rede e-Tec Brasil

Implantação e Administração de Serviços *web*

Dener Didoné

Felipe José Chaulet

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PERNAMBUCO

Recife - PE
2016

Presidência da República Federativa do Brasil

Ministério da Educação

Secretaria de Educação Profissional e Tecnológica

Este caderno foi elaborado em parceria entre o Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco e a Universidade Federal de Santa Maria para a Rede e-Tec Brasil.

Equipe de Elaboração
Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco – IFPE

Reitora
Anália Keila Rodrigues Ribeiro/IFPE

Diretor Geral
Fernanda Maria Dornellas Câmara/IFPE

Coordenação Institucional
Fabiola Nascimento dos Santos Queiros/IFPE
Fábia Gonçalves de Melo Torres/IFPE

Coordenação de Curso
Leandro Marques Queiros/IFPE

Professor-autor
Dener Didoné/IFPE
Felipe José Chaulet/ IFPE

Equipe de Acompanhamento e Validação
Colégio Técnico Industrial de Santa Maria – CTISM

Coordenação Institucional
Paulo Roberto Colusso/CTISM

Coordenação de Design
Erika Goellner/CTISM

Revisão Pedagógica
Elisiane Bortoluzzi Scrimini/CTISM
Jaqueline Müller/CTISM

Revisão Textual
Carlos Frederico Ruviaro/CTISM

Revisão Técnica
Rogério Turchetti/CTISM

Ilustração
Marcel Santos Jacques/CTISM
Ricardo Antunes Machado/CTISM

Diagramação
Emanuelle Shaiane da Rosa/CTISM
Tagiane Mai/CTISM

D557i DIDONÉ, Dener.
Implantação e Administração de Serviços web / Dener Didoné; Felipe José Chaulet. – Recife: IFPE, 2016.
87 p. : il.

Inclui bibliografia
Rede e-Tec Brasil

ISBN: 978-85-9450-013-7

1. Serviços da web. 2. Arquitetura de computadores. 3. Internet. I. Título.

CDD: 004.21

Apresentação e-Tec Brasil

Prezado estudante,
Bem-vindo a Rede e-Tec Brasil!

Você faz parte de uma rede nacional de ensino, que por sua vez constitui uma das ações do Pronatec – Programa Nacional de Acesso ao Ensino Técnico e Emprego. O Pronatec, instituído pela Lei nº 12.513/2011, tem como objetivo principal expandir, interiorizar e democratizar a oferta de cursos de Educação Profissional e Tecnológica (EPT) para a população brasileira propiciando caminho de o acesso mais rápido ao emprego.

É neste âmbito que as ações da Rede e-Tec Brasil promovem a parceria entre a Secretaria de Educação Profissional e Tecnológica (SETEC) e as instâncias promotoras de ensino técnico como os Institutos Federais, as Secretarias de Educação dos Estados, as Universidades, as Escolas e Colégios Tecnológicos e o Sistema S.

A educação a distância no nosso país, de dimensões continentais e grande diversidade regional e cultural, longe de distanciar, aproxima as pessoas ao garantir acesso à educação de qualidade, e promover o fortalecimento da formação de jovens moradores de regiões distantes, geograficamente ou economicamente, dos grandes centros.

A Rede e-Tec Brasil leva diversos cursos técnicos a todas as regiões do país, incentivando os estudantes a concluir o ensino médio e realizar uma formação e atualização contínuas. Os cursos são ofertados pelas instituições de educação profissional e o atendimento ao estudante é realizado tanto nas sedes das instituições quanto em suas unidades remotas, os polos.

Os parceiros da Rede e-Tec Brasil acreditam em uma educação profissional qualificada – integradora do ensino médio e educação técnica, – é capaz de promover o cidadão com capacidades para produzir, mas também com autonomia diante das diferentes dimensões da realidade: cultural, social, familiar, esportiva, política e ética.

Nós acreditamos em você!
Desejamos sucesso na sua formação profissional!

Ministério da Educação
Julho de 2016

Nosso contato
etecbrasil@mec.gov.br

Indicação de ícones

Os ícones são elementos gráficos utilizados para ampliar as formas de linguagem e facilitar a organização e a leitura hipertextual.

Atenção: indica pontos de maior relevância no texto.

Saiba mais: oferece novas informações que enriquecem o assunto ou "curiosidades" e notícias recentes relacionadas ao tema estudado.

Glossário: indica a definição de um termo, palavra ou expressão utilizada no texto.

Mídias integradas: sempre que se desejar que os estudantes desenvolvam atividades empregando diferentes mídias: vídeos, filmes, jornais, ambiente AVEA e outras.

Atividades de aprendizagem: apresenta atividades em diferentes níveis de aprendizagem para que o estudante possa realizá-las e conferir o seu domínio do tema estudado.

Sumário

Palavra do professor-autor

Caros(as) estudantes,

O material aqui apresentado foi elaborado com o objetivo de atender às necessidades do componente curricular Implantação e Administração de Serviços *web* do curso de Licenciatura em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco. Também é objetivo desse material, proporcionar uma experiência teórica e prática sobre os assuntos aqui relatados, de forma que o conteúdo seja absorvido mais facilmente e assim, enfatizando a aprendizagem.

Os conteúdos teóricos aqui presentes, tem como objetivo, dar a base necessária para o entendimento do assunto abordado, fazendo com que os(as) estudantes consigam compreender o funcionamento dos serviços apresentados nesse material. Para mostrar o funcionamento desses serviços, os conteúdos práticos apresentam uma estrutura de passo-a-passo, enfatizando os pontos mais importantes e trazendo dicas para melhor entendimento dos assuntos. É importante que os(as) estudantes tentem replicar o conteúdo prático, pois através da replicação poderão ver princípios teóricos em funcionamento, além de poder despertar o interesse de aumentar o aprendizado, visto que terão um serviço *web* devidamente implantado e em funcionamento.

Este material está dividido em nove aulas para auxiliar no entendimento de todo o conteúdo. As três primeiras aulas contêm assuntos teóricos, como: serviços cliente/servidor, arquitetura cliente/servidor e ambientes de servidor. As cinco aulas seguintes, tem teor prático, abordando: servidores *web* servidor FTP, servidor DNS, servidor de correio eletrônico, servidor de compartilhamento de arquivos e servidor de impressão. Finalizando, a aula nove mostra ferramentas de administração de servidores e serviços, que mostra maneiras de gerenciar os serviços presentes em um servidor.

Finalizamos essa apresentação desejando a vocês estudantes, o melhor aproveitamento possível do material aqui apresentado, pois o mesmo foi desenvolvido para lhes despertar interesse e curiosidade para continuar na jornada de estudos do curso que escolheram.

Dener Didoné
Felipe José Chaulet

Apresentação da disciplina

Prezados(as) estudantes!

Ao início de mais um componente curricular, surge a oportunidade de conhecer melhor o curso que escolheram! O trabalho da disciplina de Implantação e Administração de Serviços *web* é mostrar-lhes as possibilidades existentes no mercado atual, para que assim tenham ciência do enorme leque de ferramentas disponíveis para se inserir no mundo do trabalho. Trazemos nesta disciplina uma metodologia teórica e prática, objetivando salientar os conceitos para que as práticas complementem o conhecimento teórico.

Agora é o momento de adicionar os conceitos aqui apresentados aos conhecimentos adquiridos durante o curso. Logicamente, alguns conceitos podem parecer familiares, porém não deixam de ser importantes, pois trazem informações específicas acerca do funcionamento de serviços e ferramentas que, somadas às atividades práticas, fornecem conhecimento completo dos temas que serão abordados.

A disciplina contém uma estrutura preparada especialmente para que vocês possam colocar em prática os conceitos aprendidos e assim, terem mais clareza sobre o assunto. Portanto, é aconselhável que as atividades práticas sejam realizadas, pois é através delas que poderão ver os conceitos, serviços e ferramentas em funcionamento.

Finalizando, acreditamos que, com as teorias e práticas aqui apresentadas, vocês terão uma visão clara sobre a sua importância, não só para esta disciplina, mas para o aprendizado como um todo. Que isso lhes sirva de exemplo para o futuro como profissionais da área de Informática.

Projeto instrucional

Disciplina: Implantação e Administração de Serviços *web* (carga horária: 75h).

Ementa: Serviços cliente/servidor. Arquitetura cliente/servidor. Ambientes de servidor. Instalação, configuração e manutenção dos serviços: *web*, DNS, FTP. Correio eletrônico. Compartilhamento de arquivos. Impressão. Ferramentas utilizadas na administração de serviços.

AULA	OBJETIVOS DE APRENDIZAGEM	MATERIAIS	CARGA HORÁRIA (horas)
1. Serviços cliente/servidor	Compreender os conceitos sobre os serviços que as aplicações cliente/servidor oferecem.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: *links*, exercícios.	06
2. Arquitetura cliente/servidor	Compreender o funcionamento da arquitetura de sistemas cliente/servidor.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: *links*, exercícios.	06
3. Ambientes de servidor	Conhecer os ambientes existentes em servidores.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: *links*, exercícios.	06
4. Servidores *web*	Conhecer o funcionamento de servidores *web* e servidores DNS, instalá-los e configurá-los em um sistema operacional GNU/Linux.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: *links*, exercícios.	12
5. Servidor FTP	Instalar, configurar e gerenciar um servidor FTP (File Transfer Protocol), bem como conhecer o seu funcionamento.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: *links*, exercícios.	10
6. Servidor DNS	Instalar, configurar e gerenciar um servidor de DNS e dominar conhecimentos sobre o seu funcionamento e finalidade.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: *links*, exercícios.	08
7. Servidor de correio eletrônico	Instalar e configurar um servidor de correio eletrônico e dominar conhecimentos sobre seu funcionamento.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: *links*, exercícios.	08

AULA	OBJETIVOS DE APRENDIZAGEM	MATERIAIS	CARGA HORÁRIA (horas)
8. Servidor de compartilhamento de arquivos e servidor de impressão	Instalar e configurar um servidor de compartilhamento de arquivos Samba, da mesma forma, conhecer sobre seu funcionamento e finalidade.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: *links*, exercícios.	11
9. Ferramentas de administração de servidores e serviços	Instalar, configurar e utilizar as ferramentas mais utilizadas para administração local e remota de servidores *web*, bem como manipular os serviços operacionais.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: *links*, exercícios.	08

Aula 1 – Serviços cliente/servidor

Objetivos

Compreender os conceitos sobre os serviços que as aplicações cliente/servidor oferecem.

1.1 Considerações iniciais

Os sistemas cliente/servidor têm como objetivo oferecer, através de um computador robusto, uma série de serviços que ficam acessíveis aos clientes das mais diversas formas possíveis.

O funcionamento de um sistema cliente/servidor consiste em uma situação onde um cliente envia uma solicitação para um servidor que a processa e envia a resposta para o cliente. Com base nessa resposta, o cliente pode enviar novas requisições.

O servidor, como se mencionou anteriormente, normalmente consiste em um computador com grande poder de processamento, grande poder de armazenamento e capacidade de transmitir informações em alta velocidade, através da rede. É comum um mesmo servidor oferecer vários serviços, visto que o computador tem essas condições. Assim, é eliminada a necessidade de aquisição de vários servidores. Dessa forma, o serviço é executado no servidor, em uma porta predeterminada e sempre que um cliente realizar uma requisição naquela porta, o servidor responde de forma adequada. Em outras palavras, o servidor, ao iniciar o serviço, fica aguardando requisições de usuários.

O cliente é o responsável por dar início à troca de informações, pois é ele que envia uma requisição que será processada pelo servidor. Essa requisição pode ser enviada das mais diversas formas possíveis. É muito comum que exista um aplicativo específico para cada caso, conforme será mostrado posteriormente.

Também é muito comum, que sistemas cliente/servidor funcionem de forma que a comunicação entre o cliente e o servidor seja realizada através de rede, pois em muitos casos o cliente e o servidor não estão no mesmo computador. A Figura 1.1 mostra um básico exemplo de um serviço cliente/servidor funcionando através da internet.

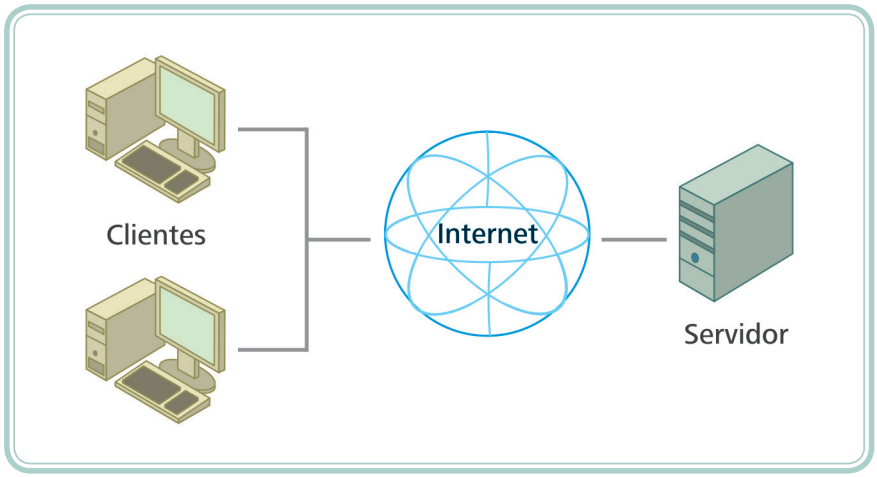


Figura 1.1: Comunicação cliente/servidor
Fonte: CTISM

1.2 Tipos de serviços

O sistema cliente/servidor mais utilizado provavelmente é o serviço *web* que consiste no acesso de páginas de internet nos navegadores.

O servidor *web* possui um serviço sendo executado, no qual a porta 80, ou 8080 (nas configurações padrão) fica sendo monitorada e quando alguma informação é enviada para o servidor através dessa porta, ele a interpreta conforme a configuração do serviço. Nesse caso, o cliente, que é um usuário qualquer acessando uma página *web* de seu computador, envia uma requisição, que é o endereço da página que ele quer carregar. O servidor, por sua vez, processa essa informação, retornando a página *web* para o cliente. O cliente pode então interagir com essa página através de formulários, enviando uma nova requisição para o servidor, iniciando o processo todo novamente. Neste caso é fácil perceber que o cliente e o servidor não estão no mesmo computador. Isso significa que a comunicação entre eles é realizada através de rede. Mas, em uma requisição como essa, o próprio servidor pode fazer o papel de cliente, acessando outro serviço que está sendo executado dentro do mesmo servidor. Por exemplo, quando o usuário utiliza um serviço de autenticação, ele envia a requisição de uma página de internet que será acessível apenas se as informações que ele enviou para o servidor estiverem cadastradas em um banco de dados no servidor. Nesse caso, o cliente envia a requisição para o servidor que precisa consultar as informações do usuário no banco de dados. Então uma requisição é enviada do servidor *web* para o servidor do banco de dados. O servidor do banco de dados retorna para o servidor *web* as informações solicitadas. O servidor *web* utiliza as informações fornecidas pelo servidor do banco de dados para montar a página *web* solicitada pelo usuário e a envia para o mesmo.

Vários clientes podem acessar o mesmo serviço no servidor simultaneamente.

É possível perceber que na comunicação entre o servidor *web* e o servidor do banco de dados, existe uma grande possibilidade de os dois serviços estarem sendo executados no mesmo computador e, nesse caso, a comunicação deles é apenas entre os serviços dentro do servidor, não dependendo da rede.

É visível que nesse ambiente de cliente/servidor, o cliente normalmente utiliza um aplicativo com o qual pode interagir das mais diversas formas para acessar os serviços do servidor. No caso de páginas *web*, o usuário utiliza o navegador. Outro serviço cliente/servidor mencionado anteriormente é o servidor de banco de dados. No exemplo citado o servidor de banco de dados é acessado pelo servidor *web*, sem ser através de uma interface de usuário, ou seja, é um serviço se comunicando com outro serviço. E, no caso, o usuário não se comunica diretamente com o servidor do banco de dados.

Pode existir, entretanto, um aplicativo qualquer que acesse diretamente o banco de dados. Como exemplo, é possível utilizar um aplicativo de automação comercial. Nesse caso, existe um servidor do banco de dados, no qual estão armazenadas todas as informações da empresa como cadastro dos clientes, funcionários, produtos, vendas dentre outros. O aplicativo, por sua vez, foi desenvolvido especificamente para aquele fim. Quando um usuário o utiliza, está se comunicando diretamente com o servidor do banco de dados através da rede. A Figura 1.2 mostra a estrutura de um aplicativo cliente/servidor que acessa diretamente um banco de dados.

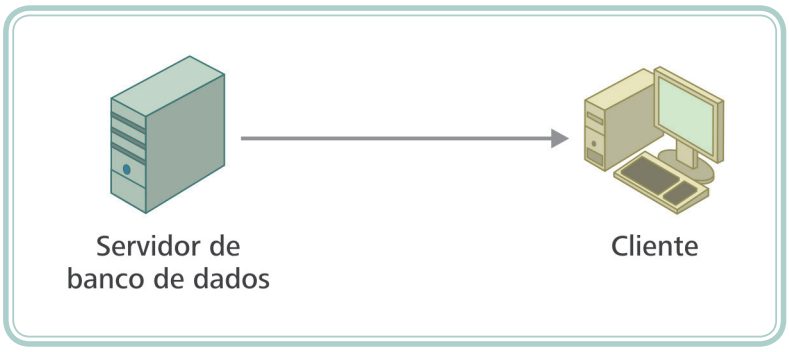


Figura 1.2: Cliente recebendo informações de um servidor de banco de dados
Fonte: CTISM

Outro serviço cliente/servidor comumente utilizado está presente em aplicativos de IM (*Instant Messenger*), que nada mais são que os aplicativos de mensagens instantâneas, como GTalk, Skype, Windows Live Messenger, dentre outros. Nesse caso, existe também um aplicativo específico que é utilizado pelo usuário. Esse aplicativo, por sua vez, se conecta ao servidor e oferece para o usuário uma série de funcionalidades.

O servidor deve ser capaz de processar as requisições enviadas pelos usuários e, para isso, ele utiliza várias regras presentes em protocolos de comunicação. Nos casos de mensagens instantâneas via texto, o protocolo utilizado para a comunicação é o TCP (*Transmission Control Protocol*). Esse protocolo define que em uma comunicação entre dois dispositivos, a comunicação deve ser aberta, existe a troca de informações e, posteriormente, essa comunicação é fechada.

É possível fazer uma analogia com a comunicação através de um rádio amador. Uma comunicação através de rádio amador se inicia quando um usuário chama verbalmente outro, mas a comunicação entre eles só acontece quando o usuário que foi chamado, responder. Quando o usuário chamado responder, eles iniciam a comunicação entre si, sendo que um usuário fala e quando quer indicar que terminou de falar, ou seja, terminou de enviar informações, utiliza a palavra "câmbio". O outro usuário, por sua vez, ao ouvir a palavra "câmbio" entende que todas as informações foram recebidas e começa a falar, a enviar suas informações. É comum também, o uso da palavra "entendido", que tem como objetivo confirmar se o outro usuário conseguiu compreender as informações enviadas. Para terminar a comunicação, utiliza-se a sentença "câmbio e desligo". Uma comunicação através de protocolo TCP funciona de forma semelhante. Um dispositivo inicia a comunicação enviando um pacote que tem como objetivo estabelecer uma comunicação entre os dois dispositivos. O outro dispositivo responde, e a troca de informações inicia. Cada vez que um dispositivo envia uma informação, o dispositivo que a recebeu, responde com uma confirmação de que a informação foi recebida, semelhante ao "entendido". Quando a troca de informações termina, os usuários trocam um pacote que indica o término da comunicação, semelhante ao "câmbio e desligo".

Esse processo é realizado para garantir que as informações não foram perdidas durante a comunicação e, caso um dispositivo enviar uma informação e não receber a confirmação de que ela foi recebida pelo destinatário, ele a envia novamente.

Em um serviço de troca de mensagens de texto, esse processo é utilizado. Quando um usuário envia uma mensagem, o destinatário retorna um pacote informando que essa mensagem foi recebida, garantindo a integridade das informações.

Vários aplicativos de mensagens instantâneas oferecem o serviço de chamadas de vídeo que consistem em uma comunicação de videoconferência. Esse

tipo de comunicação utiliza o protocolo UDP (*User Datagram Protocol*) que, diferentemente do protocolo TCP, não solicita a confirmação do recebimento das informações enviadas. Isso significa que um usuário fica enviando informações sem parar, independentemente de o destinatário recebê-las ou não. Isso acontece porque nesse tipo de comunicação, se houver a verificação das informações, a comunicação torna-se incompreensível. O mesmo acontece em chamadas de voz pela internet, o VOIP.

É muito comum, na elaboração de programas, a utilização de sockets para realizar a comunicação entre dois computadores. O socket é uma estrutura capaz de fazer com que um computador execute um serviço em uma determinada porta, operando como servidor, da mesma forma, proporciona que um cliente se comunique com essa porta. Assim é possível desenvolver um programa de computador do tipo cliente/servidor para as mais diversas necessidades.

Resumo

Essa aula apresentou os serviços cliente/servidor, os seus tipos e serviços utilizados atualmente. Para explicar o funcionamento de uma comunicação realizada através de um protocolo TCP, foi feita uma analogia à comunicação através de rádio amador.

Dentre os serviços cliente/servidor mais utilizados, está o servidor *web*. É através dele que páginas de internet são disponibilizadas para os usuários. É o serviço que controla a configuração de acesso aos recursos das páginas de internet, oferecendo assim, na maioria dos casos, sistemas escaláveis e de fácil acesso.

Atividades de aprendizagem

1. Cite três serviços cliente/servidor e dê exemplos.

2. Explique o que é um serviço cliente/servidor.

Aula 2 – Arquitetura cliente/servidor

Objetivos

Compreender o funcionamento da arquitetura de sistemas cliente/servidor.

2.1 Considerações iniciais

Conforme a Aula 1, os sistemas cliente/servidor normalmente são compostos por um servidor, no qual um ou mais serviços estão sendo executados. O servidor geralmente é um computador com grande capacidade de processamento, armazenamento e de transmissão de dados pela rede em alta velocidade. O outro componente de um sistema cliente/servidor é o cliente que vai acessar o serviço ou serviços disponibilizados pelo servidor. Claro que o acesso pode ser realizado por mais de um cliente simultaneamente.

O serviço oferecido pelo servidor opera em uma porta definida pela aplicação que o manipula e pode ser acessada das mais diversas formas. Por exemplo, uma aplicação cliente/servidor muito comum é o serviço *web*, no qual o servidor possui páginas *web* que são acessadas pelo cliente através do navegador. Existem casos onde o servidor faz também o papel de cliente, quando o mesmo redireciona uma consulta que recebeu de um cliente para outro servidor. Um exemplo disso é o servidor DNS. Quando o usuário digita o endereço de uma *website* em seu navegador, o servidor DNS pega esse endereço e o converte em um endereço IP para descobrir em qual computador na internet essa *website* está armazenada. Porém, se esse servidor DNS não possuir o endereço de IP do *site* requisitado, ele vai consultar outro servidor DNS, que possui mais informações que ele. Nessa situação, o servidor que recebeu a primeira requisição do *site*, faz o papel de cliente, quando envia a solicitação para outro servidor DNS.

