

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RN
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES**

LDAP – LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL

LDAP – LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL

Monografia apresentada ao curso de
Especialização em Redes de Computadores
da Faculdade Natalense para o
Desenvolvimento do Rio Grande do Norte
como requisito parcial para a obtenção do
título de Bacharel em
Computação

Alexandre Magnos Gadelha Guedes

Orientador

Ricardo Weber Martins

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RN
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES**

LDAP - *LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL*

ALEXANDRE MAGNOS GADELHA GUEDES

Monografia apresentada ao curso de Especialização em Redes de Computadores da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte como requisito parcial para a obtenção do título de Especialista em Redes de Computadores.

Orientador:

Ricardo Kleber Martins Galvão

RB-30.276

FARN

BIBLIOTECA

Nº Reg M000741

14/12/07

004.72

G957f

0007

Catálogo na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Guedes, Alexandre Magnos Gadelha

Lightweight Directory Access Protocol / Alexandre Magnos Gadelha Guedes.

– Natal: FARN, 2007.

41f.

Orientador: Ricardo Kleber Martins Galvão (M. Sc.)

Monografia (Especialização em Redes de Computadores) – Faculdade
Natalense para o Desenvolvimento do Rio Grande do Norte.

1. Redes de Computadores – Monografia. 2. Protocolo LDAP – Monografia.
3. Administração de Redes – Monografia. I. Título.

RN/FARN/BC

CDU 004.72

LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RN
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES

LDAP - LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL

ALEXANDRE MAGNOS GADELHA GUEDES

Monografia apresentada e aprovada em 19 de Outubro de 2007, pela banca examinadora composta pelos seguintes membros

Ricardo Kleber Martins Galvão, M.Sc.
Orientador

Aluizio Ferreira da Rocha Neto, M.Sc.
Examinador

NATAL/RN
2007

AGRADECIMENTOS

Palavras-chave:

LDAP, CONCEITOS E FUNCIONALIDADES, DIRETÓRIO

A Deus pela saúde, coragem e inteligência para enfrentar caminhos tortuosos. A senhora Jarnete Gadelha de Lima Guedes, minha mãe e ao Sr Wellington Barbosa Guedes, meu pai, que sempre me forneceram estrutura familiar para que eu conseguisse alcançar meus objetivos. Agradeço a minha querida esposa, Karina Machado Galvão Gadelha Guedes, que sempre acreditou em mim e me dá sustentação para juntos buscarmos a nossa realização profissional e prosperidade pessoal.

RESUMO

A presente monografia pretende enfocar o estudo do protocolo LDAP onde descreve-se seus conceitos e analisa-se suas funcionalidades fazendo uma comparação com uma estrutura de diretório qualquer. Mostra também a instalação e configuração desse protocolo elaborado para lidar com sistemas hierarquizados.

Key-words:

Palavras-chave:

LDAP, CONCEITOS E FUNCIONALIDADES, DIRETÓRIO

ABSTRACT

The present monograph intends to focus the study of protocol LDAP where it describes its concepts and it analyzes its functionalities making a comparison with a directory structure any. It also shows to the installation and configuration of this elaborated protocol to deal with hierarchy systems.

Key-words:

LDAP, CONCEPTS AND FUNCTIONALITIES, DIRECTORY

SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	OBJETIVOS	12
1.1.1	Objetivos Específicos.....	13
2	REFERENCIAL TEÓRICO	14
2.1	LDAP	14
2.1.1	Desenho Genérico	14
2.1.2	Simplicidade do Protocolo	14
2.1.3	Arquitetura Distribuída.....	15
2.1.4	Segurança.....	15
2.1.5	Padrão Aberto	15
2.1.6	Funcionalidades dos Esquemas do Servidor.....	16
2.2	LDAP x DIRETÓRIO.....	16
2.2.1	Diretório.....	16
2.2.2	Serviço de Diretório	17
2.3	PORQUE USAR O LDAP	18
2.4	MODELO DE DADOS	19
2.4.1	Atributos	20
2.4.2	OBJECTCLASS	22
2.5	PREENCHIMENTO DA BASE	23
2.6	NAVEGAÇÃO NOS DIRETÓRIOS	24
2.7	LDAP V2 x LDAP V3.....	25
2.7.1	Métodos de Autenticação	25
2.7.2	Outras Vantagens	26
2.8	ESQUEMA NO LDAP.....	26
2.8.1	Visão Geral do Esquema LDAP.....	26
3	INSTALAÇÃO E CONFIGURAÇÃO DO LDAP	27
3.1	CONFIGURAÇÃO	27
3.1.1	Parâmetros do OpenLDAP.....	28
4	MANIPULAÇÃO DO DIRETÓRIO.....	32
4.1	TESTE DO SERVIÇO	32
4.2	OPERAÇÕES BÁSICAS.....	32

4.2.1	<i>Adição</i>	32
4.2.2	<i>Consulta</i>	35
4.2.3	<i>Modificação</i>	36
4.2.4	<i>Remoção</i>	37
4.3	RESUMO DAS OPERACOES BÁSICAS	38
4.4	BROWSERS LDAP	38
4.4.1	<i>Chicago University Ldap Browser</i>	38
4.4.2	<i>JXplorer</i>	39
5	CONCLUSÃO	40
	REFERÊNCIAS BIBLIOGRÁFICAS	41

LISTA DE FIGURAS

Figura 1: Antes e depois do LDAP.....	19
Figura 2: Modelo de Dados.....	20
Figura 3: Exemplo de um atributo.....	21
Figura 4: Exemplo de um ObjectClass.....	23
Figura 5: Diretório contido numa DIT.....	25
Figura 6: Arquivo ldap.conf.....	27
Figura 7: Diretório LDAP.....	28
Figura 8: Teste do serviço.....	32
Figura 9: Chicago University Ldap Browser (LDAP Browser/Editor, 2007).....	39
Figura 10: JXplorer (JExplorer, 2007).....	39

LDAP significa <i>Lightweight Directory Access Protocol</i> ou seja, Protocolo de Acesso a Diretório Leve	
TABELA 1: Atributos x Objectclass	22
TABELA 2: Valores do loglevel	29
TABELA 3: Parâmetros do backend bdb	30

O LDAP é a simplificação de um longo processo de desenvolvimento das aplicações com a especificação do padrão X 500 e o correspondente *Directory Access Protocol* (DAP) do fim dos anos 1980 e início dos anos 1990.

O DAP era um protocolo difícil de trabalhar e implementar, e por isso que foram desenvolvidos com a maior parte de sua funcionalidade, mas com muito menos complexidade. Esse protocolo foi criado pela IETF no início dos anos 90 e inicialmente se baseava em um diretório para a pilha de protocolos de rede OSI. O LDAP surgiu da necessidade da IETF de portar o DAP para a pilha de protocolos TCP/IP, o que levou à criação de uma nova versão do DAP.

O LDAP é uma definição de protocolo para acesso a dados de um diretório de objetos. É similar ao SQL, no sentido que é uma linguagem para trabalhar com bancos de dados sem especificar um banco de dados particular. De fato, a base de dados de referência do LDAP é quase sempre um sistema Oracle.

O uso do LDAP para fazer a interação com o banco de dados é muito simples, pois o banco de dados, devido às suposições que o protocolo faz, é a necessidade especializada de um diretório versus um banco de dados relacional padrão. Mas, como muitos são necessários para poder obter todas as funcionalidades desejadas de um diretório.

Este projeto de monografia apresenta definições sobre o protocolo LDAP e suas aplicações para integração de serviços como Postfix e Samba, facilitando assim a implementação de redes.

1.1 OBJETIVOS

O objetivo principal desta monografia é fornecer uma visão geral das aplicações na prática, explicando e mostrando como o uso do LDAP pode ser feito de forma eficiente na vida de um administrador de rede. Para isso é necessário entender para quem foi desenvolvido o LDAP, para resolver e como ele se relaciona com outras tecnologias. É necessário entender o próprio LDAP, tanto como tecnologia quanto como ferramenta.

