Simulación de detectores con GEANT4

Adriana Vásquez

Masterclass de Astropartículas

Agosto 2021

COMPLEJO ASTRONÓMICO
Grupo Halley
Universidad Industrial de Santander

GEANT4
A SIMULATION TOOLKIT

Geant4

https://geant4.web.cern.ch/node/1

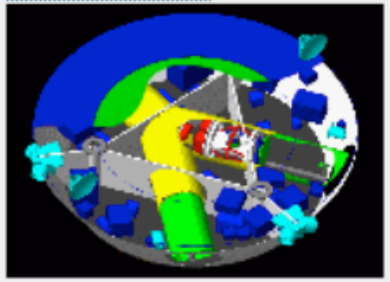
Overview

Geant4 is a toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science. The three main reference papers for Geant4 are published in Nuclear Instruments and Methods in Physics Research A 506 (2003) 250-303 , IEEE Transactions on Nuclear Science 53 No. 1 (2006) 270-278 and Nuclear Instruments and Methods in Physics Research A 835 (2016) 186-225 .

Applications

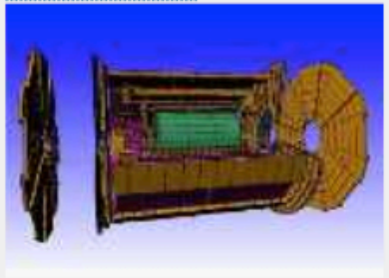
A sampling of applications, technology transfer and other uses of Geant4

User Support



Getting started, guides and information for users and developers

Publications



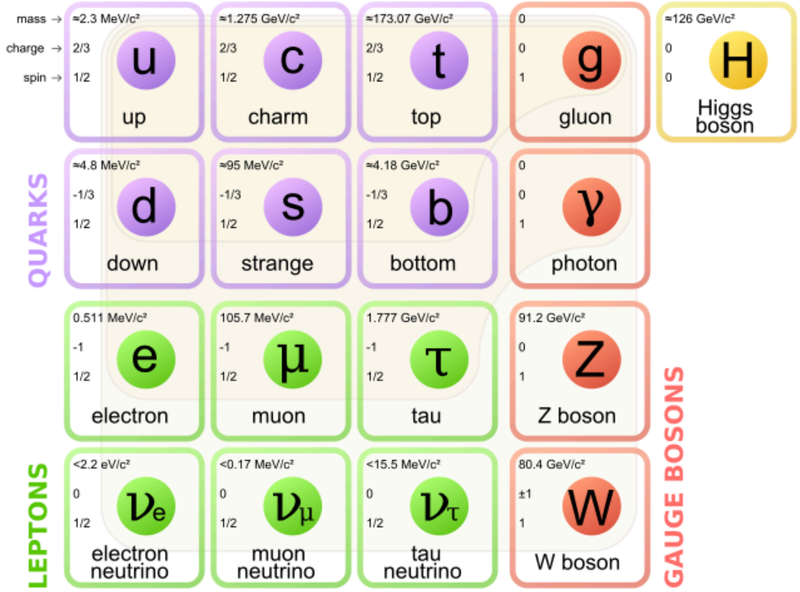
Validation of Geant4, results from experiments and publications

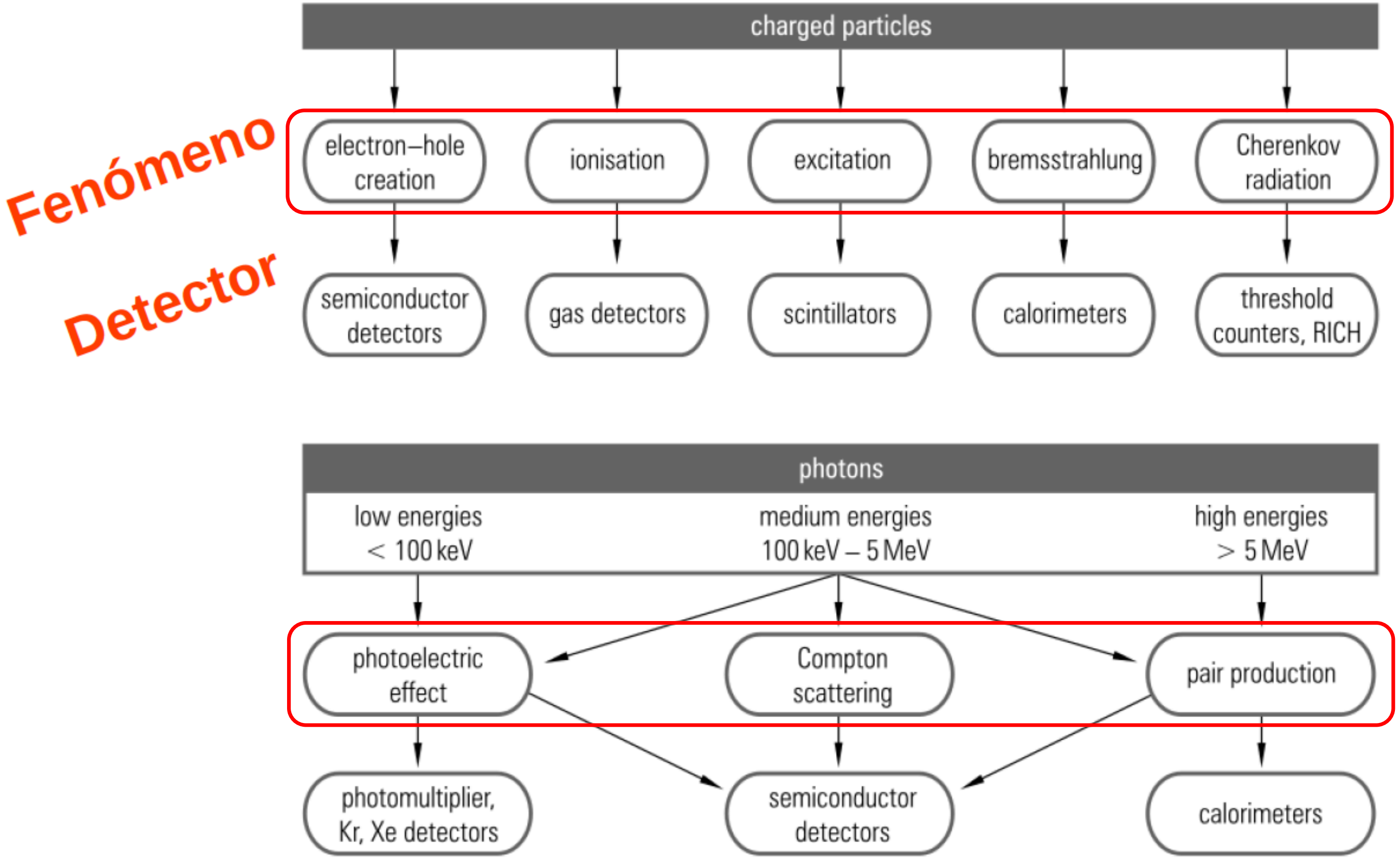
Collaboration

Who we are: collaborating institutions, members, organization and legal information

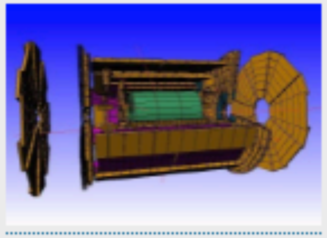
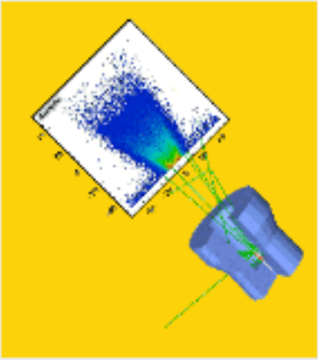
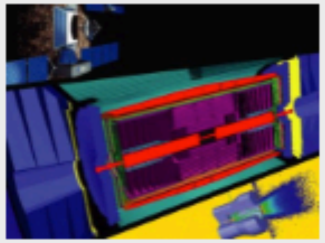
¿Cómo detectamos partículas?

