



Pós-Graduação Lato Sensu
Curso de Especialização em Tecnologia Java

APOSTILA - CRIAR PROJETO SPRING

Professor Hugo Baker Goveia
hbakergoveia@hotmail.com



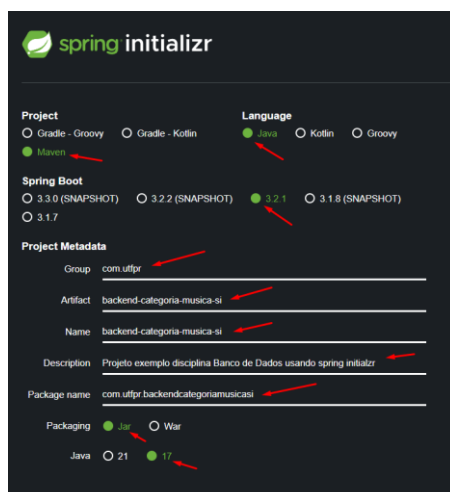
1. Criar projeto através do Spring Initializr

O **Spring Initializr** é uma ferramenta online que permite gerar um projeto inicial do Spring de forma rápida e fácil. Ele fornece uma interface web simples onde você pode configurar as dependências, versões e outras opções do projeto Spring. A versão do IntelliJ IDEA Ultimate possui um plugin que permite a criação de um projeto dentro da própria IDE, de forma que esse plugin incorpora as funções do Spring Initializr. A versão adotada na disciplina foi a Community Edition e ela não fornece esse plugin, por isso usaremos na disciplina o recurso do **Spring Initializr**.

Após configurar as opções desejadas, o Spring Initializr gera um arquivo **ZIP** contendo um projeto Spring Boot pré-configurado com as dependências e estrutura escolhidas. Você pode baixar esse arquivo ZIP e importá-lo na sua IDE de preferência (como Eclipse, IntelliJ IDEA ou Spring Tool Suite) para começar a desenvolver seu aplicativo Spring imediatamente.

O Spring Initializr simplifica o processo de criação de projetos Spring, fornecendo uma maneira rápida e padronizada de configurar um projeto com as dependências e configurações adequadas. Isso ajuda a acelerar o desenvolvimento e facilita a criação de aplicativos baseados no ecossistema do Spring.

1. Acesse o site oficial do Spring Initializr em <https://start.spring.io>.
2. Nessa parte do painel, escolher:
Project = **"Maven"**
Language = **"Java"**
Spring Boot = **"3.2.1"**
Group = **"com.utfpr"**
Artifact = **"backend-categoria-musica-si"**
Name = **"backend-categoria-musica-si"**
Description = **"Projeto exemplo disciplina Banco de Dados usando spring initializr"**
Package name = **"com.utfpr.backendcategoriamicasasi"**
Packaging = **"Jar"**
Java = **"17"**



The screenshot shows the Spring Initializr web interface with the following configuration:

- Project:** ☒ Maven
- Language:** ☒ Java
- Spring Boot:** ☒ 3.2.1
- Project Metadata:**
 - Group: com.utfpr
 - Artifact: backend-categoria-musica-si
 - Name: backend-categoria-musica-si
 - Description: Projeto exemplo disciplina Banco de Dados usando spring initializr
 - Package name: com.utfpr.backendcategoriamicasasi
- Packaging:** ☒ Jar
- Java:** ☒ 17

Sobre as informações que definimos acima:

"Project" (Tipo do projeto): Aqui informamos o tipo do projeto que vamos criar, nesse caso vamos criar um "Maven Project" (Projeto Maven), então escolhemos **"Maven"**.

"Language" (Linguagem): Foi selecionada "Java", que é a linguagem padrão do nosso curso para o backend.

"Spring Boot" (Versão do Spring Boot): Foi selecionada a "3.2.1", que é a versão adotada na disciplina.

"Group" (Grupo): Seria o domínio da empresa, nesse caso escolhemos como exemplo **"com.utfpr"**

"Artifact" (Artefato): Foi informado o nome do artefato, que é nome do projeto que é usado para nomear o arquivo JAR resultante, nesse exemplo vamos usar **backend-categoria-musica-si**.

"Name" (Nome do projeto): É o nome da sua aplicação, nesse exemplo usaremos **backend-categoria-musica-si**

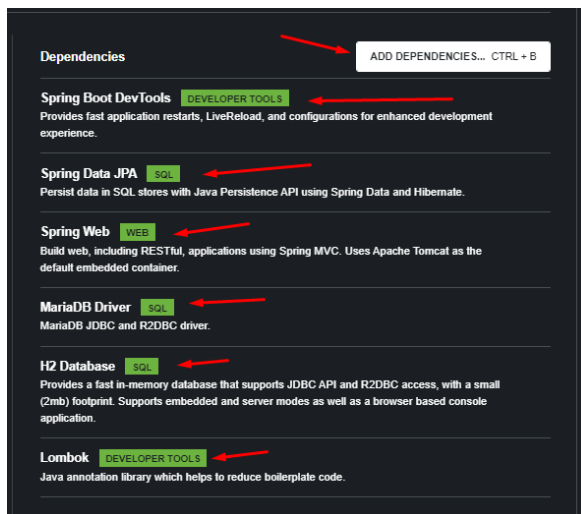
"Description" (Descrição): Aqui informamos uma breve descrição do projeto que vamos criar. Nesse exemplo usamos **"Projeto exemplo disciplina Banco de Dados usando spring initializr"**

"Package name" (Nome do Pacote): Aqui é informado o pacote que irá iniciar do projeto, o pacote raiz, nesse projeto usaremos como exemplo **"com.utfpr.backendcategoriamicasi"**.

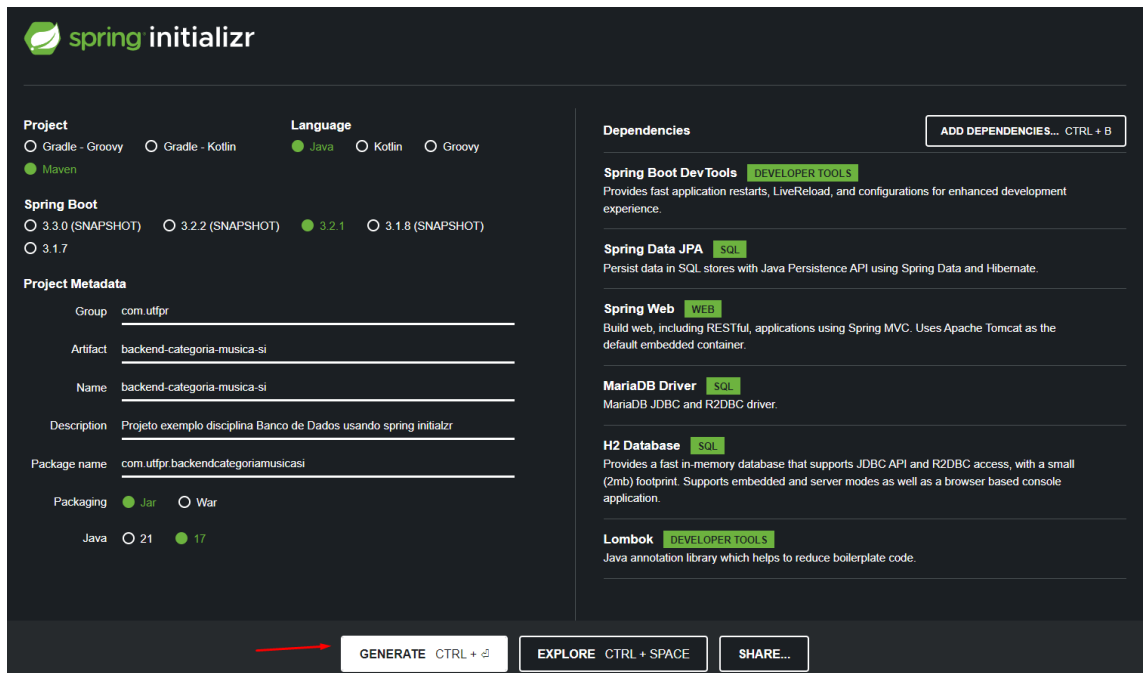
"Packaging" (Empacotamento): Aqui nós escolhemos qual é a forma de empacotamento de nossa aplicação, o padrão é o **jar**, que é onde o tomcat estará embutido. Então foi selecionado **"Jar"**.

