

Simbo

Documentação

Autor: Felipe Balabanian

12/07/2017

Introdução

O que é o Simbo ?

Simbo é uma plataforma para a simulação de botnets e outras ameaças que se propagam pela rede. Ele consiste numa aplicação sobre o simulador de eventos discretos Omnet++ e a biblioteca de rede Inet.

Com qual finalidade foi desenvolvido?

Simbo foi concebido como uma ferramenta de estudo do comportamento de botnets na rede. Com simulação como sua metodologia, porém sem perda de robustez ao utilizar-se dos protocolos de rede. Uma plataforma com versatilidade de integração com outras ferramentas computacionais como Matlab e JaCaMo visando a utilização de inteligência computacional para experimentação de novas possíveis tecnologias.

Onde posso encontrar o código fonte do projeto Simbo?

O código fonte do projeto Simbo pode ser encontrado e baixado no GitHub do grupo ReGraS localizado no seguinte endereço: <https://github.com/regras/simbo/>

Omnet++

Omnet++ é um simulador de eventos discretos baseado na linguagem C++, sendo muito utilizado para estudo em redes e telecomunicações.

Site: <https://omnetpp.org/>

OMNeT++ Discrete Event Simulator

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators.

Featured Projects



O omnet++ é apenas o core de processamento e controle da simulação, sendo necessário uma biblioteca para definir o estudo de uma área específica. Atualmente existem várias bibliotecas (Simulation Models) de redes disponíveis: Inet, Inetmanet, Veins, etc. Que podem ser encontrados em <https://omnetpp.org/models> . A biblioteca escolhida no projeto foi a Inet porquê possuía boa manutenção da comunidade e vasta extensão de módulos disponíveis.

Instalando o Omnet++

O Omnet++ não realiza uma instalação propriamente dita, o que ele faz é uma compilação do código fonte para executável baseado no processador da máquina. Por isso, uma vez compilado, você pode mover sua pasta para outro disco ou computador que possua a mesma arquitetura.

! Passo a Passo:

1 – Faça o Download dos arquivos fontes em <https://omnetpp.org/omnetpp>

OMNeT++

INTRODUCTION • DOWNLOAD • SIMULATION MODELS

Category: OMNeT++ Releases

Older versions

Test versions

OMNeT++ 5.0 (Windows)

Mirror 1Download

Release 5.0 is a result of development effort of nearly two years. This is a major release that introduces significant new features compared to the last 4.x version, for example the Canvas API (2D graphics), OpenSceneGraph-based 3D graphics support, improved logging, a new Qt-based runtime environment that will eventually replace Tkenv, and much more.

We have also taken the opportunity of the major release to improve several corners of the OMNeT++ API, and also to get rid of deprecated functionality. For porting models from OMNeT++ 4.x, see [doc/API-changes.txt](#) which lists all changes, with hints on how to update the model code.

[Read on](#) for details and notes on known issues.

[Read more](#)

04/14/2016

OMNeT++ 5.0 (Mac OS X)

Mirror 1Download

Release 5.0 is a result of development effort of nearly two years. This is a major release that introduces significant new features compared to the last 4.x version, for example the Canvas API (2D graphics), OpenSceneGraph-based 3D graphics support, improved logging, a new Qt-based runtime environment that will eventually replace Tkenv, and much more.

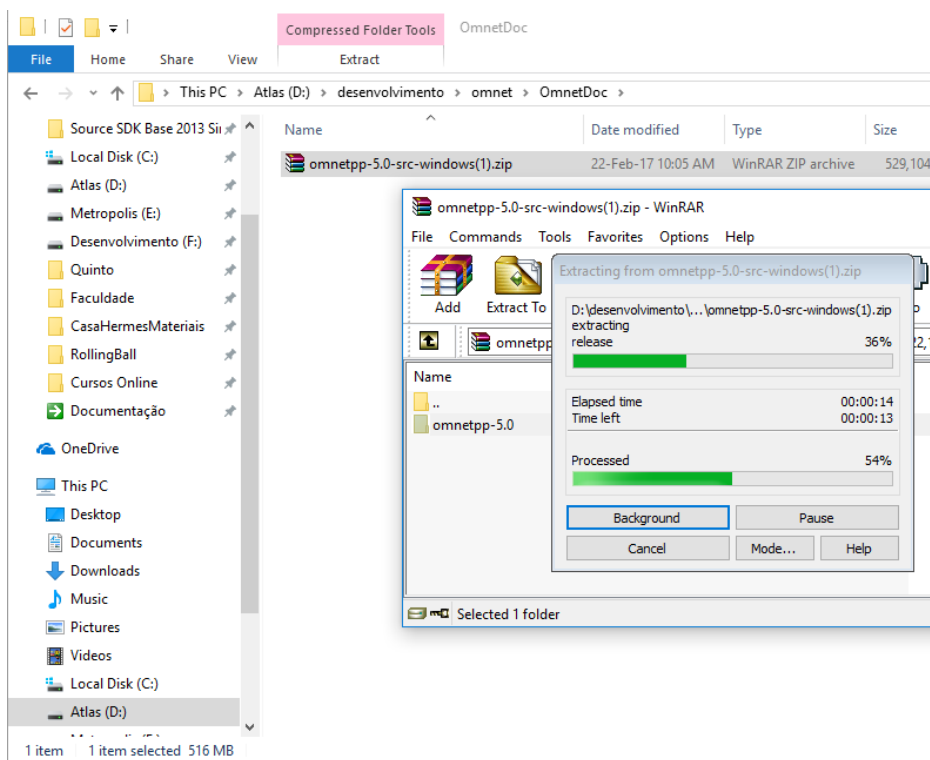
We have also taken the opportunity of the major release to improve several corners of the OMNeT++ API, and also to get rid of deprecated functionality. For porting models from OMNeT++ 4.x, see [doc/API-changes.txt](#) which lists all changes, with hints on how to update the model code.

[Read on](#) for details and notes on known issues.

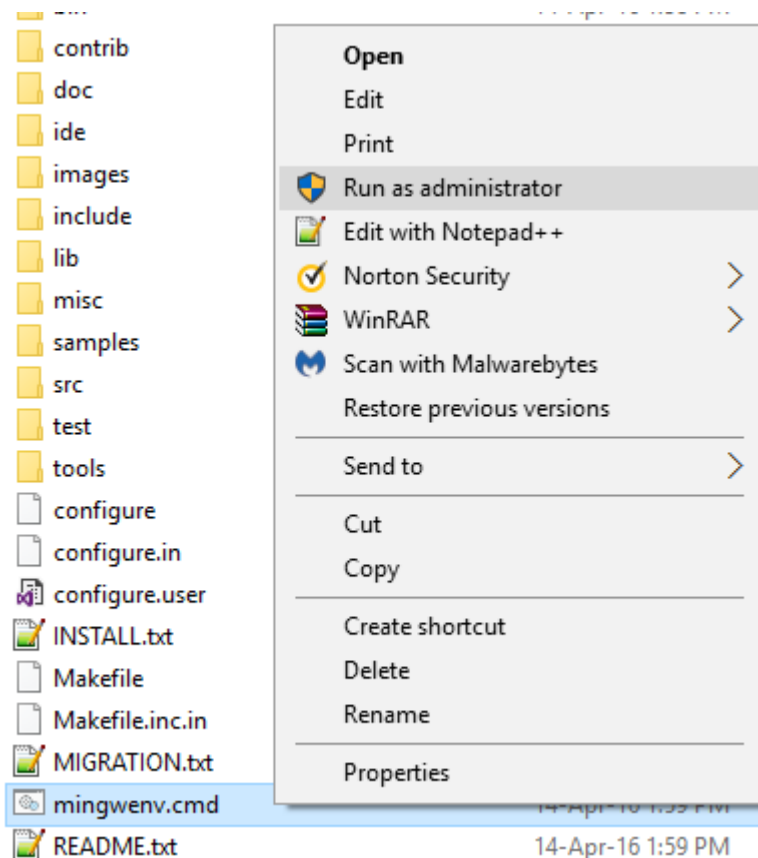
[Read more](#)

04/14/2016

2 – Extrai o zip para uma pasta.



3 – Abra o arquivo *mingwenv.cmd* com permissão de administrador.



