



Centro de Competências em Ciências Exactas e da Engenharia

Virtual Network Description: Suporte à simulação de redes através da autoria gráfica de cenários virtuais

Jesuíno da Costa Serra de Azevedo

*Tese submetida à Universidade da Madeira para a
Obtenção do Grau de Mestre em Engenharia Informática*

Funchal – Portugal

Novembro 2010

Os melhores RUMOS para os Cidadãos da Região



REGIÃO AUTÓNOMA DA
MADEIRA



REPÚBLICA PORTUGUESA



UNIÃO EUROPEIA
FSE

ORIENTADOR

Professor Doutor Paulo Nazareno Maia Sampaio

Professor Auxiliar do Centro de Competências de Ciências Exactas e Engenharias

ABSTRACT

By the utilization of network management tools it is possible to obtain a deeper knowledge of how a given network operates, as such allowing its optimization. Currently, there are several networks tools available for this aim, such as monitoring tools, topology building tools, trouble tickets/help desk, and network simulation and analysis tools. Definitely, an optimized management of the network can be provided by the appropriate integration of different tools. Nevertheless, most of these tools are not interoperable (having proprietary code) or are available commercially as closed software packages with a high cost.

In this context, there is the need for a common abstraction among different network management tools which allows their communication. In this case, the utilization of a language such as NSDL (Network Description Scenario Language) can be useful. NSDL has been proposed at University of Madeira by the PhD student Eduardo Marques as a common language for the description of network scenarios which allows the interoperability among different network management tools.

The main goal of this work is the proposal and the development of a graphical interface for the authoring of network scenarios allowing the automatic generation of NSDL, which can be further applied for the simulation and analysis of these scenarios. One of the main advantages of this tool, beyond its simple and intuitive interface, is the flexibility for customizing objects and their parameters which can be manipulated during the creation of network scenarios.

KEYWORDS

Network Management

Network Scenario Description Language

Authoring of network scenarios

Network Simulation

Quality of Service

XML

Através da utilização de ferramentas de gestão de redes é possível obter um melhor conhecimento do funcionamento de uma rede, permitindo assim a sua optimização. Diferentes ferramentas existentes podem ser utilizadas para essa finalidade, tais como ferramentas de monitorização, construção de topologias, trouble tickets/help desk, simulação e análise de redes. Uma gestão optimizada da rede pode ser obtida através da integração de diferentes ferramentas. No entanto, a maior parte dessas ferramentas não são interoperáveis (possuem código proprietário) ou então são disponibilizadas como pacotes de software fechados e com um alto custo.

Nesse contexto, surge a necessidade da existência de uma abstracção comum para diferentes ferramentas de gestão que permite que essas comuniquem entre si. Dai a utilização de uma Framework baseada na Linguagem NSDL (*Network Description Scenario Language*). NSDL foi proposta na Universidade da Madeira pelo doutorando Eduardo Marques como uma linguagem comum para a descrição de cenários de redes que permita a interoperabilidade de diferentes ferramentas de gestão.

O principal objectivo deste trabalho é a proposta e o desenvolvimento de uma ferramenta gráfica para a autoria de cenários de redes que gere automaticamente código NSDL, e que possa ser posteriormente utilizado para a simulação e análise de redes. Uma das principais vantagens dessa ferramenta, além sua simplicidade, é a sua flexibilidade na customização de objectos e parâmetros que podem ser manipulados durante a criação de cenários de redes.

PALAVRAS-CHAVE

Gestão de redes

Network Scenario Description Language

Autoria de cenários de Redes

Simulação de Redes

Qualidade de Serviço

XML

AGRADECIMENTOS

Ao professor Paulo Sampaio que mais do que um orientador, foi um amigo, que tenho consciência que merece muito dos créditos do trabalho aqui desenvolvido, muitas vezes me guiando e ajudando muito para além do que o que lhe era pedido, a ele um sincero obrigado.

Ao professor Eduardo Marques que sempre me ajudou em tudo o que lhe era possível, fornecendo muita da informação que serviu de base a este documento.

À Asia que sem compreender Português ou esta área, sempre me deu força, fazendo-me acreditar que conseguia.

DEDICATÓRIA

Dedico este trabalho ao meu pai, que presenciou o início e mais este percurso, mas que infelizmente não tem a possibilidade de estar comigo neste momento que seria de muita alegria também para ele.

À minha mãe por todo o carinho e compreensão.

À Asia que abdicou de muito do nosso tempo junto para me encorajar a trabalhar um pouco mais.

À minha família

1. INTRODUÇÃO	1
1.1. MOTIVAÇÃO	1
1.2. PRINCIPAIS CONTRIBUIÇÕES	2
1.3. ORGANIZAÇÃO DO DOCUMENTO	3
2. ESTADO DA ARTE	4
2.1. INTRODUÇÃO	4
2.2. ABORDAGEM CONCEPTUAL	4
2.2.1 AUTORIA DE CENÁRIOS DE REDE	4
2.2.2 SIMULAÇÃO DE REDES	5
2.2.3 QUALIDADE DE SERVIÇO	7
2.2.4 XML	11
2.3. ABORDAGEM A LITERATURA	14
2.4. CONCLUSÃO	17
3. NSDL	18
3.1. INTRODUÇÃO	18
3.2. CONCEITOS BÁSICOS	18
3.3. ARQUITECTURA	20
3.4. ESTRUTURA NSDL	22
▪ REDE	24
▪ CENÁRIOS	26
3.5. CONCLUSÃO	27
4. VND: A INTERFACE GRÁFICA PARA AUTORIA DE CENÁRIOS DE REDES	28
4.1. INTRODUÇÃO	28
4.2. REQUISITOS	28
REQUISITOS FUNCIONAIS	30
REQUISITOS NÃO FUNCIONAIS	31
OUTROS REQUISITOS	32
4.3. CASOS DE UTILIZAÇÃO	32
4.4. DIAGRAMAS DE ACTIVIDADE	34
4.5. ARQUITECTURA DO SISTEMA	37
4.6. PROTÓTIPOS ABSTRACTOS CANÓNICOS	39

4.7. OPÇÕES TECNOLÓGICAS	40
4.8. ASPECTOS DE CONFIGURAÇÃO	41
4.9. APRESENTAÇÃO DO SISTEMA	44
4.10. CONCLUSÃO	48
5. ESTUDO DE CASO	49
5.1. INTRODUÇÃO	49
5.2. DESCRIÇÃO DA ESTRUTURA BASE	49
5.3. PRIMEIRO ESTUDO DE CASO	53
5.4. SEGUNDO ESTUDO DE CASO	58
5.5. CONCLUSÃO	62
6. CONCLUSÕES E PERSPECTIVAS FUTURAS	63
6.1. CONCLUSÕES	63
6.2. O QUE PROPUSEMOS FAZER	63
6.3. PERSPECTIVAS FUTURAS	66
REFERÊNCIAS	67
ANEXOS	70

ÍNDICE DE IMAGENS

Figura 1 -Comparativo das ferramentas de autoria de redes.....	16
Figura 2 Arquitectura NSDL. [Marques, 2010].....	21
Figura 3 Os elementos do NSDL [Marques, 2010]	23
Figura 4 Representação do componente Network	25
Figura 5 Representação do componente escenarios.....	27
Figura 6 Casos de Utilização	33
Figura 7 Diagrama de actividade Configurar Objecto	35
Figura 8 Diagrama de actividade Criar Rede	37
Figura 9 Arquitectura básica do VND	37
Figura 10 Arquitectura por camadas do VND	38
Figura 11 Módulos e Interacção	38
Figura 12 Protótipo Abstracto Canónico do VND	40
Figura 13 Estrutura do XML de configuração do VND	43
Figura 14 Formato do XML de declaração de objectos	43
Figura 15 Aspecto gráfico genérico do VND	44
Figura 16 Exemplo de navegação na biblioteca	45
Figura 17 Exemplo de lista de objectos já criados	46
Figura 18 Área de configuração de objecto	46
Figura 19 Adicionar configurações especiais	47
Figura 20 Arrastar objectos e criar topologia	50
Figura 21 Rede de base para os casos de estudo	51
Figura 22 Edição do de um computador	52
Figura 23 Alterar configuração fibra.....	53
Figura 24 Editar configuração Ethernet	54
Figura 25 Adicionar configurações especiais	56
Figura 26 Alterar dados de uma aplicação	56
Figura 27 Ligação exportada	57

1. INTRODUÇÃO

Emuladores e simuladores de rede actualmente tornaram-se ferramentas importantes na gestão de redes e investigação de redes experimentais devido a vários motivos, incluindo os custos elevados de utilizar soluções em redes reais para realizar tais experimentos.

Através da utilização de ferramentas de gestão de redes é possível obter um melhor conhecimento do funcionamento de uma rede, permitindo assim a sua optimização. Diferentes ferramentas existentes podem ser utilizadas para essa finalidade, tais como ferramentas de monitorização, construção de topologias, trouble tickets/help desk, simulação e análise de redes.

O desenvolvimento de uma ferramenta para a gestão de redes, exige diferentes competências e conhecimentos, tais como as redes informáticas e os seus componentes e tecnologias, a usabilidade e a programação.

1.1. MOTIVAÇÃO

Como discutimos anteriormente, existem muitas ferramentas para a gestão de redes. No entanto, a maior parte dessas ferramentas não são interoperáveis (possuem código proprietário) ou então são disponibilizadas como pacotes de software fechados e com um alto custo.

Uma gestão optimizada da rede pode ser obtida através da integração de diferentes ferramentas. Portanto, a principal necessidade observada durante o estudo realizado foi da existência de uma abstracção comum para diferentes ferramentas de gestão que permite que essas comuniquem entre si.

O trabalho apresentado em [Marques & Sampaio, 2010] introduz uma *framework* para a integração de diferentes ferramentas de gestão de redes baseada em uma linguagem chamada *NSDL (Network Description Scenario Language)*. NSDL proporciona a descrição de cenários de rede que podem ter sido construídos automaticamente por uma ferramenta de monitorização e construção de topologias de redes, ou então modelizados através de uma

ferramenta de autoria gráfica para cenários de redes. Esses cenários podem ser posteriormente simulados por diferentes simuladores de rede compatíveis, possibilitando a análise comparativa da sua simulação. Portanto, NSDL é uma solução transversal a diferentes ferramentas que permite a optimização da gestão de rede.

Os simuladores de rede e emuladores frequentemente requerem que o utilizador tenha que escrever código proprietário para implementar os elementos do modelo utilizado na simulação. Uma vez que os especialistas de rede não são necessariamente especialistas em software, os simuladores de rede e emuladores de arquitecturas de software devem ter uma estrutura sólida e simples que responda às necessidades, mas também ser projectado para aliviar a carga de programação.

A principal motivação deste trabalho foi a necessidade de proporcionar ao gestor de rede uma ferramenta gráfica para a autoria de cenários de redes que gere automaticamente código NSDL, e que possa ser posteriormente mapeado para as linguagens de simulação de redes.

1.2. PRINCIPAIS CONTRIBUIÇÕES

As principais contribuições desta dissertação de mestrado são:

- O estudo conceitual na área de Redes de Computadores dos principais conceitos envolvidos no desenvolvimento deste trabalho, nomeadamente, da autoria e simulação de redes, Qualidade de Serviço, entre outros;
- O estudo e análise dos trabalhos relacionados existentes na literatura;
- A proposta de uma linguagem de configuração baseada em XML que permita a customização da biblioteca de objectos suportados na autoria de cenários de redes, e;
- A proposta e o desenvolvimento de uma aplicação gráfica (VND) que permita a criação de cenários de redes com suporte à Qualidade de Serviço, que permita a posterior geração de código NSDL, e a sua respectiva simulação.

1.3. ORGANIZAÇÃO DO DOCUMENTO

Este documento está organizado em seis capítulos, incluindo este capítulo como inicial.

No segundo capítulo apresentamos o estado da arte, introduzindo os conceitos que estiveram envolvidos no desenvolvimento do VND. Nesse capítulo é apresentada também uma análise das diferentes ferramentas de autoria de cenários de redes.

No terceiro capítulo apresentamos a linguagem que esteve na origem da necessidade que levou à criação do VND, o NSDL. Apresentaremos os conceitos básicos da linguagem, a arquitectura e a estrutura do NSDL que suportam o VND e que ajudarão a compreender algumas opções tomadas no VND.

No quarto capítulo apresentamos a nossa solução, o VND, introduzindo os principais aspectos da implementação do VND, e as opções tecnológicas escolhidas. Apresentamos igualmente os aspectos de configuração e a versatilidade obtida com a utilização de estruturas XML que deram uma total flexibilidade e adaptabilidade ao VND.

No quinto capítulo ilustramos dois estudos de caso, iremos demonstrar a criação da rede base de estudo, depois e com essa estrutura de rede genérica, que descrevesse o mais possível uma situação real, iremos estudar os dois casos.

O último capítulo é reservado às conclusões e perspectivas futuras

2. ESTADO DA ARTE

2.1. INTRODUÇÃO

Neste capítulo abordaremos os conceitos envolvidos no decorrer deste trabalho, nomeadamente através das abordagens à autoria de cenários de rede, simulação de redes, Qualidade de Serviço e XML.

Seguidamente será apresentada uma análise às diferentes ferramentas de autoria de cenários de redes comparando-as e fornecendo dados importantes para ao desenvolvimento deste projecto.

2.2. ABORDAGEM CONCEPTUAL

De entre as diversas áreas e conceitos que este projecto envolveu, foram quatro os principais conceitos que contribuíram para a execução deste trabalho, nomeadamente a autoria de cenários de rede, simulação de redes, Qualidade de Serviço e XML.

De forma a termos uma melhor compreensão, iremos apresentar cada um de forma individual.

2.2.1 AUTORIA DE CENÁRIOS DE REDE

Dado o crescimento exponencial da Internet, e o desenvolvimento dos recursos de telecomunicações, surgem cada vez mais redes domésticas e empresariais que acedem à Internet. Portanto, existem diferentes possibilidades de conectividade, protocolos de comunicação e tecnologias emergentes de alta velocidade, que levam também a necessidade de prover Qualidade de Serviço aos tráfegos prioritários nessas redes. Nesse contexto, a comunicação de dados e o comportamento das redes é completamente indeterminístico uma vez que dependem de diversos factores como, o estado dos componentes de rede, condições de tráfego existente, de entre outros.

De forma a otimizarmos a criação e a monitorização dessas redes é importante que possamos planejar, estudar e prever o comportamento das redes, usando ferramentas gestão de redes tais como autoria e monitorização de cenários de rede.

Em particular, as ferramentas de autoria de cenários de rede têm a vantagem de permitir que se consiga visualizar a conectividade dos equipamentos e comunicações existentes entre eles de uma forma estruturada. Se essas ferramentas oferecerem um ambiente gráfico, os utilizadores ainda são capazes de visualizar a topologia física dos equipamentos.

No geral, uma ferramenta a modelização (autoria) de cenários de redes deve contemplar as seguintes características [Portnoi, 2007]:

- Flexibilidade para se adaptar a diferentes tipos de utilizadores com diferentes experiências, expectativas e objectivos;
- Uso com o menor tempo de aprendizagem inicial;
- Facilidade de configuração;
- Facilidade de implementação de novos recursos ou alteração do comportamento de recursos já existentes (personalização), e;
- Documentação e estruturação que permitam, se necessário, a adaptação através do uso de somente uma linguagem de programação ou configuração, e não de várias.

Sendo a autoria de cenários uma visão estática da rede, para conseguir analisar e prever o seu comportamento é importante que essa ferramenta possa ser integrada de alguma forma a ferramentas de simulação de redes.

Os simuladores de rede, são aplicações computacionais baseadas em modelos matemáticos que permitem simular o comportamento de uma rede de comunicação, normalmente numa área específica. A rede que estiver a ser simulada (dentro do âmbito dos tipos de redes e tecnologias suportadas pelo simulador) é modelada através da descrição dos componentes do cenário de rede em causa, de suas ligações, da especificação dos protocolos e aplicações utilizadas para a comunicação e da descrição temporal do comportamento do tráfego a ser simulado.

Uma vez o cenário de rede a ser simulado estiver descrito completamente e de forma correcta na linguagem de simulação adoptada pelo simulador, é possível a realização da simulação garantindo que os resultados obtidos terão uma margem de erro muito pequena ao compararmos com o comportamento real da rede simulada.

Um simulador exerce um papel fundamental na caracterização do comportamento e operacionalização de uma rede de computadores, quando é necessário o projecto de uma rede a ser instalada ou a proposta da manutenção de uma rede no que diz respeito à alteração de alguma tecnologia ou protocolo. De facto, a utilização de um simulador de redes pode ser útil de forma a permitir o o projecto de um sistema real e a realização de experiências com os seus respectivos modelos, com o propósito de compreender o comportamento futuro deste sistema, ou avaliar diferentes estratégias para a sua operação.

Algumas das vantagens da utilização dos simuladores de redes são:

- Permitem replicar o comportamento de sistemas reais via hardware e software, de forma a podermos determinar a configuração óptima para a sua implementação;

- Permitem a análise da rede em diferentes níveis de abstracção. Permitindo um melhor entendimento do comportamento e das interacções de qualquer componente que fazem parte do sistema em estudo. Os elementos do sistema podem ser projectados de uma forma isolada, optimizados individualmente, e então unidos, representando o sistema como um todo. Todo o sistema pode ser

modelizado através de uma técnica Top-Down, e soluções alternativas podem ser investigadas sem a implementação física do sistema (em um nível de abstracção mais alto). Obviamente, dependendo do nível de abstracção adoptado, o projecto e a implementação dos modelos para simulação podem tornar-se muitas vezes tão complexos quanto os sistemas reais que são os alvos da simulação;

- A autoria e posterior simulação de cenários de rede podem ser utilizadas como abordagens para apresentação e demonstração de conceitos. Isto é verificado principalmente em simuladores que utilizam de maneira inteligente recursos de computação gráfica e animação. O desenvolvimento de modelos de rede é uma técnica útil para o projecto de novas tecnologias e configurações de redes. Da mesma forma, os modelos resultantes também podem ser usados como um importante instrumento didáctico para o ensino de redes de computadores.

No âmbito do projecto e implementação de redes emergentes e dos novos requisitos para a comunicação multimédia, a simulação de redes torna-se mais relevante quando suporta a modelização de aspectos de Qualidade de Serviços, que apresentamos em seguida.

2.2.3 QUALIDADE DE SERVIÇO

Qualidade de Serviços (QoS) pode ser definida como sendo um conjunto de requisitos de serviços que tem que ser atingidos pela rede enquanto transporta um fluxo [RFC2386 1998].

Com o grande crescimento do uso da Internet torna-se necessário o desenvolvimento de uma arquitectura escalável e que forneça garantias de QoS. A actual abordagem de "melhor esforço" (*best effort*) utilizada pela Internet mostra-se insuficiente para a satisfação das necessidades de certas aplicações cujo uso tem vindo a crescer cada vez mais. Serviços com cada vez maior utilização como o e-mail, transferência de ficheiros, comunicações de voz são suportados pela actual estrutura da Internet. Com o crescimento da largura de banda e da quantidade de informação multimédia disponíveis para os utilizadores de computadores, torna-se evidente que cada vez mais serviços a serem transmitidos pela Internet no futuro necessitarão da utilização de QoS.

A fim de implementar diferentes níveis de QoS ao protocolo IP, o IETF propõe o modelo de Serviços Integrados (Intserv) [RFC2998 2000], o modelo de Serviços Diferenciados (Diffserv) [RFC2475 1998], e o protocolo de *Switching* baseado em Etiquetas Multiprotocolo (MPLS) [RFC3031 2001]. Essas soluções não têm como objectivo a substituição do actual modelo de serviços da Internet, mas ampliar para novos serviços e, conseqüentemente, diferentes níveis de QoS ao protocolo IP.

Seguidamente apresentamos uma visão geral sobre cada um dos modelos, a apresentação dos modelos tem como objectivo a melhor compreensão do âmbito do projecto, sem ter a pretensão de apresentar detalhadamente cada uma das soluções em causa..

Integrated Services (IntServ)

O RFC 2998 define o Intserv, como sendo uma arquitectura que enuncia um conjunto de extensões para o modelo tradicional da Internet conhecido por "best-effort", sendo o seu objectivo permitir ligações ponto-a-ponto com QoS oferecido para as aplicações [RFC2998 2000].

A garantia de QoS é dada a cada fluxo individualmente, necessitando manter nos routers informações de reserva para cada um dos fluxos [RFC2998 2000].

Apesar das aplicações tradicionais na Internet envolverem comunicações entre dois computadores, existem algumas aplicações emergentes, como a teleconferência de vídeo/áudio e a recuperação de documentos multimédia, que requerem comunicação multiponto (multicasting).

Para que as aplicações possam ter um desempenho satisfatório na Internet, existe a necessidade do uso do routing multicast, que permite a comunicação de uma fonte de dados e diversos receptores, sem a necessidade do envio de uma cópia de cada pacote para cada um dos receptores, optimizando o uso da largura de banda da rede. A implementação de IntServ contempla a distribuição de fluxos multicasting.

Differentiated Services (DiffServ)

Differentiated Services (DiffServ) é definida como sendo uma arquitectura orientada para a implementação da diferenciação de serviços de forma escalável na Internet [RFC2475 1998].

O condicionamento de tráfego consiste na medição, policiamento, modelação e/ou remarcação do tráfego, de forma a assegurar que este entra num domínio DiffServ conforme a política de aprovisionamento de serviço predefinida [RFC2475 1998].

O principal objectivo é acomodar um grande número de serviços e políticas de provisionamento em comunicações ponto-a-ponto ou de difusão dentro de um conjunto de redes. Não é definido nenhum tipo de serviço em particular, apenas define o tratamento que cada pacote de dados recebe a cada salto ("hop") consoante classes de prioridades associadas a cada pacote.

Numa arquitectura DiffServ, os nós de fronteira estabelecem a ligação entre um domínio DiffServ e outros domínios, que podem suportar ou não DiffServ, existindo dois tipos: os nós de saída, que tratam o tráfego de saída do domínio e os nós de entrada, que tratam tráfego de entrada. Cada nó de fronteira (*Edge Routers*) pode desempenhar as funções de nó de saída e de entrada intercaladamente. Os nós interiores apenas se ligam a outros nós interiores ou de fronteira dentro do mesmo domínio. A maioria das operações DiffServ (monitorização e condicionamento) é efectuada nos routers situados nos nós de fronteira (*Edge Routers*). Os nós interiores (*Core Routers*) são responsáveis pela implementação de comportamento associado a uma dada classe de QoS, também chamado de *Per-Hop-Behaviour*.

Multi Protocol Label Switching (MPLS) .

O Multi Protocol Label Switching (MPLS) é definido como um protocolo orientado para a rápida comutação e encaminhamento de pacotes [RFC3031 2001].

Mais especificamente, MPLS possui mecanismos que gerem o fluxo do tráfego de diferentes granularidades. [RFC3031 2001].

O MPLS tem a capacidade de dar resposta aos problemas de performance e escalabilidade que os fornecedores de acesso a redes IP (*Internet Protocol*) desejam superar.

Com a tecnologia MPLS, as redes IP tem a possibilidade de suportar diferentes serviços com diferentes requisitos de QoS, aumentando a eficiência e fiabilidade das mesmas. Em particular, esta tecnologia melhora a leitura e envio de pacotes de informação em determinados nós da rede optimizando a forma como se processa a transmissão de informação correspondente a um serviço [Gouveia 2003].

Uma rede MPLS contém nós especiais, denominados por *Label Switching Routers (LSRs)*, que podem ser de dois tipos: nós de acesso ou nós de trânsito. Os nós de acesso são os nós de extremidade (origens e/ou destinos) de um ou mais serviços. Os nós de trânsito correspondem aos nós intermediários na rede, de entre um conjunto de nós candidatos (e que não inclui os nós de acesso) de forma a possibilitar a comunicação entre nós de acesso, sendo a sua localização objecto de decisão. O encaminhamento de um pacote IP (informação) desde um nó de acesso inicial até ao correspondente nó de acesso final é efectuado através de um caminho, designado por *Label Switched Path (LSP)*, e baseia-se na etiqueta que esse pacote recebe na origem (etiqueta na qual consta o destino e a QoS requerida por esse pacote). O encaminhamento assim baseado implica que, como referido acima, a transmissão de informação de um determinado serviço não seja condicionada pelas de outros serviços, o que representa uma vantagem relativamente a redes IP mais tradicionais.

Outro aspecto fundamental no desenvolvimento deste trabalho foi a adopção de representações baseadas em XML para a personalização da ferramenta desenvolvida. A linguagem XML é apresentada brevemente na próxima secção.

O XML, *Extensible Markup Language* [ISO 8859], é uma forma restrita do SGML (*Generalized Markup Language*) [ISO 8879] é uma norma W3C, aparecendo pela primeira vez em 1996 e sendo publicada como recomendação em 21 de Fevereiro de 1998 (versão 1.0) e mais tarde, em 2004, corrigida e aperfeiçoada. Esta metalinguagem- uma linguagem usada para descrever outras linguagens - foi desenvolvida com os seguintes objectivos:

- Ser directamente utilizada via Internet e suportar uma vasta variedade de aplicações;
- Ser compatível com o SGML, e;
- Permitir a criação de documentos que sejam produzidos com facilidade, e minimamente legíveis por um ser humano.

Existem diversas aplicações derivadas do XML, tais como:

- DocBook [DocBook, 2010] – descreve documentos, tais como: livros, manuais e artigos;
- Chemical Markup Language (CML) [CML, 2010] – descreve a estrutura de moléculas;
- Wireless Markup Language (WML) [Wapforum, 2010] – descreve dados de redes sem fios;
- Scalar Vector Graphics (SVG) [W3, 2010] – descreve gráficos bidimensionais;
- Mathematical Markup Language (MathML) [W3, 2010] – descreve expressões matemáticas;
- Extensible Style Language (XSL) [W3, 2010]– permite a formatação e transformação de documentos XML, e;
- Extensible Hypertext Markup Language (XHTML) [W3, 2010]– transição entre HTML e XML.

Apesar das facilidades que o XML pode facultar na definição de qualquer conjunto de etiquetas de marcação, existem um conjunto de

regras que têm de ser seguidas na produção de um documento XML, para que este seja considerado como bem formado.

Ao contrário da linguagem HTML (HyperText Markup Language), se um documento XML não for considerado bem formado, este não é analisado, nem interpretado, emitindo normalmente uma mensagem de erro. Os analisadores de XML são intolerantes com a sintaxe do documento.

