

# Spring Boot

Criando uma aplicação que acessa banco de dados



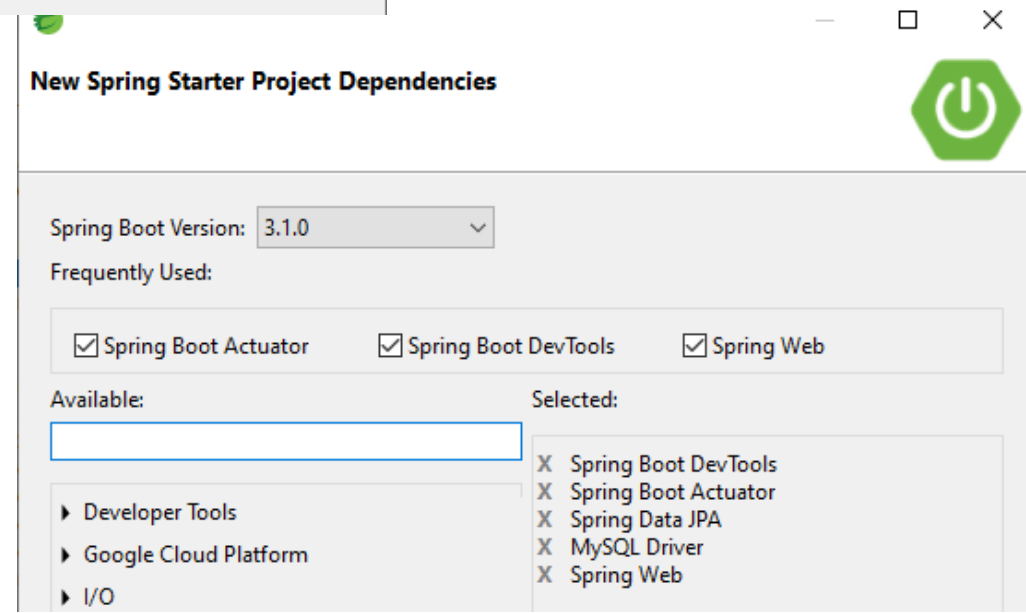
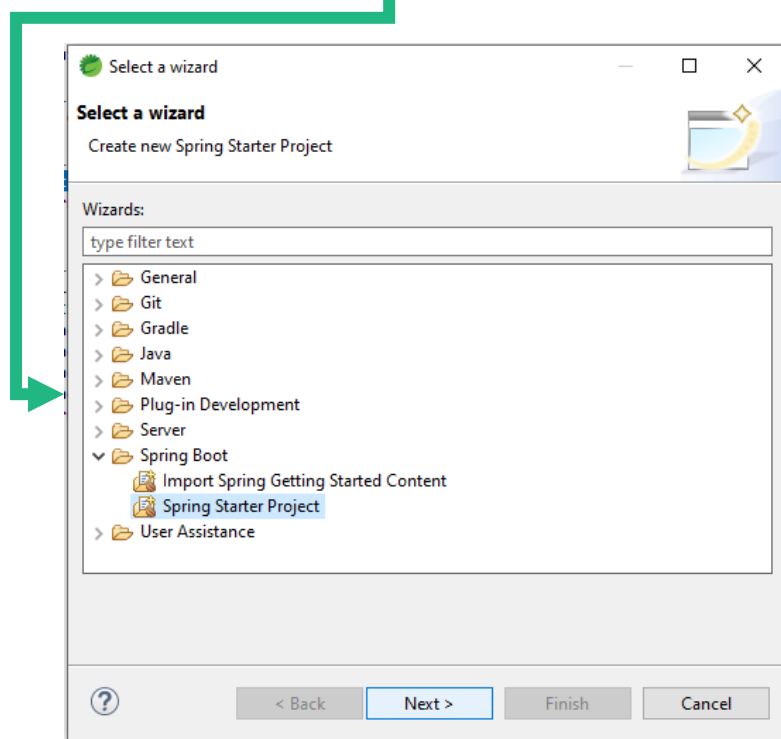
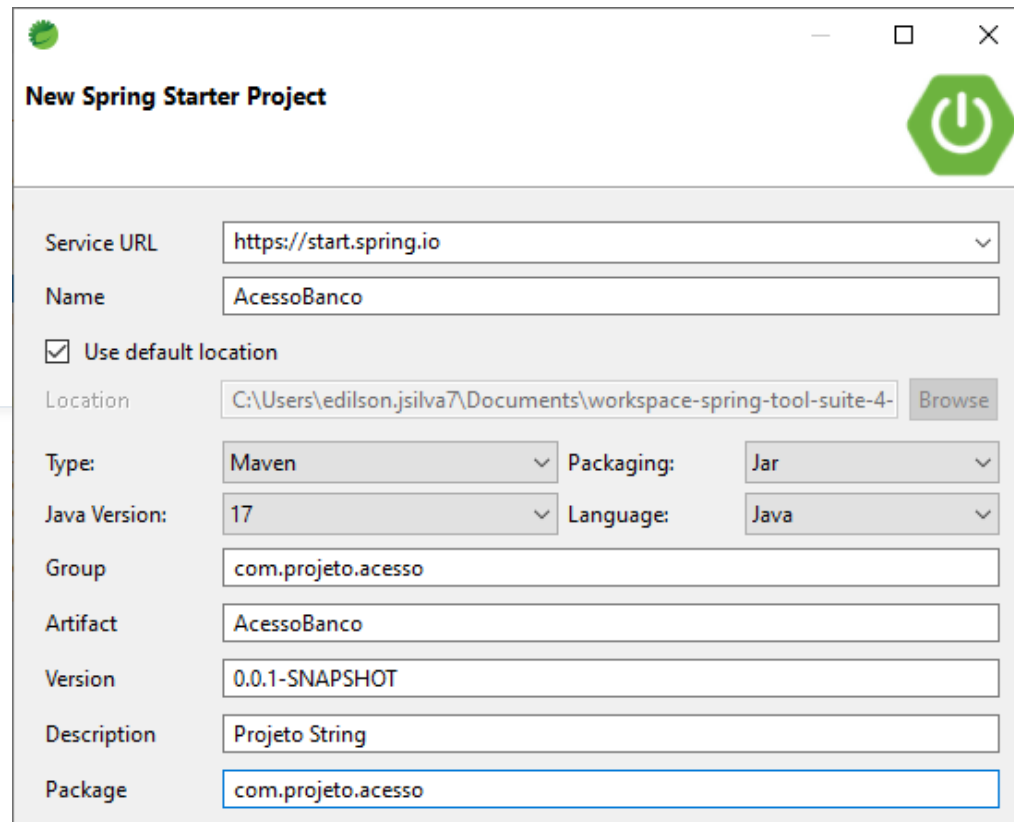
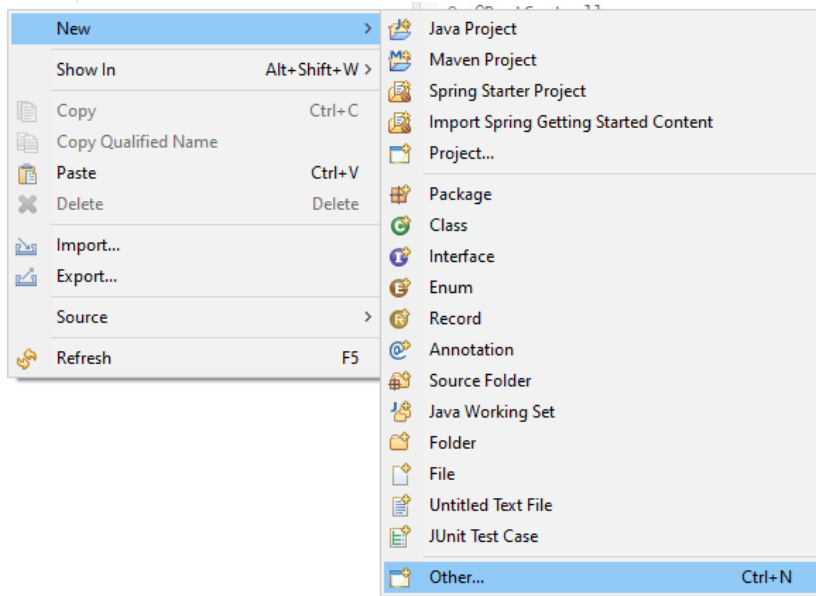
# Spring Boot

