



Tecnológico nacional de México

Instituto Tecnológico de Iztapalapa

TFRT: A New TensorFlow Runtime

Alumnos:	N° Control	Porcentaje de Participación
Santamaria Cirilo Norma Nelly	171080086	50%
Zamudio Díaz de león Cristian Alejandro	171080012	50%
Jiménez Pichardo Francisco Joel	161080157	0%
Ventura Vega Samuel Eduardo	151080099	0%

Profesor: Parra Hernández Abiel Tomas

Materia: Lenguajes y Autómatas II

Fecha de Entrega: 22 junio de 2021

Índice

Introducción	4
1. Generalidades del Proyecto	5
1.2 Objetivos	5
1.3 Justificación	5
2. Marco Teórico	6
2.2 Compilador	6
2.3 Interprete	6
2.4 Ensamblador	6
2.5 MLIR	6
2.6 TFRT	6
2.7 Bazel	7
2.8 Clang	7
2.9 Python	7
2.10 pip3	7
3. Metodología	8
4. Desarrollo e Implementación	9
5. Resultados Obtenidos	20
6. Conclusiones	23
7. Fuentes de Consulta	24

Resumen

Este proyecto se basó en el uso de compiladores extensibles y reutilizables. Basándonos en la descripción general de la representación intermedia de varios niveles la cual tiene como objetivo la fragmentación del software y mejorar la compilación para hardware heterogéneo y reducir el costo de construcción de compiladores específicos de dominio y ayudar a compiladores existentes.

Introducción

En la actualidad existe inmensa cantidad de compiladores algunos tienen costo y otros gratuitos, pero también existen varios tipos desconocidos, en este proyecto nos hemos basado en el modelo MLIR el cual es un diseño de múltiples requerimientos en una infraestructura unificada la cual nos permitirá conocer más de un compilador dinámico y extensible. Mostraremos un programa ejecutable que proporciona una capa de infraestructura unificada y extensible con un mejor rendimiento, el TFRT.

1. Generalidades del proyecto

1.2 Objetivos

General:

Aprender a usar un programa ejecutable de infraestructura unificada y extensible, conocer ampliamente el uso de este software y saber desarrollar programas asíncronos que se encuentren en bajo nivel

Específicos:

- Investigar todos los conceptos principales
- Analizar las necesidades del programa para su ejecución
- Buscar e instalar las herramientas para TRFT
- Implementar un programa ejecutable

1.3 Justificación

El presente proyecto se enfoca en el estudio de los procesadores de lenguaje de programación, enfocándose en distintos puntos como

- Representación grafica
- Optimizaciones y transformaciones

También para explicar ampliamente el compilador muy poco conocido llamado TensorFlow.

2. MARCO TEORICO

2.1 Compilador

Traductor utilizado para convertir un lenguaje de programación de alto nivel en un lenguaje de programación de bajo nivel el cual convierte todo un programa en sesión para informar de los errores detectados después de la conversión. El compilador traduce el código de alto nivel al código de nivel inferior de una vez y luego lo guarda en la memoria, el compilador dependerá del procesador y de la plataforma.

2.2 Interprete

Similar al compilador, ya que es un traductor que se utiliza para convertir un lenguaje de programación de bajo nivel. El intérprete diferencia del compilador ejecuta el código inmediatamente después de leerlo, el cual puede utilizar una herramienta de depuración para el desarrollo de software, ya que puede ejecutar una sola línea de código a la vez.

2.3 Ensamblador

Tiene la misma función que un compilador para el lenguaje ensamblador, pero funciona como un intérprete, este lenguaje es difícil de entender ya que es un lenguaje de programación de bajo nivel, este lenguaje convierte un programa de bajo nivel a un nivel más bajo como el código maquina

2.4 MLIR

Enfoque novedoso para la construcción de una infraestructura de compilador extensibles y reutilizable, tiene como objetivo abordar la fragmentación del software, mejorar la compilación para hardware heterogéneo, reducir significativamente el costo de construir compiladores específicos de dominio y ayudar a conectar compiladores existentes.

Diseñado para ser un IR híbrido que puede soportar múltiples requisitos diferentes en una infraestructura unificada el MLIR común acepta operaciones específicas de hardware.

MLIR es una representación poderosa, pero tiene objetivos, se adapta a optimizadores de nivel inferior como LLVM, proporciona la columna vertebral para representar cualquier DSL e integrarlo en el ecosistema

2.5 TFRT

Es un tiempo de ejecución de TensorFlow, su objetivo es proporcionar una capa de infraestructura unificada y extensible con el mejor rendimiento de su clase en una amplia variedad de hardware específico de dominio.

Este beneficiará una amplia gama de usuarios, pero será de especial interés para usted si es investigador, desarrollador o fabricante.

TFRT es un proyecto en etapa inicial y aún no está listo para uso general, actualmente solo es compatible con Ubuntu – 16.04 para trabajar con este requerimos Bazel y Clang.

2.6 Bazel

Bazel es una herramienta de software gratuita que permite la automatización de la construcción y prueba de software.

De manera similar a las herramientas de compilación como Make , Apache Ant o Apache Maven , Bazel crea aplicaciones de software a partir del código fuente utilizando un conjunto de reglas.

2.7 Clang

Clang es un front end de compilador para los lenguajes de programación C, C++, Objective-C y Objective-C++. Usa LLVM como su back-end y ha sido parte del ciclo de lanzamiento de LLVM desde la versión 2.6.

Está diseñado para ofrecer un reemplazo de GNU Compiler Collection (GCC). Es de código abierto,³ y varias compañías de software están involucradas en su desarrollo, incluyendo a Google y Apple.

2.8 Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

2.9 Pip 3

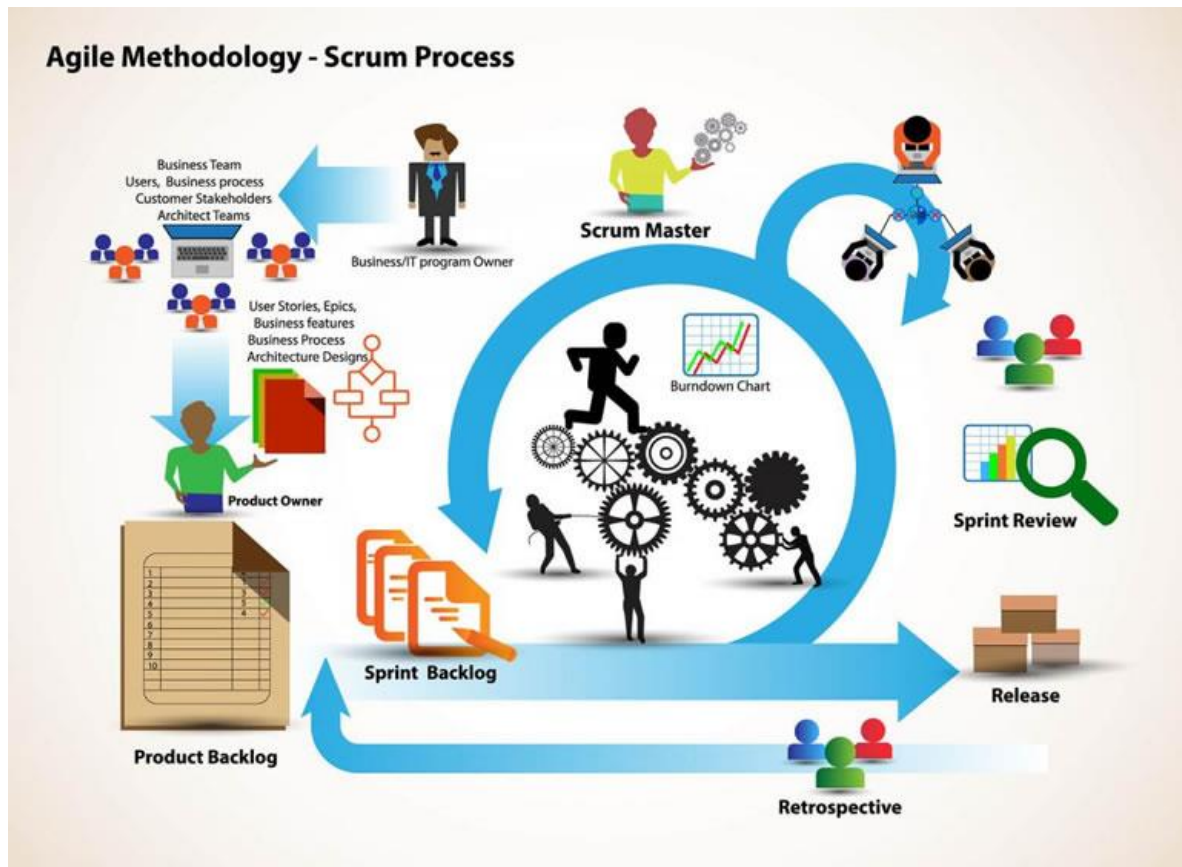
pip es un sistema de gestión de paquetes utilizado para instalar y administrar paquetes de software escritos en Python. Muchos paquetes pueden ser encontrados en el Python Package Index (PyPI). Python 2.7.9 y posteriores (en la serie Python2), Python 3.4 y posteriores incluyen pip (pip3 para Python3) por defecto.

pip es un acrónimo recursivo que se puede interpretar como Pip Instalador de Paquetes o Pip Instalador de Python

