



Pós-Graduação Lato Sensu
Curso de Especialização em Tecnologia Java

APOSTILA – USANDO COMMANDLINERUNNER E LOGGER

Professor Hugo Baker Goveia
hbakergoveia@hotmail.com



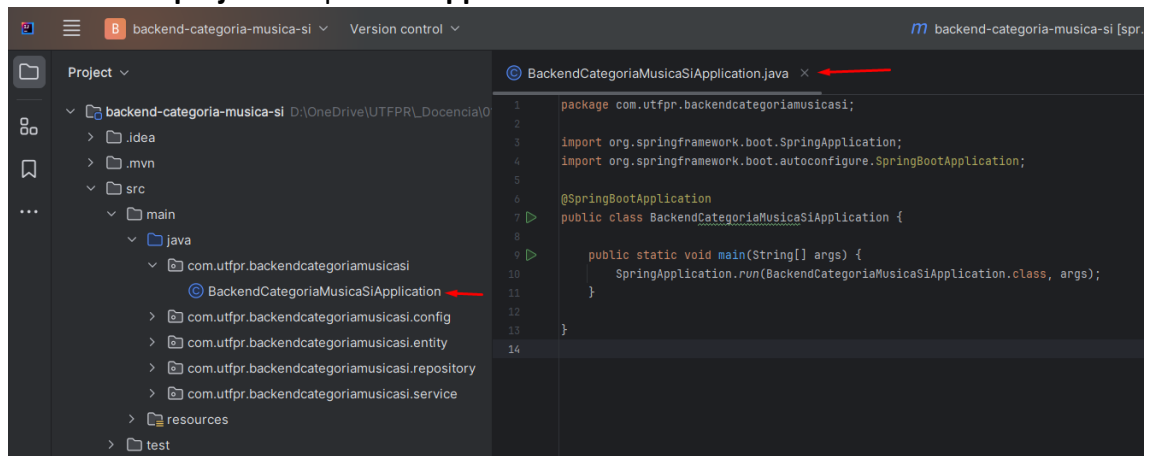
1. CommandLineRunner e Logger

As implementações para testar as atividades serão realizadas na classe “Application.java”, que no exemplo abaixo se chama “**BackendCategoriaMusicaSiApplication.java**”.

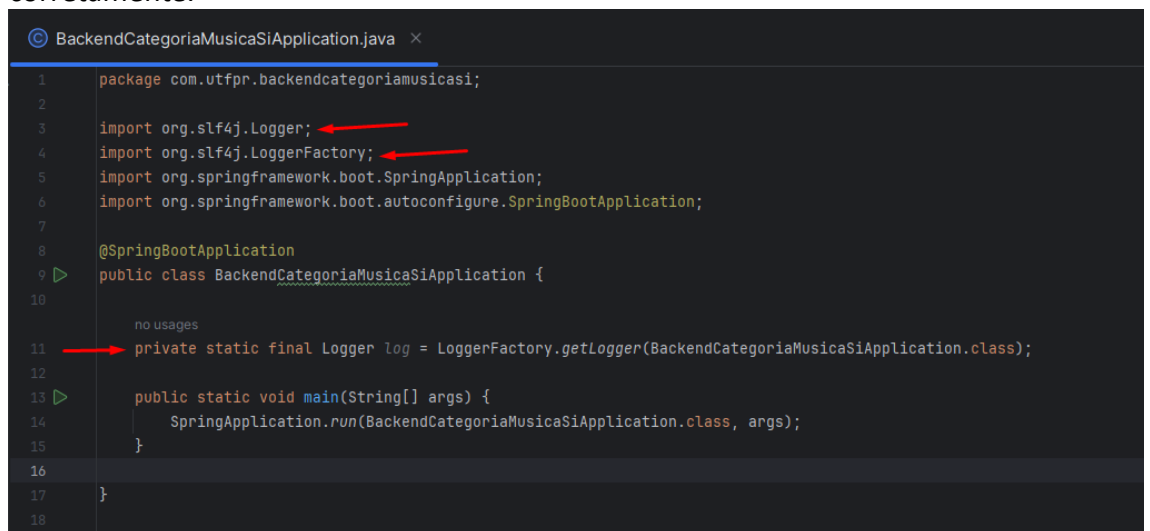
Consiste em uma classe Java comum, possuindo o método main, que também é comum em aplicações Java. E será nela que iremos usar o **CommandLineRunner** e o **Logger** para testar nossas implementações de JPA.

