
Tutorial de Instalação Geant4

Um Guia para Ubuntu

Escrito por:

Letícia Braga da Rosa

Campinas - SP

2020

Sumário

1	Preparando Ambiente de instalação	1
1.1	Criando Estrutura Para Instalação	1
1.2	Baixando Código Fonte e Bibliotecas	1
1.3	Pré Requisitos de Instalação	3
1.3.1	Requisitos de Compilação	3
1.3.2	Dependências Gráficas Externas	4
1.3.3	Interface Gráfica com Usuário Geometria GDML	4
1.3.4	Instalação das Dependências	4
2	Passo a Passo de Instalação	5
2.1	Compilação	5
2.1.1	Passo 1: cmake	5
2.1.2	Passo 2: make	8
2.1.3	Passo 3: make install	8
2.2	Configurando a Bash	10
3	Compilação de Código	11

1 Preparando Ambiente de instalação

A versão do Geant4 que será instalada aqui é a **geant4.10.06.p02**, a mais recente no presente momento de criação deste tutorial. O processo de instalação é para máquinas com sistema operacional **Linux** e aqui usaremos a versão **Ubuntu 18.04**. Este tutorial foi feito com base em **1**. Inicialmente, é preciso criar uma estrutura organizacional para a instalação do Geant4.

1.1 Criando Estrutura Para Instalação

O guia de instalação original do Geant4 pode ser encontrado no link: <http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/InstallationGuide/html/index.html>

É recomendado seguir a seguinte estrutura:

1. Criar um diretório do geant4 em sua pasta de programas na *home*: **/home/Programs/geant4**.
2. Dentro do diretório **geant4** criar os diretórios **zip** e **geant4.10.06.p02**.
3. Dentro do diretório **zip** crie uma pasta **data**.

1.2 Baixando Código Fonte e Bibliotecas

Então, podemos baixar o código fonte do Geant4 e as bibliotecas dos processos físicos. Para fazer o *download* acesse o site: <https://geant4.web.cern.ch/support/download>

1. baixe o arquivo do código fonte em formato zip no caminho **geant4/zip**. Veja figura **1**
2. baixe todos os *data files* disponíveis, no momento são 12 bibliotecas no total, para o caminho **geant4/zip/data**. Veja figura **2**.



Software Download

Geant4 10.6 Software Download

Geant4 10.6
first released 6 December 2019 (patch-02, released 29 May 2020)

The Geant4 source code is freely available. See the [licence conditions](#).

Please read the [Release Notes](#) before downloading or using this release. The patches below contain bug fixes to additional notes for [patch-01](#) and [patch-02](#), or download the complete source with the patch applied; in any case

Source files

Please choose the archive best suited to your system and archiving tool:

Download	GNU or Linux tar format, compressed using gzip (33.2Mb, 34833550 bytes) <i>After downloading, unpack using GNU tar.</i>
Download	ZIP format (47.1Mb, 49361419 bytes) <i>After downloading, unpack using e.g. WinZip.</i>

Figura 1: Em vermelho código fonte do Geant4 para download no site oficial.

Data files (*)

For specific, optional physics processes some of the following files are required. The file format is compatible with Unix, GNU, and Windows utilities.

