

---

Curso de Ciência da computação  
Universidade Estadual de Mato Grosso do Sul

---

**Configuração de um balanceador de carga para a plataforma de  
educação a distância Moodle**

Heder Gomes da Silva

Dr. Osvaldo Vargas Jaques(Orientador)

Dourados - MS  
2018



# Configuração de um balanceador de carga para a plataforma de educação a distância Moodle

Heder Gomes da Silva

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Heder Gomes da Silva e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Dourados, 25 de novembro de 2018

Dr. Osvaldo Vargas Jaques(Orientador)



## Configuração de um balanceador de carga para a plataforma de educação a distância Moodle

Heder Gomes da Silva

Novembro de 2018

### **Banca Examinadora:**

Prof. Dr. Osvaldo Vargas Jaques(Orientador)  
Área de Computação - UEMS

Prof. Me. ANDRÉ CHASTEL LIMA  
Área de Computação - UEMS

Prof. Dr. RUBENS BARBOSA FILHO  
Área de Computação - UEMS



# AGRADECIMENTOS

Agradeço em primeiro lugar a Deus, pela força e coragem durante toda esta longa caminhada acadêmica. Ao professor doutor Osvaldo Vargas Jaques, pela paciência na orientação e incentivo, que tornaram possível a conclusão deste trabalho. Agradeço também a meu supervisor de estágio Me. Fábio Augusto de Souza Seabra, por todo aprendizado durante o período de estágio. Ao meu pai Derivaldo da Silva, minha mãe Hebe Gomes da Silva, minha irmã Carolina Gomes da Silva, minha namorada Natalia Lamego Flores toda minha família que sempre me apoiaram para que eu concluísse mais esta etapa da minha vida.





# RESUMO

Apesar de já não ser novidade, a utilização do balanceamento de carga de rede para o aperfeiçoamento de recursos de infraestrutura de redes de computadores e de sistema operacional não é a primeira opção de implementar um servidor habitualmente, mesmo com a possibilidade deste componente ser implementado tanto em pequenas redes domésticas quanto em redes de grandes empresas. Até mesmo em instituições de ensino, seja fundamental, médio ou ensino superior, que possuem suas respectivas plataformas de ensino *online*, cujo funcionamento depende de servidores potentes. Fato é que tais servidores custam valores altíssimos, e, na maioria das vezes, essas instituições de ensino, sobretudo as públicas, não dispõem de recursos financeiros necessários para adquiri-los. É neste contexto que a implementação de um balanceador de cargas apresenta-se como uma alternativa extremamente benéfica, não só ao otimizar a infraestrutura de rede, mas também ao reduzir custos. Ao elaborar o presente trabalho, foi realizado um estudo do balanceador HAProxy para a plataforma de ensino Moodle, como também de ferramentas necessárias para o funcionamento do balanceador. Dentre as vantagens do emprego do HAProxy, destaca-se a redução da necessidade da utilização de vários IPs reais, a possibilidade da escalabilidade e redundância do instanciamento dos servidores de aplicação e, também, a oportunidade de uso de sessão permanente para a aplicação através do uso de *cookies*. A utilização do balanceador de cargas pode ser implementado para muitas outras aplicações trazendo os benefícios já citados e muitos outros.

**Palavras-chave:** Balanceamento de Carga, HAProxy, Moodle



# ABSTRACT

Despite the fact that it is no longer a new, the use of network load balancing, aiming the improvement of infrastructure resources of computer network and operating system, it's not the first method to implement a server that comes to mind, even with the possibility of this component being implemented in home networks as well as in big companies' network, also in Educational Institutions, for instance, elementary schools, high schools or higher education (college / university), that possess their own online teaching platforms, which operation depends on powerful web servers. Indeed, those servers cost a great amount of money, and oftentimes these Educational Institutions, mainly the public ones, can not afford them. It's in this context that the implementation of a network load balancing shows off as an extremely beneficial alternative, not only optimizing the network infrastructure, but also reducing expenses. Through the development of this undergraduate thesis a study was made regarding the use of HAProxy balancing in the Moodle teaching platform, as well as the necessary tools for its operation. Among the many benefits that the employment of HAProxy, the need of many real IPs, the possibility of scalability and instantiation redundancy stands out, besides the opportunity of using a permanent session to its application through the cookies. implemented for many other applications bringing the benefits already mentioned and many others.

**Key-words:** Load balancing, HAProxy, Moodle



# SUMÁRIO

	Lista de ilustrações . . . . .	xvii
1	INTRODUÇÃO . . . . .	1
1.1	Objetivos Gerais . . . . .	2
1.1.1	Objetivos Específicos . . . . .	2
1.2	Organizações dos capítulos . . . . .	3
2	REFERENCIAL TEÓRICO . . . . .	4
2.1	MOODLE . . . . .	4
2.2	PROXY . . . . .	5
2.3	APACHE . . . . .	6
2.4	NFS . . . . .	7
2.5	BALANCEAMENTO DE CARGAS . . . . .	8
2.6	POSTGRESQL . . . . .	9
2.7	DEBIAN . . . . .	10
2.8	PROXMOX VIRTUAL ENVIRONMENT . . . . .	10
2.9	Teste de Carga . . . . .	11
3	HAPROXY . . . . .	12
3.1	Descrição . . . . .	12
3.2	Plataformas Suportadas . . . . .	12
3.3	Atuação . . . . .	13
3.4	Confiabilidade . . . . .	15
3.5	Segurança . . . . .	16
4	CONFIGURAÇÃO DA APLICAÇÃO . . . . .	18
4.1	Servidor NFS . . . . .	18
4.1.1	Configurando interface de rede . . . . .	18
4.1.2	Configurando Hostname . . . . .	18
4.1.3	Configurando Hosts . . . . .	19
4.1.4	Instalação de programas necessários . . . . .	19
4.1.5	Formatar a partição . . . . .	19
4.1.6	Montar uma partição /var/moodledata em EXT4 . . . . .	20
4.1.7	Configurar o servidor NFS . . . . .	20
4.1.8	Configurando o Firewall . . . . .	21
4.2	Servidor de Banco de dados . . . . .	21

4.2.1	Configurando interface de rede . . . . .	22
4.2.2	Configurando Hostname . . . . .	22
4.2.3	Configurando Hosts . . . . .	22
4.2.4	Instalação de programas necessários . . . . .	22
4.2.5	Criando usuário para o moodle . . . . .	23
4.2.6	Configurando o postgresql . . . . .	23
4.2.7	Configurando o Firewall . . . . .	24
4.3	Servidor Moodle 1 . . . . .	25
4.3.1	Configurando interface de rede . . . . .	25
4.3.2	Configurando Hostname . . . . .	25
4.3.3	Configurando Hosts . . . . .	25
4.3.4	Instalação de programas necessários . . . . .	26
4.3.5	Configurando o Firewall . . . . .	26
4.3.6	Montando partição Moodledata do servidor Moodle1 . . . . .	26
4.3.7	Instalação do Moodle pelo terminal . . . . .	27
4.4	Servidor Moodle 2 . . . . .	27
4.4.1	Configurando interface de rede . . . . .	27
4.4.2	Configurando Hostname . . . . .	28
4.4.3	Configurando Hosts . . . . .	28
4.4.4	Instalação de programas necessários . . . . .	29
4.4.5	Configurando o Firewall . . . . .	29
4.4.6	Montando partição Moodledata do servidor Moodle2 . . . . .	29
4.4.7	Instalação do Moodle pelo terminal . . . . .	29
4.5	Servidor LOADBALANCE . . . . .	30
4.5.1	Configurando interface de rede . . . . .	30
4.5.2	Configurando Hostname . . . . .	31
4.5.3	Configurando Hosts . . . . .	31
4.5.4	Instalação o HAProxy . . . . .	31
4.5.5	Configurando o Firewall . . . . .	31
4.5.6	Configurando o HAProxy . . . . .	32
4.5.7	Instação via <i>Browser</i> . . . . .	33
5	RESULTADOS . . . . .	39
6	CONCLUSÃO . . . . .	43
	REFERÊNCIAS . . . . .	44

# LISTA DE ABREVIATURAS E SIGLAS

MOODLE	<i>Modular Object-Oriented Dynamic Learning Environment</i>
TCP	<i>Transmission Control Protocol</i>
IP	Protocolo de Internet
WEB	<i>World Wide Web</i>
PHP	<i>Hypertext Preprocessor</i>
HTTP	<i>Hypertext Transfer Protocol</i>
NCSA	<i>National Center for Supercomputing Applications</i>
Gb	Gigabit
Gbps	Gigabit por segundo
CPU	Unidade Central de Processamento
NFS	<i>Network File Systems</i>
VM	<i>virtual machine</i>
DEC	<i>Digital Equipment Corporation</i>
DECnet	é um conjunto de protocolos de redes criado pela <i>Digital Equipment Corporation</i>
RFC	<i>Request for Comments</i>
VSF	<i>Virtual Filesystem</i>
RPC	<i>Remote Procedure Call</i>
XRD	<i>External Data Representation</i>
UDP	<i>Universal Datagram Protocol</i>
SQL	<i>Structured Query Language</i>
APT	<i>Advanced Packaging Tool</i>
KVM	<i>Kernel-based Virtual Machine</i>

LXC	<i>Linux Containers</i>
TI	Tecnologia da Informação
DNS	<i>Domain Name System</i>
SRV	<i>Service</i>
SSL	<i>Secure Socket Layer</i>
DDoS	<i>Distributed Denial of Service</i>
AIX	<i>Advanced Interactive eXecutive</i>
Gbps	<i>Gigabits per second</i>
NIC	<i>Network Interface Card</i>
ACL	<i>Access Control List</i>
EBtree	<i>Elastic Binary Tree</i>
LTS	<i>Transport Layer Security</i>
REGEX	<i>Regular Expression</i>



# LISTA DE ILUSTRAÇÕES

Figura 1 – Forma de trabalho do balanceador . . . . .	2
Figura 2 – Roteiro . . . . .	3
Figura 3 – Arquivo Interface do servidor NFS . . . . .	18
Figura 4 – Arquivo hostname do servidor NFS . . . . .	19
Figura 5 – Arquivo hosts do servidor NFS . . . . .	19
Figura 6 – Comando fdisk -l . . . . .	20
Figura 7 – Arquivo fstab do servidor NFS . . . . .	20
Figura 8 – Arquivo exports do servidor NFS . . . . .	21
Figura 9 – Arquivo rules do servidor NFS . . . . .	21
Figura 10 – Arquivo Interface do servidor de Banco de dados . . . . .	22
Figura 11 – Arquivo hostname do servidor de Banco de dados . . . . .	22
Figura 12 – Arquivo hosts do servidor de Banco de dados . . . . .	23
Figura 13 – Arquivo pg_hba.conf . . . . .	23
Figura 14 – Arquivo postgresql.conf . . . . .	24
Figura 15 – Arquivo rules do servidor de Banco de dados . . . . .	24
Figura 16 – Arquivo Interface do servidor Moodle 1 . . . . .	25
Figura 17 – Arquivo hostname do servidor da aplicação Moodle 1 . . . . .	25
Figura 18 – Arquivo hosts do servidor Moodle 1 . . . . .	26
Figura 19 – Arquivo rules do servidor Moodle 1 . . . . .	26
Figura 20 – Arquivo fstab do servidor Moodle 1 . . . . .	27
Figura 21 – Arquivo Interface do servidor Moodle 2 . . . . .	28
Figura 22 – Arquivo hostname do servidor Moodle 2 . . . . .	28
Figura 23 – Arquivo hosts do servidor Moodle 2 . . . . .	28
Figura 24 – Arquivo rules do servidor Moodle 2 . . . . .	29
Figura 25 – Arquivo fstab do servidor Moodle2 . . . . .	30
Figura 26 – Arquivo Interface . . . . .	30
Figura 27 – Arquivo hostname . . . . .	31
Figura 28 – Arquivo hosts . . . . .	31
Figura 29 – Arquivo rules . . . . .	32
Figura 30 – Arquivo haproxy.cfg parte1 . . . . .	32
Figura 31 – Arquivo haproxy.cfg parte2 . . . . .	33
Figura 32 – Instalação Moodle passo 1 . . . . .	33
Figura 33 – Instalação Moodle passo 2 . . . . .	34
Figura 34 – Instalação Moodle passo 3 . . . . .	35
Figura 35 – Instalação Moodle passo 4 . . . . .	36

Figura 36 – Instalação Moodle passo 5 . . . . .	37
Figura 37 – Instalação Moodle passo 6 . . . . .	38
Figura 38 – Teste de Carga sem o Balanceamento de cargas . . . . .	39
Figura 39 – Teste de Carga com o Balanceamento de cargas . . . . .	40
Figura 40 – Imagens ampliadas dos resultados dos testes de carga . . . . .	40
Figura 41 – VM PROXMOX . . . . .	41
Figura 42 – Loadbalance HAProxy . . . . .	41
Figura 43 – <i>Login</i> do usuário na plataforma Moodle . . . . .	42
Figura 44 – Plataforma Moodle . . . . .	42

# 1 INTRODUÇÃO

O número de usuários de smartphones, tablets, notebook e de computadores de mesa tem aumentado com o avanço tecnológico. O desenvolvimento da tecnologia tem se tornado cada vez mais comum em várias áreas de nossas vidas, como trabalho, faculdade e lares. A quase todo momento estamos conectados utilizando o smartphone ou o notebook, seja para nos comunicarmos com algum amigo, parente ou em algum momento de lazer assistindo à uma série ou filme.

O crescimento tecnológico tem exigido conexões de internet cada vez mais velozes e que se tenham servidores o mais potente possível para suportar toda essa demanda de milhões de conexões por segundo. Ainda que muitas dessas empresas prestadoras de serviços tecnológicos possuam servidores extremamente potentes, as vezes, essas conexões são perdidas devido a quantidade maior de usuários do que o suportado. Isso poderia ser feito substituindo o servidor por um mais potente, mas isso demanda um alto custo financeiro. Uma alternativa econômica é o balanceamento de carga.

O balanceamento de cargas é uma técnica para distribuir de modo uniforme as solicitações de serviços entre os computadores, evitando assim picos de sobrecarga. Resumidamente, o balanceamento de carga é responsável por multiplicar toda essa infraestrutura para que todo conteúdo esteja sempre disponível para seu usuário. O balanceamento de carga também maximiza o desempenho dos servidores. Grosso modo, através de um algoritmo inteligente, os usuários são redistribuídos entre os servidores de aplicação para os que estiverem com menos acesso de usuário, desta forma ele diminui tempo de resposta a sites, aplicativos e demais serviços que utilizem servidor, evitando assim sobrecarga nos servidores.

Um servidor pode ter seu tempo de vida curto se houver novamente um grande crescimento de acessos inesperado, nesse caso teria que ser gasto mais dinheiro o que não é viável para nenhuma empresa, por maior que ela seja.

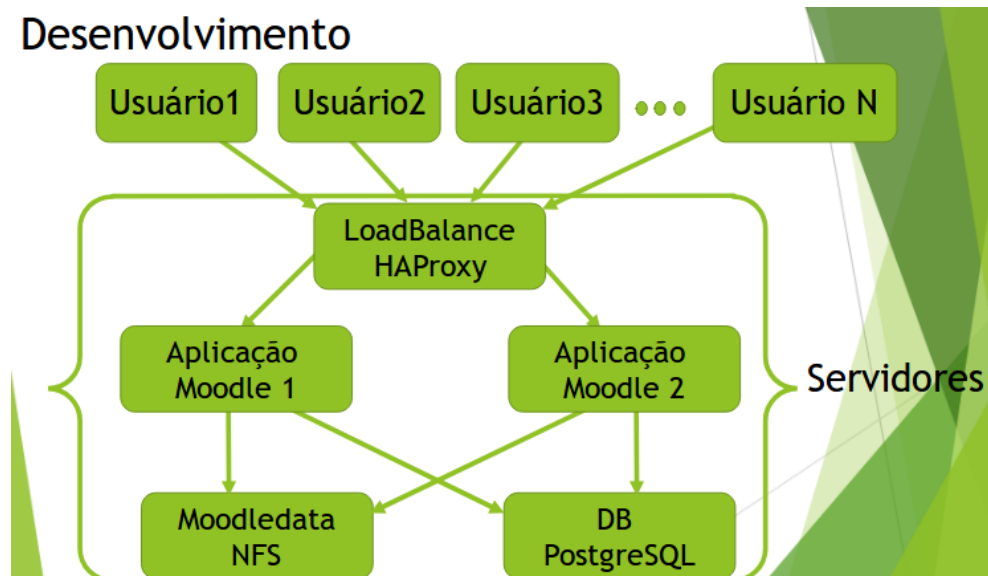
Uma infraestrutura de servidores que utiliza o balanceamento de forma bem avançada e muito bem implementada alcança um alto nível de desempenho. O balanceamento de carga tem algo muito positivo que é a capacidade de poder adicionar mais servidores para que estejam trabalhando em conjunto sempre que houver um grande aumento do número de usuários das aplicações, sem ter que gastar enormes quantidades de dinheiro comprando um super servidor pra trabalhar sozinho.

Outro benefício importante que deve ser lembrado é a alta disponibilidade que o balanceamento proporciona, caso algum servidor venha falhar, o serviço é redirecionado para outro que esteja disponível.

Com o crescimento cada vez maior de cursos ou disciplinas que são ofertados a

distância foi pensado para este trabalho a implementação de um balanceador de carga para a plataforma de educação a distância moodle, que é um software livre voltado para o aprendizado.

Figura 1 – Forma de trabalho do balanceador



Na Figura 1 mostramos a forma na qual o balanceador trabalha, Distribuindo de forma uniforme as requisições que chegam no servidor loadbalance HAProxy que distribuem as requisições entre as aplicações Moodle 1 e Moodle 2 que se conectam com o banco de dados PostgreSQL e o servidor NFS de compartilhamento de arquivos e diretórios.

## 1.1 Objetivos Gerais

Oferecer um aplicação de educação a distância com um excelente desempenho mesmo com recuso mais baixo e mais limitado utilizando os conceitos de balanceamento de cargas.