"Java Version" (Versão do Java): Foi selecionada a "17", que é a versão adotada na disciplina.

3. Nessa parte do painel, escolher as dependências clicando em **ADD DEPENDENCIES:**



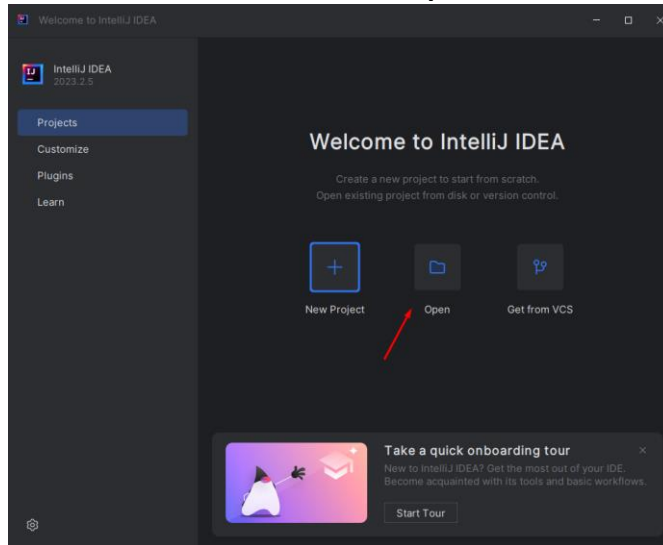
4. Clicar no **GENERATE**



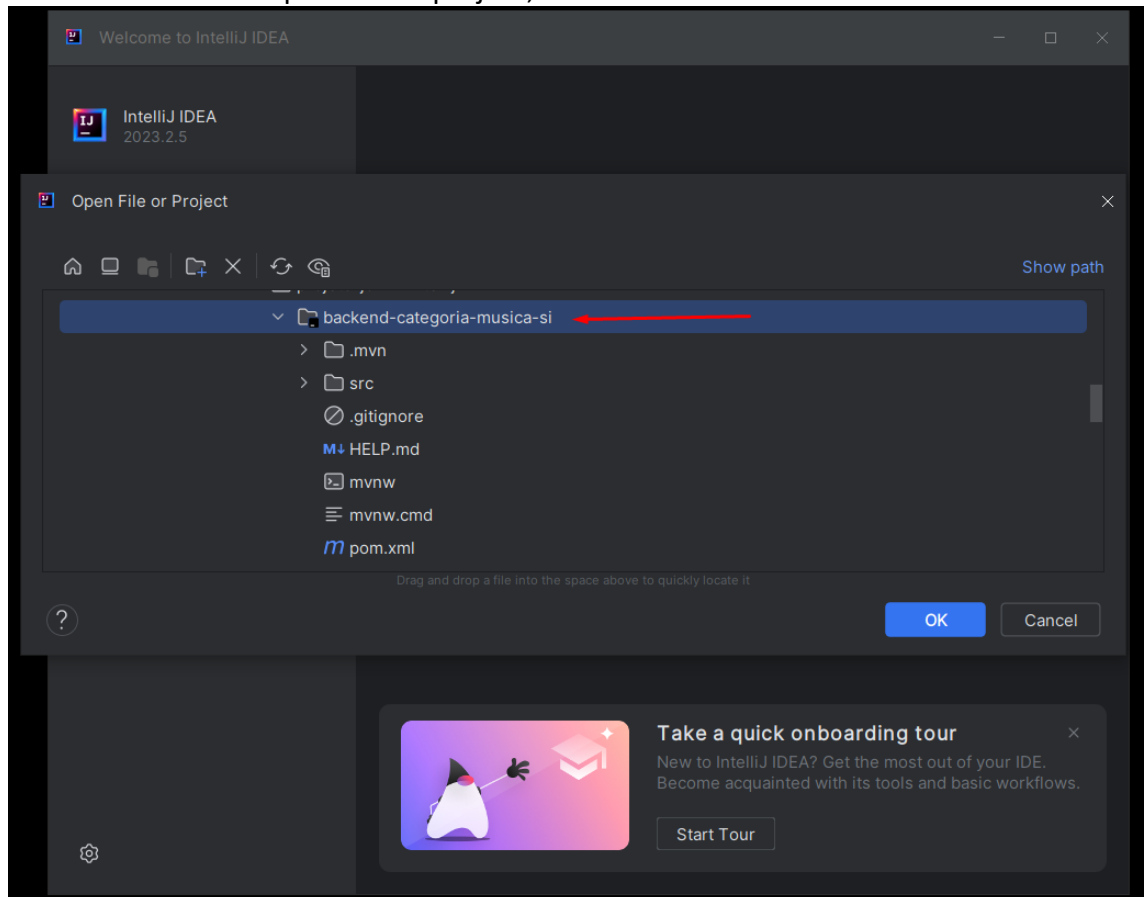
O Spring Initializr irá gerar um arquivo ZIP contendo o projeto inicial com base nas configurações que você selecionou.

5. Baixe o arquivo ZIP gerado e extraia-o para um diretório de sua escolha.
6. Importe o projeto na sua IDE preferida, como por exemplo o **IntelliJ** ou **Spring Tool Suite** ambas recomendadas nessa disciplina. Cada IDE tem um processo específico para importar projetos, portanto, siga as instruções da sua IDE para importar corretamente o projeto. Na etapa a seguir mostraremos como é feito no IntelliJ CE.

7. Para abrir no IntelliJ, clicar em **Open**



8. Entrar no diretório que está o projeto, **selecionar** o diretório e clicar no **OK**

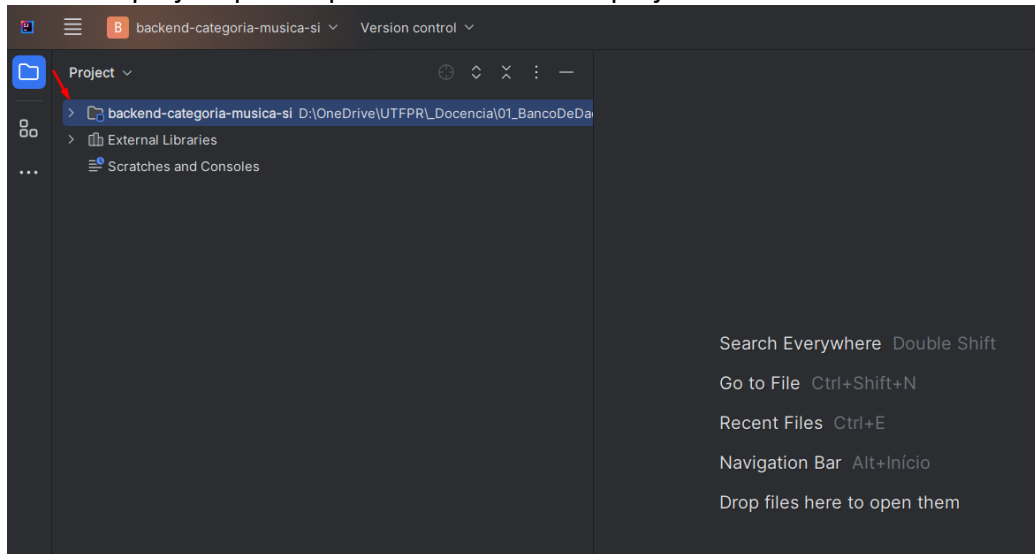


9. Após abrir o projeto, você pode começar a desenvolver o seu aplicativo Spring de acordo com a evolução da disciplina.
Você poderá adicionar as classes, serviços, repositórios, entre outros.

