

**Universidad de Nariño**  
**Facultad de Ciencias Exactas y Naturales**  
**Departamento de Física**



**SIMULACIÓN PRELIMINAR DE LA POSICIÓN ÓPTIMA DE  
INSTALACIÓN DE UN DETECTOR DE MUONES ATMOSFÉRICOS  
SOBRE EL VOLCÁN GALERAS**

**TRABAJO DE GRADO**

Para optar al título profesional de:

Físico

**DANILO ALEJANDRO ARTURO RODRIGUEZ**

San Juan de Pasto, Colombia

AGOSTO 2017

**Universidad de Nariño**  
**Facultad de Ciencias Exactas y Naturales**  
**Departamento de Física**

**SIMULACIÓN PRELIMINAR DE LA POSICIÓN ÓPTIMA DE  
INSTALACIÓN DE UN DETECTOR DE MUONES ATMOSFÉRICOS  
SOBRE EL VOLCÁN GALERAS**

**DANILO ALEJANDRO ARTURO RODRIGUEZ**

**TRABAJO DE GRADO**

Director:

**M.Sc. JAIME ALFREDO BETANCOURT**

San Juan de Pasto, Colombia

AGOSTO 2017

©2017 - DANILO ALEJANDRO ARTURO RODRIGUEZ

“Las ideas y conclusiones aportadas en la tesis de grado son responsabilidad exclusiva de los autores”

Artículo 1. del acuerdo No. 324 del 11 de Octubre de 1966, emanado por el Honorable  
Consejo Directivo de la Universidad de Nariño.

Todos los derechos reservados.

Nota de Aceptación

---

---

---

---

---

---

---

Jaime Alfredo Betancourt MS.c

Director

David Martínez Caicedo Ph.D

Jurado

Alex Tapia Casanova Ph.D

Jurado

San Juan de Pasto, AGOSTO 2017

## **Agradecimientos**

Agradezco a mi familia absolutamente por todo, especialmente a mi padre Luis Eduardo Arturo, a mi madre Mercedes Rodríguez Ruíz, a mi hermano Omar Andrés Rodríguez y mi abuela Cecilia Ruíz.

Agradezco también al grupo de investigación de altas energías en física experimental y mis amigos quienes compartieron su conocimiento sin dudar y me apoyaron en el desarrollo de este proyecto de grado, especialmente a Ms.C. Jaime Betancourth, Ph.D. Alex Tapia, Ph.D David Martinez, y Jairo Rodríguez.

*Gracias*

*A mis padres:*

*Luis Eduardo Arturo*

*Mercedes Rodríguez Ruíz*

# **SIMULACIÓN PRELIMINAR DE LA POSICIÓN ÓPTIMA DE INSTALACIÓN DE UN DETECTOR DE MUONES ATMOSFÉRICOS SOBRE EL VOLCÁN GALERAS**

## **Resumen**

*La interacción de los rayos cósmicos con la atmósfera produce lluvias atmosféricas, que son fuentes de muones. Estas partículas pueden utilizarse para diversas aplicaciones; entre otras la tomografía de volcanes.*

*A partir de la simulación de un detector de centelleo y de la geometría del volcán Galerías (VG) en GEANT4 [1], se realizó la simulación de calibración del detector con una fuente radioactiva de Cobalto 60. En el mismo software, se dispuso el detector en un punto con coordenadas geográficas de ubicación óptima previamente estudiado sobre la geometría del VG para realizar una muongrafía. Posteriormente, con base en simulaciones de lluvias atmosféricas extendidas (EAS) realizadas en CORSIKA [2] para la altura de la ciudad de Pasto, se realizó la producción del flujo de muones en función del ángulo cenital y a diferentes energías que atraviesa la estructura geológica y arriba al detector de centelleo, obteniendo la energía depositada en el detector por la partícula de interés.*

**Palabras claves:** *muón, muongrafía, detector, GEANT4, calibración, coordenadas geográficas, CORSIKA, energía.*

# PRELIMINARY SIMULATION OF THE OPTIMAL POSITION OF DEPLOYMENT FOR A DETECTOR OF ATMOSPHERIC MUONS ON GALERAS VOLCANO

## Abstract

*The interaction of cosmic rays with the atmosphere produce Extensive Air Showers (EAS), which provide abundant source of muons. These particles can be used in multiple applications, as an example to tomography of different volcanoes edifice.*

*From the simulation of a scintillation detector and the geometry of the Galeras volcano (GV) in GEANT4 [1], a simulation of calibration for the detector was performed with a radioactive source of Cobalt 60. In the same software, the detector was deployment at a point with geographic coordinates of optimal position previously studied on the geometry of the GV to perform a muonography. Subsequently, based on simulations of Extensive Air Showers (EAS) performed in CORSIKA [2] to the height of the city of Pasto, a muon flux was done as a function of the zenith angle and to different energies that cross the geological structure and hit the scintillation detector, obtaining the energy deposited in the detector by the particle of interest.*

**Keywords:** *muon, muonography, detector, GEANT4, calibration, geographical coordinates, CORSIKA, energy*



# Contenido

Título . . . . .	I
Aceptación . . . . .	IV
Agradecimientos . . . . .	V
Dedicatoria . . . . .	VI
Resumen . . . . .	VII
Abstract . . . . .	VIII
Índice de Figuras . . . . .	XI
<b>1. Introducción</b>	<b>1</b>
<b>2. Muones producidos en lluvias atmosféricas extendidas</b>	<b>3</b>
2.1. Lluvias atmosféricas extendidas . . . . .	3
2.2. Producción de muones en lluvias atmosféricas extendidas . . . . .	6
2.2.1. Componente electromagnética . . . . .	7
2.2.2. Componente hadrónica . . . . .	7
2.2.3. Componente muónica . . . . .	7
2.3. Técnica de Muongrafía . . . . .	9
2.3.1. Tomografía con muones a volcanes . . . . .	10
2.4. Simulación de lluvias atmosféricas utilizando CORSIKA . . . . .	12
<b>3. Simulaciones previas de un detector de centelleo y el perfil volcánico</b>	<b>14</b>
3.1. GEANT4 . . . . .	14
3.1.1. Platarforma GEANT4 . . . . .	15
3.2. Simulación de un detector de centelleo en GEANT4 . . . . .	17
3.3. Simulación del Volcán Galéras en GEANT4 . . . . .	18
<b>4. Calibración del detector de centelleo</b>	<b>20</b>
4.1. Calibración . . . . .	20
4.2. Cobalto 60 ( $C_{O60}$ ) . . . . .	21
4.3. Simulación de la calibración del detector de centelleo . . . . .	22
<b>5. Ubicación del detector de centelleo sobre el volcán Galeras</b>	<b>27</b>
5.1. Geometrías en una sola simulación: Detector-Volcán . . . . .	28
5.2. Puntos óptimos para la ubicación del detector de centelleo . . . . .	31
5.3. Simulación del detector en un punto óptimo sobre el VG . . . . .	33
5.4. Flujo de muones que atraviesa el cono volcánico y arriba al detector . . . . .	37

---

<b>6. Resultados</b>	<b>41</b>
6.1. CORSIKA . . . . .	41
6.2. GEANT4 . . . . .	44
6.2.1. Calibración del detector de centelleo . . . . .	44
6.2.2. Interacción de muones con el cráter volcánico y el detector . . . . .	46
<b>7. Conclusiones y Recomendaciones</b>	<b>52</b>
<b>ANEXOS</b> . . . . .	<b>54</b>
<b>A. Código del programa</b>	<b>54</b>
A.1. Macro CORSIKA . . . . .	54
A.2. GEANT4 . . . . .	55
A.2.1. Headers (Carpeta include) . . . . .	55
A.2.2. Clases Usuario (Carpeta src) . . . . .	67
<b>Bibliografía</b>	<b>94</b>

# Índice de Figuras

2.1. Flujo de rayos cósmicos de la energía donde se observar el cambio en el espectro [4]. . . . .	5
2.2. componentes de una EAS generadas por la interacción de los rayos cósmicos con las capas altas de la atmósfera terrestre [4]. . . . .	6
2.3. Ubicación de un detector de muones en el túnel de un glaciar ubicado en Suiza [9]. . . . .	10
3.1. Simulación Detector de centelleo . . . . .	17
3.2. Simulación de interacciones de muones con un detector de centelleo . . . . .	18
3.3. Simulación del volcán Galeras en GEANT4 . . . . .	19
4.1. Un neutrón impacta sobre un núcleo de $C_{o59}$ el cual se convierte en $C_{o60}$ (inestable) que se desintegrará emitiendo $\beta, \bar{\nu}_e, \gamma$ [26] . . . . .	22
4.2. Geometría esférica de una fuente de $C_{o60}$ . . . . .	24
4.3. Interacción de una fuente de $C_{o60}$ con el detector de centelleo . . . . .	26
5.1. Ubicación del volcán Galeras y poblaciones a su alrededor [3] . . . . .	28
5.2. Primera simulación del detector de centelleo con el volcan Galeras . . . . .	30
5.3. Acercamiento al detector con el volcán Galeras . . . . .	30
5.4. Puntos más convenientes para la disposición del detector de muones[3] . . . . .	32
5.5. Punto de ubicación sobre el cual se realiza la simulación[3] . . . . .	32
5.6. coordenadas cartesianas orginales del VG . . . . .	33
5.7. translación del VG al origen del sistema de coordenadas de GEANT4 [1] . . . . .	34
5.8. Conversión de coordenadas del punto 3 [27] . . . . .	35
5.9. Simulación de la ubicación óptima del detector de centelleo sobre el complejo volcánico . . . . .	36
5.10. Ángulo de inclinación del detector sobre el complejo volcánico . . . . .	37
5.11. Interacción de un muón con energía de 1TeV con el volcán Galeras y el detector de centelleo. . . . .	39
5.12. Interacción de un muón con energía de 1TeV con el cráter volcánico que arriba al detector . . . . .	40
6.1. Distribución lateral de muones que arriban a la ciudad de Pasto generada por protones de 5 y 100 TeV para los ángulos $30^\circ$ y $60^\circ$ [14] . . . . .	41
6.2. Espectro de energía de muones generada por protones de 5 y 100 TeV para los ángulos $30^\circ$ y $60^\circ$ . [14] . . . . .	42

---

6.3. Distribución lateral de muones que arriban a la ciudad de Pasto generada por núcleos de hierro de 5 y 100 TeV para los ángulos 30° y 60°[14] . . . . .	42
6.4. Espectro de energía de muones generada por núcleos de hierro de 5 y 100 TeV para los ángulos 30° y 60°[14] . . . . .	43
6.5. Parametros de ajuste para la MLDF a la altura de la ciudad de Pasto con protones como partículas primarias[14] . . . . .	43
6.6. Parametros de ajuste para la MLDF a la altura de la ciudad de Pasto con núcleos de hierro como partículas primarias[14] . . . . .	44
6.7. Procesos físicos generados por la interacción de la fuente de $Co_{60}$ con el detector . . . . .	44
6.8. Decaimiento de la fuente de $Co_{60}$ . . . . .	45
6.9. Picos característicos de radiación gamma . . . . .	45
6.10. Procesos físicos producido por las interacciones de partículas con el detector centelleo. . . . .	46
6.11. Procesos físicos producido por las interacciones de muones con el detector centelleo. . . . .	47
6.12. Energía depositada por las partículas en el detector . . . . .	48
6.13. Zoom: Energía depositada por las partículas en el detector . . . . .	48
6.14. Zoom: Energía depositada por muones que arriban al detector . . . . .	49
6.15. Proyección de muones que atraviesan el volcán y arriban al detector (cráter compuesto por aire) . . . . .	50
6.16. Proyección de muones que atraviesan el volcán y arriban al detector (cráter compuesto por roca) . . . . .	50

# Capítulo 1

## Introducción

Los rayos cósmicos son partículas provenientes del espacio exterior, algunas de ellas son más energéticas que cualquier otra partícula observada por el hombre en la naturaleza. Apesar de no conocer sus fuentes, bombardean constantemente la Tierra desde todas direcciones e interactúan con los núcleos de la atmósfera produciendo una cascada de partículas secundarias denominadas "lluvias atmosféricas extendidas" ó en inglés "Extensive Air Shower (EAS)" que llegan hasta la misma corteza terrestre. Entre las partículas generadas en las lluvias atmosféricas se encuentran los muones, los cuales poseen escasa interacción con la materia, y su energía se disipa lentamente a través de procesos de ionización. Debido a que los muones de alta energía son una de las formas naturales de radiación a la que estamos expuestos diariamente y pueden penetrar grandes distancias de roca antes de detenerse, hacen que la partícula sea de interés para muchas aplicaciones, como la tomografía de volcanes que posibilita estudiar la estructura interna e intentar predecir erupciones.

La muografía mediante la detección de muones de cascadas de secundarios generadas por rayos cósmicos nos permite la observación del interior de objetos, sin la necesidad de dañarlos, o acceder físicamente a los mismos. Esta técnica puede ser aplicada sobre diferentes estructuras geológicas (volcanes). Los volcanes resultan de gran interés debido al riesgo que presentan para la población que vive en sus proximidades, como ocurre en muchas partes del mundo, por ejemplo: en Nápoles con el Monte Vesubio o en Pasto con el Volcán Galeras que percibe su presencia como la de un guardián que podría atacar en cualquier momento.

El volcán Galeras localizado en la Ciudad de San Juan de Pasto es uno de los volcanes más activos de Colombia, con una edad de 4.500 años. Su localización es aproximadamente a 9 km al occidente de la ciudad, en las coordenadas  $1^{\circ} 13' 43,8''$  de latitud norte,  $77^{\circ} 21'$

33,0" de longitud oeste y con una altura de 4276 msnm[3]. Debido al alto riesgo que presenta para las poblaciones a su alrededor, su actividad es vigilada constantemente.

Actualmente, está en proceso de desarrollo el proyecto de muongrafía para el Volcán Galéras, con el fin de comprender la estructura interna y la dinámica presente durante una erupción. Se ha diseñado en el programa de interacción radiación-materia GEANT4 [1] un detector de muones de barras centelleadoras, que permita la detección de estas partículas al atravesar el volcán Galeras y se ha incorporado la geometría del VG con las curvas de nivel proporcionadas por el Servicio Geológico Colombiano (SGC).

Posteriormente, se ha realizado una simulación para la calibración del detector utilizando una fuente radioactiva y se llevó a cabo un estudio de los puntos más convenientes para ubicar el detector. Finalmente, a través de diferentes pasos que son mencionados durante el desarrollo de este proyecto, las dos geometrías tanto la del volcán como la del detector con sus características, se implementan en una sola simulación de GEANT4 [1].

## Capítulo 2

# Muones producidos en lluvias atmosféricas extendidas

### Lluvias atmosféricas extendidas

Cuando una partícula primaria alcanzan la atmósfera en su capa superior interactúa con las moléculas presentes generando una cascada de partículas secundarias. Los rayos cósmicos fueron descubiertos por el físico Austriaco Victor Hess en 1912 cuando ascendido a bordo de un globo aerostático con un electroscopio para medir la ionización a grandes alturas, logró notar que a medida que aumentaba la altura, el electroscopio se descargaba con una mayor rapidez considerando fuente de radiación al Sol, por lo que repitió el experimento durante un eclipse de sol, a partir de las comparaciones realizadas, Hess se arriesgó a anunciar que la radiación que descarga los electroscopios proviene del exterior, y que dicha radiación está compuesta por las partículas cargadas propuestas por Faraday y no tendría su origen en la Tierra sino en el cosmos las cuales bombardearían continuamente nuestra atmósfera.

Posteriormente, otro investigador, David Millikan (1868-1953), acuñaría el término rayos cósmicos por su evidente origen en el espacio exterior, y rayos gamma porque sospechaba que se trataba de rayos gamma, que en esa época era la radiación electromagnética más penetrante conocida. Pero fue creciendo la evidencia de que estos rayos eran en su mayoría partículas energéticas con masa.

Las energías encontradas para las partículas que entran a la atmósfera son mayores a los  $10^{20}$  eV ( electronvoltios ) cien millones de veces superior a la que se puede impartir a una

partícula subatómica en los más potentes aceleradores de partículas.

El espectro de los rayos cósmicos se refiere al flujo de partículas como función de la energía y su comportamiento puede ser ajustado a una ley de potencias dada por:

$$\frac{dN}{dE} \propto E^{-\alpha} \quad (2.1)$$

Donde N es el flujo de rayos cósmicos, E la energía de las partículas y  $\alpha$  conocido como índice espectral.

El espectro de energías de rayos cósmicos presenta variaciones en el flujo, las cuales son asociadas muchas veces al origen de los rayos cósmicos. Estas variaciones acotan 3 regiones en el espectro que vienen indicadas por el cambio en el índice espectral  $\alpha$ ; dos de estas regiones denominadas como “rodilla” y la tercera como "tobillo". Se asocia el "tobillo" para energías superiores a  $E \sim 10^{18,5}$  eV es decir  $\alpha = -2,7$  (partículas de origen extragaláctico), mientras que la primera “rodilla” para  $E \sim 10^{15,5}$  eV donde  $\alpha = -3,1$ , hace referencia a rayos cósmicos de origen galáctico, y con ausencia de elementos pesados en la segunda rodilla  $E \sim 10^{17,7}$  eV con  $\alpha = -3,3$ . La figura 2.1 muestra el espectro construido a partir de los datos proporcionados por diferentes experimentos.



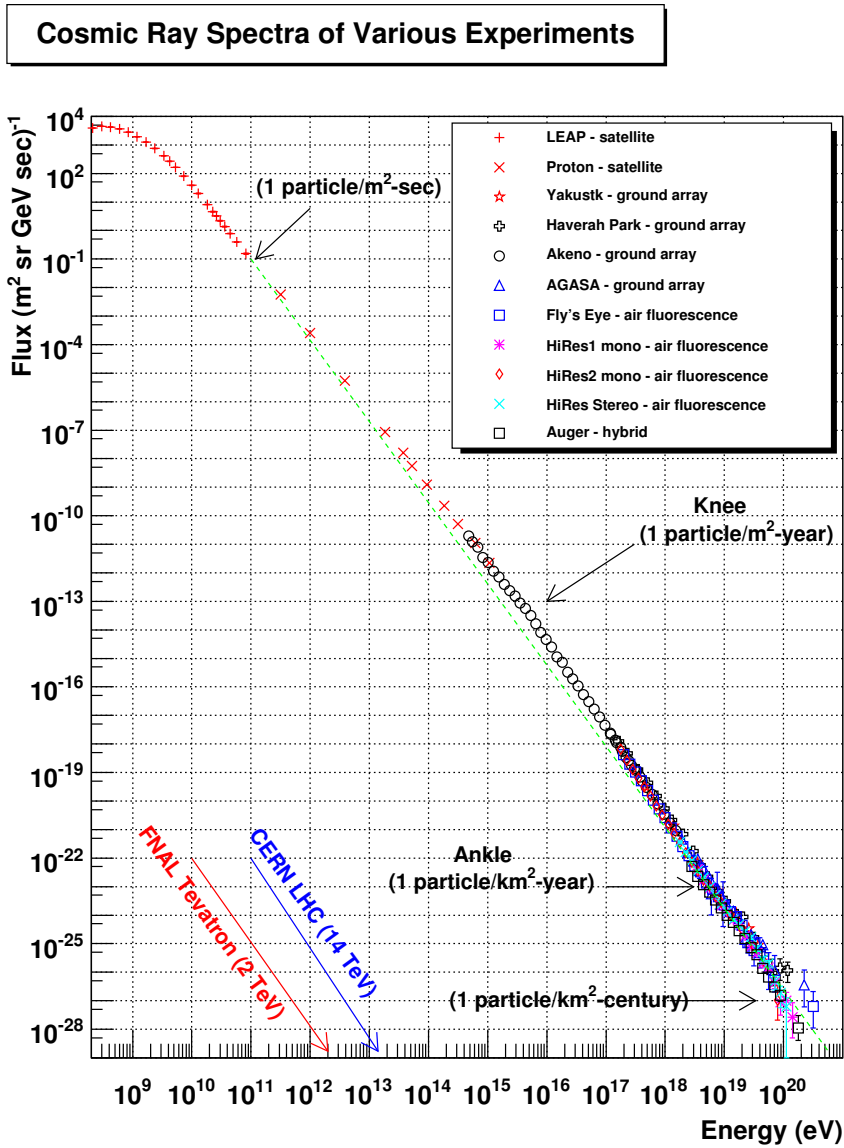


Figura 2.1: Flujo de rayos cósmicos de la energía donde se observar el cambio en el espectro [4].

## Producción de muones en lluvias atmosféricas extendidas

La atmósfera terrestre interactúa con los rayos cósmicos, en este proceso se excitan los átomos presentes en la atmósfera ( $N_2$ ,  $O_2$  y  $CO_2$ ) y genera nuevas partículas (partículas secundarias). Las partículas secundarias colisionan con partículas de la atmósfera, que dan origen a nuevas partículas. El proceso se repite generando una cascada, perdiendo energía con cada interacción, hasta que alcanzan una energía límite que impide nuevas interacciones. Así, puede formarse una cascada de más de  $10^{11}$  nuevas partículas.

Las EAS iniciadas por un primario genera tres componentes: la componente hadrónica, la electromagnética y la muónica. (Fig. 2.2).

