



**Centro Universitário de Brasília  
Instituto CEUB de Pesquisa e Desenvolvimento - ICPD**

**CARLOS HENRIQUE BARBOSA RODRIGUES**

**GERENCIAMENTO DE CONFIGURAÇÕES COM PUPPET:  
GARANTINDO UMA INFRAESTRUTURA ÁGIL, ESTÁVEL E SEGURA**

**Brasília  
2017**

**CARLOS HENRIQUE BARBOSA RODRIGUES**

**GERENCIAMENTO DE CONFIGURAÇÕES COM PUPPET:  
GARANTINDO UMA INFRAESTRUTURA ÁGIL, ESTÁVEL E SEGURA**

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB/ICPD) como pré-requisito para obtenção de Certificado de Conclusão de Curso de Pós-graduação em Rede de Computadores com Ênfase em Segurança.

**Orientador:** Prof. Msc Marco Antônio de Oliveira Araujo

**Brasília  
2017**

**CARLOS HENRIQUE BARBOSA RODRIGUES**

**GERENCIAMENTO DE CONFIGURAÇÕES COM PUPPET:  
GARANTINDO UMA INFRAESTRUTURA ÁGIL, ESTÁVEL E SEGURA**

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB/ICPD) como pré-requisito para obtenção de Certificado de Conclusão de Curso de Pós-graduação em Rede de Computadores com Ênfase em Segurança.

**Orientador:** Prof. Msc Marco Antônio de Oliveira Araujo

Brasília, \_\_\_\_ de \_\_\_\_\_ de 2017.

**Banca Examinadora**

---

Professor:

---

Professor:

Dedico este trabalho a minha esposa Mariely Silva Faria e minha filha Luiza Faria Barbosa por compreender e entender que para concluir este trabalho a minha ausência em alguns momentos era necessária.

## **AGRADECIMENTOS**

O este trabalho se tornou real devido o auxílio e ajuda de várias pessoas. Sendo assim agradeço:

- A Deus, por me sustentar e me conceder forças para chegar até aqui;
- À Mariely, minha esposa por estar ao meu lado em todos os momentos;
- À minha filha Luiza, por recarregar minhas energias através do seu sorriso;
- À minha mãe Celita, que sempre me mostrou a importância dos estudos;
- Ao João Regino, amigo de trabalho que sempre me tira dúvidas;
- Por fim, aos professores do Curso que compartilharam seus conhecimentos.

## RESUMO

À rápida evolução da tecnologia computacional, a popularização do acesso à internet e a evolução da computação em nuvem levou ao aumento da demanda por processamento e disponibilidade de recursos computacionais. Com o constante crescimento dos DataCenters e aumento na quantidade de aplicações desenvolvidas o gerenciamento manual da infraestrutura se tornou inviável, o que fez com que aflorasse o desenvolvimento de ferramentas de gerenciamento de configuração. Este trabalho apresenta o uso da ferramenta de gerenciamento de configurações Puppet, demonstrando a capacidade que a mesma tem de manter o parque de servidores padronizados, executar instalações “*deploy*” de forma automatizada, executar tarefas em larga escala e manter a sua infraestrutura ágil, atualizada e segura. O objetivo principal é demonstrar que o Puppet permite criar servidores, gerir versionamentos, instalar sistemas, alterar configurações, corrigir erros e desastres automaticamente, em larga escala envolvendo centenas ou milhares de servidores e com uma intervenção mínima.

**Palavras-chave:** Gerenciamento de Configurações. Puppet. Infraestrutura. Computação em Nuvem.

## **ABSTRACT**

The fast advance on computer technology, the popularization of the internet access and the evolution of the cloud computing led to the increase of the demand for processing and computational resources. With this regular growth of the DataCenters and the rise on the number of the applications developed, the manual management of the infrastructure became unfeasible, making the development of configuration tools surface. This work presents the use of the management tool on Puppet configuration, showing the capacity that it has to keep the server park patronized, executing “deploy” installations in and automated way, do tasks in large scale and keep its infrastructure agile, updated and secure. The main objective is to show that the Puppet allows it to create servers, manage versioning, install systems, change configurations, correct errors and disasters automatically in large scale, involving hundreds or thousands of servers with one minimum intervention.

**Key words:** Configuration Management. Puppet. Infrastructure. Clouding Computing.

## SUMÁRIO

<b>INTRODUÇÃO</b> .....	09
<b>OBJETIVOS</b> .....	10
<b>METODOLOGIA</b> .....	11
<b>1 INFRAESTRUTURA ÁGIL</b> .....	12
<b>1.1 Automação</b> .....	12
<b>1.2 Entrega</b> .....	13
<b>1.3 Métricas</b> .....	13
<b>2 DEVOPS</b> .....	14
<b>2.1 História</b> .....	14
<b>2.2 O que é DevOps</b> .....	14
<b>2.3 DevOps e seus Benefícios</b> .....	15
<b>3 GERENCIAMENTO DE CONFIGURAÇÕES</b> .....	17
<b>3.1 Principais Benefícios do Gerenciamento de Configurações</b> .....	17
<i>3.1.1 Recuperação rápida de desastres</i> .....	17
<i>3.1.2 Provisionamento de servidores ou substituição de um existente</i> .....	18
<i>3.1.3 Replicação de tarefas em múltiplos servidores</i> .....	18
<b>4 SEGURANÇA DA INFORMAÇÃO</b> .....	19
<b>4.1 Princípios da Segurança</b> .....	19
<i>4.1.1 Gerenciador de Configurações e os Princípios da Segurança</i> .....	20
<b>5 PUPPET</b> .....	21
<b>5.1 Arquitetura do PUPPET</b> .....	21
<b>5.2 Fluxo entre o Agent e o Master</b> .....	22
<b>5.3 Ambientes</b> .....	23
<b>5.4 Módulos</b> .....	23
<b>6 PROJETO</b> .....	24
<b>6.1 Cenário Anterior</b> .....	24
<i>6.1.1 Servidores sem o NTP devidamente configurado</i> .....	25
<i>6.1.2 Usuários criados de forma manual nos servidores</i> .....	25
<i>6.1.3 Servidores sem o DNS configurado</i> .....	25
<i>6.1.4 Serviços essenciais não configurados</i> .....	26

6.1.5 Serviços essenciais fora da inicialização do Sistema Operacional	26
6.1.6 Configuração do agente de backup e monitoramento nos nodes	27
<b>6.2 Solução Proposta</b>	<b>28</b>
6.2.1 Implantar um Gerenciados de Configurações PUPPET	28
<b>7 IMPLANTAÇÃO</b>	<b>30</b>
<b>7.1 Instalação e Configuração do PUPPET</b>	<b>30</b>
7.1.1 Definição da arquitetura	30
7.1.2 Topologia do PUPPET	30
7.1.3 Recursos alocados e softwares utilizados	31
7.1.4 Instalação do PUPPETSERVER e FOREMAN	31
<b>7.2 Administração do Gerenciador de Configurações PUPPET</b>	<b>32</b>
7.2.1 Criação dos ambientes	32
7.2.2 Criação dos grupos de hosts	33
7.2.3 Instalação e configuração dos módulos	34
7.2.4 Instalação e configuração dos agentes nos nodes	35
<b>8 RESULTADOS</b>	<b>36</b>
<b>9 CONCLUSÃO</b>	<b>39</b>
<b>REFERÊNCIAS</b>	<b>40</b>
<b>ANEXO I</b>	<b>41</b>
<b>A. INSTALAÇÃO E CONFIGURAÇÃO DO PUPPET E FOREMAN</b>	<b>41</b>
A.1 Criando a Máquina e Configurando no DNS	41
A.2 Atualizando o sistema operacional e configurando o hostname	41
A.3 Desabilitando o IPTABLES e SELINUX	41
A.4 Configurar Repositórios e instalar o Puppetmaster e Foreman	42
A.5 Configurando o Banco de dados do Puppet e Foreman no ambiente de produção	43
A.5.1 Configurando os parâmetros do foreman	43
<b>ANEXO II</b>	<b>41</b>
<b>B. ADMINISTRAR O GERENCIADOR DE CONFIG PUPPET</b>	<b>45</b>
B.1 Criando Ambientes	45
B.2 Criando Grupos de Hosts	47
B.3 Instalando e Configurando Módulos	49
B.3.1 Instalando os módulos no puppet master	49

<b>B.3.2 Configurando os módulos no FOREMAN.....</b>	<b>49</b>
<b>B.4 Instalando e Configurando um módulo.....</b>	<b>51</b>
<b>B.4.1 Adicionando um Node (host).....</b>	<b>51</b>
<b>B.4.2 Assinando um certificado e adicionando node no FOREMAN.....</b>	<b>51</b>

## INTRODUÇÃO

Á rápida evolução da tecnologia computacional, a criação do Cloud Computing e a popularização do acesso à internet levou ao aumento da demanda por processamento e conseqüentemente o crescimento dos DataCenters. As empresas passaram a conviver com ambientes virtualizados, onde poucas máquinas físicas tornaram-se parques com centenas ou milhares de máquinas virtuais.

Frente a este novo cenário a velocidade da informação ficou diretamente ligado ao poder de venda e processos das empresas, exigindo das mesmas o desenvolvimento contínuo de softwares ágeis, e atualizações constantes. Para atender a quantidade crescente de usuários e suas necessidades as empresas tem como estratégia de negócio a adição de recursos *"features"*, o que ocasiona maior demanda de recursos e maiores entregas da equipe de infraestrutura com prazos extremamente curtos.

Com desenvolvedores buscando aprimorar suas aplicações, disponibilizando novas versões com novos recursos para seus clientes a equipe de infraestrutura percebeu a real necessidade de abandonar o seu modelo tradicional de administração, onde as tarefas são executadas manualmente ou através de scripts de forma reativa e passe a atuar de forma proativa, uma vez que novos deploys e procedimentos de hollback envolvem equipes de infra.

A equipe de infraestrutura, composta em sua grande maioria por sysadmins, tem como objetivo principal manter os sistemas em pleno funcionamento, é sua responsabilidade manter o ambiente de produção intacto. Segundo Sarath Pillai (2012), O trabalho de um administrador de sistema divide-se em tarefas rotineiras tais como: configuração, implantação e manutenção de servidores. E tarefas mais particulares como: analisar o funcionamento e performance das aplicações, executar procedimentos de hollback e sugerir melhorias para que serviços se mantenham rápidos e estáveis. Essas tarefas estão diretamente ligadas ao ponto vital do negócio, o acordo de nível de serviço. O produto fora do ar acarreta a insatisfação do cliente e conseqüentemente prejuízos financeiros e institucionais.

A necessidade de disponibilizar e duplicar ambientes, executar mudanças em códigos de aplicações redundantes, atualizações de segurança em vários

servidores, manter os arquivos de configurações padronizados fez nascer o conceito de infraestrutura ágil que é: Pensar na infraestrutura como código, automatizando a infraestrutura, criando ambiente de testes de forma dinâmica, controle de versões, criação de ambientes temporários.

Este trabalho acadêmico visa apresentar a ferramenta de gerenciamento de configurações Puppet como uma poderosa ferramenta capaz de manter as configurações dos servidores sobre controle, controlar versionamentos de códigos, executar deploys e atualizações em larga escala, de forma automática e dinâmica.

## **OBJETIVOS**

### **Objetivo Geral**

Instalar e configura uma ferramenta de gerenciamento de configurações que seja capaz orquestrar o ambiente de forma automática e dinâmica.

### **Objetivo Específico**

O presente estudo acadêmico visa:

- Manter, através do gerenciador de configurações Puppet as configurações dos servidores sobre controle.
- Demonstrar que o Puppet é capaz de garantir as configurações default dos serviços.
- Executar versionamento de códigos através da ferramenta.
- Demonstrar o quão seguro e estável fica um ambiente que possui um gerenciamento de configurações em produção.

## **METODOLOGIA**

Para atingir o objetivo proposto neste trabalho foram realizadas pesquisas, execução de tarefas para instalação e configuração de um gerenciador de configurações Puppet, obedecendo a seguinte ordem:

- Pesquisas sobre o tema infraestrutura ágil e gerenciamento de configurações.
- Criação de um ambiente composto por clientes (NODES).

- Instalação e configuração de uma ferramenta de gerenciamento de configurações Puppet.
- Criação de módulos que aplicarão as mudanças no nodes conectados ao gerenciador de configurações Puppet.
- Validação da ferramenta demonstrando sua importância na segurança, estabilidade e agilidade do ambiente.

No ANEXO I, constara a descrição dos passos realizados para instalação do serviço PUPPET e FOREMAN.

No ANEXO II, constará a descrição dos passos realizados na configuração do PUPPET, dos módulo e instalação do cliente.

## 1. INFRAESTRUTURA ÁGIL

O termo Infraestrutura Ágil surgiu e ganhou grande atenção com o crescimento de duas tecnologias: virtualização e computação em nuvem. A virtualização fez com que pequenos DataCenters com poucas máquinas físicas se transformassem em parques com centenas ou milhares de servidores. Em paralelo a computação em nuvem “*cloud computing*” fez com que servidores virtuais se tornassem em milhares de instâncias.

Infraestrutura Ágil é um guia de adoção de boas práticas criado e organizado por um coletivo de profissionais da área de TI envolvendo atividades como automação e utilização de diversos métodos ágeis. Infraestrutura ágil é tornar sua infra mais eficiente e autônoma e para isso ela se apoia nos seguintes eixos: automação, entrega, métricas e pessoas (CARVALHO, 2013).

Para atingir o objetivo proposto pelo conceito da Infraestrutura Ágil o processo de automação torna-se parte obrigatória e para alcançá-lo o uso de ferramentas de gerenciamento de configuração é indispensável.

### 1.1 Automação

A palavra automação origina do latim *Automatus* que significa “mover-se por si”, ou seja, é fazer com que um equipamento seja capaz de, através de métodos computacionais ou mecânicos, realizar seu controle automaticamente.

O que define um sistema automático é a capacidade que o mesmo tem de verificar seu próprio funcionamento, executando correções e atualizações sem a interferência humana.

Na área de tecnologia da informação o autor Rafael Brito Gomes no 21º Seminário RNP de Capacitação e Inovação afirma que:

Automação é a possibilidade de tratar a infraestrutura como código, onde é possível automatizar a construção de ambiente, colocar novas aplicações em produção, modificar configurações de diversos ativos ao mesmo tempo. Tudo sem intervenção manual repetitiva[...].

## **1.2 Entrega**

Dentro da Infraestrutura Ágil a entrega consiste em disponibilizar o software de forma automatizada através de ferramenta específicas. O processo de entrega tem que ser capaz de disponibilizar a aplicação com segurança, em curto prazo e com baixo risco.