3. Metodología

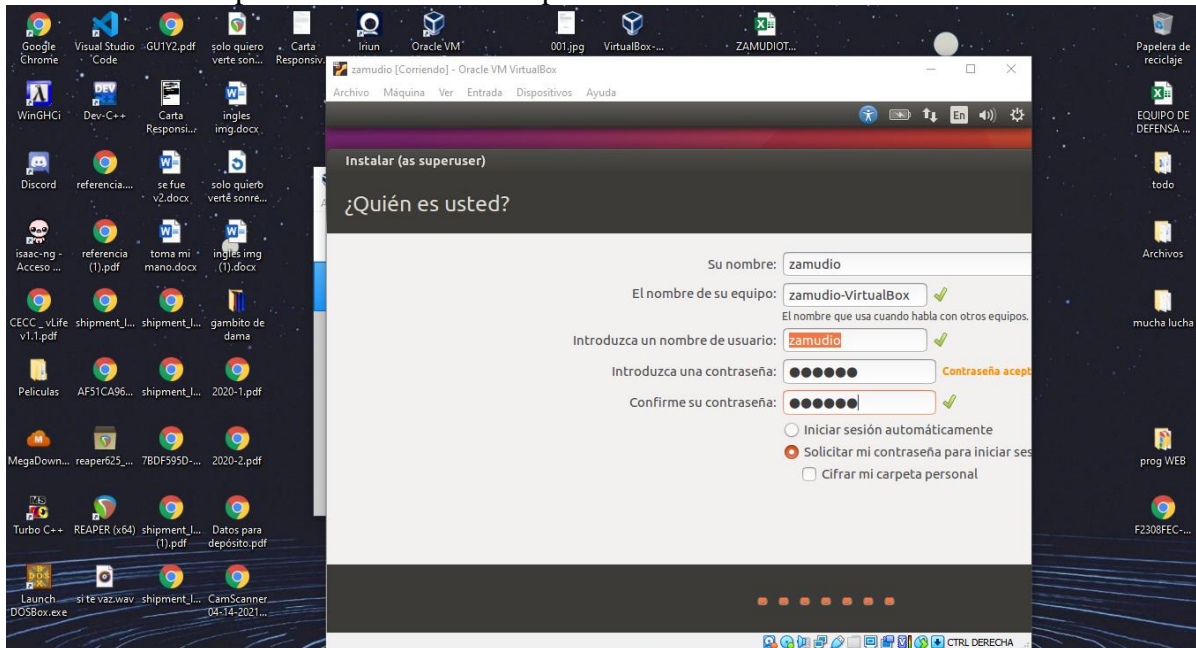
La metodología que utilizamos fue Scrum la cual se basa en un marco de trabajo o framework que se utiliza dentro de equipos que manejan proyectos complejos. Es decir, se trata de una metodología de trabajo ágil que tiene como finalidad la entrega de valor en periodos cortos de tiempos, este se basa en tres pilares principales

- Transparecía
- Inspección
- Adaptación

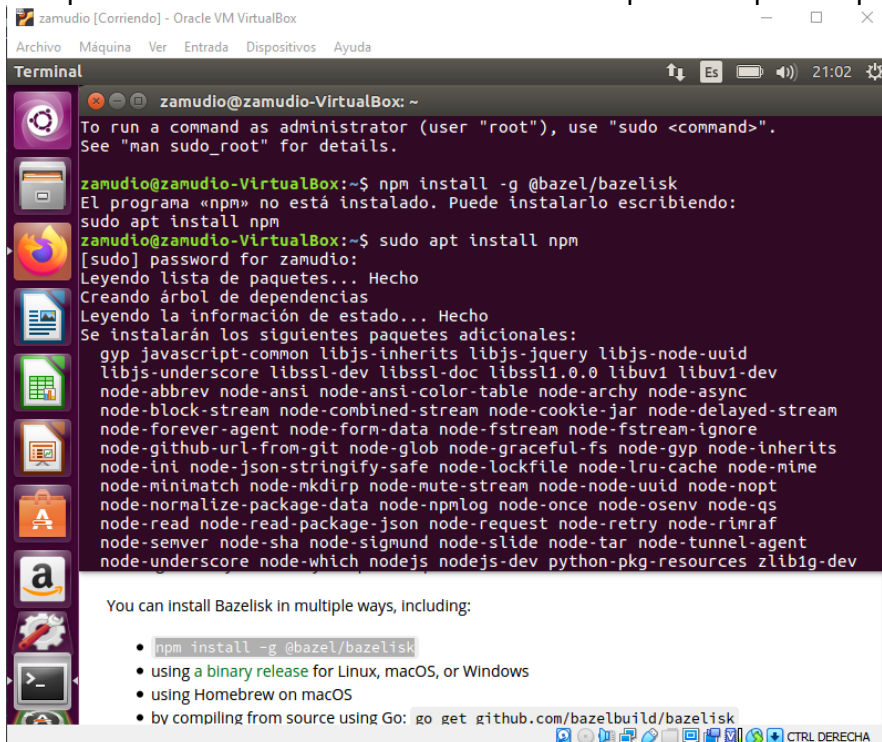


4. Desarrollo e Implementación

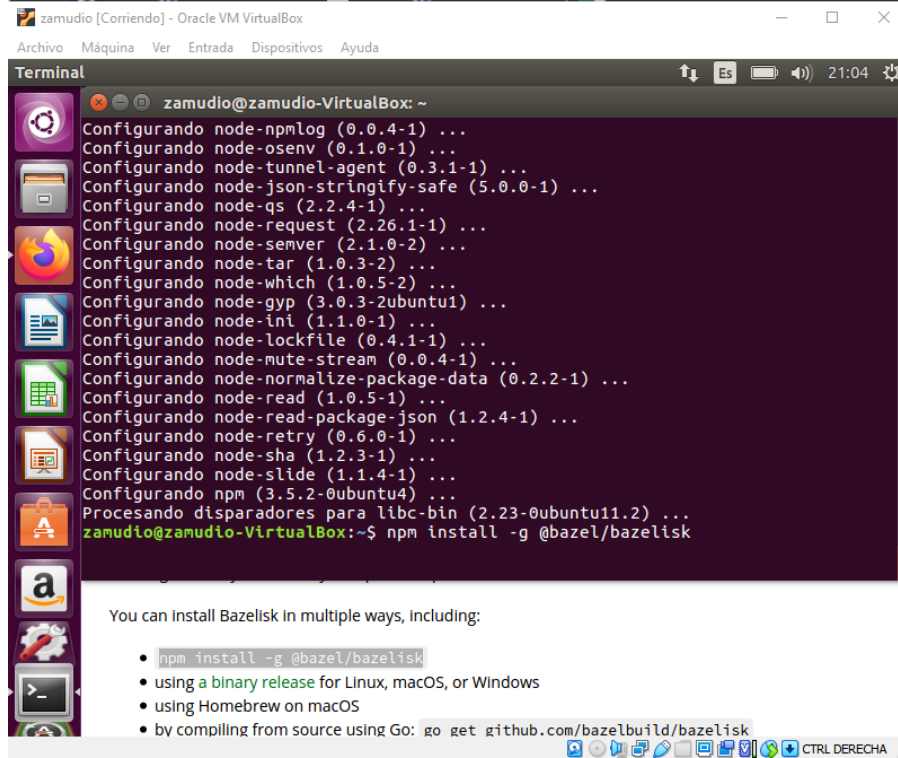
Lo primero que tuvimos que hacer fue crear un pc virtual para poder trabajar en Ubuntu pues en su repositorio de github mencionan que, aunque en un futuro se podrá trabajar en Windows o macs por el momento solo se puede en Ubuntu.



Así que se instaló ubuntu en la versión 16.04 que es la que nos pide.



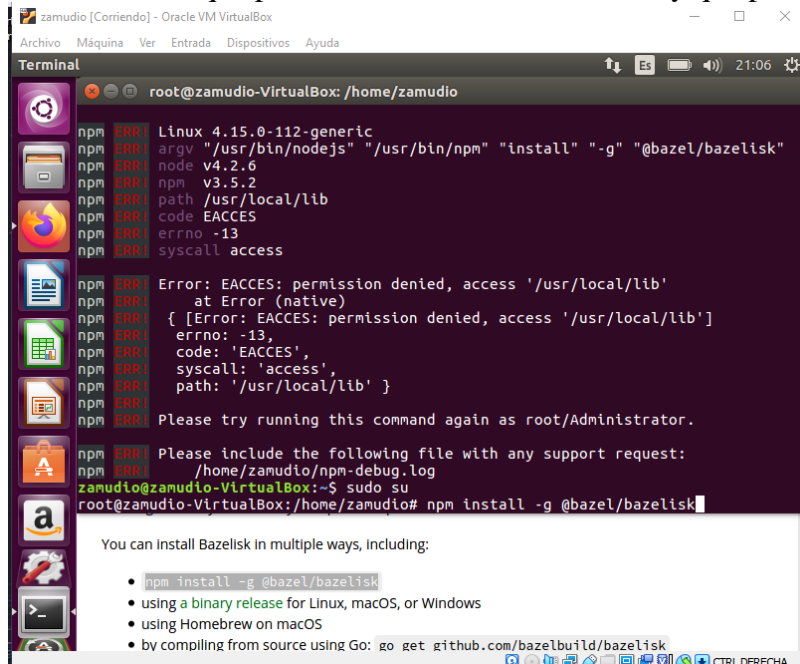
Ya con ubuntu instalado nos dirigimos a la terminal para empezar a trabajar con los códigos de instalación. El primer código es para instalar “Bazel”, al escribirlo nos pidió que primero instalaremos “npm”, así que con el código que ahí mismo nos dio lo instalamos.



```
zamudio [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Terminal
zamudio@zamudio-VirtualBox: ~
Configurando node-npmlog (0.0.4-1) ...
Configurando node-osenv (0.1.0-1) ...
Configurando node-tunnel-agent (0.3.1-1) ...
Configurando node-json-stringify-safe (5.0.0-1) ...
Configurando node-qs (2.2.4-1) ...
Configurando node-request (2.26.1-1) ...
Configurando node-semver (2.1.0-2) ...
Configurando node-tar (1.0.3-2) ...
Configurando node-which (1.0.5-2) ...
Configurando node-gyp (3.0.3-2ubuntu1) ...
Configurando node-ini (1.1.0-1) ...
Configurando node-lockfile (0.4.1-1) ...
Configurando node-mute-stream (0.0.4-1) ...
Configurando node-normalize-package-data (0.2.2-1) ...
Configurando node-read (1.0.5-1) ...
Configurando node-read-package-json (1.2.4-1) ...
Configurando node-retry (0.6.0-1) ...
Configurando node-sha (1.2.3-1) ...
Configurando node-slide (1.1.4-1) ...
Configurando npm (3.5.2-0ubuntu4) ...
Procesando disparadores para libc-bin (2.23-0ubuntu11.2) ...
zamudio@zamudio-VirtualBox:~$ npm install -g @bazel/bazelisk

You can install Bazelisk in multiple ways, including:
• npm install -g @bazel/bazelisk
• using a binary release for Linux, macOS, or Windows
• using Homebrew on macOS
• by compiling from source using Go: go get github.com/bazelbuild/bazelisk
```