1 INTRODUÇÃO

LDAP significa *Lightweight Directory Access Protocol*, ou seja, **Protocolo de Leve Acesso a Diretórios**. Como o nome sugere, é um protocolo leve para acessar serviços de diretório. O LDAP roda em cima do protocolo TCP/IP, ou seja, foi projetado para armazenar informações e consultas por TCP/IP, ou outras conexões de transferência.

O LDAP é a última interação de um longo processo de desenvolvimento que começou com a especificação de diretório X.500 e o correspondente *Directory Access Protocol* (DAP) do fim dos anos 1980 e início dos anos 1990.

O DAP era um protocolo difícil de trabalhar e implementar, e protocolos mais fáceis foram desenvolvidos com a maior parte de sua funcionalidade, mas com muito menos complexidade. Esse protocolo foi criado pela ITU no início dos anos 80 e implementava o acesso a um diretório para a pilha de protocolos do modelo OSI. O LDAP surgiu da necessidade da IETF de portar o DAP para a pilha de protocolos TCP/IP, tornando-o assim mais leve que o seu pai, o DAP.

O LDAP é uma definição de protocolo para acesso a bancos de dados especializados chamados diretórios. É similar ao SQL no sentido que é uma linguagem para interagir com bancos de dados sem especificar um banco de dados particular. De fato, o banco de dados de suporte ao LDAP é quase sempre um sistema Oracle.

O uso do LDAP para fazer a interação com o banco de dados cria algumas limitações para o banco de dados, devido às suposições que o protocolo faz e a necessidade especializada de um diretório versus um banco de dados relacional padrão. Mas estas limitações são necessárias para poder obter todas as funcionalidades desejadas de um diretório.

Este projeto de monografia apresenta definições sobre o protocolo LDAP e viabiliza recursos para integração de serviços como Postfix e Samba facilitando assim o trabalho de um administrador de redes.

1.1 OBJETIVOS

O objetivo principal desta monografia é transmitir conhecimentos adquiridos na pesquisa explicitando o porquê do uso e como o LDAP está se tornando uma ferramenta tão importante na vida de um administrador de redes. Para isso é necessário entender para quais problemas o LDAP foi desenvolvido para resolver e como ele o faz. Isto significa que também é necessário entender o próprio LDAP, tanto como tecnologia quanto como ferramenta.

1.1.1 Objetivos Específicos

A fim de alcançar o objetivo geral da monografia, as seguintes etapas foram realizadas:

- Estudo do protocolo LDAP, apresentando de forma detalhada as principais características desse protocolo.
- Instalação e configuração do open ldap.
- Apresentação de algumas ferramentas para visualização da estrutura de diretórios utilizando o ldap.

2.1.1 Desenvolvimento Genérico

Como o LDAP foi projetado para ser um diretorio de propósito geral, ele deve que ser extensível. Ele usa um esquema de definição orientado a classes baseado em hierarquia, que permite fácil extensão para qualquer uso razoável. Tal como um esquema flexível que o projeto de implementação de LDAP, e também alguns padrões de implementação de serviços. Hierarquia, que se que a maioria dos desenvolvedores de extensão de esquemas básicos.

2.1.2 Simplicidade do Protocolo

Um dos mais importantes aspectos do desenvolvimento do LDAP é que, por um lado, o protocolo base é baseado em vez do LDAP Directory Access Protocol, e que ele é relativamente simples, e é relativamente simples implementar e trabalhar com o Internet, isto não pode ser pelo fato que o LDAP é suportado pela maioria das linguagens de programação, incluindo C, Java e Perl, e ele é suportado por uma variedade de mais para os sistemas operacionais, incluindo o Solaris, OS/2, Linux, Microsoft Windows e o Mac OS.

2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados os conceitos necessários para a compreensão do funcionamento do LDAP, além de algumas características desse protocolo para acesso a diretórios.

2.1 LDAP

Segundo (SENA, 2007), o LDAP é um protocolo baseado no X.500 (um dos primeiros padrões de estrutura de diretórios) que implementa o acesso a um serviço de diretório definindo a forma como os dados são representados, importados e exportados.

A linguagem utilizada para a representação dos dados é a LDIF, *LDAP Data Interchange Format*, baseada na sintaxe ASN.1. (SENA, 2007)

Segundo (KANIES, 2007), o LDAP foi projetado especificamente para resolver os problemas causados pela proliferação de diretórios pela rede, e existem seis aspectos de sua implementação atual que lhe dão esta habilidade.

2.1.1 Desenho Genérico

Como o LDAP foi projetado para ser um diretório de propósito geral, ele teve que ser extensível. Ele usa um esquema de definição orientado a objeto, baseado em herança, que permite fácil extensão para qualquer uso razoável. Existe um esquema básico que é parte da especificação do LDAP, e existem outros padrões de fato para vários serviços. Entretanto, espera-se que a maioria dos desenvolvedores irá estender os esquemas básicos.

2.1.2 Simplicidade do Protocolo

Um dos mais importantes aspectos do desenvolvimento do LDAP, e que fez com que o mesmo fosse adotado em vez do DAP (Directory Access Protocol), é que ele é um protocolo simples e é relativamente simples implementar e trabalhar com o mesmo. Isto fica transparente pelo fato que o LDAP é suportado pela maior parte das linguagens de programação, incluindo C, Java e Perl, e ou é suportado ou será suportado pela maior parte dos sistemas operacionais, incluindo o Solaris, GNU/Linux, Microsoft Windows, e o Mac OS.

2.1.3 Arquitetura Distribuída

Segundo Luke A. Kanies 2001, com o uso de replicação de dados, é possível replicar todo ou parte de um diretório LDAP para locais separados fisicamente, o que permite que os dados tenham alta disponibilidade e coloca os mesmos tão próximos quanto necessário do cliente. Utilizando referências, domínio de dados de porções do diretório podem ser distribuídos em diferentes servidores LDAP, permitindo assim que partes de uma empresa ou projeto tenham controle sobre os dados necessários ao mesmo tempo que mantém uma única autoridade sobre cada parte dos dados.

2.1.4 Segurança

Ainda segundo Luke A. Kanies, um grande foco do desenvolvimento do LDAP foi a segurança, com a versão 3 do protocolo LDAP trazendo significativos melhoramentos. Existem três aspectos básicos na proteção de informação em um diretório: acesso, autenticação e autorização (AAA, ou Triplo-A).

- *Acesso* é a habilidade de conectar-se a um serviço e pode ser restringido baseado em detalhes como hora do dia ou endereço IP.
- *Autenticação* é a habilidade de provar ao serviço que um cliente é um usuário válido.
- *Autorização* é o serviço fornecendo ou negando direitos específicos ou funcionalidades ao cliente.

Para acesso seguro, o LDAP suporta o *Transport Layer Security* (TLS), que criptografa toda a comunicação entre cliente e servidor. Para autenticação, o LDAP suporta a *Simple Authentication and Security Layer* (SASL), que permite que o cliente e servidor negociem um método de autenticação (seguro).

O TLS e o SASL provêm funcionalidades criptográficas mas não o controle sobre o acesso e autenticação. O LDAP irá fornecer a habilidade de controlar todos os três aspectos da AAA através de *Access Control Lists* (ACLs) (Listas de Controle de Acesso). As ACLs podem ser usadas para autorizar o acesso baseado em muitos fatores diferentes. Elas podem ser usadas para forçar tipos específicos de autenticação e, uma vez que o cliente esteja plenamente autenticado como usuário válido, as ACLs são usadas para autorizar o usuário.

2.1.5 Padrão Aberto

Como o LDAP é um padrão aberto mantido pela IETF, ele pode ser utilizado por qualquer desenvolvedor, companhia, ou administrador sem receio de ficar aprisionado a

protocolos proprietários ou a vendedores específicos, e permite que a escolha da implementação seja baseada nos detalhes do projeto em vez de preocupações de interoperabilidade. Isto também significa que o LDAP pode progredir de acordo com as necessidades das pessoas que o utilizam, em vez de uma corporação que está concentrada nos lucros ou fatia de mercado.

