



RC-I 2024-2025	TRABALHO DE LABORATÓRIO	Número:	1
PREPARAÇÃO DO AMBIENTE DE LABORATÓRIO		Data:	14/10/2025
Emulação de Rede com Mininet e Ferramentas de Administração		Prazo:	18/10/2025

1. Configuração do Ambiente

Essa secção descreve a configuração mínima do ambiente de experiência para os sistemas Windows, macOS e Linux.

1.1. Microsoft Windows

O gestor de pacote recomendado para o Microsoft Windows é o Chocolatey¹, similar ao Linux apt-get ou yum. Chocolatey foi projectado para ser uma estrutura descentralizada para instalação rápida de aplicações e ferramentas do Windows. É construído na infraestrutura NuGet, actualmente utiliza o Windows PowerShell.

Em sua máquina, é recomendável que crie um directório (pasta) para os seus projectos, chamado por exemplo RC1Projectos (é preciso assegurar que há espaço nome ou PATH):

```
C:\> mkdir C:\Users\userid\Desktop\ RC1Projectos
```

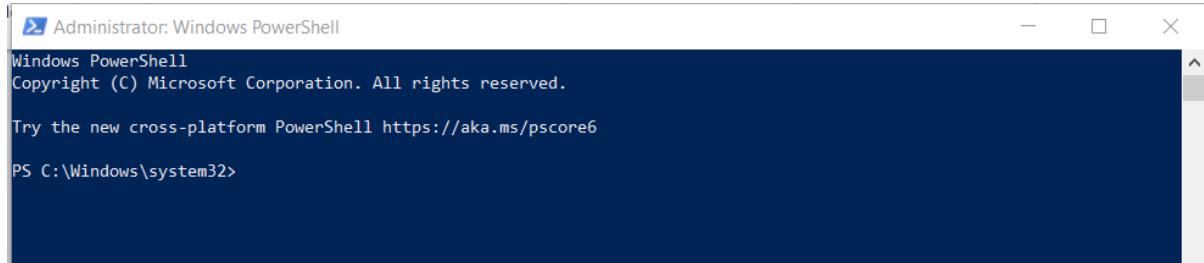
Para instalar o Chocolatey, abra em modo privilegiado (ou seja, administrativo) um Prompt de Comando do Windows (cmd.exe) e cole a sequência de texto exemplificado (no prompt de comando) e pressione Enter. Consulte o site do Chocolatey para as opções de instalação disponíveis: <https://chocolatey.org/install>.

The screenshot shows a web browser window with the URL chocolatey.org/install. The page has a purple header bar with the text "Don't miss a byte of Chocolatey goodness! Explore our livestream events now!". Below the header, there's a logo for "Chocolatey" and navigation links for Main, Community, Docs, Blog, Install, About, Product, Connect, and Try It Now. The main content area features a large heading "Installing Chocolatey". Below it, a callout box says "Step 1: Subscribe to the Chocolatey Newsletter (Optional)" with a note: "Be the first to know about upcoming features, security releases, and news about Chocolatey." There's a text input field for "Your e-mail address" and a blue "Subscribe" button. At the bottom right of the page, there's a link "Ativar o Windows".

¹ Site: <https://chocolatey.org/>



Faremos a instalação usando o PowerShell (abra em modo privilegiado):