4 – Aparecerá o prompt de comando com a mensagem da figura abaixo, pedindo para desempacotar o MinGW que vem no pacote. Tecle Enter. Se aparecer uma mensagem de erro, espere um pouco, reinicie o programa e tecle enter novamente. Até aparecer algo como na imagem X.

```
C:\WINDOWS\System32\cmd.exe

*** Welcome to OMNeT++! ***

We need to unpack the MinGW toolchain before continuing.
This can take a while, please be patient.

Press any key to continue . . .
```

```
C:\WINDOWS\system32\cmd.exe

Extracting win32\mingw32\include\osgSim\ShapeAttribute
Extracting win32\mingw32\include\osg\ShapeDrawable
Extracting win32\mingw32\include\osgEarth\SharedSARepo
Extracting win32\mingw32\include\osgDB\SharedStateManager
Extracting win32\mingw32\include\c++\5.2.0\shared_mutex
Extracting win32\usr\bin\core_perl\shasum
Extracting win32\etc\shells
Extracting win32\mingw32\lib\tcl8.6\tzdata\America\Shiprock
Extracting win32\mingw32\include\osgParticle\Shooter
Extracting win32\mingw32\lib\terminfo\73\sibo
Extracting win32\usr\lib\terminfo\73\sibo
Extracting win32\mingw32\lib\tcl8.6\tzdata\Europe\Simferopol
Extracting win32\mingw32\include\osgEarthDrivers\model_simple\SimpleModelOptions
Extracting win32\mingw32\include\osgEarthDrivers\ocean_simple\SimpleOceanOptions
Extracting win32\mingw32\include\osgEarthDrivers\sky_simple\SimpleSkyOptions
Extracting win32\mingw32\lib\terminfo\73\simpleterm
Extracting win32\usr\lib\terminfo\73\simpleterm
Extracting win32\mingw32\include\osgEarthUtil\SimplexNoise
Extracting win32\mingw32\include\osgUtil\Simplifier
Extracting win32\mingw32\lib\terminfo\73\simterm
Extracting win32\usr\lib\terminfo\73\simterm
Extracting win32\mingw32\lib\tcl8.6\tzdata\Singapore
Extracting win32\mingw32\lib\tcl8.6\tzdata\Asia\Singapore
Extracting win32\mingw32\include\osgEarthDrivers\engine_mp\SingleKeyNodeFactory
Extracting win32\mingw32\include\osgParticle\SinkOperator
Extracting win32\usr\share\bash-completion\completions\sitecopy
Extracting win32\mingw32\lib\tcl8.6\tzdata\America\Sitka
Extracting win32\mingw32\include\osgAnimation\Skeleton
Extracting win32\mingw32\include\osgEarthSymbology\Skins
Extracting win32\mingw32\lib\tcl8.6\tzdata\Europe\Skopje
```

5 – Quando toda a extração for concluída, a tela abaixo deverá aparecer. Digite: `./configure` . Para carregar as configurações da compilação.

```
/d/desenvolvimento/omnet/OmnetDoc/omnetpp-5.0

Welcome to OMNeT++ 5.0!

Type "./configure" and "make" to build the simulation libraries.
When done, type "omnetpp" to start the IDE.

/d/desenvolvimento/omnet/OmnetDoc/omnetpp-5.0$ ./configure|
```

5 - Atente para quais pacotes foram instalados ou estão ativos (no nosso caso não precisaremos de nenhum) e se o caminho(path) da pasta de bin está nas configurações do sistema (Path do Sistema Operacional)

```
WARNING: The configuration script could not detect the following packages:
      MPI (optional)  PCAP (optional)  Akaroa (optional)  Pacotes

Scroll up to see the warning messages (use shift+PgUp), and search config.log
for more details. While you can use OMNeT++ in the current configuration,
be aware that some functionality may be unavailable or incomplete.

Your PATH contains D:/desenvolvimento/omnet/OmnetDoc/omnetpp-5.0/bin. GoodPath
```

6 –Digite: *make* . O processo de compilação pode durar algumas horas. Levante e pegue um café.

```
/d/desenvolvimento/omnet/OmnetDoc/omnetpp-5.0$ make
make MODE=release
make[1]: Entering directory '/d/desenvolvimento/omnet/OmnetDoc/omnetpp-5.0'
***** Configuration: MODE=release, TOOLCHAIN_NAME=gcc, LIB_SUFFIX=.dll *****
===== Checking environment =====
===== Compiling utils =====
make[2]: Entering directory '/d/desenvolvimento/omnet/OmnetDoc/omnetpp-5.0/src/ut
ils'
Creating executable: D:/desenvolvimento/omnet/OmnetDoc/omnetpp-5.0/out/gcc-relea
se/src/utls/opp_lcg32_seedtool.exe
Creating executable: D:/desenvolvimento/omnet/OmnetDoc/omnetpp-5.0/out/gcc-relea
se/src/utls/abspath.exe
Copying scripts to bin directory...
```

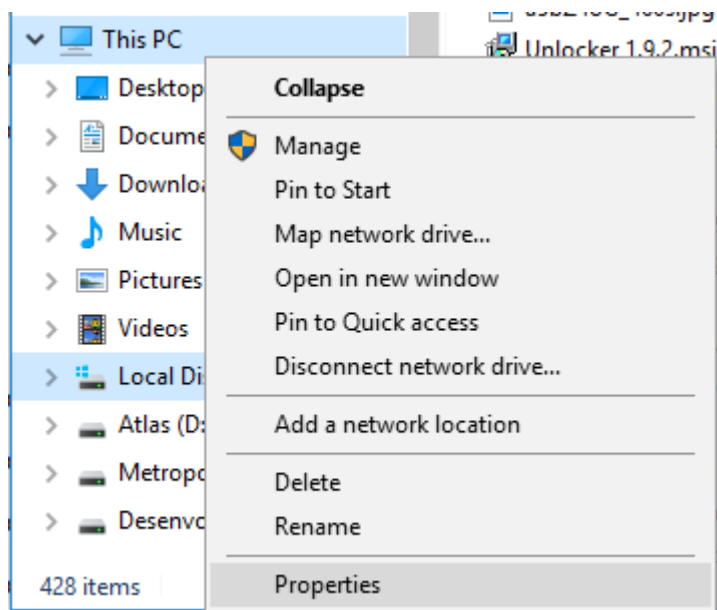
7 –Adicione o caminho até a pasta bin no PATH do sistema.

