N-MEIGA

Luigui Miranda

luigui2248385@correo.uis.edu.co

Escuela Latinoamericana
de Rayos cósmicos y sus aplicaciones

Simulacion de neutrones

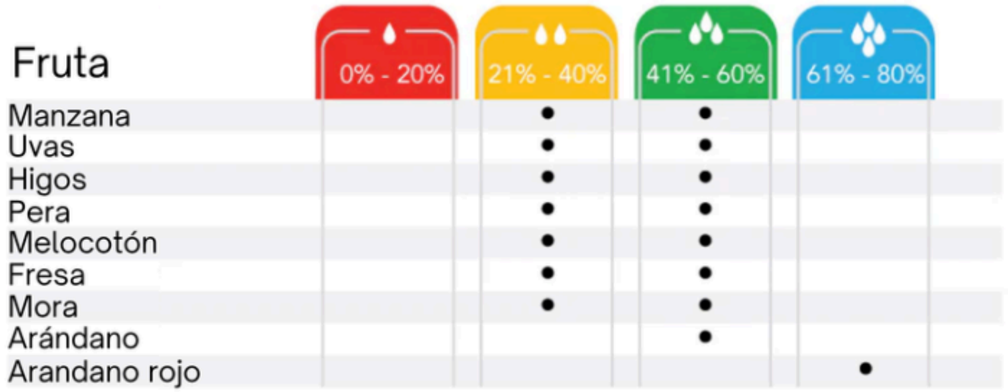
Las sequías intensificadas a nivel global amenazan los medios de vida y la seguridad alimentaria, con pérdidas en la producción agrícola reportadas entre el 5 % y el 50 % dependiendo de la región y del cultivo. Este escenario evidencia la necesidad de optimizar los procesos agrícolas para mejorar su resiliencia frente a eventos climáticos extremos.

La agricultura de precisión, mediante tecnologías como sensores y GPS, permite un uso eficiente de agua y suelo, conservando recursos y aumentando la productividad de manera sostenible.

Simulacion de neutrones

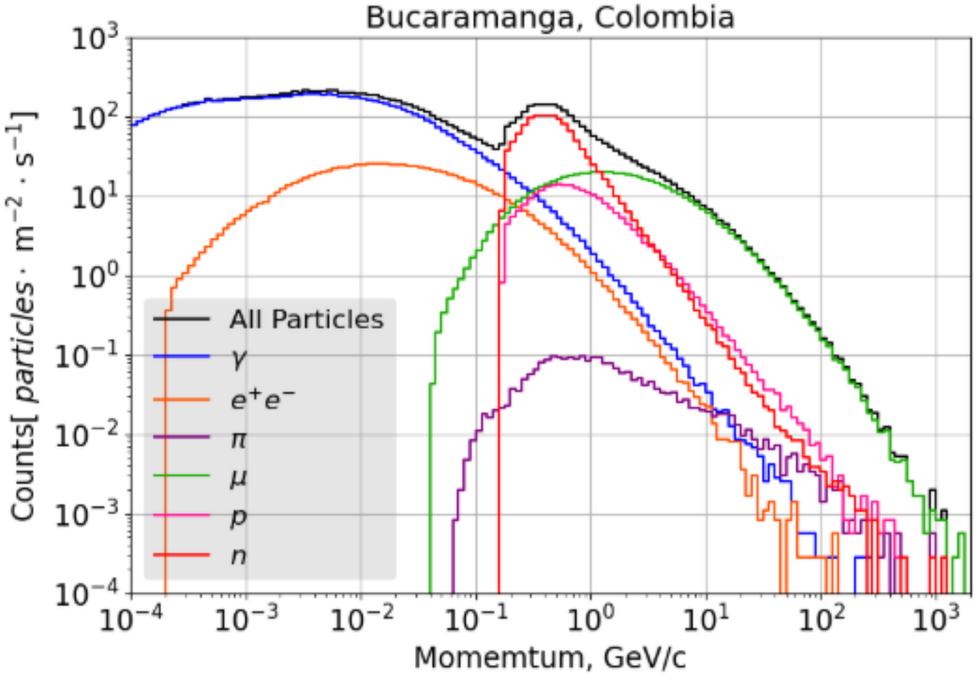
Las sequías intensificadas a nivel global amenazan los medios de vida y la seguridad alimentaria, con pérdidas en la producción agrícola reportadas entre el 5 % y el 50 % dependiendo de la región y del cultivo. Este escenario evidencia la necesidad de optimizar los procesos agrícolas para mejorar su resiliencia frente a eventos climáticos extremos.

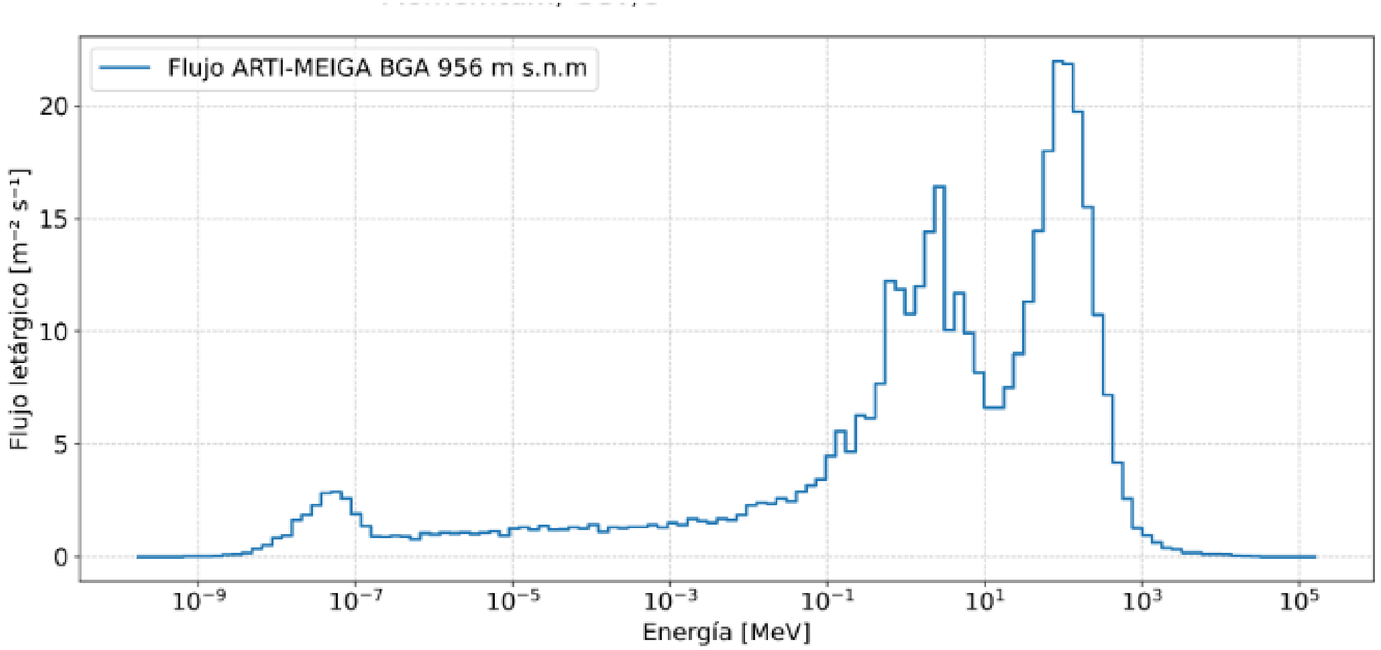
La agricultura de precisión, mediante tecnologías como sensores y GPS, permite un uso eficiente de agua y suelo, conservando recursos y aumentando la productividad de manera sostenible.



