

Implementação de um Sistema de Monitorização de TI Baseado em Ferramentas Open Source

Nuno M. T. Batuca
Instituto Politécnico de Castelo Branco
Av. Pedro Álvares Cabral
nº 12, 6000-084 Castelo Branco, Portugal
nbatuca@ipcbrcampus.pt

João M. L. P. Caldeira
Instituto Politécnico de Castelo Branco
Instituto de Telecomunicações
Portugal
jcaldeira@ipcbr.pt

Bruno Zarpelão
Universidade Estadual de Londrina
Londrina-PR, Brasil
brunozarpelao@uel.br

Vasco N. G. J. Soares
Instituto Politécnico de Castelo Branco
Instituto de Telecomunicações
Portugal
vasco.g.soares@ipcbr.pt

RESUMO

Partindo de um estudo prévio dos mesmos autores sobre o Information Technology Service Management (ITSM) e o Information Technology Infrastructure Library (ITIL), que estudou os fatores que estão associados à gestão do negócio das Tecnologias da Informação (TI), neste artigo descrevem-se os procedimentos necessários para a implementação de um sistema de monitorização de TI baseado em ferramentas *open source*. Este sistema é capaz de monitorizar e analisar em tempo real o funcionamento da infraestrutura de TI de uma organização, contribuindo para delinear a melhor estratégia de TI, com vista a melhorar a qualidade dos serviços e incrementar valor ao negócio.

Palavras-Chave

Implementação, Information Technology Service Management (ITSM), Information Technology Infrastructure Library (ITIL), Gestão de Processo de Negócio, Gestão de Serviços Empresariais, Sistema de Monitorização.

ABSTRACT

Based on a previous study of the same authors on Information Technology Service Management (ITSM) and Information Technology Infrastructure Library (ITIL), which studied the factors that are associated with the business management of Information Technology (IT), this article describes the procedures required to implement an IT monitoring system based on open source tools. This system can monitor and analyze in real time the functioning of an organization's IT infrastructure, contributing to outline the best IT strategy, aiming to improve the quality of services, and increasing business value.

Keywords

Business Process Management, Implementation, Information Technology Infrastructure Library (ITIL), Information Technology Service Management (ITSM), Monitoring System.

1. INTRODUÇÃO

O IT Service Management (ITSM) [1] descreve a forma como uma organização gere os seus serviços de Tecnologias da Informação (TI). Define as funções e responsabilidades de cada departamento e indivíduo em relação aos serviços de TI. Propõe um conjunto de boas práticas e processos, que visam a auxiliar as organizações na gestão eficiente da infraestrutura de TI (i.e., componentes e serviços). Contribui para uma maior

produtividade, redução de custos e aumento da satisfação do cliente. O Information Technology Infrastructure Library (ITIL) [2] é um *framework* utilizado no contexto da gestão do ITSM. É um conjunto específico de processos e diretrizes para o fornecimento de serviços de TI.

O trabalho apresentado no contexto deste artigo, complementa o estudo teórico prévio sobre ITSM e ITIL dos mesmos autores [3], que realça a importância de um sistema de monitorização que permita analisar em tempo real o desempenho da infraestrutura de TI de uma organização. Este sistema pode desempenhar um papel fundamental na definição da estratégia de TI mais adequada para o negócio. E tem implicação na melhoria da qualidade dos serviços e no aumento do valor agregado ao negócio.

Apesar do outsourcing de serviços ser cada vez mais uma realidade das organizações, muitas não dispõem de financiamento para adquirir serviços de monitorização. Assim, este artigo apresenta os procedimentos necessários para a implementação de um sistema de monitorização de TI baseado em ferramentas *open source*. O protótipo funcional foi montado sob o software de virtualização VirtualBox [4]. Usa Ubuntu Server [5] como sistema operativo base do servidor. Utiliza o Prometheus [6] para recolher métricas nas máquinas monitorizadas. E, por último, usa o Grafana [7] para representação e análise dos dados recolhidos.

Este artigo está organizado da seguinte forma. A secção 2 apresenta as ferramentas *open source* selecionadas para a implementação descrita neste documento. Na secção 3 apresentam-se as etapas necessárias para implementar um protótipo funcional e os resultados experimentais. Finalmente, na secção 4, apresentam-se as conclusões e o trabalho futuro.

2. FERRAMENTAS

Esta secção apresenta as ferramentas VirtualBox, Ubuntu Server, Prometheus e Grafana, que foram escolhidas para implementar um sistema de monitorização de TI, de acordo com o cenário ilustrado na Figura 1. Optou-se por ferramentas *open source* de fácil acesso e baixo custo (ou inexistente), que são uma mais-valia óbvia para orçamentos limitados. Outras vantagens são o acesso ao código aberto, a capacidade de customização, a existência de comunidades, e a independência a nível de fornecedores de software. De seguida, apresenta-se uma análise destas ferramentas a nível de licenciamento e de requisitos.

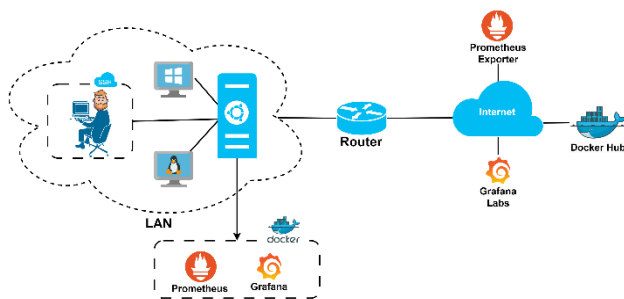


Figura 1. Arquitetura e ferramentas *open source* de suporte à implementação de um sistema de monitorização de TI.

2.1 Licenciamento

Antes de avançar para a implementação, é necessário analisar o licenciamento das ferramentas VirtualBox, Ubuntu Server, Prometheus e Grafana. Tal informação é fulcral, uma vez que se pretende ter um sistema de monitorização totalmente *open source* e que cumpra as obrigações de licenciamento de software.

O VirtualBox [8] é um software *open source*, propriedade da Oracle, que está protegido pela licença GPLv2. O código fonte do VirtualBox e os binários da plataforma, estão licenciados sob a GNU General Public License, versão 2. Como tal, é possível distribuir e modificar o pacote base, desde que se distribua todas as modificações também sob uso da licença GPLv2.

