

FUNDAÇÃO GETULIO VARGAS  
ESCOLA DE ADMINISTRAÇÃO DE EMPRESAS DE SÃO PAULO

Fabrizio Croce Turtelli

**Formação de redes sociais com base no modelo de *Preferential Attachment*:  
uma abordagem com uso de programação multiagente**

SÃO PAULO - SP  
2015

Fabrizio Croce Turtelli

**Formação de redes sociais com base no modelo de *Preferential Attachment*:  
uma abordagem com uso de programação multiagente**

Relatório de Pesquisa apresentado à Escola de Administração de Empresas de São Paulo da Fundação Getulio Vargas como requisito do Programa Institucional de Bolsas de Iniciação Científica (PIBIC) do CNPq/GVpesquisa.

Campo de conhecimento: Administração

Orientador: Professor Dr. Júlio César Bastos de Figueiredo

SÃO PAULO – SP  
2015

Fabrizio Croce Turtelli

**Formação de redes sociais com base no modelo de *Preferential Attachment*:  
uma abordagem com uso de programação multiagente**

Relatório de Pesquisa apresentado à Escola de Administração de Empresas de São Paulo da Fundação Getúlio Vargas como requisito do Programa Institucional de Bolsas de Iniciação Científica (PIBIC) do CNPq/GVpesquisa.

Campo de conhecimento: Administração

Data da aprovação: \_\_\_\_/\_\_\_\_/\_\_\_\_

**Banca Examinadora:**

---

Professor orientador – FGV-EAESP

---

Avaliador externo ou interno - Instituição

---

Coordenador da Iniciação Científica  
FGV-EAESP

## RESUMO

O objetivo deste trabalho é aprofundar o conhecimento sobre o processo de formação de redes sociais. O estudo proposto se dividiu em duas partes, concluídas por uma entrega parcial e uma entrega final. A primeira parte foi composta, principalmente, pelo aprofundamento teórico nos tópicos de *Social Network Analysis* (SNA) e pela tradução do modelo de *Preferential Attachment*, desenvolvido por Albert Barabási em 1999, cujo modelo se encontra em linguagem de programação no sistema NetLogo. Nela, também foram feitas análises quanto ao programa, no intuito de encontrar aspectos não tratados no modelo original e que possam ser aprimorados. Esses aspectos e melhorias fizeram parte da segunda parte do projeto. Nessa entrega final, foram realizadas alterações no programa a fim de aperfeiçoar, facilitar e desenvolver tanto as informações como o entendimento do usuário para com o uso do programa em si. O foco do aperfeiçoamento desses estudos tinha como ideia entender como a difusão de informações acontece em redes sociais que seguem o modelo proposto por Barabási (1999) e os resultados encontrados estão presentes nesse relatório.

**Palavras-chave:** Análise de Redes Sociais, Formação de Redes Sociais, Teoria de Relação Preferencial, Modelagem Baseada em Agentes, NetLogo, Difusão de Informações.

## **ABSTRACT**

The proposed study is divided into two parts, the first step was finished in January 2015 and the second will be ended in July 2015, thus completing the cycle of research and development proposed in the project scope. The first step was composed mainly by theoretical studies on the topics of Social Network Analysis (SNA) and the translation of the Preferential Attachment model, developed by Albert Barabási in 1999, the model is in the NetLogo programming language system. Also, analyses were made on the program in order to find aspects not addressed in the original model that can be improved. These features and improvements were part of the second stage of the project, the changes were made in order to improve the program, facilitate and develop both the information as the user's understanding for using the program in itself. The idea of these improvements of these studies was to understand the dissemination of information in social networks that follow the model of preferential attachment showed by Barabási (1999) and the founded results are presented in this reported.

**Keywords:** Social Network Analysis, Social Network Formation, Preferential Attachment Theory, Agent Based Modeling, NetLogo

## SUMÁRIO

<b>INTRODUÇÃO .....</b>	<b>7</b>
Questões da pesquisa .....	7
Contribuições esperadas .....	7
<b>REFERENCIAL TEÓRICO .....</b>	<b>8</b>
Análise de Redes Sociais .....	8
Relações Preferenciais .....	9
Modelagem Baseada em Agentes (MBA) .....	12
<b>METODOLOGIA.....</b>	<b>16</b>
<b>DESENVOLVIMENTO DO PROJETO E DO MODELO .....</b>	<b>18</b>
Difusão de informação na rede .....	21
O processo de difusão de informação .....	22
<b>COMPARAÇÃO DO MODELO COM O PRESSUPOSTO DE <i>PREFERENCIAL</i></b>	
<b><i>ATTACHMENT</i>.....</b>	<b>24</b>
Análise da simulação .....	24
<b>CONCLUSÃO.....</b>	<b>26</b>
<b>EXPANSÃO DO PROJETO .....</b>	<b>27</b>
<b>REFERÊNCIAS.....</b>	<b>28</b>
<b>APÊNDICE A – CÓDIGO FONTE COMENTADO DO MODELO ORIGINAL .....</b>	<b>30</b>
<b>APÊNDICE B – CÓDIGO FONTE COMENTADO DO MODELO ALTERADO .....</b>	<b>34</b>

# Introdução

Este relatório tem como finalidade apresentar os resultados obtidos até o final da primeira parte do projeto “Formação de redes sociais com base no modelo de *Preferential Attachment*: uma abordagem com uso de programação multiagente”. O desenvolvimento alcançado até janeiro de 2015 foi o estudo e a tradução do modelo original (ver Apêndice A) em plataforma NetLogo 5.1.0, versão utilizada atualmente. Também houve a revisão da literatura sobre a formação e análise de redes sociais, com enfoque especial ao modelo de Albert Barabási (1999), bem como sobre modelagem baseada em agentes. Ver o Apêndice B para o cronograma de trabalho completo.

## Questões da pesquisa

Como são realizadas as relações interagentes nos diferentes ambientes de acordo com o modelo proposto por Barabási (1999)? Pretende-se também, analisar como estas relações afetam as decisões administrativas, bem como estudar outras variáveis que os agentes levam em consideração para tornarem-se parte de uma rede.

## Contribuições esperadas

Como contribuições esperadas com a realização do projeto, temos primeiramente a construção de uma base de conhecimento, por parte do orientando, sobre a programação baseada em agentes e a utilização da mesma para implementação de modelos na área da administração.

O estudo do sistema de redes sociais é de grande importância para compreender questões administrativas e logísticas, como por exemplo, a questão da difusão de produtos. Essas relações são fundamentais para tal, uma vez que fortalecem a maneira pela qual um novo produto poderá vir a ser inserido no mercado.

Também é esperado como contribuição do projeto que este estudo seja de grande relevância para o desenvolvimento de novas estratégias de atuação das instituições que visam melhorar seu desempenho através do entendimento das relações sociais, uma vez que tais conhecimentos poderão servir de subsídios para futuras pesquisas de maior complexidade, bem como para o uso em sala de aula em situações de aprendizado.

# Referencial teórico

## **Análise de Redes Sociais**

Um sistema consiste de várias partes interconectadas entre si, podendo assumir desde formas mais simples de interação, como é o caso da relação entre duas pessoas, até enormes redes de complexos sistemas interligados. A complexidade por sua vez, de acordo com Casti (1996), pode ser descrita como um comportamento de sistemas cujas relações não podem ser explicadas por métodos convencionais. Nesse caso, o princípio de relações aleatórias, que deveria ser encontrado, uma vez que se assume indiferença perante um agente e outro, acaba por não acontecer nas redes complexas de relacionamentos. Em sistemas sociais organizados em redes existem poucos “centros” responsáveis por várias relações, enquanto muitos extremos contam com poucas relações. Tais modelos de redes podem ser aplicados nos mais diversos agentes no mundo e seu estudo permite a melhor interpretação de como agir nessas relações.

De acordo com Duarte, Quandt e Souza (2008), o paradigma das redes tem sido evocado como explicação estrutural para muitos dos fenômenos comunicacionais, políticos, organizacionais e sociais do nosso tempo. (pág. 13).

Ainda segundo os autores (2008),

as redes sociais são estruturas dinâmicas e complexas formadas por pessoas com valores e/ou objetivos em comum, interligadas de forma horizontal e predominantemente descentralizada. Elas têm sido utilizadas por psicólogos, sociólogos, antropólogos, cientistas da informação e pesquisadores da área da administração para explicar uma série de fenômenos caracterizados por troca intensiva de informação e conhecimento entre as pessoas. (pág. 34).

A importância da compreensão desses sistemas se faz valer, visto que, em um futuro próximo, grande parte das estruturas cognitivas, infraestruturais e sociais, funcionará sob a forma de redes, ou estará sob sua influência direta. Grandes empresas já perceberam a importância das redes na gestão dos fluxos de informação e na geração de novos conhecimentos. (DUARTE; QUANDT; SOUZA, 2008).

Partindo de outra ótica, do ponto de vista da análise de redes sociais, o ambiente social pode ser visto com padrões e regularidades nos relacionamentos entre unidades (WASSERMAN;

FAUST, 1994). Wasserman e Faust (1994) apresentam a análise de redes sociais como uma perspectiva de pesquisa distinta no âmbito das ciências sociais e comportamentais; distinta porque a análise de rede social está baseada no pressuposto da importância das interações entre os agentes.

O estudo da formação destes padrões é extrema importância hoje onde os negócios digitais, principalmente, dependem do entendimento como a informação se difunde nestas redes.

