

IT-221

IT-221



# UNIVERSIDADE EDUARDO MONDLANE

**Faculdade de Ciências**

Departamento de Informática

Trabalho de Licenciatura

Distribuição de Dados do NUIT na Direcção Geral de Impostos

Autora: Maria Amélia Daniel Tcheco

Maputo, Setembro de 2005

IT-221

R.E. 10.400



# UNIVERSIDADE EDUARDO MONDLANE

**Faculdade de Ciências**

Departamento de Informática

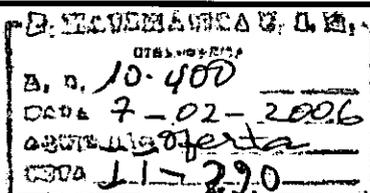
Trabalho de Licenciatura

Distribuição de Dados do NUIT na Direcção Geral de Impostos

**Supervisor: dr. Rodrigues Zicai Fazenda**

**Autora: Maria Amélia Daniel Tcheco**

**Maputo, Setembro de 2005**



## DEDICATÓRIA

Dedico este trabalho a minha família especialmente a minha querida mãe pelo apoio, amor e carinho que me proporcionou, ajudando a cuidar do meu filho quando precisava de estudar e ao meu filho que muitas vezes não pude lhe dar a devida atenção.

*M<sup>a</sup> Amélia D. Tcheco*

Maria Amélia Daniel Tcheco



## **AGRADECIMENTOS**

Quero expressar a minha profunda gratidão ao meu supervisor dr. Rodrigues Zicai Fazenda que me apoiou bastante para que este trabalho se tornasse realidade, a todos os docentes do DMI que directa ou indirectamente contribuíram para a minha formação académica, a toda minha família e amigos pelo apoio que me deram durante os meus estudos.

Um especial agradecimento aos meus colegas de serviço por toda ajuda por eles prestada ao longo da realização deste trabalho.

## DECLARAÇÃO DE HONRA

Declaro por minha honra que o presente trabalho é resultado das minhas pesquisas e que o mesmo foi realizado apenas para ser submetido como trabalho de Licenciatura em Informática na Faculdade de Ciências da Universidade Eduardo Mondlane.

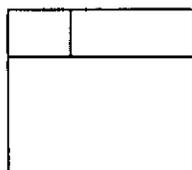
Maputo, Setembro de 2005

M<sup>ª</sup> Amélia Daniel Tcheco

(Maria Amélia Daniel Tcheco)

## SIMBOLOGIA USADA

Representa processos



Representa entidades



Representa o fluxo de informação



Representa arquivo



## **LISTA DE ABREVIATURAS**

<b>BD</b>	Base de Dados
<b>C/S</b>	Cliente/Servidor
<b>DBA</b>	Data Base Administration
<b>DDL</b>	Data Definition Language
<b>DFD</b>	Diagrama de Fluxo de Dados
<b>DGI</b>	Direcção Geral de Impostos
<b>DML</b>	Data Manipulation Language
<b>IP</b>	Internet Protocol
<b>ISO</b>	International Organization of Standardization
<b>LAN</b>	Local Area Network
<b>MAN</b>	Metropolitan Area Network
<b>NUIT</b>	Número Único de Identificação Tributária
<b>OSI</b>	Open System Interconnection
<b>PDC</b>	Primary Domain Controller
<b>RDBMS</b>	Relational Data Base Management System
<b>RF</b>	Repartição de Finanças
<b>SGBD</b>	Sistema de Gestão de Base de Dados
<b>SO</b>	Sistema Operativo
<b>SQL</b>	Structured Query Language
<b>TCP</b>	Transmission Control Protocol
<b>TDM</b>	Telecomunicações de Moçambique
<b>WAN</b>	Wide Area Network

## **Resumo**

O Processo de atribuição do Número Único de Identificação Tributária, abreviadamente designada por NUIT, aos contribuintes levado a cabo pela Direcção Geral de Impostos que é o órgão central do Ministério das Finanças, passa necessariamente pela recolha de dados dos mesmos. Esta recolha de dados é efectuada pelas repartições de finanças que se encontram dispersas por quase todo o país, lançando-os num sistema informático criado para o efeito.

Todos os dados são guardados na Direcção Geral de Impostos, central, onde se encontra a base de dados, e todo o tipo de validações é feita neste local.

Para tornar mais flexível e eficiente este processo de atribuição de NUIT, surgiu o presente trabalho cuja ideia é apresentar uma proposta para implementação de um processo de distribuição de dados para a atribuição de NUIT, baseada na arquitectura cliente/servidor usando o processamento distribuído, onde cada repartição de finanças considerada ter uma base de dados podendo, fácil e rapidamente acedê-la para qualquer tipo de consulta. Assim a Direcção Geral de Impostos, poderá controlar melhor os dados de cada repartição, através de consultas já que as duas bases de dados comunicam.

A existência de bases de dados locais, reduzirá o tempo de resposta, por exemplo nas operações de consulta, o tráfego na rede será menor pois cada repartição poderá trabalhar sem necessidade de uma ligação permanente com a central

## ÍNDICE

DEDICATÓRIA.....	I
AGRADECIMENTOS.....	II
DECLARAÇÃO DE HONRA.....	III
SIMBOLOGIA USADA .....	IV
LISTA DE ABREVIATURAS .....	V
RESUMO.....	VI
<b>CAPÍTULO I : INTRODUÇÃO E OBJECTIVOS.....</b>	<b>2</b>
1.1 INTRODUÇÃO.....	2
1.2 DESCRIÇÃO DO PROBLEMA.....	3
1.3 OBJECTIVO .....	3
<b>CAPÍTULO II: MATERIAL E METODOLOGIA .....</b>	<b>4</b>
2.1 MATERIAL.....	4
2.2 METODOLOGIA.....	4
<b>CAPÍTULO III: REVISÃO BIBLIOGRÁFICA.....</b>	<b>5</b>
3.1. ARQUITECTURA CLIENTE /SERVIDOR (C/S).....	5
3.1.1 Arquitectura Cliente/Servidor de Base de Dados.....	8
3.1.2 Alguns Tipos de Processos numa Arquitectura C/S.....	12
3.2 SISTEMAS DISTRIBUÍDOS .....	14
3.2.1 Sistema de Gestão de BD (SGBD).....	18
3.2.2 Replicação de dados.....	20
3.2.3 Segurança de Base de Dados.....	21
3.3 REDES DE COMPUTADORES .....	23
3.3.1. Componentes Básicas de uma Rede.....	24
3.3.2 Modelo de Referência OSI.....	26
3.3.3 Protocolos.....	28
<b>CAPÍTULO IV: ANÁLISE E DISCUSSÃO SOBRE AS REDES DGI E RF'S.....</b>	<b>32</b>
4.1. SITUAÇÃO ACTUAL.....	32
4.1.1 Descrição dos Componentes da Rede.....	34
4.2 PROPOSTA PARA A NOVA REDE.....	35
4.2.1 Descrição dos Componentes da Rede.....	36
4.3 PROPOSTA DO MODELO DA ARQUITECTURA DO PROCESSO DE DISTRIBUIÇÃO DADOS.....	37
4.4 MODELO CONCEPTUAL.....	38
4.4.1 <i>Front-End</i> da RF.....	40
4.4.2 Recepção de Dados na CENTRAL.....	41

4.4.3 Atribuição de NUIT/Alteração de Cadastros na CENTRAL.....	42
4.5 DESCRIÇÃO DAS ACTIVIDADES DO PROCESSO DE REPLICAÇÃO.....	44
4.5.1 Buscar Requisições (Processo 4).....	44
4.5.2 Retomar Rejeitados (Processo 8).....	45
4.5.3 Distribuir Cadastros (Processo 10).....	46
4.5.4 Temporização/Fluxo.....	47
4.6 MECANISMOS DE SEGURANÇA.....	49
4.6.1 Profile PASSWORDS_NUIT.....	49
4.6.2 Regras de verificação das passwords.....	49
4.6.3 Bloqueio de Utilizadores.....	50
4.7 BACKUP E RECUPERAÇÃO DA BD.....	50
4.7.1 Procedimentos de backup (Cold backup archiving).....	51
4.7.2 Procedimentos de Recuperação de Dados.....	52
<b>CAPÍTULO V: CONCLUSÕES E RECOMENDAÇÕES.....</b>	<b>54</b>
5.1. CONCLUSÕES.....	54
5.2. RECOMENDAÇÕES.....	54
<b>CAPÍTULO VI: BIBLIOGRAFIA.....</b>	<b>56</b>
6.1. REFERÊNCIA BIBLIOGRÁFICA.....	56
6.2. BIBLIOGRAFIA CONSULTADA.....	58

## **CAPÍTULO I : INTRODUÇÃO E OBJECTIVOS**

### **1.1 INTRODUÇÃO**

O imposto é uma das obrigações que os cidadãos devem (a todos os níveis) cumprir a nível autárquico, empresarial, do comércio, dos serviços e noutras formas de rendimento. Resulta de uma percentagem cobrada pelo estado, como forma de tributação derivado ao exercício ou ganho de valores<sup>1</sup> por parte de entidades (individuais ou colectivas).

Em Moçambique desde o ano de 2002 está a decorrer o processo de reforma do imposto, passando também pela sua informatização, para facilitar o processamento automático.

O processo de reforma do sistema de cobrança de impostos por parte do Governo Moçambicano requer a inscrição de todos os contribuintes, pois ao abrigo da Lei número 15/2002 de 26 de Junho, é obrigatório o uso do Número Único de Identificação Tributária, abreviadamente designado NUIT em todos os tributos, incluindo os aduaneiros. O NUIT é um número sequencial composto por 9 dígitos.

A Direcção Geral de Impostos (DGI) é o órgão central do Ministério das Finanças responsável pela cobrança dos Impostos. Possui 27 Repartições de Finanças (RF) em todo o País das quais 25 são Repartições de Finanças localizadas nas respectivas áreas fiscais, e 2 Repartições Especiais que se encarregam pela cobrança dos impostos às grandes empresas.

Para a atribuição do NUIT, é necessário que se faça a recolha de dados do contribuinte, através de modelos apropriados e aprovados para o efeito, processo cuja realização está a cargo repartições de finanças.

O actual sistema de atribuição de NUIT usa o modelo de arquitectura Cliente/Servido (C/S) com configuração à dois níveis, em que a aplicação está integrada com os dados a nível do servidor que se encontra na DGI (Central). Esta solução não é viável para a DGI, pois tendo esta o objectivo urgente de obter o maior número possível de contribuintes com NUIT, o número de máquinas clientes aumentará, o tráfego na rede será maior e o servidor de BD não poderá dar

---

<sup>1</sup> Os valores podem ser monetários, imobiliários, mobiliários, compensação, etc.

resposta às solicitações que lhe forem feitas no tempo requerido pela degradação do desempenho o que não é desejável.

A ideia base desta solução é de que cada Repartição de Finanças tenha a sua BD local, podendo trabalhar de forma autónoma sem necessidade de ter uma ligação permanente com a central.

## **1.2 DESCRIÇÃO DO PROBLEMA**

Todas as validações são efectuadas na DGI-Central onde se encontra a BD que contém os dados de todas as repartições. Este facto provoca morosidade na introdução dos dados dos contribuintes através dos respectivos modelos, para a atribuição do NUIT, os quais possuem campos que precisam de validação já que quando há vários utilizadores a fazer simultaneamente esta operação, requerendo uma validação do mesmo campo, esta não é feita em simultâneo (obdecendo o algoritmo FIFO), isto é, só faz uma de cada vez e na sequência das solicitações, o que consequentemente leva tempo. O mesmo acontece em relação a operações de consulta que os utilizadores queiram efectuar.

## **1.3 OBJECTIVO**

Apresentar uma proposta para implementação de um processo de distribuição de dados na atribuição de NUIT, baseado na arquitectura C/S, usando processamento distribuído.

## **CAPÍTULO II: MATERIAL E METODOLOGIA**

### **2.1 MATERIAL**

Material usado:

- Um computador Pentium 4 CPU 2.00 Ghz, 224 MB de RAM, Sistema Operativo Windows XP Professional 2002;
- Microsoft Word 2003 na elaboração do texto;
- Microsoft Visio 2003 para o desenho dos diagramas;
- Impressora HP LaserJet 2100

### **2.2 METODOLOGIA**

Este estudo é qualitativo e baseou-se na análise qualitativa e análise de conteúdo através de:

- Revisão bibliográfica, através de pesquisas em livros sobre Bases de Dados Distribuídas e consultas à *Internet*
- Revisão e aprofundamento de conhecimentos adquiridos ao longo da vida académica.
- Revisão de documentação sobre NUIT, existente na instituição (DGI).

## **CAPÍTULO III: REVISÃO BIBLIOGRÁFICA**

### **3.1. Arquitectura Cliente /Servidor (C/S)**

Com a evolução da tecnologia na área de redes de computadores e com o aperfeiçoamento da interconexão remota, algumas novas tecnologias introduzidas, proporcionaram que terminais e servidores pudessem interconectar-se transparente e eficientemente entre si, mesmo usando plataformas diferentes, permitindo assim que, dados e aplicações possam estar distribuídos pela rede.

A designação de arquitectura C/S estabelece a distinção entre dois tipos de processos com comportamentos diferentes: os servidores que implementam um conjunto de funções de interesse geral para outros processos que remotamente lhes poderão aceder e os clientes que efectuam a interface com os utilizadores, podendo executar parte das aplicações localmente e acederem remotamente a processos servidores para executarem as funções mais complexas de manipulação de dados, Entrada/Saída, impressão, etc. (Marques e Guedes, 1999).

Nesta secção e, pelo interesse que tem no trabalho, apresenta-se o conceito de C/S, a estratégia de implementação e as suas vantagens e desvantagens.

Várias definições sobre arquitectura C/S podem ser descritas:

- O termo C/S refere-se ao método de distribuição de aplicações computacionais através de muitas plataformas. Tipicamente essas aplicações estão divididas entre um provedor de acesso e uma central de dados e numerosos clientes contendo uma interface gráfica aos usuários para aceder e manipular os dados;
- O C/S geralmente refere-se a um modelo onde dois ou mais computadores interagem de modo que um oferece serviços aos outros. Este modelo permite aos usuários acederem informações e serviços de qualquer lugar;
- C/S é uma arquitectura computacional que envolve requisições de serviços de clientes para os servidores. Uma rede C/S é uma execução lógica de programação modular.

Para melhor se entender o paradigma C/S é necessário observar que o conceito chave está na ligação lógica e não na física. O cliente chave e o servidor podem coexistir ou não na mesma máquina (Renaud, 1994). Porém, um ponto importante para uma real abordagem C/S é a necessidade de que a arquitectura definida represente uma computação distribuída (Mckie, 1997).

Algumas das características do cliente e do servidor são descritas a seguir: (Salemi, 1993) e (Hulquist, 1997)

### **Cliente**

- Também denominado de “*front-end*” e “*Workstation*”, é um processo que interage com o usuário através de uma interface gráfica ou não, permitindo consultas ou comandos para recuperação de dados e análise, representando o meio pelo qual os resultados são apresentados;
- É o processo activo na relação C/S;
- Inicia e termina as conversações com o(s) servidore(s), solicitando serviços distribuído(s);
- Não se comunica com outros clientes;
- Torna a rede transparente ao usuário.