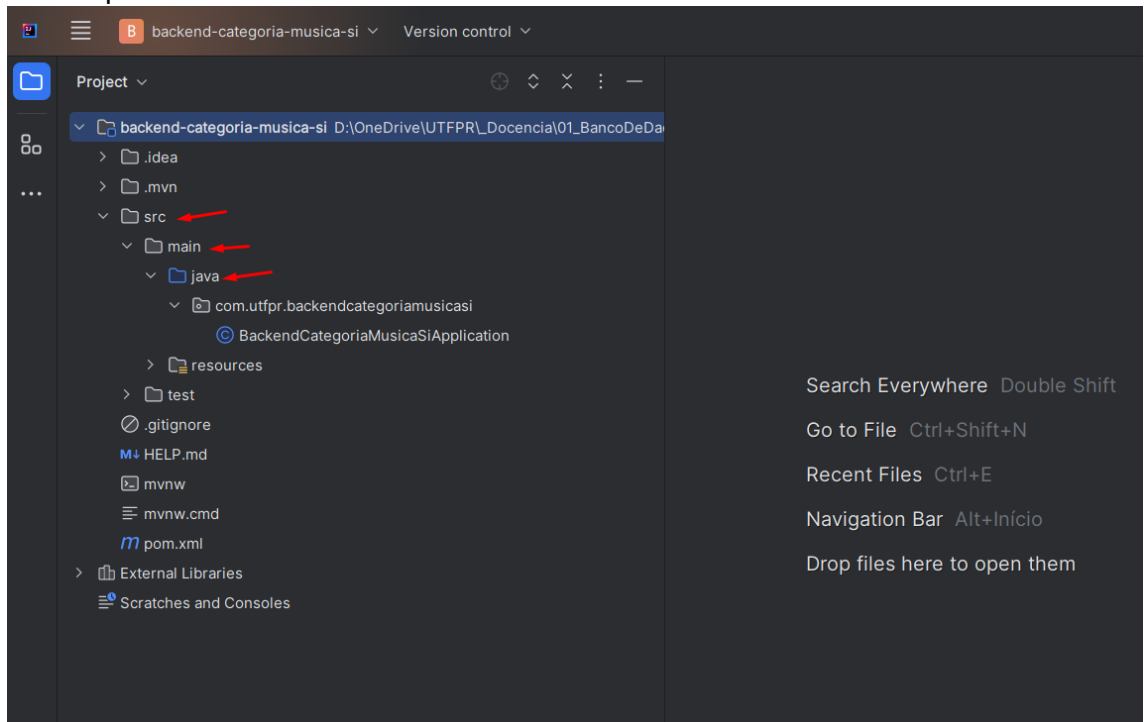
2. Estrutura inicial do projeto

Nas etapas abaixo serão apresentados para vocês os principais pontos em uma estrutura de um projeto MAVEN.

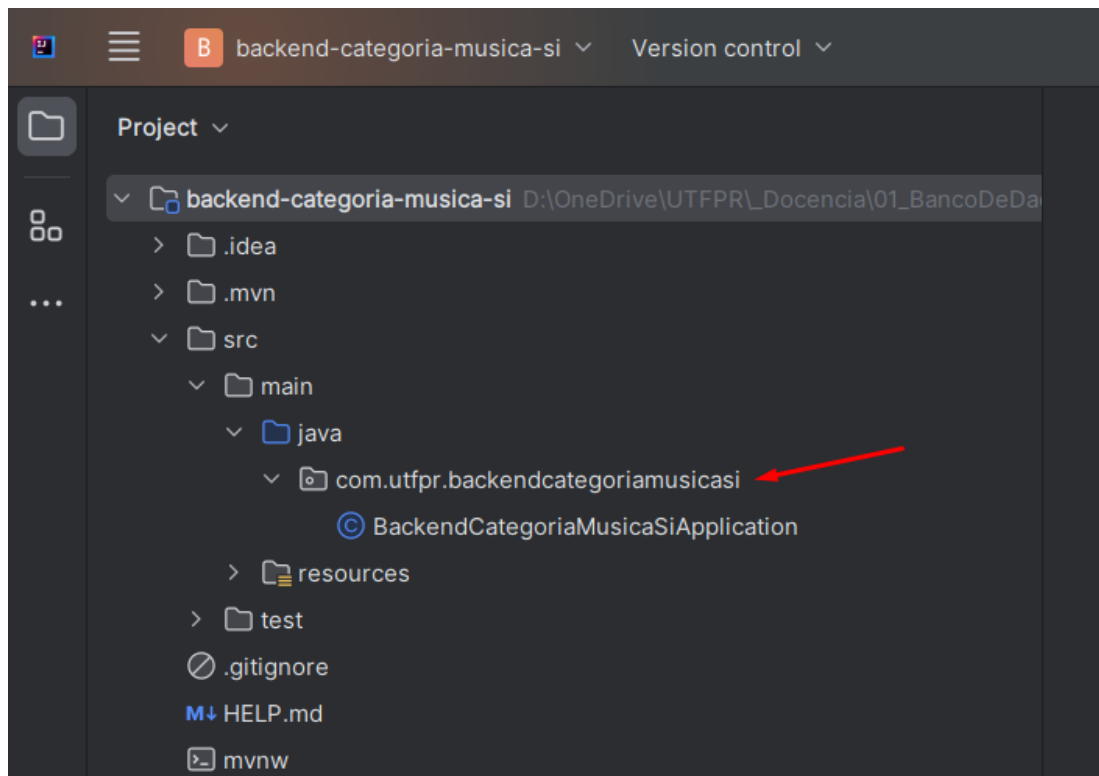
1. Após criado o projeto, você deverá clicar na seta que se encontra a frente do nome do projeto para expandir a estrutura do projeto.



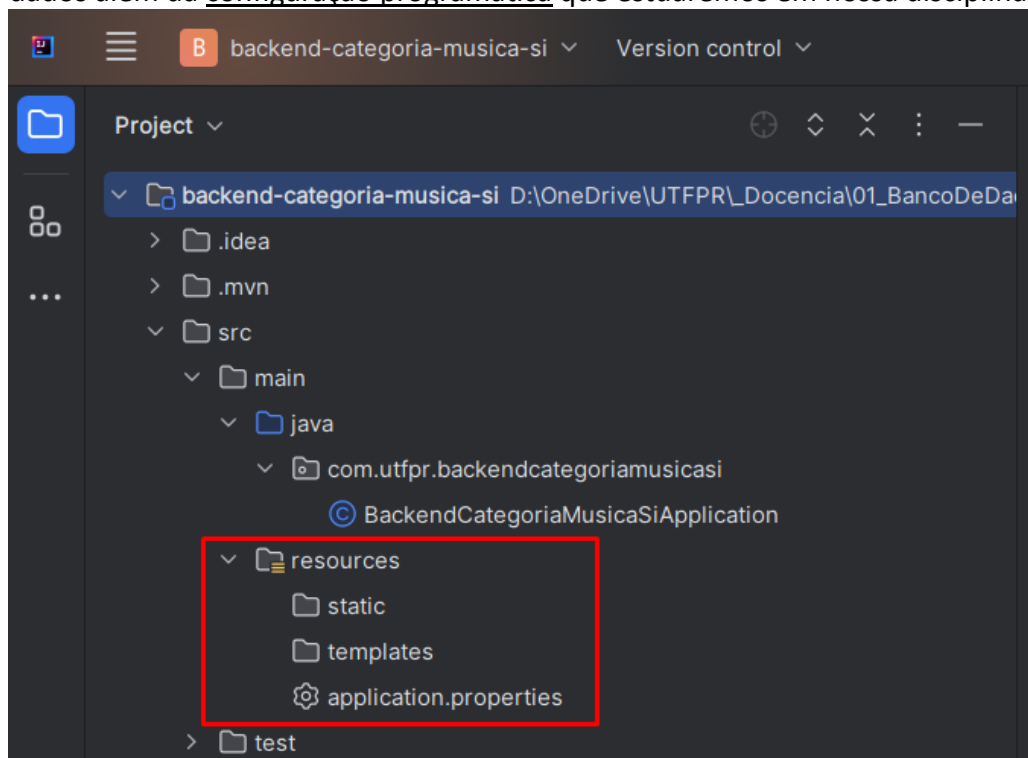
2. **src/main/java**, é onde encontra-se o código fonte do nosso projeto, contendo as regras de negócio, ou seja, tudo que vamos implementar de fato nas atividades da disciplina.



