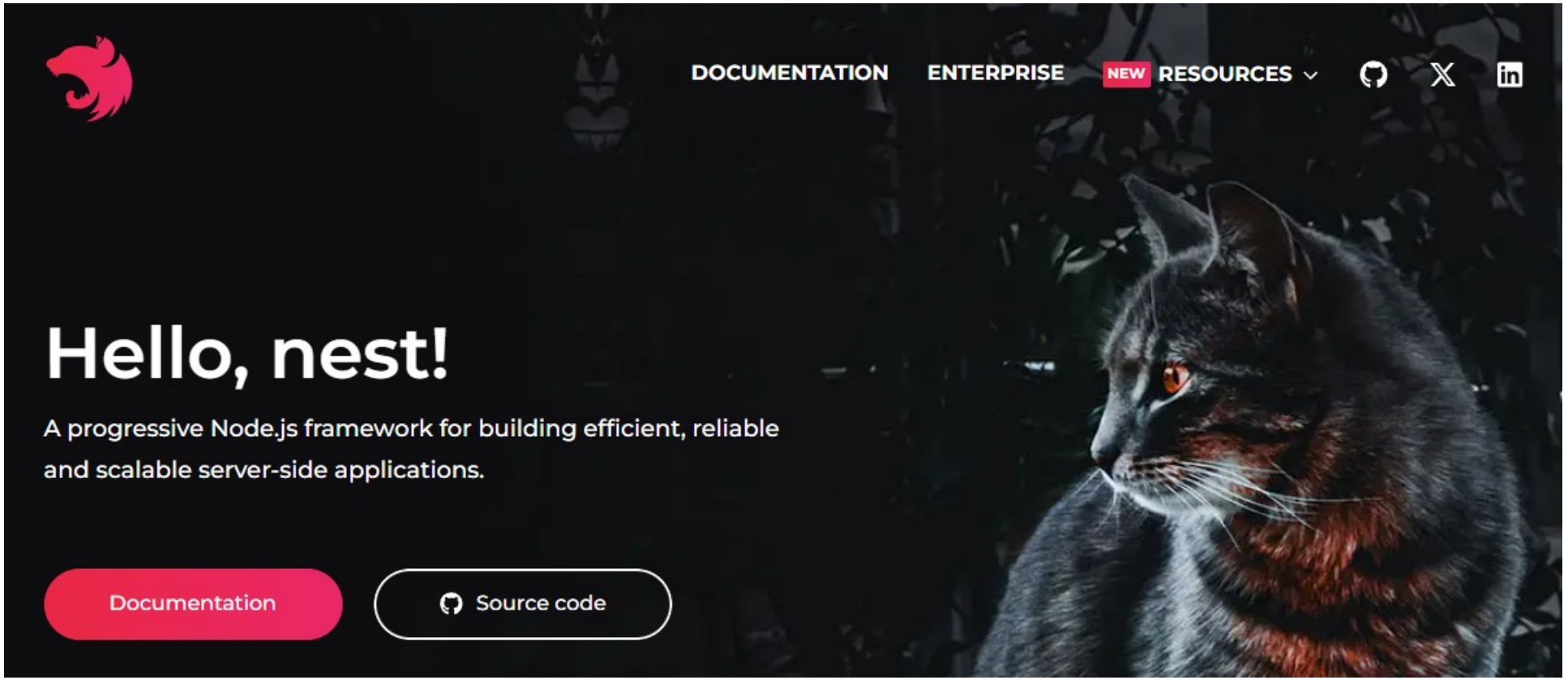
INSTITUTO FEDERAL
PARANÁ

Nest.js

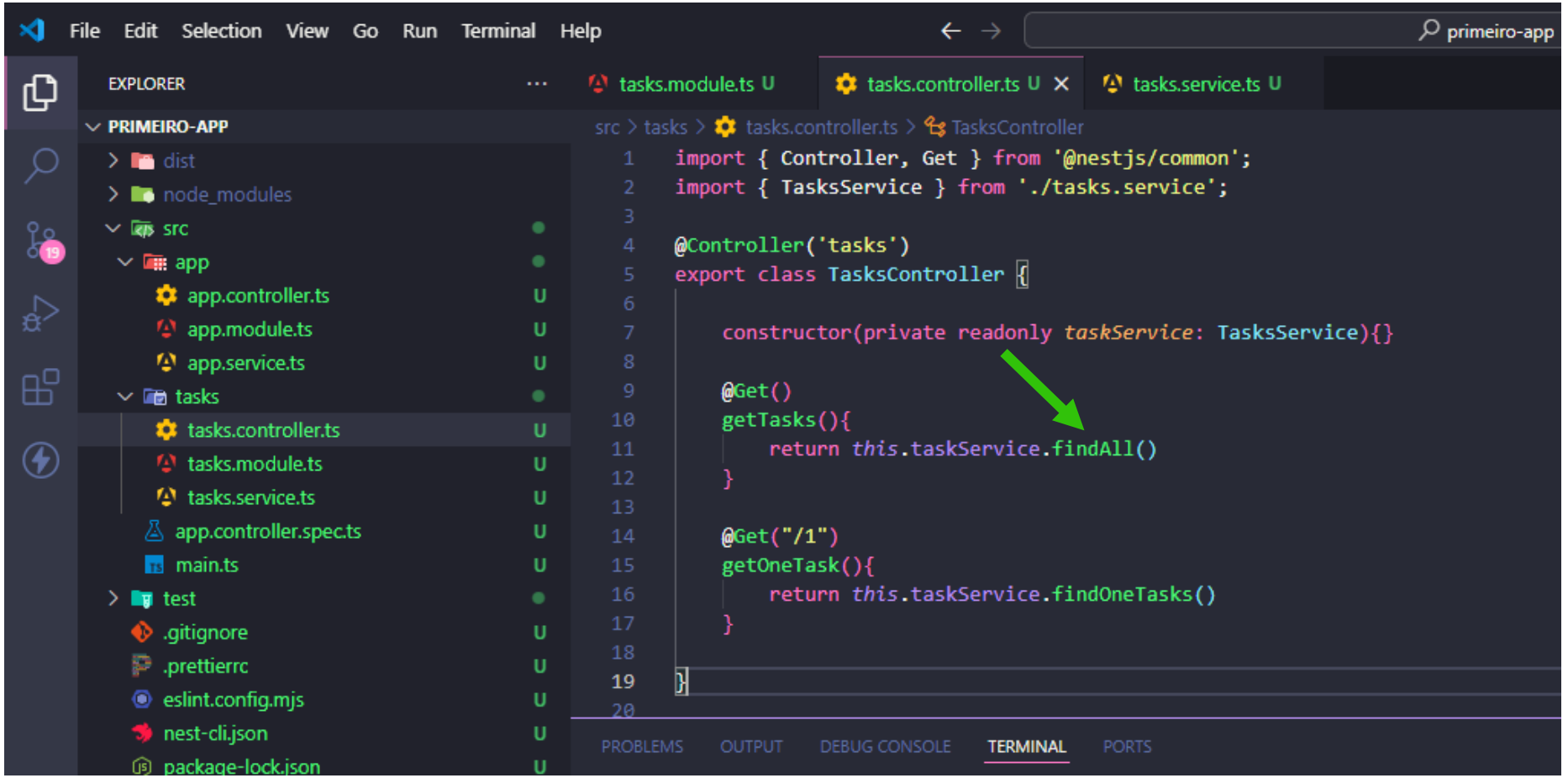


Prof. Dr. José Robyson Aggio Molinari

www.aggioirati.com.br

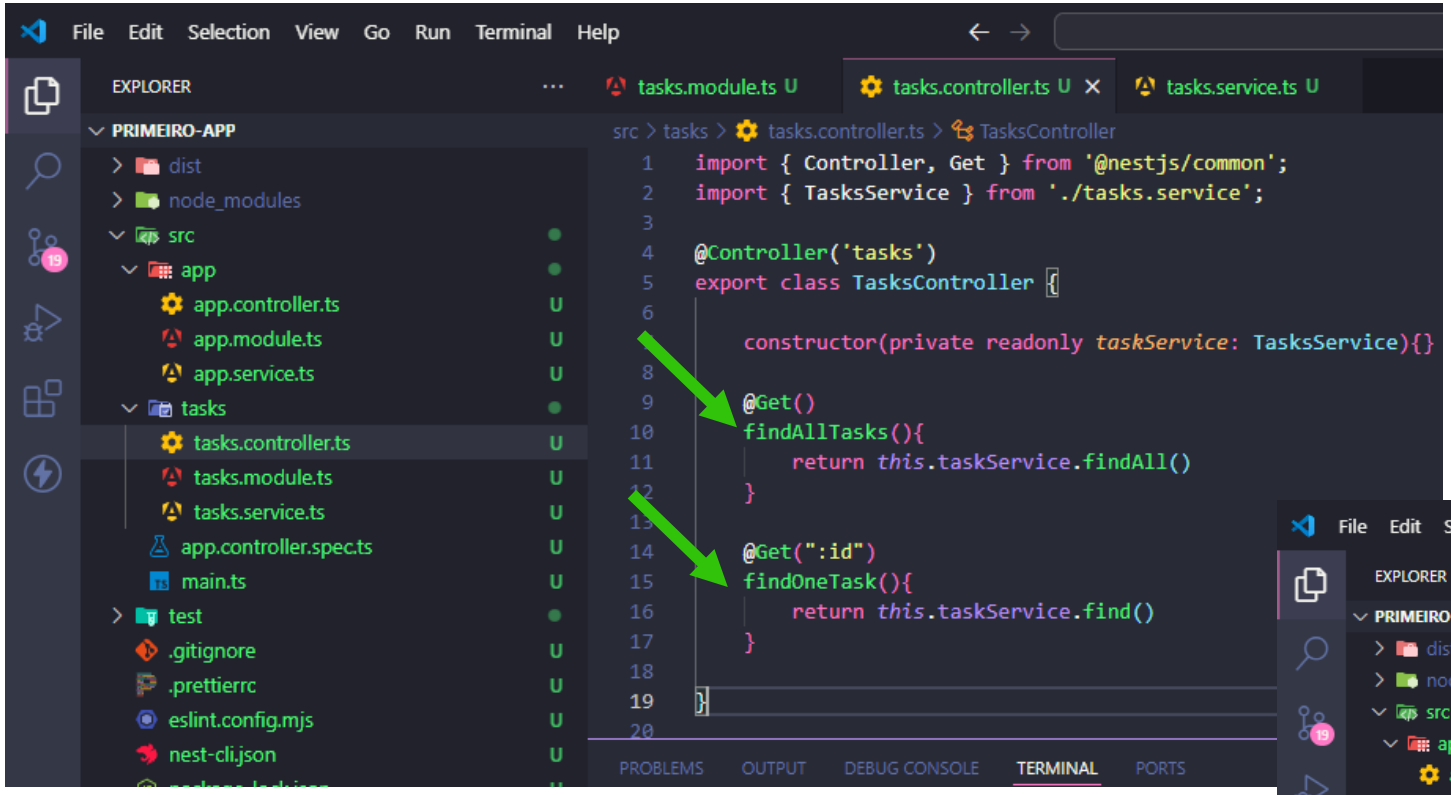
Prof. Dr. Robyson Aggio

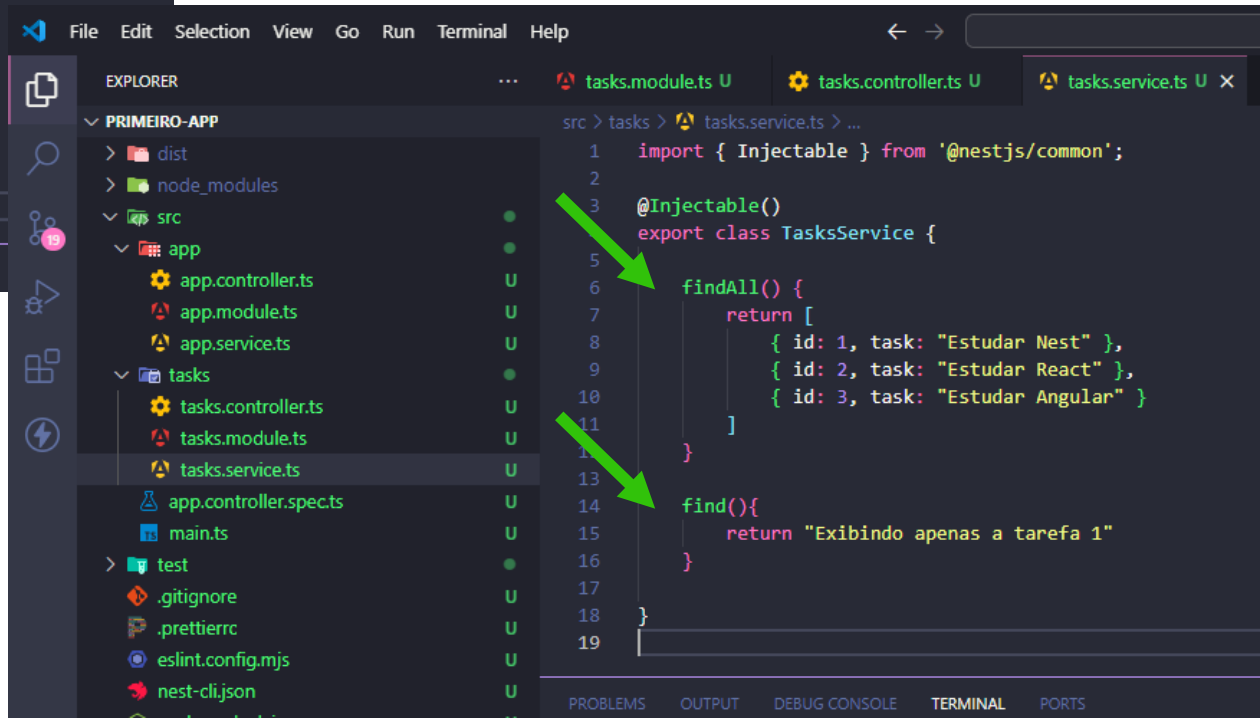
Rotas com parâmetros



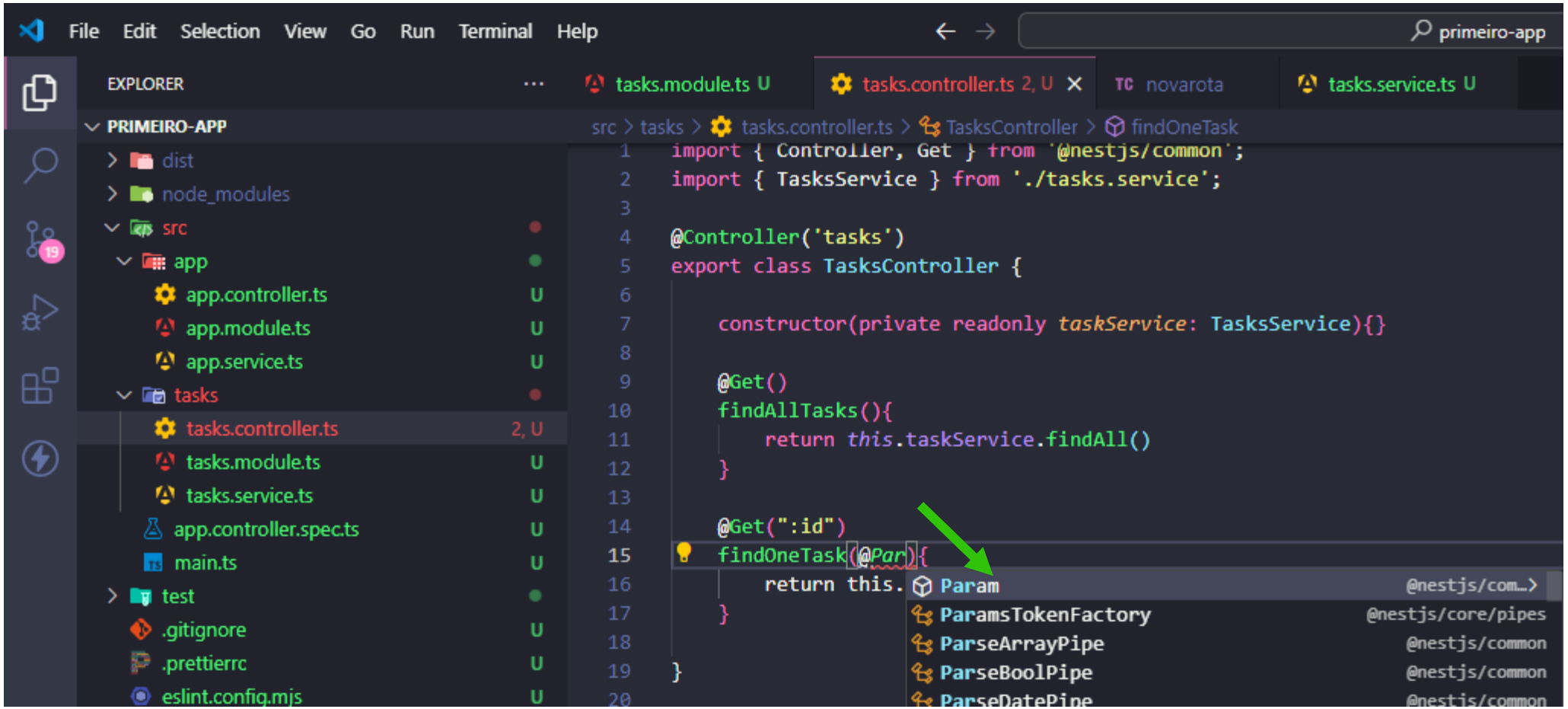
Prof. Dr. Robyson Aggio

Rotas com parâmetros



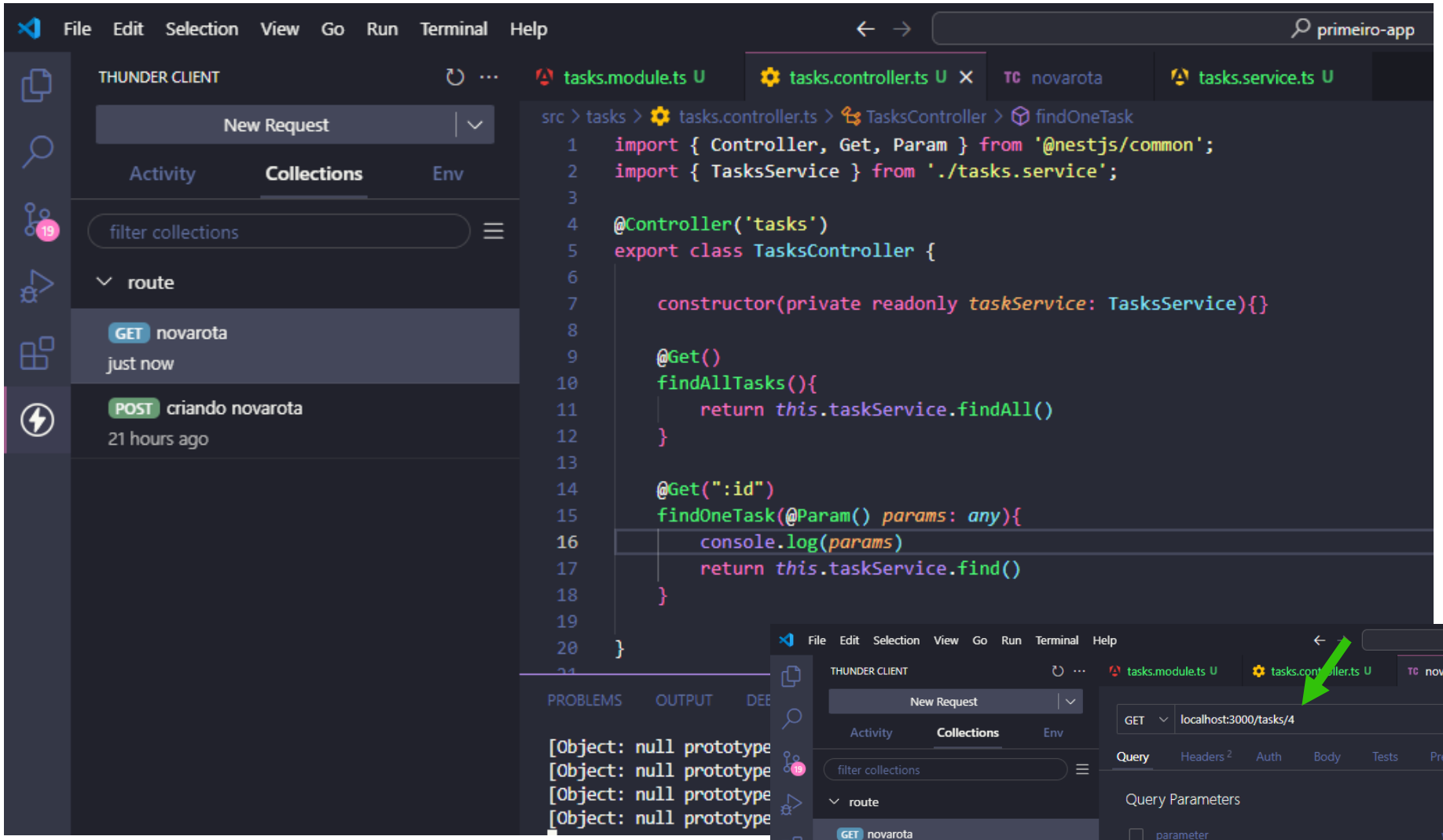


Rotas com parâmetros



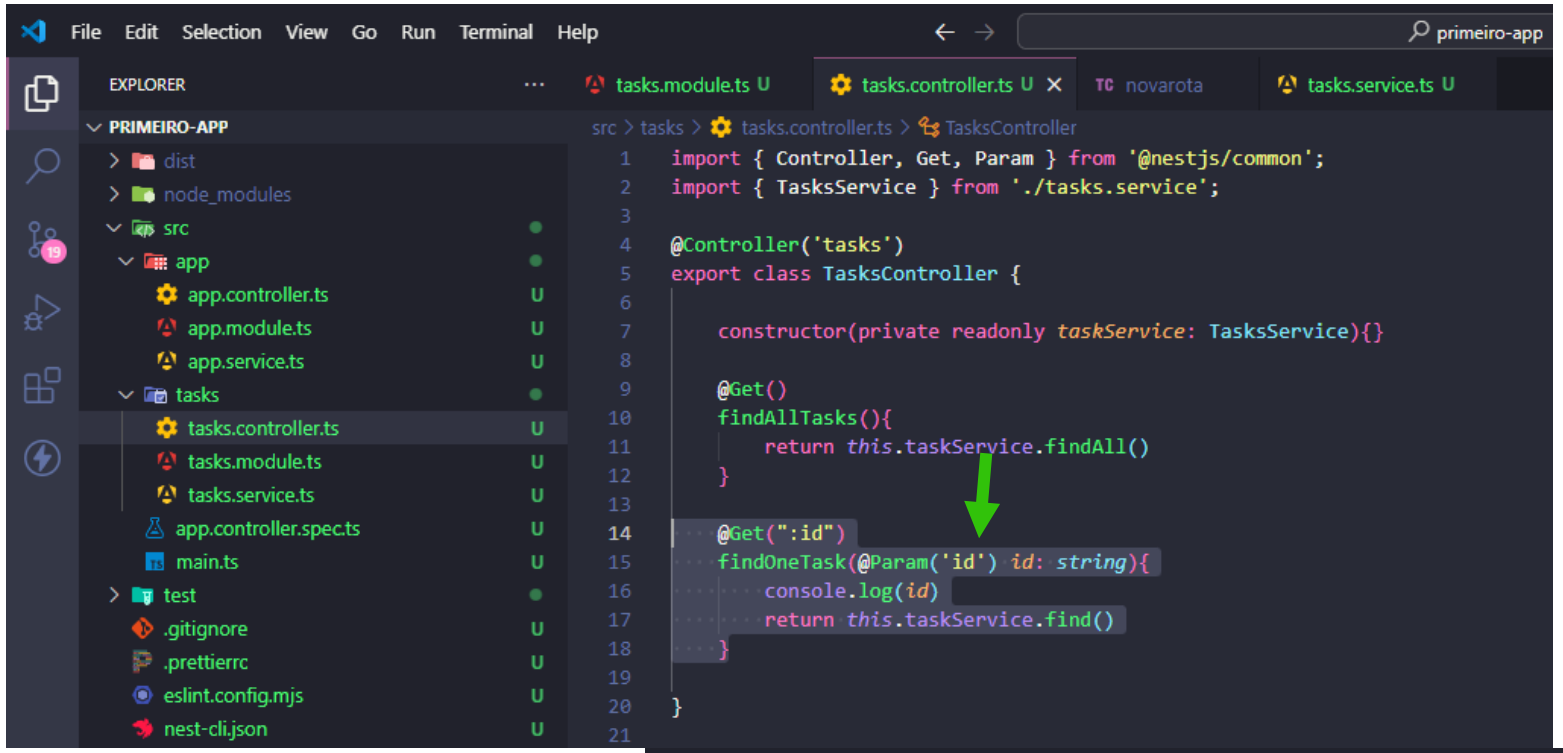
Prof. Dr. Robyson Aggio

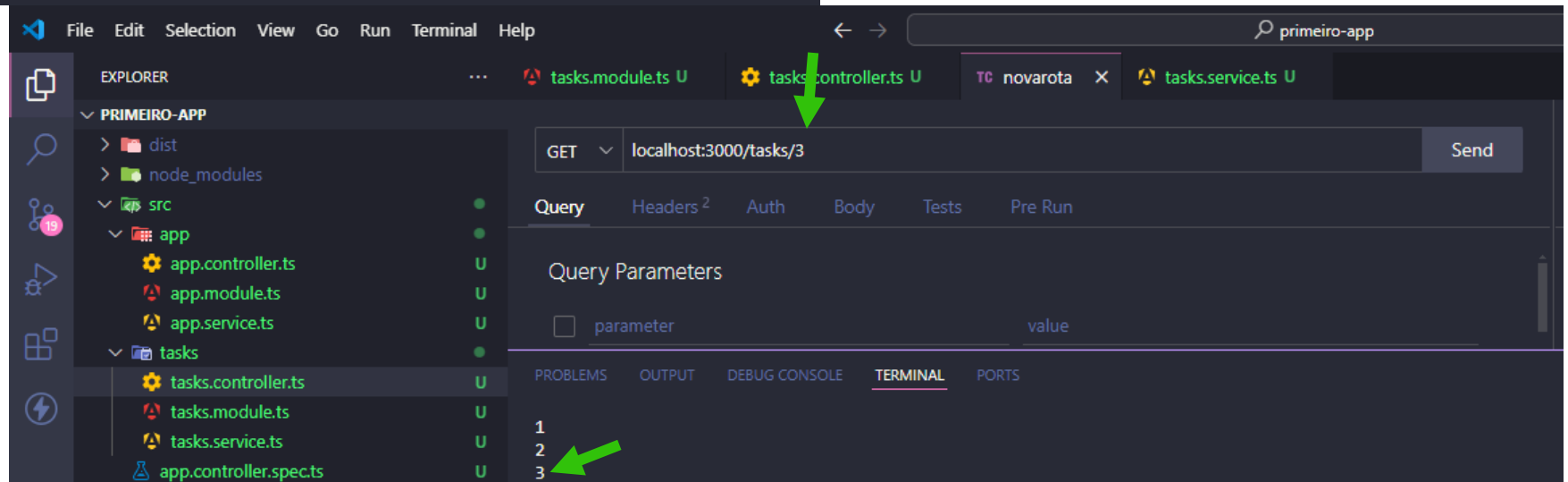
Rotas com parâmetros



Prof. Dr. Robyson Aggio

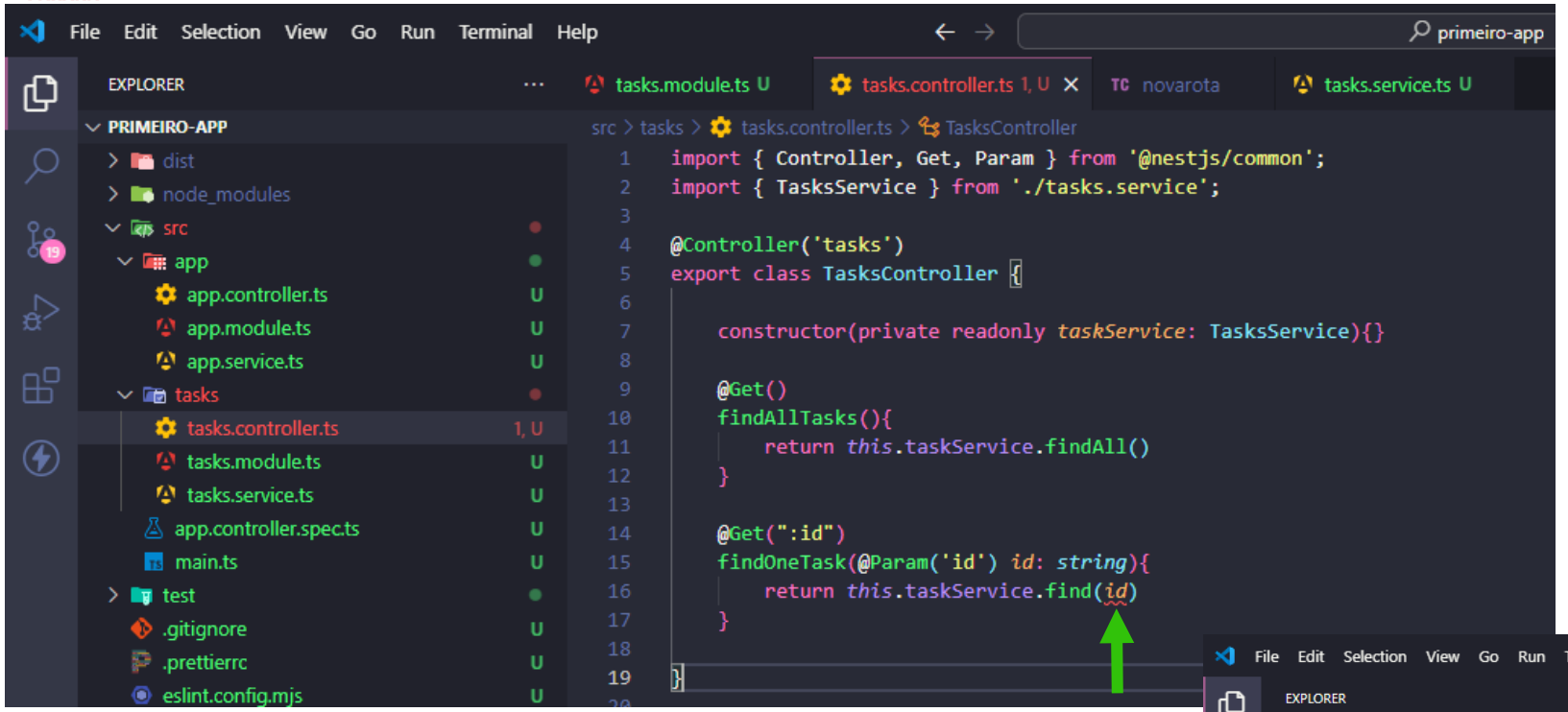
Rotas com parâmetros

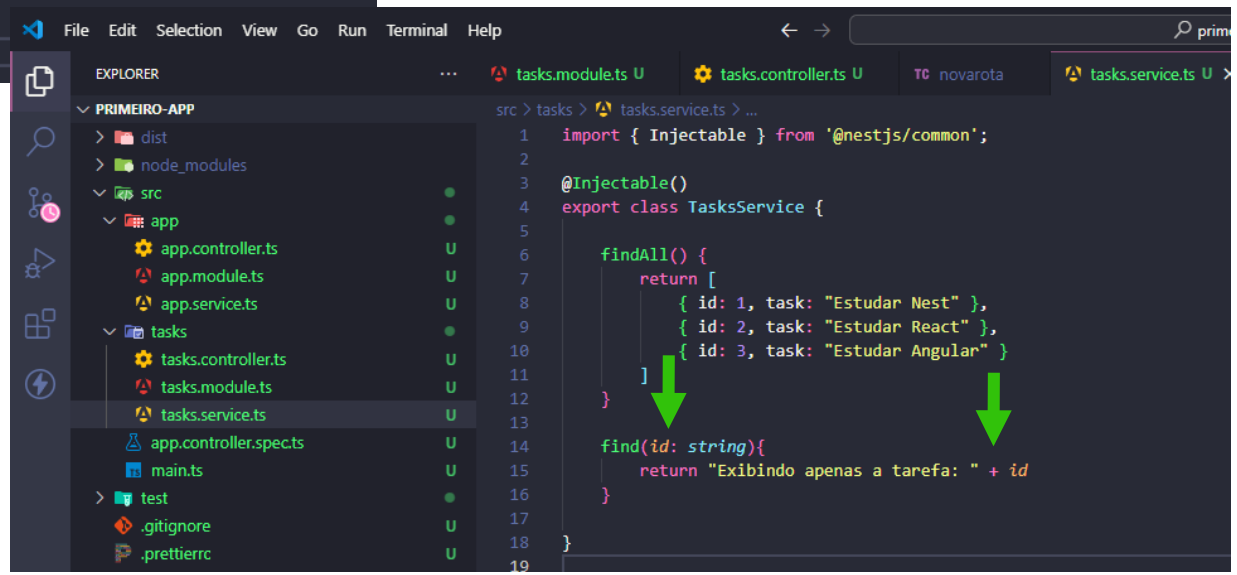




Prof. Dr. Robyson Aggio

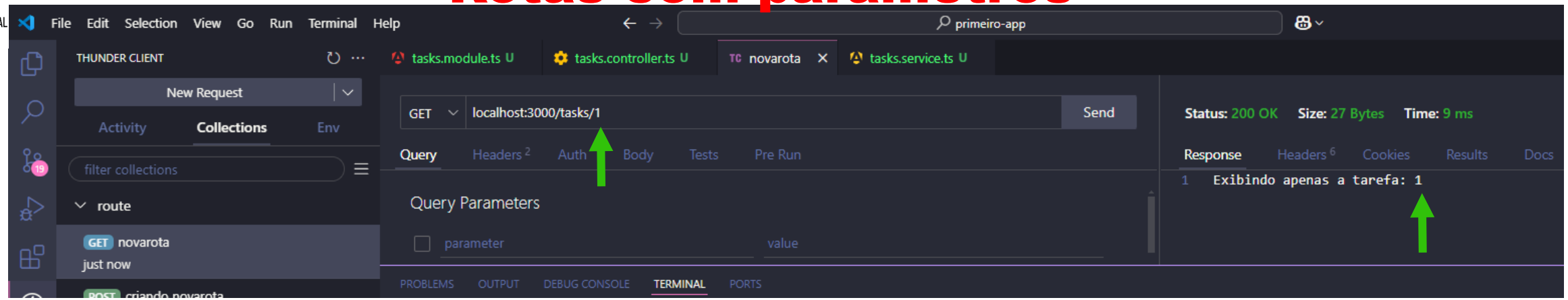
Rotas com parâmetros

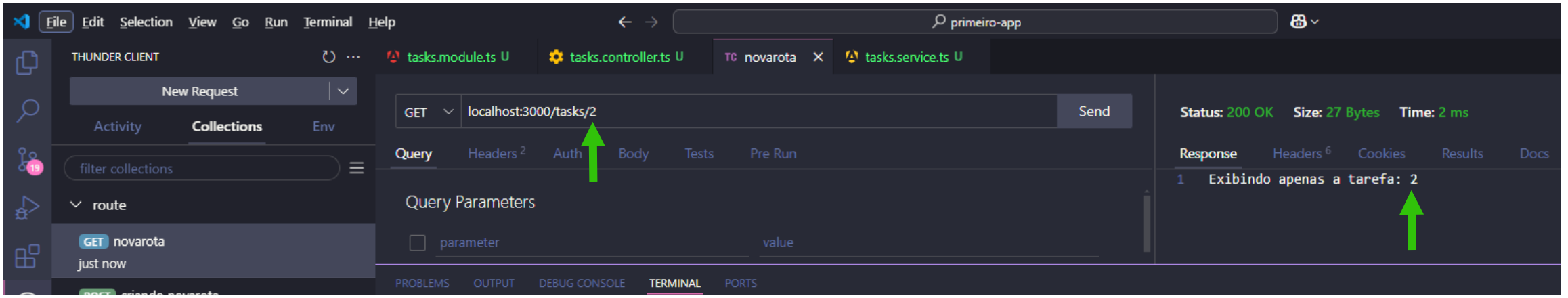


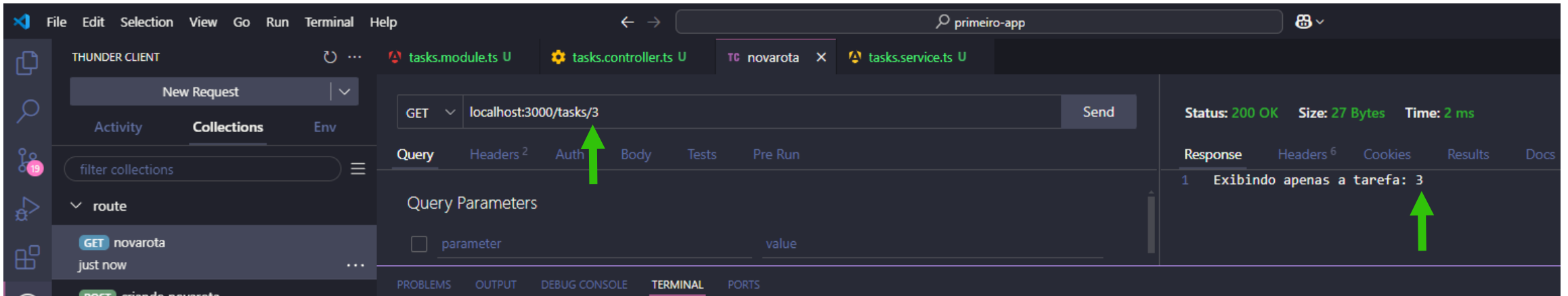


Prof. Dr. Robyson Aggio

Rotas com parâmetros

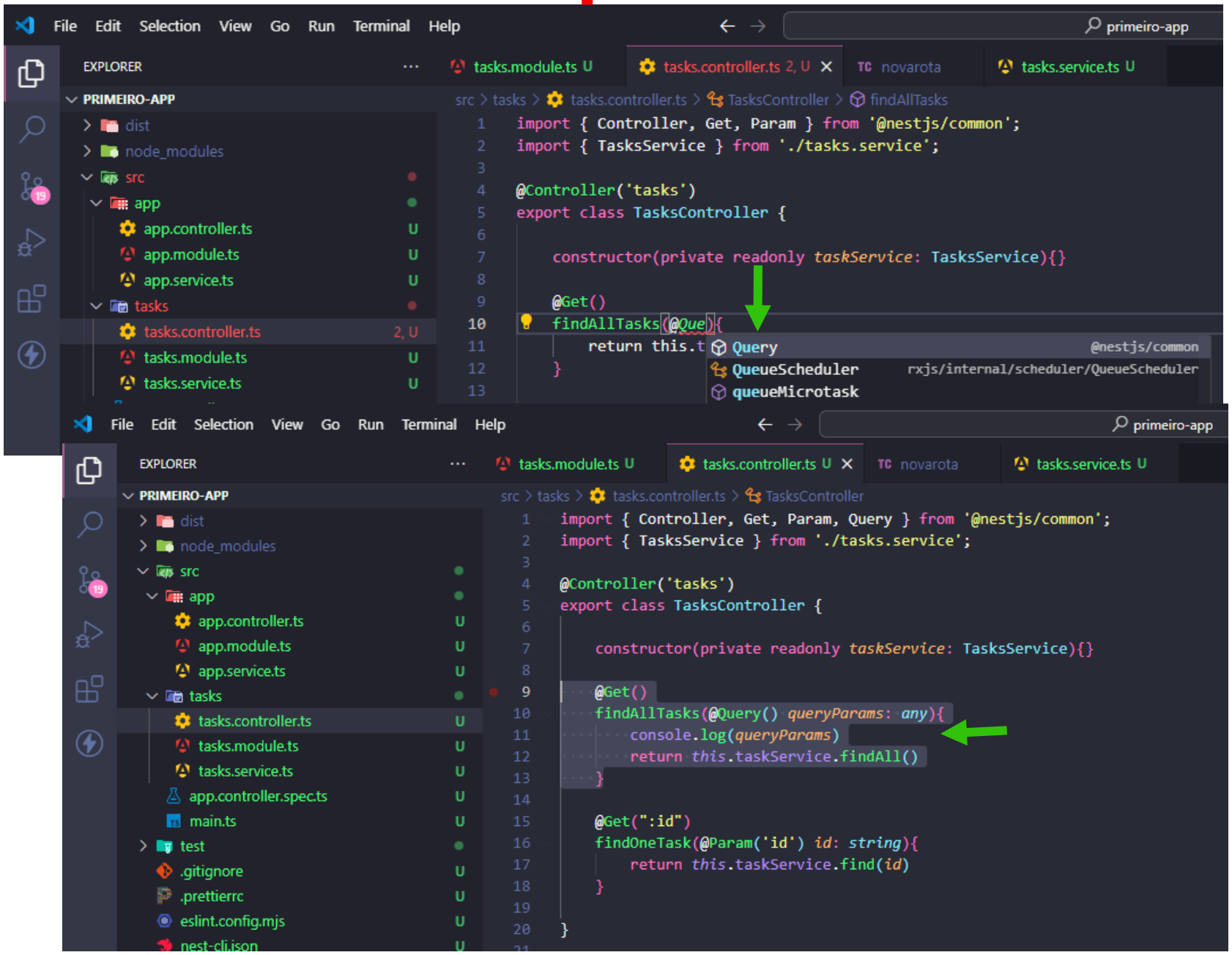






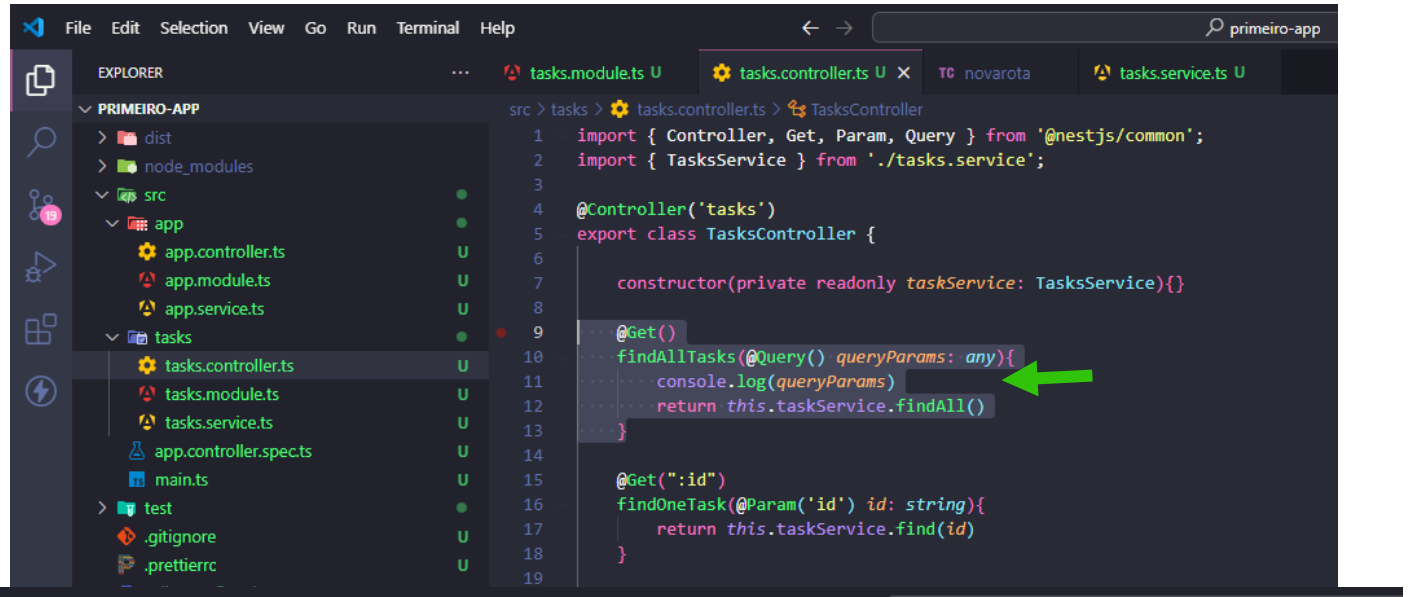
Prof. Dr. Robyson Aggio

