

Radiación-Materia: GEANT4 Hands On!

Un recurso educativo
orientado a profesores-investigadores:
Alfabetización Científica y Tecnológica

*Jorge Enrique García Farieta, PhD
Julián Andrés Salamanca Bernal, PhD
Diego Julián Rodríguez Patarroyo, PhD*



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Doctorado
en Ingeniería
UNIVERSIDAD DISTRITAL
"FRANCISCO JOSÉ DE CALDAS"

Jorge Enrique García Farieta



Licenciado en Física, Físico, Especialización en Gerencia de Proyectos, Maestro en Física y Doctor en Física. Profesor Universidad Distrital. Colabora en el consorcio EUCLID para investigación de energía y materia oscura en el universo. Participa en proyectos de I+D+i en

BigData en cosmología y astrofísica, y, alfabetización científica-tecnológica orientada a profesores-investigadores.

Julián Andrés Salamanca Bernal



Licenciado en Física, Maestría en Física, PhD en Física Aplicada (Idaho State University- Thomas Jefferson Laboratory-USA). Profesor Universidad Distrital. Participa en

proyectos de I+D+i en BigData, inteligencia computacional, física experimental de altas-energías, fuentes alternas de energía y alfabetización científica-tecnológica orientada a profesores-investigadores.

Diego Julián Rodríguez Patarroyo



Físico, Maestría en Física y Doctor en Ingeniería. Profesor Universidad Distrital. Participa en proyectos de investigación en I+D+I en BigData orientados a las fuen-

tes alternas de energía y análisis de la interacción radiación-materia. Participó en la colaboración ATLAS-LHC desarrollando algoritmos para sub-sistemas de detección.



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Doctorado
en Ingeniería
UNIVERSIDAD DISTRITAL "FRANCISCO JOSÉ DE CALDAS"

Radiación-Materia: GEANT4 Hands On!

Un recurso educativo orientado a profesores-investigadores: Alfabetización Científica y Tecnológica

Jorge Enrique García Farieta, PhD

Universidad Distrital Francisco José de Caldas
Grupo de Física e Informática, FISINFOR jegarciaf@correo.udistrital.edu.co

Julián Andrés Salamanca Bernal, PhD

Profesor Universidad Distrital Francisco José de Caldas
Facultad de Ciencias y Educación
Grupo de Física e Informática, FISINFOR
jasalamanca@udistrital.edu.co

Diego Julián Rodríguez Patarroyo, PhD

Profesor Universidad Distrital Francisco José de Caldas
Facultad de Ingeniería
Grupo de Investigación Laboratorio de Fuentes Alternas de Energía, LIFAE
djrodriguezp@udistrital.edu.co

García Farieta, Jorge Enrique

Radiación-Materia: GEANT4 hands on : un recurso educativo a profesores investigadores, alfabetización científica y tecnológica / Jorge Enrique García Farieta, Julián Andrés Salamanca Bernal, Diego Julián Rodríguez Patarroyo. -- 1a. ed. -- Bogotá : Universidad Distrital Francisco José de Caldas, 2020.

208 páginas ; 24 cm. -- (Doctorado en Ingeniería).

Contiene referencias bibliográficas.

ISBN 978-958-787-230-9 (impreso) -- 978-958-787-231-6 (digital)

1. Geant4 (Programa para computador) 2. Física - Programas para computador I. Rodríguez Patarroyo, Diego Julián II. Salamanca Bernal, Julián Andrés III. Título IV. Serie

CDD: 005.3 ed. 23

CO-BoBN- a1057505

© Universidad Distrital Francisco José de Caldas

© Doctorado en Ingeniería

© Luis Leonardo Rodríguez Bernal - Víctor Hugo Medina García

ISBN Impreso: 978-958-787-230-9

ISBN Digital: 978-958-787-231-6

Primera edición: Bogotá, octubre de 2020.

Elaborado, diagramado y editado en \LaTeX

Impresión:

Amadgraf Impresores Ltda.

Doctorado en Ingeniería

Carrera 7 # 40B-53

Bogotá

Correo electrónico:

investigacion.doctoradoing@udistrital.edu.co

Todos los derechos reservados. Esta publicación no puede ser reproducida total ni parcialmente o transmitida por un sistema de recuperación de información, en ninguna forma ni por ningún medio, sin el permiso previo del Doctorado en Ingeniería de la Universidad Distrital Francisco José de Caldas.

Hecho el depósito legal.

Impreso y hecho en Colombia Printed
and made in Colombia.

Índice general

Introducción	17
1. Simulación Computacional: paradigmas contemporáneos en la construcción de conocimiento científico	19
1.1. Formación docente-investigativa en nuevas tecnologías	19
1.2. La simulación computacional: un enfoque educativo	21
1.3. Ambientes <i>in-silico</i> de aprendizaje para el trabajo autónomo en la formación de alto nivel	24
2. Interacciones Radiación-Materia y Sistemas de Detección	27
2.1. Interacción Radiación-Materia	30
2.1.1. Interacciones Electromagnéticas con la materia	32
2.1.2. Interacción de partículas cargadas con la materia	36
2.1.3. Interacción de los Neutrones con la materia	44
2.2. Sistemas de Detección	46
3. GEANT4: Revisión general de software	49
3.1. Descripción e historia	50
3.2. Funcionalidad de GEANT4	52
3.3. Desarrollo de aplicaciones	58
3.3.1. Descripción de instalación de software	59
3.4. GEANT4: Generalidades y características	60
3.4.1. El programa principal <code>main()</code>	62
3.4.2. Definición de Geometrías	66
3.4.3. Definición de Materiales	69
3.4.4. Definición de Partículas	75
3.4.5. Procesos físicos	76

ÍNDICE GENERAL

4. Una simulación-ejemplo: GEANT4 hands on!	79
4.1. Contexto de la simulación-ejemplo: Radioterapia con acelerador lineal	80
4.2. Necesidad de la simulación-ejemplo	81
4.3. Modelo de simulación-ejemplo	81
4.4. Codificación de la simulación-ejemplo	82
4.4.1. Geometría y materiales	84
4.4.2. Definición de la fuente de partículas y del evento primario	90
4.4.3. Procesos físicos asociados a las interacciones radiación-materia	91
4.5. Compilación y ejecución de la simulación-ejemplo	93
4.5.1. Función <code>main()</code> : Código principal <code>AcLinearCerebro.cc</code>	93
4.5.2. Modificación del archivo <code>CMakeLists.txt</code> y archivos de configuración .	95
4.5.3. Compilación y ejecución de la simulación-ejemplo	99
4.5.4. Visualización, adquisición de datos y ejecución de la simulación-ejemplo: Macro files (<code>vis_BrainExp.mac</code>)	101
4.6. Resultados de la simulación-ejemplo	122
5. Simulación-ejemplo extendida: RunAction, EventAction, ROOT	125
5.1. ROOT: HistoManager	126
5.2. RunAction: AcLinearRunAction	128
5.3. EventAction: AcLinearEventAction	129
5.4. Función <code>main()</code> de la simulación-ejemplo-extendida	132
5.5. Sensitive Detector: AcLinearConstrucción	133
5.6. Notas de compilación de la simulación-ejemplo-extendida	137
5.7. Resultados de la simulación-ejemplo-extendida	140
5.7.1. Análisis de datos	142
Perspectivas y Conclusiones	149
Apéndice A. Interacción de la radiación electromagnética con la materia	151
A.1. Efecto Fotoeléctrico	151
A.2. Efecto Compton	156
A.3. Producción y aniquilación de pares	162
Apéndice B. Descripción de los principales sistemas de detección de radiación	169
B.1. Detectores de ionización gaseosa	169
B.2. Detectores de centelleo	173
B.3. Detectores de neutrones	174
B.4. Detectores semiconductores	176
B.5. Ejemplo de aplicación de los detectores de partículas	178

ÍNDICE GENERAL

Apéndice C. Licencia de Geant4	183
C.1. Copyright Holders of the Geant4 Collaboration	185
C.1.1. Colaboradores Institucionales	185
C.1.2. Colaboradores Individuales	186
Apéndice D. Comandos Básicos	187
D.1. Comandos internos (bash)	187
D.2. Comandos externos GNU : manipulación de ficheros	188
D.3. Comandos externos GNU : control de procesos	188
D.4. Editores de texto en modo consola	188
D.5. Otros comandos GNU	189
Apéndice E. Instalación de ROOT 6.14.06	191
E.1. Instalación	191
E.2. Ejecución	192
Apéndice F. Instalación de GEANT4	195
F.1. Requisitos de Software	195
F.1.1. Instalación de CLHEP	197
F.2. Instalación de GEANT4 10.4	200
F.3. Cómo ejecutar un ejemplo de GEANT4?	204
Apéndice G. Materiales predefinidos de GEANT4	209
G.1. Materiales puros	209
G.2. Compuestos NIST	212
G.3. Materiales HEP	216
Apéndice H. Programación macro de ROOT para análisis de datos de la simulación-ejemplo-extendida	219
Referencias	225

ÍNDICE GENERAL

Índice de figuras

1.1. Modelo de autogestión de conocimiento en relación a la simulación computacional.	26
2.1. Clasificación de partículas de acuerdo al Modelo Estándar de Partículas.	29
2.2. Interpretación geométrica de la sección eficaz.	32
2.3. Interacción de la radiación electromagnética con la materia.	33
2.4. Regiones de dominio de los mecanismos de absorción.	36
2.5. Pérdida de energía por unidad de recorrido en el plomo para diferentes partículas.	40
2.6. Representación pictórica del proceso de emisión de radiación de Cherenkov. La fuente se mueve a una velocidad v , y el frente de onda se mueve a una velocidad inferior, c/n (velocidad de la luz en el medio de índice de refracción n).	42
2.7. Representación pictórica del proceso de bremsstrahlung producido por un electrón de alta energía desviado en el campo de Coulomb de un núcleo atómico.	43
2.8. Clasificación de detectores de radiación.	48
3.1. Diagrama de clases de <i>GEANT4</i>	54
3.2. Diagrama de procesos de <i>GEANT4</i>	61
3.3. Ejemplos en los que se “crean” materiales puros.	72
3.4. Ejemplos en los que se “crean” materiales compuestos por moléculas.	73
3.5. Ejemplos en los que se “crean” materiales compuestos.	74
3.6. Estructura de la clase <i>G4Particle</i>	76
3.7. Clasificación de procesos físicos en <i>GEANT4</i>	77
4.1. Modelo de simulación (el dibujo no está a escala real). El panel superior izquierdo muestra un esquema de vista lateral de la geometría de la simulación.	82
4.2. Estructura de archivos de la simulación-ejemplo. Los archivos <i>CMakeLists.txt</i> y <i>vis_BrainExp.mac</i> se describen mas adelante.	84

ÍNDICE DE FIGURAS

4.3. Geometría de la simulación-ejemplo.	84
4.4. Configuración geométrica de la fuente (cilindro recto) y colimador (cilindro recto exterior, cilindro recto y cono truncado interiores).	85
4.5. Ubicación axial de la configuración geométrica de la simulación-ejemplo.	86
4.6. Interfaz gráfica de usuario propia de <i>GEANT4</i> . Note que en la parte superior izquierda aparece el nombre del archivo ejecutable de la simulación; para este caso: <i>AcLinealCerebro</i>	102
4.7. <i>/vis/open OGL</i> Note que aparece una pestaña en la parte superior con la leyenda "viewer-0 (OpenGLStoredQt)".	103
4.8. <i>/vis/drawVolume</i> Note que si utiliza la tecla "tabular" para auto-completar la línea de comandos, aparecen ayudas en la parte central de la interfaz.	104
4.9. <i>/vis/viewer/set/viewpointThetaPhi 90 0</i>	105
4.10. <i>/vis/viewer/zomm 4</i>	106
4.11. <i>/vis/scene/add/axes 0 0 0 20 cm</i>	107
4.12. <i>/vis/scene/add/trajectories</i>	108
4.13. <i>/vis/scene/endOfEventAction accumulate</i>	109
4.14. <i>/score/create/boxMesh boxMesh</i>	110
4.15. <i>/score/mesh/boxSize 15.0 15.0 2.4 cm</i>	111
4.16. <i>/score/mesh/nBin 1 1 100</i>	112
4.17. <i>/score/mesh/translate/xyz 0 0 -2.4 cm</i>	113
4.18. <i>/score/mesh/rotate/rotateY 180.0 deg</i>	114
4.19. <i>/score/quantity/energyDeposit EnerDepo</i>	115
4.20. <i>/score/quantity/nOfCollision NOfCollision</i>	116
4.21. <i>/score/drawProjection boxMesh EnerDepo</i>	117
4.22. <i>/score/close</i>	118
4.23. <i>/run/beamOn 500</i> Se visualiza la geometría del detector, que corresponde a la fuente de partículas, el colimador y el volumen que simula el cráneo del paciente. Se evidencian las trazas que siguen las partículas al interactuar con el volumen de la simulación.	119
4.24. Resultado visual de la simulación ejemplo.	122
4.25. Compilación de la simulación en un terminal de Linux.	123
5.1. Interfaz gráfica de usuario de ROOT para visualización de archivos.	141
5.2. Histograma evento a evento de la deposición de energía en el absorbente . .	142
5.3. Deposición acumulada de energía en el absorbente, E , en función del número del bin (panel izquierdo) y de la longitud Z_{abs} (panel derecho). Z_{abs} se encuentra entre 0 y 4.8 cm.	143
5.4. Número acumulado de colisiones en el absorbente, NC , en función del número del bin (panel izquierdo) y longitud (panel derecho). Z_{abs} se encuentra entre 0 y 4.8 cm.	147

ÍNDICE DE FIGURAS

A.1. Curvas Experimentales para el Efecto fotoeléctrico: a) Radiación de diferentes intensidades e igual frecuencia; b) Radiación de diferentes frecuencias e igual intensidad; c) Potencial de frenado en función de la frecuencia para diferentes materiales	152
A.2. Esquema del arreglo experimental para observar el efecto fotoeléctrico.	153
A.3. Representación pictórica del efecto fotoeléctrico.	155
A.4. Variación del coeficiente de absorción mísico para dos materiales distintos en función de la energía característica de Rayos X emitidos en el efecto fotoeléctrico.	156
A.5. Esquema del dispositivo experimental para la observación del efecto Compton.	157
A.6. Corrimiento Compton para diferentes ángulos de dispersión.	157
A.7. Esquema para la interpretación de la colisión inelástica fotón-electrón del efecto Compton.	158
A.8. Diagrama polar de la sección eficaz diferencial en función de θ para el efecto Compton.	160
A.9. Coeficientes de absorción por centímetro en función de la energía de los fotones, por efecto Compton en el Plomo.	161
A.10. Representación esquemática de los valores posibles de la energía de un electrón de acuerdo con la teoría de Dirac.	163
A.11.a) Imagen de cámara de burbujas (Fermilab, Universidad de Birmingham), en los recuadros se observan dos espirales de sentido contrario que representan la producción de un par electrón-positrón derecha, bajo un campo magnético perpendicular al plano de la imagen. b) Imagen de Anderson (<i>Phys. Rev.</i> , 43, 491, 1933) con una cámara de niebla en la que se muestra un positrón de 63 MeV que pasa a través de una lámina de plomo y emerge con una energía de 23 MeV.	164
A.12. Esquema pictórico de la producción de un par electrón-positrón.	165
A.13. Esquema pictórico de la aniquilación de un par electrón-positrón.	165
A.14. Forma de la función de partición de la energía $f(\varepsilon, Z)$ con $\frac{h\nu}{m_0c^2}$ como parámetro. La sección eficaz total de la producción de pares está dada por el área bajo la curva correspondiente, en unidades de $Z(Z+1)\alpha r_0^2$	166
A.15. Coeficiente de absorción total en el plomo, para fotones de alta energía indicando las contribuciones por efecto fotoeléctrico, efecto Compton y producción de pares.	167
B.1. a) Cámara de niebla de Charles Wilson; b) Contador proporcional comercial manufacturado por J.E. Manson CO., Inc. (Vaiana, 2011); c) Contador comercial de Geiger-Müller manufacturado por Detectron Corporation.	169
B.2. Configuración esquemática de un detector de ionización gaseosa.	171

ÍNDICE DE FIGURAS

B.3. Número de iones recolectados en función de la diferencia de potencial en una cámara de ionización gaseosa.	172
B.4. a) Tubo centellador de poliestireno (Laboratorio de radiación gamma y alpha, Gfnun); b) Tubo centellador de <i>NaI</i> (Laboratorio de radiación gamma y alpha, Gfnun) (Cristancho, 2011); c)Diagrama esquemático de un detector de centelleo que comprende un material de centelleo acoplado a un tubo fotomultiplicador.	175
B.5. Diagrama de bandas para: a)un metal, b) un semiconductor, c) un aislante.	177
B.6. Representación pictórica del detector ATLAS (Grillo, 2011).	180
B.7. a) Fotografía del electroimán y del detector interno de ATLAS; b) sección transversal de ATLAS mediante simulación; c) deposición de energía de algunas partículas en los sistemas de detección de ATLAS (Cern, 2011a).	182
E.1. Pantalla de inicio de ROOT.	193
E.2. Pantalla de inicio de ROOT.	193
F.1. Captura de pantalla página oficial de descargas de CLHEP: http://proj-clhep.web.cern.ch/proj-clhep/clhep23.html	198
F.2. Captura de pantalla página oficial de descargas de GEANT4 : https://geant4.web.cern.ch/support/download	201
F.3. Captura de pantalla de la interfaz gráfica de usuario para el ejemplo B1.	206
F.4. Ejecución del comando /run/beamOn 5: Apenas se digita /r aparece una ayuda de los posibles comando a utilizar en la interfaz de GEANT4.	206
F.5. Ejecución del comando /run/beamOn 5: Completando la escritura del comando a utilizar con sus opciones se presiona la tecla “enter”.	207
F.6. Visualización de las trayectorias de los rayos que inciden sobre la geometría de la simulación.	207
F.7. Magnificación de las trayectorias de los rayos y la geometría de la simulación.	208

Índice de tablas

2.1. Principales fuentes de radiación electromagnética.	28
2.2. Interacción de la radiación electromagnética con la materia.	34
2.3. Valores del potencial medio de excitación para algunos elementos. Los valores con * son tomados de resultados experimentales de (ICRP, 2011).	41
B.1. Energía media necesaria para formar un par electrón-ión en diferentes gases (Tomado de (Udías, Lopéz, y Fraile, 2011)).	171

ÍNDICE DE TABLAS

Prólogo

Con el propósito de apoyar la investigación científica, el grupo de Física e Informática (FISINFOR) en su línea de investigación “Alfabetización Científica y Tecnológica orientada a profesores investigadores”, en asocio con el grupo Laboratorio de Investigación en Fuentes Alternas de Energía (LIFAE), en el marco del proyecto de investigación “Simulación CPMD (Car-Pirrinello Molecular Dynamics) del proceso de formación de canales de conducción en la interface $Ti_4O_7-TiO_2$ en dispositivos memristivos basados en TiO_2 ”, adscrito a su línea de investigación “Modelos y simulaciones de nuevos dispositivos para la industria de la ingeniería”, ambos grupos pertenecientes a la Universidad Distrital Francisco José de Caldas (Bogotá-Colombia), presentan este recurso educativo orientado a la formación en investigación de alto nivel de profesores-investigadores en áreas en las que **GEANT4** se ve como un recurso computacional.

Este documento que recoje los resultados de investigación documental y formativa sobre el ejercicio computacional de uso y manejo de software libre a propósito de realizar una simulación computacional guiada en **GEANT4**, para orientar, particularmente, el objetivo del proyecto de investigación sobre la realización de una simulación CPMD en óxidos de titanio en sistemas memristivos, y generalmente, trabajos posteriores de la líneas de investigación, propone entonces estructurar, diseñar y elaborar un texto orientado a profesores-investigadores sobre el uso y manejo de software científico **GEANT4**, donde se consiente un ejercicio creativo de ver a los autores como parte del proceso de alfabetización del lector, análogo al modelo de la Mecánica Cuántica, en el que experimentador hace parte del experimento, y por ello afecta el ejercicio de medición.

Al decir de Freire, la alfabetización se puede entender como:

“El método a través del cual el hombre “se dice” y al hacerlo se reconoce como co-creador de su vida y de su mundo. Es el momento en que el hombre se reconoce como lo que realmente es y se compromete con su humanización”¹.

¹Freire, Paulo. La educación como práctica de la libertad. Siglo xxi, 1978.

PRÓLOGO

lo que direcciona el concepto de Alfabetización Científica y Tecnológica orientada a profesores-investigadores, no como un mero proceso de capacitación, ya que no encuentra en la mecánica de realización de una tarea un discernimiento y crítica frente a ella, sino como un compromiso de emancipación del profesor-investigador. Acercarse a un conocimiento tan complejo como lo es la infraestructura de software *GEANT4* (GEometry ANd Tracking), asiente una decisión por parte del profesor-investigador de hacer gestión de su propio conocimiento, un proceso de liberación, donde el individuo está dispuesto a salir de su zona de confort, a frustrarse, para así alcanzar un auto-conocimiento de sus propios procesos de pensamiento.

El proceso de alfabetización científica y tecnológica en la formación en investigación de alto nivel requiere, no solamente el área de conocimiento en la que se investiga, sino toda una amalgama de áreas que se interrelacionan, dejando al conocimiento disciplinar ser abrazado por la interdisciplinariedad, transdisciplinariedad y multidisciplinariedad.

Pensarse la escritura, la diagramación, los ejemplos y demás componentes didácticos del recurso, permite a los autores proponer una metodológica de realización que asegure un impacto positivo en la gestión de auto-aprendizaje del lector; es decir, la disminución de la curva de aprendizaje del software y ahorro en tiempo y dinero en la investigación científica.

Se destaca entonces la necesidad del establecimiento de un lenguaje común a los autores y al lector; por ello se sintetizan las características de conocimiento intelectual que debe poseer el lector para afrontar este texto: un nivel de inglés básico debido a que la gran mayoría de documentación y códigos de programación se encuentra en ese idioma; tener una formación en ciencias básicas al nivel de ingeniería, ciencias experimentales como física, química o biología, o, matemáticas y estadística; un conocimiento básico de sistema operativo Linux; y, un nivel promedio de programación en C++. Se asiente, inicialmente, un lenguaje quasi-concertado entre los autores y el lector, que posteriormente irá siendo sensibilizando por el lenguaje propio del objeto de estudio: uso y manejo de *GEANT4*.

GEANT4 es una infraestructura de software que permite realizar simulaciones de fenómenos de interacción radiación con la materia. *GEANT4* admite la simulación de detectores y de interacciones de partículas con el detector haciendo uso del método de Montecarlo en sus códigos de programación, los cuales son desarrollados y mantenidos por la Colaboración *GEANT4*. Entre las áreas de aplicación podemos encontrar ciencias de la computación, física de altas energías, física médica, física nuclear, física del espacio e ingeniería, entre otras.

PRÓLOGO

Debido a que el área propia de conocimiento de *GEANT4* es el estudio de los fenómenos de la interacción radiación-materia, el texto elabora en un inicio sobre cuatro procesos de absorción y emisión de radiación (efecto fotoeléctrico, efecto Compton, producción y aniquilación de pares, radiación de frenado o Bremsstrahlung) y sistemas de detección de partículas, lo que contextualiza al lector en la física que tiene en cuenta *GEANT4* para la realización de simulaciones y la detección de partículas.

Posteriormente se presenta una revisión general de *GEANT4*, particularmente sobre sus funcionalidades y códigos de programación, con lo que se sensibiliza al lector para realizar un ejercicio práctico “Hands On!” en base a una simulación diseñada por los autores de un caso de tratamiento médico por uso de aceleradores de partículas o LINAC (simulación-ejemplo), y finalmente, extender la simulación-ejemplo a uso y manejo de requerimientos adicionales de software que pueden ser incluidos en una simulación con las herramientas de *GEANT4*.

La disminución de la curva de aprendizaje de *GEANT4*, mediante este producto de nuevo conocimiento, avecinda su uso en el diseño de detectores sensibles a la radiación, fotónica, simulación de sistemas memristivos; esencialmente, al modelado y simulación de nuevos dispositivos para la industria de la ingeniería.

Se recomienda para el profesor-investigador-lector, una lectura inicial, detenida, de todo el recurso educativo, para sensibilizar al cerebro sobre signos, lenguaje y significación propios de la alfabetización científica y tecnológica orientada a profesores-investigadores en el uso, manejo y desarrollo de simulaciones en *GEANT4*, para después hacer una lectura práctica, una lectura “Hands On!” haciendo uso del computador.

Cabe señalar que al final del recurso, se realiza un ejercicio de análisis de datos para orientar el uso y manejo de otras infraestructuras de software, como ROOT, para un análisis profesional y consistente de datos. Los códigos de programación utilizados producto del diseño y elaboración de este recurso educativo sobre el uso y manejo de *GEANT4* y ROOT quedan a disposición del lector vía [github](#).

“Ser una persona letrada en ciencia y tecnología implica mucho más que ser capaz de leer, entender y escribir acerca de ciencia y tecnología; incluye la habilidad de aplicar conceptos científicos y usar herramientas tecnológicas en la resolución de problemas y la toma de decisiones en su quehacer diario e implica la aplicación de conceptos científicos más allá de las exigencias de la evaluación curricular” (Organización de Naciones Unidas, UNESCO).

PRÓLOGO

Introducción

Hoy día la ciencias computacionales se han convertido en una valiosa herramienta de apoyo a las diferentes disciplinas respecto a los procesos educativos, involucrados directamente por una institución y/o en particular un profesor-investigador, como componente especial de escenarios de aprendizaje. Resulta fundamental reflexionar acerca de la influencia y conveniencia de la implementación de herramientas computacionales como mecanismo facilitador de los procesos de enseñanza-aprendizaje-investigación en la formación de profesores-investigadores, tal como se aborda paulatinamente en este documento.

La disponibilidad de material informativo y formativo en la simulación de fenómenos de las interacciones entre *radiación* y *materia* como recurso educativo en la formación de alto nivel es casi exiguo, prácticamente inexistente. Con el objetivo de satisfacer esta carencia, este libro de resultados de investigación aborda la inclusión de herramientas computacionales de última tecnología, como lo es *GEANT4*, en la formación de profesores-investigadores, mostrando que aspectos inherentes a la didáctica, a la técnica, manejo de software, y a la misma física, pueden ser explorados de manera holística.

Así, la elaboración de recursos educativos orientados a profesores-investigadores (en medio físico y electrónico) asequible para la implementación de una herramienta computacional, como lo es *GEANT4*, en las clases de física a nivel de pregrado y posgrado, facilita los procesos de enseñanza-aprendizaje-investigación en los estudiantes, particularmente en temas de física e ingeniería. Al respecto, los experimentos relacionados con la interacción entre la radiación y la materia poseen un alto grado de abstracción y una baja probabilidad de ser reproducidos en el contexto cotidiano de quien aprende; es por ello que la simulación computacional resulta extremadamente conveniente para el ejercicio de alfabetización en ciencia y tecnología.

El texto se desarrolla en tres partes, donde se cuentan 5 capítulos y un capítulo de conclusiones.

La primera parte corresponde a la formación de profesores-investigadores, su procesos

Introducción

de alfabetización científica y tecnológica, involucrando nuevas tecnologías que medien y faciliten los procesos de enseñanza-aprendizaje-investigación y el contexto de su quehacer, sin dejar de lado la importancia de la *simulación computacional*, guiada por el avance científico, tecnológico y técnico del mundo (Capítulo 1).

La segunda parte abarca los aspectos concernientes a la física relacionada con los fenómenos de interacción radiación-materia, desde su “aparición” en la física, poniendo de manifiesto la evidencia experimental asociada, hasta el estado actual e importancia de los mismos; esto se realiza bajo un enfoque práctico y conceptual, sin pérdida de formalismo, asumiendo conceptos propios de asignaturas como electricidad y magnetismo, física moderna y física cuántica, pero sin entrar en detalles profundos, pues esto desvirtuaría no sólo el objetivo del recurso educativo, sino que involucraría temas no completamente descritos según el estado actual de la física (Capítulo 2).

La tercera y última parte comienza con una revisión general de *GEANT4*, donde se describen sus funcionalidades y características, desde dos puntos de vista, conceptual y computacional, para sensibilizar al lector sobre el lenguaje propio de *GEANT4*, que posteriormente deriva en su uso para el trabajo práctico de implementación de una simulación-ejemplo diseñada por los autores para el público objeto del recurso educativo. En ella se elabora sobre los componentes básicos de una simulación programada en *GEANT4*, su codificación y posterior ejecución mediante comandos bajo la interfaz gráfica de usuario de *GEANT4*. También se realiza un ejercicio práctico, extendiendo la simulación a la utilización de software adicional, con fines de realizar un estudio más detallado de la simulación, haciendo uso de otras utilidades de *GEANT4* y de software independiente, como por ejemplo ROOT (Capítulos 3, 4 y 5).

Al final del texto se presentan unas perspectivas y conclusiones donde se enfatiza en algunas recomendaciones y resultados propios de este documento.

Capítulo 1

Simulación Computacional: paradigmas contemporáneos en la construcción de conocimiento científico

"We can only see a short distance ahead, but we can see much that must be done".
Alan Turing.

1.1. Formación docente-investigativa en nuevas tecnologías

En el campo de la enseñanza, la *simulación computacional* provee un método de obtención, desarrollo y promoción del conocimiento (Kofman, 2000), que puede ser aplicado no sólo en *física*, sino también en diversas ramas de la ciencia y de la tecnología. Las nuevas tecnologías, aplicadas hoy día, involucran gran variedad de teorías didácticas del conocimiento, con el fin de favorecer el desarrollo de las capacidades de pensamiento, hacia un mejor proceso de enseñanza-aprendizaje-investigación; es por esto, que la simulación no sólo se expone como un método de enseñanza, sino también como una metodología, ampliamente utilizada en ciencia, enfocada hacia la resolución de problemas, mediante algoritmos lógicos y consistentes computacionalmente, en particular con las leyes de la física.

Desde hace varios años, la formación de profesores-investigadores ha estimulado su alfabetización en ciencia y tecnología que complementen su formación disciplinar, principalmente en tecnologías de la información y la comunicación; con este propósito,

Simulación Computacional: paradigmas contemporáneos en la construcción de conocimiento científico

se ha introducido la simulación computacional. Las “técnicas” empleadas en el contexto del enseñanza-aprendizaje a la hora de hacer transferencia de conocimiento son de gran importancia, y resultan efectivas sólo si estas se acoplan directamente con los conceptos propios del área de estudio (Guadalupe, 2003); la implementación de nuevas tecnologías en conjunto con métodos didácticos contextualizados, requiere una alfabetización sobre éstas, para que el proceso *enseñanza-aprendizaje* resulte lo más íntegro posible y se lleve a cabo de forma satisfactoria entre las partes involucradas.

En virtud de lo anterior, resulta de vital importancia el manejo y desarrollo de herramientas tecnológicas (software, procesos y aplicaciones), desde un nivel temprano de estudios de pregrado, para que profesionales en formación adquieran conocimientos contemporáneos acerca de su quehacer, ofreciendo también, una educación de calidad y competitiva.

De otra parte, la contextualización de la alfabetización científica y tecnológica implica tener en cuenta, en todos los ámbitos, el *ideal de la educación*, pues éste varía constantemente de acuerdo a las características socio-políticas y económicas de la sociedad y del entorno en particular, donde tienen cabida los procesos de enseñanza-aprendizaje-investigación. De allí que la educación en diferentes épocas y lugares no haya podido dar paso al libre examen de ideas innovadoras, pues los factores ya mencionados no lo permitían; con ello se puede presumir que las sociedades “viven” la conciencia inmediata del valor de sus ideales. Por ejemplo, en el siglo XVIII se creía haber encontrado en la formación de los individuos el ideal que debía realizar el progreso de la humanidad, pero a finales del siglo XIX esto era risible (Bustos, 2002), de manera similar, el *ideal de la educación* que se anuncia también tendrá su lugar en la historia.

Claramente ninguna metodología didáctico-tecnológica puede tener pretensiones de validez general, con tentativas “eternas” de aplicación, y derivar de allí el fin último sobre la alfabetización científica y tecnológica; por tal motivo, lo único que como sociedad en constante evolución se puede hacer, es ir al ritmo de cambio de la ciencia, la economía y demás áreas que influyen en el desarrollo de la autogestión del conocimiento como tal.

La propuesta de hacer alfabetización científica y tecnológica orientada a profesores-investigadores en la aplicación de las nuevas tecnologías, debe proporcionar respuestas concretas y eficientes a los problemas y necesidades que la gestión de conocimiento afronta en la *actualidad*, mas aún en el caso del aprendizaje de fenómenos que requieren grados superiores de abstracción, motivo por el cual muchos de esos fenómenos constituyen retos científicos modernos, como es el caso de las interacciones radiación-materia.

Las herramientas computacionales, desde el punto de vista de las TIC y la metodología *Hands-On* (entendida como “manos a la obra”), son una alternativa en el entendimiento de la diversidad de estilos y ritmos de aprendizaje, considerando además que su incorporación en los proyectos propias de auto gestión del conocimiento proporciona una visión multidimensional de quien aprende.

Las futuras generaciones de profesores-investigadores habrán de desempeñarse como educadores en escenarios complejos, atendiendo demandas que obedecen al avance científico-tecnológico; considerando además que la incidencia de las nuevas tecnologías en la vida de las personas, implica que el profesor-investigador debe conocerlas en varias “formas”, siendo capaz de analizarlas críticamente y seleccionar los recursos tecnológicos para luego integrarlos adecuadamente a su quehacer profesional; pues se debe tener en cuenta que los profesores-investigadores en formación se “enfrentan” a una era de BigData que requiere extraer información valiosa de cientos de miles de millones de datos.

1.2. La simulación computacional: un enfoque educativo

La física, en su rol teórico y experimental, tiene la posibilidad de aprovechar los diferentes sistemas informáticos, en donde el uso del computador resulta ser una potente herramienta de trabajo, pues facilita la realización de cálculos numéricos extensos, de gráficos de alta calidad, y experiencias a través de simuladores, entre otros.

Un gran número de los problemas que se presentan en situaciones de la física y la ingeniería no son simples de resolver, o no pueden ser totalmente entendidos mediante un enfoque tradicional, sea desde el punto de vista teórico o por la imposibilidad de realizar una experiencia de laboratorio; en este último caso se encuentra también el hecho de que no es sencillo contar con las condiciones naturales de una experiencia de laboratorio que requiera parámetros característicos; bajo este punto es que el empleo de la simulación adquiere importancia en el desarrollo de experiencias que contribuyan a la formación en el área de física e ingeniería.

La simulación, según la concibe Ríos (Ríos, 2000) “*consiste básicamente en construir modelos informáticos que describen la parte esencial del comportamiento de un sistema de interés, así como de diseñar y realizar experimentos con el modelo y extraer conclusiones de sus resultados para apoyar la toma de decisiones. Por lo general, se utiliza en el análisis de sistemas tan complejos que no es posible su tratamiento analítico*”.

En términos informáticos, la simulación se basa en reproducir “artificialmente” un fenómeno o las relaciones entrada-salida de un sistema, de otra forma, es un

“experimento” que se realiza empleando el computador, y del cual la información obtenida es comparable con la de un experimento de laboratorio.

En la década de 1940, los ingenieros y físicos atómicos recurrían a técnicas de simulación para calcular el blindaje que debían utilizar para frenar neutrones producidos en una fisión nuclear, para ello se empleaba el reciente método de Montecarlo, en el cual se generan diferentes eventos (interpretados como números aleatorios), que posteriormente eran evaluados estadísticamente conforme a los fenómenos físicos asociados al modelo físico particular. En la actualidad este método sigue siendo utilizado en física de partículas, en general en la investigación científica e industria, y es implementado por numerosas herramientas computacionales.

El uso de la simulación computacional representa una herramienta metodológica beneficiosa; motivo por el que no habría que esperar resultados distintos tras su implementación en el sector educativo, donde la simulación juega un papel fundamental para los procesos de enseñanza-aprendizaje-investigación de las ciencias fácticas. Su uso, enmarcado dentro de una didáctica que tenga en cuenta los procesos de aprendizaje de los estudiantes, debe contemplar un enfoque práctico bajo el esquema *Hands-On*, transige al estudiante internarse en una experiencia llena de procesos o acontecimientos, en los que puede observar cómo ellos cambian al modificar variables o condiciones iniciales y así propender a la construcción de su conocimiento.

Precisamente, los principios involucrados en la una propuesta de *Interactive Engagement* (Hake, 1998), considera algunas funciones derivadas de la inserción de la simulación (ó software) como son:

- *Motivación*: el uso de la computadora y de programas educativos genera en los estudiantes una expectativa, especialmente en aquellos que no han tenido experiencias computacionales, generando una motivación especial para el logro de los objetivos propuestos. Por ello que la motivación en los materiales computarizados es uno de los aspectos principales, transformándose en un motor de aprendizaje, ya que incita a la actividad y al pensamiento. La motivación permite que los estudiantes otorguen mayor tiempo al trabajo de un tema concreto y por lo tanto, se logre mayor aprendizaje.
- *Interacción*: otro de los aspectos que trae como consecuencia la falta de motivación e interés, se debe a que la mayoría de los materiales educacionales no son interactivos, además que el profesor no fomenta la interacción del estudiante con el material (por ejemplo, libros). La introducción de los programas educativos, genera

la interacción entre el estudiante y el material a través del computador, asignando al estudiante un rol más activo en el proceso de aprendizaje, cambiando su rol de espectador por el de un participante activo en el proceso de obtención de conocimientos.

- *Individualización:* los alumnos no presentan las mismas características, no aprenden igual, no tienen los mismos conocimientos previos, no poseen las mismas experiencias, es decir no son iguales, característica que dificulta al docente el logro de las metas educativas. El empleo del software educativo puede solucionar este problema, a través de su uso al permitir generar métodos de enseñanza que individualizan el trabajo del estudiante, adaptando su ritmo de trabajo, siendo útiles en la realización de trabajos complementarios y de reforzamiento.

El estudiante puede controlar su ritmo de aprendizaje, porque los programas permiten que tenga el control sobre el tiempo y los contenidos de aprendizaje, haciendo que el proceso de aprendizaje sea más flexible, eficaz y eficiente.

- *Evaluación como medio de aprendizaje:* a diferencia de los sistemas de evaluación tradicional que están marcados por períodos de tiempo amplios, los programas proporcionan respuestas inmediatas sobre las actividades de aprendizaje, permitiendo que los estudiantes conozcan sus aciertos y errores en el momento que se producen.

Los alumnos pueden ser reforzados inmediatamente cuando una respuesta es correcta, no señalando únicamente que su respuesta es correcta, sino explicándola. Las respuestas incorrectas no solo se identifican, sino que generan nuevas opciones para rectificar las respuestas y determinar por qué la respuesta es incorrecta, generando nuevas secuencias de aprendizaje.

Como es natural, los resultados que son obtenidos de la aplicación de simulaciones en un contexto educativo, dependerá del buen uso que el docente-investigador haga de los mismos, de los objetivos que deseé alcanzar, de la planificación de las actividades a realizar y de las temáticas a abordar durante el curso, teniendo como objetivo primordial que sus estudiantes logren un aprendizaje más significativo.

Gracias al empleo de la simulación se ha incrementado considerablemente el conocimiento científico en las últimas décadas (Hurtado y cols., 2006), obteniendo un poderoso medio para la realización de estudios a través de modelos, actuando de esta

manera como un eslabón entre el sujeto y el objeto, como un transmisor de información desde el fenómeno que se estudia hasta el investigador.

La simulación resume toda la teoría relacionada con un proceso en el cual se sustituyen las situaciones reales por otras creadas artificialmente, de las cuales, quien aprende, adopta habilidades y hábitos que posteriormente puede aplicar en la vida real con igual eficacia.

1.3. Ambientes *in-silico* de aprendizaje para el trabajo autónomo en la formación de alto nivel

En los procesos de aprendizaje para el trabajo autónomo en la formación de alto nivel; es decir, en la investigación científica, las simulaciones numéricas se han convertido en una herramienta central, que son aplicados en las ciencias básicas así como en la ingeniería, economía entre otras disciplinas. Desde la mitad del siglo XX, los primeros modelos numéricos para entender el comportamiento de la radiación dan su nacimiento.

En particular, Enrico Fermi que desarrolló en el laboratorio nacional de los Álamos (Estados Unidos) la primera pila nuclear y junto a ella, el método que denominamos en la actualidad como el método de Montecarlo, se ha extendido para modelar sistemas físicos, económicos, biológicos entre muchos otros, así como es una herramienta didáctica fundamental para comprender la fenomenología de los sistemas estocásticos.

Con el advenimiento de recursos computacionales, otros métodos se han masificado y han sido de gran importancia, de forma particular, en los estudios de la interacción radiación-materia. Este complejo entramado de técnicas, basadas en la discretización del espacio-tiempo, ha sido de ayuda tanto para el desarrollo teórico como para el diseño experimental, presentando reducción de costos en la investigación científica y aportando a la predicción empírica y constatación teórica.

Se pueden clasificar estos métodos numéricos en tres grandes grupos: el primer grupo tiene como objetivo solucionar las ecuaciones diferenciales de campo medio utilizando técnicas como diferencias finitas o elementos finitos; el segundo se basa en modelar el comportamiento cinético colectivo por medio de métodos de primeros principios (estados individuales y relaciones colectivas) que se resumen en las técnicas de Montecarlo o Autómatas Celulares; y el último, un grupo que consiste en las técnicas basadas en Dinámica Molecular donde se estudia el movimiento de cada una de las partes que

conforman el sistema, todos ellos pensados en función de la escala espacio-temporal.

Cuando se disminuye el tamaño del sistema a un nivel microscópico, las ecuaciones de campo medio pierden su precisión y se hace imperativo utilizar métodos computacionales de primeros principios que modelan el sistema a partir de la transición entre estados, métodos que por lo general están divididos en dos grandes grupos: Autómatas Celulares y Montecarlo.

La técnica de Montecarlo es la más utilizada ya que reconoce que la transición entre los estados es de carácter estocástico, lo que implica una necesidad de conocer anticipadamente las características individuales de cada estado, como son la energía, tiempo de vida media, entre muchos otros. En algunos casos es posible caracterizar el sistema por medio de la experimentación, en contraste con bastante otros casos, donde estas características no son posibles de conocer. En estas situaciones es necesario utilizar técnicas de primeros principios que permitan realizar aproximaciones computacionales a la descripción de las características del sistema en concierto (una posibilidad de medición indirecta computacional).

El aprendizaje, uso, manejo y desarrollo de los métodos numéricos, para solución de problemas del ambiente de vida de los humanos, ha permeado el currículo de la mayoría de los programas de educación formal y la formación autónoma de alto nivel de investigadores en ciencias e ingeniería.

En relación al trabajo autónomo, gestión propia de conocimiento, y al carácter experimental de las ciencias e ingeniería, las técnicas numéricas para implementaciones *in-silico*, es decir, “vía computacional”, requieren que el profesor-investigador comprenda la teoría donde se deriva un modelo base para la simulación computacional. Esto permite una relación biunívoca entre simulación y modelo, pues ya que uno es base del otro, la simulación puede realimentar el modelo, ajustando parámetros y aumentando o simplificando sus relaciones; aquí se entiende por “modelar” el establecimiento de características del sistema (como los observables) y sus relaciones.

Los observables permiten hablar en el experimento de la medición, directa o indirecta, que dialoga con la simulación, ya sea por la antesala que realiza la simulación para el diseño de experimentos, o el aporte que puede llegar a hacer en mediciones indirectas donde el experimento no tiene acceso.

La medición indirecta puede ser llevada a cabo *in-silico* si se pueden medir las variables exógenas del sistema. En un proceso de interacción radiación-materia, donde se quiere determinar la energía individual depositada por cada partícula (variable endógena)

que incide sobre la materia (por ejemplo, fotones incidiendo en una célula), es posible conocer el valor total de la radiación incidente (variable exógena), donde se puede conocer la energía promedio por partícula depositada en la materia, que sería un valor de comparación (acción de medir) con este resultado de la energía individual de cada partícula, exclusivo de la simulación computacional.

En la auto-gestión de conocimiento, entendemos las relaciones entre simulación computacional, modelo teórico y experimentación biunívocas, con centro en el profesor-investigador quien se presenta como el interlocutor entre un problema y su solución (ver Fig. 1.1).

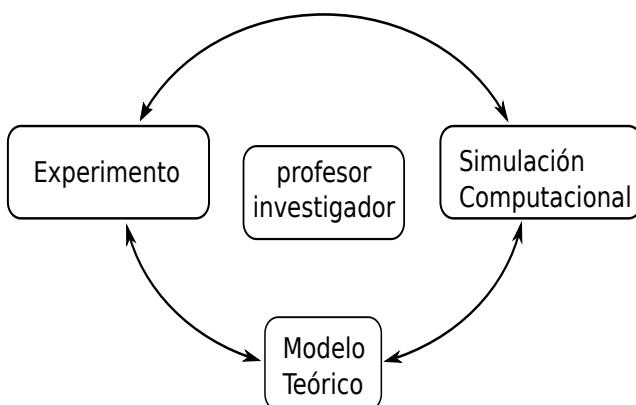


Figura 1.1: Modelo de autogestión de conocimiento en relación a la simulación computacional.

El profesor-investigador en su diálogo con esta triada reconoce la simulación computacional como recurso para la producción de nuevo conocimiento, dado que aporta una mirada de los fenómenos sistémicos más amplia y re-contextualiza conceptos disciplinares, inter-disciplinares, trans-disciplinares y multi-disciplinares, retos propios de la formación de alto nivel; es en este lugar, como interlocutor que aprende diferentes lenguajes, el de la experimentación, el de la simulación y el de la teoría, asiente su actuación sensible en aumentar su capacidad de abstracción intelectual, para así desembocar en procesos creativos de investigación, desarrollo e innovación.

Capítulo 2

Interacciones Radiación-Materia y Sistemas de Detección

"On ne saurait croire combien un mot bien choisi peut économiser de pensée".
Henri Poincaré.

Definir con exactitud “radiación” resulta ser una tarea ardua y nada fácil, ya que su significado tiene raíces en el término “energía”, que al igual que “espacio” y “tiempo” son conceptos, desde la lógica, “indefinibles”, que constituyen la base de toda definición ulterior en física (Wittgenstein, 1968). La palabra “radiación”, involucrada en diversos tópicos (pudiéndose hablar de radiación electromagnética, térmica, solar, nuclear, cósmica, etc), es un término que, históricamente, se ha utilizado para caracterizar la propagación de energía en forma de ondas electromagnéticas o de partículas subatómicas a través del vacío o de un medio material.

Algunos fenómenos relacionados con la radiación pueden describirse mediante la teoría electromagnética clásica; sin embargo, la única explicación general satisfactoria se encuentra en la teoría cuántica. En 1905 al explicar el efecto fotoeléctrico, Albert Einstein sugirió que la radiación electromagnética presenta un comportamiento cuántico,¹ esto es, se comporta como “diminutos proyectiles” llamados fotones y no como ondas. La naturaleza cuántica de la radiación había sido postulada antes de la publicación del artículo de Einstein (“*On a Heuristic Viewpoint Concerning the Production and Transformation of Light*”).

¹La discusión sobre la naturaleza de la luz se remonta a la antigüedad. En el siglo XVII, Isaac Newton se inclinó por una interpretación corpuscular de la luz, mientras que sus contemporáneos, Huygens y Hooke, apoyaron la hipótesis ondulatoria. Experimentos de interferencia, como el realizado por Young, confirmaron el modelo ondulatorio de la luz, el cual prevaleció hasta el surgimiento de la explicación cuántica.

Ya en 1900 Max Planck había empleado una hipótesis cuántica y el formalismo de la mecánica estadística para derivar la ley fundamental de la radiación. La expresión matemática de esta *Ley de Distribución de Planck*, Eq. (2.1), relaciona la intensidad de la energía radiada por un cuerpo en una longitud de onda determinada con la temperatura del mismo:

$$I(\nu, T) = \frac{2h\nu^3}{c^2} \frac{1}{e^{\frac{h\nu}{kT}} - 1} \quad (2.1)$$

Existen diversas fuentes de radiación dependiendo de la naturaleza de la misma; así, por ejemplo, la radiación electromagnética, a la que se asocian fotones, puede ser originada por transiciones electrónicas en átomos y/o moléculas, según sea la energía radiada, i.e. la energía transportada por los fotones $E = nh\nu$ (ver Tabla 2.1).

Ondas de Radio	Microondas	Infrarrojo	Visible y UV	Raios X	Raios γ
Osciladores eléctricos.	Oscilación de electrones en cavidades. Klystron y magnetrones.	Transiciones electrónicas en moléculas.	Transiciones electrónicas en átomos.	Cargas eléctricas frenadas bruscamente (Bremsstrahlung). Transiciones electrónicas profundas.	Transiciones en el núcleo del átomo o radiación emitida por núcleos atómicos.

Tabla 2.1: Principales fuentes de radiación electromagnética.

De otra parte, del descubrimiento de Henri Becquerel y de los trabajos de Pierre y Marie Curie, Ernest Rutherford y colaboradores, acerca de la radiactividad y el decaimiento de núcleos radiactivos, se estableció la existencia de tres tipos distintos de radiación, nombrados por Rutherford como: α , β y γ . En términos de partículas subatómicas se sabe que la radiación α corresponde a dos protones “mas” dos neutrones (núcleo de helio), la radiación β a electrones, y la γ a fotones de alta energía (con frecuencias superiores a $10^{19} Hz$).

De trabajos posteriores sobre radiactividad y la reciente teoría cuántica (teórica y experimentalmente) se vislumbra la existencia, hasta 1932,² de seis partículas “elementales” (fotón, electrón, protón, neutrón, positrón y neutrino (teorizado)). Sin embargo la

²Año en que James Chadwick descubre el neutrón y que junto con los trabajos de Dimitri Ivanenko, sobre este mismo como partículas de espín 1/2, se establece que el núcleo atómico está constituido por protones y neutrones.

lámina de material sin interacciones es prácticamente cero. La interacción de partículas eléctricamente neutras (neutrones, fotones, etc.) con la materia ocurre de diferentes formas, sin embargo existe una probabilidad, distinta de cero (Vallina, 2005), de que un neutrón o un rayo γ pueda atravesar cualquier espesor de material sin experimentar ninguna interacción; como resultado, no se puede definir un alcance finito para neutrones y rayos γ .

Para expresar la probabilidad de interacción entre las partículas con un material, esto es con un blanco, se utiliza el concepto de *sección eficaz* (cross-section), la cual se puede calcular si la forma básica de la interacción entre las partículas se conoce. Cuando un haz de partículas incide sobre una lámina de una sustancia determinada, la sección eficaz σ es una medida hipotética de la zona alrededor de las partículas de destino de la sustancia (por lo general sus átomos) que representa una superficie. Si una partícula del rayo atraviesa esta superficie, habrá algún tipo de interacción.

La sección eficaz se define formalmente de la siguiente manera: dada la posibilidad de que un haz de partículas incida sobre un “objetivo” como se muestra en la Fig. 2.2, y suponiendo que el haz es mucho más “amplio” que el objetivo, y que las partículas del haz están distribuidas uniformemente, se puede hablar de un flujo de partículas incidentes F por unidad de área y de tiempo. Ahora, el número de partículas dispersadas en el ángulo sólido $d\Omega$ por unidad de tiempo, fluctúa para diferentes períodos debido a la aleatoriedad de los parámetros de impacto. Sin embargo, si se tiene una medida de varios períodos de tiempo, este número tenderá hacia el valor $dN_s/d\Omega$, donde N_s es el promedio de partículas dispersas por unidad de tiempo. Operacionalmente, la sección eficaz diferencial se define como (2.2).

$$\frac{d\sigma}{d\Omega}(E, \Omega) = \frac{1}{F} \frac{dN_s}{d\Omega} \quad (2.2)$$

En general, el valor de $d\sigma/d\Omega$ variará con la energía de la interacción y el ángulo en que se dispersan las partículas. La sección eficaz total de cualquier dispersión, para una energía E , se define como la integral de $d\sigma/d\Omega$ sobre todo ángulo sólido (2.3).

$$\sigma(E) = \int d\Omega \frac{d\sigma}{d\Omega} \quad (2.3)$$

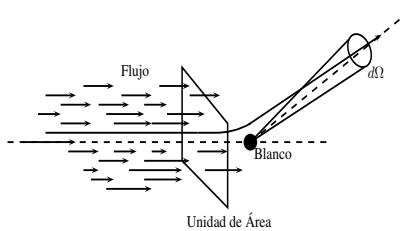


Figura 2.2: Interpretación geométrica de la sección eficaz.

2.1.1. Interacciones Electromagnéticas con la materia

Las diferentes regiones del espectro electromagnético difieren en sus efectos e interacciones con la materia (ver Fig. 2.3 y Tabla 2.2). Cuando un haz de fotones penetra en un medio material existen dos mecanismos que contribuyen a su atenuación: la difusión (en la que un fotón es desviado de la dirección del haz, sin depositar energía en el medio), y la absorción (en la que existe una conversión de la energía del fotón, total o parcial, en energía cinética de partículas cargadas, ionizantes y fácilmente absorbibles). Al incidir o penetrar en la materia, los fotones sufren choques efectivos mucho menos frecuentes que las partículas cargadas eléctricamente, existiendo entonces un camino libre medio recorrido por el fotón antes de ser absorbido o dispersado, este valor corresponde a la distancia media que alcanza antes de desaparecer por algunos de los mecanismos mencionados.

La radiación electromagnética que interesa, desde la perspectiva de su interacción con la materia, es la radiación X y γ ; en ambos casos los fotones asociados tienen longitudes de onda menores que 1 Å (Eduardo Héctor Scarnichia, 2000), aunque este valor en realidad es relativo, la única forma de interpretar adecuadamente la interacción de la radiación electromagnética con la materia es a través del comportamiento corpuscular. La atenuación de la radiación electromagnética difiere de la atenuación de las partículas cargadas en que, en éstas últimas, los alcances están definidos por la energía y el absorbente, mientras que la radiación electromagnética sólo puede ser reducida en intensidad aumentando el espesor absorbente.

De los mecanismos de interacción, solo la absorción implica una transferencia de energía; el mecanismo de absorción tiene lugar a través de alguno de los tres procesos siguientes, de acuerdo con la energía de los fotones incidentes ⁵:

⁵Una descripción detallada de estos procesos de interacción se encuentra en el Apéndice A.