### **Servidor**

- Também denominado “*back-end*”, fornece um determinado serviço que fica disponível para todo o cliente que o necessita. A natureza e escopo do serviço são definidos pelo objectivo da aplicação C/S;
- É um processo reactivo na relação C/S;
- Possui uma execução contínua;
- Recebe e responde às solicitações dos clientes;
- Não se comunica com outros servidores enquanto estiver a fazer papel de servidor ;
- Presta serviços distribuídos;

- Atende a diversos clientes.

Dentre várias vantagens e desvantagens da arquitectura C/S podem-se citar: (Salemi, 1993)

#### Vantagens

- **Confiabilidade**  
Se uma máquina apresenta algum problema, ainda que seja um dos servidores, parte do sistema continua activo;
- **Matriz de computadores agregando capacidade de processamento**  
A arquitectura C/S provê meios para que as tarefas sejam feitas sem a monopolização dos recursos. Usuários finais podem trabalhar localmente;
- **O sistema cresce facilmente**  
Torna-se fácil modernizar o sistema quando necessário;
- **O cliente e o servidor possuem ambientes operacionais individuais/sistemas abertos**  
Pode-se misturar várias plataformas para melhor atender as necessidades individuais de diversos sectores e usuários.

Além destas vantagens, pode-se encontrar dentro de uma arquitectura C/S a interoperabilidade das estações Clientes e Servidoras entre redes de computadores, a escalabilidade da arquitectura visando o crescimento e a redução dos elementos constituintes, a adaptabilidade á novas tecnologias desenvolvidas, a performance do *hardware* envolvido na arquitectura, a portabilidade entre as diversas estações que compõem a arquitectura e a segurança dos dados e processos (Mckie, 1997).

Embora o avanço da arquitectura C/S tenha trazido uma variada gama de facilidades para o desenvolvimento de aplicações distribuídas, também possui algumas desvantagens:

### Desvantagens

- **Manutenção** - As diversas partes envolvidas nem sempre funcionam bem juntas. Quando algum erro ocorre, existe uma extensa lista de itens a serem investigados;
- **Ferramentas** - A escassez de ferramentas de suporte, não raras vezes obriga ao desenvolvimento de ferramentas próprias. Em função do grande poderio das novas linguagens de programação, esta dificuldade está se tornando cada vez menor;
- **Treinamento** - A diferença entre a filosofia de desenvolvimento de *software* para o microcomputador de um fabricante para o outro, não é como de uma linguagem de programação para outra. Um treinamento mais efectivo torna-se necessário;
- **Gestão** - Aumento da complexidade do ambiente e escassez de ferramentas de auxílio tornam difícil a gestão da rede.

#### 3.1.1 Arquitectura Cliente/Servidor de Base de Dados

Em termos genéricos, existe uma arquitectura C/S sempre que há um cliente que faz solicitações a um servidor que as atende. Normalmente, as duas entidades residem em máquinas distintas, interligadas por meio de comunicação, contudo, nada impede que residam numa mesma máquina. Tendo na arquitectura, uma ou várias máquinas dotada de capacidade de disco, poderiam ser usadas para armazenar uma grande quantidade de dados (servidor), permitindo a sua partilha pelas restantes máquinas ligadas a rede (clientes). O resultado final desta cooperação entre clientes e servidores é um sistema com grande capacidade de armazenamento e processamento global (Pereira, 1998).

A utilização de computadores interligados em rede com servidor de BD, proporciona o melhor rácio preço/desempenho, consistindo basicamente, em repartir a carga total de processamento entre os dois elementos (clientes e servidores). Esta abordagem, ao dedicar máquinas a funções específicas, facilita a selecção do respectivo *hardware* (servidores com grande capacidade de

processamento e armazenamento) e clientes com boas capacidades gráficas, permitindo o aproveitamento máximo das suas funcionalidades.

A ideia base desta solução, é separar o nível aplicacional das funções de gestão de BD. Desta forma, o processamento é dividido entre os computadores clientes que correm aplicações e os computadores onde a BD está armazenada (Servidores de BD), cujo única função é correr o Sistema de Gestão de BD (SGBD).

A característica fundamental de um sistema de BD C/S, é a divisão do processamento global entre as máquinas clientes, responsáveis pelas funções "front-end" com os utilizadores (com o nível aplicacional) e as máquinas servidoras, responsáveis pela tarefa de gestão e monitoração da BD, ou seja funções "back-end".

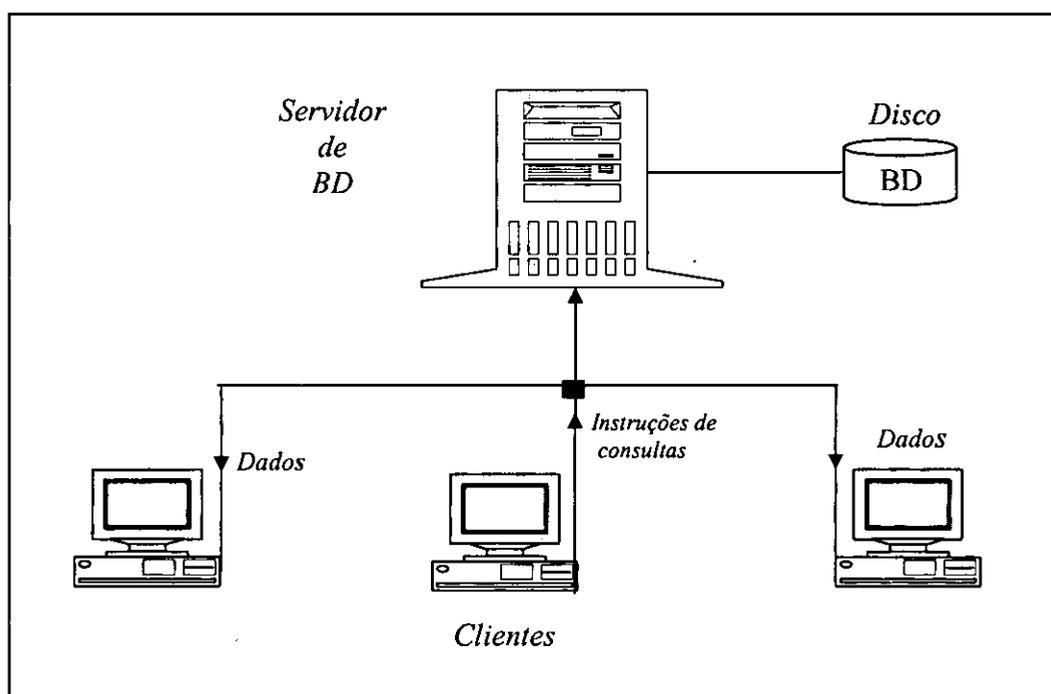


Figura 1: Sistema de BD C/S

Apesar do processamento em sistema de BD cliente/servidor ser normalmente atribuído ao cliente, resumindo-se o servidor ao papel de gestão e manipulação de dados, com a possibilidade de mapeamento de códigos para o domínio do SGBD (sob forma de *triggers*, *stored procedure*),

o papel do servidor torna-se mais activo, no processamento das aplicações, uma vez que uma parte do código das aplicações pode residir e ser executado no servidor, resultando nas seguintes vantagens:

- Tira-se maior proveito, da capacidade de processamento dos servidores, que são normalmente, máquinas com maior capacidade que os clientes;
- Diminui-se o tráfego na rede, uma vez que grande parte do código das aplicações é executado na mesma máquina onde estão os dados e ;
- Melhora-se a fiabilidade das aplicações, na medida em que há código partilhado por todas elas. Se a correcção do código for assegurada no servidor, então a correcção das aplicações está em grande parte garantida.

A utilização deste tipo de facilidades, também traz alguns inconvenientes, dado que o código em que estão escritos estes procedimentos, é na sua maioria proprietário ou específico em cada servidor de BD, assim, a portabilidade fica comprometida. Por outro lado, é necessário observar que o uso exaustivo destas facilidades pode sobrecarregar os servidores.

Num sistema C/S, é possível identificar três elementos: a interface com utilizador, a lógica das aplicações e os dados. Na sua configuração típica, um sistema C/S pressupõe a existência de apenas dois níveis, o cliente e o servidor de BD. A esta a configuração nomea-se two-tier.

#### 3.1.1.1 Configuração C/S a dois níveis

Na configuração a dois níveis (*two-tier*), o elemento “lógica das aplicações”, pode surgir de várias formas: integrado com a “ interface com os utilizadores”, ao nível dos clientes, dando origem aos chamados “*fat clients*”, ou integrado com os “dados” ao nível dos servidores, originando os chamados “*fat servers*” ou ainda integrado nos dois níveis.

A configuração *two-tier*, tem como grande vantagem, o facto de ser uma abordagem muito simples. Contudo, é reconhecido que sua utilização apenas se torna viável em situações pouco exigentes. Quando o número de máquinas clientes aumenta, os servidores de BD podem não dar resposta às solicitações que lhes são feitas no tempo requerido, pela degradação do desempenho.

Este inconveniente pode ser ultrapassado, pela introdução dum outro interveniente na arquitectura C/S, o designado servidor de aplicações, que se posiciona entre os clientes e os servidores de BD com o intuito de suportar o elemento “lógica das aplicações”. Com este elemento, surge então uma nova configuração com três níveis, em que a lógica das aplicações constitui um nível intermédio (*2nd-tier*), separado da interface com utilizadores (*1st-tier*) e da BD (*3rd-tier*).

### 3.1.1.2 Configuração C/S a três níveis

Na configuração a três níveis (*three-tier*), dada a divisão explícita de funções entre os vários intervenientes, propõe os servidores de Bases de dados, suportarem a gestão e o armazenamento dos dados, os servidores de aplicações dedicados na execução dos códigos relativos as diferentes aplicações e os clientes correndo apenas a interface das várias aplicações.

Ao trazer-se se a lógica aplicacional para o nível intermédio (*2nd-tier*), separado quer da interface com utilizadores como da BD, proporciona-se as seguintes vantagens:

- Por um lado, o maior desempenho e escalabilidade, conseguido com a divisão da carga total de processamento, por outro lado a reutilização do código aplicacional, pelos vários clientes (de forma vantajosa que a solução *fat servers*);
- Pode-se também implementar o conceito multi – tier, pela fragmentação do código dum máquina, definindo-se em que máquina na rede podem ser executados determinados blocos do código de aplicação, permitindo-se desta forma, que o desenvolvimento dum aplicação seja de forma usual (como se fosse para uma máquina).

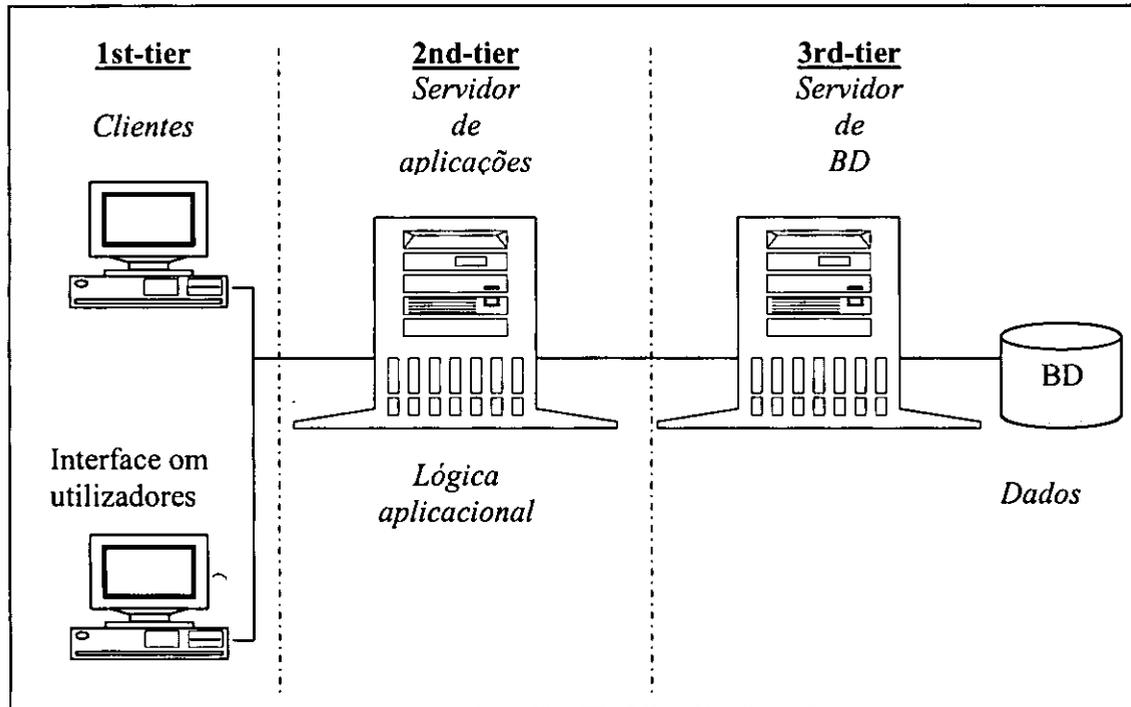


Figura 2: Cliente/Servidor three - tier

### 3.1.2 Alguns Tipos de Processos numa Arquitectura C/S

A arquitectura C/S divide uma aplicação em processos, que são executados em diferentes máquinas conectadas à uma rede de computadores, formando um único sistema. O paradigma da tecnologia C/S serve como um modelo, entre outros (Andrews, 1991), para interacção entre processos de *softwares* em execução concorrente.

Os processos ou tarefas, a serem executadas são divididas entre o Cliente e o Servidor, dependendo da aplicação envolvida e das restrições impostas pelo Sistema Operativo (SO) de Rede. Quanto mais avançado for o SO da Rede, menor será a aplicação em si, uma vez que a implementação do código para aceder a rede já se encontra definida no SO.

Actualmente dois tipos de processamentos estão sendo divulgados, Processamento Distribuído e Processamento Cooperativo. A característica de cada um destes é descrita a seguir (Kalokata, 1997).

### **Processamento Distribuído**

A distribuição de aplicações e tarefas se faz através de múltiplas plataformas de processamento. O processamento distribuído implica que essas aplicações/tarefas irão ocorrer em mais de um processo, na ordem de uma transação a ser concluída. Em outras palavras, o processamento é distribuído através de duas ou mais máquinas e os processos, na maioria, não correm ao mesmo tempo; Por exemplo, cada processador realiza parte de uma aplicação numa sequência. Geralmente, o dado usado num ambiente de processamento distribuído também é distribuído através de plataformas.

### **Processamento Cooperativo**

A cooperação requer dois ou mais processadores distintos para completar uma simples transação. O processamento cooperativo é relatado para ambos os processos cliente-servidor. É uma forma de computação distribuída onde dois ou mais processadores distintos são requeridos para completar uma simples transação. Normalmente esses programas interagem e executam concorrentemente como diferentes processos. Os processos cooperativos também são considerados como estilo de C/S através da arquitectura de mensagens, que devem obedecer a um determinado padrão.