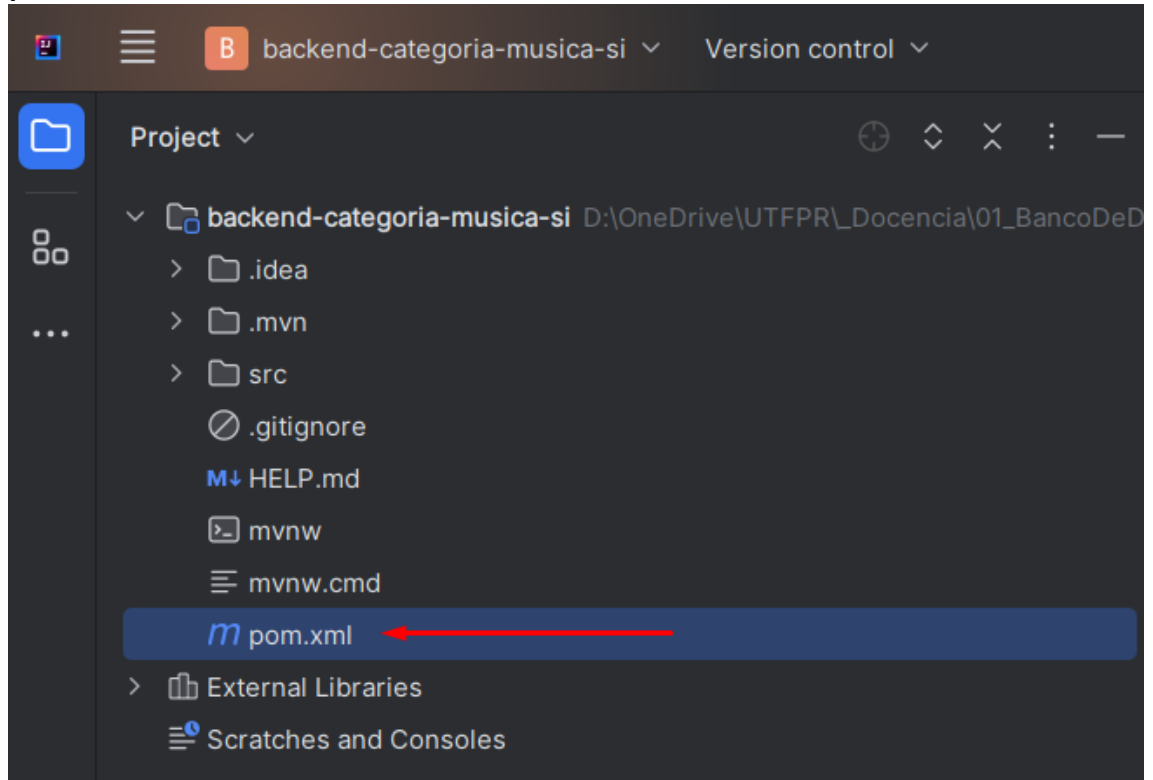
3. **com.utfpr.backendcategoriamicasi**, é o pacote raiz de nossa estrutura do projeto, será a partir dele que iremos criar os demais pacotes usados no projeto como por exemplo **Config**, **Entity**, **Repository** e **Service**.



4. **src/main/resources**, é onde colocamos arquivos de configuração, imagens, HTML, etc. É também onde encontramos o **application.properties**, que podemos usar como forma alternativa de fazer a configuração da conexão com o banco de dados além da configuração programática que estuaremos em nossa disciplina.



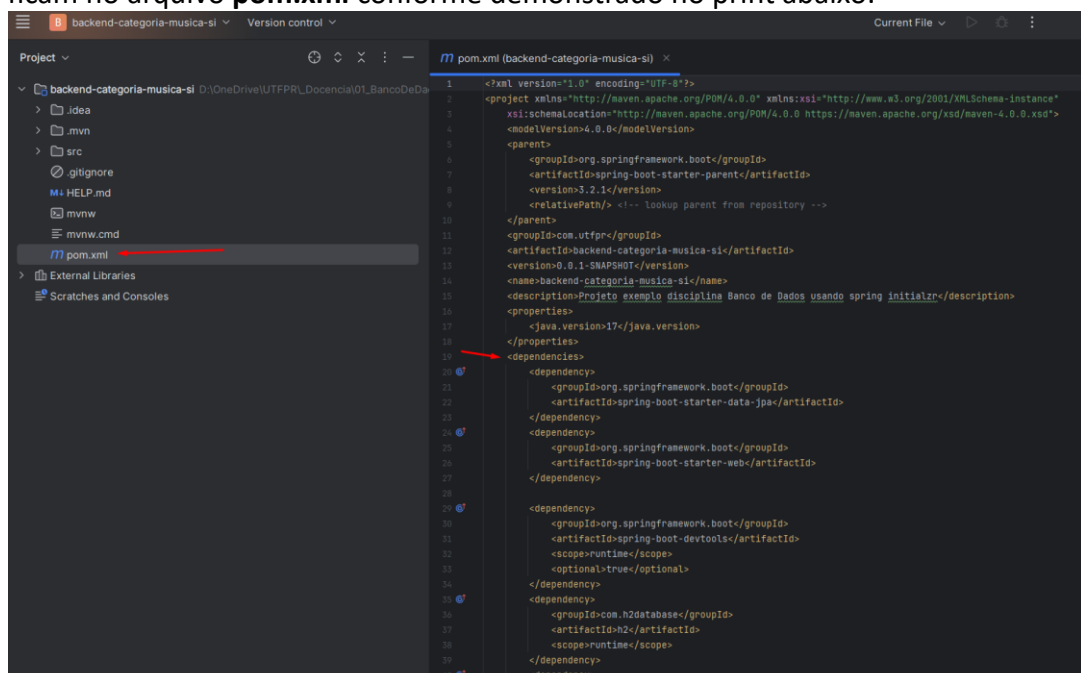
5. pom.xml



O arquivo **pom.xml** é criado na raiz do projeto e é a abreviatura da sigla *Project Object Model*. É nele que se encontram as informações do MAVEN no projeto. O pom.xml é o core do maven dentro do nosso projeto, pois é nele que ficam as configurações.

Ele é um arquivo **XML** e por isso segue a regra de formatação desse tipo de arquivo, composto por TAGS.

É importante sabermos também que todas as dependências que escolhemos ficam no arquivo **pom.xml** conforme demonstrado no print abaixo:



Isso significa que podemos usar esse arquivo para incluir, alterar a versão ou até mesmo excluir uma dependência para nosso projeto.

BREVE RESUMO DAS DEPENDÊNCIAS SELECIONADAS:

- **Sobre o Lombok**

O Lombok simplificará a criação de classes Java com recursos como geração automática de **getters/setters** e **construtores**.

- **Sobre o Spring Web**

O Spring Web fornecerá recursos para criar endpoints REST, caso você queira evoluir a aplicação da disciplina para uma API-REST.

- **Sobre o Spring Boot DevTools**

O Spring Boot DevTools oferecerá recursos de desenvolvimento, como reinicialização automática do aplicativo durante o desenvolvimento.

- **Sobre o MariaDB**

O MariaDB Driver permitirá que você se conecte ao banco de dados MariaDB.

- **Sobre o H2**

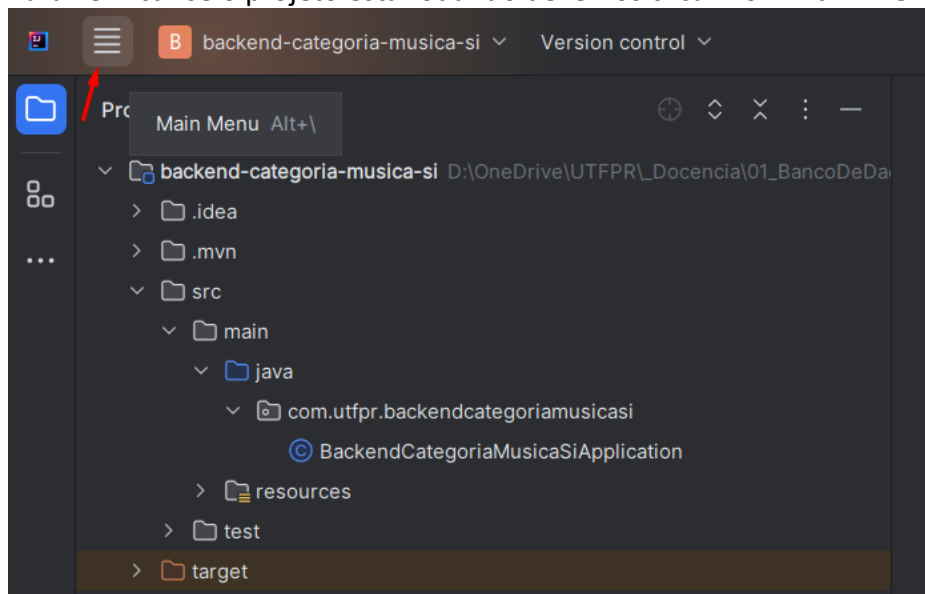
O H2 Database permitirá que você se conecte ao banco de dados em memória H2, que usaremos para testes se necessário.