Crie um novo projeto no STS (Spring Tools Suite) com o nome de AcessoBanco. Adicione as seguintes dependências:

- Spring Web;
- Spring Boot Dev Tools;
- Spring Boot Actuator;
- Spring Data JPA;
- MySQL Driver

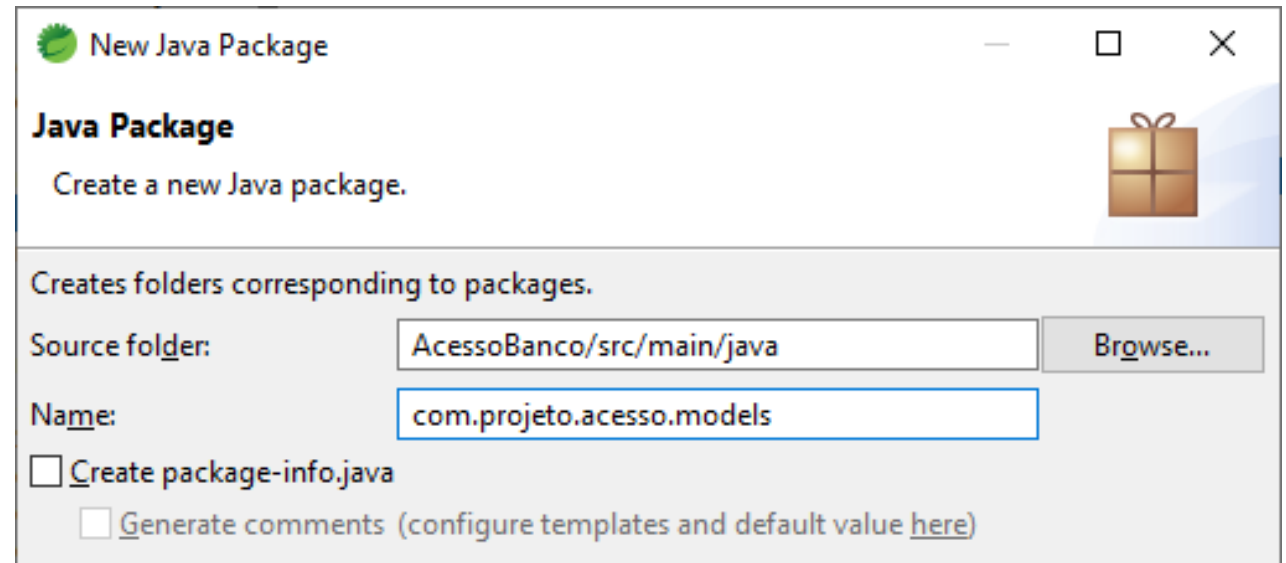
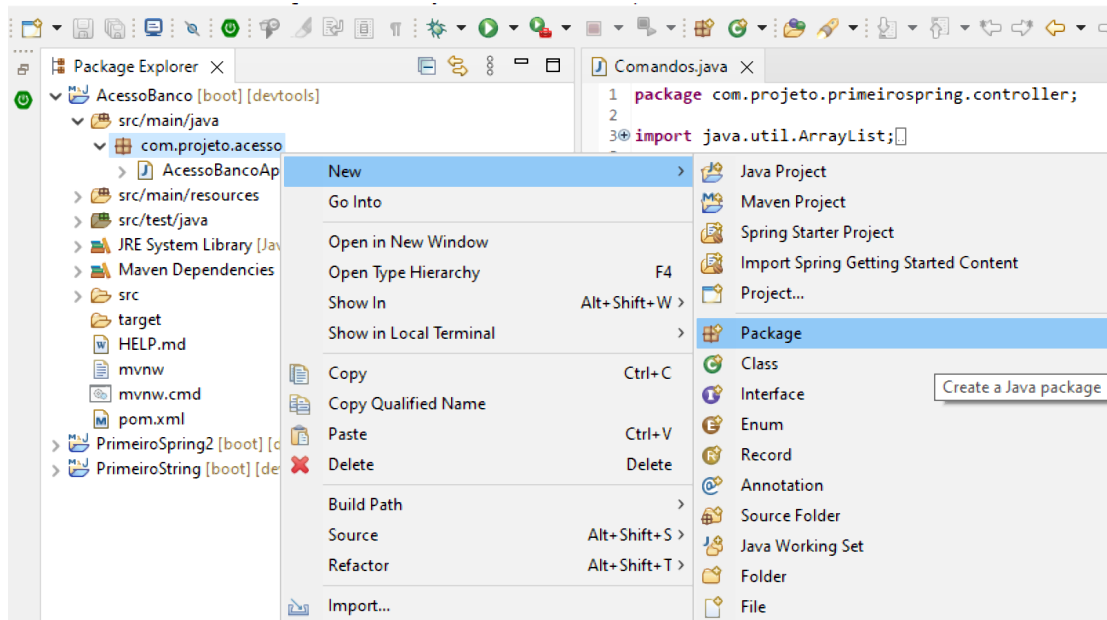
---

