

API Rest Java e Spring Boot

Estudo de caso

Sumário

- Criar o **projeto**
- Implementar o **modelo de domínio**
- Fazer o **mapeamento O-R** (Objeto Relacional)
- Configurar o **Banco de Dados**
- Criar os **end-point** com **Java e Spring**

Visão Geral do Sistema

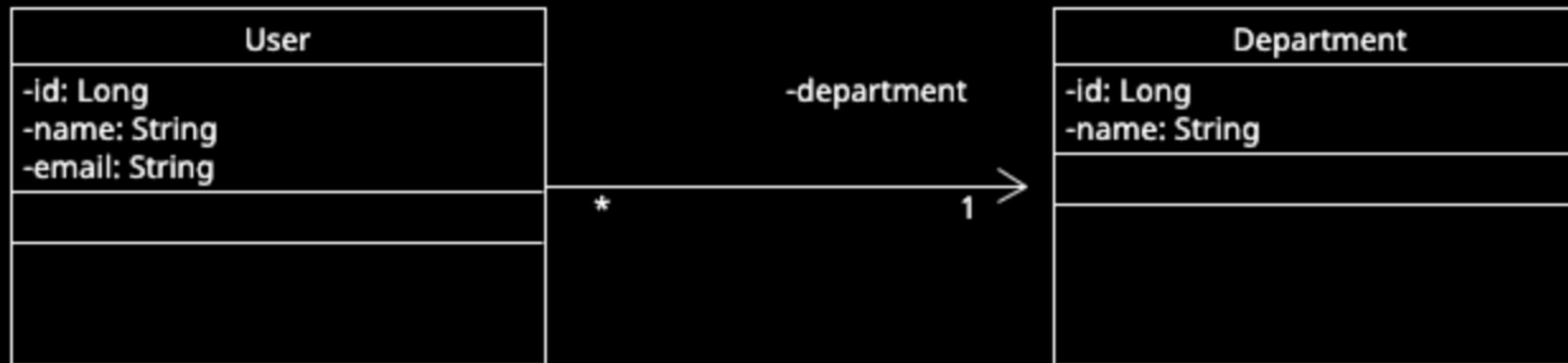
Vamos construir um pequeno sistema (API REST) de **usuários** e **departamentos**, com os seguintes casos de uso:

- Buscar todos usuários
- Buscar um usuário pelo seu id
- Inserir um novo usuário

Diagrama de Classes



Mapeamento Relacional → Objeto → json



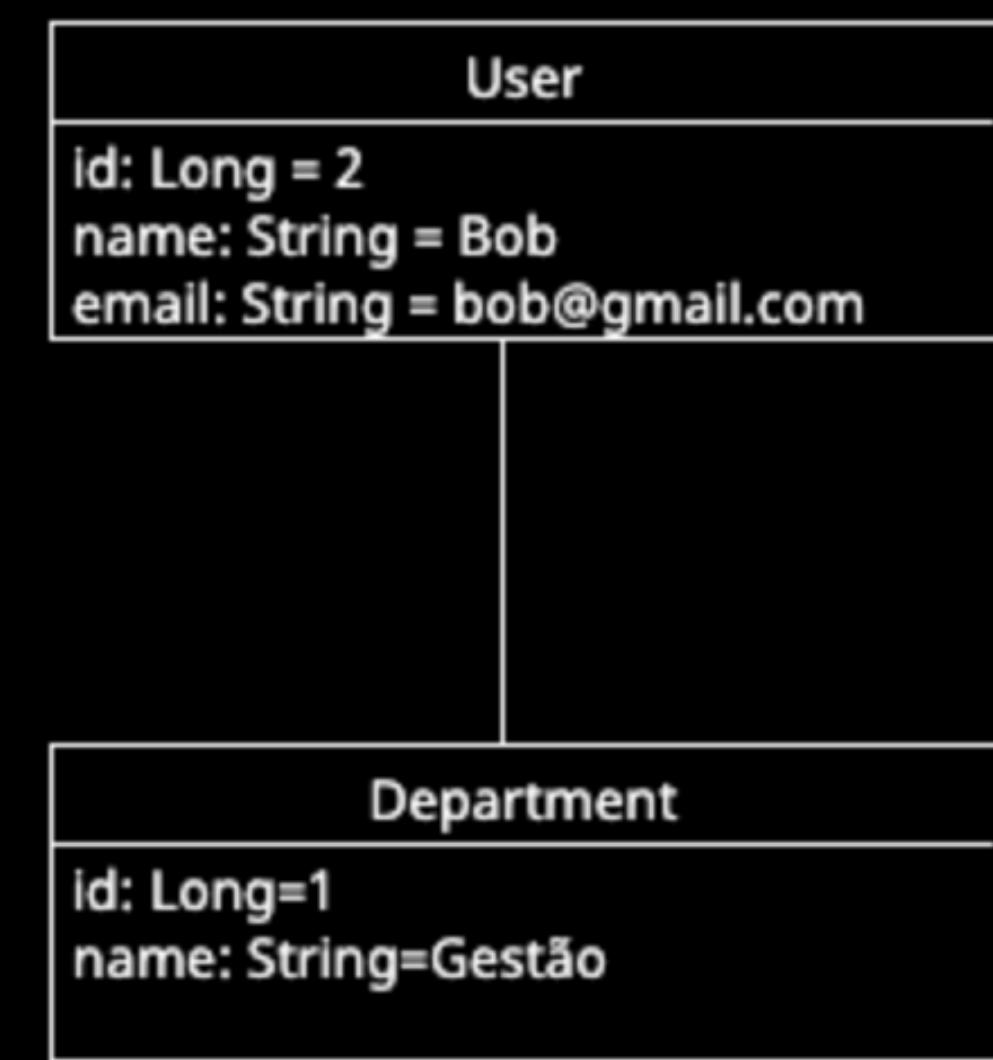
tb_department

id	name
1	Gestão
2	Informática

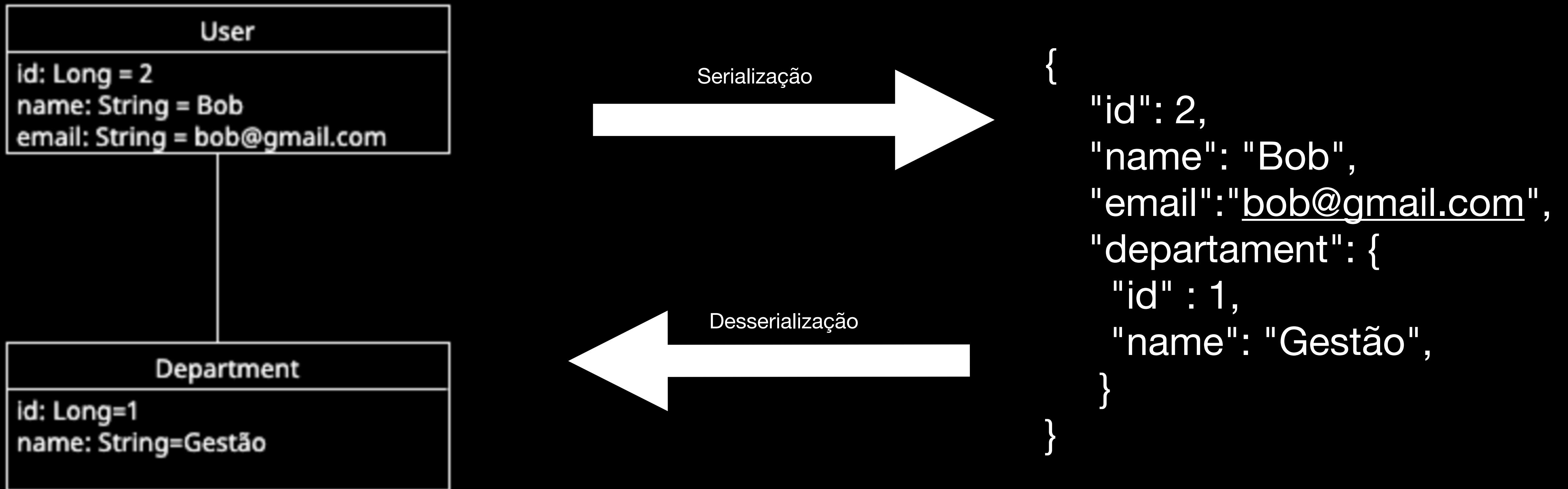
tb_user

id	name	email	department_id
1	Maria	maria@gmail.com	1
2	Bob	bob@gmail.com	1
3	Alex	alex@gmail.com	2
4	Ana	ana@gmail.com	2

ORM



Modelo - Relacional -> Objeto -> json



Criação do Projeto



Vamos criar o projeto...

Project

Gradle - Groovy

Gradle - Kotlin

Maven

Language

Java

Kotlin

Groovy

Spring Boot

3.2.0 (SNAPSHOT)

3.2.0 (RC1)

3.1.6 (SNAPSHOT)

3.1.5

3.0.13 (SNAPSHOT)

3.0.12

2.7.18 (SNAPSHOT)

2.7.17

Project Metadata

Group com.example

Artifact demo3

Name demo3

Description Aula Java web com Spring Boot

Package name com.example.demo3

Packaging Jar War

Java 21 17 11 8

Dependencies

[ADD DEPENDENCIES... ⌘ + B](#)

Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Data JPA SQL

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

H2 Database SQL

Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

[GENERATE ⌘ + ↵](#)

[EXPLORE CTRL + SPACE](#)

[SHARE...](#)



Project

Gradle - Groovy

Gradle - Kotlin

Maven

Language

Java

Kotlin

Groovy

Spring Boot

3.2.0 (SNAPSHOT)

3.2.0 (RC1)

3.1.6 (SNAPSHOT)

3.1.5

3.0.13 (SNAPSHOT)

3.0.12

2.7.18 (SNAPSHOT)

2.7.17

Project Metadata

Group com.example

Artifact demo3

Name demo3

Description Aula Java web com Spring Boot

Package name com.example.demo3

Packaging Jar War

Java 21 17 11 8

Dependencies

1

ADD DEPENDENCIES... ⌘ + B

Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Data JPA SQL

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

H2 Database SQL

Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

2

3

4

5

GENERATE ⌘ + ↵

EXPLORE CTRL + SPACE

SHARE...

demo3.zip

<descompacte o arquivo>



demo3



Voltar/Avançar

Visualizar

Grupo

Ação

Compartilhar

Editar Etiquetas

Dropbox

Buscar

Favoritos

Dropbox (Developer Ac...)

Deleted Users

Aplicativos

Recentes

Filmes

SharePlan (1)

Imagens

Música

Downloads

AirDrop

Documentos

Mesa

iCloud

Localizações

MacBook Pro de Fernan...

iPhone de Fernando... ▲

Macintosh HD

Rede

Etiquetas

Nome

^

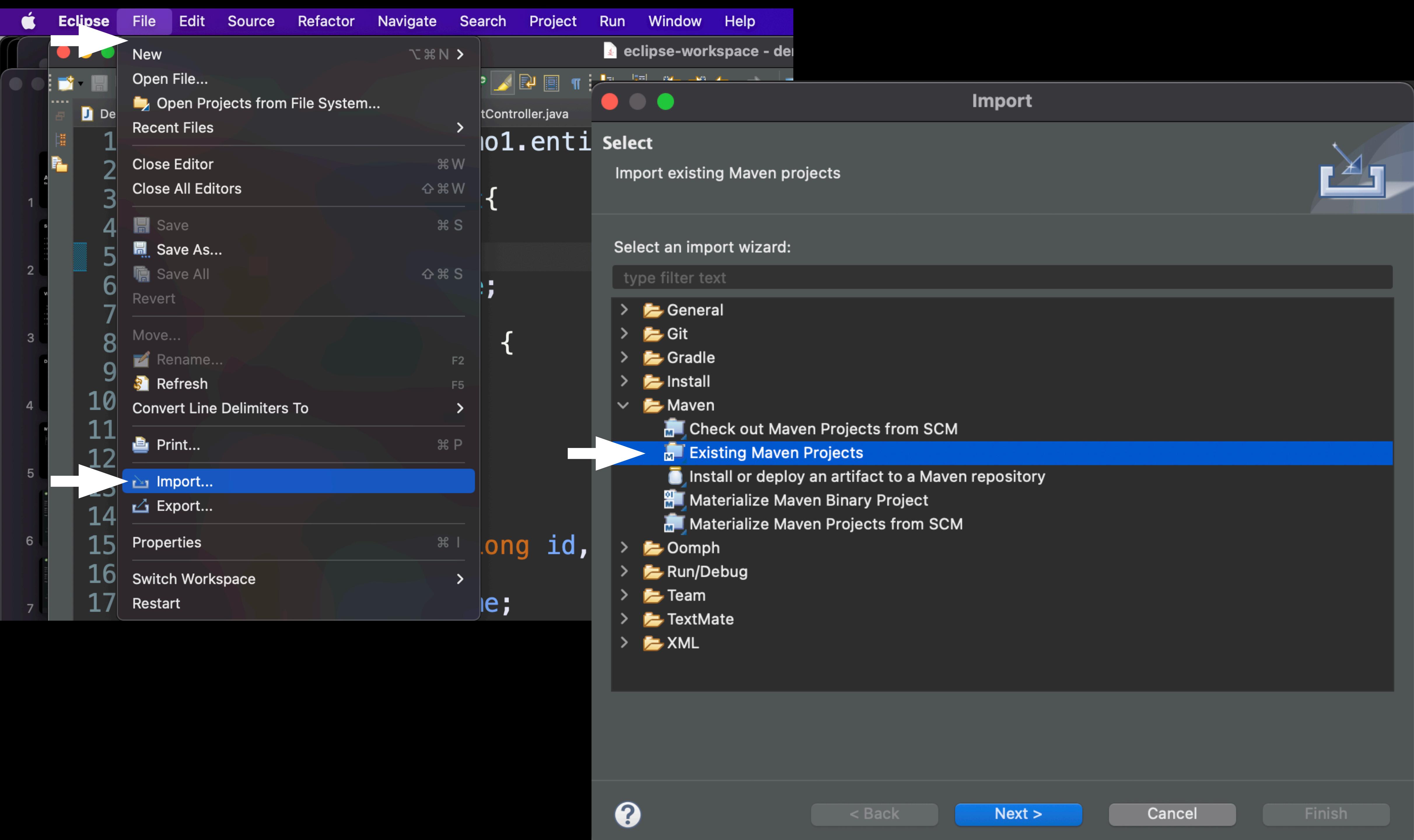
Data de Modificação

Tamanho

Tipo

HELP.md	Hoje 15:53	1 KB	Markdo...cument
mvnw	Hoje 15:53	11 KB	Arquivo...vel Unix
mvnw.cmd	Hoje 15:53	8 KB	Documento
pom.xml	Hoje 15:53	1 KB	XML Document
> src	Hoje 15:53	--	Pasta