Modelo estándar de partículas





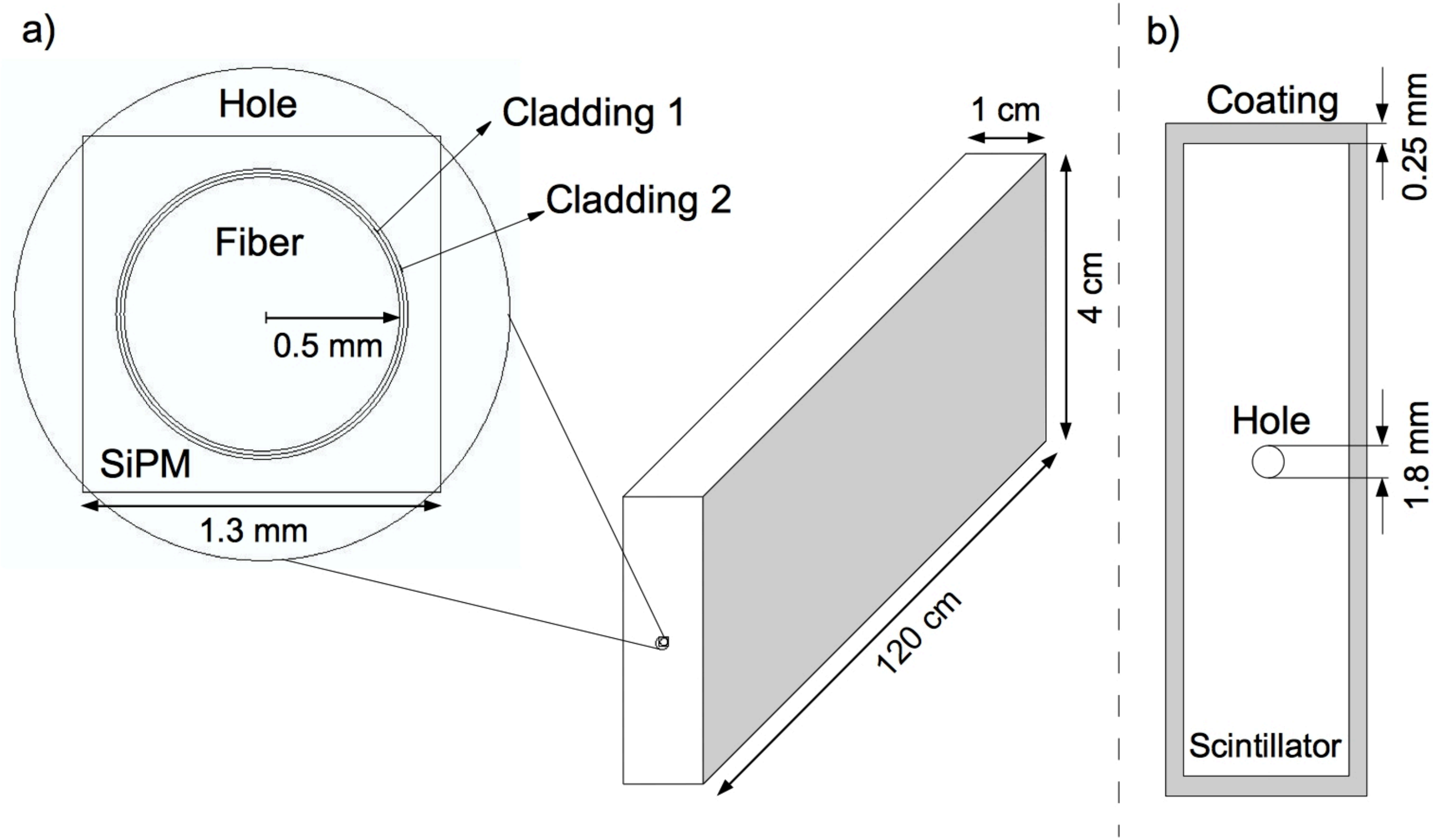
Aplicaciones

High Energy Physics	Space and Radiation	Medical	Technology Transfer
			

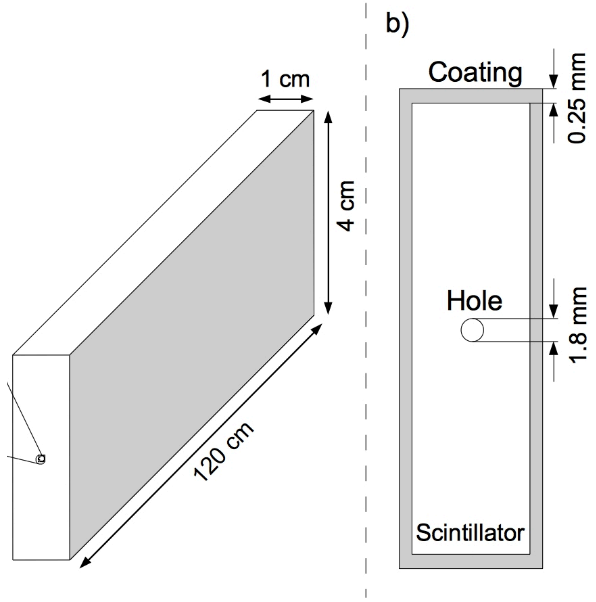
https://geant4.web.cern.ch/applications

Vamos a simular un Detector de Centelleo

1. Definir la geometría del centellador



1. Definir la geometría del centellador



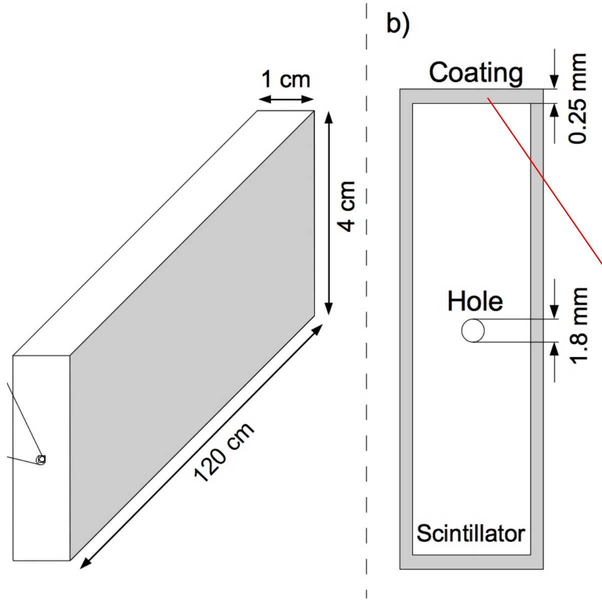

```
//-----------------
// World (air)
//-----------------
G4VSolid* solidWorld = new G4Box("World", fWorldSizeX, fWorldSizeY, fWorldSizeZ);

fLogicWorld = new G4LogicalVolume(solidWorld, Air, "World");

fPhysiWorld = new G4PVPlacement(0,G4ThreeVector(),fLogicWorld,"World",0,false,
                                0, checkOverlaps);
```

Dimensiones

Material

Ubicación

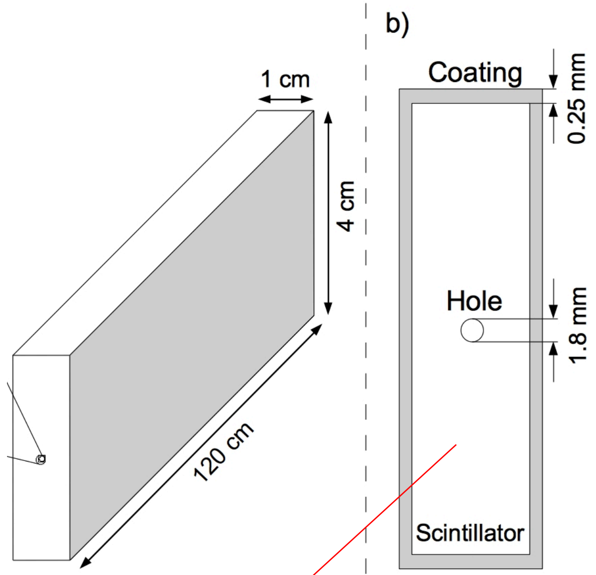
1. Definir la geometría del centellador




```
//-----------------
// World (air)
//-----------------
G4VSolid* solidWorld = new G4Box("World", fWorldSizeX, fWorldSizeY, fWorldSizeZ);

fLogicWorld = new G4LogicalVolume(solidWorld, Air, "World");

fPhysiWorld = new G4PVPlacement(0,G4ThreeVector(),fLogicWorld,"World",0,false,
                                0, checkOverlaps);
```

```
//--------------------
// Extrusion (TiO2)
//--------------------
G4VSolid* solidExtrusion =
      new G4Box("Extrusion",fBarBase,4*fBarBase,fBarLength);

G4LogicalVolume* logicExtrusion1 =
                    new G4LogicalVolume(solidExtrusion,
                                        Coating,
                                        "Extrusion1");
new G4PVPlacement(0,G4ThreeVector(0.*cm, 0.*cm, 0.*cm),logicExtrusion1,
                  "Extrusion1",fLogicWorld,false,0, checkOverlaps);
```

1. Definir la geometría del centellador

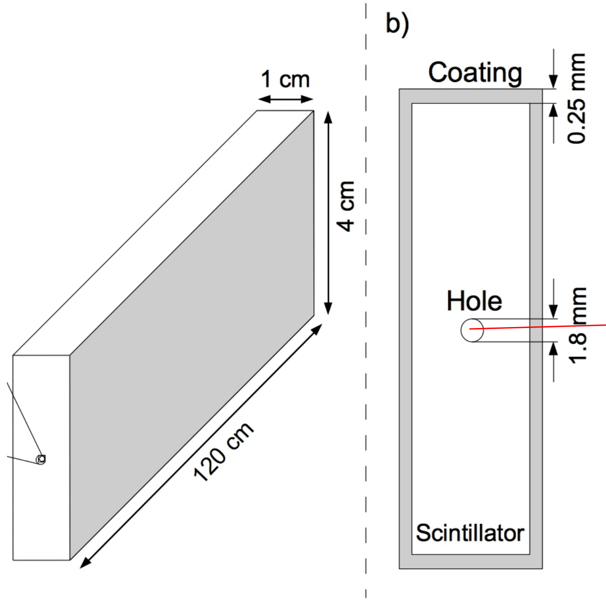



```
//-----------------
// World (air)
//-----------------
G4VSolid* solidWorld = new G4Box("World", fWorldSizeX, fWorldSizeY, fWorldSizeZ);

fLogicWorld = new G4LogicalVolume(solidWorld, Air, "World");

fPhysiWorld = new G4PVPlacement(0,G4ThreeVector(),fLogicWorld,"World",0,false,
                                0, checkOverlaps);
```

```
//-------------------
// Extrusion (TiO2)
//-------------------
G4VSolid* solidExtrusion =
      new G4Box("Extrusion",fBarBase,4*fBarBase,fBarLength);

G4LogicalVolume* logicExtrusion1 =
                         new G4LogicalVolume(solidExtrusion,
                                             Coating,
                                             "Extrusion1");
new G4PVPlacement(0,G4ThreeVector(0.*cm, 0.*cm, 0.*cm),logicExtrusion1,
                  "Extrusion1",fLogicWorld,false,0, checkOverlaps);
```

```
//---------------------------------------------------
// Scintillator (Polystyrene)
//---------------------------------------------------
G4VSolid* solidScintillator = new G4Box("Scintillator",
                              fBarBase-fCoatingThickness,
                              4*fBarBase-fCoatingThickness,
                              fHoleLength);

G4LogicalVolume* logicScintillator1 =
                          new G4LogicalVolume(solidScintillator,
                                              Polystyrene,
                                              "Scintillator1");

 new G4PVPlacement(0, G4ThreeVector(),  logicScintillator1,"Scintillator1",
                   logicExtrusion1,false,0, checkOverlaps);
```

