

# **PROJETO CRUD - BACKEND**

## **SUMÁRIO**

<b>PROJETO CRUD - BACKEND.....</b>	<b>1</b>
OBJETIVO.....	6
DEPENDÊNCIAS.....	6
CASO DE USO.....	6
DIAGRAMA DE CLASSE.....	7
TESTES A SEREM REALIZADOS NO POSTMAN.....	8
BUSCA PAGINADA DE ALUNOS.....	8
BUSCA DE ALUNO POR ID.....	8
INSERIR NOVO ALUNO.....	8
ATUALIZAR ALUNO.....	8
DELETAR ALUNO.....	8
<b>IMPLEMENTAR O BACKEND.....</b>	<b>9</b>
DEFINIR O WORKSPACE PARA O PROJETO.....	9
CRIAR O PROJETO COM O INITIALIZR.....	9
Salvar dentro do workspace.....	10
Descompactar e Renomear.....	10
IMPORTAR O PROJETO PARA O STS.....	11
TESTAR NO BROWSER.....	13
VERSIONAMENTO.....	13
Criar o projeto no Github.....	13
Sincronizar com o projeto local via git.....	15
Adicionar o README ao projeto.....	19
Atualizar o projeto local.....	21
<b>INICIAR O DESENVOLVIMENTO DO PROJETO.....</b>	<b>23</b>
STUDENT_CLASSE.....	23
DIAGRAMA DE CLASSE.....	23
IMPLEMENTAR A ESTRUTURA E A CLASSE STUDENT NO STS.....	24
Criar a estrutura e classe.....	24
Definir o Serializable e os atributos da Classe.....	24
Criar o construtor.....	26
Criar os Getters and Setters.....	27

Criar o hashCode and equals.....	29
Github-2.....	30
STUDENT_CONTROLLER.....	31
CONCEITUAL.....	31
IMPLEMENTAR A ESTRUTURA.....	32
Criar a estrutura e o recurso StudentController.....	32
Implementar as Notações Rest.....	32
Criar o Endpoint findAll para teste.....	33
Rodar o projeto.....	33
Testar com o Postman.....	34
Github-3.....	34
STUDENT_REPOSITORY.....	35
CONCEITUAL.....	35
IMPLEMENTAR A ESTRUTURA E O STUDENT_REPOSITORY.....	36
Criar a estrutura e a interface de acesso ao banco.....	36
Implementar a notação e estender a JPA.....	36
Github-4.....	37
STUDENT_SERVICE.....	38
CONCEITUAL.....	38
IMPLEMENTAR A ESTRUTURA E O STUDENT_SERVICE.....	39
Criar a estrutura e a classe de serviço.....	39
Implementar a lógica para o endpoint findAll.....	40
Github-5.....	41
INTEGRAÇÃO COM O BANCO.....	42
AJUSTAR AS CAMADAS.....	42
Implementar o StudentController.....	42
Implementar a classe Student.....	43
Rodar o projeto.....	43
Testar com o Postman.....	43
BANCO H2.....	44
Configurar o perfil de teste no application.properties.....	44
Criar o arquivo de configuração application-test.properties.....	45
Implementar o application-test.properties.....	46
Rodar o projeto.....	46
Acessar o banco H2, via web.....	47
Teste de inserção.....	48
Testar no postman.....	48
Github-6.....	49
SEEDING DA BASE DE DADOS.....	50

IMPLEMENTAR A CARGA PARA O BANCO.....	50
Criar o import.sql, no src/main/resources.....	50
Implementar os inserts para a carga inicial.....	51
Rodar o projeto.....	51
TESTAR A CARGA.....	52
Testar no banco H2.....	52
Testar no Postman.....	53
Github-7.....	54
DTO.....	55
CONCEITUAL.....	55
IMPLEMENTAR A ESTRUTURA E O STUDENT_DTO.....	56
Criar a estrutura e a classe DTO.....	56
Implementar o Serializable e os mesmos atributos da Classe Student.....	56
Implementar os construtores.....	57
Vazio.....	57
de Classe.....	57
De Entidade.....	58
Implementar os Getters and Setters.....	58
REALIZAR OS AJUSTES PARA O DTO.....	60
Implementar o DTO na classe StudentService.....	60
Implementar o DTO na classe StudentController.....	61
Rodar o projeto.....	61
TESTAR COM O POSTMAN.....	62
Github-8.....	63
ENDPOINT: FIND_BY_ID.....	64
BUSCAR ALUNOS POR ID COM GET.....	64
Implementar busca por Id, no StudentController.....	64
Implementar o método findById, no StudentService.....	64
Rodar o projeto.....	65
TESTAR NO POSTMAN.....	66
Github-9.....	67
TRATAMENTO DE EXCEÇÕES PARA O FIND_BY_ID.....	68
Rodar o projeto.....	68
Simular erro no Postman.....	68
Verificar o erro no console.....	68
Criar a estrutura de exceções no service.....	69
Implementar o ResourceNotFoundException.....	69
Implementar a exceção no findById, do StudentService.....	70
Criar a estrutura de exceções e uma classe personalizada no controller.....	70

Implementar o Serializable e os atributos do StandardError conforme erro visto anteriormente.....	71
Implementar um construtor vazio.....	72
Implementar os Getters and Setters.....	72
Criar um controller advice para manipular a exceção.....	74
Implementar o ResourceExceptionHandler.....	75
Rodar o projeto.....	75
TESTAR NO POSTMAN.....	76
Github-10.....	76
PAGINAÇÃO.....	77
AJUSTAR O FIND_ALL PARA BUSCA PAGINADA.....	77
Implementar a busca paginada, no StudentController.....	77
Ajustar a busca paginada, no StudentService.....	78
Expandir o seed do banco para teste de paginação.....	79
Rodar o projeto.....	79
TESTAR NO POSTMAN.....	80
Github-11.....	81
ENDPOINT - INSERT.....	82
INSERIR NOVO ALUNO COM POST.....	82
Implementar o insert, no StudentController.....	82
Implementar a metodologia REST ao método.....	82
Implementar o método insert, convertendo o DTO para uma entidade, no StudentService.....	83
Rodar o projeto.....	83
TESTAR NO POSTMAN.....	84
Inserir.....	84
Buscar por Id.....	85
Github-12.....	85
ENDPOINT - UPDATE.....	86
ATUALIZAR ALUNO COM PUT.....	86
Implementar o update, no StudentController.....	86
Implementar o método update, no StudentService.....	86
Rodar o projeto.....	86
TESTAR NO POSTMAN.....	87
Update.....	87
Busca por ID.....	87
TRATAMENTO DE ERRO PARA O UPDATE.....	88
Implementar o tratamento para ID Não encontrado.....	88
Rodar o projeto.....	88

TESTAR NO POSTMAN.....	89
Update.....	89
Github-13.....	89
ENDPOINT - DELETE.....	90
DELETAR UM ALUNO COM O MÉTODO REST DELETE.....	90
Implementar o update, no StudentController.....	90
Implementar o método update, no StudentService.....	90
Rodar o projeto.....	90
TESTAR NO POSTMAN.....	91
TRATAMENTO DE ERRO DO DELETE.....	91
Implementar o tratamento para ID Não encontrado.....	91
Rodar o projeto.....	92
TESTAR NO POSTMAN.....	92
Github-14.....	92
<b>IMPLEMENTAR O SWAGGER.....</b>	<b>93</b>
PROJETO.....	93
IMPLEMENTAÇÃO DO SWAGGER.....	94
DEPENDÊNCIA MAVEN.....	94
IMPLEMENTAR O MAIN PRINCIPAL.....	94
IMPLEMENTAR O CONTROLLER.....	95
FindAll.....	95
FindById.....	95
Insert.....	95
Update.....	96
Delete.....	96
ACESSAR O SWAGGER.....	96
Rodar o projeto.....	96
Acesso online.....	97
Testar o Swagger.....	98
Buscar todos.....	98
Busca por ID.....	99
Atualizar.....	100
Inserir.....	101
Deletar.....	102
<b>Fim.....</b>	<b>102</b>

## **OBJETIVO**

Criar uma API RESTful com as funcionalidades de CRUD. Criar um front end intuitivo, sincronizado com o backend, para realizar as operações de visualização, inserção, edição e deleção dos dados.

## **DEPENDÊNCIAS**

O projeto será implementado conforme abaixo:

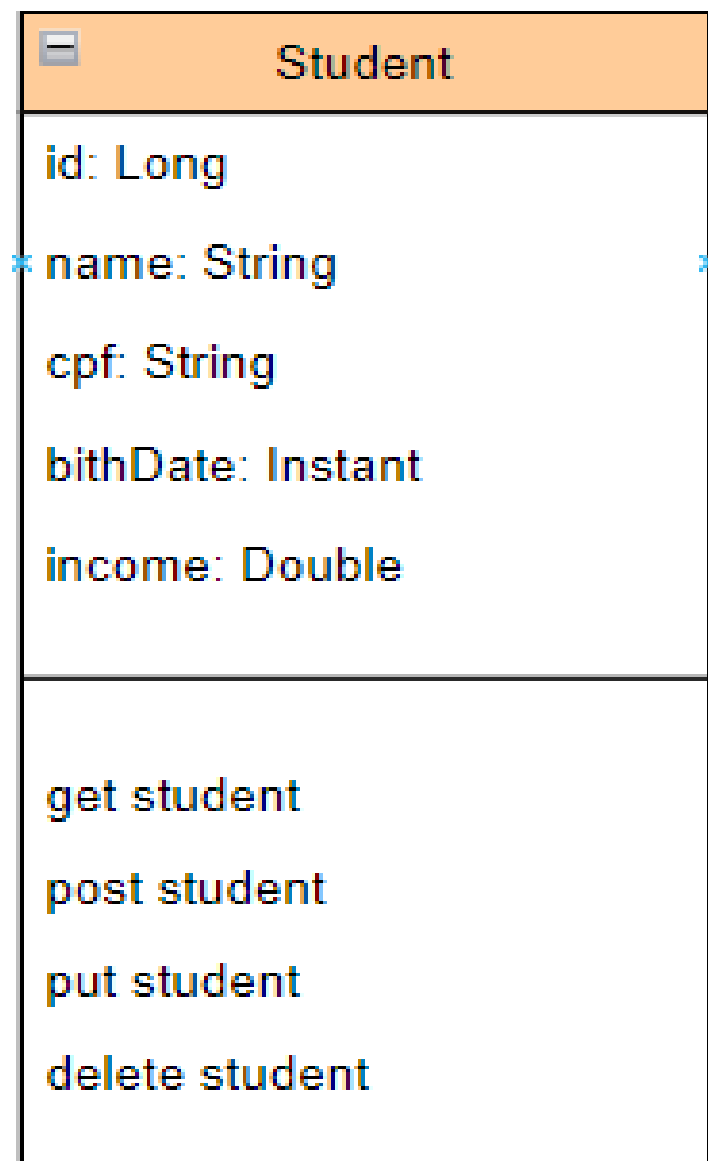
- **IDE:** Spring Tool Suite 4 (STS)
- **Banco de Dados:** H2
- **Gerenciador de Dependências:** Maven
- **Linguagem:** Java
- **Versionamento:** GitHub
- **Testes da API:** Postman
- **Documentação:** Swagger OpenApi

## **CASO DE USO**

Uma Universidade precisa cadastrar os seus alunos, conforme segue abaixo:

- NOME
- CPF
- DATA DE NASCIMENTO
- RENDA

## DIAGRAMA DE CLASSE



## TESTES A SEREM REALIZADOS NO POSTMAN

### BUSCA PAGINADA DE ALUNOS

- GET /students?page=0&sort=name,asc

### BUSCA DE ALUNO POR ID

- GET /students/1

### INSERIR NOVO ALUNO

- POST /students  
{  
 "name": "Auricelio Freitas",  
 "cpf": "12345678901",  
 "birthDate": "1982-08-28T10:30:00Z",  
 "income": 15089.0  
}

### ATUALIZAR ALUNO

- PUT /students/1  
{  
 "name": "Auricelio Moreira",  
 "cpf": "12345678901",  
 "birthDate": "1982-08-28T10:30:00Z",  
 "income": 15089.0  
}

### DELETAR ALUNO

- DELETE /students/1



## IMPLEMENTAR O BACKEND

### DEFINIR O WORKSPACE PARA O PROJETO

- C:\PROJETOS\ProjCrudBackFront

### CRIAR O PROJETO COM O INITIALIZR

#### ACESSAR A URL

- <https://start.spring.io/>

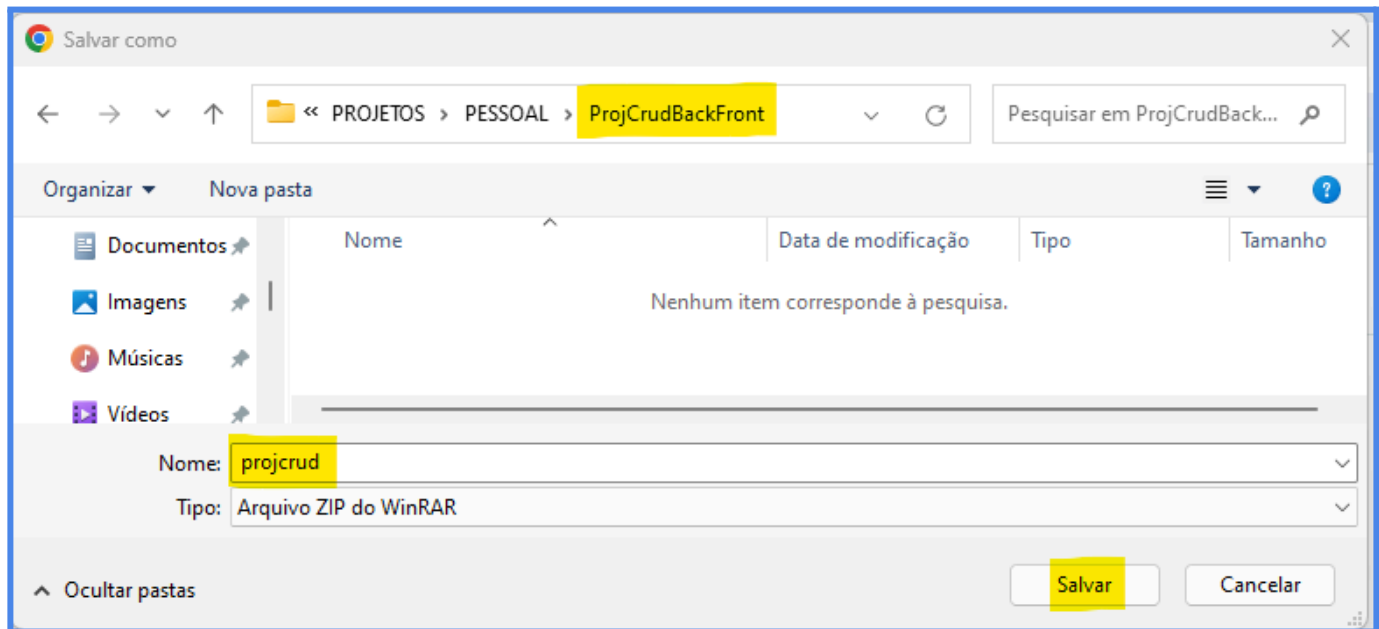
#### GERAR O PROJETO

The screenshot shows the Spring Initializr web application interface. The browser address bar displays 'start.spring.io'. The page features a sidebar with a hamburger menu and a search icon. The main content area is divided into several sections:

- Project:** Includes radio buttons for 'Gradle - Groovy', 'Gradle - Kotlin', 'Java' (selected), 'Kotlin', and 'Groovy'. Below this is a 'Maven' button.
- Spring Boot:** Includes radio buttons for '3.3.2 (SNAPSHOT)', '3.3.1' (selected), '3.2.8 (SNAPSHOT)', and '3.2.7'.
- Project Metadata:** Includes text input fields for 'Group' (filled with 'pessoal'), 'Artifact' (filled with 'projcrud'), 'Name' (filled with 'projcrud'), 'Description' (filled with 'Projeto CRUD Restful'), and 'Package name' (filled with 'pessoal.projcrud'). It also has a 'Packaging' section with 'Jar' (selected) and 'War' options, and a 'Java' version section with '22', '21', and '17' (selected).
- Dependencies:** Includes a button 'ADD DEPENDENCIES... CTRL + B'. Below it are several dependency cards: 'Spring Web' (WEB), 'Spring Data JPA' (SQL), 'H2 Database' (SQL), and 'PostgreSQL Driver' (SQL). Each card has a brief description.

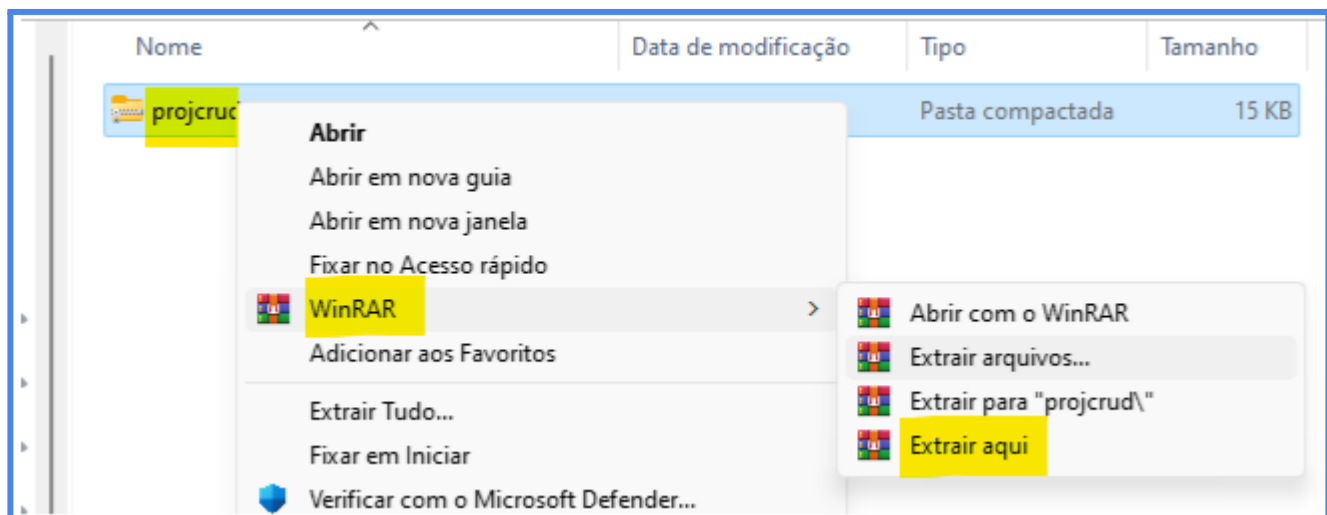
At the bottom of the page, there are three buttons: 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'.

## Salvar dentro do workspace

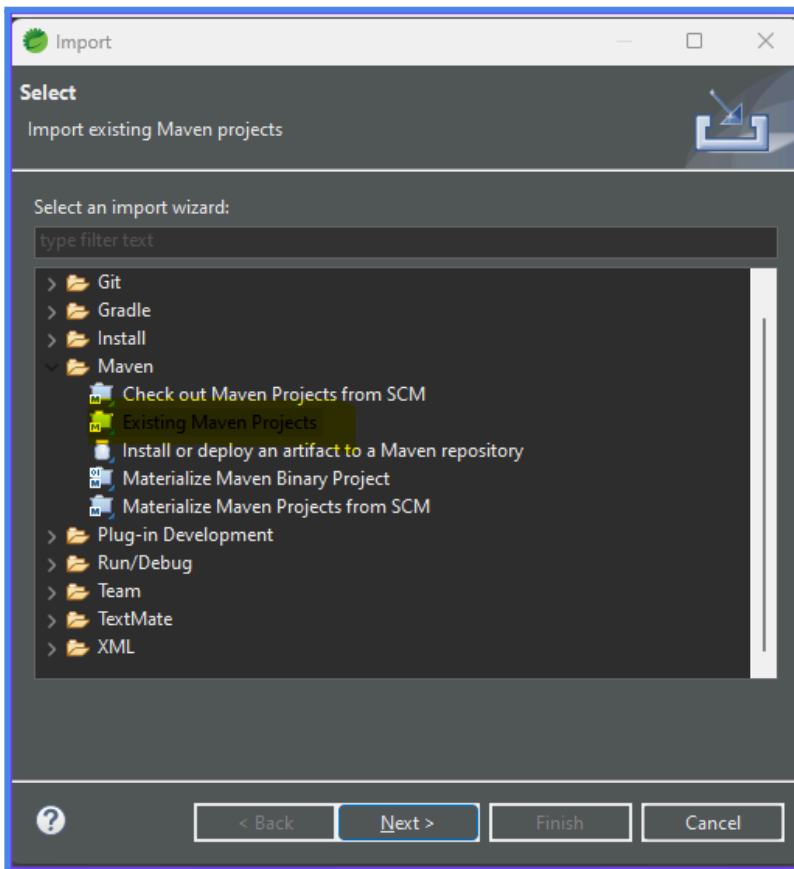
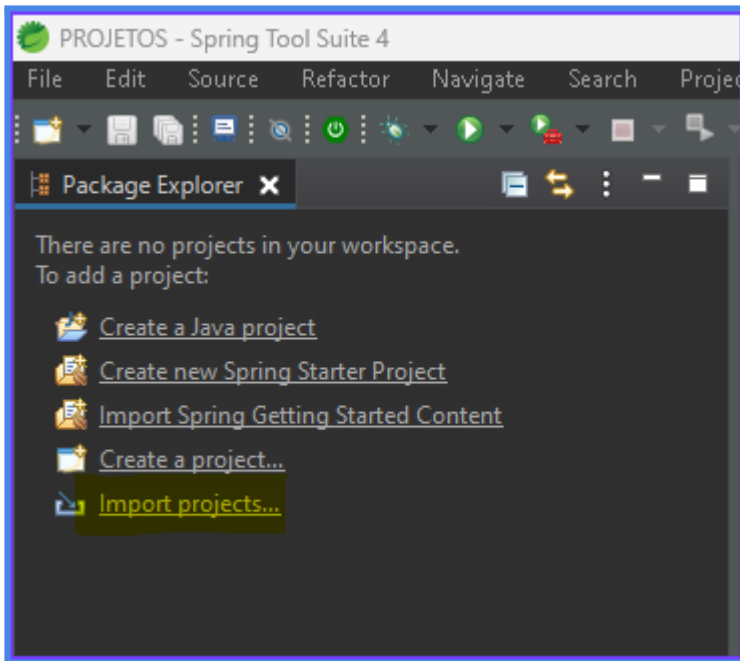


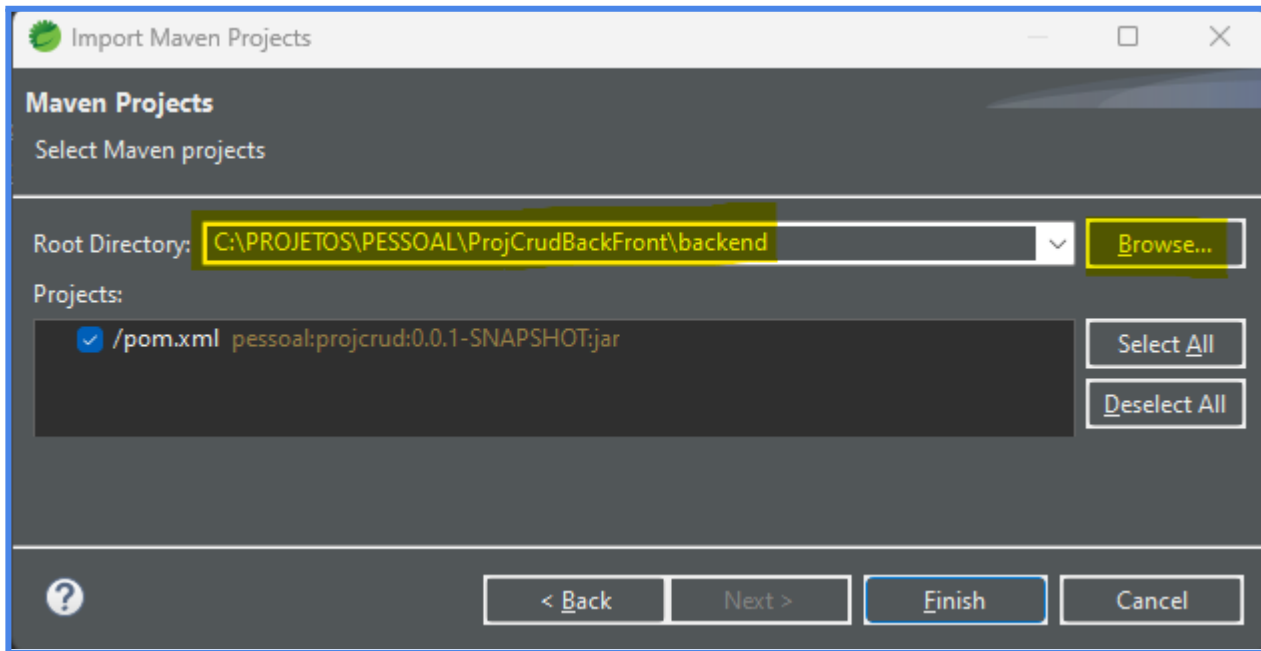
## Descompactar e Renomear

- backend

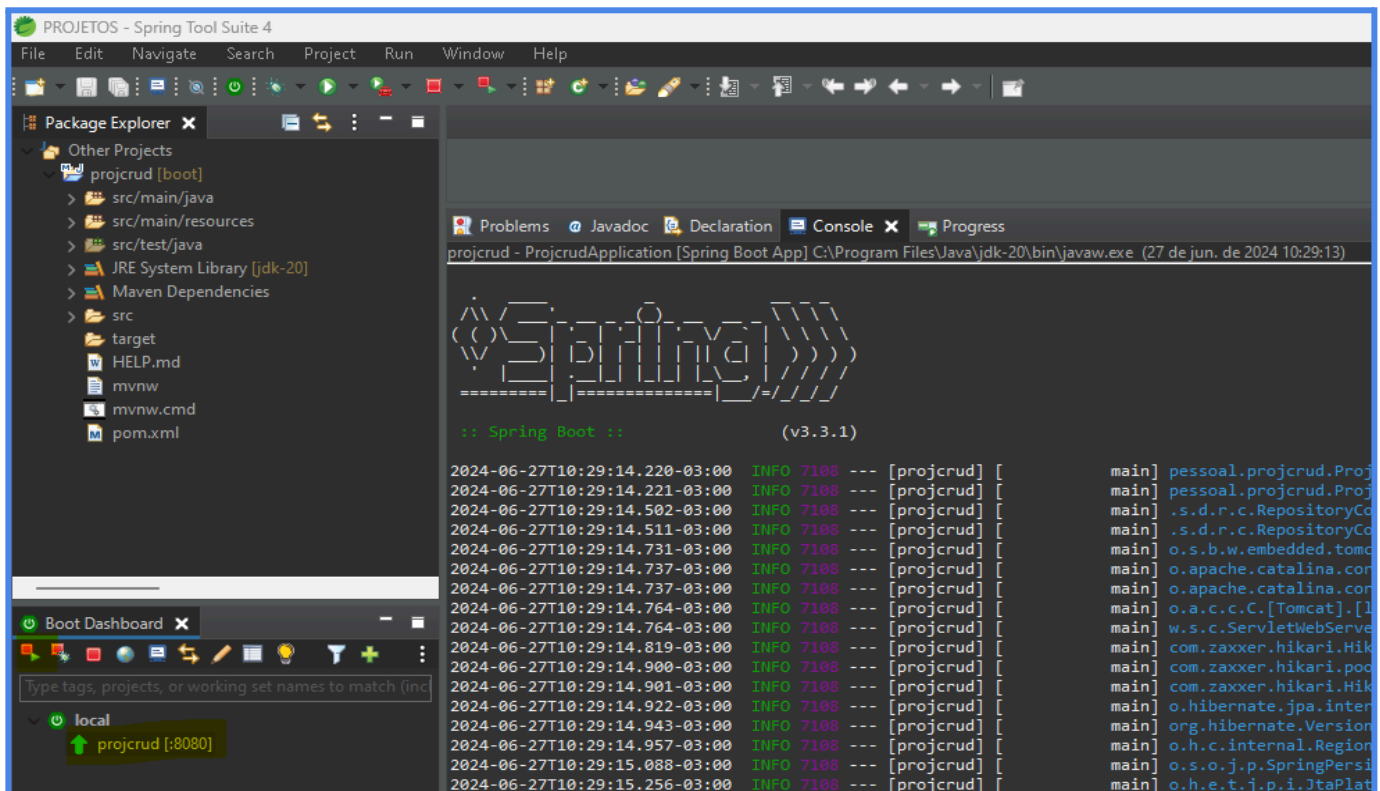


## IMPORTAR O PROJETO PARA O STS



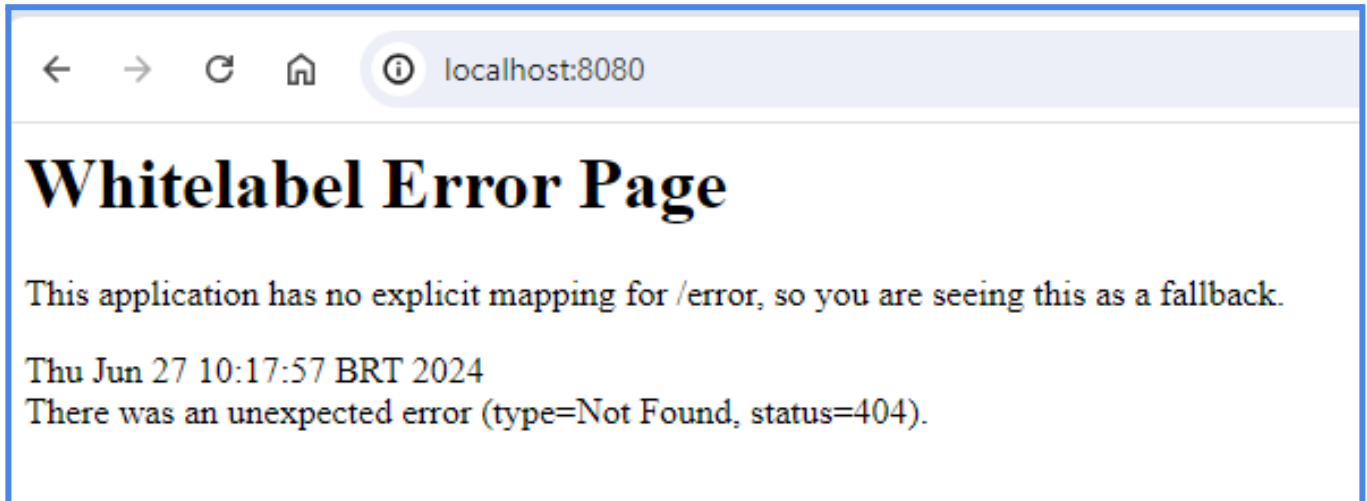


- Rodar o projeto



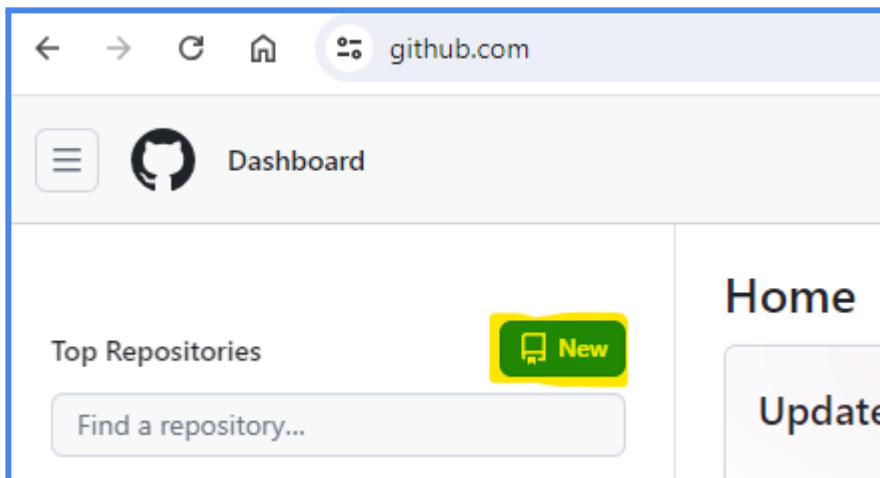
## TESTAR NO BROWSER

- URL: <http://localhost:8080/>



## VERSIONAMENTO



### Criar o projeto no Github



### Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Required fields are marked with an asterisk (\*).

Owner \*  auriceliof / Repository name \*    
  pessoal-ProjCrudBackFront is available.

Great repository names are short and memorable. Need inspiration? How about [fuzzy-octo-giggle](#) ?

Description (optional)

☒  **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)



Choose a license

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

Create repository


#### Quick setup — if you've done this kind of thing before

 Set up in Desktop or ☐ HTTPS ☐ SSH  

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


#### ...or create a new repository on the command line

```
echo "# pessoal-ProjCrudBackFront" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/auriceliof/pessoal-ProjCrudBackFront.git
git push -u origin main
```



#### ...or push an existing repository from the command line

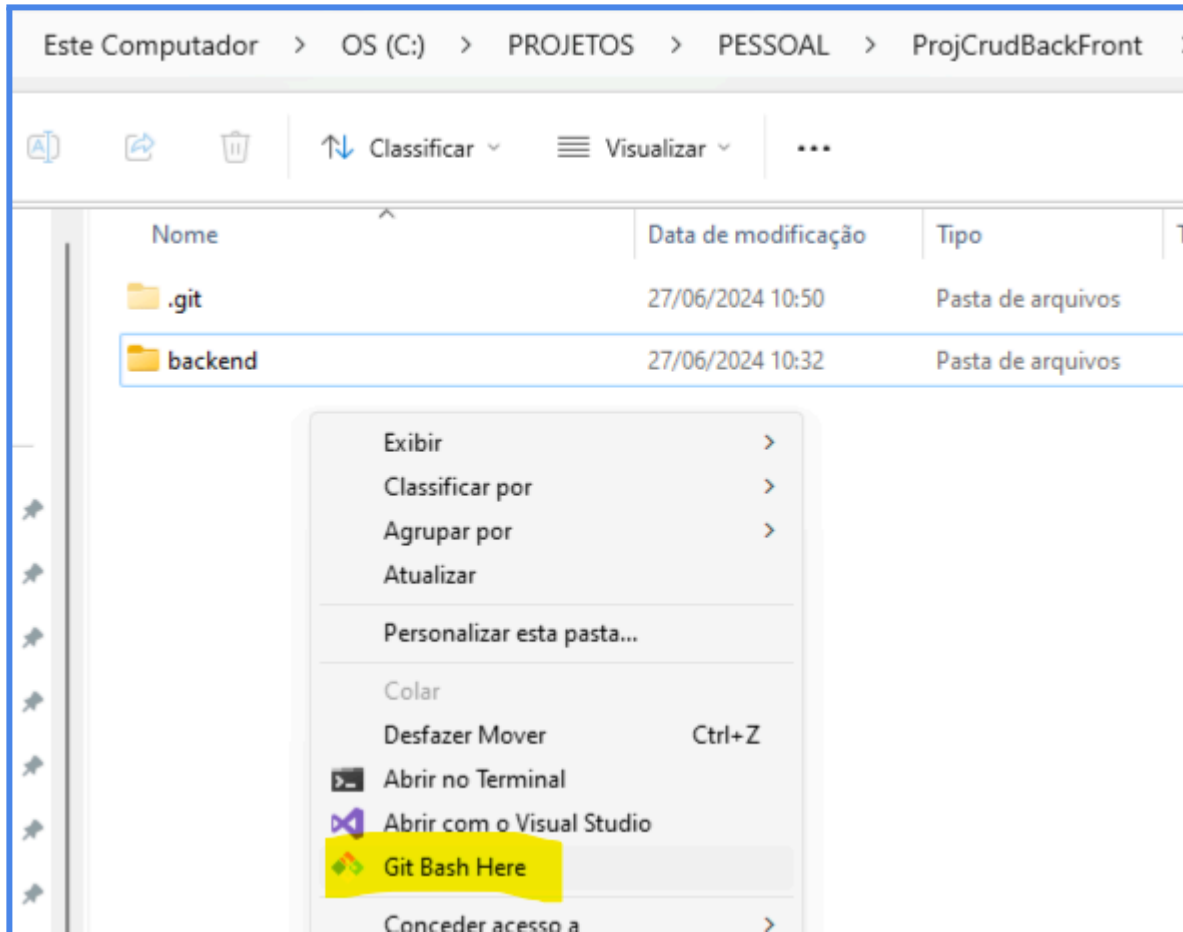
```
git remote add origin https://github.com/auriceliof/pessoal-ProjCrudBackFront.git
git branch -M main
git push -u origin main
```



### Sincronizar com o projeto local via git

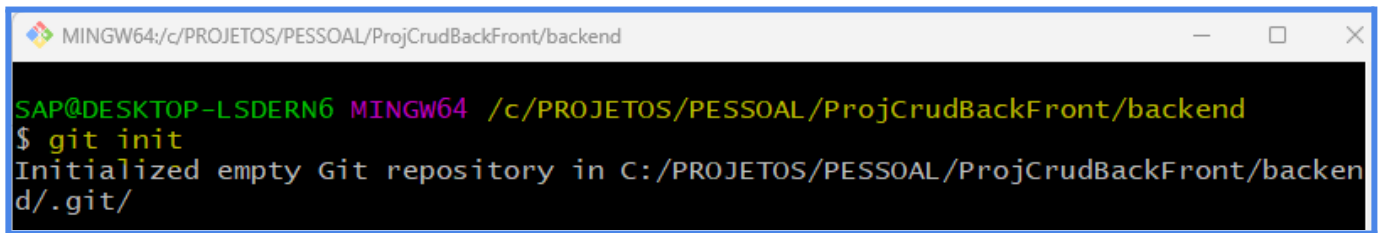
NOTA: O Git deve estar instalado no computador. Caso não tenha ainda, baixe a versão mais atual e instale, dando next até o final.

