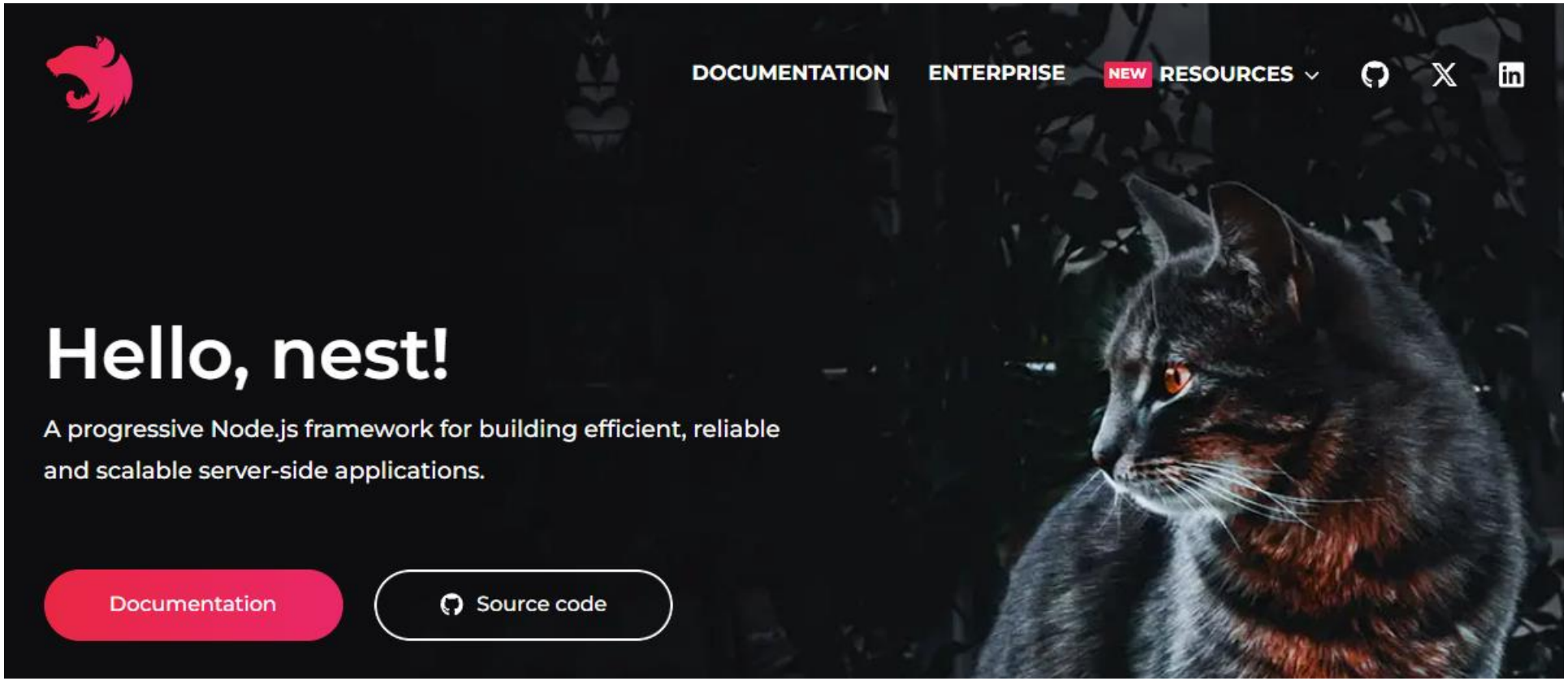
INSTITUTO FEDERAL
PARANÁ

Nest.js



Prof. Dr. Robyson Aggio

www.aggioirati.com.br

INSTITUTO FEDERAL
PARANÁ

Pipe

Prof. Dr. Robyson Aggio

File Edit Selection View Go Run Terminal Help

primeiro-app

EXPLORER

PRIMEIRO-APP

- dist
- node_modules
- src
  - app
  - tasks
    - dto
      - create-task.dto.ts U
      - update-task.dto.ts U
    - entities
      - task.entity.ts U
    - tasks.controller.ts U
    - tasks.module.ts U
    - tasks.service.ts U
  - users
  - app.controller.spec.ts U
  - main.ts U
- test
- .gitignore U
- .prettierrc U
- eslint.config.mjs U
- nest-cli.json U
- package-lock.json U
- package.json U
- README.md U
- tsconfig.build.json U
- tsconfig.json U

main.ts U

src > main.ts > bootstrap

```ts
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app/app.module';
import { ValidationPipe } from '@nestjs/common';

/*
- app.module.ts : É o módulo principal do aplicativo
- app.controller.ts : Define as rotas e lida com as requisições
- app.service.ts : Contém a lógico do negócio, separado do controlador.
 */

//Arquivo que inicia o nosso projeto
async function bootstrap() {
  const app = await NestFactory.create(AppModule);
  app.useGlobalPipes(new ValidationPipe())
  await app.listen(process.env.PORT ?? 3000);
}
bootstrap();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

node

```
PS C:\Aggio\nestjs\primeiro-app>
[20:07:49] Starting compilation in watch mode...

[20:07:51] Found 0 errors. Watching for file changes.

[Nest] 12696  - 25/02/2025, 20:07:52     LOG [NestFactory] Starting Nest application...
[Nest] 12696  - 25/02/2025, 20:07:52     LOG [InstanceLoader] AppModule dependencies initialized +9ms
[Nest] 12696  - 25/02/2025, 20:07:52     LOG [InstanceLoader] TasksModule dependencies initialized +0ms
[Nest] 12696  - 25/02/2025, 20:07:52     LOG [InstanceLoader] UsersModule dependencies initialized +1ms
[Nest] 12696  - 25/02/2025, 20:07:52   ERROR [PackageLoader] The "class-validator" package is missing. Please, make sure to install it to take advantage
 of ValidationPipe.
```

Pipe

Prof. Dr. Robyson Aggio

Global scoped pipes

Since the `ValidationPipe` was created to be as generic as possible, we can realize its full utility by setting it up as a **global-scoped** pipe so that it is applied to every route handler across the entire application.

main.ts JS

```
async function bootstrap() {
  const app = await NestFactory.create(AppModule);
  app.useGlobalPipes(new ValidationPipe());
  await app.listen(process.env.PORT ?? 3000);
}
bootstrap();
```

NOTICE

In the case of hybrid apps the `useGlobalPipes()` method doesn't set up pipes for gateways and microservices. For "standard" (non-hybrid) microservice apps, `useGlobalPipes()` does mount pipes globally.

Pipe

Prof. Dr. Robyson Aggio

INSTITUTO FEDERAL PARANÁ

Selection View Go Run Terminal Help — primeiro-app

EXPLORER

PRIMEIRO-APP

- dist
- node_modules
- src
  - app
  - tasks
    - dto
      - create-task.dto.ts U
      - update-task.dto.ts U
    - entities
      - task.entity.ts U
    - tasks.controller.ts U
    - tasks.module.ts U
    - tasks.service.ts U
  - users
  - app.controller.spec.ts U
  - main.ts U
- test
- .gitignore U
- .prettierrc U
- eslint.config.mjs U
- nest-cli.json U
- package-lock.json U
- package.json U
- README.md U
- tsconfig.build.json U
- tsconfig.json U

OUTLINE

main.ts U

src > main.ts > bootstrap

