
Think functionally, think in Scala

Requerimientos de cómputo recomendados:	2
Herramientas necesarias para el taller	2
Creación de la máquina virtual	3
Detener o eliminar la máquina virtual	6



Requerimientos de cómputo recomendados:

- Cualquier sistema operativo compatible con VirtualBox, Vagrant y Git (Windows, macOS, GNU/Linux).
- Al menos 6 GB de RAM.
- Un procesador con al menos 4 núcleos físicos.
- De preferencia, abrir el entorno de desarrollo (Polynote) en un navegador basado en Chrome (Google Chrome, Microsoft Edge, Brave, etc...).

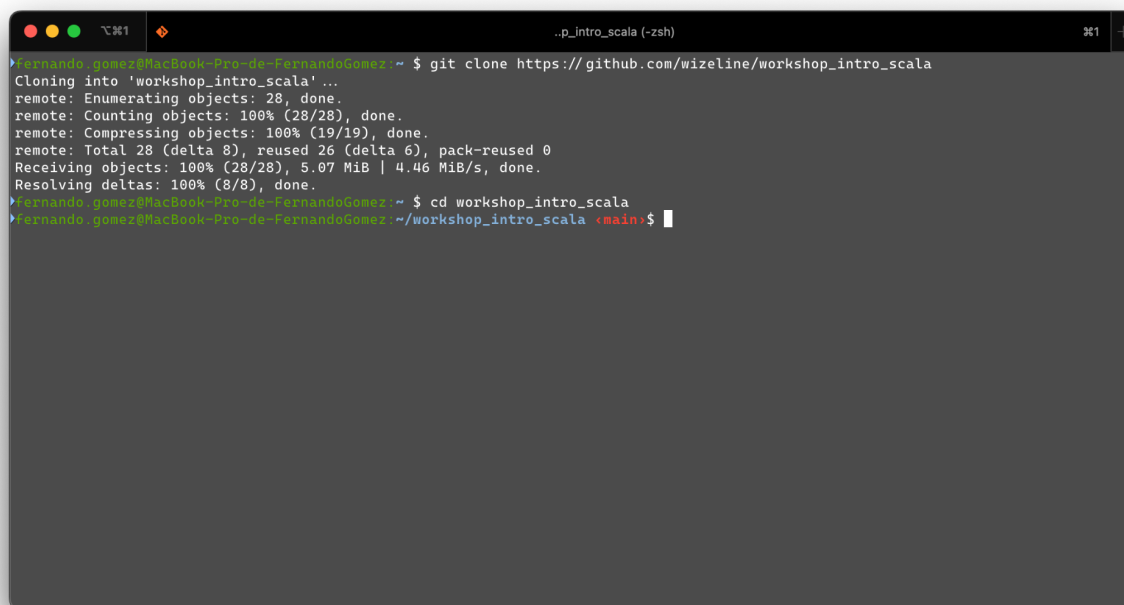
Herramientas necesarias para el taller

- Virtualbox: <https://www.virtualbox.org/wiki/Downloads>
- Vagrant: <https://www.vagrantup.com/downloads>
- Git: <https://git-scm.com/downloads>



Creación de la máquina virtual

1. Clonar el enlace del repositorio desde la terminal con
`git clone https://github.com/wizeline/workshop_intro_scala.git`
2. Moverse al directorio del repositorio clonado desde la terminal:

A terminal window titled '..p_intro_scala (-zsh)' showing the execution of two commands. The first command is 'git clone https://github.com/wizeline/workshop_intro_scala', which outputs the progress of cloning the repository. The second command is 'cd workshop_intro_scala', which changes the current directory to the newly cloned repository. The prompt changes from '~' to '~/workshop_intro_scala' after the second command.

```
fernando.gomez@MacBook-Pro-de-FernandoGomez: ~ $ git clone https://github.com/wizeline/workshop_intro_scala
Cloning into 'workshop_intro_scala' ...
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 28 (delta 8), reused 26 (delta 6), pack-reused 0
Receiving objects: 100% (28/28), 5.07 MiB | 4.46 MiB/s, done.
Resolving deltas: 100% (8/8), done.
fernando.gomez@MacBook-Pro-de-FernandoGomez: ~ $ cd workshop_intro_scala
fernando.gomez@MacBook-Pro-de-FernandoGomez: ~/workshop_intro_scala (main)$
```