1. Definir la geometría del centellador



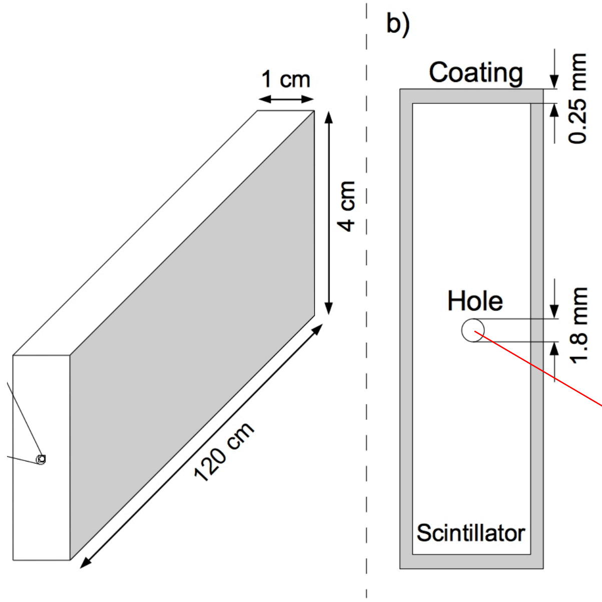

```
//------------------
// World (air)
//------------------
G4VSolid* solidWorld = new G4Box("World", fWorldSizeX, fWorldSizeY, fWorldSizeZ);

fLogicWorld = new G4LogicalVolume(solidWorld, Air, "World");

fPhysiWorld = new G4PVPlacement(0,G4ThreeVector(),fLogicWorld,"World",0,false,
                                0, checkOverlaps);
```

```
//--------------------
// Extrusion (TiO2)
//--------------------
G4VSolid* solidExtrusion =
      new G4Box("Extrusion",fBarBase,4*fBarBase,fBarLength);

G4LogicalVolume* logicExtrusion1 =
                     new G4LogicalVolume(solidExtrusion,
                                         Coating,
                                         "Extrusion1");
new G4PVPlacement(0,G4ThreeVector(0.*cm, 0.*cm, 0.*cm),logicExtrusi
                  "Extrusion1",fLogicWorld,false,0, checkOverlaps);
```

```
//----------------------------------
// Hole (air)
//----------------------------------
      G4VSolid* solidHole = new G4Tubs("Hole",
                                       0.0*cm,
                                       fHoleRadius,
                                       fHoleLength,
                                       0.*deg,
                                       360.*deg);

      fLogicHole1 = new G4LogicalVolume(solidHole, Air,
                                        "Hole1");


      fPhysiHole1 = new G4PVPlacement(0,
                                     G4ThreeVector(0.0,0.0,0.0),
                                     fLogicHole1,
                                     "Hole1",
                                     logicScintillator1,
                                     false,
                                     0, checkOverlaps);
```

```
//--------------------------------------------------
// Scintillator (Polystyrene)
//--------------------------------------------------
G4VSolid* solidScintillator = new G4Box("Scintillator",
                              fBarBase-fCoatingThickness,
                              4*fBarBase-fCoatingThickness,
                              fHoleLength);

G4LogicalVolume* logicScintillator1 =
                          new G4LogicalVolume(solidScintillator,
                                              Polystyrene,
                                              "Scintillator1");

 new G4PVPlacement(0, G4ThreeVector(),  logicScintillator1,"Scintillator1",
                   logicExtrusion1,false,0, checkOverlaps);
```

1. Definir la geometría del centellador




```
//-----------------
// World (air)
//-----------------
G4VSolid* solidWorld = new G4Box("World", fWorldSizeX, fWorldSizeY, fWorldSizeZ);

fLogicWorld = new G4LogicalVolume(solidWorld, Air, "World");

fPhysiWorld = new G4PVPlacement(0,G4ThreeVector(),fLogicWorld,"World",0,false,
                                0, checkOverlaps);
```

```
//-------------------
// Extrusion (TiO2)
//-------------------
G4VSolid* solidExtrusion =
      new G4Box("Extrusion",fBarBase,4*fBarBase,fBarLength);

G4LogicalVolume* logicExtrusion1 =
                     new G4LogicalVolume(solidExtrusion,
                                         Coating,
                                         "Extrusion1");
new G4PVPlacement(0,G4ThreeVector(0.*cm, 0.*cm, 0.*cm),logicExtrusion1,
                  "Extrusion1",fLogicWorld,false,0, checkOverlaps);
```

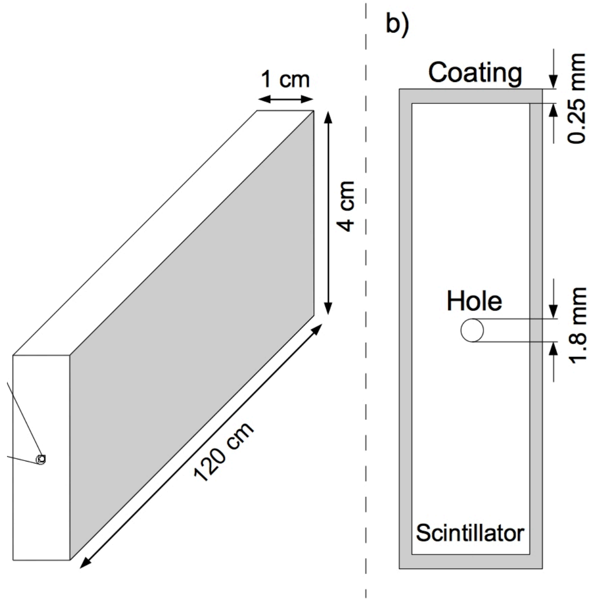
```
//--------------------------------
// Hole (air)
//--------------------------------
     G4VSolid* solidHole = new G4Tubs("Hole",
                                      0.0*cm,
                                      fHoleRadius,
                                      fHoleLength,
                                      0.*deg,
                                      360.*deg);

     fLogicHole1 = new G4LogicalVolume(solidHole, Air,
                                      "Hole1");


     fPhysiHole1 = new G4PVPlacement(0,
                                   G4ThreeVector(0.0,0.0,0.0),
                                   fLogicHole1,
                                   "Hole1",
                                   logicScintillator1,
                                   false,
                                   0, checkOverlaps);
```

```
//--------------------------------------------------
// Scintillator (Polystyrene)
//--------------------------------------------------
G4VSolid* solidScintillator = new G4Box("Scintillator",
                              fBarBase-fCoatingThickness,
                              4*fBarBase-fCoatingThickness,
                              fHoleLength);

G4LogicalVolume* logicScintillator1 =
                   new G4LogicalVolume(solidScintillator,
                                       Polystyrene,
                                        "Scintillator1");

 new G4PVPlacement(0, G4ThreeVector(),  logicScintillator1,"Scintillator1",
                   logicExtrusion1,false,0, checkOverlaps);
```

```
//--------------------------------------------------
// WLS Fiber
//--------------------------------------------------
G4VSolid* solidWLSfiber;

  solidWLSfiber =
     new G4Tubs("WLSFiber",0.,fWLSfiberRX,fWLSfiberZ,0.0*rad,twopi*rad);

 G4LogicalVolume* logicWLSfiber1 = new G4LogicalVolume(solidWLSfiber, PMMA,
 "WLSFiber1");

 G4VPhysicalVolume* physiWLSfiber1 = new G4PVPlacement(0,
                                 G4ThreeVector(0.0,0.0,fWLSfiberOrigin),
                                 logicWLSfiber1,
                                 "WLSFiber1",
                                 fLogicHole1,
                                 false,
                                 0, checkOverlaps);
```

Y faltan los 2 recubrimientos de la fibra y el SiPM

1. Definir la geometría del centellador




```
//------------------
// World (air)
//------------------
G4VSolid* solidWorld = new G4Box("World", fWorldSizeX, fWorldSizeY, fWorldSizeZ);

fLogicWorld = new G4LogicalVolume(solidWorld, Air, "World");

fPhysiWorld = new G4PVPlacement(0,G4ThreeVector(),fLogicWorld,"World",0,false
                                0, checkOverlaps);
```

```
//--------------------
// Extrusion (TiO2)
//--------------------
G4VSolid* solidExtrusion =
      new G4Box("Extrusion",fBarBas

G4LogicalVolume* logicEx

                    ));
                    0.*cm),logicExtrusion1,
                    d,false,0, checkOverlaps);
```

```
                      = new G4Tubs("Hole",
                                   0.0*cm,
                                   fHoleRadius,
                                   fHoleLength,
                                   0.*deg,
                                   360.*deg);

fLogicHole1 = new G4LogicalVolume(solidHole, Air,
                                  "Hole1");

fPhysiHole1 = new G4PVPlacement(0,
                                G4ThreeVector(0.0,0.0,0.0),
                                fLogicHole1,
                                "Hole1",
                                logicScintillator1,
                                false,
                                0, checkOverlaps);
```

```
//--------------------------
// Scintillator (Poly
//--------------------------
G4VSolid* solidScint
                    ckness,
                    oatingThickness,
                    ngth);

G4LogicalVolume* logicSc  tillator1 =
                  new G4LogicalVolume(solidScintillator,
                                      Polystyrene,
                                      "Scintillator1");

 new G4PVPlacement(0, G4ThreeVector(),  logicScintillator1,"Scintillator1",
                   logicExtrusion1,false,0, checkOverlaps);
```

```
//-----------------------------------------------------
// WLS Fiber
//-----------------------------------------------------
G4VSolid* solidWLSfiber;

  solidWLSfiber =
      new G4Tubs("WLSFiber",0.,fWLSfiberRX,fWLSfiberZ,0.0*rad,twopi*rad);

 G4LogicalVolume* logicWLSfiber1 = new G4LogicalVolume(solidWLSfiber, PMMA
 "WLSFiber1");

 G4VPhysicalVolume* physiWLSfiber1 = new G4PVPlacement(0,
                                     G4ThreeVector(0.0,0.0,fWLSfiberOrigin),
                                     logicWLSfiber1,
                                     "WLSFiber1",
                                     fLogicHole1,
                                     false,
                                     0, checkOverlaps);
```