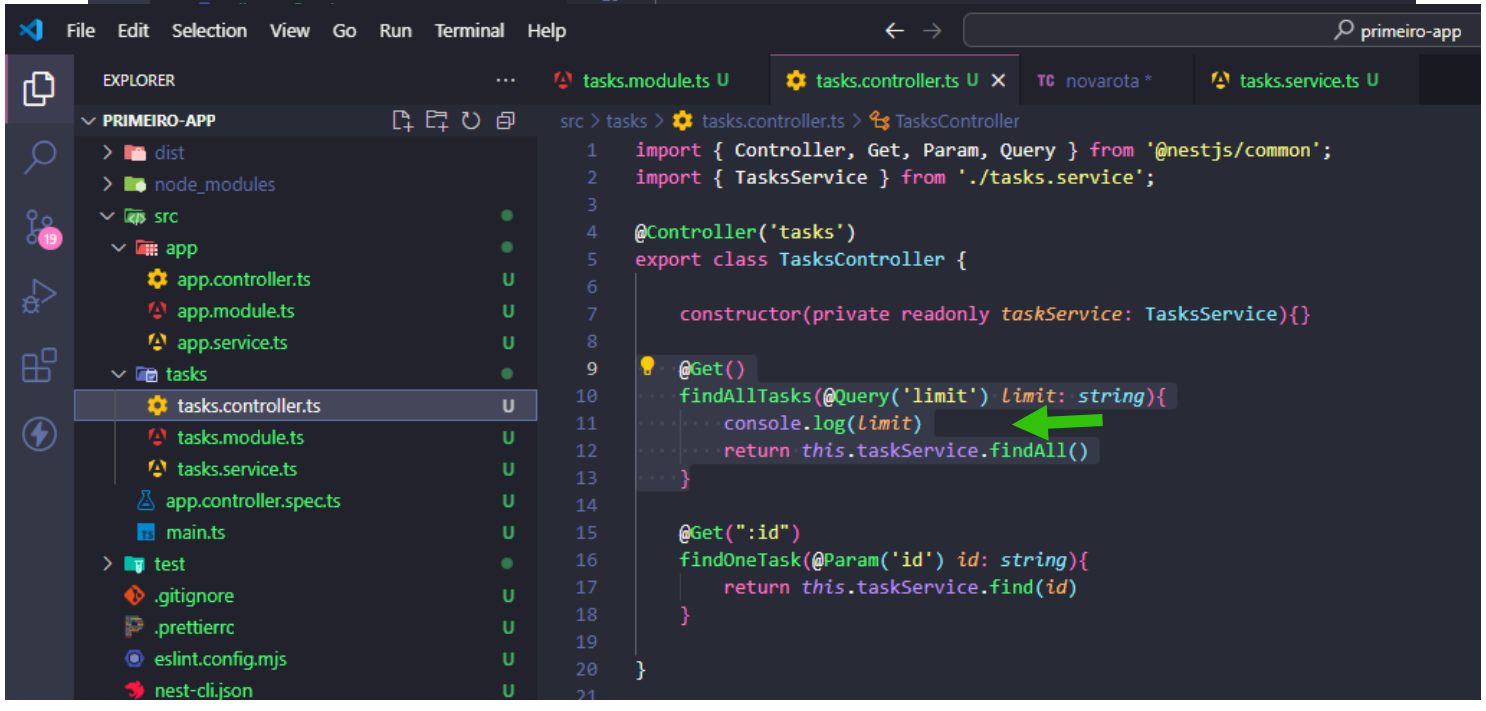
Rotas com parâmetros



Prof. Dr. Robyson Aggio

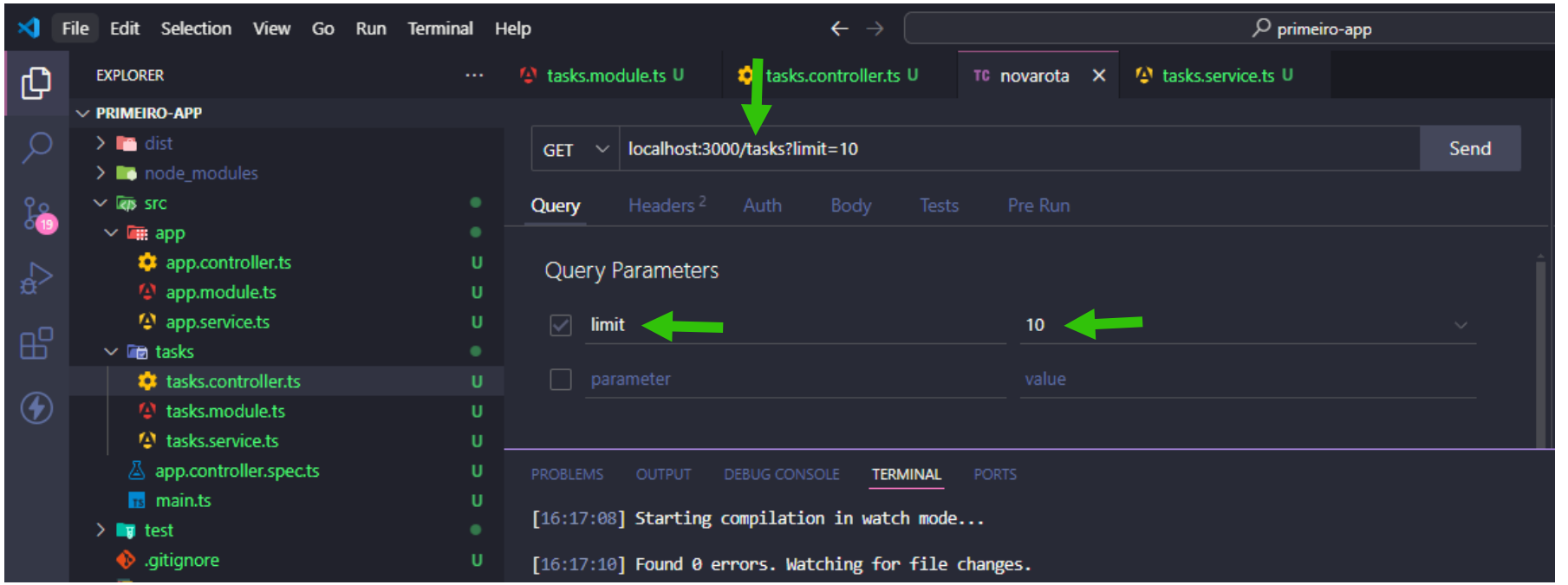
Rotas com parâmetros

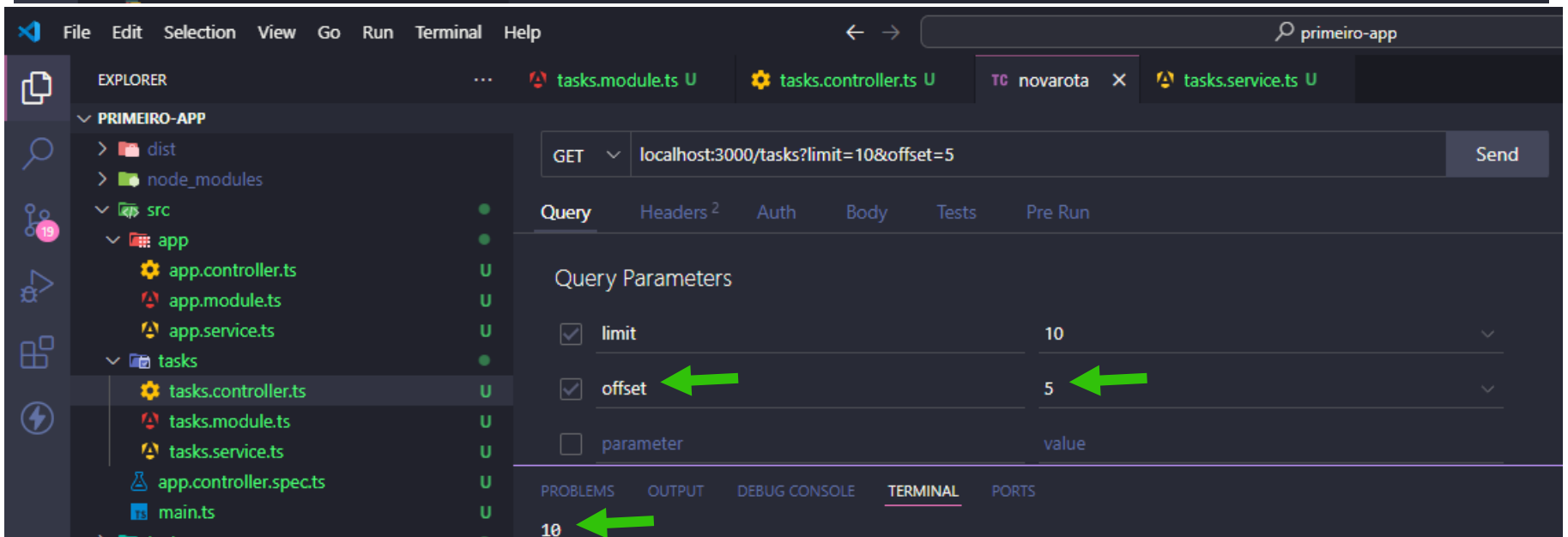




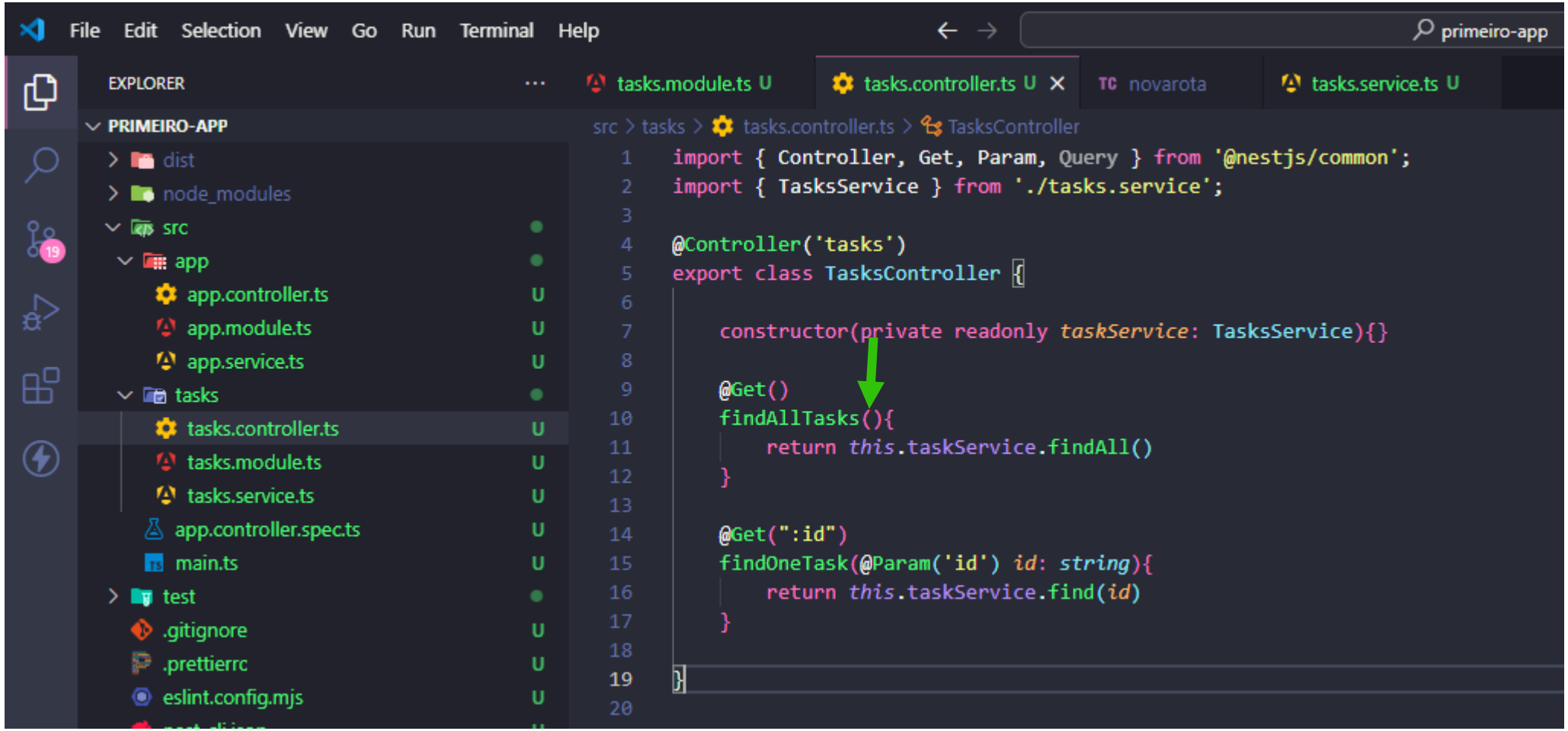
Prof. Dr. Robyson Aggio

Rotas com parâmetros

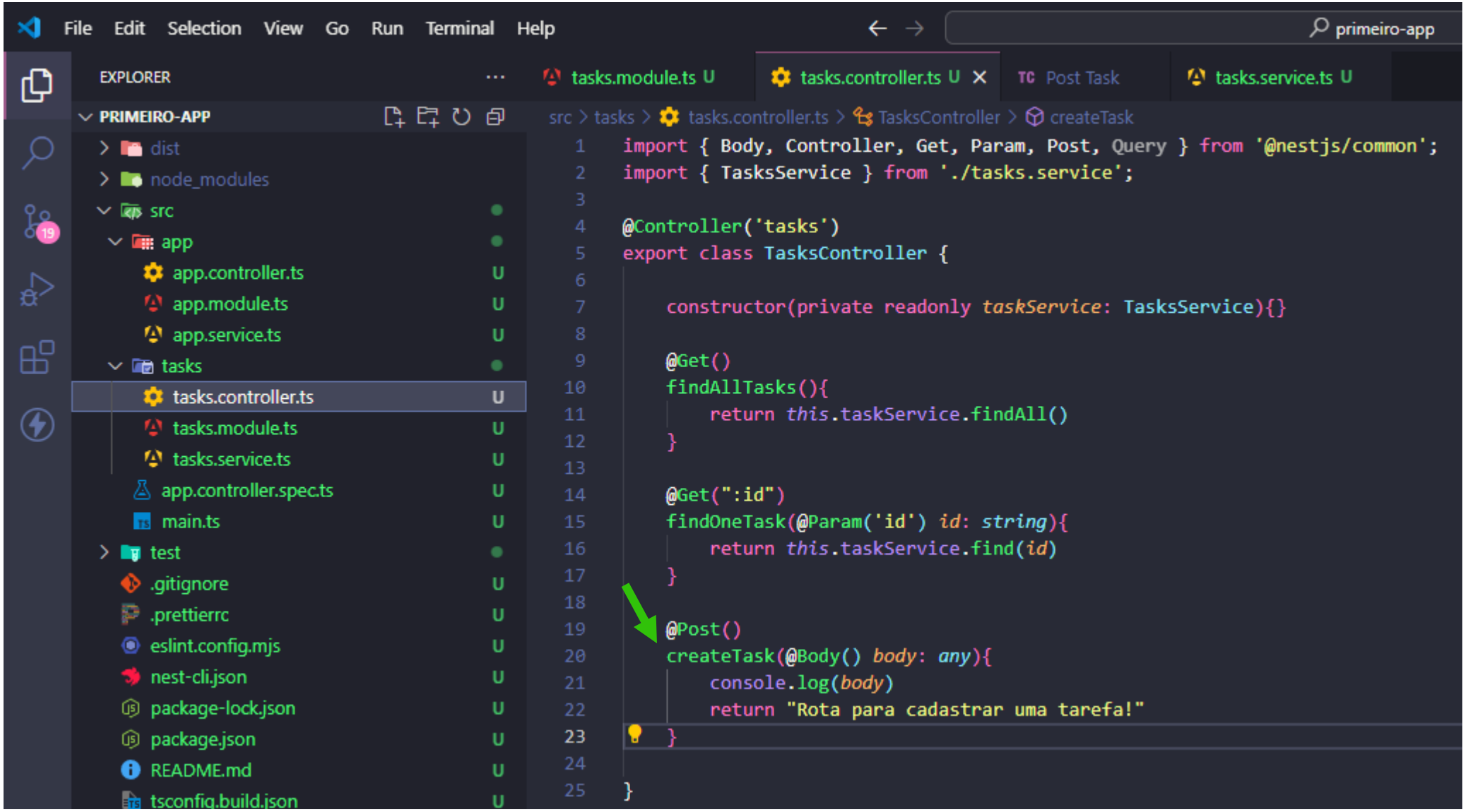




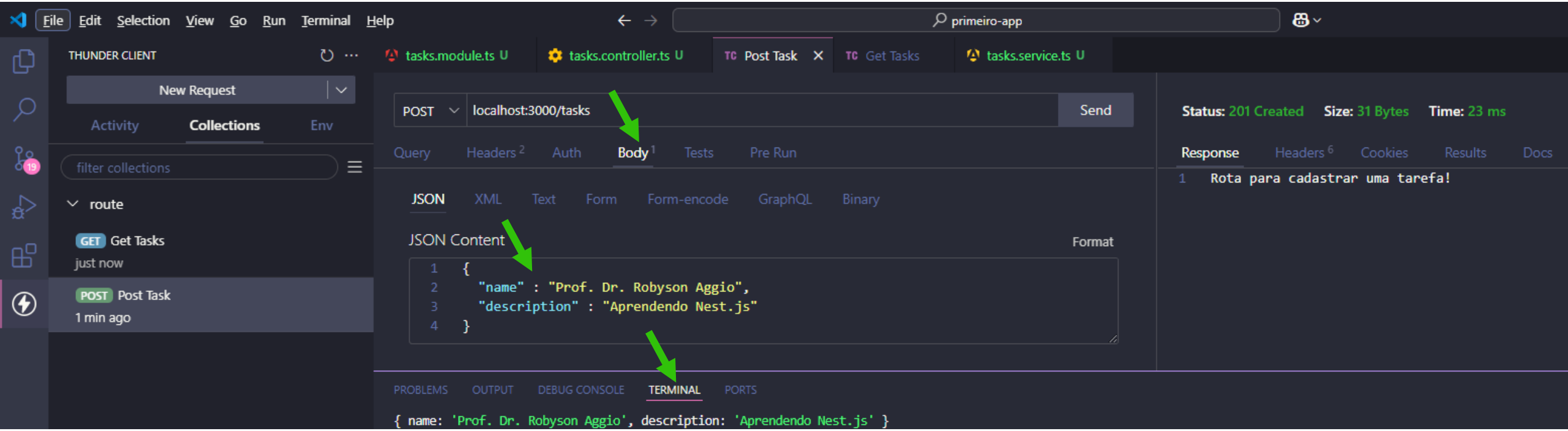
Rotas com parâmetros



Rotas com Post

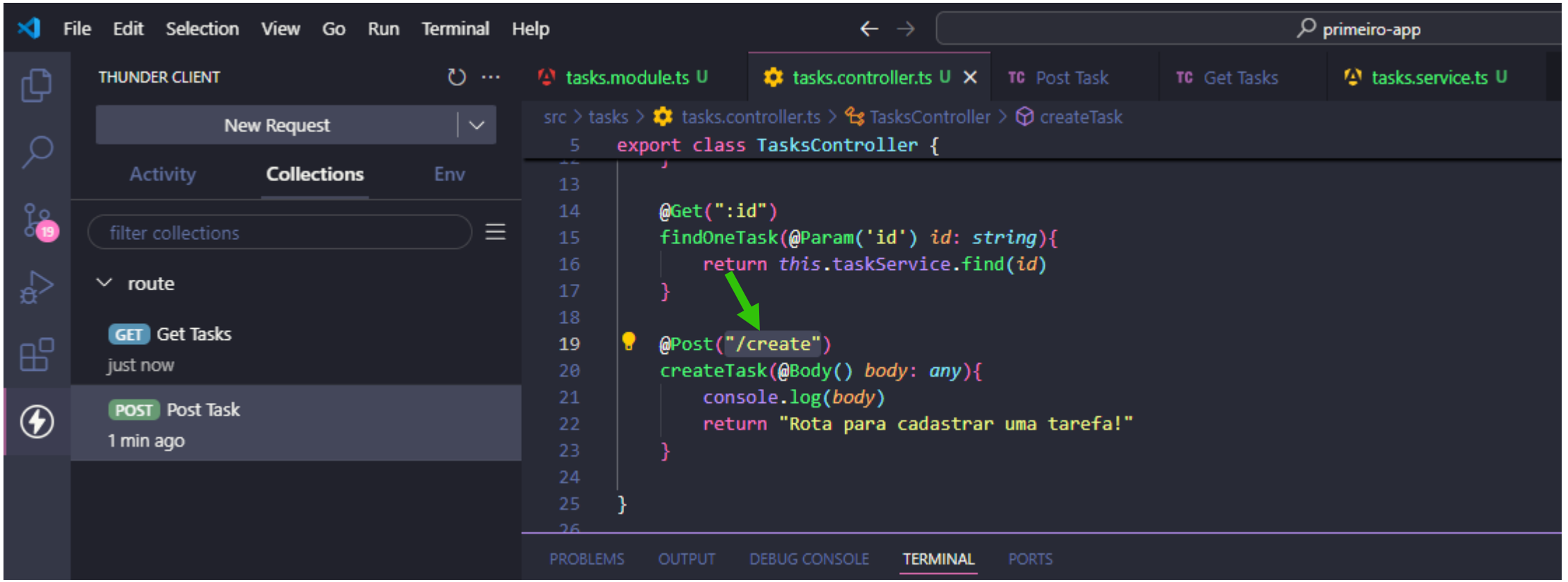


Rotas com Post

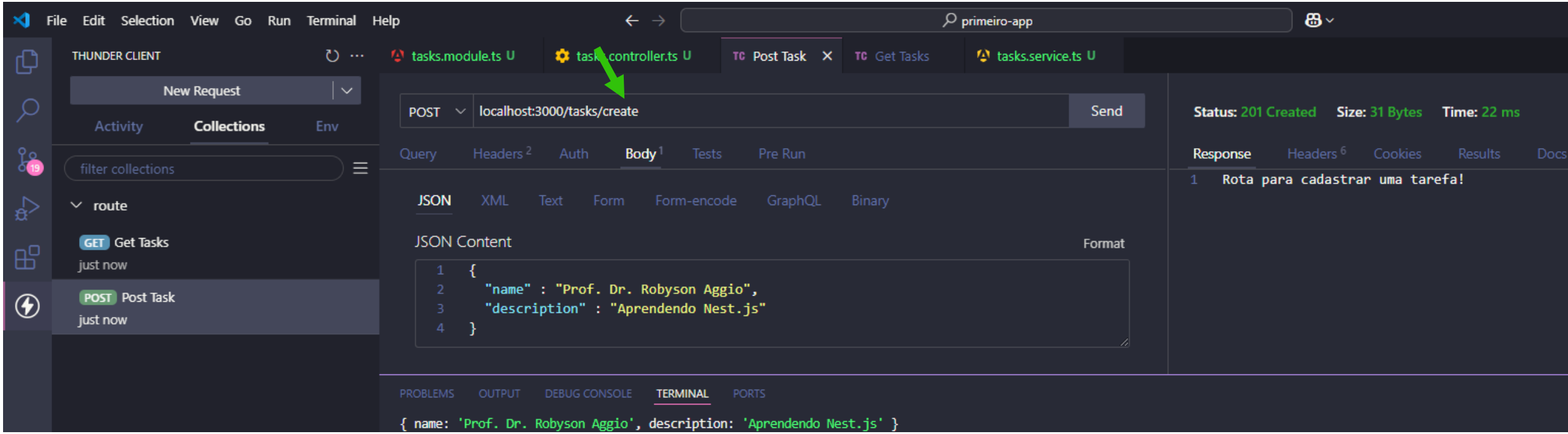


Prof. Dr. Robyson Aggio

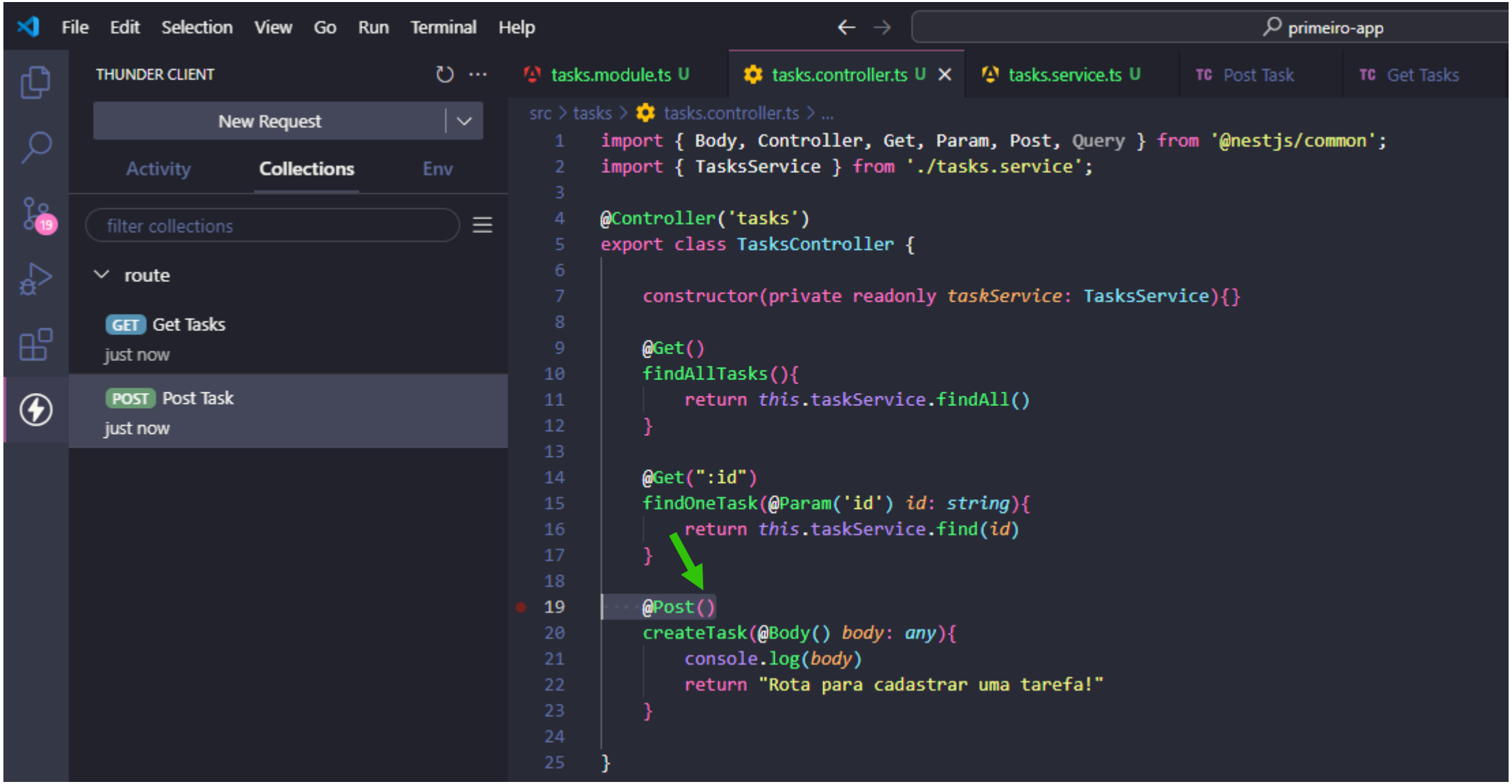
Rotas com Post



Rotas com Post

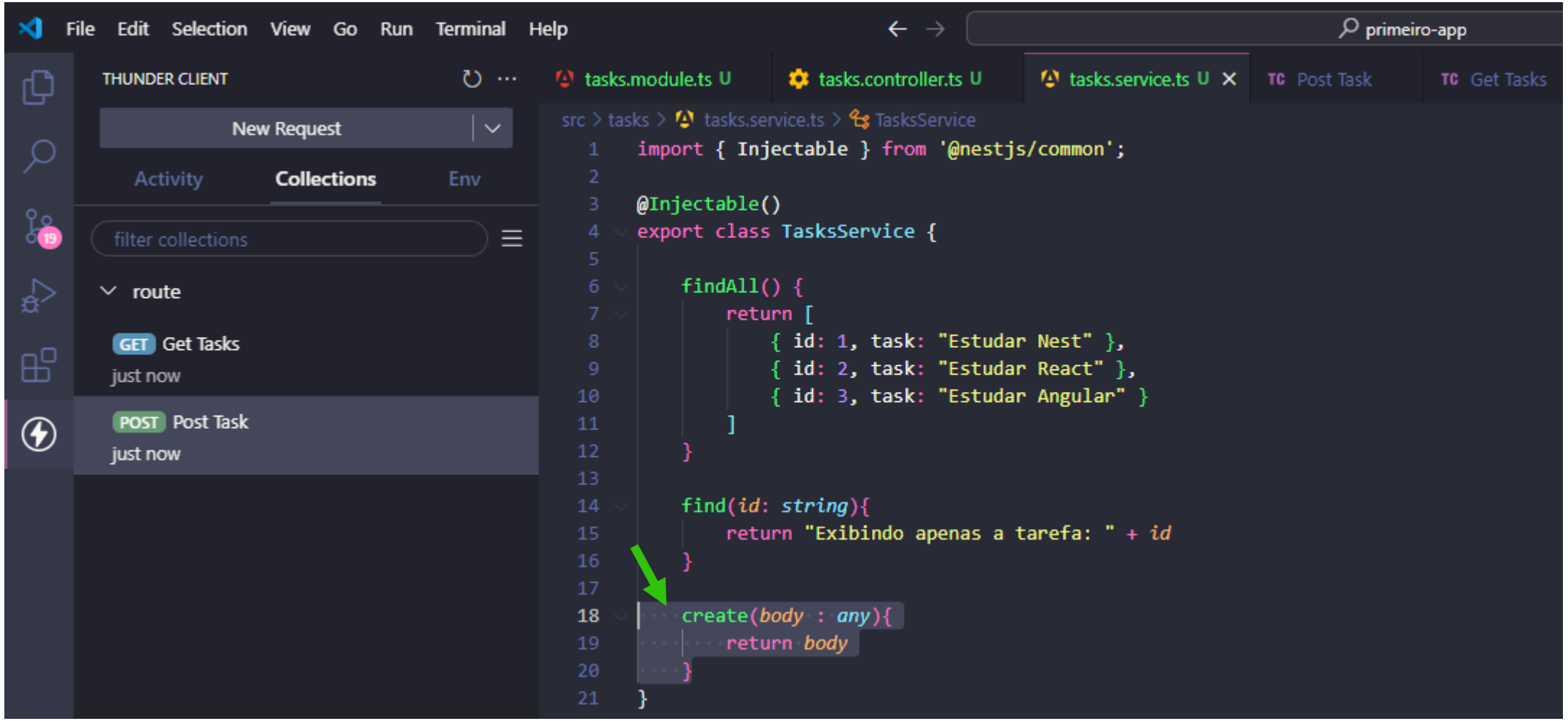


Rotas com Post

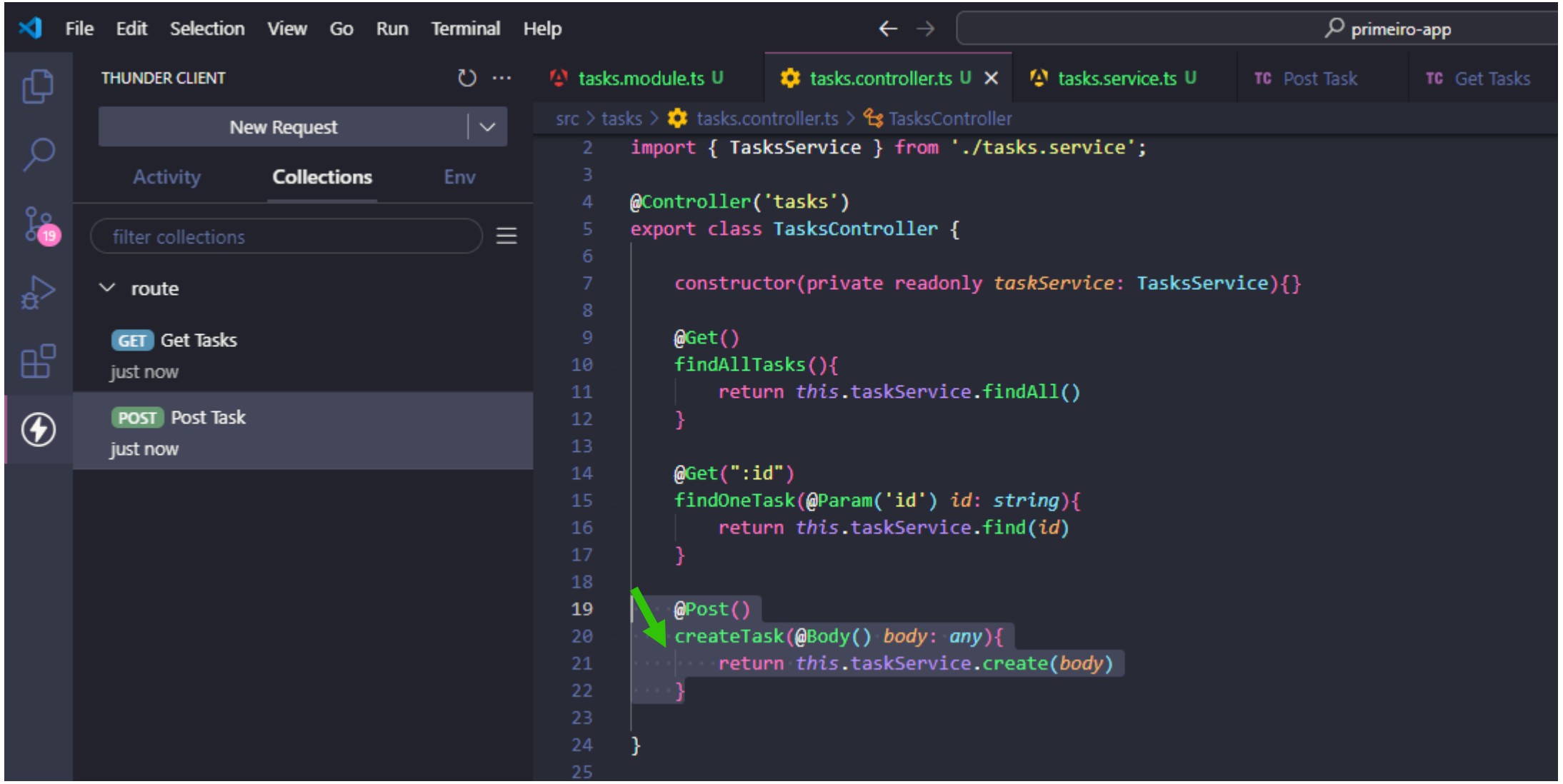


Prof. Dr. Robyson Aggio

Rotas com Post

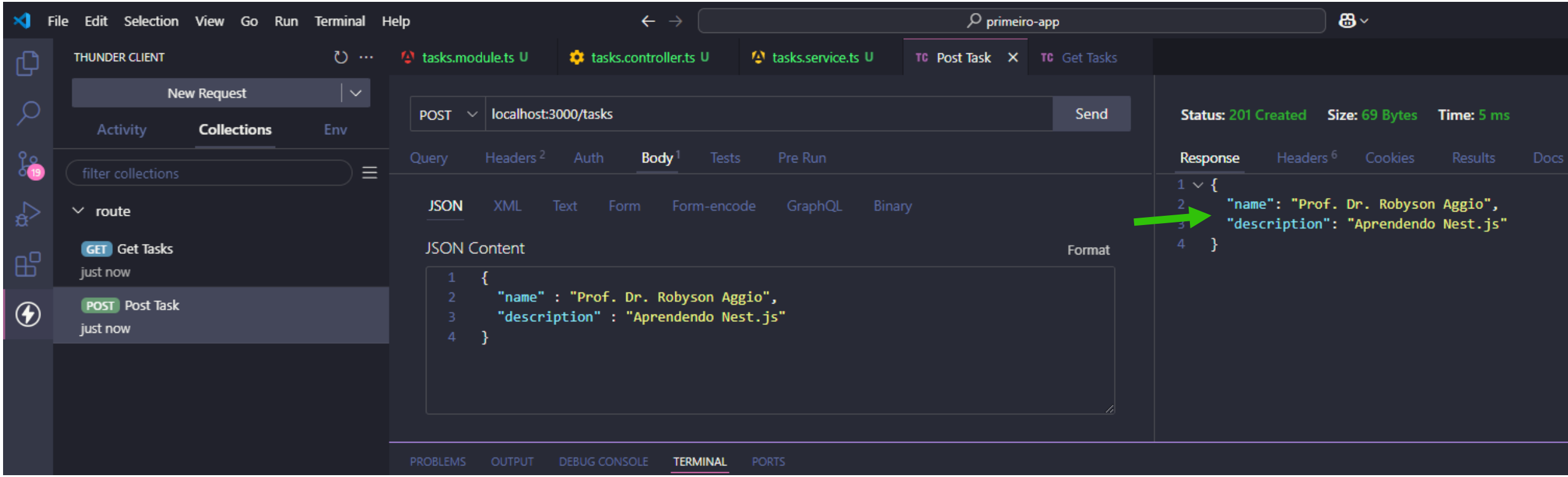


Rotas com Post



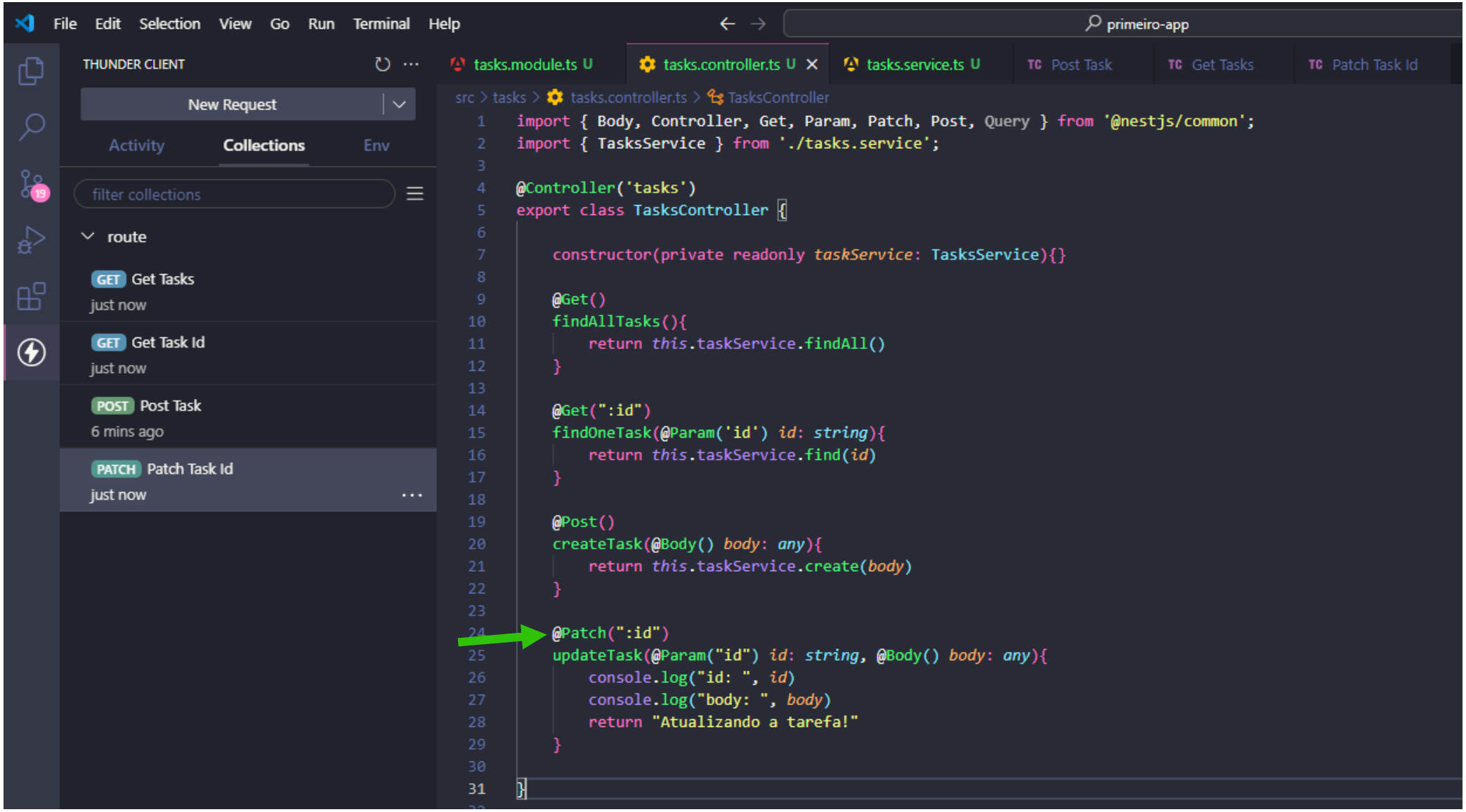
Prof. Dr. Robyson Aggio

Rotas com Post



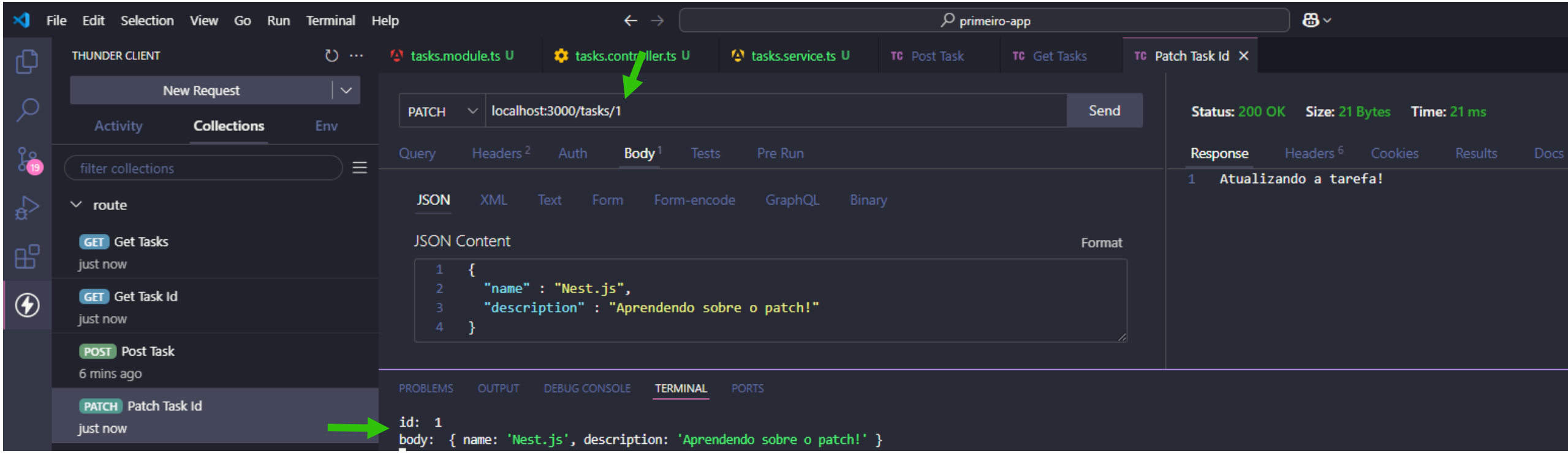
Prof. Dr. Robyson Aggio

Rotas com Patch



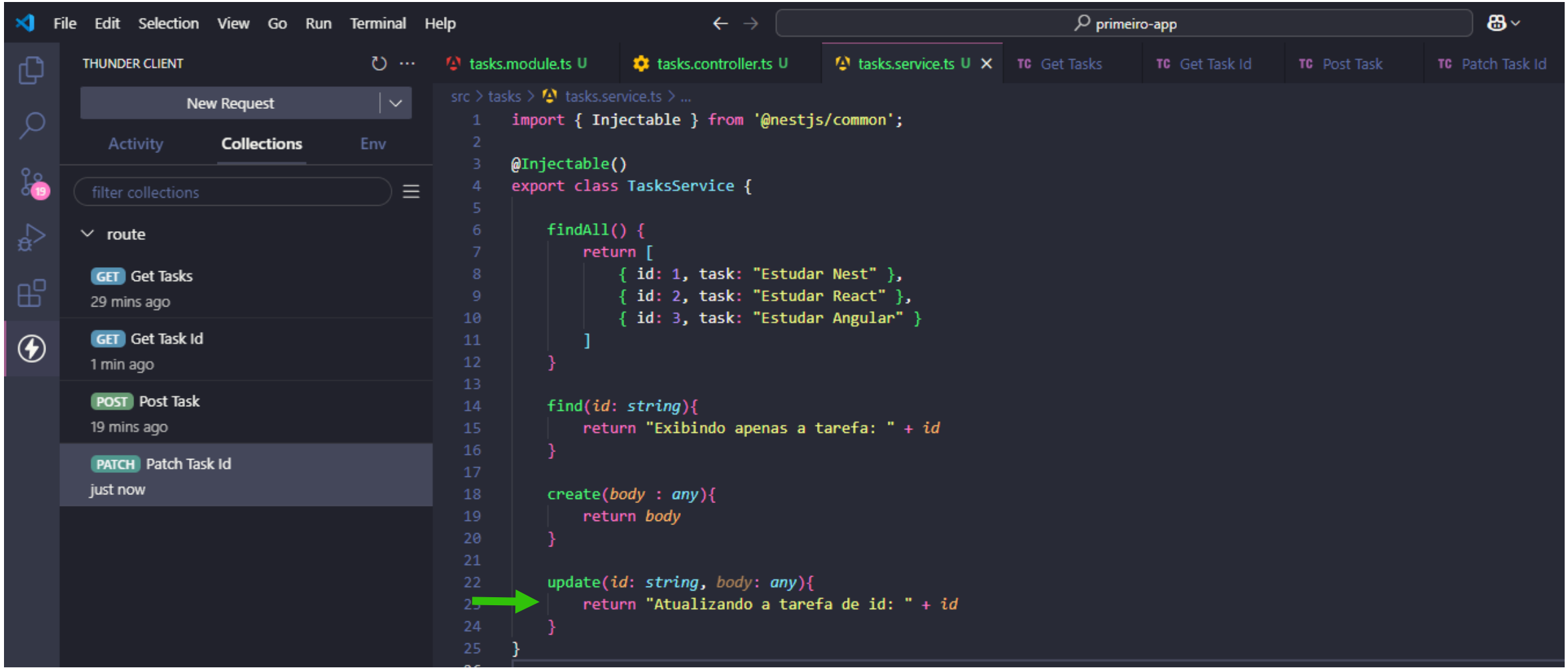
Prof. Dr. Robyson Aggio

Rotas com Patch