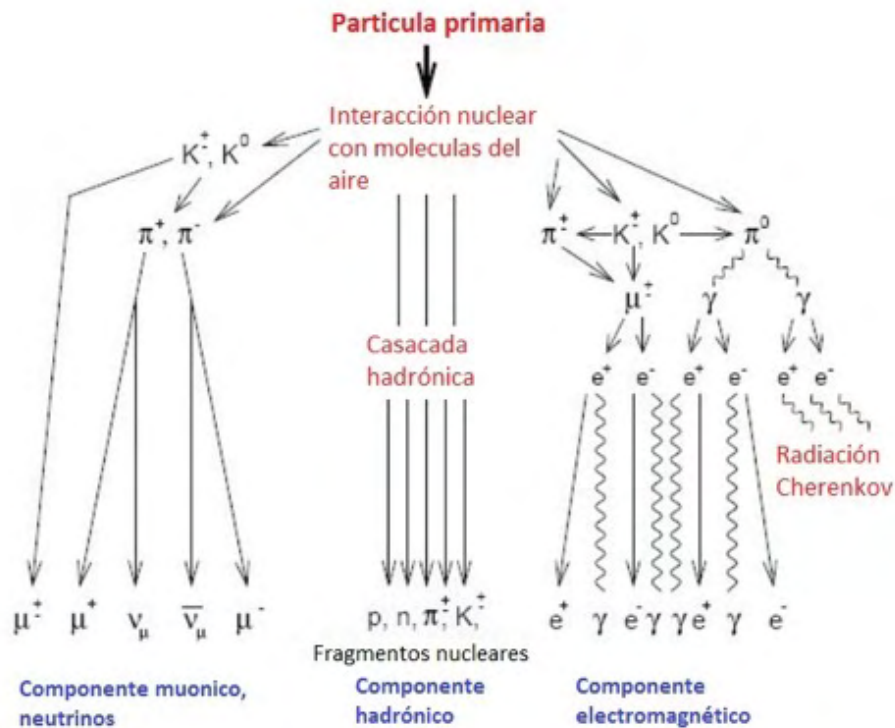


Figura 2.2: componentes de una EAS generadas por la interacción de los rayos cósmicos con las capas altas de la atmósfera terrestre [4].

## Componente electromagnética

La componente electromagnética es la parte más abundante de las EAS. Si el primario es un fotón o electrón, la cascada de secundarios presentará una componente electromagnética. Esta consta de electrones y fotones producidos principalmente por la interacción de piones neutros, donde el decaimiento de esta partícula es:

$$\pi^0 \rightarrow \gamma + \gamma$$

Los fotones generados forman la producción de pares:

$$\gamma \rightarrow e^+ + e^-$$

que a lo largo de la EAS producirán mas fotones:

$$e^\pm \rightarrow e^\pm + \gamma$$

## Componente hadrónica

Los hadrones que no se llegan a desintegrar forman la componente hadrónica. Contiene aproximadamente el 1 % de las partículas de la cascada, y está constituida por piones, kaones y bariones[5]. Las primeras interacciones con la atmósfera generan piones neutros  $\pi^0$  que alimentan la componente electromagnética; piones y kaones cargados ( $K^\pm$ ,  $\pi^\pm$ ), producen partículas como neutrones, neutrinos y muones.

## Componente muónica

Los muones y neutrinos muónicos son producidos en su mayoría por decaimiento de mesones, principalmente por  $K^\pm$  y  $\pi^\pm$ . Debido a que el muón es nuestra partícula de interés, a continuación se muestran los decaimientos que generan la componente muónica de las EAS con los respectivos porcentajes:

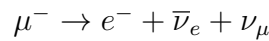
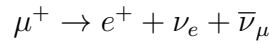
$$\pi^\pm \rightarrow \mu^\pm + \nu_m \quad [99.99 \%]$$



En comparación con la componente electromagnética, los muones debilmente interaccionan y solamente pierden una pequeña fracción de su energía por ionización.

El muón es una partícula elemental inestable que obedece el principio de exclusión de Pauli, con tiempo de vida de 2.2  $\mu\text{s}$  y forma parte de los leptones. Los muones generados en lluvias atmosféricas extendidas pueden interactuar con los núcleos atómicos por Bremsstrahlung o creación de pares.

Cuando la energía que lleva un muón es demasiado baja para permitir nuevas interacciones o alcanzan el limite de tiempo e vida, puede decaer formando nuevas partículas como electrones y neutrinos:



A continuación, se presenta las propiedades fundamentales de los muones [6]:

	$\mu^+$	$\mu^-$
Charge	+1	-1
Mass	206.768 277 (24) ( $m_e$ ) <sup>a</sup>	105.659 (1) (MeV/c <sup>2</sup> ) <sup>b</sup>
Spin	1/2	1/2
Magnetic moment ( $\mu_\mu/\mu_p$ )	3.183 345 13 (39) <sup>a</sup>	
Gyromagnetic ratio ( $\mu_\mu/2\pi I$ in $\text{kHz}/G$ , $I = 1/2$ )	13.553 42	
Gyromagnetic factor ( $g/2$ ) ( $\mu_\mu = g(e\hbar/2m_\mu)I$ , $I = 1/2$ )	1.001 165 920 3 (15) <sup>c</sup>	1.001 165 936 (12)
Free decay lifetime ( $10^{-6}$ s)	2.196 95 (6) <sup>d</sup> 2.197 078 (73) <sup>e</sup>	2.194 8 (10) <sup>f</sup> (in flight)
Decay mode	$e^+ + \bar{\nu}_\mu + \nu_e$ $e^+ + \gamma$ $e^+ + e^- + e^+$ $e^+ + \gamma + \gamma$	(100%) ( $<1.7 \times 10^{-10}$ ) ( $<1.9 \times 10^{-9}$ ) ( $<1.25 \times 10^{-8}$ )

Tabla 1: Propiedades fundamentales de los muones [6].

## **Técnica de Muongrafía**

Los muones más energéticos son capaces de atravesar grandes espesores de materiales tales como roca, metales, etc. Sin embargo a medida que atraviesan capas van perdiendo intensidad debido a procesos como radiación de frenado, producción de pares y las interacciones electromagnéticas. Si dichos muones son medidos por un detector, se obtendrá una “tomografía muónica”. Si se dispone de varios detectores en distintas posiciones se obtiene mucha más información con lo que se puede realizar una reconstrucción de una imagen 3D (tomografía) de aquello que haya atravesado nuestra partícula de interés.

El nacimiento de la tomografía de muones se remonta a 1950, fecha en la que se utilizó por primera vez para detectar la profundidad de un túnel en Australia[A5]. El primer experimento que trató de utilizar esta técnica para detectar cámaras ocultas en las pirámides de Egipto fue realizado por el físico Luis Walter Alvarez, quien utilizó un detector de muones e intentó localizar nuevas cámaras en la pirámide de Kefrén [8]. En ese entonces el sistema era muy primitivo y la medición falló, pero demostró que ésta técnica era muy eficiente a la hora de determinar el compuesto de un material.

Hoy en día, la tomografía de muones es una técnica con muchas aplicaciones especialmente para el estudio de estructuras geológicas. En Suiza, se está realizando un estudio sobre la interacción de los muones con un glaciar, por donde pasa un túnel de ferrocarril. Teniendo en cuenta su composición de hielo y roca se estudia la dependencia del flujo de muones con el ángulo cenital con que arriban las partículas al detector, el cual está ubicado en unas coordenadas previamente estudiadas [9] (Fig. 2.3).

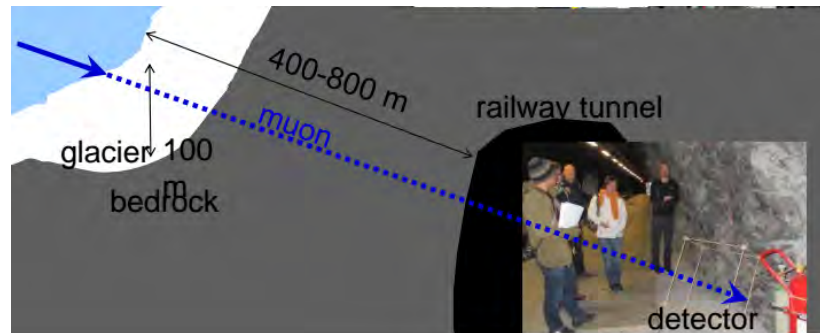


Figura 2.3: Ubicación de un detector de muones en el túnel de un glaciar ubicado en Suiza [9].

### Tomografía con muones a volcanes

Un Volcán es una formación geológica que consiste en una fisura en la corteza terrestre sobre la cual se acumula un cono de materia volcánica. Las partes básicas de un volcán son:

**Cámara de magma:** Se encuentra debajo del volcán, y es el sitio donde el magma se deposita.

**Cráter:** Está ubicado en la parte superior. Por aquí se expulsa la lava, los gases y demás materiales.

**Chimenea:** Conducto por donde suben los materiales del interior de la Tierra.

**Cono volcánico:** Es lo que da forma a la estructura, y por lo regular se desarrolla durante el lapso de varios años como consecuencia del depósito y solidificación de los materiales en la superficie.

**Fisuras:** Grietas alargadas por donde puede liberarse lava o gases.

Dependiendo del período de actividad de un volcán y el crecimiento poblacional a sus alrededores, representa un peligro para los habitantes. Las consecuencia que trae las erupciones

volcánicas, se pueden encontrar en nuestros volcanes colombianos. Por ejemplo el Volcán Galeras, que en los últimos 500 años de actividad, se identifica la ocurrencia de 27 eventos eruptivos eminentemente explosivos, dos de ellos con características efusivas, que se puede corroborar por los depósitos geológicos de 1866. Cabe resaltar que en 1993, mató a un grupo de científicos que se encontraban dentro de su cráter cuando estalló.

Los muones energéticos que penetran un volcán nos permiten sondear la estructura interna del mismo. lo cual puede contribuir a realizar una predicción temprana de una erupción y conocer más a fondo la dinámica de la actividad volcánica.

Para realizar una muongrafía volcánica, como primer paso se debe estimar el flujo de muones y su energía con que arriban al lugar sobre el cuál se ubica el volcán; lo cual se puede realizar mediante el software CORSIKA[2], que simula EAS originadas por rayos cósmicos de alta energía. Luego, se estudia las pérdidas energéticas de los muones en la materia, con el fin de determinar la mínima energía necesaria para penetrar la estructura geológica. En este proyecto, la geometría del volcán está conformada por roca estándar ( $\rho = 2,65gcm^{-3}$ ). La pérdida de energía en éstas partículas se da debido a la ionización y por interacción con las partículas que conforman el material, dicha pérdida puede ser resumida por [10]:

$$-\frac{dE}{d\varrho} = a(E) + b(E)E \quad (2.2)$$

La ecuación anterior es expresada en  $MeVg^{-1}cm^2$  Donde  $a(E)$  es la pérdida de energía debido a ionización y  $b(E)$  a procesos como creación de pares y Bremsstrahlung.  $\varrho$  es la densidad integrada a lo largo de la trayectoria de los muones, definida como:

$$\varrho(L) = \int_L \rho(\xi)d\xi \quad (2.3)$$

Donde  $\varrho$  se expresa en  $g^{-1}cm^2$ ,  $\rho$  es la densidad y  $\xi$  es la coordenada medida a lo largo de la trayectoria L del muón.

Al conocer la pérdida de energía de los muones a través de la roca, es posible determinar la energía inicial mínima ( $E_{min}$ ), necesaria para que un muón cruce una opacidad dada antes de golpear el detector. Prácticamente, se encuentra resolviendo

$$E_{mnima} - \int_0^{\varrho} \frac{dE}{d\varrho} d\varrho = E_{\mu} \quad (2.4)$$

Donde,  $E_{\mu}$  es la energía de masa en reposo. Los muones con energía de aproximadamente 30 GeV pueden ser detectados para un espesor de roca de unos 50 m.

Finalmente, se mide el flujo muónico que penetra la estructura geológica y arriba al detector, el cual está ubicado en un punto óptimo para la detección de las partículas, para poder comparar y estudiar los resultados con los de las simulaciones realizadas.

## Simulación de lluvias atmosféricas utilizando CORSIKA

CORSIKA [2] es un programa utilizado para la simulación detallada de EAS iniciadas por rayos cósmicos de alta energía. Partículas como protones, núcleos de hierro, fotones y muchas más son denominados primarios.

Para el desarrollo de este proyecto se usó una simulación de EAS para diferentes ángulos cenitales  $\theta = 0^\circ, 30^\circ, 45^\circ$  y  $60^\circ$  con energías de el primario (protones y hierro) de 5, 10 y 100 TeV para obtener la distribución lateral de muones (MLD) y el espectro de energía con el cual arriban a la altura de la ciudad de Pasto (2527 msnm), dentro de las simulaciones se usó los paquetes de interacción hadrónica QGSJET-II-04 [11] y para bajas energías GHEISHA2002d [12] con un modelo de atmósfera tropical y con las componentes de campo magnético  $B_x$  y  $B_z$  generadas por la calculadora de campo geomagnético [13] para la ciudad de Pasto. [14]

Las MLDs obtenidas de las simulaciones se comportan de acuerdo a la ecuación de Nishimura- Kamata-Greisen [15] [16] para el arreglo de los detectores de KASKADE-Grande [17].



$$\rho_{\mu} = N_{\mu} \left(\frac{r}{r_0}\right)^{-\alpha} \left(1 + \frac{r}{r_0}\right)^{-\beta} \left(1 + \left(\frac{r}{10r_0}\right)^2\right)^{-\gamma}$$

Donde  $N_{\mu}$ ,  $r_0$ ,  $\alpha$ ,  $\beta$  y  $\gamma$  son parámetros de ajustes.  $r_0$ ,  $\alpha$  y  $\gamma$  son parámetros fijos con los valores de 320m, 0.75 y 3 respectivamente [18].  $N_{\mu}$  Y  $\beta$  se obtiene del ajuste.

Esta parte del estudio fue presentado en ICHEP 2016 realizado en la ciudad de Chicago, EE.UU [14].

# Capítulo 3

## Simulaciones previas de un detector de centelleo y el perfil volcánico

Con el software GEANT4 [1] se diseñó un detector de centelleo con ciertas características y se modeló el perfil del volcán Galeras a partir de las curvas de nivel proporcionadas por el SGC [3].

### GEANT4

GEANT4 [1] es un conjunto de herramientas que nos permite simular con precisión el paso de partículas a través de materia; siendo útil en muchas áreas de la física como: física de altas energías, nuclear, de aceleradores, espacial, médica y muchas más.

Todos los aspectos del proceso de simulación se han incluido en el kit de herramientas:

- La geometría del sistema
- El material en cuestión
- Las partículas fundamentales de interés
- La generación de eventos primarios
- El seguimiento de las partículas a través de los materiales y los campos electromagnéticos
- Los procesos dominantes que gobiernan las interacciones de las partículas
- El almacenamiento de eventos y trackings
- La visualización de las trayectorias de partículas
- El análisis de datos de la simulación en diferentes niveles de detalle.

Los procesos físicos ofrecidos cubren una gama amplia, incluyendo procesos electromagnéticos, hadrónicos y ópticos, un gran conjunto de partículas, materiales y un amplio rango de energía, desde los eV hasta los TeV.

GENAT4 [1] está escrito en lenguaje de programación C++ y es el resultado de una colaboración mundial de físicos e ingenieros.

## **Platarforma GEANT4**

Como se ha dicho anteriormente, Geant4 está escrito en C++, por lo que cualquier aplicación se programa creando clases compuestas por una biblioteca *archivo.hh* y la definición de la clase mediante el *archivo.cc* de forma paralela.

Para simular un proyecto en GEANT4 se necesitan conocer tres conceptos, la geometría del detector u objeto que va a ser atravesado por las partículas, el tipo de partículas que atravesarán dicha geometría y los procesos físicos aplicables. Por este motivo, una aplicación realizada en Geant4 requiere un archivo principal (**main**) y tres archivos (**clases**) obligatorios los cuales declaran la geometría de un objeto, los procesos físicos y el lanzamiento de partículas. Los dos primeros son clases de inicialización de simulación y el último es la clase que establece las acciones de lanzamiento que se llevan a cabo. Con mayor detalle:

**G4VUserDetectorConstruction:** Esta biblioteca define la geometría del objeto que será atravesado por las partículas.

En esta biblioteca se define la clase “DetectorConstructor” que engloba las funciones necesarias para definir los materiales de los que está formado el detector y el material que forma el espacio vacío, por ejemplo *aire*. Es necesario crear un mundo y dentro de él crear la geometría del detector. se define la geometría usando tres referencias:

- SolidGeometry: Define la estructura física y su tamaño.
- LogicalGeometry: Define el material que conforma la SolidGeometry
- PhysicGeometry: Ubica la geometría en un punto específico del mundo.

**G4VUserPhysicsList:** Esta biblioteca se utiliza para definir las partículas que se quiere que aparezcan en la simulación y los procesos físicos aplicables. La definición de partículas se hace en función de los procesos que vayan a suceder; si una partícula producto de una interacción no está definida, el proceso no ocurrirá.

Una vez definidas las partículas hay que asociar procesos físicos a cada una. El proceso físico básico es llamado *Transportation*, el cual permite que la partícula pueda moverse por el medio. *Transportation* se aplica a todas las partículas por lo que ya viene definido. El resto de procesos se declaran asociados a una partícula en específico.

**G4VUserPrimaryGeneratorAction:** Esta biblioteca se utiliza para definir las partículas que atravesarán el detector. Se debe especificar el tipo de partículas, la energía, el origen del lanzamiento y el número de eventos que serán mandados. La última orden ha ejecutar define la acción de lanzamiento (*beamOn*).

Ya definidas las clases mencionadas, se crea el archivo principal **main.cc**. GEANT4 dispone de un controlador llamado *runManager* que se encarga de realizar la simulación completa. **archivo main.cc** consiste en declarar el controlador *runManager* e informarle de las clases que se han construido.

Cuando se ejecuta la aplicación, se muestra por la línea de órdenes los resultados obtenidos en modo texto y el programa queda esperando la interacción por parte del usuario. Dichas ordenes permiten realizar acciones como visualizar la simulación de forma gráfica, repetir lanzamientos, cambiar las partículas o la energía con que fueron lanzadas, eliminar procesos físicos, etc.

Toda la información obtenida a partir de las simulaciones hechas, se almacena en archivos tipo ROOT [19] a través de la clase *G4AnalysisManager*. Por lo cual se tiene que tener

un conocimiento previo del framework ROOT [19] y poder realizar el análisis de datos.

## Simulación de un detector de centelleo en GEANT4

Se diseñó y simuló en el programa de interacción radiación-materia GEANT4[1] un detector de barras plástico centelleador de poliviniltolueno  $C_9H_{10}$ , material predefinido en GEANT4. En **Fig. 3.1** se puede observar una vista frontal del detector de centelleo.

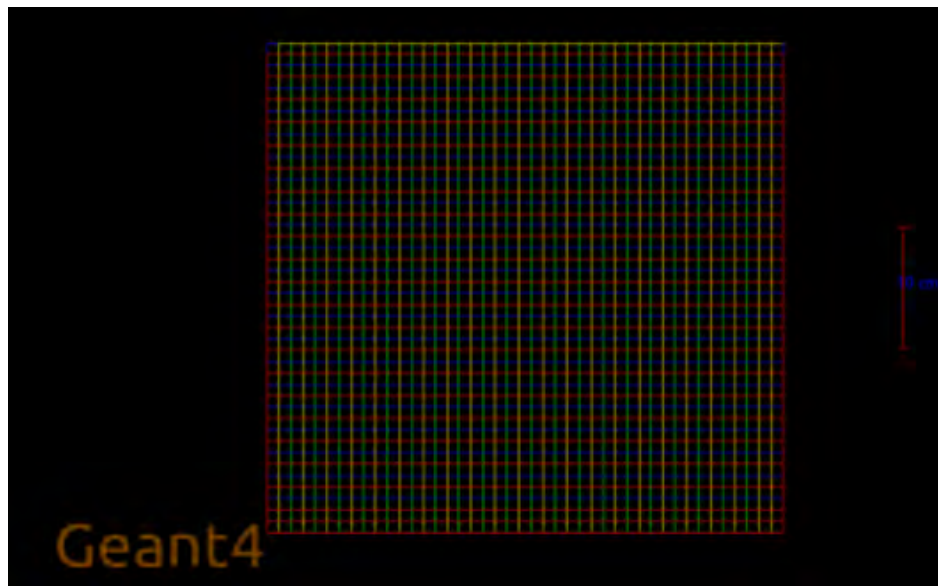


Figura 3.1: Simulación Detector de centelleo

La forma escogida de las barras se basa en el experimento MINERvA[20] que nos permite observar como los muones interaccionan al pasar el detector, los procesos físicos involucrados, la energía que depositarían en él y como sus trayectorias se verían afectadas.

En **Fig. 3.2** se observa la interacción del detector con un flujo muónico rectangular de 31 GeV, que abarca toda el área del detector, esta energía fue escogida en base a las energías medias de los muones obtenidos en los histogramas de energías de la simulación de CORSIKA para un protón de 5 TeV.

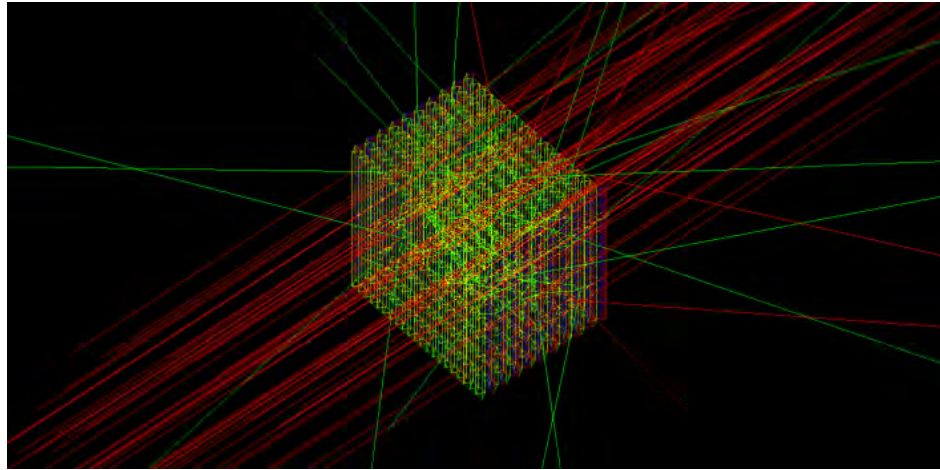


Figura 3.2: Simulación de interacciones de muones con un detector de centelleo

La simulación del detector de partículas segmentado para muones atmosféricos fue un trabajo realizado por el Físico Jairo Rodríguez de la Universidad de Nariño

## Simulación del Volcán Galéras en GEANT4

Para implementar la geometría del detector al software GEANT4 se necesitó de los siguientes programas:

**SOLIDWORKS [21]:** Es un software de modelado. En este programa se introdujeron datos de las coordenadas de las curvas de nivel, las cuales fueron proporcionadas por el SGC [3], y poder construir la geometría del volcán Galeras, exportada a un formato tipo CAD.