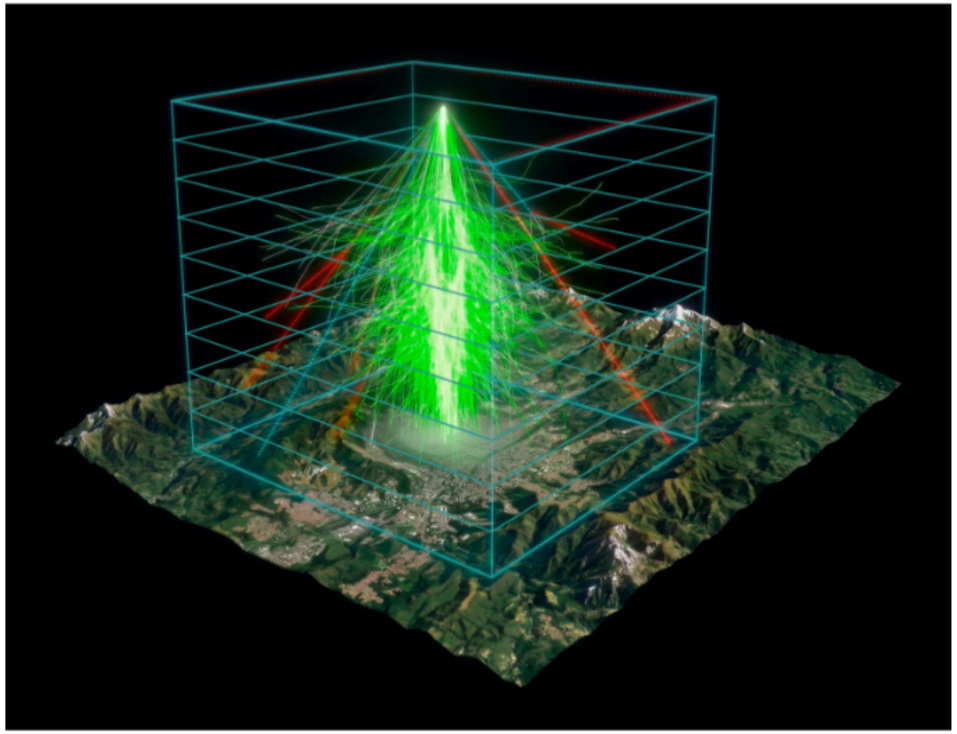
Fruta	0% - 20%	21% - 40%	41% - 60%	61% - 80%
Manzana		•	•	
Uvas		•	•	
Higos		•	•	
Pera		•	•	
Melocotón		•	•	
Fresa		•	•	
Mora		•	•	
Arándano			•	
Arandano rojo				•