Veja no próximo slide



# Criação do package models

- Depois de criar o projeto, crie um novo package chamados models no package principal. Veja abaixo



# Criação do package models

- No package criado, models, crie uma classe com o nome de Clientes

The screenshot displays the Eclipse IDE interface. On the left, the Package Explorer shows the project structure: 'AcessoBanco [boot] [devtools]' with sub-packages 'src/main/java' and 'com.projeto.acesso'. The 'com.projeto.acesso.models' package is selected, and a context menu is open with 'New' > 'Class' chosen. A tooltip 'Create a Java class' is visible over the 'Class' option. On the right, the 'New Java Class' dialog is open. It shows the 'Source folder' as 'AcessoBanco/src/main/java' and the 'Package' as 'com.projeto.acesso.models'. The 'Name' field contains 'Clientes'. The 'Modifiers' section has 'public' selected. The 'Superclass' is set to 'java.lang.Object'. The 'Interfaces' section is empty.

**Package Explorer**

- AcessoBanco [boot] [devtools]
  - src/main/java
    - com.projeto.acesso
      - AcessoBancoApplication.java
      - com.projeto.acesso.models
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
- PrimeiroSpring2 [boot] [devtools]
- PrimeiroString [boot] [devtools]

**New**

- Open in New Window
- Open Type Hierarchy F4
- Show In Alt+Shift+W
- Show in Local Terminal
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Build Path
- Source Alt+Shift+S
- Refactor Alt+Shift+T
- Import...
- Export...

**New Java Class**

Java Class

Create a new Java class.

Source folder: AcessoBanco/src/main/java Browse...

Package: com.projeto.acesso.models Browse...

Enclosing type: Browse...

Name: Clientes

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static  
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

# Criação da classe Cliente

- Adicione o código referente a classe cliente, que deve representar a tabela cliente do banco de dados na camada da aplicação. Note que a classe possui anotações que qualifica cada propriedade desta classe

```
Cliente.java X
1 package com.projeto.acesso.models;
2
3 import jakarta.persistence.Column;
4 import jakarta.persistence.Entity;
5 import jakarta.persistence.GeneratedValue;
6 import jakarta.persistence.GenerationType;
7 import jakarta.persistence.Id;
8
9 @Entity
10 public class Cliente {
11
12     @Id
13     @GeneratedValue(strategy = GenerationType.IDENTITY)
14     private Integer idcliente;
15
16     @Column
17     private String nomecliente;
18
19     @Column
20     private String email;
21
22     @Column
23     private String cpf;
24
25     @Column
26     private String telefone;
27
28     @Column
29     private Integer idade;
30
31     public Cliente() {
32
33     }
34
35     public Cliente(Integer idcliente, String nomecliente, String email, String cpf, String telefone, Integer idade) {
36         super();
37         this.idcliente = idcliente;
38         this.nomecliente = nomecliente;
39         this.email = email;
40         this.cpf = cpf;
41         this.telefone = telefone;
42         this.idade = idade;
43     }
44 }
```