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

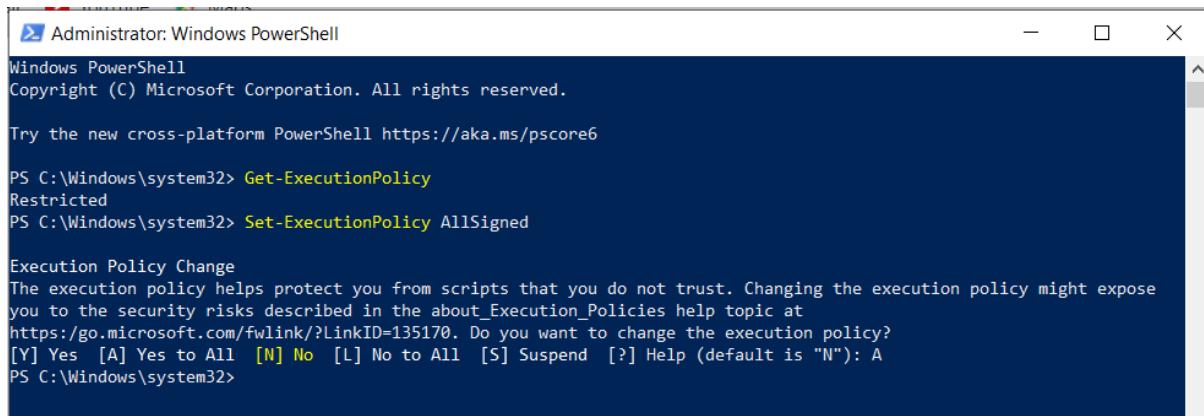
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32>
```

Figura 1: Terminal do PowerShell.

Com o PowerShell, deve garantir que Get-ExecutionPolicy não seja Restricted. Sugerimos usar Bypass para ignorar a política para instalação das ferramentas requeridas ou AllSigned para um pouco mais de segurança.

Execute o comando `Get-ExecutionPolicy`. Se retornar `Restricted`, execute `Set-ExecutionPolicy AllSigned` ou `Set-ExecutionPolicy Bypass -Scope Process`.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Get-ExecutionPolicy
Restricted
PS C:\Windows\system32> Set-ExecutionPolicy AllSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A
PS C:\Windows\system32>
```

Copie o script de instalação no site oficial (ou pode copiar no guia):

```
Set-ExecutionPolicy Bypass -Scope Process -Force;
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex
((New-Object
System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
```



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
Forcing web requests to allow TLS v1.2 (Required for requests to Chocolatey.org)
Getting latest version of the Chocolatey package for download.
```

Para uma melhor alternativa de prompt de comando para Windows, instale o ConsoleZ, um programa que fornece muitos recursos úteis, como guias, alterações de layout, suporte para digitação limpa do Windows, atalhos de teclado e muitos outros.

C:\> choco install consoleZ

```
Administrator: Windows PowerShell
PS C:\Windows\system32> choco install consoleZ
Chocolatey v2.3.0
Installing the following packages:
consoleZ
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading ConsoleZ 1.19.0.19104... 100%
ConsoleZ v1.19.0.19104 [Approved]
ConsoleZ package files install completed. Performing other installation steps.
The package ConsoleZ wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N)o/[P]rint): A
```

Após a instalação, inicie o ConsoleZ chamando 'console' no prompt de comando:

```
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N)o/[P]rint): A

Downloading ConsoleZ 64 bit
from 'https://github.com/cbucher/console/releases/download/1.19.0/ConsoleZ.x64.1.19.0.19104.zip'
Progress: 100% - Completed download of C:\Users\joaojdacosta\AppData\Local\Temp\chocolatey\ConsoleZ\1.19.0.19104\ConsoleZ.x64.1.19.0.19104.zip (2.72 MB).
Download of ConsoleZ.x64.1.19.0.19104.zip (2.72 MB) completed.
Hashes match.
Extracting C:\Users\joaojdacosta\AppData\Local\Temp\chocolatey\ConsoleZ\1.19.0.19104\ConsoleZ.x64.1.19.0.19104.zip to C:\ProgramData\chocolatey\lib\ConsoleZ\tools...
C:\ProgramData\chocolatey\lib\ConsoleZ\tools
ShimGen has successfully created a gui shim for Console.exe
The install of ConsoleZ was successful.
Deployed to 'C:\ProgramData\chocolatey\lib\ConsoleZ\tools'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Windows\system32> console
PS C:\Windows\system32>
```

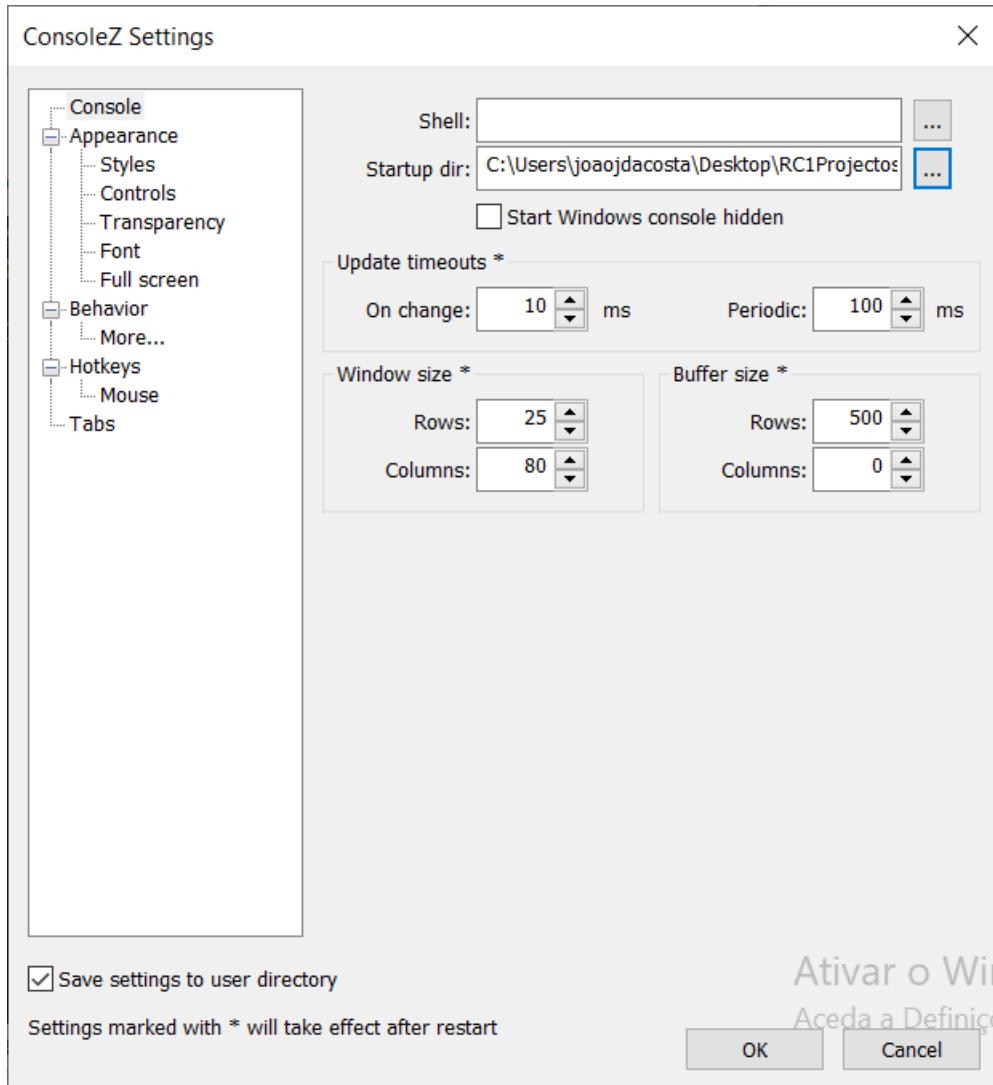