2.1.6 Funcionalidades dos Esquemas do Servidor

A especificação LDAP garante que os clientes LDAP podem solicitar a lista completa de funcionalidades de esquemas de dados de qualquer servidor LDAP, permitindo desta forma que o cliente altere sua funcionalidade de acordo com a do servidor, que deverá fornecer grande interoperabilidade entre diferentes implementações e versões do LDAP.

2.2 LDAP x DIRETÓRIO

Para entender o que é LDAP precisamos primeiro ter uma boa noção sobre o que é diretório e também o que é serviço de diretório. Como no caso de SMTP se você não sabe o que é email fica difícil entender SMTP.

2.2.1 Diretório

O conceito de diretório muitas vezes causa confusão. O nome diretório na literatura especializada tem vários significados, dependendo do contexto. No contexto de sistemas de arquivo ele possui um significado, no contexto de redes e ambientes distribuídos outro e no contexto de banco de dados um terceiro significado.

Esses significados não são excludentes como a maioria de nós poderia supor a princípio. Nisso surge uma dúvida, porque o verbete “diretório” é usado nesses três contextos? Bem, em um nível mais elementar diretório significa “lista”. E lista nada mais é do que depósitos de informação. A partir daí podemos entender porque diretório é usado nesses contextos.

- **Sistemas de arquivo:** Diretório nada mais é do que um arquivo especial que contem as *listas* dos arquivos pertencentes a esse diretório.
- **Redes e ambientes distribuídos:** Diretório é uma lista que contem informações, quase todos serviços de rede, por exemplo, exigem algum tipo de autenticação, obrigando desta forma que os serviços mantenham um diretório de usuários (uma lista de usuários).

- **Banco de dados:** Diretório é uma lista que na verdade é um depósito de informação.

Quando se fala sobre LDAP, diretórios, etc, a maioria dos profissionais de informática logo acham que se trata de servidor de arquivos, sistemas de arquivo, ou algo a respeito, são tomados como sinônimos.

Basicamente o diretório é uma base de dados especializada com o propósito de prover o acesso rápido aos dados de uma maneira padronizada

2.2.2 Serviço de Diretório

Um serviço do diretório é uma aplicação que controla os objetos e seus atributos em um diretório. Com o serviço do diretório, os objetos e os atributos podem estar disponíveis aos **usuários** e a **outras aplicações**.

Existem muitos servidores de diretórios hoje em dia, por exemplo o famoso DNS é um serviço de diretório, outro famoso serviço de diretório é o NIS.

Serviço de diretório é a implementação cliente/servidor para o conceito de diretório, sendo que podemos ter diretórios sem o serviço de diretório. É o caso do livro de endereços dos clientes de email. São diretórios locais. Também é o caso da autenticação por exemplo em sistemas linux, o /etc/passwd, que nada mais é que diretório (depósito de informações) sobre usuários.

Nos dias atuais existe uma grande necessidade de acessar diretórios remotamente. E foram criados inúmeros diretórios e serviços de diretório para os mais diversos fins.

O LDAP é um protocolo para um serviço do tipo grave-uma-vez-leia-várias-vezes, sendo que a otimização no acesso aos dados não faz parte da responsabilidade do LDAP. Ou seja, esse protocolo supõe que a leitura dos dados na sua base de dados seja bem maior que sua gravação ou modificação. Isso o diferencia dos bancos de dados existentes onde a forma de interação com os dados tem que ser feita da melhor maneira possível tanto na leitura quanto na gravação ou modificação.

O LDAP utiliza uma representação própria que normalmente não é equivalente à estrutura de dados utilizada no servidor de diretório. Essa característica do LDAP é que torna desprezível a forma como ele enxerga o armazenamento feito pelos serviços de diretórios, ele apenas não leva em consideração como os dados estão sendo armazenados.

A forma de replicação no LDAP acontece de forma assíncrona, gerando inconsistências instantâneas, diferente da maioria dos SGBDs que tem a forma de replicação síncrona.

2.3 PORQUE USAR O LDAP

As aplicações devem alcançar hoje não somente recursos em própria Rede local (LAN) ou na Intranet de uma instituição, mas também recursos em outros lugares e até mesmo em territórios menos desconhecido com a grande rede mundial a Internet.

O crescimento gigantesco das redes e usuários dos últimos anos conduziu ao fato que há em sua maioria de redes diferentes, listas especializadas com a informação redundante parcial, que pode ser usada freqüentemente somente com dificuldade junto.

Por exemplo, o servidor web apache tem a capacidade de autenticação, portanto necessita de alguma forma de armazenamento para usuário. O apache utiliza uma série de programas (htpasswd, htdigest) e arquivos (htaccess) para armazenar, adicionar e alterar a sua base de dados de usuários. Que na verdade nada mais é que um diretório.

Já o servidor de arquivos Samba usa outro diretório para armazenar dados sobre usuários. Além de usar outra série de programas (smbpasswd, smbadduser) para a manutenção dessa base de dados (diretório).

Outros serviços compartilham a mesma base de dados é o caso do servidor de email postfix, do servidor FTP proftpd e do sistema de login do linux.

Além de servidores outras aplicações clientes utilizam serviços de diretórios e/ou serviços de diretório. Observa-se na figura abaixo o caso dos clientes de email.

Além disso, aplicações desenvolvidas por terceiros também necessitam de algum tipo de diretório, seja local ou remoto.

Exemplificando com um controle de estoque onde tanto as contas dos usuários desse sistema como as permissões são mais lidas do que escritas. Essa aplicação seria muito mais aceita e integrada se a autenticação fosse feita através de um serviço de diretório, eliminado o velho problema de um usuário possuir varias contas e várias senhas, e fazer vários logins em vários sistemas.

O LDAP é capaz de unificar diferentes diretórios em um único diretório, seja para servidores, ou para aplicações clientes. A Microsoft com seu Active Directory, a Novell com seu NDS, a SUM com o iPlanet. Todas essas empresas apostam no LDAP como mecanismo centralizador de informações. Uma ótima forma de reforçar a necessidade de servidor de diretórios é o antes de depois da Figura 1.