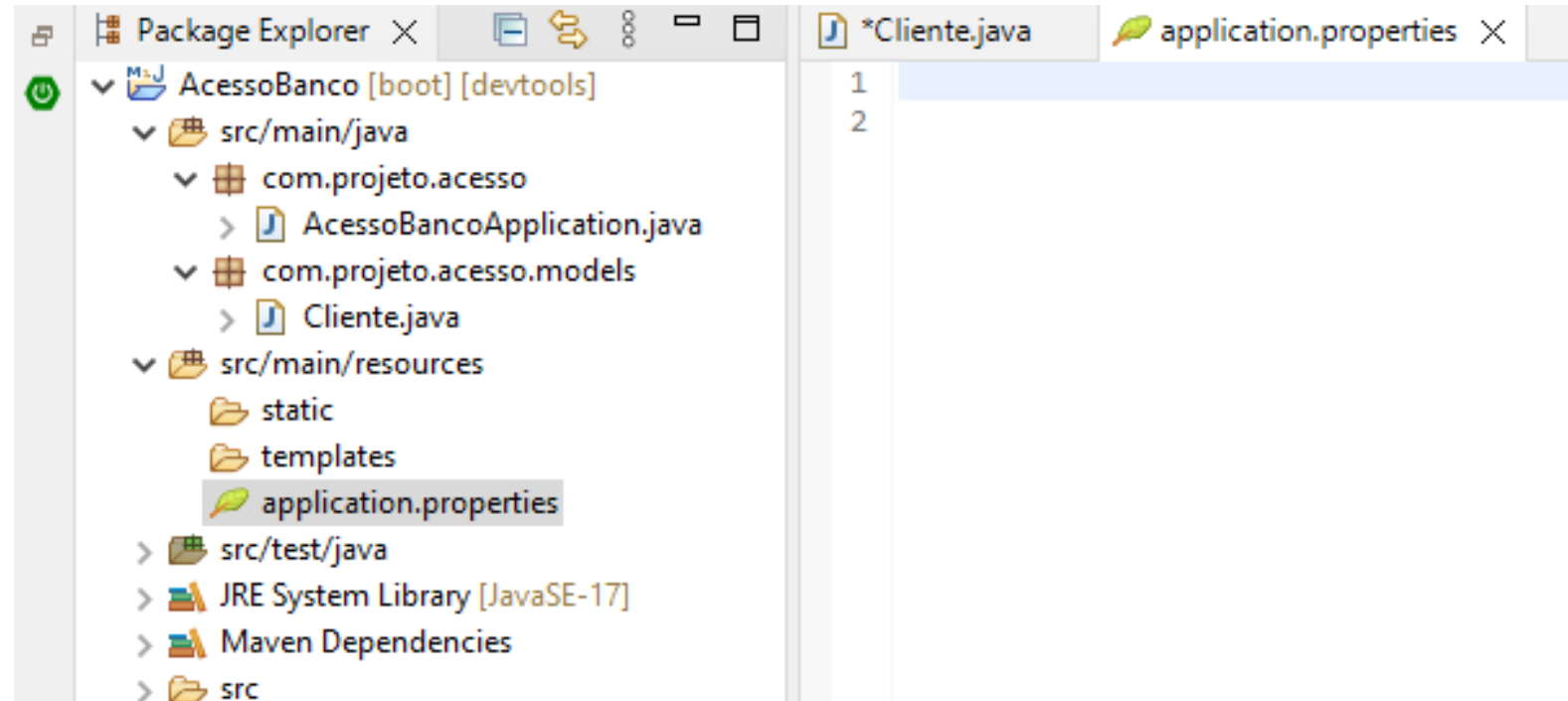
# Criação da classe Cliente

- Adicione o código referente a classe cliente, que deve representar a tabela cliente do banco de dados na camada da aplicação. Note que a classe possui anotações que qualifica cada propriedade desta classe

```
44- public Integer getIdcliente() {  
45-     return idcliente;  
46- }  
47- public void setIdcliente(Integer idcliente) {  
48-     this.idcliente = idcliente;  
49- }  
50- public String getNomecliente() {  
51-     return nomecliente;  
52- }  
53- public void setNomecliente(String nomecliente) {  
54-     this.nomecliente = nomecliente;  
55- }  
56- public String getEmail() {  
57-     return email;  
58- }  
59- public void setEmail(String email) {  
60-     this.email = email;  
61- }  
62- public String getCpf() {  
63-     return cpf;  
64- }  
65- public void setCpf(String cpf) {  
66-     this.cpf = cpf;  
67- }  
68- public String getTelefone() {  
69-     return telefone;  
70- }  
71- public void setTelefone(String telefone) {  
72-     this.telefone = telefone;  
73- }  
74- public Integer getIdade() {  
75-     return idade;  
76- }  
77- public void setIdade(Integer idade) {  
78-     this.idade = idade;  
79- }  
80- }
```

# Adição de configuração

- Agora, vamos configurar os parâmetros da aplicação para acessar o banco de dados.
- Abra o arquivo application.properties