O Ubuntu [9] é um sistema operativo *open source*, propriedade da Canonical, que se encontra protegido pela “Creative Commons CC-BY-SA version 3.0 UK licence”. É possível efetuar o download, instalar e receber atualizações do Ubuntu gratuitamente. É também permitido modificar o Ubuntu para uso pessoal ou comercial interno. É ainda possível redistribuir o Ubuntu, se não tiver sido feita qualquer modificação ao mesmo.

O Docker [10] é um sistema de virtualização de sistemas operativos *open source*, que permite criar, implementar e gerir containers. Está licenciado sob a licença Apache 2.0.

O Prometheus [11] é uma ferramenta de monitorização de código aberto. Está licenciado sob a licença Apache 2.0, o que faz com que seja totalmente *open source*.

O Grafana Lab [12] está licenciado sob a licença AGPLv3. Assim, os clientes que efetuarem modificações no código fonte do Grafana, são obrigados a disponibilizar os recursos e modificações.

2.2 Requisitos

Para a instalação do VirtualBox [13] é recomendado o uso de processadores Intel ou AMD recentes. No que diz respeito à memória RAM, a mesma deve ter em conta os requisitos mínimos do sistema operativo a executar. A nível de espaço em disco, o VirtualBox necessita somente de 30 MB. No entanto, as máquinas virtuais necessitam do espaço definido pelos requisitos mínimos do sistema operativo a virtualizar. Os sistemas operativos suportados são o Windows, Linux, Mac OS, Solaris e OpenSolaris. Em relação ao Linux, por vezes suportadas versões não listadas que utilizam o mesmo *kernel*.

Para a virtualização do Ubuntu Server [14] é aconselhável um CPU de 1 GHz ou superior, 1 GB ou mais de memória RAM, e no mínimo de 2.5GB de espaço em disco.

Os requisitos mínimos do Docker [15] são um *kernel* de 64 bits, um CPU com suporte à virtualização Kernel-based Virtual Machine (KVM), o emulador Quick Emulator (QEMU) na versão 5.2 ou superior, e no mínimo 4 GB de memória RAM.

O Prometheus [16] requer um espaço em disco mínimo de 128 MB, para armazenamento da base de dados (BD).

Para executar o Grafana [17], basta ter uma máquina com sistema operativo Windows ou Linux (Debian, Ubuntu, CentOS, Fedora, OpenSuse, RedHat). A nível de hardware, é recomendado no mínimo 255 MB de RAM e 1 CPU. No que diz respeito ao armazenamento, a BD por omissão utiliza o SQLite, sendo ainda suportado o MySQL 5.7 e o PostgreSQL 10+. O Grafana utiliza um browser Web para visualização, sendo suportados Chrome/Chromium, Firefox, Safari e Microsoft Edge nas versões mais recentes.

3. IMPLEMENTAÇÃO DO SISTEMA DE MONITORIZAÇÃO

Nesta secção, apresentam-se as etapas necessárias para implementar um protótipo funcional de um sistema de monitorização de TI, que integra as ferramentas *open source* Ubuntu Server, Docker, Prometheus e Grafana, acima descritas.

3.1 Instalação e Configuração do Ubuntu Server

Deve começar-se por efetuar o download do ISO do Ubuntu Server a partir do site da Canonical. Depois, será necessário criar uma *pen* de arranque com a ISO do Ubuntu Server. De seguida, arranque a máquina (física ou virtual) utilizando como dispositivo de arranque a *pen*. Será agora iniciado o processo de instalação e configuração do servidor Ubuntu.

Após o servidor arrancar, será disponibilizada a opção “Install Ubuntu Server”, a qual deve ser selecionada de modo a dar início à instalação, conforme apresentado na Figura 2. Em seguida, como é possível ver na Figura 3, é solicitado o idioma pretendido para o sistema operativo.



Figura 2. Processo de instalação do Ubuntu Server: início da instalação.

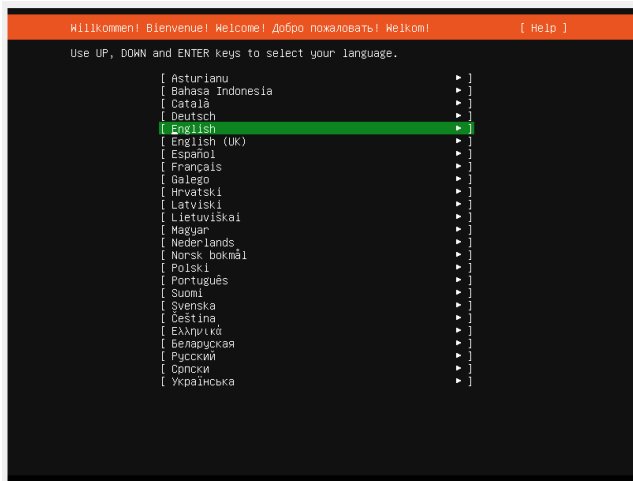


Figura 3. Processo de instalação do Ubuntu Server: seleção do idioma de instalação.

Dado que podem existir atualizações, o utilizador é questionado se deseja atualizar o sistema operativo, como ilustrado na Figura 4. Dando continuidade à instalação, na Figura 5, é solicitado qual o *layout* do teclado a utilizar.

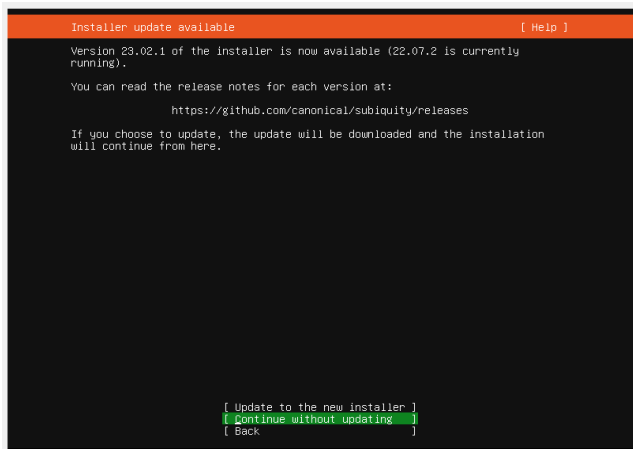


Figura 4. Processo de instalação do Ubuntu Server: *installer update*.