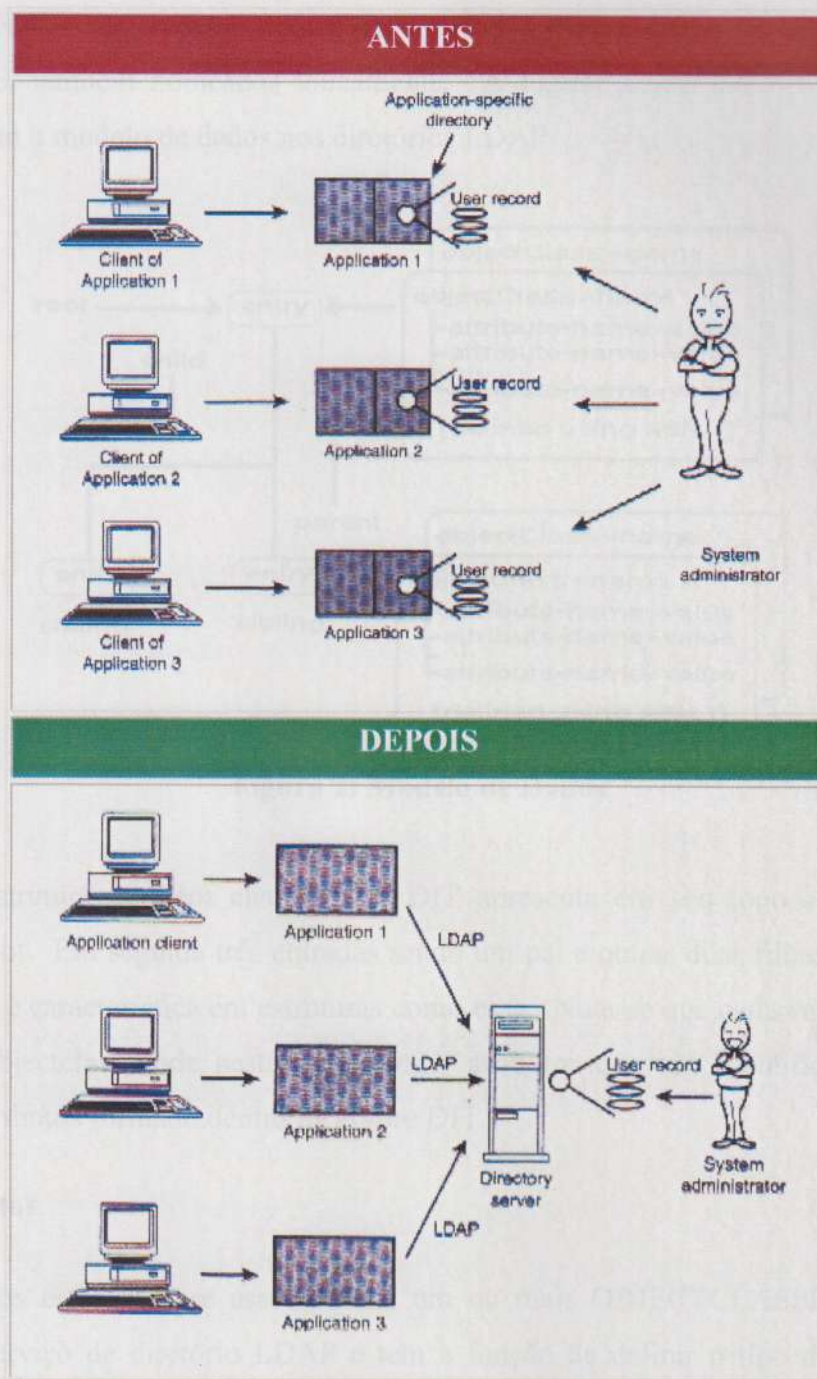


Figura 1: Antes e depois do LDAP

2.4 MODELO DE DADOS

Diretórios LDAP utilizam um modelo de dados que representam os dados como uma hierarquia de objetos, popularmente chamada de árvore.

A árvore em si é chamada de DIT, e o topo da árvore é chamada de ROOT, BASE ou SUFFIX.

Cada nó da árvore é chamado de ENTRY que é composta por uma ou mais OBJECTCLASSES que possuem nomes únicos. Cada OBJECTCLASS possui um ou mais ATTRIBUTES, também nomeados unicamente. A Figura 2 apresenta de forma sintética como é disposto o modelo de dados nos diretórios LDAP.

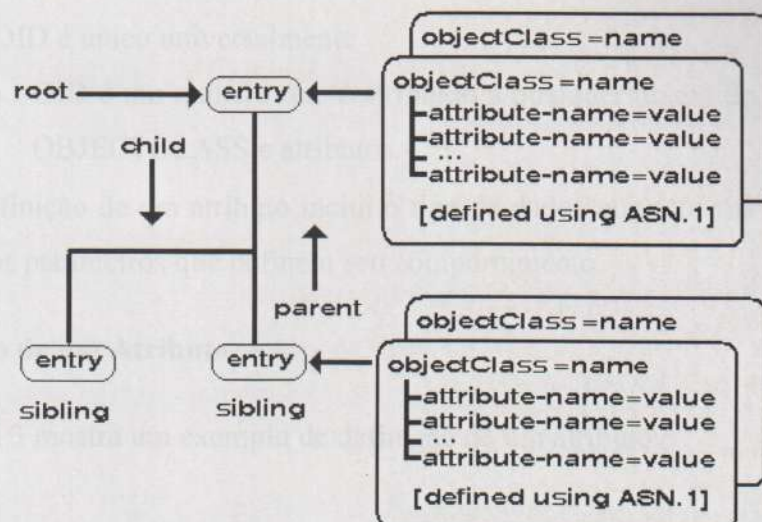


Figura 2: Modelo de Dados

Essa estrutura também chamada de DIT apresenta em seu topo a raiz da árvore chamada de root. Em seguida três entradas sendo um pai e outras duas filhas mantendo uma hierarquia que é característica em estruturas como essa. Nota-se que cada entrada contém no mínimo um objectclass onde nestas irão conter atributos que vão identificá-las como um conjunto de atributos formado dentro na árvore DIT.

2.4.1 Atributos

Atributos estão sempre associados a um ou mais OBJECTCLASSES presentes na estrutura do serviço de diretório LDAP e tem a função de definir o tipo de dados que irá conter nos objectclass. Esses atributos podem ser opcionais (mais comum) ou obrigatórios e ainda single ou multi. Opcionais quando não necessitarem esta presente na estrutura e obrigatórios quando assim o forem. Single é quando só aceitar um único registro para aquele atributo, exemplo CPF, uma pessoa só pode ter um CPF. E Multi quando poder aceitar vários registros para o atributo. Possuem nomes únicos e podem ter ainda apelidos (alias) ou abreviações.

Todo e qualquer atributo está contido em um ou mais objectclass e são escritos usando a linguagem de definição ANS.1, assim como os objectclass. Eles são hierárquicos, ou seja, um atributo filho herda características do atributo pai.

Em cada nível da árvore existem atributos especiais que são usados para nomear as entradas. Estes atributos especiais são normalmente chamados de RDN – *Relative Distinguished Name*. O RDN pode ser formado por um ou mais atributos na mesma entrada.

Algumas características adicionais de atributos estão elencadas abaixo.

- Seu nome é único universalmente
- Seu OID é único universalmente
 - OID é um numero que é atribuído a qualquer objeto do esquema incluindo OBJECTCLASS e atributos.
- A definição de um atributo inclui o tipo de dados que este vai armazenar além de outros parâmetros que definem seu comportamento.

2.4.1.1 Exemplo de um Atributo

A Figura 3 mostra um exemplo de definição de um atributo.

```
"attributetype ( 2.5.4.41 NAME 'name' EQUALITY caseIgnoreMatch SUBSTR  
caseIgnoreSubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )"
```

Figura 3: Exemplo de um atributo

O detalhamento de cada componente desse atributo é o seguinte:

- **AttributeType** - Indica o início da declaração de um atributo. Nota-se que toda a declaração está contida entre parêntesis.
- **2.5.4.41** - É o OID a que este atributo está associado. O OID deve ser único.
- **NAME 'name'** - Indica um nome para o atributo. Este nome também deve ser único. A função do "name" é tornar a leitura destes objetos mais agradável.
- **EQUALITY caseIgnoreMatch** - Indica o comportamento do atributo em uma procura. CaseIgnoreMatch informa que a procura não é case sensitive.
- **SUBSTR caseIgnoreSubstringsMatch** - Indica o comportamento do atributo em uma procura com wildcard (procura por substring).
- **SYNTAX 1.3.6.(...).121.1.15{32768}** - Indica o tipo de dado que vai ser armazenado. Nota-se que neste caso apenas o OID é informado. O valor entre chaves {32768} informa o tamanho máximo que o dado pode assumir.

2.4.2 OBJECTCLASS

Objectclass são pacotes de atributos ou conjuntos de atributos que fazem parte de uma estrutura da árvore de diretórios que são marcados (pelo próprio atributo) como sendo obrigatórios ou opcionais. Eles têm a característica de serem hierárquicos e dessa forma herdam os atributos de seus pais.

Os objectclass podem ser classificados como estrutural, usados para criar uma entrada, ou auxiliar quando fazem parte de uma entrada. É o objectclass que define se o atributo vai ser obrigatório ou opcional. Os objectclass são escritas usando a linguagem de definição ANS.1.

A tabela 2.1 simplifica o que como estão distribuídos os atributos dentro das classes que já estão pré-definidas nos esquemas que vem no pacote de instalação do ldap. Esquemas serão abordados mais adiante nesta monografia.