**FASTRAD [22]:** Gracias a este software, se transforma el archivo tipo CAD de la geometría del volcán a un formato GDML (Geometry Description Markup Language) [23] el cual puede ser leído por GEANT4 [1].

Finalmente, se usa la clase `textitG4GDMLParser` para obtener la estructura volcánica en GEANT4 [1] como se indica en la figura **Fig. 3.3**

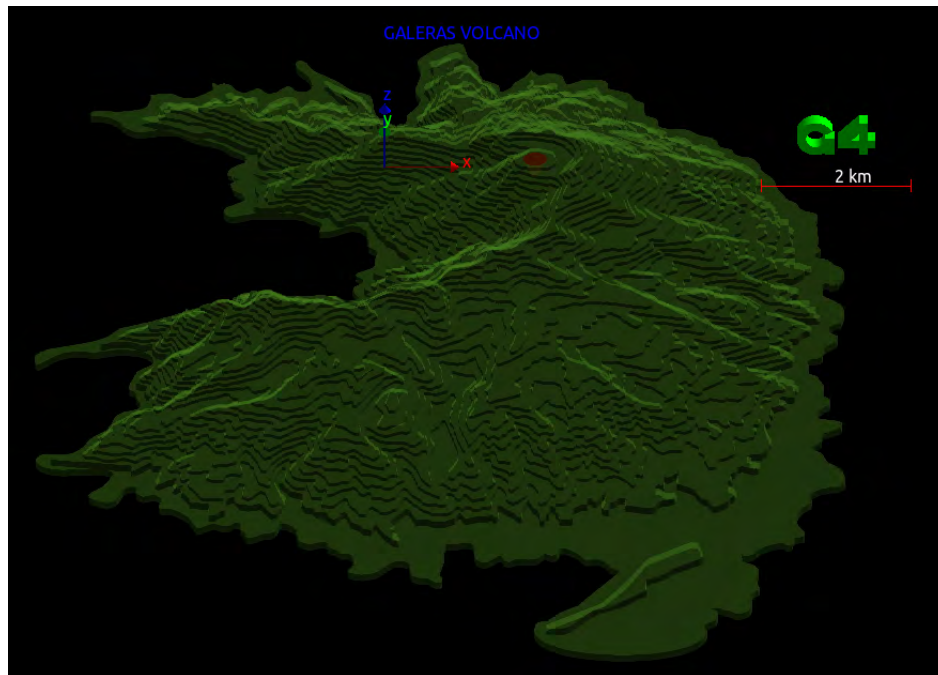


Figura 3.3: Simulación del volcán Galeras en GEANT4

La simulación de la geometría del VG está compuesta por roca estándar ( $\rho = 2,65\text{gcm}^{-3}$ ) en todas sus regiones (incluyendo el cráter) como se muestra en las siguientes tablas:

Características del Volcán	
Diametro del cráter	320m
Altura	1100m
Profundidad cráter	320m

**Tabla 1:** Características del volcán

Composición	Porcentaje en la tierra
O	46.6
Si	27.7
Al	8.3
Fe,Ca,Na,K,Mg	17.4

**Tabla 2:** Composición química de la roca estándar

La implementación de Volcán Galeras a GEANT4 fue un trabajo realizado por el Físico David Dueñas de la Universidad de Nariño.

# Capítulo 4

## Calibración del detector de centelleo

### Calibración

La calibración se define como: “el conjunto de operaciones que establecen, en unas condiciones especificadas, la relación que existe entre los valores indicados por un instrumento o sistema de medida, y los correspondientes valores conocidos de una magnitud de medida” [24].

Durante un proceso de calibración se contrasta el valor de salida (output) del instrumento a ser calibrado frente a ciertos valores estándar que se encuentren en el rango de calibración. Si el error de calibración es menor al error esperado, la calibración será aceptada. De otro modo, se requiere un ajuste del instrumento con un posterior contraste. Dado el caso de no satisfacer las tolerancias marcadas, los aparatos( para este proyecto el detector de centelleo), deben ser sustituidos por otros previamente calibrados.

Algunos motivos por los cuales es necesario realizar una calibración de un instrumento son:

- El instrumento es nuevo
- Se ha agotado un periodo de tiempo específico
- Si el instrumento ha recibido un golpe o vibraciones fuertes que pudieron descalibrarlo
- Cambios de temperatura u otra variable que presente valores superiores a los aceptables
- Cuando las observaciones obtenidas son cuestionables



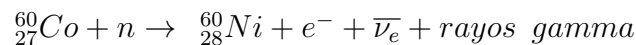
Es importante calibrar un equipo debido a que con el tiempo se vuelve menos exacto en sus mediciones; debido a factores externos. obteniendo resultado inesperados, o no notar las inexactitudes en las mediciones. Sin embargo, estas mediciones tendrán un impacto significativo a la hora de realizar el análisis de datos, lo que producirá conclusiones erróneas.

Para la calibración del detector de centelleo se usa una fuente radioactiva de Cobalto 60 ( $Co_{60}$ ), debido a que en el proceso de desintegración este isótopo radiactivo decae a Níquel 60 ( $Ni_{60}$ ); el cual emite dos rayos gamma con energías de 1,17 y 1,33 MeV respectivamente.

## Cobalto 60 ( $Co_{60}$ )

El cobalto es un elemento natural que se encuentra en la corteza terrestre, que a través de procesos químicos se convierte en un isótopo radiactivo. El  $Co_{60}$  es el isótopo más importante del cobalto y se produce al bombardear el cobalto natural,  $Co_{59}$ , con neutrones en un reactor nuclear. [25]

Aquellos átomos de  $Co_{59}$  que absorben uno de los neutrones se transforman en  $Co_{60}$ , el cual es un isótopo radiactivo, con una vida-media de 5.27 años. Este isótopo radiactivo decae emitiendo un rayo beta, o electrón, y luego el núcleo de  $Ni_{60}$  emite dos rayos gamma con energías de 1,17 y 1,33 MeV, respectivamente. El decaimiento es el siguiente:



Un esquema de la producción de  $Co_{60}$  y su desintegración se muestra en la Fig. 4.1:

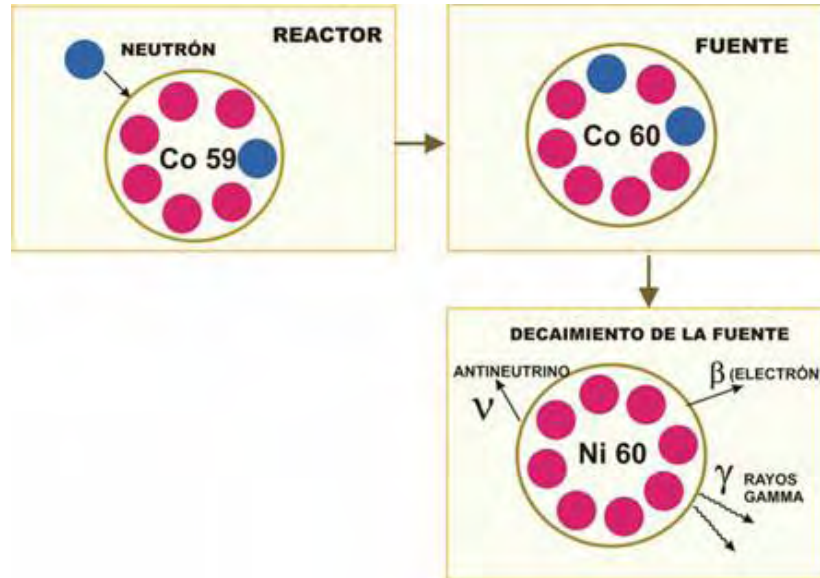


Figura 4.1: Un neutrón impacta sobre un núcleo de  $Co_{59}$  el cual se convierte en  $Co_{60}$  (inestable) que se desintegrará emitiendo  $\beta, \bar{\nu}_e, \gamma$  [26]

Las aplicaciones del  $Co_{60}$  son muchas, por ejemplo al ser aplicado a determinadas partes del cuerpo elimina las células que más rápidamente crecen, como en el caso de las cancerígenas. Sirve además para esterilizar equipos médicos, para radiografías industriales y para irradiar alimentos, entre otros usos.

En este caso, se utilizará para la calibración del equipo. El buen funcionamiento del detector de centelleo será garantizado por la detección de los picos característicos de energía de la radiación gamma en los valores mencionados (1,17 y 1,33 MeV).

## Simulación de la calibración del detector de centelleo

Para lograr la detección de los decaimientos de la fuente radioactiva de  $Co_{60}$  en la simulación del detector, fue necesario implementar nuevas clases al código del detector y modificar otras.

Inicialmente, se modificó la clase *G4VuserPrimaryGeneratorAction*. Dentro de esta clase, encontramos *G4ParticleGun* que es un generador proporcionado por GEANT4; Esta clase genera partículas primarias con un momento y posición dada sin ningún tipo de aleatorización. Para muchas aplicaciones *G4ParticleGun* es un generador de partícula adecuado. Sin embargo, para este proyecto que se desea generar partículas primarias en forma más sofisticada, se utiliza *G4GeneralParticleSource* (GPS).

**G4GeneralParticleSource (GPS):** Es parte del conjunto de herramientas Geant4, encargada del transporte de partículas de alta energía. En concreto, se permite las especificaciones de la distribución espectral, espacial y angular de las partículas de fuente primaria.

GPS permite al usuario controlar las siguientes características de las partículas primarias:

- Muestreo espacial: en superficies simples en 2D o 3D, tales como discos, esferas, y cajas.
- Distribución angular: unidireccional, isotrópico, o arbitraria (definida por el usuario).
- Espectro: lineal, exponencial, Gaussiano, etc.
- Múltiples fuentes: múltiples fuentes de partículas independientes se pueden utilizar en la misma simulación.

Además, *G4GeneralParticleSource* se puede configurar bien sea añadiendo la especificación de la fuente de partículas dentro de la clase *G4VuserPrimaryGeneratorAction*, o creando un archivo macro (.mac) el cual es un fichero que agrupa una serie de órdenes para simplificar el trabajo del usuario. Escribir una macro consiste en establecer secuencialmente las órdenes de inicialización de simulación seguidas por las que definen las acciones. Estos archivos pueden ser ejecutados desde la línea de comandos directamente. A continuación en la fig. 4.2 se muestra la simulación de la geometría esférica de una fuente de  $Co_{60}$ .

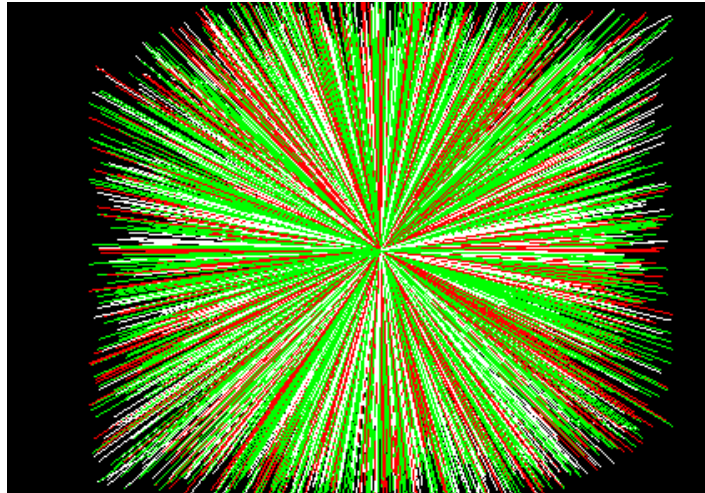


Figura 4.2: Geometría esférica de una fuente de  $Co_{60}$

Posteriormente se añadieron tres nuevas clases al código del detector, basados del ejemplo de decaimiento radioactivo proporcionado en GEANT4 con el fin de definir los decaimientos de la fuente de  $Co_{60}$ . (*geant4/examples/extended/radioactivedecay/rdecay02*).

Las nuevas clases añadidas son: *PhysicsList*, *EMStandarPhysics*, *GammaPhysics*. En *PhysicsList* se encuentra todo los procesos físicos de decaimiento, además incluye la clase *G4RadioactiveDecay* vital para simular las desintegraciones de isótopos radiactivos, así como la radiactividad inducida resultado de interacciones nucleares.

Hay una serie de archivos macros, para mostrar las características de *G4RadioactiveDecay*. La mayoría de ellos dará lugar a la creación de un archivo tipo ROOT con el mismo nombre del archivo macro en la carpeta que se haya compilado el código. Algunos ejemplos para citar son:

**th234c-b.mac:** muestra las desintegraciones de Th234 (Torio-234)

**u238c.mac:** muestra las desintegraciones de la cadena de U238 (Uranio-238, el isótopo más común del uranio.)

**as74.mac:** Las desintegraciones de As74 que tiene un esquema de desintegración bastante

complicado. (El arsénico-74 un isótopo que se utiliza como una forma de localizar tumores en el cuerpo.)

Para este proyecto se utiliza un macro denominado *Co60.mac* que se muestra a continuación:

```
1/gps/particle ion
2/gps/ion 27 60
3/gps/direction 0 0 1
4/gps/position 0 0 -40 cm
5/gps/pos/type Surface
6/gps/pos/shape Sphere
7/gps/pos/radius 8 mm
8/gps/ang/type cos
9/gps/ang/mintheta 91 deg
10/gps/ang/maxtheta 180 deg
11/run/beamOn 2000
```

En las líneas 1-2 se especifica que se utiliza un isótopo radioactivo de  $Co_{60}$ , donde 27 hace referencia al número de protones y 60 al número másico.

En 3-4 se describe la dirección y posición en la cual estará la fuente radioactiva.

A partir de 5 a 10, se especifica la geometría que tendrá la fuente de  $Co_{60}$ , en este caso se observa que es una esfera con un radio de 8 mm.

En la línea 11, se determina el número de eventos que se ejecutaran en la simulación.

Una vez implementado “GPS” y definidos los decaimientos radioactivos, finalmente se simula la interacción de una fuente radioactiva de  $Co_{60}$  con el detector de centelleo, como se muestra en la fig. 4.3:

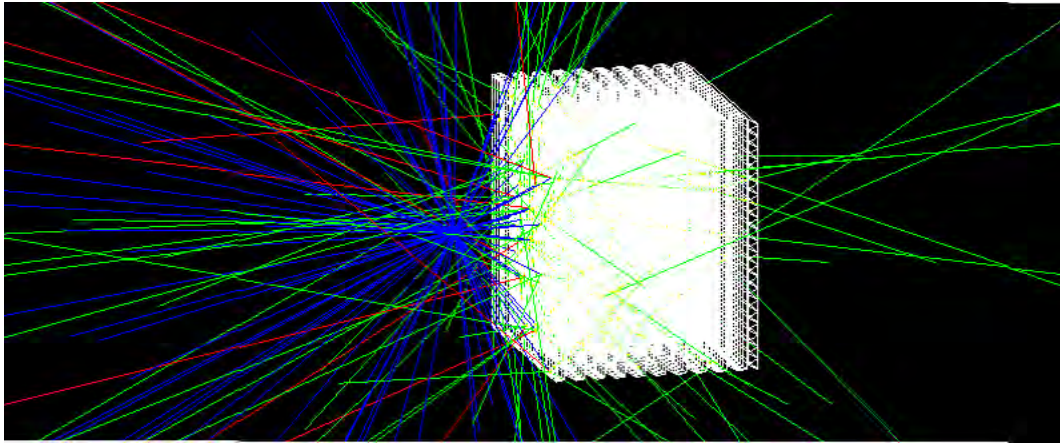


Figura 4.3: Interacción de una fuente de  $Co_{60}$  con el detector de centelleo

El color de la trayectoria dibujada depende de la carga de la partícula.

negativa: rojo

neutra: verde

positiva: azul

Los resultados obtenidos en formato tipo ROOT de este capítulo se encuentran en la sección 6 de este trabajo.

## Capítulo 5

# Ubicación del detector de centelleo sobre el volcán Galeras

El aspecto mas relevante a la hora de determinar la ubicación del detector de centello es conocer el perfil del complejo volcánico, y la energía de arribo del flujo de muones que atraviesa el cuerpo geológico; para este caso a la altura de la ciudad de Pasto con base en los resultados obtenidos en CORSIKA [2] [14].

En el caso del volcán Galeras, con la ayuda del Servicio Geológico Colombiano(SGC) [3] quienes realizaron un estudio de los posibles puntos para la ubicación del detector, obtuvieron algunas alternativas para el perfil de elevación.

En la figura 5.3 que se muestra a continuación se observa el volcán Galeras, a 9 km de este la ciudad de San Juan de Pasto y otras pobaciones a su alrededor. Las distancias existentes desde el cráter volcánico y las poblaciones se especifica bajo el nombre de cada lugar en unidades de kilometros. A la derecha se muestra el mapa de Colombia y donde esta localizado el volcán de estudio.

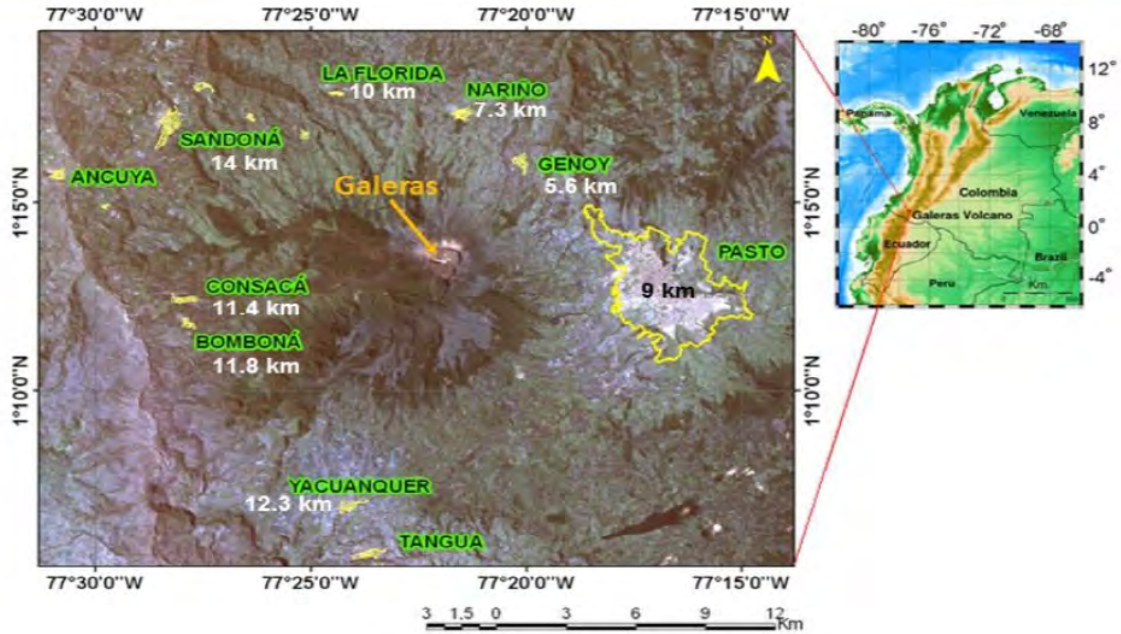


Figura 5.1: Ubicación del volcán Galeras y poblaciones a su alrededor [3]

## Geometrías en una sola simulación: Detector-Volcán

Debido a que la simulación del volcán Galeras se encuentra en formato GDML (input de GEANT4)[23], mientras que el detector se diseñó y simuló en lenguaje de programación C++ con el software de interacción-materia GEANT4[1] se agregó una nueva clase llamada *G4GDMLParser*.

Los archivos GDML se pueden importar directamente en GEANT4, utilizando el plug-in GDML. En general, se escribe las siguientes líneas en la clase *DetectorConstruction* del detector de centelleo:

### 1. Incluir los archivos header necesarios.

```
# include "G4GDMLParser.hh"
```



## 2. Declaración e inicialización

```
G4GDMLParser parser;
```

“Ejemplo.gdml” será el archivo fuente GDML, en caso de múltiples archivos GDML se debe poner el archivo principal de GDML en el siguiente paso.

## 3. Ejecución del archivo GDML

```
parser.Read(“Ejemplo.gdml”);
```

Si se encuentran sólidos u otras entidades con el mismo nombre, se genera un mensaje de precaución. Normalmente esto ocurre cuando los diferentes módulos GDML definen el mismo volumen lógico. Por lo que es recomendable revisar la forma en que se han organizado las geometrías.

## 4. Obtener el volumen mundo

```
G4VphysicalVolume* W = parser.GetWorldVolume(); //world volume  
W->GetLogicalVolume()->SetVisAttributes(G4VisAttributes::Invisible);
```

La segunda línea es opcional, pero de uso común, porque generalmente no se quiere ver el volumen mundial.

