



INSTITUTO INFNET  
PROJETO DE BLOCO: PROCESSAMENTO DE DADOS  
FACULDADE DE ENGENHARIA DE SOFTWARE

DANIEL GOMES LIPKIN  
ANTONIO SILVEIRA DEMARCHI

TESTE DE PERFORMANCE – TP5

PROFESSOR(A): LEONARDO SILVA DA GLÓRIA

16 de set. de 2024

## Repositórios de Github

<https://github.com/Bagelboi/ProjetoDeBlocoEscalaveis> - Configurações do docker-compose

<https://github.com/DemarchiWorking/EurekaServer/tree/main> - Servidor Eureka

<https://github.com/Bagelboi/GatewayServicePB/tree/main> - Serviço Gateway

<https://github.com/DemarchiWorking/service-java-micro/tree/main> - Serviço "Principal" e base de dados dos Services dos Projetos

<https://github.com/DemarchiWorking/project-java-micro/tree/main> - Serviço e base de dados dos Projetos

<https://github.com/Bagelboi/Microservico-Comentarios-PB/tree/main> - Serviço e base de dados de Comentarios nos Projetos

<https://github.com/Bagelboi/alunoservice> - Serviço e base de dados dos Alunos

## Especificação do Projeto

### Nome

Blog de Projetos

### Bloco

Engenharia de Softwares Escaláveis

### Área de Negócio

Educacional

### Resumo das funcionalidades

Permitir que os alunos criem posts em cards com dicas de programação para turma. Outros alunos podem comentar suas opiniões também.

## Changelog

**0.1.0** 19/05/2024 Código Antonio Eduardo S. Demarchi

**0.2.0** 20/05/2024 Criação Documentação Antonio Eduardo S. Demarchi

**0.3.0** 10/06/2024 Melhoria Sistema Antonio Eduardo S. Demarchi

**0.3.1** 10/06/2024 Adaptações documentação Antonio Eduardo S. Demarchi

**0.3.2** 24/07/2024 Historico de Dados Daniel Gomes Lipkin

**0.4.0** 05/07/2024 Microserviço de Comentarios Daniel Gomes Lipkin

**0.6.0** 08/2024 Implementação do RabbitMQ e criação do Serviço service e o Projeto service Antonio Eduardo S. Demarchi

**0.8.0** 09/2024 Microserviço de Alunos e implementação do Docker container Daniel Gomes Lipkin

**1.0.0** 09/2024 Finalizado

## **Escopo do Projeto**

### **Objetivo**

- Permitir que o usuário adicione projetos e serviços e veja o custo de cada item e o orçamento total, seguindo boas práticas de programação.
- Permitir que o usuário comente no projeto dos outros para expressar suas opiniões.

### **Situação e justificativa**

Blog para que os alunos postem seus projetos e consigam estabelecer valores para cada serviço individual e obtendo o valor do orçamento total de seu projeto com tópicos para cada serviço. Sem mencionar o feedback que podem receber de outros alunos.

### **Público-Alvo**

Alunos

### **Nomenclatura:**

RF = Requisito Funcional (É o conjunto de entradas, seu comportamento e sua saída, ou seja, envolve cálculos, lógicas de trabalho, manipulação e processamento de dados, entre outros).

RNF = Requisito Não Funcional

[RF-01] Cadastrar um projeto 3 Essencial

[RF-02] Ver os projetos 2 Essencial

[RF-03] Atualizar projeto 3 Essencial

[RF-04] Deletar projeto 1 Essencial

[RF-05] Cadastrar um serviço 3 Essencia

[RF-06] Ver os serviços 2 Essencial

[RF-07] Atualizar serviço 3 Essencial

[RF-08] Deletar um serviço 1 Essencial

[RF-09] Ver detalhes do projeto 3 Essencial

[RF-10] Calcular valor orçamento Total 1 Essencial

RF-C1 Postar Comentario 1 Essencial

RF-C2 Editar Comentario 1 Essencial

RF-C3 Apagar Comentario 1 Essencial

RF-A1 Fazer Login 2 Essencial

RF-A2 Mudar o estado do Projeto 1 Essencial

[RNF-01] Não permitir a modificação de projetos que não forem do aluno 3 Essencial