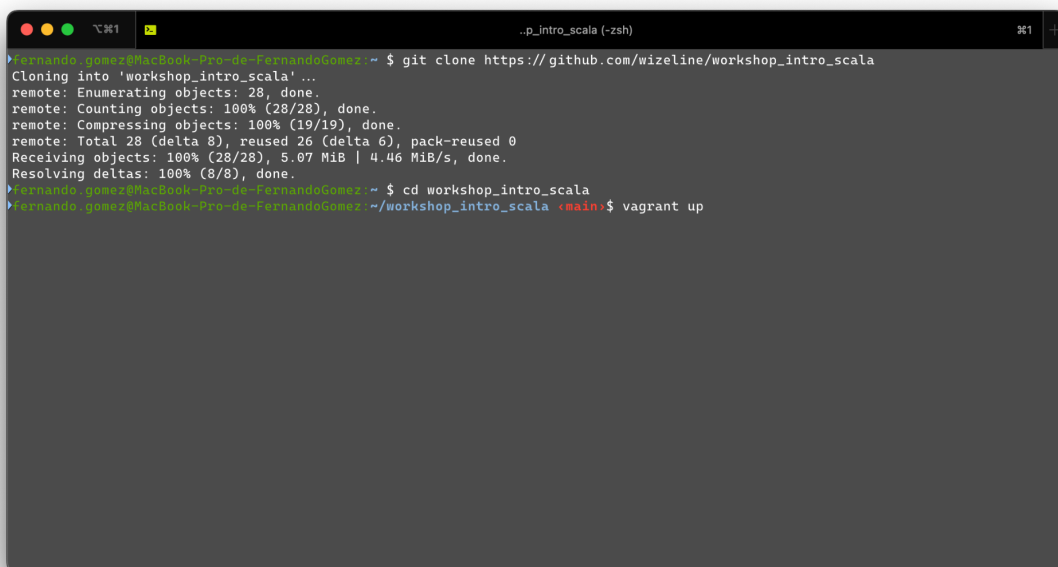


3. Ejecutar el comando “`vagrant up`”. Y esperar a que termine de crearse la máquina virtual.

Nota: Dependiendo del sistema operativo, puede que sea necesario dar permisos de administrador a VirtualBox la primera vez que se usa.

En Windows, esto sucede al momento de crear la máquina virtual.

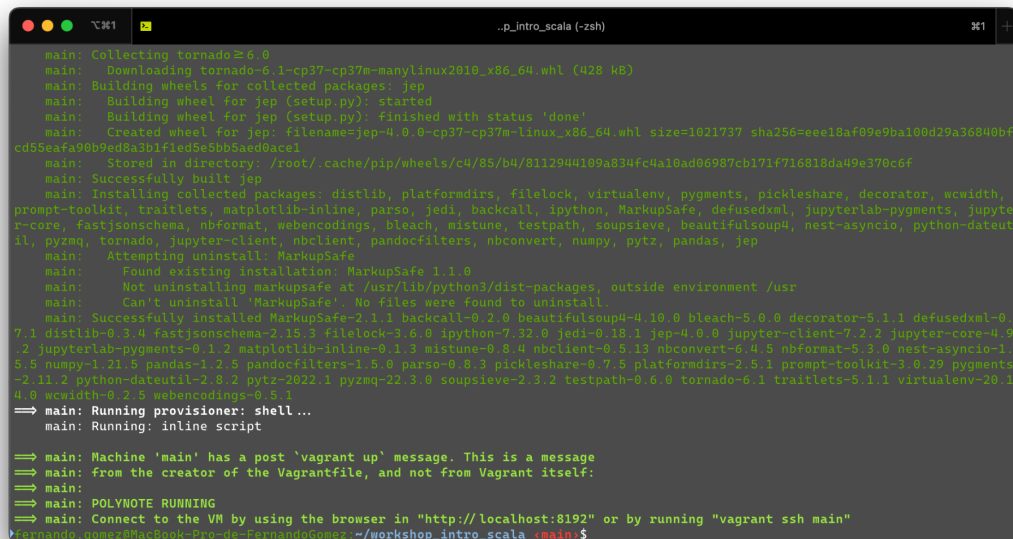
En macOS, esto se configura en un mensaje dentro de la opción “Seguridad y Privacidad” de las Preferencias del Sistema.



```
fernando.gomez@MacBook-Pro-de-FernandoGomez ~ $ git clone https://github.com/wizeline/workshop_intro_scala
Cloning into 'workshop_intro_scala' ...
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 28 (delta 8), reused 26 (delta 6), pack-reused 0
Receiving objects: 100% (28/28), 5.07 MiB | 4.46 MiB/s, done.
Resolving deltas: 100% (8/8), done.
fernando.gomez@MacBook-Pro-de-FernandoGomez ~ $ cd workshop_intro_scala
fernando.gomez@MacBook-Pro-de-FernandoGomez ~/workshop_intro_scala <main>$ vagrant up
```