Um documento XML pode ser um simples ficheiro de texto, ou um documento gerado dinamicamente, que contém um conjunto de marcas dispostas de acordo com algumas regras. Todos os documentos XML possuem um cabeçalho do tipo:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes|no"?>
```

A restante parte de um documento XML é composta por texto e etiquetas de marcação que descrevem a estrutura do documento.

As etiquetas de marcação, ou elementos, de um documento XML têm de respeitar as seguintes regras:

- A definição de *containers* tem de ser sempre colocadas aos pares, definindo uma zona demarcada entre um início (<nome>) e um fim (</nome>);

Na definição de *elementos* as marcas não necessitam de ser especificadas aos pares, fechando-se assim a marca de início (<marca/>);

- O nome das marcas não pode incluir espaços em branco, tem de começar por letras ou pelo carácter underscore "_" e não pode incluir caracteres que não sejam letras, números, *underscores*, hífenes ou pontos;

- Os nomes em XML são sensíveis ao uso de maiúsculas ou minúsculas, isto é, "nome" é diferente de "Nome";

- Tem de existir uma estrutura em árvore começando por uma etiqueta de raiz, e as marcas não se devem entrelaçar, e;

As etiquetas de marcação XML podem possuir atributos. Estes estão sempre associados a uma marca e de alguma forma qualificam-na ao funcionarem como propriedades da mesma. Os valores dos atributos devem estar delimitados por aspas "" ou plicas `.`.

Não sendo possível apresentar todas as propriedades e definições do XML num capítulo, para maiores informações sugere-se

a leitura de XML Bible (2nd Edition) [Harold 2001] e XML Schema [W3C 2000].

2.3. ABORDAGEM A LITERATURA

Como já discutido anteriormente, a principal motivação para o desenvolvimento deste trabalho foi a proposta e o desenvolvimento de uma ferramenta de autoria de cenários de rede que permitisse colmatar áreas não suportadas pelas outras ferramentas disponíveis.

Para o estudo e comparação entre as soluções existentes na literatura, foi realizado um levantamento das ferramentas mais comuns e actuais, sendo a lista completa apresentada no ANEXO I.

Dessas ferramentas foram desconsideradas todas as que estão descontinuadas ou não tem informação disponível.

Dessa forma, e após essa selecção, ficamos com 11 ferramentas, que apresentamos a seguir:

- VISIO [Microsoft, 2010]
- PACKET TRACER [Cisco, 2010]
- NS-2 [University of Southern California, 2007]
- SIMGRID [Simgrid, 2010]
- ARPO MODELER [Klugsolutions, 2008]
- BOSON NETSIM [Boson, 2007]
- GLIFFY [Gliffy, 2010]
- SIMARIS [Siemens, 2010]
- INTALIO DESIGNER [Intalio, 2010]
- CONCEPTDRAW [Conceptdraw, 2010]
- EDRAW [Edrawsoft, 2010]

Neste grupo estão incluídas:

- Ferramentas que tem por base a representação de redes e outras que permitem a representação, mas que são mais generalistas;
- Ferramentas com licenciamento pago e outras gratuitas;

- Ferramentas cujo código é proprietário e outras cujo código é aberto;
- Ferramentas que se dedicam somente a algum tipo (ou marca) de equipamento e outras que são mais generalistas;
- Ferramentas que permitem a exportação e importação para outros software e ferramentas que tem forma de exportar rígida e proprietária, e;
- Ferramentas com diferentes funcionalidades.

Assim e para melhor compreendermos e analisarmos as diferentes ferramentas, elaboramos o quadro comparativo apresentado na Figura 1.

Software	MICROSOFT VISIO	PACKET TRACER	NS-2	SIMGRID	ARPO MODELER	BOSON NETSIM	GLIFFY	SIMARIS	INTALIO DESIGNER	CONCEPTDRAW	EDRAW
Características Permite representar redes informáticas	Sim	Sim	Sim	Possível através de extensões	Sim	Sim	Sim	No futuro, principalmente eléctricas	Sim	Sim	Sim
Tipo de licenciamento	Compra de licença	Gratuito para estudantes.	Open Source	Gratuito	Académico livre e completo com baixo Custo.	Compra de licença	Compra de licença	Compra de licença	Software livre com módulos pagos	Compra de licença	Compra de licença
Código aberto Extensibilidade	Não Sim, permite um largo leque de objectos base e adicionar novos.	Não Não, só para produtos CISCO	Sim Tem imensos trabalhos desenvolvidos na área de qos com base no produto	Sim Pouca, por enquanto segundo os autores.	Sim Publicação em HTML, Exportação/Importação Entre ferramentas	Não Não, só para produtos CISCO	Não Muito pouca, software proprietário	Não Tem alguns módulos que permitem alguma extensibilidade	Sim Automação com BPM (Business Process Management)	Não Produto proprietário, dependente de actualizações	Não Produto proprietário, dependente de actualizações
Funcionalidades	Desenho técnico, estático, pouco flexível para simulação	Desenho e simulação.	Não permite chegar ao ponto que queremos.	Obrigatoriedade de saber programar em C++, software aberto que permite ir acrescentando funcionalidade através de módulos gratuitos. Com módulos específicos, ou desenvolvendo.		Desenho e simulação	Desenho técnico, estático, pouco flexível para simulação.	Mais virado para redes eléctricas, no entanto existe a possibilidade de ter desenho de rede no futuro.		Desenho de redes estático, sem simulação.	Desenho técnico, estático, pouco flexível para simulação
Exportação/Importação	Permite .xml (não W3C)	Em formato proprietário	Com desenvolvimento pode-se exportar para outros formatos		EPC		Em formato proprietário, ou vectorial/gráfico.	Em formato proprietário		PDF, Power Point, Flash, HTML	Em formato proprietário

Figura 1 -Comparativo das ferramentas de autoria de redes

Com o apresentado na tabela anterior podemos concluir que quase todos os produtos permitem representar redes, sendo que quase dois terços são gratuitos. A extensibilidade é permitida por quase todos os softwares, sendo que no entanto torna-se muitas vezes muito complicada. A simulação é permitida por três, sendo que dois são produtos comerciais. A principal limitação dessas aplicações está relacionada à possibilidade de exportação, pois quase todos os softwares não permitem a interoperabilidade com outras ferramentas.

2.4. CONCLUSÃO

Neste capítulo fizemos uma abordagem conceptual às noções envolvidas na realização deste trabalho de mestrado e apresentamos a nossa perspectiva sobre o mesmo, bem como a sua importância.

Foi efectuada uma análise das diferentes ferramentas de autoria de cenários de redes e principalmente das diferenças existentes entre elas, de forma a conhecermos as soluções existentes na literatura e percebermos melhor o que pretendemos do nosso trabalho.

3. NSDL

3.1. INTRODUÇÃO

A partir da revisão da literatura realizada no capítulo II, pudemos verificar que as ferramentas existentes para a gestão e modelização de cenários de redes existentes não dão qualquer suporte à exportação da descrição dos cenários criados, permitindo assim a interoperabilidade entre diferentes ferramentas, sobretudo entre a autoria e simulação de redes.

Neste capítulo apresentamos a *Network Scenario Description Language (NSDL)*, uma linguagem proposta no âmbito do trabalho doutoramento do Mestre Eduardo Marques na Universidade da Madeira [Marques 2010], para proporcionar a interoperabilidade de diferentes ferramentas de gestão, autoria, simulação e análise de redes.

O desenvolvimento da ferramenta de autoria de cenários de redes apresentada nesta dissertação foi baseado na proposta da linguagem NSDL.

Apresentaremos os conceitos básicos da linguagem, a arquitectura e a estrutura do NSDL, que ajudarão a compreender algumas opções tomadas no desenvolvimento do VND.

3.2. CONCEITOS BÁSICOS

O objectivo do NSDL é fornecer um vocabulário e um conjunto de regras, tanto em condições de apoiar a descrição de redes. Mais do que apenas descrever a topologia da rede com os seus objectos e características, NSDL introduz um conceito importante: separar a rede de várias perspectivas que possam existir sobre essa rede.

Portanto, NSDL visa proporcionar uma rica descrição dos objectos de rede e dos seus parâmetros e também uma descrição dos vários cenários de rede em todo o ciclo de vida da rede.

Nesse sentido, a estrutura e os parâmetros NSDL definidos também devem ser ricos o suficiente para descrever qualquer tipo de rede de dados e permitir introduzir no futuro dados para suportar objectos novos de apoio a futuras redes de dados.

O idioma seleccionado para a descrição da linguagem NSDL é o XML. A riqueza e flexibilidade do XML determinaram a sua escolha, associada à maturidade das definições XML e a um conjunto de ferramentas disponíveis. Portanto, XML assegurou ao NSDL princípios como a abstracção, a simplicidade e a extensibilidade.

Os princípios usados para projectar NSDL foram:

- A simplicidade, o que significa que a linguagem tem que ser simples e clara, não apenas a manipulação por uma aplicação ou ferramenta, mas também possível de ser editado por um utilizador humano com um simples editor de texto;
- Definição de múltiplos níveis de abstracção, permitindo especificar não apenas simples descrições de alto nível de um cenário de rede, mas também, se necessário, para que a possibilidade de criar uma descrição muito detalhada de todos os objectos de cenário de rede e seus parâmetros; e
- Extensibilidade o que significa que novos objectos e parâmetros podem ser incorporados em futuras descrições.

O uso de uma descrição linguística única em vários momentos de um ciclo de vida da rede poderia ser muito vantajosa. Se os utilizadores responsáveis pela rede estão familiarizados com essa linguagem, eles podem facilmente compreender o estado actual da rede, portanto, a gestão da rede é optimizada. Outras capacidades de integração também podem ser alcançadas já que os desenvolvedores de uma ferramenta de rede são capazes de adicionar capacidade de exportação e importação de e para NSDL, fazendo o seu trabalho compatível com a linguagem NSDL.

Como tal, o NSDL pode proporcionar a colaboração entre as ferramentas e os utilizadores.

A interoperabilidade é de facto outra importante vantagem do NSDL. Se um gestor de redes utilizar várias ferramentas compatíveis com NSDL, ele pode facilmente analisar uma rede usando cada uma delas, conseguindo assim a obtenção de resultados integrados.

Outra possibilidade de utilização do NSDL, pode ser quando o utilizador necessita executar duas simulações semelhantes sobre a mesma rede, com dois simuladores diferentes.

Além disso, podemos considerar o caso onde o utilizador precisa fazer uso de uma função específica de uma ferramenta, que normalmente não está disponível, devido às diferentes linguagens de descrição de rede aplicadas pela ferramenta. Nesse caso, ele seria obrigado a aprender uma outra linguagem a fim de implementar o seu código num formato diferente.

Sendo uma linguagem baseada em XML, NSDL pode ser estendida através da adição de novos parâmetros e criação de novos objectos sobre os elementos já definidos na linguagem.

Por exemplo, NSDL pode ser estendido para auxiliar na descrição de redes de sensores sem fio. A partir do objecto base *Node*, é possível definir um novo nó, denominado *Nó sensor Wireless*, que irá receber todas as características do objecto base *Node*, e vai acrescentar as suas componentes, tais como, interfaces especiais e protocolos de redes de sensores.

Seguidamente apresentamos uma abordagem à arquitectura do NSDL.

3.3. ARQUITECTURA

NSDL é uma linguagem para descrever cenários de rede, mas também incorpora alguns outros elementos na sua arquitectura de forma a facilitar a integração de diferentes ferramentas de gestão de redes.

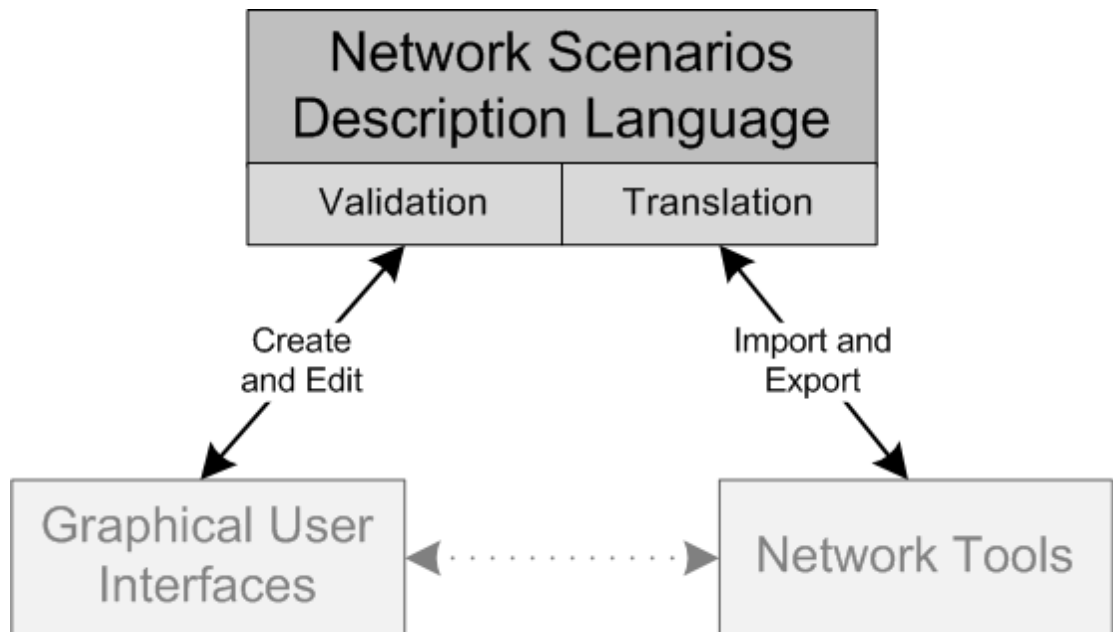


Figura 2 Arquitectura NSDL. [Marques, 2010]

A arquitectura ilustrada na Figura 2 é composta pelo núcleo da linguagem, que é apoiado pelo bloco de Validação e Tradução.

O bloco de validação contém as funções responsáveis por assegurar que um ficheiro criado ou editado em NSDL é bem formado e correcto. Ele contém as normas e valores limites para a estrutura e os parâmetros, respectivamente. O objectivo do bloco de tradução é fornecer funções para importação e exportação de dados entre os arquivos NSDL e os ficheiros de ferramentas compatíveis com NSDL. Este bloco conterá os ficheiros com as regras para converter um arquivo de determinada ferramenta ou formato para a linguagem NSDL e as regras para fazer o processo inverso.

O apoio à validação e tradução é realizado através de algumas ferramentas XML, principalmente, o uso de schemas XML para validar os ficheiros XML e a tradução para a conversão para outros formatos. No futuro, quando for necessário, outras ferramentas podem ser usadas para executar funções mais complexas sobre ficheiros NSDL.

Outros componentes importantes para NSDL são os componentes externos.

Um primeiro componente está relacionado com a criação (Modelagem) de ficheiros através de Interfaces Gráficas com o

Utilizador (GUIs). Eles auxiliam os utilizadores a especificar e interagir com rede, cenários e ferramentas.

Em segundo lugar estão os componentes relacionados às Ferramentas de Redes existentes, que permitirão a importação, exportação e aplicação dos dados de ficheiros NSDL, a fim de promover interoperabilidade entre diferentes ferramentas. Algumas dessas ferramentas podem ser os ambientes de autoria gráficos, ou as ferramentas de simulação de rede.

Em particular, as ferramentas gráficas para a autoria de cenários de redes fornecem uma interface integrada para permitir, por exemplo, a criação, edição e funções de importação / exportação.

Quando esses aplicativos não são integrados, dois cenários podem ser considerados:

- Possibilidade de aplicar NSDL para fornecer interoperabilidade entre as ferramentas, ou;
- Melhorar o código de GUIs existentes a fim de propiciar a interoperabilidade entre eles.

Em seguida apresentaremos a estrutura do NSDL.

3.4. ESTRUTURA NSDL

Um ficheiro NSDL é baseada em XML e possui dois elementos básicos (Figura 3): Rede e Cenários.

O elemento de **rede** contém a descrição de um cenário de redes, sua identificação, seus componentes e suas interligação, e sua caracterização.

O elemento **cenários** (Scenarios) pode conter várias descrições, cada uma referindo-se a uma utilização específica, ou contexto, para essa rede.

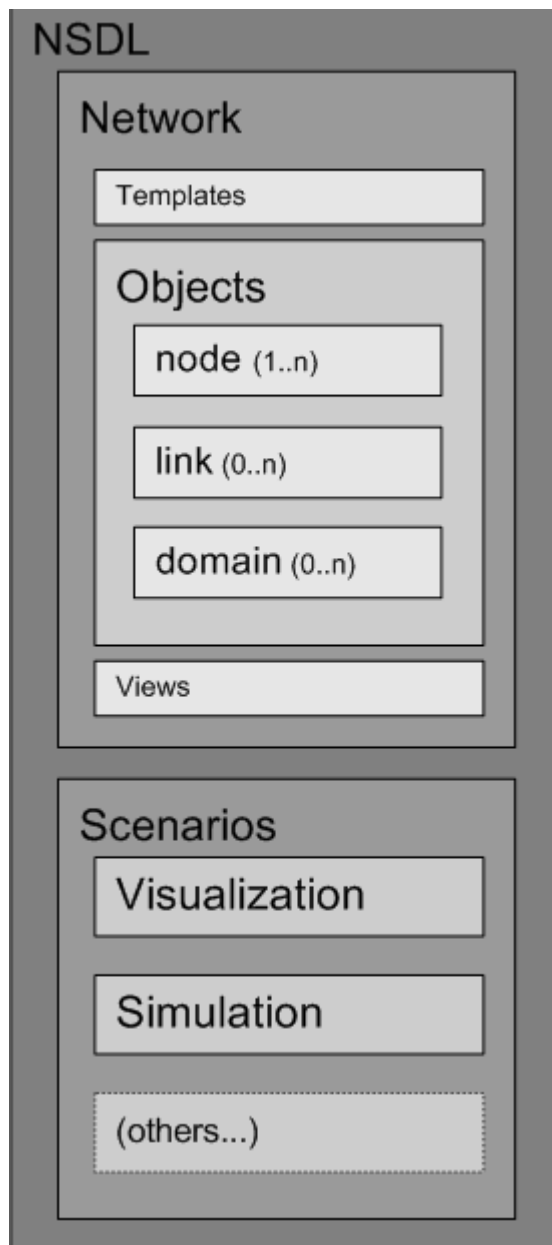


Figura 3 Os elementos do NSDL [Marques, 2010]

O elemento de **rede** é composto por: *Templates*, *Objects* e *Views*.

Como os objectos contém a descrição da topologia da rede, este é o principal componente da linguagem. Este elemento é composto por *nós*, *links* e *domínios*. Alguns outros elementos importantes, mas não obrigatório são *templates* e *views*. Os *Templates* são elementos importantes para simplificar a descrição de objectos similares. As *views* são um mecanismo aplicado para agrupar objectos de rede a serem utilizados em cenários.

No elemento **Cenários**, dois elementos são introduzidos: *Visualização* e *Simulação*. O elemento de visualização fornece informações adicionais para enriquecer a descrição da rede, como o posicionamento dos objectos no GUI.

Todos os parâmetros necessários para implementar uma simulação através da rede, nomeadamente utilizando uma simulação genérica ou ferramenta particular é definido no elemento de *Simulação*.

Nas próximas secções comentamos brevemente os elementos dos componentes de *Rede* e *Cenários*.

▪ REDE

O elemento *Objecto* pode ser considerado o elemento central da linguagem NSDL uma vez que descreve a topologia e características da rede.

A descrição é feita pela identificação de toda o cenário de rede e de seus objectos e seus respectivos parâmetros. Para esta finalidade três objectos foram definidos:

- *Nó*: representa todos os equipamentos activos numa rede, como routers, computadores, switches e outros;
- *Link*: é a conexão entre dois nós, e pode representar qualquer tecnologia tais como Ethernet, ADSL, fibra óptica, de entre outros, e;
- *Domínio*: é um objecto que permite descrever uma rede completa, para posterior reutilização. Um exemplo é a Internet ou a uma rede que suporte alguma tecnologia em particular.

Além dos parâmetros disponíveis em um nó, existem outros objectos que podem ser descritos a partir do nó, nomeadamente:

- *Interface*: representa um interface de rede e deve ser usado quando o objecto *link* não é apropriado. Um exemplo de utilização deste objecto pode ser a descrição das redes sem fio;
- *Protocolo*: é o elemento que descreve o protocolo utilizado em um objecto e os seus atributos. Um exemplo é a informação do

endereço IP. Isso deve ser descrito pelo objecto Protocolo de Internet (IP), e;

- *Aplicação*: descreve o tráfego gerado e recebido por um nó da rede. Uma aplicação pode ser uma aplicação conhecida, como HTTP, FTP, ou um gerador de tráfego particular. O uso desses objectos num nó da rede é opcional.

Outros dois sub-elementos também presentes no elemento de rede são *templates* e *views*. Eles são opcionais, e visam a favorecer o utilizador no processo criação e descrição da rede (modelagem) e organização.

Os *templates* ajudam o utilizador na descrição do ficheiro NSDL, minimizando descrições longas e redundantes. Ele contém as repetições que seriam utilizadas para descrever os componentes da rede. Os *templates* podem ser utilizados quantas vezes for necessário, e podem ser representados por qualquer tipo de objecto de rede e todos os seus parâmetros possíveis.

As *Views* permitem a criação de grupos de objectos nas redes NSDL, sem qualquer limitação. O objectivo desse objecto não é para caracterizar a rede, mas especificar grupos especiais dentro da rede (Por exemplo, um caminho, routers, aplicações).

A Figura 4 ilustra a descrição NSDL para o componente *Rede*.

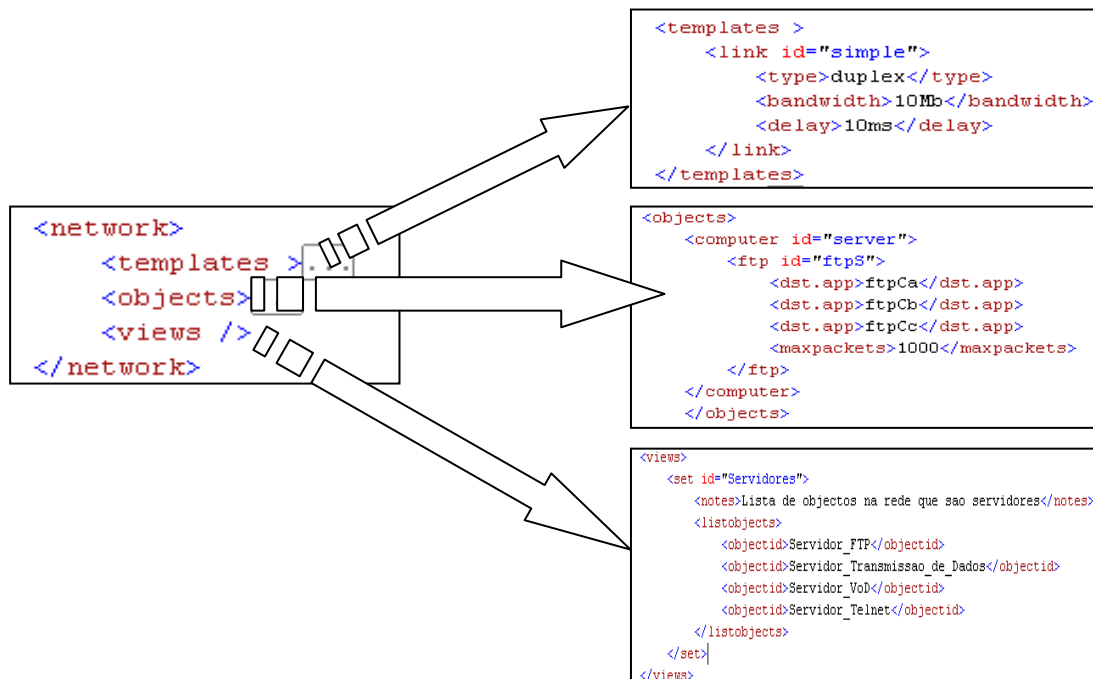


Figura 4 Representação do componente Network

▪ CENÁRIOS

O objectivo do componente *Cenários* é trazer informações adicionais e relevantes para a rede, a partir de várias perspectivas específicas. Essas perspectivas dizem respeito à utilização de diferentes ferramentas que suportem NSDL.