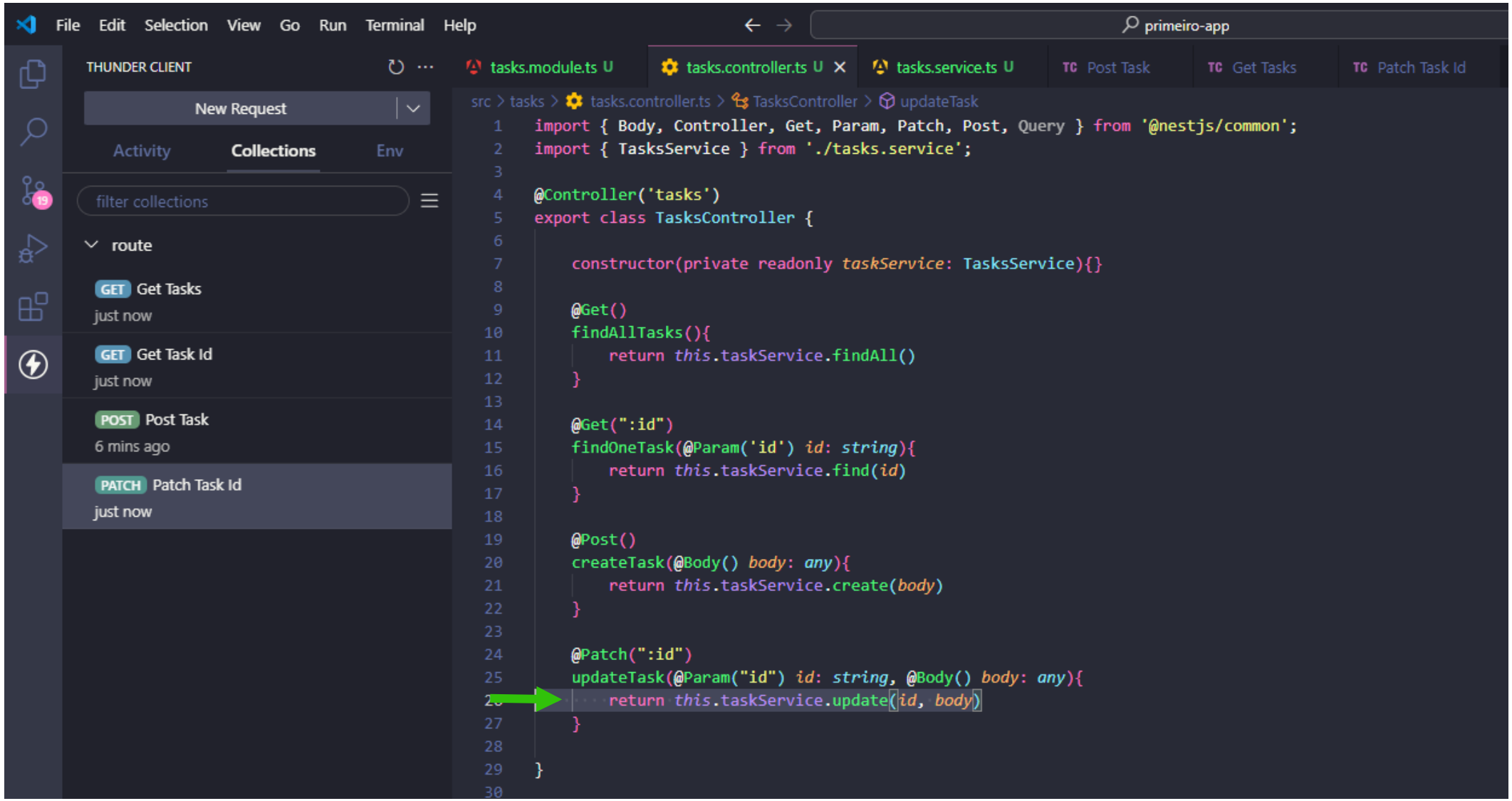
Prof. Dr. Robyson Aggio

Rotas com Patch



Prof. Dr. Robyson Aggio

Rotas com Patch

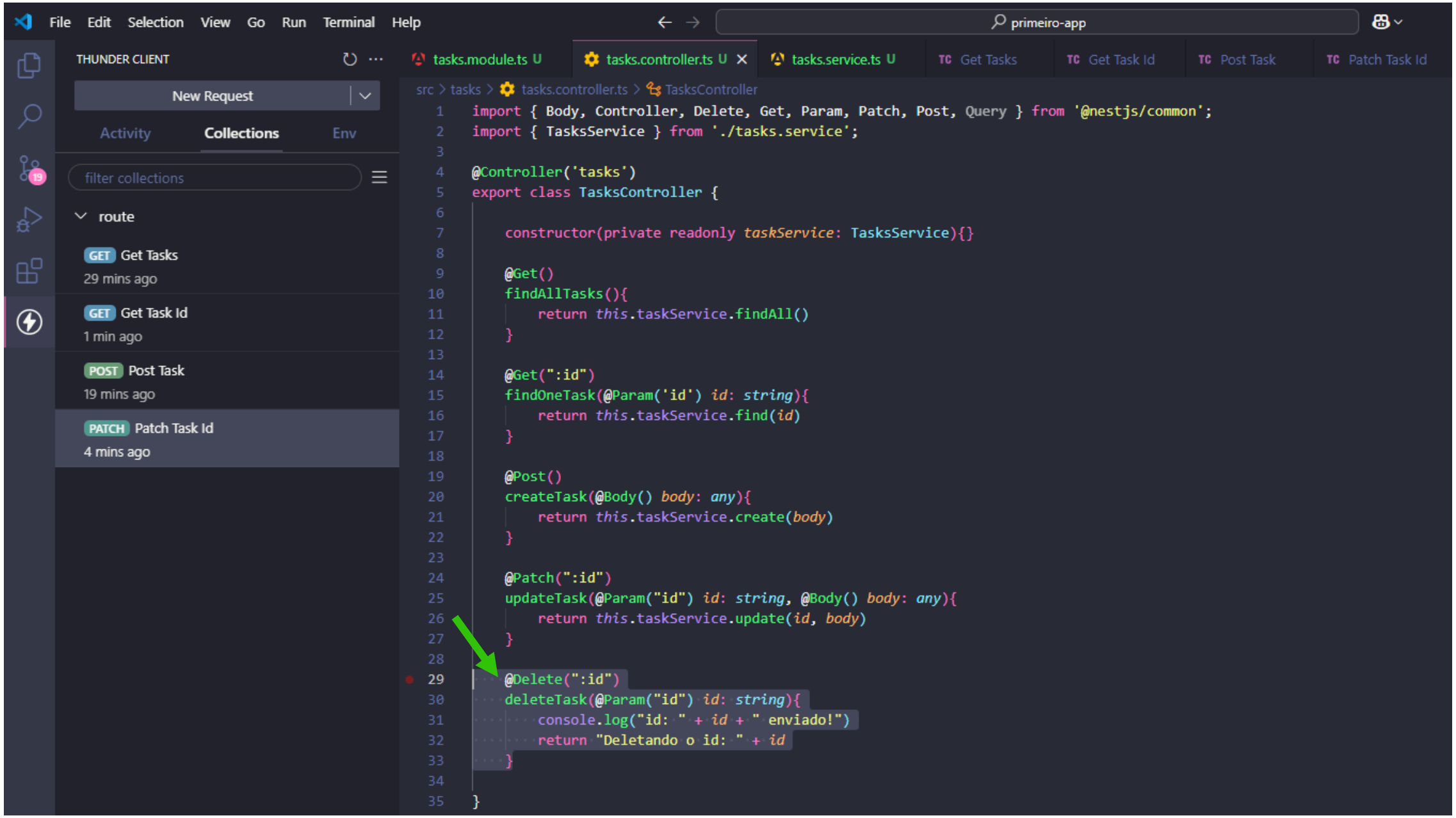


Prof. Dr. Robyson Aggio

Rotas com Patch

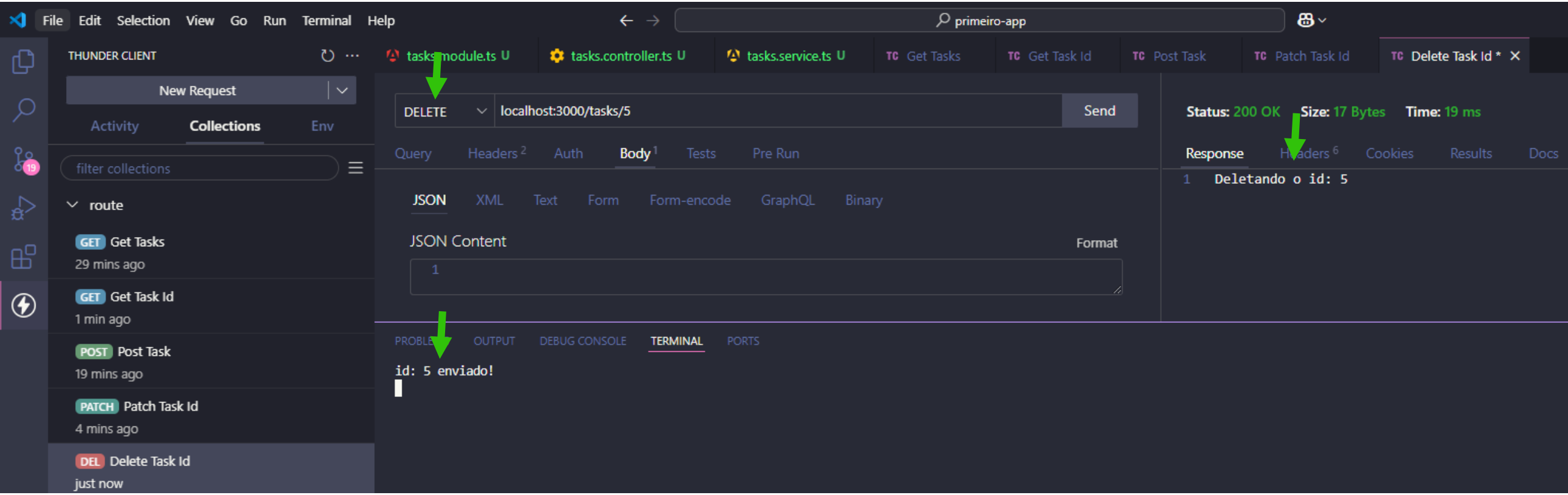
Prof. Dr. Robyson Aggio

Rotas com Delete



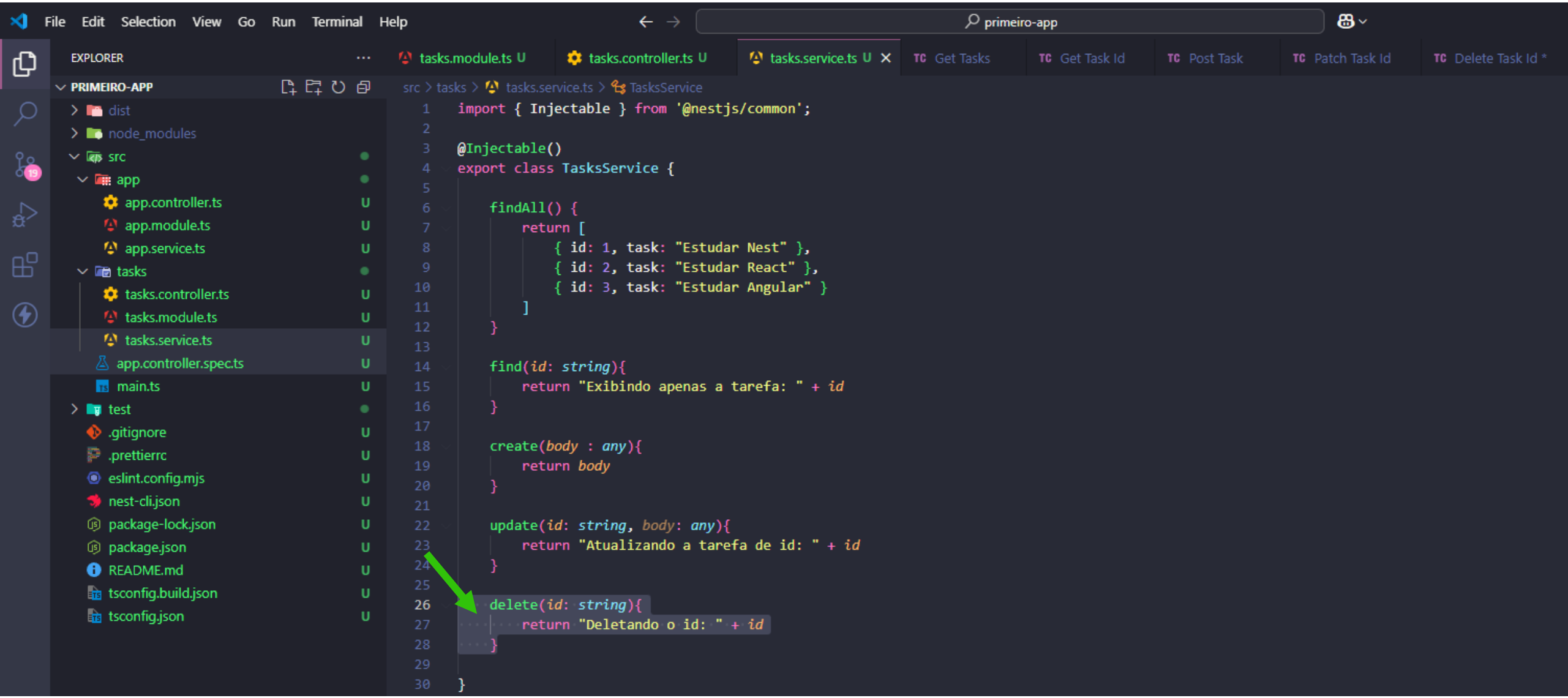
Prof. Dr. Robyson Aggio

Rotas com Delete



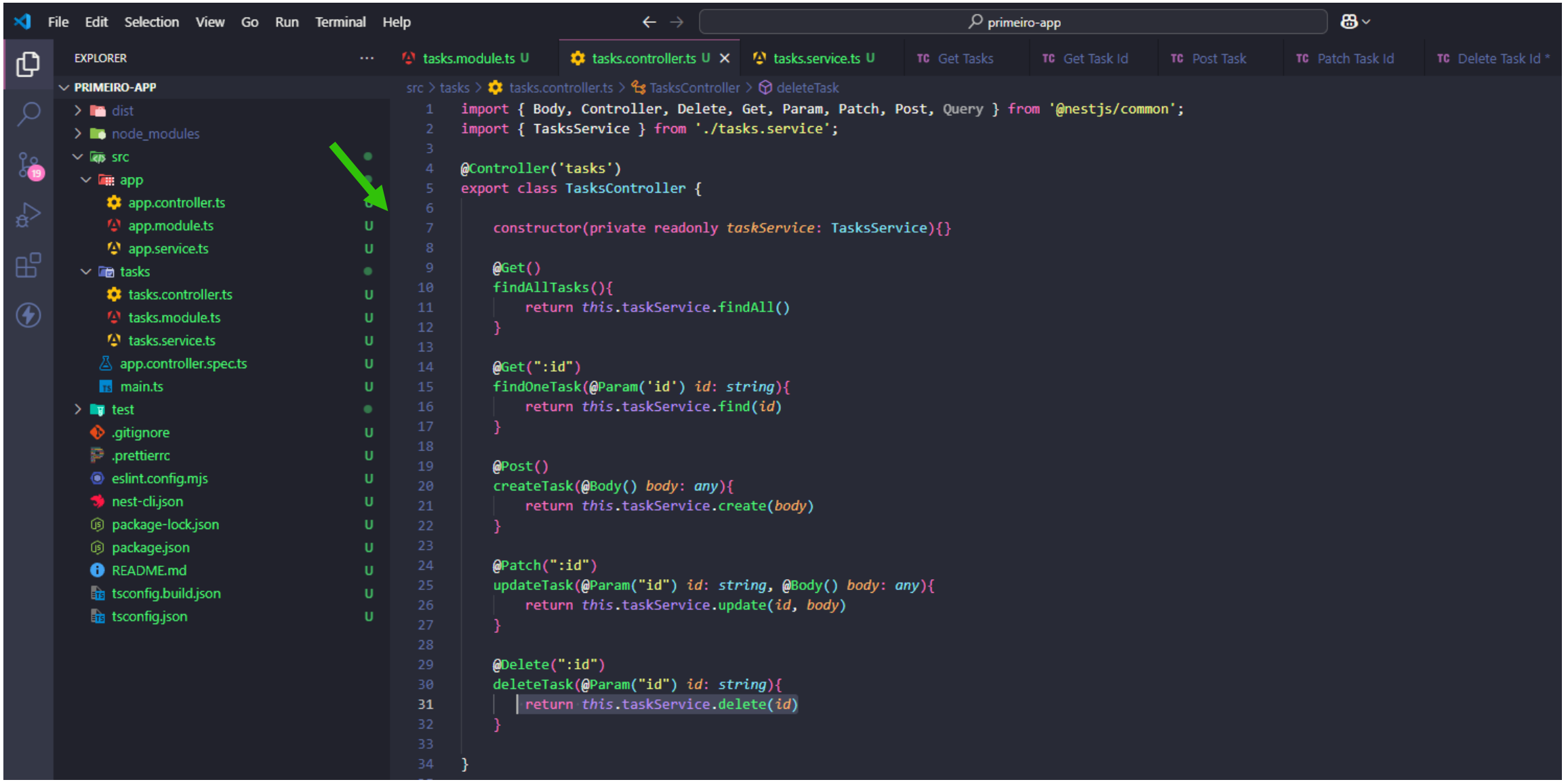
Prof. Dr. Robyson Aggio

Rotas com Delete

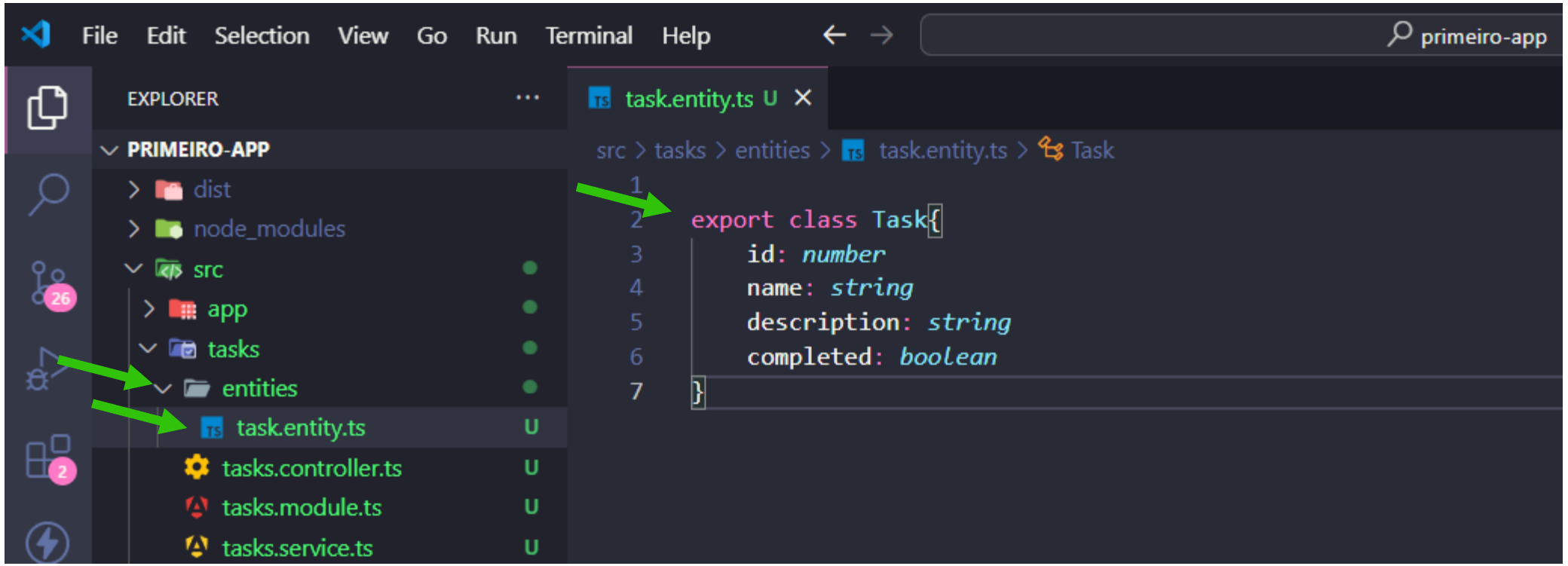


Prof. Dr. Robyson Aggio

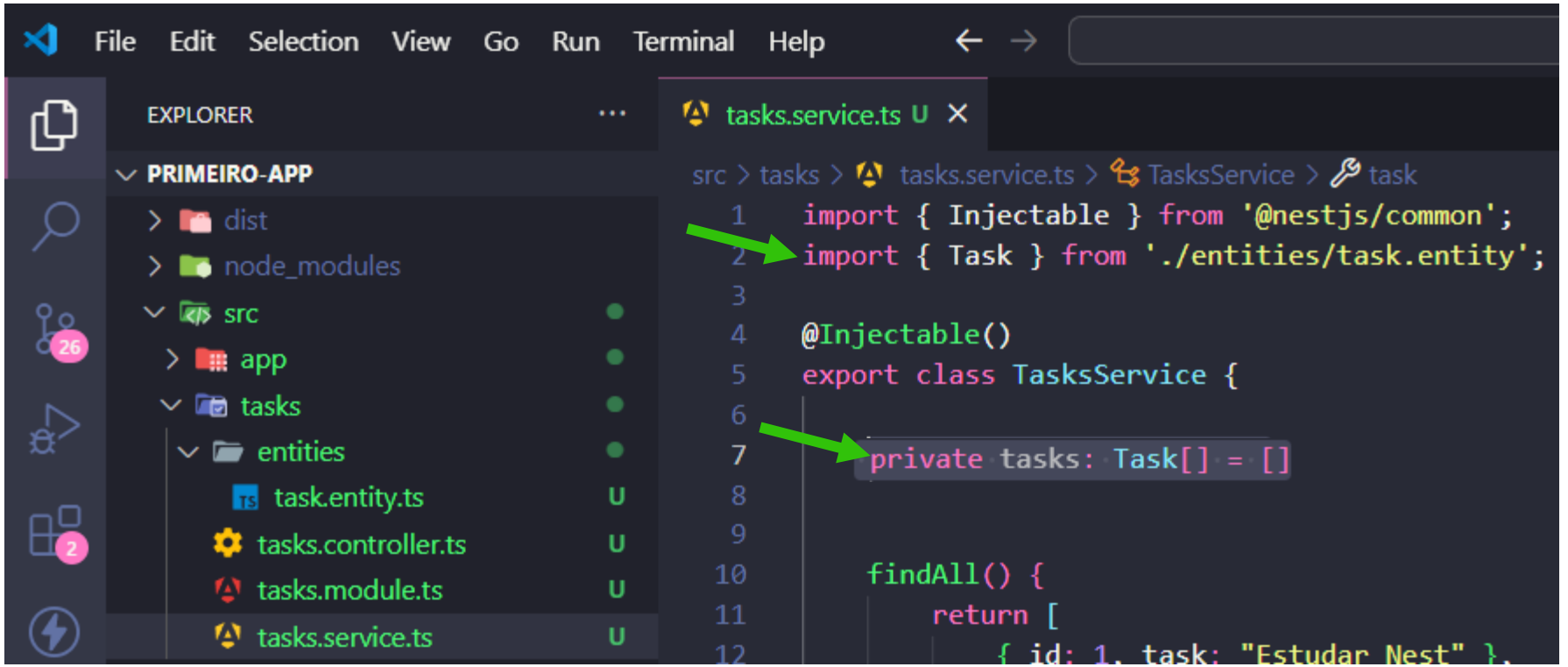
Rotas com Delete



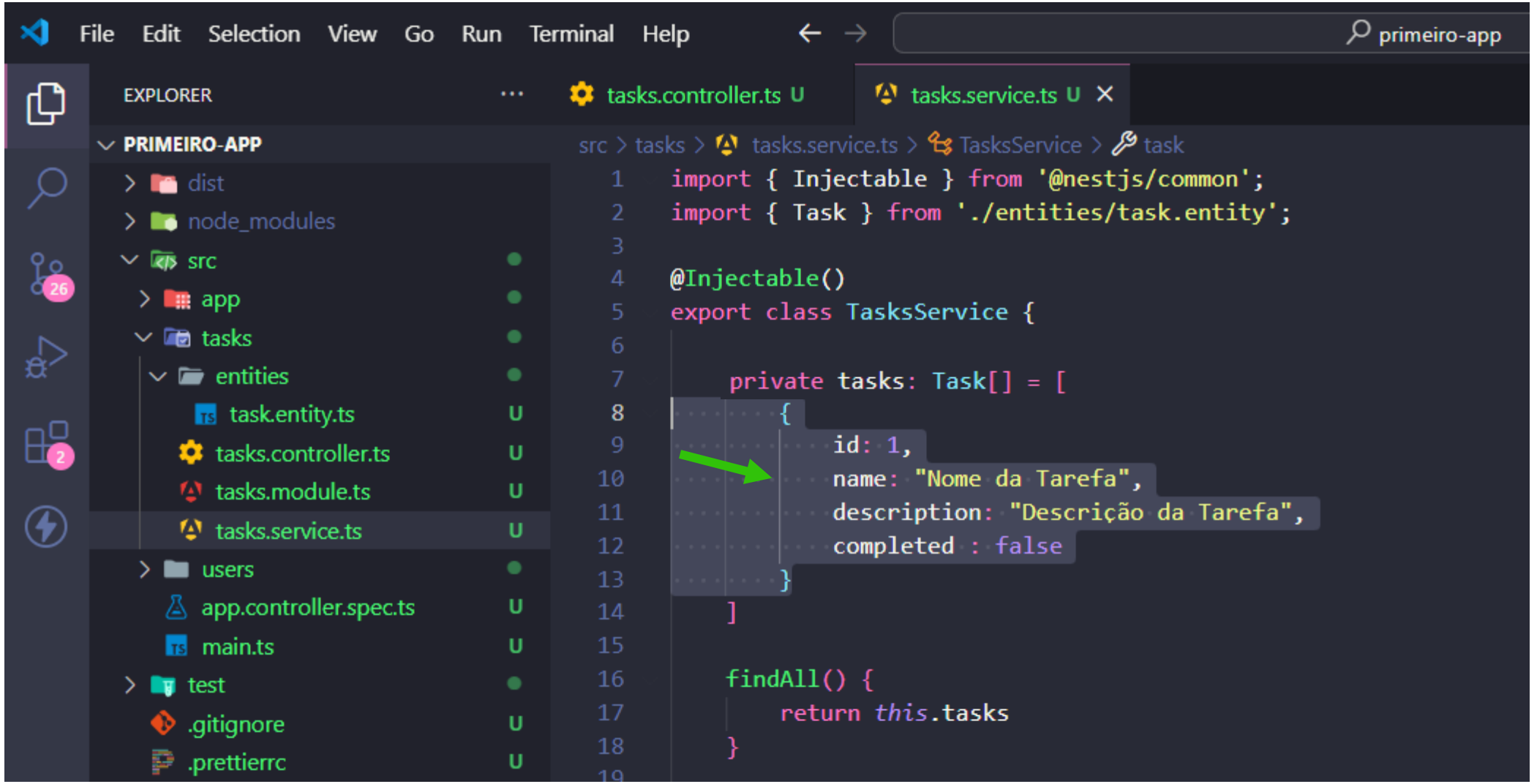
Listando Itens



Listando Itens

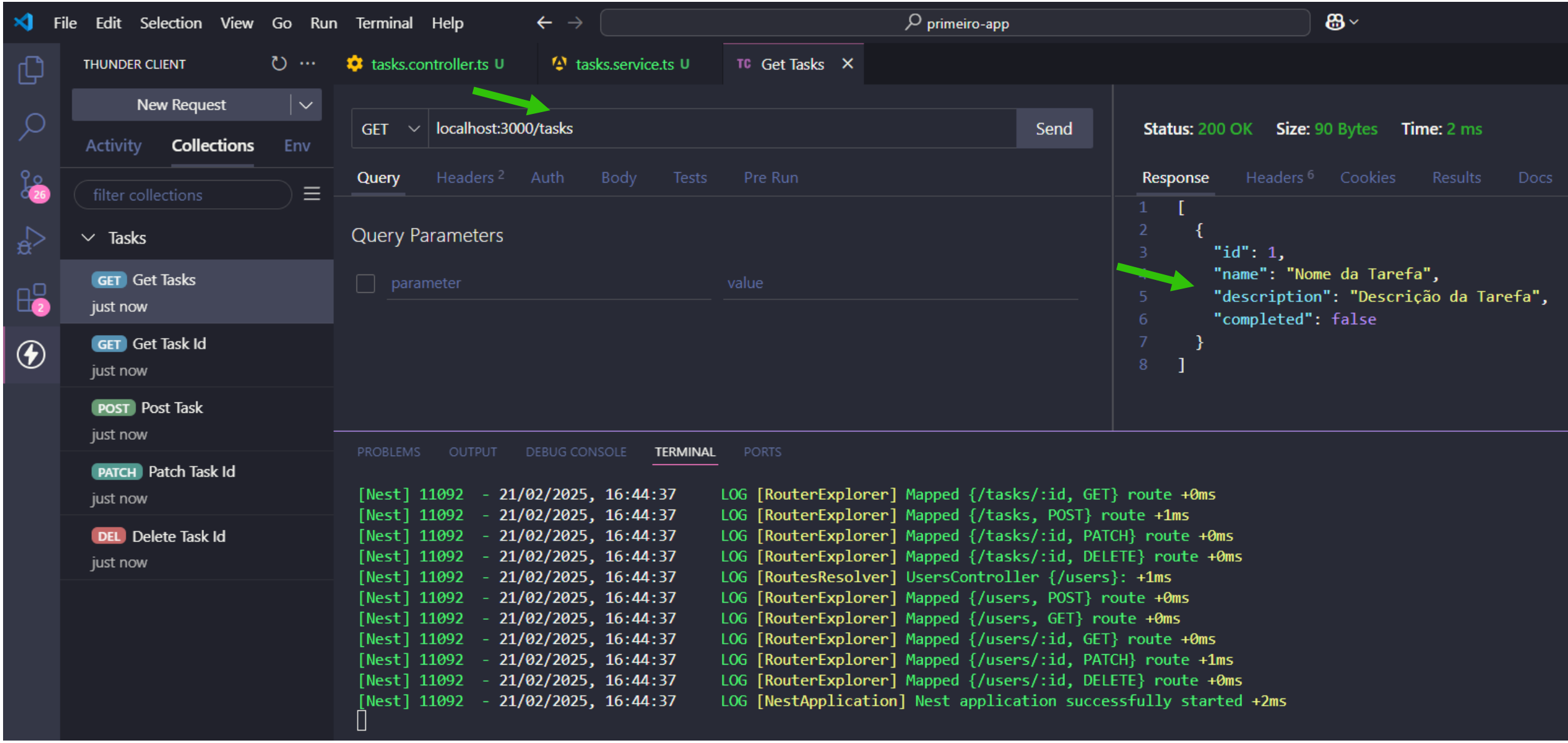


Listando Itens

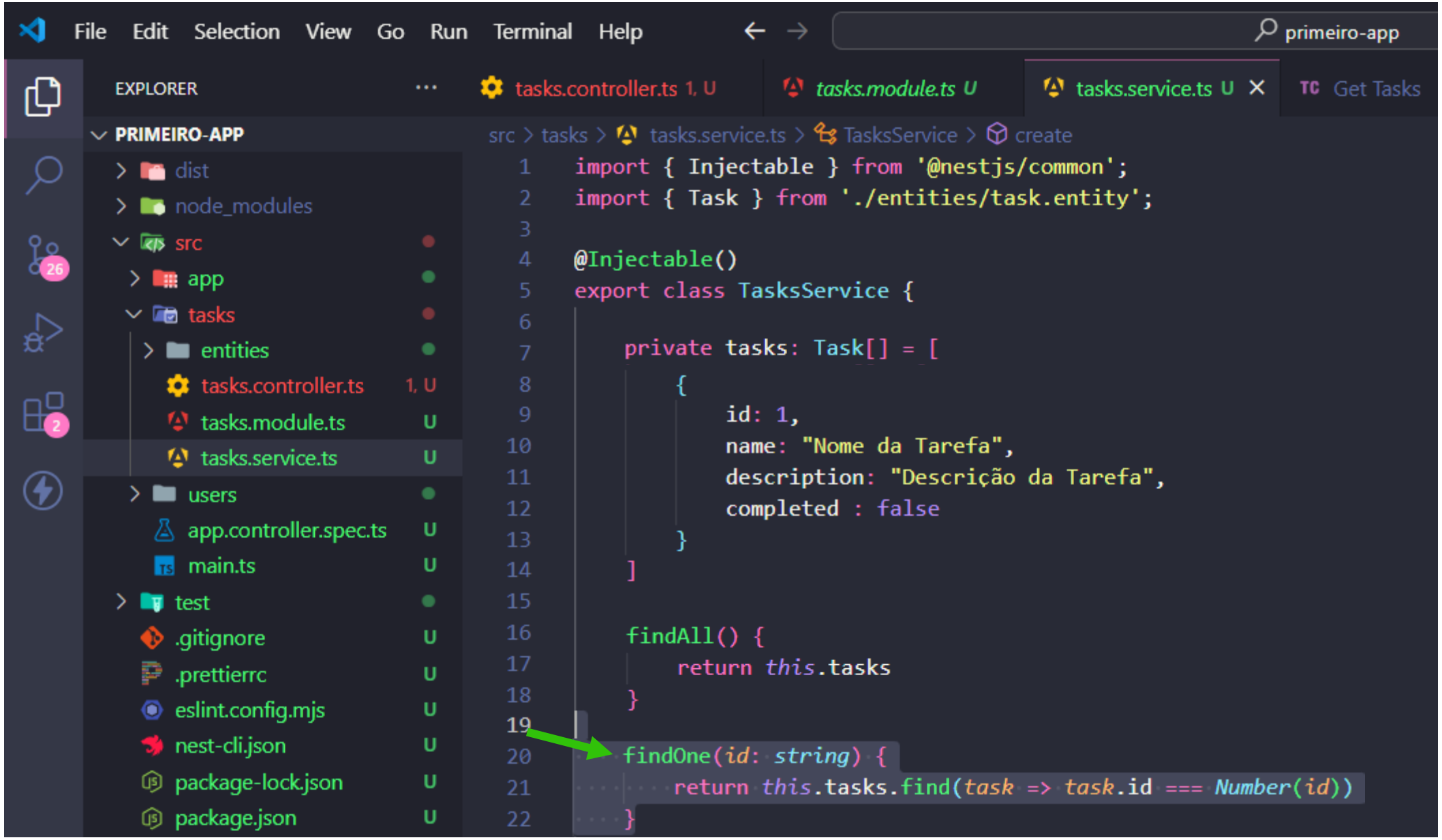


Prof. Dr. Robyson Aggio

Listando Itens

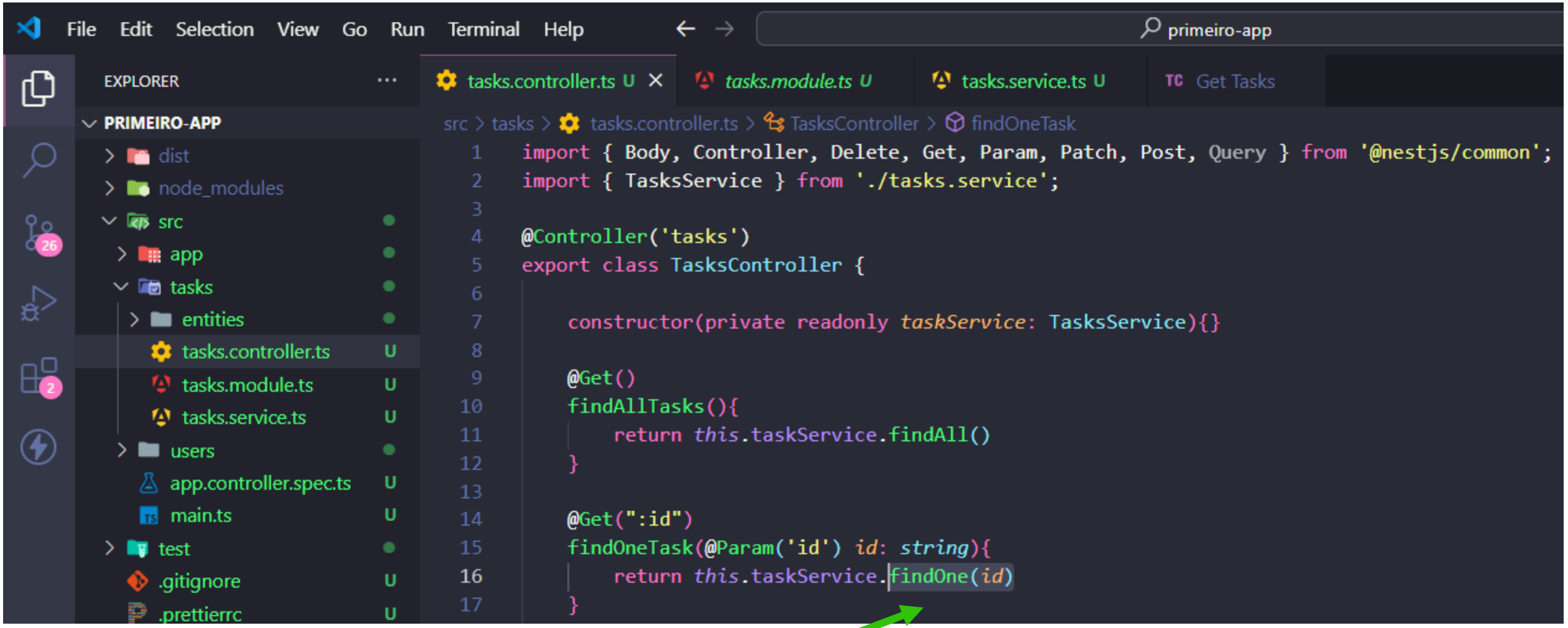


Listando Itens



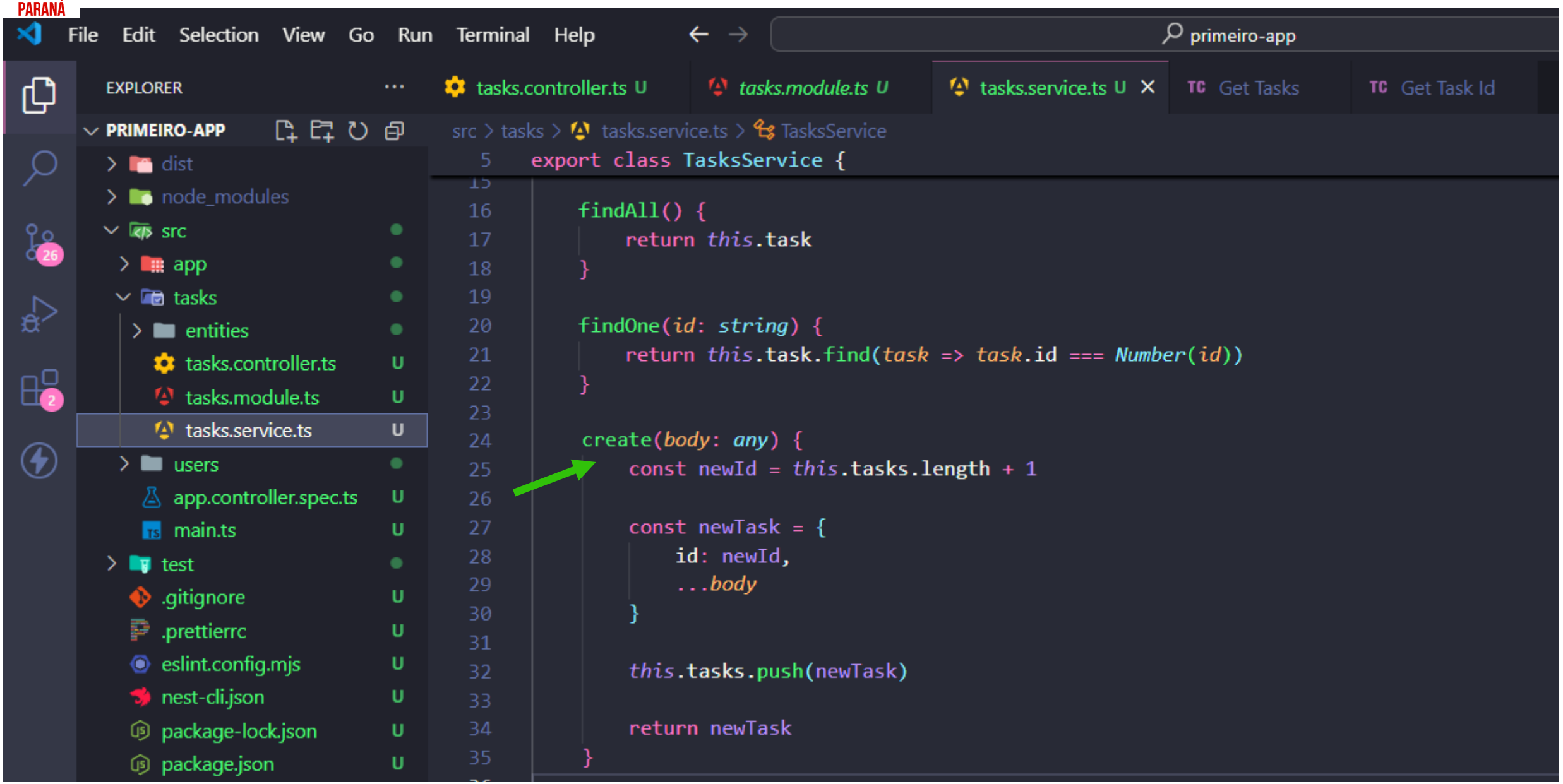
Prof. Dr. Robyson Aggio

Listando Itens



Prof. Dr. Robyson Aggio

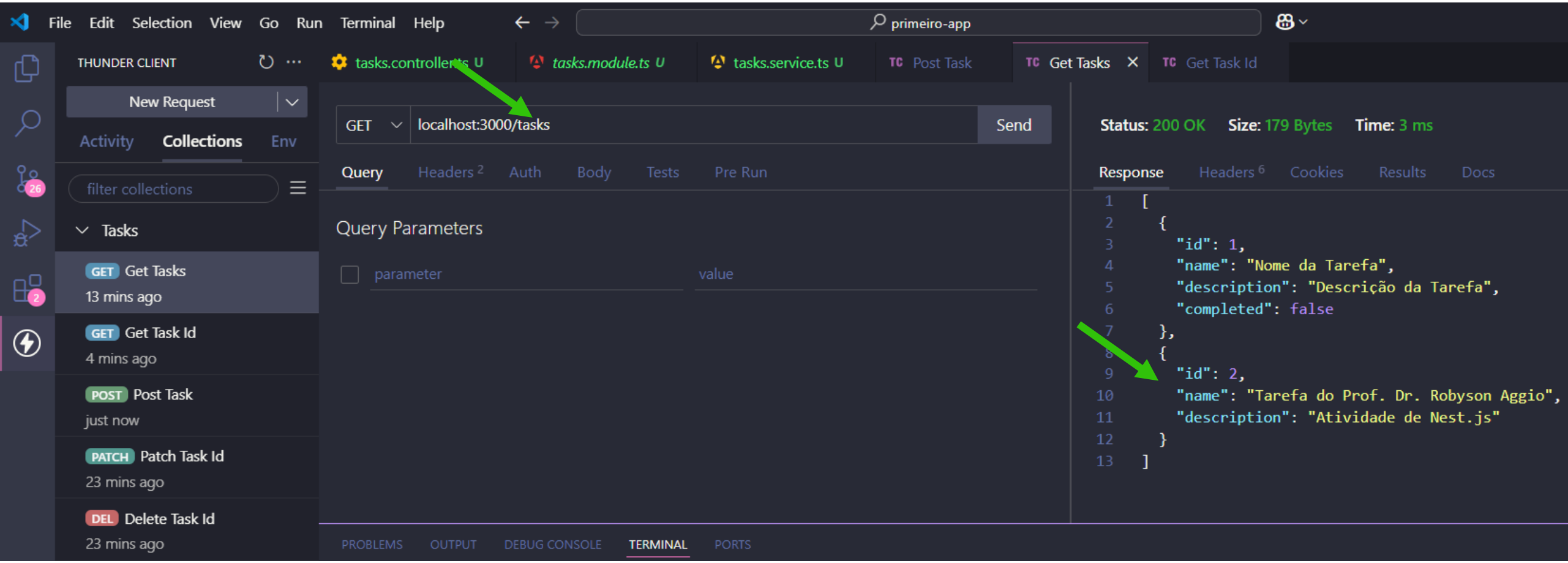
Listando Itens



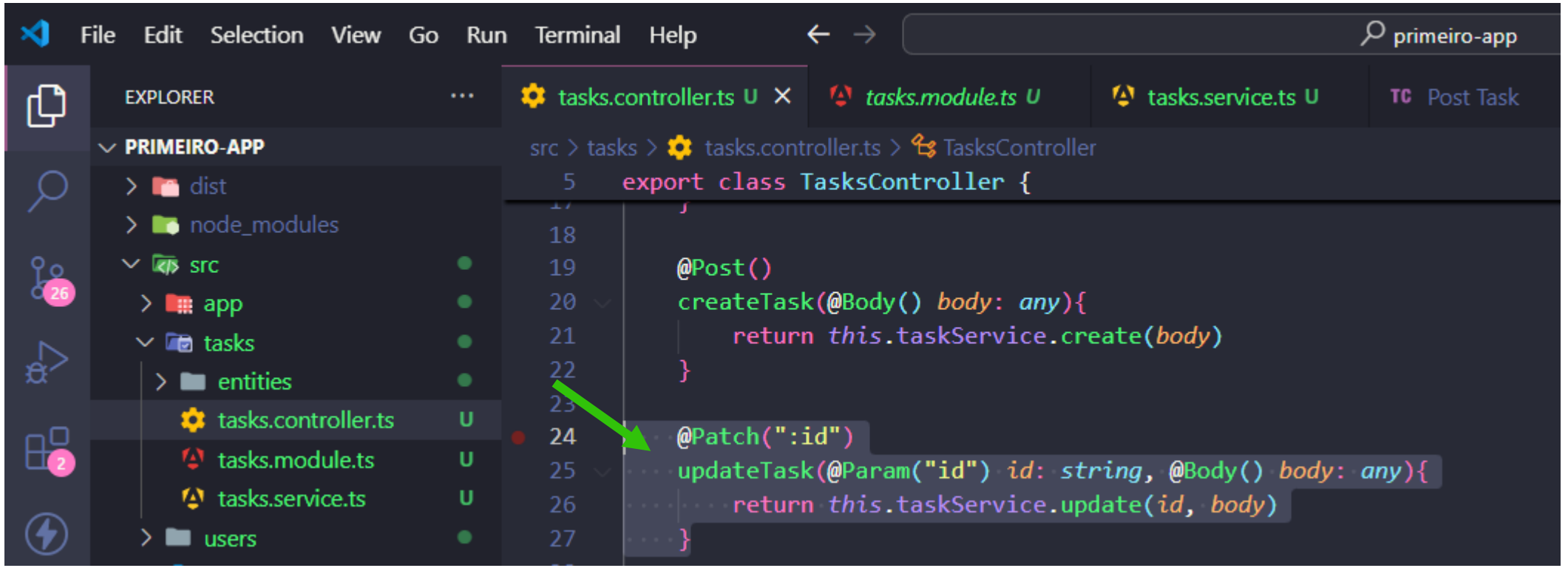