3. Si la máquina virtual se crea correctamente, debería aparecer un mensaje parecido a las últimas líneas de la terminal de la siguiente imagen:



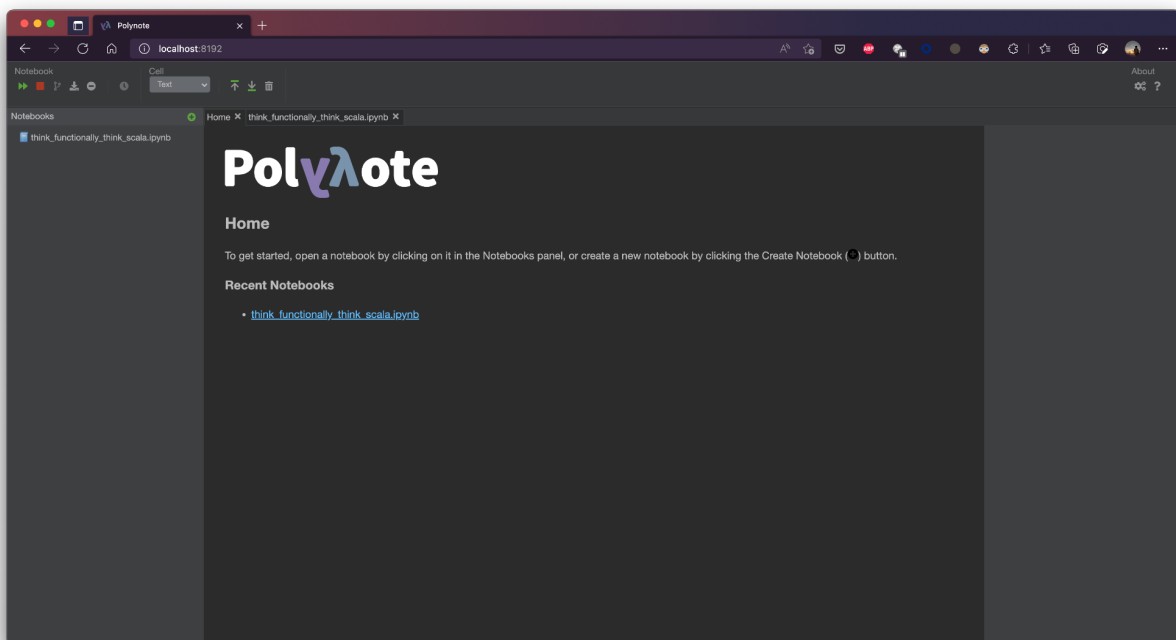
```

main Collecting tornado==6.0
main Downloading tornado-6.1-cp37-cp37m-manylinux2018_x86_64.whl (428 kB)
main Building wheels for collected packages: jep
main Building wheel for jep (setup.py) started
main Building wheel for jep (setup.py) finished with status 'done'
main Created wheel for jep: filename=jep-4.0.0-cp37-cp37m-linux_x86_64.whl size=1821737 sha256=eee18af09e9ba100d29a36840bfcd55eafa90b9ed8a3b1f5d5e5b5aed9acel
main Stored in directory: /root/.cache/pip/wheels/c4/85/b4/81129441094834fca18ad86987cb171f716818da09e370c6f
main Successfully built jep
main Installing collected packages: distlib, platformdirs, filelock, virtualenv, pygments, pickleshare, decorator, wcwidth, prompt-toolkit, traitlets, matplotlib-inline, parso, jedi, backcall, ipython, MarkupSafe, defusedxml, jupyterlab-pygments, jupyter-core, fastjsonschema, nbformat, webencodings, bleach, mistune, testpath, soupsieve, beautifulsoup4, nest-asyncio, python-dateutil, pyrmq, tornado, jupyter-client, nbclient, pandocfilters, nbconvert, numpy, pytz, pandas, jep
main Attempting uninstall: MarkupSafe
main Found existing installation: MarkupSafe 1.1.0