Download	G4NDL4.6, Neutron data files (with thermal cross-sections) - version 4.6 (572.1Mb, 599862135 bytes) <small>NEW</small>
Download	G4EMLOW7.9.1, data files for low energy electromagnetic processes - version 7.9.1 (114.2Mb, 119756769 bytes) <small>NEW</small>
Download	G4PhotonEvaporation5.5, data files for photon evaporation - version 5.5 (9.6Mb, 10087354 bytes) <small>NEW</small>
Download	G4RadioactiveDecay5.4, data files for radioactive decay hadronic processes - version 5.4 (1.0Mb, 1059968 bytes) <small>NEW</small>
Download	G4SAIDDATA2.0, data files from evaluated cross-sections in SAID data-base - version 2.0 (37.6kb, 38502 bytes)
Download	G4PARTICLEXS2.1, data files for evaluated particle cross-sections on natural composition of elements - version 2.1 (4.4Mb, 4600435 bytes) <small>NEW</small>
Download	G4ABLA3.1, data files for nuclear shell effects in INCL/ABLA hadronic mode - version 3.1 (104.8kb, 107286 bytes)
Download	G4INCL1.0, data files for proton and neutron density profiles in INCL - version 1.0 (93.6kb, 95840 bytes)
Download	G4PII1.3, data files for shell ionisation cross-sections - version 1.3 (4.1Mb, 4293607 bytes)
Download	G4ENSDFSTATE2.2, data files for nuclides properties - version 2.2 (283.8kb, 290632 bytes)
Download	G4RealSurface2.1.1, <i>Optional data files for measured optical surface reflectance</i> - version 2.1.1 (127.2Mb, 133386050 bytes)
Download	G4TENDL1.3.2, <i>Optional data files for incident particles</i> - version 1.3.2 (558.0Mb, 585100935 bytes)

Figura 2: Data files para download no site oficial.

No diretório **zip** você deverá ter o código fonte **geant4_10_0602.zip** e a pasta **data**, dentro da qual estarão todos os data files zipados.

- 3 Descompacte o código fonte dentro do diretório **geant4**. Dentro do diretório **geant4** você deverá ter agora a pasta **geant4_10_06_p02** e a pasta **zip**.
- 4 Renomeie a pasta **geant4_10_06_p02** para **geant4.10.06.p02**.
- 5 Dentro do diretório **geant4.10.06.p02** crie a pasta **data**. Nela, descompacte todos os data files que estão em **geant4/zip/data**.
- 6 Dentro do diretório **geant4.10.06.p02** crie a pasta **geant4.10.06.p02-install** e **geant4.10.06.p02-build**.

No final deste processo o esquema de pasta deverá estar conforme a figura 3.

