UNIVERSIDADE ESTADUAL DE PERNAMBUCO - CAMPOS CARUARU
SISTEMA DE INFORMAÇÃO

Matheus Oliveira Farias

DESENVOLVIMENTO DE EXTENSÃO PARA LEITURA DE TOPOLOGIAS DE REDES NO FORMATO GML PARA A FERRAMENTA NETLOGO

Caruaru

2014

UNIVERSIDADE ESTADUAL DE PERNAMBUCO - CAMPOS CARUARU
SISTEMA DE INFORMAÇÂO

Matheus Oliveira Farias

DESENVOLVIMENTO DE EXTENSÃO PARA LEITURA DE TOPOLOGIAS DE REDES NO FORMATO GML PARA A FERRAMENTA NETLOGO

Monografia apresentada ao Curso de Sistema de Informação da Universidade Estadual de Pernambuco – Campus Caruaru, como requisito parcial para obtenção de grau de Bacharelado.

Orientadora: Professora MSc. Patricia Takako Endo

Caruaru

2014

Monografia de Graduação apresentada por **Matheus Oliveira Farias** do Curso de Graduação em Sistemas de Informação da Faculdade de Ciência e Tecnologia de Caruaru – Universidade de Pernambuco, sob o título "Desenvolvimento de Extensão para Leitura da Topologia de Redes no Formato GML para a Ferramenta NetLogo", orientada pela Prof.ª **Prof. Patricia Takako Endo** e aprovada pela Banca Examinadora formada pelos professores:

Ivson Henrique Bezerra dos Santos

Prof. Ivson Henrique Bezerra dos Santos
Departamento de Sistemas de Informação / UPE

Patricia Takako Endo

Prof.ª. Patricia Takako Endo
Departamento de Sistemas de Informação / UPE

Visto e permitida a impressão.
Caruaru, 26 de junho de 2014.

Patricia Takako Endo

Prof. Patricia Takako Endo
Coordenador do Curso de Bacharelado em Sistemas de Informação da
Faculdade de Ciência e Tecnologia de Caruaru – Universidade de Pernambuco.

Patricia Takako E...
Coord Curso Sistemas de Informaçã...
Mat. 113751

Dedico este trabalho aos meus pais Tadeu e Vanusa, a meus Irmãos Débora e Rafael, a minha namorada, ao Money A Lot e a todos os meus amigos que estiveram comigo durante esses quatro anos na faculdade.

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus, que sempre me guiou durante toda a minha vida.

Em segundo gostaria de agradecer a minha orientadora Prof. Patricia Endo, que me ajudou bastante durante o desenvolvimento do meu T.C.C e pela sua dedicação com a faculdade e seus alunos.

Por último, mas, não menos importante à meus pais Tadeu e Vanusa, pelo o amor, carinho e ensinamentos que levarei comigo durante toda a minha vida;

A meus irmãos Débora e Rafael, pelos momentos de alegria e de arengas que tivemos juntos;

A minha namorada Andréa, pelo seu companheirismo, paciência e amor comigo;

A todos os meus amigos, em especial a Eolânda e Hugo, que me aguentaram durante esses quatro anos;

E a todos da UPE que me acompanharam durante esses anos, me ajudando a passar pelos desafios dessa fase da minha vida.

Não tenho palavras, nem páginas suficientes, para agradecer de maneira adequada a essas pessoas, por isso ofereço-lhes minha eterna gratidão e meu "muito obrigado, por tudo".

RESUMO

A Internet e as redes que a compõe sem dúvida fornecem serviços muito úteis para a sociedade nos dias atuais. Porem a cada ano essas redes são sobrecarregadas com a chegada de novos usuários e novos serviços, colocando assim um desafio para os desenvolvedores de sistemas de comunicação de aumentar a capacidade das redes sem interromper seu funcionamento. Sendo assim, a simulação se apresenta como uma forma de analisar os novos meios de comunicação idealizados pelos desenvolvedores, para obter resultados fiéis mais próximos ao de um sistema real. Para realizar as simulações se faz necessário o uso de ferramentas que auxiliem na modelagem e no processamento da simulação, podemos citar como uma destas a ferramenta NetLogo, que permite a modelagem simples de diversos sistemas para realização de simulações. Contudo a ferramenta NetLogo ainda exige bastante esforço por parte dos seus usuários, para a criação de um modelo que abranja as principais características de uma rede. Para diminuir este esforço o presente trabalho tem como objetivo desenvolver uma extensão para a ferramenta NetLogo que permita a importação de redes descritas no formato GML (*Graph Modelling Language).*

Palavras-Chaves: Extensão, NetLogo, Simulação, GML, Internet Topology Zoo**.**

Sumario

1. INTRODUÇÃO 8

1.1OBJETIVOS 10

1.1.1Objetivo geral 10

1.1.2 Objetivos Específicos 10

1.2 JUSTIFICATIVA 11

2. REFERENCIAL TEÓRICO 12

2.1 SISTEMAS 12

2.1.1 Tipos de sistemas 12

2.1.2 Padrões de desempenho de sistemas 13

2.1.3 Variáveis e parâmetros de sistemas 13

2.2 SIMULAÇÕES 14

2.2.1 Modelos 17

2.2.2 Tipos de simulação 19

2.4 INTERNET TOPOLOGY ZOO 22

2.4.1 GML (Graph Modeling language) 23

2.5 NETLOGO 27

2.5.1 Linguagem de programação Scala 28

3. METODOLOGIA 31

3.1.1 QUANTO AOS FINS 31

3.1.2 QUANTO AOS MEIOS 31

4. DESCRIÇÃO DA EXTENSÃO 33

5. ANALISE DA EXTENSÃO 37

6. CONCLUSÃO E TRABALHOS FUTUROS 41

REFERÊNCIAIS 42

LISTA DE IMAGENS

Figura 1: Representação de um modelo de sistema......17
Figura 2: Gráfico simples gerado pelo yEd.....24
Figura 3: Gráfico da rede Arpnet 1996 27
Figura 4: Modelo UML da Extensão ImportGML. 33
Figura 5: User Case da extensão ImportGML. 34
Figura 6: Matriz de conexão da rede Ipê. 39

LISTA DE TABELAS

Tabela 1: Classificações de sistemas.....12

LISTA DE SIGLAS

CCL	Center for Connected Learning
GD	Graph Drawing: Symposium on Graph Drawing
GML	Graph Modelling Language
HTML	HyperText Markup Language
JVM	Java Virtual Machine
PoPs	Pontos de Presença
RNP	Rede Nacional de Ensino e Pesquisa
RTF	Rich Text Format
SGML	Standard Generalized Markup Language
XML	eXtensible Markup Language

1. INTRODUÇÃO

É difícil imaginar nossas vidas sem acesso à Internet, pois a mesma disponibiliza diversos serviços que auxiliam desde um usuário comum até grandes empresas. A Internet pode ser descrita de diversas maneiras. Kurose (2010) por exemplo, descreve a mesma de duas maneiras, sendo elas: a Internet como uma rede que conectada milhares de dispositivos distintos como TVs, celulares, tabletes, laptops entre outros; e como uma infraestrutura que fornece serviços a aplicações, por exemplo, correio eletrônico e navegação na WEB. Também pode ser descrita como uma rede que interliga várias redes distintas.

Hoje existem mais de 600 milhões de dispositivos finais que estão interligados pela Internet e segundo PriMetrica (2009), estima-se que 10 Terabytes por segundo de capacidade internacional sejam utilizados todos os anos. Esses números não param de crescer fazendo-se necessário o aumento das capacidades tecnológicas, sejam elas de *hardware* ou de *software*.

Para o melhoramento ou desenvolvimento de equipamentos com maiores taxas de processamento e de comunicação, se faz necessário o empenho de grandes recursos financeiros e estudos de novos meios de comunicação. Já o desenvolvimento de novos métodos de comunicação e processamento, que não exijam tanto dos equipamentos utilizados atualmente, necessitam além dos recursos financeiros, estudos e simulações mais aprofundados, sendo estes de suma importância para garantir a qualidade e verificar se as soluções proposta irão atender as expectativas dos mesmos, antes de serem aplicadas de fato em redes existentes.

A qualidade dos métodos desenvolvidos possui grande importância, pois diminui o tempo de alocação e custos de desenvolvimento como afirma Pressman:

> "Se uma equipe de *software* enfatizar a qualidade em todas as atividades de engenharia de *software*, ela reduzirá a quantidade de reformulações que terá de fazer. Isso resulta em custos menores e mais importante ainda, menor tempo para a colocação do produto no mercado.". (Pressman, 2011, pg.358)

Existem várias técnicas para se obter a qualidade desejada nos produto desenvolvidos. Podemos citar como exemplo as técnicas de revisão, que são realizadas à medida que o produto é desenvolvido, ou os testes, que são realizados durante e após o desenvolvimento do produto.

Além de priorizar a qualidade os novos métodos devem passar por simulações, antes de serem implementadas em sistemas reais. A simulação é uma técnica importante, pois permite analisar como os métodos propostos para melhorar a capacidade da rede irão se comportar em um sistema real, podendo assim antecipar possíveis falhas de funcionamento e até mesmo conceber novas melhorias para os mesmos. Desta forma, a simulação se apresenta como um meio de verificar se as soluções propostas pelos novos métodos poderão ser implementadas em sistemas reais, evitando assim, transtornos como paralisação da rede e perda de receita por inatividade ou falhas de comunicação da mesma.

