

Instalación completa de Geant4 y ROOT

En este repositorio se muestra el proceso detallado para realizar una instalación completa e integrada entre Geant4 y ROOT. Se dirige a usuarios noveles y medios. El proceso funciona para las siguientes distribuciones basadas en *apt* y en *pacman*.

- **Debian (10), Ubuntu (18.04) Linux Mint (19.2)**

Estas distribuciones hacen uso de `apt` para gestionar los paquetes. Para instalar un paquete nuevo el comando es el siguiente:

```
$ sudo apt install nombre_paquete
```

- **Manjaro (18.0.4)**

Estas distribuciones hacen uso de `pacman` para gestionar los paquetes. Para instalar un paquete nuevo el comando es el siguiente:

```
$ sudo pacman -S nombre_paquete
```

El esquema final de la instalación será el siguiente:

Esquema general de directorios

Documents/MIENTORNO	DIRECTORIO BASE
Geant4-10.05	Directorio de instalación de geant4
▪ g4_source	Código fuente geant4
▪ data	Datos adicionales de geant4
▪ g4_build	Compilar código fuente
▪ g4_install	Instalación final de geant4
root-6.18	Directorio de instalación de root
▪ root-6.18.00	Código fuente root
▪ root-build	Instalación final de root

Geant4 10.05.p01

Geant4 es un "toolkit" (caja de herramientas) para la simulación del paso de partículas a través de la materia. Su proceso de instalación no es trivial, fácilmente puede disuadir a usuarios poco experimentados, pero constituye un paso importante para ingresar a mundo de la simulación. El proceso de instalación que usaremos será el siguiente:

- Preparación del sistema
- Instalación para apt systems y pacman systems
- Instalación para pacman systems

Preparación del sistema

1. ACTUALIZAR la distribución de Linux a la versión más reciente:

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

2. INSTALAR DEPENDENCIAS

Los paquetes han sido revisados con la base de paquetes de [Ubuntu](#).

Librerías necesarias para Geant4 y ROOT:

```
$ sudo apt install libxerces-c-dev mesa-utils mesa-utils-extra mesa-common-dev libfontconfig libfontconfig-dev libxmu-dev qt4-default libqt4-opengl libqt4-opengl-dev qt5-default libqt5opengl5 libqt5opengl5-dev
```

```
$ sudo apt install cmake cmake-qt-gui g++ gcc gfortran binutils libx11-dev libxpm-dev libxft-dev libxext-dev libpng-dev libpng++-dev libjpeg-dev
```

Librerías importantes (opcionales):

```
$ sudo apt install git libssl-dev libpcrc3-dev libftgl-dev libmysqlclient-dev libfftw3-dev libcfitsio-dev graphviz-dev libavahi-compat-libdnssd-dev libldap2-dev python-dev libxml2-dev libkrb5-dev libgsl23 libgsl-dev
```

Instalación Geant4

- [Distribuciones: Debian/Ubuntu/Mint](#)
- **Distribuciones Arch/Manjaro**

Las distribuciones Arch Linux se caracterizan por estar a la vanguardia del software, esto puede suponer ventajas o desventajas según el uso al que estén destinadas. El proceso de instalación de geant4 en estas plataformas es esencialmente el mismo que para *Debian* y derivadas. La única diferencia consiste en buscar los paquetes para `pacman`. Para realizar la búsqueda de los paquetes se recomienda utilizar el *gestor de software* propio de esas distribuciones **pamac** u **octopi**. Los comandos para fijar variables de entorno y demás son exactamente iguales.

Nota.- (pacman = terminal, pamac = interfaz gráfica de pacman).

Se puede comprobar los paquetes directamente en página oficial de Arch.

- <https://www.archlinux.org/packages/>

ROOT 6.18.00

Logo ROOT

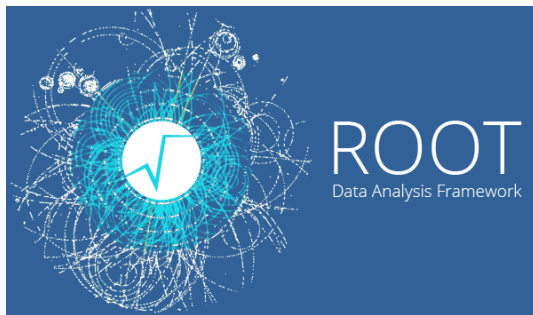
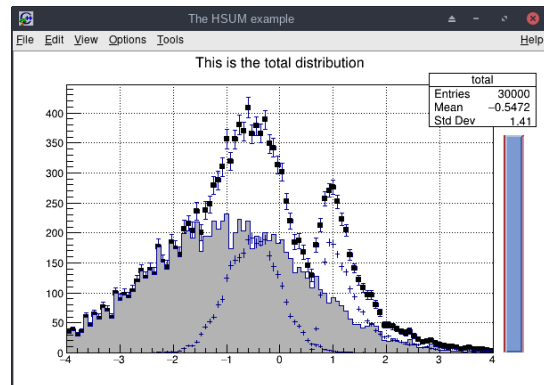