Abreviação	Nome	objectClass	Descrição	Schema
c	countryName	country	Código de país de 2 letras (ISO 3166)	core.schema
cn	commonName	person organizationalPerson organizationalRole groupOfNames applicationProcess applicationEntity posixAccount posixGroup		core.schema
dc	domainComponent	dcObject	Parte de um nome de domínio	core.schema
mail	rfc822Mailbox	inetOrgPerson	Endereço de email	core.schema
ou	organisationalUnitName	organizationUnit	Divisão lógica da empresa	core.schema
-	telephoneNumber	organizationalPerson		core.schema
userPassword	-	organization organizationalUnit person dmd simpleSecurityObject domain posixAccount	Senha para algum tipo de controle de	core.schema
uid	userid	accountTypeOrgPersonPosix	Informações de usuário	core.schema

TABELA 1: Atributos x Objectclass

A tabela acima mostra quais os atributos que fazem parte de quais objectclass. Um exemplo seria o atributo commonName que faz parte de vários objectclass como person, organizationalRole e groupOfNames. Observa-se também as abreviações ou apelidos (alias) que são dados aos atributos e suas descrições dentro do esquema.

Algumas características adicionais estão elencadas abaixo:

- Seu nome é único universalmente
- Seu OID é único universalmente

- OID é um numero que é atribuído a qualquer objeto do esquema incluindo objectclass e atributos.
- Pelo menos um objectclass deve esta atribuído a uma entrada

2.4.2.1 Exemplo de um Objectclass

A Figura 4 mostra um exemplo de definição de um atributo.

```
"objectclass ( 2.5.6.2 NAME 'country' SUP top STRUCTURAL MUST c MAY (
searchGuide $ description ) )"
```

Figura 4: Exemplo de um ObjectClass

O detalhamento de cada componente desse ObjectClass é o seguinte:

- **Objectclass** - Indica o início da declaração de um ObjectClass. Nota-se que toda a declaração está contida entre parêntesis.
- **2.5.6.2** - OID a que este ObjectClass está associado. O OID deve ser único.
- **NAME 'country'** - Indica um nome para o ObjectClass. Este nome também deve ser único. A função do "name" é tornar a leitura destes objetos mais agradável.
- **SUP top** - indica quem é o "pai" de "country". O objeto "top" é a raiz da hierarquia
- **STRUCTURAL** - Indica que o ObjectClass é usado na criação de Entries. Um ObjectClass também pode ser AUXILIARY.
- **MUST c** - Indica a lista de atributos MANDATORY, ou seja, obrigatórios que fazem parte de "country". Nota-se que não há parêntesis em "c", indicando que este é um atributo simples (1 ocorrência).
- **MAY (searchGuide \$ description)** - Indica a lista de atributos OPTIONAL que fazem parte de "country". Nota-se que há parêntesis entre os atributos, isto indica que eles são múltiplos (podem ter mais de uma ocorrência).

2.5 PREENCHIMENTO DA BASE

Todo o preenchimento da base LDAP é chamado de "população", ou seja, preencher essa base significa que vai se popular a base ldap. Para que ocorra a população da base é utilizado um arquivo chamado arquivo LDIF que basicamente é um arquivo texto que vai inicialmente determinar a estrutura ou esqueleto da DIT.

As linhas de códigos abaixo apresentam exemplos de arquivos LDIF que irá constituir uma estrutura ou árvore chamada de DIT.

Exemplo 1:

```
dn: dc=example,dc=com
dc: example
description: The best company in the whole world
objectClass: dcObject
objectClass: organization
o: Example, Inc.
```

Exemplo 2:

```
dn: ou=people, dc=example,dc=com
ou: people
description: All people in organisation
objectClass: inetOrg
```

Exemplo 3:

```
dn: cn=Joe Schmo,ou=people,dc=example,dc=com
objectclass: inetOrgPerson
cn: Joe Schmo
sn: Schmo
uid: jschmo
mail: joe@mydomain.com
mail: r.schmo@mydomain.com
```

2.6 NAVEGAÇÃO NOS DIRETÓRIOS

Para visualizar os dados no diretório é necessário informar o caminho completo ou parte do caminho das entradas que se deseja visualizar, este caminho é chamado de DN (Distinguished Name). Em cada nível da árvore DIT do diretório, o(s) atributos únicos que nomeiam as entradas são usados para montar o caminho. Tais atributos são chamados RDN.

A Figura 5 demonstra um exemplo de parte de um diretório contido na árvore DIT. Enfatiza a figura do RDN e dos DN mencionados no parágrafo acima.

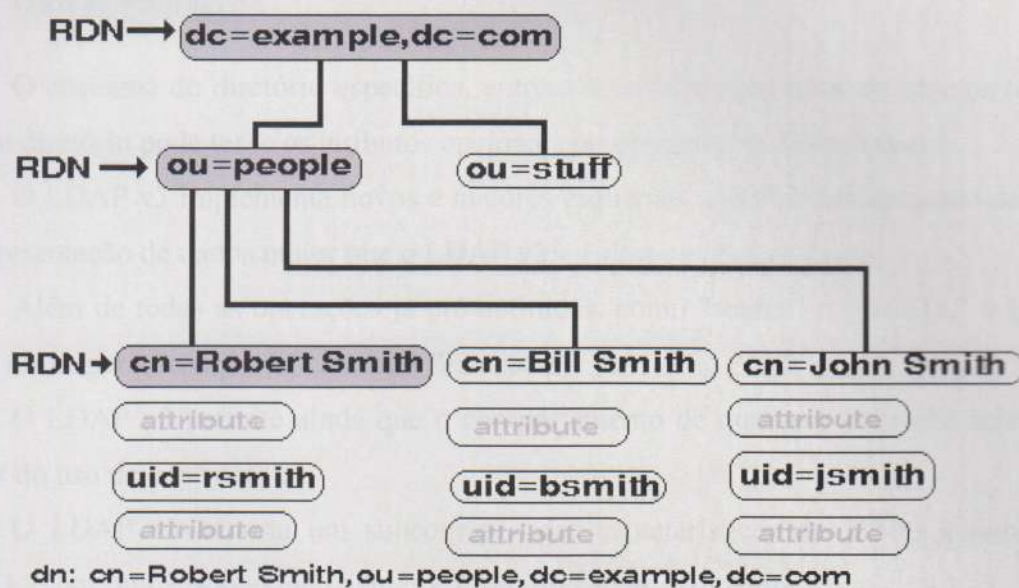


Figura 5: Diretório contido numa DIT

2.7 LDAP V2 x LDAP V3

O LDAP versão 3 foi desenvolvido para resolver algumas limitações existentes no LDAP versão 2. As limitações existiam nas áreas de internacionalização, autenticação, referral e distribuição. O LDAP versão 3 também permite que novas funcionalidade sejam implementadas sem que sejam feitas modificações ao protocolo. Isto é feito através das *extensions* e *controls*.

2.7.1 Métodos de Autenticação

O LDAP versão 2 suporta os seguintes métodos de autenticação:

- Anonymous
- Simple (clear text)
- Kerberos v4

Enquanto o LDAP versão 3 suporta estes:

- Digest-MD5
- Cram-MD5
- Anonymous
- External
- S/Key
- GSSAPI
- Kerberos V4

2.7.2 Outras Vantagens

O esquema do diretório especifica, entre outras coisas, os tipos de objetos (*entradas*) que um diretório pode ter, e os atributos opcionais ou obrigatórios destes objetos.

O LDAP v3 implementa novos e maiores esquemas, e define um conjunto de *sintaxes* de representação de dados maior que o LDAP v2.

Além de todas as operações já pré-definidas, como "search" e "modify," o LDAP v3 define uma operação especial chamada "*extended*".

O LDAP v3 permite ainda que o comportamento de qualquer operação seja alterado através do uso de *controls*.

O LDAP v3 suporta um subconjunto das características do X.500 maiores que o LDAP v2, por outro lado exige clientes e servidores mais elaborados.

O LDAP v3 possui bem mais recursos que o LDAP v2 fugindo um pouco do seu objetivo inicial de ser um protocolo simples, focado em servidores e clientes simples.