- El haz incidente es colimado, o sea que las trayectorias de los fotones son paralelas y normales a la superficie absorbente.
- El absorbente es “delgado”.⁷

Al igual que en otros procesos estadísticos, como la desintegración radiactiva, se puede definir un espesor medio o capa semirreductora $x_{1/2}$, como el ancho del absorbente que los fotones deben recorrer, para reducir la intensidad a la mitad de su valor inicial, de acuerdo con las Eqs.. (2.6) y (2.7).

$$I(x_{1/2}) = \frac{I_0}{2} = I_0 e^{-\mu x_{1/2}} \quad (2.6)$$

$$x_{1/2} = \frac{\ln 2}{\mu} \approx \frac{0.693}{\mu} \quad (2.7)$$

En la práctica el coeficiente de absorción μ varía con el estado de agregación del absorbente (estado sólido, líquido, gaseoso, plasmático o condensado); para evitar esta dificultad, la Eq. 2.7 se escribe en términos de la densidad del absorbente ρ .

$$\frac{\mu}{\rho} = \frac{\ln 2}{\rho x_{1/2}} \quad (2.8)$$

El cociente μ/ρ , denominado coeficiente de absorción másico, varía muy lentamente con Z y por consiguiente, con el estado de agregación del absorbente; de la Eq. 2.8 se infiere que cuanto mayor es ρ menor debe ser el espesor necesario para producir la absorción de una radiación, es por ello que, como absorbente, suele usarse materiales pesados, por ejemplo plomo.

Cada uno de los tres mecanismos de absorción, antes mencionados, puede caracterizarse por un coeficiente de absorción o por una sección eficaz; en general se pueden establecer intervalos aproximados de energías, de modo que en cada uno de ellos, unos procesos dominen sobre otros: a energías menores a 1 MeV, el proceso dominante es el efecto fotoeléctrico; a energías del orden de 1 MeV domina el efecto Compton; a energías superiores a 1 MeV se hace dominante la producción de pares.

La figura 2.4 ilustra un esquema de regiones de dominio de los mecanismo de absorción de energía, para determinada energía de radiación electromagnética incidente, en función del número atómico del absorbente (Z).

⁷Se define un absorbente delgado si la probabilidad de que un fotón sufra una dispersión Compton doble es menor del 1 %.

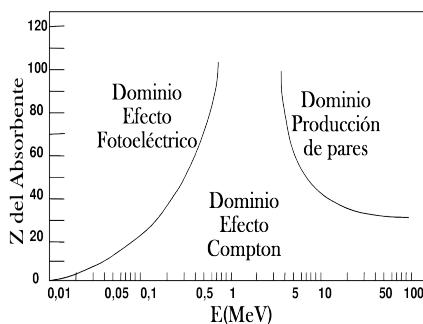


Figura 2.4: Regiones de dominio de los mecanismos de absorción.

Se sintetiza que el efecto fotoeléctrico es relevante a medida que aumenta Z , mientras que si la energía del fotón está en el intervalo en el que se lleva a cabo la producción de pares, la contribución de este proceso también aumenta al incrementar Z ; es decir, cuanto más pesado sea el material las contribuciones del efecto fotoeléctrico y la producción de pares son mayores a las del efecto Compton.

2.1.2. Interacción de partículas cargadas con la materia

Las partículas cargadas interactúan con los núcleos o los electrones orbitales del medio material mediante colisiones (choques) (Landau y Lifshitz, 1992)⁸

Estas colisiones , tanto en el caso relativista y en su aproximación clásica, se pueden clasificar como: colisiones elásticas, cuando la energía cinética, la energía en reposo y la masa del sistema se conservan; colisiones inelásticas, cuando la energía cinética disminuye, y la energía en reposo y la masa aumentan, por ejemplo, en estados excitados del núcleo o del átomo.

Dado “el vacío” relativo existente en el átomo (de acuerdo al modelo semi-clásico planteado por Rutherford y cuya evidencia se obtuvo con el experimento de dispersión de partículas α), las colisiones mecánicas por choque directo entre la partícula y los electrones o núcleos, son muy improbables; en realidad, el proceso predominante es la

⁸Dado que la “medida” de la radiación es sólo posible mediante los fenómenos de interacción, vale distinguir las partículas cargadas en dos tipos: pesadas, $m'_0 > m_0$, (μ^- , π^+ , α , p etc.) y ligeras (e^- , e^+). Tanto las partículas cargadas pesadas como las ligeras, interactúan con los electrones orbitales, como con los núcleos o átomos como un todo; en este último caso la energía de los electrones es menor a la energía de excitación, de igual manera para partículas cargadas de bajas velocidades.

colisión Coulombiana.

En general, el paso de partículas cargadas a través de la materia se caracteriza por: 1) una pérdida de energía de la partícula, y 2) una desviación de la dirección incidente de las partículas. Estos efectos son principalmente el resultado de distintos mecanismos:

- **Colisiones elásticas con electrones atómicos:** la partícula incidente es desviada por electrones atómicos del material con el cual interactúa, siendo en este caso la energía transmitida menor que el potencial de excitación más bajo de los electrones, suponiendo que la interacción se lleva a cabo con el átomo como una única partícula. Este tipo de interacción es importante sólo en el caso de electrones de baja energía (por debajo de los 100eV).
- **Colisiones inelásticas con electrones atómicos:** la partícula incidente pierde energía ionizando y excitando los electrones atómicos. Un primer caso es la producción de un par iónico (ión positivo e íon negativo), en el segundo caso, el electrón excitado vuelve al estado fundamental emitiendo un fotón de energía equivalente a la de excitación.
- **Colisiones elásticas con núcleos:** la partícula es desviada de su trayectoria por efectos de la interacción electrostática con las cargas del núcleo, perdiendo sólo una fracción de su energía cinética inicial.
- **Colisiones inelásticas con núcleos:** la partícula pierde energía al ser acelerada o frenada por interacción electrostática con el núcleo, esta energía se manifiesta como radiación electromagnética, y se denomina radiación de frenado o “Bremsstrahlung”.⁹

Bajo dichos mecanismos se modela la forma en que interactúa una partícula cargada cuando se desplaza en un medio material. La probabilidad de que interactúe mediante uno u otro proceso es determinada por la energía de la partícula incidente y el medio absorbente. Del estudio de la interacción de partículas cargadas discurren tres parámetros principales de suma importancia que son: alcance, poder de frenado y relación alcance-energía (Beiser, 2003).

Interacciones Coulombianas

Al considerar una partícula cargada que viaja a través de un material específico, e interactúa con un átomo del material, la partícula podrá interactuar bien sea con

⁹Otros procesos incluyen la emisión de radiación de Cherenkov.

los electrones atómicos o con el núcleo. Debido a que el radio nuclear se estima del orden de 10^{-14} m y el radio atómico de 10^{-10} m , se tiene que el cociente entre el número de interacciones con electrones y el número de interacciones con el núcleo es de $\sim 10^8$ (Tsoulfanidis, 1995); este argumento indica que las colisiones con los electrones atómicos son más “importantes” que con los núcleos, por lo que las colisiones nucleares no serán consideradas en este aparte.

Las características de los procesos de interacción de las partículas cargadas con la materia hacen que la pérdida de energía de la partícula a lo largo del recorrido se produzca de forma prácticamente continua (Vallina, 2005); para describir esta pérdida de energía se define el poder de frenado, como la energía perdida por unidad de longitud de su recorrido dE/dx (Leo, 1994).

Si en un instante determinado una partícula con carga Ze se encuentra a una distancia r de un electrón de energía E , la interacción se puede describir, en aproximación clásica, mediante la ley de Coulomb $\vec{F} = k \frac{Ze^2}{r^2} \hat{r}$. La acción de esta fuerza sobre el electrón, para un período de tiempo, produce la transferencia de energía de la partícula cargada en movimiento al electrón ligado, y puesto que un electrón atómico está en un estado cuántico, el resultado del paso de las partículas cargadas puede ser la ionización o excitación del átomo, mediante colisiones inelásticas.

La ionización ocurre cuando el electrón adquiere energía suficiente para abandonar el átomo y convertirse en una partícula libre, con energía cinética igual a la diferencia entre la energía neta adquirida y el potencial de ionización; el electrón libre se comporta como cualquier otra partícula cargada en movimiento, pudiendo ionizar otros átomos si su energía es suficiente para ello; los electrones rápidos, producidos por colisiones ionizantes, reciben el nombre de rayos δ . De otra parte la excitación se produce cuando el electrón atómico adquiere suficiente energía para pasar a un estado vacío en otro orbital de mayor energía, dejando así un átomo excitado.

Un análisis mas detallado de la máxima energía cinética que se puede transferir a un electrón, muestra que esta depende de la masa m'_0 y del momento lineal p de la partícula cargada. El momento lineal de esta última es $p = \gamma m'_0 \beta c$, con γ el factor de Lorentz y $c\beta = v$ su velocidad; considerando principios de conservación de la energía y del momento lineal se obtiene la ecuación (2.9), que describe la máxima energía transferida a un electrón.

$$K_{máx} = \frac{2m_0c^2\beta^2\gamma^2}{1 + 2\gamma m_0/m'_0 + (m_0/m'_0)^2} = \frac{2m_0p^2}{m_0'^2 + m_0^2 + 2m_0E/c^2} \quad (2.9)$$

Para bajas energías, $2\gamma/m_0 \ll 1$ (Grupen y Shwartz, 2008), y bajo el supuesto de que las partículas incidentes son más pesadas que los electrones ($m'_0 \gg m_0$) la Eq. (2.9) se puede aproximar a (2.10).

$$K_{máx} \approx 2m_0c^2\beta^2\gamma^2 \quad (2.10)$$

Para partículas relativistas, $K \approx E$, y $E \approx pc$, la máxima energía transferible estará dada por (2.11), sin embargo si la partícula incidente es un electrón estas aproximaciones dejan de ser válidas, en tal caso la energía transferible estará dada por (2.12).

$$K_{máx} \approx \frac{E^2}{E + \frac{m'_0c^2}{2m_0}} \quad (2.11)$$

$$K_{máx} \approx \frac{p^2}{m_0 + E^2/c^2} = \frac{E^2 - m_0^2c^4}{E + m_0c^2} = E - m_0c^2 \quad (2.12)$$

Pérdida por ionización y excitación

El poder de frenado, dE/dx , para partículas cargadas que se mueven a través de la materia interactuando con electrones atómicos, puede ser entendido mediante la fórmula de Bethe-Bloch, de la que se obtiene la pérdida media de energía por ionización o excitación.

En los cálculos de transferencia de energía se suelen parametrizar las ecuaciones en términos del momento lineal en vez del parámetro de impacto (entendido como la distancia entre el vector velocidad de la partícula incidente y la componente paralela a la misma, del centro de fuerzas, en un tiempo definido), esto se justifica ya que la transferencia de momento es una cantidad mensurable, mientras que el parámetro de impacto no lo es, por su carácter estadístico.

Suponiendo que los átomos y electrones atómicos se comportan de manera independiente, se tiene que la perdida de energía de una partícula en MeV/m, está dada por (2.13) para partículas pesadas (p, d, t, α , etc.), por (2.14) para electrones, y por (2.15) para positrones (Grupen y Shwartz, 2008), como se ilustra en la Fig. 2.5. En todos los casos N_A es el número de Avogadro, β el factor de Lorentz, m_0 la masa en reposo del electrón y r_0 el radio clásico de éste último.

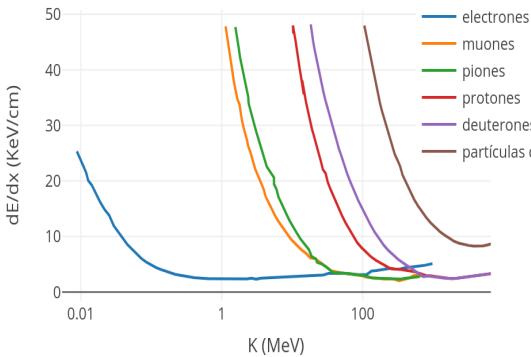


Figura 2.5: Pérdida de energía por unidad de recorrido en el plomo para diferentes partículas.

$$\frac{dE}{dx} = 4\pi r_0^2 z^2 \frac{m_0 c^2}{\beta^2} N_A Z \left[\ln \left(\frac{2m_0 c^2}{I} \beta^2 \gamma^2 \right) - \beta^2 \right] \quad (2.13)$$

$$\frac{dE}{dx} = 4\pi r_0^2 \frac{m_0 c^2}{\beta^2} N_A Z \left[\ln \left(\frac{\beta \gamma \sqrt{\gamma-1}}{I} m_0 c^2 \right) + \frac{1}{2\gamma^2} \left(\frac{(\gamma-1)^2}{8} + 1 - (\gamma^2 + 2\gamma - 1) \ln 2 \right) \right] \quad (2.14)$$

$$\frac{dE}{dx} = 4\pi r_0^2 \frac{m_0 c^2}{\beta^2} N_A Z \left[\ln \left(\frac{\beta \gamma \sqrt{\gamma-1}}{\sqrt{2} I} m_0 c^2 \right) - \frac{\beta^2}{24} \left(23 + \frac{14}{\gamma+1} + \frac{10}{(\gamma+1)^2} + \frac{4}{(\gamma+1)^3} \right) \right] \quad (2.15)$$

El potencial medio de excitación I , es un parámetro de la fórmula de Bethe-Bloch que representa el promedio de la frecuencia orbital $\bar{\nu}$ de los electrones; en la Tablas 2.3 se muestra el valor de I para algunos elementos.

De las ecuaciones (2.13), (2.14) y (2.15) cabe señalar que el poder de frenado:

- Es independiente de la masa de la partícula incidente.
- Es proporcional al cuadrado de la carga de la partícula, $z^2 e^2$.
- Depende de la velocidad v de la partícula.
- Es proporcional a la densidad del material.

Elemento	$I(\text{eV})$	Elemento	$I(\text{eV})$
H	20,4	Fe	281*
He	38,5	Ni	303*
Li	57,2	Cu	321*
Be	65,2	Ge	280,6
B	70,3	Zr	380,9
C	73,8	I	491
N	97,8	Cs	488
O	115,7	Ag	469*
Na	149	Au	771*
Al	160*	Pb	818,8
Si	174,5	U	839*

Tabla 2.3: Valores del potencial medio de excitación para algunos elementos. Los valores con * son tomados de resultados experimentales de (ICRP, 2011).

Emisión y pérdida de energía por radiación de Cherenkov

La radiación de Cherenkov se produce cuando una partícula cargada en un medio dieléctrico, se mueve más rápido que la velocidad de la luz en ese medio. Esta velocidad está dada por $\beta c = v > c/n$, donde n es el índice de refracción del medio. Al atravesar la materia, las partículas cargadas polarizan las moléculas del material, las cuales, al regresar a su estado fundamental, emiten este tipo de radiación electromagnética.

La “onda de choque” electromagnética de la radiación de Cherenkov, se produce de manera similar a la onda de choque sónica que genera un avión cuando éste supera la barrera del sonido (ver Fig. 2.6); el frente de onda coherente formado es cónico y se emite en un ángulo $\cos \theta = \frac{1}{\beta n}$ respecto a la trayectoria de la partícula¹⁰.

El poder de frenado para la radiación de Cherenkov, en una frecuencia dada, se calcula mediante la fórmula de Frank-Tamm (2.16), siendo la permeabilidad y el índice de refracción del medio funciones dependientes de la frecuencia ω .

$$\frac{dE}{dx} \text{ cherenkov} = \frac{q^2}{4\pi} \int_{v>c/n(\omega)} \omega \mu(\omega) \left(1 - \frac{c^2}{v^2 n^2(\omega)} \right) d\omega \quad (2.16)$$

¹⁰La dependencia del ángulo de emisión de radiación de Cherenkov con la velocidad de las partículas, permitió el desarrollo de los contadores Cherenkov, mediante los cuales se pueden detectar partículas de alta velocidad y determinar su velocidad.

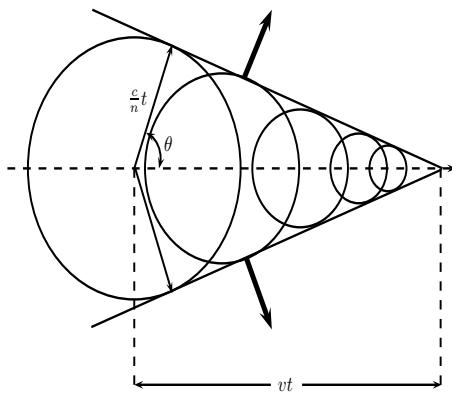


Figura 2.6: Representación pictórica del proceso de emisión de radiación de Cherenkov. La fuente se mueve a una velocidad v , y el frente de onda se mueve a una velocidad inferior, c/n (velocidad de la luz en el medio de índice de refracción n).

La perdida de energía por efecto Cherenkov se convierte en un término adicional en la ecuación de Bethe-Bloch Eq. (2.13), la cual toma mayor relevancia cuando se tienen en cuenta partículas moviéndose a velocidades relativistas. Cabe aclarar que la contribución de este nuevo término a la ecuación de Bethe-Bloch es pequeña en relación a la pérdida de energía debida a colisiones atómicas; por ejemplo, para materiales pesados se tiene que $(\frac{dE}{dx}) \simeq 10^{-3}$ MeV/cm, es cuatro órdenes de magnitud menor respecto a la pérdida por colisión en gases como el H_2 , $(\frac{dE}{dx}) \simeq 0,1$ MeV/cm (Leo, 1994).

Emisión y pérdida de energía por Bremsstrahlung

La radiación de frenado o bremsstrahlung se produce cuando un haz de partículas cargadas, que atraviesa un material, decelera por interacción con el campo de Coulomb de otra partícula cargada del material (Tavernier, 2010), por ejemplo, un núcleo atómico; tras la interacción, las partículas incidentes son desviadas y una fracción de su energía cinética se emite en forma de radiación electromagnética (ver Fig. 2.7).

La pérdida de energía por unidad de recorrido debida a bremsstrahlung puede ser descrita mediante (2.17) (Grupen y Shwartz, 2008) siendo Z el número atómico del medio y z el número de carga de la partícula incidente; en el caso particular de que las partículas incidentes sean electrones, con $E \gg \frac{m_ec^2}{\alpha Z^{1/3}}$, se obtiene consecuentemente (2.18).

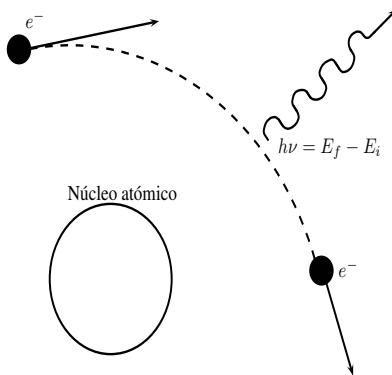


Figura 2.7: Representación pictórica del proceso de bremsstrahlung producido por un electrón de alta energía desviado en el campo de Coulomb de un núcleo atómico.

$$\frac{dE}{dx} \approx 4\alpha N_A \frac{Z^2}{A} z^2 \left(\frac{1}{4\pi\varepsilon} \frac{e^2}{mc^2} \right)^2 E \ln \frac{183}{Z^{1/3}} \quad (2.17)$$

$$\frac{dE}{dx} \approx 4\alpha N_A \frac{Z^2}{A} r_0^2 E \ln \frac{183}{Z^{1/3}} \quad (2.18)$$

En relación con (2.13), la pérdida de energía por radiación de frenado es proporcional a la energía de la partícula deflectora e inversamente proporcional al cuadrado de la masa de las partículas incidentes.

Para un haz de electrones ($z = 1, m = m_0$), las pérdidas de energía son considerables, debido al valor de m_0 . La Eq. (2.18) se puede reescribir como (2.19), la cual representa la pérdida media de energía de un electrón en un espesor dx del material para una longitud de radiación¹¹ X_0 ; un valor estimado de X_0 se halla mediante (2.20).

$$-\frac{dE}{dx} = \frac{E}{X_0} \quad (2.19)$$

$$X_0 = \frac{A}{4\alpha N_A Z^2 r_0^2 \ln(183Z^{-1/3})} \quad (2.20)$$

$$E = E_0 e^{-x/X_0} \quad (2.21)$$

¹¹La longitud de radiación X_0 es una característica propia del material, que se relaciona con la pérdida de energía en interacciones electromagnéticas.

Integrando la Eq. (2.19) se obtiene Eq. (2.21), la cual indica que electrones enérgicos irradian la mayor parte de su energía dentro de varias longitudes de radiación del material. Este comportamiento particular resulta importante en el diseño de detectores de electrones (Das, 2005).

2.1.3. Interacción de los Neutrones con la materia

Puesto que los neutrones son eléctricamente neutros no producen directamente ionización ni bremsstrahlung, sino que las interacciones se producen por colisiones directas mediante fuerzas nucleares de corto alcance. La interacción neutrónica, a diferencia de las partículas cargadas y de la radiación γ y X , se produce con los núcleos del medio absorbente y no con los electrones (Burcham, 1979), de allí que el proceso predominante sea la producción de reacciones nucleares, esto es de dispersión y de absorción; tanto en los procesos de absorción como de dispersión, se producen partículas cargadas y/o fotones, que sí pueden producir ionización en el material, razón por la cual las interacciones de neutrones con la materia se clasifican dentro de las radiaciones indirectamente ionizantes.

Los neutrones se pueden clasificar (Daoud, s.f.) como:

- **Térmicos:** aquellos que están en equilibrio termodinámico con el medio que los circunda, no habiendo en promedio, intercambio neto de energía entre neutrones y átomos; tienen una energía promedio de 0.025eV , y su velocidad promedio es comparable con las velocidades térmicas de las moléculas (Campbell y Ross, 1995).
- **Intermedios:** tienen energías desde 0.5 eV hasta aproximadamente 10^4 eV .
- **Rápidos:** aquellos cuyas energías están comprendidas entre 10^4 eV y 10^7 eV .
- **Relativistas:** aquellos cuyas energías cinéticas superan los 10 MeV , esta energía es una fracción apreciable de la energía total, por lo que se deben aplicar correcciones relativistas en el tratamiento de los mecanismos de interacción.

Para que se produzca una interacción con el núcleo, los neutrones deben estar a una distancia de aproximadamente 10^{-3}cm de este (Leo, 1994), sin embargo, cuando el neutrón no interactúa puede experimentar también una variedad de procesos nucleares en función de su energía. La observación experimental de la interacción del neutrón con los núcleos del absorbente, ha dado lugar para separarlos en procesos de dispersión, cuando el neutrón es desviado de su trayectoria, y de absorción, cuando es absorbido por el núcleo, apareciendo posteriormente otras partículas.

- **Dispersión elástica:** el neutrón interactúa con el núcleo, y después de la reacción aparecen un neutrón y un núcleo de igual naturaleza y en el mismo estado energético que el núcleo inicial, (i.e. ${}_Z^A X + n \rightarrow {}_Z^A X + n$); tras la interacción, la energía cinética de las partículas se conserva, este es el principal mecanismo de pérdida de energía para neutrones.
- **Dispersión inelástica:** el núcleo, resultado de la interacción, es idéntico al que sufre la colisión, sin embargo se encuentra en un estado excitado (i.e. ${}_Z^A X + n \rightarrow {}_Z^A X^* + n'$), normalmente este núcleo emite rayos γ para alcanzar su estado fundamental (${}_Z^A X^* \rightarrow {}_Z^A X^* + \gamma$). La energía cinética total después de la interacción resulta ser menor que la energía cinética total inicial, dicha diferencia entre energías es igual a la energía de excitación del núcleo producido.
- **Captura radiactiva:** el neutrón incidente es capturado por el núcleo y, de modo similar al caso anterior, la excitación del núcleo produce la emisión de rayos γ (${}_Z^A X + n \rightarrow {}_Z^{A+1} X + \gamma$); el núcleo resultante puede ser radiactivo o no.
- **Otras reacciones nucleares (Emisión de partículas cargadas):** el núcleo que ha absorbido al neutrón puede desintegrarse emitiendo una partícula cargada; estos procesos ocurren generalmente en la región de eV a keV (Leo, 1994).
- **Fisión:** el neutrón es capturado por un núcleo pesado, quedando este último inestable, tras lo cual se divide en dos núcleos cuyas masas son aproximadamente la mitad de la masa inicial, el excedente de masa se manifiesta como energía libre. Esta reacción es la principal fuente de energía nuclear con fines prácticos.
- **Reacciones tipo cascada:** son reacciones en la que los productos finales tras la interacción, además del núcleo residual, constan de más de una partícula masiva, en general de hadrones de alta energía (Leo, 1994) (ver Fig. 2.1); este proceso sólo ocurre para neutrones con energía $E \geq 100 \text{ MeV}$.

A cada tipo de interacción corresponde una sección eficaz ($\sigma_{\text{elástica}}$, $\sigma_{\text{inelástica}}$, $\sigma_{\text{captura-radiativa}}$, $\sigma_{\text{fisión}}$, etc.), la sección eficaz total estará dada por (2.22), en donde la suma se justifica ya que los fenómenos de interacción pueden ser entendidos como eventos mutuamente excluyentes, es decir la existencia de uno de ellos no implica la de otro. Al multiplicar σ_{tot} por la densidad de átomos se obtiene el camino libre medio (2.23) para un haz de neutrones que penetra en la materia.

En analogía con los fotones, un haz de neutrones que atraviesa un material será atenuado de acuerdo con la ley de decaimiento exponencial (2.24) (Leo, 1994).

$$\sigma_{\text{Total}} = \sigma_{\text{elástica}} + \sigma_{\text{inelástica}} + \sigma_{\text{captura-radiativa}} + \sigma_{\text{fisión}} + \dots \quad (2.22)$$

$$\frac{1}{\lambda} = N\sigma_{\text{Total}} = \frac{N_A\rho}{A}\sigma_{\text{Total}} \quad (2.23)$$

$$N = N_0 e^{-x/\lambda} \quad (2.24)$$

2.2. Sistemas de Detección

Con el fin analizar las propiedades, los procesos ocurridos y las partículas emitidas, absorbidas o dispersadas en los fenómenos de interacción radicación materia, se ha dispuesto, en la práctica, de métodos y dispositivos adecuados para la detección de la radiación, ya sea de forma directa o indirecta.

Desde finales del siglo XIX hasta la actualidad, se cuenta con una amplia variedad de detectores, todos ellos basados en un mismo principio: la transferencia, total o parcial, de la radiación al detector, donde se transforma en información accesible a la percepción humana (Leo, 1994). El proceso y mecanismo mediante el cual se “transforma” la radiación depende exclusivamente del detector y de su diseño, pese a esto se lista a continuación una serie de características propias del detector, cualesquiera que este sea.

- **Sensibilidad:** es la capacidad de producir una señal utilizable; un detector no es sensible a todas las formas de radiación, sino que se diseña para rangos específicos de energía. La sensibilidad depende de varios factores como: la sección eficaz de ionización en el detector, la masa, el ruido inherente y el material del volumen sensible del mismo.
- **Respuesta del detector:** es la relación entre la energía radiada y la “altura” de la señal de salida en respuesta a la interacción de la radiación con el detector (Leo, 1994).
- **Resolución:** indica qué tan preciso es el detector para diferenciar energías distintas pero cercanas. La resolución se puede medir mediante el envío de un haz monoenergético y observando el espectro resultante, en particular el ancho, a la altura media, del pico espectral característico. Si ΔE es el ancho del pico espectral, la resolución relativa a la energía estará dada por:

$$\text{Resolución} = \Delta E/E \quad (2.25)$$

- **Tiempo de respuesta:** es el tiempo que el detector tarda en generar una señal de salida, desde el instante en que la radiación incidió sobre este.

- **Eficiencia:** se define como la relación entre el número de eventos emitidos por la fuente y el número de eventos registrados por el detector, esto es:

$$\epsilon_{Total} = \frac{\text{Eventos registrados}}{\text{Eventos emitidos por la fuente}} \quad (2.26)$$

La eficiencia depende también de la geometría del detector y de la probabilidad de ocurrencia de una interacción de la radiación incidente con este.

- **Tiempo muerto:** representa un tiempo de inoperatividad durante el cual el detector no puede producir ni recibir otras señales, mas que la que esta siendo procesada.

Una clasificación de todos los tipos de detectores resultaría insatisfactoria, dado que el desarrollo de estos, es hoy en día una actividad prolífica en ciencia y tecnología. Sin embargo es válido, y aun actual, hablar de detectores en los cuales el paso de las partículas dejan un rastro visible en un medio específico, y de otros que mediante procesos indirectos, dan cuenta de las propiedades de la radiación, un esquema de los tipos de detectores se muestra en la Fig. 2.8.

Con el fin de precisar algunos aspectos de los detectores de radiación, cuya importancia histórica y uso actual han sido imprescindibles en el entendimiento de los fenómenos de interacción radiación-materia, se mencionan algunos de los detectores mas comunes en la práctica como son: los de ionización gaseosa, de centelleo, de neutrones y detectores semiconductores (una descripción de estos dispositivos se realiza en el Apéndice B). A continuación se listan algunos de los detectores mas utilizados para diferentes fines, tanto en la industria como en investigación científica:

- Calorímetro
- Cámara de burbujas
- Cámara de chispas
- Cámara de flujo
- Cámara de niebla
- Cámara gaseosa de microestructura
- Cámara de hilos
- Centellador
- Detector Cherenkov
- Detectores de ionización gaseosa (cámara de ionización, contador Geiger)

- Detector de transición de radiación
- Detector semiconductor
- Fotomultiplicador
- Emulsiones fotográficas
- Detector RICH
- Detector de tiempo de vuelo
- Cámara de proyección de tiempo

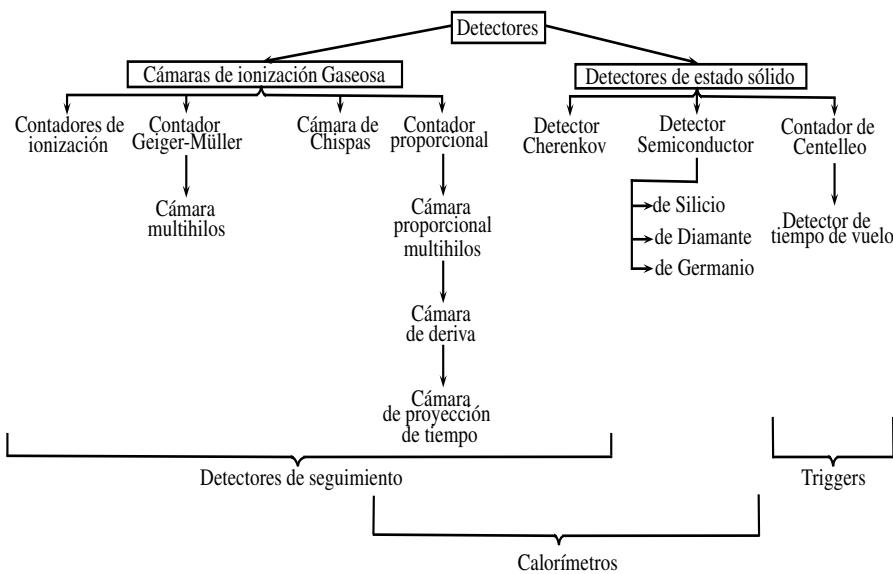


Figura 2.8: Clasificación de detectores de radiación.

Capítulo 3

GEANT4: Revisión general de software

"Computer science is not as old as physics; it lags by a couple of hundred years. However, this does not mean that there is significantly less on the computer scientist's plate than on the physicist's: younger it may be, but it has had a far more intense upbringing!".

Richard P Feynman.

Los actuales experimentos en física nuclear, física de partículas y otras disciplinas donde las interacciones radiación-materia resultan relevantes, plantean enormes desafíos respecto a la creación de software robusto y de fácil aplicación (Agostinelli, Allison, Apostolakis, y cols., 2003), contando además con la creciente demanda de sistemas de detección, que requieren simulaciones a gran escala de los detectores de partículas utilizados en dichos experimentos.

El tamaño, la demanda, la complejidad y la sensibilidad de los detectores son limitados por la disponibilidad de coste moderado, siendo viables los sistemas informáticos de alta capacidad sobre los cuales se realizan simulaciones más grandes y más complejas (Agostinelli y cols., 2003). En respuesta a esto, se desarrolla un conjunto de herramientas para la simulación de fenómenos de interacción radiación-materia, *GEANT4*, bajo un amplio espectro energético, basándose en diversos modelos físicos, provenientes de la experiencia acumulada de muchos colaboradores en el campo de la simulación por Montecarlo, de detectores de partículas y de procesos físicos.

Las problemáticas ya expuestas conllevan un desafío no solo de carácter investigativo, sino que además afecta al sector educativo desde el punto de vista en que la enseñanza de estos tópicos es guiada por el avance científico-tecnológico de los mismos; de allí que diversos fenómenos no puedan ser completamente explicados y entendidos en el "tablero", resultando imprescindible la formación de profesores-investigadores, bajo la

GEANT4: Revisión general de software

filosofía *Hands-On*, en el uso y manejo de **GEANT4** con miras educativas y/o investigativas, el cual es considerado hoy día, la herramienta predilecta por la gran mayoría de experimentalistas en los campos de la física de altas energías y la física médica.

3.1. Descripción e historia

“**GEANT4**” es el acrónimo de **Geometry and tracking** (Geometría y seguimiento), un conjunto de herramientas para la simulación de detectores de partículas y las interacciones entre éstas con la materia, que implementa el método de Montecarlo (G. Cern, 2011).

GEANT4 es el sucesor de la serie de software **GEANT** desarrollada por **CERN**, y el primero en utilizar programación orientada a objetos basada en C++; corrientemente, las contribuciones a su desarrollo y mantenimiento son atendidas por colaboración de la comunidad internacional; el desarrollo colaborativo de **GEANT4**, en términos de tamaño, de alcance del código y de número de usuarios, representa uno de los proyectos más grandes y ambiciosos de este tipo (Agostinelli y cols., 2003). Se ha demostrado que los rigurosos métodos de ingeniería de software y métodos orientados a objetos, pueden aplicarse provechosamente a la producción de un producto de software coherente y fácil de mantener, incluso con “requisitos” y necesidades en constante evolución presentados por la investigación en física.

Actualmente **GEANT4** es considerado como la herramienta computacional predilecta por la gran mayoría de experimentalistas en el campo de la física de altas energías, siendo también utilizado en otros campos como: física nuclear, física de aceleradores, diseño de detectores de partículas, radio física, ciencias del espacio, física médica entre otras. El software es utilizado en un sin número de proyectos de investigación en todo el mundo, sin embargo su objetivo inicial fue la explotación de la tecnología orientada a objetos para mejorar la transparencia de las simulaciones en física y por lo tanto validar los resultados de ésta (Allison y cols., 2006).

GEANT4 dispone de una completa gama de funcionalidades, en la que se incluye el seguimiento de partículas, geometría de detectores, y modelos físicos que abarcan procesos electromagnéticos, hadrónicos y ópticos, junto con un gran número de partículas y materiales, lográndose simulaciones en un rango de energía que comprende desde ~ 250 eV hasta ~ 7 TeV (Allison y cols., 2006).

La información y la experiencia acerca de los fenómenos físicos y de las herramientas informáticas involucradas, han sido extraídos de diversas fuentes de todo el mundo; en este sentido, **GEANT4** actúa como un repositorio que incorpora una gran parte de todo lo que

se sabe acerca de las interacciones radiación-materia.

Todos los aspectos del proceso de simulación han sido incluidos en el “kit” de herramientas de **GEANT4** como se muestra en el siguiente listado:

- Geometría del sistema
- Materiales involucrados
- Partículas de interés
- Generación de eventos
- Seguimiento de las partículas a través de los materiales y de los campos
- Procesos físicos que rigen las interacciones de las partículas
- Respuesta de los detectores
- Generación de datos de los eventos
- Almacenamiento de eventos y de trayectorias
- Visualización de detectores y de las trayectorias de las partículas
- Captura y análisis de datos de la simulación

El desarrollo de **GEANT4** se remonta a dos estudios realizados en 1993, de manera independiente por CERN y KEK (High Energy Accelerator Research Organization) (Amako y cols., 2000). Ambos grupos investigaban cómo las modernas técnicas de computación podían ser aplicadas para mejorar el software existente Geant3,¹ siendo éste un punto de referencia, fuente de ideas y de experiencias valiosas. Los dos proyectos se fusionaron, presentando una propuesta al DRDC (CERN Detector Research and Development Committee) (Dellacqua y cols., 1994) para elaborar un software de simulación basado en tecnología orientada a objetos en C++. El resultado fue RD44, una colaboración global que creció para incluir los esfuerzos de 100 científicos e ingenieros, procedentes de más de 10 experimentos realizados en Europa, Rusia, Japón, Canadá y Estados Unidos. Un primer prototipo fue entregado a finales de 1995, y la primera versión de producción fue entregada a finales de 1998. Con esta nueva versión, el proyecto RD44 había cumplido sus objetivos completando el desarrollo del software (Allison y cols., 2006).

¹Geant3, desarrollado en la década de 1970 por CERN, fue escrito en fortran bajo el marco de CERNLIB (CERN Program Library); desde el año 2000, la última versión de Geant3 ha estado esencialmente “estancada”, recibiendo ocasionalmente correcciones de errores (G. Cern, 2011).

Desde 1999, los servicios de producción, soporte al usuario y desarrollo de **GEANT4** han sido gestionados por la comunidad internacional, mediante un memorando de entendimiento (Giani y cols., 1998) firmado por todos los colaboradores formales (laboratorios participantes, experimentos e institutos nacionales). Este memorando está sujeto a renovación tácita cada 2 años y establece una estructura de colaboración integrada por un consejo de colaboración, un comité de dirección técnica y varios grupos de trabajo. En el documento también se define la forma en que los recursos económicos obtenidos por colaboración internacional son “repartidos” en las diferentes divisiones del proyecto (Agostinelli y cols., 2003).

El equipo de colaboradores de **GEANT4** proporciona documentación y soporte a los usuarios mediante diversas guías y manuales, incluyendo procedimientos que van desde la instalación del software hasta “kits” de entrenamiento para el manejo del mismo. La documentación está destinada a cubrir las necesidades que pueda tener un usuario principiante, así como un usuario experto que desee explotar las capacidades de **GEANT4**. Además se cuenta con un equipo de apoyo y un sistema de enrutamiento hacia especialistas, que ofrece a los usuarios una amplia asistencia a problemas de software, ejecución de códigos, consultorías sobre resultados, y respuesta a solicitudes de mejora del software.

GEANT4 y su código fuente se encuentra disponible, de manera libre (G. Cern, 2011) (bajo licencia BSD²), hasta la versión 8.1 (publicada en junio 30 de 2006), en la página web oficial del proyecto (<http://geant4.web.cern.ch/>). Desde la versión **8.1** hasta la versión **10.X** (10.4: 8 de diciembre de 2017, patch-02: 25 de Mayo de 2018), **GEANT4** está previsto de la licencia de software *GEANT4 Software License* (ver Apéndice C).

3.2. Funcionalidad de **GEANT4**

GEANT4 está constituido por 17 *clases*³ (Amako y cols., 2000), mediante las cuales se puede obtener una simulación, reuniendo todos los componentes esenciales de cada uno de los once aspectos listados en la sección precedente. La Fig. 3.1 se ilustra un esquema

²La licencia BSD (Berkeley Software Distribution), llamada así porque se utiliza en gran cantidad de software distribuido junto a los sistemas operativos BSD, permite que el autor, mantenga la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario (Foundation, s.f.).

³En C++ una *clase* es una estructura que ofrece la posibilidad de extender tipos predefinidos en el lenguaje básico. Cada *clase* representa un nuevo tipo, un nuevo conjunto de objetos caracterizado por ciertos valores (“propiedades”) y las operaciones (“métodos”) disponibles para crearlos, manipularlos y destruirlos. Cabe notar que los objetos en C++ son abstraídos mediante *clases*.

de las *clases* empleadas en **GEANT4**⁴.

Desde la parte inferior hacia la superior de la Fig. 3.1 (aunque la presentación de la estructura del software sea en sentido contrario), las *clases* de la parte inferior del diagrama se utilizan prácticamente en todas las *clases* superiores (Agostinelli y cols., 2003) y constituyen los cimientos del kit de herramientas de **GEANT4**.

Una descripción del diseño y arquitectura de las *clases* en **GEANT4** deriva varias categorías en su estructura: Global, Partículas y materiales, Geometría, Visualización e interfaz de usuario, Trayectorias, Procesos físicos, Seguimiento de partículas y, Eventos y ejecución⁵. Conceptualmente se pueden interpretar así:

- **Global** reúne todas las *clases*, tipos, estructuras y constantes que se consideran de uso general, incluyendo el sistema de unidades, constantes físicas, valores numéricos y el manejo de números aleatorios. En esta categoría también se define la interfaz con las bibliotecas de software de terceros, como CLHEP (*Class Library for High Energy Physics*).
- **Partículas y materiales:** estas dos categorías proveen los recursos necesarios para describir las propiedades físicas de los materiales y las partículas, a ser utilizadas en la simulación de interacciones radiación-materia.

La definición de las partículas se realiza mediante la clase **G4ParticleDefinition** (Allison y cols., 2006), la cual proporciona un amplio conjunto de partículas (más de 400), recurriendo a clases “virtuales” para declarar grupos como leptones, bosones, mesones, bariones, etc; esto permite la implementación de clases asociadas a partículas concretas, tales como **G4PionMinus**, **G4Electron**, etc., que definen las propiedades por las cuales se caracterizan las partículas reales, como nombre, masa, carga, espín, entre otras.

G4ParticleDefinition dispone de una representación compatible con la descripción dada por el PDG⁶ (Amako y cols., 2000) para cada partícula.

⁴Un diagrama de *clases* es un diagrama estático que describe la estructura de un sistema mostrando sus *clases*, atributos y las relaciones entre ellos. Los diagramas de *clases* son utilizados durante el proceso de análisis y diseño de sistemas, en los cuales existe una estructura conceptual de la información.

⁵Una revisión más profunda y técnica acerca de composición de las *clases* puede ser consultada en **GEANT4 User's Guide for Toolkit Developers** (Geant4 y cols., 2010).

⁶El Particle Data Group (PDG) es una colaboración internacional de físicos de partículas, que compila y re-analizar los resultados publicados en relación con las propiedades de las partículas y sus interacciones.

GEANT4: Revisión general de software

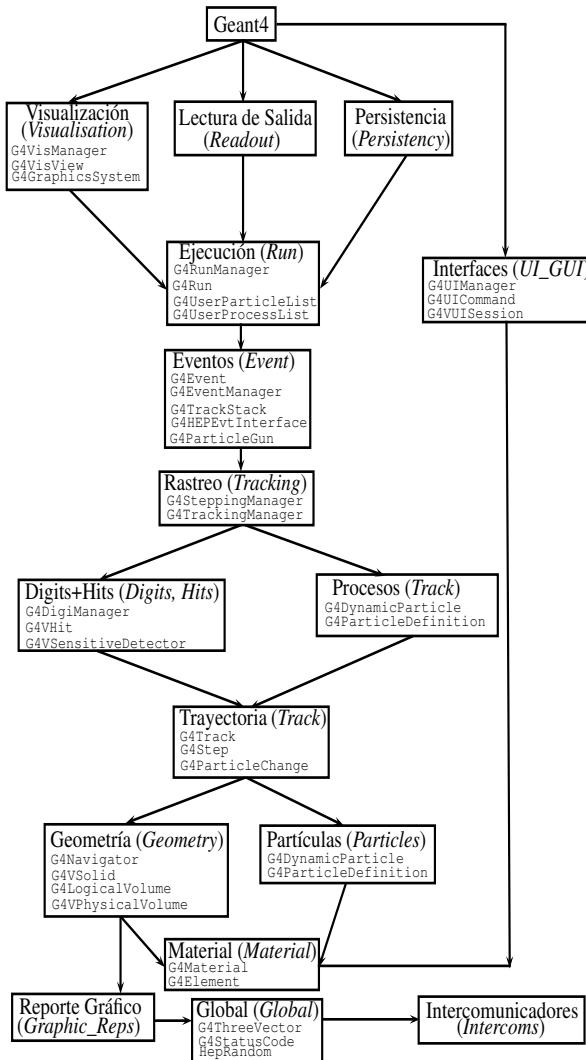


Figura 3.1: Diagrama de clases de GEANT4.

La categoría *materiales* tiene por objeto “reflejar” los atributos de todas las sustancias presentes en la naturaleza, bien sean materiales de un solo elemento, mezclas de elementos, materiales compuestos por un isótopo, o por la mezcla de varios isótopos.

Dadas las propiedades físicas de los materiales, estos pueden ser descritos de manera quasi-genérica por cantidades como la densidad o la composición, éstas sólo están presentes en clases específicas para cada sustancia.

Las características del material en respuesta a la radiación, longitud de interacción, pérdida de energía, excitación, coeficientes de Bethe-Bloch, factores de corrección etc, son calculados a partir de cada elemento (Dellacqua y cols., 1994).

- **Geometría:** ofrece la posibilidad de diseñar estructuras geométricas, que permiten el paso, la propagación y la interacción de partículas con las mismas.

La geometría es descrita por un listado de diferentes elementos, a los que se les especifica sus posiciones y orientaciones, obedeciendo la norma ISO STEP (automation syst. & integr., 1994), mediante la cual se tienen tres tipos de modelado de sólidos⁷: a) geometría constructiva de sólidos (CGS), b) representación de fronteras (B-rep), y c) áreas de barrido sólido (SWEPT) (Agostinelli y cols., 2003).

La clase G4Navigator manipula ecuaciones de movimiento realizando un tratamiento de las condiciones de contorno (Allison y cols., 2006), de donde se obtienen las posibles trayectorias por las cuales las partículas se pueden propagar en acción de diferentes campos.

- **Visualización e interfaz de usuario:** estas categorías se basan en una interfaz abstracta orientada a objetos, que resulta independiente de cualquier sistema de gráficos que se pueda utilizar. Sin embargo GEANT4 permite múltiples aplicaciones para las bibliotecas de gráficos, tales como **OpenGL**, **Inventor**, **DAWN**, **VRML** (Amako y cols., 2000) y también las modernas interfaces gráficas de usuario OPACs o Momo.

Por encima de estas *clases*, se encuentran las *clases* necesarias para realizar y describir el seguimiento de partículas y de los procesos físicos involucrados.

- **Trayectorias:** permite la descripción de las trayectorias de las partículas antes y después de las interacciones.

⁷En la terminología de la programación orientada a objetos, debe entenderse un sólido como la abstracción de un volumen que describe su forma y sus dimensiones; mediante éste se recurren a mecanismos para determinar si una partícula está dentro o fuera, y para calcular en qué medida es un punto de su frontera.

GEANT4: Revisión general de software

- **Procesos físicos:** esta categoría implementa los modelos de las interacciones físicas, como interacciones electromagnéticas de leptones, de fotones, de hadrones y de iones entre otras.

El diseño orientado a objetos es utilizado para separar la física y el proceso de *tracking* (Allison y cols., 2006), lo cual permite a los usuarios analizar un determinado proceso físico, sin necesitar conocimiento alguno acerca del algoritmo de la simulación.

El cálculo de la secciones transversales es necesario para deducir el camino libre medio de las interacciones y para normalizar las predicciones del modelo empleado. En la mayoría de los casos, estas secciones se calculan directamente o con parámetros de datos. La forma en que se calcula el estado final del sistema se puede dividir en modelos alternativos o complementarios, de acuerdo con, por ejemplo, el rango de energía, el tipo de partículas, los materiales u otros. La clase G4VProcess tiene a disposición múltiples implementaciones de procesos y modelos relacionados con la fenomenología de las interacciones radiación-materia.

- **Seguimiento de partículas (*Tracking*):** esta clase es la encargada de gestionar la propagación de partículas, teniendo en cuenta sus interacciones con la materia (G4TrackingManager). Las trayectorias proporcionan información acerca de la física (posición, velocidad, etc.), de una o mas partículas que se han propagado en un medio específico; en GEANT4 las trayectorias se describen mediante los "objetos" de la clase G4Track.

La clase G4TrackingManager es la encargada de propagar una partícula de forma gradual (Agostinelli y cols., 2003), delegando el control de un paso a la clase G4SteppingManager. Cada partícula se mueve *paso a paso* con una tolerancia que permite optimizar notablemente el desempeño de la ejecución, conservando la precisión en el proceso de *tracking*. Todos los procesos físicos asociados a una partícula se realizan por pasos, teniendo en cuenta que el paso máximo permitido por el usuario, se define en la clase G4UserLimits mediante el método SetMax-AllowedStep (Agostinelli y cols., 2003).

- **Eventos y Ejecución:** El concepto de **ejecución**, correr (*run*), es básicamente el mismo que se adopta en la realización de un experimento de física. La diferencia sustancial radica en que un conjunto de sucesos físicos es simulado, y ejecutado en un tiempo finito, utilizando la geometría del experimento (detectores, materiales, etc.), e involucrando los procesos de la física misma. GEANT4 puede realizar múltiples

“ejecuciones” en una simulación (Amako y cols., 2000), dicha configuración se encuentra en la clase `G4Run`, dentro de la cual todos sus “objetos” representan “ejecuciones”. La categoría `eventos` brinda una interfaz independiente para interactuar con fuentes externas a la generación de partículas en un evento.

La “independencia” de estas clases permite que un código de simulación, realizado en **GEANT4**, no dependa de opciones específicas de los generadores ni de los procesos físicos. La clase `G4EventManager` controla el comportamiento de todas las partículas pertenecientes a un evento de la simulación; el usuario puede controlar la prioridad de seguimiento de una partícula mediante las herramientas de esta clase.

La clase `G4Event` almacena transitoriamente los resultados de la simulación, por ejemplo, los conjuntos de trayectorias, aciertos de colisión a lo largo de las mismas, y los resultados de la digitalización.

Las potencialidades de las *clases*, junto con el acople a utilidades externas hacen de **GEANT4** un conjunto de herramientas que mediante interfaces abstractas, proporciona visualización y múltiples manipulaciones del experimento diseñado.

Las *clases* están diseñadas de una forma compacta y práctica para que el usuario pueda ampliar y/o modificar sus “servicios” de acuerdo a necesidades específicas.

La creación de *clases* con interfaces coherentes, fue justamente la que condujo a la idea de “caja de herramientas”, ello implica a su vez, que un usuario puede “diseñar” un programa en tiempo de compilación (Agostinelli y cols., 2003)⁸ bien sea seleccionando componentes de la caja de herramientas o diseñados por si mismo.

El conjunto de herramientas de **GEANT4** ofrece al usuario la posibilidad de crear un modelo geométrico con un amplio número de componentes, de diferentes formas y materiales, siendo posible la creación de detectores de partículas y/o elementos de volumen “sensible” mediante los cuales se lleva un registro de la información mediada en los procesos ocurridos, antes y después de las interacciones radiación-materia; esto es, se obtiene una respuesta, digitalizada, de los detectores.

Con fines prácticos, pensando en la interactividad usuario-software, se ha dispuesto en la caja de herramientas, de una selección de interfaces gráficas de usuario que permiten visualizar la geometría y las trayectorias de las partículas simuladas, bajo una variedad de

⁸El *tiempo de compilación* es aquel intervalo de tiempo en el que se compila un código, a una forma de código ejecutable por una máquina.

sistemas gráficos bien definidos, siendo a su vez viable la posibilidad de implementar esta interfaz con otros software.

El kit de herramientas proporciona también un conjunto completo de modelos físicos mediante el cual se describen los procesos y el comportamiento de las partículas. Los usuarios tienen la opción de elegir entre diferentes modelos y aplicaciones, y modificarlos y/o complementar el conjunto por defecto.

3.3. Desarrollo de aplicaciones

Esta sección pretende proporcionar al lector una visión general acerca del uso de **GEANT4** en el desarrollo de aplicaciones, es decir, desde un punto de vista práctico, la creación de simulaciones referidas a fenómenos interacción-materia, presentando también el entorno global de computación requerido, sin ser una revisión detallada del mismo, aunque sí proporcionando las bases “técnicas” y conceptuales suficientes. Deben tomarse como referencia algunos conceptos fundamentales de la programación orientada a objetos en C++, pues este es el andamiaje de **GEANT4**, en particular, conocimientos básicos de sistema operativo Linux, sobre el cual se enmarca el trabajo práctico de este escrito.

GEANT4 puede ser instalado en diversas plataformas bajo sistemas *Linux*, *Mac* y *Windows*.⁹ Las “guías” oficiales de instalación del software están disponibles en <http://geant4.web.cern.ch/geant4/>.

Cabe precisar que para el desarrollo del presente trabajo se realizó la instalación de la serie **10.X** de **GEANT4**, con versión **10.4**, en modo mono-usuario, junto con la herramienta ROOT-C++, una infraestructura de software desarrollada por CERN, que cuenta con la funcionalidad necesaria para gestionar y analizar grandes cantidades de datos de una manera muy eficiente ¹⁰.

Para instalar **GEANT4.10.4**, previamente, se requiere tener un compilador de lenguaje C++ en el sistema operativo (g++ en el caso de distribuciones Linux), disponer de las

⁹En los sistemas *Microsoft Windows* se debe utilizar el software *Cygwin*, el cual es una colección de herramientas desarrollada por la empresa *Cygnus Solutions* para proporcionar un comportamiento similar al de los sistemas Unix.

¹⁰Información más detallada acerca de ROOT puede se consultada en el manual de usuario de ROOT: <https://root.cern.ch/root-user-guides-and-manuals>

bibliotecas CLHEP¹¹ (es decir tenerlas instaladas), y del código fuente de GEANT4¹².

La instalación se realiza a través de un terminal de Linux¹³ en *lenguaje bash*¹⁴ (aunque se puede utilizar otro *interprete de comandos*), mediante un script¹⁵ diseñado por el grupo de colaboradores GEANT4.

3.3.1. Descripción de instalación de software

Para este recurso educativo, se llevó a cabo el trabajo práctico de instalación de CLHEP, GEANT4 y ROOT en varias distribuciones: Ubuntu 10.04LTS a 16.04 LTS, Lucid Lynx, Debian squeeze 6.0.3, y Kubuntu 9.10 Karmic Koala, en 4 computadoras diferentes con las siguientes descripciones

1. 1Tb HD, 4Gb de RAM Física, Procesador Intel Core i5-7200U 2.50GHz x 4.
2. 1Tb HD, 4Gb de RAM Física, Procesador AMD Phenom II 2.8 GHz
3. 320Gb HD, 3Gb de RAM Física, Procesador Intel Core i3-350M 2.26Ghz
4. 250Gb HD, 3Gb de RAM Física, Procesador AMD Turion 64X2 Dual-Core 2.0 GHz
5. 80Gb HD, 512Mb de RAM Física, Procesador AMD Athlon 3.0 (939)

El ejercicio práctico de instalación y ejecución de una simulación-ejemplo descrita posteriormente en los siguientes capítulos, requiere la instalación de paquetes previos para poder hacer una instalación consistente de CLHEP, ROOT¹⁶ y GEANT4. Con el objeto de garantizar una instalación satisfactoria de los requerimientos, y hacer alfabetización científica tecnológica desde la práctica con el computador sobre el uso y manejo de

¹¹CLHEP, Class Library for High Energy Physics, es una librería programada en C++ que proporciona utilidades de vectores de cantidades físicas, generador de números aleatorios, geometría y álgebra lineal, orientadas a las necesidades de la investigación científica en física de altas energías.