- **Spring Data JPA**

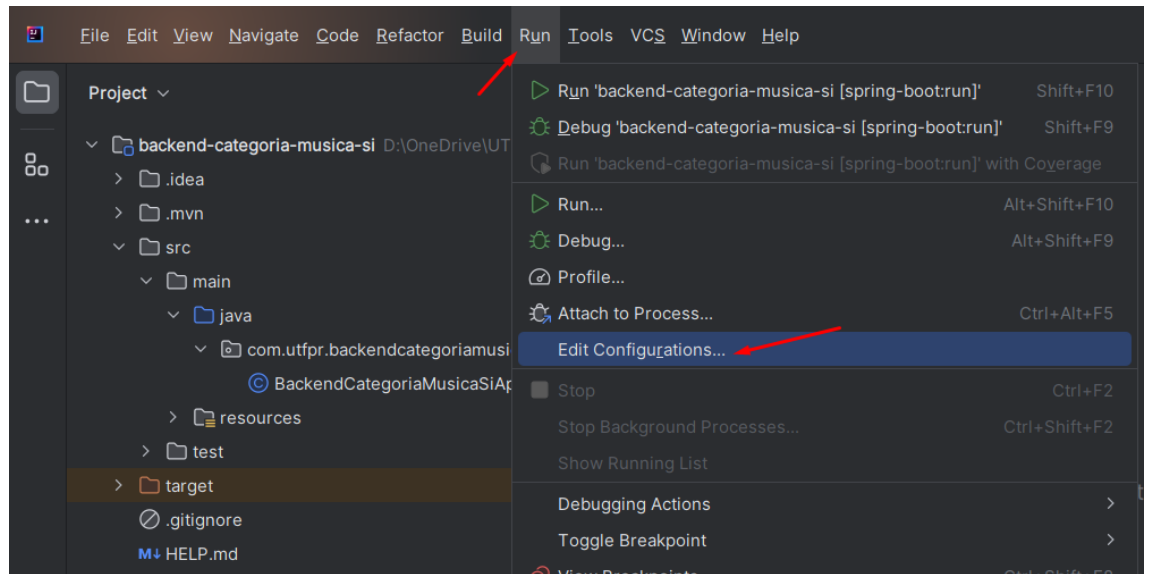
O Spring Data JPA fornecerá recursos para trabalhar com persistência de dados usando a JPA (Java Persistence API).

3. Executar o projeto

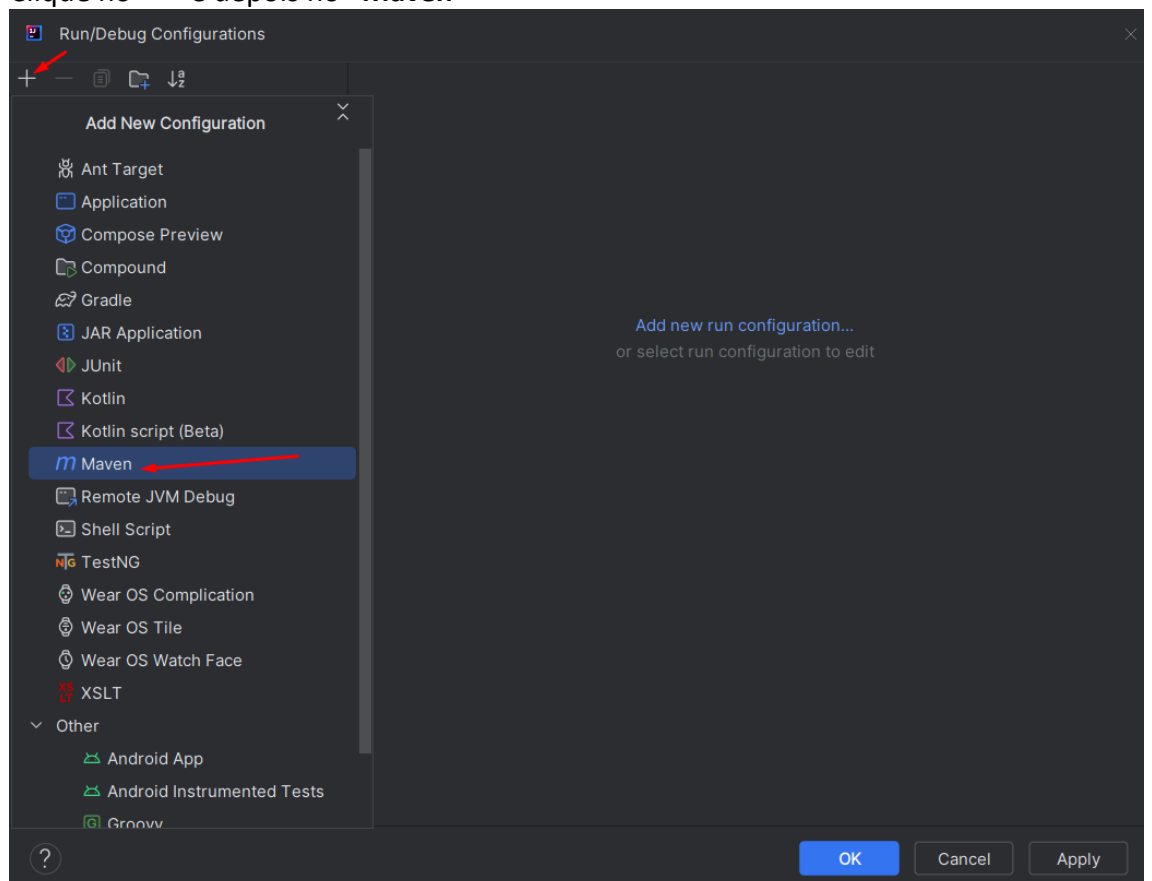
1. Para verificar se o projeto está rodando devemos clicar no **“Main Menu”**



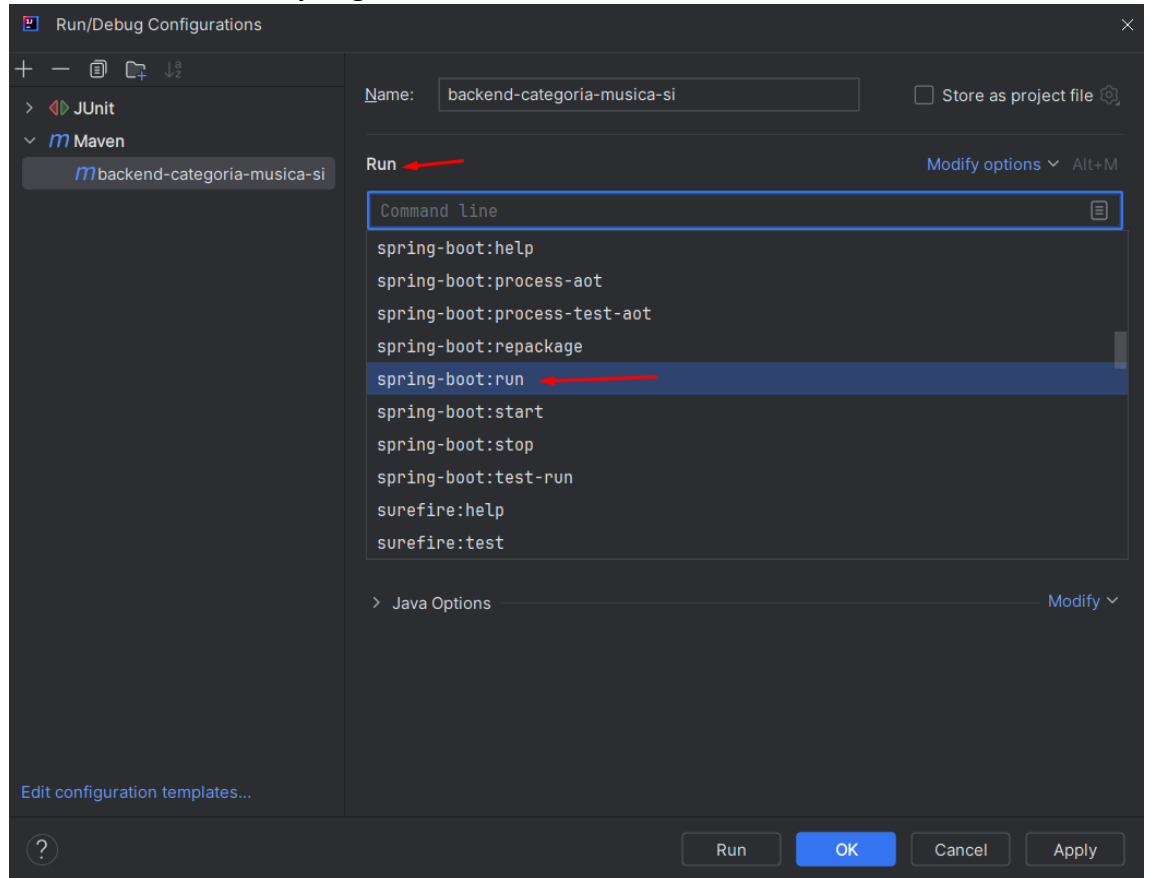
2. Clicar no **“Run->Edit Configurations...”** para configurar como o projeto irá ser executado.



