

**Fatec Ipiranga**  
**Programação Avançada Orientada a Objetos**  
**Professor Rodrigo Bossini**  
**Especificação do projeto**

Neste projeto, seu grupo atuará no projeto de microsserviços visto em aula.

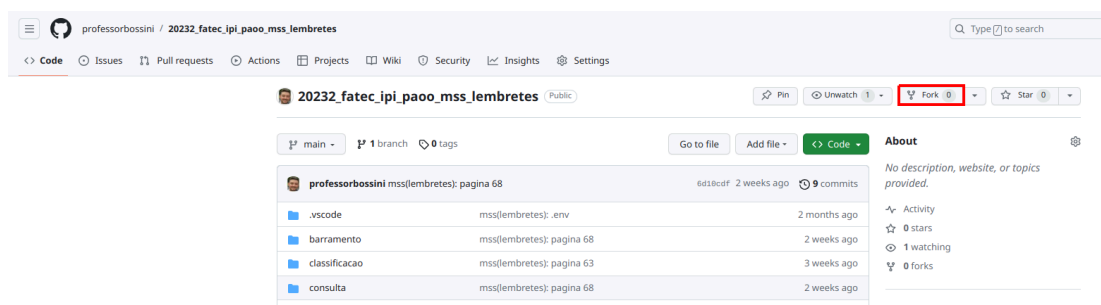
**Instruções**

1. O projeto pode ser desenvolvido por grupos de até quatro alunos.
2. Data de entrega: O projeto será desenvolvido desde o início da aula de 05/06 até o início da aula de 12/06. Os projetos serão apresentados neste dia.
3. A solução deve ser armazenada no Github de um membro do grupo. A entrega do link do repositório será feita pelo Google Forms a seguir.

[https://bit.ly/bossini\\_fatec\\_entregas\\_provas](https://bit.ly/bossini_fatec_entregas_provas)

4. No repositório, inclua um arquivo chamado README.md contendo o nome completo de cada integrante em ordem alfabética e os RAs.
5. O projeto deve ser feito em Javascript com NodeJS.
6. O projeto deve fazer uso do mecanismo **dotenv** e incluir os arquivos **.env** (fora do controle de versão) e **.env.example** (incluído no controle de versão).
7. Faça um **fork** do repositório a seguir

[https://github.com/professorbossini/20232\\_fatec\\_ipi\\_paoo\\_mss lembretes](https://github.com/professorbossini/20232_fatec_ipi_paoo_mss lembretes)



Basta que um membro do grupo faça o fork. A seguir, crie uma pasta local no seu computador. Por exemplo.

```
C:\Users\aluno\Documents\paoo\p2
```

Abra o VS Code e vincule-o à pasta com File >> Open Folder. Clique Terminal >> New Terminal para abrir um terminal interno do VS Code. Faça clone com

```
git clone link_do_seu_repositorio .
```

Observe que você deve usar o link do seu repositório, depois de ter feito o fork.

### ***Requisitos***

#### **1 Ajuste no barramento de eventos**

Faça com que o barramento de barramentos repasse os eventos apenas para que microserviços interessados neles os recebam, da seguinte forma:

mss de lembrete: não recebe evento algum

mss de observações: recebe evento ObservacaoClassificada

mss de consulta: recebe todos os eventos, exceto ObservacaoClassificada

mss de classificação: recebe o evento ObservacaoCriada

feat(barramento): deixa de fazer broadcast

#### **2 Classificação de lembretes**

Ajuste o microserviço de classificação de modo que os lembretes também sejam classificados. Um lembrete é considerado urgente se ele contém a palavra "urgente". Ele é considerado importante se contém a palavra "importante". Caso não possua nenhuma dessas palavras, ele é considerado 'comum'.

mensagem de commit: feat(classificacao): lembretes classificados

No final, crie uma tag anotada, marcando o último commit como importante.