### **Relações Preferenciais**

O *small-world model* (WATTS; STROGATZ, 1998) consiste em um modelo simplificado da estrutura de uma rede social. Esse modelo (também é conhecido como *Watts-Strogatz model*) e tem duas propriedades:

- i. O efeito *small-world*, em que a maioria dos vértices (nós ou agentes) estão conectados por um curto caminho através da rede.
- ii. Alta “clusterização”, em que duas pessoas estão mais propensas a se familiarizarem uma com a outra, se elas têm outro conhecido em comum, concentrando-se em *clusters*. (NEWMAN; BARABÁSI; WATTS, 2006)

De acordo com Wasserman e Faust (1994) uma das peculiaridades da análise de redes sociais – *Social Network Analysis* (SNA) – é exatamente o foco no aspecto relacional dos dados coletados. Em outras palavras, seu objetivo é realizar o levantamento de propriedades e conteúdos provenientes da interação entre unidades independentes.

Devido ao foco no qual as redes sociais são formadas, a simulação computacional é usada para descobrir propriedades emergentes do modelo construído e assim ganhar maior entendimento sobre a evolução de um processo dinâmico, o que seria muito difícil de modelar com técnicas matemáticas padrão. Deve-se, então, dispor de ferramentas capazes de criar, no computador, simulações de fenômenos complexos e modelos descentralizados.

O modelo estudado neste projeto é baseado no conceito de *scale-free network* (BARABÁSI, 1999), que é uma das descobertas mais importantes do campo de Network Science (BÖRNER; SANYAL; VESPIGNANI, 2007). Esse modelo é descrito também por Barabási (2003) com base em um algoritmo de formação de relações: “a probabilidade de um nó

escolher se conectar a outro nó de uma rede já existente é proporcional à quantidade de conexões que esse tenha”, o que realça as propriedades do *small-world model*.

Para Barabási (2003), as redes não são formadas de modo aleatório, uma vez que existe uma ordenação na forma como as elas se estruturam. No padrão de estruturação identificado por Barabási (2003), os “ricos ficam mais ricos” (*rich get richer*), o que quer dizer que quanto mais conexões um nó possui, maior é a chance de ele receber novas conexões. Nesse tipo de formação, chamado de relação preferencial (*preferential attachment*), um novo nó tende a se conectar com um nó pré-existente, que seja previamente mais conectado com outros nós, por isso a comparação com “ricos ficam mais ricos”. As redes, por consequência, não seriam constituídas de nós igualitários e, ao invés disso, possuiriam poucos nós altamente conectados (hubs) e uma maioria de nós com poucas conexões. Os hubs seriam os ricos, que receberiam sempre mais conexões. Estes tipos de redes, que se baseiam em características diferentes da aleatoriedade na escolha de conexões, foram denominados “redes livres de escalas”.

Em redes de formação preferencial (*preferential attachment*) as inter-relações entre os agentes não são formadas de maneira totalmente randômica. Nestas estruturas poucos agentes contam com muitas relações, enquanto muitos agentes contam com poucas e, na maioria das vezes, contam com apenas uma relação. Neste trabalho, está sendo estudado o modelo de formação preferencial proposto por Barabási (1999), modelo esse que assume as premissas de ser livre de escala (*scale-free*) (vide Figura 1 e Figura 2), em que a distribuição das relações entre os agentes assume uma forma diferente do modelo de agrupamento randômico normalmente esperado.

Com maior entendimento a respeito das interações entre agentes ocorridas durante as formações de redes, podem-se atingir modelos com maior representatividade a respeito das interações entre os mais diferentes agentes e contribuir com uma maior previsibilidade das relações entre os indivíduos.

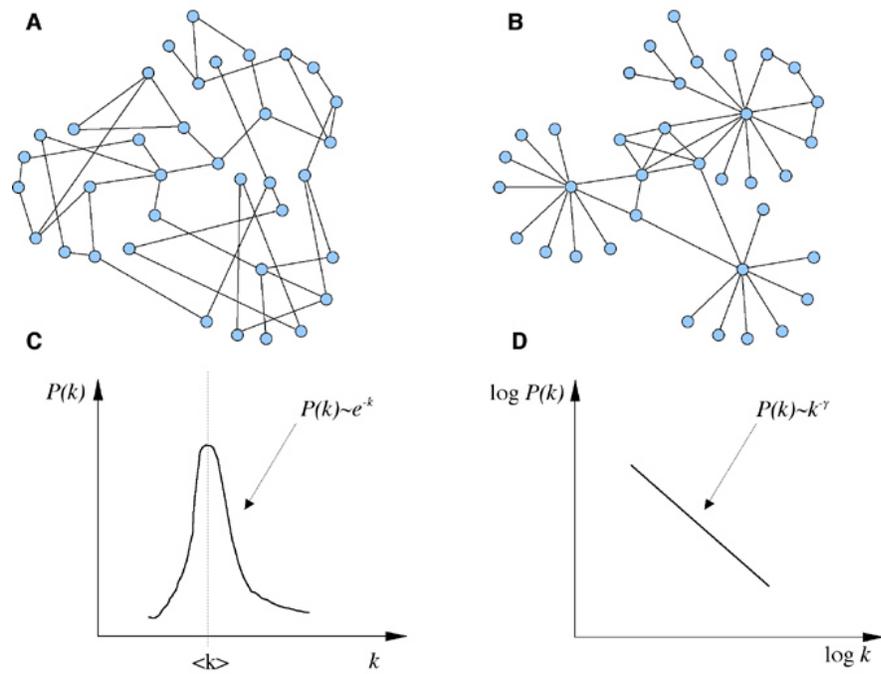


Figura 1 - Comparação entre *random network* (A) e *scale-free network* (B)  
 Fonte: <http://www.alexeikurakin.org/main/lecture4Ext.html>. Acesso em 27 mar. 14

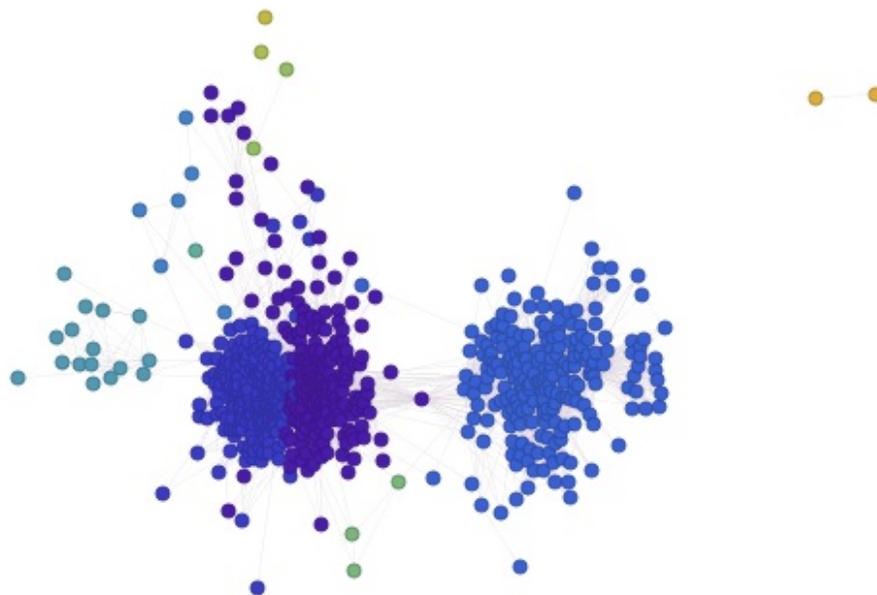


Figura 2 – Modelo *free-scale network*, obtido a partir da análise de um perfil do Facebook,  
 Fonte: <http://www.wolframalpha.com/facebook/>. Acesso em 22 mar. 14

## **Modelagem Baseada em Agentes (MBA)**

A Modelagem Baseada em Agentes (MBA) permite realizar, por meio de programação, simulações virtuais de redes sociais, que podem vir a ser observadas e analisadas posteriormente. Nesse tipo de modelagem os indivíduos, também chamados de agentes, e suas interações, entre si e com seu meio ambiente, são explicitamente representados em um programa computacional. Esses agentes, bem como os agentes das redes previamente descritos, podem ser, por exemplo, pessoas, animais, grupos ou células, além de entidades que não têm uma base física, como a execução de uma tarefa, como a coleta de informações ou, teoricamente, a modelagem da evolução da cooperação (TESFATSION, 2006). Esse é um modelo de experimentação numérica, que simula as interações entre os agentes e cobre a necessidade de outras formas de pesquisa, como pesquisas de campo.

A MBA está associada a um tipo chamado de programação orientada a objetos. Em linguagens de programação orientadas a objetos, dados e métodos de programação ou procedimentos são encapsulados em objetos que podem manipular os seus próprios dados e interagir com outros objetos de programação. O encapsulamento ou entificação de programas como objetos faz a representação de entidades do mundo real, como pessoas ou células, intuitivamente representáveis por objetos e é necessário especificar as informações que essas pessoas precisam ter e as regras de como elas interagem umas com as outras e seu meio ambiente (MACULAN FILHO; PEREIRA, 1980). Chama-se também de programação multiagente pelo fato da representação dos indivíduos (agentes) que, tecnicamente falando, são os objetos da programação.

A programação multiagente é uma técnica de modelagem que tem crescido em uso nos últimos anos (TESFATSION, 2006). Por meio dela é possível observar como agentes individuais, pessoas, produtos ou organizações, interagem entre si e com seu ambiente (AXELROD, 1997). Simulação computacional é usada para descobrir propriedades emergentes do modelo e assim ganhar entendimento dentro de um processo dinâmico, o que seria muito difícil de modelar com técnicas matemáticas usuais. Deve-se então dispor de ferramentas capazes de criar, no computador, simulações de fenômenos complexos e modelos descentralizados.

