Faculdade de Engenharia da Universidade do Porto

Integração do Samba 4 na plataforma IPBRICK para criar um Active Directory Open Source

João Pedro Leite de Castro

DISSERTAÇÃO

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Eletrotécnica e de Computadores
Major Automação

Orientador: Prof. Dr. Manuel Alberto Pereira Ricardo
Coorientadores: Eng. Miguel Ramalhão e Eng. Pedro Gouveia

Janeiro 2017

© João Pedro Leite de Castro, 2017

Resumo

O IPBRICK OS é uma plataforma de comunicações de servidores corporativos com base em Linux Debian, que usa o mais conhecido Software *Open Source* para Intranet, Comunicações Unificadas e Sistemas de Segurança.

Para a sua utilização como solução de intranet server, torna-se cada mais necessário dotar a IPBrick de uma nova solução inovadora de *Domain Controller*, *File Sharing* e *Print Server*.

Esta nova solução a implementar deverá responder a algumas limitações atuais, como dificuldades de integração no domínio IPBrick de postos com Windows 8 e Windows 10, de configuração de Políticas de Domínio, a ausência de funcionalidades de *Backup Domain Controller*, entre outras.

Isso é possível integrando o Samba 4 Active Directory *Open Source* com o sistema operativo IPBrick garantindo assim uma total interoperabilidade entre ambientes Microsoft (Windows) e IPBrick (Linux).

Abstract

IPBRICK OS is a communications platform for enterprise server systems, based on Linux Debian, and using the most known Open Source Software for Intranet, Unified Communications and Security Systems.

For its use as an intranet server solution, it becomes increasingly necessary to provide IPBrick with a new innovative solution of Domain Controller, File Sharing and Print Server.

This new solution to be implemented should respond to some current limitations, such as difficulties in integrating IPBrick domain with Windows 8 and Windows 10, Domain Policies configuration, lack of functionality of Backup Domain Controller, among others.

This is possible by integrating Samba 4 Active Directory Open Source with the IPBrick operating system, thus ensuring complete interoperability between Microsoft (Windows) and IPBrick (Linux) environments.

Agradecimentos

...

Índice

Resumo ..... **iii**

Abstract ..... **iv**

Agradecimentos ..... **v**

Índice ..... **vi**

Lista de figuras ..... **viii**

Lista de tabelas ..... **ix**

Abreviaturas e Símbolos ..... **x**

Capítulo 1 ..... **1**

Introdução 1
1.1 Enquadramento 1
1.2 Objetivos 2
1.3 Motivação 2
1.4 Estrutura do Trabalho 3

Capítulo 2 ..... **4**

Revisão Bibliográfica 4
2.1 Active Directory 4
2.1.1 Evolução histórica (X.500 e LDAP) 4
2.1.2 Objetos 6
2.1.2.1 Identificadores Únicos Globais (GUID) 6
2.1.2.1.1 Nomes Distintos (DN) 7
2.1.3 Domínios, Árvores de Domínio e Florestas 8
2.1.4 Unidades Organizacionais (OU) 9
2.1.5 Catálogo Global 9
2.1.6 Funções FSMO 10
2.1.6.1 Mestre de Esquema 10
2.1.6.2 Mestre de Atribuição de Nomes de Domínio 10
2.1.6.3 Mestre de RID 10
2.1.6.4 Mestre de Emulação de PDC 11
2.1.6.5 Mestre de Infraestruturas 11
2.1.7 Sincronização de tempo (NTP) 11
2.1.8 Sistema de Nomes de Domínio (DNS) 13
2.2 Samba 14
2.2.1 História 14

2.2.2 Funcionalidades ... 15
2.2.3 Protocolo CIFS ... 16
2.2.4 Samba 4 ... 17
2.2.4.1 Samba Active Directory ... 18

Capítulo 3 ... 21

Caracterização do Problema ... 21
3.1 IPBrick OS ... 21
3.1.1 Servidor de Intranet ... 22
3.1.2 Servidor de Comunicações Unificadas ... 23
3.2 Definição do problema ... 24
3.3 Solução proposta ... 26

Capítulo 4 ... 29

Implementação ... 29
4.1 Introdução ... 29
4.2 Plataforma de desenvolvimento ... 30
4.2.1 Ferramentas de suporte ao desenvolvimento ... 30
4.2.2 Estrutura da IPBrick ... 32
4.2.2.1 Interface ... 32
4.2.2.2 Base de dados ... 32
4.2.2.3 Classes de acesso à base de dados ... 33
4.2.2.4 Bibliotecas ... 33
4.3 Instalação e configuração do Samba 4 na IPBrick OS ... 33
4.3.1 Cenário de implementação e testes ... 34
4.3.2 Escolha da versão do Samba a instalar ... 35
4.3.3 Requisitos para instalação do Samba 4 ... 36
4.3.3.1 Suporte do sistema de ficheiros ... 36
4.3.3.2 Dependências de pacotes e bibliotecas ... 36
4.3.4 Script de instalação do Samba 4 ... 37
4.4 Configuração do Samba 4 para atuar como um Active Directory Domain Controller ... 39
4.4.1 Requisitos ... 39
4.4.2 Provisionamento do Samba 4 Active Directory Domain Controller ... 40
4.4.2.1 Validação do Samba 4 AD DC ... 42
4.4.3 RFC2307 – Instalação das extensões NIS ... 43
4.4.4 Configuração do serviço de DNS ... 45
4.4.4.1 Validação do DNS ... 46
4.4.5 Configuração do serviço Kerberos ... 46
4.4.5.1 Validação do Kerberos ... 47
4.5 Integração do Samba 4 AD DC com a IPBrick OS ... 47
4.5.1 Modo de Autenticação IPBrick AD DC ... 50
4.5.2 Extensão do Schema do LDAP do Samba 4 AD DC ... 52
4.5.3 Módulo de Gestão de Utilizadores da IPBrick AD DC ... 58
4.5.4 Módulo de Gestão de Grupos de Utilizadores da IPBrick AD DC ... 60
4.5.5 Módulo de Gestão de Computadores da IPBrick AD DC ... 61
4.5.6 Módulo de Gestão de Grupos de Computadores da IPBrick AD DC ... 62
4.5.7 Módulo de Servidor de Ficheiros da IPBrick AD DC ... 63
4.5.7.1 *Roaming profiles* ... 64
4.5.8 Módulo de Servidor de Impressão da IPBrick AD DC ... 65

Capítulo 5 ... 69

Conclusão ... 69
5.1 Conclusão da dissertação ... 69
5.2 Desenvolvimentos futuros ... 70

Referências ... 71

Lista de figuras

Figura 2.1 – Exemplo de uma hierarquia de objetos [1] 6

Figura 2.2 – Exemplo de uma árvore de domínio do domínio “mycorp.com” 8

Figura 2.3 – Exemplo da hierarquia da sincronização de tempo [1] 13

Figura 2.4 – CIFS e os seus serviços de suporte necessários [5] 17

Figura 2.5 – Integração total de serviços do Active Directory [8] 19

Figura 3.1 – Serviços fornecidos pela IPBrick como Servidor de Intranet 23

Figura 3.2 – Serviços fornecidos pela IPBrick como Servidor de Comunicações Unificadas 24

Figura 3.3 – Configuração atual da IPBrick OS 25

Figura 3.4 – Solução proposta para o Problema: IPBrik OS + Samba 4 AD DC 28

Figura 4.1 35

Figura 4.2 51

Lista de tabelas

Tabela 2.1 – Tipos de atributos DN mais utilizados nos servidores de diretório [2] 7

Tabela 4.1 41

Tabela 4.2 43

Tabela 4.3 53

Tabela 4.4 54

Tabela 4.5 54

Tabela 4.6 55

Tabela 4.7 56

Tabela 4.8 56

Tabela 4.9 58

Abreviaturas e Símbolos

Lista de Abreviaturas

ACL	Access Control List
AD	Active Directory
ADSI	Active Directory Service Interfaces
ADUC	Active Directory Users and Computers
API	Application Programming Interface
APT	Advanced Package Tool
BDC	Backup Domain Controller
CIFS	Common Internet File System
CN	Common Name
CUPS	Common Unix Printing System
DC	Domain Controller
DHCP	Dynamic Host Configuration Protocol
DLZ	Dynamically Loadable Zones
DN	Distinguished Names
DNS	Domain Name System
FQDN	Fully Qualified Domain Name
FSMO	Flexible Single Master Operation
FTP	File Transfer Protocol
GC	Global Catalog
GID	Group Identifier
GPO	Group Policy Object
GUID	Global Unique Identifier
IDE	Integrated Development Environment
IP	Internet Protocol
ISO	International Organization for Standardization
IT	Information Technology

KDC	Key Distribution Center
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Data Interchange Format
MTA	Mail Transfer Agent
NAS	Network-Attached Storage
NIS	Network Information Service
NOS	Network Operating System
NT4	Windows NT 4.0
NTLM	NT LAN Manager
NTP	Network Time Protocol
OEM	Original Equipment Manufacturer
OS	Operating System
OSI	Open System Interconnection
OU	Organizational Unit
PC	Personal Computer
PDC	Primary Domain Controller
PPD	PostScript Printer Description
RDN	Relative Distinguished Name
RFC	Request for Comments
RID	Relative ID
RSAT	Remote Server Administration Tools
SID	Security Identifier
SMB	Server Message Block
SSO	Single Sign-On
TCP	Transmission Control Protocol
TGT	Ticket-Granting Ticket
UDP	User Datagram Protocol
UID	Unique Identifier
UUID	Universal Unique Identifier
VFS	Virtual File System
WINS	Windows Internet Name Service

Capítulo 1

Introdução

Serviços partilhados de ficheiros e impressoras em rede são o coração de qualquer ambiente empresarial. Estes permitem utilizadores de uma rede guardarem os seus ficheiros com fiabilidade, partilhá-los entre si, e estabelecerem uma ponte entre múltiplos locais de trabalho, múltiplos dispositivos e ferramentas, e múltiplos sistemas operativos.

Este último aspeto é um ponto chave para um ambiente de trabalho moderno: integrar os mundos de Unix/Linux e Mac OS X com o tradicional ambiente Windows PC não só possibilita utilizadores de um sistema operativo diferente colaborarem num mundo dominado pelo Windows, como também possibilita aos ambientes Windows beneficiarem do poder dos servidores Unix/Linux com redes avançadas, alta disponibilidade, backup/recuperação e capacidades de automação assim como integração com infraestruturas e ferramentas de classe empresarial.

É por isto que a criação do Samba teve um tremendo impacto na produtividade dos ambientes empresariais de hoje em dia, desempenhando um papel de ponte de ligação entre dois mundos que não poderiam ser mais diferentes um do outro. É precisamente esta razão que torna o Samba tão fascinante: a capacidade de integrar o mundo Windows com o mundo Unix/Linux/Mac OS X, promovendo a interoperabilidade entre sistemas com protocolos de sistemas de ficheiros diferentes.

1.1 Enquadramento

Este projeto realiza-se no âmbito do tema de Dissertação “Integração do Samba 4 na plataforma IPBRICK para criar um Active Directory Open Source” proposto pela empresa IPBrick S.A. aos alunos do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores da FEUP, no ano letivo 2015/2016.

A IPBrick, S.A. é uma empresa fabricante e distribuidora, especializada em Soluções de Comunicação Corporativas. As suas grandes áreas de foco, que constituem o quadrante mágico da IPBrick, são: Comunicações Unificadas, Gestão de Documentos e Processos, Email & Groupware e Rede Social Corporativa.

Mais especificamente, o IPBrick.IC é um sistema operativo para servidores de intranet e comunicações estando já representado em inúmeros países e que conta com várias participações em eventos internacionais (CeBit, SIMO, LinuxWorld, entre outras). Para a sua utilização como solução de intranet server, torna-se cada mais necessário dotar a IPBrick de uma nova solução inovadora de *Domain Controller*, *File Sharing* e *Print Server*. Esta nova solução a implementar deverá responder a algumas limitações atuais, como dificuldades de integração no domínio IPBrick de postos com Windows 8 e Windows 10, de configuração de Políticas de Domínio, a ausência de funcionalidades de *Backup Domain Controller*, entre outras.

1.2 Objetivos

Após várias discussões com os Engenheiros Miguel Ramalhão e Pedro Gouveia definiram-se os seguintes objetivos principais:

- Integrar o Samba 4 na plataforma IPBrick;
- Configurar o Samba 4 como um Active Directory *Domain Controller*;
- Dotar o Samba 4 com os serviços *File Sharing* e *Print Server*;
- Garantir a possibilidade de Cópia de Segurança e Restauro da IPBrick com Samba 4 AD DC;
- Garantir a possibilidade de migração de um Windows AD para a IPBrick com Samba 4 AD DC;
- Garantir a possibilidade de migração de uma IPBrick com Samba NT4 para uma IPBrick com Samba 4 AD DC;
- Desenvolver uma interface web de configuração e administração para a IPBrick com Samba 4 AD DC.

1.3 Motivação

Este projeto foi escolhido, primeiro devido ao enorme interesse da minha parte pela área em que se integra, e segundo pelo fato de se tratar de um projeto em ambiente empresarial enquadrado no “mundo real” de trabalho, onde se pode obter qualificações muito valiosas e principalmente experiência profissional.

Do lado da empresa IPBrick S.A., a grande motivação é a possibilidade de poder oferecer aos clientes um serviço de Active Directory *Open Source* que garante uma total

interoperabilidade entre ambientes Microsoft (Windows) e IPBrick (Linux). Isso é possível integrando o software livre Samba 4 com a plataforma IPBrick.

Assim, sendo o Samba um software *Open Source*, e deixando de existir a atual dependência pelo Active Directory da Microsoft, os custos de desenvolvimento para a empresa e os custos relativos às licenças da Microsoft serão bastante reduzidos o que é extremamente vantajoso para a IPBrick S.A.

1.4 Estrutura do Trabalho

Esta dissertação encontra-se organizada em cinco capítulos.

O Capítulo 1, a Introdução, que apresenta o enquadramento, os objetivos, a motivação e a estrutura desta dissertação.

O Capítulo 2, a Revisão Bibliográfica, faz uma revisão de todos os conceitos e tecnologias que se revelaram importantes no contexto do desenvolvimento deste trabalho.

O Capítulo 3, a Caracterização do Problema, tem por objetivo explicar detalhadamente o Problema e apresenta várias soluções possíveis para o mesmo, entre as quais é escolhida uma que dará asas à implementação.

O Capítulo 4, a Implementação, descreve a solução implementada de forma a resolver os problemas descritos no Capítulo 2.

O Capítulo 5, a Conclusão, apresenta uma síntese do trabalho desenvolvido, as conclusões finais sobre o mesmo, e todo o trabalho futuro que vise melhorar e evoluir esta solução.

Capítulo 2

Revisão Bibliográfica

2.1 Active Directory

O Active Directory (AD) é um sistema operativo de rede (*Network Operating System* - NOS) da Microsoft. Inicialmente desenvolvido sobre o Windows 2000, o AD evoluiu durante mais de uma década através de múltiplas versões do Windows.

O AD permite a administradores gerir eficientemente a informação toda de uma empresa através de um repositório central que pode ser distribuído globalmente. Assim que a informação dos utilizadores, grupos, computadores, impressoras, aplicações e serviços for adicionada ao AD, pode ser disponibilizada para toda a empresa. A estrutura da informação pode corresponder à estrutura da organização, e os utilizadores podem consultar o AD para encontrarem a localização de uma empresa ou o endereço de e-mail de um utilizador. Com unidades organizacionais, pode-se delegar o controlo e a gestão dos dados [1].

2.1.1 Evolução histórica (X.500 e LDAP)

O sistema operativo de rede (NOS) é o termo utilizado para descrever um ambiente de rede onde vários tipos de recursos, como contas de utilizador, de grupo, e de computador, são guardados num repositório central que que é controlado por administradores e acessível para utilizadores finais. Tipicamente, um ambiente NOS é composto por um ou mais servidores que fornecem serviços NOS, tais como autenticação, autorização, e manipulação de contas, e por múltiplos utilizadores que acedem a esses serviços.

O primeiro ambiente NOS integrado da Microsoft tornou-se disponível em 1990 com o lançamento do Windows NT 3.0, que combinava várias funcionalidades dos protocolos do Gestor de LAN (*LAN Manager*) e do sistema operativo OS/2. O NOS NT evoluiu lentamente durante os oito anos seguintes até que o AD foi lançado em 1997 na versão beta.

O conceito de "domínio" foi introduzido no Windows NT, proporcionando um modo de agrupar recursos baseado nos limites administrativos e de segurança. Os domínios NT eram estruturas limitadas a cerca de 40.000 objetos, sendo estes utilizadores, grupos e computadores. Para organizações maiores, esta limitação impunha limites superficiais no design da estrutura do domínio. Frequentemente, os domínios também eram limitados geograficamente pois a replicação dos dados entre controladores de domínio tinham um desempenho pobre em altas latências ou ligações de baixa largura de banda. Outro problema significativo era a delegação de administração, que tendia a ser uma questão de "tudo ou nada" ao nível do domínio.

A Microsoft estava bem ciente destas limitações e da necessidade de reestruturar o seu modelo NOS em algo que fosse muito mais escalável e flexível, sendo os serviços de diretório baseados no protocolo *Lightweight Directory Access Protocol* (LDAP) uma possível solução.

De maneira geral, um serviço de diretório é um repositório de informação de rede, aplicação ou NOS, que é útil para múltiplas aplicações ou utilizadores, sendo que o Windows NT NOS é um tipo de serviço de diretório. Na verdade, existem vários tipos diferentes de diretórios, como páginas brancas da Internet, sistemas de e-mail, e até o sistema de nomes de domínio (*Domain Name System* - DNS). Apesar de cada um destes sistemas ter características de um serviço de diretório, os standards de como um serviço de diretório é implementado e acedido são definidos pelos protocolos X.500 e LDAP.

Em 1988, a União Internacional de Telecomunicações (*International Telecommunication Union* - ITU) e a Organização Internacional de Normalização (*International Organization for Standardization* - ISO) uniram-se para desenvolver uma série de standards de serviços de diretório, que resultaram no X.500. Apesar do X.500 ter provado ser um bom modelo para estruturar um diretório que disponibilizava bastantes funcionalidades de operações avançadas e de segurança, não era fácil implementar clientes que o pudessem utilizar. Uma das razões é o facto do X.500 ser baseado no modelo de Interconexão de Sistemas Abertos (*Open System Interconnection* - OSI) em vez do TCP/IP, que se tornou no standard para a Internet. Era bastante complexo e implementava bastantes funções que maioria dos clientes nunca precisaram. Por essa razão, na Universidade de Michigan (USA) começou-se a desenvolver um protocolo X.500 "mais leve", o LDAP.

A primeira versão do LDAP foi lançada em 1993, mas devido à falta de várias funcionalidades fornecidas pelo X.500 nunca se impôs. Só após a sua segunda versão (LDAPv2) é que começou a ganhar popularidade. Antes disso, era apenas uma *gateway* entre servidores X.500, que traduzia pedidos dos clientes e submetia-os para o servidor X.500. Assim, em 1995

uma equipa da Universidade de Michigan lançou o primeiro servidor de diretório LDAP que se tornou na base de muitos servidores futuros.

Em 1997, surgiu a última versão do LDAP (LDAPv3) que providenciava várias novas funcionalidades e que tornou o LDAP suficientemente robusto e extensível. Desde então, empresas como a Netscape, Sun, Novell, IBM, OpenLDAP Foundation, Microsoft e o Samba desenvolveram servidores de diretório baseados em LDAP [1].

2.1.2 Objetos

Os dados armazenados no AD são apresentados ao utilizador de forma hierárquica tal como num sistema de ficheiros. Cada entrada é considerada um objeto, sendo que existem dois tipos: containers e non-containers. Um ou mais containers ramificam-se hierarquicamente a partir de um container de raiz, e cada container pode conter non-containers ou outros containers. Um non-container não pode conter outros objetos. O tipo de container mais utilizado num AD é uma unidade organizacional (organizational unit - OU) que se encontra detalhada mais à frente. Na Figura 2.1 encontra-se um exemplo de uma hierarquia de objetos, que no AD é denominada de domínio [1].

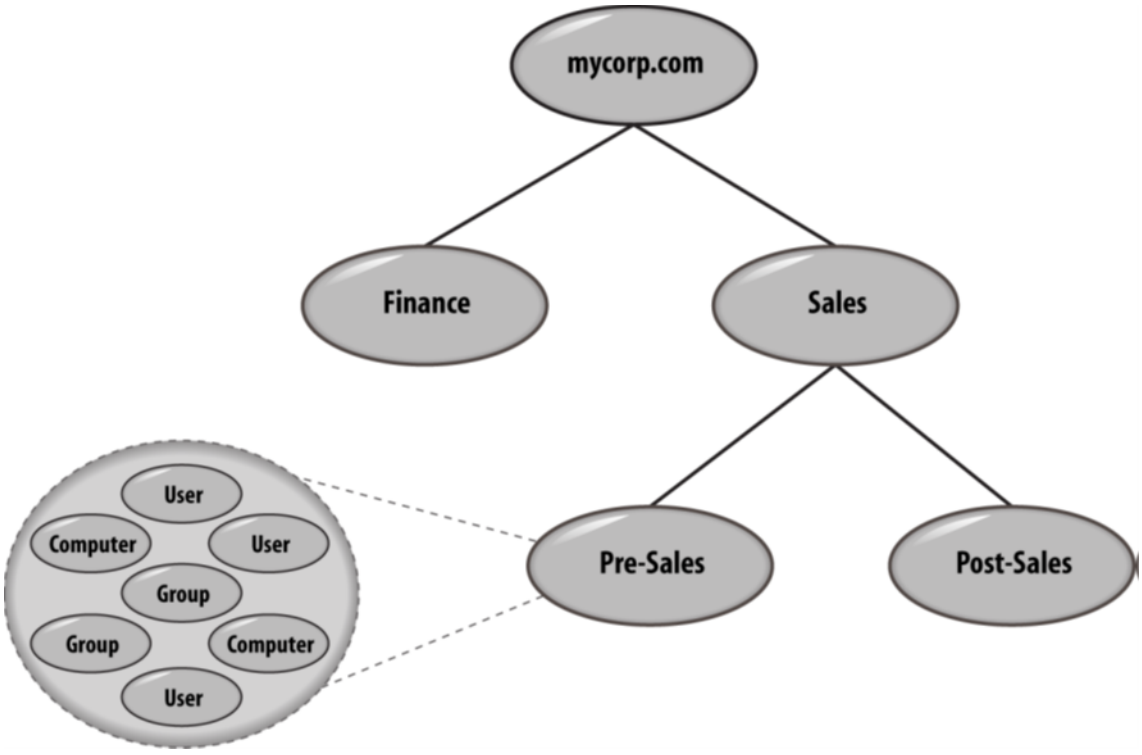

Figura 2.1 – Exemplo de uma hierarquia de objetos [1]

2.1.2.1 Identificadores Únicos Globais (GUID)

De modo a serem unicamente localizáveis e identificáveis no AD, os objetos possuem um identificador único global (globally unique identifier – GUID) que lhes é atribuído quando são criados. O GUID trata-se de um número de 128 bits e é a implementação feita pela Microsoft

do identificador universalmente único (universally unique identifier – UUID) criado pela empresa Digital Equipment Corporation (DEC). O identificador pertence a um objeto até o mesmo ser removido, independentemente do seu nome ser alterado ou de ser movido na árvore de informações de diretórios (directory information tree – DIT).