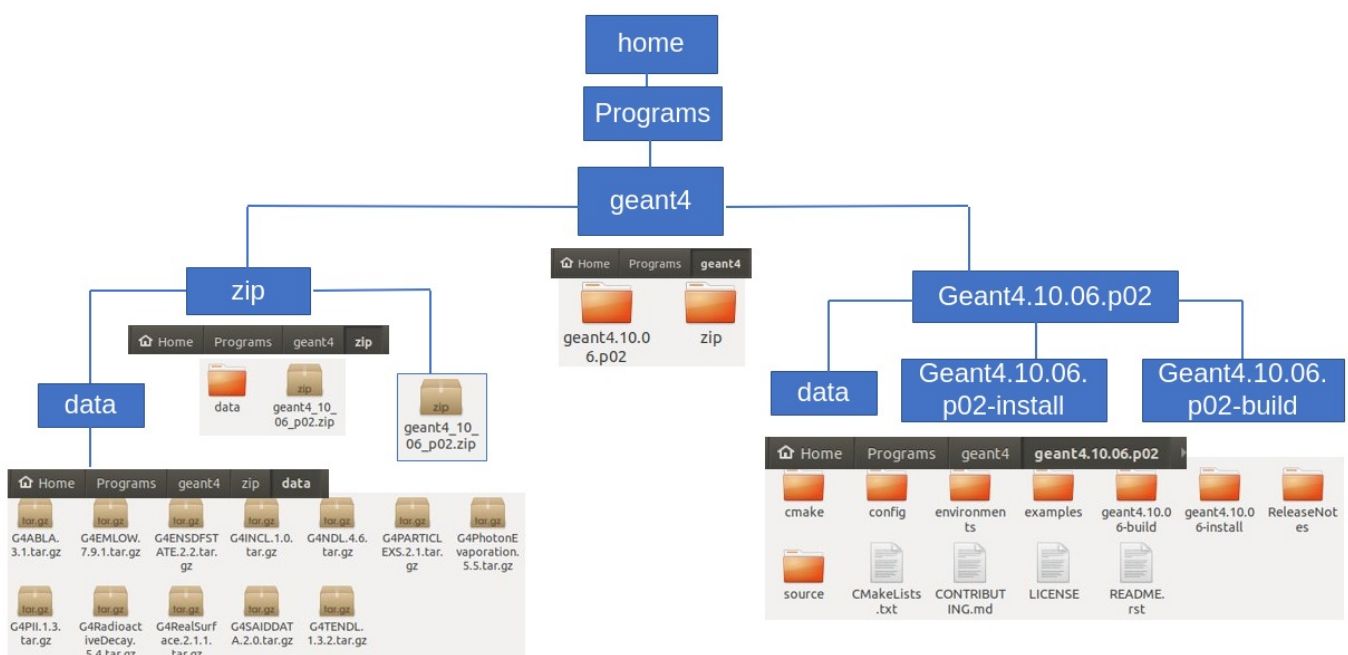


Figura 3: Esquema de pastas para instalação do Geant4.

1.3 Pré Requisitos de Instalação

Agora iremos instalar toda as dependências necessárias para operação do Geant4.

1.3.1 Requisitos de Compilação

Para a compilação do código, precisamos instalar as seguintes dependências:

- git
- dpkg-dev

- cmake
- g++
- gcc
- binutils

1.3.2 Dependências Gráficas Externas

Para as dependências gráficas precisamos instalar:

- libx11-dev
- libxmu-dev
- libxpm-dev
- libxft-dev
- libxext-dev
- xlibmesaglu-dev

1.3.3 Interface Gráfica com Usuário Geometria GDML

Já para conseguirmos utilizar um visualizador (qt), ou uma interface gráfica, para as geometrias do Geant4 e para a utilização de GDML é preciso instalar as seguintes dependências:

- libxerces-c-dev
- qt5-default

1.3.4 Instalação das Dependências

Para realizar a instalação de todas as dependências necessárias citadas anteriormente, basta executar a linha de comando a seguir no terminal, estando como super usuário da máquina:

```
1 sudo apt install \  
2 git dpkg-dev cmake g++ gcc binutils \  
3 libx11-dev libxmu-dev libxpm-dev libxft-dev libxext-dev xlibmesa-glu-dev \  
4 libxerces-c-dev qt5-default
```

Listing 1: Linha de comando para instalação das dependências necessárias para o Geant4.

2 Passo a Passo de Instalação

O processo de instalação é feito em 3 etapas, sendo elas:

1. cmake
2. make
3. make install

Veremos como fazer este processo detalhadamente.

2.1 Compilação

2.1.1 Passo 1: cmake

Primeiramente, é preciso entrar no diretório **geant4.10.06.p02-build**, e então, se sua configuração de pastas está conforme descrito anteriormente, basta rodar o comando a seguir. Nesta etapa, na linha 1 estamos informando onde queremos fazer a instalação, que no caso é na pasta **geant4.10.06.p02-install** criada, enquanto na linha 2 informamos onde serão encontrados os diretórios das bibliotecas de fenômenos físicos. Nas demais linhas habilitamos a interface gráfica de visualização de geometria qt, a de utilização de geometrias em GDML, outras dependências gráficas e o multi processamento.

```
1 cmake -DCMAKE_INSTALL_PREFIX=../geant4.10.06-install \  
2 -DGEANT4_INSTALL_DATADIR=../data \  
3 -DGEANT4_USE_GDML=ON \  
4 -DGEANT4_USE_QT=ON \  
5 -DGEANT4_USE_RAYTRACER_X11=ON \  
6 -DGEANT4_USE_OPENGL_X11=ON \  
7 -DGEANT4_BUILD_MULTITHREADED=ON \  
8 ..
```

Listing 2: Linha de comando para execução do cmake na pasta build

Caso você não tenha salvo os data files das bibliotecas físicas conforme mostrado, será necessário corrigir o caminho de onde se encontram esses diretórios na linha 2.