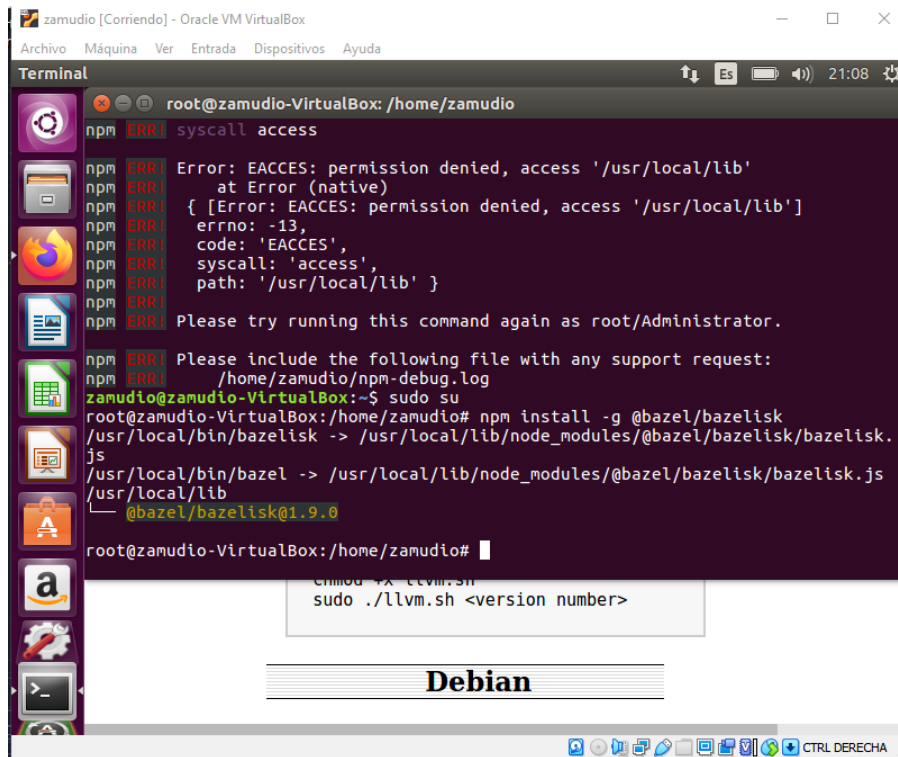
Aquí ya termino de instalar “npm”, ahora si tratamos de instalar “Bazel”, pero de nuevo me mando error y me pidió que primero entrara en modo root, y con una pequeña búsqueda en internet vimos que para entrar en modo root solo hay que poner el siguiente código.



```
zamudio [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Terminal
root@zamudio-VirtualBox: /home/zamudio
npm ERR! Linux 4.15.0-112-generic
npm ERR! argv "/usr/bin/nodejs" "/usr/bin/npm" "install" "-g" "@bazel/bazelisk"
npm ERR! node v4.2.6
npm ERR! npm v3.5.2
npm ERR! path /usr/local/lib
npm ERR! code EACCES
npm ERR! errno -13
npm ERR! syscall access
npm ERR! Error: EACCES: permission denied, access '/usr/local/lib'
npm ERR! at Error (native)
npm ERR! { [Error: EACCES: permission denied, access '/usr/local/lib']
npm ERR!   errno: -13,
npm ERR!   code: 'EACCES',
npm ERR!   syscall: 'access',
npm ERR!   path: '/usr/local/lib' }
npm ERR! Please try running this command again as root/Administrator.
npm ERR! Please include the following file with any support request:
npm ERR! /home/zamudio/npm-debug.log
zamudio@zamudio-VirtualBox:~$ sudo su
root@zamudio-VirtualBox:/home/zamudio# npm install -g @bazel/bazelisk

You can install Bazelisk in multiple ways, including:
• npm install -g @bazel/bazelisk
• using a binary release for Linux, macOS, or Windows
• using Homebrew on macOS
• by compiling from source using Go: go get github.com/bazelbuild/bazelisk
```

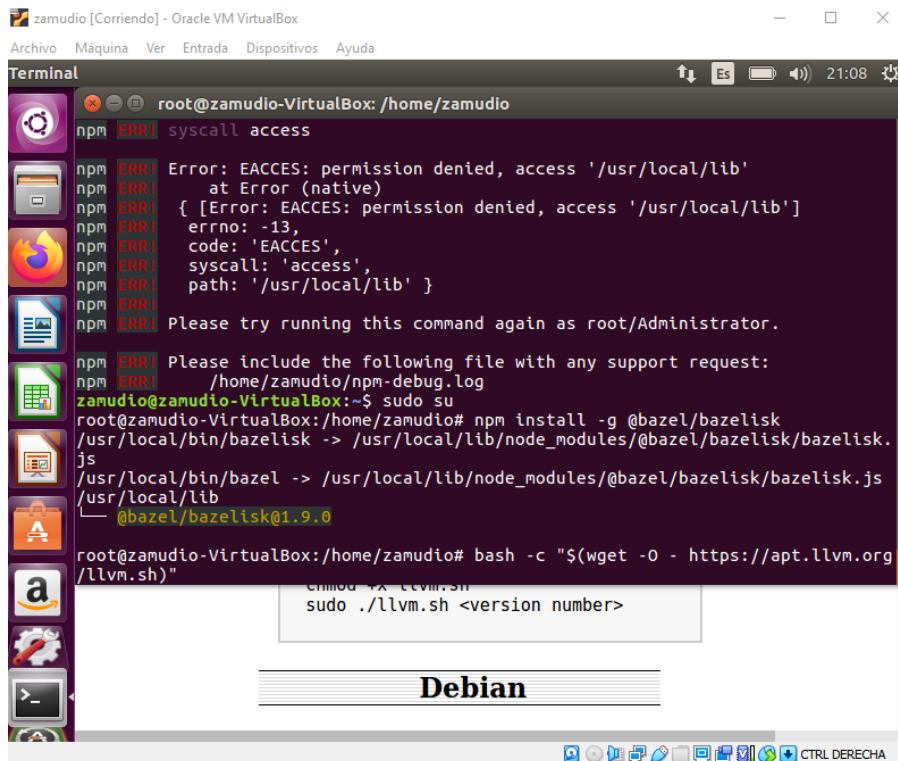
Ya en modo root volvimos a tratar de instalar Bazel con el mismo código



```
root@zamudio-VirtualBox: /home/zamudio
npm ERR! syscall access
npm ERR! Error: EACCES: permission denied, access '/usr/local/lib'
npm ERR!   at Error (native)
npm ERR!   { [Error: EACCES: permission denied, access '/usr/local/lib']
npm ERR!     errno: -13,
npm ERR!     code: 'EACCES',
npm ERR!     syscall: 'access',
npm ERR!     path: '/usr/local/lib' }
npm ERR! Please try running this command again as root/Administrator.
npm ERR! Please include the following file with any support request:
npm ERR!   /home/zamudio/npm-debug.log
zamudio@zamudio-VirtualBox:~$ sudo su
root@zamudio-VirtualBox:/home/zamudio# npm install -g @bazel/bazelisk
/usr/local/bin/bazelisk -> /usr/local/lib/node_modules/@bazel/bazelisk/bazelisk.
js
/usr/local/bin/bazel -> /usr/local/lib/node_modules/@bazel/bazelisk/bazelisk.js
/usr/local/lib
└─ @bazel/bazelisk@1.9.0
root@zamudio-VirtualBox:/home/zamudio#
```