main Not uninstalling markupsafe at /usr/lib/python3/dist-packages, outside environment /usr
main Can't uninstall 'MarkupSafe'. No files were found to uninstall.
main Successfully installed MarkupSafe-2.1.1 backcall-0.2.0 beautifulsoup4-4.10.0 bleach-5.0.0 decorator-5.1.1 defusedxml-0.7.1 distlib-0.3.4 fastjsonschema-2.15.3 filelock-3.6.0 ipython-7.32.0 jedi-0.18.1 jep-4.0.0 jupyter-client-7.2.2 jupyter-core-4.9.2 jupyterlab-pygments-0.1.2 matplotlib-inline-0.1.3 mistune-0.8.4 nbclient-0.5.13 nbconvert-6.4.5 nbformat-5.3.0 nest-asyncio-1.5.5 numpy-1.21.5 pandas-1.2.5 pandocfilters-1.5.0 parso-0.8.3 pickleshare-0.7.5 platformdirs-2.5.1 prompt-toolkit-3.0.29 pygments-2.11.2 python-dateutil-2.8.2 pytz-2022.1 pyzmq-22.3.0 soupsieve-2.3.2 testpath-0.6.0 tornado-6.1 traitlets-5.1.1 virtualenv-20.14.0 wcwidth-0.2.5 webencodings-0.5.1
=> main: Running provisioner: shell...
main: Running: inline script

=> main: Machine 'main' has a post 'vagrant up' message. This is a message
=> main: from the creator of the Vagrantfile, and not from Vagrant itself:
=> main:
=> main: POLYNOTE RUNNING
=> main: Connect to the VM by using the browser in "http://localhost:8192" or by running "vagrant ssh main"
fernando.gomez@MacBook-Pro-de-FernandoGomez: ~/workshop_intro_scala <main>$

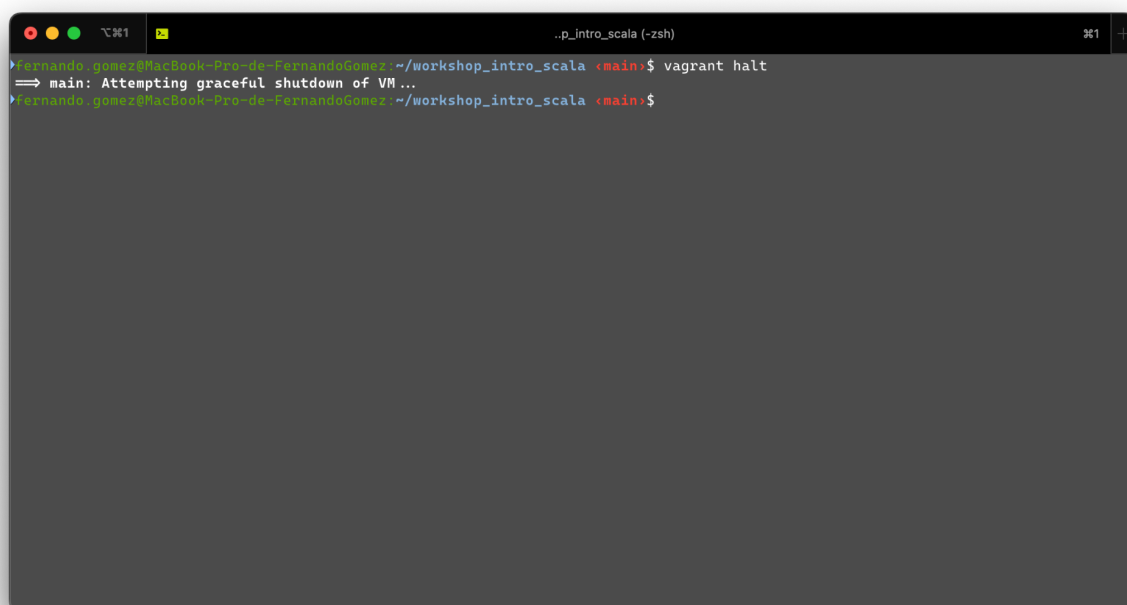
```

