

SPRINGBOOT 2022

NANO

LINKS DE AJUDA

<https://github.com/devsuperior/java-web-spring-2022>

<https://www.youtube.com/watch?v=D4frmlHAxEY>

YouTube (92) Seu primeiro projeto Java WE X Spring Initializr GitHub - devsuperior/java-web-s X +

start.spring.io

c++ C++ C++ br-C.org C++ VISUAL STUDIO Domestika ASSEMBLY ss64 kernel org Linux kernel gcc archlinux die.net devdocs Outros favoritos

Project Language Dependencies ADD DEPENDENCIES... CTRL + B

Gradle - Groovy Java Kotlin

Gradle - Kotlin Maven Groovy

Spring Boot

3.0.2 (SNAPSHOT) 3.0.1 2.7.9 (SNAPSHOT)

2.7.8

Project Metadata

Group com.devsuperior

Artifact userdept

Name userdept

Description Aula Java Spring Boot

Package name com.devsuperior.userdept

Packaging Jar War

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

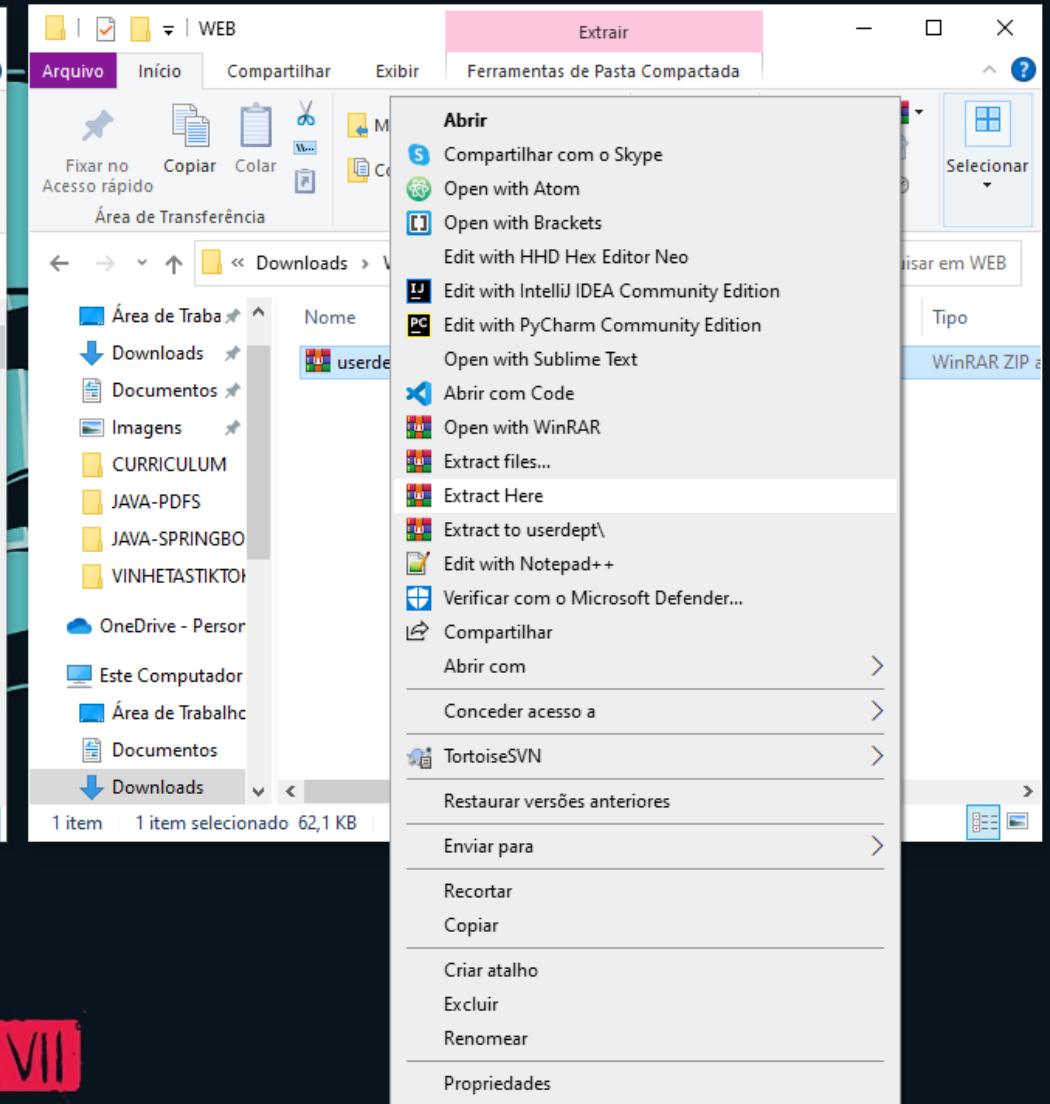
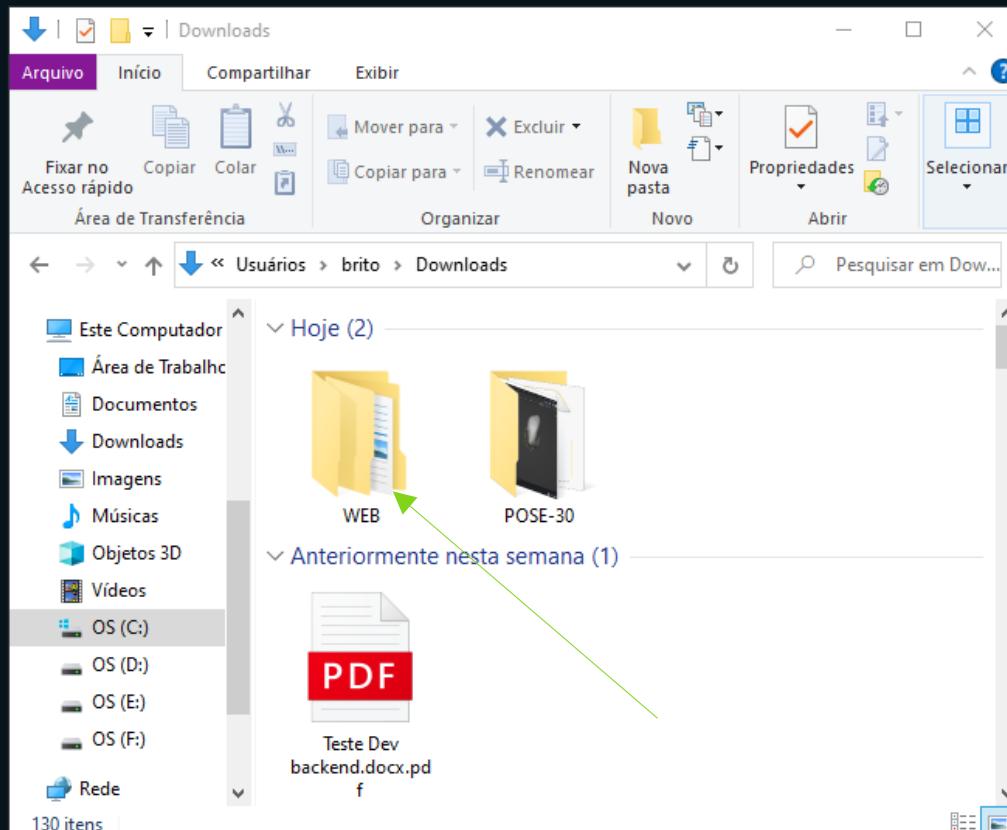
H2 Database SQL Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

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

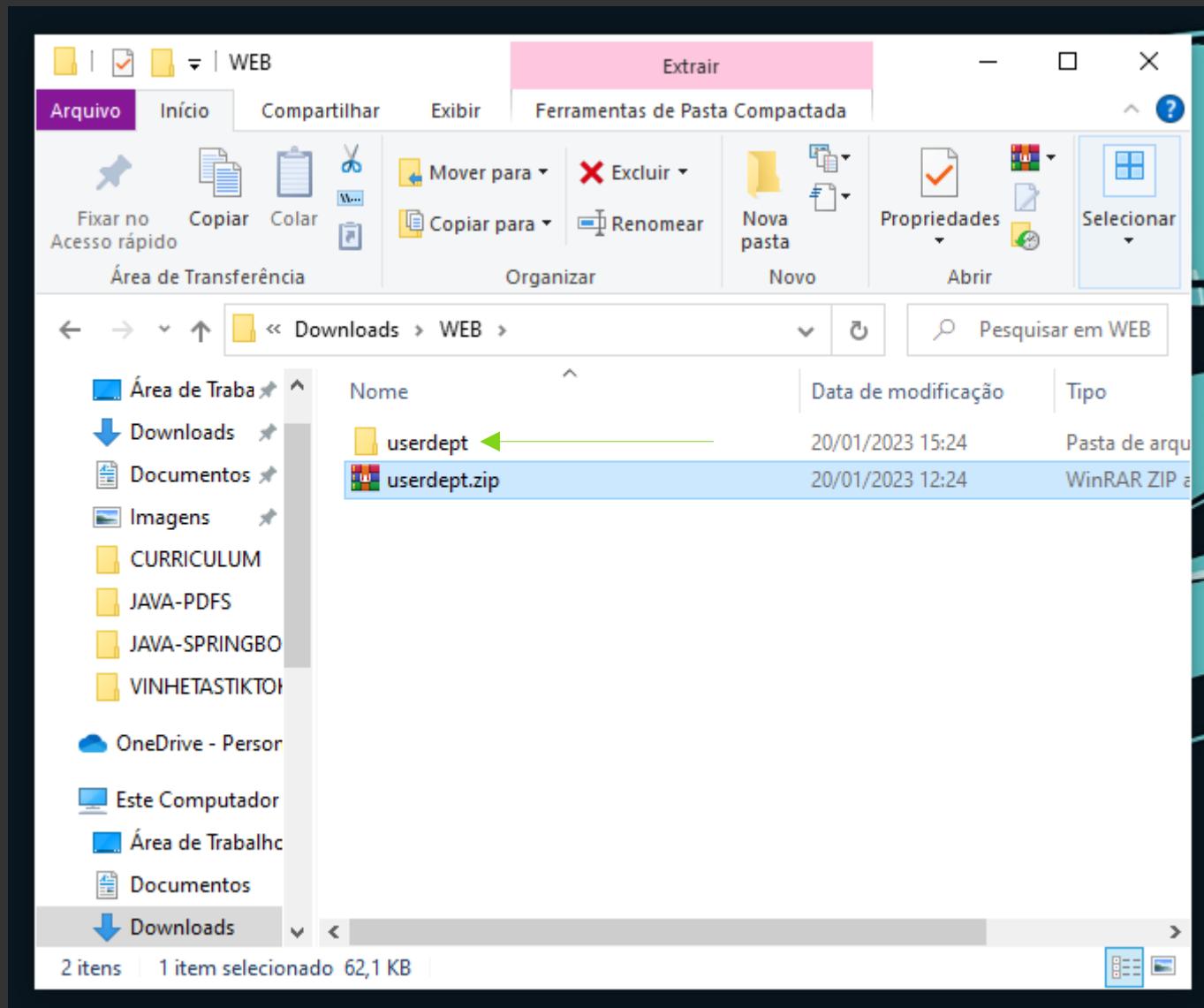
GENERATE CTRL + ↵ EXPLORE CTRL + SPACE SHARE...

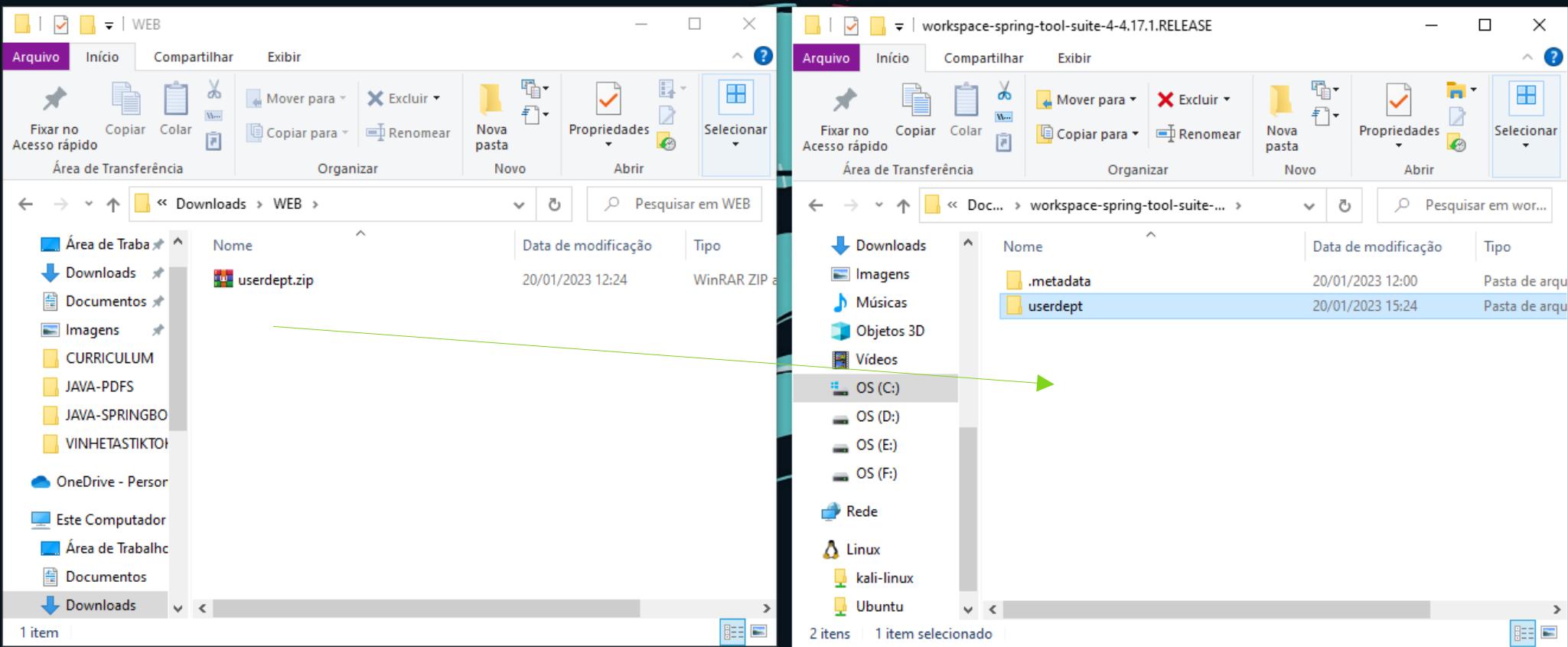
GitHub Twitter

28°C Nublado 12:23

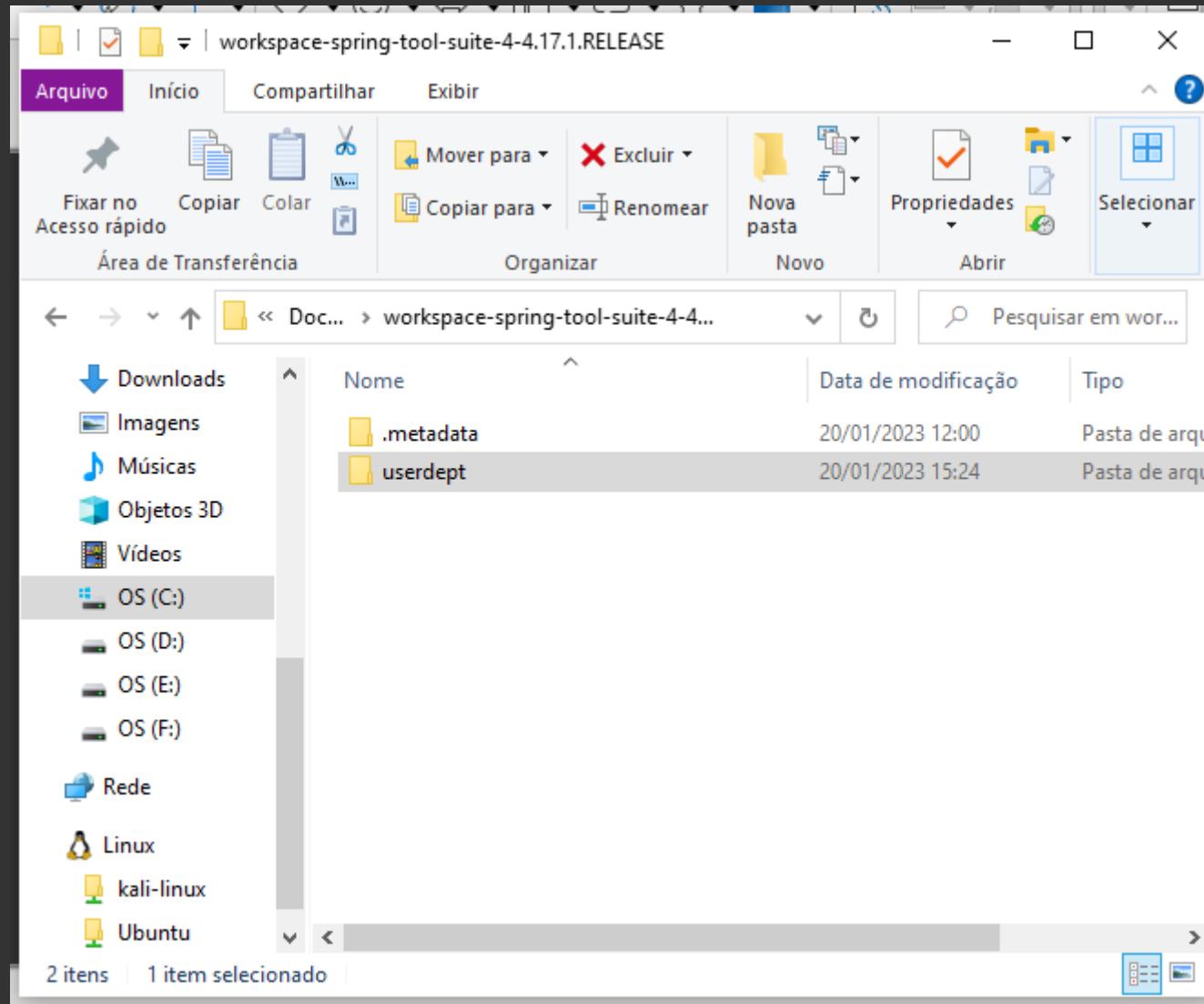


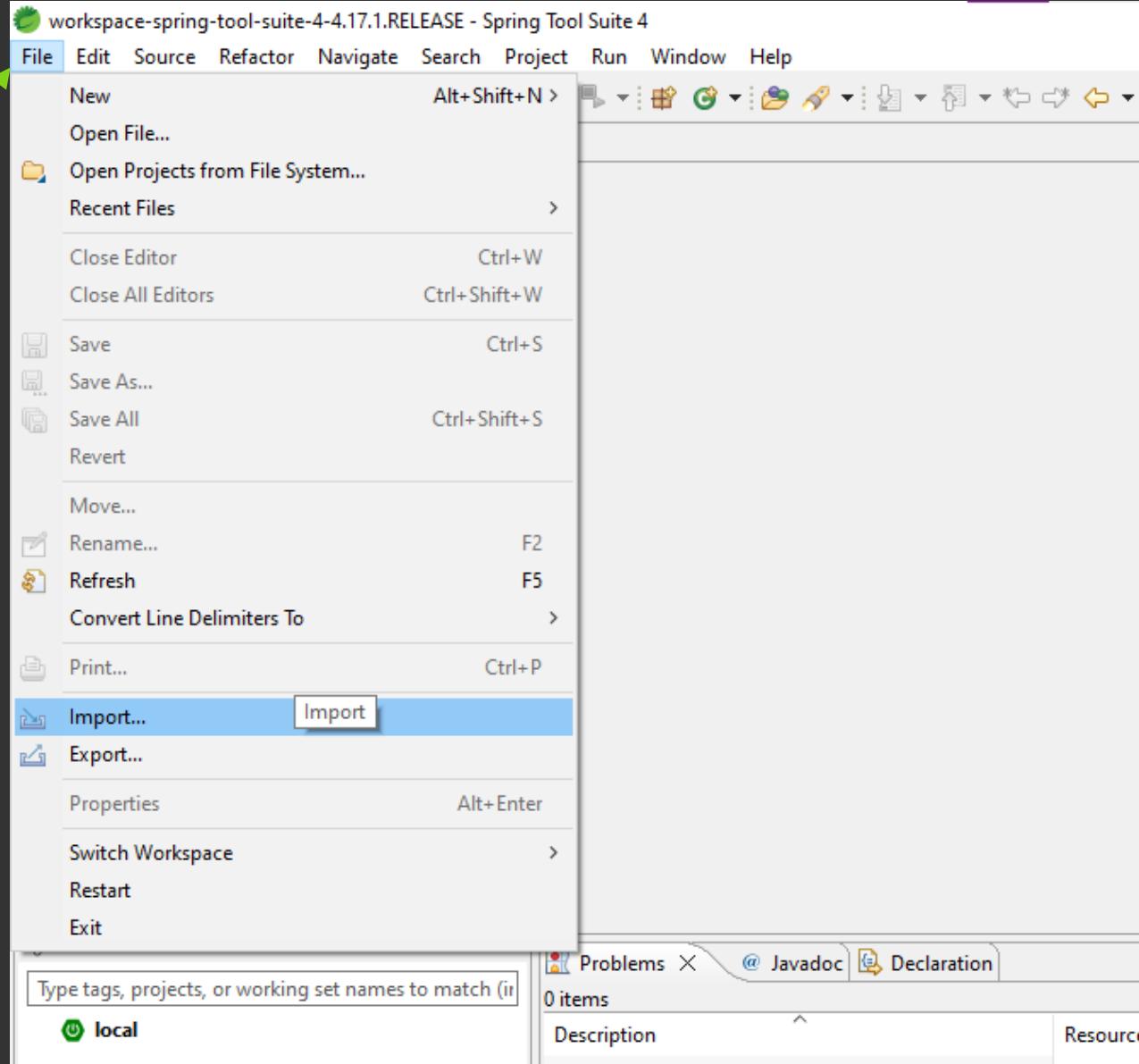
Crie uma pasta baixe e extraia o arquivo

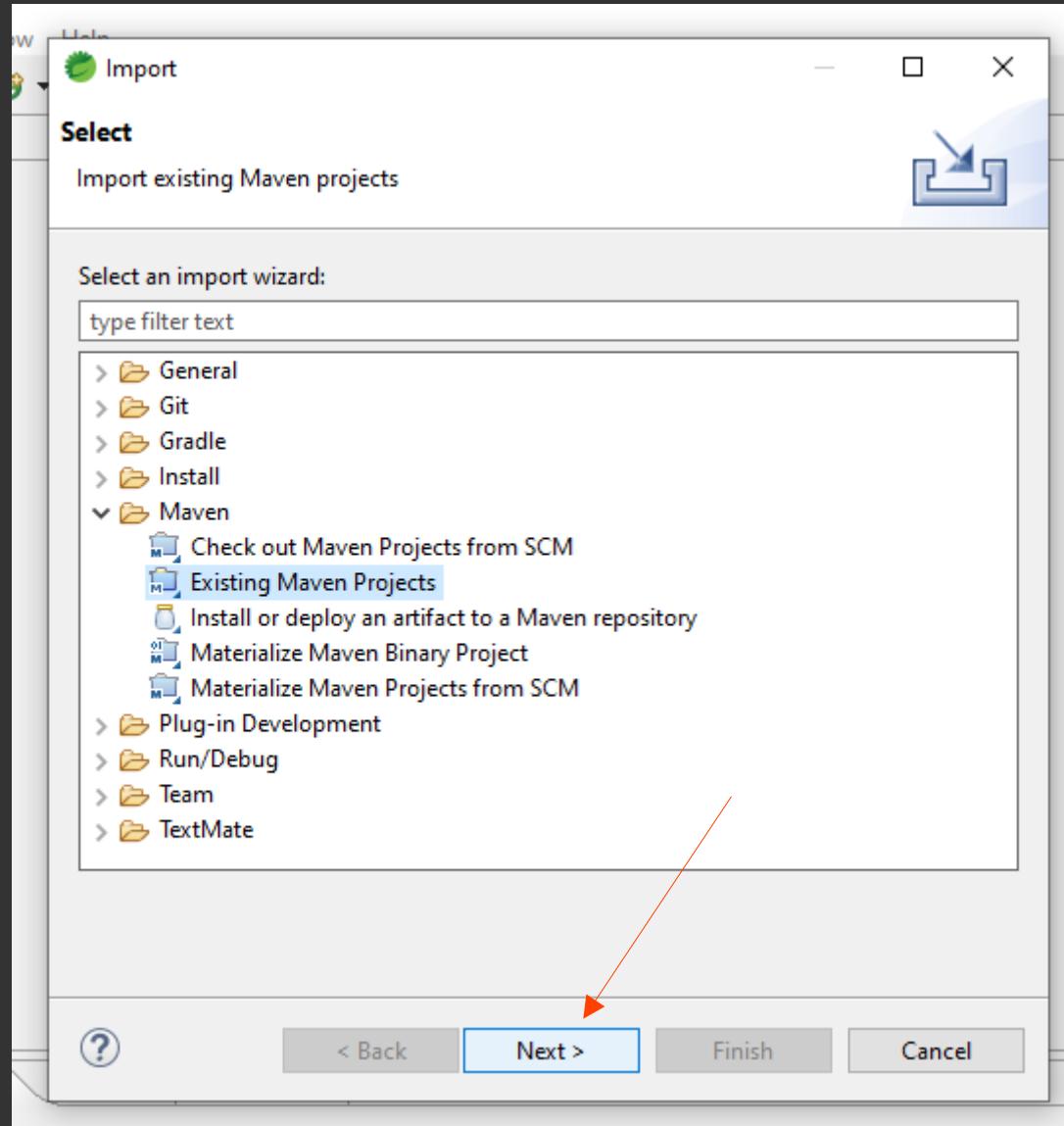


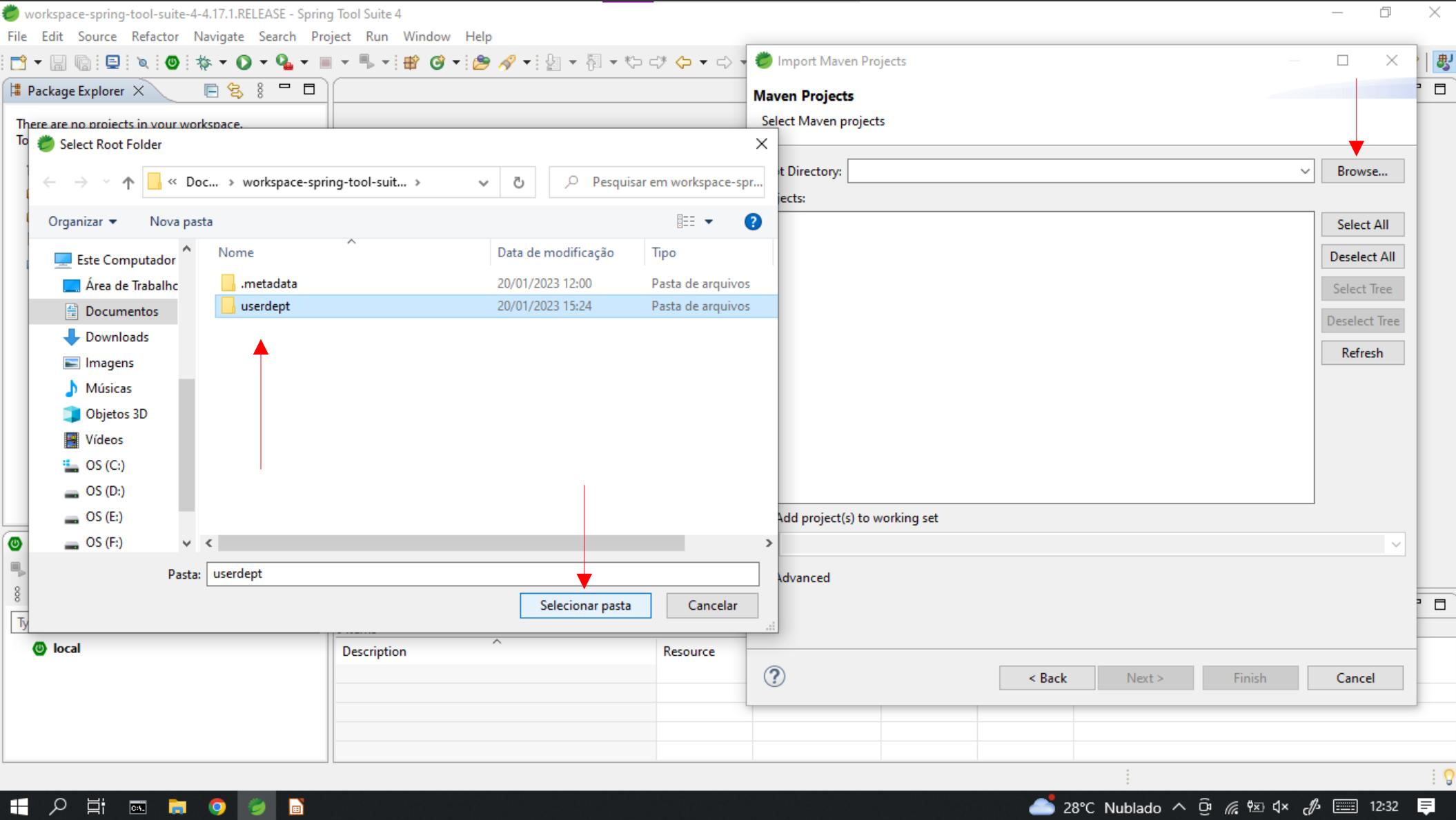


Jogue a pasta do projeto extraída para dentro do diretório do devtools
C:\Users\brito\Documents\workspace-spring-tool-suite-4-4.17.1.RELEASE











Import Maven Projects



Maven Projects

Select Maven projects

Root Directory: C:\Users\brito\Documents\workspace-spring-tool-suite-4-4.17.1.RELEASE\use

Projects:

/pom.xml com.devsuperior:userdept:0.0.1-SNAPSHOT:jar



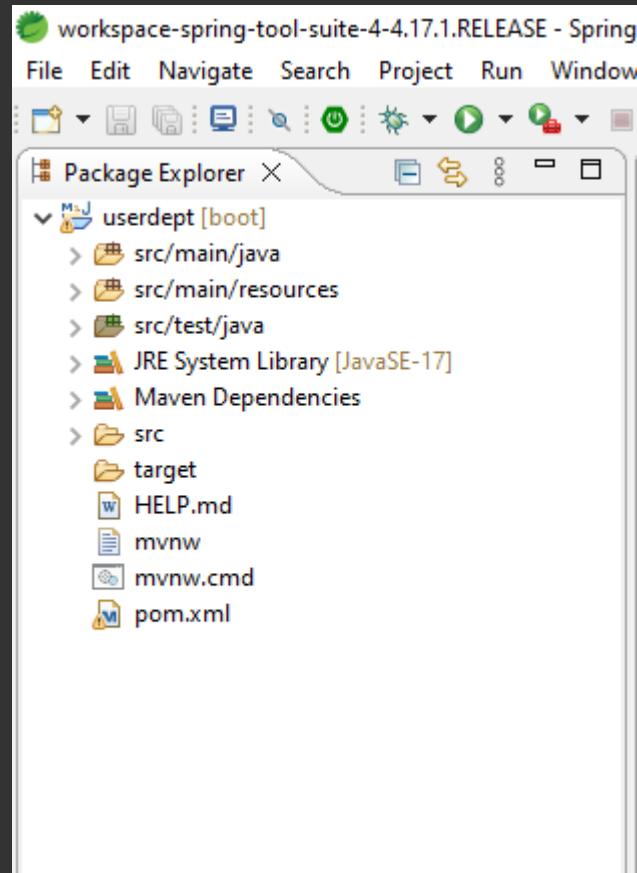
Add project(s) to working set



< Back

Next >

ESPERE ATÉ O PROJETO TODO SER CONFIGURADO



workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/UserdeptApplication.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help



Package Explorer X

```
userdept [boot]
  src/main/java
    com.devsuperior.userdept
      UserdeptApplication.java
  src/main/resources
  src/test/java
  JRE System Library [JavaSE-17]
  Maven Dependencies
  src
    target
    HELP.md
    mvnw
    mvnw.cmd
  pom.xml
```

UserdeptApplication.java X

```
1 package com.devsuperior.userdept;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 public class UserdeptApplication {
7
8     public static void main(String[] args) {
9         SpringApplication.run(UserdeptApplication.class, args);
10    }
11
12 }
13
14 }
```

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/UserdeptApplication.java

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

UserdeptApplication.java X

```
1 package com.devsuperior.userdept;
2
3+ import org.springframework.boot.S
4
5 @SpringBootApplication
6 public class UserdeptApplication {
7
8     public static void main(String
9         SpringApplication.run(User
10    }
11 }
```

(Re)start Ctrl+Alt+Shift+B, R
Start or restart the process associated with the selected elements

(Re)debug Ctrl+Alt+Shift+B, D
Stop Ctrl+Alt+Shift+B, S

Open Web Browser

Open Ngrok Admin UI

Open Console

Open In Package Explorer

Show Properties

Live Data Connections... >

Open Config Ctrl+Alt+Shift+B, F

Duplicate Config

Delete Config

(Re)start and Expose via ngrok

(Re)debug and Expose via ngrok

Add a Cloud Foundry Target

Add a Docker Target

Boot Dashboard

Type tags, projects

local

userdept

Javadoc Declaration

Warnings (1 item)

Newer patch version of Spring Boot available

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/UserdeptApplication.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

userdept [boot]

- src/main/java
 - com.devsuperior.userdept
 - UserdeptApplication.java
- src/main/resources
- src/test/java
- JRE System Library [JavaSE-17]
- Maven Dependencies
- src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml

UserdeptApplication.java X

```
1 package com.devsuperior.userdept;
2
3 import org.springframework.boot.SpringApplication;
```

Problems Declaration Console X

userdept - UserdeptApplication [Spring Boot App] C:\Users\brito\Downloads\spring\sts-4.17.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\java

CLICK DUPLO

Outline X

com.devsuperior.userdept

UserdeptApplication

S main/Configurable

Boot Dashboard X

Type tags, projects, or working set names to match (ir)

local

userdept [:8080]

28°C Nublado 12:40

2023-01-20T12:40:04.814-03:00 INFO 10032 --- [main] c.d.userdept.UserdeptApplication : Starting UserdeptApplication using Java

2023-01-20T12:40:04.821-03:00 INFO 10032 --- [main] : No active profile set, falling back to 1

2023-01-20T12:40:06.299-03:00 INFO 10032 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repository

2023-01-20T12:40:06.345-03:00 INFO 10032 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning

2023-01-20T12:40:07.174-03:00 INFO 10032 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (h

2023-01-20T12:40:07.196-03:00 INFO 10032 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]

2023-01-20T12:40:07.196-03:00 INFO 10032 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/

2023-01-20T12:40:07.455-03:00 INFO 10032 --- [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicat

2023-01-20T12:40:07.456-03:00 INFO 10032 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initializati

2023-01-20T12:40:07.673-03:00 INFO 10032 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...