Imagen .demo de ROOT



Preparación del sistema

En general, una vez instalado Geant4, la instalación de ROOT es relativamente sencilla. En caso de solamente requerir ROOT, el proceso de resume a continuación:

- Actualizar el sistema
- Instalar dependencias

```
$ sudo apt install cmake cmake-qt-gui g++ gcc gfortran binutils libx11-dev libxpm-dev libxft-dev libxext-dev libpng-dev libpng++-dev libjpeg-dev
```

- Proceder a la instalación eligiendo una de las dos formas:
 - Desde el código fuente
 - Desde paquete pre-compilado

Instalación desde el código fuente

La instalación de ROOT desde el código fuente es tradicionalmente la mejor opción.

Ventajas e inconvenientes de compilar:

- El software aprovechará toda la potencia que brinde el ordenador.
- Es especialmente útil cuando se cuenta con un ordenador con muchos núcleos e hilos de ejecución.
- Útil cuando se quiere que el software se comporte de forma optimizada en ordenadores modestos.
- Es un proceso largo y algo tedioso.

Instalación pre-compilada

La instalación pre-compilada consiste en una instalación rápida. En esta modalidad, el software ha sido compilado en otro ordenador bajo ciertas condiciones (dependencias) que tienen que cumplirse en el ordenador de destino para que se ejecute.

Ventajas e inconvenientes de usar pre-compilado:

- Es es una instalación relativamente rápida.
- Es la mejor opción para usuarios nóveles porque no requiere compilar desde el código fuente.
- Si contamos con un ordenador potente en número de núcleos e hilos no se aprovecha todos los recursos. Solo se nota al realizar análisis de datos de gran complejidad.

Instalación de Geant4 10.05.p01

Instalación desde el código fuente

La instalación final seguirá el siguiente esquema de directorios:

Esquema general de directorios

Documents/MIENTORNO	DIRECTORIO BASE
Geant4-10.05	Directorio de instalación de geant4
▪ g4_source	Código fuente geant4
▪ data	Datos adicionales de geant4
▪ g4_build	Compilar código fuente
▪ g4_install	Instalación final de geant4
root-6.18	Directorio de instalación de root
▪ root-6.18.00	Código fuente root
▪ root-build	Instalación final de root

1. Descargar el código fuente desde la página oficial de [Geant4](#)

Descargar la última versión estable: [Geant4 10.05.p01](#)

2. **Descargar** los 12 **paquetes adicionales** de la sección: *DATA FILES* ().

Son necesarios para ejecutar los ejemplos que trae Geant4. Son en total 12 archivos: G4NDL4.5, G4EMLOW7.7, G4PhotonEvaporation5.3, etc., (Aproximadamente 2.1 GB) se puede omitir este paso si se posee una conexión a Internet muy estable en latencia además velocidad de descarga.

3. **Descomprimir** el código fuente y renombrar la carpeta de la siguiente forma:

```
geant4.10.05.p01 --- a ---> g4_source
```

4. Prestar atención al esquema de instalación.

Puedes cambiar libremente el nombre del directorio base, aquí: **SIMULAR**

5. **Mover** la carpeta **g4_source** a la ruta de instalación:

```
/home/USUARIO/Documents/SIMULAR/Geant4-10.05/
```

6. **Descomprimir** los archivos adicionales dentro de una carpeta denominada: **data**

7. **Mover** la carpeta **data** a la ruta de instalación:

```
/home/USUARIO/Documents/SIMULAR/Geant4-10.05/
```

8. Dentro de la ruta de instalación crear dos directorios adicionales:

- **geant_build**
- **geant_install**

9. En este punto debemos tener el siguiente lo siguiente:

```
$ ls
```

```
output:
data  g4_build  g4_install  g4_source  geant4.10.05.p01.tar.gz
```

10. Configurar `cmake`. Utilizaremos el modo gráfico:

```
$ cmake-gui
```

En la interfaz seleccionar lo siguiente:

- Source code = /home/USUARIO/Documents/SIMULAR/Geant4/g4_source
- Build binaries = /home/USUARIO/Documents/SIMULAR/Geant4/g4_build
- Clic en *configurar*
- Clic en *finish*