No se asusten y admiren el nivel de detalle del GEANT4

2. Definir los materiales

```
// ----------------
// *** Elements ***
// ----------------
G4double a, z, density, fractionmass;

//N = new G4Element("Nitrogen", "N", z = 7 , a = 14.01*g/mole);
C = new G4Element("Carbon"   , "C", z = 6 , a = 12.01*g/mole);
O = new G4Element("Oxygen"   , "O", z = 8 , a = 16.00*g/mole);
H = new G4Element("Hydrogen", "H", z=1 , a = 1.01*g/mole);
Ti = new G4Element("Titanium", "Ti", z=22 , a = 47.867*g/mole);
```

2. Definir los materiales

```
// ----------------
// *** Elements ***
// ----------------
G4double a, z, density, fractionmass;

//N = new G4Element("Nitrogen", "N", z = 7 , a = 14.01*g/mole);
C = new G4Element("Carbon"   , "C", z = 6 , a = 12.01*g/mole);
O = new G4Element("Oxygen"   , "O", z = 8 , a = 16.00*g/mole);
H = new G4Element("Hydrogen", "H", z=1 , a = 1.01*g/mole);
Ti = new G4Element("Titanium", "Ti", z=22 , a = 47.867*g/mole);
```

Materiales a partir de elementos

```
//-----------------------------------------------------
// Polystyrene Material base del centellador
//-----------------------------------------------------
Polystyrene = new G4Material("Polystyrene", density= 1.050*g/cm3, 2);
Polystyrene->AddElement(C, 8);
Polystyrene->AddElement(H, 8);


//-----------------------------------------------------
// TiO2
//-----------------------------------------------------
TiO2 = new G4Material("TiO2", density= 4.26*g/cm3, 2);
TiO2->AddElement(Ti, 1);
TiO2->AddElement(O, 2);
```

2. Definir los materiales

```
// ----------------
// *** Elements ***
// ----------------
G4double a, z, density, fractionmass;

//N = new G4Element("Nitrogen", "N", z = 7 , a = 14.01*g/mole);
C = new G4Element("Carbon"   , "C", z = 6  , a = 12.01*g/mole);
O = new G4Element("Oxygen"   , "O", z = 8  , a = 16.00*g/mole);
H = new G4Element("Hydrogen", "H", z=1  , a = 1.01*g/mole);
Ti = new G4Element("Titanium", "Ti", z=22 , a = 47.867*g/mole);
```

Materiales a partir de elementos

```
//---------------------------------------------------
// Polystyrene Material base del centellador
//---------------------------------------------------
Polystyrene = new G4Material("Polystyrene", density= 1.050*g/cm3, 2);
Polystyrene->AddElement(C, 8);
Polystyrene->AddElement(H, 8);


//---------------------------------------------------
// TiO2
//---------------------------------------------------
TiO2 = new G4Material("TiO2", density= 4.26*g/cm3, 2);
TiO2->AddElement(Ti, 1);
TiO2->AddElement(O, 2);
```

Materiales a partir de otros materiales

```
//---------------------------------------------------
// Scintillator Coating
//---------------------------------------------------
Coating = new G4Material("Coating", density = 1.52*g/cm3, 2);
Coating->AddMaterial(TiO2, fractionmass = 15.0*perCent);
Coating->AddMaterial(Polystyrene, fractionmass = 85.0*perCent);
```

2. Definir los materiales

```
// ----------------
// *** Elements ***
// ----------------
G4double a, z, density, fractionmass;

//N = new G4Element("Nitrogen", "N", z = 7 , a = 14.01*g/mole);
C = new G4Element("Carbon"   , "C", z = 6 , a = 12.01*g/mole);
O = new G4Element("Oxygen"   , "O", z = 8 , a = 16.00*g/mole);
H = new G4Element("Hydrogen", "H", z=1 , a = 1.01*g/mole);
Ti = new G4Element("Titanium", "Ti", z=22 , a = 47.867*g/mole);
```

Materiales a partir de elementos

```
//--------------------------------------------------
// Polystyrene Material base del centellador
//--------------------------------------------------
Polystyrene = new G4Material("Polystyrene", density= 1.050*g/cm3, 2);
Polystyrene->AddElement(C, 8);
Polystyrene->AddElement(H, 8);


//--------------------------------------------------
// TiO2
//--------------------------------------------------
TiO2 = new G4Material("TiO2", density= 4.26*g/cm3, 2);
TiO2->AddElement(Ti, 1);
TiO2->AddElement(O, 2);
```

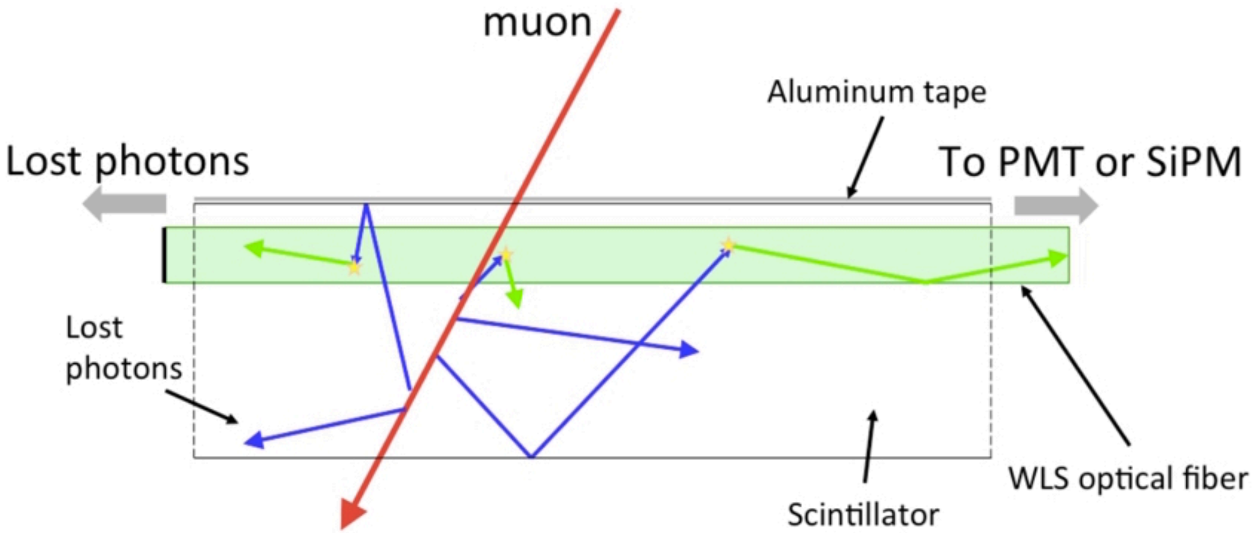
Algunos materiales ya están definidos en Geant4

```
// *** Air ***
Air = nist->FindOrBuildMaterial("G4_AIR");

// *** Aluminio ***
Aluminio = nist->FindOrBuildMaterial("G4_Al");
```

Materiales a partir de otros materiales

```
//--------------------------------------------------
// Scintillator Coating
//--------------------------------------------------
Coating = new G4Material("Coating", density = 1.52*g/cm3, 2);
Coating->AddMaterial(TiO2, fractionmass = 15.0*perCent);
Coating->AddMaterial(Polystyrene, fractionmass = 85.0*perCent);
```

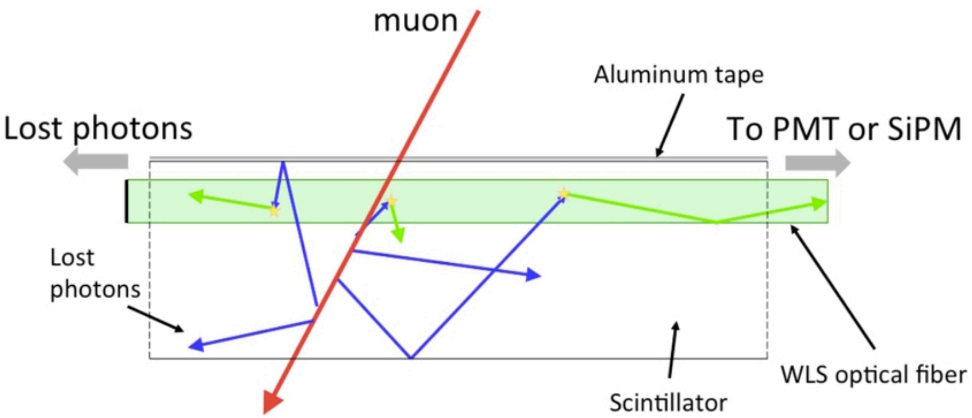
3. Definir los procesos y propiedades físicas



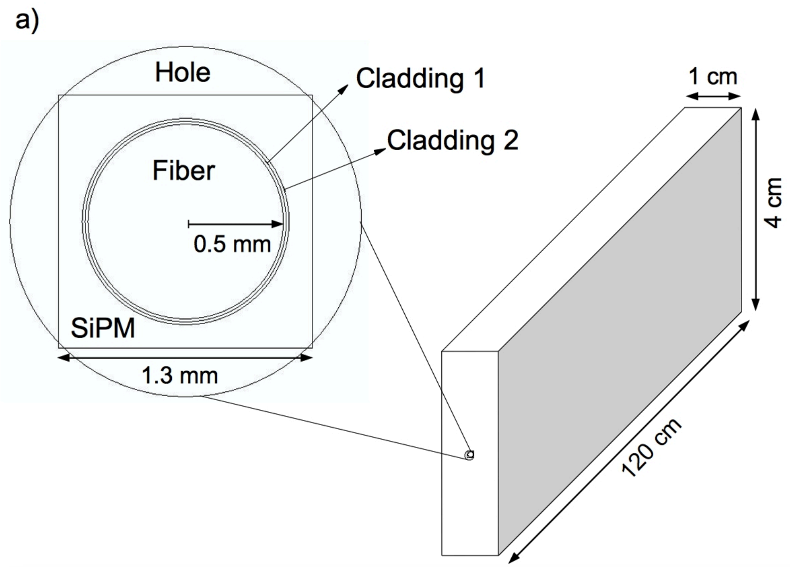
Activar procesos físicos según el detector

```
fOpticalPhysics->GetCerenkovProcess()->SetVerboseLevel(verbose);
fOpticalPhysics->GetScintillationProcess()->SetVerboseLevel(verbose);
fOpticalPhysics->GetAbsorptionProcess()->SetVerboseLevel(verbose);
fOpticalPhysics->GetRayleighScatteringProcess()->SetVerboseLevel(verbose);
fOpticalPhysics->GetMieHGScatteringProcess()->SetVerboseLevel(verbose);
fOpticalPhysics->GetBoundaryProcess()->SetVerboseLevel(verbose);
```