```ts
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app/app.module';
import { ValidationPipe } from '@nestjs/common';

/*
- app.module.ts : É o módulo principal do aplicativo
- app.controller.ts : Define as rotas e lida com as requisições
- app.service.ts : Contém a lógico do negócio, separado do controlador.
 */

//Arquivo que inicia o nosso projeto
async function bootstrap() {
  const app = await NestFactory.create(AppModule);
  app.useGlobalPipes(new ValidationPipe())
  await app.listen(process.env.PORT ?? 3000);
}
bootstrap();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS — powershell

```
[20:07:49] Starting compilation in watch mode...

[20:07:51] Found 0 errors. Watching for file changes.

[Nest] 12696  - 25/02/2025, 20:07:52     LOG [NestFactory] Starting Nest application...
[Nest] 12696  - 25/02/2025, 20:07:52     LOG [InstanceLoader] AppModule dependencies initialized +9ms
[Nest] 12696  - 25/02/2025, 20:07:52     LOG [InstanceLoader] TasksModule dependencies initialized +0ms
[Nest] 12696  - 25/02/2025, 20:07:52     LOG [InstanceLoader] UsersModule dependencies initialized +1ms
[Nest] 12696  - 25/02/2025, 20:07:52   ERROR [PackageLoader] The "class-validator" package is missing. Please, make sure to install it to take advantage
 of ValidationPipe.
PS C:\Aggio\nestjs\primeiro-app> npm install class-validator

added 4 packages, and audited 808 packages in 3s

144 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Aggio\nestjs\primeiro-app>
```

Pipe

Prof. Dr. Robyson Aggio

Selection View Go Run Terminal Help

PRIMEIRO-APP

- dist
- node_modules
- src
  - app
  - tasks
    - dto
      - create-task.dto.ts U
      - update-task.dto.ts U
    - entities
      - task.entity.ts U
    - tasks.controller.ts U
    - tasks.module.ts U
    - tasks.service.ts U
  - users
  - app.controller.spec.ts U
  - main.ts U
- test
- .gitignore U
- .prettierrc U
- eslint.config.mjs U
- nest-cli.json U
- package-lock.json U
- package.json U
- README.md U
- tsconfig.build.json U
- tsconfig.json U

main.ts U

src > main.ts > bootstrap

```ts
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app/app.module';
import { ValidationPipe } from '@nestjs/common';

/*
- app.module.ts : É o módulo principal do aplicativo
- app.controller.ts : Define as rotas e lida com as requisições
- app.service.ts : Contém a lógico do negócio, separado do controlador.
 */

//Arquivo que inicia o nosso projeto
async function bootstrap() {
  const app = await NestFactory.create(AppModule);
  app.useGlobalPipes(new ValidationPipe())
  await app.listen(process.env.PORT ?? 3000);
}
bootstrap();
```

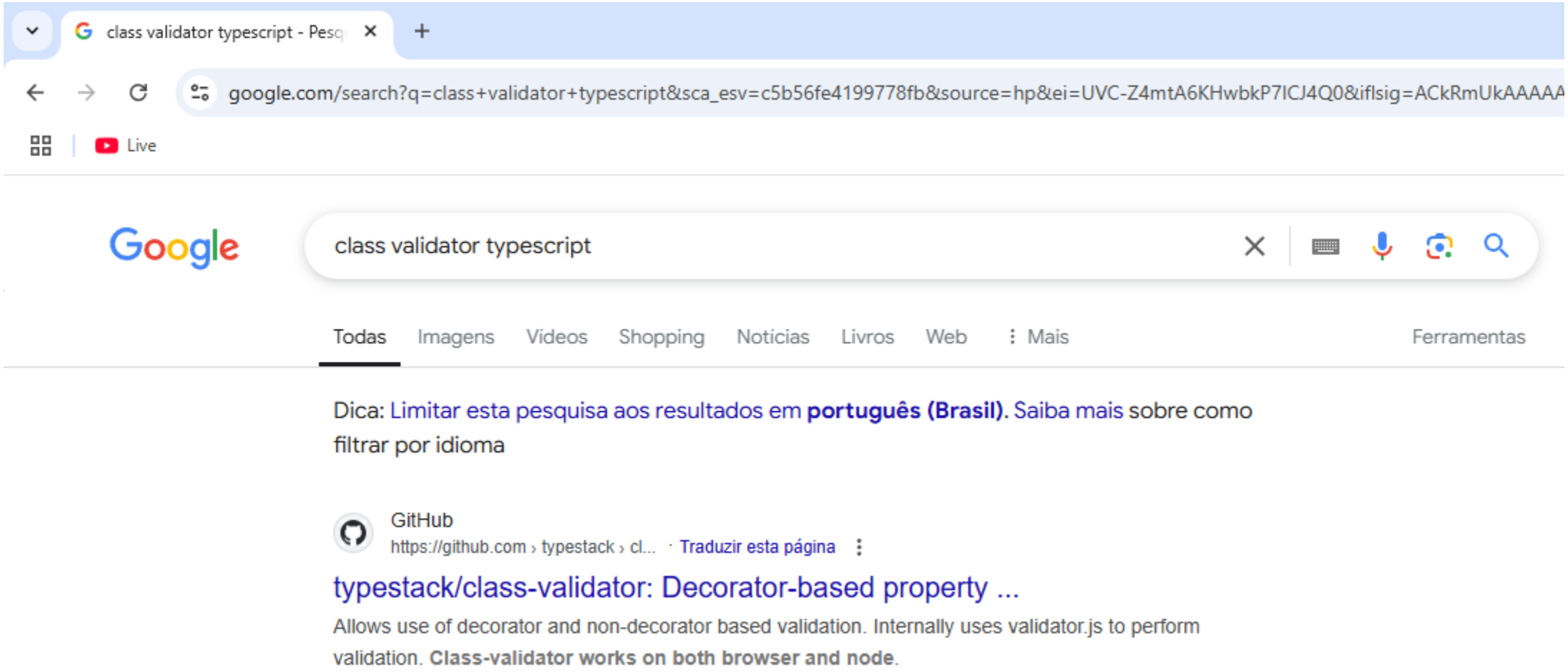
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

powershell

```
[20:11:01] Starting compilation in watch mode...

[20:11:03] Found 0 errors. Watching for file changes.

[Nest] 7684  - 25/02/2025, 20:11:04     LOG [NestFactory] Starting Nest application...
[Nest] 7684  - 25/02/2025, 20:11:04     LOG [InstanceLoader] AppModule dependencies initialized +11ms
[Nest] 7684  - 25/02/2025, 20:11:04     LOG [InstanceLoader] TasksModule dependencies initialized +1ms
[Nest] 7684  - 25/02/2025, 20:11:04     LOG [InstanceLoader] UsersModule dependencies initialized +0ms
[Nest] 7684  - 25/02/2025, 20:11:04   ERROR [PackageLoader] The "class-transformer" package is missing. Please, make sure to install it to take advantage of ValidationPipe.
PS C:\Aggio\nestjs\primeiro-app> npm install class-transformer

added 1 package, and audited 810 packages in 2s

144 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Aggio\nestjs\primeiro-app>
```