Estrutura do projeto



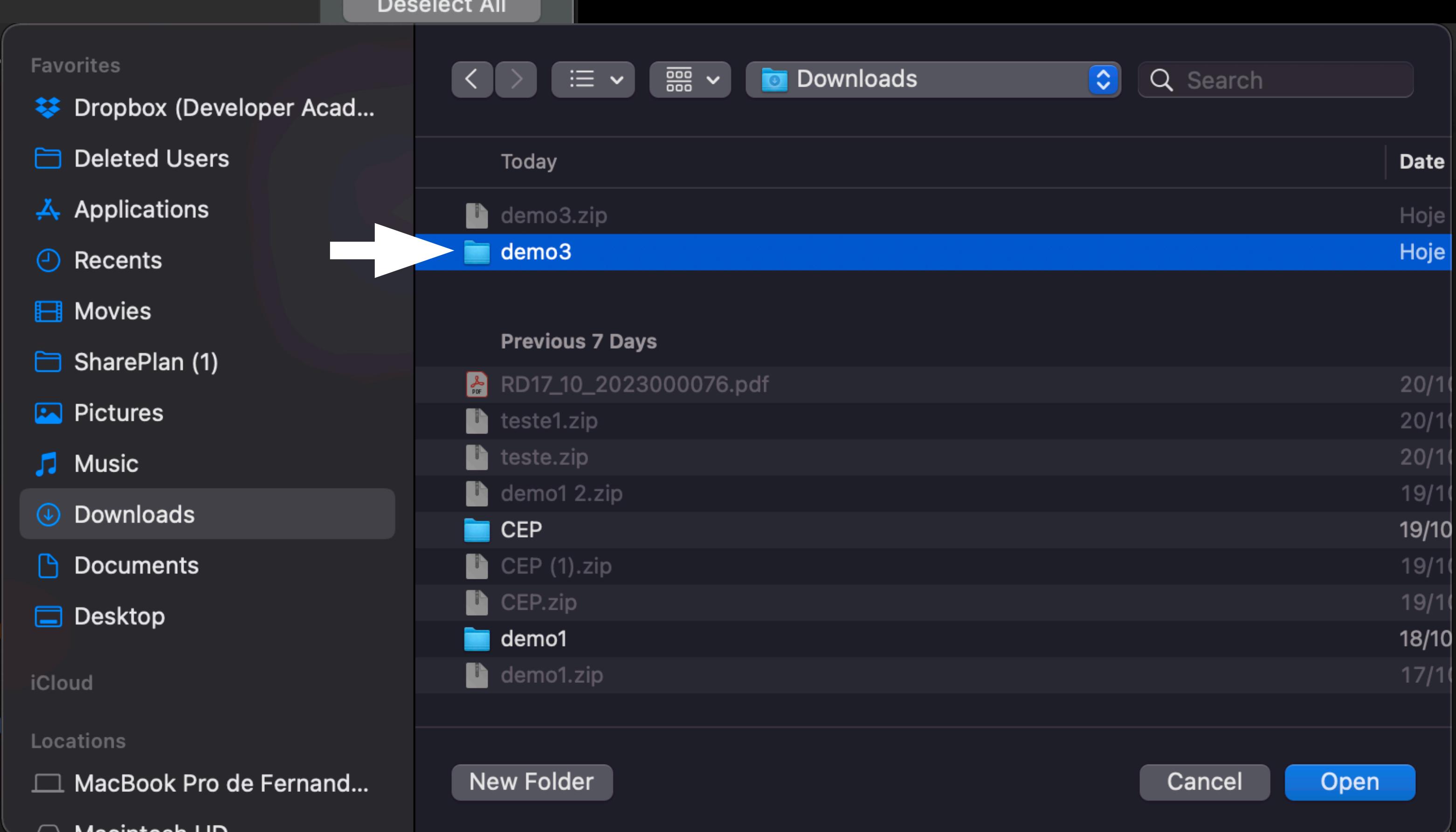
Import Maven Projects

Maven Projects

Select Maven projects

Root Directory:

Projects:



Add project(s) to working set

► Advanced



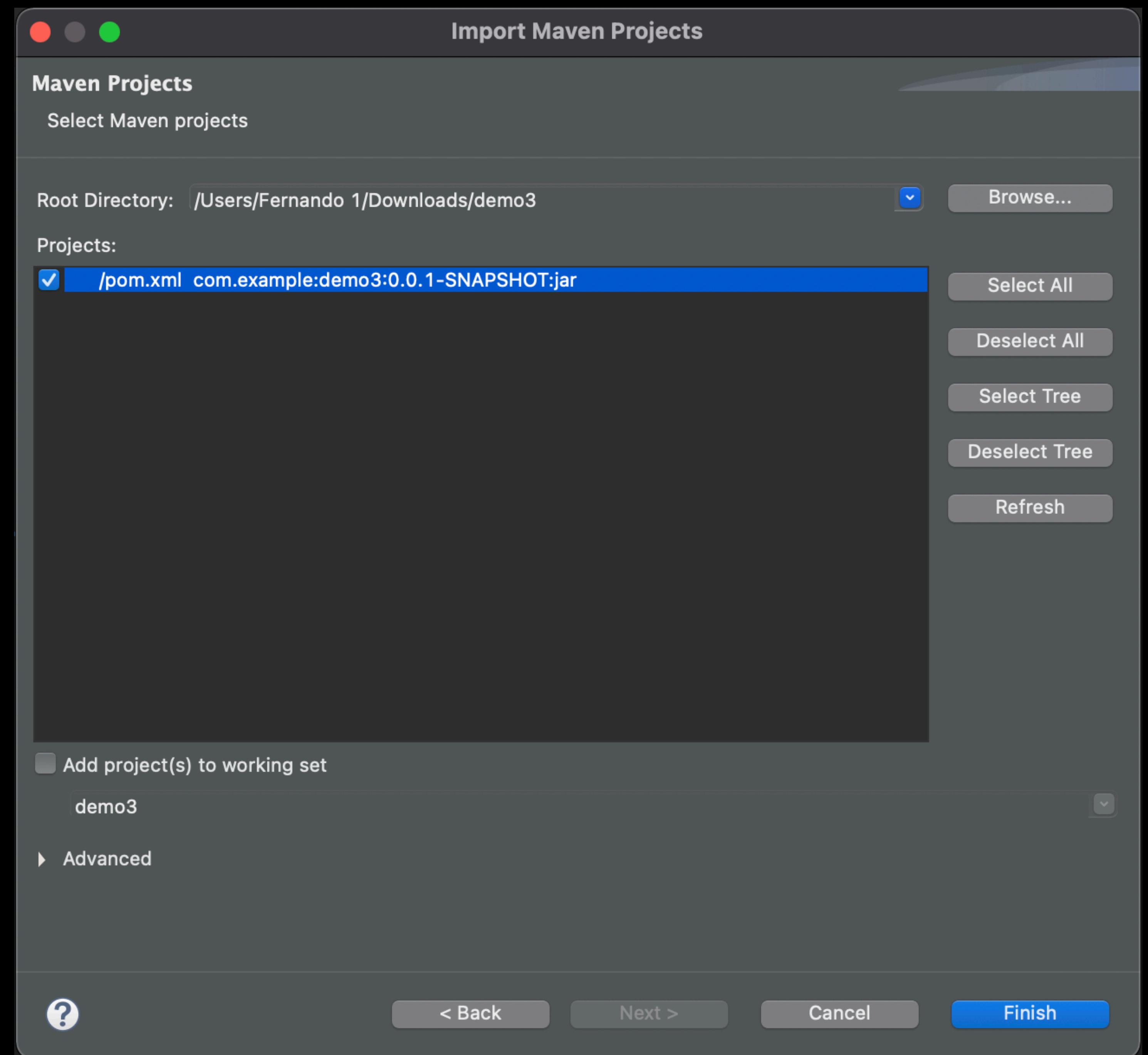
< Back

Next >

New Folder

Cancel

Open

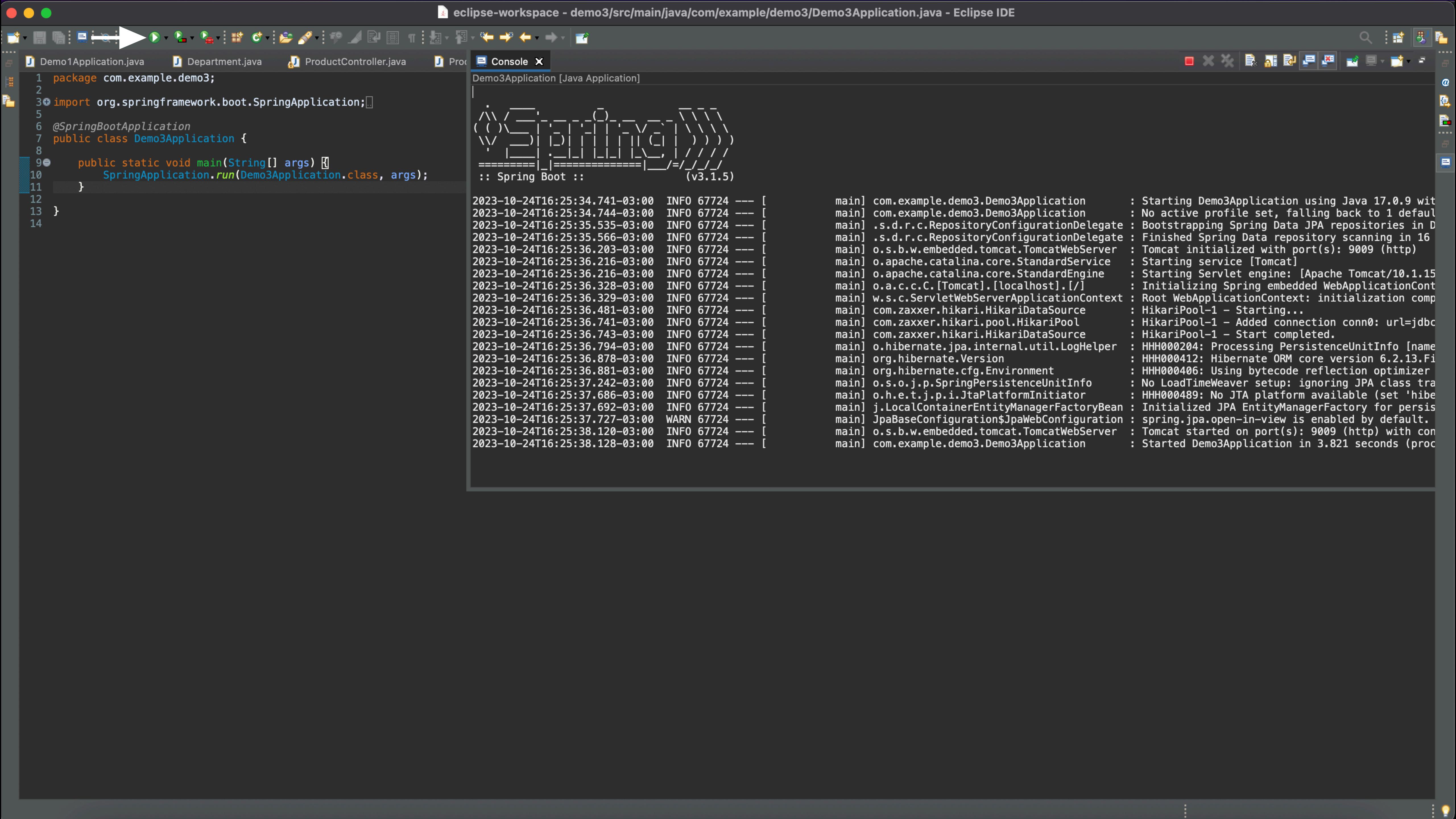


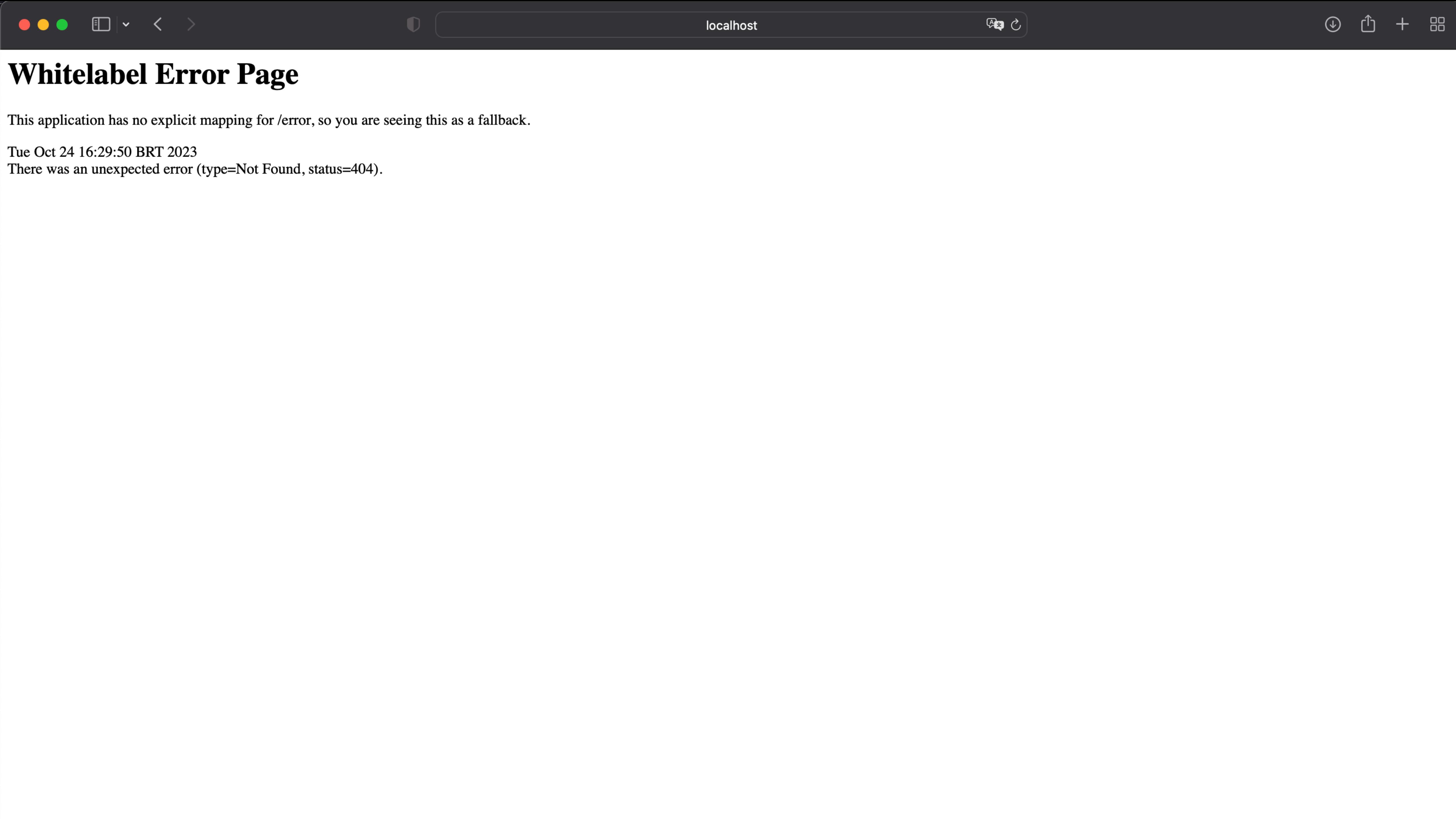
The screenshot shows the Eclipse IDE interface with two projects open:

- Project Explorer (Left):** Shows the file structure for the `demo1` project. It includes a folder named `CP1DDDJogoDaVelha`, several Java files like `crud`, `demo`, `demo1`, and `ProductController.java`, and other resources like `JRE System Library [JavaSE-15]`, `Maven Dependencies`, and `src`.
- Code Editor (Top Center):** Displays the `Demo1Application.java` file, which contains the main method for a Spring Boot application.
- Code Editor (Bottom Center):** Displays the `Demo3Application.java` file, which also contains a main method for a Spring Boot application.
- Console (Right):** Shows the output of running the `Demo3Application`. The log output includes Spring Boot initialization messages and Hibernate configuration details.

```
1 package com.example.demo3;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 public class Demo3Application {
7
8     public static void main(String[] args) {
9         SpringApplication.run(Demo3Application.class, args);
10    }
11 }
```

```
2023-10-24T16:25:34.741-03:00 INFO 67724 --- [main] com.example.demo3.Demo3Application : Starting Demo3Application
2023-10-24T16:25:34.744-03:00 INFO 67724 --- [main] com.example.demo3.Demo3Application : No active profile set, falling back to default profiles: default
2023-10-24T16:25:35.535-03:00 INFO 67724 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data repository locations
2023-10-24T16:25:35.566-03:00 INFO 67724 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository initialization
2023-10-24T16:25:36.203-03:00 INFO 67724 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080
2023-10-24T16:25:36.216-03:00 INFO 67724 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-10-24T16:25:36.216-03:00 INFO 67724 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Tomcat/9.0.60]
2023-10-24T16:25:36.328-03:00 INFO 67724 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-10-24T16:25:36.329-03:00 INFO 67724 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext
2023-10-24T16:25:36.481-03:00 INFO 67724 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-10-24T16:25:36.741-03:00 INFO 67724 --- [main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection
2023-10-24T16:25:36.743-03:00 INFO 67724 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2023-10-24T16:25:36.794-03:00 INFO 67724 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo
2023-10-24T16:25:36.878-03:00 INFO 67724 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM
2023-10-24T16:25:36.881-03:00 INFO 67724 --- [main] org.hibernate.cfg.Environment : HHH000406: Using bytecode
2023-10-24T16:25:37.242-03:00 INFO 67724 --- [main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup
2023-10-24T16:25:37.686-03:00 INFO 67724 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform
2023-10-24T16:25:37.692-03:00 INFO 67724 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2023-10-24T16:25:37.727-03:00 WARN 67724 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering
2023-10-24T16:25:38.120-03:00 INFO 67724 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s)
2023-10-24T16:25:38.128-03:00 INFO 67724 --- [main] com.example.demo3.Demo3Application : Started Demo3Application
```





Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Tue Oct 24 16:29:50 BRT 2023

There was an unexpected error (type=Not Found, status=404).

Trecho de código para copiar

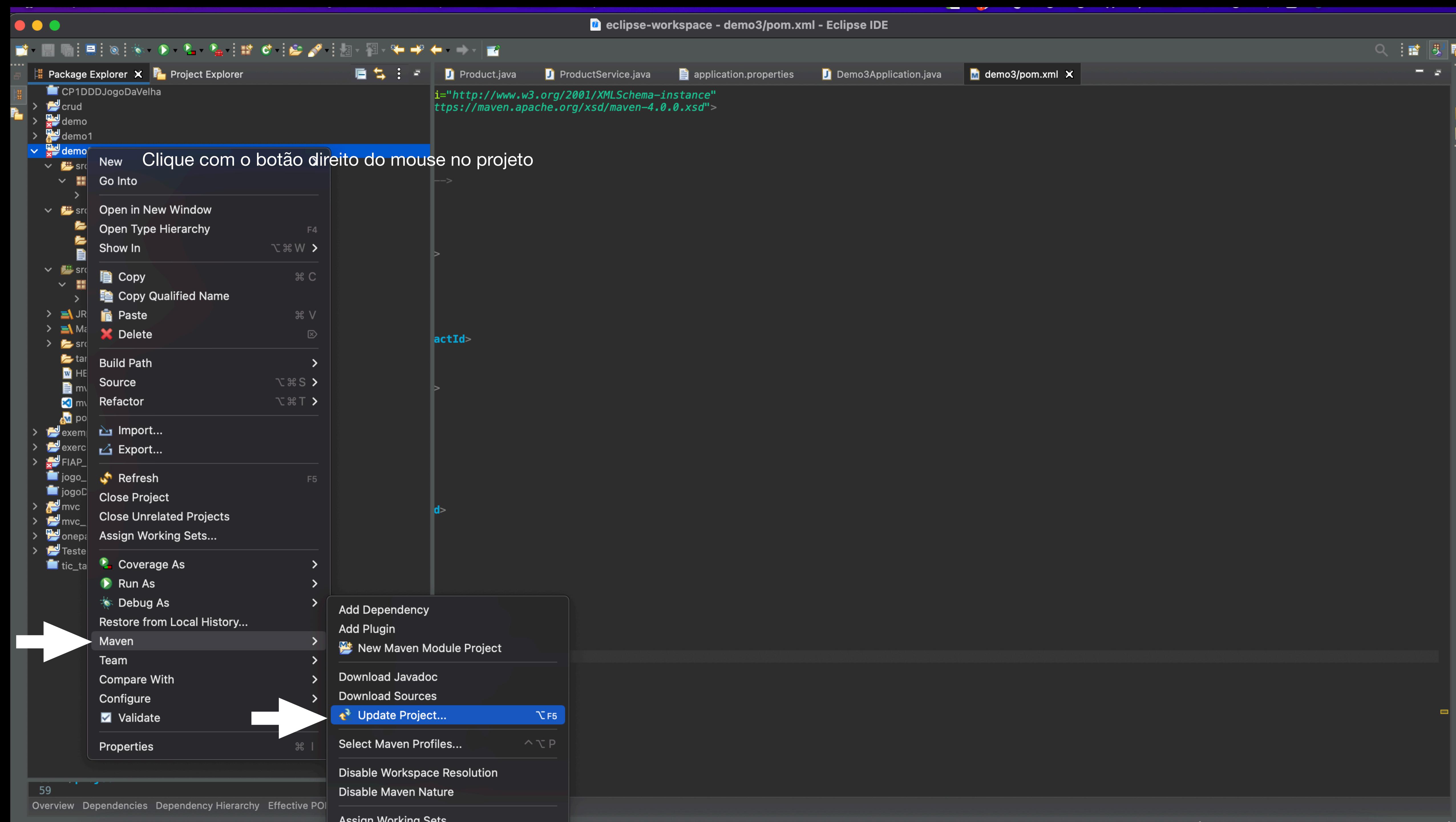
Configuração do Maven Resources Plugin

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-resources-plugin</artifactId>
  <version>3.1.0</version>
</plugin>
```

Versão customizada para evitar problemas de versão do Maven

