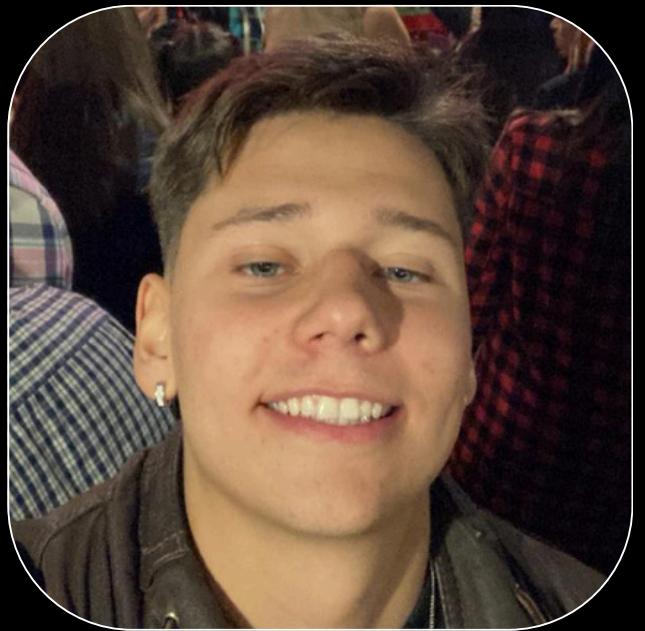




MONITORAMENTO E OBSERVABILIDADE EM CLUSTER KUBERNETES

Trabalho final - PSPD 2025.2

GRUPO



Guilherme Westphall
211061805



Lucas Martins
221022088

REVISÃO GRPC

Serviços

Aplicação bancária

P

- Stub
- Camada de tradução:
JSON → Protobuf
- Recebe as requisições e
redireciona para A e B
- Python

A

- Serviço do cliente
- CRUD de clientes
- Comunicação com o
Postgres via Prisma
(ORM)
- Node.js

B

- Serviço de contas
- CRUD de contas e
transações
- Comunicação com o
Postgres via Prisma
(ORM)
- Node.js

OBJETIVOS

Montagem do cluster multinode

- 1 Master + 2 Workers
- 4 Serviços (Stub, Account, Client, Postgres)
- Avaliar o impacto de réplicas e workers
 - Throughput médio
 - Latência (média e P95)
 - Taxa de erros

AMBIENTE E DEPLOY

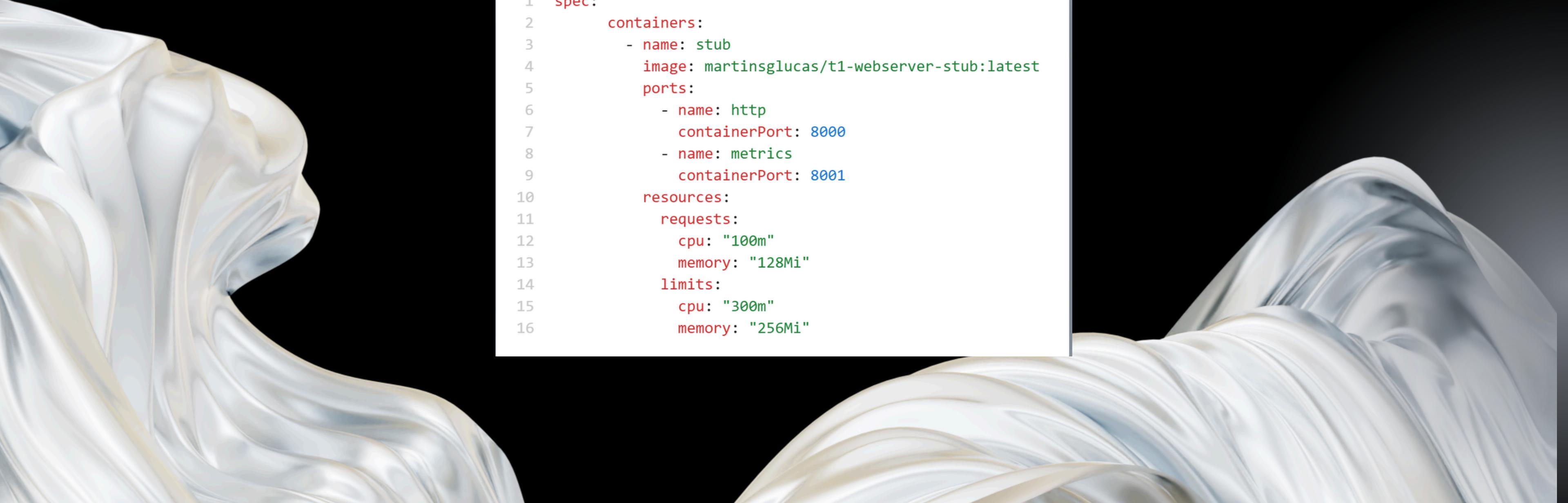
- Kubectl: ferramenta de linha de comando para interagir com o cluster
- Kind (Kubernetes in Docker): alternativa diante da dificuldade para montar cluster real
- Helm: gerenciador de pacotes
- Prometheus: ferramenta de monitoramento
 - prometheus vs kube-prometheus-stack
- Metrics-server: ferramenta para métricas de CPU e memória