Etapas para implementar:

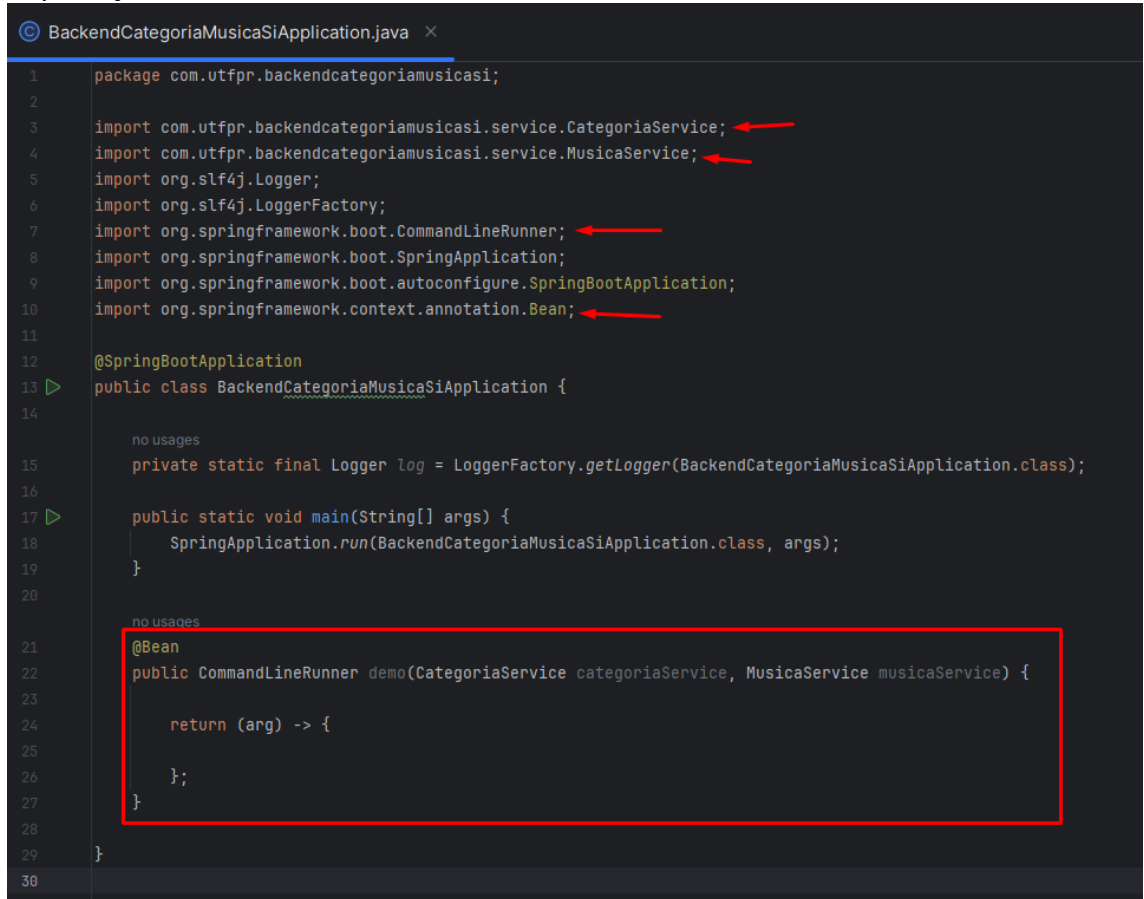
1. Abrir sua classe “**Application.java**”, normalmente essa classe tem como nome o => **nome do projeto + a palavra Application**.