Devido ao GUID não ser fácil de memorizar e de não ser baseado na hierarquia de diretórios, é mais frequentemente utilizado o modo de referenciar objetos através de nomes distintos (distinguished names - DN) [1].

2.1.2.1.1 Nomes Distintos (DN)

Os DN são caminhos hierárquicos do AD que referenciam unicamente um objeto. Representam-se utilizando as regras e a sintaxe definida nos standards do LDAP. Um caminho à raiz da Figura 2.1 pode ser "dc=mycorp,dc=com".

Este DN representa o domínio raiz "mycorp.com".

Um nome distinto relativo (Relative Distinguished Name - RDN) referencia unicamente um objeto dentro do seu container parente no diretório. Admita-se que o DN para a conta Administrator no container Users no domínio "mycorp.com" é "cn=Administrator,cn=Users,dc=mycorp,dc=com". Então o RDN do utilizador é "cn=Administrator".

O RDN é sempre único dentro do seu container correspondente. É possível ter dois objetos com o mesmo RDN no diretório, apenas se estiverem dentro de containers parentes diferentes.

Outro prefixo muito comum é o "ou" que corresponde aos containers OU. Um exemplo é "cn=Keith Cooper,ou=Northlight IT Ltd,dc=mycorp,dc=com".

Todos os RDN usam um prefixo que indica a classe do objeto à qual está a ser referenciado. Por defeito, utiliza-se o prefixo "cn", common name [1].

Na Tabela 2.1 encontram-se os tipos de atributos mais utilizados nos servidores de diretório.

Tabela 2.1 – Tipos de atributos DN mais utilizados nos servidores de diretório [2]

Prefixo	Tipo de atributo
CN	Common name
L	Locality name
ST	State or province name
O	Organization name
OU	Organizational unit name
C	Country name
STREET	Street address
DC	Domain component
UID	User ID

O AD suporta os prefixos “CN”, “L”, “O”, “OU” e “DC”, sendo os “CN” e “OU” os mais utilizados.

2.1.3 Domínios, Árvores de Domínio e Florestas

A estrutura lógica do AD tem por base o conceito de “domínio”. Os domínios foram introduzidos nos Windows NT 3.x e 4.0, no entanto a sua estrutura era bastante inflexível, e por isso sofreram bastantes atualizações até serem utilizados no AD. Um domínio do AD é constituído pelos seguintes componentes:

- Uma estrutura hierárquica de objetos e containers baseada no X.500
- Um nome de domínio DNS como um identificador único
- Um serviço de segurança, que autentica e autoriza qualquer acesso aos recursos via contas no domínio ou fidedignidades com outros domínios
- Políticas que ditam o quanto uma funcionalidade está restringida a utilizadores ou máquinas dentro do domínio

Um controlador de domínio (domain controller - DC) apenas pode ser autorizado para um domínio, portanto não é possível hospedar vários domínios num único DC. Considerando o nome de domínio DNS “mycorp.com” para o primeiro domínio a ser criado no AD, então esse domínio trata-se da raiz de uma árvore de domínio.

Uma árvore de domínio consiste numa estrutura hierárquica de domínios que utilizam um esquema de nomes contíguo. Na Figura 2.2 encontra-se um exemplo de uma árvore de domínio do domínio “mycorp.com” [1].

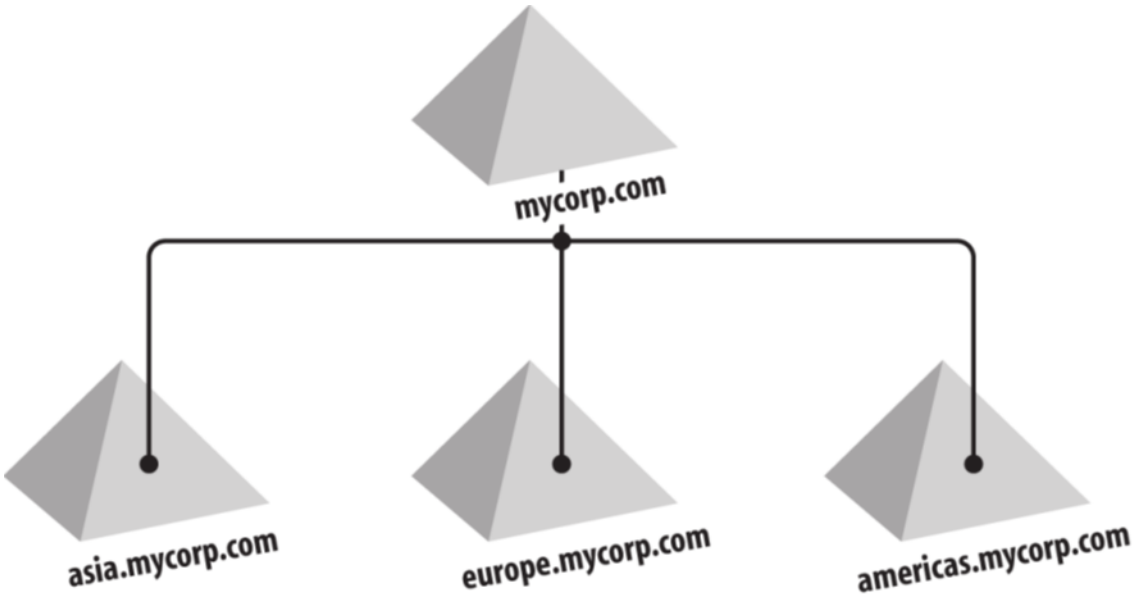

Figura 2.2 – Exemplo de uma árvore de domínio do domínio “mycorp.com”

As árvores de domínio facilitam a gestão e o acesso aos recursos, visto que todos os domínios de uma árvore de domínio implicitamente confiam uns nos outros através de fidedignidades transitórias, isto é, se o domínio A confia no domínio B e o domínio B confia no domínio C, então o domínio A também confia no domínio C.

Uma floresta é o conjunto de uma ou mais árvores de domínio que partilham um container de Esquema e Configuração comum e que estão ligadas através de fidedignidades transitórias. Assim que o primeiro domínio é criado, é também criada uma floresta. No AD, caso se remova o domínio raiz de uma floresta, a mesma é irreversivelmente destruída.

Uma fidedignidade de floresta permite a um administrador criar uma fidedignidade transitória de um ou dois sentidos entre dois domínios raiz de floresta. Esta fidedignidade permite que todos os domínios de uma floresta confiem em todos os domínios de outra floresta, e vice-versa [1].

2.1.4 Unidades Organizacionais (OU)

O tipo de container Unidade Organizacional (Organizational Unit - OU) pode conter enormes hierarquias de containers e de objetos, mas a razão pela qual é muitas vezes utilizado quase exclusivamente para armazenar as hierarquias, é o facto de suportar políticas de grupo.

Quando um domínio com Active Directory é instalado, um número de containers e de OUs por defeito são criados automaticamente, incluindo os containers de Utilizadores e Computadores e as OUs de controladores de domínio [1].

2.1.5 Catálogo Global

O Catálogo Global (Global Catalog - GC) é um componente do Active Directory muito importante pois é utilizado para realizar pesquisas nas Florestas. Trata-se de um catálogo de todos os objetos de uma floresta que contém um conjunto de atributos para cada objeto. O GC é apenas de leitura e não pode ser atualizado diretamente.

Tipicamente, em florestas com vários domínios, começa-se por efetuar uma consulta ao catálogo para localizar os objetos desejados, e só depois é que se realiza uma consulta mais direta ao controlador de domínio de modo a obter os atributos dos objetos.

O GC recebe atualizações regulares relativas a objetos existentes em todos os domínios através da replicação, para que os dados estejam sempre atualizados [1].

2.1.6 Funções FSMO

Apesar do Active Directory ser um diretório que suporta vários mestres, existem situações em que só deve existir um único controlador de domínio que possa realizar certas funções. Nesses casos, o AD elege um servidor para atuar como mestre para essas funções. Existem apenas cinco funções desse tipo, sendo que três aplicam-se a todos os domínios, e duas a toda a floresta. O servidor que é o mestre para uma particular função é designado por Mestre de Operações (FSMO) [1].

2.1.6.1 Mestre de Esquema

O controlador de domínio de mestre de esquema controla todas as atualizações e modificações efetuadas no esquema.

Para atualizar o esquema de uma floresta, tem que se recorrer ao mestre de esquema.

Só pode existir um mestre de esquema em toda a floresta [1, 3].

2.1.6.2 Mestre de Atribuição de Nomes de Domínio

O controlador de domínio que contém a função de mestre de atribuição de nomes de domínios, controla a adição ou remoção de domínios da floresta.

Só pode existir um mestre de atribuição de nomes de domínios em toda a floresta [1, 3].

2.1.6.3 Mestre de RID

O mestre de RID atribui sequências de IDs relativos (RIDs) a cada um dos vários controladores de domínio no respetivo domínio.

Sempre que um controlador de domínio cria um utilizador, grupo ou objeto de computador, atribui ao objeto um ID de segurança (SID) exclusivo.

O SID é constituído por um domínio SID, que é igual para todos os SID criados no domínio e um RID, que é exclusivo de cada SID criado no domínio.

Apenas poderá existir um controlador de domínio a funcionar como mestre de RID em cada domínio da floresta de cada vez [1, 3].

2.1.6.4 Mestre de Emulação de PDC

Para existir compatibilidade com versões anteriores, o mestre de emulação de PDC age como um controlador de domínio primário (PDC) do Windows NT.

O emulador processa as alterações de palavras-passe dos clientes e replica as atualizações para os controladores de domínio de reserva (BDC). Se uma autenticação de início de sessão falhar noutro controlador de domínio devido a uma palavra-passe errada, esse controlador de domínio irá reencaminhar o pedido de autenticação para o emulador de PDC antes de rejeitar a tentativa de início de sessão.

Por predefinição, o mestre de emulação de PDC também é responsável pela sincronização do tempo em todos os controladores de domínio no domínio.

Apenas poderá existir um controlador de domínio a agir como mestre de emulação de PDC em cada domínio da floresta de cada vez [1, 3].

2.1.6.5 Mestre de Infraestruturas

O mestre de infraestruturas é responsável pela atualização de referências de objetos no respetivo domínio a objetos noutros domínios. O mestre de infraestruturas compara os respetivos dados com os dados do catálogo global. Se o mestre de infraestruturas localiza dados desatualizados, requisita os dados atualizados a catalogo global. Em seguida, o mestre de infraestruturas efetua a replicação dos dados atualizados para os outros controladores de domínio existentes no domínio.

O mestre de infraestruturas é também responsável pela atualização de referências grupo/utilizador sempre que os membros de grupos são alterados ou quando o respetivo nome é mudado.

Só pode existir um controlador de domínio a agir como mestre de infraestruturas em cada domínio [1, 3].

2.1.7 Sincronização de tempo (NTP)

O Active Directory é altamente dependente que todos os controladores e membros de domínio tenham os relógios sincronizados. O protocolo de autenticação dos clientes do AD, o Kerberos, utiliza os relógios do sistema para verificar a autenticidade dos seus pacotes. Por defeito, o AD suporta uma tolerância de cerca de cinco minutos para os relógios. Se a variação do tempo exceder esta tolerância, os clientes ficam impossibilitados de se autenticarem, e no caso dos controladores de domínio, a replicação não vai ocorrer.

Assim sendo, o Windows e o AD (e o Samba) implementam um sistema de sincronização de tempo baseado no protocolo Network Time Protocol (NTP) que assegura que todas as

máquinas numa floresta têm o relógio sincronizado. A hierarquia da sincronização de tempo está demonstrada no exemplo da Figura 2.3 [1].

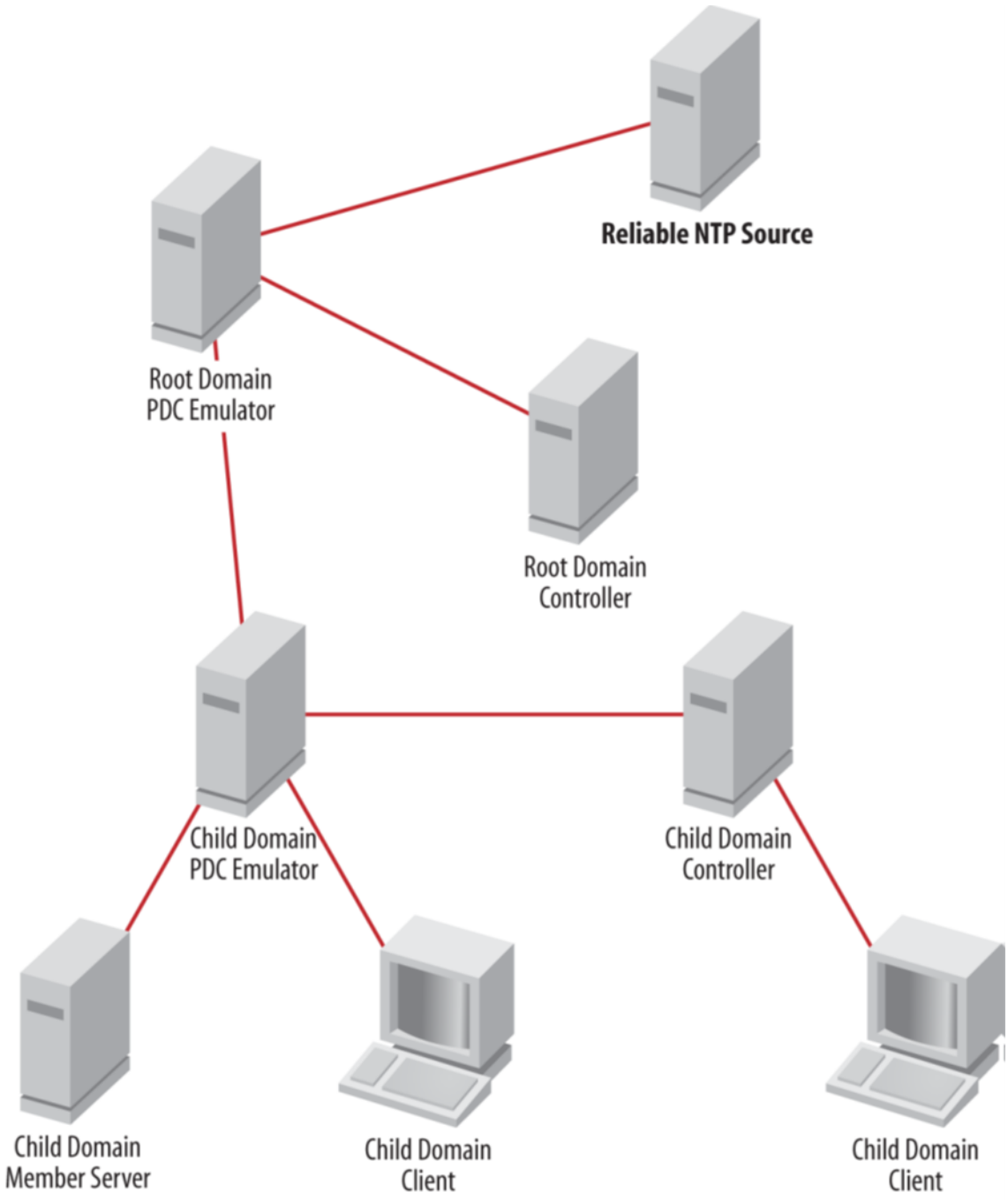

Figura 2.3 – Exemplo da hierarquia da sincronização de tempo [1]

2.1.8 Sistema de Nomes de Domínio (DNS)

Uma das grandes vantagens do Active Directory em relação ao seu precedente, Windows NT, é a sua dependência do sistema de nomes de domínio (Domain Name System – DNS) em vez do Windows Internet Naming Service (WINS) para resolução de nomes. O DNS é o serviço de resolução de nomes standard mais utilizado na Internet. Por outro lado, o WINS sempre

teve pouco suporte da indústria e tornou-se num sério candidato para extinção em muitas redes corporativas. Convém referir que apesar do AD não precisar do WINS (ou resolução de nomes NetBIOS), outros sistemas e tecnologias podem precisar desse serviço para funcionarem corretamente.

O DNS é um sistema de resolução de nomes hierárquico, e é o maior serviço de diretório público implementado. Praticamente todas as empresas utilizam o DNS para serviços de resolução de nomes, tais como hostname para endereço IP, endereço IP para hostname, e hostname para hostname.

No contexto do AD, existem três conceitos do DNS muitos importantes:

- Zonas são partes delegadas do namespace do DNS
- Os registos dos recursos contêm informação de resolução de nomes
- O DNS dinâmico permite aos clientes adicionarem e removerem registos de recursos dinamicamente

2.2 Samba

O Samba é uma suite de software que permite um sistema UNIX surgir e funcionar como um servidor Microsoft Windows, quando visto por outros sistemas numa rede. É constituído por várias componentes que operam em conjunto de modo a implementar as partes cliente e servidor do protocolo CIFS.

Existem várias razões para se utilizar o Samba em vez do Windows Server. Como muitos administradores de redes experientes podem testemunhar, o Samba oferece uma constante fiabilidade, escalabilidade e flexibilidade. Para além disso, possibilita liberdade de escolha e é gratuito. Encontra-se disponível em [4] sobre os termos da Licença Pública Geral GNU. Graças à sua portabilidade, é compatível com diversas plataformas, como FreeBSD, Linux, Solaris ou OS X.

Sendo um software open source possui uma vasta comunidade integrada no projeto, composta por uma variedade de colaboradores, e não apenas developers. A comunidade do Samba possui desde profissionais de TI a professores, consultores e dentistas. É ainda distribuído e comercialmente suportado por gigantes corporações, como a HP, IBM, Sun, Apple, RedHat e Novell [5].

2.2.1 História

Andrew Tridgell, o criador do Samba, iniciou o projeto em 1991, enquanto trabalhava com uma suite de software Digital Equipment Corporation (DEC) chamada Pathworks, criada para ligar computadores DEC VAX a computadores feitos por outras empresas. Sem ter noção do que estava a fazer, Andrew Trigdell criou um programa de servidor de ficheiros para um protocolo que era parte da Pathworks. Mais tarde, esse protocolo, viria a ser o Server

Message Block (SMB), o antecessor do CIFS. Uns anos depois, ele começou a distribuir o seu servidor SMB gratuitamente na Internet com o nome de "SMB Server". Porém, esse nome já pertencia a um produto de outra empresa, então ele utilizou a seguinte abordagem Unix para escolher um novo nome:

```
# grep -i '^s.*m.*b.*' /usr/dict/words
```

A resposta foi a seguinte:

```
# salmonberry
# samba
# sawtimber
# scramble
```

E assim nasceu o nome "Samba". Atualmente, o Samba é intensamente desenvolvido por uma equipa de programadores distribuídos por todo o mundo [5].

Para explicar o método de desenvolvimento do Samba, baseado na aprendizagem dos protocolos da Microsoft, Andrew Trigdell disse:

"Imagine que queria aprender Francês, e não existiam livros, cursos, etc, disponíveis para ensinar-lhe. Você pode optar por aprender indo até França, sentar-se num café Francês e apenas ouvir as conversas à sua volta. Você tira anotações do que os clientes dizem para o empregado e que comida é que chega. Dessa maneira você eventualmente aprende as palavras pão, café, etc." [6].

2.2.2 Funcionalidades

Uma das melhores maneiras para descrever o Samba é explicar algumas das suas funcionalidades. Alguns exemplos de serviços oferecidos pelo Samba são [5]:

- Partilhar uma ou mais árvores de diretórios
- Fornecer um Distributed Filesystem (MS-DFS) namespace
- Centralmente gerir impressoras, propriedades de impressão, e drivers associadas para acesso através de clientes Windows
- Assistir clientes com navegação de rede
- Autenticar clientes fazendo login num domínio Windows
- Providenciar resolução de nomes de servidor Windows Internet Name Service (WINS)

Inclui ainda ferramentas de cliente que permitem aos utilizados num sistema Unix aceder a pastas e impressoras que sistemas Windows e servidores Samba oferecem na rede.

O Samba funciona à base de três daemons Unix [5]:

- **Smbd** – gere a partilha de ficheiros e impressão e providencia autenticação e autorização para clientes.

- **Nmbd** – gere o registro de nomes NetBIOS do Samba, implementa um serviço NetBIOS Name Service (NBNS) compatível com Windows, também conhecido por servidor WINS, e participa em browsing elections.
- **Winbindd** – comunica com controladores de domínio para fornecer informação tal como a que grupos um utilizador pertence. Também faculta uma interface para esquemas de autenticação Windows LanManager, conhecidos por autenticação NTLM, para serviços Unix que não sejam do Samba.

