

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

Centro de Ciências Exatas e Tecnologia

Departamento de Computação

Bacharelado em Ciência da Computação

**DevOps**

Manual T2

**Professor:**

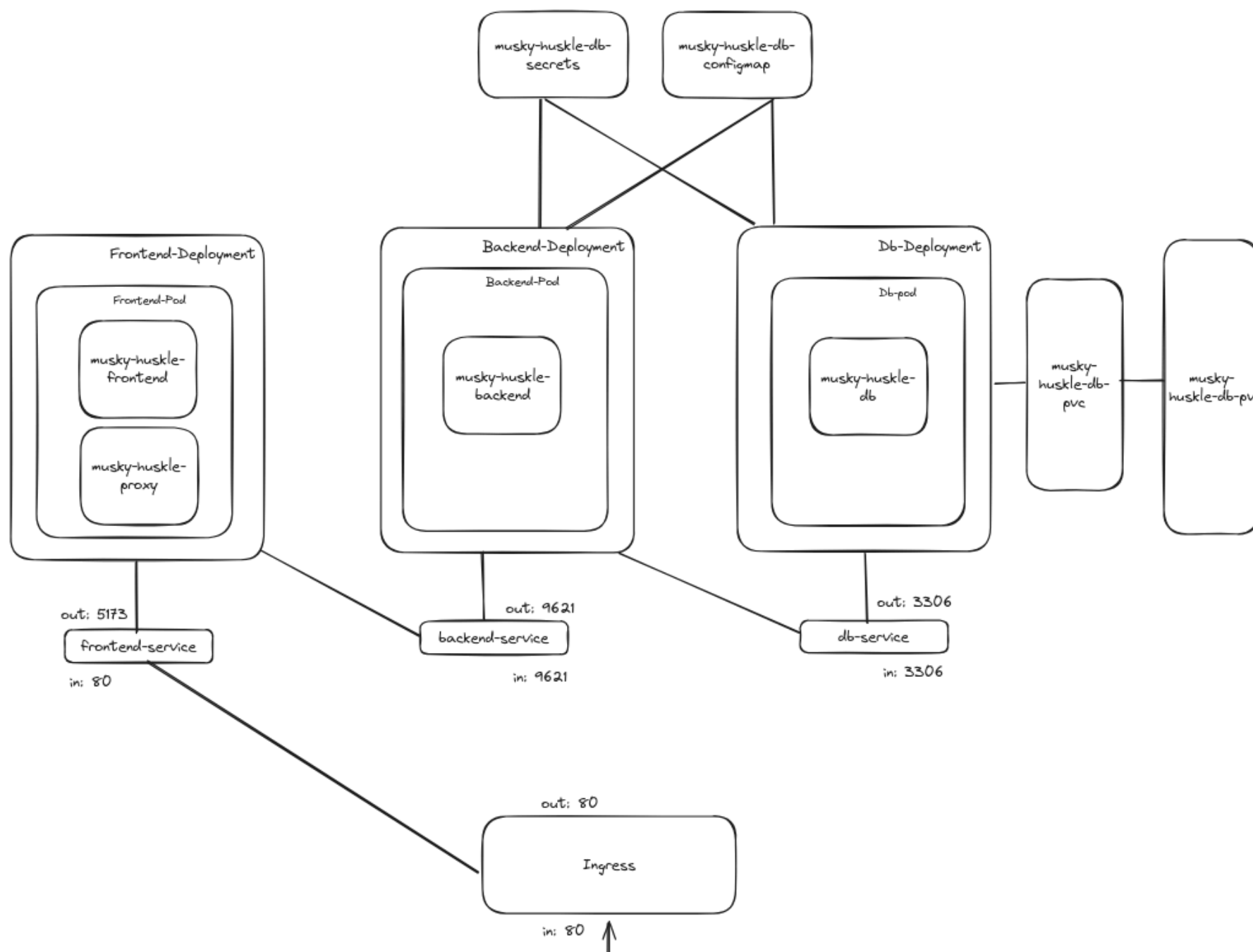
Delano Beder

**Integrantes da equipe:**

Daniel Kenichi Tiago Tateishi - RA: 790837

Rodrigo Pavão Coffani Nunes - RA: 800345

## 1. Arquitetura do projeto



## 2. Sobre a aplicação

O muskyhuskle é uma aplicação de um jogo simples baseado em jogos como Loldle, Wordle e Termooo. A aplicação foi criada para o aprendizado prático da utilização de tecnologias como docker e gRPC. O jogo é composto de 4 componentes:

1. Uma interface criada em react;
2. Um backend em gRPC responsável pela lógica do jogo e pelo CRUD de membros,;
3. Um banco de dados MySQL; e
4. Um Envoy Proxy para encaminhar chamadas HTTP 1.1 do frontend para o backend em gRPC (que apenas aceita chamadas HTTP 2).