## **1.3 Métricas**

As métricas são as responsáveis para mensurar o funcionamento, performance e prazos. A grande quantidade de aplicações e tendências que surgem nos ambientes corporativos faz com que a Infraestrutura Ágil esteja em constante evolução, logo, as métricas geram insumos para que mudanças sejam realizadas dentro de uma Infraestrutura Ágil.

## 2. DEVOPS

O aumento do consumo por tecnologia aumentou a demanda e com isso a desconexão entre as atividades e/ou profissionais de desenvolvimento com as de operação se tornou mais evidente. Esta desconexão passou a gerar ineficiência e conflitos o que fez nascer a cultura DevOps.

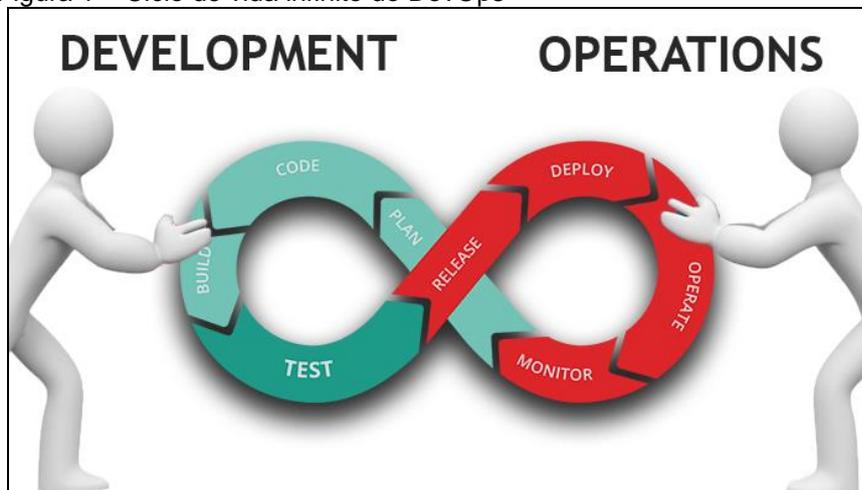
### 2.1. História

Em 2008, na O’reilly Velocity Conference dois funcionários da Flickr John Allspaw e Paul Hammond apresentaram uma palestra abordando o tema Dev versus Ops onde os palestrados argumentavam que a única maneira de construir, testar e implantar um novo software e através da integração entre desenvolvedores, Sysadmins e Analistas de Infraestrutura. Ao ver a palestra o desenvolvedor Patrick Debois resolve criar, em 2009, uma conferência com o nome Devopsdays, o que logo após daria origem ao termo DevOps.

### 2.2. O que é DevOps

DevOps é um conjunto de práticas que surgem através da colaboração entre profissionais de desenvolvimento e operação em todos os estágios do ciclo de vida do serviço, desde o processo de criação, homologação, testes até o suporte de produção. A cultura DevOps busca derrubar muros que são levantados pelo famoso enigma “Não é o meu código, são suas máquinas” e vice-versa.

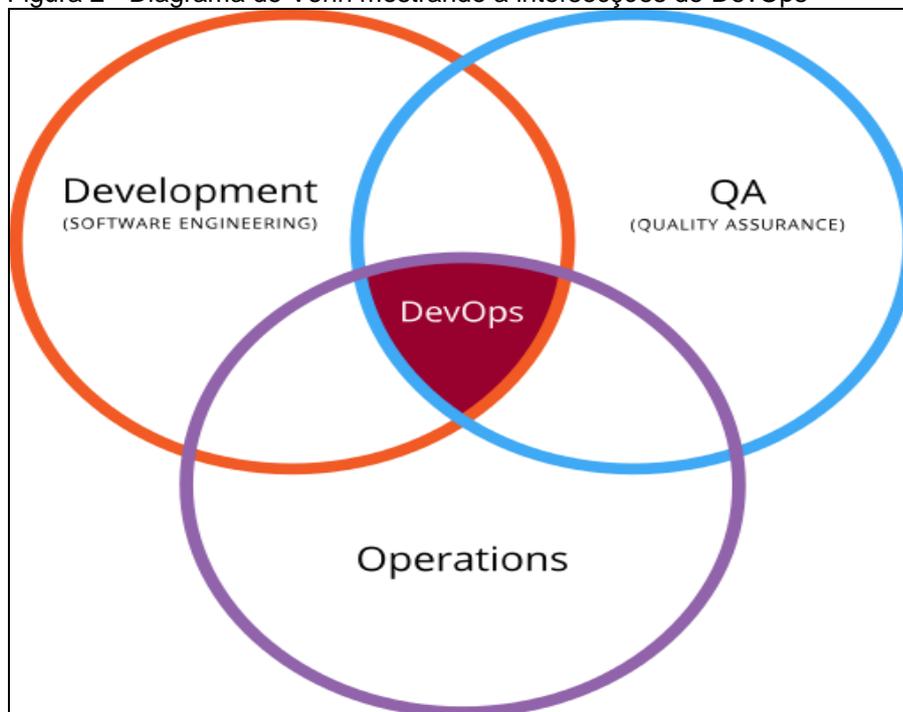
Figura 1 – Ciclo de vida infinito do DevOps



Fonte – <http://www.agilebuddha.com/agile/x-htm/>

Segundo Humble (2016, p. 4), DevOps é uma comunidade interdisciplinar dedicada ao estudo da construção, evolução e operação de sistemas capaz de enfrentar rápidas mudanças e em larga escala.

Figura 2 - Diagrama de Venn mostrando a intersecções do DevOps



Fonte – <https://en.wikipedia.org/wiki/DevOps>

### 2.3. DevOps e seus Benefícios

Com uma equipe reduzida de profissionais multifuncionais e o DevOps implementado a organização é capaz de oferecer maior velocidade, funcionalidade e inovação em seus produtos. O DevOps traz uma série de benefícios como:

- Fornecimento de atualizações de aplicações com menos complexidade e com uma resolução de falhas mais rápida.
- Mudança no processo linear de desenvolvimento e entrega do produto onde uma equipe finalizava toda a tarefa antes de passar para outra equipe. Com o DevOps a construção e implantação dos sistemas tornaram-se flexíveis e dinâmicas.
- Eliminação de barreiras entre as equipes de forma harmônica com maior envolvimento dos funcionários tornando-os mais colaboradores em todo o processo e com isso maiores oportunidades profissionais.

- Entregas mais rápidas de ambientes, recursos computacionais.
- Respostas mais rápidas para atender as demandas dos clientes e as mudanças propostas pelo mercado.

### 3. GERENCIAMENTO DE CONFIGURAÇÕES

O conceito de gerenciamento de configurações é antigo, porém aliado ao conceito de infraestrutura ágil é algo novo. Adam Bertram (2015), define o gerenciamento de configurações como um processo de estabelecer e manter a consistência de desempenho de um produto, garantindo os seus atributos ao longo de sua vida.

Todos os dias os profissionais da infraestrutura praticam o gerenciamento de configuração ainda que não tenha uma ferramenta de gerenciamento de configuração, seja instalando uma imagem de Sistema Operacional, aplicando uma GPO ou criando scripts para execução de tarefas rotineiras.

Ainda que o conceito de Gerenciamento de configurações não tenha nascido na TI o termo tem sido usado amplamente para se referir à gestão de configuração de servidores. A gestão de configuração é um mecanismo usado para fazer o servidor atingir o estado desejável, ou seja, a automação se torna o coração do gerenciamento de configuração para servidores, e por isso o termo orquestração também pode ser usado para definir o gerenciamento de configurações.

#### 3.1. Principais Benefícios do Gerenciamento de Configurações

Ainda que se trate de empresas de pequeno, médio ou grande porte um Gerenciador de Configurações devidamente implementado na infraestrutura traz uma série de benefícios.

##### 3.1.1. Recuperação rápida de desastres

Como o parque de servidores tem suas configurações estabelecidas por uma ferramenta de gerenciamento de configurações você consegue voltar as configurações de um serviço de forma rápida. Isto permite o restabelecimento de serviços que apresentaram falhas desconhecidas.

Podemos citar também falhas que surgem por alteração em arquivos de configuração por parte dos desenvolvedores, analistas e outros. Com o gerenciamento de configurações a recuperação “*howback*” é feito de forma rápida e confiável.

### *3.1.2. Provisionamento de novos Servidores ou substituição de um existente.*

Com um Gerenciador de Configurações devidamente configurado é possível automatizar a criação de novos servidores. A automação deste processo é eficiente e precisa uma vez que os programas e suas configurações já foram determinados em templates do gerenciador de configurações.

Quando um servidor fica indisponível o processo de descoberta da falha ou causa raiz pode demorar horas. Neste cenário a criação de um servidor para substituir o que apresentou o erro é a maneira mais rápida de devolver a disponibilidade do serviço.

### *3.1.3. Replicação de tarefas em múltiplos servidores.*

Ajustes de configurações, atualizações de software e correção de bugs pode ser feito em massa com o uso de uma ferramenta de Gerenciamento de Configurações. A alteração será feita apenas no Gerenciador e este fara a replicação para todos os servidores “nodes” conectados a ele.

A replicação não se restringe a configurações de serviços ou atualizações. Com o gerenciamento de configurações você pode replicar ambientes idênticos. Isso permite você replicar um ambiente de produção para realizar testes ou validar atualizações.

#### 4. SEGURANÇA DA INFORMAÇÃO

A segurança da informação é a proteção da informação contra vários tipos de ameaças para garantir a continuidade do negócio, minimizar o risco ao negócio, maximizar o retorno sobre os investimentos e as oportunidades de negócio e é obtida a partir da implementação de um conjunto de controles adequados, incluindo políticas, processos, procedimentos, estruturas organizacionais e funções de software e hardware (ABNT, 2005).

As informações digitais são consideradas críticas para a grande maioria das empresas e são de suma importância para a concretização de negócios e tomada de decisões. Para as organizações, uma falha de segurança pode gerar prejuízos incalculáveis. Imaginem dados pessoais de clientes ou informações de determinada empresa nas mãos da concorrência, se forem copiadas, totalmente apagadas ou se não puderem ser acessadas. Hoje, invadir sistemas não é algo muito complexo, já que os sistemas de informação em sua grande maioria estão conectados a redes externas (FERREIRA, 2003).

##### 4.1. Princípios da Segurança

Os princípios básicos da segurança da informação são a confidencialidade, a integridade e a disponibilidade. Desrespeitar um desses princípios em algum momento, representa uma quebra de segurança da informação, o que também pode ser chamado de incidente de segurança da informação (CAMPOS, 2006).

Figura 3 – Pilares da segurança da informação



Fonte – <https://debsolutionsti.com/iso-27000/iso-27000/>

#### 4.1.1. Gerenciador de Configurações e os Princípios da Segurança

Uma ferramenta de gerenciamento de configurações atua diretamente na correção de um incidente de segurança da informação seja ele quebra de confidencialidade, integridade e disponibilidade.

**Integridade** – Toda informação deve continuar com as mesmas condições em que foi disponibilizada, não sofrendo alterações indevidas. Um gerenciador de configurações não previne diretamente que o dado não seja alterado, porém, ele atua na correção desta quebra de integridade sendo capaz de voltar o dado no seu estado original.

**Disponibilidade** – O dado ou a informação criada deve ficar disponível aos seus usuários. Quando um sistema fica indisponível por qualquer motivo há quebra deste princípio. Toda informação deve continuar com as mesmas condições em que foi disponibilizada, não sofrendo alterações indevidas. O gerenciador de configurações e sua capacidade de replicar programas, servidores e ambientes inteiros o torna indispensável na correção da quebra de disponibilidade forma rápida e eficiente.

## 5. PUPPET

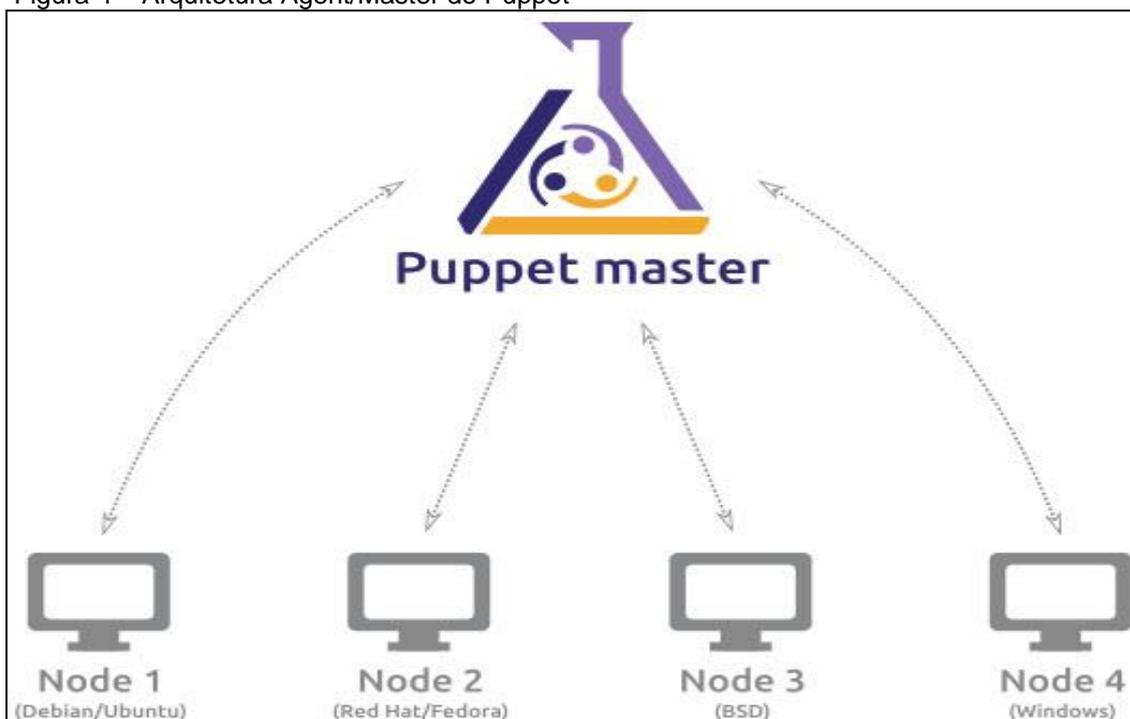
O Puppet é um sistema de gerenciamento de configurações baseado na linguagem de programação Ruby, disponível nas versões Open Source e Enterprise, que trabalha baseado no modelo cliente-servidor. O Puppet visa disponibilizar aos SysAdmins um sistema de gerenciamento de configurações consistente, transparente, flexível e de configuração simplificada.

