UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES

Documentação

Projeto 3

Gerson Revoredo Pereira - N°USP 8598728
Marco Modena Centeno - NºUSP 9377612
Rodrigo Kuba - N°USP 5364617
Thiago Bonfiglio Santos - N°USP 8598691

Sumário

1 - Front-end

Nesta seção passaremos por cima das tecnologias utilizadas na parte de front-end junto com algumas considerações.

1.1 - HTML

No HTML utilizamos o *index.html* para servir de home, com o menu fixado no topo e, utilizando a função *router* do Angular, criamos um *body* que possui a habilidade de mudar baseado na escolha do menu.

As páginas HTML são segmentadas, facilitando a edição de terceiros sem conhecimento prévio, e guardadas na pasta *Panel*.

Utilizamos também o *Font Awesome*, database de símbolos e marcas online, para achar símbolos, como o utilizado no botão de pesquisa no setor de tags

1.2 - CSS

Utilizamos o css para mudanças visuais médias e pequenas, como cores e botões simples, a maior parte visual ficou por parte do Bootstrap devido a motivos discutidos na seção do Bootstrap.

Tivemos dois “elementos” principais criados no CSS, as bordas azuis e os botões.

As bordas azuis foram criadas para serem utilizadas na vertical e na horizontal, anexadas a uma *div* (na maioria dos casos), facilitando ao usuário a distinção de áreas e criando uma identidade visual em conjunto ao resto do site.

Por fim, os botões possuem uma função similar ao das bordas, criar uma identidade visual com animações próprias e facilidade de uso.

1.3 - Bootstrap

Decidimos utilizar o Bootstrap por dois motivos principais, organização em grande escala e portabilidade. Quando nos referimos a organização de grande escala nos referimos a classes como *jumbotron* e *forms*, que são classes de suporte de grande escala com alta dificuldade para realização no CSS, e ainda aproveitamos as tecnologias providas para criar um menu mais elegante e funcional.

Apesar de tudo, o principal motivo foi a portabilidade, tendo em mente os objetivos ressaltados pela professora de que os projetos serão continuados por outro grupo, resolvemos utilizar o Bootstrap pela facilidade de fazer um site responsivo, e apesar de não termos aplicado a tecnologia, o frame e alguns elementos já estão em posição, portanto facilitando uma futura transferência para o mobile.

1.4 - AngularJS

Adotamos o Angular como método para facilitar o controle do conteúdo do site, os módulos e funções utilizados foram:

Router foi usado para modificar e controlar o conteúdo da página, em conjunto a modulação de todas os HTMLs. Ele cria um link a um HTML baseado em um arquivo que foi associado a um *controller* angular, facilitando a customização de cada item.

ng-repeat possui a função de repetir, usando um padrão pré-definido (listas, grades,

…), até que todos os itens foram passados para a página, facilitando o display de um grande conjunto de informações de forma ordenada. Foi utilizado em todos em todos os locais com display de licitações e tags.

Além das duas previamente citadas, criamos uma series de funcoes internas para ordenar, procurar e selecionar licitações e tags.

2 - Back-end

Nesta seção passaremos por cima das tecnologias utilizadas na parte de back-end junto com algumas considerações.

2.1 - NodeJS

NodeJS é um interpretador de códigos escritos em JavaScript, utilizado no lado servidor da aplicação. Nesse projeto, o NodeJS foi utilizado para fornecer APIs WebServices Restful para serem utilizados/consumidos pelo lado "Front-end" da aplicação.

Com o NodeJS, podemos utilizar diversos pacotes/módulos "*open-source*", que facilita e agiliza o desenvolvimento de aplicações.

Nesse projeto, utilizamos os seguintes pacotes:

1. body-parser: utilizado para realizar a conversão dos dados da tag "body" do HTML em forma de objeto.
2. cookie-parser: utilizado para realizar a conversão dos dados do cookie em forma de objeto.
3. debug: utilitário para ajudar na depuração dos códigos-fontes do projeto. Utilizado na fase de desenvolvimento da aplicação.
4. express: um dos principais pacotes utilizado no projeto. É um arcabouço para utilização do protocolo HTTP e também para fornecer APIs "web-services" no formato RestFul. Faz parte do MEAN, acrônimo utilizado para identificar aplicações que utilizam o banco de dados, MongoDB, o Express, o arcabouço AngularJS e o servidor web NodeJS.
5. jade: utilitário para escrever códigos em HTML.
6. jsonwebtoken: utilitário utilizado para geração de "tokens". Foi utilizado na implementação do mecanismo de autenticação do projeto.
7. moment: utilitário para o tratamento de datas e horas.
8. mongoose: utilitário ODM ("Object Document Modeling") para o MongoDB.
9. morgan: utilitário de log para NodeJS.
10. passport: utilitário de autenticação para NodeJS. Utilizado juntamente com o jwt (jsonwebtoken), faz parte do mecanismo de autenticação da aplicação.

2.2 - MongoDB

MongoDB é um banco de dados do tipo não-relacional (NoSQL - "Not Only SQL"), de alto desempenho e orientado a documentos do tipo JSON ("JavaScript Object Notation").