O elemento de cenários é composto por sub-elementos *visualização* e *Simulação*.

O elemento de *visualização* permite a descrição do posicionamento e das informações gráficas sobre um cenário de rede. Em algumas ferramentas de rede este tipo de informações podem não ser relevantes e podem eventualmente, ser ignorada, mas noutros, é fundamental. Um exemplo refere-se a ser necessário adicionar o posicionamento preciso, usando coordenadas, para identificar o local correcto onde todos os objectos estão localizados na interface gráfica.

O elemento *simulação* permite, como o nome indica, descrever todas as informações relevantes para apoiar as simulações sobre a rede. Por exemplo, podemos considerar a identificação da ferramenta/simulador, os parâmetros gerais da simulação, tais como duração, condições para terminar (ou para ser interrompida), os eventos para a simulação e, informações sobre as estatísticas e os resultados esperados.

A Figura 5 ilustra a descrição NSDL para o componente *Cenários*.

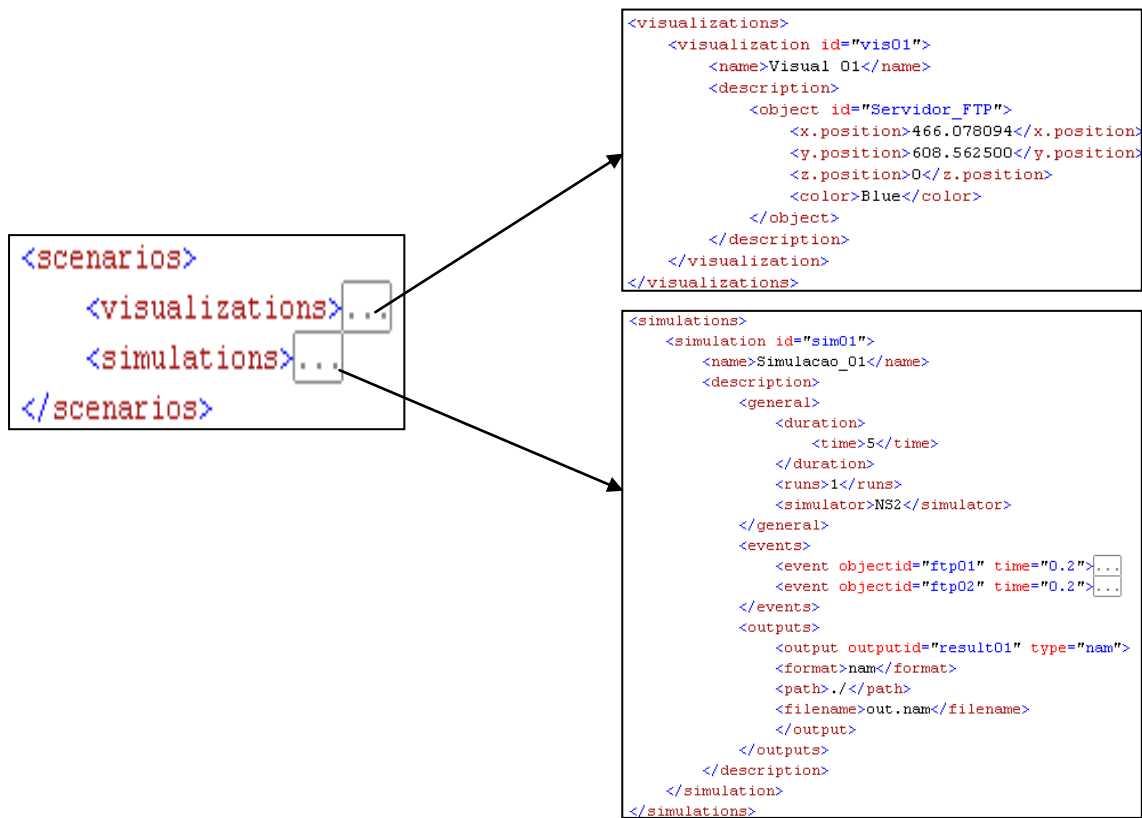


Figura 5 Representação do componente escenarios

3.5. CONCLUSÃO

Neste capítulo apresentamos a linguagem que esteve na origem do VND, o NSDL. O NSDL permite descrever completamente um cenário de redes e seus componentes, de forma que proporcione a interoperabilidade entre diferentes ferramentas de gestão de redes.

Portanto, foi realizada uma apresentação dos conceitos básicos da linguagem, dando uma breve abordagem à arquitectura e à estrutura do NSDL.

No próximo capítulo apresentaremos a principal contribuição deste trabalho, que é a implementação da ferramenta gráfica VND.

4. VND: A INTERFACE GRÁFICA PARA AUTORIA DE CENÁRIOS DE REDES

4.1. INTRODUÇÃO

Neste capítulo apresentamos os principais aspectos do desenvolvimento do Virtual Network Description (VND). O VND foi concebido como uma ferramenta simples de forma a permitir a criação de cenários de redes personalizáveis.

Portanto, são abordados os requisitos, quer sejam funcionais como os não funcionais, apresentados os casos de utilização existentes e os diagramas de actividade para os casos de utilização mais importantes. Da mesma forma, nesse capítulo são apresentados a arquitectura do sistema, os Protótipos Abstractos Canónicos entre outras características de implementação da ferramenta.

Posteriormente apresentaremos os principais aspectos da implementação do VND, e as opções tecnológicas escolhidas, tais como a linguagem FLEX.

Seguidamente apresentamos os aspectos de configuração e a versatilidade obtida com a utilização de estruturas XML que deram uma total flexibilidade e adaptabilidade ao VND.

Posteriormente realizaremos uma breve abordagem sobre o desenvolvimento da ferramenta, baseada no número de linhas de código criadas.

Da mesma forma, é realizada uma introdução ao aspecto gráfico do VND, disposição das ferramentas, e funcionalidades existentes.

4.2. REQUISITOS

O levantamento de requisitos assume uma vital importância na definição do sistema, pois reflecte as expectativas do "cliente".

Normalmente e por definição, os requisitos são divididos em requisitos funcionais e requisitos não funcionais [Sommerville, 2001]. Como alguns requisitos podem não estar sob definição de nenhuma das anteriores categorias, a experiência fez com que nos projectos de

software que somos responsáveis, adicionamos uma outra categoria, a categoria de outros requisitos, ficando dessa forma catalogadas todas as especificações do cliente.

Todos os requisitos do VND foram resultado das diversas reuniões realizadas, de forma a estruturar os mesmos, os requisitos anotados foram divididos nas três categorias descritas anteriormente e são apresentados separadamente.

REQUISITOS FUNCIONAIS

Os requisitos funcionais definem os serviços que o sistema deve oferecer, como o sistema deve reagir a certas entradas e como comportar-se em determinadas situações.

Os requisitos funcionais que foram tomados em consideração na criação do VND são:

- O sistema deverá permitir guardar o trabalho e posteriormente voltar a utilizá-lo;
- O utilizador deverá poder criar redes informáticas através do método "drag-and-drop";
- Deverá ser possível ao utilizador utilizar objectos de acordo com a sua necessidade, devendo para isso o sistema ter a capacidade de ler e criar qualquer lista de objectos criados pelo utilizador, desde que siga com as regras definidas;
- Deverá existir uma descrição, que seja possível consultar a qualquer momento, que reflecta o que o utilizador está a fazer, tendo os objectos criados e as suas configurações;
- Deverá ser previsto que alguns objectos possam ter desde nenhuma a várias instâncias, devendo ser dada a possibilidade ao utilizador de escolher consoante a configuração desejada do objecto específico;
- Quando um utilizador selecciona um objecto ele ser realçado na área de desenho;
- Deverá ser possível alterar as configurações de um objeto quando ele é seleccionado por um utilizador;
- Em cada objecto criado o ID e Nome devem ser criado automaticamente;
- Quando são criadas as ligações também deverão ter automaticamente o *id* dos objectos que ligam;
- Deve ser possível ao utilizador mover e eliminar objectos;

- As ligações deverão ajustar-se quando se move os objectos que a compõem;
- As diferentes áreas que constituem o sistema devem poder ser ajustadas;
- As opções de *undo*, *redo*, *zoom in* e *zoom out* devem ser disponibilizadas;
- Deve ser possível associar representações gráficas aos objectos, nomeadamente através de ícones;
- Um *link* não é passível de possuir uma representação gráfica através de um ícone, no entanto, deve ser possível parametrizar a cor da ligação, e;
- Cada vez que um utilizador selecciona um objecto da lista de objectos, o seu ícone deve ser mostrado como forma de facilitar o entendimento do que vai ser adicionado, caso a representação escrita não seja suficiente.

REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais definem as restrições nos serviços do sistema, tais como tempo de resposta, segurança, fiabilidade, disponibilidade, normas, etc.

Os requisitos não funcionais que foram tomados em consideração na criação do VND são:

- Deverão ser utilizadas linguagens e sistemas que permitam um acesso rápido e fluente;
- O sistema deverá ter um layout simples de utilizar, que permita a um novo utilizador ambientar-se rapidamente e com pouca formação necessária;
- Sempre que seja necessário fazer *download*, *upload* ou exportação de ficheiros, deverá preferencialmente ser utilizados ficheiros XML;

- *Objects, Links, Views, Templates, Simulations* e outras definições devem ser consideradas e implementadas de acordo com as especificações do NSDL;
- Deverá ser possível exportar o trabalho para o NSDL, e;
- A representação da lista de objectos deve ser feita de maneira organizada e fácil de utilizar.

OUTROS REQUISITOS

Nesta secção estão todos os requisitos que não se enquadram nas duas secções anteriores.

Os restantes requisitos que foram tomados em consideração na criação do VND são:

- Deverão ser criados os menus necessários, mas a zona central deverá ser deixada “limpa” para o utilizador criar a rede;
- O sistema deverá ter uma lista de objectos base;
- Todo o sistema deve estar documentado, e;
- Deverão ser criados manuais para o ficheiro de importação.

4.3. CASOS DE UTILIZAÇÃO

Os casos de utilização ajudam a compreender melhor um sistema, auxiliando o cliente da solução, mas também o desenvolvedor, a especificar claramente para cada tipo de utilizador a sua interacção com o sistema [Cockburn, 2006].

No início da especificação do sistema, foi pensado que existiriam dois tipos de utilizadores:

- O utilizador comum, tipicamente um aluno, ou qualquer outra pessoa que utilizasse o software, e;

- O supervisor, alguém que teria responsabilidade de gerir a ferramenta e decidir que tipo de bibliotecas disponibilizar.

No seguimento do desenvolvimento e na tentativa constante de criar uma ferramenta versátil e adaptável, foi proposto e prontamente aceite, ter como objectivo conseguir-se que o próprio utilizador faça download da biblioteca genérica e parametrize de acordo com as suas necessidades e use a sua própria biblioteca, deixando assim de fazer sentido ter dois utilizadores, passando a existir somente um.

Deste modo, apesar do esforço de programação aumentar, os casos de uso diminuiriam.

Os casos de uso do VND são ilustrados na Figura 6.

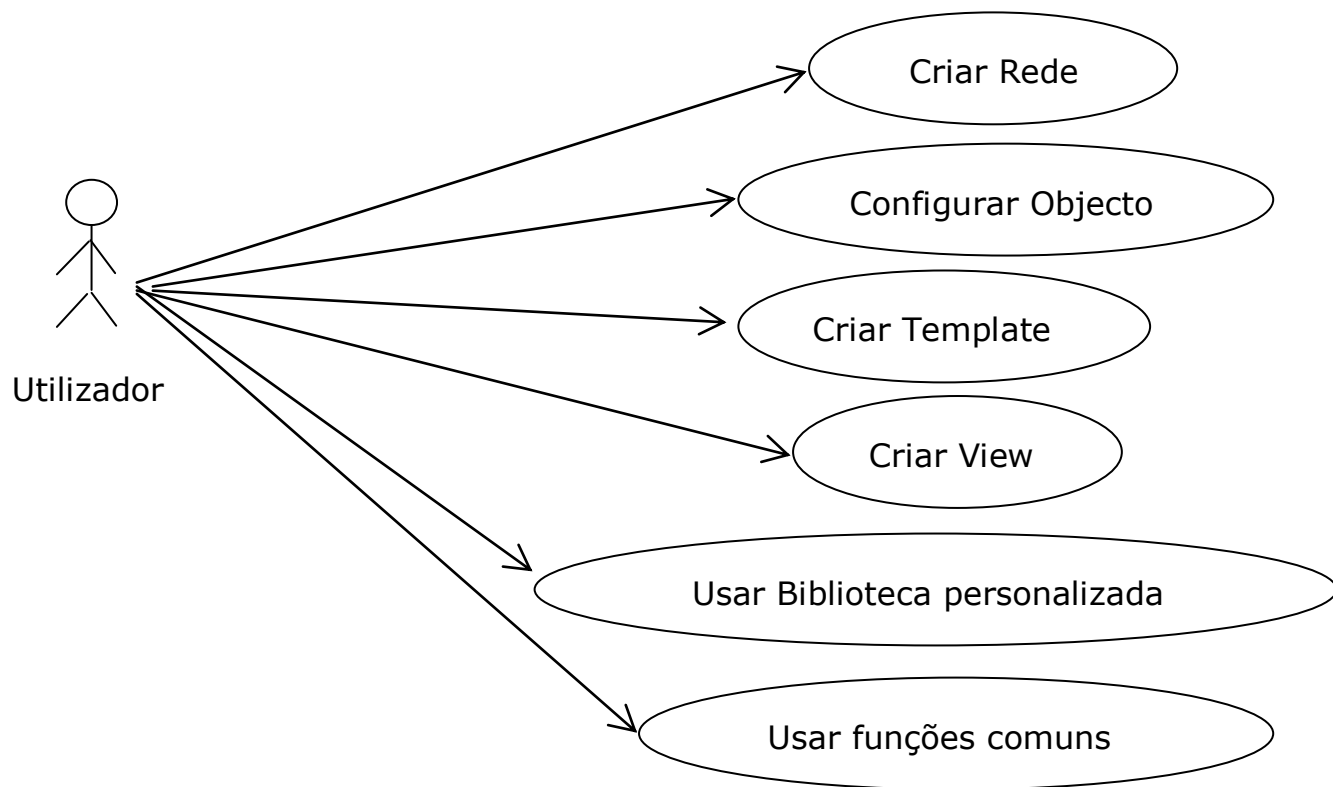


Figura 6 Casos de Utilização

A descrição de cada caso de utilização apresentado na Figura 6 é realizada a seguir:

Criar Rede – O caso de utilização “Criar Rede” prevê as situações de criar uma rede, salvar a configuração de rede, carregar uma configuração de rede previamente salva e exportar para NSDL;

Configurar Objecto – O caso de utilização “Configurar Objecto” prevê as situações de criar o objecto, alterar os dados da configuração do objecto e eliminar o objecto;

Criar Template – O caso de utilização “Criar Template” prevê as situações de criar o template, criar novos objectos com base nesse template e eliminar o Template;

Criar View – O caso de utilização “Criar View” prevê as situações de criar uma *view* e exibir objectos dessa *view*;

Usar biblioteca personalizada – O caso de utilização “Usar biblioteca personalizada” prevê as situações de carregar uma biblioteca com objectos personalizados pelo utilizador, e;

Usar funções comuns – O caso de utilização “Usar funções comuns” prevê as situações de Imprimir, Mais zoom, Menos Zoom, Retroceder e Avançar.

4.4. DIAGRAMAS DE ACTIVIDADE

O diagrama de actividades representa o fluxo de controlo entre as diferentes actividades de um sistema [Ambler, 2002]

Para o VND o diagrama de actividades mais comuns são o de configurar um objecto e criar rede, ilustrados nas Figuras 7 e 8.

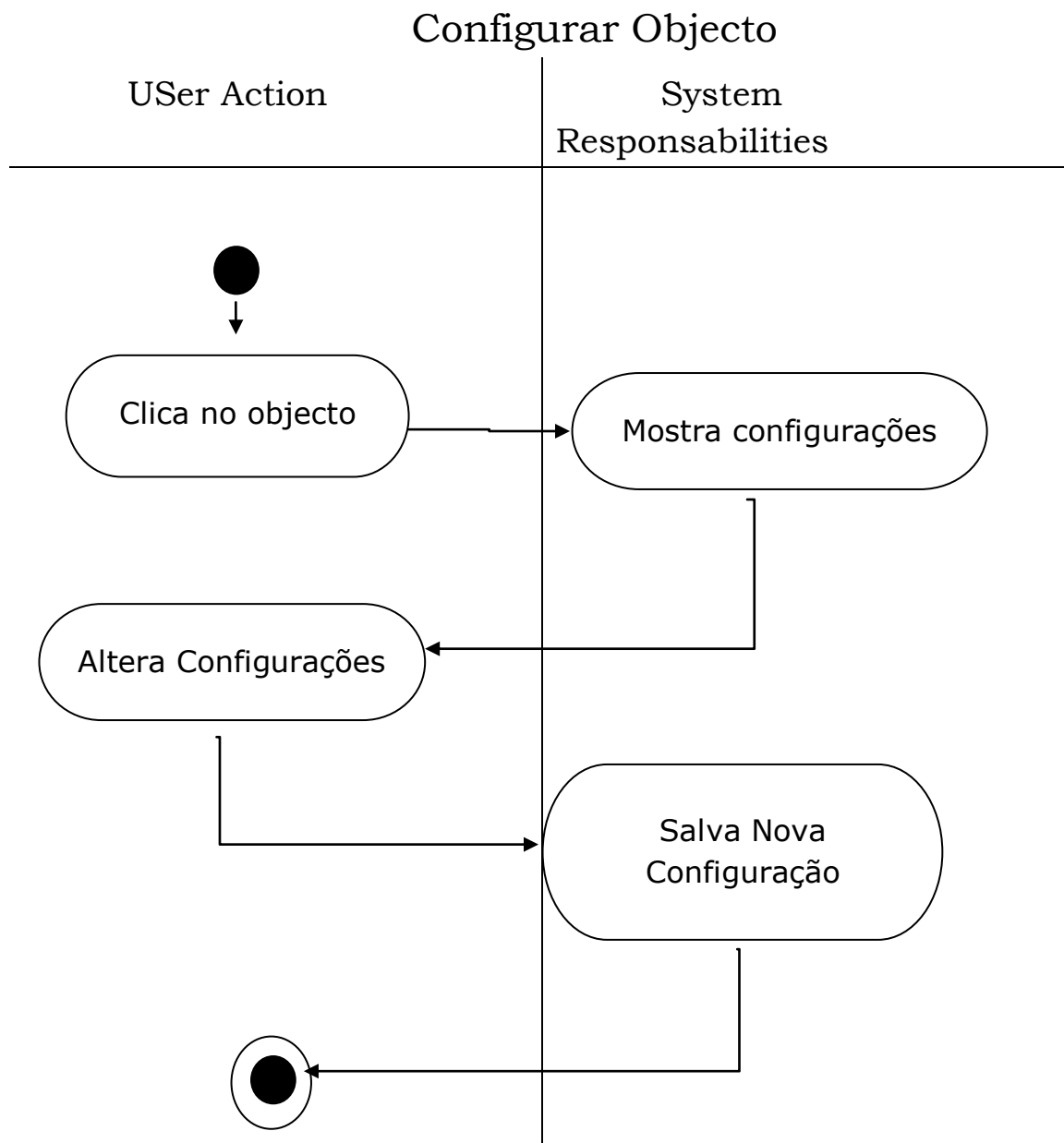


Figura 7 Diagrama de actividade Configurar Objecto

Um objectivo de todo o desenvolvimento, que foi reflectido nesta fase, foi a da intervenção do utilizador no menor número de interações necessárias para executar as tarefas pretendidas. Neste diagrama de actividades essa situação verifica-se, por exemplo, na ausência de confirmar a gravação das alterações.

Criar Rede

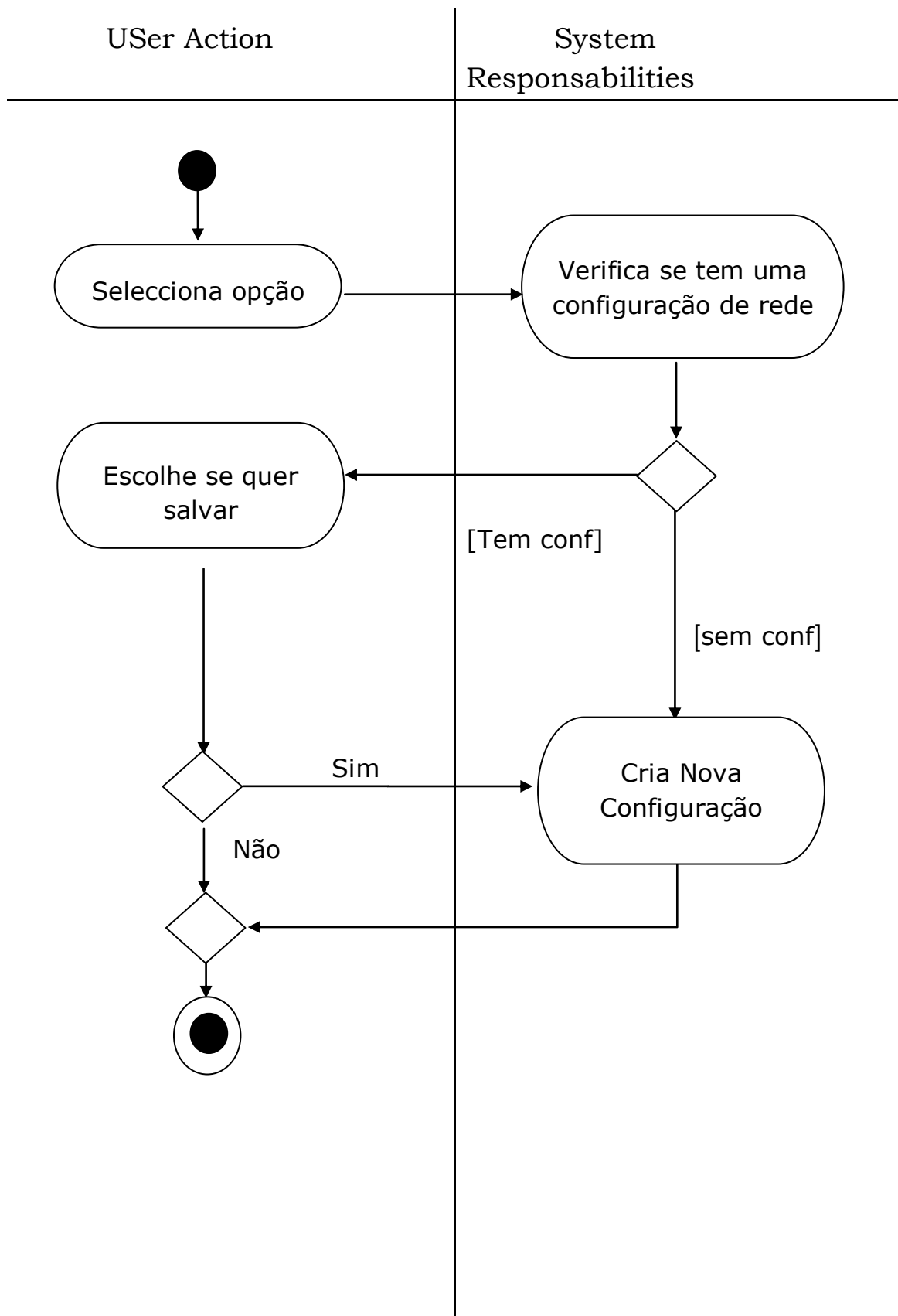


Figura 8 Diagrama de actividade Criar Rede

Mais uma vez resumimos ao máximo e simplificamos a interacção do utilizador com o sistema, sendo neste caso de uma só vez.

4.5. ARQUITECTURA DO SISTEMA

Na arquitectura do sistema são definimos os diferentes módulos utilizados e a sua comunicação.

Numa primeira abordagem de mais alto nível, podemos constatar que para o VND o funcionamento básico é ilustrado na Figura 9.

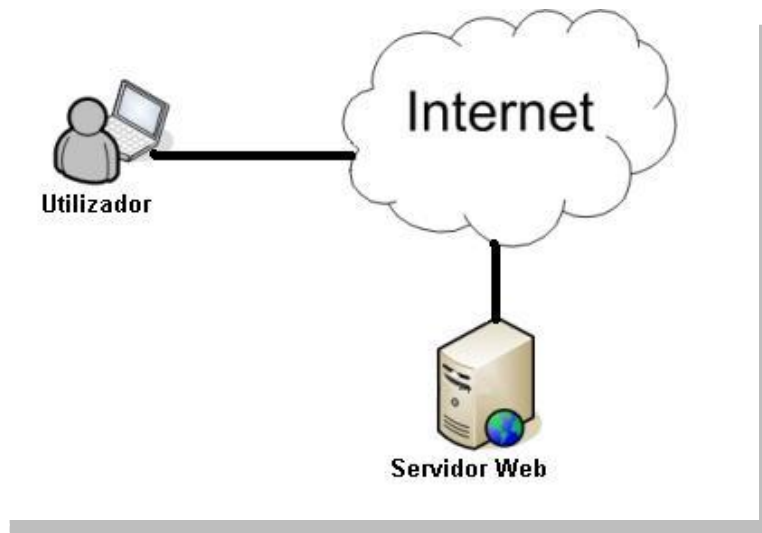


Figura 9 Arquitectura básica do VND

Analisando mais detalhadamente (Figura 9), podemos constatar que o VND está desenvolvido sobre duas camadas, sendo no entanto, no futuro extensível à terceira camada.

Assim e dentro de cada camada, podemos considerar o esquema descrito na Figura 10.

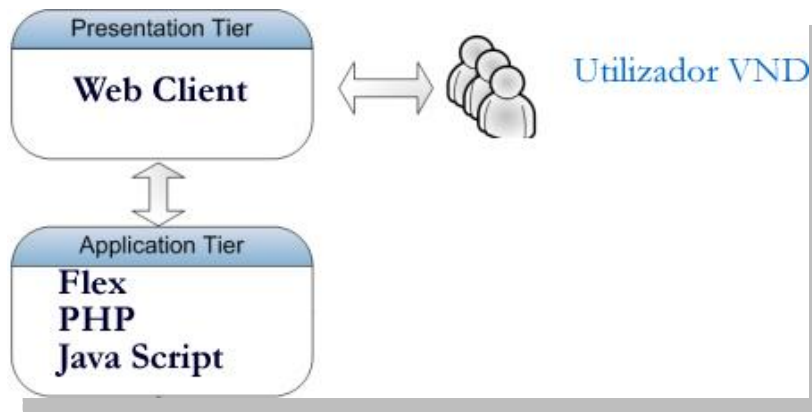


Figura 10 Arquitectura por camadas do VND

No que diz respeito à camada da aplicação, podemos verificar (Figura 10) todo o ambiente gráfico foi implementado utilizando as linguagens Flex [Adobe Flex, 2010], PHP [PHP, 2010] e Java Script foram utilizados para o tratamento de ficheiros.

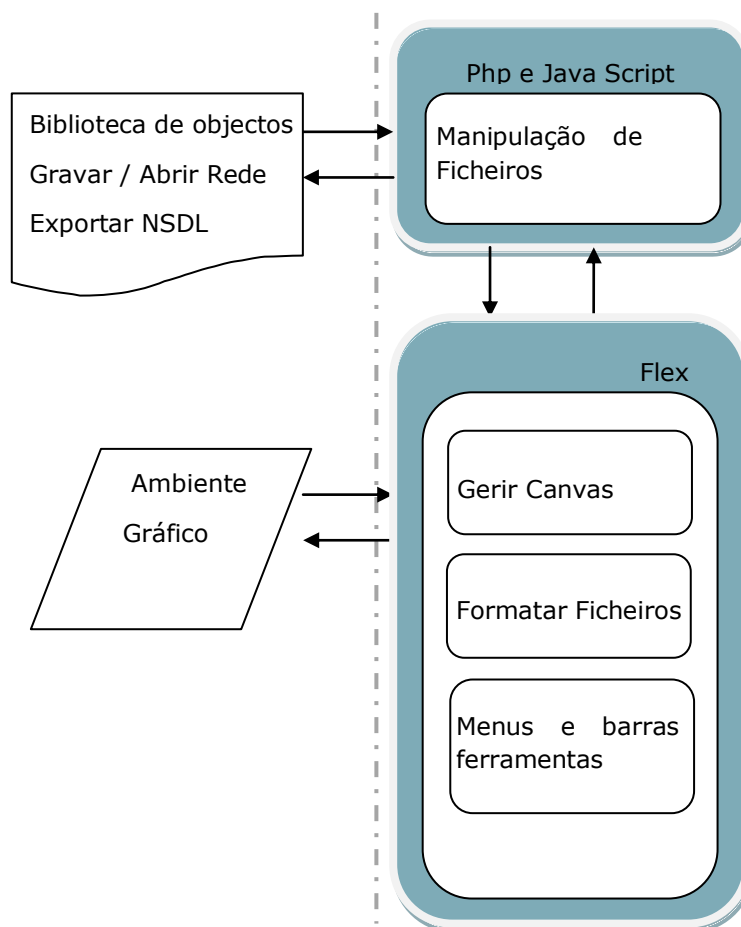


Figura 11 Módulos e Interação

Assim e como podemos verificar na Figura 11 no VND a comunicação com o utilizador através do ambiente gráfico é da responsabilidade do Flex, enquanto tudo o que tem a ver com manipulação de ficheiros (Salvar, Carregar, etc.) é realizado pelo PHP.

No Flex o desenvolvimento foi efectuado de forma a permitir a expansão, pelo que foi optado gerir de forma diferenciada:

- Os canvas;
- A formatação de ficheiros, e;
- Menus e barra de ferramentas.

A gestão dos canvas tem a responsabilidade das áreas de interacção com o utilizador (por exemplo, a área de desenho da rede). Na gestão de canvas definimos o que é permitido fazer nessa área, tal como. arrastar objectos da área da biblioteca de objectos.

A formatação de ficheiros, permite definir qual a formatação e estruturação dos diferentes ficheiros.

A responsabilidade dos menus e barras ferramentas define o que é apresentado na área respectiva área fazendo a associação entre o que o utilizador pretende fazer e a reacção do sistema, p.e. clicar no icone e isso resultar no zoom da área de desenho.

4.6. PROTÓTIPOS ABSTRACTOS CANÓNICOS

Os Protótipos Abstractos Canónicos (PAC) documentam o interface gráfico do sistema e a interacção com o utilizador [Constantine, 2003].

Para o VND optámos por criar um aspecto que conseguisse ser fácil de entender aos utilizadores mais familiarizados com este tipo de ferramentas, mas também a novos utilizadores, que fosse bastante simples e fácil de utilizar, mas que conseguisse cumprir com o que pretendemos da ferramenta.

Dessa forma, o PAC para a aplicação desenvolvida é ilustrado na Figura 12

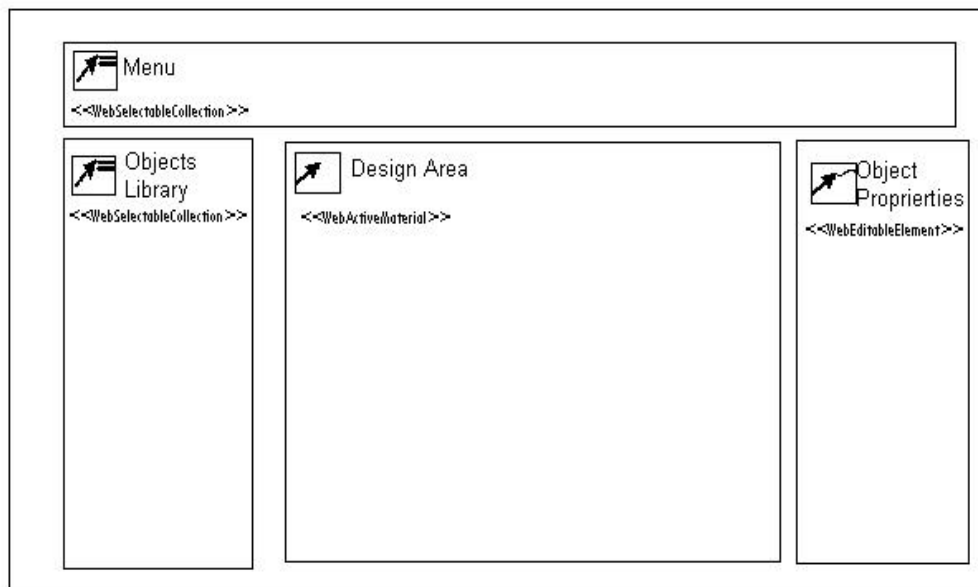


Figura 12 Protótipo Abstracto Canónico do VND

4.7. OPÇÕES TECNOLÓGICAS

Em qualquer projecto as opções tecnológicas estão directamente relacionadas e reflectem as funcionalidades que são pretendidas para a aplicação a ser desenvolvida.

Neste projecto pretendemos oferecer uma maior portabilidade à aplicação a ser desenvolvida, sendo preferível desenvolver em ambiente Web, o que condicionou imediatamente as opções tecnológicas para ferramentas de desenvolvimento.

Dessa forma, foram estudadas as linguagens de programação que permitiam fazer um interface Web, com as funcionalidades pretendidas, num produto que fosse possível expandir, sendo que as opções mais naturais seriam JAVA Swing [Java 2010] ou Adobe Flex [Adobe Flex 2010].

Deste modo, e após verificarmos que seriam os dois produtos que seriam mais indicados para o desenvolvimento, foi efectuado um levantamento comparativo sobre como poderia ser mais vantajoso utilizar cada uma das linguagens, de forma a escolher uma das linguagens.

Algumas das vantagens de Flex que levaram à sua escolha foi o facto de permitir que praticamente a mesma aplicação conseguisse ser usada numa aplicação em servidores Web, ou passada para

aplicação local, oferecendo uma maior interoperabilidade entre os diversos Web Browsers. Dada a quantidade de material disponível, apresentava-se como mais fácil de progredir rapidamente na aprendizagem. Tendo em conta o apresentado, foi escolhido o Flex como linguagem principal de desenvolvimento do projecto.

Por outro lado, a possibilidade do utilizador conseguir utilizar uma lista de objectos criada por si, conseguir gravar e carregar os cenários de redes por si criadas ,e uma vez que o NSDL já é baseada em XML, utilizou-se XML para todos os ficheiros de leitura ou gravação, nomeadamente:

- O cenário de rede criada;
- Lista de configuração de objectos, e;
- A exportação para NSDL.

Com esta informação, passou a existir a necessidade de operações com ficheiros, que o Flex não executa, pelo que tivemos que utilizar também o PHP [PHP, 2010] como linguagem de apoio para o tratamento de ficheiros.

Desta forma, ficamos com a decisão de utilizar o Flex e PHP para construir o VND.

4.8. ASPECTOS DE CONFIGURAÇÃO

O VND foi criado com a intenção de ser uma ferramenta versátil e adaptável, deixando, ao contrário das ferramentas actuais, muita liberdade ao utilizador para customizar a ferramenta consoante as suas necessidades.

A aplicação está criada de forma tão flexível que permite o utilizador parametrizar totalmente as configurações que caracterizam cada objecto. Para parametrizar o VND, o principal recurso é o ficheiro de configuração XML para a descrição da biblioteca de objectos. Com este ficheiro, o utilizador pode criar outros objectos de acordo com as suas preferências e necessidades.

Deste modo, se o utilizador quiser criar directamente o XML com a sua lista personalizada de objectos, poderá fazê-lo bastando para isso tomar em consideração as regras de criação do XML.

Transversalmente no XML é utilizado um elemento, chamado de <par>, utilizador para declarar um parâmetro.

Dentro da declaração de cada parâmetro deve-se definir os seguintes campos:

- *name*, para definir o nome do parâmetro;
- *type*, escolha de um tipo de dados, por exemplo (p.e.), string;
- *syze*, o tamanho do campo;
- *mandatory*, se é um campo de preenchimento obrigatório.

Dentro dos tipos de armazenamento permitidos no VND temos:

- *String*, para texto escrito, o mais comum;
- *Boolean*, para selecção entre verdadeiro e falso, p.e. se está activo;
- *Options*, selecção de um valor de uma lista fornecida, p.e. diferentes larguras de banda;
- Instância, se é do tipo instância, e;
- *Object* se é do tipo object, utilizado nas ligações.

Considere a Figura 13 que apresenta a estrutura geral da linguagem de configuração XML utilizada. Na linguagem apresentada podemos distinguir três grandes grupos de objectos que podem ser utilizados para criar cenários de rede no VND:

- As **instâncias**, que são abstracções que podem ser utilizadas nas definições dos objectos (a seguir), e que podem ocorrer uma ou mais vezes dentro desses objectos (tais como interface ou protocolo);
- Os **objectos**, que representam os possíveis tipos de abstracções gráficas que podem ser incluídas em um cenário de rede, e que possuem características individuais tais como nome, id, etc. Da mesma forma, nos objectos são definidas as respectivas instâncias que podem ser declaradas dentro do objecto, e;
- As ligações (**links**) que são basicamente as ligações entre os objectos, e que podem possuir características e/ou instâncias.