A ferramenta funciona de maneira modular, ou seja, é possível adicionar ou remover vários recursos sem a necessidade de reinstalação do software. Os módulos podem automatizar tarefas tais como a criação de usuários, banco de dados, servidor Web, servidor de e-mails, configuração de arquivos, DNS, dentre outros.

### 5.1. Arquitetura do PUPPET

O Puppet trabalha com o modelo máster/cliente. Onde o “*Puppet Master*” é o servidor central e os “*nodes*” são os clientes conectados ao máster. A configuração é definida no máster, compilada e aplicada nos clientes a partir do momento que os mesmos estão conectados ao Puppet (TURNBULL, 2007 p. 4).

Figura 4 – Arquitetura Agent/Master do Puppet

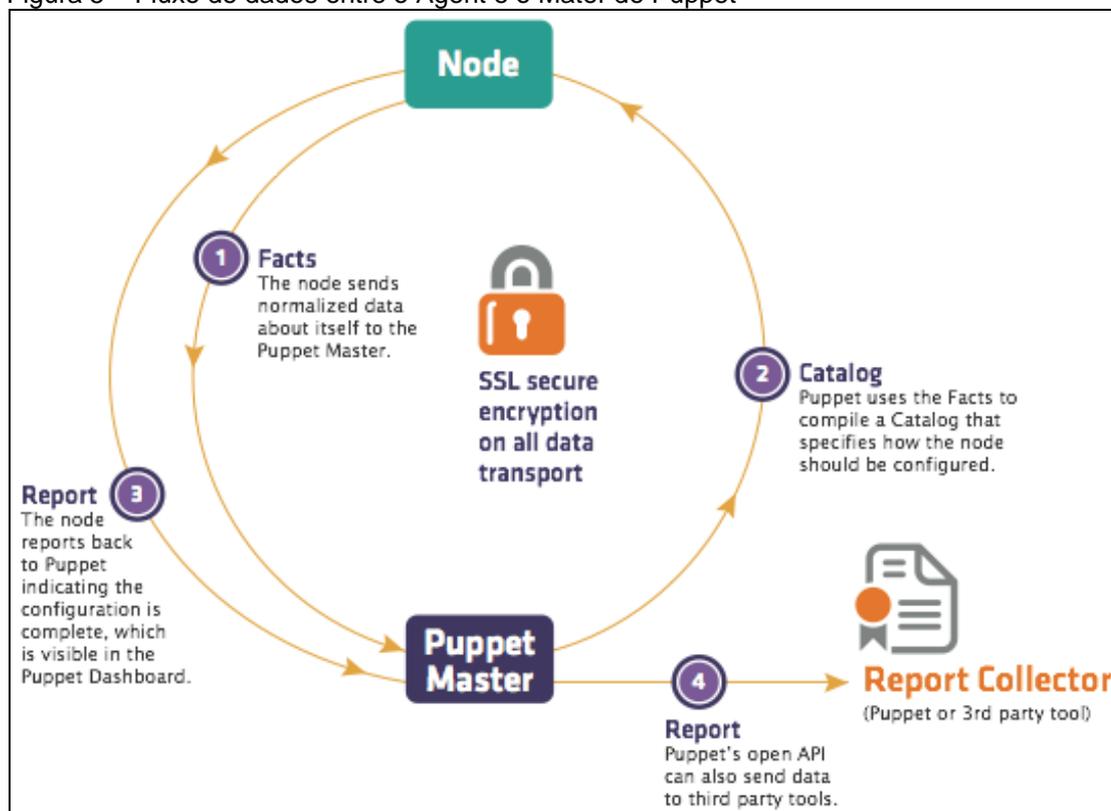


Fonte – <https://docs.puppet.com/puppet/>

## 5.2. Fluxo entre o Agent e o Master

Neste item iremos detalhar como interagem o agent “node” e o máster e o fluxo que é seguido pelos mesmos.

Figura 5 – Fluxo de dados entre o Agent e o Mater do Puppet



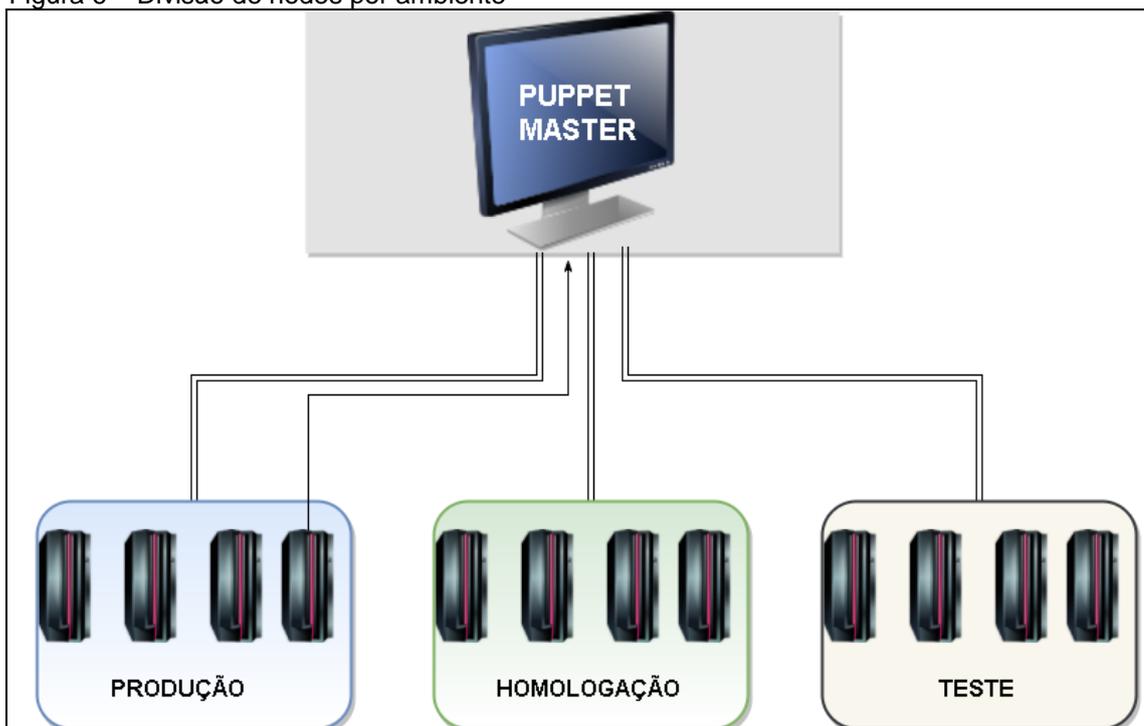
Fonte – <http://www.aosabook.org/en/puppet.html>

1. FACTS - Ao se conectar no máster o cliente (node) envia sua configuração atual (facts), o servidor máster verifica as configurações a serem aplicadas.
2. CATALOG - O puppet máster criará um catálogo com base nas informações recebidas. Neste catálogo irá constar as tarefas e mudanças que devem ser realizadas pelo cliente (node).
3. REPORT – Após o agente aplicar as mudanças descritas no catálogo o agente constrói um relatório e envia ao puppet máster.
4. REPORT – O puppet pode enviar dados e relatórios para aplicações externas.

### 5.3. Ambientes

O Puppet permite você trabalhar com ambientes isolados (teste, homologação, produção). Isso permite você usar versões diferentes de módulos para ambientes distintos. Aplicar mudanças em um ambiente apenas. E viabiliza apenas a configuração de um Puppet máster para tender todos os ambientes.

Figura 6 – Divisão de nodes por ambiente



Fonte – Próprio autor

### 5.4. Módulos

Os módulos no Puppet são de códigos e/ou dados que serão recebidos pelo node para aplica-los. É possível utilizar módulos de terceiros ou escrever seus próprios módulos. Podemos, por exemplo, desenvolver um módulo HTTP, que seja capaz de instalar o servidor apache, fazer as configurações determinadas no módulo, colocar o serviço na inicialização e verificar se o serviço está em execução.

O puppet forge é um repositório que disponibiliza inúmeros módulos para diferentes situações, serviços e cenários. O comando `puppet module` é o responsável por pesquisar e instalar um módulo disponível no repositório <http://forge.puppetlabs.com/>. Este comando também permite atualizar ou remover um módulo instalado no servidor puppet máster.

## 6. PROJETO

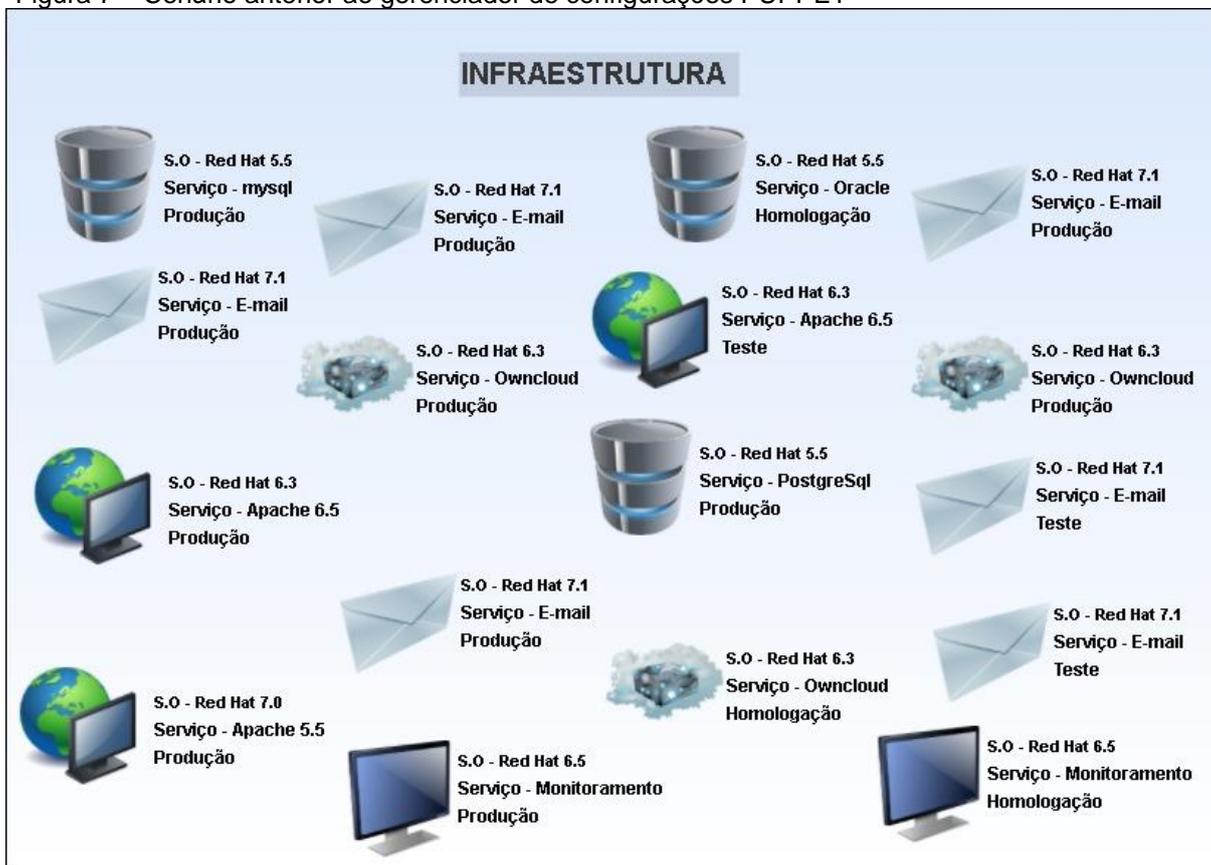
Neste capítulo abordaremos o cenário anterior ao Puppet fazendo um levantamento dos problemas existentes por não ter uma ferramenta de gerenciamento de configurações e as soluções esperadas com a configuração da ferramenta.

### 6.1. Cenário Anterior

A instituição onde foi configurada o gerenciador de configurações Puppet está presente em todo território nacional sendo representada por 27 Superintendências e mais de 550 pontos de atendimento, sua sede está situada em Brasília. Tem como missão garantir a segurança com cidadania nas rodovias federais.

Sua infraestrutura atualmente conta com mais de trinta servidores físicos e aproximadamente trezentos servidores virtuais de divididos entre produção, teste, desenvolvimento e homologação.

Figura 7 – Cenário anterior ao gerenciador de configurações PUPPET



Fonte – Próprio autor

### 6.1.1. Servidores sem o NTP devidamente configurado

A Infraestrutura tem um servidor NTP (Network Time Protocol) devidamente configurado, tendo sua hora sincronizada com os servidor NTP oficial do Brasil que é o do observatório nacional, porém foi identificado que os servidores clientes não estavam tendo sua hora sincronizada com o servidor NTP local seja por não ter o serviço ntpdate instalado ou por causa do arquivo de configuração *ntp.conf* mal configurado.

Está falha que parece ser simples acarretou tais problemas como: falhas no balanceamento de cargas; erro na sincronia entre os servidores DNS máster e slave; falhas na replicação das configurações de sistemas que tem alta disponibilidade; inconsistência no horário de envio de mensagens de e-mail com os logs do servidor SMTP; certificados expirados por causa da data e hora divergente.

### 6.1.2. Usuários criados de forma manual nos servidores

Sempre que a Infraestrutura recebia um novo colaborador era necessário criar um usuário em todos os servidores que o colaborador necessitava acesso. Como exemplo podemos citar que um novo colaborador necessitará de acesso em 10 servidores. Eram criados 10 usuários, configuradas 10 permissões, 10 mapeamentos de rede. Com o crescimento exponencial de servidores virtuais e uma rotatividade considerável de funcionários, a criação de usuários de forma manual ficou totalmente inviável.

Quando ocorria o desligamento do colaborador os usuários eram removidos dos servidores, porém, detectávamos que o usuário ainda existia em alguns servidores, seja por descuido SysAdmin ou por falta de informação em quais servidores o colaborador desligado tinha acesso.

### 6.1.3. Servidores sem o DNS configurado

A Infraestrutura possui um servidor DNS máster e um slave, ambos devidamente configurados. Porém falhas na resolução de nome ocorria com frequência pois o arquivo *resolv.conf*, responsável por configurar o DNS que será usado pelos servidores era constantemente alterado e com isso os endereços

internos não eram resolvidos. Erros em serviços como: Apache; Squid e NTP eram comuns em servidores que sofriam essa alteração no arquivo. Ao investigar percebíamos que os arquivos eram alterados para efeito de testes e possíveis falhas porém não eram reconfigurados novamente.

A ausência de um gerenciador de configurações também trouxe um grande trabalho manual durante a mudança de endereço do servidor DNS pois o arquivo *resolv.conf* teve que ser alterado de forma pontual em cada servidor.