O cliente de uma aplicação cliente/servidor tem, na grande maioria das vezes, uma interface customizada pela qual ele acessa os serviços oferecidos pelo servidor. Essa interface customizada pode ser desenvolvida especificamente para um propósito, como uma interface que acessa um banco de dados específico no servidor. Cada banco de dados possui uma estrutura diferente e mesmo que o banco de dados esteja no mesmo servidor, será necessária

uma interface ligeiramente diferente para acessar o banco de dados. Por causa desse contexto, é possível oferecer uma série de serviços personalizados, visto que a interface é desenvolvida para um fim específico. Existem casos ainda de interfaces genéricas. Estas têm como objetivo acessar um serviço disponível por um servidor, entretanto, como são genéricas, devem ser compatíveis com a estrutura do serviço oferecido pelo servidor. No exemplo do servidor *web*, a interface utilizada pelo usuário é o navegador. Atualmente, existem vários tipos de navegadores que têm o mesmo objetivo: acessar o serviço oferecido pelo servidor *web*. Os navegadores existentes hoje em dia são ligeiramente parecidos, pois eles têm que ser compatíveis com uma grande quantidade de protocolos utilizados pelas páginas de internet. Porém, eles diferem entre si porque uns oferecem algumas funcionalidades a mais, como extensões e *plugins* que têm como objetivo melhorar a sua usabilidade.

2.2 Arquitetura

A arquitetura cliente/servidor é dividida em quatro níveis diferentes, de acordo com sua complexidade e generalidade. A Figura 2.1 mostra a arquitetura cliente/servidor simples. Nesse modelo, existe a comunicação apenas entre um cliente e um servidor. O servidor fica aguardando o cliente iniciar a comunicação.

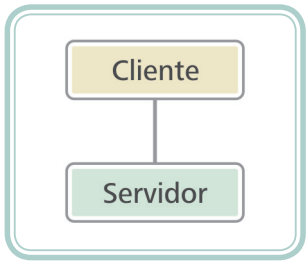


Figura 2.1: Arquitetura cliente/servidor simples
Fonte: CTISM, adaptado de http://www.unlu.edu.ar/~tyr/tyr/TYanterior/Fundamentos_da_%20Arquitetura_Cliente-Servidor.pdf

Já a Figura 2.2 mostra a arquitetura cliente/servidor em dois níveis, centralizada no servidor. Nesse modelo, vários clientes acessam o serviço ou serviços em apenas um servidor.

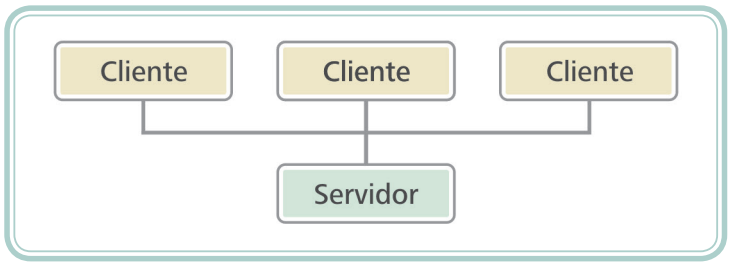


Figura 2.2: Arquitetura cliente/servidor em dois níveis – centralizada no servidor
Fonte: CTISM, adaptado de http://www.unlu.edu.ar/~tyr/tyr/TYR-anterior/Fundamentos_da_%20Arquitetura_Cliente-Servidor.pdf

A Figura 2.3 traz o modelo cliente/servidor em dois níveis centralizado no cliente. Nesse caso, pelo fato de o cliente não poder perceber a presença de outros clientes no sistema e ter acesso a vários serviços no servidor, ele pode ter a impressão de que existem vários servidores.

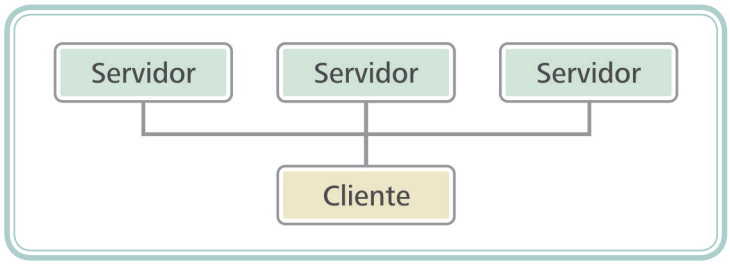


Figura 2.3: Arquitetura cliente/servidor em dois níveis – centralizada no cliente
Fonte: CTISM, adaptado de http://www.unlu.edu.ar/~tyr/tyr/TYanterior/Fundamentos_da_%20Arquitetura_Cliente-Servidor.pdf

Na realidade, o que acontece é que vários clientes se comunicam com vários servidores e vários servidores se comunicam com vários clientes simultaneamente, conforme ilustra a Figura 2.4.

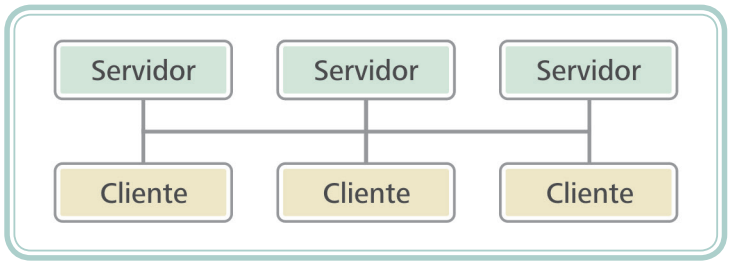


Figura 2.4: Arquitetura cliente/servidor – comunicação entre vários clientes e vários servidores
Fonte: CTISM, adaptado de http://www.unlu.edu.ar/~tyr/tyr/TYRanterior/Fundamentos_da_%20Arquitetura_Cliente-Servidor.pdf

É importante salientar que os modelos apresentados nas Figuras 2.2, 2.3 e 2.4 representam o mesmo modelo, visto sob pontos de vista distintos.

No modelo par a par, o cliente também faz o papel de servidor, e o servidor faz o papel de cliente. Na arquitetura simples, é possível perceber que o servidor fica executando o serviço, porém ele não inicia nenhum processo, fica aguardando o cliente enviar uma requisição. Quem inicia a comunicação entre o cliente e o servidor é o cliente. Já na arquitetura par a par, pelo fato de, em certos momentos um cliente se comportar como servidor, e um servidor se comportar como cliente, qualquer um pode iniciar uma comunicação. Nesse modelo, qualquer dispositivo da rede pode tornar-se um cliente ou um servidor. A Figura 2.5 representa a arquitetura par a par.

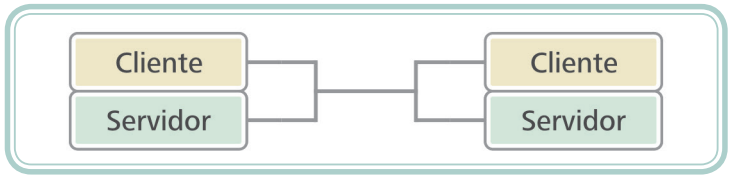


Figura 2.5: Arquitetura cliente/servidor par a par
Fonte: CTISM, adaptado de http://www.unlu.edu.ar/~tyr/tyr/TYR-anterior/Fundamentos_da_%20Arquitetura_Cliente-Servidor.pdf

O último modelo apresentado é o de comunicação em vários níveis. Nele, existe a situação onde o servidor algumas vezes faz o papel de cliente, enviando uma solicitação para outro servidor. A Figura 2.6 representa esse modelo.

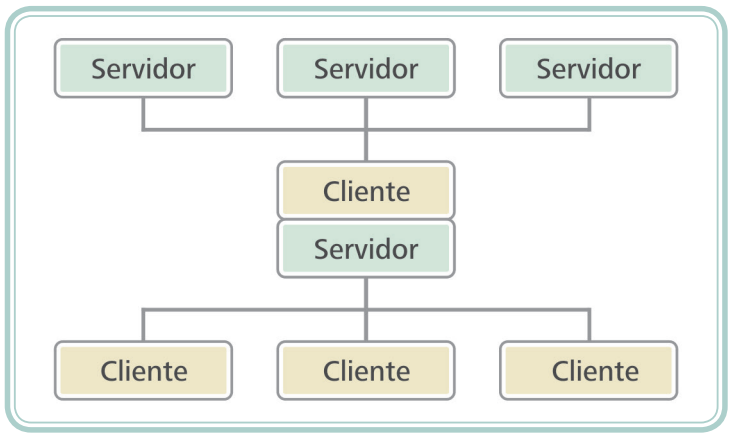


Figura 2.6: Arquitetura cliente/servidor multinível
Fonte: CTISM, adaptado de http://www.unlu.edu.ar/~tyr/tyr/TYR-anterior/Fundamentos_da_%20Arquitetura_Cliente-Servidor.pdf

Deve-se lembrar de que cada modelo é adequado para um contexto, não existe um melhor ou um pior. Cada modelo se encaixa em um caso. Um estudo deve ser realizado para averiguar o modelo que vai atender às necessidades de implementação do sistema.

Resumo

Essa aula apresentou considerações iniciais sobre sistemas cliente/servidor, trazendo seu conceito e exemplos de sua aplicação. Apresentou também, as diferentes arquiteturas de sistemas cliente/servidor juntamente com seu conceito e funcionamento. Vale ressaltar que cada arquitetura cliente/servidor adéqua-se a um caso diferente. Por esse motivo, a instalação de um sistema cliente/servidor deve ser planejada para não desperdiçar recursos.

Atividades de aprendizagem

1. Explique o que é um sistema cliente/servidor.

2. Descreva as arquiteturas cliente/servidor existentes.

Aula 3 – Ambientes de servidor

Objetivos

Conhecer os ambientes existentes em servidores.

3.1 Considerações iniciais

Como abordado em aulas anteriores, os servidores são capazes de automatizar e melhorar a estrutura de um sistema ou de uma rede de computadores. O objetivo de um servidor é oferecer um ou mais serviços. O servidor, na grande maioria das vezes, é um computador com grande capacidade de processamento e armazenamento. Destinado apenas à execução de um ou mais serviços. Dessa forma, esse computador é utilizado somente por um usuário, quando este está realizando alguma manutenção ou instalação de um novo serviço. Nesse caso, esse usuário é o administrador do sistema e não usuário de um recurso da rede. Em outras palavras, o servidor nunca é utilizado como se fosse uma máquina para uso de um recurso da rede. Por exemplo, um servidor *web* nunca é utilizado para acessar páginas de internet. É costume evitar o uso desnecessário do servidor, pois assim se economizam recursos e diminui a possibilidade de acontecerem problemas no servidor.

Os servidores têm grande participação em uma rede de computadores e podem desempenhar as mais diversas atividades. Em uma rede SOHO (*Small Office/Home Office*), por exemplo, que muitas vezes é composta apenas por poucos computadores e com acesso à internet, possui um servidor no próprio roteador que é responsável por fazer a distribuição do acesso à internet para os computadores da rede.

Dentre os serviços encontrados com grande frequência em uma rede de computadores, estão os que têm como objetivo otimizar os recursos da rede como armazenamento prévio de informações de páginas de internet, armazenamento de arquivos, servidores de *e-mail*, servidores de banco de dados, serviços que têm como objetivo melhorar a segurança da rede como um todo, serviços para implementar políticas de segurança na rede.

É possível afirmar que os servidores estão presentes na grande maioria das redes e que são de grande importância. Eles oferecem formas de melhorar a administração da rede e meios de implementar serviços em larga escala.

3.2 Soluções

Dentre as mais diversas soluções implementadas em servidores, algumas são mais utilizadas por oferecerem recursos comuns a vários casos, conforme a lista a seguir.

- **Servidores *web*** – têm como objetivo armazenar uma ou mais páginas de internet. É através desse servidor que um *website* será disponibilizado para os usuários e na grande maioria das vezes, ele também armazena um servidor de banco de dados que armazenará as informações utilizadas por formulários da *website*. A Figura 3.1 mostra o acesso a um servidor *web*.

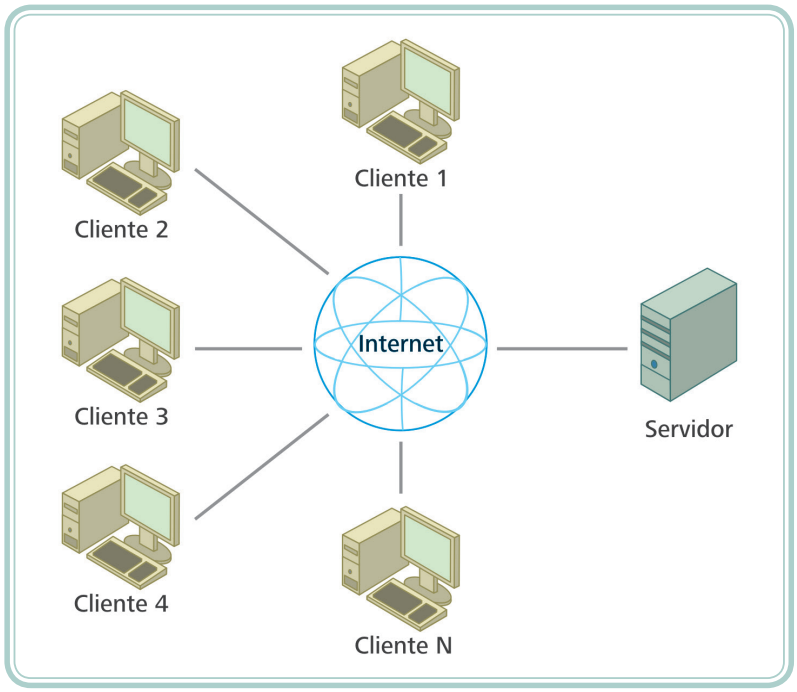


Figura 3.1: Acesso a um servidor *web*
Fonte: CTISM

- **Servidor de arquivos** – esse servidor armazena qualquer arquivo em uma pasta compartilhada na rede. Esses arquivos podem ser acessíveis a todos os usuários da rede, ou seja, é uma pasta pública. Também podem ser acessíveis apenas para usuário e senha. Nesse segundo caso, visto que o acesso é realizado por usuário e senha, pode-se fazer uma restrição de acesso a apenas um grupo de usuários. Também é possível disponibilizar uma pasta em modo leitura. Isso significa que os usuários poderão acessar a pasta e copiar o conteúdo dela, mas não poderão inserir arquivos nessa pasta nem alterar os arquivos que nela estão. A Figura 3.2 mostra o acesso a um servidor de arquivos.

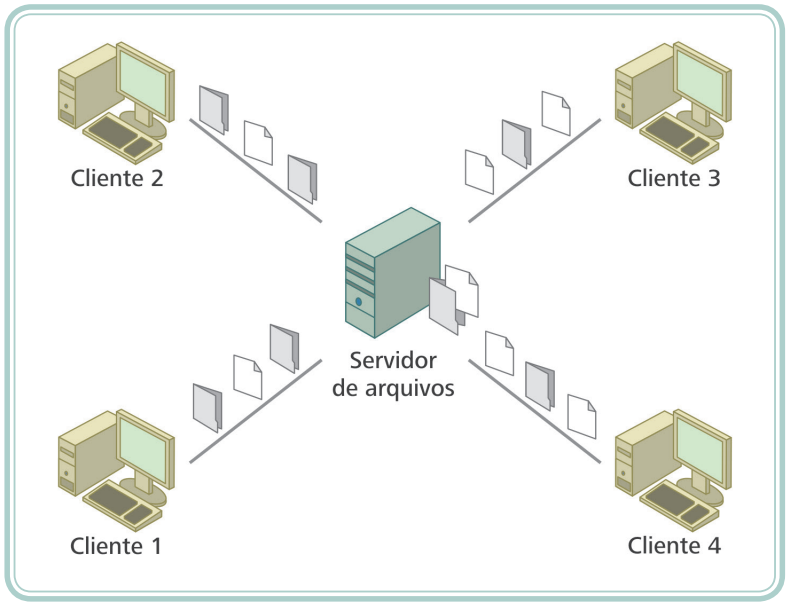


Figura 3.2: Acesso a um servidor de arquivos
Fonte: CTISM

- **Servidor de impressão** – como o próprio nome diz, esse servidor está conectado fisicamente a uma impressora e permite que os usuários da rede utilizem a impressora remotamente. É possível, através da configuração do servidor, definir cotas de impressão e limitar o uso do recurso para um grupo de usuários. A Figura 3.3 mostra um ambiente com um servidor de impressão.

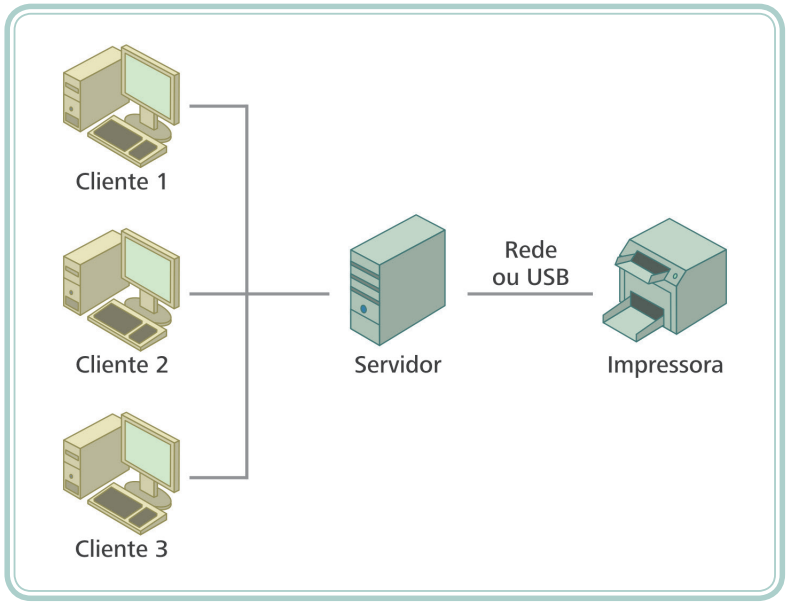


Figura 3.3: Ambiente com um servidor de impressão
Fonte: CTISM

- **Servidor de correio eletrônico** – muitas empresas têm o costume de implantar seus próprios servidores de correio eletrônico (*e-mail*) para evitar que seus funcionários utilizem seus *e-mails* pessoais para o trabalho. A Figura 3.4 mostra um ambiente com servidor de correio eletrônico Postfix.

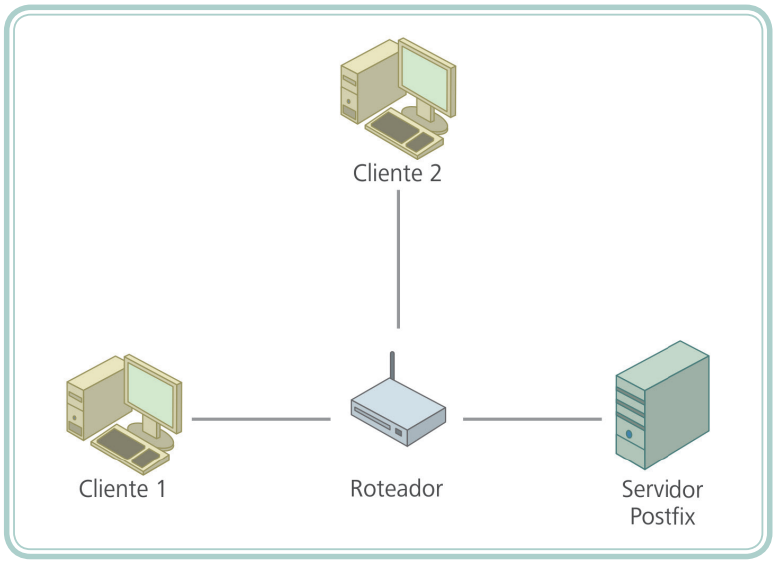


Figura 3.4: Ambiente com um servidor de correio eletrônico Postfix
Fonte: CTISM

- **Servidor DNS (*Domain Name System*)** – tem como objetivo traduzir endereços de *websites*. Quando o usuário digita o endereço de um *website* que ele quer acessar, essa requisição é enviada para o servidor DNS. Ele, por sua vez, consulta uma tabela para descobrir o endereço IP do computador que armazena a *website* solicitada. Caso ele encontre o endereço IP, ele envia a requisição da página *web* para o endereço encontrado. Caso ele não encontre, ele envia a mesma solicitação para um servidor que possua mais informações. É comum utilizar o servidor DNS em conjunto com uma aplicação para armazenamento de informações em *cache*. Isso funciona da seguinte forma: quando um usuário acessa uma página de internet, o servidor armazena a estrutura dessa página, como o posicionamento dos elementos na tela. Quando essa *website* for acessada por qualquer outro usuário da rede, essa informação é carregada do servidor de *cache* de DNS e não do servidor *web*. Assim, economiza-se a banda da conexão com a internet. A Figura 3.4 mostra uma requisição a um servidor DNS. No passo 1, o cliente envia o endereço *web* com o qual deseja se comunicar, no caso "icann.org"; ao servidor DNS, no passo 2 retorna o endereço IP onde o *site* solicitado está armazenado, 192.0.2.0, neste exemplo.

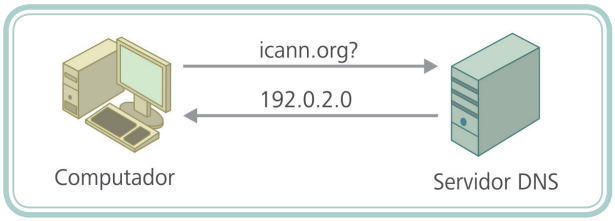


Figura 3.5: Funcionamento de um servidor DNS
Fonte: CTISM

- **Servidor de banco de dados** – várias aplicações utilizam dados de usuários e transações no decorrer do seu uso. Eles ficam armazenados em bancos de dados que possuem uma estrutura avançada de segurança e administração, bem como ferramentas de integridade de dados e ferramentas que permitem que esses dados possam ser acessados das mais diversas formas.

- **Servidor de internet/firewall** – em redes de médio e grande porte, é comum o uso de um servidor que irá compartilhar a conexão da internet com os computadores da rede. Isso é feito, porque com ele é possível aplicar uma enorme quantidade de ferramentas de segurança e filtros de conteúdo indevido, da mesma forma que limita o uso de banda. É possível afirmar que o objetivo geral de um firewall é permitir que qualquer requisição realizada dentro da rede seja enviada para a internet (caso os filtros de conteúdo permitam) e que toda e qualquer requisição enviada da internet para dentro da rede seja barrada. A Figura 3.6 mostra uma rede protegida por um firewall.

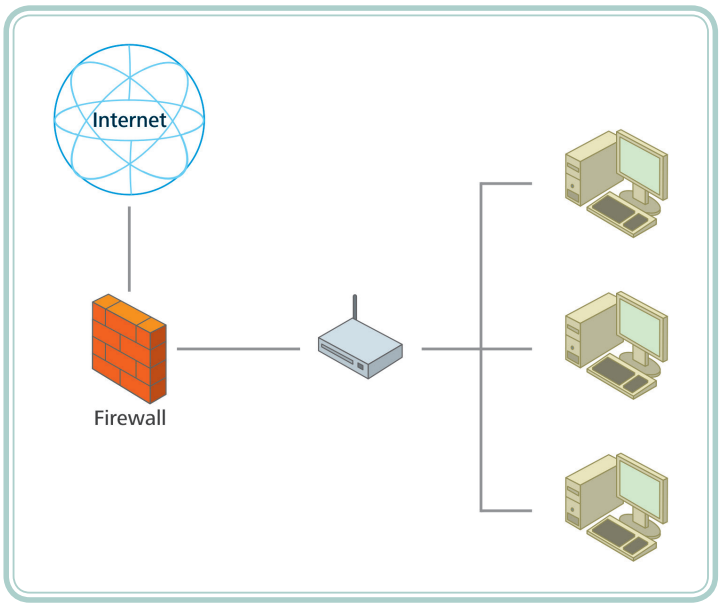


Figura 3.5: Rede protegida por firewall
Fonte: CTISM

Obviamente, os servidores possuem as mais diversas funcionalidades que incluem autenticação de usuários na rede, compartilhamento de arquivos e recursos, dentre outras. As vantagens de se utilizar um servidor é concentrar a administração do recurso. Dessa forma, caso seja necessário alterar uma configuração de um serviço utilizado pela rede, basta alterar a configuração desse serviço no servidor, reiniciar o serviço e, automaticamente, todos os outros computadores da rede passarão a operar de acordo com a nova configuração do serviço.

