

Atualização de infraestrutura de orquestração open source Puppet 3 para open source Puppet LTS 7



Licenciatura em Eng. Informática
Estágio-Projecto 2021-2022



Gonçalo André Correia

Orientador na empresa: Eng. Pedro Pessoa
Orientador no departamento: Prof. Dr. José Saias

Trabalho desenvolvido na empresa Angulo Sólido no âmbito da disciplina de Estágio-Projeto da Licenciatura em Eng. Informática.

Évora, 6 de fevereiro de 2022

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos	1
1.3	Contribuições	1
1.4	Estrutura do documento	1
2	Ambiente empresarial	2
3	Estado da arte	3
4	Ambiente de desenvolvimento	4
4.1	Ambiente técnico	4
4.1.1	Hardware	4
4.1.2	Software	4
4.1.3	Ambiente aplicativo	6
4.2	Metodologia de trabalho	6
5	Trabalho desenvolvido	7
5.1	Descrição detalhada	7
5.1.1	Repositório e documentação	7
5.1.2	Instalação de um Puppet Master e de um Puppet Agent em diferentes VM's . . .	7
5.1.3	Guia para a instalação do Puppet7 de raiz	8
5.1.4	Guia para a instalação do r10k	10
5.1.5	Guia para um eventual upgrade dos Agents Puppet 3 para Puppet 7	12
6	Avaliação crítica	13

1 Introdução

1.1 Enquadramento

O projeto de estágio foi desenvolvido pelo aluno Gonalo Andr  Correia, na institui o Angulo S lido. Enquadra-se no  mbito da disciplina de Est gio-Projeto da Licenciatura em Eng. Inform tica da Universidade de  vora. Conta com a orienta o do Eng. Pedro Pessoa, CEO da Angulo S lido, e do Dr. Jos  Saias, professor na Universidade de  vora. O projeto realizado obedeceu  s seguintes condi es previamente estabelecidas: ter in cio a 20 de dezembro de 2021 e t rmino a 22 de janeiro de 2022 apresentando uma carga hor ria compat vel acordada entre a empresa e o aluno cumprindo as tarefas propostas a executar.

A Angulo S lido, sediada em  vora,   uma empresa que se dedica   presta o de servi os de opera es baseados em solu es de c digo aberto [1]. Os principais alvos de mercado s o pequenas e m dias empresas no mercado global que pretendam aceder a uma equipa de opera es pronta. Com cerca de 15 anos de experi ncia na entrega de solu es de c digo aberto, a empresa pretende contribuir para o r pido desenvolvimento das solu es dos seus clientes.

O trabalho de est gio foi desenvolvido, unicamente, pelo estagi rio, uma vez que n o se encontrava integrado numa equipa.

1.2 Objetivos

O trabalho de est gio, desenvolvido na institui o Angulo S lido, tem como principal objetivo a partir de um sistema instalado e em utiliza o, pretende-se estudar, planejar e executar, total ou parcialmente, a atualiza o do mesmo.

1.3 Contribui es

O desempenho no desenvolvimento deste projeto de est gio n o teria sido o mesmo sem a ajuda e apoio de algumas pessoas,  s quais gostaria de aqui expressar o meu reconhecimento pela sua contribui o direta ou indiretamente:

Assim, comearia por agradecer ao orientador Prof. Dr. Jos  Saias pelo seu acompanhamento.

Um agradecimento especial ao Eng. Pedro Pessoa pelo apoio, liderana e pela forma como me recebeu na empresa Angulo S lido.

Queria agradecer t mb m ao Eng. Miguel Carvalho pela disponibilidade que mostrou sempre que surgiu alguma d vida na parte mais t cnica deste projeto.

E finalmente, aos meus familiares que me apoiaram sempre que tive que trabalhar remotamente, a partir de casa.

A todos os que est o e n o est o aqui indicados, o meu obrigado.