AMBIENTE E DEPLOY



```
1 kind: Cluster
2 apiVersion: kind.x-k8s.io/v1alpha4
3 nodes:
4   - role: control-plane
5     extraPortMappings:
6       - containerPort: 30080
7         hostPort: 30080
8         protocol: TCP
9   - role: worker
10  - role: worker
```

MODIFICAÇÃO DOS MANIFESTS



```
1 spec:
2   containers:
3     - name: stub
4       image: martinsglucas/t1-webserver-stub:latest
5       ports:
6         - name: http
7           containerPort: 8000
8         - name: metrics
9           containerPort: 8001
10      resources:
11        requests:
12          cpu: "100m"
13          memory: "128Mi"
14        limits:
15          cpu: "300m"
16          memory: "256Mi"
```

CRIAÇÃO YAMLS HPA

```
 1  apiVersion: autoscaling/v2
 2  kind: HorizontalPodAutoscaler
 3  metadata:
 4    name: stub-hpa-cpu
 5    namespace: t3
 6  spec:
 7    scaleTargetRef:
 8      apiVersion: apps/v1
 9      kind: Deployment
10      name: stub
11    minReplicas: 1
12    maxReplicas: 3
13    metrics:
14      - type: Resource
15        resource:
16          name: cpu
17        target:
18          type: Utilization
19          averageUtilization: 50
20
```

TESTES

Cenários 3/6

Baseline

- Configuração: 1 master, 2 workers; stub, client, account e postgres com 1 réplica cada.
- Objetivo: estabelecer um cenário de referência funcional, com carga estável e sem otimizações de escalonamento.

Réplica Stub

- Configuração: 1 master, 2 workers; **3 réplicas do stub**, demais serviços com 1 réplica.
- Objetivo: verificar o impacto de escalar apenas a borda HTTP (API Gateway) na capacidade de atendimento

Réplicas Serviços

- Configuração: 1 master, 2 workers; **3 réplicas para stub, client e account, postgres com 1 réplica**.
- Objetivo: verificar o impacto de escalar apenas a borda HTTP (API Gateway) na capacidade de atendimento

TESTES

Cenários 6/6

4 Workers

- Configuração: 1 master, 4 workers; stub, client, account e postgres com 1 réplica cada.
- Objetivo: avaliar se apenas aumentar o número de nós do cluster, sem mudar réplicas, traz benefício de desempenho.

4 Workers + 3rep

- Configuração: 1 master, 4 workers; 3 réplicas para stub, client e account, postgres com 1 réplica.
- Objetivo: combinar mais nós e mais réplicas de aplicação para buscar um cenário de maior throughput mantendo estabilidade.