Listando Itens

Prof. Dr. Robyson Aggio

Listando Itens

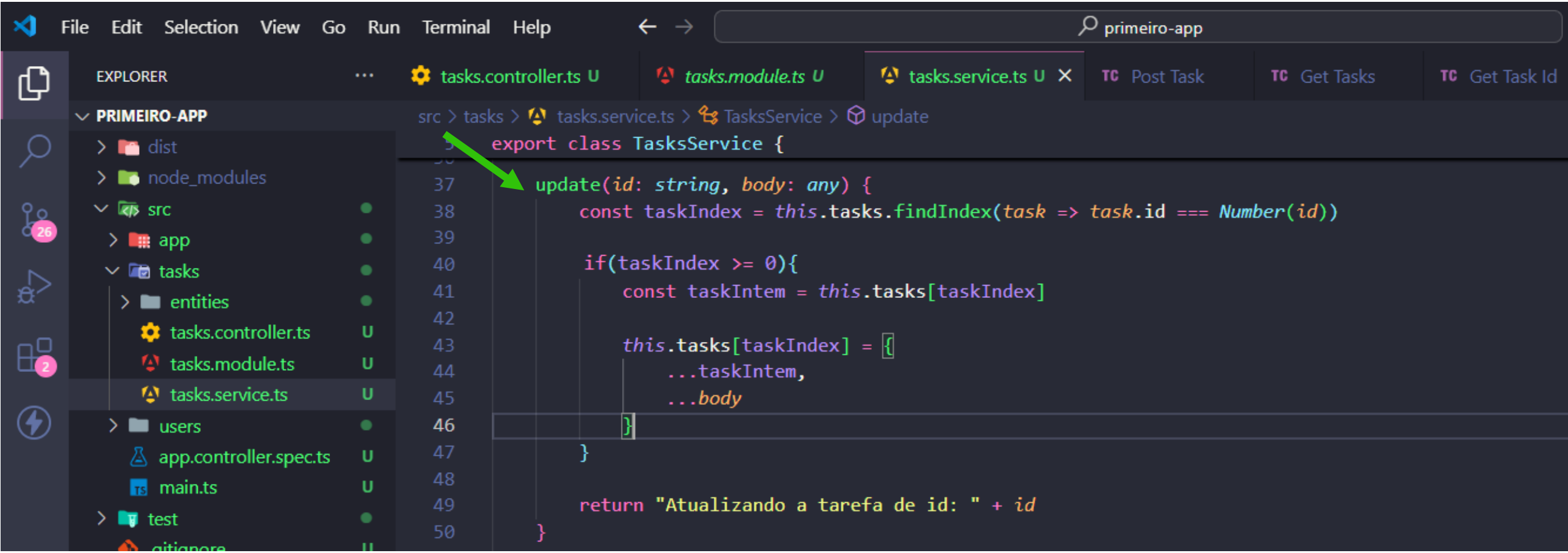


Atualizando Itens

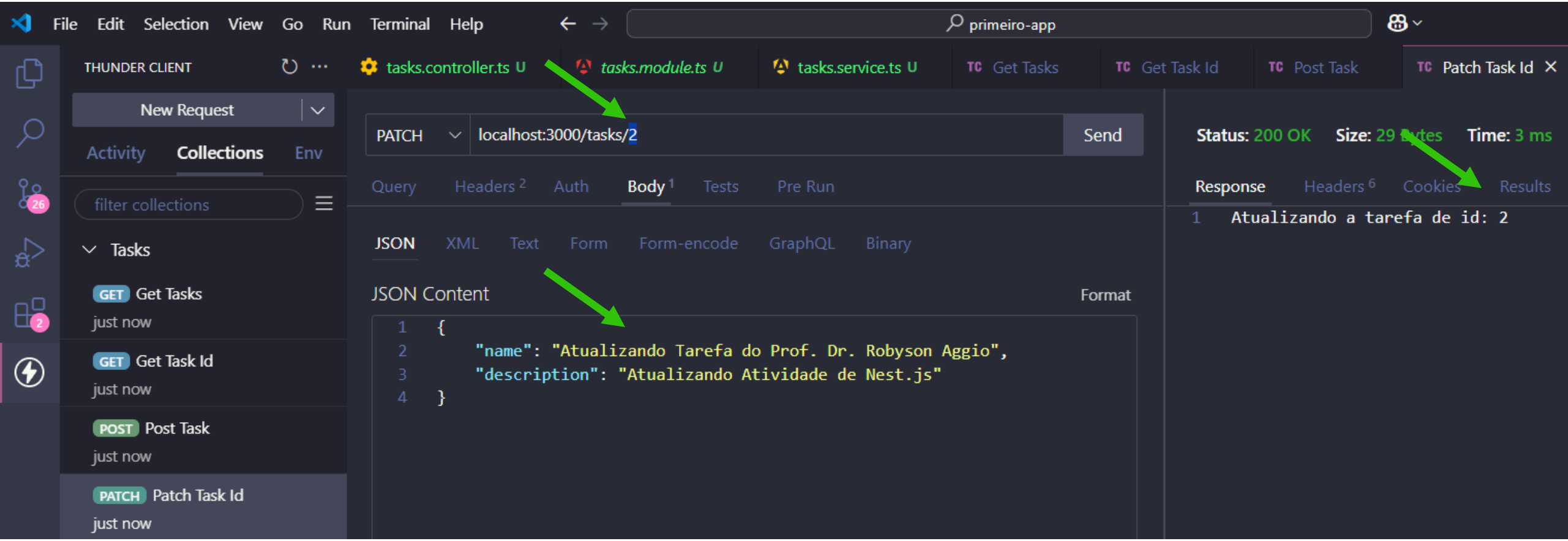


Prof. Dr. Robyson Aggio

Atualizando Itens

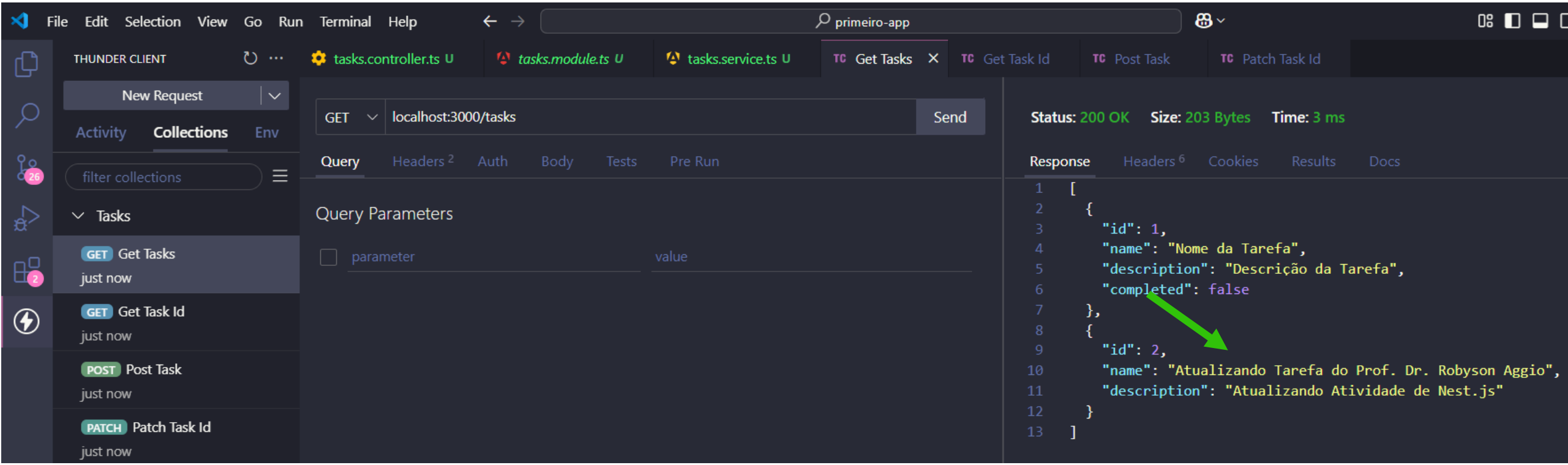


Atualizando Itens



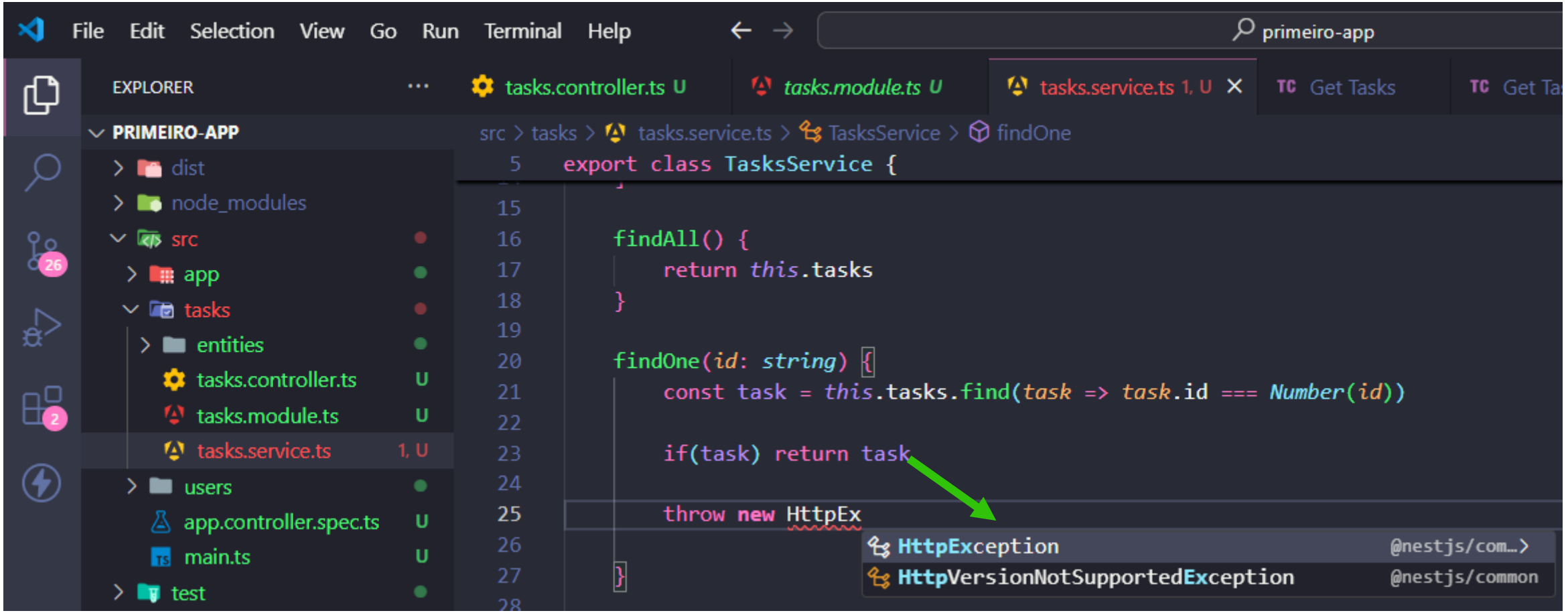
Prof. Dr. Robyson Aggio

Atualizando Itens



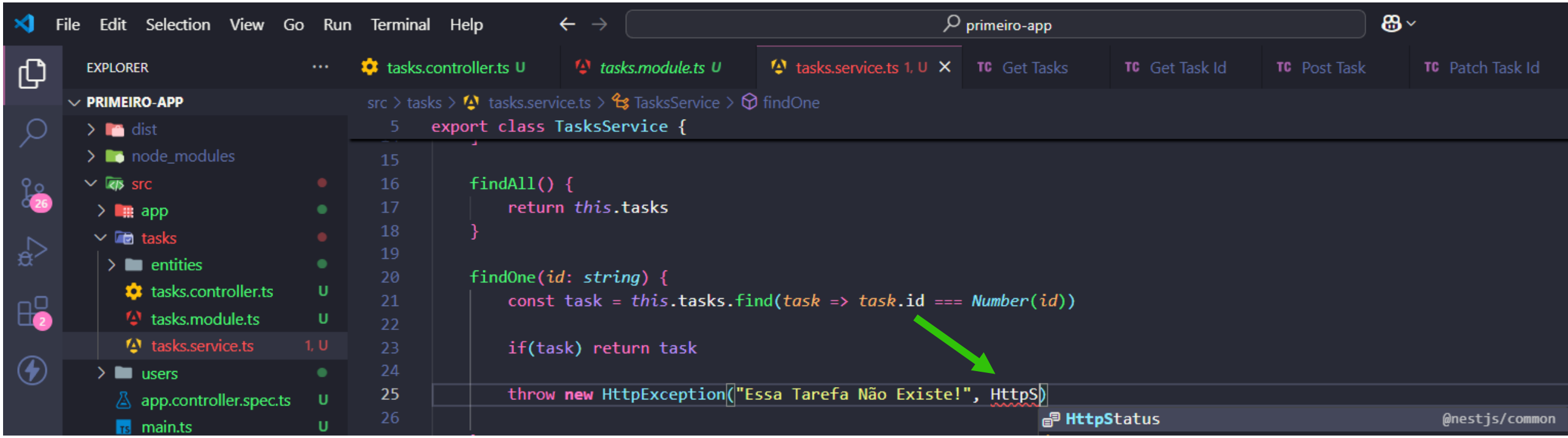
Prof. Dr. Robyson Aggio

Tratando erros

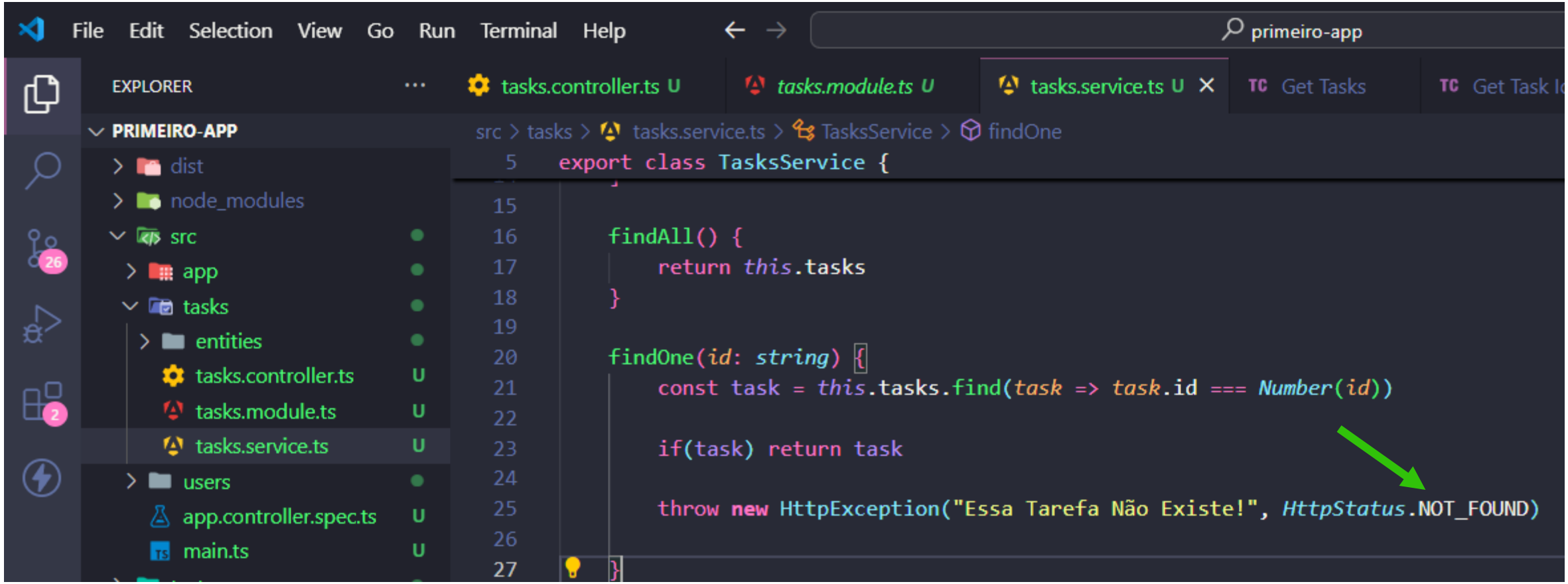


Prof. Dr. Robyson Aggio

Tratando erros

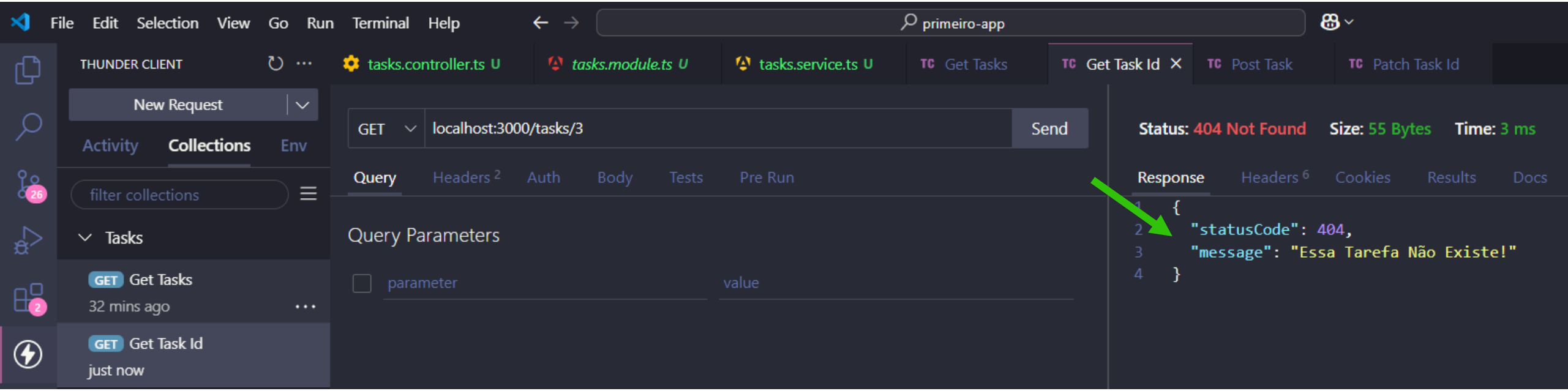


Tratando erros

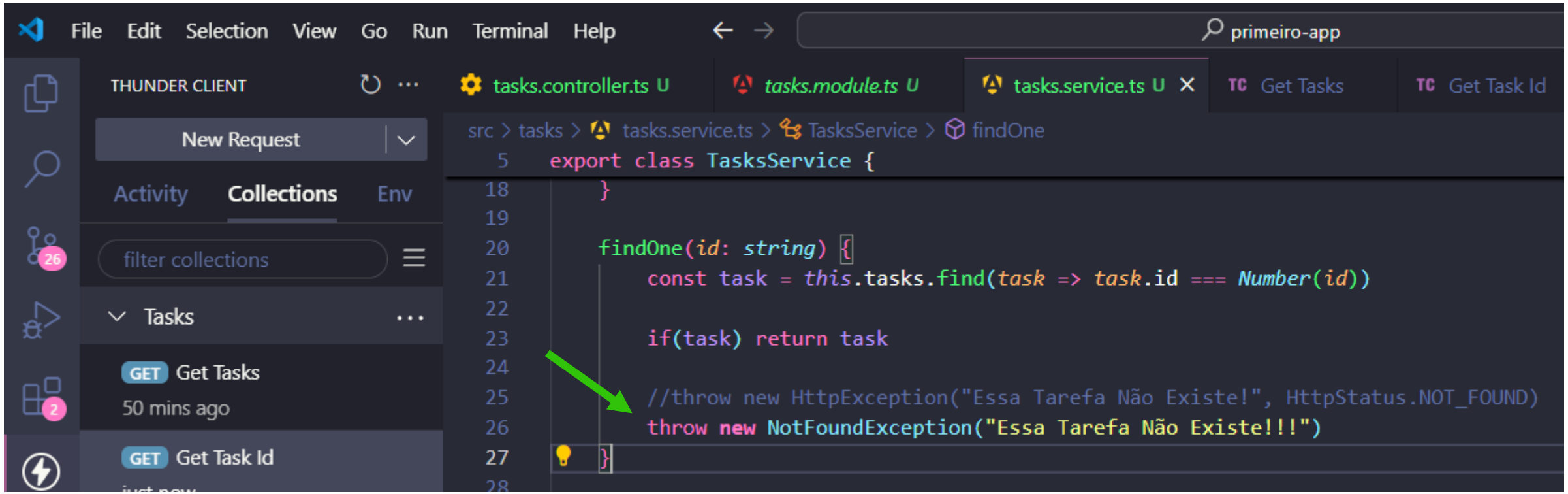


Prof. Dr. Robyson Aggio

Tratando erros

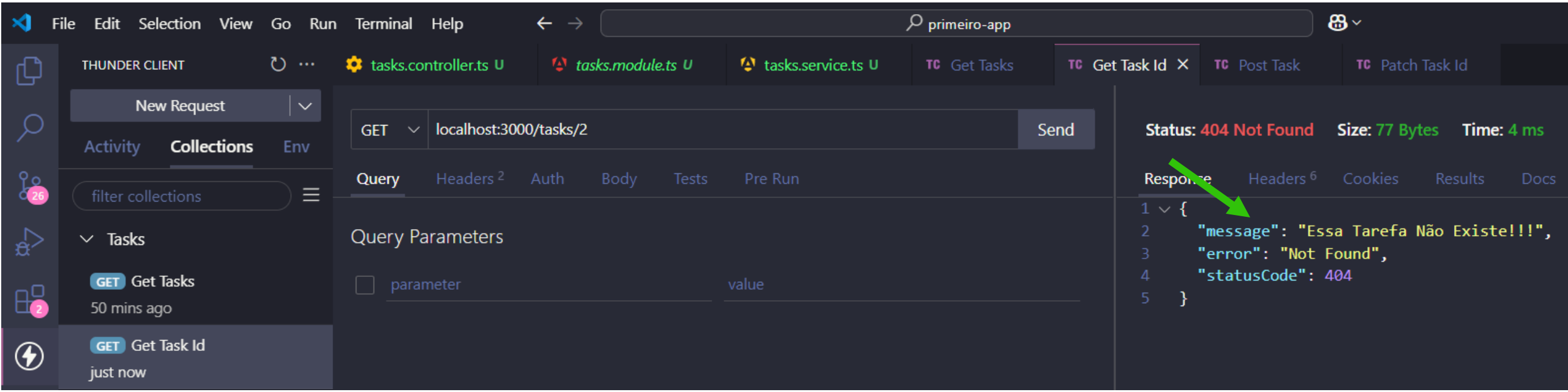


Tratando erros



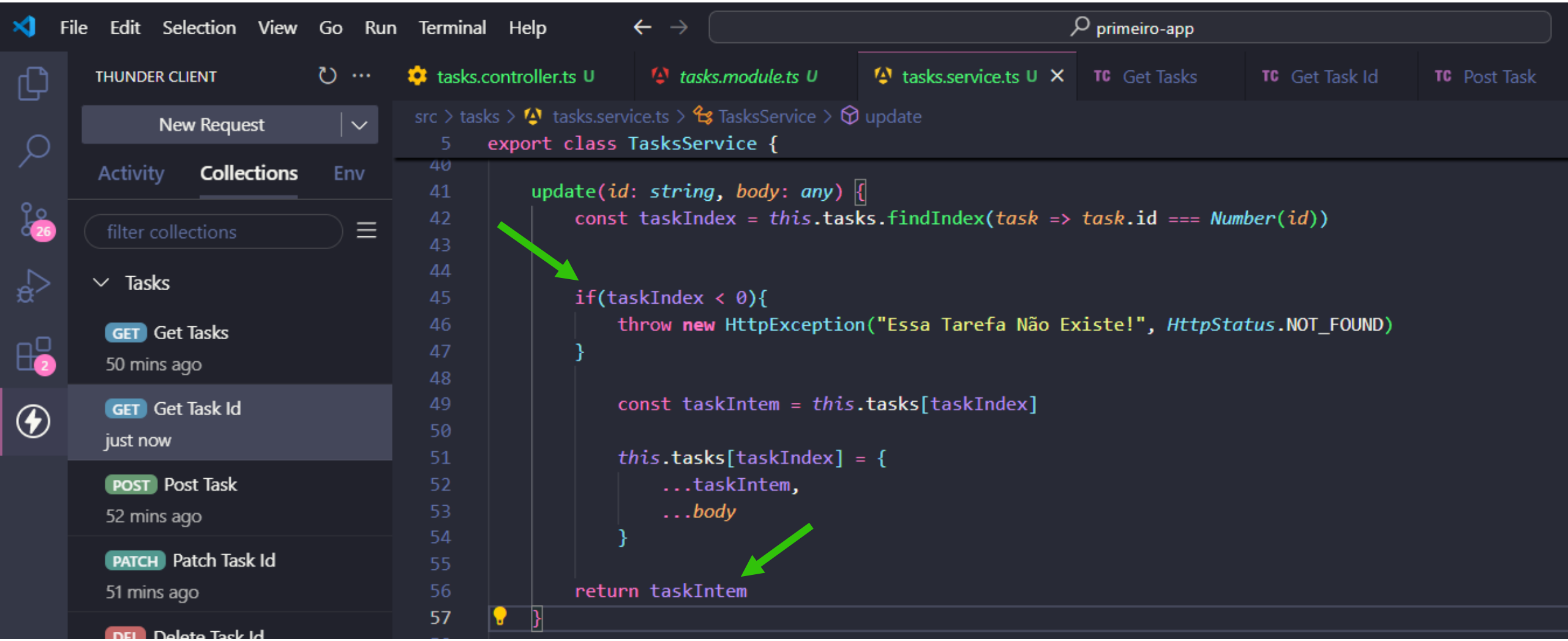
Prof. Dr. Robyson Aggio

Tratando erros

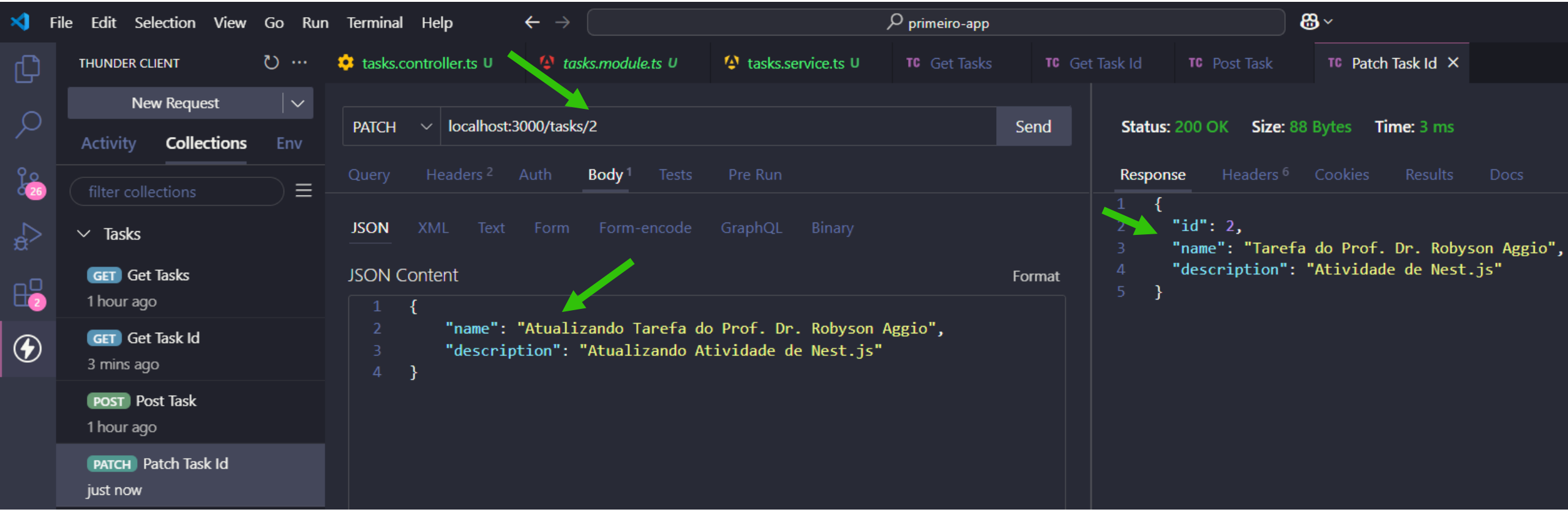


Prof. Dr. Robyson Aggio

Tratando erros

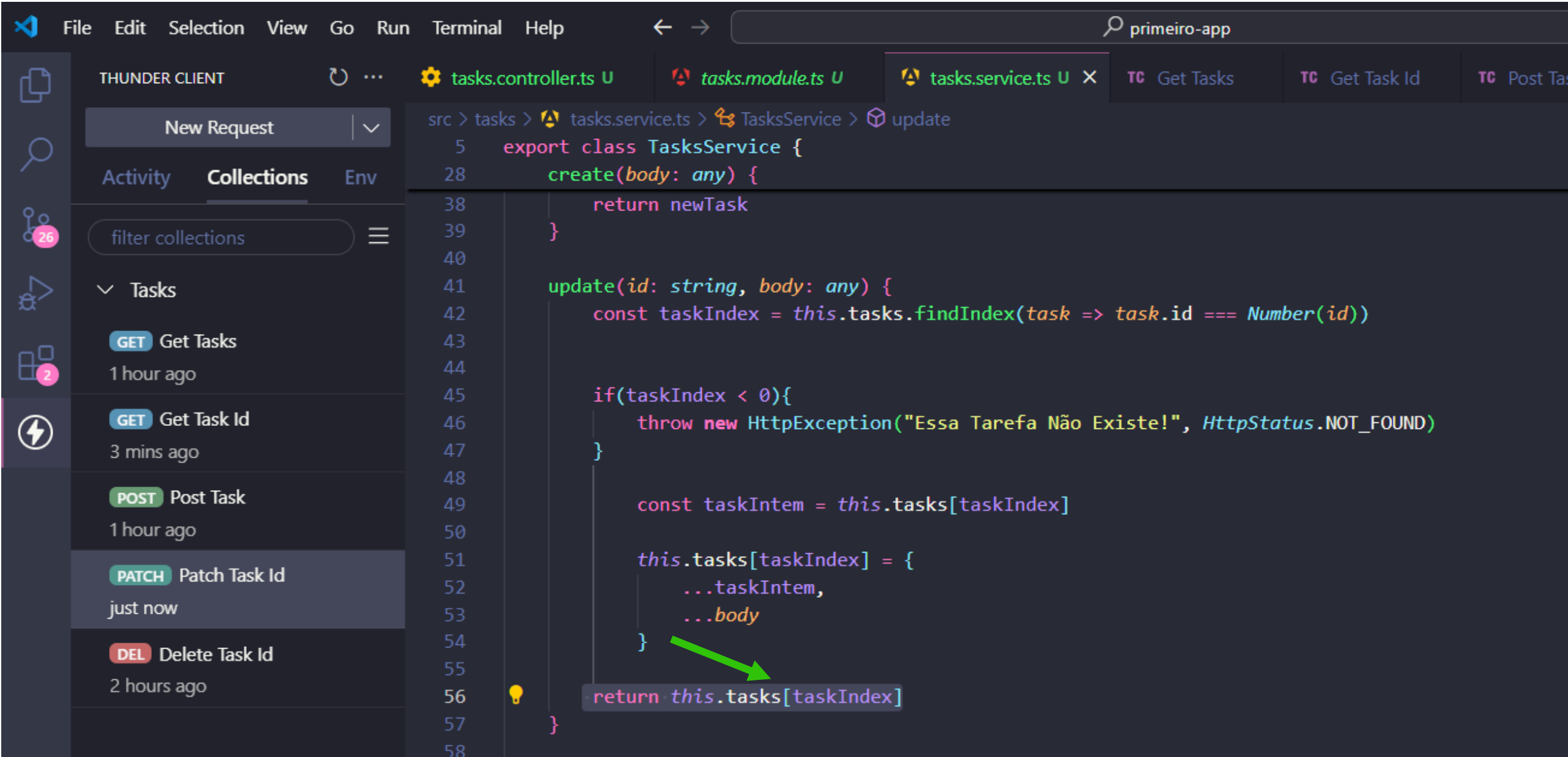


Tratando erros

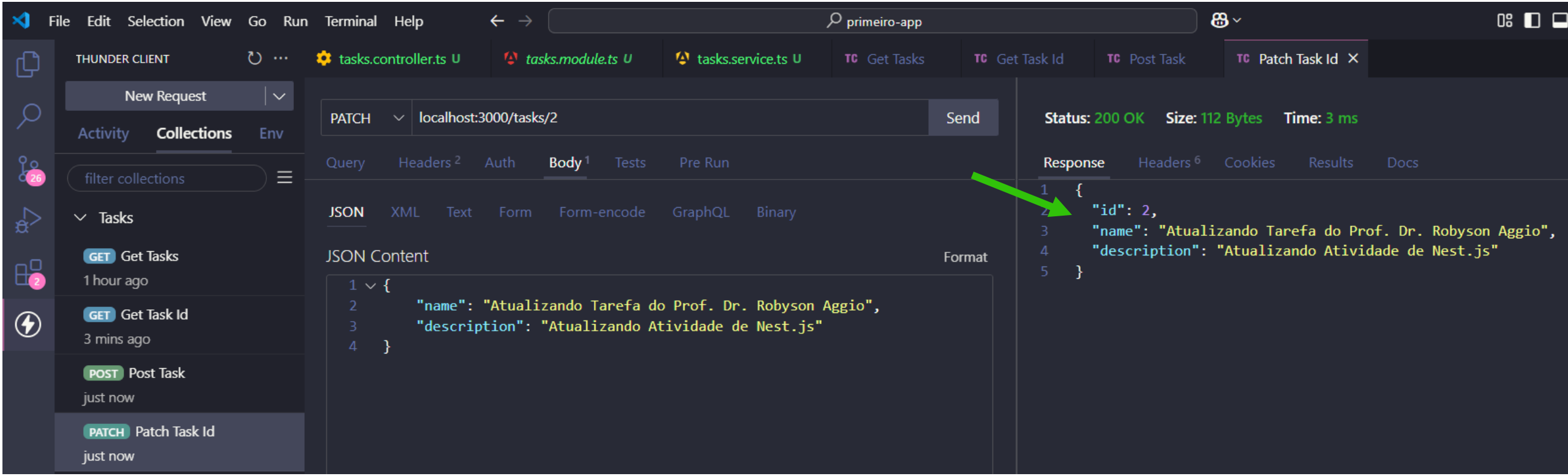


Prof. Dr. Robyson Aggio

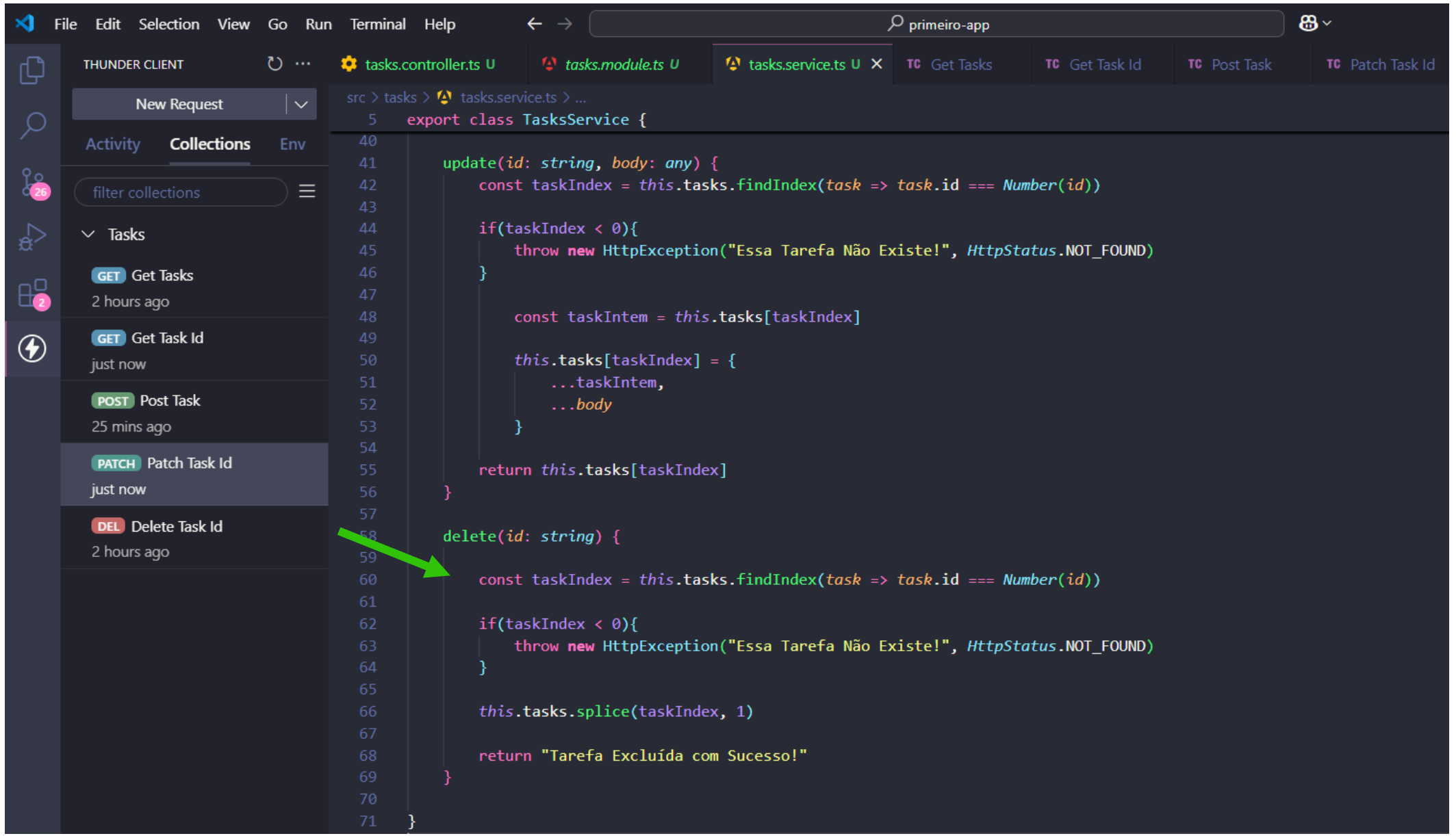