class-validator

Prof. Dr. Robyson Aggio



class-validator

README MIT license

Usage

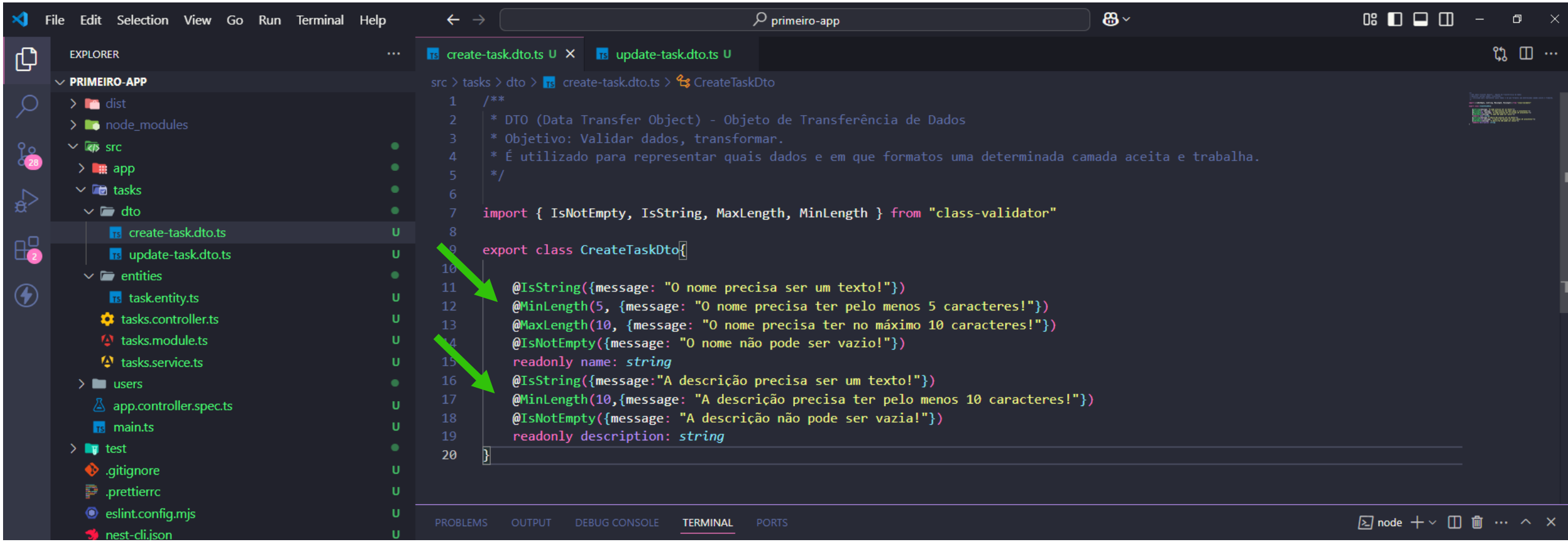
Create your class and put some validation decorators on the properties you want to validate:

```
import {
  validate,
  validateOrReject,
  Contains,
  IsInt,
  Length,
  IsEmail,
  IsFQDN,
  IsDate,
  Min,
  Max,
} from 'class-validator';

export class Post {
  @Length(10, 20)
  title: string;

  @Contains('hello')
  text: string;

  @IsInt()
  @Min(0)
  @Max(10)
  rating: number;

  @IsEmail()
  email: string;

  @IsFQDN()
  site: string;

  @IsDate()
  createDate: Date;
}
```

INSTITUTO FEDERAL
PARANÁ

class-validator

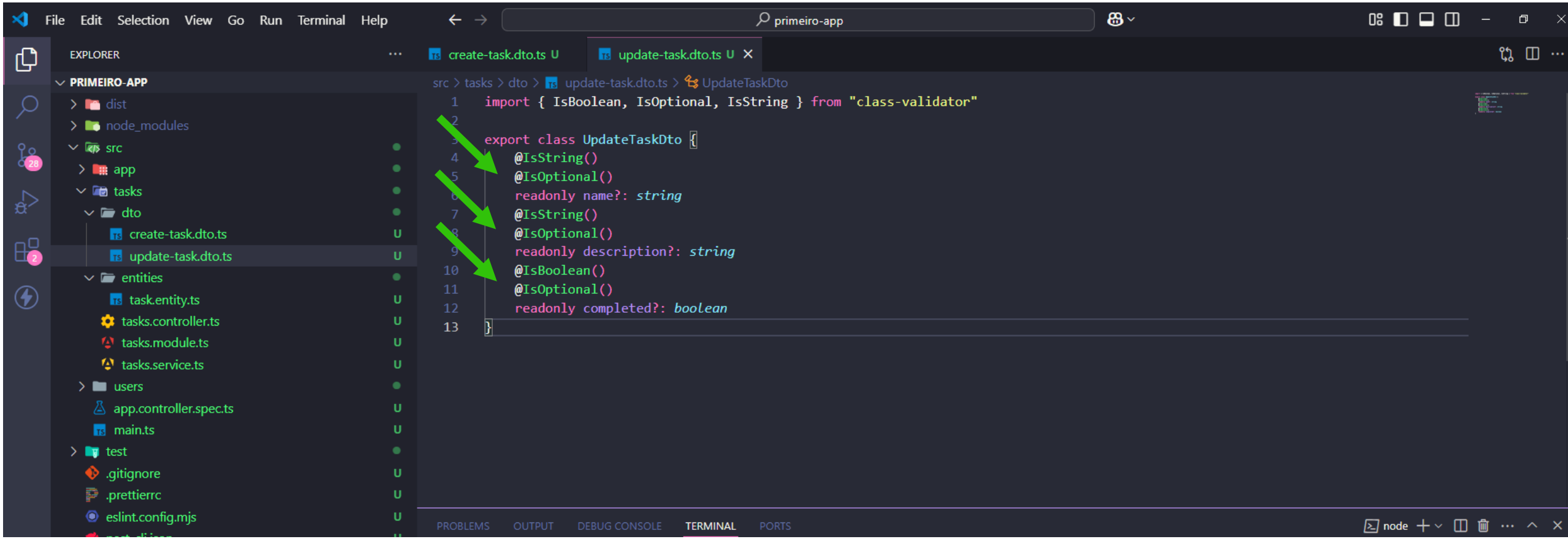
Prof. Dr. Robyson Aggio



INSTITUTO FEDERAL
PARANÁ

class-validator

Prof. Dr. Robyson Aggio



class-validator

Prof. Dr. Robyson Aggio

INSTITUTO FEDERAL PARANÁ

Edit Selection View Go Run Terminal Help

primeiro-app

EXPLORER

PRIMEIRO-APP

- dist
- node_modules
- src
  - app
  - tasks
    - dto
      - create-task.dto.ts U
      - update-task.dto.ts U
    - entities
      - task.entity.ts U
    - tasks.controller.ts U
    - tasks.module.ts U
    - tasks.service.ts U
  - users
  - app.controller.spec.ts U
  - main.ts U
- test
- .gitignore U
- .prettierrc U
- eslint.config.mjs U
- nest-cli.json U
- package-lock.json U