¹²La documentación oficial de GEANT4 no explica limitaciones de Hardware para su instalación, sin embargo el usuario debe ser consciente de las potencialidades de su equipo de computo, pues estas determinan el tiempo de ejecución de una simulación.

¹³Linux utiliza una herramienta llamada *consola de comandos* o *Terminal* desde la cual se pueden ejecutar todas las tareas dentro del sistema operativo, como acceder a lugares del sistema, visualizar reportes, instalar/desinstalar software o editar la configuración del sistema.

¹⁴Bash (*bourne again shell*), un programa informático basado en la shell de Unix cuya función consiste en interpretar órdenes, de allí que tenga un lenguaje característico. Una breve descripción de los comandos más utilizados en este *lenguaje* se encuentra en el Apéndice D.

¹⁵Un script es un archivo de texto plano que contiene una serie de sentencias y de órdenes que pueden ser interpretadas por un *interprete de comandos*, por ejemplo *bash*.

¹⁶El Apéndice E ilustra un esquema de instalación básico de ROOT.

GEANT4: Revisión general de software

GEANT4, se ha preparado el Apéndice F para guiar al lector con una experiencia real de instalación. En este punto, y dada la naturaleza de este escrito, es importante su lectura para el lector que está aprendiendo sobre el uso y manejo de *GEANT4*; complementario a esto, se sugiere revisar la documentación oficial de estos tres software:

- CLHEP: <http://proj-clhep.web.cern.ch/proj-clhep/>
- ROOT: <http://root.cern.ch/drupal/>
- *GEANT4*: <http://geant4.cern.ch/>

Las versiones utilizadas de software y los requerimientos de hardware y sistema operativo para este ejercicio práctico son:

- CLHEP: <http://proj-clhep.web.cern.ch/proj-clhep/DISTRIBUTION/tarFiles/clhep-2.4.1.0.tgz>
- ROOT: https://root.cern.ch/download/root_v6.14.06.Linux-ubuntu16-x86_64-gcc5.4.tar.gz
- *GEANT4*: <http://geant4-data.web.cern.ch/geant4-data/releases/geant4.10.04.p02.tar.gz>
- Hardware: 1Tb HD, 4Gb de RAM Física, Procesador Intel Core i5-7200U 2.50GHz x 4
- Sistema Operativo: Ubuntu 16.04LTS 64bits.

3.4. *GEANT4*: Generalidades y características

En la Fig. 3.2 se muestra el diagrama de flujo respectivo al funcionamiento de *GEANT4* para una simulación.

De acuerdo al esquema de procesos, todo lo que el usuario debe hacer se encuentra en el grupo *Programa de usuario*; es decir, el usuario construye la geometría y define los materiales del sistema de detección, adiciona las partículas y los procesos físicos relevantes (del listado o registro de física) y, finalmente, establece las condiciones de la simulación.

Antes de entrar en la codificación de una simulación, se deben tener en cuenta algunas de las convenciones utilizadas por *GEANT4*:

GEANT4: Revisión general de software

```
4 #include "MyPhysicsList.hh"           // Adicion de procesos fisicos de interes por el usuario
5 #include "MyPrimaryGenerator.hh" // Definicion del Evento primario (cinematica) por el usuario
6
7 int main () {
8
9     G4RunManager* runManager=new G4RunManager;           //Ejecucion
10
11    runManager->SetUserInitialization(new MyDetectorConstruction); //La geometria
12    runManager->SetUserInitialization(new MyPhysicsList); //La fisica
13    runManager->SetUserAction(new MyPrimaryGenerator); //La cinematica del evento generador
14    runManager->Initialize(); //Inicializacion
15
16    G4UImanager* UI=G4UImanager::GetUIpointer(); //Puntero a la interfaz de usuario
17    UI->ApplyCommand("run/verbose 1"); //Establece el nivel de la ejecucion
18    int numberOfEvent=3; //Numero de eventos a ser ejecutados
19    runManager->BeamOn(numberOfEvent); //Generacion de eventos
20    delete runManager; // Fin de la Ejecucion
21
22 }
```

Lo primero en el `main()` fue crear una instancia de la clase `G4RunManager`, que controla el flujo del programa y gestiona el número de eventos dentro de un intervalo de tiempo específico. Cuando el gestor de clase `G4RunManager` es creado, otros gestores de clases también son creados, y se destruyen automáticamente cuando la clase `G4RunManager` se destruye. Debe considerarse también las clases `G4Run`, `G4Event`, `G4Track`, `G4Step`, `G4Particle`, no presentes en este código pero si de gran utilidad como se verá mas adelante.

G4Run

La clase `G4Run` es una colección de eventos, que comparten iguales condiciones físicas dentro de un volumen sensible o detector. Para iniciar la simulación (“run”), en la interfaz de `GEANT4` (la cual aparece tras compilar el código), debe “invocarse” el comando: `/run/beamOn 8` en la interfaz gráfica derivada de la compilación de la simulación (el número 8 corresponde a ejecutar una corrida (“run”) de 8 eventos).

Luego de esto, la geometría no puede ser modificada (agregar volúmenes, cambiar materiales, etc), ni agregar o definir más procesos físicos, además de los ya incluidos por el usuario.

La estructura de esta clase puede establecer en general como:

```
1 #include "G4Run.hh"           //Inclusion de G4Run (Obligatorio)
2 #include "G4RunManager.hh" //Inclusion de G4RunManager (Obligatorio)
3 #include "UserRunAction.hh" // Definido por el usuario
4
```

GEANT4: Revisión general de software

```
5 UserRunAction::UserRunAction() {}
6 UserRunAction::~UserRunAction() {}

7
8 void UserEventAction::BeginOfRunAction(const G4Run*)
9 {
10 // Se puede definir parametros inciales para la corrida
11 }
12 void UserEventAction::EndOfRunAction(const G4Run*)
13 {
14 //Se define la colección de la estadistica final de la corrida,
15 //se cierran archivos de adquisicion de datos, por ejemplo.
16 }
```

donde se destacan en general dos métodos, `BeginOfRunAction()` y `EndOfRunAction()` los cuales permiten inicializar y ejecutar la simulación como una corrida en un experimento real.

Una forma de declararla e inicializarla dentro del `main` se ilustra enseguida (ver líneas 8 y 9):

```
1 #include "G4RunManager.hh"
2 void main(){
3 ...
4 G4RunManager* runManager = new G4RunManager;
5 ...
6 runManager->SetUserInitialization(physics);
7 ...
8 G4UserRunAction* run_action = new UserDefinedRunAction;
9 runManager->SetUserAction(run_action);
10 ...
11 delete runManager;
12 }
```

G4Event

La clase `G4Event` es la unidad básica de la simulación en `GEANT4`. En ella se declara una entrada (partículas primarias), una salida (colección de interacciones y trayectorias) y se implementa el análisis de datos junto con el “llenado” de histogramas. La estructura de esta clase puede establecer en general como:

```
1 #include "G4Event.hh"           //Inclusion de G4Event (Obligatorio)
2 #include "G4EventManager.hh"   //Inclusion de G4EventManager (Obligatorio)
```

```
3 #include "UserEventAction.hh" // Definido por el usuario
4
5 UserEventAction::UserEventAction() {}
6 UserEventAction::~UserEventAction() {}
7
8 void UserEventAction::BeginOfEventAction(const G4Event*)
9 {
10 // Se puede definir parametros para salida de datos
11 }
12 void UserEventAction::EndOfEventAction(const G4Event*)
13 {
14 //Llenado de histogramas
15 }
```

donde se destacan en general dos métodos, `BeginOfEventAction()` y `EndOfEventAction()` los cuales permiten inicializar y ejecutar la simulación evento a evento.

G4Track

Esta clase es la responsable de la trayectoria de las partículas, y puede ser interpretada como una “fotografía instantánea” de la partícula, pues mantiene las cantidades físicas de la interacción actual de la partícula. La “fotografía” se va actualizando si se sale del “mundo” (es decir, del “espacio” en el cual se realiza el experimento), desaparece (por ejemplo decae o se aniquila), si su energía cinética es nula, o si existen umbrales descartados por alguna propiedad particular definida por el usuario.

G4Step

Contiene información de la partícula durante la ejecución de un evento, por ejemplo: energía en un tiempo final, y energía en un tiempo inicial. Esta clase “conoce” el material y volumen en cada punto y finaliza en la frontera con otro material o volumen, lo cual permite la solución de problemas de frontera e interfaces.

Sistema de unidades

Es importante especificar la librería que especifica dentro del `main()` la definición del sistema de unidades:

```
#include "G4SystemOfUnits.hh"
```

en la elaboración de los códigos de simulación, en particular en la geometría (como se verá mas adelante). GEANT4 permite el uso de cantidades con unidades (mm, eV, Kelvin,

etc) y con prefijos (mega, mili, nano, etc), en el código fuente se puede tener algo similar a:

```
1 // Definicion de nuevas unidades a partir del SI
2 milimetro = mm = 1;
3 metro = m = 1000*mm;
4 m3 = m*m*m;
5
6 // Para utilizarlas se debe dar la unidad de los datos
7 // que se van a introducir
8 G4double energia = 10.4*GeV;
9 G4double densidad = 11*g/cm3;
```

Al imprimir o almacenar datos y variables, se debe especificar la magnitud y su respectiva unidad de la siguiente manera:

```
1 G4cout << "Energia particula" << Energia / MeV << "MeV" << G4endl;
2 G4cout << "Posicion x" << Posicion / m << "m" << G4endl;
```

GEANT4 permite también la definición de nuevas unidades, para ello, dentro del código fuente se procede así:

```
1 #include "SystemOfUnits.hh"
2 static const G4double pulgada = 2.54*cm;
3
4 // Implementacion
5 longitud = 3.4*pulgada;
6
7 // Otro ejemplo G4UnitDefinition(nombre, simbolo, categoria, valor)
8 G4UnitDefinition("km/hora", "km/h", "Rapidez", km/(3600*s));
```

3.4.2. Definición de Geometrías

Para definir un sistema de detección (cristales centelladores, fotomultiplicadores, calorímetros, etc.), se deben definir diversas propiedades que “representan” los elementos geométricos de los cuales esta compuesto: su forma, sus materiales y la estructura del sistema, junto con los atributos de visualización y otras propiedades.

En GEANT4, la *geometría* se constituye de un número de volúmenes; cada volumen representa un objeto, con su forma y sus características físicas, excepto el “mayor volumen posible”, llamado “*World volume*”, el cual puede ser entendido como el “laboratorio” donde se realiza el experimento (i.e. el volumen principal que contiene toda la configuración geométrica); éste, es un volumen virtual que contiene, con cierto margen, todos los otros volúmenes de la geometría del detector. Todos los volúmenes creados pueden ser “*contenidos*” por otros volúmenes, en cualquier caso, todos ellos estarán dentro del *World volume*, que en su forma más simple y eficiente, se describe como una caja.

En el caso que un volumen es contenido por otro, se nombra al primero de estos (volumen contenido) “volumen hija”; así mismo, el volumen contenedor será llamado “volumen madre”; el sistema de coordenadas del “volumen madre” es utilizado para especificar la posición del “volumen hija”.

La definición de un volumen se realiza a través de las clases `G4VSolid`, `G4LogicalVolume`, `G4VPhysicalVolume` y `G4VUserDetectorConstruction`; en la primera de estas, se especifican las *formas* y los *tamaños*; en la segunda los *volúmenes hija*, *materiales*, *sensibilidad*, *propiedades físicas*, etc; en la tercera, la *posición* y posibles *rotaciones* dentro del marco de referencia establecido; finalmente, la cuarta clase, engloba a las clases anteriores, y es ésta la que contiene toda la geometría del sistema a simular, definida por el usuario. Esta última clase se incluye en el `main()` por medio de la clase `G4RunManager`.

En “forma” de código, la clase en la que se construye la geometría, por ejemplo `MyDetectorConstruction.cc`, se ilustra en el siguiente recuadro:

```
1 G4VPhysicalVolume* MyDetectorConstruction::Construct()
2 {
3     ...
4     // Aquí se definen los materiales, sólidos, detectores sensibles,
5     //y propiedades físicas
6     return World_phys;
7 }
```

Para crear un volumen, primero se debe “dibujar” la forma geométrica del volumen; es decir crear un sólido. En GEANT4, un sólido es un objeto geométrico que tiene una forma definida y posee valores específicos para cada punto del espacio. GEANT4 cuenta con una biblioteca predefinida de sólidos (cubos, toroides, conos, cilindros, etc), de donde se puede elegir cualesquiera que se necesite, tan solo especificando su nombre y su prolongación a lo largo de cada eje cartesiano.

GEOANT4: Revisión general de software

Por ejemplo, para crear una caja, basta agregar el siguiente código en la parte de la construcción del detector (`MyDetectorConstruction.cc` en el código anterior):

```
1 G4double expHall_x = 3.0*m;
2 G4double expHall_y = 1.0*m;
3 G4double expHall_z = 1.0*m;
4 G4Box* experimentalHall_box
5   = new G4Box("expHall_box", expHall_x, expHall_y, expHall_z);
```

Esto crea una caja con el nombre “`expHall_box`” y cuyas dimensiones son de -3.0 a +3.0 m en el eje X, de -1.0 a +1.0 m en el eje Y, y de -1.0 a +1.0 m en Z.

Tras haber creado la forma sólida para el volumen, se recurre al concepto de “volumen lógico” para agregar otros atributos (material, permitividad, etc) a dicho volumen: así, bajo el código anterior, se puede crear un volumen lógico simple para llenar la caja con un gas, por ejemplo argón, introduciendo la siguiente línea:

```
1 G4LogicalVolume* experimentalHall_log
2   =new G4LogicalVolume(experimentalHall_box, Ar, "expLog");
```

Esto crea un volumen lógico llamado “`expLog`” asociado a la caja creada antes, que contiene los atributos de la misma.

Hasta el momento, sólo falta un aspecto para completar la construcción de la geometría del volumen: colocar el volumen en el interior de uno ya existente. Se debe decidir dónde estará posicionado el centro del volumen y si se efectúan rotaciones alrededor de algún eje.

Esta tarea se puede realizar mediante la creación de un volumen físico; por ejemplo, para colocar un “rastreador” de partículas que pasan a través del volumen, se requiere un volumen lógico, aquí llamado `tracker_log` (registro de seguimiento), cuya implementación vendría dada por:

```
1 G4double trackerPos_x = -1.0*m;
2 G4double trackerPos_y = 0.0*m;
3 G4double trackerPos_z = 0.0*m;
4
5 G4VPhysicalVolume*tracker_phys
6 = new G4PVPlacement(0, //no hay rotacion
```

```

7 G4ThreeVector(trackerPos_x,trackerPos_y,trackerPos_z),//Posicion
8 tracker_log, //volumen logico
9 "tracker", //nombre
10 experimentalHall_log, //volumen (logico) madre
11 falsa,0); //no hay operaciones booleanas

```

Esto sitúa al volumen lógico “tracker_log”, sin rotaciones, a -1.0 m en X del origen del volumen madre “experimentalHall_log”; el conjunto, de estos dos volúmenes, fue llamado “tracker”. De esta manera se ha completado la construcción de la geometría; volúmenes adicionales pueden ser creados de la misma manera.

A modo de ejemplo, y con el fin de solventar dudas, se presenta el siguiente código en donde se ha creado un cilindro recto circular, con radio de 60 cm y longitud de 50 cm, cuyo interior se llena de aluminio, al que se le ha llamado “tracker_tube”. Observe que para ello se ha utilizado la clase G4Tubs:

```

1 G4double innerRadiusOfTheTube = 0.*cm;
2 G4double outerRadiusOfTheTube = 60.*cm;
3 G4double hightOfTheTube = 25.*cm;
4 G4double startAngleOfTheTube = 0.*deg;
5 G4double spanningAngleOfTheTube = 360.*deg;
6
7 G4Tubs* tracker_tube
8     = new G4Tubs("tracker_tube",
9                 innerRadiusOfTheTube,
10                outerRadiusOfTheTube,
11                hightOfTheTube,
12                startAngleOfTheTube,
13                spanningAngleOfTheTube);
14
15 G4LogicalVolume* tracker_log
16     = new G4LogicalVolume(tracker_tube,A1, "tracker_log");

```

3.4.3. Definición de Materiales

Es necesario proporcionar información acerca de los materiales de los que está hecho el sistema de detección (al momento de definir la geometría), tales como los materiales del detector, del blanco, entre otros. La clase *materiales* debe contener información sobre las propiedades de los átomos y su distribución en el material. En general, en la naturaleza, los materiales que se encuentran (en cualquiera de los estados de la materia),

GEANT**4**: Revisión general de software

son compuestos químicos y/o mezclas, como se había mencionado anteriormente en la sección 3.2.

GEANT4 cuenta con una biblioteca de elementos, isótopos y materiales predeterminados; sin embargo, el usuario también puede definir nuevos materiales a través de las clases **G4Isotope**, **G4Element** y **G4Materials**.

Mediante la clase **G4Isotope** se describen las siguientes propiedades de los isótopos:

- número atómico
- numero nucleónico
- masa atómica
- energía de capa y
- otras cantidades como la sección transversal por átomo.

Se muestra un ejemplo que verifica el trabajo práctico de este recurso; en él se incluye un par de isótopos del uranio: ^{235}U (U235) y ^{238}U (U238), teniendo en cuenta que esta porción de código va dentro de **Construct()** en la definición del volumen físico, en la geometría: (Note: la única diferencia entre los isótopos radica exclusivamente en el número de nucleones y en su densidad.)

```
1 #include "G4Isotope.hh"
2 G4VPhysicalVolume* UserDetectorConstruction::Construct()
3 {
4 ...
5 G4double A, Z;
6 G4int n_elementos;
7 // Forma general: G4Isotope(Nombre, NAtomico, NNucleones, NMasico);
8 G4Isotope* U235 = new G4Isotope("U235", Z=92,
9 n_nucleones=235, A=235.01*g/mole);
10 G4Isotope* U238 = new G4Isotope("U238", Z=92,
11 n_nucleones=238, A=238.03*g/mole);
12 ...
13 return world_phys; // retorno al volumen fisico respectivo
14 }
```

De otra parte, la clase **G4Element** permite especificar diversas propiedades de los elementos, que además de las de los isótopos, se añade el *número de isótopos* que tiene

el elemento. En términos de código, la definición de los elementos es muy similar a la de los isótopos, como se observa en el siguiente ejemplo, donde se han definido hidrógeno y oxígeno:

```
1 {
2 ...
3 // Definicion Hidrogeno y Oxigeno
4 G4double A, Z;
5 // Forma general: G4Element(Nombre, Simbolo, NAtomico, NMasico)
6 G4Element* H = new G4Element("Hidrogeno", "H", Z=1, A=1.01*g/mole);
7 G4Element* O = new G4Element("Oxigeno", "O", Z=8, A=16.0*g/mole);
8 ...
9 }
```

La composición de materiales emplea la clase `G4Material`, la cual describe las propiedades macroscópicas de un volumen, teniendo en cuenta también las condiciones a las que está sometido dicho material como: densidad, estado, temperatura, presión, y otras cantidades macroscópicas, como longitud de radiación, camino libre medio, o poder de frenado dE/dx , entre otras.

Los materiales se pueden construir de un sólo elemento, a partir de una lista de componentes (elementos u otros materiales), o empleando la base de datos interna de `GEANT4`, la cual alberga una diversidad de materiales. De manera más específica, y teniendo en cuenta lo dicho anteriormente, se pueden tener materiales simples, moléculas, compuestos, mezclas de masa fraccional y/o materiales NIST (*National Institute of Standards and Technologies*); en cualquier caso, excepto para elementos puros, se debe especificar la contribución de cada elemento que compone el material; vale tener en cuenta que todas las cantidades, con o sin significado físico, que son constantes en un material determinado, se calculan y almacenan como datos derivados.

Materiales simples

Una manera simplificada para definir un material, es teniendo en cuenta aquellos materiales que están hechos de un solo elemento, i.e., un material puro. Los recuadros de la Fig. 3.3 muestran dos ejemplos en los que se “crean” materiales puros.

En el primero de ellos se define argón líquido, allí llamado `liquidArgon`, especificando su nombre, densidad, masa molar, y su número atómico. El puntero¹⁷ `lAr`, se utiliza para

¹⁷Un puntero es una variable que almacena la dirección en memoria (valida, inválida o nula) de otra variable,

GEANT4: Revisión general de software

```
1 G4double density = 1.390*g/cm3;
2 G4double A = 39.95*g/mole;
3 // Material hecho de un solo elemento
4 // Forma general: G4Material(Nombre, NAtomico, NMasico,Densidad);
5 //Argon liquido
6 G4Material* lAr
7     = new G4Material(name="liquidArgon", z=18., A, density);
```

```
1 #include "G4Material.hh"
2 {
3 ...
4 G4double Densidad = 2.7*g/cm3;
5 G4double A = 26.98*g/mole;
6 G4double Z = 13;
7 //Aluminio
8 G4Material* Aluminio = new G4Material("Aluminio", Z, A, Densidad);
9 ...
10 }
```

Figura 3.3: Ejemplos en los que se “crean” materiales puros.

especificar el material del volumen lógico. En el segundo ejemplo se define el *aluminio* considerando los mismos parámetros (nombre, densidad, masa molar, número atómico).

Materiales compuestos por moléculas

A partir de moléculas, también pueden formarse materiales, por ejemplo, agua o dióxido de carbono. La Fig. 3.4 muestra esquemas de código para definir materiales como el H_2O o el CO_2 , a este último se le suman condiciones como presión y temperatura; observe que los materiales se construyen a partir de sus componentes, especificando el número de átomos en la molécula.

Materiales compuestos

A diferencia de los ejemplos anteriores, para los compuestos, se tiene en cuenta la fracción de masa o el porcentaje de cada elemento en el material, en lugar del número la cual puede estar alojada o no en el programa.

```
1 # include " G4Material .hh"
2 ...
3 {
4 ...
5 // Codigo donde se ha definido hidrogeno y oxigeno
6 ...
7 // Forma general : G4Material (Nombre , Densidad , Nelementos )
8 //
9 Nombre -> AddElement (Elemento , NumeroDeAtomos )
10 G4Material * H2O = new G4Material ("Agua", Densidad , Nelementos );
11 H2O -> AddElement (H, 2);
12 H2O -> AddElement (O, 1);
13 ...
14 }
```

```
1 #include "G4Material.hh"
2 ...
3 {
4 //Codigo donde se ha definido carbono y oxigeno
5 ...
6 density = 27.*g/cm3 ;
7 pressure = 50.*atmosphere;
8 temperature = 325.*kelvin;
9
10 G4Material* CO2 == new G4Material ("CO2", density, nComponents=2,
11 kStateGas, temperature, pressure);
12
13 CO2->AddElement(C, nAtoms=1);
14 CO2->AddElement(O, nAtoms=2);
15
16 ...
17 }
```

Figura 3.4: Ejemplos en los que se “crean” materiales compuestos por moléculas.

de elementos.

La Fig. 3.5 ilustra dos recuadros en los que se definen H_2O con un contenido de 75% de

GEANT4: Revisión general de software

oxígeno, y otro en el que se define un compuesto con el nombre “Uranio enriquecido”, con el 90% de isótopos de ^{235}U y el 10% restante isótopos de ^{238}U .

```
1 // Se cambia H2O->AddElement(H,2) por:  
2 H2O->AddElement(0, 75*perCent);  
  
1 G4Element* U = new G4Element(name="Uranio enriquecido", symbol="U",  
2 ncomponents=2);  
3  
4 U->AddIsotope(U5, abundance= 90.*percent);  
5 U->AddIsotope(U8, abundance=10.*percent);
```

Figura 3.5: Ejemplos en los que se “crean” materiales compuestos.

Mezclas de masa fraccional

Para mezclas de masa fraccional, las cuales son constituidas por mezclas de elementos y moléculas, se debe indicar la masa fraccional de cada componente; así, por ejemplo, se puede obtener *aire*, el cual se constituye en un 78.084% de nitrógeno, 20.946% oxígeno, 0.035% de dióxido de carbono y 0.930% de gases inertes. Otro ejemplo puede ser el *aerogel*, un material sólido sintético derivado del gel con muy baja densidad (3 mg/cm^3), y altamente poroso, caracterizado principalmente por ser un excelente aislante térmico, compuesto por: *Si*, *O*, *H*, *C*, *SiO*₂ y *H*₂*O*:

```
1 {  
2 ..  
3 // Se definen los elementos Si, O, H, C y materiales SiO2, y H2O  
4 ..  
5 G4Material* AeroGel = new G4Material("AeroGel", 0.2*g/mole,3);  
6 ..  
7 AeroGel->AddMaterial(SiO2, 62.5*perCent);           // material  
8 AeroGel->AddMaterial(H2O,   37.4*perCent);           // material  
9 AeroGel->AddElement(C,     0.1*perCent);            // elemento  
10 ...  
11 }
```

Materiales NIST

Dependiendo de la necesidad, y por practicidad (tal vez!) resulte mas cómodo utilizar la base de datos del *NIST* incluida en *GEANT4*. La lista completa de los materiales definidos se encuentra en `$G4INSTALL/source/materials/src/G4NistMaterialBuilder.cc` (esto es, en el directorio de instalación de *GEANT4*). El Apéndice G muestra algunos de los materiales allí definidos.

La metodología empleada por la clase `G4NistManager` para utilizar un determinado material (por ejemplo agua, o cobre) es la siguiente:

```
1 #include "G4NistManager.hh"
2 #include "G4Material.hh"
3 ... ::Construct()
4 {
5     G4Material* AGUA = G4NistManager::Instance()->FindOrBuildMaterial("G4_WATER");
6     G4Material* Cu = G4NistManager::Instance()->FindOrBuildMaterial("G4_Cu");
7 ...
8     return world_phys;
9 }
```

3.4.4. Definición de Partículas

Como se mencionó, `G4Particle` recurre a tres clases para definir una partícula: `G4Track`, `G4DynamicParticle` y `G4ParticleDefinition`. En la Fig. 3.6 se muestra el diagrama respectivo, de las funciones y propiedades, a cargo de cada clase. De esta manera se puede disponer de las siguientes partículas:

- partículas ordinarias, tales como electrones, protones y gammas
- partículas de resonancia con vidas muy cortas, como mesones y bariones
- núcleos, tales como deuterones, alfas e iones pesados
- quarks, di-quarks y gluones

Cada partícula está representada por su propia clase, establecida en `G4ParticleDefinition` según pertenezca a una de las siguientes categorías: a) leptones, b) mesones, c) bariones, d) bosones, e) partículas de corta duración, ó, f) iones.

La clase `G4ParticleDefinition` contiene toda la información que caracteriza a una partícula. La mayoría de estas propiedades son de “sólo lectura” y no pueden ser alteradas por el usuario sin la reconstrucción de las bibliotecas. De allí que *GEANT4* proporcione más

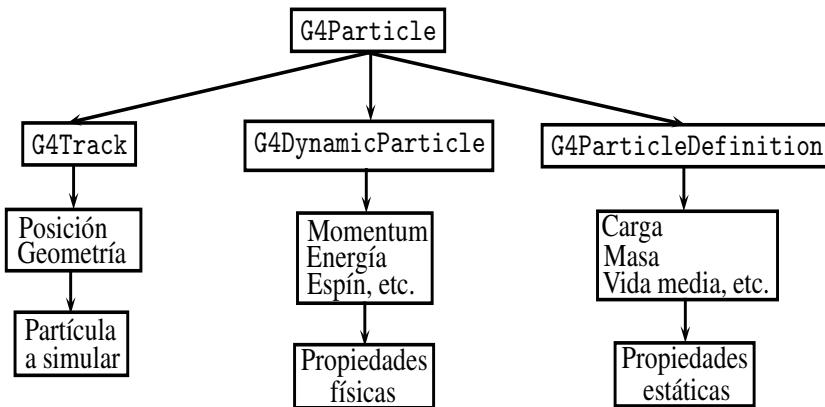


Figura 3.6: Estructura de la clase G4Particle.

de 100 tipos de partículas por defecto, para ser utilizadas en diferentes procesos físicos; esto significa que para aplicaciones “normales”, el usuario no tendrá que definir sus propias partículas.

Así por ejemplo, suponiendo que en una simulación se necesita un protón y un geantino¹⁸, en `ConstructParticle()` del `main()` se puede construir así:

```

1 void NombrePhysicsList::ConstructParticle()
2 ...
3 {
4     G4Proton::ProtonDefinition();
5     G4Geantino::GeantinoDefinition();
6 }
7 ...
  
```

3.4.5. Procesos físicos

En GEANT4 un proceso es la “acción(es)” que realiza una partícula, i.e., decaer o interactuar mediante algún mecanismo. En la Fig. 3.7 se clasifican los procesos físicos en discretos, continuos y otros. El efecto Compton, por ejemplo, es considerado un proceso

¹⁸Un geantino es una partícula de prueba, *virtual*, que se utiliza en la simulación y que no interactúa con los materiales.

discreto, ya que sólo hacia el final del G4Step se calcula la sección transversal.

De manera general, los procesos que pueden incluirse en las simulaciones de **GEANT4**, abarcan los necesarios para el transporte de radiación ionizante y no ionizante a través de materia, según se vio en el capítulo 2. Los principales procesos se listan a continuación:

- Electromagnéticos: estándar y a bajas energías
- Hadrónicos
- Transporte
- Decaimientos
- Fotones ópticos

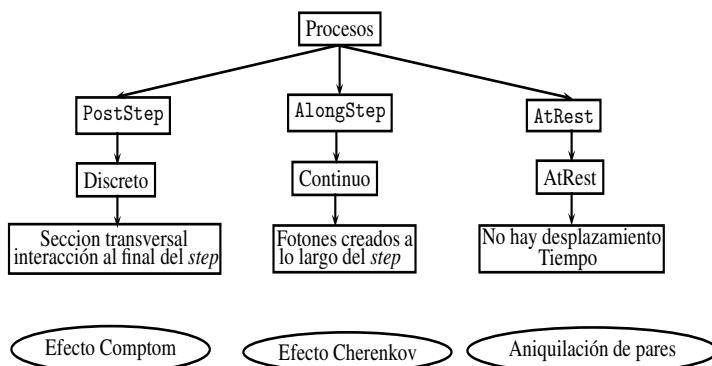


Figura 3.7: Clasificación de procesos físicos en **GEANT4**.

Especificar un proceso físico en una simulación de **GEANT4** requiere que el usuario cree una clase derivada de **G4VUserPhysicsList**, en la que se incluyan todos los procesos físicos, y que esta a su vez sea implementada en el **ConstructProcess()**; por ejemplo, si se quiere asociar un proceso de transporte a un geantino, junto con un registro de trayectoria, el código, dentro de **ConstructProcess ()**, se puede construir así:

```

1 void NombrePhysicsList::ConstructParticle()
2 ...
3 {
4     G4Proton::ProtonDefinition();
5     G4Geantino::GeantinoDefinition();

```

GEANT4: Revisión general de software

```
6 }  
7 ...
```

Aquí, AddTransportation() proporciona, desde G4VUserPhysicsList, una clase para registrar el proceso de transporte G4Transportation con todas las clases de partículas que hayan (en este caso particular un geantino). La clase G4Transportation describe el movimiento de las partículas en *espacio y tiempo*, por lo que este proceso resulta ser obligatorio para el seguimiento (*tracking*) de las partículas.

Un ejemplo en el cual se registra un proceso electromagnético de fotones de alta energía, puede ser implementado como:

```
1 void MyPhysicsList:: ConstructProcess()  
2 {  
3 //Definicion del proceso de transporte  
4   AddTransportation();  
5 //Adicion de procesos electromagnÃ©ticos  
6   ConstructEM();  
7 }  
8 void MyPhysicsList:: ConstructEM()  
9 {  
10 //Particulas (fotones gamma) a las que se les asocia los procesos  
11   G4ParticleDefinition* particle = G4Gamma::GammaDefinition();  
12   G4ProcessManager* pmanager = particle->GetProcessManager();  
13  
14 //Procesos asociados a los fotones  
15   G4PhotoElectricEffect * thePhotoElectricEffect = new G4PhotoElectricEffect();  
16   G4ComptonScattering * theComptonScattering = new G4ComptonScattering();  
17   G4GammaConversion* theGammaConversion = new G4GammaConversion();  
18  
19 //Registro de los procesos mediante el gestor de procesos pmanager  
20   pmanager->AddDiscreteProcess(thePhotoElectricEffect);  
21   pmanager->AddDiscreteProcess(theComptonScattering);  
22   pmanager->AddDiscreteProcess(theGammaConversion);  
23 }
```

Capítulo 4

Una simulación-ejemplo: *GEANT4 hands on!*

Como ejercicio de alfabetización científica y tecnológica, se ha pensado una simulación que contenga los elementos de lectura anteriores, que de alguna manera sensibilizaron al lector para poder afrontar la realización de una simulación en *GEANT4*. En ese sentido se propone una guía de realización de una simulación teniendo en cuenta: el contexto de la simulación, la necesidad de la simulación, modelo de simulación, la codificación de la simulación y la ejecución de la simulación.

Es realmente importante por parte del lector que tome un tiempo para descansar en la lectura, y así permitir que el cerebro no se embote. Esto puede conllevar a un ciclo de repetición de errores (que no encuentra solución a un problema) que posteriormente genere una afectividad negativa sobre el aprendizaje de *GEANT4*.

El ejercicio de lectura del capítulo anterior avecinda una sensibilización sobre el lenguaje de programación, estructura y filosofía de *GEANT4*, condición necesaria para establecer un diálogo software-usuario. Esto permite hacer consciente al lector que el conjunto de herramientas de *GEANT4* posee una extremada complejidad en su totalidad y aprendizaje, que deriva en una infinidad de posibilidades de simulación, y que como todo en la vida, su aprendizaje requiere “tiempo”.

La idea de disminuir la curva de aprendizaje de *GEANT4* se trata en establecer un proceso natural de aprendizaje y des-aprendizaje del ser humano, como el aprender a hablar, que es a través de la repetición y sensibilización de gestos, tonos y voz; así se aprende casi que todo en la vida, hasta la programación.

Cabe resaltar que este tipo de programación requiere un alto nivel de cuidado en la codificación, característica particular que permea toda la realización del ejercicio práctico, por lo que se debe tener especial atención en las repuestas del computador cuando se compile y ejecute cualquier código de programación, porque como se dijo anteriormente, esto puede conllevar a un ciclo de repetición del error que asiente la frustración de no encontrar un error en el código y reduce el ejercicio creativo de hacer “una simulación”, por lo que se insiste en realizar descansos en la codificación y lectura del texto.

4.1. Contexto de la simulación-ejemplo: Radioterapia con acelerador lineal.

La radioterapia¹ ha sido utilizada para combatir el cáncer por más de un siglo (AAAS, 1957). El primer informe acerca de tratamientos con radioterapia data de 1899, poco después de que Röntgen descubre los rayos X , y de que la pareja Curie descubriera el elemento radio.

La evolución de las técnicas experimentales y de herramientas computacionales, con aplicaciones a la radioterapia, hacen de ésta un campo “fructífero” para la simulación de fenómenos de interacción radiación-materia. Con el fin de examinar el comportamiento de la radiación y sus consecuencias sobre tejidos biológicos, la simulación computacional provee una poderosa herramienta para la detección y tratamiento de enfermedades, como es el caso del cáncer.

Los aceleradores lineales médicos (LINAC), empleados en radio-cirugías (un procedimiento médico de radioterapia en que se administran haces finos de radiación)², son dispositivos que aceleran partículas cargadas, específicamente electrones, a energías comprendidas entre 4 y 25 MeV a través de un tubo lineal, conocido como guía de ondas. Se debe tener en cuenta que el haz de electrones, producido por el LINAC, puede usarse en sí mismo, o puede hacerse incidir sobre un blanco (generalmente de tungsteno), para producir rayos X . En general, la linea del haz de un acelerador lineal se compone fundamentalmente de cinco partes:

- Una fuente de radiación.

¹Forma de tratamiento basada en el empleo de radiaciones ionizantes (rayos X , rayos γ , partículas alfa, etc).

²Debido a la fiabilidad, flexibilidad y precisión del haz de radiación producido por el acelerador, las terapias basadas en cobalto (por emisión radiactiva), realizadas con bisturí de rayos gamma (Gamma Knife) (Thwaites y cols., 2006) han venido siendo reemplazadas.

- una cavidad aceleradora
- un sistema de colimación
- un blanco
- sistema de detección.

4.2. Necesidad de la simulación-ejemplo

Se requiere entonces estudiar la interacción en una radioterapia por acelerador lineal aplicada con haz de electrones sobre una sección de cabeza humana (blanco o absorbente); en particular, se tiene la necesidad de proporcionar información acerca de la deposición de energía en el absorbente y el comportamiento del poder de frenado del mismo.

4.3. Modelo de simulación-ejemplo

Una simulación corresponde a un proceso creativo que requiere un conocimiento intelectual. Surge entonces la pregunta: se puede simular una sección de la cabeza de un ser humano haciendo uso de las herramientas de *GEANT4* para estudiar la deposición de energía y frenado del blanco ante un haz de electrones producido por un LINAC?

En este ejercicio de modelamiento podemos diferenciar dos sistemas: uno real, el cual corresponde a un LINAC real, con un paciente real, y el haz de electrones incidiendo sobre una sección de la cabeza del paciente; y otro, un sistema físico, que se corresponde a una abstracción de ese sistema real, si se prefiere, un sistema simplificado de dicha realidad; para le caso, el modelo de simulación puede sustraer la cavidad de aceleración y simplemente suponer un haz de electrones, el cual incide en un sistema de colimación, que a su vez hace un haz mas fino y localizado en el espacio, dirigido a un blanco o absorbente particular.

Se puede pensar inicialmente en complejizar el modelo de simulación a partir del modelamiento de la instrumentación a utilizar y del blanco. Para ello se puede suponer el colimador como un cilindro con un agujero en forma de cono truncado, y la sección de la cabeza humana, como tres capas adyacentes en forma de paralelepípedos rectos de áreas iguales pero espesores diferentes, tal como describe la Fig. 4.1.

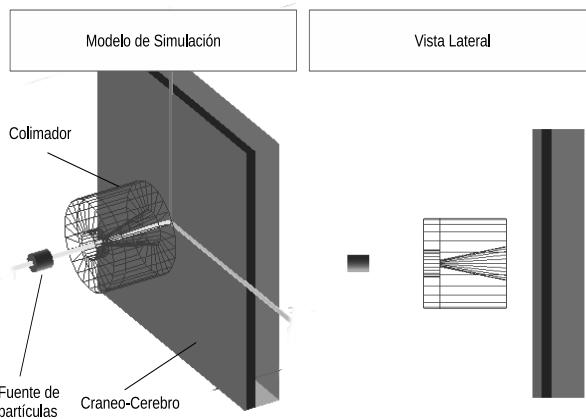


Figura 4.1: Modelo se simulación (el dibujo no esta a escala real). El panel superior izquierdo muestra un esquema de vista lateral de la geometría de la simulación.

Las condiciones iniciales de una simulación que abstrae una situación real debe cohonestar con la consulta sobre valores medidas propios de la misma; para esta simulación-ejemplo se debe tener en cuenta, por ejemplo, valores reportados en la literatura sobre piel y músculo, cráneo, y, el valor de profundidad que se quiere estudiar. En general, valores medidas y conocimientos previos son requeridos para complejizar el modelo físico y solventar las necesidades de estudio de la situación que se quiere simular.

4.4. Codificación de la simulación-ejemplo

Dado que los fines de la simulación son los de proporcionar información acerca de la deposición de energía en el blanco y del comportamiento del poder de frenado en el mismo, se requiere estructurar la codificación de la simulación alineada al modelo de simulación y a sus objetivos.

En síntesis, la simulación que ha preparado este recurso educativo orientado a profesores-investigadores sobre el uso y manejo de **GEANT4** está constituida por una serie de códigos, en los que se implementan algunas de las clases relevantes de **GEANT4**, examinadas en la sección anterior.

Importante: Es de altísima relevancia que el lector reitere la lectura del capítulo anterior para generar un lenguaje propio de un diálogo en alfabetización científica y tecnológica

entre los autores de este recurso educativo, el lector y el software (**GEANT4**) y derive en el éxito del desarrollo de esta simulación ejemplo por parte del lector.

Como estructura principal de la codificación de la simulación-ejemplo se pensó en hacer varios códigos: un código que simule el haz de electrones que incide sobre el colimador, otro código que defina la geometría de la simulación y un tercer código que defina los procesos físicos asociados a la interacción radiación-materia, acorde a los requerimientos mínimos para una simulación en **GEANT4**:

- Geometría del detector
- Procesos físicos en los cuales se está interesado
- Tipo o tipos de partículas primarias producidas para la simulación

lo que hace necesario codificar un programa principal que llame a los anteriores y un código opcional de visualización (que para este recurso resulta imperativo puesto que se quiere acortar la curva de aprendizaje de **GEANT4**) que permita, además, guardar la información suministrada por la simulación; estos códigos son:

- **AcLinealCerebro.cc**: programa principal que contiene el `main()` y el llamado a las clases empleadas en la simulación.
- **AcLinealConstruccion.cc**, **AcLinealConstruccion.hh**: archivo de geometría física y lógica del experimento, con su respectiva cabecera.
- **AcLinealFuente.cc**, **AcLinealFuente.hh**: archivos (fuente y cabecera), en los que se definen la fuente de partículas (electrones) y sus propiedades.
- **AcLinealListaFisica.cc**, **AcLinealListaFisica.hh**: archivos en los que se definen los procesos físicos asociados a la interacción de las partículas.
- **vis_BrainExp.mac**: archivo opcional, que contiene un *script* de ejecución con visualización de la simulación.

En total, este recurso educativo para esta simulación-ejemplo diseña 9 códigos 7 códigos correspondientes a archivos `.cc` y `.hh`, 1 archivo de configuración (`CMakeLists.txt`) y un archivo de visualización y ejecución de la simulación (`vis_BrainExp.mac`), a diferencia del ejemplo más básico en **GEANT4**, que corresponde a un total de 19 códigos, 12 correspondientes a archivos `.cc` y `.hh`, y otros 7 correspondientes a la visualización y ejecución de la simulación.

Una simulación-ejemplo: GEANT4 hands on!

En una estructura de archivos para desarrollo de software, los archivos con extensión .cc se alojan en un directorio **src** (códigos fuente), mientras que las cabeceras ó librerías, con extensión .hh, se consolidan en un uno llamado **include**, estos directorios se encuentran a su vez en un directorio “principal”, que para esta simulación de ha llamado **BrainExp** (ver Fig. 4.2).

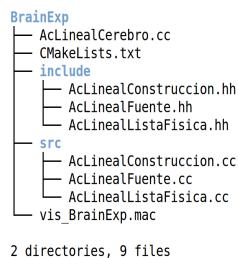


Figura 4.2: Estructura de archivos de la simulación-ejemplo. Los archivos CMakeLists.txt y vis_BrainExp.mac se describen mas adelante.

4.4.1. Geometría y materiales

La geometría del acelerador lineal médico se construyó de acuerdo con la Fig. 4.3, empleando la clase G4VUserDetectorConstruction.

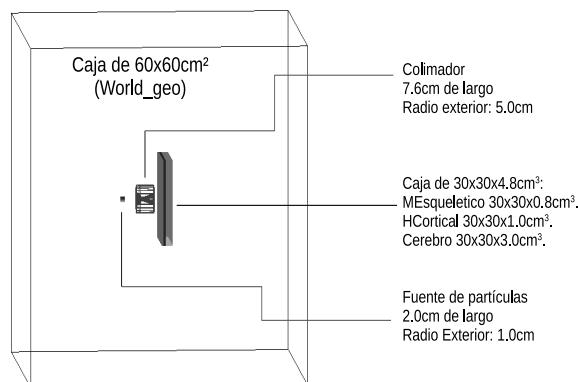


Figura 4.3: Geometría de la simulación-ejemplo.

En esta clase se definen parámetros geométricos utilizando volúmenes que simulan las diferentes geometrías involucradas en la simulación. De esta manera se tiene un volumen asociado a la fuente de partículas, al colimador y al blanco.

La fuente de partículas es una asociación “virtual”, es decir, no es un volumen sensible que interactúe de manera alguna con las partículas, al que se le ha asociado un volumen físico, en este caso un cilindro recto circular de tungsteno con radio 1.0 cm y largo 2.0 cm.

Se aclara que el tungsteno resulta ser un material arbitrario, pues no afecta en nada la producción de haz de electrones, pudiéndose cambiar por otro material. La definición de los materiales se realiza utilizando la base de datos del NIST (ver Apéndice G). La producción del haz se especifica en el archivo `AcLinealFuente`, y no en el archivo de geometría.

De manera similar, y a una distancia $s_1 = 5.0$ cm, se construye el colimador, un cono truncado de radios internos de 1.5 cm, 0.38 cm y 1.9 cm, como ilustra la Fig. 4.4.

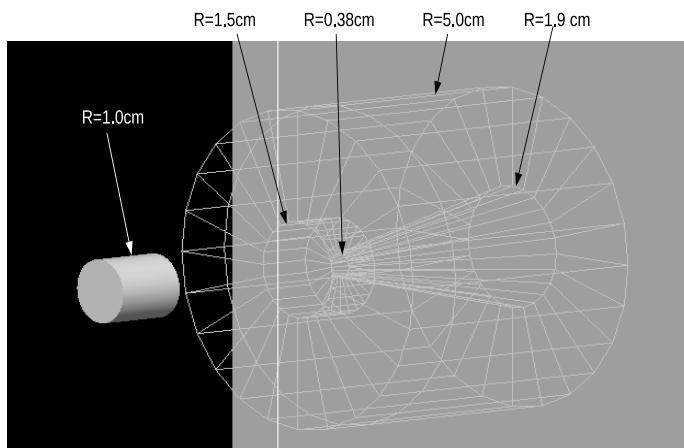


Figura 4.4: Configuración geométrica de la fuente (cilindro recto) y colimador (cilindro recto exterior, cilindro recto y cono truncado interiores).

En contraste al material de construcción de la fuente, se elige tungsteno para el material del colimador. En la práctica, este material parece poseer características adecuadas para aplicaciones de protección radiológica, tanto en entornos médicos como industriales; además, una aleación de alta densidad de tungsteno proporciona la misma absorción de

energía que el plomo, utilizando 1/3 menos de material (Rodríguez y Martínez, 1995).

La función del colimador es la de enfocar los haces de electrones hacia el campo de irradiación del absorbente, debido a que los haces de electrones se dispersan con facilidad en aire, razón por la cual el colimador debe estar relativamente cerca de la superficie del blanco, en este caso a una distancia $s_2 = 2.4$ cm. En una aplicación más general se pueden implementar colimadores auxiliares para reducir la dispersión de electrones provenientes del colimador primario.

Finalmente el absorbente es diseñado como tres paralelepípedos rectangulares contiguos, cuya sección transversal es de 30×30 cm². El primer paralelepípedo simula piel humana y músculo de 0.8 cm de espesor, el segundo simula hueso craneal de 1.0 cm de espesor, y el tercero corteza cerebral de 3.0 cm de espesor, quedando cada geometría ubicada como ilustra la Fig 4.5. Note que estos valores, característicos de un humano promedio, fueron tomados de la literatura ((Nieto, Minor, Álvarez, Alonso, y Carbajal, 2007)), mientras que, las propiedades de los materiales para el blanco corresponden a la base de datos del NIST, con valores del ICRP (*International Commission on Radiological Protection*) (ICRP, 2011).

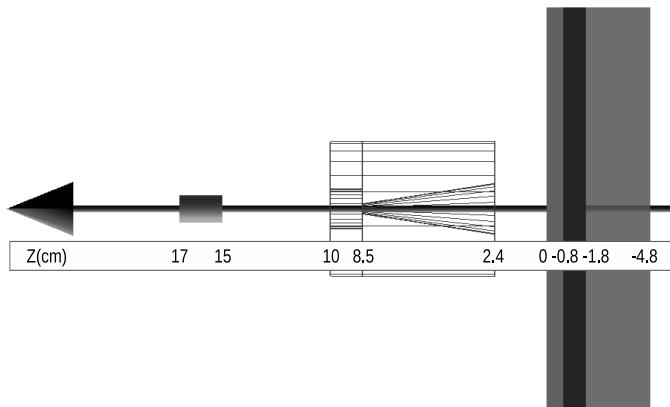


Figura 4.5: Ubicación axial de la configuración geométrica de la simulación-ejemplo.

A cada volumen de la geometría se le asocian tres objetos, cada uno pertenece a una clase determinada. El primer objeto define las dimensiones y la forma del volumen

con base en la clase de la cual se deriva, por ejemplo G4Box, G4Tubs, G4Cones, etc. Una vez diseñado el volumen físico G4VPhysicalVolume se asocia el volumen lógico, el cual proviene de la clase G4LogicalVolume, en el que se definen los materiales empleados en cada volumen.

Para crear un volumen lógico es necesario haber definido un volumen físico y a su vez, es necesario tener un volumen lógico para crear un volumen físico del cual se genera el objeto restante. Este último objeto define la rotación del volumen original, el número de copias que se van a realizar del volumen y la posición respecto al volumen madre, que en este caso coincide con el *world volume*, el cual se ha definido como una caja, a la que se le ha llamado `World_geo`, cuyas dimensiones son $60.0 \times 60.0 \times 60.0 \text{ cm}^3$, y contiene todos los elementos de la geometría (ver Fig. 4.3).

Para visualizar la geometría se deben asignar parámetros de visualización a los volúmenes que se desean observar. Primero se define un apuntador a un objeto de la clase G4VisAttributes, la cual permite definir las características requeridas; luego, mediante la función SetVisAttributes, el apuntador se asocia a la visualización del volumen lógico correspondiente, pudiéndose definir escalas, marcas, colores, texto flotante, entre otros aspectos.

De esta manera se diseñaron los archivos `AcLinealConstruccion.cc` y `AcLinealConstruccion.hh` congruentes a la construcción geométrica de la simulación, que son ilustrados en los siguientes dos recuadros:

```
1 // ----- AcLinealConstruccion.hh -----
2 #include "G4SystemOfUnits.hh"
3 #ifndef AcLinealConstruccion_hh
4 #define AcLinealConstruccion_hh
5 #include "G4VUserDetectorConstruction.hh"
6
7 class G4VPhysicalVolume;
8 class G4NistManager;
9 class AcLinealConstruccion : public G4VUserDetectorConstruction{
10
11 public:
12     AcLinealConstruccion();
13     ~AcLinealConstruccion();
14
15     G4VPhysicalVolume* Construct();
16     void Construct_Fuente (G4LogicalVolume* eumma, G4double posz);
17     void Construct_Colimador(G4LogicalVolume* eumma, G4double posz);
18     void Construct_MEsquelitico(G4LogicalVolume* eumma, G4double posz);
19     void Construct_HCortical(G4LogicalVolume* eumma, G4double posz);
20     void Construct_Cerebro (G4LogicalVolume* eumma, G4double posz);
```

Una simulación-ejemplo: GEANT4 hands on!

```
21
22 private:
23   G4NistManager *NIST;
24 };
25 #endif

1 //----- AcLinealConstruccion.cc -----
2 #include "G4Material.hh"
3 #include "G4NistManager.hh"
4 #include "G4LogicalVolume.hh"
5 #include "G4ThreeVector.hh"
6 #include "G4PVPlacement.hh"
7 #include "G4Box.hh"
8 #include "G4Tubs.hh"
9 #include "G4Polycone.hh"
10 #include "G4VisAttributes.hh"
11 #include "G4Colour.hh"
12 #include "AcLinealConstruccion.hh"
13 #include "G4SystemOfUnits.hh"
14
15 AcLinealConstruccion::AcLinealConstruccion(){NIST=G4NistManager::Instance();}
16 AcLinealConstruccion::~AcLinealConstruccion(){}
17
18 G4VPhysicalVolume* AcLinealConstruccion::Construct(){
19   G4Material* Aire = NIST->FindOrBuildMaterial("G4_AIR");
20
21   G4Box           * World_geo = new G4Box("World_g", 60.0*cm, 60.0*cm, 60.0*cm);
22   G4LogicalVolume * World_log = new G4LogicalVolume(World_geo,Aire,"World_l",0,0,0);
23   G4VPhysicalVolume * World_phys = new G4PVPlacement(0,G4ThreeVector(),World_log,"World", 0, false, 0);
24
25   Construct_Fuente(World_log, 15.0*cm);
26   Construct_Colimador(World_log, 10.0*cm);
27   Construct_Cerebro(World_log, -6.0*cm);
28   Construct_MEsquelitico(World_log, -0.7*cm);
29   Construct_HCortical(World_log, -2.0*cm);
30   return World_phys;
31 }
32 //----- Fuente de Radiacion -----
33 void AcLinealConstruccion::Construct_Fuente(G4LogicalVolume* mother, G4double posz){
34   const G4double WFuente_Espesor = 2.0*cm;
35   const G4double WFuente_pos    = posz-(WFuente_Espesor)/2.0+2.0*cm;
36
37   G4Material* Tungsteno = NIST->FindOrBuildMaterial("G4_AIR");
38   G4VisAttributes* W_Vis = new G4VisAttributes(G4Colour(1.0, 0.843, 0.161, 1.0));
39   W_Vis -> SetForceSolid(true);
40   G4Tubs* WFuente_geo = new G4Tubs("WFuente_g",0.0*cm,1.0*cm,WFuente_Espesor/2.0,0.0*deg,360.0*deg);
41   G4LogicalVolume* WFuente_log = new G4LogicalVolume(WFuente_geo, Tungsteno, "WFuente_l");
42   WFuente_log -> SetVisAttributes(W_Vis);
43   new G4PVPlacement(0, G4ThreeVector(0,0, WFuente_pos), WFuente_log, "WFuente", mother, false, 0 );
44 }
45 //----- Colimador -----
46 void AcLinealConstruccion::Construct_Colimador(G4LogicalVolume* mother, G4double posz){
47   const G4int nZ = 4 ;
48   const G4double rIn[] = {1.5*cm, 1.5*cm, 0.38*cm, 1.9*cm};
49   const G4double rOut[] = {5*cm, 5*cm, 5*cm, 5*cm};
```

Una simulación-ejemplo: GEANT4 hands on!

```
50 const G4double ZColi[] = {1.5*cm, 6.1*cm};  
51 const G4double Z[]      = {0.0, -ZColi[0], -ZColi[0], -(ZColi[0]+ZColi[1])};  
52  
53 G4Material* Tungsteno = NIST->FindOrBuildMaterial("G4_W");  
54  
55 G4Polycone* Col_geo = new G4Polycone("Col_g", 0*deg, 360*deg, nZ, Z, rIn, rOut);  
56 G4LogicalVolume* Col_log = new G4LogicalVolume(Col_geo, Tungsteno, "Col_l", 0, 0, 0);  
57 new G4PVPlacement(0, G4ThreeVector(0.0, posz), Col_log, "Col", mother, false, 0);  
58  
59 G4VisAttributes* VAttW = new G4VisAttributes(G4Colour(0.75, 0.76, 0.8, 1.0));  
60 VAttW -> SetForceAuxEdgeVisible(true);  
61 Col_log -> SetVisAttributes(VAttW);  
62 }  
63 //----- Musculo Esqueletico -----  
64 void AcLinealConstruccion::Construct_MEsquelitico(G4LogicalVolume* mother, G4double posz){  
65 //("G4_MUSCLE_SKELTAL_ICRP");  
66 G4Material* Musculo = G4NistManager::Instance()->FindOrBuildMaterial("G4_SKIN_ICRP");  
67 G4cout << Musculo << G4endl;  
68  
69 G4double musculo_tam=30.0*cm;  
70  
71 G4Box* ME_geo = new G4Box("BP_g", musculo_tam/2.0, musculo_tam/2.0, musculo_tam/100.0);  
72 G4LogicalVolume* ME_log = new G4LogicalVolume(ME_geo, Musculo, "ME_l", 0, 0, 0);  
73 new G4PVPlacement(0, G4ThreeVector(0.0*cm, 0.0*cm, posz), ME_log, "MEsquelitico", mother, false, 0);  
74  
75 G4VisAttributes* VAttMusculo = new G4VisAttributes(G4Colour(1.0, 0.6, 0.6));  
76 VAttMusculo->SetForceSolid(true);  
77 ME_log->SetVisAttributes(VAttMusculo);  
78 }  
79 //----- Hueso cortical -----  
80 void AcLinealConstruccion::Construct_HCortical(G4LogicalVolume* mother, G4double posz){  
81 G4Material* Hueso = G4NistManager::Instance()->FindOrBuildMaterial("G4_BONE_CORTICAL_ICRP");  
82 G4cout << Hueso << G4endl;  
83  
84 G4double hueso_tam=30.0*cm;  
85  
86 G4Box* HC_geo = new G4Box("HC_g", hueso_tam/2.0, hueso_tam/2.0, hueso_tam/30.0);  
87 G4LogicalVolume* HC_log = new G4LogicalVolume(HC_geo, Hueso, "HC_l", 0, 0, 0);  
88 new G4PVPlacement(0, G4ThreeVector(0.0*cm, 0.0*cm, posz), HC_log, "HCortical", mother, false, 0);  
89  
90 G4VisAttributes* VAttHueso = new G4VisAttributes(G4Colour(1.0, 1.0, 1.0));  
91 VAttHueso->SetForceSolid(true);  
92 HC_log->SetVisAttributes(VAttHueso);  
93 }  
94 //----- Cerebro -----  
95 void AcLinealConstruccion::Construct_Cerebro(G4LogicalVolume* mother, G4double posz){  
96  
97 G4Material* Cerebro = G4NistManager::Instance()->FindOrBuildMaterial("G4 BRAIN_ICRP");  
98 G4cout << Cerebro << G4endl;  
99  
100 G4double cerebro_tam=30.0*cm;  
101  
102 G4Box* Cereb_geo = new G4Box("Cereb_g", cerebro_tam/2.0, cerebro_tam/2.0, cerebro_tam/10.0);  
103 G4LogicalVolume* Cereb_log = new G4LogicalVolume(Cereb_geo, Cerebro, "Cereb_l", 0, 0, 0);  
104 new G4PVPlacement(0, G4ThreeVector(0.0*cm, 0.0*cm, posz), Cereb_log, "Cerebro", mother, false, 0);  
105 }
```

```
106 G4VisAttributes* VAttCerebro = new G4VisAttributes(G4Colour(0.75, 0.76, 0.8, 0.8));  
107 VAttCerebro->SetForceSolid(true);  
108 Cereb_log->SetVisAttributes(VAttCerebro);  
109 }
```

4.4.2. Definición de la fuente de partículas y del evento primario

La generación de partículas primarias se realiza a través de la clase `G4ParticleGun`; ésta permite definir un haz de radiación así como el estado inicial del evento primario, esto es, especificando las características de la fuente de partículas, como su posición y el tipo de partículas a emplear.