Tratando erros



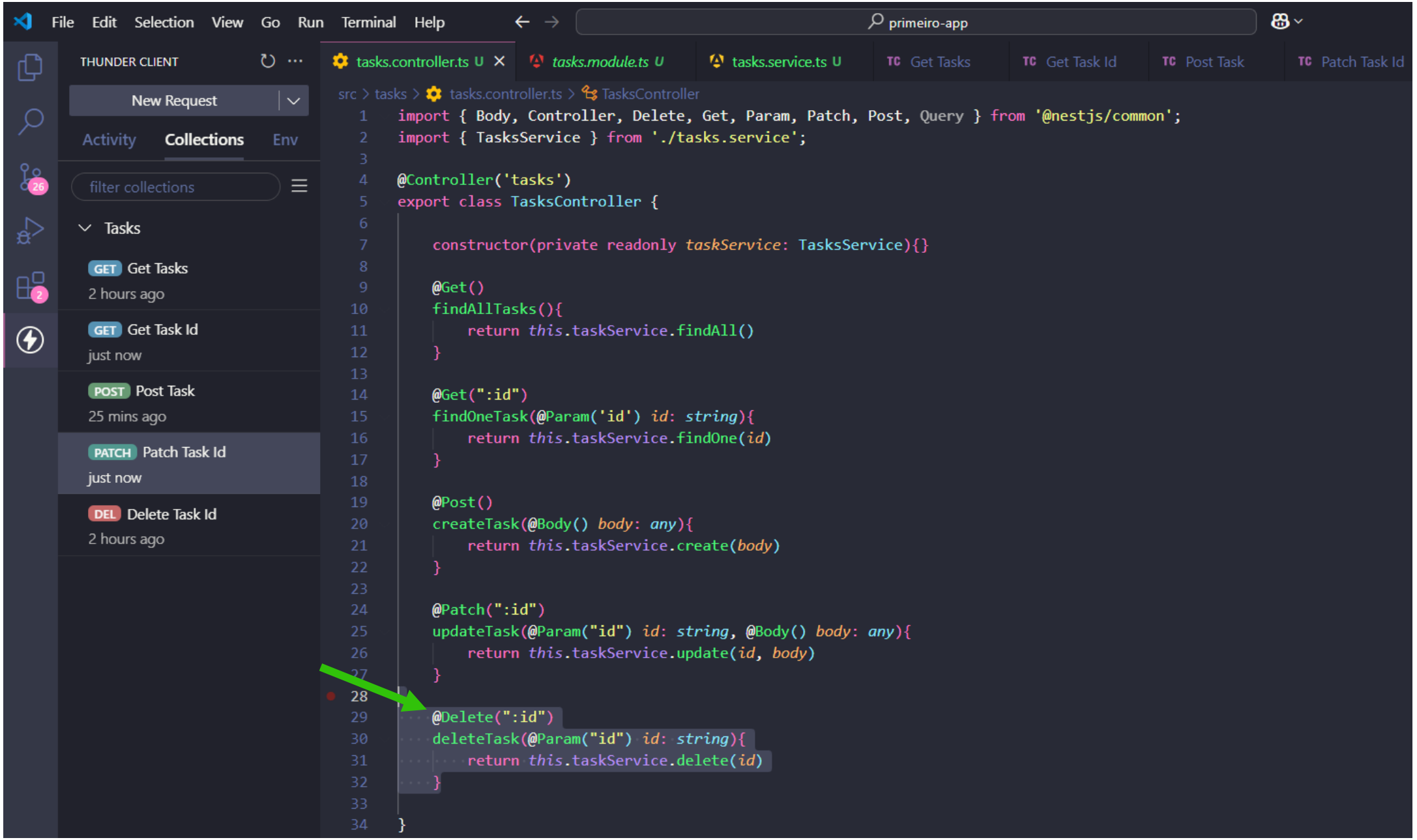
Tratando erros



Delete

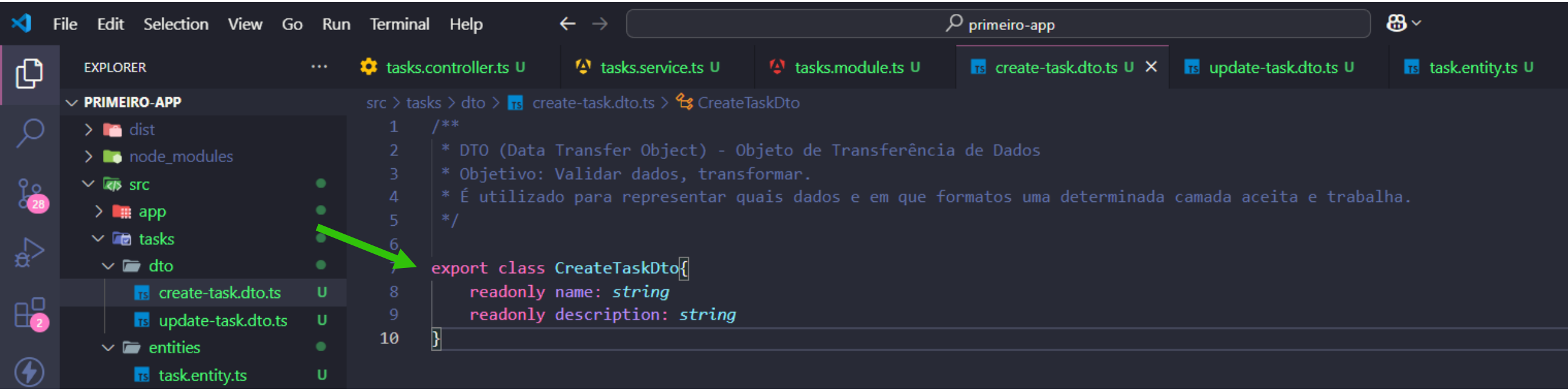

File Edit Selection View Go Run Terminal Help
primeiro-app
THUNDER CLIENT
New Request
Activity Collections Env
filter collections
Tasks
GET Get Tasks
2 hours ago
GET Get Task Id
just now
POST Post Task
25 mins ago
PATCH Patch Task Id
just now
DEL Delete Task Id
2 hours ago
tasks.controller.ts U
tasks.module.ts U
tasks.service.ts U
Get Tasks
Get Task Id
Post Task
Patch Task Id
src > tasks > tasks.service.ts > ...
export class TasksService {
update(id: string, body: any) {
const taskIndex = this.tasks.findIndex(task => task.id === Number(id))
if(taskIndex < 0){
throw new HttpException("Essa Tarefa Não Existe!", HttpStatus.NOT_FOUND)
}
const taskIntem = this.tasks[taskIndex]
this.tasks[taskIndex] = {
...taskIntem,
...body
}
return this.tasks[taskIndex]
}
delete(id: string) {
const taskIndex = this.tasks.findIndex(task => task.id === Number(id))
if(taskIndex < 0){
throw new HttpException("Essa Tarefa Não Existe!", HttpStatus.NOT_FOUND)
}