Microsoft Windows [Version 10.0.19045.4894]
(c) Microsoft Corporation. Todos os direitos reservados.
C:\Windows\system32>

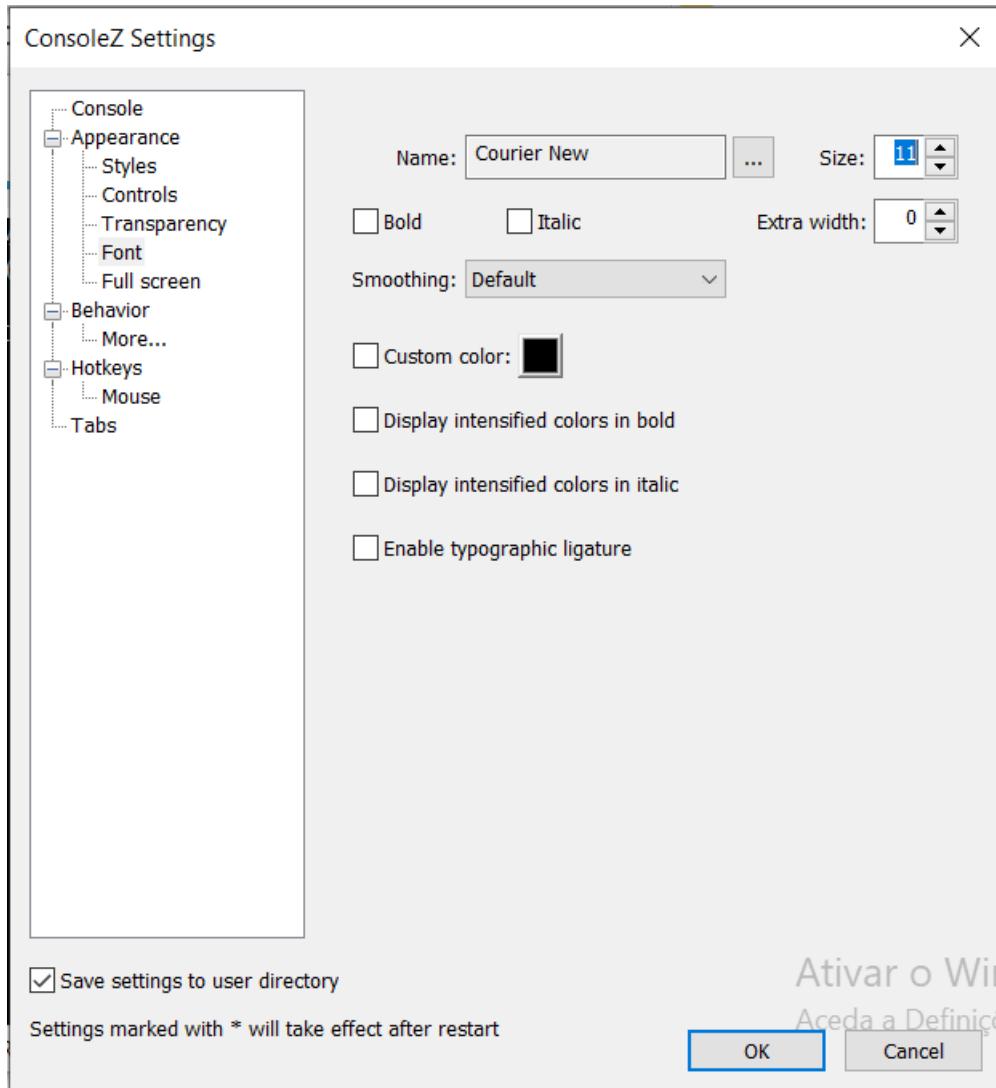
Opcionalmente, o ConsoleZ pode ser 'fixado' como um ícone na barra de tarefas do Windows.



No menu do Console, selecione “Edit” – ‘Settings...’ e, em “Console”, defina seu diretório de inicialização padrão (“Startup dir”), por exemplo,
C:\Users\Admin\Desktop\RC1Projectos.



Em “**Appearance**”, defina a fonte como Consolas 11, fundo preto.



Em “**Behavior**”, defina “**Copy on select**”. Em “**Hotkeys**”, altere a tecla de atalho “**New Tab 1**” para **Ctrl-T**, ‘Copiar seleção’ para **Ctrl-C** e ‘Colar’ para **Ctrl-V**. Em ‘Tabs’, verifique se o ‘Main’ está definido como PowerShell e se a caixa de seleção executar como ‘administrador’ está selecionada

Agora é hora de instalar os outros pacotes para o ambiente de experiência do laboratório. Inicie um Console e instale os seguintes programas usando Chocolatey:

```
C:\> choco install wget
C:\> choco install git -params " /GitAndUnixToolsOnPath "
C:\> choco install virtualbox
C:\> choco install vagrant
C:\> choco install xming
C:\> choco install atom
```



```
C:\Users\joaojdacosta\Desktop\RC1Projectos>choco install wget
Chocolatey v2.3.0
Installing the following packages:
wget
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading Wget 1.21.4... 100%
Wget v1.21.4 [Approved]
Wget package files install completed. Performing other installation steps.
The package Wget wants to run 'chocolateyinstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N)o/[P]rint): A

Getting x64 bit zip
Extracting C:\ProgramData\chocolatey\lib\Wget\tools\wget-1.21.4-win64_x64.zip to
C:\ProgramData\chocolatey\lib\Wget\tools...
C:\ProgramData\chocolatey\lib\Wget\tools
ShimGen has successfully created a shim for wget.exe
The install of Wget was successful.
Deployed to 'C:\ProgramData\chocolatey\lib\Wget\tools'

Chocolatey installed 1/1 packages.

Ready | 1956 | 80x25 | 80x500 | 100%
```

Execute os comandos restantes.

Depois deste ponto, e para resolver tudo (o Windows é bem complicado em atualizar variáveis de ambiente), feche o Console e reinicie-o novamente. Além disso, como um cliente SSH (secure shell) não é distribuído com o Windows, é necessário informar ao sistema onde encontrar o cliente SSH instalado com o GIT:

- Abra o Painel de Controlo
- Vá para Sistema e Segurança
- Clique em Sistema e depois no botão Alterar Configurações
- Exiba a guia Avançado e clique em Variáveis de Ambiente...
- Procure a variável Path na lista Variáveis do Sistema, selecione-a e depois Editar...

No final da string, confirme se existe o seguinte caminho, caso contrário adicione o seguinte caminho apontando para a pasta bin do Git:

```
C:\Program Files\Git\bin\;C:\Program Files\Git\usr\bin
```

Para que esses programas funcionem correctamente no Windows, algumas variáveis de ambiente precisam ser definidas. No seu prompt de comando, abra um Applet do Sistema para o Painel de Controle:

```
C:\> sysdm.cpl
```



Uma janela System Properties será aberta (Figura 1). Selecione a TAB chamada “Advanced” e clique em “Environment Variables...”.

Na janela que se abre, crie uma variável de sistema “New...” com o nome DISPLAY e com o valor da variável localhost:0.0, depois guarde.

Para o servidor XWindows, inicie o XMing e um ícone “X” do servidor estará visível na bandeja do sistema da Barra de Tarefas. Passe o mouse sobre ele e ele deverá exibir algo como `Xming server - 0:0`.

1.2. Apple macOS

O Gestor de Pacotes recomendado para macOS é o **Homebrew** (<http://brew.sh>), similar ao apt-get ou yum do Linux. O Homebrew instala pacotes (binários ou para serem compilados) em seu próprio diretório e então faz um link simbólico de seus executáveis em /usr/local. O Homebrew fornece o Homebrew-Cask, implementado como um comando externo do homebrew chamado cask. O Homebrew-Cask estende o Homebrew e traz sua elegância, simplicidade e velocidade para a instalação e gestão de aplicações macOS GUI, como o Google Chrome.

Como primeiro passo, pode precisar instalar o **Command Line Tools for XCode** da Apple (grátis na App Store), para que o Homebrew consiga compilar aplicações a partir do código-fonte. Os Macs não têm nenhuma das 'Command Line Tools' do desenvolvedor instaladas por padrão, então precisamos instalá-las antes de podermos chegar a algum lugar. Se não tiver o XCode instalado, abra o Terminal e no prompt do shell digite o seguinte:

```
:~$ xcode-select -install
```

Para instalar o Homebrew, no prompt do shell cole o script (começando na invocação do ruby) da caixa abaixo e pressione enter. Referência: <http://brew.sh>

```
:~$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

O prompt do shell dirá o que está prestes a fazer e perguntará se deseja prosseguir: pressione Enter para fazê-lo. O prompt do shell pode então pedir uma senha: esta é a senha para a conta Admin no seu computador. Digite sua senha e pressione Enter. Quando terminar, o prompt do shell dirá que a instalação foi bem-sucedida e pedirá que execute `brew doctor`. Faça como sugere:

```
:~$ brew doctor
```

Isso fará com que o Homebrew inspeccione seu sistema e certifique-se de que tudo esteja configurado correctamente. Se o prompt do shell informar sobre algum problema, precisará corrigi-lo e, em seguida, executar o `brew doctor` novamente para verificar se tudo foi corrigido correctamente. Quando tudo estiver configurado correctamente, verá a mensagem "Your system is ready to brew", e poderá prosseguir.