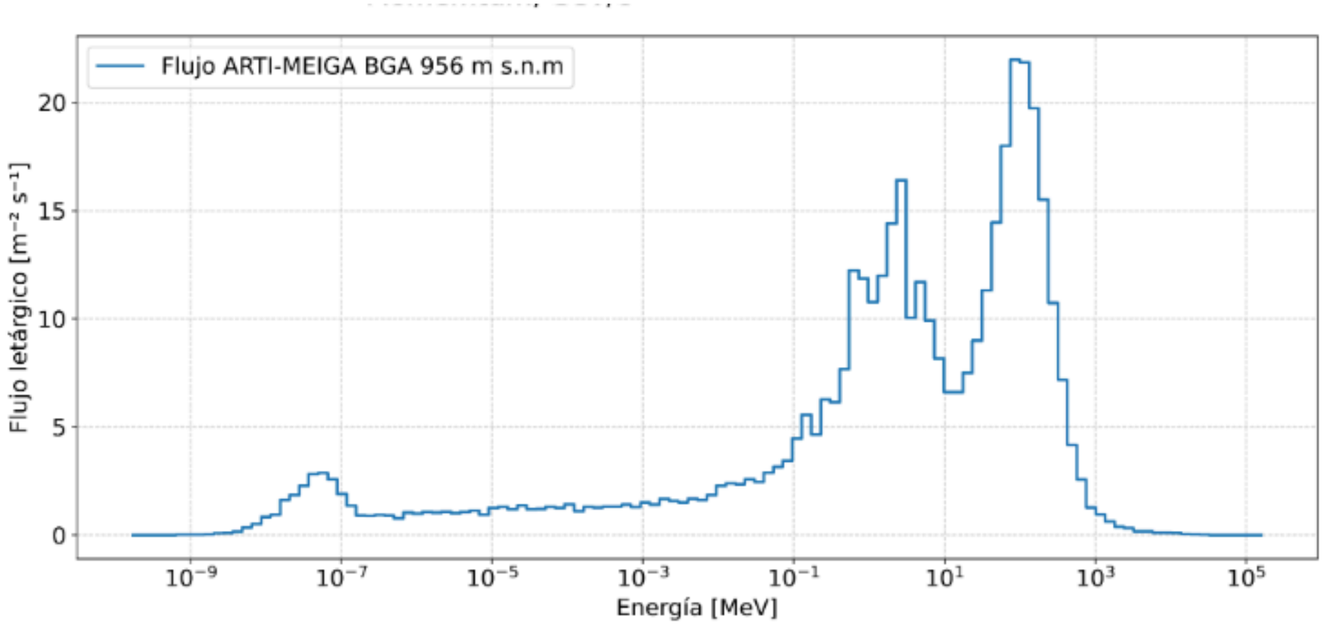
Simulacion de neutrones



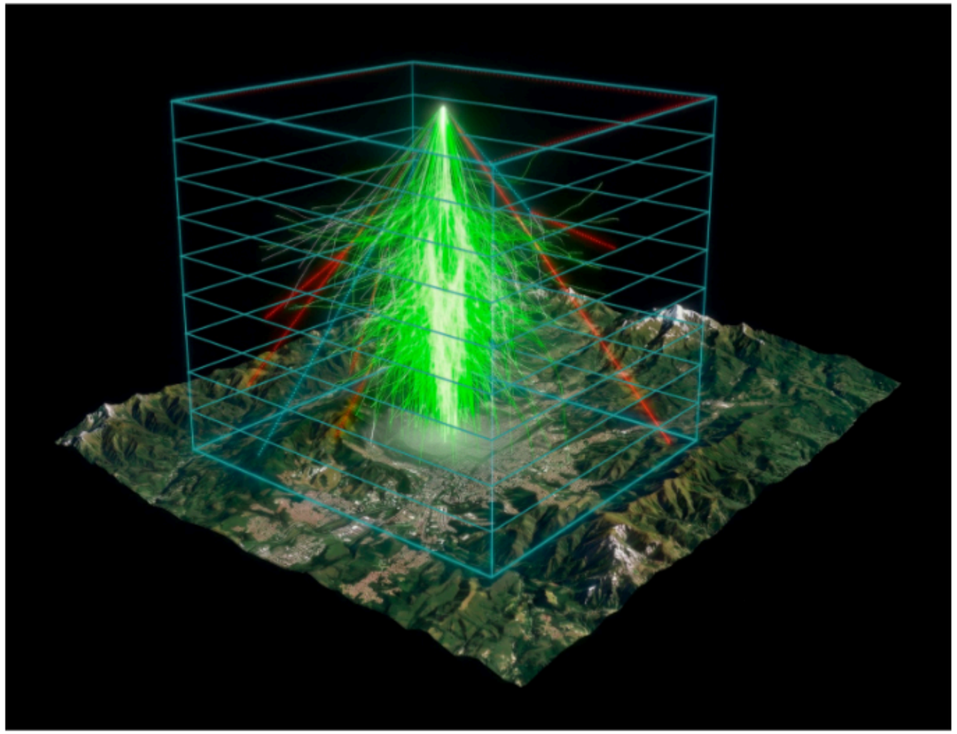


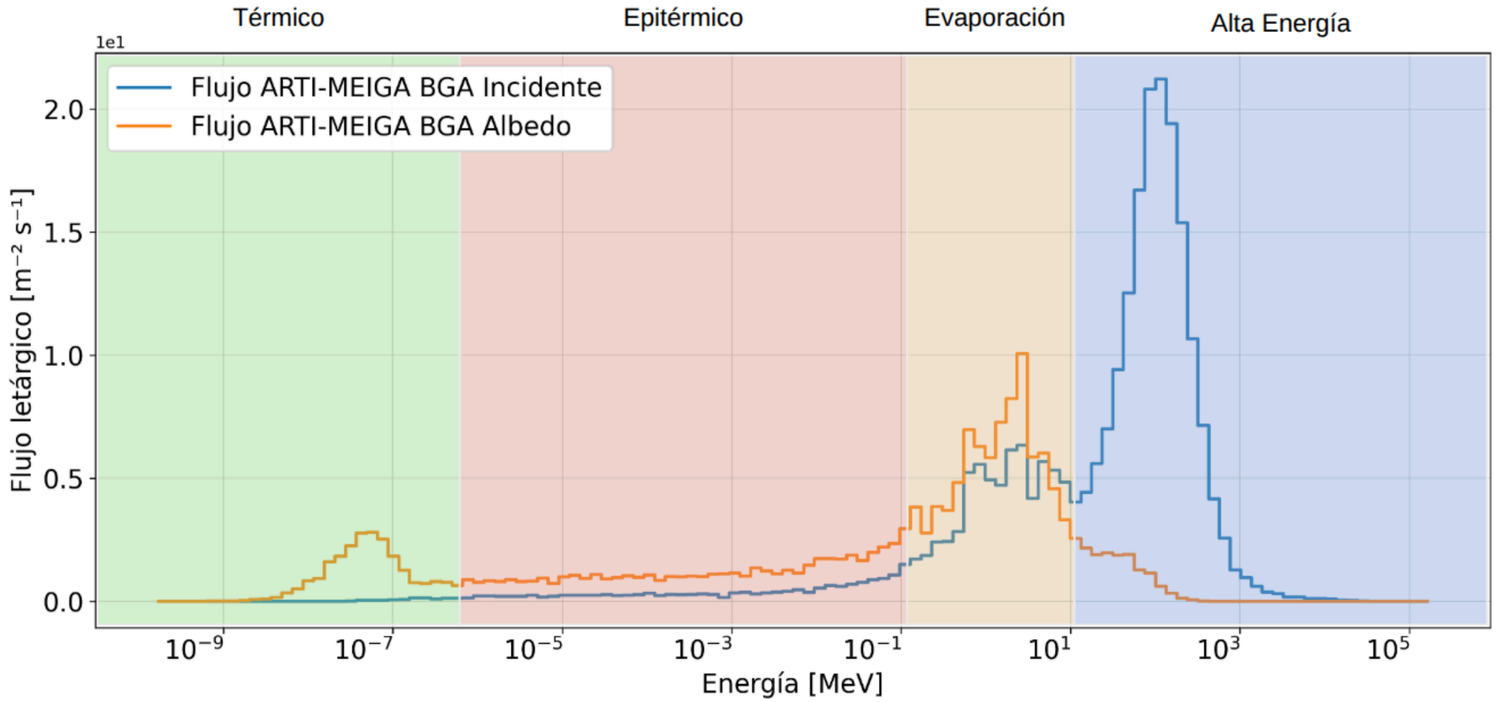
Simulación de neutrones



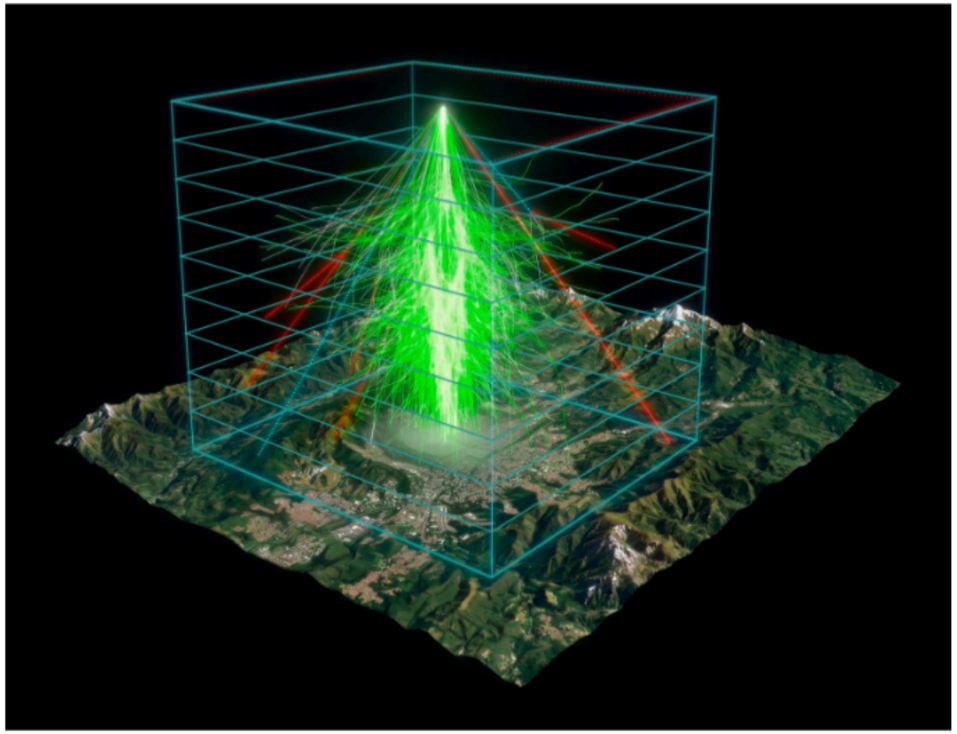


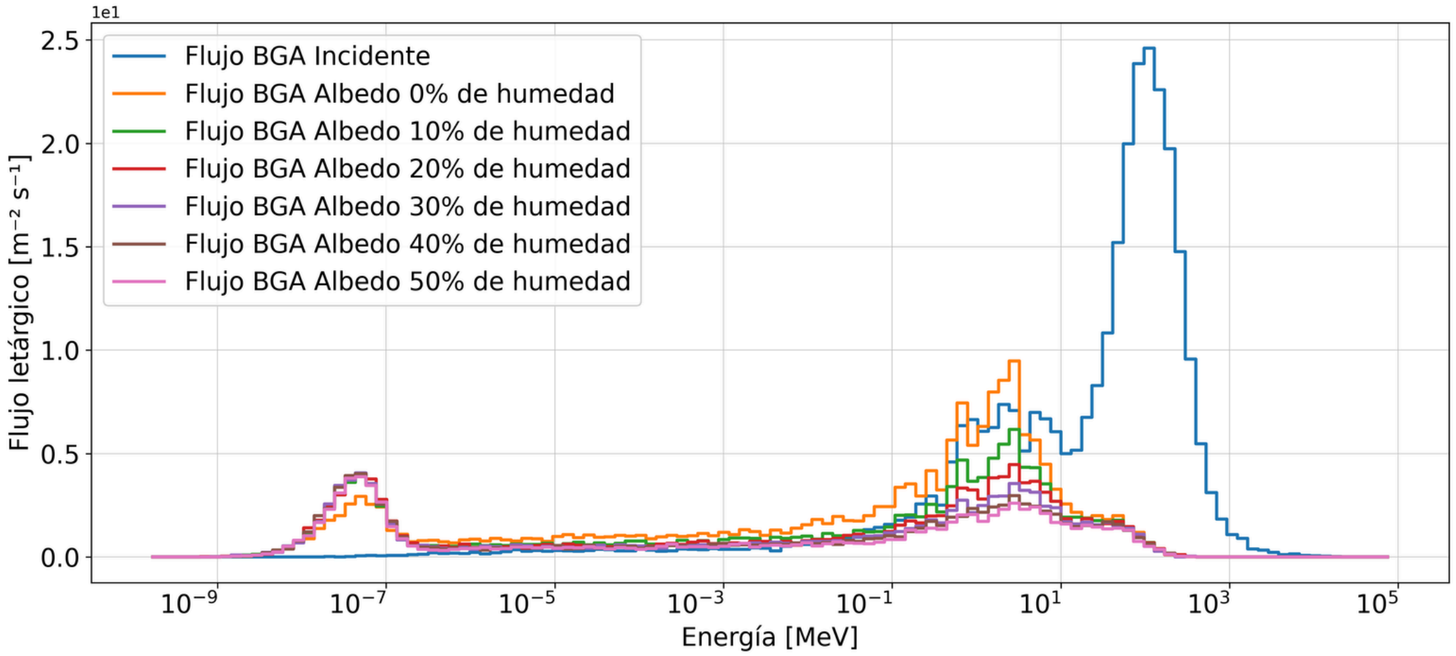
Simulación de neutrones





Simulación de neutrones



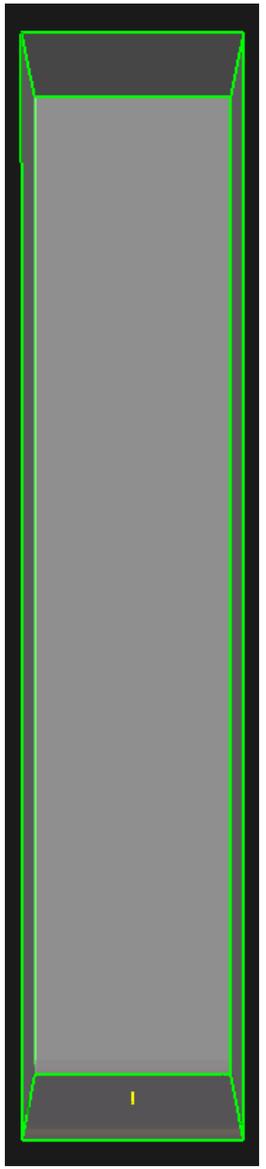


Sistema físico

El sistema físico implementado en Geant4 para medir el flujo de neutrones que llegan al nivel del suelo después de atravesar la atmósfera consiste en una columna de 200 m de altura y una base de 40 × 40 m.

sudo docker start meiga_school

sudo docker exec -it meiga_school /bin/bash



Materiales

- En **cd /opt/meiga-school/external/G4GROSimulator/meiga-source/G4Models/**
- **vim Materials.cc** y **vim Materials.h** se crean los elementos para definir los suelos.
- para editar en vim se oprime la tecla “i”
- Para salir de vim sin guardar Esc :q!
- Para salir de vim y guardar Esc :wq!
- Para usar suelo estándar o suelos con humedad usamos los siguientes indicativos:

App G4GROSimulator

cd /opt/meiga-school/external/G4GROSimulator/meiga-source/Applications/G4GROSimulator/ es una aplicación de Meiga construida en Geant4. Está diseñada para simular flujos de neutrones que alcanzan el nivel del suelo después de haber viajado a través de la atmósfera , y puede simular suelos con diferentes composiciones químicas.