this.tasks.splice(taskIndex, 1)
return "Tarefa Excluída com Sucesso!"
}
}

Delete

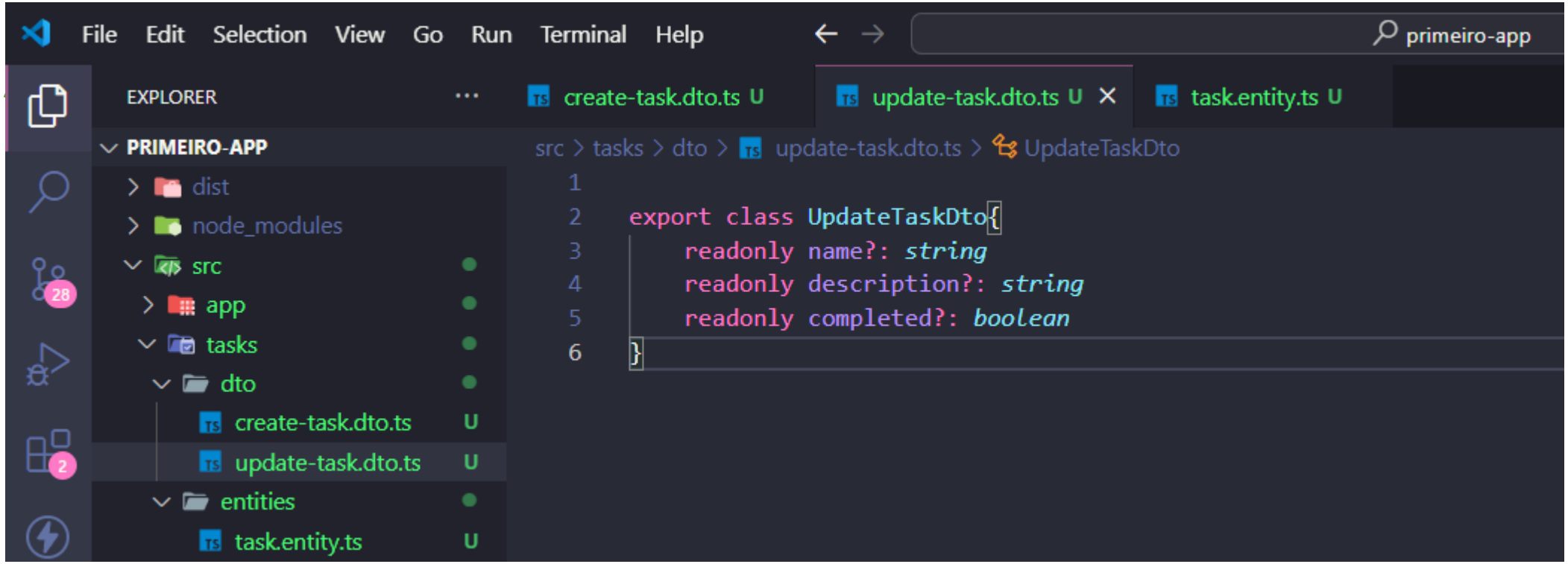

File Edit Selection View Go Run Terminal Help
primeiro-app
THUNDER CLIENT
New Request
Activity Collections Env
filter collections
Tasks
GET Get Tasks
2 hours ago
GET Get Task Id
just now
POST Post Task
25 mins ago
PATCH Patch Task Id
just now
DEL Delete Task Id
2 hours ago
tasks.controller.ts U
tasks.module.ts U
tasks.service.ts U
Get Tasks
Get Task Id
Post Task
Patch Task Id
src > tasks > tasks.controller.ts > TasksController
import { Body, Controller, Delete, Get, Param, Patch, Post, Query } from '@nestjs/common';
import { TasksService } from './tasks.service';
@Controller('tasks')
export class TasksController {
constructor(private readonly taskService: TasksService){}
@Get()
findAllTasks(){
return this.taskService.findAll()
}
@Get(":id")
findOneTask(@Param('id') id: string){
return this.taskService.findOne(id)
}
@Post()
createTask(@Body() body: any){
return this.taskService.create(body)
}
@Patch(":id")
updateTask(@Param("id") id: string, @Body() body: any){