A sequência de imagens a seguir mostra o comportamento esperado de saída no terminal após a execução do cmake.

```

leticia.rosa@es-rad02-l1:~/Programs/geant4/geant4.10.06.p02/geant4.10.06-build$ cmake
ake -DCMAKE_INSTALL_PREFIX=../geant4.10.06-install \
> -DGEANT4_INSTALL_DATADIR=../data \
> -DGEANT4_USE_GDML=ON \
> -DGEANT4_USE_QT=ON \
> -DGEANT4_USE_RAYTRACER_X11=ON \
> -DGEANT4_USE_OPENGL_X11=ON \
> -DGEANT4_BUILD_MULTITHREADED=ON \
> ..
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Performing Test HAVE_TLS
-- Performing Test HAVE_TLS - Success
-- Found EXPAT: /usr/lib/x86_64-linux-gnu/libexpat.so (found version "2.2.5")

-- Performing Test HAVE_TLS - Success
-- Found EXPAT: /usr/lib/x86_64-linux-gnu/libexpat.so (found version "2.2.5")
-- Found XercesC: /usr/lib/x86_64-linux-gnu/libxerces-c.so (found version "3.2.0")
-- Found OpenGL: /usr/lib/x86_64-linux-gnu/libOpenGL.so
-- Looking for XOpenDisplay in /usr/lib/x86_64-linux-gnu/libX11.so;/usr/lib/x86_64-linux-gnu/libXext.so
-- Looking for XOpenDisplay in /usr/lib/x86_64-linux-gnu/libX11.so;/usr/lib/x86_64-linux-gnu/libXext.so - found
-- Looking for gethostbyname
-- Looking for gethostbyname - found
-- Looking for connect
-- Looking for connect - found
-- Looking for remove
-- Looking for remove - found
-- Looking for shmctl
-- Looking for shmctl - found
-- Looking for IceConnectionNumber in ICE
-- Looking for IceConnectionNumber in ICE - found
-- Found X11: /usr/lib/x86_64-linux-gnu/libX11.so
-- Pre-configuring dataset G4NDL (4.6)
-- Pre-configuring dataset G4EMLOW (7.9.1)
-- Pre-configuring dataset PhotonEvaporation (5.5)
-- Pre-configuring dataset RadioactiveDecay (5.4)
-- Pre-configuring dataset G4PARTICLEXS (2.1)
-- Pre-configuring dataset G4PII (1.3)

-- Pre-configuring dataset G4PARTICLEXS (2.1)
-- Pre-configuring dataset G4PII (1.3)
-- Pre-configuring dataset RealSurface (2.1.1)
-- Pre-configuring dataset G4SAIDDATA (2.0)
-- Pre-configuring dataset G4ABLA (3.1)
-- Pre-configuring dataset G4INCL (1.0)
-- Pre-configuring dataset G4ENSDFSTATE (2.2)
*WARNING*
Geant4 has been pre-configured to look for datasets
in the directory:

/home/ABTLUS/leticia.rosa/Programs/geant4/geant4.10.06.p02/geant4.10.06-install/share/Geant4-10.6.2/data

but the following datasets are NOT present on disk at
that location:

G4NDL (4.6)
G4EMLOW (7.9.1)
PhotonEvaporation (5.5)
RadioactiveDecay (5.4)
G4PARTICLEXS (2.1)
G4PII (1.3)
RealSurface (2.1.1)
G4SAIDDATA (2.0)
G4ABLA (3.1)