Autoscaling (HPA)

- Configuração: 1 master, 2 workers; HPA configurado para stub, client e account, permitindo variar automaticamente o número de réplicas (1-N) conforme uso de CPU.
- Objetivo: avaliar se o escalonamento automático do Kubernetes consegue reagir à carga de forma mais eficiente que as configurações manuais de réplicas.

TESTES

RESULTADOS

Cenário	Workers / Réplicas	Throughput (req/s)	Latência	p95	http_req_failed	checks_failed
Baseline	2w - 1x stub, 1x account, 1x client	301	231 ms	711 ms	0%	0%
Replica Stub	2w - 3x stub, 1x account, 1x client	670	48 ms	133 ms	10,14%	12,62%
Replica 3xServiços	2w - 3x stub, 3x account, 3x client	670	48 ms	142 ms	10,14%	≈12%
4 Workers	4w - 1x stub, 1x account, 1x client	199	396 ms	1,37 s	0%	0%
4 Workers + 3xServiços	4w - 3x stub, 3x account, 3x client	333,77	195,92 ms	611,58 ms	0%	0%
Autoscaling (HPA Stub)	2w - HPA stub (1-N), 1x account, 1x client	125,31	770 ms	2,92 s	1,47%	1,52%

PROMETHEUS

- O stub-service expõe as métricas através de um endpoint `/metrics`

QUERIES

THROUGHPUT (req/s)

```
sum(  
  rate(http_requests_total{  
    namespace="t3",  
    service="stub-service",  
  }[15m])  
)
```

**THROUGHPUT p/ endpoint
(req/s)**

```
sum by (endpoint)(  
  rate(http_requests_total{  
    namespace="t3",  
    service="stub-service"  
  }[15m])  
)
```

Latência (ms)

```
sum(  
  rate(http_request_duration_seconds_sum{  
    namespace="t3",  
    service="stub-service"  
  }[15m])  
) /  
sum(  
  rate(http_request_duration_seconds_count{  
    namespace="t3",  
    service="stub-service"  
  }[15m])  
)
```

PROMETHEUS

QUERIES

P95 (ms)

```
histogram_quantile(  
    0.95,  
    sum by (le)(  
        rate(http_request_duration_seconds_bucket{  
            namespace="t3",  
            service="stub-service"  
        }[15m])  
    )  
)
```