Existem vários ambientes de programação especificamente projetados para modelagem baseada em agentes. Entre eles destacam-se: StarLogo, Microworlds, Swarm, RePast, NetLogo

(AXELROD, 1996). A ferramenta que será utilizada nesta pesquisa é a NetLogo que é uma linguagem de programação multiagente, de uso gratuito, em que fenômenos naturais e sociais podem ser simulados (TISUE; WILENSKY, 2004). Para os autores, a linguagem é bem adaptada para modelagem de sistemas complexos em evolução ao longo do tempo. A linguagem permite que os modeladores possam dar instruções para muitos "agentes", agindo de forma independente ou mesmo em operação simultânea, o que permite explorar as correlações entre comportamentos em nível micro de indivíduos e os padrões em nível macro que emergem de suas interações.

O NetLogo permite aos usuários construir simulações e explorar seu comportamento sob diversas condições (WILENSKY, 2004). Ele foi o escolhido para a realização do projeto, pois, se trata de um ambiente de criação simples que permite que estudantes e pesquisadores possam criar seus próprios modelos, mesmo que não sejam programadores profissionais.

A ferramenta representa a geração posterior da série de linguagens de modelagem multiagente, que inclui o StarLogo (RESNICK; WILENSKY, 1993 e RESNICK, 1993), e é um aplicativo independente escrito em Java, o que permite seja executado em todas as principais plataformas de computação. É um produto estável, rápido e de acesso e uso gratuito. Vem acompanhado de extensa documentação e tutoriais e uma grande coleção de modelos de amostra. Em termos de linguagem, NetLogo é um membro da família Lisp que suporta agentes e simultaneidade da representação de suas interações. Seu uso é muito encontrado em estudos relacionados às interações multiagente. Estas simulações cobrem áreas das ciências naturais e sociais, com vastos modelos já construídos e que podem ser utilizados em estudos e até mesmo modificados em sua biblioteca.

Agentes móveis chamado de "turtles" movem-se sobre uma grade de "patches" que são também agentes programáveis. Todos os agentes podem interagir uns com os outros e executar várias tarefas ao mesmo tempo. As "tartarugas" podem tomar a forma de moléculas, animais, e os "patches" podem ser usadas para representar árvores, muros, células, terra. Tartarugas e "patches" podem ser utilizados para visualizar e estudar abstrações matemáticas e, também, para arte e jogos. Os temas abordados incluem autômatos celulares, algoritmos genéticos, feedback positivo e negativo, dinâmica populacional, redes, mercados, caos, sociedades artificiais, vida artificial, etc. Os modelos todos compartilham temas essenciais e centrais nos sistemas complexos.

O ambiente NetLogo (vide Figura 3) é completamente programável e apresenta uma interface gráfica com botões de controle, sistema de gráficos, exportação e importação de modelos, entre outros. Ele será o local onde serão realizadas as simulações e, posteriormente, as alterações no modelo de relações preferenciais.

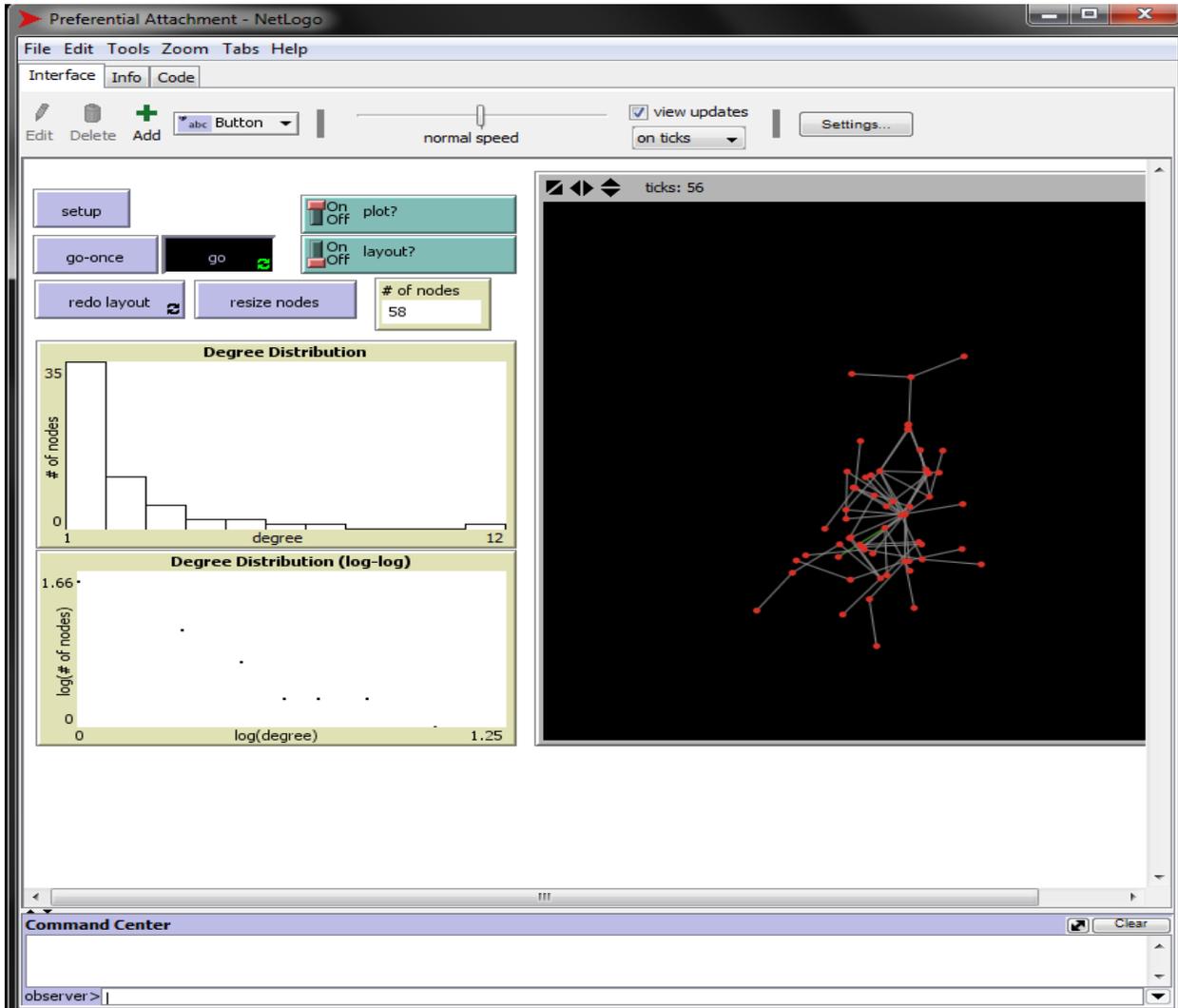


Figura 3 – ambiente de simulação NetLogo

```

File Edit Tools Zoom Tabs Help
Interface Info Code
Find... Check Procedures Indent automatically

Main Procedures

to go
  ask links [ set color gray ]
  make-node find-partner

  tick
  if layout? [ layout ]
end

to make-node [old-node]
  crt 1
  [
    set color red
    if old-node != nobody
      [ create-link-with old-node [ set color green ]
        move-to old-node
        fd 8
      ]
  ]
end

```

Figura 4 – exemplo de programação no software NetLogo.

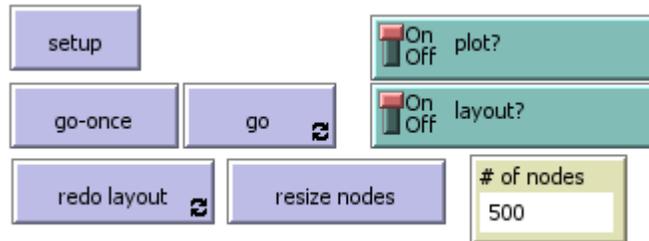


Figura 5 – botões que dão início a simulação e são responsáveis por trabalharem com o layout da simulação.

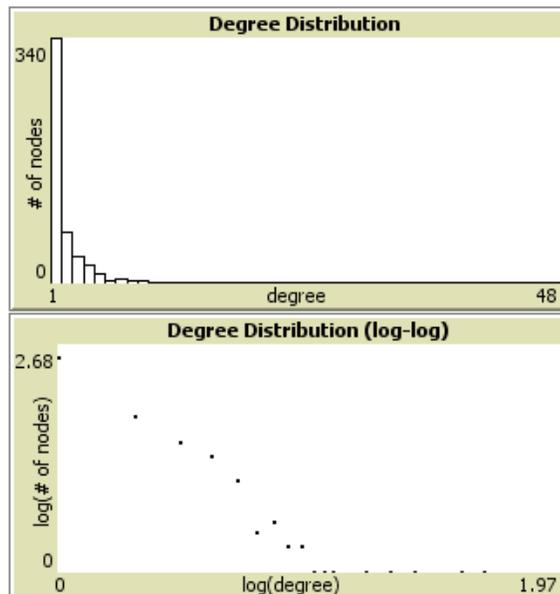


Figura 6 – tela que apresenta os resultados numéricos de uma simulação com 500 agentes.