La posición, dirección del momento lineal y energía de las partículas primarias involucradas, deben ser fijadas en el código fuente, de acuerdo con la clase base `G4VUserPrimaryGeneratorAction`, de lo contrario asumirán el valor de cero.

El evento primario consiste en disparar electrones, con energía promedio de 6 MeV hacia el blanco, en dirección negativa al eje Z.

En el archivo de cabecera, que define la fuente de partículas mediante las clases `G4Event` y `G4VUserPrimaryGeneratorAction`, se establece un único evento primario, correspondiente a una sola “especie” de partículas, a la que se le ha llamado `particleGun`, y que corresponde a electrones, donde se ha utilizando la clase `G4ParticleTable.hh` que permite ubicar las características particulares a este tipo de partículas.

También se especifican otros parámetros como la posición de generación del evento primario dentro del volumen madre, energía, y dirección de propagación del haz. Los códigos respectivos a la cabecera y al archivo fuente (con extensión .cc) se muestran a continuación:

```
1 //----- AcLinealFuente.hh -----  
2 #ifndef AcLinealFuente_hh  
3 #define AcLinealFuente_hh  
4 #include "G4VUserPrimaryGeneratorAction.hh"  
5 #include "G4ParticleGun.hh"  
6 #include "G4SystemOfUnits.hh"  
7  
8 class G4Event ;  
9 class AcLinealFuente: public G4VUserPrimaryGeneratorAction{  
10  
11 public:  
12     AcLinealFuente();  
13     ~AcLinealFuente();
```

```
14     void GeneratePrimaries(G4Event*);  
15  
16 private:  
17     G4ParticleGun* particleGun;  
18 };  
20 #endif
```

```
1 //----- AcLinealFuente.cc -----  
2  
3 #include "G4Event.hh"  
4 #include "G4ParticleGun.hh"  
5 #include "G4ParticleTable.hh"  
6 #include "AcLinealFuente.hh"  
7 #include "G4SystemOfUnits.hh"  
8  
9 AcLinealFuente::AcLinealFuente(){  
10     particleGun = new G4ParticleGun(1);  
11     G4ParticleDefinition* particle = G4ParticleTable::GetParticleTable()->FindParticle("e-");  
12     particleGun -> SetParticleDefinition(particle);  
13     particleGun -> SetParticlePosition( G4ThreeVector(0.0, 0.0, 16*cm));  
14     particleGun -> SetParticleMomentumDirection(G4ThreeVector(0.0, 0.0, -1.0));  
15     particleGun -> SetParticleEnergy(6.0*MeV);  
16 }  
17  
18 AcLinealFuente::~AcLinealFuente(){;}  
19 void AcLinealFuente::GeneratePrimaries(G4Event* anEvent ){  
20     particleGun -> GeneratePrimaryVertex(anEvent);  
21 }
```

4.4.3. Procesos físicos asociados a las interacciones radiación-materia

A partir de la clase `G4VUserPhysicsList` se establece lo concerniente a los fenómenos físicos que tienen cabida en la simulación, incluyendo las partículas definidas en la subsección anterior y los procesos físicos a los cuales se hizo referencia en el Capítulo 2.

Las partículas empleadas en la simulación son electrones (`particleGun`), pero con posibilidad de interactuar y decaer por lo que se deben emplear también positrones, rayos γ , e iones. En particular, se ha utilizado la clase `G4EmStandardPhysics` la cual considera exclusivamente procesos electromagnéticos en los que se incluyen tres mecanismos de interacción con la materia (efecto fotoeléctrico, dispersión Compton, producción de pares), y los procesos de ionización, absorción y radiación de frenado; esto es posible gracias a la implementación de las clases `G4PhotoElectricEffect`, `G4ComptonScattering`, `G4Ionisation`, `G4Bremsstrahlung`, `G4PairProduction` y `G4plusAnnihilation` en `G4EmStandardPhysics`. Se incluye también, la posibilidad de que haya transporte, es decir, que no haya interacción alguna, mediante la

Una simulación-ejemplo: GEANT4 hands on!

opción AddTransportation() de la clase G4VUserPhysicsList.

Se establecen los códigos respectivos a la cabecera y al archivo fuente (con extensión .cc) los cuales se ilustran en los siguientes dos recuadros (AcLinealListaFisica.hh y AcLinealListaFisica.cc):

```
1 //----- AcLinealListaFisica.hh -----
2 #ifndef AcLinealListaFisica_hh
3 #define AcLinealListaFisica_hh
4 #include "G4VModularPhysicsList.hh"
5 #include "globals.hh"
6
7 class AcLinealListaFisica: public G4VModularPhysicsList{
8
9 public:
10    AcLinealListaFisica();
11    virtual ~AcLinealListaFisica();
12    void ConstructParticle();
13    void ConstructProcess();
14    void SetCuts();
15
16 private:
17    G4VPhysicsConstructor* FisicaEM;
18
19 };
20 #endif
```

```
1 //----- AcLinealListaFisica.cc -----
2 #include "G4eIonisation.hh"
3 #include "G4hIonisation.hh"
4 #include "G4IonConstructor.hh"
5 #include "G4EmStandardPhysics.hh"
6 #include "G4EmLivermorePhysics.hh"
7 #include "G4EmPenelopePhysics.hh"
8 #include "G4DecayPhysics.hh"
9 #include "G4Decay.hh"
10 #include "AcLinealListaFisica.hh"
11 #include "G4SystemOfUnits.hh"
12
13 AcLinealListaFisica::AcLinealListaFisica(): G4VModularPhysicsList(){
14    FisicaEM = new G4EmStandardPhysics(1);
15 }
16 AcLinealListaFisica::~AcLinealListaFisica(){}
17 void AcLinealListaFisica::ConstructParticle(){
18    FisicaEM -> ConstructParticle();
19    G4IonConstructor ions;
20    ions.ConstructParticle();
21 }
```

```
22 void AcLinealListaFisica::ConstructProcess(){
23
24     AddTransportation();
25     FisicaEM -> ConstructProcess();
26 }
27
28 void AcLinealListaFisica::SetCuts(){
29
30     defaultCutValue = 0.1*mm;
31     G4double cutForGamma      = defaultCutValue;
32     G4double cutForElectron   = defaultCutValue;
33     G4double cutForPositron  = defaultCutValue;
34     SetCutValue(cutForGamma, "gamma");
35     SetCutValue(cutForElectron, "e-");
36     SetCutValue(cutForPositron, "e+");
37     DumpCutValuesTable();
38 }
```

donde se programan tres aspectos importantes en la simulación:

1. Partículas involucradas en la simulación, tanto genera la fuente primaria de radiación (partículas primarias: electrones) como las que se crean a partir de la interacción de las partículas primarias con los volúmenes definidos (partículas secundarias: positrones, γ).
2. Procesos físicos a considerar en la simulación, como se mencionó en el precedente párrafo.
3. `SetCut()`, una función que asocia un valor de energía a radiaciones γ , electrones y positrones que impide a **GEANT4** la generación de partículas secundarias, resultado de la interacción de dicha radiación, a energías menores o iguales a las establecidas. Cuando se ingresa una distancia a esta función, **GEANT4** asocia la energía necesaria para que determinada partícula tenga un rango igual a la distancia ingresada con base en el medio en el que esté.

4.5. Compilación y ejecución de la simulación-ejemplo

4.5.1. Función `main()`: Código principal `AcLinearCerebro.cc`

Con los códigos necesarios, se requiere elaborar un código que contenga una función `main()` que permitirá la compilación y ejecución de la simulación-ejemplo. En este sentido, se elabora el código `AcLinearCerebro.cc` que invoca los demás códigos al interior de una función principal o función `main()`, tal como se ilustra en el siguiente cuadro (ver líneas

6, 7 y 8: cabeceras .hh; líneas 24, 25 y 26: clases correspondientes a archivos fuente con extensión .cc);

```
1 //----- main(): AcLinealCerebro.cc -----
2 #include "G4RunManager.hh"
3 #include "G4UImanager.hh"
4 #include "G4ScoringManager.hh"
5 #include "G4UITerminal.hh"
6 #include "AcLinealConstruccion.hh"
7 #include "AcLinealListaFisica.hh"
8 #include "AcLinealFuente.hh"
9 #include "G4UIExecutive.hh"
10 #include "G4SystemOfUnits.hh"
11 #ifdef G4VIS_USE
12 #include "G4VisExecutive.hh"
13#endif
14
15 int main(int argc, char** argv){
16    G4RunManager *runManager = new G4RunManager;
17    G4ScoringManager* scoringManager = G4ScoringManager::GetScoringManager();
18    scoringManager->SetVerboseLevel(1);
19 #ifdef G4VIS_USE
20    G4VisManager* visManager = new G4VisExecutive;
21    visManager->Initialize();
22#endif
23
24    runManager -> SetUserInitialization(new AcLinealConstruccion);
25    runManager -> SetUserInitialization(new AcLinealListaFisica);
26    runManager -> SetUserAction(new AcLinealFuente);
27
28    runManager -> Initialize();
29    if(argc==1){
30 #ifdef G4UI_USE
31        G4UIExecutive* ui = new G4UIExecutive(argc, argv);
32        ui->SessionStart();
33        delete ui;
34#endif
35    }
36    else{
37        G4UImanager* UImanager = G4UImanager::GetUIpointer();
38        G4String command = "/control/execute ";
39        G4String fileName = argv[1];
40        UImanager->ApplyCommand(command+fileName);
41    }
42
43 #ifdef G4VIS_USE
44    delete visManager;
45#endif
```

```
46 delete runManager;  
47 return 0;  
48 }
```

La función `main()` es el cuerpo del programa o el programa principal, ya que es la primera función que el enlazador busca para ejecutar; en ésta se hace el llamado a las clases previamente definidas, de allí que se haya elaborado en último lugar.

Inicialmente se invocan las cabeceras de los archivos de construcción geométrica, adición de procesos físicos y la adición de la fuente de partículas, junto con las clases `runManager` y `G4UImanager`.

Dentro de la función `main ()` se crea una instancia de la clase `G4RunManager`, la cual controla el flujo del programa y gestiona los eventos dentro de un `run`; también cuando `G4RunManager` se crea o se elimina, las demás clases principales son creadas y eliminadas respectivamente. A través de esta clase se proporciona toda la información necesaria para generar y ejecutar la simulación, incluyendo:

1. la forma en que el detector ha sido construido
2. las partículas y procesos físicos a ser simulados,
3. la producción de partículas y eventos primarios, y
4. cualquier requisito adicional de la simulación

La línea de comando `runManager->SetUserInitialization(new Objeto);` crea un objeto que especifica la geometría del detector y los procesos físicos, pasando sus punteros al gestor de ejecución (G. Cern, 2011). De manera similar se crean objetos asociados a las demás clases de inicialización, esto es, para la fuente de radiación y la adquisición de datos.

Mediante las clases `G4UImanager` y `G4String`, de la categoría `UI_GUI` de intercomunicación (ver Fig. 3.1), se dispone de funciones para llamar y establecer métodos de comunicación en la interfaz de `GEANT4`, de esta manera se ha definido el comando `/control/execute`, el cual permite ejecutar la simulación a través de un macro de visualización.

4.5.2. Modificación del archivo `CMakeLists.txt` y archivos de configuración

El Apéndice F, sección F.3, ilustra como se compila y ejecuta uno de los ejemplos básicos, el ejemplo B1, por lo que se sugiere al lector hacer una lectura cuidadosa de

Una simulación-ejemplo: GEANT4 hands on!

dicho Apéndice para afrontar esta sección.

Para el trabajo práctico, propio de la simulación-ejemplo, la instalación de **GEANT4** define por defecto variables de entorno requeridas por el sistema operativo para hacer la compilación y ejecución de cualquier simulación de **GEANT4**. Se debe editar entonces el archivo `.bashrc`, agregando dos fuentes de archivos de configuración propios de **GEANT4**:

```
1 source /home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/geant4make/geant4make.sh
2 source /home/fisinfor/geant4/geant4.10.04.p02-install/bin/geant4.sh
```

las cuales le permiten al sistema operativo enlazar una simulación con los recursos de **GEANT4**. La línea de comando `printenv | grep "G4"`, realiza una lista de las variables de entorno propias de la instalación de **GEANT4**:

```
1 fisinfor@taozen:~$ printenv | grep "G4"
2 G4LEVELGAMMADATA=/home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/data/PhotonEvaporation5.2
3 G4INSTALL=/home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/geant4make
4 G4INCLUDE=/home/fisinfor/geant4/geant4.10.04.p02-install/include/Geant4
5 G4NEUTRONXSDATA=/home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/data/G4NEUTRONXS1.4
6 G4LEDATA=/home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/data/G4EMLOW7.3
7 G4VIS_USE_OPENGLQT=1
8 G4NEUTRONHPDATA=/home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/data/G4NDL4.5
9 G4ENSDFSTATEDATA=/home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/data/G4ENSDFSTATE2.2
10 G4RADIOACTIVEDECYDATA=/home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/data/RadioactiveDecay5.2
11 G4ABLADATA=/home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/data/G4ABLA3.1
12 G4LIB_USE_GDML=1
13 G4LIB=/home/fisinfor/geant4/geant4.10.04.p02-install/lib/Geant4-10.4.2
14 G4PIDATA=/home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/data/G4PII1.3
15 G4LIB_BUILD_SHARED=1
16 G4SAIDXSDATA=/home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/data/G4SAIDDATA1.1
17 G4VIS_USE_OPENGLX=1
18 G4UI_USE_TCSH=1
19 G4VIS_USE_RAYTRACERX=1
20 G4REALSURFACE DATA=/home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/data/RealSurface2.1.1
21 G4SYSTEM-Linux-g++
22 G4UL_USE_QT=1
23 G4WORKDIR=/home/fisinfor/geant4_geant4_workdir
24 G4LIB_USE_ZLIB=1
```

La línea 23 indica el nombre que debe llevar el directorio de trabajo donde el usuario debe codificar y compilar sus simulaciones: `geant4_workdir`. Este directorio se debe crear dentro del directorio del usuario³.

Al interior de este directorio se debe mover el directorio `BrainExp` y adicionar dos códigos mas: `CMakeList.txt` y `vis_BrainExp.mac`, y crear un directorio de instalación y ejecución de la simulación que se llame `BrainExp_build`, quedando el árbol del directorio como ilustra el siguiente recuadro:

```
1 fisinfor@taozen:~$ pwd
2 /home/fisinfor
```

³Esta carpeta se realiza bajo la línea de comando `mkdir ~/geant4_workdir`

```
3 fisinfor@taozen:~$ cd geant4_workdir1/
4 fisinfor@taozen:~/geant4_workdir$ ls
5 BrainExp
6 fisinfor@taozen:~/geant4_workdir$ cd BrainExp/
7 fisinfor@taozen:~/geant4_workdir/BrainExp$ ls -1F
8 AcLinealCerebro.cc
9 BrainExp_build/
10 CMakeLists.txt
11 include/
12 src/
13 vis_BrainExp.mac
14 fisinfor@taozen:~/geant4_workdir/BrainExp$ ls -1FR
15 ..
16 AcLinealCerebro.cc
17 BrainExp_build/
18 CMakeLists.txt
19 include/
20 src/
21 vis_BrainExp.mac
22
23 ./BrainExp_build:
24
25 ./include:
26 AcLinealConstruccion.hh
27 AcLinealFuente.hh
28 AcLinealListaFisica.hh
29
30 ./src:
31 AcLinealConstruccion.cc
32 AcLinealFuente.cc
33 AcLinealListaFisica.cc
34 fisinfor@taozen:~/geant4_workdir/BrainExp$
```

Actualmente, *GEANT4* utiliza un método de compilación que requiere un archivo de configuración llamado `CMakeLists.txt`, el cual es llamado por la utilidad `cmake` para preparar la compilación de la simulación a través de la utilidad `make`.

Para efecto del trabajo “hands on”, la compilación y ejecución de la simulación-ejemplo, y en general, de cualquier simulación en *GEANT4*, se requiere un archivo `CMakeLists.txt`, que prepara los requerimientos necesarios para obtener un

archivo ejecutable propio del proyecto de simulación. Este archivo debe ser editado acorde a los requerimientos de la simulación-ejemplo, por lo que se realiza una copia de este archivo CMakeLists.txt del ejemplo B1 (ver Apéndice F, Sección F.3) y se realizan los cambios pertinentes, quedando el archivo CMakeLists.txt con las especificaciones de la simulación-ejemplo así⁴:

```
1 # $Id: CMakeLists.txt 86065 2014-11-07 08:51:15Z gcosmo $
2
3 #-----
4 # Setup the project
5 cmake_minimum_required(VERSION 2.6 FATAL_ERROR)
6 project(BrainExp)
7
8 #-----
9 # Find Geant4 package, activating all available UI and Vis drivers by default
10 # You can set WITH_GEANT4_UIVIS to OFF via the command line or ccmake/cmake-gui
11 # to build a batch mode only executable
12 #
13 option(WITH_GEANT4_UIVIS "Build example with Geant4 UI and Vis drivers" ON)
14 if(WITH_GEANT4_UIVIS)
15   find_package(Geant4 REQUIRED ui_all vis_all)
16 else()
17   find_package(Geant4 REQUIRED)
18 endif()
19
20 #-----
21 # Setup Geant4 include directories and compile definitions
22 # Setup include directory for this project
23 #
24 include(${Geant4_USE_FILE})
25 include_directories(${PROJECT_SOURCE_DIR}/include)
26
27
28 #-----
29 # Locate sources and headers for this project
30 # NB: headers are included so they will show up in IDEs
31 #
32 file(GLOB sources ${PROJECT_SOURCE_DIR}/src/*.cc)
33 file(GLOB headers ${PROJECT_SOURCE_DIR}/include/*.hh)
34
35 #-----
36 # Add the executable, and link it to the Geant4 libraries
```

⁴Queda para el lector revisar y comparar los cambios de estos archivos CMakeLists.txt para familiarizarse con los mismos.

```
37 #
38 add_executable(AcLinealCerebro AcLinealCerebro.cc ${sources} ${headers})
39 target_link_libraries(AcLinealCerebro ${Geant4_LIBRARIES})
40
41 #
42 # Copy all scripts to the build directory, i.e. the directory in which we
43 # build B1. This is so that we can run the executable directly because it
44 # relies on these scripts being in the current working directory.
45 #
46 set(ACLINEALCEREBRO_SCRIPTS
47 # exampleB1.in
48 # exampleB1.out
49 # init_vis.mac
50 # run1.mac
51 # run2.mac
52 # vis.mac
53 vis_BrainExp.mac
54 )
55
56 foreach(_script ${ACLINEALCEREBRO_SCRIPTS})
57 configure_file(
58 ${PROJECT_SOURCE_DIR}/${_script}
59 ${PROJECT_BINARY_DIR}/${_script}
60 COPYONLY
61 )
62 endforeach()
63
64 #
65 # For internal Geant4 use - but has no effect if you build this
66 # example standalone
67 #
68 add_custom_target(BrainExp DEPENDS AcLinealCerebro)
69
70 #
71 # Install the executable to 'bin' directory under CMAKE_INSTALL_PREFIX
72 #
73 install(TARGETS AcLinealCerebro DESTINATION bin)
```

4.5.3. Compilación y ejecución de la simulación-ejemplo

El ejercicio de compilación y ejecución de la simulación requiere saber la variable de entorno que ubica la instalación de *GEANT4*, dado que es una opción de comando requerida por la utilidad *cmake*.

Una simulación-ejemplo: GEANT4 hands on!

Con la línea de comando `printenv | grep "G4INSTALL"` se obtiene la dirección de la instalación de *GEANT4* en el sistema operativo:

```
G4INSTALL=/home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/geant4make
```

Con lo anterior, el usuario se debe ubicar en el directorio donde se realizará la compilación y ejecución de la simulación-ejemplo, que para este caso corresponde a `BrainExp_build`. El siguiente recuadro ilustra la codificación en la terminal del sistema operativo⁵:

```
1 fisinfor@taozen:~$ pwd
2 /home/fisinfor
3 fisinfor@taozen:~$ cd geant4_workdir
4 fisinfor@taozen:~/geant4_workdir$ cd BrainExp
5 fisinfor@taozen:~/geant4_workdir/BrainExp$ ls
6 AcLinealCerebro.cc BrainExp_build CMakeLists.txt include src vis_BrainExp.mac
7 fisinfor@taozen:~/geant4_workdir/BrainExp$ cd BrainExp_build/
8 fisinfor@taozen:~/geant4_workdir/BrainExp/BrainExp_build$ ls
9 fisinfor@taozen:~/geant4_workdir/BrainExp/BrainExp_build$ cmake -DCMAKE_INSTALL_PREFIX=$G4INSTALL ../
10 -- The C compiler identification is GNU 5.4.0
11 -- The CXX compiler identification is GNU 5.4.0
12 -- Check for working C compiler: /usr/bin/cc
13 -- Check for working C compiler: /usr/bin/cc -- works
14 -- Detecting C compiler ABI info
15 -- Detecting C compiler ABI info - done
16 -- Detecting C compile features
17 -- Detecting C compile features - done
18 -- Check for working CXX compiler: /usr/bin/c++
19 -- Check for working CXX compiler: /usr/bin/c++ -- works
20 -- Detecting CXX compiler ABI info
21 -- Detecting CXX compiler ABI info - done
22 -- Detecting CXX compile features
23 -- Detecting CXX compile features - done
24 -- Configuring done
25 -- Generating done
26 -- Build files have been written to: /home/fisinfor/geant4_workdir/BrainExp/BrainExp_build
27 fisinfor@taozen:~/geant4_workdir/BrainExp/BrainExp_build$ make
28 Scanning dependencies of target AcLinealCerebro
29 [ 20%] Building CXX object CMakeFiles/AcLinealCerebro.dir/AcLinealCerebro.cc.o
30 [ 40%] Building CXX object CMakeFiles/AcLinealCerebro.dir/src/AcLinealFuente.cc.o
31 [ 60%] Building CXX object CMakeFiles/AcLinealCerebro.dir/src/AcLinealConstrucion.cc.o
32 [ 80%] Building CXX object CMakeFiles/AcLinealCerebro.dir/src/AcLinealListaFisica.cc.o
33 [100%] Linking CXX executable AcLinealCerebro
```

⁵Recuerde que la utilidad `cmake` utiliza dos opciones de comando, una corresponde a la dirección de sistema operativo donde se encuentra la instalación de *GEANT4*, que en este caso corresponde a la variable de entorno `$G4INSTALL`, y la otra, la dirección de sistema operativo donde se encuentra el directorio de la simulación, que en este caso es el directorio `BrainExp`, el cual se puede alcanzar desde el directorio de instalación (`BrainExp_build`) con “`..`”.

```
35 [100%] Built target AcLinealCerebro
36 fisinfor@taozen:~/geant4_workdir/BrainExp/BrainExp_build$
```

Al final de la compilación la estructura de archivos del directorio de instalación queda así:

```
37 fisinfor@taozen:~/geant4_workdir/BrainExp/BrainExp_build$ ls -1F
38 AcLinealCerebro*
39 CMakeCache.txt
40 CMakeFiles/
41 cmake_install.cmake
42 Makefile
43 vis_BrainExp.mac
44 fisinfor@taozen:~/geant4_workdir/BrainExp/BrainExp_build$
```

donde se puede encontrar el archivo ejecutable de la simulación, en este caso corresponde a `AcLinealCerebro`.

4.5.4. Visualización, adquisición de datos y ejecución de la simulación-ejemplo: Macro files (`vis_BrainExp.mac`)

El archivo ejecutable `AcLinealCerebro`, se puede correr en la terminal mediante `./` así:

```
45 fisinfor@taozen:~/geant4_workdir/BrainExp/BrainExp_build$ ./AcLinealCerebro
```

obteniéndose en la terminal la respuesta:

```
46 ****
47 Geant4 version Name: geant4-10-04-patch-02 (25-May-2018)
48 Copyright : Geant4 Collaboration
49 References : NIM A 506 (2003), 250-303
50 : IEEE-TNS 53 (2006), 270-278
51 : NIM A 835 (2016), 186-225
52 WWW : http://geant4.org/
53 ****
54
55 Visualization Manager instantiating with verbosity "warnings (3)"...
56 Visualization Manager initialising...
57 Registering graphics systems
58     --> Isotope: Ca48 Z = 20 N = 48 A = 47.95 g/mole abundance: 0.187 %
59     ElmMassFraction: 22.50 % ElmAbundance 6.53 %
60
61 ...
62 ...
63 ...
64
```

Una simulación-ejemplo: GEANT4 hands on!

```
65 G4PhysicsListHelper::AddTransportation() --- G4CoupledTransportation is used  
66 Available UI session types: [ Qt, GAG, tcsh, csh ]
```

e inmediatamente una ventana emergente con la interfaz gráfica de usuario propia de **GEANT4** (ver Fig. 4.6).

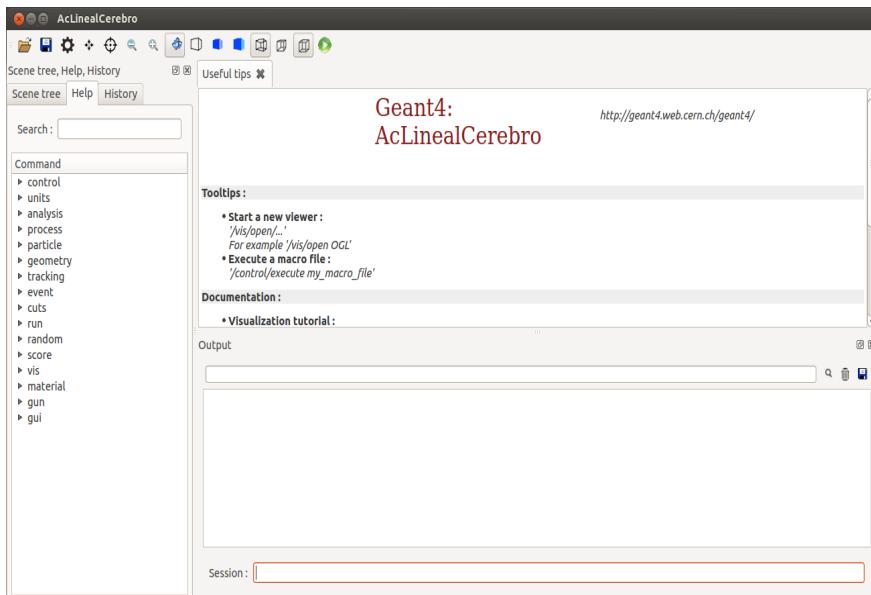


Figura 4.6: Interfaz gráfica de usuario propia de **GEANT4**. Note que en la parte superior izquierda aparece el nombre del archivo ejecutable de la simulación; para este caso: **AcLinealCerebro**.

La interfaz gráfica de **GEANT4** permite ejecutar de manera interactiva el programa de simulación. Posee varias ayudas para ejecutar comandos en la caja de texto, "Session:" sobre la parte inferior de la interfaz, y también describe una ayuda en la parte superior, que le permite al usuario evidenciar la estructura de linea de comandos de dicha interfaz.

Para el caso del ejercicio práctico, se elaboran 19 líneas de comando, para ser escritas en la caja de texto "Session:" (y dar enter para su ejecución), pensadas en las particularidades sobre las cuales, un usuario en formación inicial de **GEANT4** debería familiarizarse. Estas son:

1. **/vis/open OGL** Existen varios visualizadores en **GEANT4** como OpenGL, OpenInventor,

Una simulación-ejemplo: GEANT4 hands on!

HepRep, DAWN, VRML, RayTracer, gMocren y ASCII Tree. En este caso se utiliza la opción OGL para escoger a OpenGL como visualizador de la simulación. Recuerde, se debe escribir en la caja de texto Session: la línea de comando y dar enter. La respuesta se ilustra en las capturas de pantalla de la Fig. 4.7.

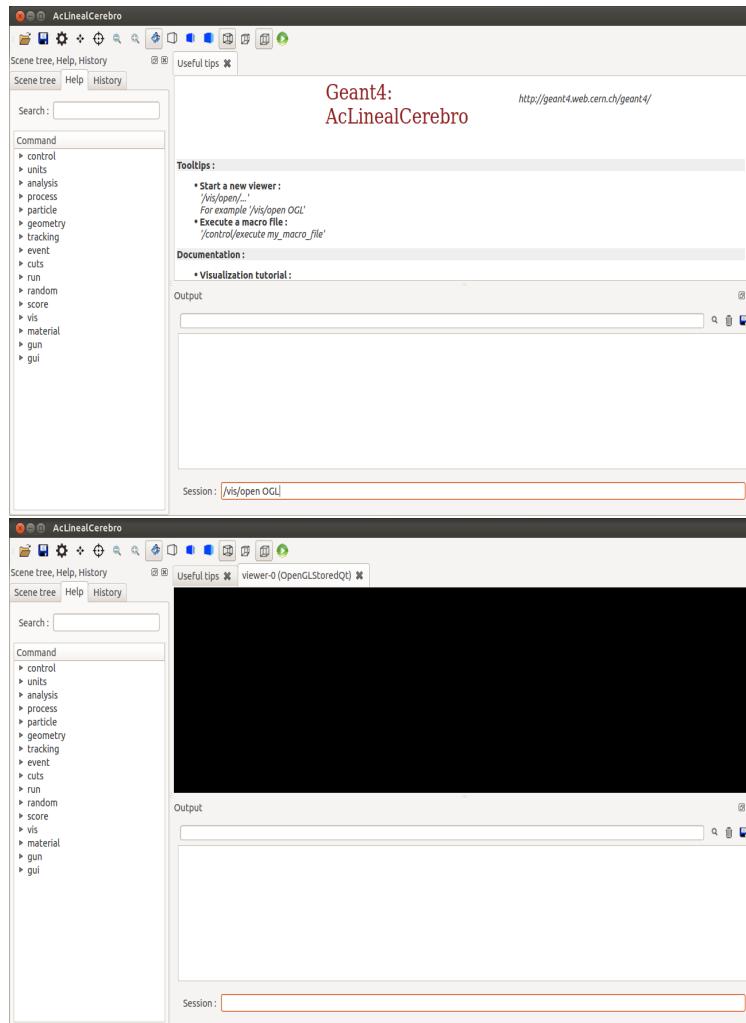


Figura 4.7: /vis/open OGL Note que aparece una pestaña en la parte superior con la leyenda "viewer-0 (OpenGLStoredQt)".

Una simulación-ejemplo: GEANT4 hands on!

2. /vis/drawVolume Permite visualizar la geometría del detector (ver Fig. 4.8).

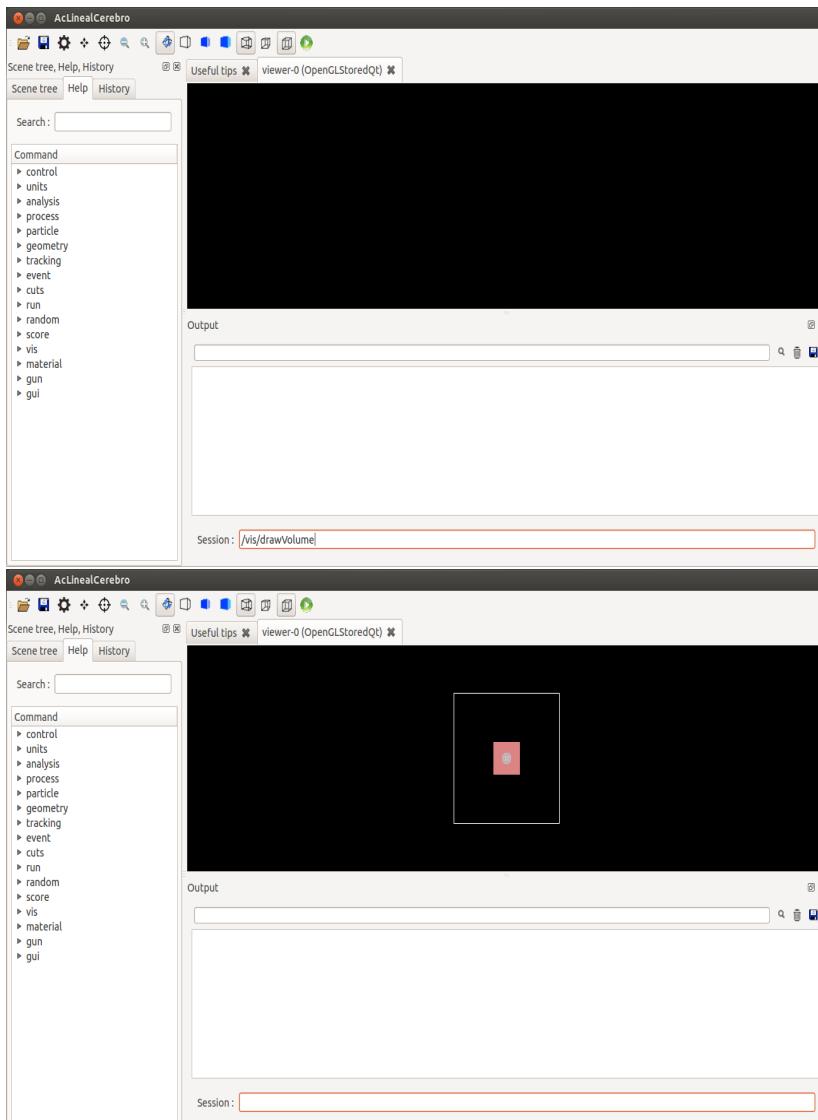


Figura 4.8: /vis/drawVolume Note que si utiliza la tecla “tabular” para auto-completar la linea de comandos, aparecen ayudas en la parte central de la interfaz.

Una simulación-ejemplo: GEANT4 hands on!

7. `/vis/scene/endOfEventAction accumulate` Prepara acumular eventos en la visualización (ver Fig. 4.13).

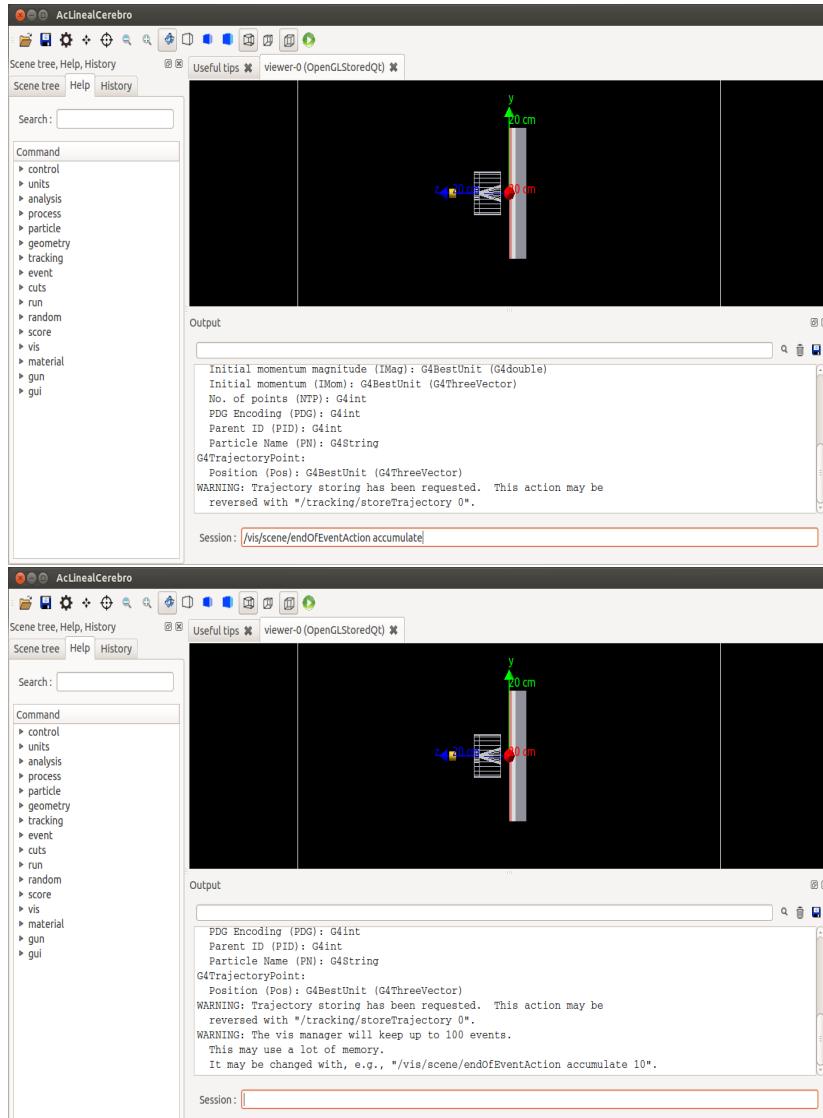


Figura 4.13: `/vis/scene/endOfEventAction accumulate`

Una simulación-ejemplo: GEANT4 hands on!

8. /score/create/boxMesh boxMesh Score, se puede traducir como “puntuar”, o si se prefiere, “anotar”. En el contexto de GEANT4 se usa para anotar las interacciones (o “hits”) que tuvieron lugar en la simulación. Lo anterior implica que se debe ubicar espacialmente cada parte del detector en una malla tridimensional, la cual esta compuesta de retículos que van a conformar el detector o parte del detector que se quiera estudiar. Se usa entonces la opción boxMesh para que los retículos se asemejen a cajas (ver Fig. 4.14).

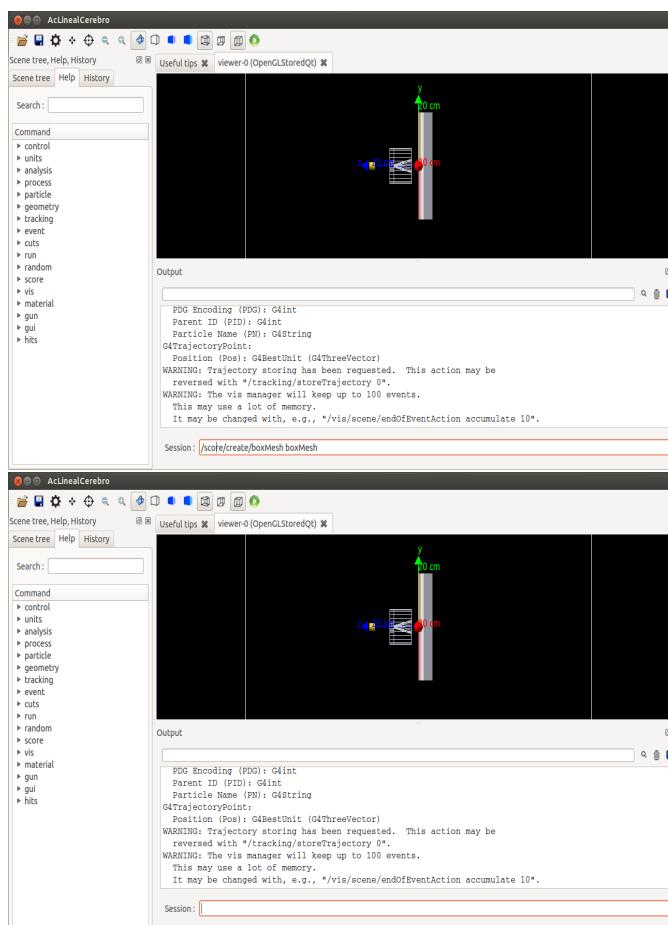


Figura 4.14: /score/create/boxMesh boxMesh

Una simulación-ejemplo: GEANT4 hands on!

9. `/score/mesh/boxSize 15.0 15.0 2.4 cm` Define la talla del volumen a subdividir en retículos (ver Fig. 4.15). Importante: En GEANT4, las cajas (box, boxes) se definen desde su centro geométrico, es decir, si se requiere una caja de $30 \times 30 \times 4.8 \text{ cm}^3$, basta con definir, para cada eje, la distancia del plano a su centro geométrico, que resulta ser la mitad de cada dimensión.

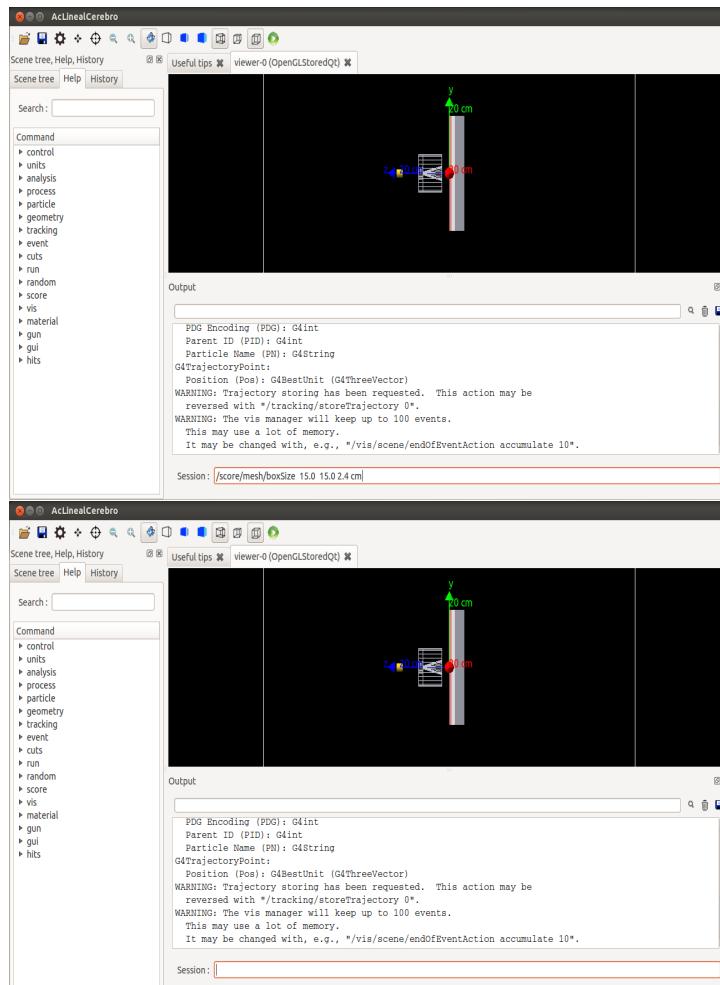


Figura 4.15: `/score/mesh/boxSize 15.0 15.0 2.4 cm`

Una simulación-ejemplo: GEANT4 hands on!

- 10. /score/mesh/nBin 1 1 100** Define la talla de las cajas, o "bins". Sus opciones corresponden a dividir el eje X de la caja en 1 bin, el eje Y de la caja en 1 bin, y el eje Z en 100 bines, de 1 a 100 en la dirección positiva del eje. Se divide entonces el volumen del boxMesh en 100 cajas (o retículas) de un bin en X (30 cm) y un bin en Y (30 cm) y un bin en Z ($4.8\text{ cm}/100=0.048\text{ cm}$). Esto se realizó con el objeto de simplificar la simulación (ver Fig. 4.16).

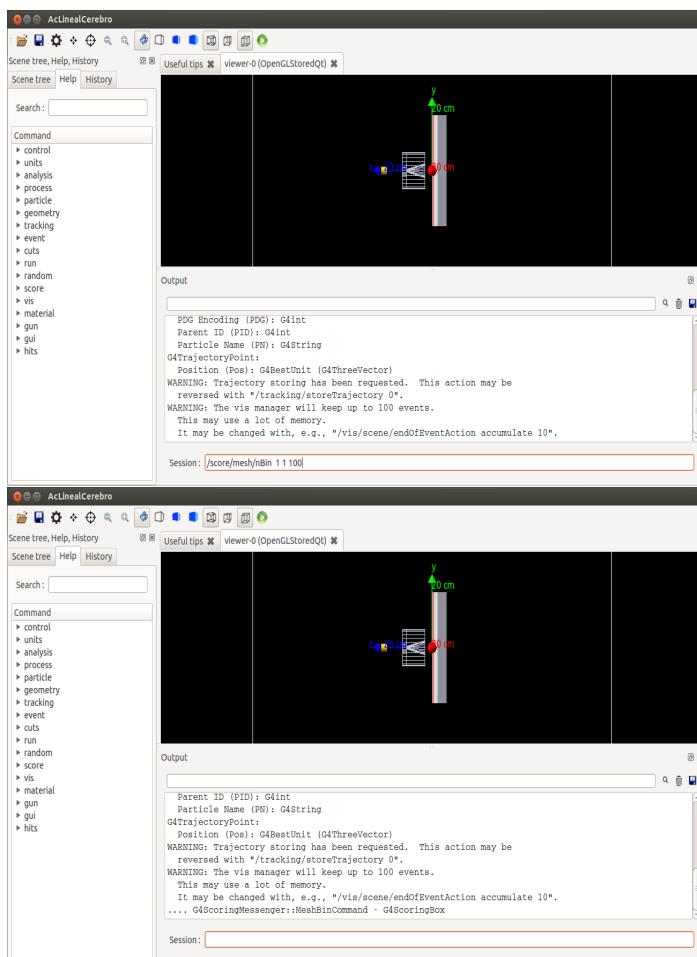


Figura 4.16: /score/mesh/nBin 1 1 100

Una simulación-ejemplo: GEANT4 hands on!

11. `/score/mesh/translate/xyz 0 0 -2.4 cm` Dado que el mesh (o “enmallado”) definido para la simulación-ejemplo es una caja, la cual se ubica automáticamente en el centro geométrico de la geometría de la simulación, esta se debe trasladar -2.4 cm en Z para que abarque todo el volumen del paralelepípedo que simula el cráneo-cerebro (ver Fig. 4.17).

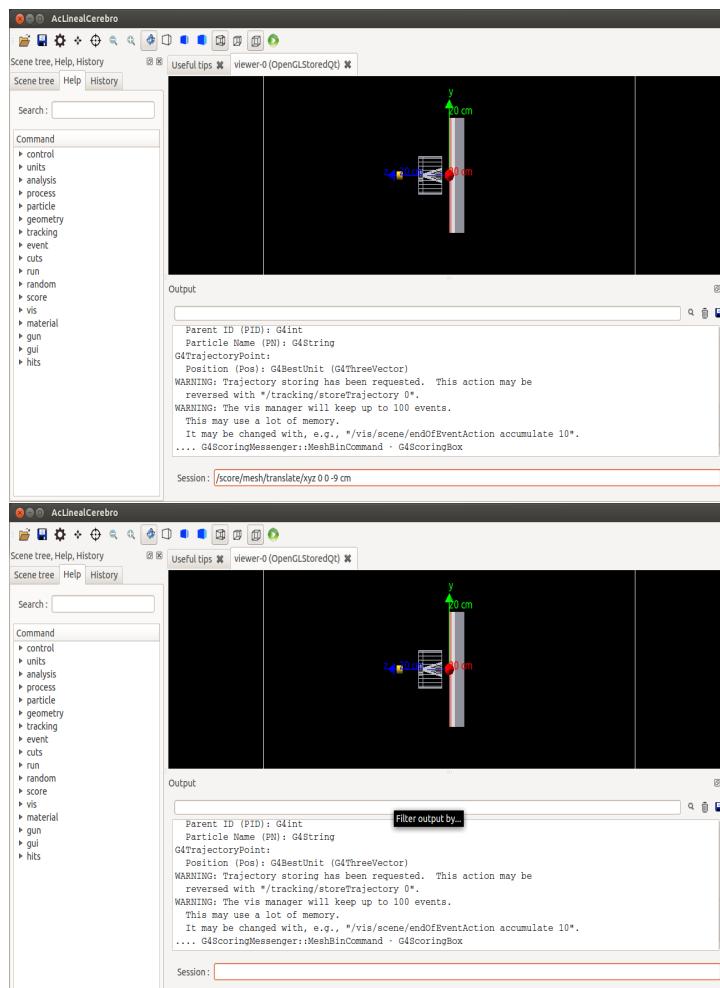


Figura 4.17: `/score/mesh/translate/xyz 0 0 -2.4 cm`

Una simulación-ejemplo: GEANT4 hands on!

12. `/score/mesh/rotate/rotateY 180.0 deg` Como se mencionó anteriormente, el eje Z de 0 cm a -4.8 cm se subdivide en 100 partes (o bins) que se ordenan de 1 a 100 en la dirección positiva del eje Z. Con el objeto de re-ordenarlos, es decir, que el primer bin quede cerca de Z=0cm y el bin 100 quede cerca de Z=-4.8cm se realiza una rotación del volumen de 180 grados sobre el eje Y (que por la simetría del problema es equivalente a realizar la rotación sobre el eje X, ver Fig. 4.18).

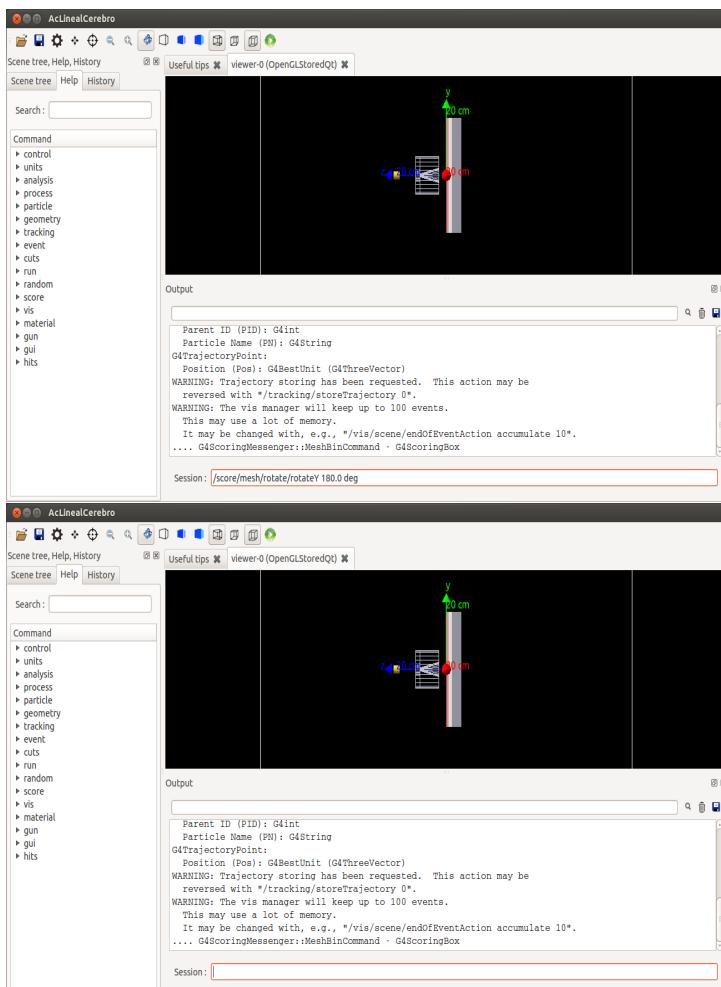


Figura 4.18: `/score/mesh/rotate/rotateY 180.0 deg`

Una simulación-ejemplo: GEANT4 hands on!

- 13, 14. `/score/quantity/energyDeposit EnerDepo` Permite guardar en memoria una cantidad particular. Para efectos de la simulación ejemplo se decidió guardar dos valores: la energía depositada en el volumen que define `boxMesh` (que simula la cabeza del paciente) (ver Fig. 4.19) y, `/score/quantity/n0fCollision N0fCollision`, el número de colisiones en cada retícula (ver Fig. 4.20).