```
inputxml.xml
<nsdlObjects>
  <instances>...
  <objects>...
  <links>...
</nsdlObjects>
```

Figura 13 Estrutura do XML de configuração do VND

A Figura 14 ilustra a composição do ficheiro de configuração XML.

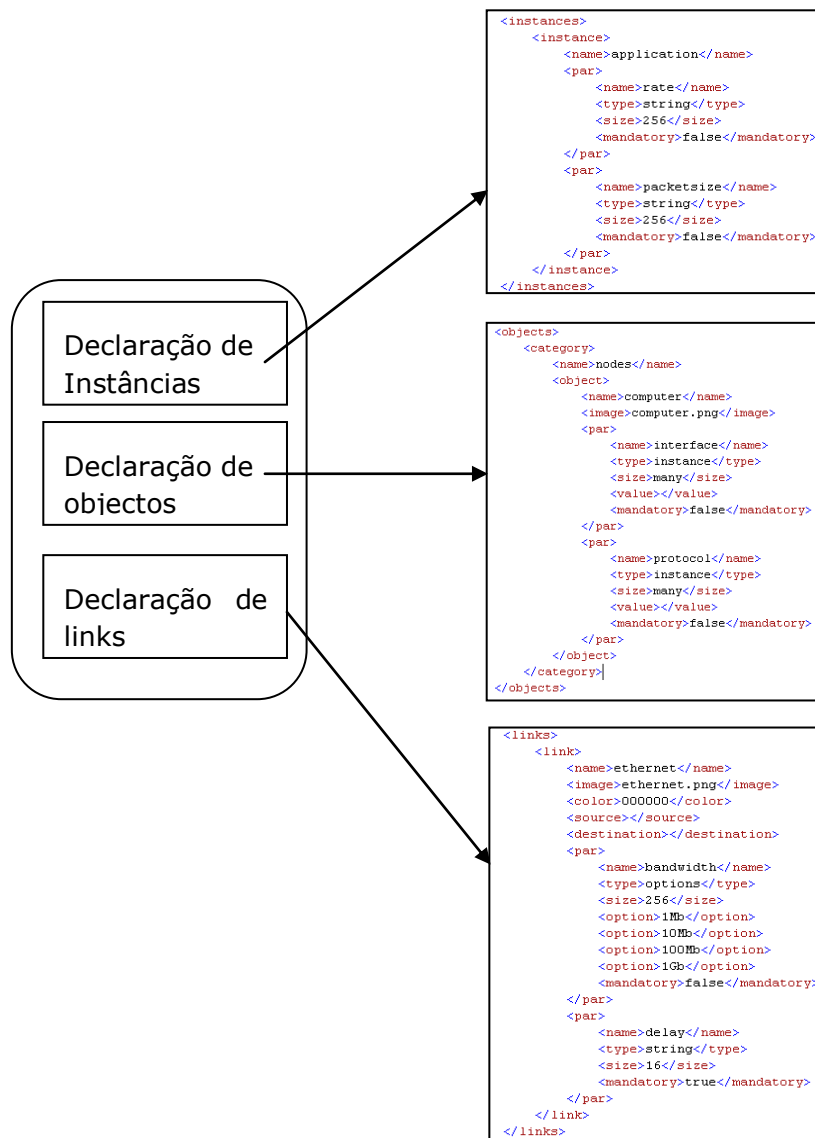


Figura 14 Formato do XML de declaração de objectos

O processo de armazenar e voltar a carregar o cenário de rede também é feito através de um ficheiro XML, o qual o utilizador não necessita editar. A edição manual desse ficheiro é desaconselhável por ser uma representação proprietária do VND.

4.9. APRESENTAÇÃO DO SISTEMA

A composição gráfica do VND foi criada com base em duas cores o azul e o cinzento, de forma a proporcionar um ambiente leve, e ao mesmo tempo contemporâneo.

Com base nas especificações existentes e resultantes dos protótipos canónicos abstractos e das funcionalidades pretendidas nos requisitos, propusemos a interface ilustrada na Figura 15.

Para consultar o projecto, pode-se ir ao site <http://apus.uma.pt/~a2007702/>.

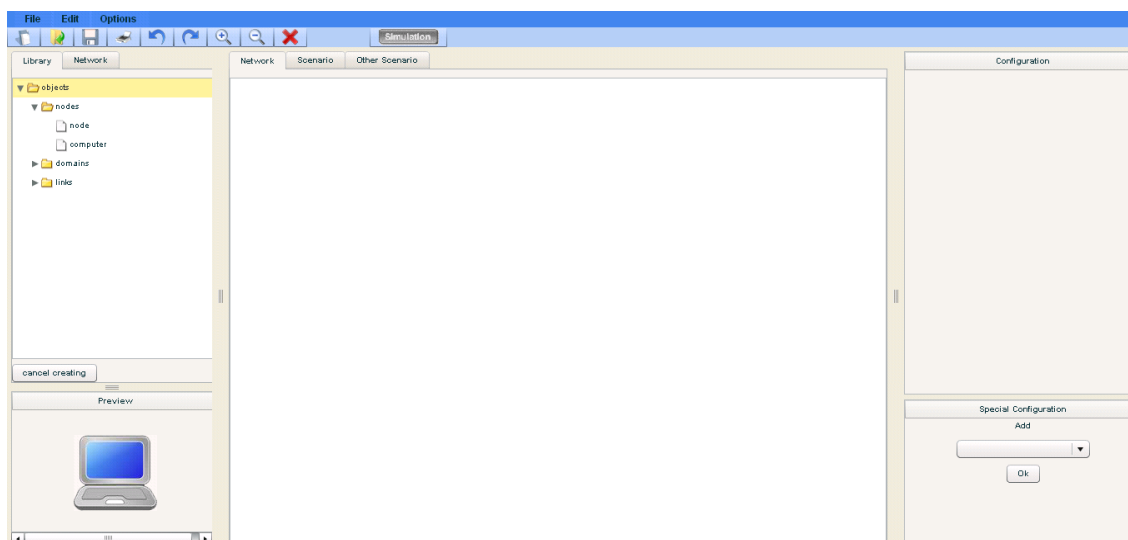


Figura 15 Aspecto gráfico genérico do VND

Para além do normal enquadramento, com menu, área de desenho e configuração, tivemos que tomar algumas opções, que dificultaram o trabalho, mas que contribuíram para a criação de uma ferramenta mais versátil:

- Na área da esquerda, disponibilizamos a biblioteca de forma hierárquica, que disponibiliza uma lista de objectos, com diversos sub-tipos e representa todos os objectos de uma forma fácil de ler e de navegar, tal como ilustrado na Figura 16;

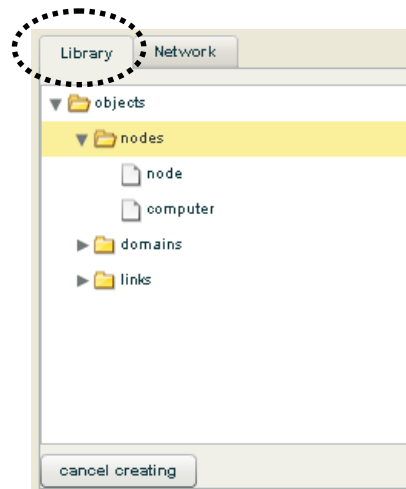


Figura 16 Exemplo de navegação na biblioteca

- Da mesma forma, foi criado um separador que permite na área da esquerda ter a lista (dinâmica) de objectos criados. Nessa lista o utilizador pode seleccionar um objecto já existente e ver as suas características e navegar dentro do objecto nas suas propriedades e abstracções. Por exemplo, ele pode seleccionar um interface em um computador, tal como mostrado na Figura 17;

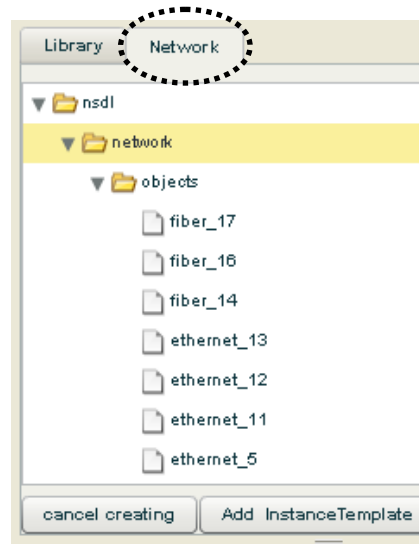


Figura 17 Exemplo de lista de objectos já criados

- Na zona da direita, no topo, aparecem as características (propriedades) dos objectos, sendo possível configurar também a sua posição de forma manual, tal como mostrado em seguida na Figura 18;

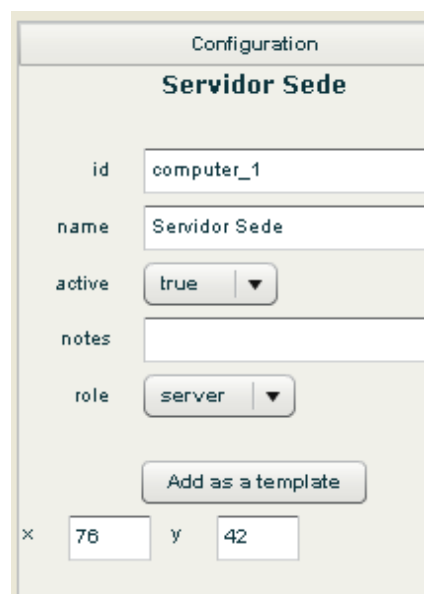


Figura 18 Área de configuração de objecto

- Ainda na zona da direita, em baixo, na zona de configurações especiais, pode-se adicionar abstracções (tais como aplicações ou interfaces) que podem ocorrer uma ou diversas vezes em um objecto. Esta é uma das características que mais

distinguem este software de outros, tal como ilustrado na Figura 19, e;

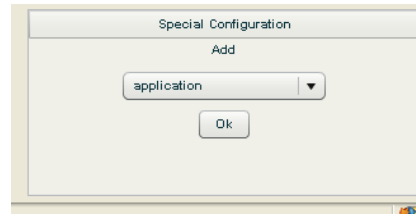


Figura 19 Adicionar configurações especiais

- O tamanho de todas as áreas apresentadas anteriormente é ajustável, bastando ao utilizador arrastar o separador que as limita.

Com o menu criado será permitido ao utilizador aceder directamente às funções existentes, nomeadamente:

- Criar novos projectos;
- Abrir projectos anteriormente guardados;
- Gravar projecto;
- Importar biblioteca de objectos (do servidor ou do cliente);
- Exportar para NSDL;
- Imprimir;
- Retroceder (undo);
- Avançar (Redo);
- Zoom in;
- Zoom out;
- Criar Templates;
- Apagar Templates;
- Criar Views;
- Apagar Views;
- Criar Simulações, e;
- Apagar Simulações.

Foi também adicionado o menu gráfico como forma de facilitar ao utilizador o uso de ferramentas usadas com mais frequência, ou que o utilizador através de ícones conseguisse perceber directamente o seu uso.

A aplicação VND sendo completamente personalizável permite acrescentar mais funcionalidades no futuro. Portanto, todo o desenvolvimento foi realizado para permitir com pouco esforço atingir esse objectivo.

4.10. CONCLUSÃO

Neste capítulo apresentamos os principais aspectos de modelização e implementação da ferramenta VND.

Inicialmente, foram apresentados os requisitos que tiveram como base a evolução do trabalho, com os mesmos foram elaborados os casos de utilização e posteriormente os diagramas de actividade.

A apresentação dos Protótipos Abstractos Canónicos finalizou definindo em linhas gerais qual seria a disposição gráfica que a ferramenta teria.

Posteriormente, os principais aspectos de implementação do VND foram discutidos, o que envolveu, as opções que foram tomadas e porquê foram tomadas.

Uma das principais contribuições do trabalho realizado também foi apresentada, ou seja, a possibilidade de personalização da aplicação VND através da utilização do ficheiro de configuração XML.

Da mesma forma, foram apresentados os principais aspectos sobre as funcionalidades e aspecto gráfico do VND implementados.

No final do capítulo descrevemos a apresentação e funcionalidades dadas ao utilizador pelo sistema.

No próximo capítulo apresentamos alguns estudos de caso.

5. ESTUDO DE CASO

5.1. INTRODUÇÃO

Neste capítulo apresentamos dois estudos de caso que permitem verificar algumas das funcionalidades do VND.

Inicialmente demonstramos a criação da rede base de estudo. A partir dessa estrutura de rede base modelada, propomos a modelação de topologias mais complexas e realísticas nos dois estudos de caso realizados.

5.2. DESCRIÇÃO DA ESTRUTURA BASE

Nesta modelação decidimos descrever a estrutura de uma rede de uma empresa com uma sede e dois sites remotos [referencia ao trabalho dos alunos de TAR]. A ligação entre os sites e a sede é realizada através de topologia em anel interligando os respectivos routers. Da mesma forma, é considerado que a sede possui um computador e um servidor, o primeiro site remoto possui um computador e um servidor, e o segundo site remoto possui somente um PC.

Para criarmos a topologia, temos que arrastar cada objecto pretendido da biblioteca de objectos para a área de desenho.

Para criar as ligações devemos, na biblioteca de objectos e dentro do separador de ligações, seleccionar a ligação pretendida e posteriormente conectar os objectos dessa ligação, a representação da nossa biblioteca de objectos é demonstrada na Figura 20.

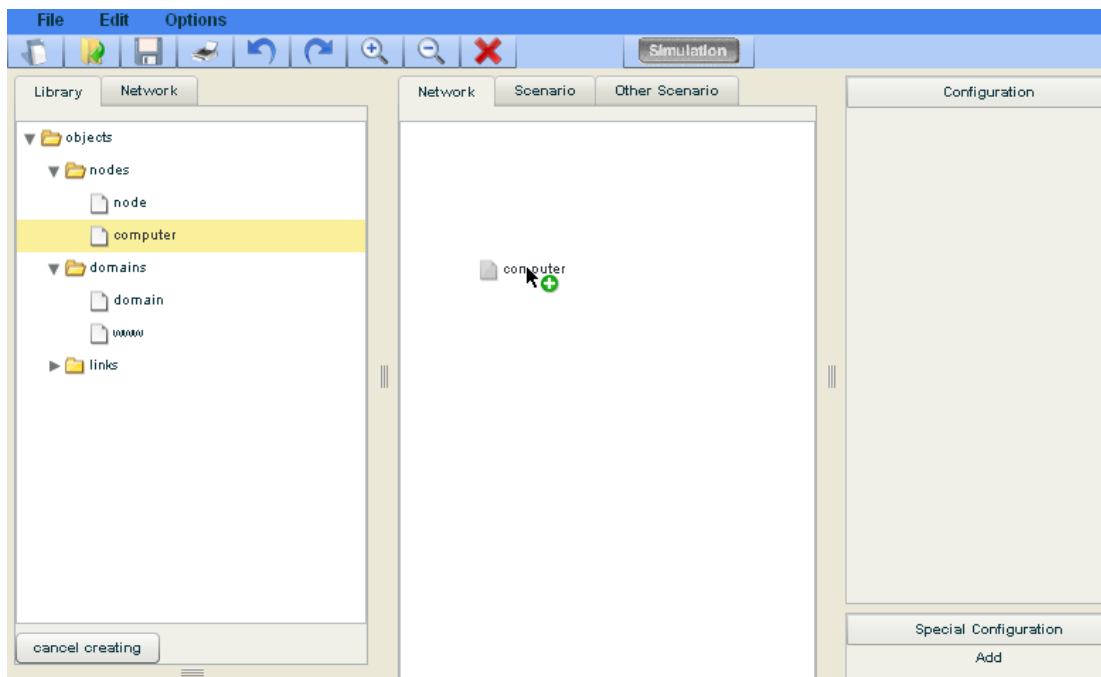


Figura 20 Arrastar objectos e criar topologia

Na representação dos objectos gráficos utilizados no cenário de rede a ser descrito decidimos aplicar uma biblioteca de objectos específicos para a representação dos routers, de um computador (com opção de utilizarmos um servidor), e dois tipos de ligações. Para a especificação dos tipos de objectos gráficos adoptados, utilizamos a linguagem baseada em XML para a configuração do VND.

A representação do cenário de rede base é ilustrada na Figura 21.

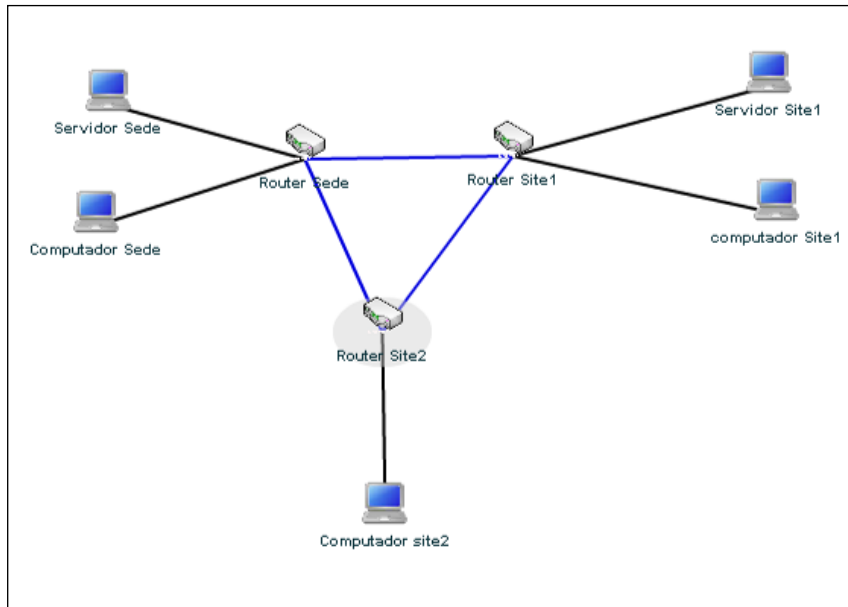


Figura 21 Rede de base para os casos de estudo

Como apenas uma abstracção "computador" foi definida no ficheiro de configuração XML, é necessário uma forma de distinguir na aplicação VND quais os objectos que irão representar um servidor e quais aqueles que irão representar um terminal. Portanto, na descrição do objecto "computador" foi adicionado um atributo "role" que permite o utilizador distinguir o objecto criado como "servidor" ou "terminal".

A Figura 22 ilustra a configuração dos objectos criados em um cenário de rede no VND. Em particular, como é definido o papel do objecto computador (servidor ou terminal). Para a configuração de um objecto, o mesmo deve ser seleccionado para que os seus respectivos parâmetros sejam disponibilizados para a edição na coluna a direita.

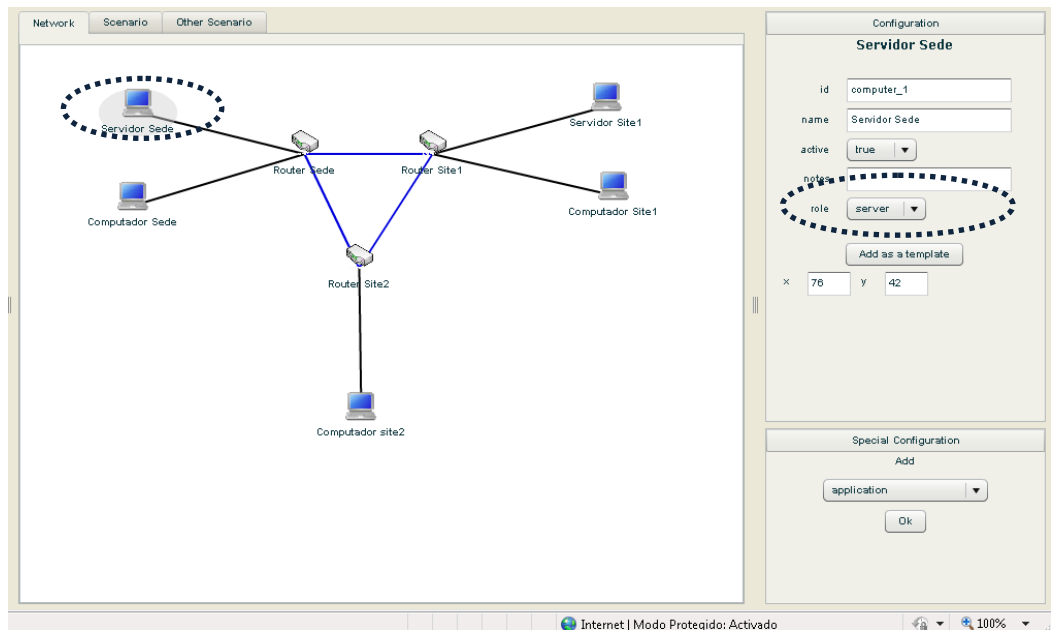


Figura 22 Edição do de um computador

Depois de realizarmos a configuração para todos os objectos do cenário, obtemos o cenário base para a realização dos estudos de caso a serem apresentados neste capítulo.

5.3. PRIMEIRO ESTUDO DE CASO

No primeiro estudo de caso, decidimos caracterizar o cenário de redes adoptado com a seguinte configuração:

1. A ligação entre os sites e a sede é feita através de uma ligação em fibra duplex a 1Gb/s;
2. A ligação interna em cada site, ou seja entre os routers e os servidores ou computadores é feita por uma conexão duplex em cabo Ethernet a 100Mb/s;
3. O servidor da sede tem duas aplicações, uma Ftp e outra http e o servidor do primeiro site tem uma aplicação Pop, e;
4. As restantes configurações base mantêm-se.

A seguir apresentamos passo-a-passo a configuração do cenário de acordo os critérios apresentados anteriormente.

1.º PASSO - A ligação entre os sites e a sede é feita através de uma ligação em fibra a 1Gb/s.

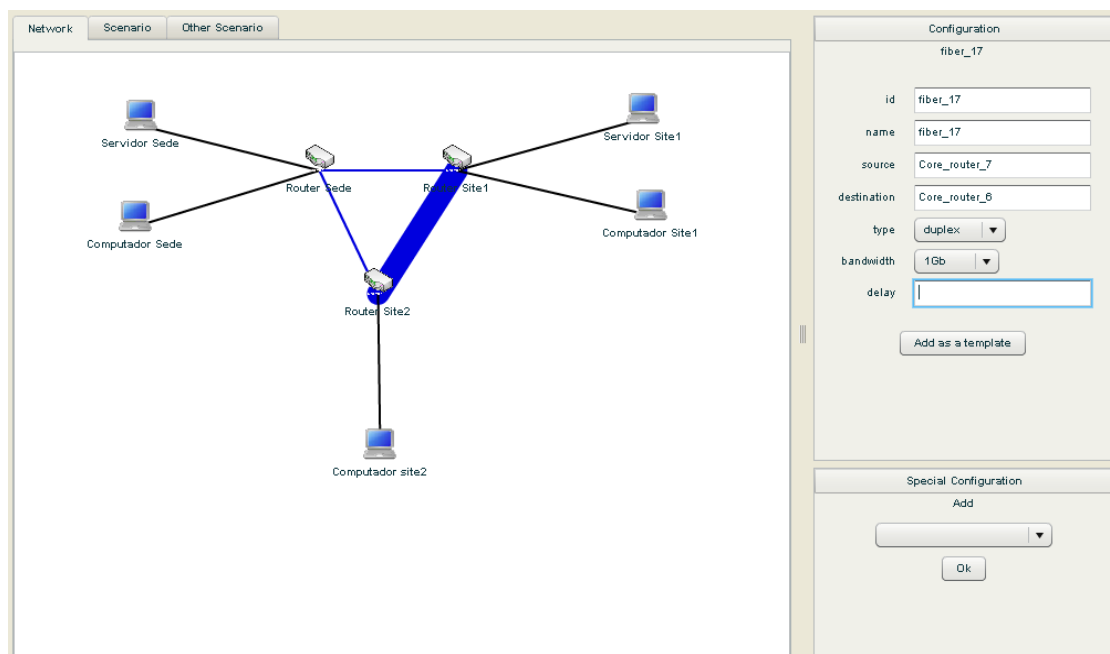


Figura 23 Alterar configuração fibra

Como podemos verificar na Figura 23 de forma a editar a configuração desejada, basta apenas clicar sobre a ligação em causa. Neste momento, a ligação seleccionada é destacada na interface do VND, e os seus respectivos parâmetros são disponibilizados para a edição na coluna a direita do ecrã. Neste momento, o utilizador deve apenas editar os valores dos parâmetros.

2.º PASSO - A ligação interna em cada site, ou seja entre os routers e os servidores ou computadores é feita por Ethernet a 100Mb/s.

Da mesma forma como procedemos para a configuração da ligação anterior, também devemos seleccionar ligação que desejamos editar, ou seja, a ligação entre o servidor e o router da sede. Após a sua selecção, podemos alterar os valores de configuração na coluna a direita, como ilustrado na Figura 24.

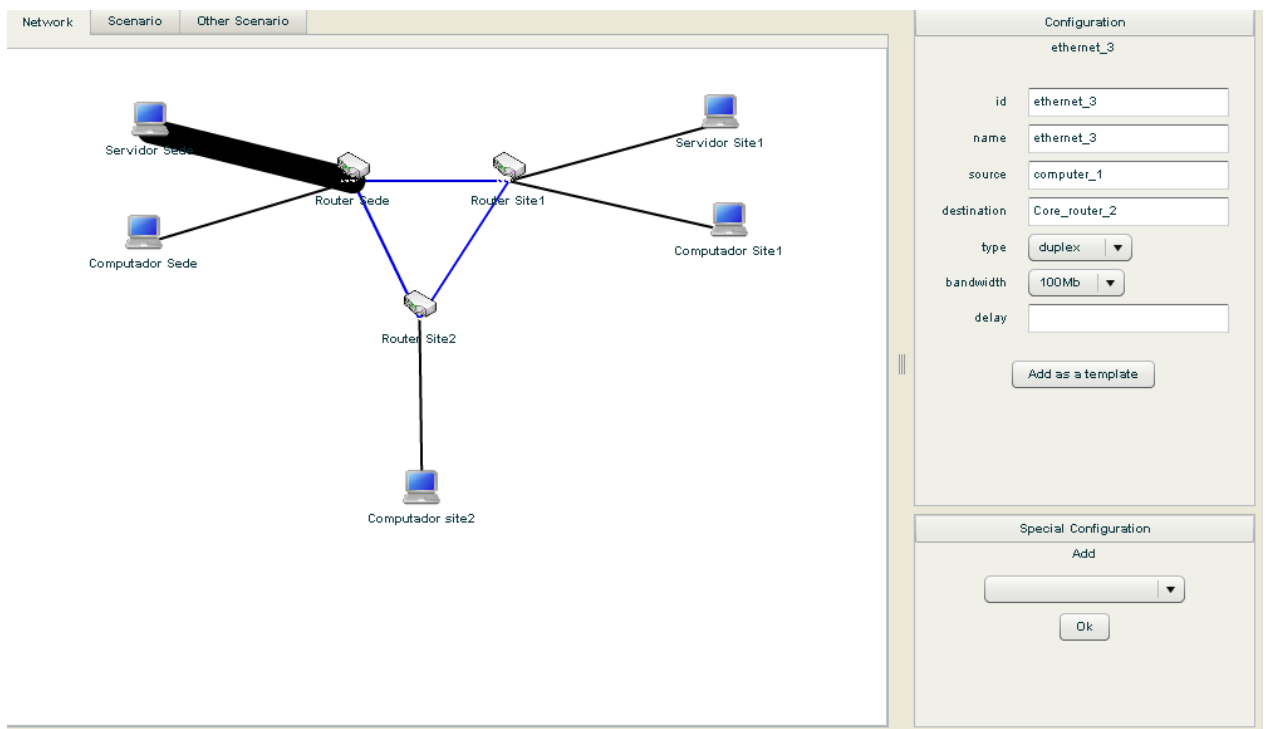


Figura 24 Editar configuração Ethernet

Se o utilizador desejar, o VND ainda permite distinguir os diferentes tipos de ligação pela cor. Neste exemplo definimos que a

cor preta definia rede Ethernet e a azul definia fibra óptica. Como apresentado, ainda definimos que por exemplo na ligação Ethernet os valores a serem configurados seriam diferentes da ligação de fibra óptica. Todas estas opções foram especificadas no ficheiro de configuração XML para a biblioteca de objectos.

3.º PASSO - O servidor da sede tem duas aplicações, Ftp e http; e o servidor do primeiro site tem uma aplicação, Pop.

Uma aplicação pode ser considerada como uma abstracções que pode ou não ocorrer em um objecto diversas vezes. A definição de uma aplicação é declarada no ficheiro de configurações XML.

Na nossa solução, oferecemos ao utilizador a liberdade para declarar essas abstracções no ficheiro de configuração XML. Dessa forma, a aplicação VND é capaz de se adaptar de forma dinâmica e automática para disponibilizar essas abstracções na biblioteca de objectos da interface e os seus respectivos parâmetros de configuração. De facto, a adaptação automática do VND consoante o ficheiro de configuração XML foi um dos maiores desafios em termos de desenho e implementação da aplicação.

A configuração das abstracções (tais como interface ou aplicação) adicionais em um objecto é realizada na parte inferior da coluna a direita da interface. Essa zona é chamada de "Special Configuration". Para adicionar uma abstracção ao objecto, o utilizador deve seleccionar a opção pretendida e clicar em "ok" (Figura 25). Após a inclusão, as respectivas abstracções aparecem automaticamente abaixo do objecto em causa na estrutura hierárquica da rede, apresentada no separador "network" da coluna a esquerda.

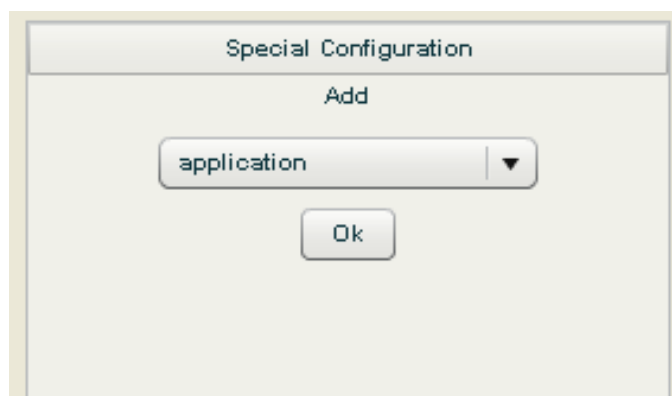


Figura 25 Adicionar configurações especiais

Da forma como apresentado anteriormente, procedemos com a inclusão de duas aplicações ao servidor da sede e uma ao servidor do site 1.

Após adicionar as respectivas aplicações aos objectos, devemos configurar cada aplicação consoante os requisitos identificados. Para tal, devemos seleccionar cada objecto em causa no separador de rede, e a partir deste seleccionar a sua respectiva aplicação. Uma vez seleccionada a aplicação a configurar, os seus parâmetros a serem editados são disponibilizados na coluna a direita (Figura 26).

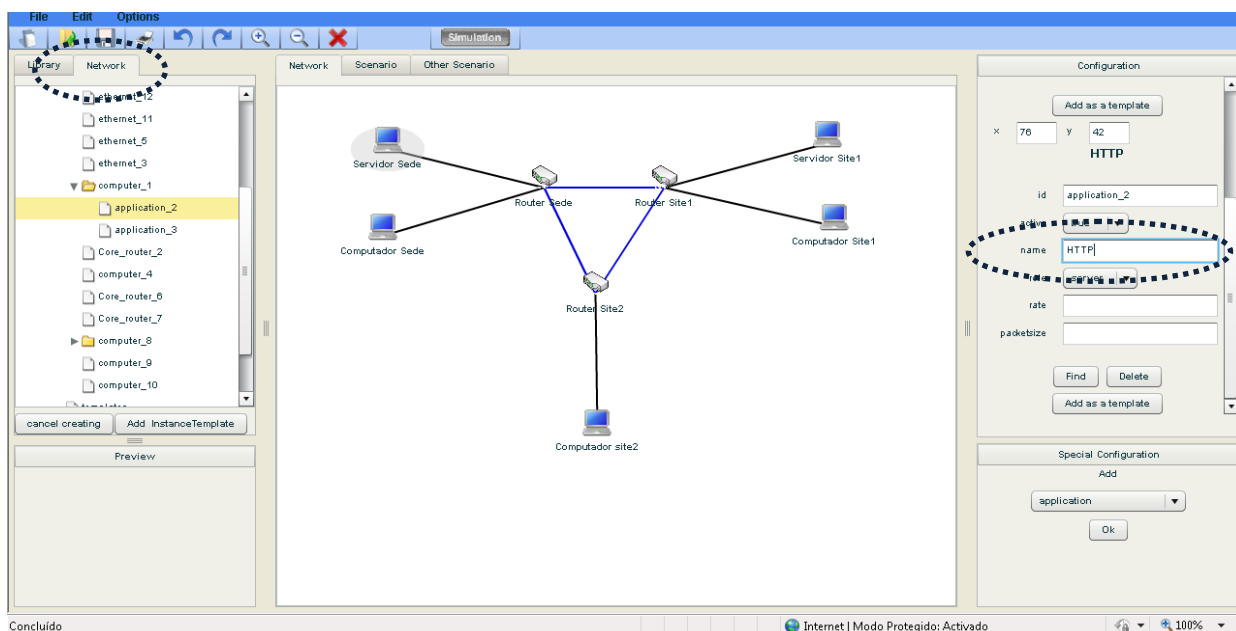


Figura 26 Alterar dados de uma aplicação

Para a configuração de todas as aplicações, devemos proceder de acordo com o apresentado anteriormente.

Após completarmos todos os critérios de configuração, é importante verificar se a salvaguarda do documento decorre como pretendido.

5.º Passo – Verificar o trabalho guardado.

Nesta etapa, verificamos como o trabalho realizado foi salvaguardado no ficheiro XML.

Com este exemplo prático apresentamos alguns excertos do ficheiro XML e os detalhes das configurações que realizamos.

Para uma consulta mais detalhada, o ficheiro XML é apresentado completamente nos ficheiros do projecto.