- Botão direito no projeto e clicar em “Open Git bash here”



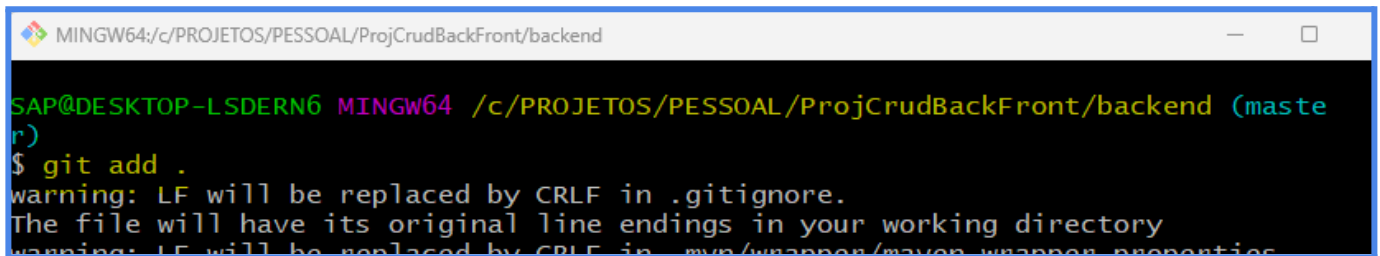
- Efetuar os comando abaixo

- git init



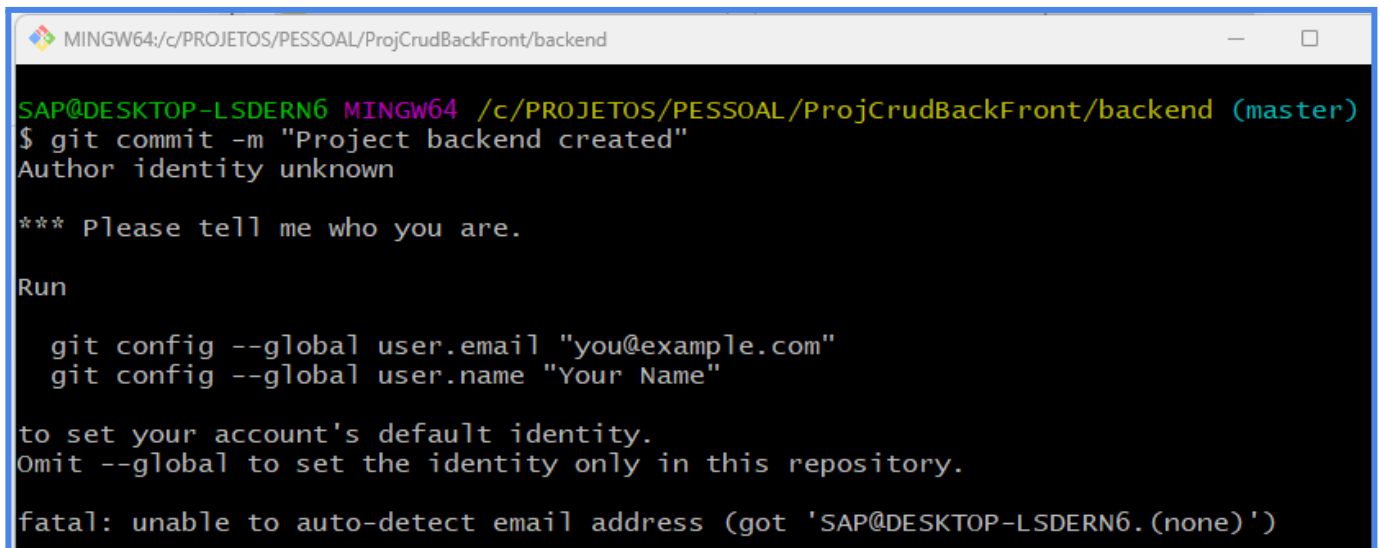
```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront/backend
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront/backend
$ git init
Initialized empty Git repository in C:/PROJETOS/PESSOAL/ProjCrudBackFront/backend/.git/
```

- git add .



```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront/backend
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront/backend (master)
$ git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in mvn/wrapper/maven-wrapper.properties
```

- git commit -m "Project backend created"



```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront/backend
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront/backend (master)
$ git commit -m "Project backend created"
Author identity unknown

*** please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'SAP@DESKTOP-LSDERN6.(none)')
```

NOTA: Caso seja a primeira vez que sincronize no github no computador atual, irá pedir para informar o email e o nome da conta GitHub. Na próxima vez isso não irá aparecer.

- git config --global user.email "auricelio.suporte@hotmail.com"



```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront/backend

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront/backend (master)
$ git config --global user.email "auricelio.suporte@hotmail.com"
```

- git config --global user.name "auriceliof"

```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront/backend

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront/backend (master)
$ git config --global user.name "auriceliof"
```

- git commit -m "Project backend created"

```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront/backend

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront/backend (master)
$ git commit -m "Project backend created"
[master (root-commit) c3b2f7a] Project backend created
 8 files changed, 555 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 .mvn/wrapper/maven-wrapper.properties
 create mode 100644 mvnw
 create mode 100644 mvnw.cmd
 create mode 100644 pom.xml
 create mode 100644 src/main/java/pessoal/projcrud/ProjcrudApplication.java
 create mode 100644 src/main/resources/application.properties
 create mode 100644 src/test/java/pessoal/projcrud/ProjcrudApplicationTests.java
```

- git branch -M main

```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront/backend

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront/backend (master)
$ git branch -M main
```

- git remote add origin https://github.com/auriceliof/pessoal-ProjCrudBackFront.git

```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront/backend

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront/backend (main)
$ git remote add origin https://github.com/auriceliof/pessoal-ProjCrudBackFront.git
```

- `git push -u origin main`

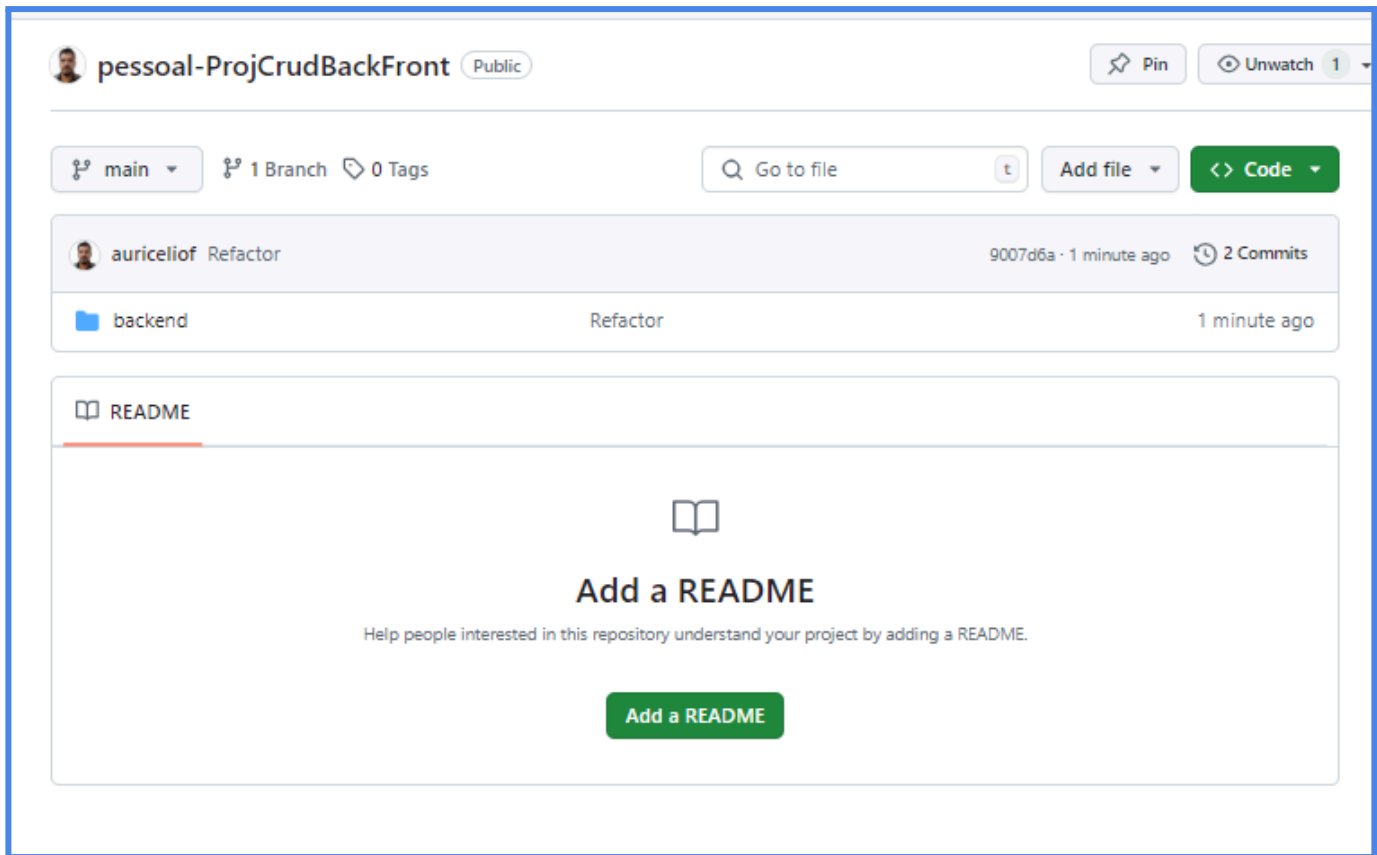
NOTA: Caso seja a primeira vez que sincronize no github no computador atual, irá pedir para realizar a autenticação no GitHub. Na próxima vez isso não irá aparecer.



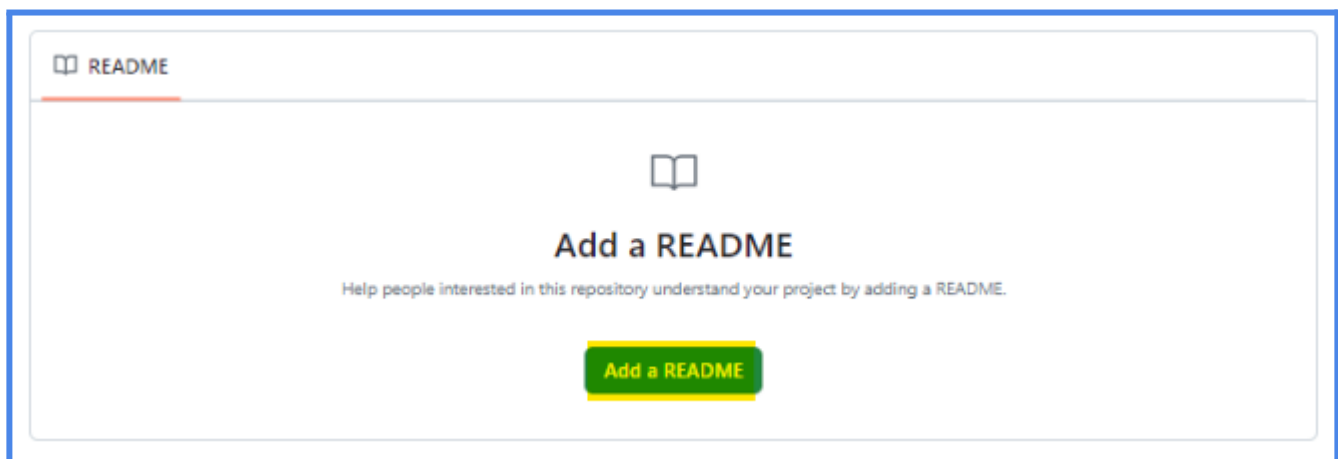
```
MINGW64; c:/PROJETOS/PESSOAL/ProjCrudBackFront/backend
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront/backend (main)
$ git push -u origin main


Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 28 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (22/22), 8.43 KiB | 8.43 MiB/s, done.
Total 22 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/auriceliof/pessoal-ProjCrudBackFront.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

- Visualizar no GitHub (F5)



### Adicionar o README ao projeto



 pessoal-ProjCrudBackFront / README.md in main

Cancel changes

Commit changes...

Edit

Preview

Spaces 2 No wrap

```
1 PROJETO CRUD RESTFUL COM FRONTEND E BACKEND (COM SWAGGER)
2
3 AUTHOR: AURICELIO FREITAS
4
5 DATA: 27/06/2024
6
7 -----
8 BACKEND
9
10 - DEPENDÊNCIAS
11
12     O projeto será implementado conforme abaixo:
13
14     IDE: Spring Tool Suite 4 (STS)
15
16     Banco de Dados: H2
17
18     Gerenciador de Dependências: Maven
19
20     Linguagem: Java
21
22     Versionamento: Git e GitHub
```

Commit changes

Commit message

Create README.md

Extended description

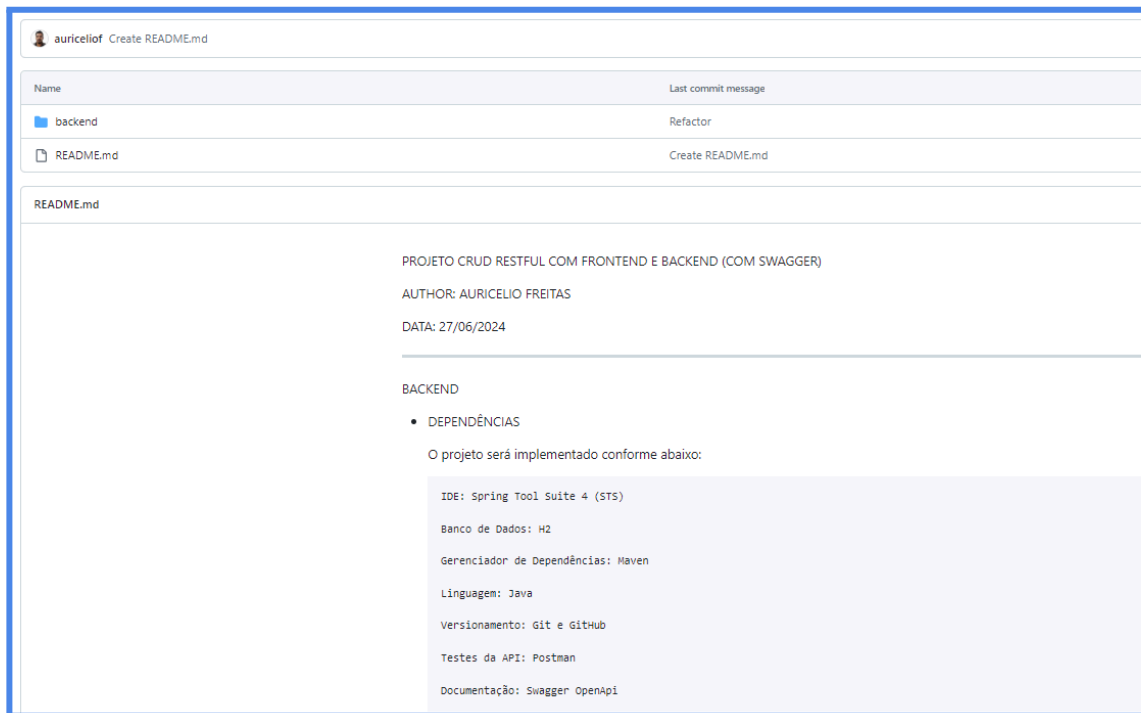
Add an optional extended description..

☒ Commit directly to the main branch

☐ Create a new branch for this commit and start a pull request [Learn more about pull requests](#)

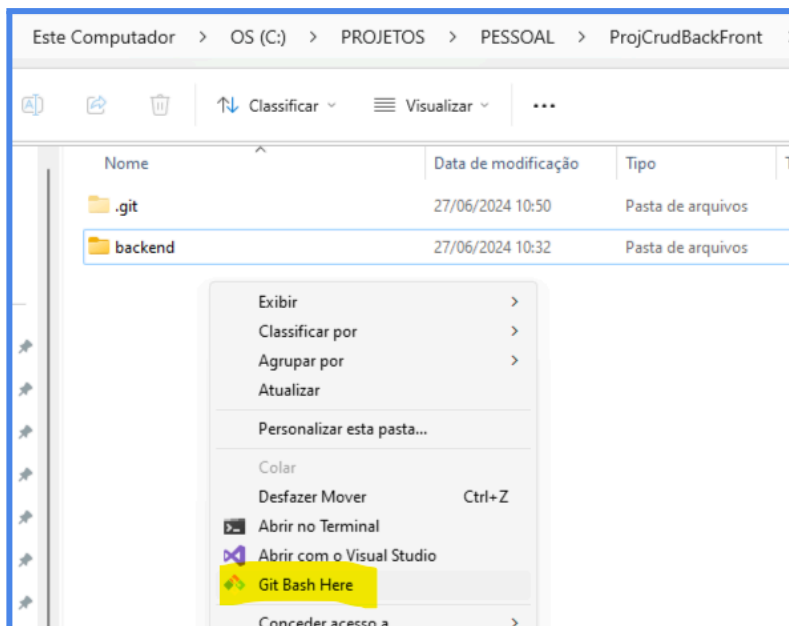
Cancel

Commit changes

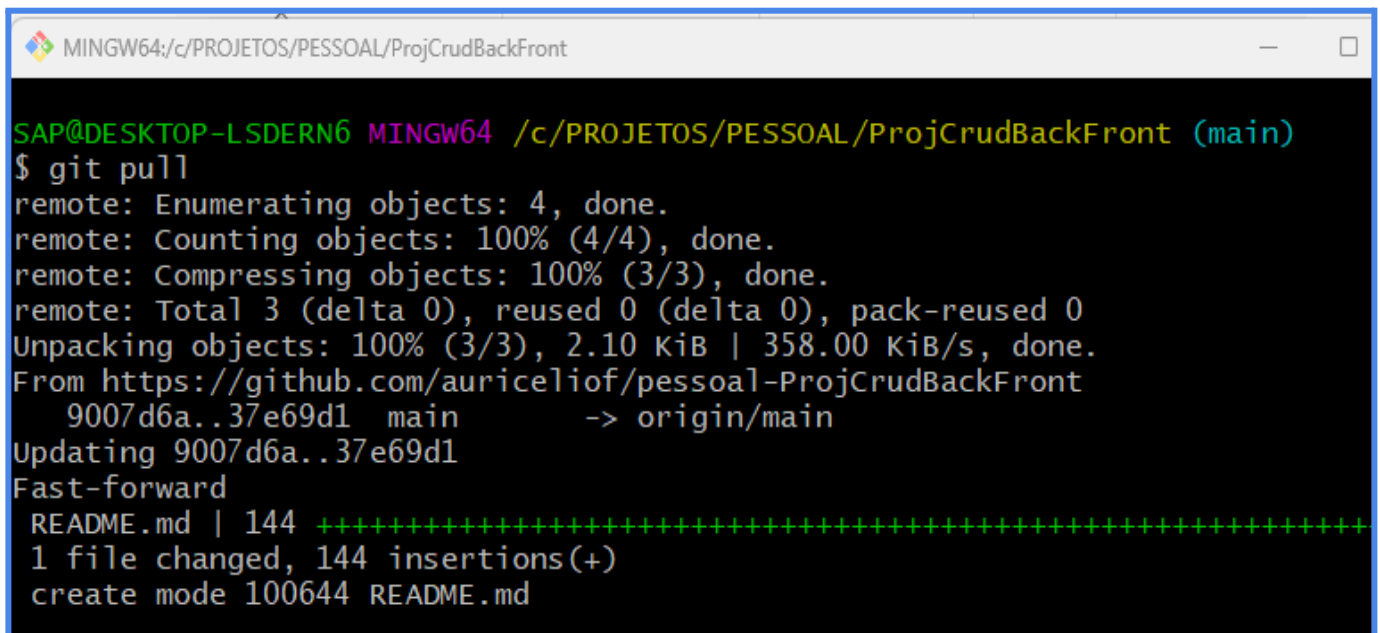


### Atualizar o projeto local

- Botão direito no projeto e clicar em “Open Git bash here”



- git pull

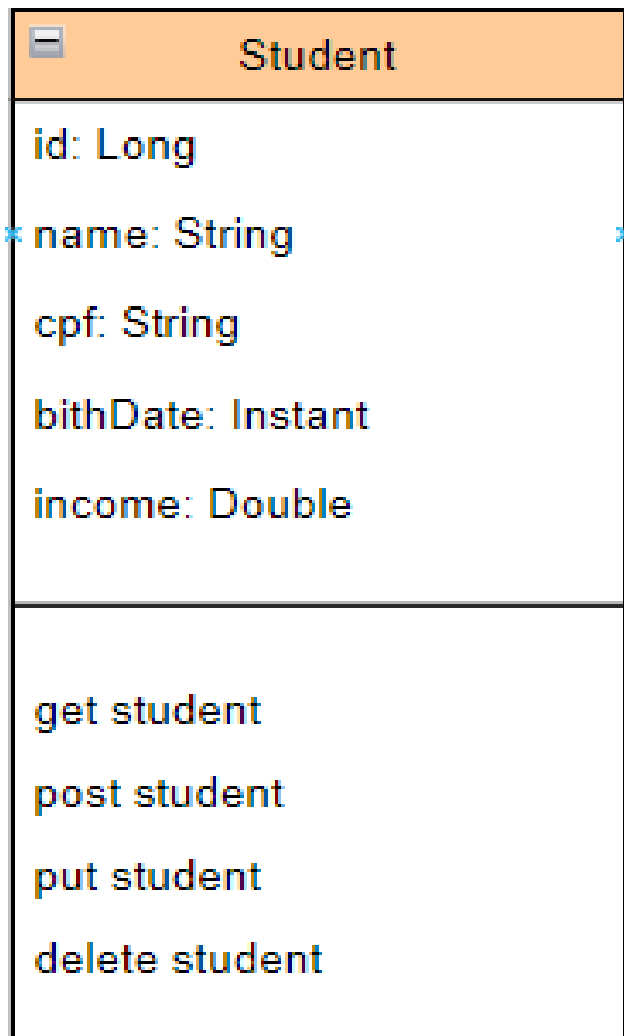
A screenshot of a Windows terminal window titled 'MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront'. The terminal shows the execution of the 'git pull' command. The output indicates that 4 objects were enumerated, 4/4 counted, and 3 objects were compressed. The total size of the update is 2.10 KiB. The update is a fast-forward from the origin/main branch. The file 'README.md' is updated with 144 insertions, creating a new mode 100644.

```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 2.10 KiB | 358.00 KiB/s, done.
From https://github.com/auriceliof/pessoal-ProjCrudBackFront
  9007d6a..37e69d1  main      -> origin/main
Updating 9007d6a..37e69d1
Fast-forward
 README.md | 144 +++++
 1 file changed, 144 insertions(+)
 create mode 100644 README.md
```

## INICIAR O DESENVOLVIMENTO DO PROJETO

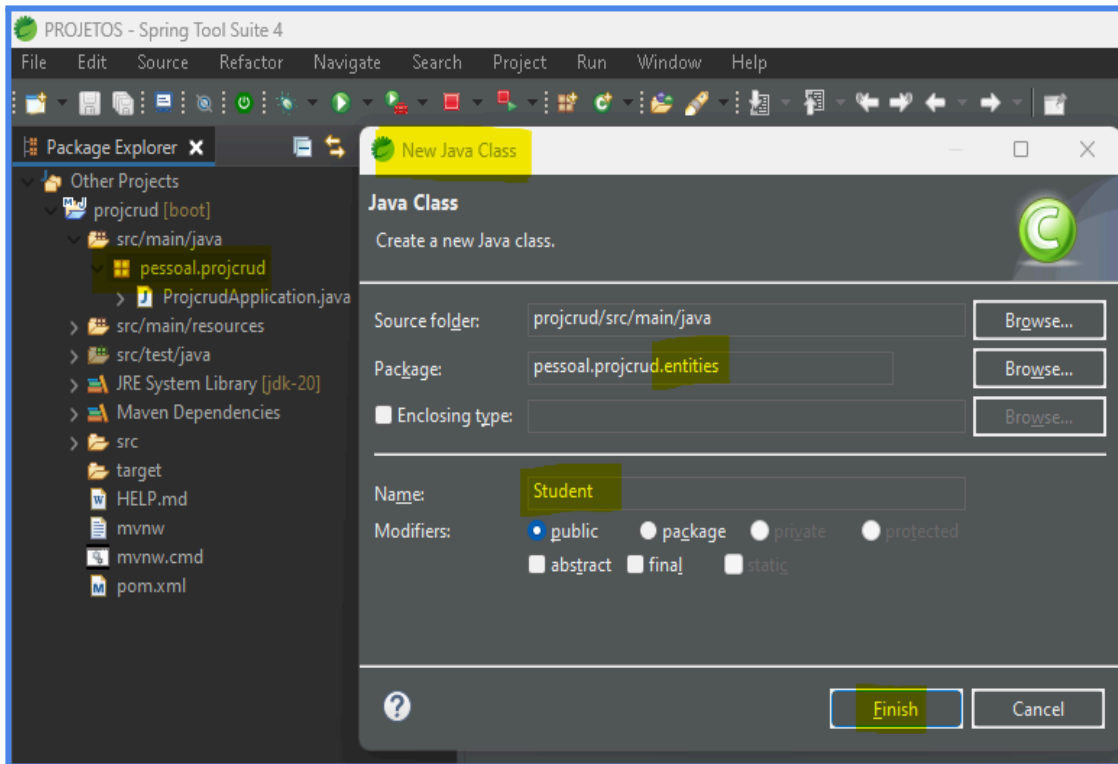
### STUDENT\_CLASSE

#### DIAGRAMA DE CLASSE

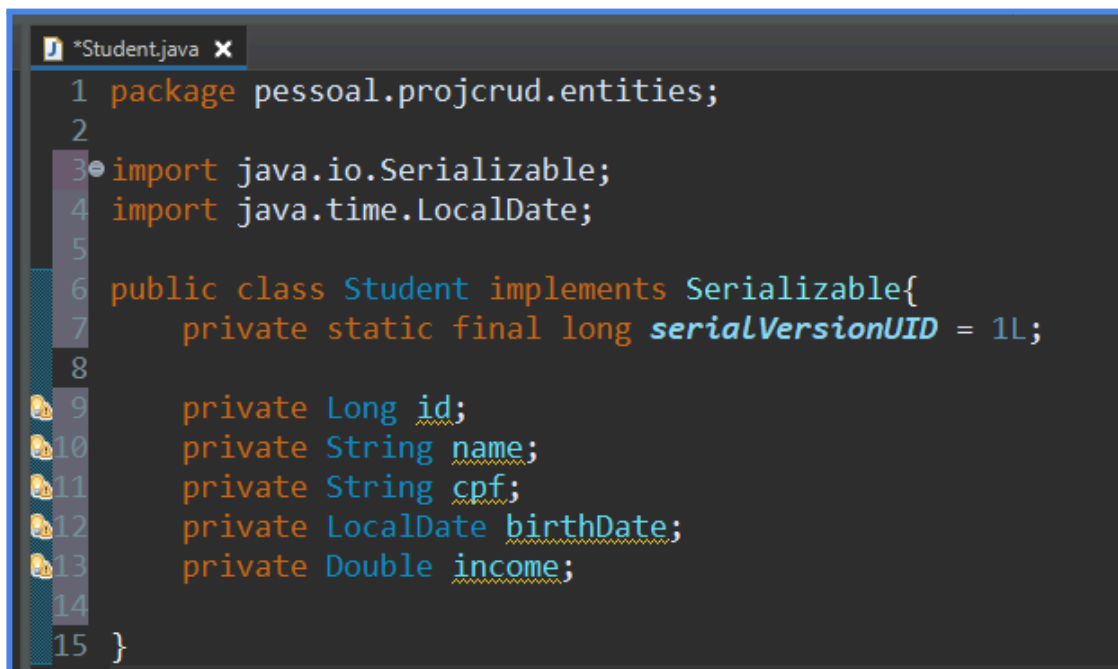


## IMPLEMENTAR A ESTRUTURA E A CLASSE STUDENT NO STS

### Criar a estrutura e classe



### Definir o Serializable e os atributos da Classe

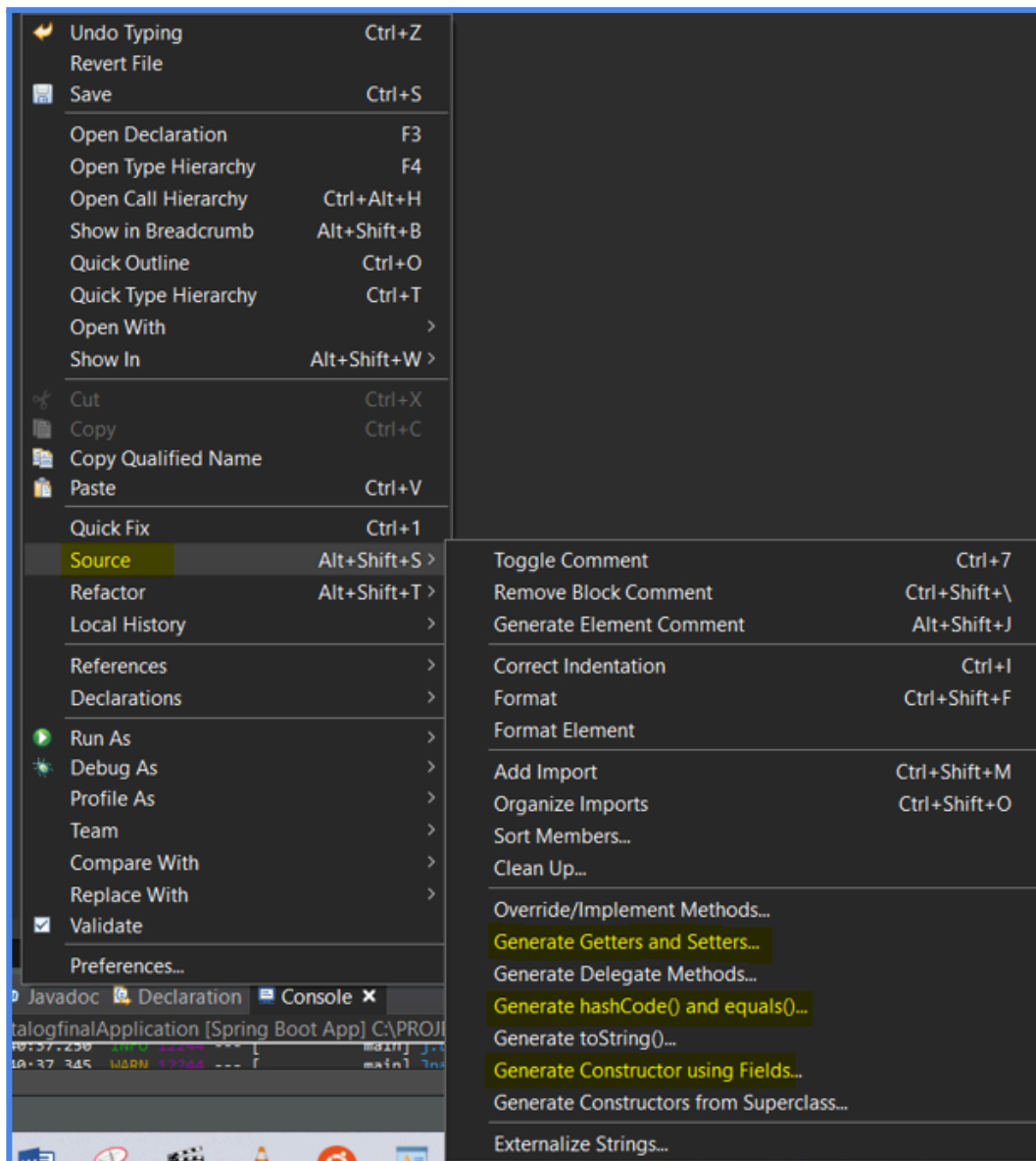




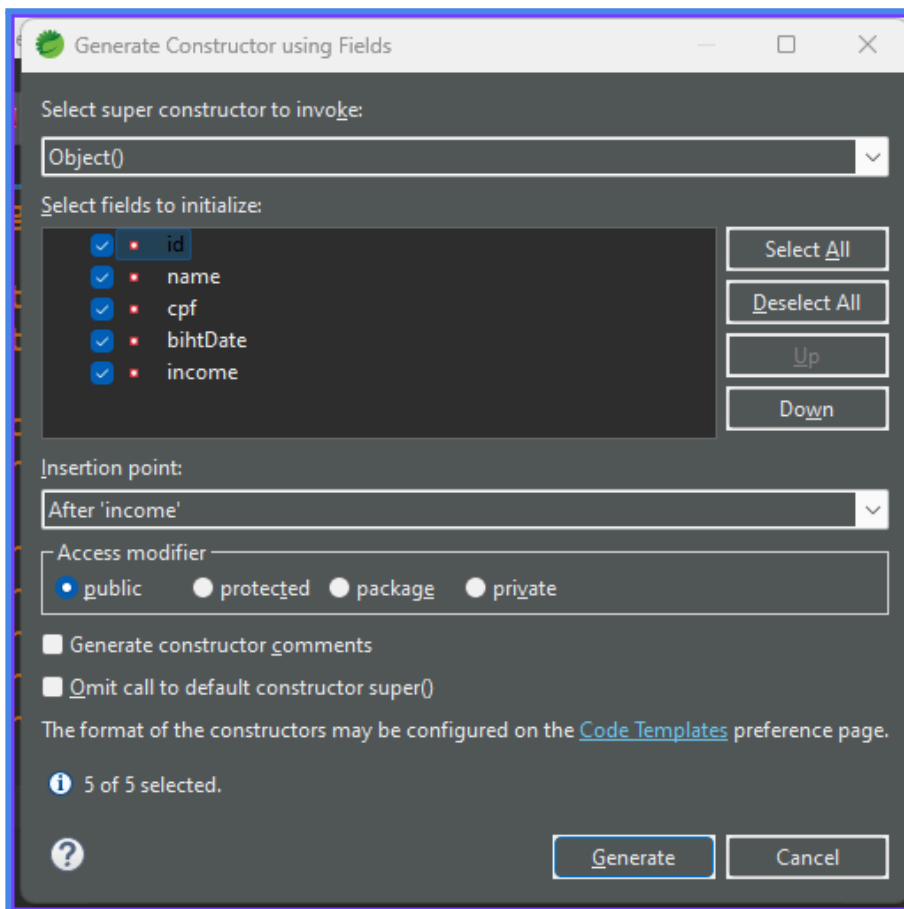
**DICA:**

Para criar os construtores, getters and setters e o hashCode, proceder conforme segue:

- Clicar com o botão direito aonde quer inserir
  - Source
    - “Escolher o método que deseja implementar”
      - Selecionar os itens
        - Clicar em: Generate



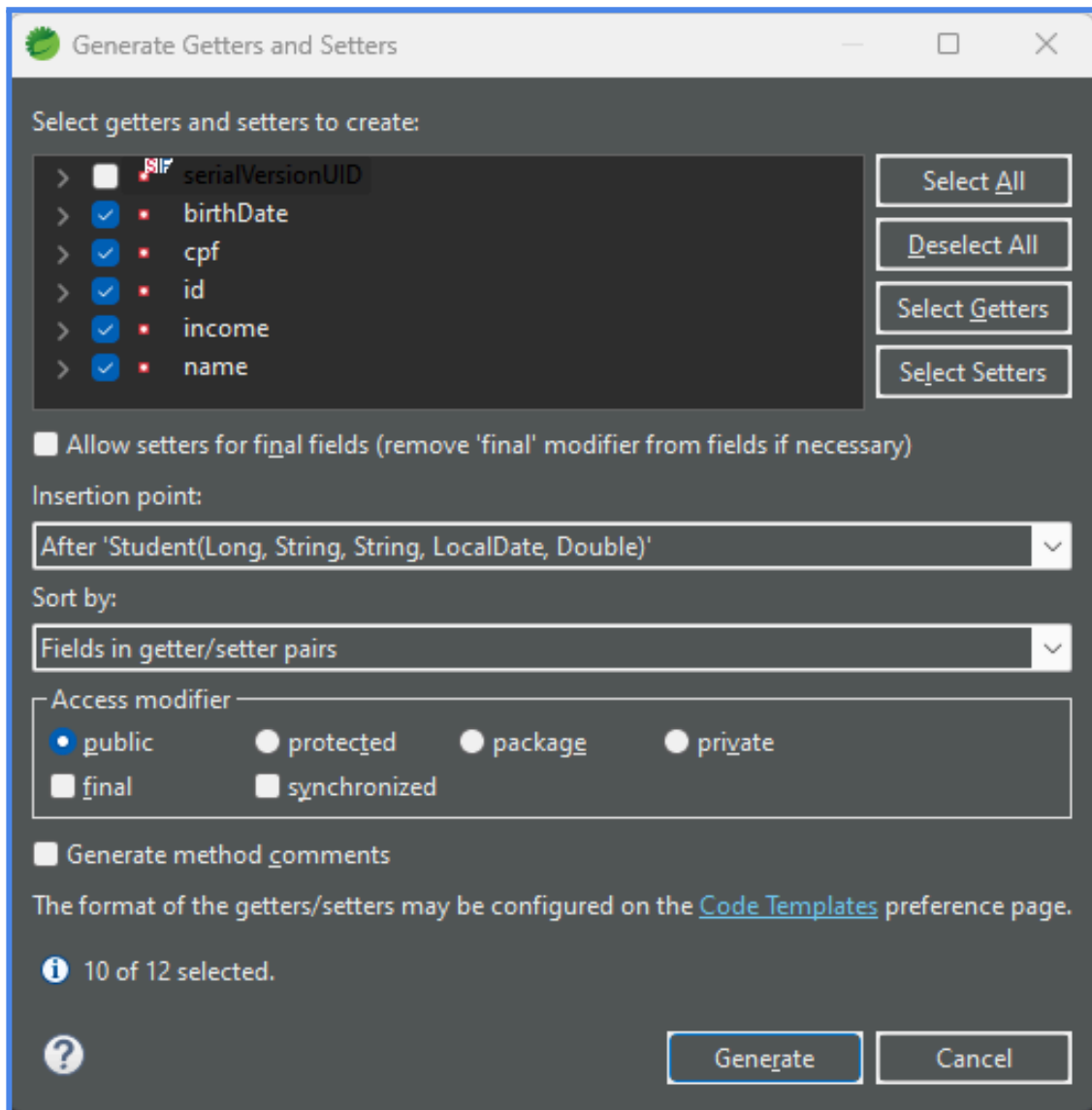
## Criar o construtor