2. Fazer a declaração do atributo log do tipo **Logger** e efetuar as importações corretamente.

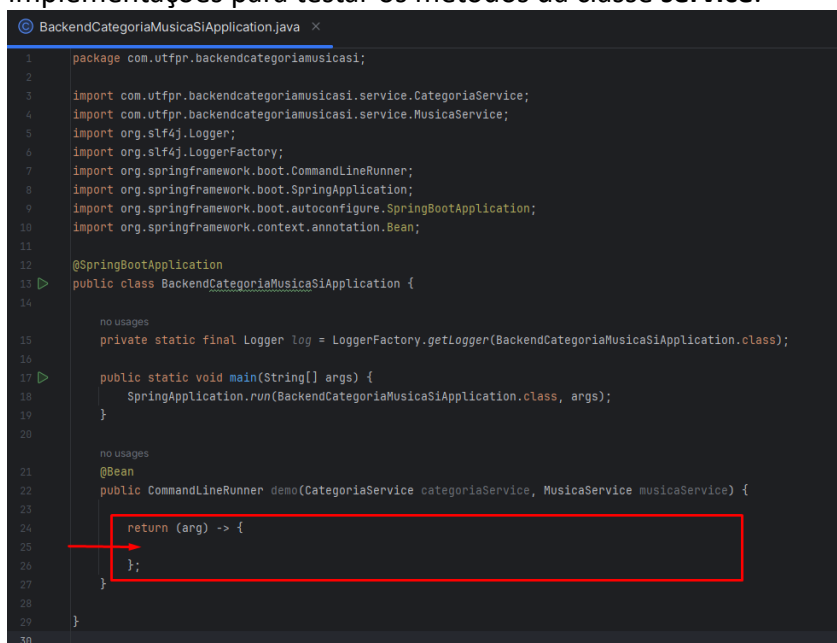


3. Incluir o código que consta na área demarcada na imagem acima, lembrar que é importante colocar como parâmetro as classes de serviço, nesse caso em específico são **CategoriaService** e **MusicaService**. Realizar também as importações necessárias.



```
1 package com.utfpr.backendcategoriamicasasi;
2
3 import com.utfpr.backendcategoriamicasasi.service.CategoriaService;
4 import com.utfpr.backendcategoriamicasasi.service.MusicaService;
5 import org.slf4j.Logger;
6 import org.slf4j.LoggerFactory;
7 import org.springframework.boot.CommandLineRunner;
8 import org.springframework.boot.SpringApplication;
9 import org.springframework.boot.autoconfigure.SpringBootApplication;
10 import org.springframework.context.annotation.Bean;
11
12 @SpringBootApplication
13 public class BackendCategoriaMusicaSiApplication {
14
15     no usages
16     private static final Logger log = LoggerFactory.getLogger(BackendCategoriaMusicaSiApplication.class);
17
18     public static void main(String[] args) {
19         SpringApplication.run(BackendCategoriaMusicaSiApplication.class, args);
20     }
21
22     no usages
23     @Bean
24     public CommandLineRunner demo(CategoriaService categoriaService, MusicaService musicaService) {
25
26         return (arg) -> {
27
28             };
29     }
30 }
```

4. Será na área demarcada na imagem abaixo que iremos inserir nossas implementações para testar os métodos da classe **service**.



```
1 package com.utfpr.backendcategoriamicasasi;
2
3 import com.utfpr.backendcategoriamicasasi.service.CategoriaService;
4 import com.utfpr.backendcategoriamicasasi.service.MusicaService;
5 import org.slf4j.Logger;
6 import org.slf4j.LoggerFactory;
7 import org.springframework.boot.CommandLineRunner;
8 import org.springframework.boot.SpringApplication;
9 import org.springframework.boot.autoconfigure.SpringBootApplication;
10 import org.springframework.context.annotation.Bean;
11
12 @SpringBootApplication
13 public class BackendCategoriaMusicaSiApplication {
14
15     no usages
16     private static final Logger log = LoggerFactory.getLogger(BackendCategoriaMusicaSiApplication.class);
17
18     public static void main(String[] args) {
19         SpringApplication.run(BackendCategoriaMusicaSiApplication.class, args);
20     }
21
22     no usages
23     @Bean
24     public CommandLineRunner demo(CategoriaService categoriaService, MusicaService musicaService) {
25
26         return (arg) -> {
27
28             };
29     }
30 }
```