# Adição de configuração

- Adicione as configurações

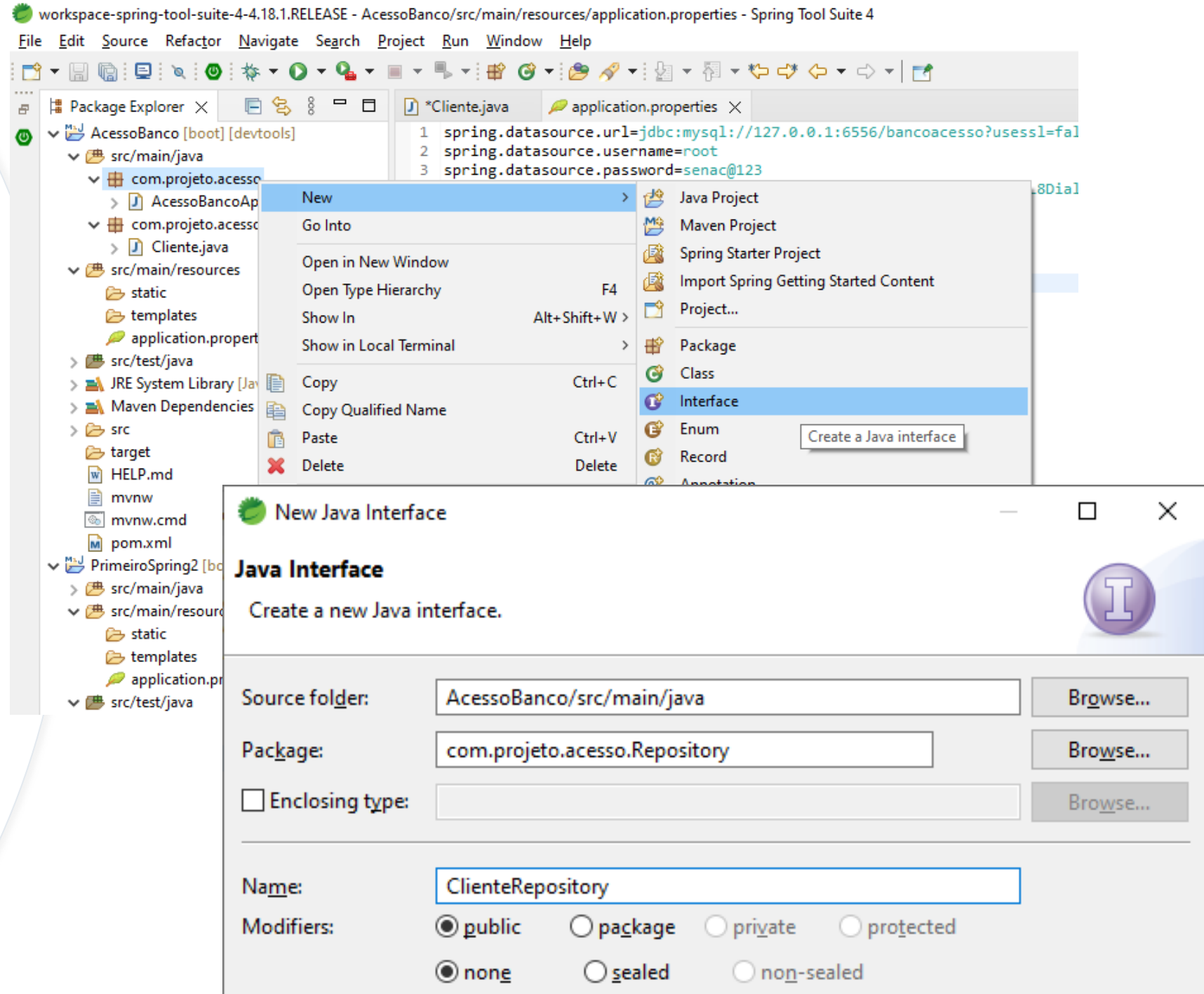


application.properties X

```
1 spring.datasource.url=jdbc:mysql://127.0.0.1:3306/bancoacesso?useSSL=false
2 spring.datasource.username=root
3 spring.datasource.password=
4 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
5 spring.jpa.hibernate.ddl-auto=update
6
7 #porta do servidor spring
8 server.port=8080
9
```

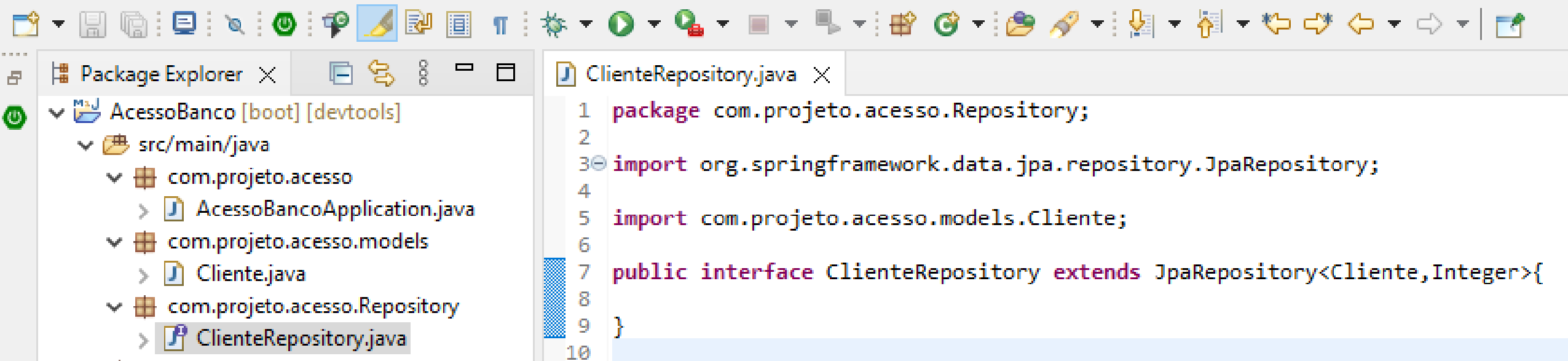
# Criando o arquivo de manipulação de dados

- Vamos criar o arquivo chamado ClienteRepository que será o responsável pela persistência dos dados em banco. Este arquivo é uma interface e deverá ser criado dentro de um package chamado Repository



# Criando o arquivo de manipulação de dados

- Adicione o seguinte código a interface criada.

