

Criando nosso mapa

Transcrição

A partir de agora desenvolveremos a funcionalidade mais complexa da aplicação, que trabalha com geolocalização. A motivação é que, como dissemos no começo do curso, a escola não possui boa localização, e quer viabilizar aos pais dos alunos informações sobre aqueles que se encontram próximos para poderem oferecer caronas, diminuindo o fluxo de veículos na localidade.

Essa funcionalidade se utilizará dos recursos de Google Maps e geolocalização, juntamente com os recursos de cálculo de proximidade do MongoDB, um dos mais interessantes deste banco de dados.

Seguindo os procedimentos a que já estamos acostumados, criaremos o cartão no `index` para o direcionamento à página do mapa.

```
<div class="col s12 m4 l4 waves-effect waves-light">
  <a href="/geolocalizacao/iniciarpesquisa">
    <div class="col s12 m4 l4 waves-effect waves-light ">
      <div class="card-panel hoverable z-depth-1 center grey lighten-4">
        <i class="large material-icons ">my_location </i>
        <div class="truncate">Pesquisar por Geolocalizacao</div>
      </div>
    </div>
  </a>
</div>
```

Criaremos também um novo *controller* chamado `GeolocalizacaoController`, somente para o carregamento do *template*:

```
@Controller
public class GeolocalizacaoController {

    @GetMapping("/geolocalizacao/iniciarpesquisa")
    public String inicializarPesquisa() {
        return "geolocalizacao/pesquisar";
    }

}
```

Criaremos também uma nova pasta em `resources/templates`, chamada `geolocalizacao`, e nela teremos o *template* inicial denominado `pesquisar.html`.

```
<!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=
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />
```

```
<title>EscolAlura</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>

</head>
<body class="grey lighten-3">
  <div id="formularioGeolocalizacao" class="container">
    <h3 class="main-title center">Visualizar Aluno</h3>
    <div class="row">

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

Nosso próximo passo é configurar devidamente o mapa em nossa aplicação. Porém, antes disso, faremos algumas observações. [A API do Google Maps \(https://developers.google.com/maps/\)](https://developers.google.com/maps/) é paga, porém, permite que utilizemos alguns dos seus recursos gratuitamente, o que será suficiente para vermos o exemplo deste curso funcionando.

Caso queira adicionar essa API em uma aplicação real, verifique os preços e planos, pois a opção gratuita é bem limitada, e pode deixar o usuário frustrado quando o mapa não deixar de aparecer em suas páginas.

A API do Google Maps está disponível para diversas plataformas, e utilizaremos a [plataforma Web com JavaScript \(https://developers.google.com/maps/documentation/javascript/\)](https://developers.google.com/maps/documentation/javascript/). Para isso, precisaremos criar uma chave que identifica a nossa aplicação. Na [página inicial da documentação da API JavaScript \(https://developers.google.com/maps/documentation/javascript/\)](https://developers.google.com/maps/documentation/javascript/) existe um botão "Obter uma chave".

Ao clicarmos no botão "Obter uma chave", será aberta uma janela pedindo para selecionarmos um projeto do Google Console. Caso o projeto não exista, ele criará um, bastando informar um nome. Depois podemos clicar em "Ativar API" ou "Criar e ativar API", e logo teremos nossa chave, similar ao texto abaixo:

AIzaSyA03zKo_Ey2LXS6y9SS12t9Fq4ZIA29l0k

Agora, como poderemos inserir o mapa em nossa aplicação? A própria documentação traz alguns exemplos, e o mais básico deles tem o seguinte código:

```
<div id="map"></div>
<script>
  function initMap() {
    // Create a map object and specify the DOM element for display.
    var map = new google.maps.Map(document.getElementById('map'), {
      center: {lat: -34.397, lng: 150.644},
      zoom: 8
    });
  }
</script>
<script src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap"
  async defer></script>
```

Note que tudo de que precisamos é uma `div` com o id **map**, um pequeno trecho de código JavaScript e o *import* de um *script* que tem um parâmetro chamado `key`, para o qual precisamos informar a chave da aplicação. Juntando-se tudo e colando o código na página `pesquisar.html`, da pasta `resources/templates/geolocalizacao`, teremos:

```
<!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="all">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />
  <title>EscolAlura</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>

</head>
<body class="grey lighten-3">
  <div id="formularioGeolocalizacao" class="container">
    <h3 class="main-title center">Visualizar Aluno</h3>
    <div class="row">

      <div id="map"></div>
      <script>
        function initMap() {
          // Create a map object and specify the DOM element for display.
          var map = new google.maps.Map(document.getElementById('map'), {
            center: {lat: -34.397, lng: 150.644},
            zoom: 8
          });
        }

      </script>
      <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyA03zKo_Ey2LXS6y9SS12t9FqQ" async"></script>

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

Se deixarmos o código como está, teremos dois problemas: um ligado ao caminho do *import* do JavaScript, com a presença do caractere `&`. O *Thymeleaf* não sabe como tratar esse caractere dessa forma e, para solucionarmos isto, trocaremos o caractere por sua representação em ASCII `&`;

O segundo problema se relaciona aos atributos `async` e `defer`, que não possuem valores atribuídos. O *Thymeleaf* não permite atributos soltos dessa forma. Solucionamos simplesmente atribuindo o valor com o mesmo nome do atributo, `async="async"` e `defer="defer"`.

```
<!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="all">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />
```

```

<title>EscolAlura</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>

</head>
<body class="grey lighten-3">
  <div id="formularioGeolocalizacao" class="container">
    <h3 class="main-title center">Visualizar Aluno</h3>
    <div class="row">

      <div id="map"></div>
      <script>
        function initMap() {
          // Create a map object and specify the DOM element for display.
          var map = new google.maps.Map(document.getElementById('map'), {
            center: {lat: -34.397, lng: 150.644},
            zoom: 8
          });
        }

      </script>
      <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyA03zKo_Ey2LXS6y9SS12t9Fq">
    </div>
  </div>
  <script type="text/javascript" src="../../materialize/js/materialize.min.js"></script>
</body>
</html>

```

Feito isto, não teremos mais erros em nossa aplicação, porém o mapa tampouco será exibido. Isso ocorre pois não indicamos características como altura e largura do espaço em que o mapa será desenhado. Neste caso, criaremos um arquivo CSS em `resources/static/css` chamado `main.css`, com o seguinte código:

```

.mapa{
  width: 100%;
  height: 300px;
}

```

Vamos lembrar também de importar este CSS no `head` da página:

```

<link type="text/css" rel="stylesheet" href="../../css/main.css" />

```

Com estas adições poderemos ver o mapa funcionando em nossa aplicação, como ilustra a imagem abaixo:



É importante lembrarmos que este curso não focará em funcionalidades específicas do Google Maps, como por exemplo o controle do zoom, entre outras, as quais podem ser exploradas diretamente na documentação da API.