```
15   <description>Aula Java web com Spring Boot</description>
16   <properties>
17     <java.version>17</java.version>
18   </properties>
19   <dependencies>
20     <dependency>
21       <groupId>org.springframework.boot</groupId>
22       <artifactId>spring-boot-starter-data-jpa</artifactId>
23     </dependency>
24     <dependency>
25       <groupId>org.springframework.boot</groupId>
26       <artifactId>spring-boot-starter-web</artifactId>
27     </dependency>
28
29     <dependency>
30       <groupId>com.h2database</groupId>
31       <artifactId>h2</artifactId>
32       <scope>runtime</scope>
33     </dependency>
34     <dependency>
35       <groupId>org.springframework.boot</groupId>
36       <artifactId>spring-boot-starter-test</artifactId>
37       <scope>test</scope>
38     </dependency>
39   </dependencies>
40
41   <build>
42     <plugins>
43       <plugin>
44         <groupId>org.springframework.boot</groupId>
45         <artifactId>spring-boot-maven-plugin</artifactId>
46       </plugin>
47
48       |
49       <plugin>
50         <groupId>org.apache.maven.plugins</groupId>
51         <artifactId>maven-resources-plugin</artifactId>
52         <version>3.1.0</version>
53       </plugin>
54
55     </plugins>
56   </build>
57
58 </project>
59
```

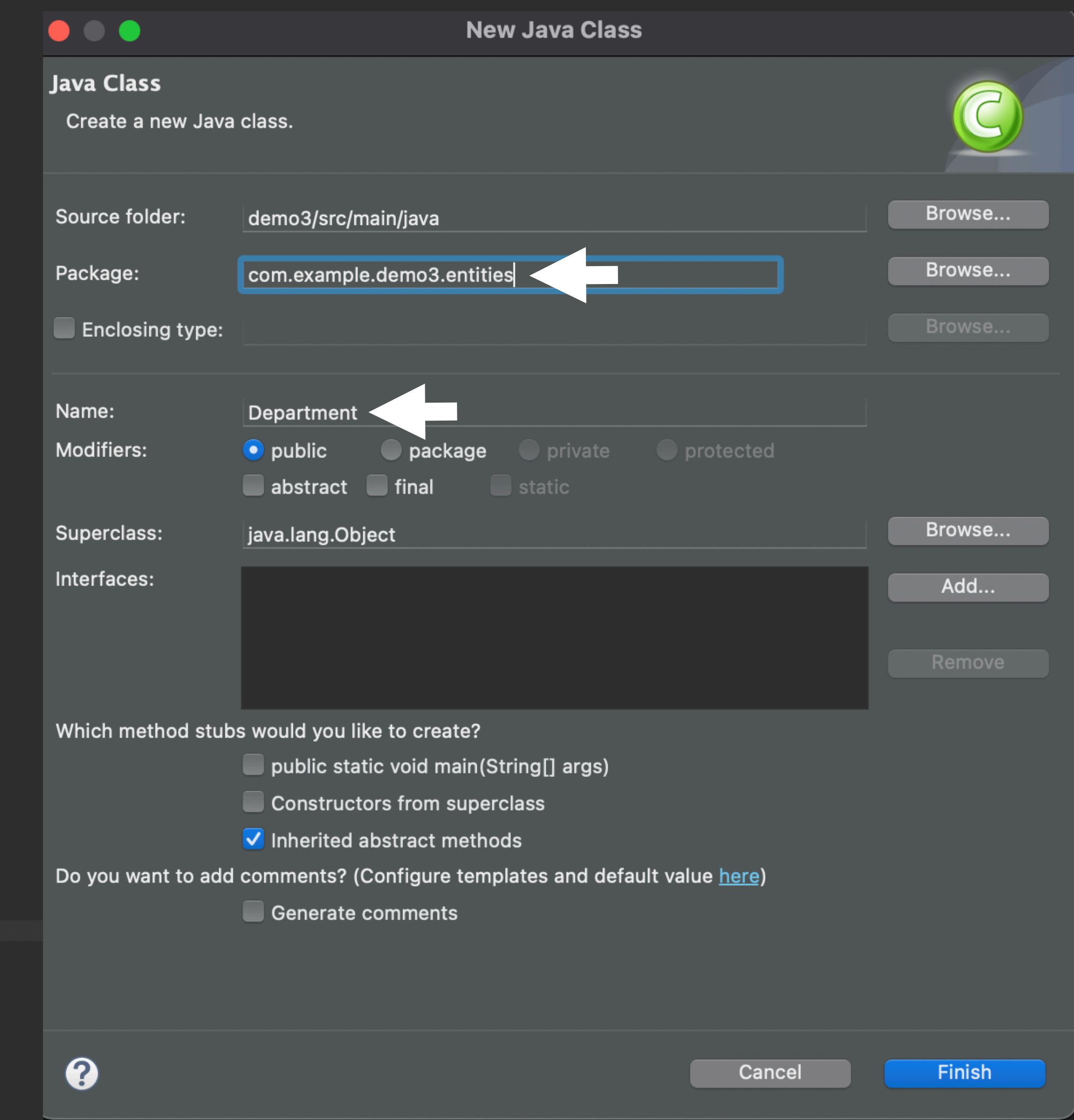
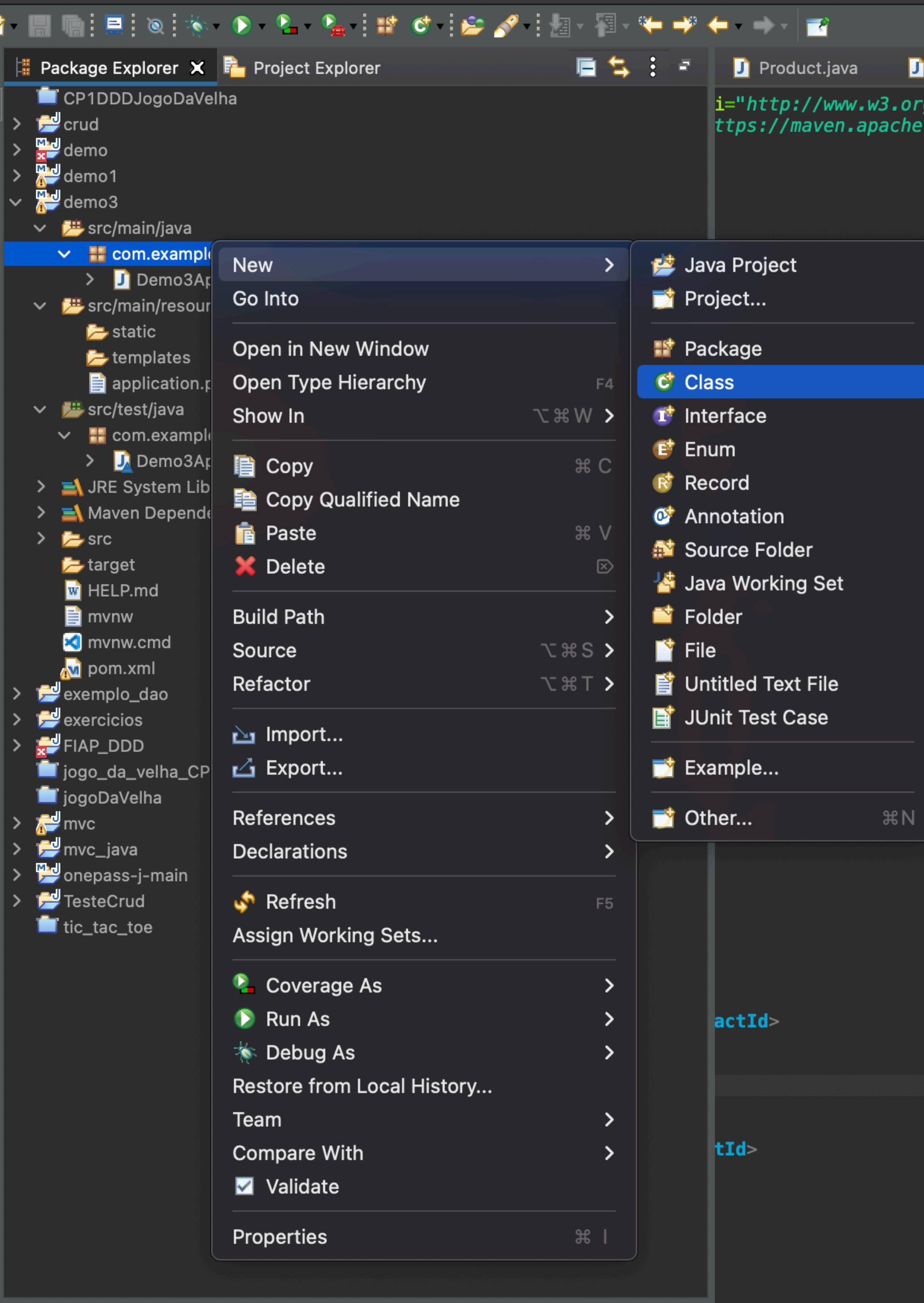
Vamos atualizar o Projeto



Criação do Modelo de Domínio

Modelo de Domínio







```

1 package com.example.demo3.entities;
2
3 public class Department {
4
5     private Long id;
6     private String name;
7
8     public Department() {
9
10    }
11
12    /**
13     * @return the id
14     */
15    public Long getId() {
16        return id;
17    }
18
19    /**
20     * @param id the id to set
21     */
22    public void setId(Long id) {
23        this.id = id;
24    }
25
26
27 }
28
  
```

Classe Department

```

1 package com.example.demo3.entities;
2
3 public class User {
4     private Long id;
5     private String name;
6     private String email;
7
8     //Composição de objetos
9     // Associação entre entidades
10    private Department department;
11
12    public User() {
13
14    }
15
16    /**
17     * @return the id
18     */
19    public Long getId() {
20        return id;
21    }
22
23    /**
24     * @param id the id to set
25     */
26    public void setId(Long id) {
27        this.id = id;
28    }
29
30    /**
31     * @return the name
32     */
33    public String getName() {
  
```

Classe User

Vamos fazer o Mapeamento
Objeto Relacional

Como que os objetos do Java serão
convertidos para o modelo relacional no
Banco de Dados relacional?

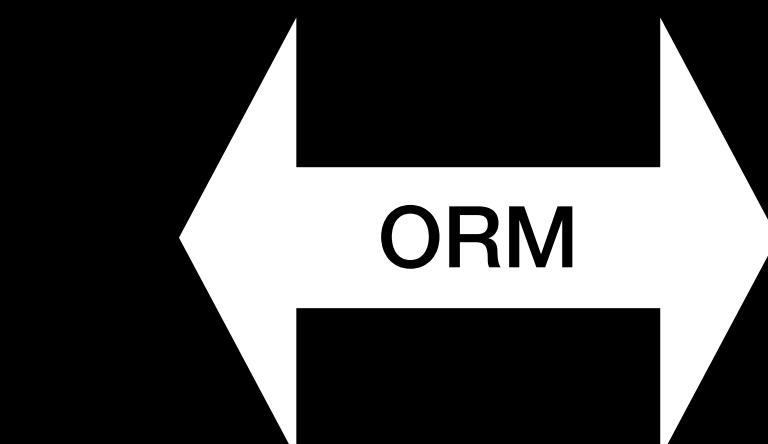
Mapeamento Objeto Relacional -> json

tb_department

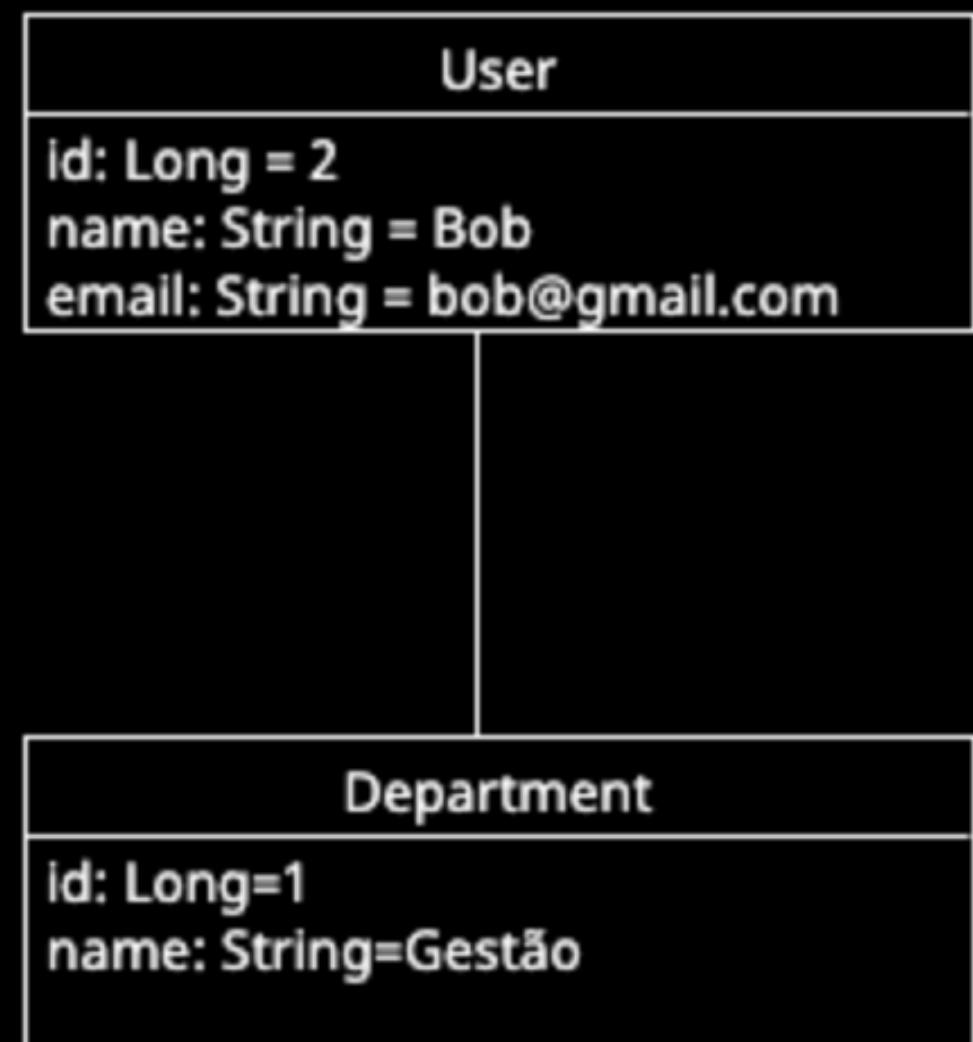
id	name
1	Gestão
2	Informática

tb_user

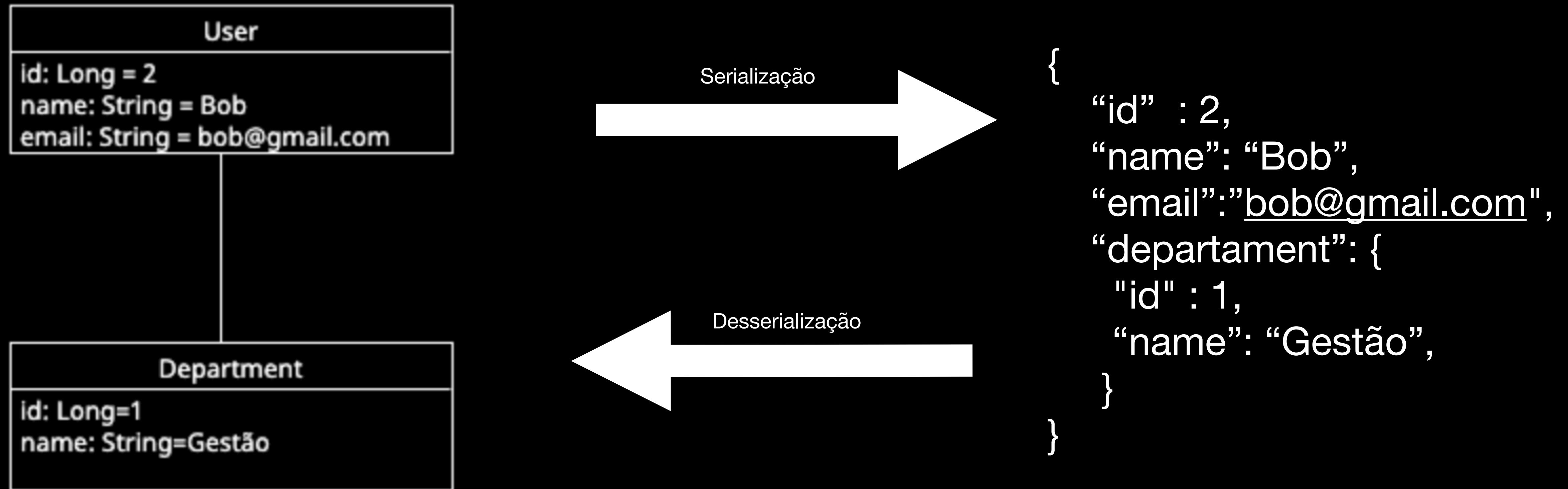
id	name	email	department_id
1	Maria	maria@gmail.com	1
2	Bob	bob@gmail.com	1
3	Alex	alex@gmail.com	2
4	Ana	ana@gmail.com	2



Mapeamento Objeto Relacional
(JPA)



Mapeamento Objeto Relacional -> json



Mapeamento dos objetos