Debido, a que se debe implementar el detector de centelleo con sus características en la misma simulación con el volcán; simplemente en el código del detector se define el volumen madre como el volumen lógico del archivo GDML. Así:

```
W->GetLogicalVolume() //its mother volume
```

Obteniendo en la interfaz gráfica lo siguiente:

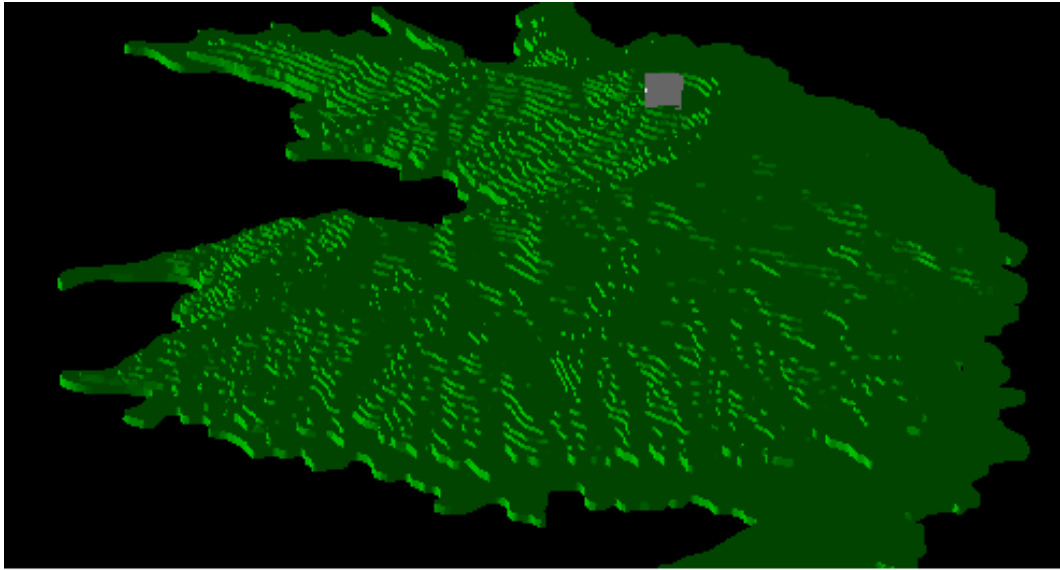


Figura 5.2: Primera simulación del detector de centelleo con el volcán Galeras

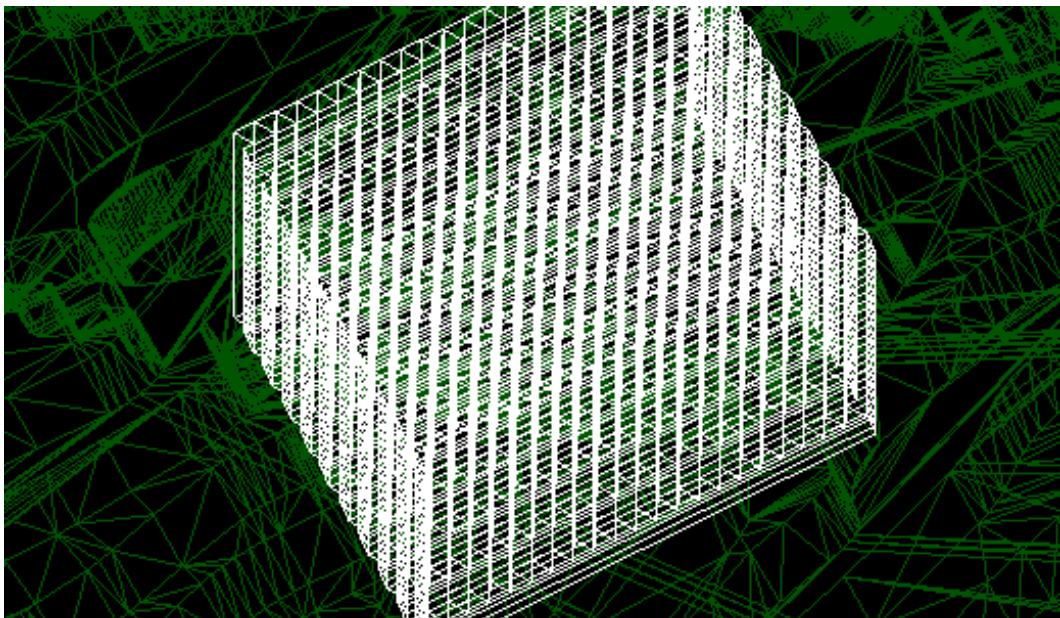


Figura 5.3: Acercamiento al detector con el volcán Galeras

Se observa que las dimensiones del detector no corresponden con el tamaño real, por lo tanto, una vez obtenidas las dos geometrías en una sola simulación, se procede a redimensionar y a ubicar el detector de centelleo con movimientos de translación y rotación sobre un punto óptimo sobre el cuerpo geológico; para poder realizar la simulación de una muongrafía al cráter volcánico.

## **Puntos óptimos para la ubicación del detector de centelleo**

La ubicación del detector es un aspecto de gran importancia a la hora de aplicar muongrafía sobre cuerpos geológicos, puesto que la distancia que recorren los muones a través de la estructura, depende de la topografía del lugar y la localización de la zona de interés.

Los sitios más convenientes para el posicionamiento del detector de centelleo se seleccionaron a partir de los puntos estudiados por el SGC [3] teniendo en cuenta la cercanía de estos a la vía (que no sea una zona de difícil acceso), la distancia límite de penetración para los muones en roca estándar, factores como la altitud y distancia entre cada uno de los puntos con el cráter volcánico. En el siguiente mapa **Fig. 5.4** se muestran las coordenadas geográficas de los puntos más óptimos para la ubicación del detector.

El punto seleccionado para la ubicación del detector en la simulación es el punto 3 con coordenadas:  $1,216328^{\circ}$  N,  $-77,358964^{\circ}$  O, 3908 m. (**Fig. 5.5**)

Se puede realizar la simulación tanto en el punto 1 y punto 2 debido a que el proceso es exactamente el mismo que para el punto 3.

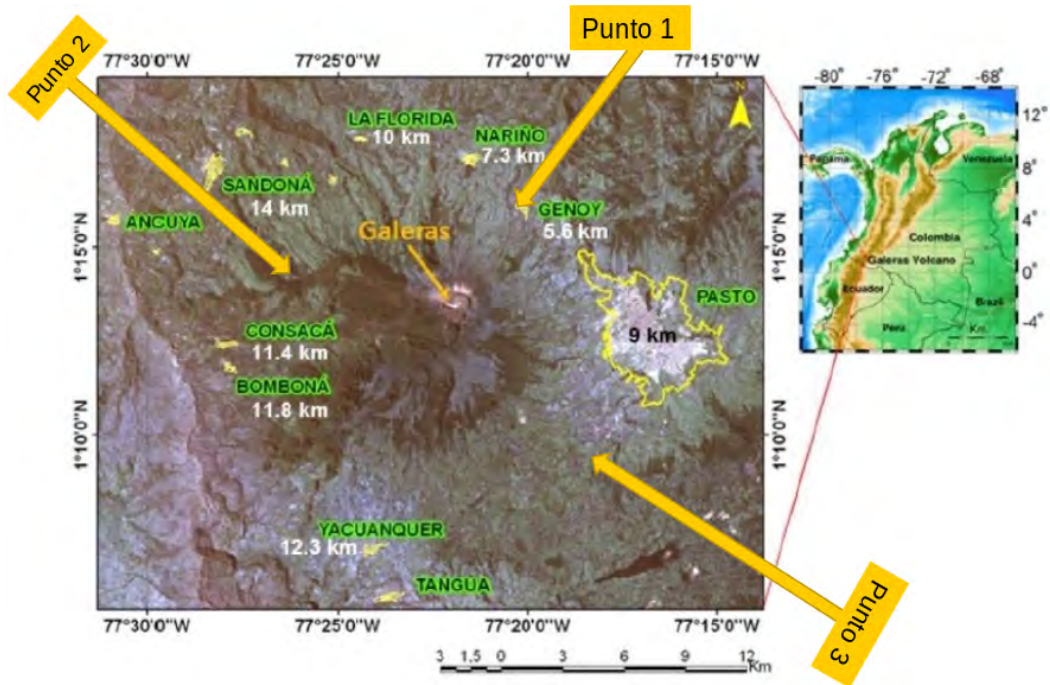


Figura 5.4: Puntos más convenientes para la disposición del detector de muones[3]



Figura 5.5: Punto de ubicación sobre el cual se realiza la simulación[3]

## Simulación del detector en un punto óptimo sobre el VG

Una vez obtenida la geometría del volcán Galeras en formato GDML, a partir de los datos de las coordenadas de las curvas de nivel proporcionadas por el SGC [3], se abrió el archivo con Gedit, el cual es un editor de texto, y se modificó las coordenadas cartesianas originales; es decir, las posiciones iniciales de tal forma que el volcán se trasladó al origen del sistema de coordenadas de GEANT4 [1]. En la **Fig. 5.6** se observa en recuadros las coordenadas originales de los elementos que conforman el complejo volcánico:

```

<volume name="Structure0x_710081264">
  <materialref ref="VACUUM"/>
  <solidref ref="__World_SphereRef"/>
  <physvol name="Element_Step_0">
    <volumeref ref="Element_Step_0_vol"/>
    <position name="posRef_1" x="237555.73640625936" y="135113.32545183244" z="3975"/>
    <rotationref ref="Identity"/>
  </physvol>
  <physvol name="Element_Step_1">
    <volumeref ref="Element_Step_1_vol"/>
    <position name="posRef_2" x="236292.18110626991" y="133517.7215714647" z="3925.000000000005"/>
    <rotationref ref="Identity"/>
  </physvol>
  <physvol name="Element_Step_2">
    <volumeref ref="Element_Step_2_vol"/>
    <position name="posRef_3" x="236225.26645955496" y="133154.85955957283" z="3475.000000000005"/>
    <rotationref ref="Identity"/>
  </physvol>
  <physvol name="Element_Step_3">
    <volumeref ref="Element_Step_3_vol"/>
    <position name="posRef_4" x="236292.60181168615" y="131945.74147118349" z="3150.000000000005"/>
    <rotationref ref="Identity"/>
  </physvol>
</volume>
</structure>

```

Figura 5.6: coordenadas cartesianas originales del VG

En la **Fig. 5.7** se muestra las nuevas coordenadas para cada elemento, ubicando la estructura geológica en el origen:

```

<volume name="Structure0x_710081264">
  <materialref ref="VACUUM"/>
  <solidref ref="World_SphereRef"/>
  <physvol name="Element_Step_0">
    <volumeref ref="Element_Step_0_vol"/>
    <position name="posRef_1" unit="mm" x="0" y="0.0" z="0.0"/>
    <rotationref ref="identity"/>
  </physvol>
  <physvol name="Element_Step_1">
    <volumeref ref="Element_Step_1_vol"/>
    <position name="posRef_2" unit="mm" x="-1263.555299989" y="-1595.603880368" z="-50.0000000000005"/>
    <rotationref ref="identity"/>
  </physvol>
  <physvol name="Element_Step_2">
    <volumeref ref="Element_Step_2_vol"/>
    <position name="posRef_3" unit="mm" x="-1330.469946704" y="-1958.46589226" z="-500.0000000000005"/>
    <rotationref ref="identity"/>
  </physvol>
  <physvol name="Element_Step_3">
    <volumeref ref="Element_Step_3_vol"/>
    <position name="posRef_4" unit="mm" x="-1263.134594573" y="-3167.583980649" z="-825.0000000000005"/>
    <rotationref ref="identity"/>
  </physvol>
</volume>
</structure>

```

Figura 5.7: translación del VG al origen del sistema de coordenadas de GEANT4 [1]

Una vez posicionado el VG, se debe redimensionar y ubicar el detector de muones en nuestro punto de interés. Para cambiar el tamaño, en la clase *DetectorConstruction* basta con modificar las unidades de los parámetros de geometría que conforman el detector y adecuarlo al tamaño requerido.

Para la ubicación del detector, como primer paso, se transformó las coordenadas geográficas del punto 3 a cartesianas para poder ser ubicado en el sistema de referencia de GEANT4 [1]. Este proceso se realizó con la ayuda de un conversor de coordenadas [27] donde el modelo de elipsoide utilizado es el WGS84. En la **Fig. 5.8** se indica la conversión de coordenadas geográficas del punto 3 (  $1,216328^\circ$  N,  $-77,358964^\circ$  O,  $3908$  m) a coordenadas cartesianas (x,y).

**Convertor de coordenadas**

Las coordenadas geográficas deben ser ingresadas y serán mostradas en grados decimales. Los números negativos indican longitudes de Oeste y latitudes de Sur. Las coordenadas UTM deben ser ingresadas y serán mostradas en metros. El modelo de elipsoide utilizado es el WGS84.

Geográficas		UTM	
Longitud:	-77.358964	X:	237490.5118740326
Latitud:	1.216328	Y:	134555.75416846646
		Zona:	18
		Hemisferio:	<input checked="" type="radio"/> N <input type="radio"/> S

Figura 5.8: Conversión de coordenadas del punto 3 [27]

De esta manera, para el punto 3, las coordenadas cartesianas de ubicación son:

$x = 237490.512 \text{ m}$ ,  $y = 134555.754 \text{ m}$ ,  $z = 3908 \text{ m}$ .

Como segundo paso, se trasladó las coordenadas cartesianas encontradas anteriormente, sobre la superficie del complejo volcánico, el cual se encuentra ubicado en el origen del sistema de coordenadas de GEANT4 [1]; se llevó a cabo efectuando la diferencia

$(X - X_o, Y - Y_o, Z - Z_o)$ . Donde  $(X, Y, Z)$  corresponden a las coordenadas cartesianas de ubicación para el detector y  $(X_o, Y_o, Z_o)$  son las coordenadas originales del primer elemento (*Element\_step\_0*) que conforma la estructura del VG (237555.73640625936m, 135113.32545183244m, 3975m), obteniendo el siguiente conjunto de coordenadas que corresponden al punto de ubicación óptimo sobre la simulación del complejo volcánico:

$x = -65.22440625936 \text{ m}$ ,  $y = 557.57145183244 \text{ m}$ ,  $z = -67 \text{ m}$ .

El tercer paso, corresponde a la rotación de el detector de tal forma que este apunte hacia el cono volcánico y que el ángulo de inclinación proporcione una mayor cantidad de direc-

ciones incidentes a través de la zona de interés de la estructura volcánica. Para ello, se creó una matriz de rotación con la clase *G4RotationMatrix* que inclinó al detector de centelleo  $30^\circ$  mirando hacia el cráter. A continuación, en las **Fig. 5.9** y **Fig. 5.10** se muestra la interfaz gráfica obtenida de el proceso anteriormente mencionado:

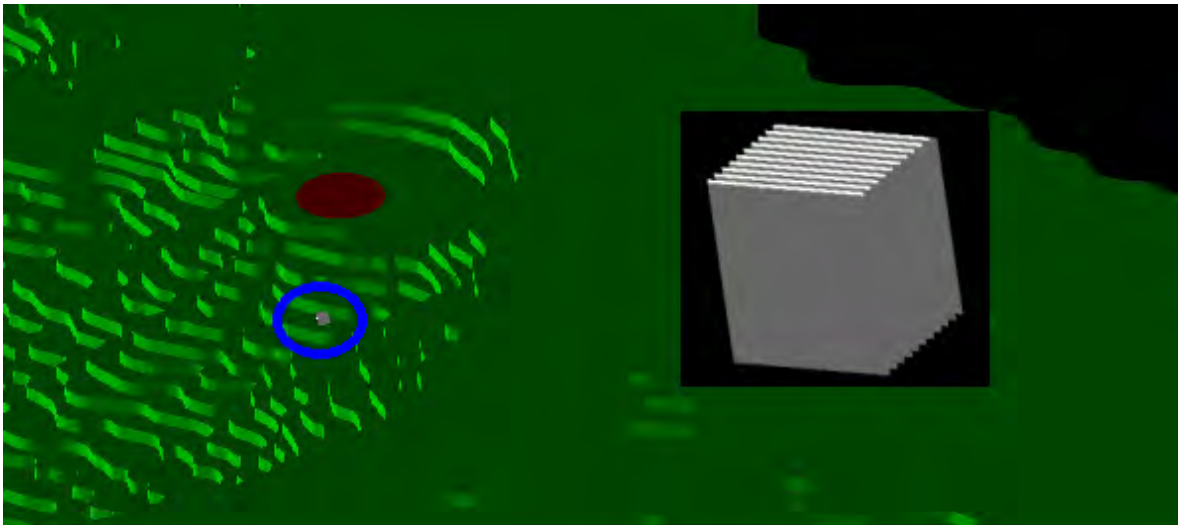


Figura 5.9: Simulación de la ubicación óptima del detector de centelleo sobre el complejo volcánico

Observe que debido a que la geometría de la estructura geológica es muy grande, para apreciar el punto de ubicación en la interfaz gráfica se ha realizado la simulación con un detector de tamaño considerable, exactamente 9.5 veces más grande que el tamaño real.



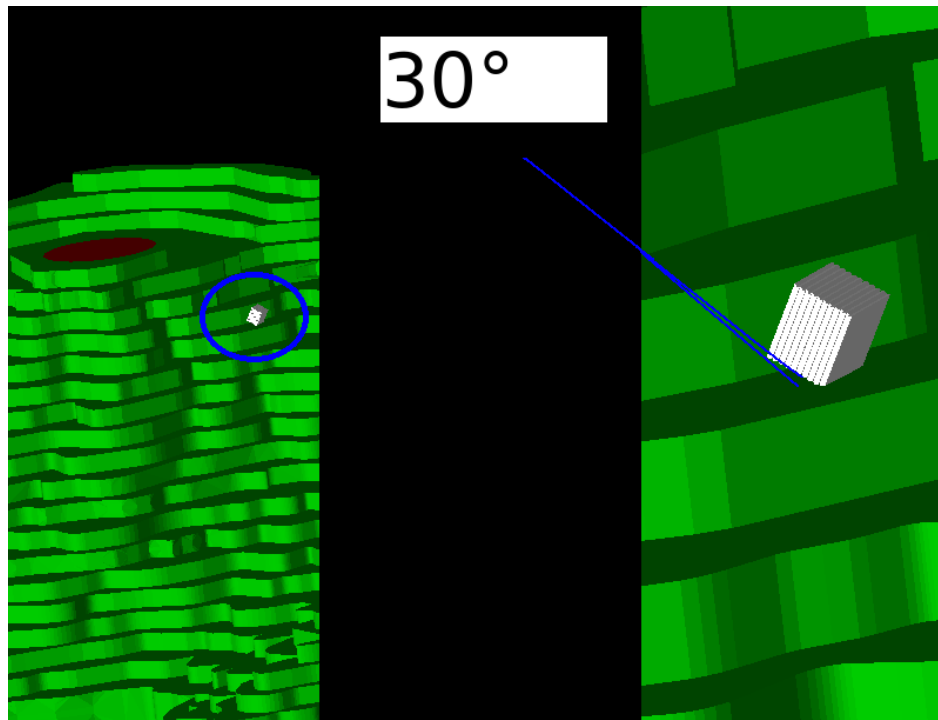


Figura 5.10: Ángulo de inclinación del detector sobre el complejo volcánico

Se prosigue a simular un flujo de muones que atraviese el cono volcánico y arribe al detector de centelleo.

## Flujo de muones que atraviesa el cono volcánico y arriba al detector

De la misma manera, en que se utilizó un archivo macro (.mac) para la simulación de una fuente radioactiva de  $C_{60}$ , para este caso se escribió un macro que simula un flujo de muones generado por encima del complejo volcánico, que atraviesa el cráter volcánico y arriba al detector. El código se muestra a continuación:

```
1/gps/particle mu-  
2/gps/energy 1 TeV  
3/gps/direction 0 0 1  
4/gps/pos/type Beam  
5/gps/pos/shape Rectangle  
6/gps/pos/rot1 0 0 1  
7/gps/pos/centre 200 6000 2000 m  
8/gps/pos/halfx 1 m  
9/gps/pos/halfy 1 m  
10/gps/direction -1 0 0  
11/gps/ang/type focused  
12/gps/ang/focuspoint -65.22440625936 -557.57145183244 -67 m  
13/run/beamOn 1
```

En las líneas 1-2 se especifica que se utiliza como partícula el muón con energía de 1 TeV.

De 3 a 10 se describe la forma, dirección y posición para el flujo de muones.

En 11-12 se ajusta el punto de enfoque, el valor predeterminado es (0, 0, 0), para nuestro caso, es el punto donde está situado el detector.

En la línea 13, se determina el número de eventos que se ejecutaran en la simulación.

Para poder observar un interfaz gráfica, se ejecutó la simulación con solo un evento, obteniendo la **Fig. 5.11**

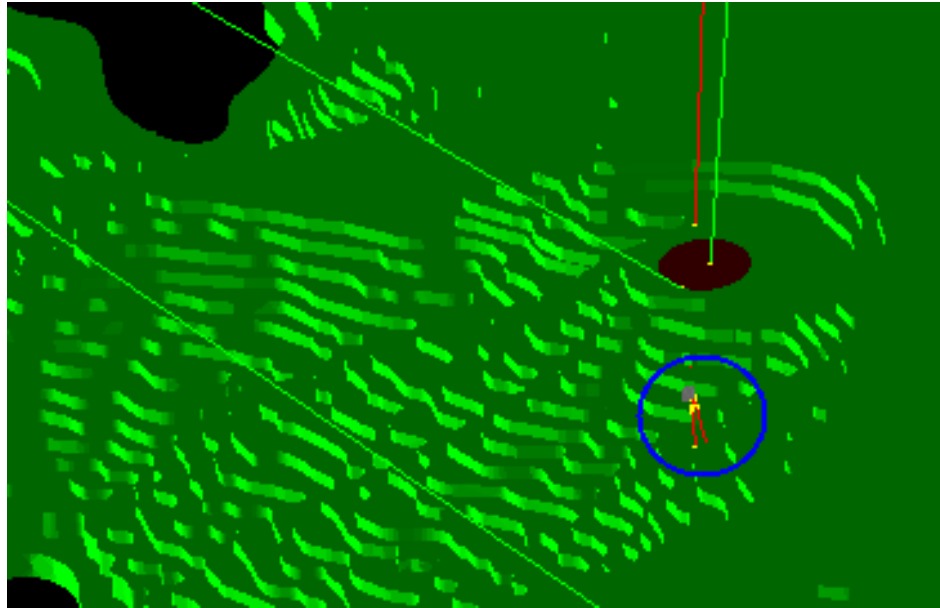


Figura 5.11: Interacción de un muón con energía de 1TeV con el volcán Galeras y el detector de centelleo.