#### **3.1.2.1 Processamento Distribuído Cliente/Servidor**

##### **Processo C/S**

- Existem processos distintos: o processo cliente é diferente do processo servidor;
- Os processos servidores tornam a estação Servidora dedicada ao seu trabalho;
- Processos clientes são sempre clientes;
- Processos servidores podem se tornar processos clientes de outros Servidores.

A característica básica da arquitectura C/S é a que processos Clientes enviam pedidos a um processo Servidor, que retorna o resultado para o Cliente. O processo Cliente fica então libertado da acção do processamento da transacção podendo realizar outros trabalhos.

Existem vários sistemas que podem ser baseados na estrutura C/S. O uso mais frequente são as aplicações de BD usando processos SQL de *front-end*, para aceder remotamente, as bases de dados.

### **3.2 SISTEMAS DISTRIBUÍDOS**

A arquitectura C/S, apesar do seu mérito, em relação ao elemento “dados”, não se afasta muito da solução centralizada de BD. Em seguida, analisa-se com algum detalhe o modelo distribuído de BD, as razões do seu interesse e os novos problemas que se colocam, em particular aos SGBDs.

De acordo com (Pereira, 1998) uma BD distribuída é um sistema de BD cujos dados se encontram fisicamente dispersos por várias máquinas ligadas por meios de comunicação e interligados logicamente.

(Date, 1995) de forma mais abrangente, define uma BD distribuída, como sendo um conjunto de centros de computação, ligados por rede de computadores em que, por um lado, cada centro constitui um sistema de BD por si, por outro lado, os vários centros concordam em cooperar, de tal forma que um utilizador de um dos centros possa utilizar dados armazenados nos outros centros como se esses dados lhes fossem locais.

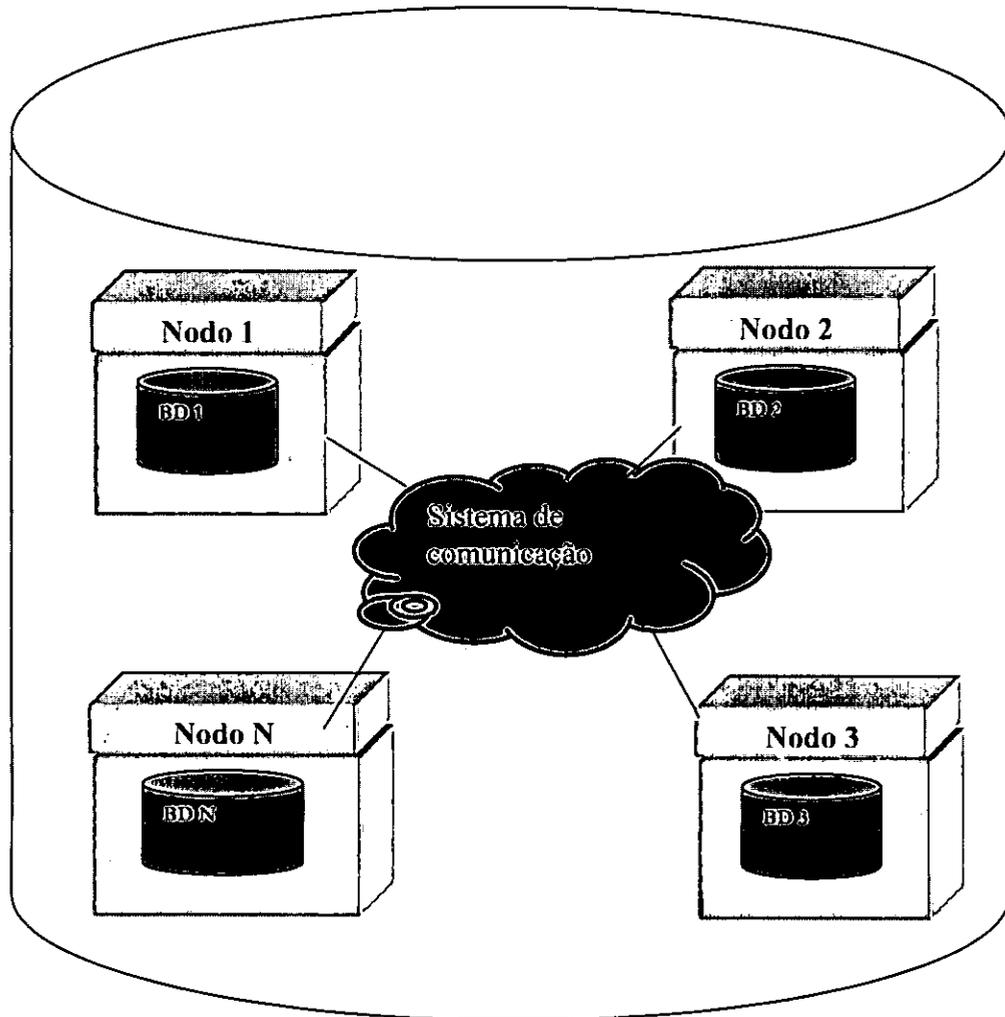


Figura. 3: BD lógica distribuída fisicamente por vários centros mas integrada através de um sistema de comunicação

Os sistemas de Bases de Dados distribuídas podem ser classificados, genericamente em duas classes: Homogénea e Heterogénea.

**Homogénea:** Em que todos os nodos usam o mesmo SGBD, como se fosse um único sistema de BD, mas em que, em vez de todos dados estarem armazenados num único repositório, estão armazenados por vários repositórios ligados por meio de comunicação. O modelo relacional constitui a melhor tecnologia para suportar a gestão de BD distribuídas. Como resultado, grande parte dos sistemas de bases de dados distribuídas existentes e em desenvolvimento são relacionais (Brigh, 1992).

**Heterogénea:** Este tipo de sistema de BD distribuída caracteriza-se pela existência de SGBDs diferentes nos vários nodos. As diferenças entre os SGBD presentes pode-se colocar a vários níveis, desde SGBDs diferentes baseados no mesmo modelo de dados, até SGBDs baseados em modelos de dados diferentes.

O conceito de BD distribuída, apesar de não ser novo, só recentemente foi encarado como uma área com grande potencial de desenvolvimento e interesse (Cohen, 1998).

Além das características de distribuição geográfica de algumas organizações, uma outra característica é a heterogeneidade dos seus recursos informáticos. As organizações, algumas vezes utilizam diferentes plataformas de computação, desde computadores pessoais, minicomputadores até mainframes, diversos sistemas operativos e tecnologias de rede. A pouca homogeneidade, se existe, reduz-se ao nível departamental. As fusões e aquisições das organizações são outros motivos para a existência da heterogeneidade dos recursos informáticos pois cada organização geralmente tem suas opções tecnológicas.

Um princípio fundamental dos sistemas de bases de dados distribuídos é a sua transparência relativamente ao nível aplicacional. Enquanto que nos sistemas homogéneos, dada a semelhança dos SGBDs em presença, o problema de transparência envolve fundamentalmente, a transparência na localização dos dados, nos sistemas heterogéneos este requisito traz constrangimentos ainda complexos. Adicionalmente à transparência na localização dos dados, surge a necessidade de transparência nos acessos. A eliminação completa da heterogeneidade (transparência total) é na prática, muito complexa (Bright, 1992).

Actualmente dada a maior flexibilidade, desempenho (quantidade de bits transmitidos por segundo) e cada vez mais baixos custos das comunicações (maior quantidade de bits transmitidos por unidade de custo), tornam viáveis os serviços oferecidos pelas tecnologias de redes de computadores em aplicações mais exigentes e intensivas. Contudo, quer seja em redes LAN como em redes WAN, as taxas de transmissão dos dados serão inferiores às taxas a que os computadores são capazes de os processar pelo que um dos factores mais importantes para otimizar o desempenho de qualquer sistema distribuindo é a minimização do tráfego nas redes.

A ideia subjacente aos sistemas distribuídos de bases de dados é a distribuição de dados por centros de processamento independentes e a partilha mais transparente desses dados, entre os vários centros (nós), fazendo o uso do sistema de comunicação. Como resultado surge a divisão de uma única BD lógica, distribuída fisicamente por vários nós que cooperam entre si partilhando dados e capacidade de processamento.

Existe, basicamente duas razões que sugerem a abordagem distribuída de uma BD que são:

1. **Reflectir o estado distribuído de uma organização:** Esta geralmente é resultado de uma organização encontrar-se dispersa por zonas geograficamente distantes, com necessidade de uma coordenação comum ou pela descentralização de funções nas próprias estruturas organizacionais.
2. **A necessidade de crescimento:** Os sistemas centralizados de BD são sujeitos a problemas de escalabilidade, ou seja, dificilmente mantêm desempenhos idênticos, quando o volume da BD e ou o número de utilizadores simultâneos aumenta.

A abordagem distribuída permite um crescimento virtualmente ilimitado da organização a medida que vão se criando novos nós na rede, que constitui o sistema de comunicação particular da organização. Mesmo em organizações não dispersas geograficamente, que possuem unidades departamentais relativamente autónomas, dispendo das suas próprias Bases de Dados com processamento específico, é desejável que os vários recursos informáticos da organização, sejam vistos como um todo e acedidos por toda organização. No entanto, a partilha concorrente de dados, vem agravar problemas existentes na abordagem centralizada, colocando novos desafios na área da gestão da BD.

Uma das características mais importantes de um sistema de BD distribuído, é a transparência na localização dos dados, relativamente ao nível aplicacional. O utilizador deve ter sensibilidade de que os dados lhes estão disponível localmente, idealmente um sistema de BD distribuído deveria apresentar-se ao nível aplicacional como um só sistema, ou seja, o nível operacional deveria aceder a dados remotos, com a mesma facilidade com que acederia se esses dados residissem numa mesma máquina (local). Ao satisfazer estas características, o sistema de BD distribuídas, combina dois conceitos divergentes (Bell & Grimson, 1992):

- Distribuição física dos dados;
- Integração lógica dos mesmos.

A transparência na localização dos dados, deverá ser uma característica do nível operacional ou seja, da mesma forma que os utilizadores não necessitam de ter o conhecimento da localização dos dados, as aplicações não devem conhecer esses detalhes deixando esta tarefa para o SGBD. Se no acesso aos dados por parte das aplicações, houvesse necessidade de referenciar explicitamente as suas localizações físicas, haveria pouca flexibilidade para o desenvolvimento e manutenção de sistemas. Desta forma, as aplicações e os utilizadores não devem ser afectados por qualquer alteração na localização dos dados (Bell & Grimson, 1992). Relativamente as características desejáveis num sistema de bases de dados distribuídas, advem do princípio fundamental desses sistemas, ou seja, que na perspectiva do utilizador, um sistema de BD distribuídas se comporte da mesma forma que um sistema de BD centralizado.

### 3.2.1 Sistema de Gestão de BD (SGBD)

Para (Date, 1991) SGBD é o *software* responsável pela gestão (armazenamento e recuperação) dos dados na BD, sendo basicamente um sistema cujo objectivo é manter as informações e torná-las disponíveis quando solicitadas.

(Korth e & Silberschatz, 1994) definem SGBD como uma colecção de dados inter-relacionados e um conjunto de programas para acedê-los, sendo que o principal objectivo do mesmo é prover um ambiente que seja conveniente e eficiente para a recuperação e o armazenamento de informações das BD.

A interface entre essas duas camadas é uma linguagem padrão para consulta, manipulação e controle de acesso aos dados. A linguagem mais utilizada para essa interface é o Structured Query Language (SQL).

Existem diversos tipos de SGBD, de pequenos sistemas que correm em computadores pessoais a gigantescos sistemas que correm em mainframes.

Alguns exemplos de SGBD: *MySQL, Oracle, PostgreSQL, SQL-Server*

Segundo (Ramalho, 2002) Uma BD *oracle* tem uma estrutura física e lógica. Como essas estruturas no servidor são separadas, o armazenamento físico dos dados pode ser gerido sem afectar o acesso às estruturas lógicas de armazenamento.

**ESTRUTURA LÓGICA** do *Oracle* é determinada por um ou mais tablespaces e pelos objectos de esquema da BD.

**Tablespaces:** são unidades lógicas de armazenamento. Elas podem estar *online* ou *off-line*, mas normalmente estão *online* para que os usuários possam aceder as informações de que necessitam. Entretanto, podem ficar *off-line* para tornar parte da BD indisponível, permitindo ao mesmo tempo, o acesso a restante BD, o que facilita a execução de muitas tarefas administrativas.

**Esquemas e Objectos de Esquema:** Esquema é uma colecção de objectos de BD. Os objectos de esquema são estruturas lógicas que se referem directamente aos dados. Os objectos de esquema incluem estruturas, tais como: Tabelas, visões, sequências, procedimentos armazenados, sinónimos, índices, clusters e links de BD.

**ESTRUTURA FÍSICA** de uma BD *oracle* é formado por três tipos de arquivos: um ou mais datafiles, dois ou mais arquivos de registo redo e um ou mais arquivos de controle. Esses arquivos fornecem para as informações do *oracle* um armazenamento físico real.

**Datafiles:** Contém todos os dados de uma BD. Os dados das estruturas lógicas, tais como as tabelas e os índices são armazenados fisicamente nos datafiles alocados.

**Arquivos de Registo Redo:** Toda a BD *oracle* tem um ou dois conjuntos de arquivos de registo redo. Eles são conhecidos como registo redo da BD. Um registo redo é formado por entradas redo no qual cada uma delas é um grupo de vectores de alteração que descrevem uma única modificação atómica feita na BD.

**Arquivos de Controle:** Toda a BD *oracle* tem um arquivo de controle o qual contém entradas que especificam a estrutura física da BD. Por exemplo, ele contém os seguintes tipos de

informação: nome da BD; nomes e localizações dos datafiles de uma BD e dos arquivos de registo redo. Para aceder a BD e os dados o *oracle* usa a Structured Query Language (SQL).

### 3.2.1.1 Structured Query Language (SQL)

É uma linguagem de pesquisa que permite a manipulação de dados numa BD. Uma BD pode ter uma ou mais tabelas, e cada tabela é composta por colunas e linhas.

Pode-se definir e manipular os dados de uma tabela com os comandos SQL. Usa-se comandos da Data Definition Language (DDL) para configurar os dados, criar e alterar as BD e as tabelas. Pode-se actualizar, excluir, extrair, alterar ou recuperar dados de uma tabela com os comandos Data Manipulation Language (DML) (Ramalho, 2002).

Segundo (Morelli, 2002), a relação seguinte explica alguns comandos SQL:

**Tabela 1: Principais comandos do SQL e o tipo de linguagem**

Comando	Descrição	Tipo
<i>select</i>	Recupera dados de uma ou mais tabelas. É o comando mais utilizado em SQL	DML
<i>insert, update e delete</i>	Servem para incluir, alterar e eliminar linhas de uma tabela respectivamente	DML
<i>commit</i> <i>rollback</i>	Responsáveis pelo controle de transações, permitem que o usuário desfça (ROLLBACK) ou confirme (COMMIT) alterações em tabelas	DML
<i>create, alter e drop</i>	Úteis para definir, alterarem e remover estruturas da BD	DDL

### 3.2.2 Replicação de dados

Replicação é o processo de copiar e actualizar os objectos de BD em vários bancos de dados que formam um sistema de BD distribuído. As alterações feitas num *site* são capturadas e armazenadas localmente antes de serem encaminhadas e aplicadas a cada uma das localizações remotas. A replicação fornece ao usuário acesso rápido e local aos dados compartilhados, e

protege a disponibilidade de aplicativos porque alterna as opções de acesso aos dados que existem (Ramalho, 2002).

