UNIVERSIDADE FUMEC
FACULDADE DE CIÊNCIAS EMPRESARIAIS - FACE

BRENO DA SILVA AMARAL

LDAP: Centralização e disponibilidade de informações

BELO HORIZONTE
2010

BRENO DA SILVA AMARAL

LDAP: Centralização e disponibilidade de informações

Monografia apresentada à Universidade FUMEC, no curso de Ciência da Computação, apresentado à disciplina Trabalho de Conclusão de Curso.

Orientadores:
Professor Osvaldo Manoel Corrêa
Professor Ricardo Terra Nunes Bueno Villela.

BELO HORIZONTE
2010

Aos meus orientadores, em especial ao professor Ricardo Terra que confiou e apoiou-me na realização deste trabalho, além de oferecer-me a sua amizade. Aos meus "co-orientadores", Marden Cicarelli Pinheiro e Júlio Márcio Dutra Couto, sobretudo pelos ensinamentos, sejam eles profissionais ou de vida, e pela grande e bela amizade que temos.

Ao meu pai, Lucas Amaral, que foi e continua sendo minha inspiração profissional, fazendo-me seguir seus passos na área de tecnologia computacional.

Sem vocês, esse trabalho não seria possível.

"É melhor tentar e falhar,
que preocupar-se e ver a vida passar;
é melhor tentar, ainda que em vão,
que sentar-se fazendo nada até o final.
Eu prefiro na chuva caminhar,
que em dias tristes em casa me esconder.
Prefiro ser feliz, embora louco,
que em conformidade viver ..."

Martin Luther King

RESUMO

A redundância de dados devido à descentralização de base de dados e à falta de integração entre sistemas são responsáveis por diversos problemas enfrentados por empresas onde a TI (Tecnologia da Informação) é de vital importância. Uma das possíveis soluções para esse problema é a utilização de tecnologias cujas principais características são centralização da informação e integração entre sistemas. Nesses casos, a implementação do LDAP (*Lightweight Directory Access Protocol*) é adequada, pois é focada exatamente na centralização e disponibilidade de informação. Portanto, este estudo sumariza os conceitos do LDAP trazendo informações importantes que vão desde a sua origem até implementações existentes. Ademais, este estudo ilustra um estudo de caso que demonstra uma implementação do LDAP utilizando OpenLDAP – uma solução de código aberto – e também apresenta informações detalhadas de todo o processo, tais como arquivos de configuração e testes, que demonstram a eficácia do protocolo para a solução dos problemas supracitados.

Palavras-chave: LDAP, OpenLDAP, Disponibilidade, Centralização, Segurança, Serviço de Diretórios.

ABSTRACT

Duplicate of information due to the decentralization of databases and the lack of systems integration are responsible for several problems faced by companies where IT (Information Technology) is crucial. One way to solve this problem is using technologies whose main characteristics are the information centralization and system integration. In these cases, the implementation of LDAP (Lightweight Directory Access Protocol) is suitable because it is exactly focused on information centralization and availability. Therefore, this study summarizes the LDAP concepts giving important informations since its origin until the existing implementations. Moreover, it illustrates a case study demonstrating the LDAP implementation using the OpenLDAP open source solution and also presents detailed process informations such as configuration files and tests to demonstrate the effectiveness of the protocol to solve aforementioned problems.

Keywords: LDAP, OpenLDAP, Availability, Centralization, Security, Directories Service.

LISTA DE FIGURAS

FIGURA 1 - Conexão entre Servidor LDAP e Cliente. 14
FIGURA 2 - Utilização do LDAP como base para diversos serviços na rede. 15
FIGURA 3 - Estrutura da árvore de busca do LDAP. 18
FIGURA 4 - Alguns atributos de classe do esquema `inetOrgPerson`. 20
FIGURA 5 - Definição do objeto de classe do esquema `inetOrgPerson` 21
FIGURA 6 - Arquivo `insertUser.ldif`. 22
FIGURA 7 - Arquivo `updateUser.ldif`. 22
FIGURA 8 - Modelo de sistema baseado em Domínio. 27
FIGURA 9 - Exemplo de implementação do *Active Directory*. 28
FIGURA 10 - Representação de um nome de objeto distinto em uma árvore de diretórios. ... 29
FIGURA 11 - Estrutura hierárquica da Empresa *KingTech* 29
FIGURA 12 - Tela de propriedades de uma conta criada a partir de um objeto `inetOrgPerson` no AD. 30
FIGURA 13 - Implementação do OpenLDAP. 32
FIGURA 14 - Organograma das atividades planejadas para implementação do OpenLDAP. 41
FIGURA 15 - Parte do arquivo de configuração `smb.conf` 44
FIGURA 16 - Arquivo de configuração `smbldap_bind.conf` 44
FIGURA 17 - Parte do arquivo de configuração `smbldap.conf` 45
FIGURA 18 - Tela de configuração dos parâmetros de autenticação do *CentOS* que integra ao Sistema Operacional os módulos de autenticação do LDAP. 48
FIGURA 19 - Tela de configuração posterior a FIG.15 dos parâmetros de autenticação do *CentOS* que integra ao Sistema Operacional os módulos de autenticação do LDAP. 48
FIGURA 20 - Inserção de estação cliente no domínio FOOBAR.COM 51
FIGURA 21 - Confirmação de sucesso no ingresso ao domínio FOOBAR.COM 52
FIGURA 22 - Pedido de credenciais para autenticação na estação de trabalho. 53
FIGURA 23 - Área de trabalho do usuário `bsa` após autenticação na estação de trabalho. 54
FIGURA 24 - Configuração de Proxy no navegador Internet Explorer. 54
FIGURA 25 - Pedido de credenciais LDAP pelo serviço de Proxy no navegador de Internet. 55

LISTA DE SIGLAS

ACL	*Access Control List*
AD	*Active Directory*
ADS	*Active Directory Service*
API	*Application Programming Interface*
BDB	*Berkeley's Data Base*
BSD	*Berkeley Software Distribution*
DAP	*Directory Access Protocol*
DNS	*Domain Name System*
HTTP	*Hypertext Transfer Protocol*
IMAP	*Internet Message Access Protocol*
IP	*Internet Protocol*
ISO	*International Organization Standardization*
LDAP	*Lightweight Directory Access Protocol*
LDIF	*LDAP Interchange Format*
OSI	*Open System Interconnection*
RFC	*Request for Comments*
SQL	*Structured Query Language*
SASL	*Simple Authentication and Security Layer*
SMTP	*Simple Mail Transfer Protocol*
SNMP	*Simple Network Management Protocol*
SSL	*Secure Socket Layer*
TCP	*Transmission Control Protocol*
TLS	*Transport Layer Security*

SUMÁRIO

INTRODUÇÃO 10

CAPÍTULO 1 - LDAP: *LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL* 12

1.1 Visão Geral 13
1.1.1 Diretórios 13
1.1.2 O protocolo 13
1.2 Modelos de diretórios de informação e organização estrutural 15
1.2.1 Distribuição hierárquica 16
1.2.2 Tipos de informação 19
1.2.3 Esquemas (*Schemas*) 19
1.2.4 Arquivos LDIF 21
1.3 Segurança 22
1.3.1 Autenticação básica 23
1.3.2 *Simple Authentication and Security Layer* (SASL) 23
1.3.3 *Secure Socket Layer* (SSL) e *Transport Layer Security* (TLS) 24

CAPÍTULO 2 - IMPLEMENTAÇÕES DO LDAP 26

2.1 *Active Directory* (AD) 26
2.1.1 *Active Directory Service* e o LDAP 28
2.2 OpenLDAP 31
2.2.1 Servidor LDAP 33
2.2.2 Cliente LDAP 34
2.2.3 Utilitários do OpenLDAP 34
2.2.4 Bibliotecas do OpenLDAP 35

CAPÍTULO 3 - ESTUDO DE CASO 37

3.1 Cenário Atual 37
3.2 Planejamento 38
3.2.1 Introdução 38
3.2.2 Requisitos necessários de software 39
3.2.3 Estrutura Analítica do Projeto (EAP) 40
3.3 Implementação 42
3.4 Testes 50

CONCLUSÃO 56

REFERÊNCIAS 59

ANEXOS 61

INTRODUÇÃO

Com o surgimento das redes de computadores e a rede mundial (Internet) em ambientes empresariais, deu-se início a disponibilização das informações por meios digitais. Viu-se então que era necessário organizar essas informações de forma a manter o controle e o acesso otimizado sobre elas (MENDES, 2007).

Assim, para que essas informações fossem controladas e organizadas, desenvolveu-se o Serviço de Diretórios de Rede (*Network Directory Service*) ou Serviço de Informação Corporativa (*Enterprise Information Service*), visando o monitoramento de todos os nomes, perfis, endereços de email e endereços de máquinas de cada usuário na rede.

Dessa forma, na década de 1980, como primeiro passo ao *Network Directory Service*, a *International Telecommunications Union* (ITU) propôs "[...] um conjunto de padrões e um modelo de informações para os serviços de diretórios que pudesse ser acessado globalmente [...]" (SCRIMGER *et al*, 2000. p. 388), surgindo assim o protocolo X.500 DAP (*Directory Access Protocol*).

No entanto, como desvantagem, por ser o protocolo X.500 DAP baseado no modelo OSI (*Open System Interconnection*), não possuía suporte para acesso a Internet, afinal, o modelo OSI é um modelo de transmissão de dados pré-Internet. Além disto, o protocolo X.500 DAP era extremamente difícil de ser implementado, o que resultava em aplicações lentas e complexas (TRIGO, 2007).

Sendo assim, segundo Trigo (2007. p. 22), com o crescimento da Internet e com a popularidade da arquitetura TCP/IP, "[...] foi criado um protocolo de acesso a diretórios que se encaixasse melhor nos moldes do TCP/IP [...]", surgindo assim o LDAP (*Lightweight Directory Access Protocol*) ou Protocolo Leve de Acesso a Diretórios, um protocolo de simples implementação que oferece facilidade e leveza no acesso aos serviços de diretórios, capaz de se integrar a diversos serviços de rede e de forma centralizada.

Assim, este trabalho destina-se a apresentação de alguns conceitos e aplicações do protocolo LDAP, trazendo como tema: LDAP: Disponibilidade e centralização de informações. Nesse sentido, esta monografia apresenta, além dos conceitos mais importantes referentes ao protocolo LDAP, algumas de suas implementações e um estudo de caso.

Esta monografia está organizada conforme a seguir: O Capítulo 1 trata a parte conceitual necessária desse protocolo trazendo uma visão geral sobre ele, seu modelo de

diretórios de informações e organização estrutural e questões de segurança. É provido todo o entendimento a respeito do protocolo LDAP esclarecendo qual é sua finalidade, por que e quando utilizá-lo. Já o 2 apresenta duas das implementações desse protocolo existentes, destacando o *Active Directory* (uma implementação do protocolo LDAP feita pela Microsoft) e o OpenLDAP (implementação de código aberto do protocolo LDAP utilizada principalmente em sistemas operacionais baseados na plataforma UNIX). O Capítulo 3 abordará um estudo de caso em uma empresa que apresenta problemas que podem ser solucionados a partir da implementação do protocolo LDAP. Para esse caso, é descrito o cenário atual que elenca os principais problemas que direcionam ao uso do LDAP. Dessa forma, um plano de implementação é descrito trazendo informações de tudo o que é necessário para a sua execução; por fim, é demonstrada a execução desse plano de implementação, realizando os devidos testes que comprovam o sucesso do projeto e o cumprimento das necessidades levantadas na resolução dos problemas.

CAPÍTULO 1 - LDAP: *Lightweight Directory Access Protocol*

Por definição, de acordo com o Linux *Dictionary*[1], protocolo é um conjunto de regras que definem exatamente como as informações devem ser trocadas entre dois sistemas assegurando assim a uniformidade na troca de informações entre computadores e aplicações.

Nesse contexto, segundo Trigo, (2007, p. 17), "[...] LDAP (*Lightweight Directory Access Protocol*) ou Protocolo Leve de Acesso a Diretórios, é um conjunto de regras que controlam a comunicação entre serviços de diretórios e seus clientes".

Por ser um protocolo de Internet, o LDAP está descrito sob o RFC[2] 1487 da *Internet Engineering Task Force* (IETF) ou Força Tarefa de Engenharia da Internet, onde se encontram os detalhes técnicos sobre seu funcionamento e os padrões que devem ser seguidos para sua implantação e implementação.

Atualmente, o LDAP está em sua terceira versão, denominada LDAPv3, desenvolvida em 1997 e descrita no RFC 2251. Recentemente, em junho de 2006, a especificação original foi atualizada e descrita nos RFCs 4510-4519, que fornecem especificações mais claras e coesas desse protocolo. (BUTCHER, 2007, p. 7-8)

Sendo assim, este capítulo abordará de uma maneira simplificada esses RFCs, uma vez que o objetivo deste estudo não é detalhar os conceitos técnicos da implementação desse protocolo, mas sim, demonstrar características mais marcantes na disponibilização de informações de forma centralizada e consistente. Na seção 1.1 é apresentada uma visão geral do protocolo LDAP e, em seguida, na seção 1.2, é apresentado o modo como o LDAP organiza sua estrutura e como constitui seu modelo de diretório de informações; já a seção 1.3 demonstra como assegurar as informações providas pelo LDAP, trazendo como tema "Segurança".

---

[1] PROTOCOL. In: **Linux Dictionary**. 2004. Disponível em: <http://tldp.org/LDP/Linux-Dictionary/html/p.html>. Acesso em: 31 ago. 2010.

[2] RFC – Acrônimo de *Request for Comments* – o nome do resultado e do processo de criação de um padrão na Internet (RFC. In: **Linux Dictionary**. 2004. Disponível em: <http://tlpd.org/LDP/Linux-Dictionary/html/r.html>. Acesso em: 31 ago. 2010)

1.1 Visão Geral

Nesta seção são apresentados alguns dos conceitos que são importantes para a compreensão do LDAP.

1.1.1 Diretórios

Segundo Trigo (2007, p. 17, grifo do autor), “diretório significa, literalmente, ‘algo usado para indicar direções’, ou seja, algo que indica um caminho para se chegar àquilo que se procura”. Nessa perspectiva, um bom exemplo de diretório é uma lista telefônica, onde com um nome obtem-se o telefone correspondente.

Em uma definição formal, podemos dizer que um diretório é um serviço que tem por finalidade armazenar informações de forma hierárquica e com o objetivo de facilitar a busca e a recuperação dessas informações. (TRIGO, 2007)

Analogamente, Butcher (2007) diz que um diretório tem como função manter informações sobre um conjunto de entidades, como pessoas ou organizações, fornecendo serviços para acessá-las.

1.1.2 O protocolo

Originalmente desenvolvido na Universidade de Michigan por Tim Howes e Steve Killie, o LDAP foi concebido como um protocolo de rede capaz de obter informações a partir de diretórios do modelo X.500 - arquitetura de servidores de diretórios desenvolvida na década de 1980 e padronizada em 1988 - tendo como aplicação principal a busca e a modificação dos dados em serviços de diretórios. Entretanto, com a popularidade da

arquitetura TCP/IP[3] e o advento da Internet, esse protocolo foi amadurecido e uma série de novas normas foram criadas, tornando-o bastante difundido nos ambientes corporativos. (BUTCHER, 2007)

O LDAP define um protocolo de troca de mensagens usado por clientes/servidores de diretórios. Assim, o cliente envia uma requisição de informações ao servidor que a processa e realiza as operações necessárias em seu diretório de busca. Assim, é devolvida uma resposta contendo as informações solicitadas ou erros para a requisição desse cliente, como exemplificado pela FIG. 1. (RFC 2251)

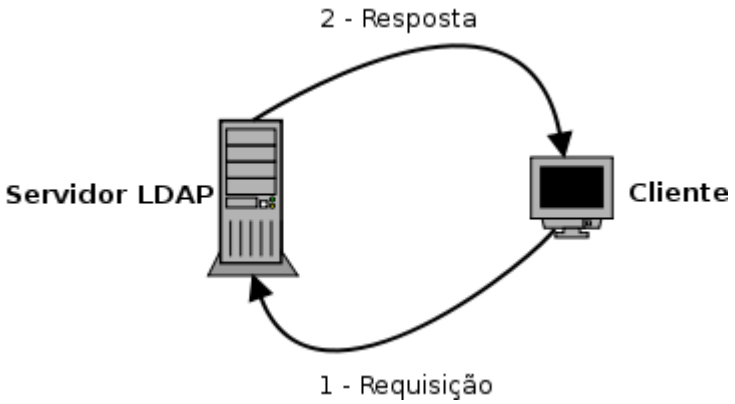


FIGURA 1 - Conexão entre Servidor LDAP e Cliente.
Fonte: Do autor.

