



Universidade do Minho

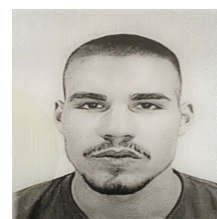
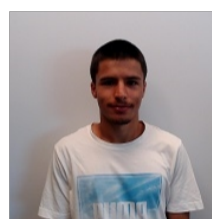
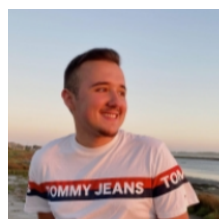
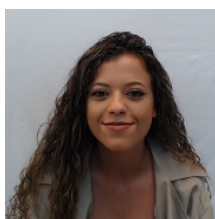
UMinho

# Mestrado Engenharia Informática

## Aplicações de Serviços em Computação em Nuvem (2022/23)

Grupo: 33

pg50633 Mariana Rocha Marques  
pg50229 António Luís de Macedo Fernandes  
pg50483 João Paulo Sousa Mendes  
pg50518 José Diogo Martins Vieira  
pg50499 João Silva Torres



Repositório: (<https://github.com/joaopsmendes/ascn-ghost>).  
Braga, 9 de agosto de 2023

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Ghost</b>	<b>3</b>
<b>3</b>	<b>Arquitetura e Componentes</b>	<b>4</b>
<b>4</b>	<b>Ferramentas para a instalação e configuração automática da aplicação.</b>	<b>6</b>
4.1	Ansible . . . . .	6
4.2	Kubernetes Deployment . . . . .	6
4.2.1	Deployment do mysql . . . . .	6
4.2.2	Deployment do Ghost . . . . .	7
<b>5</b>	<b>Ferramentas de monitorização, métricas e visualização escolhidas, justificando a sua escolha</b>	<b>8</b>
5.1	DashBoard CPU . . . . .	8
5.2	DashBoard RAM . . . . .	8
5.3	DashBoard I/O . . . . .	9
<b>6</b>	<b>Ferramentas de avaliação e testes desenvolvidos.</b>	<b>10</b>
6.1	Teste 1- Conexão à pagina principal . . . . .	10
6.2	Teste 2- Sign In . . . . .	10
6.3	Teste 3- Subscribe . . . . .	10
<b>7</b>	<b>Monitorização durante os Testes</b>	<b>12</b>
<b>8</b>	<b>Conclusão</b>	<b>13</b>

# 1. Introdução

No âmbito do desenvolvimento do projeto da Unidade Curricular de Aplicações e Serviços de Computação em Nuvem, foi-nos proposto desenvolver a automatização do processo de instalação, configuração, monitorização e avaliação da aplicação Ghost.

Assim sendo, o presente relatório pretende expor todo o trabalho realizado, de forma detalhada, para a perceção das etapas efetuadas na realização e justificação das decisões tomadas enquanto grupo.

Numa fase inicial, apresentamos a arquitetura e os componentes do sistema. De seguida, iremos identificar as ferramentas e abordagem utilizadas para a instalação e configuração automática da aplicação. Por fim, as últimas fases dizem respeito às ferramentas de monitorização e avaliação onde justificamos as nossas escolhas. Também faremos um análise dos resultados da avaliação experimental.

Na última secção deste relatório faremos uma reflexão final sobre todo o trabalho desenvolvido pelo grupo, apontando os principais pontos fortes e os pontos a melhorar.