#### **3.2.2.1 Necessidade da Replicação de dados**

Utilizando a replicação, os dados compartilhados podem ser distribuídos automaticamente para a plataforma destino, diminuindo o tráfego na rede e melhorando os requisitos de disponibilidade. Isto pode ser usado por aplicações que são obrigadas a compartilhar dados com outras aplicações.

A tecnologia de replicação pode viabilizar as aplicações de suporte a decisão pela melhora de disponibilização dos dados, performance no acesso aos dados, e sua usabilidade. Isto ajuda funcionários de uma empresa a terem informações seguras e actualizadas para poderem tomar decisões efectivas e exactas.

Os clientes que migram suas aplicações antigas ou replicam os dados entre vários ambientes podem utilizar a replicação para reduzir o tempo de desenvolvimento e o custo de manutenção da aplicação, os dados podem ser formatados através do acesso compartilhado de várias fontes de dados.

#### **3.2.3 Segurança de Base de Dados**

Os sistemas de BD de multiusuários, tais como o *Oracle*, incluem recursos de segurança que controlam o modo como a BD é acedida e usada. Por exemplo os mecanismos de segurança (Ramalho, 2002):

- Evitam o acesso a BD sem autorização;
- Evitam o acesso aos objectos de esquema sem autorização;
- Controlam o uso de recursos do sistema;
- Fazem auditoria das acções do usuário.

Cada usuário de BD tem um esquema associado com o mesmo nome. Um esquema, é uma colecção lógica de objectos de BD (tabelas, visões, funções, procedimentos, sequências, sinónimos, índices, agrupamentos, pacotes e links de BD). Como padrão, cada usuário cria e tem acesso a todos os objectos do esquema correspondente.

A segurança da BD pode ser classificada em duas categorias distintas: segurança de sistema e segurança de dados.

A **segurança de sistema** inclui os mecanismos que controlam o acesso e o uso da BD num determinado nível do sistema. Por exemplo, a segurança de sistema inclui:

- Combinações válidas de nome de usuário e senha;
- A quantidade de espaço em disco disponível para os objectos de esquema de um usuário;
- Os limites de recurso de um usuário.

Os mecanismos de segurança de sistema verificam se um usuário está autorizado a se conectar a BD, se a auditoria da BD está activa e quais operações de sistema um usuário pode executar (Ramalho, 2002).

A **segurança de dados** inclui os mecanismos que controlam o acesso e o uso da BD no nível de objecto de esquema. Por exemplo, a segurança de dados inclui:

- Quais usuários têm acesso a um objecto de esquema específico e aos tipos específicos de acções que cada um pode executar no objecto de esquema;
- As acções, se houverem, que são auditadas para cada objecto de esquema.

### **3.2.3.1 Mecanismos de Segurança**

O servidor oracle fornece controle arbitrário de acesso o que é um meio de restringir o acesso às informações privilegiadas. O privilégio apropriado deve ser atribuído por um usuário para que ele acesse a um objecto de esquema. Os usuários com privilégios apropriados podem concedê-los a outros, segundo o seu critério; por esse motivo, esse tipo de segurança é chamado de 'arbitrário' (Ramalho, 2002).

O *Oracle* gere a segurança da BD usando diversos recursos diferentes. Entre eles:

**Usuários e esquemas de BD** - Cada BD *Oracle* tem uma lista de nomes de usuários. Para aceder a uma BD, um usuário deve usar um aplicativo desse tipo e tentar uma conexão com um nome de usuário válido. Cada nome tem uma senha associada para evitar uso sem autorização;

**Privilégios** - Um privilégio é um direito para executar um determinado tipo de declaração SQL. Alguns exemplos de privilégios incluem: Direito de conectar-se a BD, direito de criar uma tabela no seu esquema, direito de seleccionar linhas da tabela de outra pessoa, os privilégios de uma BD *oracle* podem ser divididos em duas categorias distintas: os privilégios de sistema e os de objecto. Os privilégios de sistema permitem que os usuários executem determinado tipo de objectos de esquema como privilégios de criar tabela, alterar tabelas, etc. Os privilégios de objecto permitem que os usuários executem determinada acção num objecto de esquema também específico. Por exemplo, o privilégio de excluir linhas de uma tabela específica.

**Papéis (Roles)** - É um conjunto de privilégios (sistema ou objecto), reunidos sob um único nome que pode ser concedido a um ou mais usuário/roles.

**Profiles** - Quando os recursos disponíveis (tempo de CPU, tempo de conexão, espaço em memória) forem factores críticos, o DBA pode criar *profiles* (perfis) e também atribuí-los a usuários;

**Auditoria** - Representa um interessante recurso capaz de rastrear acções executadas por usuários. Apesar de onerar o funcionamento geral da BD, recomenda-se utilizá-lo sempre que for necessário documentar o que cada indivíduo estiver fazendo; talvez para melhorar políticas de utilização de recursos ou para investigar possíveis desvios. Ela pode ser executada em três níveis diferentes: auditoria de declaração, de privilégio e de objecto de esquema.

### 3.3 REDES DE COMPUTADORES

(Soares, Lemos & Colcher, 1995) definem que uma rede de computadores é um conjunto de módulos processadores capazes de trocar informações e compartilhar recursos, interligados por um sistema de comunicação.

Segundo (Tanenbaum, 1997) as redes de computadores classificam-se em:

LAN - Redes Locais de Computadores (Local Area Network); compreendem uma pequena área, geralmente um laboratório, uma escola não chegando a ultrapassar uma sala. Possuem altas taxas de transmissão, baixas taxas de erro, são geralmente redes privadas. Suportam com bom desempenho aplicações como transferência de dados, voz, vídeo, comunicação entre terminais e computadores etc.

MAN – Redes Metropolitanas (Metropolitan Area Network); compreendem grandes áreas metropolitanas como cidades ou distritos. Abrangem áreas superiores as LAN's e geralmente operam em velocidades superiores.

WAN - Redes Geograficamente Distribuídas (Wide Area Network) Compreendem grandes áreas como países, continentes e até o mundo, o maior exemplo deste tipo de rede é a *Internet*. Cabeamento dedicado como troncos de fibra óptica entre países ou links de satélite são usados neste tipo de rede.

Segundo (Turban, 1999), as redes podem ser caracterizadas segundo a sua topologia a qual determina a distribuição dos seus componentes. Os três principais tipos de topologia são:

Topologia Bus – Todos os componentes estão conectados a um cabo central, conhecido como *bus* ou *backbone*.

Topologia em anel (*ring*) – Todos os componentes são conectados no formato de um círculo, de forma que cada um deles esteja conectado directamente com outros dois, um de cada lado.

Topologia em estrela (*Star*) – Todos os componentes são conectados a um *hub/switch* central.

### 3.3.1. Componentes Básicas de uma Rede

Neste ponto aborda-se componentes básicos de uma rede, mas concretamente, naqueles que são utilizados na rede da DGI e RF's

**Hub** – é um dispositivo usado para conectar outros equipamentos que compõem uma rede. Com ele, as conexões da rede são concentradas, facto esse que faz com que o *hub* também seja conhecido como concentrador, ficando cada equipamento num segmento próprio. O *hub* é o elemento central de uma rede local, responsável por receber informações que chegam de várias direções e passar adiante em uma ou mais direções.

**Switch** – é um dispositivo de diversas portas, podendo cada uma delas ser conectada ou a várias estações ( sob forma de uma outra rede) ou a uma única estação. A sua função é segmentar uma rede muito grande em redes menores e menos congestionadas, de forma a melhorar o desempenho da rede. Esse aumento de performance é obtido fornecendo a cada porta do *switch* uma largura de banda dedicada. O *switch* prevê uma filtragem de pacotes entre redes que estejam separadas (Soares, 1997).

**Router** – é um dispositivo que conecta duas redes diferentes, roteando os pacotes entre elas. É operado nas camadas física, de enlace de dados e de rede do modelo OSI. O seu funcionamento é similar a uma *bridge*. A diferença é que o *router* prevê funcionalidades adicionais, como por exemplo, a capacidade de filtrar pacotes e transmiti-los para lugares diferentes, baseando-se em critérios que tenham sido pré-estabelecidos. Já uma *bridge* é um dispositivo independente de protocolo; isto é, ela simplesmente transmite pacotes sem analisá-los ou roteá-los novamente. Consequentemente, uma *bridge* é mais rápida que um *router*, mas, em compensação, o *router* é mais flexível.

A interligação entre os componentes acima descritos é feita através de meios de comunicação que podem ser: fibra óptica, satélite, microondas, etc.

**Fibra óptica:** são Filamentos finos de vidro ou plástico que transportam o feixe de luz gerado por um LED ou “*laser*”. Sua capacidade de transmissão de dados, em número de canais e velocidade, supera a tecnologia de fios de cobre (usados por exemplo, em para trançado). As fibras para comunicações ópticas são fios flexíveis de diâmetro de um fio de cabelo. Formados por dois materiais cristalinos e homogêneos. O material que ocupa o centro da fibra é denominado de núcleo e o externo, que o envolve, é denominado de casca, o qual possui um

índice de refração menor que o núcleo para possibilitar o fenómeno da reflexão total e, consequentemente, a propagação da luz.

**Satélite:** Um satélite pode ser comparado a um repetidor de microondas.

**Microondas:** Onda electromagnética que vibra a 1 GHz ou mais e corresponde a um comprimento de onda inferior a 30 centímetros.

Para se estabelecer um padrão único para as redes de computadores, foram definidos vários modelos de referência como o modelo de referência *OSI/ISO (Open System Interconnection /International Organization of Standardization)*.

### 3.3.2 Modelo de Referência OSI

Segundo (Tarouco, 1986) e (Tanenbaum, 1997) Em Março de 1977, foi constituído pela *ISO* um grupo de trabalho para a padronização da interconexão de sistemas de comunicação. Foi definida, inicialmente, uma arquitectura geral, denominada *Modelo de Referência OSI*, para servir de base para a padronização da conexão de sistemas.

O modelo *OSI* possui sete camadas como apresentado na figura. 3.5. Cada camada especifica um grupo específico de tarefas de comunicação. A norma descreve o escopo funcional de cada camada e os requisitos para a interface com as camadas adjacentes (serviços).

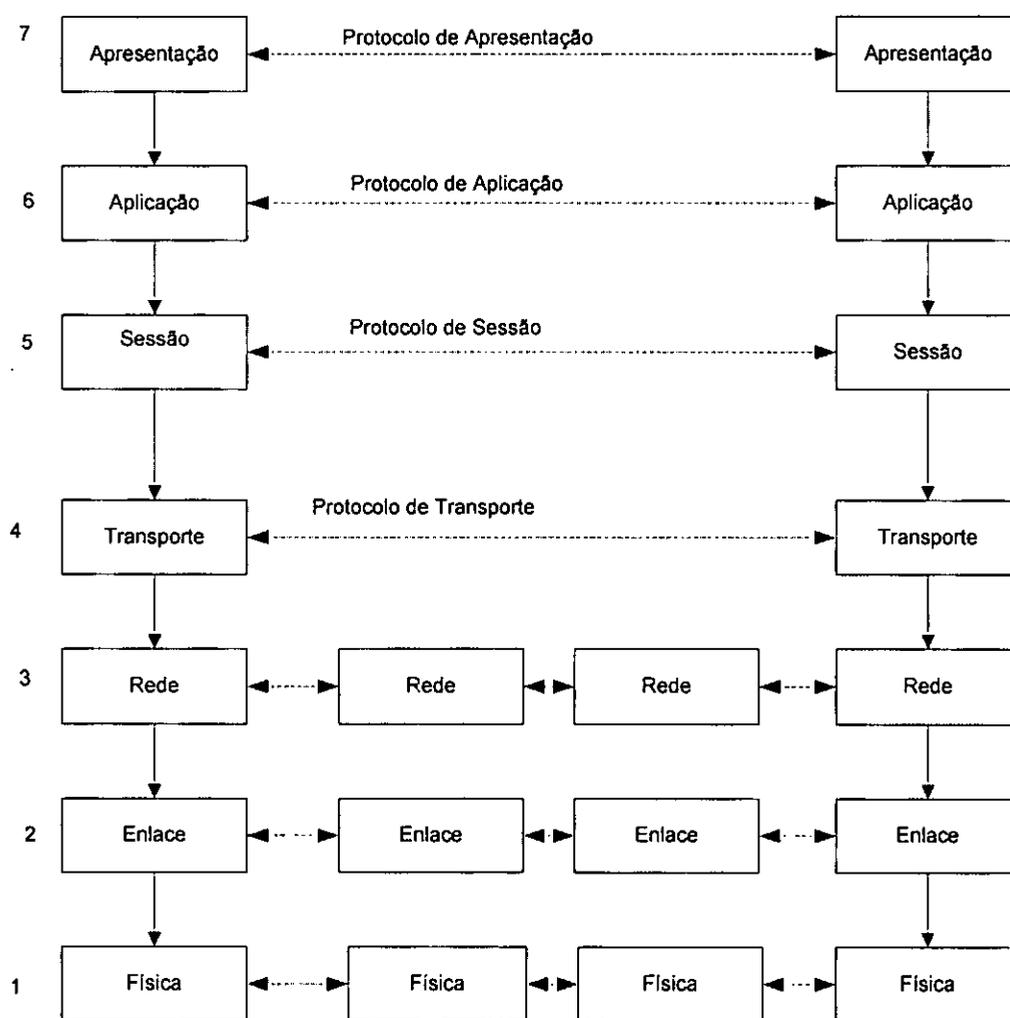


Figura 4: Arquitectura de Rede Baseada no modelo OSI

A tabela seguinte dá a descrição das funcionalidades de cada camada do modelo OSI

**Tabela 2: Classificação das camadas do modelo OSI**

NÍVEL	CAMADA	DESCRIÇÃO DAS FUNCIONALIDADES
7	Aplicação	Esta camada funciona como uma interface de ligação entre os processos de comunicação de rede e as aplicações utilizadas pelo usuário
6	Apresentação	Aqui os dados são convertidos e garantidos num formato universal
5	Sessão	Estabelece o processo de sequenciamento e, em alguns casos, confirmação de recebimento dos pacotes de dados.
4	Transporte	Efectua e encerra os enlaces de comunicação
3	Rede	Faz o roteamento dos dados através da rede
2	Enlace	Formata a informação em quadros "frames". Um quadro representa a exacta estrutura dos dados fisicamente transmitidos através do fio ou outro meio.
1	Física	Define a conexão física entre o sistema computacional e a rede. Especifica o conector, a "pingagem", níveis de tensão, dimensões físicas, características mecânicas e eléctricas, etc.

### 3.3.3 Protocolos

Para que os computadores de uma rede possam trocar informações é necessário que todos adoptem as mesmas regras para o envio e o recebimento de informações. Este conjunto de regras é conhecido como Protocolo de comunicação. Por outras palavras pode-se afirmar: "Para que os

computadores de uma rede possam trocar informações entre si é necessário que todos estejam utilizando o mesmo protocolo” (Comer, 1993).