11. Activar características de Geant4 (según se requiera)

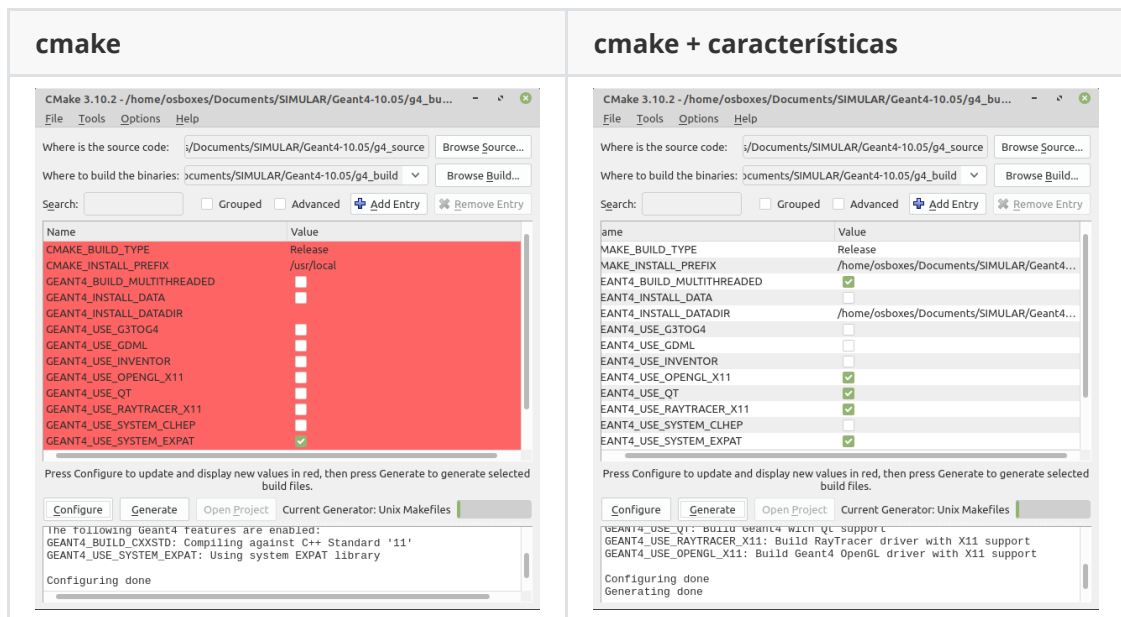
Recomiendo instalar las siguientes opciones, son útiles en la mayoría de situaciones, sin embargo, depende de la finalidad del usuario, los detalles se pueden ver en [aquí](#)

- ☒ MAKE_INSTALL_PREFIX = /home/USUARIO/Documents/SIMULAR/Geant4/g4_build
- ☒ GEANT4_BUILD_MULTITHREADED
- ☒ GEANT4_INSTALL_DATA

- ☒ GEANT4_INSTALL_DATADIR = /home/USUARIO/Documents/SIMULAR/Geant4/data
- ☒ GEANT4_USE_OPENGL_X11
- ☒ GEANT4_USE_QT
- ☒ GEANT4_USE_RAYTRACER_X11
- ☒ GEANT4_USE_SYSTEM_EXPAT

12. Clic en **Configure** dos veces si fuera necesario (no se debe ver marcadores rojos)

13. Clic en **Generate** y cerrar cmake.



14. Abrir una terminal en el directorio g4_build

```
$ cd /home/USUARIO/Documents/SIMULAR/Geant4-10.05/g4_build/
```

15. Compilar

```
$ make -j4
```

Aquí `j4` indica el número de núcleos que asignaremos a la tarea. A mayor cantidad de núcleos menor tiempo en realizar la tarea y viceversa. Depende de tu ordenador. Este proceso puede tomar 2 horas o más.

```
osboxes@osboxes: ~/Documents/SIMULAR/Geant4-10.05/g4_build
File Edit View Search Terminal Help
osboxes@osboxes:~/Documents/SIMULAR/Geant4-10.05/g4_build$ make -j2
Scanning dependencies of target G4zlib
Scanning dependencies of target G4clhep
[ 0%] Building C object source/externals/zlib/CMakeFiles/G4zlib.dir/src/adler32.c.o
[ 0%] Building CXX object source/externals/clhep/CMakeFiles/G4clhep.dir/src/AxisAngle.cc.o
[ 0%] Building C object source/externals/zlib/CMakeFiles/G4zlib.dir/src/compress.c.o
[ 0%] Building C object source/externals/zlib/CMakeFiles/G4zlib.dir/src/crc32.c.o
[ 0%] Building C object source/externals/zlib/CMakeFiles/G4zlib.dir/src/deflate.c.o
[ 0%] Building CXX object source/externals/clhep/CMakeFiles/G4clhep.dir/src/BasicVector3D.cc.o
[ 0%] Building C object source/externals/zlib/CMakeFiles/G4zlib.dir/src/gzclose.c.o
[ 0%] Building C object source/externals/zlib/CMakeFiles/G4zlib.dir/src/gzlib.c.o
[ 0%] Building C object source/externals/zlib/CMakeFiles/G4zlib.dir/src/gzread.c.o
```

16. Instalar Geant4

```
$ make install
```

17. Fijar al variables de entorno (temporal)

Esto método se tiene que hacer cada vez que se cierre la terminal. Caso contrario, se tiene que fijar las variables de forma persistente (recomendado).

```
$ source /home/USUARIO/Documents/SIMULAR/Geant4-10.05/g4_install/bin/geant4.sh
```