Para a realização de simulações de redes se faz necessário a modelagem da mesma com a suas características mais relevantes para a simulação. Podemos citar como exemplo dessas características a sua topologia, padrão em que os dispositivos da rede estão conectados de maneira física e lógica, a largura de banda e atraso entre os links. Existem vários tipos de topologias em uso, como topologia em estrela e anel, cada uma delas possui características que permite uma melhor implementação da rede em determinados cenários, podendo ainda, existir redes com mais de uma topologia para que o mesmo atenda todas as necessidades do cenário em que está descrita.

Com o intuito de mapear as principais redes existentes ao redor do mundo, a Universidade de Adelaide desenvolveu e mantém até hoje o projeto Internet Topology Zoo[1], que atualmente possui mais de 250 redes em dois formatos de arquivos: o GML e o GraphML. As redes fornecidas pelo o projeto Internet Topology Zoo são extremamente úteis e permitem que diversas ferramentas de modelagem de redes possam utilizá-la como entrada para a criação de um modelo de simulação. O NetLogo é uma das ferramentas que pode se beneficiar das redes mapeadas e fornecidas pelo Internet Topology Zoo, para acelerar o processo de modelagem e simulação em redes já mapeadas.

O NetLogo pode ser utilizado como ferramenta de simulação em diversas áreas por permitir a modelagem de vários sistemas, por gerar dados e estatísticas oriundas das simulações, podendo estas serem analisadas posteriormente, por possui várias extensões que auxiliam na modelagem e também por permitir a criação de novas extensões para a solução de problemas mais específicos dos usuários. Apesar da variedade de extensões e funcionalidades o mesmo ainda não possui uma extensão eficiente para importar mapas de redes, isto gera um grande empecilho para os usuários que desejam realizar simulações em sistemas já existentes,

[1] http://www.topology-zoo.org/index.html

pois os mesmo terão que modelá-las e configurá-las manualmente, causando um grande desperdício de tempo.

Partindo desses pressupostos a problemática desta monografia se apresenta da seguinte forma: de que maneira pode-se importar mapas de sistemas de redes no formato GML para a ferramenta NetLogo? O presente trabalho aponta como solução para a problemática acima, o desenvolvimento de uma extensão para a ferramenta NetLogo, que permita a importação de arquivos no formato GML para a geração de redes de maneira simples e eficiente dentro da ferramenta.

1.1Objetivos

Este tópico descreve quais são os resultados finais esperados e os meios para se alcançar os mesmos.

1.1.1Objetivo geral

Desenvolver uma extensão para a ferramenta NetLogo que permita a importação de topologias de redes descritas no formato GML.

1.1.2 Objetivos Específicos

Para se alcançar o objetivo geral, os seguintes objetivos específicos foram delineados:

- Realizar o levantamento bibliográfico sobre os tipos de arquivos que descrevem topologias de redes, sobre simulação, sobre a ferramenta NetLogo e sobre o projeto Internet Topology Zoo;
- Criar embasamento para desenvolver extensões para a ferramenta NetLogo;
- Desenvolver a extensão proposta;
- Aplicar testes na extensão desenvolvida;

- Criar a documentação da extensão, descrevendo suas funcionalidades e modo de utilização.

1.2 Justificativa

Apesar de ser possível modelar sistemas de redes existentes no mundo real utilizando-se a ferramenta NetLogo, esta tarefa pode acabar se tornando complexa e demandar muito tempo do usuário, devido aos detalhes e as diversas informações necessárias para se modelar uma rede mais complexa. Este trabalho pode ser simplificado com a importação das redes já mapeadas e disponibilizadas em formatos de arquivos como o GML.

Até o presente momento, a única extensão que permite a importação de topologias de redes para a ferramenta NetLogo é a NW-Extension, onde é possível importar arquivos no formato de texto, que contém uma matriz de conexão representando a rede; e no formato GraphML, arquivo baseado em XML (*eXtensible Markup Language*) que especifica a rede através de linguagem de marcação. Apesar de possuir diversas funcionalidades, a extensão apresenta alguns empecilhos com os seus modelos de importação, sendo eles:

Importação do arquivo no formato texto:

- É limitada apenas à disposição física dos nós da rede, não abrangendo outros aspectos como, por exemplo, custo dos enlaces;

Importação do arquivo no formato GraphML:

- Também é limitada apenas à disposição física dos nós da rede, não abrangendo outros aspectos como, por exemplo, custo dos enlaces;

Tendo em vista as limitações acima, torna-se claro a importância da criação de uma extensão que possibilite a importação de topologias de redes e suas características relevantes, como custo de enlaces, para a ferramenta NetLogo, permitindo aos usuários da ferramenta um desenvolvimento dos modelos para a simulação mais simples, rápidos e que contenham todos os dados necessários para que os resultados gerados ao final das simulações sejam os mais completos e próximos aos de uma rede real.

2. REFERENCIAL TEÓRICO

2.1 Sistemas

Segundo Stair e Reynolds(2006) um sistema é definido como um conjunto de elementos que interagem para alcançar objetivos. O funcionamento de um sistema é determinado pelos elementos e pela a relação que existe entre eles. Os sistemas possuem entradas, mecanismos de processamento, saídas e mecanismos de realimentação. As entradas são os valores que irão alimentar o sistema, o mecanismo de processamento utiliza os valores de entrada para gerar resultados, podendo estes serem utilizados por mecanismos de realimentação ou como valores para a saída do sistema.

Todos os sistemas estão dentro de um ambiente, este podendo interagir com o sistema ou não, o que define um sistema dentro de um ambiente é a fronteira do sistema, sendo esta uma linha imaginária que marca o que pertence ao sistema e o que não pertence.

2.1.1 Tipos de sistemas

Stair e Reynolds apresentam, na tabela a seguir, um resumo das diversas classificações e suas definições:

Tabela 1: Classificações de sistemas.

Simples	**Complexo**
Tem poucos componentes, e a relação ou interação entre os elementos é descomplicada e direta	Tem muitos elementos que são altamente relacionados e interconectados
Aberto	**Fechado**
Interage com o ambiente	Não interage com o ambiente
Estável	**Dinâmico**
Sofre muito poucas mudanças ao longo do tempo	Sofre mudanças rápidas e constantes ao longo do tempo
Adaptativo	**Não adaptativo**
Pode mudar em resposta a mudanças no ambiente	Não pode mudar em resposta a mudanças no ambiente
Permanente	**Temporário**
Existe por um período de tempo relativamente longo	Existe por um período de tempo relativamente curto

Fonte: Princípios de Sistemas de Informação. Stair e Reynolds (2006, pag. 9).

Os sistemas podem, também, possuir mais de uma classificação, por exemplo uma empresa de lava-jato poderia se classificar como simples, pois o relacionamento entre os seus componentes é descomplicada e direta, além de ser também classificada como estável e

permanente, pois o seu processo sofre poucas mudanças com o passar do tempo e o sistema existe há um longo período de tempo.

2.1.2 Padrões de desempenho de sistemas

Podemos identificar três maneiras principais de medir o desempenho e identificar padrões para sistemas, são eles: eficiência, eficácia e padrão de desempenho de sistemas.

A eficiência está relacionada com a produção do sistema. Trata-se de como realizar as tarefas, quanto maior a eficiência menor serão os gastos de produção e maior será a produtividade. Segundo Bio (2008, pág. 25) a eficiência "É definida pela relação entre volumes produzidos/recursos consumidos."

A eficácia está relacionada aos objetivos do sistema. Trata-se do que fazer, quanto maior a eficácia mais perto está o sistema de atingir seus objetivos. Segundo Bio (2008, pág. 25) a eficácia "É definida pela relação entre resultados pretendidos/resultados obtidos."

O padrão de desempenho de sistemas é o estabelecimento de padrões em que o sistema deve estar adequado, por exemplo, uma empresa de venda determina que sua meta de mensal de vendas é de 100 unidades do seu produto. Após o levantamento das vendas ficou constatado que a empresa atingiu sua meta e consequentemente está dentro do padrão estabelecido pela mesma. Stair e Reynolds (2006, pág. 9) define o padrão de desempenho de sistema como o "objetivo específico do sistema"

2.1.3 Variáveis e parâmetros de sistemas

Dentro de um sistema existem elementos que recebem valores. Podendo estes serem controlados pelo responsável do sistema, nestes caso denominado de variáveis do sistema, ou não controlados pelo responsável do sistema, sendo estes denominados parâmetros do sistema.

Stair e Reynolds (2006, pág. 10) diz que, “variável de sistema é a quantidade ou item que pode ser controlado pelo tomador de decisão e parâmetro de sistema é o valor ou quantidade que não pode ser controlado, como o custo de matéria-prima.”

Um exemplo bem simples seria quanto a uma empresa de creme dental, ela poderia cobrar o valor que achar conveniente para o seu produto (variável de sistema), porém ela não pode estabelecer o valor das matérias primas utilizadas para a criação do produto (parâmetro de sistema).

As variáveis de sistema podem ainda ser classificadas como endógenas (dependentes) e as exógenas (independentes). As endógenas são produzidas dentro do sistema ou resultantes de causas internas, elas mostram o estado do sistema e geram os resultados oriundos do mesmo. Já as exógenas são originadas por causas externas, elas são consideradas as variáveis de entrada do sistema em sistemas abertos.

2.2 Simulações

Segundo Shannon a simulação é definida como:

> “O processo de desenvolvimento de um modelo de um sistema real, e a condução de experimentos nesse modelo, com o propósito de entender o comportamento do sistema e/ou avaliar várias estratégias (com os limites impostos por um critério ou conjunto de critérios) para a operação do sistema” (Shannon, 1975, pág. 2)

Dessa forma, entende-se simulação como uma técnica da pesquisa operacional muito utilizada para analisar projetos e operações de sistemas complexos, por ser bastante flexível, poderosa e intuitiva. Esta técnica estuda os comportamentos de sistemas reais, sendo realizados experimentos através de modelos, que imitam totalmente ou parcialmente o sistema estudado.