```
14
15 public Student() {
16
17 }
18
19 public Student(Long id, String name, String cpf, LocalDate birthDate, Double income) {
20     super();
21     this.id = id;
22     this.name = name;
23     this.cpf = cpf;
24     this.birthDate = birthDate;
25     this.income = income;
26 }
27 }
```

*NOTA: Por melhores práticas, também criaremos um construtor vazio.*

## Criar os Getters and Setters



Generate Getters and Setters

Select getters and setters to create:

- ☐ serialVersionUID
- ☒ birthDate
- ☒ cpf
- ☒ id
- ☒ income
- ☒ name

☐ Allow setters for final fields (remove 'final' modifier from fields if necessary)

Insertion point:  
After 'Student(Long, String, String, LocalDate, Double)'

Sort by:  
Fields in getter/setter pairs

Access modifier

☒ public ☐ protected ☐ package ☐ private

☐ final ☐ synchronized

☐ Generate method comments

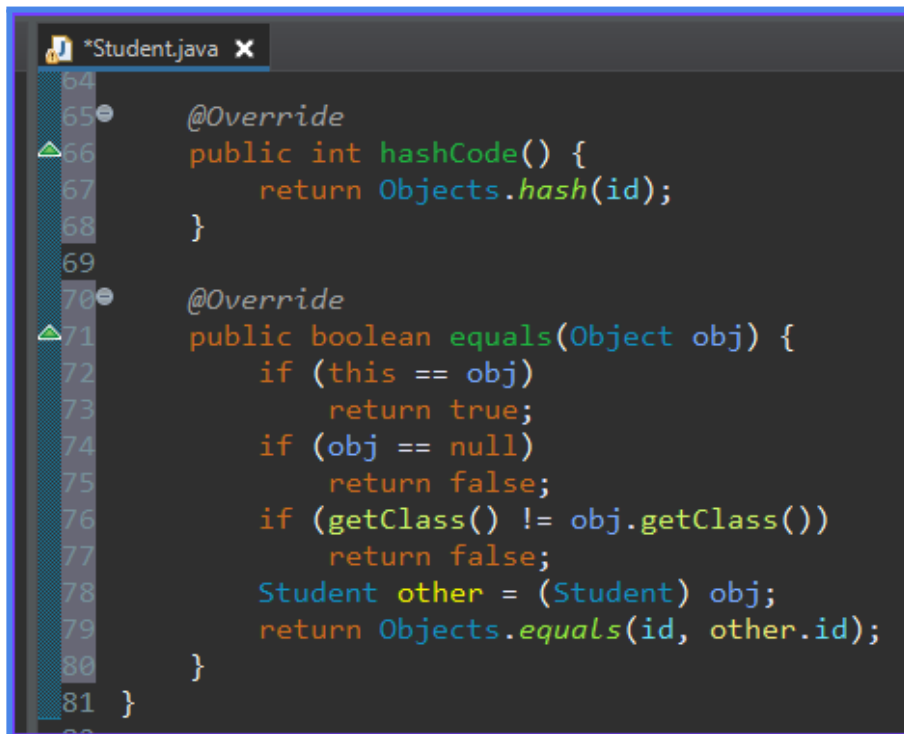
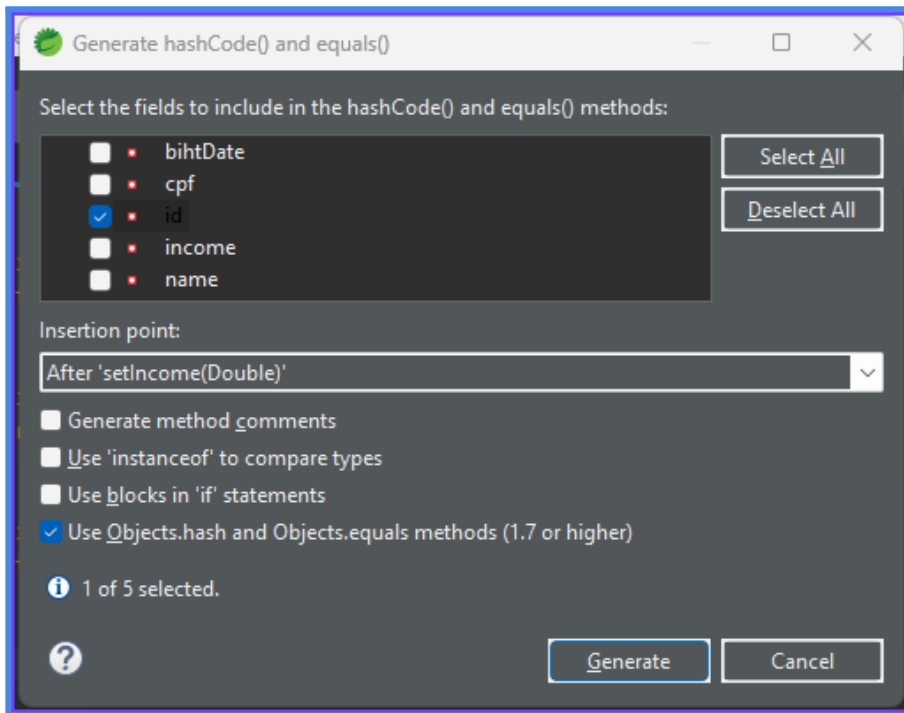
The format of the getters/setters may be configured on the [Code Templates](#) preference page.

**i** 10 of 12 selected.

```
*Student.java X
23
24 public Long getId() {
25     return id;
26 }
27
28 public void setId(Long id) {
29     this.id = id;
30 }
31
32 public String getName() {
33     return name;
34 }
35
36 public void setName(String name) {
37     this.name = name;
38 }
39
```

```
43
44 public String getCpf() {
45     return cpf;
46 }
47
48 public void setCpf(String cpf) {
49     this.cpf = cpf;
50 }
51
52 public LocalDate getBirthDate() {
53     return birthDate;
54 }
55
56 public void setBirthDate(LocalDate birthDate) {
57     this.birthDate = birthDate;
58 }
59
60 public Double getIncome() {
61     return income;
62 }
63
64 public void setIncome(Double income) {
65     this.income = income;
66 }
67 }
```

## Criar o hashCode e equals




### Github-2


- “Git bash here” no diretório do projeto
  - git add backend
  - git commit -m “Created class Student”
  - git push


```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git add .


SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git commit -m "Created class Student"
[main 8670f80] Created class Student
1 file changed, 99 insertions(+)
create mode 100644 backend/src/main/java/pessoal/projcrud/entities/Student.java

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git push
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 28 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 1.20 KiB | 1.20 MiB/s, done.
Total 10 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/auriceliof/pessoal-ProjCrudBackFront.git
37e69d1..8670f80 main -> main
```

 **pessoal-ProjCrudBackFront** Public Pin Unwatch 1


 main


 1 Branch


 0 Tags

Go to file


Add file

 Code

 **auriceliof** Created class Student 8670f80 · 1 minute ago 4 Commits

 backend

Created class Student1 minute ago

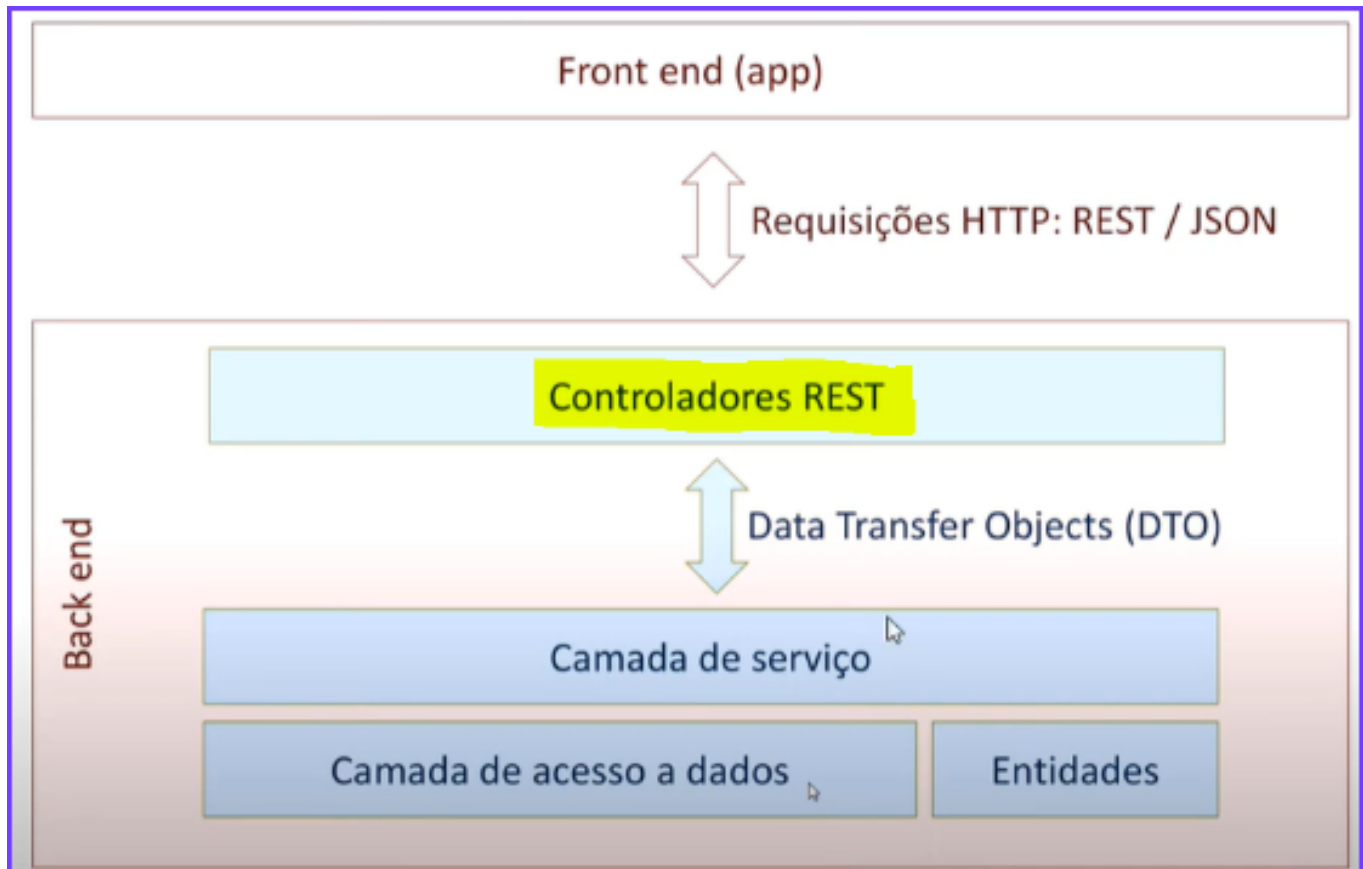
 README.md

Create README.md1 hour ago

## STUDENT\_CONTROLLER

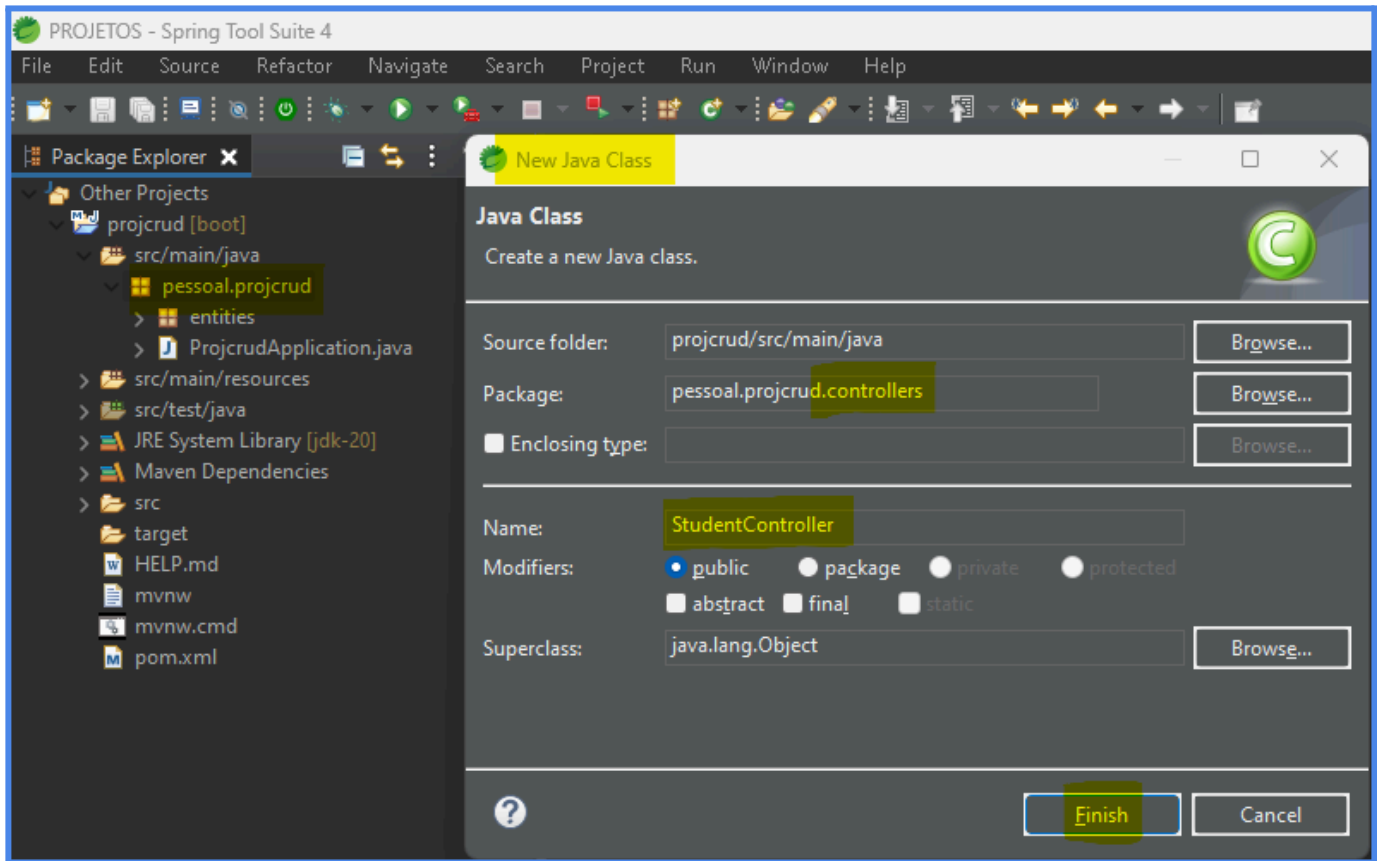
- O controlador é quem gerencia as requisições, podendo ser chamado de Controller ou Resource. É onde implantamos nossos endpoints.

### CONCEITUAL

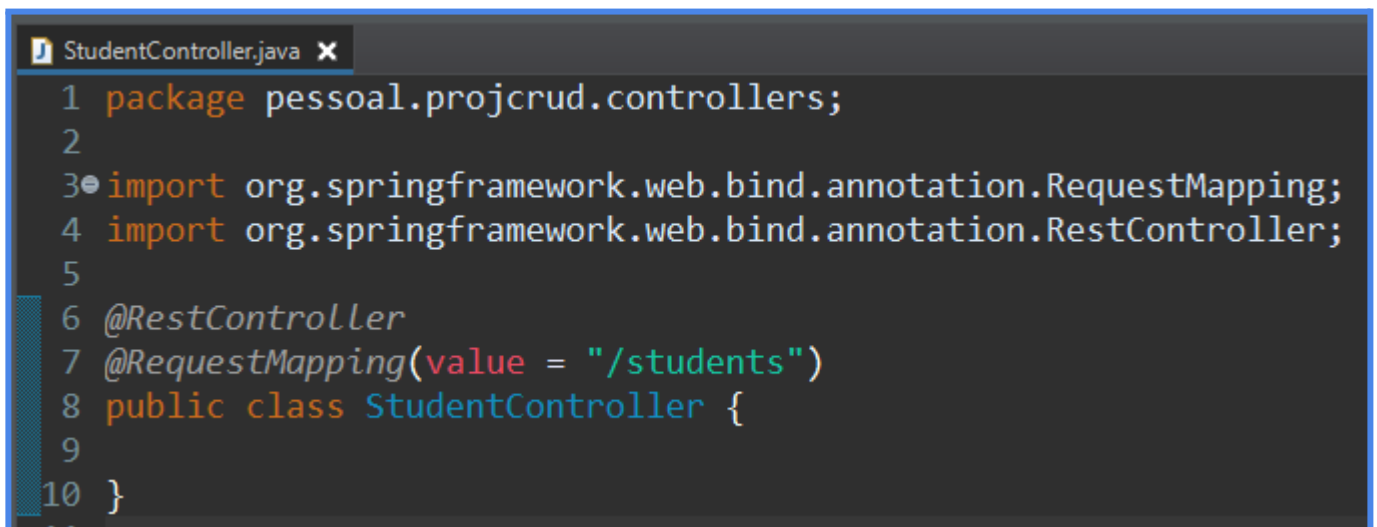


## IMPLEMENTAR A ESTRUTURA

### Criar a estrutura e o recurso StudentController



### Implementar as Notações Rest





## Criar o Endpoint findAll para teste

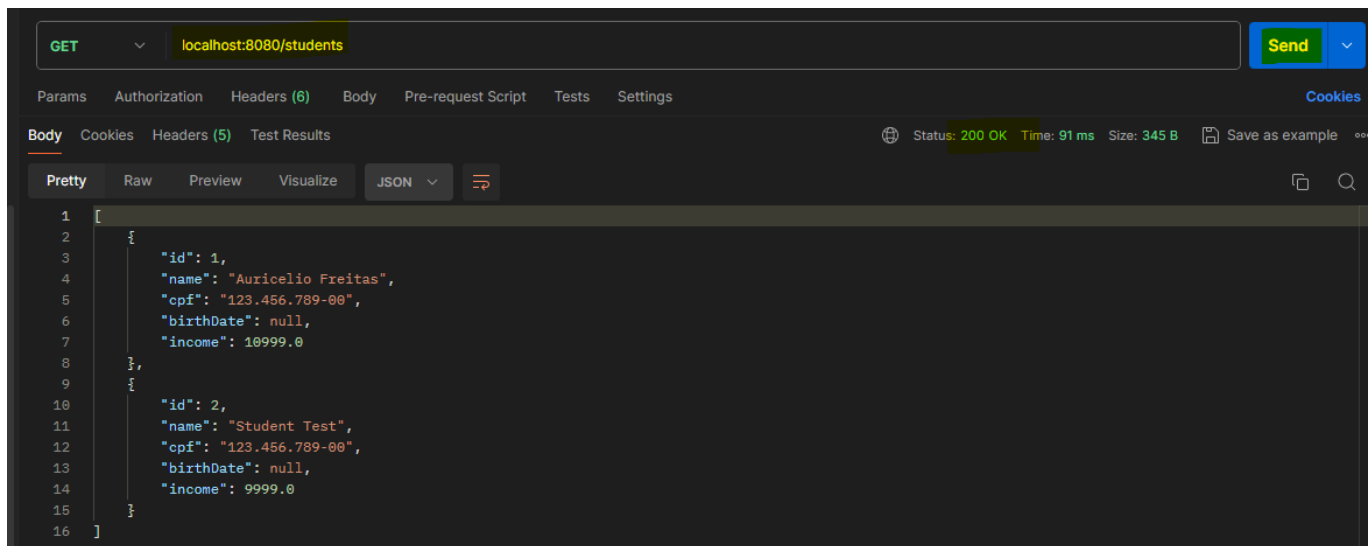
```
StudentController.java x
1 package pessoal.projcrud.controllers;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import org.springframework.http.ResponseEntity;
7 import org.springframework.web.bind.annotation.GetMapping;
8 import org.springframework.web.bind.annotation.RequestMapping;
9 import org.springframework.web.bind.annotation.RestController;
10
11 import pessoal.projcrud.entities.Student;
12
13 @RestController
14 @RequestMapping(value = "/students")
15 public class StudentController {
16
17     @GetMapping
18     public ResponseEntity<List<Student>> findAll() {
19         List<Student> list = new ArrayList<>();
20
21         list.add(new Student(1L, "Auricelio Freitas", "123.456.789-00", null, 10999.0));
22         list.add(new Student(2L, "Student Test", "123.456.789-00", null, 9999.0));
23
24         return ResponseEntity.ok().body(list);
25     }
26 }
27
```

## Rodar o projeto

The screenshot shows the Spring Boot IDE interface. On the left, the 'Boot Dashboard' tab is active, displaying a search bar and a list of running applications. The 'local' environment is selected, and the 'projcrud [:8080]' application is shown as running. On the right, the 'Console' tab is active, showing the application's output. The output consists of four log entries, all of which are 'INFO' messages with the PID '27136'. The timestamps are 2024-06-27T13:25:37.651-03:00, 2024-06-27T13:25:40.753-03:00, 2024-06-27T13:25:40.753-03:00, and 2024-06-27T13:25:40.754-03:00.

```
projcrud - ProjcrudApplication [Spring Boot App] C:\Program Files\Java\j
2024-06-27T13:25:37.651-03:00 INFO 27136 -
2024-06-27T13:25:40.753-03:00 INFO 27136 -
2024-06-27T13:25:40.753-03:00 INFO 27136 -
2024-06-27T13:25:40.754-03:00 INFO 27136 -
```

## Testar com o Postman



## Github-3

- “Git bash here” no diretório do projeto
  - git add .
  - git commit -m “Created StudentController and the Endpoint findAll”
  - git push

```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git add .

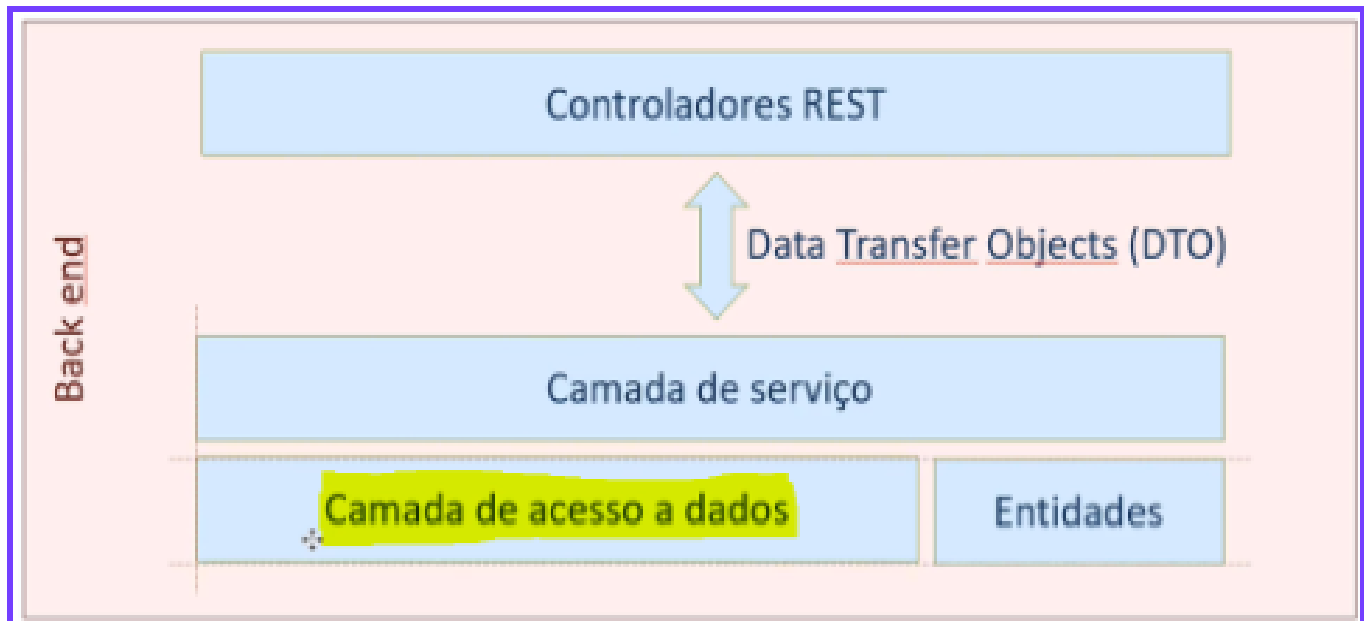
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git commit -m "Created StudentController and the Endpoint findAll"
[main 099ead8] Created StudentController and the Endpoint findAll
1 file changed, 26 insertions(+)
create mode 100644 backend/src/main/java/pessoal/projcrud/controllers/StudentController.java

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git push
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 28 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 1.11 KiB | 1.11 MiB/s, done.
Total 10 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/auriceliof/pessoal-ProjCrudBackFront.git
8670f80..099ead8 main -> main
```

## STUDENT\_REPOSITORY

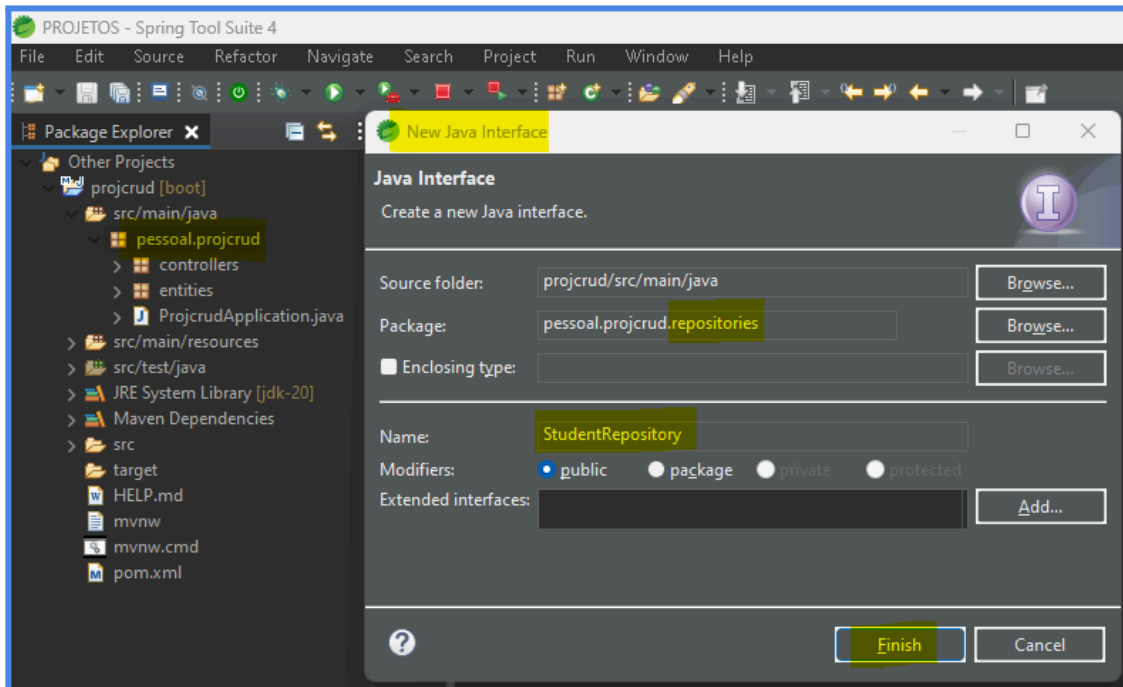
- Iremos implementar a “Camada de acesso a dados”, chamada de Repositories. É a camada responsável pelo acesso ao banco.

### CONCEITUAL

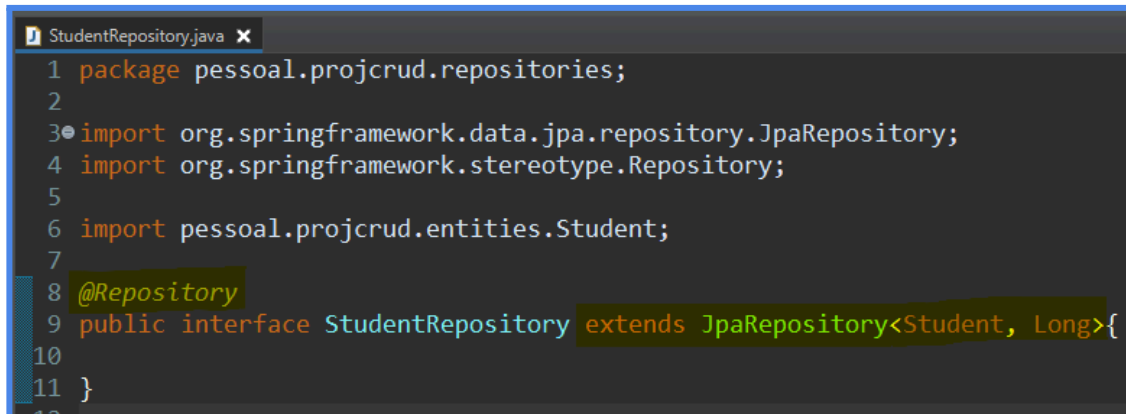


## IMPLEMENTAR A ESTRUTURA E O STUDENT\_REPOSITORY

### Criar a estrutura e a interface de acesso ao banco



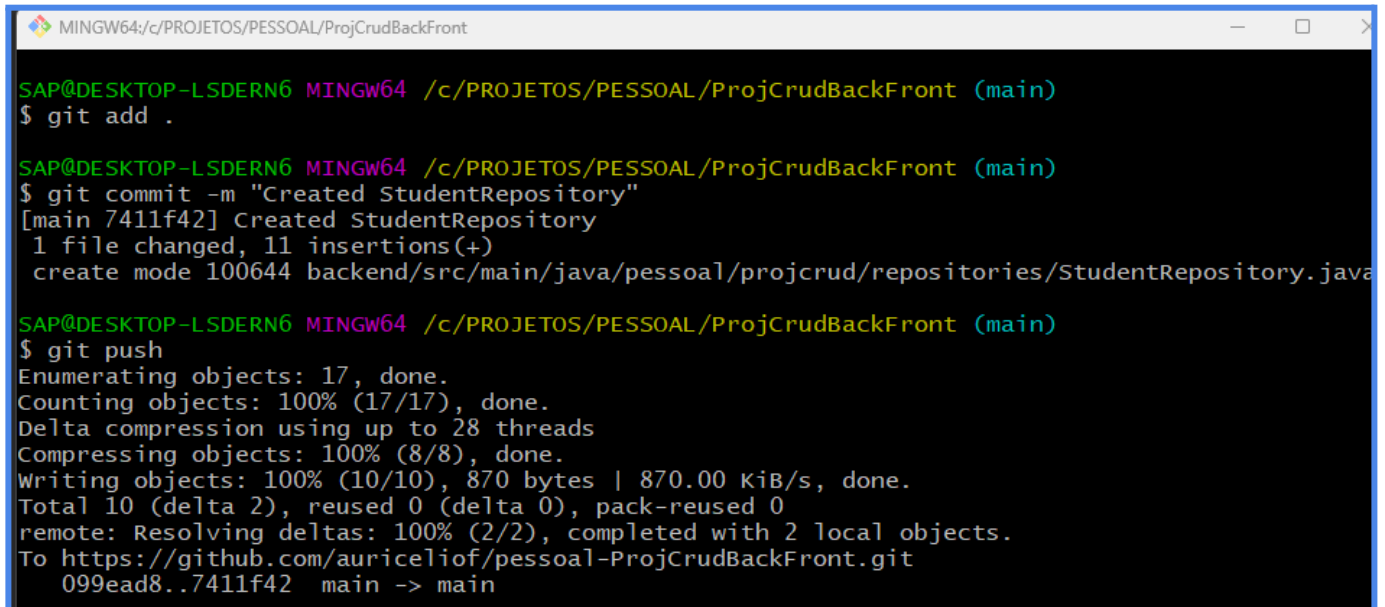
### Implementar a notação e estender a JPA



NOTA: Apenas ao estender o *JpaRepository*, o spring já nos fornece vários métodos de manipulação aos dados com o banco.

### Github-4

- “Git bash here” no diretório do projeto
  - git add backend
  - git commit -m “Created StudentRepository”
  - git push



```
MINGW64: c:/PROJETOS/PESSOAL/ProjCrudBackFront
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git add .