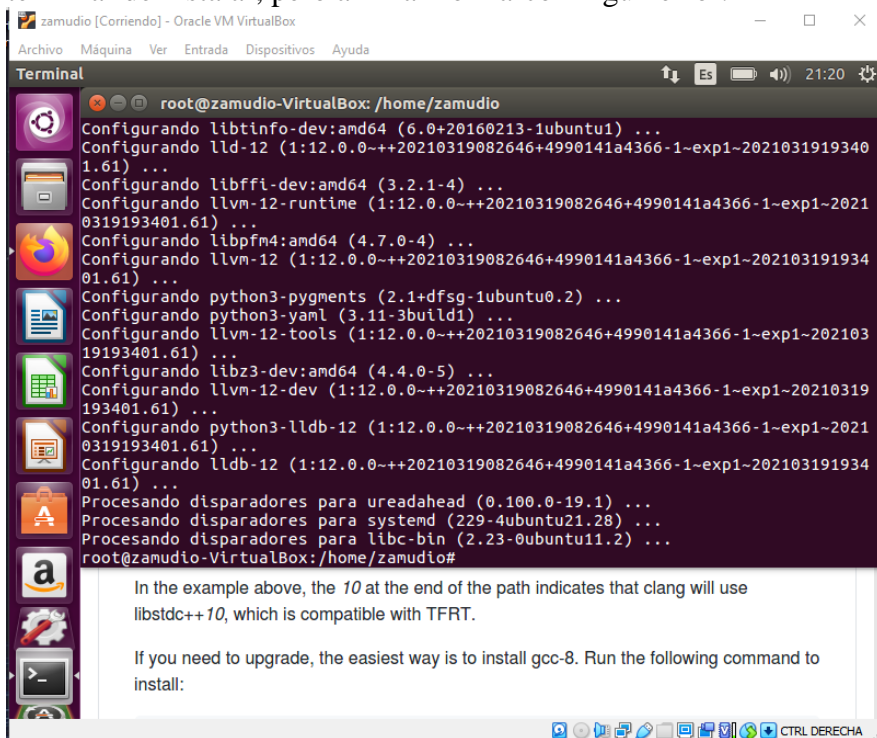
Y esta vez ya no marco ningún error.

Lo siguiente a instalar es Clang, así que con el código que encontramos en la página oficial lo tratamos de instalar



```
root@zamudio-VirtualBox: /home/zamudio
npm ERR! syscall access
npm ERR! Error: EACCES: permission denied, access '/usr/local/lib'
npm ERR!   at Error (native)
npm ERR!   { [Error: EACCES: permission denied, access '/usr/local/lib']
npm ERR!     errno: -13,
npm ERR!     code: 'EACCES',
npm ERR!     syscall: 'access',
npm ERR!     path: '/usr/local/lib' }
npm ERR! Please try running this command again as root/Administrator.
npm ERR! Please include the following file with any support request:
npm ERR!   /home/zamudio/npm-debug.log
zamudio@zamudio-VirtualBox:~$ sudo su
root@zamudio-VirtualBox:/home/zamudio# npm install -g @bazel/bazelisk
/usr/local/bin/bazelisk -> /usr/local/lib/node_modules/@bazel/bazelisk/bazelisk.
js
/usr/local/bin/bazel -> /usr/local/lib/node_modules/@bazel/bazelisk/bazelisk.js
/usr/local/lib
└─ @bazel/bazelisk@1.9.0
root@zamudio-VirtualBox:/home/zamudio# bash -c "$(wget -O - https://apt.llvm.org
/llvm.sh)"
sudo ./llvm.sh <version number>
```

Como la instalación la hicimos con el script de instalación automática demoro bastante en terminar de instalar, pero al final no marco ningún error.

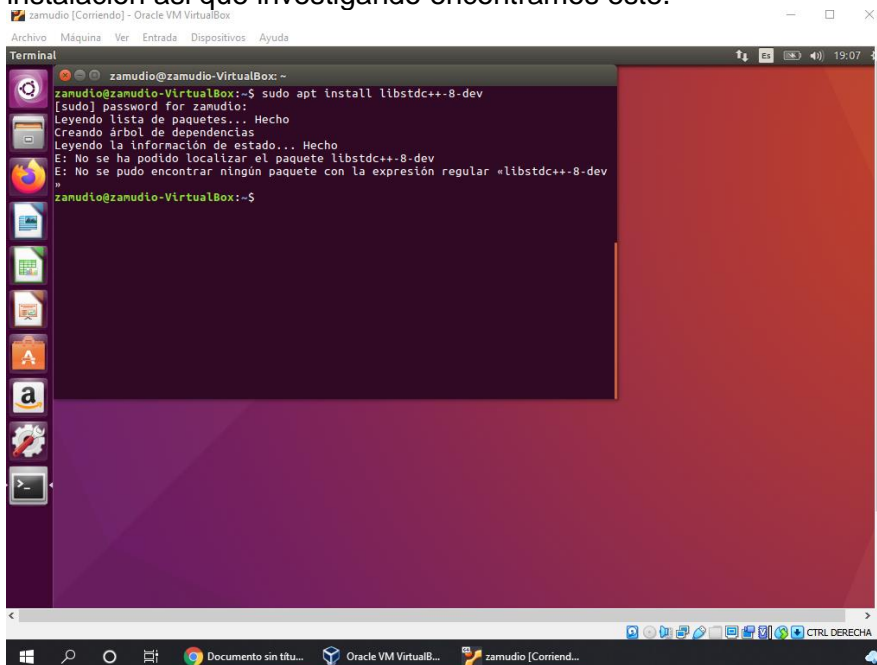


```
root@zamudio-VirtualBox: /home/zamudio
Configurando libtinfo-dev:amd64 (6.0+20160213-1ubuntu1) ...
Configurando lld-12 (1:12.0.0-++20210319082646+4990141a4366-1-exp1~20210319193401.61) ...
Configurando libffi-dev:amd64 (3.2.1-4) ...
Configurando llvm-12-runtime (1:12.0.0-++20210319082646+4990141a4366-1-exp1~20210319193401.61) ...
Configurando libpfm4:amd64 (4.7.0-4) ...
Configurando llvm-12 (1:12.0.0-++20210319082646+4990141a4366-1-exp1~20210319193401.61) ...
Configurando python3-pygments (2.1+dfsg-1ubuntu0.2) ...
Configurando python3-yaml (3.11-3build1) ...
Configurando llvm-12-tools (1:12.0.0-++20210319082646+4990141a4366-1-exp1~20210319193401.61) ...
Configurando libz3-dev:amd64 (4.4.0-5) ...
Configurando llvm-12-dev (1:12.0.0-++20210319082646+4990141a4366-1-exp1~20210319193401.61) ...
Configurando python3-lldb-12 (1:12.0.0-++20210319082646+4990141a4366-1-exp1~20210319193401.61) ...
Configurando lldb-12 (1:12.0.0-++20210319082646+4990141a4366-1-exp1~20210319193401.61) ...
Procesando disparadores para ureadahead (0.100.0-19.1) ...
Procesando disparadores para systemd (229-4ubuntu21.28) ...
Procesando disparadores para libc-bin (2.23-0ubuntu11.2) ...
root@zamudio-VirtualBox:/home/zamudio#
```

In the example above, the `10` at the end of the path indicates that clang will use libstdc++ `10`, which is compatible with TFRT.

If you need to upgrade, the easiest way is to install gcc-8. Run the following command to install:

Lo siguiente a instalar es Libstdc++ version 8. Como tal no nos dio ningun codigo de instalacion asi que investigando encontramos este.



```
zamudio@zamudio-VirtualBox:~$ sudo apt install libstdc++-8-dev
[sudo] password for zamudio:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
E: No se ha podido localizar el paquete libstdc++-8-dev
E: No se pudo encontrar ningún paquete con la expresión regular «libstdc++-8-dev»
zamudio@zamudio-VirtualBox:~$
```

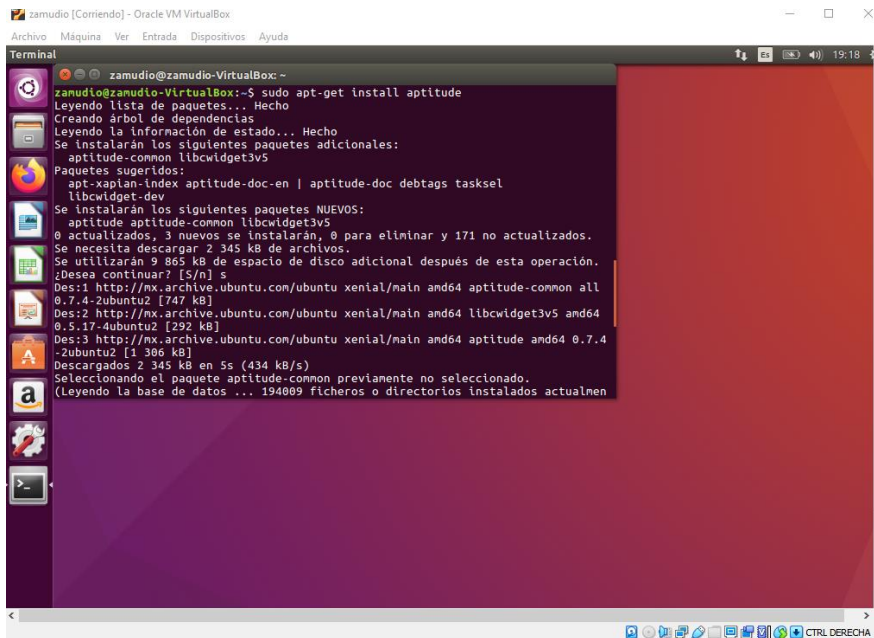

Como vemos no reconocia el codigo que escribimos y estuvimos tratando algunos otros per con todos era lo mismo, asi que buscamos por que nos daba ese error y encontramos lo siguiente

When apt doesn't find an exact match to a package name, it treats the given string as a regular expression. In this case the `++` causes it to try to install *any* package whose name matches `libstd` followed by one or more `c` - many of which conflict, as you can see.