4. Se puede acceder a la interfaz del programa desde el navegador visitando la dirección <http://localhost:8192>



Detener o eliminar la máquina virtual

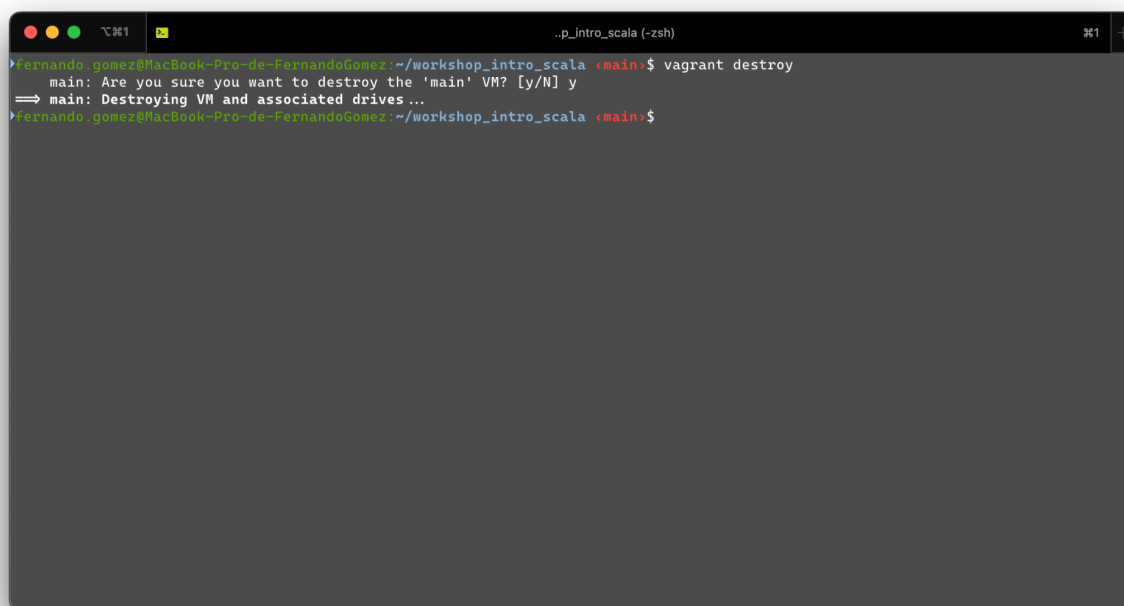
1. Para detener la máquina virtual se debe ejecutar el comando “vagrant halt” estando dentro del directorio del repositorio desde la terminal.



```
fernando.gomez@MacBook-Pro-de-FernandoGomez:~/workshop_intro_scala <main>$ vagrant halt
=> main: Attempting graceful shutdown of VM ...
fernando.gomez@MacBook-Pro-de-FernandoGomez:~/workshop_intro_scala <main>$
```



2. Para eliminar la máquina virtual se debe ejecutar el comando “vagrant destroy” estando dentro del directorio del repositorio desde la terminal. Al pedir la confirmación de eliminación, escribir la letra “y” para confirmar.

A terminal window titled '..p_intro_scala (-zsh)' showing the execution of the 'vagrant destroy' command. The prompt is 'fernando.gomez@MacBook-Pro-de-FernandoGomez: ~/workshop_intro_scala <main>'. The command 'vagrant destroy' is entered, followed by a confirmation prompt 'main: Are you sure you want to destroy the 'main' VM? [y/N]'. The user enters 'y', and the terminal shows '⇒ main: Destroying VM and associated drives...'. The prompt returns to 'fernando.gomez@MacBook-Pro-de-FernandoGomez: ~/workshop_intro_scala <main>'.

```
fernando.gomez@MacBook-Pro-de-FernandoGomez: ~/workshop_intro_scala <main>$ vagrant destroy
main: Are you sure you want to destroy the 'main' VM? [y/N] y
⇒ main: Destroying VM and associated drives...
fernando.gomez@MacBook-Pro-de-FernandoGomez: ~/workshop_intro_scala <main>$
```