```
- <objects>
  <type>links</type>
  <id>ethernet</id>
  <name>ethernet_13</name>
  <source>Core_router_7</source>
  <destination>computer_10</destination>
  <isMOtherTemplate>false</isMOtherTemplate>
  <isChildTemplatedOne>false</isChildTemplatedOne>
  <motherTemplate />
  <lineColor>0</lineColor>
  <beginX>328.5</beginX>
  <beginY>215</beginY>
  <lineLength>330.5</lineLength>
  <lineHeight>358</lineHeight>
+ <par>
+ <par>
+ <par>
+ <par>
+ <par>
+ <par>
+ <par>
- <valueObjArr>
  <id>bandwidth</id>
  <name>2</name>
  <haschanged>false</haschanged>
  <IsCombo>true</IsCombo>
  <combeValue>100Mb</combeValue>
</valueObjArr>
+ <valueObjArr>
</objects>
```

Figura 27 Ligação exportada

No excerto apresentado na Figura 27 podemos ver como os dados associados a uma ligação foram salvos no ficheiro XML do VND. Nesta representação podemos ver toda a informação que é

necessária para reconstruir o objecto, nomeadamente a posição e valores de configuração no caso da ligação.

5.4. SEGUNDO ESTUDO DE CASO

Para demonstrar as capacidades do VND seleccionamos como segundo caso de estudo o trabalho realizado com a ferramenta por Jorge Sousa, aluno da Universidade, que por sua vez utilizou um documento da disciplina de Tecnologias Avançadas de Redes (TAR), leccionada pelo professor Paulo Sampaio, dos alunos Fernando Rodrigues e Nélio Sousa.

Foi seleccionado um cenário de rede onde testamos alguns requisitos de QoS, a figura 28 demonstra a topologia adoptada.

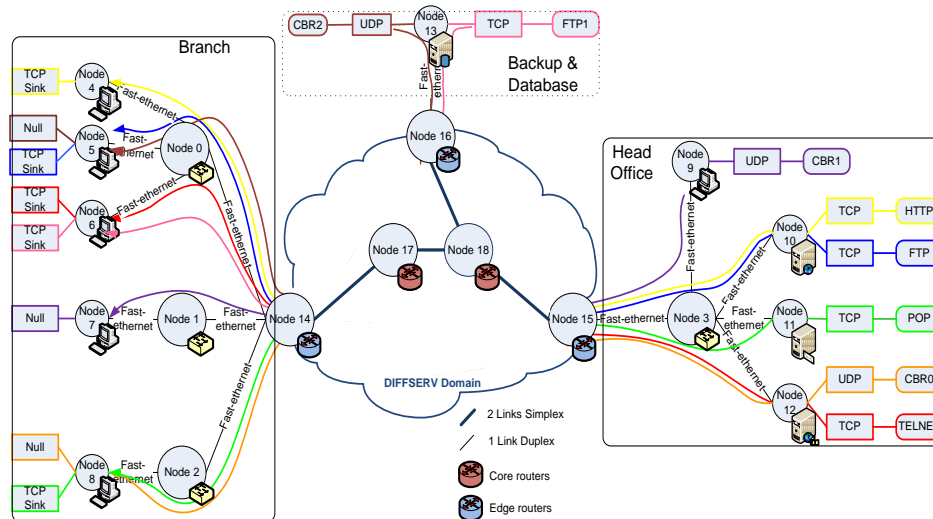


Figura 28 Cenário de Rede

O cenário de rede apresentado na Figura 28 representa a rede de uma empresa, interligando três LAN's (Local Area Networks), o escritório sede, um local remoto e um local de backups e bases de dados (redundância).

Para usufruir de objectos que permitam Diffserv, deverá ser utilizado o ficheiro de biblioteca de objectos alterado, enviado no Anexo V.

Para efectuar este caso de estudo consideraremos como condicionantes à configuração que:

1. Cada uma das LANs tem implementado uma rede Fast Ethernet 100Base-T, que permite ligações até 100MBps.
2. O core do backbone da WAN (Wide Area Network) é gerido por operador public, que vai disponibilizar o acesso através de uma conexão xDSL e a ligação ao gabinete da sede é feita por um access link de 20Mbps, sendo que as restantes (local remote e redundância) são feitos a 10Mbps;
3. Cada ligação entre os core routers e os core edge routers é feita a 12Mbps.
4. Nesta implementação os core routers irão providenciar QoS através da utilização de DiffServ [RFC2475 1998]. Para atingir este objectivo definimos três classes de serviços:

BE – Best Effort;

AF –Gold (Assured Forward);

AF Platinum.

A figura 29 descreve os tráficos existentes e a respectiva classe de serviço (Class), o Jitter e o Bitrate. Como podemos verificar na figura 29, definir diferentes classes de tráfico originará também diferentes fluxos de tráfico.

Application	Source -> Destination Traffic	Class	Jitter (ms)	Bitrate (Mbps)
POP/Mail	1->8	Best- Effort	500	0.1
POP/Mail-FTP	3->6	Best- Effort	500	.01
CBR 0/Video-on-Demand	2->8	AF – Platinum	9	4.5
CBR 1/Video RT/HDTV	->7	Best- Effort	6	4.5
CBR 2/ Video-on-Demand	3->5	AF - Platinum	10	4.5
TELNET	2->6	AF – Gold	60	0.5
FTP	0->5	AF – Gold	500	2
HTTP- Exponential traffic	0->4	AF – Gold	50	2.5
FTP1	3->6	Best- Effort	500	1.5

Figura 29 Links e classes de tráfego

Da mesma forma que fizemos para o primeiro caso de estudo, em seguida apresentamos pormenorizadamente como configuramos a rede para cumprir com cada um dos passos.

1.º PASSO - Cada uma das LANs tem implementado uma rede Fast Ethernet 100Base-T, que permite ligações até 100MBps.

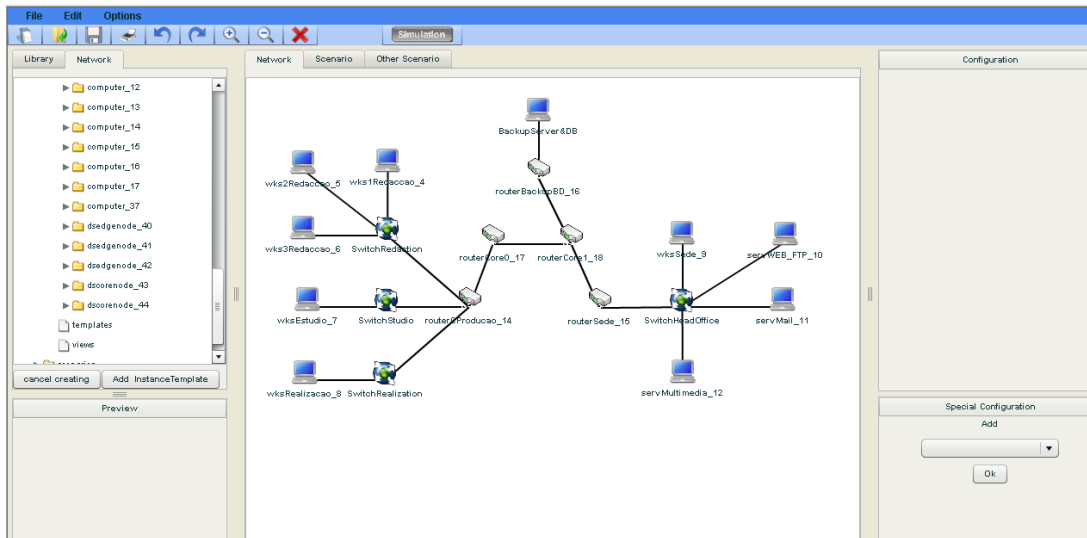


Figura 29 Vista geral da rede

Com primeiro passo criamos a rede que podemos ver na figura 29 como ficou no VND.

Na figura 30 verificamos como se procedeu para alterar as configurações das ligações Ethernet

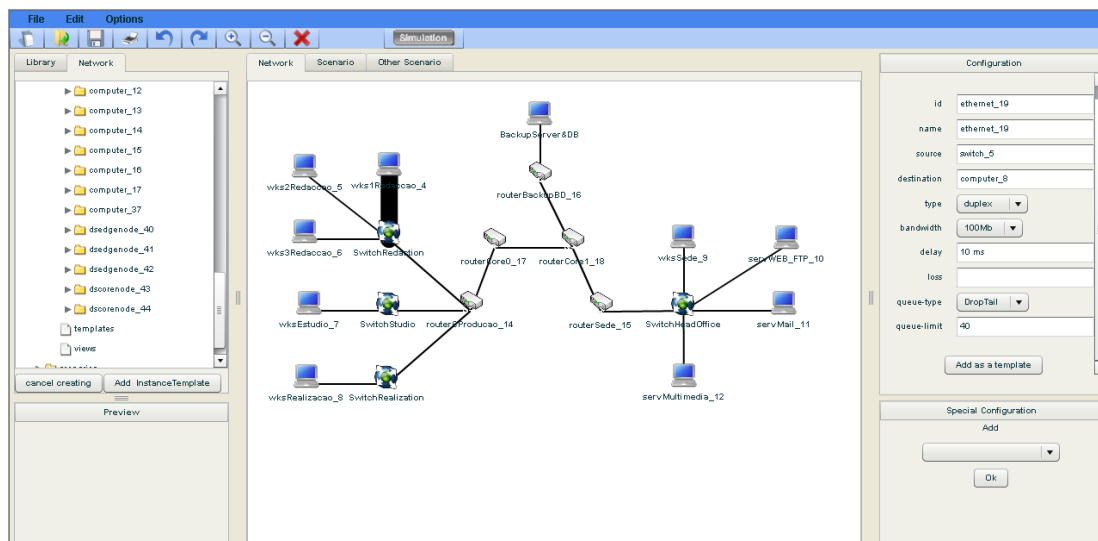


Figura 30 Vista geral da rede

2.º PASSO - O core do backbone da WAN (Wide Area Network) é gerido pelo operador, que vai disponibilizar o acesso através de uma conexão xDSL, sendo que a ligação ao gabinete da sede é feita por um access link de 20Mbps e as restantes (local remote e redundância) são feitas a 10Mbps;

Efectuar este passo implicou alterar o tipo de ligação e débitos, para o fazer tivemos que clicar em cada uma das ligações individualmente.

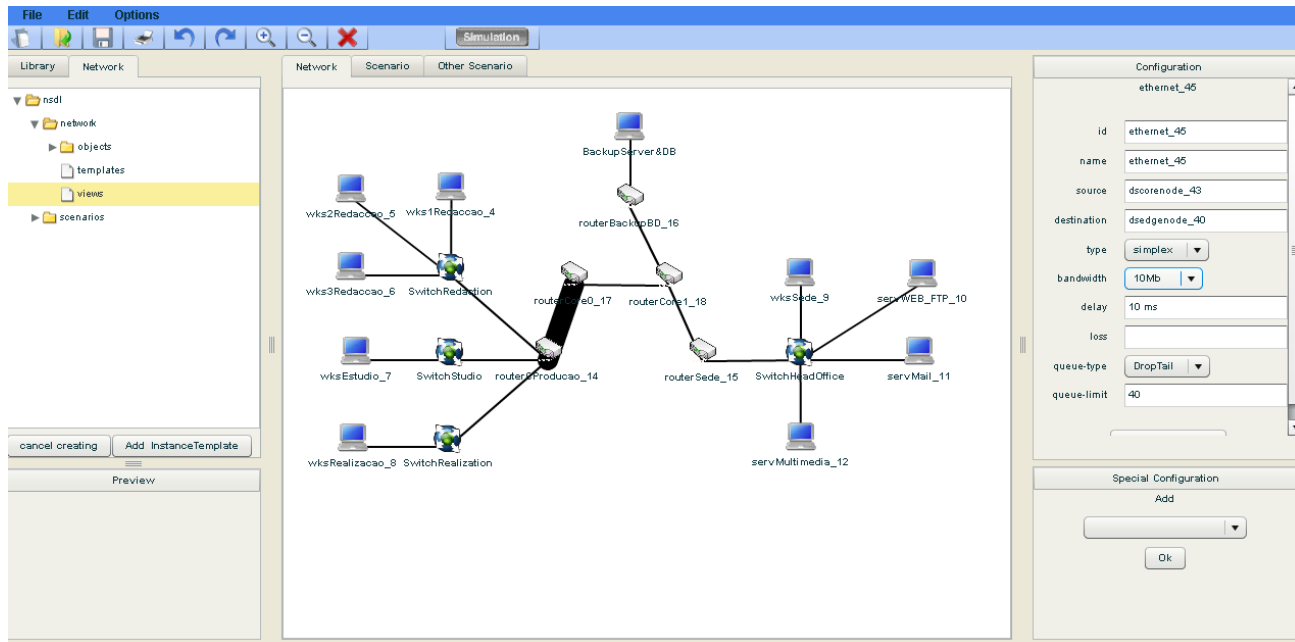


Figura 31 ligação entre Core Router e Edge Router

A figura 31 mostra um aspecto de configuração, no caso a alteração da configuração da ligação entre um dos routers de core e o router da sede.

3.º PASSO - Cada ligação entre os core routers é feita a 12Mbps.

Este passo implica verificar cada core router e da mesma forma, alterar a configuração das ligações entre eles, clicando em cima de cada uma e alterando o respectivo valor.

4.º PASSO - Nesta implementação os core routers irão providenciar QoS através da utilização de DiffServ [RFC2475 1998].

Para atingir este objectivo definimos três classes de serviços:

BE – Best Effort;

AF –Gold (Assured Forward);

AF Platinum.

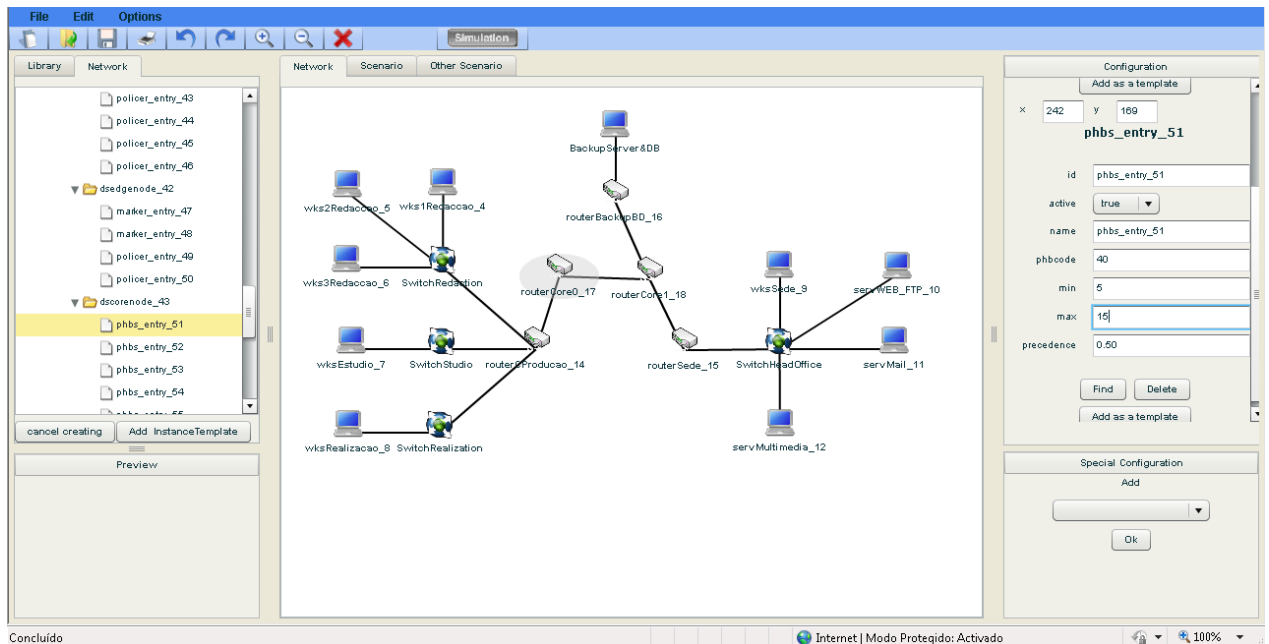


Figura 31 Configuração de Core Router para Diffserv

Na figura 31, demonstra-se como é alterada a configuração do router de core, adicionando um phbs entry.

Desta forma completamos todos os passos que pretendíamos para demonstrar os dois estudos de caso.

5.5. CONCLUSÃO

Com este capítulo demonstramos dois casos de uso bastante distintos.

No primeiro fizemos uma breve apresentação de uma solução simples.

No segundo apresentamos uma solução mais complexa com Diffserv.

6. CONCLUSÕES E PERSPECTIVAS FUTURAS

6.1. CONCLUSÕES

Nesta dissertação de mestrado apresentamos os principais aspectos sobre a proposta e o desenvolvimento de uma ferramenta gráfica para a autoria personalizada de cenários de redes, o VND. Além de desenvolvermos uma ferramenta flexível, capaz de evoluir com as tecnologias emergentes baseada no seu ficheiro de configuração XML, propomos uma ferramenta interoperável com outras ferramentas de gestão de redes através da linguagem NSDL.

Em um primeiro momento, o projecto de desenvolvimento do VND afigurou-se como algo de grande envergadura, já que o desenvolvimento de uma ferramenta de autoria de redes passível de ser integrada com outras ferramentas de gestão era algo inovador.

Numa primeira fase parecia algo impossível, sendo que com o passar do tempo e com o trabalho a aparecer, foi tornando-se cada vez mais um objectivo possível. Da mesma forma, com o tempo verificou-se que nem tudo seria desenvolvido, deixando uma grande base para que num futuro desenvolvimento, com pouco esforço, outro projecto consiga levar a ferramenta a mais altos objectivos.

Na próxima secção apresentamos um balanço da realização do VND, primeiro o que nos propusemos fazer em que incluímos o que conseguimos e o que não conseguimos, depois o que poderá ainda ser feito, seguido de algumas conclusões a este trabalho.

6.2. O QUE PROPUSEMOS FAZER

O grande objectivo da criação do VND era o desenvolvimento de uma ferramenta que permitisse a criação e configuração e um ambiente gráfico de cenários de redes com suporte à Qualidade de Serviço, e posterior geração de código para o NSDL.

Com o passar do tempo, tornamos a funcionalidades suportadas pela ferramenta mais ambiciosas, alargando os objectivos de forma a permitir a expansibilidade e adaptabilidade da ferramenta a diferentes cenários.

Como principais objectivos a serem alcançados, tinham como base o conjunto de requisitos, já enumerados anteriormente. De uma forma geral, esses requisitos foram atingidos, no entanto e numa análise crítica ao trabalho, podemos enumerar os seguintes objectivos e comentários com sendo pertinentes para avaliação do que foi realizado, apresentado na Tabela 1.

Objectivo	Concretizado	Comentários
Permitir desenhar cenários de rede, salvar, carregar, imprimir, etc.	Sim	Sendo algo básico de uma ferramenta deste tipo, foi desenvolvido numa primeira fase de adaptação à linguagem.
Utilizar todas as potencialidades do NSDL	Em parte	- Os templates e a simulação estão desenvolvidos, mas necessitam de maior refinamento, e; -A recursividade na declaração de objectos tem que ser implementada.
Permitir expansão das potencialidades da ferramenta no futuro	Sim	Todo o desenvolvimento foi feito sempre pensando que a ferramenta poderá ser expandida no futuro.
Ser amigável e simples de perceber	Em parte	Apesar de quem se submeteu a testar conseguir perceber as funcionalidades mais básicas, nas mais complexas deverá ser feito um estudo e refinamento da forma de interacção com o utilizador.
Criar ferramenta flexível e adaptável	Sim	Da ideia inicial, esta foi a maior alteração (baseada em XML), pois com este objectivo, permitimos que agora a ferramenta já possa ser utilizada em diversas áreas e que, com algum desenvolvimento, ainda possa ser melhor explorada.

Figura 28 Objectivos propostos e sua concretização

6.3. PERSPECTIVAS FUTURAS

Como próximos desenvolvimentos, podemos contemplar as seguintes propostas:

- Ajuste aos templates e a simulação de forma a estarem 100% de acordo com o NSDL;
- Implementação da recursividade na declaração de objectos;
- Implementação da ligação a bases de dados e partilha de informação com outros sistemas;
- Implementação do comportamento dinâmico do cenário de rede, dando maior atenção ao ambiente gráfico da simulação que deverá oferecer marcos temporais e progresso temporal da simulação, o que permitirá estudar mais a fundo os estados e evolução do ambiente de rede de acordo com os parâmetros fornecidos;
- Permitir a criação de relatórios que proporcionem a quem está a criar os cenários, visualizar e analisar consoante o que é o seu objectivo, e;
- Paralelamente a todas as propostas anteriores, também deverá ser estudada toda a interacção com o utilizador e pequenos defeitos, melhorando sempre que possível, ou alterando-os para que a ferramenta de autoria seja mais eficiente.

REFERÊNCIAS

- [Adobe Flex, 2010] , disponível em, <http://www.adobe.com/products/flex/>, acesso em Novembro 2009
- [Ambler, 2002] Ambler, S. "Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process", John Wiley & Sons Inc, ISBN 0471202827, 2002.
- [Boson, 2007], disponível em, <http://www.boson.com/AboutNetSim.html>, acesso em Janeiro 2010
- [Cisco, 2010], disponível em, http://www.cisco.com/web/learning/netacad/course_catalog/PacketTracer2010.html, acesso em Janeiro 2010
- [CML, 2010], disponível em, <http://www.xml-cml.org/>, Outubro 2010
- [Cockburn, 2006] Cockburn, A. "Agile Software Development", Addison-Wesley, ISBN 0321482751, 2006.
- [Conceptdraw, 2010], disponível em, <http://www.conceptdraw.com/en/>, acesso em Janeiro 2010
- [Constantine, 2003] Constantine, L., " Canonical Abstract Prototypes, for Abstract Visual and Interaction Design", disponível em <http://www.foruse.com/articles/abstract.pdf> , consultado em Janeiro 2010.
- [DocBook, 2010] , disponível em, <http://www.docbook.org/>, acesso em Outubro 2010
- [Edrawsoft, 2010], disponível em, <http://www.edrawsoft.com/>, acesso em Janeiro 2010
- [Gliffy, 2010], disponível em, <http://www.gliffy.com/>, 2010
- [Gouveia 2003] Gouveia, L., Patrício, P.F., Sousa, A.F. "Optimal Survivable Routing with a Small Number of Hops", pag. 252 a 323, Springer, ISBN: 9780387777795, 2008.
- [Intalio, 2010], disponível em, <http://www.intalio.com/bpms/designer>, acesso em Janeiro 2010

[ISO 8859] , disponível em,
http://www.iso.org/iso/catalogue_detail.htm?csnumber=16387,
acesso em Outubro 2010

[ISO 8879] , disponível em,
http://www.iso.org/iso/catalogue_detail.htm?csnumber=38580 ,
acesso em Outubro 2010

[Klugsolutions, 2008], disponível em,
http://www.klugsolutions.com/_Files/ARPO%20-20Apresentacao%20completa.pdf, acesso em Janeiro 2010

[Marques & Sampaio, 2010] Marques, E.M.D.; Sampaio, P.N.M. A Framework for the Integration of Network Modelling and Simulation Tools. To appear in Proceedings of EUROSIS - The European Multidisciplinary Society for Modelling and Simulation Technology (ESM´ 2010). Hasselt, Belgium, October 25th-27th 2010.

[Microsoft, 2010] , <http://office.microsoft.com/en-us/visio/>, acesso em Janeiro 2010

[PHP, 2010] , disponível em, <http://www.php.net/>, acesso em Novembro 2009

[Portnoi, 2007] Portnoi, M. "Um Protótipo de Simulador de Redes de Computadores para Aplicações Específicas Baseadas no Protocolo MPLS", Tese de Mestrado, Universidade de Salvador, 2007.

[RFC2386 1998], disponível em, <http://www.ietf.org/rfc/rfc2386.txt>, acesso em Outubro 2010

[RFC2475 1998] , disponível em, <http://www.ietf.org/rfc/rfc2475.txt>, acesso em Outubro 2010

[RFC2998 2000] , disponível em, <http://www.ietf.org/rfc/rfc2998.txt> , acesso em Outubro 2010

[RFC3031 2001] , disponível em, <http://www.ietf.org/rfc/rfc3031.txt>, acesso em Outubro 2010

[Rumbaugh, 99] Rumbaugh, J., Booch, G., Jacobson, I. "The Unified Modeling Language Reference Manual", Addison-Wesley, 1999. ISBN: 9780321718952

[Siemens, 2010], disponível em, <http://www.automation.siemens.com/mcms/electrical-design-software/pt/Pages/software-para-projetos-eletricos.aspx>, acesso em Janeiro 2010

[Simgrid, 2010], disponível em, <http://simgrid.gforge.inria.fr/>, acesso em Janeiro 2010

[Sommerville, 2001] SOMMERVILLE, I. "Engenharia de software.", Pearson Education, ISBN 8588639289, 2001.

[University of Southern California, 2007], disponível em, disponível em, <http://www.isi.edu/nsnam/ns/ns-build.html>, acesso em Janeiro 2010

[W3, 2010], disponível em, <http://www.w3.org/>, acesso em Outubro 2010

[Wapforum, 2010], disponível em, <http://www.wapforum.org/what/technical.htm>, acesso em Outubro 2010

ANEXOS

ANEXO I –LISTA COMPLETA DE FERRAMENTAS

- TeamWorks™ 5 www.lombardisoftware.com
- Together Architect® www.borland.com
- TrackNShare Business Process Manager
- Ultimus Process Designer
- Vignette Enterprise Content Management <http://www.vignette.com/>
- Witness™ www.lanner.com
- XIP BPM Platform www.santeon.com
- ActiveBPEL <http://www.active-endpoints.com>
- ActiveModeler www.active-flow.com
- ADONIS® www.boc-eu.com
- AllFusion Process Modeler
- Appsolut Enterprise Portal Suite 2.0
- ARIA Business Management <http://freshmeat.net/projects/arias/>
- ARIS
- ARPO www.klugsolutions.com
- aXway: Process Manager™ <http://www.axway.com>
- BPI Studio www.m1global.com
- BPMSuite www.pega.com
- Business Optimization Server - Process Sketchpad
- Charter - Visio (FirstSTEP Designer)
- CimTool - CIMOSA
- Corporate Modeler™ - Casewise www.casewise.com
- Eclipse
- EIP v7.1 http://www.bcssystem.com/IBM_EIP.asp
- Enterprise Architect 6.1 www.sparxsystems.com
- Enterprise Business Process Manager (EBPM)
- Enterprise Portal Sybase

- Entopia <http://www.entopia.com/home.htm>
- Frame-IT <http://www.x25.com.br/frame-it/>
- FreerP <http://freshmeat.net/projects/freerp/>
- Fuego 5™ www.fuego.com
- GT-X www.gtnet.com
- iGrafx Flowcharter
- iGrafx™ www.igrafx.com
- Integrated Composite Application Network (ICAN) Suite™
- Interstage Business Process Manager 7.1
- Introduces Reactor 6.0 www.oakgrovesystems.org
- Java Business Process Mgmt <http://freshmeat.net/projects/jbpm/>
- Java Graphical Process Designer
- JUDE <http://jude.change-vision.com/jude-web/index.html>
- JViews™ www.ilog.com
- Know Base <http://www.ominet.com/opensource/office/knowledge/>
- MagicDraw UML 10.0 www.nomagic.com
- Mega Suite™ www.mega.com
- Metis 3.6 Enterprise Architecture Suite™ www.troux.com
- Microsoft Visio
- n³ Designer™ www.intalio.com
- PowerDesigner® 12 www.sybase.com/developmentintegration
- Process Asset Management www.savvion.com
- ProVision™ www.proformacorp.com
- QPR <http://www.qpr.com>
- Smart Draw
- Studio www.cordys.com/products/studio
- Studio Enterprise Edition www.sun.com
- System Architect™ www.popkin.com

ANEXO II DTD DO FICHEIRO DE CONFIGURAÇÃO

<!ELEMENT nsdlObjects (instances, objects, links) >

<!ELEMENT instances (instance+) >

<!ELEMENT objects (category+) >

<!ELEMENT links (link+) >

<!ELEMENT category (name, object+) >

<!ELEMENT instance (name, par+) >

<!ELEMENT link (name, image, color, source, destination, par+) >

<!ELEMENT object (name, image, par+) >

<!ELEMENT par (mandatory | name | option | par | size | type | value) * >

<!ELEMENT color (#PCDATA) >

<!ELEMENT destination (#PCDATA) >

<!ELEMENT image (#PCDATA) >

<!ELEMENT mandatory (#PCDATA) >

<!ELEMENT name (#PCDATA) >

<!ELEMENT option (#PCDATA) >

<!ELEMENT size (#PCDATA) >

<!ELEMENT source (#PCDATA) >

<!ELEMENT type (#PCDATA) >

<!ELEMENT value (#PCDATA) >

ANEXO IV - XML DE EXEMPLO PARA O FICHEIRO DE CONFIGURAÇÃO (1.º CASO DE USO)

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE nsdlObjects SYSTEM "inputxml.dtd" >
<nsdlObjects>
  <instances>
    <instance>
      <name>application</name>
      <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <mandatory>true</mandatory>
      </par>
      <par>
        <name>active</name>
        <type>boolean</type>
        <size>16</size>
        <mandatory>false</mandatory>
      </par>
      <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
      </par>
      <par>
        <par>
          <name>notes</name>
          <type>string</type>
          <size>256</size>
          <mandatory>false</mandatory>
        </par>
        <name>role</name>
        <type>options</type>
        <size>256</size>
        <option>server</option>
        <option>client</option>
        <mandatory>false</mandatory>
      </par>
      <par>
        <name>rate</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
      </par>
      <par>
        <name>packetsize</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
      </par>
    </instance>
    <instance>
      <name>protocol</name>
      <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <mandatory>true</mandatory>
      </par>
      <par>
        <name>active</name>
        <type>boolean</type>
        <size>16</size>
      </par>
    </instance>
  </instances>
</nsdlObjects>
```

```

        <mandatory>false</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>notes</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>TcplpLayer</name>
        <type>options</type>
        <size>256</size>
        <option>1</option>
        <option>2</option>
        <option>3</option>
        <option>4</option>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>OsiLayer</name>
        <type>options</type>
        <size>256</size>
        <option>Application</option>
        <option>Presentation</option>
        <option>Session</option>
        <option>Transport</option>
        <option>Network</option>
        <option>Data-Link</option>
        <option>Physical</option>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>version</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>packetsize</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
</instance>
<instance>
    <name>interface</name>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <mandatory>true</mandatory>
    </par>
    <par>
        <name>active</name>
        <type>boolean</type>
        <size>16</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>