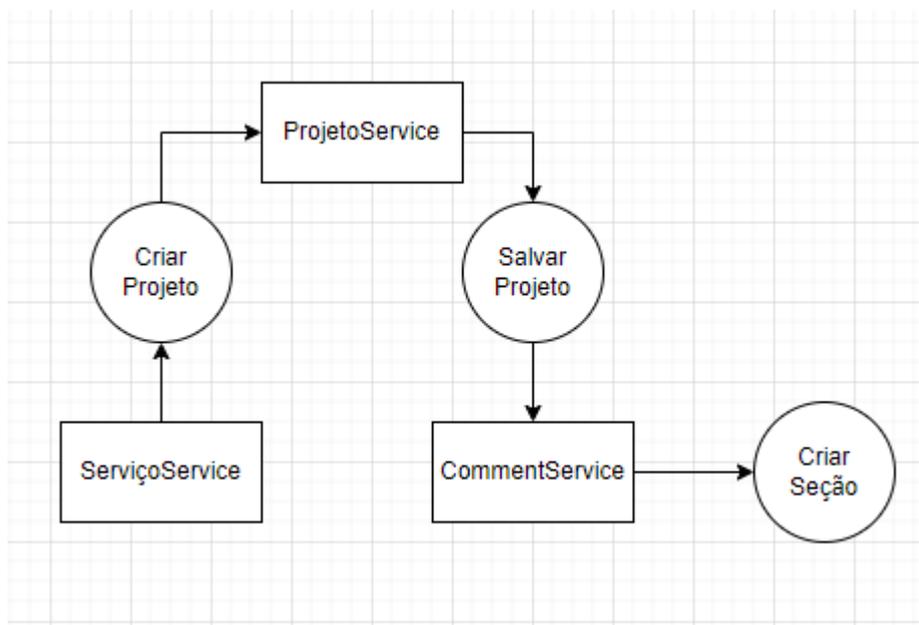
[RNF-02] Senha criptografada 3 Essencial

[RNF-01] Autenticação de Usuario 3 Essencial

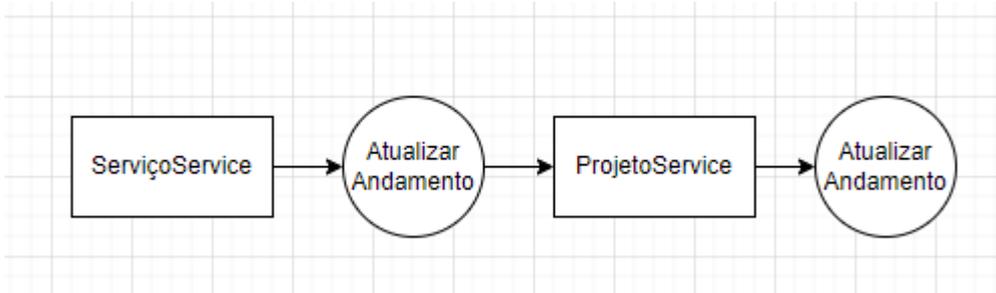
[RNF-C1] Bloquear comentarios com conteudo malicioso 2 Não Essencial

