

## Buscando alunos próximos

### Transcrição

Estamos bem próximos de finalizar a funcionalidade de busca por proximidade dos alunos. Agora precisaremos disponibilizar a seleção de um aluno na página `pesquisar.html` da pasta de `geolocalizacao`, para que assim possamos localizar os outros geograficamente próximos a ele.

Primeiro substituiremos o código HTML da página para o que é apresentado abaixo, inserindo nele o código de um formulário simples com um `select`, em que iteramos uma lista de alunos:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8" />
<link type="text/css" rel="stylesheet"
      href="....../materialize/css/materialize.min.css"
      media="screen,projection" />
<link href="https://fonts.googleapis.com/icon?family=Material+Icons"
      rel="stylesheet" />
<title>EscolaAlura</title>
<script type="text/javascript"
       src="https://code.jquery.com/jquery-2.1.1.min.js"></script>
<script type="text/javascript" src="../js/inicializar.js"></script>
<script type="text/javascript" src="../js/googlemaps.js"></script>

<link type="text/css" rel="stylesheet" href="..../css/main.css" />
</head>
<body class="grey lighten-3">
  <div id="formularioGeolocalizacao" class="container">
    <h3 class="main-title center">Visualizar Aluno</h3>
    <form class="col s12" action="/geolocalizacao/pesquisar" method="get">
      <div class="row">
        <div class="input-field col s12">
          <select name="alunoId" id="alunoId">
            <option th:each="aluno : ${alunos}" th:value="${aluno.id}"
                   th:text="${aluno.nome}">Nome do Aluno</option>
          </select> <label for="aluno">Alunos</label>
        </div>
      </div>
      <div th:if="${alunosProximos} != null">
        <div class="row">
          <div id="map" class="mapa"></div>
        </div>
      </div>
      <div class="row">
        <div class="input-field col s12 center">
          <button class="btn waves-effect waves-light" type="submit">Pesquisar</button>
        </div>
      </div>
    </form>
  </div>
</body>
```

```

</form>

<script
  src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAIPw0BOFbsav12S0XJm366GqIL8W&
  async="async" defer="defer"></script>

</div>
<script type="text/javascript"
  src="../../materialize/js/materialize.min.js"></script>
</body>
</html>

```

Mas sabemos que a página de pesquisa ainda não possui estes alunos disponíveis para a criação do *select* corretamente. Precisamos adicionar esta listagem no `GeolocalizacaoController` :

```

@Controller
public class GeolocalizacaoController {

    @Autowired
    AlunoRepository alunoRepository;

    @GetMapping("/geolocalizacao/iniciarpesquisa")
    public String inicializarPesquisa(Model model) {
        List<Aluno> alunos = alunoRepository.obterTodosAlunos();
        model.addAttribute("alunos", alunos);
        return "geolocalizacao/pesquisar";
    }

}

```

Com isso já teremos o *select* com os nomes dos alunos. Criaremos o código que irá responder à submissão do formulário, o qual, depois de selecionarmos o aluno e o submetermos, capturaremos seu *id*, buscando outros alunos próximos (com base em suas coordenadas) e retornando-os para serem exibidos no mapa da mesma página.

Caso testemos a aplicação agora, veremos que o mapa não aparece mais. Isso porque no HTML incluímos uma condição de que só apresentaremos o mapa caso o objeto `alunosProximos` seja retornado pelo Java.

```

<div th:if="${alunosProximos} != null">
    <div class="row">
        <div id="map" class="mapa"></div>
    </div>
</div>

```

No *controller*, implementaremos a busca dos alunos próximos. Primeiramente receberemos o *id* do aluno e o buscaremos no banco de dados, coletando suas informações geográficas.

```

@GetMapping("/geolocalizacao/pesquisar")
public String pesquisar(@RequestParam("alunoId") String alunoId, Model model) {
    Aluno aluno = alunoRepository.obterAlunoPor(alunoId);
    List<Aluno> alunosProximos = alunoRepository.pesquisaPorGeolocalizacao(aluno);
}

```

```
model.addAttribute("alunosProximos", alunosProximos);
return "geolocalizacao/pesquisar";
}
```

Estamos utilizando nosso repositório de alunos para efetuar a busca por geolocalização, porém esse método ainda não está disponível, pois não o criamos. O restante já não é novidade. Vamos implementar essa busca, então. O código do método se encontra abaixo:

```
public List<Aluno> pesquisaPorGeolocalizao(Aluno aluno) {
    criarConexao();
    MongoCollection<Aluno> alunoCollection = this.bancoDeDados.getCollection("alunos", Aluno.class);
    alunoCollection.createIndex(Indexes.geo2dsphere("contato"));

    List<Double> coordinates = aluno.getContato().getCoordinates();
    Point pontoReferencia = new Point(new Position(coordinates.get(0), coordinates.get(1)));

    MongoCursor<Aluno> resultados = alunoCollection.find(Filters.nearSphere("contato", pontoReferencia, 2));
    List<Aluno> alunos = popularAlunos(resultados);

    fecharConexao();
    return alunos;
}
```

Neste método fizemos uma série de passos; vamos ver um a um. Inicialmente abrimos uma conexão com o banco de dados, e em seguida selecionamos a coleção de alunos, a partir do qual criamos um índice baseado no campo `contato` de cada aluno.

O índice é um passo importante por ser nele que indicamos que o determinado campo possui dados relacionados à geolocalização, o que fizemos pelo método `geo2dsphere` da classe `Indexes`. Consideramos o campo `contato` um objeto esférico 2D.

O terceiro passo é criar um ponto de referência baseado nas coordenadas do aluno selecionado. Depois utilizamos esse ponto de referência em um novo filtro chamado `nearSphere`, em que indicamos o campo a ser filtrado, por qual ponto de referência, as distâncias máxima e mínima de comparação.

Feito isso, limitamos o número de alunos próximos a 2 e pulamos o primeiro deles, já que aluno mais próximo de você é você mesmo! Por fim, recebemos um iterador dos resultados dessa busca, criamos a lista de alunos e retornamos.

Após este passo, vamos testar! O mapa aparecerá na pesquisa, porém, como não configuramos os pontos do mapa, não há nenhum indício além da aparição do próprio mapa. Os pontos no mapa indicando a posição dos alunos faz parte do nosso próximo passo.