```

```

<par>
  <name>notes</name>
  <type>string</type>
  <size>256</size>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>bandwith</name>
  <type>options</type>
  <size>256</size>
  <option>tcp</option>
  <option>osi</option>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>delay</name>
  <type>options</type>
  <size>256</size>
  <option>tcp</option>
  <option>osi</option>
  <mandatory>>false</mandatory>
</par>
</instance>
<instance>
  <name>phbs</name>
  <par>
    <name>id</name>
    <type>string</type>
    <size>16</size>
    <mandatory>>true</mandatory>
  </par>
  <par>
    <name>active</name>
    <type>boolean</type>
    <size>16</size>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>name</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>phb code</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>min</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>max</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>precedence</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
</instance>
<instance>
  <name>scheduler</name>

```

```

    <par>
      <name>id</name>
      <type>string</type>
      <size>16</size>
      <mandatory>true</mandatory>
    </par>
  </par>
  <par>
    <name>active</name>
    <type>boolean</type>
    <size>16</size>
    <mandatory>false</mandatory>
  </par>
</par>
<par>
  <name>name</name>
  <type>string</type>
  <size>256</size>
  <mandatory>false</mandatory>
</par>
<par>
  <name>queue</name>
  <type>string</type>
  <size>256</size>
  <mandatory>false</mandatory>
</par>
<par>
  <name>weight</name>
  <type>string</type>
  <size>256</size>
  <mandatory>false</mandatory>
</par>
</instance>
</instances>
<objects>
  <category>
    <name>nodes</name>
    <object>
      <name>node</name>
      <image>node.png</image>
      <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <value></value>
        <mandatory>true</mandatory>
      </par>
      <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>false</mandatory>
      </par>
      <par>
        <name>notes</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>false</mandatory>
      </par>
      <par>
        <name>interface</name>
        <type>instance</type>
        <size>many</size>
        <value></value>
        <mandatory>false</mandatory>
      </par>
      <par>
        <name>protocol</name>
        <type>instance</type>
        <size>many</size>

```

```

        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>application</name>
        <type>instance</type>
        <size>many</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
</object>
<object>
    <name>computer</name>
    <image>computer.png</image>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <value></value>
        <mandatory>true</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>active</name>
        <type>boolean</type>
        <size>16</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>notes</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>interface</name>
        <type>instance</type>
        <size>many</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>protocol</name>
        <type>instance</type>
        <size>many</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>application</name>
        <type>instance</type>
        <size>many</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>role</name>
        <type>options</type>
        <size>256</size>
        <option>server</option>
        <option>peer</option>
        <mandatory>false</mandatory>
    </par>

```

```

    </par>
  </object>
</object>
<object>
  <name>Core_router</name>
  <image>router.png</image>
  <par>
    <name>id</name>
    <type>string</type>
    <size>16</size>
    <value></value>
    <mandatory>true</mandatory>
  </par>
  <par>
    <name>name</name>
    <type>string</type>
    <size>256</size>
    <value></value>
    <mandatory>false</mandatory>
  </par>
  <par>
    <name>notes</name>
    <type>string</type>
    <size>256</size>
    <value></value>
    <mandatory>false</mandatory>
  </par>
  <par>
    <name>interface</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>false</mandatory>
  </par>
  <par>
    <name>protocol</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>false</mandatory>
  </par>
  <par>
    <name>phbs</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>false</mandatory>
  </par>
  <par>
    <name>scheduler</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>false</mandatory>
  </par>
</object>
</category>
<category>
  <name>domains</name>
  <object>
    <name>domain</name>
    <image>domain.png</image>
    <par>
      <name>id</name>
      <type>string</type>
      <size>16</size>
      <value></value>
      <mandatory>true</mandatory>
    </par>
    <par>
      <name>name</name>

```

```

        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>notes</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
</object>
<object>
    <name>diffserv</name>
    <image>diffserv.png</image>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <value></value>
        <mandatory>>true</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>active</name>
        <type>boolean</type>
        <size>16</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>notes</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>bandwidth</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>delay</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>loss</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>jitter</name>
        <type>string</type>
        <size>256</size>
        <value></value>
    </par>

```

```

        <mandatory>false</mandatory>
    </par>
    <par>
        <name>PerHopBehaviour</name>
        <type>options</type>
        <size>256</size>
        <option>Default PHB</option>
        <option>Expedited Forwarding</option>
        <option>Assured Forwarding</option>
        <option>Class Selector</option>
        <mandatory>false</mandatory>
    </par>
</object>
</category>
</objects>
<links>
    <link>
        <name>ethernet</name>
        <image>ethernet.png</image>
        <color>000000</color>
        <source></source>
        <destination></destination>
        <par>
            <name>id</name>
            <type>string</type>
            <size>16</size>
            <mandatory>true</mandatory>
        </par>
        <par>
            <name>name</name>
            <type>string</type>
            <size>16</size>
            <mandatory>true</mandatory>
        </par>
        <par>
            <name>source</name>
            <type>string</type>
            <size>16</size>
            <mandatory>true</mandatory>
        </par>
        <par>
            <name>destination</name>
            <type>string</type>
            <size>16</size>
            <mandatory>true</mandatory>
        </par>
        <par>
            <name>type</name>
            <type>options</type>
            <size>256</size>
            <option>simplex</option>
            <option>duplex</option>
            <mandatory>false</mandatory>
        </par>
        <par>
            <name>bandwidth</name>
            <type>options</type>
            <size>256</size>
            <option>1Mb</option>
            <option>10Mb</option>
            <option>100Mb</option>
            <option>1Gb</option>
            <mandatory>false</mandatory>
        </par>
        <par>
            <name>delay</name>
            <type>string</type>
            <size>16</size>
            <mandatory>true</mandatory>
        </par>
    </link>

```

```

</link>
<link>
  <name>fiber</name>
  <image>fiber.png</image>
  <color>000222</color>
  <source></source>
  <destination></destination>
  <par>
    <name>id</name>
    <type>string</type>
    <size>16</size>
    <mandatory>true</mandatory>
  </par>
  <par>
    <name>name</name>
    <type>string</type>
    <size>16</size>
    <mandatory>true</mandatory>
  </par>
  <par>
    <name>source</name>
    <type>string</type>
    <size>16</size>
    <mandatory>true</mandatory>
  </par>
  <par>
    <name>destination</name>
    <type>string</type>
    <size>16</size>
    <mandatory>true</mandatory>
  </par>
  <par>
    <name>type</name>
    <type>options</type>
    <size>256</size>
    <option>simplex</option>
    <option>duplex</option>
    <mandatory>false</mandatory>
  </par>
  <par>
    <name>bandwidth</name>
    <type>options</type>
    <size>256</size>
    <option>100Mb</option>
    <option>1Gb</option>
    <option>10Gb</option>
    <option>100Gb</option>
    <option>300Gb</option>
    <mandatory>false</mandatory>
  </par>
  <par>
    <name>delay</name>
    <type>string</type>
    <size>16</size>
    <mandatory>true</mandatory>
  </par>
</link>
</links>
</nsdlObjects>

```

ANEXO V – XML DE EXEMPLO PARA O FICHEIRO DE CONFIGURAÇÃO (2.º CASO DE USO)

<!-- ***** changelog from v2 *****- added instances: application, phbs_entry, queue, marker_entry
policer_entry, TSW3CM, TokenBucket, srtcm, trtcm- abstraction of scheduler instance- specification change for objects
nsedgenode nscorenode ***** changelog from v4.2 *****- added instance rip for nsedgenode and nscorenode-->

```
<nsdlObjects>
  <instances>
    <instance>
      <name>application</name>
      <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <mandatory>true</mandatory>
      </par>
      <par>
        <name>active</name>
        <type>boolean</type>
        <size>16</size>
        <mandatory>false</mandatory>
      </par>
      <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
      </par>
      <!-- Appllication attributes -->
      <par>
        <par>
          <name>notes</name>
          <type>string</type>
          <size>256</size>
          <mandatory>false</mandatory>
        </par>
        <name>role</name>
        <type>options</type>
        <size>256</size>
        <option>server</option>
        <option>client</option>
        <mandatory>false</mandatory>
      </par>
      <par>
        <name>rate</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
      </par>
      <par>
        <name>packetsize</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
      </par>
      <par>
        <name>src.app</name>
```

```

        <type>string</type>
        <size>256</size>
        <mandatory>true</mandatory>
    </par>
</par>
<par>
    <name>dst.app</name>
    <type>string</type>
    <size>256</size>
    <mandatory>true</mandatory>
</par>
<par>
    <name>protocol.id</name>
    <type>string</type>
    <size>256</size>
    <mandatory>true</mandatory>
</par>
<!-- Application attributes end of declaration -->
</instance>
--> <!-- Application instance end of declaration -->
<!-- CBR instance -->
<instance>
    <name>cbr</name>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <mandatory>true</mandatory>
    </par>
    <par>
        <name>active</name>
        <type>boolean</type>
        <size>16</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
    <!-- CBR attributes -->
    <par>
        <par>
            <name>notes</name>
            <type>string</type>
            <size>256</size>
            <mandatory>false</mandatory>
        </par>
        <name>role</name>
        <type>options</type>
        <size>256</size>
        <option>server</option>
        <option>client</option>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>rate</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>