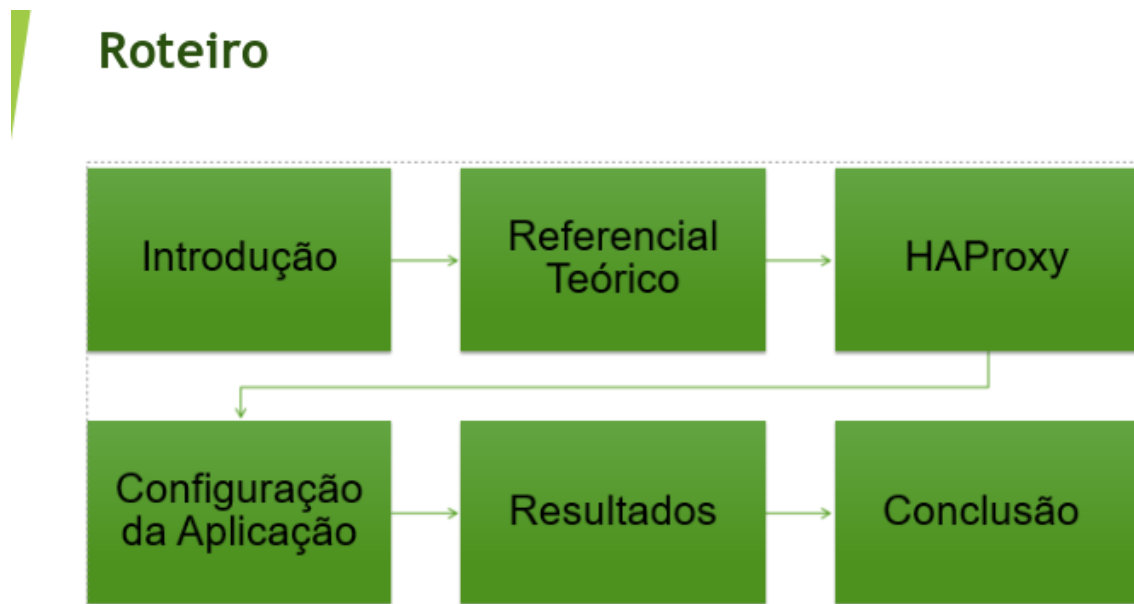
### 1.1.1 Objetivos Específicos

- Uma aplicação de educação de alto desempenho.
- Otimização de Recursos dos servidores
- Maximizar o desempenho da aplicação
- Minimizar o tempo de resposta a aplicação
- Evitar sobrecarga

## 1.2 Organizações dos capítulos

O trabalho é composto por 6 capítulos. O capítulo 1 apresenta a introdução e objetivos do trabalho. O capítulo 2 apresenta o referencial teórico utilizado neste projeto. O capítulo 3, é apresentado sobre o balanceador HAProxy. Já o capítulo 4 apresenta como foi feita toda a configuração da aplicação escolhida para configuração do projeto. O capítulo 5, são apresentadas os resultados obtidos no desenvolvimento do projeto. E, finalmente, no capítulo 6, são apresentadas as conclusões finais do projeto. Como mostra a figura 2.

Figura 2 – Roteiro



## 2 REFERENCIAL TEÓRICO

Neste capítulo é apresentado o referencial teórico que serviu como base de estudos para o desenvolvimento do trabalho de conclusão de curso.

### 2.1 MOODLE

O Moodle é uma plataforma de aprendizado online desenvolvida para oferecer aos educadores e alunos um sistema eficiente, seguro e para criar uma forma de aprendizado personalizado que. Fornecem milhares de meios de aprendizagem, é confiável para instituições e organizações de todos tamanhos. O Moodle possui mais de 90 milhões de usuários, em diversos setores da sociedade não só no academico, o que o torna a plataforma de aprendizado mais utilizado no mundo. Possui mais de 10 anos de desenvolvimento, e oferece conjunto de ferramentas muito completas, voltada para o aluno e os ambientes de aprendizagem auxiliando no aprendizado.(MOODLE, 2018)

O Moodle possui interface simples, recursos de arrastar e soltar que facilitam a utilização, recursos bem documentados, tornam o Moodle fácil de aprender e usar. O Moodle é oferecido gratuitamente como software *Open Source*, qualquer pessoa tem a liberdade de modificar o Moodle para projetos comerciais e não comerciais sem qualquer taxa. O Moodle possuir o código aberto o que significa que ele está continuamente passando por revisões e atualizações para atender às necessidades que vão aparecendo de acordo com a evolução dos usuários e das tecnologias.(MOODLE, 2018)

Devido ao fato do Moodle ser multilinguístico garante que não haja limitações para o aprendizado online, já que o moodle possui tradução para mais de 120 idiomas para que seus usuários tenha facilidade para localizar seu site Moodle, além de suporte, recursos em vários idiomas. O Moodle oferece muitas ferramentas flexíveis para auxiliar o aprendizado em cursos 100% online. O Moodle possibilita habilitar ou desabilitar os principais recursos e de uma forma bem simples integra tudo o que é necessário para um curso usando uma série de recursos integrados, também dispõem de fóruns para tirar qualquer duvida. De maneira que facilite e auxilie no aprendizado e adaptado às necessidades individuais de seus usuários.(MOODLE, 2018)

O Moodle possui a capacidade de um sistema informatizado ou não de se comunicar de forma transparente com outro sistema semelhante ou não. Permite aos desenvolvedores criem extensões e integrem aplicativos externos para obter funcionalidades específicas. Com os plugins gratuito, possibilita estender o Moodle de infinitas formas. O Moodle pode ser implementado para suportar de alguns alunos a milhões de usuários, suportando as necessidades de pequenas disciplinas e grandes organizações. Devido à sua flexibilidade

pode ser adaptado para uso educacional, empresarial, sem fins lucrativos e governamental. O Moodle se compromete com a segurança e a proteção dos dados e da privacidade do usuário, os controles de segurança estão sempre em atualização e desenvolvido para proteger contra acesso não autorizado, perda de dados e uso indevido.(MOODLE, 2018)

O Moodle pode facilmente ser implantado em uma nuvem privada ou servidor para controle total. O Moodle pode ser acessado de qualquer lugar do mundo, pois é baseado na web. Com uma interface compatível com dispositivos móveis e compatibilidade entre navegadores, o conteúdo é de fácil acesso e consistente em diversos navegadores. O Moodle disponibiliza extensas documentações e fóruns de usuários em vários idiomas, de forma gratuita e cursos compartilhados por usuários em todo o mundo. O projeto Moodle possui o apoio de uma comunidade internacional bem atuante e um grupo de desenvolvedores que dedicam tempo integral. Promovido pela colaboração desses grupo de desenvolvedores o projeto mantém correções e aprimoramentos de bugs rapidamente, com novos lançamentos a cada seis meses.(MOODLE, 2018)

## 2.2 PROXY

Um servidor proxy atua como intermediário entre um computador e a internet. Para enviar informações ou fazer uma requisição a uma página Web e receber a solicitação de volta para este computador é utilizado um endereço IP público. O IP, de *internet protocol*, é o protocolo de Internet responsável por fazer com que as informações consigam retornar ao seu local de origem. O endereço IP é fornecido pelo provedor de internet. O IP fornece a localização física do cliente as vezes até de forma bem precisa, mostrando as vezes até mesmo a rua em que o mesmo se encontra, com isso sites podem bloquear acesso se o cliente estiver fora da área estabelecida por ele. Seu nome e endereço residencial são vinculados ao seu endereço IP, pelo fato de ser um serviço pago com conexão a Internet em um local específico. Uma das formas de mudar essa realidade, é usar um serviço de proxy.(ADDRESS, 2018)

O proxy permite ficar online com uma identidade de endereço IP diferente da real. Não há necessidade de mudar o provedor de Internet, é necessário apenas procurar proxies gratuitos que fornecem listas de proxies gratuitos. Um servidor proxy é um servidor web que redireciona a nossa atividade de navegação. Quando é digitado um nome de site, o provedor de serviços de internet faz a solicitação e se conecta ao destino, revelando o endereço IP real. Quando um proxy é utilizado, as solicitações on-line são reencaminhadas. Usando um proxy, a solicitação de Internet vai do computador cliente para o provedor, de forma normal, mas depois é enviada para o servidor proxy e reencaminhada para o site de destino. O proxy utiliza o endereço IP escolhido na lista de proxy em sua configuração, mascarando seu endereço IP real.(SHIELD, 2018)

Algumas pessoas recorrem ao uso de um proxy por alguns motivos mas os principais

deles são a restrição de alguns serviços de streaming de vídeo que só podem ser exibido em determinados países, não possuir mais o acesso devido a localização geográfica, o bloqueio de acesso a determinados sites em alguma instituição particular ou privada, ou pelo simples fato de não ser rastreado até seu endereço IP e sua localização. O proxy possui diversas vantagens mas também podem trazer alguns riscos que devem ser levados em conta. Muitos serviços de Proxy gratuitos não possuem muitos recursos financeiros fazendo com que esses servidores proxy não possuam um bom hardware, software e pessoal necessário para desempenhar um serviço de qualidade com uma boa velocidade e segurança. Existem também uma infinidade de serviços de proxy gratuitos com o objetivo de interceptar suas informações, de forma maliciosa.(SHIELD, 2018)

## 2.3 APACHE

O servidor HTTP Apache foi desenvolvido com o objetivo de criar um código fonte compacto, com o mesmo nível de servidores comerciais, com uma infinidade de recursos completos de forma completamente gratuita de servidores HTTP. O projeto é gerenciado por um grupo de voluntários localizados em todo mundo, usando a internet para se comunicar, planejar, desenvolver e documentar todo desenvolvimento. Também são beneficiados com diversas ideias vindas dos usuário que contribuem com código e documentação para o projeto. Dessa forma todos contribuem para a melhora do apache.(APACHE, 2018)

O projeto Apache teve seu início depois da saída do criador do Daemon HTTP um dos software de servidor mais popular na Web e de domínio público na época e teve seu desenvolvimento parado. Com isso muitos *webmasters* desenvolveram suas próprias extensões e correções de *bugs* que precisavam de uma distribuição comum. uma pequena parte desse pequeno grupo de *webmasters* se reuniram com o objetivo de coordenar suas alterações criando o Apache. Foram adicionadas todas as correções de *bugs* publicadas e melhorias foram implementadas, também foram realizados testes para o lançamento do servidor Apache em abril de 1995. Nesta mesma época a equipe de desenvolvimento da NCSA retomou o projeto de desenvolvimento do Daemon.(APACHE, 2018)

Membros de desenvolvimento das duas equipes se uniram para compartilhar ideias e correções. O primeiro servidor Apache alcançou bastante sucesso, mas como todo software precisava de revisão e melhorias. Durante a segunda versão do servidor Apache enquanto uma parte da equipe de desenvolvimento novos recursos para serem implementados outra equipe criou uma nova arquitetura. Após uma série de testes beta, um nova documentação e uma série de recursos foram adicionados nos módulos padrões, dando forma ao Apache 1.0, lançado em dezembro de 1995. Em pouco menos de um ano o Apache passou o servidor http da NCSA, ficando como o servidor número um da internet, até os dias atuais. O grupo Apache em 1999 formaram a *Apache Software Foundation* fornecendo suporte organizacional, legal e financeiro. Com isso o software obteve uma base sólida de



desenvolvimento e expansão no número de projetos de código aberto.(APACHE, 2018)

## 2.4 NFS

NFS permite que um cliente remoto o acesse a rede de uma forma semelhante a um sistema de arquivos local. Apesar de não ser o primeiro desse tipo de sistema, o NFS alcançou um grande crescimento e evoluiu bastante, se tornando o sistema de arquivos de rede mais poderoso no momento. O NFS permite compartilhar um sistema de arquivos comum entre centenas de usuários e oferece a centralização de dados para minimizar o armazenamento necessário. O NFS foi o primeiro sistema de arquivos de rede moderno usando protocolo IP. No início era um sistema de arquivos experimental desenvolvido pela *Sun Microsystems*.(JONES, 2018)

Devido a popularidade o protocolo do NFS foi documentado e evoluiu para a versão NFSv2. O NFS cresceu rapidamente dada a sua capacidade de interoperar com outros clientes e servidores. Com a evolução contínua chegou na versão NFSv3. Essa versão era muito mais escalável que as versões anteriores, suportava arquivos grandes, gravações assíncronas, e TCP como protocolo de transporte, também abrindo caminho para os sistemas de arquivos de rede de longa distância. Em 2000, a Sun trouxe o NFS para o ambiente corporativo e apresentou o NFSv4, com muita segurança. Hoje, o NFS está na versão 4.1, adicionando suporte de protocolo para acesso paralelo por servidores distribuídos. O NFS encontra-se em desenvolvimento aproximadamente a 30 anos. Representando um sistema de arquivos em rede extremamente estável, escalável, possuindo alto desempenho e qualidade.(JONES, 2018)

O NFS é uma opção muito atraente para fornecer um sistema de arquivos em uma rede. Até em ambientes de rede locais, a virtualização leva o armazenamento para a rede para suportar mais máquinas virtuais remotas. O NFS suporta os modelos computacionais mais recentes para otimizar as infraestruturas virtualizadas. O NFS segue o modelo cliente/servidor, o servidor implementa o sistema de arquivos e os clientes se conectam ao armazenamento compartilhado. Os clientes implementam a interface com o usuário para o sistema de arquivo compartilhado. O comutador do sistema de arquivo virtual (VFS) do Linux fornece meios para suportar vários sistemas de arquivos simultaneamente. O VFS determina onde cada solicitação é armazenada e qual sistema de arquivos vai ser utilizado para cada solicitação.(JONES, 2018)

O NFS é um sistema de arquivos conectável como qualquer outro. A diferença com o NFS é que as solicitações de entrada e de saída podem não ser atendidas localmente, necessitando atravessar a rede para sua conclusão. Quando a solicitação é destinada para o NFS, o VFS passa a solicitação para a instância do NFS no kernel. Quando um procedimento é escolhido por uma solicitação de entrada e saída, ele é realizado na camada de chamada de procedimento remoto (RPC). A RPC é responsável por fornecer meios para

a realização de chamadas de procedimentos entre sistemas. A RPC inclui uma camada de interoperabilidade chamada *external data representation*(XDR), assegurando que todos os componentes do NFS se comuniquem com facilidade ao tratar todos os tipos de dados. Quando determinada arquitetura realiza uma solicitação, a representação do tipo de dado pode ser de um tipo diferente do host de destino que atende à solicitação.(JONES, 2018)

A XDR converte os tipos para a representação comum, possibilitando que as arquiteturas possam interoperar e compartilhar sistemas de arquivos. A XDR é mais conhecida por seu uso no NFS, mas sempre é útil quando estiver lidando com várias arquiteturas em um ambiente comum. Enquanto a XDR converte os dados para a representação comum, a solicitação é transferida pela rede, fornecendo um protocolo de camada de transporte. Os NFS mais antigos usavam o Universal Datagram Protocol (UDP), atualmente o TCP é o mais usado por proporcionar mais confiabilidade. O NFS trabalha de forma parecida. A solicitação percorre a pilha de rede, através das camadas RPC e XDR convertendo os tipos de dado para a arquitetura do servidor), para o servidor NFS. O servidor NFS é responsável por atender à solicitação, e logo é passada para o daemon do NFS, responsável por identificar a árvore do sistema de arquivos de destino necessária para a solicitação, novamente o VFS é usado para obter o sistema de arquivos no armazenamento local(JONES, 2018)

O NFS não é um sistema de arquivos tradicional, mas um protocolo para acessar sistemas de arquivos remotamente. o NFSv4 implementou o procedimento composto. Esse procedimento permite que várias chamadas RPC sejam incorporadas a uma única solicitação minimizando a taxa de transferência da solicitação pela rede. Também implementando um esquema de retorno de chamada para respostas. A primeira operação que ocorre no NFS é a chamada de montagem. Esse processo é encaminhado através do VFS para o componente do NFS. Após estabelecer o número da porta para a montagem, o cliente realiza uma solicitação RPC de montagem. Essa solicitação ocorre entre o cliente e um *daemon* especial responsável pelo protocolo de montagem. Nesse processo o *daemon* verifica a solicitação do cliente em relação à lista do servidor de sistemas de arquivos exportados atualmente, se o sistema solicitado de arquivos existir e o cliente tiver acesso, uma resposta de montagem estabelece o identificador de arquivos para o sistema de arquivos.(JONES, 2018)

No lado do cliente, ele armazena as informações remotas de montagem com o ponto de montagem local e estabelece a capacidade de realizar solicitações de entrada e saída. Esse protocolo representa um problema de segurança potencial, por isso, o NFSv4 substituiu esse protocolo de montagem auxiliar com chamadas RPC internas para gerenciar o ponto de montagem. Para ler um arquivo, o arquivo precisa primeiro ser aberto.(JONES, 2018)

## 2.5 BALANCEAMENTO DE CARGAS

O Balanceamento de cargas é um processo, grosso modo, que realiza a distribuição de trabalho entre servidores na mesma rede. O Balanceamento de cargas distribui as

solicitações, de forma que nenhum servidor venha ficar sobrecarregado, e caso isso ocorra, sendo o número de usuário maior que o suportado pelos servidores, o balanceador de carga possui uma flexibilidade para que mais servidores sejam adicionados de forma rápida. Com o balanceamento de carga não existe a necessidade de investir em servidores com um *hardware* muito potente. O Balanceamento de cargas implementado de forma inteligente é capaz de atingir um alto desempenho muitas vezes até melhor do que com um super servidor, que também é explicado pela alta disponibilidade que ele oferece devido ao fato de que se um servidor falhar, ele pode facilmente ser substituído, e as solicitações que vão chegando na rede são redirecionadas para os outros servidores ativos. (TELIUM, 2018)

## 2.6 POSTGRESQL

O PostgreSQL é um eficiente sistema de banco de dados objeto-relacional, possui código-fonte aberto que utiliza e estende a linguagem SQL combinada com muitos recursos que armazenam e dimensionam de forma segura as cargas de dados mais complexos. A origem do PostgreSQL é de 1986 que fazia parte do projeto POSTGRES da Universidade da Califórnia em Berkeley e possui mais de 30 anos de desenvolvimento. O PostgreSQL alcançou uma forte reputação devido sua arquitetura comprovada, confiabilidade, integridade de dados, conjunto robusto de recursos, extensibilidade e esforço da comunidade de código aberto, no desenvolvimento por trás do software para fornecer soluções eficazes e inovadoras. (POSTGRESQL, 2018)

O PostgreSQL é executado em diversos sistemas operacionais, e possui complementos eficientes, como o extensor de banco de dados geoespacial *PostGIS*. O PostgreSQL se tornou o banco de dados relacional de código aberto preferido por diversos programadores e organizações. O PostgreSQL é muito fácil de ser manuseado, basta escolher um projeto que você deseja construir e o PostgreSQL armazena seus dados de maneira segura e robusta. (POSTGRESQL, 2018)

O PostgreSQL vem acompanhado com uma infinidade de recursos reservados para ajudar os desenvolvedores a criar aplicativos, proteger a integridade dos dados e criar ambientes que suportam pequenas falhas, auxilia no gerenciamento de dados, independentemente do tamanho do conjunto de dados. Além de ser altamente extensível. O PostgreSQL possibilita definir seus próprios tipos de dados, criar funções personalizadas e escrever códigos de diferentes linguagens de programação sem a necessidade de recompilar seu banco de dados. (POSTGRESQL, 2018)