2.8 ESQUEMA NO LDAP

2.8.1 Visão Geral do Esquema LDAP

Um esquema ou Schema é um "pacote" de ObjectClasses e atributos que vem contido no pacote de instalação do ldap. Todos os ObjectClasses e atributos estão necessariamente definidos em um Schema. Todos os Schemas utilizados devem ser conhecidos pelo servidor ldap. No OpenLDAP por exemplo isto é feito via declaração "Include" no arquivo slapd.conf, este arquivo é o principal arquivo de configuração do ldap. Um atributo definido em um Schema pode integrar um ObjectClass em outro Schema.

```
# This file is the configuration file for slapd.
# It contains the configuration for the slapd daemon.
# The configuration is divided into sections.
# The sections are:
# - Global configuration
# - Database configuration
# - Access control configuration
# - Logging configuration
# - Network configuration
# - Plugins configuration
# - Schemas configuration
# - Modules configuration
# - Miscellaneous configuration
# The configuration is divided into sections.
# The sections are:
# - Global configuration
# - Database configuration
# - Access control configuration
# - Logging configuration
# - Network configuration
# - Plugins configuration
# - Schemas configuration
# - Modules configuration
# - Miscellaneous configuration
```

Figura 6: Arquivo slapd.conf

3 INSTALAÇÃO E CONFIGURAÇÃO DO LDAP

A instalação do LDAP apresentado nesta monografia foi feita em um sistema operacional UNIX mais precisamente o Linux na distribuição Debian versão 4.0.

Com o APT corretamente configurado no servidor é preciso instalar o `openldap` e seus utilitários que conterão os esquemas e os programas para manipulação dos diretórios. O seguinte comando deve ser executado para iniciar a instalação do `ldap` e seus utilitários.

```
apt-get install slapd ldap-utils
```

Após terminada a instalação se faz necessário a verificação do serviço observando se o daemon esta rodando adequadamente. O seguinte comando deve ser executado para fazer tal verificação:

```
netstat -lt | grep ldap
```

ou ainda:

```
ps -A | grep slapd
```

3.1 CONFIGURAÇÃO

Para fazer a configuração é necessário compreender que existem uma configuração que deve ser feita no cliente e outra no servidor. A configuração feita no cliente é feita no arquivo `/etc/ldap/ldap.conf`. A Figura 6 apresenta uma configuração no lado do cliente.

```
debian:/etc/ldap# cat ldap.conf
# $OpenLDAP: pkg/ldap/libraries/libldap/ldap.conf,v 1.9 2000/09/04 19:57:01 kurt
Exp $
#
# LDAP Defaults
#
# See ldap.conf(5) for details
# This file should be world readable but not world writable.

#BASE    dc=example, dc=com
#URI      ldap://ldap.example.com ldap://ldap-master.example.com:666

#SIZELIMIT    12
#TIMELIMIT    15
#DEREF        never
debian:/etc/ldap# _
```

Figura 6: Arquivo `ldap.conf`

A configuração no servidor é feita no arquivo `slapd.conf` que está dentro do diretório `/etc/ldap` como mostra a figura 7.

```
debian:~# cd /etc/ldap
debian:/etc/ldap# ls
ldap.conf  sasl2  schema  slapd.conf
debian:/etc/ldap#
```

Figura 7: Diretório LDAP

O `slapd.conf` é dividido em 3 seções principais: **Global**, **Backend** e **Database**.

Os parâmetros globais são aplicados a todos os diretórios do servidor. Há uma hierarquia entre estas seções, de forma que as diretivas definidas em **global** pode ser substituídas pela mesma diretiva em **backend** ou **database**, da mesma forma que as diretivas em **backend** podem ser substituídas pelas diretivas em **database**.

As linhas começando com `#` são ignorados. Linhas em branco são ignoradas (algumas versões do OpenLDAP pode interpretar uma linha começando com espaço como a continuação da linha anterior).

As DITs e as suas propriedades são definidas na seção **Database**.

Para cada DIT, deve haver uma seção **Database**. Por exemplo, se houver a necessidade de 2 DNs raízes `dc=domain,dc=com` e `dc=domain,dc=net` então será necessário criar 2 seções **Database**.

3.1.1 Parâmetros do OpenLDAP

3.1.1.1 Parâmetros da Seção Global

- **Allow bind_v2**: Permite que clientes LDAP v2 se conectem e se autenticuem (bind) ao servidor. Padrão = off
- **Include /etc/ldap/schema/core.schema**: Informa os schemas que o servidor reconhecerá. A ordem usada é muito importante, pois schemas normalmente possuem muitas dependências.
- **Pidfile /var/run/slapd/slapd.pid**: Informa o arquivo que armazenará o PID do daemon slapd.
- **Argsfile /var/run/slapd/slapd.args**: Informa o arquivo que armazena os argumentos que foram passados ao binário do slapd quando este foi executado.

- **LogLevel 0** : Informa o nível de detalhe do log gerado pelo OpenLDAP. A tabela 3.1 exibe os possíveis valores do loglevel.

loglevel	Logging description
-1	enable all debugging
0*	no debugging
1	trace function calls
2	debug packet handling
4	heavy trace debugging
8	connection management
16	print out packets sent and received
32	search filter processing
64	configuration file processing
128	access control list processing
256	stats log connections/operations/results
512	stats log entries sent
1024	print communication with shell backends
2048	print entry parsing debugging

TABELA 2: Valores do loglevel

- **Modulepath /usr/lib/ldap**: Informa o caminho de procura dos módulos do OpenLDAP.
- **Moduleload back_bdb**: Especifica a carga de um módulo específico. O módulo back_bdb é necessário para utilizar o BDB como banco de dados. O BDB (ou DB4) é o banco de dados padrão do OpenLDAP.
- **Tool-threads**: Especifica o número de threads utilizadas para a indexação do banco de dados. Quanto maior, maior o uso da CPU para esta atividade.
- **Threads**: Especifica o número máximo de threads que podem ser utilizadas simultaneamente pelo OpenLDAP para requisições.

3.1.1.2 Parâmetros da Seção Backend

A Seção **Backend** possui apenas um parâmetro principal, e parâmetros que são repassados para o backend utilizado. Este parâmetro é o backend bdb.

- **Backend bdb:** Informa o backend a ser utilizado, neste caso o BDB. A tabela 3.2 mostra as opções para este parâmetro.

Type	Description
bdb	Berkeley Database - preferred OpenLDAP option.
dnssrv	not really a database but allows slapd to supply referrals by interrogating the DNS records
ldap	Allows LDAP to follow (resolve) referrals rather than simply returning a referral.
ldbm	LDAP DBM - May use any one of BDB, GNU DBM, MDBM or NDBM.
meta	Allows LDAP to follow (resolve) referrals rather than simply returning a referral. An enhanced
monitor	Currently experimental LDAP backend.
null	LDAP equivalent of /dev/null.
passwd	Very specific configuration which maps searches to the system
perl	Allows LDAP to map requests directly to a PERL object to handle the request.
shell	Allows LDAP to map requests through a shell interface and run an external program - a poor
sql	Allows LDAP to use ODBC to map a RDBMS solution onto LDAP - an RDBMS wrapper.

TABELA 3: Parâmetros do backend bdb

3.1.1.3 Parâmetros da Seção Database

- **Database bdb:** Este parâmetro determina o início dos parâmetros de um banco de dados e também qual o tipo de banco que será utilizado. As opções possíveis são idênticas às do parâmetro Backend, presentes na tabela 3.2.
- **Suffix “dc=domínio”:** Informa a base ou sufixo do diretório armazenado por este banco de dados. Nota-se que um servidor pode possuir mais de um suffix.
- **Rootdn “cn=admin,dc=domínio”:** Determina uma ENTRY que terá privilégios de superusuário (root), independente dos parâmetros “access” definidos.
- **Rootpw {algoritmo}password:** Password para o root definido no parâmetro rootdn. Efetivo dentro da seção **Database** apenas. Pode ser declarado como plain text ou um hash. O comando **slappasswd** é usado para gerar os hashes.
- **Directory “/var/lib/ldap”:** Local onde os arquivos do banco de dados estarão fisicamente armazenados.
- **Dbconfig *:** Parâmetros que são repassados ao Backend do banco de dados.
- **Dbconfig set_cachesize 0 2000000 0:** Define a quantidade de RAM a ser utilizada pelo banco.
- **Index objectClass eq:** Define quais objetos serão indexados pelo banco de dados.