Donde la línea roja superior corresponde a la trayectoria del muon; en color amarillo se identifican las interacciones de la partícula con la roca. Finalmente, producto de la interacción del muon con el volcán y el detector se generan partículas neutras (en color verde) y partículas negativas (en color rojo).

La **Fig. 5.12** es un acercamiento a las interacciones del muon con el cráter volcánico que arriba al detector.

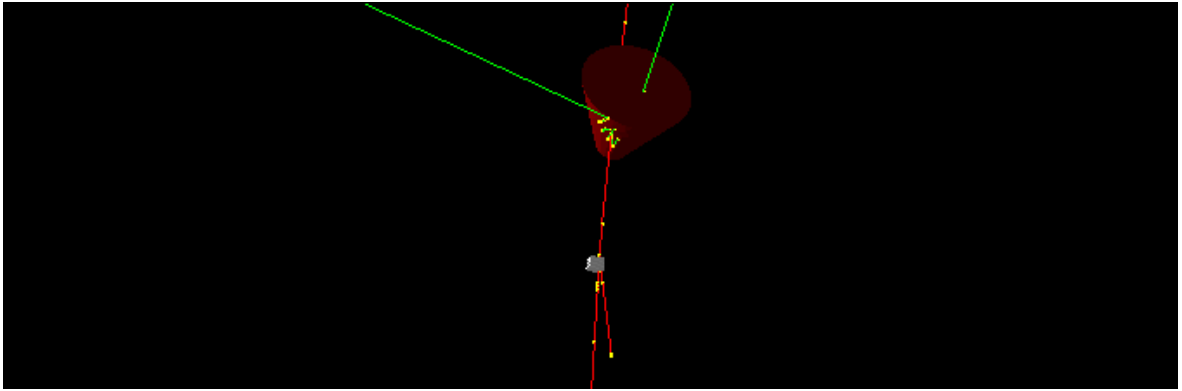


Figura 5.12: Interacción de un muón con energía de 1TeV con el cráter volcánico que arriba al detector

# Capítulo 6

## Resultados

### CORSIKA

A continuación muestran los resultados previos obtenidos con CORSIKA [2] [14].

Se simularon 200 EAS para cada ángulo y energía descrito en el capítulo 2. Con dichos datos se obtuvo el número estimado de muones que arriban a la altura de la ciudad de Pasto con los cuales se obtuvo una MLD (muon lateral distribution) y gráficas del espectro de energías.

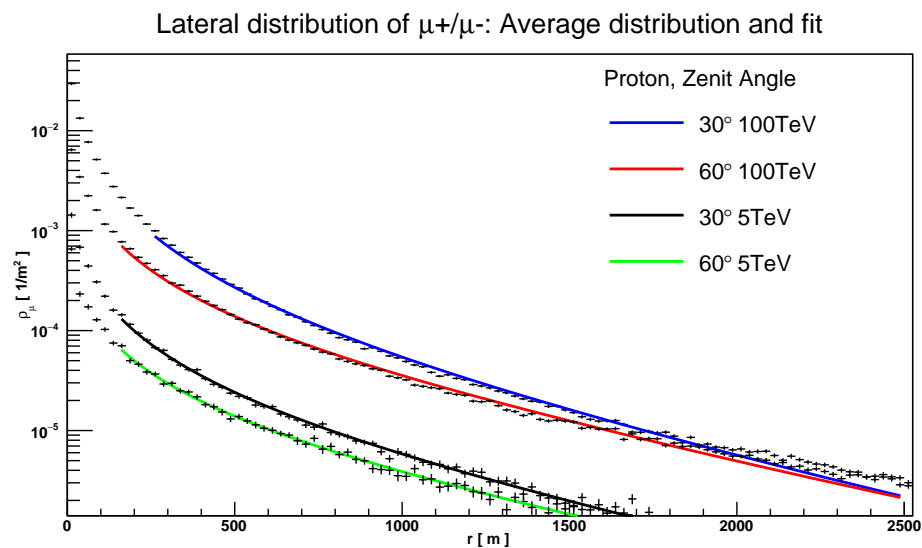


Figura 6.1: Distribución lateral de muones que arriban a la ciudad de Pasto generada por protones de 5 y 100 TeV para los ángulos  $30^\circ$  y  $60^\circ$ [14]

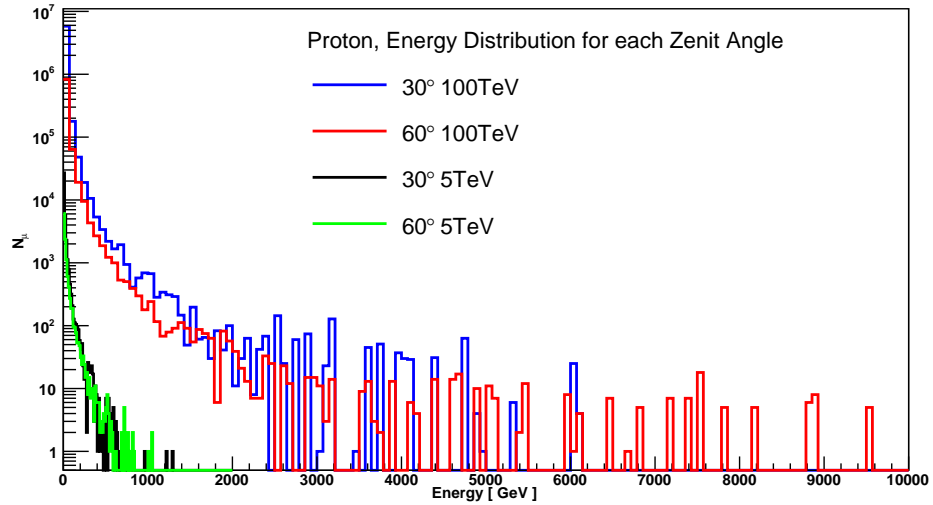


Figura 6.2: Espectro de energía de muones generada por protones de 5 y 100 TeV para los ángulos  $30^\circ$  y  $60^\circ$ . [14]

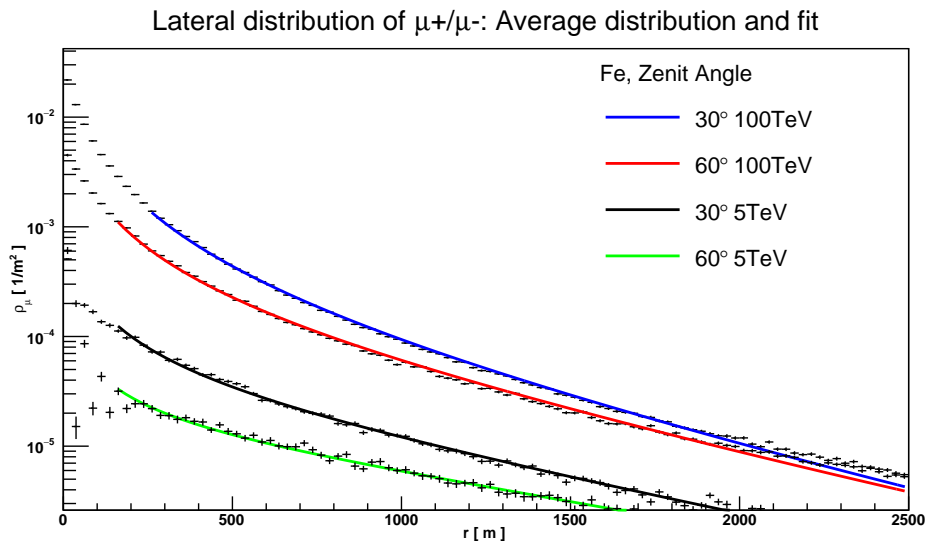


Figura 6.3: Distribución lateral de muones que arriban a la ciudad de Pasto generada por núcleos de hierro de 5 y 100 TeV para los ángulos  $30^\circ$  y  $60^\circ$ . [14]

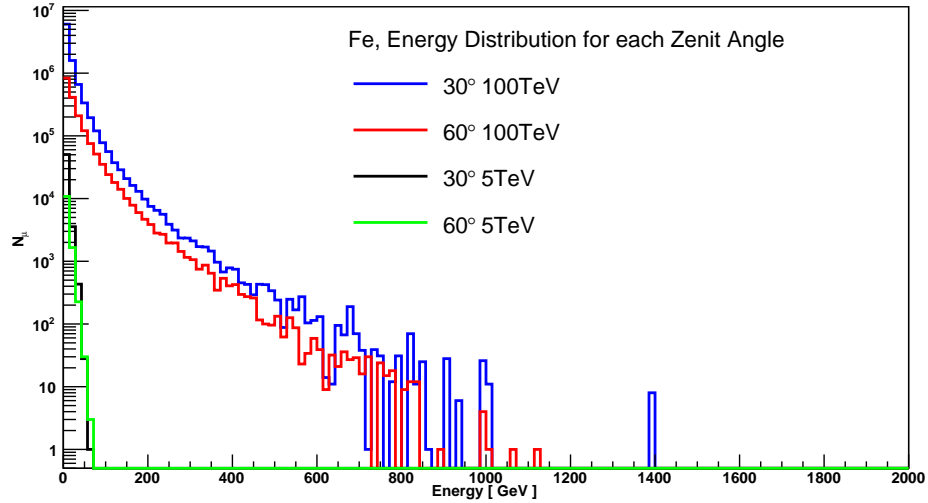


Figura 6.4: Espectro de energía de muones generada por núcleos de hierro de 5 y 100 TeV para los ángulos 30° y 60°[14]

Como se mencionó en el capítulo 2, las MLDs obtenidas de las simulaciones se comportan de acuerdo a la ecuación de Nishimura- Kamata-Greisen[15] [16] para el arreglo de los detectores de KASCADE-Grande[17]. Donde se obtuvo:

proton									
		$N_\mu$	$\beta$		$N_\mu$	$\beta$		$N_\mu$	$\beta$
5TeV	0°	0.00014	1.5	10TeV	0.00025	1.5	100TeV	0.0029	2.0
	30°	0.00014	1.4		0.00021	1.4		0.0023	1.9
	45°	0.000097	1.3		0.00015	1.3		0.0015	1.6
	60°	0.000063	1.2		0.000010	1.1		0.00074	1.3

Figura 6.5: Parametros de ajuste para la MLDF a la altura de la ciudad de Pasto con protones como partículas primarias[14]

Fe									
		$N_\mu$	$\beta$		$N_\mu$	$\beta$		$N_\mu$	$\beta$
		5TeV	0°		0.00016	0.92		10TeV	0.00041
	30°	0.00010	0.71		0.00027	1.0		0.0033	1.7
	45°	0.000060	0.50		0.00015	0.79		0.0022	1.5
	60°	0.000022	0.12		0.000060	0.38		0.0011	1.2

Figura 6.6: Parametros de ajuste para la MLDF a la altura de la ciudad de Pasto con núcleos de hierro como partículas primarias[14]

## GEANT4

### Calibración del detector de centelleo

Se simularon 2000 eventos de interacciones de la fuente de  $C_{60}$  con el detector de centelleo. En **Fig. 6.7** se puede observar los procesos físicos generados por la interacción de la fuente radioactiva con el detector de centelleo

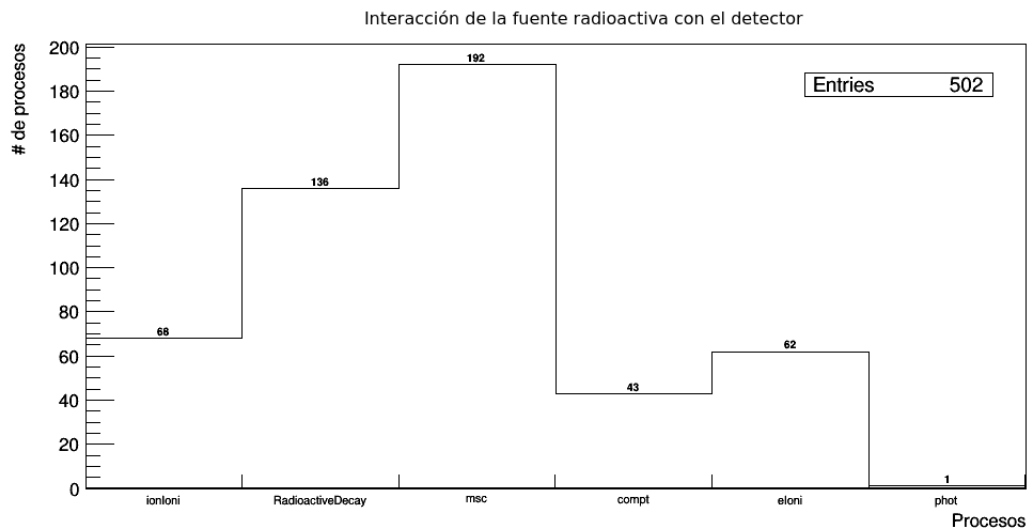


Figura 6.7: Procesos físicos generados por la interacción de la fuente de  $C_{60}$  con el detector

En **Fig. 6.8** se muestra los decaimientos de la fuente radioactiva de  $C_{60}$ . Observe la alta



emisión de radiación gamma.

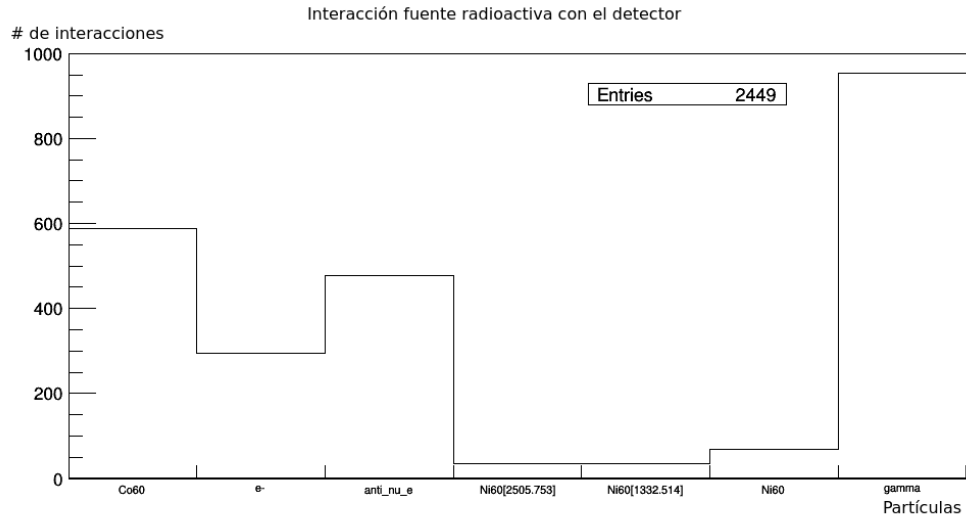


Figura 6.8: Decaimiento de la fuente de  $Co_{60}$

En **Fig. 6.9** se muestra la energía de la radiación gamma. Como se mencionó en el capítulo 4 cuando este isótopo radiactivo decae a  $Ni_{60}$ ; emite dos rayos gamma con energías de 1,17 y 1,33 MeV respectivamente. El siguiente resultado verifica lo anteriormente mencionado.

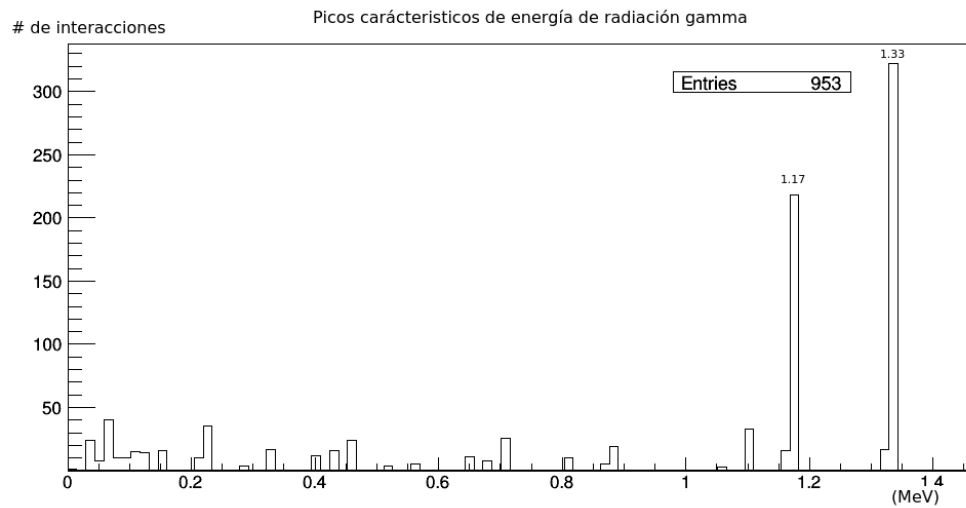


Figura 6.9: Picos característicos de radiación gamma

Los resultados anteriores además de mostrar los decaimientos de una fuente radioactiva, garantiza el buen funcionamiento del detector para la calibración del mismo.

## Interacción de muones con el cráter volcánico y el detector

Para generar un flujo de muones considerable, se escaló la simulación a unidades de milímetros (mm) y se modeló un flujo muónico de 1000 evento con energía de 1 TeV que interacciona con el cono volcánico y arriba al detector.

Se obtuvo todos los procesos correspondientes a las interacciones de las partículas al paso por el detector de centelleo. Obteniendo el histograma **6.10**

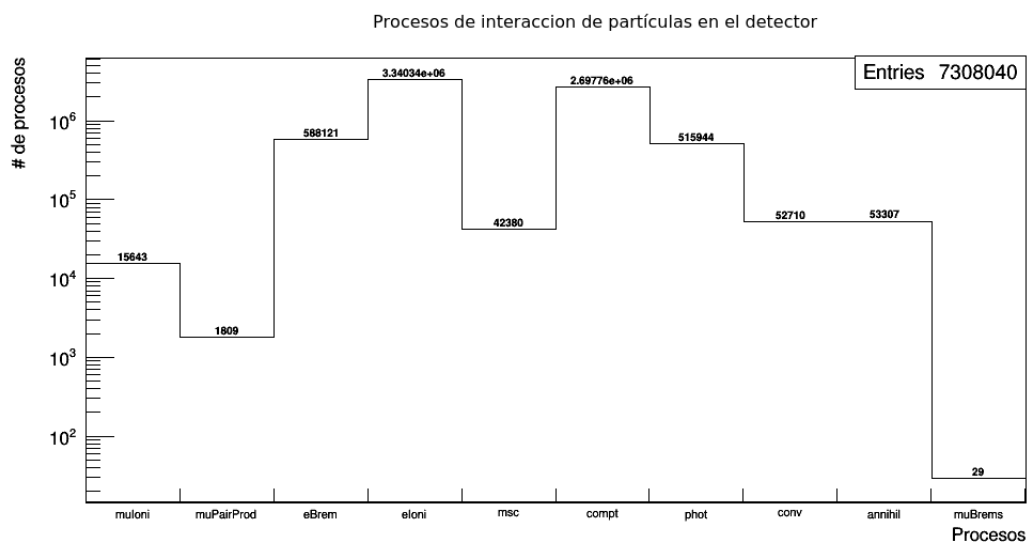


Figura 6.10: Procesos físicos producido por las interacciones de partículas con el detector centelleo.

En el anterior histograma se aprecian todos los procesos físico generados por las partículas que arriban al detector. Se observa que los procesos con mas aporte son *eIoni* o ionización generada por electrones y *compt* que hace referencia a la dispersión Compton. También se realizó el histograma **6.11** con solo los procesos físicos producido por las interacciones de los muones que logran arribar al detector centelleo.

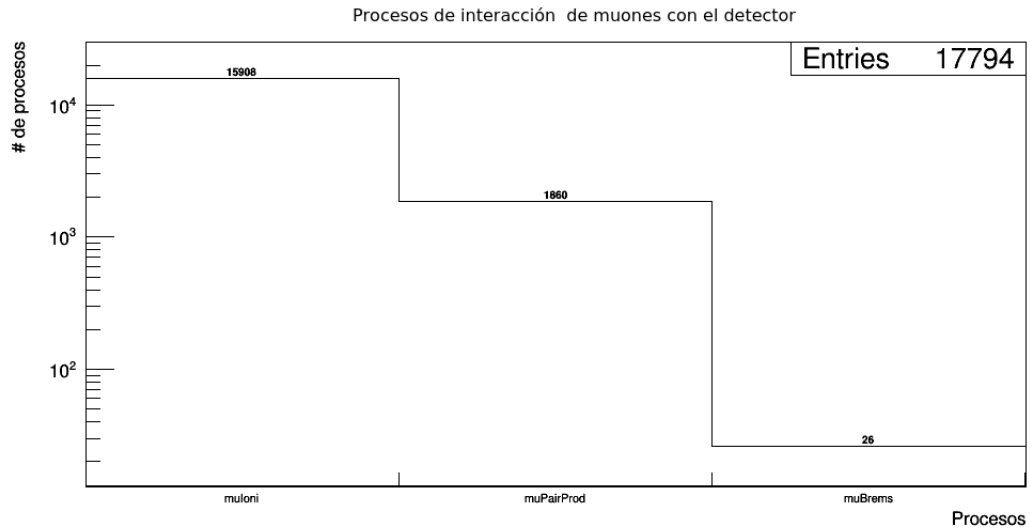


Figura 6.11: Procesos físicos producido por las interacciones de muones con el detector centelleo.

En el histograma **6.11** el proceso con mayor aporte es el *muioni* o ionización generada por muones, en segundo lugar *muPairProd* o producción de pares y finalmente *muBrems* o Breemstralung por muones.

A continuación se encuentra la información sobre la energía depositada en el detector tanto para las partículas involucradas y la de solo los muones.

En el histograma **6.12** se encuentra la energía depositada por las partículas involucradas en el detector.