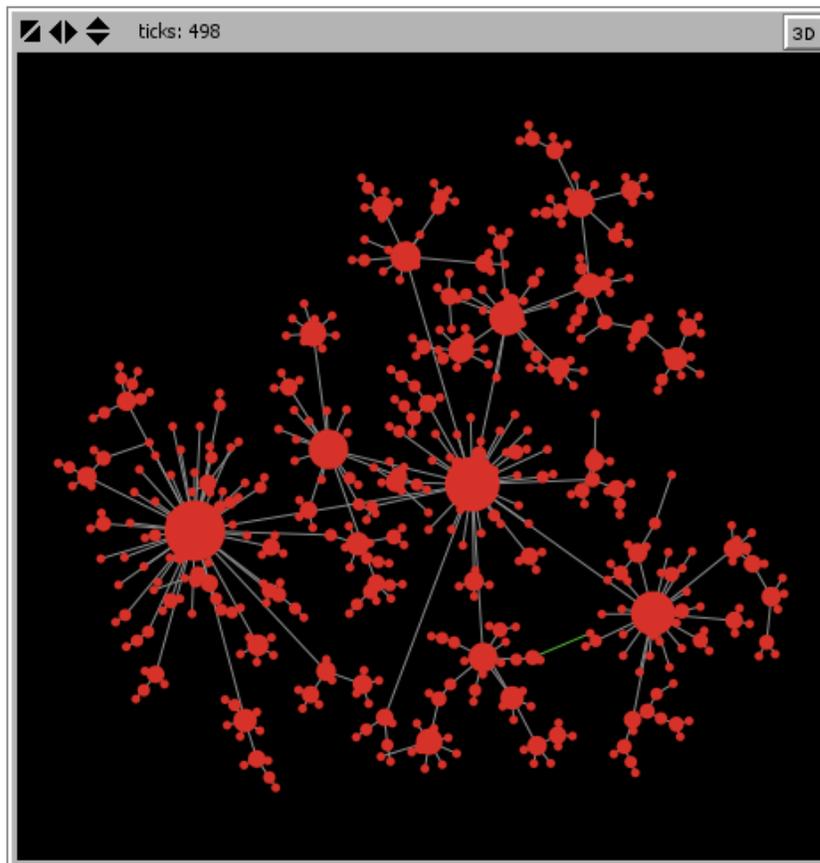


Figura 7 – tela que apresenta a representação gráfica do resultado de uma simulação com 500 agentes.

Para o estudo do funcionamento de uma rede social através do princípio de relações preferenciais, o modelo computacional original pode ser encontrado na página do software NetLogo sob o nome de “*preferential attachment*” (WILENSKY, 1999 e 2005). Este modelo simula o crescimento de redes onde, alguns “*hubs*” possuem muitas conexões, enquanto todos os outros agentes têm poucas conexões, levando em consideração os modelos elaborados por Barabási (1999).

## Metodologia

Redes sociais são usadas para se referir a um conjunto de conceitos e procedimentos analíticos e metodológicos específicos que facilitam a coleta de dados e estudo sistemático de tal padronização (FREEMAN; WHITE; ROMNEY, 1989). Para a coleta desses dados, são utilizadas então, algumas metodologias. A análise de redes sociais (*Social Network Analysis – SNA*), em particular, é uma ferramenta metodológica de ordem multidisciplinar (psicologia,

sociologia, antropologia, matemática, estatística), cuja principal vantagem é a possibilidade de formalização gráfica e quantitativa de conceitos abstraídos a partir de propriedades e processos característicos da realidade social (DUARTE; QUANDT; SOUZA, 2008). Dessa forma, modelos e teorias formulados com base em conceitos sociais podem ser matematicamente testados.

Como dito anteriormente, uma das peculiaridades da SNA é o foco no aspecto relacional (WASSERMAN; FAUST, 1994) e para obter-se informações desses, existem três fundamentos teóricos que acompanham a SNA (DUARTE; QUANDT; SOUZA, 2008):

- i. A teoria dos grafos (*graph theory*);
- ii. A teoria estatística (*statistics*)/probabilística (*probability theory*);
- iii. Os modelos algébricos (*algebraic models*).

Como descrito por Duarte, Quandt e Souza (2008), enquanto a teoria dos grafos privilegia uma análise descritiva/qualitativa de dados, os outros dois modelos são mais utilizados para teste de hipóteses e análise de redes multirrelacionais. Dessa forma, as medidas de redes permitem formalizar conceitos teóricos, avaliar modelos ou teorias e analisar estaticamente sistemas.

O sistema NetLogo, que foi escolhido para realização das simulações do modelo, é um programa escrito na linguagem de programação Java e que possui sua própria linguagem, na qual são realizada os experimentos dos modelos. Ele é uma linguagem de programação multiagente e ambiente de modelagem para simulação de fenômenos naturais e sociais complexos (TISUE; WILENSKY, 2004). Seu uso se torna adequado para a pesquisa em questão, pois ele permite aos seus usuários modelar as instruções dos agentes (chamadas de *turtles*) e do ambiente (chamado de *path*), a fim de explorar as ligações entre os comportamentos das pessoas e empresas, explorando seus comportamentos sob várias condições.

Devido as características da Modelagem Baseada em Agentes e do software NetLogo, que trabalha com a MBA, essa é a metodologia escolhida para a construção dos modelos teóricos propostos por Barabási (1999). Além disso, ela mostra-se como uma ferramenta ideal, uma vez que tem a capacidade de trabalhar nos três aspectos teóricos da *Social Network Analysis*: graficamente, probabilisticamente e algebricamente.

# Desenvolvimento do Projeto e do Modelo

Ao longo da primeira etapa do projeto, realizou-se a familiarização com a linguagem de programação multiagente e com o embasamento teórico que sustenta a teoria de relações preferenciais apresentada por Barabási (1999) e foi realizado o estudo e implementação da representação do modelo no software NetLogo. Sua representação foi feita por Wilensky (1999 e 2005) e está presente na biblioteca de modelos que o NetLogo traz e consistia nas seguintes funcionalidades descritas na figura (vide Figura 8):

- i. Botão “setup”: responsável por dar início à simulação, adicionando dois nós (agentes) interligados;
- ii. Botão “go-once”: adiciona um novo nó;
- iii. Botão “go”: adiciona infinitamente novos nós;
- iv. Botão “redo layout”: trabalha com a aparência da rede em sua representação no ambiente, tornando mais fácil sua visualização;
- v. Botão “resize nodes”: faz todos os nós terem um tamanho proporcional ao seu grau, ou seja, proporcional à quantidade de conexões que esse tem;
- vi. Chave “plot”: quando ligada, ela acelera a simulação do modelo;
- vii. Chave “layout”: quando ligada, movimenta a rede ao longo do ambiente, facilitando também a sua visualização.

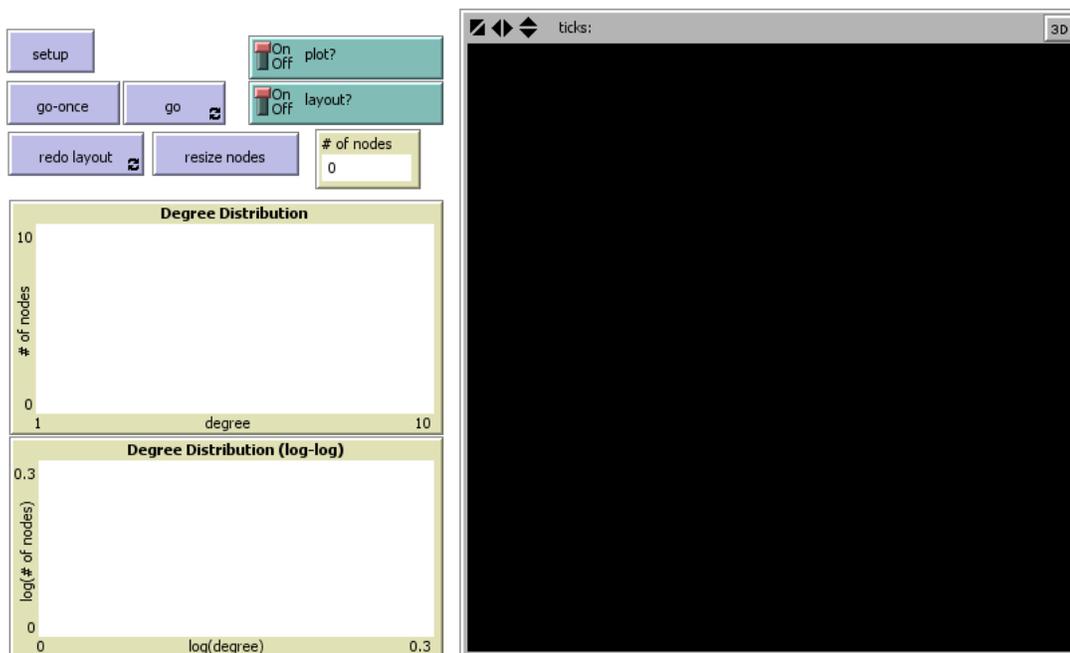


Figura 8 – ambiente de simulação completo do modelo de *Preferential Attachment* no NetLogo.

Além dos comandos acima descritos, o modelo trazia também a composição de dois gráficos: *Degree Distribution* e *Degree Distribution (log-log)*. Como o próprio nome já diz, o primeiro dele tem, no eixo x, o número de graus que os nós têm e, no eixo y, o número de nós que possui cada grau. O segundo, por sua vez, traz as mesmas informações, porém em número logarítmico, visando facilitar a visualização e a interpretação. Temos também o “# of nodes”, que apresenta ao usuário o número de nós (agentes) na simulação em questão. Por fim, temos o ambiente da simulação, que apresenta uma representação da rede criada.