- **Access to dn.base=""by * read:** Acesso especial que permite a aplicações externas como o SASL ler informações básicas do diretório.
- **Access to attrs=userPassword by * auth:** Protege password de forma que somente o próprio usuário é capaz de ler ou modificar seu password.

3. Configurar o servidor LDAP no Samba

3.1. Criar o arquivo de configuração do Samba

```
# Samba configuration file
# See the smb.conf file in the Samba distribution for more information
# This file should be installed in the /etc directory
# The following lines are defaults
# You should change them to suit your needs
```

```
[global]
workgroup = SAMBA
server string = Samba Server %u
```

```
[local_auth]
enable = yes
```

Figura 8: Teste do servidor

4.2. OPERAÇÕES BÁSICAS

As operações de manipulação básica de LDAP são: adicionar, remover, modificar e consultar. Pode-se distinguir algumas operações básicas que se realizam no diretório.

- Adição (importação)
- Consulta
- Remoção
- Modificação

4.2.1. Adição

A adição ou importação de dados no LDAP é feita através de arquivos de texto. O Samba fornece um utilitário para importar dados no diretório.

A título de exemplo foram criados arquivos de texto com os dados de teste.

4 MANIPULAÇÃO DO DIRETÓRIO

4.1 TESTE DO SERVIÇO

Após a instalação correta e configuração adequada do OpenLDAP, pode-se testá-lo com o comando visto na figura 8.

```
debian:~# ldapsearch -x
# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# search result
search: 2
result: 32 No such object

# numResponses: 1
debian:~# _
```

Figura 8: Teste do serviço

4.2 OPERAÇÕES BÁSICAS

As ferramentas de manipulação básica do OpenLDAP se encontram no pacote **ldap-utils**. Pode-se distinguir algumas operações básicas que se realiza no diretório:

- Adição (importação)
- Consulta
- Remoção
- Modificação

4.2.1 Adição

A adição ou importação de entradas no LDAP, através do **ldap-utils**, pressupõe a criação de arquivos LDIF contendo os dados que serão adicionados ou importados à base de dados do diretório.

A título de exemplo foram criados diversos arquivos LDIF onde contém as entradas do diretório.

Arquivo country.ldif

```
dn: c=BR
objectClass: country
c: BR
```

Arquivo company.ldif

```
dn: o=Company, c=BR
objectClass: organization
o:
```

Arquivo IT.ldif

```
dn: ou=IT, o=Company, c=BR
objectclass: organizationalUnit
ou:IT
```

Arquivo HR.ldif

```
dn: ou=HR, o=Company, c=BR
objectclass: organizationalUnit
ou:HR
```

Arquivo fulano.ldif

```
dn: uid=fulano, ou=IT, o=Company, c=BR
objectClass: inetOrgPerson
uid: fulano
cn: fulado de tal
mail: fulanodetal@tal.com.br
sn: de tal
serPassword: {SSHA}LOXIchMsOauhKXMUYHyKaWvWvrbXRwnW
```

Arquivo sicrano.ldif

```
dn: uid=sicrano, ou=HR, o=Company, c=BR
objectClass: inetOrgPerson
uid: sicrano
cn: sicrano de tal
mail: sicranodetal@tal.com.br
sn: de tal
serPassword: {SSHA}LOXIchMsOauhKXMUYHyKaWvWvrbXRwnW
```

Após a criação dos arquivos .ldif precisa-se importar estes dados contidos dentro dos arquivos para a base de dados do ldap executando o seguinte comando. Este comando é repetido para todos os arquivos.

Arquivo country.ldif

```
debian:~# ldapadd -x -D "cn=admin,c=BR" -w 123 -v -f country.ldif
ldap_initialize( <DEFAULT> )
```

```
add objectClass:
    country
add c:
    BR
adding new entry "c=BR"
modify complete
```

Arquivo company.ldif

```
debian:~# ldapadd -x -D "cn=admin,c=BR" -w 123 -v -f company.ldif
ldap_initialize( <DEFAULT> )
add objectClass:
    organization
add o:
    Company
adding new entry "o=Company, c=BR"
modify complete
```

Arquivo it.ldif

```
debian:~# ldapadd -x -D "cn=admin,c=BR" -w 123 -v -f it.ldif
ldap_initialize( <DEFAULT> )
add objectClass:
    organizationalUnit
add ou:
    IT
adding new entry "ou=IT, o=Company, c=BR"
modify complete
```

Arquivo hr.ldif

```
debian:~# ldapadd -x -D "cn=admin,c=BR" -w 123 -v -f hr.ldif
ldap_initialize( <DEFAULT> )
add objectClass:
    organizationalUnit
add ou:
    HR
adding new entry "ou=HR, o=Company, c=BR"
modify complete
```

Arquivo fulano.ldif

```
debian:~# ldapadd -x -D "cn=admin,c=BR" -w 123 -v -f fulano.ldif
ldap_initialize( <DEFAULT> )
add objectClass:
    inetOrgPerson
```

```

add uid:
    fulano
add cn:
    fulano de tal
add mail:
    fulanodetal@tal.com.br
add sn:
    de tal
add userPassword:
    {SSHA}LOXIChMsOauhKXMUYHyKaWvWvrbXRwnW
adding new entry "uid=fulado, ou=IT, o=Company, c=BR"
modify complete

```

Arquivo sicrano.ldif

```

debian:~# ldapadd -x -D "cn=admin,c=BR" -w 123 -v -f sicrano.ldif
ldap_initialize( <DEFAULT> )
add objectClass:
    inetOrgPerson
add uid:
    sicrano
add cn:
    sicrano de tal
add mail:
    sicradetal@tal.com.br
add sn:
    de tal
add userPassword:
    {SSHA} LOXIChMsOauhKXMUYHyKaWvWvrbXRwnW
adding new entry "uid=sicrano, ou=IT, o=Company, c=BR"
modify complete

```

4.2.2 Consulta

Após feita a importação das entradas na estrutura da DIT pode-se executar consultas na base de dados para testar se esta funcionando corretamente o serviço de diretórios LDAP.

O programa `ldapsearch` está contido no pacote de utilitários do LDAP-UTILS. Este comando irá retornar a entrada BR presente na estrutura DIT.

```

debian:~# ldapsearch -x -b c=BR
# extended LDIF
# LDAPv3

```

```
# base <c=BR> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
# BR
dn: c=BR
objectClass: country
c: BR
# Company, BR
```

Outro exemplo de consulta é mostrado nas linhas abaixo, desta vez é aplicado um filtro pelo UID=fulano.

```
debian:~# ldapsearch -x -b c=BR '(uid=fulano)'
# extended LDIF
#
# LDAPv3
# base <c=BR> with scope subtree
# filter: (uid=fulano)
# requesting: ALL
#
# fulano, IT, Company, BR
dn: uid=fulano,ou=IT,o=Company,c=BR
objectClass: inetOrgPerson
uid: fulano
cn: Fulano de tal
mail: fulanodetal@detal.com.br
sn: de tal
```

4.2.3 Modificação

A modificação de uma Entry com o ldap-utils passa pelo mesmo procedimento utilizado na adição das entradas. Edita-se o arquivo usado na criação caso esteja disponível ou constrói-se um novo arquivo contendo a informação atualizada.

Segue um exemplo de modificação do email da entry 'cn=fulado de tal'.

1. Arquivo fulano.ldif original:

Arquivo fulano.ldif

```
dn: uid=fulano, ou=IT, o=Company, c=BR
objectClass: inetOrgPerson
uid: fulano
cn: fulado de tal
```

```
mail: fulanodetal@tal.com.br
sn: de tal
serPassword: {SSHA}LOXIchMsOauhKXMUYHyKaWvWvrbXRwnW
```

2. Programa para fazer a modificação e resposta do servidor:

```
debian:~# ldapmodify -x -w 123 -v -D 'mail=fulado@detal.com.br' -f \
fulano.ldif
ldap_initialize( <DEFAULT> )
replace objectClass:
    inetOrgPerson
replace uid:
    fulano
replace cn:
    fulado de tal
replace mail:
    fulano@tal.com.br
replace sn:
    de tal
replace userPassword:
    {SSHA}LOXIchMsOauhKXMUYHyKaWvWvrbXRwnW
modifying entry "uid=fulano, ou=IT, o=Company, c=BR"
modify complete
```

4.2.4 Remoção

No ldap-utils, a remoção dispensa o uso de um arquivo LDIF, entretanto é necessário saber a dn (Distinguished Name) da entrada que se quer apagar ou remover. A remoção também pode ser recursiva, utilizando-se o parâmetro **-r** com o **ldapdelete**.