```
1 package com.example.demo3.entities;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.GenerationType;
6 import jakarta.persistence.Id;
7 import jakarta.persistence.Table;
8
9 @Entity
10 @Table(name = "tb_department")
11
12 public class Department {
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private Long id;
16     private String name;
17
18     public Department() {
19
20     }
21
22     /**
23      * @return the id
24      */
25     public Long getId() {
26         return id;
27     }
28
29     /**
30      * @param id the id to set
31      */
32 }
```

Mapeamento da Classe **Department**
para **tb_department**



```
1 package com.example.demo3.entities;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.GenerationType;
6 import jakarta.persistence.Id;
7 import jakarta.persistence.JoinColumn;
8 import jakarta.persistence.ManyToMany;
9 import jakarta.persistence.Table;
10
11 @Entity
12 @Table(name = "tb_user")
13 public class User {
14
15     @Id
16     @GeneratedValue(strategy = GenerationType.IDENTITY)
17     private Long id;
18     private String name;
19     private String email;
20
21
22     @ManyToMany
23     @JoinColumn(name = "department_id") //configurando a chave estrangeira do banco
24     //Composição de objetos
25     // Associação entre entidades
26     private Department department;
27
28     public User() {
29
30 }
```

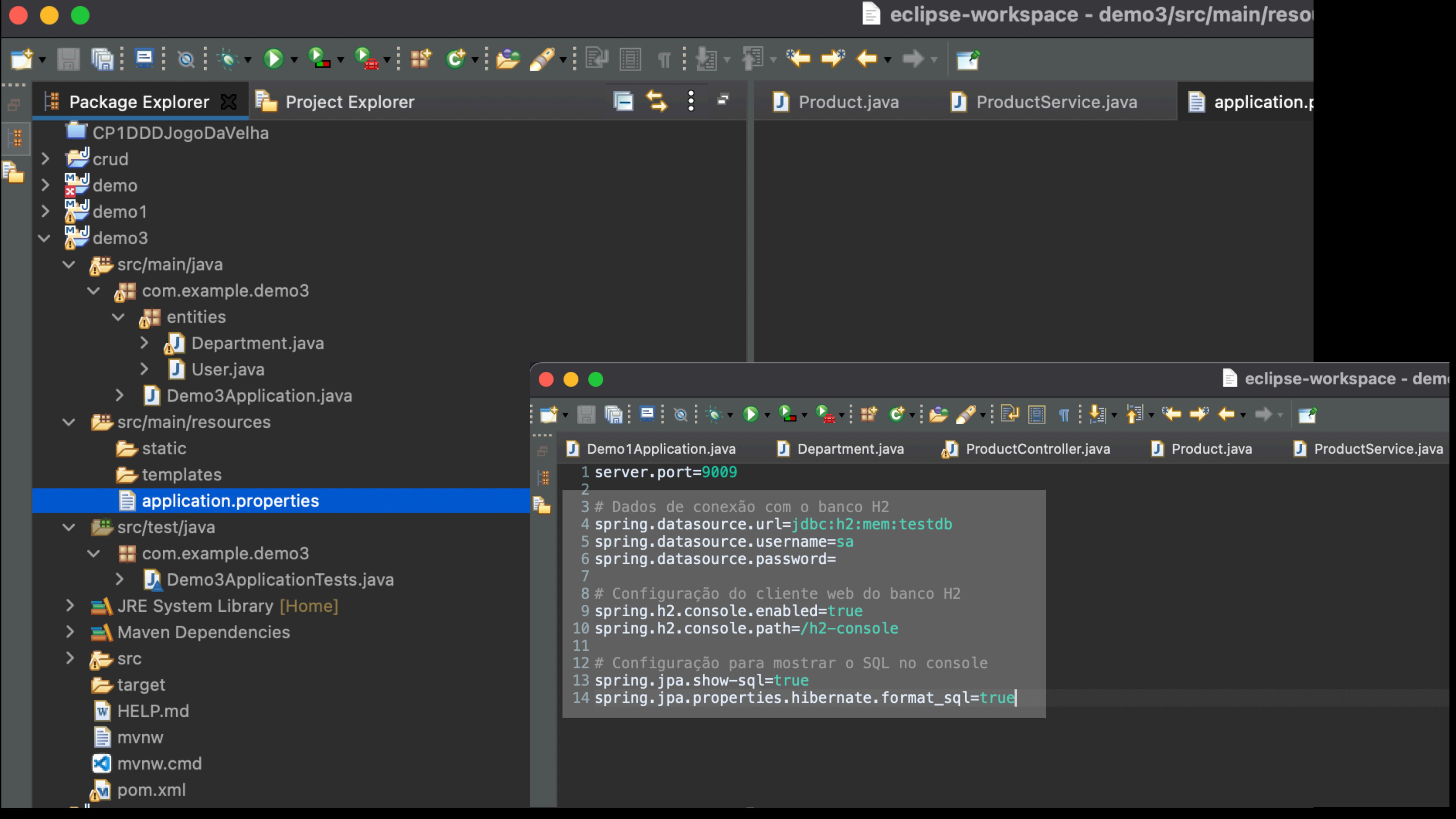
Mapeamento da Classe **User**
para **tb_user**

Configuração do Banco de Dados

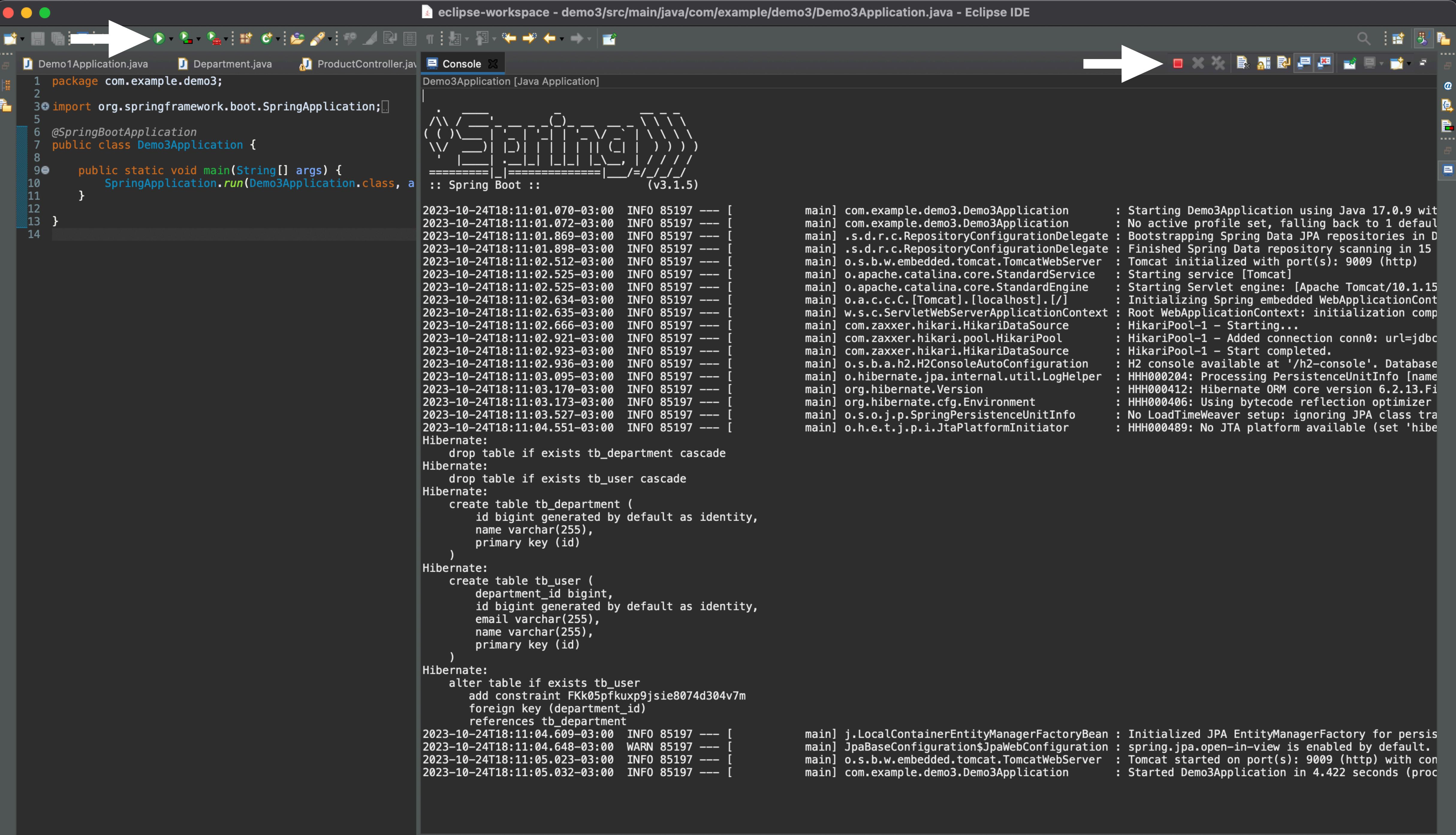
```
# Dados de conexão com o banco H2
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.username=sa
spring.datasource.password=

# Configuração do cliente web do banco H2
spring.h2.console.enabled=true
spring.h2.console.path=/h2-console

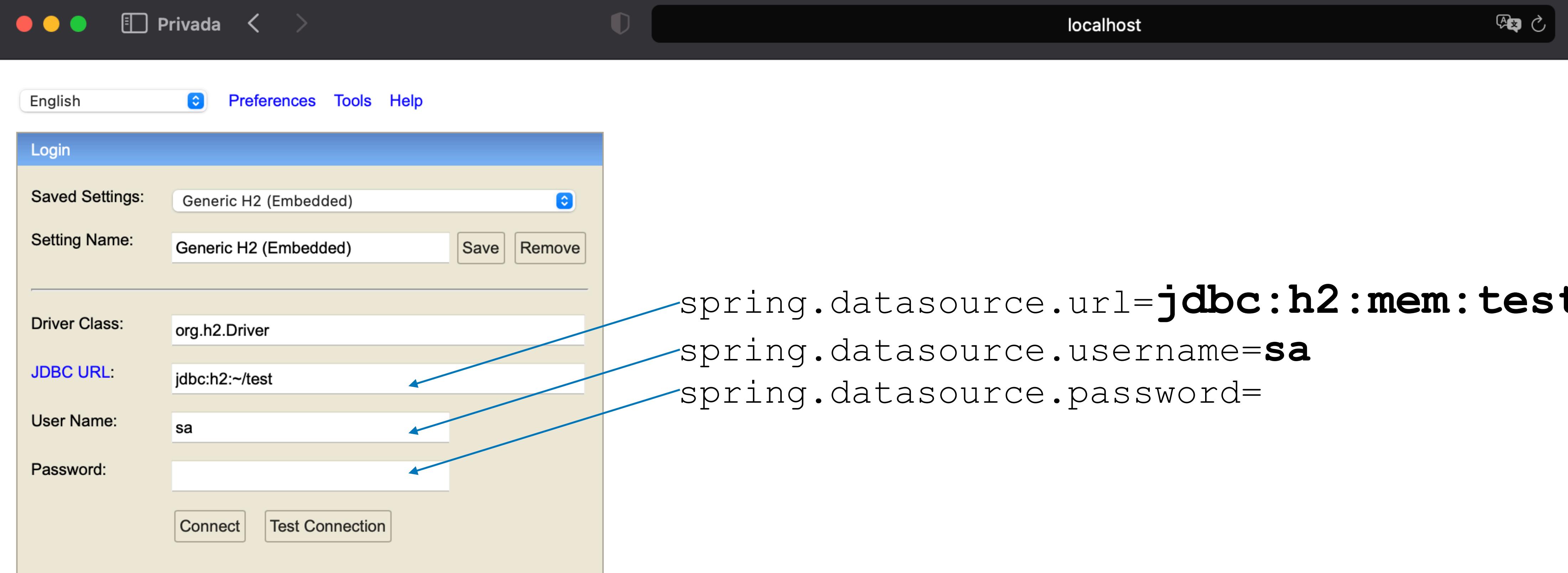
# Configuração para mostrar o SQL no console
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
```



Vamos atualizar a aplicação



Vamos acessar o Banco de Dados



http://localhost:8080/h2-console

H2 Console

localhost:9009/h2-console/login.do?jsessionid=c6f9da03c6358f7bc81f5dae97ccc4c0

Anônima

Auto commit | Max rows: 1000 | Auto complete | Auto select

jdbc:h2:mem:testdb

TB_DEPARTMENT

TB_USER

INFORMATION_SCHEMA

Users

H2 2.1.214 (2022-06-13)

Run | Run Selected | Auto complete | Clear | SQL statement:

Important Commands

	Displays this Help Page
	Shows the Command History
	Executes the current SQL statement
	Shift+Enter Executes the SQL statement defined by the text selection
	Ctrl+Space Auto complete
	Disconnects from the database

Sample SQL Script

Delete the table if it exists	DROP TABLE IF EXISTS TEST;
Create a new table with ID and NAME columns	CREATE TABLE TEST(ID INT PRIMARY KEY, NAME VARCHAR(255));
Add a new row	INSERT INTO TEST VALUES(1, 'Hello');
Add another row	INSERT INTO TEST VALUES(2, 'World');
Query the table	SELECT * FROM TEST ORDER BY ID;
Change data in a row	UPDATE TEST SET NAME='Hi' WHERE ID=1;
Remove a row	DELETE FROM TEST WHERE ID=2;
Help	HELP ...

Adding Database Drivers

Additional database drivers can be registered by adding the Jar file location of the driver to the environment variables H2DRIVERS or CLASSPATH. Example (Windows): to add the database driver library C:/Programs/hsqldb/lib/hsqldb.jar, set the environment variable H2DRIVERS to C:/Programs/hsqldb/lib/hsqldb.jar.

H2 Console

localhost:9009/h2-console/login.do?jsessionid=c6f9da03c6358f7bc81f5dae97ccc4c0

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbch2:mem:testdb

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM TB_DEPARTMENT

TB_DEPARTMENT
TB_USER
INFORMATION_SCHEMA
Users

H2 2.1.214 (2022-06-13)

Important Commands

	Displays this Help Page
	Shows the Command History
	Ctrl+Enter Executes the current SQL statement
	Shift+Enter Executes the SQL statement defined by the text selection
	Ctrl+Space Auto complete
	Disconnects from the database

Sample SQL Script

Delete the table if it exists	DROP TABLE IF EXISTS TEST;
Create a new table with ID and NAME columns	CREATE TABLE TEST(ID INT PRIMARY KEY, NAME VARCHAR(255));
Add a new row	INSERT INTO TEST VALUES(1, 'Hello');
Add another row	INSERT INTO TEST VALUES(2, 'World');
Query the table	SELECT * FROM TEST ORDER BY ID;
Change data in a row	UPDATE TEST SET NAME='Hi' WHERE ID=1;
Remove a row	DELETE FROM TEST WHERE ID=2;
Help	HELP ...

Adding Database Drivers

Additional database drivers can be registered by adding the Jar file location of the driver to the environment variables H2DRIVERS or CLASSPATH. Example (Windows): to add the database driver library C:/Programs/hsqldb/lib/hsqldb.jar, set the environment variable H2DRIVERS to C:/Programs/hsqldb/lib/hsqldb.jar.

H2 Console

localhost:9090/h2-console/login.do?jsessionid=c6f9da03c6358f7bc81f5dae97ccc4c0

Auto commit | Max rows: 1000 | Auto complete Off | Auto select On

jdbc:h2:mem:testdb | TB_DEPARTMENT | TB_USER | INFORMATION_SCHEMA | Users | H2 2.1.214 (2022-06-13)

Run | Run Selected | Auto complete | Clear | SQL statement:

```
SELECT * FROM TB_DEPARTMENT
```

SELECT * FROM TB_DEPARTMENT;

ID	NAME
(no rows, 2 ms)	

Edit

H2 Console

localhost:9090/h2-console/login.do?jsessionid=c6f9da03c6358f7bc81f5dae97ccc4c0

Anônima

Auto commit | Max rows: 1000 | Auto complete Off | Auto select On

jdbc:h2:mem:testdb

TB_DEPARTMENT

TB_USER

INFORMATION_SCHEMA

Users

H2 2.1.214 (2022-06-13)

Run Selected | Auto complete | Clear SQL statement:

SELECT * FROM TB_USER |

SELECT * FROM TB_USER;

DEPARTMENT_ID	ID	EMAIL	NAME
---------------	----	-------	------

(no rows, 1 ms)

Edit

Populando o Banco de Dados

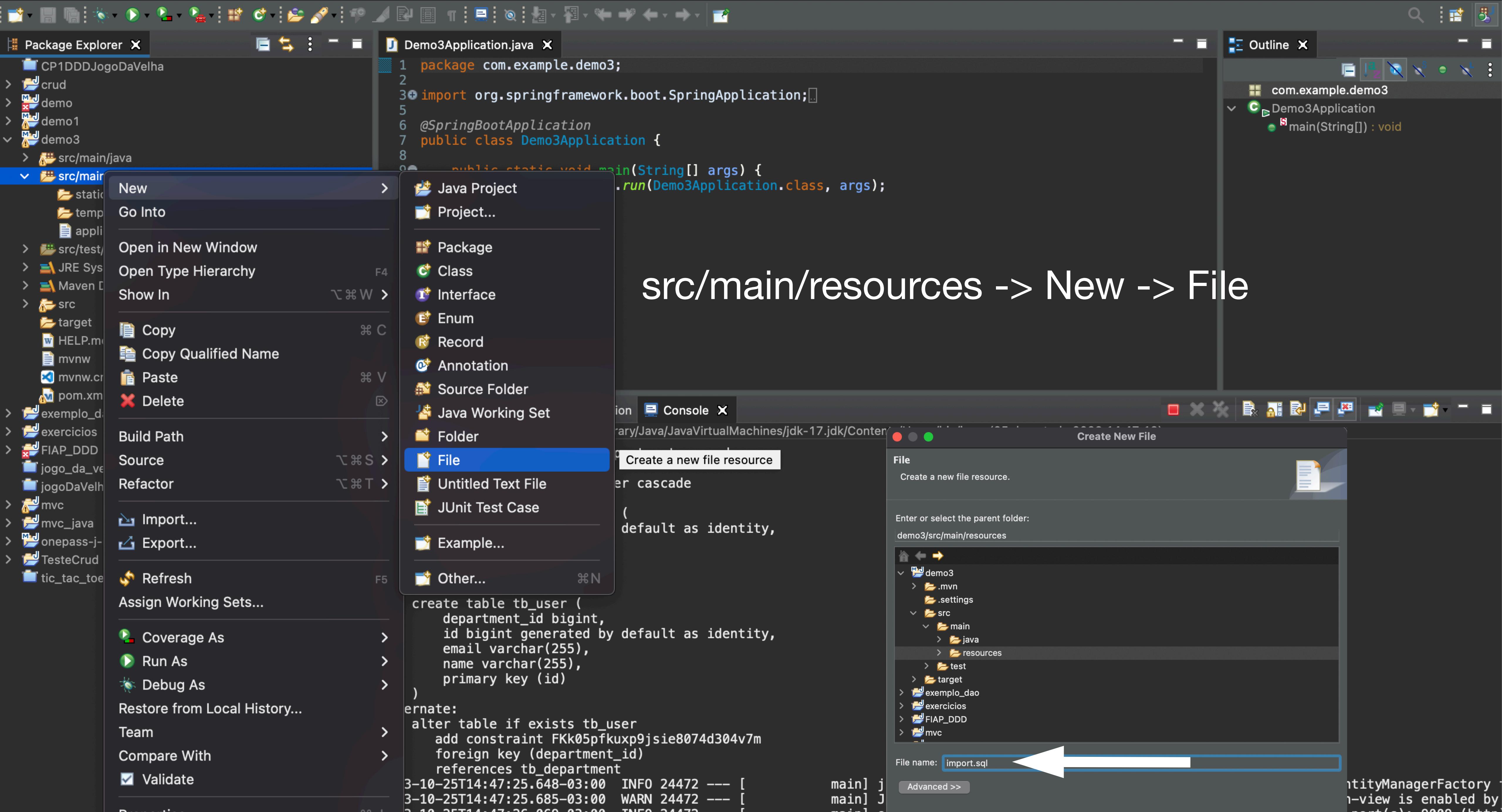
Script SQL

```
INSERT INTO tb_department(name) VALUES ('Gestão');
INSERT INTO tb_department(name) VALUES ('Informática');

INSERT INTO tb_user(department_id, name, email) VALUES (1, 'Maria', 'maria@gmail.com');
INSERT INTO tb_user(department_id, name, email) VALUES (1, 'Bob', 'bob@gmail.com');
INSERT INTO tb_user(department_id, name, email) VALUES (2, 'Alex', 'alex@gmail.com');
INSERT INTO tb_user(department_id, name, email) VALUES (2, 'Ana', 'ana@gmail.com');
```

```
INSERT INTO tb_department(name) VALUES ('Gestão');
INSERT INTO tb_department(name) VALUES ('Informática');
```

```
INSERT INTO tb_user(department_id, name, email) VALUES (1, 'Maria', 'maria@gmail.com');
INSERT INTO tb_user(department_id, name, email) VALUES (1, 'Bob', 'bob@gmail.com');
INSERT INTO tb_user(department_id, name, email) VALUES (2, 'Alex', 'alex@gmail.com');
INSERT INTO tb_user(department_id, name, email) VALUES (2, 'Ana', 'ana@gmail.com');
```



Create New File

File

Create a new file resource.



Enter or select the parent folder:

demo3/src/main/resources



- demo3
 - .mvn
 - .settings
 - src
 - main
 - java
 - resources
 - test
 - target
- exemplo_dao
- exercicios
- FIAP_DDD
- mvc

File name:

Advanced >>

File name: **import.sql**



Cancel

Finish

eclipse-workspace - demo3/src/main/resources/import.sql - Eclipse IDE

The screenshot shows the Eclipse IDE interface. The title bar reads "eclipse-workspace - demo3/src/main/resources/import.sql - Eclipse IDE". The toolbar has various icons for file operations, code navigation, and project management. The Package Explorer view on the left shows a project structure with several packages and files, including "src/main/java" and "src/main/resources/import.sql". The central workspace contains two tabs: "Demo3Application.java" and "*import.sql x". The "*import.sql" tab is active, displaying the following SQL script:

```
1 INSERT INTO tb_department(name) VALUES ('Gestão');
2 INSERT INTO tb_department(name) VALUES ('Informática');
3
4 INSERT INTO tb_user(department_id, name, email) VALUES (1, 'Maria', 'maria@gmail.com');
5 INSERT INTO tb_user(department_id, name, email) VALUES (1, 'Bob', 'bob@gmail.com');
6 INSERT INTO tb_user(department_id, name, email) VALUES (2, 'Alex', 'alex@gmail.com');
7 INSERT INTO tb_user(department_id, name, email) VALUES (2, 'Ana', 'ana@gmail.com');
```

Below the workspace, there is a large text area with the following instructions:

Copie o Script SQL aqui...
ATENÇÃO para o nome do ARQUIVO (import.sql)

eclipse-workspace - demo3/src/main/java/com/example/demo3/Demo3Application.java - Eclipse IDE

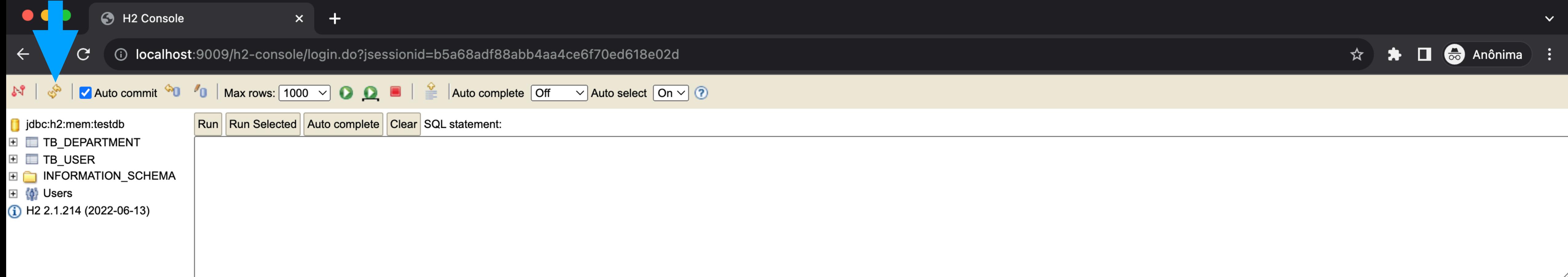
The screenshot shows the Eclipse IDE interface with the following details:

- Package Explorer:** Shows the project structure with packages like CP1DDDDJogoDaVelha, crud, demo, demo1, demo3, src/main/java (containing com.example.demo3 entities and Demo3Application.java), and src/main/resources.
- Editor:** Displays the file `Demo3Application.java` containing the main method of a Spring Boot application.
- Outline:** Shows the class structure of `Demo3Application`.
- Console:** Shows the output of the application's execution, including Hibernate SQL logs and Spring Boot startup messages.