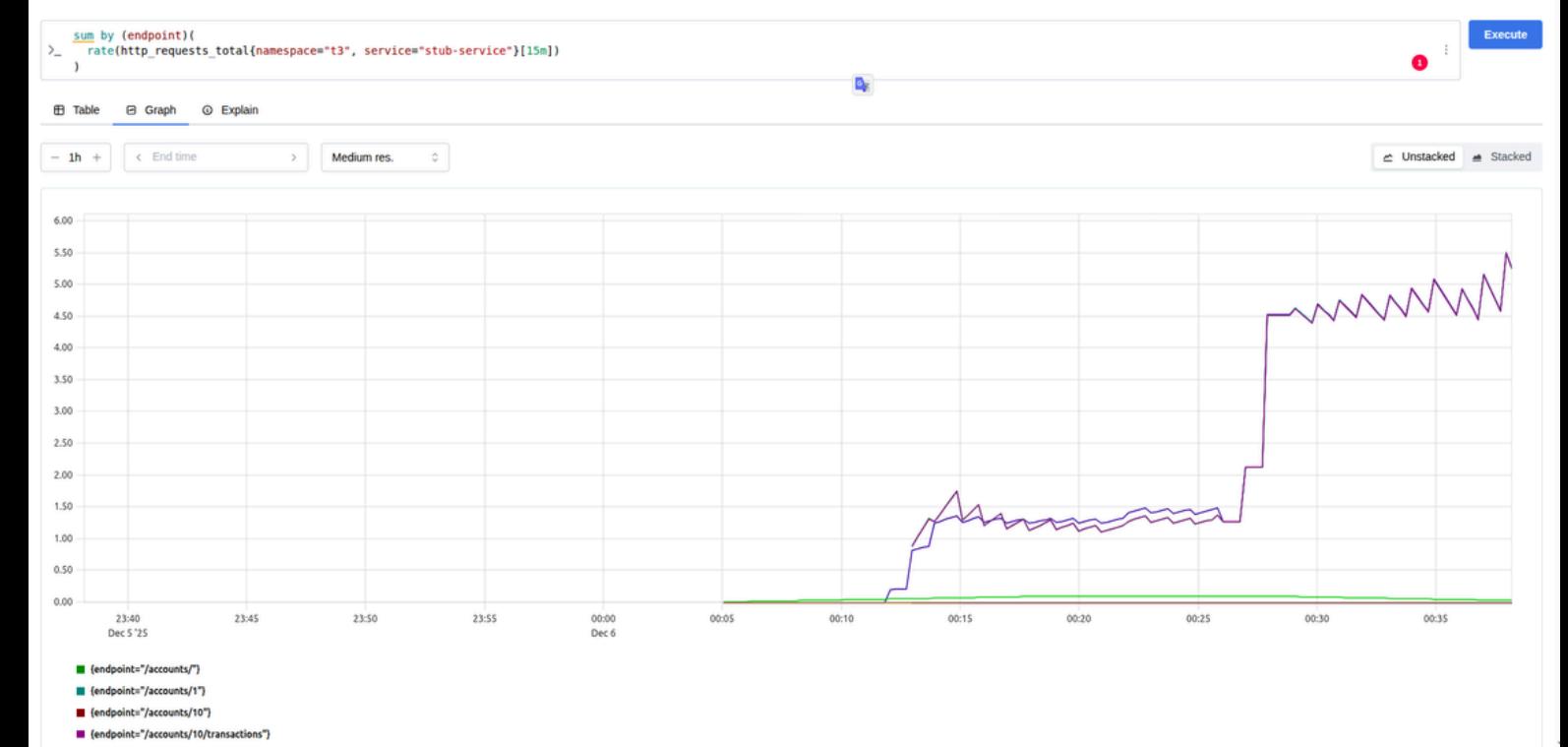
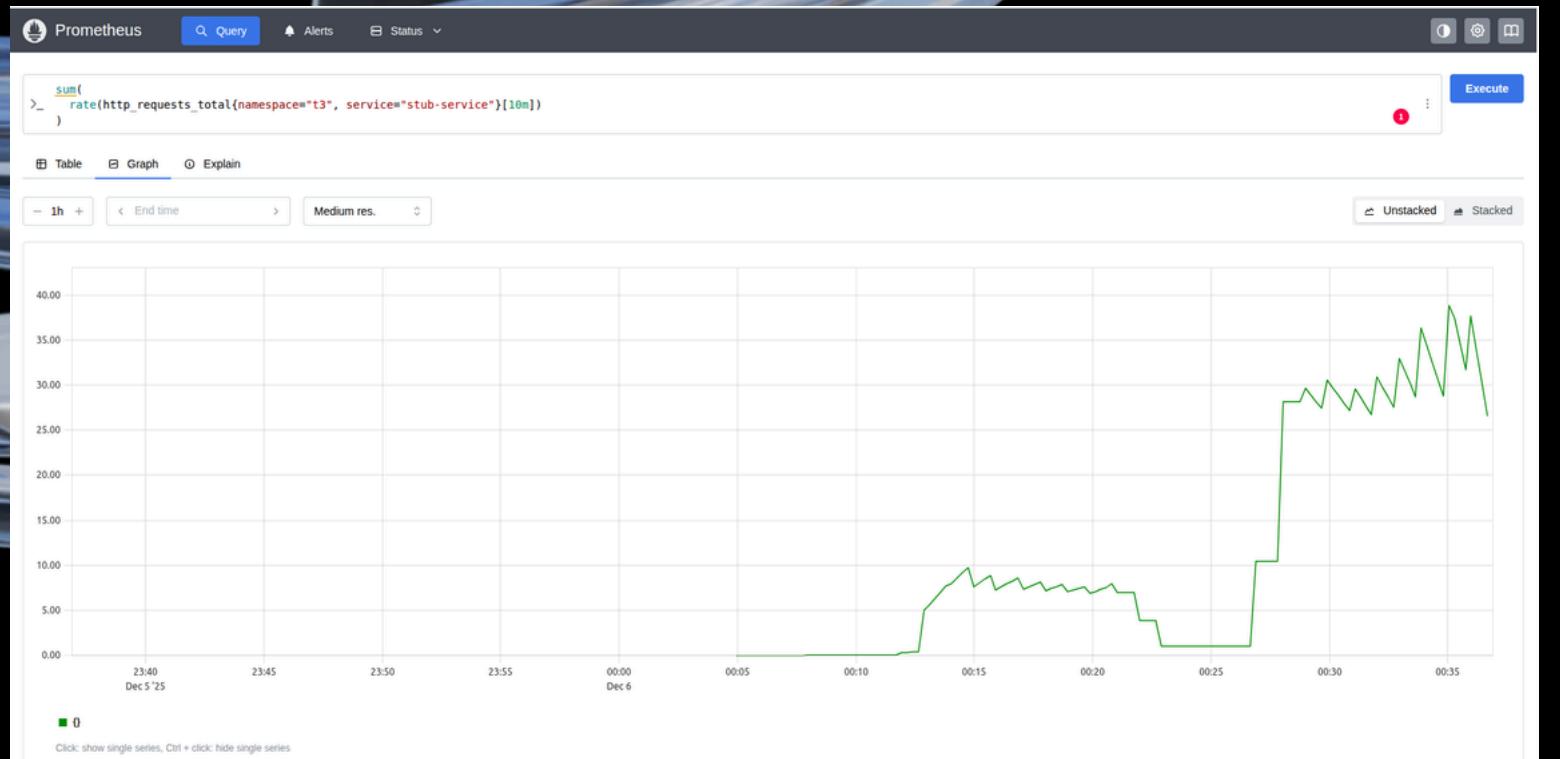
CPU (cores de CPU)

```
sum by (pod)(  
    rate(container_cpu_usage_seconds_total{  
        namespace="t3",  
        pod=~"stub-.*"  
    }[5m])  
)
```