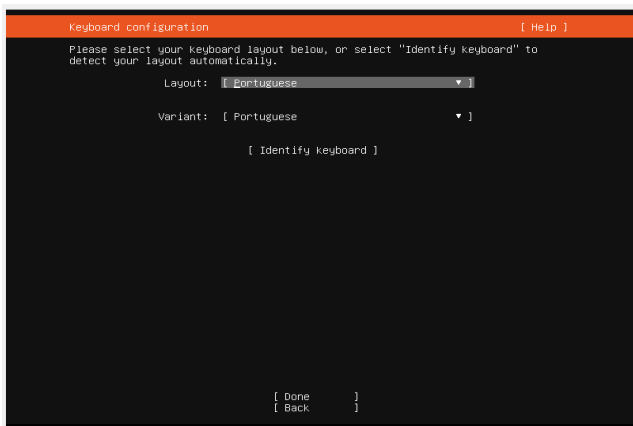


Figura 5. Processo de instalação do Ubuntu Server: idioma do teclado.

Na Figura 6, são apresentadas as opções que permitem escolher a versão do sistema operativo que se pretende instalar. É possível optar por uma versão completa, ou uma versão minimizada. Após selecionar a versão a instalar, será necessário efetuar as configurações de rede. No cenário implementado no contexto deste trabalho, o router dispõe de servidor de DHCP, pelo que o IP é atribuído pelo mesmo. No entanto, como apresentado na Figura 7, é possível configurar manualmente os parâmetros de rede neste momento, ou configurar mais tarde após a instalação.

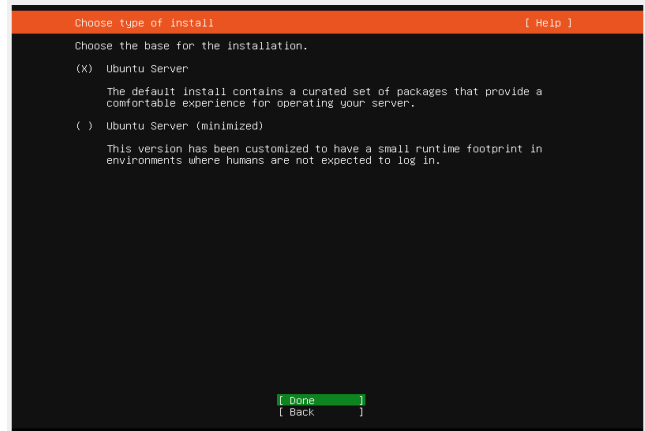


Figura 6. Processo de instalação do Ubuntu Server: seleção da versão a instalar.

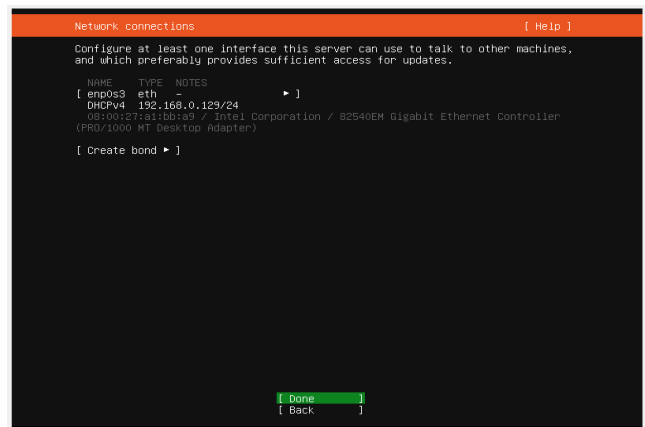


Figura 7. Processo de instalação do Ubuntu Server: configurações de rede.

Na Figura 8 é apresentada a opção que permite configurar um *proxy* para ligação à Internet. Neste exemplo, assume-se que não vai ser usado um *proxy*.

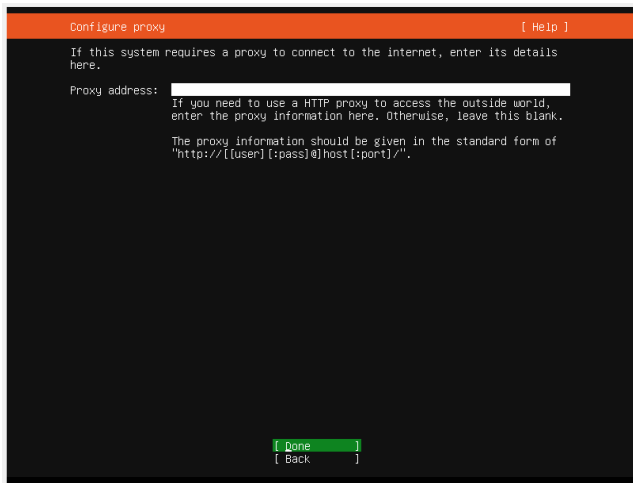


Figura 8. Processo de instalação do Ubuntu Server: configuração de proxy.

Neste ponto, será configurada a partição ou partições do disco, como apresentado na Figura 9. Neste caso, foi alocado o tamanho total definido para o disco. Na Figura 10, ainda dentro da configuração de armazenamento, o sistema apresenta as partições criadas e a alocação a cada setor necessário para o sistema funcionar.

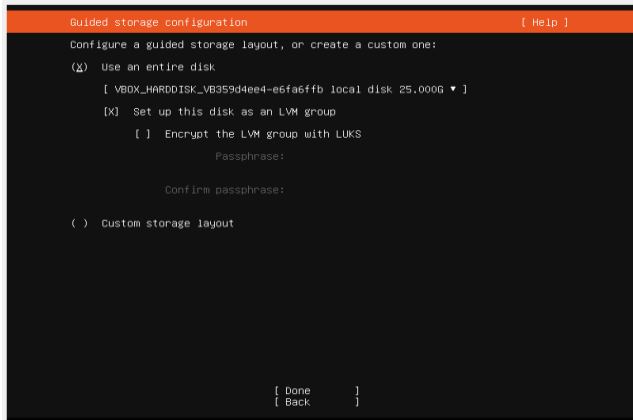


Figura 9. Processo de instalação do Ubuntu Server: configuração de armazenamento.

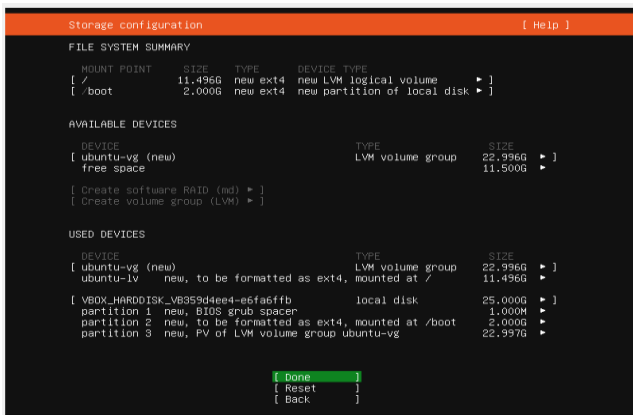


Figura 10. Processo de instalação do Ubuntu Server: partições de armazenamento.