Agora é hora de instalar os outros pacotes para o ambiente de experiências de laboratório. Abra o Terminal e instale os seguintes programas usando o Homebrew:

```
:~$ brew install wget  
:~$ brew install git  
:~$ brew cask install virtualbox  
:~$ brew cask install vagrant  
:~$ brew cask install xquartz  
:~$ brew cask install atom
```

Para cada comando, o Terminal começará a apresentar muitas informações, pois o Homebrew está a lhe atualizar sobre o que está a fazer. Esse fluxo de informações será um guia para que saiba se o computador ainda está a executar ou não e, portanto, não o interrompa.

Agora é hora de criar um directório/pasta para seus projectos de laboratório, por exemplo:

```
:~$ mkdir ~/Desktop/RC1Projectos
```

1.3. Linux

Para um sistema Linux, nomeadamente Ubuntu, o gestor de pacotes padrão 'apt-get' já está presente, assim como alguns pacotes, como 'wget'.

Para instalar os outros pacotes, abra um Terminal e execute os seguintes comandos (responda Y aos prompts e várias dependências também serão instaladas).

```
:~$ sudo apt-get install git
```

Observe que para as distribuições mais recentes do Ubuntu, uma instalação cuidadosa do virtualbox é necessária, pois a arquitectura do pacote precisa corresponder à arquitetura do kernel do Linux. Além disso, o pacote VirtualBox no Ubuntu pode ter alguns problemas ao ser executado em ambientes com restrição de RAM.

Para isso, precisa garantir que tenha as fontes adequadas referenciadas para sua arquitectura de kernel.

Comece por editar a lista de fontes, execute o seguinte comando:

```
:~$ sudo nano /etc/apt/sources.list
```

Adicione a seguinte linha à lista (exemplificada para o kernel 16.04 'Xenial'):

```
deb http://download.virtualbox.org/virtualbox/debian xenial contrib
```

De acordo com sua distribuição, substitua 'xenial' por 'vivid', 'utopic', 'trusty', etc. também é necessário adicionar e registrar a chave pública do Oracle (comando combinado):

```
:~$ wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc -O- | sudo apt-key add -
```



```
:~$ wget -q https://www.virtualbox.org/download/oracle_vbox.asc -O -
| sudo apt-key add -
```

Agora pode adicionar o Virtualbox:

```
:~$ sudo apt-get update
:~$ sudo apt-get install virtualbox-5.2
```

Para o Vagrant ainda há um problema e precisamos obter a versão mais recente (não no repositório do Ubuntu) baixando o pacote do site do Vagrant <http://downloads.vagrantup.com/> e então emitindo o seguinte comando (da pasta de downloads, adapte à sua versão):

```
:~$ wget
https://releases.hashicorp.com/vagrant/2.1.5/vagrant_2.1.5_x86_64.de
b
:~$ dpkg -i vagrant_2.1.5_x86_64.deb
```

Para um sistema Linux, nomeadamente Ubuntu, o sistema XWindows normalmente já está instalado. Pode precisar apenas instalar o editor Atom.

```
:~$ sudo apt-get update
:~$ curl -sL https://packagecloud.io/AtomEditor/atom/gpgkey | sudo
apt-key add -
:~$ sudo sh -c 'echo "deb [arch = amd64]
https://packagecloud.io/AtomEditor/atom/any/ any main" >
/etc/apt/sources.list.d/atom.list'
:~$ sudo apt-get update
:~$ sudo apt-get install atom
```

Reinic peace o sistema e crie um directório para seus projectos de laboratório, por exemplo:

```
:~$ mkdir ~/Desktop/RC1Projectos
```

2. Submissão dos Resultados da Experiência

Explicação do procedimento de submissão do trabalho será dada em sala de aula.

3. Primeira Experiência

A primeira experiência, válido para todos os sistemas (Windows, Mac e Linux) consistirá no lançamento de uma rede emulada usando **Mininet** (<http://mininet.org>). O Mininet cria uma rede virtual realista, executando kernel real, switch e código de aplicação, em uma única máquina (neste caso, uma Máquina Virtual).

Para esse propósito, é necessário provisionar a máquina virtual Mininet usando o Vagrant.