O PostgreSQL se esforça para estar sempre em conformidade com o padrão SQL, não contradizendo os recursos tradicionais ou levando a decisões arquitetônicas inadequadas. Muitos dos recursos exigidos pelo padrão SQL são suportados, às vezes com uma sintaxe ou função levemente diferente. Outros processos em direção à conformidade são esperados ao longo do tempo. Atualmente, nenhum outro banco de dados relacional atende à

conformidade total do SQL.(POSTGRESQL, 2018)

## 2.7 DEBIAN

O Debian foi criado por um grupo de desenvolvedores que tinha como objetivo criar um sistema operacional livre. O sistema operacional é o conjunto de programas básicos responsáveis pelo funcionamento do computador. O sistema operacional possui em seu núcleo um *kernel*. O *kernel* é o principal programa do computador sendo responsável por todas as operações básicas, permitindo que outros programas sejam executados. O sistema Debian utiliza o *kernel Linux* ou o *kernel FreeBSD*. (DEBIAN, 2018)

O *Linux* é um sistema operacional, criado por Linus Torvalds em conjunto com centenas de programadores separados por todo o mundo. O *FreeBSD* é um sistema operacional que inclui um *kernel* e outros softwares. Existe um trabalho em andamento para fornecer o Debian com outros *kernels*, principalmente com o *Hurd*. "O *Hurd* é um conjunto de servidores que rodam no topo de um micro *kernel*, os quais implementam diferentes características."(DEBIAN, 2018) O *Hurd* é um software livre produzido pelo projeto *GNU*. A maioria das ferramentas básicas que constituem o sistema operacional são originadas do projeto *GNU*. O Debian vem acompanhado com mais de 51000 pacotes pré-compilados, que facilita na instalação em sua máquina, um gerenciador de pacotes (APT) e utilitários para gerenciar milhares de pacotes em milhares de computadores de forma simples quanto instalar um único aplicativo. Todos os pacotes e utilitários são livres.(DEBIAN, 2018)

O sistema operacional funciona como uma torre: Na base está o kernel. Sobre o *Kernel* as ferramentas básicas e acima estão todos os outros softwares que você normalmente executa em seu computador. No topo está o Debian — organizando cuidadosamente as coisas, para que tudo funcione corretamente enquanto todos os componentes trabalham em conjunto. O Debian é livre por diversos motivos, alguns querem ajudar outras pessoas, muitos programam para aprender mais, outras estão procurando formas de evitar softwares caros, um grande grupo contribui por agradecimento a todo software livre de qualidade que receberam de outros. Muitos na universidade criam softwares para melhorar os resultados de suas pesquisas e para um melhor uso. Empresas ajudam a manter o software livre para explicar como ele é desenvolvido.(DEBIAN, 2018)

## 2.8 PROXMOX VIRTUAL ENVIRONMENT

O Proxmox *Virtual Environment* é uma poderosa plataforma de virtualização de servidores de código aberto para gerenciar duas tecnologias de virtualização, o KVM (Kernel-based Virtual Machine) para máquinas virtuais e a LXC(Linux Containers), com uma única interface baseada na web. O Proxmox possui armazenamento definido por software e funcionalidade de rede em uma única plataforma, e gerencia facilmente *clusters* de

alta disponibilidade. Os recursos 100% baseado em software fazem do Proxmox *Virtual Environment* a melhor opção para virtualizar sua infraestrutura de TI, otimizar os recursos existentes e aumentar a eficiência com um custo mínimo. Capaz de virtualizar facilmente até mesmo as cargas de trabalho de aplicativos Linux e Windows mais exigentes e dimensionar dinamicamente seu armazenamento conforme suas necessidades cresçam, garantindo a capacidade de se adaptar para o crescimento futuro de seu servidor.(GMBH, 2018)

## 2.9 Teste de Carga

Alguns fatores são de extrema importância para um site ser bem avaliado. Um dos principais fatores é a velocidade de navegação que o site proporciona, uma vez que nenhum usuário deseja navegar em um site lento, e dependendo da função que o site exerce, se ele for lento pode perder até mesmo dinheiro. Criar um site e testar em algum outro site que disponibiliza esse teste de velocidade é simples mas não é tão fácil afirmar com esse teste simples se o site é rápido e eficiente pois a capacidade do site deve ser criada com o número máximo de usuários possíveis em um intervalo de tempo curto. O teste de carga serve exatamente para testar a capacidade suportada em determinado intervalo de tempo definido, permitindo detectar falhas e prevenir os desenvolvedores para tentar aperfeiçoar o site. O teste de carga trabalha estressando os servidores de aplicação para testar como o site se comporta com grandes tráfegos, simulando ações padrões realizadas pelo usuário convencional.(LUCAS, 2018)

# 3 HAPROXY

## 3.1 Descrição

O HAProxy é uma software gratuito, rápido e seguro que oferece que sua aplicação(site) permaneça sempre online, balanceamento de carga e proxy para aplicativos baseados em TCP e HTTP. É apropriado para sites de tráfego intenso. Após muitos anos, tornou-se o balanceador de carga referência, agora é disponibilizado com a maioria dos sistemas operacionais e também está disponível em plataformas de nuvem. Só é possível saber se o HAProxy é utilizado quando quem implementou revela. Possui fácil integração com as arquiteturas existentes e sem riscos, e torna mais seguro não expondo servidores frágeis de forma direta à rede.(EDITION, 2018)

O HYProxy costuma oferecer no mínimo duas versões ativas atualizadas e uma mais antiga. Atualmente essa são as versões suportadas:

- versão 1.8 : *multi-threading, HTTP/2, cache, on-the fly server addition/removal, seamless reloads, DNS SRV, hardware SSL engines,*
- versão 1.7 : *added server hot reconfiguration, content processing agents, multi-type certs,*
- versão 1.6 : *added DNS resolution support, HTTP connection multiplexing, full stick-table replication, stateless compression,*
- versão 1.5 : *added SSL, IPv6, keep-alive, DDoS protection,*

## 3.2 Plataformas Suportadas

O HAProxy é executado com segurança nos seguintes sistemas operacionais:

- Linux 2.4 *on* x86, x86\_64, Alpha, Sparc, MIPS, PARISC
- Linux 2.6 / 3.x *on* x86, x86\_64, ARM, Sparc, PPC64
- Solaris 8/9 *on* UltraSPARC 2 *and* 3
- Solaris 10 *on* Opteron *and* UltraSPARC
- FreeBSD 4.10 - 10 *on* x86
- OpenBSD 3.1 *to* -current *on* i386, amd64, macppc, alpha, sparc64 *and* VAX (textit-check the ports)

- AIX 5.1 - 5.3 on Power<sup>TM</sup> architecture

Obtém-se o melhor desempenho com sistemas operacionais mais atualizados que suportam mecanismos de pesquisa escalonáveis, como o Linux 2.6 / 3.x ou o FreeBSD e no OpenBSD. Para isso ocorrer é necessário a versão haproxy mais recente que a 1.2.5. Para transferências mais rápidas de dados são possíveis com o uso do Linux 3.x e o haproxy 1.4 ou 1.5. Taxas de encaminhamento (número de pacotes de rede que podem ser processados de até 40 Gbps já foram alcançadas nessas plataformas após uma ótima configuração. Apesar do Solaris e o AIX ainda serem suportados, não é recomendado que os utilizem se for necessário um desempenho muito alto. (EDITION, 2018)

### 3.3 Atuação

A partir 2007 o desempenho do HAProxy aumentou consideravelmente e o hardware a cada dia que passa se torna muito mais eficaz. O HAProxy envolve uma série de procedimentos encontradas nas arquiteturas de sistemas operacionais para atingir o melhor desempenho possível.

O modelo baseado em evento e processo único são capaz de reduzir de forma considerável o custo a troca de contexto e o uso da memória. O processamento de centenas de tarefas em um milissegundo é possível, e o uso de memória é da ordem de alguns kilobytes por sessão, enquanto a memória consumida em servidores pré-configurados é mais da ordem de megabytes por processo.

O verificador de eventos em sistemas que permitem Linux e FreeBSD facilitam a detecção instantânea de qualquer evento em qualquer conexão entre dezenas de milhares.

Atualizações atrasadas no verificador de eventos e utilizando um cache de eventos preguiçosos garantem que nunca atualizemos um evento, a menos que seja absolutamente necessário, economizando chamadas do sistema.

*Single-buffering* sem nenhuma cópia de dados entre leituras e gravações sempre que possível. Economizando ciclos de CPU e largura de banda de memória útil. Muitas vezes, o gargalo será os barramentos de entrada e saída entre a CPU e as interfaces de rede. Em 10 a 100 Gbps, a largura de banda da memória também pode se tornar um gargalo.

O encaminhamento de cópia zero é possível usando a chamada de sistema no Linux, e resulta em cópia zero real começando com o Linux 3.5. Permitindo que um dispositivo pequeno encaminhe o tráfego HTTP em um gigabit por segundo.

Alocador de memória MRU usando *pools* de memória de tamanho fixo para alocação de memória imediata, favorece regiões de *hot cache* sobre as de *cold cache*. Reduzindo radicalmente o tempo necessário para criar uma nova sessão.

Work factoring com acesso múltiplo é a capacidade de limitar o número de acessos por iteração ao executar no modo multiprocesso, para distribuição de carga de forma uniforme entre os processos.

CPU-affinity é suportada durante a execução no modo multiprocesso, ou se adaptando ao hardware e ser o mais próximo possível do núcleo da CPU que gerencia as placas de rede, ainda que não exista conflito com elas.

Fazendo uso da Elastic Binary tree que é usado para manter os *timers* ordenados, a fila de execução ordenada.

Optimized timer queue: os *timers* não são movidos na árvore se forem adiados, porque a possibilidade de que sejam atendidos é próxima de zero, eles são usados principalmente para o tratamento de tempo limite. Otimizando ainda mais o uso do Elastic Binary tree.

Análise de cabeçalho HTTP otimizada: os cabeçalhos são interpretados em tempo real e a análise é otimizada evitando uma releitura de qualquer área de memória já realizada. Quando um fim de *buffer* é atingido com um cabeçalho incompleto, o ponto de verificação é usado para que a análise não comece novamente no início, quando mais dados, forem lidos.

Redução cuidadosa do número de chamadas caras do sistema. A maioria do trabalho é feita no espaço do usuário, como leitura de tempo, agregação de *buffer*, habilitação e desabilitação do descritor de arquivo.

A análise de conteúdo é otimizada para transportar apenas ponteiros para dados originais e nunca copiar, a menos que os dados precisem ser transformados. Garantindo que estruturas muito pequenas sejam transportadas e o conteúdo nunca seja copiado se não for necessário.

Todas essas pequenas otimizações resultam na diminuição do uso da CPU, mesmo em pequenas cargas. Quando trabalham com cargas muito altas, saturam a CPU, o processo HAProxy consome aproximadamente 20 vezes menos do que seu equivalente no sistema. Explicando porque é de extrema importância o ajuste do sistema operacional. O HAProxy foi instalado várias vezes como uma solução de emergência quando balanceadores de carga de hardware de ponta que falharam de repente. Alguns balanceadores de carga de hardware ainda não usam proxies e processam solicitações no nível do pacote e têm uma grande dificuldade em suportar solicitações em vários pacotes e altos tempos de resposta porque não armazenam nenhum buffer. Já os balanceadores de carga de software utilizam o *buffer* TCP e são insensíveis a solicitações longas e a tempos de resposta altos. Um efeito colateral positivo do *buffer* HTTP é que ele aumenta a aceitação da conexão do servidor, reduzindo a duração da sessão, deixando espaço para novas solicitações. (EDITION, 2018)

Três fatores são de extrema importância para avaliar o desempenho de um balanceador de carga:

A taxa de sessão é muito importante, pois determina quando o balanceador de carga não poderá distribuir todas as solicitações recebidas. É extremamente dependente da CPU. Pedidos com *keep-alive* ativado geralmente são muito mais altos, mas normalmente não fazem sentido para implementações direcionadas à Internet, porque os clientes costumam abrir uma grande quantidade de conexões e não enviam muitas solicitações por conexão.



Esse fator é calculado com tamanhos variáveis de objetos, sendo os resultados mais rápidos geralmente resultantes de objetos vazios. (EDITION, 2018)

A simultaneidade da sessão está ligado a taxa de sessão. Geralmente, a taxa de sessão diminui quando o número de sessões simultâneas aumentam. Quanto mais lento os servidores, maior o número de sessões simultâneas para uma mesma taxa de sessão. Caso um balanceador de carga receber 10000 sessões por segundo e os servidores responderem em 100 milissegundos, o balanceador de carga terá 1.000 sessões simultâneas. Esse número é limitado pela quantidade de memória e pela quantidade de descritores de arquivo que o sistema pode manipular. Os balanceadores de carga da camada 4 geralmente anunciam milhões de sessões simultâneas porque precisam lidar com os *soquetes TIME\_WAIT* que o sistema manipula gratuitamente em um proxy. também não processam nenhum dado, desse modo, não precisam de nenhum buffer. (EDITION, 2018)

A taxa de encaminhamento de dados é o oposto da taxa de sessão. Ela é medido em megabit por segundo ou em gigabit por segundo. As taxas de dados mais altas são alcançadas com objetos grandes para minimizar a sobrecarga causada pela configuração e desmontagem da sessão. Objetos grandes geralmente aumentam a simultaneidade da sessão e a alta simultaneidade de sessão com alta taxa de dados requer grandes quantidades de memória para suportar grandes janelas. As altas taxas de dados demandam de muitos ciclos de CPU e barramento em balanceadores de carga de software porque os dados precisam ser copiados da interface de entrada para a memória e, logo após, de volta para o dispositivo de saída. Os balanceadores de carga de hardware tendem a alternar os pacotes da porta de entrada para a porta de saída para uma taxa de dados mais alta, mas não podem processá-los.(EDITION, 2018)

O desempenho de um balanceador de carga está relacionado a esses fatores é normalmente para o melhor caso. Objetos vazios para taxa de sessão e objetos grandes para taxa de dados. Isso ocorre porque não é possível dizer exatamente como se comportará em cada combinação. Sendo assim, quando esses três limites são conhecidos, o cliente deve estar ciente de que, terá desempenho abaixo de todos eles. É bom considerar um desempenho médio de metade da sessão máxima e taxas de dados para objetos de tamanho médio.(EDITION, 2018)

## 3.4 Confiabilidade

A curto prazo é mais difícil projetar algo confiável a partir do zero, mas a longo prazo se mostra mais fácil para se manter do que um código quebrado e com muitos *bugs*. Em programas de processo único o menor bug irá travar seu programa, funcionando até que venha congelar. Não ocorreu nenhum bug desse tipo em versões estáveis nos últimos treze anos, apesar de ter ocorrido algumas vezes com o código de desenvolvimento em execução na seu desenvolvimento. O HAProxy foi instalado em sistemas *Linux* 2.4 que atendem

milhões de páginas diariamente e que só eram reiniciado em 3 anos para uma atualização completa do sistema operacional. No momento atual, o Haproxy está sendo utilizado em muitas empresas que estão na lista da revista *Fortune* 500 em todo o mundo oferecendo bilhões de páginas por diariamente de forma segura ou retransmitindo enormes quantias de dinheiro. Algumas pessoas confiam tanto HAProxy que o utilizam como solução padrão para resolver problemas simples. O HAProxy de fato é apropriado para esses cenários, porque os fatores retornados fornecem muitas informações importantes sobre a integridade, o comportamento e os defeitos do aplicativo, que são usados para que ele se tornar ainda mais confiável. (EDITION, 2018)

A maior parte do trabalho é executada pelo sistema operacional. Por isso, uma grande parte da confiabilidade envolve o próprio sistema operacional. As versões mais recentes do *Linux* 2.4 são conhecidas por oferecer o maior nível de estabilidade de todos os tempos. Contudo, ele solicita um monte de correções para alcançar um alto nível de desempenho, e este kernel está de fato desatualizado neste momento, então executá-lo em *hardware* moderno geralmente será difícil. O *Linux* 2.6 e 3.x incluem os recursos necessários para atingir esse nível de desempenho, mas versões antigas de LTS devem ser consideradas apenas para operações realmente estáveis sem atualizar mais de uma vez por ano. Alguns desenvolvedores preferem executá-lo no *Solaris*. O *Solaris* 8 e 9 são atualmente muito estáveis, oferecendo um nível de desempenho semelhante ao *Linux* 2.4. O *Solaris* 10 pode mostrar desempenho mais próximas do início do *Linux* 2.6. O *FreeBSD* mostra bom desempenho, mas o firewall consome metade dele e precisa ser desabilitado para se aproximar do *Linux*. O *OpenBSD* algumas vezes mostra falhas de alocação de *socket* devido a *socket* permanecerem no estado *FIN\_WAIT2* quando o cliente perde a conexão. A confiabilidade pode diminuir consideravelmente quando o sistema é levado aos seus extremos. Não há regra padrão, todo sistema e toda aplicação serão específicos. Um sistema corretamente ajustado deve ser capaz de funcionar por anos a plena carga sem desacelerar nem travar. (EDITION, 2018)

### 3.5 Segurança

A segurança é de extrema importância na implementação um balanceador de carga de software. Limitando o número de serviços e de portas abertas, mas o próprio balanceador de carga continua exposto. É extremamente difícil encontrar vulnerabilidades no HAProxy, pois sua arquitetura limita de forma significativa esse impacto e normalmente traz soluções fáceis de serem implementadas para melhorar a segurança. Por seu processamento se comportar até mesmo de forma imprevisível, torna muito difícil explorar com segurança qualquer *bug*, e se o processo falha, o *bug* é descoberto. Todos eles foram possíveis de serem encontrados através de análise reversa. De qualquer maneira, é bom ter muito cuidado ao escrever código para manipular os cabeçalhos. Todas as combinações de estado

são verificadas e retornadas, e os erros encontrados são processados desde a criação até a finalização de uma sessão. Um grupo de pessoas ao redor do mundo revisaram o código e sugeriram limpezas para uma melhor compreensão para facilitar a análise. É indicado que se inicie o HAProxy como *root* porque ele pode se prender em um chroot retirando todos os seus privilégios antes de iniciar as instâncias. Isso só é possível se o HAProxy for iniciado como *root*, pois somente o *root* pode executar o chroot, ao contrário do que alguns administradores acreditam. Os *logs* fornecem muitas informações para ajudar a manter um nível de segurança adequado. Eles são normalmente enviados via UDP porque, uma vez *chrooted*, o *socket UNIX/dev/log* é inacessível e não deve ser possível gravar em um arquivo.(EDITION, 2018)

As seguintes informações são particularmente úteis:

- O IP de origem e a porta do solicitante são encontrados em sua origem nos *logs* do *firewall*;
- A data de configuração da sessão geralmente corresponde aos *logs* do *firewall*, enquanto a data de desativação corresponde às datas dos proxies;
- A codificação de solicitação adequada garante que o solicitante não possa ocultar caracteres não imprimíveis nem burlar um terminal.
- Solicitação arbitrária e cabeçalho de resposta e captura de *cookie* ajudam a detectar ataques de varredura, *proxies* e *host* infectados. *Timers* ajudam a diferenciar solicitações digitadas manualmente de navegadores.