## Fluxo de Eventos

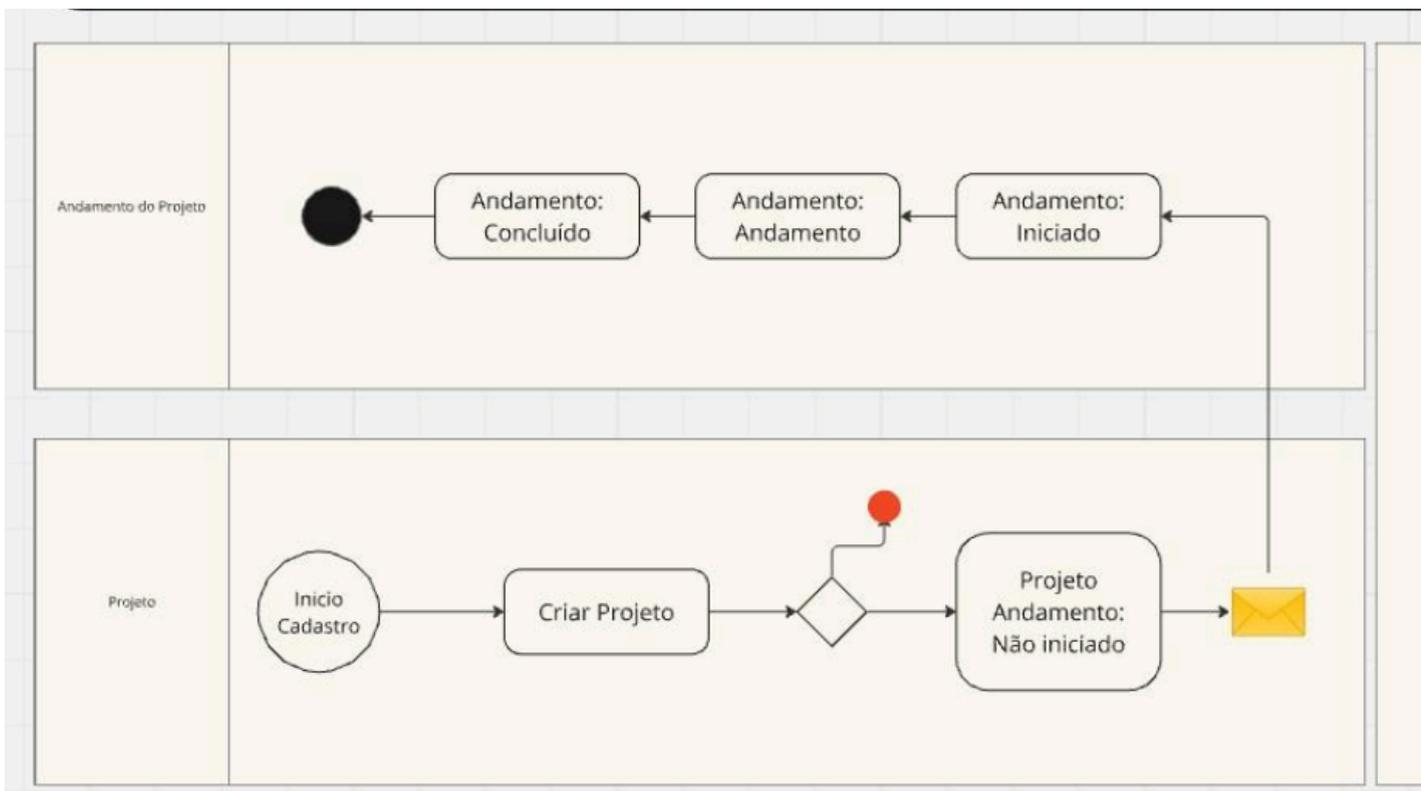
FE1) Criação de Projeto



## FE2) Atualização do Andamento do Projeto



## Diagrama de estados



## Endpoints

### Serviço Service

/main/andamento - POST

Atualiza o Andamento do Projeto (FE2)

/main/ - POST

Cria um projeto novo (FE1)

/main/{id} - DELETE

Apaga um projeto correspondente ao ID fornecido.

/main/{id} - PUT

Atualiza um projeto correspondente ao ID fornecido.

/service/ - GET

Retorna todos Serviços

/service/{id} - GET

Retorna o serviço correspondente ao ID fornecido.

/service/project/{projectId} - GET

Pega todos serviços correspondentes ao projeto com projectId

/service/ - POST

Cria um Serviço novo

/service/{id} - DELETE

Apaga um serviço correspondente ao ID fornecido.

SWAGGER /service/swagger-ui.html

## **Projeto Service**

/project/ - GET

Retorna todos projetos

/project/{id} - GET

Retorna detalhes do projeto com id

/project/ - POST

Salva um projeto novo (FE1)

/project/{id} - DELETE

Apaga um projeto correspondente ao ID fornecido.

/project/{id} - PUT

Atualiza um projeto correspondente ao ID fornecido.

SWAGGER /project/swagger-ui.html

### **Comments Service**

/secoes/{secao\_id} - GET

Retorna a seção de comentarios correspondente ao ID fornecido.

/secoes/ - GET

Retorna uma lista de todas as seções.

/secoes/por\_proj/{projeto\_id} - GET

Retorna a seção correspondente ao ID do projeto fornecido.

/secoes - POST

Cria uma nova seção com base nos dados fornecidos.

/secoes/{secao\_id} - DELETE

Deleta a seção correspondente ao ID fornecido.

/secoes/{secao\_id}/comentarios - POST

Adiciona um comentário à seção correspondente ao ID fornecido.

/comentarios/{comentario\_id} - PUT

Edita o comentário correspondente ao ID fornecido.

/comentarios/{comentario\_id} - DELETE

Deleta o comentário correspondente ao ID fornecido.

SWAGGER /comentarios/swagger-ui.html

### **Alunos Service**

/aluno - GET

Retorna todos alunos

/aluno/nome/{nome} - GET

Retorna o aluno com o nome correspondente

/aluno/id/{id} - GET

Retorna o aluno com o ID correspondente

/aluno/by\_proj/{id} - GET

Retorna o aluno com o projeto de ID correspondente

POST /aluno

Cria um novo aluno.

PUT /aluno/{id}/senha

Atualiza a senha de um aluno.

POST /aluno/{id}/projetos/{projId}

Adiciona um projeto ao aluno com o ID correspondente.

DELETE /aluno/projetos/{projId}

Apaga o projeto com o ID correspondente.