O entrar al directorio de instalación dentro de la carpeta **bin**

```
$ source geant4.sh
```

18. Listo! Geant4 esta correctamente instalado.

Configurar entorno de variables (persistente)

El proceso es el siguiente:

- Abrir el explorador de archivos en la siguiente ruta:

```
/home/Usuario/
```

- Presionar **Ctrl + H** para ver los archivos ocultos.
- Localizar el archivo **.bashrc**
- Abrir **.bashrc** con un editor de texto, ir al final del todo y pegar lo siguiente:
(Cambia USUARIO por el tuyo)

```
# Entorno de Geant4 y ROOT
source /home/USUARIO/Documents/SIMULAR/Geant4-10.05/g4_install/bin/geant4.sh
```

- Guardamos, salimos y presionar nuevamente `Crtl + H` para regresar todo a la normalidad.
- Ahora se puede ejecutar *geant4* desde cualquier parte del ordenador cuando se necesite.

Ejecutar una simulación

1. **Crear un directorio de trabajo** dentro del directorio base

```
$ cd /home/USUARIO/Documents/SIMULAR/
$ mkdir g4_work
```

2. **Copiar un ejemplo** de Geant4 en el directorio de trabajo

Los ejemplos se encuentran en:

```
/home/USUARIO/Documents/SIMULAR/geant4-10.5.1/g4_install/share/Geant4-10.5.1/examples/
```

3. En mi caso he copiado el ejemplo: **/Basic/B1**
4. Dentro del directorio del ejemplo hacemos lo siguiente:

```
$ mkdir build
$ cd build
```

5. Construimos el paquete con `cmake`

```
$ cmake ../
```

6. Compilar

```
$ make -j2
```

7. Listar los objetos del directorio y debemos ver un ejecutable

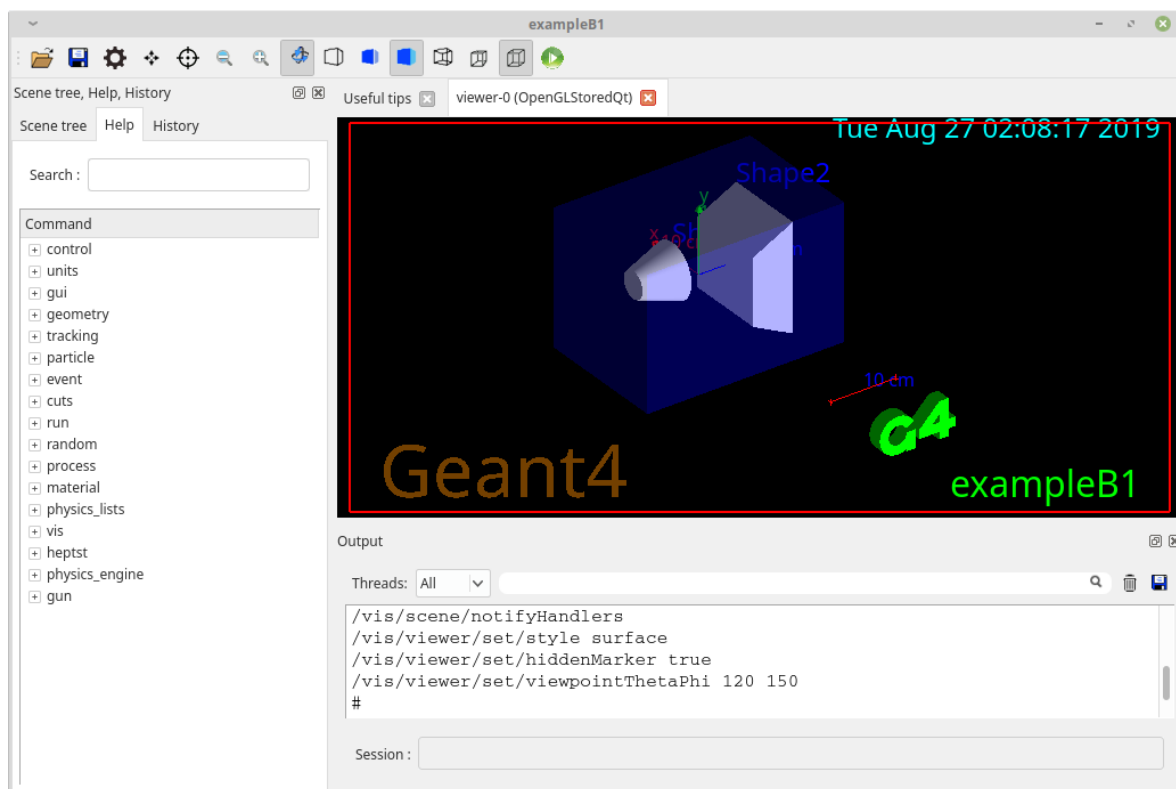
```
$ ls
```

```
output:
CMakeCache.txt  cmake_install.cmake  exampleB1.in  G4History.macro
Makefile run2.mac
CMakeFiles      exampleB1            exampleB1.out  init_vis.mac
run1.mac  vis.mac
```

8. Ejecutar la simulación

```
$ ./exampleB1
```


Resultado



Instalación de ROOT 6.18

Este proceso de instalación recomienda utilizar el *gestor de archivos* para moverse entre los distintos directorios. La instalación se realiza según el esquema que se muestra en el esquema (todos los directorios son carpetas).