Antes de continuar com a instalação, será necessário configurar um utilizador administrador, a respetiva *password* e o nome do servidor, conforme exemplificado na Figura 11. Na Figura 12, são apresentadas algumas *features* que podem ser adicionadas durante a instalação do sistema operativo. Aqui foi instalado apenas o Docker. Tanto o Prometheus como o Grafana serão instalados a posteriori num *container* Docker.

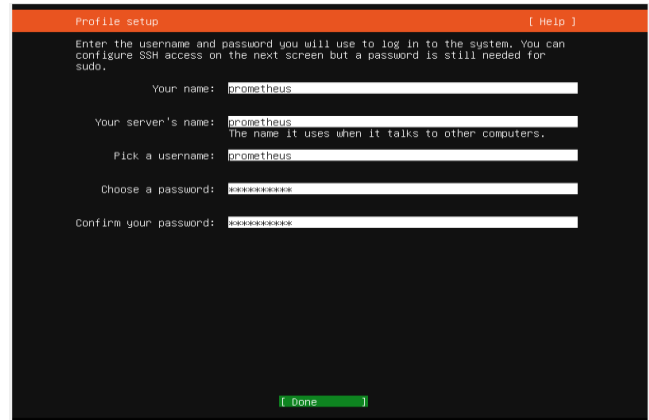


Figura 11. Processo de instalação do Ubuntu Server: configuração de utilizador.

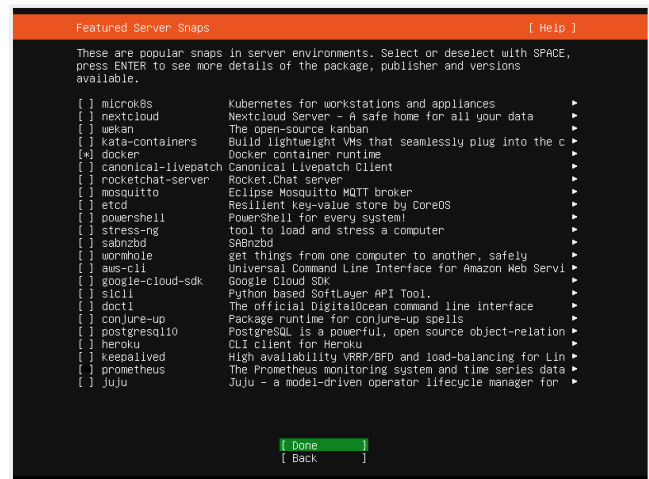


Figura 12. Processo de instalação do Ubuntu Server: lista de features disponíveis.

Após estar concluída a instalação, será necessário reiniciar a máquina, como apresentado na Figura 13. Após esta reiniciar, é possível efetuar *login*. Se o mesmo for bem-sucedido, são apresentadas algumas informações sobre o sistema, como ilustrado na Figura 14.

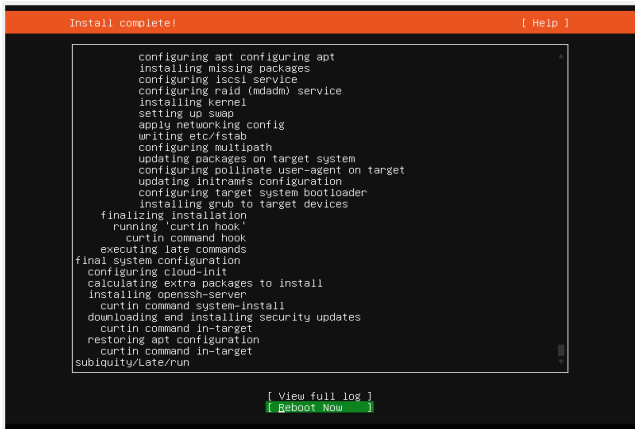


Figura 13. Processo de instalação do Ubuntu Server: instalação concluída.

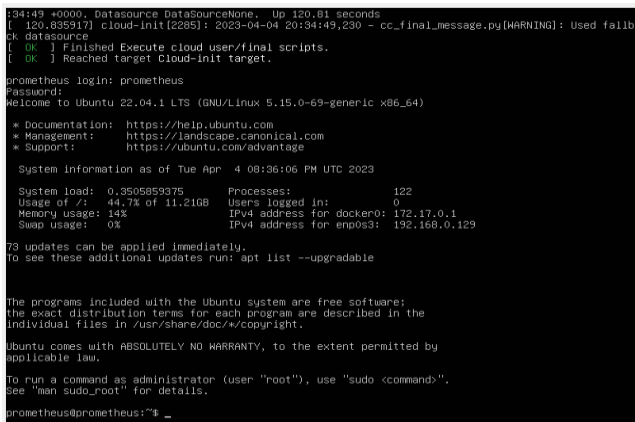


Figura 14. Processo de instalação do Ubuntu Server: login na máquina.

3.2 Configuração do Docker

Como descrito acima, o Docker foi instalado no decorrer da instalação do Ubuntu Server. Assim, de seguida, serão apresentados os comandos para a instalação e configuração dos *containers* necessários para o cenário descrito neste trabalho.

O Docker em Ubuntu Server funciona em modo linha de comandos. Pelo que é requerido conhecer os comandos básicos para as configurações necessárias. No Docker Hub [18] está disponível um repositório com imagens que podem ser instaladas de forma fácil, tais como o Prometheus e o Grafana.

A execução do comando `$sudo docker run -itd --rm -p 9090:9090 --name prometheus prom/prometheus`, apresentada na Figura 15, vai efetuar o *download* e instalar o *container* Prometheus no Docker. Tem como opções o `-p` o qual permite definir o porto a utilizar por parte do Prometheus, bem como o `-name` que permite definir um nome ao *container* [19].