Dessa forma, por trabalhar sobre a arquitetura TCP/IP, o LDAP usufrui de todas as suas vantagens, oferecendo a garantia da entrega de pacotes de mensagens livres de erro, na sequencia correta e sem perdas ou duplicações de dados. (SCRIMGER *et al*, 2000)

Por ser leve, rápido e de simples implementação, o LDAP não é considerado somente um protocolo, mas sim um diretório de serviço escalonável, tornando-o completamente aceito como o padrão Internet para serviços de diretórios que executam sobre a arquitetura TCP/IP. (SCRIMGER *et al*, 2000)

Por essa razão e, em função de suas características, principalmente por ser padrão aberto, o protocolo LDAP oferece suporte a diversos serviços, tais como: aplicações *web*[4], Intranets[5], servidores IMAP[6], bancos de dados etc. Lozano (2002), *apud* Trigo (2007, p. 23)

[3] TCP/IP - (suite TCP/IP, pilha TCP/IP) Descreve os protocolos usados pela Internet. Conceito dado ao transporte genérico de dados que englobam comunicação entre máquinas e comunicação entre aplicações. (TCP/IP in: **Linux Dictionary**. 2004. Disponível em: <http://tldp.org/LDP/Linux-Dictionary/html/t.html>.)

[4] *Web*, *World Wide Web* ou WWW é o nome dado ao corpo de informações na *Internet* caracterizado por imagens gráficas coloridas e *links* de hipertexto. Por outro lado, um navegador é a ferramenta que permite visualizar as informações que contém as imagens gráficas coloridas e *links*. (SCRIMGER *et al*, 2000, p. 341)

[5] *Intranet* - Rede de computadores privada que utiliza-se dos protocolos de *Internet* (Nota do autor)

“ressalta a integração do LDAP com outros serviços, complementando a infraestrutura de redes, fornecendo novos recursos e, especialmente, maior integração, diferentemente de outros protocolos e linguagens estabelecidos como SNMP, HTTP, SMTP, IMAP ou SQL”, assim como representado na FIG. 2.

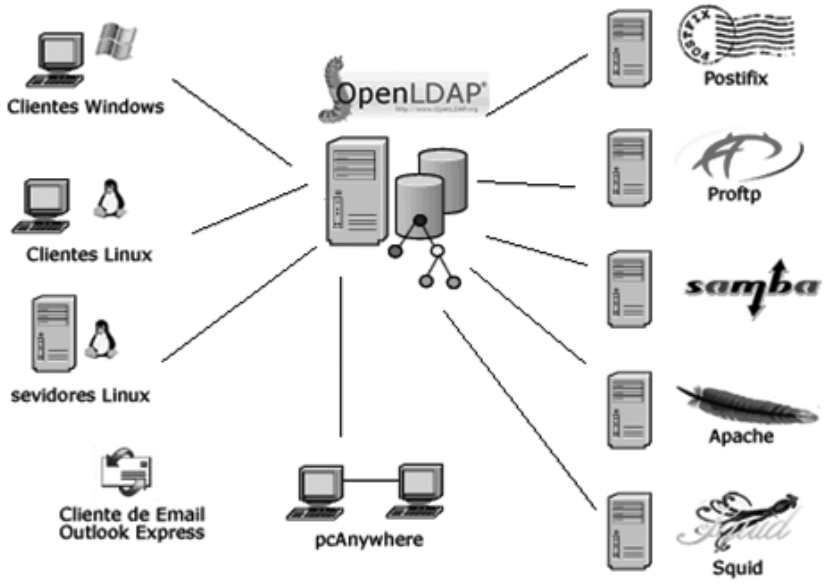


FIGURA 2 - Utilização do LDAP como base para diversos serviços na rede.
Fonte: OpenLDAP Fondation. Disponível em: <http://www.openldap.org>. Acesso em: 20 out. 2010.

1.2 Modelos de diretórios de informação e organização estrutural

Ainda como exemplo a lista telefônica citada na sub-seção 1.1.1, é demonstrada a sua relação com um servidor de diretórios. Uma lista telefônica contém um tipo muito específico de informações, organizadas em uma maneira e para um propósito específico. Segue um exemplo de uma entrada de lista telefônica:

```
Universidade FUMEC
Rua Cobre, 200
Cruzeiro, Belo Horizonte
MG, Brasil
30310-190
(31) 3228-3000
```

[6] IMAP - *Internet Message Access Protocol* ou Protocolo de Acesso a Mensagens de *Internet* – Protocolo que permite ao cliente o acesso e a manipulação de mensagens eletrônicas em um servidor (RFC 3501)

Conforme mencionado, o tipo de diretório anteriormente possui informações, organização e propósito específicos. Nesse caso, as informações visam uma forma de se contatar uma instituição específica (Universidade FUMEC), organizada em um padrão familiar a esse propósito (endereço e número de telefone).

Assim, pode-se dizer que essas informações estão organizadas de tal forma que uma pessoa consiga obter rapidamente as informações desse contato a partir do nome da instituição. Entretanto, vale observar que o formato dessa entrada só faz sentido se o leitor é capaz de compreendê-lo.

Em seguida, um exemplo extrapolado a partir do conhecimento sobre as informações contidas nesse diretório:

```
Nome da organização: Universidade FUMEC
Endereço: Rua Cobre, 200
Bairro: Cruzeiro
Cidade: Belo Horizonte
Estado: MG
País: Brasil
CEP: 30310-190
Telefone: (31) 3228-3000
```

Através do exemplo descrito anteriormente, fica claro e explícito o significado de cada informação, pois as informações são divididas em pedaços mais específicos (no caso, bairro e cidade, estado e país estão em linhas separadas) e cada uma das informações é precedida por um identificador denominado **entrada**. (grifo nosso)

Essa é uma forma mais próxima de como esse registro seria armazenado em um diretório LDAP. No entanto, ainda existe o problema de não duplicidade de informações. Isso faz com que seja necessário um identificador único para esse registro. (BUTCHER, 2007).

1.2.1 Distribuição hierárquica

Trigo (2007) afirma que a flexibilidade do LDAP se dá em função da organização das informações de maneira hierárquica, possuindo um elemento raiz - local por onde as buscas se iniciam - e nós filhos que são percorridos até que o elemento desejado seja encontrado.

Nesse sentido, a raiz e os nós são representações de diretórios, sendo que cada um pode conter outros diretórios, ou elementos denominados entradas. Cada entrada pode conter um ou mais atributos que farão referência a um ou mais valores associados a eles, de acordo com o tipo de dado definido previamente. (TRIGO, 2007)

Segundo Trigo (2007), "[...] uma característica herdada do padrão X.500 foi o uso de mnemônicos para definir nomes de atributos de diretórios e entradas [...]", que irão dar composição a estruturação organizacional da informação, tendo ainda o estilo de árvore de diretórios do DNS[7] como forma de representação de sua hierarquia.

Como exemplo, para diretórios, tem-se os seguintes atributos:

Atributo	**Descrição**
`c`	Para diretórios que representam países (do inglês *country*).
`o`	Para o nome da empresa (do inglês *organization*).
`ou`	Para departamento (do inglês *organization unit*).

Tabela: Lista de atributos para diretórios
Fonte: OpenLDAP: Uma abordagem integrada (TRIGO, 2007)

Ainda, para as entradas, pode ter-se:

Atributo	**Descrição**
`cn`	Como atributo de nome (do inglês *common name*).
`dc`	Como atributo de Componente de Domínio (do inglês *Domain Component*)
`uid`	Para identidade de usuários (do inglês *user* ID).
`gn`	Para nome próprio de uma pessoa (do inglês *given name*).
`sn`	Para o sobrenome de uma pessoa (do inglês *surname*)

Tabela: Lista de atributos para entradas
Fonte: OpenLDAP: Uma abordagem integrada (TRIGO, 2007)

Nesse sentido, uma forma de se distinguir registros semelhantes é a criação de um identificador único para cada um deles, denominando assim o *Distinguished Name* ou Nome Distinto (`DN`). (BUTCHER, 2007)

[7] DNS - *Domain Name System* ou Sistema de Nomes de Domínio é o sistema que mapeia nomes de máquinas (*hostnames*) para endereços IP. Ele converte nomes de computador aos endereços IP que todas as máquinas em uma rede têm (DNS In: **Linux Dictionary**. 2004. Disponível em <http://tldp.org/LDP/Linux-Dictionary/html/d.html>. Acesso em: 26. out. 2010).

Cada entrada possui um identificador único `DN` que é composto de um *Relative Distinguished Name* ou Nome Distinto Relativo (RDN), que é formado pela junção dos atributos de entrada, precedido do `DN`. Analogamente, podemos atribuir ao `DN` o caminho completo de um arquivo em um sistema de arquivos e o `RDN` como apenas o nome relativo do arquivo dentro de uma pasta.

Segue um exemplo de composição de `DN` e seus atributos.

```
dn: uid=breno, ou=cpd, dc=fea, dc=fumec, dc=br
```

No que se refere à organização estrutural das informações do LDAP e a ordem de leitura da informação, temos os primeiros níveis de diretórios refletindo o nome de domínio da empresa, conforme pode ser observado na FIG. 3:

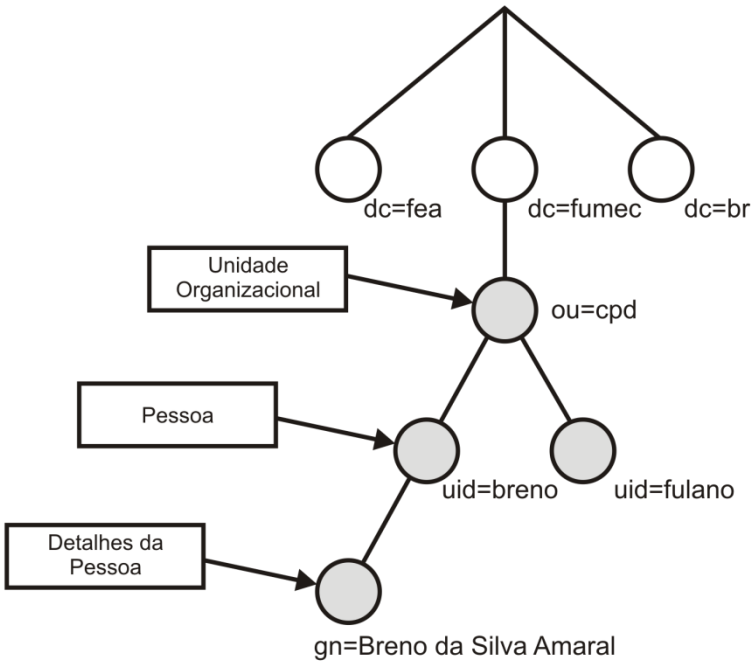


FIGURA 3 - Estrutura da árvore de busca do LDAP.
Fonte: Do autor.

1.2.2 Tipos de informação

Trigo (2007) afirma que qualquer tipo de informação pode ser armazenado nos atributos LDAP, como nomes, identificadores de usuários, senhas, emails, locais de trabalho, telefone, fotos etc. Isso é definido através de *schemas*[8] - arquivos que definem a estrutura das entradas e dos atributos.

Para tanto, as entradas são definidas em classes de objetos, que definem uma lista de atributos, podendo serem obrigatórios ou opcionais. Cada entrada deve possuir pelo menos uma classe de objetos, que definirá quais atributos cada entrada deve conter. Convém ainda mencionar que pode-se possuir atributos opcionais.

1.2.3 Esquemas (*Schemas*)

Em sua definição, Butcher (2007) diz que "*Schemas* é a maneira padrão de descrever a estrutura dos objetos que podem ser armazenados dentro do diretório". Em geral, esquemas LDAP são compactos e utilizam-se basicamente de duas diretivas: *Attribute Type* e *Object Class*, cada qual com suas variedades de argumentos, conforme detalhamento a seguir:

Attribute Type: Define um atributo, incluindo ainda os nomes dos atributos que poderá conter;

Object Class: Especifica o nome da classe do objeto, os atributos obrigatórios e opcionais, e seu tipo de objeto.

Diante disso, esquemas LDAP são usados para definir atributos, classes de objetos e várias outras regras que determinam a estrutura da árvore de informações do diretório. Sucintamente, o esquema LDAP refere-se a um conjunto de definições conceituais.

Por exemplo, o esquema `inetOrgPerson` contém a definição de sua classe de objetos, bem como todos os atributos permitidos ou exigidos. Ela representa como as pessoas são associadas a uma organização de alguma forma. (RFC 2798)

[8] A sessão 1.2.3 desta pesquisa monográfica traz os detalhes e informações a respeito dos arquivos schema. (Nota do autor)

Segue um exemplo de como esse esquema - `inetOrgPerson` - é estruturado. Primeiramente, é demonstrado como os atributos são descritos (FIG. 4) e, em seguida, como o objeto a ser utilizado pelo LDAP é formado a partir desses atributos (FIG. 5).

```
# departmentNumber
# Code for department to which a person belongs.  This can also be
# strictly numeric (e.g., 1234) or alphanumeric (e.g., ABC/123).
attributetype ( 2.16.840.1.113730.3.1.2
   NAME 'departmentNumber'
   DESC 'RFC2798: identifies a department within an organization'
   EQUALITY caseIgnoreMatch
   SUBSTR caseIgnoreSubstringsMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

# displayName
# When displaying an entry, especially within a one-line summary list, it
# is useful to be able to identify a name to be used.  Since other attri-
# bute types such as 'cn' are multivalued, an additional attribute type is
# needed.  Display name is defined for this purpose.
attributetype ( 2.16.840.1.113730.3.1.241
   NAME 'displayName'
   DESC 'RFC2798: preferred name to be used when displaying entries'
   EQUALITY caseIgnoreMatch
   SUBSTR caseIgnoreSubstringsMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
   SINGLE-VALUE )

# employeeNumber
# Numeric or alphanumeric identifier assigned to a person, typically based
# on order of hire or association with an organization.  Single valued.
attributetype ( 2.16.840.1.113730.3.1.3
   NAME 'employeeNumber'
   DESC 'RFC2798: numerically identifies an employee within an
organization'
   EQUALITY caseIgnoreMatch
   SUBSTR caseIgnoreSubstringsMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
   SINGLE-VALUE )
```

FIGURA 4 - Alguns atributos de classe do esquema `inetOrgPerson`.
Fonte: RFC 2798.

Na FIG. 4 notam-se alguns dos atributos como: código do departamento (`departmentNumber`), nome para exibição (`displayName`) e identidade organizacional da pessoa (`employeeNumber`). Podem ser notadas ainda particularidades de cada atributo, como `EQUALITY`, que define que buscas na entidade representada pelo atributo serão sensíveis a tamanhos de caracteres - *case sensitives*[9], `SINGLE-VALUE`, que determina que o atributo receberá uma palavra única no armazenamento de sua informação etc.