Como explicado anteriormente, o Samba é uma ferramenta muito útil para possibilitar computadores Windows e Unix a coexistirem na mesma rede. Mas para além disso existem várias razões específicas para se querer implementar um servidor com o Samba numa rede [5]:

- Deixa de ser necessário pagar uma licença por um servidor Windows.
- Possibilidade de obter uma área comum de dados e diretorias de utilizadores para transitar de um servidor Windows para um Unix, ou vice-versa.
- Partilha de impressoras entre estações de trabalho Windows e Unix.
- Suporte a grupos de computadores com um misto de computadores Windows e Unix.
- Integração de autenticação em Windows e Unix mantendo uma única base de dados com as contas dos utilizadores que funciona em ambos os sistemas.
- Interligação de várias redes como Unix, Windows e Macintosh (OS X) utilizando apenas um protocolo.

2.2.3 Protocolo CIFS

Os sistemas operativos Microsoft mais modernos dependem do protocolo de partilha de recursos Common Internet File System (CIFS). Este protocolo disponibiliza Application Programming Interfaces (APIs) para manipular ficheiros e para implementar funcionalidades de administração remota como mudanças de password de utilizadores e serviços de impressão.

O protocolo CIFS, também conhecido por SMB/CIFS, é a última variante do protocolo SMB. Foi implementado para os sistemas operativos Windows 2000 e sequentes, e não suporta o API NetBIOS. Já o protocolo SMB refere-se aos sistemas Windows 9x/ME e NT, e utiliza o NetBIOS.

É um protocolo orientado às ligações, tendo a capacidade de manter o estado da aplicação/processo, e que depende de três serviços de rede de suporte:

- Um serviço de nomes
- Um meio de enviar datagramas para um único ou grupo de hosts
- Um meio de estabelecer ligações de longo prazo entre clientes e servidores

Tanto o Samba como o Windows 2000 (ou mais recentes) suportam a utilização de serviços standard de IP para cumprir estes requisitos. Por exemplo, o serviço Domain Name Service (DNS) traduz nomes para endereços de IP, o protocolo User Datagram Protocol (UDP)

providencia o serviço de datagramas, e o TCP/IP possibilita o suporta necessário para as sessões do CIFS.

Em sistemas operativos anteriores ao Windows 2000, os clientes Microsoft dependiam de uma "camada" chamada NetBIOS para permitir esta infraestrutura de suporte. Já os clientes e servidores modernos que utilizam o CIFS, incluindo o Samba, funcionam sem utilizarem os serviços do NetBIOS, no entanto têm modos de operação para comunicar com implementações do CIFS mais antigas [5]. A Figura 2.4 ilustra a relação entre o CIFS, os hosts numa rede, e os serviços de rede principais.

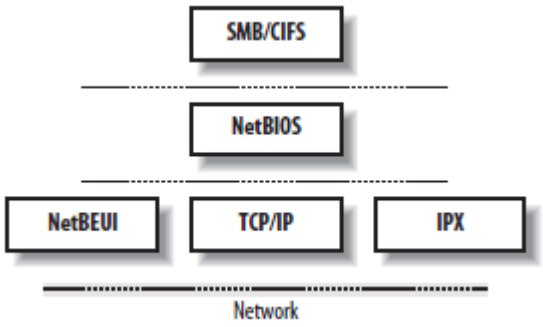

Figura 2.4 – CIFS e os seus serviços de suporte necessários [5]

2.2.4 Samba 4

A versão 4 do Samba foi lançada em dezembro de 2012 após 10 anos de desenvolvimento. Foi a primeira implementação de software livre dos protocolos do Active Directory da Microsoft. Familiar a todos os administradores de redes, os protocolos do AD são o coração das implementações mais modernas de serviços de diretorias.

A grande novidade desta versão é a integração do AD que se encontra detalhada mais à frente, no entanto existem outras melhorias importantes que devem ser mencionadas.

Uma importante mudança, está relacionada com o próprio projeto do Samba e com o fluxo de trabalho do seu desenvolvimento. O processo de novo commit da equipa do Samba requer que todos os commit sejam revistos por dois desenvolvedores, e cada vez que um commit é submetido, realiza-se um test suit. Assim, assegura-se a alta qualidade de código.

Outra melhoria, é o fato do Samba 4 em modo servidor de ficheiros ser altamente assíncrono, providenciando uma taxa de transferência de dados superior e uma utilização dos recursos do servidor melhorada.

Por último, uma alteração também relacionada com o desempenho do servidor, é o suporte dos protocolos SMB2.1 e SMB3. Estas versões têm uma compatibilidade muito superior

com clientes Microsoft mais recentes, suportando encriptação, links simbólicos, server-side copy, e recursos de alta disponibilidade como multipath e failover.

O Samba 4 oferece ainda scripts de upgrade para organizações que utilizem funcionalidade Microsoft Windows NT Domain Controller do Samba 3.x, para permitir-lhes migrar para o Samba 4 sem dificuldades.

Quanto à sua segurança, o Samba 4 é um dos onze projetos open source que a empresa líder em integridade de software Coverity certificou como "seguro". A equipa do Samba disponibiliza respostas imediatas a qualquer vulnerabilidade de segurança, e fornece fixes a todas as empresas que utilizam o código do Samba em coordenação com agências de informação da segurança standard da indústria.

Sendo que é um software livre, o Samba 4 é a escolha ideal para Original Equipment Manufacturers para utilizarem os seus produtos de ficheiros, impressão e autenticação. É facilmente integrável num vasto leque de diferentes tarefas, e pode ser customizado indiscriminadamente pelas empresas para satisfazer as suas necessidades. Posto isto, o Samba 4 inclui uma interface Virtual File System modular que as empresas podem utilizar para personalizarem rapidamente e eficientemente o Samba, de modo a tirarem vantagem de quaisquer características especificas da sua tecnologia sem terem que modificar nenhuma parte do código principal do Samba. Desde sistemas de ficheiros avançados a análises de tráfego de rede, a camada VFS do Samba permite código externo ser facilmente integrado com o Samba. Módulos de exemplo são fornecidos como código fonte para as empresas editarem conforme as suas preferências.

Para concluir, o Samba é a escolha líder em tecnologia para serviços de ficheiros Windows nas plataformas Linux e Unix e em soluções integradas Network Attached Storage. O Samba é utilizado pelas empresas que vendem soluções NAS abrangendo desde sistemas críticos de negócios de alta qualidade, a dispositivos de consumo de baixo nível. É totalmente compatível com o protocolo de internet IPv6 e cumpre todos os mandatos para interoperabilidade da rede moderna. Uma grande variedade de empresas oferece suporte comercial para o Samba.

Ideal para aplicações embarcadas e de baixo nível, o Samba 4 é eficiente e flexível quando adaptado a grandes aglomerados. A sua interface de programação em Python e o seu toolkit de administração são bastante valiosos em desenvolvimentos empresariais [7, 8].

2.2.4.1 Samba Active Directory

O servidor compatível com AD do Samba 4 foi criado com a ajuda da documentação de protocolos oficial publicada pela Microsoft e com os testes de interoperabilidade efetuados pelos engenheiros da Microsoft.

Thomas Pfenning, diretor de desenvolvimento do Windows Server, disse:

"Active Directory é um pilar nos ambientes empresariais IT, e a Microsoft está comprometida em apoiar a interoperabilidade entre plataformas. Estamos satisfeitos que a documentação e os testes de interoperabilidade que a Microsoft providenciou tenham sido um fator chave no desenvolvimento da funcionalidade Active Directory do Samba 4."

Para uma integração total do AD, o Samba 4 inclui um servidor dinâmico DNS seguro, um servidor de autenticação Heimdal Kerberos, um servidor de diretórios LDAP, e implementações de todas as chamadas de procedimento remotas necessárias para o AD. Desta maneira, o Samba 4 oferece tudo o que é necessário para funcionar como um AD DC compatível com todas as versões de clientes Microsoft Windows atualmente suportadas pela Microsoft [7, 8].

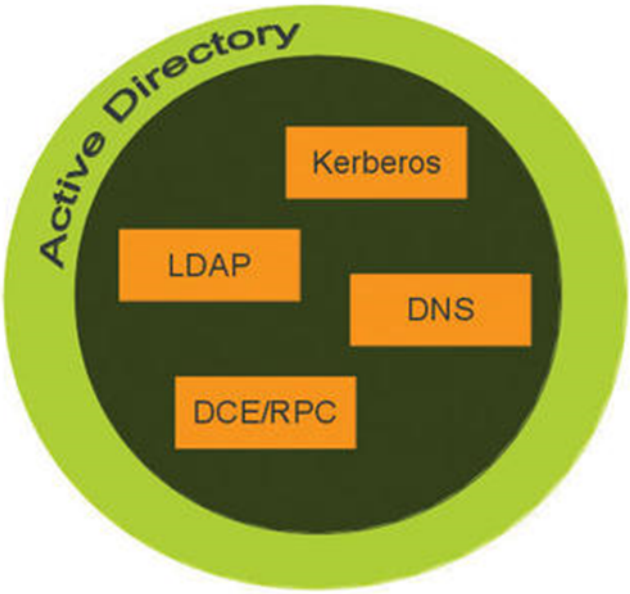

Figura 2.5 – Integração total de serviços do Active Directory [8]

A principal vantagem de um domínio AD é que todos os componentes (DNS, LDAP, KDC) estão integrados e podem ser ampliados sobre múltiplos controladores de domínio. Portanto, o Samba 4 incorpora as mesmas vantagens: integração total e um schema predefinido.

O servidor Samba 4 compatível com AD também proporciona suporte a políticas de grupo, roaming profiles, ferramentas de Administração do Windows e integra-se com o Microsoft Exchange e serviços de software livre como o OpenChange.

Pode ainda se juntar a um domínio AD da Microsoft, e os AD DC da Microsoft também se podem unir a um servidor Samba 4 AD, o que demonstra uma verdadeira interoperabilidade peer-to-peer das implementações de protocolos AD da Microsoft e do Samba.

Um dos seus daemons principais, o winbind, foi melhorado, o que permite aos servidores de ficheiros Samba 4 facilmente integrarem-se com um AD existente como membros do servidor. Tanto o Microsoft AD como o Samba 4 AD são suportados.

Reconhecendo o valor da interoperabilidade do Samba 4 AD, Steve van Maanen, cofundador da Starsphere LCC, uma empresa de serviços IT em Tokyo, disse:

"Graças ao Samba 4, eu tenho dois controladores de domínio AD totalmente a replicar que iniciam em menos de 10 segundos! É bom ter alternativas, e o Samba 4 é uma alternativa muito boa." [7, 8].

Capítulo 3

Caracterização do Problema

Neste capítulo é apresentado o sistema operativo IPBrick OS sobre o qual se debruça o problema.

Depois é explicado mais detalhadamente qual o problema tendo em conta a solução atual da IPBrick e os objetivos propostos nesta dissertação.

Por último, são apresentadas várias soluções possíveis para o problema, entre as quais é escolhida uma que dará asas à implementação.

3.1 IPBrick OS

O sistema operativo IPBrick OS é uma plataforma de comunicações para sistemas de servidores corporativos, com base em Linux Debian, que usa o mais conhecido Software *Open Source* para Intranet, Comunicações Unificadas e Sistemas de Segurança.

O software IPBrick OS (Servidor Linux) foi projetado de modo a que, estes três tipos de servidores sejam geridos pelo mesmo administrador:

- Servidor de Intranet (Figura 3.1)
- Servidor de Comunicações Unificadas (Figura 3.2)
- Servidor de Segurança (Figura 3.1 e 3.2)

Sendo que não é necessária uma pessoa especializada para configurar um servidor de última geração, em qualquer uma das anteriores áreas.

A IPBrick OS providencia também um alto nível de integração entre pacotes de Software *Open Source* previamente configurados e nunca perde configurações importantes que façam o servidor funcionar de modo instável.

A integração não se limita apenas a serviços da IPBrick OS, aplica-se também a Software de terceiros que são totalmente compatíveis com a IPBrick OS. Por exemplo, o Livro de Endereços contido na base de dados do servidor, é usado por todos os serviços IPBrick e Software de terceiros.

O software IPBrick OS está também preparado para uma evolução suave. Significa isto que foi projetado para simplificar a evolução entre as versões, não tendo o utilizador que saber

de mudanças significativas a nível do sistema operativo Linux Debian. Bem como versões de pacotes de software e versões de sintaxes do ficheiro de configuração.

De facto, é uma solução de alta performance que permite uma instalação e processos de configuração extremamente rápidos, incluindo capacidade de recuperação de desastres a nível corporativo. Isto permite à IPBrick oferecer uma solução, com um custo total de aquisição extremamente baixo.

3.1.1 Servidor de Intranet

A Intranet consiste na aplicação das tecnologias da Internet às redes internas das empresas e organizações, melhorando os processos de negócio e baixando os custos. Todos os colaboradores têm o acesso adequado à informação estratégica que necessitam, de forma a também reduzir os custos operacionais. A solução da IPBrick é baseada em tecnologia aberta e integra com todos os tipos de software, possibilitando uma instalação em tempo recorde.

Os principais serviços da IPBrick como Servidor de Intranet são os seguintes:

- **Correio Eletrónico e Ferramentas Colaborativas**
  - É um servidor de correio eletrónico (protocolos SMTP, IMAP e POP) e disponibiliza ferramentas colaborativas como Agenda, Contatos e Calendário, proporcionando um verdadeiro ambiente colaborativo.
- **Controlador de Domínio**
  - Trabalha com protocolo LDAP e integra com o Active Directory do Windows Server da Microsoft, herdando todas as configurações das contas já criadas. A passagem da tecnologia Microsoft para a tecnologia IPBrick é transparente para o utilizador;
  - Oferece também serviços de suporte à rede como os protocolos DNS e DHCP;
  - Servidor Radius.
- **Servidor de Ficheiros**
  - Áreas de trabalho individuais e de grupo.
- **Servidor de VMware**
  - Possibilidade de ter sessões simultâneas em sistemas operativos diferentes a correr no servidor de VMware. Instala-se como uma aplicação na IPBrick e divide o servidor físico em múltiplas máquinas virtuais.
- **Servidor de Impressoras**
- **Servidor de Base de Dados**
- **Segurança dos Dados**
  - Serviço de backup das áreas de trabalho;
  - Antivírus pré-instalado para correio eletrónico e áreas de trabalho.
  - Antisspam pré-instalado.
- **Suporte de Aplicações de Negócio**
  - Backup - Bacula, SEP sesam;
  - *Customer Relationship Management* (CRM) – CRM, Better.CRM;
  - Educação - Moodle;
  - Gestão Documental - iPortalDoc;

- *Enterprise Resource Planning* (ERP) – OpenERP, Gestix, Better.BIZ;
- *Groupware* – EGroupware, DotProject, Open-Xchange, Zarafa;
- Monitorização – Nagios, NTop;
- Gestão de Incidências – iPTicket;
- Videovigilância – ZoneMinder.

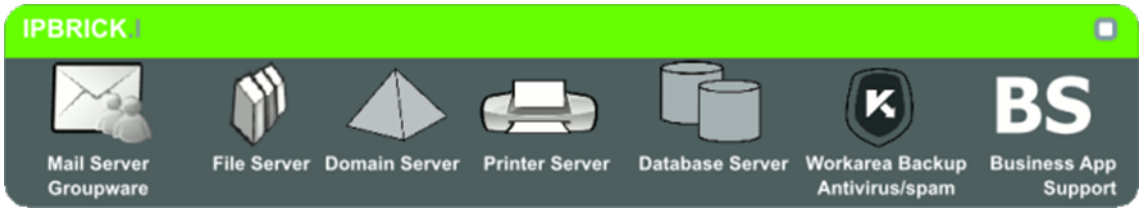

Figura 3.1 – Serviços fornecidos pela IPBrick como Servidor de Intranet

3.1.2 Servidor de Comunicações Unificadas

Voz, Vídeo, Fax, Mail, Web e *Instant Messaging* constituem o núcleo das comunicações de todas as empresas.

A IPBrick como Servidor de Comunicações Unificadas é pioneira na implementação do conceito UCoIP (*Unified Communications over IP*), respondendo aos desafios da mobilidade e da integração das comunicações empresariais. Adapta-se às suas necessidades com toda a flexibilidade e o mais baixo TCO (Custo Total de Propriedade).

Esta solução possui uma integração nativa, com gestão documental (iPortalDoc) para gravação das comunicações (chamadas, faxes, emails e conversas instantâneas) e (PHC e CRM) para registo automático das comunicações.

Trata-se de uma solução escalável possibilitando operação em alta disponibilidade ou operação em modo distribuído.

Destaca-se a sua simplicidade no VoIP com um único equipamento, PBX IP, Media Gateway que interliga com PBX existentes e à rede PSTN, Media Proxy, Firewall e VPN.

Os principais serviços da IPBrick como Servidor de Comunicações Unificadas são os seguintes:

- **Correio Eletrónico**
  - *Relay* de mail – protocolos SMTP, IMAP e POP;
  - WebMail.
- **Fax 2 Mail e Mail 2 Fax**
- **Mail 2 SMS**
- **Serviços de Telefonia**
  - Gateway VoIP;
  - PBX IP;
  - SIP Proxy.

- **Serviços Web**
  - Servidor Web – protocolos HTTP e HTTPS;
  - WebPhone;
  - Proxy e Cache HTTP/FTP.
- **Sistema de Mensagens Instantâneas**
  - WebChat;
  - Servidor de *Instant Messaging*.
- **Segurança das Comunicações**
  - Firewall;
  - VPN Server – protocolos SSL, IPSec e PPTP;
  - IDS (*Intrusion Detection System*);
  - Antivírus pré-instalado para Correio Eletrónico e Proxy;
  - Antisspam pré-instalado

Figura 3.2 – Serviços fornecidos pela IPBrick como Servidor de Comunicações Unificadas

3.2 Definição do problema

No contexto do Problema, o foco está principalmente na solução IPBrick como Servidor de Intranet, visto que após a integração do Samba 4 com a IPBrick OS, os seus serviços *core* vão passar a ser fornecidos e controlados pelo Samba 4. Estes serviços são os seguintes:

- Controlador de Domínio;
- Servidor de LDAP;
- Servidor de Ficheiros;
- Servidor de Impressoras;
- Servidor de DNS;
- Servidor de NTP;
- Módulo de Gestão de Utilizadores;
- Módulo de Gestão de Computadores/Máquinas.

Esta integração também afetará a solução IPBrick como Servidor de Comunicações Unificadas pois a mesma utiliza estes serviços acabados de listar, dependendo indispensavelmente do serviço de LDAP onde guarda informações necessárias para o seu funcionamento e do Módulo de Gestão de Computadores/Máquinas através do qual se faz a administração dos telefones por exemplo. Outro serviço em causa importante de referir é o Servidor de Email, que também utiliza o LDAP para guardar informações.

Atualmente a IPBrick OS recorre à anterior versão do Samba, a Versão 3, para fornecer os serviços de Controlador de Domínio, Servidor de Ficheiros, Servidor de Impressoras, Módulo de Gestão de Utilizadores e Módulo de Gestão de Computadores/Máquinas. Para todos estes serviços funcionarem corretamente é necessário um servidor LDAP onde são armazenadas as informações dos mesmos. Assim, a IPBrick OS tem configurado como *backend* do Samba 3 o OpenLDAP, uma implementação *open source* de LDAP [9]. Os serviços de DNS, NTP e DHCP, também essenciais para o funcionamento de todo o sistema, são disponibilizados, geridos e controlados por módulos independentes do Samba 3, apesar de estarem intrinsecamente interligados. Na Figura 3.3 pode-se visualizar a atual configuração da IPBrick OS, de acordo com a perspetiva em questão.

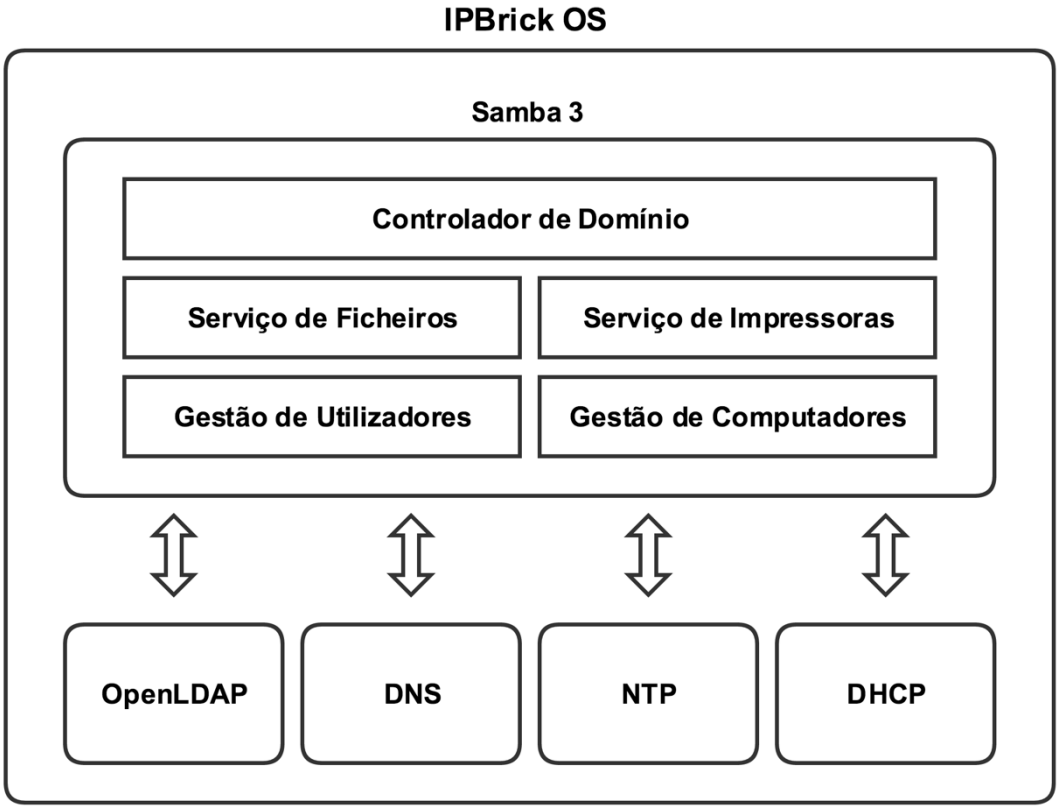

Figura 3.3 – Configuração atual da IPBrick OS

O Samba 4 é capaz de oferecer todos estes serviços de forma integrada e com uma estrutura muito mais avançada, robusta e atualizada, com a grande vantagem de poder atuar como um Active Directory Domain Controller.