7.1 –Abra o System do Windons

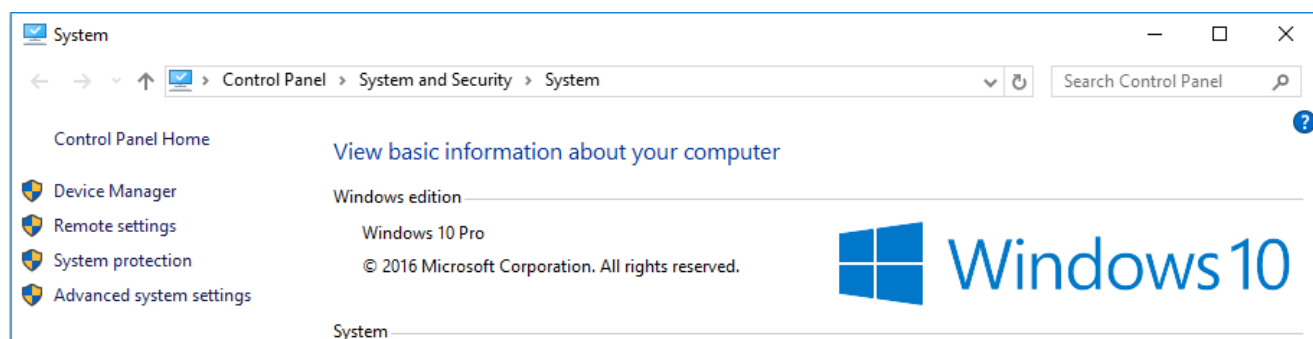
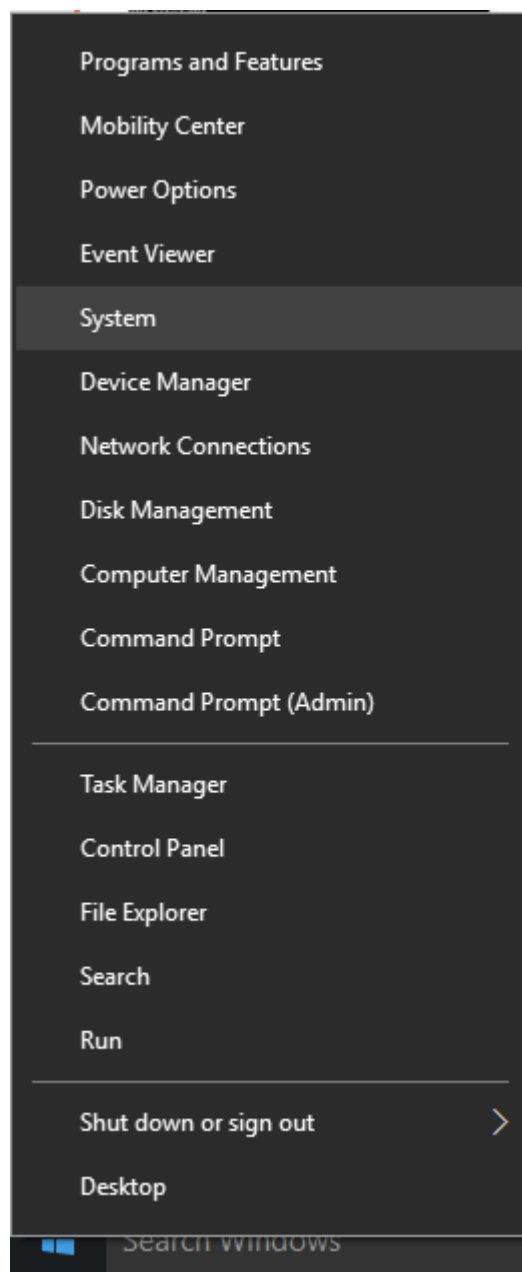
Shortcut: **Windows Logo+Break: System Properties** dialog box

7.1 –Abra o Windows System:

Pasta: Click botão direito na pasta
Meu Computador ->Propriedades

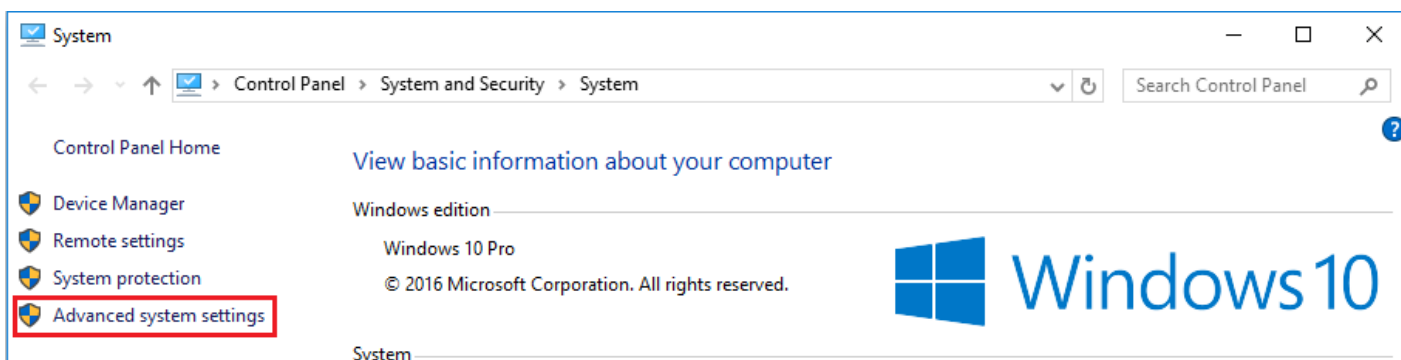


Menu Windows:

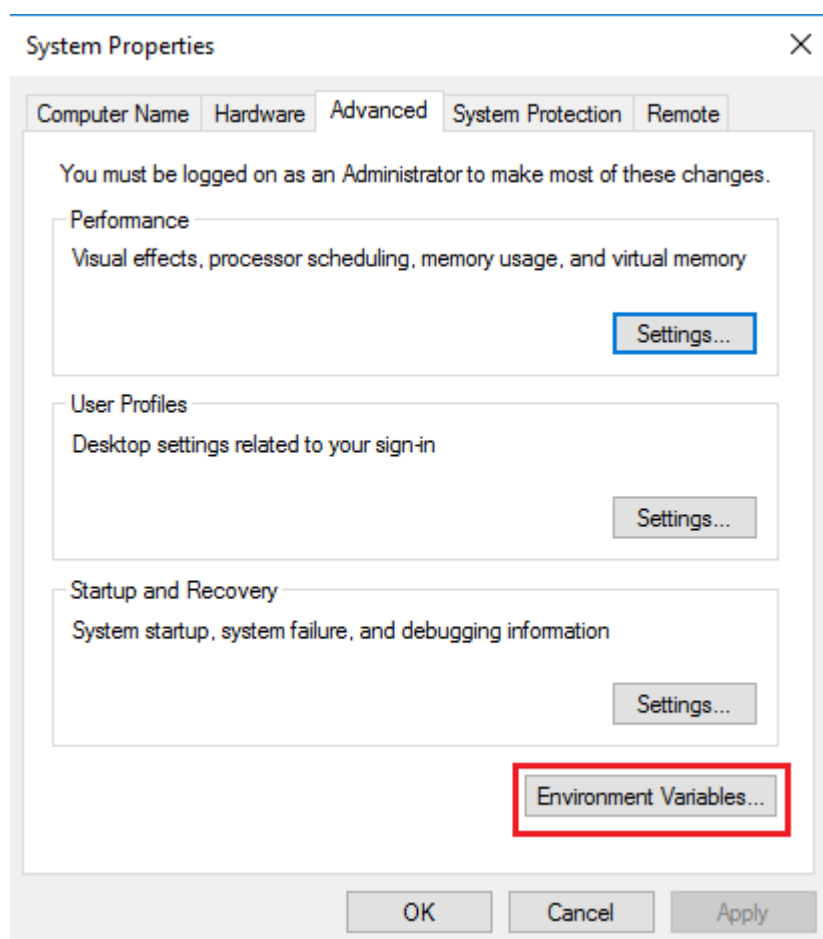


Janela Windows System

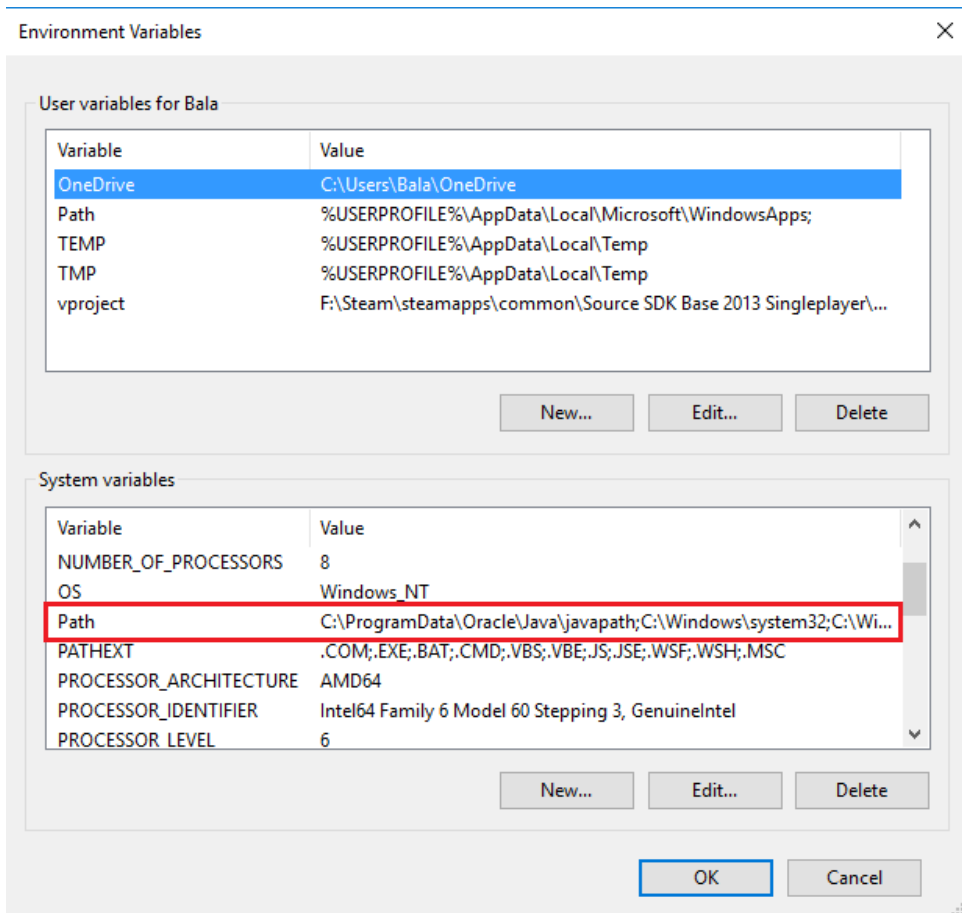
7.2 –Click no botão Configurações Avançadas do sistema.