[9] case sensitive: distinção entre a caixa de caracteres, sendo que eles podem ser caixa alta ou caixa baixa. (Nota do autor)

O exemplo a seguir descreve como o esquema implementa o objeto composto pelos seus atributos, que será posteriormente utilizado pelo LDAP:

```
# inetOrgPerson
# The inetOrgPerson represents people who are associated with an
# organization in some way.  It is a structural class and is derived
# from the organizationalPerson which is defined in X.521 [X521].
objectclass ( 2.16.840.1.113730.3.2.2
   NAME 'inetOrgPerson'
   DESC 'RFC2798: Internet Organizational Person'
   SUP organizationalPerson
   STRUCTURAL
   MAY ( audio $ businessCategory $ carLicense $ departmentNumber $
    displayName $ employeeNumber $ employeeType $ givenName $
    homePhone $ homePostalAddress $ initials $ jpegPhoto $
    labeledURI $ mail $ manager $ mobile $ o $ pager $ photo $
    roomNumber $ secretary $ uid $ userCertificate $
    x500uniqueIdentifier $ preferredLanguage $
    userSMIMECertificate $ userPKCS12 )
   )
```

FIGURA 5 - Definição do objeto de classe do esquema `inetOrgPerson`.
Fonte: RFC 2798.

Conforme pode ser visto na FIG. 5, o objeto de classe - `objectclass` - é um atributo especial que fornece informações sobre seu tipo de objeto - no caso, conforme já mencionado, um objeto contendo informações de uma pessoa associada a uma organização.

1.2.4 Arquivos LDIF

Segundo Howes *et al* (2003), arquivos LDIF (LDAP *Data Interchange Format*) são arquivos de texto definidos no RFC 2849 que servem para descrever as entradas de um diretório LDAP. Ainda, segundo Butcher (2007), o padrão LDIF não serve somente para representar o conteúdo de um diretório, mas também para representar certas operações no LDAP, como adições, alterações e exclusões.

Trigo (2007) afirma que esse formato é o único meio de entrada de dados em um diretório LDAP e que é a única forma possível de importar e exportar dados LDAP. A seguir, exemplos de arquivos LDIF e seus respectivos objetivos:

```
dn: cn=breno,dc=fea,dc=fumec,dc=br
cn: breno
objectClass: simpleSecurityObject
objectClass: organizationalRole
userPassword: {SSHA}thUlS1AJ+DOMXOI0iUpH4P4F2gCtCh7L
description: Usuário Breno
```

FIGURA 6 - Arquivo `insertUser.ldif`.
Fonte: Do autor.

```
dn: cn=breno,dc=fea,dc=fumec,dc=br
changetype: modify
replace: userPassword
userPassword: {SSHA}+DOMthUlF2gCtCh7LS1AJXOI0iUpH4P4
```

FIGURA 7 - Arquivo `updateUser.ldif`.
Fonte: Do autor.

1.3 Segurança

Tornar a informação segura é de grande importância em ambientes de rede, sobretudo na Internet. Ao trafegarem dados através da rede interna ou externa, informações sigilosas podem estar vulneráveis. Logo, servidores LDAP proveem mecanismos de proteção. (TUTTLE, *et al*, 2004)

Tuttle *et al* diz que a segurança abrange quatro aspectos[10]:

1. Autenticação: garantia de que a parte oposta (máquina ou pessoa) é realmente quem que diz ser;
2. Integridade: garantia de que a informação que chega é a que foi enviada;
3. Confidencialidade: Proteção na divulgação de informações por meio de criptografia de dados para aqueles que não são destinatários dessa informação;
4. Autorização: Garantia de que a parte está devidamente autorizada.

O protocolo LDAP, por padrão, envia e recebe mensagens textuais, o que implica em facilidade de configuração, manutenção e eficiência - esse último, em função da não necessidade de processamento no ato da codificação e decodificação da mensagem. Porém,

[10] TUTTLE, 2004, p. 77-78.

essas vantagens são verdadeiras desvantagens do ponto de vista de segurança. (BUTCHER, 2007, p. 153-154)

Em vistas a isso, as próximas seções apresentam como o LDAP trata a segurança no tráfego de informações.

1.3.1 Autenticação básica

Nesse caso, o mecanismo de segurança do LDAP é negociado em tempo de conexão entre o servidor e o cliente, assim como no protocolo HTTP. (TUTTLE *et al*, 2004)

Ao utilizar-se de autenticação básica, o cliente se identifica ao servidor através de uma `DN` e uma senha. Uma vez que o servidor autentica essas credenciais informadas, tem-se acesso às informações solicitadas. (BUTCHER, 2007)

1.3.2 *Simple Authentication and Security Layer* (SASL)

Desenvolvido por John Myers da Netscape, o SASL - protocolo descrito no RFC 2222 - é uma estrutura genérica que atua em protocolos orientados a conexão e que tem como principal finalidade a mediação entre diversos protocolos e sistemas de autenticação. (HOWES *et al*, 2003)

No SASL, o LDAP é tido como sendo uma extensão do protocolo SASL, o que possibilita trabalharem juntos; opcionalmente, uma camada de segurança pode ser negociada para criptografar dados após a autenticação e assim garantir a confidencialidade da informação. (TUTTLE *et al*, 2004)

Segundo Howes *et al* (2003, p. 429), embora o SASL não ofereça segurança, ele permite que protocolos de aplicação, como o LDAP, negociem parâmetros de segurança. É importante mencionar que a versão 3.0 do LDAP (LDAPv3) inclui suporte nativo ao SASL.

1.3.3 *Secure Socket Layer* (SSL) e *Transport Layer Security* (TLS)

Howes *et al* (2003, p. 429) apresentam as seguintes definições para os protocolos SSL e TLS:

SSL: Desenvolvido originalmente por Kipp Hickman da Netscape, o SSL é um mecanismo genérico projetado para tornar protocolos orientados a conexão, como o LDAP, seguros. É baseado em criptografia de chave pública e pode prover alto grau de segurança. Inclui autenticação reforçada, assinaturas e serviços de criptografia. O SSL permite que duas partes comunicantes negociem um nível de segurança apropriado. Além disso, uma variedade de algoritmos ou níveis de segurança pode ser negociada.

TLS: Com a padronização e o amadurecimento do SSL, que chegou à versão 3.0, o IETF o rebatizou com o nome de *Transport Layer Security* (TLS), lançando assim sua versão 1.0, documentada no RFC 2246.

Basicamente, há pouca diferença entre o SSL 3.0 e o TLS 1.0, destacando principalmente que TLS usa o algoritmo *keyed-Hashing for Message Authentication Code* (HMAC) enquanto o SSL usa apenas *Message Authentication Code* (MAC). Nesse caso, o algoritmo HMAC produz *hashes*[11] mais seguros que o algoritmo MAC.

Tuttle *et al* dizem que, originalmente, "o SSL foi concebido para fornecer tanto autenticação quanto segurança de dados. Ele encapsula *sockets*[12] TCP/IP de modo que, basicamente, toda aplicação TCP/IP pode utilizá-lo para tornar segura a sua comunicação"[13]. (2004, p. 79)

Tuttle *et al* (2004, p. 80) ainda dizem que o protocolo SSL/TLS provê autenticação mútua entre servidores e clientes. Além disso, provê privacidade dos dados enviados pela rede por meio de criptografia. Para isso, utiliza um modelo baseado em pares de chaves públicas / privadas. Assim, a comunicação entre um cliente e um servidor LDAP tem-se da seguinte forma:

---

[11] *Hash* é uma operação criptográfica onde dada uma mensagem, é executada sobre ela algumas operações matemáticas que resultará em uma sequência de caracteres alfanuméricos em um tamanho fixo (exemplo, 128 bits) e que provavelmente será única. (HASH In: **Linux Dictionary**. 2004. Disponível em: <http://tldp.org/LDP/Linux-Dictionary/html/h.html>. Acesso em: 29. out. 2010)

[12] Socket ou Soquete é um mecanismo de comunicação entre processos que serve como um ponto final da comunicação. Um soquete é uma combinação de um endereço IP do nó e o número da porta de TCP ou UDP. (SCRIMGER *et al*, 2000. P. 154)

[13] The Secure Socket Layer (SSL) protocol was devised to provide both authentication and data security. It encapsulates the TCP/IP socket so that basically every TCP/IP application can use it to secure its communication (tradução nossa)

1. O cliente requisita ao servidor uma sessão SSL/TLS, e nesse caso, inclui opções de SSL/TLS em seu pedido;
2. O servidor devolve ao cliente suas opções de SSL/TLS e um certificado que inclui, dentre outras coisas, uma chave pública do servidor, a identidade para quem o certificado foi emitido - como uma `DN` -, nome da certificadora e o prazo de validade desse certificado. Esse certificado depende de uma *Certificate Authority* ou Autoridade de Certificados (CA) para assinar as chaves a fim de saber se uma chave é verdadeira ou falsa;
3. Então, o cliente solicita ao servidor uma prova de sua identidade. Isso faz com que se tenha certeza de que o certificado não tenha sido enviado por alguém que tenha interceptado a mensagem na rede;
4. O servidor retorna uma mensagem incluindo um resumo de verificação, que é criptografado com sua chave privada. O cliente, por sua vez, descriptografa o resumo de verificação com a chave pública que lhe foi enviada pelo servidor. Se ambas as chaves forem idênticas, a identidade do servidor é provada e o processo de autenticação é concluído;
5. No próximo passo, o servidor e o cliente combinam uma chave simétrica que será utilizada para criptografar os dados. Os dados são criptografados por meio dessa chave. Então, o cliente gera uma chave simétrica, criptografa com a chave pública do servidor e envia os dados. Apenas o servidor pode descriptografar essa chave através de sua chave privada;
6. O servidor descriptografa a chave e envia de volta uma mensagem de teste criptografada com a chave secreta recebida, atestando assim a segurança na comunicação. A partir desse ponto, eles podem começam a trocar mensagens utilizando a chave simétrica gerada para criptografar os dados.

CAPÍTULO 2 - Implementações do LDAP

Como visto no Capítulo 1 deste estudo, o LDAP é um protocolo de Internet que possibilita uma padronização em serviços de diretórios e que atende a diversos fins em uma rede de computadores.

Por ter se tornado um padrão, diversas empresas de software fizeram implementações do LDAP, criando assim ferramentas de uso corporativo que proveem serviços de obtenção de informações a partir de diretórios no padrão X.500.

Este capítulo apresenta as duas principais implementações do LDAP. Assim, na seção 2.1 é apresentado o *Active Directory* (AD), que é uma implementação feita pela Microsoft e, em seguida, na seção 2.2, é apresentado o OpenLDAP, uma implementação de código aberto.

2.1 *Active Directory* (AD)

Criado pela empresa fundada por Bill Gates, a Microsoft *Corporation*, o *Active Directory* (AD), ou ainda *Active Directory Service* (ADS), veio como uma solução de serviço de diretórios implementada através do protocolo LDAP versus serviço de gerenciamento de usuários existente na versão NT do sistema operacional *Windows*. Essa solução não era parte integrante do sistema operacional *Windows* NT, no que veio a se integrar na versão 2000 do sistema operacional *Windows*. (SHARMA, 2010)

King (2000) ressalta que, assim como o *Windows* NT foi um sistema operacional projetado originalmente para superar as deficiências dos sistemas operacionais baseados em serviços de rede, o *Windows* 2000 *Server* com o ADS foi projetado para superar as fragilidades existentes em um ambiente baseado em domínio[14].

Antigamente, quando ainda não se existia o ADS, criavam-se vários servidores de domínio com o *Windows* NT nos diversos departamentos ou localidades de uma empresa, o que acarretava em possuir diversas bases de informações de usuários. Quando era preciso que um usuário de uma filial ou departamento trabalhasse em outro local, devia-se estabelecer

[14] A Microsoft define um Domínio como um agrupamento lógico de usuários e computadores gerenciados através de um banco de dados centralizado e compartilhado. (KING, 2000, p. 79)

uma conexão de confiança entre os servidores NT ou criava-se uma nova conta de usuário no estabelecimento em que se encontraria. Isso trazia um sério problema de gerenciamento para os administradores de rede devido ao alto grau de complexidade no gerenciamento das informações de usuários. (KING, 2000)

A FIG. 8 exemplifica como o *Windows* NT trabalhava a questão de sistemas baseadas em domínio:

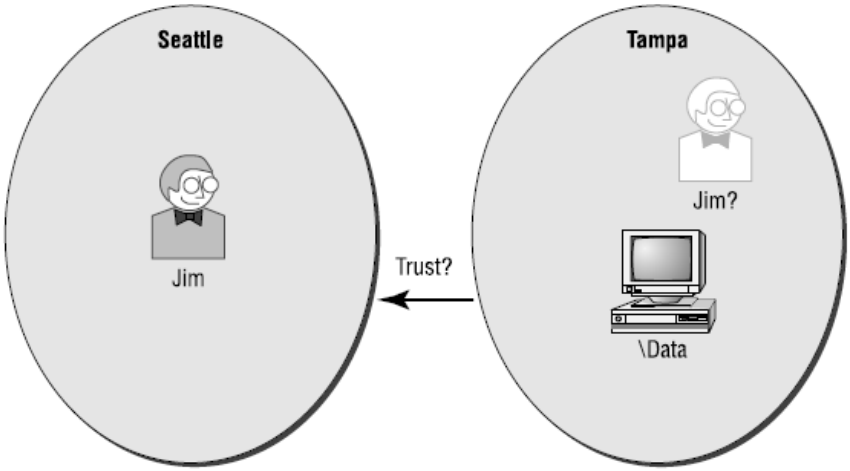


FIGURA 8 - Modelo de sistema baseado em Domínio.
Fonte: KING, 2000, p. 108.

No caso demonstrado, em sistemas baseados em domínio existem duas possíveis soluções para que *Jim* utilize um recurso em *Tampa*: ou é estabelecida uma relação de confiança entre os domínios de *Seattle* e *Tampa* ou se cria uma conta local no domínio em *Tampa* para *Jim.*

Com o surgimento do ADS, que tem como base o protocolo LDAP, essa complexidade diminuiu, pois permite que os administradores de rede gerenciem toda sua rede e todos os seus recursos associados de forma concisa e unificada, tudo isso através de uma série de ferramentas utilizadas para acessar as informações e configurações presentes no banco de dados do ADS. (KING, 2000)

A FIG. 9 mostra uma tela do *Active Directory* implementando em um servidor *Windows*.

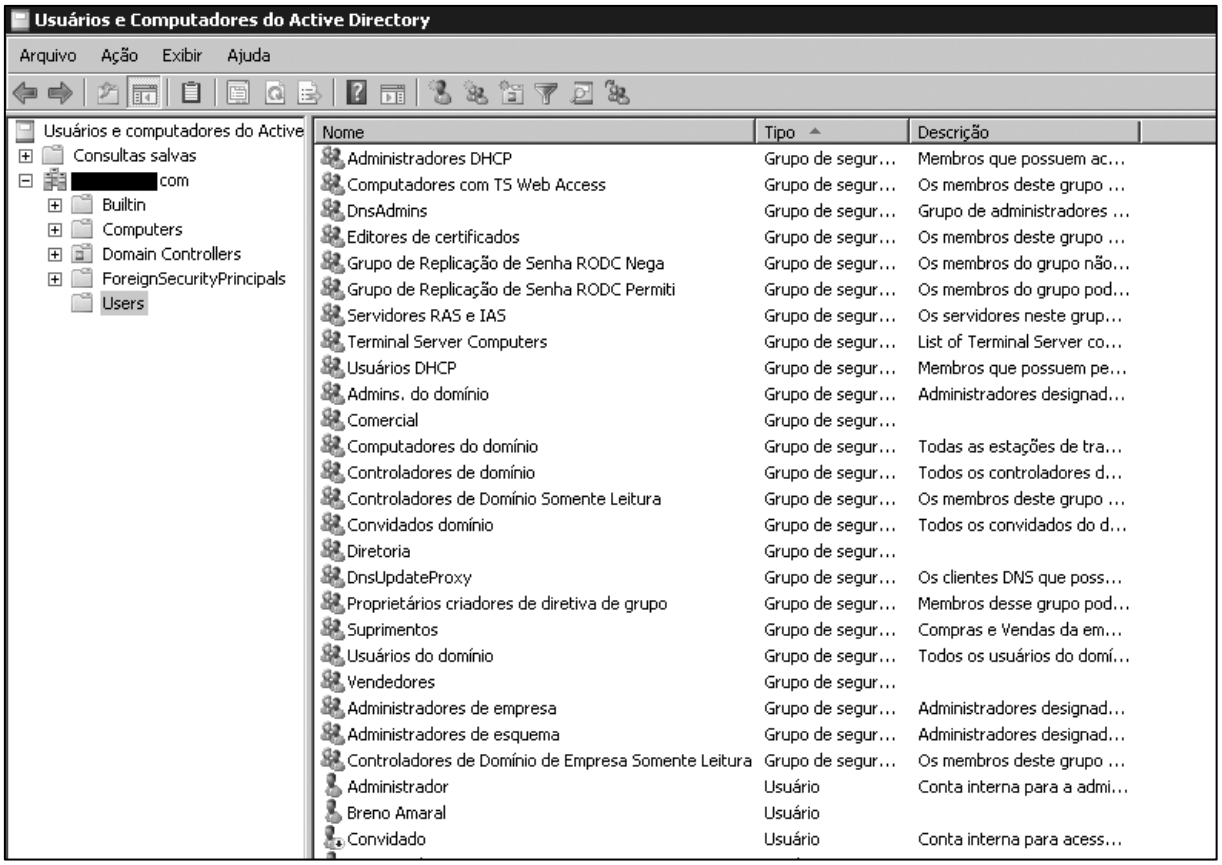


FIGURA 9 - Exemplo de implementação do *Active Directory*.
Fonte: Do autor.