3. Definir los procesos y propiedades físicas



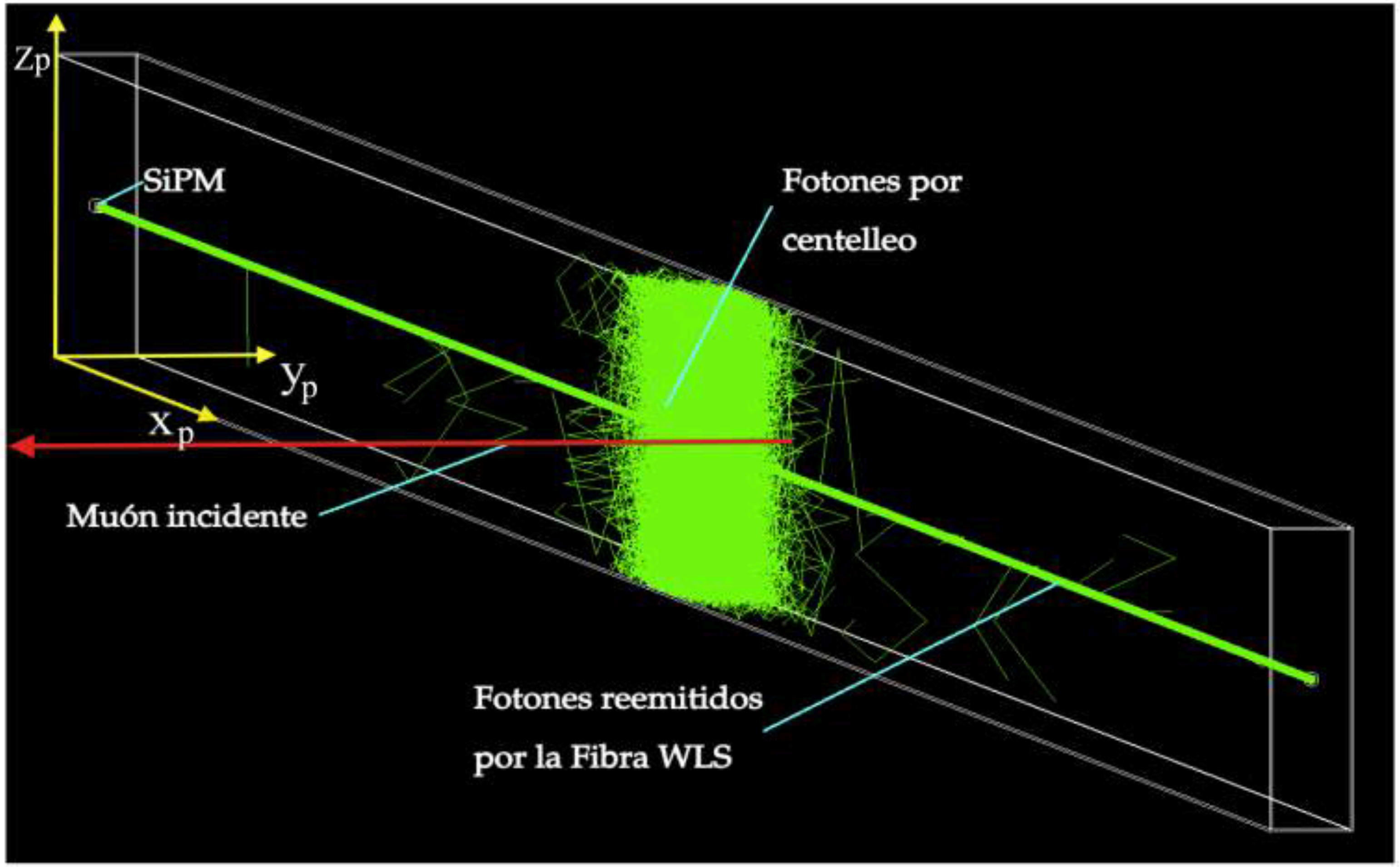
Activar procesos físicos según el detector



Propiedades físicas de los volúmenes

```
// Add entries into properties table
G4MaterialPropertiesTable* mptWLSfiber
    = new G4MaterialPropertiesTable();
mptWLSfiber->AddProperty("RINDEX",photonEnergy,
                         refractiveIndexWLSfiber,nEntries);
mptWLSfiber->AddProperty("WLSABSLENGTH",
                         photonEnergy,absWLSfiber,nEntries);
mptWLSfiber->AddProperty("WLSCOMPONENT",
                         photonEnergy,emissionFib,nEntries);
mptWLSfiber->AddConstProperty("WLSTIMECONSTANT", 0.5*ns);

PMMA->SetMaterialPropertiesTable(mptWLSfiber);
```

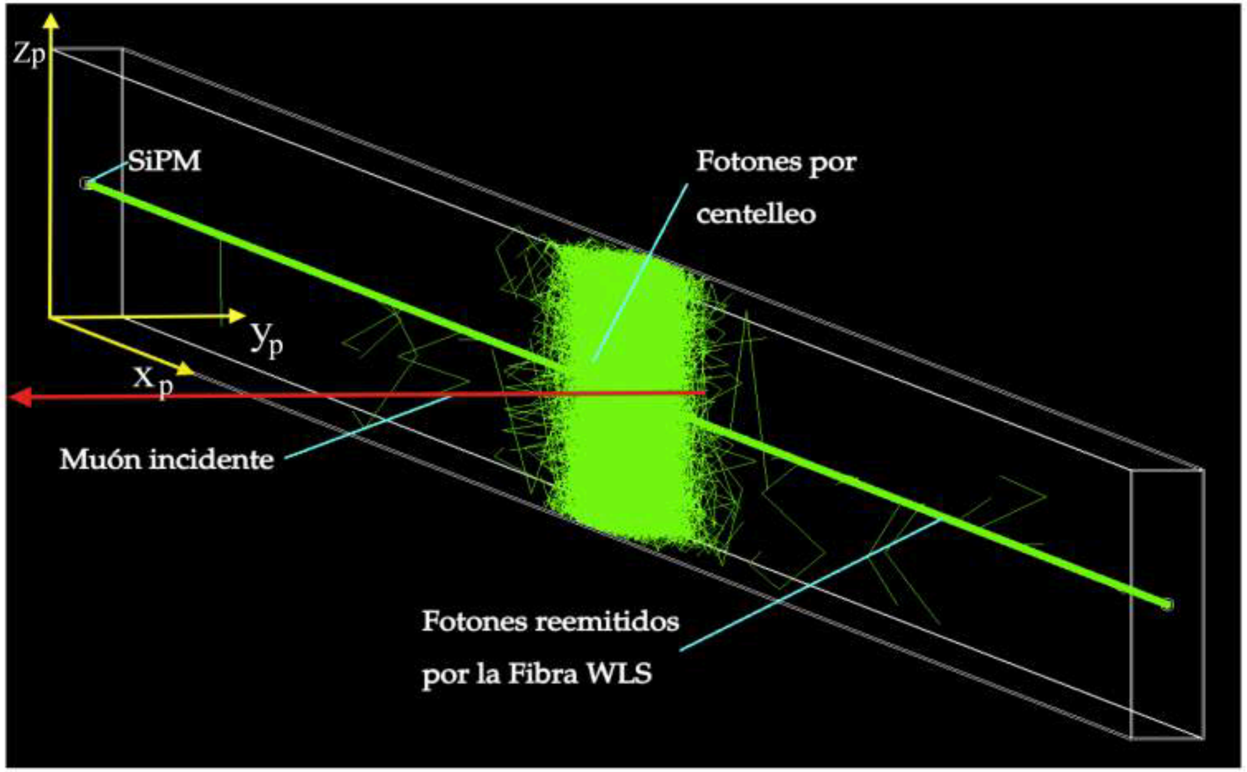
4. Escoger las condiciones iniciales



Partículas incidentes

Tipo: Muón
N partículas: 10^4
Energía: 1 GeV
Posición inicial: x_p = (2 + 4n) cm
y_p = 1 cm
z_p = 2 cm
Dirección: (0, -1, 0)

4. Escoger las condiciones iniciales



Partículas incidentes

Tipo: Muón
N partículas: 10^4
Energía: 1 GeV
Posición inicial: x_p = (2 + 4n) cm
y_p = 1 cm
z_p = 2 cm
Dirección: (0, -1, 0)

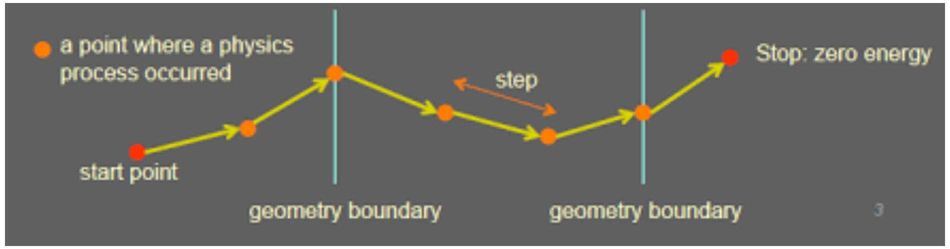
Tipo de partícula

Más partículas

Más energía

Más tiempo de cómputo!!!