```

Figura 4: Primeira parte da saída do cmake.


```

G4SAIDDATA (2.0)
G4ABLA (3.1)
G4INCL (1.0)
G4ENSDSTATE (2.2)

If you want to have these datasets installed automatically
simply re-run cmake and set the GEANT4_INSTALL_DATA
variable to ON. This will configure the build to download
and install these datasets for you. For example, on the
command line, do:

cmake -DGEANT4_INSTALL_DATA=ON <otherargs>

The variable can also be toggled in cmake or cmake-gui.
If you're running on a Windows system, this is the best
solution as CMake will unpack the datasets for you
without any further software being required

Alternatively, you can install these datasets manually
now or after you have installed Geant4. To do this,
download the following files:

https://cern.ch/geant4-data/datasets/G4NDL.4.6.tar.gz
https://cern.ch/geant4-data/datasets/G4EMLow.7.9.1.tar.gz
https://cern.ch/geant4-data/datasets/G4PhotonEvaporation.5.5.tar.gz
https://cern.ch/geant4-data/datasets/G4RadioactiveDecay.5.4.tar.gz

https://cern.ch/geant4-data/datasets/G4ABLA.3.1.tar.gz
https://cern.ch/geant4-data/datasets/G4INCL.1.0.tar.gz
https://cern.ch/geant4-data/datasets/G4ENSDSTATE.2.2.tar.gz

and unpack them under the directory:

/home/ABTLUS/leticia.rosa/Programs/geant4/geant4.10.06.p02/geant4.10.06-inst
all/share/Geant4-10.6.2/data

As we supply the datasets packed in gzipped tar files,
you will need the 'tar' utility to unpack them.

Nota bene: Missing datasets will not affect or break
compilation and installation of the Geant4
libraries.

-- Looking for sys/types.h
-- Looking for sys/types.h - found
-- Looking for stdint.h
-- Looking for stdint.h - found
-- Looking for stddef.h
-- Looking for stddef.h - found
-- Check size of off64_t
-- Check size of off64_t - done
-- Looking for fseeko
-- Looking for fseeko - found

-- Looking for stdint.h
-- Looking for stdint.h - found
-- Looking for stddef.h
-- Looking for stddef.h - found
-- Check size of off64_t
-- Check size of off64_t - done
-- Looking for fseeko
-- Looking for fseeko - found
-- Looking forunistd.h
-- Looking forunistd.h - found
-- The following Geant4 features are enabled:
GEANT4_BUILD_CXXSTD: Compiling against C++ Standard '11'
GEANT4_BUILD_MULTITHREADED: Build multithread enabled libraries
GEANT4_BUILD_TLS_MODEL: Building with TLS model 'initial-exec'
GEANT4_USE_SYSTEM_EXPAT: Using system EXPAT library
GEANT4_USE_GDML: Building Geant4 with GDML support
GEANT4_USE_QT: Build Geant4 with Qt support
GEANT4_USE_RAYTRACER_X11: Build RayTracer driver with X11 support
GEANT4_USE_OPENGL_X11: Build Geant4 OpenGL driver with X11 support

-- Configuring done
-- Generating done
-- Build files have been written to: /home/ABTLUS/leticia.rosa/Programs/geant4/g
eant4.10.06.p02/geant4.10.06-build

```

Figura 5: Segunda parte da saída do cmake.

Note que o próprio Geant4 tem a recomendação do caminho adequado para a localização dos diretórios das bibliotecas físicas. Seguiremos essa recomendação no final.

No final deste processo haverá diversos diretórios na pasta **geant4.10.06.p02-build**, conforme pode ser visto a seguir.

```

leticia.rosa@s-rad02-l:~/Programs/geant4/geant4.10.06.p02/geant4.10.06-build$ ls
BuildProducts          Geant4Config.cmake
CMakeCache.txt         Geant4ConfigVersion.cmake
CMakeCPackOptions.cmake Geant4LibraryDepends.cmake
CMakeFiles             geant4make.csh
cmake_install.cmake    geant4make.sh
cmake_uninstall.cmake  Geant4PackageCache.cmake
cmake_uninstall.cmake.in geant4_validate_sources.cmake
CPackConfig.cmake      install_manifest.txt
CPackSourceConfig.cmake InstallTreeFiles
G4EXPATShim.cmake      LICENSE.txt
G4FreetypeShim.cmake   Makefile
G4HDF5Shim.cmake       Modules
G4MotifShim.cmake      README.txt
G4VecGeomShim.cmake    source
G4X11Shim.cmake        UseGeant4.cmake
geant4-config          UseGeant4_internal.cmake

```

Figura 6: Diretórios gerados na pasta build.