create-task.dto.ts U | update-task.dto.ts U

src > tasks > dto > update-task.dto.ts > UpdateTaskDto

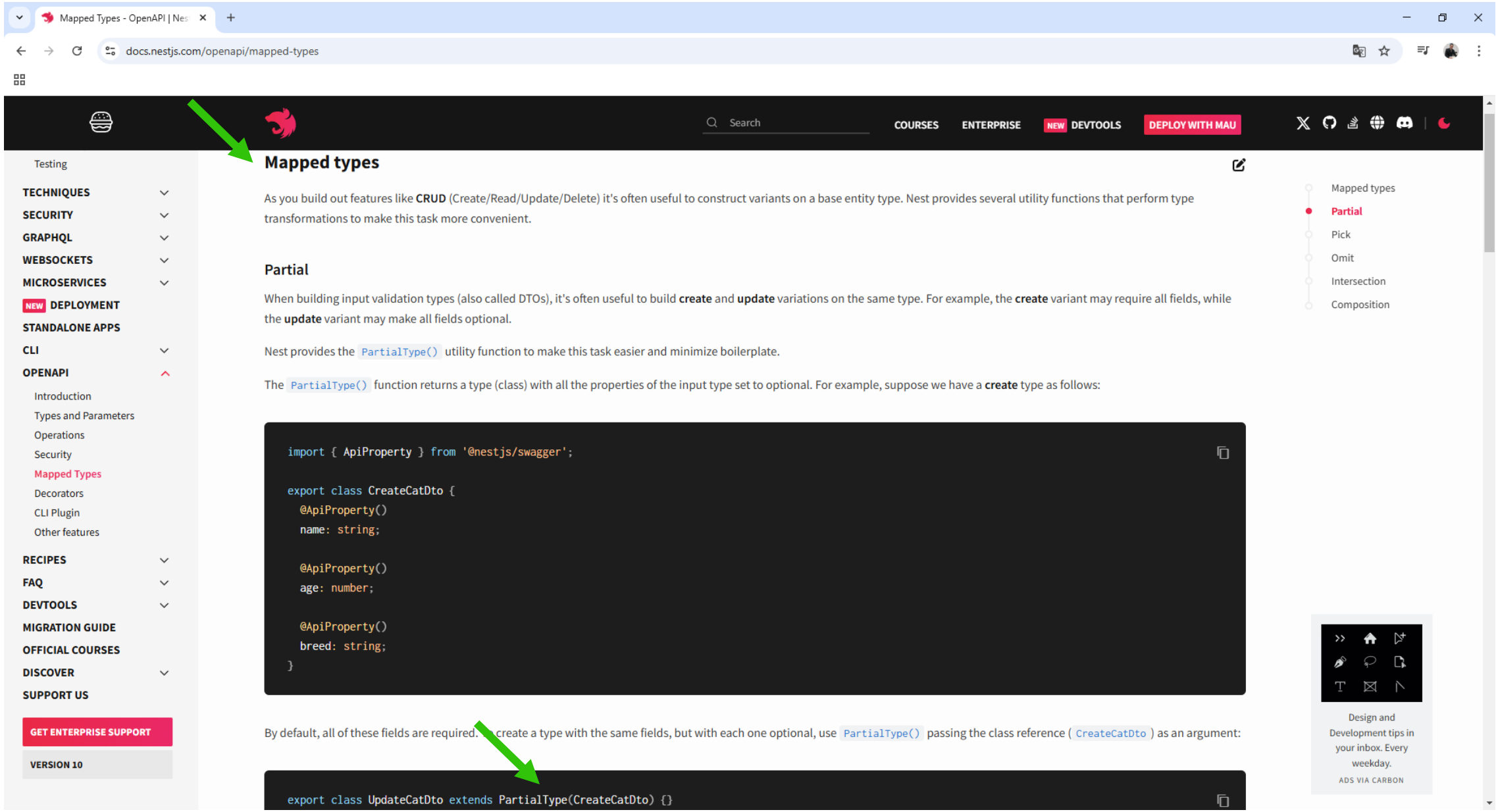
```ts
import { IsBoolean, IsOptional, IsString } from "class-validator"

export class UpdateTaskDto {
    @IsString()
    @IsOptional()
    readonly name?: string
    @IsString()
    @IsOptional()
    readonly description?: string
    @IsBoolean()
    @IsOptional()
    readonly completed?: boolean
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Aggio\nestjs\primeiro-app> npm install @nestjs/mapped-types

up to date, audited 810 packages in 2s

144 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Aggio\nestjs\primeiro-app>
```

partial-type

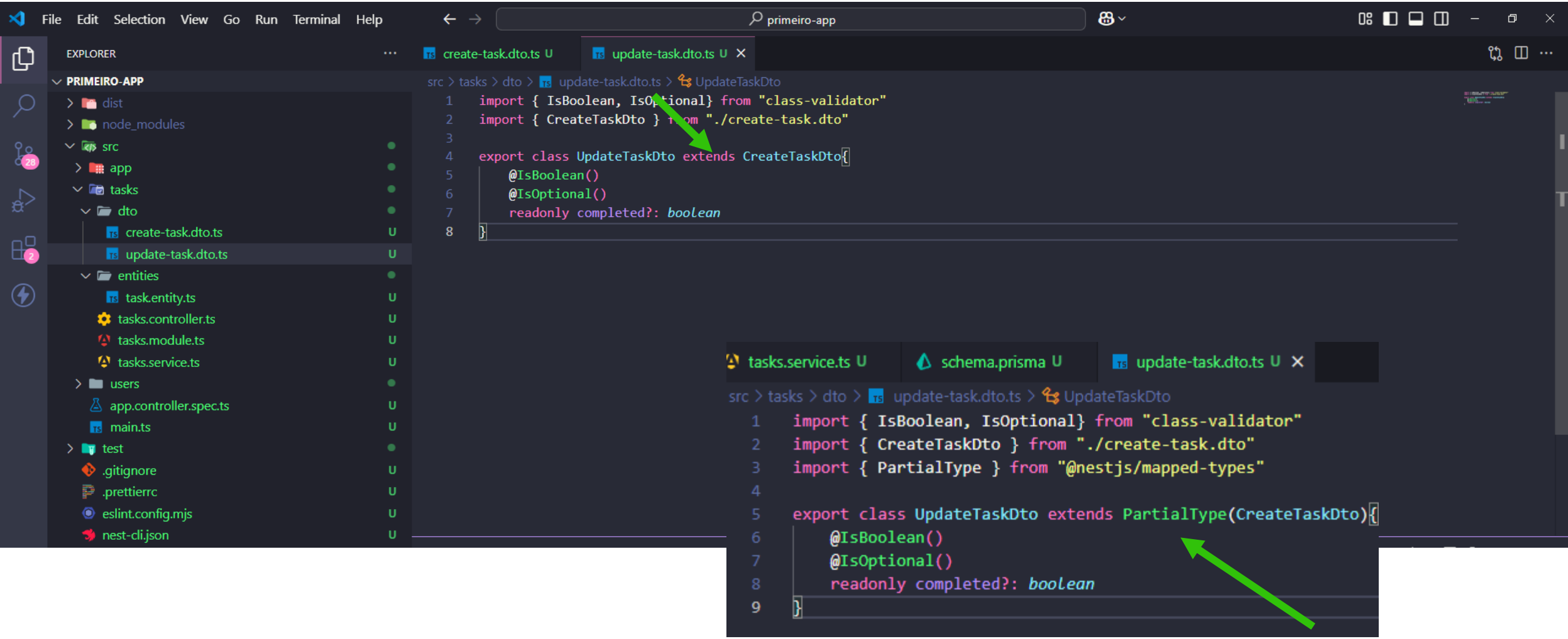
Prof. Dr. Robyson Aggio



INSTITUTO FEDERAL
PARANÁ

partial-type

Prof. Dr. Robyson Aggio



Transformar Dados - Pipe

Prof. Dr. Robyson Aggio

INSTITUTO FEDERAL PARANÁ

File Edit Selection View Go Run Terminal Help

primeiro-app

EXPLORER

PRIMEIRO-APP

- dist
- node_modules
- src
  - app
  - tasks
    - dto
      - create-task.dto.ts U
      - update-task.dto.ts U
    - entities
      - task.entity.ts U
    - tasks.controller.ts U
    - tasks.module.ts U
    - tasks.service.ts U
  - users
  - app.controller.spec.ts U
  - main.ts U
- test
- .gitignore U
- .prettierrc U
- eslint.config.mjs U
- nest-cli.json U
- package-lock.json U
- package.json U
- README.md U
- tsconfig.build.json U
- tsconfig.json U