1. Identificar o dn da entrada que possui 'cn=fulano de tal'.

```
debian:~# ldapsearch -x -b c=BR -LLL '(cn=fulano de tal)'
dn: uid=fulano,ou=IT,o=Company,c=BR
objectClass: inetOrgPerson
uid: fulano
cn: Fulano de tal
mail:: ZWR1YXJkb2NvZWxob2xpbWFAaG90bWFpbC5jb20g
```

2. Após a identificação do dn pode-se remover com segurança a entrada cuja a DN é UID=fulano,o=IT,c=BR.

```

debian:#      ldapdelete -x -D 'cn=admin,c=BR' -w 123 -v \
uid=fulano,ou=IT,o=Company,c=BR
ldap_initialize( <DEFAULT> )
deleting entry "uid=fulano,ou=IT,o=Company,c=BR"

```

4.3 RESUMO DAS OPERACOES BÁSICAS

Aqui serão listados os principais comandos do pacote ldap-utils.

- Ldapsearch# ldapsearch -x -b c=BR -LLL '(cn=fulano de tal)'
 - Operação para procurar por uma entrada na árvore DIT
- Ldapadd# ldapadd -x -D "cn=admin,c=BR" -w 123 -v -f fulano.ldif
 - Operação para importar um arquivo ldif para a base de dados na estrutura de diretório do ldap.
- Ldapmodify# ldapmodify -x -w 123 -v -D 'cn=admin,c=BR' -f fulano.ldif
 - Operação para modificar alguma entrada de um dn qualquer.
- Ldapdelete# ldapdelete -x -D 'cn=admin,c=BR' -w 123 -v uid=eduardo,ou=IT,o=Company,c=BR
 - Operação para remover uma entrada qualquer da base do ldap.
- Slappasswd (pacote openldap)# slappasswd -h {SHA} -s 123
 - Alterar senha da base do ldap.

4.4 BROWSERS LDAP

4.4.1 Chicago University Ldap Browser

É um *browser* LDAP *freeware* em Java. Possui diversos recursos. Pode ser encontrado no na página da web: <http://www.mcs.anl.gov/~gawor/ldap>. Sua última versão é a 2.8.2 beta. A figura 9 mostra a interface do *Chicago University Ldap Browser*.

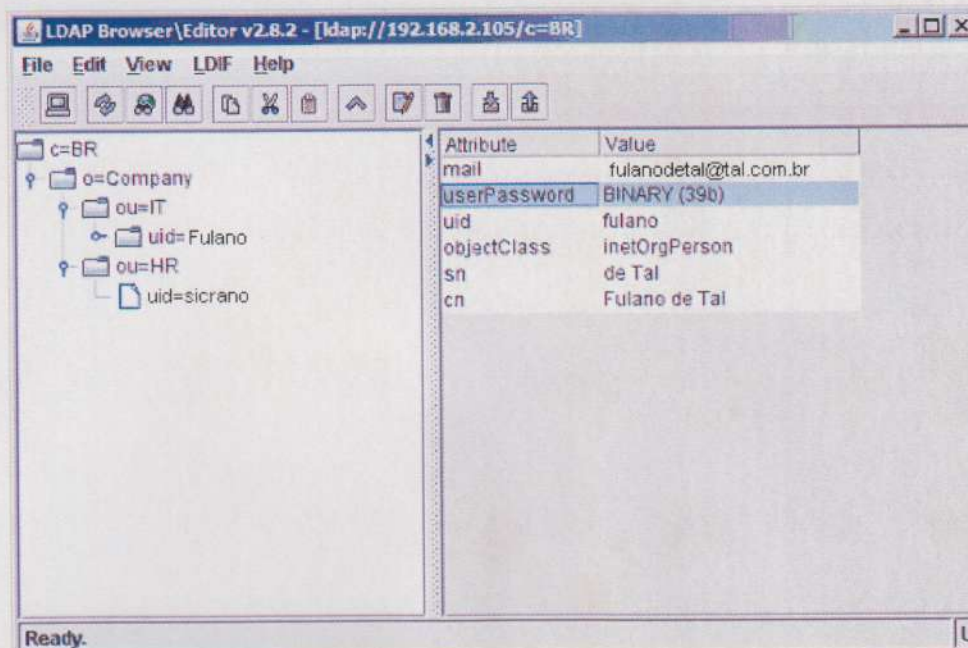


Figura 9: Chicago University Ldap Browser (LDAP Browser/Editor, 2007)

4.4.2 JXplorer

É um *browser* LDAP *freeware* também em Java. Possui boa interface e muitos recursos. Pode ser baixado da sua página na internet: <http://home.megapass.co.kr/~woosjung/>. A versão mais recente até o momento é a 1.8.0.1. A figura 10 mostra a interface do JXplorer.

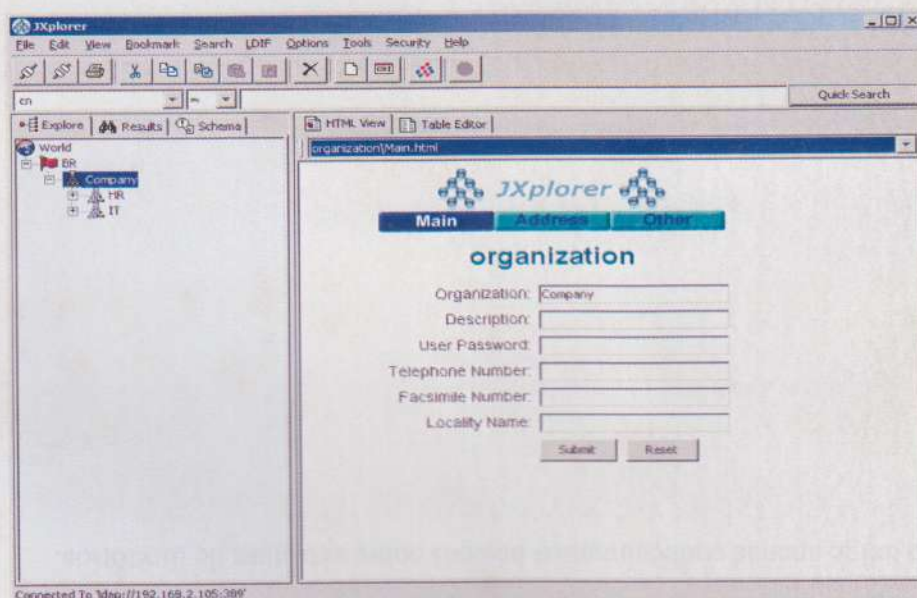


Figura 10: JXplorer (JExplorer, 2007)

5 CONCLUSÃO

Conclui-se que o protocolo LDAP é uma ferramenta eficaz na manipulação de diretórios, o que torna o trabalho do administrador de redes mais fácil quando se precisa implementar serviços de diretórios. Sua instalação na plataforma linux não é complicada e sua configuração exige apenas conhecimentos básicos sobre estrutura de diretórios.

REFERÊNCIAS BIBLIOGRÁFICAS

SENA, Clóvis LDAP. Lightweight Directory Access Protocol - Um guia prático. Ed. Ciência Moderna, 2005

KANIES, Luke A. Uma Introdução ao LDAP. Disponível em <http://br.geocities.com/cesarakg/IntroLDAP-ptBR.html>. Acessado em Junho/2007.

LDAP Browser/Editor. Disponível em www.mcs.anl.gov/~gawor/ldap. Acessado em Junho/2007.

JExplorer. Disponível em <http://home.megapass.co.kr/~woosjung/>. Acessado em Junho/2007.