Esquema general de directorios

Documents/MIENTORNO	DIRECTORIO BASE
Geant4-10.05	Directorio de instalación de geant4
▪ g4_source	Código fuente geant4
▪ data	Datos adicionales de geant4
▪ g4_build	Compilar código fuente
▪ g4_install	Instalación final de geant4
root-6.18	Directorio de instalación de root
▪ root-6.18.00	Código fuente root
▪ root-build	Instalación final de root

Pasos a seguir:

1. ACTUALIZAR la distribución de Linux a la versión más reciente:

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

2. INSTALAR DEPENDENCIAS

La lista detallada se puede revisar aquí [ROOT](#). Los paquetes han sido cuidadosamente revisados de acuerdo a base de datos de paquetes de [Ubuntu/bionic](#)

Librerías necesarias (sin esto no funciona)

```
$ sudo apt install cmake cmake-qt-gui g++ gcc gfortran binutils libx11-dev  
libxpm-dev libxft-dev libxext-dev libpng-dev libpng++-dev libjpeg-dev
```

Librerías importantes (opcionales)

```
$ sudo apt install git libssl-dev libpcre3-dev libftgl-dev libmysqlclient-  
dev libfftw3-dev libcfitsio-dev graphviz-dev libavahi-compat-libdnssd-dev  
libldap2-dev python-dev libxml2-dev libkrb5-dev libgsl23 libgsl-dev
```

3. **Descargar** los binarios desde la página oficial de [ROOT](#) . Seleccionar la última versión estable (PRO).

[root v6.18.00.source.tar.gz](#), (158 MB)

4. **Mover** el archivo descargado al directorio de instalación:

```
/home/USUARIO/Documents/MIENTORNO/root-6.18/
```

5. **Descomprimir** el archivo `.tar.gz` para obtener una carpeta llamada: `root`. ¡Renombrar! ese directorio a **`root-6.18.00`** (para evitar problemas con los ficheros de Ubuntu)

- renombrar: `root --- a ---> root-6.18.00`

6. Crear el directorio **`root-build`**.

Comprobar que los directorios coincidan con el esquema.

7. Abrir una terminal y cambiar al directorio actual: **`root-build`**

```
$ cd /home/USUARIO/Documents/ENTORNO/root-6.18/root-build/
```

8. Ejecutar `cmake` (Prepara la compilación)

```
$ cmake ../root-6.18.00/
```

9. Compilar

Aquí `j4` significa que se compilará usando 4 núcleos del procesador, depende de la potencia de tu ordenador. Mientras más núcleos físicos poseas menos tiempo tomará el proceso y viceversa.

```
$ make -j4
```

En mi caso, usando usando 4 núcleos el proceso tardó: 1h y 30 min.

```
Terminal - jonasdx@T-Satellite:~/Documents/MIENTORNO/root-6.18/root-build
File Edit View Terminal Tabs Help
[ 26%] Copying header /home/jonasdx/Documents/MIENTORNO/root-6.18/root-6.18.00/roofit/roostats/inc/RooSta
ts/SimpleInterval.h to /home/jonasdx/Documents/MIENTORNO/root-6.18/root-build/include
[ 26%] Copying header /home/jonasdx/Documents/MIENTORNO/root-6.18/root-6.18.00/roofit/roostats/inc/RooSta
ts/SimpleLikelihoodRatioTestStat.h to /home/jonasdx/Documents/MIENTORNO/root-6.18/root-build/include
[ 26%] Copying header /home/jonasdx/Documents/MIENTORNO/root-6.18/root-6.18.00/roofit/roostats/inc/RooSta
ts/TestStatistic.h to /home/jonasdx/Documents/MIENTORNO/root-6.18/root-build/include
[ 26%] Copying header /home/jonasdx/Documents/MIENTORNO/root-6.18/root-6.18.00/roofit/roostats/inc/RooSta
ts/TestStatSampler.h to /home/jonasdx/Documents/MIENTORNO/root-6.18/root-build/include
[ 26%] Copying header /home/jonasdx/Documents/MIENTORNO/root-6.18/root-6.18.00/roofit/roostats/inc/RooSta
ts/ToyMCImportanceSampler.h to /home/jonasdx/Documents/MIENTORNO/root-6.18/root-build/include
[ 26%] Copying header /home/jonasdx/Documents/MIENTORNO/root-6.18/root-6.18.00/roofit/roostats/inc/RooSta
ts/ToyMCSampler.h to /home/jonasdx/Documents/MIENTORNO/root-6.18/root-build/include
[ 26%] Copying header /home/jonasdx/Documents/MIENTORNO/root-6.18/root-6.18.00/roofit/roostats/inc/RooSta
ts/ToyMCStudy.h to /home/jonasdx/Documents/MIENTORNO/root-6.18/root-build/include
[ 26%] Copying header /home/jonasdx/Documents/MIENTORNO/root-6.18/root-6.18.00/roofit/roostats/inc/RooSta
ts/UniformProposal.h to /home/jonasdx/Documents/MIENTORNO/root-6.18/root-build/include
[ 26%] Copying header /home/jonasdx/Documents/MIENTORNO/root-6.18/root-6.18.00/roofit/roostats/inc/RooSta
ts/UpperLimitMCSModule.h to /home/jonasdx/Documents/MIENTORNO/root-6.18/root-build/include
[ 26%] Built target move_header_roofit_roostats
Scanning dependencies of target obj.clang-tblgen
[ 26%] Building CXX object interpreter/llvm/src/tools/clang/utils/TableGen/CMakeFiles/obj.clang-tblgen.dir/
r/ClangASTNodesEmitter.cpp.o
-- AFTERIMAGE configure command succeeded. See also /home/jonasdx/Documents/MIENTORNO/root-6.18/root-bui
ld/AFTERIMAGE-prefix/src/AFTERIMAGE-stamp/AFTERIMAGE-configure-*.log
[ 26%] Performing build step for 'AFTERIMAGE'
[ 26%] Building CXX object interpreter/llvm/src/tools/clang/utils/TableGen/CMakeFiles/obj.clang-tblgen.dir/
r/ClangAttrEmitter.cpp.o
```