O modelo e seu funcionamento foi destacado no relatório parcial da pesquisa e a segunda etapa trouxe como objetivo realizar alterações no modelo de Wilensky (1999 e 2005), que era uma representação do processo de formação de redes sociais baseadas no modelo de *preferential attachment* de Barabási (1999). Essa alteração tinha como ideia central adicionar uma representação de como informações seriam difundidas em uma rede social. Foram explicadas na entrega parcial do projeto tiveram alterações, no sentido de fundir informações em uma rede. Ver botões acrescentados na figura (vide Figura 9):

- i. Botão “Redo Layout”: trabalha com a aparência da rede em sua representação no ambiente, tornando mais fácil sua visualização;

- ii. Botão “Resize Nodes”: faz todos os nós terem um tamanho proporcional ao seu grau, ou seja, proporcional à quantidade de conexões que esse tem;
- iii. Chave “plot”: quando ligada, ela acelera a simulação do modelo;
- iv. Chave “layout”: quando ligada, movimenta a rede ao longo do ambiente, facilitando também a sua visualização;
- v. Slider “Tamanho”: escolhe quantos agentes (turtles) a rede terá;
- vi. Botão “Criar Rede”: cria uma rede com o número de agentes indicado no slider e difunde a informação entre os agentes;
- vii. Botão “Limpa Tudo”: apaga a rede criada;
- viii. Monitor “# de links”: mostra o número de conexões que a última turtle criada possui;
- ix. Monitor “# de passos”: mostra quantos passos foram necessários para a difusão de informação na rede
- x. Monitor grande: consiste na compilação dos dados número de links e número de passos de cada turtle, para exportação ao Excel, onde os dados são analisados.

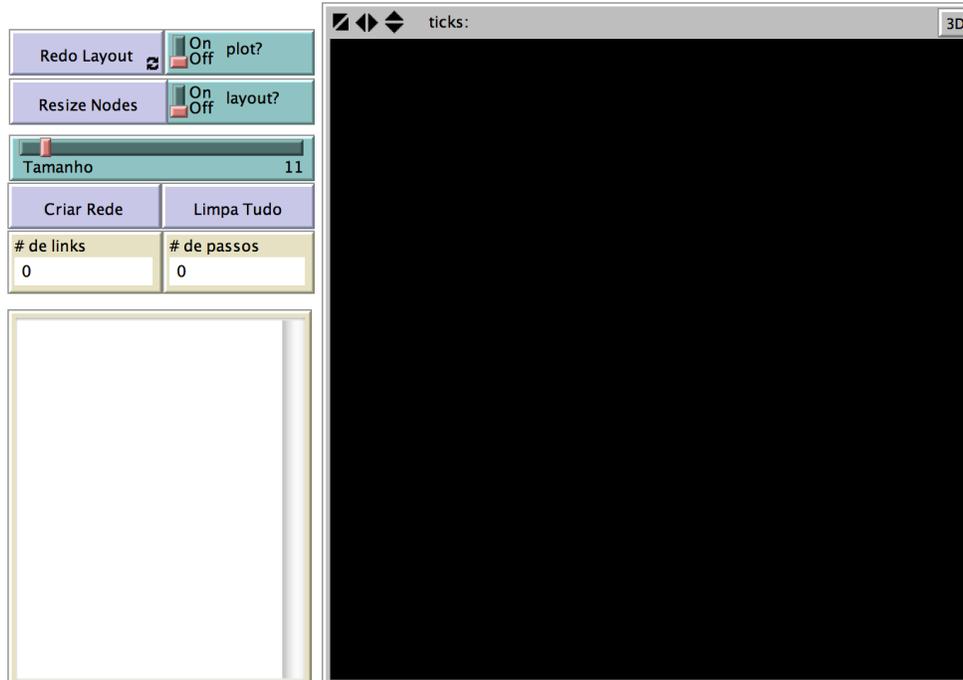


Figura 9 – ambiente de simulação alterado do modelo de *Preferential Attachment* no NetLogo.

## Difusão de informação na rede

Ao clicar-se no botão “Criar Rede”, o programa irá construir uma rede com o tamanho especificado pelo slider “Tamanho” e, imediatamente após sua criação, ela irá simular a difusão de informações por cada um dos nós dessa rede, retomando como informação o número de links que cada turtle possui, bem como o número de passos necessários para que todos os agentes da rede recebam a informação. Quando uma turtle é tida como “selecionada”, o que é realizado através de um processo aleatório do sistema sendo que, a cada simulação, cada turtle passa pelo papel de “selecionada” uma vez, ela inicia a difusão de informação na rede. Esse processo é feito por níveis, ou seja, no primeiro passo a turtle recebe a informação; no segundo passo suas vizinhas (neighbors) recebem a informação; em um terceiro passo, os vizinhos dos vizinhos recebem a informação, até que todas as turtles tenham recebido o conteúdo. Acabado o processo, o sistema plota o par ordenado “número de links X número de passos” e passa a realizar o mesmo processo com a próxima turtle selecionada. O resultado dessa simulação pode ser observado na simulação com 5 agentes da Figura 10.

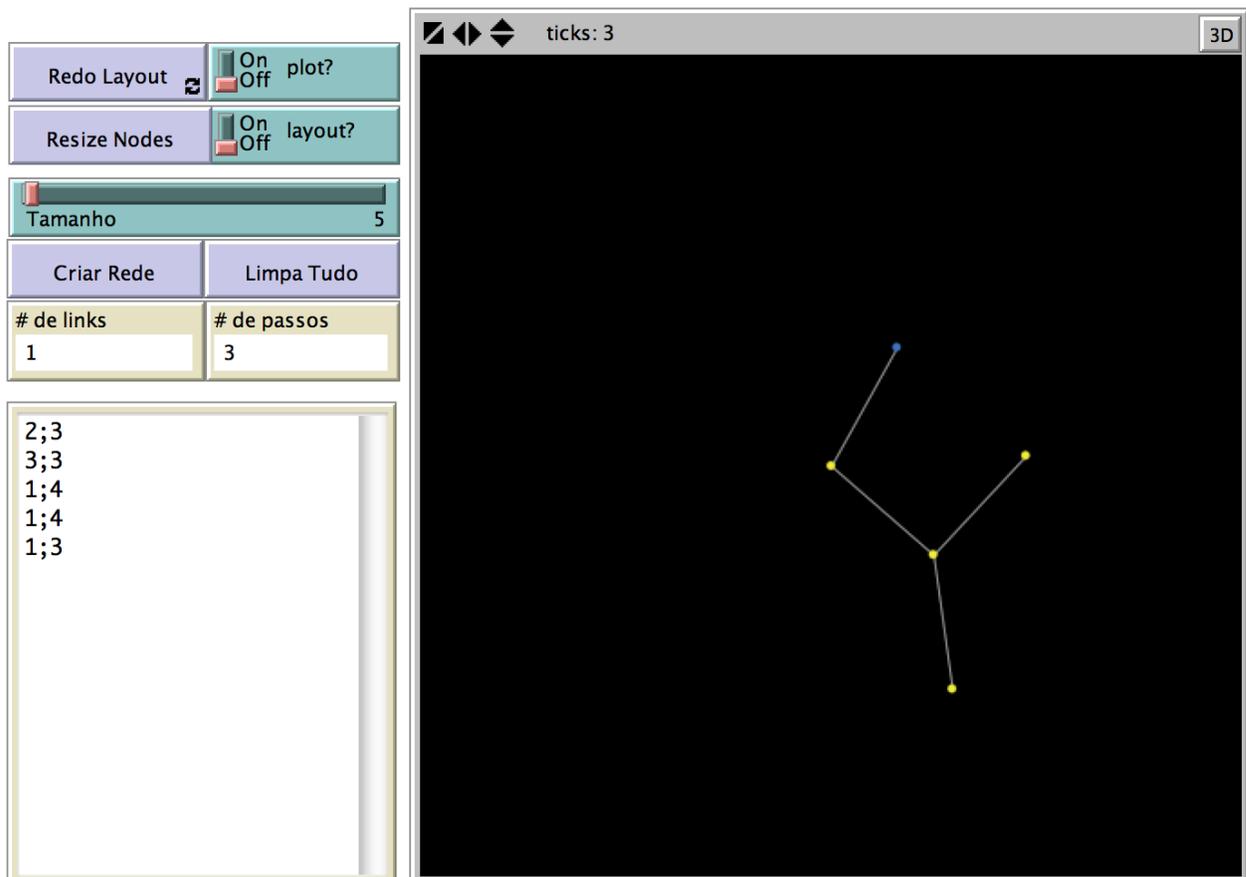


Figura 10 – simulação de difusão de informação em uma rede com 5 agentes.

## O processo de difusão de informação

A simulação exemplifica como se transmite uma informação em uma rede. Esse processo serve como base para entender ações empresariais, como por exemplo, a estratégia de onde implementar uma campanha publicitária, visando atingir o maior número de pessoas (representado pelas turtles), em um menor tempo (representado pelos passos). A difusão de informação na rede, bem como a contagem dos passos, funcionam da seguinte forma:

1. **Primeiro passo:** uma turtle selecionada recebe a cor azul, indicando que ela é a portadora da informação; as outras turtles mantêm a cor vermelha, indicando que ainda não possuem a informação. Ver figura 11.

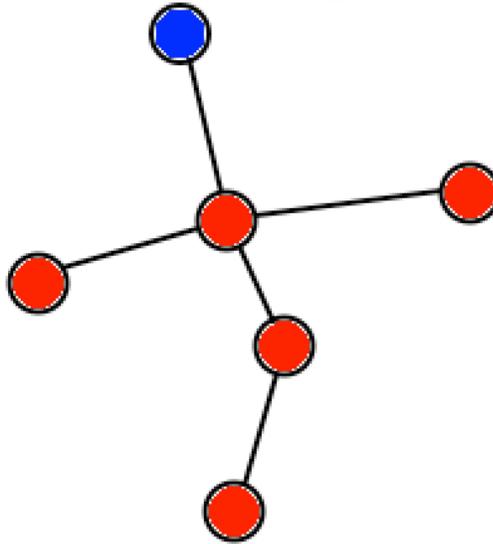


Figura 11 – primeiro passo da difusão de informação na rede.