1.4 Estrutura do documento

Este documento est  organizado da seguinte maneira:

- Sec o 2 - Ambiente empresarial - contexto da empresa e o meu projeto.
- Sec o 3 - Estado da Arte - levantamento e an lise de v rias ferramentas ou metodologias.
- Sec o 4 - Ambiente de desenvolvimento - descri o do ambiente de hardware e software.
- Sec o 5 - Trabalho desenvolvido - trabalho elaborado durante o est gio.
- Sec o 6 - Avalia o cr tica - an lise da experi ncia do ponto vista t cnico e n o t cnico.

2 Ambiente empresarial

A Angulo Sólido, é uma empresa portuguesa, com a sua sede em Évora e uma filial em Lisboa. Trata-se de uma empresa privada detida e gerida pelo Eng. Pedro Pessoa e pelo Eng. Gustavo Homem. Dedicase à prestação de serviços de operações baseados em soluções de código aberto [1].

Apresenta cerca de 15 anos de experiência e pretende contribuir para o rápido desenvolvimento das soluções dos seus clientes.

A empresa é constituída por equipas, onde cada equipa da empresa é associada a um ou mais clientes e é responsável por trabalhar com eles.

No primeiro dia do estágio, dirigi-me à sede em Évora, tal como noutras ocasiões em que precisei de apoio mais detalhado sobre certas temáticas. Nos restantes dias o estágio foi realizado remotamente.

3 Estado da arte

No âmbito do trabalho de estágio foi necessário realizar análise de entre o leque de ferramentas propostas, quais iriam ser utilizadas, sendo elas Puppet [2], Hiera [3], r10k [4] e PuppetDB [5]. Foi feita então a decisão da não inclusão da ferramenta PuppetDB [5] e a inclusão das outras tecnologias que se revelaram totalmente pertinentes, visto que cada uma delas iria ter um papel crucial no desenvolvimento do projeto tendo em conta o objetivo do mesmo. A PuppetDB é um serviço open source [1] para guardar dados gerados pelo Puppet. Foi concebido para reforçar o uso do Puppet com o aumento da performance em mente, guardando toda a informação assíncronamente, libertando o Puppet Master [6] para compilar mais catálogos em simultâneo, garantindo também a integridade dos dados guardados na própria PuppetDB.

Apesar da revisão de todas as vantagens em utilizar PuppetDB, foi decidido neste projeto a não utilização da ferramenta, visto que não se iria justificar a introdução de uma nova peça com este conjunto de funcionalidades.

4 Ambiente de desenvolvimento

4.1 Ambiente técnico

4.1.1 Hardware

Na realização deste estágio foram utilizados apenas computadores pessoais, quer o portátil presencialmente, quer o fixo remotamente. Ambas as máquinas utilizam o sistema operativo Ubuntu [7].

Foi necessária também a utilização de duas Virtual Machines, ambas com 8GB de memória RAM no mínimo para poderem suportar o serviço Puppet[2] nos nodes, quer do Puppet Master[6], quer do Puppet Agent[8].

4.1.2 Software

Ao longo do estágio utilizei várias ferramentas de software:

- slack;
- github;
- Linux;
- Ubuntu;
- Puppet;
- Facter;
- Hiera;
- r10k;

Slack [9] é uma plataforma de comunicação empresarial proprietária. O Slack oferece muitos recursos do estilo do IRC (Internet Relay Chat), incluindo salas de chat persistentes organizadas por tópico, grupos privados e mensagens diretas. Esta plataforma é utilizada pelos membros da Angulo Sólido para comunicarem entre si, através de texto e de chamadas de áudio e/ou vídeo, o que permite que possam esclarecer dúvidas entre si e discutir as melhores soluções para os projetos.

Github [10] é uma plataforma de hospedagem de código-fonte e arquivos com controlo de versões usando Git [11]. Permitindo que os programadores ou qualquer usuário registado na plataforma contribuam em projetos privados e/ou Open Source [1] a partir de qualquer canto do mundo. A plataforma de hospedagem foi responsável por hospedar o meu repositório do projeto de estágio que contém toda a documentação que utilizei para me guiar durante o processo.

Linux [12] é o termo vulgarmente utilizado para referir os sistemas operativos que utilizam o Kernel Linux. O Kernel foi desenvolvido pelo programador finlandês Linus Torvalds, inspirado no sistema Minix. Base dos sistemas operativos usado durante a realização do projeto de estágio.