Figura 8 – Exemplo do arquivo resolv.conf

```
[root@vm04 ~]# cat /etc/resolv.conf
domain tcc
nameserver 10.11.12.13
nameserver 10.11.12.14
```

Fonte – Próprio autor

#### 6.1.4. Serviços essenciais não configurados

Detectamos que os profissionais da TI gastava um tempo considerável ferramentas de trabalho pois a quantidade de servidores administrados pelos mesmos é grande. Notamos que os Sysadmins ao trabalhar em uma máquina tinham que instalar editores, serviço de FTP, navegadores, compactadores. Os DBA's instalavam ferramentas de administração de SGBD, análise de performance. Os analistas de rede e segurança instalavam frequentemente as ferramentas telnet, traceroute, tcpdump.

#### 6.1.5. Serviços essenciais fora da inicialização do Sistema Operacional

Durante a janela de atualização, manobra ou reconfiguração dos virtualizadores e storages da Infraestrutura alguns servidores eram desligados conforme mapeamento prévio, porém ao religar estes servidores alguns serviços não subiam por simplesmente não terem configurado os mesmos na inicialização do Sistema Operacional. Durante as janelas de manutenção tal falha trazia um retrabalho excessivo pois era necessário checar, através das ferramentas de monitoramento ou de forma pontual, os serviços que possivelmente não eram inicializados.

Figura 9 – Exemplo de serviços configurados na inicialização do S.O

```
[root@smtpl1 ~]# chkconfig --list
Galaxy          0:não  1:não  2:não  3:sím  4:não  5:sím  6:não
auditd          0:não  1:não  2:sím  3:sím  4:sím  5:sím  6:não
blk-availability 0:não  1:sím  2:sím  3:sím  4:sím  5:sím  6:sím
cgconfig        0:não  1:não  2:não  3:não  4:não  5:não  6:não
cgred           0:não  1:não  2:não  3:não  4:não  5:não  6:não
crond           0:não  1:não  2:sím  3:sím  4:sím  5:sím  6:não
fail2ban        0:não  1:não  2:sím  3:sím  4:sím  5:sím  6:não
ip6tables       0:não  1:não  2:sím  3:sím  4:sím  5:sím  6:não
ipset           0:não  1:não  2:sím  3:sím  4:sím  5:sím  6:não
iptables        0:não  1:não  2:sím  3:sím  4:sím  5:sím  6:não
lvm2-monitor    0:não  1:sím  2:sím  3:sím  4:sím  5:sím  6:não
munin-node      0:não  1:não  2:sím  3:sím  4:sím  5:sím  6:não
```

Fonte – Próprio autor

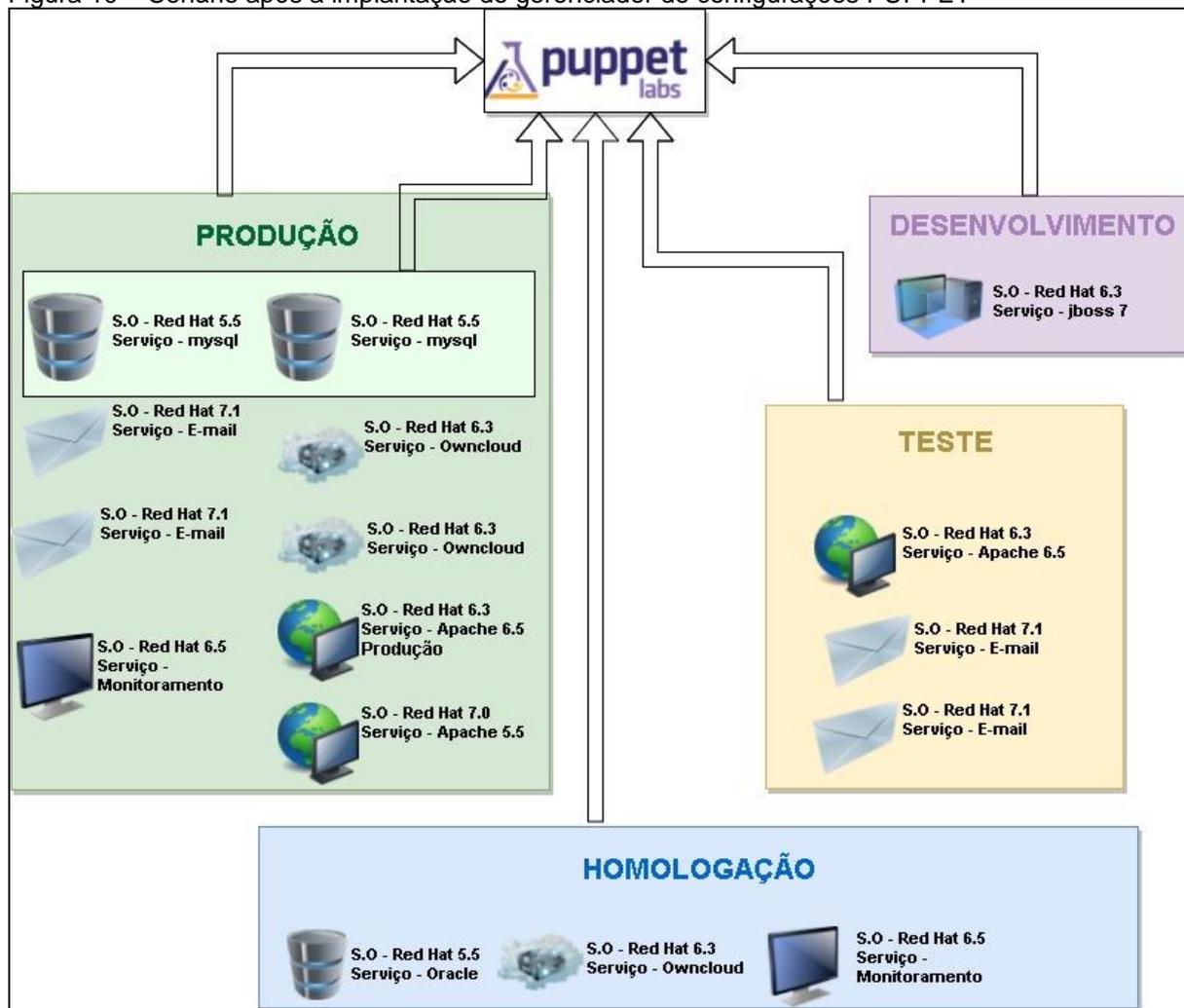
### 6.1.6. Configuração do agente de backup e monitoramento nos Servidores.

Quando surgiu a demanda de implementar a ferramenta de backup e monitoramento exigiu um esforço muito grande para desenvolver um script capaz de instalar e configurar os agentes de backup e de monitoramento nos servidores clientes. Após esta primeira etapa da configuração a configuração de novas máquinas passou a ser manual o que acarretou dois pontos relevantes. O primeiro é o tempo gasto uma vez que a criação de novos servidores virtuais é constante e o segundo é que por ser um processo manual alguns servidores não tem os agentes instalados. Tal falha compromete dois princípios básicos da segurança da informação, integridade e disponibilidade.

## 6.2. Solução Proposta

Este capítulo tem por objetivo apresentar a solução proposta para resolução dos problemas citados no item 6.1 e seus subitens e os motivos para escolha da mesma.

Figura 10 – Cenário após a implantação do gerenciador de configurações PUPPET



Fonte – Próprio autor

### 6.2.1. Implantar um gerenciador de configurações PUPPET

Ao chegar à conclusão que era necessário implantar uma ferramenta para gerenciar as configurações o passo seguinte foi a escolha da ferramenta. Devido à grande atenção dada ao tema Gerência de Configuração e DevOps várias ferramentas foram criadas. A escolha pela ferramenta Puppet se deu pelos seguintes motivos:

- Teve sua criação dentro dos conceitos básicos da “Gerência de Configuração” e por isso é uma ferramenta idempotente, convergente e sua gerência é feita através de agentes instalados nos clientes (nodes).
- Não depende de outros serviços como SSH para funcionar pois utiliza o seu próprio agente para realizar a comunicação entre o cliente e o máster.
- Como a Infraestrutura tem mais de 300 servidores outra preocupação foi quanto ao throuput (quantidade de dados transferidos na rede) que podia ser gerado na rede. O Puppet se mostrou o ideal pois o máster nunca se conecta nos nodes, apenas os nodes se conectam ao servidor o que dificilmente elevará o throuput da rede.
- Facilidade na configuração de regra no firewall pois para haver comunicação basta o node conectar a porta 8140 TCP do máster.
- Toda comunicação é criptografada.
- Facilidade de identificar qual sistema operacional e qual versão está rodando no servidor cliente.
- Capacidade de se integrar com o The Foreman que é uma ferramenta para gerenciamento de ciclo de vida dos servidores, provisionamento orquestração e monitoramento.
- Boa documentação e a versão free atende ambientes com até 1000 servidores o que permite um planejamento a longo prazo.

## 7. IMPLANTAÇÃO

Detalharemos neste capítulo os passos seguidos para implantação do Puppet e The Foreman, ferramentas escolhidas para solucionar os problemas e dificuldades levantados nos capítulos anteriores.

### 7.1. Instalação e Configuração do PUPPET

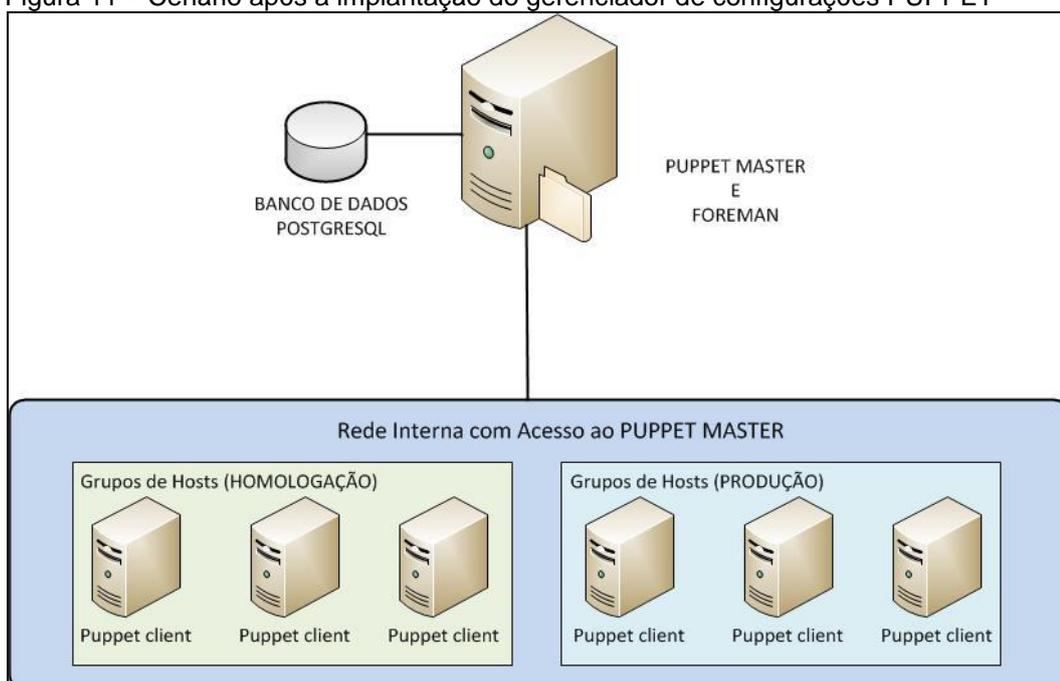
#### 7.1.1. Definição da arquitetura

A arquitetura para o gerenciamento de configurações através do PUPPET ficou definida da seguinte forma:

- Um servidor Master, responsável pela sincronização de todos os clientes da infraestrutura que estão conectados ao máster.
- A interface web de gerenciamento do PUPPET será o FOREMAN que será instalado no servidor PUPPET máster.
- O banco de dados utilizado como repositório dos dados será o PostgreSQL.

#### 7.1.2. Topologia do PUPPET

Figura 11 – Cenário após a implantação do gerenciador de configurações PUPPET



Fonte – Próprio autor

### 7.1.3. Recursos alocados e softwares utilizados

O Hardware disponibilizado foi o recomendado para suportar ambientes com até 1000 nodes, conforme descrito no manual oficial do fabricante. Os recursos computacionais e softwares utilizados foram:

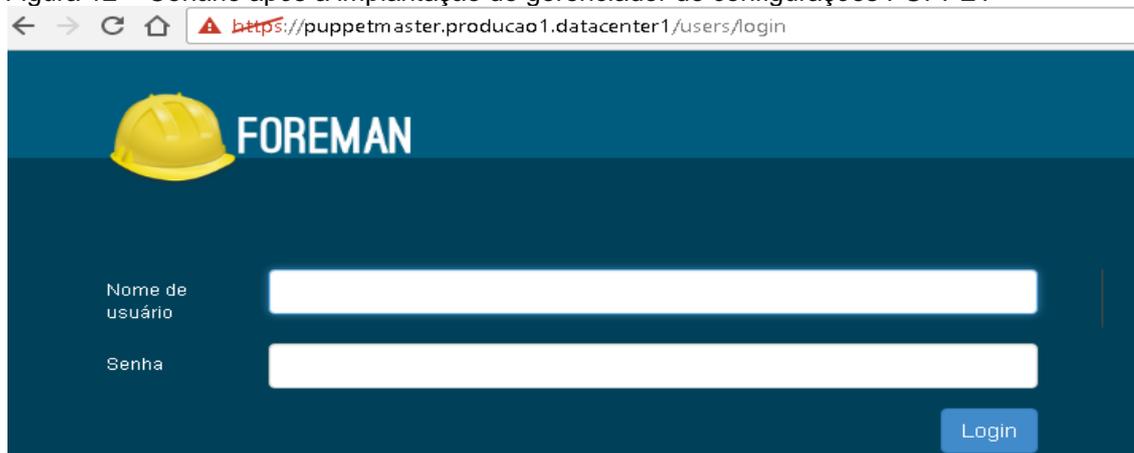
- 4 processadores x64
- 16 GB de memória RAM
- 50 GB de disco
- 1 Interface de rede Gigabit
- S.O. Oracle Linux 7
- Puppetserver 3.8
- Foreman 1.8

### 7.1.4. Instalação do PUPPETSERVER e FOREMAN

Após a definição do ambiente, recursos alocados e software utilizados foi dado início a instalação do PUPPETSERVER e FOREMAN. A instalação é simples uma vez que os pré-requisitos foram atendidos nos itens descritos no capítulo 7.1.3.

Como usamos a distribuição Oracle Linux que é baseada no CentOS foi possível realizar a instalação a partir do repositório oficial através do comando yum. O processo de instalação está detalhado no ANEXO I. Após a instalação a página de login no Foreman estará disponível conforme a figura 12.