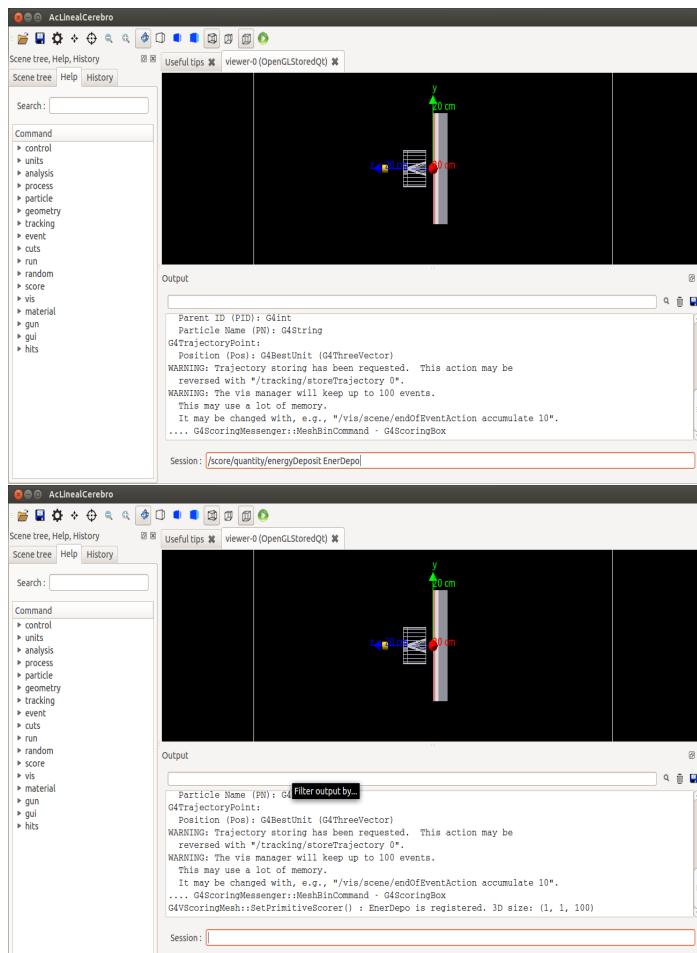


Figura 4.19: `/score/quantity/energyDeposit EnerDepo`

Una simulación-ejemplo: GEANT4 hands on!

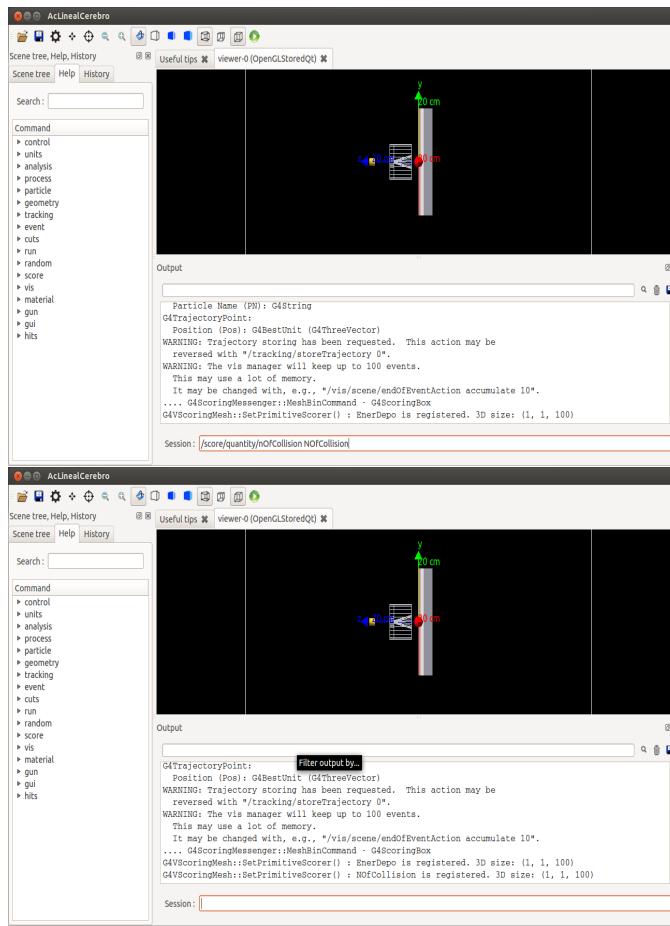


Figura 4.20: /score/quantity/nOfCollision NOfCollision

Note que

```
/score/quantity/energyDeposit EnerDepo
/score/quantity/nOfCollision NOfCollision
```

registran dos puntuadores o anotadores ("scorers") con nombres EnerDepo y NOfCollision, esto permite a GEANT4 guardar en memoria valores para posteriormente ser imprimidos en un archivo de datos.

Una simulación-ejemplo: GEANT4 hands on!

17. `/run/beamOn 500` Asiente la ejecución de la simulación. Para el caso realiza una corrida de 500 eventos producto de la producción de partículas por la fuente de partículas de la simulación y el detector (ver Fig. 4.20).

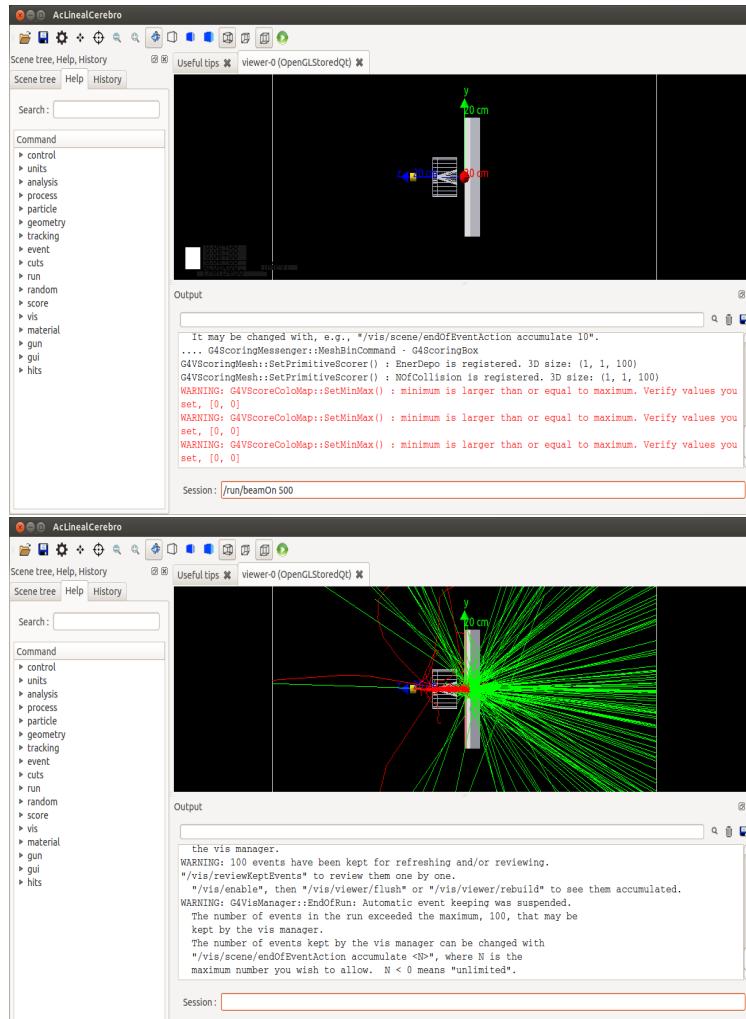


Figura 4.23: `/run/beamOn 500` Se visualiza la geometría del detector, que corresponde a la fuente de partículas, el colimador y el volumen que simula el cráneo del paciente. Se evidencian las trazas que siguen las partículas al interactuar con el volumen de la simulación.

Una simulación-ejemplo: GEANT4 hands on!

18, 19. `/score/dumpQuantityToFile boxMesh EnerDepo EnerDepo.csv`
`/score/dumpQuantityToFile boxMesh NOfCollision NOfCollision.csv`
generan dos archivos de datos con extensión .csv ("comma separated values"), que poseen en su estructura, una cabecera de información y seis campos de valores:

```
# mesh name: boxMesh  
# primitive scorer name: EnerDepo  
# iX, iY, iZ, total(value) [MeV], total(val^2), entry
```

Tres campos donde ubica cada uno de los retículos acordes a su posición "en bins", a lo largo de los tres ejes (iX, iY, iZ), un campo para el valor de la variable, total(value), y su unidad correspondiente (si la hay), la incertidumbre de dicha variable (para el caso que aplique, o simplemente el valor de la variable elevado al cuadrado), total(val^2), y el número de interacciones (o "hits"), entry, que registró la simulación en cada retícula iX, iY, iZ.

IMPORTANTE: Se resalta el hecho de que ir a lo largo de las 19 líneas de comando que se ejecutan en la interfaz de GEANT4 permite visualizar el paso a paso de la simulación, haciendo evidente el funcionamiento del ejercicio de implementación de los códigos diseñados para esta simulación ejemplo; se sigue entonces, antes de ejecutar cualquier parte de la simulación que el lector novato realice una lectura de este capítulo antes de hacer el primer acercamiento a la implementación para que su cerebro sensibilice "los detalles" del trabajo práctico.

Las 19 líneas de comando se pueden recoger en un archivo o macro de simulación, que se codifica bajo la extensión .mac, y que para esta simulación se ha llamado vis_BrainExp.mac:

```
1 //----- Macro de ejecucion: vis_BrainExp.mac -----  
2 /vis/open OGL  
3 /vis/drawVolume  
4 /vis/viewer/set/viewpointThetaPhi 90 0  
5 /vis/viewer/zoom 4  
6 /vis/scene/add/axes 0 0 0 20 cm  
7 /vis/scene/add/trajectories  
8 /vis/scene/endOfEventAction accumulate  
9 /score/create/boxMesh boxMesh  
10 /score/mesh/boxSize 15.0 15.0 2.4 cm  
11 /score/mesh/nBin 1 1 100  
12 /score/mesh/translate/xyz 0 0 -2.4 cm  
13 /score/mesh/rotate/rotateY 180.0 deg
```

Capítulo 5

Simulación-ejemplo extendida: RunAction, EventAction, ROOT

Una simulación en *GEANT4*puede contener requerimientos de software “adicionales” (subsección 3.4.1), para estudiar, por ejemplo, la estadística final de una corrida (G4Run), lo que sucede en la simulación evento a evento (G4Event), o, las trazas que siguen las partículas (G4Track, G4Step).

Estos requerimientos “adicionales” dependen de la necesidad de estudio que diseña una simulación. Para el ejercicio de alfabetización científica y tecnológica orientado a profesores-investigadores, se diseña una extensión de la simulación-ejemplo donde se hace uso de tres requerimientos de software adicionales a los básicos (detallados en el capítulo anterior):

- **RunAction:** Establece las condiciones inicial y final de la ejecución de la corrida. Pertenece al software de *GEANT4*.
- **EventAction:** Describe la dinámica evento a evento, y se encuentra embebido en *RunAction*. Para el caso, permite extraer la información evento a evento de la deposición de energía.
- **ROOT** Una infraestructura de análisis de datos orientada a objetos, que no pertenece al software de *GEANT4*, y permite hacer un análisis profesional de datos, en ella se puede programar interfaces gráficas de usuario para el control de experimentos y simulaciones (Vargas-Hernández y Salamanca-Bernal, 2015). En este caso se utiliza para almacenar la información de la deposición de energía en el absorbente evento a evento, y para hacer un análisis de la deposición de energía acumulada en el absorbente por unidad de longitud. La codificación de las herramientas

Simulación-ejemplo extendida: RunAction, EventAction, ROOT

computacionales de ROOT se encuentran embebidas tanto en `RunAction` y `EventAction`.

con lo que se quiere hacer seguimiento de la deposición de energía sobre el absorbente evento a evento, y así tener información que permita calcular el valor mas probable de deposición de energía de un evento en un tratamiento con fuente de electrones¹.

5.1. ROOT: HistoManager

Inicialmente se programa una clase llamada `HistoManager`, donde se definen objetos de ROOT para ser usados en la simulación. Se requiere hacer un histograma del valor de deposición de energía en todo el absorbente por evento. Para ello se debe declarar el objeto histograma a utilizar (`TH1D fHisto[]`) y crear el archivo de salida `BrainExp.root` en el cual se grabará el histograma propio de la deposición de energía `EnergyDeposited_per_event`, como ilustra los dos siguientes recuadros:

```
1 //----- HistoManager.hh -----
2
3 #ifndef HistoManager_h
4 #define HistoManager_h 1
5
6 #include "globals.hh"
7
8 class TFile;
9 class TH1D;
10
11 const G4int kMaxHisto = 1; // number of histos, in this case just 1 histo
12
13 class HistoManager
14 {
15 public:
16     HistoManager();
17     ~HistoManager();
18
19     void Book();
20     void Save();
21     void FillHisto(G4int id, G4double bin, G4double weight = 1.0);
22
23 private:
24     TFile*   fRootFile;
25     TH1D*   fHisto[kMaxHisto];
26 };
27
28 #endif
```

¹Simplemente por establecer una razón para ilustrar la codificación de la simulación-ejemplo-extendida

```
1 //----- HistoManager.cc -----
2 #include <TH1D.h>
3 #include <TFile.h>
4 #include <CLHEP/Units/SystemOfUnits.h>
5
6 #include "HistoManager.hh"
7 #include "G4UnitsTable.hh"
8
9 HistoManager::HistoManager()
10   :fRootFile(0)
11 {
12
13   // histograms
14   for (G4int k=0; k<kMaxHisto; k++)
15     fHisto[k] = 0;
16 }
17
18 HistoManager::~HistoManager()
19 {
20   if (fRootFile)
21     delete fRootFile;
22 }
23
24 void HistoManager::Book()
25 {
26   // Creating a tree container to handle histograms and ntuples.
27   // This tree is associated to an output file.
28   //
29   G4String fileName = "BrainExp.root";
30   fRootFile = new TFile(fileName,"RECREATE");
31   if (! fRootFile) {
32     G4cout << " HistoManager::Book : "
33           << " problem creating the ROOT TFile "
34           << G4endl;
35     return;
36   }
37
38   // id = 0
39   fHisto[0] = new TH1D("EnergyDeposited_per_event",
40                       "Energía depositada en el absorbente (MeV)", 300, 0., 0.3*CLHEP::MeV);
41   fHisto[0]->GetXaxis()->SetTitle("MeV");
42   fHisto[0]->GetYaxis()->SetTitle("Eventos");
43
44   // id = 1
45   // ...
46
47   // id = 2
48   // ...
49
50   for ( G4int i=0; i<kMaxHisto; ++i ) {
51     if (! fHisto[i]) G4cout << "\n can't create histo " << i << G4endl;
52   }
53
54   G4cout << "\n----> Output file is open in " << fileName << G4endl;
55 }
```

Simulación-ejemplo extendida: RunAction, EventAction, ROOT

```
57 void HistoManager::Save()
58 {
59     if (! fRootFile) return;
60
61     fRootFile->Write();           // Writing the histograms to the file
62     fRootFile->Close();          // and closing the tree (and the file)
63
64     G4cout << "\n----> Histograms and ntuples are saved\n" << G4endl;
65 }
66
67 void HistoManager::FillHisto(G4int ih, G4double xbin, G4double weight)
68 {
69     if (ih >= kMaxHisto) {
70         G4cerr << "----> warning from HistoManager::FillHisto() : histo " << ih
71             << " does not exist. (xbin=" << xbin << " weight=" << weight << ")"
72             << G4endl;
73     return;
74 }
75     if (fHisto[ih]) {
76         fHisto[ih]->Fill(xbin, weight);
77     }
78 }
```

5.2. RunAction: AcLinealRunAction

Una corrida en un experimento real (Run) se compone de condiciones iniciales de la corrida, que a su vez se compone de eventos, estos a su vez de "hits" en los detectores, con unas trayectorias particulares.

La anterior estructura de ejecución de un experimento es implementada en GEANT4, por lo que la clase `AcLinealRunAction`, programada para la simulación-ejemplo-extendida, define los métodos `BeginRunAction` y `EndOfRunAction` que preparan el inicio de la corrida y su final. Para el caso se requiere que al inicio de la corrida se declaren los histogramas y archivo de salida `BrainExp.root`, que serán llamados por ROOT, y al final, guarde los histogramas y cierre el archivo:

```
1 //----- AcLinealRunAction.hh -----
2
3 #ifndef RunAction_h
4 #define RunAction_h 1
5
6 #include "G4UserRunAction.hh"
7 #include "globals.hh"
8
9 class G4Run;
10 class HistoManager;
11
12 class AcLinealRunAction : public G4UserRunAction
```

```
13 {
14 public:
15   AcLinealRunAction(HistoManager* );
16   virtual ~AcLinealRunAction();
17
18   virtual void BeginOfRunAction(const G4Run* );
19   virtual void EndOfRunAction(const G4Run* );
20
21 private:
22   HistoManager* fHistoManager;
23 };
24 #endif
```

```
1 //----- AcLinealRunAction.cc -----
2 #include "AcLinealRunAction.hh"
3 #include "HistoManager.hh"
4
5 #include "G4Run.hh"
6 #include "G4RunManager.hh"
7 #include "G4UnitsTable.hh"
8
9 AcLinealRunAction::AcLinealRunAction(HistoManager* histo)
10 : G4UserRunAction(),
11   fHistoManager(histo)
12 {}
13
14 AcLinealRunAction::~AcLinealRunAction()
15 {}
16
17 void AcLinealRunAction::BeginOfRunAction(const G4Run* aRun)
18 {
19   G4cout << "## Run " << aRun->GetRunID() << " start." << G4endl;
20   fHistoManager->Book();
21 }
22
23 void AcLinealRunAction::EndOfRunAction(const G4Run* aRun)
24 {
25   G4int NbOfEvents = aRun->GetNumberOfEvent();
26   if (NbOfEvents == 0) return;
27
28   //save histograms
29   fHistoManager->Save();
30 }
```

5.3. EventAction: AcLinealEventAction

Puesto que se tiene la necesidad de llenar un histograma evento a evento, se programa la clase AcLinealEventAction que permite extraer la información evento a evento y llenar los histogramas programados en ROOT:

Simulación-ejemplo extendida: RunAction, EventAction, ROOT

```
1 //----- AcLinealEventAction.hh -----
2
3 #ifndef EventAction_h
4 #define EventAction_h 1
5
6 #include "G4UserEventAction.hh"
7 #include "G4THitsMap.hh"
8
9 #include "globals.hh"
10
11 class AcLinealRunAction;
12 class HistoManager;
13
14 class AcLinealEventAction : public G4UserEventAction
15 {
16 public:
17     AcLinealEventAction(AcLinealRunAction*, HistoManager*);
18     virtual ~AcLinealEventAction();
19
20     virtual void BeginOfEventAction(const G4Event* );
21     virtual void EndOfEventAction(const G4Event* );
22
23 private:
24     AcLinealRunAction* fRunAct;
25     HistoManager* fHistoManager;
26
27     G4THitsMap<G4double>* GetHitsCollection(G4int hcID, const G4Event* event) const;
28     G4double fEdep;
29
30     G4int fCollID;
31     G4int fPrintModulo;
32 };
33 #endif
```

```
1 //----- AcLinealEventAction.cc -----
2
3 #include "AcLinealEventAction.hh"
4
5 #include "AcLinealRunAction.hh"
6 #include "HistoManager.hh"
7
8 #include "G4RunManager.hh"
9 #include "G4Event.hh"
10 #include "G4SDManager.hh"
11 #include "G4HCofThisEvent.hh"
12 #include "G4UnitsTable.hh"
13 #include "G4THitsMap.hh"
14
15 AcLinealEventAction::AcLinealEventAction(AcLinealRunAction* run, HistoManager* histo)
16     :G4UserEventAction(),
17     fRunAct(run), fHistoManager(histo), fEdep(0.),
18     fCollID(-1),
19     fPrintModulo(0)
20 {
21     fPrintModulo = 100;
```

```
22}
23
24 AcLinealEventAction::~AcLinealEventAction()
25 { }
26
27 G4THitsMap<G4double>* AcLinealEventAction::GetHitsCollection(G4int hcID, const G4Event* event) const
28 {
29     auto hitsCollection = static_cast<G4THitsMap<G4double>*>(event->GetHCoThisEvent()->GetHC(hcID));
30
31     if ( ! hitsCollection ) {
32         G4ExceptionDescription msg;
33         msg << "Cannot access hitsCollection ID " << hcID;
34         G4Exception("B4dEventAction::GetHitsCollection()", "MyCode0003", FatalException, msg);
35     }
36
37     return hitsCollection;
38 }
39
40 void AcLinealEventAction::BeginOfEventAction(const G4Event* evt)
41 {
42     G4int evtNb = evt->GetEventID();
43     if (evtNb%PrintModulo == 0)
44         G4cout << "\n--- Begin of event: " << evtNb << G4endl;
45
46 // initialisation per event
47 fEdep = 0.;
48 }
49
50 void AcLinealEventAction::EndOfEventAction(const G4Event* evt)
51 {
52     G4HCofThisEvent* HCE = evt->GetHCoThisEvent();
53     if(!HCE) return;
54
55     if (fCollID < 0) {
56         G4SDManager* SDMan = G4SDManager::GetSDMpointer();
57         fCollID = SDMan->GetCollectionID("myCellScorer/EnergyDepo");
58     }
59
60     G4THitsMap<G4double>* evtMap = (G4THitsMap<G4double>*)(HCE->GetHC(fCollID));
61     std::map<G4int, G4double*>::iterator itr;
62
63
64     for (itr = evtMap->GetMap()->begin(); itr != evtMap->GetMap()->end(); itr++) {
65         G4cout << "Cheking Second on itr ... " << G4endl;
66         fEdep = *(itr->second);
67     }
68
69 // filling histos
70 fHistoManager->FillHisto(0, fEdep);
71
72 G4cout << "Energy deposition on event: " << fEdep << G4endl;
73
74 }
```

Nótese que primero se debe programar la función GetHitsCollection para obtener información de cada coincidencia o hit y su asociación con cada evento. Cabe destacar

Simulación-ejemplo extendida: RunAction, EventAction, ROOT

que la función `BeginOfEventAction` obtiene el identificador del objeto, para luego poder extraer la información de la deposición de energía (`fEdep`) y así llenar el histograma correspondiente.

5.4. Función main() de la simulación-ejemplo-extendida

La función `main()` definida en el archivo `AcLinealCerebroEXTENDED` requiere declarar tres objetos de las clases `HistoManager`, `AcLinealRunAction` y `AcLinealEventAction` como acciones adicionales programadas para el usuario de la simulación:

```
1 //---- EXTENDED main(): AcLinealCerebro.cc -----
2 #include "G4RunManager.hh"
3 #include "G4UImanager.hh"
4 #include "G4ScoringManager.hh"
5 #include "G4Uterminal.hh"
6 #include "AcLinealConstruccion.hh"
7 #include "AcLinealListaFisica.hh"
8 #include "AcLinealFuente.hh"
9 #include "AcLinealRunAction.hh"
10 #include "AcLinealEventAction.hh"
11 #include "HistoManager.hh"
12
13 #include "G4UIExecutive.hh"
14 #include "G4SystemOfUnits.hh"
15 #ifdef G4VIS_USE
16 #include "G4VisExecutive.hh"
17#endif
18
19 int main(int argc, char** argv){
20    G4RunManager *runManager = new G4RunManager;
21    G4ScoringManager* scoringManager = G4ScoringManager::GetScoringManager();
22    scoringManager->SetVerboseLevel(1);
23 #ifdef G4VIS_USE
24    G4VisManager* visManager = new G4VisExecutive;
25    visManager->Initialize();
26#endif
27    runManager -> SetUserInitialization(new AcLinealConstruccion);
28    runManager -> SetUserInitialization(new AcLinealListaFisica);
29    runManager -> SetUserAction(new AcLinealFuente);
30
31    HistoManager* histo = new HistoManager();
32
33    AcLinealRunAction* run_action = new AcLinealRunAction(histo);
34    runManager -> SetUserAction(run_action);
35
36    AcLinealEventAction* event_action = new AcLinealEventAction(run_action,histo);
37    runManager -> SetUserAction(event_action);
38
39    runManager -> Initialize();
```

```
40 |     if(argc==1){
41 | #ifdef G4UI_USE
42 |     G4UIExecutive* ui = new G4UIExecutive(argc, argv);
43 |     ui->SessionStart();
44 |     delete ui;
45 | #endif
46 |     }
47 | else{
48 |     G4UImanager* UImanager = G4UImanager::GetUIpointer();
49 |     G4String command = "/control/execute ";
50 |     G4String fileName = argv[1];
51 |     UImanager->ApplyCommand(command+fileName);
52 |     }
53 |
54 | #ifdef G4VIS_USE
55 |     delete visManager;
56 | #endif
57 |     delete runManager;
58 |     return 0;
59 | }
```

Fíjese que el objeto `run_action` contiene el parámetro `histo`, y que `event_action` contiene dos parámetros, `run_action` y `histo`, como se comentó anteriormente.

5.5. Sensitive Detector: A_cLinealConstrucción

Una simulación requiere información útil que generalmente está relacionada con un estudio evento a evento de los procesos que se simulan, o necesidades particulares del experimento a simular.

Se requiere entonces extraer información, no solo de la globalidad del experimento, sino información particular de un volumen que representa un elemento de la geometría de la simulación. A cada volumen lógico que se defina en `GEANT4` se puede asociar una “sensibilidad” sobre el paso de partículas, la cual se define a través de la clase `G4SensitiveDetector` y se interpreta como un “detector virtual”, que permite extraer información física de la interacción del volumen lógico con la radiación-materia.

Principalmente se requiere colectar las coincidencias (interacciones o hits) por evento (`G4HCofThisEvent`) donde la información estará disponible al final del evento (`EndOfEventAction`), y se pueda almacenar en archivos de datos (o histogramas) para su posterior análisis.

`GEANT4` posee un detector sensible por defecto (“sensitive detector”) el cual se declara bajo la clase `G4MultiFunctionalDetector` que permite asociar valores primitivos de anotaciones (“scores”) de los diferentes valores definidos por defecto, como el de la

Simulación-ejemplo extendida: RunAction, EventAction, ROOT

energía depositada (G4PSEnergyDeposit).

Por lo anterior, el detector sensible que se programe debe estar ubicado en el archivo que define la construcción geométrica del detector, pues es allí donde se definen los volúmenes lógicos de la simulación:

```
1 //---- EXTENDED main(): AcLinealConstruccion.cc -----
2 #include "G4SystemOfUnits.hh"
3 #ifndef AcLinealConstruccion_hh
4 #define AcLinealConstruccion_hh
5 #include "G4VUserDetectorConstruction.hh"
6
7 class G4VPhysicalVolume;
8 class G4NistManager;
9
10 class AcLinealConstruccion : public G4VUserDetectorConstruction{
11 public:
12     AcLinealConstruccion();
13     ~AcLinealConstruccion();
14
15     G4VPhysicalVolume* Construct();
16     void Construct_Fuente(G4LogicalVolume* eumma, G4double posz);
17     void Construct_Colimador(G4LogicalVolume* eumma, G4double posz);
18     void Construct_MEsquelitico(G4LogicalVolume* eumma, G4double posz);
19     void Construct_HCortical(G4LogicalVolume* eumma, G4double posz=0.0*cm);
20     void Construct_Cerebro(G4LogicalVolume* eumma, G4double posz);
21 private:
22     G4NistManager *NIST;
23 };
24 #endif
```

```
1 //---- EXTENDED main(): AcLinealConstruccion.cc -----
2 #include "G4Material.hh"
3 #include "G4NistManager.hh"
4 #include "G4LogicalVolume.hh"
5 #include "G4ThreeVector.hh"
6 #include "G4PVPlacement.hh"
7 #include "G4Box.hh"
8 #include "G4Tubs.hh"
9 #include "G4Polycone.hh"
10 #include "G4VisAttributes.hh"
11 #include "G4Colour.hh"
12 #include "AcLinealConstruccion.hh"
13 #include "G4SystemOfUnits.hh"
14
15 //sensitive detector
16 #include "G4MultiFunctionalDetector.hh"
17 #include "G4VPrimitiveScorer.hh"
18 #include "G4PSEnergyDeposit.hh"
19 #include "G4SDManager.hh"
20
21 AcLinealConstruccion::AcLinealConstruccion(){NIST=G4NistManager::Instance();}
22 AcLinealConstruccion::~AcLinealConstruccion(){}
```

```

23
24 G4VPhysicalVolume* AcLinealConstruccion::Construct(){
25
26     G4Material* Aire = NIST->FindOrBuildMaterial("G4_AIR");
27
28     G4Box           * World_geo = new G4Box("World_g", 60.0*cm , 60.0*cm , 60.0*cm);
29     G4LogicalVolume * World_log = new G4LogicalVolume(World_geo,Aire,"World_l",0,0,0);
30     G4VPhysicalVolume * World_phy = new G4PVPlacement(0,G4ThreeVector(),World_log,"World", 0,false,0);
31
32     Construct_Fuente(World_log, 15.0*cm);
33     Construct_Colimador(World_log, 10.0*cm);
34     Construct_MEsquelitico(World_log, -0.4*cm);
35     Construct_HCortical(World_log, -1.3*cm);
36     Construct_Cerebro(World_log, -3.3*cm);
37
38     // Sensitive detector
39     G4MultiFunctionalDetector* myScorer = new G4MultiFunctionalDetector("myCellScorer");
40     G4SDManager::GetSDMpointer() -> AddNewDetector(myScorer);
41     World_log->SetSensitiveDetector(myScorer); // attached to logical volume
42
43     G4PSEnergyDeposit* energyDepo = new G4PSEnergyDeposit("EnergyDepo");
44     myScorer->RegisterPrimitive(energyDepo);
45
46     G4cout << "Sensitive detector DONE!!!" << G4endl;
47
48     return World_phy;
49 }
50 //----- Fuente de Radiacion -----
51 void AcLinealConstruccion::Construct_Fuente(G4LogicalVolume* mother, G4double posz){
52     const G4double WFuente_Espesor = 2.0*cm;
53     const G4double WFuente_pos      = posz-(WFuente_Espesor)/2.0+2.0*cm;
54
55     G4Material* Tungsteno = NIST->FindOrBuildMaterial("G4_AIR");
56     G4VisAttributes* W_Vis = new G4VisAttributes(G4Colour(1.0, 0.843, 0.161, 1.0));
57     W_Vis -> SetForceSolid(true);
58     G4Tubs*           WFuente_geo = new G4Tubs("WFuente_g",
59                                         0.0*cm, 1.0*cm, WFuente_Espesor/2.0 , 0.0*deg, 360.0*deg );
60     G4LogicalVolume* WFuente_log = new G4LogicalVolume(WFuente_geo, Tungsteno, "WFuente_l" );
61     WFuente_log -> SetVisAttributes(W_Vis);
62     new G4PVPlacement(0, G4ThreeVector(0,0, WFuente_pos), WFuente_log, "WFuente", mother, false, 0 );
63 }
64 //----- Colimador -----
65 void AcLinealConstruccion::Construct_Colimador(G4LogicalVolume* mother, G4double posz){
66     const G4int nZ      = 4 ;
67     const G4double rIn[] = {1.5*cm, 1.5*cm, 0.38*cm, 1.9*cm};
68     const G4double rOut[] = {5*cm, 5*cm, 5*cm, 5*cm};
69     const G4double ZColi[] = {1.5*cm, 6.1*cm};
70     const G4double Z[]     = {0.0, -ZColi[0], -ZColi[0], -(ZColi[0]+ZColi[1])};
71
72     G4Material* Tungsteno = NIST->FindOrBuildMaterial("G4_W");
73
74     G4Polycone*   Col_geo = new G4Polycone("Col_g", 0*deg, 360*deg, nZ, Z, rIn, rOut);
75     G4LogicalVolume* Col_log = new G4LogicalVolume(Col_geo, Tungsteno, "Col_l",0,0,0);
76     new G4PVPlacement(0, G4ThreeVector(0,0, posz), Col_log, "Col",mother, false , 0);
77
78     G4VisAttributes* VAttW = new G4VisAttributes(G4Colour(0.75, 0.76, 0.8, 1.0));

```

Simulación-ejemplo extendida: RunAction, EventAction, ROOT

```
79 VAttW -> SetForceAuxEdgeVisible(true);
80 Col_log -> SetVisAttributes(VAttW);
81 }
82 //----- Musculo Esqueletico -----
83 void AcLinealConstrucion::Construct_MEsquelitico(G4LogicalVolume* mother, G4double posz){
84
85 //Se puede intentar ("G4_MUSCLE_SKELETAL_ICRP");
86 G4Material* Musculo = G4NistManager::Instance()->FindOrBuildMaterial("G4_SKIN_ICRP");
87 G4cout << Musculo << G4endl;
88 G4double musculo_tam=30.0*cm;
89
90 G4Box* ME_geo = new G4Box("BP_g", musculo_tam/2.0, musculo_tam/2.0, musculo_tam/30*0.4);
91 G4LogicalVolume* ME_log = new G4LogicalVolume(ME_geo, Musculo, "ME_l", 0,0,0);
92 new G4PVPlacement(0, G4ThreeVector(0.0*cm, 0.0*cm, posz), ME_log, "MEsquelitico", mother, false,0);
93
94 G4VisAttributes* VAttMusculo = new G4VisAttributes(G4Colour(1.0, 0.6, 0.6));
95 VAttMusculo->SetForceSolid(true);
96 ME_log->SetVisAttributes(VAttMusculo);
97 }
98 //----- Hueso cortical -----
99 void AcLinealConstrucion::Construct_HCortical(G4LogicalVolume* mother, G4double posz){
100 G4Material* Hueso = G4NistManager::Instance()->FindOrBuildMaterial("G4_BONE_CORTICAL_ICRP");
101 G4cout << Hueso << G4endl;
102
103 G4double hueso_tam=30.0*cm;
104
105 G4Box* HC_geo = new G4Box("HC_g", hueso_tam/2.0, hueso_tam/2.0, hueso_tam/30.0*0.5);
106 G4LogicalVolume* HC_log = new G4LogicalVolume(HC_geo, Hueso, "HC_l", 0,0,0);
107 new G4PVPlacement(0, G4ThreeVector(0.0*cm, 0.0*cm, posz), HC_log, "HCortical", mother, false,0);
108
109 G4VisAttributes* VAttHueso = new G4VisAttributes(G4Colour(1.0, 1.0, 1.0));
110 VAttHueso->SetForceSolid(true);
111 HC_log->SetVisAttributes(VAttHueso);
112 }
113 //----- Cerebro -----
114 void AcLinealConstrucion::Construct_Cerebro(G4LogicalVolume* mother, G4double posz){
115
116 G4Material* Cerebro = G4NistManager::Instance()->FindOrBuildMaterial("G4 BRAIN_ICRP");
117 G4cout << Cerebro << G4endl;
118
119 G4double cerebro_tam=30.0*cm;
120
121 G4Box* Cereb_geo = new G4Box("Cereb_g", cerebro_tam/2.0, cerebro_tam/2.0, cerebro_tam/10.0*0.5);
122 G4LogicalVolume* Cereb_log = new G4LogicalVolume(Cereb_geo, Cerebro, "Cereb_l", 0,0,0);
123 new G4PVPlacement(0, G4ThreeVector(0.0*cm, 0.0*cm, posz), Cereb_log, "Cerebro", mother, false,0);
124
125 G4VisAttributes* VAttCerebro = new G4VisAttributes(G4Colour(0.75, 0.76, 0.8, 0.8));
126 VAttCerebro->SetForceSolid(true);
127 Cereb_log->SetVisAttributes(VAttCerebro);
128 }
```

5.6. Notas de compilación de la simulación-ejemplo-extendida

NOTA 1: **IMPORTANTE:** Para mayor referencia, los códigos utilizados en este capítulo fueron altamente modificados del los ejemplos de **GEANT4**, particularmente del ejemplo **AnaEx02**, ubicado en la dirección: `../geant4.10.04.p02/examples/extended/analysis`.

NOTA 2: El archivo `CMakeLists.txt` necesario para enlazar los códigos y scripts requeridos por la simulación se corresponde a una modificación del archivo propio del ejemplo **AnaEx02**. Dicha modificación incluye encontrar las librerías de **ROOT** en el sistema operativo y enlazarlas con el software de **GEANT4**(ver líneas 26, 33, 42):

```
1 #-----
2 # Setup the project
3 cmake_minimum_required(VERSION 2.6 FATAL_ERROR)
4 project(BrainExpEXTENDED)
5 #
6 #
7 # Find Geant4 package, activating all available UI and Vis drivers by default
8 # You can set WITH_GEANT4_UIVIS to OFF via the command line or ccmake/cmake-gui
9 # to build a batch mode only executable
10 #
11 option(WITH_GEANT4_UIVIS "Build example with Geant4 UI and Vis drivers" ON)
12 if(WITH_GEANT4_UIVIS)
13     find_package(Geant4 REQUIRED ui_all vis_all)
14 else()
15     find_package(Geant4 REQUIRED)
16 endif()
17 #
18 #
19 # Setup Geant4 include directories and compile definitions
20 #
21 include(${Geant4_USE_FILE})
22 #
23 #
24 # Find ROOT (required package)
25 #
26 find_package(ROOT REQUIRED)
27 #
28 #
29 # Locate sources and headers for this project
30 #
31 include_directories(${PROJECT_SOURCE_DIR}/include
32                     ${Geant4_INCLUDE_DIR}
33                     ${ROOT_INCLUDE_DIRS})
34 file(GLOB sources ${PROJECT_SOURCE_DIR}/src/*.cc)
35 file(GLOB headers ${PROJECT_SOURCE_DIR}/include/*.hh)
36 #
37 #
38 #
39 # Add the executable, and link it to the Geant4 libraries
```

Simulación-ejemplo extendida: RunAction, EventAction, ROOT

```
40 #
41 add_executable(AcLinealCerebroEXTENDED AcLinealCerebroEXTENDED.cc ${sources} ${headers})
42 target_link_libraries(AcLinealCerebroEXTENDED ${Geant4_LIBRARIES} ${ROOT_LIBRARIES} )
43
44 #-----
45 # Copy all scripts to the build directory, i.e. the directory in which we
46 # build AnaEx02. This is so that we can run the executable directly because it
47 # relies on these scripts being in the current working directory.
48 #
49 set(ACLINEALCEREBROEXTENDED_SCRIPTS
50     #AnaEx02.in
51     #run.mac
52     #init_vis.mac
53     vis_BrainExp.mac
54 )
55
56 foreach(_script ${ACLINEALCEREBROEXTENDED_SCRIPTS})
57     configure_file(
58         ${PROJECT_SOURCE_DIR}/${_script}
59         ${PROJECT_BINARY_DIR}/${_script}
60         COPYONLY
61     )
62 endforeach()
63
64 #-----
65 # Install the executable to 'bin' directory under CMAKE_INSTALL_PREFIX
66 #
67 install(TARGETS AcLinealCerebroEXTENDED DESTINATION bin)
```

NOTA 4: La compilación de la simulación-ejemplo-extendida sigue los mismos pasos ilustrados en el capítulo anterior. El árbol de la carpeta de instalación BrainExpEXTENDED posee la siguiente estructura:

```
1 fisinfor@taozen:~$ pwd
2 /home/fisinfor
3 fisinfor@taozen:~$ cd geant4_workdir/BrainExpEXTENDED/
4 fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED$ ls -1F
5 AcLinealCerebroEXTENDED.cc
6 CMakeLists.txt
7 GNUmakefile
8 include/
9 src/
10 vis_BrainExp.mac
11 fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED$ ls -1FR
12 ..
13 AcLinealCerebroEXTENDED.cc
14 CMakeLists.txt
15 GNUmakefile
16 include/
17 src/
18 vis_BrainExp.mac
```

```
19 |  
20 | ./include:  
21 | AcLinealConstrucion.hh  
22 | AcLinealEventAction.hh  
23 | AcLinealFuente.hh  
24 | AcLinealListaFisica.hh  
25 | AcLinealRunAction.hh  
26 | HistoManager.hh  
27 |  
28 | ./src:  
29 | AcLinealConstrucion.cc  
30 | AcLinealEventAction.cc  
31 | AcLinealFuente.cc  
32 | AcLineallistaFisica.cc  
33 | AcLinealRunAction.cc  
34 | HistoManager.cc  
35 | fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED$
```

A partir de allí se crea la carpeta de instalación de la simulación BrainExpEXTENDED_build/ y se realiza la compilación con las utilidades cmake y make:

```
36 | fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED$ mkdir BrainExpEXTENDED_build  
37 | fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED$ cd BrainExpEXTENDED_build/  
38 | fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED/BrainExpEXTENDED_build$ cmake -DCMAKE_INSTALL_PREFIX=$G4INSTALL ../  
39 | -- The C compiler identification is GNU 5.4.0  
40 | -- The CXX compiler identification is GNU 5.4.0  
41 | -- Check for working C compiler: /usr/bin/cc  
42 | -- Check for working C compiler: /usr/bin/cc -- works  
43 | -- Detecting C compiler ABI info  
44 | -- Detecting C compiler ABI info - done  
45 | -- Detecting C compile features  
46 | -- Detecting C compile features - done  
47 | -- Check for working CXX compiler: /usr/bin/c++  
48 | -- Check for working CXX compiler: /usr/bin/c++ -- works  
49 | -- Detecting CXX compiler ABI info  
50 | -- Detecting CXX compiler ABI info - done  
51 | -- Detecting CXX compile features  
52 | -- Detecting CXX compile features - done  
53 | -- Configuring done  
54 | -- Generating done  
55 | -- Build files have been written to: /home/fisinfor/geant4_workdir/BrainExpEXTENDED/BrainExpEXTENDED_build  
56 | fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED/BrainExpEXTENDED_build$  
57 | fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED/BrainExpEXTENDED_build$ make  
58 | Scanning dependencies of target AcLinealCerebroEXTENDED  
59 | [ 12%] Building CXX object CMakeFiles/AcLinealCerebroEXTENDED.dir/AcLinealCerebroEXTENDED.cc.o  
60 | [ 25%] Building CXX object CMakeFiles/AcLinealCerebroEXTENDED.dir/src/AcLinealFuente.cc.o  
61 | [ 37%] Building CXX object CMakeFiles/AcLinealCerebroEXTENDED.dir/src/AcLinealRunAction.cc.o  
62 | [ 50%] Building CXX object CMakeFiles/AcLinealCerebroEXTENDED.dir/src/AcLinealConstrucion.cc.o  
63 | [ 62%] Building CXX object CMakeFiles/AcLinealCerebroEXTENDED.dir/src/AcLinealEventAction.cc.o  
64 | [ 75%] Building CXX object CMakeFiles/AcLinealCerebroEXTENDED.dir/src/HistoManager.cc.o  
65 | [ 87%] Building CXX object CMakeFiles/AcLinealCerebroEXTENDED.dir/src/AclineallistaFisica.cc.o  
66 | [100%] Linking CXX executable AcLinealcerebroEXTENDED  
67 | [100%] Built target AcLinealCerebroEXTENDED  
68 | fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED/BrainExpEXTENDED_build$
```

obteniéndose la estructura de archivos de la carpeta de instalación así:

Simulación-ejemplo extendida: RunAction, EventAction, ROOT

```
69 fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED/BrainExpEXTENDED_build$ ls -1F
70 AcLinealCerebroEXTENDED*
71 CMakeCache.txt
72 CMakeFiles/
73 cmake_install.cmake
74 Makefile
75 vis_BrainExp.mac
76 fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED/BrainExpEXTENDED_build$
```

producido así el archivo ejecutable de la simulación `AcLinealCerebroEXTENDED`, que se ejecuta así:

```
77 fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED/BrainExpEXTENDED_build$ ./AcLinealCerebroEXTENDED vis_BrainExp.mac
78 ...
79 ...
80 The number of events kept by the vis manager can be changed with
81 "/vis/scene/endOfEventAction accumulate <N>", where N is the
82 maximum number you wish to allow. N < 0 means "unlimited".
83 Graphics systems deleted.
84 Visualization Manager deleting...
85 fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED/BrainExpEXTENDED_build$
```

La respuesta de la ejecución del ejecutable presenta tres nuevos archivos, dos archivos `.csv` comentados anteriormente y un archivo adicional con extensión del software ROOT `BrainExp.root`:

```
86 fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED/BrainExpEXTENDED_build$ ls -1F
87 AcLinealCerebroEXTENDED*
88 BrainExp.root
89 CMakeCache.txt
90 CMakeFiles/
91 cmake_install.cmake
92 EnerDepo.csv
93 Makefile
94 NOFCollision.csv
95 vis_BrainExp.mac
96 fisinfor@taozen:~/geant4_workdir/BrainExpEXTENDED/BrainExpEXTENDED_build$
```

5.7. Resultados de la simulación-ejemplo-extendida

El resultado adicional de la simulación-ejemplo-extendida atañe a la información consignada en el archivo `BrainExp.root`, que para el ejercicio que ocupa este documento, contiene información de la deposición de energía por evento en la simulación; es decir, un histograma el cual se llama `EnergyDeposited_per_event`.

Para obtener la información de este archivo `.root` se debe realizar una ejecución de ROOT en la terminal y abrir la interfaz gráfica de usuario de ROOT. Esto permitirá visualizar el

electrones), en un material³.

La curva de la Fig. 5.3, conocida como el poder de frenado en función de la distancia, también se menciona como la curva de Bragg, que junto con la Eq. 5.1 indica que el poder de frenado de los electrones aumenta lentamente con la profundidad de penetración, debido a que su dirección cambia drásticamente por ser partículas cargadas eléctricamente (Elford y Cunningham, 1983).

A medida que la tasa de variación de la energía disminuye con la distancia, la perdida de energía específica aumenta de acuerdo con el modelo de Bethe-Bloch, Eq. 2.14; sin embargo, importante destacar que existen alternativas de enfoques analíticos y estadísticos que se centran en la descripción de la curva de Bragg (Levin y Gonin, 1988), por lo que una descripción más completa de estas curvas, requiere de un análisis paramétrico que involucra no solo conceptos fuera del alcance de este trabajo, sino también métodos complejos y en constante desarrollo de los mismos.

En la práctica la curva de Bragg proporciona información para dirigir la radiación de forma tal que alcance un objetivo específico (por ejemplo una región que contiene un tumor), a determinada profundidad, evitando dañar o afectar tejidos cercanos.

En el caso de emplear haces de electrones se nota un pico centrado, siendo el caso óptimo aquel en el que el pico se encuentre, abruptamente, en el lado derecho de la gráfica a la distancia especificada. De allí que las radioterapias de electrones se utilicen

³La distribución de Landau se caracteriza por ser una distribución asimétrica⁴, con un pico y un achatamiento hacia valores positivos (debido al pequeño número de colisiones individuales, cada una con una probabilidad de transferencia de energía, relativamente grande). La definición matemática de la función de densidad de probabilidad esta dada por la Eq. (5.2), naturalmente la distribución será (5.3)

$$\phi(\lambda) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} e^{s \log s + \lambda s} ds \quad (5.2)$$

$$\Phi(\lambda) = \int_{-\infty}^{\lambda} \phi(\lambda) d\lambda \quad (5.3)$$

La ecuación (5.3) no puede ser escrita en una forma analítica, sin embargo para usos prácticos, y teniendo en cuenta parámetros de la fórmula de Bethe-Bloch, se utiliza la expresión aproximada deducida por J.E. Moyal en 1955 (Moyal, 1955), de donde resulta la ecuación (5.4), siendo $\lambda = \frac{z-\mu}{\xi}$, y $\xi = k \frac{Z}{A} \frac{\rho}{\beta^2} z$.

$$\Phi(\lambda) \approx \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\lambda + e^{-\lambda})} \quad (5.4)$$

muy cerca de la superficie a tratar, por ejemplo tumores externos (como los cutáneos), y no para atacar zonas profundas (Bevelacqua, 2004).

Por otro lado, haciendo uso de la información consignada el archivo `BrainExp.root` se encuentra que el valor mas probable de absorción de energía corresponde a 0.026 MeV⁵ con un número de 48 entradas o eventos (ver Apéndice H).

Se muestra entonces que la piel y el músculo son materiales que absorben un valor de energía menor que el hueso craneal, actuando este como un poderoso absorbente de electrones, de donde se deduce que las radioterapias con *LINAC*, para blancos cercanos, permiten allegar la radiación secundaria (rayos *X*) al cerebro.

En contraste con otros estudios, los valores de deposición de energía resultado de la simulación, concuerdan cualitativamente con los reportes experimentales citados en Pazdur y cols. (2003) de la *Office of Oncology Drug Products*, del *Center for Drug Evaluation and Research* y del *Curtis and Elizabeth Anderson Cancer Institute*.

Es preciso señalar que la pérdida de energía en el blanco está determinada, además, por el potencial de excitación, en este caso para un material compuesto, y por la energía de los electrones incidentes, según como lo modela la fórmula de Bethe-Bloch Eq. (2.14).

Bethe-Bloch implica suposiciones que no son siempre válidas, en concreto, Bethe supuso que los electrones del medio están en reposo y la energía de ligadura de los mismos es despreciable (Capítulo 2), además, que la trayectoria de la partícula no se ve afectada por la interacción con los electrones del medio, llevando a inconsistencias para bajas energías. A pesar de estas consideraciones, para energías relativamente altas, la ecuación de Bethe-Bloch satisface en muy buena aproximación el comportamiento de los electrones en el blanco.

Las radiaciones secundarias producidas en el blanco, con energías suficientes para ionizar moléculas cercanas, hasta el punto de la extinción total de radiación, contribuyen también en la medición de la deposición de energía, o si se prefiere, del poder de frenado del material.

La Fig. 5.4 describe el comportamiento del número acumulado de colisiones por bin de longitud.

⁵Aquí se hace uso de varios métodos de extracción de información para un objeto histograma de tipo `TH1F`, con métodos como `GetEntries()`, `GetMaximumBin()`, `GetNbinsX()`, `GetXaxis()->GetXmin()`.

Perspectivas y Conclusiones

La formación en investigación de alto nivel se vuelve extremadamente costosa por la falta de recursos educativos que acerquen conocimientos extremadamente complejos a la formación del investigador; de allí que los tiempos de aprendizaje de recursos computacionales para la investigación científica sean largos y repetitivos en cuanto a que una investigación que necesito saber unos conocimientos particulares (por ejemplo el conocimiento del uso de un software) no permita la transferencia de conocimiento a otra investigación que también requiera dicha base de conocimiento.

Se presenta entonces un marco conceptual sobre lo que significa el proceso de Alfabetización Científica y Tecnológica (ACyT), orientado a la formación de investigadores de alto nivel, donde se propone como un proceso de liberación que incluye la autogestión de conocimiento interdisciplinario, multidisciplinario y transdisciplinario de diferentes áreas de conocimiento.

Este recurso educativo, "Radiación-Materia: *GEANT4Hands On!*", como resultado de investigación formativa en ACyT, orientada a profesores e investigadores, constituye una de las pocas referencias cuyo objetivo es la formación en investigación en el software científico *GEANT4*, con miras al desarrollo de aplicaciones relacionadas con los fenómenos de interacción radiación-materia en diferentes campos de conocimiento de las ciencias básicas e ingeniería.

Se enfatiza, como se menciona en el prólogo, para asegurar un impacto positivo en el profesor-investigador-lector, una lectura de todo el documento, es decir, una lectura inicial, para sensibilizar el cerebro sobre signos, lenguaje y significación propios de la ACyT orientada a profesores-investigadores en el uso, manejo y desarrollo de simulaciones en *GEANT4*, para después hacer una lectura práctica (*Hands On!*) haciendo uso reiterativo del computador.

Cabe anotar que el diseño, codificación y resultados de la simulación-ejemplo que presenta este recurso, simulación de un LINAC para tratamiento de radioterapia por

Simulación-ejemplo extendida: RunAction, EventAction, ROOT

electrones, ha sido utilizado como recurso educativo en la orientación de cursos sobre uso y manejo de *GEANT4*, en los que se ha tenido una respuesta satisfactoria con estudiantes de pregrado de Licenciatura en Física de la Universidad Distrital, y algunas asesorías para estudiantes de posgrado en Física y Física Médica de diferentes universidades, asintiendo tiempos de curvas de aprendizajes entre 3 y 5 meses.

Se espera que ya con los resultados de investigación presentados en este documento, el profesor-investigador pueda, en su procesos de autogestión, aprender sobre el uso y manejo de *GEANT4* en tiempos de 1 a 2 meses (teniendo en cuenta las características del público objeto, como se describe en el prólogo), en contraste, a tiempos mayores a 7 meses de aprendizaje del software, como lo declaran desde su propia experiencia los autores de este documento, sin que se tenga un recurso educativo pensado que lleve consigo un diseño metodológico y una estructura didáctica-pedagógica basada en el público objeto, el software y sus relaciones de interlocución.

Apéndice A

Interacción de la radiación electromagnética con la materia

A.1. Efecto Fotoeléctrico

A finales del siglo XIX una serie de experimentos pusieron de manifiesto que una superficie metálica emite electrones cuando es irradiada con luz ultravioleta; este fenómeno, denominado efecto fotoeléctrico, descubierto por Hertz en 1887 (al observar que una descarga eléctrica entre dos electrodos ocurre más fácilmente cuando sobre uno de ellos incide luz UV), y estudiado por Lenard en 1900, requirió de más de veinte años de resultados experimentales que recopilaron completamente, y sin ninguna explicación teórica, sus características fundamentales. En 1905 Einstein propuso para el efecto fotoeléctrico un mecanismo que, basado en la teoría de Planck de los cuantos de energía, permitiera explicar los hechos experimentales (ver Fig. A.1). Mediante el dispositivo mostrado en la Fig. A.2 y de los trabajos Millikan se obtienen los siguientes resultados experimentales:

- Existe una frecuencia umbral (ν_0) para la cual se produce emisión de fotoelectrones (electrones desprendidos de la placa A en la Fig. A.2), de acuerdo al material de la placa radiada.
- Al incrementar el potencial acelerador V , la photocorriente (corriente eléctrica inducida por fotoelectrones)¹ alcanza un valor constante, de saturación, independiente de V .
- Al iluminar la placa metálica, con una frecuencia por encima de ν_0 , se induce rápidamente ($\approx 10^{-9}$ s) la photocorriente, incluso para bajas intensidades.

¹Experimentalmente la photocorriente producida es del orden de 10^{-10} A, por lo que se suele utilizar un amplificador de corriente que permite hacer mediciones proporcionales a cierto valor, en la Fig. A.2 se ha supuesto que el amperímetro tiene incorporado dicho amplificador.

Interacción de la radiación electromagnética con la materia

- La photocorriente de saturación es proporcional a la intensidad de la radiación incidente; El potencial de frenado V_0 permanece constante, siendo independiente de la intensidad.
- V_0 depende linealmente de la frecuencia de la radiación incidente; a mayor frecuencia mayor es debe ser el potencial de frenado necesario para que la photocorriente sea nula.