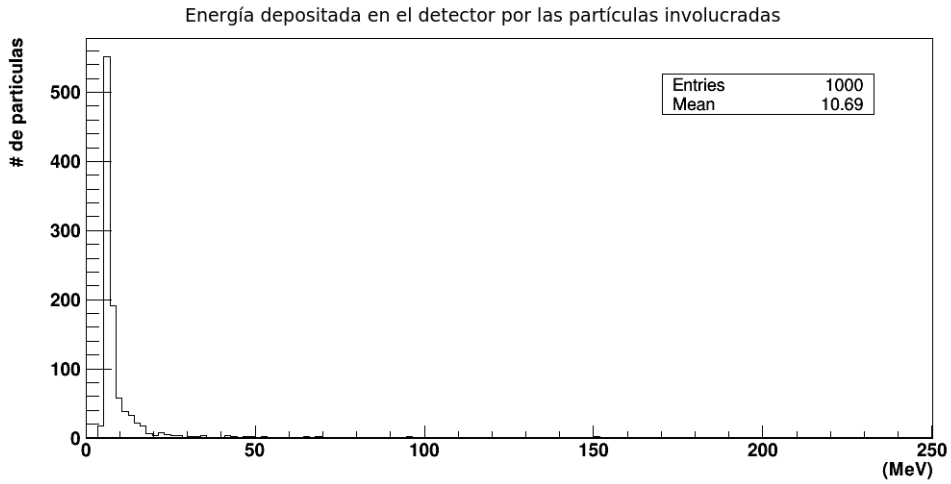


Figura 6.12: Energía depositada por las partículas en el detector

Realizando un acercamiento en el eje Y, se observa ciertas energías que no se lograban visualizar.

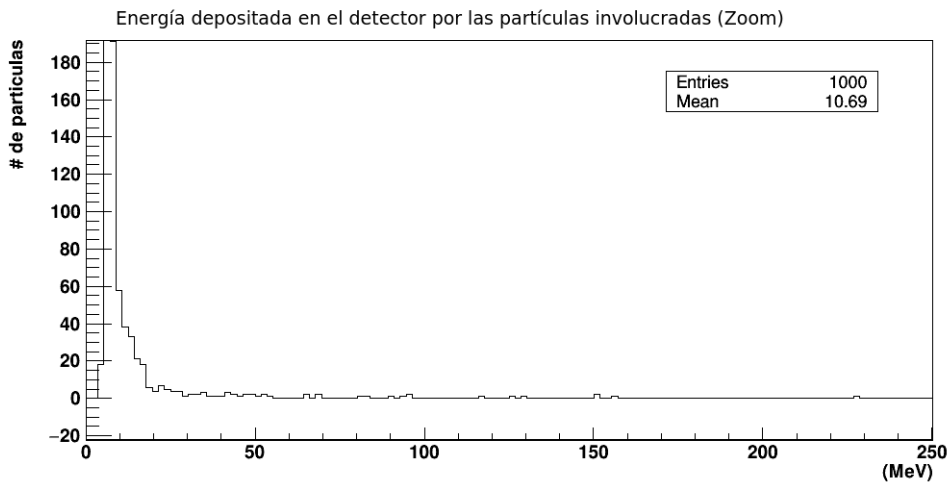


Figura 6.13: Zoom: Energía depositada por las partículas en el detector

También se obtuvo información sobre le energía depositada por los muones en el detector de centelleo, este resultado se muestra en el histograma **6.14**

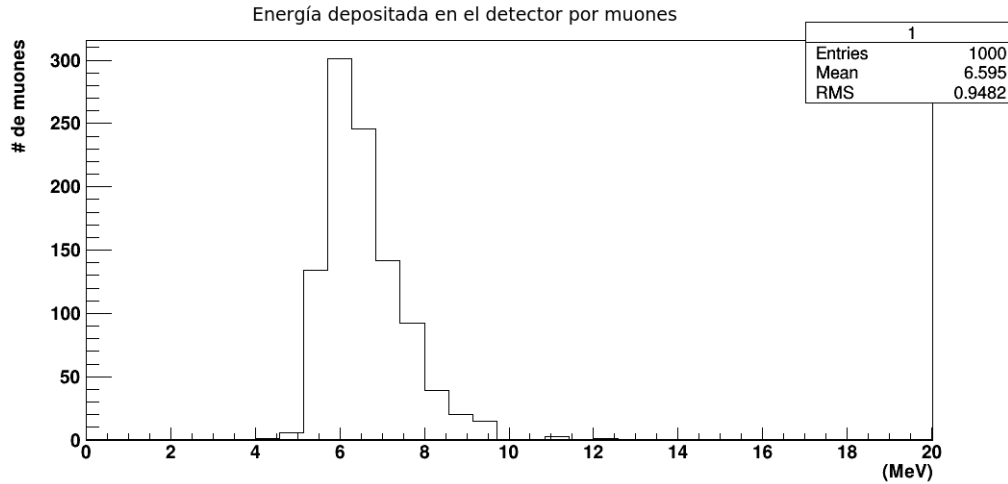


Figura 6.14: Zoom: Energía depositada por muones que arriban al detector

Del histograma anterior se deduce que la energía media depositada por los muones que logran arribar al detector de centelleo es de  $6.595 \text{ MeV} \pm 0.95 \text{ MeV}$ .

Finalmente, se realizó una proyección de los muones que logran atravesar el complejo volcánico y arribar al detector. Para ello, se implementó dos tipos de volcán; el primero compuesto por roca estándar en todas sus zonas y el segundo donde su cráter es de aire. Con el fin de obtener una mayor resolución de las proyecciones se trabajó con un flujo muónico mucho mayor, que consiste de 20000 eventos y una energía de 200 MeV.

En la **Fig. 6.15** se muestra la proyección YZ del detector donde el volcán está formado por roca estándar y el cráter por aire.

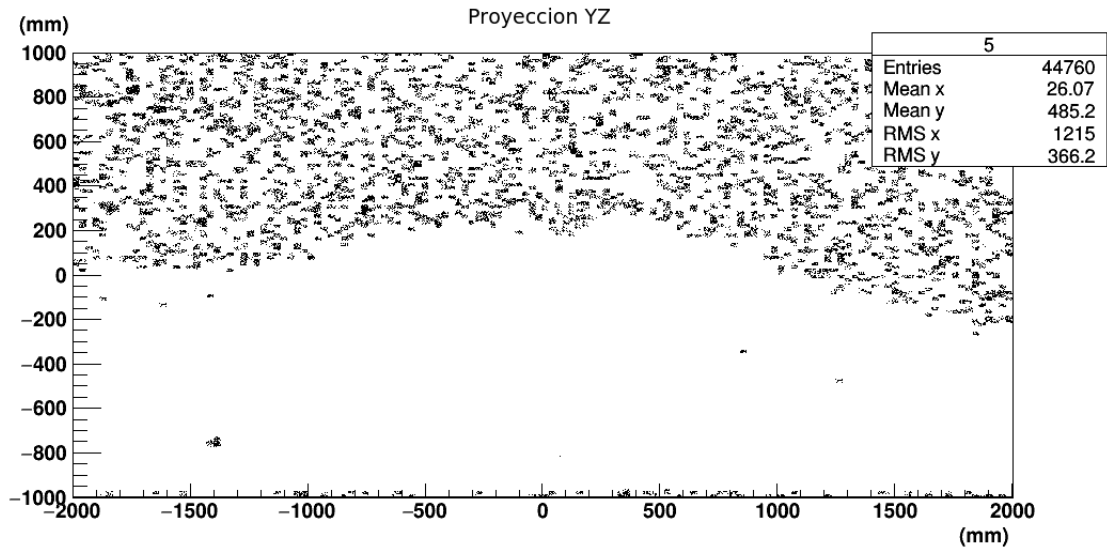


Figura 6.15: Proyección de muones que atraviesan el volcán y arriban al detector (cráter compuesto por aire)

En la 6.16 se muestra la proyección YZ del detector donde el volcán está formado por roca estándar en todas sus regiones.

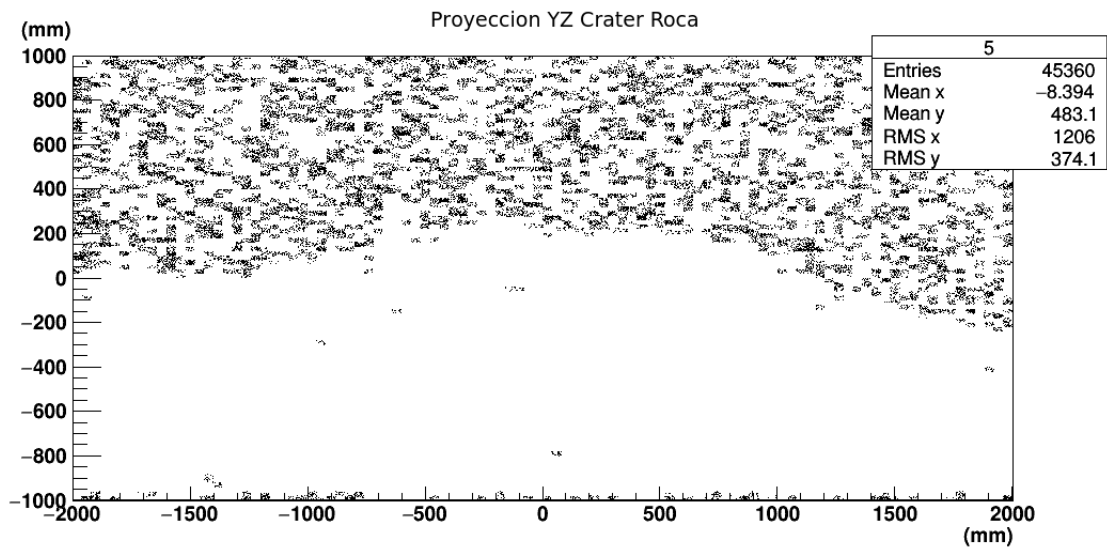


Figura 6.16: Proyección de muones que atraviesan el volcán y arriban al detector (cráter compuesto por roca)

Se concluye, que la chimenea volcánica es notable con el primero modelo, sin embargo el número de interacciones es mayor cuando el cráter está conformado por roca estándar.

# Capítulo 7

## Conclusiones y Recomendaciones

En el presente trabajo se desarrolló la simulación de la interacción de una fuente radioactiva de  $C_{60}$  con un detector de centelleo con el fin de lograr su calibración obteniendo la detección de los picos característicos de energía de radiación gamma en los valores 1,17 y 1,33 MeV satisfactoriamente.

Teniendo en cuenta parámetros iniciales como la simulación del complejo volcánico, la geometría de un detector de centelleo en GEANT4 [1], la densidad del material que conforma el modelo de la estructura y las coordenadas geográficas del punto de observación suministradas por el Servicio Geológico Colombiano [3]; se implementó las geometrías del VG y el detector de centelleo en la misma simulación. Además se ubicó dicho detector en el punto de mayor conveniencia que puede ser empleado en aplicaciones de muografía volcánica. El análisis se realizó con base en la distancia recorrida por los muones de lluvias atmosféricas extendidas a través de la estructura del Galeras y la accesibilidad de las diferentes zonas del volcán

En base, a la simulación de las EAS en CORSIKA, se estimó la energía de los muones que logran alcanzar a la altura de la ciudad de Pasto y se creó un código de un flujo de muones en función del ángulo cenital a diferentes energías que atraviesa la estructura volcánica y arriba al detector de centelleo.

Un siguiente paso, es calcular el perfil atmosférico de la ciudad de Pasto y poder obtener un flujo de muones mas próximo a la realidad. En un futuro cercano se desarrollará una simulación en escala real con herramientas computacionales más poderosas y se pretende hacer una medición preeliminar de flujo de muones con el detector prototipo en las 3 ubica-



ciones discutidas en este proyecto.

Cronológicamente, el trabajo realizado en la plataforma GEANT4 [1] fue presentado en la quinta escuela de partículas llevada a cabo en la Universidad de los ANDES en Mayo del 2017; siendo elegido como el mejor proyecto a nivel de pregrado, y ocupando el tercer lugar, entre trabajos a nivel de maestría y doctorado. En Agosto del 2017 fue presentado en la conferencia DPF (Division of Particles & Field) en FERMILAB titulado “Muon Tomography of Galeras Volcano: first results leded by young scientists in Colombia” presentado por Ph.D. David Martinez y Ph.D. Alex Tapia.

# Apéndice A

## Código del programa

A continuación se muestra el código de CORSIKA y GEANT4:

### Macro CORSIKA

```
RUNNR 200 run number
EVTNR 200 number of first shower event
SEED 2576 0 0 seed for hadronic part
SEED 28837 0 0 seed for EGS4 part
THETAP 60 60 range of zenith angle (degree)
DIRECT ~/corsika/CORSIKAWORKSPACE/ckrun/Energy_5E3/Angle_60/ output directory
ERANGE 5E3 5E3 energy range of primary particle (GeV)
PRMPAR 14 particle type of prim. particle
ESLOPE 0. slope of primary energy spectrum
NSHOW 1 number of showers to generate
PHIP -180. 180. range of azimuth angle (degree)
OBSLEV 2530.E2 Pasto observation level (in cm)
FIXCHI 0. starting altitude (g/cm**2)
MAGNET 27.1 -11.8 Pasto magnetic field (muT)
HADFLG 0 0 0 0 0 2 flags hadr.interact.&fragmentation
ECUTS 0.05 0.05 0.000264 0.000264 energy cuts for particles (hadr. muon elec. photon)
MUADDI T additional info for muons
MUMULT T muon multiple scattering angle
ELMFLG T T em. interaction flags (NKG,EGS)
STEPFC 1.0 mult. scattering step length fact.
RADNKG 200.E2 outer radius for NKG lat.dens.distr.
LONGI T 5. T T longit.distr. & step size & fit & out
ECTMAP 25000. cut on gamma factor for printout
MAXPRT 1 max. number of printed events
THIN 1.E-06 10000. 50. thinning level, maximum weight factor and radial distance
THINH 1. 100. hadronic thinning limit
PAROUT T T output tables
USER jairo user
DEBUG F 6 F 1000000 debug flag and log.unit for out
ATMOSPHERE 1 F atmospheric model
EXIT terminates input
```

# GEANT4

## Headers (Carpeta include)

### B1ActionInitialization.hh

```
//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
// By: Danilo Arturo Rodríguez
//
/// \file B1ActionInitialization.hh
/// \brief Definition of the B1ActionInitialization class

#ifndef B1ActionInitialization_h
#define B1ActionInitialization_h 1

#include "G4VUserActionInitialization.hh"

class B1DetectorConstruction;

/// Action initialization class.

class B1ActionInitialization : public G4VUserActionInitialization
{
public:
    B1ActionInitialization(B1DetectorConstruction*);
    virtual ~B1ActionInitialization();

    virtual void BuildForMaster() const;
    virtual void Build() const;
};
```

```

private:
    B1DetectorConstruction* fDetConstruction;
};
//.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

#endif

```

## B1Analysis.hh

```

//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
// By: Danilo Arturo Rodríguez
//
/// \file B4Analysis.hh
/// \brief Selection of the analysis technology

#ifndef B1Analysis_h
#define B1Analysis_h 1

#include "g4root.hh"

#endif

```

## B1DetectorConstruction.hh

```

//
// *****
// * License and Disclaimer *

```

```

// *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
// By: Danilo Arturo Rodríguez
//
/// \file B1DetectorConstruction.hh
/// \brief Definition of the B1DetectorConstruction class

#ifndef B1DetectorConstruction_h
#define B1DetectorConstruction_h 1

#include "G4VUserDetectorConstruction.hh"
#include "globals.hh"

class G4VPhysicalVolume;
class G4LogicalVolume;

/// Detector construction class to define materials and geometry.

class B1DetectorConstruction : public G4VUserDetectorConstruction
{
public:
    B1DetectorConstruction();
    virtual ~B1DetectorConstruction();

    virtual G4VPhysicalVolume* Construct();

    const G4VPhysicalVolume* GetAbsPsX() const;
    const G4VPhysicalVolume* GetAbsPsX1() const;
    const G4VPhysicalVolume* GetAbsPsY() const;
    const G4VPhysicalVolume* GetAbsPsY1() const;
    const G4VPhysicalVolume* GetGapPb() const;
    const G4VPhysicalVolume* GetGapPb1() const;

    G4VPhysicalVolume* fAbsPsX;
    G4VPhysicalVolume* fAbsPsX1;
    G4VPhysicalVolume* fAbsPsY;
    G4VPhysicalVolume* fAbsPsY1;
    G4VPhysicalVolume* fGapPb;

```

```

    G4VPhysicalVolume* fGapPb1;
};

inline const G4VPhysicalVolume* B1DetectorConstruction::GetAbsPsX() const {
    return fAbsPsX;
}

inline const G4VPhysicalVolume* B1DetectorConstruction::GetAbsPsX1() const {
    return fAbsPsX1;
}

inline const G4VPhysicalVolume* B1DetectorConstruction::GetAbsPsY() const {
    return fAbsPsY;
}

inline const G4VPhysicalVolume* B1DetectorConstruction::GetAbsPsY1() const {
    return fAbsPsY1;
}

inline const G4VPhysicalVolume* B1DetectorConstruction::GetGapPb() const {
    return fGapPb;
}

inline const G4VPhysicalVolume* B1DetectorConstruction::GetGapPb1() const {
    return fGapPb1;
}

//....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

#endif

```

## B1EventAction.hh

```

//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *

```

```

// *****
//
// By: Danilo Arturo Rodríguez
//
// \file B1EventAction.hh
// \brief Definition of the B1EventAction class

#ifndef B1EventAction_h
#define B1EventAction_h 1

#include "G4UserEventAction.hh"
#include "globals.hh"

class B1EventAction : public G4UserEventAction
{
public:
    B1EventAction();
    virtual ~B1EventAction();

    virtual void BeginOfEventAction(const G4Event* event);
    virtual void EndOfEventAction(const G4Event* event);

    void AddAbsPsX(G4double de, G4double dl);
    void AddAbsPsX1(G4double de, G4double dl);
    void AddAbsPsY(G4double de, G4double dl);
    void AddAbsPsY1(G4double de, G4double dl);
    void AddGapPb(G4double de, G4double dl);
    void AddGapPb1(G4double de, G4double dl);
    void AddPos(G4double de);

private:
    G4double fEnergyAbsPsX;
    G4double fEnergyAbsPsX1;
    G4double fEnergyAbsPsY;
    G4double fEnergyAbsPsY1;
    G4double fEnergyGapPb;
    G4double fEnergyGapPb1;
    G4double fTrackLAbsPsX;
    G4double fTrackLAbsPsX1;
    G4double fTrackLAbsPsY;
    G4double fTrackLAbsPsY1;
    G4double fTrackLGapPb;
    G4double fTrackLGapPb1;
    G4double fPos;

};

// inline functions

inline void B1EventAction::AddAbsPsX(G4double de, G4double dl) {
    fEnergyAbsPsX += de;
    fTrackLAbsPsX += dl;
}

inline void B1EventAction::AddAbsPsX1(G4double de, G4double dl) {
    fEnergyAbsPsX1 += de;
    fTrackLAbsPsX1 += dl;
}

```

```

inline void B1EventAction::AddAbsPsY(G4double de, G4double dl) {
    fEnergyAbsPsY += de;
    fTrackLabsPsY += dl;
}

inline void B1EventAction::AddAbsPsY1(G4double de, G4double dl) {
    fEnergyAbsPsY1 += de;
    fTrackLabsPsY1 += dl;
}

inline void B1EventAction::AddGapPb(G4double de, G4double dl) {
    fEnergyGapPb += de;
    fTrackLGapPb += dl;
}

inline void B1EventAction::AddGapPb1(G4double de, G4double dl) {
    fEnergyGapPb1 += de;
    fTrackLGapPb1 += dl;
}

inline void B1EventAction::AddPos(G4double de) {
    fPos = de;
}

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
#endif

```

## B1PrimaryGeneratorAction.hh

```

//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *

```



```

// * acceptance of all terms of the Geant4 Software license.          *
// *****
//
// By: Danilo Arturo Rodríguez
//
// \file B1PrimaryGeneratorAction.hh
// \brief Definition of the B1PrimaryGeneratorAction class

#ifndef B1PrimaryGeneratorAction_h
#define B1PrimaryGeneratorAction_h 1

#include "G4VUserPrimaryGeneratorAction.hh"
#include "G4ParticleGun.hh"
#include "globals.hh"
#include "G4GeneralParticleSource.hh"

class G4GeneralParticleSource;
class G4Event;

// The primary generator action class with particle gun.
//
// The default kinematic is a 6 MeV gamma, randomly distributed
// in front of the phantom across 80% of the (X,Y) phantom size.

class B1PrimaryGeneratorAction : public G4VUserPrimaryGeneratorAction
{
public:
    B1PrimaryGeneratorAction();
    virtual ~B1PrimaryGeneratorAction();

    // method from the base class
public:
    virtual void GeneratePrimaries(G4Event*);
    G4GeneralParticleSource* GetParticleGun() { return fParticleGun; };

private:
    G4GeneralParticleSource* fParticleGun;
};

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

#endif

```

## B1RunAction.hh

```

//
// *****
// * License and Disclaimer                                          *
// *                                                                *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders.                            *

```

```

// *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
// By: Danilo Arturo Rodríguez
//
/// \file B1RunAction.hh
/// \brief Definition of the B1RunAction class

#ifndef B1RunAction_h
#define B1RunAction_h 1

#include "G4UserRunAction.hh"
#include "globals.hh"

class G4Run;

/// Run action class
///
/// In EndOfRunAction(), it calculates the dose in the selected volume
/// from the energy deposit accumulated via stepping and event actions.
/// The computed dose is then printed on the screen.

class B1RunAction : public G4UserRunAction
{
public:
    B1RunAction();
    virtual ~B1RunAction();

    virtual void BeginOfRunAction(const G4Run*);
    virtual void EndOfRunAction(const G4Run*);
};

#endif

```

## B1SteppingAction.hh

```

//
// *****
// * License and Disclaimer *
// *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *

```

```

// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
// By: Danilo Arturo Rodríguez
//
/// \file B1SteppingAction.hh
/// \brief Definition of the B1SteppingAction class

#ifndef B1SteppingAction_h
#define B1SteppingAction_h 1

#include "G4UserSteppingAction.hh"
#include "globals.hh"

class B1EventAction;

class B1DetectorConstruction;

/// Stepping action class
///

class B1SteppingAction : public G4UserSteppingAction
{
public:
    B1SteppingAction(const B1DetectorConstruction* detectorConstruction,
                    B1EventAction* eventAction);
    virtual ~B1SteppingAction();

    virtual void UserSteppingAction(const G4Step* step);

private:
    const B1DetectorConstruction* fDetConstruction;
    B1EventAction* fEventAction;
};

//....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

#endif

```

EmStandardPhysics.hh

```

//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
/// \file electromagnetic/TestEm1/include/EmStandardPhysics.hh
/// \brief Definition of the EmStandardPhysics class
//
// By: Danilo Arturo Rodríguez
//
//....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....
//....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

#ifndef EmStandardPhysics_h
#define EmStandardPhysics_h 1

#include "G4VPhysicsConstructor.hh"
#include "globals.hh"

//....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

class EmStandardPhysics : public G4VPhysicsConstructor
{
public:
    EmStandardPhysics(const G4String& name = "standard");
    ~EmStandardPhysics();

public:
    // This method is dummy for physics
    virtual void ConstructParticle() {};

    // This method will be invoked in the Construct() method.
    // each physics process will be instantiated and
    // registered to the process manager of each particle type
    virtual void ConstructProcess();
};

//....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

```

```
#endif
```

## GammaPhysics.hh

```
//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
/// \file hadronic/Hadr03/include/GammaPhysics.hh
/// \brief Definition of the GammaPhysics class
//
// By: Danilo Arturo Rodríguez
//
//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

#ifndef GammaPhysics_h
#define GammaPhysics_h 1

#include "globals.hh"
#include "G4VPhysicsConstructor.hh"

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

class GammaPhysics : public G4VPhysicsConstructor
{
public:
    GammaPhysics(const G4String& name="gamma");
    ~GammaPhysics();

public:
    virtual void ConstructParticle() { };
    virtual void ConstructProcess();
};
```

```
};

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
#endif
```

## PhysicsList.hh

```
//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
/// \file hadronic/Hadr03/include/PhysicsList.hh
/// \brief Definition of the PhysicsList class
//
// By: Danilo Arturo Rodríguez
//

#ifndef PhysicsList_h
#define PhysicsList_h 1

#include "G4VModularPhysicsList.hh"
#include "globals.hh"

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

class PhysicsList: public G4VModularPhysicsList
{
public:
    PhysicsList();
    ~PhysicsList();

public:
    virtual void ConstructParticle();
};
```

```

    virtual void SetCuts();
};

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

#endif

```

## Clases Usuario (Carpeta src)

### B1ActionInitialization.cc

```

//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
// By: Danilo Alejandro Arturo Rodríguez
//
/// \file B1ActionInitialization.cc
/// \brief Implementation of the B1ActionInitialization class

#include "B1ActionInitialization.hh"
#include "B1PrimaryGeneratorAction.hh"
#include "B1RunAction.hh"
#include "B1EventAction.hh"
#include "B1SteppingAction.hh"
#include "B1DetectorConstruction.hh"

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

B1ActionInitialization::B1ActionInitialization
    (B1DetectorConstruction* detConstruction)

```

```

: G4UserActionInitialization(),
  fDetConstruction(detConstruction)
{}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

B1ActionInitialization::~B1ActionInitialization()
{}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

void B1ActionInitialization::BuildForMaster() const
{
  SetUserAction(new B1RunAction);
}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

void B1ActionInitialization::Build() const
{
  SetUserAction(new B1PrimaryGeneratorAction);
  SetUserAction(new B1RunAction);
  B1EventAction* eventAction = new B1EventAction;
  SetUserAction(eventAction);
  SetUserAction(new B1SteppingAction(fDetConstruction,eventAction));
}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

```

### B1DetectorConstruction.cc

```

//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****

```



```

//
// By: Danilo Arturo Rodríguez
//
/// \file B1DetectorConstruction.cc
/// \brief Implementation of the B1DetectorConstruction class

#include "B1DetectorConstruction.hh"

#include "G4NistManager.hh"
#include "G4Box.hh"
#include "G4Trap.hh"
#include "G4UnionSolid.hh"
#include "G4LogicalVolume.hh"
#include "G4PVPlacement.hh"
#include "G4SystemOfUnits.hh"
#include "G4RotationMatrix.hh"
#include "G4Transform3D.hh"

#include "G4Material.hh"
#include "G4PVReplica.hh"
#include "G4AutoDelete.hh"

#include "G4GeometryManager.hh"
#include "G4PhysicalVolumeStore.hh"
#include "G4LogicalVolumeStore.hh"
#include "G4SolidStore.hh"

#include "G4VisAttributes.hh"
#include "G4Colour.hh"

#include "G4PhysicalConstants.hh"

//Parser
#include "G4GDMLParser.hh"

//DETECTOR
//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

B1DetectorConstruction::B1DetectorConstruction()
: G4VUserDetectorConstruction(),
  fAbsPsX(0),
  fAbsPsX1(0),
  fAbsPsY(0),
  fAbsPsY1(0),
  fGapPb(0),
  fGapPb1(0)
{ }

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

B1DetectorConstruction::~B1DetectorConstruction()
{ }

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

G4VPhysicalVolume* B1DetectorConstruction::Construct()

{   G4RotationMatrix* rot = new G4RotationMatrix();

```

```
rot->rotateX(-120*deg);
rot->rotateY(175*deg);
rot->rotateZ(90*deg); // rotate

G4DMLParser parser;

parser.Read("crater_variable_rock.m.gdml");

G4VPhysicalVolume* W = parser.GetWorldVolume(); //world volume

// Get nist material manager
// Get nist material manager
G4NistManager* nistManager = G4NistManager::Instance();

// Build materials
// Lead material defined using NIST Manager

nistManager->FindOrBuildMaterial("G4_Galactic");
nistManager->FindOrBuildMaterial("G4_PLASTIC_SC_VINYLTOLUENE");

// Geometry parameters
G4int nofLayers = 21;
G4int noLayersXY = 10; // numero de capas de barras orientadas en x e y

G4double pDx1 = 0.2*cm;
G4double pDx2 = pDx1;
G4double pDy1 = nofLayers*pDx1+pDx1/2;
G4double pDy2 = pDy1;
G4double pDx3 = 0.000000000000000001*mm;
G4double pDx4 = pDx3;
G4double pDz = (sqrt(3)/2)*pDx1;
G4double pTheta = 0*deg;
G4double pDphi = 0*deg;
G4double pAlph1 = 0*deg;
G4double pAlph2 = 0*deg;

G4double worldSizeX = 3.8*cm;
G4double worldSizeY = 3.8*cm;
G4double worldSizeZ = 3.8*cm;

// Get materials
G4Material* world_mat = G4Material::GetMaterial("G4_Galactic");
G4Material* plasticsc = G4Material::GetMaterial("G4_PLASTIC_SC_VINYLTOLUENE");

// WorldDetect
//
G4ThreeVector pos2 = G4ThreeVector(-65.22440625936*mm, -557.57145183244*mm, -57*mm); // coordenadas originales(237490.512,
134555.754, 3908)

G4Box* solidWorld =
    new G4Box("WorldDetec", // its name
            worldSizeX, worldSizeY, worldSizeZ); // its size
```

```

G4LogicalVolume* logicWorld =
  new G4LogicalVolume(solidWorld,          //its solid
                     world_mat,          //its material
                     "WorldDetec");      //its name

  new G4PVPlacement(rot,
                   pos2,                  //rotation and position
                   logicWorld,            //its logical volume
                   "WorldDetec",         //its name
                   W->GetLogicalVolume(), //its mother volume
                   false,                 //no boolean operation
                   0,                      //copy number
                   true);                 //overlaps checking

//Bars

G4VSolid* calorS
= new G4Trap("calorS",
            pDz/2, 30*deg, pDphi, pDy1/2, nofLayers*(pDx1/2+pDx3/2), nofLayers*(pDx2/2+pDx4/2), pAlph1, pDy2/2, (
            nofLayers)*(pDx1/2+pDx3/2), (nofLayers)*(pDx2/2+pDx4/2), pAlph2);

G4LogicalVolume* calorLV
= new G4LogicalVolume(
  calorS,          // its solid
  world_mat,      // its material
  "calorLV");     // its name

for(G4int i = 0; i < noLayersXY; i++){
  new G4PVPlacement(
    0,              // no rotation
    G4ThreeVector(0,0,2*i*pDz), // its position
    calorLV,        // its logical volume
    "calor",        // its name
    logicWorld,     // its mother volume
    false,          // no boolean operation
    0,              // copy number
    true);          // checking overlaps
}

G4VSolid* trapS
= new G4Trap("trap",
            pDz/2, 30*deg, pDphi, pDy1/2, pDx1/2+pDx3/2, pDx2/2+pDx4/2, pAlph1, pDy2/2, pDx1/2+pDx3/2, pDx2/2+pDx4/2,
            pAlph2);

G4LogicalVolume* trapLV
= new G4LogicalVolume(
  trapS,          // its solid
  world_mat,     // its material
  "trap");       // its name

  new G4PVReplica(
    "trap",       // its name
    trapLV,      // its logical volume
    calorLV,     // its mother

```

```

        kXAxis,          // axis of replication
        nofLayers,     // number of replica
        (pDx1+pDx3)); // width of replica

G4VSolid* trap2S
= new G4Trap("trap2",
            pDz/2, pTheta, pDphi, pDy1/2, pDx1/2, pDx2/2, pAlph1, pDy2/2, pDx3/2, pDx4/2, pAlph2);

G4LogicalVolume* trap2LV
= new G4LogicalVolume(
    trap2S,          // its solid
    plasticsc,      // its material
    "trap2");       // its name

fAbsPsY
= new G4PVPlacement(
    0,              // no rotation
    G4ThreeVector(-pDx1/4,0,0), // its position
    trap2LV,       // its logical volume
    "trap2",       // its name
    trapLV,        // its mother volume
    false,         // no boolean operation
    0,             // copy number
    true);        // checking overlaps

G4VSolid* trap3S
= new G4Trap("trap3",
            pDz/2, pTheta, pDphi, pDy1/2, pDx3/2, pDx4/2, pAlph1, pDy2/2, pDx1/2, pDx2/2, pAlph2);

G4LogicalVolume* trap3LV
= new G4LogicalVolume(
    trap3S,          // its solid
    plasticsc,      // its material
    "trap3");       // its name

fAbsPsY1
= new G4PVPlacement(
    0,              // no rotation
    G4ThreeVector(pDx1/4,0,0), // its position
    trap3LV,       // its logical volume
    "trap3",       // its name
    trapLV,        // its mother volume
    false,         // no boolean operation
    0,             // copy number
    true);        // checking overlaps

//-----

G4VSolid* calor2S
= new G4Trap("calor2S",
            pDz/2, 30*deg, pDphi, pDy1/2, nofLayers*(pDx1/2+pDx3/2), nofLayers*(pDx2/2+pDx4/2), pAlph1, pDy2/2, (
            nofLayers)*(pDx1/2+pDx3/2), (nofLayers)*(pDx2/2+pDx4/2), pAlph2);

G4LogicalVolume* calor2LV
= new G4LogicalVolume(
    calor2S,          // its solid
    world_mat,       // its material
    "calor2LV");     // its name

```

```

G4RotationMatrix* rm = new G4RotationMatrix();

rm->rotateZ(90*deg);

for(G4int i = 0; i < noLayersXY; i++){
new G4PVPlacement(
    rm, // no rotation
    G4ThreeVector(0,0,(2*i+1)*pDz), // its position
    calor2LV, // its logical volume
    "calor2", // its name
    logicWorld, // its mother volume
    false, // no boolean operation
    0, // copy number
    true); // checking overlaps
}

G4VSolid* trap4S
= new G4Trap("trap4",
    pDz/2, 30*deg, pDphi, pDy1/2, pDx1/2+pDx3/2, pDx2/2+pDx4/2, pAlph1, pDy2/2, pDx1/2+pDx3/2, pDx2/2+pDx4/2,
    pAlph2);

G4LogicalVolume* trap4LV
= new G4LogicalVolume(
    trap4S, // its solid
    world_mat, // its material
    "trap4"); // its name

new G4PVReplica(
    "trap4", // its name
    trap4LV, // its logical volume
    calor2LV, // its mother
    kXAxis, // axis of replication
    noLayers, // number of replica
    (pDx1+pDx3)); // width of replica

G4VSolid* trap5S
= new G4Trap("trap5",
    pDz/2, pTheta, pDphi, pDy1/2, pDx1/2, pDx2/2, pAlph1, pDy2/2, pDx3/2, pDx4/2, pAlph2);

G4LogicalVolume* trap5LV
= new G4LogicalVolume(
    trap5S, // its solid
    plasticsc, // its material
    "trap5"); // its name

fAbsPsX
= new G4PVPlacement(
    0, // no rotation
    G4ThreeVector(-pDx1/4,0,0), // its position
    trap5LV, // its logical volume
    "trap5", // its name
    trap4LV, // its mother volume
    false, // no boolean operation
    0, // copy number
    true); // checking overlaps

```

```

G4VSolid* trap6S
= new G4Trap("trap6",
            pDz/2, pTheta, pDphi, pDy1/2, pDx3/2, pDx4/2, pAlph1, pDy2/2, pDx1/2, pDx2/2, pAlph2);

G4LogicalVolume* trap6LV
= new G4LogicalVolume(
    trap6S,          // its solid
    plasticsc,      // its material
    "trap6");       // its name

fAbsPsX1
= new G4PVPlacement(
    0,              // no rotation
    G4ThreeVector(pDx1/4,0,0), // its position
    trap6LV,       // its logical volume
    "trap3",       // its name
    trap4LV,       // its mother volume
    false,         // no boolean operation
    0,             // copy number
    true); // checking overlaps

//
//always return the physical World
//
logicWorld->SetVisAttributes (G4VisAttributes::Invisible);

return W;
}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

```

### B1EventAction.cc

```

//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes,nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *

```

```

// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license.      *
// *****
//
// By: Danilo Arturo Rodríguez
//
/// \file B1EventAction.cc
/// \brief Implementation of the B1EventAction class

#include "B1EventAction.hh"
#include "B1RunAction.hh"
#include "B1Analysis.hh"

#include "G4RunManager.hh"
#include "G4Event.hh"
#include "G4UnitsTable.hh"

#include "Randomize.hh"
#include <iomanip>

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

B1EventAction::B1EventAction()
: G4UserEventAction(),
  fEnergyAbsPsX(0),
  fEnergyAbsPsX1(0),
  fEnergyAbsPsY(0),
  fEnergyAbsPsY1(0),
  fEnergyGapPb(0),
  fEnergyGapPb1(0),
  fTrackLAbsPsX(0),
  fTrackLAbsPsX1(0),
  fTrackLAbsPsY(0),
  fTrackLAbsPsY1(0),
  fTrackLGapPb(0),
  fTrackLGapPb1(0),
  fPos(0)
{}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

B1EventAction::~B1EventAction()
{}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

void B1EventAction::BeginOfEventAction(const G4Event* /*event*/)
{
  // initialisation per event
  fEnergyAbsPsX = 0.;
  fEnergyAbsPsX1 = 0.;
  fEnergyAbsPsY = 0.;
  fEnergyAbsPsY1 = 0.;
  fEnergyGapPb = 0.;
  fEnergyGapPb1 = 0.;
  fTrackLAbsPsX = 0.;
  fTrackLAbsPsX1 = 0.;
}

```

```

    fTrackLabsPsY = 0.;
    fTrackLabsPsY1 = 0.;
    fTrackLGapPb = 0.;
    fTrackLGapPb1 = 0.;
    fPos = 0;
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

void B1EventAction::EndOfEventAction(const G4Event* event)
{
    // Accumulate statistics
    //

    // get analysis manager
    G4AnalysisManager* analysisManager = G4AnalysisManager::Instance();

    // fill histograms
    analysisManager->FillH1(1, fEnergyAbsPsX + fEnergyAbsPsX1 + fEnergyAbsPsY + fEnergyAbsPsY1 + fEnergyGapPb + fEnergyGapPb1);
    analysisManager->FillH1(2, fTrackLabsPsX + fTrackLabsPsX1 + fTrackLabsPsY + fTrackLabsPsY1 + fTrackLGapPb + fTrackLGapPb1);
    analysisManager->FillH1(3, fPos);

    // fill ntuple
    //analysisManager->FillNtupleDColumn(0, fEnergyAbsPsX + fEnergyAbsPsX1 + fEnergyAbsPsY + fEnergyAbsPsY1 + fEnergyGapPb +
    //    fEnergyGapPb1);
    //analysisManager->FillNtupleDColumn(1, fTrackLabsPsX + fTrackLabsPsX1 + fTrackLabsPsY + fTrackLabsPsY1 + fTrackLGapPb +
    //    fTrackLGapPb1);
    //analysisManager->AddNtupleRow();

    // Print per event (modulo n)
    //
    G4int eventID = event->GetEventID();
    G4int printModulo = G4RunManager::GetRunManager()->GetPrintProgress();
    if ( ( printModulo > 0 ) && ( eventID % printModulo == 0 ) ) {
        G4cout << "---> End of event: " << eventID << G4endl;

        G4cout
            << "   Total energy: " << std::setw(7)
                << G4BestUnit(fEnergyAbsPsX + fEnergyAbsPsX1 + fEnergyAbsPsY + fEnergyAbsPsY1 +
                    fEnergyGapPb + fEnergyGapPb1,"Energy")
            << "   Total track length: " << std::setw(7)
                << G4BestUnit(fTrackLabsPsX + fTrackLabsPsX1 + fTrackLabsPsY + fTrackLabsPsY1 +
                    fTrackLGapPb + fTrackLGapPb1,"Length")
            << G4endl;
    }
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

```

B1PrimaryGeneratorAction.cc



```

//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
/// \file eventgenerator/Gun/src/PrimaryGeneratorGPS.cc
/// \brief Implementation of the PrimaryGeneratorfAction class
//
//
// By: Danilo Arturo Rodríguez
//
//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....
//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

#include "B1PrimaryGeneratorAction.hh"

#include "G4Event.hh"
#include "G4GeneralParticleSource.hh"

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

B1PrimaryGeneratorAction::B1PrimaryGeneratorAction()
: G4VUserPrimaryGeneratorAction(), fParticleGun(0)
{
    fParticleGun = new G4GeneralParticleSource();
}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

B1PrimaryGeneratorAction::~B1PrimaryGeneratorAction()
{
    delete fParticleGun;
}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

void B1PrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)

```

```

{
  //this function is called at the beginning of event
  //
  fParticleGun->GeneratePrimaryVertex(anEvent);
}

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

```

### B1RunAction.cc

```

//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes,nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
// By: Danilo Arturo Rodríguez
//
/// \file B1RunAction.cc
/// \brief Implementation of the B1RunAction class

#include "B1RunAction.hh"
#include "B1Analysis.hh"

#include "G4Run.hh"
#include "G4RunManager.hh"
#include "G4UnitsTable.hh"
#include "G4SystemOfUnits.hh"

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

B1RunAction::B1RunAction()
: G4UserRunAction()
{
  // set printing event number per each event
  G4RunManager::GetRunManager()->SetPrintProgress(1);
}

```

```

// Create analysis manager
// The choice of analysis technology is done via selectin of a namespace
// in B1Analysis.hh
G4AnalysisManager* analysisManager = G4AnalysisManager::Instance();
G4cout << "Using " << analysisManager->GetType() << G4endl;

// Create directories
//analysisManager->SetHistoDirectoryName("histograms");
//analysisManager->SetNtupleDirectoryName("ntuple");
analysisManager->SetVerboseLevel(1);
analysisManager->SetFirstHistoId(1);

// Book histograms, ntuple
//

// Creating histograms
analysisManager->CreateH1("1","Energy ABS", 140, 0., 80*MeV);
analysisManager->CreateH1("2","trackL in absorber", 140, 0., 1*m);
analysisManager->CreateH2("3","PosXY", 140, -1*m, 1*m, 140, -1*m, 1*m);
analysisManager->CreateH2("4","PosXZ", 140, -1*m, 1*m, 140, -1*m, 1*m);
analysisManager->CreateH2("5","PosYZ", 140, -1*m, 1*m, 140, -1*m, 1*m);
analysisManager->CreateH3("6","PosXYZ", 140, -1*m, 1*m, 140, -1*m, 1*m, 140, -1*m, 1*m);
analysisManager->CreateH1("7","pos", 140, -1*m, 1*m);

// Creating ntuple
//
analysisManager->CreateNtuple("B1", "Edep");
analysisManager->CreateNtupleSColumn("Process");
analysisManager->CreateNtupleSColumn("Particle");
analysisManager->CreateNtupleDColumn("PositionX");
analysisManager->CreateNtupleDColumn("PositionY");
analysisManager->CreateNtupleDColumn("PositionZ");

analysisManager->CreateNtupleDColumn("Ek");

analysisManager->FinishNtuple();
}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

B1RunAction::~B1RunAction()
{
  delete G4AnalysisManager::Instance();
}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

void B1RunAction::BeginOfRunAction(const G4Run*)
{
  // Get analysis manager
  G4AnalysisManager* analysisManager = G4AnalysisManager::Instance();

  // Open an output file
  //

```

```

G4String fileName = "B1";
analysisManager->OpenFile(fileName);

}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

void B1RunAction::EndOfRunAction(const G4Run*)
{
  // print histogram statistics
  //
  G4AnalysisManager* analysisManager = G4AnalysisManager::Instance();
  if ( analysisManager->GetH1(1) ) {
    G4cout << G4endl << " ----> print histograms statistic ";
    if(isMaster) {
      G4cout << "for the entire run " << G4endl << G4endl;
    }
    else {
      G4cout << "for the local thread " << G4endl << G4endl;
    }

    G4cout << " EAbs : mean = "
      << G4BestUnit(analysisManager->GetH1(1)->mean(), "Energy")
      << " rms = "
      << G4BestUnit(analysisManager->GetH1(1)->rms(), "Energy") << G4endl;

  }

  // save histograms & ntuple
  //
  analysisManager->Write();
  analysisManager->CloseFile();
}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