Figura 12 – Cenário após a implantação do gerenciador de configurações PUPPET



Fonte – Próprio autor

## 7.2. Administração do Gerenciador de Configurações PUPPET

O Puppet máster será administrado através da interface web do FOREMAN. Nos subcapítulos do 7.2 iremos abordar a criação de ambientes, grupos de hosts e assinatura de certificados dos nodes.

### 7.2.1. Criação dos ambientes

Foi definido que os servidores que integram a infraestrutura do Datacenter serão divididos nos ambientes: produção, homologação, desenvolvimento e teste.

Figura 13 – Ambientes no puppet



Fonte – Próprio autor

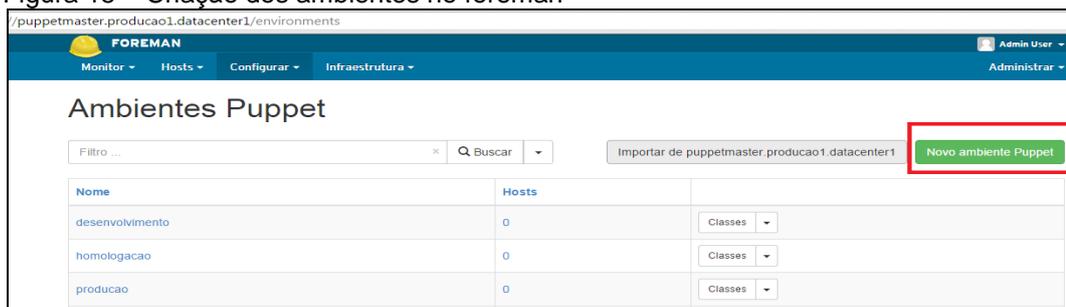
A criação dos ambientes consiste em dois passos. O primeiro é a criação dos diretórios no servidor puppet master, local onde serão armazenados os arquivos de configuração dos módulos (Figura 14) e o segundo passo é a criação do ambiente no FOREMAN (Figura 15). O detalhamento dos passos constam no ANEXO II.

Figura 14 – Criação de diretório do ambiente de teste no puppet

```
[root@puppetmaster ~]# cd /etc/puppet/environments/
[root@puppetmaster environments]# cp -rfvp example_env/ teste
```

Fonte – Próprio autor

Figura 15 – Criação dos ambientes no foreman



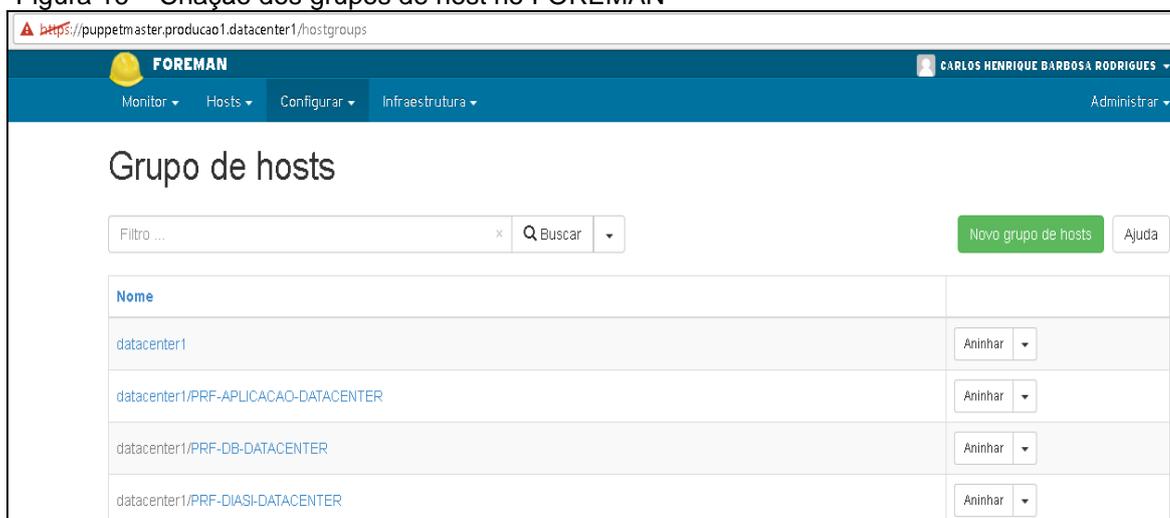
Fonte – Próprio autor

### 7.2.2. Criação dos grupos de hosts

O uso de grupo de hosts viabiliza o gerenciamento das configurações de uma forma mais específica. Como temos servidores de diversas áreas como banco de dados, aplicação, infra em um mesmo ambiente este recurso permitirá dividir os nodes por área, departamento, usuários dentre outros. Foi definido que os nomes dos grupos de host seguirão o padrão: **EMPRESA-DEPARTAMENTO-AMBIENTE**

Exemplo: **PRF-APLICACAO-PRODUCAO**

Figura 16 – Criação dos grupos de host no FOREMAN



Fonte – Próprio autor

O detalhamento dos passos para criação dos grupos de hosts está descrito no ANEXO II.

### 7.2.3. Instalação e configuração dos módulos

O manual oficial do PUPPET descreve o módulo como sendo um conjunto de configurações e dados, independentes e reutilizáveis, contidos no mesmo diretório. Na prática é a forma mais adequada de manter o seu Gerenciador de Configurações organizado uma vez que ele é usado para gerenciar e configurar diversos sistemas e serviços.

Os módulos são portáteis, não sendo necessário a existência de um Puppetmaster para instalar o mesmo, podendo sua instalação ser feita através do comando `#puppet apply` no node. Na versão usada, 3.8, os módulos são configurados por padrão no diretório `/etc/puppet/environments`.

Existem duas formas de se trabalhar com módulo são:

1. Instalando um módulo existente no repositório oficial do puppet <https://forge.puppet.com/> usando o comando `#puppet module install <nome do modulo>`
2. Criando um novo módulo através do comando `#puppet module generate <nome do modulo>`

Após a instalação ou criação de um módulo é necessário importa-lo para o FOREMAN determinando em qual ambiente o módulo estará disponível, conforme ilustrado na figura 17. O detalhamento dos passos para instalação e configuração de módulos constam no ANEXO II.

Figura 17 – Importando módulos para o FOREMAN

The screenshot shows the FOREMAN web interface. The page title is 'Ambientes Puppet'. There is a search bar and a button labeled 'Importar de puppetmaster.producao1.datacenter1' which is highlighted with a red box. Below the search bar is a table with the following data:

Nome	Hosts	
desenvolvimento	0	Classes ▾
homologacao	0	Classes ▾
producao	0	Classes ▾
production	1	Classes ▾
teste	0	Classes ▾

Displaying all 5 entries

Fonte – Próprio autor

#### 7.2.4. Instalação e configuração dos Agentes nos clientes (nodes).

Nodes são todos os servidores que tem sua gerencia feita pelo Puppet Master. Para criar uma conexão cliente-servidor é necessário instalar o serviço puppet no cliente, esta instalação pode ser feita via instalador de pacotes do sistema operacional e logo após a instalação é necessário assinar o certificado que foi gerado no servidor máster. A assinatura do certificado pode ser feita através da interface web do FOREMAN (Figura 18).

Figura 18 – Instalando o puppet no cliente.

```
Para os Sistemas Operacionais derivados do RHEL
#yum install puppet

Para os Sistemas Operacionais derivados do Debian
#apt-get install puppet
```

Fonte – Próprio autor

O node realiza a conexão com o servidor máster através da porta 8140 e por padrão a cada 30 minutos ele busca novas atualizações no servidor máster (Figura 19). Os passo para instalação e configuração do puppet no cliente estão descritos no ANEXO II.

Figura 19 – Assinando um certificado no FOREMAN.

The screenshot shows the FOREMAN web interface. The main heading is "PuppetCA em puppetmaster.producao1.datacenter1". Below the heading is a "Filter by state:" dropdown menu. To the right of the filter is a red-bordered button labeled "Auto-assinar entradas". Below this is a table with the following data:

Nome do certificado	Estado	Válido a partir de	Expira	Identidade	
puppetmaster.producao1.datacenter1	válido	13 dias atrás	em quase 5 anos	SHA256	Apagar

At the bottom of the table area, it says "Displaying 1 entry".

Fonte – Próprio autor

## 8. RESULTADOS

A implementação do Puppet e sua devida configuração tornou viável a solução dos problemas identificados no capítulo 6.1 e seus subcapítulos. As melhorias do Puppet não se restringiram aos problemas levantados. A configuração do Gerenciador Puppet possibilitou melhorias imediatas como também viabilizou um processo de padronização contínuo.

A instalação do Puppet na infraestrutura da organização permitiu identificar de forma rápida em qual ambiente o servidor se encontra conforme mostra a figura 20. Antes do puppet só era possível após logar no servidor e verificar em qual rede a mesma pertencia. Esta melhoria permitiu uma resposta rápida à equipe de gestão de TI ao questionarem a quantidade de servidores por ambiente.

Figura 20 – Hosts separados por ambiente.

### Ambientes Puppet

Nome	Hosts
desenvolvimento	13
homologacao	60
producao	132
production	5
teste	23
treinamento	7

Fonte – Próprio autor

A instalação do módulo NTP, capaz de instalar, configurar e executar o serviço de ntp, fez com que todos os *nodes* tivessem sua hora ajustada com o servidor de hora da infraestrutura uma vez que ele força o arquivo de configuração ntp.conf usar o endereço do servidor.

Foi desenvolvido um módulo, chamado chutils, que instala os serviços padrões utilizados no dia a dia pelos técnicos tais como: tcpdump, traceroute, lynx, unzip, nmap, dentre outros. O módulo foi feito através de variáveis o que permite adicionar ou remover serviços a qualquer momento.

O módulo `resolv_conf` que foi criado com o objetivo de fazer com que os servidores clientes passem a usar os servidores DNS configurados na nossa infraestrutura. Desta forma, ao sofrer alteração, o arquivo volta a configuração determinada pelo puppet máster assim que o cliente for executado como mostra a figura 21.

Figura 21 – Arquivo `resolv.conf` configurado pelo Puppet Master.

```
[root@vm07 ~]# cat /etc/resolv.conf
# File managed by puppet

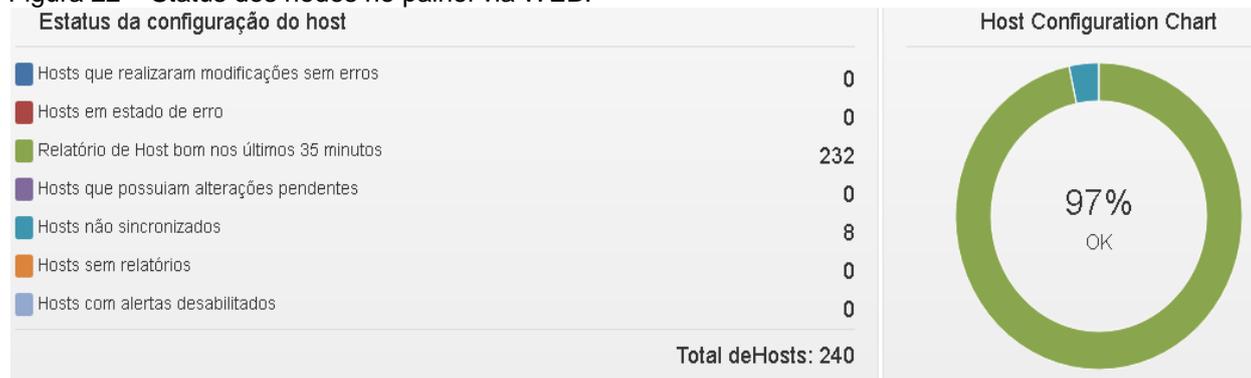
domain prf
nameserver 10.0.11.1
nameserver 10.0.102.1
```

Fonte – Próprio autor

O módulo `startonboot` foi criado com o objetivo de colocar os serviços essenciais como cliente de backup, monitoramento e puppet para iniciarem durante a inicialização do Sistema Operacional. O módulo `yumrepos`, instalado a partir do repositório oficial do puppet, faz com que os arquivos com a extensão `.repo` sejam configurados pelo puppet máster, assim todos os servidores fazem a instalação dos pacotes através dos repositórios locais configurados na infraestrutura.

Outro grande benefício foi o dashboard fornecido pelo FOREMAN pois permite verificar o status dos nodes nos últimos 35 minutos. Através do painel podemos verificar se os clientes estão apresentando erro durante a sincronização, se sofreram alteração, se foram atualizados e se a conectividade cliente-servidor está ativa, conforme mostra a figura 22.

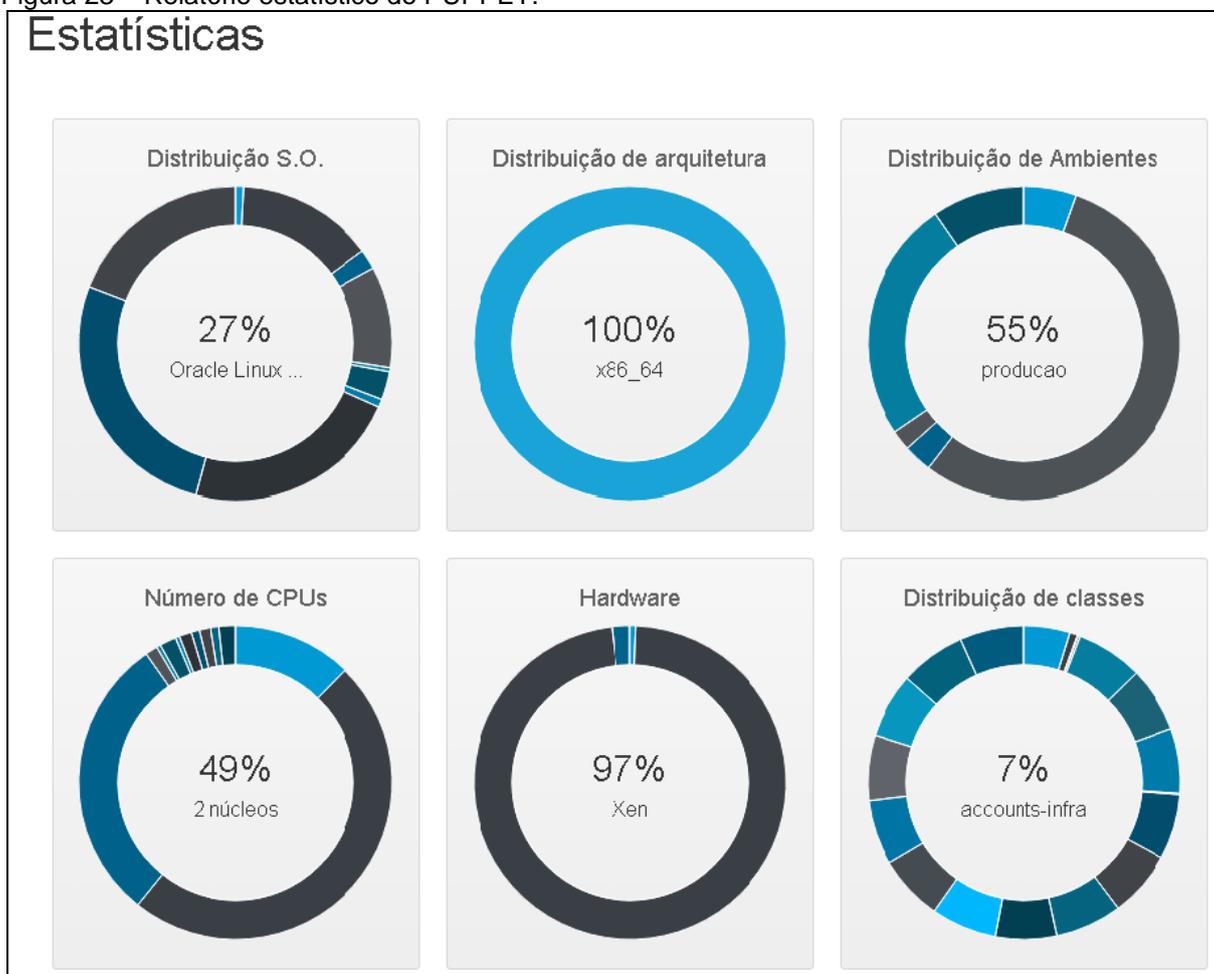
Figura 22 – Status dos nodes no painel via WEB.