No protocolo de comunicação estão definidas todas as regras necessárias para que o computador de destino, “entenda” as informações no formato que foram enviadas pelo computador de origem.

Existem vários tipos de protocolos, tais como: TCP/IP, NetBios, NetBEUI, Token Ring, Ethernet, etc.

Neste trabalho falar-se-á, apenas do protocolo TCP/IP porque é o protocolo usado na rede da DGI.

### **3.3.3.1 Transmission Control Protocol/Internet Protocol (TCP/IP)**

O TCP/IP distingue-se dos demais protocolos pelo seu endereçamento universal. Cada máquina numa rede possui um endereço que a identifica. A camada TCP é orientada à conexão, enquanto a camada de IP trabalha sem conexão (Comer, 1993) e (Davis, 1994).

Cabe a camada IP o trabalho de distribuir os datagramas entre as máquinas de uma rede. Funciona na camada de rede do modelo OSI, para fazer este serviço, ele possui um único endereço de 32 bits que contém informações suficientes para identificar univocamente uma rede de um determinado computador. Este endereço é comumente escrito em decimal de quatro bytes ou quatro octetos.

O protocolo IP utiliza três classes diferentes de endereços. A definição de classes de endereços deve-se ao facto do tamanho das redes que estão interligadas variar muito, indo desde redes locais de computadores a redes públicas interligando milhares de *hosts* (Soares, 1997).

*Host* é definido como sendo um computador ligado a uma rede de comunicação que possa usar os serviços providos pela rede para trocar dados com outros sistemas interligados.

Na primeira classe de endereços, a classe A, o bit mais significativo é 0, os outros 7 bits do primeiro octeto identificam a rede, e os 24 bits restantes definem o endereço local. Essa classe é

usada para redes de grande porte, seus endereços que variam de 1 a 126, e cada rede tem capacidade de endereçar cerca de 16 milhões de *hosts*.

A classe B de endereços usa dois octetos para o número da rede e dois para endereços de *hosts*. Os endereços de redes classe B variam na faixa de 128.1 até 191.255 (os números 0 e 255 do segundo octeto, e 127 do primeiro octeto são usados para funções especiais e testes), e cada rede pode interligar cerca de 65 mil *hosts*.

A classe C, utiliza três octetos para identificar a rede e apenas um octeto para o *host*. Os endereços de rede situam-se na faixa de 192.1.1 até 223.254.254, e cada rede pode endereçar 254 *hosts*.

Segundo (Comer, 1998) o TCP é um protocolo da camada de transporte da arquitetura Internet TCP/IP. O protocolo é orientado a conexão e fornece um serviço confiável de transferência de arquivos fim-a-fim. Ele é responsável por inserir as mensagens das aplicações dentro do datagrama de transporte, reenviar datagramas perdidos e ordenar a chegada de datagramas enviados por outro micro. O TCP foi projectado para funcionar com base num serviço de rede sem conexão e sem confirmação, fornecido pelo protocolo IP.

O protocolo TCP interage de um lado com processos das aplicações e do outro com o protocolo da camada de rede (níveis do modelo OSI). A interface entre o protocolo e a camada superior consiste num conjunto de chamadas. Existem chamadas, por exemplo, para abrir e fechar conexões e para enviar e receber dados em conexões previamente estabelecidas. Já a interface entre o TCP e a camada inferior define um mecanismo através do qual as duas camadas trocam informações assincronamente.

Este protocolo é capaz de transferir uma cadeia (*stream*) contínua de octetos, nas duas direcções, entre seus usuários. Normalmente o próprio protocolo decide o momento de parar de agrupar os octetos e de, conseqüentemente, transmitir o segmento formado por esse agrupamento. Porém, caso seja necessário, o usuário do TCP pode requerer a transmissão imediata dos octetos que estão no buffer de transmissão, através da função *push* (Soares, 1997).

Conforme mencionado, o protocolo TCP não exige um serviço de rede confiável para operar, logo, responsabiliza-se pela recuperação de dados corrompidos, perdidos, duplicados ou entregues fora de ordem pelo protocolo de rede. Isto é feito associando-se cada octeto a um número de sequência. O número de sequência do primeiro octeto dos dados contidos num segmento é transmitido junto com o segmento e é denominado número de sequência do segmento (Tanenbaum, 1997).

O reconhecimento constitui-se do número de sequência do próximo octeto que a entidade TCP transmissora espera receber do TCP receptor na direcção oposta da conexão. Por exemplo, se o número de sequência X for transmitido no campo *Acknowledge* (ACK), ele indica que a estação TCP transmissora recebeu correctamente os octetos com número de sequência menores que X, e que ele espera receber o octeto X na próxima mensagem (Soares, 1997).

## **CAPÍTULO IV: ANÁLISE E DISCUSSÃO SOBRE AS REDES DGI E RF'S**

A rede da DGI é uma rede LAN que se liga às diferentes RF's espalhadas pelo país. O sistema de comunicação para interligar as RF's com a DGI é o sistema das Telecomunicações de Moçambique (TDM), através dos seguintes meios/canais de comunicação: fibra óptica, satélite e microondas.

É usada a topologia em estrela que caracteriza a espinha dorsal da rede nacional de telecomunicações suportada pela tecnologia via satélite e pelo *link* de fibra de vidro instalado entre as cidades de Maputo e Beira e pontos de amarração em Xai-Xai, Inhambane e Vilankulos.

### **4.1. Situação actual**

O sistema actual funciona com a BD centralizada, em que a aplicação está integrada com os dados a nível do servidor que se encontra na central, DGI, onde corre SGBD e a própria aplicação.

O processo de atribuição de NUIT é moroso, pois existem várias RF's e cada uma com vários contribuintes solicitando NUIT. Cada pedido entregue a RF pelo contribuinte requer a inserção dos dados deste no sistema para posterior atribuição do NUIT. Estes dados precisam de validação a qual é feita na central. Como são vários utilizadores solicitando o mesmo serviço, simultaneamente, o tempo de resposta é longo provocando lentidão no sistema.

Quando o número de máquinas clientes aumenta, o servidor da BD leva muito tempo a responder às solicitações que lhe são feitas devido a degradação do desempenho causada pelo grande volume de dados e grande número solicitações.

A figura. 5 apresentada a seguir dá uma visão da actual arquitectura da rede que faz a interligação entre a DGI e as RF's.

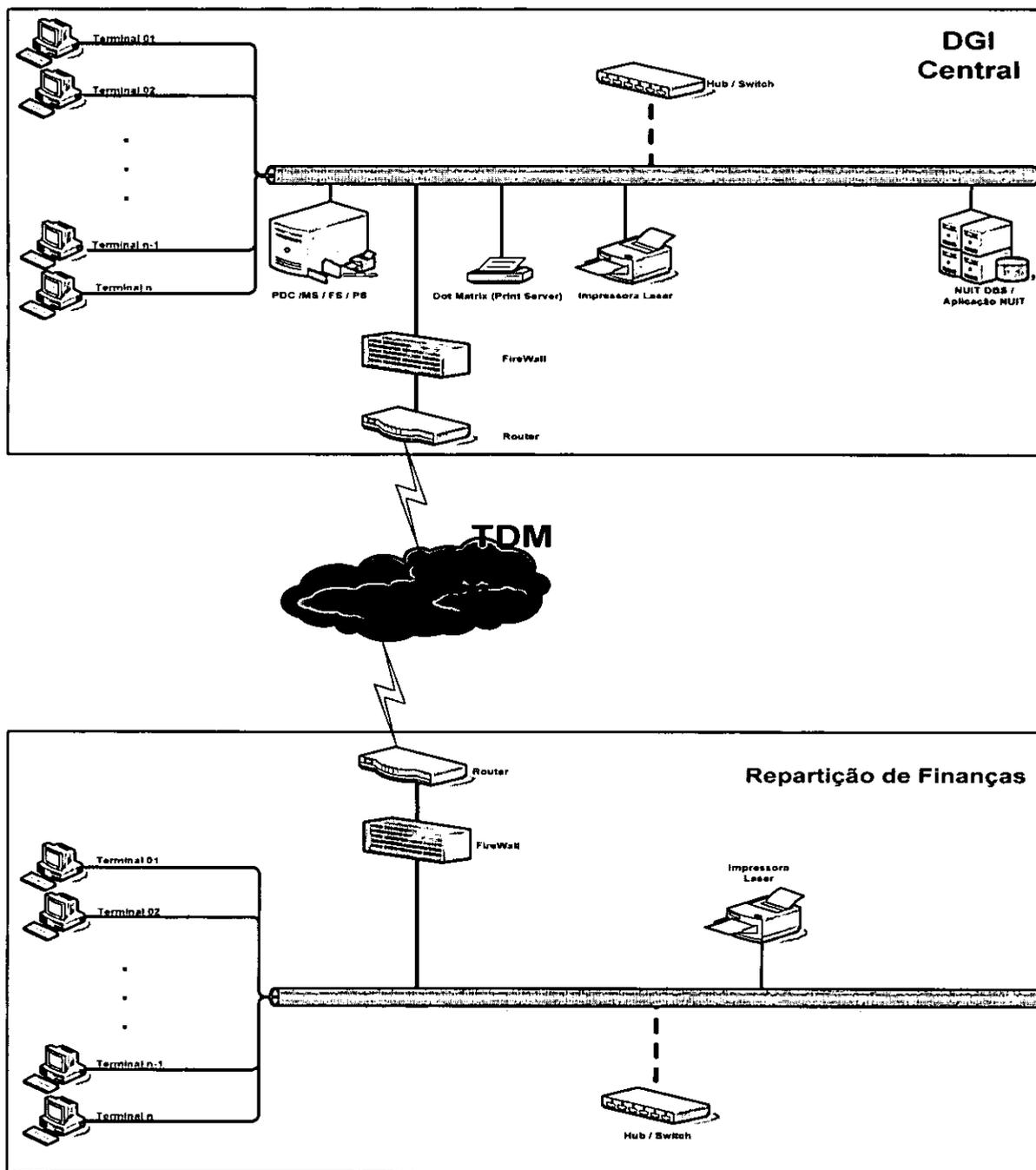


Figura. 5 Rede actual que interliga a DGI e as RF's

## 4.1.1 Descrição dos Componentes da Rede

TABELA 3: Descrição dos componentes da rede da DGI e RF

REDE	EQUIPAMENT Ⓞ	FUNÇÃO	SISTEMA OPERATIVO	BASE DE DADOS
DGI	<i>Hub</i>	Faz a interligação entre os restantes componentes dentro da rede	-	-
	<i>Router</i>	Faz a conexão entre esta rede e a rede RF	-	-
	<i>Firewall</i>	Barreira entre as duas redes para evitar intrusos	-	-
	Computador	PDC autentica os utilizadores e gere os recursos da rede	Windows 2000 Server	-
	Computador	Servidor de BD /Aplicação NUIT	Windows 2000 Server	Oracle 9i DB
	Computadores	Estações de trabalho "Workstation's"	Windows XP	Oracle Forms & Reports
RF	<i>Hub</i>	Faz a interligação entre os restantes componentes dentro da rede	-	-
	<i>Router</i>	Faz a conexão entre esta rede e a rede RF	-	-
	<i>Firewall</i>	Barreira entre as duas redes para evitar intrusos	-	-
	Computadores	Estações de trabalho "Worstation's"	Windows XP	Oracle Forms & Reports

## 4.2 Proposta para a nova rede

A nova proposta de rede é semelhante a actual, a diferença é que haverá um acréscimo de dois computadores nas RF's, uma que funcionará como "Primary Domain Controller" (PDC) para autenticação dos utilizadores e gestão dos recursos da rede e outro para funcionar como servidor de BD e da aplicação NUIT.

Esta opção prende-se com a necessidade de aproveitar a infraestrutura e os recursos existentes para minimizar os custos, tendo em vista o objectivo a alcançar.

A figura 6 dá-nos a visão geral da proposta de arquitectura de como deverá ser a interligação da rede DGI/RF's.

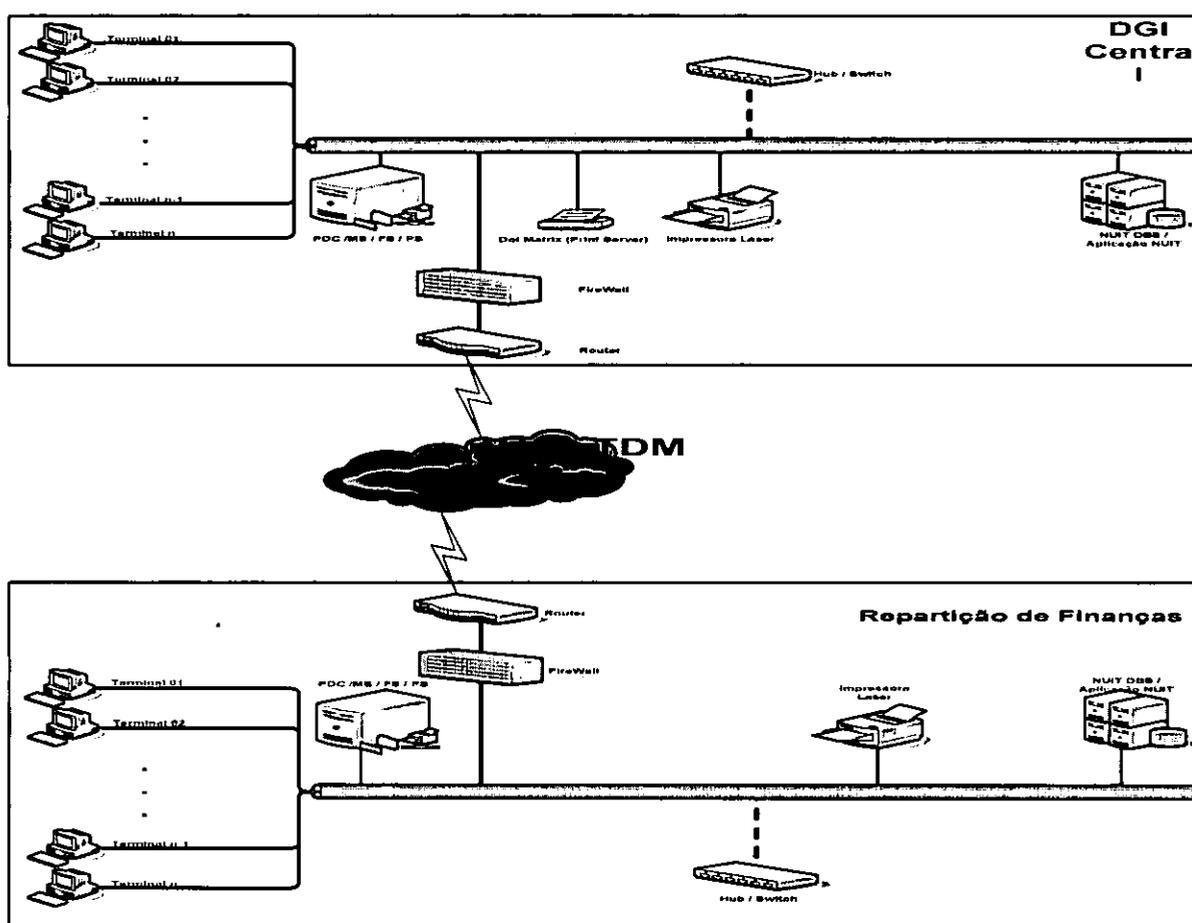


Figura. 6 Rede que interliga a DGI e as RF's

4.2.1 Descrição dos Componentes da Rede

REDE	EQUIPAMENTO	FUNÇÃO	SISTEMA OPERATIVO	BASE DE DADOS
DGI	<i>Hub</i>	Faz a interligação entre os restantes componentes dentro da rede	-	-
	<i>Router</i>	Faz a conexão entre esta rede e a rede RF	-	-
	<i>Firewall</i>	Barreira entre as duas redes para evitar intrusos	-	-
	Computador	PDC autentica os utilizadores e gere os recursos da rede	Windows 2000 Server	-
	Computador	Servidor de BD/Aplicação NUIT	Windows 2000 Server	Oracle 9i DB
	Computador	Estações de trabalho "Workstation's"	Windows XP	Oracle Forms & Reports
RF	<i>Hub</i>	Faz a interligação entre os restantes componentes dentro da rede	-	-
	<i>Router</i>	Faz a conexão entre esta rede e a rede RF	-	-
	<i>Firewall</i>	Barreira entre as duas redes para evitar intrusos	-	-
	Computador	PDC, autentica os utilizadores e gere os recursos da rede	Windows 2000 Server	-
	Computador	Servidor da BD/Aplicação	Windows2000Server	Oracle 9i DB
	Computadores	Estações de trabalho "Workstation"	Windows XP	Oracle Forms & Reports