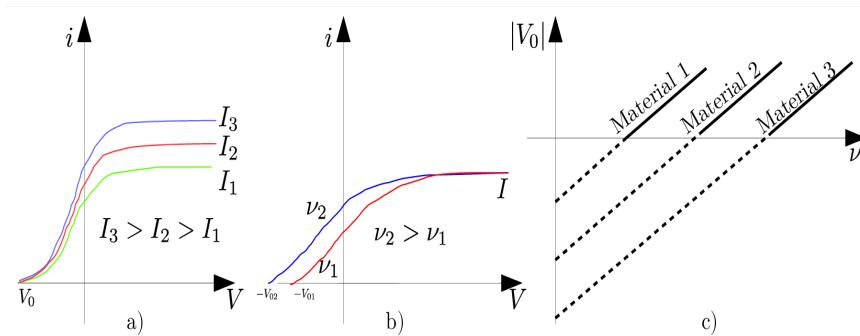


Figura A.1: Curvas Experimentales para el Efecto fotoeléctrico: a) Radiación de diferentes intensidades e igual frecuencia; b) Radiación de diferentes frecuencias e igual intensidad; c) Potencial de frenado en función de la frecuencia para diferentes materiales

Así, al considerar la teoría electromagnética clásica no se puede explicar este fenómeno en concordancia con los resultados empíricos, pues de ella se deduce que: 1) la energía de los electrones debe aumentar con la intensidad (i.e $K \propto I \equiv V_0 \propto I$); 2) La energía de los electrones disminuye al aumentar la frecuencia; 3) Entre el instante en que se irradia la placa metálica y el que se produce la photocorriente debe transcurrir un cierto intervalo de tiempo, medible (\sim minutos); 4) no existe una frecuencia umbral bajo la cual no se pueda observar la producción de fotoelectrones, es decir el efecto fotoeléctrico se puede observar para cualquier frecuencia.

Ninguna de estas “predicciones” se observa experimentalmente, al contrario, contradicen dichos resultados. Einstein demostró que estas experiencias podían entenderse suponiendo que la energía radiada no se distribuye de manera continua, sino cuántica, asumiendo que el fotoelectrón emitido desde la superficie de la placa metálica es de alguna forma “desprendido” por el impacto con el fotón, de forma que toda la energía del fotón pasa al electrón. Ahora bien, el electrón recibe su energía de un único fotón; así cuando se aumenta la intensidad de la luz lo que sucede es que al incidir más fotones por

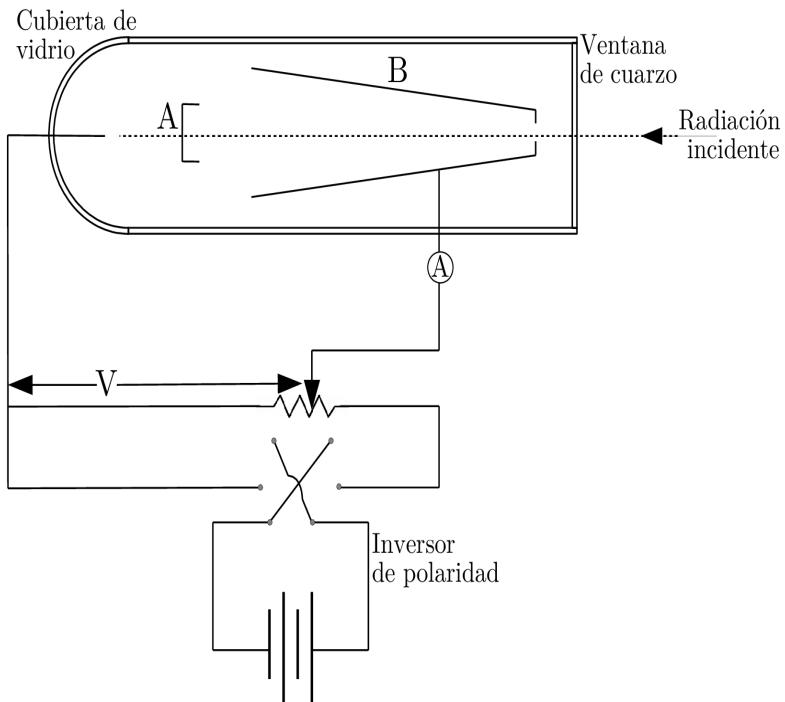


Figura A.2: Esquema del arreglo experimental para observar el efecto fotoeléctrico.

unidad de tiempo, sobre la placa queda un mayor número de electrones libres, sin embargo la energía que ha absorbido cada electrón no varía, es la misma; siguiendo argumentos de conservación de la energía, la máxima energía cinética que puede tener un fotoelectrón está dada por (A.1).

$$K_{\max} = h\nu - \phi_0 \quad (\text{A.1})$$

Siendo (ϕ_0) la función trabajo o energía de enlace del electrón, y que representa la energía mínima con la cual el electrón está ligado al metal.

En relación con el experimento de la Fig. A.2, se sabe que la energía cinética de un electrón se relaciona con el potencial acelerador como $K = eV$. Valga tener en cuenta que cuando un fotón incide sobre sobre el material, le cede a un electrón la energía necesaria para vencer la atracción que el núcleo ejerce sobre él y además para salir a la superficie del material, a través de la distribución atómica del mismo; la primera energía se

llama de ionización y la segunda es propiamente la función trabajo.

Una característica de todo material es la llamada energía de Fermi ² E_F ; cuando dos materiales de diferente E_F están próximos se crea un potencial de contacto (potencial que se crea al poner dos materiales en contacto). Si no se tiene en cuenta este potencial se estará introduciendo un error sistemático en la medición de la energía máxima de los fotoelectrones (Castañeda y De-Geus, 2008); sin embargo cuando los electrodos o placas son de un mismo material, tendrán igual E_F y no habrá potencial de contacto, en tal caso es válida la expresión A.1.

Centrándonos en el efecto fotoeléctrico propiamente como mecanismo de interacción, se sabe que este proceso domina a energías menores 1 MeV, y que se puede considerar como la interacción de un haz de fotones con la nube electrónica de los átomos que componen un material, como resultado de esta, los fotones son totalmente absorbidos emergiendo fotoelectrones de la superficie del material ($\gamma + \text{átomo} \rightarrow \text{átomo}^+ + e^-$); esto es, un fotón interacciona con un electrón atómico ³ (Leo, 1994), que absorbe toda su energía “desapareciendo” al fotón, y expulsando al electrón con un exceso de energía cinética $h\nu - \phi_0$, más del 80 % de las interacciones ocurre con electrones de la capa K (Ayala, 2008). La probabilidad de ocurrencia del efecto es mayor cuanto más fuerte está ligado el e^- al núcleo, es decir que para un átomo es más probable el evento con electrones de la capa K, L, M ⁴ que con los de las capas más externas.

El fotoelectrón viajará 1 o 2 mm a través del material, creando pares de iones de otros átomos hasta ceder toda su energía. El fotoelectrón, ya en “reposo”, es incorporado a un ion positivo, mientras que el átomo residual (un ion positivo) queda con una vacancia en una de sus capas internas; esto provoca una reordenación electrónica, pudiéndose llenar la vacancia: **1)** con un electrón de las capas superiores, dejando a su vez una vacancia en su capa correspondiente y produciendo la emisión secundaria de un fotón característico cuya energía corresponde a la diferencia de energías entre ambos niveles (rayo X característico); **5** o **2)** con un electrón de la capa externa del mismo átomo y expulsando electrones fuera del átomo (e^- Auger), ver Fig. A.3.

²La Energía de Fermi es la energía del nivel más alto ocupado por un sistema cuántico a temperatura 0 K

³Puesto que un electrón libre no puede absorber un fotón y también conservar el momento, el efecto fotoeléctrico se produce siempre con electrones ligados al núcleo, el cual absorbe el impulso de retroceso.

⁴Para átomos multi-electrónicos se dice que los electrones residen en “capas”, que corresponden a los niveles cuánticos principales. Los estados de mayor energía, asociados a electrones fuertemente ligados al núcleo, son los del nivel $n = 1$ capa K, $n = 2$ capa L, $n = 3$ capa M, etc.

⁵La energía de los rayos X característicos puede ser calculada mediante la ley de Moseley, $E = R_y(Z - 1)^2 \left(\frac{1}{n^2} - \frac{1}{m^2} \right)$, en donde $R_y = 13.6$ eV es la constante de Rydberg y, n y m son los números cuánticos principales característicos de las capas atómicas.

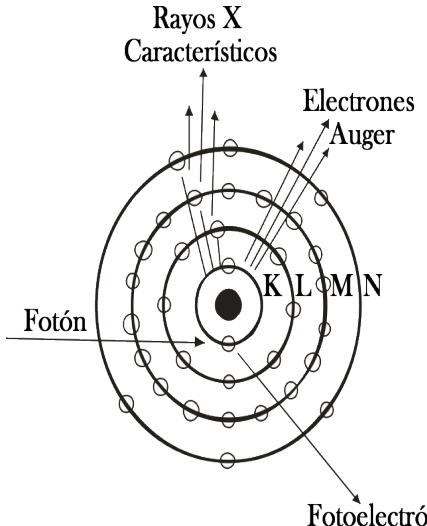


Figura A.3: Representación pictórica del efecto fotoeléctrico.

El coeficiente de absorción mísico disminuye al aumentar la energía de los fotones, es decir, fotones de alta energía son más penetrantes que radiación de baja energía. Para un valor determinado de energía el coeficiente de atenuación aumenta con Z , esto se describe mediante la relación empírica (A.2), en donde $E_{fotón}$ es la energía de los fotones incidentes. En la Fig. A.4 se muestra cómo la atenuación varía con la energía de los fotones incidentes, presentando picos de absorción, característicos para cada elemento.

$$\frac{\mu}{\rho} \propto E_{fotón}^{-3.5} Z^{4.5} \quad (\text{A.2})$$

Teóricamente, el efecto fotoeléctrico, es difícil de tratar rigurosamente a causa de la complejidad de las funciones de onda de Dirac para electrones atómicos (Leo, 1994). Sin embargo suponiendo fotones con energías por encima de la energía de la capa K, en un régimen no relativista, es decir $h\nu \leq m_e c^2$, la sección eficaz puede ser estimada bajo la aproximación de Born, obteniéndose (A.3) por átomo.

$$\sigma_{fotoeléctrico} = 4\alpha\sqrt{2}Z^5\sigma_0 \left(\frac{m_e c^2}{h\nu} \right)^{7/2} \quad (\text{A.3})$$

Donde, $\sigma_0 = \frac{8\pi r_e^2}{3}$ y $\alpha = \frac{e^2}{4\pi\hbar c} \approx \frac{1}{137}$ es la constante de estructura fina.⁶ Dado que

⁶La constante de estructura fina o de Sommerfeld caracteriza la interacción electromagnética, esto es, entre

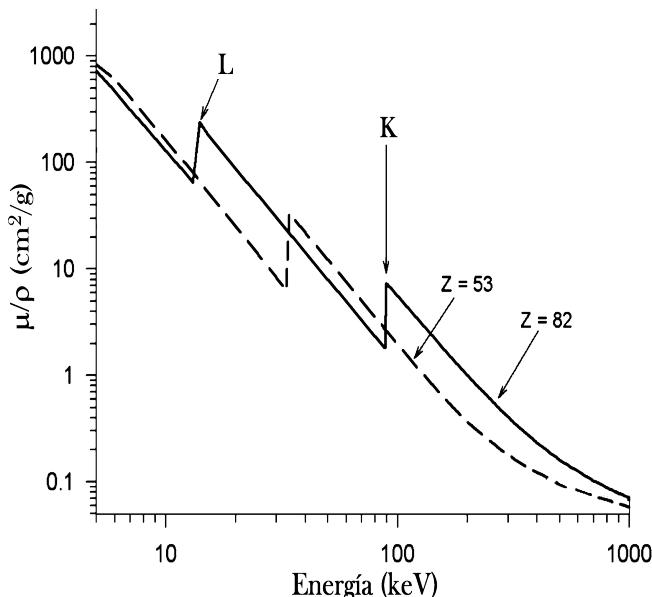


Figura A.4: Variación del coeficiente de absorción másico para dos materiales distintos en función de la energía característica de Rayos X emitidos en el efecto fotoeléctrico.

$\sigma_{fotoeléctrico}$ varía en forma aproximada como Z^5 , materiales con alto Z resultan ser buenos absorbentes de fotones, pudiendo ser utilizados con propósitos de protección para fotones de baja energía, $\sim 10^2$ keV.

Entre los valores más precisos que se disponen de $\sigma_{fotoeléctrico}$ están los obtenidos numéricamente por Storm e Israel (Montesinos, 2001). Estas secciones eficaces se presentan en forma de tablas para cada Z y para cada energía del fotón, contando así con una base de datos distinta para cada elemento; estas bases de datos se utilizan ampliamente en los códigos profesionales de Montecarlo para la simulación de fenómenos en los que domina el efecto fotoeléctrico como mecanismo de interacción.

A.2. Efecto Compton

En el estudio de la dispersión de rayos X por la materia, Compton mostró, en 1923, que cuando un haz de rayos X monocromático es dispersado por un elemento ligero, como partículas cargadas y fotones,

el grafito, la radiación dispersa se “constituye” en dos haces, uno con igual longitud de onda que la radiación incidente λ y otro con longitud de onda λ' ligeramente superior. Al medir λ con un espectrómetro de rayos X tipo Bragg (Kaplan, 1964) (ver Fig. A.5), Compton observó que la diferencia $\Delta\lambda$ entre ambas radiaciones variaba según el ángulo θ de dispersión, aumentando cuando el valor de este ángulo es grande; así, cuando el ángulo entre la radiación incidente y la dispersada era de 90° se tenía que $\Delta\lambda = 0.0236 \times 10^{-10}$ m, independientemente de la longitud de onda de la radiación incidente y de la naturaleza del material utilizado como blanco dispersor. En la Fig. A.6 se observan los resultados experimentales, para algunos ángulos de dispersión, según el experimento de Compton.

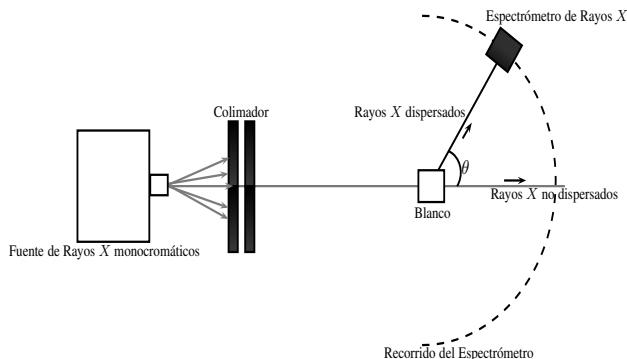


Figura A.5: Esquema del dispositivo experimental para la observación del efecto Compton.

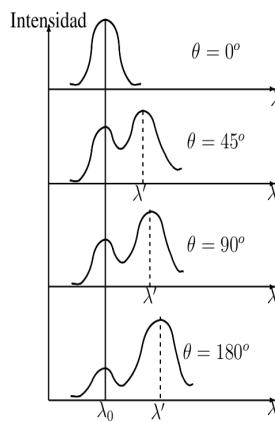


Figura A.6: Corrimiento Compton para diferentes ángulos de dispersión.

Interacción de la radiación electromagnética con la materia

La presencia de una longitud de onda dispersada λ' no es posible entenderla bajo la teoría electromagnética clásica, pues en esta, el vector de campo eléctrico de la onda incidente, con frecuencia ν , actuaría sobre los electrones libres del blanco de forma tal que estos oscilaren con la misma frecuencia ν . En consecuencia los electrones “oscilantes” radiarían ondas electromagnéticas, que tienen nuevamente frecuencia ν ; por lo tanto la onda dispersada debería tener la misma frecuencia y longitud de onda que la onda incidente, lo cual es contrario a los resultados experimentales de Compton.

Los resultados experimentales, según Compton e independientemente Debye (Eisberg y Resnick, 1985), se pueden interpretar como la colisión elástica entre un fotón y un electrón libre en reposo, dado que la interacción no depende del blanco dispersor;⁷ de acuerdo con este punto de vista los fotones de “retroceso” que emergen del blanco constituyen la radiación dispersada (ver Fig. A.7), pues el fotón incidente transfiere energía al electrón con el cual interactúa, así, el fotón dispersado un ángulo θ de la dirección del fotón incidente, deberá tener una energía menor E' , lo cual implica una longitud de onda mayor $\lambda' = c/\nu'$, lo cual confirma el carácter corpuscular de la radiación.

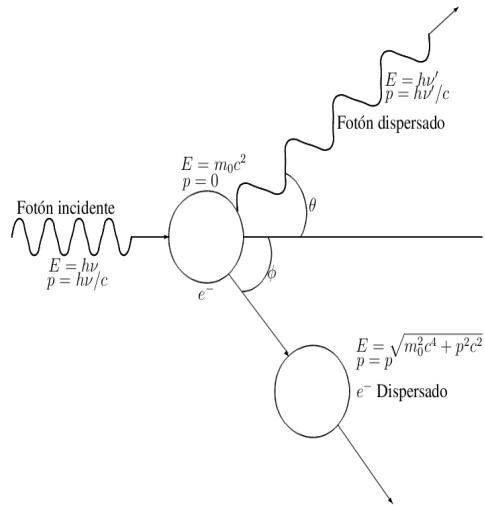


Figura A.7: Esquema para la interpretación de la colisión inelástica fotón-electrón del efecto Compton.

⁷ Esta suposición es válida debido a que la energía de un fotón de rayos X es mucho mayor que la energía que mantiene unido al electrón en el material del blanco dispersor.

A partir de principios de conservación de la energía y del momento lineal se obtiene la Eq. (A.4) que describe el corrimiento Compton; en una descripción mas minuciosa del efecto, se requiere un tratamiento rigurosamente relativista, pese a que en la Eq. (A.4) aparezca el término m_0 .

$$\Delta\lambda = \lambda - \lambda' = \frac{h}{m_0 c} (1 - \cos \theta) \quad (\text{A.4})$$

Nótese que $\frac{h}{m_0 c}$ corresponde a 0,0242 Å; la energía del electrón de retroceso esta dada por (A.5) y la energía del fotón dispersado por (A.6).

$$E_e = h\nu_0 \frac{(1 - \cos \theta)\alpha}{1 + \alpha(1 - \cos \theta)} \quad (\text{A.5})$$

$$E_{\text{fotón}'} = \frac{h\nu_0}{1 + \alpha(1 - \cos \theta)} \quad (\text{A.6})$$

con $\alpha = h\nu_0/m_0 c^2$.

Al detectar los electrones en retroceso, se observa que sus energías concuerdan con los valores de (A.5), lo cual explica también la variación observada en la longitud de onda, pero no la presencia de la línea espectral no modificada. Esto se debe a que en el análisis previo se supuso que los fotones incidentes interactúan con electrones libres (caso en el que es válida la Eq. (A.4)); se debe tener en cuenta que los electrones están ligados a los átomos, necesitándose una cantidad de energía adicional para arrancarlos de estos. Si la cantidad de energía que el fotón le cede al electrón es mucho mayor que el trabajo necesario para separarlo del átomo, el electrón se comporta como libre. En caso de que la colisión ocurra de tal forma que el electrón no se desprenda del átomo, la masa en reposo m_0 del electrón en (A.4) debe sustituirse por la del átomo, resultando $\Delta\lambda$ muy pequeño para ser detectado; en consecuencia un fotón que interactué con un electrón ligado no sufre variación en su longitud de onda, lo que explica la presencia de líneas espectrales no desplazadas (Kaplan, 1964).

El ángulo θ puede tomar todos los valores posibles desde 0 hasta π , dependiendo su probabilidad relativa de la energía de los fotones incidentes; de acuerdo con (A.5) el electrón de retroceso tendrá una energía comprendida entre 0 y $\frac{4h^2\nu^2}{1+4h\nu}$ correspondiente a $\theta = \pi$.

El análisis realizado para este mecanismo de interacción (dominante para valores intermedios de energía entre 0.5 – 10.0 MeV aproximadamente, y caracterizado por $\gamma + e \rightarrow \gamma' + e'$), se refiere al caso en que se produzca un proceso aislado de dispersión Compton; para estudiar la contribución de este efecto a la atenuación de un haz de rayos γ en

la materia, es necesario calcular la probabilidad de que ocurra un proceso de este tipo. La probabilidad de ocurrencia del efecto no se puede calcular con sencillez, puesto que depende de los detalles de la interacción fotón-electrón, concretamente de la relación entre la polarización de la radiación y la dirección del espín del electrón difusor (Burcham, 1979). Un cálculo realizado por Klein y Nishina aplicando los principios de la mecánica cuántica relativista, muestra que la distribución angular de dispersión está dada por (A.7), y la sección eficaz total por electrón por (A.8),⁸ en la Fig. A.8 se muestra la sección eficaz como función del ángulo de dispersión.

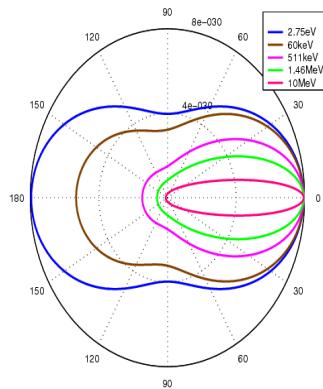


Figura A.8: Diagrama polar de la sección eficaz diferencial en función de θ para el efecto Compton.

$$\frac{d\sigma_{Compton}}{d\Omega} = Zr_0^2 \left(\frac{1}{1 + \alpha(1 - \cos \theta)} \right)^2 \left(\frac{1 + \cos^2 \theta}{2} \right) \left(1 + \frac{\alpha^2(1 - \cos^2 \theta)^2}{(1 + \cos^2 \theta)[1 + \alpha(1 - \cos \theta)]} \right) \quad (\text{A.7})$$

$$\sigma_{Compton} = 2\pi r_0^2 \left\{ \frac{1 + \alpha}{\alpha^2} \left[\frac{2(1 + \alpha)}{1 + 2\alpha} - \frac{1}{\alpha} \ln(1 + 2\alpha) \right] + \frac{1}{2\alpha} \ln(1 + 2\alpha) - \frac{1 + 3\alpha}{(1 + 2\alpha)^2} \right\} \quad (\text{A.8})$$

con $r_0 = \frac{e^2}{8\pi\varepsilon_0 m_e c^2} \approx 2.818 \times 10^{-15} \text{ m}$ el radio clásico del electrón.

La probabilidad de que un fotón salga del haz por dispersión al atravesar un espesor absorbente de 1.0 cm está dada por el coeficiente de absorción, definido como

⁸Para bajas energías, esto es $E_{fotón} = h\nu \ll m_e c^2$ la expresión (A.8) se reduce a la propuesta por Thomson, es decir, $\sigma_e = \frac{8\pi}{3} r_0^2$, lo cual implica que $h\nu = h\nu'$.

$\sigma = \rho N \frac{Z}{A} \sigma_e$, el cual mide también la cantidad total de energía perdida por el haz por unidad de recorrido (Kaplan, 1964). Para una energía dada de los fotones, el coeficiente de absorción másico $\frac{\sigma}{\rho} = N \frac{Z}{A} \sigma_e$ en elementos ligeros resulta prácticamente constante (ver Fig. A.9) puesto que Z/A tiende a 1/2. Para conocer la energía de los fotones dispersados, o la absorbida por los electrones de retroceso se emplean dos términos auxiliares, el de *sección eficaz de dispersión Compton* $\sigma_{Compton_s}$, y el de *sección trasversal de absorción Compton* $\sigma_{Compton_a}$; siendo la sección eficaz total la definida en (A.8) y que corresponde a $\sigma_{Compton_s} + \sigma_{Compton_a}$.

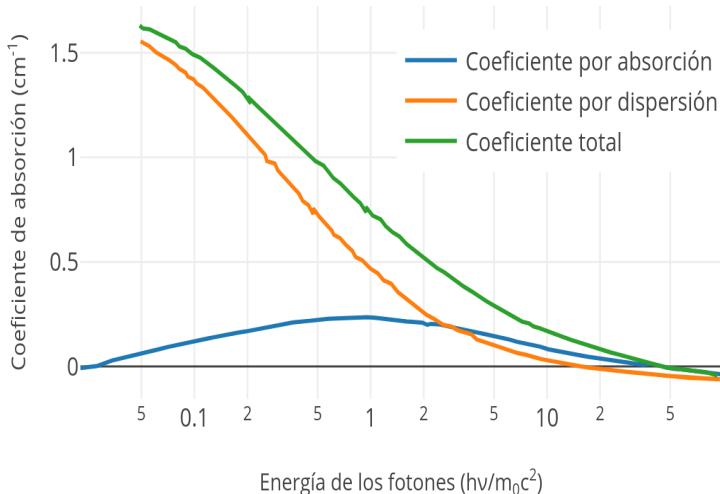


Figura A.9: Coeficientes de absorción por centímetro en función de la energía de los fotones, por efecto Compton en el Plomo.

La fórmula de Nishina-Klein, definida en (A.8), es válida únicamente para la dispersión de fotones debido a la interacción con electrones libres; si la energía del fotón es mucho mayor que la energía de enlace de los electrones atómicos, los efectos debidos a esta “unión” son pequeños. Si la energía es pequeña, existe una alta probabilidad de que el electrón de retroceso se mantenga “ligado” al átomo después de la colisión. Cuando el átomo, en su conjunto, adquiere la energía y la cantidad de movimiento transferida al electrón la interacción se denomina dispersión de Rayleigh (dispersión elástica coherente), si la interacción expulsa al electrón del átomo, esta se dice que es una dispersión inelástica incoherente, i.e. dispersión Compton.

La dispersión coherente se observa para bajas energías de la radiación incidente, por debajo de 30 keV (Azorín, 2009); en ésta, un fotón incidente interactúa con un átomo como un “todo”, el fotón no cede ni adquiere energía, y es dispersado en pequeños ángulos. En la dispersión de Thomson (dispersión elástica coherente), la radiación electromagnética, desde la perspectiva clásica del electromagnetismo, se dispersa de manera elástica por una partícula cargada libre, siendo el límite de baja energía para la dispersión Compton, en donde la energía cinética de las partículas y la frecuencia de los fotones permanecen invariantes frente a la dispersión. De acuerdo con el formalismo del electromagnetismo clásico la sección eficaz para las dispersiones elásticas, de Rayleigh y de Thomson, se describe mediante $\sigma_e = \frac{8\pi}{3} r_0^2 \frac{\omega^4}{(\omega^2 + \omega_0^2)^2}$ donde ω_0 es la frecuencia angular natural con la que oscilan los electrones y ω es la frecuencia angular con la cual es dispersada la onda electromagnética, en el caso de la dispersión de Rayleigh $\omega \ll \omega_0$, en la dispersión de Thomson se tiene que $\omega_0 \rightarrow 0$.

Un tratamiento más completo del efecto Compton, da como resultado consideraciones de orden superior en las cuales surgen procesos como la dispersión Compton doble (*soft-hard photonic Bremsstrahlung*) (Denner y Dittmaier, 1998), o la creación de pares Compton (Carrillo, 2006).

A.3. Producción y aniquilación de pares

El tercer mecanismo mediante el cual la radiación interactúa con la materia es la producción y aniquilación de pares electrón-positrón, constituyendo un ejemplo de conservación de la energía que no tiene análogo en la física clásica. La explicación de este proceso se basa en la conclusión de que los electrones poseen estados de energía negativa (resultado del estudio de la ecuación de onda relativista por Dirac), siendo los valores posibles de energía de un electrón, mayores que $+m_0c^2$ o menores que $-m_0c^2$, no existiendo ningún valor intermedio (ver Fig. A.10).

En 1933 Anderson, estudiando los rayos cósmicos mediante una cámara de niebla, observó partículas que tenían igual masa que el electrón y carga eléctrica de igual magnitud pero de signo contrario que la de este, a esta partícula se le dio el nombre de positrón (ver Fig. A.11).

De acuerdo con Dirac la producción de un positrón puede interpretarse como la interacción entre un fotón, con energía mayor a $2m_0c^2$, que arranca un electrón de un estado de energía negativa y lo traslada a otro de energía positiva, así, la desaparición del electrón que se hallaba en un estado de energía negativa deja un lugar vacío que se traduce en la aparición de un positrón; por otro lado el cambio de estado de un electrón

a un estado de energía positiva supone la aparición de un electrón ordinario, creándose un par de partículas como resultado global del proceso (ver Fig. A.12). Una vez formado un positrón este va perdiendo velocidad por colisión con los átomos hasta que queda prácticamente en reposo, momento en que puede interaccionar con un electrón que esté en el mismo estado, desapareciendo ambas partículas, y apareciendo en su lugar dos fotones que se mueven en direcciones opuestas con una energía de 0.511 MeV, igual a la energía de un electrón en reposo (ver Fig. A.13).

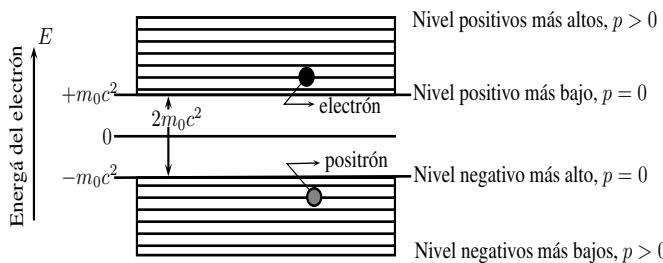


Figura A.10: Representación esquemática de los valores posibles de la energía de un electrón de acuerdo con la teoría de Dirac.

El proceso de creación de pares suele tener lugar bajo el campo eléctrico ⁹ de un núcleo atómico ($\gamma + \text{núcleo} \rightarrow e^+ + e^- + \text{núcleo}'$), pues debe haber algún cuerpo al que se le transfiera una energía de retroceso y una cantidad de movimiento para que se cumplan las leyes de conservación respectivas.

La producción de pares electrón-positrón en el campo de Coulomb de un núcleo sólo es posible si la energía del fotón supera una energía umbral, dada por las masas en reposo de dos electrones más la energía de retroceso que se transfiere al núcleo; empleando principios de conservación de la energía y de la cantidad de movimiento, se obtiene el umbral de energía, calculado como $E_{\text{fotón}} \geq 2m_0c^2 + 2\frac{m_0^2}{m_{\text{núcleo}}}c^2$; para masas nucleares grandes, $m_{\text{núcleo}} \gg m_0$ el umbral se puede aproximar a $E_{\text{fotón}} \geq 2m_0c^2$. Sin embargo, si se considera un proceso de producción electrón-positrón en el campo de Coulomb de un electrón, la energía umbral estará dada por $E_{\text{fotón}} \geq 4m_0c^2$, (en tal caso se tendrá $\gamma + e^- \rightarrow e^+ + e^+ + e^-$).

Empleando la ecuación de Einstein ($E^2 - p^2c^2 = m^2c^4$) se obtiene la energía total del proceso, indicada en (A.9), siendo K_+ y K_- las energías cinéticas del positrón y del electrón respectivamente, y $m_0c^2 \approx 0.511 \text{ MeV}$ la energía en reposo del electrón. El ángulo promedio

⁹La producción de pares electrón-positrón únicamente ocurre en campos eléctricos de partículas cargadas.

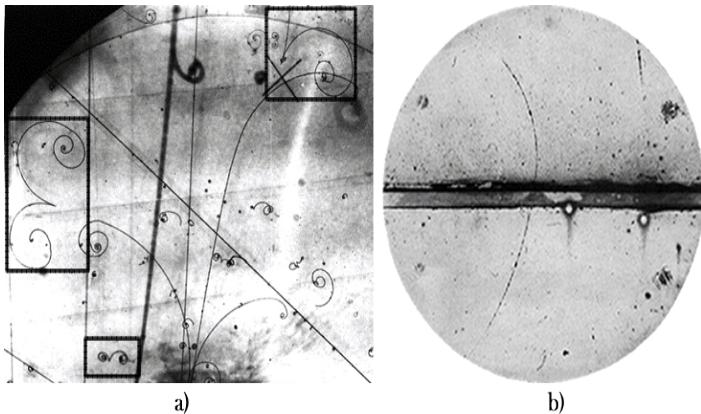


Figura A.11: a) Imagen de cámara de burbujas (Fermilab, Universidad de Birmingham), en los recuadros se observan dos espirales de sentido contrario que representan la producción de un par electrón-positrón derecha, bajo un campo magnético perpendicular al plano de la imagen. b) Imagen de Anderson (*Phys. Rev.*, 43, 491, 1933) con una cámara de niebla en la que se muestra un positrón de 63 MeV que pasa a través de una lámina de plomo y emerge con una energía de 23 MeV.

en el que son expulsados el positrón y el electrón, respecto a la dirección de incidencia del fotón se muestra en (A.10), con \bar{K} la energía cinética promedio del electrón y del positrón, esto es (A.11).

$$h\nu = m_+c^2 + m_-c^2 + 2m_0c^2 = (K_- + m_0c^2) + (K_+ + m_0c^2) \quad (\text{A.9})$$

$$\bar{\theta} = \frac{m_0c^2}{\bar{K}} = \frac{2m_0c^2}{h\nu - 2m_0c^2} \quad (\text{A.10})$$

$$\bar{K} = \frac{h\nu - 2m_0c^2}{2} \quad (\text{A.11})$$

El cálculo de la sección eficaz para este mecanismo debe ser considerado en dos casos: 1) en el que la carga nuclear no está “envuelta” por electrones atómicos,¹⁰ con $1 \ll h\nu/m_0c^2 \ll \frac{1}{\alpha Z^{1/3}}$ siendo la sección eficaz la indicada en (A.12); 2) en el que existe un solapamiento de campo eléctrico producido por la carga nuclear y los electrones

¹⁰Para bajas energías del fotón, este debe estar relativamente cerca del núcleo para que sea probable la producción de pares, lo que significa que el fotón interactúa con el núcleo como un proceso aislado.

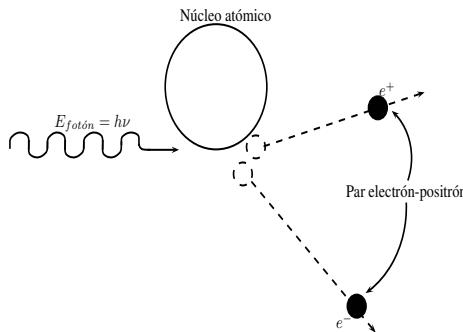


Figura A.12: Esquema pictórico de la producción de un par electrón-positrón.

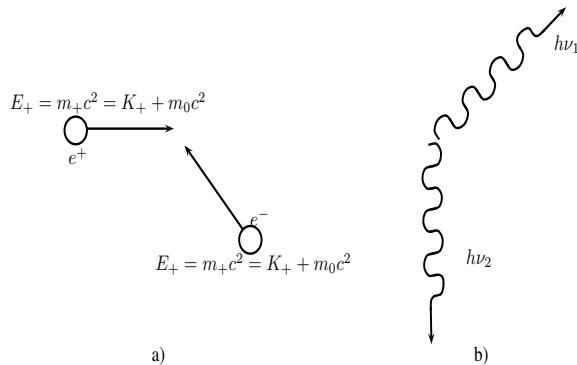


Figura A.13: Esquema pictórico de la aniquilación de un par electrón-positrón.

atómicos, lo que implica que $h\nu \gg \frac{1}{\alpha Z^{1/3}}$, obteniéndose la sección eficaz como (A.13).

La sección eficaz diferencial para la producción de un positrón, cuya energía varía entre E_+ y $E_+ + dE_+$, con un electrón de energía total E_- , está dada por (A.14), ver Fig. A.14; mientras que el coeficiente de atenuación μ_{pares} para este mecanismo, varía con la energía del fotón incidente y el número atómico del absorbente como $\mu \propto \rho Z^2 \ln(\frac{h\nu}{E_{\text{referencia}}})$, donde $E_{\text{referencia}}$ es una energía de referencia arbitraria (Brand y cols., 1993).

$$\sigma_{\text{pares}} = 4\alpha r_0^2 Z^2 \left(\frac{7}{9} \ln \frac{2h\nu}{m_0 c^2} - \frac{109}{54} \right) \{ \text{cm}^2/\text{átomo} \} \quad (\text{A.12})$$

$$\sigma_{pares} = 4\alpha r_0^2 Z^2 \left(\frac{7}{9} \ln \frac{183}{Z^{1/3}} - \frac{1}{54} \right) \{cm^2/\text{átomo}\} \quad (\text{A.13})$$

$$\frac{d\sigma_{pares}}{dE_+} = \frac{\alpha r_0^2}{h\nu - 2m_0c^2} Z^2 f \left(\frac{h\nu}{m_0c^2}, Z \right) \{cm^2/(MeV \cdot \text{átomo})\} \quad (\text{A.14})$$

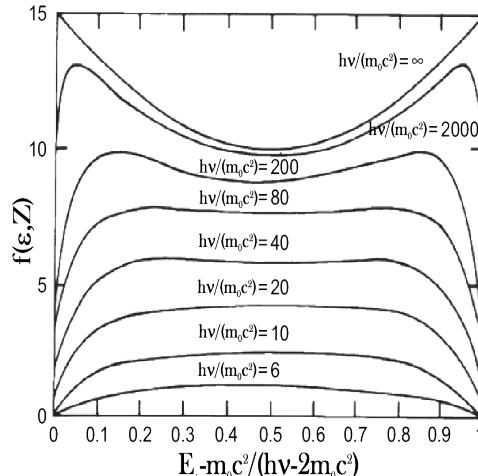


Figura A.14: Forma de la función de partición de la energía $f(\varepsilon, Z)$ con $\frac{h\nu}{m_0c^2}$ como parámetro. La sección eficaz total de la producción de pares está dada por el área bajo la curva correspondiente, en unidades de $Z(Z + 1)\alpha r_0^2$.

Nota sobre la interacción de fotones

La propiedad fundamental de la absorción de fotones por la materia es el carácter exponencial de la disminución de la intensidad de la radiación, esto es, si un haz homogéneo de fotones atraviesa una lámina delgada de un material. Así, cuando una haz de fotones de intensidad I incide sobre una lámina de espesor Δx , la variación de la intensidad al atravesar dicho material es proporcional a su espesor y a la intensidad de la radiación incidente como $\Delta I = -\mu I \Delta x$; la intensidad del haz se reducirá a medida que pasa a través del material debido a que los fotones serán absorbidos o dispersados según una combinación de efecto fotoeléctrico, efecto Compton y producción de pares.

La atenuación de la intensidad obedece la ley exponencial (A.15), siendo μ una constante que depende del material y de la energía de la radiación, dadas las interacciones que predominan a diferentes energías.¹¹

$$I(x) = I_0 e^{-\mu x} \quad (\text{A.15})$$

La probabilidad de interacción de los mecanismos tratados hasta ahora es independiente entre ellos, por lo que la probabilidad de interacción total será la suma de las probabilidades de interacción individual (Leo, 1994), esto es, $\sigma_{Total} = \sigma_{fotoeléctrico} + \sigma_{Compton} + \sigma_{pares}$; consecuentemente el coeficiente de absorción es la suma de los coeficientes de absorción de cada proceso, como en (A.16), ver Fig. A.15.

$$\mu_{Total} = \mu_{fotoeléctrico} + \mu_{Compton} + \mu_{Pares} \quad (\text{A.16})$$

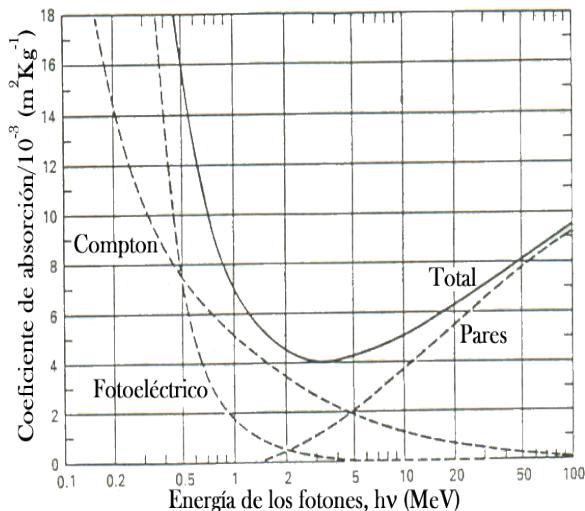


Figura A.15: Coeficiente de absorción total en el plomo, para fotones de alta energía indicando las contribuciones por efecto fotoeléctrico, efecto Compton y producción de pares.

¹¹Una manera alternativa de expresar la intensidad es $I = Bh\nu$, con B el número de fotones que atraviesan la unidad de superficie por unidad de tiempo (flujo), obteniéndose $\frac{B}{B_0} = e^{-\mu x}$.

Apéndice B

Descripción de los principales sistemas de detección de radiación

B.1. Detectores de ionización gaseosa

La facilidad relativa de movimiento de electrones y de iones en un gas, hace que este sea un medio idóneo para la captación de señales eléctricas inducidas por la ionización del gas cuando una radiación interactúa con este. Los fenómenos de ionización han sido ampliamente estudiados, siendo la base para el diseño y construcción de diversos detectores de partículas; los tres dispositivos originales basados en la ionización de un gas son: la cámara de niebla, el contador proporcional y el contador Geiger-Müller (ver Fig. B.1); estos detectores son en realidad el mismo dispositivo, pero operando bajo diferentes parámetros.



Figura B.1: a) Cámara de niebla de Charles Wilson; b) Contador proporcional comercial manufacturado por J.E. Manson CO., Inc. (Vaiana, 2011); c) Contador comercial de Geiger-Müller manufacturado por Detectron Corporation.

Descripción de los principales sistemas de detección de radiación

La configuración básica de un detector de ionización gaseosa consiste en un recipiente cilíndrico lleno de un gas, a determinada presión, con paredes conductoras y una ventanilla delgada al final del recipiente.¹ A lo largo del eje principal se suspende un hilo conductor sometido a una tensión $+V_0$, en relación con las paredes, generando un campo eléctrico radial descrito por (B.1) (ver Fig. B.2); esta configuración puede ser entendida como dos electrodos aislados, entre los cuales hay una diferencia de potencial.²

$$E = \frac{1}{r} \frac{V_0}{\ln \frac{b}{a}} \quad (\text{B.1})$$

r es la distancia radial, b el radio interno del cátodo y a el radio externo del ánodo. Al incidir radiación sobre la cámara cilíndrica, se ionizará el gas debido a las interacciones de las partículas incidentes con las partículas cargadas del medio: electrones y núcleos atómicos; con ello se creará un cierto número de pares electrón-ion, de forma directa, si la radiación incidente está compuesta por partículas cargadas, o indirecta, mediante reacciones secundarias, si la radiación incidente se compone de partículas neutras. En virtud del campo eléctrico del detector, las partículas cargadas, libres, se pondrán en movimiento hacia el electrodo de signo contrario, esto es, los electrones se acelerarán hacia el ánodo y los iones hacia el cátodo. De esta forma se origina en el detector una corriente eléctrica que puede ser amplificada y medida con un amperímetro; la señal obtenida depende de la intensidad del campo eléctrico y es independiente de la diferencia de potencial entre electrodos, siendo proporcional al número de iones creados y por tanto, a la energía depositada por la radiación.

El potencial de ionización de los gases es del orden de decenas de eV por molécula (ver Tabla B.1); así una partícula incidente con energía del orden de MeV, podrá occasionar la ionización de miles o millones de moléculas de gas. Para describir la cinemática de los iones y/o moléculas dentro de la cámara, se puede obtener de la teoría cinético molecular, la velocidad media con que estos se mueven. La energía cinética de los iones, por efectos térmicos y de acuerdo con el teorema de equipartición de la energía, es $\frac{3}{2}k_B T$, por lo que la velocidad media de un ión estará dada por (B.2); de otra parte si λ es el camino libre medio de los iones, el tiempo medio para dos colisiones sucesivas será $\Delta t = \lambda/v_t$. Dado que sobre el sistema actúa un campo eléctrico externo, generado por el hilo conductor axial al cilindro, los iones se moverán en la dirección del campo eléctrico, con una velocidad definida por (B.3).

¹ Esta ventanilla es una pared esencialmente transparente a la radiación, para facilitar la entrada de la misma al recipiente gaseoso. La atenuación que sufre la radiación depende directamente del ancho de la ventanilla.

² Por lo general se utiliza un gas noble como el argón. Al ser un "buen" aislante, este no permite, bajo condiciones normales, el paso de corrientes eléctricas apreciables entre los electrodos.

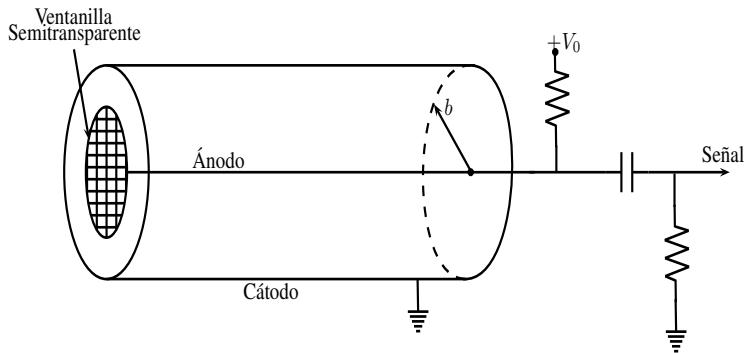


Figura B.2: Configuración esquemática de un detector de ionización gaseosa.

$$v_t = \sqrt{\frac{3k_B T}{M}} \quad (\text{B.2})$$

$$v = \frac{a}{2} \Delta t = \frac{eE}{2M} \frac{\lambda}{v_t} = \frac{e\lambda}{\sqrt{12k_B T M}} E \quad (\text{B.3})$$

Gas	E (eV/par iónico)
H_2	37
He	41
N_2	35
O_2	31
Aire	35
Ne	36
Ar	26

Tabla B.1: Energía media necesaria para formar un par electrón-ión en diferentes gases (Tomado de (Udías y cols., 2011)).

Si la diferencia de potencial entre los electrodos es pequeña, los electrones y los iones se pueden recombinar poco después de producirse la ionización, sólo una pequeña fracción de estos alcanzará los electrodos respectivos, generando una señal correspondiente a un menor número de pares iónicos electrón-ión; el rango de voltaje en el que esto sucede, se conoce como *región de recombinación*.

Descripción de los principales sistemas de detección de radiación

Aumentando la diferencia de potencial por encima del punto en que los pares iónicos no se pueden recombinar, se obtiene una señal que representa la cantidad total de iones producidos, este rango de operación se llama *región de ionización*. Una diferencia de potencial aún mayor proporciona a los electrones, liberados inicialmente, energía suficiente para ionizar átomos adicionales del gas, este aumento en la ionización se interpreta como una amplificación de la señal; la señal de salida, en esta región, es mayor pero proporcional a la ionización producida inicialmente, por lo que se le llama *rango proporcional*.

Al incrementar aún más la diferencia de potencial, se tiene una “avalancha” de pares iónicos (región del contador Geiger-Müller), la energía de los electrones, producto de la ionización, aumenta lo suficiente como para excitar o ionizar más átomos, originando más electrones libres y/o fotones, esto, a su vez, produce más pares electrón-ión, y finalmente se genera una descarga eléctrica, es decir, la señal de salida será tan alta, que su tamaño no dependerá de la cantidad de energía depositada por la ionización inicial. El aumento de la tensión más allá del rango del contador Geiger-Müller genera una ruptura del dieléctrico produciendo descargas continuas, por lo que el detector ya no será sensible a la ionización provocada por la radiación incidente (ver Fig. B.3).

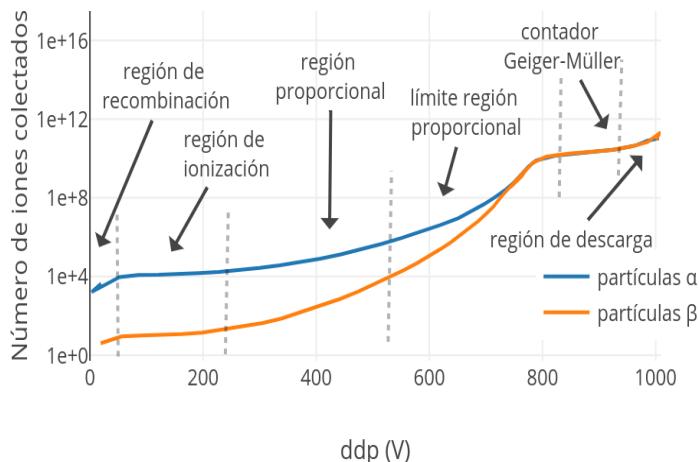


Figura B.3: Número de iones recolectados en función de la diferencia de potencial en una cámara de ionización gaseosa.

B.2. Detectores de centelleo

Son diversos los detectores de radiación cuyo funcionamiento no se basa en la ionización de un gas. Uno de los más empleados actualmente son los detectores de centelleo. En éstos se aprovecha el hecho de que la radiación produce pequeños destellos luminosos al incidir sobre ciertas sustancias. Cuando a este detector se acopla un dispositivo amplificador, usualmente un tubo *fotomultiplicador*, estos destellos pueden generar impulsos eléctricos detectables, los cuales al ser analizados y contados electrónicamente proporcionan información sobre la radiación incidente.

Los detectores de centelleo tienen su origen en el espantariscopio, inventado por Crookes en 1903 (Leo, 1994); este instrumento disponía de una pantalla de *ZnS* que en interacción con partículas alfa produce tenues destellos luminosos, detectables a simple vista, dando cabida así, a los contadores de centelleo mediante el “conteo” de destellos de luz.

Con la invención de los detectores de ionización gaseosa, el contador de centelleo óptico cayó en desuso rápidamente, sin embargo mediante este aparato Geiger y Marsden, bajo dirección de Rutherford, realizaron los famosos experimentos de dispersión de partículas alfa. En 1944 Curran y Baker reemplazaron el ojo humano por el tubo fotomultiplicador recién desarrollado, de esta manera el débil centelleo ahora se podía contar con una eficiencia y fiabilidad igual a la de los instrumentos de ionización gaseosa, dando origen al detector de centelleo moderno. Los nuevos desarrollos y mejoras que han seguido desde 1950 hasta hoy, hacen de los detectores de centelleo uno de los más fiables y convenientes en diversos experimentos.

Cuando la interacción de la radiación ionizante con la materia provoca defectos reversibles en la red cristalina del material, mediante la excitación o ionización de un gran número de moléculas, estas alcanzan un estado energético mayor, y al volver espontáneamente a la configuración de equilibrio se emiten fotones en el espectro visible o cerca del rango de energía visible; este fenómeno, denominado “luminiscencia” o centelleo, tiene lugar especialmente con sustancias cristalinas luminiscentes o materiales centelladores, que pueden ser gaseosos, líquidos o sólidos, orgánicos o inorgánicos (Knoll, 2000).

Los detectores de centelleo se componen esencialmente de un material centellador y un fotodetector, bien sea un tubo fotomultiplicador (PMT) o un fotodiodo (ver Fig. B.4.c), en ambos casos este último se basa en el efecto fotoeléctrico. Los fotodetectores son fabricados con materiales metálicos, compuestos usualmente por aleaciones de

Descripción de los principales sistemas de detección de radiación

cesio-plata o semiconductores, con un potencial de ionización muy bajo, de modo que los fotones incidentes sean capaces de arrancar electrones de la red cristalina; los tubos fotomultiplicadores son los fotodetectores más comunes, siendo recipientes de vidrio sellados y al alto vacío, se componen de un photocátodo seguido por una serie de electrodos denominados dinodos.

Cuando los fotones resultantes de la luminiscencia del centelleador chocan con el photocátodo, se emiten fotoelectrones que son acelerados en etapas sucesivas mediante dinodos, buscando un efecto multiplicativo al arrancar más electrones de éstos (la presencia de 9 dinodos da origen a una cascada de $10^6\text{-}10^7$ electrones por cada fotón incidente (Pérez, 1996)). Finalmente la cascada de fotones se transduce en una señal eléctrica amplificada, y se recoge en el ánodo, direccionándola hacia los circuitos de medición.

Cabe tener en cuenta que el número de fotones producido dentro de la red cristalina del centelleador es directamente proporcional a la cantidad de energía absorbida, y por tanto también lo es el pulso eléctrico obtenido a la salida del fotomultiplicador; de este modo, se puede realizar una discriminación energética de las partículas incidentes. La producción de fotones, e indirectamente del pulso eléctrico, hace referencia, entonces, a la energía depositada dentro del centelleador, lo que permite determinar la energía correspondiente a las partículas incidentes.

Dada la mayor densidad de un sólido que la de un gas, este resulta más eficiente para detener la radiación, especialmente rayos γ , lo cual establece una ventaja de los detectores de centelleo sobre los de ionización gaseosa. Asimismo, el proceso de luminiscencia, i.e., la absorción de radiación y la posterior emisión de luz, es muy rápida, lo que disminuye el tiempo muerto del detector (Campbell y Ross, 1995).

B.3. Detectores de neutrones

Dado que los neutrones no actúan de manera apreciable sobre electrones atómicos, no producen ionización directa en la materia, de allí que los detectores de neutrones se basen en efectos secundarios debidos a interacciones nucleares o dispersiones, en las que se transfiere energía y/o momento de un neutrón incidente a un núcleo.

Como se mencionó en el Capítulo 2, existen 6 formas de interacción de los neutrones, todas ellas pueden emplearse y se emplean en la detección de neutrones. Los detectores de ionización gaseosa, de centelleo y los contadores proporcionales no son sensibles a este tipo de radiación (Campbell y Ross, 1995), por lo que deben ser modificados; en general

Descripción de los principales sistemas de detección de radiación

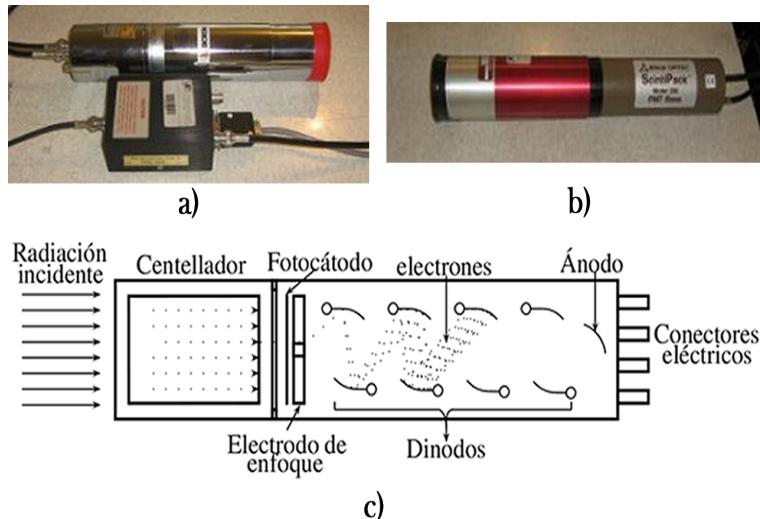


Figura B.4: a) Tubo centellador de poliestireno (Laboratorio de radiación gamma y alpha, Gfnun); b) Tubo centellador de *NaI* (Laboratorio de radiación gamma y alpha, Gfnun) (Cristancho, 2011); c) Diagrama esquemático de un detector de centelleo que comprende un material de centelleo acoplado a un tubo fotomultiplicador.

para que un detector pueda dar cuenta de la interacción de neutrones con la materia, debe estar diseñado de manera tal que las dispersiones o las reacciones nucleares sean abundantes, en todo caso la señal obtenida del detector proporciona una medida de la ionización secundaria.