Fonte – Próprio autor

O FOREMAN também é capaz de gerar relatórios dos acontecimentos passados descrevendo a quantidade de falhas e atualizações sofridas por cada node. Com base nos facts enviados pelo nodes o FOREMAN passou a fornecer relatórios estatísticos de grande valia para equipe de gestão de infraestrutura pois nele contém informações da arquitetura dos servidores, distribuição do S.O. que está sendo usada, distribuição dos hosts em porcentagem por ambiente e número de CPU's por servidores, conforme mostra a figura 23.

Figura 23 – Relatório estatístico do PUPPET.



Fonte – Próprio autor

## 9. CONCLUSÃO

Manter os servidores padronizados e atender novas demandas de forma manual se tornou uma tarefa árdua e bastante onerosa, todos os dias são criados novos servidores, aplicações são desenvolvidas e lançadas, servidores são atualizados, usuários são criados e removidos e deploys são feitos. Todas essas demandas surgem de forma simultânea e o não atendimento dentro do prazo traz enormes prejuízos à organização.

A ausência de um Gerenciador de Configurações se tornou um grande problema em empresas que possuem grandes quantidades de servidores e principalmente equipes de desenvolvedores e infraestrutura.

Este trabalho acadêmico demonstrou a real necessidade de um Gerenciador de Configurações dentro de um DataCenter que possui um grande parque de servidores. Através de um levantamento prévio das dificuldades e falhas geradas pela administração manual, das ferramentas de gerência de configurações disponíveis no mercado e das limitações e exigências impostas pela organização, foi proposto a implementação da ferramenta Puppet afim de demonstrar seu grande potencial em automatizar, controlar e padronizar as configurações dos servidores.

O Puppet se mostrou uma excelente ferramenta de Gerência de Configuração apresentando baixa complexidade de instalação, uma boa curva de aprendizado, grande quantidade de usuários e uma vasta documentação porem durante a criação deste trabalho verificou-se que ela carece de uma interface gráfica para administração uma vez que a mesma só oferece um painel com informações básicas sobre os nodes. Esta carência foi suprida com o uso da ferramenta FOREMAN que possibilitou realizar configurações através de uma interface gráfica via WEB, gerar relatórios estatísticos e realizar auditorias.

O processo de automação de uma infraestrutura é um ato contínuo e a Gerencia de configuração é um dos eixos deste processo. Este trabalho foi o a colocação da primeira peça para que o trem passe a andar nos trilhos. Podemos como sugestão para trabalhos futuros a configuração de uma ferramenta de provisionamento dinâmico de servidores e ferramentas de versionamento de código e arquivos de configuração (git).

## REFERÊNCIAS

- FRANK, Felix. **Puppet Essentials**. Estados Unidos: Packt Publishing, 2004
- GOMES, Rafael. **Infraestrutura Ágil** – 21º Seminário RNP de Capacitação e Inovação. 19 Out. 2015. Disponível em: < <https://esr.rnp.br/sci41/rnp-201510.1621>>. Acesso em: Fev. 2017.
- UPHILL, Thomas. et al. **Puppet Cookbook**. 3 ed. Estados Unidos: Packt Publishing, 2005
- TURNBULL, James. **Pulling Strings with Puppet**. Estados Unidos: Apress, 2007
- HILLEBRANDT, Tiago **Entendendo o Puppet** – Curso online de Puppet. 24 Out. 2015. Disponível em: < <https://tiagohillebrandt.eti.br/puppet-1-entendendo-o-puppet.html>>. Acesso em: Fev. 2017.
- UPHILL, Thomas. **Mastering Puppet**. 2 ed. Estados Unidos: Packt Publishing, 2006
- DEBOIS, Patrick. **Agile infrastructure and operations: how infra-gile are you?**. Estados Unidos: Artigo, 2010.
- GEERLING, Jeff. **Ansible for DevOps**. Estados Unidos: Leanpub, 2015.
- MUELLER, Ernest. **What is DevOps** – 2 Aug, 2010. Disponível em <<https://theagileadmin.com/what-is-devops/>>. Acesso em: Fev. 2017.
- UDAYA, Chanaka. **How puppet works in your IT infrastructure** – Artigo. 27 Nov. 2013. Disponível em: < <http://soatutorials.blogspot.com.br/2013/11/how-puppet-works-in-your-it.html>>. Acesso em: Mar. 2017.
- MOONEY, Mike. **What is DevOps really?** – Artigo. 8 Mai. 2014. Disponível em: < <https://www.simple-talk.com/opinion/opinion-pieces/what-is-devops-really/>>. Acesso em: Mar. 2017.
- HUTTERMANN, Michael. **DevOps for Developers**. Estados Unidos: Apress, 2012
- CARVALHO, Augusto. **Webinar Trabalhando com módulos** – Webinar. 12 Jun. 2014. Disponível em: < <https://de.slideshare.net/GutoCarvalho/trabalhando-com-mdulos-no-puppet>>. Acesso em: Mar. 2017.
- PUPPET. **Puppet 3.8 Reference Manual** – Webinar. 12 Jun. 2014. Disponível em: < <https://de.slideshare.net/GutoCarvalho/trabalhando-com-mdulos-no-puppet>>. Acesso em: Mar. 2017.

## ANEXO I

### A. INSTALAÇÃO E CONFIGURAÇÃO DO PUPPET E FOREMAN

#### A.1 Criando a Máquina e Configurando no DNS

Servidor virtualizado no OVM (Oracle Virtual Machine)

```
Hostname: puppetmaster  
  
S.O. Oracle Linux 7  
  
DNS: puppetmaster.producao1.datacenter1
```

#### A.2 Atualizando o sistema operacional e configurando o hostname

```
[root@puppetmaster ~]# yum update -y  
  
[root@puppetmaster ~]# reboot  
  
O manual de boas práticas do PUPPET recomenda usar o FQDN no hostname.  
  
[root@puppet-master ~]# vim /etc/hostname  
  
puppetmaster.producao1.datacenter1
```

#### A.3 Desabilitando o IPTABLES e SELINUX

```
Desabilitando o iptables  
  
[root@puppetmaster ~]# systemctl stop firewalld  
  
[root@puppetmaster ~]# systemctl disable firewalld  
  
Desabilitando o SELINUX  
  
[root@puppetmaster ~]# vim /etc/sysconfig/selinux
```

```
SELINUX=disabled

[root@puppetmaster ~]# setenforce 0

setenforce: SELinux is disabled
```

## A.4 Configurar os Repositórios e instalar o Puppetmaster e Foreman

Adicionando o repositório do puppet para tornar possível a instalação do puppet-server

```
[root@puppetmaster ~]# rpm -ivh https://yum.puppetlabs.com/puppetlabs-release-el-7.noarch.rpm
```

Habilitando o repositório do foreman

```
[root@puppetmaster ~]# yum -y install epel-release
http://yum.theforeman.org/releases/1.8/el7/x86_64/foreman-release.rpm
```

Habilitando o repositório EPEL para resolução de dependências do FOREMAN.

```
[root@puppetmaster ~]# wget
http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm
```

```
[root@puppetmaster ~]# rpm -ivh epel-release-7-5.noarch.rpm
```

Habilitar o repositório opcional da Red Hat para resolver as dependências do Rubygem.

```
[root@puppetmaster ~]#vim /etc/yum.repos.d/puplic-yum-ol7.repo
```

```
[ol7_optional_latest]
```

```
name=Oracle Linux $releasever Optional Latest ($basearch)
```

```
baseurl=http://public-  
yum.oracle.com/repo/OracleLinux/OL7/optional/latest/$basearch/
```

```
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
```

```
gpgcheck=1
```

```
enabled=1
```

Download do instalador do foreman

```
[root@puppetmaster ~]# yum -y install foreman-installer
```

Instalando o foreman e puppet-server

```
[root@puppetmaster foreman]# foreman-installer
```

Adicionando o servidor puppetmaster no gerenciamento web FOREMAN.

```
[root@puppetmaster ~]# puppet agent -t
```

## A.5 Configurando o Banco de dados do Puppet e Foreman no ambiente de produção

O banco de dados utilizado pelo Puppetmaster e Foreman foi configurado no ambiente de produção de Banco de dados. O Banco de dados destes serviços no ambiente de produção é devido a performance, monitoramento e melhores práticas executadas pela torre de Banco de Dados. As configurações feitas para tal medida serão descritas nos próximos tópicos.

### A.5.1 Configurando os parâmetros do foreman

O arquivo foreman-installer-answers.yaml contém as opções desejadas para instalação do foreman. Neste arquivo configura-se as entradas do banco de dados, senha de administrador do foreman, dentre outros.

Editando o arquivo foreman-installer-answers.yaml

```
[root@puppetmaster ~]# vim /etc/foreman/foreman-installer-answers.yaml
```

```
db_manage: true

db_type: postgresql

db_adapter: postgresql

db_host: "10.0.11.x"

db_port: 5432

db_database: foremanp

db_username: foremanp

db_password: "senha_desejada"

admin_username: admin

admin_password: "senha_desejada"
```

Para que as novas configurações entre em vigor é necessário executar o comando `foreman-installer` novamente.

## ANEXO II

### B. ADMINISTRANDO O GERENCIADOR DE CONFIGURAÇÕES PUPPET

## B.1 Criando Ambientes

A configuração de um ambiente consiste em dois passos.

- Criação do diretório em `/etc/puppet/environments/` com o nome do ambiente desejado.

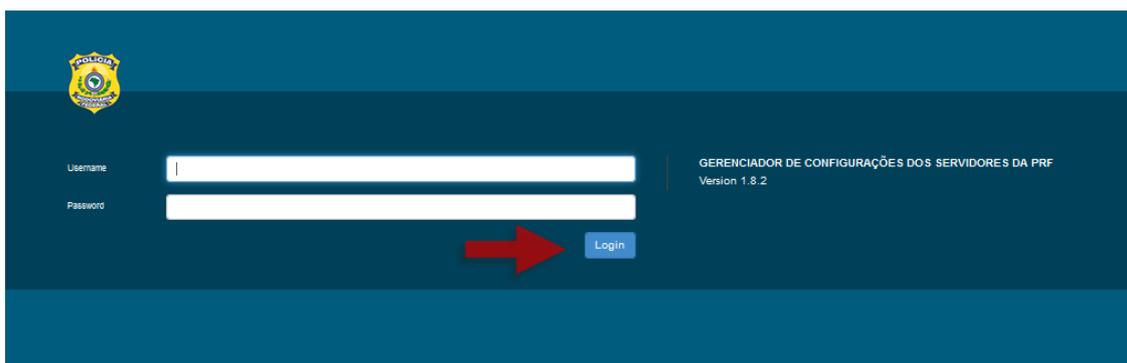
```
[root@puppetmaster ~]# cd /etc/puppet/environments/
```

```
[root@puppetmaster environments]# cp -rfvp example_env/ teste
```

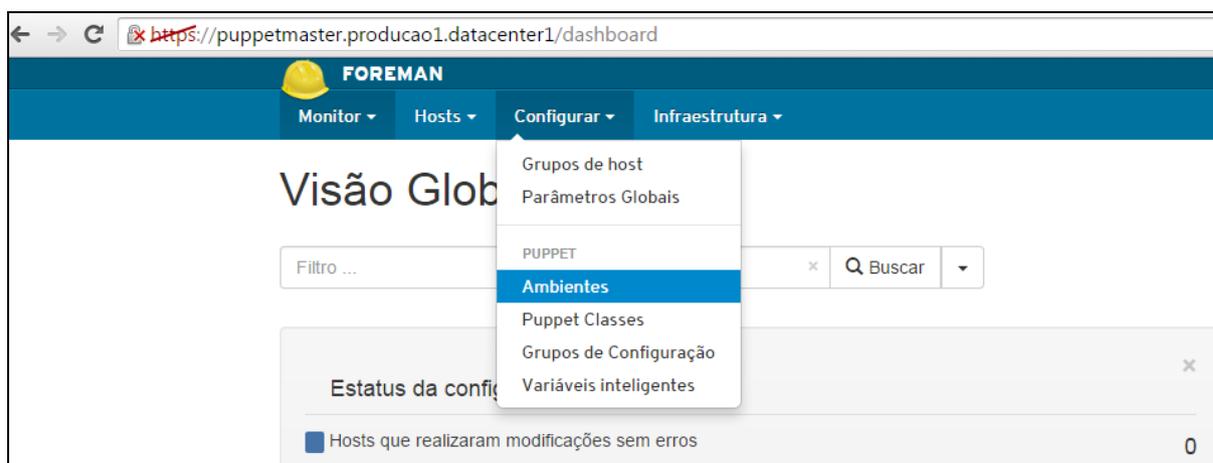
- Criação dos ambientes no foreman.

Para criar os ambientes no foreman acesse o link: da interface web e siga os passos descritos abaixo.