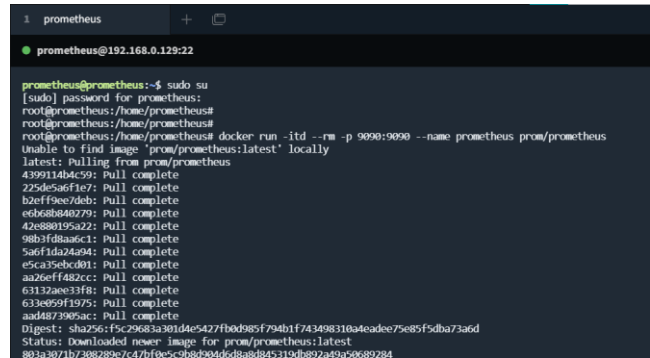


Figura 15. Instalação do *container* Prometheus.

É necessário repetir o passo anterior para instalar o Grafana. A execução do comando `$sudo docker run -itd --rm -p 3000:3000 --name grafana grafana/grafana`, apresentada na Figura 16, permite efetuar o *download* e a instalação do *container* Grafana no Docker [20].

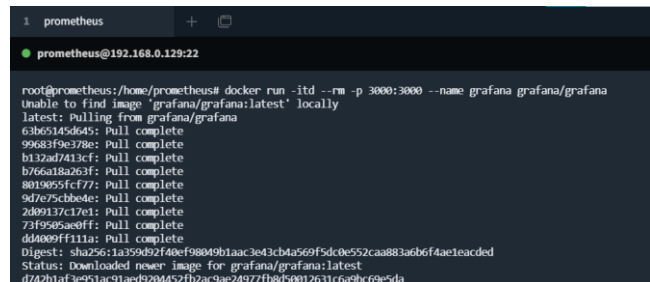


Figura 16. Instalação do *container* Grafana.

Após instalar as imagens no Docker, é possível visualizá-las utilizando o comando `$docker images`, ilustrado na Figura 17. Neste caso, é possível ver que ambas as imagens necessárias para a implementação deste cenário já se encontram instaladas.

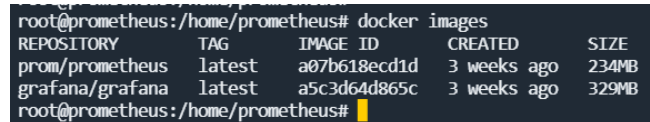


Figura 17. Imagens instaladas.

A execução do comando `$sudo lsof -i -P -n | grep LISTEN`, apresentada na Figura 18, permite listar os portos que se encontram à escuta no Ubuntu Server. Tal permite confirmar se os portos selecionados no Docker se encontram devidamente reservados.

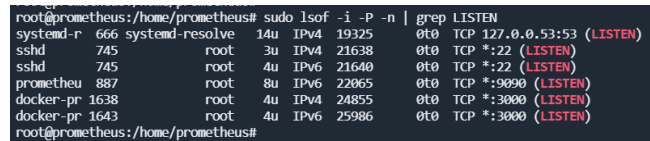


Figura 18. Portos em uso pelo servidor Ubuntu.

O comando `$docker ps` permite visualizar os *containers* que se encontram em execução, como ilustrado na Figura 19. Neste caso, verifica-se que não existe qualquer *container* em execução. O comando `$docker ps -a` permite visualizar todos os *containers* instalados, mesmo que não estejam em execução. Para inicializar os *containers* pretendidos, basta utilizar o comando `$docker start <nome do container>` [21].

Dentro do *container* do Prometheus, é necessário aceder à pasta onde se encontra o ficheiro de configuração do Prometheus. Para tal, deve-se usar o comando `$cd etc/prometheus`. Para verificar a existência do ficheiro podemos listar o conteúdo da pasta utilizando o comando `$ls -la`. Tal possibilita verificar a existência do ficheiro de configuração do Prometheus, que tem o nome `prometheus.yml`. A Figura 27 mostra o *output* resultante da execução destes comandos.

```
root@prometheus:/# docker exec -u root -it prometheus sh
/prometheus #
/prometheus # cd /etc/prometheus/
/etc/prometheus # ls -la
total 20
drwxr-xr-x 1 nobody nobody 4096 Mar 21 13:37 .
drwxr-xr-x 1 root root 4096 May 9 21:06 ..
lrwxrwxrwx 1 nobody nobody 39 Mar 21 13:37 console_libraries -> /usr/share/prometheus/console_libraries
lrwxrwxrwx 1 nobody nobody 31 Mar 21 13:37 consoles -> /usr/share/prometheus/consoles/
-rw-r--r-- 1 nobody nobody 1283 May 10 20:17 prometheus.yml
/etc/prometheus # vi prometheus.yml
```

Figura 27. Localização do ficheiro de configuração do Prometheus.

As configurações do Prometheus encontram-se dentro do ficheiro `prometheus.yml`, apresentado na Figura 28. Para configurar o mesmo, é necessário utilizar um editor de texto, neste caso o `vi`.

Assim, após executar o comando `$vi prometheus.yml`, será apresentado o conteúdo do ficheiro de configuração do Prometheus. Para adicionar uma nova máquina a monitorizar, é necessário configurar dois campos: `job_name` e `static_configs`. O `job_name` é o nome que irá ser dado à máquina monitorizada e apresentada no Prometheus. O `static_configs` é o IP da máquina a monitorizar e o porto que a mesma vai usar para que o Prometheus tenha acesso às métricas [24].

Neste caso, será monitorizada uma máquina com o Windows 10, à qual foi dado o nome `W10 prometheus`, que tem o IP `192.168.0.160` e o porto `9182`. Para adicionar mais máquinas, basta repetir o processo anterior.

```
my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        - alertmanager:9093

# load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label 'job=<job_name>' to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]

  - job_name: "W10 prometheus"

    static_configs:
      - targets: ["192.168.0.160:9182"]
```

Figura 28. Configuração do Prometheus.

Após adicionar as máquinas a monitorizar, é necessário reiniciar o serviço do Prometheus, como ilustrado na Figura 29. O comando `$ps aux | grep prometheus` permite visualizar todos os processos que se encontram em execução. Neste caso, é possível identificar o número de processo facilmente, pois o mesmo encontra-se com o `prometheus.yml` em execução. De modo a terminar o processo, deve utilizar-se o comando `$sudo kill -HUP <nº processo>`. Neste caso `40968` é o número do processo que se encontra em execução.

```
root@prometheus:/home/prometheus# ps aux | grep prometheus
nobody 40968 0.3 3.4 789228 65812 ? Ssl 21:06 0:01 /bin/prometheus --config.file=/etc/prometheus/prometheus.yml
nobody 41000 0.0 0.5 17172 11100 ? Ss 21:09 0:00 sshd: prometheus [priv]
root 41265 0.0 0.4 17464 8548 ? S 21:09 0:00 sshd: prometheus@pts/2
root 41400 0.0 0.1 6688 2272 pts/3 S+ 21:13 0:00 grep --color=auto prometheus
root@prometheus:/home/prometheus#
root@prometheus:/home/prometheus#
root@prometheus:/home/prometheus# sudo kill -HUP 40968
```

Figura 29. Reiniciar o processo Prometheus.