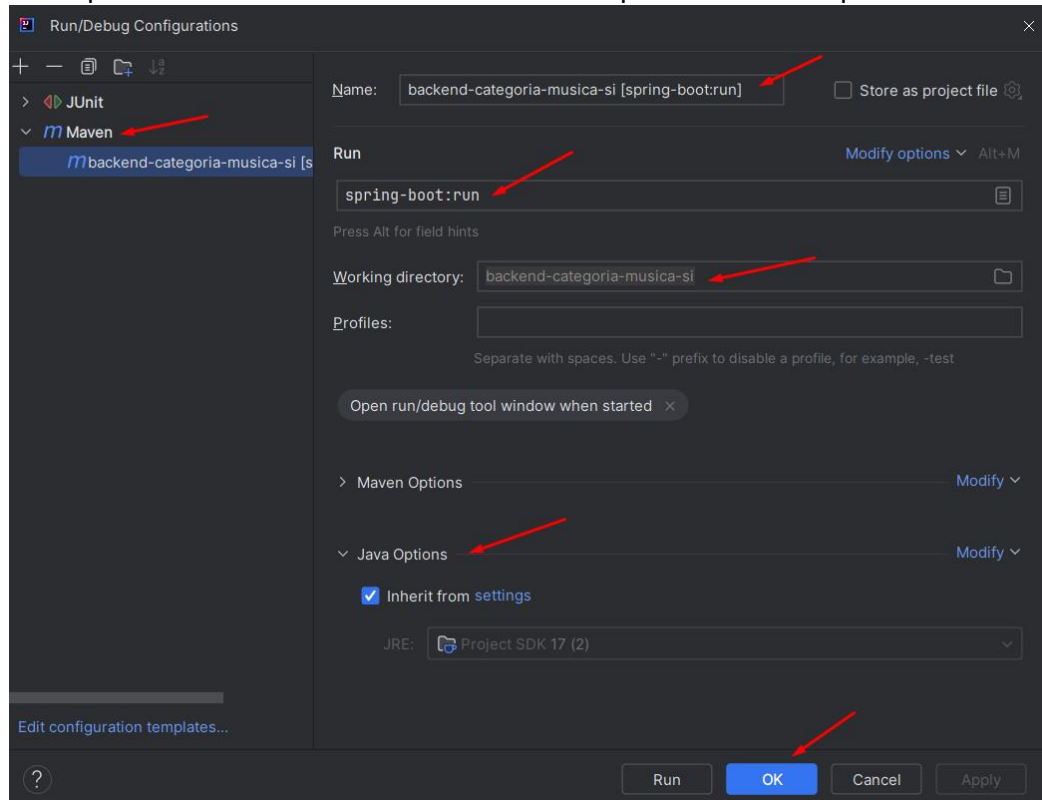
3. Clique no **“+”** e depois no **“Maven”**



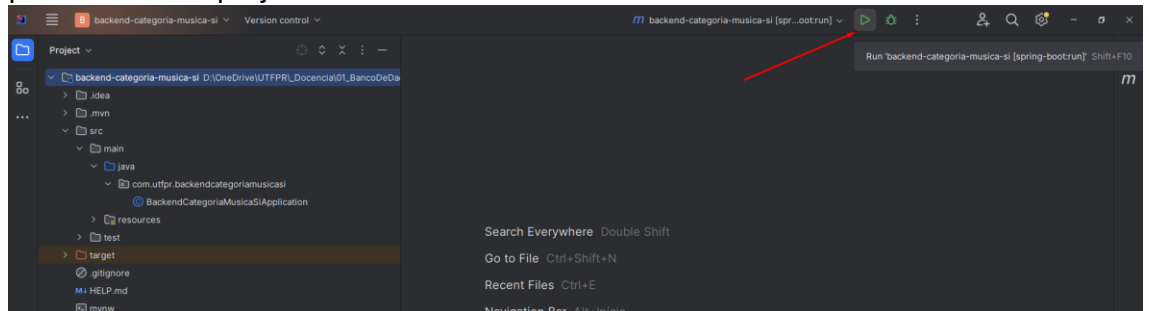
4. No “Run” escolha o “spring-boot:run”



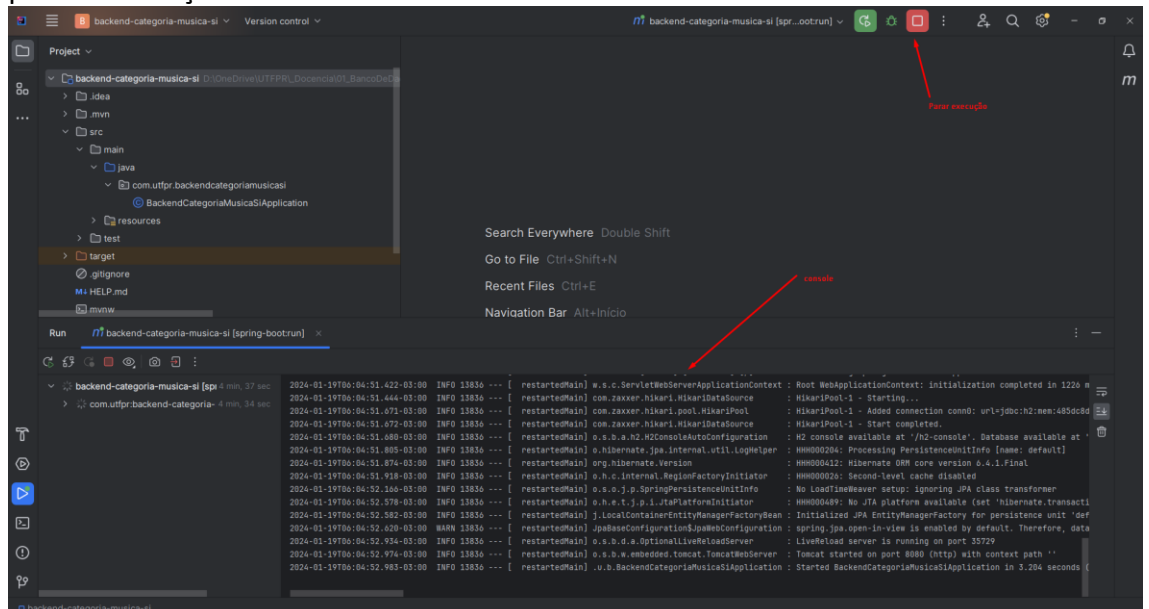
5. Verifique se está tudo correto de acordo com o print abaixo e clique em “OK”



6. Clique no botão **“Run”** para executar o projeto. Os passos anteriores usados para configuração da execução são necessários apenas uma vez. Depois de feita a configuração você pode clicar sempre no **“Run”** seguindo a partir daqui para executar o projeto.