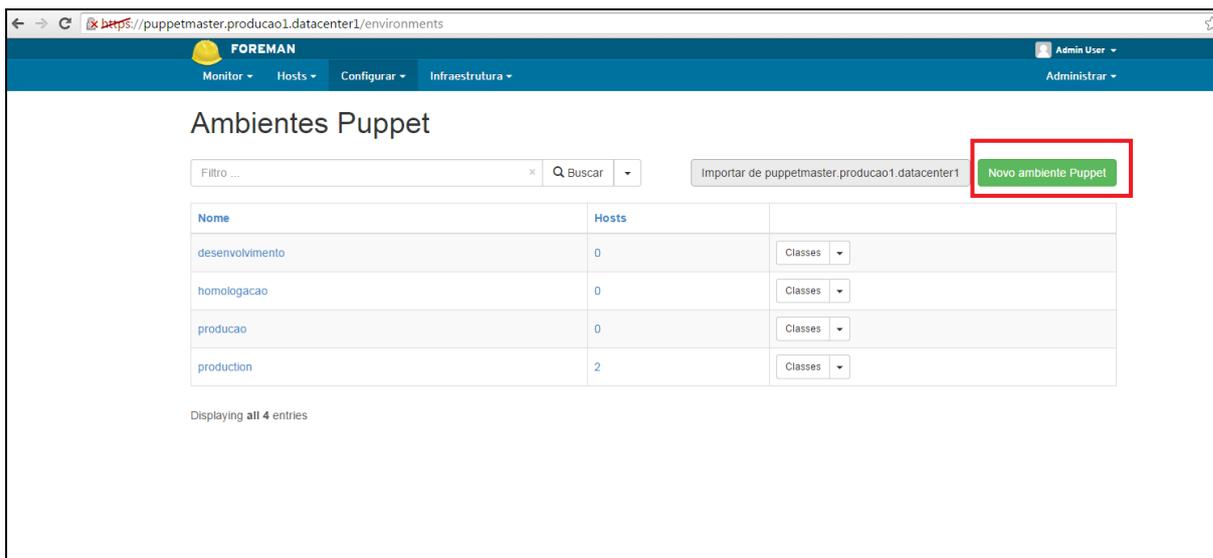
Informe o usuário e senha, clique em **“Login**



Selecione o menu configurar > ambiente



Clique em novo ambiente

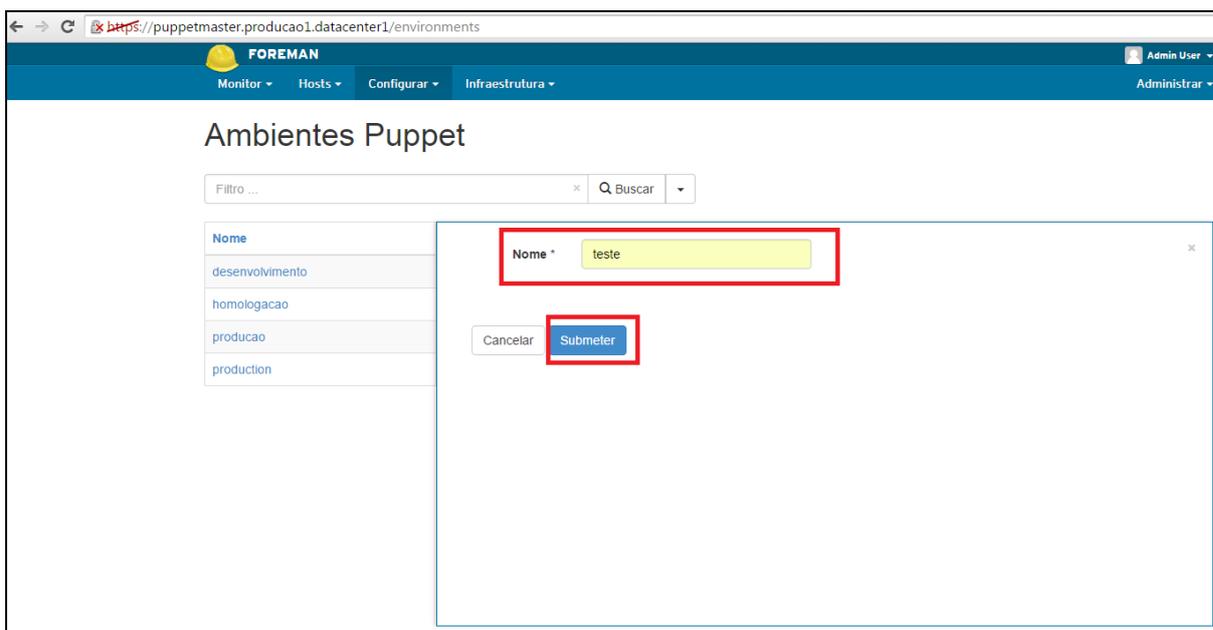


The screenshot shows the Foreman web interface for managing Puppet environments. The page title is "Ambientes Puppet". At the top right, there is a green button labeled "Novo ambiente Puppet" which is highlighted with a red rectangular box. Below the header, there is a search bar and a table listing existing environments.

Nome	Hosts	
desenvolvimento	0	Classes ▾
homologacao	0	Classes ▾
producao	0	Classes ▾
production	2	Classes ▾

Displaying all 4 entries

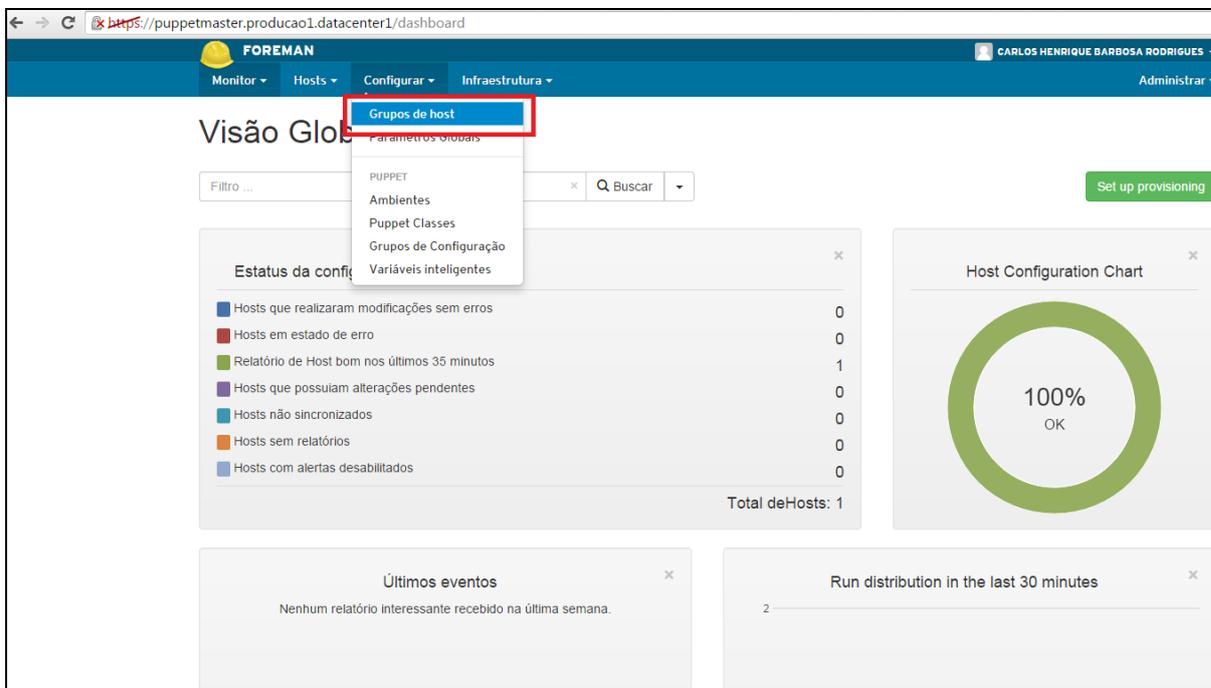
Preencha o nome do ambiente desejado (este nome tem que ser o mesmo do diretório criado em /etc/puppet/environments) e clique em submeter.



The screenshot shows the "Novo ambiente Puppet" form in the Foreman interface. The form has a text input field for "Nome \*" containing the text "teste", which is highlighted with a red rectangular box. Below the input field, there are two buttons: "Cancelar" and "Submeter", with the "Submeter" button also highlighted by a red rectangular box. The background shows the same environment list as the previous screenshot.

## B.2 Criando Grupos de Hosts

Selecione o menu configurar > Grupos de Hosts



The screenshot shows the Foreman dashboard interface. The top navigation bar includes 'Monitor', 'Hosts', 'Configurar', and 'Infraestrutura'. The 'Configurar' menu is open, and 'Grupos de host' is highlighted with a red box. Below the menu, the dashboard displays a 'Visão Global' section with a search bar and a 'Set up provisioning' button. A table titled 'Estatus da configuração' shows various host states and their counts. A 'Host Configuration Chart' shows a 100% OK status. Other sections include 'Últimos eventos' and 'Run distribution in the last 30 minutes'.

Estatus da configuração	
Hosts que realizaram modificações sem erros	0
Hosts em estado de erro	0
Relatório de Host bom nos últimos 35 minutos	1
Hosts que possuíam alterações pendentes	0
Hosts não sincronizados	0
Hosts sem relatórios	0
Hosts com alertas desabilitados	0
<b>Total de Hosts:</b>	<b>1</b>

Clique em novo grupo de host



The screenshot shows the 'Configuração do grupo de host' page in Foreman. The page title is 'Configuração do grupo de host'. A green button labeled 'Novo grupo de hosts' is highlighted with a red box. Below the title, there is a paragraph explaining host groups and another paragraph discussing Puppet classes and variables.

Um grupo de hosts é de certa forma semelhante à uma declaração de nó herdado, ou seja, é um agrupamento de alto nível de classes que pode ser chamado e tratado como uma unidade. Este é, então, tratado como um modelo e é selecionável durante a criação de um novo host e assegura que o host está configurado em um dos seus estados pré-definidos. Além de definir quais classes puppet são incluídos quando a construção deste tipo de host, você também é capaz de atribuir as variáveis e informações de provisionamento de um grupo de host para refinar ainda mais o comportamento do tempo de execução de puppets. As classes do grupo de host e variáveis do grupo de host são incluídas nas informações de nó externo quando o puppetmaster compila a configuração do host.

Existem duas estratégias ao utilizar os grupos de host. Você pode criar classes de puppets que representam configurações de host de alto nível, por exemplo, a classe **host-type-idap-server**, que inclui todas as funcionalidades necessárias a partir de outros módulos ou você pode decidir criar um grupo de host chamado **host-type-idap-server** e adicionar as classes necessárias para a configuração do grupo de host. Estas duas opções são decisões pessoais e são de sua escolha (onde a diferença principal seria as configurações de parâmetro/variáveis).

Selecione as classes que deseja aplicar para grupo de host que está sendo criado.

The screenshot shows the Foreman web interface for creating a new host group. The browser address bar displays `https://puppetmaster.producao1.datacenter1/hostgroups/new`. The page title is "Novo grupo de hosts". The navigation menu includes "Monitor", "Hosts", "Configurar", "Infraestrutura", and "Administrar". The user is logged in as "CARLOS HENRIQUE BARBOSA RODRIGUES".

The main content area has tabs for "Grupo de host", "Puppet Classes", "Rede", "Sistema Operacional", and "Parâmetros". The "Puppet Classes" tab is active. It contains two columns: "Classes incluídas" and "Classes disponíveis". A search box labeled "Filtrar classes" is located under the "Classes disponíveis" column. At the bottom of the form, there are "Cancelar" and "Submeter" buttons.

## B.3 Instalando e Configurando um Módulo

### B.3.1 instalando os módulos no puppet master

Os módulos usados no gerenciador de configurações da PRF foram adquiridos no repositório do puppetlabs, sempre com base nas avaliações e atualizações dos módulos. Para instalar um módulo no puppet o comando usado foi: **puppet module install --environment <nome\_do\_ambiente> <nome\_do\_módulo>**

Segue abaixo um exemplo de instalação do módulo.

```
[root@puppetmaster ~]# puppet module install --environment producao puppetlabs-ntp
```

### B.3.2 Configurando os módulos no FOREMAN

Após a instalação dos módulos no puppet master, conforme descrito no item A.3.1, é necessário configura-los no FOREMAN para que os mesmos entrem em produção. Os passos descritos neste tópico detalha como configurar um módulo no FOREMAN.

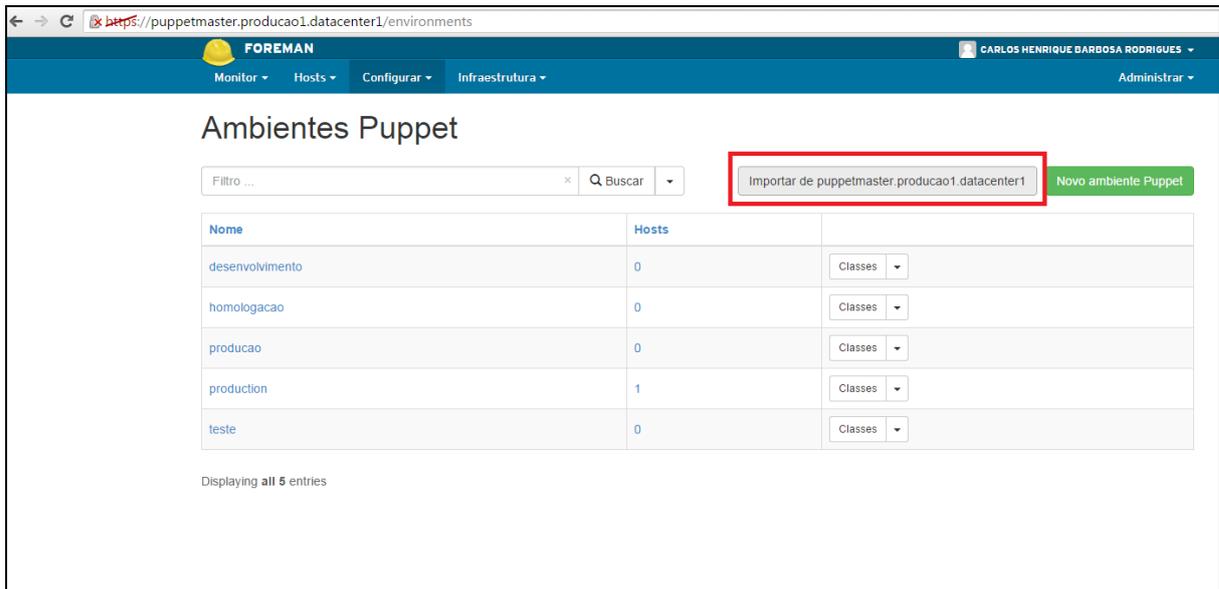
**OBS: Para que um módulo funcione no FOREMAN é necessário que ele esteja configurado com HIERA**

Clique no menu Configurar > Ambientes

The screenshot shows the Foreman web interface. The top navigation bar includes 'Monitor', 'Hosts', 'Configurar', and 'Infraestrutura'. The 'Configurar' menu is open, showing options like 'Grupos de host', 'Parâmetros Globais', 'Ambientes', 'Puppet Classes', 'Grupos de Configuração', and 'Variáveis inteligentes'. The 'Ambientes' option is highlighted. Below the menu, the 'Visão Global' dashboard is visible, featuring a table of configuration status, a 'Host Configuration Chart' showing 100% OK, and sections for 'Últimos eventos' and 'Run distribution in the last 30 minutes'.