## 2. Ghost

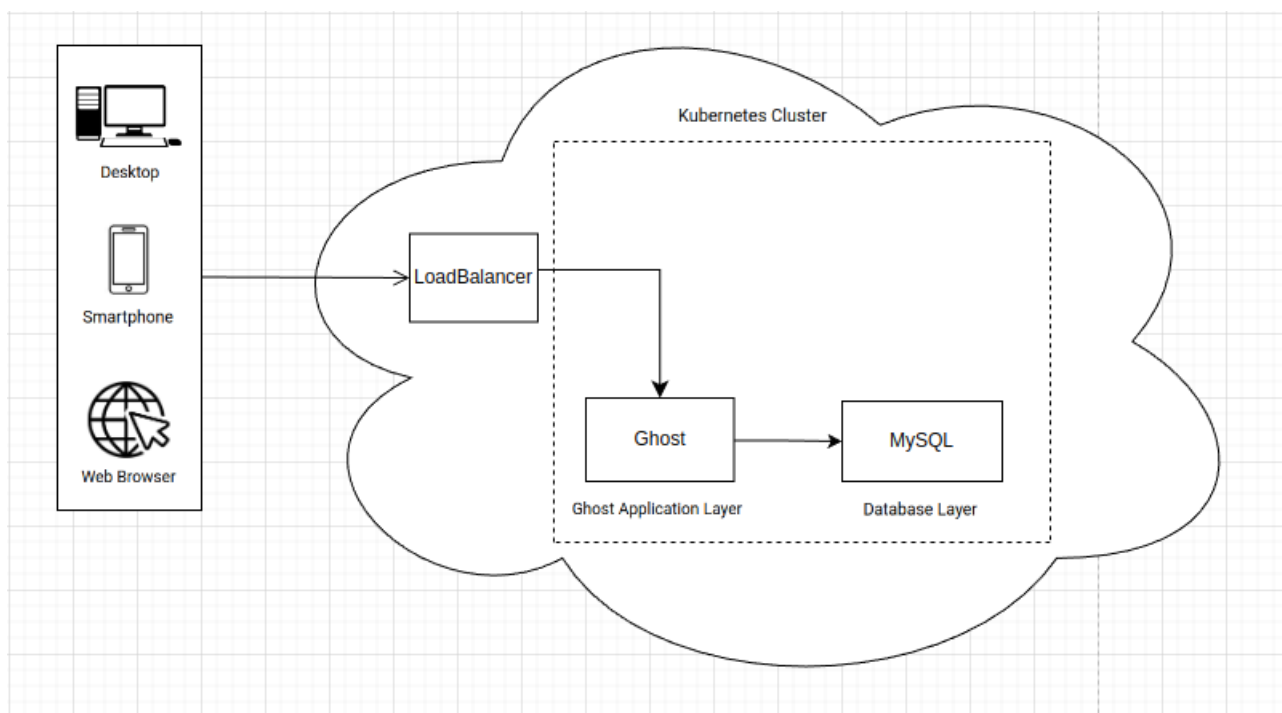
O Ghost é uma aplicação para criadores de novos conteúdos publicarem, partilharem e expandirem os seus negócios de acordo com o seu conteúdo. Além disso, possui ferramentas modernas para construir um site, publicar conteúdo, enviar *newsletters* e oferecer assinaturas pagas aos membros.



**Figura 2.1:** Logo Ghost

### 3. Arquitetura e Componentes

Para a elaboração da arquitetura do sistema baseamos-nos no padrão multi camada, uma vez que possui a camada de interface para os utilizadores, a camada de serviço da aplicação e, por fim, uma camada responsável pela persistência de dados. Com isto, através da arquitetura é possível a separação dos diversos componentes e suas respectivas funcionalidades, ainda que as mesmas consigam comunicar e propagar o seu estado entre elas.



**Figura 3.1:** Arquitetura Implementada

Assim sendo, com a arquitetura criada, foi mais fácil elaborar a estrutura do projeto, que se segue em diante:

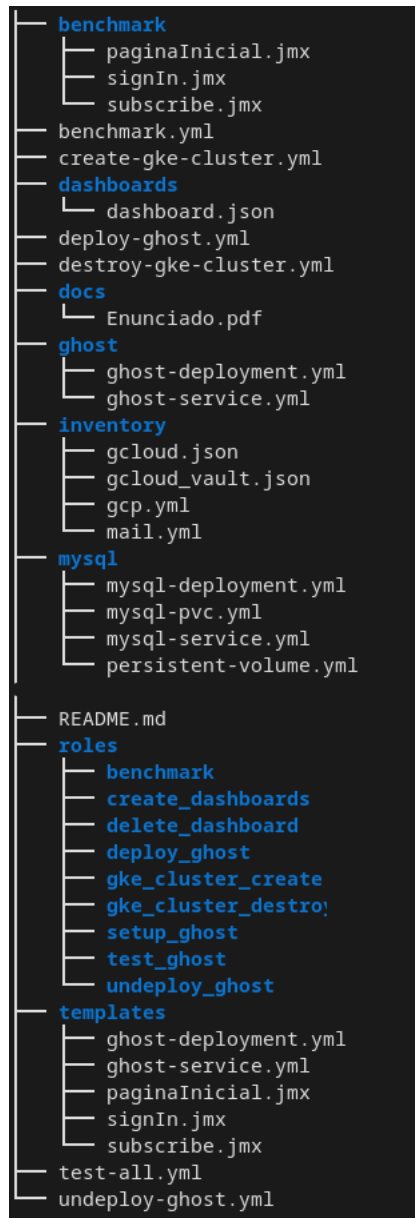


Figura 3.2: Estrutura do Projeto

## 4. Ferramentas para a instalação e configuração automática da aplicação.

A ferramenta **Ansible** possibilitou a instalação e configuração automática da aplicação Ghost no serviço **Google Kubernetes Engine (GKE)** da Google Cloud. Este objetivo é apresentado na *Tarefa Base* e é acompanhado por uma lista de requisitos, dos quais foram todos cumpridos.

### 4.1 Ansible

Foram definidos 5 playbooks:

- **create-gke-cluster.yml** e **destroy-gke-cluster.yml**. Responsável pela criação e destruição do cluster Kubernetes no serviço GKE, tendo apenas um role associado a cada playbook, **gke\_cluster\_create** e **gke\_cluster\_destroy**, respetivamente.
- **deploy-ghost.yml** - Instala, configura e executa, no cluster todos as componentes da aplicação Ghost. Este playbook irá executar três roles, pela seguinte ordem:
  - **setup\_ghost** - Responsável por atualizar os templates com as variáveis dinâmicas ou credenciais, que não podem estar nestes ficheiros.
  - **deploy\_ghost** - Responsável pela aplicação de todos os ficheiros necessários para a componente da base de dados e da aplicação no Kubernetes. Assim como a atualização das credenciais do admin. Este role termina, quando a o url da aplicação possa ser acedido.
  - **create\_dashboards** - Cria as dashboards necessárias para a monitorização das várias componentes do cluster.
- **undeploy\_ghost** - Termina a execução da instalação atual da aplicação Ghost. Removendo todos os ficheiros que foram atualizados aquando do **setup\_ghost**. Quando for especificada a flag (-e delete\_data=true), irá se proceder a remoção de todos os dados presentes na base de dados. Também é responsável pela remoção das dashboards anteriormente criadas.
- **test\_all** - Efetua um conjunto de testes automáticos para validar a execução do Ghost.

### 4.2 Kubernetes Deployment

Para os ficheiros de kubernetes, relativos às duas componentes:

#### 4.2.1 Deployment do mysql

Para o deployment da base de dados foram utilizados 4 ficheiros:

- **persistent-volume.yml** - Este ficheiro irá criar o volume de dados que irá ser usado para a persistência dos dados. Atribuímos 10Gi.

- **mysql-deployment** - Este ficheiro irá fazer a implementação da base de dados mysql, que vai ser utilizada pela aplicação *ghost* através da porta 3306. Aqui será criado o utilizador e a sua password, como o nome da BD. Irá também referenciar o volume de dados a utilizar.
- **mysql-pvc.yml** - Este ficheiro irá fazer o *claim* do volume de dados criado.
- **mysql-service.yml** - Este ficheiro irá criar o serviço do tipo ClusterIP, para apenas poder ser acedido pelos restantes nodos do cluster. Irá também referenciar a porta na qual este serviço está disponível, 3306.

#### 4.2.2 Deployment do Ghost

Para o deployment do ghost teremos 2 ficheiros:

- **ghost-deployment.yml** - Neste ficheiro encontra-se toda a informação necessária para instalação do ghost. Essa informação será relativa à base de dados que irá utilizar, ao url onde ficará disponível e às credenciais do serviço de email.
- **ghost-service.yml** - Este ficheiro irá criar o serviço do tipo LoadBalancer, este permite que a aplicação possa ser acedida pelo exterior e não apenas pelos nodos do cluster, através da porta 2368.

Todas as variáveis dinâmicas ou credenciais necessárias para estes ficheiros, irão ser devidamente atribuídas antes destes serem aplicados.



## 5. Ferramentas de monitorização, métricas e visualização escolhidas, justificando a sua escolha

Na realização deste trabalho prático, utilizámos a API da Google Cloud de forma a monitorizar e visualizar o comportamento da utilização de recursos na nossa aplicação. As métricas utilizadas nesta monitorização são referentes à performance do sistema, isto é, gastos do CPU, utilização de memória RAM e I/O. Para tal, utilizaram-se as dashboards fronecidas para visualizar as diferentes métricas quando corremos a nossa aplicação Ghost.