2. **Segundo passo:** as turtles vizinhas da selecionada recebem a informação; sua cor é mudada para amarelo, indicando que elas já estão informadas. Ver figura 12.

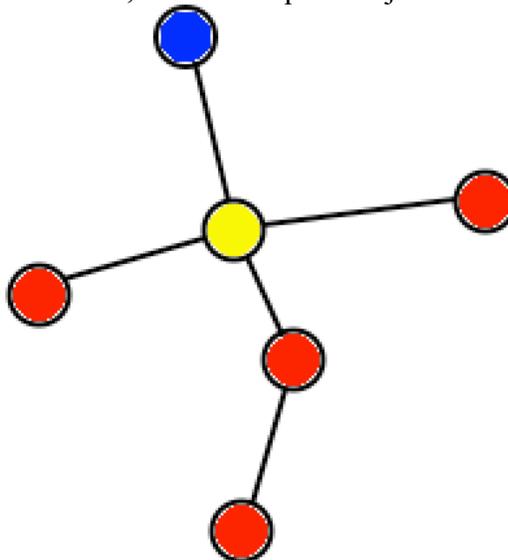


Figura 12 – segundo passo da difusão de informação na rede.

3. **Terceiro passo:** as turtles vizinhas das vizinhas da selecionada recebem a informação; suas cores mudam para amarelo indicando que foram informadas. Ver figura 13.

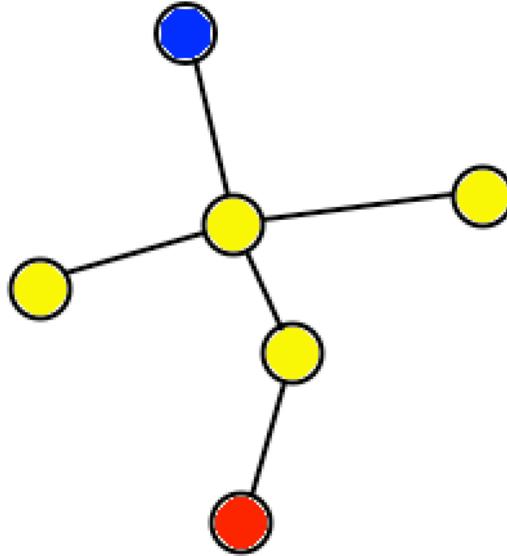


Figura 13 – terceiro passo da difusão de informação na rede.

4. **Quarto passo:** o processo se repete até que todas as turtles tenham recebido a informação. Nessa simulação a turtle selecionada (cor azul) tinha apenas 1 link e foram necessários 4 passos para difundir a informação para toda a rede, retomando como resultado o par ordenado "1;4". Ver figura 14.

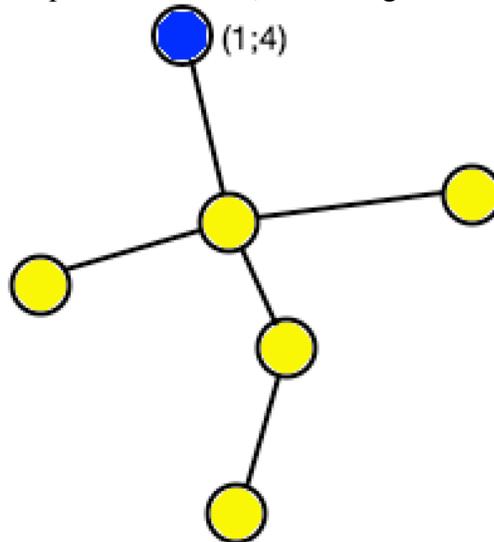


Figura 14 – quarto passo da difusão de informação na rede.

5. **Quinto passo:** após a retomada do resultado para a turtle selecionada, o mesmo processo se aplica para as outras turtles, obtendo um par ordenado de número de conexões e passos necessários para a difusão da informação para cada agente do modelo.

# Comparação do modelo com o pressuposto de *preferencial attachment*

Para analisar os impactos da difusão de informação em redes sociais formadas com o pressuposto de *preferencial attachment*, realizou-se simulações com redes compostas por 10 agentes cada e os resultados foram compilados em apenas um gráfico para melhor visualização. O resultado esperado era que quando o início da difusão da mensagem fosse realizado por um *hub* com múltiplas conexões, essa informação levaria menos tempo para atingir toda a rede. Em um ambiente real, por exemplo, isso poderia ser explicado pela facilidade de comunicação, uma vez que esse agente possuiria mais conexões com outros participantes da rede. A seguir, pode-se observar a análise dos dados, bem como as observações nela ela realizadas.

## Análise da simulação

O gráfico da figura 15 consiste na compilação de 10 simulações de redes formadas com o princípio descrito por Barabási (1999), cada uma dessas redes contendo 10 agentes. O objetivo dessa plotagem foi procurar uma relação matemática entre o número de conexões que um agente possui, representado pelo eixo número de links, e o número de passos que são necessários para a que todos os outros 9 agentes dessa rede recebam a informação que ele está a difundir.

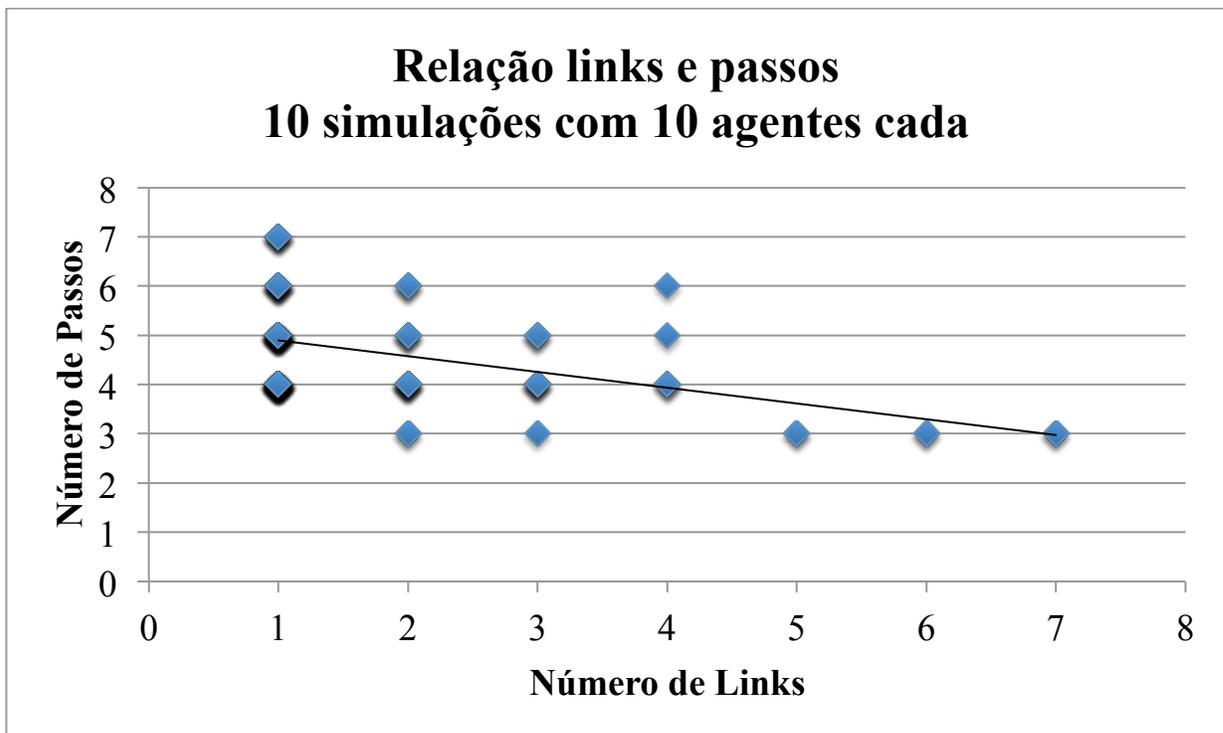


Figura 15 – gráfico da compilação dos resultados das simulações

O resultado encontrado mostra que de fato há uma relação entre esses dois fatores, todavia essa relação não é tão expressiva como era-se esperado e isso pode levantar algumas hipóteses que podem ser trabalhadas de modo a melhorar a representatividade do modelo construído. Uma das questões a serem debatidas é o fato de as simulações não levarem em consideração a probabilidade de um agente aceitar a informação que está sendo enviada para ele, fazendo com que *hubs populares* tenham potencial de difusão igual à agentes com poucas conexões, o que não deve ser observado em ambiente real.