Descrição	Quantidade
Hosts que realizaram modificações sem erros	0
Hosts em estado de erro	0
Relatório de Host bom nos últimos 35 minutos	1
Hosts que possuem alterações pendentes	0
Hosts não sincronizados	0
Hosts sem relatórios	0
Hosts com alertas desabilitados	0
<b>Total de Hosts:</b>	<b>1</b>

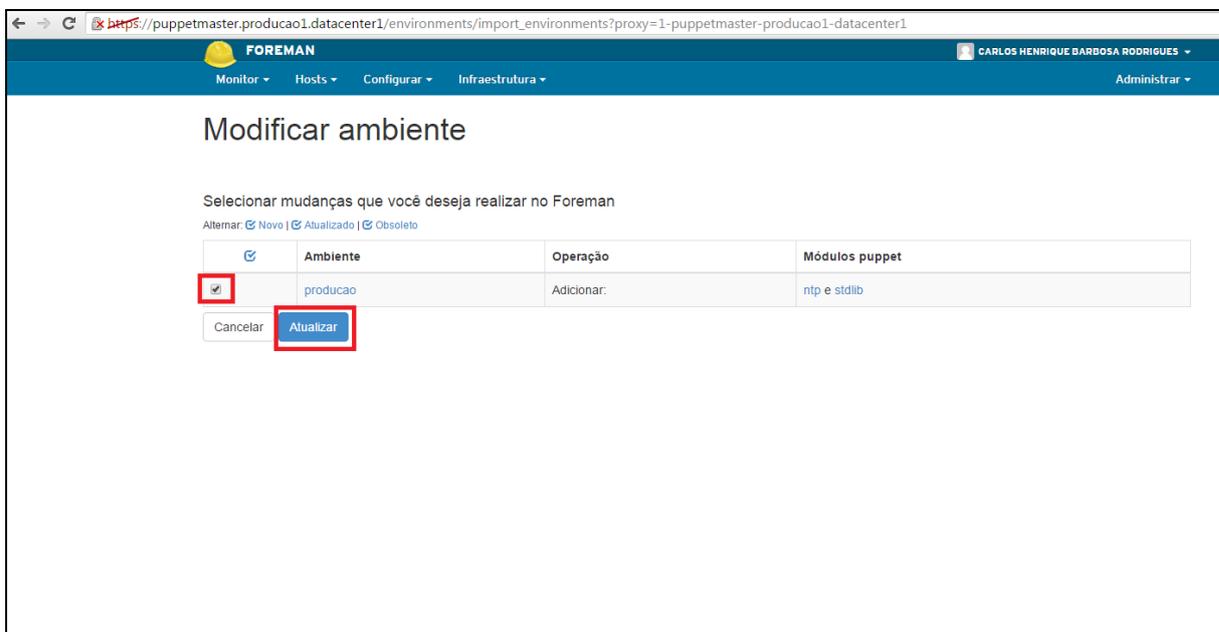
Clique em importar de <servidor puppet master>



The screenshot shows the 'Ambientes Puppet' page in the Foreman interface. The page title is 'Ambientes Puppet'. There is a search bar with 'Filtro ...' and 'Buscar'. A button labeled 'Importar de puppetmaster.producao1.datacenter1' is highlighted with a red box. To its right is a green button labeled 'Novo ambiente Puppet'. Below these is a table with columns 'Nome', 'Hosts', and 'Classes'. The table contains five rows: 'desenvolvimento' (0 hosts), 'homologacao' (0 hosts), 'producao' (0 hosts), 'production' (1 host), and 'teste' (0 hosts). Each row has a 'Classes' dropdown menu. At the bottom, it says 'Displaying all 5 entries'.

Nome	Hosts	Classes
desenvolvimento	0	Classes
homologacao	0	Classes
producao	0	Classes
production	1	Classes
teste	0	Classes

Marque o módulo desejado e clique em atualizar



The screenshot shows the 'Modificar ambiente' page in the Foreman interface. The page title is 'Modificar ambiente'. Below the title, it says 'Selecionar mudanças que você deseja realizar no Foreman'. There are three radio buttons: 'Novo' (selected), 'Atualizado', and 'Obsoleto'. Below this is a table with columns 'Ambiente', 'Operação', and 'Módulos puppet'. The table contains one row: 'producao' with 'Adicionar:' operation and 'ntp e stdlib' modules. The checkbox in the first column is checked and highlighted with a red box. Below the table are two buttons: 'Cancelar' and 'Atualizar', with the 'Atualizar' button highlighted with a red box.

Ambiente	Operação	Módulos puppet
<input checked="" type="checkbox"/> producao	Adicionar:	ntp e stdlib

## B.4 Instalando e Configurando um Módulo

### B.4.1 Adicionando um Node (host)

Para os Sistemas Operacionais derivados do RHEL

```
#yum install puppet
```

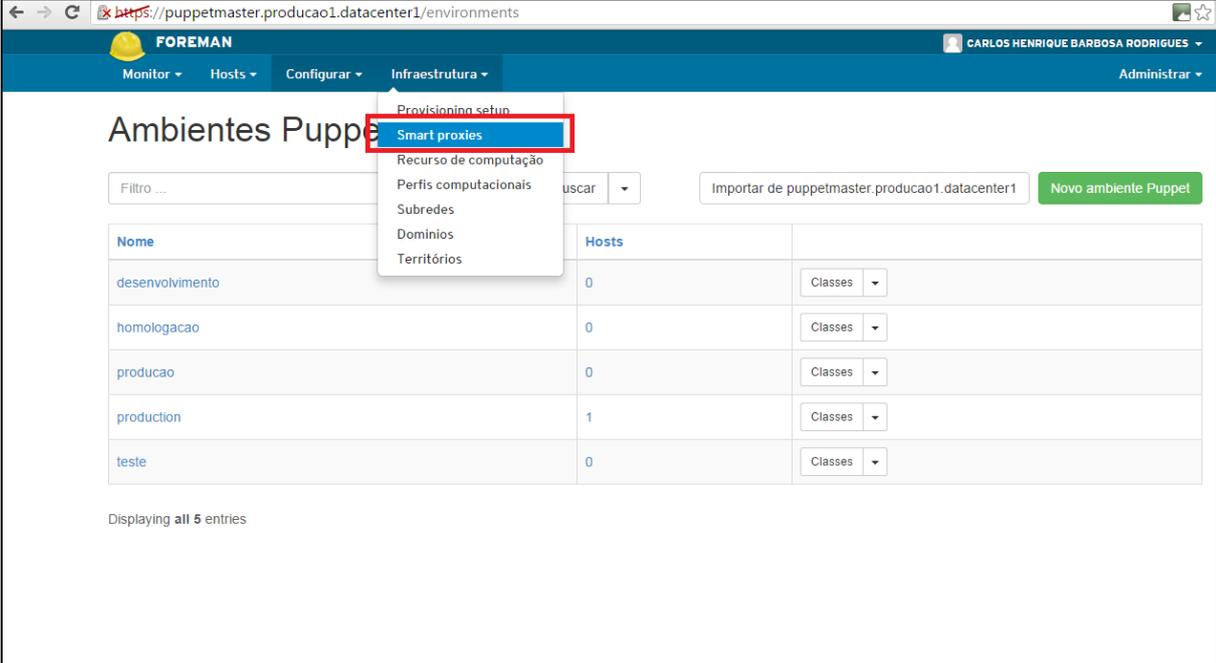
Para os Sistemas Operacionais derivados do Debian

```
#apt-get install puppet
```

### B.4.2 Assinando um certificado e adicionando um Node ao FOREMAN

Após a instalação do puppet no host cliente, é necessário seguir os passos abaixo para que o mesmo seja atualizado pelo puppet master.

Clique em Infraestrutura > Smart Proxies

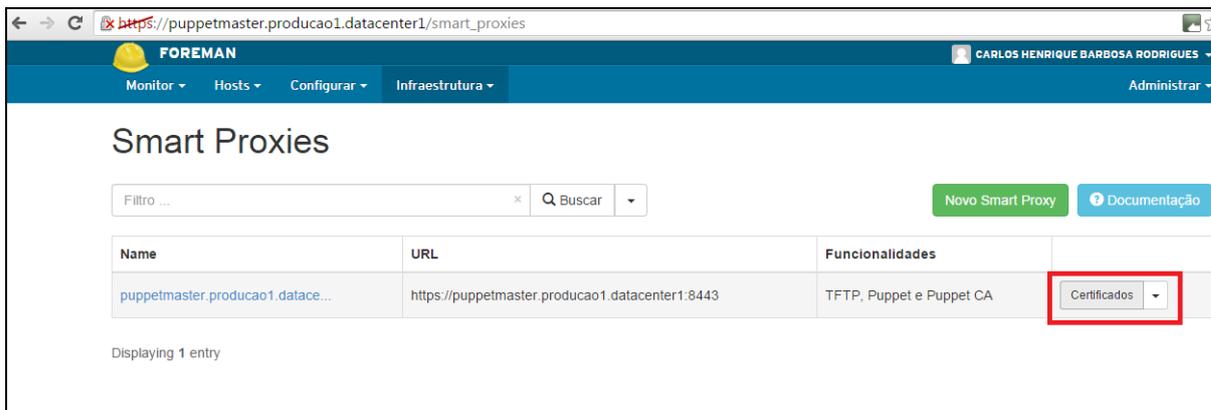


The screenshot shows the Foreman web interface. The browser address bar displays `https://puppetmaster.producao1.datacenter1/environments`. The page title is "Ambientes Puppet". The navigation menu includes "Monitor", "Hosts", "Configurar", "Infraestrutura", and "Administrar". The "Infraestrutura" menu is expanded, showing options: "Provisioning setup", "Smart proxies" (highlighted with a red box), "Recurso de computação", "Perfis computacionais", "Subredes", "Domínios", and "Territórios". Below the menu, there is a search bar and a button "Importar de puppetmaster.producao1.datacenter1". A table lists Puppet environments:

Nome	Hosts	Classes
desenvolvimento	0	Classes
homologacao	0	Classes
producao	0	Classes
production	1	Classes
teste	0	Classes

At the bottom, it says "Displaying all 5 entries".

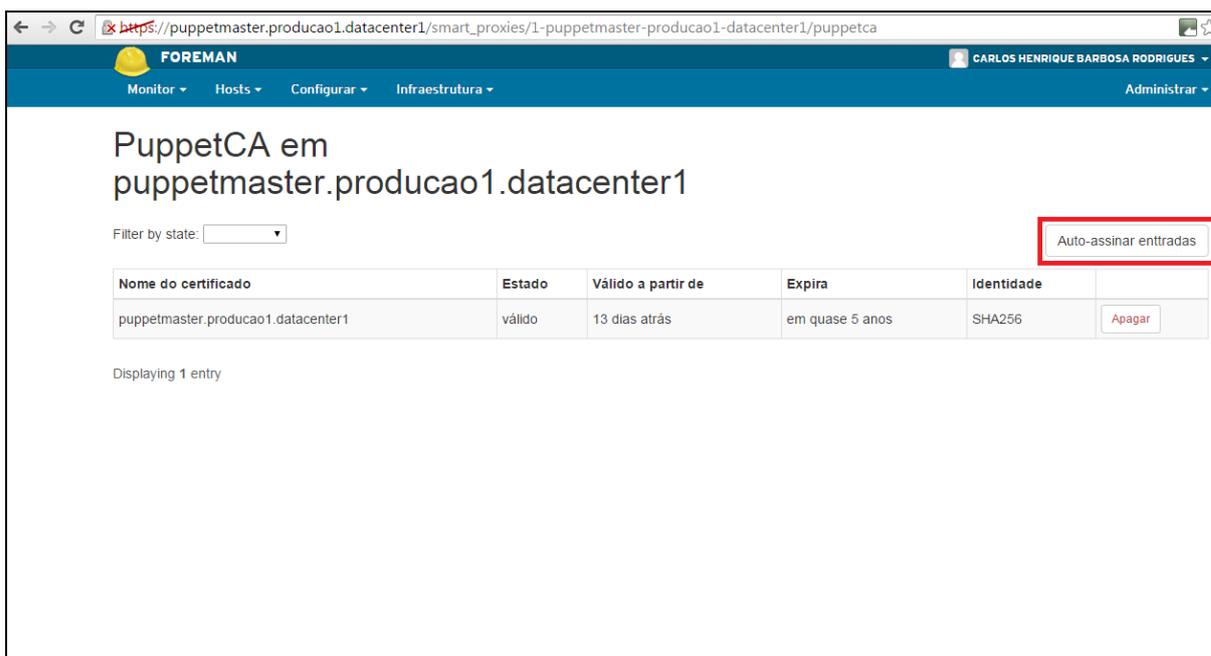
Clique em certificados, conforme mostra a figura abaixo.



The screenshot shows the Foreman web interface. The page title is "Smart Proxies". There is a search bar and a "Buscar" button. A table lists smart proxies. The first entry is "puppetmaster.producao1.datacenter1" with URL "https://puppetmaster.producao1.datacenter1:8443" and functionalities "TFTP, Puppet e Puppet CA". A dropdown menu labeled "Certificados" is highlighted with a red box.

Name	URL	Funcionalidades	
puppetmaster.producao1.datacenter1	https://puppetmaster.producao1.datacenter1:8443	TFTP, Puppet e Puppet CA	Certificados

Clique em Auto-assinar entradas



The screenshot shows the Foreman web interface for "PuppetCA em puppetmaster.producao1.datacenter1". There is a "Filter by state:" dropdown. A table lists certificates. The first entry is "puppetmaster.producao1.datacenter1" with state "válido", valid from "13 dias atrás", expires "em quase 5 anos", and identity "SHA256". A button labeled "Auto-assinar entradas" is highlighted with a red box.

Nome do certificado	Estado	Válido a partir de	Expira	Identidade	
puppetmaster.producao1.datacenter1	válido	13 dias atrás	em quase 5 anos	SHA256	Apagar

Após estes passos o node terá sua sincronização com o puppet ativa e a cada 30 minutos realizará sua atualização.