2.1.2 Passo 2: make

Ainda no diretório **geant4.10.06.p02-build** é preciso executar o seguinte comando:

```
1 make -jN
```

Listing 3: Linha de comando para execução do make na pasta build com N núcleos.

em que N é o número de núcleos disponíveis em seu computador para execução do processo. Esta etapa é bastante demorada, levando vários minutos dependendo do desempenho do computador e da quantidade de núcleos disponíveis. Se tudo ocorreu como deveria, ao final deste processo você não deverá encontrar nenhuma mensagem de erro.

2.1.3 Passo 3: make install

Finalmente, ainda na pasta **geant4.10.06.p02-build**, é preciso executar o comando de instalação:

```
1 make install
```

Listing 4: Linha de comando para instalação.

Este processo é bem rápido, e após isso o Geant4 está instalado e pronto para ser usado. Porém, conforme falado anteriormente, iremos mover o diretório data que contém as bibliotecas físicas para deixar de acordo com a recomendação feita pelo próprio guia de instalação do Geant4. Para isso, siga os seguintes passos:

1. Mova a pasta **data** que está em **geant4.10.06.p02** para **geant4.10.06.p02/geant4.10.06.p02-install/share**.

2. Execute novamente o passo 1 com a correção de caminho do diretório das bibliotecas físicas:

```
1 cmake -DCMAKE_INSTALL_PREFIX=../geant4.10.06-install \
2 -DGEANT4_INSTALL_DATADIR=../geant4.10.06-install/share/data \
3 -DGEANT4_USE_GDML=ON \
4 -DGEANT4_USE_QT=ON \
5 -DGEANT4_USE_RAYTRACER_X11=ON \
6 -DGEANT4_USE_OPENGL_X11=ON \
7 -DGEANT4_BUILD_MULTITHREADED=ON \
8 ..
```

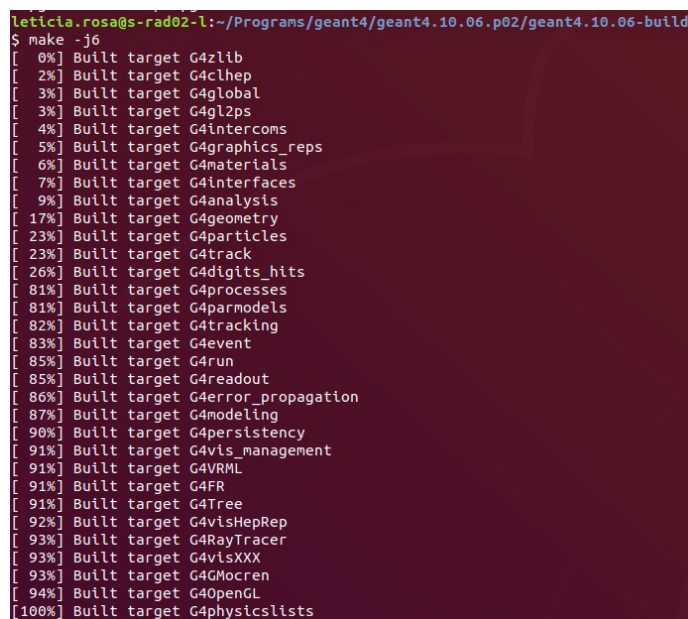
Listing 5: Linha de comando para instalação.

3. Execute o passo 2 novamente. Agora como já foi tudo compilado anteriormente será bem rápido, apenas mostraremos um novo caminho que o Geant4 deve buscar para utilizar as bibliotecas físicas.

```
1 make -jN
```

Listing 6: Linha de comando para execução do make na pasta build com N núcleos.