Para aceder ao ambiente gráfico do Prometheus, no *browser* acede-se ao IP e porto `192.168.0.130:9090` (i.e., *localhost*) configurados anteriormente no ficheiro `prometheus.yml`. Este ambiente é ilustrado na Figura 30, onde se apresentam informações sobre o estado dos *targets* (i.e., máquinas monitorizadas). Neste exemplo estamos a monitorizar o próprio serviço do Prometheus, e a máquina `W10 Prometheus` configurada anteriormente.

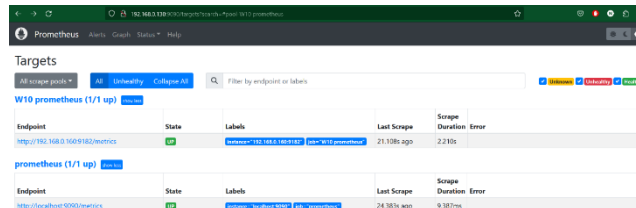


Figura 30. Interface gráfica do Prometheus.

Nesta mesma interface, na opção `Graph` é possível criar gráficos no Prometheus, para análise de dados das máquinas monitorizadas. Esta função executa uma *query* que somente apresenta os dados no momento da consulta. Não permite guardar os gráficos para uma apresentação posterior [25]. Contudo, esta opção permite testar se o Prometheus se encontra a recolher métricas como esperado. Por exemplo, na Figura 31 pode ver-se o resultado de execução de uma *query* que mostra o número de *bytes* recebidos na interface de rede da máquina monitorizada no período de um minuto.

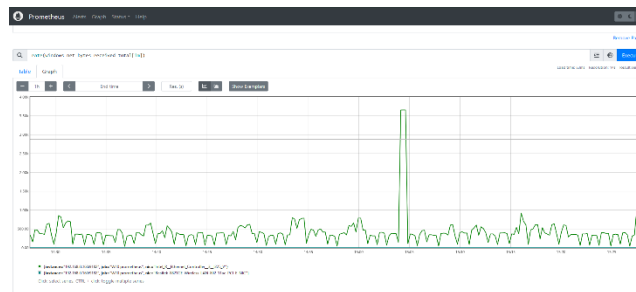


Figura 31. Número de bytes recebidos na interface de rede da máquina monitorizada.

A Figura 32 apresenta outro exemplo em que se consulta o número de *bytes* de memória alocados numa máquina, em períodos de cinco minutos.



Figura 32. Número de bytes de memória alocados numa máquina monitorizada.

3.5 Configuração do Grafana

O Grafana complementa a análise dos dados recolhidos pelo Prometheus. Permite construir *dashboards* personalizados, com informação em tempo real do estado das máquinas e/ou serviços [26].

Para configurar o Grafana, no *browser* acede-se ao IP e porto 192.168.0.130:3000 (i.e., *localhost*), como ilustrado na Figura 33. Será necessário adicionar a base de dados do Prometheus. Para tal, em Configuration deve seleccionar a aba Data Sources e escolher a opção Add data source.

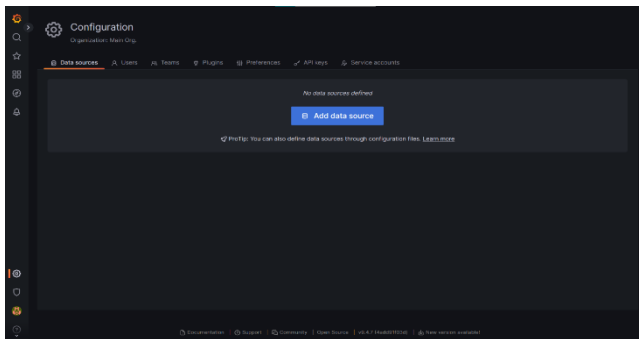


Figura 33. Configurações do Grafana.

Depois será questionado qual o tipo de base de dados. Deverá ser seleccionada a opção Prometheus, como ilustrado na Figura 34.

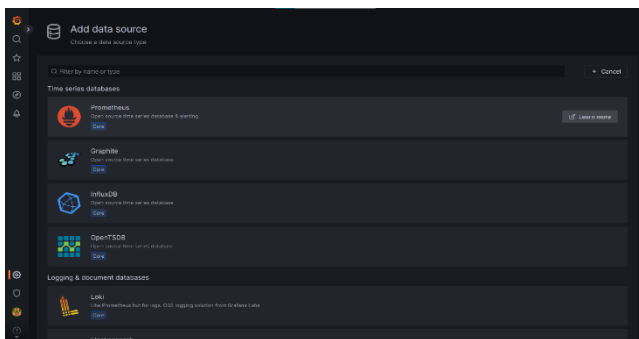


Figura 34. Opções de base de dados para integração com o Grafana.

De seguida, será necessário definir um nome para a configuração da base de dados, bem como o URL da máquina onde se encontra a base de dados do Prometheus, como ilustrado na Figura 35.

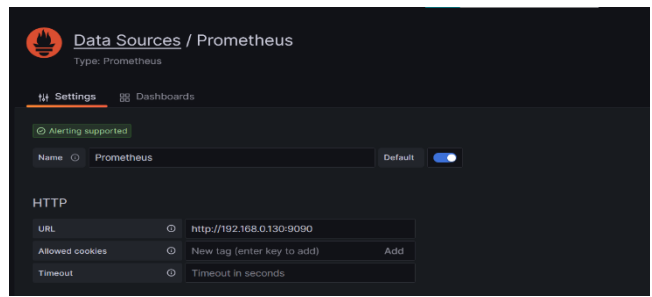


Figura 35. Localização da base de dados do Prometheus.

A configuração do *dashboard* vai permitir que sejam apresentados os dados recolhidos de uma forma mais intuitiva e interativa. Isto irá facilitar a leitura e análise dos mesmos. No menu apresentado na Figura 36 será necessário seleccionar New dashboard e depois a opção Add a new panel.