2023-01-20T12:40:07.965-03:00 INFO 10032 --- [main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection conn: u

2023-01-20T12:40:07.968-03:00 INFO 10032 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.

2023-01-20T12:40:08.052-03:00 INFO 10032 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInf

2023-01-20T12:40:08.138-03:00 INFO 10032 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.

2023-01-20T12:40:08.391-03:00 WARN 10032 --- [main] org.hibernate.orm.deprecation : HHH90000021: Encountered deprecated sett

2023-01-20T12:40:08.661-03:00 INFO 10032 --- [main] SQL dialect : HHH000400: Using dialect: org.hibernate.

2023-01-20T12:40:09.049-03:00 INFO 10032 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementat

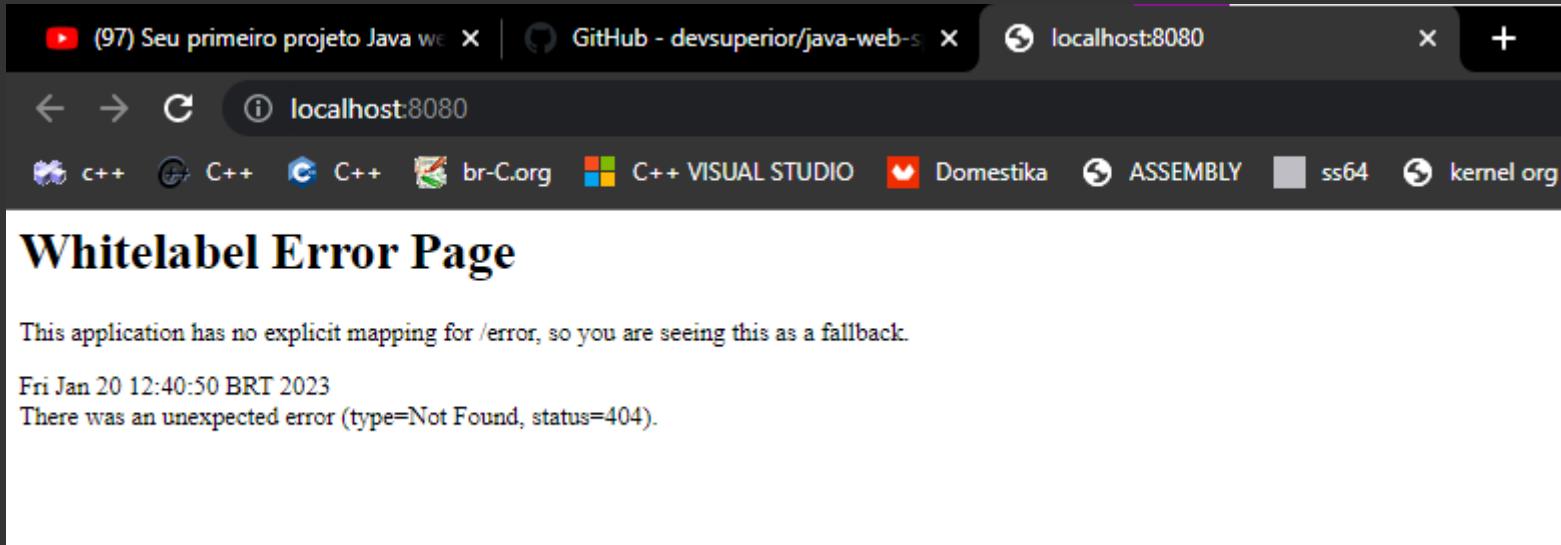
2023-01-20T12:40:09.069-03:00 INFO 10032 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for

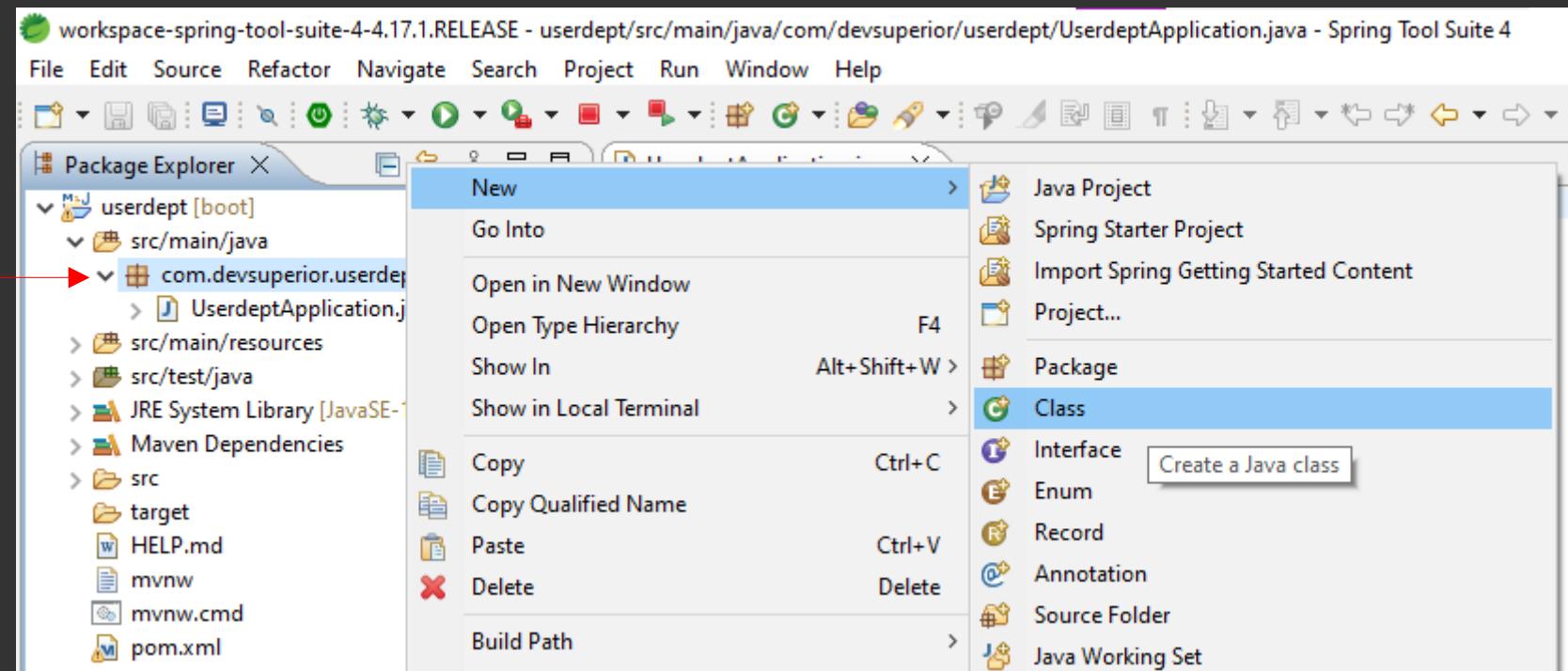
2023-01-20T12:40:09.136-03:00 WARN 10032 --- [main] JpaBaseConfiguration\$JpaWebConfiguration : spring.jpa.open-in-view is enabled by de

2023-01-20T12:40:09.680-03:00 INFO 10032 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) w

2023-01-20T12:40:09.699-03:00 INFO 10032 --- [main] c.d.userdept.UserdeptApplication : Started UserdeptApplication in 5.599 sec

A APLICAÇÃO JÁ RODA





Seu primeiro projeto Java web no Spring Boot 2022

☰ README.md



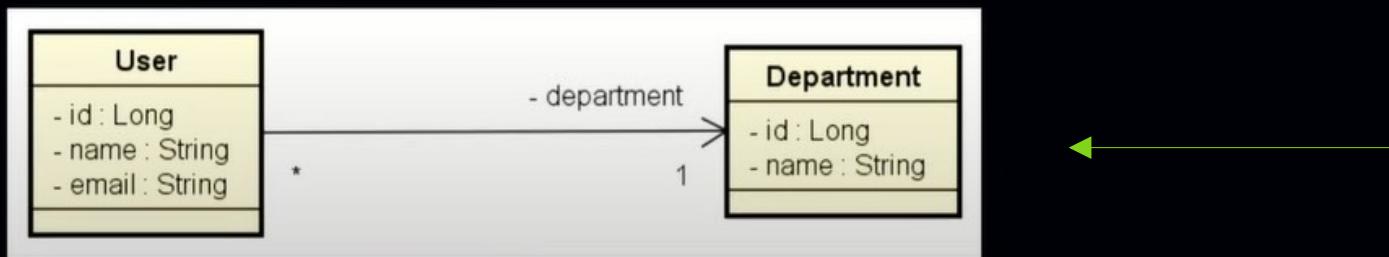
- Resgatar fundamentos de programação
- Colocar em prática esses fundamentos
- Criar um pequeno sistema com ferramentas e práticas de mercado
- Dar mais um passo em direção à preparação para o mercado

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

DE ONDE VEM O MODELO
QUE ESTAMOS CRIANDO



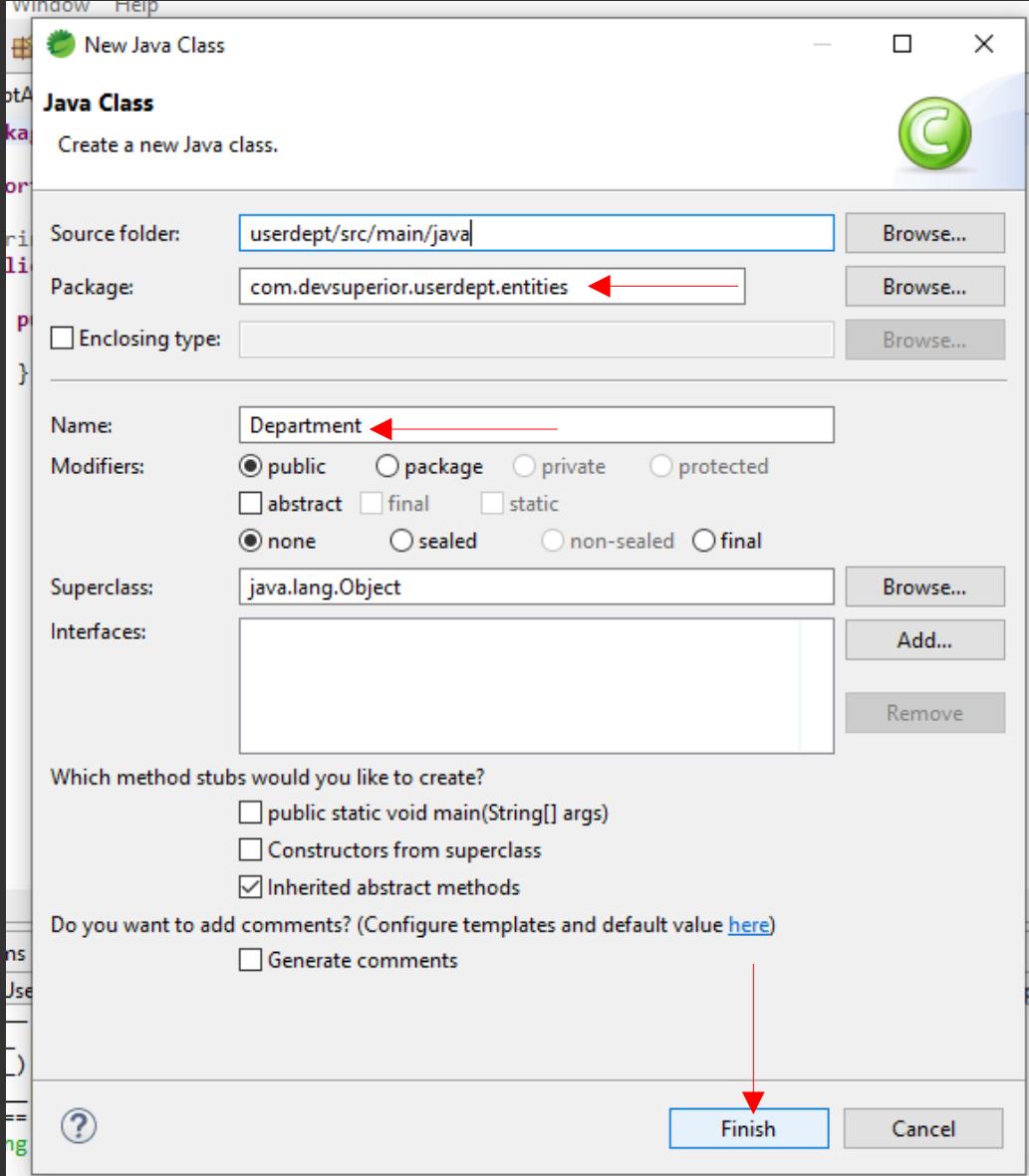
Desenvolvimento moderno: relacional -> objeto -> json

tb_department

`id` `name`

`User`

Acrescente .entities



workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/Department.java

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X UserdeptApplication.java Department.java X

src/main/java com.devsuperior.userdept UserDeptApplication.java com.devsuperior.userdept.entities Department.java src/main/resources src/test/java JRE System Library [JavaSE-17] Maven Dependencies src target HELP.md mvnw mvnw.cmd pom.xml

```
1 package com.devsuperior.userdept.entities;
2
3 public class Department {
4
5 }
6
```



```

1 package com.devsuperior.userdept.entities;
2
3 public class Department {
4
5     private Long id;
6     private String name;
7
8     public Department() {
9
10    }
11
12 }
13

```

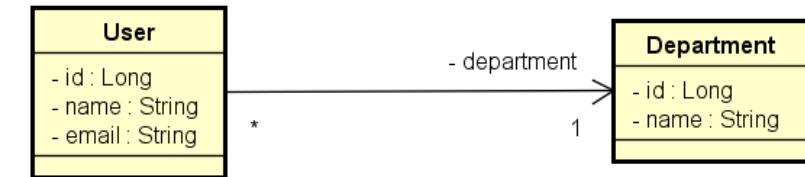
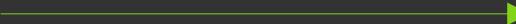
Digite o code acima

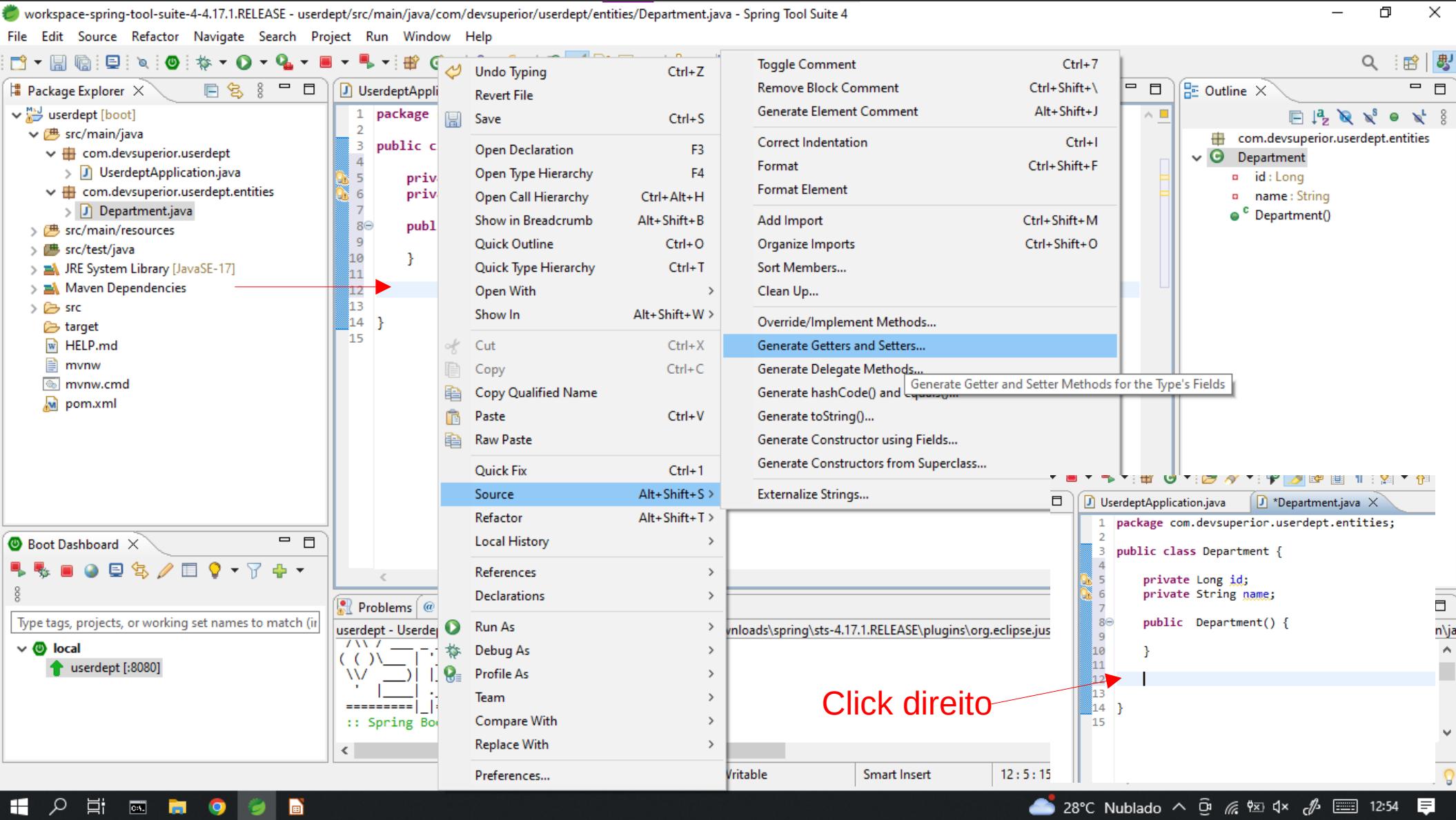
Visão geral do sistema

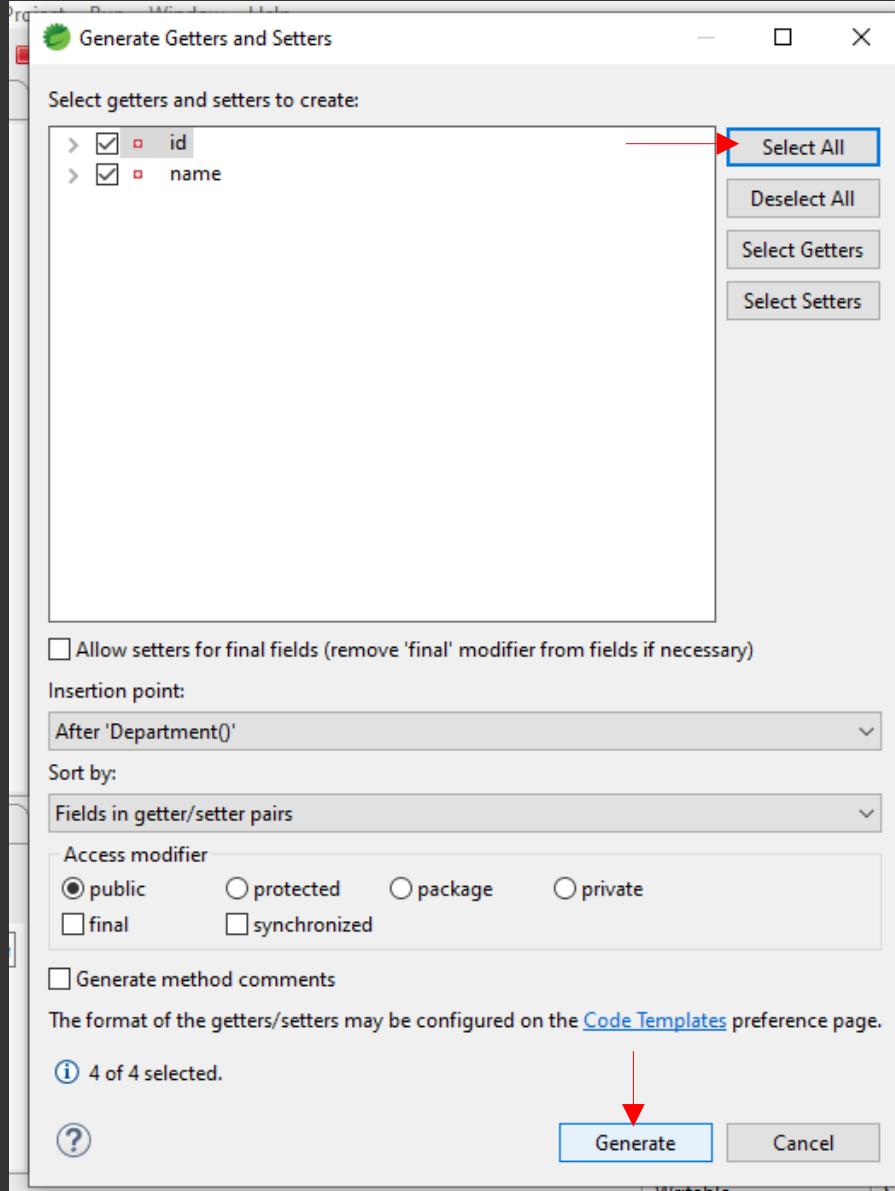
Vamos construir um pequeno sistema (API REST) de usuários e departamentos, com os

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

De onde vem os dados







workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/Department.java

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer UserdeptApplication.java *Department.java

```
2 public class Department {  
3     private Long id;  
4     private String name;  
5  
6     public Department() {  
7         }  
8  
9     public Long getId() {  
10         return id;  
11     }  
12  
13     public void setId(Long id) {  
14         this.id = id;  
15     }  
16  
17     public String getName() {  
18         return name;  
19     }  
20  
21     public void setName(String name) {  
22         this.name = name;  
23     }  
24  
25 }  
26  
27  
28  
29  
30 }
```

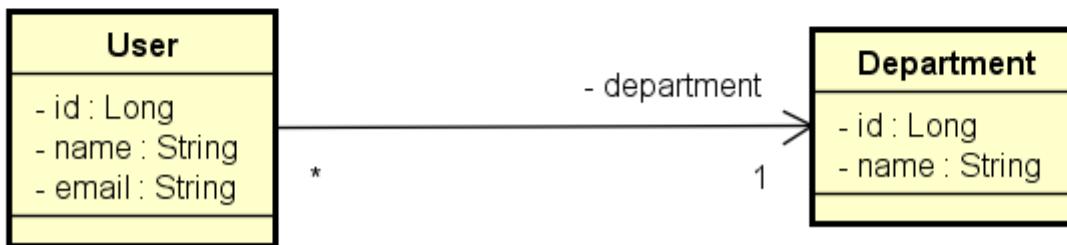
Boot Dashboard

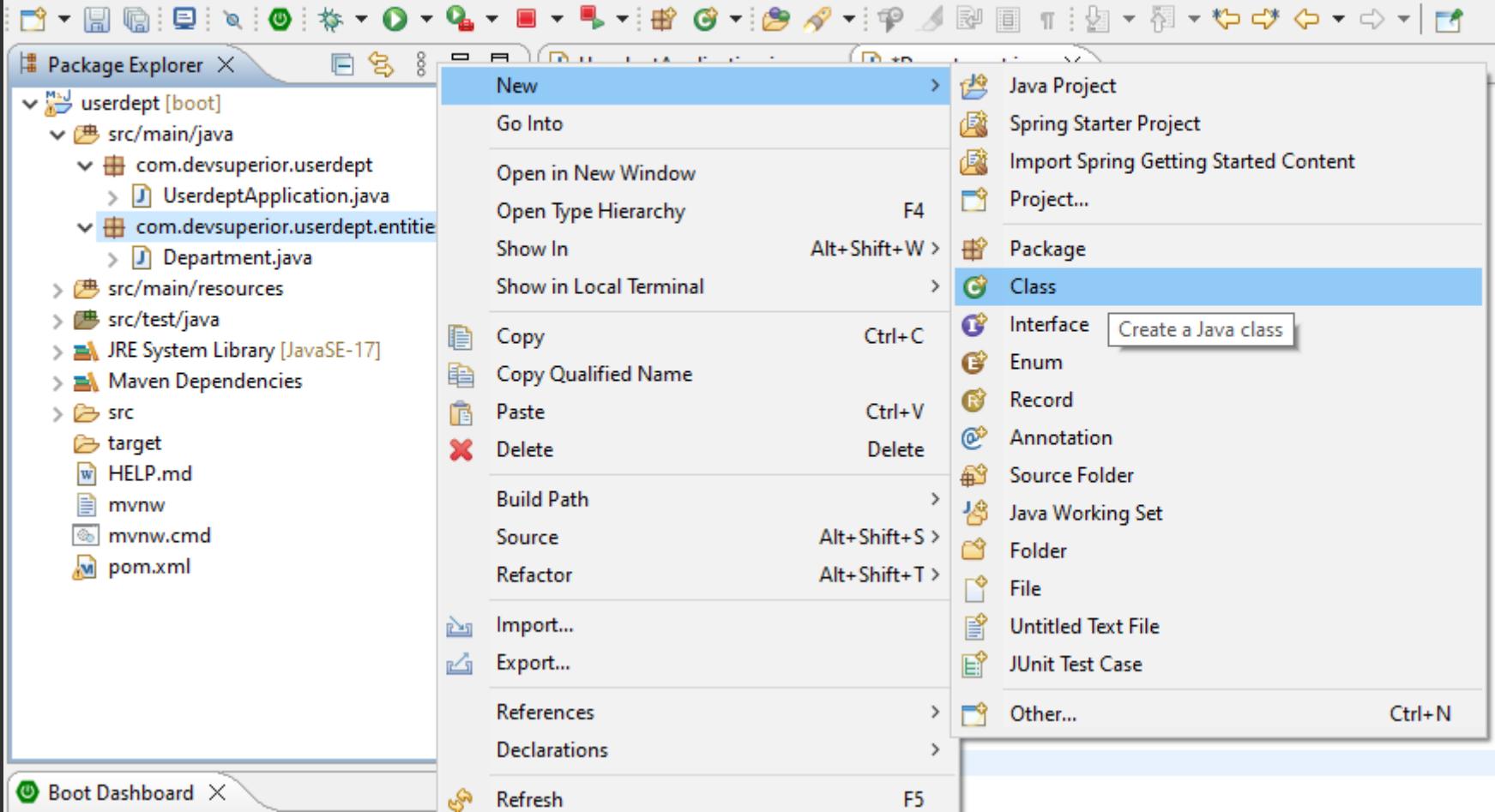
README.md

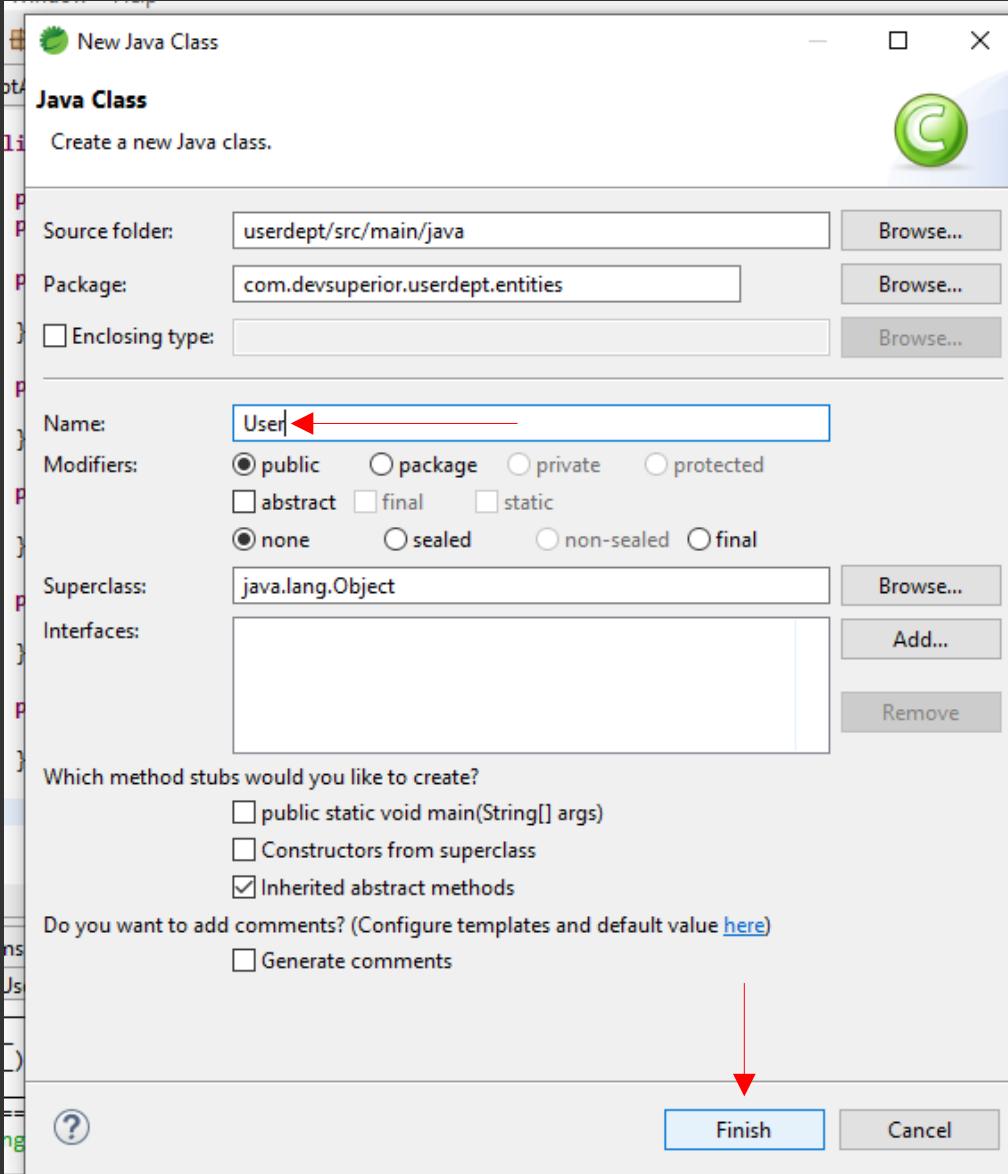
Visão geral do sistema

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

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







workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/User.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help



Package Explorer X

userdept [boot]
 src/main/java
 com.devsuperior.userdept
 UserDeptApplication.java
 com.devsuperior.userdept.entities
 Department.java
 User.java
 src/main/resources
 src/test/java
 JRE System Library [JavaSE-17]
 Maven Dependencies
 src
 target
 HELP.md
 mvnw
 mvnw.cmd
 pom.xml

UserDeptApplication.java

```
1 package com.devsuperior.userdept.entities;  
2  
3 public class User {  
4  
5 }  
6
```

*Department.java

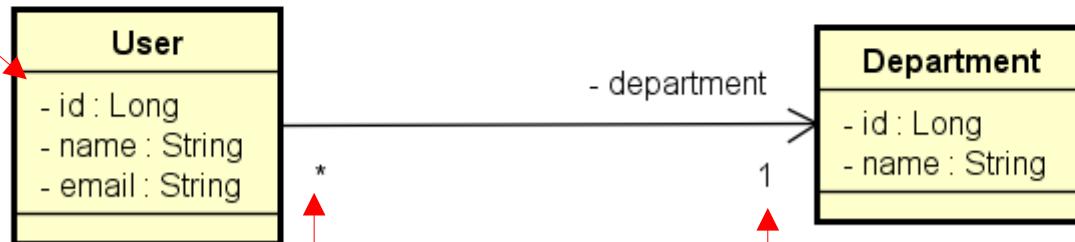
User.java X

De onde vem os dados pra fazer a classe e seus métodos

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



Muitos pra um relacionamento UML

The screenshot shows the Spring Tool Suite interface. The left panel is the Package Explorer, displaying the project structure for 'userdept'. The right panel is the code editor for 'User.java'.

```
1 package com.devsuperior.userdept.entities;
2
3 public class User {
4
5     private Long id;
6     private String name;
7     private String email;
8
9     private Department department;
10
11 }
```

A red arrow points upwards from the bottom of the code editor towards the 'User.java' tab in the top bar.