```

```

<par>
  <name>packetize</name>
  <type>string</type>
  <size>256</size>
  <mandatory>false</mandatory>
</par>
<par>
  <name>src.app</name>
  <type>string</type>
  <size>256</size>
  <mandatory>true</mandatory>
</par>
<par>
  <name>dst.app</name>
  <type>string</type>
  <size>256</size>
  <mandatory>true</mandatory>
</par>
<par>
  <name>protocol.id</name>
  <type>string</type>
  <size>256</size>
  <mandatory>true</mandatory>
</par>
<par>
  <name>random</name>
  <type>string</type>
  <size>256</size>
  <mandatory>false</mandatory>
</par>
<par>
  <name>maxpackets</name>
  <type>string</type>
  <size>256</size>
  <mandatory>false</mandatory>
</par>
--><!-- CBR attributes end of declaration -->
</instance>
<!-- CBR instance end of declaration -->
<!-- Exponential instance -->
<instance>
  <name>exponential</name>
  <par>
    <name>id</name>
    <type>string</type>
    <size>16</size>
    <mandatory>true</mandatory>
  </par>
  <par>
    <name>active</name>
    <type>boolean</type>
    <size>16</size>
    <mandatory>false</mandatory>
  </par>
  <par>
    <name>name</name>
    <type>string</type>
    <size>256</size>
    <mandatory>false</mandatory>
  </par>
<!-- Exponential attributes -->

```

```

<par>
  <par>
    <name>notes</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
  <name>role</name>
  <type>options</type>
  <size>256</size>
  <option>server</option>
  <option>client</option>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>rate</name>
  <type>string</type>
  <size>256</size>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>packetsize</name>
  <type>string</type>
  <size>256</size>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>src.app</name>
  <type>string</type>
  <size>256</size>
  <mandatory>>true</mandatory>
</par>
<par>
  <name>dst.app</name>
  <type>string</type>
  <size>256</size>
  <mandatory>>true</mandatory>
</par>
<par>
  <name>protocol.id</name>
  <type>string</type>
  <size>256</size>
  <mandatory>>true</mandatory>
</par>
<par>
  <name>bursttime</name>
  <type>string</type>
  <size>256</size>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>idletime</name>
  <type>string</type>
  <size>256</size>
  <mandatory>>false</mandatory>
</par>
<!-- Exponential attributes end of declaration -->
</instance>
<!-- Exponential instance end of declaration -->
<!-- connection instance (optional) -->
<!-- <instance> <name>connection</name>connections attributes <par> <name>source</name> <type>string</type>

```

```

<size>256</size> <mandatory>>false</mandatory> </par><par> <name>destination</name> <type>string</type>
<size>256</size> <mandatory>>false</mandatory> </par>Connection attributes end of declaration </instance> -->
<!-- Connection instance end of declaration -->
<!-- FTP instance -->
<instance>
  <name>ftp</name>
  <par>
    <name>id</name>
    <type>string</type>
    <size>16</size>
    <mandatory>>true</mandatory>
  </par>
  <par>
    <name>active</name>
    <type>boolean</type>
    <size>16</size>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>name</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
  <!-- FTP attributes -->
  <par>
    <par>
      <name>notes</name>
      <type>string</type>
      <size>256</size>
      <mandatory>>false</mandatory>
    </par>
    <name>role</name>
    <type>options</type>
    <size>256</size>
    <option>server</option>
    <option>client</option>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>rate</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>packetsize</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>src.app</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>true</mandatory>
  </par>
  <par>
    <name>dst.app</name>
    <type>string</type>
    <size>256</size>
  </par>

```

```

    <mandatory>true</mandatory>
</par>
<par>
    <name>protocol.id</name>
    <type>string</type>
    <size>256</size>
    <mandatory>true</mandatory>
</par>
<par>
    <name>maxpackets</name>
    <type>string</type>
    <size>256</size>
    <mandatory>false</mandatory>
</par>
<!-- FTP attributes end of declaration -->
</instance>
<!-- FTP instance end of declaration -->
<!-- Telnet instance -->
<instance>
    <name>telnet</name>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <mandatory>true</mandatory>
    </par>
    <par>
        <name>active</name>
        <type>boolean</type>
        <size>16</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
    <!-- Telnet attributes -->
    <par>
        <par>
            <name>notes</name>
            <type>string</type>
            <size>256</size>
            <mandatory>false</mandatory>
        </par>
        <name>role</name>
        <type>options</type>
        <size>256</size>
        <option>server</option>
        <option>client</option>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>rate</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>packetsize</name>

```

```

        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
</par>
<par>
    <name>src.app</name>
    <type>string</type>
    <size>256</size>
    <mandatory>true</mandatory>
</par>
<par>
    <name>dst.app</name>
    <type>string</type>
    <size>256</size>
    <mandatory>true</mandatory>
</par>
<par>
    <name>protocol.id</name>
    <type>string</type>
    <size>256</size>
    <mandatory>true</mandatory>
</par>
<par>
    <name>interval</name>
    <type>string</type>
    <mandatory>false</mandatory>
</par>
<!-- Telnet attributes end of declaration -->
</instance>
<!-- Telnet instance end of declaration -->
<!-- Protocol instance -->
<instance>
    <name>protocol</name>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <mandatory>true</mandatory>
    </par>
    <par>
        <name>active</name>
        <type>boolean</type>
        <size>16</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
    <!-- Protocol attributes -->
    <par>
        <name>notes</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>TcpIpLayer</name>
        <type>options</type>

```

```

    <size>256</size>
    <option>1</option>
    <option>2</option>
    <option>3</option>
    <option>4</option>
    <mandatory>>false</mandatory>
</par>
<par>
  <name>OsiLayer</name>
  <type>options</type>
  <size>256</size>
  <option>Application</option>
  <option>Presentation</option>
  <option>Session</option>
  <option>Transport</option>
  <option>Network</option>
  <option>Data-Link</option>
  <option>Physical</option>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>version</name>
  <type>string</type>
  <size>256</size>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>packetsize</name>
  <type>string</type>
  <size>256</size>
  <mandatory>>false</mandatory>
</par>
<!-- Protocol attributes end of declaration -->
</instance>
<!-- Protocol instance end of declaration -->
<!-- Interface instance -->
<instance>
  <name>interface</name>
  <par>
    <name>id</name>
    <type>string</type>
    <size>16</size>
    <mandatory>>true</mandatory>
  </par>
  <par>
    <name>active</name>
    <type>boolean</type>
    <size>16</size>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>name</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
  <!-- Interface attributes -->
  <par>
    <name>notes</name>
    <type>string</type>
    <size>256</size>

```

```

    <mandatory>false</mandatory>
</par>
<par>
    <name>bandwidth</name>
    <type>string</type>
    <size>256</size>
    <mandatory>false</mandatory>
</par>
<par>
    <name>delay</name>
    <type>string</type>
    <size>256</size>
    <mandatory>false</mandatory>
</par>
<!-- Interface attributes end of declaration -->
</instance>
<!-- Interface instance end of declaration -->
<!-- PHB instance -->
<instance>
    <name>phbs_entry</name>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <mandatory>true</mandatory>
    </par>
    <par>
        <name>active</name>
        <type>boolean</type>
        <size>16</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
    <!-- PHB Attributes -->
    <par>
        <name>phbcode</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>min</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>max</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>precedence</name>
        <type>string</type>
        <size>256</size>

```

```

        <mandatory>false</mandatory>
    </par>
</instance>
<!-- PHB attributes end of declaration -->
</instance>
<!-- PHB instance end of declaration -->
<!-- ***** NSDL Validation Structure for phbs ***** -->
<nsdl_name name= "phbs">
    <par></par>
    <par></par>
    <par></par>
    <nsdl_name name= "entry">
        <par>phbcode</par>
        <par> min</par>
        <par>max</par>
        <par>precedence</par>
    </nsdl_name>
</nsdl_name>
<!-- ***** NSDL Validation Structure for phbs end of declaration ***** -->
<!-- Scheduler instance -->
<instance>
    <name>scheduler</name>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <mandatory>true</mandatory>
    </par>
    <par>
        <name>active</name>
        <type>boolean</type>
        <size>16</size>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <mandatory>false</mandatory>
    </par>
<!-- Scheduler attributes -->
    <par>
        <name>scheduler type</name>
        <type>options</type>
        <option>RR</option>
        <option>PRI</option>
        <option>WRR</option>
        <option>WIRR</option>
        <mandatory>false</mandatory>
    </par>
<!-- Scheduler attributes end of declaration -->
</instance>
<!-- Scheduler instance end of declaration -->
<instance>
    <name>queue</name>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <mandatory>true</mandatory>
    </par>
    <par>

```

```

        <name>active</name>
        <type>boolean</type>
        <size>16</size>
        <mandatory>>false</mandatory>
    </par>
</par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <mandatory>>false</mandatory>
    </par>
<!-- queue attributes -->
    <par>
        <name>weight</name>
        <type>string</type>
        <size>256</size>
        <mandatory>>false</mandatory>
    </par>
</instance>
<!-- ***** NSDL Validation Structure for scheduler ***** -->
<nsdl_name name= "scheduler">
    <par></par>
    <par></par>
    <par></par>
    <par></par>
    <nsdl_name name= "queue">
        <par>id</par>
        <par></par>
        <par></par>
        <par>weight</par>
    </nsdl_name>
</nsdl_name>
<!-- ***** NSDL Validation Structure for scheduler end of declaration ***** -->
<instance>
    <name>marker_entry</name>
    <par>
        <name>scr.node</name>
        <type>string</type>
        <size>256</size>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>dst.node</name>
        <type>string</type>
        <size>256</size>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>phb.code</name>
        <type>string</type>
        <size>256</size>
        <mandatory>>false</mandatory>
    </par>
</instance>
<!-- ***** NSDL Validation Structure for marker ***** -->
<nsdl_name name= "marker">
    <nsdl_name name= "entry">
        <par>src.node</par>
        <par>dst.node</par>
        <par>phb.code</par>
    </nsdl_name>
</nsdl_name>

```

```

</nsdl_name>
<!-- ***** NSDL Validation Structure for marker end of declaration ***** -->
<instance>
  <name>policer_entry</name>
  <par>
    <name>phb.code</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>new.phb.code</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>cir</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>policy type</name>
    <type>options</type>
    <option>TSW2CM</option>
    <option>TSW3CM</option>
    <option>TokenBucket</option>
    <option>srtc</option>
    <option>trtc</option>
    <mandatory>>true</mandatory>
  </par>
</instance>
<instance>
  <name>TSW3CM</name>
  <par>
    <name>pir</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
  <par>
    <name>sec.phb.code</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
</instance>
<instance>
  <name>TokenBucket</name>
  <par>
    <name>cbs</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
  </par>
</instance>
<instance>
  <name>srtc</name>
  <par>
    <name>cbs</name>

```

```

        <type>string</type>
        <size>256</size>
        <mandatory>>false</mandatory>
    </par>
</par>
<par>
    <name>ebs</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
</par>
<par>
    <name>sec.phb.code</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
</par>
</instance>
<instance>
    <name>trtcm</name>
    <par>
        <name>cbs</name>
        <type>string</type>
        <size>256</size>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>pir</name>
        <type>string</type>
        <size>256</size>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>pbs</name>
        <type>string</type>
        <size>256</size>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>sec.phb.code</name>
        <type>string</type>
        <size>256</size>
        <mandatory>>false</mandatory>
    </par>
</instance>
<!-- ***** NSDL Validation Structure for policer ***** -->
<nsdl_name name= "policer">
    <nsdl_name name= "entry">
        <par>phb.node</par>
        <par>new.phb.node</par>
        <par>cir</par>
        <par></par>
    </nsdl_name>
</nsdl_name>
<!-- ***** NSDL Validation Structure for policer end of declaration ***** -->
<!-- RIP instance -->
<instance>
    <name>rip</name>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>

```

```

    <mandatory>true</mandatory>
</par>
<par>
  <name>active</name>
  <type>boolean</type>
  <size>16</size>
  <mandatory>false</mandatory>
</par>
<par>
  <name>name</name>
  <type>string</type>
  <size>256</size>
  <mandatory>false</mandatory>
</par>
<!-- RIP attributes -->
<par>
  <name>version</name>
  <type>string</type>
  <size>256</size>
  <mandatory>false</mandatory>
</par>
<par>
  <name>admdst</name>
  <type>string</type>
  <size>256</size>
  <mandatory>false</mandatory>
</par>
<par>
  <name>up dint</name>
  <type>string</type>
  <size>256</size>
  <mandatory>false</mandatory>
</par>
<!-- RIP attributes end of declaration -->
</instance>
<!-- RIP instance end of declaration -->
<!-- Static Route instance -->
<instance>
  <name>staticroute</name>
  <par>
    <name>id</name>
    <type>string</type>
    <size>16</size>
    <mandatory>true</mandatory>
  </par>
  <par>
    <name>active</name>
    <type>boolean</type>
    <size>16</size>
    <mandatory>false</mandatory>
  </par>
  <par>
    <name>name</name>
    <type>string</type>
    <size>256</size>
    <mandatory>false</mandatory>
  </par>
  <!-- Static Route attributes -->
  <par>
    <name>src.address</name>
    <type>string</type>

```

```

        <size>256</size>
        <mandatory>>false</mandatory>
    </par>
</par>
<par>
    <name>dst.address</name>
    <type>string</type>
    <size>256</size>
    <mandatory>>false</mandatory>
</par>
<!-- Static Route attributes end of declaration -->
</instance>
<!-- Static Route instance end of declaration -->
</instances>
<!-- NSDObjects instances end of declaration -->
<!-- Objects -->
<objects>
    <category>
        <name>nodes</name>
        <!-- Node object -->
        <object>
            <name>node</name>
            <image>node.png</image>
            <par>
                <name>id</name>
                <type>string</type>
                <size>16</size>
                <value></value>
                <mandatory>>true</mandatory>
            </par>
            <par>
                <name>name</name>
                <type>string</type>
                <size>256</size>
                <value></value>
                <mandatory>>false</mandatory>
            </par>
            <par>
                <name>notes</name>
                <type>string</type>
                <size>256</size>
                <value></value>
                <mandatory>>false</mandatory>
            </par>
            <par>
                <name>interface</name>
                <type>instance</type>
                <size>many</size>
                <value></value>
                <mandatory>>false</mandatory>
            </par>
            <par>
                <name>protocol</name>
                <type>instance</type>
                <size>many</size>
                <value></value>
                <mandatory>>false</mandatory>
            </par>
            <par>
                <name>application</name>
                <type>instance</type>
                <size>many</size>
            </par>
        </object>
    </category>
</objects>

```

```

        <value></value>
        <mandatory>false</mandatory>
    </par>
</object>
<!-- Node object end of declaration -->
<!-- Computer object -->
<object>
    <name>computer</name>
    <image>computer.png</image>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <value></value>
        <mandatory>true</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>active</name>
        <type>boolean</type>
        <size>16</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>notes</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>interface</name>
        <type>instance</type>
        <size>many</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>protocol</name>
        <type>instance</type>
        <size>many</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>cbr</name>
        <type>instance</type>
        <size>many</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>exponential</name>
        <type>instance</type>

```

```

    <size>many</size>
    <value></value>
    <mandatory>>false</mandatory>
</par>
<par>
    <name>ftp</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>>false</mandatory>
</par>
<par>
    <name>telnet</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>>false</mandatory>
</par>
<par>
    <name>application</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>>false</mandatory>
</par>
<par>
    <name>role</name>
    <type>options</type>
    <size>256</size>
    <option>server</option>
    <option>peer</option>
    <mandatory>>false</mandatory>
</par>
</object>
<!-- Computer object end of declaration -->
<!-- Switch object -->
<object>
    <name>switch</name>
    <image>node.png</image>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <value></value>
        <mandatory>>true</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>active</name>
        <type>boolean</type>
        <size>16</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
</object>

```

```

    <name>notes</name>
    <type>string</type>
    <size>256</size>
    <value></value>
    <mandatory>>false</mandatory>
</par>
<par>
    <name>interfaces</name>
    <type>string</type>
    <value></value>
    <mandatory>>false</mandatory>
</par>
</object>
<!-- Switch object end of declaration -->
<!-- Generic router object -->
<object>
    <name>router</name>
    <image>router.png</image>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <value></value>
        <mandatory>>true</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>notes</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
    <!-- Generic router attributes -->
    <par>
        <name>interface</name>
        <type>instance</type>
        <size>many</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
    <!-- <par> (Router doesn't use these kind of
protocols)<name>protocol</name><type>instance</type><size>many</size><value></value><mandatory>>false</mandatory>
</par> -->
    <par>
        <name>phbs</name>
        <type>instance</type>
        <size>many</size>
        <value></value>
        <mandatory>>false</mandatory>
    </par>
    <par>
        <name>scheduler</name>
        <type>instance</type>
        <size>many</size>

```

```

        <value></value>
        <mandatory>false</mandatory>
    </par>
</par>
<par>
    <name>rip</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>false</mandatory>
</par>
<par>
    <name>staticroute</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>false</mandatory>
</par>
<!-- Generic router attributes end of declaration -->
</object>
<!-- Generic router object end of declaration -->
<!-- Edge_router object -->
<object>
    <name>dsedgenode</name>
    <image>router.png</image>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <value></value>
        <mandatory>true</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>notes</name>
        <type>string</type>
        <size>256</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>interface</name>
        <type>instance</type>
        <size>many</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>protocol</name>
        <type>instance</type>
        <size>many</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
    <par>
        <name>marker_entry</name>

```

```

        <type>instance</type>
        <size>many</size>
        <value></value>
        <mandatory>false</mandatory>
    </par>
</par>
<par>
    <name>policer_entry</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>false</mandatory>
</par>
<par>
    <name>TSW3CM</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>false</mandatory>
</par>
<par>
    <name>TokenBucket</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>false</mandatory>
</par>
<par>
    <name>srtcm</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>false</mandatory>
</par>
<par>
    <name>trtcm</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>false</mandatory>
</par>
<par>
    <name>rip</name>
    <type>instance</type>
    <size>many</size>
    <value></value>
    <mandatory>false</mandatory>
</par>
</object>
<!-- Edge_router object end of declaration -->
<object>
    <name>dscorenode</name>
    <image>router.png</image>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <value></value>
        <mandatory>true</mandatory>
    </par>
    <par>
        <name>name</name>

```

```
<type>string</type>
<size>256</size>
<value></value>
<mandatory>>false</mandatory>
</par>
<par>
  <name>notes</name>
  <type>string</type>
  <size>256</size>
  <value></value>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>interface</name>
  <type>instance</type>
  <size>many</size>
  <value></value>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>protocol</name>
  <type>instance</type>
  <size>many</size>
  <value></value>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>staticroute</name>
  <type>instance</type>
  <size>many</size>
  <value></value>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>phbs_entry</name>
  <type>instance</type>
  <size>many</size>
  <value></value>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>scheduler</name>
  <type>instance</type>
  <size>many</size>
  <value></value>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>queue</name>
  <type>instance</type>
  <size>many</size>
  <value></value>
  <mandatory>>false</mandatory>
</par>
<par>
  <name>rip</name>
  <type>instance</type>
  <size>many</size>
  <value></value>
  <mandatory>>false</mandatory>
</par>
```

```

</object>
</category>
<category>
  <name>domains</name>
  <!-- Domain object -->
  <object>
    <name>domain</name>
    <image>domain.png</image>
    <par>
      <name>id</name>
      <type>string</type>
      <size>16</size>
      <value></value>
      <mandatory>true</mandatory>
    </par>
    <par>
      <name>name</name>
      <type>string</type>
      <size>256</size>
      <value></value>
      <mandatory>false</mandatory>
    </par>
    <par>
      <name>notes</name>
      <type>string</type>
      <size>256</size>
      <value></value>
      <mandatory>false</mandatory>
    </par>
  </object>
  <!-- Domain object end of declaration -->
  <!-- Web object -->
  <object>
    <name>www</name>
    <image>www.png</image>
    <par>
      <name>id</name>
      <type>string</type>
      <size>16</size>
      <value></value>
      <mandatory>true</mandatory>
    </par>
    <par>
      <name>name</name>
      <type>string</type>
      <size>256</size>
      <value></value>
      <mandatory>false</mandatory>
    </par>
    <par>
      <name>active</name>
      <type>boolean</type>
      <size>16</size>
      <value></value>
      <mandatory>false</mandatory>
    </par>
    <par>
      <name>notes</name>
      <type>string</type>
      <size>256</size>
      <value></value>
    </par>
  </object>

```

```

        <mandatory>false</mandatory>
    </par>
</par>
<par>
    <name>bandwidth</name>
    <type>string</type>
    <size>256</size>
    <value></value>
    <mandatory>false</mandatory>
</par>
<par>
    <name>delay</name>
    <type>string</type>
    <size>256</size>
    <value></value>
    <mandatory>false</mandatory>
</par>
<par>
    <name>loss</name>
    <type>string</type>
    <size>256</size>
    <value></value>
    <mandatory>false</mandatory>
</par>
<par>
    <name>jitter</name>
    <type>string</type>
    <size>256</size>
    <value></value>
    <mandatory>false</mandatory>
</par>
</object>
<!-- Web object end of declaration -->
</category>
</objects>
<links>
<!-- Link object (ethernet)-->
<link>
    <name>ethernet</name>
    <image>ethernet.png</image>
    <color>000000</color>
    <source/>
    <destination/>
    <par>
        <name>id</name>
        <type>string</type>
        <size>16</size>
        <mandatory>true</mandatory>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
        <size>16</size>
        <mandatory>true</mandatory>
    </par>
    <par>
        <name>source</name>
        <type>string</type>
        <size>16</size>
        <mandatory>true</mandatory>
    </par>
</link>
</links>

```

```

    <name>destination</name>
    <type>string</type>
    <size>16</size>
    <mandatory>true</mandatory>
</par>
<par>
    <name>type</name>
    <type>options</type>
    <size>256</size>
    <option>simplex</option>
    <option>duplex</option>
    <mandatory>false</mandatory>
</par>
<par>
    <name>bandwidth</name>
    <type>options</type>
    <size>256</size>
    <option>1Mb</option>
    <option>2Mb</option>
    <option>4Mb</option>
    <option>6Mb</option>
    <option>8Mb</option>
    <option>10Mb</option>
    <option>100Mb</option>
    <option>1Gb</option>
    <mandatory>false</mandatory>
</par>
<par>
    <name>delay</name>
    <type>string</type>
    <size>16</size>
    <mandatory>true</mandatory>
</par>
<par>
    <name>loss</name>
    <type>string</type>
    <size>16</size>
    <mandatory>true</mandatory>
</par>
<par>
    <name>queue-type</name>
    <type>options</type>
    <size>256</size>
    <option>DropTail</option>
    <option>RED</option>
    <mandatory>false</mandatory>
</par>
<par>
    <name>queue-limit</name>
    <type>string</type>
    <size>256</size>
    <mandatory>false</mandatory>
</par>
</link>
<!-- Link object (ethernet) end of declaration -->
<!-- Link object (fiber) -->
<link>
    <name>fiber</name>
    <image>fiber.png</image>
    <color>000222</color>
    <source/>

```

```
<destination/>
<par>
  <name>id</name>
  <type>string</type>
  <size>16</size>
  <mandatory>true</mandatory>
</par>
<par>
  <name>name</name>
  <type>string</type>
  <size>16</size>
  <mandatory>true</mandatory>
</par>
<par>
  <name>source</name>
  <type>string</type>
  <size>16</size>
  <mandatory>true</mandatory>
</par>
<par>
  <name>destination</name>
  <type>string</type>
  <size>16</size>
  <mandatory>true</mandatory>
</par>
<par>
  <name>type</name>
  <type>options</type>
  <size>256</size>
  <option>simplex</option>
  <option>duplex</option>
  <mandatory>false</mandatory>
</par>
<par>
  <name>bandwidth</name>
  <type>options</type>
  <size>256</size>
  <option>100Mb</option>
  <option>1Gb</option>
  <option>10Gb</option>
  <option>100Gb</option>
  <option>300Gb</option>
  <mandatory>false</mandatory>
</par>
<par>
  <name>delay</name>
  <type>string</type>
  <size>16</size>
  <mandatory>true</mandatory>
</par>
<par>
  <name>queue-type</name>
  <type>options</type>
  <size>256</size>
  <option>DropTail</option>
  <option>RED</option>
  <mandatory>false</mandatory>
</par>
<par>
  <name>queue-limit</name>
  <type>string</type>
```

```
<size>256</size>
<mandatory>false</mandatory>
</par>
</link>
<!-- Link object (fiber) end of declaration -->
</links>
</nsdlObjects>
```

ANEXO VI – DTD PARA FICHEIRO DE EXPORTAÇÃO

```
<!ELEMENT root ( objects+, templates, numberofnodes, numberOfInstances, numberOfViews ) >
<!ELEMENT objects ( instances | beginX | beginY | destination | id | image | isChildTemplatedOne | isMOtherTemplate |
lineColor | lineHeight | lineLength | linesEndingArray | linesStartingArray | motherTemplate | name | par | source | type |
valueObjArr | x | y )* >
<!ELEMENT instances ( id, name, isMOtherTemplate, isChildTemplatedOne, motherTemplate, par+, valueObjArr? ) >

<!ELEMENT valueObjArr ( id, name, haschanged, isCombo, combeValue ) >

<!ELEMENT par ( name, type, option* ) >

<!ELEMENT isCombo ( #PCDATA ) >

<!ELEMENT beginX ( #PCDATA ) >

<!ELEMENT beginY ( #PCDATA ) >

<!ELEMENT combeValue ( #PCDATA ) >

<!ELEMENT destination ( #PCDATA ) >

<!ELEMENT haschanged ( #PCDATA ) >

<!ELEMENT id ( #PCDATA ) >

<!ELEMENT image ( #PCDATA ) >

<!ELEMENT isChildTemplatedOne ( #PCDATA ) >

<!ELEMENT isMOtherTemplate ( #PCDATA ) >

<!ELEMENT lineColor ( #PCDATA ) >

<!ELEMENT lineHeight ( #PCDATA ) >

<!ELEMENT lineLength ( #PCDATA ) >

<!ELEMENT linesEndingArray ( name* ) >

<!ELEMENT linesStartingArray ( name* ) >

<!ELEMENT motherTemplate ( #PCDATA ) >

<!ELEMENT name ( #PCDATA ) >

<!ELEMENT numberOfInstances ( #PCDATA ) >

<!ELEMENT numberOfViews ( #PCDATA ) >

<!ELEMENT numberofnodes ( #PCDATA ) >

<!ELEMENT option ( #PCDATA ) >

<!ELEMENT source ( #PCDATA ) >

<!ELEMENT templates ( #PCDATA ) >

<!ELEMENT type ( #PCDATA ) >

<!ELEMENT x ( #PCDATA ) >

<!ELEMENT y ( #PCDATA ) >
```