10. Fijar las variables de entorno

```
$ source /bin/thisroot.sh
```

11. Ya tenemos instalado ROOT. Ejecutar:

```
$ root
```

ROOT de forma persistente (source)

Con el proceso anterior ya se tiene correctamente instalado *ROOT*, sin embargo, es necesario que fijemos las variables de entorno cada vez que cerremos la terminal. La solución es fijar *ROOT* de forma persistente.

El proceso es el siguiente:

- Abrir el explorador de archivos en la siguiente ruta:

```
/home/Usuario/
```

- Presionar **Ctrl + H** para ver los archivos ocultos.
- Localizar el archivo `.bashrc`
- Abrir `.bashrc` con un editor de texto, ir al final del todo y pegar lo siguiente:

(Cambia USUARIO por el tuyo)

```
# Entorno de Geant4 y ROOT
source /home/USUARIO/Documents/MIENTORNO/root-6.18/root-
build/bin/thisroot.sh
```

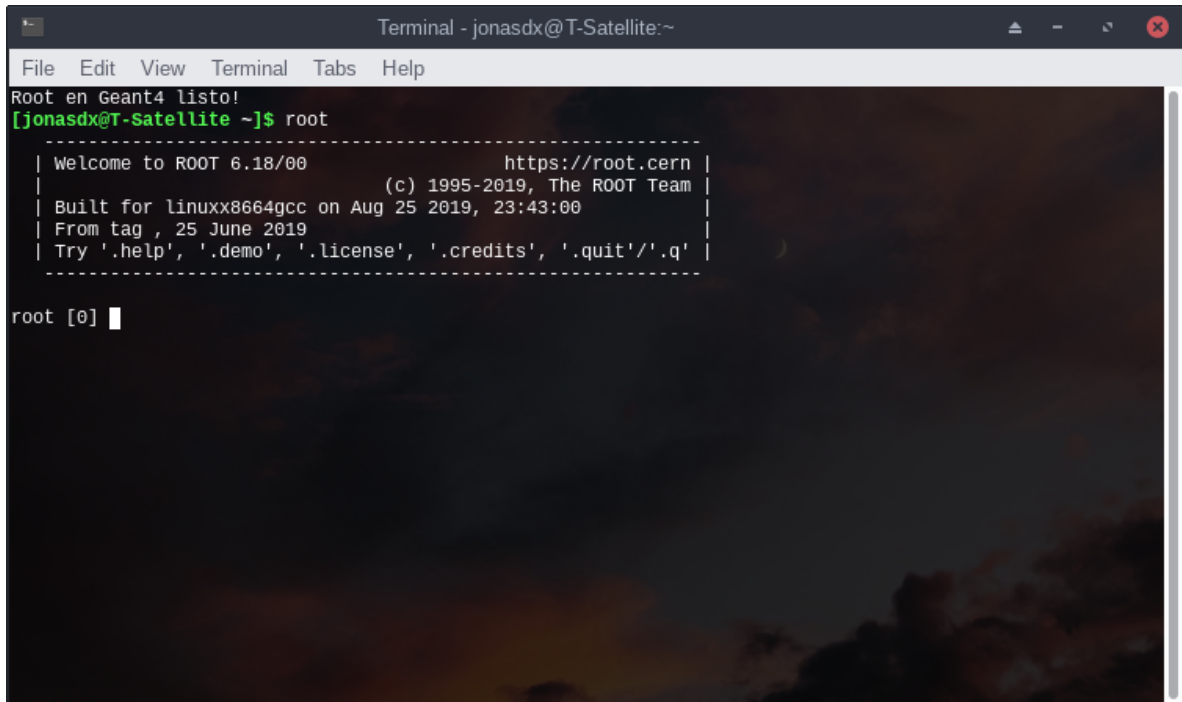
- Guardamos, salimos y presionar nuevamente **Ctrl + H** para regresar todo a la normalidad.
- Ahora podemos ejecutar *root* desde cualquier parte del ordenador

```
$ root
```