```
package com.example.demo3;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
public class Demo3Application {
    public static void main(String[] args) {
        SpringApplication.run(Demo3Application.class, args);
    }
}
```

```
2023-10-25T15:23:49.662-03:00  INFO 31732 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator      : HHH000489: No JTA platform available
Hibernate:
    drop table if exists tb_department cascade
Hibernate:
    drop table if exists tb_user cascade
Hibernate:
    create table tb_department (
        id bigint generated by default as identity,
        name varchar(255),
        primary key (id)
    )
Hibernate:
    create table tb_user (
        department_id bigint,
        id bigint generated by default as identity,
        email varchar(255),
        name varchar(255),
        primary key (id)
    )
Hibernate:
    alter table if exists tb_user
        add constraint FKk05pfkuxp9jsie8074d304v7m
        foreign key (department_id)
        references tb_department
Hibernate: INSERT INTO tb_department(name) VALUES ('Gestão')
Hibernate: INSERT INTO tb_department(name) VALUES ('Informática')
Hibernate: INSERT INTO tb_user(department_id, name, email) VALUES (1, 'Maria', 'maria@gmail.com')
Hibernate: INSERT INTO tb_user(department_id, name, email) VALUES (1, 'Bob', 'bob@gmail.com')
Hibernate: INSERT INTO tb_user(department_id, name, email) VALUES (2, 'Alex', 'alex@gmail.com')
Hibernate: INSERT INTO tb_user(department_id, name, email) VALUES (2, 'Ana', 'ana@gmail.com')
2023-10-25T15:23:49.711-03:00  INFO 31732 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory
2023-10-25T15:23:49.744-03:00  WARN 31732 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. This is discouraged.
2023-10-25T15:23:50.078-03:00  INFO 31732 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 9009 (http)
2023-10-25T15:23:50.086-03:00  INFO 31732 --- [main] com.example.demo3.Demo3Application : Started Demo3Application in 3.536 seconds
```

Rode a aplicação



A screenshot of the H2 Console interface. At the top, there's a toolbar with a blue arrow icon pointing down, a refresh button, and a search bar containing 'localhost:9009/h2-console/login.do?jsessionid=b5a68adf88abb4aa4ce6f70ed618e02d'. On the right side of the toolbar are icons for star, puzzle, square, and a user profile labeled 'Anônima'. Below the toolbar, there are several status indicators: 'Auto commit' (checked), 'Max rows: 1000', and 'Auto complete' (set to Off). The main workspace shows a database connection named 'jdbc:h2:mem:testdb'. To the left is a sidebar with a tree view of tables: 'TB_DEPARTMENT', 'TB_USER', 'INFORMATION_SCHEMA', and 'Users'. At the bottom left, it says 'H2 2.1.214 (2022-06-13)'. The main area has buttons for 'Run', 'Run Selected', 'Auto complete', and 'Clear'. A text input field is labeled 'SQL statement:'.

Important Commands

	Displays this Help Page
	Shows the Command History
	Ctrl+Enter Executes the current SQL statement
	Shift+Enter Executes the SQL statement defined by the text selection
	Ctrl+Space Auto complete
	Disconnects from the database

Sample SQL Script

Delete the table if it exists	DROP TABLE IF EXISTS TEST;
Create a new table with ID and NAME columns	CREATE TABLE TEST(ID INT PRIMARY KEY, NAME VARCHAR(255));
Add a new row	INSERT INTO TEST VALUES(1, 'Hello');
Add another row	INSERT INTO TEST VALUES(2, 'World');
Query the table	SELECT * FROM TEST ORDER BY ID;
Change data in a row	UPDATE TEST SET NAME='Hi' WHERE ID=1;
Remove a row	DELETE FROM TEST WHERE ID=2;
Help	HELP ...

Adding Database Drivers

Additional database drivers can be registered by adding the Jar file location of the driver to the environment variables H2DRIVERS or CLASSPATH. Example (Windows): to add the database driver library C:/Programs/hsqldb/lib/hsqldb.jar, set the environment variable H2DRIVERS to C:/Programs/hsqldb/lib/hsqldb.jar.

H2 Console

localhost:9009/h2-console/login.do?jsessionid=b5a68adf88abb4aa4ce6f70ed618e02d

Anônima

Auto commit | Max rows: 1000 | Auto complete | Auto select

jdbc:h2:mem:testdb

TB_DEPARTMENT

TB_USER

INFORMATION_SCHEMA

Users

2.1.214 (2022-06-13)

SQL statement:

```
SELECT * FROM TB_DEPARTMENT
```

SELECT * FROM TB_DEPARTMENT;

ID	NAME
1	Gestão
2	Informática

(2 rows, 2 ms)

Edit

The screenshot shows the H2 Console interface with a database named 'testdb'. The left sidebar lists tables: TB_DEPARTMENT, TB_USER, INFORMATION_SCHEMA, Users, and a version entry. The main area displays a SQL statement 'SELECT * FROM TB_DEPARTMENT' and its results. The results show two rows: ID 1 with NAME 'Gestão' and ID 2 with NAME 'Informática'. A large blue box highlights the entire SQL statement and the results table. Two arrows point from the schema section at the top left towards the highlighted area.

H2 Console

localhost:9009/h2-console/login.do?jsessionid=b5a68adf88abb4aa4ce6f70ed618e02d

Auto commit | Max rows: 1000 | Auto complete | Auto select

jdb:h2:mem:testdb
+ TB_DEPARTMENT
+ TB_USER ←
+ INFORMATION_SCHEMA
+ Users
H2 2.1.214 (2022-06-13)

Run | Run Selected | Auto complete | Clear | SQL statement:

```
SELECT * FROM TB_USER
```

SELECT * FROM TB_USER;

DEPARTMENT_ID	ID	EMAIL	NAME
1	1	maria@gmail.com	Maria
1	2	bob@gmail.com	Bob
2	3	alex@gmail.com	Alex
2	4	ana@gmail.com	Ana

(4 rows, 0 ms)

Edit

Criando os endpoint API Rest

```
18      </properties>
19 ❸  <dependencies>
20 ❹    <dependency>
21        <groupId>org.springframework.boot</groupId>
22        <artifactId>spring-boot-starter-data-jpa</artifactId>
23    </dependency>
24 ❺    <dependency>
25        <groupId>org.springframework.boot</groupId>
26        <artifactId>spring-boot-starter-web</artifactId>
```

Sub framework Spring que usa a JPA do Java

The screenshot shows the Eclipse IDE interface with the following details:

- Toolbar:** Standard Eclipse toolbar with various icons for file operations, search, and navigation.
- Package Explorer:** Shows a project structure with several packages and source files.
- Current File:** The active file is `Demo3Application.java`, which contains the following code:

```
1 package com.example.demo3;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 public class Demo3Application {
7
8     public static void main(String[] args) {
9         SpringApplication.run(Demo3Application.class, args);
10    }
11 }
```
- Context Menu:** A context menu is open over the code, with the "Interface" option highlighted. The tooltip for "Interface" says "Create a Java interface". Other options in the menu include Java Project, Project..., Package, Class, Copy, Paste, Delete, and others.
- Bottom Status Bar:** Shows the status bar with tabs for Problems, Javadoc, Declaration, and Console, along with some log output.
- Bottom Right:** A large text overlay reads "Vamos criar uma nova interface...".

eclipse-workspace - demo3/src/main/java/com/example/demo3/repositories/UserRepository.java - Eclipse IDE

Package Explorer Demo3Application.java import.sql UserRepository.java

```
1 package com.example.demo3.repositories;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 import com.example.demo3.entities.User;
6
7 public interface UserRepository extends JpaRepository<User, Long>{
8
9 }
10
```

Upward arrows pointing to the imports and the interface declaration.

Problems Javadoc Declaration Console

Demo3Application [Java Application] 2023-10-25T15:23:48.647-03:00 INFO 31732 --- [main] o.s.d.j.p.SpringResourceDatabaseInit 2023-10-25T15:23:49.662-03:00 INFO 31732 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available

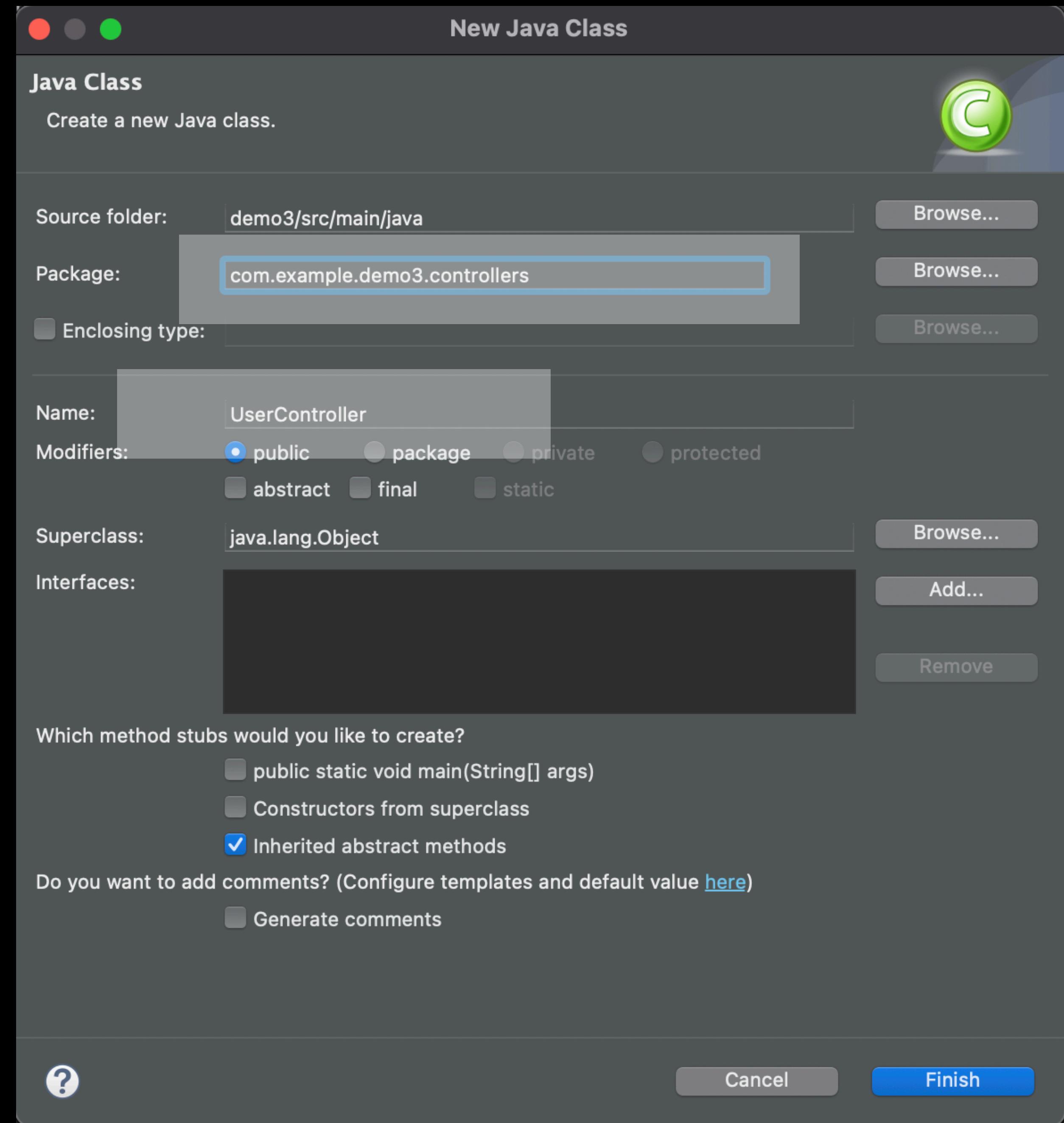
Vamos criar os endpoints

Controlador REST

Objeto responsável por receber as requisições dos usuários e responder as requisições

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with several Java packages and source files.
- Current File:** UserRepository.java is open in the editor.
- Context Menu:** A context menu is open over the code, specifically over the declaration of the UserRepository interface.
- Context Menu Options:**
 - New > Class (highlighted)
 - Java Project
 - Project...
 - Package
 - Interface
 - Enum
 - Record
 - Annotation
 - Source Folder
 - Java Working Set
 - Folder
 - File
 - Untitled Text File
 - JUnit Test Case
 - Example...
 - Other...
- Editor Content:** The code defines a UserRepository interface extending JpaRepository<User, Long>. It imports org.springframework.data.jpa.repository.JpaRepository and com.example.demo3.entities.User.
- Bottom Status Bar:** Shows application logs and system status.



The screenshot shows the Eclipse IDE interface with the following details:

- Left Side (Package Explorer):** Displays the project structure. The `demo3` project is expanded, showing its contents:
 - `src/main/java`:
 - `com.example.demo3`:
 - `controllers`: `UserController.java`
 - `entities`
 - `repositories`: `UserRepository.java`
 - `Demo3Application.java`
 - `src/main/resources`:
 - `static`
 - `templates`
 - `application.properties`
 - `import.sql`
 - `src/test/java`
 - `JRE System Library [Home]`
 - `Maven Dependencies`
 - `src`:
 - `target`
 - `HELP.md`
 - `mvnw`
 - `mvnw.cmd`
 - `pom.xml`
 - `exemplo_dao`
 - `exercicios`
 - `FIAP_DDD`
 - `jogo_da_velha_CP2`
 - `jogoDaVelha`
 - `mvc`
 - `mvc_java`
 - `onepass-j-main`
 - `TesteCrud`
 - `tic_tac_toe`
- Center (Editor Area):** The `UserController.java` file is open in the editor. The code is as follows:

```
1 package com.example.demo3.controllers;
2
3 public class UserController {
4
5 }
6
```
- Bottom (Status Bar):** Shows the following information:
 - `Demo3Application [Java Application]`
 - `2023-10-25T15:23:49.662-03:00 INFO 31732 --- [main] o.s.s.j.p.SpringContextReloader : main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available`

The screenshot shows the Eclipse IDE interface with the following details:

- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and navigation.
- Package Explorer:** Shows the project structure with the following packages:
 - CP1DDDJogoDaVelha
 - crud
 - demo
 - demo1
 - demo3 (selected)
 - src/main/java
 - com.example.demo3
 - controllers (UserController.java selected)
 - entities
 - repositories (UserRepository.java)
 - Demo3Application.java
 - src/main/resources
 - static
 - templates
 - application.properties
 - import.sql
 - src/test/java
 - JRE System Library [Home]
 - Maven Dependencies
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - exemplo_dao
 - exercicios
 - FIAP_DDD
 - jogo_da_velha_CP2
 - jogoDaVelha
 - mvc
 - mvc_java
 - onepass-j-main
 - TesteCrud
 - tic_tac_toe
- Editor:** The main editor window displays the `UserController.java` file content:

```
1 package com.example.demo3.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.RequestMapping;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import com.example.demo3.entities.User;
11 import com.example.demo3.repositories.UserRepository;
12
13 @RestController
14 @RequestMapping(value = "/users") //caminho que o controlador irá responder
15 public class UserController {
16
17     @Autowired //injeção de dependência
18     private UserRepository repository;
19
20     @GetMapping //requisição GET
21     public List<User> findAll(){
22         //acesso ao UserRepository
23         List<User> result = repository.findAll();
24         return result;
25     }
26 }
```

- Bottom Status Bar:** Shows the following information:
 - Demo3Application [Java Application]
 - 2023-10-25T15:23:49.662-03:00 INFO 31732 --- [main] o.s.s.j.p.SpringResourceListenerContainer : No LoadTimeWeaver setup. Ignoring JPA
 - 2023-10-25T15:23:49.662-03:00 INFO 31732 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available

Reinicie a Aplicação

```

1 package com.example.demo3;
2
3+import org.springframework.boot.SpringApplication;
5
6 @SpringBootApplication
7 public class Demo3Application {
8
9+    public static void main(String[] args) {
10        SpringApplication.run(Demo3Application.class, args);
11    }
12
13 }
14

```

023-10-25T16:01:18.817-03:00 INFO 38983 --- [main] com.example.demo3.Demo3Application : Starting Demo3Application using Java 1
023-10-25T16:01:18.820-03:00 INFO 38983 --- [main] com.example.demo3.Demo3Application : No active profile set, falling back to
023-10-25T16:01:19.442-03:00 INFO 38983 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA reposito

http://localhost:9009/users



A screenshot of a web browser window. The address bar shows 'localhost:9009/users'. The page content displays a JSON array:

```
[{"id":1,"name":"Maria","email":"maria@gmail.com","department":{"id":1}},{ "id":2,"name":"Bob","email":"bob@gmail.com","department":{"id":1}},{ "id":3,"name":"Alex","email":"alex@gmail.com","department":{"id":1}},{ "id":4,"name":"Ana","email":"ana@gmail.com","department":{"id":2}}]
```

↑
json

← → C Tabbed Postman - REST Client | chrome-extension://coohjcphdfgbiolnekdpbcijmhambjff/index.html

POSTMAN

Normal Basic Auth Digest Auth OAuth 1.0 No environment ▾

http://localhost:9009/users GET URL params

History Collections

You haven't created any collections yet. Collections let you group requests together for quick access.

Send Preview Add to collection

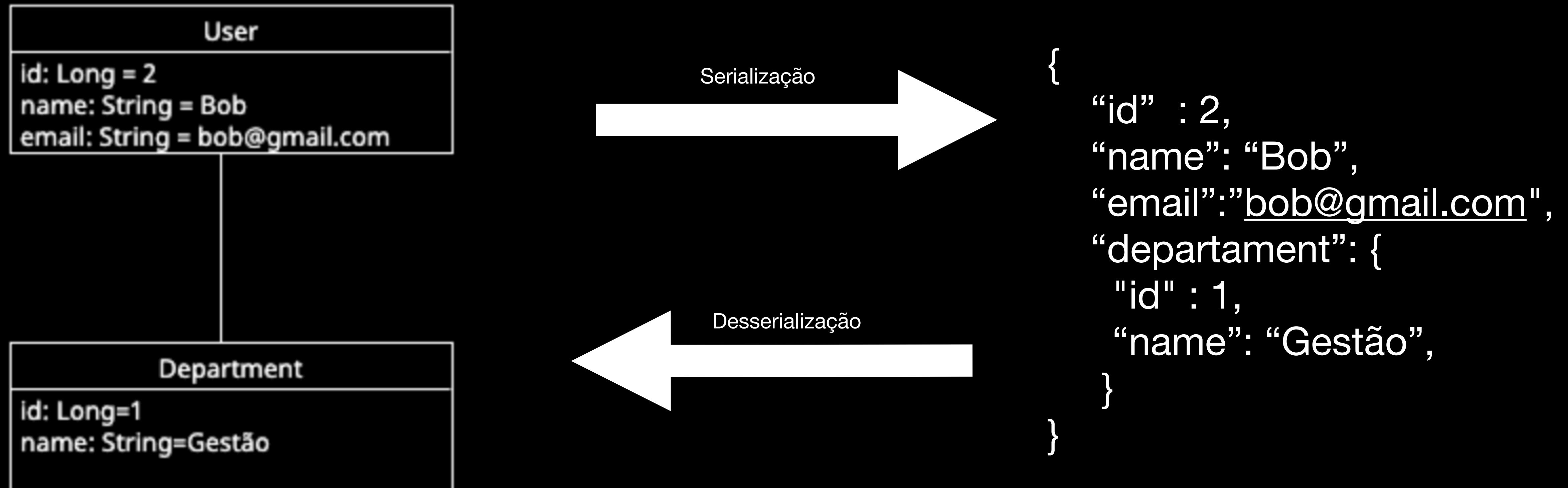
Body Cookies (undefined) Headers (5) STATUS 200 TIME 2308 ms

Pretty Raw Preview JSON XML

```
1 [  
2 {  
3     "id": 1,  
4     "name": "Maria",  
5     "email": "maria@gmail.com",  
6     "department": {  
7         "id": 1  
8     }  
9 },  
10 {  
11     "id": 2,  
12     "name": "Bob",  
13     "email": "bob@gmail.com",  
14     "department": {  
15         "id": 1  
16     }  
17 },  
18 {  
19     "id": 3,  
20     "name": "Alex",  
21     "email": "alex@gmail.com",  
22     "department": {  
23         "id": 2  
24     }  
25 },  
26 {  
27     "id": 4,  
28     "name": "Ana",  
29     "email": "ana@gmail.com",  
30     "department": {  
31         "id": 2  
32     }  
33 }  
34 ]
```

Objetos serializados

Modelagem Relacional -> Objeto -> json



```
List<User> result = repository.findAll();
```

Buscando um usuário

eclipse-workspace - demo3/src/main/java/com/example/demo3/controllers/UserController.java - Eclipse IDE

Package Explorer Demo3Application.java import.sql UserRepository.java UserController.java

```
14 @RestController
15 @RequestMapping(value = "/users") //caminho que o controlador irá responder
16 public class UserController {
17
18     @Autowired //injeção de dependência
19     private UserRepository repository;
20
21     @GetMapping //requisição GET
22     public List<User> findAll(){
23         //acesso ao UserRepository
24         List<User> result = repository.findAll();
25         return result;
26     }
27
28     @GetMapping(value = "/{id}")
29     public User findById(@PathVariable Long id){
30         //acesso ao UserRepository
31         User result = repository.findById(id).get();
32         return result;
33     }
34
35 }
36
```

Tratamento de exceções é bem vindo aqui...

Problems Javadoc Declaration Console

No consoles to display at this time.

Vamos testar...

Reinicie a Aplicação

```

1 package com.example.demo3;
2
3+import org.springframework.boot.SpringApplication;
5
6 @SpringBootApplication
7 public class Demo3Application {
8
9+    public static void main(String[] args) {
10        SpringApplication.run(Demo3Application.class, args);
11    }
12
13 }
14

```

023-10-25T16:01:18.817-03:00 INFO 38983 --- [main] com.example.demo3.Demo3Application : Starting Demo3Application using Java 1
023-10-25T16:01:18.820-03:00 INFO 38983 --- [main] com.example.demo3.Demo3Application : No active profile set, falling back to
023-10-25T16:01:19.442-03:00 INFO 38983 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA reposito

Normal Basic Auth Digest Auth OAuth 1.0 No environment ▾

http://localhost:9009/users/1

Send Preview Add to collection

Body Cookies (undefined) Headers (5) STATUS 200 TIME 284 ms

Pretty Raw Preview JSON XML

```
1 {  
2   "id": 1,  
3   "name": "Maria",  
4   "email": "maria@gmail.com",  
5   "department": {  
6     "id": 1  
7   }  
8 }
```



Normal Basic Auth Digest Auth OAuth 1.0 No environment ▾

http://localhost:9009/users/3

Send Preview Add to collection

Body Cookies (undefined) Headers (5) STATUS 200 TIME 12 ms

Pretty Raw Preview JSON XML

```
1 {  
2   "id": 3,  
3   "name": "Alex",  
4   "email": "alex@gmail.com",  
5   "department": {  
6     "id": 2  
7   }  
8 }
```



Salvando um usuário

Reinicie a Aplicação

```

1 package com.example.demo3;
2
3+import org.springframework.boot.SpringApplication;
5
6 @SpringBootApplication
7 public class Demo3Application {
8
9+    public static void main(String[] args) {
10        SpringApplication.run(Demo3Application.class, args);
11    }
12
13 }
14

```

023-10-25T16:01:18.817-03:00 INFO 38983 --- [main] com.example.demo3.Demo3Application : Starting Demo3Application using Java 1
023-10-25T16:01:18.820-03:00 INFO 38983 --- [main] com.example.demo3.Demo3Application : No active profile set, falling back to
023-10-25T16:01:19.442-03:00 INFO 38983 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA reposito

eclipse-workspace - demo3/src/main/java/com/example/demo3/controllers/UserController.java - Eclipse IDE

Package Explorer Demo3Application.java import.sql UserRepository.java UserController.java

```
private UserRepository repository;

@GetMapping //requisição GET
public List<User> findAll(){
    //acesso ao UserRepository
    List<User> result = repository.findAll();
    return result;
}

@GetMapping(value = "/{id}")
public User findById(@PathVariable Long id){
    //acesso ao UserRepository
    User result = repository.findById(id).get();
    return result;
}

@PostMapping
public User insert(@RequestBody User user){
    User result = repository.save(user);
    return result;
}
```

Problems Javadoc Declaration Console

Demo3Application [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java (25 de out. de 2023 16:24:45)

Hibernate:

```
select
    u1_.id,
    d1_.id,
    d1_.name,
    u1_.email,
    u1_.name
from
    tb_user u1_
left join
    tb_department d1_
        on d1_.id=u1_.department_id
where
    u1_.id=?
```

Writable Smart Insert 41 : 6 : 1234

Normal Basic Auth Digest Auth OAuth 1.0 No environment ▾

http://localhost:9009/users 

form-data x-www-form-urlencoded raw

Key Value 

Send Preview Add to collection

Body Cookies (undefined) Headers (5) STATUS 200 TIME 12 ms

Pretty Raw 

```
{  
  "name": "Fernando",  
  "email": "fernando@gmail.com",  
  "department": {  
    "id": 2  
  }  
}
```

Normal Basic Auth Digest Auth OAuth 1.0 No environment ▾

http://localhost:9009/users POST URL params Headers (0)

form-data x-www-form-urlencoded raw JSON

```
1 {  
2   "id": 1,  
3   "name": "Fernando",  
4   "email": "fernando@gmail.com",  
5   "department": {  
6     "id": 1  
7   }  
8 }
```

Send Preview Add to collection Reset

Normal Basic Auth Digest Auth OAuth 1.0 No environment ▾

0

http://localhost:9009/users

POST ▾

URL params

Headers (1)

Content-Type

application/json



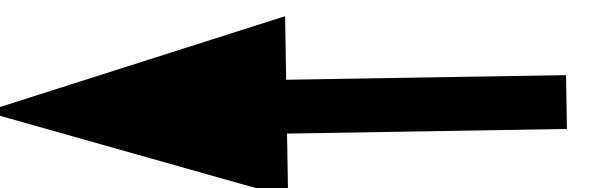
Manage presets

Header

Value

form-data x-www-form-urlencoded raw **JSON** ▾

```
1 {  
2     "name": "Joaquim",  
3     "email": "joaquim@gmail.com",  
4     "department": {  
5         "id": 1  
6     }  
7 }  
8 }
```



Send

Preview

Add to collection

Reset

Body

Cookies (undefined)

Headers (5)

STATUS 200 TIME 9 ms

Pretty

Raw

Preview



```
{  
    "id": 6,  
    "name": "Joaquim",  
    "email": "joaquim@gmail.com",  
    "department": {  
        "id": 1  
    }  
}
```

Content-Type

application/json



Manage presets

Header

Value

Send

Preview

Add to collection

Reset

Body

Cookies (undefined)

Headers (5)

STATUS 200 TIME 12 ms

Pretty

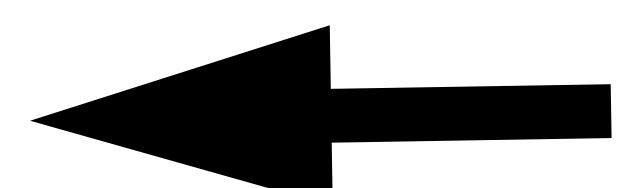
Raw

Preview



```
},  
{  
  "id": 3,  
  "name": "Alex",  
  "email": "alex@gmail.com",  
  "department": {  
    "id": 2  
  }  
,  
{  
  "id": 4,  
  "name": "Ana",  
  "email": "ana@gmail.com",  

```



Ótimos Estudos!!!