Esta nova versão do Samba 4 apresenta como novidades um servidor DNS e protocolo NTP integrados, um servidor LDAP e um servidor de autenticação Kerberos *Key Distribution Center* (KDC). Suporta ainda as versões mais recentes do protocolo SMB, o SMB 2.1 e o SMB 3.0, e disponibiliza uma interface de *scripting* em Python [10].

Com todas estas novas caraterísticas a atualização para o Samba 4 torna-se quase obrigatória, visto que o mesmo consegue fornecer serviços de rede completos sendo *open source*, uma alternativa gratuita aos servidores Windows da Microsoft.

No entanto, convém ter em conta que o Samba 3 é um software já com muitos anos, bastante robusto e estável, principalmente nos seus serviços de ficheiros e impressoras, sendo que nessa vertente o seu único grande problema é o facto de não suportar os sistemas operativos Windows mais recentes. Portanto, o upgrade para o Samba 4 pode nem sempre ser viável, a não ser que os serviços de Active Directory façam parte dos requisitos do cliente que é o mais provável.

Posto isto, o grande problema desta solução atual da IPBrick é sem dúvida a dependência pelos Microsoft Windows Servers para fornecer serviços de Active Directory, que implica enormes custos adicionais para os seus clientes, e a falta de uma total interoperabilidade entre ambientes Microsoft (Windows) e IPBrick (Linux). Outras limitações importantes de realçar são também a falta de suporte para Gestão de Políticas de Domínio e Backup de Controladores de Domínio.

Sendo um projeto em âmbito empresarial, onde inúmeras empresas estão envolvidas e totalmente dependentes dos servidores IPBrick que utilizam, a solução para este Problema não passa apenas por criar uma nova IPBrick OS com o Samba 4 integrado, mas também por garantir a conservação da integridade das soluções oferecidas pela empresa, ou seja, assegurar que os clientes vão poder usufruir das novas funcionalidades mas sem perder as atuais, e garantir a possibilidade de migração de toda a informação.

3.3 Solução proposta

A solução para o Problema passa claramente pela integração do Samba 4 na plataforma IPBrick OS de modo a disponibilizar um serviço *open source* de Active Directory. Esta integração irá substituir o atual Samba 3 pelo Samba 4, portanto será necessário garantir que todos os serviços da IPBrick OS já existentes, discutidos nas duas secções anteriores, continuem operacionais, com principal preocupação sobre os apresentados na Figura 3.3 que irão mesmo ser assumidos, fornecidos e controlados pelo Samba 4, à exceção do servidor de DHCP que será uma aplicação externa.

Depois de assegurar que a integração do Samba 4 com a IPBrick OS ao nível do sistema está completa, isto é, com todos os seus serviços a funcionarem corretamente, a atual interface web da IPBrick OS será adaptada de modo a acrescentar as novas funcionalidades e para que seja possível a gestão e administração do AD, de forma simples e intuitiva.

Posto isto, após investigar e analisar possíveis abordagens para integrar o Samba 4 com a IPBrick OS que garantam a conservação do funcionamento de todos os seus serviços para além de adicionarem as novas funcionalidades, definiram-se duas hipóteses numa primeira análise consideradas viáveis, que se distinguem pelo tipo de *backend* de LDAP que o Samba 4 irá utilizar.

A primeira hipótese consiste em manter o atual serviço de LDAP da IPBrick OS, o OpenLDAP, configurando o Samba 4 para utilizá-lo como o seu *backend* de LDAP. Já a segunda hipótese implica substituir o OpenLDAP da IPBrick OS por um novo servidor LDAP, o LDAP interno do Samba 4 que já vem integrado aquando a instalação do mesmo.

A primeira abordagem seria a ideal pois implicaria menos esforços tornando o processo de integração muito mais simples e muito menos suscetível a erros, pelas seguintes razões:

- Não seria necessário remover do sistema o OpenLDAP e todas as ligações aos serviços associados ao mesmo;
- Não seria necessário adaptar todos os serviços que recorrem ao OpenLDAP ao novo LDAP interno do Samba 4;
- Não seria necessário estender e alterar o *schema* do OpenLDAP de acordo com a nova semântica do LDAP interno do Samba 4;
- Não seria necessário migrar as bases de dados de OpenLDAP dos clientes atuais para a nova base de dados do LDAP interno do Samba 4.

No entanto, após uma análise mais profunda verificou-se que esta primeira hipótese não é aceitável pois apresenta uma grande limitação: não é compatível com o Samba 4 no modo Active Directory Domain Controller. Isto é afirmado pela própria equipa do Samba [11].

O provisionamento do Samba 4 com o OpenLDAP como *backend* é possível, contudo as comunicações vão ser sempre feitas ao serviço de LDAP interno do Samba 4 pela porta 389 para um correto funcionamento como AD DC, e também o *schema* do AD será sempre forçado a ser utilizado. Para além disso, esta hipótese não suporta o serviço de DRS (*Directory Replication Service*), que essencialmente replica os dados de um AD em todos os seus controladores de domínio [12]. Esta abordagem só funcionaria com o Samba 4 configurado no modo de controlador de domínio clássico (NT4 PDC [13]), o que não acrescentaria qualquer vantagem visto ser uma configuração idêntica à atual disponibilizada pela IPBrick OS através do Samba 3, portanto esta hipótese está fora de questão.

Posto isto, resta-nos a segunda hipótese: instalar o Samba 4 na IPBrick OS e configurá-lo como um AD DC com o seu LDAP interno que vem de origem a funcionar como *backend* substituindo o atual OpenLDAP.

Esta é sem dúvida a abordagem mais complexa e trabalhosa pelas razões já enunciadas anteriormente, no entanto é a que realmente tira partido das vantagens do Samba 4 e a que se apresenta como uma solução válida para o Problema, sendo capaz de assegurar o funcionamento (com melhorias) de todos os serviços atuais da IPBrick OS e de disponibilizar um inovador serviço de Active Directory com todas as funcionalidades inerentes ao mesmo.

Estando escolhida a solução proposta, que se encontra representada na Figura 3.4, pôde-se então dar início à parte da implementação.

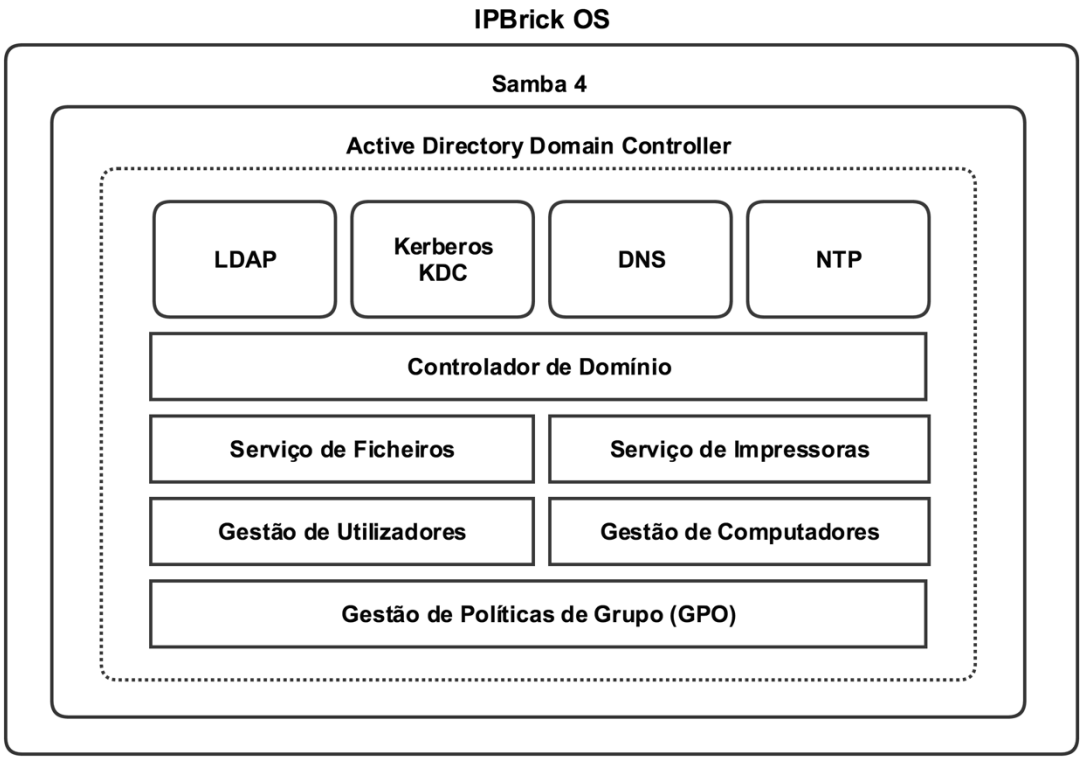

Figura 3.4 – Solução proposta para o Problema: IPBrik OS + Samba 4 AD DC

Capítulo 4

Implementação

Neste capítulo é descrita a forma de como este projeto foi desenvolvido.

É apresentada a plataforma de desenvolvimento através da qual foi realizada a implementação, as ferramentas de suporte ao desenvolvimento mais importantes, a estrutura do sistema operativo IPBrick OS que serviu como base para a implementação, e a arquitetura e descrição de todos os procedimentos necessários para a obtenção da solução final.

4.1 Introdução

De uma maneira geral, a implementação deste projeto consistiu essencialmente em duas fases.

A primeira fase, muito crítica pois sem estar concluída com sucesso não se poderia avançar, foi a instalação e configuração do Samba 4 no sistema operativo IPBrick OS. Teve-se que assegurar uma instalação dos pacotes do Samba 4 bem sucedida que não entre em conflito com os outros pacotes e serviços do sistema. Após isso, foi necessário garantir todo o correto funcionamento das funcionalidades do Samba 4 respetivamente aos objetivos desejados, e principalmente configurá-lo para atuar como um Active Directory Domain Controller.

Tendo uma instalação limpa (sem erros) do sistema operativo IPBrick OS com o Samba 4 embutido e pronto para providenciar qualquer uma das suas funcionalidades de forma bem sucedida, passou-se para a segunda fase da implementação.

Esta segunda fase consistiu então na verdadeira integração completa do Samba 4 na IPBrick OS.

Adaptou-se e alterou-se o código fonte da IPBrick, de modo a que quando esta for configurada para atuar como um Active Directory Domain Controller, fornece os devidos

serviços recorrendo ao Samba 4, e todos os outros serviços da IPBrick que não dependem do Samba 4 também se mantêm funcionais.

Durante toda a implementação, desenvolveram-se também, paralelamente a estas duas fases, procedimentos para migrar um Windows Server AD DC para uma IPBrick AD DC, migrar um Samba NT4 DC para uma IPBrick AD DC, criar um Backup DC, gerir Políticas de Grupo do domínio, entre outros.

4.2 Plataforma de desenvolvimento

Este projeto foi totalmente desenvolvido nas instalações da empresa IPBrick SA, onde foi facultado todo o material necessário ao seu desenvolvimento. Todos os componentes foram desenvolvidos num terminal com sistema operativo Ubuntu e posteriormente instalados sobre uma plataforma IPBrick versão 6.1.

Assim, a empresa disponibilizou todos os ficheiros que constituem o código fonte da IPBrick OS para serem analisados e servirem como base à implementação.

É importante referir que o sistema de ficheiros e contas de utilizadores estão alojados num servidor próprio para o efeito, possibilitando a sincronização em tempo real dos ficheiros criados no terminal de desenvolvimento com os ficheiros instalados na IPBrick, através do comando “mount”. Desta forma, durante todo o desenvolvimento foi possível verificar e depurar a execução do código escrito.

4.2.1 Ferramentas de suporte ao desenvolvimento

De todas as ferramentas usadas durante o desenvolvimento destacam-se o Archipel, a linha de comandos, o NetBeans e o RSAT.

O Archipel é uma solução para gerir e supervisionar máquinas virtuais. Foi através deste programa que se criaram as máquinas com o sistema operativo IPBrick necessárias para os vários cenários de implementação e testes que se efetuaram. Uma ferramenta muito importante devido à segurança que transmite ao utilizador ao assegurar que nenhuma máquina ou servidor físico é danificado (visto tratar-se de uma virtualização), e depois graças à sua valiosíssima utilidade de criar *snapshots* permitindo reverter a máquina a qualquer momento para um dado *snapshot*.

A linha de comandos do Linux foi sem dúvida a ferramenta mais utilizada em todo o projeto. A primeira fase da implementação, que foi a instalação e configuração do Samba 4 na IPBrick, foi feita recorrendo praticamente apenas à linha de comandos. Foi uma ferramenta crucial para testes e validações do projeto, utilizou-se para instalar ou atualizar pacotes do sistema, fazer pesquisas ou alterações no LDAP, aceder à base de dados da IPBrick, correr scripts que foram sendo desenvolvidos, fazer *debug*, entre muitas outras funções.

O NetBeans é um IDE (*Integrated Development Environment*) para desenvolvimento de software que suporta linguagens como HTML, PHP, Javascript, C/C++, entre outras. Uma das suas funcionalidades mais importantes é a capacidade de se adaptar à linguagem de programação em que se está a trabalhar, conseguindo apresentar sugestões para completar automaticamente a escrita de variáveis e funções. Esta ferramenta foi indispensável na segunda fase da implementação, que consistiu na integração completa do Samba 4 com a IPBrick, tendo sido utilizado para analisar e alterar, consoante as necessidades, o código fonte do sistema operativo IPBrick OS. Visto tratar-se de uma quantidade de código muita extensa, com bastantes pastas e ficheiros, este programa foi crucial com a sua capacidade de gestão de projetos, permitindo visualizar o histórico de alterações do código e reverter se necessário, e proporcionando um sistema de pesquisa muito rápido para encontrar tudo o que se desejasse, tal como bibliotecas, classes, variáveis ou funções.

O RSAT (*Remote Server Administration Tools*) é um programa do Windows que contém o melhor conjunto de ferramentas para administrar um Active Directory. Tem uma interface simples e intuitiva e providencia todas as tarefas de administração possíveis respetivas a um AD.

Das inúmeras funcionalidades desta ferramenta, destacam-se as que foram mais utilizadas:

- Gestão de utilizadores, grupos e computadores
- Gestão de políticas de grupo
- Gestão de DNS
- Gestão de impressoras
- Editor de ADSI (*Active Directory Service Interfaces*)

Assim, este programa foi preponderante ao nível de validações e testes do Samba 4 e da solução final, permitindo verificar sempre se as configurações estavam a atuar como esperado e também certificar a interoperabilidade entre Windows e Linux.

Convém ainda referir os dois métodos de *debug* mais utilizados durante todo o desenvolvimento, muito úteis para detetar todo o tipo de erros que foram surgindo. O primeiro método consiste na leitura e análise de todo o tipo de *logs* do sistema recorrendo ao comando “tail”, e o segundo é a inserção da função de PHP “error_log()” em pontos estratégicos do código para localizar os erros e a sua causa.

4.2.2 Estrutura da IPBrick

Uma vez que o objetivo do projeto consiste em integrar o Samba 4 na IPBrick OS, este sistema operativo apresenta-se como o componente principal de todo este trabalho. Sendo assim, esta secção pretende caracterizar a sua estrutura e os seus componentes que, de uma maneira ou de outra, estão diretamente ligados ao projeto desenvolvido. Será analisada a interface da IPBrick, a estrutura e constituição da sua base de dados e o modo como a comunicação entre estes dois componentes é realizada.

Uma máquina IPBrick funciona de maneira a que todas as modificações necessárias de efetuar no sistema tenham de ser registadas em bases de dados, validadas e só depois implementadas. Além disso, a introdução de dados numa base de dados IPBrick implica uma chamada a uma biblioteca IPBrick que, por sua vez, utiliza uma classe para acesso a funções de manipulação de elementos de tabelas, pelo que todo o percurso de configuração de um sistema pode ser altamente controlado e gerido desde a introdução de dados até à modificação de configurações do sistema.

4.2.2.1 Interface

A interface Web da IPBrick é baseada em HTML, PHP E AJAX e é composta essencialmente pelos links disponibilizados por um menu que fornece acesso às opções para configuração dos vários serviços. Assim, a interface atua como sendo o meio que exibe as informações recolhidas a partir da base de dados.

O acesso à interface é feito através da inserção do endereço IP ou FQDN como URL no browser. Feito isto, irá exigir que o utilizador obtenha uma sessão autenticada através de um formulário onde devem ser preenchidos os respetivos dados (utilizador e palavra-passe).

A partir deste ponto a interface estará pronta para permitir ao utilizador a configuração, através de formulários e campos de seleção, dos vários serviços que são disponibilizados pela IPBrick.

A interface possui uma opção que permite ao utilizador aplicar as alterações efetuadas, sendo que neste caso é feita uma sincronização do sistema com a base de dados e todas as modificações são lá armazenadas.

4.2.2.2 Base de dados

A solução IPBrick utiliza o sistema PostgreSQL para gestão das suas bases de dados e correspondentes tabelas. Estas bases de dados armazenam desde todas a informações que estão diretamente ligadas à construção da interface gráfica com a qual o utilizador interage às configurações de todos os serviços do sistema operativo. As configurações presentes nestas bases de dados aquando a ordem a aplicação de modificação serão as aplicadas a todo o sistema, estando a manipulação de ficheiros, execução de comandos e validação de alguns registos à responsabilidade do software IPBrick. Utilizadores, máquinas, grupos e partilhas

criadas, impressoras adicionadas, regras de DNS, regras de Firewall, domínio e endereço IP do servidor, são exemplos de informações que as bases de dados IPBrick armazenam.

4.2.2.3 Classes de acesso à base de dados

Como enunciado anteriormente, a modificação de qualquer tabela nunca é direta mas sim feita com recurso a classes com funções próprias. Esta camada garante não só a tradução de parâmetros em *queries* como também implementa, em certos casos, verificação desses mesmos parâmetros. Desta maneira é também limitada a utilização e criação de *queries* a uma camada controlada pela IPBrick, garantido segurança contra possíveis abusos como é o caso de tentativas de *SQL injection*. Estas funções podem também manipular uma tabela de alterações que regista os serviços que sofreram modificações e que, numa aplicação de configurações, terão de ser revistos, garantindo uma gestão de recursos mais eficaz por não haver necessidade de revisão de todos os serviços aquando a introdução de modificações no sistema. É assegurada pela IPBrick classes de acesso e manipulação para todas as tabelas existentes, ou seja, existe uma classe para cada serviço com todas as *queries* que efetuam os pedidos de informação alojada nas bases de dados.

4.2.2.4 Bibliotecas

As bibliotecas IPBrick serão o ponto de comunicação entre o sistema IPBrick e o Samba 4. São responsáveis pela gestão das chamadas às classes de acesso e estão divididas por serviços. Esta camada da estrutura é maioritariamente responsável pela análise dos parâmetros fornecidos, reestruturação desses mesmos parâmetros para envio numa chamada a uma classe de acesso, verificação do estado do sistema e obtenção de informações relevantes. É nestas bibliotecas que toda a lógica se encontra e onde é feita a maior filtragem de pedidos com valores corretos ou inválidos.

4.3 Instalação e configuração do Samba 4 na IPBrick OS

Esta secção apresenta todos os desenvolvimentos e procedimentos que se efetuaram na primeira fase da implementação, que como já foi dito teve como principal objetivo instalar o Samba 4 no sistema operativo IPBrick OS e configurá-lo de forma a funcionar como um *Active Directory Domain Controller*.

4.3.1 Cenário de implementação e testes

Durante toda a implementação houve a necessidade de planear e montar vários cenários de rede, recorrendo a várias máquinas, para que se pudessem efetuar todo o tipo de testes e validações.

A Figura 4.1 consiste num resumo de todos esses cenários e apresenta as principais máquinas que foram utilizados ao longo do desenvolvimento.

Após a instalação do sistema operativo IPBrick OS, o primeiro procedimento é atribuir um nome, um IP e um domínio para a respetiva máquina. O mesmo se aplica às máquinas Windows que se juntaram ao domínio.

Assim definiram-se as seguintes configurações de rede:

Servidor IPBrick – Active Directory Domain Controller 1				
Sistema operativo	**Nome**	**Domínio**	**Endereço IP**	**FQDN**
IBrick OS v6.1	DC1	jcastro.pt	172.31.3.185	dc1.jcastro.pt

Servidor IPBrick – Active Directory Domain Controller 2				
Sistema operativo	**Nome**	**Domínio**	**Endereço IP**	**FQDN**
IBrick OS v6.1	DC2	jcastro.pt	172.31.3.186	dc2.jcastro.pt

Cliente Windows		
Sistema operativo	**Domínio**	**Endereço IP**
Windows XP/7/8/10	jcastro.pt	172.31.3.56

Cliente Linux – IPBrick AD Member		
Sistema operativo	**Domínio**	**Endereço IP**
IPBrick OS v6.1	jcastro.pt	172.31.3.57

A escolha do nome de domínio foi arbitrária e os endereços de IP atribuídos foram os disponibilizados pela empresa para este projeto.