7.3 –Click no botão Variáveis do Ambiente.



7.2 –Na aba Variáveis do Sistema, click em Path.



7.3 –Adicione os seguinte caminhos:

- bin\
- tools\win32\mingw32\bin\
- tools\win32\usr\bin
- samples\inet\src\

Os caminhos são separados por ponto-virgula(;).

Exemplo do PATH do meu PC:

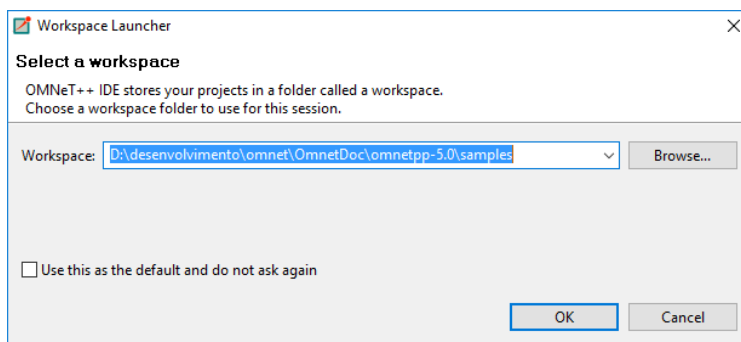
D:\desenvolvimento\omnet\omnetpp5.0\tools\win32\mingw32\bin\;D:\desenvolvimento\omnet\omnetpp5.0\tools\win32\usr\bin;D:\desenvolvimento\omnet\omnetpp5.0\samples\inet\src\;F:\IDE\Matlab\runtime\win64;F:\IDE\Matlab\bin;F:\IDE\Matlab\polyspace\bin;F:\IDE\Git\cmd;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;

8 –Digite *omnetpp* para abrir o simulador Omnet++.

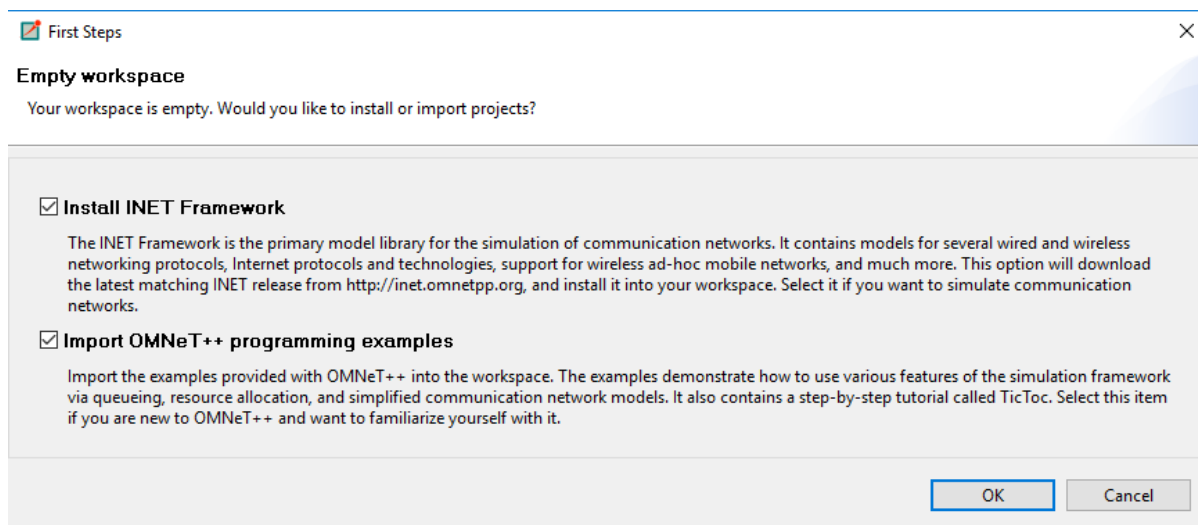
```
Now you can type "omnetpp" to start the IDE  
/d/desenvolvimento/omnet/OmnetDoc/omnetpp-5.0$ omnetpp
```



9 –Escolha um diretório para salvar seus trabalhos (workspace).



10 – Instale a biblioteca Inet. Uma janela deve aparecer automaticamente perguntado se você gostaria de baixar e instalar a biblioteca Inet ao entrar na IDE do Omnet++.



Pronto !!! O Omnet++ está Instalado. Se tiver dúvidas ou questões mais avançadas sobre a instalação, dê uma olhada no guia que vem junto com os arquivos em doc/InstallGuide.pdf

Aprendendo a usar o Omnet++

O Omnet++ já vem com vários exemplos, sugiro dar uma olhada e explorar o que está disponível. Principalmente no exemplo TicToc, que apresenta de forma gradativa um exemplo simples como um tictoc de eventos e vai incrementando com os recursos providenciados pelo simulador Omnet++ até chegar em um exemplo complexo.

Para um estudo mais aprofundado de como o Omnet++ funciona, dê uma olhada nos materiais abaixo:

Manual do Omnet++:

(doc/manual/index.html)

Ou

(<https://omnetpp.org/doc/omnetpp/manual>)

Referência da API:

(/doc/api/index.html)

Ou

(<https://omnetpp.org/doc/omnetpp/api>)

Tutorial TicToc:

<https://omnetpp.org/doc/omnetpp/tictoc-tutorial/part1.html>

<https://omnetpp.org/doc/omnetpp/tictoc-tutorial/part2.html>

<https://omnetpp.org/doc/omnetpp/tictoc-tutorial/part3.html>

<https://omnetpp.org/doc/omnetpp/tictoc-tutorial/part4.html>

<https://omnetpp.org/doc/omnetpp/tictoc-tutorial/part5.html>

OverView:

<https://omnetpp.org/doc/omnetpp/>

Canal Youtube:

https://www.youtube.com/view_play_list?p=EDBBAEA836A0A89E



Código Comentado Omnet++

Estrutura básica do Código Omnet++ (Exemplo Tictoc - Txc01.cc):

```
1
2 #include <string.h> //Biblioteca de string
3 #include <omnetpp.h> //Biblioteca básica do Omnet++
4
5 using namespace omnetpp;
6 //Define que tudo a baixo está dentro da biblioteca padrão omnetpp.h
7
8 class Txc1 : public cSimpleModule
9 //Define que a classe atual(Txc1) descende de cSimpleModule, a qual é a classe
10 //básica e necessária para criar módulos na simulação.
11 {
12     //Declaração de escopo, virtual indica que as funções
13     //podem estar definidos em outro arquivo.
14     protected:
15         virtual void initialize() override;
16         virtual void handleMessage(cMessage *msg) override;
17 };
18
19 Define_Module(Txc1);
20 //Instrução Macro que vincula essa classe (seu código, funções e variáveis)
21 //a um módulo de simulação (arquivo NED). Em analogia, o arquivo NED representa
22 //as ligações (input/output) e variáveis dinâmicas ou parametrizadas do módulo,
23 //enquanto o arquivo CC representa o comportamento e processamento do módulo
24 //na simulação pelas suas funções.
25
26 //Cada módulo simples, que descende de cSimpleModule possui 4 métodos básico:
27 //initialize,handleMessage,activity(não usada) e finish.
28
29 void Txc1::initialize()
30 {
31     //Antes de a simulação começar, o método initialize de todos os módulos é
32     //chamado para iniciar as variáveis e fazer o pré-processamento.
33
34     // Am I Tic or Toc?
35     if (strcmp("tic", getName()) == 0) {
36         //Verifica se o nome dado ao módulo na simulação é tic, e se for,
37         //criar uma mensagem nomeada tictocMsg e envia para a saída out.
38         //No arquivo NED está declarado que a saída out está conectada
39         //a entrada in do módulo toc.
40
41         cMessage *msg = new cMessage("tictocMsg");
42         //Todas a mensagens enviadas pertencem a classe de mensagens
43         //denominda cMessage.
44
45         send(msg, "out");
46         //Envia mensagem. cSimpleModule.send(Mensagem, Porta).
47     }
48 }
49
50 void Txc1::handleMessage(cMessage *msg)
51 {
52     //Quando o módulo recebe uma mensagem, este método é involcado para
53     //trata-la. Neste exemplo a mensagem simplesmente é enviada de volta
54     //para o outro módulo.
55     send(msg, "out");
56 }
57
58 void Txc1::finish() {
59     //Quando a simulação acaba ou quando o módulo é terminado, está função
60     //denominada finish é chamada para fazer o pós-processamento e limpar
61     //principalmente as mensagens pendentes do módulo.
62 }
63
```