A simulação pode ser empregada em diversas áreas sendo aplicada, de preferência, em sistemas estocásticos. Segue abaixo uma lista não exaustiva de sistemas aptos para a utilização da técnica de simulação:

- Sistemas administrativos:
  Seguradoras, Operadores de crédito, Financeiras;
- Sistemas de prestação de serviços diretos ao público:

Hospitais, Bancos, Restaurantes industriais e tipos de *fast food*, Serviços de emergência (polícia, bombeiros etc.), Serviços de assistência jurídica;

- Sistemas de produção:
  Manufatura e montagem, Movimentação de peças e matéria prima, Alocação de mão de obra, Áreas de armazenagem, Layout;
- Sistemas de transporte e estocagem:
  Redes de distribuição, Armazéns e entrepostos, Frotas;
- Sistemas computacionais:
  Redes de computadores, Redes de comunicação, Servidores de redes, Arquitetura de computadores, Sistemas operacionais, Gerenciadores de bases de dados.

A realização de simulações é geralmente recomendado em dois casos como diz Mello:

> ” De modo geral, o uso da simulação é recomendado principalmente em dois casos. Primeiro, quando a solução de problemas é muito cara ou mesmo impossível através de experimentos. E em segundo, quando os problemas são muito complexos para tratamento analítico.” (Mello, 2007, pg.04)

A dificuldade em ter acesso a certos sistemas fechados, longos períodos necessários para realização de experimentos, a impossibilidade de implementar soluções em sistemas muito grandes sem há certeza de que irá funcionar corretamente e a necessidade de antecipar os impactos de certos sistemas ao serem inseridos em ambientes distintos são exemplos de alguns outros casos que levam a utilização de simulações como um meio mais confiável de obter respostas, gerar hipóteses, antecipar fatos e impactos, e permitir a análise do sistema estudado. Geralmente utilizamos as simulações para obter resposta da seguinte pergunta: “o que aconteceria se?”.

Como vimos a técnica de simulação é recomendada para sistemas estocásticos e complexos, pois caso contrário, poderiam ser utilizadas outras técnicas como modelos matemáticos que realizam a modelagem do sistema real a parti de fórmulas matemáticas, abstraindo assim, a essência do problema e fornecendo as relações de causa-efeito do sistema de maneira mais simples e menos custosa do que a técnica de simulação. Contudo muitos sistemas possuem problemas com um nível de complexidade muito elevada para que se possam utilizar estas técnicas.

A técnica de simulação utiliza computadores para construir o modelo de simulação que melhor represente o sistema real, pelo fato da simulação trabalhar com uma quantidade enorme de dados e processos, devido ao nível de complexidade do modelo. Geralmente é

utilizada a técnica de simulação, pois a mesma se apresenta mais flexível permitindo respostas rápidas ao passo de várias modificações no modelo. Abaixo segue mais alguns fatores que incentivam a utilização da técnica de simulação demostrados por Mello (2007, pág. 4), sendo eles:

- Tempo: Os computadores são capazes de realizar vários experimentos em curto espaço de tempo, também podem realizar experimentos que levariam anos para serem executados em sistemas reais;
- Custo: Mesmo com os gastos em recursos humanos e equipamentos a simulação em computadores tendem a possuir um custo abaixo se comparado à simulações em sistemas reais ou em miniaturas dos sistemas reais;
- Impossibilidade de experimentação direta: Como dito acima existem situações em que os sistemas reais não permitem experimentos, por questões como acesso aos sistemas, tempo, segurança e por inexistência (sistemas ainda em construção);
- Visualização: Computadores podem gerar imagens, animações, tabelas e gráficos que possibilitam visualizar melhores os resultados gerados pela simulação;
- Repetição: Após o desenvolvimentos do modelo em computador, pode-se realizar n vezes a simulação;
- Interferência: Um sistema desenvolvido em computador pode realizar várias mudanças em seus eventos, de maneira simples, comparado a um sistema real.

Jain (1991, pág. 411) diz que a simulação requer pelo menos as seguintes 4 áreas de competência:

a) Liderança de Projeto: A capacidade de motivar, liderar e gerenciar os membros da equipe de simulação.
b) Modelagem e Estatística: A capacidade de identificar as características-chaves do sistema e modelá-los no nível necessário de detalhes.
c) Programação: A capacidade de escrever um programa de computador legível e verificável que implementa o modelo corretamente.
d) O conhecimento do sistema modelado: A capacidade de compreender o sistema, explicar isso para a equipe de modelagem, e interpretar os resultados da modelagem em termos de seu impacto sobre o projeto do sistema.

Jain (1991) ainda afirma que, os membros da equipe de simulação devem possuir pelo menos uma dessas habilidades e ter, de preferência, um líder que possua algum conhecimento de todas as habilidades.

Tendo em vista esses fatos, fica claro que para se obter bons resultados de uma simulação se faz necessária a formulação de um modelo de simulação detalhado, para que a mesma possa descrever as operações do sistema e como ele devera proceder os experimentos.

2.2.1 Modelos

Segundo Hiller & Lieberman (2006, pág. 478) um modelo de simulação: "sintetiza o sistema construindo-o componente por componente, e evento por evento."

Sendo assim, os modelos tem por objetivo descrever de maneira simples e completa o sistema estudado imitando os seus funcionamentos. Como os sistemas que utilizam simulações são muito complexos se faz necessário certas abstrações, para que possa ser analisado apenas os seus aspectos mais relevantes para a simulação. A Imagem a seguir mostra a ideia de um modelo de um sistema real:

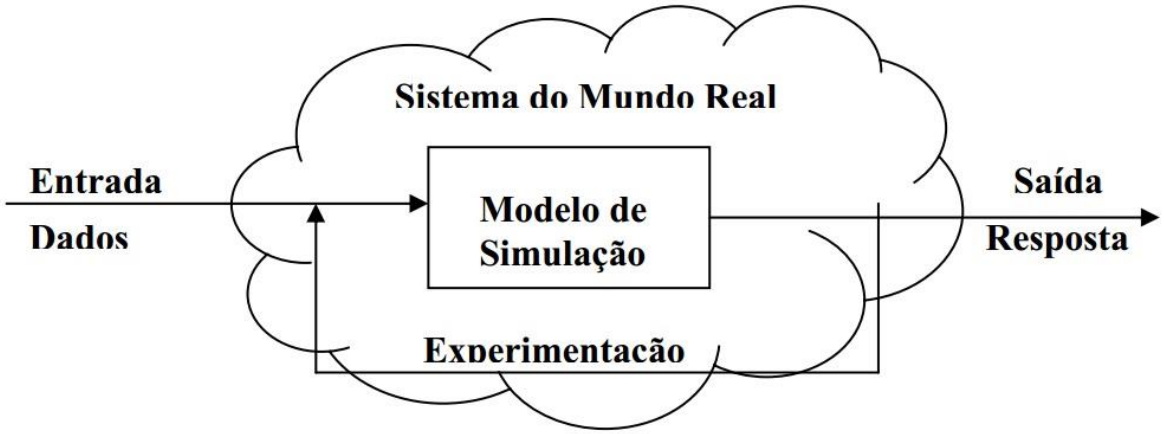

Figura 1: Representação de um modelo de sistema.
Fonte: Introdução a modelagem e Simulação de Sistemas. Freitas Filho(2008).

Onde, a nuvem representa o sistema real em sua totalidade e o modelo representa a parte essencial do sistema. Para que se obtenha um bom modelo o mesmo não pode possuir todos os detalhes do sistema real, pois poderia gerar complicações desnecessárias no momento da simulação como maior gasto de tempo para modelagem do sistema, maior probabilidade de surgirem novos bugs e a necessidade de computadores com maior poder de

processamento, para analisar a grande variedade de detalhes do sistema. Também não pode ser retirada características importantes do sistema, pois faria com que o modelo não se comporta-se de maneira semelhante ao sistema real, invalidando assim o modelo para a realização de simulações. Logo, sabe-se que a modelagem de um sistema real e sua respectiva validação não é tarefa fácil.

Hillier & Lieberman (2006, pág. 478) mostram os seguintes blocos construtivos de um modelo de simulação:

- Uma definição do estado do sistema (por exemplo, o número de clientes em um sistema de filas);
- Identificar os possíveis estados do sistema que podem ocorrer;
- Identificar os possíveis eventos (por exemplo, chegadas e términos de atendimento em um sistema de filas) que mudariam o estado do sistema;
- Uma provisão para um relógio simulado, localizado no mesmo endereço do programa de simulação, que vai registrar a passagem do tempo (simulado);
- Um método para gerar eventos aleatoriamente de diversos tipos;
- Uma fórmula para identificar as transições de estado que são geradas pelos diversos tipos de eventos.

Estes itens servem como guia para a modelagem de sistemas onde cada um deles devem ser levados em consideração, para que se obtenha um modelo que atenda as exigências para realizar a simulação.

Segundo Jain (1991, pág. 414) "À uma série de termos que são comumente usados na modelagem". Sendo necessário, o conhecimento desses termos para melhor construção e manuseio dos modelos. Abaixo segue uma lista abstraída de Jain (1991):

Variáveis de Estado: são as variáveis que indicam o estado do sistema em um determinado instante de tempo. Caso a simulação seja interrompida no meio de um experimento ela poderá ser reiniciada, se somente se, forem conhecidas todas as variáveis de estado do momento em que ela foi interrompida.