Memória (bytes)

```
sum by (pod)(  
    container_memory_usage_bytes{  
        namespace="t3",  
        pod=~"stub-.*"  
    }  
)
```

PROMETHEUS



MONITORAMENTO HPA

The image shows a Windows desktop with four terminal windows open, illustrating the monitoring and management of Horizontal Pod Autoscaling (HPA) in a Kubernetes cluster.

- Top Left Terminal:** Displays the command `kubectl get hpa -n t3 -w`. The output shows three HPA resources: `accountserver-hpa`, `clientserver-hpa`, and `stub-hpa-cpu`, each with its reference, target CPU usage, and current settings.
- Top Right Terminal:** Displays the command `kubectl get pods -n t3 -w`. The output lists the pods in the `t3` namespace, including their names, readiness, status, restart counts, and creation age.
- Bottom Left Terminal:** Shows the output of a k6 load testing script. It includes configuration details like execution type (`local`), script name (`k6-autoscaling.js`), and output file (`-`). It also specifies test scenarios: one scenario with 200 max VUs over 11m30s. A note indicates a default behavior of looping VUs for 11m0s over 6 stages.
- Bottom Right Terminal:** Displays the command `Every 2.0s: kubectl top pod -n t3 --containers`. The output shows the CPU and memory usage for each pod in the `t3` namespace at 17:05:02 on Saturday, December 6, 2025.

The taskbar at the bottom of the screen shows standard icons for search, file, and system controls, along with the date and time (06/12/2025, 17:05).

