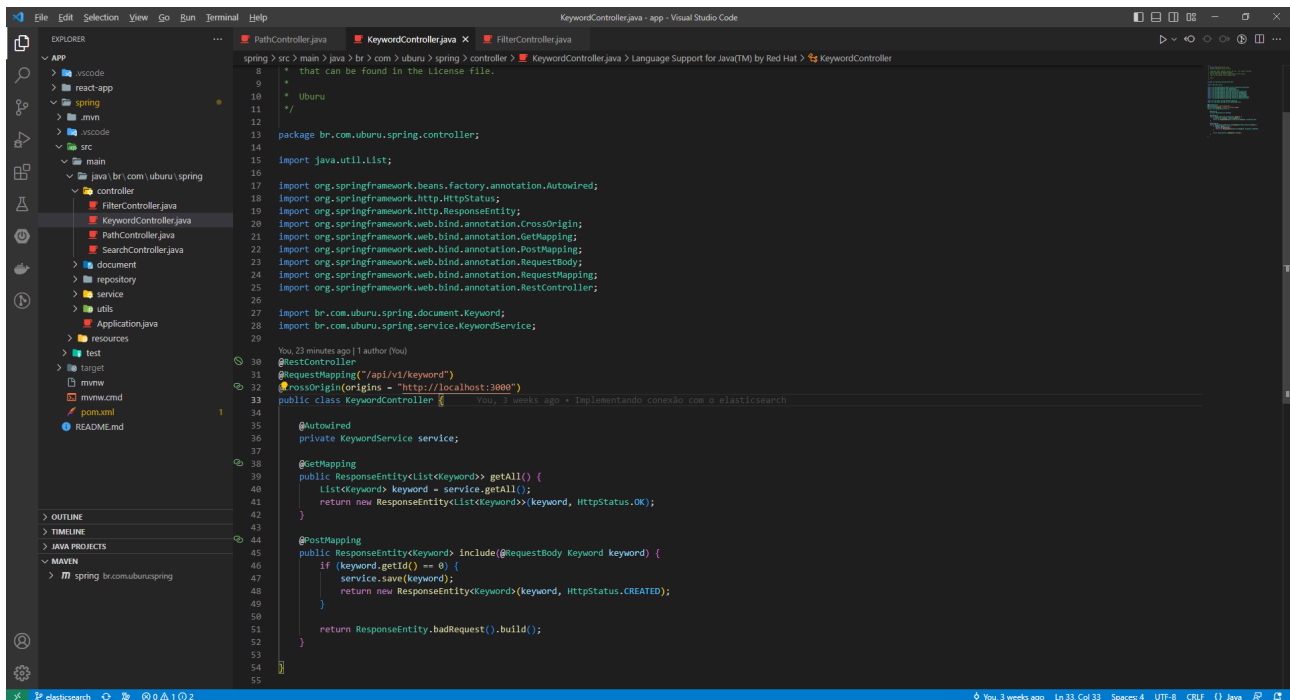


DOCUMENTAÇÃO DA API DE PESQUISA DO UBURU

1 Histórico de pesquisa

1.1 Registro do histórico de pesquisa

Os *endpoints* responsáveis por manter o histórico de pesquisa ficarão localizados no endereço local, na porta **8080** + '/' + identificador do *endpoint*. Todos os *endpoints* estão configurados para aceitar a conexão *cross-origin* apenas dos serviços localizados no endereço local, na porta **3000**:



Os métodos aceitos serão apenas GET – para buscar TODOS os registros do histórico – e POST – para inserir um novo registro. O método POST deve receber como parâmetro a data em que o registro foi feito e o valor (palavra-chave, caminho ou filtro) de acordo com o *endpoint*.