```

### B1SteppingAction.cc

```

//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *

```

```

// * for the full disclaimer and the limitation of liability.      *
// *                                                              *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration.                 *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license.     *
// *****
//
// By: Danilo Arturo Rodríguez
//
/// \file B1SteppingAction.cc
/// \brief Implementation of the B1SteppingAction class

#include "B1SteppingAction.hh"
#include "B1EventAction.hh"
#include "B1DetectorConstruction.hh"
#include "B1Analysis.hh"
#include "G4SystemOfUnits.hh"

#include "G4Step.hh"
#include "G4RunManager.hh"

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

B1SteppingAction::B1SteppingAction(const B1DetectorConstruction* detectorConstruction,
                                   B1EventAction* eventAction)
: G4UserSteppingAction(),
  fDetConstruction(detectorConstruction),
  fEventAction(eventAction)
{}

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

B1SteppingAction::~B1SteppingAction()
{}

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

void B1SteppingAction::UserSteppingAction(const G4Step* step)
{
    // Collect energy and track length step by step

    G4String procName = step->GetPostStepPoint()->GetProcessDefinedStep()->GetProcessName();
    G4String particleName = step->GetTrack()->GetDynamicParticle()->GetDefinition()->GetParticleName();
    G4StepPoint* prepoint = step->GetPreStepPoint();
    G4StepPoint* postpoint = step->GetPostStepPoint();
    G4ThreeVector pos1 = prepoint->GetPosition();
    G4ThreeVector pos2 = postpoint->GetPosition();
    G4double posX = pos1.x();
    G4double posY = pos1.y();
    G4double posZ = pos1.z();

    G4double trackid = step->GetTrack()->GetTrackID();
    if (trackid == 1){

        // get volume of the current step

```

```
G4VPhysicalVolume* volume
= step->GetPreStepPoint()->GetTouchableHandle()->GetVolume();

// energy deposit

G4double edep = step->GetTotalEnergyDeposit();
G4double kinEnergy = step->GetTrack()->GetKineticEnergy();

// step length

G4double stepLength = 0.;

if ( step->GetTrack()->GetDefinition()->GetPDGCharge() != 0. ) {
    stepLength = step->GetStepLength();
}

if ( volume == fDetConstruction->GetAbsPsX() ) {
    fEventAction->AddAbsPsX(edep,stepLength);
    fEventAction->AddPos(pos1.x());
}

if ( volume == fDetConstruction->GetAbsPsX1() ) {
    fEventAction->AddAbsPsX1(edep,stepLength);
}

if ( volume == fDetConstruction->GetAbsPsY() ) {
    fEventAction->AddAbsPsY(edep,stepLength);
}

if ( volume == fDetConstruction->GetAbsPsY1() ) {
    fEventAction->AddAbsPsY1(edep,stepLength);
}

if ( volume == fDetConstruction->GetGapPb() ) {
    fEventAction->AddGapPb(edep,stepLength);
}

if ( volume == fDetConstruction->GetGapPb1() ) {
    fEventAction->AddGapPb1(edep,stepLength);
}

G4AnalysisManager* analysisManager = G4AnalysisManager::Instance();

if( volume == fDetConstruction->GetAbsPsX() || volume == fDetConstruction->GetAbsPsX1() || volume == fDetConstruction->
    GetAbsPsY() || volume == fDetConstruction->GetAbsPsY1() )
{
    analysisManager->FillH2(1,posX,posY);
}
```

```

analysisManager->FillH2(2,posX,posZ);
analysisManager->FillH2(3,posY,posZ);

analysisManager->FillH3(1,posX,posY,posZ);

}

analysisManager->FillNtupleSColumn(0,procName);
analysisManager->FillNtupleSColumn(1,particleName);
analysisManager->FillNtupleDColumn(2,posX);
analysisManager->FillNtupleDColumn(3,posY);
analysisManager->FillNtupleDColumn(4,posZ);
analysisManager->FillNtupleDColumn(5,kinEnergy);

analysisManager->AddNtupleRow();
}
}
//.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

```

### EmStandardPhysics.cc

```

//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes,nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
// By: Danilo Arturo Rodriguez

```

```

// GEANT4 tag $Name: not supported by cvs2svn $
//
//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

#include "EmStandardPhysics.hh"
#include "G4ParticleDefinition.hh"
#include "G4ProcessManager.hh"
#include "G4PhysicsListHelper.hh"

#include "G4ComptonScattering.hh"
#include "G4GammaConversion.hh"
#include "G4PhotoElectricEffect.hh"
#include "G4RayleighScattering.hh"
#include "G4KleinNishinaModel.hh"

#include "G4eMultipleScattering.hh"
#include "G4eIonisation.hh"
#include "G4eBremsstrahlung.hh"
#include "G4eplusAnnihilation.hh"

#include "G4MuMultipleScattering.hh"
#include "G4MuIonisation.hh"
#include "G4MuBremsstrahlung.hh"
#include "G4MuPairProduction.hh"

#include "G4hMultipleScattering.hh"
#include "G4hIonisation.hh"
#include "G4hBremsstrahlung.hh"
#include "G4hPairProduction.hh"

#include "G4ionIonisation.hh"
#include "G4IonParametrisedLossModel.hh"
#include "G4NuclearStopping.hh"

#include "G4EmProcessOptions.hh"
#include "G4MscStepLimitType.hh"

#include "G4LossTableManager.hh"
#include "G4UAtomicDeexcitation.hh"

#include "G4SystemOfUnits.hh"

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

EmStandardPhysics::EmStandardPhysics(const G4String& name)
: G4VPhysicsConstructor(name)
{}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

EmStandardPhysics::~EmStandardPhysics()
{}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

void EmStandardPhysics::ConstructProcess()
{

```



```
G4PhysicsListHelper* ph = G4PhysicsListHelper::GetPhysicsListHelper();

// Add standard EM Processes
//
auto aParticleIterator=GetParticleIterator();
aParticleIterator->reset();
while( (*aParticleIterator)() ){
  G4ParticleDefinition* particle = aParticleIterator->value();
  G4String particleName = particle->GetParticleName();

  if (particleName == "gamma") {

    ph->RegisterProcess(new G4PhotoElectricEffect, particle);
    ph->RegisterProcess(new G4ComptonScattering, particle);
    ph->RegisterProcess(new G4GammaConversion, particle);

  } else if (particleName == "e-") {

    ph->RegisterProcess(new G4eMultipleScattering(), particle);
    ph->RegisterProcess(new G4eIonisation, particle);
    ph->RegisterProcess(new G4eBremsstrahlung(), particle);

  } else if (particleName == "e+") {

    ph->RegisterProcess(new G4eMultipleScattering(), particle);
    ph->RegisterProcess(new G4eIonisation, particle);
    ph->RegisterProcess(new G4eBremsstrahlung(), particle);
    ph->RegisterProcess(new G4eplusAnnihilation(), particle);

  } else if (particleName == "mu+" ||
             particleName == "mu-" ) {

    ph->RegisterProcess(new G4muMultipleScattering(), particle);
    ph->RegisterProcess(new G4muIonisation, particle);
    ph->RegisterProcess(new G4muBremsstrahlung(), particle);
    ph->RegisterProcess(new G4muPairProduction(), particle);

  } else if ( particleName == "proton" ||
             particleName == "pi-" ||
             particleName == "pi+" ) {

    ph->RegisterProcess(new G4hMultipleScattering(), particle);
    ph->RegisterProcess(new G4hIonisation, particle);

  } else if( particleName == "alpha" ||
            particleName == "He3" ) {

    ph->RegisterProcess(new G4hMultipleScattering(), particle);
    ph->RegisterProcess(new G4ionIonisation, particle);
    ph->RegisterProcess(new G4NuclearStopping(), particle);

  } else if( particleName == "GenericIon" ) {

    ph->RegisterProcess(new G4hMultipleScattering(), particle);
    G4ionIonisation* ionIoni = new G4ionIonisation();
    ionIoni->SetEmModel(new G4IonParametrisedLossModel());
    ph->RegisterProcess(ionIoni, particle);
    ph->RegisterProcess(new G4NuclearStopping(), particle);
  }
}
```

```

} else if (!(particle->IsShortLived()) &&
            (particle->GetPDGCharge() != 0.0) &&
            (particle->GetParticleName() != "chargedgeantino")) {

    //all others charged particles except geantino
    ph->RegisterProcess(new G4hMultipleScattering(), particle);
    ph->RegisterProcess(new G4hIonisation(), particle);
}
}

// Em options
//
// Main options and setting parameters are shown here.
// Several of them have default values.
//
G4EmProcessOptions emOptions;

//ionisation
//
emOptions.SetStepFunction(1., 1*mm); //default= 0.1, 100*um

// Deexcitation
//
G4VAtomDeexcitation* de = new G4UAtomicDeexcitation();
de->SetFluo(true);
de->SetAuger(true);
G4LossTableManager::Instance()->SetAtomDeexcitation(de);
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

```

## GammaPhysics.cc

```

//
// *****
// * License and Disclaimer *
// * * * * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * * * * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * * * * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *

```

```

// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license.      *
// *****
//
// \file hadronic/Hadr03/src/GammaPhysics.cc
// \brief Implementation of the GammaPhysics class
//
// By: Danilo Arturo Rodríguez
//
//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

#include "GammaPhysics.hh"

#include "G4ParticleDefinition.hh"
#include "G4ProcessManager.hh"

// Processes

#include "G4PhotoNuclearProcess.hh"
#include "G4CascadeInterface.hh"

#include "G4SystemOfUnits.hh"

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

GammaPhysics::GammaPhysics(const G4String& name)
: G4VPhysicsConstructor(name)
{ }

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

GammaPhysics::~GammaPhysics()
{ }

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

void GammaPhysics::ConstructProcess()
{
    G4ProcessManager* pManager = G4Gamma::Gamma()->GetProcessManager();
    //
    G4PhotoNuclearProcess* process = new G4PhotoNuclearProcess();
    //
    G4CascadeInterface* bertini = new G4CascadeInterface();
    bertini->SetMaxEnergy(10*GeV);
    process->RegisterMe(bertini);
    //
    pManager->AddDiscreteProcess(process);
}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

```

PhysicsList.cc

```

//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
/// \file hadronic/Hadr03/src/PhysicsList.cc
/// \brief Implementation of the PhysicsList class
//
// By: Danilo Arturo Rodríguez

//....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

#include "PhysicsList.hh"

#include "G4SystemOfUnits.hh"
#include "G4UnitsTable.hh"

#include "EmStandardPhysics.hh"
#include "G4DecayPhysics.hh"
#include "G4RadioactiveDecayPhysics.hh"

#include "G4HadronElasticPhysicsHP.hh"
#include "G4HadronPhysicsFTFP_BERT_HP.hh"
#include "G4HadronPhysicsQGSP_BIC_HP.hh"
#include "G4HadronInelasticQBBC.hh"
#include "G4HadronPhysicsINCLXX.hh"
#include "G4IonElasticPhysics.hh"
#include "G4IonPhysics.hh"
#include "G4IonINCLXXPhysics.hh"
#include "GammaPhysics.hh"

// particles

#include "G4BosonConstructor.hh"
#include "G4LeptonConstructor.hh"
#include "G4MesonConstructor.hh"
#include "G4BosonConstructor.hh"

```

```

#include "G4BaryonConstructor.hh"
#include "G4IonConstructor.hh"
#include "G4ShortLivedConstructor.hh"

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

PhysicsList::PhysicsList()
:G4VModularPhysicsList()
{
    G4int verb = 1;
    SetVerboseLevel(verb);

    // EM physics
    RegisterPhysics(new EmStandardPhysics("standard"));

    // Decay
    RegisterPhysics(new G4DecayPhysics());

    // Radioactive decay
    RegisterPhysics(new G4RadioactiveDecayPhysics());

    // Hadron Elastic scattering
    RegisterPhysics( new G4HadronElasticPhysicsHP(verb) );

    // Hadron Inelastic physics
    /* RegisterPhysics( new G4HadronPhysicsFTFP_BERT_HP(verb));
    RegisterPhysics( new G4HadronPhysicsQGSP_BIC_HP(verb));
    RegisterPhysics( new G4HadronInelasticQBBC(verb));
    RegisterPhysics( new G4HadronPhysicsINCLXX(verb));

    // Ion Elastic scattering
    RegisterPhysics( new G4IonElasticPhysics(verb));

    // Ion Inelastic physics
    RegisterPhysics( new G4IonPhysics(verb));
    RegisterPhysics( new G4IonINCLXXPhysics(verb));

    // Gamma-Nuclear Physics
    RegisterPhysics( new GammaPhysics("gamma"));*/
}

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

PhysicsList::~PhysicsList()
{ }

//...ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....

void PhysicsList::ConstructParticle()
{
    G4BosonConstructor pBosonConstructor;
    pBosonConstructor.ConstructParticle();

    G4LeptonConstructor pLeptonConstructor;
    pLeptonConstructor.ConstructParticle();
}

```

```

G4MesonConstructor pMesonConstructor;
pMesonConstructor.ConstructParticle();

G4BaryonConstructor pBaryonConstructor;
pBaryonConstructor.ConstructParticle();

G4IonConstructor pIonConstructor;
pIonConstructor.ConstructParticle();

G4ShortLivedConstructor pShortLivedConstructor;
pShortLivedConstructor.ConstructParticle();
}

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

void PhysicsList::SetCuts()
{
SetCutValue(0*mm, "proton");
SetCutValue(10*mm, "e-");
SetCutValue(10*mm, "e+");
SetCutValue(10*mm, "gamma");
}

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

```

### tesis.cc

```

//
// *****
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes,nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
// By: Danilo Arturo Rodriguez
//

```

```

///
///

#include "B1DetectorConstruction.hh"
#include "B1ActionInitialization.hh"
#include "PhysicsList.hh"

#ifdef G4MULTITHREADED
#include "G4MTRunManager.hh"
#else
#include "G4RunManager.hh"
#endif

#include "G4UImanager.hh"
#include "G4UIcommand.hh"
#include "FTFP_BERT.hh"

#include "Randomize.hh"

#include "G4VisExecutive.hh"
#include "G4UIExecutive.hh"

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

namespace {
void PrintUsage() {
    G4cerr << " Usage: " << G4endl;
    G4cerr << " exampleB1 [-m macro ] [-u UIsession] [-t nThreads]" << G4endl;
    G4cerr << "   note: -t option is available only for multi-threaded mode."
        << G4endl;
}
}

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

int main(int argc, char** argv)
{
    // Evaluate arguments
    //
    if ( argc > 7 ) {
        PrintUsage();
        return 1;
    }

    G4String macro;
    G4String session;
#ifdef G4MULTITHREADED
    G4int nThreads = 0;
#endif
    #endif
    for ( G4int i=1; i<argc; i=i+2 ) {
        if ( G4String(argv[i]) == "-m" ) macro = argv[i+1];
        else if ( G4String(argv[i]) == "-u" ) session = argv[i+1];
#ifdef G4MULTITHREADED
        else if ( G4String(argv[i]) == "-t" ) {
            nThreads = G4UIcommand::ConvertToInt(argv[i+1]);
        }
    }
#endif
}

```

```

else {
    PrintUsage();
    return 1;
}
}

// Detect interactive mode (if no macro provided) and define UI session
//
G4UIExecutive* ui = 0;
if ( ! macro.size() ) {
    ui = new G4UIExecutive(argc, argv);
}

// Choose the Random engine
//
G4Random::setTheEngine(new CLHEP::RanecuEngine);

//Random runs
//-----
long seeds[2];
time_t systime = time(NULL);
seeds[0] = (long) systime;
seeds[1] = (long) (systime * G4UniformRand());
G4Random::setTheSeeds(seeds);
//-----

// Construct the default run manager
//
#ifdef G4MULTITHREADED
G4MTRunManager * runManager = new G4MTRunManager;
if ( nThreads > 0 ) {
    runManager->SetNumberOfThreads(nThreads);
}
#else
G4RunManager * runManager = new G4RunManager;
#endif

// Set mandatory initialization classes
//
B1DetectorConstruction* detConstruction = new B1DetectorConstruction();
runManager->SetUserInitialization(detConstruction);

PhysicsList* phys = new PhysicsList;
runManager->SetUserInitialization(phys);

B1ActionInitialization* actionInitialization
    = new B1ActionInitialization(detConstruction);
runManager->SetUserInitialization(actionInitialization);

// Initialize visualization
//
G4VisManager* visManager = new G4VisExecutive;
// G4VisExecutive can take a verbosity argument - see /vis/verbose guidance.
// G4VisManager* visManager = new G4VisExecutive("Quiet");
visManager->Initialize();

// Get the pointer to the User Interface manager
G4UIManager* UImanager = G4UIManager::GetUIpointer();

```



```
// Process macro or start UI session
//
if ( macro.size() ) {
    // batch mode
    G4String command = "/control/execute ";
    UImanager->ApplyCommand(command+macro);
}
else {
    // interactive mode : define UI session
    UImanager->ApplyCommand("/control/execute init_vis.mac");
    if (ui->IsGUI()) {
        UImanager->ApplyCommand("/control/execute icons.mac");
    }
    ui->SessionStart();
    delete ui;
}

// Job termination
// Free the store: user actions, physics_list and detector_description are
// owned and deleted by the run manager, so they should not be deleted
// in the main() program !

delete visManager;
delete runManager;
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
```

# Bibliografía

- [1] J. A LLISON ET AL. , *Geant4 Developments and Applications, IEEE Transactions on Nuclear Science 53 No. 1, (2006) 270-278.*
- [2] HECK ET AL. *CORSIKA: A Monte Carlo Code to Simulate Extensive Air Showers. CORSIKA 7.4004.*
- [3] SERVICIO GEOLÓGICO COLOMBIANO <http://www2.sgc.gov.co/Pasto/Volcanes/Volcan-Galeras/Generalidades.aspx>
- [4] W. F. HANLON. *The Energy Spectrum of Ultra High Energy Cosmic Rays Measured by the High Resolution Fly's Eye Observatory in Stereoscopic Mode. ProQuest, 2008.*
- [5] MINAYA FLORES IGNACIO *Composition of ultra-high energy cosmic radiation from the Pierre Auger Observatory data*
- [6] NAGAMINE KAMETADA. *Introductory Muon Science*
- [7] E. P. GEORGE. *Cosmic rays measure overburden of tunnel. Commonwealth Engineer, 455, 1955.*
- [8] L.W. ALVAREZ, J.A. ANDERSON, F. EL BEDWEI, M. SHARKAWI, AND L. YAZOLINO. *Search for hidden chambers in the pyramids. Science, 167(3919):832-839, 1970'.*
- [9] DR. ARIGA AKITA *Study of Active Glacier 3D Morphology with High Energy Particle Detectors*
- [10] N. LESPARRE, D. GIBERT, J. MARTEAU, Y. DÉCLAIS, D. CARBONE, AND E. GALICHET. *Geophysical muon imaging: feasibility and limits.*
- [11] S. OSTAPCHENKO *QGSJET-II, PRD 83 (2011) 014018.*
- [12] H.C. FESEFELDT *GHEISHA program, Technical Report PITHA 85-02 (1985).*

- [13] NOAA NATIONAL CENTERS FOR ENVIRONMENTAL INFORMATION  
*Magnetic Field Calculators*, <http://www.ngdc.noaa.gov/geomag-web/>.
- [14] ICHEP 2016 [https://indico.cern.ch/event/432527/contributions/1071873/attachments/1333362/2004792/poster\\_tomografia\\_final.pdf](https://indico.cern.ch/event/432527/contributions/1071873/attachments/1333362/2004792/poster_tomografia_final.pdf)
- [15] K. GREISEN, *ANN. T Rev. Nucl. Sci.* 10 (1960) 63.
- [16] K. KAMATA, J. NISHIMURA *Prog. Theoret. Phys. Suppl.* 6 (1958) 93.
- [17] KASCADE-GRANDE COLLAB *Supanitsky PhD Thesis. Detectores de Superficie y la Composicion Quimica de los Rayos Cosmicos* (2007).
- [18] A TAPIA ET. AL *Study of the chemical composition of high energy cosmic rays using the muon LDF of EAS between  $10^{17,25}$  eV and  $10^{17,50}$  eV*
- [19] RENE BRUN AND FONS RADEMAKERS. *ROOT - An Object Oriented Data Analysis Framework. Proceedings AIHENP 96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. Meth. in Phys. Res. A 389 (1997) 81-86. See also <http://root.cern.ch/>, 1997.*
- [20] L. A LIAGA , L. B AGBY , D.A. M ARTINEZ C AICEDO ET AL  
*Design, Calibration, and Performance of the MINERvA Detector*, [arXiv:1305.5199](https://arxiv.org/abs/1305.5199)  
[physics.ins-det] 2013
- [21] SOLIDWORKS <http://www.solidworks.es/>
- [22] THE 3D CAD TOOL FOR RADIATION SHIELDING ANALYSIS  
<http://www.fastrad.net/>
- [23] GDML USER'S GUIDE <http://gdml.web.cern.ch/GDML/doc/GDMLmanual.pdf>
- [24] BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, OIML *International vocabulary of basic and general terms in metrology, VIM. ISO, Ginebra (1993)*
- [25] ATSDR [https://www.atsdr.cdc.gov/es/phs/es\\_phs33.pdf](https://www.atsdr.cdc.gov/es/phs/es_phs33.pdf)

- [26] COBALTO 60 *http : /caebis.cnea.gov.ar/iden/conoc\_la\_energia  
\_nucx/capitulo\_4\_difusion/la\_tecnologia\_nuclear/cobalto\_60.htm*
- [27] CONVERSION DE COORDENADAS *http://www.atlascajamarca.info/conversor/*