2.1.1 *Active Directory Service* e o LDAP

Baseado no pré-requisito de se ter um modelo padrão para acessar e armazenar informações, com o ADS e a padronização feita pelo protocolo LDAP, a Microsoft definiu como as informações devem ser formatadas. Com isso, foi possível fazer com que diversos sistemas distintos na rede pudessem autenticar-se na base de dados do ADS ou até mesmo buscarem informações armazenadas em diretórios. (KING, 2000)

Com o ADS, cada objeto em um diretório possui um nome único para referenciá-lo, pois utiliza nomes no padrão X.500 para representar cada um dos objetos que ele contém. Como já visto, em ambientes X.500, o nome completo ou distinto (`DN`) de qualquer objeto representa o caminho completo para o topo da estrutura da árvore de diretórios, conforme pode ser observado na FIG. 10.

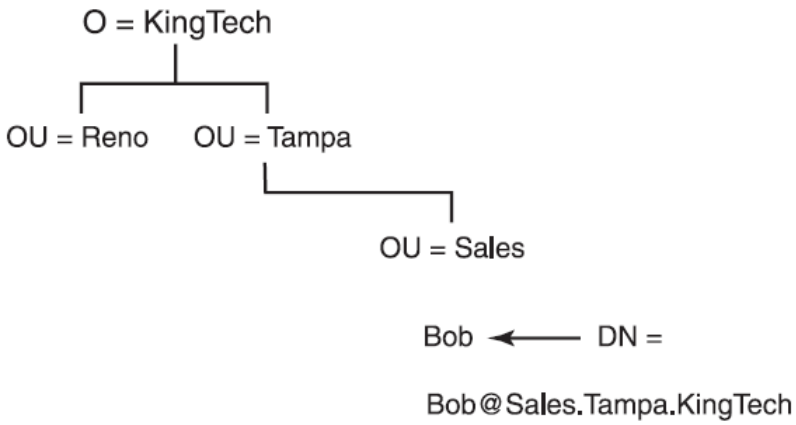


FIGURA 10 - Representação de um nome de objeto distinto em uma árvore de diretórios.
Fonte: KING, 2000, p. 121.

Essa estrutura hierárquica do LDAP possibilitou ao sistema operacional *Windows* e ao ADS trabalharem com uma base de dados global, não sendo mais necessário possuir diversos domínios em uma empresa. Como exemplo, observe a FIG. 11:

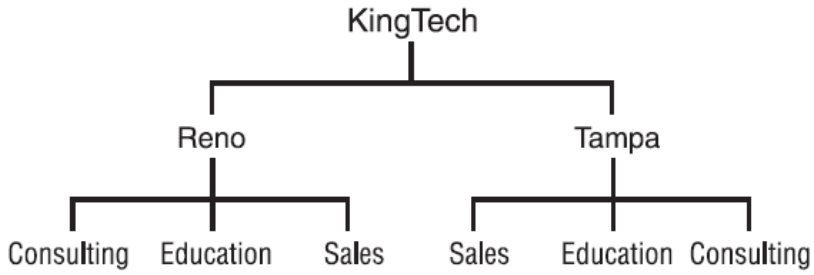


FIGURA 11 - Estrutura hierárquica da Empresa *KingTech*
Fonte: KING, 2000, p. 123.

Esse exemplo mostra que a empresa *KingTech* possui duas filiais: *Reno* e *Tampa*. Cada uma delas possui diversos departamentos (no caso, *Consulting*, *Education* e *Sales*). A integração LDAP + ADS possibilitou a existência de um domínio único representado pela organização (`o` - *organization*) *KingTech*, com duas unidades organizacionais (`ou` - *organization unit*), *Reno* e *Tampa*. Assim, temos as seguintes representações para cada uma das filiais:

1. Tampa.KingTech
2. Reno.KingTech

Com essa estrutura centralizada e organizada hierarquicamente, o caso demonstrado na FIG. 8 torna-se facilmente resolvível uma vez que não haverá mais a complexidade de vários domínios para uma única empresa, passando a existir apenas um

domínio central (no caso *KingTech*) e unidades organizacionais ligadas a ele (no caso, *Reno* e *Tampa*). Nesse caso os usuários estariam ligados a organização central, podendo ainda pertencer a uma ou várias unidades organizacionais.

Price *et al* (2008) dizem que o ADS pode utilizar o esquema `inetOrgPerson` como base de informações de pessoas. Isso trará a vantagem de se ter interoperabilidade com outros serviços de diretórios que implementem o protocolo LDAP, como o Novell eDirectory[15].

A FIG. 12 mostra uma captura de uma tela a partir do *Active Directory* que teve o uso do objeto `inetOrgPerson` como base na criação de um usuário:

FIGURA 12 - Tela de propriedades de uma conta criada a partir de um objeto `inetOrgPerson` no AD.
Fonte: Price *et al*, 2008, p. 146

[15] eDirectory: Atual nome para o serviço de diretórios da *Novell Inc*. e que era chamado NDS (*Novell Directory Service*). NDS/eDirectory utiliza-se do LDAP para implementar o gerenciamento de informações. (KUO, Peter; HENDERSON, Jim. **Novell's Guide to Trobleshooting eDirectory**. Novell Press, 2004.)

2.2 OpenLDAP

O OpenLDAP é um projeto de código aberto para implementação do protocolo LDAP iniciado em 1998 por Kurt Zeilenga. Algum tempo depois, Howard Chu, da Universidade de Michigan, juntou-se ao projeto. Hoje, em sua versão 2.4, o OpenLDAP está presente em quase todas as principais distribuições *Linux* existentes. (BUTCHER, 2007)

Atualmente, o OpenLDAP é mantido pela *OpenLDAP Foundation*. Essa fundação é responsável também por manter diversas ferramentas que se integram ao serviço provido pelo OpenLDAP.

O OpenLDAP pode ser implementado em diversos sistemas operacionais, entretanto, o projeto OpenLDAP não provê versões executáveis para esses sistemas, ficando a cargo dos mantenedores de cada um desses compilar e fornecer suas próprias fontes de executáveis. O projeto fornece versões já compiladas para a maioria dos sistemas operacionais baseados na plataforma UNIX, incluindo MAC OS X[16], HP-UX[17] e algumas variações do sistema BSD[18]. (OpenLDAP Fondation)

A FIG. 13 traz uma tela que mostra uma implementação do OpenLDAP:

[16] MAC OS X: Sistema operacional da *Apple* para computadores *Macintosh* e que possui núcleo UNIX. (**O que é o MAC OS X?**: Disponível em: < http://www.apple.com/pt/macosx/what-is-macosx>. Acesso em: 25. oiut. 2010)

[17] HP-UX: implementação proprietária do sistema operacional UNIX desenvolvido pela empresa de software HP (nota do autor)

[18] BSD ou *Berkeley Software Distribution* é um sistema operacional baseado em UNIX desenvolvido e distribuído pela Universidade da Califórnia – *Berkeley*. (***A Narrative History of BSD***: Disponível em: <http://extension.berkeley.edu/cat/course1556.html>. Acesso em: 16 out. 2010)

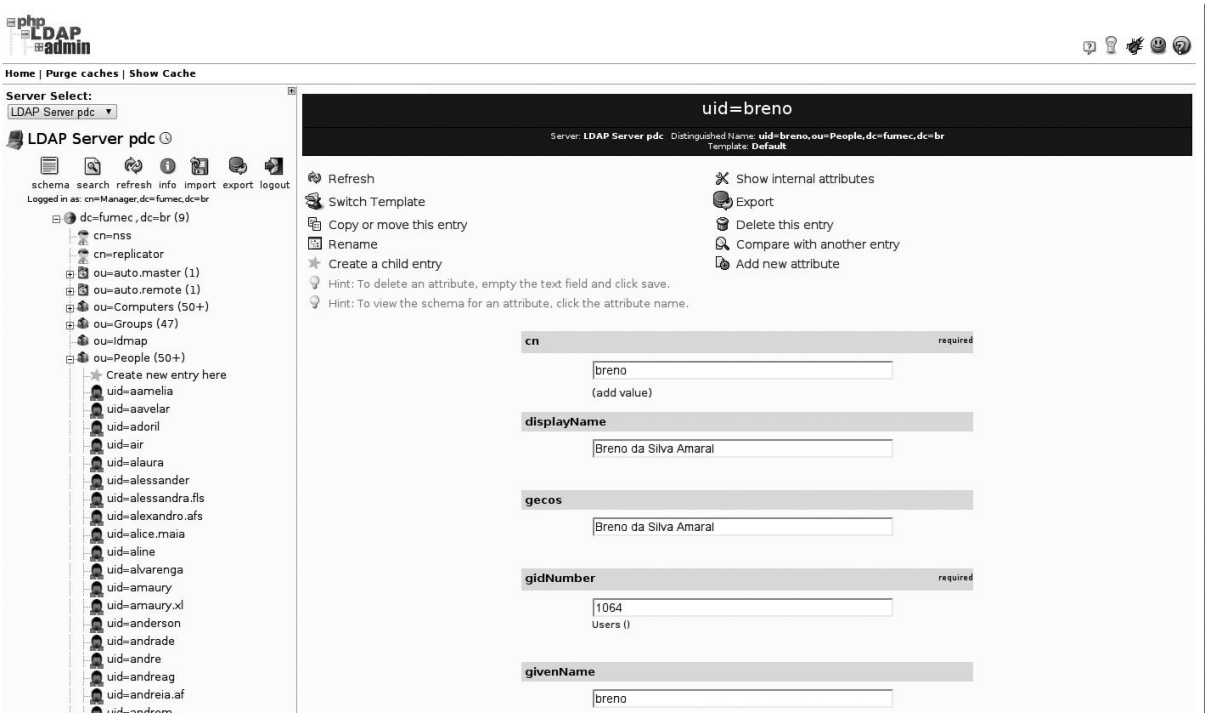


FIGURA 13 - Implementação do OpenLDAP.
Fonte: Do autor.

Trigo (2004, p. 30) destaca que as principais características do OpenLDAP são:

1. Suporte a IPv4 e IPv6;
2. Autenticação (Cryrus Sasl-Kerberos V, GSSAPI, Digest-MD5);
3. Segurança no transporte – SSL e TLS;
4. Controle de acessos;
5. Escolha entre bancos de dados (GDBM ou DBD);
6. Capacidade de atender a múltiplos bancos ao mesmo tempo;
7. Alto desempenho em múltiplas chamadas;
8. Replicação de base.

Butcher (2007, p. 21-22) aponta que a suíte OpenLDAP pode ser desmembrada em quatro componentes:

1. Servidor: Provê o serviço de LDAP;
2. Clientes: Manipula dados do LDAP;
3. Utilitários: Presta suporte a servidores LDAP;
4. Bibliotecas: Provê interfaces programáveis para LDAP.

Sendo assim, nas próximas sub-seções é apresentado um maior detalhamento sobre cada um desses componentes.

2.2.1 Servidor LDAP

No OpenLDAP, o servidor principal de LDAP chama-se **slapd** (*Standalone LDAP Daemon*) e através dele são permitidos acessos às árvores de diretórios de informações. Embora esse servidor de LDAP possua um *socket* ouvinte para estabelecimento de conexões, geralmente os clientes se conectam no servidor LDAP usando uma conexão de rede simples e o protocolo de rede LDAP. (grifo nosso)

O servidor **slapd** tem como função armazenar dados no diretório local ou prover acesso a fontes de dados externas. Tipicamente, além de oferecer serviços de autenticação e busca de informações em diretórios, também oferece suporte nas operações de adição, exclusão e modificação de dados em diretórios. Não menos importante, esse servidor é o responsável por oferecer controle de acesso detalhado ao diretório.

Exemplificando, em forma de diálogo entre pessoas, a ação de conexão de um cliente LDAP a um servidor **slapd** juntamente de uma solicitação qualquer, seria:

Cliente: *Servidor LDAP, quero autenticar-me*

Servidor: *Cliente, informe suas credenciais*

Cliente: *Autentique-me como usuário Breno e senha 12345*

Servidor: *Breno agora está autenticado*

Cliente: *Breno quer pesquisar todos os usuários onde haja a palavra "Terra" em seu sobrenome*

Servidor: *Existe um usuário cujo sobrenome possui a palavra "Terra": sua identificação é: ricardo.terra*

Cliente: *Desconecte-me*

Servidor: *Desconexão sucedida*

2.2.2 Cliente LDAP

Clientes LDAP normalmente se conectam em um servidor **slapd** através do protocolo de rede LDAP, solicitando a execução de diversas operações. Como já visto, pelo LDAP ser um protocolo que atua sobre a pilha TCP/IP, geralmente, um cliente estabelecerá uma conexão com o servidor de diretórios primeiramente. Estabelecida essa conexão, ele autenticar-se-á e, em seguida, executará as operações desejadas (pesquisas, adições, modificações, exclusões etc). Ao final, o cliente se desvincula do servidor e se desconecta.

Alguns comandos do OpenLDAP executados pelos clientes são:

1. `ldapadd`: utilizado para incluir informações na base do LDAP;
2. `ldapdelete`: utilizado para remover algum registro da base do LDAP;
3. `ldapmodify`: utilizado para alterar campos da base do LDAP;
4. `ldapmodrdn`: utilizado para renomear uma entrada do LDAP;
5. `ldappasswd`: utilizado para alterar a senha de um usuário presente na base do LDAP;
6. `ldapsearch`: utilizado para efetuar pesquisa na base do LDAP.

2.2.3 Utilitários do OpenLDAP

Diferentemente dos clientes, os utilitários do OpenLDAP não executam operações utilizando o protocolo LDAP. Ao contrário, eles manipulam os dados em um nível inferior e sem a mediação do servidor, ajudando assim a manter esse servidor.

A seguir, alguns comandos utilitários do OpenLDAP:

1. `slapadd`: utilizado para incluir dados na base do LDAP em modo *off-line;*
2. `slapcat`: utilizado para listar todo conteúdo da base de dados do LDAP. Pode ser utilizado também para criar arquivos LDIF com esse conteúdo, o que é útil quando se deseja efetuar *backup* em um arquivo;

3. `slapindex`: utilizado para recriar os índices da base de dados do LDAP. À medida que operações de adição e exclusão vão sendo executadas nessa base, é necessário re-indexar a base para que o retorno das pesquisas melhore;
4. `slaptest`: utilizado para testar a sintaxe do arquivo de configuração do servidor LDAP (`slapd.conf`);
5. `slapacl`: utilizado para testar o comportamento das regras de acesso (ACL ou *Access Control List*) criadas no arquivo de configuração do LDAP;
6. `slapauth`: utilizado para verificar o comportamento do servidor LDAP (slapd) quando mapeia uma identidade com propósito de autenticação e autorização;
7. `slapdn`: utilizado para verificar se uma entrada `DN` confere com os esquemas definidos no arquivo de configuração do servidor LDAP (`slapd.conf`);
8. `slappasswd`: utilitário para gerar senhas criptografadas.