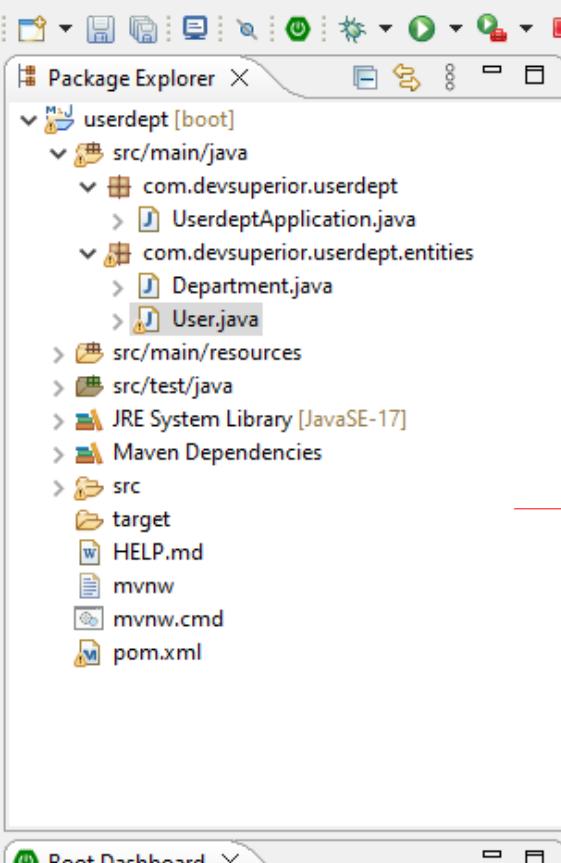
Esse ultimo campo diz respeito ao relacionamento muitos pra um

The screenshot shows the Spring Tool Suite interface. The title bar reads "workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/User.java - Spring Tool Suite". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar has various icons for file operations like Open, Save, Copy, Paste, and Build.

The left side features the Package Explorer view, which displays the project structure:

- userdept [boot]
 - src/main/java
 - com.devsuperior.userdept
 - UserDeptApplication.java
 - com.devsuperior.userdept.entities
 - Department.java
 - User.java
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-17]
 - Maven Dependencies
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml

Criando o metodo

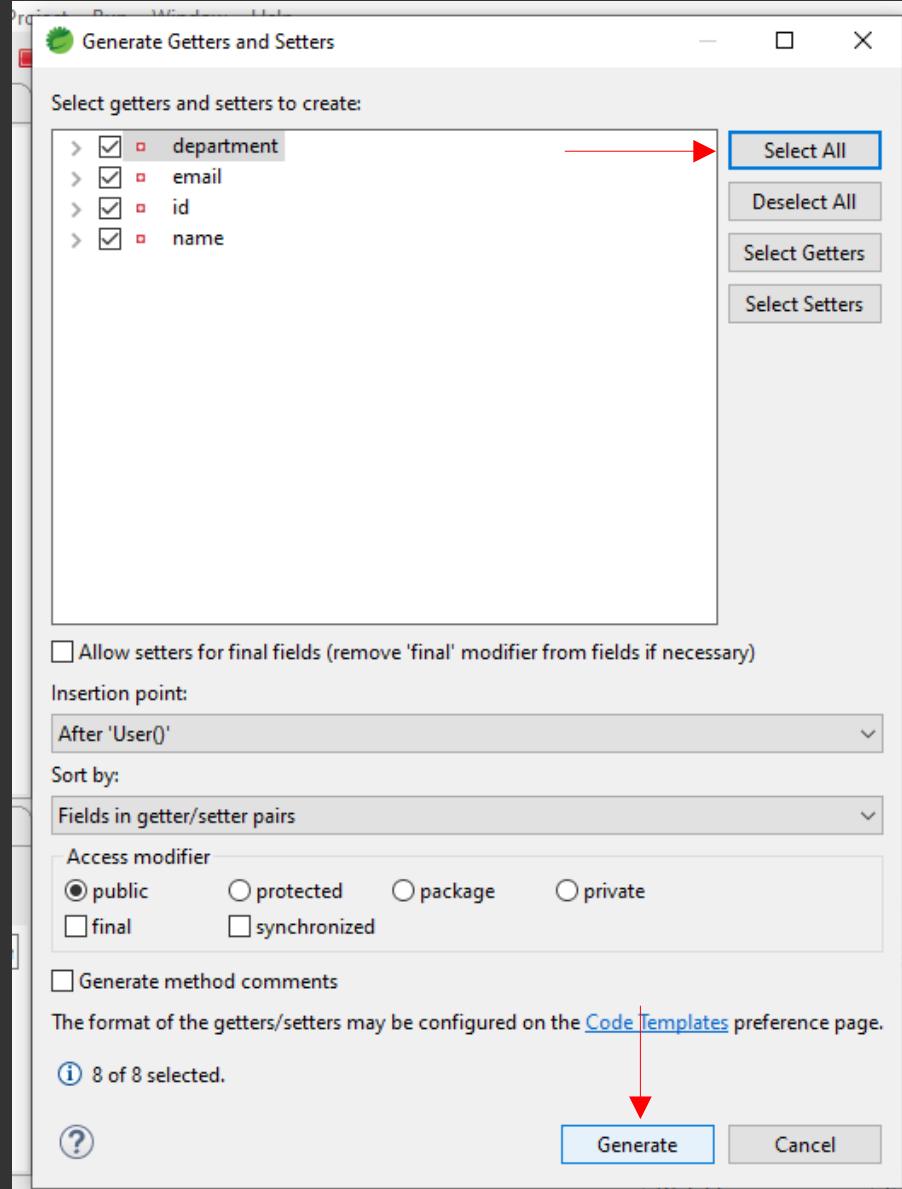


The code editor shows the `User.java` file with the following code:

```
1 package com.devsuperior.userdept.entities;
2
3 public class User {
4     private Long id;
5     private String name;
6     private String email;
7     private Department department;
8
9     public User() {
10 }
11
12 }
13
14 }
```

A context menu is open over the closing brace of the `User` class definition (line 14). A red arrow points from the text "Click direito" at the bottom to the context menu. The menu items include: Undo Typing (Ctrl+Z), Revert File, Save (Ctrl+S), Open Declaration (F3), Open Type Hierarchy (F4), Open Call Hierarchy (Ctrl+Alt+H), Show in Breadcrumb (Alt+Shift+B), Quick Outline (Ctrl+O), Quick Type Hierarchy (Ctrl+T), Open With (with a submenu), Show In (Alt+Shift+W), Cut (Ctrl+X), Copy (Ctrl+C), Copy Qualified Name, Paste (Ctrl+V), Raw Paste, Quick Fix (Ctrl+1), Source (Alt+Shift+S), Refactor (Alt+Shift+T), Local History (with a submenu), Toggle Comment, Remove Block Comment, Generate Element Comment, Correct Indentation (Ctrl+I), Format (Ctrl+Shift+F), Format Element, Add Import (Ctrl+Shift+M), Organize Imports (Ctrl+Shift+O), Sort Members..., Clean Up..., Override/Implement Methods..., Generate Getters and Setters..., Generate Delegate Methods (highlighted in blue), Generate hashCode() and equals()..., Generate toString()..., Generate Constructor using Fields..., Generate Constructors from Superclass..., and Externalize Strings... .

Click direito





Package Explorer X

```
userdept [boot]
  src/main/java
    com.devsuperior.userdept
      UserDeptApplication.java
    com.devsuperior.userdept.entities
      Department.java
      User.java
  src/main/resources
  src/test/java
  JRE System Library [JavaSE-17]
  Maven Dependencies
  src
    target
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
```

UserDeptApplication.java

```
22
23  public String getName() {
24      return name;
25  }
26
27  public void setName(String name) {
28      this.name = name;
29  }
30
31  public String getEmail() {
32      return email;
33  }
34
35  public void setEmail(String email) {
36      this.email = email;
37  }
38
39  public Department getDepartment() {
40      return department;
41  }
42
43  public void setDepartment(Department department) {
44      this.department = department;
45  }
46
47
48
49 }
```

<

Boot Dashboard X



De onde vem os dados

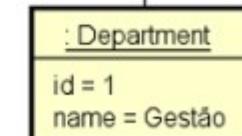
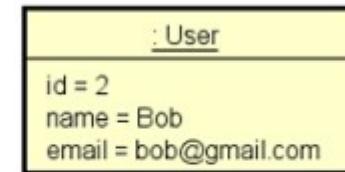
Desenvolvimento moderno: relacional -> objeto -> json

→ 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



The screenshot shows the Spring Tool Suite interface with the following details:

- Title Bar:** workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/Department.java - Spring Tool
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Includes icons for file operations like Open, Save, Find, and Run.
- Package Explorer:** Shows the project structure:
 - userdept [boot]
 - src/main/java
 - com.devsuperior.userdept
 - UserdeptApplication.java
 - com.devsuperior.userdept.entities
 - Department.java (highlighted with a red arrow)
 - User.java
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-17]
 - Maven Dependencies
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
- Code Editor:** Displays the `Department.java` file content:

```
1 package com.devsuperior.userdept.entities;
2
3 @Entity
4 @Table(name = "tb_department")
5 public class Department {
6
7     private Long id;
8     private String name;
9
10    public Department() {
11
12    }
13
14    public Long getId() {
15        return id;
16    }
17
18    public void setId(Long id) {
19        this.id = id;
20    }
21}
```

A red arrow points from the red arrow in the Package Explorer to the `@Table` annotation in the code editor.

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/Department.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X UserdeptApplication.java *Department.java X User.java

```
1 package com.devsuperior.userdept.entities;
2
3 @Entity
4 @Table(name = "Department")
5 public class Department {
6     @Id
7     @GeneratedValue(strategy = GenerationType.IDENTITY)
8     private Long id;
9
10    @Column(name = "name")
11    private String name;
12
13    @Column(name = "description")
14    private String description;
15
16    @Column(name = "value")
17    private Double value;
18
19    @Column(name = "order")
20    private Integer order;
21 }
```

Import 'Entity' (jakarta.persistence) ← Create annotation 'Entity'
Change to 'EntityScan' (org.springframework.boot.autoconfigure.domain.EntityScan)
Rename in file (Ctrl+2, R)
Fix project setup...

... package com.devsuperior.userdept.entities ...
import jakarta.persistence.Entity; ...
@Entity
@Table(name = "Department") ...

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/Department.java - Spring Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X UserdeptApplication.java *Department.java X User.java

```
1 package com.devsuperior.userdept.entities;
2
3 import jakarta.persistence.Entity;
4
5 @Entity
6 @Table(name = "tb_department")
7 pu ← Import 'Table' (jakarta.persistence) ←
8 ← Import 'Table' (org.hibernate.annotations)
9 ← Create annotation 'Table'
10 ← Change to 'Immutable' (org.hibernate.annotations)
11 ← Change to 'Immutable' (org.springframework.data.annotation)
12 ← Change to 'Tables' (org.hibernate.annotations)
13 ← Rename in file (Ctrl+2, R)
14 ← Fix project setup...
15
16
17
18
19
20
21
```

userdept [boot]
src/main/java
com.devsuperior.userdept
UserDeptApplication.java
com.devsuperior.userdept.entities
Department.java
User.java
src/main/resources
src/test/java
JRE System Library [JavaSE-17]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml

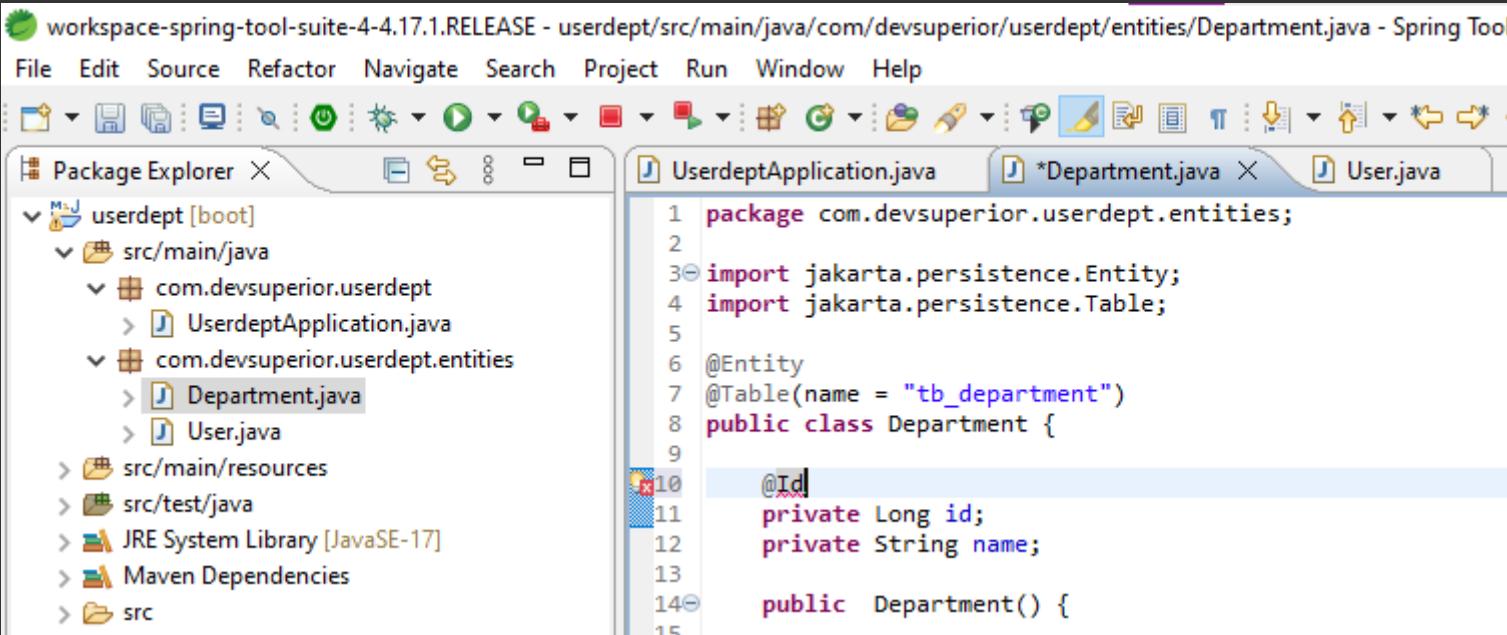
The screenshot shows the Spring Tool Suite (STS) interface. The title bar reads "workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/Department.java - Spring Tool". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar has various icons for file operations like Open, Save, Copy, Paste, and Find. The left sidebar is the "Package Explorer" showing the project structure:

- userdept [boot]
- src/main/java
 - com.devsuperior.userdept
 - UserDeptApplication.java
 - com.devsuperior.userdept.entities
 - Department.java
 - User.java
 - src/main/resources
 - src/test/java
- JRE System Library [JavaSE-17]

The right side is the code editor for "Department.java" under the package "com.devsuperior.userdept.entities". The code is as follows:

```
1 package com.devsuperior.userdept.entities;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.Table;
5
6 @Entity
7 @Table(name = "tb_department")
8 public class Department {
9
10     private Long id;
11     private String name;
12
13 }
```

Continuando com as anotações



The screenshot shows the Spring Tool Suite interface with the following details:

- Title Bar:** workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/Department.java - Spring Tool Suite
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Includes icons for file operations like Open, Save, Find, and Run.
- Package Explorer:** Shows the project structure:
 - userdept [boot]
 - src/main/java
 - com.devsuperior.userdept
 - UserDeptApplication.java
 - entities
 - Department.java
 - User.java
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-17]
 - Maven Dependencies
 - src
 - Code Editor:** Displays the `Department.java` file content:

```
1 package com.devsuperior.userdept.entities;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.Table;
5
6 @Entity
7 @Table(name = "tb_department")
8 public class Department {
9
10    @Id
11    private Long id;
12    private String name;
13
14    public Department() {
15
```
 - Tab Bar:** Shows tabs for UserDeptApplication.java, *Department.java, and User.java.

The screenshot shows the Spring Tool Suite 4 interface with the following details:

- Project Explorer:** Shows the project structure under "userdept [boot]".
 - src/main/java:** Contains:
 - com.devsuperior.userdept:** UserdeptApplication.java
 - com.devsuperior.userdept.entities:** Department.java, User.java
 - src/main/resources:**
 - src/test/java:**
 - JRE System Library [JavaSE-17]:**
 - Maven Dependencies:**
 - src:**
 - target:**
 - HELP.md:**
 - mvnw:**
- Editor:** Displays the file `UserdeptApplication.java`. The cursor is at line 10, column 1, where the code `@Id` is being typed.
- Code Completion:** A tooltip shows the following options:
 - `pr` ← Import 'Id' (jakarta.persistence)
 - `pr` ← Import 'Id' (org.springframework.data.annotation)
 - `pu` @ Create annotation 'Id'
 - `}` Rename in file (Ctrl+2, R)
 - `nu` Fix project setup...
- Right-hand pane:** Shows the imports for the current file:

```
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
```

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/Department.java - Spring Tool

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X UserdeptApplication.java *Department.java X User.java

```
1 package com.devsuperior.userdept.entities;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.Id;
5 import jakarta.persistence.Table;
6
7 @Entity
8 @Table(name = "tb_department")
9 public class Department {
10
11     @Id
12     private Long id;
13     private String name;
```

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/Department.java - Spring Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X UserdeptApplication.java *Department.java X User.java

```
1 package com.devsuperior.userdept.entities;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.Id;
5 import jakarta.persistence.Table;
6
7 @Entity
8 @Table(name = "tb_department")
9 public class Department {
10
11     @Id
12     @GeneratedValue
13     private Long id;
14     private String name;
15
16     public Department() {
17
18     }
19
20     public Long getId() {
21         return id;
22     }
23 }
```

The screenshot shows the Spring Tool Suite 4 interface with the following details:

- Project Explorer:** Shows the project structure under "userdept [boot]". The "src/main/java" folder contains packages "com.devsuperior.userdept" and "com.devsuperior.userdept.entities". "com.devsuperior.userdept.entities" contains files "Department.java" and "User.java".
- Code Editor:** Displays the file "Department.java". The cursor is at line 12, column 12, where the annotation "@GeneratedValue" is being typed.
- Code Completion Pop-up:** A tooltip provides options for the annotation:
 - Import 'GeneratedValue' (jakarta.persistence)
 - Create annotation 'GeneratedValue'
 - Change to 'Generated' (jakarta.annotation)
 - Change to 'Generated' (javax.annotation.processing)
 - Change to 'Generated' (org.hibernate.annotations)
 - Change to 'GeneratedColumns' (org.hibernate.annotations.DialectContext)
- Right-hand pane:** Shows the imports for the current file:

```
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
```

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/Department.java - Spring Tool

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X UserdeptApplication.java *Department.java X User.java

```
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.Id;
6 import jakarta.persistence.Table;
7
8 @Entity
9 @Table(name = "tb_department")
10 public class Department {
11
12     @Id
13     @GeneratedValue
14     private Long id;
15     private String name;
16
17     public Department() {
18
19     }
20
21     public Long getId() {
22         return id;
23     }
24 }
```

Complete a linha com os imports

The screenshot shows the Spring Tool Suite 4 interface with the following details:

- Title Bar:** workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/Department.java - Spring Tool Suite 4
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Toolbar:** Includes icons for file operations like Open, Save, Find, and Run.
- Package Explorer:** Shows the project structure:
 - userdept [boot]
 - src/main/java
 - com.devsuperior.userdept
 - UserdeptApplication.java
 - com.devsuperior.userdept.entities
 - Department.java
 - User.java
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-17]
 - Maven Dependencies
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
- Editor Area:** Displays the `Department.java` file content with code completion suggestions.

```
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.GenerationType; ← Completion suggestion
6 import jakarta.persistence.Id;
7 import jakarta.persistence.Table;
8
9 @Entity
10 @Table(name = "tb_department")
11 public class Department {
12
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY) ← Completion suggestion
15     private Long id;
16     private String name;
17
18     public Department() {
19
20     }
21
22     public Long getId() {
```

copy

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/Department.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

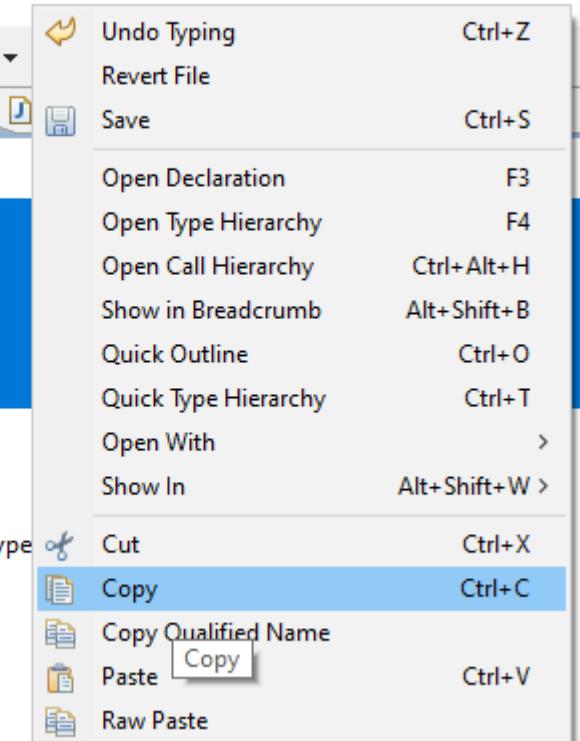
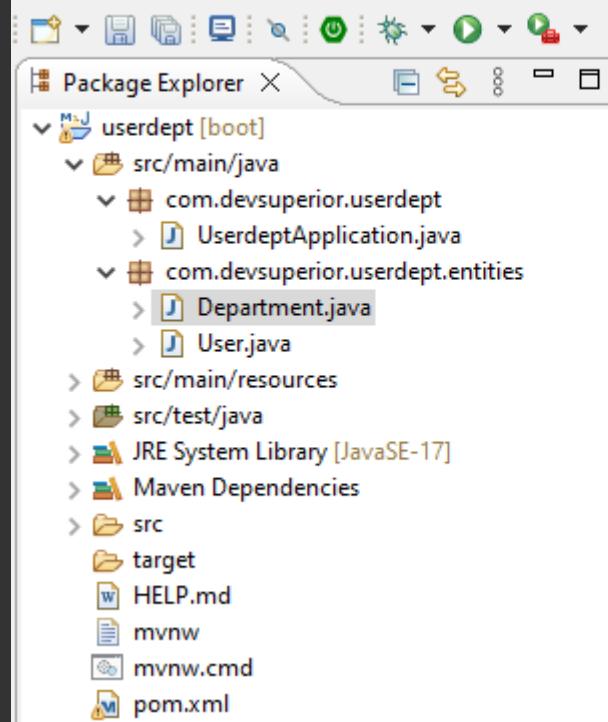
Package Explorer X

userdept [boot]

- src/main/java
 - com.devsuperior.userdept
 - UserdeptApplication.java
 - Department.java
 - User.java
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-17]
 - Maven Dependencies
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml

UserdeptApplication.java *Department.java X User.java

```
1 package com.devsuperior.userdept.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 public class Department {
12
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private Long id;
16     private String name;
17
18     public Department() {
19
20     }
21 }
```

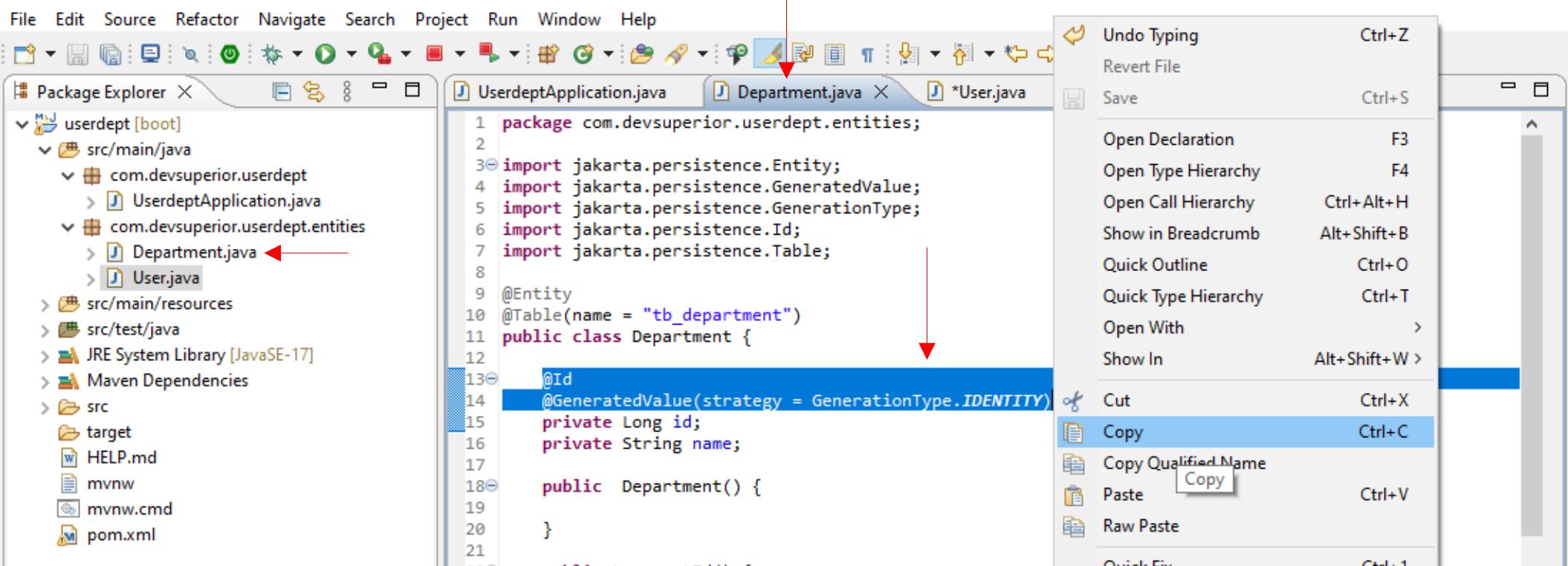


workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/User.java - Spring Tool Suite 4

User Application.java *Department.java *User.java

```
1 package com.devsuperior.userdept.entities;
2
3
4 import jakarta.persistence.Entity;
5 import jakarta.persistence.GeneratedValue;
6 import jakarta.persistence.GenerationType;
7 import jakarta.persistence.Id;
8 import jakarta.persistence.Table;
9
10 @Entity
11 @Table(name = "tb_department")
12 public class User {
13
14     private Long id;
15     private String name;
16     private String email;
17
18     private Department department;
19
20     public User() {
21
22 }
```

paste



Repita o procedimento copy da class Department.java e cole na User.java

The screenshot shows the Spring Tool Suite 4 interface. The left side features the Package Explorer view, which displays the project structure of 'userdept'. The 'src/main/java' folder contains packages 'com.devsuperior.userdept' and 'com.devsuperior.userdept.entities'. Within 'com.devsuperior.userdept.entities', files 'Department.java' and 'User.java' are listed. The right side shows the code editor with three tabs: 'UserdeptApplication.java', 'Department.java', and 'User.java'. The 'User.java' tab is active, displaying Java code for a 'User' entity. The code includes imports for jakarta.persistence.Entity, GeneratedValue, GenerationType, Id, and Table. It defines an @Entity with a @Table(name = "tb_user") annotation. The class 'User' has an @Id annotated with @GeneratedValue(strategy = GenerationType.IDENTITY). It also includes private fields for id, name, email, and department, and a constructor.

```
1 package com.devsuperior.userdept.entities;
2
3
4 import jakarta.persistence.Entity;
5 import jakarta.persistence.GeneratedValue;
6 import jakarta.persistence.GenerationType;
7 import jakarta.persistence.Id;
8 import jakarta.persistence.Table;
9
10 @Entity
11 @Table(name = "tb_user")
12 public class User {
13
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private Long id;
17     private String name;
18     private String email;
19
20     private Department department;
21
22     public User() {
```

De onde vem os dados

☰ README.md

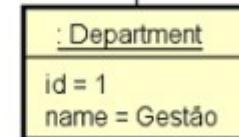
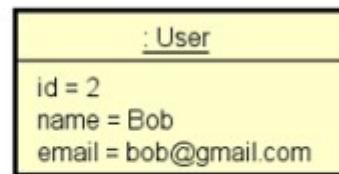
Desenvolvimento moderno: relacional -> objeto -> json

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



Adicione as anotações

The screenshot shows the Spring Tool Suite 4 interface with the following details:

- Project Explorer:** Shows the project structure under the package `userdept`. The `User.java` file is selected.
- Editor:** Displays the code for `User.java`. Red arrows highlight specific annotations:
 - An arrow points to the `@Entity` annotation at line 10.
 - An arrow points to the `@Table(name = "tb_user")` annotation at line 11.
 - An arrow points to the `@Id` annotation at line 14.
 - An arrow points to the `@GeneratedValue(strategy = GenerationType.IDENTITY)` annotation at line 15.
 - An arrow points to the `@ManyToOne` annotation at line 20.
- Toolbar:** The toolbar includes various icons for file operations, search, and navigation.
- MenuBar:** Includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.

```
1 package com.devsuperior.userdept.entities;
2
3
4 import jakarta.persistence.Entity;
5 import jakarta.persistence.GeneratedValue;
6 import jakarta.persistence.GenerationType;
7 import jakarta.persistence.Id;
8 import jakarta.persistence.Table;
9
10 @Entity
11 @Table(name = "tb_user")
12 public class User {
13
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private Long id;
17     private String name;
18     private String email;
19
20     @ManyToOne
21     private Department department;
22
23     public User() {
24
25     }
26 }
```



```
1 package com.devsuperior.userdept.entities;
2
3
4 import jakarta.persistence.Entity;
5 import jakarta.persistence.GeneratedValue;
6 import jakarta.persistence.GenerationType;
7 import jakarta.persistence.Id;
8 import jakarta.persistence.Table;
9
10 @Entity
11 @Table(name = "tb_user")
12 public class User {
13
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private Long id;
17     private String name;
18     private String email;
```

@ManyToOne

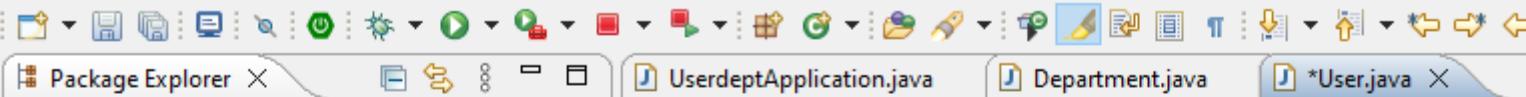
- pr ← Import 'ManyToOne' (jakarta.persistence)
- pu @ Create annotation 'ManyToOne'
- ↳ Change to 'ManyToMany' (org.hibernate.annotations)
- ↳ Change to 'ManyToOne' (jakarta.persistence)
- pu Rename in file (Ctrl+2, R)
- ↳ Fix project setup...

```
...
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table;
...

```

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/User.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help



```
2
3
4① import jakarta.persistence.Entity;
5 import jakarta.persistence.GeneratedValue;
6 import jakarta.persistence.GenerationType;
7 import jakarta.persistence.Id;
8 import jakarta.persistence.ManyToOne; ←
9 import jakarta.persistence.Table;
10
11 @Entity
12 @Table(name = "tb_user")
13 public class User {
14
15②     @Id
16     @GeneratedValue(strategy = GenerationType.IDENTITY)
17     private Long id;
18     private String name;
19     private String email;
20
21③     @ManyToOne
22     private Department department;
23
24④     public User() {
25
26     }
```

De onde vem os dados

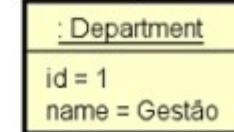
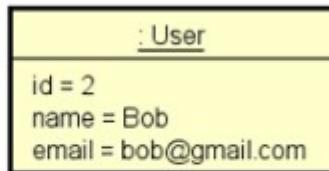
Desenvolvimento moderno: relacional -> objeto -> json

tb_department

id	name
1	Gestão
2	Informática

tb_user

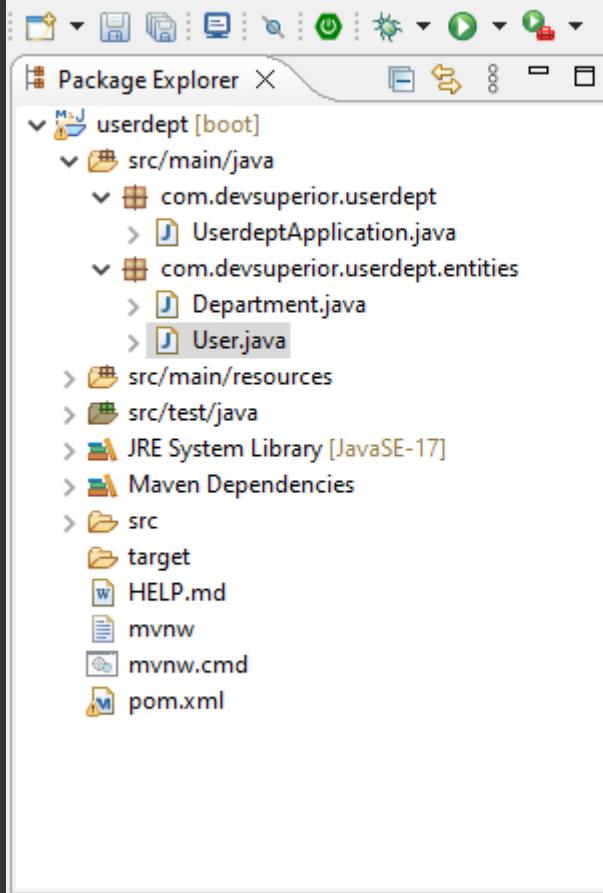
id	name	email	department_id
1	Maria	maria@gmail.com	1
2	Bob	bob@gmail.com	1
3	Alex	alex@gmail.com	2
4	Ana	ana@gmail.com	2





workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/User.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help



```
2
3
4 import jakarta.persistence.Entity;
5 import jakarta.persistence.GeneratedValue;
6 import jakarta.persistence.GenerationType;
7 import jakarta.persistence.Id;
8 import jakarta.persistence.ManyToOne;
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     @ManyToOne
22     @JoinColumn(←
23     private Department department;
24
25     public User() {
26
27     }
```

Package Explorer X

- userdept [boot]
 - src/main/java
 - com.devsuperior.userdept
 - UserdeptApplication.java
 - com.devsuperior.userdept.entities
 - Department.java
 - User.java
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-17]
 - Maven Dependencies
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml

```

2
3
4 import jakarta.persistence.Entity;
5 import jakarta.persistence.GeneratedValue;
6 import jakarta.persistence.GenerationType;
7 import jakarta.persistence.Id;
8 import jakarta.persistence.ManyToOne;
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     @ManyToOne
22     @JoinColumn(
23         name = "fk_department",
24         referencedColumnName = "id",
25         nullable = false
26     )
27     private Department department;
28
29 }
30

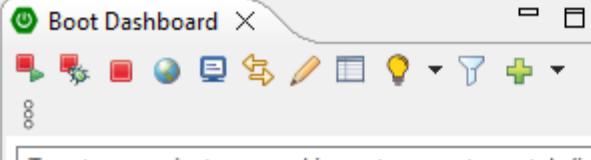
```

- Import 'JoinColumn' (jakarta.persistence)**
- Create annotation 'JoinColumn'**
 - Change to 'JoinColumnOrFormula' (org.hibernate.annotations)
 - Change to 'JoinColumns' (jakarta.persistence)
 - Change to 'MapKeyJoinColumn' (jakarta.persistence)
 - Change to 'PrimaryKeyJoinColumn' (jakarta.persistence)
- Rename in file (Ctrl+2, R)
- Fix project setup...

```

...
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
...