Instalando o Simbo

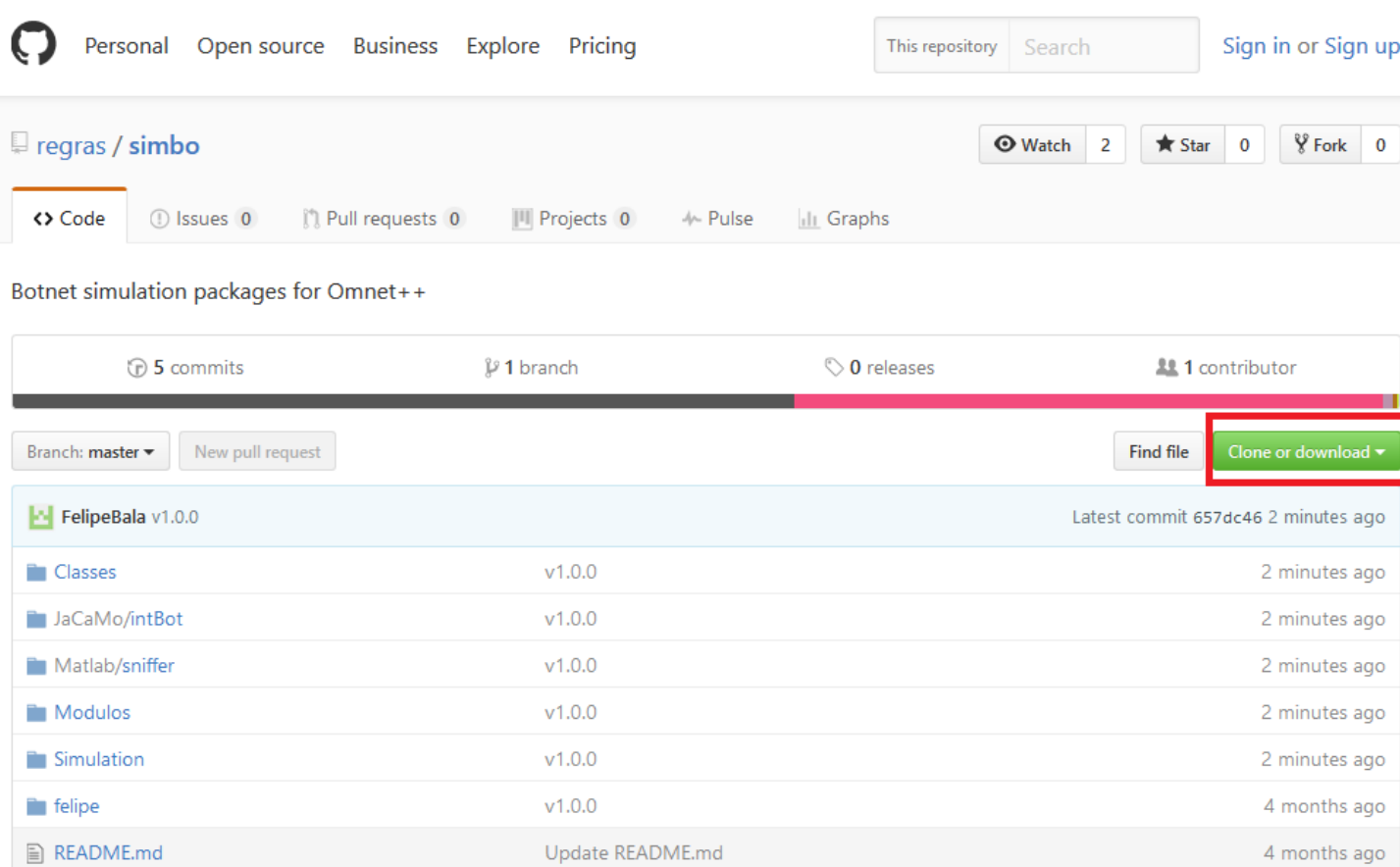
Para instalar o Simbo é simples, considerando que o omnet++ e o Inet estejam instalados basta seguir o passo a passo a baixo:

! Passo a Passo:

1 – Acesse o endereço:

<https://github.com/regras/simbo/>

2 – Click no botão “Clone or Download”



Personal Open source Business Explore Pricing This repository Search Sign in or Sign up

regras / simbo Watch 2 Star 0 Fork 0

<> Code Issues 0 Pull requests 0 Projects 0 Pulse Graphs

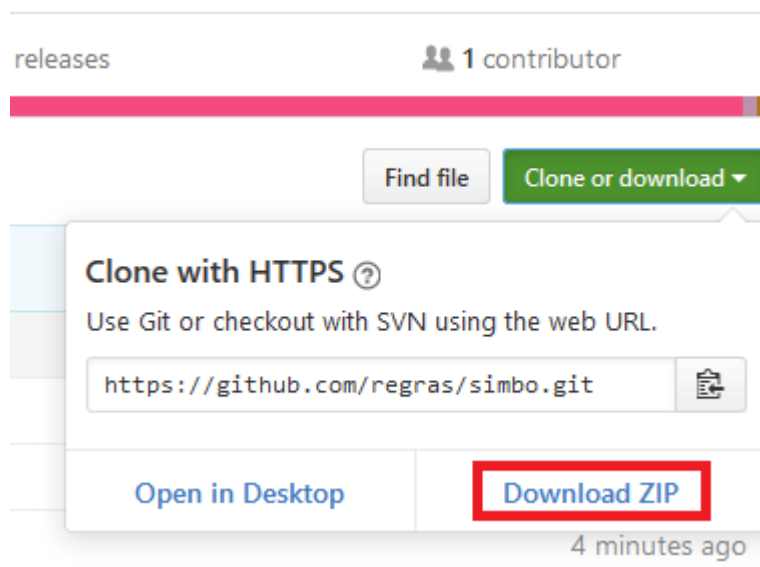
Botnet simulation packages for Omnet++

5 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Find file Clone or download

FelipeBala v1.0.0		Latest commit 657dc46 2 minutes ago
Classes	v1.0.0	2 minutes ago
JaCaMo/intBot	v1.0.0	2 minutes ago
Matlab/sniffer	v1.0.0	2 minutes ago
Modulos	v1.0.0	2 minutes ago
Simulation	v1.0.0	2 minutes ago
felipe	v1.0.0	4 months ago
README.md	Update README.md	4 months ago

2.1 – Caso queira fazer o download, click em “Download Zip”

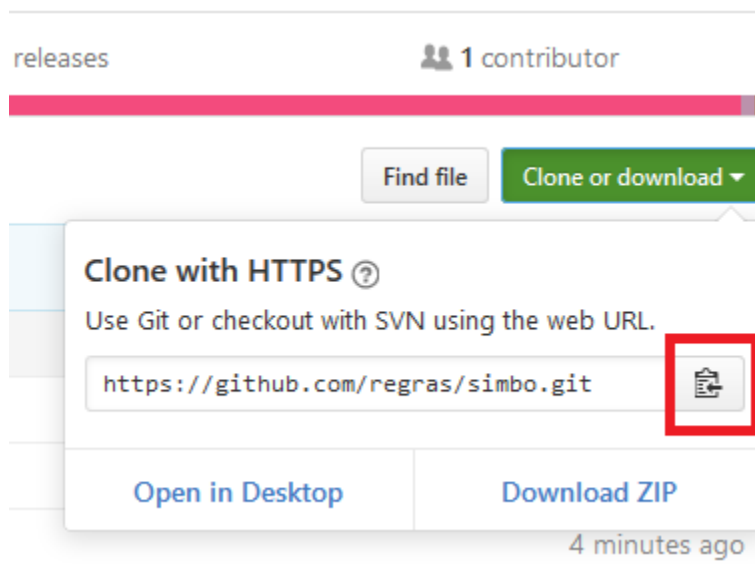


Extraia os arquivos em uma pasta chamada Simbo e mova a pasta para `\samples\inet\src\inet\applications`.