ANEXO VIII – XML DE EXEMPLO DO FICHEIRO PROPRIETÁRIO DO VND (1.º CASO DE USO)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nsdlObjects SYSTEM "output.dtd" >
<root>
  <objects>
    <type>links</type>
    <id>fiber</id>
    <name>fiber_17</name>
    <source>Core_router_7</source>
    <destination>Core_router_6</destination>
    <isMOtherTemplate>>false</isMOtherTemplate>
    <isChildTemplatedOne>>false</isChildTemplatedOne>
    <motherTemplate></motherTemplate>
    <lineColor>222</lineColor>
    <beginX>328.5</beginX>
    <beginY>213</beginY>
    <lineLength>399.5</lineLength>
    <lineHeight>105</lineHeight>
    <par>
      <name>id</name>
      <type>string</type>
    </par>
    <par>
      <name>name</name>
      <type>string</type>
    </par>
    <par>
      <name>source</name>
      <type>string</type>
    </par>
    <par>
      <name>destination</name>
      <type>string</type>
    </par>
    <par>
      <name>type</name>
      <type>options</type>
      <option>simplex</option>
      <option>duplex</option>
    </par>
    <par>
      <name>bandwidth</name>
      <type>options</type>
      <option>100Mb</option>
      <option>1Gb</option>
      <option>10Gb</option>
      <option>100Gb</option>
      <option>300Gb</option>
    </par>
    <par>
      <name>delay</name>
      <type>string</type>
    </par>
    <valueObjArr>
      <id>type</id>
      <name>1</name>
      <haschanged>>false</haschanged>
      <lsCombo>true</lsCombo>
      <combeValue>duplex</combeValue>
    </valueObjArr>
    <valueObjArr>
      <id>bandwidth</id>
```

```

        <name>1</name>
        <haschanged>false</haschanged>
        <IsCombo>true</IsCombo>
        <combeValue>1Gb</combeValue>
    </valueObjArr>
</objects>
<objects>
    <type>links</type>
    <id>fiber</id>
    <name>fiber_16</name>
    <source>Core_router_7</source>
    <destination>Core_router_2</destination>
    <isMOtherTemplate>false</isMOtherTemplate>
    <isChildTemplatedOne>false</isChildTemplatedOne>
    <motherTemplate></motherTemplate>
    <lineColor>222</lineColor>
    <beginX>328.5</beginX>
    <beginY>213</beginY>
    <lineLength>275.5</lineLength>
    <lineHeight>105</lineHeight>
    <par>
        <name>id</name>
        <type>string</type>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
    </par>
    <par>
        <name>source</name>
        <type>string</type>
    </par>
    <par>
        <name>destination</name>
        <type>string</type>
    </par>
    <par>
        <name>type</name>
        <type>options</type>
        <option>simplex</option>
        <option>duplex</option>
    </par>
    <par>
        <name>bandwidth</name>
        <type>options</type>
        <option>100Mb</option>
        <option>1Gb</option>
        <option>10Gb</option>
        <option>100Gb</option>
        <option>300Gb</option>
    </par>
    <par>
        <name>delay</name>
        <type>string</type>
    </par>
    <valueObjArr>
        <id>type</id>
        <name>1</name>
        <haschanged>false</haschanged>
        <IsCombo>true</IsCombo>
        <combeValue>duplex</combeValue>
    </valueObjArr>
    <valueObjArr>
        <id>bandwidth</id>
        <name>1</name>
        <haschanged>false</haschanged>
        <IsCombo>true</IsCombo>
        <combeValue>1Gb</combeValue>
    </valueObjArr>
</objects>

```

```

<objects>
  <type>links</type>
  <id>fiber</id>
  <name>fiber_14</name>
  <source>Core_router_2</source>
  <destination>Core_router_6</destination>
  <isMOtherTemplate>>false</isMOtherTemplate>
  <isChildTemplatedOne>>false</isChildTemplatedOne>
  <motherTemplate></motherTemplate>
  <lineColor>222</lineColor>
  <beginX>275.5</beginX>
  <beginY>105</beginY>
  <lineLength>399.5</lineLength>
  <lineHeight>105</lineHeight>
  <par>
    <name>id</name>
    <type>string</type>
  </par>
  <par>
    <name>name</name>
    <type>string</type>
  </par>
  <par>
    <name>source</name>
    <type>string</type>
  </par>
  <par>
    <name>destination</name>
    <type>string</type>
  </par>
  <par>
    <name>type</name>
    <type>options</type>
    <option>simplex</option>
    <option>duplex</option>
  </par>
  <par>
    <name>bandwidth</name>
    <type>options</type>
    <option>100Mb</option>
    <option>1Gb</option>
    <option>10Gb</option>
    <option>100Gb</option>
    <option>300Gb</option>
  </par>
  <par>
    <name>delay</name>
    <type>string</type>
  </par>
  <valueObjArr>
    <id>type</id>
    <name>1</name>
    <haschanged>>false</haschanged>
    <isCombo>>true</isCombo>
    <combeValue>duplex</combeValue>
  </valueObjArr>
  <valueObjArr>
    <id>bandwidth</id>
    <name>1</name>
    <haschanged>>false</haschanged>
    <isCombo>>true</isCombo>
    <combeValue>1Gb</combeValue>
  </valueObjArr>
</objects>
<objects>
  <type>links</type>
  <id>ethernet</id>
  <name>ethernet_13</name>
  <source>Core_router_7</source>
  <destination>computer_10</destination>

```

```

<isMOtherTemplate>>false</isMOtherTemplate>
<isChildTemplatedOne>>false</isChildTemplatedOne>
<motherTemplate></motherTemplate>
<lineColor>0</lineColor>
<beginX>328.5</beginX>
<beginY>213</beginY>
<lineLength>330.5</lineLength>
<lineHeight>358</lineHeight>
<par>
  <name>id</name>
  <type>string</type>
</par>
<par>
  <name>name</name>
  <type>string</type>
</par>
<par>
  <name>source</name>
  <type>string</type>
</par>
<par>
  <name>destination</name>
  <type>string</type>
</par>
<par>
  <name>type</name>
  <type>options</type>
  <option>simplex</option>
  <option>duplex</option>
</par>
<par>
  <name>bandwidth</name>
  <type>options</type>
  <option>1Mb</option>
  <option>10Mb</option>
  <option>100Mb</option>
  <option>1Gb</option>
</par>
<par>
  <name>delay</name>
  <type>string</type>
</par>
<valueObjArr>
  <id>bandwidth</id>
  <name>2</name>
  <haschanged>>false</haschanged>
  <isCombo>>true</isCombo>
  <combeValue>100Mb</combeValue>
</valueObjArr>
<valueObjArr>
  <id>type</id>
  <name>1</name>
  <haschanged>>false</haschanged>
  <isCombo>>true</isCombo>
  <combeValue>duplex</combeValue>
</valueObjArr>
</objects>
<objects>
  <type>links</type>
  <id>ethernet</id>
  <name>ethernet_12</name>
  <source>Core_router_6</source>
  <destination>computer_9</destination>
  <isMOtherTemplate>>false</isMOtherTemplate>
  <isChildTemplatedOne>>false</isChildTemplatedOne>
  <motherTemplate></motherTemplate>
  <lineColor>0</lineColor>
  <beginX>399.5</beginX>
  <beginY>105</beginY>
  <lineLength>575</lineLength>

```

```

<lineHeight>145</lineHeight>
<par>
  <name>id</name>
  <type>string</type>
</par>
<par>
  <name>name</name>
  <type>string</type>
</par>
<par>
  <name>source</name>
  <type>string</type>
</par>
<par>
  <name>destination</name>
  <type>string</type>
</par>
<par>
  <name>type</name>
  <type>options</type>
  <option>simplex</option>
  <option>duplex</option>
</par>
<par>
  <name>bandwidth</name>
  <type>options</type>
  <option>1Mb</option>
  <option>10Mb</option>
  <option>100Mb</option>
  <option>1Gb</option>
</par>
<par>
  <name>delay</name>
  <type>string</type>
</par>
<valueObjArr>
  <id>type</id>
  <name>1</name>
  <haschanged>>false</haschanged>
  <IsCombo>>true</IsCombo>
  <combeValue>duplex</combeValue>
</valueObjArr>
<valueObjArr>
  <id>bandwidth</id>
  <name>2</name>
  <haschanged>>false</haschanged>
  <IsCombo>>true</IsCombo>
  <combeValue>100Mb</combeValue>
</valueObjArr>
</objects>
<objects>
  <type>links</type>
  <id>ethernet</id>
  <name>ethernet_11</name>
  <source>Core_router_6</source>
  <destination>computer_8</destination>
  <isMOtherTemplate>>false</isMOtherTemplate>
  <isChildTemplatedOne>>false</isChildTemplatedOne>
  <motherTemplate></motherTemplate>
  <lineColor>0</lineColor>
  <beginX>399.5</beginX>
  <beginY>105</beginY>
  <lineLength>568</lineLength>
  <lineHeight>59</lineHeight>
  <par>
    <name>id</name>
    <type>string</type>
  </par>
  <par>
    <name>name</name>

```

```

        <type>string</type>
    </par>
    <par>
        <name>source</name>
        <type>string</type>
    </par>
    <par>
        <name>destination</name>
        <type>string</type>
    </par>
    <par>
        <name>type</name>
        <type>options</type>
        <option>simplex</option>
        <option>duplex</option>
    </par>
    <par>
        <name>bandwidth</name>
        <type>options</type>
        <option>1Mb</option>
        <option>10Mb</option>
        <option>100Mb</option>
        <option>1Gb</option>
    </par>
    <par>
        <name>delay</name>
        <type>string</type>
    </par>
    <valueObjArr>
        <id>type</id>
        <name>1</name>
        <haschanged>>false</haschanged>
        <isCombo>>true</isCombo>
        <combeValue>duplex</combeValue>
    </valueObjArr>
    <valueObjArr>
        <id>bandwidth</id>
        <name>2</name>
        <haschanged>>false</haschanged>
        <isCombo>>true</isCombo>
        <combeValue>100Mb</combeValue>
    </valueObjArr>
</objects>
<objects>
    <type>links</type>
    <id>ethernet</id>
    <name>ethernet_5</name>
    <source>computer_4</source>
    <destination>Core_router_2</destination>
    <isMOtherTemplate>>false</isMOtherTemplate>
    <isChildTemplatedOne>>false</isChildTemplatedOne>
    <motherTemplate></motherTemplate>
    <lineColor>0</lineColor>
    <beginX>109</beginX>
    <beginY>155</beginY>
    <lineLength>275.5</lineLength>
    <lineHeight>105</lineHeight>
    <par>
        <name>id</name>
        <type>string</type>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
    </par>
    <par>
        <name>source</name>
        <type>string</type>
    </par>
    <par>
        <name>source</name>
        <type>string</type>
    </par>
    <par>
        <name>source</name>
        <type>string</type>
    </par>
    <par>
        <name>source</name>
        <type>string</type>
    </par>

```

```

        <name>destination</name>
        <type>string</type>
    </par>
    <par>
        <name>type</name>
        <type>options</type>
        <option>simplex</option>
        <option>duplex</option>
    </par>
    <par>
        <name>bandwidth</name>
        <type>options</type>
        <option>1Mb</option>
        <option>10Mb</option>
        <option>100Mb</option>
        <option>1Gb</option>
    </par>
    <par>
        <name>delay</name>
        <type>string</type>
    </par>
    <valueObjArr>
        <id>type</id>
        <name>1</name>
        <haschanged>>false</haschanged>
        <lsCombo>>true</lsCombo>
        <combeValue>duplex</combeValue>
    </valueObjArr>
    <valueObjArr>
        <id>bandwidth</id>
        <name>2</name>
        <haschanged>>false</haschanged>
        <lsCombo>>true</lsCombo>
        <combeValue>100Mb</combeValue>
    </valueObjArr>
</objects>
<objects>
    <type>links</type>
    <id>ethernet</id>
    <name>ethernet_3</name>
    <source>computer_1</source>
    <destination>Core_router_2</destination>
    <isMOtherTemplate>>false</isMOtherTemplate>
    <isChildTemplatedOne>>false</isChildTemplatedOne>
    <motherTemplate></motherTemplate>
    <lineColor>0</lineColor>
    <beginX>114</beginX>
    <beginY>65</beginY>
    <lineLength>275.5</lineLength>
    <lineHeight>105</lineHeight>
    <par>
        <name>id</name>
        <type>string</type>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
    </par>
    <par>
        <name>source</name>
        <type>string</type>
    </par>
    <par>
        <name>destination</name>
        <type>string</type>
    </par>
    <par>
        <name>type</name>
        <type>options</type>
        <option>simplex</option>

```

```

        <option>duplex</option>
    </par>
    <par>
        <name>bandwidth</name>
        <type>options</type>
        <option>1Mb</option>
        <option>10Mb</option>
        <option>100Mb</option>
        <option>1Gb</option>
    </par>
    <par>
        <name>delay</name>
        <type>string</type>
    </par>
    <valueObjArr>
        <id>type</id>
        <name>1</name>
        <haschanged>false</haschanged>
        <IsCombo>true</IsCombo>
        <combeValue>duplex</combeValue>
    </valueObjArr>
    <valueObjArr>
        <id>bandwidth</id>
        <name>2</name>
        <haschanged>false</haschanged>
        <IsCombo>true</IsCombo>
        <combeValue>100Mb</combeValue>
    </valueObjArr>
</objects>
<objects>
    <type>nodes</type>
    <id>computer</id>
    <name>computer_1</name>
    <image>computer.png</image>
    <isMOtherTemplate>false</isMOtherTemplate>
    <isChildTemplatedOne>false</isChildTemplatedOne>
    <motherTemplate></motherTemplate>
    <x>76</x>
    <y>42</y>
    <par>
        <name>id</name>
        <type>string</type>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
    </par>
    <par>
        <name>active</name>
        <type>boolean</type>
    </par>
    <par>
        <name>notes</name>
        <type>string</type>
    </par>
    <par>
        <name>interface</name>
        <type>instance</type>
    </par>
    <par>
        <name>protocol</name>
        <type>instance</type>
    </par>
    <par>
        <name>application</name>
        <type>instance</type>
    </par>
    <par>
        <name>role</name>
        <type>options</type>

```

```

        <option>server</option>
        <option>peer</option>
    </par>
    <valueObjArr>
        <id>name</id>
        <name>Servidor Sede</name>
        <haschanged>>false</haschanged>
        <IsCombo>>false</IsCombo>
        <combeValue>>null</combeValue>
    </valueObjArr>
    <Instances>
        <id>application</id>
        <name>application_2</name>
        <isMOtherTemplate>>false</isMOtherTemplate>
        <isChildTemplatedOne>>false</isChildTemplatedOne>
        <motherTemplate></motherTemplate>
        <par>
            <name>id</name>
            <type>string</type>
        </par>
        <par>
            <name>active</name>
            <type>boolean</type>
        </par>
        <par>
            <name>name</name>
            <type>string</type>
        </par>
        <par>
            <name>role</name>
            <type>options</type>
            <option>server</option>
            <option>client</option>
        </par>
        <par>
            <name>rate</name>
            <type>string</type>
        </par>
        <par>
            <name>packetSize</name>
            <type>string</type>
        </par>
        <valueObjArr>
            <id>name</id>
            <name>HTTP</name>
            <haschanged>>false</haschanged>
            <IsCombo>>false</IsCombo>
            <combeValue>>null</combeValue>
        </valueObjArr>
    </Instances>
    <Instances>
        <id>application</id>
        <name>application_3</name>
        <isMOtherTemplate>>false</isMOtherTemplate>
        <isChildTemplatedOne>>false</isChildTemplatedOne>
        <motherTemplate></motherTemplate>
        <par>
            <name>id</name>
            <type>string</type>
        </par>
        <par>
            <name>active</name>
            <type>boolean</type>
        </par>
        <par>
            <name>name</name>
            <type>string</type>
        </par>
        <par>
            <name>role</name>

```

```

                <type>options</type>
                <option>server</option>
                <option>client</option>
            </par>
        <par>
            <name>rate</name>
            <type>string</type>
        </par>
        <par>
            <name>packetSize</name>
            <type>string</type>
        </par>
    </Instances>
    <linesStartingArray>
        <name>ethernet_3</name>
    </linesStartingArray>
    <linesEndingArray></linesEndingArray>
</objects>
<objects>
    <type>nodes</type>
    <id>Core_router</id>
    <name>Core_router_2</name>
    <image>router.png</image>
    <isMOtherTemplate>false</isMOtherTemplate>
    <isChildTemplatedOne>false</isChildTemplatedOne>
    <motherTemplate></motherTemplate>
    <x>243</x>
    <y>82</y>
    <par>
        <name>id</name>
        <type>string</type>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
    </par>
    <par>
        <name>notes</name>
        <type>string</type>
    </par>
    <par>
        <name>interface</name>
        <type>instance</type>
    </par>
    <par>
        <name>protocol</name>
        <type>instance</type>
    </par>
    <par>
        <name>phbs</name>
        <type>instance</type>
    </par>
    <par>
        <name>scheduler</name>
        <type>instance</type>
    </par>
    <valueObjArr>
        <id>name</id>
        <name>Router Sede</name>
        <haschanged>false</haschanged>
        <isCombo>false</isCombo>
        <combeValue>null</combeValue>
    </valueObjArr>
    <linesStartingArray>
        <name>fiber_14</name>
    </linesStartingArray>
    <linesEndingArray>
        <name>ethernet_3</name>
        <name>ethernet_5</name>
        <name>fiber_16</name>

```

```

        </linesEndingArray>
</objects>
<objects>
  <type>nodes</type>
  <id>computer</id>
  <name>computer_4</name>
  <image>computer.png</image>
  <isMOtherTemplate>>false</isMOtherTemplate>
  <isChildTemplatedOne>>false</isChildTemplatedOne>
  <motherTemplate></motherTemplate>
  <x>63</x>
  <y>132</y>
  <par>
    <name>id</name>
    <type>string</type>
  </par>
  <par>
    <name>name</name>
    <type>string</type>
  </par>
  <par>
    <name>active</name>
    <type>boolean</type>
  </par>
  <par>
    <name>notes</name>
    <type>string</type>
  </par>
  <par>
    <name>interface</name>
    <type>instance</type>
  </par>
  <par>
    <name>protocol</name>
    <type>instance</type>
  </par>
  <par>
    <name>application</name>
    <type>instance</type>
  </par>
  <par>
    <name>role</name>
    <type>options</type>
    <option>server</option>
    <option>peer</option>
  </par>
  <valueObjArr>
    <id>name</id>
    <name>Computador Sede</name>
    <haschanged>>false</haschanged>
    <IsCombo>>false</IsCombo>
    <combeValue>>null</combeValue>
  </valueObjArr>
  <linesStartingArray>
    <name>ethernet_5</name>
  </linesStartingArray>
  <linesEndingArray></linesEndingArray>
</objects>
<objects>
  <type>nodes</type>
  <id>Core_router</id>
  <name>Core_router_6</name>
  <image>router.png</image>
  <isMOtherTemplate>>false</isMOtherTemplate>
  <isChildTemplatedOne>>false</isChildTemplatedOne>
  <motherTemplate></motherTemplate>
  <x>367</x>
  <y>82</y>
  <par>
    <name>id</name>

```

```

        <type>string</type>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
    </par>
    <par>
        <name>notes</name>
        <type>string</type>
    </par>
    <par>
        <name>interface</name>
        <type>instance</type>
    </par>
    <par>
        <name>protocol</name>
        <type>instance</type>
    </par>
    <par>
        <name>phbs</name>
        <type>instance</type>
    </par>
    <par>
        <name>scheduler</name>
        <type>instance</type>
    </par>
    <valueObjArr>
        <id>name</id>
        <name>Router Site1</name>
        <haschanged>>false</haschanged>
        <isCombo>>false</isCombo>
        <combeValue>>null</combeValue>
    </valueObjArr>
    <linesStartingArray>
        <name>ethernet_11</name>
        <name>ethernet_12</name>
    </linesStartingArray>
    <linesEndingArray>
        <name>fiber_14</name>
        <name>fiber_17</name>
    </linesEndingArray>
</objects>
<objects>
    <type>nodes</type>
    <id>Core_router</id>
    <name>Core_router_7</name>
    <image>router.png</image>
    <isMOtherTemplate>>false</isMOtherTemplate>
    <isChildTemplatedOne>>false</isChildTemplatedOne>
    <motherTemplate></motherTemplate>
    <x>296</x>
    <y>190</y>
    <par>
        <name>id</name>
        <type>string</type>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
    </par>
    <par>
        <name>notes</name>
        <type>string</type>
    </par>
    <par>
        <name>interface</name>
        <type>instance</type>
    </par>
    <par>
        <name>protocol</name>

```

```

        <type>instance</type>
    </par>
    <par>
        <name>phbs</name>
        <type>instance</type>
    </par>
    <par>
        <name>scheduler</name>
        <type>instance</type>
    </par>
    <valueObjArr>
        <id>name</id>
        <name>Router Site2</name>
        <haschanged>>false</haschanged>
        <isCombo>>false</isCombo>
        <combeValue>>null</combeValue>
    </valueObjArr>
    <linesStartingArray>
        <name>ethernet_13</name>
        <name>fiber_16</name>
        <name>fiber_17</name>
    </linesStartingArray>
    <linesEndingArray></linesEndingArray>
</objects>
<objects>
    <type>nodes</type>
    <id>computer</id>
    <name>computer_8</name>
    <image>computer.png</image>
    <isMOtherTemplate>>false</isMOtherTemplate>
    <isChildTemplatedOne>>false</isChildTemplatedOne>
    <motherTemplate></motherTemplate>
    <x>530</x>
    <y>36</y>
    <par>
        <name>id</name>
        <type>string</type>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
    </par>
    <par>
        <name>active</name>
        <type>boolean</type>
    </par>
    <par>
        <name>notes</name>
        <type>string</type>
    </par>
    <par>
        <name>interface</name>
        <type>instance</type>
    </par>
    <par>
        <name>protocol</name>
        <type>instance</type>
    </par>
    <par>
        <name>application</name>
        <type>instance</type>
    </par>
    <par>
        <name>role</name>
        <type>options</type>
        <option>server</option>
        <option>peer</option>
    </par>
    <valueObjArr>
        <id>name</id>

```

```

        <name>Servidor Site1</name>
        <haschanged>>false</haschanged>
        <IsCombo>>false</IsCombo>
        <combeValue>>null</combeValue>
    </valueObjArr>
    <Instances>
        <id>application</id>
        <name>application_1</name>
        <isMOtherTemplate>>false</isMOtherTemplate>
        <isChildTemplatedOne>>false</isChildTemplatedOne>
        <motherTemplate></motherTemplate>
        <par>
            <name>id</name>
            <type>string</type>
        </par>
        <par>
            <name>active</name>
            <type>boolean</type>
        </par>
        <par>
            <name>name</name>
            <type>string</type>
        </par>
        <par>
            <name>role</name>
            <type>options</type>
            <option>server</option>
            <option>client</option>
        </par>
        <par>
            <name>rate</name>
            <type>string</type>
        </par>
        <par>
            <name>packetsize</name>
            <type>string</type>
        </par>
    </Instances>
    <linesStartingArray></linesStartingArray>
    <linesEndingArray>
        <name>ethernet_11</name>
    </linesEndingArray>
</objects>
<objects>
    <type>nodes</type>
    <id>computer</id>
    <name>computer_9</name>
    <image>computer.png</image>
    <isMOtherTemplate>>false</isMOtherTemplate>
    <isChildTemplatedOne>>false</isChildTemplatedOne>
    <motherTemplate></motherTemplate>
    <x>529</x>
    <y>122</y>
    <par>
        <name>id</name>
        <type>string</type>
    </par>
    <par>
        <name>name</name>
        <type>string</type>
    </par>
    <par>
        <name>active</name>
        <type>boolean</type>
    </par>
    <par>
        <name>notes</name>
        <type>string</type>
    </par>
</objects>

```

```

        <name>interface</name>
        <type>instance</type>
</par>
<par>
        <name>protocol</name>
        <type>instance</type>
</par>
<par>
        <name>application</name>
        <type>instance</type>
</par>
<par>
        <name>role</name>
        <type>options</type>
        <option>server</option>
        <option>peer</option>
</par>
<valueObjArr>
        <id>name</id>
        <name>Computador Site1</name>
        <haschanged>false</haschanged>
        <IsCombo>false</IsCombo>
        <combeValue>null</combeValue>
</valueObjArr>
<linesStartingArray></linesStartingArray>
<linesEndingArray>
        <name>ethernet_12</name>
</linesEndingArray>
</objects>
<objects>
        <type>nodes</type>
        <id>computer</id>
        <name>computer_10</name>
        <image>computer.png</image>
        <isMOtherTemplate>false</isMOtherTemplate>
        <isChildTemplatedOne>false</isChildTemplatedOne>
        <motherTemplate></motherTemplate>
        <x>285</x>
        <y>335</y>
        <par>
                <name>id</name>
                <type>string</type>
        </par>
        <par>
                <name>name</name>
                <type>string</type>
        </par>
        <par>
                <name>active</name>
                <type>boolean</type>
        </par>
        <par>
                <name>notes</name>
                <type>string</type>
        </par>
        <par>
                <name>interface</name>
                <type>instance</type>
        </par>
        <par>
                <name>protocol</name>
                <type>instance</type>
        </par>
        <par>
                <name>application</name>
                <type>instance</type>
        </par>
        <par>
                <name>role</name>
                <type>options</type>

```

```
        <option>server</option>
        <option>peer</option>
    </par>
    <valueObjArr>
        <id>name</id>
        <name>Computador site2</name>
        <haschanged>false</haschanged>
        <IsCombo>false</IsCombo>
        <combeValue>null</combeValue>
    </valueObjArr>
    <linesStartingArray></linesStartingArray>
    <linesEndingArray>
        <name>ethernet_13</name>
    </linesEndingArray>
</objects>
<templates></templates>
<numberofnodes>17</numberofnodes>
<numberOfInstances>3</numberOfInstances>
<numberOfViews>0</numberOfViews>
</root>
```




Os melhores RUMOS para os Cidadãos da Região



REGIÃO AUTÓNOMA DA
MADEIRA



REPÚBLICA PORTUGUESA



UNIÃO EUROPEIA
FSE