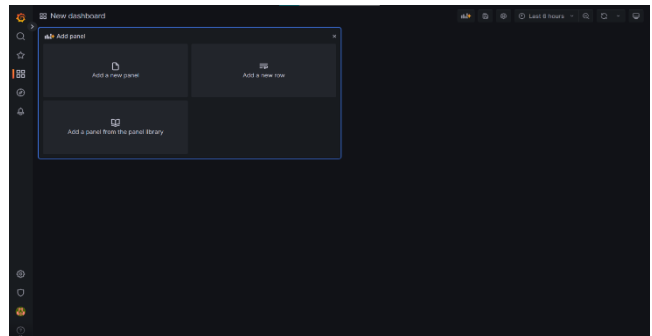


Figura 36. Menu *dashboard* Grafana.

A aba de edição do *dashboard* apresentada na Figura 37, permite configurar e personalizar o mesmo, bem como o tipo de gráfico a apresentar. Para apresentar a informação no gráfico do *dashboard* é necessário executar uma *query* que vai recolher os dados da base de dados do Prometheus.

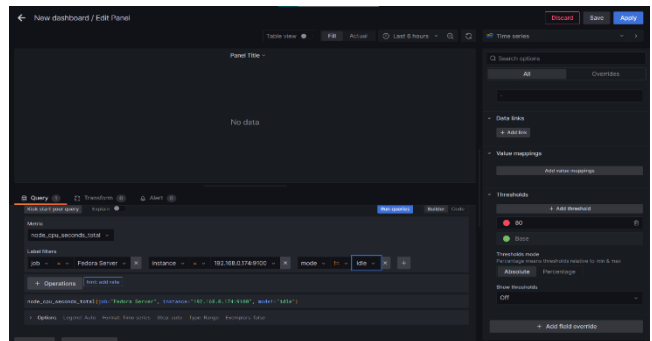


Figura 37. Construção do gráfico no Grafana.

A Figura 38 apresenta o gráfico gerado pela execução de uma *query* que vai obter os dados de tempo de CPU gasto pelo sistema da máquina monitorizada (em segundos). A linguagem de *query* da base de dados utilizada pelo Prometheus é denominada por Prometheus Query Language (PromQL). É possível obter mais informação sobre a mesma na documentação do Prometheus. Na Figura 39 ajustou-se a janela temporal para um intervalo de cinco minutos.

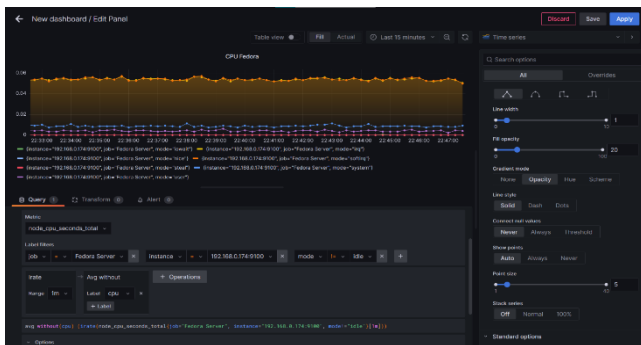


Figura 38. Resultado da execução de uma *query* no Grafana, para obter os dados de tempo de CPU gasto pelo sistema da máquina monitorizada num período de quinze minutos.

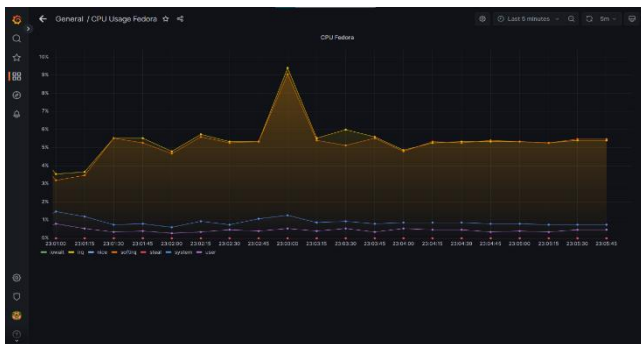


Figura 39. Resultado da execução de uma *query* no Grafana, para obter os dados de tempo de CPU gasto pelo sistema da máquina monitorizada num período de cinco minutos.

No Grafana Labs [27] são disponibilizados *dashboards* pré-configurados. A Figura 40 apresenta um exemplo de um *dashboard* pré-configurado. Neste caso as métricas retiradas são o *uptime* da máquina, CPU, memória, entre outras. Mas, estas podem ser configuradas com base nas necessidades definidas pelo administrador do sistema de monitorização. As métricas recolhidas podem auxiliar na identificação de padrões de carga e na deteção de problemas que possam originar interrupção dos serviços. Poder-se-á assim intervir de modo a prevenir interrupções.

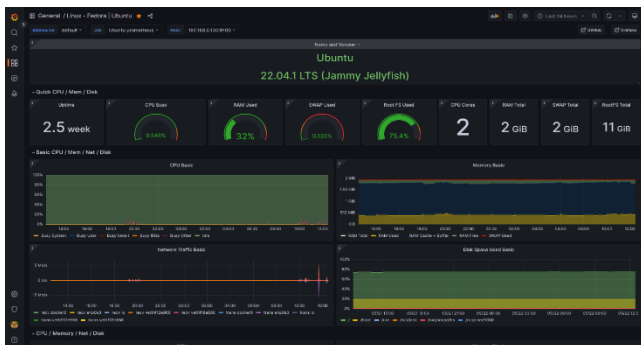


Figura 40. *Dashboard* Grafana pré-configurado para servidor Ubuntu.

4. CONCLUSÃO

O presente trabalho surge no âmbito do projeto aplicado do Mestrado em Engenharia Informática do Instituto Politécnico de Castelo Branco [28]. Teve por objetivo estudar a gestão de serviços TI no contexto atual das organizações.

Numa primeira etapa [3] foram enquadrados e clarificados os conceitos do Information Technology Service Management (ITSM) e da *framework* Information Technology Infrastructure Library (ITIL). O ITSM permite relacionar pessoas e serviços, que são suportados por uma camada tecnológica que agiliza o processo de transformação digital nas organizações. O ITIL é o conjunto de boas práticas em infraestrutura, manutenção e operação, que têm por objetivo otimizar os recursos de TI. O ITIL é a biblioteca de infraestrutura de TI e o ITSM a abordagem estratégica à gestão de TI. Nesta segunda etapa, descrita neste artigo, apresenta-se uma proposta de uma solução para um sistema de monitorização de TI baseado em ferramentas *open source*.