No meu computador a pasta Simbo encontra-se em:

`D:\desenvolvimento\omnet\omnetpp5.0\samples\inet\src\inet\applications\simbo`

2.2 – Caso queira fazer o clone por git, click no ícone de copiar para copiar o endereço do repositório.



Caminhe até a pasta `\samples\inet\src\inet\applications`.

No meu computador a pasta Simbo encontra-se em:

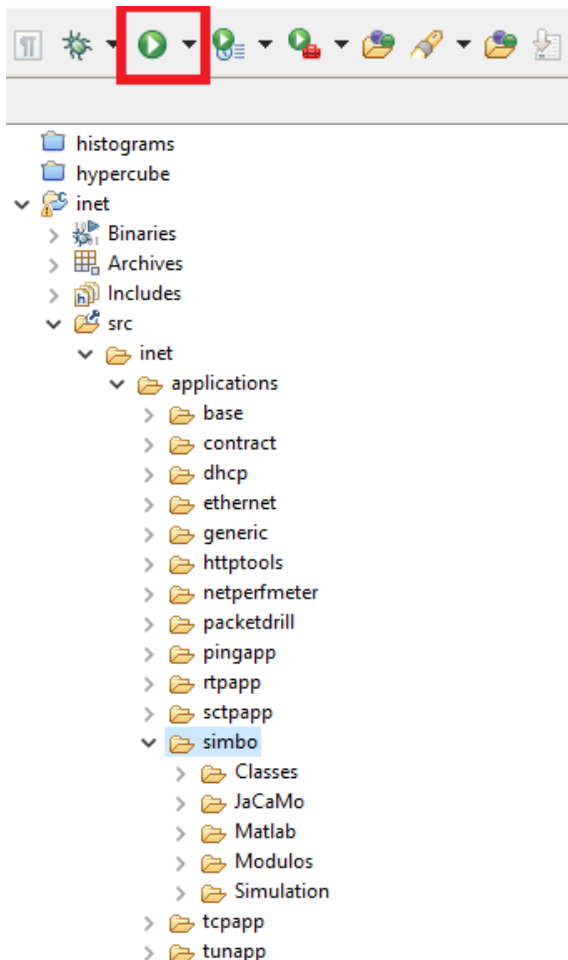
`D:\desenvolvimento\omnet\omnetpp5.0\samples\inet\src\inet\applications`

Abra o terminal git (shift+botão direito) e digite:

`git clone https://github.com/regras/simbo.git`

```
Bala@DESKTOP-QRP002V MINGW64 /f/omnet++backup/GitHub
$ git clone https://github.com/regras/simbo.git
Cloning into 'simbo'...
remote: Counting objects: 125, done.
remote: Total 125 (delta 0), reused 0 (delta 0), pack-reused 125
Receiving objects: 100% (125/125), 174.54 KiB | 140.00 KiB/s, done.
Resolving deltas: 100% (27/27), done.
```

3 – No IDE do Omnet, vá até a pasta `simbo` e depois click em run para compilar os arquivos e executar.



E pronto, o Simbo já está instalado !!!

Simbo Aprendendo a Usar

O framework Simbo encontra-se como último framework na pilha de execuções do Omnet, conforme ilustrado abaixo:

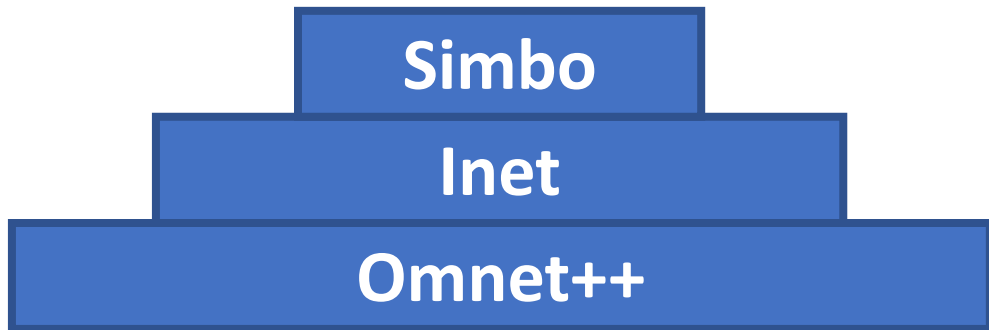
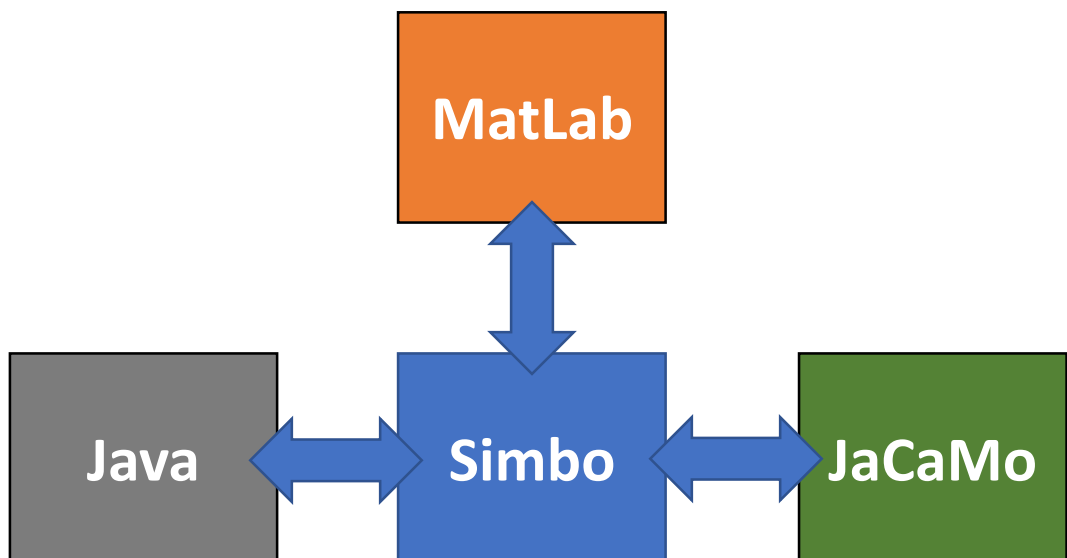


Figura: Pilha de frameworks necessários para utilizar o Simbo.

E foi desenvolvido para se comunicar com múltiplas ferramentas e programas.



Para se comunicar com os outros programas o Simbo utiliza-se de 4 arquivos. Dois de controle e dois para a transferencia de dados e comandos. A troca de informações e a simulação respectiva é feita em rodadas/turnos.