5. Extracción de datos



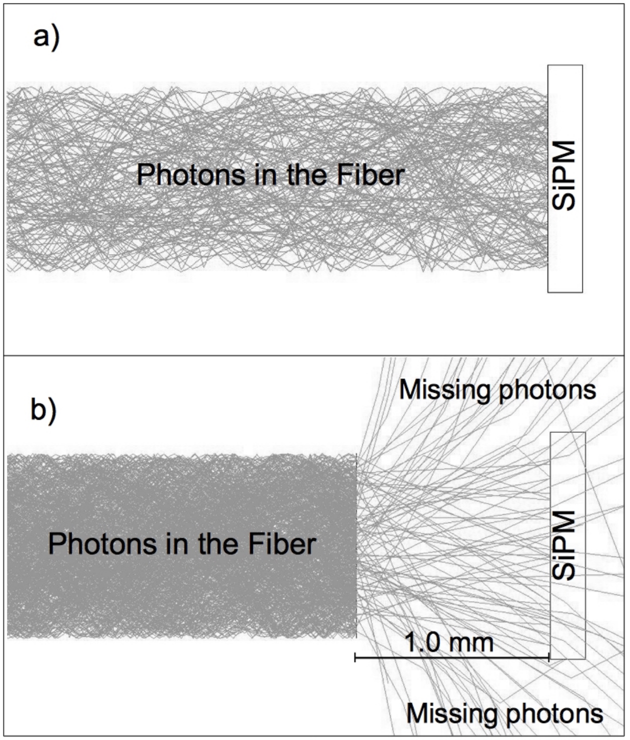
https://indico.cern.ch/event/472305/contributions/1982331/attachments/1223729/1790331/Tracking.pdf

Los datos de salida se muestran en cada interacción de la partícula:

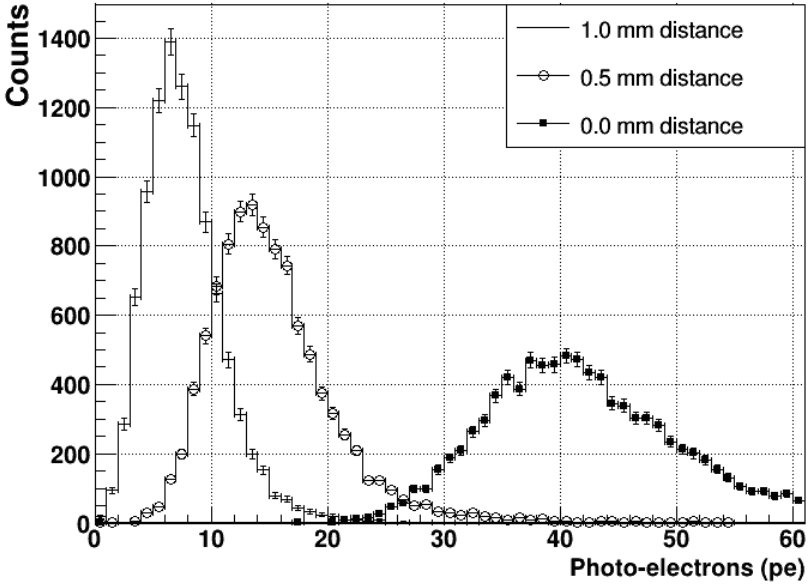
```
>>> Start event: 49

*********************************************************************************
* G4Track Information:   Particle = proton,   Track ID = 1,   Parent ID = 0
*********************************************************************************

Step#    X(mm)    Y(mm)    Z(mm) KinE(MeV)  dE(MeV) StepLeng TrackLeng  NextVolume ProcName
    0    0.125   -0.439     -150       100        0        0         0       World initStep
    1    0.125   -0.439       -5       100 1.59e-23      145       145        Cube Transportation
    2    0.125   -0.439     2.81      94.4     5.49     7.81       153        Cube hIoni
    3    0.123   -0.442     3.19      93.9    0.342    0.384       153        Cube hIoni
    4    0.119    -0.45      4.1      93.1     0.71    0.913       154        Cube hIoni
    5    0.118   -0.453     4.35      92.8     0.16    0.249       154        Cube hIoni
    6    0.117   -0.461        5      92.4    0.434    0.647       155       World Transportation
    7    -5.46      -31    2e+03      92.4 2.32e-22    2e+03  2.15e+03  OutOfWorld Transportation
```

- Tipo de partícula
- Primaria o secundaria
- Posición
- Energía actual
- Cambio de energía
- Distancia recorrida
- Volumen donde está
- Tipo de interacción / Proceso Físico

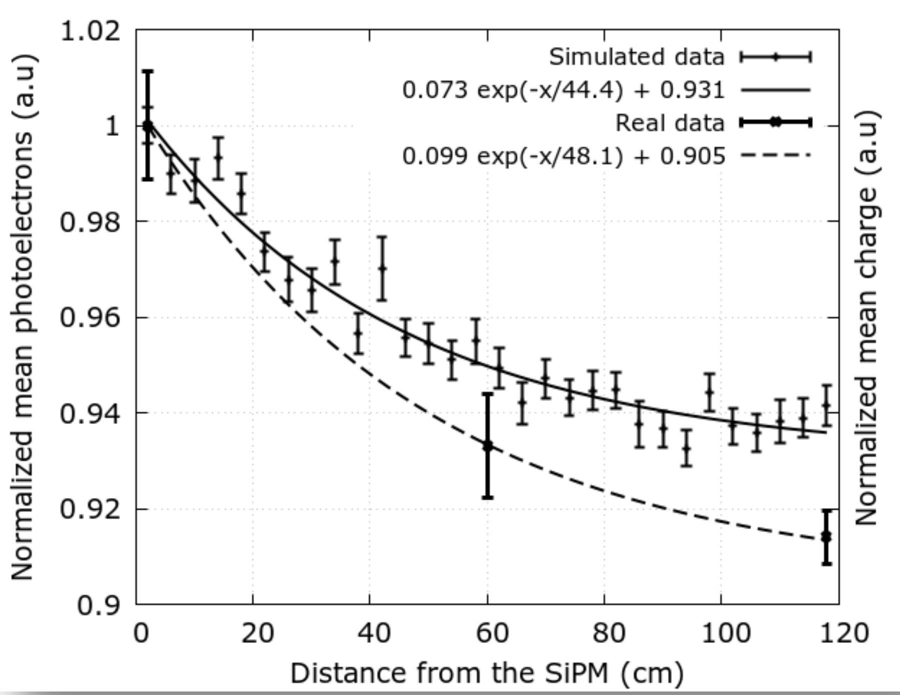
5. Extracción de datos



Pérdida de señal debido a un mal acople entre la fibra y el SiPM



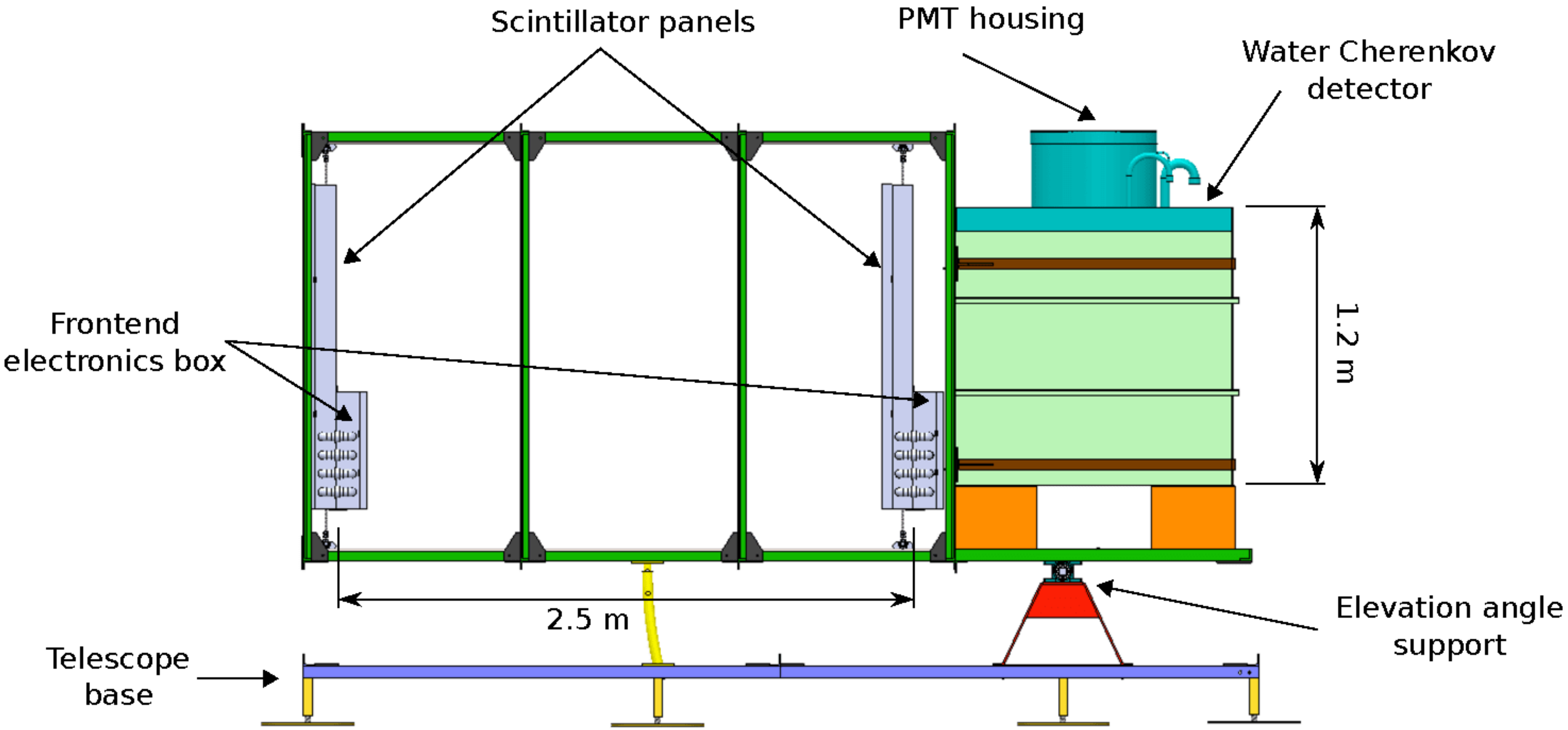
Atenuación del número de fotones respect a la posición de impacto del muón en el centellador