Los neutrones rápidos, con energías por encima de 1 eV, 0.1 MeV ó 1 MeV (Daoud, s.f.), generalmente son detectados por las dispersiones que producen al interactuar con los núcleos (Campbell y Ross, 1995); los neutrones lentos, cuyas energías no superan los 0.4 eV, se detectan por las reacciones nucleares que puedan provocar.

Los núcleos ligeros como los del hidrógeno o el carbono, al colisionar con un neutrón, adquieren suficiente energía para ser detectados, en particular los centelladores orgánicos que contienen gases ligeros son útiles en la detección neutrones rápidos (Sanz, 1970).

De otra parte, dado que los neutrones lentos se detectan de acuerdo a las reacciones nucleares que produzcan, se utilizan en los detectores, materiales que reaccionen nuclearmente y con alta probabilidad al paso de neutrones. Así por ejemplo se pueden

Descripción de los principales sistemas de detección de radiación

detectar neutrones mediante un contador proporcional o Geiger que contenga una alta concentración de BF_3 (trifluoruro de boro) (Campbell y Ross, 1995), pues este gas produce una cascada de partículas cargadas de fácil detección. También suelen usarse detectores de gas que contienen 3He , LiI , 6Li para provocar reacciones secundarias.

Otros detectores, con el fin de aprovechar el proceso de fisión, tienen un recubrimiento interno de ^{235}U ; los contadores proporcionales diseñados de esta forma son llamados cámaras de fisión, en éstas los neutrones incidentes causan la fisión a dicho material y los productos de fisión, con fuerte carga eléctrica y altamente energéticos (Sanz, 1970), producen una ionización detectable; en este tipo de detectores la sensibilidad respecto a una u otra clase de neutrones, varía de acuerdo con la proporción de isótopos de uranio en el revestimiento del detector.

B.4. Detectores semiconductores

Sin entrar en detalles acerca de la teoría de semiconductores se debe mencionar que estos materiales se caracterizan por estar en un estado eléctrico intermedio entre aislantes y conductores, su resistividad, a temperatura ambiente, es mayor que la de la plata, pero al someterlos a una diferencia de potencial adecuada conducen electricidad.

La condensación de un gran número de átomos, del orden de 10^{23} en un material sólido, implica desde la mecánica cuántica ondulatoria, un traslape de funciones de onda electrónicas entre átomos cercanos (Blatt, 1983); los electrones de los primeros orbitales se ven poco afectados por la condensación de átomos, y sus niveles de energía permanecen invariantes bajo esta consideración; de otro lado las funciones de onda de los electrones de valencia pueden solaparse considerablemente respondiendo a una teoría de átomos multielectrónicos. Como resultado, los niveles de energía para los electrones de valencia se “ensanchan” formando un espectro de energía estrechamente espaciado (Kittel, 1995), conocido como banda de energía. Se suelen emplear diagramas de bandas en los cuales se representa, esquemáticamente, el estado ligado o libre de los electrones dentro del semiconductor.

Los electrones ligados tienen un rango de energías asociadas a la banda de valencia, es decir que para liberar un electrón de dicho estado, es necesario “suministrarle” cierta cantidad de energía (ancho de la banda prohibida E_g), para que pase a un estado libre, moviéndose a la banda de conducción; el “lugar” dejado por el electrón en la banda de valencia, denominado hueco, tiene un movimiento opuesto al del electrón. En la Fig. B.5 se presenta el diagrama de bandas para materiales aislantes, semiconductores y metales de acuerdo con (Kittel, 1995), en ésta se nota que el valor de E_g es mayor para un aislante,

Descripción de los principales sistemas de detección de radiación

un poco menos para un semiconductor y casi nulo para los conductores; el valor de la energía prohibida para materiales semiconductores como el silicio es de 1.1 eV, este valor de energía es cercano al promedio de la energía de un fotón de la región visible del espectro electromagnético.

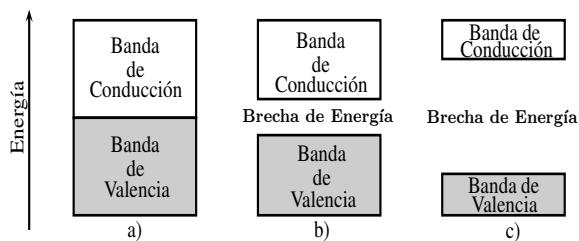


Figura B.5: Diagrama de bandas para: a) un metal, b) un semiconductor, c) un aislante.

La conductividad de los semiconductores es proporcional a la temperatura, pues por excitación térmica un mayor número de electrones pasan de la banda de valencia a la de conducción. En la práctica, para mejorar la conductividad eléctrica a temperaturas determinadas de los semiconductores, se utilizan impurezas añadidas voluntariamente, este proceso se denomina dopaje.

Aprovechando el fenómeno de conducción que se presenta en los materiales semiconductores, bajo determinadas condiciones energéticas, se diseñan los *detectores semiconductores*. Estos se basan en un proceso similar al que ocurre en las cámaras de ionización gaseosa, sin embargo su funcionamiento es totalmente distinto; en ambos tipos de detectores se provoca excitación sobre electrones atómicos, de modo que estos obtengan movilidad y puedan ser colectados mediante un juego de electrodos. En los detectores semiconductores el material es un semiconductor sólido, en lugar de un gas. En el estado base, el material semiconductor no conduce electricidad, pero la incidencia de radiación produce la aparición de una gran cantidad de pares electrón-hueco, que inducen una corriente eléctrica medible.

El principio de funcionamiento de un detector semiconductor es el siguiente: si una partícula ionizante penetra en el detector, ésta produce pares electrón-hueco a lo largo de su trayectoria, el número de pares creados es proporcional a la pérdida de energía por unidad de recorrido (Bock y Vasilescu, 1998). Mediante un campo eléctrico, aplicado externamente, se separan los pares antes de que se recombinen, de esta manera los electrones se dirigen hacia el ánodo y los huecos hacia el cátodo. La carga colectada

Descripción de los principales sistemas de detección de radiación

en los electrodos produce un pulso de corriente, y la suma de pulsos es igual a la carga total por unidad de tiempo, generada por la partícula incidente, es decir, es una medida de la energía depositada en el detector. La señal de salida pasa a través de un sistema pre-amplificador, seguido por un amplificador de pulsos obteniéndose la lectura final.

Una ventaja de los detectores semiconductores sobre los de ionización gaseosa, es que la energía necesaria para producir un par electrón-hueco es de un factor de 1 a 10 veces menor. Actualmente todos los detectores semiconductores se utilizan sin ningún tipo de amplificación integrada (Leo, 1994). Como resultado, las señales de los detectores semiconductores son muy pequeñas y de bajo nivel de ruido electrónica.

B.5. Ejemplo de aplicación de los detectores de partículas

Los detectores de partículas se pueden emplear con diferentes fines y en distintas ramas de la física de acuerdo con las propiedades a medir o que resulten de interés en un experimento. Como caso particular, aunque de gran interés en el mundo de la física de partículas, consideremos el experimento ATLAS (*A Toroidal LHC ApparatuS*), el cual hace parte del conjunto de los cuatro experimentos realizados en el LHC (*Large Hadron Collider*³), en CERN (*Organisation Européenne pour la Recherche Nucléaire*).

ATLAS es uno de los cinco detectores de partículas, junto con ALICE, CMS, TOTEM y LHCb, construidos en el Gran Colisionador de Hadrones (ver figuras B.6, B.7a, B.7b), y es uno de los dos detectores multipropósito del mismo, siendo el detector de partículas más grande jamás construido (Cern, 2011a). En el proyecto de ATLAS están implicados cerca de 2900 científicos e ingenieros de 172 instituciones pertenecientes a 37 países diferentes; otra descripción del detector se muestra a continuación:

- Tamaño: 46 m de largo, 25 m de altura y 25 m de ancho
- Masa: 7000 toneladas
- Diseño: Cilíndrico más tapas finales
- Ubicación: Meyrin, Suiza

³El Gran Colisionador de Hadrones (LHC) es un gigantesco instrumento científico ubicado cerca de Ginebra, frontera entre Francia y Suiza, a unos 100 m bajo tierra. Se trata de un acelerador de partículas, el más grande en la actualidad tanto en infraestructura como en energía alcanzada en la aceleración, utilizado para estudiar las partículas más pequeñas conocidas. Entre los años 2011 y 2013 se descubre una partícula con masa de unos 125 GeV/c² la cuál se denominó como el "Boson de Higgs" (ó "partícula de Dios"), que permite "completar" el Modelo Estándar de Partículas.

Descripción de los principales sistemas de detección de radiación

Con este experimento se espera detectar partículas masivas no detectables anteriormente, bajo otros experimentos como el Tevatrón y el LEP que operaban a energías menores; con los resultados de ATLAS se espera también tener un soporte experimental que permita poner a “prueba” las nuevas teorías físicas más allá del modelo estándar de partículas; entre los tópicos investigación se incluyeron (en su momento) la búsqueda del bosón Higgs, dimensiones adicionales, y partículas que podrían conformar la materia oscura.

ATLAS cuenta con seis subsistemas de detección de partículas, mediante los cuales se identifican las partículas y se miden sus energías junto con otras variables de interés (Cern, 2011a); tras la interacción de los haces de partículas en el centro del detector (ver Fig. B.6), se produce una cascada de partículas con un amplio rango de energías, sin embargo ATLAS ha sido diseñado para que mida el mayor intervalo posible de energías, pues se pretende que sea cual sea el proceso producido o las partículas generadas, sean eventos detectables y medibles. Los sistemas de análisis de datos constan de un disparador de información para detectar los eventos más interesantes entre los 40 millones de haces que chocan cada segundo en el centro del detector (Cern, 2011a), produciendo 100 Mb de información por segundo, un equivalente a 1Pb por año.

El detector ATLAS consta de una serie de cilindros concéntricos de diversos tamaños que rodean el punto en el cual interactúan las partículas, es decir donde colisionan los haces. Tiene cuatro partes principales: el detector interno, los calorímetros, el espectrómetro muónico y los imanes externos; cada parte se subdivide a su vez en más partes. Los detectores son complementarios: el detector interno determina la trayectoria de cada partícula (detectores de seguimiento), los calorímetros (ver Fig. 2.8) miden la energía de las partículas poco penetrantes, y los sistemas muónicos miden parámetros adicionales de los muones penetrantes.⁴ Los imanes externos deflectan las trayectorias de las partículas detectadas en el detector interno y en el espectrómetro muónico, permitiendo medir su momento lineal (Cern, 2011b). Los neutrinos son las únicas partículas no detectables directamente (Cern, 2011a).

En el siguiente listado se hace una descripción breve de cada una de las partes principales de las que se compone ATLAS:

- **Detector interno:** se encuentra ubicado a pocos centímetros del eje de colisión, y se

⁴El muon es una partícula “elemental” del grupo de los Leptones (ver Fig. 2.1) descubierta por Carl Anderson en 1936, que responde a interacciones gravitacionales, electromagnéticas, e interacciones débiles. Tiene espín 1/2 y posee carga eléctrica negativa, con una masa 200 veces mayor que la del electrón y una vida media de 2,2 μ s (Yao, 2006).

Descripción de los principales sistemas de detección de radiación

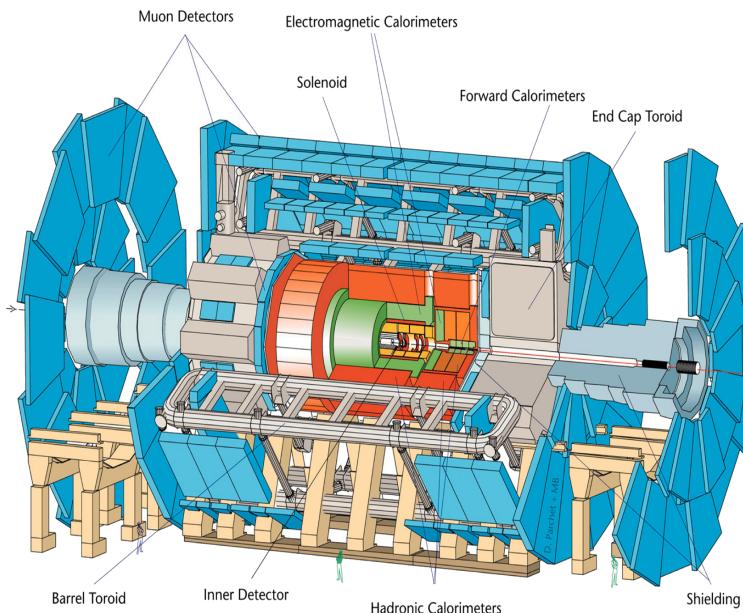


Figura B.6: Representación pictórica del detector ATLAS (Grillo, 2011).

extiende 1,2 m alrededor del mismo; tiene 7 m de longitud a lo largo del conducto del haz y su principal función es determinar la trayectoria de las partículas cargadas midiendo su interacción con la materia, revelando así el tipo de partículas producidas y su momento lineal. El detector interno se encuentra en presencia de un campo magnético uniforme que provoca que las partículas cargadas sigan trayectorias curvas; la dirección las curvas revela la carga de la partícula, y el radio de curvatura su momento lineal.

El detector interno se compone a su vez de:

- Un detector de píxel (detector semiconductor), formado por 1744 módulos electrónicos, cada uno de $6 \times 2\text{cm}$ con 16 circuitos integrados. El material detector es una película de silicio de $250\mu\text{m}$, y la mínima unidad mensurable es un píxel de 50×400 micras.
- Un rastreador semiconductor, similar al detector de píxel en cuanto a diseño y funcionamiento. Sin embargo cada módulo electrónico es más largo y delgado, con cuatro capas dobles de silicio, permitiendo cubrir un área total de 61 m^2 .
- Un detector de radiación de transición (TRT; equivalente a los detectores de

Descripción de los principales sistemas de detección de radiación

ionización gaseosa). Esta ubicado en la parte más externa del detector interno y contiene 351000 recipientes cilíndricos de 4 mm de diámetro y 144cm de longitud, llenos de gas (70 %Xe, 27 %CO₂, 3 %O₂) que se ioniza cuando interactúa con una partícula cargada.

- **Calorímetros:** están situados en el exterior del solenoide magnético (el cual alcanza una intensidad máxima de hasta 2T, ver Fig. B.7), rodeando al detector interno. El objetivo de los calorímetros es medir la energía depositada por cada partícula mediante los procesos de absorción. Dispone de dos sistemas básicos: un calorímetro electromagnético interno y un calorímetro hadrónico externo; ambos son de muestreo, es decir, absorben la energía mediante metales densos y miden la forma de la cascada de la partícula resultante, de donde se infiere la energía de la partícula inicial.

El calorímetro electromagnético absorbe la energía de las partículas que interaccionan electromagnéticamente (los materiales absorbentes son plomo y acero inoxidable, refrigerados con argón líquido). El calorímetro hadrónico absorbe la energía de las partículas que atraviesan el calorímetro electromagnético, pero que interaccionan mediante la fuerza fuerte (el absorbente es acero con dispositivos centelladores).

- **Espectrómetro muónico:** es un sistema de seguimiento de 12000 m² que se extiende desde los calorímetros hasta la parte más externa del detector. Su gran tamaño es necesario para medir con precisión el momento lineal de los muones, pues estos son capaces de atravesar las partes más internas del detector.

En la Fig. B.7c se muestra un ejemplo de la detección de diversas partículas en el detector ATLAS. De acuerdo con la interacción a las que las partículas respondan, cada una de ellas deposita su energía en un sistema de detección particular; así por ejemplo, los fotones y electrones junto con otras partículas que respondan a interacciones electromagnéticas depositarán su energía en el calorímetro electromagnético; partículas "pesadas", compuestas (ver Fig. 2.1), como los neutrones y los protones la depositaran en el calorímetro hadrónico, y los muones en el espectrómetro muónico. Todas las partículas pasan por el detector de píxel y el TRT, mediante los cuales se realiza el proceso de Tracking (seguimiento y medición de la dirección y magnitud de los momentos lineales).

Descripción de los principales sistemas de detección de radiación

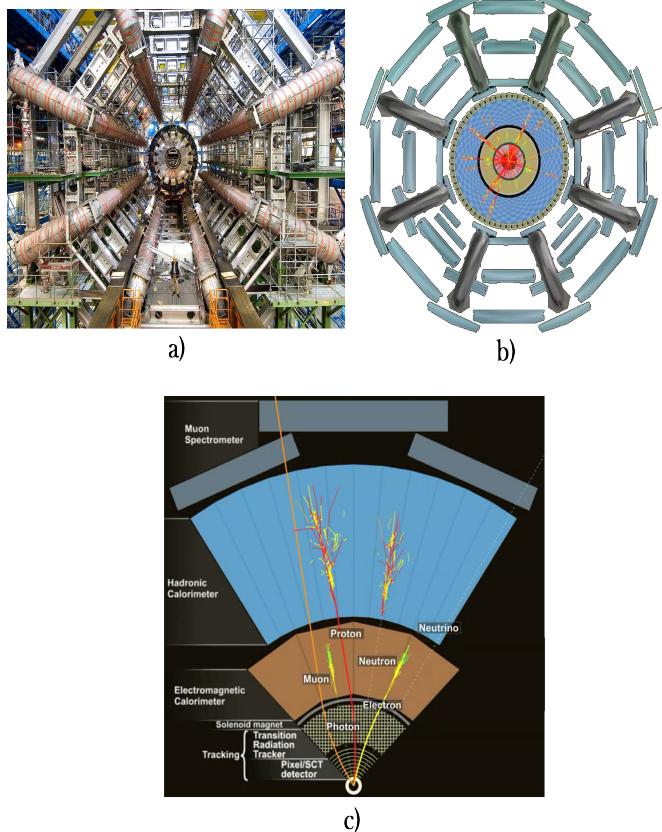


Figura B.7: a) Fotografía del electroimán y del detector interno de ATLAS; b) sección transversal de ATLAS mediante simulación; c) deposición de energía de algunas partículas en los sistemas de detección de ATLAS (Cern, 2011a).

Apéndice C

Licencia de Geant4

Se presenta el extracto textual, de la página web oficial (<http://geant4.cern.ch/license/LICENSE.html>), que contiene la licencia de GEANT4 como sigue:

The Geant4 Software License
Established 30 June 2006 for Geant4 release 8.1, subsequent patches and releases.
Previous releases are covered by the disclaimer included in the release.

Version 1.0, 28 June 2006
Copyright (c) Copyright Holders of the Geant4 Collaboration, 1994-2006.

This software includes voluntary contributions made to Geant4. See <http://cern.ch/geant4> for more information on Geant4.

Installation, use, reproduction, display, modification and redistribution of this software, with or without modification, in source and binary forms, are permitted on a non-exclusive basis. Any exercise of rights by you under this license is subject to the following conditions:

1. Redistributions of this software, in whole or in part, with or without modification, must reproduce the above copyright notice and these license conditions in this software, the user documentation and any other materials provided with the redistributed software.
2. The user documentation, if any, included with a redistribution, must include the following notice:
"This product includes software developed by Members of the Geant4 Collaboration (<http://cern.ch/geant4>)."

If that is where third-party acknowledgments normally appear, this acknowledgment

must be reproduced in the modified version of this software itself.

3. The names "Geant4" and "The Geant4 toolkit" may not be used to endorse or promote software, or products derived therefrom, except with prior written permission by license@geant4.org. If this software is redistributed in modified form, the name and reference of the modified version must be clearly distinguishable from that of this software.
4. You are under no obligation to provide anyone with any modifications of this software that you may develop, including but not limited to bug fixes, patches, upgrades or other enhancements or derivatives of the features, functionality or performance of this software. However, if you publish or distribute your modifications without contemporaneously requiring users to enter into a separate written license agreement, then you are deemed to have granted all Members and all Copyright Holders of the Geant4 Collaboration a license to your modifications, including modifications protected by any patent owned by you, under the conditions of this license.
5. You may not include this software in whole or in part in any patent or patent application in respect of any modification of this software developed by you.

6. DISCLAIMER

THIS SOFTWARE IS PROVIDED BY THE MEMBERS AND COPYRIGHT HOLDERS OF THE GEANT4 COLLABORATION AND CONTRIBUTORS AS IS, AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, OF SATISFACTORY QUALITY, AND FITNESS FOR A PARTICULAR PURPOSE OR USE ARE DISCLAIMED. THE MEMBERS OF THE GEANT4 COLLABORATION AND CONTRIBUTORS MAKE NO REPRESENTATION THAT THE SOFTWARE AND MODIFICATIONS THEREOF, WILL NOT INFRINGE ANY PATENT, COPYRIGHT, TRADE SECRET OR OTHER PROPRIETARY RIGHT.

7. LIMITATION OF LIABILITY

THE MEMBERS AND COPYRIGHT HOLDERS OF THE GEANT4 COLLABORATION AND CONTRIBUTORS SHALL HAVE NO LIABILITY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY, OR PUNITIVE DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, LOSS OF USE, DATA OR PROFITS, OR BUSINESS INTERRUPTION, HOWEVER CAUSED AND ON ANY THEORY OF CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE, ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

- 8.** This license shall terminate with immediate effect and without notice if you fail to comply with any of the terms of this license, or if you institute litigation against any Member or Copyright Holder of the Geant4 Collaboration with regard to this software.

C.1. Copyright Holders of the Geant4 Collaboration

Last revision: 30 June 2006

The collaboration has established the following list of institutions and individuals who hold copyright of parts of the Geant4 toolkit. In case there are attributions missing, the collaboration invites potential contributors to inform us of their contribution and who holds copyright over it, by writing to John.Apostolakis@cern.ch.

C.1.1. Colaboradores Institucionales

Bath University, Bath, UK; Budker Institute Nuclear Physics, Novosibirsk, Russia; Budapest Technical University, Budapest, Hungary; California Institute of Technology, Pasadena, USA; CERN, European Organization for Nuclear Research, Geneva, Switzerland; CIEMAT, Madrid, Spain; CNRS-IN2P3, Institut National de Physique Nucléaire, France; ESA, European Space Agency; ETH, Zurich, Switzerland; Fermi National Accelerator Laboratory, Batavia, USA; Helsinki Institute of Physics, Helsinki, Finland; IHEP, Protvino, Russia; Imperial College, London, UK; Institut für experimentelle Kernphysik, Karlsruhe University, Karlsruhe, Germany; Instituto de Física de Cantabria, Santander, Spain; INFN, Istituto Nazionale di Fisica Nucleare, Italy; IST National Institute for Cancer Research, Italy; Jefferson Laboratory, USA; JINR, Dubna, Russia; J. W. Goethe-Universität, Frankfurt am Main, Germany; Karolinska Institutet, Stockholm, Sweden; KFKI Research Institute for Particle and Nuclear Physics, Budapest, Hungary; Laboratório de Instrumentação e Física Experimental de Partículas (LIP), Lisbon, Portugal; Lebedev Institute, Moscow, Russia; Manchester University, Manchester, UK; MIT, Massachusetts Institute of Technology, Cambridge, USA; Moscow Engineering Physics Institute (State University), Moscow, Russia; Northeastern University, Boston, USA; Pittsburgh University, Pittsburgh, USA; Rutherford Appleton Laboratory, UK; Southampton University, Southampton, UK; Stanford University (for SLAC, Stanford Linear Accelerator Center), Stanford, USA; Tampere University, Tampere, Finland; TRIUMF, Vancouver, Canada; University of British Columbia, Vancouver, Canada; University of California, Santa-Cruz, USA; University of Cordoba, Cordoba, Spain; University of Maryland, USA.

C.1.2. Colaboradores Individuales

Akinori Kimura; Eri Suzuki; Glenn Horton-Smith; Harm Fesefeldt; Hisaya Kurashige; Hajime Yoshida; Kotoyo Hoshina; Katsuya Amako; Koichi Murakami; Makoto Takahata; Makoto Asai; Masayasu Nagamatu; Marcus H. Mendelhall; Kaz Minamimoto; Robert A. Weller; Shunji Sei; Satoshi Tanaka; Yasuhide Sawada; Toshiaki Kodama; Tetsuya Yamada; Tsukasa Aso; Takashi Sasaki; Youhei Morita.

Apéndice D

Comandos Básicos

En esta sección se realiza una descripción básica de algunos comandos a manejar en un terminal de GNU/Linux. Como regla general, se puede decir que: “*Todo lo que se puede hacer en modo gráfico, se puede hacer también en modo texto, a base de comandos. Pero no todo lo que se puede hacer en modo texto, se puede hacer en modo gráfico*”. Cabe anotar que hasta el momento, ningún programa gráfico supera en versatilidad y potencia a una consola de texto donde ejecutar comandos directamente. Todos los comandos citados en este apartado, pueden usarse con multitud de opciones, para conseguir información detallada de cada comando puede utilizar `man` comando.

D.1. Comandos internos (bash)

- `help`: muestra la ayuda de bash, la lista de comandos disponibles y su sintaxis.
- `cd`: cambia de directorio.
- `pwd`: muestra la ruta actual en la cual se está ubicado.
- `alias`: permite la creación de comandos propios.
- `history`: muestra el listado de todos los comandos ejecutados, también puede acceder a éste listado con las teclas “arriba” y “abajo”.
- `!!`: ejecuta el último comando.
- `!n`: ejecuta el comando número “n” del historial.
- `!texto`: ejecuta el último comando que empiece por el texto que escrito.
- `exit` y `logout`: cierran la sesión y la consola vuelve a su estado inicial.

D.2. Comandos externos GNU : manipulación de ficheros

- **ls**: lista el contenido del directorio actual.
- **cp**: copia un fichero.
- **mv**: mueve un fichero a otro directorio o le asigna un nuevo nombre.
- **rm**: borra un fichero.
- **ln**: crea un enlace simbólico o físico de un fichero.
- **mkdir** y **rmdir**: crea y borra un directorio, respectivamente.
- **chmod** y **chown**: permiten cambiar los permisos de un fichero y el usuario/grupo, respectivamente.
- **cat**: muestra el contenido del fichero especificado.
- **touch**: crea un fichero nuevo.
- **more** y **less**: paginan el contenido del fichero especificado.
- **man**: es el manual del sistema, muestra la ayuda de un comando específico.
- **find**: realiza una búsqueda de un archivo determinado.
- **jobs** y **fg**: permiten ver y reactivar comandos que están siendo ejecutados en background.

D.3. Comandos externos GNU : control de procesos

- **top**: muestra dinámicamente los procesos activos del sistema, además de información relativa a ellos.
- **ps**: muestra los procesos activos del sistema e información relativa a ellos, pero no dinámicamente.
- **kill**: permite detener un proceso, es útil cuando este se ha “colgado”.

D.4. Editores de texto en modo consola

- **nano**: es un editor de texto ligero que permite editar en modo similar al notepad.
- **vi/vim**: es un editor de texto rápido de usar, basado en comandos de teclado y edición en distintos modos (inserción, reemplazo, visual, comando y edición rápida).
- **emacs**: es un editor de texto visual que cuenta con gran cantidad de opciones.

D.5. Otros comandos GNU

- **grep**: permite aplicar expresiones regulares a una entrada de texto y devuelve las líneas que coinciden.
- **tar, bzip2 y gzip**: son comandos de archivación de ficheros, el primero almacena ficheros y los otros dos son compresores de datos.
- **whatis**: muestra un resumen descriptivo de un comando.
- **whereis**: muestra la ubicación física del fichero correspondiente a un comando.
- **apropos**: permite buscar comandos por su funcionalidad en vez de por su nombre.

Comandos Básicos

Apéndice E

Instalación de ROOT 6.14.06

E.1. Instalación

ROOT es una infraestructura de software en lenguaje C++ (R. Cern, 2011), creado por CERN en 1994, para desarrollar aplicaciones de análisis de datos a gran escala. Para instalar ROOT¹ en su versión 6.14 se debe descargar el archivo binario de:

https://root.cern.ch/download/root_v6.14.06.Linux-ubuntu16-x86_64-gcc5.4.tar.gz

Para las características de instalación en sistema operativo Ubuntu16.04LTS (64bits), desde una terminal (Ctrl-Alt T) se deben ejecutar las siguientes líneas de comandos:

```
1 fisinfor@taozen:~$ cd
2 fisinfor@taozen:~$ pwd
3 /home/fisinfor
4
5 fisinfor@taozen:~$ mkdir root
6 fisinfor@taozen:~$ cd root
7 fisinfor@taozen:~/root$ cp ~/Downloads/root_v6.14.06.Linux-ubuntu16-x86_64-gcc5.4.tar.gz .
8 fisinfor@taozen:~/root$ tar zxfv root_v6.14.06.Linux-ubuntu16-x86_64-gcc5.4.tar.gz
9 root/
10 root/fonts/
11 root/fonts/STIXSiz1Sym.otf
12 root/fonts/FreeSerifItalic.otf
13 root/fonts/verdanai.ttf
14 root/fonts/arialbi.ttf
15 ...
16 ...
17 root/icons/refresh2.xpm
18 root/icons/interrupt.xpm
```

¹Es importante no confundir **ROOT** (Un sistema de análisis de datos) con **root** (superusuario o usuario con privilegios).

```
19 root/icons/sm_export.xpm
20 root/icons/eve_scene.xpm
21 root/icons/query_connected.xpm
22 root/icons/branch_folder_s.xpm
23
24 fisinfor@taozen:~/root$ ls
25 root root_v6.14.06.Linux-ubuntu16-x86_64-gcc5.4.tar.gz
26
27 fisinfor@taozen:~/root$ cd root/
28 fisinfor@taozen:~/root/root$ ls
29 aclocal cmake emacs fonts icons lib macros README tmva
30 bin config etc geom include LICENSE man test tutorials
31
32 fisinfor@taozen:~/root/root$
```

E.2. Ejecución

La ejecución de ROOT requiere un paso mas en la instalación y es el de configurar el archivo .bashrc, agregando en el estas linea de comando:

```
1 #root configuracion variables de entorno
2
3 export ROOTSYS=/home/fisinfor/root/root
4 export PATH=$ROOTSYS/bin:$PATH
5 export LD_LIBRARY_PATH=$ROOTSYS/lib:$LD_LIBRARY_PATH
```

Desde lo anterior, se debe abrir una nueva terminal, que automaticamente agrega las lineaas a la configuración anteriores, y ejecutar la linea de comando:

```
1 fisinfor@taozen:~$ root
```

que permite visualizar la pantalla de bienvenida ilustradas en las Figs. E.1 y E.2 .

Instalación de ROOT 6.14.06

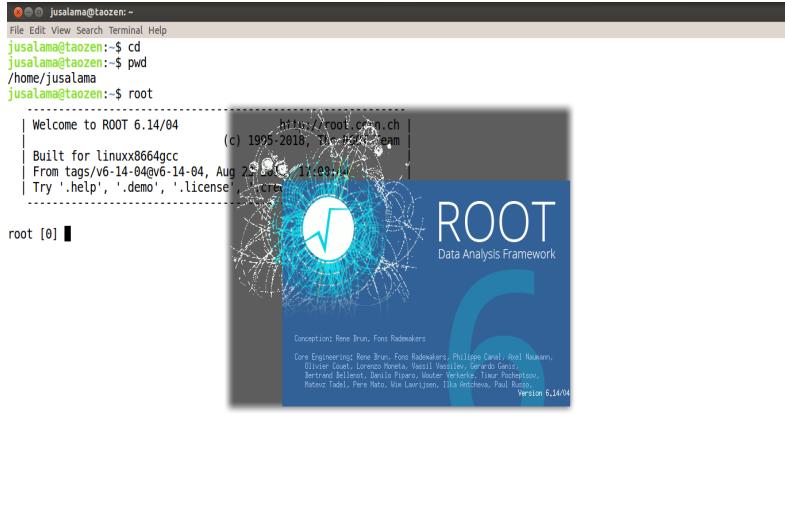


Figura E.1: Pantalla de inicio de ROOT.

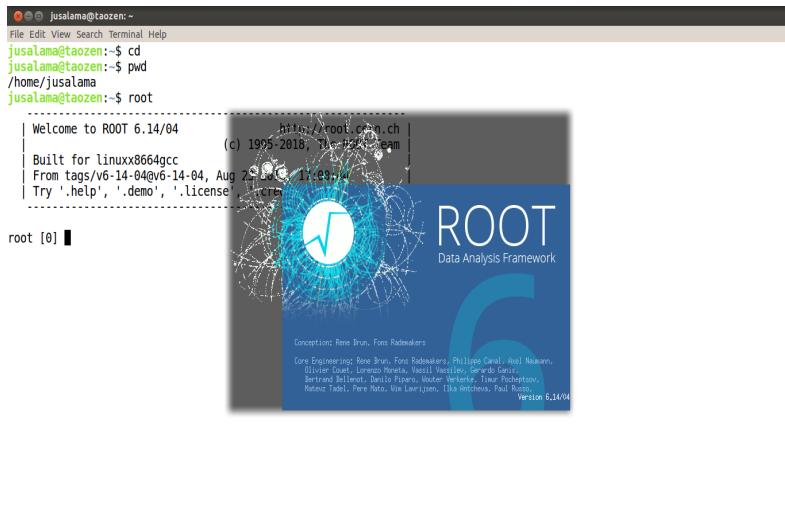


Figura E.2: Pantalla de inicio de ROOT.

Instalación de ROOT 6.14.06

Apéndice F

Instalación de *GEANT4*

En este apartado se muestra un procedimiento detallado para instalar adecuadamente el conjunto de paquetes de *GEANT4* bajo el sistema operativo GNU/Linux Ubuntu16.04LTS (64bits).

Estas líneas de comando se han adaptado de las instrucciones proporcionadas por el equipo de SLAC específicamente para un usuario que está instalando *GEANT4* en una distribución “ limpia ” de Ubuntu. Este documento está escrito para el usuario principiante sobre el uso y manejo de *GEANT4* , con algunos conocimientos básicos de programación orientada a objetos bajo lenguaje de C++ y manejo básico del intérprete de comandos, o terminal de Linux.

El hardware utilizado para la instalación corresponde a un equipo de estas características: 1Tb HD, 4Gb de RAM Física, Procesador Intel Core i5-7200U 2.50GHz x 4.

F.1. Requisitos de Software

GEANT4 debe suplirse de paquetes (software) adicionales para su correcto funcionamiento, siendo **gcc**, **g++** y **CLHEP** los “ pilares ” que dan total funcionalidad a éste conjunto de herramientas.

gcc, el compilador recomendado (Perl, 2011) para *GEANT4* en sistemas operativos GNU/Linux, es un compilador integrado del proyecto GNU para C, C++, Objective C y Fortran. **gcc** es capaz de recibir un programa fuente en cualquiera de estos lenguajes y generar un fichero binario, ejecutable, en el lenguaje de la máquina donde ha de ejecutarse. La sigla **gcc** significa “ GNU Compiler Collection ”, originalmente, su significado

era “GNU C Compiler”, en tanto **g++** se refiere a una compilación en C++.

CLHEP (Class Library for High Energy Physics) es un conjunto de bibliotecas en C++ que han proporcionado durante mucho tiempo una gran funcionalidad para la comunidad de física de partículas, ofreciendo clases útiles para programación numérica, cálculo vectorial, geometría, pseudo-generación de números aleatorios y álgebra lineal. Los desarrolladores de *GEANT4* han optado por utilizar las librerías de CLHEP en lugar de “re-inventarlas”; sin embargo, se debe tener presente que son softwares diferentes, esto es, el grupo de desarrollo y de colaboradores de *GEANT4* no es responsable del mantenimiento, la distribución o la documentación de CLHEP y viceversa.

Para comprobar que **gcc** se encuentra instalado en el sistema operativo se escribe en una ventana de terminal **gcc -v**, si la respuesta de la línea de comando es afirmativa no es necesario instalar este paquete. En particular *GEANT4* requiere de **gcc-4.9.3**, **gcc-5.3.0** o **gcc-6.3.0** para distribuciones Linux.

La preparación previa a la instalación de *GEANT4* requiere que la distribución GNU/Linux a utilizar cuente con un conjunto de paquetes “base”, para lo cual es aconsejable escribir una terminal:

```
sudo apt-get install build-essential.
```

Además el usuario debe asegurar que los paquetes listados a continuación se encuentran instalados en el ordenador:

- libxerces-c-dev
- qt4-dev-tools
- freeglut3-dev
- libmotif-dev
- tk-dev
- cmake
- libxpm-dev
- libxmu-dev
- libxi-dev

La ejecución de estas líneas de comando:

```
1 sudo apt-get install libxerces-c-dev qt4-dev-tools freeglut3-dev libmotif-dev  
2 tk-dev cmake libxpm-dev libxmu-dev libxi-dev
```

aseguró los paquetes necesarios para la instalación de *GEANT4*.

Nota: La familia de sistemas operativos Ubuntu tiene por defecto el comando, *sudo* (“SUperuser DO”) que proporciona permisos de “superusuario” (usuario *root*).

F.1.1. Instalación de CLHEP

GEANT4 requiere de la versión adecuada de CLHEP El caso particular de *GEANT4* 10.4, la versión recomendada por el grupo de desarrolladores (Perl, 2011) es CLHEP release 2.4.0.0 (o la mas reciente). La utilización de cualquier otra versión de CLHEP mas antigua puede hacer que *GEANT4* funcione; sin embargo, se pueden obtener resultados incorrectos. Se escoge entonces a la fecha la versión CLHEP 2.4.2.0.

Inicialmente se realiza la descarga de CLHEP desde la página web oficial <http://proj-clhep.web.cern.ch/proj-clhep/DISTRIBUTION/clhep.html>, en la sección “Source code and precompiled binary distribution kits” (parte inferior de la página), seleccione “clhep-2.4.1.0.tgz” (ver Figura F.1) que corresponde a: proj-clhep.web.cern.ch/proj-clhep/DISTRIBUTION/tarFiles/clhep-2.4.1.0.tgz(aproximadamente 1.5 MB, ver Fig. F.1).

Este archivo se debe descomprimir en una carpeta (CLHEP) donde se realizará la instalación de CLHEP-2.4.1.0, haciendo uso de las líneas de comando:

```
1 fisinfor@taozen:~$ cd  
2 fisinfor@taozen:~$ pwd  
3 /home/fisinfor  
4 fisinfor@taozen:~$ mkdir CLHEP  
5 fisinfor@taozen:~$ cp ~/Downloads/clhep-  
6 fisinfor@taozen:~$ cd CLHEP/  
7 fisinfor@taozen:~/CLHEP$ cp ~/Downloads/clhep-2.4.1.0.tgz .  
8 fisinfor@taozen:~/CLHEP$ tar zxfv clhep-2.4.1.0.tgz  
9 ...  
10 ...  
11 2.4.1.0/CLHEP/makeBinaryTar.in  
12 2.4.1.0/CLHEP/makeSourceDist.in  
13 2.4.1.0/CLHEP/setup.cygwin-VC71
```

Al interior de esta carpeta se crea otra que se llame 2.4.1.0-build, que dentro de esta

Release	Source	ChangeLog	Notes
2.4.1.0	clhep-2.4.1.0.tgz	ChangeLog for 2.4.1.0	various improvements for gcc 8.1.0 add operator double(), which returns flat(), to Random engines
2.4.0.4	clhep-2.4.0.4.tgz	ChangeLog for 2.4.0.4	allow thread local and atomic for Windows VC++
2.4.0.2	clhep-2.4.0.2.tgz	ChangeLog for 2.4.0.2	bug fix for CLHEP-147
2.4.0.1	clhep-2.4.0.1.tgz	ChangeLog for 2.4.0.1	bug fix for CLHEP-146
2.4.0.0	clhep-2.4.0.0.tgz	ChangeLog for 2.4.0.0	Update MixMax and make it the default for HepRandom
2.3.4.6	clhep-2.3.4.6.tgz	ChangeLog for 2.3.4.6	bug fix for CLHEP-147
2.3.4.5	clhep-2.3.4.5.tgz	ChangeLog for 2.3.4.5	Add support for -std=c++11 Resolve CLHEP-142
2.3.4.4	clhep-2.3.4.4.tgz	ChangeLog for 2.3.4.4	Fix bugs CLHEP-137 and CLHEP-141
2.3.4.3	clhep-2.3.4.3.tgz	ChangeLog for 2.3.4.3	Bug fix for thread local with the Intel compiler
2.3.4.2	clhep-2.3.4.2.tgz	ChangeLog for 2.3.4.2	Bug fix for thread local and AppleClang Implemented CLHEP-138.

Figura F.1: Captura de pantalla página oficial de descargas de CLHEP: <http://proj-clhep.web.cern.ch/proj-clhep/clhep23.html>

Última se ejecuta la línea de comando a CMAKE¹, cuyas opciones son:

```
cmake -DCMAKE_INSTALL_PREFIX=<ruta de archivo de instalación> <ruta carpeta contenedora del software>
```

Para tal fin se ejecutan las líneas de comando:

```
14 fisinfor@taozen:~/CLHEP$ mkdir 2.4.1.0-build
15 fisinfor@taozen:~/CLHEP$ ls
16 2.4.1.0 2.4.1.0-build clhep-2.4.1.0.tgz
17 fisinfor@taozen:~/CLHEP$ cd 2.4.1.0-build/
18 fisinfor@taozen:~/CLHEP/2.4.1.0-build$ 
19 cmake -DCMAKE_INSTALL_PREFIX=/home/fisinfor/CLHEP/2.4.1.0-install/ /home/fisinfor/CLHEP/2.4.1.0/CLHEP/
20 ...
21 ...
22 -- clhep CPACK_SYSTEM_NAME:      x86_64-ubuntu16-gcc54-opt
23 -- Configuring done
24 -- Generating done
25 -- Build files have been written to: /home/fisinfor/CLHEP/2.4.1.0-build
```

¹Es un conjunto de herramientas que sirven para compilar y probar un software. Para esta aplicación se requiere la versión 3.3. Mayor referencia se puede encontrar en <https://cmake.org/>

Una vez cmake preparé al sistema se ejecuta la instrucción `make -j4` donde "4" corresponde al número de cores del procesador. El número de núcleos que posee un procesador del equipo de computo con el que se realiza este ejercicio práctico corresponde a un Intel Core i5, el cual posee 4 núcleos físicos. Así, se espera tener esta respuesta en la ejecución:

```
26 fisinfor@taozen:~/CLHEP/2.4.1.0-build$ make -j4
27 ...
28 ...
29 [ 15%] Building CXX object CMakeFiles/CLHEPS.dir/Vector/src/ThreeVector.cc.o
30 [ 15%] Building CXX object CMakeFiles/CLHEP.dir/Vector/src/ThreeVector.cc.o
31 ...
32 ...
33 NOTA: ESTE PROCESO SE DEMORA DEL ORDEN DE 4 MINUTOS
34
35 [ 99%] Building CXX object RandomObjects/src/CMakeFiles/RandomObjects.dir/RandomVector.cc.o
36 [ 99%] Built target RandomObjectsS
37 Scanning dependencies of target testVectorDists
38 [ 99%] Building CXX object RandomObjects/test/CMakeFiles/testVectorDists.dir/testVectorDists.cc.o
39 [100%] Linking CXX executable testInstanceRestore
40 [100%] Built target testInstanceRestore
41 [100%] Linking CXX shared library ../../lib/libCLHEP-RandomObjects-2.4.1.0.so
42 [100%] Built target RandomObjects
43 [100%] Linking CXX executable testVectorDists
44 [100%] Built target testVectorDists
```

El proceso anterior demora del orden de unos 4 minutos (con las especificaciones del hardware utilizado), el cual es validado por la instrucción `make test`, que permite hace una prueba de la compilación de las librerías de CLHEP2.4.1.0:

```
45 fisinfor@taozen:~/CLHEP/2.4.1.0-build$ make test
46 Running tests...
47 Test project /home/fisinfor/CLHEP/2.4.1.0-build
48     Start 1: testUnits
49 1/44 Test #1: testUnits ..... Passed  0.01 sec
50 ...
51 ...
52 23/44 Test #23: testBug58950 ..... Passed  0.00 sec
53     Start 24: testBug73093
54 24/44 Test #24: testBug73093 ..... Passed  0.01 sec
55 ...
56 ...
57 44/44 Test #44: testzmex.sh ..... Passed  1.99 sec
58
59 100% tests passed, 0 tests failed out of 44
60
61 Total Test time (real) = 32.00 sec
```

Para finalizar, se requiere hacer la instalación de las librerías con la utilidad `make install`, el cual genera un nuevo directorio llamado `2.4.1.0-install`, que aloja la instalación de CLHEP2.4.1.0:

```
62 fisinfor@taozen:~/CLHEP/2.4.1.0-build$ make install
63 [ 22%] Built target CLHEPS
64 [ 44%] Built target CLHEP
65 [ 45%] Built target testUnits
66 [ 45%] Built target testSharedPtrConvertible
67 ...
68 ...
69 -- Installing: /home/fisinfor/CLHEP/2.4.1.0-install/include/CLHEP/Exceptions/link_Exceptions.mk
70 -- Installing: /home/fisinfor/CLHEP/2.4.1.0-install/lib/libCLHEP-Exceptions-2.4.1.0.so
71 -- Set runtime path of "/home/fisinfor/CLHEP/2.4.1.0-install/lib/libCLHEP-Exceptions-2.4.1.0.so" to ""
72 -- Installing: /home/fisinfor/CLHEP/2.4.1.0-install/lib/libCLHEP-Exceptions-2.4.1.0.a
```

F.2. Instalación de *GEANT4* 10.4

La documentación oficial de instalación de *GEANT4* se encuentra, a la fecha, publicada en: <http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/InstallationGuide/html/>, que fue donde se realizó la mayor consulta para el trabajo práctico de instalación que aquí se presenta.

Cabe advertir que la instalación de *GEANT4* ha cambiado ostensiblemente de las versiones anteriores *GEANT4* 9.4. Se utiliza, al igual que para compilar las librerías de CLHEP, la utilidad *cmake*.

El inicio de la instalación de *GEANT4* 10.4 comienza con la descarga del archivo fuente (source code) de la página oficial de *GEANT4* : <https://geant4.web.cern.ch/support/download>, o de un enlace directo de descarga: <http://geant4-data.web.cern.ch/geant4-data/releases/geant4.10.04.p02.tar.gz> (ver Fig. F.2).

Generalmente, la descarga de este archivo comprimido queda en Descargas (o Download), el cual requiere ser colocado en una carpeta donde se realizará todo el trabajo de instalación (que se llamará *geant4*), como se ilustra en las siguientes líneas de comando:

```
1 fisinfor@taozen:~$ cd
2 fisinfor@taozen:~$ pwd
3 /home/fisinfor
4 fisinfor@taozen:~$ mkdir geant4
5 fisinfor@taozen:~$ cd geant4/
6 fisinfor@taozen:~/geant4$ cp ~/Downloads/geant4.10.04.p02.tar.gz .
7 fisinfor@taozen:~/geant4$ tar zxfv geant4.10.04.p02.tar.gz
8 ...
9 ...
10 geant4.10.04.p02/environments/g4py/site-modules/materials/NISTmaterials/pyNISTmaterials.cc
11 geant4.10.04.p02/environments/g4py/site-modules/materials/NISTmaterials/NISTmaterials.cc
12 geant4.10.04.p02/environments/g4py/site-modules/python/
```

Instalación de GEANT4

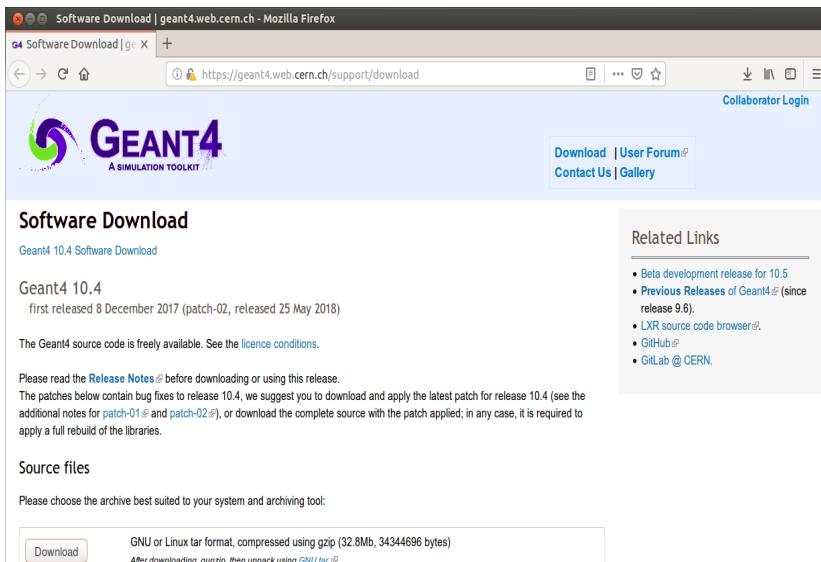


Figura F.2: Captura de pantalla página oficial de descargas de GEANT4: <https://geant4.web.cern.ch/support/download>

```
13 | geant4.10.04.p02/environments/g4py/site-modules/python/CMakeLists.txt
14 | geant4.10.04.p02/environments/g4py/site-modules/python/__init__.py
15 | geant4.10.04.p02/environments/g4py/README.md
```

Al igual que en la instalación de CLHEP2.4.1.0 se debe crear una carpeta para construir la base necesaria de instalación, la cual se llamó `geant4.10.04.p02-build` y se agrego a la carpeta de instalación `geant4` así:

```
16 | fisinfor@taozn:~/geant4$ mkdir geant4.10.04.p02-build
17 | fisinfor@taozn:~/geant4$ ls
18 | geant4.10.04.p02  geant4.10.04.p02-build  geant4.10.04.p02.tar.gz
```

Fíjese que se hace una lista del directorio donde aparecen dos carpetas y el archivo `.tar` de `GEANT4`.

Seguidamente se debe realizar la configuración para la compilación y enlace del software a través de la utilidad `cmake` dentro de la carpeta `geant4.10.04.p02-build`. En este punto es importante resaltar que la linea de comando correspondiente a `cmake` es realmente larga, con lo que se advierte que es una sola línea se comando con las siguientes opciones:

```
cmake -DCMAKE_INSTALL_PREFIX=/home/fisinfor/geant4/geant4.10.04.p02-install/  
-DGEANT4_USE_QT=ON  
-DGEANT4_USE_OPENGL_X11=ON  
-DGEANT4_INSTALL_DATA=ON  
-DGEANT4_USE_RAYTRACER_X11=ON  
-DGEANT4_USE_GDML=ON  
-DGEANT4_USE_SYSTEM_CLHEP=ON  
-DCLHEP_INCLUDE_DIR=/home/fisinfor/CLHEP/2.4.1.0-install/include/  
-DCLHEP_LIBRARY=/home/fisinfor/CLHEP/2.4.1.0-install/lib/libCLHEP.so  
-DGEANT4_INSTALL_EXAMPLES=ON  
-DCMAKE_PREFIX_PATH=/home/fisinfor/CLHEP/2.4.1.0-build/  
/home/fisinfor/geant4/geant4.10.04.p02/
```

De lo antedicho, su ejecución queda así:

```
19 fisinfor@taozen:~/geant4$ cd geant4.10.04.p02-build/  
20 fisinfor@taozen:~/geant4/geant4.10.04.p02-build$  
21 cmake -DCMAKE_INSTALL_PREFIX=/home/fisinfor/geant4/geant4.10.04.p02-install/  
22 -DGEANT4_USE_QT=ON -DGEANT4_USE_OPENGL_X11=ON -DGEANT4_INSTALL_DATA=ON  
23 -DGEANT4_USE_RAYTRACER_X11=ON -DGEANT4_USE_GDML=ON -DGEANT4_USE_SYSTEM_CLHEP=ON  
24 -DCLHEP_INCLUDE_DIR=/home/fisinfor/CLHEP/2.4.1.0-install/include/  
25 -DCLHEP_LIBRARY=/home/fisinfor/CLHEP/2.4.1.0-install/lib/libCLHEP.so -DGEANT4_INSTALL_EXAMPLES=ON  
26 -DCMAKE_PREFIX_PATH=/home/fisinfor/CLHEP/2.4.1.0-build/ /home/fisinfor/geant4/geant4.10.04.p02/  
27 ...  
28 NOTA: RECUERDE QUE LA LINEA DE COMANDO DE CMAKE ES UNA SOLA.  
29 ESTA LINEA FUE EDITADA PARA SER VISUALIZADA EN EL TEXTO.  
30 ...  
31 -- Configuring download of missing dataset G4NDL (4.5)  
32 -- Configuring download of missing dataset G4EMLOW (7.3)  
33 -- Configuring download of missing dataset PhotonEvaporation (5.2)  
34 -- Configuring download of missing dataset RadioactiveDecay (5.2)  
35 -- Configuring download of missing dataset G4NEUTRONXS (1.4)  
36 -- Configuring download of missing dataset G4PII (1.3)  
37 -- Configuring download of missing dataset RealSurface (2.1.1)  
38 -- Configuring download of missing dataset G4SAIDDATA (1.1)  
39 -- Configuring download of missing dataset G4ABLA (3.1)  
40 -- Configuring download of missing dataset G4ENSDFSTATE (2.2)  
41 -- The following Geant4 features are enabled:  
42 GEANT4_BUILD_CXXSTD: Compiling against C++ Standard '11'  
43 GEANT4_USE_SYSTEM_CLHEP: Using system CLHEP library (singular)  
44 GEANT4_USE_SYSTEM_EXPAT: Using system EXPAT library  
45 GEANT4_USE_GDML: Building Geant4 with GDML support  
46 GEANT4_USE_QT: Build Geant4 with Qt support  
47 GEANT4_USE_RAYTRACER_X11: Build RayTracer driver with X11 support  
48 GEANT4_USE_OPENGL_X11: Build Geant4 OpenGL driver with X11 support  
49  
50 -- Configuring done
```

con lo que se ejecuta la utilidad make -j4 (acorde al número de núcleos del procesador), para obtener la salida:

```
51 -- Generating done
```

Instalación de GEANT4

```
52 -- Build files have been written to: /home/fisinfor/geant4/geant4.10.04.p02-build
53 fisinfor@taozen:~/geant4/geant4.10.04.p02-build$ make -j4
54 ...
55 ...
56 HETCEmissionFactory.cc.o
57 [ 73%] Building CXX object source/processes/CMakeFiles/G4processes.dir/hadronic/models/pre...
58 [ 74%] Building CXX object source/processes/CMakeFiles/G4processes.dir/hadronic/models/pre...
59 [ 74%] Building CXX object source/processes/CMakeFiles/G4processes.dir/hadronic/models/pre...
60 [ 74%] Building CXX object source/processes/CMakeFiles/G4processes.dir/hadronic/models/pre...
61 ...
62 ...
63 NOTA: ESTE PROCESO DEMORA AL REDEDOR DE 30 min (se contabilizo 28min 55 segundos)
64 ...
65 ...
66 t.cc.o
67 [ 98%] Building CXX object source/physics_lists/CMakeFiles/G4physicslists.dir/constructors/lim...
68 [ 98%] Building CXX object source/physics_lists/CMakeFiles/G4physicslists.dir/constructors/lim...
69 [ 98%] Building CXX object source/physics_lists/CMakeFiles/G4physicslists.dir/constructors/lim...
70 [ 98%] Building CXX object source/physics_lists/CMakeFiles/G4physicslists.dir/constructors/sto...
71 [ 98%] Building CXX object source/physics_lists/CMakeFiles/G4physicslists.dir/lists/src/G4Phys...
72 [ 98%] Building CXX object source/physics_lists/CMakeFiles/G4physicslists.dir/lists/src/G4Phys...
73 [ 98%] Building CXX object source/physics_lists/CMakeFiles/G4physicslists.dir/lists/src/QBBC.c...
74 [100%] Building CXX object source/physics_lists/CMakeFiles/G4physicslists.dir/lists/src/G4Phy...
75 [100%] Building CXX object source/physics_lists/CMakeFiles/G4physicslists.dir/lists/src/G4Phy...
76 [100%] Building CXX object source/physics_lists/CMakeFiles/G4physicslists.dir/util/src/G4Phys...
77 [100%] Building CXX object source/physics_lists/CMakeFiles/G4physicslists.dir/util/src/G4War...
```

Cabe anotar que este proceso demoró 28 minutos y 55 segundos (al rededor de media hora).