Entidades: São os elementos do sistema que podem ser identificados individualmente (pessoas ou objetos pertencente ao sistema) possuindo atributos. As entidades podem ser classificadas como dinâmicas ou estáticas. São classificadas como dinâmicas quando, estas realizam suas atividades em um determinado espaço de tempo e se retiram do sistema

podendo retorna ou não. Já as entidades denominadas como estáticas realizam suas atividades e permanecem dentro do sistema.

Recursos: São as entidades estáticas que fornecem serviços as entidades dinâmicas.

Atributo: São as características de cada entidade do sistema. Os atributos podem ser alterados durante a simulação.

Evento: É a mudança do estado do sistema em um dado momento. Normalmente a ocorrência de um evento dispara a ocorrência de outro(s) evento(s).

Processos: Ações que são realizadas sobre as entidades durante a simulação. Podem modificar os atributos das entidades e gerar eventos.

Tempo simulado e Tempo de simulação: Tempo simulado é a forma de medir o tempo de duração de uma simulação, quanto tempo se passou desde o início da simulação até o seu termino. Tempo de simulação é o tempo de execução de uma "rodada" de simulação, também conhecido como tempo logico, sendo este determinado pela pessoa responsável pela modelagem.

2.2.2 Tipos de simulação

Existe uma grande variedade de simulações que são descritos na literatura, podemos ressaltar quatro dessas que são de grande importância para a simulação computacional sendo elas Simulação de Monte Carlos, Simulação *Trace-Driven*, Simulação de Eventos Contínuos e Simulação de Eventos Discretos, onde irmos discutir melhor a seguir:

Simulação de Monte Carlos é segundo Jain (1991, pág. 420) "uma simulação estática ou sem eixo de tempo". Sendo assim, a Simulação de Monte Carlos fornece vários resultados possíveis sem alterar as características de acordo com tempo decorrido e também fornece a probabilidade de ocorrência desses resultados. Jain também diz que:

> "Estas simulações são usadas para modelar fenômeno probabilístico que não alteraram as características com o tempo. Assim como a uma simulação dinâmica elas requerem a geração de números pseudoaleatórios" (Jain, 1991, pág. 420).

A Simulação de Monte Carlos foi utilizada inicialmente pelos cientistas que trabalhavam na bomba atômica e recebeu essa denominação como referência a cidade de

Mônaco e seus casinos. Além disso, a Simulação de Monte Carlos é muito utilizada onde se faz necessária a tomada de decisões com várias possibilidades para as variáveis de entrada, mas sem modificação durante a simulação, como exemplo podemos citar simulações da bolsa de valores ou de jogos de cassinos.

Simulação *Trace-Driven* segundo Jain (1991) é uma simulação onde se utiliza um traço como entrada, onde o traço é um registro temporal de eventos em um sistema real. Podemos dar como exemplo a simulação de um servidor web, onde uma sequência de requisições http devem ser adicionadas ao traço. Se os dados do traço representarem fielmente os dados do sistema real obteremos bons resultados da simulação, sendo desnecessário escrever o código que modela a carga. Porém é difícil montar um traço, tendo em vista que podem haver vários dados a serem inseridos no mesmo.

Simulação de Eventos Contínuos segundo Hillier & Lieberman (2006, pág.479) "é aquela na qual as mudanças no estado do sistema ocorrem continuamente ao longo do tempo". Sendo assim, o estado do sistema assume valores contínuos. Jain (1991) diz que as Simulação de Eventos Contínuos são empregados em simulações químicas, onde o estado do sistema é dito pela concentração da substancia química.

Porém Hillier & Liberman (2006) mostram como exemplo, um sistema onde existe um avião em voo e seu estado é definido pela posição atual da aeronave. Ele explica que nesse caso o estado está mudando continuamente ao longo do tempo de forma continua.

Simulação de Eventos Discretos segundo Hillier & Lieberman (2006, pág.479) "é aquela em que as mudanças no estado do sistema ocorrem instantaneamente em pontos aleatórios no tempo como resultado da ocorrência de eventos discretos.". Para melhor entendimento podemos tomar como exemplo, um sistema de fila no qual o estado do sistema é o número de clientes dentro do sistema, onde os eventos discretos que mudam o estado do sistema são as chegadas e saídas dos clientes no sistema, sendo estas em decorrência da necessidade da realização e finalização do serviço desejado pelo cliente.

Vale salientar que o termo discreto não se aplica em relação aos valores de tempo utilizados na simulação, sendo que uma Simulação de eventos discretos pode utilizar valores de tempo continuo ou discreto.

A simulação vem sendo muito utilizada na área de sistemas de comunicação, por permiti a análise de sistemas complexos e caros, fornecendo assim uma maneira de analisar e

aperfeiçoar os protocolos de comunicação nos sistemas. A simulação desses sistemas se caracteriza de maneira geral como simulações de Eventos Contínuos e/ou de Eventos Discretos, segundo Issariyakul & Hossain (2009, pag. 7) em geral a simulação de redes de computadores consiste em três principais partes sendo elas:

- Planejamento: faz parte desta etapa a definição do problema, a criação do modelo, do sistema em questão e a criação de um conjunto de experimentos para o modelo de simulação.
- Implementação: implementação de programas de simulação. Está parte se define em três etapas, são elas:
    - 1ª etapa - Inicialização: é onde são definida as condições iniciais (por exemplo, redefinição do relógio de simulação e variáveis), isto é feito para que a simulação sempre comece a parti de uma estado conhecido.
    - 2ª etapa – Geração de resultados: nesta etapa a simulação cria e executa os eventos, recolhendo os dados necessários gerados pelos mesmos.
    - 3ª etapa – Pós-processamento da simulação: ocorre nesta etapa, o processamento dos dados brutos, coletados de uma simulação, sendo estes traduzidos em medidas de interesse.
- Testando: esta parte verifica e valida os modelos de simulação, experimentando os cenários definidos na 1ª parte, melhorando-os e analisando seus resultados.

Na fase de planejamento, mais especificamente na construção do modelo da simulação, deve-se ter um cuidado maior com as características que iram compor o modelo, pois elas iram influenciar diretamente nos resultados da simulação. Podemos citar como principais características de um sistema de comunicação a topologia do sistema (sendo ela física ou lógica), seus componentes (roteadores, servidores e etc), o protocolo de comunicação utilizado e os meios de comunicação.

Segundo Carissimi, Rochol & Granville (2009, pag.44) “A maneira de como são distribuídos espacialmente os nós de uma rede e, principalmente, a maneira como estão interligados esses nós, é o que definimos como topologia física dessa rede.”. Em geral, a topologia de uma rede é escolhida com o intuito de anteder a uma exigência especifica da rede, podemos citar algumas dessas como: alta vazão, baixo atraso, alta confiabilidade e economia de enlaces. As topologias definem como será realizada a comunicação entre os nós,

sendo uma das características mais importantes a ser levada em consideração na hora de implementar ou modelar uma rede.

2.4 Internet Topology Zoo

O Internet Topology Zoo é um projeto, ainda em andamento da Universidade de Adelaide, desenvolvido com o intuito de mapear as redes existentes, juntamente com suas topologias, do mundo todo. Sendo o projeto, apoiado pelo governo australiano através de uma premiação de pós-graduação da Austrália e de subsídios do conselho de pesquisa e descoberta da Austrália.

Hoje o projeto possui mais de duzentos e cinquenta redes em seu banco de dados, onde essas redes são armazenadas em dois formatos o GML e o GraphML, que são disponibilizados para download no site[2] do projeto, podendo estas serem visualizadas através do mesmo. As redes apresentadas para visualizações são geradas a parti da criação do gráfico, que representa a rede, em uma imagem da região onde se encontra a mesma, para isso utiliza-se o arquivo GML da rede como fonte de dados para a sua criação.

A documentação do projeto Internet Topology Zoo (2014) mostra que as redes são mapeadas a parti de imagens disponíveis na web e que contenham a rede a ser mapeada, onde é utilizado um programa chamado yEd, sendo este um editor de gráficos usado para traçar a rede. O programa yEd adiciona os nós e arestas de acordo com o formato e posição dos mesmos na imagem, gerando assim a topologia da rede e o arquivo GML que irá conter os dados da rede. A documentação também diz que:

> "Se for o caso, podemos indicar diferentes tipos de nós (como provedores de trafico externo), utilizando diferentes contornos. Finalmente, podemos classificar os nós (como núcleo ou borda) utilizando cores. A rede traçada é então guardada no formato GML de troca de gráfico, adequado para o processamento." (Documentation, 2014)

Após traçada a rede e salvo o arquivo GML, o projeto pode realizar vários processos no mesmo através de scripts, desenvolvidos pelo próprio projeto na linguagem Python. Entre os scripts utilizados pelo projeto podemos citar o *convert* que realiza a conversão do arquivo

[2] http://www.topology-zoo.org/dataset.html

GML em outros formatos de arquivos, como o GraphML e matrizes de adjacência adequados para uso no Matlab.

2.4.1 GML (Graph Modeling language)

Existem diversos tipos de formatos de arquivos para realiza a representação de gráficos, sendo que cada um possui características especificas para sua atuação, contudo isso os tornam incompatíveis. Assim, a troca de gráficos entre programas que trabalham com formatos distintos acaba se tornando muito complicada e as vezes impossível.