Os 4 arquivos são:

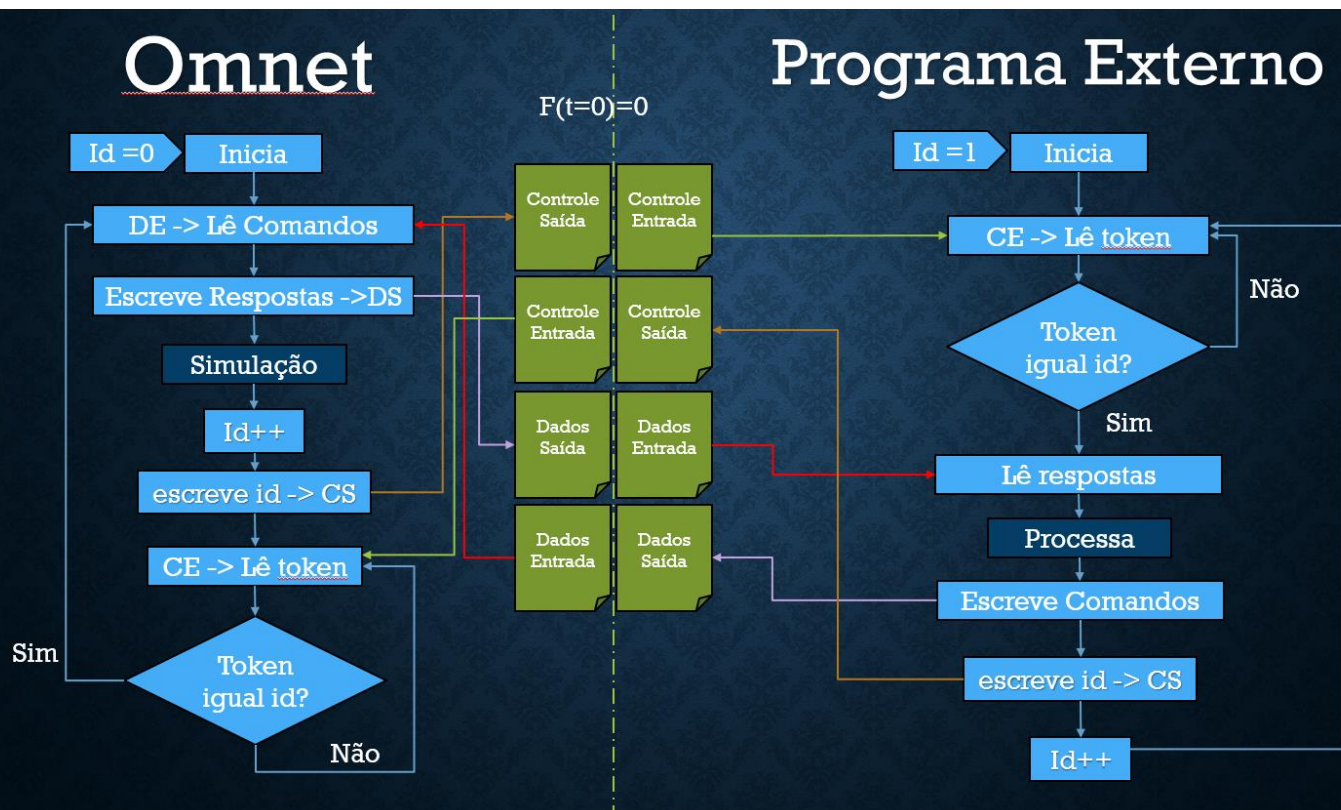
cccontrol.fl- Simbo escreve Token de Controle.

ccontrol.fl- Simbo lê controle Token de Controle.

datainput.fl - Simbo lê dados.

dataoutput.fl- Simbo escreve dados.

Abaixo é mostrado o protocolo de comunicação entre o Omnet/Simbo e outro programa externo.



Para se controlar o Simbo externamente, basta implementar o protocolo em qualquer linguagem ou ambiente.

Isso pode ser feito manualmente editando-se os arquivos de comunicação no editor de texto.

Ou utilizando-se o arquivo `automatlab.fl` que oferece uma pseudo-linguagem para se escrever um script com uma sequencia de eventos. Abaixo um exemplo de script:

Tinfecta inicial	//Infecta o computador cujo nome é inicial
Ipbotmaster	//retorna o ip do botmaster.
##	//Passa para o próximo turno (executa comandos acima).
Tinfecta comp[0]	//Infecta o computador cujo nome é comp[0].
##	//Passa turno e o simulador irá executar os comandos.
Finish	//Encerra a simulação.

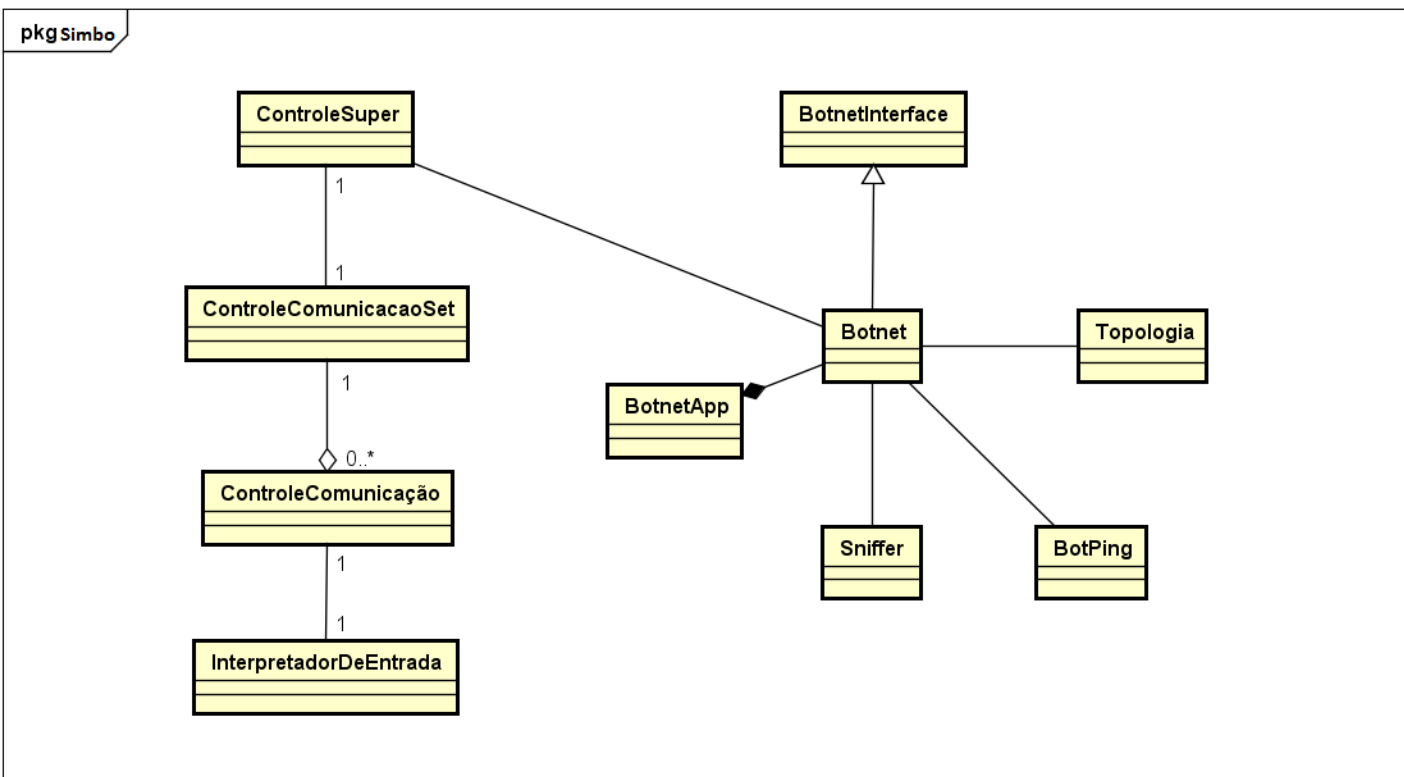
Lista de comandos implementados:

- Tinfecta:
- Ipbotmaster: Retorna o IP do botmaster.