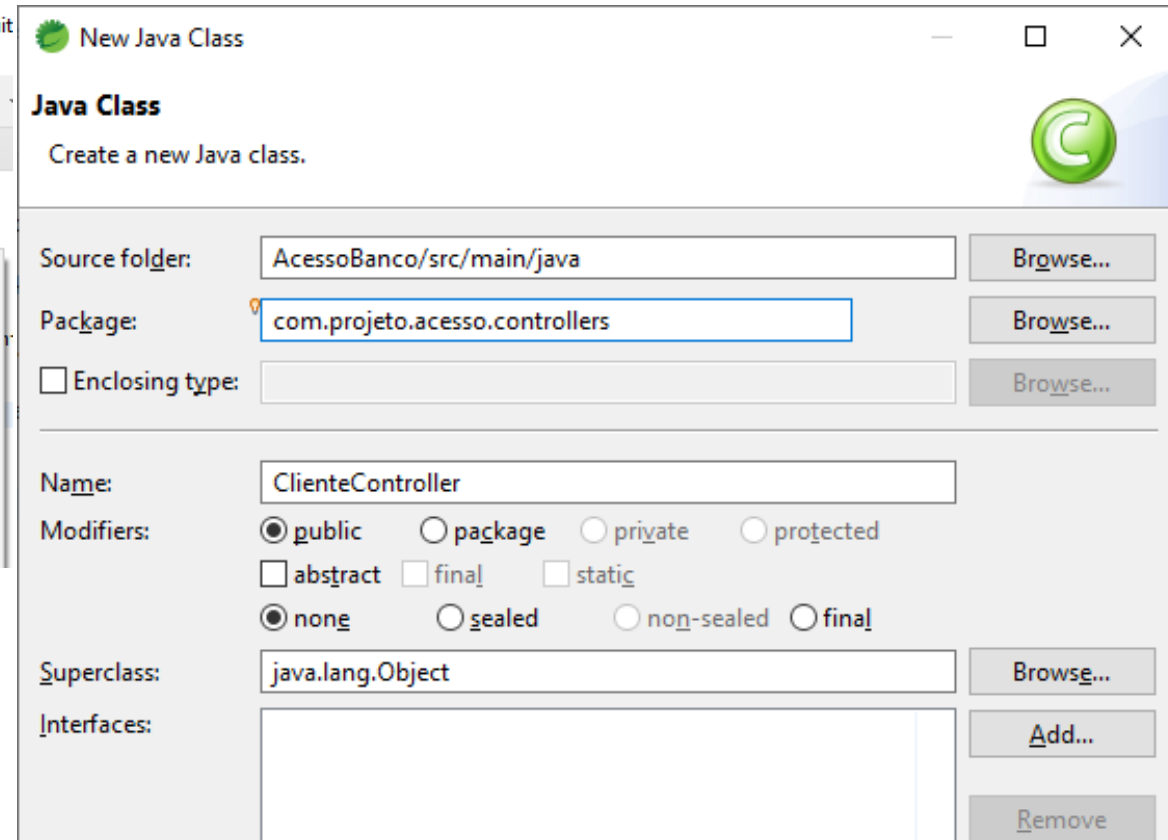
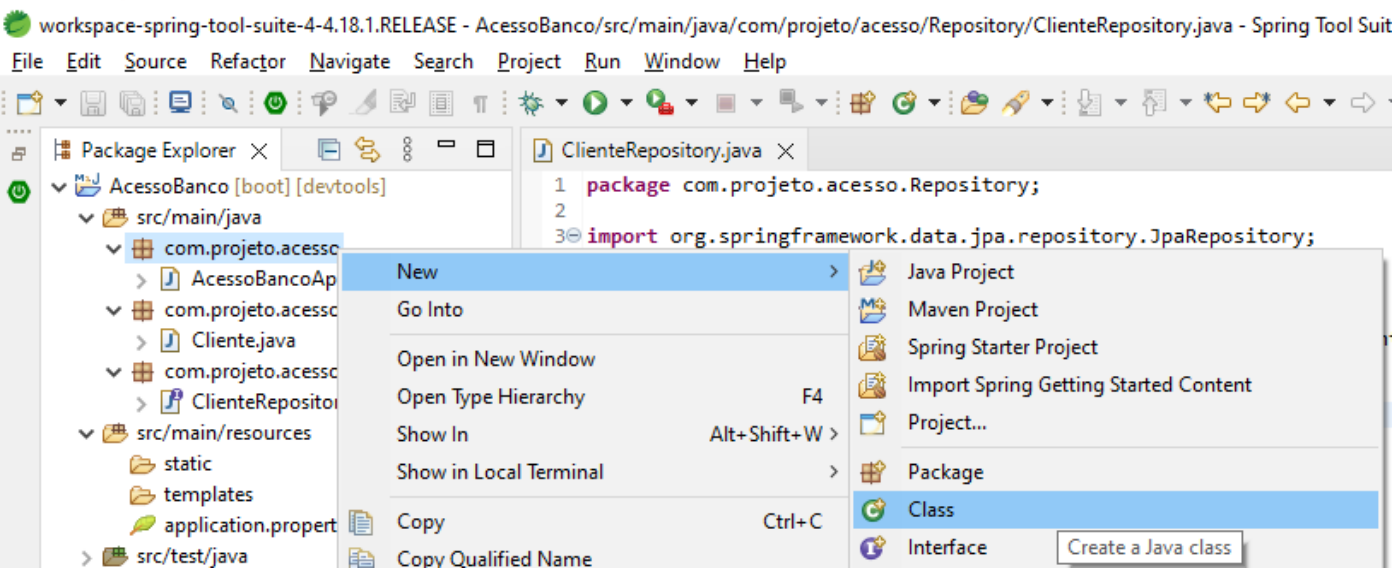


The screenshot shows an IDE interface with a Package Explorer on the left and a code editor on the right. The Package Explorer shows a project named 'AcessoBanco [boot] [devtools]' with a package structure: 'src/main/java' containing 'com.projeto.acesso' (with 'AcessoBancoApplication.java'), 'com.projeto.acesso.models' (with 'Cliente.java'), and 'com.projeto.acesso.Repository' (with 'ClienteRepository.java' selected). The code editor shows the content of 'ClienteRepository.java' with the following code:

```
1 package com.projeto.acesso.Repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 import com.projeto.acesso.models.Cliente;
6
7 public interface ClienteRepository extends JpaRepository<Cliente,Integer>{
8
9 }
10
```

# Agora, vamos criar o controlador de dados

- Crie um novo package chamado controllers e adicione uma nova classe chamada ClienteController.