Como solução a Universidade de Passau, situada na Inglaterra, desenvolveu o formato GML, que segundo Himsolt (1996) foi idealizado na GD'95(*Graph Drawing: Symposium on Graph Drawing***)** com o intuito de permitir a portabilidade de gráficos entre diversos formatos existentes como o RTF (*Rich Text Format*), HTML (*HyperText Markup Language*), SGML (*Standard Generalized Markup Language*), Postscript entre outras.

Himsolt (1996) mostra como características para um formato de intercambio, como o GML, a independência de plataforma, a fácil implementação, possuir a capacidade de uma estrutura com dados arbitrários, para programas que necessitem adicionar dados específicos, e ser flexível, para que uma ordem especifica de declarações não seja necessária e que quaisquer dados que não sejam essenciais possam ser omitidos.

O formato GML trabalha com um sistema de lista de valores-chaves que seguem uma hierarquia definida pelo formato. Abaixo segue um exemplo de um gráfico com três nós e três arestas, estando estes interligados, e sua descrição no formato GML:

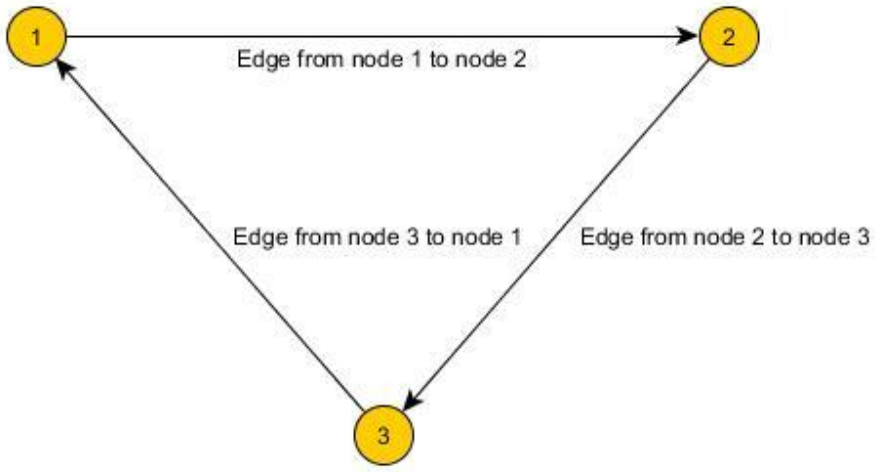

Figura 2: Gráfico simples gerado pelo yEd, aparte do arquivo firtsExemle.gml.

```
graph [
  comment "This is a sample graph"
  directed 1
  IsPlanar 1
  node [
    id 1
    label "1"
  ]
  node [
    id 2
    label "2"
  ]
  node [
    id 3
    label "3"
  ]
  edge [
    source 1
    target 2
    label "Edge from node 1 to node 2"
  ]
  edge [
    source 2
    target 3
    label "Edge from node 2 to node 3"
  ]
  edge [
    source 3
    target 1
      label "Edge from node 3 to node 1"
  ]
]
```

Arquivo firtsExemle.gml.

Fonte: Retirado do documento GML: A portable Graph File Format.

Porém o padrão GML não possui todos os atributos para certas aplicações, sendo assim ele fornece duas alternativas como solução para esse problema. A primeira é ignorar

estes atributos omitindo eles do arquivo, se for escolhida esta opção ocorrera perda de dados relevantes do gráfico, o que torna esta solução não muito atrativa. A segunda é mais complexa pois sugere a criação de uma estrutura ou atributo genérico, com isso se garante a integridade dos dados. Apesar de mais atrativa a segunda opção, por manter todos os dados do gráfico, a mesma pode ocasionar problemas de inconsistência, pois ao adicionar estruturas ou atributos genéricos no arquivo algumas aplicações podem não reconhecê-las gerando erro ao tentar utilizar o arquivo. Abaixo segue um exemplo de arquivo GML com alguns atributos genéricos, como Longitude e Latitude, além de uma imagem com sua representação:

```
graph [
  DateObtained "25/01/11"
  GeoLocation "United States"
  GeoExtent "Country"
  Network "ARPANET"
  Provenance "Secondary"
  Access 0
  Source "http://som.csudh.edu/cis/lpress/history/arpamaps/"
  Version "1.0"
  Type "REN"
  DateType "Historic"
  Backbone 1
  Commercial 0
  label "Arpanet196912"
  ToolsetVersion "0.3.34dev-20120328"
  Customer 0
  IX 0
  SourceGitVersion "e278b1b"
  DateModifier "="
  DateMonth "12"
  LastAccess "25/01/11"
  Layer "ARPA"
  Creator "Topology Zoo Toolset"
  Developed 1
  Transit 0
  NetworkDate "1969_12"
  DateYear "1969"
  LastProcessed "2011_09_01"
  Testbed 0
  node [
    id 0
    label "SRI"
    Country "United States"
    Longitude -122.18219
    Internal 1
    Latitude 37.45383
  ]
  node [
    id 1
    label "USCB"
    Country "United States"
    Longitude -119.69819
    Internal 1
    Latitude 34.42083
  ]
  node [
```

```
    id 2
    label "UCLA"
    Country "United States"
    Longitude -118.24368
    Internal 1
    Latitude 34.05223
  ]
  node [
    id 3
    label "UTAH"
    Country "United States"
    Longitude -111.89105
    Internal 1
    Latitude 40.76078
  ]
  edge [
    source 0
    target 1
    id "e0"
  ]
  edge [
    source 0
    target 2
    id "e1"
  ]
  edge [
    source 0
    target 3
    id "e2"
  ]
  edge [
    source 1
    target 2
    id "e3"
  ]
]
```

Arquivo Arpnet196912.gml

Fonte: Retirado do site do projeto Internet Topology Zoo.

Figura 3: Gráfico da rede Arpnet 1996
Fonte: http://www.topology-zoo.org/maps/Arpanet196912.jpg

De forma geral o formato GML é utilizado como intermediador entre vários formatos e entre vários programas, facilitando assim a difusão de vários gráficos gerados em ambientes distintos, mantendo a estrutura principal dos gráficos durante a mudança.

2.5 NetLogo

Segundo Wilensky (2014), o NetLogo é uma ferramenta para a modelagem de ambientes com multiagentes, sendo utilizado para realizar simulações de diversos sistemas. O autor dessa ferramenta, Uri Wilensky, a desenvolveu em 1999 e desde então vem sendo aprimorada continuamente pela CCL (*Center for Connected Learning*).

No manual do NetLogo Wilensky (2014) diz que, a ferramenta é adequada para modelar e desenvolver sistemas complexos ao longo do tempo. Isso se deve ao fato do NetLogo permitir aos modeladores passar instruções para centenas ou milhares de “agentes” que atuam de forma independente, tornando possível visualizar como os indivíduos se

comportam em pequenos grupos e como padrões de grupos maiores surgem aparte desses pequenos grupos.

O NetLogo é utilizado por centenas de estudantes, professores e pesquisadores pelo mundo. Ele oferece segundo Wilensky (2014), uma extensa documentação e tutoriais além de possuir uma biblioteca de modelos com uma grande coleção de simulações pré-gravadas, que podem ser utilizadas ou modificadas. Estas simulações são de diversas áreas como ciências naturais e sociais, biologia, medicina, física, química, matemática, ciência da computação, economia, psicologia social entre outras áreas.

A ferramenta NetLogo é a nova geração de uma série de ferramentas de modelagem de ambientes com multiagentes, como StarLogo e StarLogoT, além de rodar em cima da Java Virtual Machine (JVM), podendo assim ser utilizada na maioria dos sistemas operacionais, como uma aplicação independente de plataforma.

Segundo Wilensky (2014) algumas das características da ferramenta são sistema gratuito e de código aberto, multe-plataforma, totalmente programável e de sintaxe acessível aos usuários, baseado na linguagem de programação Logo, entre outras características que fazem do NetLogo uma ferramenta poderosa de modelagem e simulação de sistemas, para uma lista completa das características acesse site[3].

Sobre extensões, o NetLogo permite e incentiva os seus usuários a criarem novas extensões, aumentando assim as suas capacidades de uso. As extensões podem ser escritas nas linguagens que utilizam a JVM, como as linguagens Java e Scala. O NetLogo traz com sigo as extensões desenvolvidas pelo CCL e disponibiliza várias outras através do GitHub.

O NetLogo disponibiliza alguns tutoriais explicando como se criar uma extensão, para as linguagens Java e Scala. Ressalta-se ainda que a maioria das extensões são desenvolvidas na linguagem Scala, devido a sua ágil programação e compatibilidade com a ferramenta.

2.5.1 Linguagem de programação Scala

Segundo a documentação da linguagem Scala:

[3] http://ccl.northwestern.edu/netlogo/docs/whatis.html

> "Scala é uma linguagem de programação moderna e multe-paradigma, projetada para expressar padrões de programação comuns de uma forma concisa, elegante e type-safe. Ele integra os recursos de linguagens orientadas a objetos e funcional." (2014, disponível no site http://docs.scala-lang.org/tutorials/tour/tour-of-scala.html)

A linguagem foi criada na Escola Politécnica Federal de Lausanne em 2001 por Martin Odersky e foi liberada para o uso em conjunto com a JVM em janeiro de 2004. Segundo Odersky (2014) a mesma é uma linguagem escalável, ou seja ela cresce com o usuário, podendo ser utilizada para escrever pequenos códigos como para desenvolver grandes sistemas. Podemos citar ainda o Twitter e o LinkedIn como grandes empresas que utilizam está linguagem.