En el foro donde hablan de este problema dan algunas alternativas para solucionarlo, asi que las probamos

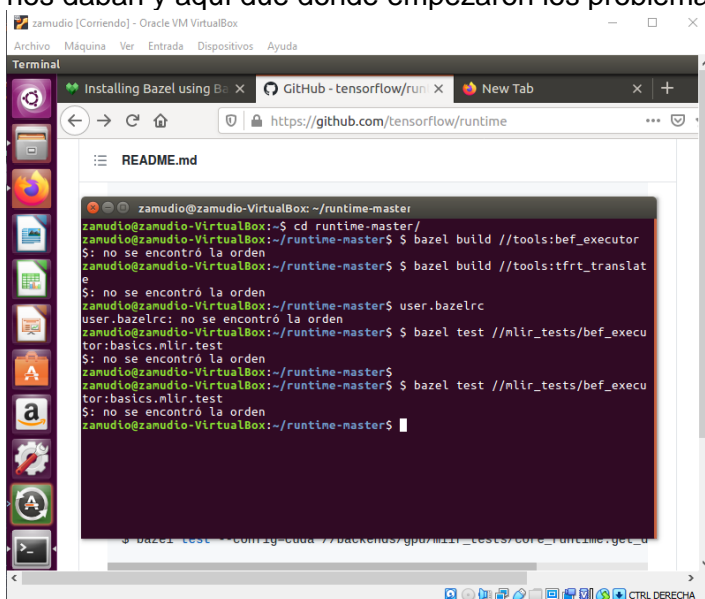
The image shows two screenshots of a terminal window running inside an Oracle VM VirtualBox. The terminal is titled 'Terminal' and shows the user 'zamudio' at the prompt 'zamudio@zamudio-VirtualBox: ~'. The first screenshot shows the user running 'sudo apt-get install build-essential'. The output shows the download of build-essential and its dependencies, including gcc, g++ and their associated libraries. The second screenshot shows the user running 'sudo apt-get update && sudo apt-get upgrade'. The output shows the update of various packages, including gcc, g++ and their associated libraries. The terminal window is part of a larger application window titled 'zamudio [Corriendo] - Oracle VM VirtualBox'.

```
zamudio@zamudio-VirtualBox: ~  
Des:7 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 DEP-11 Metadata [130 kB]  
Des:8 http://mx.archive.ubuntu.com/ubuntu xenial-updates/main amd64 DEP-11 Metadata [327 kB]  
Des:9 http://mx.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 DEP-11 Metadata [281 kB]  
Des:10 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 DEP-11 Metadata [2 464 B]  
Des:11 http://mx.archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 DEP-11 Metadata [5 960 B]  
Des:12 http://mx.archive.ubuntu.com/ubuntu xenial-backports/main amd64 DEP-11 Metadata [3 328 B]  
Des:13 http://mx.archive.ubuntu.com/ubuntu xenial-backports/universe amd64 DEP-11 Metadata [6 616 B]  
Descargados 1 175 kB en 5s (224 kB/s)  
Leyendo lista de paquetes... Hecho  
sudo: apt-get: orden no encontrada  
zamudio@zamudio-VirtualBox:~$ sudo apt-get install build-essential  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
build-essential ya está en su versión más reciente (12.1ubuntu2).  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 171 no actualizados.  
zamudio@zamudio-VirtualBox:~$  
  
zamudio@zamudio-VirtualBox:~$ sudo apt-get update && sudo apt-get upgrade  
Des:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]  
Obj:2 http://mx.archive.ubuntu.com/ubuntu xenial InRelease  
Des:4 http://mx.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]  
Obj:3 http://apt.lvm.org/xenial lvm-toolchain-xenial-12 InRelease  
Des:5 http://security.ubuntu.com/ubuntu xenial-security/main amd64 DEP-11 Metadata [93.7 kB]  
Des:6 http://mx.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]  
Des:7 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 DEP-11 Metadata [130 kB]  
Des:8 http://mx.archive.ubuntu.com/ubuntu xenial-updates/main amd64 DEP-11 Metadata [327 kB]  
Des:9 http://mx.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 DEP-11 Metadata [281 kB]  
Des:10 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 DEP-11 Metadata [2 464 B]  
Des:11 http://mx.archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 DEP-11 Metadata [5 960 B]  
Des:12 http://mx.archive.ubuntu.com/ubuntu xenial-backports/main amd64 DEP-11 Metadata [3 328 B]  
Des:13 http://mx.archive.ubuntu.com/ubuntu xenial-backports/universe amd64 DEP-11 Metadata [6 616 B]  
Descargados 1 175 kB en 5s (224 kB/s)
```



```
zamudio@zamudio-VirtualBox:~$ sudo apt-get install aptitude
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
aptitude-common libcwldget3v5
Paquetes sugeridos:
apt-xapian-index aptitude-doc-en | aptitude-doc debtags tasksel
libcwldget-dev
Se instalarán los siguientes paquetes NUEVOS:
aptitude aptitude-common libcwldget3v5
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 171 no actualizados.
Se necesita descargar 2 345 kB de archivos.
Se utilizarán 9 865 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://mx.archive.ubuntu.com/ubuntu xenial/main amd64 aptitude-common all
0.7.4-2ubuntu2 [747 kB]
Des:2 http://mx.archive.ubuntu.com/ubuntu xenial/main amd64 libcwldget3v5 amd64
0.5.17-4ubuntu2 [292 kB]
Des:3 http://mx.archive.ubuntu.com/ubuntu xenial/main amd64 aptitude amd64 0.7.4
-2ubuntu2 [1 306 kB]
Descargados 2 345 kB en 5s (434 kB/s)
Seleccionando el paquete aptitude-common previamente no seleccionado.
(Leyendo la base de datos ... 194089 ficheros o directorios instalados actualmen
```

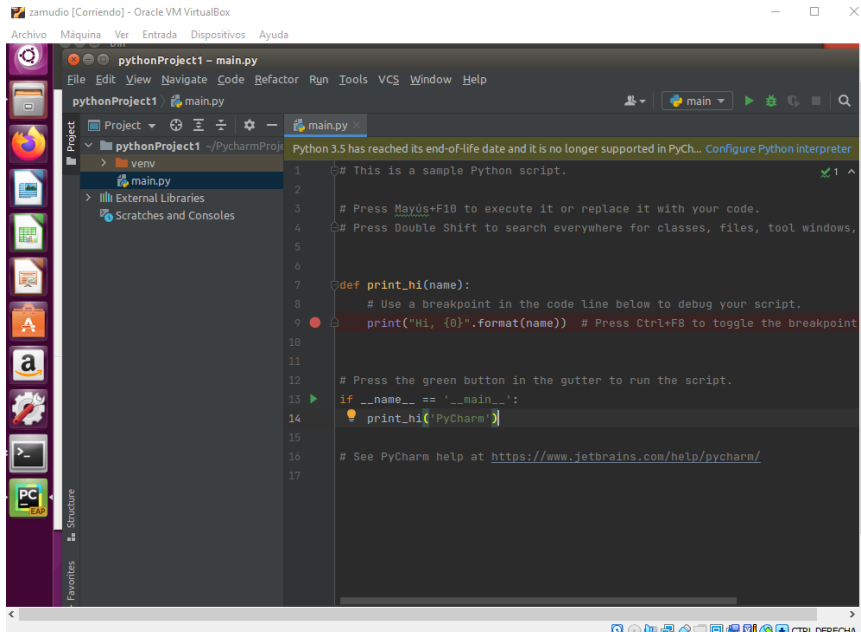
Ya con ese tema aparentemente solucionado y guiandonos en lo que dice en el repositorio de github, hicimos un “cd” a la carpeta del código igual obtenido del repositorio. Después teníamos que construir “tfrt_translate” y “bef_executor” con los códigos que ahí nos daban y aquí fue donde empezaron los problemas