### 5.1 DashBoard CPU

**VM Instance - CPU utilization** - Fração de utilização de CPU nestas instâncias. Estes valores variam normalmente entre 0.0 e 1.0.

**Kubernetes Container - CPU usage time** - Uso de CPU acumulado em todos os núcleos utilizados pelo container. O valor está em segundos.

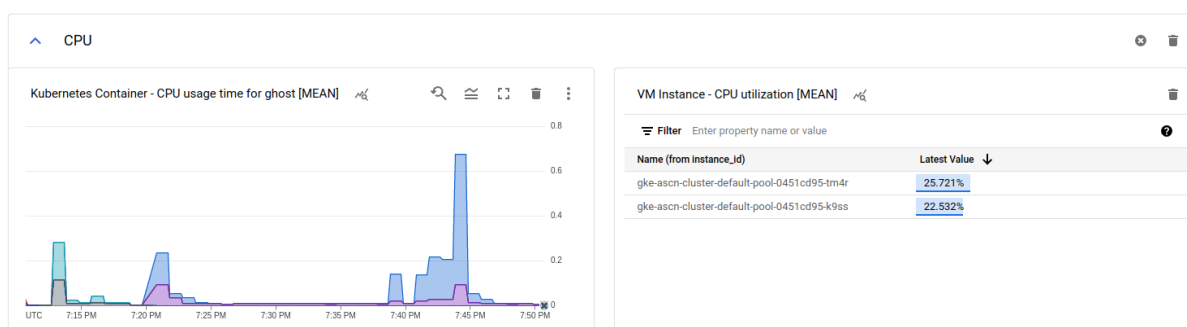


Figura 5.1: DashBoard CPU

### 5.2 DashBoard RAM

**Kubernetes Container - Memory usage** - Memória utilizada em bytes, por cada container.

**Kubernetes Container - Ephemeral storage usage** - Memória efêmera em bytes, por cada container.

**Kubernetes Container - Page faults** - Número de *page faults*, por cada container.

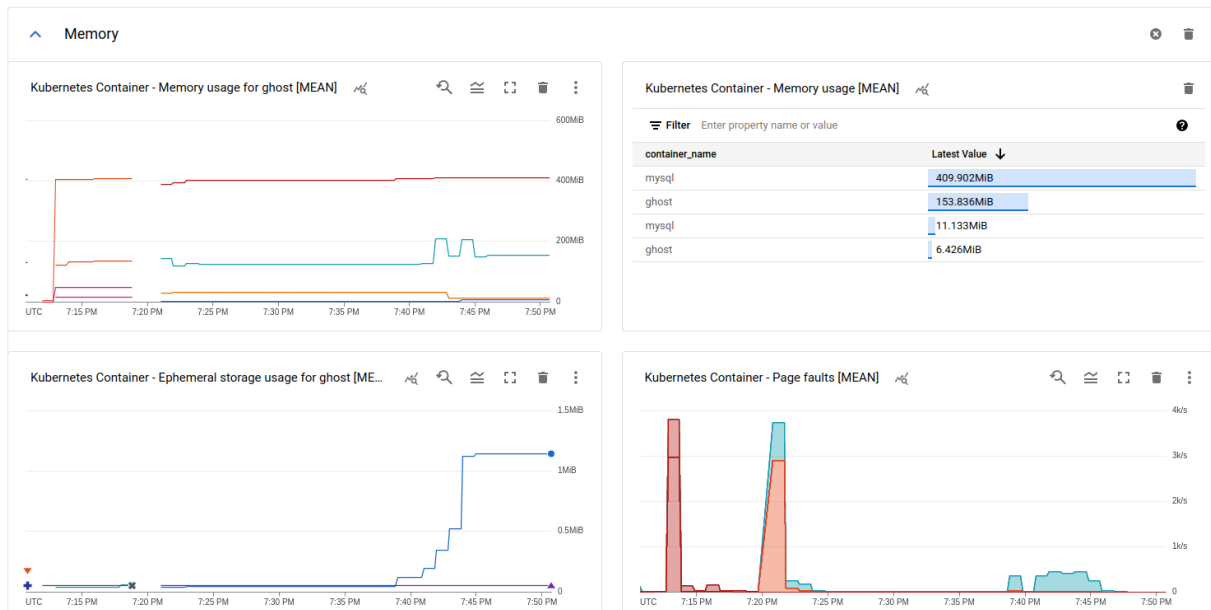


Figura 5.2: DashBoard CPU

### 5.3 DashBoard I/O

**Kubernetes Pod - Bytes received** - Número acumulado de bytes recebidos pela network, para cada container.

**RTT latencies per GKE Pod** - Uma distribuição do RTT medido sobre conexões TCP do GKE pod para Cloud endpoints.

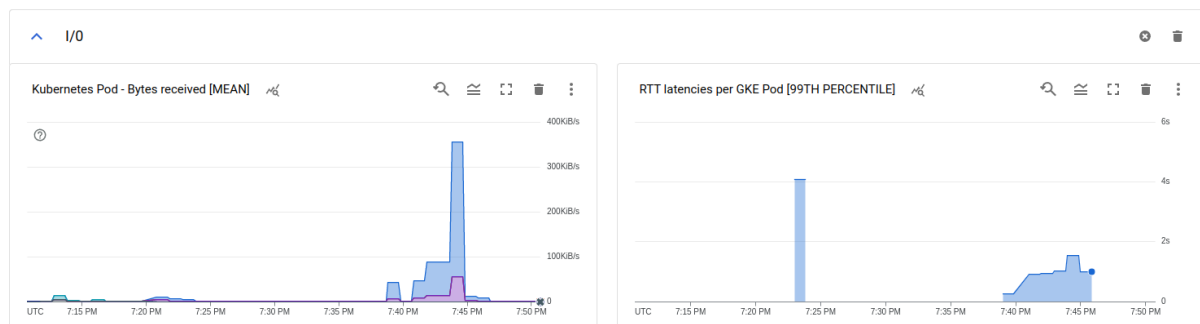


Figura 5.3: DashBoard I/O

## 6. Ferramentas de avaliação e testes desenvolvidos.

Para a realização da avaliação e da testagem da aplicação *Ghost*, utilizamos a ferramenta **JMeter**. Esta ferramenta é utilizada para carregar o comportamento funcional de teste e medir o desempenho, assim sendo permite realizar diferentes testes ao sistema.