Ubuntu [7] é um sistema operativo de código aberto [1], construído a partir do *kernel* Linux [12], baseado na distribuição Debian e utiliza o GNOME como ambiente de trabalho e é desenvolvido pela Canonical.

Puppet [2] é uma ferramenta de código aberto [1] para gestão de configurações cuja ideia principal é que se tenha a configuração centralizada num ponto, para que seja distribuída entre os diversos nós de uma determinada rede.

Puppet Server [6]: O Puppet [2] é configurado numa arquitetura agente-servidor, em que o servidor primário (Puppet Master) gere as configurações de todos os agent nodes. O Puppet Server atua como o servidor primário, sendo ele uma aplicação que corre numa Java Virtual Machine (JVM) e tem como objetivo compilar os catálogos [13] e determinados ficheiros.

Facter [14] uma ferramenta que retorna factos sobre cada agente, como o seu hostname, o endereço IP, o sistema operativo e a sua versão, juntamente com muitos outros dados, que são recolhidos a partir do momento em que o agente entra em execução. Os factos são enviados para o Puppet Master [6], que automaticamente são transformados em variáveis para o Puppet [2]. Caso seja necessário visualizar os factos disponíveis, basta correr o *facter* binary na linha de comandos. Cada facto é retornado como um par `key => value`. Por exemplo:

```
operatingsystem => Ubuntu
```

```
ipaddress => 10.0.0.10
```

Quando combinados com a configuração definida no Puppet, estes factos permitem personalizar as configurações para cada host, como por exemplo, criar recursos genéricos como definições de network e personalizar com dados de cada agente, ou até mesmo escolher de acordo com o sistema operativo em uso que comandos usar para instalar um determinado package, como por exemplo, se o Facter verificar que se trata de um sistema Ubuntu, utilizar o comando `aptitude`, se for Red Hat usar `yum`, etc.

Hiera [3]: Um dos pontos fortes do Puppet é o facto de o código ser reutilizável. Um código que serve várias necessidades tem que ser configurável: colocar informação específica numa configuração externa, em vez de estar presente no código fonte. O Puppet utiliza o Hiera em duas vertentes, sendo elas guardar os dados das configurações em pares `key-value` e procurar que dados é que um certo módulo precisa para aplicar a um certo node durante a compilação de um catálogo. Estas duas operações são feitas, respetivamente, através de Automatic Parameter Lookup para classes incluídas num catálogo e Explicit lookup calls. A hierarquia de lookups do Hiera funciona através de um padrão de "defaults, with overrides", ou seja, se a solução default não funcionar temos uma hierarquia de alternativas. O Hiera usa os factos do Puppet [2] para especificar as data sources, para que seja possível existir um override na infraestrutura em causa. Esta ferramenta funciona como se fosse uma função, que recebe argumentos de cada um dos nodes e age em função dos mesmos.

r10k [4] é uma ferramenta que permite gerir as configurações de um determinado ambiente. As alterações no código são feitas através da linha de comandos do r10k no servidor primário. Com base no código nos branches dos repositórios de controlo, o r10k cria ambientes no server primário para aplicar em cada node. No fundo, é bastante útil para não termos que aplicar a totalidade do código num determinado node mas sim dividi-lo em branches e aplicá-los em nodes separadamente.

4.1.3 Ambiente aplicacional

Este sistema é composto por dois ambientes distintos:

- *Puppet Master* [6];
- *Puppet Agent* [8].

Estes dois ambientes interagem da seguinte forma: o Puppet Master [6] controla toda a informação relativa a uma ou mais configurações, já do outro lado, cada node Puppet Agent [8] faz um pedido ao Puppet Master [6] para que este lhe aplique uma determinada configuração.

4.2 Metodologia de trabalho

Na realização do projeto não foi desenvolvido um software, logo, não existiu um processo de desenvolvimento e consequentemente gestão do mesmo.