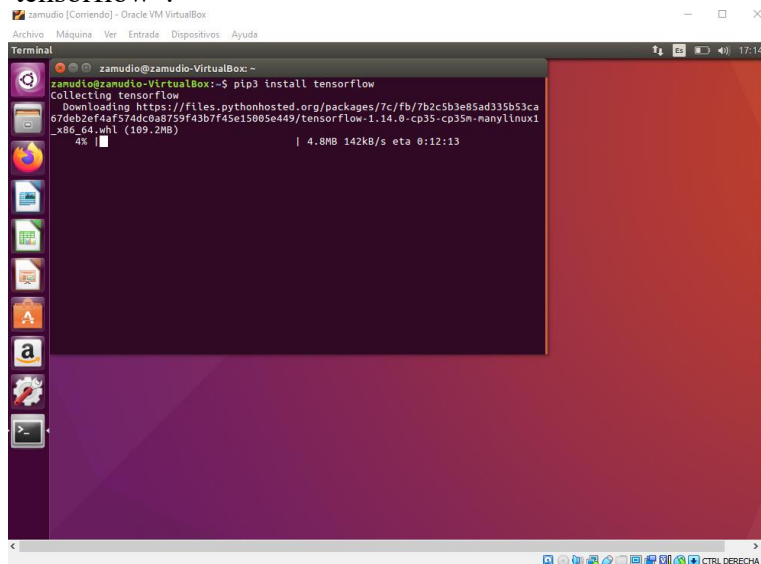
```
zamudio@zamudio-VirtualBox:~/runtime-master$ cd runtime-master/
zamudio@zamudio-VirtualBox:~/runtime-master$ bazel build //tools:bef_executor
$ no se encontró la orden
zamudio@zamudio-VirtualBox:~/runtime-master$ bazel build //tools:tfrt_translate
$ no se encontró la orden
zamudio@zamudio-VirtualBox:~/runtime-master$ user.bazelrc
user.bazelrc: no se encontró la orden
zamudio@zamudio-VirtualBox:~/runtime-master$ bazel test //nlir_tests:bef_executor:basics.nlir.test
$ no se encontró la orden
zamudio@zamudio-VirtualBox:~/runtime-master$ bazel test //nlir_tests:bef_executor:basics.nlir.test
$ no se encontró la orden
zamudio@zamudio-VirtualBox:~/runtime-master$
```

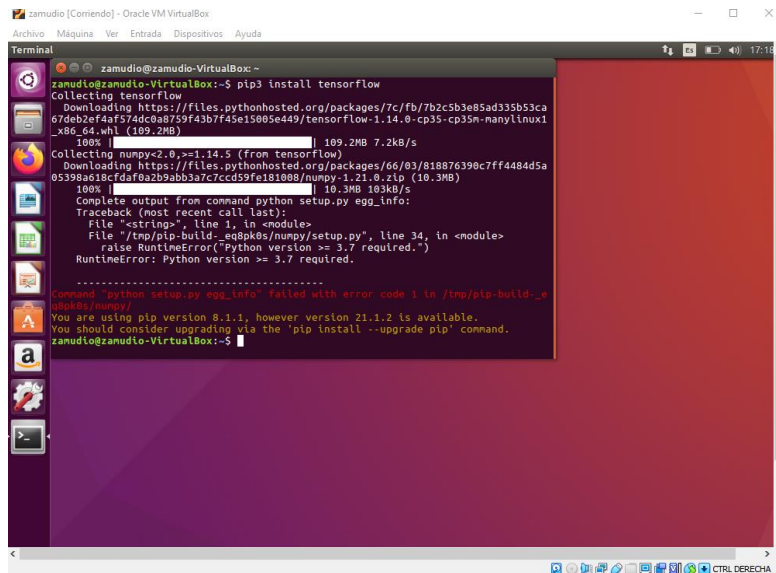
Como se ve en la imagen nos decía que no se encontraba la orden, y aunque buscamos más información no llegamos a nada concreto

The screenshot shows a Linux desktop environment. The top panel displays the system menu with the name 'zamudio [Corriendo] - Oracle VM VirtualBox'. Below the panel is a dock with application icons including a file manager, Firefox, LibreOffice, and Amazon. The main window is a file manager showing the contents of the directory '/Escritorio/pycharm-community-2021.1.2/bin'. The files listed include 'format.sh', 'fsnotifier', 'fsnotifier64', 'idea.properties', 'inspect.sh', 'libdbm64.so', 'log.xml', 'ltedit.sh', 'printenv.py', 'pycharm.png', 'pycharm.sh', 'pycharm.svg', 'pycharm.vmpointer.sh', and 'pycharm.vmpointer.sh'. A terminal window is open in the foreground, showing the command 'python3 -m pip install --user -r requirements.txt' and its output, which includes a warning about a plugin manager and a message about a bundle redefinition.



Apartir de aquí dejamos de basarnos solo en repositorio en github y empezamos a buscar mas información. Encontramos algunos programas que corrian en tensorflow asi que quisimos probarlos para empezar a experimentar. Solo teníamos que instalar una cosa mas: “tensorflow”.



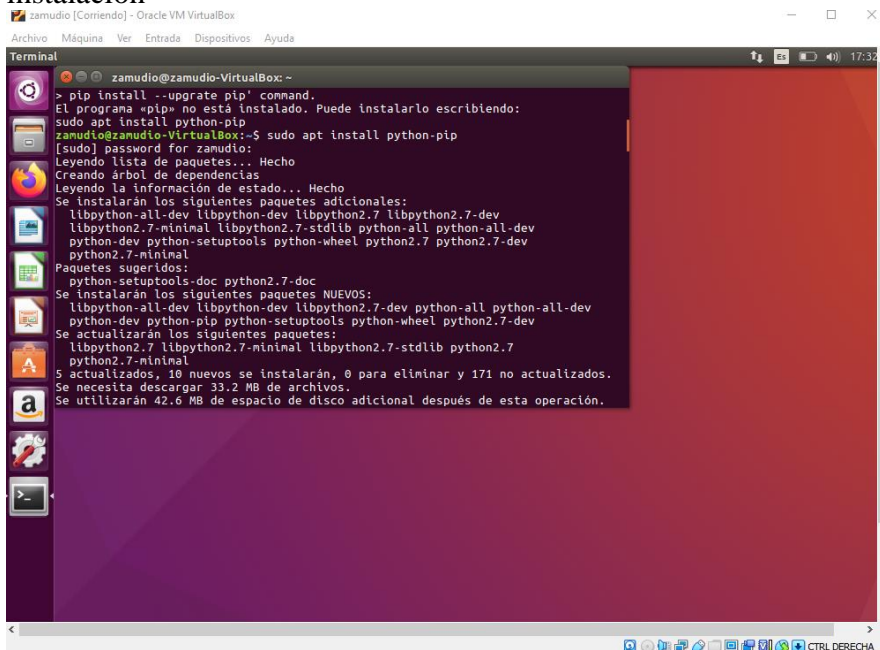


```
zamudio@zamudio-VirtualBox:~$ pip3 install tensorflow
Collecting tensorflow
  Downloading https://files.pythonhosted.org/packages/7c/fb/7b2c5b3e85ad335b53ca67deb2ef4af574dca08759f43b7f45e1508e449/tensorflow-1.14.0-cp35-cp35m-manylinux1_x86_64.whl (109.2MB)
    100% |#####| 109.2MB 7.2kB/s
Collecting numpy<2.0,=>1.14.5 (from tensorflow)
  Downloading https://files.pythonhosted.org/packages/66/03/818876390c7ff4484d5a05398a618cfdaf0a2b9abb3a7c7ccd59fe181008/numpy-1.21.0.zip (10.3MB)
    100% |#####| 10.3MB 103kB/s
Complete output from command python setup.py egg_info:
Traceback (most recent call last):
  File "<string>", line 1, in <module>
    File "/tmp/pip-build-qq8pkas/numpy/setup.py", line 34, in <module>
      raise RuntimeError("Python version >= 3.7 required.")
RuntimeError: Python version >= 3.7 required.

Command "python setup.py egg_info" failed with error code 1 in /tmp/pip-build-qq8pkas/numpy/
You are using pip version 8.1.1, however version 21.1.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
zamudio@zamudio-VirtualBox:~$
```

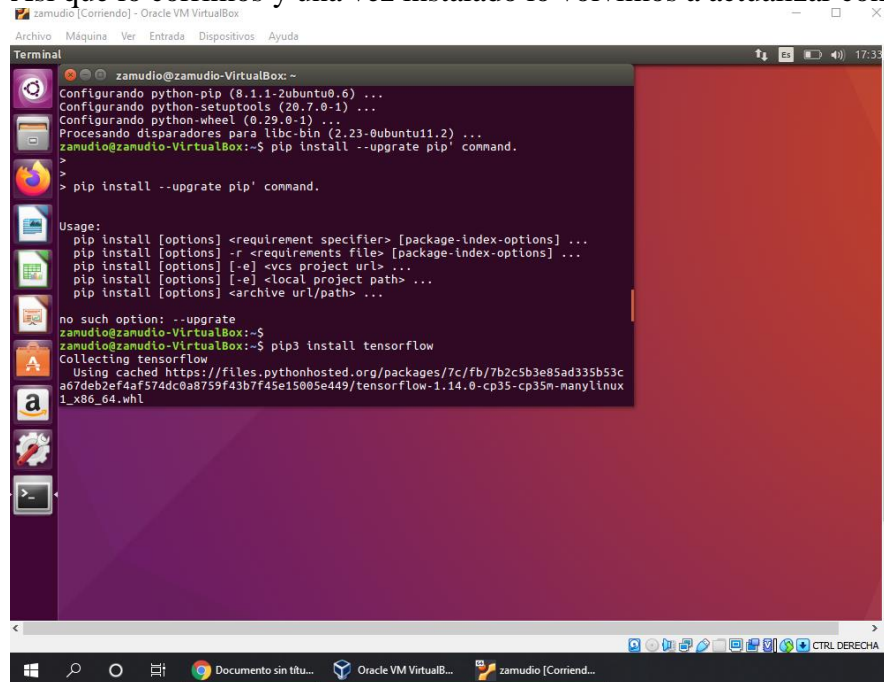
Encontramos este código para instalarlo, pero como se ve nos marca un error y nos dice básicamente que tenemos una versión muy antigua de pip, así que con el mismo código que nos da lo tratamos de actualizar

Cuando lo probamos nos dijo que aún no estaba instalado pip y nos dio un código de instalación