src > main.ts > bootstrap

```ts
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app/app.module';
import { ValidationPipe } from '@nestjs/common';

/*
- app.module.ts : É o módulo principal do aplicativo
- app.controller.ts : Define as rotas e lida com as requisições
- app.service.ts : Contém a lógico do negócio, separado do controlador.
 */

//Arquivo que inicia o nosso projeto
async function bootstrap() {
  const app = await NestFactory.create(AppModule);
  app.useGlobalPipes(new ValidationPipe({
    whitelist: true //Se o usuário enviar algum parâmetro inexistente!
  }))
  await app.listen(process.env.PORT ?? 3000);
}
bootstrap();
```

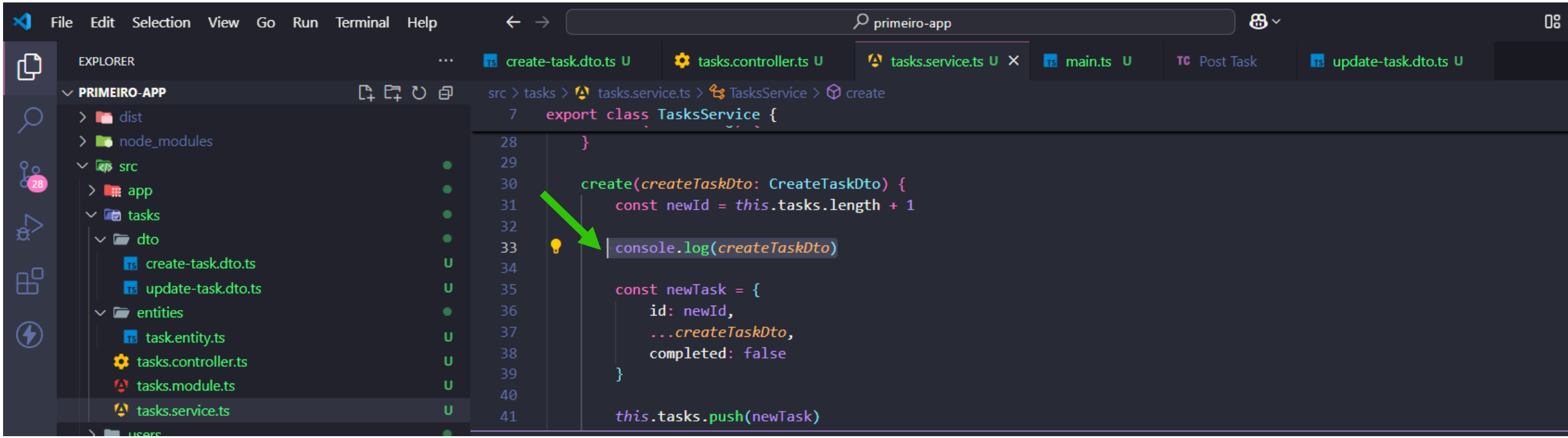
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Aggio\nestjs\primeiro-app> npm install @nestjs/mapped-types