```



workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/entities/User.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X UserdeptApplication.java Department.java *User.java X

```
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.ManyToOne;
9 import jakarta.persistence.Table;
10
11
12 @Entity
13 @Table(name = "tb_user")
14 public class User {
15
16     @Id
17     @GeneratedValue(strategy = GenerationType.IDENTITY)
18     private Long id;
19     private String name;
20     private String email;
21
22     @ManyToOne
23     @JoinColumn
24     private Department department;
25
26     public User() {
27
28     }
29 }
```



The screenshot shows the Spring Tool Suite interface with the following details:

- Package Explorer:** Shows the project structure under the package `userdept`. The `src/main/java` folder contains the `UserDeptApplication.java`, `Department.java`, and `User.java` files.
- User.java Code Editor:** Displays the Java code for the `User` entity. The code uses Jakarta Persistence annotations (`@Entity`, `@Table`, `@Id`, `@GeneratedValue`, `@ManyToMany`, `@JoinColumn`) and imports from `jakarta.persistence`.
- Toolbars:** Standard Eclipse-style toolbars for file operations, search, and navigation.
- Status Bar:** Shows the path `userdept/src/main/java/com/devsuperior/userdept/entities/User.java` and the version `4.17.1.RELEASE`.

```
3  
4 import jakarta.persistence.Entity;  
5 import jakarta.persistence.GeneratedValue;  
6 import jakarta.persistence.GenerationType;  
7 import jakarta.persistence.Id;  
8 import jakarta.persistence.JoinColumn;  
9 import jakarta.persistence.ManyToOne;  
10 import jakarta.persistence.Table;  
11  
12 @Entity  
13 @Table(name = "tb_user")  
14 public class User {  
15  
16 @Id  
17 @GeneratedValue(strategy = GenerationType.IDENTITY)  
18 private Long id;  
19 private String name;  
20 private String email;  
21  
22 @ManyToOne  
23 @JoinColumn(name = "Department_id")|  
24 private Department department;  
25  
26 public User() {  
27  
28 }  
29
```

Configurando o banco de dados h2

(101) Seu primeiro projeto Java x GitHub - devsuperior/java-web-s x localhost:8080 x +

← → C 🔒 github.com/devsuperior/java-web-spring-2022

c++ C++ C++ br-C.org C++ VISUAL STUDIO Domestika ASSEMBLY ss64 kernel org Linux kernel gcc A

☰ README.md

Configurações 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
```

copy

Configurações 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= [REDACTED]

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

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

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/resources/application.properties - Spring Tool Suite 4

File Edit Navigate Search Project Run Window Help

Package Explorer X UserdeptApplication.java Department.java *User.java *application.properties X

```
1 # Dados de conexão com o banco H2
2 spring.datasource.url=jdbc:h2:mem:testdb
3 spring.datasource.username=sa
4 spring.datasource.password=
5
6 # Configuração do cliente web do banco H2
7 spring.h2.console.enabled=true
8 spring.h2.console.path=/h2-console
9
10 # Configuração para mostrar o SQL no console
11 spring.jpa.show-sql=true
12 spring.jpa.properties.hibernate.format_sql=true
13
```

The screenshot shows the Spring Tool Suite 4 interface. On the left, the Package Explorer displays the project structure for 'userdept'. It includes a 'src/main/java' folder containing 'UserdeptApplication.java' and 'com.devsuperior.userdept' package with 'Department.java' and 'User.java'. It also contains a 'src/main/resources' folder with 'application.properties', 'static', and 'templates' subfolders. Below these are 'src/test/java', 'JRE System Library [JavaSE-17]', 'Maven Dependencies', 'src', 'target', 'HELP.md', 'mvnw', 'mvnw.cmd', and 'pom.xml'. A red arrow points from the 'application.properties' file in the Package Explorer to its tab in the central editor. Another red arrow points from the 'application.properties' file in the editor to the bottom right corner of the screen.

cole

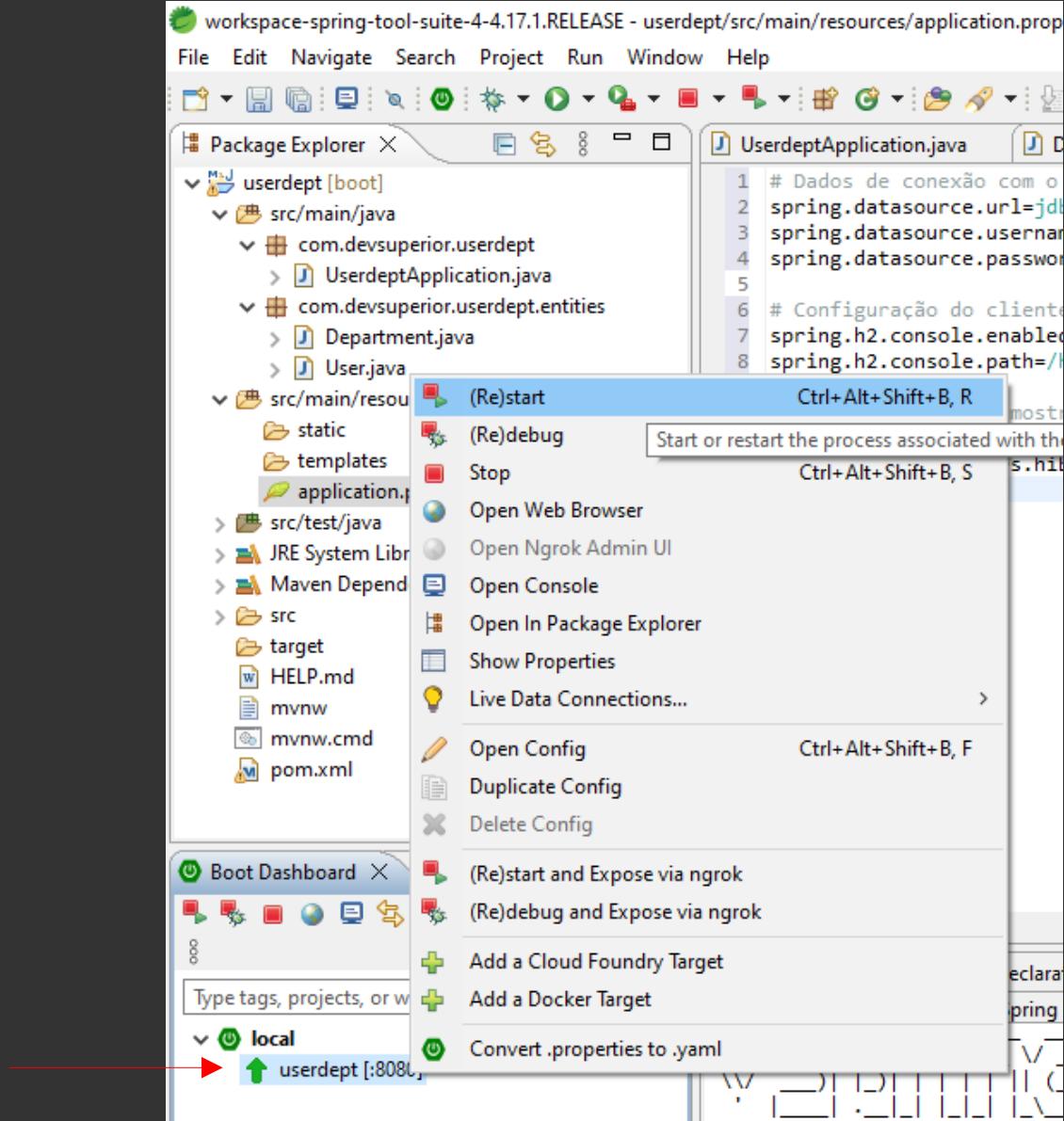
workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/resources/application.properties - Spring Tool Suite 4

File Edit Navigate Search Project Run Window Help

Package Explorer X UserdeptApplication.java Department.java *User.java *application.properties X

```
1 # Dados de conexão com o banco H2
2 spring.datasource.url=jdbc:h2:mem:testdb ←
3 spring.datasource.username=sa ←
4 spring.datasource.password= ←
5
6 # Configuração do cliente web do banco H2
7 spring.h2.console.enabled=true ←
8 spring.h2.console.path=/h2-console ←
9
10 # Configuração para mostrar o SQL no console
11 spring.jpa.show-sql=true ←
12 spring.jpa.properties.hibernate.format_sql=true ←
13
```

Leias esses dados de conexão são importantes



No original ele mostrava formato sql
aqui não sei se é erro

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

- Package Explorer:** Shows the project structure for "userdept [boot]".
- Editor:** Displays the contents of the "application.properties" file. A red arrow points to the line "spring.jpa.show-sql=true".
- Outline:** Shows the class hierarchy for "UserdeptApplication.java", "Department.java", "User.java", and "application.properties".
- Problems:** Shows no errors.
- Console:** Shows the application logs from "UserdeptApplication [Spring Boot App]". The logs indicate the application is running on port 8080 and shows various initialization steps for Spring Data JPA, Tomcat, and Hibernate.
- Boot Dashboard:** Shows the status of the application "userdept [:8080]".
- Bottom Bar:** Shows system icons for search, file, folder, browser, file manager, and others, along with the current weather (27°C Nublado) and time (14:10).

```

# 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

```

```

2023-01-20T14:10:31.782-03:00  INFO 7520 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repository scanning for repository interfaces...
2023-01-20T14:10:31.830-03:00  INFO 7520 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning
2023-01-20T14:10:32.724-03:00  INFO 7520 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-01-20T14:10:32.750-03:00  INFO 7520 --- [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-01-20T14:10:32.751-03:00  INFO 7520 --- [           main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/1]
2023-01-20T14:10:32.976-03:00  INFO 7520 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplication Context
2023-01-20T14:10:32.976-03:00  INFO 7520 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization...
2023-01-20T14:10:33.250-03:00  INFO 7520 --- [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-01-20T14:10:33.611-03:00  INFO 7520 --- [           main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection conn0: ur...
2023-01-20T14:10:33.616-03:00  INFO 7520 --- [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2023-01-20T14:10:33.716-03:00  INFO 7520 --- [           main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo
2023-01-20T14:10:33.847-03:00  INFO 7520 --- [           main] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.1
2023-01-20T14:10:34.160-03:00  WARN 7520 --- [           main] org.hibernate.orm.deprecation : HHH9000021: Encountered deprecated setti...
2023-01-20T14:10:34.705-03:00  INFO 7520 --- [           main] SQL dialect
2023-01-20T14:10:36.139-03:00  INFO 7520 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementati...
2023-01-20T14:10:36.156-03:00  INFO 7520 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for ...
2023-01-20T14:10:36.222-03:00  WARN 7520 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by def...
2023-01-20T14:10:36.802-03:00  INFO 7520 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) wi...
2023-01-20T14:10:36.821-03:00  INFO 7520 --- [           main] c.d.userdept.UserdeptApplication : Started UserdeptApplication in 7.025 seconds

```

Seu primeiro projeto Java web no Spring Boot 2022

Pressione **Esc** para sair do modo tela cheia

```

2
3 import javax.persistence.Entity;
4 import javax.persistence.GeneratedValue;
5 import javax.persistence.GenerationType;
6 import javax.persistence.Id;
7 import javax.persistence.JoinColumn;
8 import javax.persistence.ManyToOne;
9 import javax.persistence.Table;
10
11 @Entity
12 @Table(name = "tb_user")
13 public class User {
    
```

primary key (id)

)

Hibernate:

```

alter table tb_user
    add constraint FKk05pfkuxp9jsie8074d304v7m
        foreign key (department_id)
            references tb_department

```

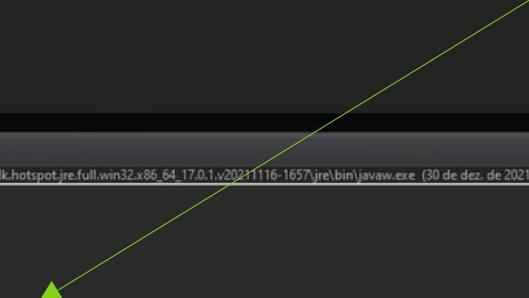
2021-12-30 11:15:03.401 INFO 22320 --- [main] o.h.e.t.j.p.i.J

2021-12-30 11:15:03.429 INFO 22320 --- [main] j.LocalContainer

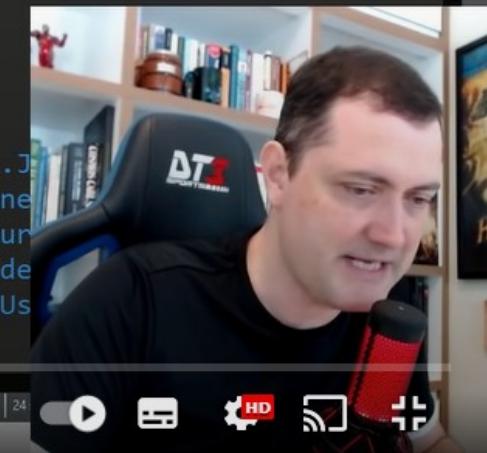
2021-12-30 11:15:03.550 WARN 22320 --- [main] JpaBaseConfigur

2021-12-30 11:15:04.666 INFO 22320 --- [main] o.s.b.w.embedde

2021-12-30 11:15:04.695 INFO 22320 --- [main] c.d.userdept.Us



original



A screenshot of a web browser window. The address bar shows the URL `localhost:8080`. The page content is a "Whitelabel Error Page". The text on the page reads:

This application has no explicit mapping for /error, so you are seeing this as a fallback.
Fri Jan 20 12:40:50 BRT 2023
There was an unexpected error (type=Not Found, status=404).

Tentando resolver erro

(104) Seu primeiro projeto Java w X GitHub - devssuperior/java-web-s X localhost:8080/h2-console X | +

← → C github.com/devsuperior/java-web-spring-2022

c++ C++ C++ br-C.org C++ VISUAL STUDIO Domestika ASSEMBLY ss64 kernel org Linux kernel gcc

☰ README.md

- Criar os endpoints da API REST

Trechos 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>
```

copy

paste

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/pom.xml - Spring Tool Suite 4

File Edit Source Navigate Search Project Run Window Help

Package Explorer X UserdeptApplication.java Department.java User.java application.properties userdept/pom.xml X

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
</dependency>
</dependencies>
<build> ←
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-resources-plugin</artifactId>
            <version>3.1.0</version>
        </plugin>
    </plugins> ←
</build>
</project>
```

userdept [boot]

- src/main/java
 - com.devsuperior.userdept
 - UserDeptApplication.java
 - com.devsuperior.userdept.entities
 - Department.java
 - User.java
- src/main/resources
 - static
 - templates
 - application.properties
- src/test/java
- JRE System Library [JavaSE-17]
- Maven Dependencies
- src
- target
- HELP.md
- mvnw
- mvnw.cmd

pom.xml ←

The screenshot shows the Spring Tool Suite interface with the following components:

- Package Explorer:** Shows the project structure under "userdept [boot]". It includes:
 - src/main/java:** Contains `com.devsuperior.userdept` (with `UserDeptApplication.java`) and `com.devsuperior.userdept.entities` (with `Department.java` and `User.java`).
 - src/main/resources:** Contains `static`, `templates`, and `application.properties`.
 - src/test/java**
 - JRE System Library [JavaSE-17]**
 - Maven Dependencies**
 - src**
 - target**
 - HELP.md**
 - mvnw**
 - mvnw.cmd**
 - pom.xml**
- UserDeptApplication.java Editor:** Displays the main application class code.
- pom.xml Editor:** Displays the Maven configuration file, specifically the `<dependencies>` section.

Como resolvi esse problema

The screenshot shows the Spring Tool Suite 4 interface. On the left is the Package Explorer view, which displays the project structure for 'userdept [boot]'. The right side shows the content of the 'pom.xml' file in the editor. The code is as follows:

```
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-resources-plugin</artifactId>
            <version>3.1.0</version>
        </plugin>
    </plugins>
</build>
</project>
```

A red arrow points from the text 'Repare nesse warning do arquivo pom.xml' at the bottom of the slide to the line containing the 'maven-resources-plugin' element in the code.

Repare nesse warning do arquivo pom.xml

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

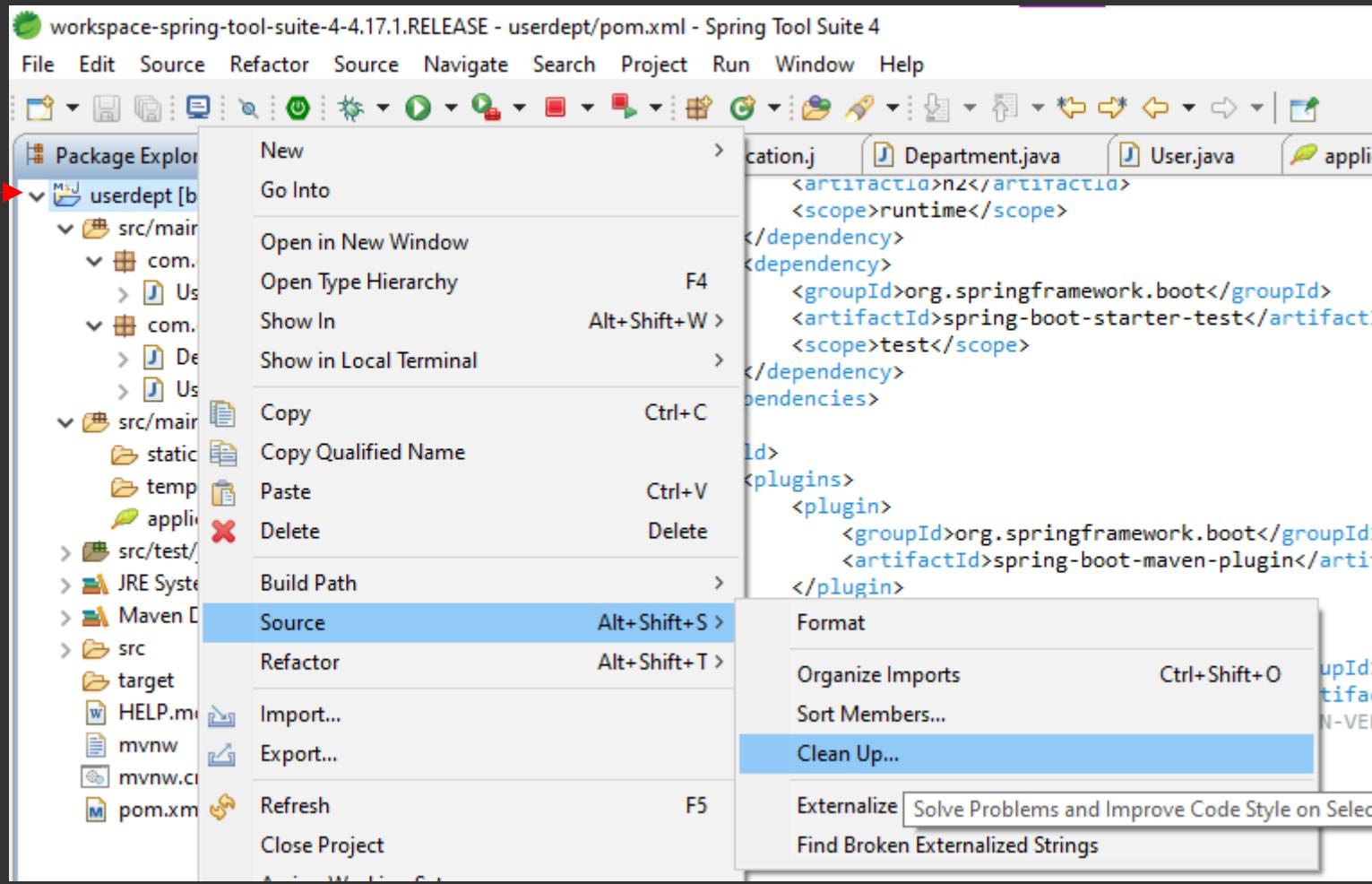
- Project Explorer:** Shows the project structure with files like `src/main/resources`, `src/test/java`, `JRE System Library [JavaSE-17]`, `Maven Dependencies`, `src`, `target`, `HELP.md`, `mvnw`, `mvnw.cmd`, and `pom.xml`.
- Java Editor:** Displays the contents of the `pom.xml` file.
- Problems View:** Shows the "Problems" tab selected, indicating there are 0 items.

A red arrow points from the text "Selecionei essa guia problems cliquei com o direito e fui na opção quick fix" to the "Problems" tab in the bottom navigation bar.

Text Overlay:

Selecionei essa guia problems cliquei com o direito e fui na opção quick fix
Fix isso para todos os warnings do arquivo pom.xml
E depois fui na guia do projeto e limpei só depois disso dei restart

```
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-resources-plugin</artifactId>
            <version>3.0.1</version><!--$NO-MVN-MAN-VER$-->
        </plugin>
    </plugins>
</build>
</project>
```



workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/pom.xml - Spring Tool Suite 4

File Edit Source Refactor Source Navigate Search Project Run Window Help

Package Explorer X UserdeptApplication.java

userdept [boot]

src/main/java

com.devsuperior.userdept

UserdeptApplication.java

com.devsuperior.userdept.entities

Department.java

User.java

src/main/resources

static

templates

application.properties

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

target

HELP.md

mvnw

mvnw.cmd

pom.xml

(Re)start Ctrl+Alt+Shift+B, R

(Re)debug Ctrl+Alt+Shift+B, D

Stop

Start or restart the process associated with the current configuration

Open Web Browser

Open Ngrok Admin UI

Open Console

Open In Package Explorer

Show Properties

Live Data Connections...

Open Config Ctrl+Alt+Shift+B, F

Duplicate Config

Delete Config

(Re)start and Expose via ngrok

(Re)debug and Expose via ngrok

Add a Cloud Foundry Target

Add a Docker Target

Boot Dashboard X

Type tags, projects, or words

local

userdept [:8080]

File Edit Source Navigate Project Run Window Help

Package Explorer X UserdeptApplication.java Department.java User.java application.properties userdept/pom.xml X

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Problems @ Javadoc Declaration Console X

userdept - UserdeptApplication [Spring Boot App] C:\Users\brito\Downloads\spring\sts-4.17.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\java

```
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 (
        id bigint generated by default as identity,
        email varchar(255),
        name varchar(255),
        department_id bigint,
        primary key (id)
    )
Hibernate:
    alter table if exists tb_user
        add constraint FKk05pfkuxp9jsie8074d304v7m
        foreign key (department_id)
        references tb_department
```

2023-01-20T14:27:34.557-03:00 INFO 10252 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation

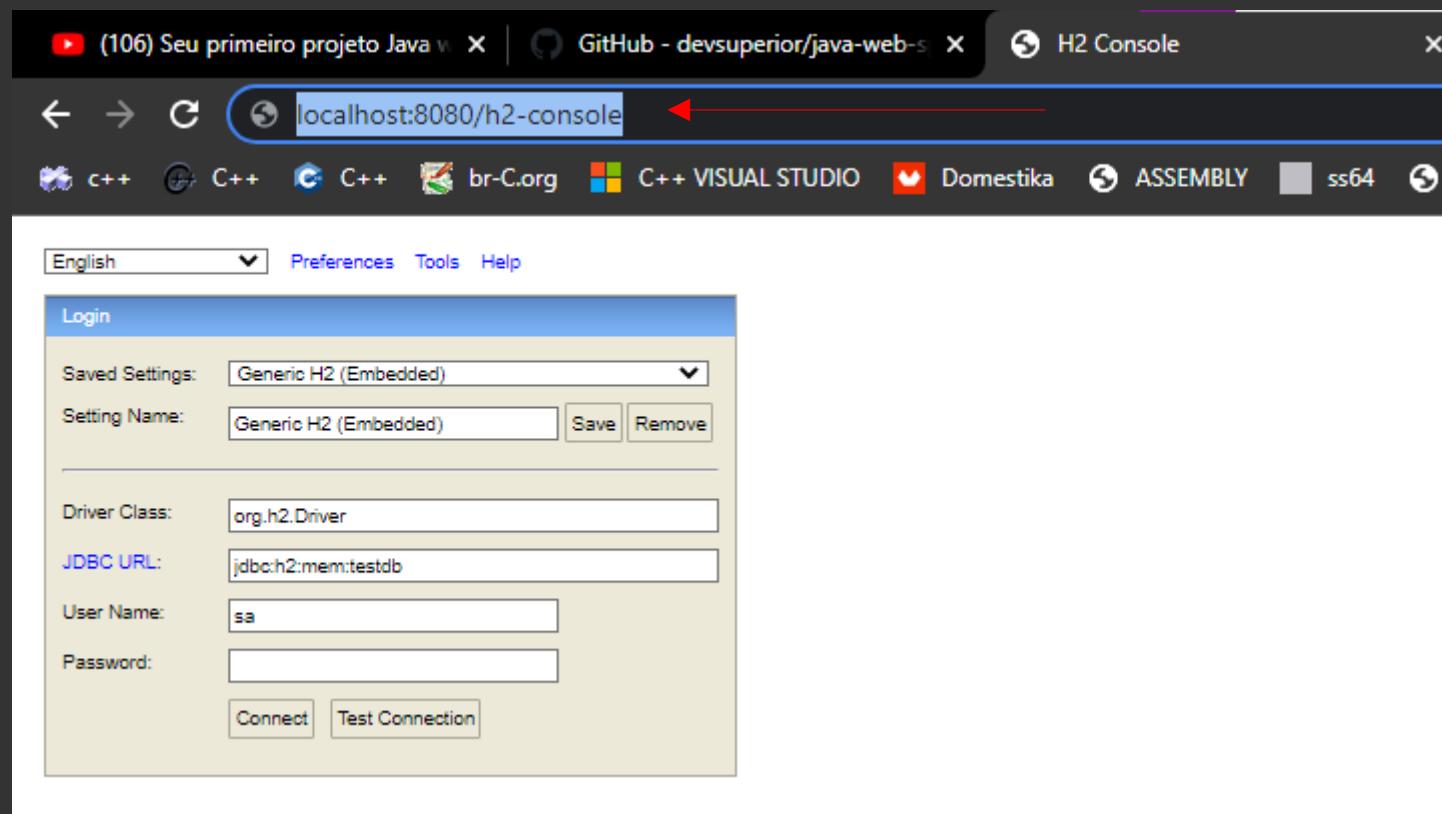
2023-01-20T14:27:34.579-03:00 INFO 10252 --- [main] f.l.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'userdept'

Outline X Cache file:///C:/Users/brito/.lemon

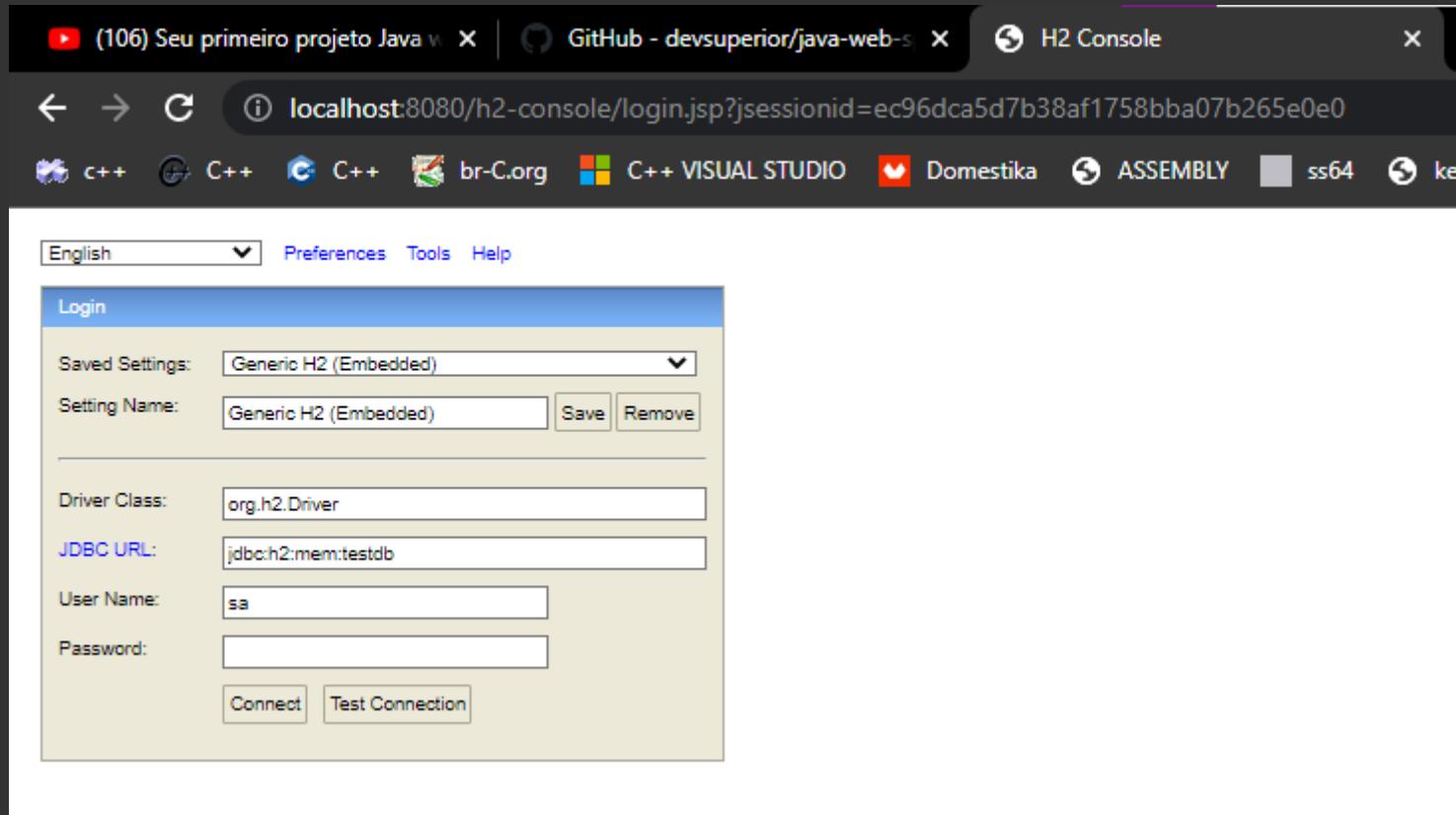
xml project modelVersion

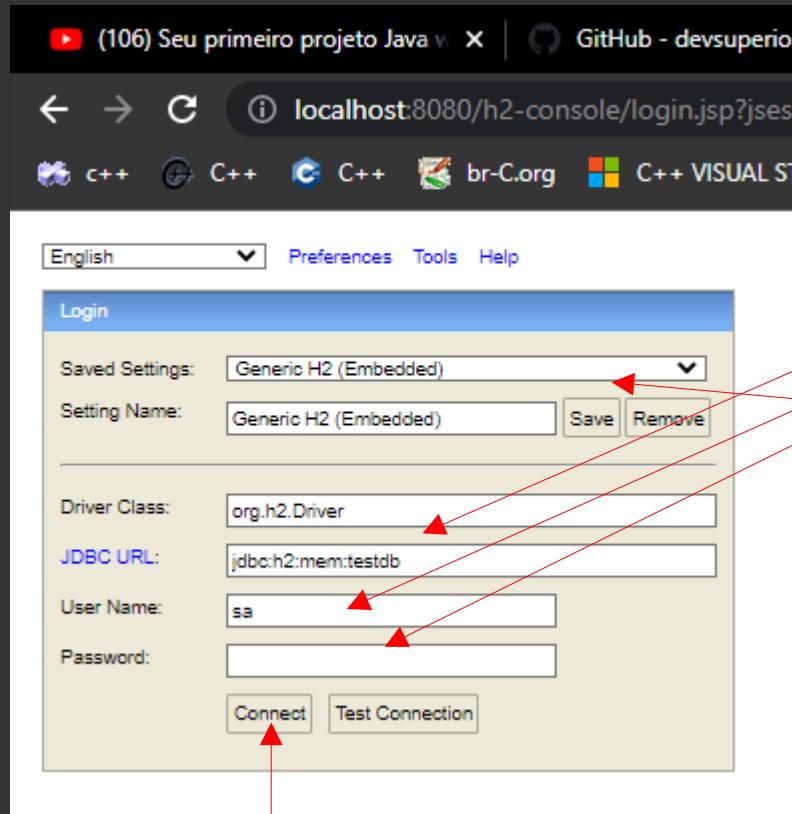
O console deve exibir isso

Digite no navegador



Problema resolvido