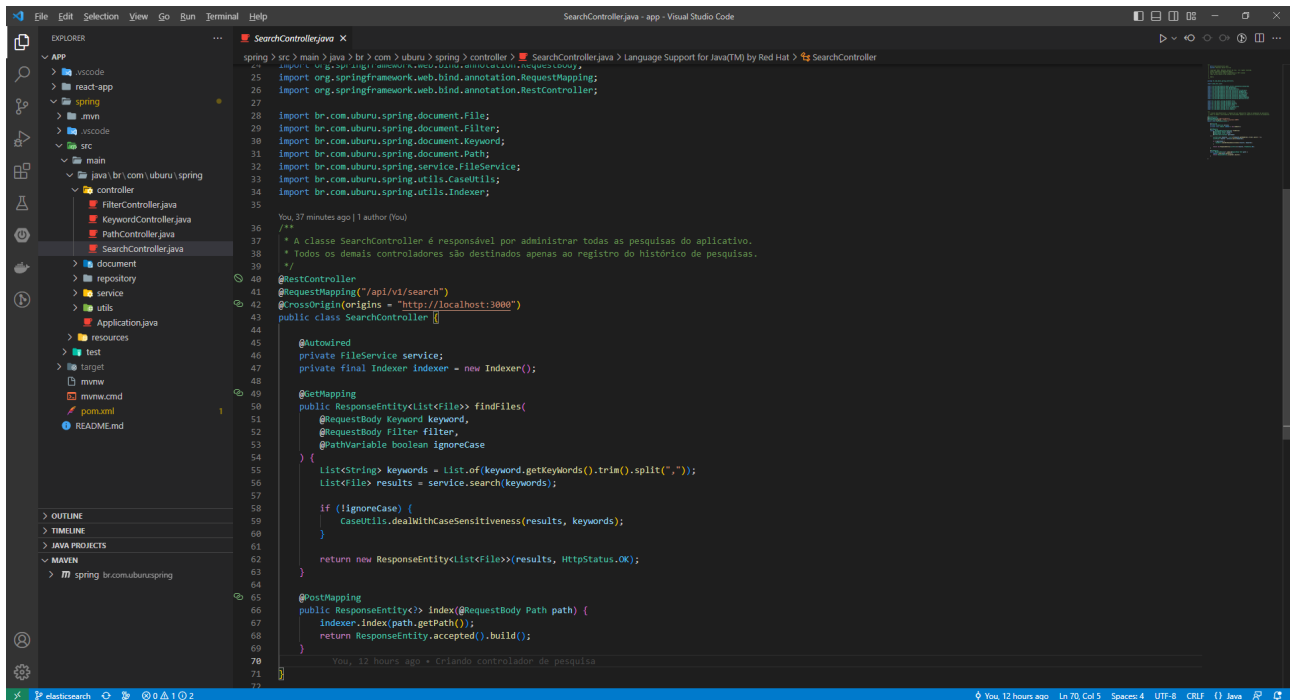
1.2 Funcionamento da API do histórico

A API do histórico é composta pelos componentes padrão de uma API SpringBoot comum, contendo uma entidade correspondente a cada campo de input, um repository e um service para utilizarem a entidade. Mesmo os componentes do histórico fazer uso do elasticsearch para armazenar os arquivos.

2 API de Pesquisa

2.1 Comunicação com a API

O endereço do *endpoint* responsável pelo mecanismo de pesquisa será o endereço local, na porta **8080**, assim como os demais, e os métodos HTTP podem ser GET ou POST. O método GET será responsável por buscar um dado no banco, enquanto o POST ficará responsável por receber um caminho para realizar a indexação.



```
spring > src > main > java > br > com > uburu > spring > controller > SearchController.java > Language Support for Java(TM) by Red Hat > SearchController
24 import br.com.uburu.spring.document.File;
25 import org.springframework.web.bind.annotation.RequestMapping;
26 import org.springframework.web.bind.annotation.RestController;
27
28 import br.com.uburu.spring.document.File;
29 import br.com.uburu.spring.document.Filter;
30 import br.com.uburu.spring.document.Keyword;
31 import br.com.uburu.spring.document.Path;
32 import br.com.uburu.spring.service.FileService;
33 import br.com.uburu.spring.utils.CaseUtils;
34 import br.com.uburu.spring.utils.Indexer;
35
36 /**
37  * A classe SearchController é responsável por administrar todas as pesquisas do aplicativo.
38  * Todos os demais controladores são destinados apenas ao registro do histórico de pesquisas.
39  */
40 @RestController
41 @RequestMapping("/api/v1/search")
42 @CrossOrigin(origins = "http://localhost:3000")
43 public class SearchController {
44
45     @Autowired
46     private FileService service;
47     private final Indexer indexer = new Indexer();
48
49     @GetMapping
50     public ResponseEntity<List<File>> findFiles(
51         @RequestBody Keyword keyword,
52         @RequestBody Filter filter,
53         @PathVariable boolean ignoreCase
54     ) {
55         List<String> keywords = List.of(keyword.getKeywords().trim().split(","));
56         List<File> results = service.search(keywords);
57
58         if (!ignoreCase) {
59             CaseUtils.dealWithCaseSensitiveness(results, keywords);
60         }
61
62         return new ResponseEntity<List<File>>(results, HttpStatus.OK);
63     }
64
65     @PostMapping
66     public ResponseEntity<> index(@RequestBody Path path) {
67         indexer.index(path.getPath());
68         return ResponseEntity.accepted().build();
69     }
70
71     You, 12 hours ago • Criando controlador de pesquisa
72 }
```

2.2 Funcionamento do backend

A classe `SearchController.java` é o endpoint que recebe a requisição. Ela cria instâncias das classes `Indexer.java` e `FileService.java` para realizar as buscas e a indexação dos repositórios. Na classe `Indexer.java` ficam os métodos que irão efetivamente gerar os registros, enquanto o `FileService.java` será responsável por se comunicar com o repository, que, por sua vez, se comunica com o elasticsearch.

Por fim, para explicar o que cada entidade representa:

- Os filtros determinam as extensões de arquivos aceitas;
- Os caminhos determinam quais caminhos serão aceitos e caso nenhum caminho seja inserido, o sistema buscará em todo o repositório indexado. Caso seja definido um (ou mais) caminhos, o sistema buscará apenas neles;
- As palavras-chave são os termos que serão buscados nos índices.