Dentro de G4GROSimulator encontraremos,los siguientes archivos:

```
root@910e19d1b62c:/opt/meiga-school/external/G4GROSimulator/meiga-source/Applications/G4GROSimulator# ls
CMakeLists.txt          G4GROEventAction.cc   G4GROSensitiveDetector.cc   G4GROSimulator.json      G4GROSteppingAction.h   all_neutrons.shw
DetectorList.xml        G4GROEventAction.h    G4GROSensitiveDetector.h    G4GROStackingAction.cc   G4GROTrackingAction.cc  bga_30_segundos.shw
G4GROConstruction.cc    G4GRORunAction.cc     G4GROSimulator.cc           G4GROStackingAction.h    G4GROTrackingAction.h   practica.shw
G4GROConstruction.h     G4GRORunAction.h      G4GROSimulator.h            G4GROSteppingAction.cc   README                  vertical_muon.txt
```

Editar para cambiar las medidas de los volumenes

Editar para cambiar la humedad

Editar para cambiar el archivo de salida

Input

Archivos de inyección

Materiales

- Se define el material a usar dentro del archivo usando los indicativos definidos en Materials.cc
- **vim G4GROConstruction.cc**

```
//------------------------------------------------------------------------
//Soil's Model
//------------------------------------------------------------------------
void G4ATMConstruction::CreateGround()
{
    solidGround = new G4Box("Layer_Ground", fGroundSizeX / 2, fGroundSizeY / 2, fGroundSizeZ / 2);
    G4double center_ground = (-1*fWorldSizeZ + fGroundSizeZ) / 2.0; //Este no habia que modificarlo
    logicGround = new G4LogicalVolume(solidGround, Materials().StdRock, "Ground"); //Select the soil : SoilBH5, 25, std.rock
    physGround = new G4PVPlacement(nullptr, G4ThreeVector(0, 0, center_ground), logicGround, "Ground", logicWorld, false, 0, fCheckOverlaps);
    G4VisAttributes brown(G4Colour::Brown());
    logicGround->SetVisAttributes(brown);
    G4cout << "*******************************************"<< endl;
    G4cout << "...G4GROUNDConstruction : CREATE GROUND" << endl;
    G4cout << " Atmosphere center location :  (0 , 0,"<<center_ground/m <<")"<< " in m"<< endl;
    G4cout << "*******************************************"<< endl;

}
```