return this.taskService.update(id, body)
}
@Delete(":id")
deleteTask(@Param("id") id: string){
return this.taskService.delete(id)
}
}

Prof. Dr. Robyson Aggio

Criando DTO



Atualizando DTO



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

create-task.dto.ts U | tasks.controller.ts U | tasks.service.ts U | update-task.dto.ts U | task.entity.ts U

src > tasks > tasks.controller.ts > TasksController

```ts
import { Body, Controller, Delete, Get, Param, Patch, Post, Query } 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') id: string){
        return this.taskService.findOne(id)
    }

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

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

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

}
```

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

src > tasks > tasks.service.ts > TasksService > delete

```
7   export class TasksService {
29
30      create(createTaskDto: CreateTaskDto) {
31          const newId = this.tasks.length + 1
32
33          const newTask = {
34              id: newId,
35              ...createTaskDto,
36              completed: false
37          }
38
39          this.tasks.push(newTask)
40
41          return newTask
42      }
43
44      update(id: string, updateTaskDto: UpdateTaskDto) {
45          const taskIndex = this.tasks.findIndex(task => task.id === Number(id))
46
47          if(taskIndex < 0){
48              throw new HttpException("Essa Tarefa Não Existe!", HttpStatus.NOT_FOUND)
49          }
50
51              const taskIntem = this.tasks[taskIndex]
52
53              this.tasks[taskIndex] = {
54                  ...taskIntem,
55                  ...updateTaskDto
56              }
57
58          return this.tasks[taskIndex]
59      }
```

INSTITUTO FEDERAL
PARANÁ

Atividade

1) Dentro de cada camada: Guests, Users e Teachers, implemente os verbos: Post, Patch e Delete

2) Crie a pasta entities para cada camada e a respectiva Entidade.

3) Crie a pasta dto para cada camada e dentro os arquivos: create*Dto e update*Dto

4) Teste todos os verbos HTPP (Get, Post, Patch, Delete)