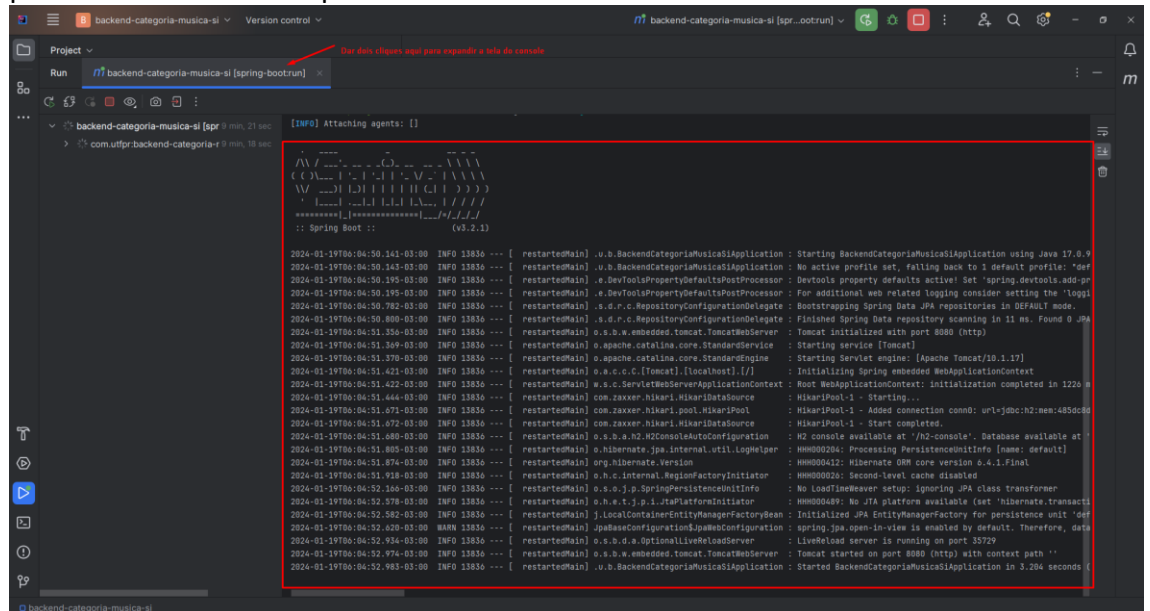


7. Observe que abriu o **“console”** e temos também o botão para **“Stop”** para parar a execução.

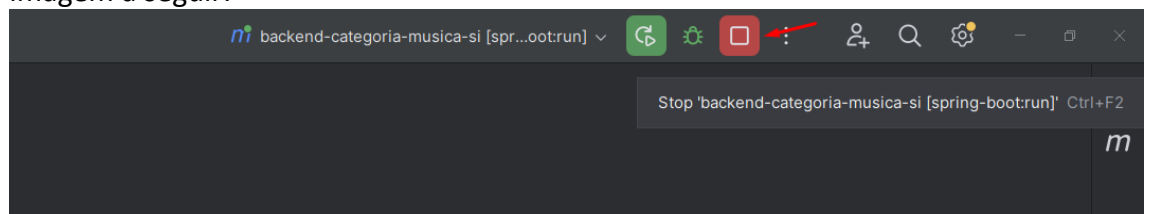


Se no console aparecer algum tipo de mensagem contendo a palavra **“Error”** logo perceberemos que nossa aplicação não deu **build** e não **rodou** corretamente.

8. Caso você queira visualizar o Log completo do console, você pode navegar na barra de rolagem do próprio console, ou dar dois cliques com o mouse na aba **“console”**, após isso a tela do console é expandida para ocupar a maior área possível da IDE. Para voltar na visualização parcial do console, basta repetir o processo dando dois cliques novamente na aba **“console”**.

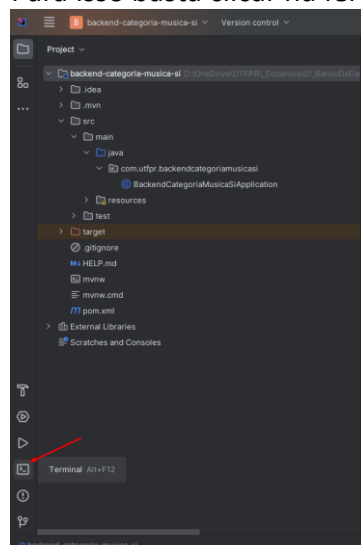


9. Para encerrar a aplicação, basta clicar em **no** botão **“Stop”** indicado na imagem a seguir:

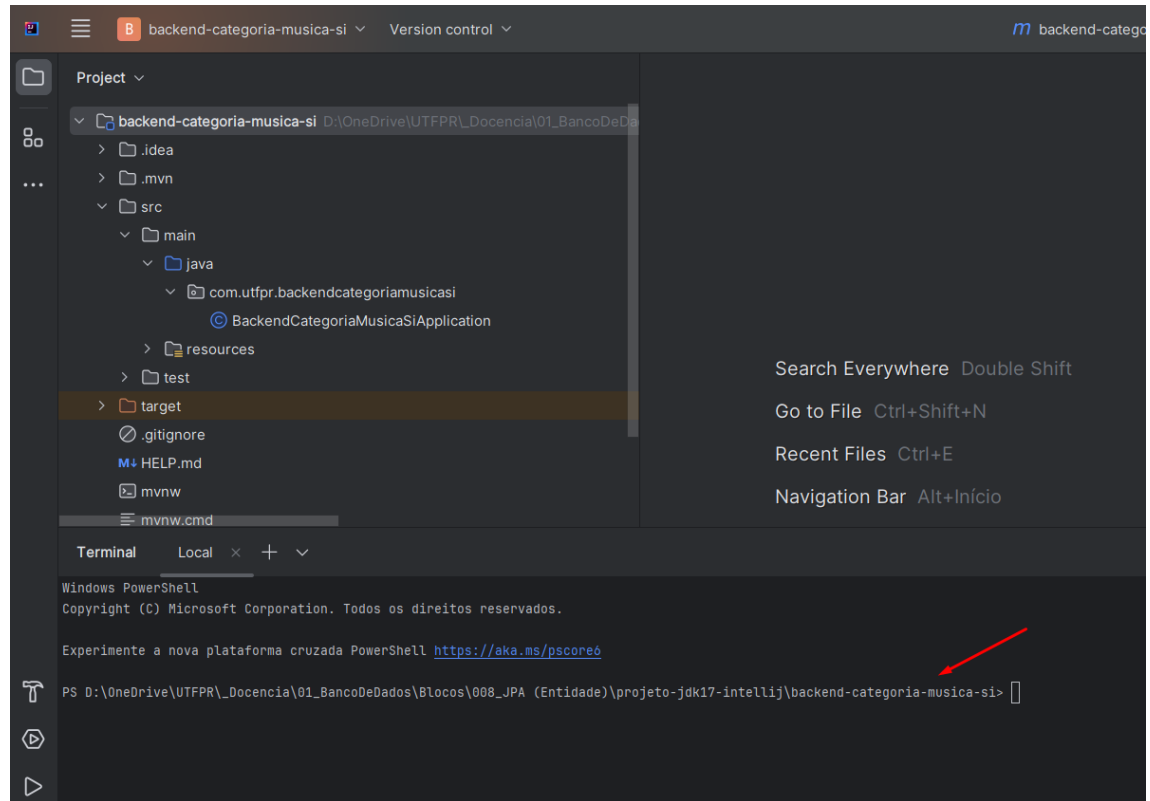


10. **Outra forma** de executar o projeto é usando a linha de comando abrindo o terminal da IDE:

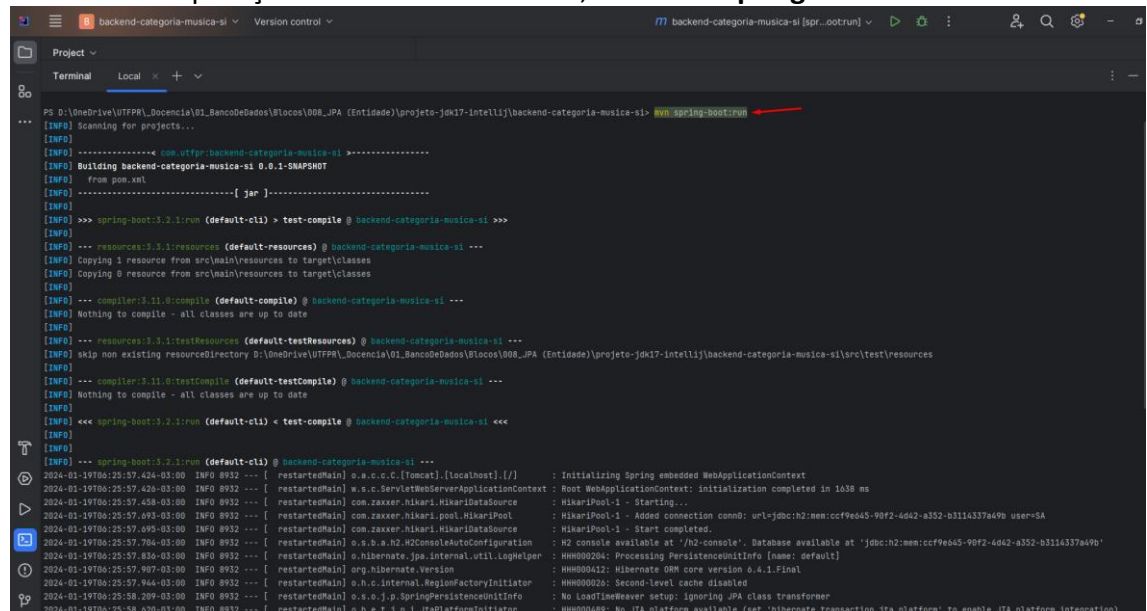
- a. Para isso basta clicar na ferramenta **“Terminal”**



- b. Por padrão ele abre no diretório do projeto, mas caso não aconteça, basta acessar o diretório do projeto usando o comando **cd <caminho-do-diretório>** na linha de comando CLI.



- c. Para rodar a aplicação via linha de comando, usar **mvn spring-boot:run**



```
[INFO] >>> spring-boot:3.2.1:run (default-cli) > test-compile @ backend-categoria-musica-s1 >>>
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ backend-categoria-musica-s1 ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 0 resource from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ backend-categoria-musica-s1 ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ backend-categoria-musica-s1 ---
[INFO] skip non existing resourceDirectory D:\OneDrive\UTPPW_Bocencia\01_BancoDeDados\Blocos\008_JPA (Entidade)\projeto-jdk17-intellij\backend-categoria-musica-s1\src\test\resources
[INFO]
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ backend-categoria-musica-s1 ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] <<< spring-boot:3.2.1:run (default-cli) < test-compile @ backend-categoria-musica-s1 <<<
[INFO]
[INFO] --- spring-boot:3.2.1:run (default-cli) @ backend-categoria-musica-s1 ---
2024-01-19T06:25:57.424-03:00 INFO 8932 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2024-01-19T06:25:57.426-03:00 INFO 8932 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1638 ms
2024-01-19T06:25:57.448-03:00 INFO 8932 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2024-01-19T06:25:57.693-03:00 INFO 8932 --- [ restartedMain] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection conn0: url=jdbc:h2:mem:ccf9e645-90f2-4d42-a352-b3114337a49b user=SA
2024-01-19T06:25:57.695-03:00 INFO 8932 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2024-01-19T06:25:57.704-03:00 INFO 8932 --- [ restartedMain] o.s.b.a.h2.H2ConsoleAutoConfiguration : H2 console available at '/h2-console'. Database available at 'jdbc:h2:mem:ccf9e645-90f2-4d42-a352-b3114337a49b'
2024-01-19T06:25:57.836-03:00 INFO 8932 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HH0000204: Processing PersistenceUnitInfo [name: default]
2024-01-19T06:25:57.907-03:00 INFO 8932 --- [ restartedMain] org.hibernate.Version : HH0000412: Hibernate ORM core version 6.4.1.Final
2024-01-19T06:25:57.944-03:00 INFO 8932 --- [ restartedMain] o.h.e.t.j.p.i.RegionFactoryInitiator : HH0000026: Second-level cache disabled
2024-01-19T06:25:58.209-03:00 INFO 8932 --- [ restartedMain] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup; ignoring JPA class transformer
2024-01-19T06:25:58.620-03:00 INFO 8932 --- [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HH0000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
2024-01-19T06:25:58.624-03:00 INFO 8932 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2024-01-19T06:25:58.666-03:00 WARN 8932 --- [ restartedMain] jpahibernate.jpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Exp
icly configure spring.jpa.open-in-view to disable this warning
2024-01-19T06:25:58.973-03:00 INFO 8932 --- [ restartedMain] o.s.b.a.o.OptionalLiveReloadServer : LiveReload server is running on port 35729
2024-01-19T06:25:59.004-03:00 INFO 8932 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path ''
2024-01-19T06:25:59.016-03:00 INFO 8932 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Started BackendCategoriaMusicaS1Application in 4.619 seconds (process running for 7.012)
2024-01-19T06:25:59.016-03:00 INFO 8932 --- [ restartedMain] .u.b.BackendCategoriaMusicaS1Application : Started BackendCategoriaMusicaS1Application in 4.619 seconds (process running for 7.012)
```

d. Clique no console e pressione **CTRL + C** para encerrar a execução

```
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ backend-categoria-musica-s1 ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] <<< spring-boot:3.2.1:run (default-cli) < test-compile @ backend-categoria-musica-s1 <<<
[INFO]
[INFO] --- spring-boot:3.2.1:run (default-cli) @ backend-categoria-musica-s1 ---
2024-01-19T06:25:57.424-03:00 INFO 8932 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2024-01-19T06:25:57.426-03:00 INFO 8932 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1638 ms
2024-01-19T06:25:57.448-03:00 INFO 8932 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2024-01-19T06:25:57.693-03:00 INFO 8932 --- [ restartedMain] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection conn0: url=jdbc:h2:mem:ccf9e645-90f2-4d42-a352-b3114337a49b user=SA
2024-01-19T06:25:57.695-03:00 INFO 8932 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2024-01-19T06:25:57.704-03:00 INFO 8932 --- [ restartedMain] o.s.b.a.h2.H2ConsoleAutoConfiguration : H2 console available at '/h2-console'. Database available at 'jdbc:h2:mem:ccf9e645-90f2-4d42-a352-b3114337a49b'
2024-01-19T06:25:57.836-03:00 INFO 8932 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HH0000204: Processing PersistenceUnitInfo [name: default]
2024-01-19T06:25:57.907-03:00 INFO 8932 --- [ restartedMain] org.hibernate.Version : HH0000412: Hibernate ORM core version 6.4.1.Final
2024-01-19T06:25:57.944-03:00 INFO 8932 --- [ restartedMain] o.h.e.t.j.p.i.RegionFactoryInitiator : HH0000026: Second-level cache disabled
2024-01-19T06:25:58.209-03:00 INFO 8932 --- [ restartedMain] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup; ignoring JPA class transformer
2024-01-19T06:25:58.620-03:00 INFO 8932 --- [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HH0000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
2024-01-19T06:25:58.624-03:00 INFO 8932 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2024-01-19T06:25:58.666-03:00 WARN 8932 --- [ restartedMain] jpahibernate.jpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Exp
icly configure spring.jpa.open-in-view to disable this warning
2024-01-19T06:25:58.973-03:00 INFO 8932 --- [ restartedMain] o.s.b.a.o.OptionalLiveReloadServer : LiveReload server is running on port 35729
2024-01-19T06:25:59.004-03:00 INFO 8932 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path ''
2024-01-19T06:25:59.016-03:00 INFO 8932 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Started BackendCategoriaMusicaS1Application in 4.619 seconds (process running for 7.012)
2024-01-19T06:28:31.179-03:00 INFO 8932 --- [ shutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
2024-01-19T06:28:31.180-03:00 INFO 8932 --- [ shutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
[INFO] -----
[INFO] Total time: 02:43 min
[INFO] Finished at: 2024-01-19T06:28:31-03:00
[INFO] -----
Deseja finalizar o arquivo em lote (S/N)? s
PS D:\OneDrive\UTPPW_Bocencia\01_BancoDeDados\Blocos\008_JPA (Entidade)\projeto-jdk17-intellij\backend-categoria-musica-s1>
```