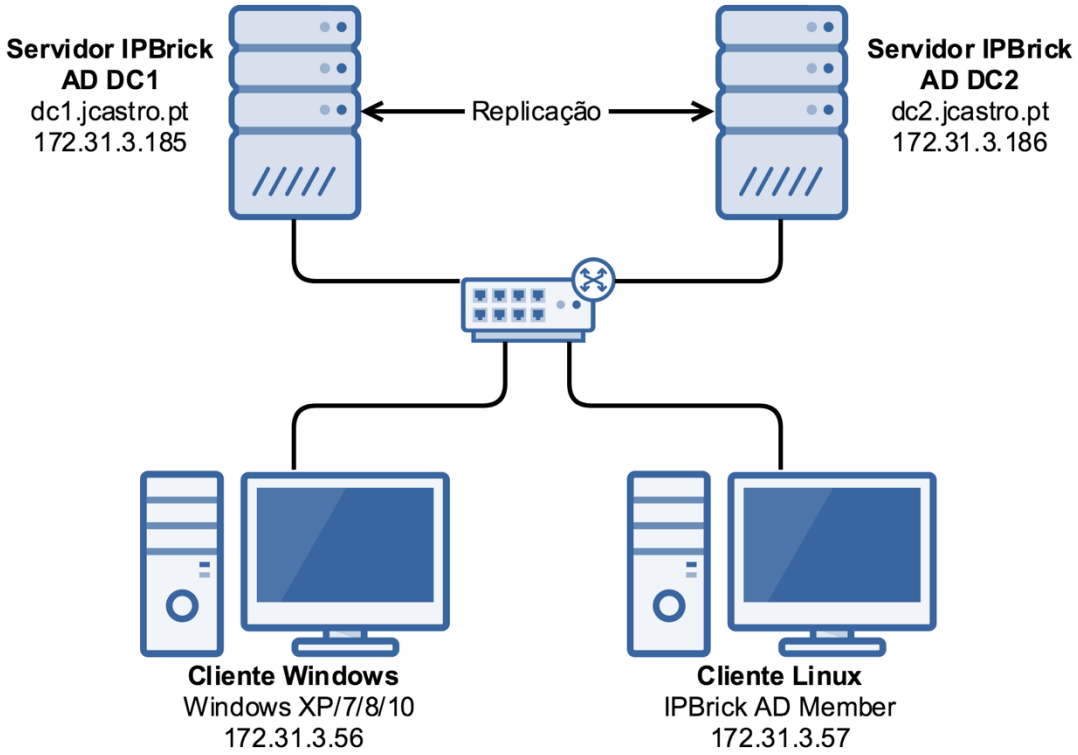

Figura 4.1

A máquina DC1 foi onde se realizaram todos os desenvolvimentos, testes e validações de modo a obter a solução final, uma IPBrick Active Directory Domain Controller.

A máquina DC2 utilizou-se essencialmente para avaliar o comportamento de múltiplos controladores de domínio e a replicação de dados entre eles.

Através da máquina Windows validou-se praticamente todo o funcionamento da IPBrick a atuar como AD DC e verificou-se a sua interoperabilidade com o Windows.

Para testar a correta interação de um cliente Linux com a IPBrick AD DC, utilizou-se uma máquina IPBrick a atuar como AD Member.

Todas estas propriedades e funcionalidades serão abordadas mais à frente.

4.3.2 Escolha da versão do Samba a instalar

O primeiro passo foi fazer o download da versão mais recente do Samba através da sua página *web* oficial [4]. Uma versão mais recente implica sempre melhoramentos e correções ao nível de *bugs* e segurança.

Sendo a IPBrick OS um sistema operativo baseado em Linux Debian versão 7 (Debian Wheezy), escolheu-se a última versão do Samba compatível esse sistema, o Samba v4.1.17.

O ideal seria instalar o Samba sobre um Debian versão 8 (Debian Jessie) ou Debian versão 9 (Debian Stretch) que suportam as versões mais atuais do Samba que neste momento vai na versão 4.4.4, no entanto isso requer uma atualização de todo o sistema operativo da IPBrick para um Debian mais recente, o que implica um árduo e extenso trabalho impensável relativamente à duração deste projeto e que já saí fora do âmbito do mesmo. Portanto esta questão é sem dúvida um aspeto a ter em conta em termos de desenvolvimentos futuros.

4.3.3 Requisitos para instalação do Samba 4

Para uma instalação bem sucedida do Samba 4 teve-se que cumprir com alguns requisitos, garantindo assim um completo funcionamento do mesmo, sem erros e sem incompatibilidades.

4.3.3.1 Suporte do sistema de ficheiros

Para uma execução apropriada do Samba 4, com todas as funcionalidades avançadas do mesmo quando atua como um *Active Directory Domain Controller*, é necessário verificar se o sistema de ficheiros da IPBrick OS é capaz de suportar ACLs (*Access Control List*) e *extended attributes*, sendo que estas extensões estão relacionadas com as permissões de ficheiros.

O sistema de ficheiros da IPBrick é do tipo "ext4" e, por defeito, já vem corretamente configurado, portanto não foi necessário efetuar alterações.

4.3.3.2 Dependências de pacotes e bibliotecas

Ao instalar todas as dependências do Samba 4 que são fornecidas diretamente pela distribuição GNU/Linux assegura-se um valioso suporte por parte da comunidade do Debian para manter o sistema atualizado contra problemas de segurança.

Pode-se dividir as dependências para esta instalação do Samba em dois grupos. O primeiro grupo, que é comum a todos os sistemas operativos e que consiste nos seguintes pacotes e bibliotecas:

- Necessários – python, perl
- Opcionais, mas recomendados – acl, xattr, python-crypto, blkid, gnutls, readline, cups, bsd/setproctitle, xsltproc/docbook, pam-devel/libmap0g-dev, openldap, ntp

O segundo grupo corresponde especificamente à distribuição Debian e consiste nas seguintes dependências:

- Acl, attr, autoconf, bison, build-essential, debhelper, dnsutils, docbook-xml, docbook-xsl, flex, gdb, krb5-user, libacl1-dev, libaio-dev, libattr1-dev, libblkid-dev, libbsd-dev, libcap-dev, libcups2-dev, libgnutls-dev, libjson-perl, libldap2-dev, libncurses5-dev, libpam0g-dev, libparse-yapp-perl, libpopt-dev, libreadline-

dev, perl, perl-modules, pkg-config, python-all-dev, python-dev, python-dnspython, python-crypto, xsltproc, zlib1g-dev

De todas estas dependências há algumas que devem ser destacadas.

"Acl" e "attr" são necessárias para um correto funcionamento das Windows ACLs responsáveis pelas permissões de ficheiros.

O pacote do sistema de impressão "cups" que neste caso vai ser a base do servidor de impressoras do Samba.

O "krb5-user" indispensável para o protocolo de autenticação Kerberos que a que o Samba recorre.

Por último, o "ntp" que consiste no *Network Time Protocol*, e que é usado para manter a sincronização do relógio do servidor altamente precisa, absolutamente necessário para os serviços *Active Directory* e Kerberos do Samba.

Para efeitos de configuração, testes e validação do Samba as dependências poderiam ser todas instaladas de uma só vez simplesmente através do comando "apt-get install", sendo o APT (*Advanced Packaging Tool*) um programa de administração de pacotes do Debian. No entanto, no contexto de uma implementação real para a empresa, o objetivo é seguir todos os procedimentos estandardizados para a criação de um *update* para a IPBrick OS que instale e configure corretamente o Samba. Esta abordagem implica a utilização de outra ferramenta de gestão de pacotes, o "dpkg", que requer a obtenção prévia dos ficheiros de instalação para que esta possa ser realizada, e que é muito útil pois este programa é capaz de fornecer toda a informação relativa aos pacotes instalados. Assim, o primeiro passo foi obter todos os ficheiros necessários respetivos às dependências. Tendo em conta que a IPBrick OS já contém vários dos pacotes e bibliotecas em questão instalados, teve-se que identificar apenas os em falta e os desatualizados e proceder ao seu download, e para isso recorreu-se ao APT. Com isto garante-se que o *update*, em termos de tamanho, é o menor possível, e obviamente que a sua instalação é mais rápida. De seguida, procedeu-se com o registo dos *MD5 sums*, que são *strings* de 32 bytes únicas, de todos os ficheiros através do programa "md5sum". Este método da IPBrick assegura um controle sobre os pacotes do seu sistema operativo e a integridade dos mesmos. Após estes dois passos, pôde-se então passar ao processo de instalação das dependências através do "dpkg", que foi incorporado no script de instalação do Samba que vai ser abordado já de seguida.

4.3.4 Script de instalação do Samba 4

Com a análise de requisitos concluída, e com todos os procedimentos necessários relativamente aos mesmos validados, procedeu-se então com a criação de um *update* para o sistema operativo IPBrick OS, denominado "specific_update_21_v61".

Este *update* incorpora variadíssimos ficheiros necessários para a atualização de um sistema operativo, e que vão de encontro à abordagem habitual da empresa quando efetua atualizações do seu sistema, mas o que realmente interessa no que diz respeito a este projeto é o *script* de instalação do Samba 4 que se desenvolveu e que é componente principal deste *update*.

O *script* implementou-se na linguagem de programação Bash e executa sequencialmente um conjunto de ações das quais se destacam as seguintes:

1. Paragem dos serviços do sistema "samba3", "BIND9" (Serviço de DNS), "slapd" (Serviço de OpenLDAP) e "winbind", que iriam entrar em conflito com a execução do Samba 4 pois este já possui os seus próprios serviços de DNS, LDAP e Winbind.

```
# /etc/init.d/samba3 stop
# /etc/init.d/BIND9 stop
# /etc/init.d/slapd stop
# /etc/init.d/winbind stop
```

2. Remoção dos pacotes "samba3" e "winbind", visto não serem mais necessários. Optou-se por não eliminar os pacotes do OpenLDAP, pois é um dos grandes pilares da IPBrick, podendo haver a necessidade de recorrer novamente aos seus serviços, e também porque contém ficheiros que irão ser utilizados mais à frente, como é o caso do *schema* da IPBrick. De qualquer forma o serviço está desativado e não prejudica a execução do Samba4.

```
# dpkg --purge samba3
# dpkg --purge winbind
```

3. Instalação dos pacotes do Samba4 e de todas as suas dependências. (O excerto de código seguinte contém apenas uma pequena parte de todos os pacotes instalados.)

```
# dpkg -E --force-overwrite,overwrite-dir --refuse-downgrade
${INSTALL_SRC}/files/bind9_1%3a9.9.5.dfsg-4~bpo70+1_amd64.deb \
# ${INSTALL_SRC}/files/bind9utils_1%3a9.9.5.dfsg-4~bpo70+1_amd64.deb \
# ${INSTALL_SRC}/files/cups_1.5.3-5+deb7u6_amd64.deb \
# ${INSTALL_SRC}/files/cups-bsd_1.5.3-5+deb7u6_amd64.deb \
# ${INSTALL_SRC}/files/cups-client_1.5.3-5+deb7u6_amd64.deb \
# (…)
# ${INSTALL_SRC}/files/samba_2%3a4.1.17+dfsg-1~bpo70+1_amd64.deb
# ${INSTALL_SRC}/files/samba-common_2%3a4.1.17+dfsg-1~bpo70+1_all.deb \
# ${INSTALL_SRC}/files/samba-common-bin_2%3a4.1.17+dfsg-1~bpo70+1_amd64.deb \
# ${INSTALL_SRC}/files/samba-dsdb-modules_2%3a4.1.17+dfsg-1~bpo70+1_amd64.deb \
# ${INSTALL_SRC}/files/samba-libs_2%3a4.1.17+dfsg-1~bpo70+1_amd64.deb \
# ${INSTALL_SRC}/files/samba-vfs-modules_2%3a4.1.17+dfsg-1~bpo70+1_amd64.deb \
# ${INSTALL_SRC}/files/smbclient_2%3a4.1.17+dfsg-1~bpo70+1_amd64.deb \
# ${INSTALL_SRC}/files/smbldap-tools_0.9.7-1+deb7u1_all.deb \
# ${INSTALL_SRC}/files/tdb-tools_1.3.0-1.1~bpo70+1_amd64.deb \
# ${INSTALL_SRC}/files/winbind_4.1.17+dfsg-1-bpo70+1_amd64.deb
```

Posto isto, avançou-se com a execução do update "specific_update_21_v61" na IPBrick OS, passando-se a ter agora uma IPBrick com o Samba v4.1.17 corretamente instalado.

Esta IPBrick atualizada será a base de desenvolvimento do resto do projeto, é nela que se vai efetuar todas as configurações necessárias e toda a integração entre o Samba 4 e a mesma.

4.4 Configuração do Samba 4 para atuar como um Active Directory Domain Controller

É neste ponto que se vai configurar o Samba 4 de modo a que o servidor IPBrick OS herde a valiosa funcionalidade inovadora deste projeto, que é a capacidade de atuar como um controlador de domínio que é compatível com o Active Directory da Microsoft. Isto é possível pois esta versão do Samba inclui o seu próprio serviço de autenticação Kerberos KDC e a sua própria implementação LDAP para todos os processos backend, tornando-se assim num verdadeiro Active Directory.

4.4.1 Requisitos

Antes de se passar ao provisionamento do Samba 4 Active Directory, os seguintes requisitos tiverem que ser validados:

- **Endereço de IP estático.**

  A alteração do IP de um servidor como controlador de domínio durante o seu funcionamento pode causar problemas relacionados com o DHCP.

  Este requisito é automaticamente assegurado pela IPBrick, pois quando se instala o seu sistema operativo, a primeira configuração é a atribuição de um IP e de um domínio.

```
auto eth0
iface eth0 inet static
     address 172.31.3.185
     netmask 255.255.255.0
     network 172.31.3.0
     broadcast 172.31.3.255
     gateway 172.31.3.254
```

- **Correta resolução do nome do servidor (hostname) em relação ao seu IP.**

  Este requisito também é previamente cumprido pela IPBrick.

```
127.0.0.1       localhost
172.31.3.185      dc1.jcastro.pt     dc1
```

- **Nome de domínio e NetBIOS apropriados.**

  O Samba Active Directory não suporta a alteração do nome de domínio, portanto deve ser uma escolha muito bem ponderada.

  Este requisito enquadra-se mais ao nível do cliente, no entanto aborda algumas questões consideradas relevantes.

  - O nome de domínio deve ser globalmente único assegurando-se assim uma correta resolução DNS do nome.
  - O nome de domínio deve estar relacionado com a organização em causa de modo a ser mais facilmente "memorizável".

- O nome de domínio deve estar registado deve modo a prevenir utilizações maliciosas do mesmo.
- O nome NetBIOS deve ter no máximo 15 carateres sem qualquer pontuação, de acordo com as convenções da Microsoft.

4.4.2 Provisionamento do Samba 4 Active Directory Domain Controller

Tendo os requisitos validados, e as definições de rede da máquina IPBrick configuradas, pôde-se avançar com o provisionamento do Samba 4 Active Directory como controlador de domínio na IPBrick OS. Este provisionamento implica a configuração dos serviços internos do Samba 4: LDAP, Kerberos e DNS. E também a realização de todas as configurações básicas do Active Directory.

Trata-se de um processo simples, pois o Samba 4 contém uma ferramenta de administração, o "samba-tool", que realiza o provisionamento de uma só vez após o fornecimento de certos parâmetros de configuração. O "samba-tool" é executado através da consola, e o comando em questão é o seguinte:

```
# samba-tool domain provision
```

Este comando requer a estipulação de vários parâmetros e opções de provisionamento, consoante as necessidades. Segue-se a explicação desses argumentos e respetivas escolhas:

- **"--use-rfc2307"** – Habilita as extensões NIS (*Network Information Service*) que permitem uma administração central dos atributos Unix (UIDs, GIDs, *shells*, etc.) no Active Directory. Isto é necessário, por exemplo, para a autenticação em Linux. Esta funcionalidade requer a extensão do *schema* do AD que será abordada já de seguida.
- **"--use-xattrs=yes"** – Habilita a utilização dos Unix *extended attributes* (ACLs) para os ficheiros do servidor. É uma opção muito importante para o correto funcionamento do Samba 4 como servidor de ficheiros e partilha de ficheiros.
- **"--realm"** – Corresponde ao Kerberos Realm. Deve ser preenchido exatamente com o mesmo nome do domínio DNS do AD em letras maiúsculas. De acordo com o nome definido em (), o nome atribuído foi "JCASTRO.PT".
- **"--domain"** – Corresponde ao nome do domínio NetBIOS em letras maiúsculas por questões de compatibilidade. O nome atribuído foi "JCASTRO".
- **"--server-role"** – Neste contexto, a única opção logicamente válida é a função de controlador de domínio ("DC").
- **"--dns-backend"** – Este parâmetro seleciona o DNS *backend* que o AD vai utilizar. Existem duas possibilidades, o serviço de DNS interno do Samba ("SAMBA_INTERNAL") ou o módulo de DNS BIND9 ("BIND9_DLZ"). Tendo em conta que a IPBrick já possui o seu serviço de DNS implementado e também baseado no BIND9, escolheu-se a opção "BIND9_DLZ".

- **"--adminpass"** – Define a palavra-passe dos Administradores do Domínio. A mesma deve cumprir os requisitos de complexidade convencionados pela Microsoft. Se a palavra-passe não respeitar esses requisitos o provisionamento resulta num erro e não é finalizado.

Com isto, especificou-se o Active Directory com a seguinte tabela que resume as suas configurações:

Tabela 4.1

Função	Controlador de domínio (DC)
DNS *backend*	BIND9
Endereço IP	172.31.3.185
Nome de domínio	jcastro.pt
Nome de domínio NetBIOS	JCASTRO
Nome de DNS	jcastro.pt
Kerberos *realm*	JCASTRO.PT
DN do domínio do LDAP	DC=jcastro,DC=pt
Administrador do domínio	JCASTRO\Administrator

Assim, com todos os parâmetros de configuração definidos, executou-se o seguinte comando para provisionar o Samba 4 Active Directory:

```
# samba-tool domain provision --use-rfc2307 --use-xattrs=yes --realm=JCASTRO.PT --
domain=JCASTRO --server-role=DC --dns-backend=BIND9_DLZ
```

Quando o Samba configura o primeiro controlador de domínio de um domínio, o provisionamento cria uma base de dados inicial para o Active Directory. Portanto, o mesmo foi efetuado com privilégios *root*, para permitir a escrita na diretoria de instalação e a atribuição das permissões corretas nos ficheiros e pastas.

Durante o provisionamento o comando imprime na consola informação das suas várias fases, em que se pode ver mensagens relativas à configuração de alguns recursos, à criação de alguns *containers* padrão, utilizadores, grupos, assim como configurações de DNS.

As últimas linhas fornecem alguma informação útil; indicam o nome do domínio e o seu SID; também mostram a localização do ficheiro de configuração do Kerberos que é automaticamente criado.

O *output* gerado pelo comando de provisionamento foi o seguinte:

```
Looking up IPv4 addresses
Looking up IPv6 addresses
No IPv6 address will be assigned
Setting up share.ldb
Setting up secrets.ldb
Setting up the registry
Setting up the privileges database
Setting up idmap db
Setting up SAM db
Setting up sam.ldb partitions and settings
Setting up sam.ldb rootDSE
Pre-loading the Samba 4 and AD schema
Adding DomainDN: DC=jcastro,DC=pt
Adding configuration container
Setting up sam.ldb schema
Setting up sam.ldb configuration data
Setting up display specifiers
Modifying display specifiers
Adding users container
Modifying users container
Adding computers container
Modifying computers container
Setting up sam.ldb data
Setting up well known security principals
Setting up sam.ldb users and groups
Setting up self join
Adding DNS accounts
Creating CN=MicrosoftDNS,CN=System,DC=jcastro,DC=pt
Creating DomainDnsZones and ForestDnsZones partitions
Populating DomainDnsZones and ForestDnsZones partitions
Setting up sam.ldb rootDSE marking as synchronized
Fixing provision GUIDs
A Kerberos configuration suitable for Samba 4 has been generated at
/usr/local/samba/private/krb5.conf
Setting up fake yp server settings
Once the above files are installed, your Samba4 server will be ready to use
Server Role:           active directory domain controller
Hostname:              DC1
NetBIOS Domain:        JCASTRO
DNS Domain:            jcastro.pt
DOMAIN SID:            S-1-5-21-2614513918-2685075268-614796884
```

Verifica-se que o provisionamento ocorreu conforme o esperado, com sucesso.

4.4.2.1 Validação do Samba 4 AD DC

Para validar o correto funcionamento da IPBrick com Samba 4 Active Directory a atuar como controlador de domínio recorreu-se à ferramenta "smbclient", que funciona como um cliente do tipo FTP (*File Transfer Protocol*) que acede aos recursos SMB/CIFS de servidores.

Efetuaram-se os seguintes testes:

- Verificou-se que o Samba 4 está a providenciar as partilhas padrão do AD DC "netlogon" e "sysvol", que são automaticamente criadas no ficheiro de configuração do Samba 4 ("smb.conf") durante o provisionamento.

```
# smbclient -L localhost -U%
# (…)
```

- Testou-se a autenticação através da conexão à partilha "netlogon" recorrendo à conta de administrador do domínio criada também durante o provisionamento.

```
# smbclient //localhost/netlogon -UAdministrator -c 'ls'
# (…)
```

Os resultados foram os esperados, concluindo-se assim a validação.

Posto isto, faltam ainda algumas configurações relativas ao RFC2307, DNS, Kerberos e NTP.

4.4.3 RFC2307 – Instalação das extensões NIS

O protocolo RFC2307 proporciona a capacidade de armazenar informação dos utilizadores e grupos no LDAP do Active Diretory, o que resulta em inúmeras vantagens:

- Administração central dos IDs dentro do AD;
- IDs consistentes em todos os membros do domínio;
- Rápida configuração dos atributos: grupos apenas precisam da atribuição de um GID, novos utilizadores de um UID, domínio NIS, *login shell*, diretoria *home* e grupo primário;
- A administração central elimina a necessidade de mapear localmente os IDs, o que pode resultar na perda da posse dos ficheiros caso a base de dados corrompa;
- Caminhos para *login shells* e diretorias *home* individuais para os utilizadores;
- As configurações de *login shell* e diretoria *home* são as mesmas em todos os membros do domínio;
- Fácil gestão de utilizadores e grupos através da ferramenta *Active Directory Users and Computers* (ADUC) que faz parte do RSAT.