## 3. Iniciando a aplicação

### Clonando o repositório

- Comando: `git clone https://github.com/parmenashp/musky-huskle.git`
- Entre na branch Kubernetes: `git checkout Kubernetes`

**Alterar o /etc/hosts:**

- Comando: `sudo nano /etc/hosts`
- O endereço IP deve ser aquele obtido com o comando `minikube ip`
- Adicionar no arquivo:

```
# Minikube
192.168.49.2    k8s.local
192.168.49.2    k8s.back
```

**Subindo o ambiente e buildando imagens:**

- Entre na pasta "Kubernetes": `cd Kubernetes`
- Suba o ambiente: `./minikube-up.sh -b`
- A tag -b avisa que é necessário buildar as imagens do front e backend.
- Resultado esperado:

Workload Status

Running: 3

Deployments

Running: 3

Pods

Running: 3

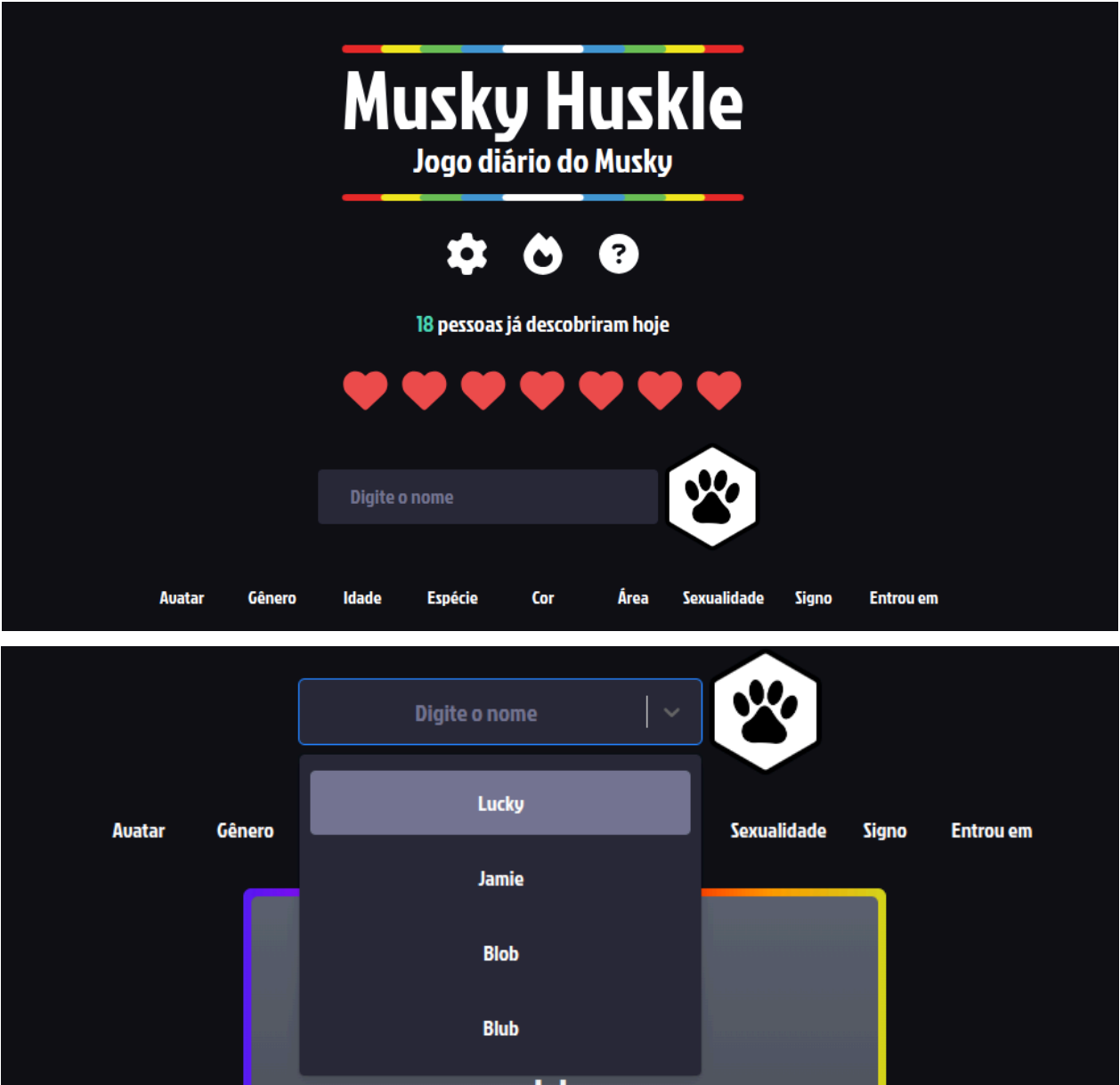
Replica Sets

Deployments

Name	Images	Labels	Pods	Created ↑
musky-huskle-frontend	musky-huskle_frontend envoyproxy/envoy:v1.30-latest	app: musky-huskle-frontend	1 / 1	2 hours ago
musky-huskle-db	mysql:latest	app: musky-huskle-db	1 / 1	2 hours ago
musky-huskle-server	musky-huskle_backend	app: musky-huskle-server	1 / 1	2 hours ago

**Acessando o frontend:**

- Acesse o frontend da aplicação pelo endereço `k8s.local`



Testando interação entre os containers:

- Com o frontend aberto, comece um jogo: selecione uma das opções disponíveis e clique para enviar;
- As requests (quando o campo de nome é enviado) são enviadas para o proxy;
- O proxy realiza a tradução da request de HTTP 1.1 para HTTP 2 (aceito pelo gRPC);
- A request então é enviada para o backend, onde os dados são consultados no banco de dados;
- A resposta gerada é então recebida pelo proxy e utilizada no frontend;
- Uma nova linha é gerada, apresentando de forma visual o resultado da requisição (acerto, quase e erro);