Outra grande vantagem do uso de servidores é que em alguns casos, ele concentra também informações sobre os usuários da rede. Se ele não existisse, existiriam várias cópias da mesma informação em vários computadores da rede. Um exemplo simples disso é a autenticação dos usuários nos computadores. Imagine que exista uma rede de uma empresa que possui uma enorme quantidade de usuários. Cada usuário possui sua mesa com seu computador e nesse computador ele possui vários arquivos que utiliza para trabalhar. Agora imagine que esse computador apresentou um problema e deve ser substituído. Todos os dados do usuário e seus arquivos serão perdidos. Caso exista uma cópia dessas informações, elas devem ser colocadas no novo computador. Esse problema não parece ser grande, quando visto como um problema de um usuário, mas imagine se esse problema ocorrer simultaneamente com uma grande quantidade de usuários.

Para resolver esse problema, pode se utilizar um servidor que possua as informações de todos os usuários da rede, bem como uma pasta compartilhada para cada um deles. Assim, quando um usuário autenticar-se em um computador qualquer da rede, esse computador irá verificar no servidor se o usuário e senha inseridos estão corretos. Caso estejam, irá carregar todas as configurações que o usuário utiliza em seu computador, da mesma forma que irá carregar o acesso à sua pasta que está, na verdade, armazenada no servidor. Dessa forma, qualquer usuário pode utilizar qualquer computador da rede que suas informações estarão disponíveis a qualquer momento.

Esse cenário também pode vir acompanhado com políticas de segurança implementadas na rede. Um grupo de usuários pode ter acesso limitado dentro do sistema de uma empresa. Por exemplo, o setor de recursos humanos pode ter acesso somente ao sistema que gerencia os recursos humanos, enquanto o presidente dessa empresa terá acesso a todo o sistema. Assim, caso um funcionário do setor de recursos humanos se autentique no computador utilizado pelo presidente, com seu usuário e senha, as permissões que esse usuário terá

são as de um funcionário do setor de recursos humanos e somente terá acesso à parte de recursos humanos do sistema. Já caso o presidente da empresa se autentique em um computador do setor de recursos humanos, ele terá acesso total ao sistema da empresa. É visível, portanto, que as políticas de acesso e segurança estão vinculadas ao usuário e não ao computador. Entretanto, vale lembrar que essas políticas também são aplicáveis aos computadores. Nesse caso, mesmo que o presidente da empresa se autentique em um computador do setor de recursos humanos, ele terá acesso apenas ao que é permitido para um computador do setor de recursos humanos.

Resumo

Nessa aula apresentaram-se ambientes de servidor, características de servidor de rede, sua utilização e sua utilidade para o sistema como um todo.

Também foram apresentados diversos tipos de servidores largamente utilizados, dada a importância dos serviços por eles prestados.

Atividades de aprendizagem

1. Cite três tipos de servidores apresentados nessa aula e sua finalidade.

2. Quantos serviços podem ser executados em um mesmo servidor?

Aula 4 – Servidores *web*

Objetivos

Conhecer o funcionamento de servidores *web* e servidores DNS, instalá-los e configurá-los em um sistema operacional GNU/Linux.

4.1 Servidor *web*

Servidores *web* (também conhecidos como servidores HTTP) são os responsáveis por processar requisições HTTP (*Hypertext Transfer Protocol*) que, por sua vez, são as responsáveis pela troca de solicitações de páginas de internet entre um usuário e um servidor *web*.

Em outras palavras, o servidor *web* é quem disponibiliza as páginas *web* de cada *site*. Por exemplo, cada vez que um usuário abre um navegador, digita o endereço de uma *website* e pressiona um botão para carregá-la, o navegador envia uma requisição HTTP para o endereço digitado pelo usuário. Essa *website*, disponibilizado por um servidor *web*, recebe a requisição HTTP e responde com a página *web* solicitada. Essa página é o conteúdo de um arquivo de extensão .html ou .htm que é interpretada pelo navegador e exibida para o usuário.

Normalmente, um servidor *web* não é implementado sozinho, pois se for, ele será capaz de disponibilizar para os usuários apenas páginas HTML (*Hypertext Markup Language*), que permitem somente a exibição estática de conteúdo. Isso significa que cada informação presente na *website* deve ser inserida manualmente, página por página, alterando-lhe o código HTML. Dessa forma, não é possível a construção de páginas com recursos como formulários, cadastros e gerenciamento de conteúdo. É comum os servidores *web* virem acompanhados de servidores PHP (*Hypertext PreProcessor*) ou servidores ASP (*Active Server Pages*), que permitem o uso de conteúdo dinâmico, juntamente com um servidor de banco de dados (como o MySQL) responsável pelo armazenamento de informações, ou seja, a persistência de dados.

É muito comum, em um servidor *web*, encontrar-se o que é conhecido como LAMP, que significa Linux+Apache+MySQL+PHP que são, respectivamente, o sistema operacional presente no servidor, o servidor *web*, o servidor de banco de dados e o servidor de conteúdo dinâmico, capaz de interpretar

páginas dinâmicas desenvolvidas na linguagem PHP. Vale lembrar que todos os aplicativos citados são gratuitos. Isso significa que qualquer usuário pode baixá-los e utilizá-los sem nenhum custo. Ainda assim, os aplicativos Apache, MySQL e PHP podem ser instalados em um sistema operacional Windows.

Já as páginas dinâmicas desenvolvidas na linguagem ASP são normalmente encontradas em servidores Microsoft Windows, pois a linguagem também é desenvolvida pela Microsoft e por esse motivo, em um computador com uma versão para servidores do sistema operacional da Microsoft, o conjunto necessário para ter um servidor *web* completo (servidor *web*, servidor de banco de dados e servidor de conteúdo dinâmico) já vem pré-instalado, bastando apenas configurá-lo. Dessa forma, os aplicativos para as finalidades de servidor *web* e persistência são também desenvolvidos pela Microsoft. É possível instalar um servidor de conteúdo dinâmico em um servidor Linux, porém a solução não é desenvolvida pela Microsoft.

4.2 Implementação

Geralmente, instalar um servidor *web* em um sistema operacional Linux, consiste na instalação dos pacotes necessários para a sua execução, e na inicialização de um serviço que fará com que o servidor seja efetivamente ativado. No caso de um servidor *web* (e também na grande maioria dos aplicativos semelhantes no Linux), quando o aplicativo é instalado, ele já traz consigo versões dos arquivos de configuração que contém uma configuração padrão que permite que o serviço seja executado com as configurações básicas. Portanto, basta alterar o(s) arquivo(s) de configuração para personalizar o serviço de acordo com a necessidade de cada caso.

Basicamente, existem duas formas de instalar os pacotes necessários para um servidor *web* completo em um sistema operacional Linux. A maneira mais fácil é usar um gerenciador de pacotes. Nesse caso, utiliza-se um gerenciador de pacotes presente na distribuição Linux (o gerenciador de pacotes pode variar de distribuição para distribuição). As vantagens de sua utilização são:

- A instalação das bibliotecas necessárias é realizada automaticamente.
- As versões dos pacotes são sempre as mais estáveis.
- Praticidade, pois basta uma linha de comando.

- Menor probabilidade de erros, pois os pacotes são designados àquela distribuição.

A única desvantagem marcante desse método é que geralmente, em repositórios de pacotes designados para servidores, a versão dos pacotes é sempre a mais estável, o que significa que ela provavelmente não é a mais atual. Isso é feito para evitar conflito entre bibliotecas e possíveis instabilidades que pacotes novos possam apresentar. Dessa forma, caso seja necessária alguma funcionalidade existente apenas em uma versão que não está presente nos repositórios, esse pacote precisa ser instalado manualmente e de forma cautelosa.

Os gerenciadores de pacotes baixam os pacotes designados para a distribuição Linux de um repositório mantido pela equipe que desenvolveu a distribuição, portanto os pacotes dos repositórios sempre são os mais recomendados.

A outra maneira de instalar pacotes, mais complicada, consiste em baixá-los com os códigos fontes dos aplicativos e instalá-los manualmente, compilando um por um, inclusive as bibliotecas. A única vantagem deste método, como se descreveu anteriormente, é que os pacotes presentes nos repositórios de servidores são os mais estáveis e assim, uma nova versão que apresente funcionalidade necessária precisa ser instalada manualmente.

Em contrapartida, as desvantagens desse método são:

- Apresenta grande probabilidade de problemas de compilação e incompatibilidade de bibliotecas.
- É trabalhoso, pois cada pacote deve ser compilado e instalado manualmente.
- Requer mais conhecimento do profissional que irá instalar os pacotes.
- Pode haver problemas nas instalações desse tipo em um servidor que já esteja operante.

Pelo fato de o objetivo dessa aula não ser instalação de pacotes, o método utilizado será através de gerenciador de pacotes. Para isso, a distribuição Linux tomada como base será o Ubuntu, que é baseado no Debian e que utiliza o gerenciador apt-get. Essa distribuição foi escolhida, porque é uma das mais utilizadas no mundo atualmente, possuindo uma versão para computadores pessoais com uma interface gráfica amigável, inclusive para o gerenciador de pacotes e uma versão para servidores, que não possuem interface gráfica, eliminando assim a possibilidade de problemas causados por mau uso do sistema operacional.

Para instalar um pacote, utilizando o comando apt-get, basta seguir a seguinte estrutura:

```
apt-get install nome-do-pacote
```

Existem inúmeras distribuições Linux e também inúmeros gerenciadores de pacotes, sendo que cada distribuição tem o seu gerenciador. Caso o sistema operacional Linux não seja um baseado em Debian, basta procurar no manual da distribuição para descobrir o gerenciador de pacotes que ela usa.

No entanto, para descobrir o nome do pacote, podemos utilizar o sistema de buscas do apt-get, através do apt-cache (Versão 10.04 ou mais recente do Ubuntu). Ex.: apt-cache search nome-do-pacote. A Figura 4.1 mostra um exemplo da execução do comando de busca. Buscado o termo apache2, significa que todos os pacotes que tiverem o termo "apache2" no nome ou descrição e que estiverem presentes no repositório aparecerão como resultado. A partir disso, é possível filtrar os pacotes necessários para a instalação de qualquer coisa em um sistema Linux e, conforme já mencionado, ao instalar um aplicativo através do gerenciador de pacotes, todos os outros pacotes que forem requisitos do que está sendo instalado, serão instalados também.

Uma das vantagens da utilização de servidores com sistema operacional Microsoft Windows, é que eles têm assistentes gráficos que facilitam a instalação, configuração e gerenciamento de pacotes e serviços, porém esse tipo de sistema operacional normalmente tem um custo elevado e, por utilizarem uma interface gráfica, consomem mais recursos do computador. Outro problema que pode ocorrer é a má utilização do sistema operacional, pois ele oferece uma interface gráfica semelhante à versão para computadores pessoais. Já os sistemas operacionais Linux, quando instalados em servidores, são mais leves (nesse caso geralmente não possuem interface gráfica) e gratuitos, porém mais difíceis de configurar.

Mais informações sobre como alterar um arquivo de configuração de um aplicativo, basta olhar o seu manual.

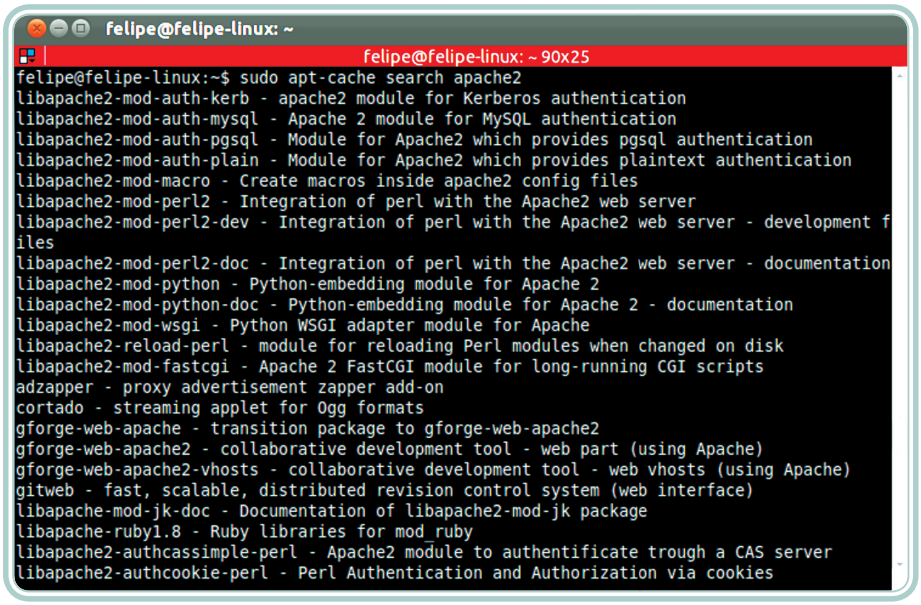


Figura 4.1: Exemplo de comando apt-cache
Fonte: Autores

Dentre os pacotes mostrados no resultado da busca, é possível identificar os pacotes do servidor HTTP Apache2, como mostra a Figura 4.2.

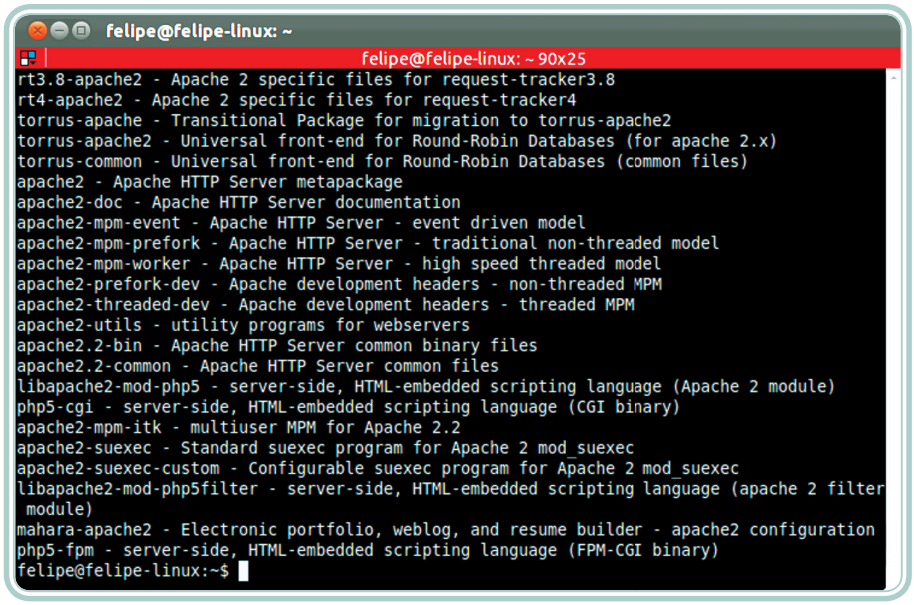


Figura 4.2: Pacotes do Apache2 no resultado da busca
Fonte: Autores

Após a execução do comando apt-cache, é possível verificar os pacotes presentes no repositório e escolher qual(is) será(ão) instalado(s). Para instalar um pacote ou mais de um, é utilizado o comando apt-get, conforme a seguinte estrutura:

apt-get install nome-do-pacote

Nesse caso, é obrigatório que seja informado exatamente o nome do pacote. É possível instalar mais de um pacote por vez, inserindo os nomes dos pacotes um após o outro, separados apenas por espaços. A Figura 4.3 mostra a execução do comando apt-get.

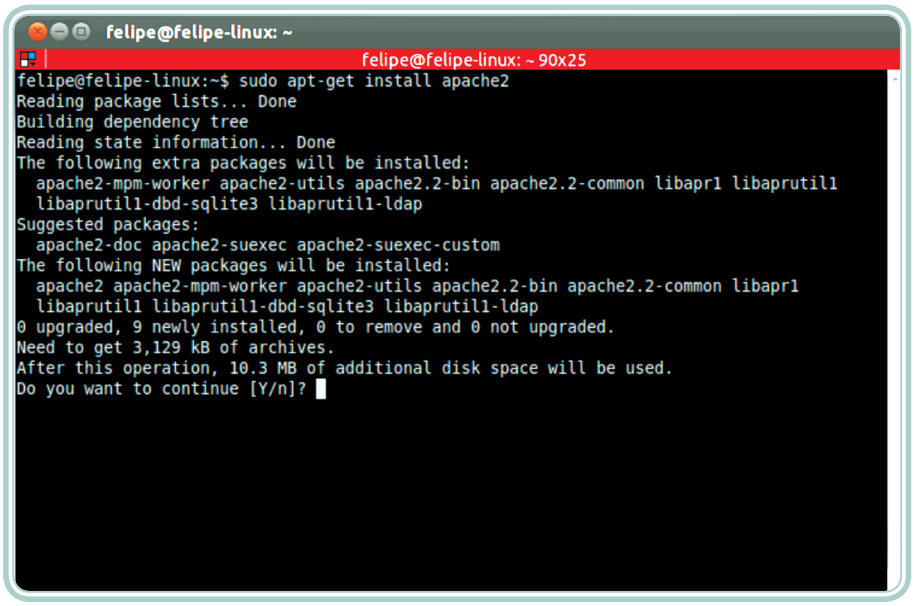


Figura 4.3: Execução do comando apt-get com o pacote encontrado na busca
Fonte: Autores

4.3 Configuração

Como se mencionou anteriormente, as ferramentas desenvolvidas para servidores Linux, em sua instalação, trazem consigo uma configuração padrão, capaz de fazer com que o serviço seja executado normalmente. Entretanto, por ser uma configuração padrão, pode não atender às necessidades do local onde ela está sendo instalada e, por isso, alterações são necessárias para fazer com que a aplicação funcione conforme o desejado.

O editor Vim é uma melhoria do editor VI, portanto, os dois são extremamente semelhantes. Mesmo sem aparentar, o Vim é um editor extremamente poderoso. Ainda assim, algumas pessoas preferem o Nano, por possuir uma interface mais amigável.

Em sistemas Linux, a configuração dessas ferramentas é realizada através da alteração de um arquivo de configuração que é lido pela aplicação e interpretado, conforme uma linguagem não específica pré-determinada. Esse arquivo, na maioria das vezes, vai conter a configuração padrão, bastando alterá-lo em determinados lugares para personalizá-lo. Assim, não é necessário o conhecimento prévio da linguagem utilizada no arquivo.

Portanto, basta editar o arquivo com um editor de textos qualquer. Caso o sistema possua interface gráfica, pode-se utilizar um editor gráfico. Por exemplo, na interface de usuário Gnome, o editor de textos padrão é o Gedit (Figura 4.4) e na interface de usuário KDE, é o Kate (Figura 4.5) ou basta

executar um aplicativo que permita acesso ao terminal de comandos e usar um editor de texto que utilize apenas recursos do terminal de comandos (os editores VI, Vim (Figura 4.6) e Nano (Figura 4.7) são os mais conhecidos e utilizados). Caso o sistema não possua interface gráfica, apenas os editores desenvolvidos para o terminal de comandos poderão ser utilizados. Nesse documento, o editor utilizado será o Nano, por possuir uma interface de fácil manuseio.