No projeto, o banco de dados "Licitation" é criado automaticamente pela aplicação. Esse banco de dados possui dois tipos de documentos: "user" e "licitation":

```
var tagSchema = new Schema({
    tag:
    {
      type: String,
      required: true,
      unique: true
    }
});

var user = new Schema({
    username: String,
    password: String,
    email:
    {
      type: String,
      required: true
    },
    documentNumber: Number,
    admin:
    {
        type: Boolean,
        default: false
    },
    tags: [tagSchema]
});
```

Figura 1: Esquema do documento "user"

```
var licitation = new Schema({
    title: String,
    description: String,
    deliveryDate: Date,
    publishDate: Date
});
```

Figura 2: Esquema do documento "licitation"

Esses esquemas são utilizados pelo utilitário mongoose para validação dos dados a serem inseridos no banco de dados e também para realizar a modelagem dos documentos.

2.3 - APIs RESTful

As seguintes APIs RESTful são fornecidas pela aplicação:

1. "/users"
    a. verbo GET: Retorna os dados do usuário. Necessário autenticação.
        a. JSON de retorno:


```
{
  "_id": "594afb8b1232b81e4073e7b0",
  "username": "leonilda",
  "email": "leonilda@gmail.com",
  "documentNumber": 66788433,
  "__v": 2,
  "tags": [
    {
      "tag": "tomate",
      "_id": "594afbce1232b81e4073e7b1"
    },
    {
      "tag": "suco",
      "_id": "594afbd21232b81e4073e7b2"
    }
  ],
  "admin": false
}
```


2. "/users/register"
    a. verbo POST: Utilizado para criação de um novo usuário.
        a. JSON de entrada:


```
{
  "username": "usuario",
  "password": "senha",
  "email": "email@gmail.com.br",
  "documentNumber": 123456777
}
```


        b. JSON de retorno OK:


```
{
  "status": "Registration Successful!"
}
```


        c. JSON de retorno Erro:


```
{
  "err": {
    "name": "UserExistsError",
    "message": "A user with the given username is already registered"
  }
}
```


3. "/users/login"
    a. verbo POST: Utilizado para autenticação do usuário.
        a. JSON de entrada:

```
{
  "username": "usuario",
  "password": "senha"
}
```

b. JSON de retorno OK:


```
{
  "status": "Login successful!",
  "success": true,
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
      .eyIkX18iOnsic3RyaWN0TW9kZSI6dHJ1ZSwic2VsZWN0ZWQiOn
      iOiJpbml0IiwiYWRtaW4iOiJpbml0IiwidGFncyI6ImluaXQiLC
      iLCJzYWx0IjoiaW5pdCIsIl9pZCI6ImluaXQifSwic3RhdGVzIj
      ydWUsImRvY3VtZW50TnVtYmVyIjp0cnVlLCJlbWFpbCI6dHJ1ZS
      lcXVpcmUiOnt9fSwic3RhdGVOYW1lcyI6WyJyZXF1aXJlIiwibW
      zIjp7InNhdmUiOltudWxsLG51bGxdLCJpc05ldyI6W251bGwsbn
      kbWluIjpmYWxzZSwidGFncyI6W10sIl9fdiI6MCwiZG9jdW1lbn
      iLCJoYXNoIjoiOTRlNzRiYzI5M2YwMDU3YTAyMTViYzAxYTQ1MD
      mZTZkNTNkNGUzN2ZiNzMwNjU4OGE3YmY0YzgzZDQ4YTM2ZmEzNz
      wMzliN2Q3MWYwZGNjOTMwNjY2ZmNjNjVhNmVmNDNlM2RiMWEzOD
      5ZTA4OGM1OTE4ZmI2YjY4OWFkMTM0NTVjNGE4MTBlNTQ2ZDE2Yz
      1YWU2OTU0MGU2ZWQ4NTljN2M4MmQ3ZTU0NmVkODFhYmJiYmQ1YW
      2ZjM1NjM0NmY4Njg1NTMyY2I4NzJlOTZkNzA2ODEwYTY0NTc0MW
      iMmRlNWEwYzdhNDgxNTRhNmQxZWI4NDkzMWVmMWI5YTcxOThkMD
      jN2ZhZGU0YTJlNTQyZDhhNzYwYjEwNDI2YWJiNjk3YTY4NjU5Zm
      3MmFmMDA5M2FjNDUzZjEwNTY5YjFmMmRjMjg5Zjc4Zjc1MGYzYT
      iMWI1Y2VkMDE3MjU3YzI2YmI2NzI2MDk4OWYwOGE0ZmNhNjNhZW
      kNDA0NGZlNmE1YjgwNjhhZGNiYWJhMDJlYWY1ZDRhMGMxZTQ5OW
      lLCJpYXQiOjE0OTgxMDAxNjksImV4cCI6MTQ5ODEwMzc2OX0.QW
  "userId": "594b30f21232b81e4073e7b4"
}
```


c. JSON de retorno Erro:

```
{
  "err": {
    "name": "IncorrectPasswordError",
    "message": "Password or username are incorrect"
  }
}
```