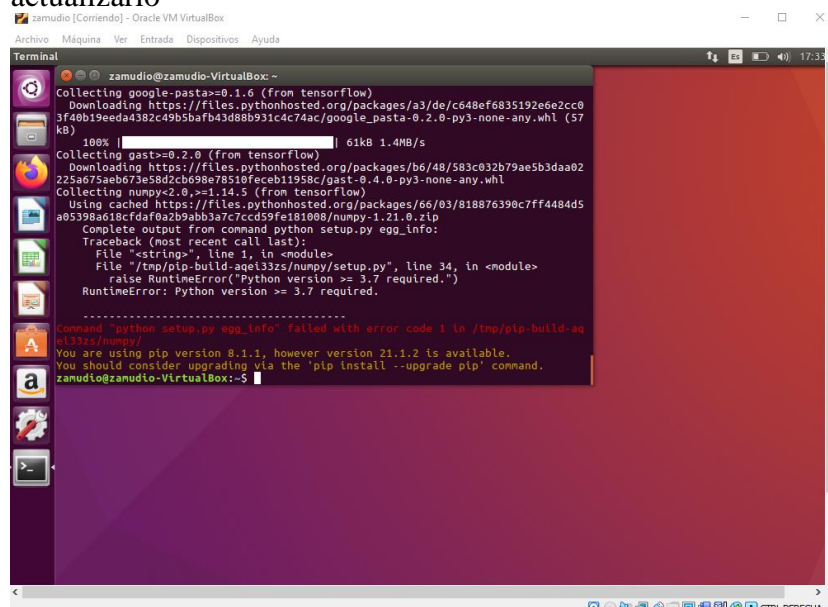
```
zamudio@zamudio-VirtualBox:~$ pip install --upgrade pip
El programa «pip» no está instalado. Puede instalarlo escribiendo:
sudo apt install python-pip
zamudio@zamudio-VirtualBox:~$ sudo apt install python-pip
[sudo] password for zamudio:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
libpython-all-dev libpython-dev libpython2.7 libpython2.7-dev
libpython2.7-minimal libpython2.7-stdlib python-all python-all-dev
python-dev python-setuptools python-wheel python2.7 python2.7-dev
python2.7-minimal
Paquetes sugeridos:
python-setuptools-doc python2.7-doc
Se instalarán los siguientes paquetes NUEVOS:
libpython-all-dev libpython-dev libpython2.7-dev python-all python-all-dev
python-dev python-pip python-setuptools python-wheel python2.7-dev
Se actualizarán los siguientes paquetes:
libpython2.7 libpython2.7-minimal libpython2.7-stdlib python2.7
python2.7-minimal
5 actualizados, 10 nuevos se instalarán, 0 para eliminar y 171 no actualizados.
Se necesita descargar 33.2 MB de archivos.
Se utilizarán 42.6 MB de espacio de disco adicional después de esta operación.
```

Así que lo corrimos y una vez instalado lo volvimos a actualizar con el código anterior



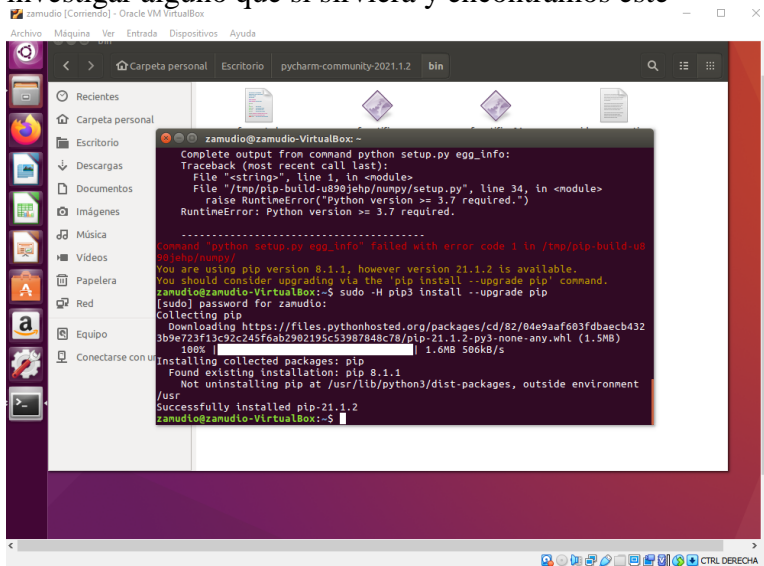
```
zamudio@zamudio-VirtualBox: ~  
Archivo Máquina Ver Entrada Dispositivos Ayuda  
Terminal  
zamudio@zamudio-VirtualBox:~$  
Configurando python-pip (8.1.1-2ubuntu0.6) ...  
Configurando python-setuptools (20.7.0-1) ...  
Configurando python-wheel (0.29.0-1) ...  
Procesando disparadores para libc-bin (2.23-0ubuntu11.2) ...  
zamudio@zamudio-VirtualBox:~$ pip install --upgrade pip' command.  
>  
> pip install --upgrade pip' command.  
Usage:  
pip install [options] <requirement specifier> [package-index-options] ...  
pip install [options] -r <requirements file> [package-index-options] ...  
pip install [options] [-e] <vcs project url> ...  
pip install [options] [-e] <local project path> ...  
pip install [options] <archive url/path> ...  
no such option: --upgrade  
zamudio@zamudio-VirtualBox:~$  
zamudio@zamudio-VirtualBox:~$ pip3 install tensorflow  
Collecting tensorflow  
Using cached https://files.pythonhosted.org/packages/7c/fb/7b2c5b3e85ad335b53c  
a6deb2ef4af574dc0a8759f43b7f45e15005e449/tensorflow-1.14.0-cp35-cp35m-manylinux  
1_x86_64.whl
```

Creímos que ya se había actualizado así que seguimos con la instalación de tensorflow pero como vemos el código de actualización no funciono puesto que nos sigue pidiendo actualizarlo



```
zamudio@zamudio-VirtualBox: ~  
Archivo Máquina Ver Entrada Dispositivos Ayuda  
Terminal  
zamudio@zamudio-VirtualBox:~$  
Collecting google-pasta>=0.1.6 (from tensorflow)  
Downloading https://files.pythonhosted.org/packages/a3/de/c648ef6835192e6e2cc0  
3f40b19eeda4382c49b5baf43d88b931c4c74ac/google_pasta-0.2.0-py3-none-any.whl (57  
kB)  
100% |#####| 61kB 1.4MB/s  
Collecting gast>=0.2.0 (from tensorflow)  
Downloading https://files.pythonhosted.org/packages/b6/48/583c032b79ae5b3daa02  
225a675ae6073e58d2c0698078510fceb11958c/gast-0.4.0-py3-none-any.whl  
Collecting numpy>=2.0.0, <=1.14.5 (from tensorflow)  
Using cached https://files.pythonhosted.org/packages/66/03/818876398c7ff4484d5  
a05398a618cfdaf0a2b9abb3a7c7ccd59fe181008/numpy-1.21.0.zip  
Complete output from command python setup.py egg_info:  
Traceback (most recent call last):  
File "<string>", line 1, in <module>  
File "/tmp/pip-build-aei33zs/numpy/setup.py", line 34, in <module>  
raise RuntimeError("Python version >= 3.7 required.")  
RuntimeError: Python version >= 3.7 required.  
-----  
Command "python setup.py egg_info" failed with error code 1 in /tmp/pip-build-ae  
i33zs/numpy/  
You are using pip version 8.1.1, however version 21.1.2 is available.  
You should consider upgrading via the 'pip install --upgrade pip' command.  
zamudio@zamudio-VirtualBox:~$
```

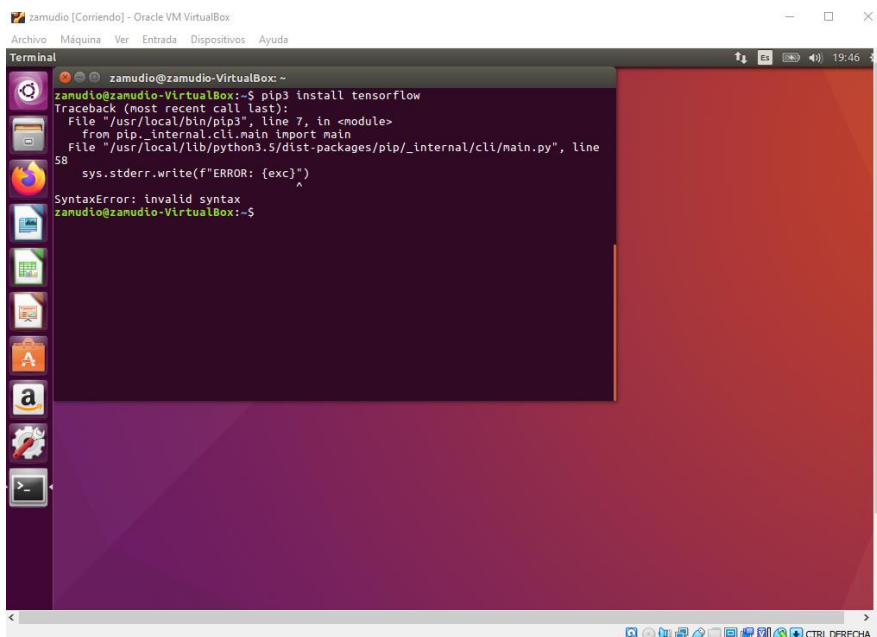
Así que como ese código no nos funcionó para actualizarlo, nos dimos a la tarea de investigar alguno que si sirviera y encontramos este



```
zamudio@zamudio-VirtualBox: ~  
Complete output from command python setup.py egg_info:  
Traceback (most recent call last):  
  File "<string>", line 1, in <module>  
    raise RuntimeError("Python version >= 3.7 required.")  
RuntimeError: Python version >= 3.7 required.  
-----  
Command "python setup.py egg_info" failed with error code 1 in /tmp/pip-build-ub  
30j6e723f13c32c245f6ab298219sc53987848c78/pip-21.1.2-py3-none-any.whl (1.5MB)  
You are using pip version 8.1.1, however version 21.1.2 is available.  
You should consider upgrading via the 'pip install --upgrade pip' command.  
zamudio@zamudio-VirtualBox:~$ sudo -H pip3 install --upgrade pip  
[sudo] password for zamudio:  
Collecting pip  
  Downloading https://files.pythonhosted.org/packages/cd/82/04e9aaf603fdbaecb432  
30j6e723f13c32c245f6ab298219sc53987848c78/pip-21.1.2-py3-none-any.whl (1.5MB)  
    100% |#####| 1.6MB 506kB/s  
Installing collected packages: pip  
  Found existing installation: pip 8.1.1  
  Not uninstalling pip at /usr/lib/python3.5/dist-packages, outside environment  
  /usr  
Successfully installed pip-21.1.2  
zamudio@zamudio-VirtualBox:~$
```