Você deverá ver algo como a figura a seguir, sem erros:



```
leticia.rosa@s-rad02-l:~/Programs/geant4/geant4.10.06.p02/geant4.10.06-build
$ make -j6
[ 0%] Built target G4zlib
[ 2%] Built target G4clhep
[ 3%] Built target G4global
[ 3%] Built target G4gl2ps
[ 4%] Built target G4intercoms
[ 5%] Built target G4graphics_reps
[ 6%] Built target G4materials
[ 7%] Built target G4interfaces
[ 9%] Built target G4analysis
[ 17%] Built target G4geometry
[ 23%] Built target G4particles
[ 23%] Built target G4track
[ 26%] Built target G4digits_hits
[ 81%] Built target G4processes
[ 81%] Built target G4parmodels
[ 82%] Built target G4tracking
[ 83%] Built target G4event
[ 85%] Built target G4run
[ 85%] Built target G4readout
[ 86%] Built target G4error_propagation
[ 87%] Built target G4modeling
[ 90%] Built target G4persistence
[ 91%] Built target G4vis_management
[ 91%] Built target G4VRML
[ 91%] Built target G4FR
[ 91%] Built target G4Tree
[ 92%] Built target G4visHepRep
[ 93%] Built target G4RayTracer
[ 93%] Built target G4visXXX
[ 93%] Built target G4GMocren
[ 94%] Built target G4OpenGL
[100%] Built target G4physicslists
```

Figura 7: Saída após execução do comando make.

4. Por fim, executamos o passo 3 novamente.

```
1 make install
```

Listing 7: Linha de comando para instalação.

E deverá ver algo como a figura a seguir, sem erros.

```

leticia.rosa@s-rad02-l:~/Programs/geant4/geant4.10.06.p02/geant4.10.06-b
uild$ make install
[ 2%] Built target G4clhep
[ 3%] Built target G4global
[ 4%] Built target G4intercons
[ 4%] Built target G4zlib
[ 6%] Built target G4analysis
[ 7%] Built target G4materials
[ 8%] Built target G4graphics_reps
[16%] Built target G4geometry
[22%] Built target G4particles
[22%] Built target G4track
[25%] Built target G4digits_hits
[80%] Built target G4processes
[81%] Built target G4tracking
[82%] Built target G4event
[84%] Built target G4run
[85%] Built target G4error_propagation
[86%] Built target G4interfaces
[86%] Built target G4parmodels
[89%] Built target G4persistence
[94%] Built target G4physicslists
[94%] Built target G4readout
[95%] Built target G4modeling
[96%] Built target G4vis_management
[96%] Built target G4FR
[97%] Built target G4visHepRep
[98%] Built target G4RayTracer
[98%] Built target G4Tree
[98%] Built target G4VRML
[98%] Built target G4visXXX
[98%] Built target G4gl2ps
[98%] Built target G4CMocren
[100%] Built target G4OpenGL
Install the project...
-- Install configuration: "Release"
-- Installing: /home/ABTLUS/leticia.rosa/Programs/geant4/geant4.10.06.p0
2/geant4.10.06-install/share/Geant4-10.6.2/geant4make/geant4make.sh
-- Installing: /home/ABTLUS/leticia.rosa/Programs/geant4/geant4.10.06.p0
2/geant4.10.06-install/share/Geant4-10.6.2/geant4make/geant4make.csh
-- Installing: /home/ABTLUS/leticia.rosa/Programs/geant4/geant4.10.06.p0
2/geant4.10.06-install/bin/geant4.sh
-- Installing: /home/ABTLUS/leticia.rosa/Programs/geant4/geant4.10.06.p0
2/geant4.10.06-install/bin/geant4.csh
-- Installing: /home/ABTLUS/leticia.rosa/Programs/geant4/geant4.10.06.p0
2/geant4.10.06-install/bin/geant4-config
-- Installing: /home/ABTLUS/leticia.rosa/Programs/geant4/geant4.10.06.p0
2/geant4.10.06-install/lib/Geant4-10.6.2/Geant4LibraryDepends-release.cm
ake
-- Installing: /home/ABTLUS/leticia.rosa/Programs/geant4/geant4.10.06.p0
2/geant4.10.06-install/lib/Geant4-10.6.2/Geant4Config.cmake

```

Figura 8: Saída após execução do comando `make install`.