A. Vásquez-Ramírez *et al* 2020 *JINST* **15** P08004

https://iopscience.iop.org/article/10.1088/1748-0221/15/08/P08004/meta

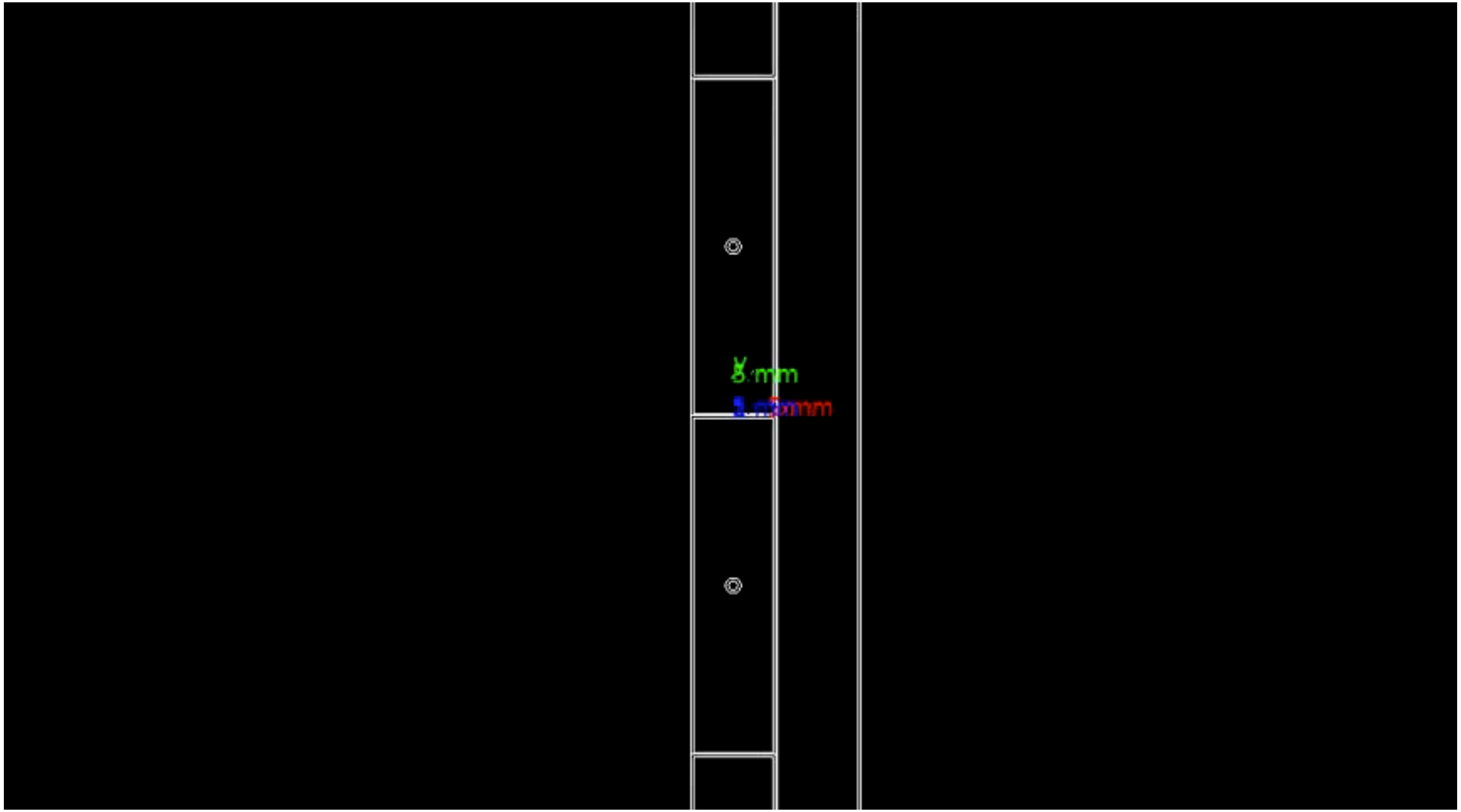
Acabamos de simular la respuesta de una barra centelladora del MuTe!!



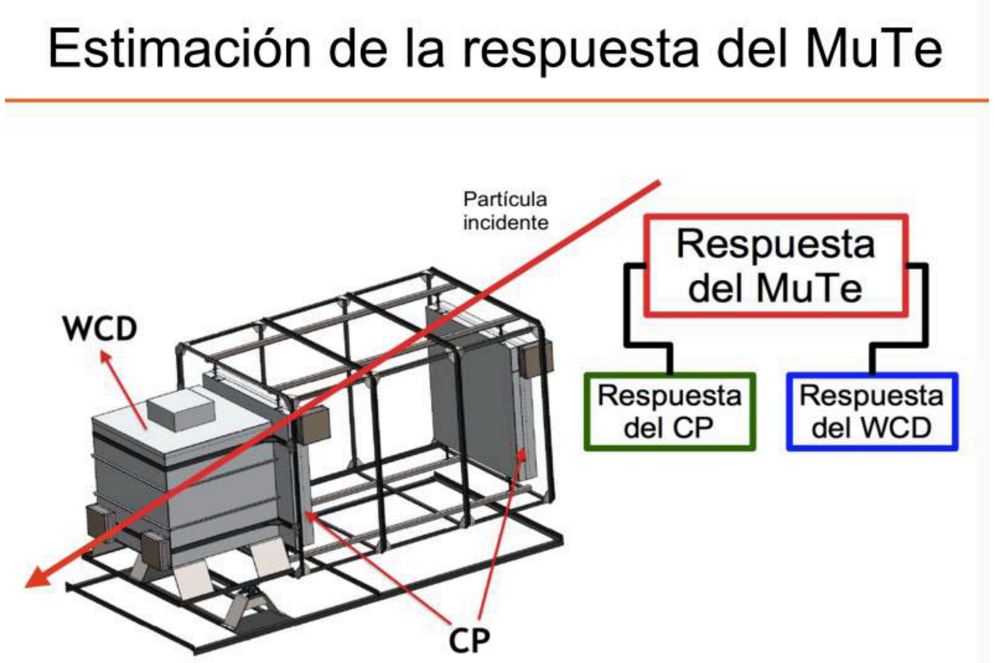
Estos son los pasos básicos para simular otro(s) detector(es)

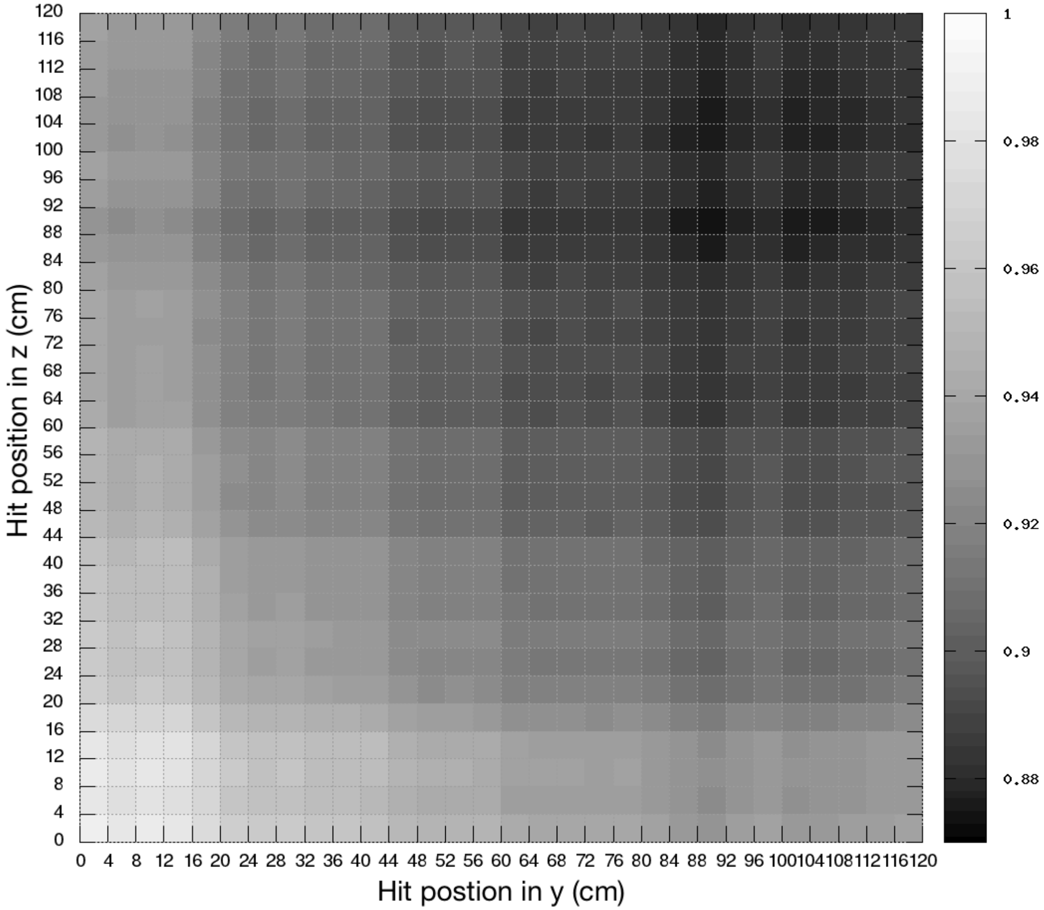
https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/html/GettingStarted/gettingStarted.html