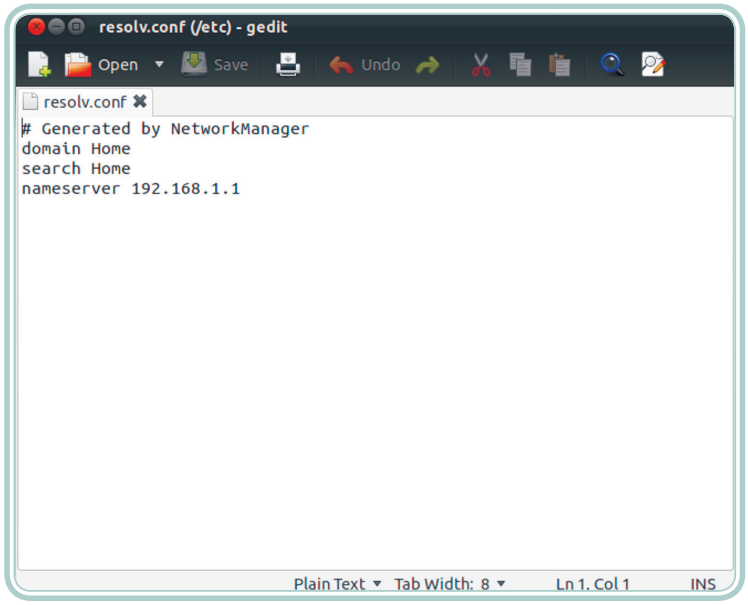


Figura 4.4: Editor de textos Gedit
Fonte: Autores

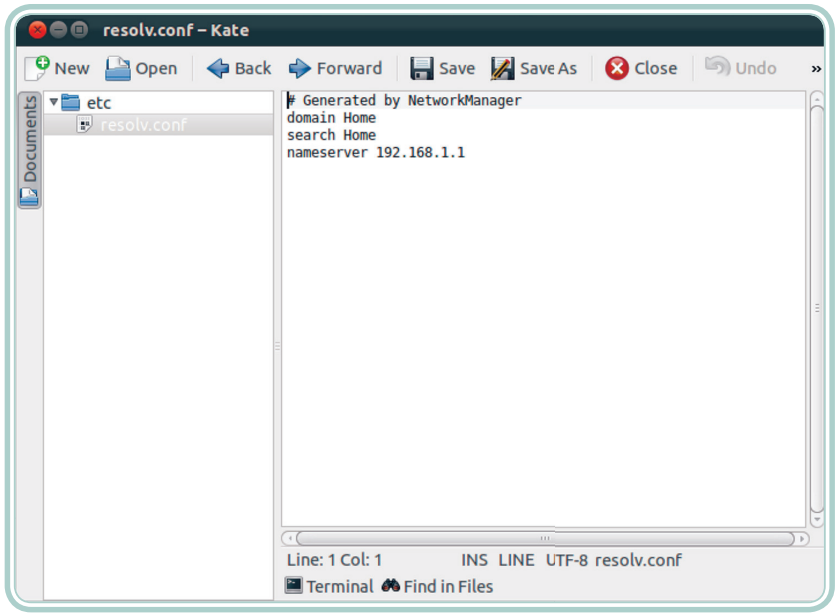


Figura 4.5: Editor de textos Kate
Fonte: Autores

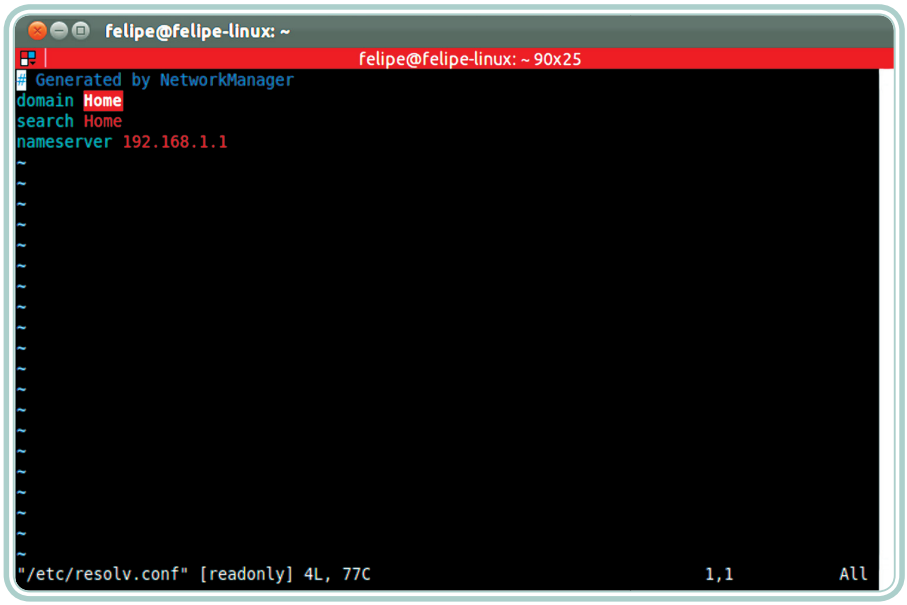


Figura 4.6: Editor de textos Vim
Fonte: Autores

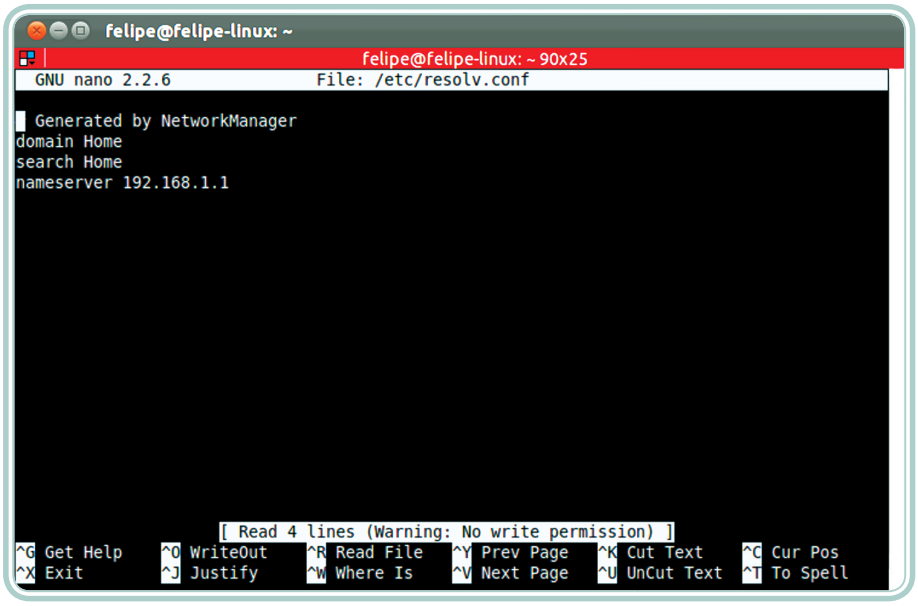


Figura 4.7: Editor de textos Nano
Fonte: Autores

Para editar um arquivo no Linux, utilizando o editor Nano, basta executar o comando "nano" seguindo do nome do arquivo, juntamente com o endereço completo no qual ele se encontra. Por exemplo, para editar o arquivo resolv.conf, que fica na pasta /etc do Linux, que possui o endereço /etc/resolv.conf, basta executar o seguinte comando:

```
nano /etc/resolv.conf
```

É importante lembrar que os arquivos que estão dentro da pasta /etc são arquivos de configuração de aplicativos e, portanto, um usuário sem permissão do administrador será capaz de abrir e editar o arquivo, mas não será capaz de sobrescrever o seu conteúdo. Para isso, em um sistema Ubuntu Linux, basta executar o comando "sudo" antes do comando "nano", como se mostra a seguir:

```
sudo nano /etc/resolv.conf
```

Assim, o sistema pedirá que o usuário insira a sua senha e permitirá que ele sobrescreva o arquivo.

O comando sudo pode não aparentar aumento de segurança, mas seu objetivo é que o usuário esteja ciente do que está fazendo. Por exemplo, caso um usuário esteja alterando um arquivo através do comando sudo, ele está ciente de que aquele arquivo é importante para o funcionamento do sistema e de aplicativos e por isso são necessárias permissões de administrador para editá-lo. Assim o usuário poderá tomar precauções para evitar danos ao sistema.

Uma vez instalado o servidor HTTP, basta iniciar o serviço que ele já estará funcionando, porém caso seja necessário editar seu arquivo de configurações, ele está presente no diretório /etc/apache2/httpd.conf. Um dos parâmetros que normalmente é alterado é a porta na qual o servidor irá funcionar. A porta padrão é a 80. Caso a porta seja alterada, deve-se informar no navegador ao acessar o servidor. Por exemplo, caso a porta for alterada para 8080, para acessar o servidor HTTP é necessário acessar o endereço http://localhost:8080 no navegador.

Não existe um valor certo ou errado para as configurações do arquivo. As alterações devem ser feitas conforme cada caso.

Após o término da edição do arquivo, basta salvá-lo e iniciar ou reiniciar o serviço que o mantém funcionando. É possível fazer isso executando o seguinte comando:

```
sudo /etc/init.d/httpd start (ou restart)
```

Ou o seguinte:

```
sudo servisse httpd start (ou restart)
```

Para salvar um arquivo que está sendo editado no editor Nano, basta pressionar simultaneamente as teclas Ctrl e O. Para fechar o Nano, a combinação de teclas é Ctrl e W.

Para saber o estado de um serviço basta utilizar o parâmetro status. Por exemplo, para saber o estado do serviço httpd, basta executar sudo /etc/init.d/httpd *status* ou sudo servisse httpd status

A vantagem do primeiro modo é que ele consiste na execução de um arquivo (/etc/init.d/httpd) acompanhado de um parâmetro (start, stop, restart ou status) e com o auxílio do recurso oferecido pela tecla TAB no terminal do Linux, ao digitar o comando sudo /etc/init.d no terminal e pressionar a tecla TAB duas vezes, aparecerão todos os arquivos executáveis presentes dentro da pasta /etc/init.d. Assim é possível saber exatamente o nome do arquivo (no caso o arquivo se chama httpd) e executá-lo. A Figura 4.8 mostra a reinicialização do serviço httpd; a Figura 4.9, o seu estado.

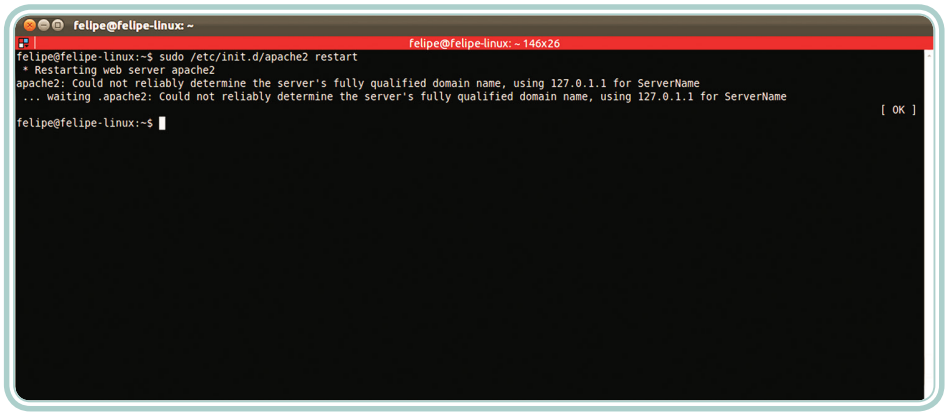


Figura 4.8: Reinicialização do serviço httpd
Fonte: Autores

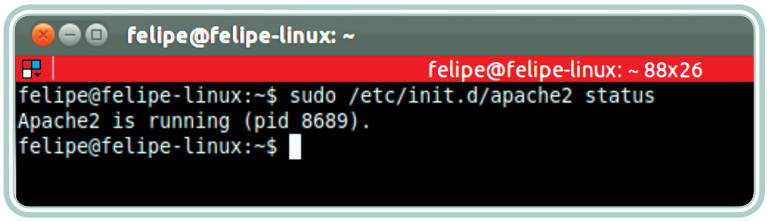


Figura 4.9: Estado do serviço httpd
Fonte: Autores

Após o término da instalação e configuração, o servidor HTTP já está funcionando. Para testá-lo, basta acessar o endereço http://localhost ou http://127.0.0.1 através de um navegador qualquer na mesma máquina onde o servidor HTTP está instalado. Esses endereços, quando acessados, têm como destino o próprio computador. Para acessar o servidor HTTP de outro computador, basta acessar através do endereço IP (*Internet Protocol*) do computador onde o servidor HTTP está instalado. A Figura 4.10 mostra a página padrão trazida pelo servidor HTTP Apache.

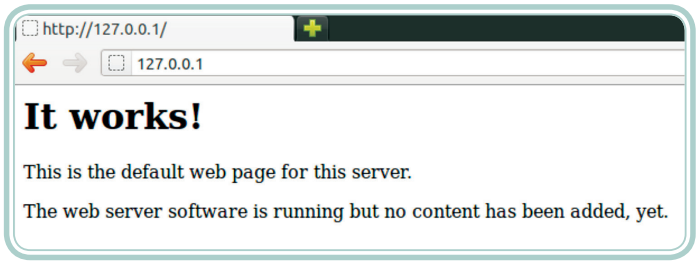


Figura 4.10: Servidor HTTP sendo acessado pelo navegador
Fonte: Autores

Até o momento, o servidor HTTP pode ser acessado apenas através da rede local, ou seja, apenas em computadores que estiverem na mesma rede (ou na mesma sub-rede de endereços IP) poderá acessar o servidor. Para que esse servidor possa ser acessado pela internet, a conexão deve ser do tipo empresarial, pois o servidor HTTP usa a porta 80 que, normalmente, é bloqueada pelos provedores de internet. Com uma conexão empresarial e devidamente funcional (velocidade e qualidade adequadas) e com o endereço IP público (fornecido pelo provedor de acesso à internet), é possível acessar o *site* de qualquer lugar do mundo, através do endereço IP. Mas, quando queremos acessar um *site*, digitamos o endereço em um navegador e não o seu endereço IP. Para poder acessar o servidor HTTP através do nome do *site*, é necessário registrar o endereço do *site* no órgão do país que rege isso. No Brasil, ou seja, para domínios .com.br o responsável é o Registro BR. O responsável pela conversão dos endereços de *sites* para seus respectivos endereços IP é o servidor DNS (*Domain Name System*), que será estudado mais adiante.

4.4 Servidor de conteúdo dinâmico (PHP)

Foi mencionado anteriormente, que um servidor HTTP permite que um usuário requisite uma página de *web* de um *site*, porém ele é capaz de exibir apenas páginas estáticas (em HTML). Isso significa que o servidor não será capaz de interpretar dados enviados pelo usuário como os usados em formulários, funcionalidades que necessitem autenticação ou armazenamento de informações em um banco de dados. Em outras palavras, o servidor não terá condições de exibir conteúdo dinâmico que, no caso dessa aula, é interpretado pela linguagem PHP, através de arquivos com a extensão PHP.

Para fazer com que o servidor HTTP seja capaz de interpretar e interagir com a linguagem PHP, é necessário a instalação do PHP propriamente dito. Para isso, basta buscar os pacotes relacionados ao PHP no gerenciador de pacotes, através do comando apt-cache search nome-do-pacote. A Figura 4.11 mostra os pacotes PHP disponíveis no repositório.

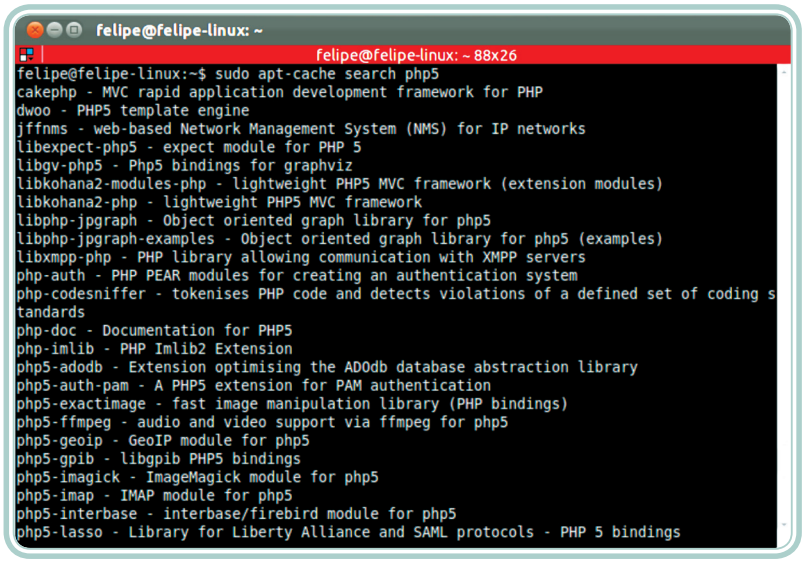


Figura 4.11: Busca de pacotes para a instalação do PHP
Fonte: Autores

O PHP, diferentemente do Apache (que é o servidor *web*, ou servidor HTTP) não tem um serviço rodando no servidor. Ele traz apenas as ferramentas necessárias para que o servidor *web* consiga manipular as páginas PHP. Assim, não é preciso realizar nenhuma configuração, bastando as configurações que vêm por padrão na instalação.

Para testar se a configuração do PHP foi realizada corretamente, é necessária a criação de uma página na linguagem PHP para ser executada pelo servidor. Em sistemas operacionais Linux, as páginas *web* ficam dentro do diretório /var/www/. Após a instalação do servidor HTTP, um arquivo deve ter sido criado dentro desse diretório. Por padrão, esse arquivo se chama index.htm ou index.html. Também por padrão, quando o usuário acessar através de um navegador no próprio servidor os endereços http://localhost ou http://127.0.0.1, o arquivo exibido é o index.htm ou index.html. O servidor *web* é programado para procurar por um arquivo de nome index, com as extensões de páginas *web* (html, htm, php, dentre outras). Ainda assim, caso o usuário queira acessar o conteúdo de uma página html contida em um arquivo com nome diferente, por exemplo, pagina.html, basta incluí-la no final dos endereços anteriormente citados que, no caso, seriam http://localhost/pagina.html ou http://127.0.0.1/pagina.html.

Para realizar o teste do funcionamento do PHP, será criado o arquivo phpinfo.php dentro do diretório /var/www/. Sua localização completa será /var/www/phpinfo.php. Com um editor de texto qualquer (Nano, por exemplo, através do comando sudo nano /var/www/phpinfo.php), é necessário inserir o seguinte conteúdo dentro do arquivo.

```
<?php
phpinfo();
?>
```

A Figura 4.12 mostra o arquivo sendo editado no editor Nano.

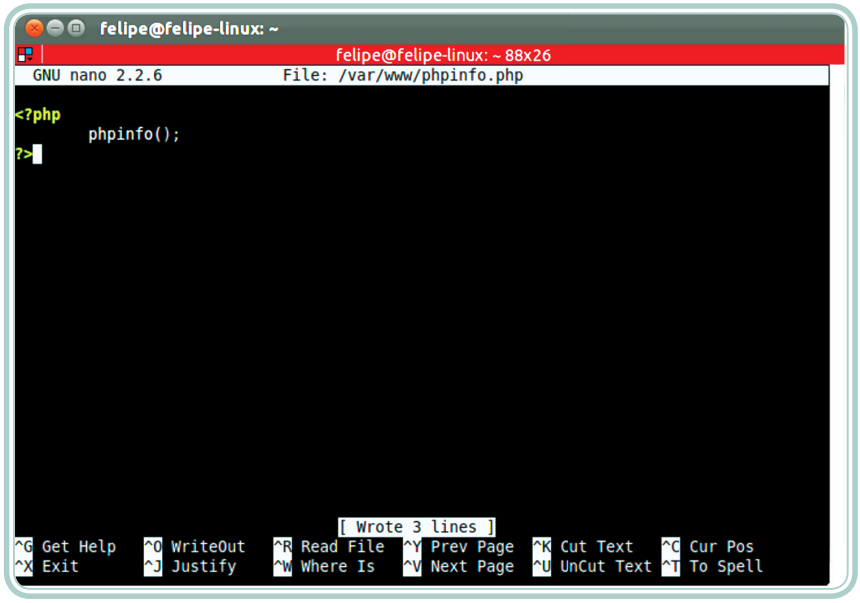


Figura 4.12: Edição do arquivo phpinfo.php
Fonte: Autores

Essa estrutura de comandos irá fazer com que, sempre que a página phpinfo.php for acessada, ela exiba as informações da versão do PHP instalada no servidor. Assim, é possível testar o funcionamento do PHP, acessando o endereço http://localhost/phpinfo.php ou http://127.0.0.1/phpinfo.php. A Figura 4.13 mostra a página phpinfo.php sendo exibida em um navegador.

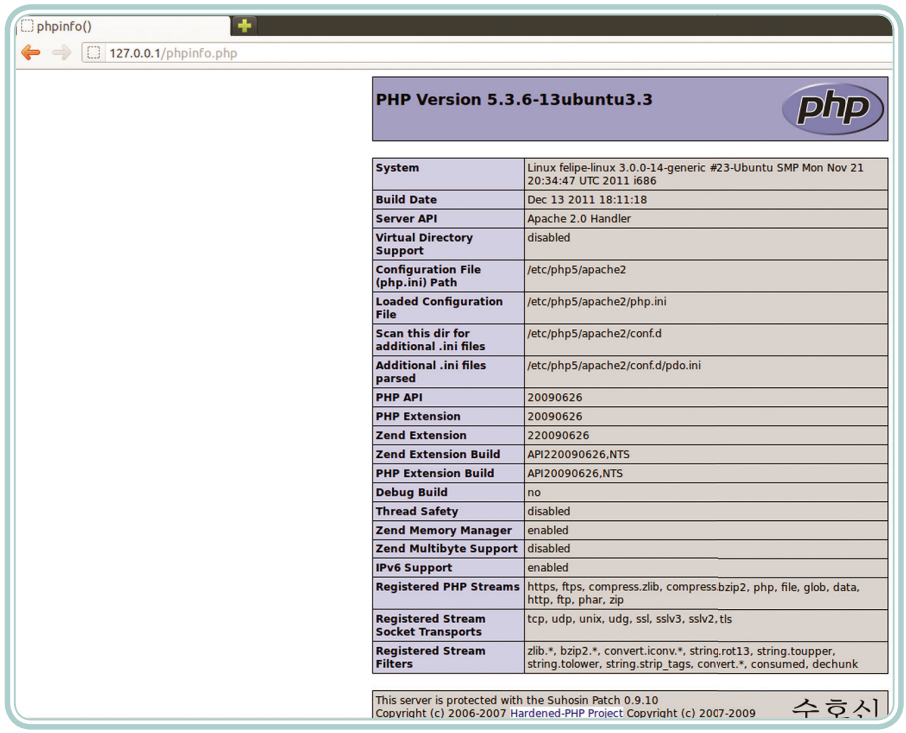


Figura 4.13: Execução de uma página PHP
Fonte: Autores

4.5 Servidor de banco de dados (MySQL)

Para finalizar a configuração de um servidor *web* básico falta a instalação de um servidor de banco de dados. Um banco de dados nada mais é que um sistema capaz de armazenar informações em tabelas, de forma segura, íntegra e que permita acesso rápido às mesmas. Na grande maioria das vezes, um servidor *web* não possui a capacidade de interpretar apenas páginas em HTML, trazendo consigo a capacidade de interpretar conteúdo dinâmico, através do PHP que, por sua vez, precisa armazenar os dados utilizados em seus formulários, necessitando de um banco de dados. Por exemplo, em um *site* de notícias que são atualizadas várias vezes durante o dia, caso um usuário queira acessar uma notícia específica, que foi colocada no *site* há vários dias, basta ele pesquisar a notícia. O servidor do *site* irá buscar no banco de dados, as informações referentes à busca realizada pelo usuário e irá disponibilizá-las na tela, através do PHP.

A instalação do MySQL é realizada da mesma forma que as instalações do servidor HTTP e do PHP através do repositório. A Figura 4.14 mostra os pacotes do MySQL existentes no repositório, através do comando apt-cache.

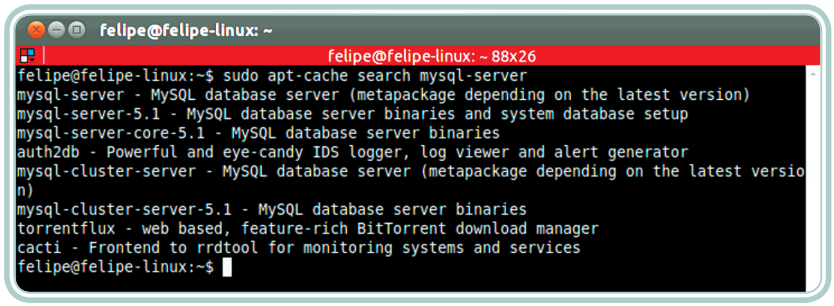


Figura 4.14: Pacotes do MySQL presentes no repositório
Fonte: Autores

Durante a instalação do MySQL, solicita-se ao usuário, que insira uma senha para o usuário root do banco de dados, ou seja, uma senha de um usuário com permissões de administrador, capaz de realizar qualquer operação dentro do banco de dados. É importante, que esse usuário seja usado apenas pelo administrador do banco de dados ou pelo administrador do servidor, pois com ele é possível apagar todas as informações contidas no banco de dados. A linguagem PHP consegue se comunicar com o banco de dados através de um usuário e uma senha que devem ser capazes de manipular os dados referentes apenas ao *site* a que pertencem. Ainda assim, a linguagem PHP presente nos *sites* deve apenas inserir, alterar e excluir informações das tabelas dos bancos de dados e não devem alterar suas estruturas, pois isso pode gerar problemas nos relacionamentos entre as tabelas. A Figura 4.15 mostra a tela onde a senha para o usuário root é solicitada.

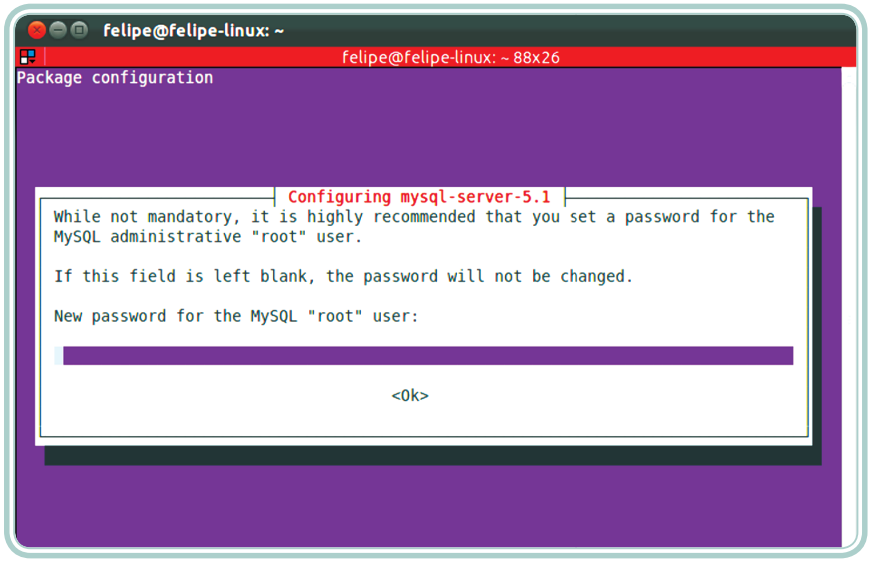


Figura 4.15: Tela do MySQL que solicita a criação da senha para o usuário "root"
Fonte: Autores

A administração do MySQL fica mais fácil através do aplicativo PHPMyAdmin, que pode ser facilmente instalado pelo gerenciador de pacotes. Após sua instalação, basta acessá-la através do endereço http://localhost/phpmyadmin

ou http://127.0.0.1/phpmyadmin. Antes, o serviço do MySQL deve ser iniciado, pelo comando sudo /etc/init.d/mysqld start ou sudo service mysqld start. A Figura 4.16 mostra o PHPMyAdmin sendo executado.