Resultado

Se sabe que el programa ha sido compilado desde el código fuente, por el tag:

```
Built for linuxx8664gcc on Aug 25 2019.  
From tag, 25 June 2019
```



```
Terminal - jonasdx@T-Satellite:~  
File Edit View Terminal Tabs Help  
Root en Geant4 listo!  
[jonasdx@T-Satellite ~]$ root  
-----  
| Welcome to ROOT 6.18/00                               https://root.cern |  
|                                     (c) 1995-2019, The ROOT Team |  
| Built for linuxx8664gcc on Aug 25 2019, 23:43:00 |  
| From tag , 25 June 2019 |  
| Try '.help', '.demo', '.license', '.credits', '.quit'/'.q' |  
-----  
root [0]
```

Adicional

Vincular *ROOT* con *Geant4* para simulaciones

Esto es especialmente útil cuando se requiere que la simulación llame a ROOT para llenar un histograma directamente, etc.

El procedimiento es el siguiente:

1. Movernos a la ruta de instalación de root y crear un archivo nuevo llamado: `root.sh`, así:

```
$ cd /home/user/Documents/MIENTORNO/root-6.18/  
$ touch script_root.sh
```

2. Abrir el archivo que hemos creado y pegar dentro lo siguiente:

```
#Mensaje para mostrar en la terminal
echo Root en Geant4 listo!

#Para usar ROOT con Geant4 en algunas aplicaciones
export G4ROOT_USE=1
```

3. Guardar el archivo y cerrar.
4. Hacer un **source** (persistente) del script en `.bashrc` como en el caso anterior.

```
source /home/user/Documents/MIENTORNO/root-6.18/script_root.sh
```

5. Ejecutar root

```
$ root
```

Instalación pre-compilada de ROOT 6.18

Este proceso de instalación se recomienda para usuarios noveles o siempre que no se desee hacer un uso intensivo de ROOT. Para otros fines es mejor utilizar el proceso de [compilación](#).

Esquema general de directorios

Documents/MIENTORNO	DIRECTORIO BASE
Geant4-10.05	Directorio de instalación de geant4
▪ g4_source	Código fuente geant4
▪ data	Datos adicionales de geant4
▪ g4_build	Compilar código fuente
▪ g4_install	Instalación final de geant4
root-6.18	Directorio de instalación de root
▪ root-6.18.00	Código fuente root
▪ root_build	Instalación final de root

Pasos a seguir:

1. ACTUALIZAR la distribución de Linux a la versión más reciente:

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

2. INSTALAR DEPENDENCIAS

La lista detallada se puede revisar aquí [ROOT](#). Los paquetes han sido cuidadosamente revisados de acuerdo a base de datos de paquetes de [Ubuntu/bionic](#).

Librerías necesarias (sin esto no funciona)

```
$ sudo apt install cmake cmake-qt-gui g++ gcc gfortran binutils libx11-dev  
libxpm-dev libxft-dev libxext-dev libpng-dev libpng++-dev libjpeg-dev
```

Librerías importantes (opcionales)

```
$ sudo apt install git libssl-dev libpcre3-dev libftgl-dev libmysqlclient-  
dev libfftw3-dev libcfitsio-dev graphviz-dev libavahi-compat-libdnssd-dev  
libldap2-dev python-dev libxml2-dev libkrb5-dev libgsl23 libgsl-dev
```

3. Descargar los binarios compilados (**Binary distributions**) desde la página oficial de [ROOT](#).
Seleccionar la versión que mejor se adecue a su distribución de Linux. En el caso de las
listadas al inicio, la versión que mejor ajusta las dependencias es:

Plataforma	Paquete
Ubuntu_18_gcc7.4	root v6.18.02.Linux-ubuntu18-x86_64-gcc7.4.tar.gz

4. Mover el archivo descargado al directorio de instalación:

```
/home/USUARIO/Documents/MIENTORNO/root-6.18/
```

5. Descomprimir el archivo .tar.gz para obtener una carpeta llamada: *root*. ¡Renombrar! ese
directorio a **root-6.18.00** (para evitar problemas con los ficheros de Ubuntu)

- renombrar: *root* --- *a* ---> *root-6.18.00*

6. Listar los archivos del directorio actual.

```
$ ls
```

output:

```
$ root-6.18.00 root_v6.18.02.Linux-ubuntu18-x86_64-gcc7.4.tar.gz
```

7. Fijamos las variables de entorno (temporales)

```
$ source /root-6.18.00/bin/thisroot.sh
```

8. Ejecutar root:

```
$ root
```

ROOT de forma persistente (source)

Con el proceso anterior ya se tiene correctamente instalado *ROOT*, sin embargo, es necesario que
fijemos las variables de entorno cada cerremos la terminal. La solución es fijar *ROOT* de forma
persistente.