2.2.4 Bibliotecas do OpenLDAP

O OpenLDAP disponibiliza diversas bibliotecas LDAP para serem utilizadas por diversos aplicativos. Essas bibliotecas disponibilizam funções de integração ao serviço de LDAP através de APIs (*Application Programming Interfaces*). Embora o OpenLDAP forneça APIs escritas na linguagem C, o projeto fornece também duas bibliotecas escritas na linguagem *Java*.

A seguir, algumas dessas bibliotecas fornecidas pelo OpenLDAP:

1. **JLDAP**: biblioteca escrita em *Java* projetada para fornecer acesso a serviço de diretórios LDAP. Define duas interfaces síncronas e assíncronas para o LDAP de forma a atender uma ampla variedade de aplicações.

2. **JDBCLDAP**: biblioteca escrita em *Java* que tem como função prover acesso a serviço de diretórios LDAP àqueles que preferem utilizar linguagem SQL[19] e JDBC[20].
3. **ldapc++**: biblioteca escrita na linguagem C++ que tem como função prover acesso a serviços de diretórios LDAP a partir de programas escritos nessa linguagem.

Além dessas bibliotecas, existem outras diversas escritas em várias linguagens, como *Perl*, *Python*, *Ruby*, PHP etc.

[19] SQL ou *Structured Query Language* (Linguagem de consulta estruturada) é uma linguagem de banco de dados desenvolvida para manipular dados em sistemas de banco de dados relacionais. (nota do autor)
[20] JDBC ou *Java Database Conectivity* é uma API desenvolvida em *Java* que provê suporte a conexão entre aplicações escritas nessa linguagem e bancos de dados relacionais baseados em SQL. (ORACLE. **Java SE Technologies**: Database. Disponível em: <http://java.sun.com/products/jdbc>. Acesso em: 16 out. 2010.)

CAPÍTULO 3 - Estudo de Caso

Este capítulo apresenta um estudo de caso desenvolvido para uma empresa fictícia denominada de *Foobar Ltda.*

Na seção 3.1, é apresentado um cenário inicial onde as informações referentes aos usuários e funcionários da empresa e que são utilizadas por diversos sistemas e serviços estão localizados em bancos de dados distintos e de forma redundante, necessitando, portanto, de uma reestruturação que faça com que esses dados sejam concentrados em um único sistema, garantindo a unicidade, a confiabilidade e centralização das informações, tornando-as acessíveis a quaisquer sistemas pretendidos pela empresa. Na seção 3.2, é apresentado o planejamento de uma solução viável e adequada para a situação supracitada. Na seção 3.3, é apresentada a implementação baseada no planejamento proposto e descrito na seção 3.2. Na seção 3.4, são apresentados testes que validam o sucesso da implementação proposta e executada.

3.1 Cenário Atual

A *Foobar ltda*. é uma empresa no ramo de Tecnologia da Informação que possui 35 (trinta e cinco) funcionários, sendo que todos têm à sua disposição um computador com sistema operacional *Windows* 7 para executarem suas devidas tarefas. Além disso, a empresa possui em seu parque tecnológico um servidor que hospeda diversos serviços e sistemas, dentre eles o sistema gerenciador de domínio e autenticação SaMBa[21] e sistema de Proxy[22] SQUID[23]. Para cada um desses sistemas existe um banco de dados com informações diversas dos usuários; ainda, esses bancos de dados não são necessariamente pertencentes a um único sistema gerenciador de banco de dados (SGBD).

[21] SaMBa: Software que provê os serviços de compartilhamento de arquivos e compartilhamento de impressoras em uma rede de computadores. ECKSTEIN, Robert *et al.* **Using Samba**: A File & Print Server for Heterogeneous Networks. O'Reilly, 2000, p. 2.

[22] Proxy é um servidor HTTP com características especiais de filtragem de pacotes que tipicamente são executados em uma máquina firewall. MARCELO, Antônio. **Squid**: Configurando o Proxy para Linux. Brasport, 2006, p. 3.

[23] SQUID é um Proxy para Linux criado a partir do projeto Harverst da ARPA. MARCELO, Antônio (2006, p. 3.)

Para os sistemas e serviços, cada usuário possui, além das informações inerentes e complementares a cada um desses sistemas e serviços, conta de usuário e senha de acesso a eles. Em alguns casos, essas informações diferem de um sistema para outro.

Os computadores estão configurados de forma a buscarem informações de contas de usuários através do sistema gerenciador de domínio e autenticação SaMBa. A base de dados desse serviço não se integra aos demais sistemas e serviços existentes na empresa.

Apesar de haver certo controle sobre os acessos a Internet por meio do serviço de Proxy SQUID, os usuários não precisam se autenticar para fazer uso desse serviço. Além disso, esse ambiente não permite controle nem descrição detalhada sobre os acessos de cada usuário.

Em vistas, nota-se claramente que essa infraestrutura computacional existente na empresa encontra-se em um estado de difícil gerenciamento. Logo, necessita de uma reestruturação que proporcione a centralização dessas diversas informações referentes aos usuários e que seja, além de acessível por todos os sistemas e serviços existentes, de fácil manutenibilidade e passiva de escalabilidade.

3.2 Planejamento

Para a confecção do planejamento de implementação, adotaremos o padrão de confecção de "Plano de Projeto de Software"[24] proposto por Ian Sommerville e largamente utilizado em engenharia de software, porém com algumas adaptações que visam atender projetos de infraestrutura computacional.

3.2.1 Introdução

Este projeto tem por objetivo propor uma solução viável e suficiente para os diversos problemas levantados na empresa *Foorbar Ltda.*, conforme descritos na seção 3.1 desta pesquisa monográfica.

[24] SOMMERVILLE, Ian. Engenharia de Software. São Paulo: Addison Wesley, 2004.

Considera-se como principal requisito a integração da autenticação entre os diversos serviços e sistemas disponíveis na empresa, destacando o serviço de autenticação de usuários através de um sistema gerenciador de domínio e autenticação e o serviço de *Proxy*. Esse requisito visa garantir a unicidade e a não redundância das informações relativas aos usuários, além da centralização das informações.

3.2.2 Requisitos necessários de software

Como solução viável e suficiente foi adotada a utilização de soluções *OpenSource*[25], destacando a solução de software OpenLDAP descrita na seção 2.2 desta pesquisa na implementação do protocolo LDAP.

O sistema operacional utilizado na implementação deste projeto será o Linux. Nesse caso, adotaremos como distribuição Linux para este estudo de caso o CentOS[26]; tal distribuição foi escolhida por ser robusta, estável, possuir uma comunidade muito ativa e disposta a ajudar na solução de problemas e, acima de tudo, possuir total integração com o software OpenLDAP.

Para o sistema gerenciador de domínio e autenticação (PDC[27]), utilizar-se-á a integração entre os aplicativos provedores de serviço OpenLDAP + SaMBa. Por não ser o foco desta pesquisa, não será detalhado o software SaMBa. Uma leve explicação dele pode ser encontrada na página principal do site do projeto (http://www.samba.org), onde diz que o SaMBa é o conjunto de programas do padrão Windows que visam a interoperabilidade com os sistemas operacionais Linux e Unix. Ainda, diz que o SaMBa provê serviço seguro, estável e rápido na solução de compartilhamento de arquivos e impressoras em uma rede de computadores.

---

[25] Opensource não significa apenas acesso ao código fonte. Os termos de distribuição de software de código aberto devem cumprir os seguintes critérios: redistribuição livre, código fonte aberto, trabalhos derivados, integridade do código fonte do autor, sem discriminação contra pessoas ou grupos, sem discriminação contra campos de trabalho, distribuição da licença, licença não deve ser específica para o produto, a licença não deve restringir outros sistemas, a licença deve ser tecnologicamente neutra. (Open Source Initiative. **The Open Source Definition (Annotated)**. Disponível em: <http://www.opensource.org/osd.html>. Acesso em: 15 nov. 2010)

[26] CentOS é distribuição Linux coorporativa derivada de fontes fornecidas gratuitamente ao público por uma empresa norte americana que comercializa uma distribuição Linux. (CentOS: The Community Enterprise Operation System. **CentOS Overview**. Disponível em: <http://www.centos.org>. Acesso em: 15 nov. 2010)

[27] Primary Domain Controller (PDC) ou Controlador de Domínio Primário. (PDC. In: **Linux Dictionary**. 2004. Disponível em: <http://tldp.org/LDP/Linux-Dictionary/html/p.html>. Acesso em: 15 nov. 2010)

Para o sistema de Proxy, utilizar-se-á a solução de software SQUID. Também não será detalhado esse software por não ser foco principal deste trabalho.

3.2.3 Estrutura Analítica do Projeto (EAP)

Para este projeto, tem-se como objetivo a implementação do protocolo LDAP na empresa *Foobar Ltda*. a fim de prover autenticação centralizada aos diversos sistemas e serviços providos. Assim, segue o planejamento das atividades como FIG. 14:

1. Verificar infraestrutura de servidores e capacidade de integração
    a. Verificar compatibilidade de Sistema Operacional do servidor.
    b. Verificar compatibilidade com sistema gerenciador de domínio e autenticação SaMBa (PDC);
    c. Verificar compatibilidade do Sistema Operacional das estações de trabalho com o sistema gerenciador de domínio e autenticação SaMBa (PDC);
    d. Verificar compatibilidade com sistema de *Proxy* SQUID.
2. Instalar e configurar OpenLDAP
    a. Instalar o software OpenLDAP e suas respectivas dependências;
    b. Configurar o OpenLDAP;
    c. Popular base de dados do OpenLDAP com os grupos e usuários de rede da empresa.
3. Integrar autenticação dos diversos serviços e sistemas
    a. Configurar sistema gerenciador de domínio e autenticação SaMBa (PDC) para integração ao OpenLDAP;
    b. Configurar sistema de *Proxy* Squid para integrar-se ao OpenLDAP;
    c. Configurar estações de trabalho para que busquem autenticação através do sistema gerenciador de domínio e autenticação SaMBa (PDC) + OpenLDAP;
    d. Configurar navegador da estações de trabalho para que busquem autenticação através do *Proxy* SQUID.
4. Testar e homologar a solução proposta

a. Testar configurações do sistema OpenLDAP;
b. Testar integração do sistema gerenciador de domínio e autenticação SaMBa com o sistema OpenLDAP;
c. Testar autenticação das estações de trabalho através de contas criadas no OpenLDAP;
d. Testar integração do sistema de *Proxy* SQUID com o sistema OpenLDAP.

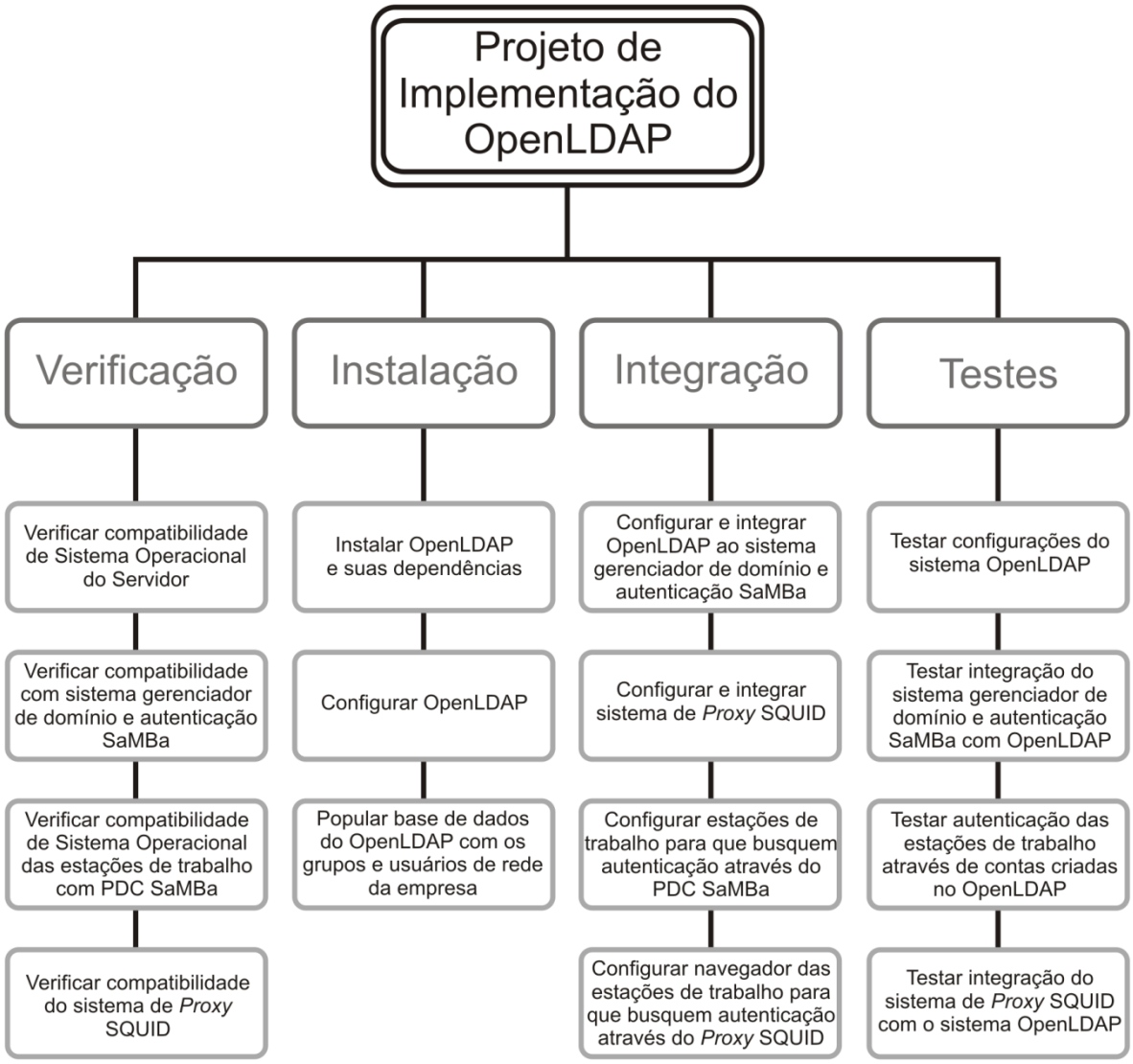


FIGURA 14 - Organograma das atividades planejadas para implementação do OpenLDAP.
Fonte: do autor.

3.3 Implementação

Como mencionado na seção 3.2.2, a distribuição Linux utilizada na implementação do OpenLDAP é o *CentOS*. Portanto, de acordo com o projeto, verifica-se a compatibilidade desse sistema operacional com os softwares necessários elencados no projeto.