### 4.3 Proposta do modelo da arquitectura do processo de distribuição dados

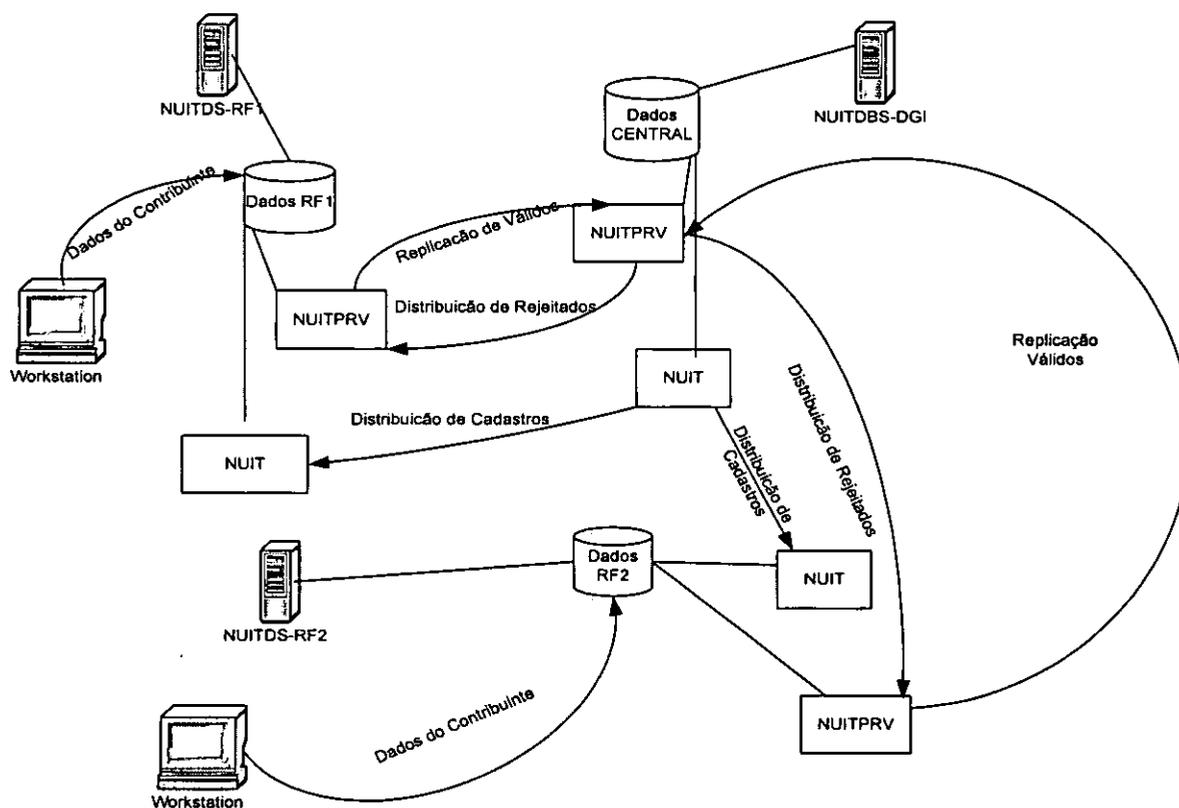


Figura 7: Arquitectura da Distribuição de Dados do NUIT

A figura 7 dá a visão do modelo proposto para arquitectura do processo de distribuição de dados.

Este processo envolve os seguintes componentes:

NUITDGS - DGI é o servidor da BD do aplicativo NUIT na CENTRAL ;

NUITDGS - RF1 é o servidor da BD do aplicativo NUIT na RF1;

NUITDGS - RF2 é o servidor da BD do aplicativo NUIT na RF2;

Uma ou mais estações de trabalho "Workstations" em cada repartição que são as *front-end*.

Neste caso considerou-se apenas duas repartições (1º e 2º Bairro de Maputo) como exemplo, mas na realidade existem mais repartições que estarão, à semelhança destas, ligadas a Central.

As RF's serão implantadas, localizadas em diversos pontos do país e em comunicação com a DGI Central. O processamento será distribuído, cada RF poderá operar independentemente excepto na geração de novos NUIT's que, é realizado apenas na Central.

No fim de cada operação, a Central contacta as diversas RF's e inicia o processo de sincronização com as mesmas. Após esta operação, os dados são distribuídos de modo que cada RF tenha acesso apenas a seu próprio subconjunto de dados. A Central terá o conjunto completo de dados de todas as repartições.

#### **4.4 Modelo conceptual**

A distribuição de dados de NUIT é reflectido no DFD representado na figura. 8, sob o ponto de vista da replicação/distribuição de dados o qual está dividido em seis fases a seguir numeradas:

1. Entrada e Validação de Dados nas RFs (processos 1, 2 e 3);
2. Busca de Dados nas RFs (processo 4);
3. Validação de Dados e Atribuição de NUITs /Alteração de Cadastros (processos 5,6,7);
4. Retornar requisições inválidas/rejeitadas à RF de origem para correcção (processo 8);
5. Distribuir novos dados e alterações de cadastro para as RFs pertinentes (processo 10);
6. Solicitar correcções ao Contribuinte por carta (processo 9).

Constatou-se, no processo de colecta de dados de Contribuinte, que uma parte significativa destes precisa de ser validado em cada RF. Consequentemente, propõe-se três (3) fases de filtragem, que têm lugar em momentos diferentes, designadamente:

1. No *front-end* da RF, quando da interacção Contribuinte/RF;
2. Na CENTRAL, após a recepção dos dados oriundos das RFs;

3. Na CENTRAL, durante a Atribuição do NUIT/Alteração de Cadastros, para casos que eventualmente não tenham sido convenientemente depurados nas duas fases anteriores.

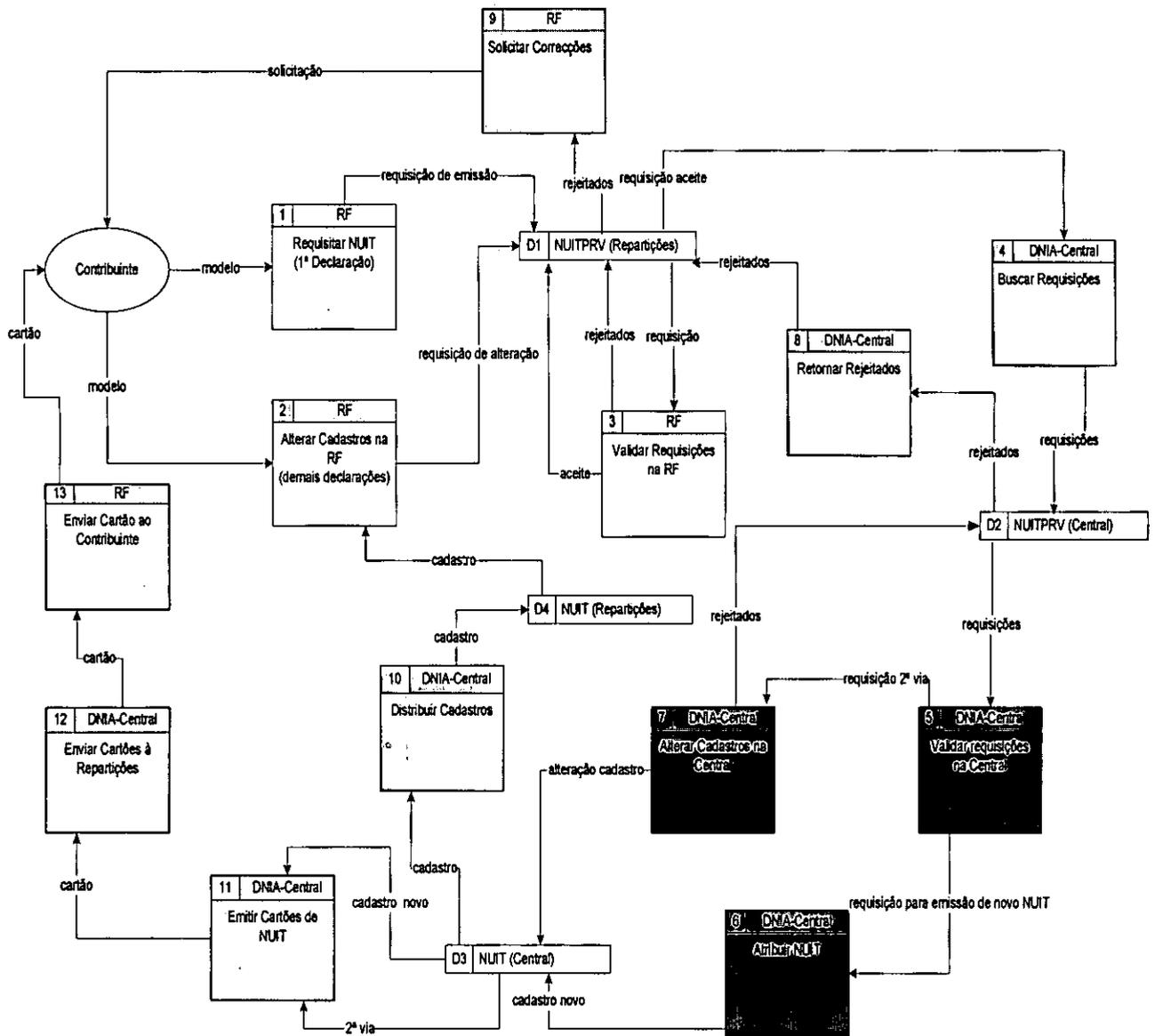


Figura 8: Diagrama de fluxo de Dados

Cada uma das fases de filtragem acima mencionadas são descritas a seguir com o auxílio de diagrama de transição de estados, em UML.

#### 4.4.1 Front-End da RF

Após a inserção de dados providos pelos modelos apropriados e o respectivo "commit" na BD NUITPRV pelo digitador da RF, um relatório contendo a informação inserida, será entregue ao Contribuinte, se este ainda estiver presente na RF, para efeitos de confrontação. Se forem detectadas omissões então a correcção far-se-á de imediato na presença do Contribuinte.

De salientar que, este relatório será alimentado por um processo de validação mínimo (vide processo 3 do DFD), cujo propósito é eliminar algum tipo de irregularidade, por exemplo, informação que envolva algum Contribuinte registado na RF, como se ilustra na figura 9

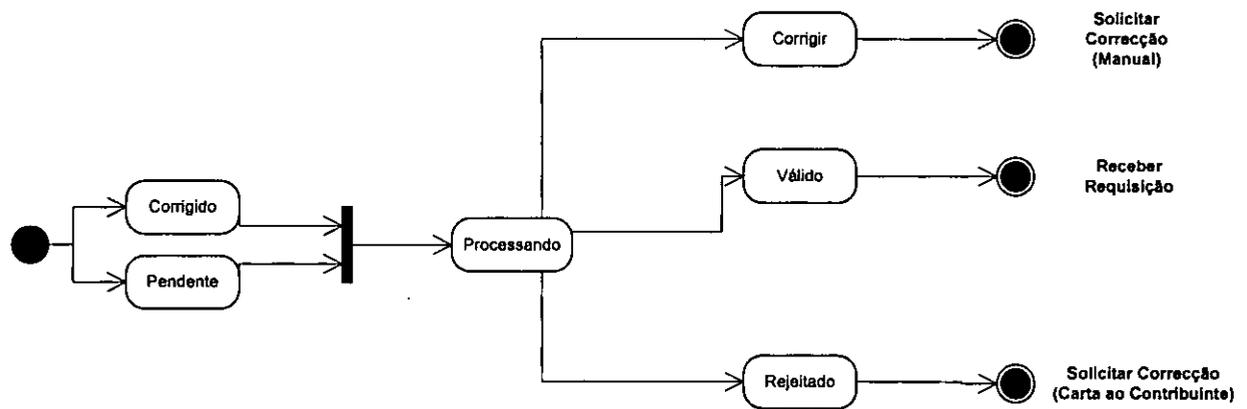


Figura 9: Validação na RF

Como se percebe da figura 9, a solicitação de correcções é um processo manual/físico cujos detalhes são mostrados na figura 10.

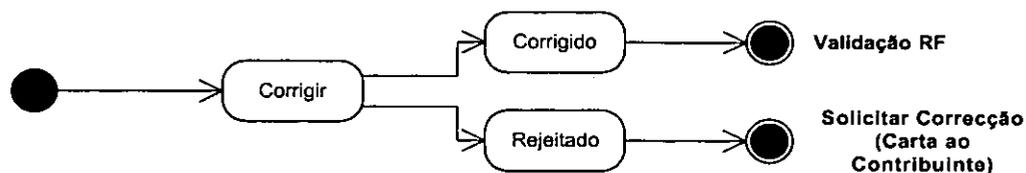


Figura 10: Correção Manual na RF

O processo de correcção na RF admite apenas dois cenários:

1. Os dados são depurados e corrigidos e quando ficam prontos recebem o *status* CORRIGIDO estando aptos para reiniciar a validação na RF;
2. Os dados que são depurados e rejeitados, recebem o *status* REJEITADO; Não são reconduzidos para validação na RF e desencadeiam um processo de notificação do Contribuinte pela RF com origem nesta.

Estão envolvidos nesta fase os processos 1 e 2, como inputs, o 3, 4 e 9 como *outputs*.