```
UserdeptApplication.java Department.java User.java application.p  
1 # Dados de conexão com o banco H2  
2 spring.datasource.url=jdbc:h2:mem:testdb  
3 spring.datasource.username=sa  
4 spring.datasource.password=  
5  
6 # Configuração do cliente web do banco H2  
7 spring.h2.console.enabled=true  
8 spring.h2.console.path=/h2-console  
9  
10 # Configuração para mostrar o SQL no console  
11 spring.jpa.show-sql=true  
12 spring.jpa.properties.hibernate.format_sql=true  
13
```

Click em conectar

YouTube (106) Seu primeiro projeto Java | GitHub - devsuperior/java-web-s | H2 Console

localhost:8080/h2-console/login.do?jsessionid=ec96dca5d7b38af1758bba07b265e0e0

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

Run Selected | Auto complete | Clear | SQL statement:

jdbs:h2:mem:testdb
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.

28°C Nublado | 14:44

The screenshot shows the H2 Console interface running in a browser window. The title bar indicates the connection is to 'localhost:8080/h2-console/login.do?jsessionid=ec96dca5d7b38af1758bba07b265e0e0'. The toolbar includes various icons for file operations and database management. The main area has a sidebar with database schema information and a central SQL editor window. A red arrow points from the left towards the sidebar, and another red arrow points down from the top towards the SQL editor.

Auto commit

Max rows: 1000

Auto complete Off

Auto select On

Run Selected

SQL statement:

```
SELECT * FROM TB_DEPARTMENT
```

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

Selecione a tb_department depois click em run

The screenshot shows the H2 Console interface running in a browser window. The title bar indicates the connection is to 'Seu primeiro projeto Java' at port 8080. The main area displays a database schema on the left and a SQL query editor on the right.

Schemas:

- jdbc:h2:mem:testdb
- TB_DEPARTMENT
- TB_USER
- INFORMATION_SCHEMA
- Users

SQL Statement:

```
SELECT * FROM TB_DEPARTMENT;
```

Result:

ID	NAME
(no rows, 7 ms)	

Toolbar Buttons:

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

Header Options:

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

Selecione clear e repita pra tb_user

The screenshot shows the H2 Console interface running in a web browser. The title bar indicates the window is titled "H2 Console". The address bar shows the URL "localhost:8080/h2-console/login.do?jsessionid=ec96dca5d7b38af1758bba07b265e0e0". Below the address bar is a toolbar with various icons, including C++, C, and assembly language options. The main area is divided into two sections: a database browser on the left and a SQL editor on the right.

Database Browser: On the left, there is a tree view of the database schema. It includes a connection node "jdbc:h2:mem:testdb", tables "TB_DEPARTMENT" and "TB_USER", a folder "INFORMATION_SCHEMA", a users section, and a version entry "H2 2.1.214 (2022-08-13)".

SQL Editor: On the right, there is a large text input field for SQL statements. At the top of this field, there are several buttons: "Run", "Run Selected", "Auto complete", "Clear", and "SQL statement:". A red arrow points upwards from the bottom of the SQL editor towards the "Clear" button. Below the editor, the results of a previous query are displayed:

```
SELECT * FROM TB_DEPARTMENT;
ID NAME
(no rows, 7 ms)
```

At the bottom of the SQL editor, there is an "Edit" button.

(106) Seu primeiro projeto Java w GitHub - devsuperior/java-web-s H2 Console

localhost:8080/h2-console/login.do?jsessionid=ec96dca5d7b38af1758bba07b265e0e0

c++ C++ C++ br-C.org C++ VISUAL STUDIO Domestika ASSEMBLY ss64 kernel org

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_USER;

SELECT * FROM TB_USER;

ID	EMAIL	NAME	DEPARTMENT_ID
(no rows, 1 ms)			

Edit

☰ README.md

Populando o banco de dados copy

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

Script SQL

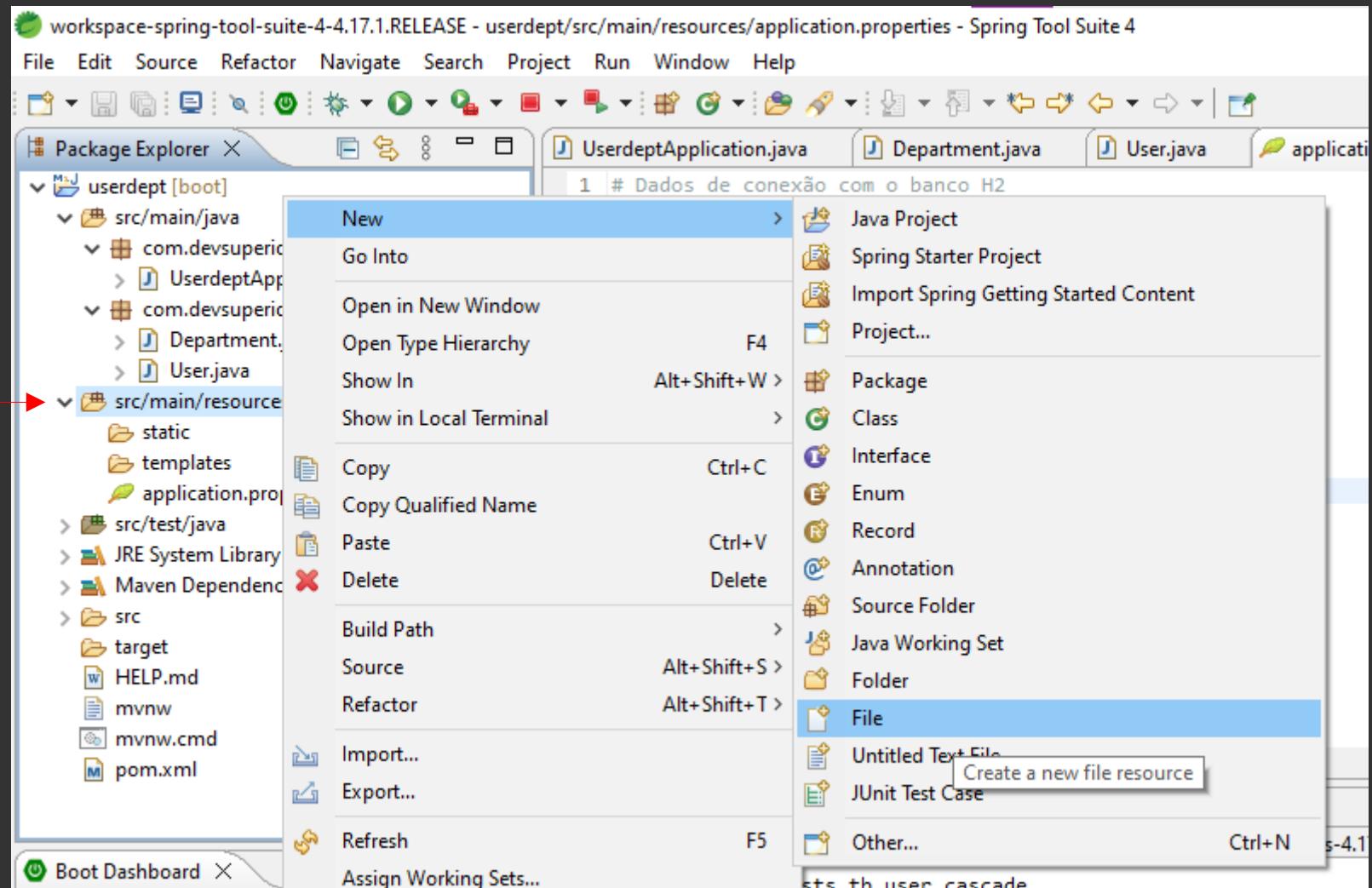
```
INSERT INTO tb_department(name) VALUES ('Gestão');
INSERT INTO tb_department(name) VALUES ('Informática');

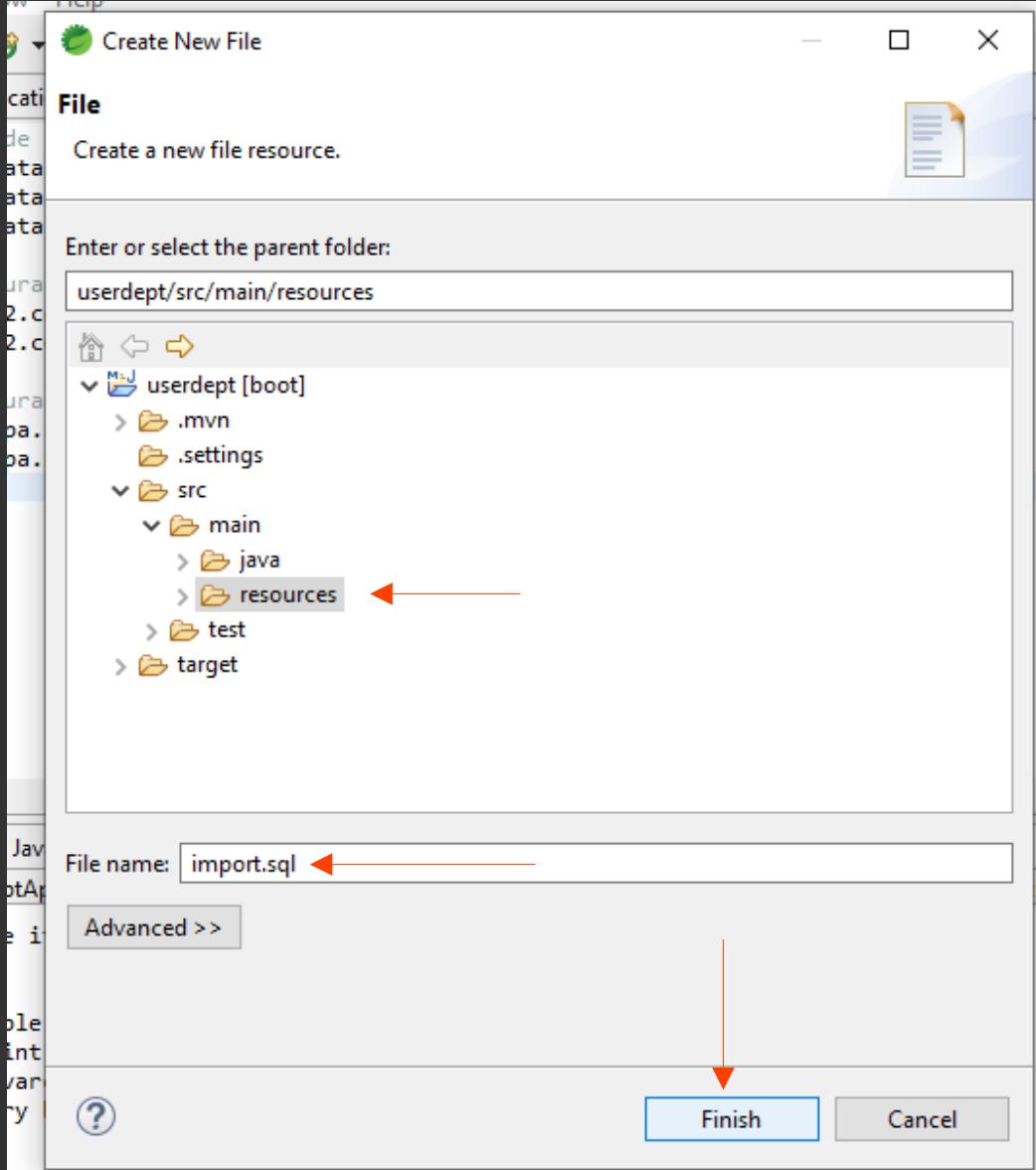
INSERT INTO tb_user(department_id, name, email) VALUES (1, 'Maria', 'maria@gmail.com');
INSERT INTO tb_user(department_id, name, email) VALUES (1, 'Bob', 'bob@gmail.com');
INSERT INTO tb_user(department_id, name, email) VALUES (2, 'Alex', 'alex@gmail.com');
INSERT INTO tb_user(department_id, name, email)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

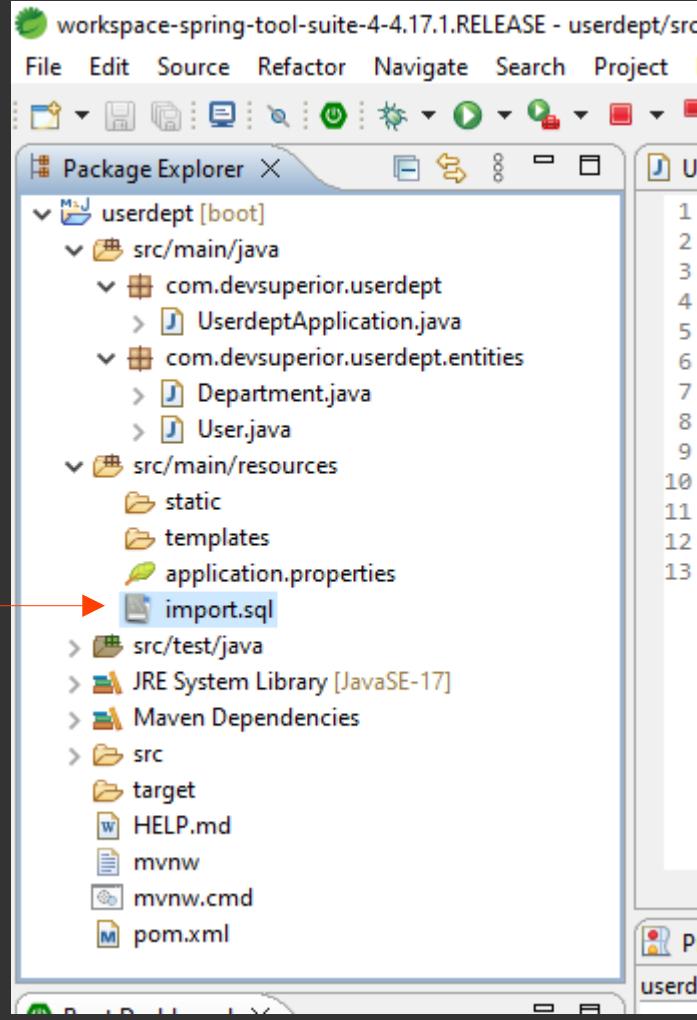


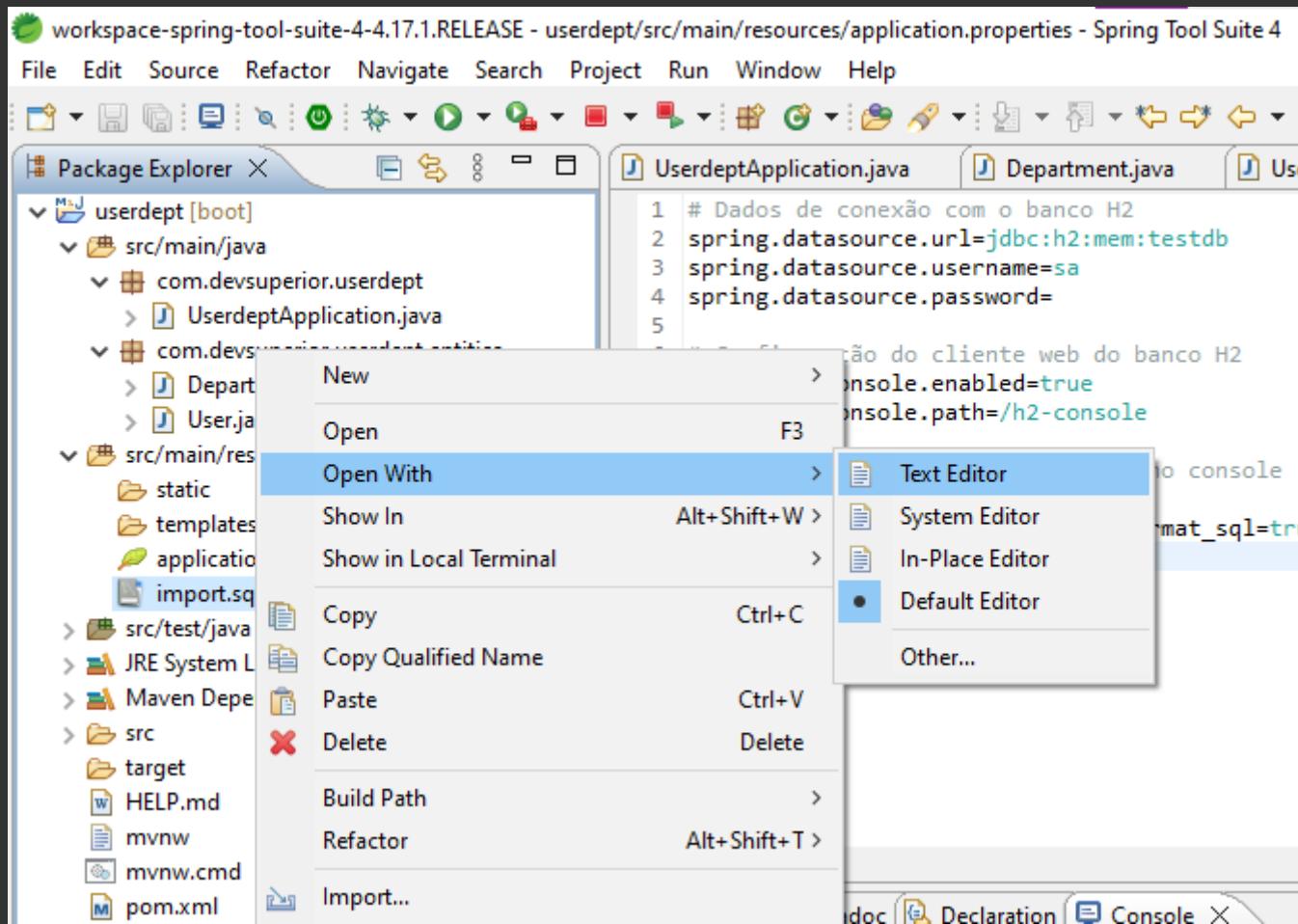
Collection Postman

<https://www.getpostman.com/collections/ac4a49113c4024e47d4f>





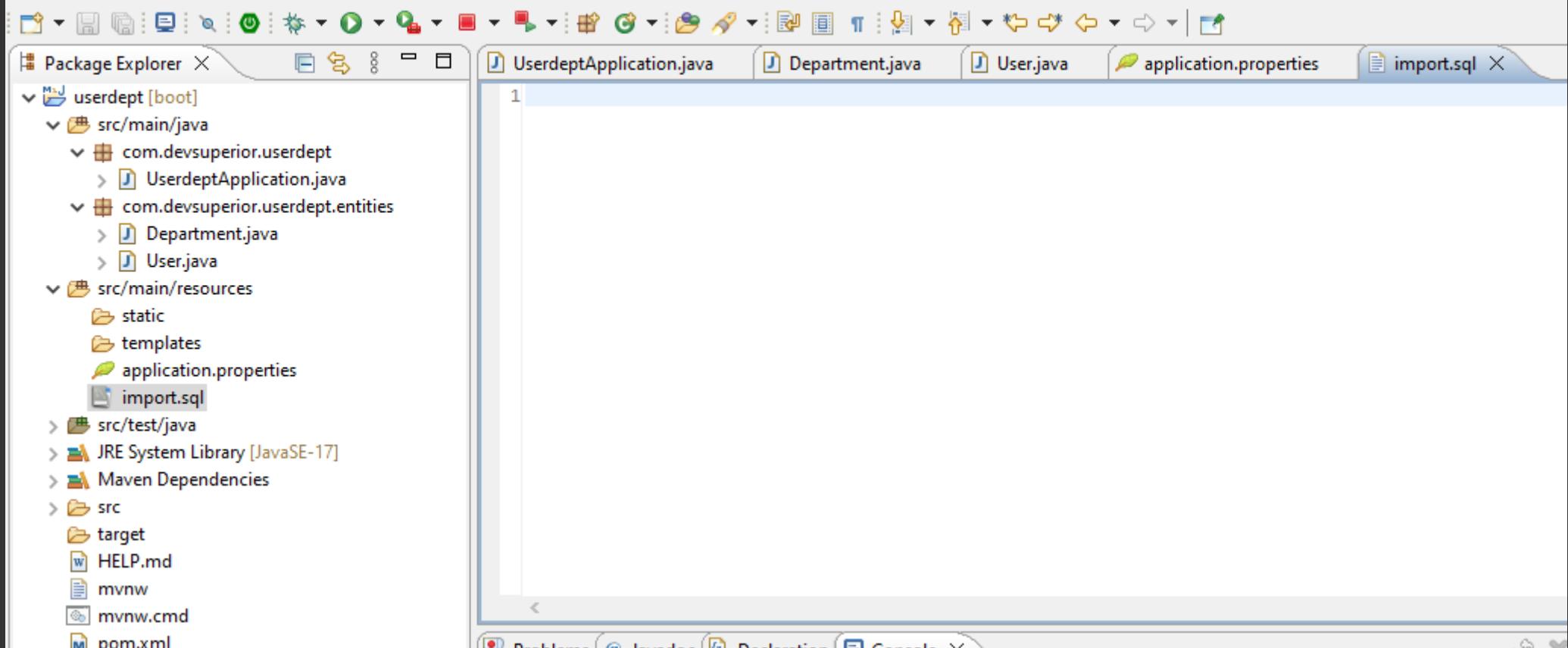




Cole ai

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/resources/import.sql - Spring Tool Suite 4

File Edit Navigate Search Project Run Window Help





Package Explorer X

- ✓ userdept [boot]
- ✓ src/main/java
 - ✓ com.devsuperior.userdept
 - > UserdeptApplication.java
 - ✓ com.devsuperior.userdept.entities
 - > Department.java
 - > User.java
 - ✓ src/main/resources
 - ✓ static
 - ✓ templates
 - ✓ application.properties
 - ✓ import.sql
 - > src/test/java
 - > JRE System Library [JavaSE-17]
 - > Maven Dependencies
 - > src
 - ✓ target
 - ✓ HELP.md
 - ✓ mvnw
 - ✓ mvnw.cmd
 - ✓ pom.xml

UserdeptApplication.java

```
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');
```

8 |

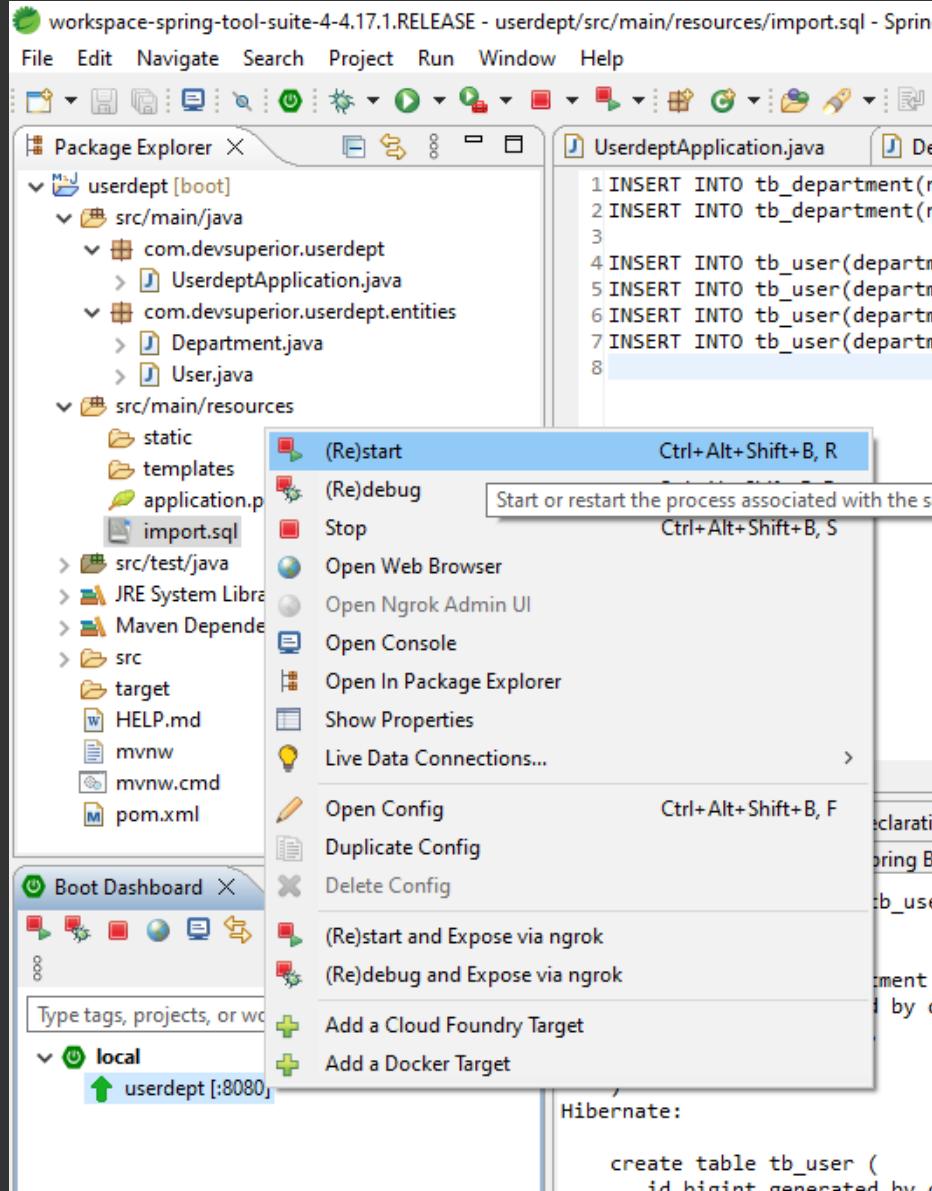
Problems

@ Javadoc

Declaration

Console X

Re um restart e
confira o console



Esses dados devem aparecer

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

- Project Explorer (left):** Displays the project structure with files like User.java, application.properties, import.sql, and various configuration and build files.
- Boot Dashboard (bottom left):** Shows the application 'userdept' running on port 8080.
- Console (bottom right):** Displays the application logs and database schema creation commands.

Console Output:

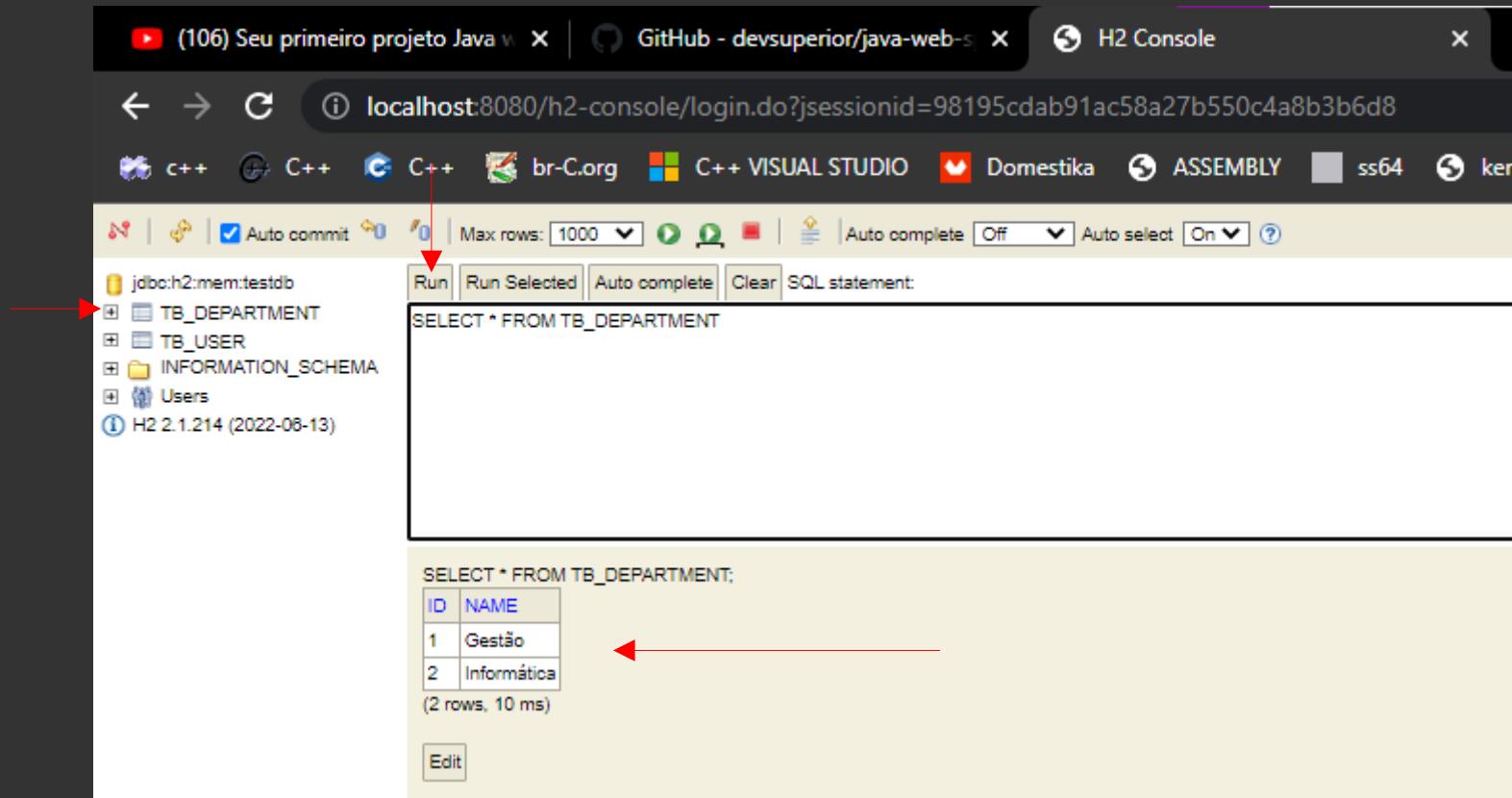
```
userdept - UserdeptApplication [Spring Boot App] C:\Users\brito\Downloads\spring\sts-4.17.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\lib\main.jar
Hibernate:
    alter table if exists tb_user
        add constraint FKk05pfkuxp9jsie8074d304v7m
            foreign key (department_id)
                references tb_department
2023-01-20T14:59:31.293-03:00  INFO 9232 --- [           main] o.h.t.schema.internal.SchemaCreatorImpl : HHH000476:
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')
2023-01-20T14:59:31.307-03:00  INFO 9232 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator      : HHH000490:
2023-01-20T14:59:31.323-03:00  INFO 9232 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized
2023-01-20T14:59:31.385-03:00  WARN 9232 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.
2023-01-20T14:59:31.905-03:00  INFO 9232 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat star
2023-01-20T14:59:31.922-03:00  INFO 9232 --- [           main] c.d.userdept.UserdeptApplication       : Started Use
```

The screenshot shows the H2 Console interface running in a browser window. The title bar indicates the connection is to 'localhost:8080/h2-console/login.do?jsessionid=ec96dca5d7b38af1758bba07b265e0e0'. The left sidebar lists database objects: 'jdbc:h2:mem:testdb', 'TB_DEPARTMENT', 'TB_USER', 'INFORMATION_SCHEMA', 'Users', and the version 'H2 2.1.214 (2022-06-13)'. The main area contains a toolbar with various icons, including a play button highlighted with a red arrow. Below the toolbar is a SQL statement input field with the text 'SELECT * FROM TB_USER;'. The results section shows the same SQL statement and the message '(no rows, 1 ms)'.

```
SELECT * FROM TB_USER;
```

ID	EMAIL	NAME	DEPARTMENT_ID
(no rows, 1 ms)			