A utilização deste protocolo foi selecionada no provisionamento do AD, e para o seu correto funcionamento é necessário estender o seu *schema* tal como já foi dito.

Para isso é preciso instalar as extensões NIS, que são um conjunto de atributos e classes para LDAP definidos pelo RFC2307, e que se encontram na Tabela 4.2.

Tabela 4.2

Atributos	**Classes**
uidNumber	posixAccount
gidNumber	shadowAccount
gecos	posixGroup
homeDirectory	ipService
loginShell	ipProtocol
shadowLastChange	oncRpc
shadowMin	ipHost
shadowMax	ipNetwork
shadowWarning	nisNetgroup

shadowInactive	nisMap
shadowExpire	nisObject
shadowFlag	ieee802Device
memberUid	bootableDevice
memberNisNetgroup	
nisNetgroupTriple	
ipServicePort	
ipServiceProtocol	
ipProtocolNumber	
oncRpcNumber	
ipHostNumber	
ipNetworkNumber	
ipNetmaskNumber	
macAddress	
bootParameter	
bootFile	
nisMapName	
nisMapEntry	

Assim, a extensão do *schema* do AD foi feita através de um ficheiro LDIF (*LDAP Data Interchange Format*), que é um tipo de ficheiro muito útil para fazer alterações no LDAP.

O procedimento foi o seguinte:

1. Criou-se uma cópia do ficheiro LDIF "ypServ30.ldif" que se encontra numa das diretorias do Samba 4.

```
# cp /usr/share/samba/setup/ypServ30.ldif /tmp/
```

2. Substituiu-se as variáveis do ficheiro LDIF de acordo com o nome do domínio do AD.

```
# sed -i -e 's/${DOMAINDN}/DC=jcastro,DC=pt/g' /tmp/ypServ30.ldif
# sed -i -e 's/${NETBIOSNAME}/jcastro/g' /tmp/ypServ30.ldif
# sed -i -e 's/${NISDOMAIN}/jcastro/g' /tmp/ypServ30.ldif
```

3. Alterou-se o LDAP do Samba 4, recorrendo à sua ferramenta de base de dados LDB, adicionando assim o novo *schema*.

```
# ldbmodify -H /var/lib/samba/private/sam.ldb /tmp/ypServ30.ldif --
option="dsdb:schema update allowed"=true
```

A informação contida no ficheiro "ypServ30.ldif", com todos os atributos e classes envolvidos nesta extensão do *schema*, pode ser visualizado no Anexo.

4.4.4 Configuração do serviço de DNS

O serviço de DNS é essencial para a correta operação de um Active Directory. Por exemplo, sem as entradas corretas de DNS, o serviço Kerberos não irá funcionar, o que basicamente significa que maioria das funcionalidades básicas do AD não irão funcionar.

Primeiro foi necessário garantir que o ficheiro de configuração "/etc/resolv.conf" está corretamente definido, tendo em conta que é neste ficheiro que se atribui o DNS *resolver* do servidor.

A IPBrick com Samba 4 como controlador de domínio requer um *nameserver* que seja capaz de resolver consultas de DNS às zonas do AD. Visto que este servidor também é servidor de DNS, faz todo o sentido a utilização do seu domínio e IP no ficheiro "/etc/resolv.conf" da seguinte maneira:

```
# domain jcastro.pt
# nameserver 172.31.3.185
```

O sistema operativo IPBrick já vem com esta configuração corretamente definida por defeito.

O segundo passo foi configurar o *backend* de DNS escolhido no provisionamento, o BIND9.

Ao provisionar o AD DC teve que se escolher entre o *backend* de DNS interno do Samba 4 ou do BIND9. O DNS interno do Samba 4 tem a vantagem da sua simples e rápida configuração. No entanto a IPBrick, que impõe requisitos de DNS mais complexos, já contém o seu serviço de DNS com BIND9 instalado, portanto escolheu-se esta opção.

O BIND é um software *open source* que implementa o protocolo DNS. Este programa fornece resolução de DNS para zonas geridas no AD. Essas zonas são acedidas pelo BIND através do *plug-in* DLZ (*Dynamically Loadable Zones*).

Sendo que já se encontra instalado na IPBrick, falta apenas configurá-lo para atuar como o servidor de DNS do Samba 4 AD DC.

Durante o provisionamento do Samba 4 AD DC foi criado um ficheiro de configuração do BIND9 para o Samba 4 ("/var/lib/samba/private/named.conf"). De modo a completar a interligação entre o Samba 4 e o módulo do BIND9 foi necessário incluir esse ficheiro criado no ficheiro de configuração do BIND9 ("/etc/bind/named.conf") através do seguinte comando:

```
# include "/var/lib/samba/private/named.conf";
```

Também foi preciso editar o ficheiro de configuração do BIND9 do Samba 4 ("/var/lib/samba/private/named.conf") de modo a que o mesmo utilize a correta versão do módulo DLZ, tendo-se "descomentado" a seguinte linha do ficheiro respetiva à versão v9.8:

```
dlz "AD DNS Zone" {
    # For BIND 9.8.x
    database "dlopen /usr/lib/x86_64-linux-gnu/samba/bind9/dlz_bind9.so";

    # For BIND 9.9.x
    # database "dlopen /usr/lib/x86_64-linux-gnu/samba/bind9/dlz_bind9_9.so";

    # For BIND 9.10.x
    # database "dlopen /usr/lib/x86_64-linux-gnu/samba/bind9/dlz_bind9_10.so";
};
```

Por último, configurou-se a funcionalidade de atualizações dinâmicas do DNS.

O Samba 4 possui a capacidade de atualizar automaticamente as zonas de DNS do BIND através do Kerberos. Ao utilizar a versão v9.8 do BIND garanta-se um conjunto de *patches* realizados pela equipa do Samba que tornam as atualizações dinâmicas do DNS muito mais robustas e fáceis de configurar.

Ao provisionar o Samba 4 AD DC foi também criado um ficheiro *keytab* de DNS ("/var/lib/samba/private/dns.keytab") necessário para o serviço Kerberos. Com isto, concluiu-se esta configuração adicionando a seguinte linha na secção "options" do ficheiro de configuração do BIND9 ("/etc/bind/named.conf"):

```
# options
# {
#       (...)
#       tkey-gssapi-keytab "/var/lib/samba/private/dns.keytab ";
#       (...)
# };
```

4.4.4.1 Validação do DNS

O Active Directory utiliza o *record* SRV para localizar controladores de domínio, sendo que este *record* é utilizado para identificar servidores que proporcionam determinados serviços. Há dois *records* SRV que têm que ser criados em qualquer controlador de domínio com Active Directory: "_kerberos" and "_ldap". Posto isto, é necessário validar que o sistema contém estes *records*, de modo a que os clientes possam comunicar e encontrar os serviços do AD DC na rede.

Assim, para testar se o serviço de DNS do Samba 4 AD DC está a funcionar corretamente, executaram-se os seguintes comandos:

```
# host -t SRV _ldap._tcp.jcastro.pt.
# (…)
# host -t SRV _kerberos._udp.jcastro.pt.
# (…)
# host -t A dc1.jcastro.pt.
# (…)
```

Os comandos não retornaram erros e verifica-se que os *records* SRV para o "_ldap" e "_kerberos" estão corretamente configurados. Sendo assim, o serviço de DNS está validado.

4.4.5 Configuração do serviço Kerberos

O serviço de autenticação Kerberos é uma parte muito importante do Active Directory e o seu ficheiro de configuração encontra-se em "/etc/krb5.conf". Durante o provisionamento do

Samba 4 AD DC é gerado uma cópia, já funcional, desse ficheiro em "/var/lib/samba/private/krb5.conf". Portanto, para concluir a configuração do Samba 4 AD DC teve-se que copiar o ficheiro de configuração do Kerberos criado pelo provisionamento para a sua correta localização:

```
# cp /var/lib/samba/private/krb5.conf /etc/krb5.conf
```

Desta forma, o serviço de autenticação Kerberos do Samba 4 AD DC fica funcional.

4.4.5.1 Validação do Kerberos

Para testar a autenticação via Kerberos recorreu-se à ferramenta "kinit", que basicamente obtém e coloca em *cache* o *ticket* TGT (*Ticket-Granting Ticket*) do Kerberos. Assim, executou-se o seguinte comando:

```
# kinit administrator@JCASTRO.PT
# Password for administrator@JCASTRO.PT: …
# …
```

Este comando solicita a palavra-passe do "administrator" de modo a que a autenticação possa ocorrer. O *output* devolvido mostra que a autenticação ocorreu com sucesso.

Posto isto, verificou-se se o "administrator" realmente recebeu um *ticket*, confirmando assim o correto funcionamento do Kerberos. Para isso, recorreu-se ao comando "klist":

```
# klist
# (…)
```

Constata-se pelo *output* deste último comando que o "administrator" recebeu um *ticket*, portanto o serviço de autenticação Kerberos também está validado.

4.5 Integração do Samba 4 AD DC com a IPBrick OS

Com a primeira fase da implementação finalizada, passa-se a ter então uma IPBrick OS com o Samba 4 incorporado, com todas as suas funcionalidades validadas e com processos de configuração das mesmas definidos, estando prontas para serem postas em prática. Destaca-se o funcionamento do Samba 4 *Active Directory* como Controlador de Domínio, que é o objetivo principal deste projeto.

Sendo assim, deu-se finalmente início à segunda e última fase da implementação, que tal como já foi dito, consistiu na verdadeira integração completa do Samba 4 AD DC com a IPBrick OS, ou seja, adaptou-se o sistema operativo IPBrick OS de modo a que este utilize o Samba 4 AD DC para fornecer os seus serviços quando devidamente necessário. Esta integração ocorreu tanto ao nível de operações de sistema como ao nível de operações da interface da IPBrick.

Foi necessário alterar várias partes do código fonte da IPBrick OS, em que estiveram envolvidas várias linguagens de programação: PHP que constitui maior parte do código da IPBrick; Bash utilizado nos *scripts* criados para o Samba 4; HTML para modificações da interface; e SQL para as bases de dados da IPBrick.

Antes de se avançar com qualquer desenvolvimento, foi fundamental identificar quais os serviços da IPBrick OS que o Samba 4 irá interagir ou até mesmo assumir controlo, quando a mesma for configurada para atuar como um controlador de domínio com *Active Directory*. Esses serviços já foram apresentados na solução proposta, sendo os seguintes:

- Protocolo LDAP;
- Servidor de DNS;
- Servidor de DHCP;
- Gestão de utilizadores e grupos de utilizadores;
  - Políticas de palavras-passe;
- Gestão de computadores e grupos de computadores;
- Servidor de ficheiros;
  - *Roaming profiles*;
- Servidor de impressão;
- Servidor de correio eletrónico;
- Serviço de *backup* do sistema (*Disaster recovery)*;
- Outros serviços como VoIP, Voice-mail, VPN e Proxy.

Posto isto, efetuou-se uma análise profunda do código do sistema operativo IPBrick OS de modo a definir uma abordagem para integrar o Samba 4 com estes serviços.

Desta análise concluiu-se que o ponto de partida para a integração seria através do módulo de autenticação do servidor IPBrick.

A partir do momento em que um utilizador é criado na IPBrick, passa a existir um registo na base de dados LDAP do servidor de autenticação. É no LDAP que é guardada a informação relativo aos recursos informáticos da empresa e seus utilizadores. Sempre que um utilizador se pretende autenticar num determinado serviço com o seu nome de utilizador e palavra-passe, é feita uma consulta à base de dados LDAP da IPBrick para efetivamente validar ou não o acesso.

A IPBrick OS permite vários modos de autenticação, estando por defeito configurada para todos os utilizadores se autenticarem na própria IPBrick. Destacam-se os seguintes:

- **IPBrick Master** – Modo pré-definido. Todos os serviços no servidor utilizam o próprio servidor LDAP.
- **IPBrick Slave** – O servidor LDAP é uma réplica sincronizada do servidor IPBrick Master, sendo este modo usado num cenário com vários servidores. Os utilizadores podem autenticar-se neste servidor, uma vez que há uma sincronização temporizada da base de dados LDAP com a IPBrick Master, não

havendo, no entanto, a possibilidade de adicionar utilizadores. Este tipo de autenticação é útil nos seguintes cenários;

  - Redes com elevado número de utilizadores onde existem muitas autenticações;
  - Redes geograficamente distribuídas.

- **IPBrick Client** - Os serviços autenticam remotamente no servidor LDAP IPBrick indicado. Neste caso não há qualquer cópia da base de dados localmente, sendo necessário especificar qual o servidor IPBrick Master/Slave. Normalmente este modo de autenticação é usado no âmbito dos serviços VPN, PPTP e Proxy.
- **AD Domain Member** – IPBrick é um membro de um domínio gerido por um servidor Windows com *Active Directory*. Os utilizadores da rede necessitam, como sempre, de se autenticarem no AD. É de salientar que este servidor Windows deixa de ser necessário com a implementação da IPBrick AD DC.

Os modos de autenticação podem ser configurados pela interface *web* da IPBrick em "Configurações avançadas → IPBrick → Autenticação".

Posto isto, a abordagem passou claramente por criar um novo modo de autenticação: a IPBrick AD DC. Na realidade, este modo é uma versão melhorada da IPBrick Master com a grande vantagem de atuar como um controlador de domínio com *Active Directory*, em que o OpenLDAP da IPBrick é substituído pelo LDAP interno do Samba 4.

Assim, ao configurar este novo modo de autenticação com os parâmetros necessários para o provisionamento do Samba 4 AD DC, a IPBrick entrará num novo estado de funcionamento em que recorre ao Samba 4 para fornecer os serviços referidos há pouco.

Em relação à interface optou-se por alterar o menos possível. Primeiro porque o verdadeiro foco deste projeto incide sobre uma integração plena ao nível do sistema do Samba 4 com toda a IPBrick, sendo que este objetivo implica extensos e árduos desenvolvimentos, desvalorizando assim possíveis melhoramentos da interface IPBrick. Depois porque se considerou que a interface já possui os acessos necessários para a configuração e gestão de uma IPBrick AD DC.

Da mesma forma, escolheu-se modificar o menos possível o sistema de base de dados PostgreSQL da IPBrick e respetivas tabelas, tendo-se aproveitado ao máximo as variáveis e a informação já existente para avançar com todo o tipo de configurações da IPBrick AD DC.

Com estas decisões, concentrou-se o esforço nas operações ao nível do sistema necessárias para a integração do Samba 4 com a IPBrick, reduzindo-se a probabilidade de ocorrerem erros e de se tornar numa IPBrick com incompatibilidades, ou até mesmo não funcional.

Todos os procedimentos envolvidos nesta integração e respetiva arquitetura encontram-se detalhados de seguida.

4.5.1 Modo de Autenticação IPBrick AD DC

Começou-se por especificar os parâmetros de entrada na interface *web* da IPBrick para seja possível submeter através da mesma a ativação e configuração do modo de autenticação IPBrick AD DC, sendo os seguintes:

- **Tipo de autenticação** – Esta variável corresponde ao modo de autenticação que se pretende acionar, que neste caso é a IPBrick AD DC. É um parâmetro muito importante pois vai servir como uma *flag* para todo o sistema IPBrick conseguir reconhecer que está a atuar nesse modo e fornecer as respetivas funcionalidades.
- **Domínio** – Corresponde ao domínio do servidor necessário para o provisionamento do Samba 4 AD DC. Por exemplo: "jcastro.pt".
- **Administrador do domínio** – Corresponde ao administrador de domínio que se pretende escolher. Por defeito é o "Administrador".
- **DN base** – Define o DN base do LDAP. Por defeito é:
  "dc=".str_replace(".",",dc=",$Domínio)
  Por exemplo: "dc=jcastro,dc=pt"
- **DN do Administrador** – Corresponde ao DN do administrador do LDAP, necessário para efetuar leituras e escritas no LDAP. Por defeito é:
  "cn=".$(Administrador do domínio).$(DN base)
  Por exemplo: "cn=administrador,dc=jcastro,dc=pt"

Não houve a necessidade de criar estas variáveis, pois já existiam tanto no sistema como nas bases de dados PostgresSQL da IPBrick.

Assim, criou-se um novo formulário na interface da IPBrick (em "Configurações avançadas → IPBrick → Autenticação ") para o preenchimento dos argumentos de configuração necessários para este modo de autenticação.

Ao submeter estes parâmetros através do botão "Modificar", a IPBrick chama uma função de acesso à tabela da sua base de dados correspondente ao módulo de autenticação, que guarda a informação das configurações do tipo de autenticação selecionado. É também ativada uma *flag* no sistema IPBrick que indica que o seu modo de funcionamento passou a ser o selecionado no formulário, neste caso o IPBrick AD DC.

Como já foi dito em X, as alterações feitas na interface *web* da IPBrick e consequentemente guardadas na sua base de dados, só tomam verdadeiramente efeito quando se aplica configurações do sistema em "Aplicar configurações".

Ao aplicar configurações do sistema, é executado o módulo "actualiza_def.php" da IPBrick que essencialmente deteta alterações nas configurações guardadas na base de dados,

e dependendo disso executa os procedimentos necessários para implementar essas configurações.

Posto isto, modificou-se a biblioteca "actualiza_def.php" de modo a que quando a *flag*, que informa que o tipo de autenticação IPBrick AD DC foi selecionado, estiver ativa, são realizados os procedimentos necessários para tornar a IPBrick num controlador de domínio com *Active Directory* através do Samba 4.

Estes procedimentos consistem essencialmente na execução de um *script* ("samba_provision.sh") com todas as operações necessárias para provisionamento do Samba 4 AD DC já discutido em X, e numa extensão do *schema* do LDAP do Samba 4, adicionando alguns atributos do anterior OpenLDAP da IPBrick que são necessários para o funcionamento dos serviços do sistema. Esta extensão do *schema* encontra-se detalhada na próxima subsecção ().

A Figura 4.2 descreve, com uma visão de alto nível, a arquitetura deste novo modo de autenticação.

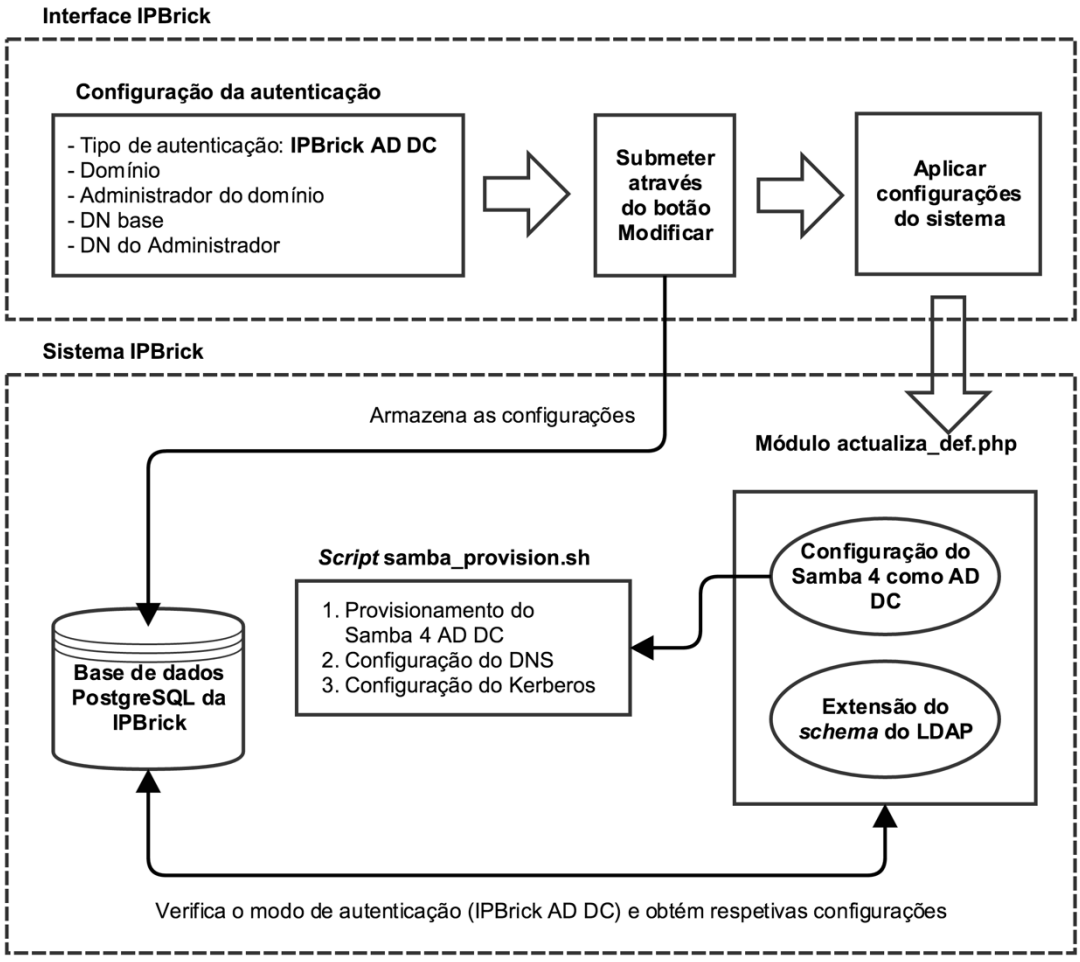

Figura 4.2

4.5.2 Extensão do Schema do LDAP do Samba 4 AD DC

Um dos alicerces principais do sistema IPBrick é o seu serviço de diretório OpenLDAP que armazena informação de muitos dos seus componentes (listados no Capítulo 3), em que os atributos e classes desses dados seguem a estrutura e formatação do *schema* definido pelo OpenLDAP.