El proceso es el siguiente:

- Abrir el explorador de archivos en la siguiente ruta:

```
/home/Usuario/
```

- Presionar `Ctrl + H` para ver los archivos ocultos.
- Localizar el archivo `.bashrc`
- Abrir `.bashrc` con un editor de texto. Al final del todo y pegar lo siguiente:

(Cambia USUARIO por el tuyo)

```
# Entorno de Geant4 y ROOT
source /home/USUARIO/Documents/MIENTORNO/root-6.18/root-
6.18.00/bin/thisroot.sh
```

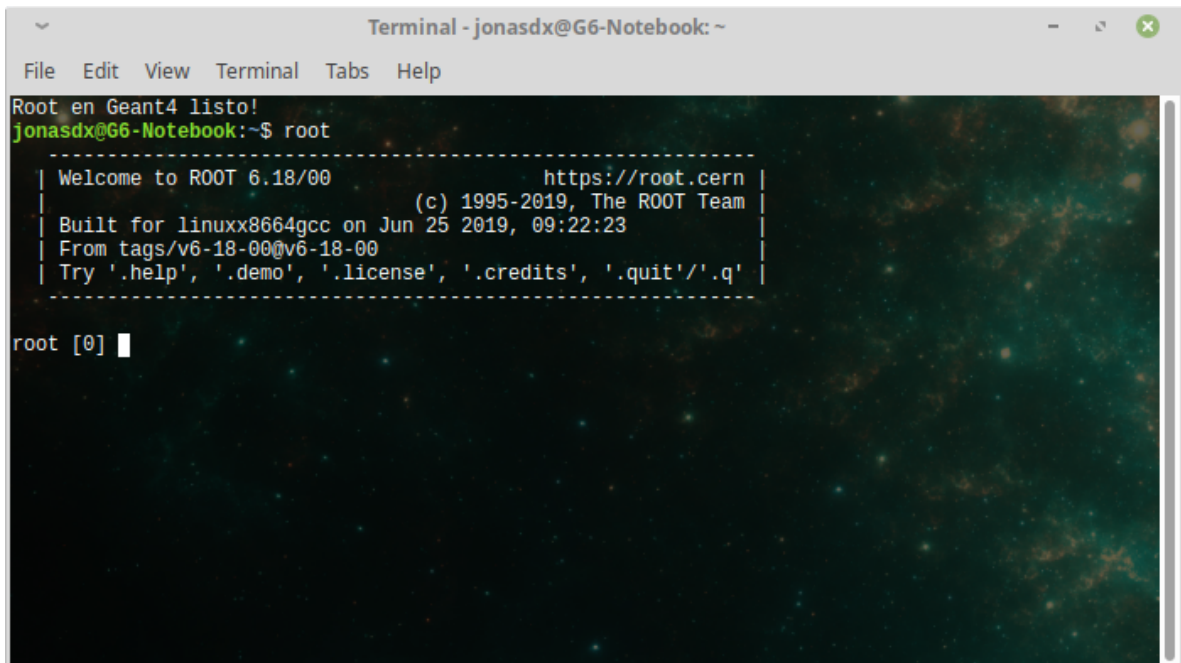
- Guardamos, salimos y presionar nuevamente `Ctrl + H` para regresar todo a la normalidad.
- Ahora podemos ejecutar `root` desde cualquier parte del ordenador.

```
$ root
```

Resultado

Se sabe que el programa ha sido compilado externamente por el tag:

```
Build for linuxx8664gcc on Jun 25 2019, 09:22:23
From tags/v6-18-00@v6-18-00
```



```
Terminal - jonasdx@G6-Notebook: ~
File Edit View Terminal Tabs Help
Root en Geant4 listo!
jonasdx@G6-Notebook:~$ root
-----
Welcome to ROOT 6.18/00                                https://root.cern
(c) 1995-2019, The ROOT Team
Built for linuxx8664gcc on Jun 25 2019, 09:22:23
From tags/v6-18-00@v6-18-00
Try '.help', '.demo', '.license', '.credits', '.quit'/'.'
-----
root [0] █
```

Fuentes y Recursos:

- [Página oficial Geant 4](#)
- [Guía de instalación oficial de Geant4](#)
- [Página oficial de ROOT](#)

- [ROOT Documentation](#)
- [Dependencias Root](#)
- [Paquetes Ubuntu](#)