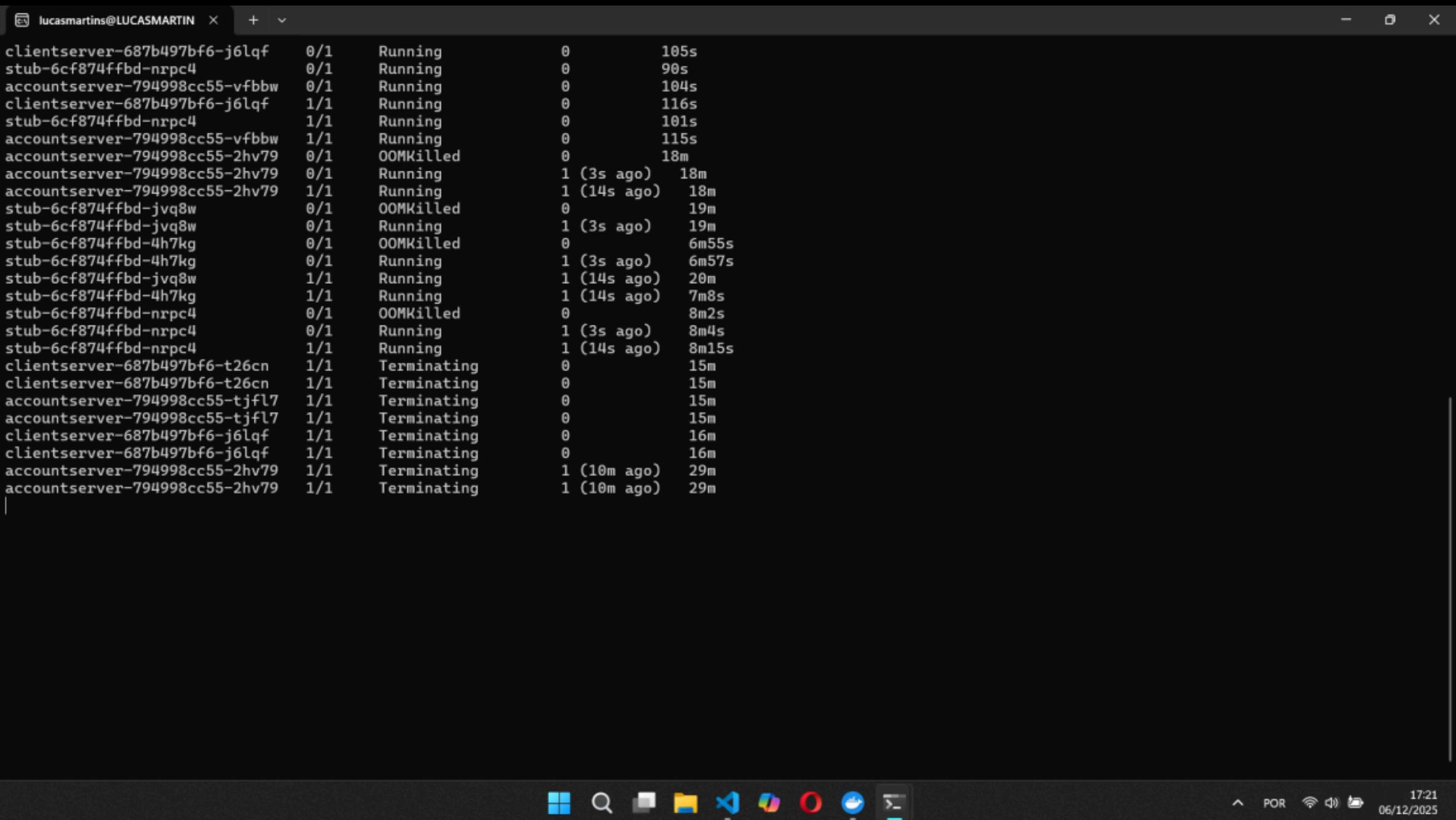
MONITORAMENTO HPA

The image shows a Windows desktop environment with four terminal windows open, illustrating the monitoring of Horizontal Pod Autoscaler (HPA) performance across different components.