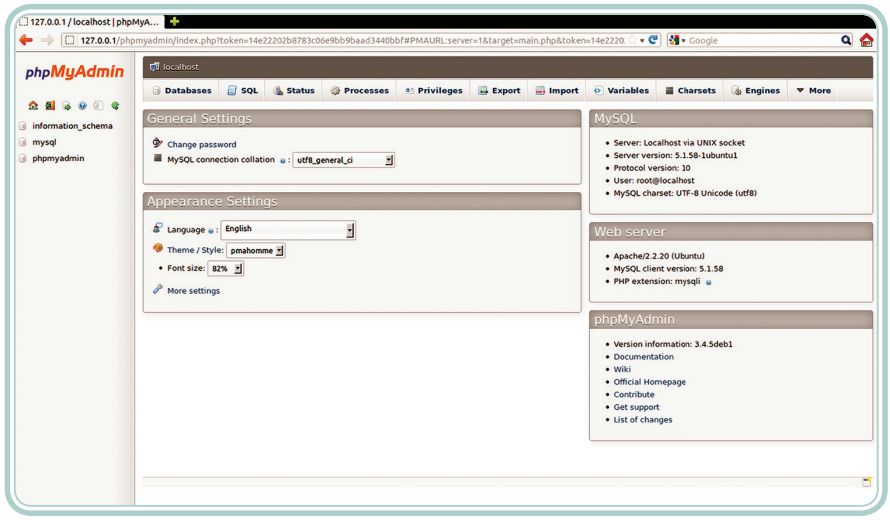


Figura 4.16: PHPMyAdmin
Fonte: Autores

Resumo

Nessa aula, foi explanada uma forma fácil e prática de se instalar um servidor HTTP. A instalação dos aplicativos e suas bibliotecas através de um gerenciador de pacotes facilita todo o processo de instalação, pois é rápida e instala automaticamente todas as dependências requisitadas pelos aplicativos. Ainda são abordadas as alterações necessárias nos arquivos de configuração dos aplicativos para fazer com que os mesmos se adequem ao cenário no qual o servidor será instalado. Com o objetivo de mostrar o funcionamento básico de um servidor de um *website*, por exemplo, foram mostradas as instalações do servidor HTTP (através do aplicativo Apache), fazendo com que o servidor seja capaz de responder a requisições HTTP, ou seja, exibir páginas *web* (escritas na linguagem HTML), de páginas de conteúdo dinâmico (através da linguagem PHP), permitindo assim que o servidor seja capaz de interpretar envio e recebimento de dados através de formulários escritos em PHP e por fim, a utilização de um sistema de gerenciamento de banco de dados (através do MySQL), permitindo que informações enviadas para o servidor possam ser armazenadas nele e, quando solicitadas, exibidas para o usuário.

Atividades de aprendizagem

1. Instalar e configurar um servidor *web*, de acordo com o que se apresentou em aula.

Aula 5 – Servidor FTP

Objetivos

Instalar, configurar e gerenciar um servidor FTP (*File Transfer Protocol*), bem como conhecer o seu funcionamento.

5.1 Considerações iniciais

Um servidor FTP tem como objetivo a disponibilização de arquivos para os usuários. Um exemplo muito comum são os repositórios de sistemas Linux, pois eles trazem uma enorme quantidade de arquivos que ficam acessíveis a todos os usuários do sistema. Outro caso onde é muito comum o uso de servidores FTP é em servidores *web*, pois as páginas de internet ficam em constante desenvolvimento e a cada alteração o arquivo presente no servidor onde o *site* está, precisa ser substituído pela versão mais atual. Assim, é necessário enviar esses arquivos para o servidor. Instalando um servidor FTP, juntamente com o servidor HTTP (já visto na Aula 4), facilita-se a transferência desses arquivos. Aplica-se também em empresas que desenvolvem sistemas Linux que, ao disponibilizar uma nova versão do sistema, permitem que os usuários baixem a imagem do sistema através de FTP.

5.2 Instalação

A instalação de um servidor FTP é tão simples quanto a instalação do dnsmasq, pois consiste apenas em uma busca pelos nomes dos pacotes corretos no repositório da distribuição Linux e, posteriormente, em sua instalação, através do gerenciador de pacotes.

Existem atualmente vários servidores FTP para Linux, mas nesta aula, será abordado apenas o proftpd. Entretanto, a instalação e configuração de outros aplicativos de servidor FTP seguem o mesmo padrão, de forma que as configurações a serem alteradas são bastante semelhantes.

Para realizar a instalação do proftpd, basta pesquisarmos pacotes disponíveis no repositório do Linux, conforme mostra a Figura 5.1.

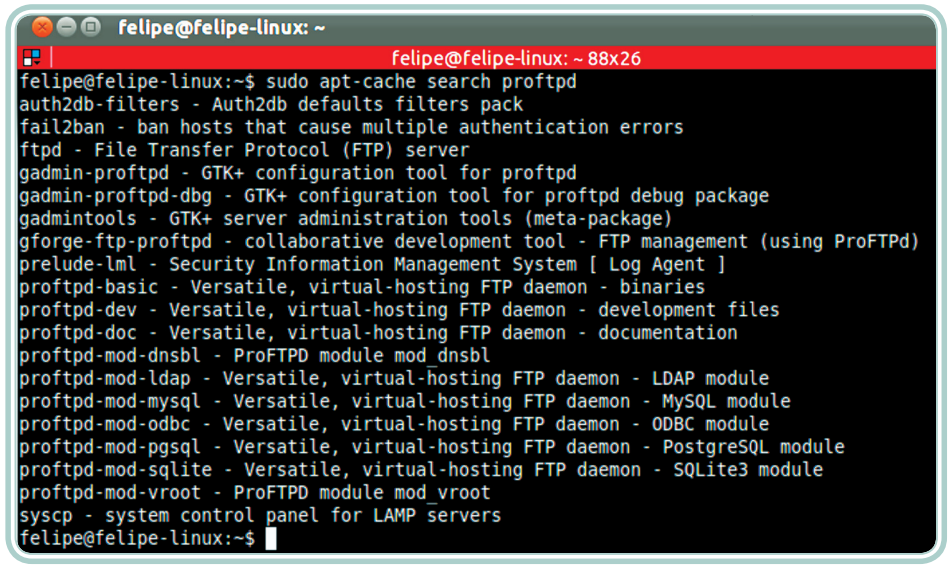


Figura 5.1: Pacotes do proftpd disponíveis no repositório do Ubuntu
Fonte: Autores

5.3 Configuração

Após realizar a instalação, é necessário alterar os parâmetros de configuração presentes no arquivo /etc/proftpd/proftpd.conf. Para isso, basta editar o arquivo com um editor de texto qualquer, através do comando sudo nano /etc/proftpd/proftpd.conf, por exemplo. O Quadro 5.1 mostra os parâmetros mais importantes e sua descrição.

Quadro 5.1: Parâmetros de configuração do proftpd

Parâmetro	Descrição
DisplayConnect	Permite a edição de um arquivo que exibirá uma mensagem personalizada, o arquivo é /usr/local/etc/proftpd.banner.
ServerIdent	Exibe ou não informações sobre o sistema no qual o servidor FTP está rodando.
Port	Porta na qual o servidor irá operar (a porta padrão é a 21).
MaxClients	Número máximo de clientes conectados simultaneamente.
MaxClientsPerHost	Número máximo de clientes por computador conectados simultaneamente.
DefaultServer	Indica se o servidor FTP é o principal ou não.
UserAlias	Usuários que terão permissão de acessar o servidor, através de *login* e senha.
ServerName	Nome do servidor.
RootLogin	Habilita ou não o uso do usuário root. Por questões de segurança, a melhor opção é não permitir o uso do usuário root.
User	Usuário no qual o servidor normalmente deve operar.
Group	Grupo no qual o servidor normalmente deve operar.
DefaultRoot	Indica se o usuário poderá sair de sua pasta pessoal depois de conectado ao servidor. Por questões de segurança, a melhor opção é não permitir que os usuários saiam de sua pasta pessoal. Para fazer isso, basta adicionar ~ após o parâmetro.

Fonte: http://proftpd.open-source-solution.org/docs/directives/linked/by-name.html

Os usuários que terão acesso ao servidor FTP são usuários reais do sistema operacional do servidor e assim, cada um deles possui uma pasta pessoal no servidor. Portanto, cada usuário que se conectar, terá acesso apenas à sua pasta, caso a opção DefaultRoot esteja definida como ~. Ainda assim, é possível permitir acesso de usuários anônimos. Isso significa que um usuário poderia acessar o servidor sem se identificar. É comum, nesse caso, fazer com que usuários anônimos não tenham permissões para escrever na pasta onde os arquivos públicos estão. Assim eles só poderão copiar os arquivos que estão na pasta, mas não poderão apagá-los nem alterá-los.

Na instalação, o proftpd cria um usuário chamado ftp e cria a pasta pessoal para esse usuário em /home/ftp/. Da mesma forma, cada usuário que acessar o servidor autenticando-se, terá sua pasta /home/nome-do-usuario.

Para permitir acesso por usuários anônimos, é necessário, no arquivo de configuração, criar uma estrutura indicando que o acesso anônimo será permitido, e indicar a pasta na qual ficarão os arquivos de acesso público.

O Quadro 5.2 mostra o trecho do arquivo de configuração, devidamente comentado, e como realizar essa configuração.

Quadro 5.2: Estrutura para habilitar acesso anônimo

```
1.  <Anonymous /home/ftp          #indica que a pasta será /home/proftpd
2.      User ftp                  #indica o usuário a ser utilizado.
3.      Group ftp                 #indica o grupo a ser utilizado.
4.      UserAlias anonymous ftp   #permite que os usuários se conectem anonimamente com os
                                    usuários anonymous e Proftpd.
5.      MaxClients 10             #número máximo de clientes simultâneos.
6.          <Directory *>         #qualquer diretório dentro da pasta pública.
7.              <Limit WRITE>     #limitação de escrita.
8.                  DenyAll       #nega qualquer escrita.
9.              </Limit>
10.         </Directory>
11. </Anonymous>
```

Fonte: Autores

Ainda existem outros parâmetros de configuração que podem ser encontrados no manual.

Após a configuração, é necessário reiniciar o servidor, através do comando sudo /etc/init.d/proftpd restart ou sudo service proftpd restart para que as configurações tenham efeito.

Para acessar o servidor FTP, basta inserir o endereço IP do servidor em um navegador, utilizando o protocolo ftp. Ex.: ftp://127.0.0.1. Caso a porta do

servidor tenha sido alterada, é necessário informá-la da seguinte forma: ftp://127.0.0.1:2000, para o caso em que a porta é a 2000. Ao realizar isso, abrir-se-á uma janela solicitando um nome de usuário e senha, conforme mostra a Figura 5.2. Caso sejam inseridos um usuário e senha, a página exibida conterá os arquivos presentes na pasta do usuário digitado, como mostra a Figura 5.3. Para acessar como anônimo, basta inserir "anonymous" no nome de usuário e deixar a senha em branco. A Figura 5.4 mostra o conteúdo da pasta pública.

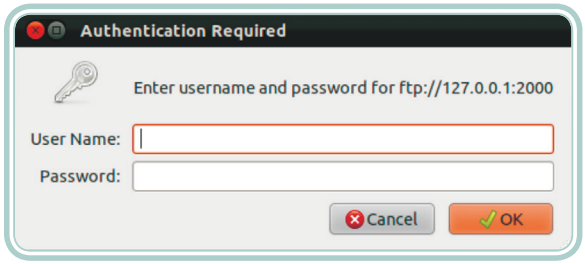


Figura 5.2: Solicitação de usuário e senha
Fonte: Autores

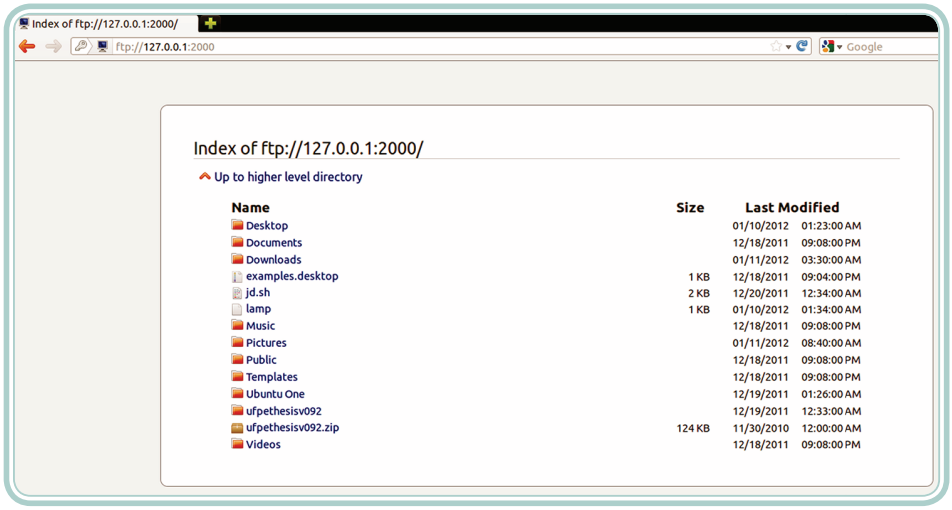


Figura 5.3: Acesso de um usuário com autenticação
Fonte: Autores

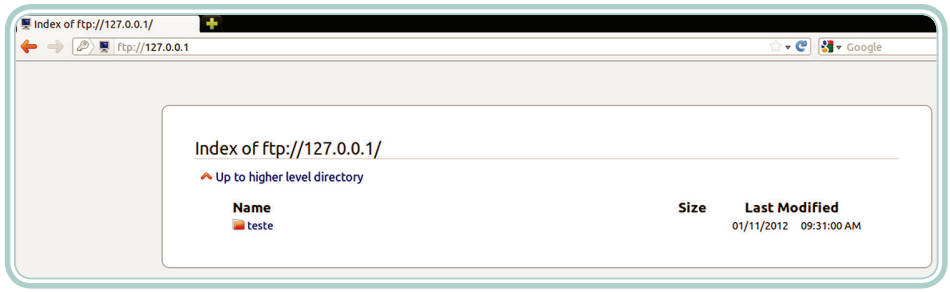


Figura 5.4: Acesso anônimo
Fonte: Autores

Entretanto, através de um navegador, é possível apenas visualizar e baixar os arquivos presentes no servidor, mesmo com um usuário que tenha permissão para alterar o conteúdo de sua pasta.

Esse problema é facilmente contornado utilizando um cliente FTP como o gFTP ou Filezilla ou algum outro disponível. Existem alguns *plugins* que são extensões para os navegadores que permitem a utilização de um cliente FTP. Independentemente do cliente utilizado, ele permite que um usuário, após autenticar-se, possa transferir arquivos para o servidor e copiar os arquivos presentes nele. Caso o cliente acesse o servidor FTP através de um computador com sistema operacional Microsoft Windows, é possível acessar um servidor FTP digitando o endereço do servidor (ex.: ftp://127.0.0.1:2000) no Windows Explorer. Será solicitado um usuário e senha e, após a autenticação, será possível enviar arquivos para o servidor simplesmente copiando-os de sua origem e colando na pasta aberta pelo FTP.

Com o objetivo de melhor visualizar o funcionamento dos processos citados, algumas operações serão mostradas a seguir. A Figura 5.5 mostra o conteúdo da pasta /home/proftpd, que será a pasta pública do servidor FTP. As Figuras 5.6 e 5.7 mostram, respectivamente, o acesso ao servidor através do Microsoft Windows Explorer e o acesso através do cliente Filezilla.

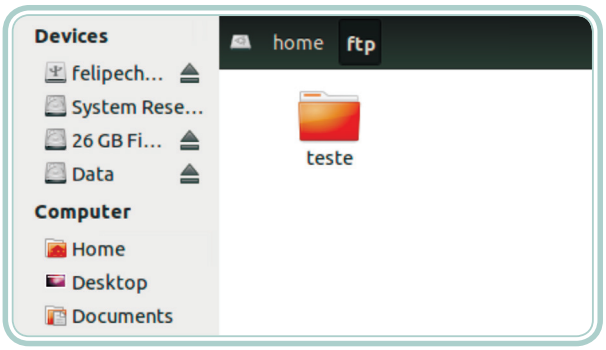


Figura 5.5: Conteúdo da pasta /home/proftpd
Fonte: Autores

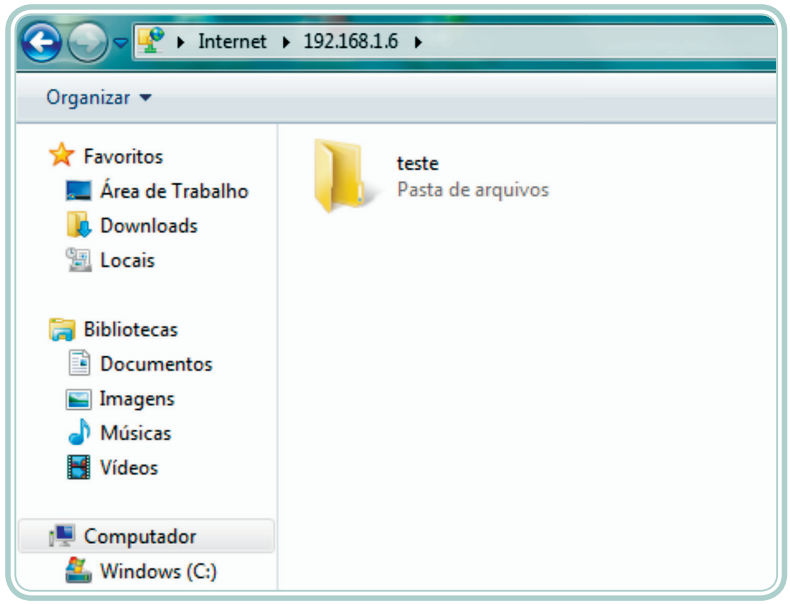


Figura 5.6: Acesso através do Microsoft Windows Explorer
Fonte: Autores

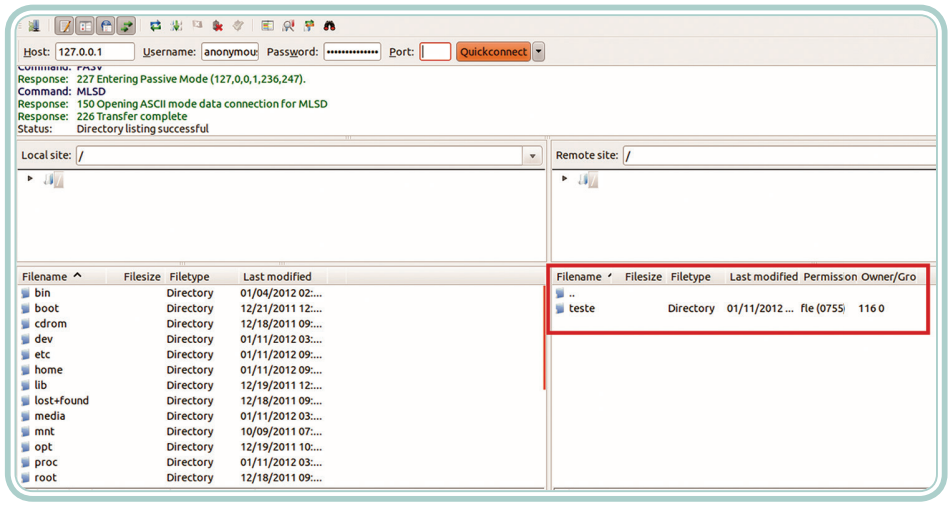


Figura 5.7: Acesso através do cliente Filezilla
Fonte: Autores

Resumo

Essa aula apresentou o conceito de servidores FTP, trazendo sua utilidade e aplicação. Com foco em uma abordagem prática, a aula trouxe um passo a passo para realizar a instalação de um servidor FTP, através de um gerenciador de pacotes de um sistema GNU/Linux Ubuntu. Após a instalação, o servidor FTP já está operacional, pois em sua instalação, o aplicativo responsável por gerenciar o serviço (proftp, no caso) traz configurações capazes de tornar o serviço operante. Porém, é necessário personalizar o serviço para se adequar às necessidades do ambiente no qual ele será instalado. Por esse motivo,

apresentou-se também um passo a passo sobre os parâmetros de configuração mais importantes do serviço. Por fim, exemplificou-se o acesso ao serviço, mostrando como ele pode ser acessado pelos clientes, deixando clara a finalidade das configurações realizadas.

Atividades de aprendizagem

1. Realize a programação de um servidor FTP, de acordo com o que foi apresentado na aula.

Aula 6 – Servidor DNS

Objetivos

Instalar, configurar e gerenciar um servidor de DNS (*Domain Name System*) e dominar conhecimentos sobre seu funcionamento e finalidade.

6.1 Considerações iniciais

Os servidores DNS estão presentes em todas as redes de computadores que possuem acesso à internet, pois são eles os responsáveis pela tradução dos nomes de endereços de *sites*. Cada vez que uma página da internet é solicitada em uma rede, o servidor DNS traduz o endereço do *site* para o IP do servidor onde ele se encontra.

Ratificando, o servidor DNS traduz os endereços digitados no navegador, como por exemplo www.w3c.org, para o endereço IP do servidor *web* no qual o *site* está acessado. É muito mais fácil decorar um endereço pelo nome do que por um endereço IP.

O servidor DNS possui uma lista com uma enorme quantidade de endereços de *sites* e seus respectivos endereços IP. Assim, quando um *site* for solicitado por um usuário, essa lista é verificada, e o endereço do *site* é acessado. Caso o servidor DNS não possua o endereço IP do *site* requisitado, ele mandará para o usuário uma página informando que não conseguiu encontrar a *website* ou mandará esta solicitação para outro servidor DNS que, por sua vez, caso não tenha, pode responder da mesma forma, informando que não conseguiu encontrar a página ou enviar a solicitação para outro servidor e mandar a resposta correta, caso o outro servidor a encontre. Esse segundo caso é chamada de busca recursiva, isto é, o primeiro servidor DNS vai buscando em outros servidores até encontrar a resposta.