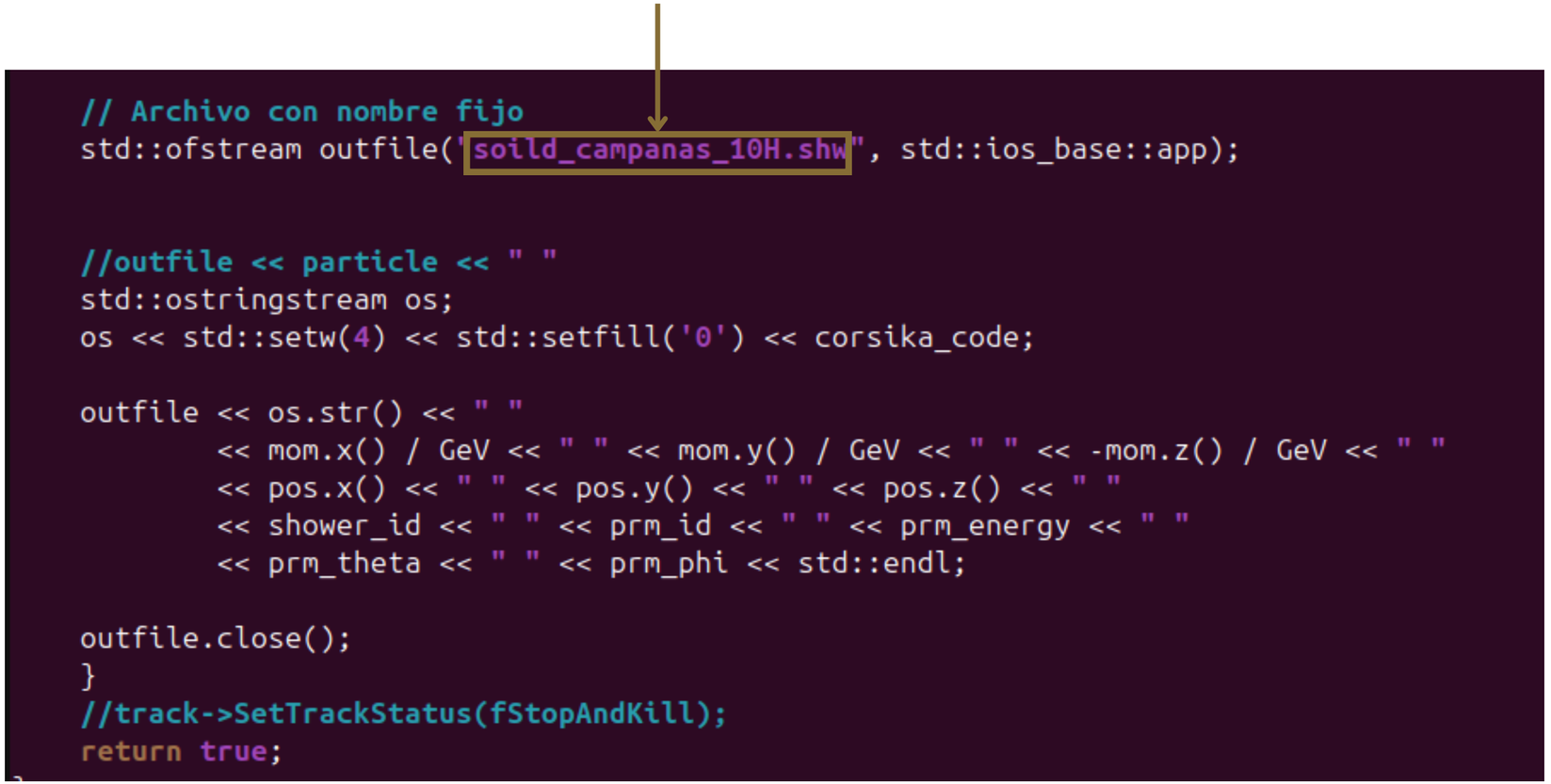
Medidas

- Se defina las medidas en el archivo
- **vim G4GROConstruction.h**


```
void CreateWorld();
void CreateUniver();
void CreateSens();
void CreateGround();

void PlaceDetector(Event& theEvent);
G4VPhysicalVolume* CreateDetector();

bool fCheckOverlaps = false;

// solids
G4Box* solidWorld = nullptr;
G4Box* solidUniver = nullptr;
G4Box* solidSens1 = nullptr;
G4Box* solidSens2 = nullptr;
G4Box* solidGround = nullptr;

// logical and physical volumes
G4LogicalVolume* logicWorld = nullptr;
G4PVPlacement*   physWorld  = nullptr;
G4LogicalVolume* logicUniver = nullptr;
G4PVPlacement*   physUniver  = nullptr;
G4LogicalVolume* logicSens1 = nullptr;
G4PVPlacement*   physSens1  = nullptr;
//G4LogicalVolume* logicSens2 = nullptr;
//G4PVPlacement*   physSens2  = nullptr;
G4LogicalVolume* logicGround = nullptr;
G4PVPlacement*   physGround  = nullptr;

// size definitions
G4double fWorldSizeX = 4 *CLHEP::m;
G4double fWorldSizeY = 40 *CLHEP::m;
G4double fWorldSizeZ = 200 *CLHEP::m;

G4double fUniverSizeX = 40.1 *CLHEP::m;
G4double fUniverSizeY = 40.1 *CLHEP::m;
G4double fUniverSizeZ = 200 *CLHEP::m;

G4double fSens1SizeX = 40 *CLHEP::m;
G4double fSens1SizeY = 40 *CLHEP::m;
G4double fSens1SizeZ = 0.01 *CLHEP::m;

G4double fGroundSizeX = 40 *CLHEP::m;
G4double fGroundSizeY = 40 *CLHEP::m;
G4double fGroundSizeZ = 1 *CLHEP::m;

G4int NLayers = 1;

Event& fEvent;
```

Archivo de salida

- Se define el archivo de salida dentro del archivo **vim G4GROSensitiveDetector.cc**



```
// Archivo con nombre fijo
std::ofstream outfile("soild_campanas_10H.shw", std::ios_base::app);


//outfile << particle << " "
std::ostringstream os;
os << std::setw(4) << std::setfill('0') << corsika_code;

outfile << os.str() << " "
        << mom.x() / GeV << " " << mom.y() / GeV << " " << -mom.z() / GeV << " "
        << pos.x() << " " << pos.y() << " " << pos.z() << " "
        << shower_id << " " << prm_id << " " << prm_energy << " "
        << prm_theta << " " << prm_phi << std::endl;

outfile.close();
}
//track->SetTrackStatus(fStopAndKill);
return true;
```

Input

- Se definen las variables iniciales dentro del archivo **vim G4GROSimulator.json**

```
{
        "Input" :
        {
                "Mode" : "UseARTI",
                "InputFileName" : "/opt/meiga-school/external/G4GROSimulator/meiga-source/Applications/G4GROSimulator/all_neutrons.shw",
                "InputNParticles" : 0
        },

        "Output" :
        {
                "OutputFile" : "./",
                "CompressOutput" : true,
                "SaveInput": false,
                "SavePETimeDistribution" : false,
                "SaveComponentsPETimeDistribution" : false,
                "SaveEnergy" : false,
                "SaveComponentsEnergy" : false
        },
        "DetectorList" : "./DetectorList.xml",
        "DetectorProperties" : "./DetectorProperties.xml",
        "Simulation" :
        {
                "SimulationMode" : "eFast",
                "InjectionMode" : "eRandom",
                "GeoVisOn" : false,
                "TrajVisOn" : false,
                "CheckOverlaps" : false,
                "Verbosity" : 1,
                "RenderFile" : "VRML2FILE",
                "PhysicsName" : "QGSP_BERT_HP"
        }
}
```

Input

- Flujo que se inyecta **vim all_neutrons.shw**

```
# CorsikaId px py pz x y z shower_id prm_id prm_energy prm_theta prm_phi
0013 +0.00000000e+00 +0.00000000e+00 +1.69679993e+00 -3.10175e+03 -1.77430e+04 +9.01175e+02 00000001 0013 +1.00000e+00 +00.000 +0033.985
0013 -0.00000000e+00 +0.00000000e+00 +1.69679993e+00 +7.62179e+03 -1.12032e+04 +9.07328e+02 00000002 0013 +1.00000e+00 +00.000 +0107.522
0013 +0.00000000e+00 -0.00000000e+00 +1.69679993e+00 -1.25015e+04 +1.53212e+03 +9.12855e+02 00000003 0013 +1.00000e+00 +00.000 +0339.518
```

- En **vim DetectorList.xml** se edita el punto de inyección

```
<?xml version='1.0' encoding='ISO-8859-1'?>

<detectorList>

        <injectionMode type="eCircle">
            <x unit="m">0</x>
            <y unit="m">0</y>
            <z unit="m">-97.6</z>
            <radius unit="m">0.56419</radius
            <height unit="m">-97.6</height>
        </injectionMode>


        <detector id="0" type="eDummy">
                <x unit="cm"> 0.0 </x>
                <y unit="cm"> 0.0 </y>
                <z unit="cm"> 0.0 </z>
                <NLayers> 10 </NLayers>
        </detector>


</detectorList>
```

Archivos necesarios

cd /opt/meiga-school/external/G4GROSimulator/meiga-source/Applications/

Editamos **vim CMakeLists.txt**

```
cmake_minimum_required(VERSION 3.10)
#add_subdirectory(G4ExSimulator)
#add_subdirectory(G4WCDSimulator)
#add_subdirectory(G4MuDecSimulator)
#add_subdirectory(G4HodoscopeSimulator)
#add_subdirectory(G4LeadSimulator)
#add_subdirectory(G4TowSimulator)
#add_subdirectory(G4RCSimulator)
add_subdirectory(G4GROSimulator)
```

- En **cd /opt/meiga-school/external/G4GROSimulator/build**
- se debe ejecutar el comando **make -j4**

- En **cd /opt/meiga-school/external/G4GROSimulator/build/Applications/G4GROSimulator/**
- encontraremos los archivos
  - **DetectorList.xml**
  - **DetectorProperties.xml**
  - **G4GROSimulator**
  - **G4GROSimulator.json**

```
root@910e19d1b62c:/opt/meiga-school/external/G4GROSimulator/build/Applications/G4GROSimulator# ls
CMakeFiles  DetectorList.xml  DetectorProperties.xml  G4GROSimulator  G4GROSimulator.json  Makefile  cmake_install.cmake  soild_campanas_10H.shw  vertical_muon.txt
```