Edit



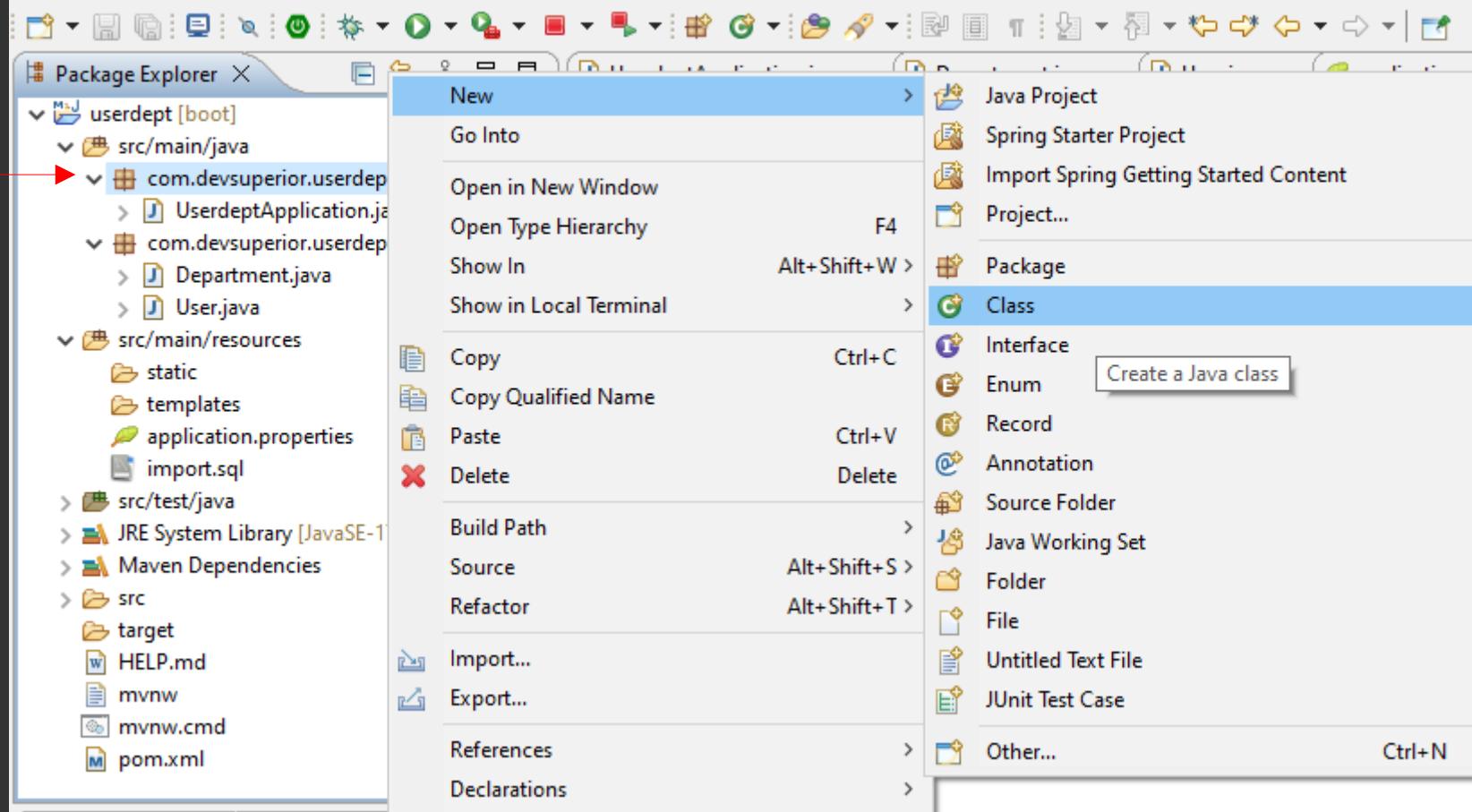
Selecione tb_department e de um run note que os dados da tabela foram inseridos

De um clear e repita para a tabela tb_user

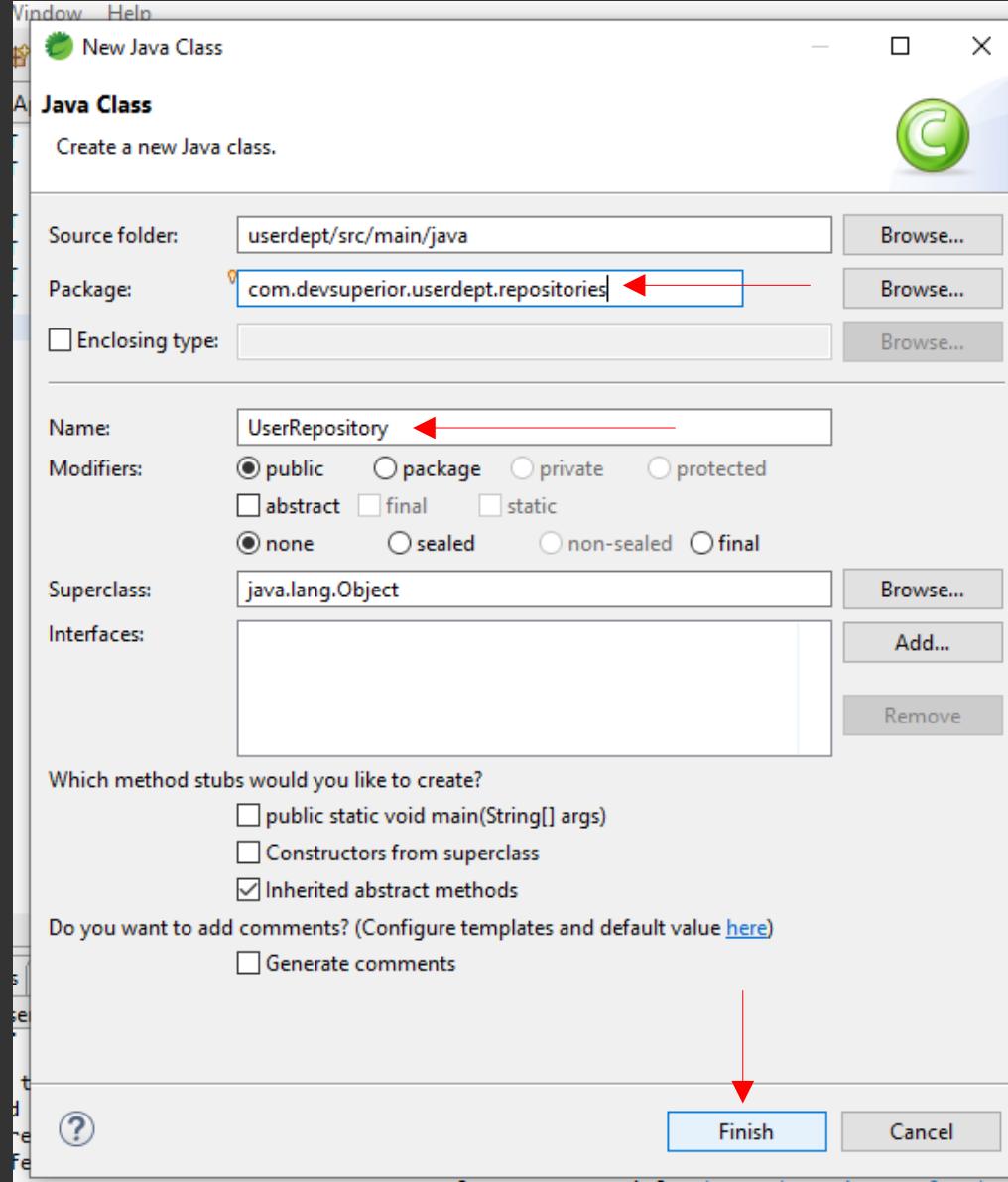
The screenshot shows the H2 Console interface. At the top, there are tabs for YouTube, GitHub, and H2 Console. Below the tabs, the URL is localhost:8080/h2-console/login.do?jsessionid=98195cdab91ac58a27b550c4a8b3b6d8. The main area has a toolbar with various icons and dropdowns for 'Auto commit' (checked), 'Max rows' (set to 1000), and 'SQL statement'. A red arrow points from the 'Clear' button in the toolbar to the SQL statement input field, which contains 'SELECT * FROM TB_USER;'. Another red arrow points from the result table back to the 'Clear' button. On the left, a sidebar lists database objects: 'jdbc:h2:mem:testdb', 'TB_DEPARTMENT', 'TB_USER', 'INFORMATION_SCHEMA', 'Users', and 'H2 2.1.214 (2022-06-13)'. The bottom half of the screen displays the results of the SQL query:

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

(4 rows, 0 ms)



Adicione ao final o .repositories



De nome a class

The screenshot shows the Spring Tool Suite 4 interface. The top bar displays the title "workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/repositories/UserRepository.java - Spring Tool Suite 4". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Find, and Run.

The left pane is the "Package Explorer" showing the project structure:

- userdept [boot]
 - src/main/java
 - com.devsuperior.userdept
 - UserdeptApplication.java
 - com.devsuperior.userdept.entities
 - Department.java
 - User.java
 - com.devsuperior.userdept.repositories
 - UserRepository.java
 - src/main/resources
 - static
 - templates
 - application.properties
 - import.sql
 - src/test/java
 - JRE System Library [JavaSE-17]
 - Maven Dependencies
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd

A red arrow points to the "UserRepository.java" file in the package explorer.

The right pane is the code editor with the tab "UserRepository.java" selected. The code shown is:

```
1 package com.devsuperior.userdept.repositories;
2
3 public class UserRepository {
4
5 }
```

Veja o pacote criado e sua class

The screenshot shows the Spring Tool Suite 4 interface. The left pane is the Package Explorer, displaying the project structure under 'userdept [boot]'. The right pane is the Java code editor with the file 'UserRepository.java' open. The code is as follows:

```
1 package com.devsuperior.serdept.repositories;
2
3 public interface UserRepository extends JpaRepository {
4
5 }
6
```

A red arrow points to the line 'public interface UserRepository extends JpaRepository {'. The code editor has tabs for 'Department.java', 'User.java', 'application.pro', 'import.sql', and '*UserRepository'. The title bar indicates the current file is 'UserRepository.java'.

**Apague o class e digite o code acima para extends depois
import a JpaRepository**

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/repositories/UserRepository.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Project Run Window Help

Package Explorer X Department.java User.java application.pro import.sql *UserRepository X »1

userdept [boot]

src/main/java

- com.devsuperior.userdept
 - UserdeptApplication.java
- com.devsuperior.userdept.entity
 - Department.java
 - User.java
- com.devsuperior.userdept.repositories
 - UserRepository.java

src/main/resources

- static
- templates
- application.properties
- import.sql

src/test/java

JRE System Library [JavaSE-17]

Maven Dependencies

```
1 package com.devsuperior.userdept.repositories;
2
3 public interface UserRepository extends JpaRepository {
```

...
package com.devsuperior.userdept.repositories;
import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository {
}

Press 'Tab' from proposal table or click for focus

- Import 'JpaRepository' (org.springframework.data.jpa.repository)
- Create interface 'JpaRepository'
- Rename in file (Ctrl+2, R)
- Fix project setup...

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/repositories/UserRepository.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Spring Tool Suite interface with the following details:

- Package Explorer:** Shows the project structure under the package `userdept [boot]`. The `src/main/java` folder contains:
 - `com.devsuperior.userdept`: Contains `UserDeptApplication.java`.
 - `com.devsuperior.userdept.entities`: Contains `Department.java` and `User.java`.
 - `com.devsuperior.userdept.repositories`: Contains `UserRepository.java`.
- Code Editor:** Displays the file `UserRepository.java` with the following code:

```
1 package com.devsuperior.userdept.repositories;
2
3 import org.springframework.data.jpa.repository.JpaRepository; ←
4
5 public interface UserRepository extends JpaRepository {
6
7 }
```

A red arrow points to the import statement `import org.springframework.data.jpa.repository.JpaRepository;`.
- Toolbars:** Standard Eclipse-style toolbars for file operations, search, and navigation.
- Right-hand pane:** Shows tabs for other files: `Department.java`, `User.java`, `application.pro`, `import.sql`, and `*UserRepository`.

Acrecente o code e depois import

The screenshot shows the Spring Tool Suite 4 interface. The title bar reads "workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/repositories/UserRepository.java - Spring Tool Suite 4". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar has various icons for file operations like Open, Save, Copy, Paste, and Find. The Package Explorer view on the left shows the project structure under "userdept [boot]". The UserRepository.java file is open in the editor, displaying the following code:

```
1 package com.devsuperior.userdept.repositories;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 public interface UserRepository extends JpaRepository<User, Long>{
6
7 }
```

Red arrows point from the text "User" and "Long" in the code back to the "User.java" and "application.properties" files in the package explorer, indicating they are being imported or used.

The screenshot shows the Spring Tool Suite 4 interface with a Java code editor open. The code being edited is:

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

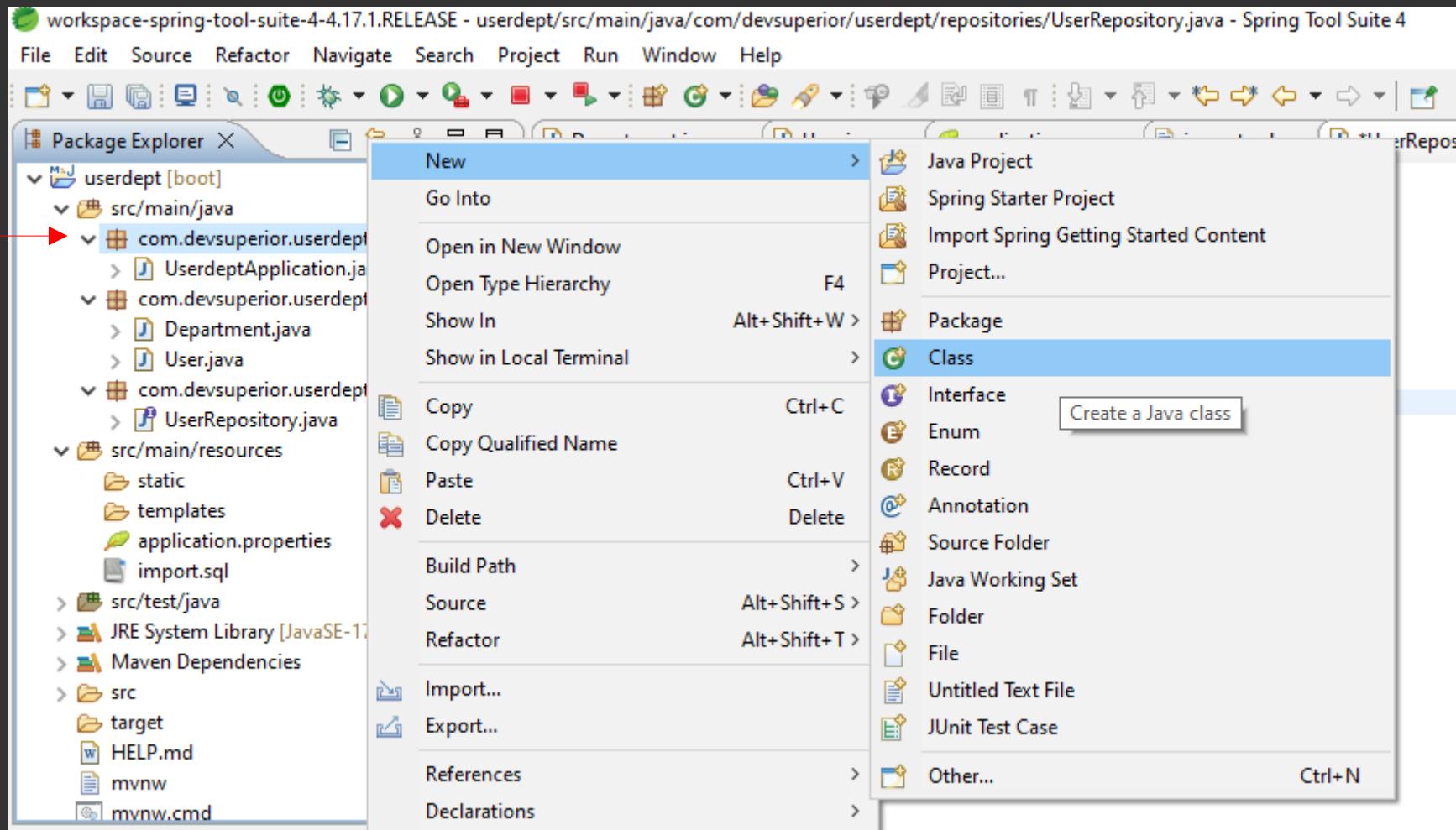
A code completion dropdown menu is displayed on the right side of the editor, listing several suggestions:

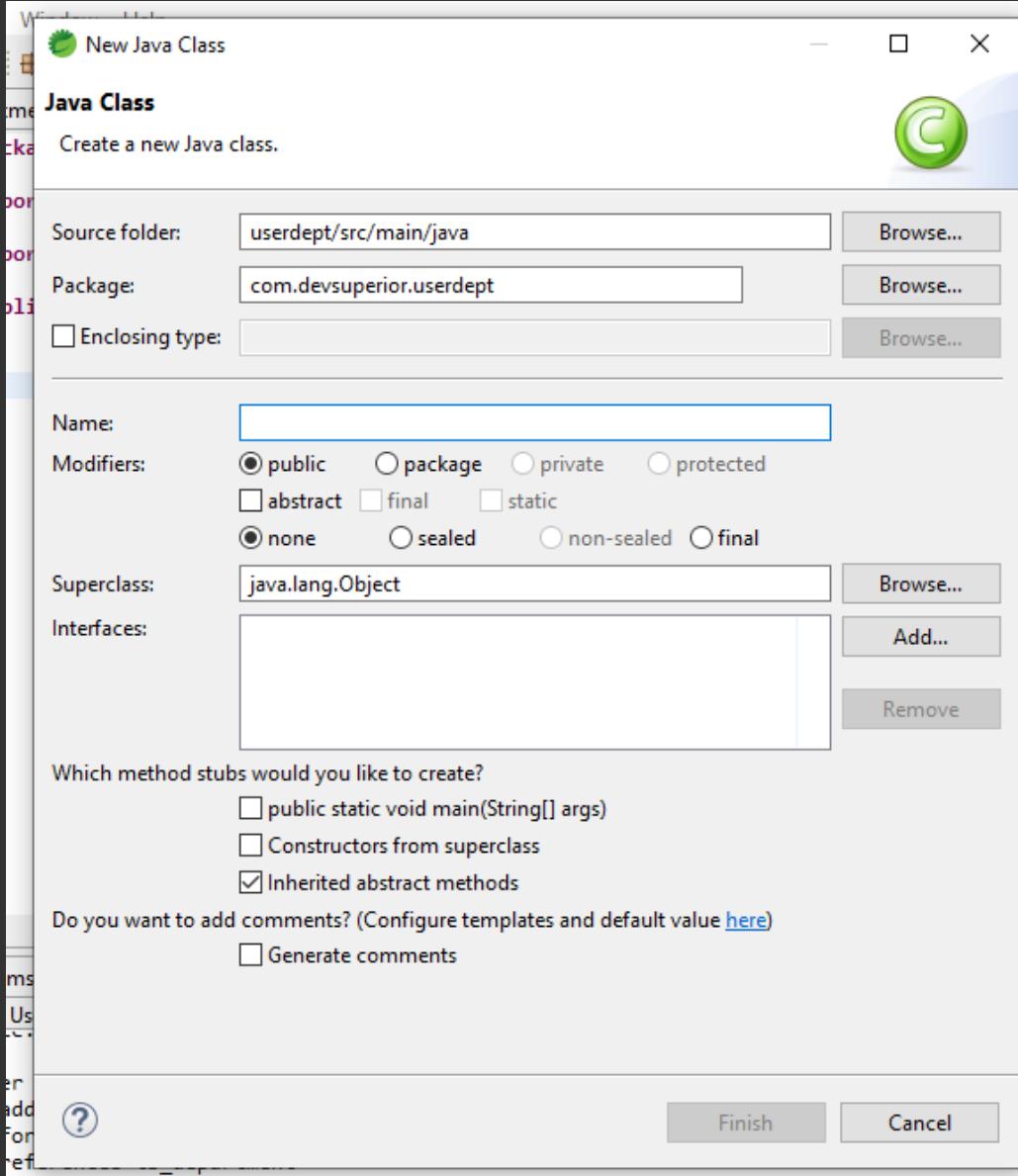
- ← Import 'User' (com.devsuperior.userdept.entities)
- ← Import 'User' (org.apache.catalina)
- ← Import 'User' (org.springframework.boot.autoconfigure.security)
- C Create class 'User'
- R Create record 'User'
- I Create interface 'User'
- G Change to 'UserType' (org.hibernate.usertype)
- G Change to 'Uses' (org.aspectj.apache.bcel.classfile.Module)
- E Create enum 'User'
- Add type parameter 'User' to 'UserRepository'

At the bottom of the dropdown, there is a note: "Press 'Tab' from proposal table or click for focus".

The screenshot shows the Spring Tool Suite 4 interface. The left side features the Package Explorer view, which displays the project structure of 'userdept'. The 'src/main/java' folder contains packages like 'com.devsuperior.userdept', 'com.devsuperior.userdept.entities', and 'com.devsuperior.userdept.repositories'. Under 'com.devsuperior.userdept.repositories', the file 'UserRepository.java' is selected and shown in the central code editor. The code editor has tabs for 'Department.java', 'User.java', 'application.pro', 'import.sql', and '*UserRepository'. The code itself is:

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





File Edit Source Refactor Navigate Search Project Run Window Help



userdept [boot]
src/main/java
com.devsuperior.userdept
UserdeptApplication.java
com.devsuperior.userdept.entities
Department.java
User.java
com.devsuperior.userdept.repositories
UserRepository.java
src/main/resources
static
templates
application.properties
import.sql
src/test/java
JRE System Library [JavaSE-17]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd



Type tags, projects, or working set names to match (i)

local
userdept [8080]

New Java Class

Java Class

Create a new Java class.

Source folder: userdept/src/main/java

Package: com.devsuperior.userdept.controllers

Enclosing type:

Name: UserController

Modifiers:

public package private protected
 abstract final static
 none sealed non-sealed final

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

public static void main(String[] args)
 Constructors from superclass
 Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

Generate comments

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/controllers/UserController.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Project Run Window Help

Package Explorer User.java application.pro import.sql *UserController UserController.java

```
1 package com.devsuperior.userdept.controllers;
2
3 public class UserController {
4
5 }
6
```

The screenshot shows the Spring Tool Suite 4 interface. The title bar reads "workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/controllers/UserController.java - Spring Tool Suite 4". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar has various icons for file operations like Open, Save, Find, and Run. The Package Explorer view on the left shows a project structure under "userdept [boot]". The "src/main/java" folder contains packages like "com.devsuperior.userdept", "com.devsuperior.userdept.controllers", "com.devsuperior.userdept.entities", and "com.devsuperior.userdept.repositories". Inside "com.devsuperior.userdept.controllers", there is a "UserController.java" file. The code editor on the right displays the following Java code:

```
1 package com.devsuperior.userdept.controllers;
2
3 @RestController ←
4 public class UserController {
5
6 }
7
```

Crie a anotação e depois importe para virar um controlador

The screenshot shows the Spring Tool Suite 4 interface with the following details:

- Project Explorer:** Shows the project structure under "userdept".
- Code Editor:** Displays the file `UserController.java` containing the following code:

```
1 package com.devsuperior.userdept.controllers;
2
3 @RestController
4 public class UserController { ... }
```
- Code Completion Pop-up:** A tooltip is open over the `@RestController` annotation, listing the following options:
 - Import 'RestController' (org.springframework.web.bind.annotation)
 - Create annotation 'RestController'
 - Change to 'Controller' (org.springframework.stereotype)
 - Change to 'RestControllerAdvice' (org.springframework.web.bind.a)
 - Rename in file (Ctrl+2, R)
 - Fix project setup...
- Output View:** Shows the message "Press 'Tab' from proposal table or click for focus".

The screenshot shows the Spring Tool Suite 4 interface. The left pane is the Package Explorer, displaying the project structure for 'userdept [boot]'. The 'src/main/java' folder contains packages: 'com.devsuperior.userdept' (with 'UserDeptApplication.java'), 'com.devsuperior.userdept.controllers' (with 'UserController.java' selected), 'com.devsuperior.userdept.entities' (with 'Department.java' and 'User.java'), 'com.devsuperior.userdept.repositories' (with 'UserRepository.java'), and 'src/main/resources' (with 'static', 'templates', 'application.properties', and 'import.sql'). The right pane is the code editor for 'UserController.java', showing the following code:

```
1 package com.devsuperior.userdept.controllers;
2
3 import org.springframework.web.bind.annotation.RestController;
4
5 @RestController
6 public class UserController {
7
8 }
9
```

The screenshot shows the Spring Tool Suite interface. On the left is the Package Explorer view, displaying the project structure under 'userdept [boot]'. The 'src/main/java' folder contains packages like 'com.devsuperior.userdept', 'com.devsuperior.userdept.controllers', 'com.devsuperior.userdept.entities', and 'com.devsuperior.userdept.repositories'. It also includes 'User.java', 'UserRepository.java', and 'UserController.java'. The 'src/main/resources' folder contains 'application.properties', 'import.sql', and static resources. Other sections include 'src/test/java', 'JRE System Library [JavaSE-17]', 'Maven Dependencies', 'src', and 'target'. The central area is a code editor for 'UserController.java'. The code is:

```
1 package com.devsuperior.userdept.controllers;
2
3 import org.springframework.web.bind.annotation.RestController;
4
5 @RestController
6 @RequestMapping
7 public class UserController {
8
9 }
10
```

A green annotation `@RequestMapping` is highlighted. Below the code editor, a large green text overlay reads: "Igualmente crie a anotation e seu import".

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/controllers/UserController.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X User.java application.pro import.sql UserRepository.*UserController X »2

```
1 package com.devsuperior.userdept.controllers;
2
3 import org.springframework.web.bind.annotation.RequestMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 @RequestMapping
8 public class UserController {
9
10 }
11
```

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/controllers/UserController.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Spring Tool Suite 4 interface. On the left is the Package Explorer view, displaying the project structure under 'userdept [boot]'. The 'src/main/java' folder contains several packages: 'com.devsuperior.userdept' (with 'UserApplication.java'), 'com.devsuperior.userdept.controllers' (with 'UserController.java'), 'com.devsuperior.userdept.entities' (with 'Department.java' and 'User.java'), and 'com.devsuperior.userdept.repositories' (with 'UserRepository.java'). The 'src/main/resources' folder contains 'static', 'templates', 'application.properties', and 'import.sql'. The right side of the screen shows the code editor for 'UserController.java'. The code is as follows:

```
1 package com.devsuperior.userdept.controllers;
2
3 import org.springframework.web.bind.annotation.RequestMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 @RequestMapping(value = "/users")
8 public class UserController {
9
10 }
```

A red arrow points from the bottom of the image to the '@RequestMapping(value = "/users")' annotation in the code.

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/controllers/UserController.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X User.java application.pro import.sql UserRepository.*UserController X »₂

```
1 package com.devsuperior.userdept.controllers;
2
3 import org.springframework.web.bind.annotation.RequestMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 @RequestMapping(value = "/users")
8 public class UserController {
9
10    public List<User> findAll(){
11
12    }
13
14 }
```

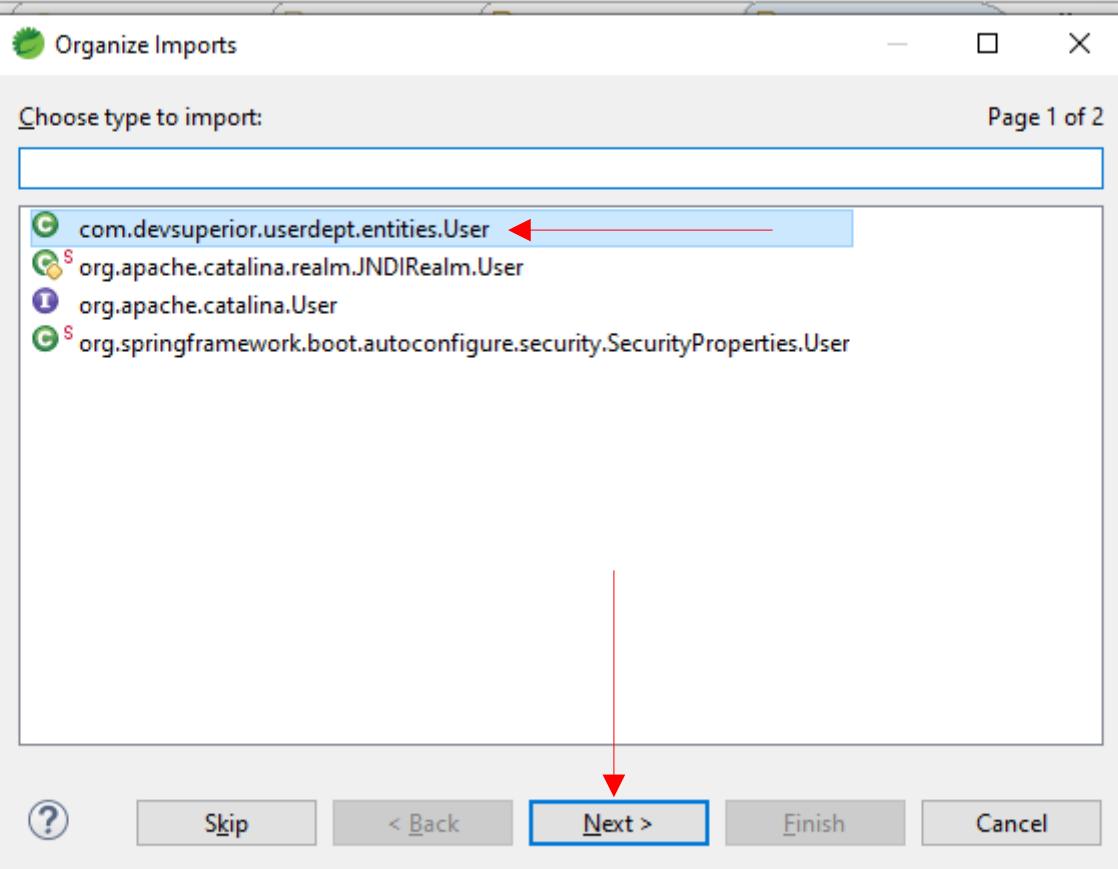
Deixe o cursor ai e tecle ctrl + shift + o



o
Controllers
Services
Repositories

User.java

```
1 package
2
3 import
4 import
5
6 @RestController
7 @Request
8 public
9
10 public
11
12 }
13 }
14 }
```

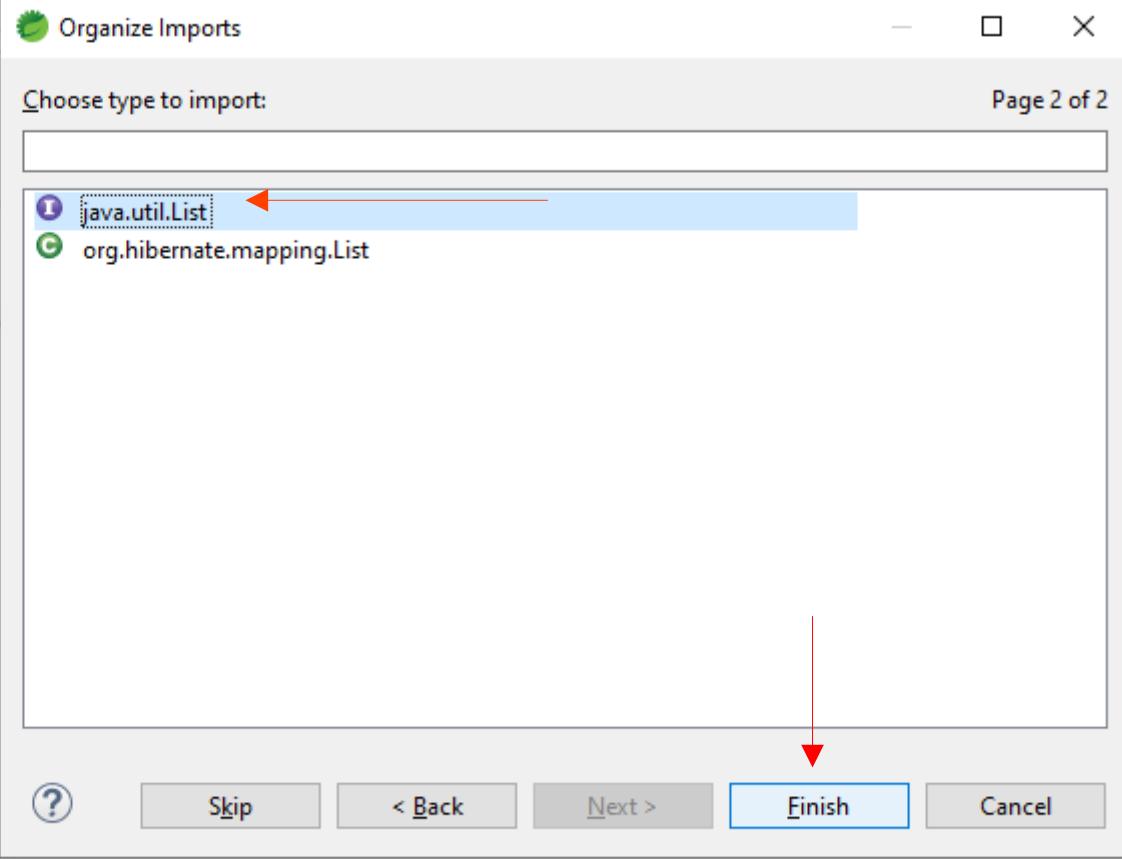




```
src  
  controllers  
  services  
  repositories
```

User.java

```
1 package com.example.  
2  
3 import java.util.List;  
4 import org.hibernate.mapping.List;  
5  
6 @RestController  
7 @RequestMapping("/users")  
8 public class UserController {  
9  
10    @GetMapping  
11    public List<User> getAllUsers() {  
12        return userRepository.findAll();  
13    }  
14}
```



The screenshot shows the Spring Tool Suite 4 interface. The left pane is the Package Explorer, displaying the project structure:

- userdept [boot]
 - src/main/java
 - com.devsuperior.userdept
 - UserDeptApplication.java
 - com.devsuperior.userdept.controllers
 - UserController.java
 - com.devsuperior.userdept.entities
 - Department.java
 - User.java
 - com.devsuperior.userdept.repositories
 - UserRepository.java
 - src/main/resources
 - static
 - templates
 - application.properties
 - import.sql
 - src/test/java
- JRE System Library [JavaSE-17]
- Maven Dependencies
- src
- target
- HELP

The right pane shows the UserController.java code editor. The code is as follows:

```
1 package com.devsuperior.userdept.controllers;
2
3 import java.util.List; ←
4
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RestController;
7
8 import com.devsuperior.userdept.entities.User;
9
10 @RestController
11 @RequestMapping(value = "/users")
12 public class UserController {
13
14     public List<User> findAll(){
15         |
16     }
17 }
18
```

A red arrow points to the import statement for List, indicating a potential issue or highlighting it.