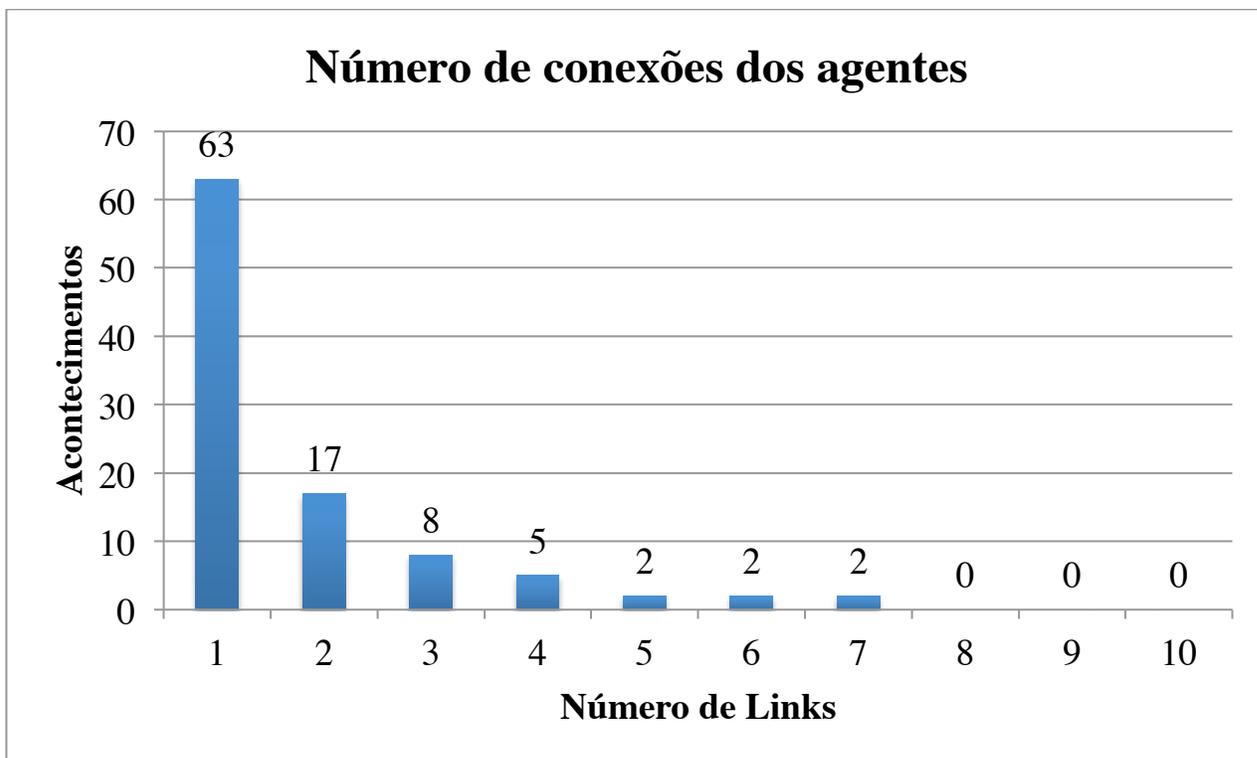


Figura 16 – histograma referente ao número de conexões que os agentes possuem

Outra comprovação interessante do modelo foi a relação entre o número de links que pode ser observada através da compilação dos dados. O histograma da figura 16 mostra que, nas simulações, obtiveram-se muitos agentes com poucas conexões (vide que 63% das turtles possuíam apenas um único link). Isso é resultado da simulação, que fora programada a fim de atender o pressuposto de *preferential attachment* de Barabási (1999) e, com essa comprovação obtida, possível ressaltar a alta representação do modelo.

# Conclusão

Esse relatório conclui meu ciclo de pesquisa sobre a formação de redes sociais baseada na teoria de *preferential attachment* proposta por Barabási (1999), bem como a modificação realizada no modelo. A modificação visou adicionar na simulação o processo de difusão de informação em uma rede, de modo a verificar as vantagens práticas que o estudo e entendimento das redes sociais podem trazer. Muitos resultados foram atingidos, tanto no âmbito pessoal, como no estudo das redes sociais e seu funcionamento.

Os modelos criados no software NetLogo ilustram a formação de redes sociais através da interação entre os agentes criados virtualmente. Analisando os cenários, observou-se que o processo de difusão de informação ocorre de maneira mais rápida quando o agente responsável por iniciar a transmissão desta é *hub* com várias conexões. Isso acelera o processo, diminuindo o tempo necessário para que todos os agentes tenham recebido a mensagem, o que em termos administrativos pode ilustrar a redução de custos, aumento da eficiência comunicativa e explicar o porque *hubs* com mais conexões tendem a receber mais conexões do que outros *hubs* com menos parceiros atrelados a eles.

A análise dos cenários simulados trouxeram os resultados que comprovavam o que era aguardado, onde a difusão através de *hubs* com mais conexões, chamados por Barabási e Bonabeau (2003) de "nós populares", diminuiria o tempo do processo. Todavia, isso ocorreu de maneira menos acentuada do que era-se esperado, visto que as curvas geradas mostram apenas uma leve tendência da redução dos passos. Uma das justificativas que pode ser atribuído a isso é o fato de que o modelo não considera a probabilidade de absorção da informação que está sendo passada. Ou seja, nesse caso, toda e qualquer informação transmitida seria automaticamente aceita pelo agente receptor, o que nem sempre é verdade no mundo real. A questão do número de conexões que a turtle possui pode ser um fator decisivo para isso, onde agentes com mais conexões têm mais credibilidade que outros com menos conexões, sendo assim "formadores de opiniões", tendo probabilidade maior que os agentes receptores aceitem a mensagem que eles estão transmitindo.

Além da comprovação do modelo, ainda que de maneira menos acentuada que o esperado, o projeto possibilitou grande aprendizado nos temas abordados, como programação baseada em agentes, estudo de redes sociais e o modelo proposto por Barabási (1999). Trabalhar em um projeto de iniciação científica foi uma oportunidade valiosa e que permitiu conhecer o funcionamento de pesquisas e trabalhos científicos, sendo um projeto muito envolvente em minha formação como aluno. Como próximas etapas têm-se o aprimoramento da simulação dos modelos de redes sociais, como aqui descritos.

# Expansão do Projeto

A entrega desse relatório conclui o ciclo de iniciação científica relacionado à temática de formação de redes sociais com o pressuposto assumido por Barabási (1999) de *preferential attachment*, bem como a análise de difusão de informações na rede. Porém, existem abordagens mais profundas que podem vir a ser feitas, visando o aperfeiçoamento das temáticas abordadas. Um exemplo seria adicionar o nível de adoção da informação, dada o número de nodes que um agente difusor possui.

Com a adição desse pressuposto, espera-se a comprovação de que a difusão de informações ocorra de maneira mais rápida e eficiente quando iniciada por *hubs* populares. Esses *hubs* assumiriam a papel de agentes formadores de opinião, sendo pontos estratégicos na análise do processo de comunicação em redes sociais. Esse processo foi descrito na conclusão do projeto, entretanto, existem outras modificações que podem vir a aprimorar a representatividade do modelo e serem incluídas em uma possível expansão do projeto, são elas:

- Buscar novas equações que ajudem a aproximar o modelo da realidade;
- Inserir fatores de restrição geográficas na rede, ou seja, quanto mais longe uma turtle encontrar-se da outra, mais difícil seria a recepção da informação;
- Inserir mais gráficos dentro do programa que resultem em mais informações disponíveis por simulação;
- Encontrar forma de deixar o programa ainda mais intuitivo para os pesquisadores.

# Referências

AXELROD, R. **The complexity of cooperation**: agent-based models of competition and Collaboration. Princeton, NJ, Princeton Univ. Press, 1996.

\_\_\_\_\_. Advancing the art of simulation in the social sciences. **Complexity**, v. 3, n. 2, p. 16-22, 1997.

BARABÁSI, A-L; ALBERT, R. Emergence of scaling in random networks, **Science**, v. 286, p. 509-512, 1999.

BARABÁSI, A-L. **Linked**: how everything is connected to everything else and what it means for business, science and everyday life. Cambridge: Plume, 2003.

BARABASI, A-L; BONABEAU, E. Scale-free. **Scientific-american**, 2003.

BÖRNER, K.; SANYAL, S.; VESPIGNANI, A. Network Science. **Annual Review of Information Science and Technology**, v. 41 (1), p. 537–607, 2007.

CASTI, J. L. **Would-be worlds**: how simulation is changing the frontiers of science. New York: John Wiley & Sons, 1996.

DUARTE, F.; QUANDT, C.; SOUZA, Q. **O tempo das redes**. São Paulo: Perspectiva, 2008.

FREEMAN, L. C.; WHITE, D. R.; ROMNEY, A. K. (Ed.). **Research methods in social network analysis**. Transaction Publishers, 1989.

NEWMAN, M; BARABÁSI, A.-L.; WATTS, D. J. (Ed.). **The structure and dynamics of networks**. Princeton University Press, 2006.

MACULAN FILHO, N.; PEREIRA, M.V.F. **Programação linear**, Atlas, São Paulo, 1980.

RESNICK, M.; WILENSKY, U. Beyond the deterministic, centralized mindsets: New thinking for new sciences. **Annual Conference of the American Educational Research Association**, Atlanta, GA, 1993.

RESNICK, M.; Changing the Centralized Mind. **Technology Review**, 32-40, 1994.

TESFATSION, L. Agent-Based computational economics. In: TEFATSION, L.; HARRINGTON, J. E. (eds). **Handbook of Computational Economics**, v. II , 2006, p.831-880.

WASSERMAN, S.; FAUST, K. **Social network analysis**: Methods and Applications. NY: Cambridge, 1994.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of ‘small-world’ networks. **Nature**, v. 393, n. 6684, 440-442, 1998.

WILENSKY, U. NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999. Disponível em: <<http://ccl.northwestern.edu/netlogo/>>. Acesso em agosto de 2014.

WILENSKY, U. NetLogo preferential attachment model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. 2005. Disponível em <<http://ccl.northwestern.edu/netlogo/models/PreferentialAttachment>>. Acesso em agosto de 2014.