Archivos necesarios

Editamos el archivo
vim G4GROSimulator.json

"InputFileName" :/opt/meiga-school/external/G4GROSimulator/meiga-source/Applications/G4GROSimulator/all_neutrons.shw

```
{
        "OutputFile" : "./",
        "CompressOutput" : true,
        "SaveInput": false,
        "SavePETimeDistribution" : false,
        "SaveComponentsPETimeDistribution" : false,
        "SaveEnergy" : false,
        "SaveComponentsEnergy" : false
    },
    "DetectorList" : "./DetectorList.xml",
    "DetectorProperties" : "./DetectorProperties.xml",
    "Simulation" :
    {
        "SimulationMode" : "eFast",
        "InjectionMode" : "eRandom",
        "GeoVisOn" : false, true
        "TrajVisOn" : false, true
        "CheckOverlaps" : false,
        "Verbosity" : 1,
        "RenderFile" : "VRML2FILE",
        "PhysicsName" : "QGSP_BERT_HP"
    }
}
```

- encontraremos los archivos
  - **DetectorList.xml**
  - **DetectorProperties.xml**
  - **G4GROSimulator**
  - **G4GROSimulator.json**

```
oot@910e19d1b62c:/opt/meiga-school/external/G4GROSimulator/build/Applications/G4GROSimulator# ls
MakeFiles  DetectorList.xml  DetectorProperties.xml  G4GROSimulator  G4GROSimulator.json  Makefile  cmake_install.cmake  soild_campanas_10H.shw  vertical_muon.txt
```

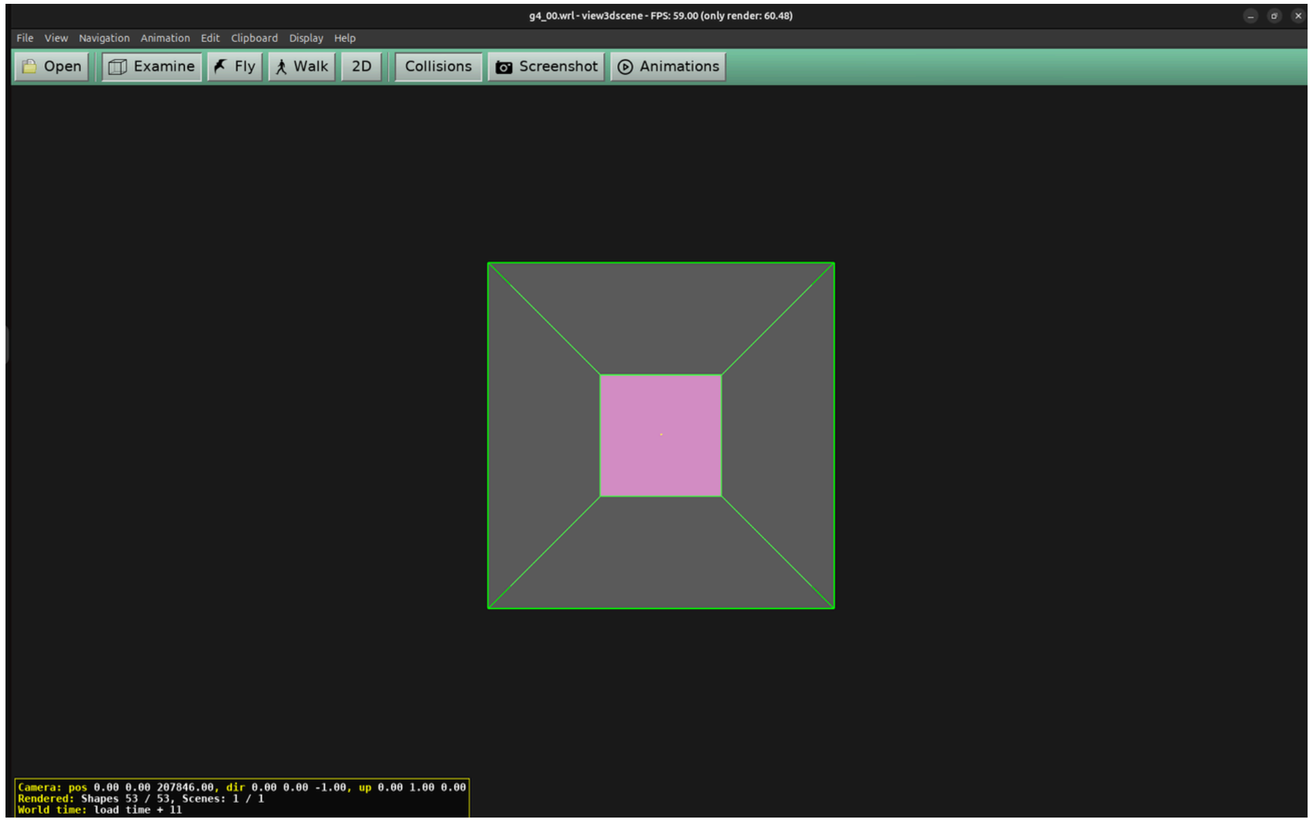
Se ejecuta con: **./G4GROSimulator -c G4GROSimulator.json**

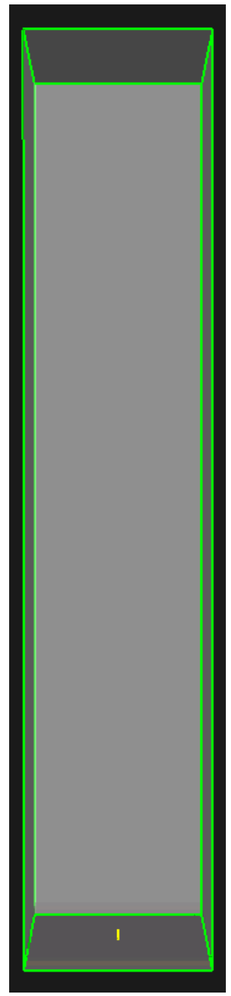
Visualizar lo que hicimos (linux)

Fuera del docker, (abrimos una nueva terminal): Sacar archivos con: **sudo docker cp meiga_school:/opt/meiga-school/external/G4GROSimulator/build/Applications/G4GROSimulator/g4_00.wrl ./**

Instalamos **: sudo apt install view3dscene**

Para visualizar : **view3dscene g4_00.wrl**





Visualizar lo que hicimos (codespaces)