up to date, audited 810 packages in 2s

144 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Aggio\nestjs\primeiro-app>
```

INSTITUTO FEDERAL
PARANÁ

Transformar Dados - Pipe

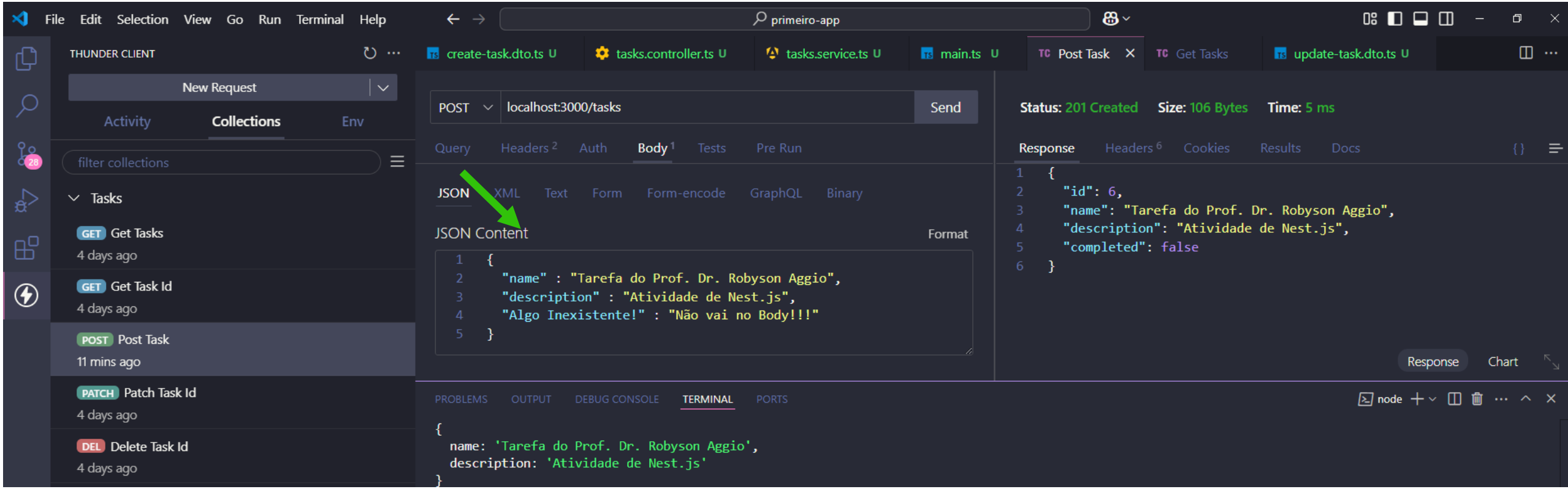
Prof. Dr. Robyson Aggio



INSTITUTO FEDERAL
PARANÁ

Transformar Dados - Pipe

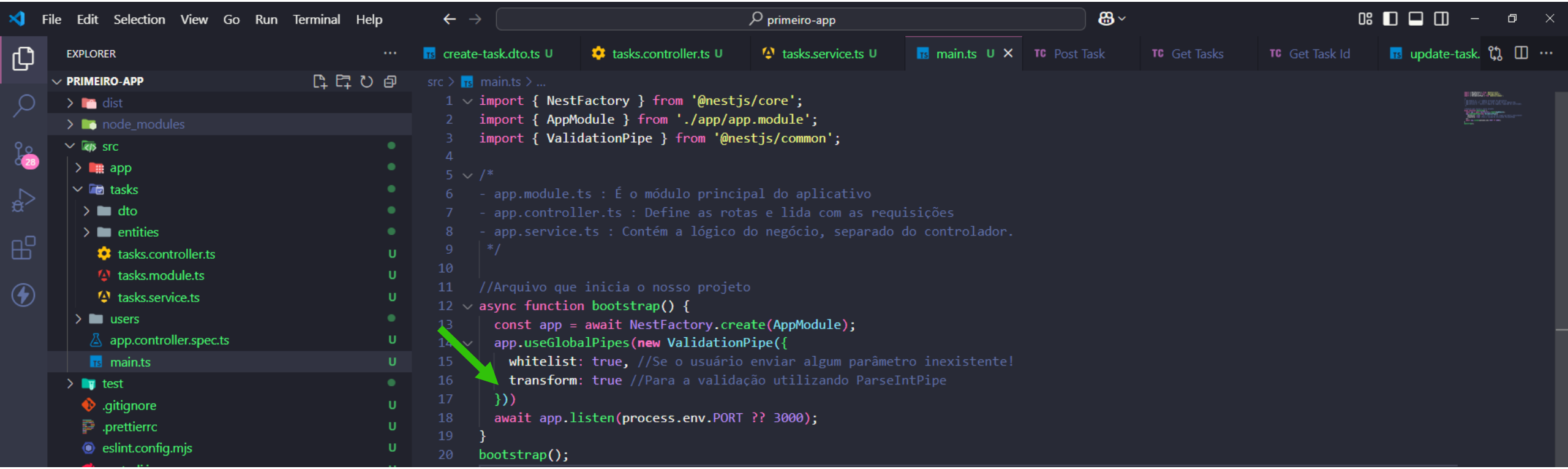
Prof. Dr. Robyson Aggio



INSTITUTO FEDERAL
PARANÁ

Transformar Dados - Pipe

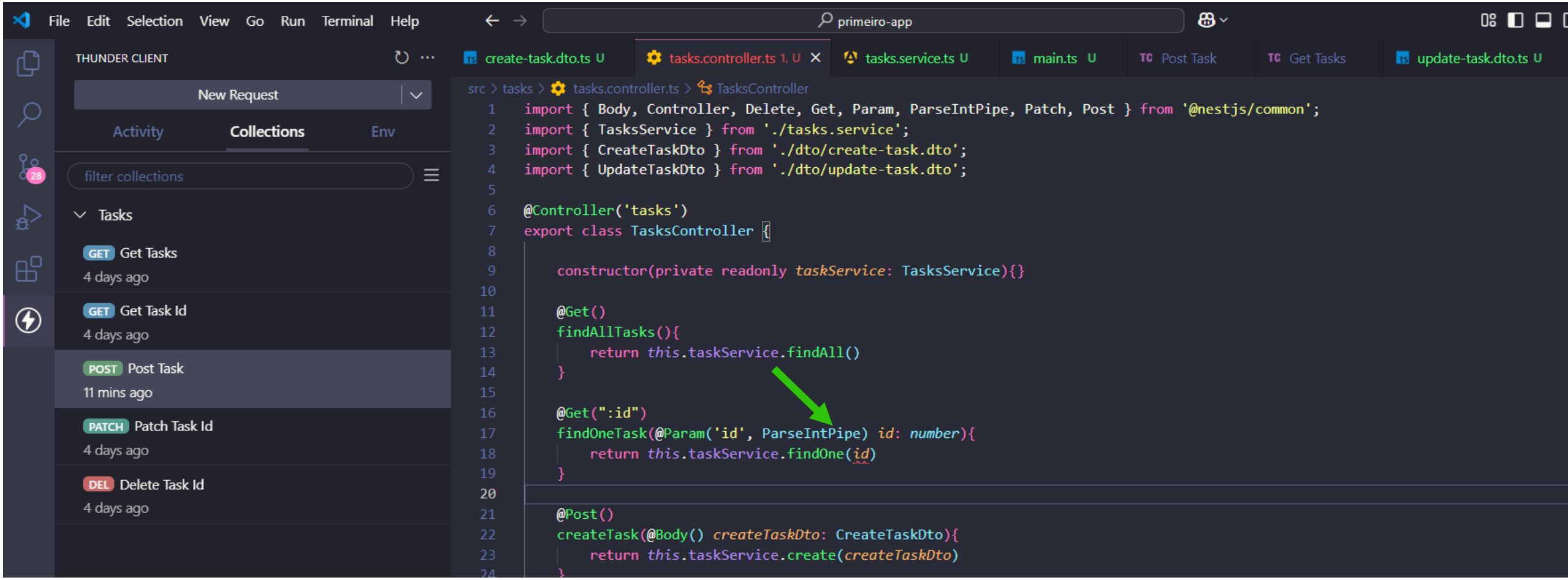
Prof. Dr. Robyson Aggio



INSTITUTO FEDERAL
PARANÁ

Transformar Dados - Pipe

Prof. Dr. Robyson Aggio



INSTITUTO FEDERAL
PARANÁ

Transformar Dados - Pipe

Prof. Dr. Robyson Aggio

File Edit Selection View Go Run Terminal Help

primeiro-app

THUNDER CLIENT

New Request

Activity Collections Env

filter collections

Tasks

GET Get Tasks
4 days ago

GET Get Task Id
4 days ago

POST Post Task
11 mins ago

PATCH Patch Task Id
4 days ago

DEL Delete Task Id
4 days ago

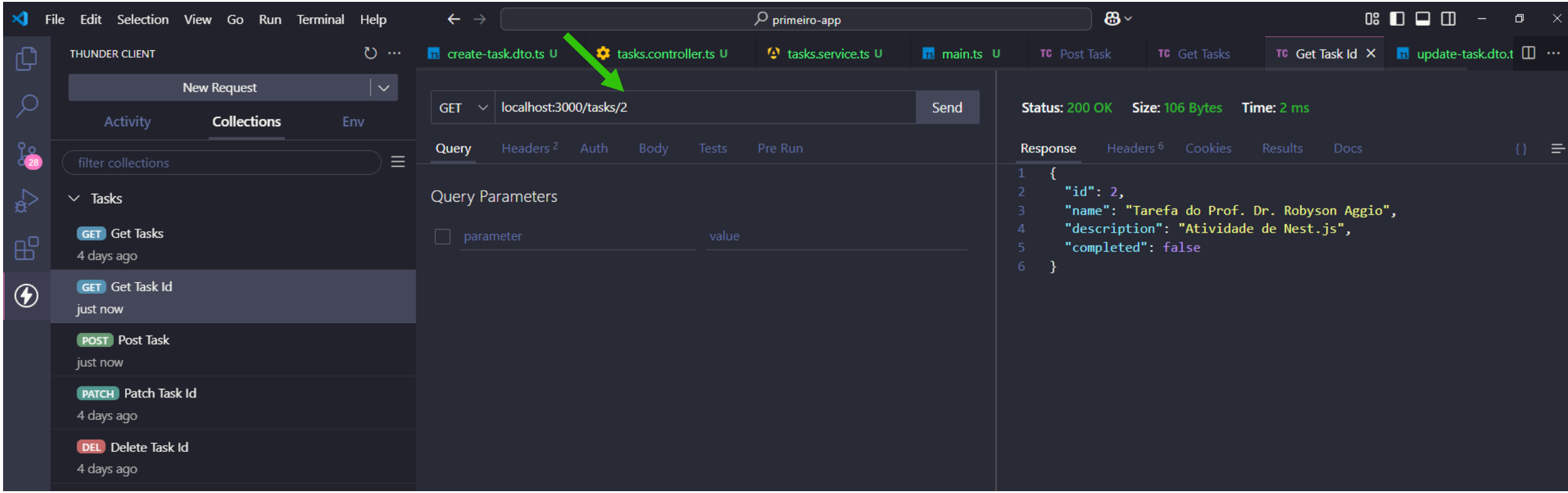
create-task.dto.ts U | tasks.controller.ts U | tasks.service.ts U | main.ts U | Post Task | Get Tasks | update-task.dto.ts U

src > tasks > tasks.service.ts > TasksService