Para efetuar a instalação do OpenLDAP e suas respectivas dependências no *CentOS* utilizaremos o gerenciador de pacotes dessa distribuição, denominado YUM, executa-se como a seguir:

```
[root@ldap ~]# yum install openldap-servers
```

Ainda, como esse software se integrará ao serviço SaMBa, que é totalmente compatível com o serviço do OpenLDAP, instalaremos o `smbldap-tools` (uma ferramenta que facilitará o trabalho na adição de usuários e grupos de usuários na base do OpenLDAP):

```
[root@ldap ~]# yum install smbldap-tools
```

O próximo passo é a configuração[28] do OpenLDAP. Para isso, edita-se o arquivo `/etc/openldap/slapd.conf` , primeiramente incluindo o módulo para integração ao serviço SaMBa:

```
include    /etc/openldap/schema/samba.schema
```

Em seguida, define-se a raiz de busca do LDAP e qual usuário administrará a base de informações de diretórios:

```
suffix      "dc=foorbar,dc=br"
rootdn      "cn=admin,dc=foorbar,dc=br"
```

O passo seguinte é a definição da senha desse usuário que administrará a base de informações de diretórios. Para isso, salva-se e fecha-se o arquivo `slapd.conf` e executa-se o seguinte comando:

[28] Todos os arquivos de configuração mencionados e utilizados durante o processo de implementação podem ser encontrados nos Anexos A, B, C, D e E desta monografia. (Nota do autor)

```
[root@ldap ~]# slappasswd
```

Será solicitado então para que digite a senha desejada. Desse modo, esta senha será criptografada e será retornado o seguinte resultado (caso a senha digitada seja “123456”):

```
{SSHA}XDR5xidsuWz2sBUrZp2kHFHEcdyK6cZt
```

De posse desta chave, edite novamente o arquivo `slapd.conf` e adicione a cadeia de caracteres retornada pelo comando `slappasswd`, na seguinte linha:

```
rootpw    {SSHA}XDR5xidsuWz2sBUrZp2kHFHEcdyK6cZt
```

Copie o arquivo `/etc/openldap/DB_CONFIG.example` para dentro do diretório `/var/lib/ldap` com o nome de `DB_CONFIG`, conforme comando abaixo:

```
[root@ldap ~]# cp /etc/openldap/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
```

Realizados todos os processos supracitados, inicia-se o serviço do OpenLDAP a fim de verificar se todas as configurações funcionam corretamente. Para isso, executa-se o comando:

```
[root@ldap ~]# service ldap start
Checking configuration files for slapd: config file testing succeeded
                                                           [   OK   ]
Starting slapd:                                            [   OK   ]
```

Desse modo, basta fazer as configurações iniciais do serviço SaMBa e configurar o `smbldap-tools` para que acesse a base de informações de diretórios do OpenLDAP. Para isso, edita-se primeiramente o arquivo `/etc/samba/smb.conf` e em seguida os arquivos `smbldap_bind.conf` e `smbldap.conf` que se encontram no diretório `/etc/smbldap-tools`. Antes desse passo, verificar-se-á através do comando `net getlocalsid`, qual é a chave de segurança gerada pelo LDAP no servidor. Esta chave é utilizada na configuração do arquivo `smbldap.conf`.

```
[root@ldap ~]# net getlocalsid
SID for domain LDAP is: SID=”S-1-5-21-2621065664-1578420217-34586684127”
```

As FIG. 15 a FIG.17 apresentam os arquivos de configuração supracitados.

```
57 [global]
58
59 # ----------------------- Network Related Options -------------------------
60 #
61 # workgroup = NT-Domain-Name or Workgroup-Name, eg: MIDEARTH
62 #
63 # server string is the equivalent of the NT Description field
64 #
65 # netbios name can be used to specify a server name not tied to the hostname
66 #
67 # Interfaces lets you configure Samba to use multiple interfaces
68 # If you have multiple network interfaces then you can list the ones
69 # you want to listen on (never omit localhost)
70 #
71 # Hosts Allow/Hosts Deny lets you restrict who can connect, and you can
72 # specifiy it as a per share option as well
73 #
74         workgroup = foobar.com
75         server string = Foobar Domain Server
76
77 ;       netbios name = MYSERVER
```

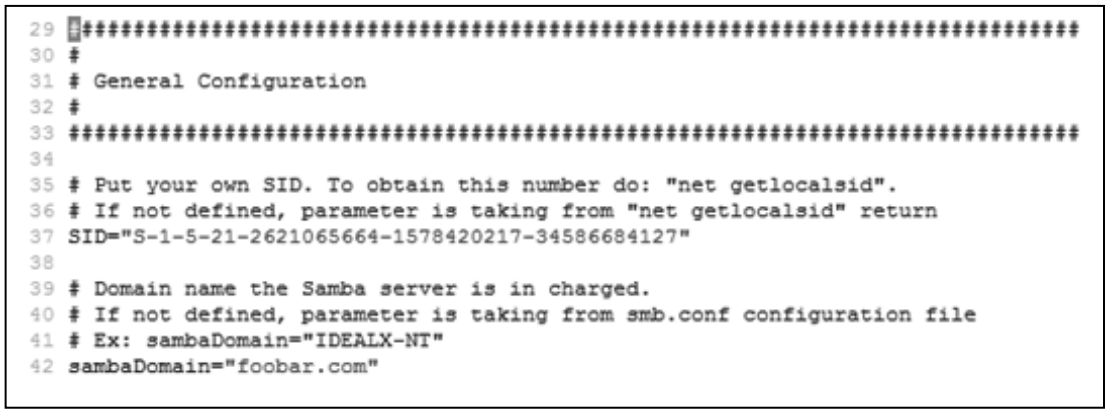
FIGURA 15 - Parte do arquivo de configuração `smb.conf`
Fonte: Do autor.

Observe que a linha 74 do arquivo apresenta o nome do domínio (no caso, foorbar.com) e a Linha 75 apresenta uma descrição textual desse domínio.

```
 1 ############################
 2 # Credential Configuration #
 3 ############################
 4 # Notes: you can specify two differents configuration if you use a
 5 # master ldap for writing access and a slave ldap server for reading access
 6 # By default, we will use the same DN (so it will work for standard Samba
 7 # release)
 8 slaveDN="cn=admin,dc=foobar,dc=com"
 9 slavePw="123456"
10 masterDN="cn=admin,dc=foobar,dc=com"
11 masterPw="123456"
12
```

FIGURA 16 - Arquivo de configuração `smbldap_bind.conf`
Fonte: Do autor.

Em análise, a linha 10 do arquivo apresenta a identificação da raiz de diretórios LDAP (`DN`) do servidor primário junto do nome do usuário que irá administrá-lo. A linha 11 apresenta a senha desse usuário. No caso desta implementação, como o servidor é único, serão usadas as mesmas informações para o servidor secundário (ou servidor escravo), conforme pode ser visto nas linhas 10 e 11.



```
29 ##############################################################################
30 #
31 # General Configuration
32 #
33 ##############################################################################
34
35 # Put your own SID. To obtain this number do: "net getlocalsid".
36 # If not defined, parameter is taking from "net getlocalsid" return
37 SID="S-1-5-21-2621065664-1578420217-34586684127"
38
39 # Domain name the Samba server is in charged.
40 # If not defined, parameter is taking from smb.conf configuration file
41 # Ex: sambaDomain="IDEALX-NT"
42 sambaDomain="foobar.com"
```

FIGURA 17 - Parte do arquivo de configuração `smbldap.conf`
Fonte: Do autor.

Observe que a linha 37 do arquivo deve conter a chave de segurança gerada pelo LDAP e a linha 42 o nome do domínio registrado no arquivo `smb.conf`, conforme mostra a FIG. 15.

Com os arquivos configurados, o próximo passo é popular a base de informações e diretórios do LDAP com os dados de configuração iniciais. Para isso, utilizar-se-á o seguinte comando que gerará a seguinte resposta:

```
[root@ldap ~]# smbldap-populate
Populating LDAP directory for domain foobar.com (S-1-5-21-2621065664-
1578420217-34586684127)
(using builtin directory structure)

adding new entry: dc=foobar,dc=com
adding new entry: ou=People,dc=foobar,dc=com
adding new entry: ou=Group,dc=foobar,dc=com
adding new entry: ou=Computers,dc=foobar,dc=com
adding new entry: ou=Idmap,dc=foobar,dc=com
adding new entry: uid=root,ou=People,dc=foobar,dc=com
adding new entry: uid=nobody,ou=People,dc=foobar,dc=com
adding new entry: cn=Domain Admins,ou=Group,dc=foobar,dc=com
adding new entry: cn=Domain Users,ou=Group,dc=foobar,dc=com
adding new entry: cn=Domain Guests,ou=Group,dc=foobar,dc=com
adding new entry: cn=Domain Computers,ou=Group,dc=foobar,dc=com
adding new entry: cn=Administrators,ou=Group,dc=foobar,dc=com
adding new entry: cn=Account Operators,ou=Group,dc=foobar,dc=com
adding new entry: cn=Print Operators,ou=Group,dc=foobar,dc=com
adding new entry: cn=Backup Operators,ou=Group,dc=foobar,dc=com
adding new entry: cn=Replicators,ou=Group,dc=foobar,dc=com
adding new entry: sambaDomainName=foobar.com,dc=foobar,dc=com
```

Uma vez que a base de informações de diretórios do LDAP já foi povoada com as informações básicas necessárias, o serviço estará pronto para receber os diversos cadastros de

grupos e usuários que se fizerem necessário. Sendo assim, para adicionar um grupo na base de informações de diretórios providos pelo OpenLDAP, utiliza-se o seguinte comando:

```
[root@ldap ~]# smbldap-groupadd –a cpd
```

donde "cpd" é o nome do grupo e o parâmetro "-a" indica que o grupo será automaticamente mapeado no domínio de autenticação do SaMBa. Para testar se o grupo foi inserido corretamente e verificar as suas informações relativas utiliza-se o comando e recebe-se a seguinte resposta:

```
[root@ldap ~]# smbldap-groupshow cpd
dn: cn=cpd,ou=Group,dc=foobar,dc=com
objectClass: top,posixGroup,sambaGroupMapping
cn: cpd
gidNumber: 1001
sambaSID: S-1-5-21-2621065664-1578420217-34586684127-3003
sambaGroupType: 2
displayName: cpd
```

Uma vez que o grupo foi criado, é plausível a criação de um usuário. Para isso, utilizar-se-á o comando a seguir:

```
[root@ldap ~]# smbldap-useradd -c "Breno da Silva Amaral" -a -g cpd -G 512
-d /home/bsa -m -D "U:" -E "cpd.bat" bsa
```

Foi definida para este usuário a identificação "bsa". Nota-se que alguns parâmetros foram adicionados ao comando. Para os citados acima, tem-se os seguintes significados:

-c : Nome completo do usuário

-a : Indica que este usuário é um usuário Windows

-g : Grupo primário do usuário

-G : Grupo suplementar do usuário

-d : Diretório de arquivos do usuário

-m : Cria o diretório do usuário automaticamente

-D : Associa uma letra a unidade referente ao diretório do usuário

-E : Define qual será o *script* de *login* do usuário

A fim de verificar se o usuário foi inserido corretamente, utilizar-se-á o seguinte comando obtendo-se a seguinte resposta:

```
[root@ldap ~]# smbldap-usershow bsa
dn: uid=bsa,ou=People,dc=foobar,dc=com
objectClass:
top,person,organizationalPerson,inetOrgPerson,posixAccount,shadowAccount,sa
mbaSamAccount
cn: bsa
sn: bsa
givenName: bsa
uid: bsa
uidNumber: 1002
gidNumber: 1001
homeDirectory: /home/bsa
loginShell: /bin/false
gecos: Breno da Silva Amaral
userPassword: {crypt}x
sambaPwdLastSet: 0
sambaLogonTime: 0
sambaLogoffTime: 2147483647
sambaKickoffTime: 2147483647
sambaPwdCanChange: 0
sambaPwdMustChange: 2147483647
displayName: bsa
sambaAcctFlags: [UX]
sambaSID: S-1-5-21-2621065664-1578420217-34586684127-3004
sambaLMPassword: XXX
sambaPrimaryGroupSID: S-1-5-21-2621065664-1578420217-34586684127-3003
sambaNTPassword: XXX
sambaLogonScript: cpd.bat
sambaHomeDrive: U:
```

Feitos os processos descritos durante a fase de implementação, deve-se adicionar e configurar junto ao sistema operacional Linux os módulos de autenticação integrados ao LDAP. No caso do CentOS, utiliza-se o seguinte comando cujo resultado pode ser observado nas FIG18 e FIG19:

```
[root@ldap ~]# authconfig-tui
```

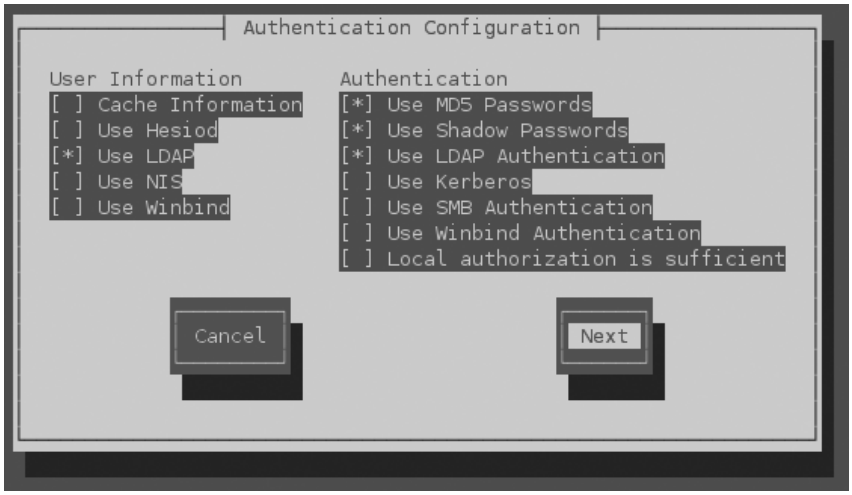


FIGURA 18 - Tela de configuração dos parâmetros de autenticação do *CentOS* que integra ao Sistema Operacional os módulos de autenticação do LDAP.
Fonte: do autor

Observe que foram marcadas as opções "*Use* LDAP" e "*Use* LDAP *Authentication*". Essas opções são importantes para que o sistema operacional Linux ou Unix reconheça as credenciais do LDAP como sendo válidas para autenticação no próprio sistema.

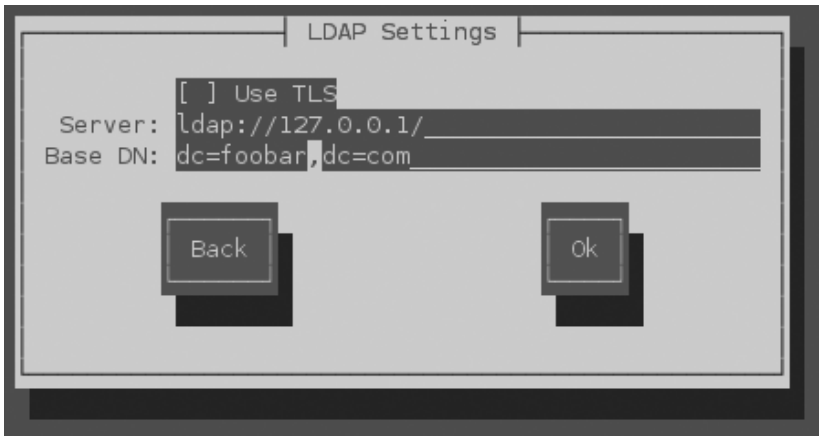


FIGURA 19 - Tela de configuração posterior a FIG.15 dos parâmetros de autenticação do *CentOS* que integra ao Sistema Operacional os módulos de autenticação do LDAP.
Fonte: do autor

Observe que na opção "*Server*" foi inserido o endereço local do servidor, pois como é sabido, para este estudo de caso tem-se apenas 1 (um) servidor. Ainda, na opção "*Base* DN", foi definida a raiz de pesquisa (`DN`) do LDAP (no caso, `dc=foobar,dc=com`).

Com isso, tem-se o LDAP implementado por meio do software OpenLDAP. Resta agora a integração dos demais serviços providos (SaMBa e SQUID) na empresa a esse software.

Primeiramente, será integrado o serviço gerenciador de domínio e autenticação SaMBa uma vez que esse processo já foi iniciado em função da configuração do `smbldap-tools`.

Para isso, edita-se novamente o arquivo `smb.conf` localizado no diretório `/etc/samba` e adiciona-se as seguintes linhas:

```
ldap admin dn = cn=admin,dc=foobar,dc=com
ldap user suffix = ou=People
ldap group suffix = ou=Groups
ldap idmap suffix = ou=Idmap
ldap machine suffix = ou=Computers
ldap passwd sync = Yes
ldap ssl = Off
ldap suffix = dc=foobar,dc=com
passdb backend = ldapsam:ldap://127.0.0.1
add machine script = /usr/sbin/smbldap-useradd –t 0 –w "%u"
```

Adicionadas as linhas apresentadas, reinicie o serviço do SaMBa com o comando a seguir, obtendo ainda a seguinte resposta:

```
[root@ldap ~]# service smb restart && service nmb restart
Shutting down SMB services:         [  OK  ]
Starting SMB services:              [  OK  ]
Shutting down NMB services:         [  OK  ]
Starting NMB services:              [  OK  ]
```

Feitas as devidas configurações mencionadas, encontram-se integrados os serviços SaMBa e OpenLDAP.