HAProxy também fornece controle de cabeçalho baseado em *regex*. Partes da solicitação, bem como cabeçalhos de solicitação e resposta podem ser negados, permitidos, removidos, reescritos ou adicionados. Isso é normalmente usado para bloquear solicitações ou codificações perigosas evitando vazamento acidental de informações do servidor para o cliente.(EDITION, 2018)

# 4 CONFIGURAÇÃO DA APLICAÇÃO

Neste capítulo está sendo apresentado algumas codificações que foram feitas nos servidores responsáveis pelo funcionamento do balanceador e da aplicação e tudo que foi necessário para seu funcionamento. Para esse trabalho foram utilizados cinco servidores um Servidor de arquivos NFS, um servidor HAProxy para o balanceamento e dois para a aplicação moodle e um para o banco de dados.

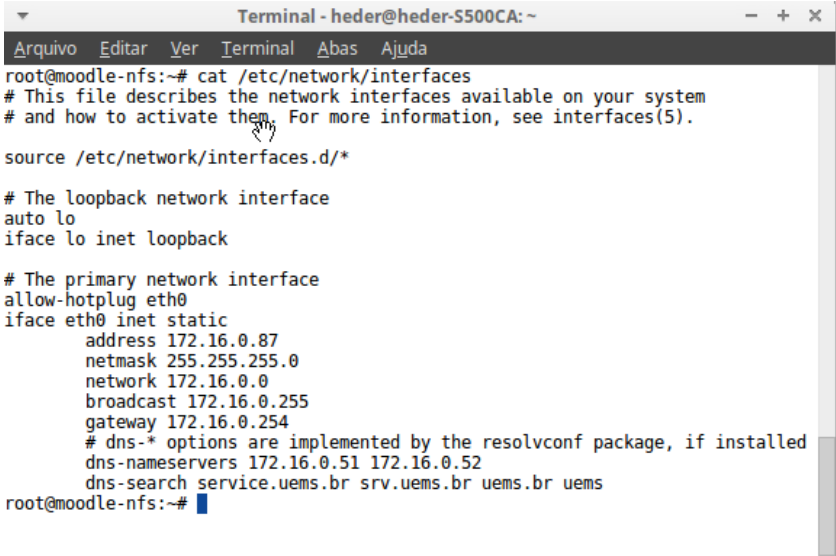
## 4.1 Servidor NFS

O servidor NFS é o servidor responsável pelo compartilhamento de arquivos e diretórios na rede.

### 4.1.1 Configurando interface de rede

Como mostra a figura 3 o arquivo interfaces localizado em `/etc/network/` foram configurados os IPs da placa de rede do servidor NFS.

Figura 3 – Arquivo Interface do servidor NFS



```
Terminal - heder@heder-S500CA: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
root@moodle-nfs:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

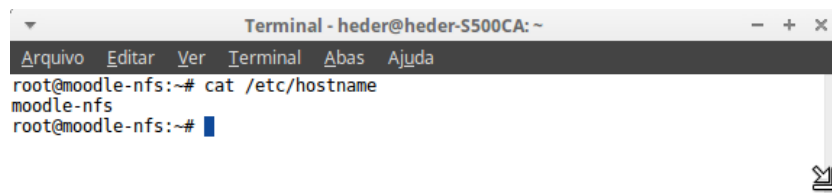
# The primary network interface
allow-hotplug eth0
iface eth0 inet static
    address 172.16.0.87
    netmask 255.255.255.0
    network 172.16.0.0
    broadcast 172.16.0.255
    gateway 172.16.0.254
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 172.16.0.51 172.16.0.52
    dns-search service.uems.br srv.uems.br uems.br uems
root@moodle-nfs:~#
```

### 4.1.2 Configurando Hostname

Para que uma máquina seja identificada na rede é necessária que ela possua um nome diferente, Hostname é o nome dado para o computador para que possa ser identificada

com mais facilidade do que por um IP como mostra a figura 4.

Figura 4 – Arquivo hostname do servidor NFS

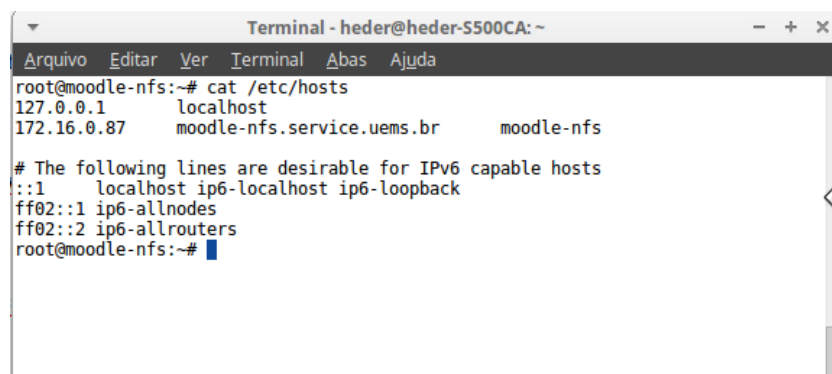


```
Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@moodle-nfs:~# cat /etc/hostname
moodle-nfs
root@moodle-nfs:~#
```

### 4.1.3 Configurando Hosts

Na figura 5 foi configurado o arquivo hosts responsável por relacionar hostnames e endereços IP.

Figura 5 – Arquivo hosts do servidor NFS



```
Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@moodle-nfs:~# cat /etc/hosts
127.0.0.1    localhost
172.16.0.87 moodle-nfs.service.uems.br moodle-nfs

# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
root@moodle-nfs:~#
```

### 4.1.4 Instalação de programas necessários

Nesta etapa foi feita a instalação de um ou mais programas para que o servidor NFS possa desempenhar sua função de maneira correta. Foi instalado o nfs-kernel-server.

### 4.1.5 Formatar a partição

A figura 6 mostra a formatação da partição que foi utilizada.

Figura 6 – Comando fdisk -l

```

Terminal - heder@heder-S500CA: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
root@moodle-nfs:~# fdisk -l

Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x62173547

/dev/sda1 *          2048  5859327  5857280   2,8G  83  Linux
/dev/sda2           5859328  7813119  1953792   954M  82  Linux swap / Solaris
/dev/sda3           7813120  14678015  6864896   3,3G  83  Linux

Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@moodle-nfs:~#

```

#### 4.1.6 Montar uma partição /var/moodledata em EXT4

Foi adicionado a partição criada na figura 6 no arquivo fstab como mostrado na imagem 7 na parte selecionado em vermelho.

Figura 7 – Arquivo fstab do servidor NFS

```

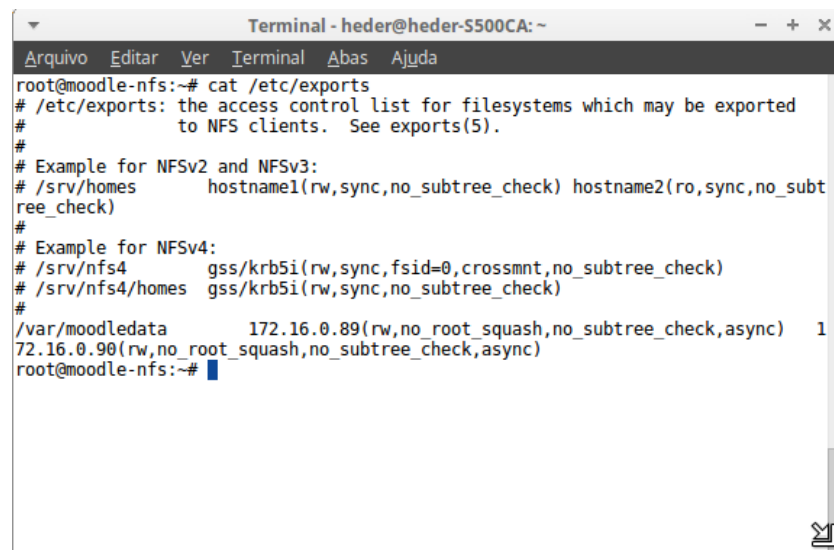
Terminal - heder@heder-S500CA: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
root@moodle-nfs:~# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>      <dump> <pass>
# / was on /dev/vda1 during installation
UUID=a95ce312-8935-4324-a6f0-e6d35e95a662 /          ext4    errors=remount
-ro 0      1
# /var was on /dev/vda3 during installation
UUID=7f7ab5c3-c234-428e-8c65-4663168572f9 /var      ext4    defaults
0         2
# swap was on /dev/vda2 during installation
UUID=0b2c64ce-bc28-4a25-8202-605366cbl3a none     swap    sw
0         0
/dev/sr0    /media/cdrom0  udf,iso9660 user,noauto 0      0
UUID=8e206615-06b4-4846-803e-674fd1bc198f /var/moodledata ext4    relatime
0         2
root@moodle-nfs:~#

```

#### 4.1.7 Configurar o servidor NFS

A imagem 8 mostra como foram configurados os arquivos e diretórios utilizados nos servidores Moodle 1 e Moodle 2.

Figura 8 – Arquivo exports do servidor NFS

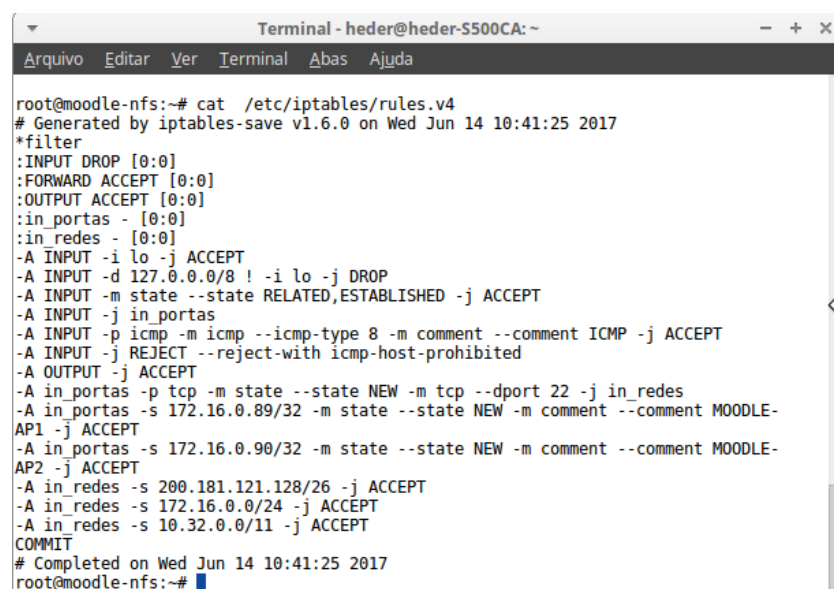


```
Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@moodle-nfs:~# cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/var/moodledata 172.16.0.89(rw,no_root_squash,no_subtree_check,async) 172.16.0.90(rw,no_root_squash,no_subtree_check,async)
root@moodle-nfs:~#
```

### 4.1.8 Configurando o Firewall

Na figura 9 mostra as regras de firewall que foram utilizadas e as solicitações que são aceitas e negadas para que o servidor NFS funcione com total segurança.

Figura 9 – Arquivo rules do servidor NFS



```
Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@moodle-nfs:~# cat /etc/iptables/rules.v4
# Generated by iptables-save v1.6.0 on Wed Jun 14 10:41:25 2017
*filter
:INPUT DROP [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:in_portas - [0:0]
:in_redes - [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 ! -i lo -j DROP
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j in_portas
-A INPUT -p icmp -m icmp --icmp-type 8 -m comment --comment ICMP -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A OUTPUT -j ACCEPT
-A in_portas -p tcp -m state --state NEW -m tcp --dport 22 -j in_redes
-A in_portas -s 172.16.0.89/32 -m state --state NEW -m comment --comment MOODLE-AP1 -j ACCEPT
-A in_portas -s 172.16.0.90/32 -m state --state NEW -m comment --comment MOODLE-AP2 -j ACCEPT
-A in_redes -s 200.181.121.128/26 -j ACCEPT
-A in_redes -s 172.16.0.0/24 -j ACCEPT
-A in_redes -s 10.32.0.0/11 -j ACCEPT
COMMIT
# Completed on Wed Jun 14 10:41:25 2017
root@moodle-nfs:~#
```

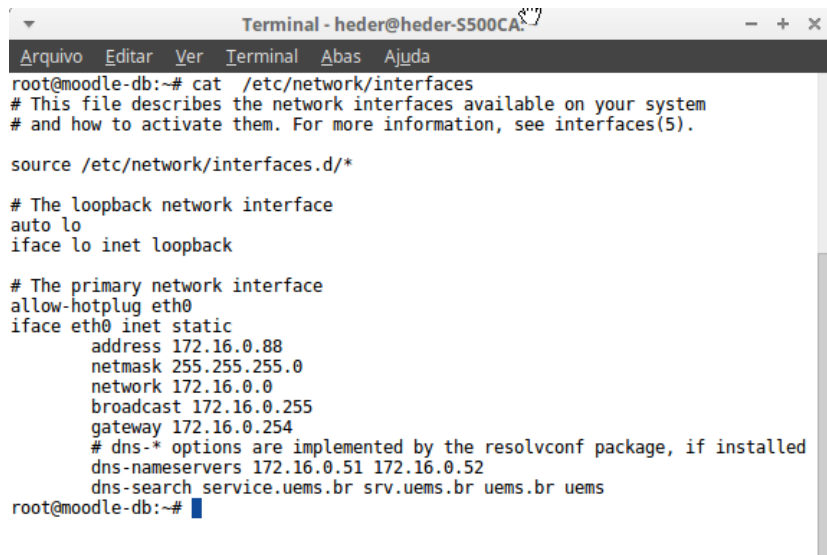
## 4.2 Servidor de Banco de dados

O Servidor de Banco de Dados é responsável pelo armazenamento das informações do Moodle.

### 4.2.1 Configurando interface de rede

Como mostra a figura 10 o arquivo interfaces localizado em `/etc/network/` foram configurados os IPs na placa de rede do servidor de Banco de dados.

Figura 10 – Arquivo Interface do servidor de Banco de dados

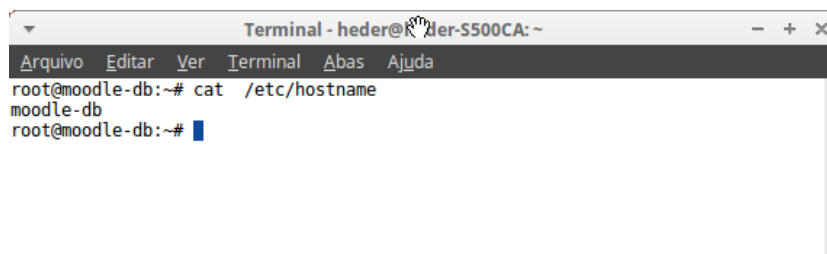


```
Terminal - heder@heder-S500CA:~  
Arquivo Editar Ver Terminal Abas Ajuda  
root@moodle-db:~# cat /etc/network/interfaces  
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).  
  
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
# The primary network interface  
allow-hotplug eth0  
iface eth0 inet static  
    address 172.16.0.88  
    netmask 255.255.255.0  
    network 172.16.0.0  
    broadcast 172.16.0.255  
    gateway 172.16.0.254  
    # dns-* options are implemented by the resolvconf package, if installed  
    dns-nameservers 172.16.0.51 172.16.0.52  
    dns-search service.uems.br srv.uems.br uems.br uems  
root@moodle-db:~#
```

### 4.2.2 Configurando Hostname

Para que uma máquina seja identificada na rede é necessária que ela possua um nome diferente, Hostname é o nome dado para o computador para que possa ser identificada com mais facilidade do que por um IP como mostra a figura 11.

Figura 11 – Arquivo hostname do servidor de Banco de dados



```
Terminal - heder@heder-S500CA:~  
Arquivo Editar Ver Terminal Abas Ajuda  
root@moodle-db:~# cat /etc/hostname  
moodle-db  
root@moodle-db:~#
```

### 4.2.3 Configurando Hosts

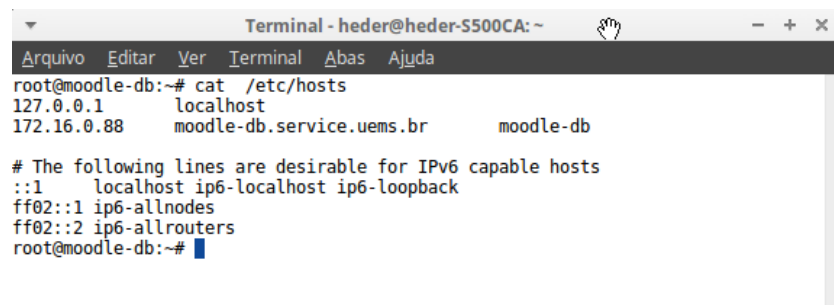
Na figura 12 foi configurado o arquivo hosts responsável por relacionar hostnames e endereços IP.

### 4.2.4 Instalação de programas necessários

Nesta etapa foi feita a instalação de um ou mais programas para que o servidor de banco de dados possa desempenhar sua função de maneira correta. Foi instalado o postgresql.



Figura 12 – Arquivo hosts do servidor de Banco de dados



```

Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@moodle-db:~# cat /etc/hosts
127.0.0.1      localhost
172.16.0.88   moodle-db.service.uems.br   moodle-db

# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
root@moodle-db:~#

```

### 4.2.5 Criando usuário para o moodle

Apos a instalação do postgresql foi criado o usuário e senha para o Moodle e o banco de dados que foi utilizado como mostram os comandos abaixo.

```

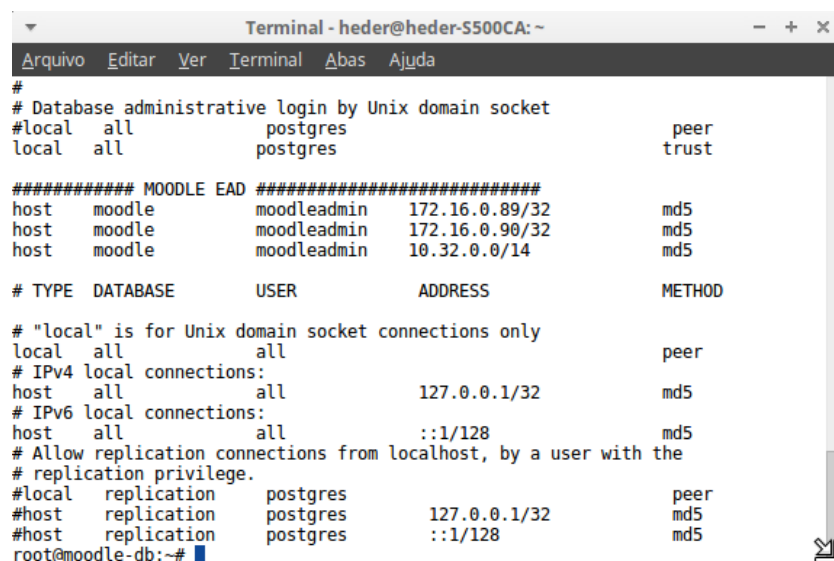
# psql -U postgres
postgres=# CREATE USER moodleadadmin WITH PASSWORD 'Hedxxx17';
postgres=# CREATE DATABASE moodle WITH OWNER moodleadadmin;
postgres=# \f

```

### 4.2.6 Configurando o postgresql

Na figura 13 mostra como foi feita a ligação dos IPs dos servidores Moodle 1 e Moodle 2 no postgresql no arquivo pg\_hba.conf.