Com integração do Samba 4 AD DC na IPBrick OS, o OpenLDAP é substituído pelo LDAP do Samba 4, e consequentemente é criado um novo *schema* baseado na versão do Windows Server 2008 R2. A criação deste *schema* é feita automaticamente durante o provisionamento do Samba 4 AD DC.

Posto isto, é óbvio que este novo *schema* não contém os atributos e classes que a IPBrick utiliza para gerir a informação dos seus serviços. Portanto, para o funcionamento dos mesmos é necessário estender o *schema* com esses atributos e classes de modo a que o AD consiga interpretar essa informação.

É muito importante ter em conta que estender o *schema* de um AD é uma operação bastante sensível, pois qualquer alteração da sua estrutura e suas definições pode levar a resultados inesperados, como a falta de atributos ou classes críticos que impossibilitam o funcionamento do serviço. Assim, o procedimento de extensão do *schema* teve que ser muito bem planeado e especificado.

Começou-se por analisar os vários ficheiros do *schema* do OpenLDAP, cada um relativo a um determinado serviço da IPBrick., os quais descrevem os atributos, classes e respetiva sintaxe para representar os objetos desses serviços no LDAP do AD.

Assim, escolheram-se os ficheiros que contêm a informação que é necessária adicionar ao novo LDAP do Samba 4 de modo a manter todos os serviços da IPBrick funcionais. Os ficheiros escolhidos foram os seguintes:

- **qmail.schema** – Indispensável para o funcionamento do serviço de correio eletrónico da IPBrick. O "qmail" é um MTA (*Mail Transfer Protocol*) *open source* para sistemas Unix, e é utilizado pela IPBrick para implementar a funcionalidade de servidor de SMTP (*Simple Mail Transfer Protocol*), ou seja, para enviar emails.
- **sipidentity.schema** – Contém informação necessária para o funcionamento dos serviços de VoIP, Proxy e Voicemail da IPBrick.
- **system.schema** – Contém informação necessária para o funcionamento dos serviços de Gestão de Computadores, Grupos de Computadores e VPN.

A extensão do *schema* voltou a ser realizada recorrendo à ferramenta LDIF tal como no ponto X. Para isso, foi necessário converter a informação dos ficheiros do OpenLDAP respeitando a sintaxe do AD do Samba 4, e assim criar os ficheiros LDIF para proceder com a extensão.

A conversão foi feita recorrendo à informação disponibilizada pelo RFC2252 relativa às definições da sintaxe dos atributos do protocolo LDAP, e pela Microsoft relativa à sintaxe dos objetos, atributos e classes do *schema* de um AD, visto que o Samba 4 utiliza um *schema* com a versão do Windows Server 2008 R2. Os detalhes da mesma encontram-se nas seguintes tabelas.

Tabela 4.3

Atributo	**ID**	**Descrição**	**Sintaxe do OpenLDAP**
IPBVpnPassword	1954, 36	PPTP VPN password	1.3.6.1.4.1.1466.115.121.1.5, Binary
IPBMachinePassword	1954, 37	Machine password	1.3.6.1.4.1.1466.115.121.1.5, Binary
IPBMachineType	1954, 38	Machine Type	1.3.6.1.4.1.1466.115.121.1.15, caseIgnore
IPBUserQuota	1954, 39	User Quota	1.3.6.1.4.1.1466.115.121.1.15, caseIgnore
IPBUserPassType	1954. 40	User Password Type	1.3.6.1.4.1.1466.115.121.1.15, caseIgnore
IPBIpAddress	1954, 41	IP address	1.3.6.1.4.1.1466.115.121.1.26{128}, caseIgnoreIA5Match
IPBMacAddress	1954, 42	MAC address	1.3.6.1.4.1.1466.115.121.1.26{128}, caseIgnoreIA5Match
IPBComputerGroupType	1954, 43	Computer Type	1.3.6.1.4.1.1466.115.121.1.27, integerMatch, SINGLE-VALUE
IPBComputerGroupNetStart	1954, 44	Computer Group Network	1.3.6.1.4.1.1466.115.121.1.26{128}, caseIgnoreIA5Match
IPBComputerGroupNetStop	1954, 45	Computer Group Mask	1.3.6.1.4.1.1466.115.121.1.26{128}, caseIgnoreIA5Match

IPBFollowMeMode	1954, 46	Follow Me Mode	1.3.6.1.4.1.1466.115.121.1.27

Na tabela 4.3 apresentam-se os atributos do ficheiro “system.schema” e respetiva sintaxe utilizada pelo OpenLDAP, que foi convertida para a sintaxe do LDAP do AD presente na tabela 4.4.

Tabela 4.4

Atributo	**Sintaxe do AD**				
	Nome	**attributeSyntax**	**oMSyntax**	**isSingleValued**	**rangeUpper**
IPBVpnPassword	String(Octet)	2.5.5.10	4		
IPBMachinePassword	String(Octet)	2.5.5.10	4		
IPBMachineType	String(Teletex)	2.5.5.4	20		
IPBUserQuota	String(Teletex)	2.5.5.4	20		
IPBUserPassType	String(Teletex)	2.5.5.4	20		
IPBIpAddress	String(IA5)	2.5.5.5	22		128
IPBMacAddress	String(IA5)	2.5.5.5	22		128
IPBComputerGroupType	Integer	2.5.5.9	2	VERDADEIRO	
IPBComputerGroupNetStart	String(IA5)	2.5.5.5	22		128
IPBComputerGroupNetStop	String(IA5)	2.5.5.5	22		128
IPBFollowMeMode	Integer	2.5.5.9	2		

A sintaxe do AD relativa à única classe do ficheiro “system.schema” encontra-se na tabela 4.5.

Tabela 4.5

Classe	**ID**	**Descrição**	**AD Syntax**		
			subClassOf	**mustContain**	**mayContain**
System	1954,1	System object	top		(todos os atributos da Tabela X)

O mesmo se aplicou para os ficheiros “sipidentity.schema” (Tabelas 4.6, 4.7 e 4.8) e “qmail.schema”.

Tabela 4.6

Atributo	**ID**	**Descrição**	**Sintaxe do OpenLDAP**
SIPIdentitySIPURI	0.0.8.350.1.1.6.1.1	Universal Resource Indicator of the SIP UA	1.3.6.1.4.1.1466.115.121.1.15, caseExact
SIPIdentityRegistrarAddress	0.0.8.350.1.1.6.1.2	specifies the location of the registrar	1.3.6.1.4.1.1466.115.121.1.26, caseIgnoreIA5Match
SIPIdentityProxyAddress	0.0.8.350.1.1.6.1.3	Specifies the location of the SIP Proxy	1.3.6.1.4.1.1466.115.121.1.26, caseIgnoreIA5Match
SIPIdentityAddress	0.0.8.350.1.1.6.1.4	IP address of the UA	1.3.6.1.4.1.1466.115.121.1.26, caseIgnoreIA5Match
SIPIdentityPassword	0.0.8.350.1.1.6.1.5	The user agent SIP password	1.3.6.1.4.1.1466.115.121.1.40, octetStringMatch
SIPIdentityUserName	0.0.8.350.1.1.6.1.6	The user agent user name.	1.3.6.1.4.1.1466.115.121.1.15, caseIgnore
SIPIdentityServiceLevel	0.0.8.350.1.1.6.1.7	To define services that a user can belong to.	1.3.6.1.4.1.1466.115.121.1.26, caseIgnoreIA5Match
SIPIdentityVoipUserAlias	0.0.8.350.1.1.6.1.8	Specifies Voip User alias	1.3.6.1.4.1.1466.115.121.1.15, caseExact
SIPIdentityVoipUserCallerId	0.0.8.350.1.1.6.1.9	Specifies Voip User Callerid	1.3.6.1.4.1.1466.115.121.1.15, caseExact
SIPIdentityDefaultPhone	0.0.8.350.1.1.6.1.10	Specifies the user default phone	1.3.6.1.4.1.1466.115.121.1.15, caseExact
SIPIdentityAccountPassword	0.0.8.350.1.1.6.1.11	Specifies the user account password	1.3.6.1.4.1.1466.115.121.1.15, caseExact

SIPIdentityVoipUserLanguage	0.0.8.350.1.1.6.1.12	Specifies Voip User Language	1.3.6.1.4.1.1466.115.121.1.15, caseExact
IPBVoicemail	0.0.8.350.1.1.6.1.13	VoiceMail	1.3.6.1.4.1.1466.115.121.1.15, caseExact
SIPIdentityVoipUserInternalCallerId	0.0.8.350.1.1.6.1.14	Specifies Voip User Internal Callerid	1.3.6.1.4.1.1466.115.121.1.15, caseExact

Tabela 4.7

Atributo	**Sintaxe do AD**		
	Nome	**attributeSyntax**	**oMSyntax**
SIPIdentitySIPURI	String(Case Sensitive)	2.5.5.3	27
SIPIdentityRegistrarAddress	String(IA5)	2.5.5.5	22
SIPIdentityProxyAddress	String(IA5)	2.5.5.5	22
SIPIdentityAddress	String(IA5)	2.5.5.5	22
SIPIdentityPassword	String(Octet)	2.5.5.10	4
SIPIdentityUserName	String(Teletex)	2.5.5.4	20
SIPIdentityServiceLevel	String(IA5)	2.5.5.5	22
SIPIdentityVoipUserAlias	String(Case Sensitive)	2.5.5.3	27
SIPIdentityVoipUserCallerId	String(Case Sensitive)	2.5.5.3	27
SIPIdentityDefaultPhone	String(Case Sensitive)	2.5.5.3	27
SIPIdentityAccountPassword	String(Case Sensitive)	2.5.5.3	27
SIPIdentityVoipUserLanguage	String(Case Sensitive)	2.5.5.3	27
IPBVoicemail	String(Case Sensitive)	2.5.5.3	27
SIPIdentityVoipUserInternalCallerId	String(Case Sensitive)	2.5.5.3	27

Tabela 4.8

Classe	**ID**	**Descriçao**	**AD Syntax**		
			subClassOf	**mustContain**	**mayContain**
SIPIdentity	0.0.8.350.1.1.6.2.1	SIPIdentity object	top		(todos os atributos da tabela X)

Com os dados obtidos na conversão criaram-se quatro ficheiros LDIF: “atributos_system.ldif”, “atributos_sipidentity.ldif”, “atributos_qmail.ldif” e “classes.ldif”.

Para então finalizar a extensão do *schema* efetuaram-se seguintes operações:

```
sed -i 's/DC=DOMAIN,DC=COM/'$BASEDN'/' /usr/share/samba/schemas/atributos_system.ldif
sed -i 's/DC=DOMAIN,DC=COM/'$BASEDN'/'
/usr/share/samba/schemas/atributos_sipidentity.ldif
sed -i 's/DC=DOMAIN,DC=COM/'$BASEDN'/' /usr/share/samba/schemas/atributos_qmail.ldif
sed -i 's/DC=DOMAIN,DC=COM/'$BASEDN'/' /usr/share/samba/schemas/classes.ldif
ldbmodify -H /var/lib/samba/private/sam.ldb
/usr/share/samba/schemas/atributos_system.ldif --option="dsdb:schema update
allowed"=true
ldbmodify -H /var/lib/samba/private/sam.ldb
/usr/share/samba/schemas/atributos_sipidentity.ldif --option="dsdb:schema update
allowed"=true
ldbmodify -H /var/lib/samba/private/sam.ldb
/usr/share/samba/schemas/atributos_qmail.ldif --option="dsdb:schema update allowed"=true
ldbmodify -H /var/lib/samba/private/sam.ldb /usr/share/samba/schemas/classes.ldif --
option="dsdb:schema update allowed"=true
```

Foi ainda necessário incluir adequadamente os atributos estendidos nas classes do LDAP "User", "Computer" e "Group", de forma a que a IPBrick AD DC possa criar objetos dessas classes com esses novos atributos preenchidos, tendo em conta que são indispensáveis para o funcionamento de vários serviços do sistema. Esta criação de objetos na prática consiste nas operações de adicionar utilizadores, computadores e grupos.

Para a classe "User", para além de se ter associado os atributos estendidos respetivos aos serviços de Email, VoIP, Voicemail e VPN, adicionaram-se também os atributos correspondentes às "Opções Extras" que a interface da IPBrick disponibiliza aquando a criação de um utilizador. Em relação à classe "Computer", incluíram-se os atributos estendidos que a IPBrick utiliza para a gestão de computadores e os relativos aos serviços de VoIP e Voicemail. Quanto à classe "Group" associaram-se os atributos estendidos que a IPBrick utiliza para a gestão de grupos de computadores.

Posto isto, criaram-se mais dois ficheiros LDIF para proceder com a alteração dessas classes: "user.ldif" e "group.ldif". É de realçar que a extensão da classe "Computer" é feita através do LDIF de alteração da classe "User", visto primeira tratar-se de uma subclasse da segunda, ou seja, a classe "Computer" herda os atributos da classe "User", conforme está definido na sintaxe do *schema* do Samba 4.

E concluiu-se este procedimento com as seguintes operações:

```
sed -i 's/DC=DOMAIN,DC=COM/'$BASEDN'/' /usr/share/samba/schemas/user.ldif
sed -i 's/DC=DOMAIN,DC=COM/'$BASEDN'/' /usr/share/samba/schemas/group.ldif
ldbmodify -H /var/lib/samba/private/sam.ldb /usr/share/samba/schemas/user.ldif --
option="dsdb:schema update allowed"=true
ldbmodify -H /var/lib/samba/private/sam.ldb /usr/share/samba/schemas/group.ldif --
option="dsdb:schema update allowed"=true
```

Os atributos estendidos que as classes "User", "Computer" e "Group" adquiriram apresentam-se na tabela 4.9.

Tabela 4.9

Classes	Atributos adquiridos
User	**qmail** - qmailUID, qmailGID, mailMessageStore, mailAlternateAddress, mailHost, mailForwardingAddress, deliveryProgramPath, qmailDotMode, deliveryMode, mailReplyText, accountStatus, qmailAccountPurge, mailQuotaSize, mailQuotaCount, mailSizeMax **system** - IPBVpnPassword, IPBUserQuota, IPBUserPassType, IPBFollowMeMode **sipidentity** - SIPIdentitySIPURI, SIPIdentityDefaultPhone, SIPIdentityVoipUserAlias, SIPIdentityUserName, SIPIdentityPassword, SIPIdentityAccountPassword, IPBVoicemail, SIPIdentityVoipUserCallerId, SIPIdentityVoipUserInternalCallerId, SIPIdentityVoipUserLanguage **Opções Extras** – employeeNumber, departmentNumber, roomNumber, pager, employeeType, businessCategory, jpegPhoto
Computer	**system** – IPBMachineType, IPBMachinePassword, IPBIpAddress, IPBMacAddress **sipidentity** – SIPIdentityVoipUserCallerId, SIPIdentityVoipUserInternalCallerId, SIPIdentityVoipUserLanguage, SIPIdentityVoipUserAlias, IPBVoicemail
Group	**system** – IPBComputerGroupType, IPBComputerGroupNetStart, IPBComputerGroupNetStop

4.5.3 Módulo de Gestão de Utilizadores da IPBrick AD DC

Quando se cria, modifica ou elimina um utilizador através da interface *web* da IPBrick, essa informação é guardada na sua base de dados PostgresSQL. Depois, quando se aplica configurações do sistema é chamada a biblioteca "actualiza_def.php" que, recorrendo aos dados armazenados na base de dados, executa as funções "systemCreateUser()", "systemModifyUser()" ou "systemDeleteUser()", respetivamente. Este procedimento da IPBrick é baseado na mesma arquitetura demonstrada na Figura 4.2. Todos os outros módulos correspondentes aos serviços da IPBrick, em que alguns serão discutidos nas próximas secções, também seguem esta lógica de funcionamento.

Desta forma, para integrar o Samba 4 com este módulo da IPBrick, alteraram-se as funções do mesmo referidas no parágrafo anterior, de maneira a que quando são invocadas,

caso verifiquem que o modo de autenticação é o IPBrick AD DC, executam um conjunto de operações utilizando o Samba 4.

A função “systemCreateUser()” passa a realizar o seguinte procedimento:

1. Obtém os parâmetros de configuração do utilizador que se encontram na base de dados da IPBrick:

```
$utilizador = getSystemUserInfoByUidnumber ($uidnumber);
```

2. Insere o utilizador no LDAP através da ferramenta de administração do Samba 4, o “samba-tool”. O comando do “samba-tool” para adicionar utilizadores (“samba-tool user add”) ainda só permite a passagem como argumentos de alguns atributos de configuração dos utilizadores, tais como o “login”, “password”, “givenName”, “profilePath”, “homeDirectory”, “mailAddress”, “homeDrive”, etc. Outros atributos como o “uidNumber”, “gidNumber” e “unixHomeDirectory” tiveram que ser inseridos manualmente no LDAP através da função de PHP “ldap_add()”.

```
$info_samba4["uidnumber"] = $utilizador[0]->uidnumber;
$info_samba4["gidnumber"] = $utilizador[0]->gidnumber;
$info_samba4["unixHomeDirectory"] = "/home/".$utilizador[0]->login;
$homeDirectory  = "\\\\\\".$utilizador[0]->ipserver."\\".$utilizador[0]->login;
$homedrive      =  $utilizador[0]->homedrive;
if ($utilizador[0]->roamingprofile == 1)
{
        $profilepath = "\\\\\\".$utilizador[0]-
>ipserver."\\profiles\\".$utilizador[0]->login;
        $command ="/usr/bin/samba-tool user add  ".$utilizador[0]->login."
".$utilizador[0]->password." --given-name=\"".$utilizador[0]->name."\" --profile-
path=\"".$profilepath."\" --home-directory=\"".$homeDirectory."\" --mail-
address=\"".$utilizador[0]->mail."\"  --home-drive=\"".$homedrive."\" > /dev/null
2> /dev/null";
}
else
{
        $profilepath = "";
        $command ="/usr/bin/samba-tool user add  ".$utilizador[0]->login."
".$utilizador[0]->password." --given-name=\"".$utilizador[0]->name."\"  --home-
directory=\"".$homeDirectory."\" --mail-address=\"".$utilizador[0]->mail."\"  --
home-drive=\"".$homedrive."\" > /dev/null 2> /dev/null";
}
$result = array();
execAsRoot ($command, $result);
$dn_samba4 = "cn=".$utilizador[0]->name.",cn=Users,".$dn_base;
$dn_samba4 = str_replace ("cn", "CN", $dn_samba4);
$dn_samba4 = str_replace ("dc", "DC", $dn_samba4);
$dbldap->My_Connect_Ldap ();
$dbldap->My_V3_Ldap ();
$dbldap->My_Bind_Ldap ();
$dbldap->My_Modify_Ldap ($dn_samba4, $info_samba4);
$dbldap->My_Close_Ldap ();
```

3. Os restantes atributos que a IPBrick utiliza para a gestão de utilizadores, apresentados na Tabela 4.9, também são preenchidos manualmente através da função PHP “ldap_add()”.

As funções “systemModifyUser()” e “systemDeleteUser()” foram modificadas seguindo a mesma abordagem.

4.5.4 Módulo de Gestão de Grupos de Utilizadores da IPBrick AD DC

A IPBrick utiliza os grupos de utilizadores para permitir que os membros de um determinado grupo tenham em comum as mesmas permissões para certas diretórias ou ficheiros. Por defeito são criados dois grupos: “Administradores” e “Membros do Domínio”.

Analogamente ao módulo anterior, para integrar o Samba 4 com este módulo modificaram-se as funções “systemCreateGroup” e “systemAddUsersToGroup()”, que são invocadas quando se cria um grupo de utilizadores na IPBrick ou quando se adiciona utilizadores a um grupo, respetivamente.

Função “systemCreateGroup()”:

1. Consulta a base de dados IPBrick para obter os parâmetros de configuração do grupo a ser criado:

```
$group = getSystemGroupInfoByGidnumber ($gidnumber);
```

2. Cria o grupo através do comando “samba-tool group add”:

```
$result = array();
$command ="/usr/bin/samba-tool group add ".$group[0]->name." > /dev/null 2>
/dev/null";
execAsRoot ($command, $result);
```

3. Os atributos de configuração, tais como o “gidNumber” e “description” são adicionados manualmente recorrendo à função PHP “ldap_add()”.

Função “systemAddUsersToGroup()”:

1. Consulta a base de dados IPBrick para obter os utilizadores que se adicionaram a um determinado grupo:

```
$group = getSystemGroupInfoByGidnumber ($gidnumber);
```

2. Adiciona os utilizadores ao grupo através do comando “samba-tool group addmembers”:

```
$result = array();
if (count($group[0]->member)>0)
{
        for ($i=0; $i<count($group[0]->member); $i++)
        {
                $command="";
                $command ="/usr/bin/samba-tool group addmembers ".$group[0]-
>name." '".$group[0]->member[$i]->name."' > /dev/null 2> /dev/null";
                execAsRoot ($command, $result);
        }
}
```

4.5.5 Módulo de Gestão de Computadores da IPBrick AD DC

Este módulo permite a inserção ou alteração de máquinas no LDAP da IPBrick. Estas máquinas são representadas pelo seu tipo, nome, grupo, endereço IP e endereço MAC. Ainda possuem outros atributos de configuração que dependem do tipo de máquina selecionado.

Para integrar este módulo com o Samba 4 modificou-se a função "add_maquina()" que é invocada quando é adicionada uma máquina através da interface *web* da IPBrick.

Desta vez não foi possível utilizar a ferramenta "samba-tool" pois ainda não suporta as funcionalidades de inserção, modificação ou remoção de computadores. Sendo assim, o processo de adicionar uma máquina teve que ser feito inserindo-a manualmente no LDAP, ou seja, cria-se um objeto da classe "Computer" com os devidos atributos corretamente preenchidos no que diz respeito à sintaxe do *schema* do AD.