# Apêndice A – código fonte comentado do modelo original

```
;;Setup Procedures
;; sequência de comandos que dá início a programação (executada ao clicar-se no botão "setup")
to setup
  clear-all
    ;; "limpa" o ambiente de trabalho
  set-default-shape turtles "circle"
    ;; define o formato dos agentes (turtles) para círculo
  ;; a sequência de comandos a seguir realiza uma conexão (link) entre os dois primeiros nodes
  ("nós"), que são os agentes (turtles)
  make-node nobody
    ;; make node é uma procedure (sequência de comandos) que será explicada mais
    adiante no programa e aqui é usada para criar o primeiro node sem estar ligado à
    nenhum outro agente
  make-node turtle 0
    ;; cria o segundo node, já conectado ao primeiro
  reset-ticks
    ;; comando que deve ser inserido ao final das procedures setup
end
;;Main Procedures
;; as procedures a seguir (go, make node e find-partner) são responsáveis por realizar as interações da
simulação
to go
  ask links [ set color gray ]
    ;; é atribuída a representação gráfica do novo link a cor verde, enquanto os links
    anteriores recebem a cor cinza
  make-node find-partner
    ;; find-partner é uma outra procedure (que será explicada a seguir) e nesse caso serve
    como parâmetro à outra procedure (make-node)
    ;; ou seja, é criado um novo agente, já criado à um "partner" (outro agente já existente)
  tick
    ;; tick serve de contador e vem ao final das procedures go
```

```

    if layout? [ layout ]
        ;; essa é um chamado a uma das procedures responsáveis pelo layout do programa, ou
        ;; seja, que influenciam apenas a parte da representação gráfica do modelo
    end

to make-node [old-node]
    ;; essa procedure é utilizada para a criação de um novo nó (ou agente, como
    ;; costumamos chamar) e
    crt 1
        ;; crt é uma abreviação ao comando "create turtle" e o 1 é o número de turtles (agentes)
        ;; que devem ser criadas
    [
        set color red
            ;; atribui-se a cor vermelha a essa nova turtle
        if old-node != nobody
            [
                create-link-with old-node [ set color green ]
                move-to old-node
                    ;; posiciona esse novo nó próximo ao seu "parceiro" a qual ele está
                    ;; conectado
                fd 8
                    ;; o comando fd trata-se a uma abreviatura para foward, onde a turtle
                    ;; move-se o número de posições indicadas
            ]
    ]
end

```

;; A procedure a seguir é o coração do mecanismo de "preferential attachment.

;; Ela trabalha de maneira semelhante à ideia de loteria, onde quanto mais cupons (conexões) um agente tiver, maior será a chance de ele ser escolhido (no caso, para uma nova conexão).

;; Primeiro é selecionado uma conexão e após isso, um dos nós (agentes) que realizam essa conexão para receber mais uma.

```
to-report find-partner
```

```
    report [one-of both-ends] of one-of links
```

```
    ;; essa linha de comando reporta uma das extremidades do link (conexão) escolhido
```

```
end
```

```
;;Layout
```

```
;; as procedures a seguir trabalham com a parte gráfica do modelo, ou seja, com o seu layout a  
procedure resize-nodes, muda de volta o tamanho das turtles para 1
```

```
to resize-nodes
```

```
  ifelse all? turtles [size <= 1]
```

```
  [
```

```
    ;; um nó, que é uma turtle, tem formato de círculo, com diâmetro determinado pelo  
    tamanho da variável, usando SQRT (função de raiz quadrado) da quantidade de  
    conexões que ela tem. Ou seja, sua área será proporcional ao seu grau (número de  
    conexões)
```

```
    ask turtles [ set size sqrt count link-neighbors ]
```

```
  ]
```

```
  [
```

```
    ask turtles [ set size 1 ]
```

```
  ]
```

```
end
```

```
to layout
```

```
;; o número 3 a seguir é arbitrário, mais repetições trariam demora à simulação, enquanto menos  
trariam layouts pobres no queisto de representatividade
```

```
  repeat 3
```

```
  [
```

```
    ;; repeat serve para repetir a sequencia de comandos entre conxetes, quantas vezes for  
    indicado os comandos a seguir visão reduzir o tamanho das turtles ara que elas possam  
    caber no mesmo espaço, ainda que exista várias delas
```

```
    ;; quanto mais turtles nós tivermos para caber no mesmo espaço, menor deverão ser os  
    inputs
```

```
    let factor sqrt count turtles
```

```
      ;; atribui-se à variável "factor" o valor da raiz quadrada do número de turtles que  
      houver
```

```
      ;; os números aqui também são arbitrários
```

```
    layout-spring turtles links (1 / factor) (7 / factor) (1 / factor)
```

```
      ;; o comando em questão está relacionado à como as turtles serão arranjadas na  
      representação gráfica
```

```
    display
```

```

;; utiliza-se esse comando para criar uma anização mais suavizada
]
;; a sequência a seguir impede que as turtles saiam dos limites da representação gráfica
let x-offset max [xcor] of turtles + min [xcor] of turtles
let y-offset max [ycor] of turtles + min [ycor] of turtles
;; a sequência de comandos a seguir limitam a ocorrência de "grandes pulos" entre as tartarugas
set x-offset limit-magnitude x-offset 0.1
set y-offset limit-magnitude y-offset 0.1
ask turtles [ setxy (xcor - x-offset / 2) (ycor - y-offset / 2) ]
end

;; os "grandes pulos" são limitados através da procedure limit-magnitude, a seguir
to-report limit-magnitude [number limit]
  if number > limit [ report limit ]
  if number < (- limit) [ report (- limit) ]
  report number
  ;; impedimos que o número a ser reportado seja maior que o limite estipulado, e assim
  impedimos que esses "grandes pulos" ocorram
end

; Comentários por Fabrizio Croce Turtelli
; Copyright 2005 Uri Wilensky.

```

## Apêndice B – código fonte comentado do modelo alterado

```
globals
[
  passos
  num-links
  contador-turtles
]

;; used for creating a new node
to make-node [old-node]
  crt 1
  [
    set color red
    if old-node != nobody
    [
      create-link-with old-node
      [ set color gray ]
      ;; position the new node near its partner
      move-to old-node
      fd 15
    ]
  ]
end

;; This code is the heart of the "preferential attachment" mechanism, and acts like
;; a lottery where each node gets a ticket for every connection it already has.
;; While the basic idea is the same as in the Lottery Example (in the Code Examples
;; section of the Models Library), things are made simpler here by the fact that we
;; can just use the links as if they were the "tickets": we first pick a random link,
;; and than we pick one of the two ends of that link.
to-report find-partner
```

```
report [one-of both-ends] of one-of links
end
```

```
;;;;;;;;;;
;;; Layout ;;;
;;;;;;;;;;
```

```
;; resize-nodes, change back and forth from size based on degree to a size of 1
to resize-nodes
```

```
  ifelse all? turtles [size <= 1]
```

```
  [
```

```
    ;; a node is a circle with diameter determined by
    ;; the SIZE variable; using SQRT makes the circle's
    ;; area proportional to its degree
```

```
    ask turtles [ set size sqrt count link-neighbors ]
```

```
  ]
```

```
  [
```

```
    ask turtles [ set size 1 ]
```

```
  ]
```

```
end
```

```
to layout
```

```
  ;; the number 3 here is arbitrary; more repetitions slows down the
```

```
  ;; model, but too few gives poor layouts
```

```
  repeat 3 [
```

```
    ;; the more turtles we have to fit into the same amount of space,
```

```
    ;; the smaller the inputs to layout-spring we'll need to use
```

```
    let factor sqrt count turtles
```

```
    ;; numbers here are arbitrarily chosen for pleasing appearance
```

```
    layout-spring turtles links (1 / factor) (7 / factor) (1 / factor)
```

```
    display ;; for smooth animation
```

```
  ]
```

```
  ;; don't bump the edges of the world
```

```

let x-offset max [xcor] of turtles + min [xcor] of turtles
let y-offset max [ycor] of turtles + min [ycor] of turtles
;; big jumps look funny, so only adjust a little each time
set x-offset limit-magnitude x-offset 0.1
set y-offset limit-magnitude y-offset 0.1
ask turtles [ setxy (xcor - x-offset / 2) (ycor - y-offset / 2) ]
end

```

```

to-report limit-magnitude [number limit]
  if number > limit [ report limit ]
  if number < (- limit) [ report (- limit) ]
  report number
end

```

; Copyright 2005 Uri Wilensky.

; See Info tab for full copyright and license.

; MODIFICAÇÕES

;;PROCEDURE PARA CRIAR A REDE

```

to criar_rede
  clear-all
  set-default-shape turtles "circle"
  ;; make the initial network of two turtles and an edge
  make-node nobody    ;; first node, unattached
  make-node turtle 0  ;; second node, attached to first node
  reset-ticks
  while [ count turtles < Tamanho ]
  [
    make-node find-partner    ;; find partner & use it as attachment
                                ;; point for new node
    tick
  ]

```

```
if layout? [ layout ]
if (count turtles > Tamanho) [stop]
]
```

set contador-turtles 0 ;é a variável que permite em uma mesma rede, rodar todas as turtles

```
while [contador-turtles < Tamanho]
```

```
[
```

```
ask turtles
```

```
[
```

```
set color red
```

```
]
```

```
set passos 0
```

```
let contador 0
```

```
let contador2 0
```

```
let selecionado turtle contador-turtles
```

```
set num-links [count link-neighbors] of selecionado
```

```
ask selecionado
```

```
[
```

```
;set color blue
```

```
set passos 2
```

while [any? turtles with [color = red]] ;se está com a cor vermelha, significa que a turtle ainda não recebeu a informação

```
[
```

```
ask selecionado
```

```
[
```

```
ask link-neighbors
```

```
[
```

```
set color yellow
```

```
ask link-neighbors
```

```
[
```

```
if color = red
```

```
[
```

```
set color yellow
```

```

        set contador2 (contador2 + 1)
    ]
]
]
if (contador2 > 0)
[
    set passos (passos + 1)
]
]
set contador2 0
set contador (contador + 1)
set selecionado turtle contador
]
set color blue
]
output-print word word num-links ";" passos
set contador-turtles (contador-turtles + 1)
]
end

to-report conta-passos
    report passos
end

```