DELETE /aluno/{id}/projetos

Apaga todos os projetos do aluno com o ID correspondente.

POST /aluno/login

Realiza login do aluno com verificação do hash da senha enviada e da senha criptografada na base de dados.

SWAGGER /aluno/swagger-ui.html

## Docker

The screenshot shows the Docker Desktop interface for a user named 'danieldemarchipb'. The top bar includes a back arrow, the Docker logo, the username 'danieldemarchipb', the path 'C:\Users\donke\Documents\Infnet\daniel demarchi PB', and buttons for 'Open', play, stop, and delete. The main area is divided into two columns: a list of containers on the left and a log viewer on the right.

**Container List:**

- danieldemarchipb...**  
[danieldemarchipb-pr](#)  
Running
- danieldemarchipb...**  
[danieldemarchipb-cc](#)  
Running
- danieldemarchipb...**  
[danieldemarchipb-se](#)  
Running
- danieldemarchipb...**  
[rabbitmq:4.0.0-beta.1](#)  
Running  
15672:15672  
[Show all ports \(2\)](#)
- danieldemarchipb...**  
[danieldemarchipb-gc](#)  
Running  
8080:8080
- danieldemarchipb...**  
[danieldemarchipb-el](#)  
Running  
8761:8761

**Log Viewer:**

```
.netflix.discovery.DiscoveryClient - Application version is -1: false
2024-09-09 21:31:51 service-db-1 | 2024-09-10 00:31:51 INFO c
.netflix.discovery.DiscoveryClient - Getting all instance registry
info from the eureka server
2024-09-09 21:31:51 service-db-1 | 2024-09-10 00:31:51 INFO c
.netflix.discovery.DiscoveryClient - The response status is 200
2024-09-09 21:31:51 comments-db-1 | 2024-09-10 00:31:51 INFO c
.netflix.discovery.DiscoveryClient - Disable delta property : false
2024-09-09 21:31:51 comments-db-1 | 2024-09-10 00:31:51 INFO c
.netflix.discovery.DiscoveryClient - Single vip registry refresh pr
operty : null
2024-09-09 21:31:51 comments-db-1 | 2024-09-10 00:31:51 INFO c
.netflix.discovery.DiscoveryClient - Force full registry fetch : fa
lse
2024-09-09 21:31:51 comments-db-1 | 2024-09-10 00:31:51 INFO c
.netflix.discovery.DiscoveryClient - Application is null : false
2024-09-09 21:31:51 comments-db-1 | 2024-09-10 00:31:51 INFO c
.netflix.discovery.DiscoveryClient - Registered Applications size i
s zero : true
2024-09-09 21:31:51 comments-db-1 | 2024-09-10 00:31:51 INFO c
.netflix.discovery.DiscoveryClient - Application version is -1: fal
se
2024-09-09 21:31:51 comments-db-1 | 2024-09-10 00:31:51 INFO c
.netflix.discovery.DiscoveryClient - Getting all instance registry
info from the eureka server
2024-09-09 21:31:51 comments-db-1 | 2024-09-10 00:31:51 INFO c
.netflix.discovery.DiscoveryClient - The response status is 200
2024-09-09 21:32:08 eureka-server-1 | 2024-09-10T00:32:08.048Z I
NFO 1 --- [microEurekaServer] [a-EvictionTimer] c.n.e.registry.Abst
ractInstanceRegistry : Running the evict task with compensationTim
e 0ms
```

Application	AMIs	Zones	Status
COMMENTS	n/a (1)	(1)	UP (1) - <a href="#">comments:80b5d772a9f864405572b3b0b985f50f</a>
GATEWAY-SERVICE	n/a (1)	(1)	UP (1) - <a href="#">gateway-service:761d805ac5c0a301f0c3804c1a408676</a>
PROJETO-SERVICE	n/a (1)	(1)	UP (1) - <a href="#">projeto-service:b43d060e2f16287b2662aee7834b8bf2</a>
SERVICE-SERVICE	n/a (1)	(1)	UP (1) - <a href="#">service-service:5180c4f178969778602c7a006aeca39a</a>

## General Info

Name	Value
total-avail-memory	93mb
num-of-cpus	8
current-memory-usage	43mb (46%)
server-uptime	00:03
registered-replicas	<a href="http://localhost:8761/eureka/">http://localhost:8761/eureka/</a>
unavailable-replicas	<a href="http://localhost:8761/eureka/">http://localhost:8761/eureka/</a> ,
available-replicas	

## Instance Info

Name	Value
ipAddr	172.20.0.2
status	UP

## Diagrama de Classes Atualizado