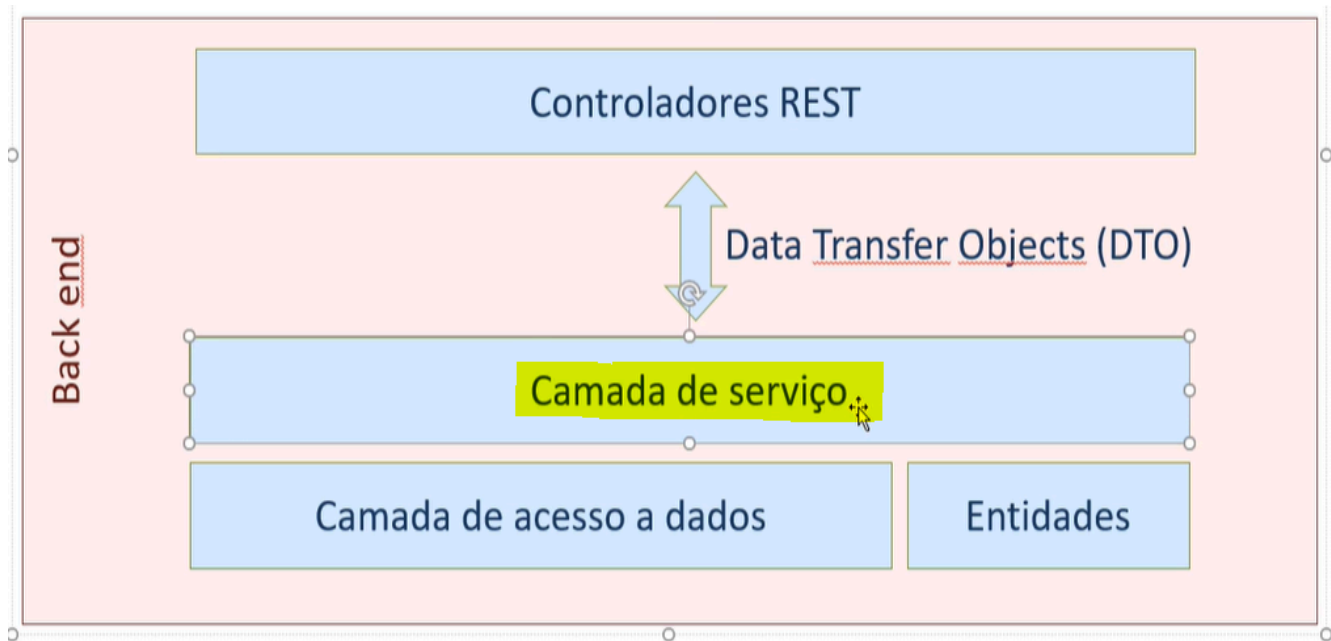
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git commit -m "Created StudentRepository"
[main 7411f42] Created StudentRepository
1 file changed, 11 insertions(+)
create mode 100644 backend/src/main/java/pessoal/projcrud/repositories/StudentRepository.java

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git push
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 28 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 870 bytes | 870.00 KiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/auriceliof/pessoal-ProjCrudBackFront.git
099ead8..7411f42 main -> main
```

## STUDENT\_SERVICE

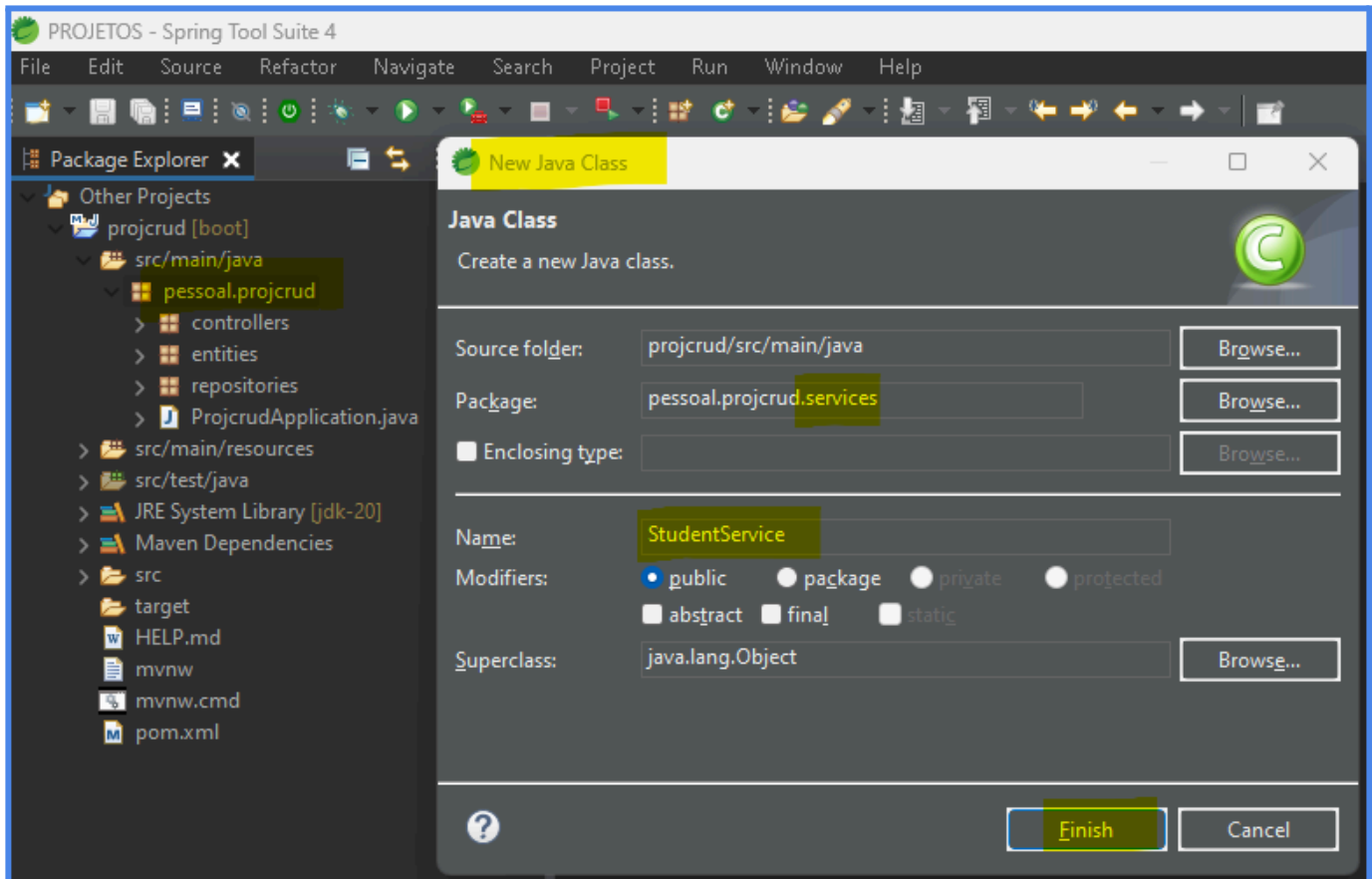
- Iremos implementar a “Camada de serviço”, onde concentramos as regras de negócio e toda a lógica do projeto.

### CONCEITUAL



## IMPLEMENTAR A ESTRUTURA E O STUDENT\_SERVICE

### Criar a estrutura e a classe de serviço



## Implementar a lógica para o endpoint findAll

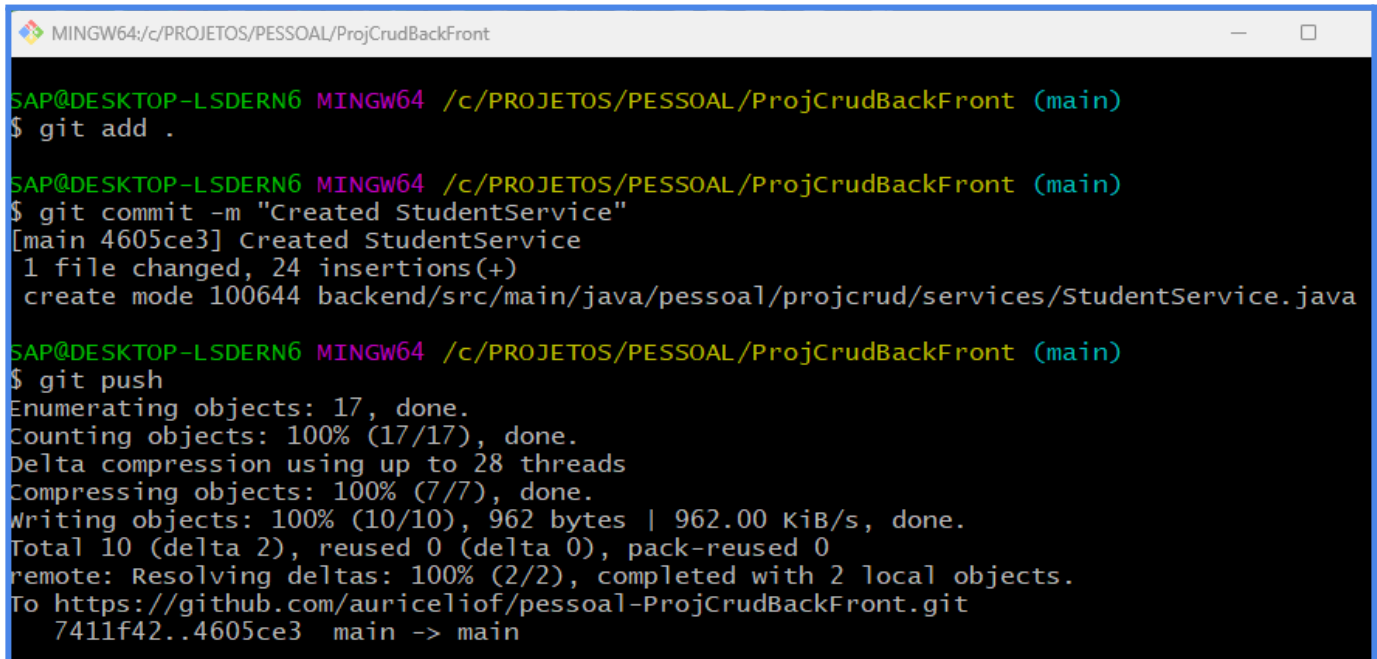
```
StudentService.java x
1 package pessoal.projcrud.services;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Service;
7 import org.springframework.transaction.annotation.Transactional;
8
9 import pessoal.projcrud.entities.Student;
10 import pessoal.projcrud.repositories.StudentRepository;
11
12 @Service
13 public class StudentService {
14
15     @Autowired
16     private StudentRepository repository;
17
18     @Transactional(readonly = true)
19     public List<Student> findAll() {
20
21         return repository.findAll();
22     }
23 }
```

NOTA: O “@Transactional”, garante a integridade da transação de um método junto ao banco. No caso de pesquisa, utilizar o “readonly” para evitar o locking desnecessário ao banco.



## Github-5

- “Git bash here” no diretório do projeto
  - git add backend
  - git commit -m “Created StudentService”
  - git push



```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git add .

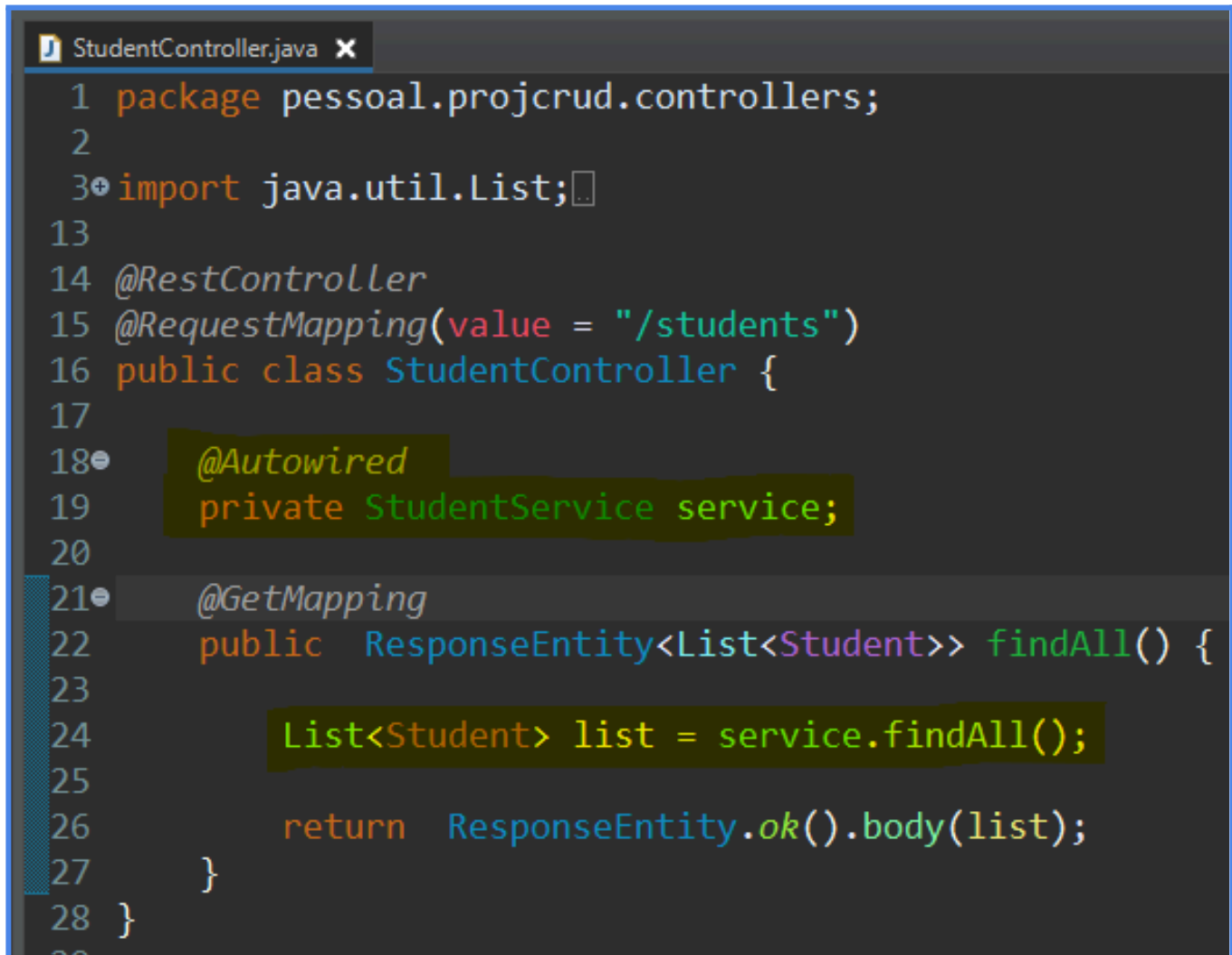
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git commit -m "Created StudentService"
[main 4605ce3] Created StudentService
1 file changed, 24 insertions(+)
create mode 100644 backend/src/main/java/pessoal/projcrud/services/StudentService.java

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git push
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 28 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 962 bytes | 962.00 KiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/auriceliof/pessoal-ProjCrudBackFront.git
7411f42..4605ce3 main -> main
```

## INTEGRAÇÃO COM O BANCO

### AJUSTAR AS CAMADAS

#### Implementar o StudentController



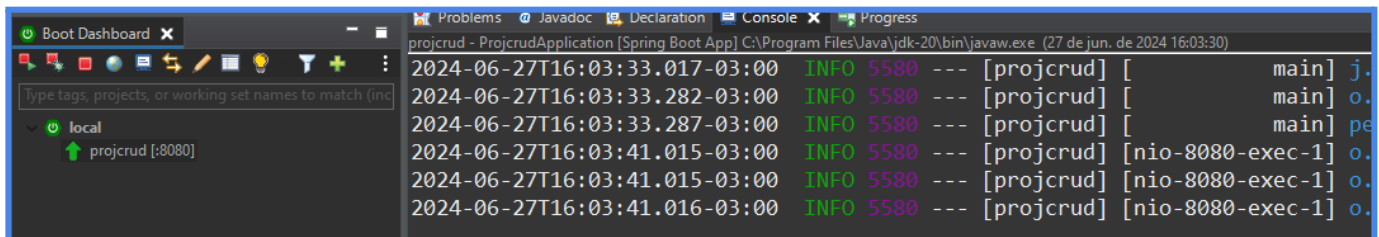
```
StudentController.java x
1 package pessoal.projcrud.controllers;
2
3 import java.util.List;
13
14 @RestController
15 @RequestMapping(value = "/students")
16 public class StudentController {
17
18     @Autowired
19     private StudentService service;
20
21     @GetMapping
22     public ResponseEntity<List<Student>> findAll() {
23
24         List<Student> list = service.findAll();
25
26         return ResponseEntity.ok().body(list);
27     }
28 }
```

## Implementar a classe Student

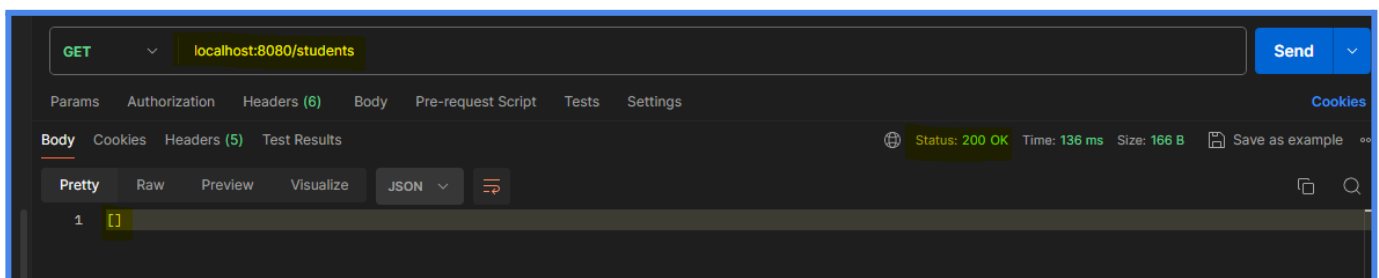
```
Student.java x
13
14 @Entity
15 @Table(name = "tb_student")
16 public class Student implements Serializable{
17     private static final long serialVersionUID = 1L;
18
19     @Id
20     @GeneratedValue(strategy = GenerationType.IDENTITY)
21     private Long id;
22     private String name;
23     private String cpf;
24
25     @Column(columnDefinition = "TIMESTAMP WITH TIME ZONE")
26     private LocalDate birthDate;
27     private Double income;
28 }
```

**NOTA:.** Ao instanciar as notações em questão, sempre escolher o pacote referente à especificação “*jakarta.persistence*”.

## Rodar o projeto



## Testar com o Postman



**NOTA:** Como não configuramos o banco ainda, nos é retornada, apenas, uma lista vazia.

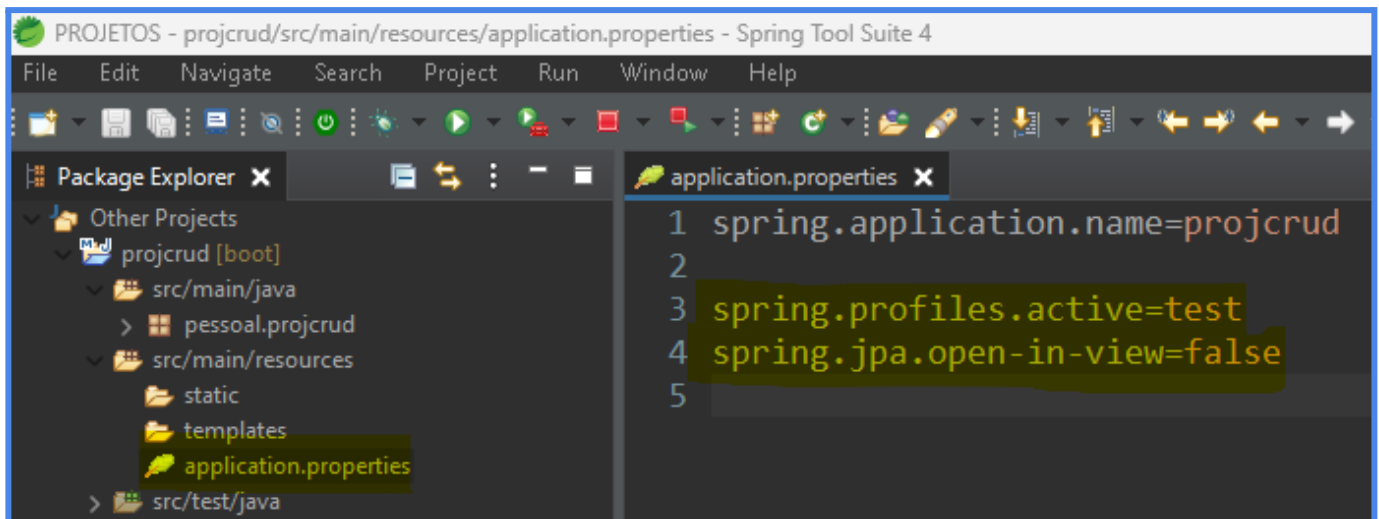
## BANCO H2

### Configurar o perfil de teste no application.properties

#### Perfil de teste

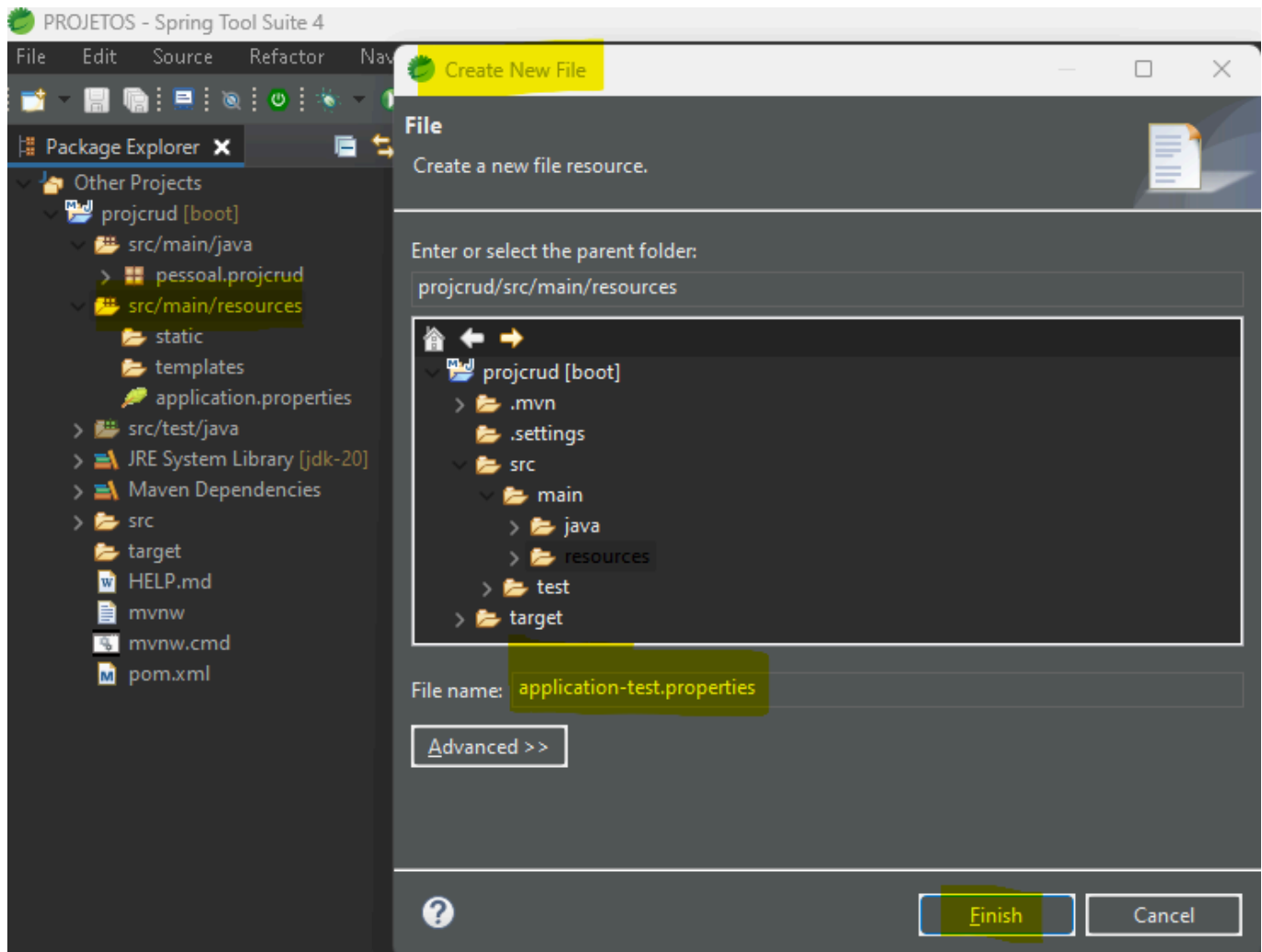
*spring.profiles.active=test*

*spring.jpa.open-in-view=false*



**NOTA:** O “spring.jpa.open-in-view”, faz com que as transações ao banco com JPA sejam encerradas na camada de Serviço. Não passando para a camada de Controle.

## Criar o arquivo de configuração application-test.properties



## Implementar o application-test.properties

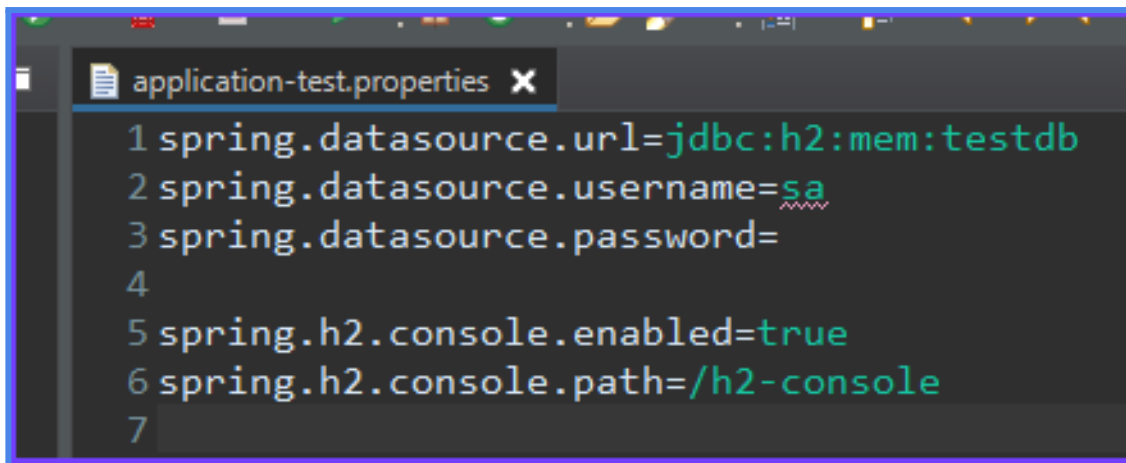
*spring.datasource.url=jdbc:h2:mem:testdb*

*spring.datasource.username=sa*

*spring.datasource.password=*

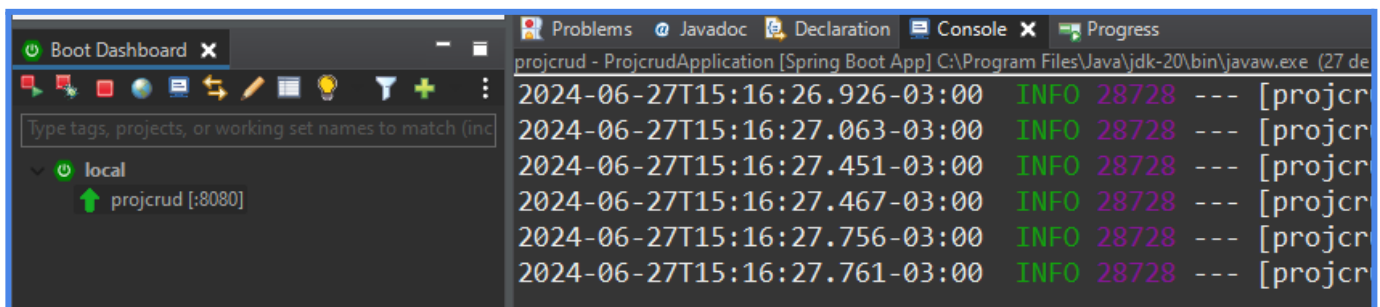
*spring.h2.console.enabled=true*

*spring.h2.console.path=/h2-console*



```
application-test.properties
1 spring.datasource.url=jdbc:h2:mem:testdb
2 spring.datasource.username=sa
3 spring.datasource.password=
4
5 spring.h2.console.enabled=true
6 spring.h2.console.path=/h2-console
7
```

## Rodar o projeto

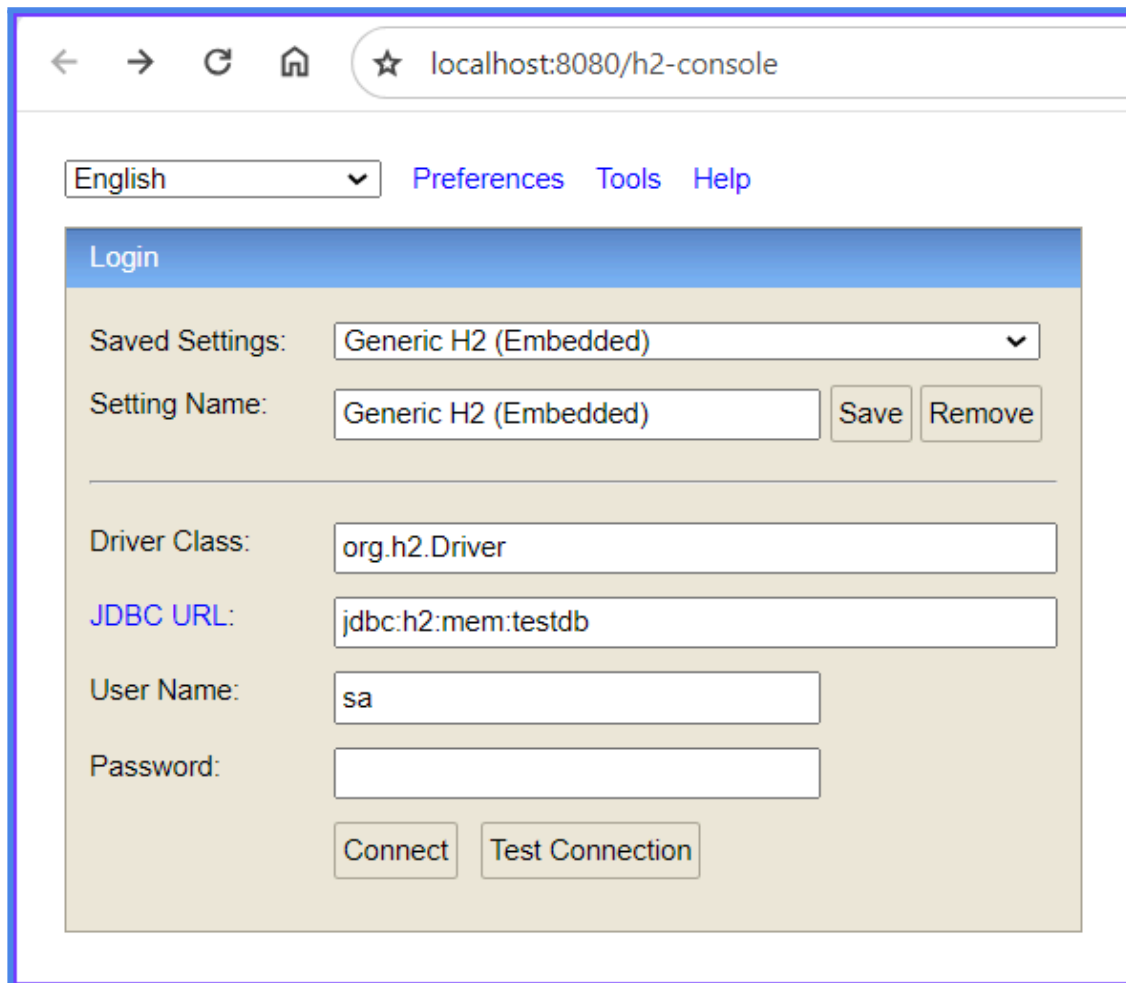


```
Boot Dashboard
Type tags, projects, or working set names to match (inc
local
  ↑ projcrud [:8080]

Problems Javadoc Declaration Console X Progress
projcrud - ProjcrudApplication [Spring Boot App] C:\Program Files\Java\jdk-20\bin\javaw.exe (27 de
2024-06-27T15:16:26.926-03:00 INFO 28728 --- [projcr
2024-06-27T15:16:27.063-03:00 INFO 28728 --- [projcr
2024-06-27T15:16:27.451-03:00 INFO 28728 --- [projcr
2024-06-27T15:16:27.467-03:00 INFO 28728 --- [projcr
2024-06-27T15:16:27.756-03:00 INFO 28728 --- [projcr
2024-06-27T15:16:27.761-03:00 INFO 28728 --- [projcr
```

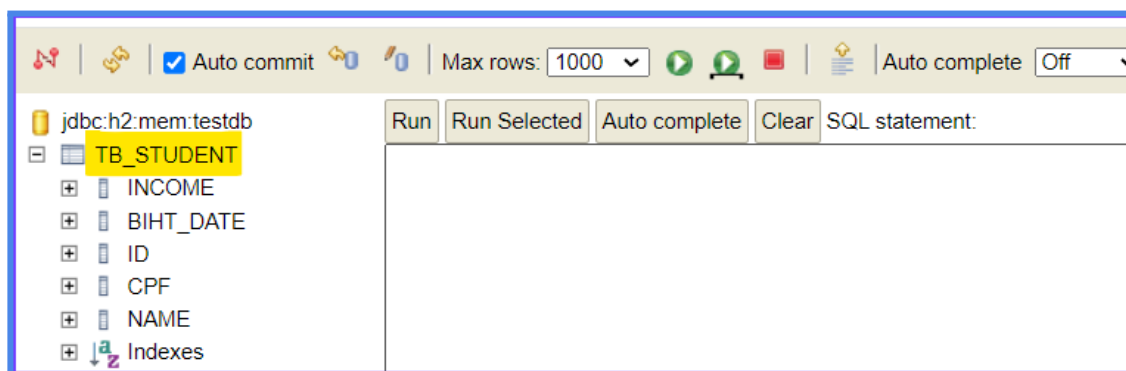
## Acessar o banco H2, via web

- <http://localhost:8080/h2-console>



The screenshot shows the H2 console login page in a web browser. The address bar displays 'localhost:8080/h2-console'. The page has a navigation bar with 'English' (a dropdown menu), 'Preferences', 'Tools', and 'Help'. The main content area is titled 'Login' and contains the following fields and buttons:

- Saved Settings:** A dropdown menu showing 'Generic H2 (Embedded)'.
- Setting Name:** A text input field containing 'Generic H2 (Embedded)', with 'Save' and 'Remove' buttons to its right.
- Driver Class:** A text input field containing 'org.h2.Driver'.
- JDBC URL:** A text input field containing 'jdbc:h2:mem:testdb'.
- User Name:** A text input field containing 'sa'.
- Password:** An empty text input field.
- Buttons:** 'Connect' and 'Test Connection' buttons at the bottom.



The screenshot shows the main interface of the H2 console. The top toolbar includes icons for connection, settings, and execution, along with 'Auto commit' (checked), 'Max rows: 1000', and 'Auto complete' (set to 'Off'). The left sidebar shows a tree view of the database structure:

- jdbc:h2:mem:testdb
  - TB\_STUDENT** (highlighted in yellow)
  - INCOME
  - BIHT\_DATE
  - ID
  - CPF
  - NAME
  - Indexes

The right side of the interface contains buttons for 'Run', 'Run Selected', 'Auto complete', and 'Clear', followed by a text area labeled 'SQL statement:'.

## Teste de inserção

Run
Run Selected
Auto complete
Clear
SQL statement:

INSERT INTO TB\_STUDENT (NAME) VALUES ('Auricelio')

INSERT INTO TB\_STUDENT (NAME) VALUES ('Auricelio');  
Update count: 1  
(3 ms)

Run
Run Selected
Auto complete
Clear
SQL statement:

SELECT \* FROM TB\_STUDENT

SELECT \* FROM TB\_STUDENT;  

INCOME	BIRTH_DATE	ID	CPF	NAME
null	null	1	null	Auricelio

(1 row, 3 ms)

## Testar no postman

GET
localhost:8080/students
Send

Params
Auth
Headers (7)
Body
Pre-req.
Tests
Settings
Cookies

Body
Cookies
Headers (5)
Test Results
200 OK
51 ms
234 B
Save as example

Pretty
Raw
Preview
Visualize
JSON

```

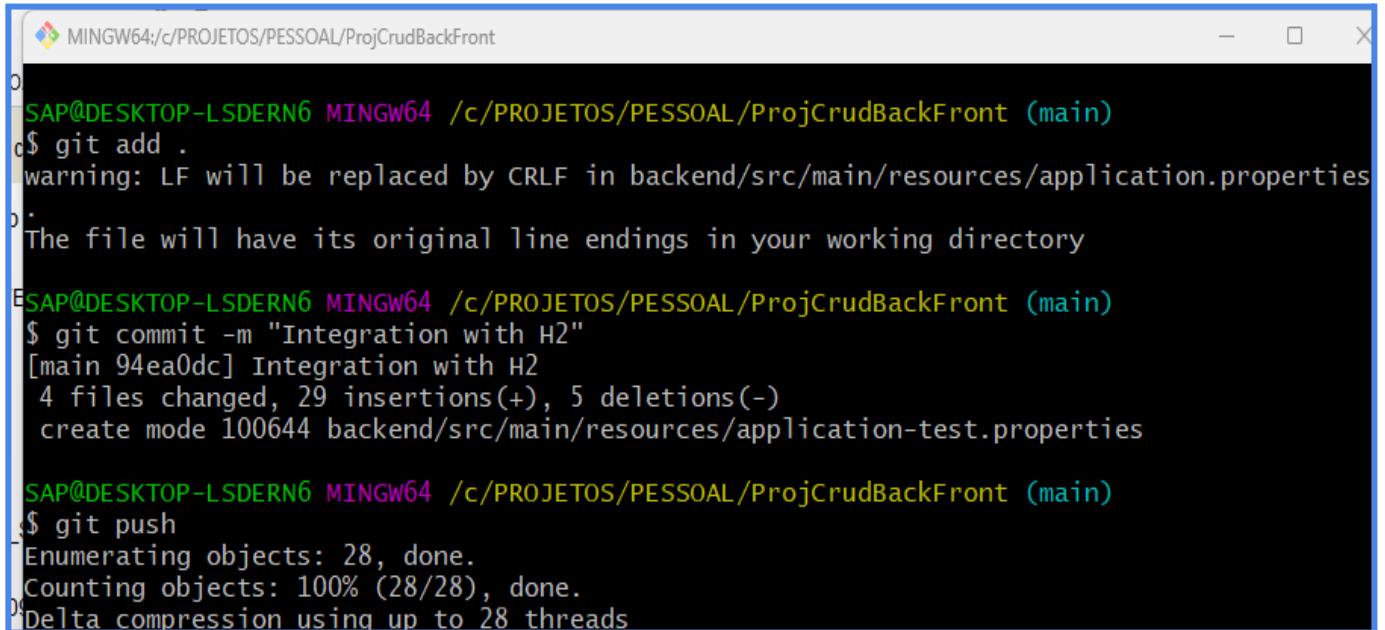
1 [
2   {
3     "id": 1,
4     "name": "Auricelio",
5     "cpf": null,
6     "birthDate": null,
7     "income": null
8   }
9 ]

```



### Github-6

- “Git bash here” no diretório do projeto
  - git add .
  - git commit -m “Integration with H2”
  - git push



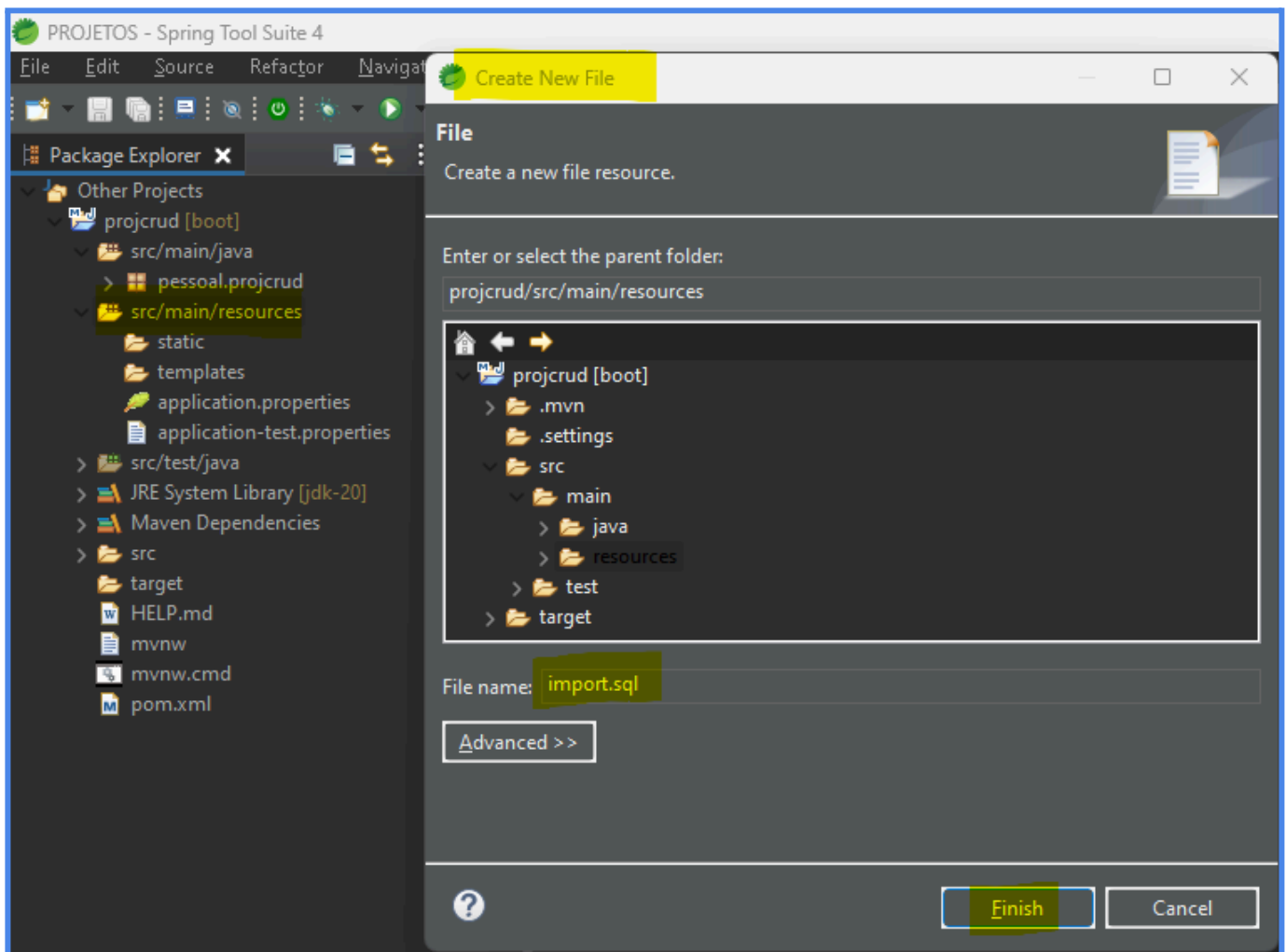
```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git add .
warning: LF will be replaced by CRLF in backend/src/main/resources/application.properties
The file will have its original line endings in your working directory
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git commit -m "Integration with H2"
[main 94ea0dc] Integration with H2
4 files changed, 29 insertions(+), 5 deletions(-)
create mode 100644 backend/src/main/resources/application-test.properties
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git push
Enumerating objects: 28, done.
Counting objects: 100% (28/28), done.
Delta compression using up to 28 threads
```

## SEEDING DA BASE DE DADOS

- É o termo utilizado para dar uma carga inicial de dados ao banco, a fim de conseguirmos realizar testes com dados. Toda vez que o sistema reiniciar a carga inicial é gerada novamente.

### IMPLEMENTAR A CARGA PARA O BANCO

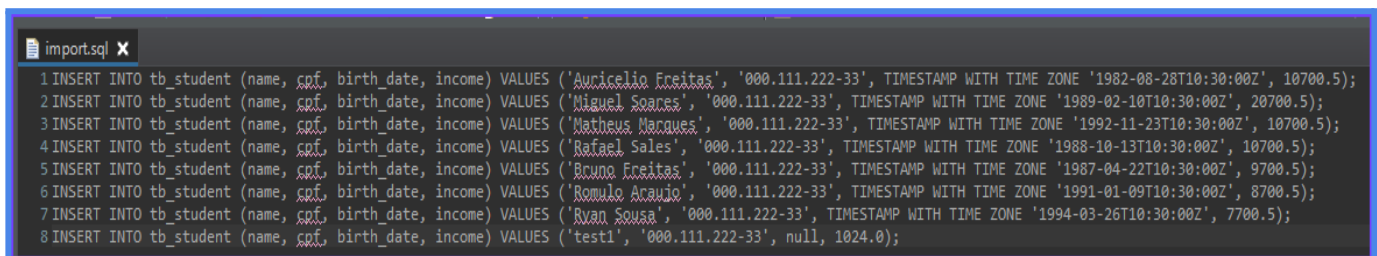
**Criar o import.sql, no src/main/resources**



**NOTA:** Nas versões antigas do STS, usava-se o termo *data.sql*. A partir da versão STS 3, passou-se a utilizar a nomenclatura *import.sql*.

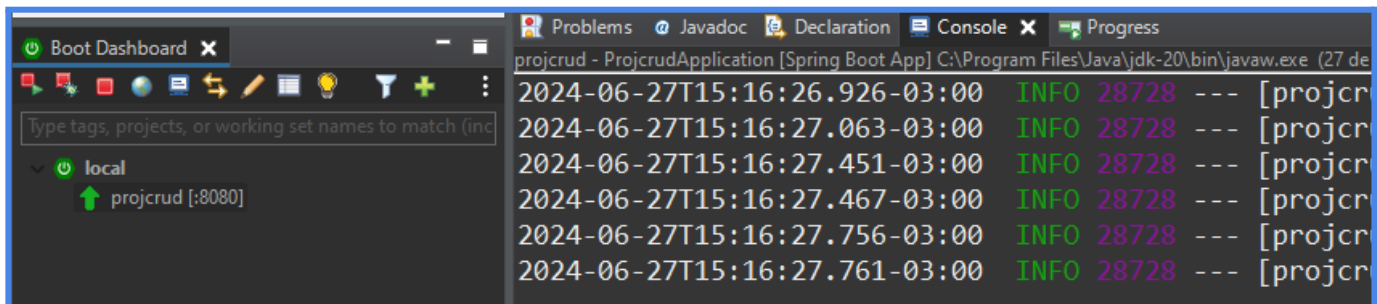
## Implementar os inserts para a carga inicial