funcionó bien y ahora si ya teníamos la versión 21.1.2 de pip, que es la necesaria para instalar tensorflow.

Entonces con el código que ya habíamos intentado en un inicio, tratamos de nuevo instalar tensorflow y nos dio este error del que ya no supimos cómo solucionarlo,



```
zamudio@zamudio-VirtualBox: ~  
zamudio@zamudio-VirtualBox:~$ pip3 install tensorflow  
Traceback (most recent call last):  
  File "/usr/local/bin/pip3", line 7, in <module>  
    from pip._internal.cli.main import main  
  File "/usr/local/lib/python3.5/dist-packages/pip/_internal/cli/main.py", line  
58 sys.stderr.write(f"ERROR: {exc}")  
SyntaxError: invalid syntax  
zamudio@zamudio-VirtualBox:~$
```

Solo nos decía que era error de sintaxis, pero si vemos capturas anteriores, es el mismo código que se probó en un inicio, se probó reiniciando el equipo, pero el error seguía apareciendo.

5. Resultados Obtenidos

Nuestros resultados no fueron exitosos ya que el programa no lo pudimos instalar ya que nos marca errores y desde el principio nos dice que tenemos que tener una buena tarjeta gráfica y debe instalarse en Ubuntu versión 16.04.

Investigamos los resultados que debíamos tener y son los siguientes

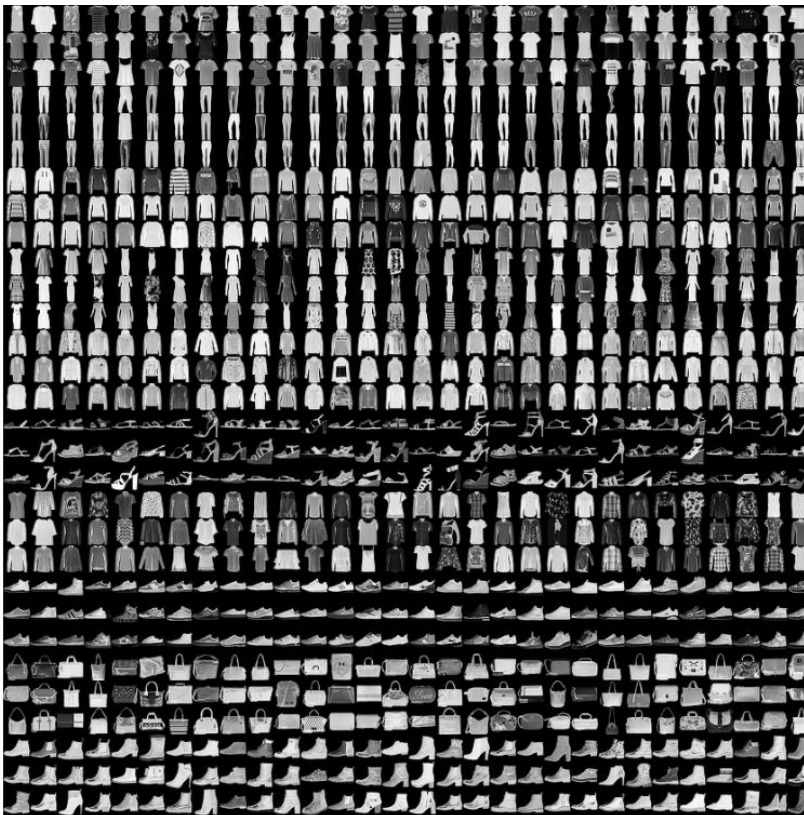
Este programa entrena una red neural para clasificar imágenes de ropa como, tennís y camisetas.

```
# TensorFlow y tf.keras
import tensorflow as tf
from tensorflow import keras

# Librerías de ayuda
import numpy as np
import matplotlib.pyplot as plt

print(tf.__version__)
```

Para mostrar los resultados que debimos obtener, utilizamos los datos de Fashion MNIST el cual contiene más de 70,000 imágenes en 10 categorías, los cuales mostraran artículos individuales de ropa a una resolución baja (28 por 28 píxeles)



Se puede acceder al set de moda de MNIST directamente desde TensorFlow. Para importar y cargar el set de datos de MNIST directamente de TensorFlow

```
fashion_mnist = keras.datasets.fashion_mnist

(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
32768/29515 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26427392/26421880 [=====] - 1s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
8192/5148 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4423680/4422102 [=====] - 0s 0us/step
```

Al cargar el set de datos retorna cuatro arreglos en NumPy:

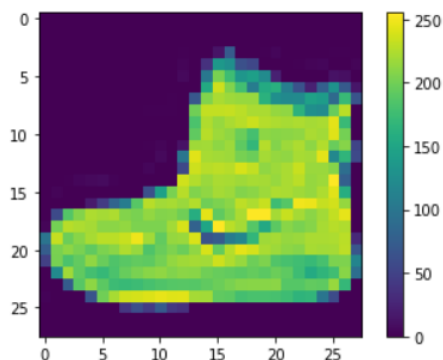
- El arreglo train_imagenes y train_labels son los arreglos que training set – el modelo de datos usa para aprender
- El modelo es probado contra los arreglos test set, el test_imagenes, y test_labels

Cada imagen es mapeada a una única etiqueta. Ya que los Class names no están incluido en el dataset, almacénelo acá para usarlos luego cuando se visualicen las imágenes:

```
class_names = ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat',
               'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']
```

El set de datos debe ser pre-procesada antes de entrenar la red. Si usted inspecciona la primera imagen en el set de entrenamiento, va a encontrar que los valores de los pixeles estan entre 0 y 255:

```
plt.figure()
plt.imshow(train_images[0])
plt.colorbar()
plt.grid(False)
plt.show()
```



Para verificar que el set de datos está en el formato adecuado y que están listos para construir y entrenar la red, vamos a desplegar las primeras 25 imágenes del training set y despleguemos el nombre de cada clase debajo de cada imagen.

```
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(train_images[i], cmap=plt.cm.binary)
    plt.xlabel(class_names[train_labels[i]])
plt.show()
```



6. Conclusiones

*Santamaria

Al finalizar el proyecto logre aprender lo que es el uso de varios ejecutores que no solo existe uno o unos cuantos si no existen más que están ya sin utilizar y que como en el caso de TensorFlow ya no tiene actualizaciones en su software por lo cual uno ya no puede hacer uso de este aunque por la investigación que se realizo es interesante ver cómo trabaja una de sus desventajas es que pide muchos programas para que este funcione y una calidad de tarjeta gráfica, pude ver que su programación es distinta a las que ya conocía como Python es distinto, para mí fue un excelente proyecto ya que nos adentramos un poco más en el mundo de la compilación

*Zamudio

Como pudimos ver el proyecto una red neural está compuesta por muchos elementos trabajando todos ellos en conjunto para dar como resultado final una IA capas de aprender por si misma; esto la convierte en una de las disciplinas dentro de la computación más interesante y compleja la vez con posibilidades infinitas que van desde un ejercicio como el que vimos en donde acomoda las imágenes de acuerdo a su color, tamaño y forma, hasta ¿Por qué no? Pensar en un sistema capas de revolucionar el mundo como lo conocemos

7. Fuentes de Consulta

<https://askubuntu.com/questions/726539/sudo-apt-get-install-libstdc>

<https://github.com/tensorflow/runtime>

<https://apt.llvm.org/>

https://www.tensorflow.org/tutorials/keras/classification?hl=es_419#importar_el_set_de_datos_de_moda_de_mnist

<https://docs.bazel.build/versions/main/install-ubuntu.html>

<https://www.youtube.com/watch?v=Ejzubp-B83o>

<https://github.com/tensorflow/runtime>

https://www.tensorflow.org/tutorials/keras/classification?hl=es_419#importar_el_set_de_datos_de_moda_de_mnist

<https://www.tensorflow.org/tutorials?hl=es-419>

<https://github.com/tensorflow/community>

<https://llvm.org/devmtg/2019-04/slides/Keynote-ShpeismanLattner-MLIR.pdf>