O *outsourcing* de serviços é cada vez mais uma realidade nas organizações, explorando-se o know-how das empresas prestadoras de serviços, e a atribuição do ónus da responsabilidade. No entanto, isto implica grandes investimentos a nível de terceirização de serviços. Habitualmente as pequenas e médias empresas não dispõem de capacidade financeira para o *outsourcing*. Com recurso a ferramentas *open source* é possível implementar um sistema de monitorização com um custo bastante reduzido. Ainda que possam existir problemas associados tais como, falta de suporte, complexidade e curva de aprendizagem, incompatibilidades futuras, sustentabilidade dos projetos, e riscos de segurança.

Neste artigo, procedeu-se ao levantamento e seleção de ferramentas *open source* e ao desenho da arquitetura de suporte ao sistema de monitorização. Descreveu-se detalhadamente o processo de implementação e integração de ferramentas. Demonstrou-se o protótipo funcional. Este sistema de monitorização proativa permite, em tempo real, recolher dados, visualizar métricas e o estado dos serviços em *dashboards*. Estes painéis visuais ajudam a identificar uma falha ou interrupção de serviço de forma fácil e eficaz. Ou seja, se existir uma falha num servidor, esta poderá ser vista em tempo real, sem que sejam os utilizadores finais a notificá-la à posteriori, depois de detetarem a indisponibilidade do serviço.

Quanto ao trabalho futuro identificam-se as seguintes possibilidades para complementar o sistema de monitorização. Criar um mecanismo que alerte proativamente a equipa de monitorização de incidentes, via e-mail ou serviço de mensagens instantâneas. Criar um mecanismo de automação que interliga o sistema de monitorização com a plataforma de *ticketing* da organização. Considerar a utilização de mecanismos de inteligência artificial, que podem ajudar a analisar e interpretar incidentes em tempo real, identificar padrões, detetar anomalias e fornecer insights valiosos, para melhorar a eficiência e a tomada de decisões.

CONTRIBUIÇÕES DOS AUTORES

Nuno M. T. Bataca: investigação, metodologia, análise formal, validação, preparação de rascunho de redacção-origina.

Bruno Zarpelão: revisão-escrita e edição, supervisão.

Vasco N. G. J. Soares, João M. L. P. Caldeira: análise formal, validação, revisão-escrita e edição, supervisão.

Os autores declaram não haver conflito de interesses.

REFERÊNCIAS

- [1] S. D. Galup, R. Dattero, J. J. Quan, and S. Conger, "An overview of it service management," *Communications of the ACM*, vol. 52, no. 5, 2009, doi: 10.1145/1506409.1506439.

- [2] A. Krishna Kaiser, *Become ITIL Foundation Certified in 7 Days*. 2017. doi: 10.1007/978-1-4842-2164-8.
- [3] N. Batuca, B. B. Zarpelão, J. M. L. P. Caldeira, and V. N. G. J. Soares, "IT Service Management: Conceitos e Processos," *Revista de Ciências da Computação*, vol. 18, pp. 17-42, 2023, doi: 10.34627/rcc.v18i0.294
- [4] "VirtualBox," 2023. <https://www.virtualbox.org/> (accessed Jun. 19, 2023).
- [5] "Ubuntu," 2023. <https://ubuntu.com/server/docs> (accessed Jun. 19, 2023).
- [6] "Prometheus," 2023. <https://prometheus.io/> (accessed Jun. 16, 2023).
- [7] "Grafana," 2023. <https://grafana.com/> (accessed Jun. 19, 2023).
- [8] "Licensing VM VirtualBox," 2023. https://www.virtualbox.org/wiki/Licensing_FAQ (accessed Jun. 19, 2023).
- [9] "Licensing Ubuntu," 2023. <https://ubuntu.com/legal/intellectual-property-policy> (accessed Jun. 19, 2023).
- [10] "Licensing Docker," 2023. <https://docs.docker.com/engine/>
- [11] "Licensing Prometheus," 2023. <https://github.com/prometheus/prometheus/blob/main/LICENSE> (accessed Jun. 19, 2023).
- [12] "Licensing Grafana," 2023. <https://grafana.com/licensing/> (accessed Jun. 19, 2023).
- [13] "Requirements VM VirtualBox," 2023. https://www.virtualbox.org/wiki/End-user_documentation (accessed Jun. 19, 2023).
- [14] "Requirements Ubuntu," 2023. <https://ubuntu.com/server/docs/installation> (accessed Jun. 19, 2023).
- [15] "Requirements Docker," 2023. <https://docs.docker.com/desktop/install/linux-install/>
- [16] "Requirements Prometheus," 2023. <https://prometheus.io/docs/prometheus/latest/storage/> (accessed Jun. 19, 2023).
- [17] "Requirements Grafana," 2023. <https://grafana.com/docs/grafana/latest/setup-grafana/installation/> (accessed Jun. 19, 2023).
- [18] "Docker Hub," 2023. <https://hub.docker.com/>
- [19] "Install Prometheus," 2023. <https://prometheus.io/docs/prometheus/latest/installation/>
- [20] "Install Grafana," 2023. <https://grafana.com/docs/grafana/latest/setup-grafana/installation/docker/>
- [21] "Start Container Docker," 2023. https://docs.docker.com/engine/reference/commandline/container_start/
- [22] "Node Exporter Prometheus," 2023, [Online]. Available: <https://prometheus.io/docs/guides/node-exporter/#monitoring-linux-host-metrics-with-the-node-exporter>
- [23] "Windows Exporter Prometheus," 2023. https://github.com/prometheus-community/windows_exporter/
- [24] "Target Exporter Prometheus," 2023, [Online]. Available: <https://prometheus.io/docs/guides/multi-target-exporter/#querying-multi-target-exporters-with-prometheus>
- [25] "Query Examples Prometheus," 2023. <https://prometheus.io/docs/prometheus/latest/querying/examples/>
- [26] "Documents Grafana," 2023. <https://grafana.com/docs/grafana/latest/>
- [27] "Dashboards Grafana Labs," 2023. <https://grafana.com/grafana/dashboards/>
- [28] "Mestrado Eng Informática IPCB," 2023. <https://www.ipcb.pt/estcb/ensino/mestrado-em-engenharia-informatica> (accessed Jun. 19, 2023).