```
INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Auricelio Freitas', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1982-08-28T10:30:00Z', 10700.5);
INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Miguel Soares', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1989-02-10T10:30:00Z', 20700.5);
INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Matheus Marques', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1992-11-23T10:30:00Z', 10700.5);
INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Rafael Sales', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1988-10-13T10:30:00Z', 10700.5);
INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Bruno Freitas', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1987-04-22T10:30:00Z', 9700.5);
INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Romulo Araujo', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1991-01-09T10:30:00Z', 8700.5);
INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Ryan Sousa', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1994-03-26T10:30:00Z', 7700.5);
INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('test1', '000.111.222-33', null, 1024.0);
```



```
import.sql x
1 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Auricelio Freitas', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1982-08-28T10:30:00Z', 10700.5);
2 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Miguel Soares', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1989-02-10T10:30:00Z', 20700.5);
3 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Matheus Marques', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1992-11-23T10:30:00Z', 10700.5);
4 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Rafael Sales', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1988-10-13T10:30:00Z', 10700.5);
5 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Bruno Freitas', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1987-04-22T10:30:00Z', 9700.5);
6 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Romulo Araujo', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1991-01-09T10:30:00Z', 8700.5);
7 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Ryan Sousa', '000.111.222-33', TIMESTAMP WITH TIME ZONE '1994-03-26T10:30:00Z', 7700.5);
8 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('test1', '000.111.222-33', null, 1024.0);
```

## Rodar o projeto

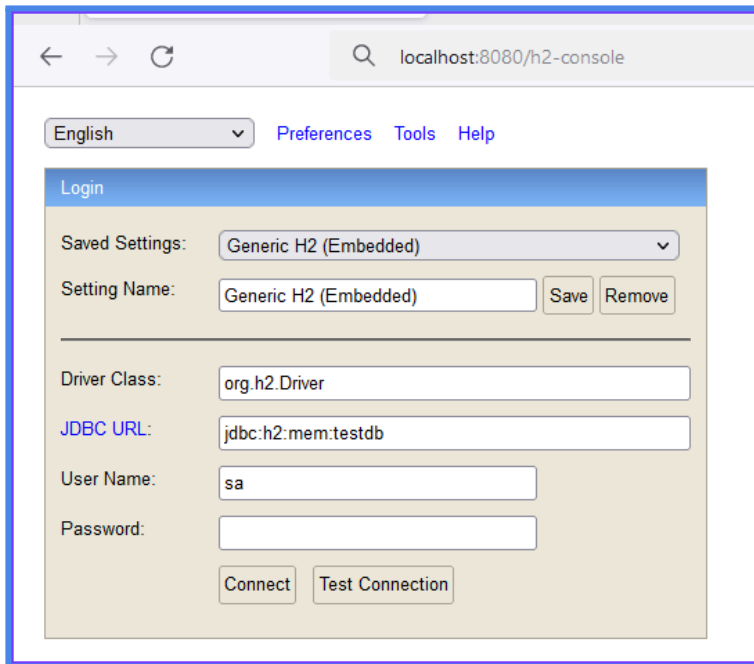


```
Boot Dashboard x
Type tags, projects, or working set names to match (inc
local
projcrud [:8080]

Problems Javadoc Declaration Console x Progress
projcrud - ProjcrudApplication [Spring Boot App] C:\Program Files\Java\jdk-20\bin\javaw.exe (27 de
2024-06-27T15:16:26.926-03:00 INFO 28728 --- [projcr
2024-06-27T15:16:27.063-03:00 INFO 28728 --- [projcr
2024-06-27T15:16:27.451-03:00 INFO 28728 --- [projcr
2024-06-27T15:16:27.467-03:00 INFO 28728 --- [projcr
2024-06-27T15:16:27.756-03:00 INFO 28728 --- [projcr
2024-06-27T15:16:27.761-03:00 INFO 28728 --- [projcr
```

## TESTAR A CARGA

### Testar no banco H2



English Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded) Save Remove

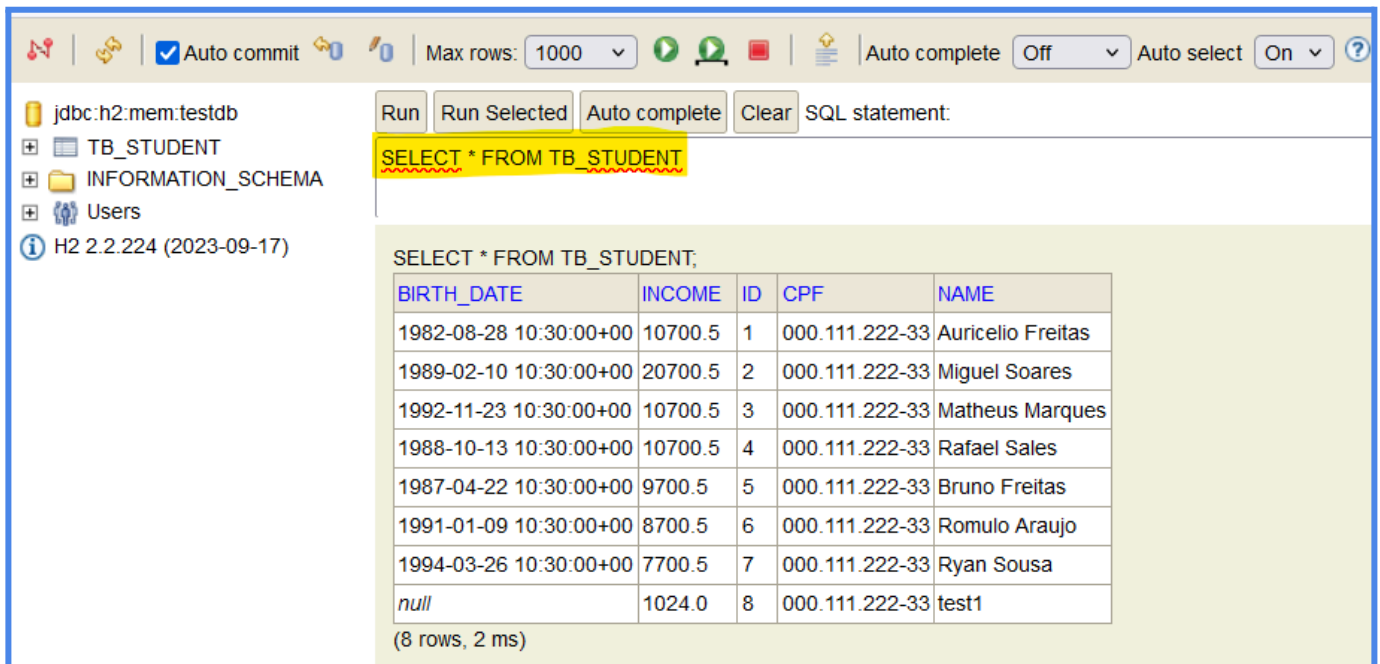
Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:testdb

User Name: sa

Password:

Connect Test Connection



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

jdbc:h2:mem:testdb

TB\_STUDENT

INFORMATION\_SCHEMA

Users

H2 2.2.224 (2023-09-17)

Run Run Selected Auto complete Clear SQL statement:

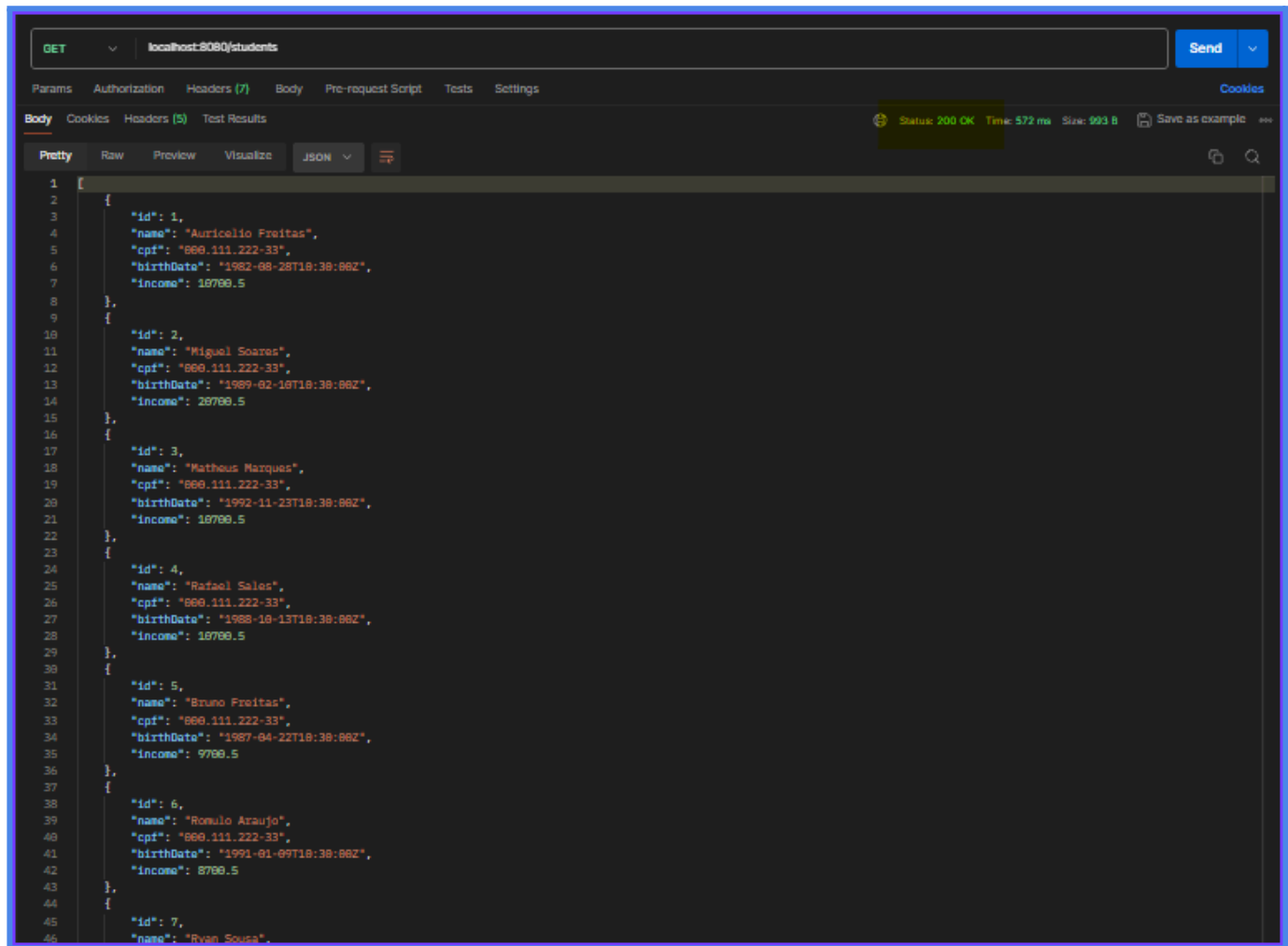
SELECT \* FROM TB\_STUDENT

SELECT \* FROM TB\_STUDENT;

BIRTH_DATE	INCOME	ID	CPF	NAME
1982-08-28 10:30:00+00	10700.5	1	000.111.222-33	Auricelio Freitas
1989-02-10 10:30:00+00	20700.5	2	000.111.222-33	Miguel Soares
1992-11-23 10:30:00+00	10700.5	3	000.111.222-33	Matheus Marques
1988-10-13 10:30:00+00	10700.5	4	000.111.222-33	Rafael Sales
1987-04-22 10:30:00+00	9700.5	5	000.111.222-33	Bruno Freitas
1991-01-09 10:30:00+00	8700.5	6	000.111.222-33	Romulo Araujo
1994-03-26 10:30:00+00	7700.5	7	000.111.222-33	Ryan Sousa
null	1024.0	8	000.111.222-33	test1

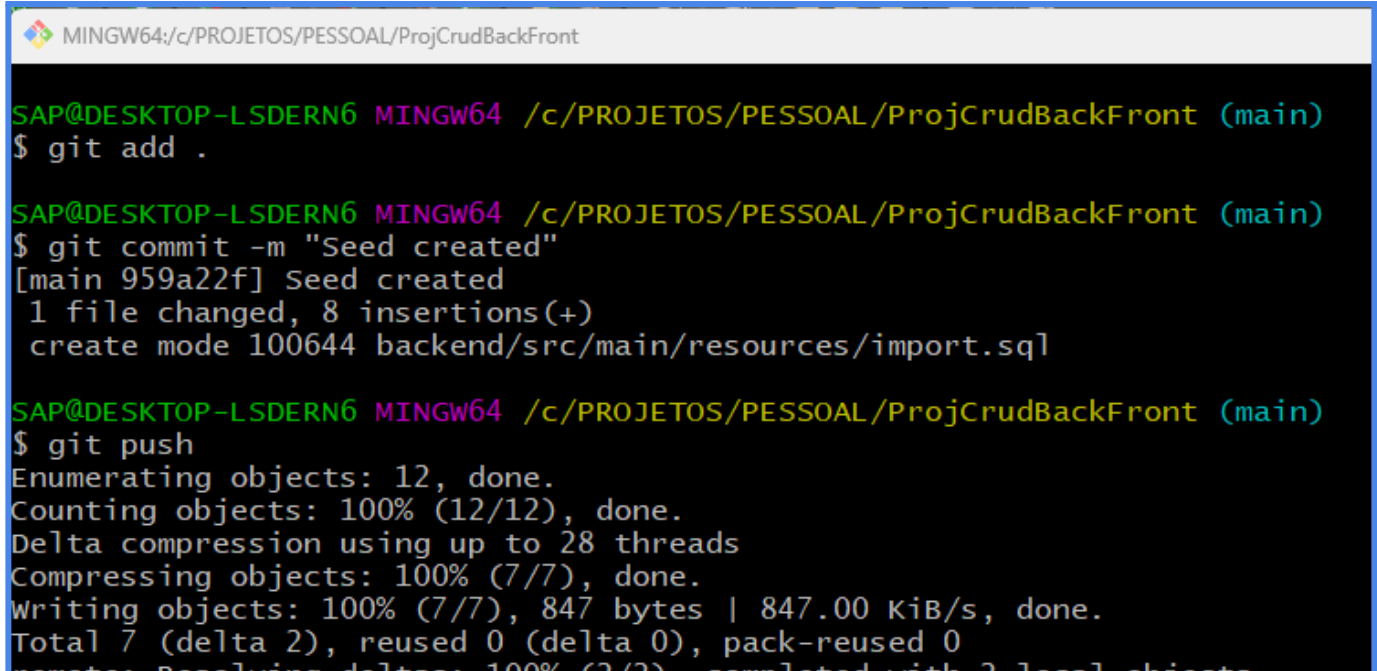
(8 rows, 2 ms)

## Testar no Postman



### Github-7

- “Git bash here” no diretório do projeto
  - git add backend
  - git commit -m “Seed created”
  - git push



```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git add .

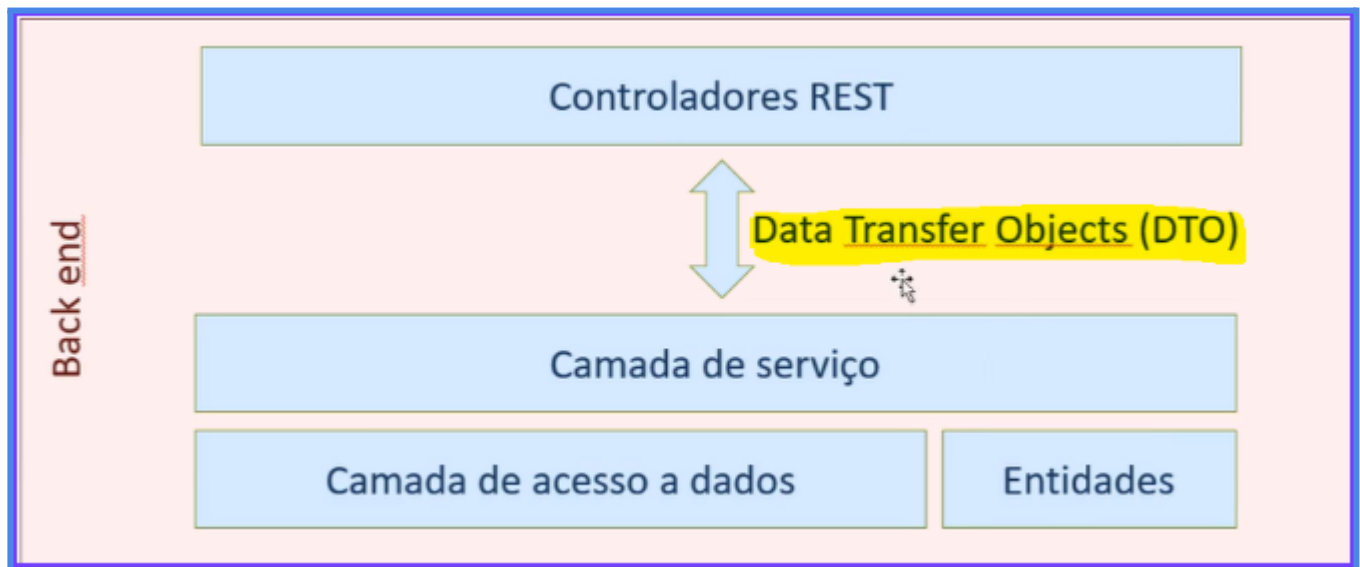
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git commit -m "Seed created"
[main 959a22f] Seed created
1 file changed, 8 insertions(+)
create mode 100644 backend/src/main/resources/import.sql

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 28 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 847 bytes | 847.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0
```

## DTO

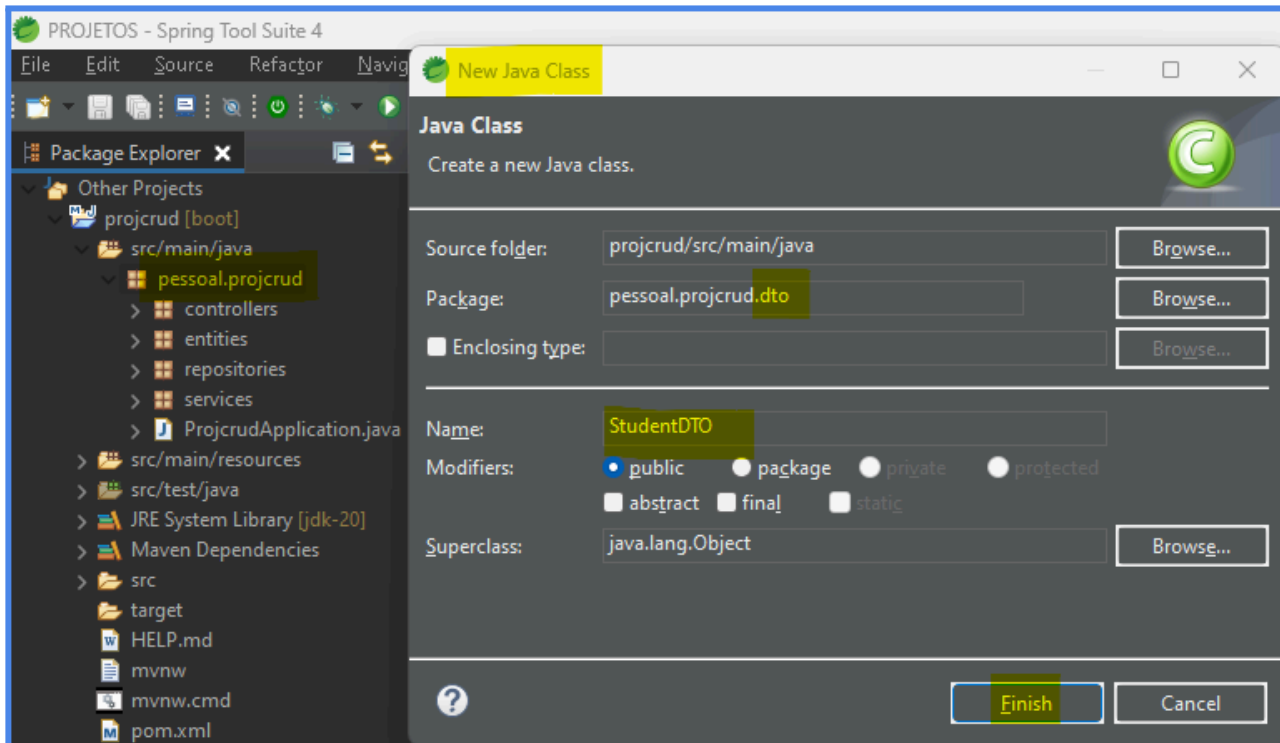
- DTO é um objeto que serve, apenas, para transferência de dados.
- Não tem relação com a JPA.
- O Controlador Rest não tem integração direta com a Entidade
- A comunicação entre o Controlador e o Serviço é feito por meio do DTO
- Podemos controlar quais dados serão entregues para a API

## CONCEITUAL



## IMPLEMENTAR A ESTRUTURA E O STUDENT\_DTO

### Criar a estrutura e a classe DTO



### Implementar o Serializable e os mesmos atributos da Classe Student

```
*StudentDTO.java x
6 public class StudentDTO implements Serializable {
7     private static final long serialVersionUID = 1L;
8
9     private Long id;
10    private String name;
11    private String cpf;
12    private Instant birthDate;
13    private Double income;
```

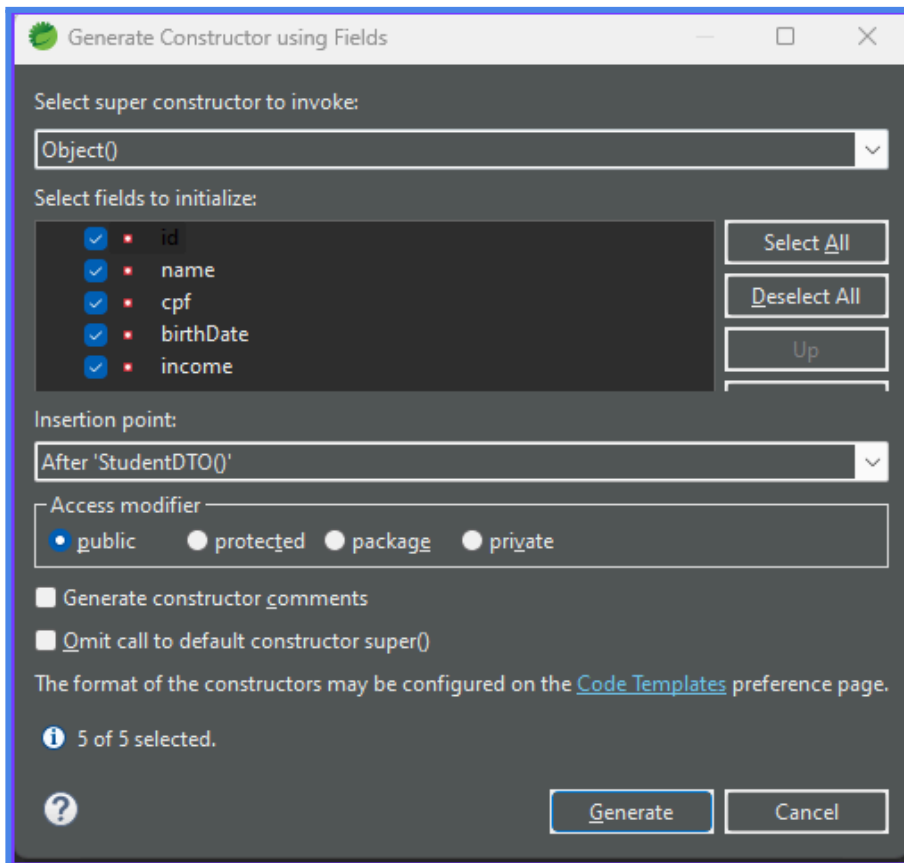


## Implementar os construtores

Vazio

```
*StudentDTO.java x
15 public StudentDTO() {
16
17 }
```

de Classe

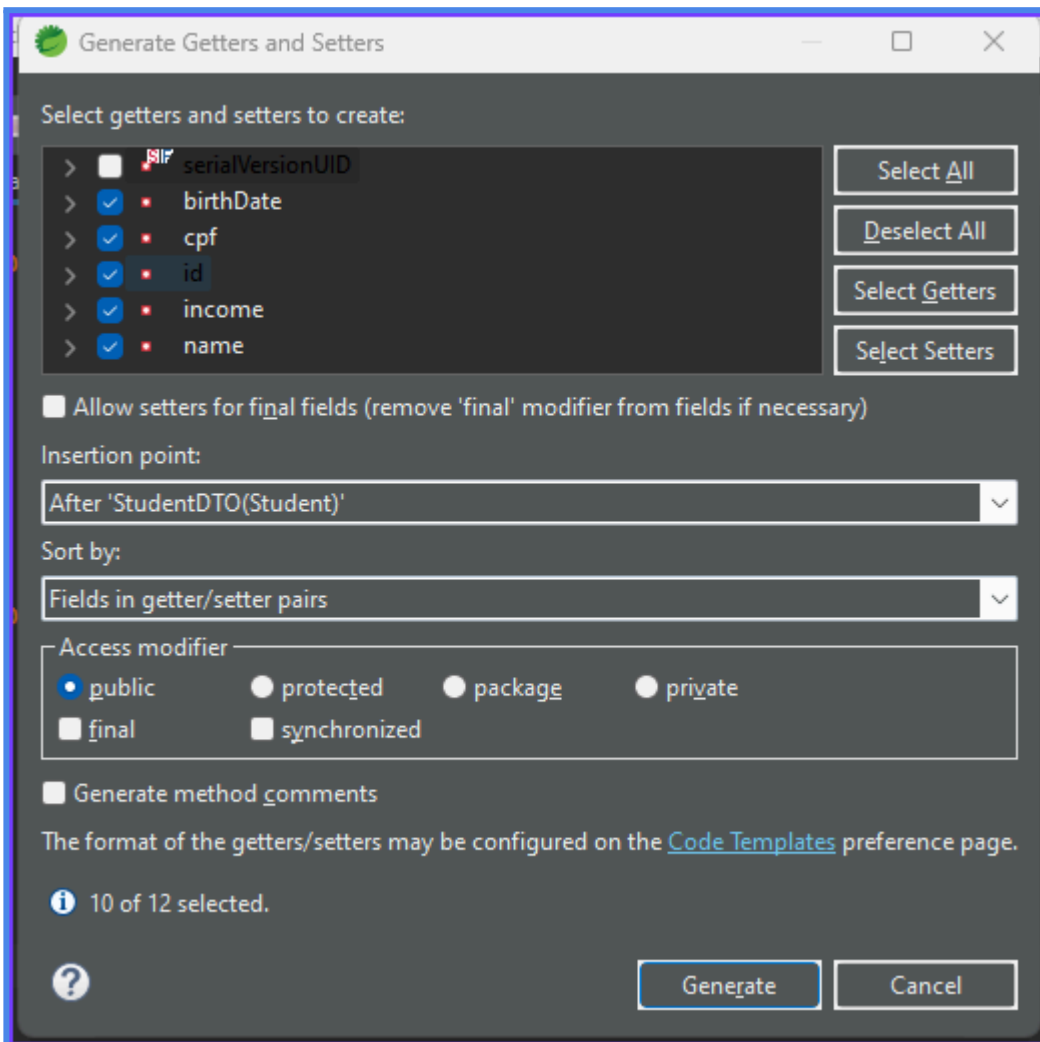


```
*StudentDTO.java x
20
21 public StudentDTO(Long id, String name, String cpf, Instant birthDate, Double income) {
22     this.id = id;
23     this.name = name;
24     this.cpf = cpf;
25     this.birthDate = birthDate;
26     this.income = income;
27 }
```

## De Entidade

```
StudentDTO.java x
29
30 public StudentDTO(Student entity) {
31     id = entity.getId();
32     name = entity.getName();
33     cpf = entity.getCpf();
34     birthDate = entity.getBirthDate();
35     income = entity.getIncome();
36 }
37
```

### Implementar os Getters and Setters

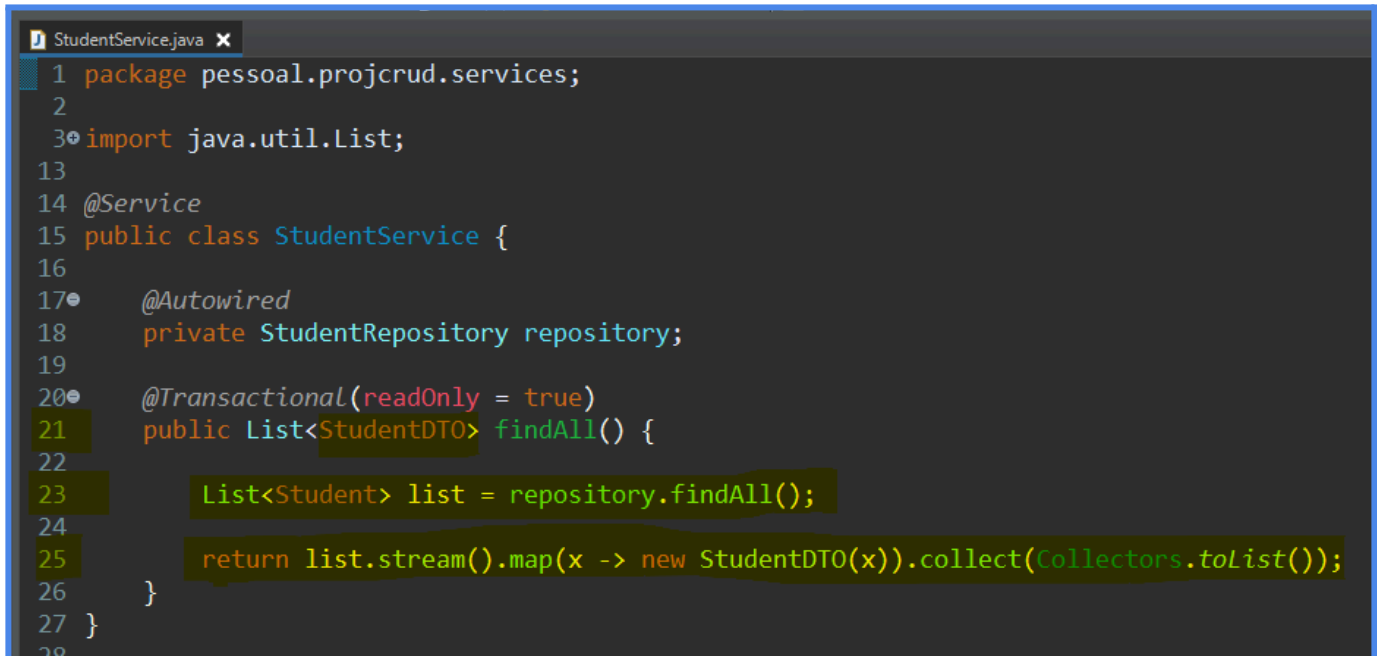


```
*StudentDTO.java x
37 public Long getId() {
38     return id;
39 }
40
41 public void setId(Long id) {
42     this.id = id;
43 }
44
45 public String getName() {
46     return name;
47 }
48
49 public void setName(String name) {
50     this.name = name;
51 }
52
53 public String getCpf() {
54     return cpf;
55 }
56
57 public void setCpf(String cpf) {
58     this.cpf = cpf;
59 }
```

```
60
61 public LocalDate getBirthDate() {
62     return birthDate;
63 }
64
65 public void setBirthDate(LocalDate birthDate) {
66     this.birthDate = birthDate;
67 }
68
69 public Double getIncome() {
70     return income;
71 }
72
73 public void setIncome(Double income) {
74     this.income = income;
75 }
76 }
77
```

## REALIZAR OS AJUSTES PARA O DTO

### Implementar o DTO na classe StudentService

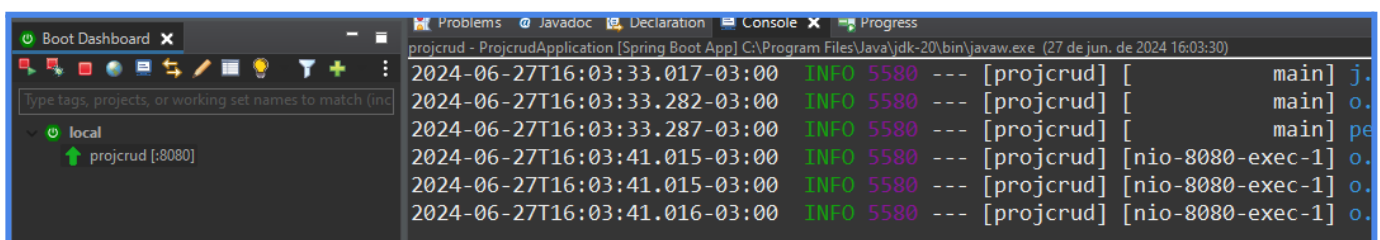


```
StudentService.java x
1 package pessoal.projcrud.services;
2
3 import java.util.List;
13
14 @Service
15 public class StudentService {
16
17     @Autowired
18     private StudentRepository repository;
19
20     @Transactional(readOnly = true)
21     public List<StudentDTO> findAll() {
22
23         List<Student> list = repository.findAll();
24
25         return list.stream().map(x -> new StudentDTO(x)).collect(Collectors.toList());
26     }
27 }
28
```

## Implementar o DTO na classe StudentController