Figura 13 – Arquivo pg\_hba.conf



```

Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
#
# Database administrative login by Unix domain socket
#local all postgres peer
local all postgres trust

##### MOODLE EAD #####
host moodle moodleadadmin 172.16.0.89/32 md5
host moodle moodleadadmin 172.16.0.90/32 md5
host moodle moodleadadmin 10.32.0.0/14 md5

# TYPE DATABASE USER ADDRESS METHOD

# "local" is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local replication postgres peer
#host replication postgres 127.0.0.1/32 md5
#host replication postgres ::1/128 md5
root@moodle-db:~#

```

Foram definidos os parâmetros no arquivo postgresql.conf, que normalmente é mantido no diretório de dados. Como mostra a figura 14.

Figura 14 – Arquivo postgresql.conf

```

# CONNECTIONS AND AUTHENTICATION
#-----
# - Connection Settings -

listen_addresses = '*'
#listen_addresses = 'localhost'          # what IP address(es) to listen on;
#                                          # comma-separated list of addresses;
#                                          # defaults to 'localhost'; use '*' for a
ll
#                                          # (change requires restart)
port = 5432                               # (change requires restart)
max_connections = 100                     # (change requires restart)
#superuser_reserved_connections = 3      # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of direct
ories
#                                          # (change requires restart)
#unix_socket_group = ''                  # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
#                                          # (change requires restart)
#bonjour = off                           # advertise server via Bonjour
#                                          # (change requires restart)
#bonjour_name = ''                       # defaults to the computer name
#                                          # (change requires restart)

```

## 4.2.7 Configurando o Firewall

Na figura 15 mostra as regras de firewall que foram utilizadas e as solicitações que são aceitas e negadas para que o servidor de Banco de dados funcione com total segurança.

Figura 15 – Arquivo rules do servidor de Banco de dados

```

root@moodle-db:~# cat /etc/iptables/rules.v4
# Generated by iptables-save v1.6.0 on Fri Jun 16 00:55:39 2017
*filter
:INPUT DROP [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:in_portas - [0:0]
:in_redes - [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 ! -i lo -j DROP
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j in_portas
-A INPUT -p icmp -m icmp --icmp-type 8 -m comment --comment ICMP -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A OUTPUT -j ACCEPT
-A in_portas -p tcp -m state --state NEW -m tcp --dport 22 -j in_redes
-A in_portas -p tcp -m state --state NEW -m tcp --dport 5432 -m comment --comment PostgreSQL -j in_redes
-A in_redes -s 10.32.0.0/11 -j ACCEPT
-A in_redes -s 172.16.0.0/24 -j ACCEPT
-A in_redes -s 200.181.121.128/26 -j ACCEPT
COMMIT
# Completed on Fri Jun 16 00:55:39 2017
root@moodle-db:~#

```

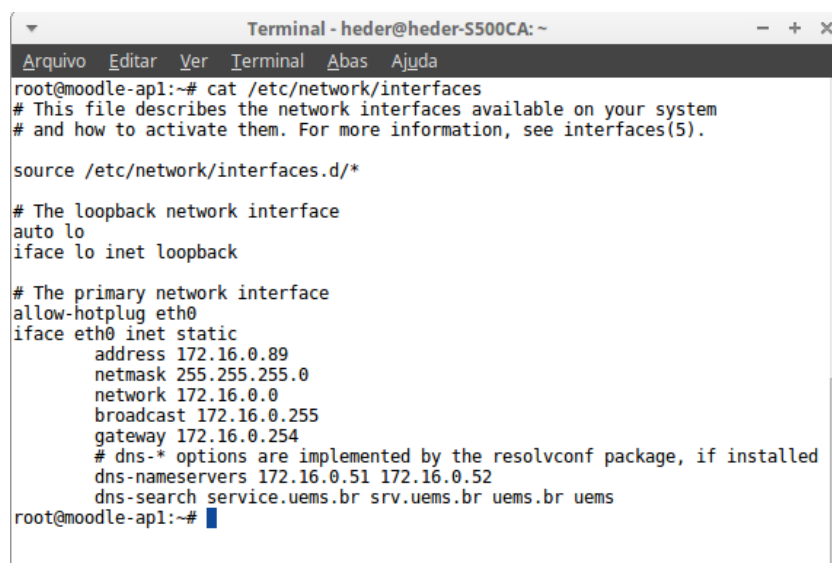
## 4.3 Servidor Moodle 1

O servidor Moodle 1 é um dos servidores de aplicação que foi balanceado pelo servidor Loadbance.

### 4.3.1 Configurando interface de rede

Como mostra a figura 16 o arquivo interfaces localizado em `/etc/network/` foram configurados os IPs na placa de rede do servidor Moodle 1.

Figura 16 – Arquivo Interface do servidor Moodle 1



```
Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@moodle-ap1:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

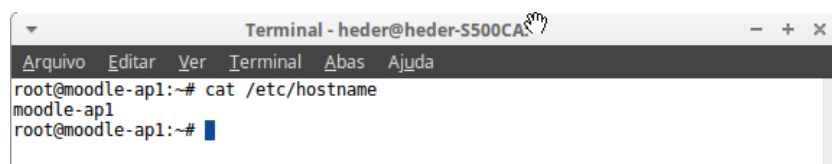
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
    address 172.16.0.89
    netmask 255.255.255.0
    network 172.16.0.0
    broadcast 172.16.0.255
    gateway 172.16.0.254
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 172.16.0.51 172.16.0.52
    dns-search service.uems.br srv.uems.br uems.br uems
root@moodle-ap1:~#
```

### 4.3.2 Configurando Hostname

Para que uma máquina seja identificada na rede é necessária que ela possua um nome diferente, Hostname é o nome dado para o computador para que possa ser identificada com mais facilidade do que por um IP como mostra a figura 17.

Figura 17 – Arquivo hostname do servidor da aplicação Moodle 1

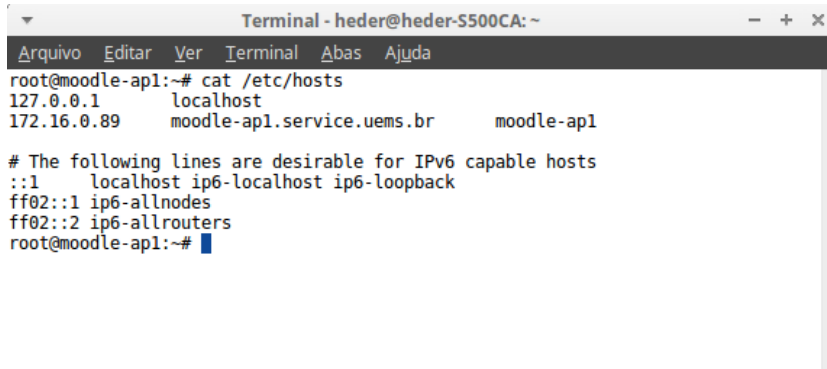


```
Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@moodle-ap1:~# cat /etc/hostname
moodle-ap1
root@moodle-ap1:~#
```

### 4.3.3 Configurando Hosts

Na figura 18 foi configurado o arquivo hosts responsável por relacionar hostnames e endereços IP.

Figura 18 – Arquivo hosts do servidor Moodle 1



```

Terminal - heder@heder-S500CA: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
root@moodle-ap1:~# cat /etc/hosts
127.0.0.1    localhost
172.16.0.89 moodle-ap1.service.uems.br    moodle-ap1

# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
root@moodle-ap1:~#

```

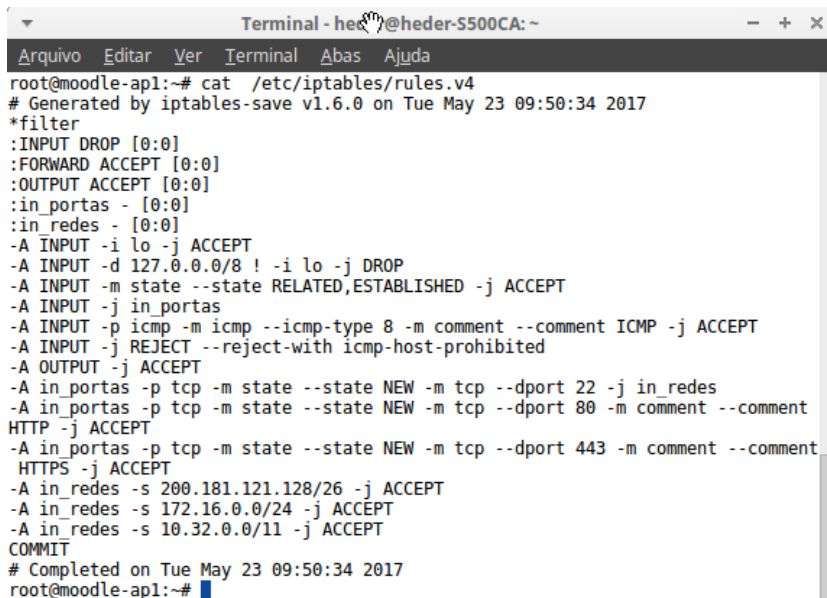
#### 4.3.4 Instalação de programas necessários

Nesta etapa foram feitas a instalação de um ou mais programas para que o servidor Moodle 1 possa desempenhar sua função de maneira correta. Foram instalados o Apache2, o PHP e o NFS-common.

#### 4.3.5 Configurando o Firewall

Na figura 19 mostra as regras de firewall que foram utilizadas e as solicitações que são aceitas e negadas para que o servidor Moodle 1 funcione com total segurança.

Figura 19 – Arquivo rules do servidor Moodle 1



```

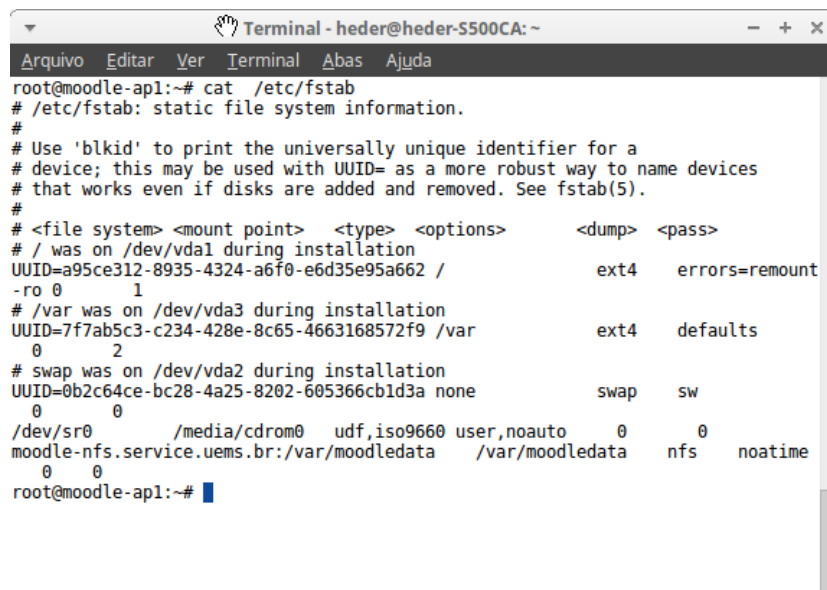
Terminal - heder@heder-S500CA: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
root@moodle-ap1:~# cat /etc/iptables/rules.v4
# Generated by iptables-save v1.6.0 on Tue May 23 09:50:34 2017
*filter
:INPUT DROP [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:in_portas - [0:0]
:in_redes - [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 ! -i lo -j DROP
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j in_portas
-A INPUT -p icmp -m icmp --icmp-type 8 -m comment --comment ICMP -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A OUTPUT -j ACCEPT
-A in_portas -p tcp -m state --state NEW -m tcp --dport 22 -j in_redes
-A in_portas -p tcp -m state --state NEW -m tcp --dport 80 -m comment --comment HTTP -j ACCEPT
-A in_portas -p tcp -m state --state NEW -m tcp --dport 443 -m comment --comment HTTPS -j ACCEPT
-A in_redes -s 200.181.121.128/26 -j ACCEPT
-A in_redes -s 172.16.0.0/24 -j ACCEPT
-A in_redes -s 10.32.0.0/11 -j ACCEPT
COMMIT
# Completed on Tue May 23 09:50:34 2017
root@moodle-ap1:~#

```

#### 4.3.6 Montando partição Moodledata do servidor Moodle1

A figura 20 mostra como foi feita a ligação do servidor Moodle1 com o servidor NFS para o compartilhamento de arquivos e diretórios em rede.

Figura 20 – Arquivo fstab do servidor Moodle 1



```

Terminal - heder@heder-S500CA: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
root@moodle-ap1:~# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/vda1 during installation
UUID=a95ce312-8935-4324-a6f0-e6d35e95a662 /
-ro 0 1
# /var was on /dev/vda3 during installation
UUID=7f7ab5c3-c234-428e-8c65-4663168572f9 /var
0 2
# swap was on /dev/vda2 during installation
UUID=0b2c64ce-bc28-4a25-8202-605366cb1d3a none
0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
moodle-nfs.service.uems.br:/var/moodledata /var/moodledata nfs noatime
0 0
root@moodle-ap1:~#

```

### 4.3.7 Instalação do Moodle pelo terminal

As etapas abaixo foram responsáveis por realizar o download do moodle e acertar as permissões para instalação.

```
cd /var/www/html
```

```
wget https://download.moodle.org/download.php/direct/stable35/moodle-latest-35.tgz
```

```
tar -zxvf moodle-latest-35.tgz
```

```
chown -R www-data /var/moodledata
```

```
chmod -R 0770 /var/moodledata
```

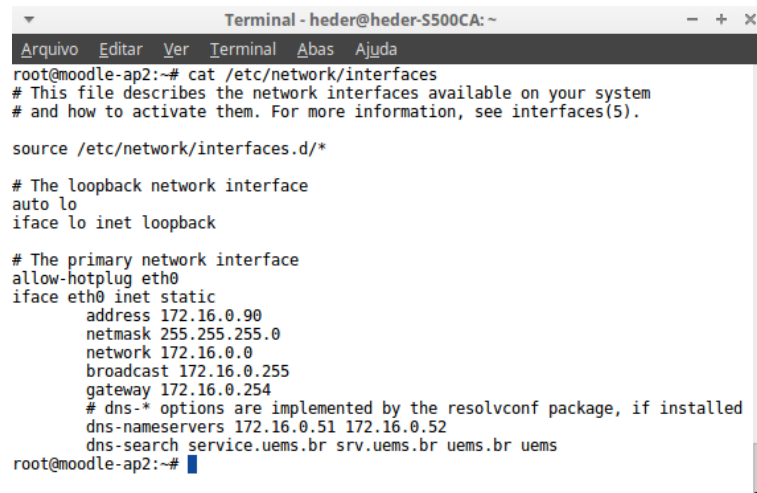
## 4.4 Servidor Moodle 2

O servidor Moodle 2 é um dos servidores de aplicação que foi balanceado pelo servidor Loadbalance a diferença para o Moodle 1 é apenas o IP.

### 4.4.1 Configurando interface de rede

Como mostra a figura 21 o arquivo interfaces localizado em /etc/network/ foram configurados os IPs na placa de rede do servidor Moodle 2.

Figura 21 – Arquivo Interface do servidor Moodle 2



```
Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@moodle-ap2:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

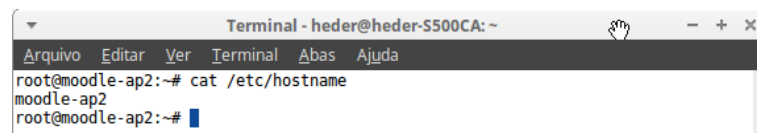
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
    address 172.16.0.90
    netmask 255.255.255.0
    network 172.16.0.0
    broadcast 172.16.0.255
    gateway 172.16.0.254
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 172.16.0.51 172.16.0.52
    dns-search service.uems.br srv.uems.br uems.br uems
root@moodle-ap2:~#
```

#### 4.4.2 Configurando Hostname

Para que uma máquina seja identificada na rede é necessária que ela possua um nome diferente, Hostname é o nome dado para o computador para que possa ser identificada com mais facilidade do que por um IP como mostra a figura 22.

Figura 22 – Arquivo hostname do servidor Moodle 2

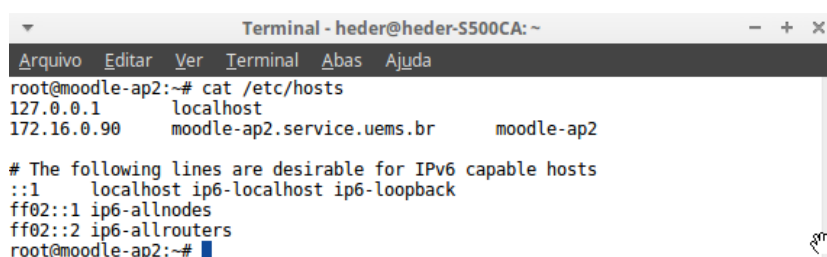


```
Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@moodle-ap2:~# cat /etc/hostname
moodle-ap2
root@moodle-ap2:~#
```

#### 4.4.3 Configurando Hosts

Na figura 23 foi configurado o arquivo hosts responsável por relacionar hostnames e endereços IP.