Função "add_maquina()":

1. Consulta a base de dados IPBrick para obter os parâmetros de configuração do computador a ser criado:

```
$maquinas = $dbmaquinasldap->getMaquinasldapByAccao ('I');
$maquinainfo = getSystemComputerInfoByUidnumber ($maquinas[$i]->uidnumber);
```

2. Cria um objeto da classe "Computer" no LDAP para a máquina adicionada:

```
$info1["objectclass"][0] = "top";
$info1["objectclass"][1] = "person";
$info1["objectclass"][2] = "organizationalPerson";
$info1["objectclass"][3] = "user";
$info1["objectclass"][4] = "computer";
$dn_samba4 = "cn=".$nome.",cn=Computers,".$dn_base;
$dn_samba4 = str_replace ("cn", "CN", $dn_samba4);
$dn_samba4 = str_replace ("dc", "DC", $dn_samba4);
$this->My_Connect_Ldap ();
$this->My_V3_Ldap ();
$this->My_Bind_Ldap ();
$this->My_Add_Ldap ($dn_samba4, $info1);
$this->My_Close_Ldap ();
```

3. Adiciona o computador ao grupo que é criado por defeito pelo Samba 4, o "Domain Computers", e define-o como grupo primário:

```
$result = array();
$command = "/usr/bin/samba-tool group addmembers \"Domain Computers\" ".$nome." >
/dev/null 2> /dev/null";
execAsRoot ($command, $result);
$info2["primaryGroupID"] = "515";
```

4. Insere atributos necessários como o "sAMAccountName", "uidNumber" e "gidNumber", e todos os outros atributos que a IPBrick utiliza para a gestão de computadores, apresentados na Tabela 4.9.

4.5.6 Módulo de Gestão de Grupos de Computadores da IPBrick AD DC

Este módulo da IPBrick possibilita a criação de grupos de máquinas com *subnets* definidas estabelecendo uma gama de IPs para esse grupo. É útil, por exemplo, para configurar acessos de *web proxy*.

Para integrá-lo com o Samba 4 alteram-se as funções "add_grupomaquina()" e "set_maquinas_ao_grupomaquina()" que são invocadas quando se cria um grupo de computadores na IPBrick ou quando se adiciona computadores a um grupo, respetivamente.

Função "add_grupomaquina()":

1. Consulta a base de dados IPBrick para obter os parâmetros de configuração do grupo de máquinas a ser criado:

```
$grupomaquinas = $dbgrupomaquinas->getGrupomaquinasByAccao ('I');
```

2. Cria o grupo através do comando "samba-tool group add":

```
$result = array();
$command = "/usr/bin/samba-tool group add ".$nome." --groupou='cn=Computers' >
/dev/null 2> /dev/null";
execAsRoot ($command, $result);
```

3. O "gidNumber" e os restantes atributos que a IPBrick utiliza para a gestão de grupos de computadores, presentes na Tabela 4.9, são preenchidos manualmente através da função PHP "ldap_add()".

Função "set_maquinas_ao_grupomaquina()":

1. Consulta a base de dados IPBrick para obter as máquinas que se adicionaram a um determinado grupo:

```
$maquinas_grupomaquinas = $dbligamaquinas_ldapgrupomaquinas-
>getLigaMaquinas_ldapGrupomaquinasByNotAccao ('N');
for ($i=0; $i<count($maquinas_grupomaquinas); $i++)
{
        $member = $dbligamaquinas_ldapgrupomaquinas-
>getLigaMaquinas_ldapGrupomaquinasByIdgrupomaquinas ($maquinas_grupomaquinas[$i]-
>idgrupomaquinas);
        unset ($memberuid);
        for ($j=0; $j<count($member); $j++)
        {
                $memberuid[$j] = $member[$j]->nome;
        }
        $grupomaquinas = $dbgrupomaquinas->getGrupomaquinasByIdgrupomaquinas
($maquinas_grupomaquinas[$i]->idgrupomaquinas);
        $dbligaldap->set_maquinas_ao_grupomaquina_ByGidnumber ($memberuid,
$grupomaquinas[0]->gidnumber);
```

2. Adiciona as máquinas ao grupo através do comando "samba-tool group addmembers":

```
for ($i=0; $i<count($logins); $i++)
{
        $command="";
        $command ="/usr/bin/samba-tool group addmembers ".$nome."
".$logins[$i]."$ > /dev/null 2> /dev/null";
        execAsRoot ($command, $result);
```

4.5.7 Módulo de Servidor de Ficheiros da IPBrick AD DC

O servidor de ficheiros da IPBrick providencia duas áreas de trabalho por defeito, "Work Area 1" e "Work Area 2", que correspondem a partições físicas no disco localizadas em "/home1" e "/home2". Quando um utilizador é criado, o sistema cria uma estrutura de pastas que representa a sua conta pessoal, e que contém os ficheiros de email e o perfil de utilizador.

Nestas áreas de trabalho podem ser criadas partilhas de ficheiros entre os utilizadores, que implica a configuração das permissões de acesso a esses ficheiros. As permissões possíveis são "None", "Read" e "Read+Write", e podem ser atribuídas tanto a utilizadores como a grupos de utilizadores.

Para criar uma partilha no Samba 4 basta adicionar o seguinte bloco de código ao seu ficheiro de configuração "/etc/samba/smb.conf":

```
[Nome_da_Partilha]
        path = /pasta_a_partilhar/
        read only = no
```

Para definir as permissões dessa partilha estabelecida pelo Samba 4, adicionam-se as seguintes linhas ao bloco de código anterior:

- Administradores da partilha:
  admin users = utilizador1, utilizador2, … , @grupo1, @grupo2, …
- Permissões de leitura:
  valid users = utilizador1, utilizador2, … , @grupo1, @grupo2, …
- Permissões de leitura e escrita:
  write list = utilizador1, utilizador2, …, @grupo1, @grupo2, …

Posto isto, para integrar o Samba 4 com este módulo da IPBrick alterou-se a função "export_sharesconf()", que é invocada quando se cria, modifica ou elimina partilhas ou se altera as permissões das mesmas.

Esta função, de modo a adicionar o bloco de código referido mesmo há pouco ao "smb.conf" e com as devidas configurações, passa a efetuar as seguintes operações:

1. Consulta a base de dados IPBrick de modo a obter as definições das partilhas inseridas, alteradas ou removidas e as respetivas permissões:

```
$Allpartilhas = $dbpartilhas->getPartilhas();
for ($i=0; $i<count($Allpartilhas); $i++)
{
        $grupos_a = $dbacessopartilhasgrupo-
>getAcessoPartilhasGrupoByIdpartilhaTipoacesso ($Allpartilhas[$i]->idpartilha,
3);
        $users_a = $dbacessopartilhasutilizador-
>getAcessoPartilhasUtilizadorByIdpartilhaTipoacesso ($Allpartilhas[$i]-
>idpartilha, 3);
        $grupos_w = $dbacessopartilhasgrupo-
>getAcessoPartilhasGrupoByIdpartilhaTipoacesso ($Allpartilhas[$i]->idpartilha,
2);
        $users_w = $dbacessopartilhasutilizador-
>getAcessoPartilhasUtilizadorByIdpartilhaTipoacesso ($Allpartilhas[$i]-
>idpartilha, 2);
        $grupos_r = $dbacessopartilhasgrupo-
>getAcessoPartilhasGrupoByIdpartilhaTipoacesso ($Allpartilhas[$i]->idpartilha,
1);
        $users_r = $dbacessopartilhasutilizador-
>getAcessoPartilhasUtilizadorByIdpartilhaTipoacesso ($Allpartilhas[$i]-
>idpartilha, 1);
```

2. Adiciona as partilhas ao “smb.conf” com as devidas permissões:

```
for ($i=0; $i<count($Allpartilhas); $i++)
{
        $data .= "[".$Allpartilhas[$i]->nome."]\n";
        $data .= "        comment = ".$Allpartilhas[$i]->descricao."\n";
        $data .= "        path = ".$Allpartilhas[$i]->path."\n";
        if ($Allpartilhas[$i]->browseable=='t') {
                $data .= "        browseable = yes\n";
        }
        else
        {
                $data .= "        browseable = no\n";
        }
        $data .= "  read only = no\n";
        $data .= "  admin users = ".$admin_list."\n";
        $data .= "  valid users = ".$valid_list."\n";
        $data .= "  write list = ".$write_list."\n";
}
$socket->IpSocket_Write_Data ("GERA_CONFIG_FILE");
if ($socket->IpSocket_Read_Data ($temp) < 0)
{
        die ("<br><center>Error Socket Read!</center><br>");
}
$socket->IpSocket_Write_Data ("smb.conf");
$socket->IpSocket_Write_Data ($data);
$socket->IpSocket_Write_Data ("END");
if ($socket->IpSocket_Read_Data ($temp) < 0)
{
        die ("<br><center>Error Socket Read!</center><br>");
```

4.5.7.1 *Roaming profiles*

A IPBrick permite a opção de ativar a funcionalidade de *roaming profiles* para os utilizadores, que se trata de um tipo de partilha especial. Estes perfis, dizem respeito aos postos de trabalho Windows que pertencem ao domínio da IPBrick, e consistem num conjunto de pastas com as definições de cada utilizador que são armazenadas em “C:\Documents and Settings\user_login”. Com os *roaming profiles* configurados, quando um utilizador efetua *logout* numa estação de trabalho Windows, todos as pastas do seu perfil são sincronizadas para a sua conta pessoal da IPBrick, localizada em " \\ipbrick\user_login\.profiles”. Quando esse utilizador volta a fazer *login* no mesmo posto de trabalho ou noutro diferente, o seu perfil é sincronizado de volta para esse posto de trabalho onde o utilizador se encontra.

Para integrar o Samba 4 com esta funcionalidade, adicionou-se a partilha especial "Profiles" ao "smb.conf" com as permissões devidamente definidas para o correto funcionamento entre o Windows e o Samba 4.

```
[Profiles]
        comment = Network Profiles Share
        path = /home/%U/conta/.profiles
        read only = No
        store dos attributes = Yes
        create mask = 0600
        directory mask = 0700
        guest ok = no
        printable = no
        profile acls = yes
        csc policy = disable
```

Um *roaming profile* só fica realmente ativado para um utilizador quando se define o atributo LDAP "profilePath" desse utilizador que corresponde à localização da partilha "Profiles". Isso já é garantido quando a se cria um utilizador na IPBrick AD DC.

```
(…)
if ($utilizador[0]->roamingprofile == 1)
{
        $profilepath = "\\\\\\".$utilizador[0]->ipserver."\\profiles\\".$utilizador[0]-
>login;
(…)
```

4.5.8 Módulo de Servidor de Impressão da IPBrick AD DC

A interface *web* da IPBrick permite a gestão de impressoras que se pretendam tornar disponíveis na rede do servidor. Os campos de configuração das mesmas são o nome, descrição, localização física e o tipo de ligação entre o servidor e a impressora. O tipo de ligação pode ser porta Paralela, porta Serial, porta USB ou impressora de Rede (ligada a um LAN *switch*). No caso de ser uma impressora de rede é necessário o seu endereço IP e porta para configurá-la.

Um servidor de impressão aceita ordens de impressão de computadores em rede, gere essas ordens localmente e envia-as para as devidas impressoras. O Samba 4 tem a capacidade de atuar como um servidor de impressão compatível com o Windows. Enquanto o Samba 4 proporciona a interface para máquinas Windows/SMB, a ferramenta CUPS é utilizada pelo Samba 4 para enviar ordens de impressão para os dispositivos.

O CUPS é um sistema de impressão *open source* para sistemas operativos Unix, sendo que o Samba 4, por defeito, vem incorporado com suporte total para o mesmo. O CUPS utiliza o protocolo IPP (*Internet Printing Protocol*) para se ligar às impressoras.

Posto isto, para integrar o Samba 4 com este módulo da IPBrick começou-se por configurar o "smb.conf" de modo a tornar o Samba 4 num servidor de impressão. Para isso,

criou-se uma partilha própria do Samba 4 que define as configurações gerais do serviço de impressão. O suporte para o CUPS é ativado através da linha "printing = CUPS".

```
[Printers]
        comment = All Printers
        path = /tmp
        browseable = no
        public = yes
        writable = no
        printable = yes
        write list = @\"Domain Admins\"
        printing = CUPS
```

Depois, modificou-se a função "export_smbprintconf()" que é invocada quando se adiciona uma impressora através da interface *web* da IPBrick, de maneira a adicionar uma partilha no "smb.conf" respetiva a essa impressora, necessária para a mesma ser disponibilizada pelo Samba 4.

Função "export_smbprintconf()":

1. Consulta a base de dados IPBrick de modo a obter as impressoras inseridas e respetivas definições:

```
$printers = $dbimpressoras->getImpressoras ();
```

2. Adiciona as partilhas das impressoras no "smb.conf":

```
for ($i=0; $i<count($printers); $i++)
{
        $texto .= "
        [".$printers[$i]->nome."]
                comment = ".$printers[$i]->descricao."
        Printer Name = ".$printers[$i]->nome."
                path = /tmp
                browseable = Yes
                printable = Yes
                printer admin = @\"Domain Admins\"
                write list = @\"Domain Admins\"";
}
$socket->IpSocket_Write_Data ("GERA_CONFIG_FILE");
if ($socket->IpSocket_Read_Data ($temp) < 0)
{
        die ("<br><center>Error Socket Read!</center><br>");
}
$socket->IpSocket_Write_Data ("smb.conf");
$socket->IpSocket_Write_Data ($texto);
$socket->IpSocket_Write_Data ("END");
if ($socket->IpSocket_Read_Data ($temp) < 0)
{
        die ("<br><center>Error Socket Read!</center><br>");
```

Também foi necessário criar um processo de inserção automática no CUPS das impressoras adicionadas pela interface *web* da IPBrick. Para associar uma impressora ao CUPS é necessário fornecer o seu endereço IP, porta e o ficheiro PPD correspondente. Um ficheiro PPD (*PostScript Printer Description*) descreve as capacidades da impressora e é utilizado pelo CUPS para suportar funcionalidades específicas da impressora e filtragem inteligente.

Assim, para adicionar as impressoras ao CUPS automaticamente recorreu-se à ferramenta de configuração do mesmo, a "lpadmin", passando-lhe os devidos argumentos:

```
$command = "/usr/sbin/lpadmin -p ".$nome." -v socket://".$endereco.":".$porto." -E -P
/etc/cups/ppd/".$_FILES["ppd"]["name"]." > /dev/null 2> /dev/null";
$result = array();
execAsRoot($command, $result);
```

Por último, ainda se implementou um procedimento para os *drivers* de qualquer impressora adicionada serem instalados automaticamente nos postos de trabalho Windows, sem que os utilizadores tenham que fazer qualquer configuração. Para isso, recorreu-se ao serviço Point and Print do Windows que efetua o *upload* e pré configura *drivers* no servidor de impressão do Samba 4.

De modo a garantir um processo totalmente automatizado, teve-se que incluir *drivers* universais no servidor IPBrick que funcionem para qualquer impressora. Para tal, obteve-se os *drivers* Postscript da Microsoft para Windows que se encontram na pasta "%WINDIR%\SYSTEM32\SPOOL\DRIVERS\" e os *drivers* Postscript do CUPS para Windows que estão disponíveis no *website* do CUPS. Os ficheiros desses *drivers* foram então copiados para a pasta "/usr/share/cups/drivers/" da IPBrick e são os seguintes:

- *Drivers* Postscript da Microsoft para Windows
  - ps5ui.dll
  - pscript.hlp
  - pscript.ntf
  - pscript5.dll
- *Drivers* Postscript do CUPS para Windows
  - cups6.inf
  - cups6.ini
  - cupsps6.dll
  - cupsui6.dll

Posto isto, o procedimento é o seguinte:

1. Adiciona-se a partilha especial "print$" ao "smb.conf" para ativar o suporte do serviço Point and Print. A linha "path = /home1/_smbdrivers" define a localização dos *drivers*.

```
[print\$]
        comment = Printer Drivers
        path = /home1/_smbdrivers
        browseable = yes
        read only = yes
        write list = @\"Domain Admins\"
```

2. Recorre-se à ferramenta "cupsaddsmb" do CUPS, para se providenciar os *drivers* das impressoras e respetivos ficheiros PPD para clientes de máquina Windows. Os drivers são exportados para a pasta "/home1/_smbdrivers/.

```
$command = "/usr/sbin/cupsaddsmb -H localhost -U administrator%".$admin_pass[0]-
>pass." -v ".$nome." > /dev/null 2> /dev/null";
$result = array();
execAsRoot($command, $result);
```

3. Utiliza-se a ferramenta “rpcclient” para associar os *drivers* às impressoras no servidor de impressão do Samba 4.

```
$command = "/usr/bin/rpcclient localhost -U administrator%".$admin_pass[0]-
>pass." -c 'setdriver ".$nome." " .$nome."' > /dev/null 2> /dev/null";
$result = array();
execAsRoot($command, $result);
```

Capítulo 5

Conclusão

Este é o último capítulo deste documento onde serão apresentadas as conclusões sobre o trabalho desenvolvido. Também serão apresentadas propostas de futuros desenvolvimentos relacionados com este tema.

5.1 Conclusão da dissertação

O trabalho desenvolvido resultou numa IPBrick OS com o software *open source* Samba 4 totalmente integrado dotando-a com uma funcionalidade inovadora que só o Microsoft Windows Server consegue oferecer: um serviço de controlador de domínio com Active Directory.

Um controlador de domínio com Active Directory proporciona uma gestão centralizada dos dispositivos de uma rede, com controlo absoluto sobre grandes quantidades de objetos, por exemplo, utilizadores e máquinas. Isto é crucial para alcançar menores custos em tarefas administrativas, controlo de recursos, e gestão de segurança (autenticação e autorização). É a chave para organizar utilizadores e recursos de uma forma simples de gerir e é escalável (delegação de permissões).

Sendo um tema bastante complexo que engloba uma quantidade enorme de conhecimentos enquadrados na área de redes e na área de sistemas Linux e Windows, exigiu estudos e investigações intensivas de forma a conseguir-se ter uma visão concreta de como avançar para uma solução viável e possível de ser concretizada dentro dos limites de tempo de uma dissertação de mestrado.

Além do mais, a integração do Samba 4 na IPBrick OS coloca-o numa posição bastante crítica perante o sistema, visto que passa a ser responsável pelos serviços principais e interage praticamente com todos os outros, o que aumenta seriamente os riscos de toda esta operação.

Portanto foi necessário começar por estudar e analisar a estrutura e funcionamento do sistema operativo IPBrick OS. Depois procedeu-se com a instalação do Samba 4 no sistema tendo em conta os seus requisitos, seguida pela sua configuração de modo a poder provisioná-lo como um Active Directory Domain Controller. Por último, realizou-se a integração final entre o Samba 4 e todos os serviços da IPBrick OS, garantindo um funcionamento completo de todo o sistema operativo sistema para além das novas funcionalidades. Foi ainda necessário garantir procedimentos de migração de servidores Windows AD DC e IPBrick Os com Samba 3 para a nova IPBrick OS AD DC com Samba 4, e também um método de backup para os controladores de domínio, de forma a esta solução poder satisfazer todo o tipo de clientes. Todos os desenvolvimentos foram testados e validados.

Para concluir, pode-se afirmar que os objetivos deste projeto foram atingindos com sucesso. A IPBrick OS apresenta-se agora como uma alternativa gratuita aos servidores Windows da Microsoft, disponibilizando um serviço de Active Directory *open source* com todas as suas funcionalidades características que garante uma total interoperabilidade entre ambientes Linux e Windows.

5.2 Desenvolvimentos futuros

Sendo que esta integração engloba inúmeros serviços e funcionalidades, e que o Samba 4 continua em constante desenvolvimento, este tema tem muito por melhorar e/ou desenvolver. Destacam-se os seguintes pontos:

- Atualizar a versão do Debian do sistema operativo IPBrick OS de modo a ser possível instalar as versões mais recentes do Samba 4;
- Dotar esta solução com Single sign-on (SSO);
- Melhorar a interface de gestão da IPBrick OS de forma a ser possível administrar o Samba 4 AD DC de forma mais avançada tal como a ferramenta RSAT permite.

Referências

[1] Desmond, B., J. Richards, R. Allen e A.G. Lowe-Norris. 2013. *Active Directory: Designing, Deploying, and Running Active Directory*. O'Reilly Media.

[2] Wahl, M., S Kille e T. Howes. 1997. "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names". http://www.ietf.org/rfc/rfc2253.

[3] TechNet, Microsoft. 2016. "Funções de mestre de operações". https://technet.microsoft.com/pt-pt/library/cc773108(v=ws.10).aspx.

[4] Samba.org. "Download". https://www.samba.org/samba/download/.

[5] Carter, G., J. Ts e R. Eckstein. 2007. *Using Samba: A File & Print Server for Linux, Unix & Mac OS X*. O'Reilly Media.

[6] Tridgell, Andrew. 2003. "How Samba was written". https://www.samba.org/ftp/tridge/misc/french_cafe.txt.

[7] Samba.org. 2012. "Samba Team Releases Samba 4.0". https://www.samba.org/samba/news/releases/4.0.0.html.

[8] Williams, Adam. 2013. "What's New in Samba 4". https://www.linux-magazine.com/Online/Features/What-s-New-in-Samba-4.

[9] Foundation, OpenLDAP. "community developed LDAP software". http://www.openldap.org/.

[10] Samba.org. 2012. "Release Notes for Samba 4.0.0". https://www.samba.org/samba/history/samba-4.0.0.html.

[11] SambaWiki. 2016. "Samba4/LDAP Backend". https://wiki.samba.org/index.php/Samba4/LDAP_Backend.

[12] TechNet, Microsoft. 2016. "How the Active Directory Replication Model Works". https://technet.microsoft.com/en-us/library/cc772726%28v=ws.10%29.aspx.

[13] SambaWiki. 2017. "Setting up Samba as an NT4 PDC (Quick Start)". https://wiki.samba.org/index.php/Setting_up_Samba_as_an_NT4_PDC_(Quick_Start).

[14] ———. 2016. "User Documentation". https://wiki.samba.org/.