```
StudentController.java X
1 package pessoal.projcrud.controllers;
2
3 import java.util.List;
13
14 @RestController
15 @RequestMapping(value = "/students")
16 public class StudentController {
17
18     @Autowired
19     private StudentService service;
20
21     @GetMapping
22     public ResponseEntity<List<StudentDTO>> findAll() {
23
24         List<StudentDTO> list = service.findAll();
25
26         return ResponseEntity.ok().body(list);
27     }
28 }
```

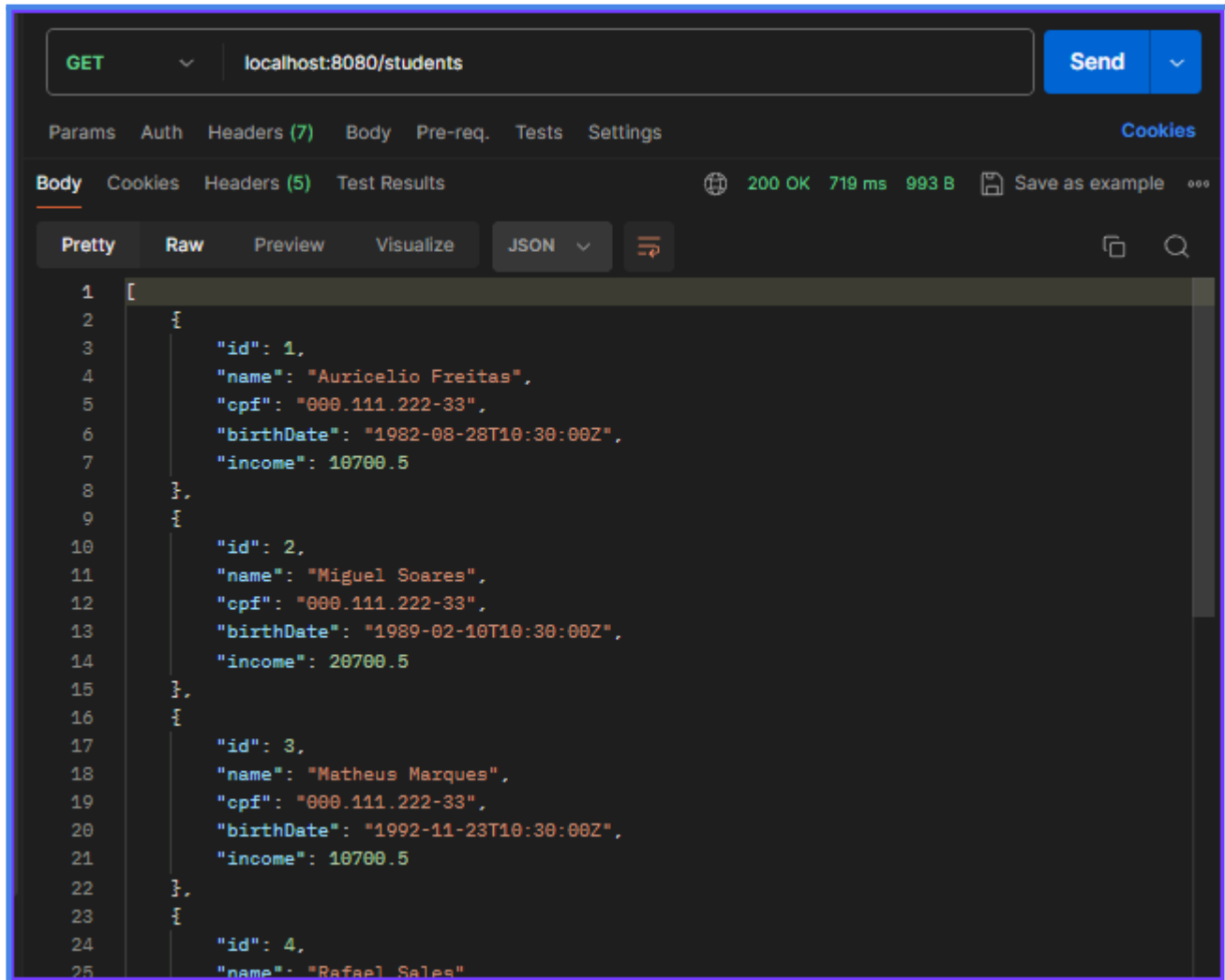
## Rodar o projeto



The screenshot shows the Spring Boot IDE interface. On the left, the 'Boot Dashboard' shows the project 'projcrud' running on port 8080. The main window displays the 'Console' tab with the following output:

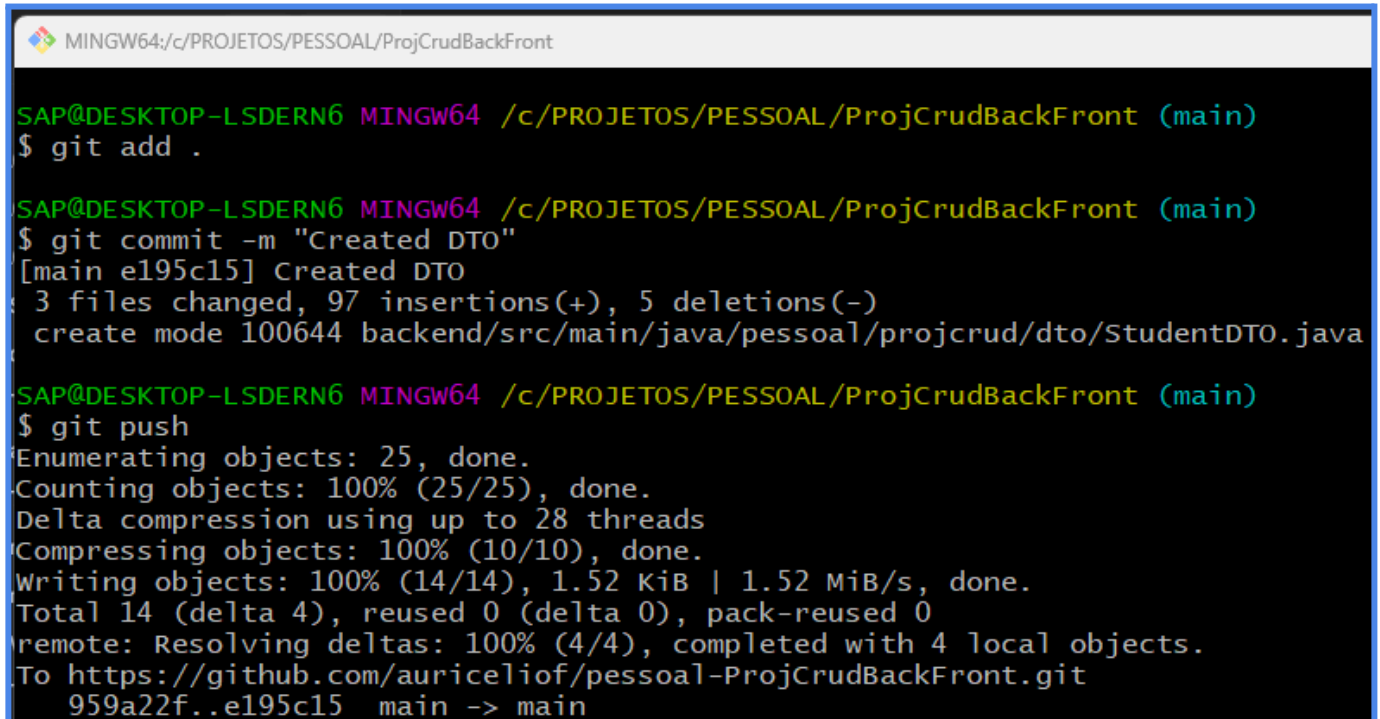
```
projcrud - ProjcrudApplication [Spring Boot App] C:\Program Files\Java\jdk-20\bin\javaw.exe (27 de jun. de 2024 16:03:30)
2024-06-27T16:03:33.017-03:00 INFO 5580 --- [projcrud] [main] j.
2024-06-27T16:03:33.282-03:00 INFO 5580 --- [projcrud] [main] o.
2024-06-27T16:03:33.287-03:00 INFO 5580 --- [projcrud] [main] pe
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.016-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
```

## TESTAR COM O POSTMAN



## Github-8

- “Git bash here” no diretório do projeto
  - git add backend
  - git commit -m “Created DTO”
  - git push



```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git add .

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git commit -m "Created DTO"
[main e195c15] Created DTO
3 files changed, 97 insertions(+), 5 deletions(-)
create mode 100644 backend/src/main/java/pessoal/projcrud/dto/StudentDTO.java

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git push
Enumerating objects: 25, done.
Counting objects: 100% (25/25), done.
Delta compression using up to 28 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (14/14), 1.52 KiB | 1.52 MiB/s, done.
Total 14 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/auriceliof/pessoal-ProjCrudBackFront.git
959a22f..e195c15  main -> main
```

## ENDPOINT: FIND\_BY\_ID

### BUSCAR ALUNOS POR ID COM GET

Implementar busca por Id, no StudentController

```
*StudentController.java x
29
30 @GetMapping(value =("/{id}")
31 public ResponseEntity<StudentDTO> findById(@PathVariable Long id) {
32
33     StudentDTO dto = service.findById(id);
34
35     return ResponseEntity.ok().body(dto);
36 }
37 }
```

Implementar o método findById, no StudentService

**DICA:** No StudentController, clicando sobre o método e em sobre o Create method ..., será criado um método default no StudentService, onde podemos implementá-lo em seguida.

```
*StudentController.java x
29
30 @GetMapping(value =("/{id}")
31 public ResponseEntity<StudentDTO> findById(@PathVariable Long id) {
32
33     StudentDTO dto = service.findById(id);
34
35     return ResponseEntity.ok(
36 }
37 }
38
```

The method findById(Long) is undefined for the type StudentService  
2 quick fixes available:  
• Create method 'findById(Long)' in type 'StudentService'  
• Add cast to 'service'



```
*StudentService.java X
27
28 public StudentDTO findById(Long id) {
29     // TODO Auto-generated method stub
30     return null;
31 }
```

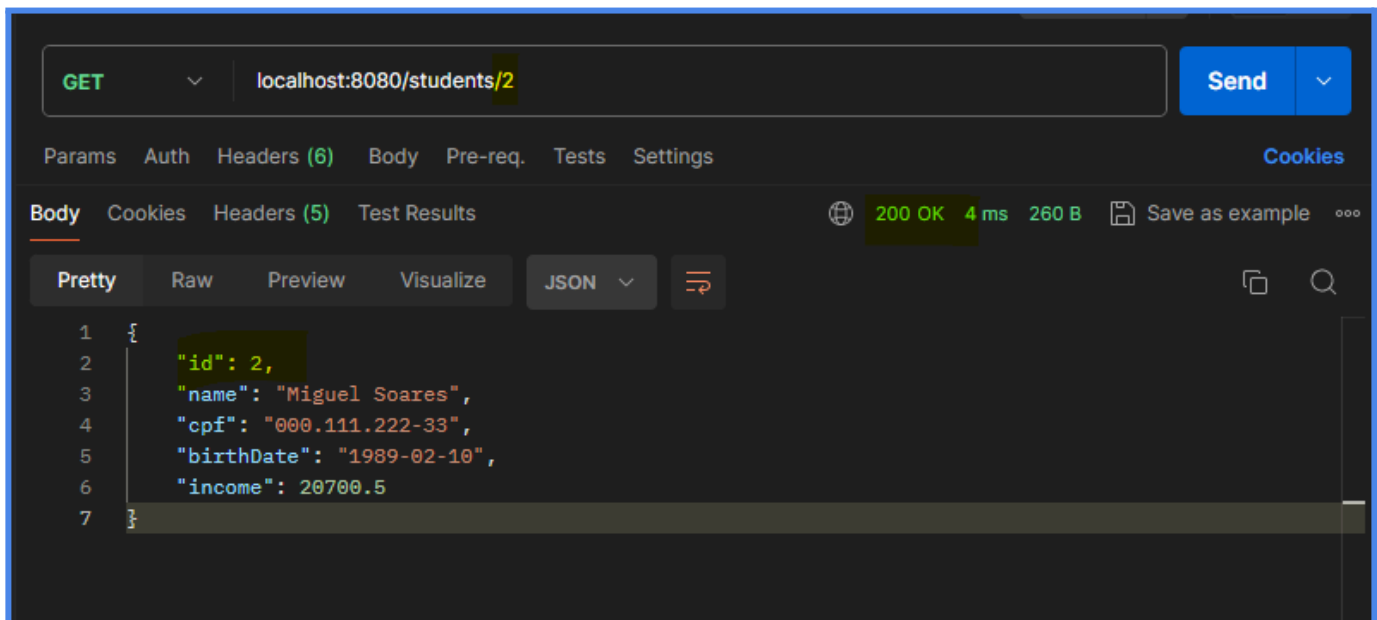
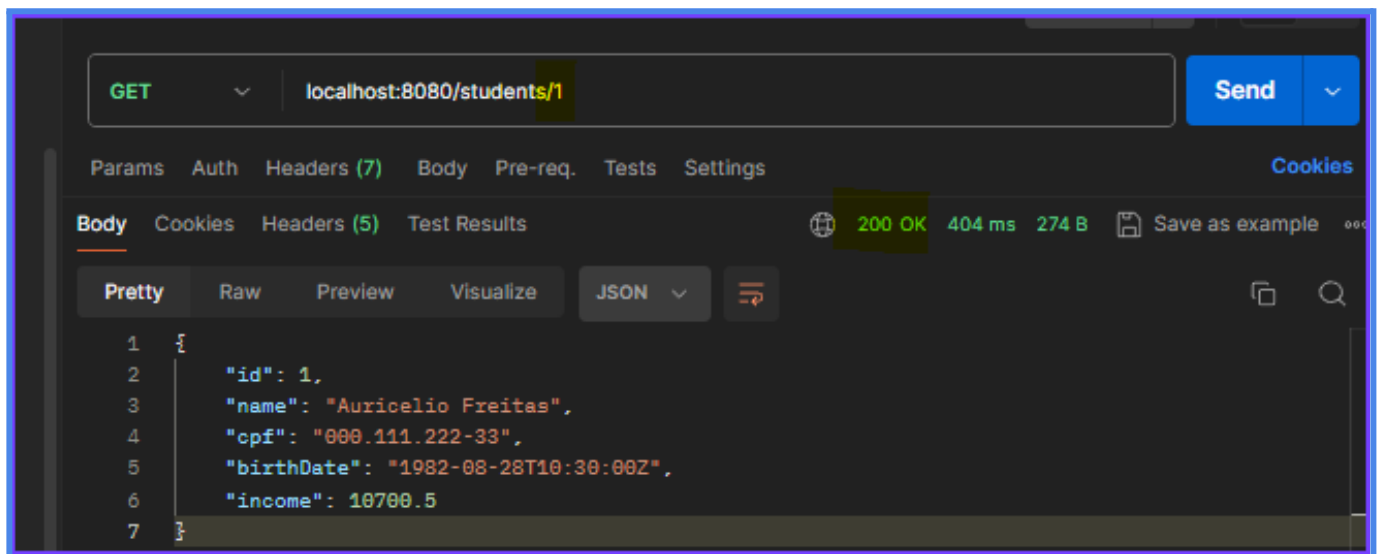
```
StudentService.java X
28
29 @Transactional
30 public StudentDTO findById(Long id) {
31
32     Optional<Student> obj = repository.findById(id);
33     Student entity = obj.get();
34
35     return new StudentDTO(entity);
36 }
37 }
```

## Rodar o projeto

```
Boot Dashboard X
Type tags, projects, or working set names to match (inc
local
projcrud [:8080]

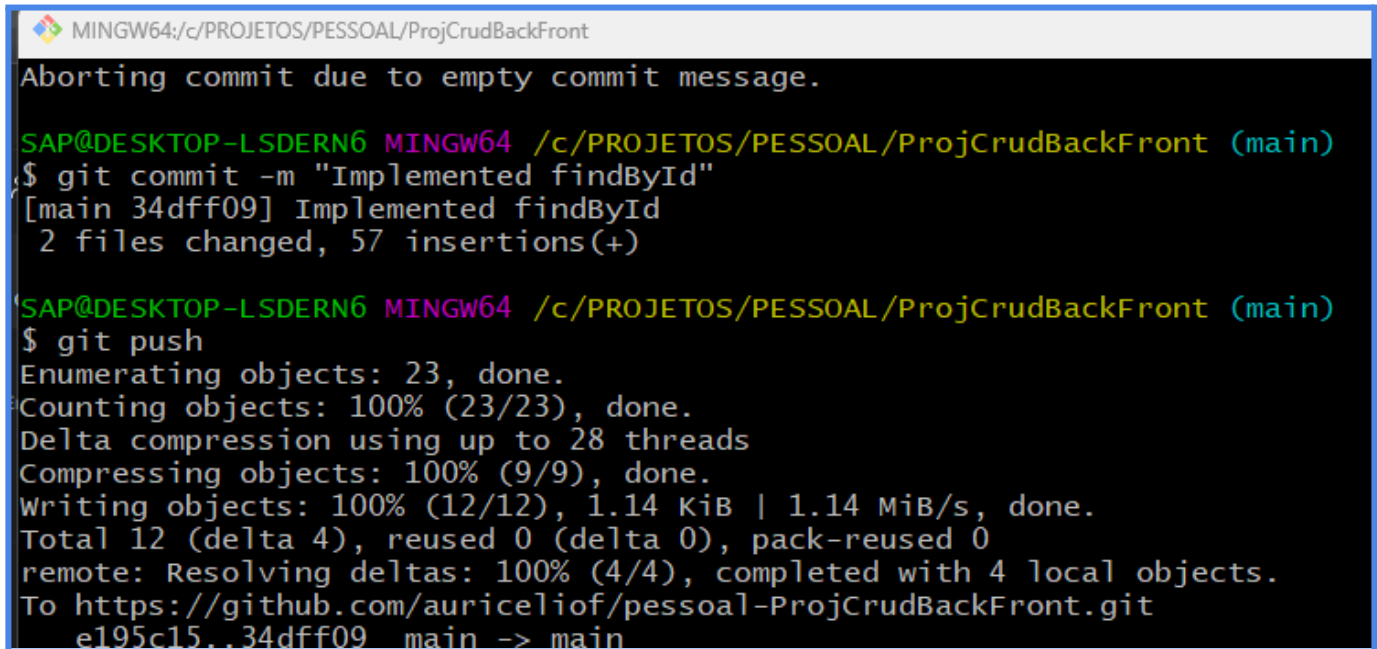
Problems Javadoc Declaration Console X Progress
projcrud - ProjcrudApplication [Spring Boot App] C:\Program Files\Java\jdk-20\bin\javaw.exe (27 de jun. de 2024 16:03:30)
2024-06-27T16:03:33.017-03:00 INFO 5580 --- [projcrud] [main] j.
2024-06-27T16:03:33.282-03:00 INFO 5580 --- [projcrud] [main] o.
2024-06-27T16:03:33.287-03:00 INFO 5580 --- [projcrud] [main] pe
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.016-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
```

## TESTAR NO POSTMAN



### Github-9

- “Git bash here” no diretório do projeto
  - git add backend
  - git commit -m “Implemented findById”
  - git push



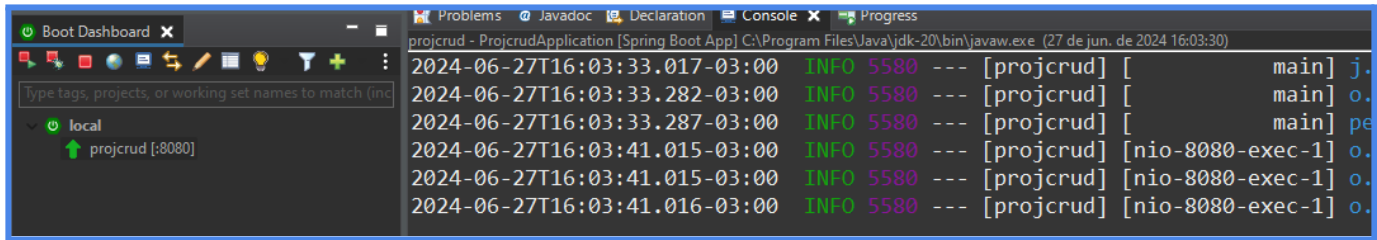
```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront
Aborting commit due to empty commit message.

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git commit -m "Implemented findById"
[main 34dff09] Implemented findById
2 files changed, 57 insertions(+)

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git push
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 28 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (12/12), 1.14 KiB | 1.14 MiB/s, done.
Total 12 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/auriceliof/pessoal-ProjCrudBackFront.git
e195c15..34dff09 main -> main
```

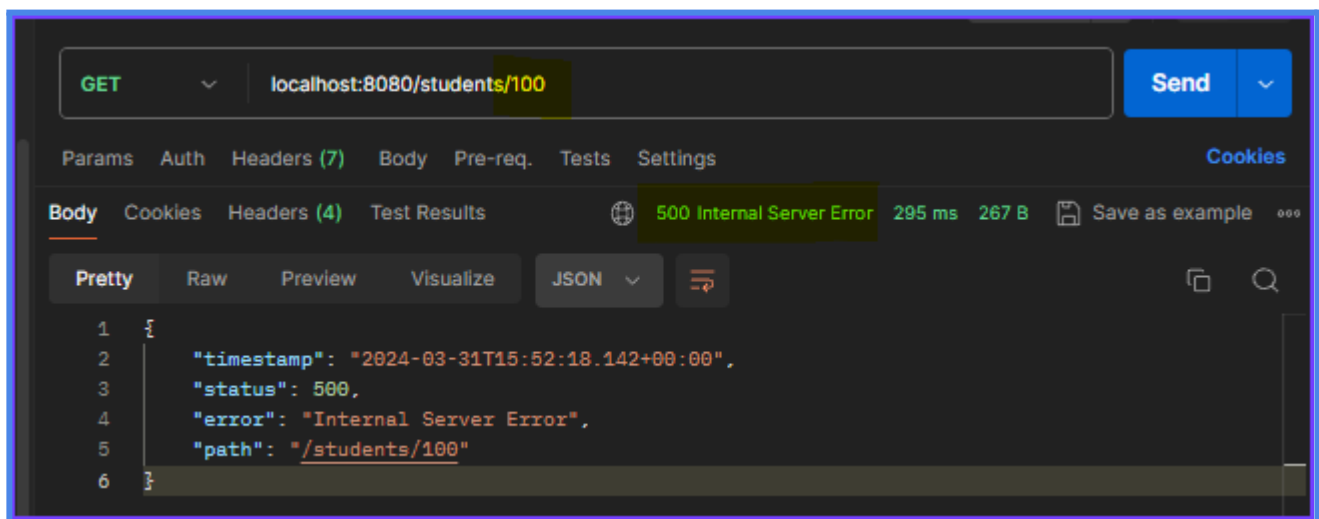
## TRATAMENTO DE EXCEÇÕES PARA O FIND BY ID

### Rodar o projeto



```
projcrud - ProjcrudApplication [Spring Boot App] C:\Program Files\Java\jdk-20\bin\javaw.exe (27 de jun. de 2024 16:03:30)
2024-06-27T16:03:33.017-03:00 INFO 5580 --- [projcrud] [main] j.
2024-06-27T16:03:33.282-03:00 INFO 5580 --- [projcrud] [main] o.
2024-06-27T16:03:33.287-03:00 INFO 5580 --- [projcrud] [main] pe
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.016-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
```

### Simular erro no Postman

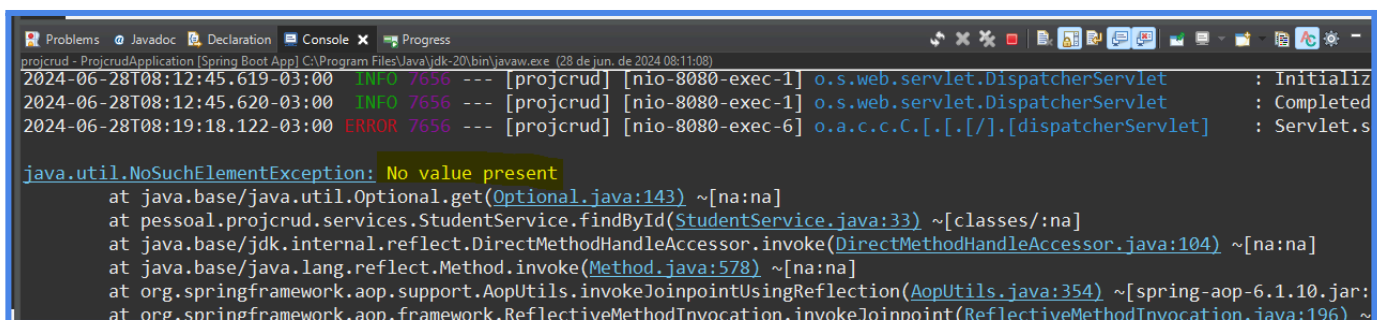


```
GET localhost:8080/students/100
Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies
Body Cookies Headers (4) Test Results 500 Internal Server Error 295 ms 267 B Save as example
Pretty Raw Preview Visualize JSON
1 {
2   "timestamp": "2024-03-31T15:52:18.142+00:00",
3   "status": 500,
4   "error": "Internal Server Error",
5   "path": "/students/100"
6 }
```

NOTA: Quando realizamos uma busca por um ID que não existe, é retornado o erro 500. Iremos tratá-lo a seguir.

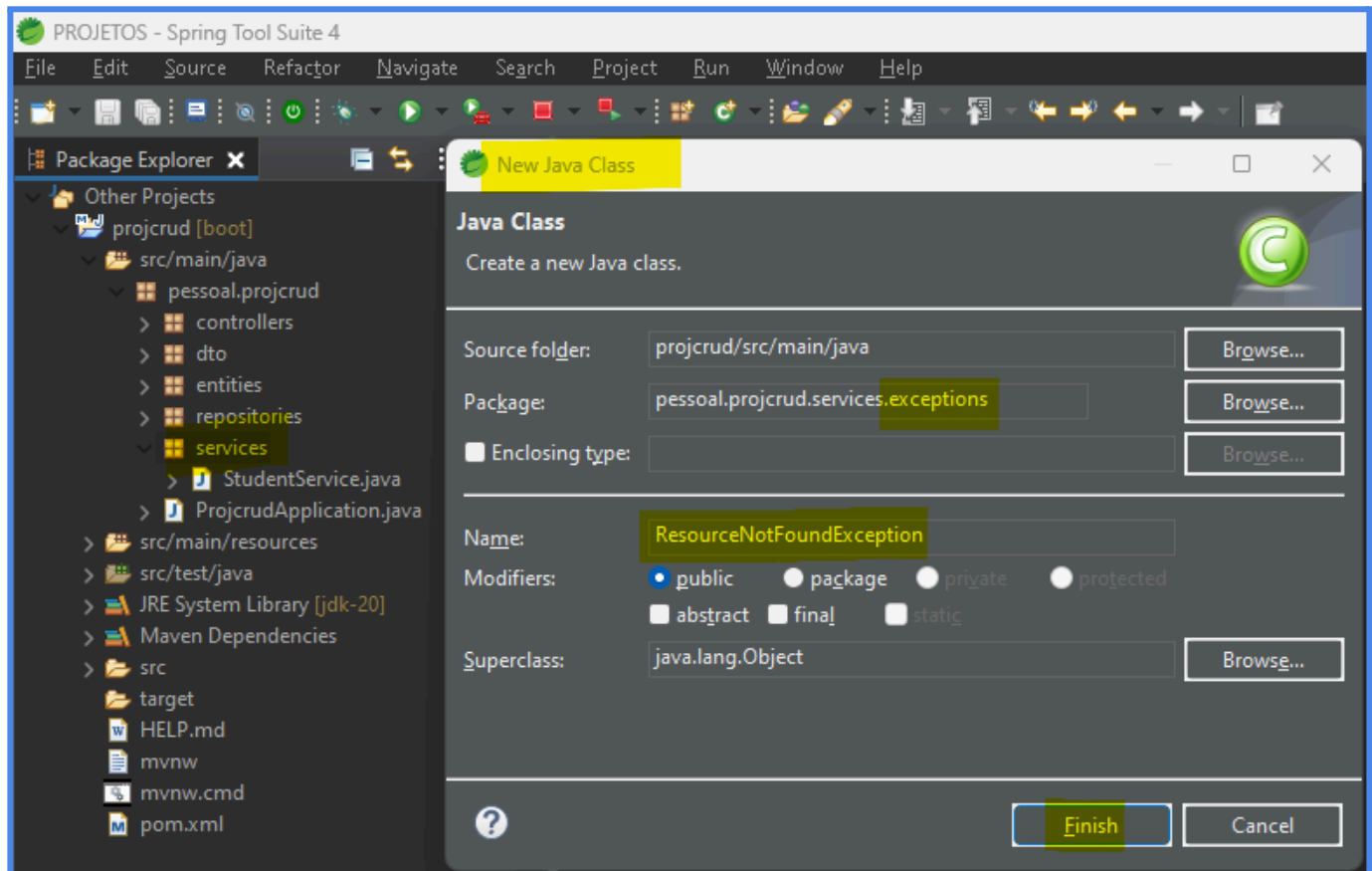
### Verificar o erro no console



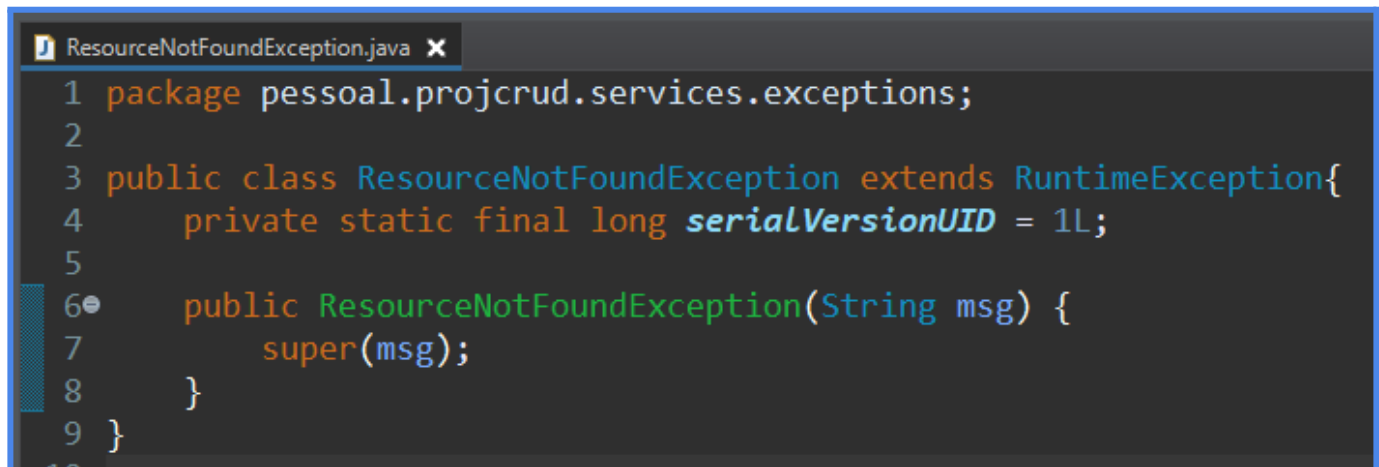
```
projcrud - ProjcrudApplication [Spring Boot App] C:\Program Files\Java\jdk-20\bin\javaw.exe (28 de jun. de 2024 08:11:08)
2024-06-28T08:12:45.619-03:00 INFO 7656 --- [projcrud] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializ
2024-06-28T08:12:45.620-03:00 INFO 7656 --- [projcrud] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed
2024-06-28T08:19:18.122-03:00 ERROR 7656 --- [projcrud] [nio-8080-exec-6] o.a.c.c.C.[.[/].[dispatcherServlet] : Servlet.s

java.util.NoSuchElementException: No value present
    at java.base/java.util.Optional.get(Optional.java:143) ~[na:na]
    at pessoal.projcrud.services.StudentService.findById(StudentService.java:33) ~[classes/:na]
    at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:104) ~[na:na]
    at java.base/java.lang.reflect.Method.invoke(Method.java:578) ~[na:na]
    at org.springframework.aop.support.AopUtils.invokeJoinpointUsingReflection(AopUtils.java:354) ~[spring-aop-6.1.10.jar:
    at org.springframework.aop.framework.ReflectiveMethodInvocation.invokeJoinpoint(ReflectiveMethodInvocation.java:196) ~
```

## Criar a estrutura de exceções no service



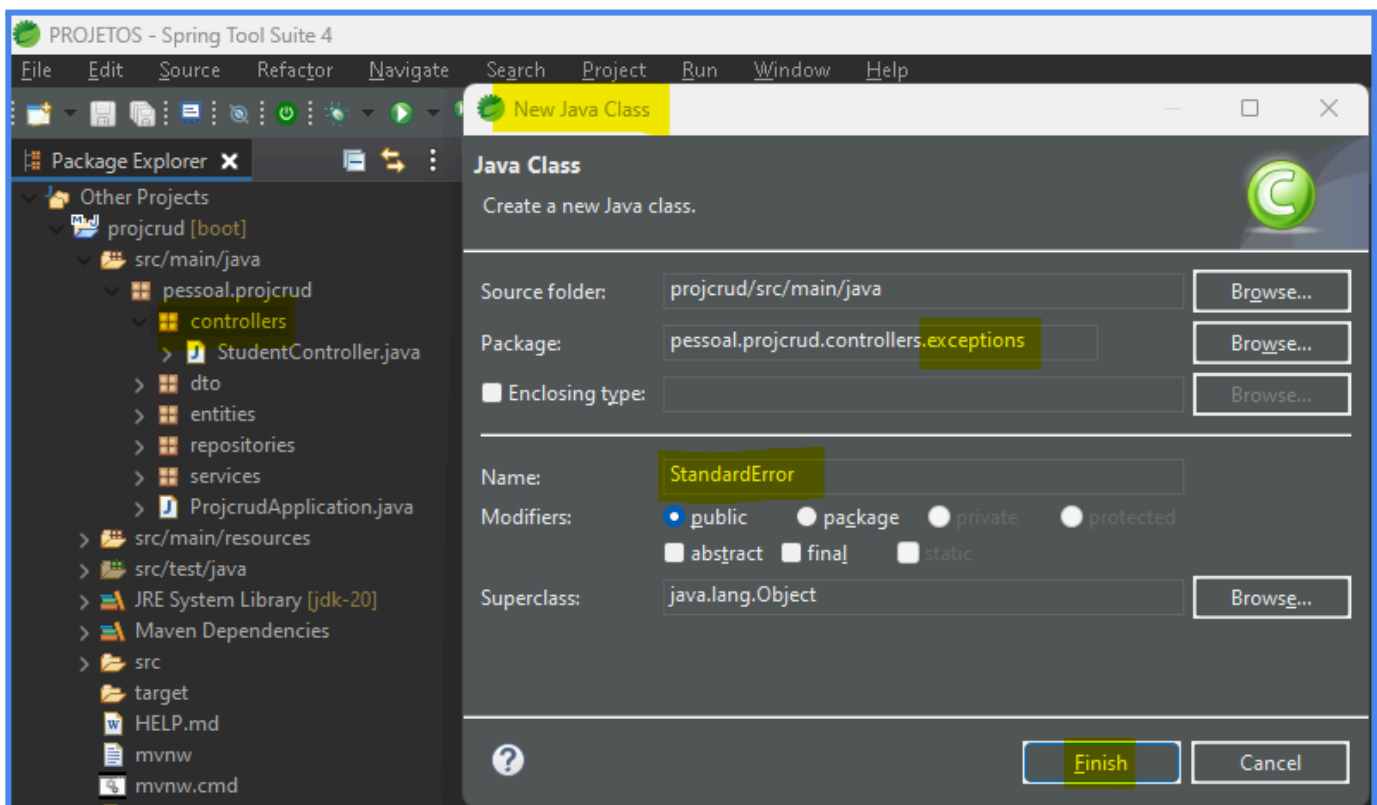
## Implementar o ResourceNotFoundException



## Implementar a exceção no findById, do StudentService

```
*StudentService.java x
29
30 @Transactional(readOnly = true)
31 public StudentDTO findById(Long id) {
32
33     Optional<Student> obj = repository.findById(id);
34     Student entity = obj.orElseThrow(() -> new ResourceNotFoundException("Entity not found"));
35
36     return new StudentDTO(entity);
37 }
```

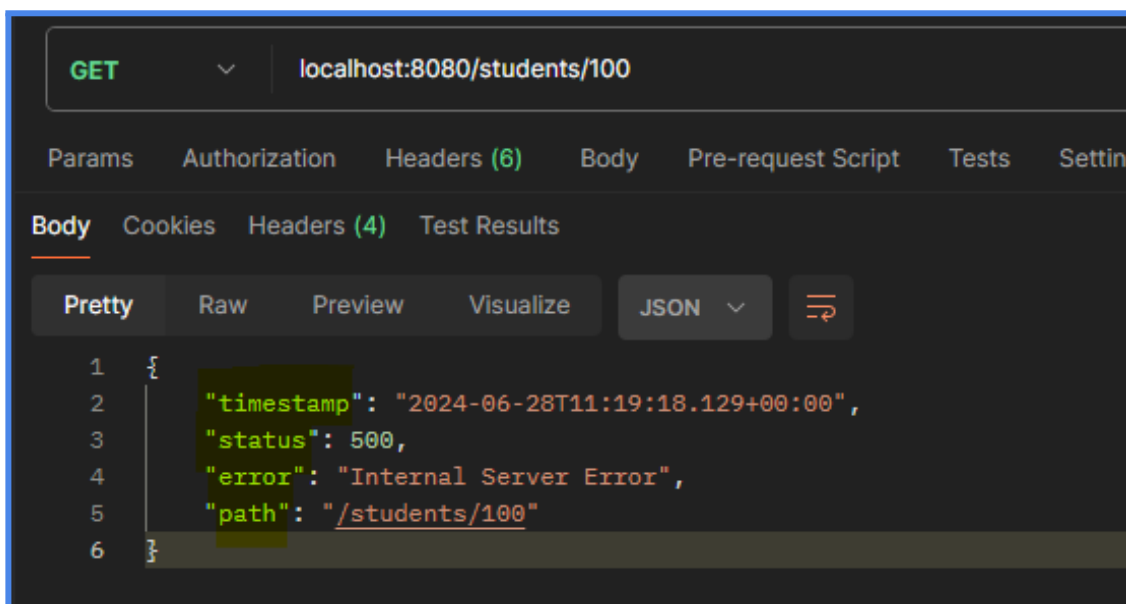
## Criar a estrutura de exceções e uma classe personalizada no controller



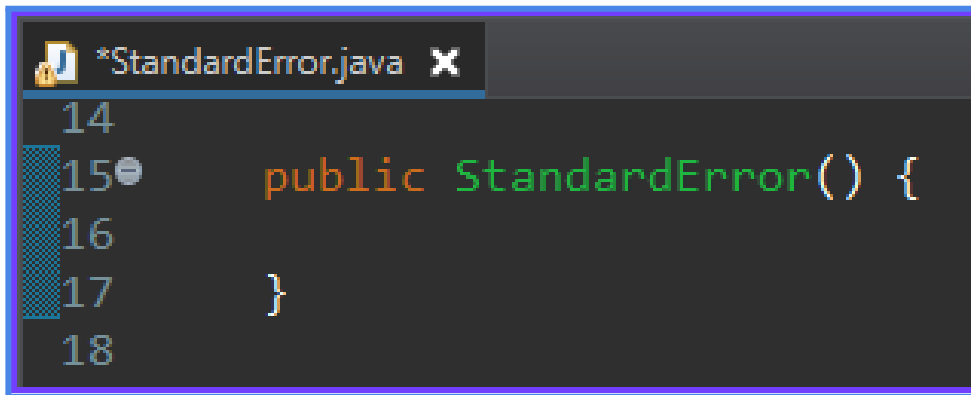
Implementar o Serializable e os atributos do StandardError conforme erro visto anteriormente

