso a recursos públicos ou a recursos próprios caso tenha sido previamente um produtor):

- Consultar lista de todos os recursos.

- Consultar lista de recursos novos.
- Filtrar, ordenar e procurar (por título e hashtag) recursos.
- Consultar, comentar e classificar um recurso.
- Eliminar o próprio comentário.
- Fazer *download* ou *preview* de um recurso.
- Eliminar e alterar a visualização de recursos próprios.
- Consultar todos os recursos com determinada hashtag.
- Consultar o perfil de utilizadores (informações, comentários e recursos).
- Pedir para ser produtor.

3.2 Produtor

Acesso às páginas:

- Os meus recursos
- Upload recurso
- Editar recurso

Ações permitidas (apenas tem acesso a recursos públicos ou a recursos próprios):

- Consultar a lista dos seus recursos.
- Adicionar um novo recurso ao sistema.
- Eliminar e editar os seus recursos pessoais.

3.3 Admin

Acesso às páginas:

- Lista de Utilizadores
 - Lista de Consumidores
 - Lista de produtores

Ações permitidas (tem acesso a todos os recursos):

- Consultar a lista de utilizadores.
- *Upgrade* (tornar produtor) e *Downgrade* (tornar consumidor) um utilizador.
- Editar qualquer recurso.
- Eliminar qualquer recurso, comentário ou utilizador.
- Eliminar todos os comentários de um recurso.

4 Conclusão

Este projeto foi desenvolvido com o objetivo de criar uma WebApp para fazer a Gestão e Disponibilização de Recursos Educativos.

Com este trabalho conseguimos ganhar muita experiência numa área que outrora nunca tinha sido abordada no curso. Tivemos de aprender a improvisar e a descobrir por nós próprios soluções para os problemas e contratempos com os quais nos deparamos.

Conseguimos pôr em prática tudo o que aprendemos durante o semestre, desde autenticação, gestão de rotas e persistência de dados com o MongoDB usando o mongoose.

Em suma, este projeto foi dos projetos mais trabalhosos que já tivemos de realizar, mas também dos mais gratificantes pois sentimos que aprendemos imenso e consideramos que todos os objetivos foram cumpridos e todas as funcionalidades implementadas.