Consequentemente, podemos avaliar diversos acessos à aplicação, nomeadamente o acesso à página principal da aplicação. Ademais, através da ferramenta, é possível definir o número de clientes (Threads) que acedem à aplicação ao mesmo tempo para executar as operações pré-definidas. Após os testes é gerada uma *dashboard*, onde é possível ver as diferentes métricas avaliadas relativas ao sistema.

### 6.1 Teste 1- Conexão à pagina principal

Os testes simulando vários números de Utilizadores a realizarem o acesso à página.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
http://95.194.17.83/2...	100	3461	1298	5572	1038.06	0.00%	15.8/sec	28762.95	61.34	1829666.4

Figura 6.1: N° Threads = 100

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
http://95.194.17.83/2...	200	7496	1734	13850	2399.29	0.00%	13.5/sec	24119.65	52.35	1829666.6

Figura 6.2: N° Threads = 200

### 6.2 Teste 2- Sign In

Os testes simulando 5 números de Utilizadores a realizarem o *sign in*.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
http://34.123.159.18...	5	739	732	752	6.95	0.00%	3.2/sec	0.98	1.57	310.0

Figura 6.3: N° Threads = 5

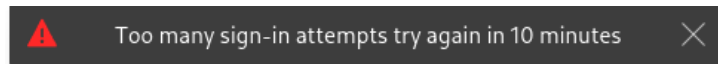
### 6.3 Teste 3- Subscribe

Os testes simulando 5 números de Utilizadores a realizarem o *subscribe*

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
http://34.66.215.137...	5	753	723	764	15.32	0.00%	3.2/sec	0.97	1.53	310.0

Figura 6.4: N° Threads = 5

Quando tentamos realizar tanto o *SignIn* como o *Subscribe* com o número de threads superior a 5, o Ghost impede que isso aconteça, bloqueando as tentativas:



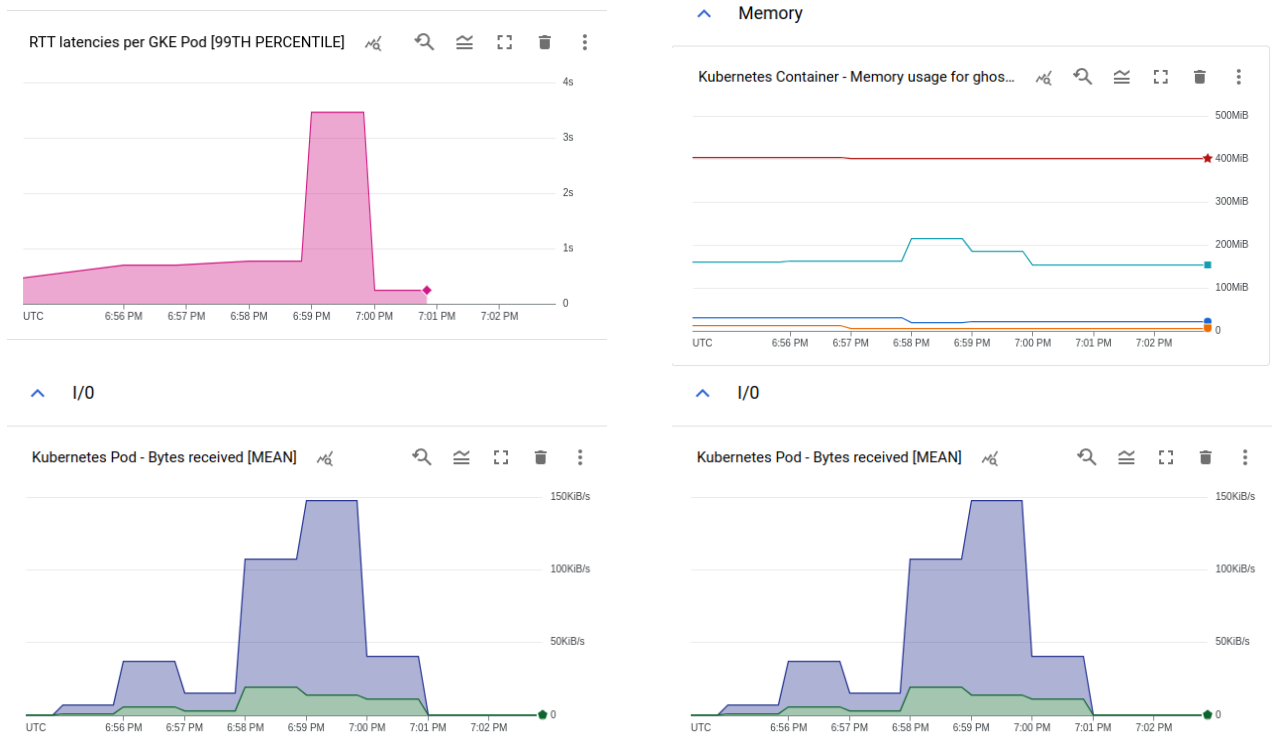
**Figura 6.5:** Mensagem de erro

As figuras 6.1 e 6.2 ilustram as simulações de acesso à página por 100 e 200 utilizadores, respectivamente. O "average" representa a média do tempo que a página demora a estar operacional.

Ora, os tempos médios são de 3461ms e 7496ms, respetivamente. Concluimos que ao dobrar os utilizadores o tempo médio aumenta para um pouco mais do dobro.

## 7. Monitorização durante os Testes

De maneira a perceber melhor o custo na performance e o poder computacional necessário para lidar com tantos pedidos a aceder a página principal precisamos de analisar as métricas através das *Dashboards*. As métricas consideradas são:



**Figura 7.1:** Monitorização dos testes

## 8. Conclusão

Dado por concluído o trabalho prático, iremos apresentar uma análise geral e crítica acerca do trabalho produzido.

Assim sendo, a instalação e a configuração automática da aplicação ocorreu sem falhas e foi possível monitorizá-la de acordo com métricas relevantes.

No entanto, na perspectiva de elaboração e melhoramento num trabalho futuro, consideramos que seria benéfico implementar mecanismos de replicação que permitam melhorar o desempenho e/ou resiliência da instalação do *Ghost*, com intuito de prevenir eventuais falhas que fossem detetadas.

Em suma, consideramos que o balanço do trabalho é positivo, uma vez que as dificuldades sentidas foram superadas com sucesso e os requisitos foram cumpridos.