Embora isto, foi feito o devido planeamento de todo o trabalho a desenvolver, dividindo-o em fases distintas para se trabalhar ao longo do estágio, e sempre que fora necessário procedeu-se à alteração do mesmo.

Uma vez que não existiu equipa de trabalho, interagi apenas com o Eng. Pedro Pessoa.

5 Trabalho desenvolvido

No início do estágio comecei por ter uma reunião com o CEO da empresa, Eng. Pedro Pessoa, na qual me foram apresentadas as diferentes etapas para a realização do estágio:

Na primeira etapa do projeto, foi-me proposto fazer um resumo acerca do que é o Puppet [2] em si e as principais diferenças entre cada nova versão do mesmo, funções deprecated e alterações a serem feitas no código, uma vez que o principal objetivo do estágio era atualizar a infraestrutura da outrora atual versão 3 para a versão 7. Além desta distinção entre versões foi também feita a pesquisa intensa e anotação das principais características de ferramentas dentro do próprio Puppet, tais como r10k [4], Facter [14] e Hiera [3]. Foi feito também o mesmo trabalho para guiar a instalação de um ambiente em Puppet 7 e para um eventual upgrade de uma infraestrutura Puppet 3 para Puppet 7.

No que diz respeito à segunda etapa, acompanhado da respetiva documentação, a instalação de um ambiente Puppet Master [6]-Puppet Agent [8] na versão 7, adicionando as ferramentas r10k [4] e Hiera [3] e finalmente testar a sua execução com módulos do próprio Puppet [2].

5.1 Descrição detalhada

5.1.1 Repositório e documentação

Numa primeira fase do estágio, ficou acordado começar por resumir todo o trabalho escrevendo toda a documentação necessária para a realização do mesmo, e para esse efeito criei um repositório na plataforma Github [10] em que coloquei toda a informação que fui obtendo e registando em formato Markdown [15]. Realizei este mesmo processo para a uma parte introdutória sobre o Puppet [2] e as suas ferramentas, para instalação de um sistema em Puppet7, para instalação do r10k [4], o porquê de não ter optado pelo uso de PuppetDB [5] e finalmente para o processo de upgrade de versões do Puppet, neste caso da versão 3 para a versão 7.

5.1.2 Instalação de um Puppet Master e de um Puppet Agent em diferentes VM's

Esta etapa do projeto foi realizada inteiramente a operar em duas Virtual Machines, uma que iria conter o node do Puppet Master [6] e outra para conter o node de um Puppet Agent [8].

Comecei por fazer a devida instalação do Puppet Master [6] numa das máquinas com os respetivos comandos de instalação para um sistema Ubuntu [7]. Depois procedi a instalar o respetivo Puppet Agent [8] na outra máquina da mesma forma, sempre a consultar a respetiva documentação de apoio.

Para colocar estes dois nodes a funcionar em conjunto foi necessária uma pequena configuração em ambos os lados do ambiente Puppet [2] que se dividiu em 3 etapas. Inicialmente foi necessário alterar a variável PATH do lado do Agente devido ao diretório em que estão inseridos os comandos Puppet [2] não coincidir com a mesma por default (após a instalação do Agente). Em seguida, foi necessário alterar a setting server do Agente para coincidir com o IP da máquina do Puppet Master [6]. Finalmente, para completar a conexão Master-Agent, foi necessário validar o certificado deste Agente, e para isso teve que ser gerado um novo certificado do lado do Agente que posteriormente teve que ser validado pelo Master.

Após o sistema estar devidamente instalado e em pleno funcionamento, procedi à instalação da ferramenta r10k no Puppet Master [6], que funciona através de um repositório, neste caso criado no Github [10], que contém todas configurações feitas dentro do próprio r10k [4] em que cada branch corresponde a um diferente ambiente de produção. Ferramenta esta (r10k) que posteriormente trabalhou em conjunto com o Hiera [3].