Figura 23 – Arquivo hosts do servidor Moodle 2



```
Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@moodle-ap2:~# cat /etc/hosts
127.0.0.1 localhost
172.16.0.90 moodle-ap2.service.uems.br moodle-ap2

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
root@moodle-ap2:~#
```

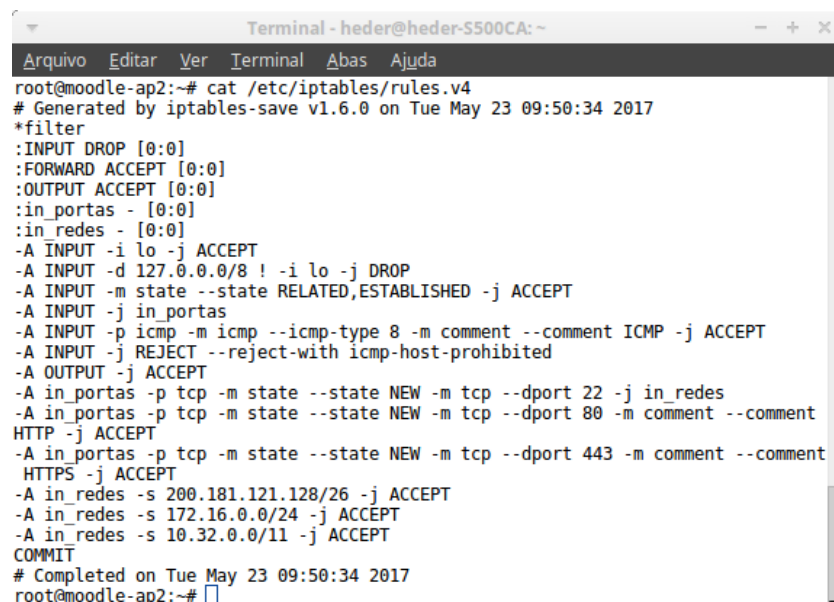
#### 4.4.4 Instalação de programas necessários

Nesta etapa foi feita a instalação de um ou mais programas para que o servidor Moodle 2 possa desempenhar sua função de maneira correta. Foram instalados o Apache2, o PHP e o NFS-common.

#### 4.4.5 Configurando o Firewall

Na figura 24 mostra as regras de firewall que foram utilizadas e as solicitações que são aceitas e negadas para que o servidor Moodle 2 funcione com total segurança.

Figura 24 – Arquivo rules do servidor Moodle 2



```

Terminal - heder@heder-S500CA: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
root@moodle-ap2:~# cat /etc/iptables/rules.v4
# Generated by iptables-save v1.6.0 on Tue May 23 09:50:34 2017
*filter
:INPUT DROP [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:in_portas - [0:0]
:in_redes - [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 ! -i lo -j DROP
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j in_portas
-A INPUT -p icmp -m icmp --icmp-type 8 -m comment --comment ICMP -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A OUTPUT -j ACCEPT
-A in_portas -p tcp -m state --state NEW -m tcp --dport 22 -j in_redes
-A in_portas -p tcp -m state --state NEW -m tcp --dport 80 -m comment --comment
HTTP -j ACCEPT
-A in_portas -p tcp -m state --state NEW -m tcp --dport 443 -m comment --comment
HTTPS -j ACCEPT
-A in_redes -s 200.181.121.128/26 -j ACCEPT
-A in_redes -s 172.16.0.0/24 -j ACCEPT
-A in_redes -s 10.32.0.0/11 -j ACCEPT
COMMIT
# Completed on Tue May 23 09:50:34 2017
root@moodle-ap2:~#

```

#### 4.4.6 Montando partição Moodledata do servidor Moodle2

A figura 25 mostra como foi feita a ligação do servidor Moodle2 com o servidor NFS para o compartilhamento de arquivos e diretórios em rede.

#### 4.4.7 Instalação do Moodle pelo terminal

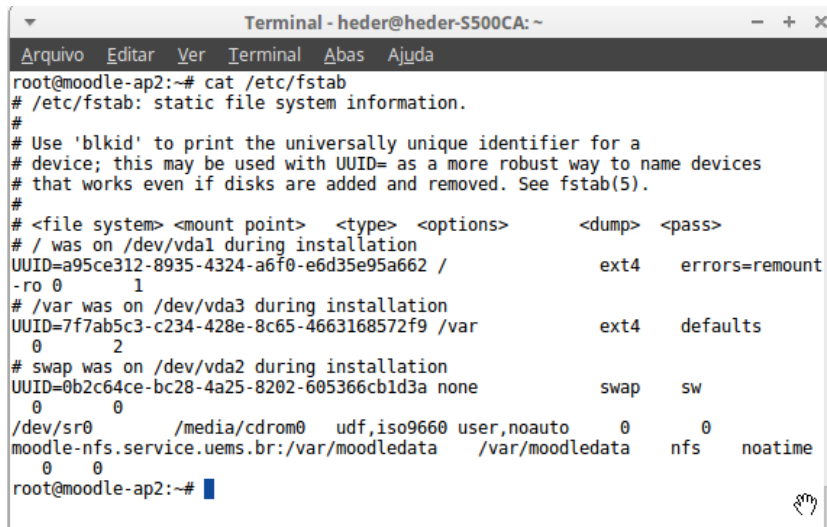
As etapas abaixo foram responsáveis por realizar o download do moodle e acertar as permissões para instalação.

```

cd /var/www/html
wget https://download.moodle.org/download.php/direct/stable35/moodle-latest-35.tgz
tar -zxvf moodle-latest-35.tgz
chown -R www-data /var/moodledata
chmod -R 0770 /var/moodledata

```

Figura 25 – Arquivo fstab do servidor Moodle2



```

Terminal - heder@heder-S500CA: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
root@moodle-ap2:~# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/vda1 during installation
UUID=a95ce312-8935-4324-a6f0-e6d35e95a662 / ext4 errors=remount
-ro 0 1
# /var was on /dev/vda3 during installation
UUID=7f7ab5c3-c234-428e-8c65-4663168572f9 /var ext4 defaults
0 2
# swap was on /dev/vda2 during installation
UUID=0b2c64ce-bc28-4a25-8202-605366cb1d3a none swap sw
0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
moodle-nfs.service.uems.br:/var/moodledata /var/moodledata nfs noatime
0 0
root@moodle-ap2:~#

```

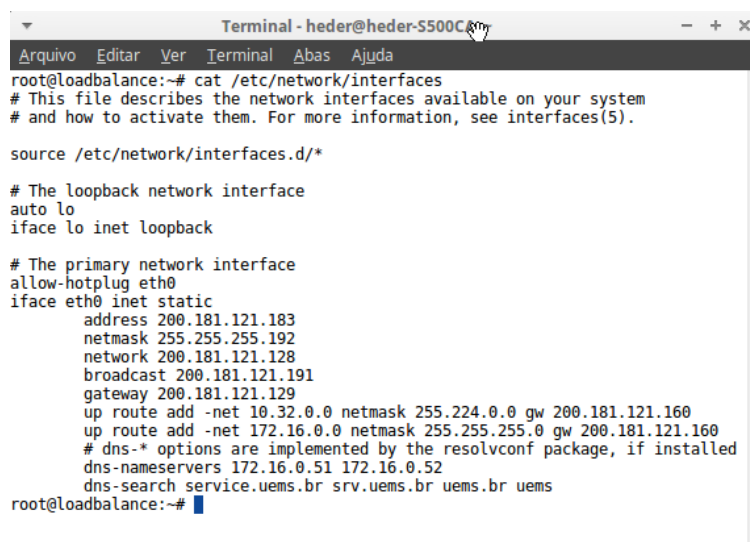
## 4.5 Servidor LOADBALANCE

O servidor Loadbalance foi o responsável por fazer o balanceamento dos servidores Moodle 1 e Moodle 2.

### 4.5.1 Configurando interface de rede

Como mostra a figura 26 o arquivo interfaces localizado em /etc/network/ foram configurados os IPs na placa de rede do servidor loadbalance.

Figura 26 – Arquivo Interface



```

Terminal - heder@heder-S500C
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
root@loadbalance:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
    address 200.181.121.183
    netmask 255.255.255.192
    network 200.181.121.128
    broadcast 200.181.121.191
    gateway 200.181.121.129
    up route add -net 10.32.0.0 netmask 255.224.0.0 gw 200.181.121.160
    up route add -net 172.16.0.0 netmask 255.255.255.0 gw 200.181.121.160
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 172.16.0.51 172.16.0.52
    dns-search service.uems.br srv.uems.br uems.br uems
root@loadbalance:~#

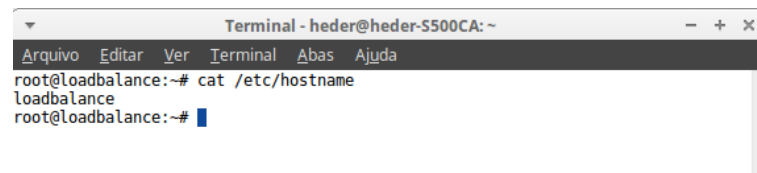
```



## 4.5.2 Configurando Hostname

Para que uma máquina seja identificada na rede é necessária que ela possua um nome diferente, Hostname é o nome dado para o computador para que possa ser identificada com mais facilidade do que por um IP como mostra a figura 27.

Figura 27 – Arquivo hostname

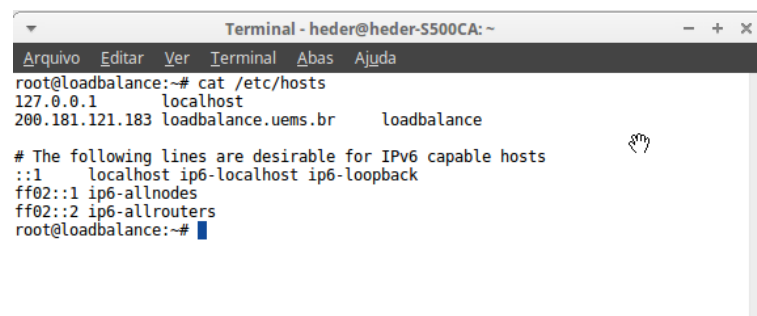


```
Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@loadbalance:~# cat /etc/hostname
loadbalance
root@loadbalance:~#
```

## 4.5.3 Configurando Hosts

Na figura 28 foi configurado o arquivo hosts responsável por relacionar hostnames e endereços IP.

Figura 28 – Arquivo hosts



```
Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@loadbalance:~# cat /etc/hosts
127.0.0.1 localhost
200.181.121.183 loadbalance.uems.br loadbalance

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
root@loadbalance:~#
```

## 4.5.4 Instalação o HAProxy

Nesta etapa foi instalando o HAProxy pelo terminal no repositório do linux.

## 4.5.5 Configurando o Firewall

Na figura 29 mostra as regras de firewall que foram utilizadas e as solicitações que são aceitas e negadas para que o servidor Loadbalance funcione com total segurança.

Figura 29 – Arquivo rules

```

Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@loadbalance:~# cat /etc/iptables/rules.v4
# Generated by iptables-save v1.6.0 on Fri May 26 10:28:01 2017
*filter
:INPUT DROP [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:fw_portas - [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 ! -i lo -j DROP
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j fw_portas
-A INPUT -p icmp -m icmp --icmp-type 8 -m comment --comment ICMP -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A OUTPUT -j ACCEPT
-A fw_portas -s 172.16.0.0/24 -p tcp -m state --state NEW -m tcp --dport 22 -m comment --comment SSH -j ACCEPT
-A fw_portas -s 200.181.121.128/26 -p tcp -m state --state NEW -m tcp --dport 22 -m comment --comment SSH -j ACCEPT
-A fw_portas -s 10.32.0.0/11 -p tcp -m state --state NEW -m tcp --dport 22 -m comment --comment SSH -j ACCEPT
-A fw_portas -p tcp -m state --state NEW -m tcp --dport 80 -m comment --comment HTTP -j ACCEPT
-A fw_portas -p tcp -m state --state NEW -m tcp --dport 443 -m comment --comment HTTPS -j ACCEPT
COMMIT
# Completed on Fri May 26 10:28:01 2017
root@loadbalance:~#

```

#### 4.5.6 Configurando o HAProxy

As figuras 30 e 31 mostram as configurações foram que feitas no HAProxy como uso do balanceador *roundrobin*, o uso de *cookies* para o uso de sessão permanente.

Figura 30 – Arquivo haproxy.cfg parte1

```

Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@loadbalance:~# cat /etc/haproxy/haproxy.cfg
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # Default ciphers to use on SSL-enabled listening sockets.
    # For more information, see ciphers(1SSL). This list is from:
    # https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/
    # An alternative list with additional directives can be obtained from
    # https://mozilla.github.io/server-side-tls/ssl-config-generator/?server=haproxy
    ssl-default-bind-ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS
    ssl-default-bind-options no-sslv3

defaults
    log          global
    mode        http
    option      httplog
    option      dontlognull
    timeout    connect 5000
    timeout    client  50000
    timeout    server  50000
    errorfile  400 /etc/haproxy/errors/400.http
    errorfile  403 /etc/haproxy/errors/403.http

```

Figura 31 – Arquivo haproxy.cfg parte2

```
Terminal - heder@heder-S500CA: ~
Arquivo Editar Ver Terminal Abas Ajuda
ssl-default-bind-options no-sslv3

defaults
    log          global
    mode         http
    option       httplog
    option       dontlognull
    timeout     connect 5000
    timeout     client 50000
    timeout     server 50000
    errorfile   400 /etc/haproxy/errors/400.http
    errorfile   403 /etc/haproxy/errors/403.http
    errorfile   408 /etc/haproxy/errors/408.http
    errorfile   500 /etc/haproxy/errors/500.http
    errorfile   502 /etc/haproxy/errors/502.http
    errorfile   503 /etc/haproxy/errors/503.http
    errorfile   504 /etc/haproxy/errors/504.http

frontend HA entrada
    bind *:80
    stats uri /loadbalanceinfo
    stats auth heder:uems

##### ACL Moodle - EAD #####
acl moodle_hdr(host) -i moodle-ead.uems.br
use_backend moodle_saida if moodle

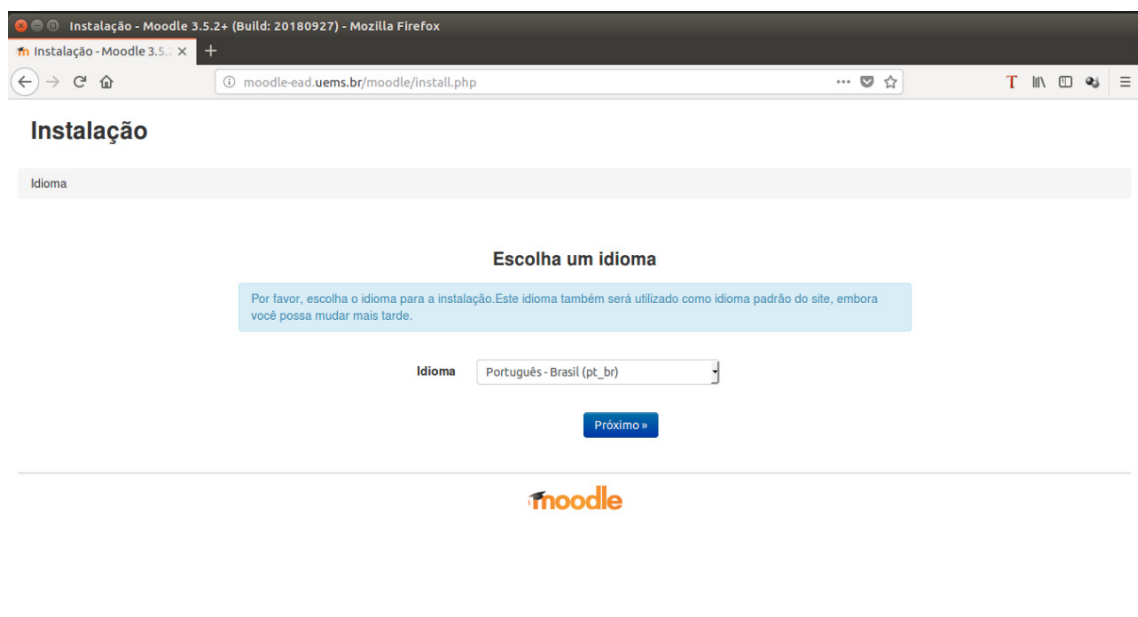
backend moodle_saida
    balance roundrobin
    mode http
    cookie SERVERID insert indirect nocache
    server moodle-ap1 moodle-ap1.service.uems.br:80 check cookie moodle-ap1
    server moodle-ap2 moodle-ap2.service.uems.br:80 check cookie moodle-ap2

root@loadbalance:~#
```

### 4.5.7 Instalação via *Browser*

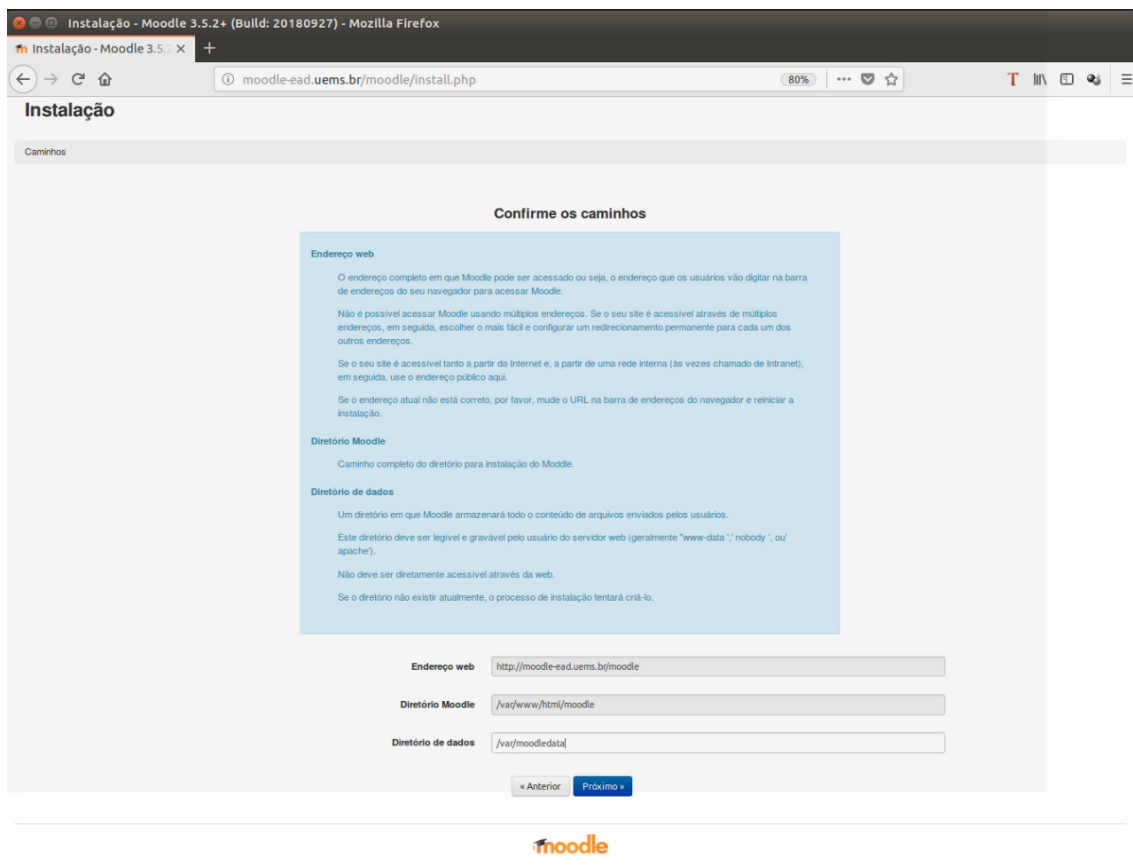
Na figura 32 foi feita a configuração do Moodle pelo *Browser* e nesse primeiro passo foi feita a escolha do idioma que foi utilizado.