Antes de iniciar o provisionamento, apresentamos aqui uma etapa de optimização opcional, recomendada apenas para acelerar a implantação. Ela armazena em cache os ficheiros necessários para executar o Vagrant da rede local.

A primeira etapa é baixar o Vagrant Box do Mininet (`mininet-update.box`). Em seguida, na pasta Downloads da sua máquina, faça o seguinte:

```
:~$ vagrant box add ktr/mininet ~/Downloads/mininet-updated.box
```

```
C:\Users\LENOVO\Desktop\MyProjects>vagrant box add ktr/mininet "C:\Users\LENOVO\Downloads\mininet-updated.box"
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'ktr/mininet' (v0) for provider:
  box: Unpacking necessary files from: file:///C:/Users/LENOVO/Downloads/mininet-updated.box
  box:
==> box: Successfully added box 'ktr/mininet' (v0) for ''!
```

Crie a pasta `mininet` dentro da pasta do projecto `RC1Projectos`. Então, pode iniciar o Mininet com:

```
:~$ cd mininet
:~$ vagrant init ktr/mininet
```

```
C:\Users\LENOVO\Desktop\MyProjects>cd mininet
C:\Users\LENOVO\Desktop\MyProjects\mininet>vagrant init ktr/mininet
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
```

É gerado dentro da pasta `mininet` o ficheiro `Vagrantfile`:



The screenshot shows the Atom code editor interface. The title bar reads "Vagrantfile — C:\Users\LENOVO\Desktop\MyProjects\mininet — Atom". The menu bar includes File, Edit, View, Selection, Find, Packages, and Help. The left sidebar shows a "Project" tree with a folder named "mininet" and a file named "Vagrantfile". The main editor area displays the following Vagrantfile code:

```
1  # -*- mode: ruby -*-
2  # vi: set ft=ruby :
3
4  # All Vagrant configuration is done below. The "2" in Vagrant.configure
5  # configures the configuration version (we support older styles for
6  # backwards compatibility). Please don't change it unless you know what
7  # you're doing.
8  Vagrant.configure("2") do |config|
9    # The most common configuration options are documented and commented below.
10   # For a complete reference, please see the online documentation at
11   # https://docs.vagrantup.com.
12
13   # Every Vagrant development environment requires a box. You can search for
14   # boxes at https://vagrantcloud.com/search.
15   config.vm.box = "ktr/mininet"
16
17   # Disable automatic box update checking. If you disable this, then
18   # boxes will only be checked for updates when the user runs
19   # `vagrant box outdated`. This is not recommended.
20   # config.vm.box_check_update = false
21
22   # Create a forwarded port mapping which allows access to a specific port
23   # within the machine from a port on the host machine. In the example below,
24   # accessing "localhost:8080" will access port 80 on the guest machine.
25   # NOTE: This will enable public access to the opened port
26   # config.vm.network "forwarded_port", guest: 80, host: 8080
27
28   # Create a forwarded port mapping which allows access to a specific port
29   # within the machine from a port on the host machine and only allow access
30   # via 127.0.0.1 to disable public access
31   # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"
32
33   # Create a private network, which allows host-only access to the machine
34   # using a specific IP.
```

Agora que buscamos o sistema Mininet já preparado em um sistema operativo Ubuntu. Podemos inicializar o sistema:

```
:~$ vagrant up --provider virtualbox
```

Esta fase pode levar alguns minutos, dependendo da velocidade de ligação do sistema host. Pode receber mensagem de erro como as seguintes:



1. ConsoleZ

```
File Edit View Tabs Help
[Icons] Search... [Icons]
`vagrantup.com` for more information on using Vagrant.

C:\Users\LENOVO\Desktop\MyProjects\mininet>vagrant up --provider virtualbox
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ktr/mininet'...
==> default: Matching MAC address for NAT networking...
==> default: Setting the name of the VM: mininet_default_1728567858875_23829
==> default: Fixed port collision for 22 => 2222. Now on port 2200.
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2200 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2200
    default: SSH username: vagrant
    default: SSH auth method: private key
    default: Warning: Connection reset. Retrying...
    default: Warning: Connection aborted. Retrying...
    default: Warning: Connection reset. Retrying...
    default: Warning: Connection aborted. Retrying...
    default: Warning: Connection reset. Retrying...
    default: Warning: Connection aborted. Retrying...
    default: Warning: Connection reset. Retrying...
```