A escalabilidade da linguagem se deve ao resultado de uma cuidadosa interação entre os conceitos de orientação a objeto e linguagem funcional. Essa interação resultou em uma linguagem bastante agradável e de fácil programação tanto que em 2012 a linguagem Scala ganhou o concurso de ScriptBowl[4] na conferência JavaOne.

A sintaxe da linguagem é bem concisa e com poucas formalidades além da mesma ser puramente orientada a objetos, no sentido de que tudo é um objeto, incluindo números e funções. Neste caso cada valor é um objeto e cada operação é uma chamada de método.

A linguagem Scala, diferente de outras linguagens que suportam heranças múltiplas, tem uma noção mais geral de reutilização de classe. Para obter está noção que substitui a herança múltipla a linguagem Scala possui um mecanismo denominado *traits*.

Schinz e Haller explica o que são traits da seguinte forma:

> "Talvez a forma mais fácil para um programador Java entender o que são traitss é vê-los como interfaces na qual podem também conter código. Em Scala, quando uma classe é sub-classe de um traits, ela implementa aquela interface, trats, e herda todo o código contido nele." (Schinz e Haller, 2014, Pag. 12)

Scala também é uma linguagem funcional, no sentido de que toda função é um valor e como dito acima todo valor é um objeto. Segundo a documentação da linguagem (2014) "Scala fornece uma sintaxe simples para definir funções anônimas, que suportam funções de ordem maior, permitindo assim que as funções se aninhem, além de suporta Currying."

As funções de ordem maior, são funções, que recebem outras funções como parâmetros ou cujo o retorno seja uma função. Já Currying é uma técnica que permite a

[4] Concurso, onde as linguagens que rodam na JVM, representadas por seus gurus, batalham para exigir o direito de ser denominada como a linguagem mais popular.

função receber múltiplos parâmetros, de forma que possa ser chamada como uma cadeia de funções que recebem somente um parâmetro cada.

A documentação da linguagem Scala (2014) diz também que a mesma, possui um expressivo sistema de tipos, que estaticamente impõe quais abstrações devem ser usadas de maneira segura e coerente.

3. METODOLOGIA

O Presente trabalho tem como objetivo o desenvolvimento de uma extensão para a ferramenta NetLogo, sendo assim o trabalho se caracteriza da seguinte maneira:

3.1 Natureza da pesquisa

O trabalho possui como natureza de pesquisa uma pesquisa aplicada, pois busca gerar conhecimento para a solução de problemas específicos, em questão a importação de topologias para a ferramenta NetLogo.

3.1.1 Quanto aos fins

O trabalho possui caráter, quanto aos fins, de pesquisa Exploratória e aplicada sendo demonstradas a seguir:

Segundo ANDRADE (2009, pág. 114) “São finalidades de uma pesquisa exploratória, sobretudo quando a bibliográfica, proporcionar maiores informações sobre determinado assunto”. O trabalho se enquadra nessa pesquisa, pois explora um novo meio de integrar formatos gráficos de arquivos à ferramenta NetLogo, além de examinar uma área com poucos estudos.

Segundo VERGARA (1998, pág. 45) “A pesquisa aplicada é fundamentalmente motivada pela necessidade de resolver problemas concretos; mais imediatos, ou não”. Sendo assim o trabalho se enquadra perfeitamente nesse âmbito, pois tem como prioridade resolver o problema de importação de topologias de redes para a ferramenta NetLogo.

3.1.2 Quanto aos meios

Quanto aos meios, o trabalho apresenta se como Pesquisa bibliográfica.

> “Pesquisa bibliográfica é o estudo sistematizado desenvolvido com base em material publicado em livros, revistas, jornais, redes eletrônicas, isto é, material acessível ao público em geral.”. VERGARA (1998, pg. 46)

Por tanto o trabalho se caracteriza como pesquisa bibliográfica, pois será necessário realizar estudos sobre como desenvolver novas extensões e de como o arquivo GML é definido além de fontes para o embasamento teórico do trabalho.

4. DESCRIÇÃO DA EXTENSÃO

A extensão ImportGML foi desenvolvida na linguagem Scala para maior compatibilidade com a ferramenta NetLogo e para maior velocidade no desenvolvimento da mesma.

Como mostra o modelo UML da extensão, logo abaixo, a mesma é formada por oito classes, onde sete dessas foram desenvolvidas neste trabalho e uma foi retirada da extensão NW-extension, sendo demostrada a função de cada uma delas mais a diante.

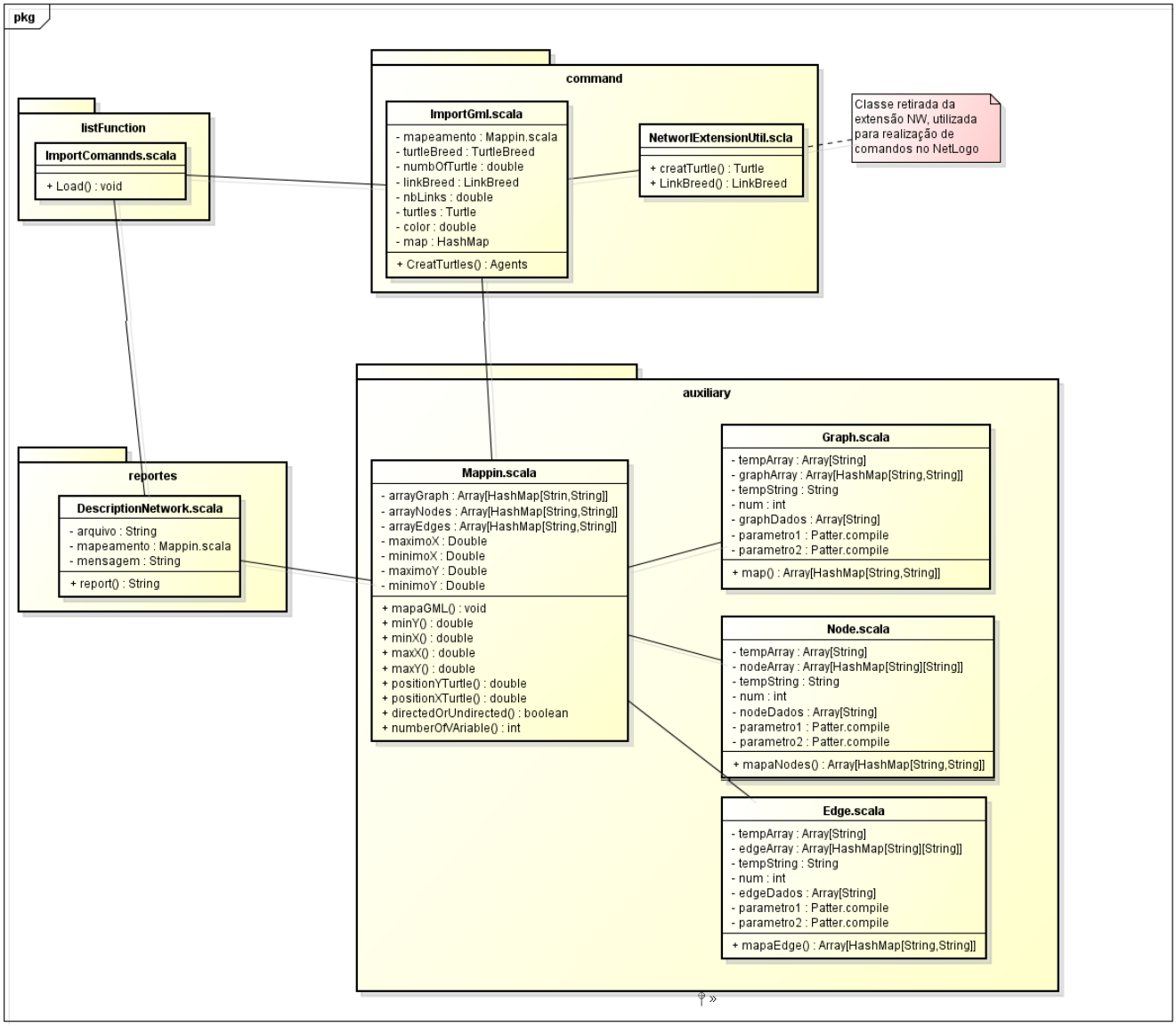

Figura 4: Modelo UML da Extensão ImportGML.

Como descrito nos tutoriais para desenvolvimento de extensões para à ferramenta NetLogo, as extensões necessitam ter uma classe que contenha os comandos que serão utilizados pelos usuários. Além de conter os comandos esta classe tem a tarefa de designar qual classe será chamada para executar o comando solicitado. Esta classe deve estender a classe DefaultClasseManager, para designar que a mesma será responsável pelo controle dos comandos da extensão.

A classe ImportCommands da extensão ImportGML é a responsável por esta tarefa, ela possui dois comandos o **description-file** e o **import-file.** Logo Abaixo segue o *User Case* da extensão para melhor compreensão dos seus comandos.