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/controllers/UserController.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X User.java application.pro import.sql UserRepository.*UserController X

```
1 package com.devsuperior.userdept.controllers;
2
3 import java.util.List;
4
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RestController;
7
8 import com.devsuperior.userdept.entities.User;
9
10 @RestController
11 @RequestMapping(value = "/users")
12 public class UserController {
13
14     private UserRepository repository; ←
15
16     public List<User> findAll(){
17
18     }
19
20 }
```

Adicione a linha e depois seu import

The screenshot shows the Spring Tool Suite 4 interface with the following details:

- Project Explorer:** Shows the project structure under "userdept [boot]".
- Code Editor:** Displays the `UserController.java` file content:

```
1 package com.devsuperior.userdept.controllers;
2
3 import java.util.List;
4
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RestController;
7
8 import com.devsuperior.userdept.entities.User;
9
10 @RestController
11 @RequestMapping(value = "/users")
12 public class UserController {
13
14     private UserRepository repository;
15
16     public Li...}
```
- Code Completion Pop-up:** A tooltip is open at the cursor position, listing options for the `repository` field:
 - Import 'UserRepository' (com.devsuperior.userdept.repositories)
 - Create class 'UserRepository'
 - Create record 'UserRepository'
 - Create interface 'UserRepository'
 - Change to 'CrudRepository' (org.springframework.data.repository)
 - Change to 'DefaultLoaderRepository' (javax.management)
 - Change to 'DefaultLoaderRepository' (javax.management.loadable)
 - Change to 'ListCrudRepository' (org.springframework.data.repository)
 - Create enum 'UserRepository'
 - Add type parameter 'UserRepository' to 'UserController'
 - Rename in file (Ctrl+Shift R)
- Outline View:** Shows the class structure with `UserController` expanded, showing its methods: `repository` and `findAll()`.
- Bottom Status Bar:** Shows tabs for Problems, JavaDoc, Declaration, and Console.

```
8 □ □ User.java application.pro import.sql UserRepository.* UserController »2
  ^

  controllers
  ties
  ositories

  1 package com.devsuperior.userdept.controllers;
  2
  3 import java.util.List;
  4
  5 import org.springframework.web.bind.annotation.RequestMapping;
  6 import org.springframework.web.bind.annotation.RestController;
  7
  8 import com.devsuperior.userdept.entities.User;
  9 import com.devsuperior.userdept.repositories.UserRepository;
 10
 11 @RestController
 12 @RequestMapping(value = "/users")
 13 public class UserController {
 14
 15     private UserRepository repository;
 16
 17     public List<User> findAll(){
 18
 19     }
 20
 21 }
```

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/controllers/UserController.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X User.java application.pro import.sql UserRepository.*UserController X >2

```
1 package com.devsuperior.userdept.controllers;
2
3 import java.util.List;
4
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RestController;
7
8 import com.devsuperior.userdept.entities.User;
9 import com.devsuperior.userdept.repositories.UserRepository;
10
11 @RestController
12 @RequestMapping(value = "/users")
13 public class UserController {
14
15     @Autowired ←
16     private UserRepository repository;
17
18     public List<User> findAll(){
19
20     }
21 }
22 }
```

Adicione a anotação e seu import

```
Stories 12 @RequestMapping(value = "/users")
13 public class UserController {
14
15     @Autowired
16     private ...
17
18     @PutMapping
19     public void updateUser(@RequestBody User user) {
20         ...
21     }
22 }
```

The screenshot shows an IDE interface with a code editor displaying Java code. A tooltip is open over the `@Autowired` annotation at line 15. The tooltip contains the following information:

- ← Import 'Autowired' (org.springframework.beans.factory.annotation)
- >Create annotation 'Autowired'
- Rename in file (Ctrl+2, R)
- Fix project setup...

The code editor shows imports for `java.util.List`, `org.springframework.beans.factory.annotation.Autowired`, `org.springframework.web.bind.annotation.RequestMapping`, and `org.springframework.web.bind.annotation.RestController`. The code itself defines a `UserController` with a `updateUser` method.

```
1 package com.devsuperior.userdept.controllers;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.web.bind.annotation.RequestMapping;
7 import org.springframework.web.bind.annotation.RestController;
8
9 import com.devsuperior.userdept.entities.User;
10 import com.devsuperior.userdept.repositories.UserRepository;
11
12 @RestController
13 @RequestMapping(value = "/users")
14 public class UserController {
15
16     @Autowired
17     private UserRepository repository;
18
19     public List<User> findAll(){
20
21     }
22
23 }
```

Termine o code

The screenshot shows the Spring Tool Suite 4 interface. The Package Explorer view on the left displays the project structure under the package `userdept`. The UserController.java file is open in the editor on the right. The code implements a REST controller for users, using annotations from the Spring framework. A red arrow points to the `@GetMapping` annotation on line 19, indicating where the user needs to add the import statement.

```
1 package com.devsuperior.userdept.controllers;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.web.bind.annotation.RequestMapping;
7 import org.springframework.web.bind.annotation.RestController;
8
9 import com.devsuperior.userdept.entities.User;
10 import com.devsuperior.userdept.repositories.UserRepository;
11
12 @RestController
13 @RequestMapping(value = "/users")
14 public class UserController {
15
16     @Autowired
17     private UserRepository repository;
18
19     @GetMapping
20     public List<User> findAll(){
21         List<User> result = repository.findAll();
22         return result;
23     }
24 }
25
```

Adicione a anotação e seu import

```
16@Autowired  
17 private UserRepository repository;  
18  
19 @GetMapping  
20 public ...  
21     ^-- Import 'GetMapping' (org.springframework.web.bind.annotation)  
22     @ Create annotation 'GetMapping'  
23 }  
24     ^-- Change to 'DeleteMapping' (org.springframework.web.bind.annotation)  
25     ^-- Change to 'Mapping' (org.springframework.web.bind.annotation)  
     ^-- Change to 'PatchMapping' (org.springframework.web.bind.annotation)  
     ^-- Change to 'PostMapping' (org.springframework.web.bind.annotation)  
     ^-- Change to 'PutMapping' (org.springframework.web.bind.annotation)  
     ^-- Change to 'RequestMapping' (org.springframework.web.bind.annotation)  
     ^-- Rename in file (Ctrl+2, R)  
     ^-- Fix imports
```

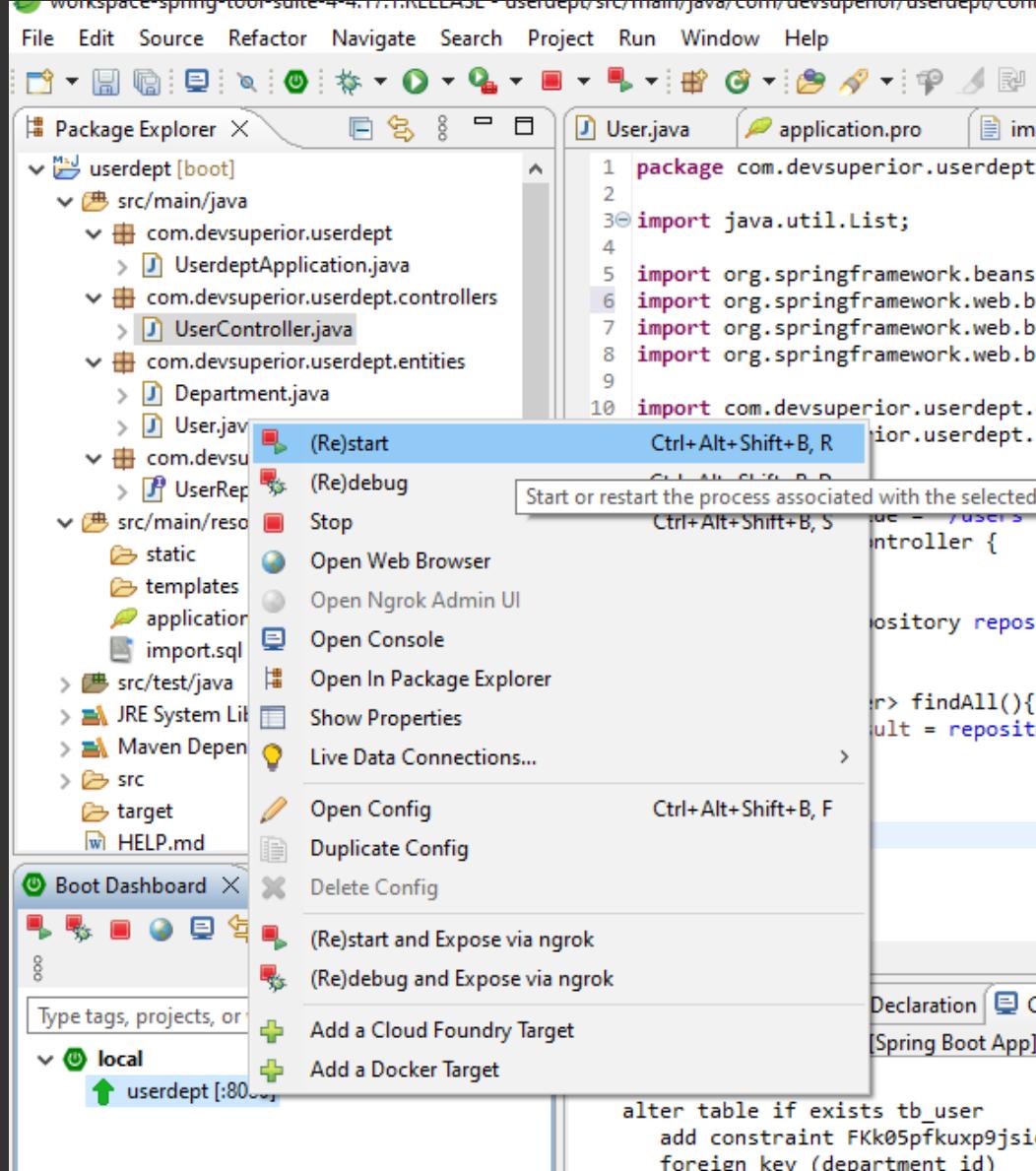
...
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
...

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/userdept/controllers/UserController.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X User.java application.pro import.sql UserRepository.*UserController X 2

```
1 package com.devsuperior.userdept.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.devsuperior.userdept.entities.User;
11 import com.devsuperior.userdept.repositories.UserRepository;
12
13 @RestController
14 @RequestMapping(value = "/users")
15 public class UserController {
16
17     @Autowired
18     private UserRepository repository;
19
20     @GetMapping
21     public List<User> findAll(){
22         List<User> result = repository.findAll();
23         return result;
24     }
25
26 }
```

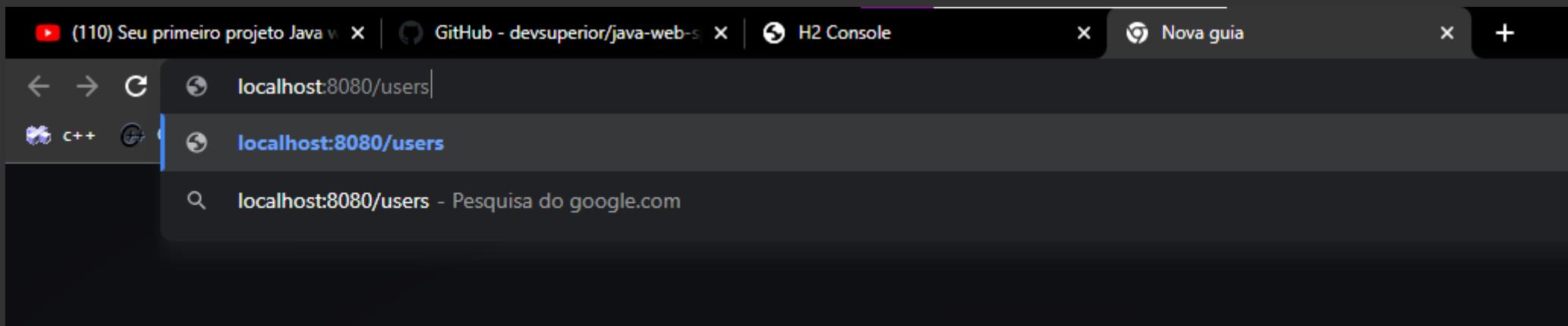


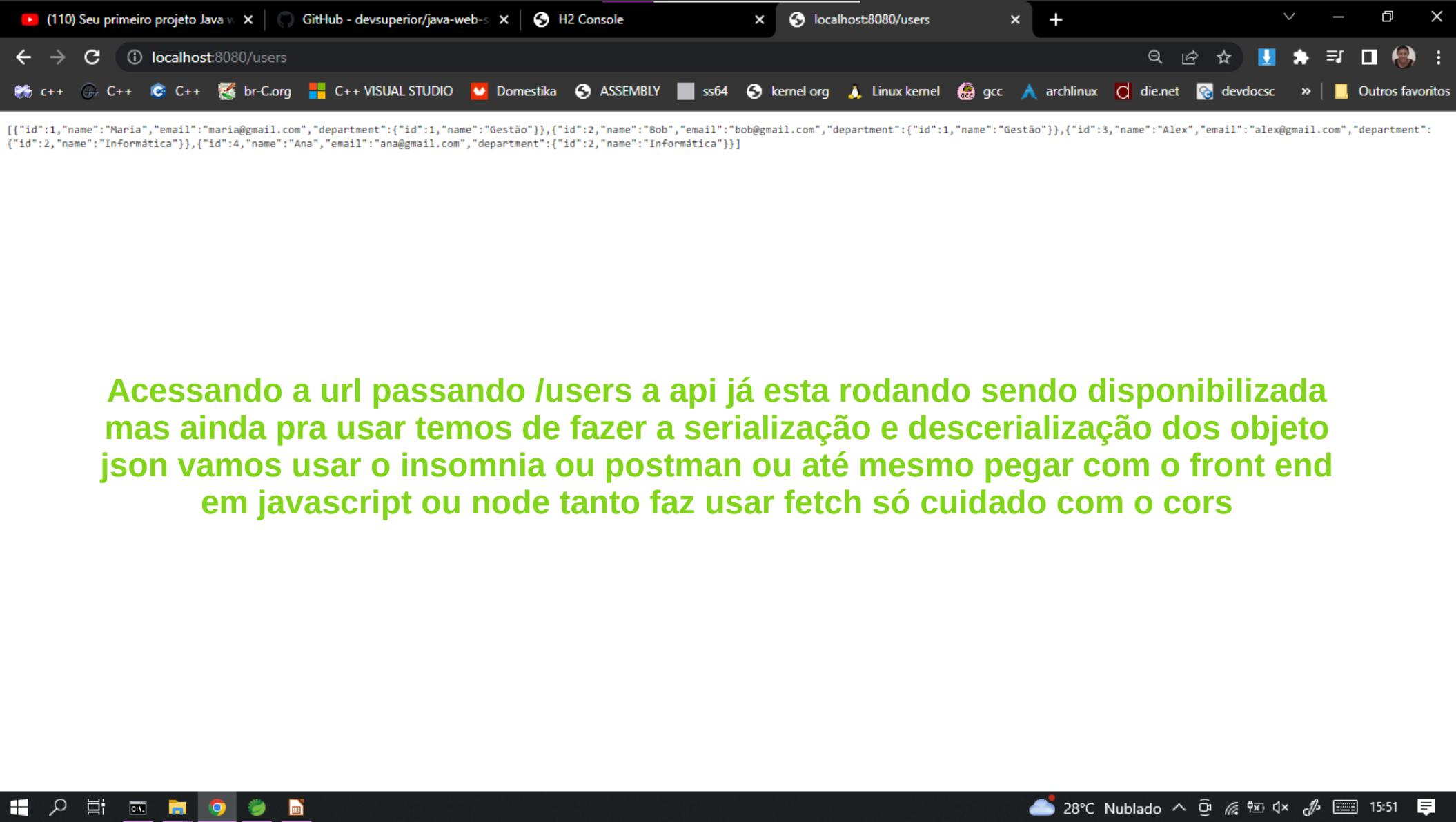
17 @Autowired

Problems @ Javadoc Declaration Console

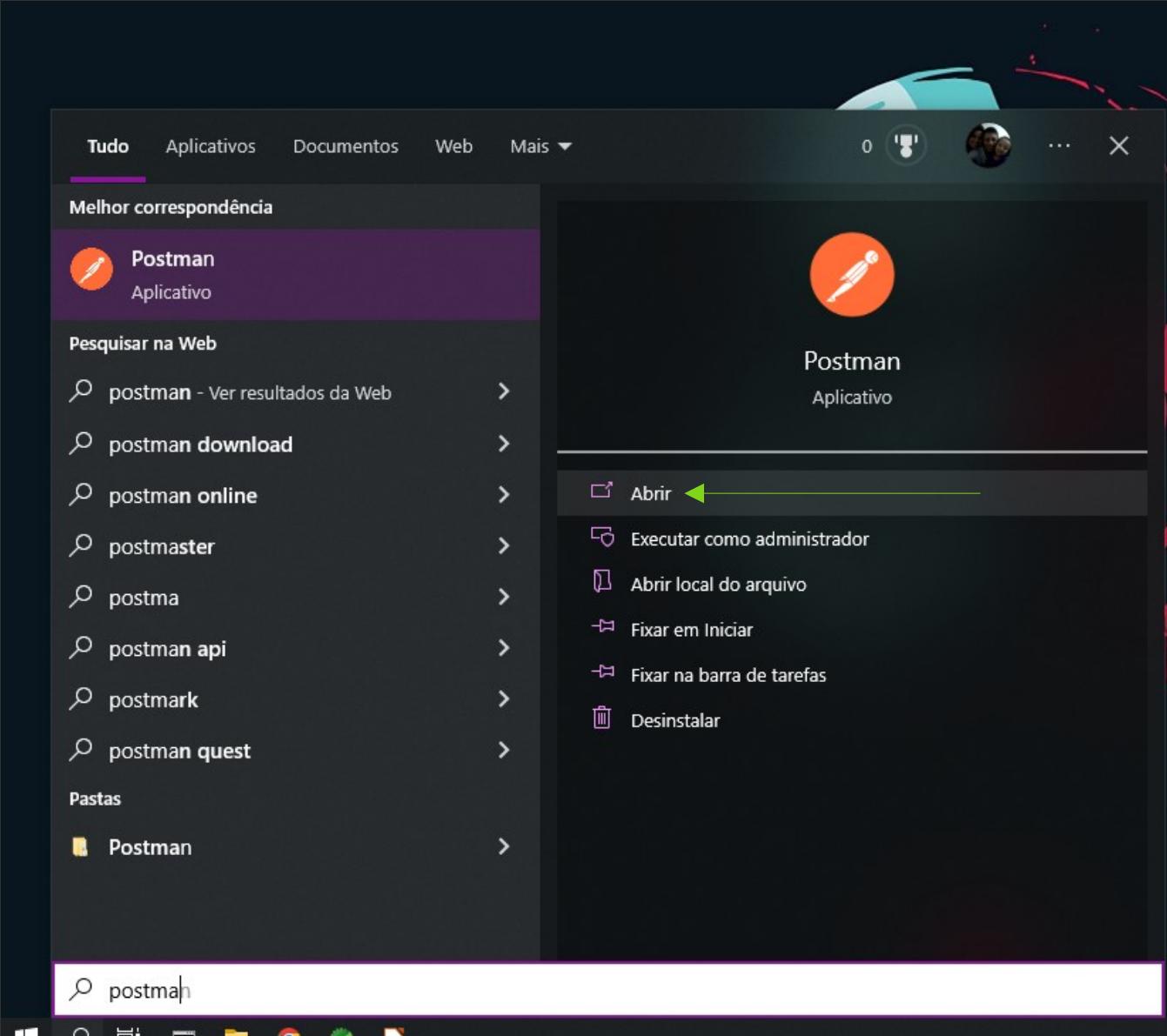
userdept - UserdeptApplication [Spring Boot App] C:\Users\brito\Downloads\spring\sts-4.17.1.RELEASE\plugins\org.eclipse.jdt.core\src\main\java\c\d\userdept\UserdeptApplication.java

```
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')
2023-01-20T15:50:35.532-03:00  INFO 7688 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiat
2023-01-20T15:50:35.550-03:00  INFO 7688 --- [           main] j.LocalContainerEntityManagerFac
2023-01-20T15:50:35.945-03:00  WARN 7688 --- [           main] JpaBaseConfiguration$JpaWebConfig
2023-01-20T15:50:36.443-03:00  INFO 7688 --- [           main] o.s.b.w.embedded.tomcat.TomcatWe
2023-01-20T15:50:36.458-03:00  INFO 7688 --- [           main] c.d.userdept.UserdeptApplication
```





Acessando a url passando /users a api já esta rodando sendo disponibilizada mas ainda pra usar temos de fazer a serialização e deserialização dos objeto json vamos usar o insomnia ou postman ou até mesmo pegar com o front end em javascript ou node tanto faz usar fetch só cuidado com o cors



My Workspace

[New](#) [Import](#)

GET http://localhost:3000/c



...

No Environment

[Collections](#)[+](#) [≡](#)

...

[Create new Collection](#)

My first collection

- First folder inside collection
 - GET
 - POST
 - GET
- Second folder inside collection
 - GET
 - GET



Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

[Create collection](#)

Request not found

We could not find the request you are looking for.

click

My Workspace

New

Import

GET http://loca...



Collections



> New Collection



...



APIs



Environments



Mock Servers



Monitors



Flows



History

- Share
- Move
- Run collection
- Edit
- Add request
- Add folder
- Monitor collection
- Mock collection
- Create a fork
- Create pull request
- Merge changes
- View changelog
- View documentation
- Rename Ctrl+E
- Duplicate Ctrl+D
- Export
- Manage roles

New Collec...

Authorization

This authori...
override thi...

Type

This co...
Learn m...

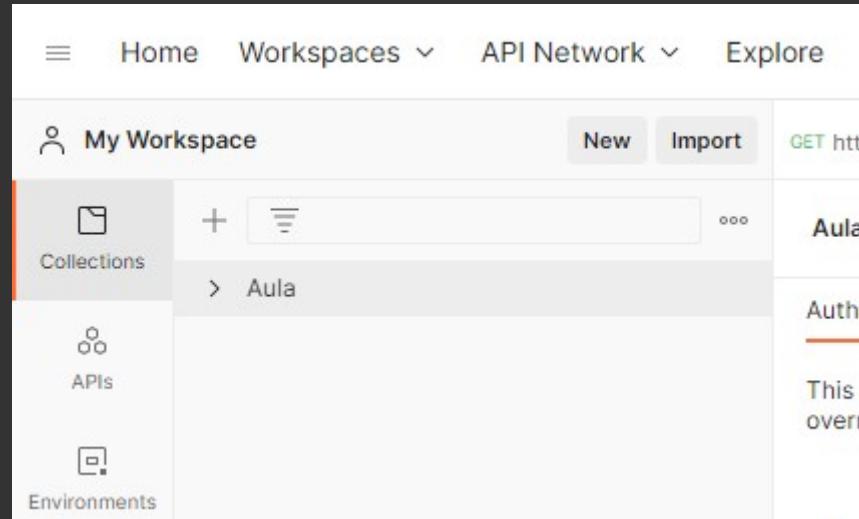
New Collec...

Authorization

This authori...
override thi...

Type

This co...
Learn m...



My Workspace

New Import

GET http://localhost:3000/c



Aula

+ ...

Collections

+ ⚡ ...
Aula

Share

Move

Run collection

Edit

Add request

Add folder

Monitor collection

Mock collection

Create a fork

Create pull request

Merge changes

View changelog

View documentation

Rename Ctrl+E

Duplicate Ctrl+D

Export

Manage roles



Aula

Authorization

Pre-request Script

Tests

Variables

This authorization method will be used for every request in this collection. Y

Type

No Auth

This collection does not use any authorization. Learn more about auth

Click direito

The screenshot shows the Postman application interface. At the top, there is a navigation bar with 'Home', 'Workspaces', 'API Network', and 'Explore' buttons. Below the navigation bar, the 'My Workspace' section is visible. On the left side, there is a sidebar with 'Collections' (selected), 'APIs', and 'Environments'. Under 'Collections', there is a section for 'Aula' which contains a 'New Request' button labeled 'GET New Request'. To the right of the sidebar, there is a search bar with the text 'Users' and a red arrow pointing to it. Below the search bar, there is a 'GET' method dropdown and a 'Enter request URL' input field. Further down, there are tabs for 'Params' (selected), 'Authorization', and 'Query Params'.

≡ Home Workspaces API Network Explore

Search Postman

Invite

My Workspace

New Import

GET http://localhost:3000/c Aula

GET Users

Aula / Users

Collections

+ Aula

APIs

GET Users

Environments

Mock Servers

Monitors

+

...

GET

Enter request URL

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

Query Params

	KEY	VALUE
	Key	Value

My Workspace New Import GET http://localhost:3000/c Aula GET Users + ... No Environment

Collections APIs Environments Mock Servers Monitors Flows History

Aula / Users

GET http://localhost:8080/users

Send

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

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Response

Copy a url e depois send

Click Send to get a response



My Workspace New Import GET http://localhost:3000/c Aula GET Users + ... No Environment

Aula / Users

GET http://localhost:8080/users

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

Query Params

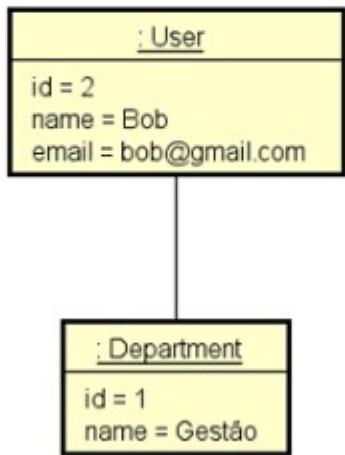
KEY	VALUE	DESCRIPTION	... Bulk Edit
Key	Value	Description	

Body Cookies Headers (5) Test Results Status: 200 OK Time: 173 ms Size: 521 B Save Response

Pretty Raw Preview Visualize JSON

```
2 |   {
3 |     "id": 1,
4 |     "name": "Maria",
5 |     "email": "maria@gmail.com",
6 |     "department": {
7 |       "id": 1,
8 |       "name": "Gestão"
9 |     }
10 |   },
11 |   {
12 |     "id": 2,
13 |     "name": "Bob",
14 |     "email": "bob@gmail.com",
```

O metodo get
retorna os
dados em json
isso comprova
que a api esta
rodando



copy

The screenshot shows the Spring Tool Suite interface with the following details:

- Project Explorer:** Shows the project structure under "userdept [boot]".
 - src/main/java:** Contains:
 - com.devsuperior.userdept: UserdeptApplication.java
 - com.devsuperior.userdept.controllers: UserController.java
 - com.devsuperior.userdept.entities: Department.java, User.java
 - com.devsuperior.userdept.repositories: UserRepository.java
 - src/main/resources:** Contains:
 - static
 - templates
 - application.properties
 - import.sql
 - src/test/java**
 - JRE System Library [JavaSE-17]**
 - Maven Dependencies**
 - src**
 - target**
 - HELP.md**
- UserController.java:** The code is displayed in the main editor window.

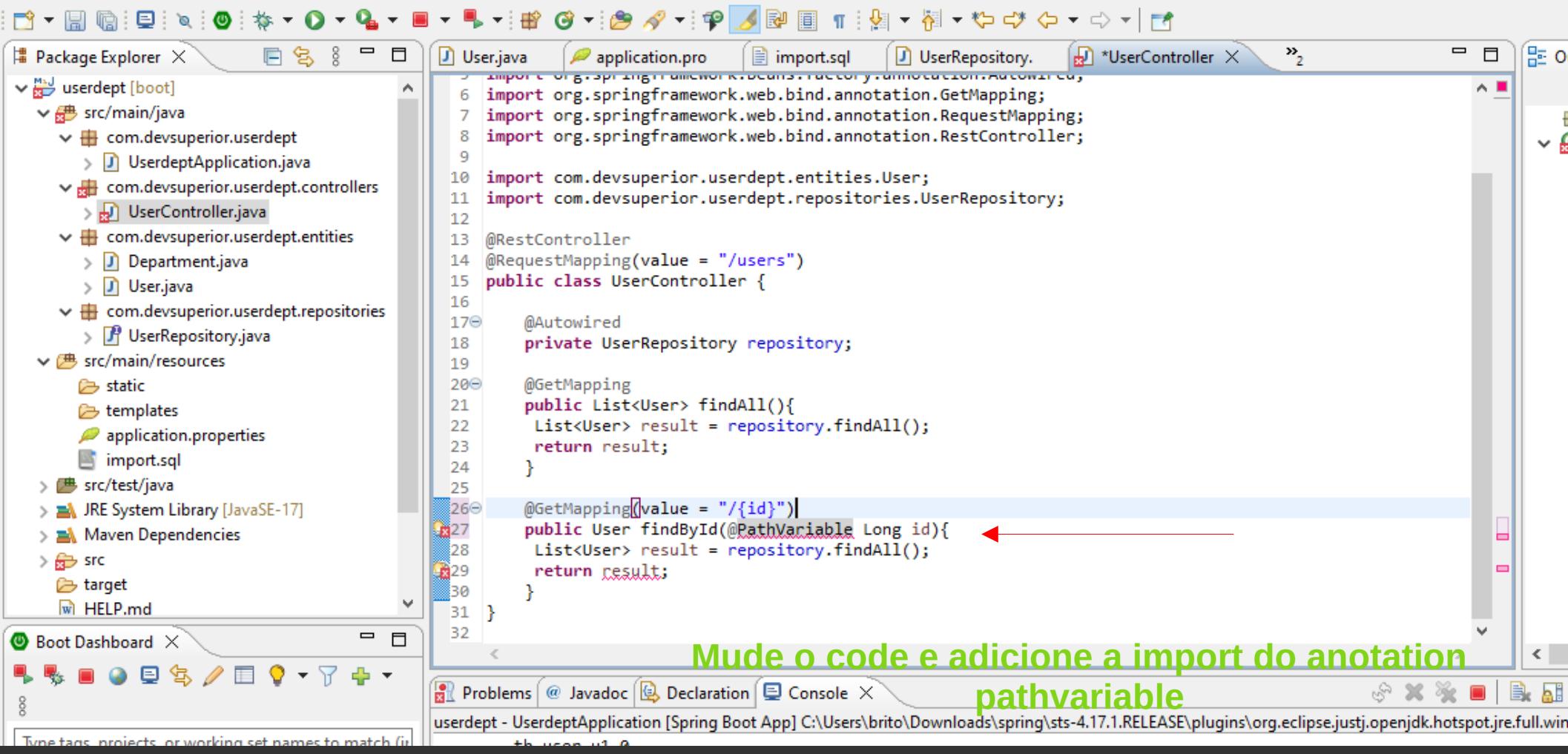
```
1 package com.devsuperior.userdept.controllers;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.web.bind.annotation.GetMapping;
7 import org.springframework.web.bind.annotation.RequestMapping;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import com.devsuperior.userdept.entities.User;
11 import com.devsuperior.userdept.repositories.UserRepository;
12
13 @RestController
14 @RequestMapping(value = "/users")
15 public class UserController {
16
17     @Autowired
18     private UserRepository repository;
19
20     @GetMapping
21     public List<User> findAll(){
22         List<User> result = repository.findAll();
23         return result;
24     }
25 }
26
```

A blue selection bar highlights the code from line 20 to line 25. A red arrow points to the start of the "findAll()" method definition.
- Toolbars and Status Bar:** Standard Eclipse-style toolbars and a status bar at the bottom.