```ts
import { HttpException, HttpStatus, Injectable, NotFoundException } from '@nestjs/common';
import { Task } from './entities/task.entity';
import { CreateTaskDto } from './dto/create-task.dto';
import { UpdateTaskDto } from './dto/update-task.dto';

@Injectable()
export class TasksService {

    private tasks: Task[] = [
        {
            id: 1,
            name: "Nome da Tarefa",
            description: "Descrição da Tarefa",
            completed: false
        }
    ]

    findAll() {
        return this.tasks
    }

    findOne(id: number) {
        const task = this.tasks.find(task => task.id === id)

        if(task) return task

        throw new HttpException("Essa Tarefa Não Existe!", HttpStatus.NOT_FOUND)
    }

    create(createTaskDto: CreateTaskDto) {
```

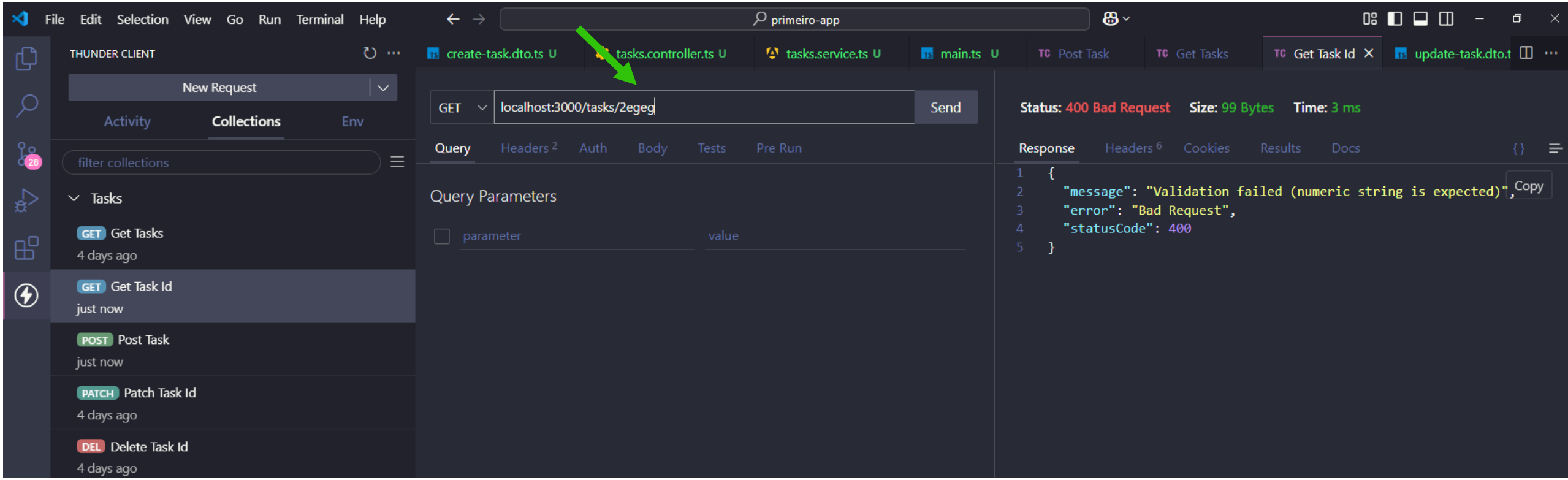
Transformar Dados - Pipe

Prof. Dr. Robyson Aggio



Transformar Dados - Pipe

Prof. Dr. Robyson Aggio



Transformar Dados - Pipe

Prof. Dr. Robyson Aggio

INSTITUTO FEDERAL
PARANÁ

File Edit Selection View Go Run Terminal Help

primeiro-app

EXPLORER

PRIMEIRO-APP

- dist
- node_modules
- src
  - app
  - tasks
    - dto
      - create-task.dto.ts U
      - update-task.dto.ts U
    - entities
      - task.entity.ts U
    - tasks.controller.ts 1, U
    - tasks.module.ts U
    - tasks.service.ts U
  - users
  - app.controller.spec.ts U
  - main.ts U
- test
- .gitignore U
- .prettierrc U
- eslint.config.mjs U
- nest-cli.json U
- package-lock.json U
- package.json U
- README.md U
- tsconfig.build.json U
- tsconfig.json U

create-task.dto.ts U | tasks.controller.ts 1, U | tasks.service.ts U | main.ts U | Post Task | Get Tasks | Get Task Id | update-tas

src > tasks > tasks.controller.ts > TasksController > updateTask

```ts
import { Body, Controller, Delete, Get, Param, ParseIntPipe, Patch, Post } from '@nestjs/common';
import { TasksService } from './tasks.service';
import { CreateTaskDto } from './dto/create-task.dto';
import { UpdateTaskDto } from './dto/update-task.dto';

@Controller('tasks')
export class TasksController {

    constructor(private readonly taskService: TasksService){}

    @Get()
    findAllTasks(){
        return this.taskService.findAll()
    }

    @Get(":id")
    findOneTask(@Param('id', ParseIntPipe) id: number){
        return this.taskService.findOne(id)
    }

    @Post()
    createTask(@Body() createTaskDto: CreateTaskDto){
        return this.taskService.create(createTaskDto)
    }

    @Patch(":id")
    updateTask(@Param("id", ParseIntPipe) id: number, @Body() updateTaskDto: UpdateTaskDto){
        return this.taskService.update(id, updateTaskDto)
    }

    @Delete(":id")
    deleteTask(@Param("id") id: string){
        return this.taskService.delete(id)
    }

}
```

Transformar Dados - Pipe

Prof. Dr. Robyson Aggio

INSTITUTO FEDERAL
PARANÁ

File Edit Selection View Go Run Terminal Help

primeiro-app

EXPLORER

PRIMEIRO-APP

- dist
- node_modules
- src
  - app
  - tasks
    - dto
      - create-task.dto.ts U
      - update-task.dto.ts U
    - entities
      - task.entity.ts U
    - tasks.controller.ts U
    - tasks.module.ts U
    - tasks.service.ts U
  - users
  - app.controller.spec.ts U
  - main.ts U
- test
- .gitignore U
- .prettierrc U
- eslint.config.mjs U
- nest-cli.json U
- package-lock.json U
- package.json U
- README.md U
- tsconfig.build.json U
- tsconfig.json U

create-task.dto.ts U | tasks.controller.ts U | tasks.service.ts U | main.ts U | Post Task | Get Tasks | Get Task Id | update-task.

src > tasks > tasks.service.ts > TasksService > update