Agora, integrar-se-á o serviço de *Proxy* SQUID. Para isso, edita-se o arquivo `squid.conf` situado no diretório `/etc/squid`.

O primeiro passo é definir os parâmetros de autenticação do SQUID adicionando ao arquivo mencionado as seguintes linhas:

```
auth_param basic program /usr/lib/squid/squid_ldap_auth –b
"ou=People,dc=foobar,dc=com" –v3 –H ldap://127.0.0.1/
auth_param basic children 5
auth_param basic realm Entre com suas credenciais (as mesmas usadas para
fazer login):
auth_param basic credentialsttl 2 hour
auth_param basic casesensitive off
```

O próximo passo é definir uma regra que force o uso das credenciais de acesso do usuário de forma a permitir a navegação na Internet. Para isso, adicione as seguintes linhas no arquivo de configuração do SQUID:

```
acl password proxy_auth REQUIRED
http_acccess allow password
```

Feito isso, encontram-se integrados os serviços de *Proxy* SQUID e OpenLDAP.

3.4 Testes

Uma vez que o OpenLDAP foi implementado e integrado a outros serviços providos pela rede da empresa, deverão ser executados testes a fim de garantir o sucesso dessa implementação e o cumprimento dos requisitos.

Sendo assim, primeiramente será testado a integração do OpenLDAP com o serviço gerenciador de domínio e autenticação SaMBa. Para isso, através da conta criada para o usuário de rede “Breno da Silva Amaral”, identificado pelo *login* “bsa”, executa-se o seguinte comando e obtem-se a seguinte resposta:

```
[root@ldap ~]# smbclient -L localhost -U bsa
Enter bsa's password:
Domain=[FOOBAR.COM] OS=[Unix] Server=[Samba 3.3.8-0.52.el5_5.2]

Sharename       Type      Comment
---------       ----      -------
IPC$             IPC       IPC Service (Foobar Domain Server Version 3.3.8-
0.52.el5_5.2)
bsa             Disk      Home Directories
Domain=[FOOBAR.COM] OS=[Unix] Server=[Samba 3.3.8-0.52.el5_5.2]

Server               Comment
---------            -------

Workgroup            Master
---------            -------
```

Conforme pode ser visto, o teste foi realizado com sucesso, pois foram trazidas as informações relativas ao domínio do SaMBa utilizando uma conta criada na base de diretórios do LDAP.

O próximo teste a ser executado é o ingresso das estações de trabalho da empresa ao domínio “FOOBAR.COM” e a autenticação nesse domínio utilizando uma conta LDAP. Desse modo, seguem os procedimentos e as imagens relativas (FIG20 a FIG22):

Coloca-se a máquina cliente no domínio “FOOBAR.COM”. Será pedida a credencial administrativa do serviço para validar o ingresso da estação. Uma mensagem de boas vindas será emitida em caso de sucesso nesse processo.

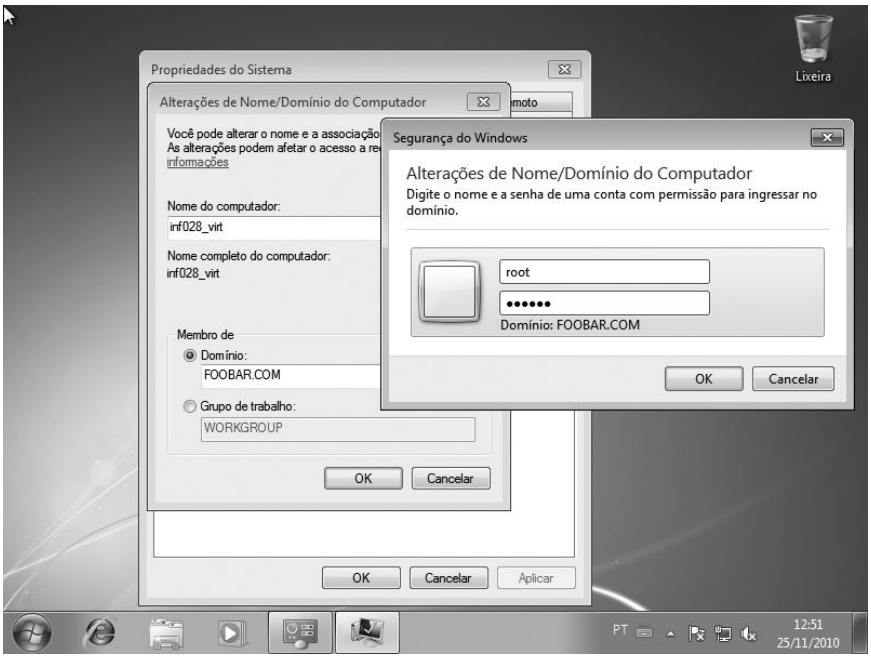


FIGURA 20 - Inserção de estação cliente no domínio FOOBAR.COM
Fonte: Do autor.

Observa-se que na FIG. 20 que foi registrado no campo “Domínio” o domínio `FOOBAR.COM`. Com isso, foram solicitadas as credenciais administrativas do servidor de LDAP para validação e inserção da máquina na base de diretórios de informações. A FIG. 21 a seguir mostra que as informações inseridas foram validadas e que a máquina foi registrada no domínio especificado.

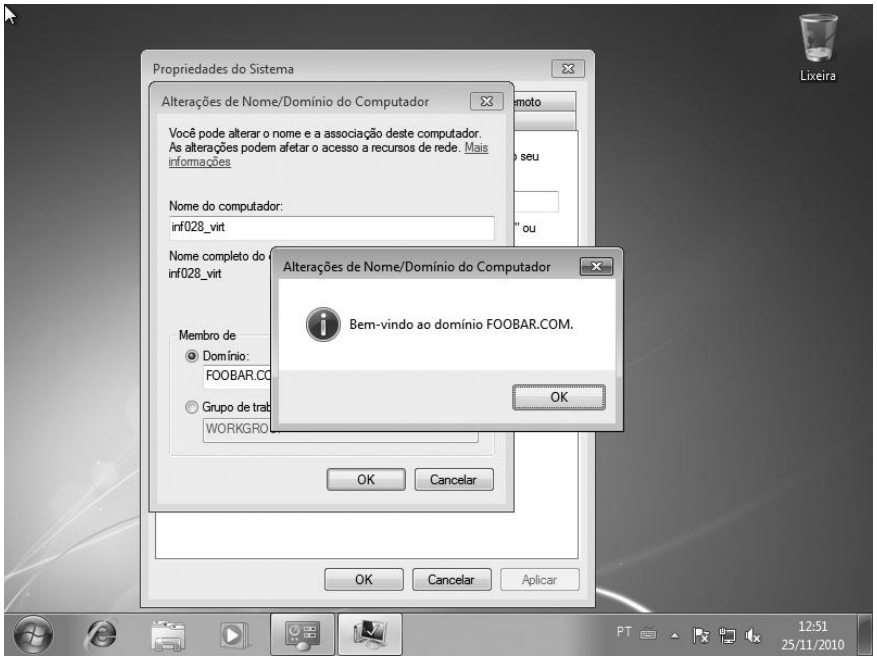


FIGURA 21 - Confirmação de sucesso no ingresso ao domínio FOOBAR.COM
Fonte: Do autor.

A seguir, mensagens de LOG do servidor que confirmam o sucesso da operação. Conforme pode ser visto, o serviço de gerenciamento de domínio e autenticação SaMBa adicionou o nome da máquina (`inf028_virt`) à base de diretórios de informações do LDAP.

```
[2010/11/25 12:52:29,  0]
passdb/pdb_interface.c:pdb_default_create_user(342)
   _samr_create_user: Running the command `/usr/sbin/smbldap-useradd -t 0 -w
"inf028_virt$"' gave 126
```

Ao reiniciar o computador, será pedido para que entre com alguma credencial de autenticação válida para o domínio. Trata-se das credenciais adicionadas na base de informações e diretórios do LDAP. Como mostrado durante a fase de implementação, foi criado um usuário identificado pela credencial "`bsa`". Para a realização dos testes de autenticação das estações de trabalho, será utilizada essa conta nesse processo.

FIGURA 22 - Pedido de credenciais para autenticação na estação de trabalho.
Fonte: Do autor.

A seguir, mensagens de LOG do servidor que confirmam o sucesso da operação. Conforme pode ser visto, o serviço de gerenciamento de domínio e autenticação SaMBa registrou o acesso à estação "`inf028_virt`" pelo usuário "`bsa`" no dia 25/11/2010 às 14:21:30 horas.

```
[2010/11/25 14:21:30,  1] smbd/service.c:make_connection_snum(1119)
  inf028_virt (::ffff:192.168.122.194) connect to service bsa initially as
user bsa (uid=1002, gid=1001) (pid 3125)
[2010/11/25 14:21:31,  1] smbd/service.c:make_connection_snum(1119)
  inf028_virt (::ffff:192.168.122.194) connect to service netlogon
initially as user bsa (uid=1002, gid=1001) (pid 3125)
[2010/11/25 14:22:10,  1] smbd/service.c:close_cnum(1331)
  inf028_virt (::ffff:192.168.122.194) closed connection to service
netlogon
```

A FIG. 24 demonstra a área de trabalho da estação após a autenticação do usuário "`bsa`". Detalhe para as informações "Local da rede" e "Domínio: FOOBAR.COM".

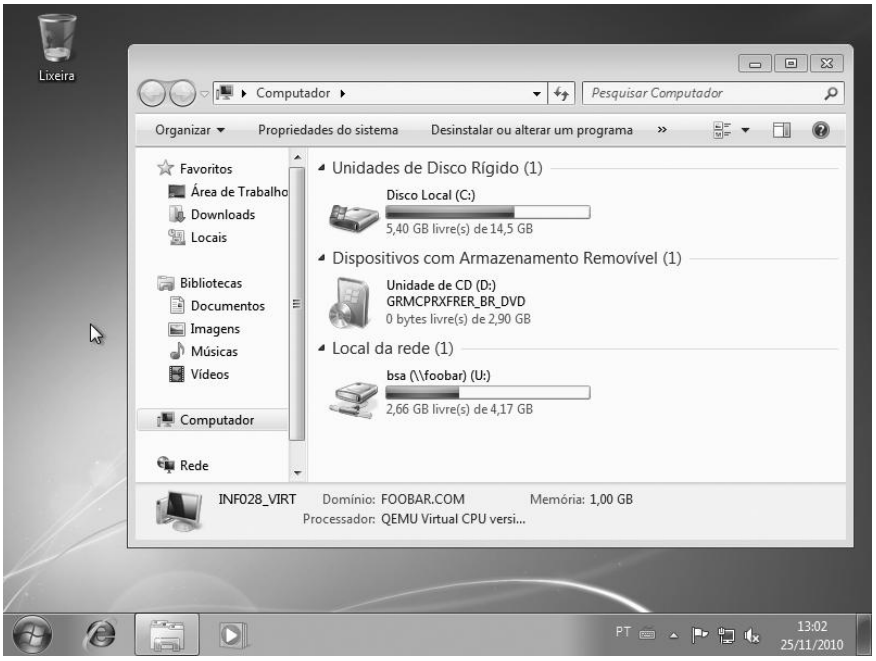


FIGURA 23 - Área de trabalho do usuário `bsa` após autenticação na estação de trabalho.
Fonte: Do autor.

Com isso, conclui-se com sucesso os testes de autenticação das estações de trabalho através de contas criadas na base de informações e diretórios do LDAP.

O próximo teste a ser feito é a integração do serviço de Proxy SQUID ao LDAP. Para isso, inicialmente será configurado o endereço de Proxy nos navegadores das estações. Como demonstração, será configurado o navegador "Internet *Explorer*" conforme FIG. 22:

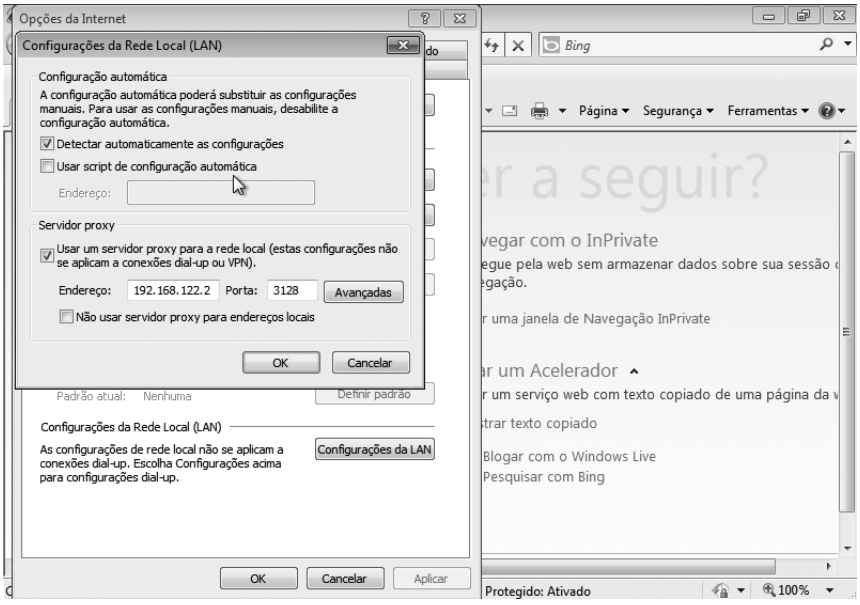


FIGURA 24 - Configuração de Proxy no navegador Internet Explorer.
Fonte: Do autor.

Observa-se na FIG. 25 que a opção “Usar um servidor de Proxy para a rede local...” foi marcada, o endereço do servidor de Proxy (no caso, `192.168.122.2`) e sua respectiva porta (no caso, `3128`) foram informados. Estas informações são importantes para que as requisições de páginas que saem das estações de trabalho passem pelo servidor Proxy a fim de validar as informações de contas de usuários antes de irem para a Internet.

Uma vez configurado o navegador, digita-se um endereço de uma página web qualquer a fim de verificar se será solicitada a entrada de credenciais LDAP para autorizar a navegação. Mais uma vez será utilizada a conta de usuário LDAP `bsa` para validar essas credenciais. A FIG23 a seguir demonstra esse processo.

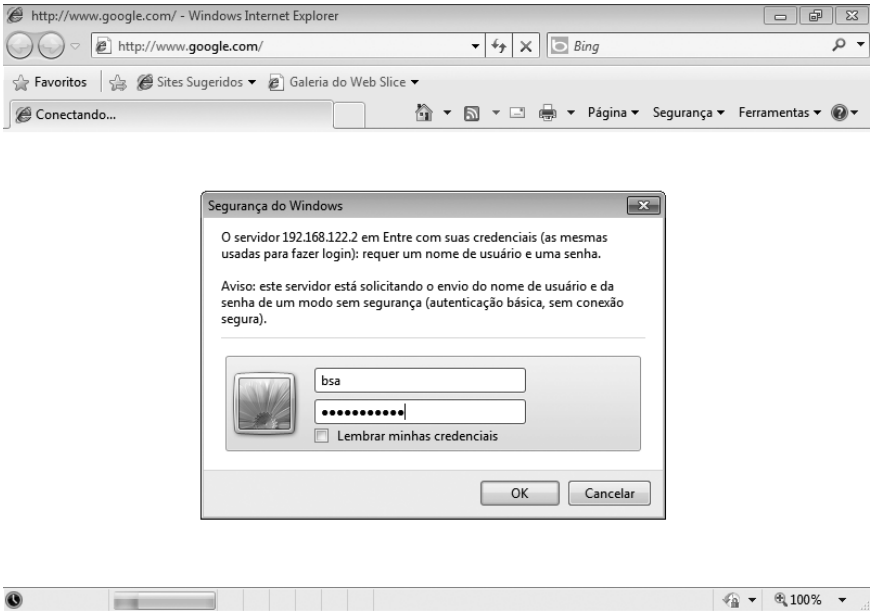


FIGURA 25 - Pedido de credenciais LDAP pelo serviço de Proxy no navegador de Internet.
Fonte: Do autor.

Caso as credenciais sejam válidas, tem-se o acesso à Internet liberado.