- Top Left Terminal:** Displays CPU usage for various pods. For example, "accountserver-hpa" has a CPU usage of 91%/50, while "stub-hpa-cpu" has 289%/5. The output includes deployment counts (e.g., 1/3), resource requests (e.g., 3 66m), and current metrics (e.g., 3 16m).
- Top Right Terminal:** Shows a list of pods and their current status. It includes columns for name, status (e.g., Running, PodInitializing), and duration (e.g., 10s, 15s, 17s, 19s, 30s, 55s, 58s, 94s, 101s, 105s, 90s, 104s, 116s, 101s, 115s, 18m, 18m, 18m, 18m).
- Bottom Left Terminal:** Provides k6 test feedback. It mentions a warning about generating 200022 unique time series, which is higher than the limit of 100000. It also indicates running 172/200 VUs, 4643 complete iterations, and 0 interrupted iterations. The progress bar shows 172/200 VUs at 06m21.6s/11m00.0s.
- Bottom Right Terminal:** Runs a command to monitor pod metrics every 2.0s. The output shows the following table:

POD	NAME	CPU(cores)	MEMORY(bytes)
accountserver-794998cc55-tjfl7	accountserver	23m	25Mi
accountserver-794998cc55-vfbbw	accountserver	1m	19Mi
clientserver-687b497bf6-j6lqf	clientserver	71m	45Mi
clientserver-687b497bf6-r6qtp	clientserver	131m	67Mi
clientserver-687b497bf6-t26cn	clientserver	19m	22Mi
postgres-6fd886f58b-26c2k	postgres	108m	86Mi
stub-6cf874ffbd-4h7kg	stub	273m	169Mi
stub-6cf874ffbd-jvq8w	stub	287m	192Mi
stub-6cf874ffbd-nrpc4	stub	270m	140Mi