## Agora, vamos criar o controlador de dados

- Adicionar o código do controlador do ClienteController

```
*ClienteController.java ×
1 package com.projeto.acesso.controllers;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.web.bind.annotation.GetMapping;
7 import org.springframework.web.bind.annotation.RestController;
8
9 import com.projeto.acesso.Repository.ClienteRepository;
10 import com.projeto.acesso.models.Cliente;
11
12 @RestController
13 public class ClienteController {
14
15     @Autowired
16     private ClienteRepository clienteRepo;
17
18     @GetMapping("/clientes/listar")
19     public List<Cliente> listar(){
20         return clienteRepo.findAll();
21     }
22
23 }
```

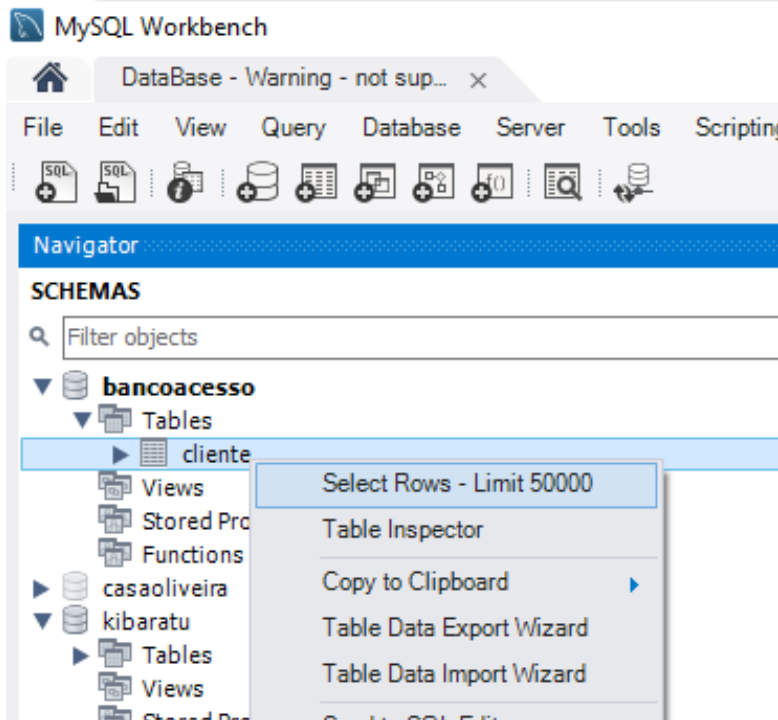
# Construindo o banco de dados

- Agora, vamos criar o banco de dados. Crie conforme abaixo

```
1 • create database bancoacesso charset=utf8mb4 collate=utf8mb4_general_ci;
2 • use bancoacesso;
3 • create table cliente(
4     idcliente int auto_increment primary key,
5     nomecliente varchar(100) not null,
6     email varchar(100) not null,
7     cpf varchar(20) not null unique,
8     telefone varchar(15) not null,
9     idade int not null
10 )engine innodb charset=utf8mb4 collate=utf8mb4_general_ci;
```

# Cadastre alguns clientes

- Selecione a tabela clicando sobre ela e selecionando a opção Select Rows



The screenshot shows the 'Result Grid' pane in MySQL Workbench. It displays a single row of data with the following columns: 'iddcliente', 'nomecliente', 'email', 'cpf', 'telefone', and 'idade'. All cells in the row contain the value 'NULL'.


	iddcliente	nomecliente	email	cpf	telefone	idade
▶▶	NULL	NULL	NULL	NULL	NULL	NULL




# Cadastre alguns clientes



- Digite diretamente nos campos os dados pedidos, com exceção do idcliente. Depois clique em Apply. Cadastre ao menos 10 clientes


<

Result Grid

 Filter Rows:

Edit:   

Export/Import:  

Wrap Cell Content: 

☐

	idcliente	nomecliente	email	cpf	telefone	idade
	NULL	Roberto	roberto@uol.com.br	54566	(11) 8959-5454	25
	NULL	Márcia	marcia@terra.com.br	5487545	(11) 8598-9568	41
▶*		NULL	NULL	NULL	NULL	NULL

cliente 1 ×

Apply

Output



# Cadastre alguns clientes

- Clique em Apply novamente e depois Finish

Apply SQL Script to Database

Review SQL Script

Apply SQL Script

Review the SQL Script to be Applied on the Database

```
1  INSERT INTO `bancoacesso`.`cliente` (`nomecliente`, `email`, `cpf`, `telefone`, `
2  INSERT INTO `bancoacesso`.`cliente` (`nomecliente`, `email`, `cpf`, `telefone`, `
3
```

< | >

Back

Apply

Cancel

Apply SQL Script to Database

Review SQL Script

Apply SQL Script

Applying SQL script to the database

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

☒ Execute SQL Statements

SQL script was successfully applied to the database.