A seguir, mensagens de LOG do serviço de Proxy SQUID no servidor que confirmam o sucesso da operação. Conforme pode ser visto, o serviço de Proxy SQUID registrou o acesso à página “`http://www.google.com`” pelo usuário `bsa`.

```
1290697575.627    367 192.168.122.194 TCP_MISS/302 791 GET
http://www.google.com/ bsa DIRECT/64.233.163.104 text/html
1290697575.868    240 192.168.122.194 TCP_MISS/200 7435 GET
http://www.google.com.br/ bsa DIRECT/64.233.163.104 text/html
1290697576.397    112 192.168.122.194 TCP_NEGATIVE_HIT/204 287 GET
http://clients1.google.com.br/generate_204 bsa NONE/- text/html
1290697576.639    242 192.168.122.194 TCP_MISS/204 369 GET
http://www.google.com.br/csi? bsa DIRECT/64.233.163.104 text/html
```

CONCLUSÃO

Juntamente com o uso massivo das redes de computadores nas corporações veio também o uso de diversos sistemas e serviços computacionais. Na medida em que informações de funcionários/usuários/clientes da empresa eram armazenadas, viu-se que era necessário um ambiente que propiciasse além da não redundância desses dados, a centralização e uma estrutura que se integrasse a esses diversos sistemas e serviços providos pela empresa. Com o intuito de atender a esse requisito, um protocolo de Internet foi desenvolvido; o LDAP ou *Lightweight Directory Access Protocol*. Dessa forma, nesta monografia foi abordado o uso do protocolo LDAP como solução em centralização e disponibilidade de informações.

O protocolo LDAP traz em sua arquitetura uma estrutura de organização hierárquica que permite buscas em seus diretórios de informações de maneira eficiente, eficaz e segura. Ainda, por trabalhar sobre a arquitetura TCP/IP, o protocolo LDAP possui a vantagem da garantia de entrega de pacotes de mensagens livre de erros, ou seja, sempre que solicitada uma informação existente, chegará ao seu destino de forma concisa.

Por ser um protocolo de código aberto, em nível de informações que podem ser armazenadas, o LDAP não limita o administrador a utilizar somente os atributos padrões. Pode-se acrescentar tantos atributos quantos forem necessários aos *schemas* LDAP. São esses *schemas* que descrevem a estrutura dos objetos que podem ser armazenados dentro dos diretórios de informações do protocolo.

Não menos importante, existe a preocupação com a segurança da informação que é armazenada e transmitida. Nesse sentido, diversos tipos de criptografia e modelos de segurança, como a utilização do protocolo SASL (*Simple Authentication and Security Layer*) ou a utilização do protocolo SSL/TLS (*Secure Socket Layer / Transport Layer Security*), podem ser incorporados ao LDAP visando a segurança da informação.

Tendo em vista todas essas vantagens e por ter se tornado um padrão em serviços de diretórios, diversas empresas de software buscaram implementar soluções baseadas no protocolo LDAP, destacando o *Active Directory* da Microsoft e o OpenLDAP da OpenLDAP *Foundation*. Cada uma dessas implementações possui suas particularidades e vantagens, mas todas elas visam o mesmo objetivo: a centralização e a disponibilidade integrada de informações. A principal diferença entre essas duas implementações é que o *Active Diretory* é

uma implementação comercial proprietária e de código fechado, enquanto que o OpenLDAP é uma implementação livre de licença e de código aberto. Em função da característica de possuir código aberto e ser livre de licenciamento, o projeto OpenLDAP provê versões compiladas de sua implementação para a maioria dos sistemas operacionais baseados na plataforma UNIX, destacando MAC OS X, HP-UX, algumas versões do BSD e o Linux.

Uma vez levantadas as características de cada uma dessas implememtações, foi realizado um estudo de caso que teve como requisitos: (i) capacidade de integração entre sistemas operacionais heterogêneos; (ii) capacidade de integração com diversos sistemas e serviços da empresa; (iii) centralização e disponibilidade da informação; (iv) baixo custo na implementação do projeto. Logo, adotou-se como solução a implementação do OpenLDAP.

Antes de implementar a solução proposta, um projeto de implementação foi realizado de forma a garantir que todos os requisitos fossem cumpridos. Basicamente, esse projeto foi divido em quatro fases: verificação de capacidade de integração com os demais serviços e sistemas da empresa; instalação e configuração do OpenLDAP; integração do OpenLDAP com os sistemas e serviços providos; realização de testes que comprovam o sucesso do projeto e posterior homologação da solução.

A implementação do OpenLDAP foi feita sobre o sistema operacional Linux *CentOS*. Dentre os serviços providos na empresa e que deveriam se integrar ao OpenLDAP, destacaram-se o sistema gerenciador de domínio e autenticação SaMBa (que atua como controlador de domínio primário da empresa) e o sistema de Proxy SQUID. Levantadas essas informações, primeiramente foi feita uma verificação de integração do OpenLDAP com esses serviços e sistemas. A partir de então, foi realizada a instalação do OpenLDAP e sua posterior configuração. Um teste inicial que valida as configurações foi realizado de forma a garantir que o OpenLDAP funcionasse livre de erros e conforme esperado.

Uma vez instalado e validado o OpenLDAP, povoou-se a base de dados com informações iniciais de configuração do próprio protocolo e foram adicionados informações de grupos de usuários e usuários de rede. Partiu-se então para a integração com os serviços e sistemas da empresa. Foi realizada primeiramente a integração com o sistema de gerenciamento de domínio e autenticação SaMBa e, logo em seguida, o sistema de Proxy SQUID. Mais uma vez, testes de configuração que visam o funcionamento livre de erros dessas integrações foram realizados, garantindo assim que os serviços estariam disponíveis para uso.

Integrados os serviços e sistemas ao OpenLDAP, foram realizados testes que comprovaram o sucesso do projeto e o cumprimento dos requisitos levantados na empresa.

Dentre os testes realizados, os principais foram o teste de autenticação das estações de trabalho ao OpenLDAP e testes de autenticação de forma a liberar a navegação na Internet.

Diante dessas informações, demonstrou-se a eficácia do OpenLDAP e do protocolo LDAP como solução na centralização e disponibilidade de informações.

REFERÊNCIAS

BUTCHER, Matt. **Mastering OpenLDAP**: Configuring, Securing, and Integrating Directory Services. Olton: Packt Publishing Ltd., 2007.

ECKSTEIN, Robert; COLLIER-BROWN, David; KELLY, Peter. **Using Samba**: A File & Print Server for Heterogeneous Networks. Sebastapol: O'Reilly & Associates Inc, 2000.

GOOD, Gordon. **The LDAP Data Interchange Format (LDIF)**: Technical Specification. iPlanet e-commerce Solutions, 2000. Disponível em: <ftp://ftp.rfc-editor.org/in-notes/rfc2849.txt>. Acesso em: 23 set. 2010.

HARRISON, Roger. **Lightweight Directory Access Protocol (LDAP)**: Authentication Methods and Security Mechanisms. Novell, Inc., 2006. Disponível em: <ftp://ftp.rfc-editor.org/in-notes/rfc4513.txt>. Acesso em: 23 set. 2010.

HOWES, Timothy A.; SMITH, Mark C.; GOOD, Gordon S. **Understanding and Deploying LDAP Directory Services**. 2. ed. Addison Wesley, 2003.

KING, Robert R. **Mastering Active Directory**: Sybex, 2000.

KOUTSONIKOLA, Vassiliki; VAKALI, Athena. **LDAP**: Framework, Practices, and Trends. IEEE Internet Computing, v. 8, n. 5, p. 66-72, set./out. 2004.

MENDES, Douglas Rocha. **Redes de Computadores**: Teoria e Prática. São Paulo: Novatec, 2007.

OLIVEIRA, Luiz Cláudio Vieira de; CORREA, Osvaldo Manoel. **Normas para redação de trabalhos acadêmicos, dissertações e teses**. Belo Horizonte: Universidade FUMEC, 2008.

OPENLDAP FOUNDATION. **The OpenLDAP Foundation Overview**. Nevada. Disponível em: <http://www.openldap.org/foundation>. Acesso em: 23 set. 2010.

SCRIMGER, Rob *et al*. **TCP/IP**: A Bíblia. Rio de Janeiro: Campos, 2002.

SHARMA, Anuj. **History of Active Directory**. 2010. Disponível em: <http://www.articlesbase.com/operating-systems-articles/history-of-active-directory-3012999.html>. Acesso em: 12 out. 2010.

SHERMERSHEIM, Jim. **Lightweight Directory Access Protocol (LDAP)**: The Protocol. Novell, Inc., 2006. Disponível em: <ftp://ftp.rfc-editor.org/in-notes/rfc4511.txt>. Acesso em: 22 set. 2010.

SMITH, Mark. **Definition of the inetOrgPerson LDAP Object Class**. Netscape Communications, 2000. Disponível em: <ftp://ftp.rfc-editor.org/in-notes/rfc2798.txt>. Acesso em: 22 set. 2010.

TRIGO, Clodonil Honório. **OpenLDAP**: Uma abordagem integrada. São Paulo: Novatec, 2007.

TUTTLE, Steven *et al*. **Understanding LDAP**: Design and Implementation. 2. ed. Redbooks, 2004.

ZEILENGA, Kurt. **Lightweight Directory Access Protocol (LDAP)**: Directory information models. OpenLDAP Fondation, 2006. Disponível em: <ftp://ftp.rfc-editor.org/in-notes/rfc4512.txt>. Acesso em: 22 set. 2010.

ANEXO A - ARQUIVO DE CONFIGURAÇÃO `slapd.conf`

```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
#######################################################################
# OBS: Todos os comentários deste arquivo foram removidos. No total, o arquivo
#      completo constitui-se de 111 linhas.


include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/samba.schema

allow bind_v2

pidfile         /var/run/openldap/slapd.pid
argsfile        /var/run/openldap/slapd.args

database        bdb

suffix          "dc=foobar,dc=com"
rootdn          "cn=admin,dc=foobar,dc=com"

rootpw          {SSHA}V/qEsXXlBY/7tHS3yCoO8/8KCSG/Lz7n

directory       /var/lib/ldap

index objectClass                       eq,pres
index ou,cn,mail,surname,givenname      eq,pres,sub
index uidNumber,gidNumber,loginShell    eq,pres
index uid,memberUid                     eq,pres,sub
index nisMapName,nisMapEntry            eq,pres,sub
```

ANEXO B - ARQUIVO DE CONFIGURAÇÃO `smbldap_bind.conf`

```
############################
# Credential Configuration #
############################
# Notes: you can specify two differents configuration if you use a
# master ldap for writing access and a slave ldap server for reading access
# By default, we will use the same DN (so it will work for standard Samba
# release)

slaveDN="cn=admin,dc=foobar,dc=com"
slavePw="123456"

masterDN="cn=admin,dc=foobar,dc=com"
masterPw="123456"
```

ANEXO C - ARQUIVO DE CONFIGURAÇÃO `smbldap.conf`

```
##############################################################################
#
# General Configuration
#
##############################################################################
# OBS: Todos os comentários deste arquivo foram removidos. No total, o arquivo
#      completo constitui-se de 226 linhas.


SID="S-1-5-21-2621065664-1578420217-34586684127"

sambaDomain="foobar.com"

slaveLDAP="127.0.0.1"
slavePort="389"

masterLDAP="127.0.0.1"
masterPort="389"

ldapTLS="0"

verify="require"

cafile="/etc/pki/tls/certs/ldapserverca.pem"
clientcert="/etc/pki/tls/certs/ldapclient.pem"
clientkey="/etc/pki/tls/certs/ldapclientkey.pem"

suffix="dc=foobar,dc=com"
usersdn="ou=People,${suffix}"
computersdn="ou=Computers,${suffix}"
groupsdn="ou=Group,${suffix}"
idmapdn="ou=Idmap,${suffix}"
sambaUnixIdPooldn="sambaDomainName=${sambaDomain},${suffix}"

scope="sub"
hash_encrypt="SSHA"
crypt_salt_format="%s"

userLoginShell="/bin/bash"
userHome="/home/%U"
userHomeDirectoryMode="700"
userGecos="System User"
defaultUserGid="513"
defaultComputerGid="515"

skeletonDir="/etc/skel"
defaultMaxPasswordAge="45"

userSmbHome="\\PDC-SRV\%U"
userProfile="\\PDC-SRV\profiles\%U"
userHomeDrive="H:"
userScript="logon.bat"

mailDomain="idealx.com"
with_smbpasswd="0"
smbpasswd="/usr/bin/smbpasswd"
with_slappasswd="0"
slappasswd="/usr/sbin/slappasswd"
```

ANEXO D - ARQUIVO DE CONFIGURAÇÃO `smb.conf`

```
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options (perhaps too
# many!) most of which are not shown in this example
#
#---------------
# OBS: Todos os comentários deste arquivo foram removidos. No total, o arquivo
#      completo constitui-se de 312 linhas.

[global]

        workgroup = foobar.com
        server string = Foobar Domain Server Version %v

        netbios name = foobar

        log file = /var/log/samba/log.%m
        max log size = 50

        security = user

        domain master = yes
        domain logons = yes

        logon script = %G.bat
        logon path =
        logon drive = U:

        preferred master = yes

        load printers = yes
        cups options = raw

        ldap admin dn = cn=admin,dc=foobar,dc=com
        ldap user suffix = ou=People
        ldap group suffix = ou=Groups
        ldap idmap suffix = ou=Idmap
        ldap machine suffix = ou=Computers
        ldap passwd sync = Yes
        ldap ssl = Off
        ldap suffix = dc=foobar,dc=com
        passdb backend = ldapsam:ldap://127.0.0.1
        add machine script = /usr/sbin/smbldap-useradd -t 0 -w "%u"

[homes]
        comment = Home Directories
        browseable = no
        writable = yes

[printers]
        comment = All Printers
        path = /var/spool/samba
        browseable = no
        guest ok = no
        writable = no
        printable = yes

[netlogon]
        comment = Network Logon Service
        path = /var/lib/samba/netlogon
        guest ok = yes
        writable = no
        share modes = no
        root preexec = /var/lib/samba/scripts/mklogon.pl -u %U -m %m -s %L -o %a -i %I -d %t

[CPD]
        comment = Pasta do CPD
        path = /srv/shares/CPD
        read only = No
        browseable = No
        create mask = 664
        directory mask = 775
        veto files = /*.mp3/*.wma/*.wmv/*.avi/*.mpg/*.wav/*.cmd/*.pif/*.inf/*.vmx/*.rmvb
        nt acl support = no
```

ANEXO E - ARQUIVO DE CONFIGURAÇÃO `squid.conf`

```
#       WELCOME TO SQUID 2.6.STABLE21
#       ----------------------------
#
#       This is the default Squid configuration file. You may wish
#       to look at the Squid home page (http://www.squid-cache.org/)
#       for the FAQ and other documentation.
#
# OBS: Todos os comentários deste arquivo foram removidos. No total, o arquivo
#      completo constitui-se de 4538 linhas.
# -----------------------------------------------------------------------------

auth_param basic program /usr/lib/squid/squid_ldap_auth -b "ou=People,dc=foobar,dc=com" -v3 ↳
-H ldap://127.0.0.1/
auth_param basic children 5
auth_param basic realm Entre com suas credenciais (as mesmas usadas para fazer login):
auth_param basic credentialsttl 2 hour
auth_param basic casesensitive off

acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT

http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports

acl our_networks src 192.168.122.0/24
acl password proxy_auth REQUIRED
acl proibidos   url_regex       "/etc/squid/acl/proibidos"

http_access deny proibidos
http_access allow password
http_access allow our_networks
http_access deny all

icp_access allow all

http_port 192.168.122.2:3128

hierarchy_stoplist cgi-bin ?

access_log /var/log/squid/access.log squid

acl QUERY urlpath_regex cgi-bin \?

cache deny QUERY

refresh_pattern ^ftp:           1440    20%     10080

refresh_pattern ^gopher:        1440    0%      1440

refresh_pattern .               0       20%     4320

acl apache rep_header Server ^Apache

broken_vary_encoding allow apache

visible_hostname ldap.foobar

coredump_dir /var/spool/squid
```