Outra grande funcionalidade de um servidor DNS em uma rede é realizar melhoria na velocidade da conexão com a internet, fazendo *cache* de DNS. Isso funciona da seguinte forma: cada vez que um usuário acessa um *site*, o servidor DNS verifica se o *site* acessado está presente em uma lista. Caso esteja, o servidor vai armazenar uma série de informações desse *site* (como a

estrutura do *site*, por exemplo, que é uma informação que não muda com muita frequência) e, a partir de então, caso outro usuário acesse aquele *site*, essas informações serão carregadas do servidor DNS, diminuindo a quantidade de informações que serão acessadas via internet. Existem informações, entretanto, que não permitem o armazenamento através da *cache*. Por exemplo, um *site* de notícias deve ser carregado completamente pela internet, buscando no servidor onde o *site* está todas as suas informações. Ele pode trazer informações como notícias atualizadas que mudam com grande frequência. Porém, a estrutura do *site*, ou seja, a disposição das informações na tela raramente muda e, isso sim, pode ser armazenado na *cache* do servidor DNS.

Caso o usuário digitar o endereço de IP de um *site* no lugar de seu endereço de nome no navegador, o resultado será o mesmo.

Ainda assim, nesse caso, essas informações são poucas e não imprimem grande diferença na velocidade da conexão, dependendo da quantidade de usuários da rede. Informações como vídeos que normalmente consomem grande quantidade de banda, ao serem armazenados em uma *cache* no servidor DNS, fazem com que, um vídeo seja acessado pela primeira vez e armazenado na *cache*, se acessado posteriormente, será carregado do servidor DNS, pois o vídeo é o mesmo, não havendo necessidade de carregá-lo da internet novamente.

O servidor DNS também opera quando um *e-mail* é enviado ou uma transação FTP (*File Transfer Protocol*) é executada.

6.2 Instalação

Conforme a Aula 4, a instalação de pacotes realizada através de gerenciadores de pacotes é mais prática e fácil, pois, dessa forma, as dependências necessárias para os aplicativos que estão sendo instalados são instaladas automaticamente, trazendo melhor compatibilidade entre as versões dos pacotes possíveis.

Após uma modificação no arquivo de configuração é necessário reiniciar o serviço através do comando sudo /etc/init.d/dnsmasq restart ou sudo service dnsmasq restart.

Todo sistema operacional Linux possui uma interface de rede virtual chamada loopback (lo) que serve para indicar o próprio computador. Portanto, para acessar um recurso instalado no próprio computador, usa-se essa interface, cujo endereço IP é 127.0.0.1.

É importante lembrar que as aulas são baseadas no sistema operacional Ubuntu, que é uma distribuição Linux.

Apesar de existirem vários aplicativos capazes de fazer o trabalho de um servidor DNS, nessa aula será abordado apenas o dnsmasq, um dos mais conhecidos e utilizados atualmente.

Para a instalação, basta pesquisar pelos respectivos pacotes das aplicações no repositório de pacotes do Ubuntu, através do comando apt-cache. A Figura 6.1 mostra o resultado da busca por pacotes do dnsmasq.

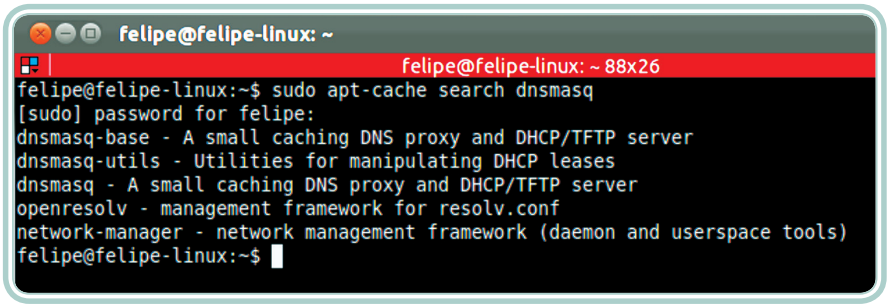


Figura 6.1: Busca por pacotes do dnsmasq
Fonte: Autores

6.3 Configuração

A princípio, o dnsmasq funciona ao ser instalado e iniciado, caso já exista um servidor DNS na rede, pois sua configuração padrão permite seu funcionamento. Entretanto, pode ser necessário alterar a configuração para atender a alguma necessidade de rede. Para isso, basta a edição do arquivo /etc/dnsmasq.conf. O Quadro 6.1 mostra algumas modificações que podem ser realizadas para personalizar o dnsmasq. O manual do dnsmasq traz mais alterações.

Quadro 6.1: Alguns parâmetros de configuração do dnsmasq

Parâmetro	Descrição
domain-needed	Esse parâmetro evita que requisições sem domínio sejam enviadas. Assim, é possível, por exemplo, acessar recursos da rede local, através do endereço IP de uma máquina da rede local.
filterwin2k	Realiza filtros a requisições que computadores com sistema Microsoft Windows 2000/XP geram automaticamente.
resolv-file	Por padrão, os endereços de IP dos outros servidores DNS que serão consultados, ficam no arquivo /etc/resolv.conf. Porém, através deste parâmetro é possível indicar outro endereço.
address	Permite definir um endereço IP fixo para um determinado endereço de *site*. Ativando essa linha, sempre que o *site* configurado for acessado, o endereço IP configurado vai ser acessado. Assim é possível, por exemplo, bloquear *sites*, colocando um endereço IP de um servidor *web* que mostre uma página, informando que aquele *site* está bloqueado.
listen-address	É o endereço IP da interface na qual dnsmasq vai monitorar. Como a configuração é realizada no próprio servidor, o endereço colocado deve ser o da interface loopback (127.0.0.1).
port	Porta do servidor DNS (a porta padrão é a 53).
dns-forwards-max	Número máximo de requisições DNS concorrentes que o servidor irá responder (padrão é 150).

Fonte: http://pragasdigitais.blogspot.com/2009/04/dnsmasq-um-servidor-dhcpdns-para.html

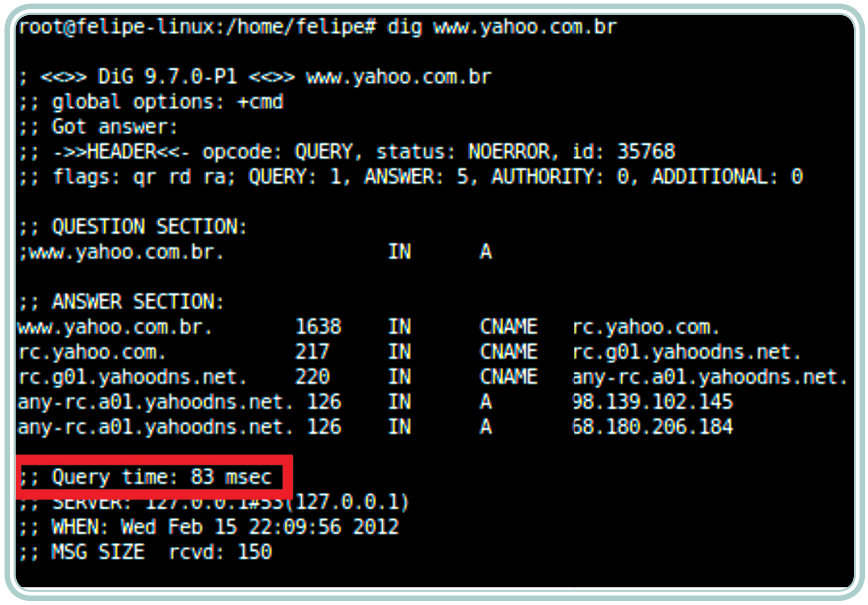
No Quadro 6.1, mencionou-se que os servidores DNS a serem consultados são inseridos, por padrão, no arquivo /etc/resolv.conf. Para inserir um servidor DNS no arquivo, basta inserir uma linha com a estrutura "nameserver endereço-ip". Portanto, é importante colocar uma linha que acrescente o endereço do próprio servidor para que ele seja consultado, inserindo uma linha contendo "nameserver 127.0.0.1".

6.4 *Cache* de DNS

Uma das grandes aplicações do servidor DNS é a economia de banda. Quando um usuário acessa uma página de internet, ele recebe informações diversas sobre essa página. Dentre essas informações, está a estrutura da página, ou seja, o posicionamento das informações na tela, da mesma forma que as cores padrões da página. Junto com isso, vêm as informações sobre o conteúdo da página que é atualizado constantemente. No caso de uma página de notícias, a estrutura seria como a notícia fica disposta na tela e vai ser igual para todas as notícias dia após dia. O conteúdo seria a notícia em si. Essa informação muda sempre que uma nova notícia é inserida no *site*. Dessa forma, sem um servidor de *cache* de DNS, sempre que um usuário acessar um *site*, todas as informações estruturais e de conteúdo serão acessadas diretamente do servidor no qual o *site* está hospedado. O uso de um servidor *cache* de DNS faz com que, quando qualquer usuário da rede acessar uma página, as informações sobre a estrutura dessa página fiquem armazenadas no servidor. Assim, na próxima vez que um usuário acessar essa página, parte das informações serão carregadas do servidor de *cache* de DNS, economizando banda. Já as informações sobre o conteúdo da página, por mudarem com certa frequência, são carregadas diretamente do servidor do *site*. Essa solução, em uma rede grande, com um número elevado de usuários é muito útil, pois usuários acessando a mesma página várias vezes ao dia provocam grande consumo de banda. Outro uso comum do servidor de *cache* de DNS é para armazenamento de arquivos e vídeos. Quando um usuário acessa um determinado vídeo em um *site*, o servidor o armazena e quando outro usuário acessar o mesmo vídeo, este já estará salvo no servidor. Outro exemplo são atualizações de sistema, onde vários arquivos são baixados de um determinado servidor para atualizar aplicativos e sistemas operacionais. Esses arquivos podem ser armazenados no *cache* do DNS e, quando solicitados novamente, são carregados.

Para configurar um servidor de *cache* de DNS, inicialmente é necessário inserir uma linha no arquivo /etc/resolv.conf. O arquivo /etc/resolv.conf contém os endereços dos servidores DNS que o computador irá utilizar. É necessário, portanto, inserir, nesse arquivo, antes de todos os outros servidores, o endereço do próprio servidor (assumindo que essa configuração está sendo realizada no próprio servidor), uma linha contendo o conteúdo "nameserver 127.0.0.1". É necessário também, inserir uma linha contendo "listen-address = 127.0.0.1" no arquivo /etc/dnsmasq.conf. As informações dessas linhas indicam que quando uma página de internet for solicitada, o servidor de DNS verificará o endereço 127.0.0.1, nele próprio, ou seja, onde as informações do *cache* estão armazenadas.

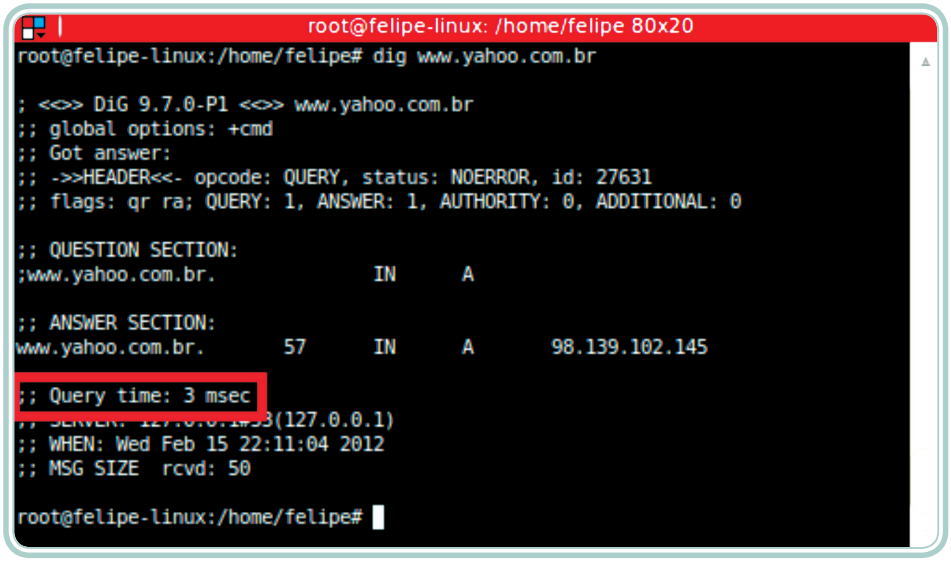
Para testar o servidor, basta utilizar o comando "dig endereço-do-site". Esse comando vai mostrar o tempo em milissegundos que foi necessário para realizar a requisição do *site* e ter sua resposta. Feito isso, basta realizar o comando novamente com o mesmo *site*, visto que na primeira execução, o *cache* de DNS armazenou parte das informações desse *site* e na segunda, essas informações foram carregadas do servidor de *cache* de DNS. A Figura 6.2 mostra a primeira execução do comando, enquanto a Figura 6.3 mostra a segunda execução.



A configuração de cache de DNS pode ser realizada em um computador que não seja servidor. Com isso o acesso a páginas de internet deve ser mais rápido.

```
root@felipe-linux:/home/felipe# dig www.yahoo.com.br

; <<>> DiG 9.7.0-P1 <<>> www.yahoo.com.br
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35768
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.yahoo.com.br.              IN      A

;; ANSWER SECTION:
www.yahoo.com.br.       1638    IN      CNAME   rc.yahoo.com.
rc.yahoo.com.           217     IN      CNAME   rc.g01.yahoodns.net.
rc.g01.yahoodns.net.    220     IN      CNAME   any-rc.a01.yahoodns.net.
any-rc.a01.yahoodns.net. 126    IN      A       98.139.102.145
any-rc.a01.yahoodns.net. 126    IN      A       68.180.206.184

;; Query time: 83 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed Feb 15 22:09:56 2012
;; MSG SIZE  rcvd: 150
```

Figura 6.2: Primeira execução do comando dig
Fonte: Autores



```
root@felipe-linux: /home/felipe 80x20
root@felipe-linux:/home/felipe# dig www.yahoo.com.br

; <<>> DiG 9.7.0-P1 <<>> www.yahoo.com.br
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27631
;; flags: qr ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.yahoo.com.br.              IN      A

;; ANSWER SECTION:
www.yahoo.com.br.       57      IN      A       98.139.102.145

;; Query time: 3 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed Feb 15 22:11:04 2012
;; MSG SIZE  rcvd: 50

root@felipe-linux:/home/felipe#
```

Figura 6.3: Segunda execução do comando dig
Fonte: Autores

É possível observar nas Figuras 6.2 e 6.3 a diferença entre o tempo da consulta. A primeira vez em que o *site* foi acessado pelo comando dig, o tempo gasto foi 83 milissegundos; na segunda, o tempo foi de apenas 6.3 milissegundos.

Resumo

Essa aula apresentou o funcionamento de um servidor DNS, trazendo seu conceito e aplicação. Para realizar uma abordagem prática, explanou-se o processo de instalação em um sistema GNU/Linux Ubuntu.

Em seguida, os parâmetros de configuração do servidor DNS foram apresentados, juntamente com um passo a passo para realizar as configurações. Por fim, foi explicada uma das aplicações de um servidor DNS, o *cache* de DNS, que tem como objetivo economizar banda da rede. O *cache* de DNS é muito útil em redes com uma grande quantidade de usuários. Por fim, foi mostrado o servidor DNS em funcionamento, quando se demonstrou que através do comando dig é visível a economia de banda em apenas uma solicitação.

Atividades de aprendizagem

1. Programe um servidor DNS de acordo com o que foi apresentado na aula.

Aula 7 – Servidor de correio eletrônico

Objetivos

Instalar e configurar um servidor de correio eletrônico e dominar conhecimentos sobre seu funcionamento.

7.1 Considerações iniciais

Servidores de correio eletrônico ou simplesmente servidores de *e-mail* (*Eletronic Mail*) têm como objetivo oferecer uma solução de correio eletrônico corporativo personalizado, possibilitando personalizar o domínio do servidor. Outro uso muito comum para servidores de correio eletrônico é a eliminação de acesso a recursos externos, aumentando a produtividade, pois os usuários não têm acesso ao correio eletrônico pessoal. Em uma rede com um grande número de usuários, através de um bom gerenciamento, o uso de um servidor de correio eletrônico economiza banda da conexão com a internet, pois os usuários irão acessar os recursos do correio eletrônico em execução em um servidor dentro da rede e não em um servidor na internet.

Em sistemas Linux, existem vários servidores de correio eletrônico que, na maioria das vezes, tornam-se bastante complexos em sua configuração. Um dos mais utilizados e abordados nesta aula é o Postfix.

7.2 Instalação

Conforme se abordou nas aulas anteriores, a instalação de pacotes realizada através de gerenciadores de pacotes é mais prática e fácil, pois dessa forma, as dependências necessárias para os aplicativos que estão sendo instalados são instaladas automaticamente, trazendo compatibilidade entre as versões dos pacotes possíveis.

É importante lembrar que as aulas são baseadas no sistema operacional Ubuntu, que é uma distribuição Linux.

A instalação do Postfix se dá da mesma maneira abordada anteriormente. A Figura 7.1 mostra os pacotes disponíveis no repositório, através do comando apt-cache.

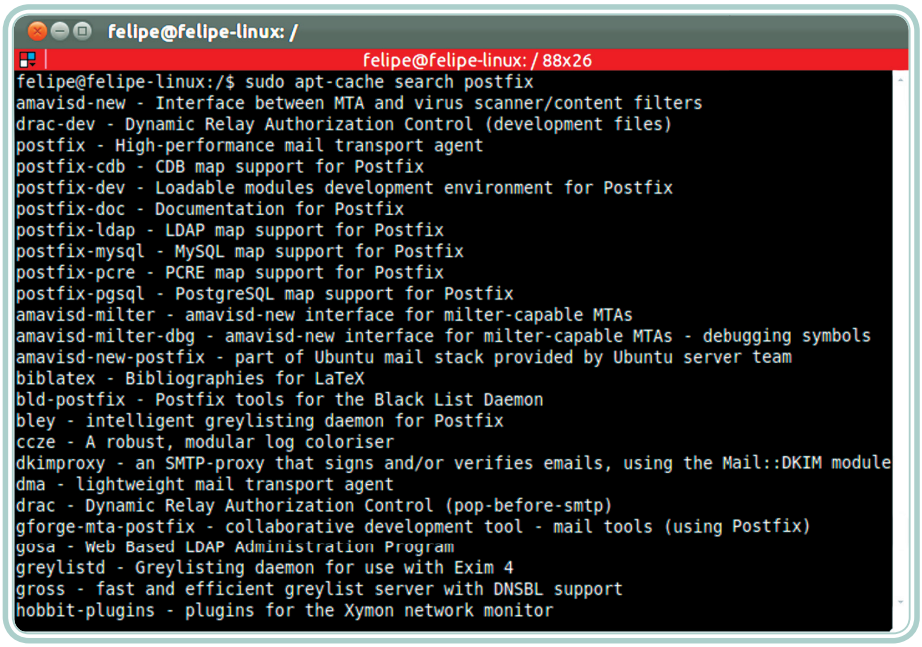


Figura 7.1: Busca de pacotes do Postfix no repositório
Fonte: Autores

É possível depreender pelas aulas anteriores que a instalação dos aplicativos é extremamente simples, bastando um comando no terminal do Linux. É importante lembrar que a instalação através do gerenciador de pacotes traz os pacotes mais estáveis e designados para aquela distribuição. Por esse motivo, a versão dos pacotes instalados pode não ser a mais atual e, caso seja necessária alguma funcionalidade presente apenas em versão mais atual, uma instalação manual é necessária. Ainda assim, na maioria dos casos, os pacotes presentes nos repositórios são suficientes para satisfazer as necessidades dos usuários.

Após uma modificação no arquivo de configuração, é necessário reiniciar o serviço através do comando sudo /etc/init.d/dnsmasq restart ou sudo service dnsmasq restart.

Todo sistema operacional Linux possui uma interface de rede virtual chamada loopback (lo) que serve para indicar o próprio computador. Portanto, para acessar um recurso instalado no próprio computador, usa-se essa interface, cujo endereço é IP 127.0.0.1.

Durante a instalação, o Postfix faz duas perguntas para o usuário. A primeira, presente na Figura 7.2, é sobre a função do servidor, se ele apenas receberá mensagens ou se também as enviará. No caso de ele apenas receber, obviamente deve haver outro servidor que realize a função de envio. Por isso, com o objetivo de evitar a instalação de outro serviço ou servidor, a opção "Internet *site*" geralmente mais usada faz todo o trabalho, tanto o de enviar, quanto o de receber mensagens.

Já a segunda pergunta (Figura 7.3) é sobre o domínio que será utilizado pelas mensagens enviadas pelo servidor. Caso exista um domínio já registrado (como se aplicou na Aula 5), basta inseri-lo. Caso não tenha um domínio registrado, basta que se defina um.

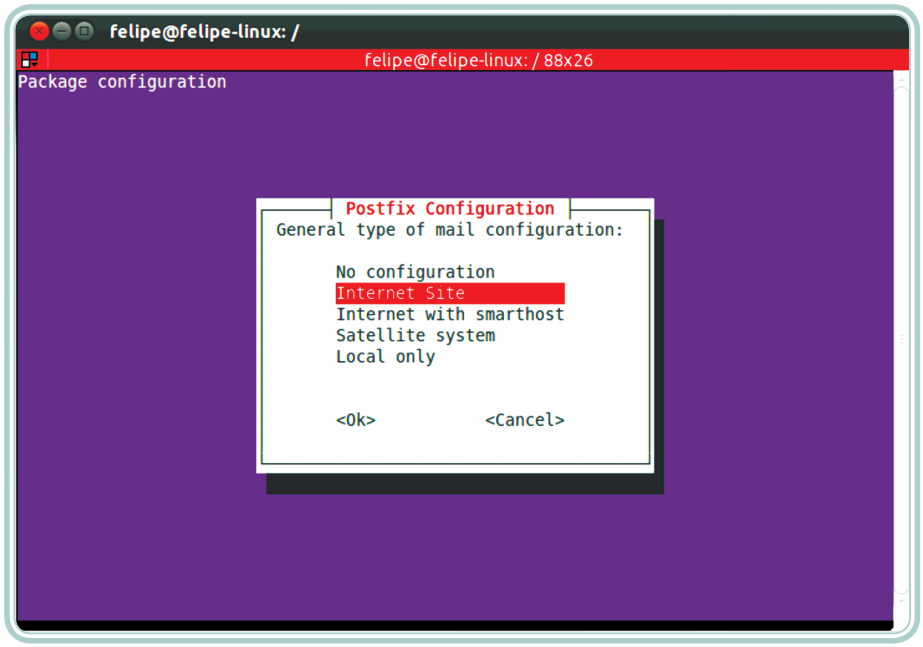


Figura 7.2: Função do servidor
Fonte: Autores

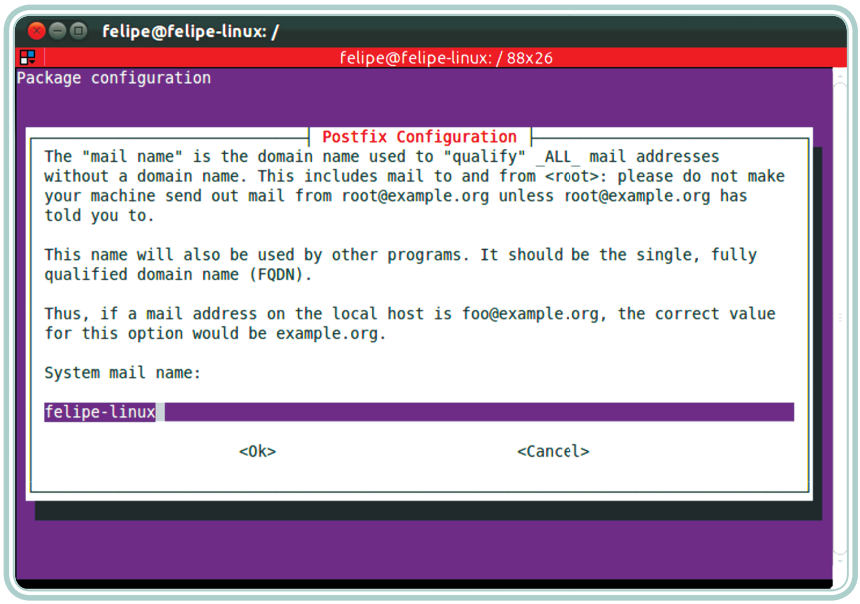


Figura 7.3: Domínio do servidor
Fonte: Autores

7.3 Configuração

Após a instalação, é necessária a configuração do Postfix, que se dá através da edição de arquivos de configuração, do mesmo modo que os aplicativos apresentados nas aulas anteriores. Os arquivos de configuração do Postfix ficam na pasta /etc/postfix/. Dentro dela existe o arquivo main.cf que é o arquivo de configuração. Para editá-lo, basta usar um editor de textos qualquer, como o Nano, por exemplo, através do comando sudo nano /etc/postfix/main.cf.