The screenshot shows the Spring Tool Suite 4 interface with the following details:

- Project Explorer:** Shows the project structure under "userdept [boot]".
- Code Editor:** Displays the `UserController.java` file. The cursor is at the end of the first `findAll()` method definition. A red arrow points from the word `repository` to the `repository.findAll()` call in the second `findAll()` method definition.
- Status Bar:** Shows the message "paste" in green text, indicating a paste operation is in progress.
- Bottom Navigation:** Includes tabs for Problems, Javadoc, Declaration, and Console.

```
1 import org.springframework.beans.factory.annotation.Autowired;
2 import org.springframework.web.bind.annotation.GetMapping;
3 import org.springframework.web.bind.annotation.RequestMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 import com.devsuperior.userdept.entities.User;
7 import com.devsuperior.userdept.repositories.UserRepository;
8
9
10 @RestController
11 @RequestMapping(value = "/users")
12 public class UserController {
13
14     @Autowired
15     private UserRepository repository;
16
17     @GetMapping
18     public List<User> findAll(){
19         List<User> result = repository.findAll();
20         return result;
21     }
22
23     @GetMapping
24     public List<User> findAll(){█
25         List<User> result = repository.findAll();
26         return result;
27     }
28
29     }
30
31 }
```



The screenshot shows the Spring Tool Suite 4 interface with the following details:

- Project Explorer:** Shows the project structure under "userdept".
 - src/main/java:** Contains "UserdeptApplication.java", "UserController.java", "Department.java", "User.java", and "UserRepository.java".
 - src/main/resources:** Contains "application.properties", "import.sql", and "static".
- User.java (Controller):** The code defines a REST controller for users. It includes imports for Spring annotations and repository. The controller has two methods: `findAll()` and `findById()`. The `findById()` method is annotated with `@GetMapping(value = "/{id}")` and `@PathVariable Long id`.
- Annotations:** A red arrow points to the `@PathVariable` annotation in the `findById` method.
- Bottom Status Bar:** Shows the current file is "UserController.java" and the path is "userdept - UserdeptApplication [Spring Boot App] C:\Users\brito\Downloads\spring\sts-4.17.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\lib\main\UserController.java".
- Text at Bottom:** "Mude o code e adicione a import do anotation pathvariable"

Package Explorer X

- userdept [boot]
 - src/main/java
 - com.devsuperior.userdept
 - UserDeptApplication.java
 - com.devsuperior.userdept.controllers
 - UserController.java
 - com.devsuperior.userdept.entities
 - Department.java
 - User.java
 - com.devsuperior.userdept.repositories
 - UserRepository.java
 - src/main/resources
 - static
 - templates
 - application.properties
 - import.sql
 - src/test/java
 - JRE System Library [JavaSE-17]
 - Maven Dependencies
 - src
 - target
 - HELP.md

User.java application.pro import.sql UserRepository UserController

```
1 import org.springframework.beans.factory.annotation.Autowired;
2 import org.springframework.web.bind.annotation.GetMapping;
3 import org.springframework.web.bind.annotation.RequestMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 import com.devsuperior.userdept.entities.User;
7 import com.devsuperior.userdept.repositories.UserRepository;
8
9 @RestController
10 @RequestMapping(value = "/users")
11 public class UserController {
12
13     @Autowired
14     private UserRepository repository;
15
16     @GetMapping
17     public List<User> findAll(){
18         List<User> result = repository.findAll();
19         return result;
20     }
21
22     @GetMapping(value = "/{id}")
23     public User findById(@PathVariable Long id){
24         List<User> result = re
25         return result;
26     }
27 }
```

Outline X

- com.devsuperior.userdept.controllers
 - UserController
 - repository : UserRepository
 - findAll() : List<User>
 - findById(@PathVariable Long) : User

Problems Javadoc Declaration

userdept - UserDeptApplication [Spring Boot]

Hibernate:

```
tb_user u1_0
select
```

Type tags, projects, or working set names to match (i)

local

Import 'PathVariable' (org.springframework.web.bind.annotation)
Create annotation 'PathVariable'
Rename in file (Ctrl+2, R)
Fix project setup...

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
```

The screenshot shows the Spring Tool Suite 4 interface with the following components:

- Package Explorer:** Shows the project structure under the package `userdept [boot]`. It includes the `src/main/java` folder containing `UserDeptApplication.java`, `UserController.java`, `Department.java`, `User.java`, and `userRepository.java`; the `src/main/resources` folder containing `application.properties` and `import.sql`; and the `src/test/java`, `JRE System Library [JavaSE-17]`, `Maven Dependencies`, `src`, `target`, and `HELP.md` folders.
- User.java Code Editor:** Displays the Java code for the `UserController`. The code defines a REST controller for users, using `@RestController` and `@RequestMapping` annotations. It injects a `UserRepository` via `@Autowired` and provides two methods: `findAll()` which returns a list of `User` objects, and `findById()` which takes a `PathVariable Long id` and returns a single `User` object.
- Boot Dashboard:** A small dashboard window at the bottom left showing various project status indicators.
- Bottom Navigation Bar:** Includes tabs for `Problems`, `Javadoc`, `Declaration`, and `Console`.

The screenshot shows the Spring Tool Suite 4 interface with the following details:

- Package Explorer:** Shows the project structure under "userdept [boot]".
 - src/main/java:** Contains com.devsuperior.userdept (with UserDeptApplication.java), com.devsuperior.userdept.controllers (with UserController.java), com.devsuperior.userdept.entities (with Department.java, User.java), and com.devsuperior.userdept.repositories (with UserRepository.java).
 - src/main/resources:** Contains static, templates, application.properties, import.sql.
 - src/test/java:**
 - JRE System Library [JavaSE-17]:**
 - Maven Dependencies:**
 - src:**
 - target:**
 - HELP.md:**
- User.java Content:** The code defines a REST controller for users.

```
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

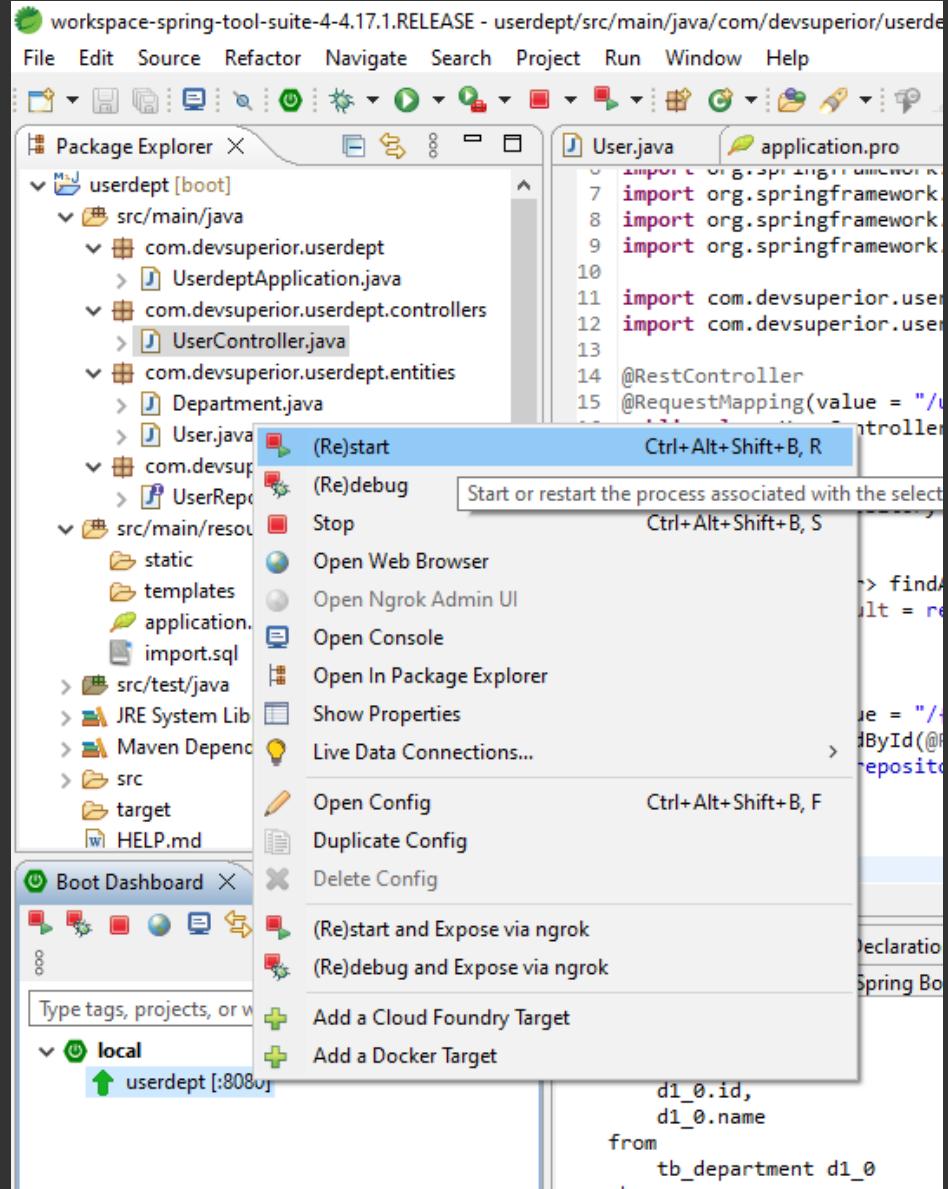
import com.devsuperior.userdept.entities.User;
import com.devsuperior.userdept.repositories.UserRepository;

@RestController
@RequestMapping(value = "/users")
public class UserController {

    @Autowired
    private UserRepository repository;

    @GetMapping
    public List<User> findAll(){
        List<User> result = repository.findAll();
        return result;
    }

    @GetMapping(value = "/{id}")
    public User findById(@PathVariable Long id){
        User result = repository.findById(id).get();
        return result;
    }
}
```
- Outline:** Shows the class structure with methods and fields.
- Bottom Bar:** Includes tabs for Problems, Javadoc, Declaration, and Console, along with various icons for build, run, and navigation.



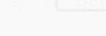
My Workspace

New

Import

GET http://localhost:3

Collections



...

Aula / Users

APIs

GET Users

...

Environments

Mock Servers

Monitors

Flows



Open in Tab

Add example

View Documentation

Rename Ctrl+E

Duplicate Ctrl+D

Paste Ctrl+V

Delete Del

GET

Params

Autho

Query Params

KEY

Key

Body

Cookies

Pretty

Raw

My Workspace

New

Import

GET http://localhost:3000/c



Collections



...

Aula / Users Copy



APIs



Environments



Mock Servers



Monitors



Flows



History



...

▼ Aula

GET Users

GET Users Copy

...

Open in Tab

Add example

View Documentation

Rename Ctrl+E

Duplicate Ctrl+D

Paste Ctrl+V

Delete Del

GET

...

Enter

Params

Authorization

Query Params

KEY

Key

Response

My Workspace

New

Import

GET http://localhost:3000/c ● GET



Collections



...

Aula



APIs



GET Users



Environments



Mock Servers



Monitors



Flows



History

Aula / Users By Id

GET



Enter request

Params

Authorization

Help

Query Params

KEY

Key

Response

The screenshot shows the Postman application interface. At the top, there are navigation links: Home, Workspaces, API Network, and Explore. A search bar labeled "Search Postman" is also at the top. On the right side, there is an "Invite" button.

The main area is titled "My Workspace". On the left, there is a sidebar with three sections: Collections, APIs, and Environments. Under Collections, there is a "Aula" folder containing two items: "GET Users" and "GET Users By Id". Under APIs, there is one item: "GET Users".

The "Aula / Users" section is currently selected. It shows a "GET" request with the URL "http://localhost:8080/users". Below the URL, there are tabs for "Params", "Authorization", "Headers (6)", "Body", "Pre-request Script", "Tests", and "Settings".

Copy URL

My Workspace New Import GET http://localhost:3000/c ● GET Users By Id ● GET Users ● + ... No Environment

Collections + ⌂ ... Save Aula / Users By Id Edit

APIs GET Users GET Users By Id

Environments

Mock Servers

Monitors

Flows

History

Aula / Users By Id

GET http://localhost:8080/users/1

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: 574 ms Size: 252 B Save Response

Pretty Raw Preview Visualize JSON

```
1 "id": 1,
2 "name": "Maria",
3 "email": "maria@gmail.com",
4 "department": {
5   "id": 1,
6   "name": "Gestão"
7 }
8
9 }
```

My Workspace New Import GET http://localhost:3000/c ● GET Users By Id ● GET Users ● + ... No Environment

Aula / Users By Id

GET http://localhost:8080/users/3

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

Query Params

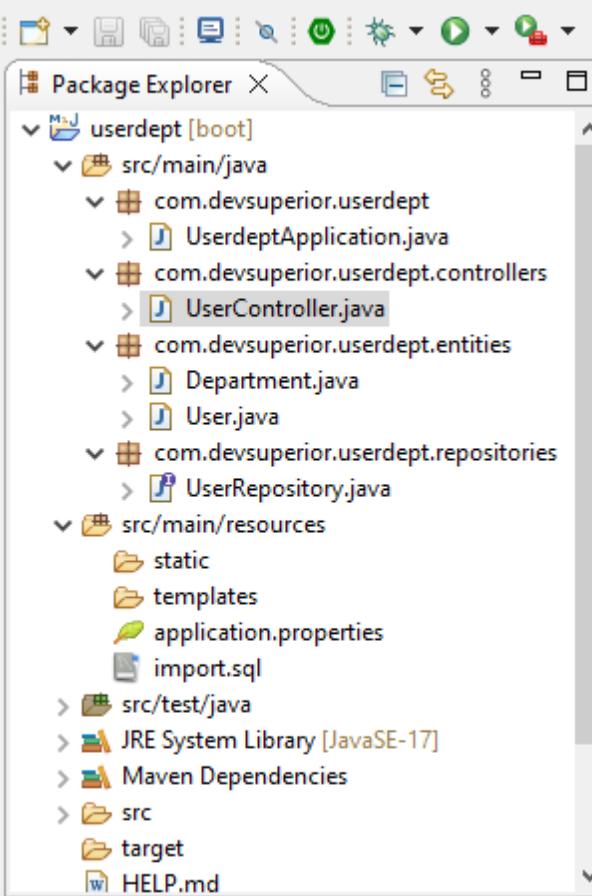
KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description	...	

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 17 ms Size: 255 B Save Response

Pretty Raw Preview Visualize JSON

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



```
User.java application.pro import.sql UserRepository UserController.java

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import com.devsuperior.userdept.entities.User;
import com.devsuperior.userdept.repositories.UserRepository;
@RestController
@RequestMapping(value = "/users")
public class UserController {
    @Autowired
    private UserRepository repository;
    @GetMapping
    public List<User> findAll(){
        List<User> result = repository.findAll();
        return result;
    }
    @GetMapping(value = "/{id}")
    public User findById(@PathVariable Long id){
        User result = repository.findById(id).get();
        return result;
    }
}
```

Copy e paste logo
abaixo

The screenshot shows the Spring Tool Suite 4 interface with the following details:

- Project Explorer:** Shows the project structure under "userdept [boot]".
 - src/main/java:** Contains `UserdeptApplication.java`, `UserController.java`, `Department.java`, `User.java`, and `UserRepository.java`.
 - src/main/resources:** Contains `application.properties`, `import.sql`, and static files.
- Code Editor:** Displays the `UserController.java` file with Java code for a REST controller.

```
12 import com.devsuperior.userdept.repositories.UserRepository;
13
14 @RestController
15 @RequestMapping(value = "/users")
16 public class UserController {
17
18     @Autowired
19     private UserRepository repository;
20
21     @GetMapping
22     public List<User> findAll(){
23         List<User> result = repository.findAll();
24         return result;
25     }
26
27     @GetMapping(value = "/{id}")
28     public User findById(@PathVariable Long id){
29         User result = repository.findById(id).get();
30         return result;
31     }
32
33     @GetMapping(value = "/{id}")
34     public User findById(@PathVariable Long id){}
35         User result = repository.findById(id).get();
36         return result;
37     }
38 }
39 }
```
- Bottom Bar:** Includes tabs for `Problems`, `Javadoc`, `Declaration`, and `Console`.

The screenshot shows the Spring Tool Suite 4 interface with the following details:

- Project Explorer:** Shows the project structure under the package `userdept [boot]`. It includes `src/main/java` with `UserDeptApplication.java`, `UserController.java`, `Department.java`, `User.java`, and `userRepository.java`; `src/main/resources` with `application.properties` and `import.sql`; and `src/test/java`.
- Code Editor:** Displays the `UserController.java` file with the following code:

```
12 import com.devsuperior.userdept.repositories.UserRepository;
13
14 @RestController
15 @RequestMapping(value = "/users")
16 public class UserController {
17
18     @Autowired
19     private UserRepository repository;
20
21     @GetMapping
22     public List<User> findAll(){
23         List<User> result = repository.findAll();
24         return result;
25     }
26
27     @GetMapping(value = "/{id}")
28     public User findById(@PathVariable Long id){
29         User result = repository.findById(id).get();
30         return result;
31     }
32
33     @PostMapping
34     public User insert(User user){
35         User result = repository.findById(id).get();
36         return result;
37     }
38 }
```
- Annotations:** A red arrow points from the text "Digite o code e adicione o import da anotation" to the `@PostMapping` annotation at line 33.
- Bottom Navigation:** Shows tabs for `Problems`, `Javadoc`, `Declaration`, and `Console`.

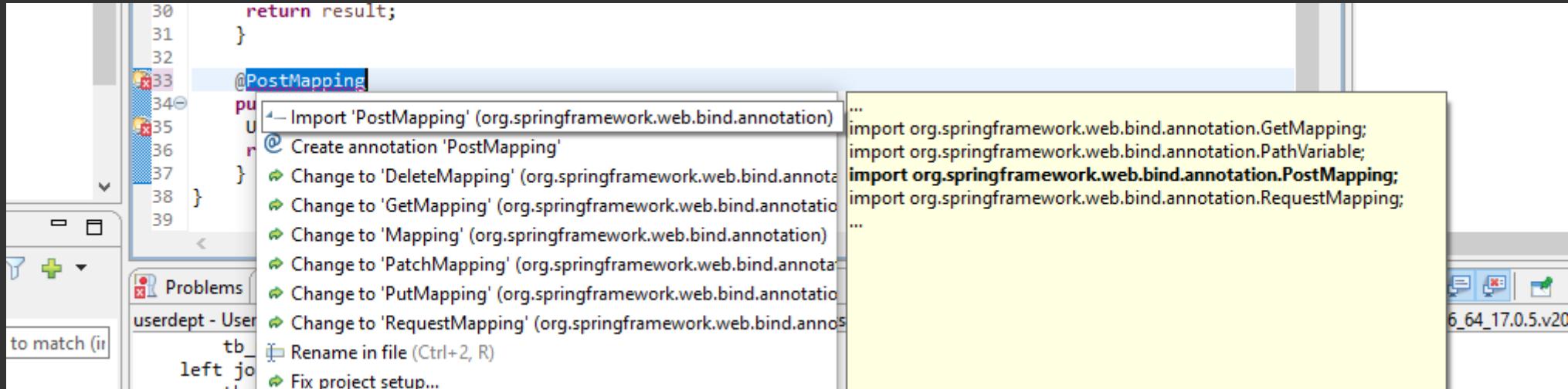
**Digite o code e
adicione o import da
anotation**

```
    return result;  
}
```

```
@PostMapping
```

- pu Import 'PostMapping' (org.springframework.web.bind.annotation)
- U Create annotation 'PostMapping'
- r Change to 'DeleteMapping' (org.springframework.web.bind.annotation)
- Change to 'GetMapping' (org.springframework.web.bind.annotation)
- Change to 'Mapping' (org.springframework.web.bind.annotation)
- Change to 'PatchMapping' (org.springframework.web.bind.annotation)
- Change to 'PutMapping' (org.springframework.web.bind.annotation)
- Change to 'RequestMapping' (org.springframework.web.bind.annotation)
- Rename in file (Ctrl+2, R)
- Fix project setup...

```
...  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
...  
...
```



A screenshot of an IDE interface, likely IntelliJ IDEA, showing a Java code editor. The code is part of a controller class:

```
31     return result;
32 }
33
34 @PostMapping
35 public User insert(@RequestBody User user){
36     User result = reposi
37     return result;
38 }
39
40 }
```

The cursor is positioned at the end of the line `@RequestBody User user)` and a red arrow points down to the code completion dropdown menu. The menu lists several options related to the `@RequestBody` annotation:

- Import 'RequestBody' (org.springframework.web.bind.annotation)
- @ Create annotation 'RequestBody'
- Change to 'RequestHeader' (org.springframework.web.bind.annotation)
- Change to 'RequestMapping' (org.springframework.web.bind.annotation)
- Change to 'RequestParam' (org.springframework.web.bind.annotation)
- Change to 'RequestScope' (org.springframework.web.context.annotation)

Below the completion menu, a preview of the generated imports is shown:

```
...
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
...

```

Adicione a anotation e seu import

The screenshot shows the Spring Tool Suite 4 interface with the following details:

- Package Explorer:** Shows the project structure under the package `userdept`. It includes the `src/main/java` directory containing `UserDeptApplication.java`, `UserController.java`, `Department.java`, `User.java`, and `UserRepository.java`; and the `src/main/resources` directory containing `application.properties`, `import.sql`, and static/templates.
- Editor:** Displays the code for `UserController.java`. The code defines a REST controller for users, using `@RestController` and `@RequestMapping` annotations. It injects a `UserRepository` via `@Autowired`. The code handles three methods: `findAll()` which returns all users, `findById()` which returns a user by ID, and `insert()` which inserts a new user.
- Status Bar:** A green message "Termine o code" is displayed in the status bar.

```
14 import com.devsuperior.userdept.repositories.UserRepository;
15
16 @RestController
17 @RequestMapping(value = "/users")
18 public class UserController {
19
20     @Autowired
21     private UserRepository repository;
22
23     @GetMapping
24     public List<User> findAll(){
25         List<User> result = repository.findAll();
26         return result;
27     }
28
29     @GetMapping(value = "/{id}")
30     public User findById(@PathVariable Long id){
31         User result = repository.findById(id).get();
32         return result;
33     }
34
35     @PostMapping
36     public User insert(@RequestBody User user){
37         User result = repository.save(user);
38         return result;
39     }
40 }
```

workspace-spring-tool-suite-4-4.17.1.RELEASE - userdept/src/main/java/com/devsuperior/

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

userdept [boot]

src/main/java

- com.devsuperior.userdept
- UserdeptApplication.java
- com.devsuperior.userdept.controllers
- UserController.java
- com.devsuperior.userdept.entities
- Department.java
- User.java
- com.devsuperior.userdept.repositories
- UserRepository.java

src/main/resources

- static
- templates
- application.properties
- import.sql

src/test/java

JRE System Library

Maven Dependencies

src

target

HELP.md

User.java application.properties

```
15  
16 @RestController  
17 @RequestMapping(value = "/users")  
18 public class UserController {  
19     @Autowired  
20     private UserRepository userRepository;  
21     @GetMapping  
22     public List<User> getAllUsers() {  
23         return userRepository.findAll();  
24     }  
25 }
```

(Re)start Ctrl+Alt+Shift+B, R

(Re)debug Start or restart the process associated with the current configuration

Stop Ctrl+Alt+Shift+B, S

Open Web Browser

Open Ngrok Admin UI

Open Console

Open In Package Explorer

Show Properties

Live Data Connections...

Open Config Ctrl+Alt+Shift+B, F

Duplicate Config

Delete Config

(Re)start and Expose via ngrok

(Re)debug and Expose via ngrok

Add a Cloud Foundry Target

Add a Docker Target

Type tags, projects, or words

local

userdept [:8080]

on d1_0.id=u1_0 where u1_0.id=3

My Workspace

New

Import

GET http://localhost:300

Collections

+



...

Aula

APIs

GET Users

Environments

GET Users By Id

...

Mock Servers

Monitors

Flows

History

Open in Tab

Add example

View Documentation

Rename Ctrl+E

Duplicate Ctrl+D

Paste Ctrl+V

Delete Del

Aula / Users By Id

GET

▼

Params

Query Params

KEY

Key

Body Cookies Headers

Pretty Raw

```
1
2      "id": "123"
3      "name": "John Doe"
```

The screenshot shows a workspace interface with the following structure:

- Collections:** Aula
- APIs:**
 - GET Users
 - GET Users By Id
 - GET new user
- Environments:**
- Mock Servers:**
- Monitors:**

At the top right, there are "New" and "Import" buttons.

≡ Home Workspaces API Network Explore

My Workspace

New Import

GET http://localhost:3001 Aula / new user

Collections + ⚙️ ... Aula

APIs GET Users

Environments GET Users By Id

Mock Servers GET new user

Aula / new user

POST Enter request URL

Params Authorization Headers

Query Params

KEY

Key

The screenshot shows a user interface for a REST API testing tool. On the left, there's a sidebar with sections for Collections, APIs, Environments, and Mock Servers. Under Collections, 'Aula' is expanded, showing three endpoints: 'GET Users', 'GET Users By Id', and 'GET new user'. The 'GET new user' endpoint is currently selected, highlighted with a grey background. To the right of the sidebar, a main panel displays the details of this endpoint. At the top of this panel, there's a title 'Aula / new user' with a red arrow pointing down to it. Below the title, there are fields for 'POST' method and 'Enter request URL'. Underneath these, there are tabs for 'Params', 'Authorization', and 'Headers', with 'Params' being the active tab. The 'Params' section is titled 'Query Params' and contains two entries: 'KEY' and 'Key'. The overall theme of the interface is dark, with light-colored cards for each API endpoint.

My Workspace New Import GET http://localhost:3001 ● GET Users By Id ● GET Users ● POST new user ● + ... No Environment

Aula / new user

POST Enter request URL

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Response



Enter the URL and click Send to get a response

Collections APIs Environments Mock Servers Monitors Flows History

My Workspace New Import GET http://localhost:3001 ● GET Users By Id ● GET Users ● POST new user ● + ... No Environment

Aula / new user

POST http://localhost:8080/users

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL Text

Text
JavaScript
JSON
HTML
XML

1

Response

Click Send to get a response



Online Find and Replace Console Cookies Capture requests Bootcamp Runner Trash

My Workspace

New Import

GET http://localhost:3001

GET Users By Id

GET Users

Collections

+ ⚙ ...

Aula

GET Users

GET Users By Id

GET new user

APIs

Environments

Mock Servers

Monitors

Flows

History

Aula / Users By Id

GET

http://localhost:8080/users/3

Params

Authorization

Headers (6)

Body

Pre-request Script

Query Params

	KEY	VALUE
	Key	Value

Body Cookies Headers (5) Test Results

Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "id": 3,  
3   "name": "Alex",  
4   "email": "alex@gmail.com",  
5   "department": {  
6     "id": 2,  
7     "name": "Informática"  
8   }  
9 }
```

copy

The screenshot shows the Postman application interface. On the left, there's a sidebar with icons for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main area displays a workspace titled "Aula / new user". A POST request is being prepared to the URL `http://localhost:8080/users`. The "Body" tab is selected, showing a JSON payload:

```
1 {
2     "id": 3,
3     "name": "Alex",
4     "email": "alex@gmail.com",
5     "department": {
6         "id": 2,
7         "name": "Informática"
8     }
9 }
```

A red arrow points from the text "Paste e mude os dados" to the JSON code in the "Body" section.

Paste e mude os dados

My Workspace New Import GET http://localhost:3001 ● GET Users By Id ● GET Users ● POST new user ● + ... No Environment

Aula / new user

POST http://localhost:8080/users

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "name": "Joyce",
3   "email": "joyce@gmail.com",
4   "department": "IT"
5   "id": 2
6 }
7 }
```

Send Cookies Beautify

Response

Click Send to get a response



My Workspace New Import GET http://localhost:3001 ● GET Users By Id ● GET Users ● POST new user ● + ... No Environment

Aula / new user

POST http://localhost:8080/users

Params Authorization Headers (8) Body Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {  
2   "name": "Joyce",  
3   "email": "joyce@gmail.com",  
4   "department": {}  
5   |   "id": 2  
6 }  
7 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 604 ms Size: 247 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": 5,  
3   "name": "Joyce",  
4   "email": "joyce@gmail.com",  
5   "department": {}  
6   |   "id": 2,  
7   |   "name": null  
8 }  
9 }
```

Online Find and Replace Console Cookies Capture requests Bootcamp Runner Trash

Tempestades a... 16:56

Collections APIs Environments Mock Servers Monitors Flows History

A screenshot of the Postman interface. On the left sidebar, under 'Collections', there is a section for 'Aula' containing three items: 'GET Users' (selected), 'GET Users By Id', and 'GET new user'. The main workspace shows a request for 'GET http://localhost:8080/users'. The 'Headers' tab is selected, showing '(6)' entries. Below it is a table for 'Query Params' with columns: KEY, VALUE, DESCRIPTION, and Bulk Edit. The 'Body' tab is selected in the response pane, which displays a JSON response with two users: Maria and Bob. A red arrow points from the text 'Volte em users e veja se as informações foram repassadas' to the 'Body' tab.

My Workspace

New Import

GET http://localhost:3001

GET Users By Id

GET Users

POST new user

No Environment

Collections

+ Aula

APIs

Environments

Mock Servers

Monitors

Flows

History

Aula / Users

GET http://localhost:8080/users

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

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 173 ms Size: 521 B Save Response

Pretty Raw Preview Visualize JSON

```
2  {
3   "id": 1,
4   "name": "Maria",
5   "email": "maria@gmail.com",
6   "department": {
7     "id": 1,
8     "name": "Gestão"
9   }
10 },
11 {
12   "id": 2,
13   "name": "Bob",
14   "email": "bob@gmail.com",
```

Volte em users e veja se as informações foram repassadas

My Workspace

New Import

GET http://localhost:3001 | GET Users By Id | GET Users | POST new user | No Environment

Aula / Users

GET http://localhost:8080/users

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

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

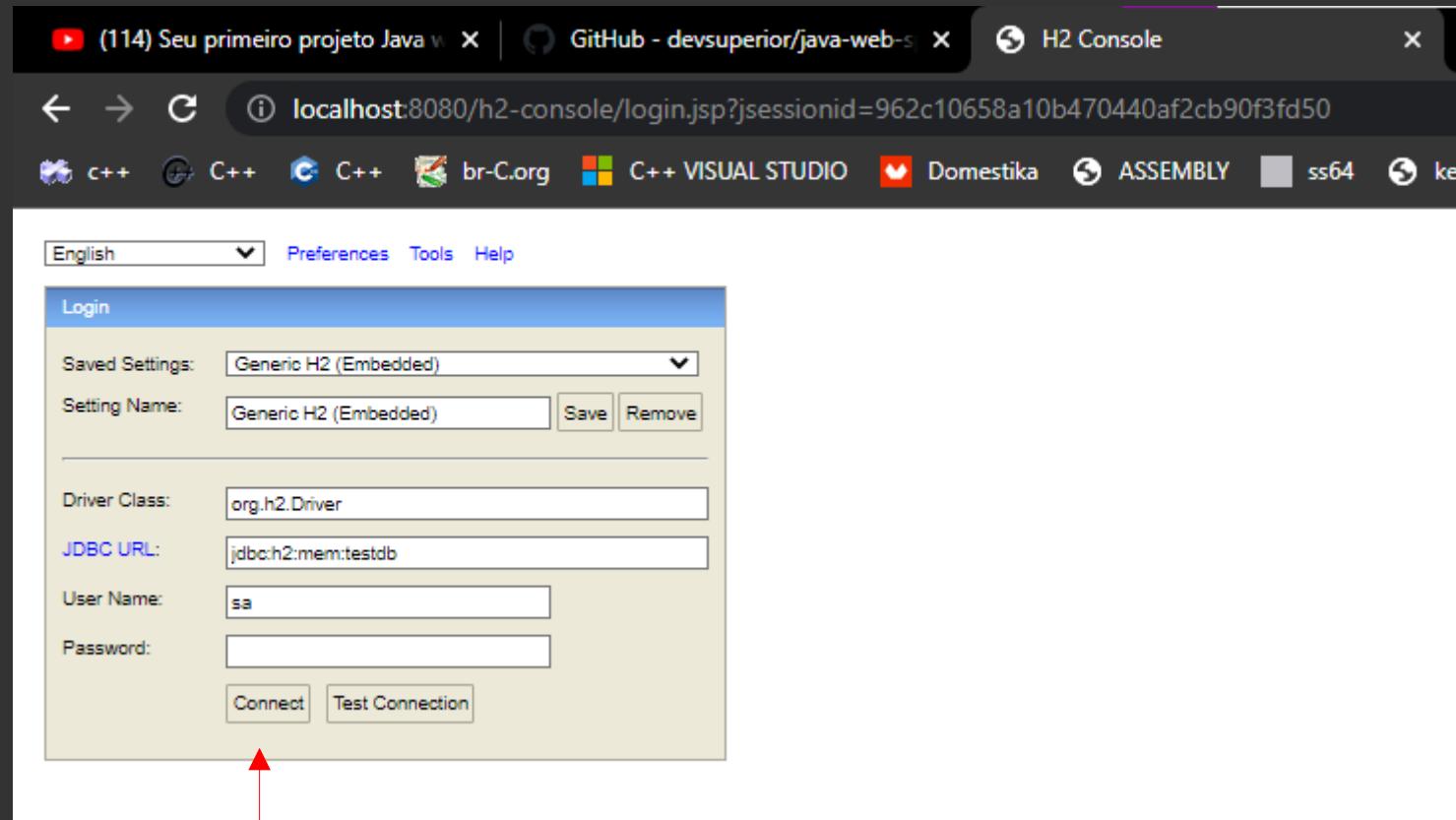
Body Cookies Headers (5) Test Results

Status: 200 OK Time: 338 ms Size: 615 B Save Response

Pretty Raw Preview Visualize JSON

29 | {
30 | | "id": 4,
31 | | "name": "Ana",
32 | | "email": "ana@gmail.com",
33 | | "department": {
34 | | | "id": 2,
35 | | | "name": "Informática"
36 | }
37 | ,
38 | {
39 | | "id": 5,
40 | | "name": "Joyce",
41 | | "email": "joyce@gmail.com",
42 | | "department": {
43 | | | "id": 2,
44 | | | "name": "Informática"
45 | }
46 | }

Foram repassadas pois a joyce foi adicionada



Funcionando perfeitamente a api se comunida interna e externa

The screenshot shows the H2 Console interface running in a browser window. The title bar indicates the page is at `localhost:8080/h2-console/login.do?jsessionid=962c10658a10b470440af2cb90f3fd50`. The main area displays a SQL query and its results.

SQL Statement:

```
SELECT * FROM TB_USER;
```

Results:

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

(5 rows, 2 ms)

Toolbars and Buttons:

- Auto commit checkbox (checked)
- Max rows dropdown set to 1000
- Run button (highlighted by a red arrow)
- Run Selected button
- Auto complete dropdown set to Off
- Auto select dropdown set to On

Sidebar:

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

Passando os dados em json

A partir daqui os dados estão sendo disponibilizados em uma url ou seja uma api em formato json. Podendo ser usada em qualquer aplicação front-end

Nano 2023