#### 4.4.2 Recepção de Dados na CENTRAL

O processo 4 descrito no DFD, inicia o evento de recepção de dados na CENTRAL por replicação dos dados das RF. Como se mostra na figura 11:

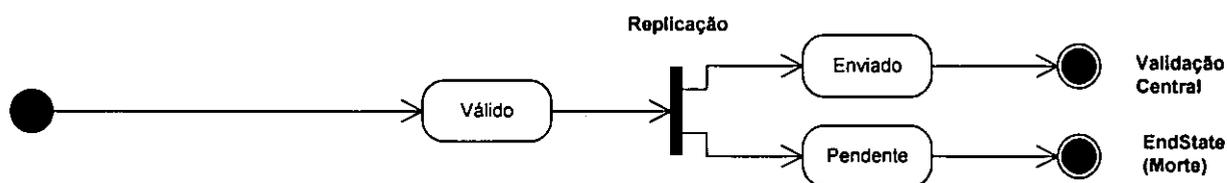
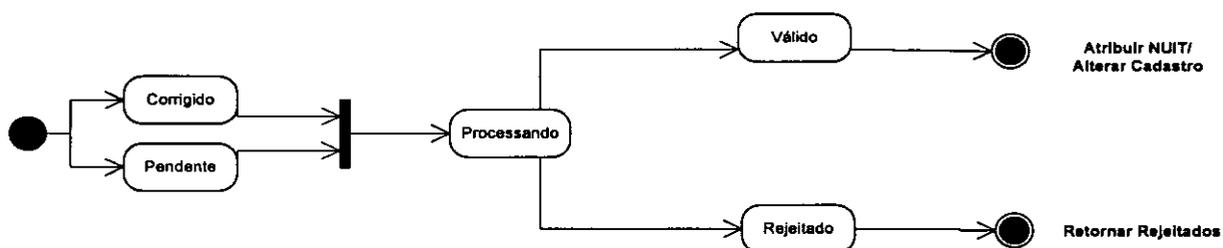


Figura 11: Replicação de dados RF ->CENTRAL

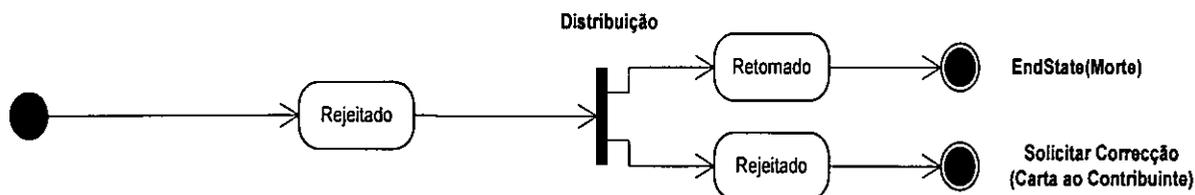
O processo de replicação para a CENTRAL é sempre iniciado com o *status* VÁLIDO e caracteriza-se pela bifurcação de estados em ENVIADO para os dados validados e replicados com êxito. E, em PENDENTE para aqueles que por algum motivo imprevisto não tenham sido replicados com êxito, ficando portanto confinados à RF.



Figurra 12: Validação na CENTRAL

Na CENTRAL, após a replicação, realiza-se a segunda fase de filtragem dos dados de Contribuinte, que é na essência, similar ao processo seguido na RF, já descrito em 4.3.1, tal como se observa na figuraura 12 com excepção da transição para o *status* VÁLIDO que, neste caso, origina os processos *batch* de Atribuição do NUIT/Alteração de Cadastro.

A Distribuição de Dados rejeitados a partir da CENTRAL é descrito na figuraura 13:



Figuraura 13: Distribuição de rejeitados CENTRAL ->RF após a validação

Ela consiste no processo revertido da Replicação de Dados RF-> CENTRAL descrito na figura 9.

#### 4.4.3 Atribuição de NUIT/Alteração de Cadastros na CENTRAL

Este estágio começa com a última fase da filtragem dos dados de Contribuinte, que porventura ainda possuam irregularidades relevantes. A sua mecânica de tratamento de dados rejeitados é similar a descrição da figura 10, como se ilustra na figuraura 14:

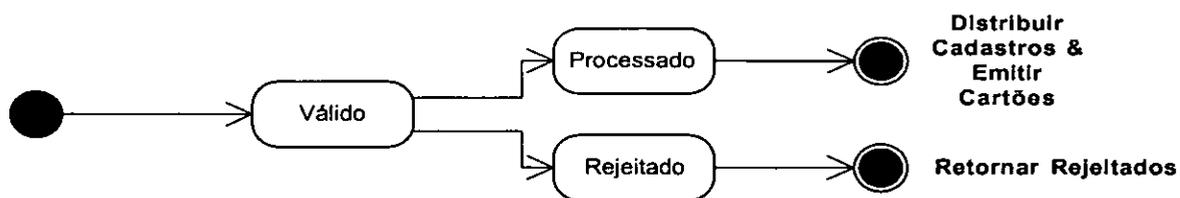


Figura 14: Atribuir NUIT/ Alterar Cadastros na CENTRAL

Não se realiza nenhuma correcção para os dados provenientes da validação na CENTRAL e que entram com o *status* VÁLIDO. Admite-se aqui, duas situações:

1. Os dados são depurados e corrigidos e quando ficam prontos recebem o *status* PROCESSADO estando aptos para serem distribuídos para as RF ou proceder-se à emissão dos respectivos cartões de Contribuinte;

A distribuição de dados partindo da CENTRAL após a fase de Atribuição do NUIT/Alteração de Cadastro é unidireccional e descreve-se na figura 15:

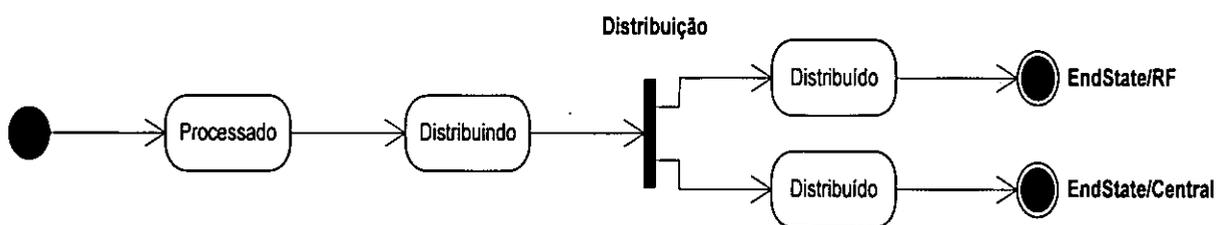


Figura 15: Distribuição de dados de NUIT/Alteração de Cadastros CENTRAL->RFs

A interpretação dada aos estados finais EndState/RF e EndState/CENTRAL é a seguinte. Tratando-se de um fluxo unidireccional CENTRAL -> RF não existe nenhuma mensagem de confirmação de recepção pelas RF. Consequentemente, estabelece-se uma *flag* na CENTRAL, descrito em documento à parte que controlará as distribuições de dados recebidas ou não pelas RFs

- Os dados que são depurados e rejeitados, recebem o *status* REJEITADO, não são reconduzidos para validação na CENTRAL, desencadeiam um processo de notificação do Contribuinte pela RF com origem na CENTRAL.

A Distribuição de Dados rejeitados da CENTRAL após a fase de Atribuição do NUIT/Alteração de Cadastro é logicamente descrita na figura 16.

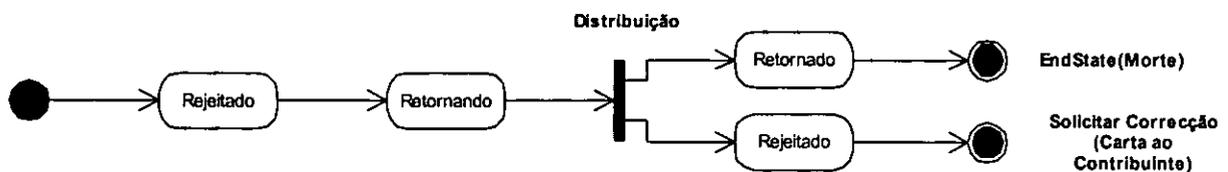


Figura 16: Distribuição de rejeitados CENTRAL ->RF após a Atribuição/Alter. do NUIT

Esta figura encerra unicamente o fundamento lógico, dado a informação dos cadastros a distribuir se fazer directamente das tabelas definitivas do NUIT, que não têm a necessidade de reter a informação sobre os diversos *status* de processamento.

#### 4.5 Descrição das actividades do processo de replicação

O processo de replicação e distribuição de dados é um processo *batch* ( em lotes) disparado a partir da Central em Maputo e executado na Central. Desse modo o controle de todo o processo para todas as RF's encontra-se sempre numa única localização, permitindo que um único recurso obtenha informações e influencie todo o processo e sem necessidade de intervenção humana nas RF's para que o processo se complete.

Esta descrição envolve os processos 4, 8 e 10 do DFD mostrado na figura 8.

##### 4.5.1 Buscar Requisições (Processo 4)

Este processo contacta cada Repartição numa ordem pré definida, e para cada Repartição executa as seguintes tarefas:

1. Identifica todas as **Requisições** daquela Repartição cujo status seja válido;
2. Para todas as **Requisições** assim identificadas, actualiza o status para **Enviando**;
3. Copia as **Requisições**, em lotes ou individualmente, para a **Central**, onde são deixados com status **Pendente** em preparação para processamento de **Validar Requisições na Central**;
4. A cada cópia bem sucedida de uma **Requisição**, o status dessa **Requisição** na RF é actualizada para **Enviado**, completando o ciclo de vida dessa **Requisição** na RF.

A partir desse momento, a cópia dessa **Requisição** na Central continua sendo processada até que essa seja **Rejeitada** ou transformada num **Cadastro** pelos processos de **Validar Requisições na Central e Atribuir NUIT /Alterar Cadastros na Central**, enquanto a cópia na Repartição Fiscal é mantida apenas por questões de auditorias por um curto período de tempo, mas, para fins do processo de replicação, chega ao seu estado final.

#### **4.5.2 Retornar Rejeitados (Processo 8)**

O processo contacta cada RF numa ordem pré definida, e para cada uma executa as seguintes tarefas (algoritmo):

1. Identifica todas as **Requisições** daquela Repartição na Central cujo status seja **Rejeitado**;
2. Para todas as **Requisições** assim identificadas, actualiza o status para **Enviando**;
3. Copia as **Requisições**, em lotes ou individualmente, para a Repartição, onde são deixados com status **Rejeitado** em preparação para processamento de **Solicitar Correção**;
4. A cada cópia bem sucedida de uma **Requisição**, o status dessa **Requisição** na Repartição Fiscal é actualizado para **Enviado**, completando o ciclo de vida dessa **Requisição** na Central.

A partir desse momento, a cópia dessa **Requisição** na Repartição Fiscal é processada em **Solicitar Correção**, porém sem novas mudanças de estado. Para todos os fins, o ciclo de vida de uma **Requisição Rejeitada** acaba, na Central, com a replicação e, na Repartição Fiscal, com a solicitação de correção.

### **4.5.3 Distribuir Cadastros (Processo 10)**

O processo consiste de duas etapas, executadas sequencialmente:

1. Transmissão de todos os **Cadastros** criados desde a última execução;
2. Transmissão de todas as **Cadastros** alterados desde a última execução.

O processo contacta cada Repartição em uma ordem pré definida, e para cada Repartição executa as seguintes tarefas:

1. Recupera a data e hora do início da última execução para essa RF e armazena internamente;
2. Recupera a data e hora actual e armazena com a data e hora de início da execução corrente;
3. Registra o início da execução actual para essa RF;
4. Identifica todos os **Cadastros** criados na Central entre a data e hora de início da última execução do processo para aquela Repartição e a data e hora de início da execução corrente;
5. Copia cada **Cadastro**, em lotes ou individualmente, para a RF, onde estarão disponíveis para consultas ou alterações;
6. Identifica todos os **Cadastros** alterados na Central entre a data e hora de início da última execução do processo para aquela RF e a data e hora de início da execução corrente e que não tenham sido criados após a data e hora de início da última execução;
7. Actualiza cada **Cadastro**, em lotes ou individualmente, na RF, onde estarão disponíveis para consultas ou alterações;
8. Regista o final da execução para essa RF.

A partir desse momento, o **Cadastro** deixa de fazer parte do processo de distribuição de dados e passa a fazer parte da BD regular do sistema NUIT, sendo recapturado para distribuição apenas quando uma nova requisição de alteração for iniciada pela RF.

#### **4.5.4 Temporização/Fluxo**

Todos os procedimentos de distribuição descritos acima devem ser executados numa ordem específica para que o processo como um todo seja o mais eficiente possível. A execução numa ordem qualquer, aleatória não causaria nenhum dano ao sistema excepto na forma de atrasos no processamento de requisições de emissão de cartões e atribuição/correção dos cadastros NUIT.

O diagrama da figura. 17 descreve a sequência proposta para se conseguir o processamento mais efectivo das requisições e os processamentos na Central, visto que todos esses processos serão executados sobre os mesmos canais de comunicações entre Repartições Fiscais e Central e no mesmo servidor na Central.

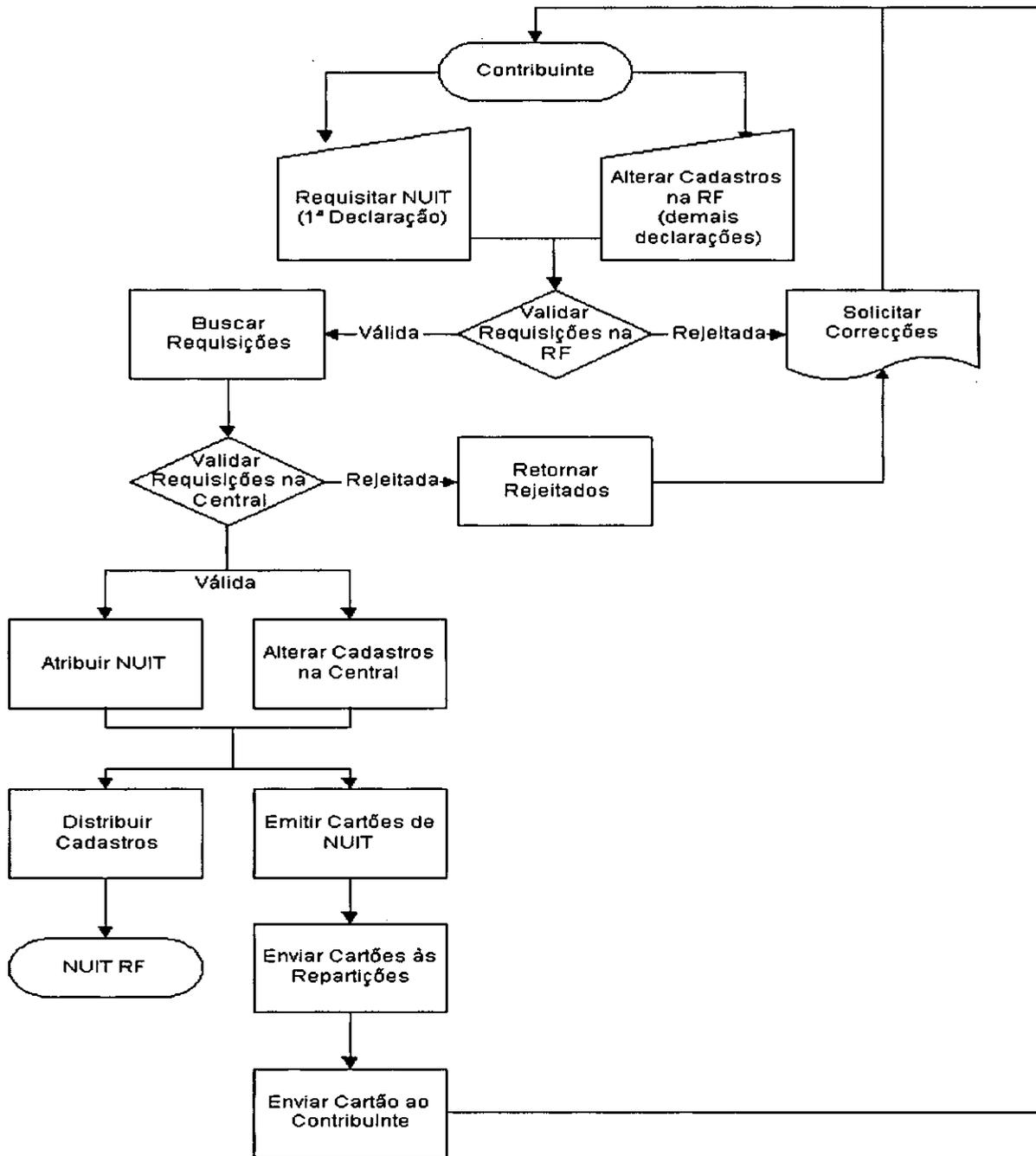


Figura 17: Fluxo/Temporização

## 4.6 Mecanismos de Segurança

O controlo e a segurança dos dados do NUIT será feito mediante a definição de “*profiles*” apropriados. Cada utilizador terá um *profile* associado e a instância *Oracle* deverá ser configurada de acordo com os requisitos de cada *profile*.