```
7   export class TasksService {
44      }
45
46      update(id: number, updateTaskDto: UpdateTaskDto) {
47          const taskIndex = this.tasks.findIndex(task => task.id === id)
48
49          if(taskIndex < 0){
50              throw new HttpException("Essa Tarefa Não Existe!", HttpStatus.NOT_FOUND)
51          }
52
53              const taskIntem = this.tasks[taskIndex]
54
55              this.tasks[taskIndex] = {
56                  ...taskIntem,
57                  ...updateTaskDto
58              }
59
60          return this.tasks[taskIndex]
61      }
62
63      delete(id: string) {
64
65          const taskIndex = this.tasks.findIndex(task => task.id === Number(id))
66
67          if(taskIndex < 0){
68              throw new HttpException("Essa Tarefa Não Existe!", HttpStatus.NOT_FOUND)
69          }
70
71          this.tasks.splice(taskIndex, 1)
72
73          return "Tarefa Excluída com Sucesso!"
74      }
75
76  }
```

Transformar Dados - Pipe

Prof. Dr. Robyson Aggio

INSTITUTO FEDERAL PARANÁ

File Edit Selection View Go Run Terminal Help — primeiro-app

EXPLORER

PRIMEIRO-APP

- dist
- node_modules
- src
  - app
  - tasks
    - dto
      - create-task.dto.ts U
      - update-task.dto.ts U
    - entities
      - task.entity.ts U
    - tasks.controller.ts 1, U
    - tasks.module.ts U
    - tasks.service.ts U
  - users
  - app.controller.spec.ts U
  - main.ts U
- test
- .gitignore U
- .prettierrc U
- eslint.config.mjs U
- nest-cli.json U
- package-lock.json U
- package.json U
- README.md U
- tsconfig.build.json U
- tsconfig.json U

create-task.dto.ts U | tasks.controller.ts 1, U | tasks.service.ts U | main.ts U | Post Task | Get Tasks | Get Task Id | update-tas

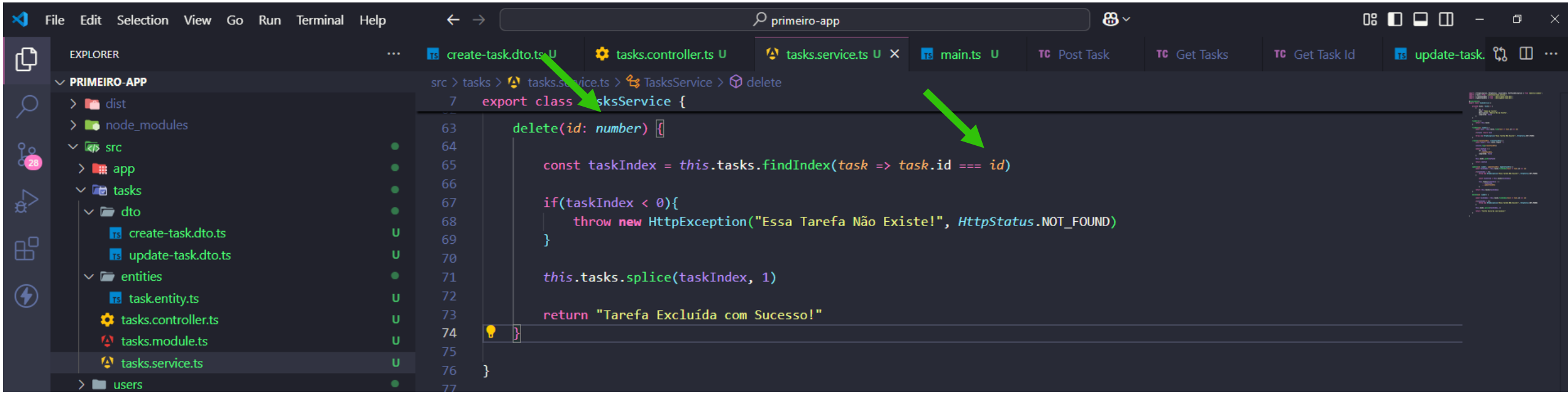
src > tasks > tasks.controller.ts > TasksController

```ts
import { Body, Controller, Delete, Get, Param, ParseIntPipe, Patch, Post } from '@nestjs/common';
import { TasksService } from './tasks.service';
import { CreateTaskDto } from './dto/create-task.dto';
import { UpdateTaskDto } from './dto/update-task.dto';

@Controller('tasks')
export class TasksController {

    constructor(private readonly taskService: TasksService){}

    @Get()
    findAllTasks(){
        return this.taskService.findAll()
    }

    @Get(":id")
    findOneTask(@Param('id', ParseIntPipe) id: number){
        return this.taskService.findOne(id)
    }

    @Post()
    createTask(@Body() createTaskDto: CreateTaskDto){
        return this.taskService.create(createTaskDto)
    }

    @Patch(":id")
    updateTask(@Param("id", ParseIntPipe) id: number, @Body() updateTaskDto: UpdateTaskDto){
        return this.taskService.update(id, updateTaskDto)
    }

    @Delete(":id")
    deleteTask(@Param("id", ParseIntPipe) id:  number){
        return this.taskService.delete(id)
    }

}
```

INSTITUTO FEDERAL
PARANÁ

Transformar Dados - Pipe

Prof. Dr. Robyson Aggio



Transformar Dados - Pipe

Prof. Dr. Robyson Aggio

INSTITUTO FEDERAL PARANÁ

docs.nestjs.com/techniques/validation

Search COURSES ENTERPRISE NEW DEVTOOLS DEPLOY WITH MAU

Validation

It is best practice to validate the correctness of any data sent into a web application. To automatically validate incoming requests, Nest provides several pipes available right out-of-the-box:

- `ValidationPipe`
- `ParseIntPipe`
- `ParseBoolPipe`
- `ParseArrayPipe`
- `ParseUUIDPipe`

The `ValidationPipe` makes use of the powerful **class-validator** package and its declarative validation decorators. The `ValidationPipe` provides a convenient approach to enforce validation rules for all incoming client payloads, where the specific rules are declared with simple annotations in local class/DTO declarations in each module.

Overview

In the **Pipes** chapter, we went through the process of building simple pipes and binding them to controllers, methods or to the global app to demonstrate how the process works. Be sure to review that chapter to best understand the topics of this chapter. Here, we'll focus on various **real world** use cases of the `ValidationPipe`, and show how to use some of its advanced customization features.

Using the built-in ValidationPipe

To begin using it, we first install the required dependency.

```
$ npm i --save class-validator class-transformer
```

HINT
The `ValidationPipe` is exported from the `@nestjs/common` package.

Prof. Dr. Robyson Aggio

INSTITUTO FEDERAL
PARANÁ

Atividade

1) Dentro de cada camada: Guests, Users e Teachers, implemente o ValidationPipe

2) Dentro de cada camada no update*Dto utilize o PartialType

3) Em cada camada utilize ParseIntPipe, nos verbos: Get(), Patch(), Delete(), onde passa o parâmetro id

4) Em nosso desenvolvimento em sala, iremos criar uma camada de usuário e faremos a associação com as tarefas (1 (usuário) → n (tarefas).

Crie um projeto do zero com 2 camadas (em breve serão associadas: 1 → n).
Implemente todos os códigos até chegar nesse slide.

Obs: Na próxima aula, iniciaremos o estudo do Prisma ORM.

O Prisma ORM é uma ferramenta de código aberto que permite definir modelos e interagir com bancos de dados.