Figura 32 – Instalação Moodle passo 1



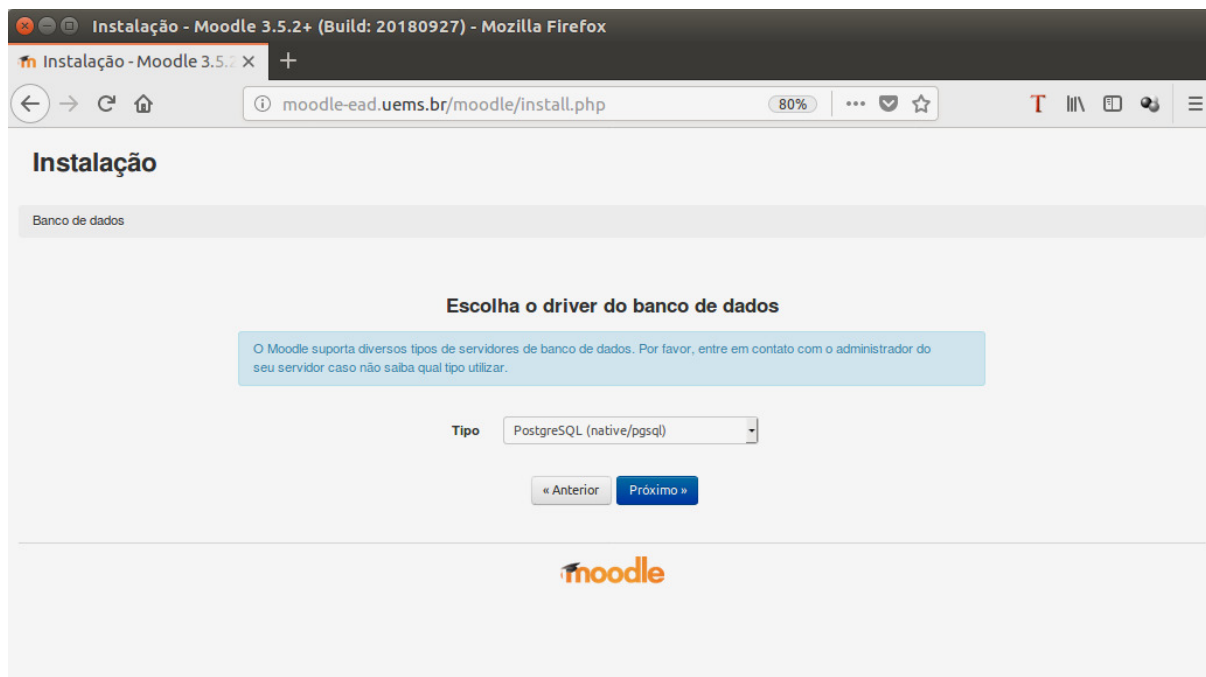
A Figura 33 mostra a configuração do diretório de dados para o armazenamento dos dados.

Figura 33 – Instalação Moodle passo 2



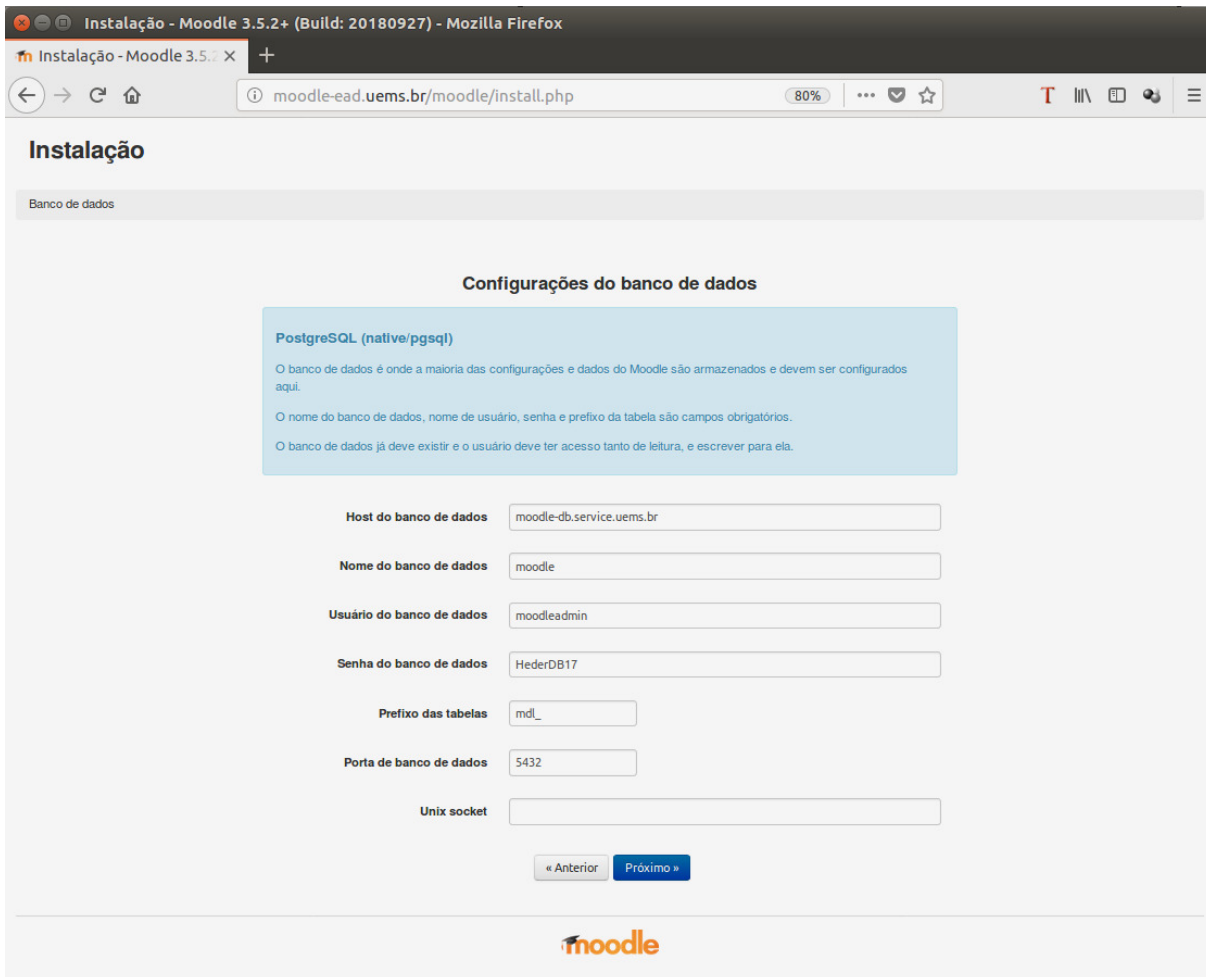
A Figura 34 mostra a configuração do driver de banco de dados utilizado, e o banco de dados utilizado foi do postgresql.

Figura 34 – Instalação Moodle passo 3



A figura 35 mostra algumas configurações que foram feitas no Moodle, como por exemplo prefixo mdl\_ que é um prefixo padrão das tabelas do banco de dados do Moodle e a porta 5432 que é a porta padrão do postgresql.

Figura 35 – Instalação Moodle passo 4



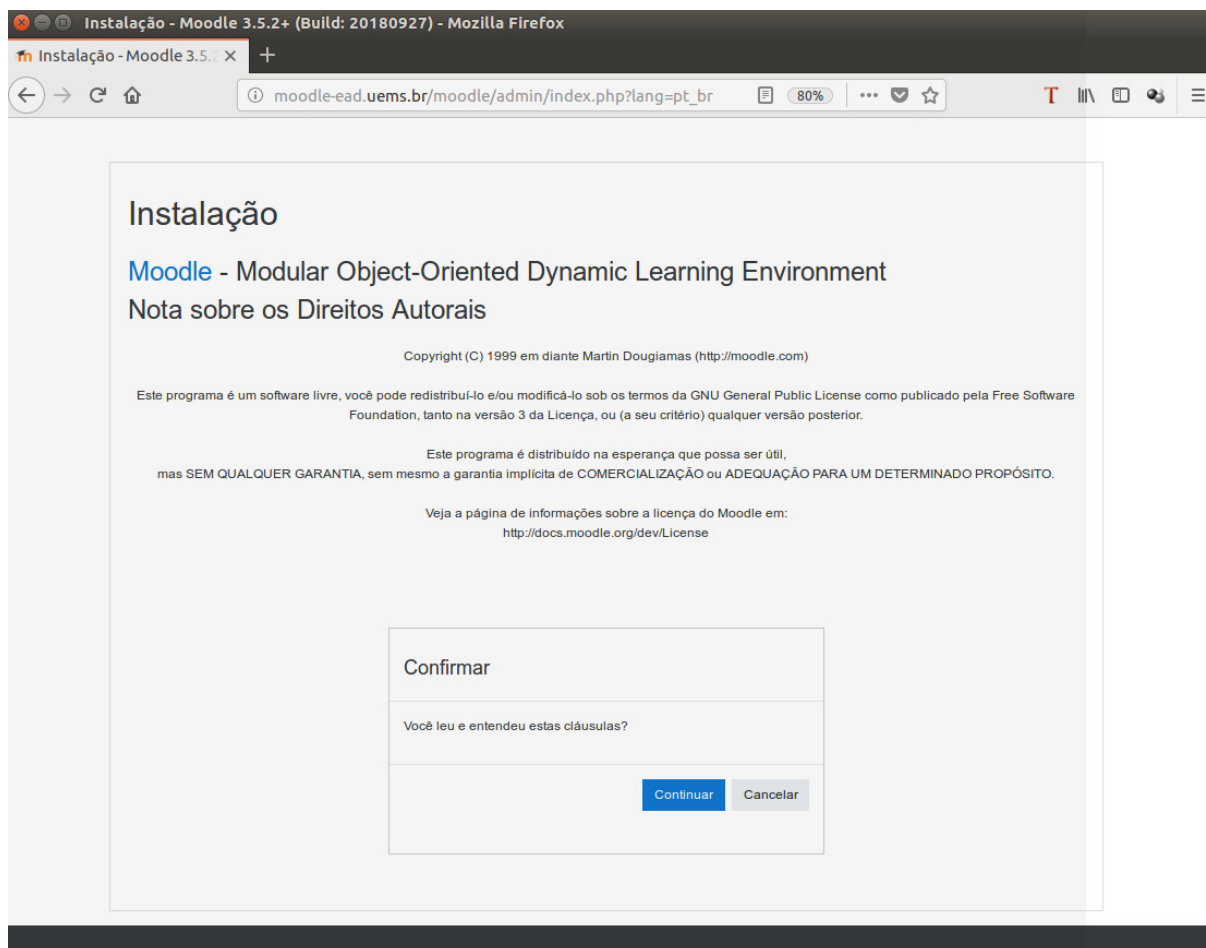
The screenshot shows the Moodle installation configuration page for PostgreSQL. The browser title is "Instalação - Moodle 3.5.2+ (Build: 20180927) - Mozilla Firefox". The URL is "moodle-ead.uems.br/moodle/install.php". The page is titled "Instalação" and shows the "Configurações do banco de dados" section. A blue box contains instructions for PostgreSQL configuration. Below this, there are several input fields for database configuration:

Field	Value
Host do banco de dados	moodle-db.service.uems.br
Nome do banco de dados	moodle
Usuário do banco de dados	moodleadmin
Senha do banco de dados	HederDB17
Prefixo das tabelas	mdl_
Porta de banco de dados	5432
Unix socket	

At the bottom of the form, there are two buttons: "« Anterior" and "Próximo »". The Moodle logo is visible at the bottom of the page.

A figura 36 mostra as cláusulas do termo de aceite para finalizar a instalação do Moodle.

Figura 36 – Instalação Moodle passo 5



A última etapa exibida na figura 37 mostra um pacote de extensões que foram instaladas para um perfeito funcionamento.

Figura 37 – Instalação Moodle passo 6

Instalação - Moodle 3.5.2+ (Build: 20180927)

Moodle 3.5.2+ (Build: 20180927)

Para obter informações sobre esta versão de Moodle, por favor visite a página [Notas sobre a Versão](#)

### Verificações do servidor

Nome	Informação	Relatório	Plugin	Status
unicode		deve ser instalado e habilitado		OK
database	postgres (9.6.10)	a versão 9.3 é necessária e você está usando a versão 9.6.10		OK
php		a versão 7.0.0 é necessária e você está usando a versão 7.0.30.0.9.1		OK
pcreunicode		recomendado como instalado/habilitado		OK
php_extension	iconv	deve ser instalado e habilitado		OK
php_extension	mbstring	recomendado como instalado/habilitado		OK
php_extension	curl	deve ser instalado e habilitado		OK
php_extension	openssl	deve ser instalado e habilitado		OK
php_extension	tokenizer	recomendado como instalado/habilitado		OK
php_extension	xmlrpc	recomendado como instalado/habilitado		OK
php_extension	soap	recomendado como instalado/habilitado		OK
php_extension	ctype	deve ser instalado e habilitado		OK
php_extension	zip	deve ser instalado e habilitado		OK
php_extension	zlib	deve ser instalado e habilitado		OK
php_extension	gd	deve ser instalado e habilitado		OK
php_extension	simplexml	deve ser instalado e habilitado		OK
php_extension	spl	deve ser instalado e habilitado		OK
php_extension	pcre	deve ser instalado e habilitado		OK
php_extension	dom	deve ser instalado e habilitado		OK
php_extension	xml	deve ser instalado e habilitado		OK
php_extension	xmlreader	deve ser instalado e habilitado		OK
php_extension	intl	deve ser instalado e habilitado		OK
php_extension	json	deve ser instalado e habilitado		OK
php_extension	hash	deve ser instalado e habilitado		OK
php_extension	fileinfo	deve ser instalado e habilitado		OK
php_setting	memory_limit	configuração recomendada detectada		OK
php_setting	file_uploads	configuração recomendada detectada		OK
php_setting	opcache.enable	configuração recomendada detectada		OK

### Outras verificações

Informação	Relatório	Plugin	Status
site not https	Se esse teste falhar, será indicio de um possível problema Foi detectado que seu site não está seguro porque não utiliza HTTPS. É altamente recomendável migrar seu site para HTTPS para aumentar a segurança e melhorar a integração com outros sistemas.		Verificar

O ambiente do seu servidor reúne todos os requisitos mínimos.

[Continuar](#)



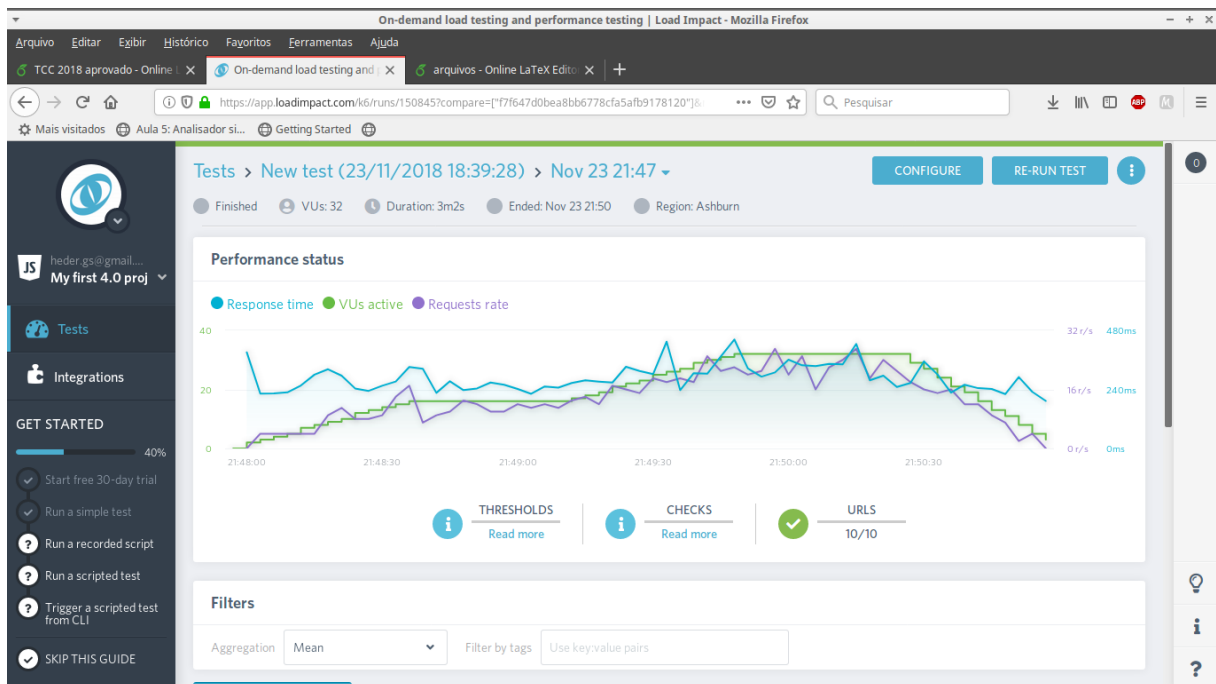
## 5 RESULTADOS

Os resultados obtidos com o presente estudo acarretaram na disponibilização de uma aplicação de aprendizado online, de baixo custo, para as instituições de ensino, neste caso, superior. A aplicação possui um alto grau de escalabilidade dos servidores de rede, pois permite o provisionamento sempre que o número de usuários aumentar ou quando algum servidor de aplicação, venha a falhar pode se substituir por outro sem com que o serviço fique fora do ar, também extrema segurança, pois onde a mesma está sendo executada, não há exposição à tentativa de acessos externos. O HAProxy reduziu a necessidades da utilização de IPs reais. Também foi aplicado o uso de sessão permanente para a aplicação, através do uso de cookies.

O balanceador de cargas HAProxy, associado com o uso da plataforma Moodle, apresentou um ótimo resultado de funcionamento. Os resultados gerados podem ser visualizados nas próximas imagens.