O Quadro 7.1 mostra os parâmetros de configuração e sua descrição.

Quadro 7.1: Parâmetros de configuração do Postfix

Parâmetro	Descrição
myhostname	Nome do host no domínio. Ex.: meuhost.meudominio.com
mydomain	Nome do domínio. Ex.: meudominio.com
myorigin	Deve-se comentar o parâmetro myhostname inserindo o símbolo # no início da linha e deixar mydomain descomentado. Ex.: myorigin=$mydomain. Isso significa que myorigin terá o mesmo valor que mydomain.
inet_interfaces	Define quem terá acesso ao servidor. Para permitir acesso a todos os usuários da rede, esse parâmetro deve ser definido como "all". Caso seja necessário definir os computadores que terão acesso, basta inserir os endereços IPs, separados por vírgula.
mydestination	Define os destinos com os quais o servidor pode comunicar-se. Uma forma de definir que esse servidor poderá enviar mensagens livremente dentro do domínio, é definindo-o. $myhostname, localhost.$mydomain, $mydomain, mail.$mydomail, www.$mydomain, ftp.$mydomainmydestination".
relay_domains	Determina os domínios para retransmissão. É bom definir como $mydomain.

Fonte: http://www.vivaolinux.com.br/artigo/Como-configurar-o-servidor-de-correio-eletronico-Postfix

Conforme se mencionou anteriormente, a configuração de um servidor de correio eletrônico pode tornar-se algo extremamente complexo, dependendo de cada caso. Para mais detalhes sobre os parâmetros de configuração basta consultar o manual do Postfix.

Após o término da configuração, é necessário reiniciar o serviço do Postfix. No caso deste servidor, o processo é um pouco diferente dos que se mostram nas aulas anteriores, necessitando dos seguintes comandos:

a) sudo service postfix stop (ou sudo /etc/init.d/postfix stop).

b) sudo service postfix start (ou sudo /etc/init.d/postfix start).

c) sudo service postfix reload (ou sudo /etc/init.d/postfix reload).

Para permitir a comunicação entre o servidor e os clientes, ou seja, para permitir que um cliente receba as mensagens eletrônicas é necessário executar o comando sudo /etc/init.d/inet start ou sudo service inet start para iniciar o serviço de rede do servidor.

Dentro do diretório /var/spool fica a fila de mensagens que será manipulada pelo Postfix. Como esse é um diretório do sistema, é necessário alterar as suas permissões para que o usuário Postfix (criado na instalação) possa alterá-lo. Para isso, é utilizado o comando chown, da seguinte forma: sudo chown

usuário-que-terá-permissões –R /var/spool/postfix. O diretório /var/spoll/postfix no final do comando é o diretório no qual as permissões serão aplicadas. Já o parâmetro –R significa que as permissões serão aplicadas recursivamente, ou seja, todas as pastas dentro da pasta /var/spoll/postfix receberão a nova permissão. No caso, o comando ficará: sudo chown postfix –R /var/spoll/postfix.

Para que o usuário root do sistema Linux receba mensagens com alertas do sistema, basta criar um arquivo chamado aliases na pasta /etc, com o comando sudo nano /etc/aliases. Dentro dele, basta inserir uma linha com o seguinte conteúdo: "root: administrador@dominio", onde administrador é o nome do usuário que receberá as mensagens e domínio é o domínio configurado na instalação do Postfix.

Depois basta executar o comando sudo newaliases para recarregar as informações inseridas no arquivo.

Por fim, basta inserir os usuários através do comando adduser do Linux.

É possível também, inserir os usuários em um banco de dados do MySQL.

Resumo

Essa aula trouxe a apresentação de um servidor de correio eletrônico. Servidores de correio eletrônico são bastante utilizados em empresas para direcionar ao uso de ferramentas profissionais de seus funcionários, pois a utilização de um correio eletrônico da empresa faz com que os usuários não precisem utilizar um sistema de correio eletrônico pessoal, misturando assuntos pessoais e profissionais.

Abordaram-se a instalação e a configuração de um servidor de correio eletrônico, explicando os parâmetros de configuração do serviço.

Vale ressaltar que o servidor de correio eletrônico Postfix possui uma ótima integração com outras ferramentas de rede, tornando-o assim, mais útil. Entretanto, essas integrações aumentam a complexidade de sua instalação, configuração e gerenciamento.

Atividades de aprendizagem

1. Programe um servidor de correio eletrônico de acordo com esta aula.

Aula 8 – Servidor de compartilhamento de arquivos e servidor de impressão

Objetivos

Instalar e configurar um servidor de compartilhamento de arquivos Samba, da mesma forma, conhecer sobre seu funcionamento e finalidade.

8.1 Considerações iniciais

O compartilhamento de arquivos entre computadores é muito comum em redes corporativas, principalmente em casos onde um mesmo arquivo é utilizado por uma grande parte dos usuários da rede. Um exemplo comum é o compartilhamento de instaladores de programas. Isso se dá mantendo uma pasta compartilhada em um servidor na qual os instaladores de todos os aplicativos utilizados pelos usuários da rede ficam armazenados. Assim, quando um usuário precisar de um desses instaladores, basta acessar a pasta e copiá-lo. A utilização de perfis de rede também é muito usada. Isso funciona criando um servidor controlador de domínio, colocando todos os computadores da rede nesse domínio e fazendo os usuários acessarem os computadores através de uma autenticação de usuário e senha. Em outras palavras, os usuários irão inserir nos computadores, um usuário e senha que estarão armazenados no servidor e assim poderão acessar qualquer computador da rede que esteja no domínio. Então, várias informações do usuário serão carregadas do servidor e, independentemente do computador que ele utilizar, suas configurações, históricos de navegação e outras informações pessoais serão carregadas no computador acessado. Ainda é possível, por parte dos administradores da rede, realizarem uma série de controles dos computadores da rede. Por exemplo, um usuário autenticado em um computador através de um domínio, permite que o usuário utilize qualquer aplicativo normalmente, porém impede (desde que configurado adequadamente) que ele instale ou desinstale um aplicativo ou realize operações em nível de administrador, evitando assim, que aplicativos desnecessários sejam instalados nos computadores, ocupando espaço.

No sistema operacional Linux, o servidor responsável pelo compartilhamento de arquivos mais utilizado é o Samba. Ele permite o compartilhamento de arquivos entre computadores com sistema operacionais GNU/Linux e Microsoft Windows.

Um dos grandes atrativos do Samba é a compatibilidade com uma gama de outros aplicativos, o que permite a adição de várias funcionalidades do servidor. Por exemplo, o Samba permite que outros tipos de autenticação sejam utilizados, pois pode acontecer de ele ser implementado em uma rede já existente, com um servidor de autenticação já operacional. A utilização de um novo sistema de autenticação obrigará os administradores da rede a refazerem os cadastros de todos os usuários, o que seria extremamente trabalhoso e, no caso de uma grande rede, completamente inviável. O Samba permite ainda que cada usuário tenha uma pasta no servidor que, ao autenticar-se em um computador da rede, passe a ser acessível, como se fosse uma pasta local do computador, que permite ao usuário guardar nela seus arquivos pessoais. Essa pasta ainda pode ser acessível pela rede, mediante autenticação de usuário e senha. Ainda é possível definir uma cota para cada usuário, de modo que a pasta terá um espaço disponível predeterminado pelo administrador da rede. E, devido à existência de vários níveis de usuários dentro de uma rede, usuários de diferentes níveis podem ter diferentes cotas de espaço em disco no servidor. Por exemplo, a implementação de um servidor Samba em uma rede corporativa, na qual existe um usuário, presidente da empresa e outro funcionário, permite ao presidente ter uma cota de espaço em disco maior que a cota do funcionário.

O Samba ou outro servidor que tenham o mesmo propósito em uma rede corporativa é muito comum, pois eles oferecem várias maneiras de gerenciar a rede e controlar o acesso dos usuários. Em uma rede com um servidor Microsoft Windows, o servidor que tem esse propósito chama-se Controlador de Domínio. Já em redes onde o servidor que tem essa função é um servidor Linux, ele é denominado PDC (*Personal Domain Controller*) ou Controlador de Domínio Pessoal.

8.2 Instalação

Como foi abordado nas aulas anteriores, a instalação de pacotes realizada através de gerenciadores de pacotes é mais prática e fácil, pois, dessa forma, as dependências necessárias para os aplicativos são instaladas automaticamente, trazendo assim melhor compatibilidade entre as versões dos pacotes possíveis.

É importante lembrar outra vez que as aulas são baseadas no sistema operacional Ubuntu, que é uma distribuição Linux.

A instalação do Samba ocorre da mesma forma que a abordada anteriormente. A Figura 8.1 mostra os pacotes disponíveis no repositório, através do comando apt-cache.

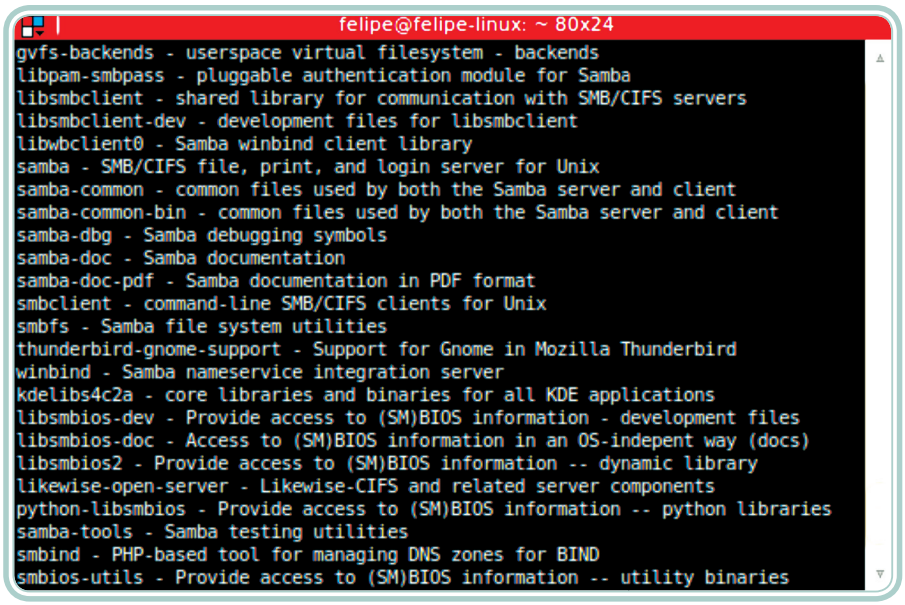


Figura 8.1: Busca de pacotes do Samba no repositório
Fonte: Autores

Após verificar os nomes dos pacotes, basta instalá-los através do comando sudo apt-get install nome-do-pacote.

8.3 Configuração

A configuração do Samba consiste na alteração de parâmetros presentes no arquivo /etc/samba/smb.conf. Para editá-lo, basta utilizar um editor de textos qualquer, como por exemplo, o Nano, através do comando sudo nano /etc/samba/smb.conf.

O arquivo de configuração do Samba possui uma estrutura especial para determinar o funcionamento de várias funcionalidades do Samba. O arquivo deve ser dividido em seções. Para indicar o início de uma seção, utiliza-se o nome da seção entre colchetes. Ex.: [global].

Após a determinação do início de uma seção, todas as configurações referentes a ela devem ser declaradas, através dos parâmetros. Ao término das configurações de todos os parâmetros da seção, inicia-se uma nova seção, declarando seu nome entre colchetes. A Figura 8.2 exemplifica a estrutura do arquivo de configuração do Samba.

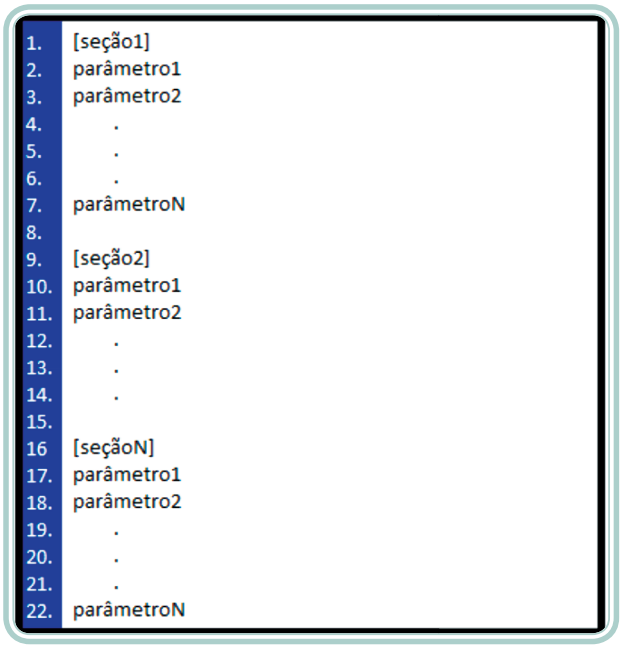


Figura 8.2: Estrutura do arquivo de configuração do Samba
Fonte: Autores

A primeira seção discutida nesta aula é a seção global que determina configurações do servidor como segurança, forma de acesso, nome, grupo de trabalho, dentre outras. Para iniciá-la, basta declará-la com [global] logo no início do arquivo. O Quadro 8.1 mostra os parâmetros mais utilizados para configurar essa seção, juntamente com suas descrições.

Quadro 8.1: Parâmetros da seção [global] do arquivo de configuração do Samba

Parâmetro	Descrição
workgroup	Grupo de trabalho.
server string	Nome do servidor na rede.
security	Método de autenticação para o acesso. Por exemplo, configurando como *user* determina que o acesso seja através de usuário e senha e configurando como *share*, o acesso será sem usuário e senha.
comment	Comentário para o servidor.

Fonte: http://www.brunorusso.eti.br/documentacao/samba_v_1_1.pdf

Nessa aula, serão descritas mais três seções bastante utilizadas, por sua grande utilidade. É importante que por meio da explicação delas, que contêm configurações de permissão de pastas, é possível criar outras seções, alterando informações julgadas necessárias.

Essas três seções são correspondentes à pasta pessoal dos usuários, uma pasta pública, que seja acessível a qualquer um, mediante usuário e senha. Nesse caso, qualquer usuário que acessar a pasta verá o mesmo conteúdo, mas não terá permissão para alterá-lo. A terceira seção é referente às impressoras.

Para criar uma seção para uma pasta pública, basta iniciar uma seção com um nome qualquer, usado para descrever o conteúdo da pasta. Na aula, a pasta será destinada a guardar instaladores de programas e, por isso, seu nome será "instaladores". Então, para iniciar a nova seção, basta inserir, após o último parâmetro da seção [global], uma linha contendo [instaladores].

O Quadro 8.2 mostra os parâmetros mais utilizados e sua descrição, para configurar a seção do exemplo de instaladores.

Quadro 8.2: Parâmetros de configuração da seção de instaladores

Parâmetro	Descrição
comment	Comentário da pasta. Ex.: Instaladores.
path	Diretório da pasta no servidor. Ex.: /home/samba/Instaladores.
public	Se esta for uma pasta visível a outros usuários da rede.
browseable	Se a pasta ficará visível na rede.
writeable	Se a pasta permitirá escrita.
read only	Se a pasta permitirá apenas leitura.
valid users	Usuários que poderão acessar a pasta.

Fonte: http://www.brunorusso.eti.br/documentacao/samba_v_1_1.pdf

É importante lembrar que a pasta descrita no parâmetro path deve ser criada, caso não exista, pois o Samba não a cria automaticamente.

A segunda seção, responsável pelo acesso à pasta pessoal de cada usuário, inicia-se com a palavra "*homes*" entre colchetes. Assim, o Samba interpreta que aquela seção determinará a configuração do acesso às pastas dos usuários. Ela deve vir após o último parâmetro de uma seção qualquer, seguindo a estrutura apresentada na Figura 8.2.

O Quadro 8.3 mostra os parâmetros e suas descrições na seção [homes].

Quadro 8.3: Parâmetros de configuração da seção referentes às pastas dos usuários

Parâmetro	Descrição
comment	Comentário da pasta.
public	Se a pasta será visível para outros usuários. Nesse caso, esse parâmetro pode ser definido como "no". Assim, somente o usuário dono da pasta poderá ver a mesma.
browseable	Se a pasta ficará visível na rede.
writeable	Se a pasta permitirá escrita.

Fonte: http://www.brunorusso.eti.br/documentacao/samba_v_1_1.pdf

As pastas dos usuários são criadas quando o usuário é criado. As pastas dos usuários ficam no diretório /home/nome-do-usuário e, por isso, não é necessário

informar o caminho, conforme foi feito na seção de Instaladores. Além do mais, para cada usuário, a pasta muda. Por exemplo, quando o usuário "pedro" acessar sua pasta pessoal, o diretório carregado pelo Samba será /home/pedro. Já quando o usuário "paulo" acessar sua pasta pessoal, a pasta carregada será /home/paulo.

A terceira e última seção é relacionada à impressão e define como compartilhar uma impressora na rede. Antes de realizar o compartilhamento, deve-se instalar a impressora no servidor, para depois poder compartilhá-la. Em sistemas operacionais Linux, o responsável por gerenciar as impressoras é o cups que, ao ser instalado, oferece uma maneira de gerenciar impressoras e impressões. A Figura 8.3 mostra os pacotes do cups disponíveis no repositório do Ubuntu.

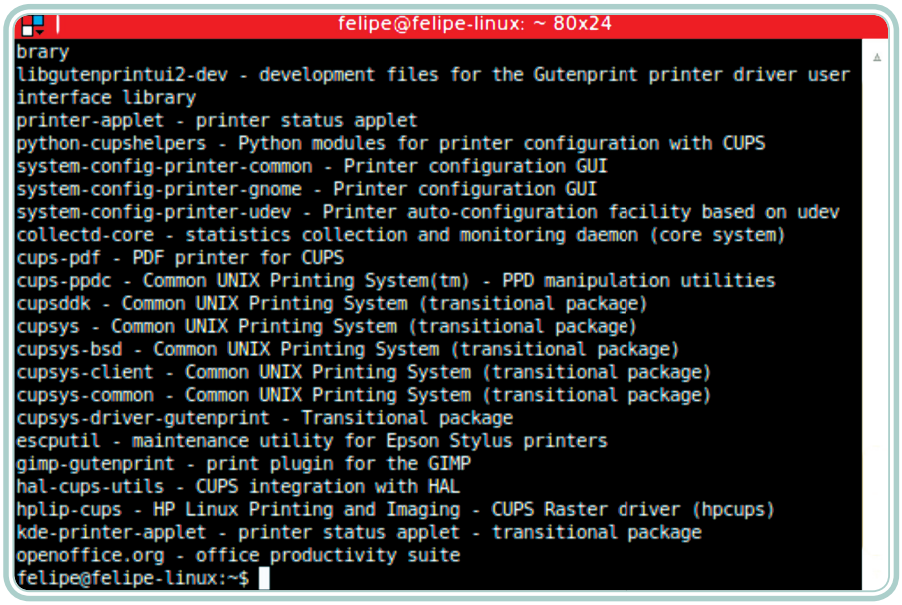


Figura 8.3: Pacotes do cups presentes no repositório
Fonte: Autores

Após a instalação é necessário configurar o cups, alterando o arquivo /etc/cups/cupsd.conf. Nesse arquivo, a seção "admin" determina, além de outras coisas, quem tem permissão de acesso às impressoras. Portanto, para permitir que usuários possam gerenciar as impressoras instaladas no servidor, essa seção deve ser alterada. A alteração consiste basicamente em inserir uma linha na qual se habilita a permissão dos computadores da rede (ou de apenas um computador) a acessarem e gerenciarem as impressoras. Essa linha deve conter a estrutura "Allow endereço-ip", onde endereço-ip é o endereço IP do(s) computador(es) que poderá(ão) acessar as impressoras. Por exemplo, para inserir o computador com endereço IP 192.168.10.10, basta inserir a linha "Allow 192.168.10.10".

É importante lembrar que após as alterações é necessário reiniciar o serviço de impressão através do comando "sudo /etc/init.d/cups restart" ou sudo service cups restart.

O cups também oferece uma interface acessível via navegador para gerenciamento de impressoras, visto que, muitas vezes, o servidor não possui interface gráfica. Assim, instalar e gerenciar impressoras fica mais fácil. Para acessar o gerenciador de impressoras do cups, basta digitar no navegador o endereço IP do servidor, com a porta 631. Por exemplo, caso o endereço IP do servidor seja 192.168.10.1, basta digitar o endereço http://192.168.10.1:631. A Figura 8.4 mostra a interface de gerenciamento do cups.

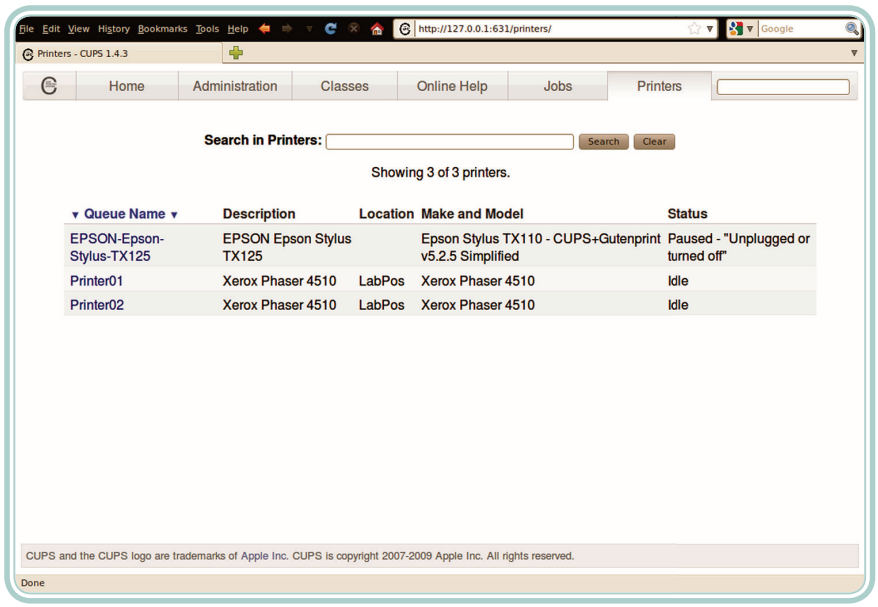


Figura 8.4: Interface de gerenciamento de impressoras do cups
Fonte: Autores

Com as impressoras devidamente instaladas no servidor, basta inserir duas linhas na seção [global] do Samba para que ele compartilhe as impressoras. O Quadro 8.4 mostra os parâmetros e sua descrição referentes a essas duas linhas.