```
StandardError.java X
1 package pessoal.projcrud.controllers.exceptions;
2
3 import java.io.Serializable;
4 import java.time.Instant;
5
6 public class StandardError implements Serializable {
7     private static final long serialVersionUID = 1L;
8
9     private Instant timestamp;
10    private Integer status;
11    private String error;
12    private String message;
13    private String path;
14 }
```

**NOTA:** Devemos definir os mesmos atributos mostrados no teste de erro do postman.

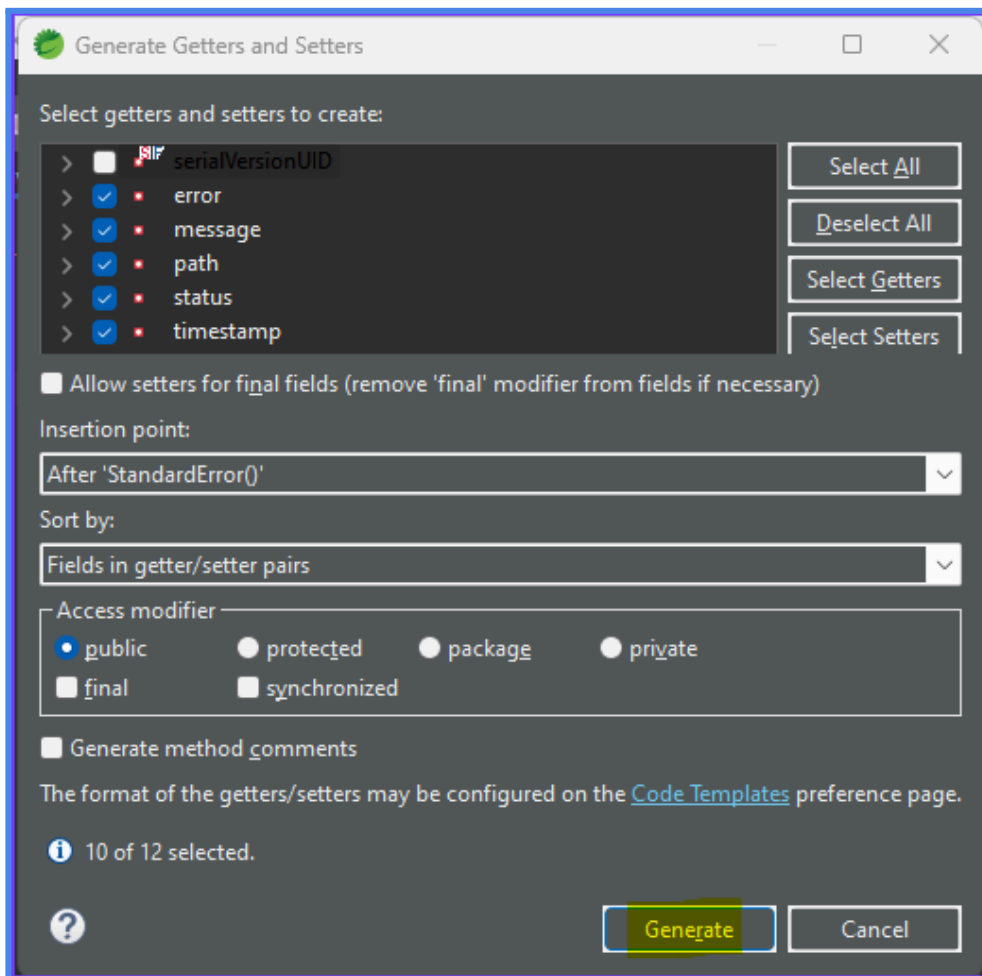


Implementar um construtor vazio



```
14
15 public StandardError() {
16
17 }
18
```

Implementar os Getters and Setters



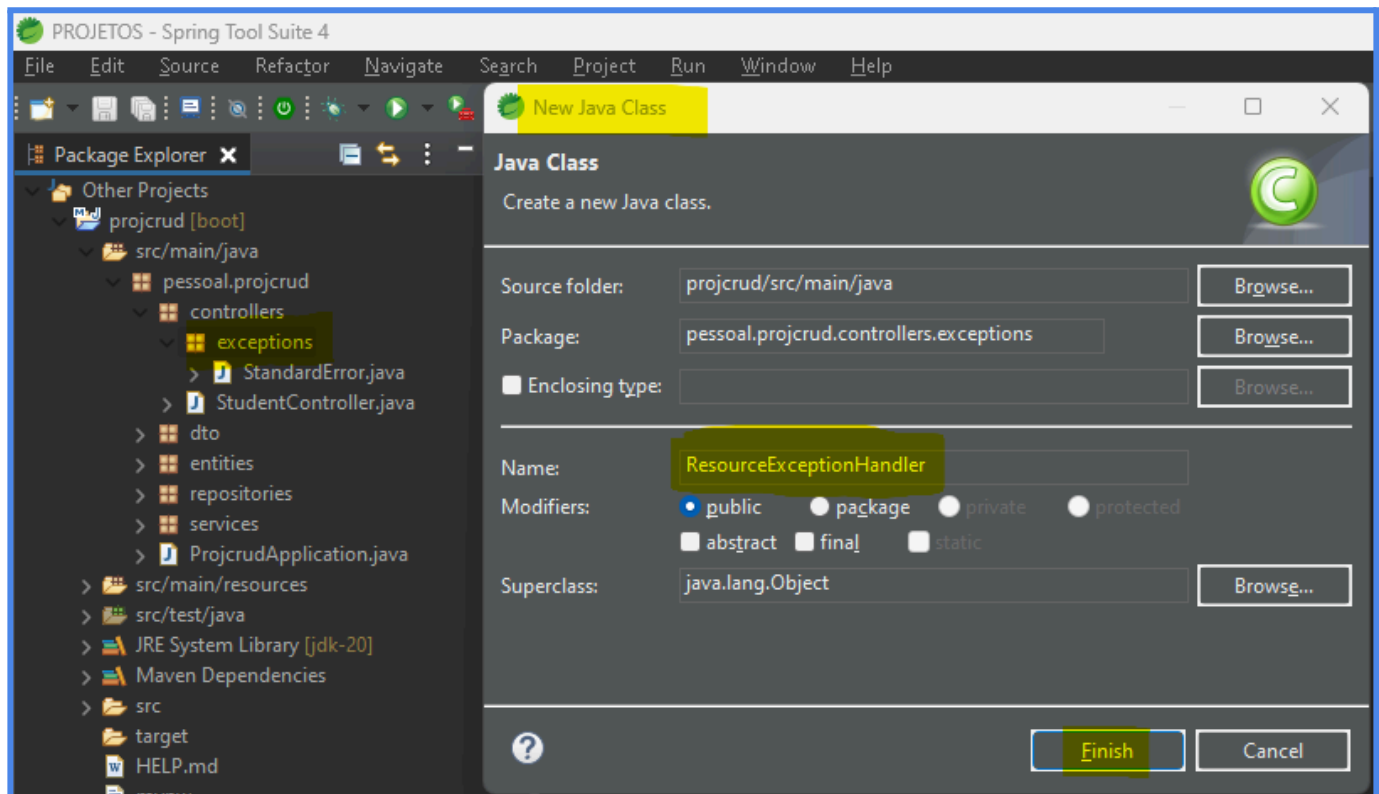


StandardError.java x

```
18
19 public Instant getTimestamp() {
20     return timestamp;
21 }
22
23 public void setTimestamp(Instant timestamp) {
24     this.timestamp = timestamp;
25 }
26
27 public Integer getStatus() {
28     return status;
29 }
30
31 public void setStatus(Integer status) {
32     this.status = status;
33 }
34
35 public String getError() {
36     return error;
37 }
38
39 public void setError(String error) {
40     this.error = error;
41 }
42
```

```
43 public String getMessage() {
44     return message;
45 }
46
47 public void setMessage(String message) {
48     this.message = message;
49 }
50
51 public String getPath() {
52     return path;
53 }
54
55 public void setPath(String path) {
56     this.path = path;
57 }
58
59 }
60
```

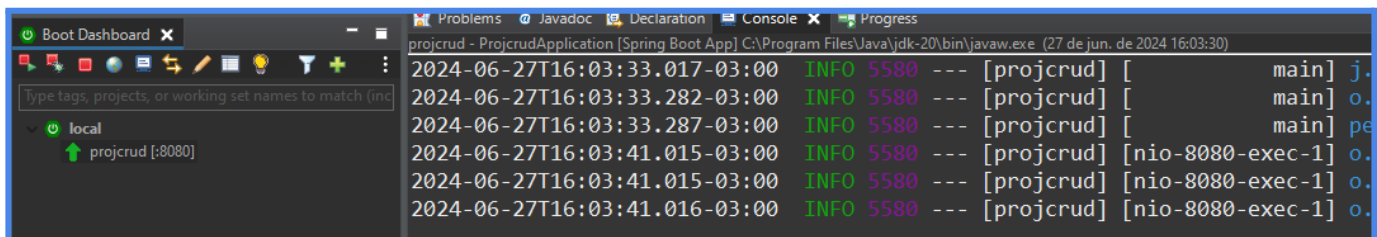
## Criar um controller advice para manipular a exceção



## Implementar o ResourceExceptionHandler

```
ResourceExceptionHandler.java x
4
5 import org.springframework.http.HttpStatus;
6 import org.springframework.http.ResponseEntity;
7 import org.springframework.web.bind.annotation.ControllerAdvice;
8 import org.springframework.web.bind.annotation.ExceptionHandler;
9
10 import jakarta.servlet.http.HttpServletRequest;
11 import pessoal.projcrud.services.exceptions.ResourceNotFoundException;
12
13 @ControllerAdvice
14 public class ResourceExceptionHandler {
15
16     @ExceptionHandler(ResourceNotFoundException.class)
17     public ResponseEntity<StandardError> resourceNotFound(ResourceNotFoundException e, HttpServletRequest request) {
18
19         HttpStatus status = HttpStatus.NOT_FOUND;
20
21         StandardError err = new StandardError();
22
23         err.setTimestamp(Instant.now());
24         err.setStatus(status.value());
25         err.setError("Resource not found");
26         err.setMessage(e.getMessage());
27         err.setPath(request.getRequestURI());
28
29         return ResponseEntity.status(status).body(err);
30     }
31 }
```

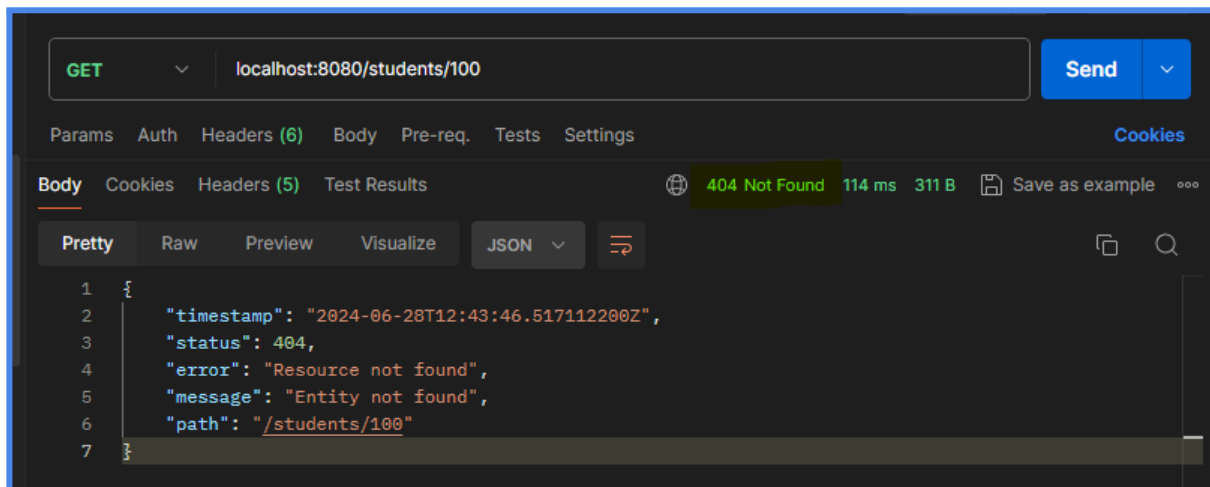
## Rodar o projeto



The screenshot shows the Spring Boot IDE interface. On the left, the 'Boot Dashboard' tab is active, displaying a list of local projects with 'projcrud [:8080]' highlighted. On the right, the 'Console' tab shows the application's startup logs. The logs indicate that the application 'projcrud - ProjcrudApplication [Spring Boot App]' is running on 'C:\Program Files\Java\jdk-20\bin\javaw.exe' at 16:03:30 on June 27, 2024. The output shows several 'INFO' messages from the 'main' thread and 'nio-8080-exec-1' threads, all indicating successful operations.

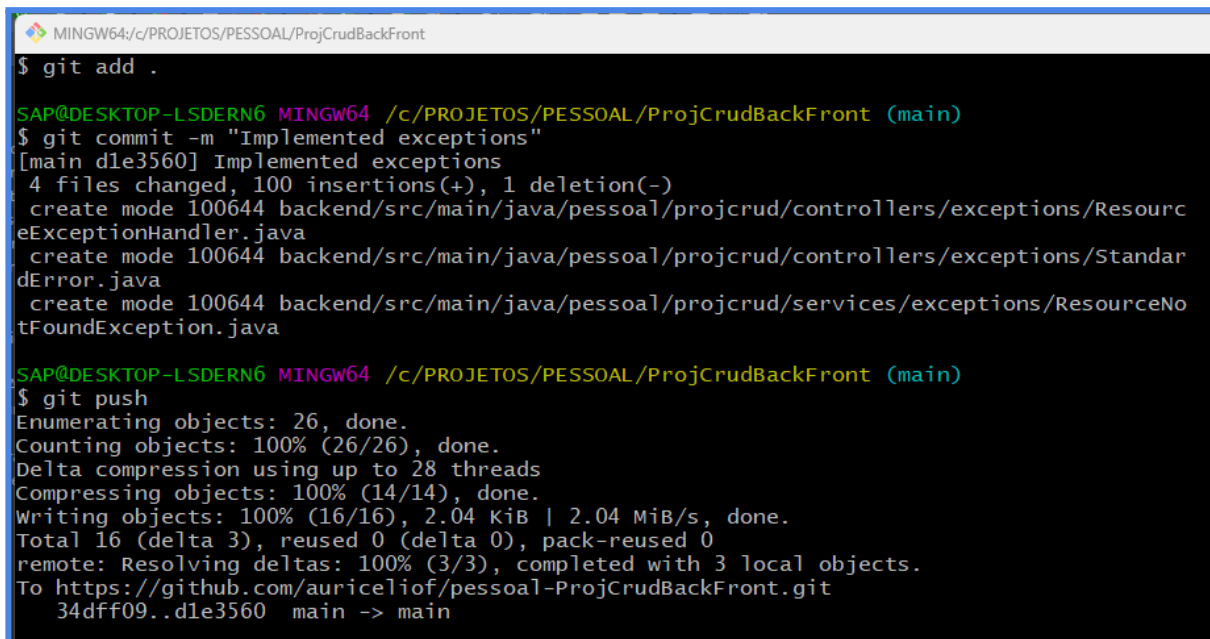
```
projcrud - ProjcrudApplication [Spring Boot App] C:\Program Files\Java\jdk-20\bin\javaw.exe (27 de jun, de 2024 16:03:30)
2024-06-27T16:03:33.017-03:00 INFO 5580 --- [projcrud] [main] j.
2024-06-27T16:03:33.282-03:00 INFO 5580 --- [projcrud] [main] o.
2024-06-27T16:03:33.287-03:00 INFO 5580 --- [projcrud] [main] pe
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.016-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
```

## TESTAR NO POSTMAN



## Github-10

- “Git bash here” no diretório do projeto
  - git add backend
  - git commit -m “Implemented exception for findById”
  - git push



## PAGINAÇÃO

### AJUSTAR O FIND ALL PARA BUSCA PAGINADA

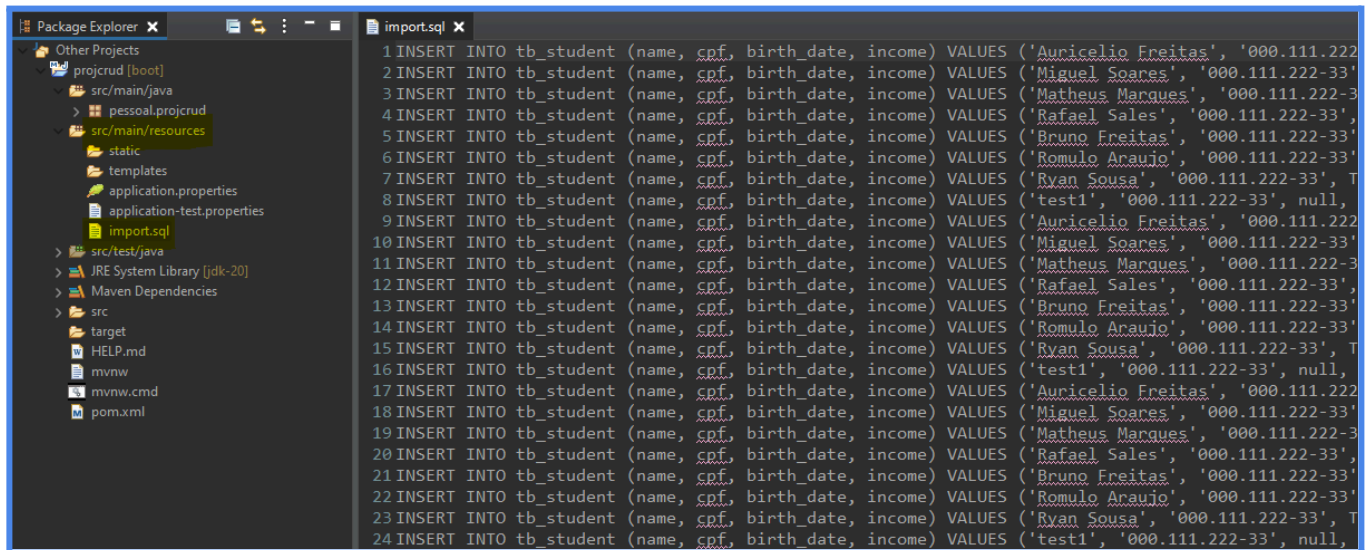
Implementar a busca paginada, no StudentController

```
StudentController.java X
1 package pessoal.projcrud.controllers;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.data.domain.Page;
5 import org.springframework.data.domain.Pageable;
6 import org.springframework.http.ResponseEntity;
7 import org.springframework.web.bind.annotation.GetMapping;
8 import org.springframework.web.bind.annotation.PathVariable;
9 import org.springframework.web.bind.annotation.RequestMapping;
10 import org.springframework.web.bind.annotation.RestController;
11
12 import pessoal.projcrud.dto.StudentDTO;
13 import pessoal.projcrud.services.StudentService;
14
15 @RestController
16 @RequestMapping(value = "/students")
17 public class StudentController {
18
19     @Autowired
20     private StudentService service;
21
22     @GetMapping
23     public ResponseEntity<Page<StudentDTO>> findAll(Pageable pageable) {
24
25         Page<StudentDTO> list = service.findAllPaged(pageable);
26
27         return ResponseEntity.ok().body(list);
28     }
29 }
```

## Ajustar a busca paginada, no StudentService

```
StudentService.java x
1 package pessoal.projcrud.services;
2
3 import java.util.Optional;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.data.domain.Page;
7 import org.springframework.data.domain.Pageable;
8 import org.springframework.stereotype.Service;
9 import org.springframework.transaction.annotation.Transactional;
10
11 import pessoal.projcrud.dto.StudentDTO;
12 import pessoal.projcrud.entities.Student;
13 import pessoal.projcrud.repositories.StudentRepository;
14 import pessoal.projcrud.services.exceptions.ResourceNotFoundException;
15
16 @Service
17 public class StudentService {
18
19     @Autowired
20     private StudentRepository repository;
21
22     @Transactional(readonly = true)
23     public Page<StudentDTO> findAllPaged(Pageable pageable) {
24
25         Page<Student> list = repository.findAll(pageable);
26
27         return list.map(x -> new StudentDTO(x));
28     }
29 }
```

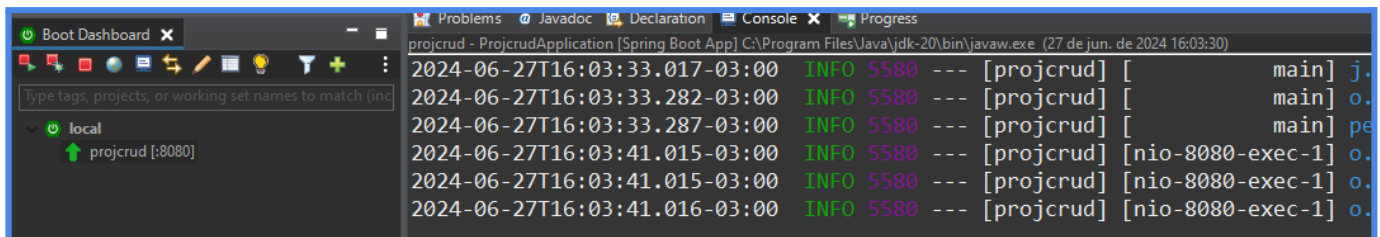
## Expandir o seed do banco para teste de paginação



```
1 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Auricelio Freitas', '000.111.222-33')
2 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Miguel Soares', '000.111.222-33')
3 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Matheus Marques', '000.111.222-33')
4 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Rafael Sales', '000.111.222-33')
5 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Bruno Freitas', '000.111.222-33')
6 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Romulo Araujo', '000.111.222-33')
7 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Ryan Sousa', '000.111.222-33')
8 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('test1', '000.111.222-33', null, 0)
9 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Auricelio Freitas', '000.111.222-33')
10 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Miguel Soares', '000.111.222-33')
11 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Matheus Marques', '000.111.222-33')
12 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Rafael Sales', '000.111.222-33')
13 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Bruno Freitas', '000.111.222-33')
14 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Romulo Araujo', '000.111.222-33')
15 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Ryan Sousa', '000.111.222-33')
16 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('test1', '000.111.222-33', null, 0)
17 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Auricelio Freitas', '000.111.222-33')
18 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Miguel Soares', '000.111.222-33')
19 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Matheus Marques', '000.111.222-33')
20 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Rafael Sales', '000.111.222-33')
21 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Bruno Freitas', '000.111.222-33')
22 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Romulo Araujo', '000.111.222-33')
23 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('Ryan Sousa', '000.111.222-33')
24 INSERT INTO tb_student (name, cpf, birth_date, income) VALUES ('test1', '000.111.222-33', null, 0)
```

**NOTA:** Neste caso, apenas replicar os existentes (copiar, colar).

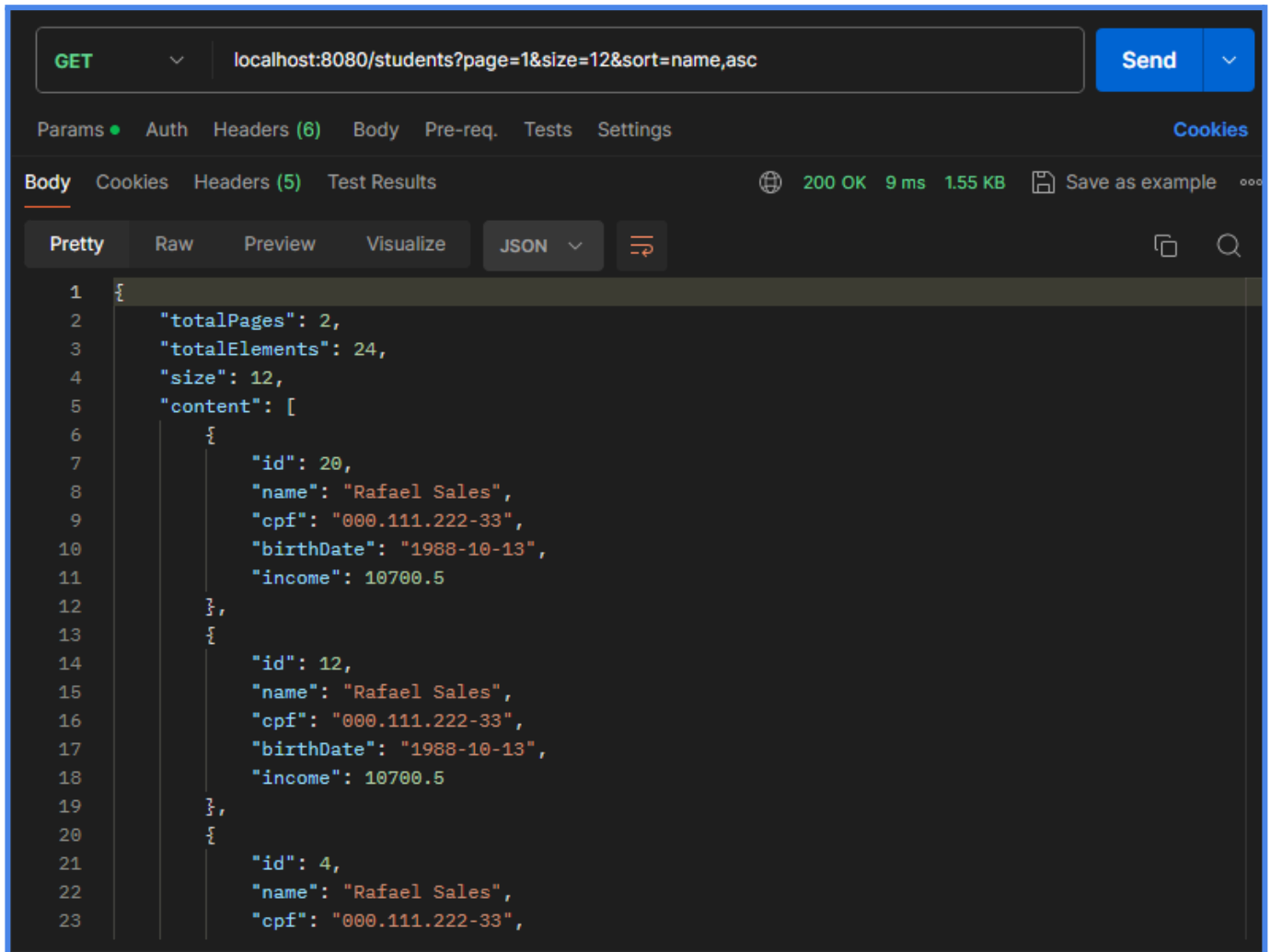
## Rodar o projeto



```
2024-06-27T16:03:33.017-03:00 INFO 5580 --- [projcrud] [main] j...
2024-06-27T16:03:33.282-03:00 INFO 5580 --- [projcrud] [main] o...
2024-06-27T16:03:33.287-03:00 INFO 5580 --- [projcrud] [main] pe...
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o...
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o...
2024-06-27T16:03:41.016-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o...
```

## TESTAR NO POSTMAN

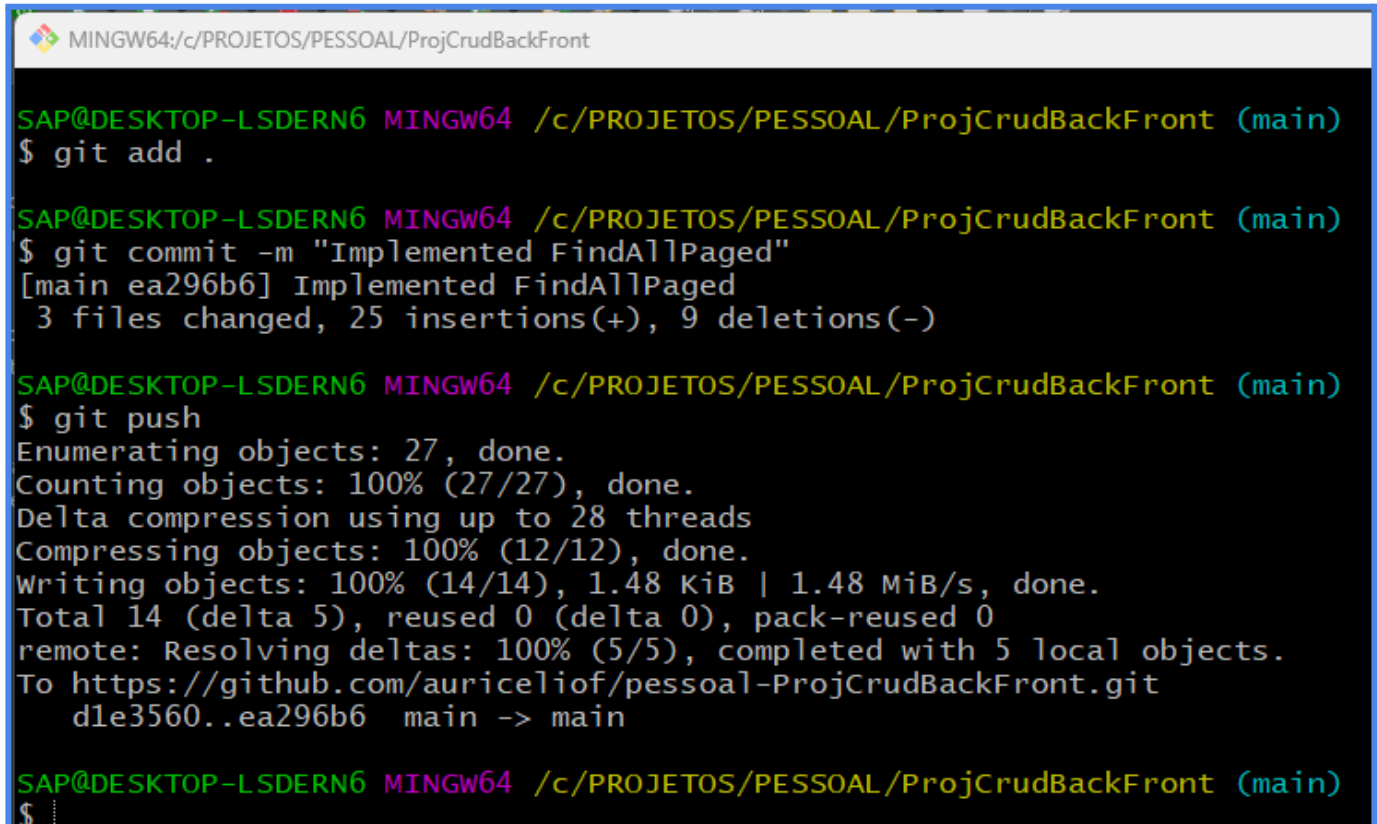
- *GET: localhost:8080/students?page=0&size=12&sort=name,asc*





## Github-11

- “Git bash here” no diretório do projeto
  - git add backend
  - git commit -m “Implemented FindAllPaged”
  - git push



```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git add .

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git commit -m "Implemented FindAllPaged"
[main ea296b6] Implemented FindAllPaged
3 files changed, 25 insertions(+), 9 deletions(-)

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git push
Enumerating objects: 27, done.
Counting objects: 100% (27/27), done.
Delta compression using up to 28 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (14/14), 1.48 KiB | 1.48 MiB/s, done.
Total 14 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/auriceliof/pessoal-ProjCrudBackFront.git
d1e3560..ea296b6  main -> main

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ !
```

## ENDPOINT - INSERT

### INSERIR NOVO ALUNO COM POST

#### Implementar o insert, no StudentController

```
StudentController.java x
40 @PostMapping
41 public ResponseEntity<StudentDTO> insert(@RequestBody StudentDTO dto) {
42
43     dto = service.insert(dto);
44
45     return ResponseEntity.ok().body(dto);
46 }
47 }
```

#### Implementar a metodologia REST ao método

URI uri = ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}").buildAndExpand(dto.getId()).toUri();

```
StudentController.java x
42
43 @PostMapping
44 public ResponseEntity<StudentDTO> insert(@RequestBody StudentDTO dto) {
45
46     dto = service.insert(dto);
47
48     URI uri = ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}")
49         .buildAndExpand(dto.getId()).toUri();
50
51     return ResponseEntity.created(uri).body(dto);
52 }
53 }
```

**OBS:** Importar o URI do "java.net"

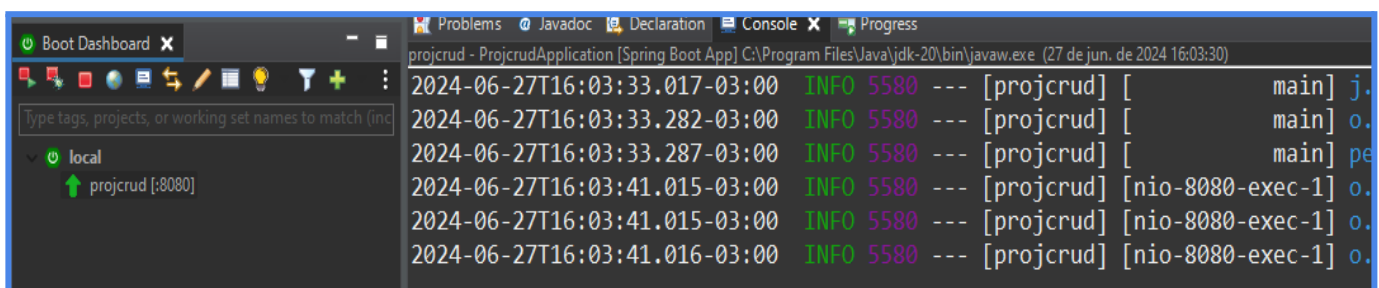
**NOTA:** Implantamos o caminho no Header da requisição e corrigimos o retorno de 200 (padrão) para 201 (recomendação REST), com o created.

Implementar o método insert, convertendo o DTO para uma entidade, no StudentService

```
StudentService.java X
38
39 @Transactional
40 public StudentDTO insert(StudentDTO dto) {
41
42     Student entity = new Student();
43
44     entity.setName(dto.getName());
45     entity.setCpf(dto.getCpf());
46     entity.setBirthDate(dto.getBirthDate());
47     entity.setIncome(dto.getIncome());
48
49     entity = repository.save(entity);
50
51     return new StudentDTO(entity);
52 }
53 }
```

**NOTA:** Não colocar o atributo ID, pois é o banco que irá autoincrementar.

## Rodar o projeto



The screenshot shows the Spring Boot IDE interface. On the left, the 'Boot Dashboard' shows the project 'projcrud' running on port 8080. The main console window displays the following output:

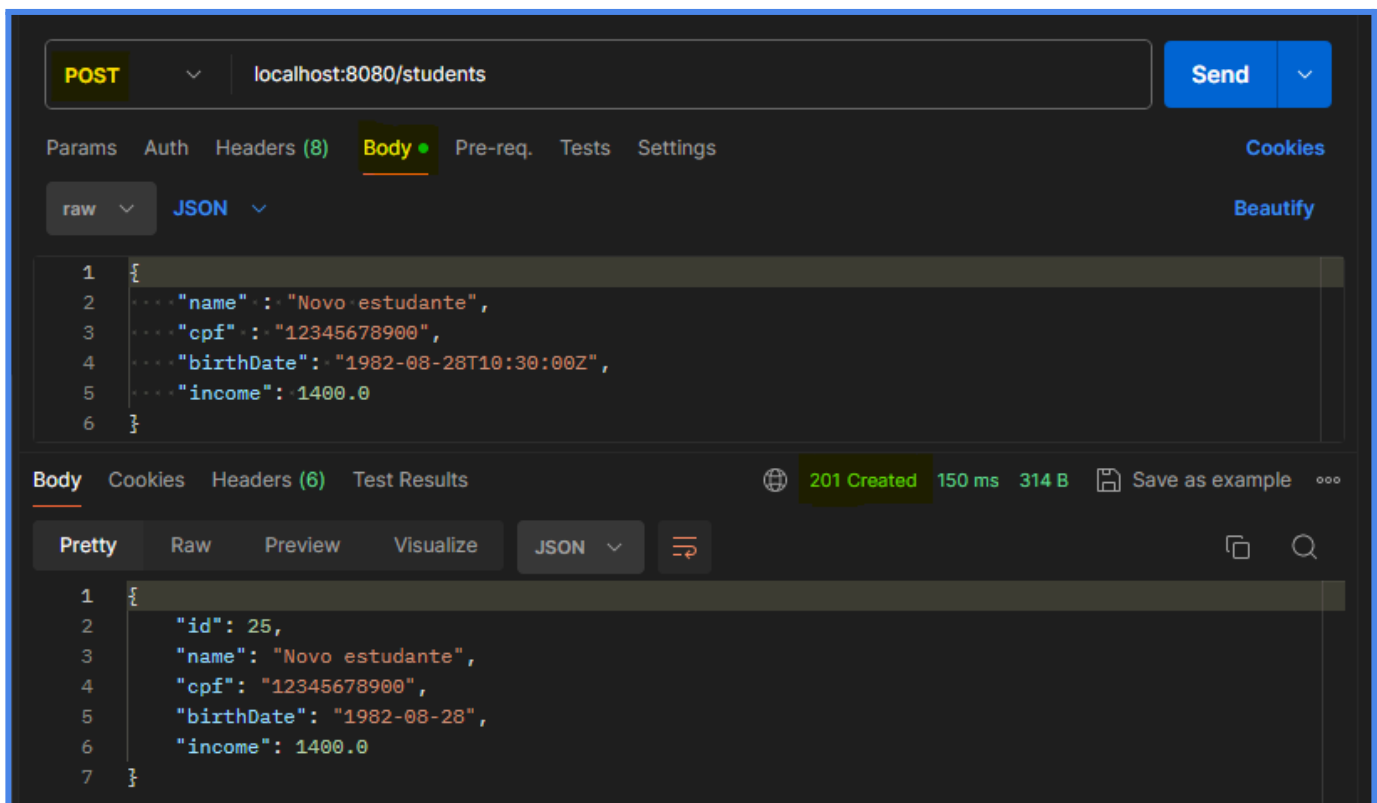
```
projcrud - ProjcrudApplication [Spring Boot App] C:\Program Files\Java\jdk-20\bin\javaw.exe (27 de jun. de 2024 16:03:30)
2024-06-27T16:03:33.017-03:00 INFO 5580 --- [projcrud] [main] j.
2024-06-27T16:03:33.282-03:00 INFO 5580 --- [projcrud] [main] o.
2024-06-27T16:03:33.287-03:00 INFO 5580 --- [projcrud] [main] pe
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.016-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
```

## TESTAR NO POSTMAN

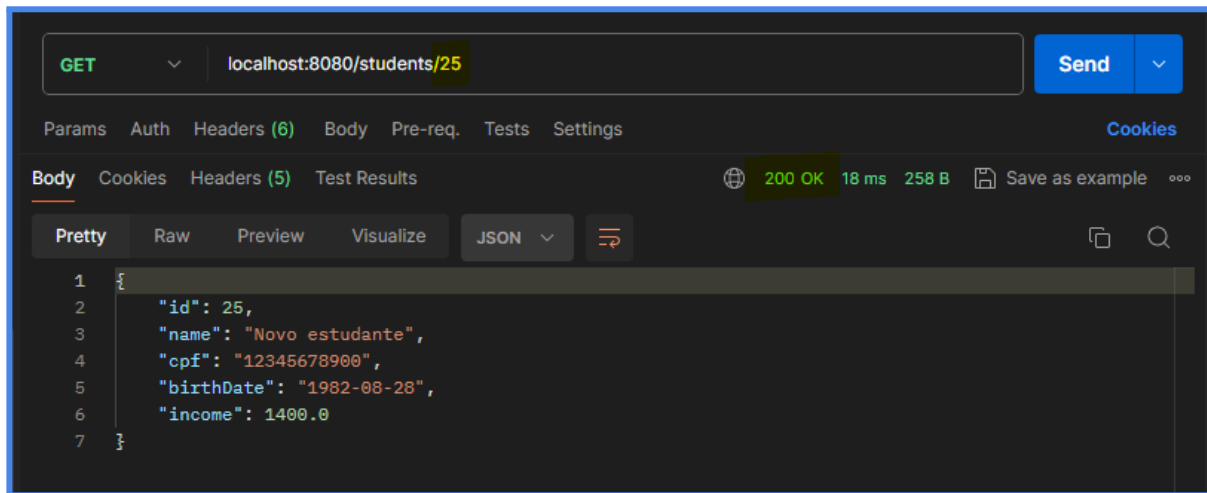
- **POST: localhost:8080/students**
  - **Body:**

```
{
  "name": "Novo aluno",
  "cpf": "123.456.789-00",
  "birthDate": "2005-01-12T10:28:00Z",
  "income": 1920.0
}
```

### Inserir



## Buscar por Id



## Github-12

- “Git bash here” no diretório do projeto
  - git add backend
  - git commit -m “Implemented Insert”
  - git push

```
MINGW64:/c/PROJETOS/PESSOAL/ProjCrudBackFront

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git add .