Esta mensagem de aviso/erro é resultado da versão desactualizada dos drivers do Virtualbox dentro da caixa e pode ignorá-la com segurança agora. Mostraremos como consertá-la nas próximas etapas. Em seguida, estabeleceremos uma sessão com o sistema usando o seguinte comando:

```
:~$ vagrant ssh
```

A sessão é estabelecida e receberemos o prompt da máquina semelhante ao seguinte:



1. ConsoleZ - vagrant@mininet: ~

File Edit View Tabs Help

C:\Users\LENOVO\Desktop\MyProjects\mininet>vagrant ssh
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.13.0-132-generic x86_64)

* Documentation: <https://help.ubuntu.com/>

System information as of Thu Oct 10 13:49:05 UTC 2024

System load: 0.42 Processes: 85
Usage of /: 6.0% of 39.34GB Users logged in: 0
Memory usage: 2% IP address for eth0: 10.0.2.15
Swap usage: 0% IP address for docker0: 172.17.0.1

Graph this data and manage this system at:
<https://landscape.canonical.com/>

Get cloud support with Ubuntu Advantage Cloud Guest:
<http://www.ubuntu.com/business/services/cloud>

New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Sep 20 17:15:12 2017 from 10.0.2.2
vagrant@mininet:~\$ █

Ready

Um teste muito simples pode ser realizado para verificar se tudo está a funcionar correctamente, emitindo o comando 'sudo mn' no prompt do shell do mininet, para criar uma **topologia de rede simples** de dois hosts (h1, h2) e um switch (s1):



1. ConsoleZ - vagrant@mininet: ~

```
File Edit View Tabs Help
[Icons] Search...
vagrant@mininet:~$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> 
```

Ready

A partir deste ponto podemos emitir o comando 'pingall' para verificar se temos conectividade entre os hosts:

1. ConsoleZ - vagrant@mininet: ~

```
File Edit View Tabs Help
[Icons] Search...
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> 
```

Ready

Com o comando 'net' podemos verificar como os nós estão conectados, ou seja, o host1 está conectado pela sua porta eth0 ao switch1 na porta eth1, etc.:



1. ConsoleZ - vagrant@mininet: ~

File Edit View Tabs Help

mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
mininet> █

Ready

Emitindo o comando ‘dump’ podemos ver os endereços IP dos hosts (10.0.0.1 e 10.0.0.2):

1. ConsoleZ - vagrant@mininet: ~

File Edit View Tabs Help

mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=2270>
<Host h2: h2-eth0:10.0.0.2 pid=2274>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=2279>
<Controller c0: 127.0.0.1:6653 pid=2263>
mininet> █

Ready

Para encerrar a emulação do Mininet usamos o comando ‘exit’ e também é aconselhável ‘clean’ quaisquer processos Mininet fantasma no sistema com o comando ‘sudo mn’:



1. ConsoleZ - vagrant@mininet: ~

```
File Edit View Tabs Help
[Icons] Search... [Icons]
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 937.385 seconds
vagrant@mininet:~$ sudo mn -c
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ov
null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core
ev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
*** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([-_.[:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
vagrant@mininet:~$ 
```

Ready

Agora podemos sair da sessão:



1. ConsoleZ

```
File Edit View Tabs Help
[Icons] Search...
vagrant@mininet:~$ exit
logout
Connection to 127.0.0.1 closed.

C:\Users\LENOVO\Desktop\MyProjects\mininet>
```

Ready

Para parar a Máquina Virtual e verificar o estado global de todos os ambientes Vagrant activos no sistema, podemos emitir os seguintes comandos:

1. ConsoleZ

```
File Edit View Tabs Help
[Icons] Search...
C:\Users\LENOVO\Desktop\MyProjects\mininet>vagrant halt
==> default: Attempting graceful shutdown of VM...

C:\Users\LENOVO\Desktop\MyProjects\mininet>vagrant global-status
```

Confirme se o status das VMs é 'powered off'.

Bom trabalho!