Comecei por testar o seu funcionamento com a inserção do módulo NTP [16] de modo a alterar certas configurações no agente, ou seja, foi feita uma alteração num branch com os dados deste módulo e após o update dos mesmos para o repositório foi feito um deployment destas alterações ainda no Puppet Master [6]. Para concluir este teste, do lado do Puppet Agent [8] é executado um comando que tem como argumento o nome do ambiente (corresponde ao nome do branch criado no repositório) que queremos executar e podemos após alguns segundos verificar as alterações que foram feitas na própria consola do Agente.

Todo este processo irá estar presente no capítulo seguinte com o guia completo para a instalação desta infraestrutura.

5.1.3 Guia para a instalação do Puppet7 de raiz

Para utilizar o Puppet, há que fazer uma instalação inicial e um processo de setup dos seus componentes. O Puppet está distribuído em vários packages, incluindo puppet server, puppet agent e puppetdb. O Puppet Server controla toda a informação das configurações de um ou mais agent nodes (equivalente ao Puppet Master). Esta instalação é dividida em diversos passos:

- Instalação do repositório da plataforma Puppet
- Instalação do Puppet Server
- Instalação de um Puppet Agent

A partir da instalação destes componentes separadamente, podemos começar a personalizar as configurações para cada node e instalar outros componentes à medida que a nossa infraestrutura vai crescendo.

Os comandos de instalação apresentados mais à frente irão ser apenas do tipo apt e funcionais apenas em sistemas Ubuntu.

5.1.3.1 Instalação do repositório da plataforma Puppet

A partir de um login do utilizador em root, fazer o download do package e correr o dpkg em install mode:

```
sudo apt-get update  
sudo dpkg -i <FILENAME>.deb
```

Utilizando por exemplo, um repositório focal:

```
wget https://apt.puppet.com/puppet7-release-focal.deb  
sudo dpkg -i puppet7-release-focal.deb
```

E em seguida dar update à lista de packages apt:

```
sudo apt-get update
```

5.1.3.2 Instalação do Puppet Server

Antes de instalar:

Antes da instalação, há que rever os sistemas operativos suportados. Ao contrário do Puppet Agent, o Puppet Server não é suportado para MacOS, mas sim em Red Hat Enterprise Linux 6, 7, 8, Debian 9 (Stretch), 10 (Buster) e Ubuntu 16.04 (Xenial, amd64 only). É necessário certificar também que o sistema possui pelo menos 2GB de RAM disponíveis.

Instalação:

Agora sim, proceder à instalação do package em si através do seguinte comando:

```
sudo apt-get install puppetserver
```

Iniciar o serviço Puppet Server, executando:

```
sudo systemctl start puppetserver
```

Verificar se o serviço está instalado corretamente:

```
puppetserver -v
```

5.1.3.3 Instalação e configuração dos Puppet Agents

Instalação do Agente:

Para instalar um agente basta executar o seguinte comando na consola:

```
sudo apt-get install puppet-agent
```

Em seguida, iniciar o serviço Puppet:

```
sudo /opt/puppetlabs/bin/puppet resource service puppet ensure=running enable=true
```

Configuração:

A configuração dos agentes divide-se em 3 etapas:

1. Configurar o PATH para o acesso aos comandos Puppet: Os comandos Puppet estão localizados no diretório bin (/opt/puppetlabs/bin/). Este diretório não é a variável PATH por default , ou seja, vamos ter que adicionar o diretório bin ao PATH, através do seguinte comando:

```
export PATH=/opt/puppetlabs/bin:$PATH
```

Esta operação também pode ser feita através da alteração do PATH nos ficheiros de configuração .profile ou .bashrc.

2. Configuração do server setting:

Esta etapa permitirá definir a setting server = puppetserver.example.com em puppet.conf:

```
sudo -i puppet config set server puppetserver.example.com --section main
```

Além desta operação é possível alterar outras definições como por exemplo:

```
serverport , ca_server , ca_port , report_server e report_port .
```

3. Conectar o agent ao Server primário e validar o certificado:

Depois de adicionado o Server, é necessário conectar o Puppet agent ao Server primário para que este verifique em intervalos regulares o estado do mesmo, retornando o respetivo catálogo e atualizar a configuração, se necessário.