A BD do NUIT deverá possuir o *profile* *PASSWORDS\_NUIT* para a gestão das *passwords* de utilizador. Mas, poderão ser criados mais *profiles* sempre que houver necessidade e todos serão regulados e criados pelo Administrador de BD (DBA).

### 4.6.1 *Profile* *PASSWORDS\_NUIT*

Este *profile* deverá respeitar os seguintes parâmetros:

**Manutenção do histórico de *passwords*:** Número de dias que é necessário passar até que uma *password* possa ser novamente reutilizada e o número de vezes que uma *password* pode ser reutilizada.

**Prazo de validade das *passwords*:** Número de dias de validade da *password* e o número de dias de tolerância para o utilizador modificar a sua *password* após esta ter expirado. Findo o período de tolerância, apenas o DBA poderá alterar a *password*.

### 4.6.2 Regras de verificação das *passwords*

O DBA deverá activar um *script* (tipo de programa que não necessita ser convertido em linguagem de máquina para ser executado), cujo template se encontra no directório dos *scripts* administrativos do Sistema Relacional de Gestão de BD (RDBMS) *Oracle*, garantindo:

- Uma *password* não nula;
- Uma *password* diferente do nome do utilizador;
- Comprimento da *password* não inferior a oito caracteres;
- A *Password* deve conter pelo menos um caracter alfanumérico;
- Na alteração da *password*, a nova deve diferir da antiga em pelo menos três caracteres

### 4.6.3 Bloqueio de Utilizadores

O DBA deverá configurar o sistema com os parâmetros:

- Número máximo de tentativas que um utilizador dispõe para introduzir a sua *password*;
- Número máximo de dias que o utilizador ficará bloqueado até a *password* expirar. Findo este prazo de validade, só o DBA poderá desbloqueá-la.

## 4.7 Backup e Recuperação da BD

Em todo o sistema de BD sempre existe a possibilidade de uma falha de sistema ou de *hardware*. Caso ocorra algo que afecte a BD ele deve ser recuperado. Os objectivos esperados depois de uma falha devem garantir que os efeitos de todas as transacções submetidas se reflectam naquela BD que será recuperada e o retorno seja o mais rápido possível à operação normal, isolando os usuários dos problemas que foram causados (Ramalho, 2002).

*Cold backup archiving* é uma das várias estratégias de backup e recuperação da BD disponíveis para uma BD *Oracle* e será esta a estratégia a adoptar no presente trabalho por ser mais simples.

Para definir esta estratégia, começa-se por definir a estratégia *Cold backup* pois são idênticas.

*Cold backup* - É o método mais simples e directo que consiste em retirar a BD do ar (*shutdown normal*) e copiar todos os ficheiros da BD para outra máquina e (ou) suporte, de onde podem ser recuperados em caso de perda da BD; porém não permite recuperação completa das informações até o momento da perda do serviço.

*Cold backup archiving* - Consiste em realizar *Cold backup* com o recurso de *archiving* (cópia online) dos *redo log files* da instância habilitado, ficheiros esses utilizados pelo *Oracle* para refazer todas as operações realizadas na BD até o último momento em que a BD funcionou antes do desastre que tenha exigido uma recuperação.

Possui as mesmas vantagens de simplicidade do *Cold backup*, mas permite recuperação completa das informações até ao momento da perda. Porém exige que a BD esteja temporariamente indisponível para as cópias de segurança.

#### 4.7.1 Procedimentos de backup (Cold backup archiving)

Na consola do servidor, para cada instância de BD a ser salva, abrir um comando no MS Windows (Start → Run; cmd) e lá executar, em uma única linha, o comando seguinte:

```
ORADIM -shutdown -sid <SID>
```

Onde: <SID> é o nome da BD

Após a execução do comando, abrir o utilitário de controlo de serviços do Windows (Start → Settings → Control Panel → Administrative Tools → Services) e parar todos os serviços *Oracle*, aqueles cujos nomes começam com "Oracle\*".

Executar cópias de segurança pelo sistema operativo em fita, tendo-se certeza de copiar *todos* os ficheiros nas directorias \oracle\oradata\<SID> existentes em *todos* os volumes lógicos do servidor.

Separar a fita para ser guardada a parte durante o período de retenção especificado. Criar uma etiqueta e fazer cadastro da fita no mecanismo de controle de cópias de segurança implementado.

No fim da cópia de segurança, voltar ao utilitário de serviços do MS Windows (Start → Settings → Control Panel → Administrative Tools → Services) e reinicializar todos os serviços Oracle parados anteriormente.

Tem que se garantir que os *archive redo log files* sejam periodicamente (por exemplo a cada hora) copiados para um sítio independente e seguro e mantidos em fita, em paralelo com as cópias de segurança diárias.

Os *archive redo log files* estão armazenados num directório específico para cada instância. Esse directório, não se altera frequentemente, portanto não exige a intervenção do DBA. Cabe a este, identificar a restante equipa de operadores qual a directoria para cada instância.

Entretanto, a equipa de operadores com acesso à BD com direitos de DBA ou SELECT ANY TABLE e CONNECT, pode facilmente descobrir qual é esse directório através do utilitário do *Oracle Enterprise Manager*, seguindo as instruções a seguir:

Start → Programs → Oracle-OraHome92 → Enterprise Manager Console → Launch Standalone;

Seleccionar a BD desejada;

Digitar o Username e a senha;

Seguir a estrutura em árvore: Network → Database → <SID> → Instance → Configuration → Recovery;

Recuperar o nome do directório com os *archive redo log files* na tabela “*Archive Log Destination(s)*” cujo o status seja “*VALID*”.

#### **4.7.2 Procedimentos de Recuperação de Dados**

Começa-se exactamente com o mesmo procedimento aplicado no caso de *Cold backup archiving*, e aplica-se o conteúdo dos *archive redo log files*.

Uma vez restaurados os arquivos de dados e de configuração, acede-se à BD usando o *SQL\*Plus* com uma senha possuidora de privilégios SYSDBA e fazer o comando:

```
SELECT archive_name FROM v$recovery_log;
```

Copiam-se os ficheiros especificados pelo resultado da consulta acima para o caminho indicado pelo conteúdo da variável LOG\_ARCHIVE\_DEST\_1.

No caso de falta de espaço em disco para os ficheiros, escolhe-se outra directoria indicada a seguir pelo parâmetro LOGSOURCE.

Então procede-se a sua aplicação à BD:

```
RECOVER FROM <caminho>;
```

Por fim acede-se ao caminho contendo os ficheiros, acima especificados, e:

DEL \*.ARC

## **CAPÍTULO V: CONCLUSÕES E RECOMENDAÇÕES**

### **5.1. CONCLUSÕES**

A proposta apresentada é uma opção adequada para a DGI pois minimiza o problema de inconsistência dos dados através das validações que serão efectuadas tanto na Central como nas RF's.

Esta abordagem trará para aplicação NUIT, entre vários benefícios os seguintes:

- Bons índices de disponibilidade pois tendo cada RF a sua BD local, poderá acedê-la a qualquer momento para qualquer operação que pretenda executar;
- Bom desempenho da BD pois a fácil localização dos dados reduz a demora pelo acesso remoto aos dados;
- Redução de tráfego na rede.

Também, garante que cada RF possa trabalhar de forma autónoma evitando-se a necessidade de ter uma ligação permanente com a Central. Esta autonomia permite não só a facilidade de gestão de acesso aos dados mas também no caso de falha, por exemplo, de comunicações as RF's podem continuar o seu funcionamento.

Facilita o dimensionamento à medida que o sistema cresce. Pode-se adicionar diversos servidores para distribuir a carga de processamento da base de dados em toda rede. Como maior parte das RF's ainda não estão incorporadas, facilitará o incremento das mesmas à medida que for necessário.

## **5.2. RECOMENDAÇÕES**

Introduzir mais um interveniente na arquitectura cliente/servidor, nomeado servidor de aplicações, que se posiciona entre o cliente e os servidores de base dados com o objectivo de suportar o elemento “lógica das aplicações” o qual constituirá o nível intermédio entre a interface com os utilizadores e a base de dados, de maneira a facilitar a incorporação, futura, de mais máquinas clientes sem degradar o desempenho da base de dados .

Para minimizar as despesas em papel na emissão de cartas aos contribuintes para confirmação dos dados introduzidos e personalizar o atendimento a estes, aconselha-se a colocação de um terminal para impressão, no ecrã, da respectiva carta onde o contribuinte poderá confirmar os seus dados e posterior correcção dos mesmos em caso de erros.

## CAPÍTULO VI: BIBLIOGRAFIA

### 6.1. Referência Bibliográfica

Andrews, G. R. (1991). *Concurrent Programming: Principles and Practice*. CA: Benjamin Cummings.

Bell, D., & Grimson, J. (1992). *Distributed Database Systems*. Reading, Mass: Addison Wesley.

Bright, M. W. (1992). *Automated Resolution of Semantic Heterogeneity in Multi-Databases*. New York: McGraw-Hill.

Comer, D. E. & Stevens, David L. (1993). *Internetworking With TCP/IP Client/Server Programming and Application (Socket Version)*. New Jersey: Prentice Hall.

Comer, D. E. (1998). *Inerligação em Redes com TCP/IP*. Rio de Janeiro: Campus.

Cohen, W. W. (1998). *Integration of Heterogeneous Databases without Common Domains*. New York: North-Holland.

Date, C. J. (1991). *Introdução a Sistemas de Base de Dados*. Rio de Janeiro: Campus.

Date, C. J. (1995). *An Introduction to Database System*. New York: Addison Wesley.

Davis, R. (1994). *Windows NT Networking Programming*. New York: Addison Wesley,

Hulquist, S. (1997). "FAQ about Client/Server". Recuperado em 28/07/2005 de

<http://non.com/news.answers/client-server-faq.html>.

Kalakota, R. (1997). *FAQ about Client/Server*. Recuperado em 28/07/2005 de

<http://non.com/news.answers/client-server-faq.html>

Kaorth, H.F. & Silberschatz, A. (1994). *Sistema de Base de Dados*. São Paulo: Makron Books.

Marques, J.A. & Guedes, P. (1999), "*Tecnologia de Sistemas Distribuídos*". FCA. Lisboa

Mckie, S. (1997). "*Everything you wanted to know about Client/Server Computing but were afraid to ask*". Recuperado em 16/07/2005 de <http://www.duke.com/controller/Issues/ecJan/Cliente.htm>.

Morelli, E. M. T. (2002). *Oracle9i Fundamental: SQL, PL/SQL e Administração*. São Paulo: Érica.

Pereira, J. L. (1998). *Tecnologia de Base de Dados*. Lisboa: FCA – Editora de Informática.

Ramalho, J. A. (2002). *Oracle 9i: Aprenda a criar bancos de dados, tabelas, índices, visões e outros objectos*. São Paulo: Berkeley.

Renaud, P. E. (1995). *Introdução aos Sistemas C/S*. Rio de Janeiro: IBPI Press.

Salemi, J. (1993). *BD C/S*. Rio de Janeiro: IBPI Press

Soares, L. F. G. (1997). *Redes de Computadores: das LANS, MANS e WANS às redes ATM*. Rio de Janeiro: Campus.

Soares, L.F.G., Gomes, L., Colcher, S. (1995). *Redes de Computadores: das LANS, MANS e WANS às redes ATM*. Rio de Janeiro: Campus.

Tárouco, L.M.R. (1986). *Redes de Computadores*. São Paulo: McGraw-Hill

Tanenbaum, A. (1997). *Computer Networks*. London: Prentice-Hall.

Turban, E. (1999). *Information Technology for Management*. New York: Wiley.

## **6.2. Bibliografia Consultada**

Adkoil, A. & Velpuri, R.(2002). *Oracle9i for Windows Handbook*. Clifornia: McGraw-Hill.

Berson, A. (1996). *Client/Server Architecture*. Rio de Janeiro: McGraw-Hill.

Castano, S., F. Miagrazia, Mortella, G. & Samarati, P. (1995). *Database Security*. New York: Addison Wesley.

Ceri, S. & Pelagatti, G. (1985). *Distributed Databases – Principles & Systems*. California: McGraw-Hill.

Comer, D. (1998). *Interligação em Rede com TCP/IP – Princípios, Protocolos e Arquitectura*. Rio de Janeiro: Campus.

Decreto nº 52/2003 de 24 de Dezembro do Boletim da República I Série – Número 52 : Maputo.

Elmesri, R. & Navathe, S.B. (1994). *Fundamentals of Database systems*. CA: Benjamin Cummings.

Fantinatt, J. M. (1988). *Segurança em Informática – Metodologia e Prática*. São Paulo: McGraw-Hill.

Gomes, O. J. A. (200). *Segurança Total – Protegendo-se contra os hackers*. São Paulo: Makron Books.

Greenwald, R. & Kreines, D. C. (2003). *Oracle in a NUTSHELL*. USA: O’Reilly.

Gutta, R. (2002). *Oracle DBA Scripts de Automação*. Rio de Janeiro: Campus.

Macome, E. (1995). “*Introdução a Metodologia de Investigação*”. Maputo: Autor.

Ozsu, M. T. & Valdureiz, P.(1991). *Principles of Distributed Database Systems*. New Jersey: Prentice Hall

Stevens, W. P. (1999). *TCP/IP Illustrated*. Califórnia: Addison Wesley .

Verissimo, P. & Rodrigues, L. (2001). *Distributed Systems Architecture*. Norwell: Kluwer.