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git commit -m "Implemented Insert"
[main 86ad26a] Implemented Insert
2 files changed, 44 insertions(+)

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git push
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 28 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (12/12), 1.29 KiB | 1.29 MiB/s, done.
Total 12 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/auriceliof/pessoal-ProjCrudBackFront.git
ea296b6..86ad26a  main -> main
```

## ENDPOINT - UPDATE

### ATUALIZAR ALUNO COM PUT

Implementar o update, no StudentController

```
StudentController.java x
54
55 @PutMapping(value =("/{id}")
56 public ResponseEntity<StudentDTO> update(@PathVariable Long id, @RequestBody StudentDTO dto) {
57
58     dto = service.update(id, dto);
59
60     return ResponseEntity.ok().body(dto);
61 }
62 }
```

Implementar o método update, no StudentService

```
StudentService.java x
53
54 public StudentDTO update(Long id, StudentDTO dto) {
55
56     Student entity = repository.getReferenceById(id);
57
58     entity.setName(dto.getName());
59     entity.setCpf(dto.getCpf());
60     entity.setBirthDate(dto.getBirthDate());
61     entity.setIncome(dto.getIncome());
62
63     entity = repository.save(entity);
64
65     return new StudentDTO(entity);
66 }
67 }
68 }
```

Rodar o projeto

```
Boot Dashboard x
type tags, projects, or working set names to match (inc
local
projcrud [:8080]

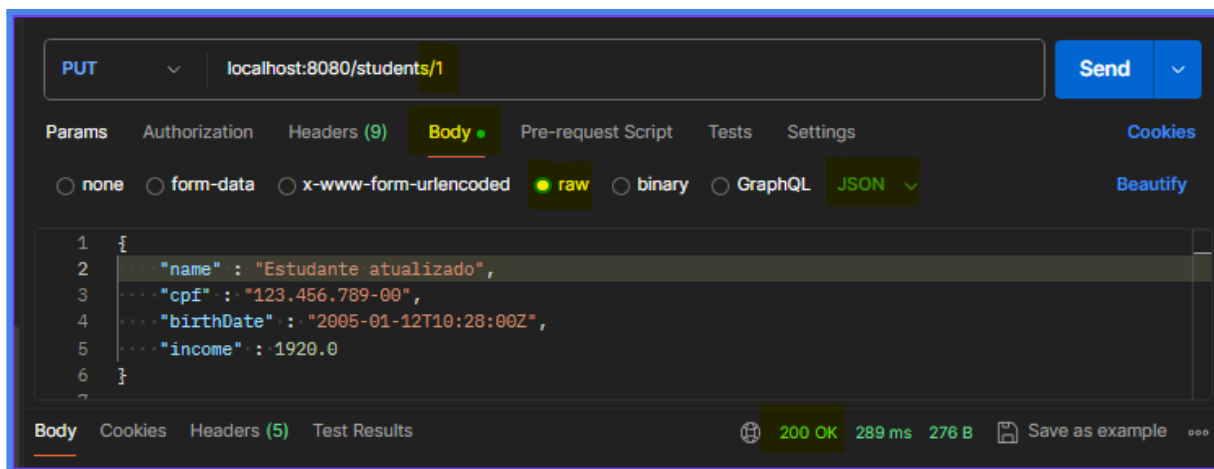
Problems Javadoc Declaration Console x Progress
projcrud - ProjcrudApplication [Spring Boot App] C:\Program Files\Java\jdk-20\bin\javaw.exe (27 de jun. de 2024 16:03:30)
2024-06-27T16:03:33.017-03:00 INFO 5580 --- [projcrud] [main] j.
2024-06-27T16:03:33.282-03:00 INFO 5580 --- [projcrud] [main] o.
2024-06-27T16:03:33.287-03:00 INFO 5580 --- [projcrud] [main] pe
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.016-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
```

## TESTAR NO POSTMAN

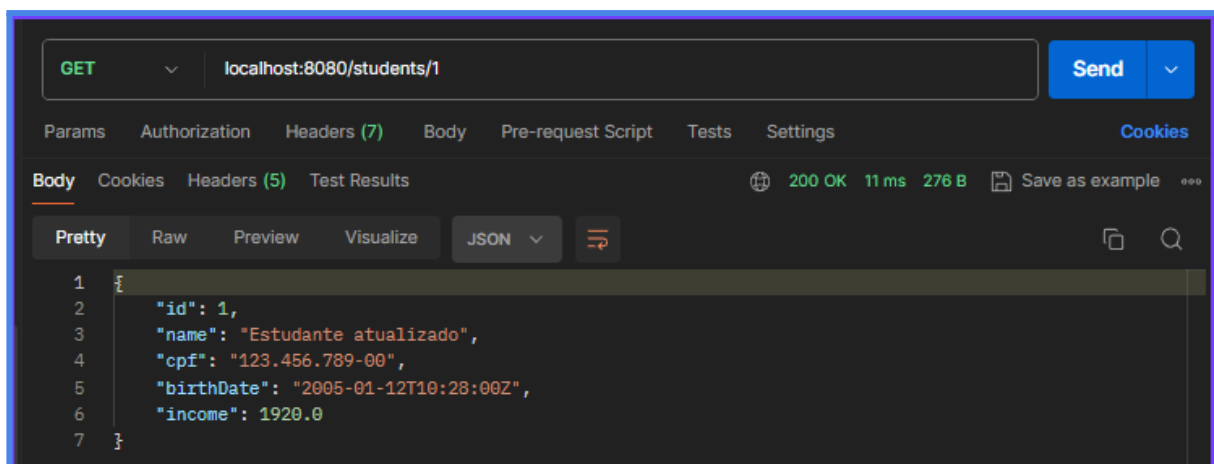
- **PUT: localhost:8080/students/1**
  - **Body:**

```
{  
  "name": "Aluno atualizado",  
  "cpf": "123.456.789-00",  
  "birthDate": "2005-01-12T10:28:00Z",  
  "income": 1920.0  
}
```

### Update



### Busca por ID



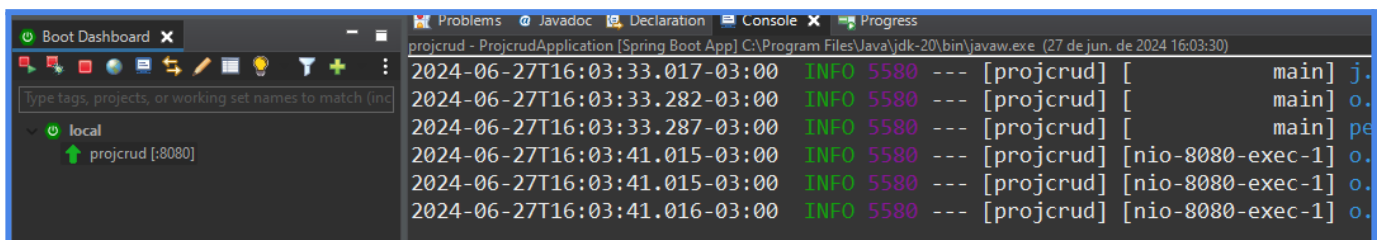
## TRATAMENTO DE ERRO PARA O UPDATE

### Implementar o tratamento para ID Não encontrado

```
StudentService.java x
55 @Transactional
56 public StudentDTO update(Long id, StudentDTO dto) {
57     try {
58         Student entity = repository.getReferenceById(id);
59
60         entity.setName(dto.getName());
61         entity.setCpf(dto.getCpf());
62         entity.setBirthDate(dto.getBirthDate());
63         entity.setIncome(dto.getIncome());
64
65         entity = repository.save(entity);
66
67         return new StudentDTO(entity);
68     }
69     catch (EntityNotFoundException e) {
70         throw new ResourceNotFoundException("ID not found: " + id);
71     }
72 }
73 }
```

NOTA: Devemos colocar o método update num bloco “Try Catch”, pois ao atualizar um ID, este pode não existir.

### Rodar o projeto



```
Boot Dashboard x
type tags, projects, or working set names to match (inc
local
projcrud [:8080]

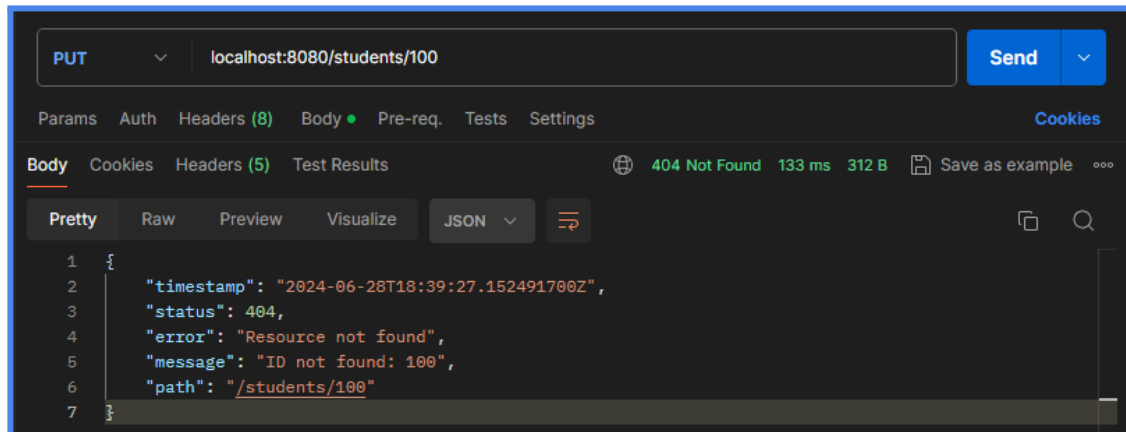
Problems Javadoc Declaration Console X Progress
projcrud - ProjcrudApplication [Spring Boot App] C:\Program Files\Java\jdk-20\bin\javaw.exe (27 de jun. de 2024 16:03:30)
2024-06-27T16:03:33.017-03:00 INFO 5580 --- [projcrud] [main] j.
2024-06-27T16:03:33.282-03:00 INFO 5580 --- [projcrud] [main] o.
2024-06-27T16:03:33.287-03:00 INFO 5580 --- [projcrud] [main] pe
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.016-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
```



## TESTAR NO POSTMAN

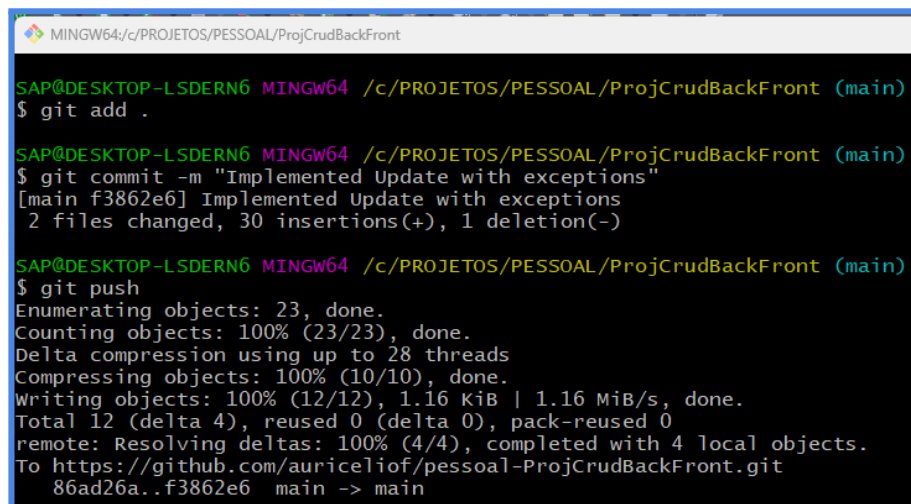
- *PUT: localhost:8080/students/100*

### Update



## Github-13

- “Git bash here” no diretório do projeto
  - `git add backend`
  - `git commit -m “Implemented Update with exceptions”`
  - `git push`



## ENDPOINT - DELETE

### DELETAR UM ALUNO COM O MÉTODO REST DELETE

Implementar o delete, no StudentController

```
StudentController.java X
63
64 @DeleteMapping(value =("/{id}")
65 public ResponseEntity<StudentDTO> delete(@PathVariable Long id) {
66
67     service.delete(id);
68
69     return ResponseEntity.noContent().build();
70 }
71 }
72
```

Implementar o método delete, no StudentService

```
StudentService.java X
73
74 public void delete(Long id) {
75
76     repository.deleteById(id);
77 }
78 }
79
```

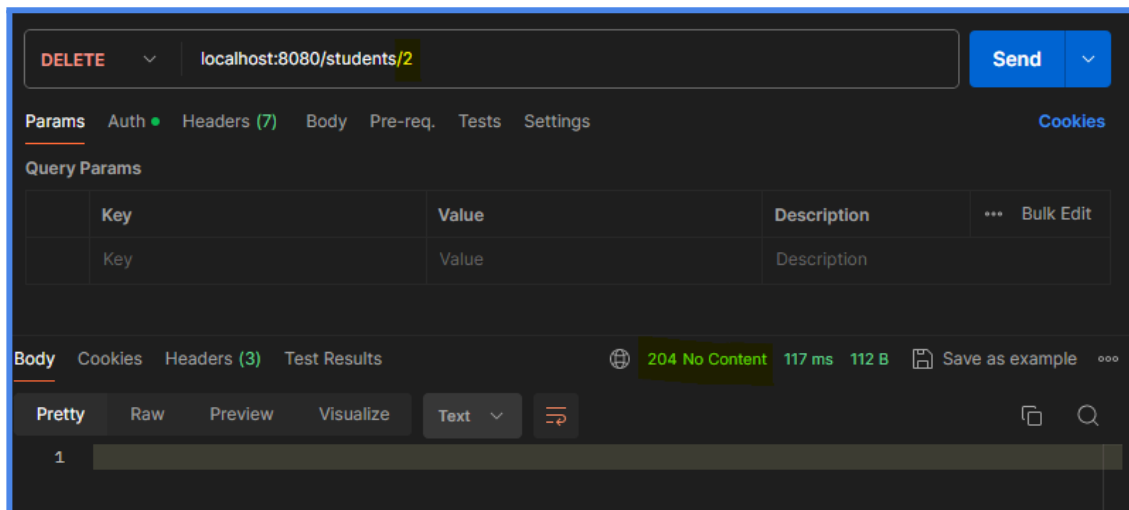
**NOTA:** No método delete não colocamos o `@Transactional`.

Rodar o projeto

```
Boot Dashboard X
Type tags, projects, or working set names to match (inc
local
projcrud [:8080]

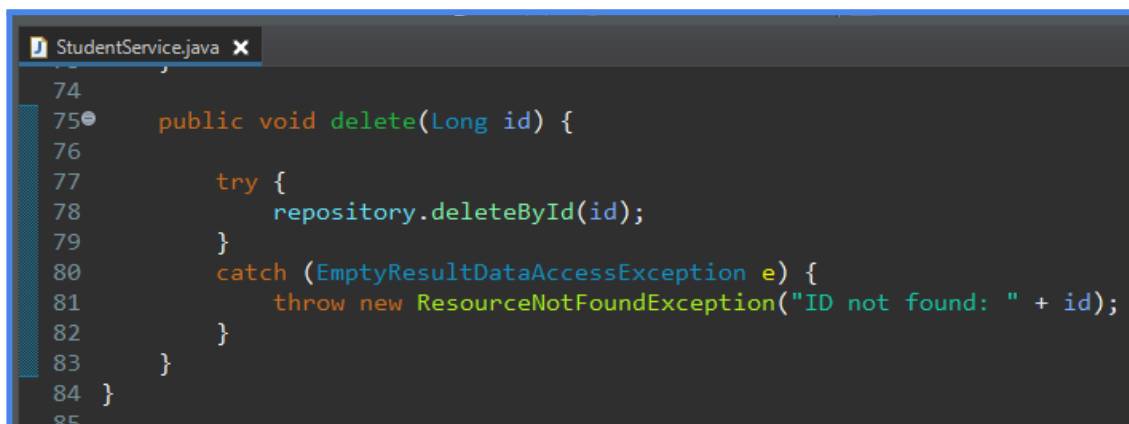
Problems Javadoc Declaration Console X Progress
projcrud - ProjcrudApplication [Spring Boot App] C:\Program Files\Java\jdk-20\bin\javaw.exe (27 de jun, de 2024 16:03:30)
2024-06-27T16:03:33.017-03:00 INFO 5580 --- [projcrud] [main] j.
2024-06-27T16:03:33.282-03:00 INFO 5580 --- [projcrud] [main] o.
2024-06-27T16:03:33.287-03:00 INFO 5580 --- [projcrud] [main] pe
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.016-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
```

## TESTAR NO POSTMAN



## TRATAMENTO DE ERRO DO DELETE

### Implementar o tratamento para ID Não encontrado



**NOTA:** Devemos colocar o método delete num bloco “Try Catch”, pois ao deletar um ID, este pode não existir.

**OBS:** Neste projeto, não iremos implementar o tratamento de erro para “integridade referencial”, pois a classe student não se relaciona com nenhuma outra, ao menos por enquanto.

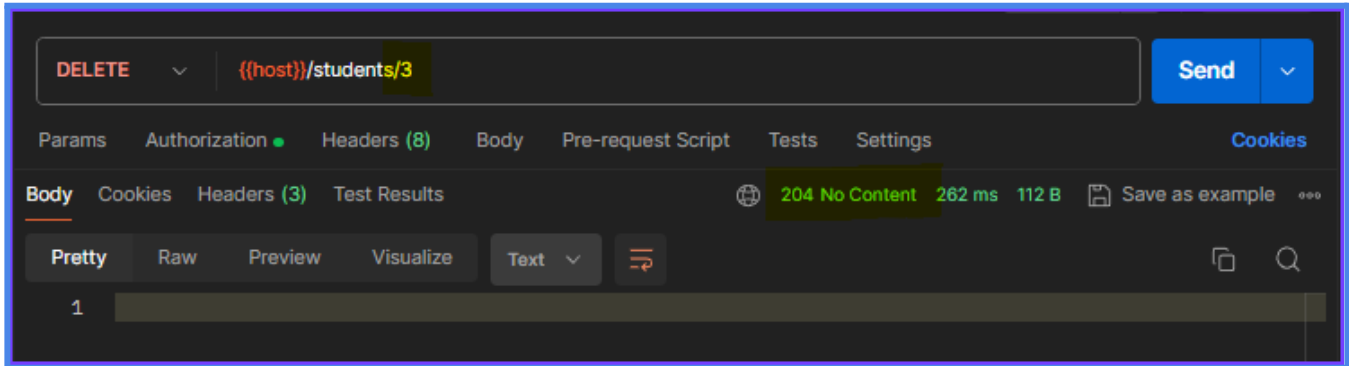
## Rodar o projeto

```

2024-06-27T16:03:33.017-03:00 INFO 5580 --- [projcrud] [main] j.
2024-06-27T16:03:33.282-03:00 INFO 5580 --- [projcrud] [main] o.
2024-06-27T16:03:33.287-03:00 INFO 5580 --- [projcrud] [main] pe
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.016-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.

```

## TESTAR NO POSTMAN



## Github-14

- “Git bash here” no diretório do projeto
  - git add backend
  - git commit -m “Implemented Delete”
  - git push

```

MINGW64/c:/PROJETOS/PESSOAL/ProjCrudBackFront
SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git add .

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git commit -m "Implemented Delete"
[main 1d8929a] Implemented Delete
2 files changed, 27 insertions(+)

SAP@DESKTOP-LSDERN6 MINGW64 /c/PROJETOS/PESSOAL/ProjCrudBackFront (main)
$ git push
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 28 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (12/12), 1.11 KiB | 1.11 MiB/s, done.
Total 12 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/auriceliof/pessoal-ProjCrudBackFront.git
f3862e6..1d8929a main -> main

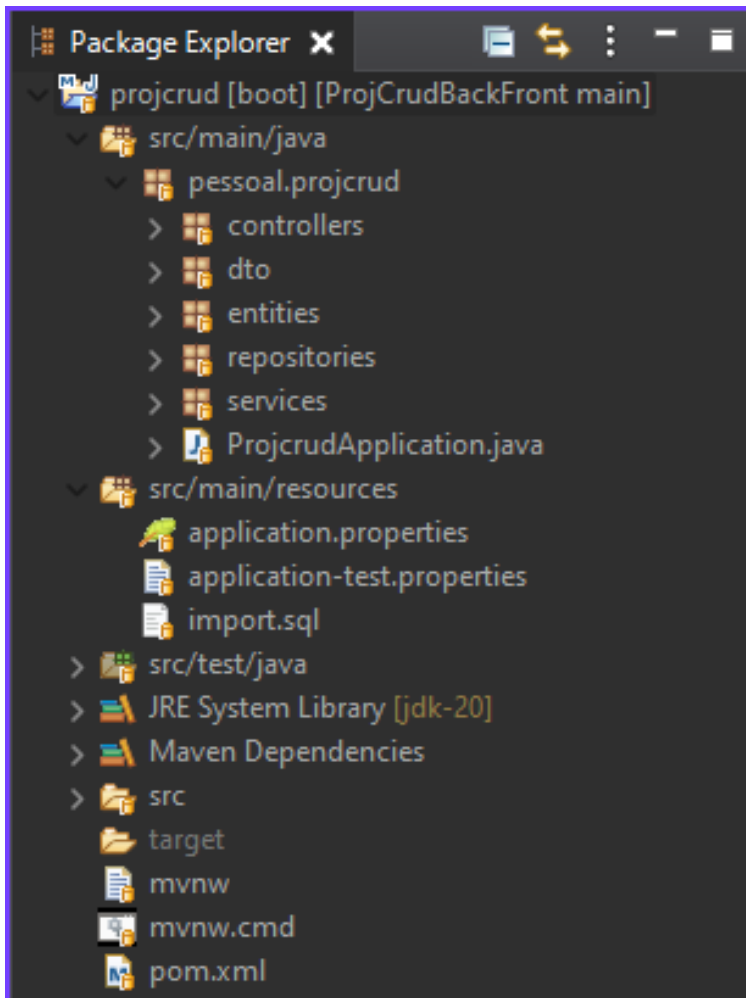
```

## **IMPLEMENTAR O SWAGGER**

O objetivo deste capítulo é implementar o swagger no Spring Tool Suite 4.

O Swagger é uma ferramenta amplamente utilizada para criar, visualizar e manter a documentação de APIs de maneira automatizada.

### **PROJETO**



*NOTA: Neste momento, o projeto já deve estar em funcionamento, com todos os endpoints respondendo.*

## IMPLEMENTAÇÃO DO SWAGGER

### DEPENDÊNCIA MAVEN

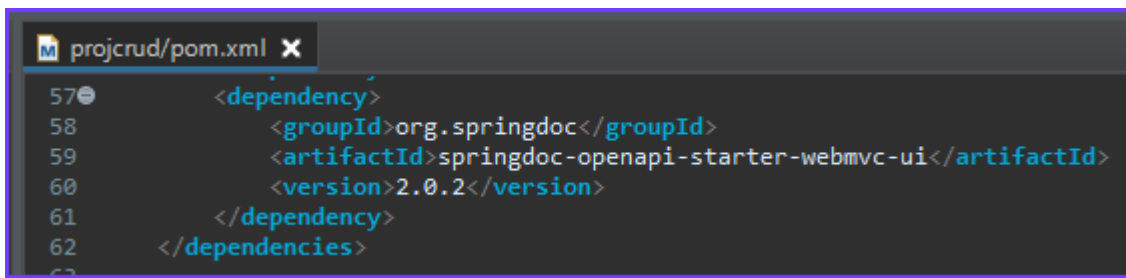
<dependency>

<groupId>org.springdoc</groupId>

<artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>

<version>2.0.2</version>

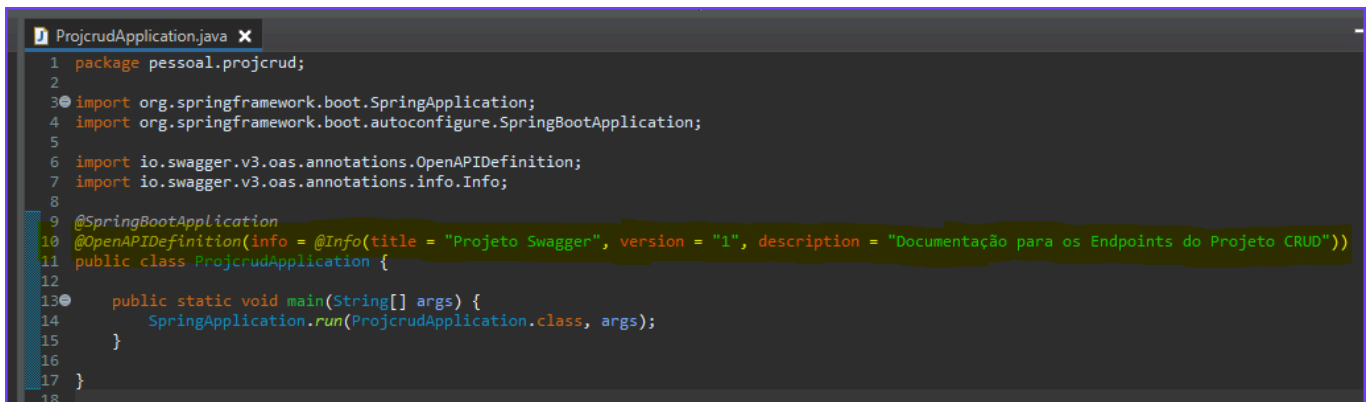
</dependency>



```
projcrud/pom.xml x
57 <dependency>
58     <groupId>org.springdoc</groupId>
59     <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
60     <version>2.0.2</version>
61 </dependency>
62 </dependencies>
63
```

### IMPLEMENTAR O MAIN PRINCIPAL

@OpenAPIDefinition(info = @Info(title = "Projeto Swagger", version = "1", description = "Documentação para os Endpoints do Projeto CRUD"))



```
ProjcrudApplication.java x
1 package pessoal.projcrud;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 import io.swagger.v3.oas.annotations.OpenAPIDefinition;
7 import io.swagger.v3.oas.annotations.info.Info;
8
9 @SpringBootApplication
10 @OpenAPIDefinition(info = @Info(title = "Projeto Swagger", version = "1", description = "Documentação para os Endpoints do Projeto CRUD"))
11 public class ProjcrudApplication {
12
13     public static void main(String[] args) {
14         SpringApplication.run(ProjcrudApplication.class, args);
15     }
16 }
17
18
```

## IMPLEMENTAR O CONTROLLER

### FindAll

*@Operation(summary = "Busca todos os alunos", method = "GET")*

```
StudentController.java X
32
33 @Operation(summary = "Busca todos os alunos", method = "GET")
34 @GetMapping()
35 public ResponseEntity<Page<StudentDTO>> findAll(Pageable pageable){
36
37     Page<StudentDTO> list = service.findAllPaged(pageable);
38
39     return ResponseEntity.ok().body(list);
40 }
41
```

### FindById

*@Operation(summary = "Busca alunos po ID", method = "GET")*

```
StudentController.java X
41
42 @Operation(summary = "Busca alunos po ID", method = "GET")
43 @GetMapping(value =("/{id}")
44 public ResponseEntity<StudentDTO> findById(@PathVariable Long id){
45
46     StudentDTO dto = service.findById(id);
47
48     return ResponseEntity.ok().body(dto);
49 }
50
```

### Insert

*@Operation(summary = "Cadastra novos alunos", method = "POST")*

```
*StudentController.java X
50
51 @Operation(summary = "Cadastra novos alunos", method = "POST")
52 @PostMapping
53 public ResponseEntity<StudentDTO> insert(@RequestBody StudentDTO dto){
54
55     dto = service.insert(dto);
56
57     URI uri = ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}").buildAndExpand(dto.getId()).toUri();
58
59     return ResponseEntity.created(uri).body(dto);
60 }
61
```

### Update

`@Operation(summary = "Atualiza aluno existente", method = "PUT")`

```
*StudentController.java X
62 @Operation(summary = "Atualiza aluno existente", method = "PUT")
63 @PutMapping(value =("/{id}")
64 public ResponseEntity<StudentDTO> update(@PathVariable Long id, @RequestBody StudentDTO dto){
65
66     dto = service.update(id, dto);
67
68     return ResponseEntity.ok().body(dto);
69 }
70
```

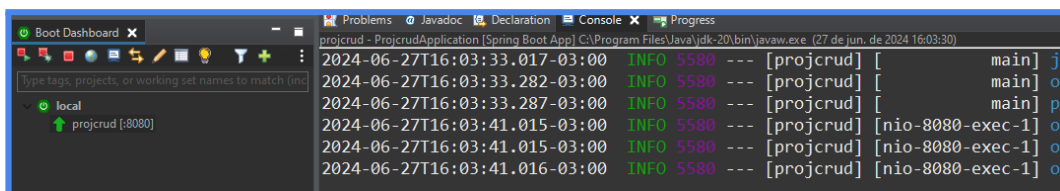
### Delete

`@Operation(summary = "Deleta aluno existente", method = "DELETE")`

```
*StudentController.java X
71 @Operation(summary = "Deleta aluno existente", method = "DELETE")
72 @DeleteMapping(value =("/{id}")
73 public ResponseEntity<StudentDTO> delete(@PathVariable Long id){
74
75     service.delete(id);
76
77     return ResponseEntity.noContent().build();
78 }
79 }
80
```

## ACESSAR O SWAGGER

### Rodar o projeto



```
Boot Dashboard X
local
projcrud [8080]

2024-06-27T16:03:33.017-03:00 INFO 5580 --- [projcrud] [main] j.
2024-06-27T16:03:33.282-03:00 INFO 5580 --- [projcrud] [main] o.
2024-06-27T16:03:33.287-03:00 INFO 5580 --- [projcrud] [main] pe
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.015-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
2024-06-27T16:03:41.016-03:00 INFO 5580 --- [projcrud] [nio-8080-exec-1] o.
```



## Acesso online

- <http://localhost:8080/swagger-ui/index.html#>

The screenshot displays the Swagger UI interface in a web browser. The address bar shows the URL `localhost:8080/swagger-ui/index.html#`. The Swagger logo is in the top left, and the API path `/v3/api-docs` is in the top right. The main title is **Projeto Swagger** with a version tag `1.0.0`. Below the title, it says "Documentação para os Endpoints do Projeto CRUD". A "Servers" section shows a dropdown menu with the selected server `http://localhost:8080 - Generated server url`.

The **student-controller** section is expanded, showing five API endpoints:

- GET** `/students/{id}`: Busca alunos po ID
- PUT** `/students/{id}`: Atualiza aluno existente
- DELETE** `/students/{id}`: Deleta aluno existente
- GET** `/students`: Busca todos os alunos
- POST** `/students`: Cadastra novos alunos

The **Schemas** section is also expanded, listing five data models:

- StudentDTO
- Pageable
- PageStudentDTO
- PageableObject
- SortObject

## Testar o Swagger

Buscar todos

GET /students Busca todos os alunos

Parameters

Name	Description
<b>pageable</b> <small>required</small> object (query)	<pre>{   "page": 0,   "size": 1,   "sort": [     "string"   ] }</pre>

Execute

Clear

200 OK

Media type  
application/json

Controls Accept header

Example Value | Schema

```
{
  "totalElements": 0,
  "totalPages": 0,
  "size": 0,
  "content": [
    {
      "id": 0,
      "name": "string",
      "cpf": "string",
      "birthDate": "2024-06-30",
      "income": 0
    }
  ],
  "number": 0,
  "sort": {
    "unsorted": true,
    "empty": true,
    "sorted": true
  },
  "pageable": {
    "unpaged": true,
    "offset": 0,
    "sort": {
      "unsorted": true,
      "empty": true,
      "sorted": true
    },
    "paged": true,
    "pageNumber": 0,
    "pageSize": 0
  }
}
```

Busca por ID

GET

/students/{id}

Busca alunos po ID

Cancel

Parameters

Name	Description
id <small>* required</small>	
integer(\$int4)	
(path)	

2

Execute

Clear

Responses

Curl

curl -X 'GET' \nhttps://localhost:8080/students/2' \n-H 'accept: \*/\*'

Request URL

http://localhost:8080/students/2

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{\n  "id": 2,\n  "name": "Miguel Soares",\n  "cpf": "800.111.222-33",\n  "birthdate": "1989-02-10",\n  "income": 20700.5\n}</pre></div><div>Download</div></div> <div><div>Response headers</div><div><pre>connection: keep-alive\ncontent-type: application/json\ndate: Sun, 30 Jun 2024 15:05:14 GMT\nkeep-alive: timeout=60\ntransfer-encoding: chunked</pre></div></div>

Responses

Code	Description
------	-------------

Links

## Atualizar

PUT /students/{id} Atualiza aluno existente

Parameters

Name	Description
id <small>required</small>	
integer(\$int32)	
(path)	

Request body required

application/json

```
{
  "id": 0,
  "name": "string",
  "cpf": "string",
  "birthdate": "2020-06-06",
  "income": 0
}
```

Execute

Clear

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:8080/students/0' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 0,
    "name": "string",
    "cpf": "string",
    "birthdate": "2020-06-06",
    "income": 0
  }'
```

Request URL

http://localhost:8080/students/0

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{   "id": 0,   "name": "string",   "cpf": "string",   "birthdate": "2020-06-06",   "income": 0 }</pre></div><div>Download</div></div>

Inserir

POST /students Cadastrar novos alunos

Parameters

No parameters

Request body required

application/json

```
{
  "id": 0,
  "name": "string",
  "cpf": "string",
  "birthdate": "2024-08-30",
  "income": 0
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/students' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 0,
    "name": "string",
    "cpf": "string",
    "birthdate": "2024-08-30",
    "income": 0
  }'
```

Request URL

http://localhost:8080/students

Server response

Code	Details
201	<div><div>Response body</div><div><pre>{   "id": 25,   "name": "string",   "cpf": "string",   "birthdate": "2024-08-30",   "income": 0 }</pre></div><div>Download</div></div>

## Deletar

DELETE /students/{id} Deleta aluno existente

Parameters

Name	Description
id <small>* required</small>	
integer(\$int64)	
(path)	

2

Execute

Clear

Responses

Curl

curl -X 'DELETE' \ 'http://localhost:8080/students/2' \ -H 'accept: \*/\*'

Request URL

http://localhost:8080/students/2

Server response

Code	Details
204	Response headers
Undocumented	connections: keep-alive date: Sun, 30 Jun 2024 15:13:08 GMT keep-alive: timeout=60

Fim