Para conectar o agente ao Server primário, executar:

```
sudo -i puppet ssl bootstrap
```

Irá aparecer uma mensagem do tipo:

```
Info: Creating a new RSA SSL key for <agent node>
```

No node do server primário, validar o certificado:

```
sudo -i puppetserver ca sign --certname <name>
```

No node do agente, executar de novo o agente:

```
sudo -i puppet ssl bootstrap
```

Para testar a ligação entre master e agente:

```
sudo -i puppet agent -t
```

5.1.4 Guia para a instalação do r10k

Instalação do r10k via Ruby Gems:

```
/opt/puppetlabs/puppet/bin/gem install r10k
```

Configurar o r10k criando a seguinte estrutura de diretórios e o ficheiro `/etc/puppetlabs/r10k/r10k.yaml` assegurando que contém a seguinte informação:

```
# The location to use for storing cached Git repos
:cachedir: '/var/cache/r10k'
# A list of git repositories to create
:sources:
  # This will clone the git repository and instantiate an environment per
  # branch in /etc/puppetlabs/code/environments
  :my-org:
    remote: 'git@github.com:$_Insert GitHub Organization Here_$_Insert GitHub Repository Here'
    basedir: '/etc/puppetlabs/code/environments'
```

Configurar também o ficheiro `/etc/puppetlabs/puppet/hiera.yaml` com os seguintes conteúdos:

```
-----
# Hiera 5 Global configuration file

version: 5

defaults:
  datadir: /etc/puppetlabs/code/environments/%{environment}/hieradata/
  data_hash: yaml_data

hierarchy:
  - name: "Common"
    paths:
      - "common.yaml"
```

5.1.4.1 Configurar o Repositório para o Puppet Code utilizando o módulo NTP

Popular o repositório clonando o mesmo localmente e executando cada uma das seguintes ações:

Ter em conta que o environment default do puppet é production. É recomendado alterar o git branch default de master para production de modo a corresponder. Em alternativa, é possível também definir o environment do agente para master.

```
mkdir -p {modules,site/profile/manifests,hieradata}
touch hieradata/common.yaml
touch environment.conf
touch Puppetfile
touch site.pp
```

Editar `environment.conf` e garantir que tem os seguintes conteúdos:

```
manifest = site.pp
modulepath = modules:site
```

Editar `site.pp` e garantir que tem os seguintes conteúdos:

```
node agent1.local {
  include ntp
}

class { [ 'ntp' ]:
```

Editar `hieradata/common.yaml` e garantir que tem os seguintes conteúdos:

```
ntp::servers:
  - 0.europe.pool.ntp.org
  - 1.europe.pool.ntp.org
  - 2.europe.pool.ntp.org
  - 3.europe.pool.ntp.org
```

Editar Puppetfile e garantir que tem os seguintes conteúdos:

```
forge 'forge.puppetlabs.com'

# Forge Modules
mod 'puppetlabs-ntp', '9.1.0'
mod 'puppetlabs/stdlib'
```

Garantir que o `r10k` corre de maneira a aceder ao repositório `git`. Pode-se testar o acesso ao mesmo usando `sudo git clone yourrepoURL`.

5.1.4.2 Resumindo

Temos agora as seguintes peças a funcionar:

1. Puppet Master
2. Hiera
3. `r10k`
4. Repositório para o Puppet Code
5. 'profile' inicial com o nome 'base' que irá configurar o NTP nos agentes. Esta base irá servir para fazer variadas funções. A parte mais interessante é mesmo a capacidade de utilizar Git branches para ajudar a gerir a infraestrutura como parte do ciclo de vida do desenvolvimento de software.