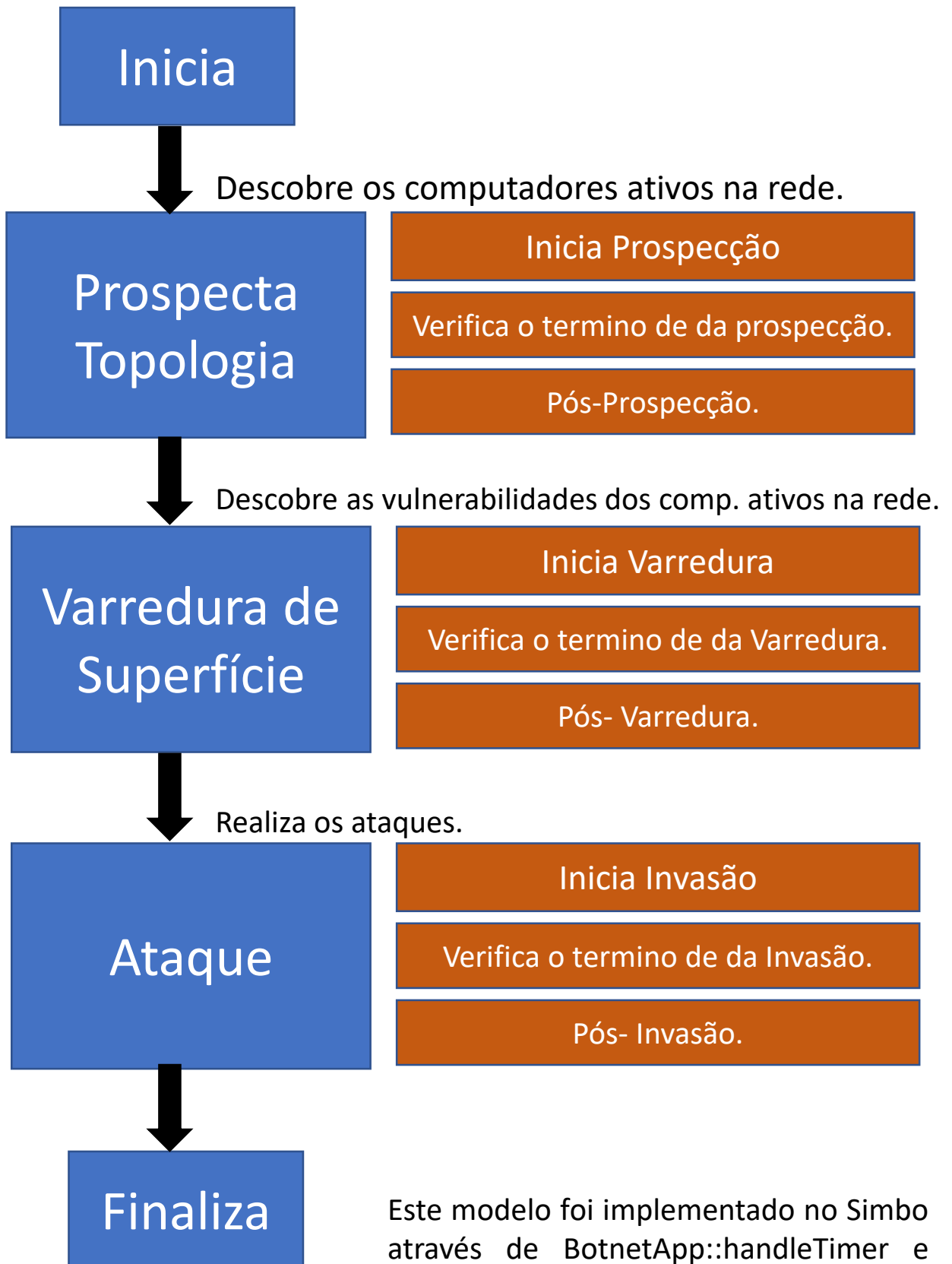
Mas também oferecemos uma implementação do protocolo para o Matlab. Permitindo que o Matlab receba as informações, analise e emita novos comandos para o simulador.

Estrutura do Simbo

Simbo é constituído das seguinte principais classes:

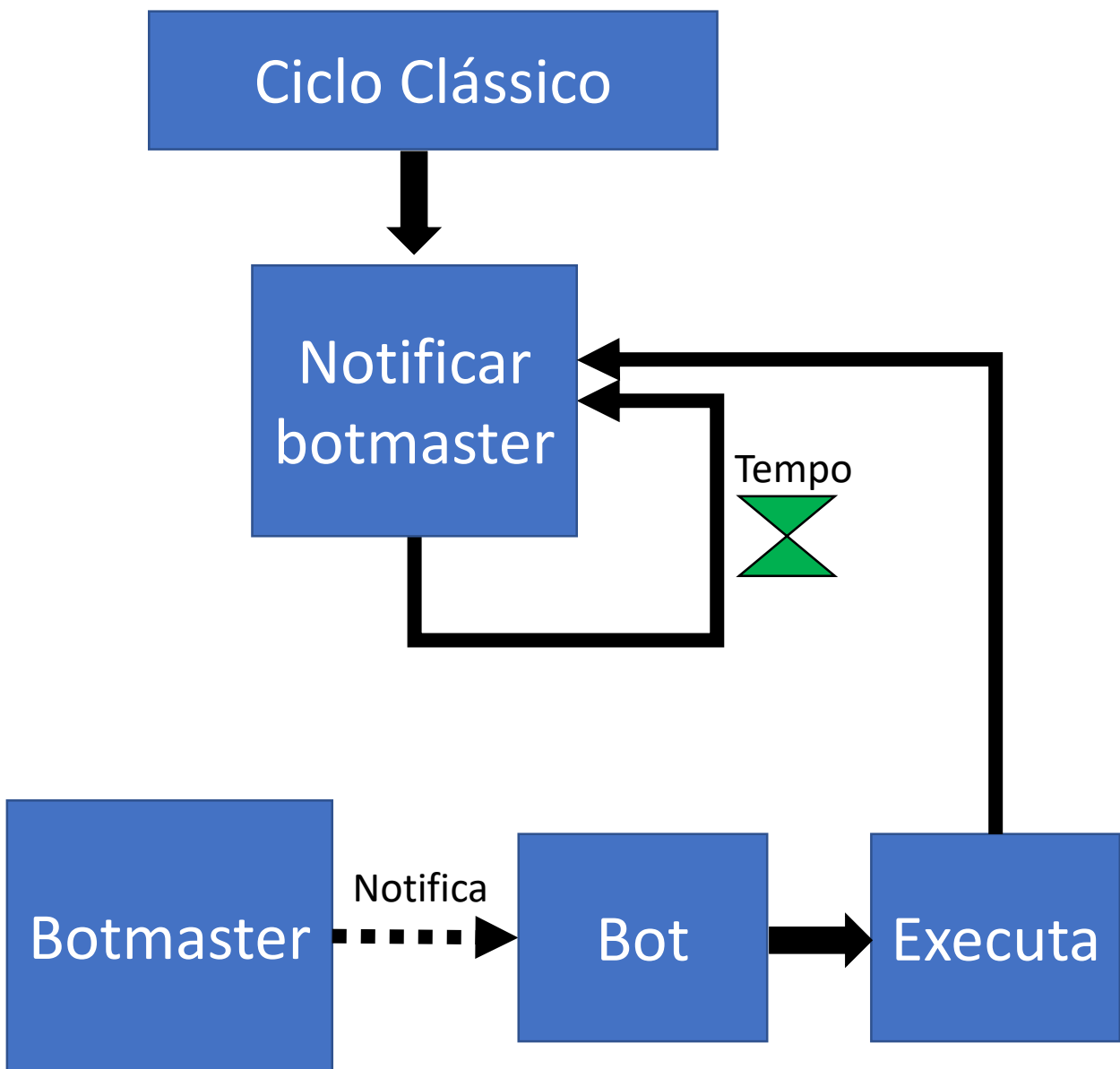


Ciclo clássico de operação do bot.



Pós Ciclo Clássico

Após o ciclo clássico ser realizado pelo bot. Esse entra em operação cíclica (loop) de avisar o botmaster de sua existência e executar os comandos enviados pelo botmaster.



Anexo I - Git

Para facilitar o versionamento (controle do conteúdo de cada versão do projeto), recomenda-se a usar o git. Não apenas porque o projeto público está armazenado no github (<https://github.com/regras/simbo/>), mas também porque o versionamento permite desenvolver o projeto sem medo de perder ou estragar o trabalho já feito. Além de proteger nas nuvens um backup que preveni contra a perda do computador ou seus dados.

Existem vários provedores Git na internet, eu recomendo o **github** para trabalhos públicos e o **bitbucket** para trabalhos privados.

Se você não tiver experiência ao lidar com Git, sugiro que dê uma olhada no material a baixo:

Tutorial Fácil de Git:

(http://rogerdudler.github.io/git-guide/index.pt_BR.html)

Referência Git para dúvidas mais complexas:

(<https://git-scm.com/book/pt-br/v1/Primeiros-passos-No%C3%A7%C3%B5es-B%C3%A1sicas-de-Git>)