Simulación del panel de centelleo del MuTe



Simulación del panel de centelleo del MuTe

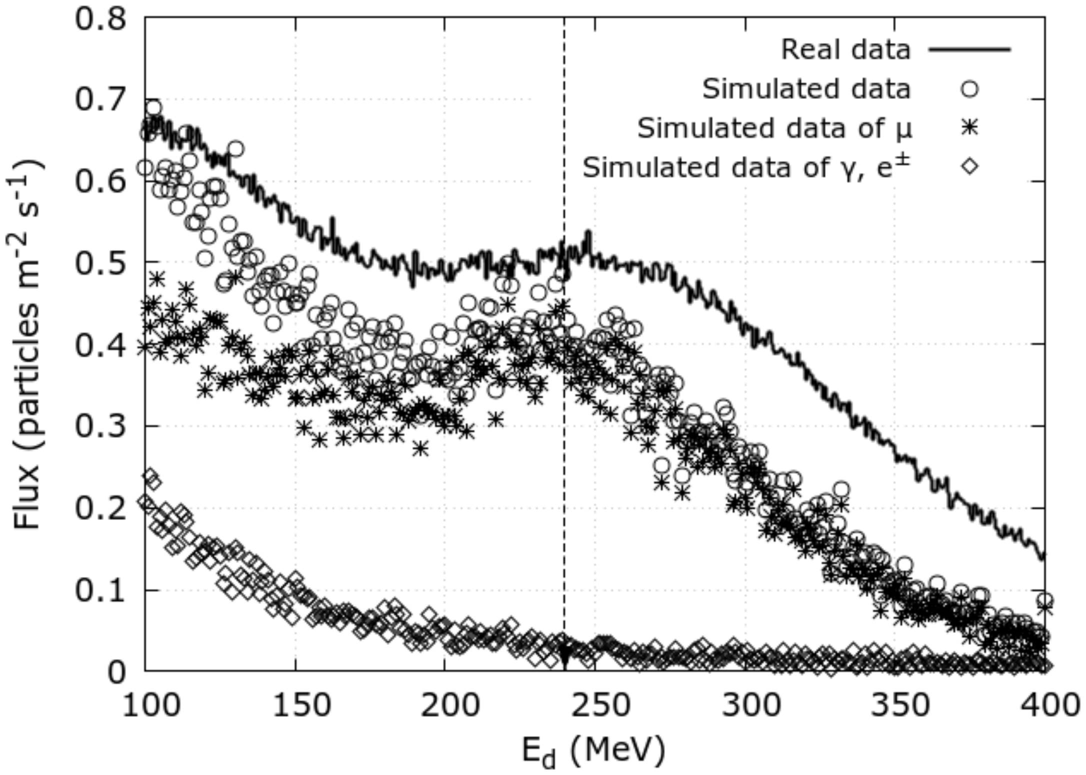




A. Vásquez-Ramírez *et al* 2020 *JINST* **15** P08004

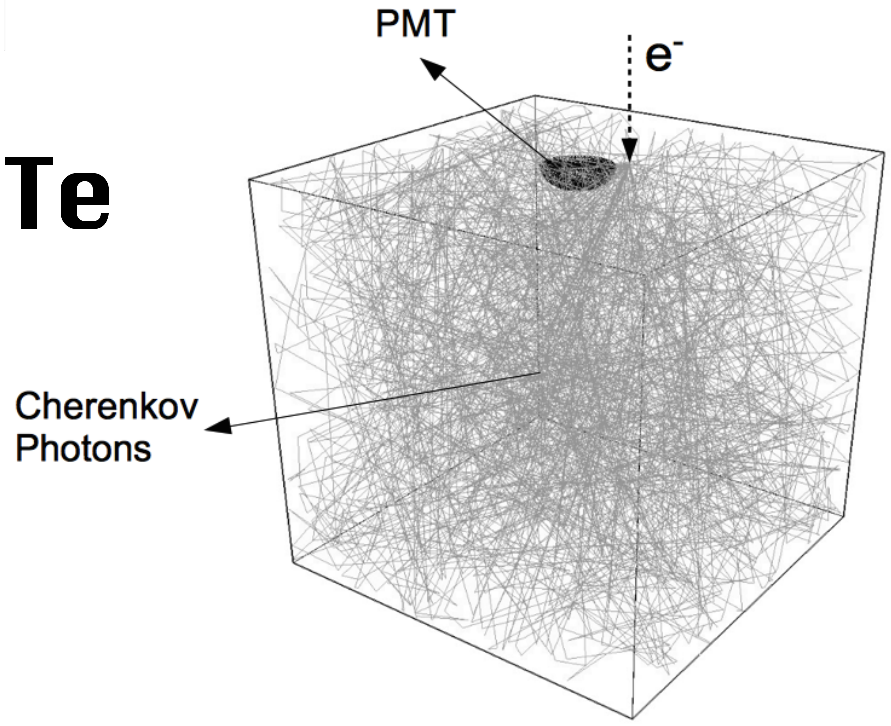
https://iopscience.iop.org/article/10.1088/1748-0221/15/08/P08004/meta

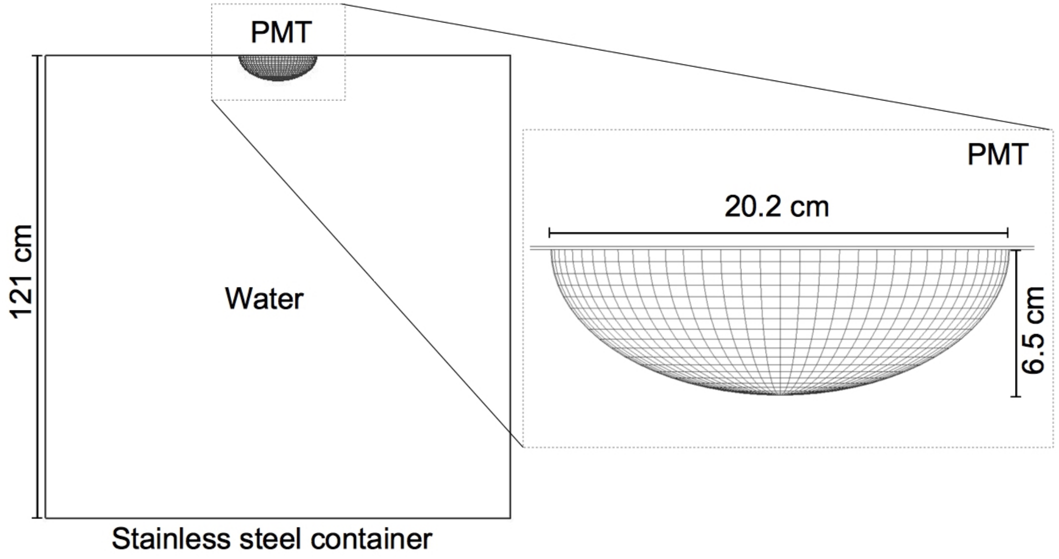
Simulación del WCD de MuTe



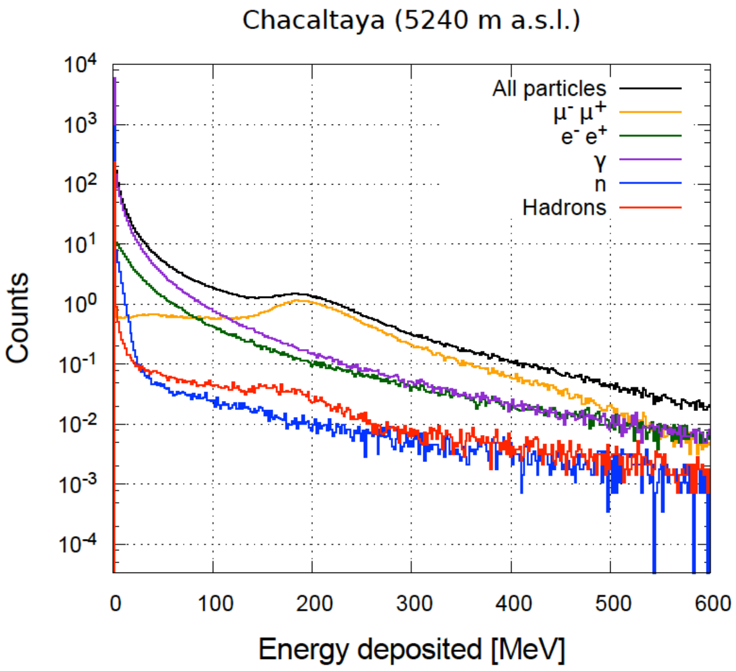
A. Vásquez-Ramírez *et al* 2020 *JINST* **15** P08004

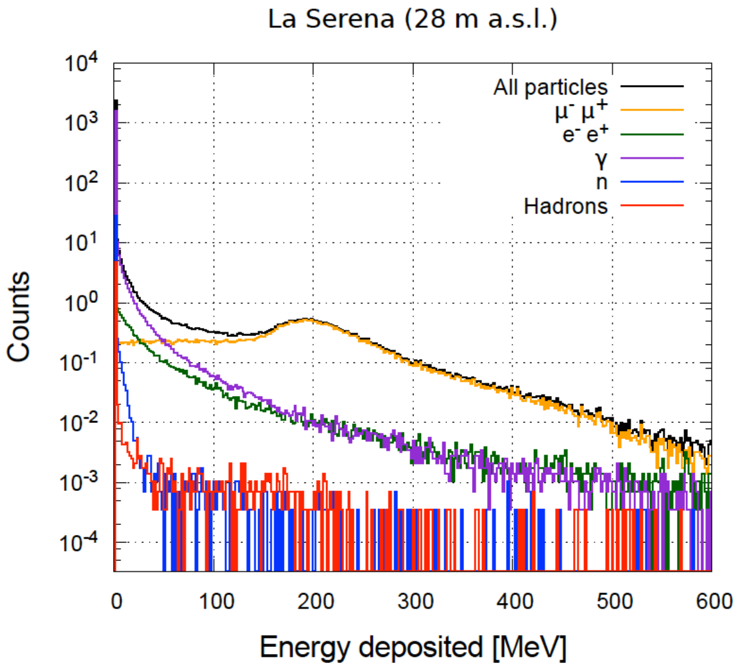
https://iopscience.iop.org/article/10.1088/1748-0221/15/08/P08004/meta





Simulación de WCDs en varias ubicaciones del Latin American Giant Observatory (LAGO)





C. Sarmiento-Cano *et al* 2019 *PoS ICRC2019 412*

https://pos.sissa.it/358/412/pdf