Show Logs

Back

Finish

Cancel



## Teste sua aplicação

- Vamos subir o servidor e testar a aplicação. Siga como é mostrado













The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays the project structure for 'AcessoBanco' and 'PrimeiroSpring2'. The 'AcessoBanco' project is expanded, showing the 'src/main/java' directory with the 'com.projeto.acesso' package. The 'PrimeiroSpring2' project is also expanded, showing the 'src/main/java' directory with the 'com.projeto.primeirosp' package. On the right, the Run menu is open, showing options like 'Run As', 'Debug As', and 'Profile As'. The 'Run As' option is highlighted.

```

  ____ _
 / ___ \| | | |
| |___ \ |_| |
 \___)_____|_/_
:: Spring Boot ::      (v3.1.0)

2023-06-06T13:00:55.761-03:00 INFO 11824 --- [ restartedMain] c.projeto.acesso.AcessoBancoApplication : Starting AcessoBancoApplication usin
2023-06-06T13:00:55.764-03:00 INFO 11824 --- [ restartedMain] c.projeto.acesso.AcessoBancoApplication : No active profile set, falling back
2023-06-06T13:00:55.811-03:00 INFO 11824 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active!
2023-06-06T13:00:55.811-03:00 INFO 11824 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging
2023-06-06T13:00:56.588-03:00 INFO 11824 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repos:
2023-06-06T13:00:56.640-03:00 INFO 11824 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scan
2023-06-06T13:00:57.135-03:00 INFO 11824 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 808
2023-06-06T13:00:57.145-03:00 INFO 11824 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-06-06T13:00:57.145-03:00 INFO 11824 --- [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tor
2023-06-06T13:00:57.210-03:00 INFO 11824 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApp:
2023-06-06T13:00:57.210-03:00 INFO 11824 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initial:
2023-06-06T13:00:57.413-03:00 INFO 11824 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUn:
2023-06-06T13:00:57.462-03:00 INFO 11824 --- [ restartedMain] org.hibernate.Version : HHH000412: Hibernate ORM core versio
2023-06-06T13:00:57.465-03:00 INFO 11824 --- [ restartedMain] org.hibernate.cfg.Environment : HHH000406: Using bytecode reflection
2023-06-06T13:00:57.602-03:00 INFO 11824 --- [ restartedMain] o.h.b.i.BytecodeProviderInitiator : HHH000021: Bytecode provider name :
2023-06-06T13:00:57.720-03:00 INFO 11824 --- [ restartedMain] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup; ignoring JI
2023-06-06T13:00:57.736-03:00 INFO 11824 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-06-06T13:00:57.876-03:00 INFO 11824 --- [ restartedMain] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection com.
2023-06-06T13:00:57.878-03:00 INFO 11824 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2023-06-06T13:00:57.900-03:00 INFO 11824 --- [ restartedMain] org.hibernate.orm.dialect : HHH035001: Using dialect: org.hiberr
2023-06-06T13:00:57.902-03:00 WARN 11824 --- [ restartedMain] org.hibernate.orm.deprecation : HHH90000026: MySQL8Dialect has been
2023-06-06T13:00:58.087-03:00 INFO 11824 --- [ restartedMain] o.h.b.i.BytecodeProviderInitiator : HHH000021: Bytecode provider name :
2023-06-06T13:00:58.613-03:00 INFO 11824 --- [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implem
2023-06-06T13:00:58.668-03:00 INFO 11824 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory:
2023-06-06T13:00:58.907-03:00 WARN 11824 --- [ restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled l
2023-06-06T13:00:59.323-03:00 INFO 11824 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port
2023-06-06T13:00:59.328-03:00 INFO 11824 --- [ restartedMain] o.s.b.a.e.web.EndpointLinksResolver : Exposing 1 endpoint(s) beneath base
2023-06-06T13:00:59.399-03:00 INFO 11824 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (ht
2023-06-06T13:00:59.413-03:00 INFO 11824 --- [ restartedMain] c.projeto.acesso.AcessoBancoApplication : Started AcessoBancoApplication in 3t
2023-06-06T13:01:00.538-03:00 INFO 11824 --- [on(3)-127.0.0.1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServl
2023-06-06T13:01:00.538-03:00 INFO 11824 --- [on(3)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServl
2023-06-06T13:01:00.539-03:00 INFO 11824 --- [on(3)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms

```

	1 Java Application	Alt+Shift+X, J
	2 Java Application In Container	
	3 JUnit Test	Alt+Shift+X, T
	4 Maven build	Alt+Shift+X, M
	5 Maven build...	
	6 Maven clean	
	7 Maven generate-sources	
	8 Maven install	
	9 Maven test	
	Maven verify	
	Spring Boot App	Alt+Shift+X, B
	Spring Boot Client	

# Teste a rota no postman

- Vamos acessar o postman e com o verbo de solicitação GET digite a seguinte url:
- <http://127.0.0.1:8080/clientes/listar>
- E Clique no botão Send

The screenshot shows the Postman application interface. At the top, the URL bar displays 'http://127.0.0.1:8080/clientes/listar'. Below the URL bar, the HTTP method is set to 'GET'. To the right of the method and URL is a 'Send' button. Below the URL bar, there are tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Params' tab is currently selected. Under the 'Params' tab, there is a section for 'Query Params' which is currently empty. On the right side of the interface, there are buttons for 'Save', 'Send', and 'Bulk Edit'.

Key	Value	Description	...	Bulk Edit
-----	-------	-------------	-----	-----------

http://127.0.0.1:8080/clientes/listar

Save



GET

http://127.0.0.1:8080/clientes/listar

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Cookies

### Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 228 ms Size: 412 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 [
2   {
3     "idcliente": 1,
4     "nomecliente": "Roberto",
5     "email": "roberto@uol.com.br",
6     "cpf": "54566",
7     "telefone": "(11) 8959-5454",
8     "idade": 25
9   },
10  {
11    "idcliente": 2,
12    "nomecliente": "Márcia",
13    "email": "marcia@terra.com.br",
14    "cpf": "5487545",
15    "telefone": "(11) 8598-9568",
16    "idade": 41
17  }
18 ]
```