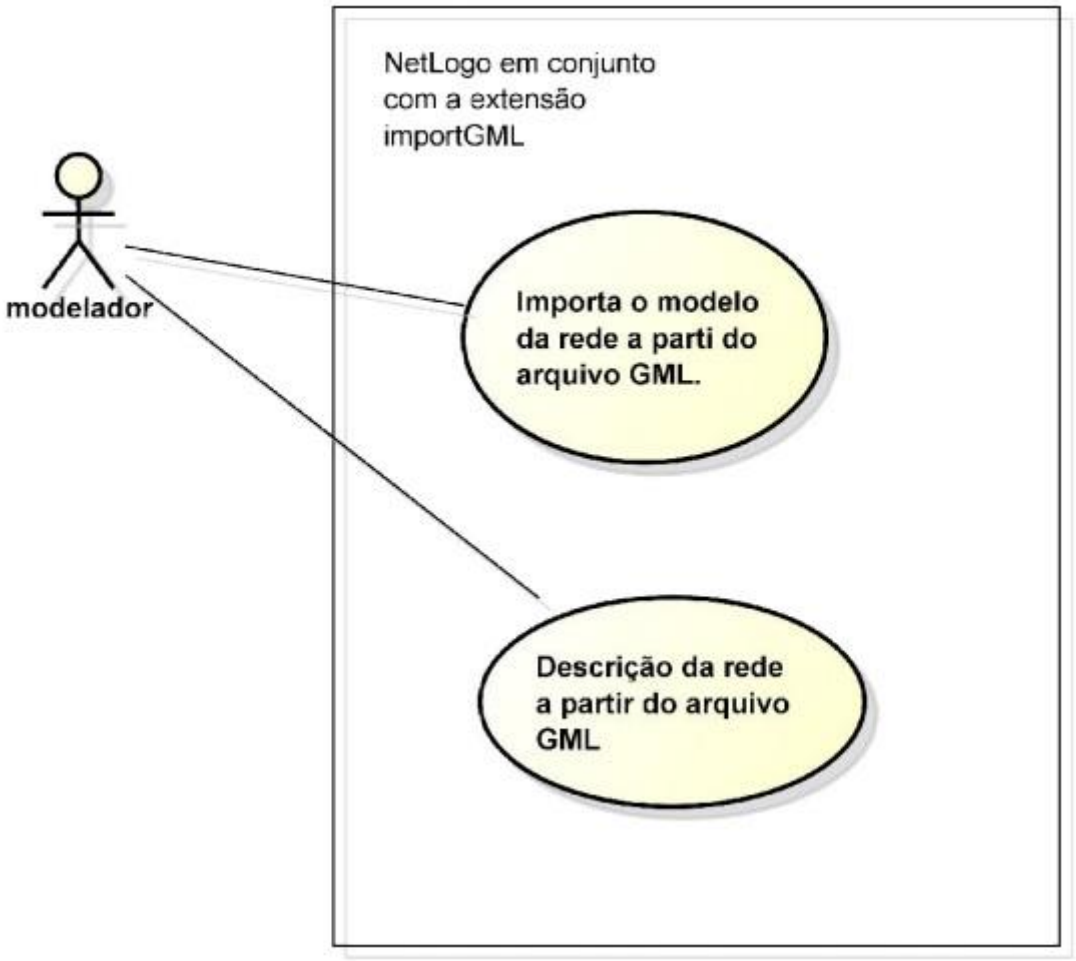

Figura 5: User Case da extensão ImportGML.

O comando description-file possui a função de exibir as informações contidas no valor-chave Graph do arquivo GML, estas são informações que descrevem a rede. Podemos citar como exemplo dessas informações o criador da rede, o tipo do link, a versão do arquivo

entre outras informações que tenham sido registradas nesse valor-chave. Para apresentar estes dados o comando description-file chama a classe DescriptionNetwork, sendo esta uma classe do tipo reporte. As classes reportes tem como função reportar alguma ação ou informação para a ferramenta NetLogo.

A classe DescriptionNetwork recebe como parâmetro uma String contendo o caminho do arquivo GML. Ela repassa este caminho para a Classe Mapping que mapeia o arquivo GML e guarda as informações necessárias para usos posteriores. Após receber os dados necessários da classe Mapping, a classe DescriptionNetwork monta e retorna uma String contendo os dados da rede, para serem exibidas na ferramenta NetLogo.

A classe Mapping, por sua vez, realiza as seguintes tarefas: verificação do caminho e do tipo do arquivo passados por parâmetro, retirada das tags, caso existam no arquivo, dividir as informações contidas no arquivo, na ordem hierárquica do padrão GML e armazena as informações da rede necessárias para realização dos comandos da extensão ImportGML. A classe Mapping utiliza três classes para obter e mapear as informações do arquivo GML, são as classes Graph, Node e Edge.

As classes Graph, Node e Edge assemelham-se pois possuem a tarefa de receber os dados pertencentes da sua respectiva chave-valor, de analisar e tratar estes dados, separando os atributos dos seus valores e armazenando-os em uma coleção de mapas onde à chave é o atributo e o valor do mapa, é o valor do atributo. Logo após o tratamento dos dados as classes retornam a coleção gerada para a classe Mapping.

O comando import-file realiza a importação da rede contida no arquivo GML, para isso o comando chama a classe ImportGML que tem como função principal montar a rede a parti de informações como número de nós, número de links, tipo do link, posição dos nós, nós base e destino dos links entre outras informações. Estas informações são adquiridas na classe Mapping como descrito acima, e são utilizadas para a criação dos agentes que iram representar a rede na ferramenta NetLogo.

Os agentes são criados na classe NetworkExtensionUtil, esta classe foi retirada da extensão NW-network pois possuía todas as funcionalidades que a extensão ImporGML necessitava para a criação dos agentes além de realizar validações e comparações facilitando assim o manuseio dos mesmos.

Por fim vale ressaltar que a classe ImportGML e NetworkExtensionUtil são do tipo command, este tipo de classe tem a função de gerar, destruir e controlar os agentes na ferramenta NetLogo.

5. ANALISE DA EXTENSÃO

Para avaliar a capacidade da extensão importGML de conseguir importa redes já mapeadas no formato GML foram realizados testes para verificar se a rede gerada pela extensão estava de acordo com as descrições da rede existentes no arquivo que a continha. Foram escolhidas três redes a Arpnet de 1996, Belnet de 2006 e a Cogent de 2010, pois estas possuem as principais características distintas umas das outras, abrangendo assim diversos cenários para realização dos testes.

Ao final de todos os testes foram obtidas as redes específicas de cada importação, onde cada uma possuía características especificas de sua rede, estando estás de acordo com as descritas no arquivo GML. Além disso os testes demostram que é possível especificar determinados tipos de nós na rede, como nós de bordas.

No geral a extensão importGML demostrou durante os teste ser uma maneira eficaz de importação de redes a partir de arquivos no formato GML gerados pelo projeto Internet Topology Zoo.

Abaixo iremos realizar algumas comparações de como é a modelagem das redes na ferramenta NetLogo, utilizando outras extensões, utilizando a ferramenta em si e a extensão ImportGML.

Esta comparação tem o intuído de demonstrar como a extensão se torna útil para os usuários que necessitem modelar redes já existentes na ferramenta NetLogo, poupando-lhes tempo e garantindo uma rede com maior semelhança a rede existente.

A rede selecionada para ser modelada é o backbone da rede Ipê, que segundo o site da Rede Nacional de Ensino e Pesquisa (RNP, 2014) a mesma é “uma infraestrutura de rede Internet voltada para a comunidade brasileira de ensino e pesquisa.”.

A rede possui 28 Pontos de Presença (PoPs). Estes pontos estão presentes em todas as 27 unidades da federação, sendo que à unidade da federação da Paraíba possui 2 PoPs, onde os mesmos estão dispostos em várias topologias. Além disso, a rede possui acesso a três redes acadêmicas sendo elas a Clara situada na américa latina, Internet2 situada nos Estados Unidos da America e a Géant situada na Europa.

Para iniciar a modelagem utilizando somente a ferramenta NetLogo se faz necessário o conhecimento das características da rede sendo as principais o número de nós existentes, o número de links, tipo dos links, se são direcionados ou não e as ligações que existem dentro da rede. A rede Ipê possui 31 nós levando em consideração os nós de acesso as redes acadêmicas e possui 34 links, sendo eles não-direcionados.

Após o recolhimento dos dados necessários os usuários deveram criar os agentes. Em primeiro lugar devem ser criados os agentes nós, no caso da rede Ipê devem ser criados 31 agentes nós, após a criação dos agentes nós os agentes links deverão ser criados um a um para indicar quais são os nós que eles interligam, gerando assim a topologia da rede. Além disso o usuário deverá estabelecer padrões para os agentes como tamanho dos agentes seu formato entre outros detalhes.

Analisando este método para modelar à rede Ipê vemos que o mesmo exige muita atenção e tempo por parte do usuário, pois este deve ficar atento a vários detalhes da rede sem mencionar que se deve criar 34 links um de cada vez para ser gerada a rede.

Existem três maneiras distintas para realizar a modelagem da rede Ipê utilizando a extensão NW-network, sendo elas demostradas a seguir.

A primeira seria utilizando os comandos para criação de redes com determinadas topologias fornecidas pela extensão. Estes comandos criam os agentes nós e links com as devidas ligações a parti do número de nós e do tipo da topologia especificada pelo o usuário.

Porem a rede Ipê possui mais de uma topologia, tornando-se necessário o levantamento das topologias e como elas se ligam entre si na rede. Após a obtenção dos dados e das característica da rede o usuário irá utilizar os comandos para criação de topologias, gerando cada uma das topologias presentes na rede, em seguida o usuário deverá realizar os devidos ajustes de ligações entre as topologias e criar novos nós e ligações caso necessários para compor toda a rede.

Este método se mostra mais rápido que o primeiro pois não é mais necessário a criação individual de cada link, sendo apenas necessário a realização alguns ajustes para a criação da rede. Porem este método ainda necessita de bastante atenção por parte do usuário para que nenhum aspecto da rede seja esquecido durante a sua criação.