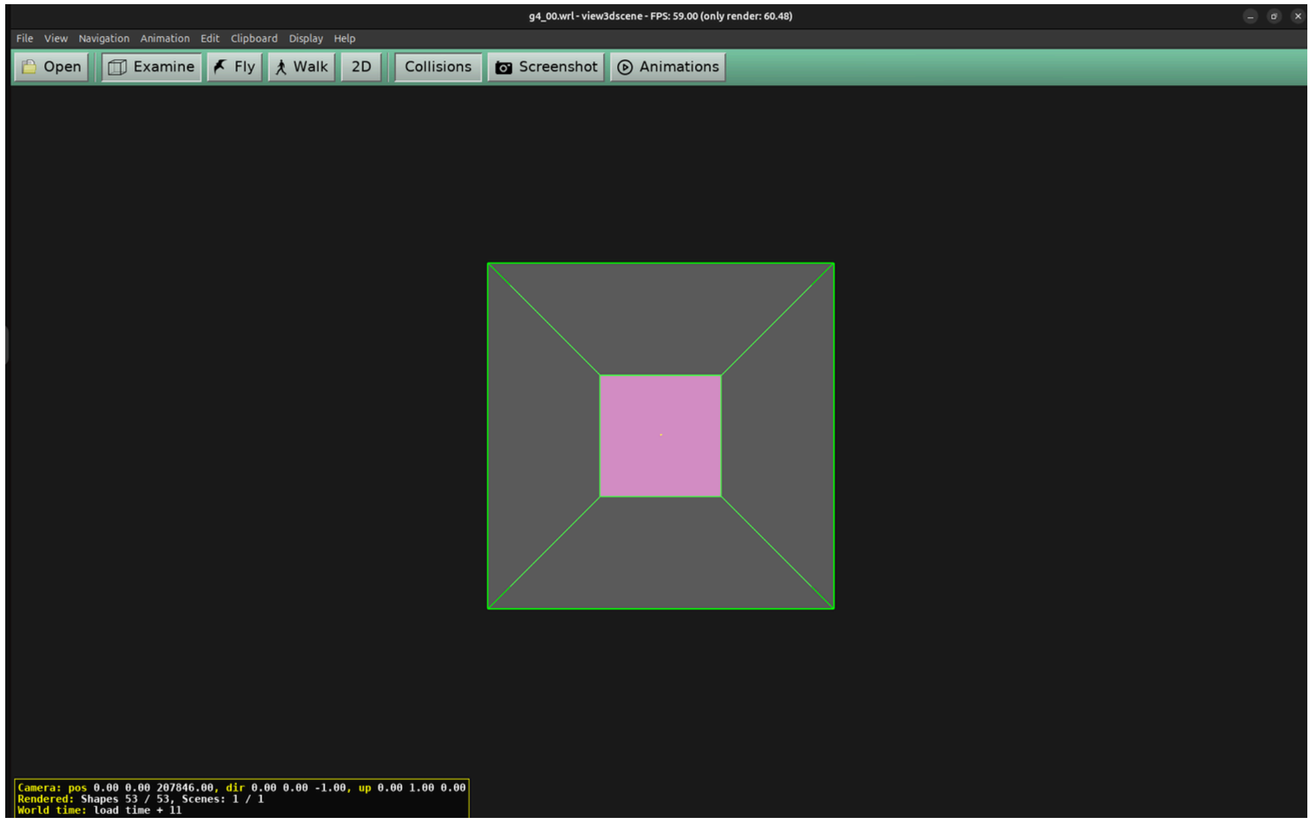
exit

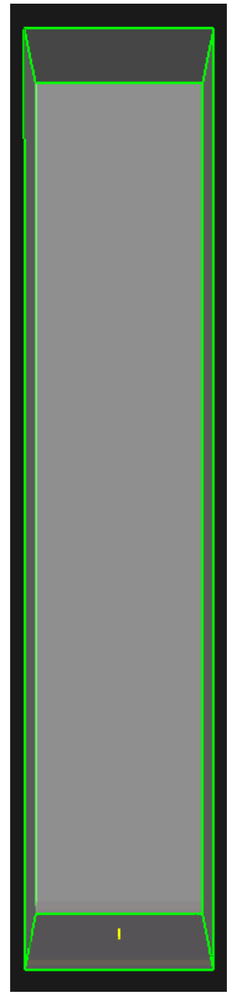
mkdir -p results/visualization

docker cp \meiga_school:/opt/meiga-school/external/G4GROSimulator/build/Applications/G4GROSimulator/g4_00.wrl \results/visualization/

./meiga-school view results/visualization/g4_00.wrl

./meiga-school view --stop





Segunda simulación

Borrar archivos de la simulación anterior

rm soild_campanas_10H.shw g4_00.wrl g4_01.wrl g4_02.wrl

Nueva simulación

vim G4GROSimulator.json

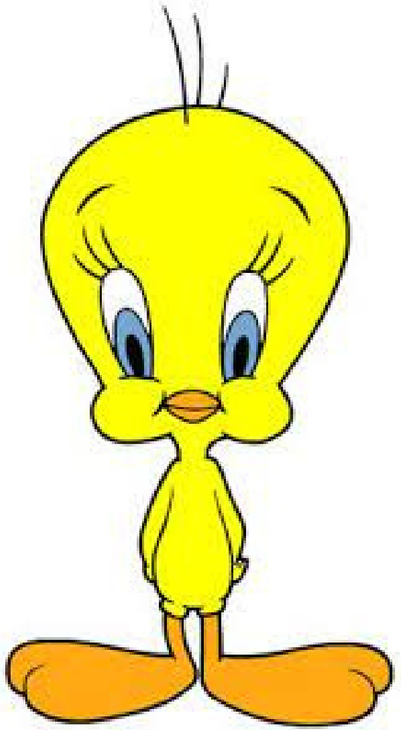
"InputFileName" : **"/opt/meiga-school/external/G4GROSimulator/meiga-source/Applications/G4GROSimulator/bga_30_segundos.shw"**

"TrajVisOn" : **false,**
"CheckOverlaps" : **false,**

./G4GROSimulator -c G4GROSimulator.json

Sacar archivos con: **sudo docker cp meiga_school:/opt/meiga-school/external/G4GROSimulator/meiga-source/Applications/G4GROSimulator/soild_campanas_10H.shw ./**

Descargar archivo de analisis:
https://colab.research.google.com/drive/1LD74Fnwu87pnZWPTFxdoS7FgLn72eSSZ?usp=sharing



Gracias por su atención

Instalara docker y MEIGA

- Instalara docker: **sudo apt-get install docker**
- Instalara MEIGA: https://hub.docker.com/r/asoreyh/meiga <**docker pull asoreyh/meiga**>
- **docker run -d --name meiga_school --init -p 127.0.0.1:6080:6080 rmartinezmaple/meiga-school:3.4.1-g4gro-viewer sleep infinity**
- Para poder ver si la imagen ha sido creada correctamente hacemos**: sudo docker images -a**
- Para iniciar el docker**: docker start nombre_de_la_imagen**
- Ingresar al docker**: docker attach nombre_de_la_imagen**

borrar el archivo src que viene en el docker y fuera del docker ingresar el nuevo archivo src.