Agora o ambiente está todo configurado conforme as recomendações, e todos os modos de instalação estão em um único diretório **geant4.10.06.p02-install**.

2.2 Configurando a Bash

Por fim, colocaremos as configurações necessárias do Geant4 em sua bash para fácil compilação dos códigos futuros. Para isso, abra um terminal, e em sua home abra a bash em qualquer editor de preferência, por exemplo com o comando:

```
1 code .bashrc
```

Listing 8: Linha de comando para abrir a bash.

ou

`leticiaabraga.rosa@gmail.com`

```
1 gedit .bashrc
```

Listing 9: Linha de comando para abrir a bash.

e então vamos inserir as configurações necessárias, passando os caminhos com os arquivos necessários para compilação de códigos com as bibliotecas do Geant4. Insira as linhas de código a seguir no final de sua bash.

```
1 # GEANT4 Configuration
2
3 source ${HOME}/Programs/geant4/geant4.10.06.p02/geant4.10.06-install/bin/
  geant4.sh
4 export G4INCLUDES=${HOME}/Programs/geant4/geant4.10.06.p02/geant4.10.06-
  install/include/Geant4
```

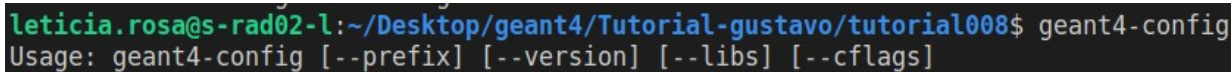
Listing 10: Comandos do Geant4 para inserir na bash .

Após adicionar essas linhas ao final de sua bash, salve e feche o terminal para exportar as alterações feitas. Para testar se o ambiente está configurado, execute o comando:

```
1 geant4-config
```

Listing 11: Comando para teste de ambiente chamando o geant4

e então deverá ver algo como na figura 9, e o ambiente está pronto para utilizar o Geant4.



```
leticia.rosa@s-rad02-l:~/Desktop/geant4/Tutorial-gustavo/tutorial008$ geant4-config
Usage: geant4-config [--prefix] [--version] [--libs] [--cflags]
```

Figura 9: Saída após execução do comando geant4-config.

3 Compilação de Código

Formas de compilar o código GEANT

-Na pasta do projeto existem duas formas:

1. Usar o arquivo compile.sh criado, executando no terminal:

```
source compile.sh ./file macros/photons.mac
```

obs.: o "file" é o que está determinado no arquivo compile para o nome de arquivo executável gerado

2. Incluir o arquivo Cmakelist na pasta do projeto e, nessa pasta fazer:

```
mkdir build cd build cmake.. make
```

****** esta é uma maneira mais segura de compilação, ele checa os requisitos do sistema, checa se tem o geant instalado, tudo sozinho, antes de tentar compilar o programa, não depende de um shell script que é mais passível de falha. e é feito pra não falhar mesmo mudando de sistema operacional.

obs.: caso a estrutura de pastas seja mudada, por exemplo criando uma pasta header para todos os arquivos .h e source apenas para os .cpp, é preciso alterar no arquivo cmakefile usado para compilar.

Estrutura de pastas atual: ——build——* ——source—— ——macro—— main.cpp compile.sh CMakeLists.txt*

*para compilar com o cmakefile

dentro de source todos os .cpp e .h

Referências

[1] <https://www.youtube.com/watch?v=tjc-ARiawEs>