A segunda maneira de realizar a modelagem da rede utilizando a extensão NW-network, seria através da criação e importação de um arquivo no formato texto contendo uma matriz de conexão. Abaixo segue a matriz que o usuário teria de montar para realizar a importação da rede Ipê.

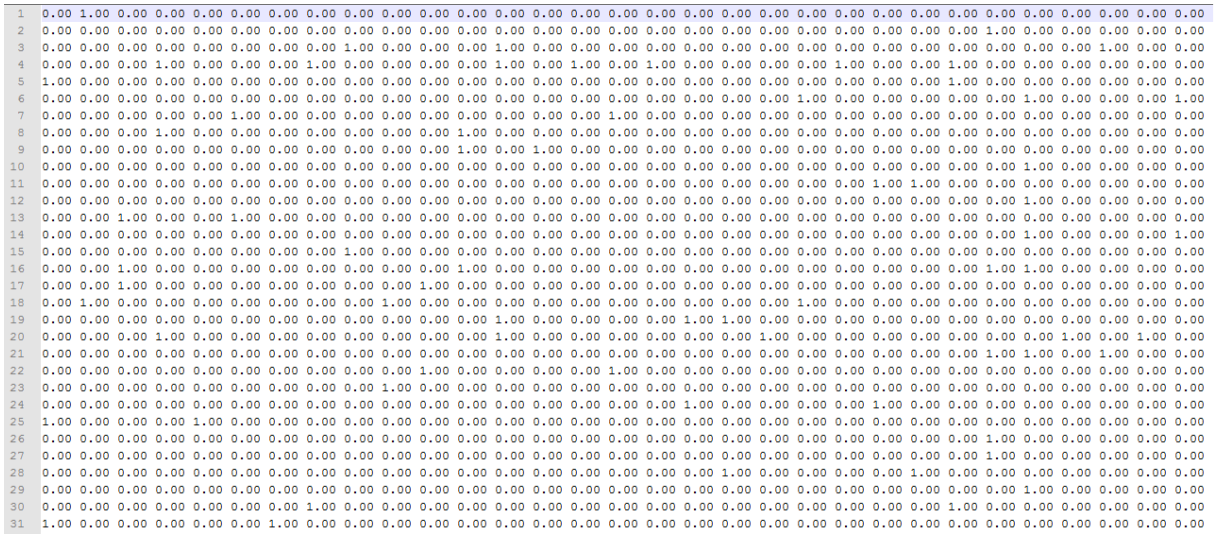

Figura 6: Matriz de conexão da rede Ipê.

Como podemos ver a construção desta tabela não é uma tarefa trivial e caso queira mudar o tipo dos links a tabela também irá mudar, contudo após feita a tabela basta somente utilizar o comando de importação do arquivo, tornando este método lento mas com boa reutilização do arquivo para novas simulações.

A terceira o opção também abrange a construção e importação de um arquivo só que no formato graphML. Esta opção é a mais próxima da utilizada pela extensão desenvolvida neste trabalho, levando em consideração que ambos os formatos utilizados pelas extensões possuem repositórios com redes mapeadas na internet disponíveis para download, onde ambos só necessitam baixar o arquivo e utilizar os respectivos comandos para gerar a rede desejada na ferramenta NetLogo. Como citado o projeto Internet Topology Zoo possui várias redes mapeadas nos formatos GML e graphML sendo uma dessas redes a Ipê, bastando para o nosso exemplo apenas realizar o download do arquivo e utilizar os devidos comandos para importação.

Apesar das semelhanças podemos citar algumas vantagens em se utilizar a extensão ImportGML, sendo a primeira o formato do arquivo GML utilizado para a importação, como mencionado o formato foi idealizado para ser um intercâmbio entre diversos outros formatos, facilitando assim a obtenção e criação de novas redes.

Além disso o formato permiti a criação e manipulação de novos atributos permitindo também a fácil manipulação e visualizam da rede no arquivo, atendendo assim a várias necessidades dos usuários no momento da modelagem.

Outra vantagem ao se utilizar a extensão é que a mesma permite a importação de atributos distintos como peso dos nós ou velocidade dos links, bastando apenas que o usuário crie um agente que possua as mesmas características existentes no arquivo a ser importado.

Um outro fato que favorece a extensão é que ela permite a visualização dos dados da rede na própria fermenta através do comando descrption-file, como descrito no tópico anterior.

6. CONCLUSÃO E TRABALHOS FUTUROS

Como apresentado a extensão ImportGML foi desenvolvida na linguagem Scala com a função de permitir a importação de arquivos no formato GML para a ferramenta NetLogo. Também foi apresentado um tópico explicando as funções e as classes que compões a extensão.

Ainda foi revelado que a extensão mostrou-se um meio eficaz de importa redes mapeadas em arquivos no formato GML para a ferramenta NetLogo, pois proporciona uma importação mais completa das redes mapeadas do que outras extensões existentes. Além de utilizar o formato GML que permite a conversão do mesmo para outros formatos de grafos e vice-versa.

Portanto o trabalho apresenta como contribuição a extensão importGML e está monografia como fonte de pesquisas para trabalhos futuros.

Apesar de se demostrar bastante útil a extensão ainda não se encontra no seu estado final, sendo proposto aqui como trabalhos futuros para mesma a implementação de um mecanismo de exportação da rede contida na ferramenta NetLogo para um arquivo no formato GML, pois sabe-se que em vários casos a rede importada passa por algumas modificações, como adição ou retirada de nós e links, isso ocorre para o melhoramento do desempenho das redes ou para melhorar os dados gerados pelas simulações.

Também se propõe o melhoramento do desempenho da extensão, podendo este ser alcançado a parti do melhoramento dos meios de adquirir e gerenciar os dados obtidos do arquivo a ser importado e a parti do melhorar dos meios de criação dos agentes na ferramenta, obtendo com isso o aumentando da velocidade de importação e diminuindo o consumo de processamento do equipamento utilizado.

REFERÊNCIAIS

ANDRADE, Maria Margarida de. **Introdução à metodologia do trabalho cientifico:** elaboração de trabalhos na graduação, 9. ed. São Paulo: Atlas, 2009.

BIO, Sergio Rodrigues. **Sistema de Informação** Um Enfoque Gerencial, 2ª Ed. São Paulo: ATLAS S.A., 2008.

CARISSIMI, Alexandre da Silva, ROCHOL, Juergen & GRANVILLE, Lissandro Zambeneditti. **Redes de computadores:** da Série Livros didáticos informática UFRGS, Volume 20 Porto Alegre: Bookman, 2009.

DOCUMENTAÇÃO DA LINGUAGEM SCALA. **Introduction,** Disponível em: <http://docs.scala-lang.org/tutorials/tour/tour-of-scala.html> acesso em 31 de março de 2014.

DOCUMENTAÇÃO INTERNET TOPOLOGY ZOO. **Documentation,** Disponível em**:** **<**http://www.topology-zoo.org/documentation.html> acesso em 31 de março de 2014.

FREITAS FILHO, Paulo. J.ose de. **Introdução a modelagem e Simulação de Sistemas.:** com Aplicação na Área,2ª Ed. Santa CatarinaFlorianópolis: Visual

HIMSOLT, Micael. **GML:** A portable Graph File Format, Passau, Alemanha: Universidade de Passau,1996.

ISSARIYAKUL, Teerawat & HOSSAIN, Ekram. **Introduction to Network Simulator NS2,** 2ª Ed. Nova Iorque: Spring, 2009.

JAIN, Raj. **The Art of Computer Systems Performance Analysis:** Techniques for Experimental Design, Measurement, Simulation, and Modeling, 1ª Ed. Nova York: Wiley-Interscience Abril, 1991.

KUROSE, James F. & ROSS, Keith W. **Redes de computadores e a internet:** Uma abordagem top-down, 5. ed. São Paulo: Editora Pearson, 2010.

MELLO, Braulio Adriano de. **Modelagem e Simulação de Sistemas.** Santo Ângelo: Universidade Regional Integrada do Alto Uruguai e das Missões, outubro de 2001.

ODERSKY, Martin. **What is Scala?,** Disponível em: <http://www.scala-lang.org/what-is-scala.html> acesso em 31 de março de 2014.

PRESSMAN, Roger S. **Engenharia de software:** uma abordagem profissional, 7. Ed. Porto Alegre: AMGH 2001.

PRIMETRICA. **Global Internet Geography**, disponível em: <http://www.telegeography.com/products/gig/index.php> acesso em 31 de março de 2014.

REDE NACIONAL DE ENSINO, **RedeIpê.** Disponivel em: <http://www.rnp.br/ipe/> acesso em 31 de março de 2014.

SCHINZ, Michel e HALLER, Philipp. **A Scala Tutorial for Java Programmers,** versão 1.3 Suiça, EPFL 16 de Janeiro de 2014.

SHANNON, Robert E. **System simulation:** the art and science. Englewood Cliffs, N.J: Prentice Hall, 1975.

HILLIER, Frederick S. e LIEBERMAN, Gerald J. **Introdução à Pesquisa Operacional.** 8ª Ed. São Paulo: MacGraw-Hill, 2006.

STAIR, Ralph M. e Reynolds. **Princípios de Sistemas de Informação,** 6ª Ed. São Paulo: CENGACE Learnig, 2006.

VERGARA, Silvia Constant. **Projetos e Relatórios de Pesquisa em Administração.** 2ª Ed. São Paulo: ATLAS S.A., 1998.

WILENSKY, Uri. **What is NetLogo?,** Disponivel em: <http://ccl.northwestern.edu/netlogo/docs/> acesso em 31 de março de 2014.