4. "/users/tag"
   a. verbo GET: Utilizado para recuperar as "tags" do usuário. Necessário autenticação.
      a. JSON de retorno:

```
[
  {
    "tag": "tomate",
    "_id": "594afbce1232b81e4073e7b1"
  },
  {
    "tag": "suco",
    "_id": "594afbd21232b81e4073e7b2"
  }
]
```

   b. verbo POST: Utilizado para incluir uma nova "tag" do usuário. Necessário

autenticação.

a. JSON de entrada:

```
{
  "tag": "tomate"
}
```

b. JSON de retorno:

```
{
  "_id": "594afb8b1232b81e4073e7b0",
  "username": "leonilda",
  "email": "leonilda@gmail.com",
  "documentNumber": 66788433,
  "__v": 3,
  "tags": [
    {
      "tag": "tomate",
      "_id": "594afbce1232b81e4073e7b1"
    },
    {
      "tag": "suco",
      "_id": "594afbd21232b81e4073e7b2"
    },
    {
      "tag": "carro5",
      "_id": "594b34471232b81e4073e7b6"
    }
  ],
  "admin": false
}
```

5. "/users/tag/:tagId"
   a. verbo DELETE: Utilizado para deletar uma tag com o id "tagId". Necessário autenticação.
      a. JSON de retorno:

```
{
  "_id": "594afb8b1232b81e4073e7b0",
  "username": "leonilda",
  "email": "leonilda@gmail.com",
  "documentNumber": 66788433,
  "__v": 4,
  "tags": [
    {
      "tag": "tomate",
      "_id": "594afbce1232b81e4073e7b1"
    },
    {
      "tag": "suco",
      "_id": "594afbd21232b81e4073e7b2"
    }
  ],
  "admin": false
}
```

6. "/licitations"
   a. verbo GET: Utilizado para retornar as licitações cadastradas na aplicação.
      a. JSON de retorno:

```
{
  "licitations": [
    {
      "_id": "594744d9203eb91269f7e087",
      "title": "Licitacao Exemplo 1",
      "description": "Descrição exemplo de Licitação",
      "deliveryDate": "2017-06-19T03:28:25.000Z",
      "publishDate": "2017-06-19T03:28:25.000Z",
      "__v": 0
    },
    {
      "_id": "5947472d203eb91269f7e08b",
      "title": "Licitacao Exemplo 2",
      "description": "Descrição exemplo de Licitação 2",
      "deliveryDate": "2017-06-19T03:38:21.000Z",
      "publishDate": "2017-06-19T03:38:21.000Z",
      "__v": 0
    },
    {
      "_id": "59474916576aaf12866bd38b",
      "title": "Licitacao Exemplo 3",
      "description": "Descrição exemplo de Licitação 2",
      "deliveryDate": "2017-10-22T02:00:00.000Z",
      "publishDate": "2017-07-19T03:00:00.000Z",
      "__v": 0
    },
```


7. "/licitations/user"
    a. verbo POST: Utilizado para retornar as licitações cadastradas na aplicação e que contenham as "tags" cadastradas pelo usuário. Necessário autenticação.
        a. JSON de retorno:


```
{
  "licitations": [
    {
      "_id": "594abd771232b81e4073e7ac",
      "title": "Licitacao legumes, verduras e frutas",
      "description": "Descrição  de Licitação de tomate, alface, laranja e cebolas",
      "deliveryDate": "2017-12-01T02:00:00.000Z",
      "publishDate": "2017-06-19T03:00:00.000Z",
      "__v": 0
    },
    {
      "_id": "594afc251232b81e4073e7b3",
      "title": "Licitacao bebidas",
      "description": "Descrição  de Licitação de suco e água",
      "deliveryDate": "2017-12-01T02:00:00.000Z",
      "publishDate": "2017-06-19T03:00:00.000Z",
      "__v": 0
    }
  ]
}
```


4 - Itens Incompletos

Dentre os itens que não conseguimos implementar a tempo, mas que eram parte do escopo:

- Exclusão de Tags: atualmente não está removendo do banco de dados;
- Envio de emails: não chegamos a implementar a função de enviar emails no site;
- Troca de senha e esqueci minha senha: telas foram implementadas, mas não estão funcionais;
- Pesquisar: foi usando um filtro simples e não um conjunto com vários filtros;
- Telas do menu Sobre: Objetivos, Quem Somos, Contatos e Mapa do Site não foram implementadas.
- Telas de licitações novas e do painel não foram implementadas também;
- Conexão com o banco de dados da outra equipe.

Dentre possíveis melhorias ficou pendente:

- Elastic Search para ter um filtro mais poderoso nas buscas;
- Filtro por valor;
- Aplicativo mobile.

5 - Prioridades

Priorizamos um MVP do site com as funcionalidades básicas abrindo mão de tentar conexão com a base do outro grupo e nos focando mais no site propriamente dito.

Dentro das prioridades estava construir um banco de dados nosso e uma estrutura de webservices para intermediar a conexão do banco de dados e com o site. Depois montamos as telas e fomos aplicando a inteligência delas com Angular buscando os dados por meios dos webservices.