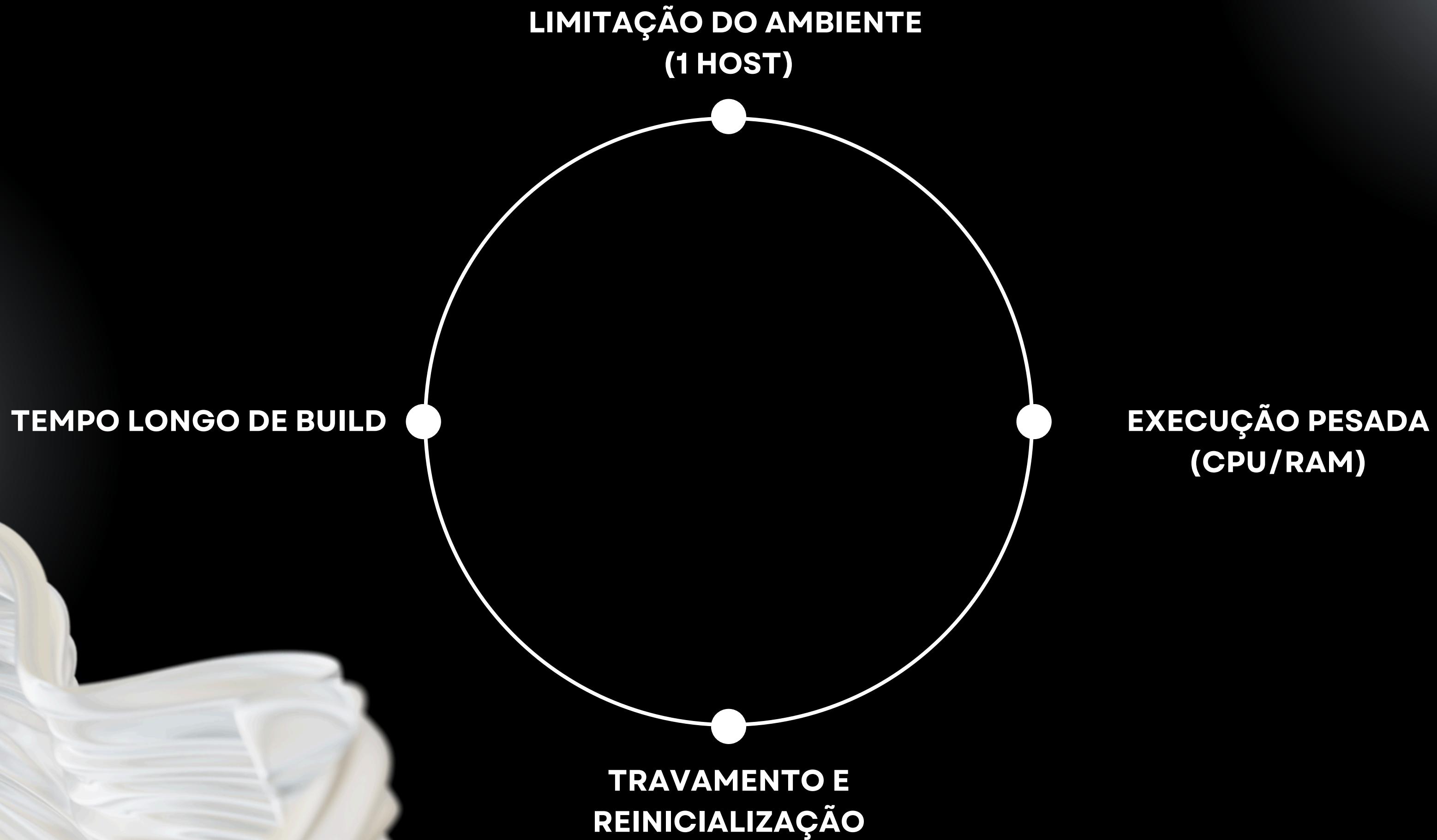
MONITORAMENTO HPA



```
lucasmartins@LUCASMARTIN ~ + ~
clientserver-687b497bf6-j6lqf  0/1   Running      0      105s
stub-6cf874ffbd-nrpc4        0/1   Running      0      90s
accountserver-794998cc55-vfbbw 0/1   Running      0      104s
clientserver-687b497bf6-j6lqf  1/1   Running      0      116s
stub-6cf874ffbd-nrpc4        1/1   Running      0      101s
accountserver-794998cc55-vfbbw 1/1   Running      0      115s
accountserver-794998cc55-2hv79 0/1   OOMKilled    0      18m
accountserver-794998cc55-2hv79 0/1   Running      1 (3s ago) 18m
accountserver-794998cc55-2hv79 1/1   Running      1 (14s ago) 18m
stub-6cf874ffbd-jvq8w         0/1   OOMKilled    0      19m
stub-6cf874ffbd-jvq8w         0/1   Running      1 (3s ago) 19m
stub-6cf874ffbd-4h7kg         0/1   OOMKilled    0      6m55s
stub-6cf874ffbd-4h7kg         0/1   Running      1 (3s ago) 6m57s
stub-6cf874ffbd-jvq8w         1/1   Running      1 (14s ago) 20m
stub-6cf874ffbd-4h7kg         1/1   Running      1 (14s ago) 7m8s
stub-6cf874ffbd-nrpc4        0/1   OOMKilled    0      8m2s
stub-6cf874ffbd-nrpc4        0/1   Running      1 (3s ago) 8m4s
stub-6cf874ffbd-nrpc4        1/1   Running      1 (14s ago) 8m15s
clientserver-687b497bf6-t26cn 1/1   Terminating  0      15m
clientserver-687b497bf6-t26cn 1/1   Terminating  0      15m
accountserver-794998cc55-tjfl7 1/1   Terminating  0      15m
accountserver-794998cc55-tjfl7 1/1   Terminating  0      15m
clientserver-687b497bf6-j6lqf  1/1   Terminating  0      16m
clientserver-687b497bf6-j6lqf  1/1   Terminating  0      16m
accountserver-794998cc55-2hv79 1/1   Terminating  1 (10m ago) 29m
accountserver-794998cc55-2hv79 1/1   Terminating  1 (10m ago) 29m
```

The screenshot shows a terminal window titled "lucasmartins@LUCASMARTIN ~ + ~" displaying a list of Kubernetes pods. The table includes columns for pod name, replicas, status, last seen error, and creation time. Most pods are in a "Running" state, while some are "OOMKilled" or "Terminating". The terminal is running on a Windows 10 desktop, as indicated by the taskbar icons at the bottom.

DIFICULDADES





OBRIGADO