5. Implemente as instruções de acordo com a imagem abaixo.

```
public static void main(String[] args) {
    SpringApplication.run(BackendCategoriaMusicaSiApplication.class, args);
}

no usages
@Bean
public CommandLineRunner demo(CategoriaService categoriaService, MusicaService musicaService) {

    return (arg) -> {

        1 {
            log.info("");
            log.info("");
            log.info("=====Listagem de todas as MUSICAS");
            for(Musica m : musicaService.listarTodasMusicas())
                log.info(m.toString());
        }

        2 {
            log.info("");
            log.info("");
            log.info("=====Listagem de todas as CATEGORIAS");
            for(Categoria c : categoriaService.listarTodasCategorias())
                log.info(c.toString());
        }
    };
}
```

Observe a parte 1 do print acima

log.info("=====Listagem de todas as MUSICAS");

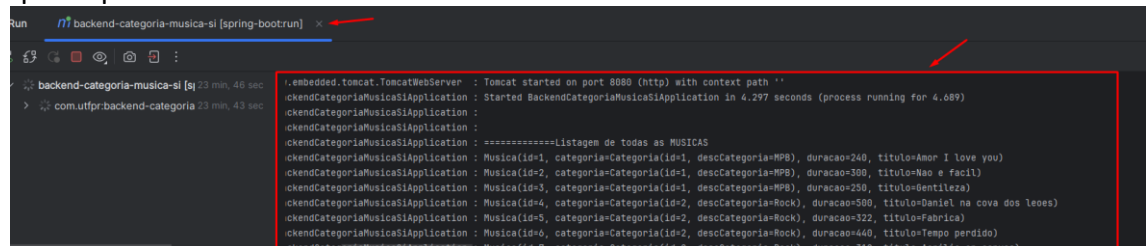
Essa instrução irá imprimir no console o texto "(=====Listagem de todas as MUSICAS"

musicaService.listarTodasMusicas()

O método **listarTodasMusicas()** está implementado na classe **MusicaService** e na implementação dele temos um **findAll()** por exemplo.

O mesmo se repete na parte 2, mas para **categorias**.

6. Rode a aplicação e verifique o resultado no Console. Nele podemos conferir se o que implementamos nas atividades está correto.



```
Run  backend-categoria-musica-si [spring-boot:run] x
back-end-categoria-musica-si [s] 23 min, 46 sec
com.utfrpr.backend-categoria 23 min, 43 sec

./embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path ''
ckendCategoriaMusicaSiApplication : Started BackendCategoriaMusicaSiApplication in 4.297 seconds (process running for 4.689)
ckendCategoriaMusicaSiApplication :
ckendCategoriaMusicaSiApplication : =====Listagem de todas as MUSICAS
ckendCategoriaMusicaSiApplication : Musica(id=1, categoria=Categoria(id=1, descCategoria=MFB), duracao=240, titulo=Amor I love you)
ckendCategoriaMusicaSiApplication : Musica(id=2, categoria=Categoria(id=1, descCategoria=MFB), duracao=300, titulo=Mao e facil)
ckendCategoriaMusicaSiApplication : Musica(id=3, categoria=Categoria(id=1, descCategoria=MFB), duracao=250, titulo=Gentileza)
ckendCategoriaMusicaSiApplication : Musica(id=4, categoria=Categoria(id=2, descCategoria=Rock), duracao=500, titulo=Daniel na cova dos leoes)
ckendCategoriaMusicaSiApplication : Musica(id=5, categoria=Categoria(id=2, descCategoria=Rock), duracao=322, titulo=Fabrica)
ckendCategoriaMusicaSiApplication : Musica(id=6, categoria=Categoria(id=2, descCategoria=Rock), duracao=440, titulo=Tempo perdido)
ckendCategoriaMusicaSiApplication : Musica(id=7, categoria=Categoria(id=2, descCategoria=Rock), duracao=312, titulo=Acrilic on canvas)
```