Para finalizar la instalación se ejecuta la utilidad `make install` la cual crea una nueva carpeta, `geant4 .10.04. p02 - install`, donde quedará la instalación el software de **GEANT4**:

```
78 [100%] Linking CXX shared library ../../BuildProducts/lib/libG4physicslists.so
79 [100%] Built target G4physicslists
80 fisinfor@taozen:~/geant4/geant4.10.04.p02-build$ make install
81 [ 0%] Built target G4ENSDFSTATE
82 [ 0%] Built target G4NDL
83 [ 0%] Built target PhotonEvaporation
84 [ 0%] Built target RadioactiveDecay
85 ...
86 ...
87 ESTE PROCESO SE DEMORA AL REDEDOR DE 3MIN
88 ...
89 ...
90 -- Installing: /home/fisinfor/geant4/geant4.10.04.p02-install/include/Geant4/G4OpenGLQtExpo...
91 -- Installing: /home/fisinfor/geant4/geant4.10.04.p02-install/include/Geant4/G4OpenGLQtMovi...
92 -- Installing: /home/fisinfor/geant4/geant4.10.04.p02-install/include/Geant4/G4OpenGLVboDra...
93 -- Installing: /home/fisinfor/geant4/geant4.10.04.p02-install/include/Geant4/G4OpenGLQtViewer.hh
94 -- Installing: /home/fisinfor/geant4/geant4.10.04.p02-install/include/Geant4/G4OpenGLStoredQt.hh
95 -- Installing: /home/fisinfor/geant4/geant4.10.04.p02-install/include/Geant4/G4OpenGLStored...
96 -- Installing: /home/fisinfor/geant4/geant4.10.04.p02-install/include/Geant4/G4OpenGLStored...
```

```
97 fisinfor@taozen:~/geant4/geant4.10.04.p02-build$  
98 fisinfor@taozen:~/geant4/geant4.10.04.p02-build$ cd ..  
99 fisinfor@taozen:~/geant4$ ls
```

y se modifica el archivo de configuración del interprete de comando en Linux, es decir, se modifica el archivo `.bashrc`, agregando esta línea:

```
source /home/fisinfor/geant4/geant4.10.04.p02-install/bin/geant4.sh  
source /home/fisinfor/geant4/geant4.10.04.p02-install/bin/geant4.sh
```

F.3. Cómo ejecutar un ejemplo de *GEANT4*?

GEANT4 tiene a disposición una página de soporte al usuario ² donde se puede consultar los detalles de los ejemplos suministrados por los desarrolladores de *GEANT4* (http://geant4-userdoc.web.cern.ch/geant4-userdoc/Doxygen/examples_doc/html/).

Para este ejercicio práctico utilizaremos el ejemplo ³ "básico" denominado B1, el cual se encuentra de la ruta:

```
/home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/examples/basic/B1
```

Para su ejecución, se debe hacer una carpeta de construcción del ejemplo donde se alojará la compilación y software adicional permite ejecutar la simulación correspondiente al ejemplo B1 ⁴, realizando una ejecución del comando `cmake` y haciendo uso de la utilidad `make -j4` para obtener un ejecutable, `exampleB1`, que permite la ejecución del ejemplo; esto es:

```
1 fisinfor@taozen:~$ pwd  
2 /home/fisinfor  
3 fisinfor@taozen:~$  
4 cd /home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/examples/basic/B1/  
5  
6 fisinfor@taozen:~/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/examples/basic/B1$  
7 mkdir B1-build  
8  
9 fisinfor@taozen:~/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/examples/basic/B1$  
10 cd B1-build/  
11  
12 fisinfor@taozen:~/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/examples/basic/B1/B1-build$  
13 cmake -DGeant4_DIR=/home/fisinfor/geant4/geant4.10.04.p02
```

²<https://geant4.web.cern.ch/support>

³Para la fecha hay tres tipos de ejemplos: Básicos, Extendidos y Avanzados.

⁴Este ejemplo muestra una aplicación donde se encuentra el valor de la disposición de energía en un usuario y los objetos asociados a él (también se calcula la dosis en un volumen asociado).

Instalación de GEANT4

```
14| /home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/examples/basic/B1/
15|
16| -- The C compiler identification is GNU 5.4.0
17| -- The CXX compiler identification is GNU 5.4.0
18| -- Check for working C compiler: /usr/bin/cc
19| -- Check for working C compiler: /usr/bin/cc -- works
20| -- Detecting C compiler ABI info
21| -- Detecting C compiler ABI info - done
22| -- Detecting C compile features
23| -- Detecting C compile features - done
24| -- Check for working CXX compiler: /usr/bin/c++
25| -- Check for working CXX compiler: /usr/bin/c++ -- works
26| -- Detecting CXX compiler ABI info
27| -- Detecting CXX compiler ABI info - done
28| -- Detecting CXX compile features
29| -- Detecting CXX compile features - done
30| -- Configuring done
31| -- Generating done
32| -- Build files have been written to:
33| /home/fisinfor/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/examples/basic/B1/B1-build
34|
35| fisinfor@taozen:~/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/examples/basic/B1/B1-build$ make -j4
36|
37|
38| Scanning dependencies of target exampleB1
39| [ 12%] Building CXX object CMakeFiles/exampleB1.dir/exampleB1.cc.o
40| [ 25%] Building CXX object CMakeFiles/exampleB1.dir/src/B1DetectorConstruction.cc.o
41| [ 37%] Building CXX object CMakeFiles/exampleB1.dir/src/B1ActionInitialization.cc.o
42| [ 50%] Building CXX object CMakeFiles/exampleB1.dir/src/B1SteppingAction.cc.o
43| [ 62%] Building CXX object CMakeFiles/exampleB1.dir/src/B1PrimaryGeneratorAction.cc.o
44| [ 87%] Building CXX object CMakeFiles/exampleB1.dir/src/B1RunAction.cc.o
45| [ 87%] Building CXX object CMakeFiles/exampleB1.dir/src/B1EventAction.cc.o
46| [100%] Linking CXX executable exampleB1
47| [100%] Built target exampleB1
48| fisinfor@taozen:~/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/examples/basic/B1/B1-build$ ls
49| fisinfor@taozen:~/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/examples/basic/B1/B1-build$ ls
50| CMakeCache.txt cmake_install.cmake exampleB1.in init_vis.mac run1.mac vis.mac
51| CMakeFiles exampleB1 exampleB1.out Makefile run2.mac
```

Para ejecutar la simulación se debe correr en la terminal, dentro de la carpeta de construcción `../B1_build`, el comando `./ exampleB1`:

```
53| fisinfor@taozen:~/geant4/geant4.10.04.p02-install/share/Geant4-10.4.2/examples/basic/B1/B1-build$ ./example
```

donde aparecerá una interfaz gráfica de usuario como se ilustra en la Fig. F.3, que ilustra la geometría de la simulación, un menú con una guía de comandos, y una caja ejecución bajo el nombre de "Session".

Como ejercicio, se escogió ejecutar un haz con 5 rayos de partículas incidiendo en la geometría, utilizando al opción `/run/beamOn 5`, como ilustran las Figs. F.4 F.5:

Se obtiene entonces la simulación de las trayectorias de los 5 rayos que inciden sobre la geometría de la simulación, como lo ilustran las Figs. F.6 y F.7.

Instalación de GEANT4

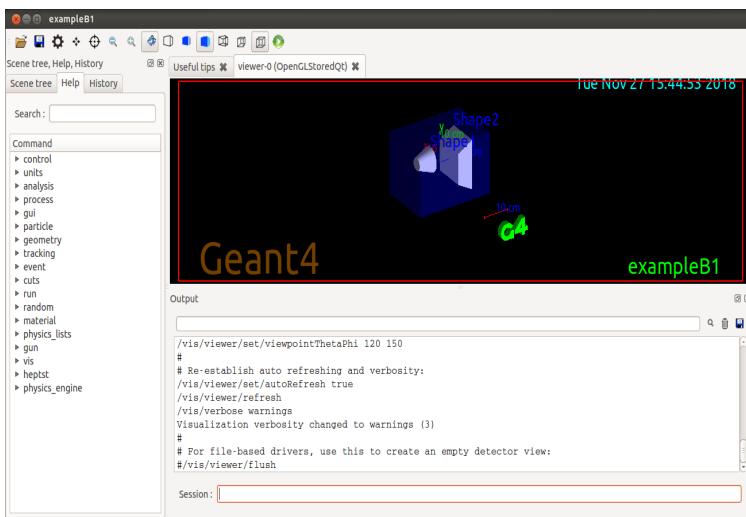


Figura F.3: Captura de pantalla de la interfaz gráfica de usuario para el ejemplo B1.

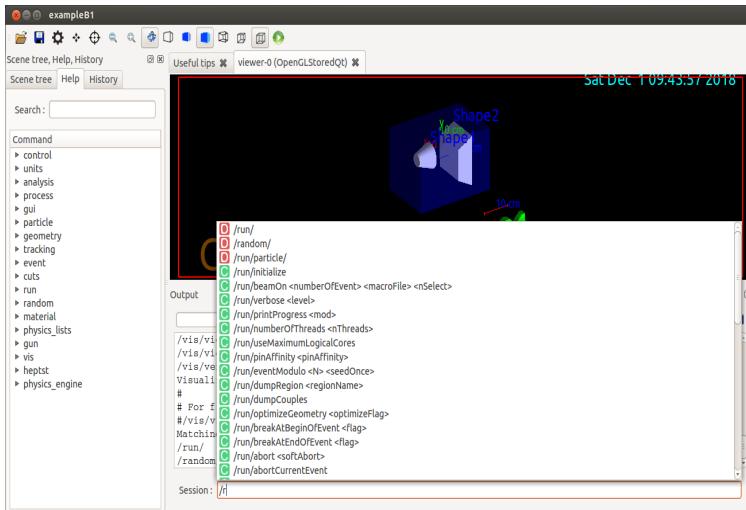


Figura F.4: Ejecución del comando `/run/beamOn 5`: Apenas se digita `/r` aparece una ayuda de los posibles comando a utilizar en la interfaz de GEANT4.

Instalación de GEANT4

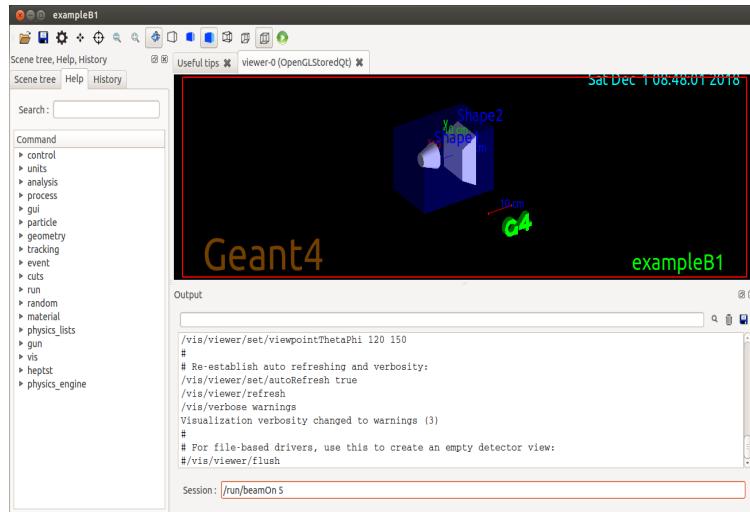


Figura F.5: Ejecución del comando `/run/beamOn 5`: Completando la escritura del comando a utilizar con sus opciones se presiona la tecla “enter”.

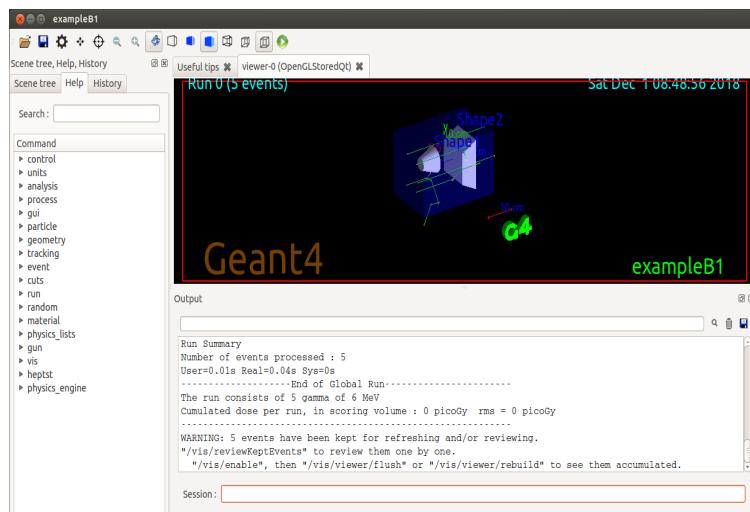


Figura F.6: Visualización de las trayectorias de los rayos que inciden sobre la geometría de la simulación.

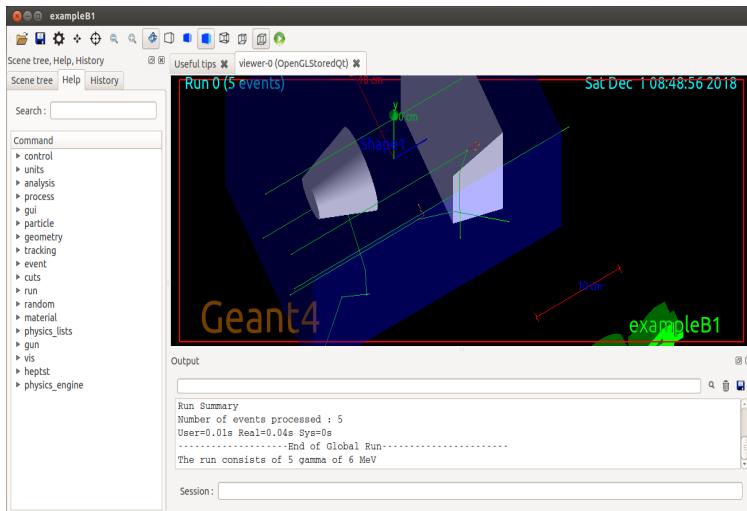


Figura F.7: Magnificación de las trayectorias de los rayos y la geometría de la simulación.

Apéndice G

Materiales predefinidos de *GEANT4*

G.1. Materiales puros

Z	Nombre	Densidad(g/cm^3)	I(eV)
1	G4_H8.3748e-05	19.2	
2	G4_He	0.000166322	41.8
3	G4_Li	0.534	40
4	G4_Be	1.848	63.7
5	G4_B	2.37	76
6	G4_C	2	81
7	G4_N	0.0011652	82
8	G4_O	0.00133151	95
9	G4_F	0.00158029	115
10	G4_Ne	0.000838505	137
11	G4_Na	0.971	149
12	G4_Mg	1.74	156
13	G4_Al	2.699	166
14	G4_Si	2.33	173
15	G4_P	2.2	173
16	G4_S	2	180
17	G4_Cl	0.00299473	174
18	G4_Ar	0.00166201	188
19	G4_K	0.862	190
20	G4_Ca	1.55	191

21	G4_Sc	2.989	216
22	G4_Ti	4.54	233
23	G4_V	6.11	245
24	G4_Cr	7.18	257
25	G4_Mn	7.44	272
26	G4_Fe	7.874	286
27	G4_Co	8.9	297
28	G4_Ni	8.902	311
29	G4_Cu	8.96	322
30	G4_Zn	7.133	330
31	G4_Ga	5.904	334
32	G4_Ge	5.323	350
33	G4_As	5.73	347
34	G4_Se	4.5	348
35	G4_Br	0.0070721	343
36	G4_Kr	0.00347832	352
37	G4_Rb	1.532	363
38	G4_Sr	2.54	366
39	G4_Y	4.469	379
40	G4_Zr	6.506	393
41	G4_Nb	8.57	417
42	G4_Mo	10.22	424
43	G4_Tc	11.5	428
44	G4_Ru	12.41	441
45	G4_Rh	12.41	449
46	G4_Pd	12.02	470
47	G4_Ag	10.5	470
48	G4_Cd	8.65	469
49	G4_In	7.31	488
50	G4_Sn	7.31	488
51	G4_Sb	6.691	487
52	G4_Te	6.24	485
53	G4_I	4.93	491
54	G4_Xe	0.00548536	482
55	G4_Cs	1.873	488
56	G4_Ba	3.5	491
57	G4_La	6.154	501
58	G4_Ce	6.657	523
59	G4_Pr	6.71	535
60	G4_Nd	6.9	546

Materiales predefinidos de GEANT4

61	G4_Pm	7.22	560
62	G4_Sm	7.46	574
63	G4_Eu	5.243	580
64	G4_Gd	7.9004	591
65	G4_Tb	8.229	614
66	G4_Dy	8.55	628
67	G4_Ho	8.795	650
68	G4_Er	9.066	658
69	G4_Tm	9.321	674
70	G4_Yb	6.73	684
71	G4_Lu	9.84	694
72	G4_Hf	13.31	705
73	G4_Ta	16.654	718
74	G4_W	19.3	727
75	G4_Re	21.02	736
76	G4_Os	22.57	746
77	G4_Ir	22.42	757
78	G4_Pt	21.45	790
79	G4_Au	19.32	790
80	G4_Hg	13.546	800
81	G4_Tl	11.72	810
82	G4_Pb	11.35	823
83	G4_Bi	9.747	823
84	G4_Po	9.32	830
85	G4_At	9.32	825
86	G4_Rn	0.00900662	794
87	G4_Fr	1	827
88	G4_Ra	5	826
89	G4_Ac	10.07	841
90	G4_Th	11.72	847
91	G4_Pa	15.37	878
92	G4_U	18.95	890
93	G4_Np	20.25	902
94	G4_Pu	19.84	921
95	G4_Am	13.67	934
96	G4_Cm	13.51	939
97	G4_Bk	14	952
98	G4_Cf	10	966

G.2. Compuestos NIST

Ncomp	Nombre	Densidad(g/cm^3)	I(eV)
6	G4_A-150_TISSUE	1.127	65.1
3	G4_ACETONE	0.7899	64.2
2	G4_ACETYLENE	0.0010967	58.2
3	G4_ADENINE	1.35	71.4
13	G4_ADIPPOSE_TISSUE_ICRP	0.92	63.2
4	G4_AIR	0.00120479	85.7
4	G4_ALANINE	1.42	71.9
2	G4_ALUMINUM_OXIDE_Al_20_3	3.97	145.2
3	G4_AMBER	1.1	63.2
2	G4_AMMONIA	0.000826019	53.7
3	G4_ANILINE	1.0235	66.2
2	G4_ANTHRACENE	1.283	69.5
6	G4_B-100_BONE	1.45	85.9
3	G4_BAKELITE	1.25	72.4
2	G4_BARIUM_FLUORIDE	4.89	375.9
3	G4_BARIUM_SULFATE	4.5	285.7
2	G4_BENZENE	0.87865	63.4
2	G4_BERYLLIUM_OXIDE	3.01	93.2
3	G4_BGO	7.13	534.1
14	G4_BLOOD_ICRP	1.06	75.2
8	G4_BONE_COMPACT_ICRU	1.85	91.9
9	G4_BONE_CORTICAL_ICRP	1.85	106.4
2	G4_BORON_CARBIDE	2.52	84.7
2	G4_BORON_OXIDE	1.812	99.6
13	G4_BRAIN_ICRP	1.03	73.3
2	G4_BUTANE	0.00249343	48.3
3	G4_N-BUTYL_ALCOHOL	0.8098	59.9
5	G4_C-552	1.76	86.8
2	G4_CADMIUM_TELLURIDE	6.2	539.3
3	G4_CADMIUM_TUNGSTATE	7.9	468.3
3	G4_CALCIUM_CARBONATE	2.8	136.4
2	G4_CALCIUM_FLUORIDE	3.18	166
2	G4_CALCIUM_OXIDE	3.3	176.1
3	G4_CALCIUM_SULFATE	2.96	152.3

Materiales predefinidos de **GEANT4**

38	3	G4_CALCIUM_TUNGSTATE	6.062	395
39	3	G4_CELLULOSE_CELLOPHANE	1.42	77.6
40	3	G4_CELLULOSE_BUTYRATE	1.2	74.6
41	4	G4_CELLULOSE_NITRATE	1.49	87
42	5	G4_CERIC_SULFATE	1.03	76.7
43	2	G4_CESIUM_FLUORIDE	4.115	440.7
44	2	G4_CESIUM_IODIDE	4.51	553.1
45	3	G4_CHLOROBENZENE	1.1058	89.1
46	3	G4_CHLOROFORM	1.4832	156
47	10	G4_CONCRETE	2.3	135.2
48	2	G4_CYCLOHEXANE	0.779	56.4
49	3	G4_1,2-DICHLOROBENZENE	1.3048	106.5
50	4	G4_DICHLORODIETHYL_ETHER	1.2199	103.3
51	3	G4_1,2-DICHLOROETHANE	1.2351	111.9
52	3	G4_DIETHYL_ETHER	0.71378	60
53	4	G4_N,N-DIMETHYL_FORMAMIDE	0.9487	66.6
54	4	G4_DIMETHYL_SULFOXIDE	1.1014	98.6
55	2	G4_ETHANE	0.00125324	45.4
56	3	G4_ETHYL_ALCOHOL	0.7893	62.9
57	3	G4_ETHYL_CELLULOSE	1.13	69.3
58	2	G4_ETHYLENE	0.00117497	50.7
59	4	G4_EYE_LENS_ICRP	1.1	73.3
60	2	G4_FERRIC_OXIDE	5.2	227.3
61	2	G4_FERROBORIDE	7.15	261
62	2	G4_FERROUS_OXIDE	5.7	248.6
63	7	G4_FERROUS_SULFATE	1.024	76.4
64	3	G4_FREON-12	1.12	143
65	3	G4_FREON-12B2	1.8	284.9
66	3	G4_FREON-13	0.95	126.6
67	3	G4_FREON-13B1	1.5	210.5
68	3	G4_FREON-13I1	1.8	293.5
69	3	G4_GADOLINIUM_OXYSULFIDE	7.44	493.3
70	2	G4_GALLIUM_ARSENIDE	5.31	384.9
71	5	G4_GEL_PHOTO_EMULSION	1.2914	74.8
72	6	G4_Pyrex_Glass	2.23	134
73	5	G4_GLASS LEAD	6.22	526.4
74	4	G4_GLASS_PLATE	2.4	145.4
75	3	G4_GLUCOSE	1.54	77.2
76	4	G4 GLUTAMINE	1.46	73.3
77	3	G4_LANTHANUM_OXYBROMIDE	6.28	439.7

Materiales predefinidos de GEANT4

78	3	G4_LANTHANUM_OXYSULFIDE	5.86	421.2
79	2	G4_LEAD_OXIDE	9.53	766.7
80	3	G4_LITHIUM_AMIDE	1.178	55.5
81	3	G4_LITHIUM_CARBONATE	2.11	87.9
82	2	G4_LITHIUM_FLUORIDE	2.635	94
83	2	G4_LITHIUM_HYDRIDE	0.82	36.5
84	2	G4_LITHIUM_IODIDE	3.494	485.1
85	2	G4_LITHIUM_OXIDE	2.013	73.6
86	3	G4_LITHIUM_TETRABORATE	2.44	94.6
87	13	G4_LUNG_ICRP	1.05	75.3
88	5	G4_M3_WAX	1.05	67.9
89	3	G4_MAGNESIUM_CARBONATE	2.958	118
90	2	G4_MAGNESIUM_FLUORIDE	3	134.3
91	2	G4_MAGNESIUM_OXIDE	3.58	143.8
92	3	G4_MAGNESIUM_TETRABORATE	2.53	108.3
93	2	G4_MERCURIC_IODIDE	6.36	684.5
94	2	G4 METHANE	0.000667151	41.7
95	3	G4 METHANOL	0.7914	67.6
96	5	G4_MIX_D_WAX	0.99	60.9
97	6	G4_MS20_TISSUE	1	75.1
98	13	G4_MUSCLE_SKELETAL_ICRP	1.04	75.3
99	9	G4_MUSCLE_STRIATED_ICRU	1.04	74.7
100	4	G4_MUSCLE_WITH_SUCROSE	1.11	74.3
101	4	G4_MUSCLE_WITHOUT_SUCROSE	1.07	74.2
102	2	G4_NAPHTHALENE	1.145	68.4
103	4	G4_NITROBENZENE	1.19867	75.8
104	2	G4_NITROUS_OXIDE	0.00183094	84.9
105	4	G4_NYLON-8062	1.08	64.3
106	4	G4_NYLON-6/6	1.14	63.9
107	4	G4_NYLON-6/10	1.14	63.2
108	4	G4_NYLON-11_RILSAN	1.425	61.6
109	2	G4_OCTANE	0.7026	54.7
110	2	G4_PARAFFIN	0.93	55.9
111	2	G4_N-PENTANE	0.6262	53.6
112	8	G4_PHOTO_EMULSION	3.815	331
113	2	G4_PLASTIC_SC_VINYLTOLUENE	1.032	64.7
114	2	G4_PLUTONIUM_DIOXIDE	11.46	746.5
115	3	G4_POLYACRYLONITRILE	1.17	69.6
116	3	G4_POLYCARBONATE	1.2	73.1
117	3	G4_POLYCHLOROSTYRENE	1.3	81.7

Materiales predefinidos de **GEANT4**

118	2	G4_POLYETHYLENE	(C_2H_4)_N-Polyethylene		
119	3	G4_MYLAR	1.4	78.7	
120	3	G4_PLEXIGLASS	1.19	74	
121	3	G4_POLYOXYMETHYLENE	1.425	77.4	
122	2	G4_POLYPROPYLENE	(C_2H_4)_N-Polypropylene		
123	2	G4_POLYSTYRENE	1.06	68.7	
124	2	G4_TEFLON	2.2	99.1	
125	3	G4_POLYTRIFLUOROCHLOROETHYLENE			
126	3	G4_POLYVINYL_ACETATE	1.19	73.7	
127	3	G4_POLYVINYL_ALCOHOL	1.3	69.7	
128	3	G4_POLYVINYL_BUTYRAL	1.12	67.2	
129	3	G4_POLYVINYL_CHLORIDE	1.3	108.2	
130	3	G4_POLYVINYLDENE_CHLORIDE	1.7	134.3	
131	3	G4_POLYVINYLDENE_FLUORIDE	1.76	88.8	
132	4	G4_POLYVINYL_PYRROLIDONE	1.25	67.7	
133	2	G4_POTASSIUM_IODIDE	3.13	431.9	
134	2	G4_POTASSIUM_OXIDE	2.32	189.9	
135	2	G4_PROPANE	0.00187939	47.1	
136	2	G4_lPROPANE	0.43	52	
137	3	G4_N-PROPYL_ALCOHOL	0.8035	61.1	
138	3	G4_PYRIDINE	0.9819	66.2	
139	2	G4_RUBBER_BUTYL	0.92	56.5	
140	2	G4_RUBBER_NATURAL	0.92	59.8	
141	3	G4_RUBBER_NEOPRENE	1.23	93	
142	2	G4_SILICON_DIOXIDE	SiO_2	2.32	139.2
143	2	G4_SILVER_BROMIDE		6.473	486.6
144	2	G4_SILVER_CHLORIDE		5.56	398.4
145	3	G4_SILVER_HALIDES		6.47	487.1
146	2	G4_SILVER_IODIDE		6.01	543.5
147	13	G4_SKIN_ICRP		1.1	72.7
148	3	G4_SODIUM_CARBONATE		2.532	125
149	2	G4_SODIUM_IODIDE		3.667	452
150	2	G4_SODIUM_MONOXIDE		2.27	148.8
151	3	G4_SODIUM_NITRATE		2.261	114.6
152	2	G4_STILBENE		0.9707	67.7
153	3	G4_SUCROSE		1.5805	77.5
154	2	G4_TERPHENYL		1.234	71.7
155	13	G4_TESTES_ICRP		1.04	75
156	2	G4_TETRACHLOROETHYLENE		1.625	159.2
157	2	G4_THALLIUM_CHLORIDE		7.004	690.3

Materiales predefinidos de **GEANT4**

158	13	G4_TISSUE_SOFT_ICRP	1	72.3
159	4	G4_TISSUE_SOFT_ICRU-4	1	74.9
160	4	G4_TISSUE-METHANE	0.00106409	61.2
161	4	G4_TISSUE-PROPANE	0.00182628	59.5
162	2	G4_TITANIUM_DIOXIDE	4.26	179.5
163	2	G4_TOLUENE	0.8669	62.5
164	3	G4_TRICHLOROETHYLENE	1.46	148.1
165	4	G4_TRIETHYL_PHOSPHATE	1.07	81.2
166	2	G4_TUNGSTEN_HEXAFLUORIDE	2.4	354.4
167	2	G4_URANIUMDICARBIDE	11.28	752
168	2	G4_URANIUM_MONOCARBIDE	13.63	862
169	2	G4_URANIUM_OXIDE	10.96	720.6
170	4	G4_UREA	1.323	72.8
171	4	G4_VALINE	1.23	67.7
172	3	G4_VITON	1.8	98.6
173	2	G4_WATER_H_20	1	75
174	2	G4_WATER_VAPOR_H_20-Gas	0.000756182	71.6
175	2	G4_XYLENE	0.87	61.8
176	1	G4_GRAPHITE Graphite	1.7	78
177	1	G4_lH2	0.0708	21.8
178	1	G4_lAr	1.396	188
179	1	G4_lKr	2.418	352
180	1	G4_lXe	2.953	482
181	3	G4_PbW04	8.28	0
182		8	0.140637	
183		82	0.455366	
184		74	0.403998	
185	1	G4_Galactic	1e-25	21.8

G.3. Materiales HEP

1	=====				
2	Ncomp	Nombre	Densidad(g/cm^3)	I(eV)	
3	=====	=====	=====	=====	
4	5	1	G4_lH2	0.0708	21.8
6	6	1	G4_lAr	1.396	188
7	7	1	G4_lKr	2.418	352

Materiales predefinidos de GEANT4

8	1	G4_lXe	2.953	482
9	3	G4_PbWO4	8.28	0
10		8	0.140637	
11		82	0.455366	
12		74	0.403998	
13	1	G4_Galactic	1e-25	21.8

Materiales predefinidos de **GEANT4**

Apéndice H

Programación macro de ROOT para análisis de datos de la simulación-ejemplo-extendida

```
1 #include <fstream>
2 #include <string>
3
4 void anaBrainExp(){
5
6     //parametros para el estilo de las graficas
7     gStyle->SetNdivisions(505,"Y");
8     gStyle->SetNdivisions(505,"X");
9     gStyle->SetLabelSize(0.06,"X");
10    gStyle->SetLabelSize(0.06,"Y");
11    gStyle->SetTitleXOffset(0.8);
12    gStyle->SetTitleYOffset(0.8);
13    gStyle->SetTitleXSize(0.06);
14    gStyle->SetTitleYSize(0.06);
15    gStyle->SetGridColor(4);
16    gStyle->SetOptFit(0100);
17    gStyle->SetOptFit(111);
18    gStyle->SetStatW(0.2);
19    gStyle->SetStatH(0.3);
20
21
22 //*****INICIO ANALISIS EnerDepo.dat ****
23 //*****INICIO ANALISIS EnerDepo.dat ****
24 ifstream edepIn;
25 TCanvas *c1 = new TCanvas("c1","Analisis EnerDepo.dat",10,10,600,1000);
26 c1->Divide(1,2);
```

*Programación macro de ROOT para análisis de datos de la
simulación-ejemplo-extendida*

```
28 c1->SetGrid();
29
30 TH1F *histo = new TH1F("histo","Deposici#acute{o}n acumulada por bin",100,0,100);
31 histo->SetMarkerStyle(21);
32 histo->SetMarkerSize(0.8);
33 histo->SetStats(0);
34 histo->GetXaxis()->SetTitle("N#acute{u}mero de Bines");
35 histo->GetYaxis()->SetTitle("E(bin) (MeV)");
36 histo->GetYaxis()->SetRangeUser(0,140);
37
38 edepIn.open("EnerDepo.dat");
39 string headLine;
40 double datEdepIn[6];
41
42 //saltando las tres lineas de cabecera
43 for(int i=0; i<3; i++){
44     getline(edepIn, headLine);
45     cout << headLine << endl;
46 }
47
48 int nPoints;
49 double x[100];
50 double y[100];
51 double ex[100];
52 double ey[100]; //barra de error en Y
53
54 for(nPoints=0; nPoints<100; nPoints++){
55     for(int nCol=0; nCol<6; nCol++){
56         edepIn >> datEdepIn[nCol];
57         //cout << setprecision(9) << datEdepIn[nCol] << "\t";
58     }
59     histo->SetBinContent(datEdepIn[2], datEdepIn[3]);
60
61     //conversion de bines a longitud:
62     //4.8cm -> 100bines
63     x[nPoints] = datEdepIn[2]*4.8/100.0;
64     y[nPoints] = datEdepIn[3];
65     ey[nPoints] = TMath::Sqrt(datEdepIn[4]);
66     ex[nPoints] = 0;
67
68     cout << x[nPoints] << " " << y[nPoints] << endl ;
69 }
70
71 edepIn.close();
72
73 Double_t par[6];
74
75 //fit gausiano para piel y musculo
76 TF1 *g1      = new TF1("g1", "gaus", 0, 0.13);
```

```
77 //fit gausiano hueso y cerebro
78 TF1 *g2 = new TF1("g2","gaus",0.12, 4.8);
79
80 TF1 *total = new TF1("total","gaus(0)+gaus(3)",0.0, 4.8);
81 total->SetLineColor(4);
82 total->SetParName(0,"N_{1}");
83 total->SetParName(1,"#mu_{1}");
84 total->SetParName(2,"#sigma_{1}");
85 total->SetParName(3,"N_{2}");
86 total->SetParName(4,"#mu_{2}");
87 total->SetParName(5,"#sigma_{2}");
88
89 TGraphErrors *gr = new TGraphErrors(nPoints,x,y,ex,ey);
90 gr->SetTitle("Deposición acumulada por unidad de longitud");
91 gr->GetXaxis()->SetTitle("z_{abs} (cm)");
92 gr->GetYaxis()->SetTitle("E(z_{abs}) (MeV)");
93
94 c1->cd(1);
95 c1->cd(1)->SetGrid();
96 histo->DrawClone();
97
98 c1->cd(2);
99 c1->cd(2)->SetGrid();
100 gr->GetYaxis()->SetRangeUser(0,140);
101 gr->GetXaxis()->SetRangeUser(0,4.8);
102 gr->Fit(g1,"R0+");
103 gr->Fit(g2,"R0+");
104 g1->GetParameters(&par[0]);
105 g2->GetParameters(&par[3]);
106 total->SetParameters(par);
107
108 gr->Fit(total);
109 gr->SetMarkerStyle(33);
110 gr->SetMarkerSize(1);
111 gr->DrawClone("ap");
112
113 cout << "Emax: " << total->Eval(1.207) << endl;
114 //***** FIN ANALISIS EnerDepo.dat *****
115 //***** INICIO ANALISIS BrainExp.root *****
116
117
118
119
120 //////////////// INICIO ANALISIS BrainExp.root ///////////////
121 //      INICIO ANALISIS BrainExp.root          //
122 //////////////////////////////// BrainExp.root //////////////////
123 TCanvas *c2 = new TCanvas("c2","Analisis BrainExp.root",10,10,600,1000);
124 c2->SetGrid();
125 TFile* file = TFile::Open("BrainExp.root");
```

Programación macro de ROOT para análisis de datos de la simulación-ejemplo-extendida

```
126 TH1D* readThis = 0;
127 file->GetObject("EnergyDeposited_per_event", readThis);
128
129 cout << "Entradas: " << readThis->GetEntries() << endl;
130 cout << "Bin del valor maximo (bin) " << readThis->GetMaximumBin() << endl;
131 cout << "Numero de bines (bin) " << readThis->GetNbinsX() << endl;
132 cout << "xmin: " << readThis->GetXaxis()->GetXmin() << endl;
133 cout << "xmax: " << readThis->GetXaxis()->GetXmax() << endl;
134 cout << "Valor mas probable de deposición: " <<
135     (readThis->GetMaximumBin())*
136     (readThis->GetXaxis()->GetXmax())/(readThis->GetNbinsX()) << endl;
137 cout << "Area: " << readThis->Integral("width") << endl;
138
139 c2->cd();
140 readThis->Draw();
141 ///////////////////////////////////////////////////////////////////
142 // FIN ANALISIS BrainExp.root
143 ///////////////////////////////////////////////////////////////////
144
145
146 //*****INICIO ANALISIS NOfCollision.dat*****
147 //      INICIO ANALISIS NOfCollision.dat      *
148 //*****INICIO ANALISIS NOfCollision.dat*****
149 ifstream nofColIn;
150
151 TCanvas *c3 = new TCanvas("c3","Analisis NOfCollision.dat",10,10,600,1000);
152 c3->Divide(1,2);
153 c3->SetGrid();
154
155 TH1F *histo1 = new TH1F("histo1","Número acumulado de colisiones por bin",
156                          100,0,1000);
157 histo1->SetStats(0);
158 histo1->GetXaxis()->SetTitle("Número de Bines");
159 histo1->GetYaxis()->SetTitle("Número de Colisiones");
160 histo1->GetYaxis()->SetRangeUser(0,1500);
161
162 nofColIn.open("NOfCollision.dat");
163 string headLine1;
164 double datNofColIn[6];
165
166 //saltando las tres líneas de cabecera
167 for(int i=0; i<3; i++){
168     getline(nofColIn, headLine1);
169     cout << headLine1 << endl;
170 }
171
172 int nPoints1;
173 double x1[100];
174 double y1[100];
```

```
175 double ex1[100];
176 double ey1[100]; //barra de error en Y
177
178 for(nPoints1=0; nPoints1<100; nPoints1++){
179     for(int nCol1=0; nCol1<6; nCol1++){
180         nofColIn >> datNofColIn[nCol1];
181         //cout << setprecision(9) << datNofColIn[nCol] << "\t";
182     }
183     histo1->SetBinContent(datNofColIn[2], datNofColIn[3]);
184
185     //conversion de bines a longitud:
186     //4.8cm -> 100bines
187     x1[nPoints1] = datNofColIn[2]*4.8/100.0;
188     y1[nPoints1] = datNofColIn[3];
189     ey1[nPoints1] = 0;
190     ex1[nPoints1] = 0;
191
192     cout << x1[nPoints1] << " " << y1[nPoints1] << endl ;
193 }
194
195 nofColIn.close();
196
197 Double_t par1[6];
198
199 //fit gausiano para piel y musculo
200 TF1 *g11 = new TF1("g11","gaus",0, 0.13);
201 //fit gausiano hueso y cerebro
202 TF1 *g22 = new TF1("g22","gaus",0.12, 4.8);
203
204 TF1 *total1 = new TF1("total1","gaus(0)+gaus(3)",0.0, 4.8);
205 total1->SetLineColor(4);
206 total1->SetParName(0,"N_{1}");
207 total1->SetParName(1,"#mu_{1}");
208 total1->SetParName(2,"#sigma_{1}");
209 total1->SetParName(3,"N_{2}");
210 total1->SetParName(4,"#mu_{2}");
211 total1->SetParName(5,"#sigma_{2}");
212
213 TGraphErrors *gr1 = new TGraphErrors(nPoints1,x1,y1,ex1,ey1);
214 gr1->SetTitle("Número acumulado de colisiones por unidad de longitud");
215 gr1->GetXaxis()->SetTitle("z_{abs} (cm)");
216 gr1->GetYaxis()->SetTitle("NC(z_{abs}) (MeV)");
217 gr1->GetYaxis()->SetRangeUser(0,1500);
218 gr1->GetXaxis()->SetRangeUser(0,4.8);
219 gr1->SetMarkerStyle(33);
220 gr1->SetMarkerSize(1);
221
222 c3->cd(1);
223 c3->cd(1)->SetGrid();
```

*Programación macro de ROOT para análisis de datos de la
simulación-ejemplo-extendida*

```
224 histo1->DrawClone();  
225  
226 c3->cd(2);  
227 c3->cd(2)->SetGrid();  
228 gr1->DrawClone("ap");  
229 //*****  
230 //      FIN ANALISIS NOfCollision.dat      *  
231 //*****  
232  
233  
234 } //void anaBrainExp
```

anaBrainExp.C

Referencias

- AAAS. (1957). News of science. *Science*, 125(3236), 18–22. Descargado de <http://science.sciencemag.org/content/125/3236/18> doi: 10.1126/science.125.3236.18
- Agostinelli, S., Allison, J., Apostolakis, J., y cols. (2003). Geant4 – a simulation toolkit. *Nuclear Instruments and Methods in Physics Research A*, 506, 250–303.
- Allison, J., y cols. (2006). Geant4 developments and applications. *EEE Trans. Nucl. Sci.*, 53(2), 270–278.
- Amako, K., y cols. (2000). Present status of geant4. *Nuclear Instruments and Methods in Physics Research A*, 453, 455–460.
- automation syst. & integr., l. (1994). *Int. standard iso-10303*.
- Ayala, E. (2008, Marzo). Notas del curso de física nuclear y de radiaciones (Manual de software informático). Quito.
- Azorín, C. G. (2009). *Simulación de la interacción de fotones en la materia usando el método monte carlo* (Tesis Doctoral no publicada). Instituto Politécnico Nacional, Centro de investigación de ciencia aplicada y tecnología, México DF.
- Beiser, A. (2003). *Concepts of modern physics* (6.^a ed.). New York: McGraw-Hill.
- Bevelacqua, J. (2004). *Contemporary health physics*. WILEY-VCH GmbH & Co. KGaA.
- Blatt, F. (1983). *Principles of physics*. Allyn and Bacon.
- Bock, R., y Vasilescu, A. (1998). *The particle detector briefbook*. Geneva: Springer.
- Brand, F., Falconer, I., Large, M., Sefton, I., Ulrichs, J., Vorlicek, G., y Winn, M. (1993). *Atoms and nuclei* (3.^a ed.) (Manual de software informático). Sydney.
- Burcham, W. (1979). *Elements of nuclear physics*. Harlow: Longman.
- Bustos, F. (2002). *Estrategias para el mejoramiento de las interacciones docente-alumnos*. Serie Nuevos Horizontes Educativos.
- Campbell, J. R., y Ross, R. C. (1995). *Las radiaciones ii. el manejo de las radiaciones nucleares*. México DF: Fondo de cultura económica.
- Carrillo, G. (2006). *Simulación con geant4 de un instrumento para hacer imagenología por retrodispersión gamma* (Tesis de Master no publicada). Facultad de Ciencias Universidad Nacional de Colombia, Bogotá.
- Castañeda, M. G., y De-Geus, J. E. (2008). *Introducción a la física moderna* (3.^a ed.). Universidad Nacional de Colombia: Unibiblos.
- Cern. (2011a, Citado en Octubre de). *Atlas*. <http://public.web.cern.ch/Public/en/LHC/ATLAS-en.html>. Descargado de <http://public.web.cern.ch/Public/en/LHC/ATLAS-en.html>
- Cern. (2011b, Citado en Febrero de). *Atlas experiment*. <http://atlas.web.cern.ch/Atlas/Collaboration/>
- Cern, G. (2011, Citado en Febrero de). *Documentación de cabecera (en línea)*. <http://geant4.cern.ch/>. Descargado de <http://geant4.cern.ch/>

- Cern, R. (2011, Citado en Febrero de). *Root users guide*. <http://root.cern.ch/drupal/>. Descargado de <http://root.cern.ch/drupal/>
- Cristancho, F. (2011, Citado en Octubre de). *Laboratorio de radiación gamma y alpha gfnun*. <http://www.gfnun.unal.edu.co/index.php?pagina=labgammas>. Descargado de <http://www.gfnun.unal.edu.co/index.php?pagina=labgammas>
- Daoud, A. (s.f.). *Curso de introducción a la tecnología nuclear* (Manual de software informático).
- Das, A. . F. T. (2005). *Introduction to nuclear and particle physics* (2.^a ed.). Rochester: World scientific publishing.
- Dellacqua, A., y cols. (1994). Geant4: an object-oriented toolkit for simulation in hep. *CERN-DRDC*.
- Denner, A., y Dittmaier, S. (1998). *Complete $\alpha(\alpha)$ qed corrections to polarized compton scattering* (Inf. Téc.). CERN, Suiza: Paul Scherrer Institut - Theory Division CERN.
- Eduardo Héctor Scarnichia, C. D. F. (2000). *Protección radiológica*. Instituto Balseiro de Protección Radiológica. Bariloche.
- Eisberg, R., y Resnick, R. (1985). *Quantum physics of atoms, molecules, solids, nuclei, and particles* (2.^a ed.). New York: John Wiley & Sons.
- Elford, H., y Cunningham, J. (1983). *The physics of radiology*. Charles Thomas Publisher.
- Falconer, J. (2011, Citado en Septiembre de). *The museum at the cavendish laboratory*. <http://www-outreach.phy.cam.ac.uk/camphy/museum/area2/cabinet1.htm>. Descargado de <http://www-outreach.phy.cam.ac.uk/camphy/museum/area2/cabinet1.htm>
- Foundation, F. S. (s.f.). *Open source initiative osi - the bsd license: Licensing*. <http://www.xfree86.org/3.3.6/COPYRIGHT2.html>. Descargado de <http://www.xfree86.org/3.3.6/COPYRIGHT2.html#6>
- Geant4, C., y cols. (2010). Geant4 user's guide for toolkit developers (Manual de software informático). Cern Geant4.
- Giani, S., y cols. (1998). Geant4: An object-oriented toolkit for simulation in hep. *CERN/LHCC*.
- Grillo, A. (2011, Citado en Febrero de). *The atlas experiment*. <http://scipp.ucsc.edu/personnel/atlas.html>. Descargado de <http://scipp.ucsc.edu/personnel/atlas.html>
- Grupen, C., y Schwartz, B. (2008). *Particle detectors* (2.^a ed.). Cambridge: Cambridge University Press.
- Guadalupe, S. (2003). El profesor ante las nuevas tecnologías de información y comunicación, ntic. *Con Texto Educativo, Revista digital de educación y nuevas tecnologías*(28).
- Hake, R. (1998, enero). Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *Am. J. Phys.*(1).
- Hurtado, A., y cols. (2006). *Experimento y simulación: Opciones didácticas en la enseñanza-aprendizaje de la física*. Bogotá: Fondo de Publicaciones Universidad Distrital Francisco José de Caldas.

- ICRP, C. (2011, Citado en Julio de). *International commission on radiological protection*. <http://www.icrp.org/>. Descargado de <http://www.icrp.org/>
- Kaplan, I. (1964). *Nuclear physics* (2.^a ed.). Addison-Wesley Publishing Company, Inc.
- Kittel, C. (1995). *Introduction to solid state physics* (7.^a ed.). Wiley.
- Knoll, G. (2000). *Radiation detection and measurement* (3.^a ed.). Wiley.
- Kofman, H. (2000, diciembre). Modelos y simulaciones computacionales en la enseñanza de la física. *Revista Educación en Física*, 13–22.
- Landau, L. (1944). On the energy loss of fast particles by ionization. *J. Phys. USSR*, 8(201).
- Landau, L., y Lifshitz, E. (1992). *The classical theory of fields* (3.^a ed., Vol. 2). Pergamon: Reverté.
- Leo, W. R. (1994). *Techniques for nuclear and particle physics experiments* (2.^a ed.). Lausanne: Springer.
- Levin, C., y Gonin, R. (1988). Computed limitations of spot scanning for therapeutic proton beams. *Int. J. Biomed. Comput.*(23), 33–41.
- Majola, C., y Kettledas, A. (2011, Citado en Septiembre de). *Measuring radiation*. <http://library.thinkquest.org/C0126323/measure.htm>. Descargado de <http://library.thinkquest.org/C0126323/measure.htm>
- Mathematica. (2012). *Ionizing radiation*. <http://www.wolframalpha.com/>.
- Meyer, P. (1970). *Introductory probability and statistical applications* (2.^a ed.). Addison Wesley.
- Montesinos, G. D. (2001). *Un algoritmo simple para la simulación del paso de fotones ($e \leq 1$ mev) a través de la materia* (Tesis Doctoral no publicada). Universidad Complutense de Madrid, Madrid.
- Moyal, J. (1955). Theory of ionization fluctuations. *Phil. Mag.*, 46, 263–80.
- Nieto, J., Minor, A., Álvarez, J., Alonso, M., y Carballo, M. (2007). Desarrollo de un marco estereotáctico para aplicación en neurocirugía y radiocirugía. *Científica*, 11(001), 11–15.
- Pazdur, R., y cols. (2003). *Cancer management: A multidisciplinary approach: Medical, surgical and radiation oncology*. PRR Inc.
- Pérez, C. (1996). *Sensores ópticos*. Valencia: Universitat de València.
- Perl, J. (2011, Citado en Febrero de). *Installing geant4 release 4.9.4 on linux*. http://geant4.slac.stanford.edu/tutorial/installation/Geant4.9.4/Linux/Geant4_9_4_Linux_Installation.htm. Descargado de http://geant4.slac.stanford.edu/tutorial/installation/Geant4.9.4/Linux/Geant4_9_4_Linux_Installation.htm
- Ríos, D. (2000). *Simulación, métodos y aplicaciones*. Alfa Omega.
- Rodríguez, M., y Martínez, A. (1995). *El uso de los rayos x en la medicina*. México: Instituto de Física, UNAM. Descargado de <http://www.smf.mx/boletin/oct-95/ray-med.html>
- Sanz, A. T. (1970). *Instrumentación nuclear*. Madrid: Servicio de Publicaciones de la Junta de Energía Nuclear.

- Tavernier, S. (2010). *Experimental techniques in nuclear and particle physics*. New York: Springer.
- Thwaites, D., y cols. (2006). Back to the future: the history and development of the clinical linear accelerator. *Phys. Med. Biol.*, 51.
- Tsoulfanidis, N. (1995). *Measurement and detection of radiation* (2.^a ed.). University of Missouri-Rolla: Taylor&Francis.
- Udías, J., Lopéz, J., y Fraile, L. (2011, Citado en Septiembre de). *Detectores gaseosos*. http://nuc1.fis.ucm.es/webgrupo_2007/Lab_Detector_Gaseoso.html#proporcional. Descargado de http://nuc1.fis.ucm.es/webgrupo_2007/Lab_Detector_Gaseoso.html#proporcional
- Vaiana, G. (2011, Citado en Septiembre de). *Xact facility - technical description*. <http://www.astropa.unipa.it/XACT/Xdetect.html>. Descargado de <http://www.astropa.unipa.it/XACT/Xdetect.html>
- Vallina, M. M. (2005). *Fundamentos de ingeniería nuclear*. Madrid: UNED.
- Vargas-Hernández, A., y Salamanca-Bernal, J. (2015). *Rootlets, interfaces gráficas en root. un recurso educativo para la enseñanza de la física: Alfabetización científica y tecnológica*. Editorial Académica Española.
- Wittgenstein, L. (1968). *Philosophical investigations* (3.^a ed.). New York: Macmillan.
- Yao, W. (2006). Review of particle physics. *Journal of Physics G: Nuclear and Particle Physics*, 33(1).

Impreso en papel bond 90 gr.
en familia tipográfica Candara a 11,5 pts.

Amadgraf Impresores Ltda.
Bogotá, D.C., Colombia
Octubre de 2020.

OTROS TÍTULOS DE ESTA COLECCIÓN

**REDES NEURONALES
CONVOLUCIONALES
USANDO KERAS Y
ACELERANDO CON GPU**

**GESTIÓN DE LA ENERGÍA:
EL USUARIO DE ENERGÍA
COMO PARTE ACTIVA
DEL SISTEMA**

**GESTIÓN Y CIBERSEGURIDAD
PARA MICRORREDES
ELÉCTRICAS RESIDENCIALES**

**DETECCIÓN Y CORRECCIÓN
DE PROPAGACIONES
ANÓMALAS EN RADARES
METEOROLÓGICOS**

**INTRODUCCIÓN A LA
INVESTIGACIÓN SOBRE
DESASTRES NATURALES Y
CIUDADES INTELIGENTES**

**INVESTIGACIÓN EN INGENIERÍA
FUNDAMENTADA EN LA
GESTIÓN DEL CONOCIMIENTO**

**LOS RECURSOS DISTRIBUIDOS
DE BIOENERGÍA EN COLOMBIA**

**ARQUITECTURAS DE RED
NEURO-CONVOLUCIONAL
PARA APLICACIONES DE
ROBÓTICA ASISTENCIAL**

Se presentan resultados de investigación documental y formativa sobre el ejercicio computacional de uso, manejo y apropiación de software libre en sistema operativo Linux, a propósito de elaborar una simulación computacional guiada (acelerador lineal para tratamiento de médico) en GEANT4, con la finalidad de orientar el objetivo de un proyecto de investigación, sobre la realización de una simulación CPMD de óxidos de titanio en sistemas memristivos, en el marco de colaboración conjunta de la líneas de investigación de los grupos I+D+i FISINFOR (Física e Informática) y LIFAE (Laboratorio de Investigación en Fuentes Alternas de Energía), de la Universidad Distrital. Adicionalmente, los productos de investigación presentados se extienden a la formación de alto nivel de profesores-investigadores en una basta gama de áreas temáticas en ciencias e ingeniería que utilizan la plataforma computacional GEANT4 como conjunto de recursos computacionales para casos específicos de interacción radiación - materia.

Palabras Clave: Simulación computacional, software científico, software libre Linux, GEANT4, Alfabetización Científica y Tecnológica.

ISBN 978-958-787-230-9



9 789587 872309