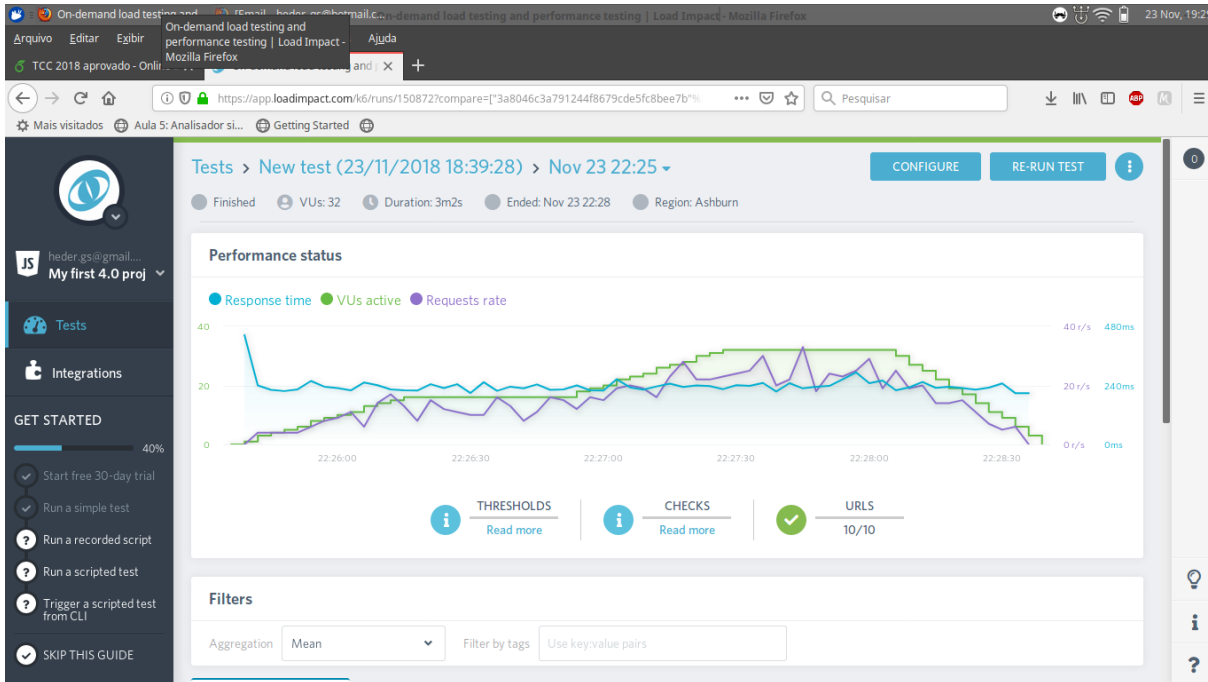
A figura 38 mostra o teste de carga que foi realizado no site <https://app.loadimpact.com>, sem o balanceador de carga HAProxy, para saber como o site se comporta com um grande número de acessos simultâneos em um determinado espaço de tempo. A linha azul significa o tempo de resposta a linha verde os números de usuários virtuais ativos e a linha roxa significa taxa de requisição. Uma requisição é qualquer solicitação dos usuários no site.

Figura 38 – Teste de Carga sem o Balanceamento de cargas



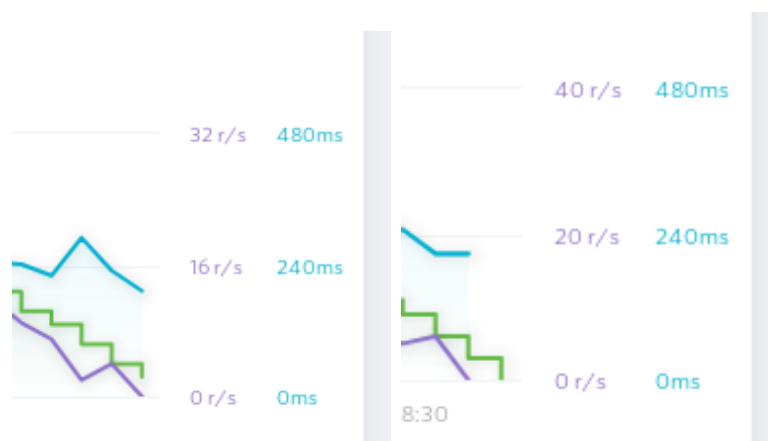
A figura 39 mostra o teste de carga que foi realizado no site <https://app.loadimpact.com>, com o balanceador de carga HAProxy, para saber como o site se comporta com um grande número de acessos simultâneos em um determinado espaço de tempo.

Figura 39 – Teste de Carga com o Balanceamento de cargas



A figura 40 mostram os resultados ampliados do teste de carga com e sem o balanceamento. O resultado a esquerda é do teste de carga sem o balanceamento de carga, alcançando 16 requisições por segundo o site leva 240 milissegundos, já com 32 requisições por segundo levam 480 milissegundo. O resultado a direita é do teste de carga com o balanceamento de carga, alcançando Com 20 requisições por segundo o site leva 240 milissegundos, já com 40 requisições por segundo levam 480 milissegundo.

Figura 40 – Imagens ampliadas dos resultados dos testes de carga



A figura 41 mostra o Proxmox que foi o ambiente de virtualização utilizado para a criação dos servidores utilizados nesse trabalho.

Figura 41 – VM PROXMOX

The screenshot shows the Proxmox VE web interface. The main content area displays a table of resources for the 'cicllope' cluster:

Type	Description	Disk usage...	Memory us...	CPU usage	Uptime
node	cicllope	2.0 %	39.0 %	0.7% of 8C...	9 days 06:44...
qemu	100 (nsslave3)	-	-	-	-
qemu	101 (loadbalance)	-	78.3 %	0.5% of 1C...	9 days 06:43...
qemu	102 (moodle-nfs)	-	71.6 %	0.7% of 1C...	9 days 06:43...
qemu	103 (moodle-db)	-	65.7 %	0.7% of 1C...	9 days 06:43...
qemu	104 (moodle-ap1)	-	78.2 %	0.7% of 1C...	9 days 06:43...
qemu	105 (moodle-ap2)	-	80.3 %	0.6% of 1C...	9 days 06:43...
storage	local (cicllope)	2.0 %	-	-	-
storage	local-lvm (cicllope)	5.2 %	-	-	-
storage	local-lvm2 (cicllope)	0.9 %	-	-	-

Below the resource list, there is a 'Tasks' section with a 'Cluster log' tab. The log shows several successful authentication events for the user 'root@pam' and one error event for the task 'UPID:cicllope:000039F4:049E86EE:5BC83ECE:aptupdate:root@pam: command 'apt-g...'.

A figura 42 mostra a página do HAProxy que exibe as informações de *Backend* e *Frontend*

Figura 42 – Loadbalance HAProxy

The screenshot shows the HAProxy Statistics Report for pid 1122. The page title is 'HAProxy version 1.7.5-2, released 2017/05/17' and the subtitle is 'Statistics Report for pid 1122'.

**General process information:**

- pid = 1122 (process #1, nbproc = 1)
- uptime = 98.23630m32s
- system limits: memmax = unlimited; ulimit-n = 4033
- maxsock = 4033; maxconn = 2000; maxpipes = 0
- current conns = 1; current pipes = 0/0; conn rate = 0/sec
- Running tasks: 1/7; idle = 100 %

**Legend:**

- active UP, going down
- active DOWN, going up
- active or backup DOWN
- active or backup DOWN for maintenance (MAINT)
- active or backup SOFT STOPPED for maintenance
- Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.
- backup UP, going down
- backup DOWN, going up
- [ ] not checked

**Display option:** Scope: [ ]

**External resources:**

- Primary site
- Updates (v1.7)
- Online manual
- Hide 'DOWN' servers
- Refresh now
- CSV export

**HA\_entrada**

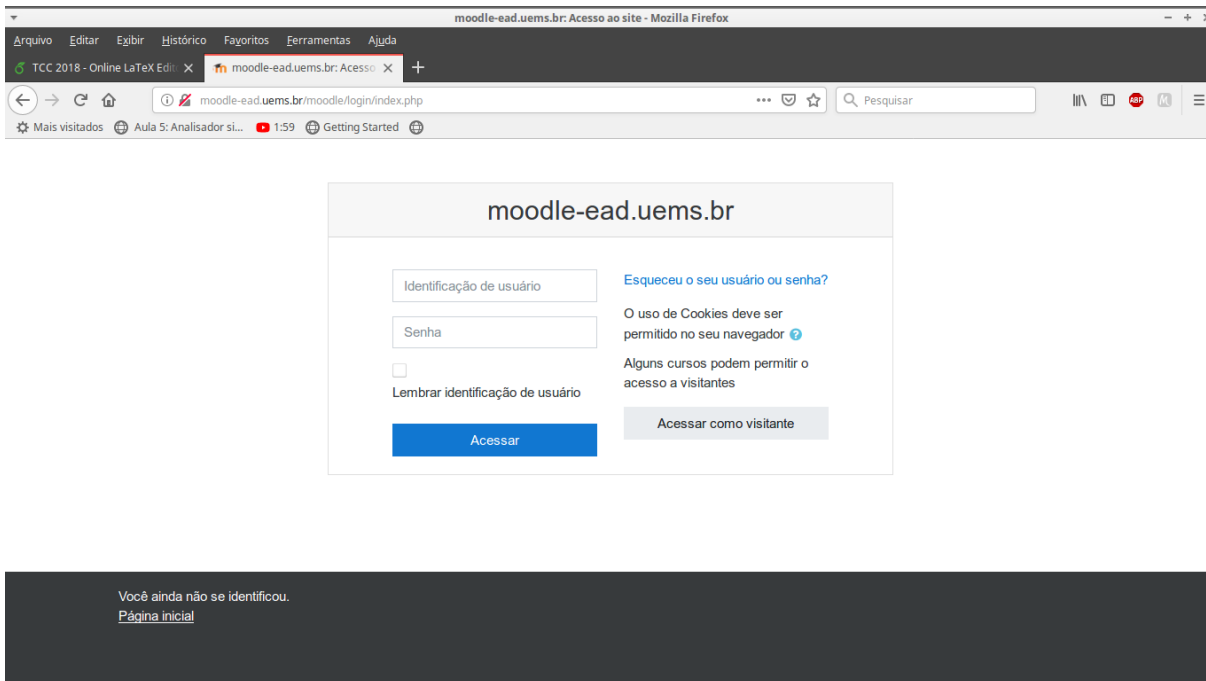
	Queue			Session rate			Sessions				Bytes			Denied		Errors		Warnings		Status	Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Conn	Resp	Ret		Redis	LastChk	Wght	Act	Bck	Chk	Dwn	Downtm	Thrtle
Frontend	0	0	-	0	18	-	0	6	-	398	10	9h9m	280 860	5 805 095	0	0	0	0	0	0	19h29m UP	L4OK in 1ms	1	Y	-	13	5	7d17m	-
	0	0	-	0	12	-	0	5	-	200	7	9h9m	130 159	2 801 691	0	0	1	0	3	0	7s UP	L4OK in 0ms	1	Y	-	7	3	9d2h	-
Backend	0	0	-	0	18	-	0	6	200	595	17	9h9m	411 019	8 606 786	0	0	1	0	3	0	21h59m UP		2	2	0	4	7d17m	-	

**moodle\_saida**

	Queue			Session rate			Sessions				Bytes			Denied		Errors		Warnings		Status	Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Conn	Resp	Ret		Redis	LastChk	Wght	Act	Bck	Chk	Dwn	Downtm	Thrtle
moodle-ap1	0	0	-	0	18	-	0	6	-	398	10	9h9m	280 860	5 805 095	0	0	0	0	0	0	19h29m UP	L4OK in 1ms	1	Y	-	13	5	7d17m	-
moodle-ap2	0	0	-	0	12	-	0	5	-	200	7	9h9m	130 159	2 801 691	0	0	1	0	3	0	7s UP	L4OK in 0ms	1	Y	-	7	3	9d2h	-
Backend	0	0	-	0	18	-	0	6	200	595	17	9h9m	411 019	8 606 786	0	0	1	0	3	0	21h59m UP		2	2	0	4	7d17m	-	

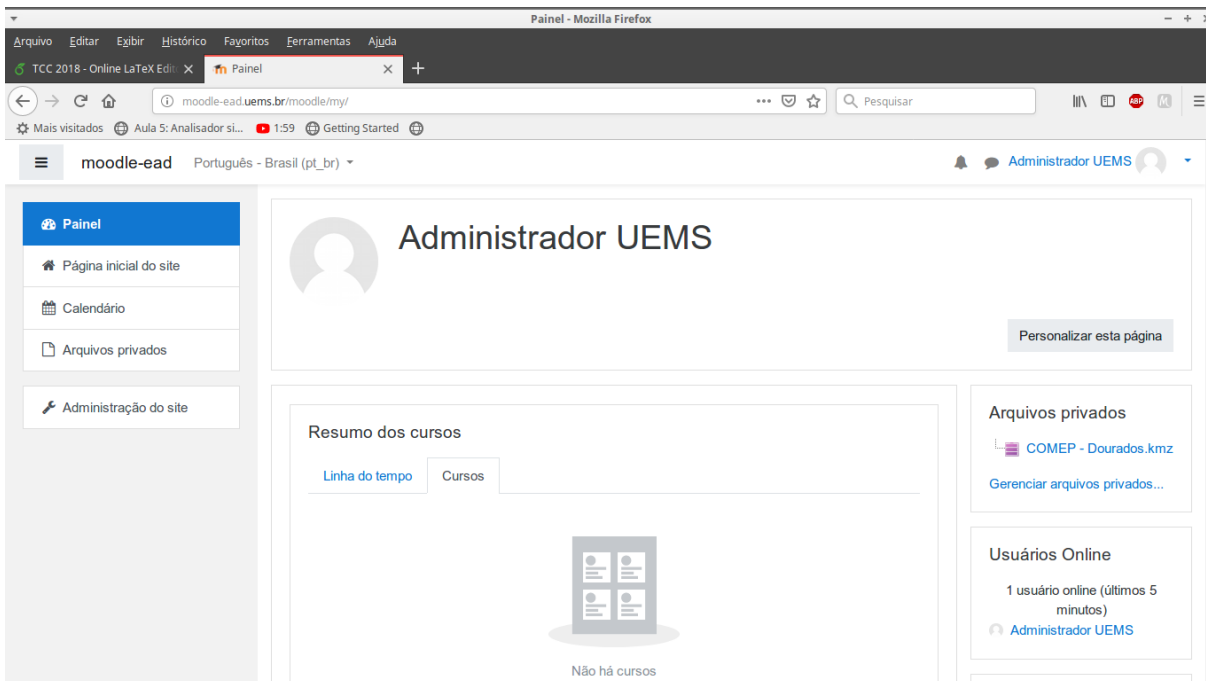
A figura 43 mostra a página inicial de *login* do Moodle.

Figura 43 – *Login* do usuário na plataforma Moodle



A figura 44 mostra a página depois do *login* no Moodle.

Figura 44 – Plataforma Moodle



## 6 CONCLUSÃO

O presente estudo objetivou a disponibilização de uma aplicação de alto grau de escalabilidade dos servidores de baixo custo, para as instituições de ensino superior. Possui extrema segurança, pois onde a mesma está sendo executada, não há exposição à tentativa de acessos externos.

Observou-se que o uso do balanceador de cargas HAProxy apresentou 20% a mais de desempenho, o que pode não parecer muito, mas se colocarmos em um um número com um milhão a diferença é de 200 mil. Alcançou um bom resultado de funcionamento associado com o uso da plataforma Moodle, uma vez que permite o provisionamento de mais instâncias da aplicação, conforme a demanda da mesma aumenta. Assim, os resultados gerados e visualizados na aplicação Moodle desenvolvida, mostram-se como uma alternativa de baixo custo para a implementação de uma plataforma de aprendizado online.

Durante o desenvolvimento deste trabalho inúmeras descobertas foram feitas, entretanto, várias etapas tiveram que de ser superadas. Dentre elas, destacam-se a busca da compreensão do funcionamento de um balanceador de cargas, como ele se comporta, os protocolos utilizados, bem como, de que forma ele pode ser implementado para a plataforma de ensino online.

A conclusão do presente trabalho foi possível, virtude do estágio realizado na área de atuação deste projeto. Outro fator determinante para que a pesquisa pudesse prosseguir de forma mais ágil, foi a facilidade de integração das tecnologias utilizadas, sendo elas NFS, Proxmox, HAProxy e Linux Debian.

Resta como sugestão para trabalhos futuros, o estudo de um balanceador de carga diferente como, por exemplo, o Nginx, podendo ser elaborado um comparativo entre os balanceadores de carga, a fim de verificar quem melhor realiza a tarefa.

Por fim, as descobertas e novos aprendizados resultantes da supramencionada pesquisa contribuíram para um crescimento acadêmico, pessoal, teórico e profissional, acrescentando valor intelectual e experiência.

# REFERÊNCIAS

ADDRESS what is my ip. *Hide Your Real IP Address Behind a Proxy*. 2018. Disponível em: <<https://whatismyipaddress.com/proxy-server>>. Acesso em: 12-05-2018.

APACHE. *About Apache*. 2018. Disponível em: <[https://httpd.apache.org/ABOUT\\_APACHE.html](https://httpd.apache.org/ABOUT_APACHE.html)>. Acesso em: 11-06-2018.

DEBIAN. *WHAT is Debian?* 2018. Disponível em: <<https://www.debian.org/intro/about.en.html>>. Acesso em: 19-07-2018.

EDITION, H. C. *The Reliable, High Performance TCP/HTTP Load Balancer*. 2018. Disponível em: <<http://www.haproxy.org/>>. Acesso em: 03-09-2018.

GMBH, P. S. S. *Open-Source Virtualization Platform*. 2018. Disponível em: <<https://www.proxmox.com/en/proxmox-ve>>. Acesso em: 28-07-2018.

JONES, M. T. *Network file systems and Linux*. 2018. Disponível em: <<https://www.ibm.com/developerworks/library/l-network-filestems/index.html>>. Acesso em: 24-06-2018.

LUCAS, A. *Por que fazer teste de carga no seu website*. 2018. Disponível em: <<https://www.igluonline.com/por-que-fazer-teste-de-carga-no-seu-website/>>. Acesso em: 21-11-2018.

MOODLE. *Moodle*. 2018. Disponível em: <[https://docs.moodle.org/35/en/About\\_Moodle](https://docs.moodle.org/35/en/About_Moodle)>. Acesso em: 11-04-2018.

POSTGRESQL. *What is PostgreSQL?* 2018. Disponível em: <<https://www.postgresql.org/about/>>. Acesso em: 08-07-2018.

SHIELD, H. *Servidor Proxy*. 2018. Disponível em: <<https://www.hotspotshield.com/pt/resources/o-que-e-servidor-proxy>>. Acesso em: 06-05-2018.

TELIUM. *Veja como o load balance pode ajudar na alta disponibilidade de rede do negócio*. 2018. Disponível em: <<https://blog.telium.com.br/veja-como-o-load-balance-pode-ajudar-na-alta-disponibilidade-de-rede-do-negocio/>>. Acesso em: 10-08-2018.