Quadro 8.4: Inserção do compartilhamento de impressoras no Samba

Parâmetro	Descrição
printing	Indica quem vai ser o responsável por gerenciar as impressoras. No caso é o "cups". Portanto essa linha deve conter "printing cups".
load printers	Indica se as impressoras do gerenciador devem ser carregadas pelo Samba. Para carregá-las, basta definir esse parâmetro com "yes". Portanto essa linha deve conter "load printers yes" para carregar as impressoras e "load printers no" para não carregá-las.

Fonte: http://www.brunorusso.eti.br/documentacao/samba_v_1_1.pdf

Após a configuração da seção [global] do Samba, basta inserir uma nova seção referente às impressoras, com uma linha contendo [printers], e configurar seus parâmetros presentes no Quadro 8.5.

Quadro 8.5: Seção [printers]

Parâmetro	Descrição
comment	Comentário para as impressoras.
print ok	Esse parâmetro define se as impressoras estarão compartilhadas ou não. Definindo-o com "*yes*" ativa o compartilhamento e definindo-o com "*no*" desativa.
guest ok	Permite que usuários que não estejam cadastrados no Samba imprimam, caso definido com o valor "*yes*". Definindo com o valor "*no*", apenas usuários cadastrados no Samba terão permissão para imprimir.
path	Caminho das impressoras. Normalmente, elas ficam em /var/spool/samba.
browseable	Define se as impressoras serão acessíveis caso os usuários acessem o servidor de impressão pela rede. Por exemplo, como o servidor tratado nesta aula possui uma pasta compartilhada (Instaladores), os usuários podem acessar esse computador pela rede e acessar assim essa pasta. Definindo esse parâmetro com "*yes*", o servidor também mostrará que possui as impressoras instaladas. Definindo com "*no*", não mostrará. Mesmo definindo esse parâmetro como "*no*", os usuários ainda poderão utilizar as impressoras normalmente.

Fonte: http://www.brunorusso.eti.br/documentacao/samba_v_1_1.pdf

É possível também especificar esses mesmos parâmetros para cada impressora instalada no servidor. Pode haver caso onde apenas alguns usuários podem imprimir em apenas uma impressora. Para isso, basta trocar a seção [printers] pela que contenha o nome da impressora. Assim, inserindo o parâmetro "valid users", pode-se definir os usuários que utilizarão a impressora separando seus nomes por vírgula. Por exemplo, a linha "valid users = pedro, paulo", indica que apenas os usuários pedro e paulo podem utilizar as impressoras. Ainda é possível indicar os computadores que podem utilizar as impressoras através do parâmetro "hosts allow", que funciona de forma semelhante ao "valid users".

Esses parâmetros devem ser informados para todas as impressoras, inserindo uma nova seção no arquivo para cada uma.

Após a configuração de todas as seções do Samba, é necessário que o serviço seja reiniciado para que as alterações tenham efeito. A reinicialização do serviço pode ser realizada pelo comando sudo restart smbd. Por fim, basta cadastrar os usuários no Samba. Caso o usuário ainda não esteja cadastrado no sistema, basta inseri-lo normalmente através do comando adduser e, depois de criá-lo ou se o mesmo já estiver presente no sistema, basta cadastrá-lo no Samba com o comando sudo smbpasswd nome-do-usuario. Isso finaliza a configuração do servidor Samba.

Posteriormente, basta inserir os outros computadores no domínio definido no servidor Samba, para que os usuários possam acessar suas pastas armazenadas no servidor.

Resumo

Nessa aula foram apresentadas instalações e configurações de um servidor de compartilhamento de arquivos e de impressoras e a configuração dos serviços para situações comuns em redes corporativas.

O servidor de compartilhamento de arquivos é bastante utilizado pela administração da rede, pois através dele é possível acessar pastas compartilhadas que contenham aplicativos e documentos necessários para realizar manutenção na rede.

O servidor de compartilhamento de arquivos em sistemas GNU/Linux também é responsável pelo compartilhamento de impressoras. É importante deixar claro que o servidor de impressão é a ferramenta cups. É ele que manipula diretamente as impressoras, ou seja, é ele que envia os documentos para a impressora e é através dele que um documento pode ser removido da fila de impressão. O servidor Samba define apenas como vai ser realizado o acesso às impressoras.

Atividades de aprendizagem

1. Programe um servidor de compartilhamento de arquivos, de acordo com o apresentado nesta aula.

2. Programe um servidor de impressão, de acordo com o apresentado nesta aula.

Aula 9 – Ferramentas de administração de servidores e serviços

Objetivos

Instalar, configurar e utilizar as ferramentas mais utilizadas para administração local e remota de servidores *web*, bem como manipular os serviços operacionais.

9.1 Considerações iniciais

É muito comum, em servidores Linux, que o sistema instalado não possua interface gráfica, pois ela apenas significa consumo de recursos do computador, visto que os serviços instalados em um servidor podem ser instalados, configurados e gerenciados através de comandos do terminal do Linux ou utilizando algum aplicativo que também esteja disponível apenas em modo texto. Outro motivo para que os sistemas Linux de servidores não possuam interface gráfica é que assim é possível evitar que outros usuários, além do próprio administrador da rede, utilizem o computador, evitando que lhe aconteça algum problema ocasionado por mau uso.

É comum também, em grandes redes, que exista mais de um computador que tenha a função de servidor. Isso acontece quando um serviço demanda uma grande quantidade de processamento e opta-se por utilizar um computador apenas para aquele serviço. Dessa forma, pode acontecer de existirem vários servidores (inclusive com sistemas operacionais diferentes) em uma mesma rede. Normalmente sem possuírem nem monitor, devido ao grande espaço que eles ocupam. Muitas empresas costumam adotar computadores especialmente desenvolvidos para operarem como servidores, pois são potentes, ocupam pouco espaço e foram desenvolvidos com o propósito de realizarem tarefas que demandam grande poder de processamento, tendo assim, um melhor desempenho. Eles ocupam pouco espaço físico pelo fato de que podem ser organizados em estantes. A Figura 9.1 mostra um servidor da marca Dell.

Figura 9.1: Servidor Dell
Fonte: www.dell.com.br

Já a Figura 9.2 mostra uma estante com vários servidores instalados. Assim é possível colocar vários servidores em um pequeno espaço.

Figura 9.2: Estante de servidores Dell
Fonte: http://under.com.br/institucional/infra_estrutura

Visto que os servidores não possuem monitores, o acesso a eles é realizado remotamente. Isso significa que o servidor terá um serviço operante em uma determinada porta, pela qual outros computadores poderão acessar o servidor e gerenciá-lo de várias formas, dependendo do sistema operacional instalado no servidor.

Em servidores com sistema operacional Linux, geralmente instala-se um servidor SSH (*Secure Shell*). Através dele é possível ter acesso ao terminal do sistema do

servidor e executar qualquer comando como se o usuário estivesse utilizando o servidor *in loco*. Essa prática é muito comum, pois através do SSH, o servidor fica acessível a toda rede, desde que os usuários tenham permissão para acessá-lo. Alguns aplicativos ainda oferecem uma interface gráfica que permite sua administração remotamente. Essa interface gráfica pode ser acessível através de um navegador, conforme o que se viu nas aulas anteriores ou pode existir um aplicativo específico que permita o acesso ao serviço.

Esta aula apresentará como fazer a instalação do serviço de SSH no servidor e como acessá-lo remotamente. Geralmente, o acesso ao servidor se dá através de uma ferramenta de configuração e gerenciamento que um serviço ofereça, seja ela via *browser* ou terminal de comandos.

Esta aula ainda mostrará como acessar um servidor de banco de dados MySQL através do aplicativo MySQL Administrator, que tem como objetivo oferecer uma alternativa ao PHPMyAdmin para o gerenciamento do banco de dados.

9.2 Instalação

Os serviços que oferecem interfaces gráficas acessíveis via navegador já instalam essas interfaces automaticamente, na instalação do próprio aplicativo. Já ferramentas específicas devem ser instaladas manualmente. O mesmo acontece com o acesso via SSH, que necessita da instalação de um aplicativo no servidor que ficará monitorando uma porta, aguardando por conexões. Um aplicativo semelhante ao SSH é o Telnet, porém o seu uso só é aconselhável caso a conexão remota seja feita em uma rede muito segura, de preferência em uma conexão direta com o servidor, pois o Telnet não oferece nenhum mecanismo de segurança durante a conexão, fazendo com que seja possível para um atacante obter informações de usuários e senhas da conexão. Já o SSH oferece uma conexão criptografada, permitindo que usuários utilizem a conexão com a internet para acessar o servidor com segurança.

Primeiramente, será abordada a instalação do servidor SSH e, posteriormente, a instalação do cliente de acesso ao banco de dados MySQL (MySQL Administrator).

A Figura 9.3 mostra os pacotes do servidor SSH disponíveis no repositório, através do comando apt-cache.

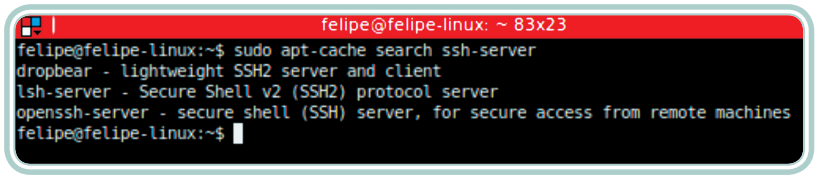


Figura 9.3: Busca de pacotes do servidor SSH no repositório
Fonte: Autores

Após verificar os nomes dos pacotes, basta instalá-los através do comando sudo apt-get install nome-do-pacote.

O MySQL Admininstrator é uma ferramenta presente no MySQL Workbench, que é um pacote de ferramentas gratuitas, tanto para sistemas Microsoft Windows quanto para sistemas Linux, que tem como objetivo oferecer meios gráficos para gerenciamentos do banco de dados. É possível obtê-lo através do *website* www.mysql.com na seção de ferramentas GUI (*Graphical User Interface*).

9.3 Configuração

A configuração do servidor SSH consiste na alteração de parâmetros do arquivo de configuração /etc/ssh/ssh_config, que pode ser realizada editando o arquivo com qualquer editor de textos. Dentre os vários parâmetros presentes no arquivo padrão, dois são considerados mais importantes, pois são mais úteis à personalização do serviço. O Quadro 9.1 mostra esses parâmetros e sua descrição.

Quadro 9.1: Parâmetros de configuração do servidor SSH

Parâmetro	Descrição
port	Porta na qual o serviço irá operar. A porta padrão é a 22.
permitrootlogin	Indica se o usuário root poderá acessar o servidor via SSH. Caso essa opção seja definida como "no", e o usuário queira acesso root ao servidor, basta acessar com um usuário qualquer e, depois de conectado, executar os comandos utilizando o sudo.

Fonte: http://www.openbsd.org/cgi-bin/man.cgi?query=sshd_config&sektion=5

Após a alteração dos parâmetros é necessário reiniciar o serviço do SSH, através do comando sudo /etc/init.d/sshd restart ou sudo service sshd restart.

Depois de ter realizado a configuração, basta acessar o servidor utilizando o comando SSH nome-de-usuario@ip-do-servidor. Por exemplo, caso o nome do usuário seja pedro e o endereço de IP do servidor seja 192.168.1.10, o acesso se dá através do comando SSH pedro@192.168.1.10. Dessa forma, o acesso será realizado pela porta padrão. Caso o número da porta tenha

sido alterado no parâmetro de configuração, ele deve ser definido no acesso, através do parâmetro "-P numero-da-porta". Por exemplo, caso a porta definida tenha sido a 2000, o comando ficaria SSH pedro@192.168.1.10 –P 2000. Para acessar com outro usuário, basta criá-lo no servidor.

Já o acesso ao servidor através de um aplicativo específico vai depender de cada aplicativo, pois o acesso é definido nele e normalmente diferem um do outro. Nesta aula, será apresentado o acesso a um servidor de banco de dados MySQL apenas para exemplificar o processo. Ao abrir o MySQL Workbench, o usuário irá encontrar várias maneiras de gerenciar o banco de dados. A Figura 9.4 mostra a interface do MySQL Workbench no sistema Linux.

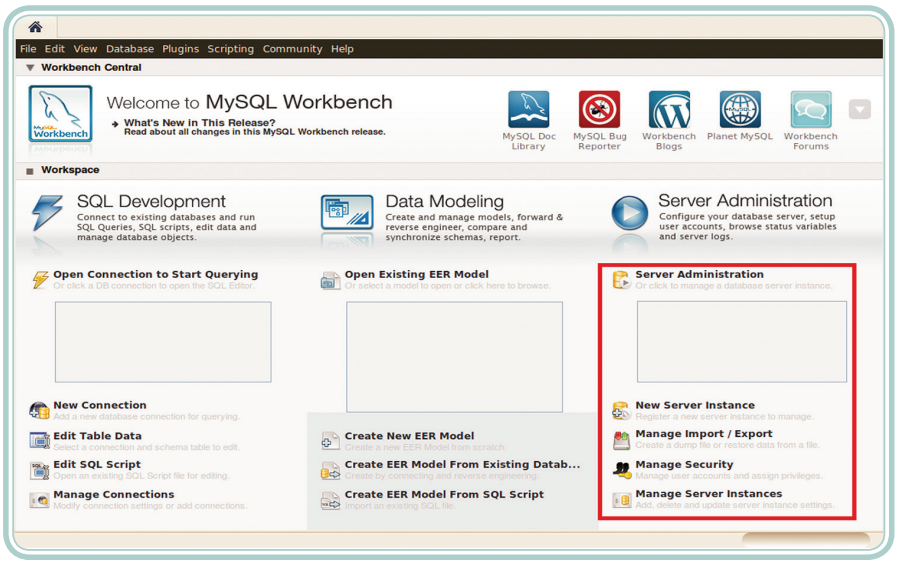


Figura 9.4: Tela inicial do MySQL Workbench
Fonte: Autores

Na Figura 9.4, é possível notar na parte destacada em vermelho, uma seção correspondente à administração do servidor, chamada Server Administration. Através dela é possível administrar o servidor. Clicando em New Server Instance, aparecerá um assistente que ajudará o usuário a configurar uma conexão com o banco de dados.

A Figura 9.5 mostra o MySQL Workbench já conectado ao servidor, onde no menu à esquerda, existem várias ferramentas de administração do servidor.

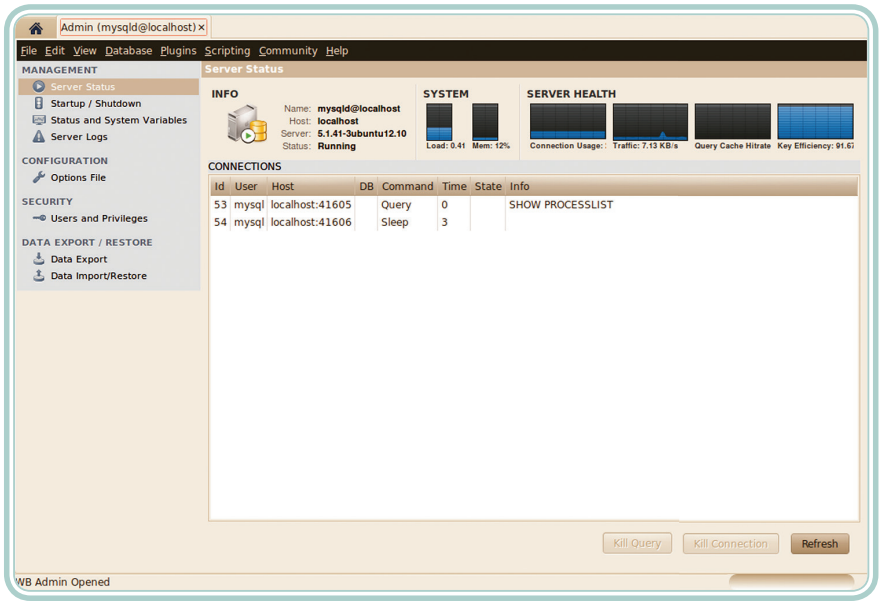


Figura 9.5: MySQL Workbench já conectado no servidor
Fonte: Autores

Resumo

Esta aula apresentou a instalação e configuração de um servidor SSH, uma ferramenta de grande auxílio para gerenciamento remoto de servidores.

Mostrou, também, um exemplo de aplicação que oferece acesso remoto a um serviço presente no servidor. Nesse último caso, a ferramenta tem o propósito de gerenciar o aplicativo em questão (MySQL, no caso). Para gerenciar o serviço, o acesso SSH é suficiente.

O serviço de acesso via SSH é muito importante em servidores, pois através dele o administrador é capaz de realizar qualquer comando no sistema, como se estivesse operando o servidor in loco. É possível iniciar, reiniciar e parar serviços, da mesma forma que alterar arquivos de configuração.

Já em casos específicos, podem existir ferramentas desenvolvidas especialmente para gerenciar um tipo de serviço apenas, como é o caso do MySQL Administrator. Através dele é possível gerenciar o serviço do servidor MySQL, da mesma forma que gerenciar as tabelas presentes no banco de dados, realizar consultas, dentre outras funcionalidades.

Atividades de aprendizagem

1. Instale as ferramentas apresentadas nesta aula e faça com que elas se comuniquem com seus respectivos serviços, caso houver.

Referências

APACHE SOFTWARE FOUNDATION. **Apache HTTP server version 2.2 documentation**. Disponível em: <http://httpd.apache.org/docs/2.2/>. Acesso em: 12 dez. 2011.

BRAIN, Marshall. **Como funcionam os servidores da web**. Disponível em: <http://informatica.hsw.uol.com.br/servidores-da-web9.htm>. Acesso em: 22 dez. 2011.

COPETTI, Rodrigo. **Instalar proftpd básico no Ubuntu 8.10 intrepid ibex**. Disponível em: <http://wiki.ubuntu-br.org/proftpd>. Acesso em: 13 dez. 2011.

DNSMASQ. **dnsmasq manual**. Disponível em: <http://thekelleys.org.uk/dnsmasq/docs/dnsmasq-man.html>. Acesso em: 2 dez. 2011.

FERREIRA, Johnny. **Configurando um servidor Samba no Ubuntu server 8.10**. Disponível em: <http://www.vivaolinux.com.br/dica/Configurando-um-servidor-Samba-no-Ubuntu-Server-8.10>. Acesso em: 20 jan. 2012.

FILETO, Renato. **Sistemas cliente-servidor**. Disponível em: <http://www.inf.ufsc.br/~fileto/Disciplinas/BD-Avancado/Aulas/03-ClienteServidor.pdf>. Acesso em: 14 jan. 2012.

Fundamentos da arquitetura cliente/servidor. Disponível em: <http://www.unlu.edu.ar/~tyr/tyr/TYR-anterior/Fundamentos_da_%20Arquitetura_Cliente-Servidor.pdf>. Acesso em: 26 jan. 2012.

GONDIM, André. **FTP no Ubuntu em dez passos**. Disponível em: <http://andregondim.eti.br/ubuntu/ftp-no-ubuntu-em-de-passos/>. Acesso em: 6 dez. 2011.

KANASHIRO, Eduardo. **Planejamento a migração de aplicações comerciais para o Linux/GNU**. Disponível em: <http://www.vivaolinux.com.br/artigo/Planejando-a-migracao-de-aplicacoes-comerciais-para-o-Linux-GNU?pagina=2>. Acesso em: 21 jan. 2012.

Local DNS cache for faster browsing on Ubuntu machine. Disponível em: <http://www.ubuntugeek.com/local-dns-cache-for-faster-browsing-on-ubuntu-machine.html>. Acesso em: 7 fev. 2011.

MORIMOTO, Carlos E. **Samba – Parte 4**: compartilhando impressoras no Samba. Disponível em: <http://www.hardware.com.br/tutoriais/impressoras-samba/>. Acesso em: 27 jan. 2012.

_____. **UDP**. Disponível em: <http://www.hardware.com.br/termos/udp>. Acesso em: 18 jan. 2012.

_____. **Instalando servidores Debian e Ubuntu**. Disponível em: <http://www.hardware.com.br/tutoriais/servidores-debian-ubuntu/pagina4.html>. Acesso em: 25 jan. 2012.

MOSMANN, Arlei. **Como configurar o servidor de correio eletrônico postfix**. Disponível em: <http://www.vivaolinux.com.br/artigo/Como-configurar-o-servidor-de-correio-eletronico-Postfix>. Acesso em: 21 dez. 2011.

MUNHOZ, Felipe. **Instalando o servidor de e-mail postfix no Ubuntu server**. Disponível em: <http://blog.felipemunhoz.com/instalando-o-servidor-de-email-postfix-no-ubuntu-server/>. Acesso em: 20 dez. 2011.

MySQL. **MySQL 5.6 reference manual**. Disponível em: <http://dev.mysql.com/doc/refman/5.6/en/index.html>. Acesso em: 8 dez. 2011.

PHP. **Manual do PHP**. Disponível em: <http://www.php.net/manual/pt_BR/>. Acesso em: 19 dez. 2011.

PIRES, Fabiano. **Dnsmasq – Um servidor DHCP/DNS para pequenas e médias redes**. Disponível em: <http://pragasdigitais.blogspot.com/2009/04/dnsmasq-um-servidor-dhcpdns-para.html>. Acesso em: 16 dez. 2011.

POSTFIX. **Postfix basic configuration**. Disponível em: <http://www.postfix.org/BASIC_CONFIGURATION_README.html>. Acesso em: 10 dez. 2011.

RUSSO, Bruno T. **Configurando o Samba**. Disponível em: <http://www.brunorusso.eti.br/documentacao/samba_v_1_1.pdf>. Acesso em: 13 jan. 2012.

SILVA, Danilo Rodrigues da. **Evolução da arquitetura cliente/servidor**. Disponível em: <http://knol.google.com/k/evolu%C3%A7%C3%A3o-da-arquitetura-cliente-servidor>. Acesso em: 20 jan. 2012.

TORRES, Gabriel; LIMA, Cássio. **Como o protocolo TCP/IP funciona – Parte 1**. Disponível em: <http://www.clubedohardware.com.br/artigos/1351>. Acesso em: 24 jan. 2012.

UBUNTU. **Ubuntu server guide**. Disponível em: <https://help.ubuntu.com/11.10/serverguide/C/serverguide.pdf>. Acesso em: 4 dez. 2011.

Currículo do professor-autor

Dener Didoné possui graduação em Ciência da Computação pela Universidade do Estado de Mato Grosso (2009) e mestrado em Ciências da Computação pela Universidade Federal de Pernambuco (2011). Tem experiência na área de Ciência da Computação, com ênfase em Segurança, especificamente na área de Perícia Computacional. Atuou como professor de diversas disciplinas relacionadas à computação pelas Faculdades Unidas do Vale do Araguaia – Univar. Atualmente é funcionário do Banco do Brasil, consultor de TI e *freelancer* como *designer* (comunicação visual impressa e digital), criação de identidades visuais, diagramação e criação de conteúdo para a internet.

Felipe José Chaulet possui graduação em Ciência da Computação pela Universidade do Estado de Mato Grosso (2009). Tem experiência na área de Ciência da Computação, com ênfase em Segurança e Administração de Redes e Administração de Sistemas e Servidores. Atuou como administrador de sistemas no projeto Brazil-IP. Atualmente é administrador de sistemas na SiliconReef.