Agora, quando for necessário testar um novo profile, podemos fazer o seguinte:

- Criar um novo branch no repositório
- Criar código Puppet neste novo branch
- Push deste novo branch para o repositório
- Deploy como um novo environment usando:

```
sudo /opt/puppetlabs/puppet/bin/r10k deploy environment -p
```

A partir de qualquer agente (incluindo o master), podemos correr o teste a este novo environment, especificando-o na linha de comandos. Por exemplo, se a nova branch tiver o nome `teste`, utilizar o seguinte comando:

```
sudo -i puppet agent -t --environment teste
```

Como foi dito anteriormente, podemos também modificar `/etc/puppetlabs/puppet/puppet.conf` num node e adicionar um environment predefinido:

```
...
[agent]
environment = teste
```

5.1.5 Guia para um eventual upgrade dos Agents Puppet 3 para Puppet 7

5.1.5.1 Puppet 3 para Puppet 5

Apesar de terem havido várias mudanças no Puppet desde a versão 3.8, o processo de upgrade dos agentes do Puppet 3 pode ser automatizado ao contrário dos seus servidores. O módulo puppet agent executa automaticamente as seguintes tarefas:

- Instalação do repositório Puppet, caso não esteja presente
- Instala a versão mais recente do package puppet agent, que substitui as versões instaladas do Puppet, Facter e Hiera
- Copia os ficheiros SSL do Puppet para a nova localização
- Copia os ficheiros puppet.conf antigos para a nova localização e limpa settings antigos que tenham sido removidos no Puppet 4 ou que necessitem de um reset para os seus valores default
- Garante que o serviço Puppet está a funcionar corretamente

5.1.5.2 Puppet 5 para Puppet 7

Para dar upgrade aos puppet agents, apenas é necessário adicionar as duas keys que se seguem ao global Hiera e esperar por dois puppet-runs:

```
profile::puppet::collection: 'puppet7'

puppet_agent::package_version: 'auto'
```

6 Avaliação crítica

Durante o período de duração deste estágio, estive em contacto com uma área que nunca tinha interagido e sempre tive curiosidade em perceber como era trabalhada, principalmente por estar um pouco distante dos conhecimentos que fui desenvolvendo ao longo da licenciatura em Eng. Informática e por isso quis complementar um pouco mais o que aprendi com algo diferente.

Achei particularmente interessante o facto de que a maior parte das ações se fazem a partir da linha de comandos e não através do desenvolvimento de código num editor, o que confesso que me senti à vontade para fazer.

Inicialmente, achei um pouco aborrecida a fase da pesquisa e escrita da documentação, mas fiquei a perceber que se trata de uma parte essencial do trabalho assim que pus em prática o que tinha escrito, pois já estava tudo previamente resumido detalhadamente, o que tornou o trabalho muito mais fácil.

Apesar de ter tido algumas dificuldades nos processos de instalação e deployment do software, acho que todos esses obstáculos foram bastante úteis para ganhar conhecimento de como resolver certos problemas que vão surgindo à medida que se desenvolvem projetos deste género e tenho a certeza que vai ser útil no futuro.

Em suma, foi um projeto de estágio em que na minha opinião ganhei uma certa experiência do que é trabalhar na área de Engenharia Informática e me deu ainda mais vontade de continuar a ganhar mais e mais conhecimento para poder prosseguir uma carreira que seja o melhor possível dentro desta profissão.

Referências

- [1] Open source. https://pt.wikipedia.org/wiki/C%C3%B3digo_aberto.
- [2] Puppet. <https://puppet.com/>.
- [3] Hiera. https://puppet.com/docs/puppet/7/hiera_intro.html.
- [4] r10k. <https://puppet.com/docs/pe/2019.8/r10k.html>.
- [5] Puppetdb. <https://puppet.com/docs/puppetdb/7/overview.html>.
- [6] Puppet master. https://puppet.com/docs/puppet/7/server/about_server.html.
- [7] Ubuntu. <https://ubuntu.com/>.
- [8] Puppet agent. https://puppet.com/docs/puppet/7/install_agents.html.
- [9] Slack. <https://slack.com/>.
- [10] Github. <https://github.com/>.
- [11] Git. <https://git-scm.com/>.
- [12] Linux. <https://linux.org/>.
- [13] Catalog. https://puppet.com/docs/puppet/7/subsystem_catalog_compilation.html.
- [14] Facter. <https://puppet.com/docs/puppet/7/facter.html>.
- [15] Markdown. <https://www.markdownguide.org/>.
- [16] Ntp. <https://forge.puppet.com/modules/puppetlabs/ntp>.