

UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE CIENCIAS

Sección de Posgrado y Segunda Especialización Profesional



Tesis para Optar el Grado Académico
de Maestro en Ciencias con Mención en
Física

**“Calibración del detector Minerva
mediante la detección de Muones”**

Presentada por

Kenyi Paolo Hurtado Anampa

LIMA-PERÚ

2009

*Dedicado a todos aquellos que hicieron posible
la realización de este trabajo.*

Agradecimientos

Quiero agradecer a mi asesor Javier Solano, por el constante apoyo, las horas de dedicación y los consejos brindados, sin mencionar que sin su ayuda no habría sido posible mi estancia en el Fermilab, y por tanto me hubiera sido imposible la realización de ésta tesis.

Al Director del Fermilab Dr. Piermaría Oddone, por el apoyo a los estudiantes peruanos.

A toda la colaboración MINERVA , especialmente al Dr. Jorge Morfin, Mike Kordosky, Steve Dytman, Paul Rubinov y Ben Ziemer, por la asesoría que me dieron durante mi estancia en el proyecto.

A CONCYTEC, por brindarme el apoyo financiero necesario para realizar el viaje.

A CTIC (Centro de Tecnologías de Información y Comunicaciones), por el apoyo brindado durante todo este tiempo.

Al IGI (Instituto General de Investigación de la UNI), por el apoyo económico concedido.

A la Sección de Posgrado de la Facultad de Ciencias de la UNI, por el apoyo brindado.

Quiero agradecer a mi familia, a mis padres José y Rosa por apoyarme siempre en todo momento, a mis hermanos Kervi y Keiko que quiero mucho y siempre preguntaban por mí. A mi tíos Lucho, Oscar y Gladys y mi prima Joisy, por preocuparse por mí y apoyarme cuando lo necesitaba.

A mi enamorada Angela, por ser tan comprensiva y paciente durante los meses que estuve fuera, y por incentivarme siempre a seguir adelante, gracias mi vida.

Finalmente, a mis amigos de la UNI, PUCP, CBPF y Guanajuato, a Jorge, Marcos, Adolfo, Yvan, Abraham, Jorge C., Arturo, Carlos, Carmen, Aaron, Douglas y Francisco.

Resumen

El experimento $\text{Miner}\nu\text{A}$ (Main Injector Experiment $\nu - A$ interaction) [1] es un detector -altamente segmentado- de neutrinos, capaz de registrar eventos con alta precisión (más de trece millones de eventos en cuatro años de toma de datos), que utiliza el haz NuMI (Neutrino Main Injector) en el Fermi National Accelerator Laboratory [2].

El detector permitirá un estudio detallado de interacciones neutrino-núcleo que servirá para el soporte de experimentos de estudios de oscilaciones de neutrinos como MINOS [3], además del estudio de efectos nucleares en interacciones de neutrinos. Actualmente el experimento se encuentra en su segundo año de construcción, y en la finalización del detector prototipo de $\text{Miner}\nu\text{A}$ para el rastreo de trayectorias. En esta tesis de maestría mostramos el trabajo hecho en el detector prototipo del experimento, en el grupo de reconstrucción de trayectorias en:

- Análisis de estabilidad y calibración de tubos fotomultiplicadores y tarjetas electrónicas de lectura del experimento $\text{Miner}\nu\text{A}$ en el FERMILAB.
- Búsqueda de muones en un detector prototipo de $\text{MINER}\nu\text{A}$ para la detección de trayectorias
- Mejoras a la visualización de datos con la geometría del detector mediante el software VISTA e implementación inicial del Objeto FilterTagList para la etiquetación de diferentes eventos físicos en los datos tomados en el paquete de definición de objetos del software $\text{MINER}\nu\text{A}$ (MinervaEvent) basado en Gaudi, y el empleo del mismo en un algoritmo básico para clasificar eventos con muones

Abstract

MINER ν A Experiment (Main Injector Experiment $\nu - A$ interaction) [1] is a highly segmented neutrinos detector, able to record events with high precision (over than thirteen million events in a four year run), using the NuMI Beam (Neutrino Main Injector) at the Fermi National Accelerator Laboratory [2].

The detector will allow a detailed study of neutrino-nucleon interactions that will be useful for supporting neutrino oscillations experiments like MINOS [3] besides neutrino interactions nuclear effects studies. Currently, the experiment is in its second year of construction and has finalised the Tracking Prototype Detector. In this thesis we present the work done in such Prototype Detector in the tracking reconstruction team on:

- PMT stability lectures and FEB calibration for the MINER ν A experiment at Fermilab
- Muons searching in the Tracking Prototype Detector at MINER ν A for tracking detection
- Improvemens in data visualization with detector geometry through VISTA software and initial implementacion of the FIIterTagList objets for labeling different kind of physics events in the data taken in the Object Model Package MinervaEvent based on Gaudi, and the employment of this object in the basic algorithm to classify muon events.

Colaboración MINER ν A

I.Albayrak⁷, L. Aliaga¹⁵, M. Andrews⁵, S. Avvakumov¹⁶, S. Angelidakis¹, C.Araujo¹⁵, Baldin⁵, J.L. Bazo¹⁵, G. Blazey¹², D.Beringer²¹, A.Butkevich⁸, A. Bodek¹⁶, D. Boehnlein⁵, S. Boyd¹⁴, R. Bradford¹⁶, W.K. Brooks¹⁰, A. Bruell¹⁰, D. Buchholz¹³, H. Budd¹⁶, D. Casper³, C. Castromonte², D. Cherdack¹⁹, M.E. Christy⁷, J. Chvojka¹⁶, J. Castorena⁶, M.A.C. Cummings¹², **A.Chamorro**²⁰, D.S. Damiani²¹, S. Dytman¹⁴, R.DeMaat⁵, R. Ent¹⁰, B.Eberly¹⁴, G.A.Fiorentini², J. Felix⁶, R. Flight¹⁶, A. Gago¹⁵, H. Gallagher¹⁹, D. Gaskell¹⁰, R. Gilman¹⁷, C. Gingu⁵, R. Gran¹¹, B. Gobbi¹², J. Grange²², R. Gutierrez⁶, N. Grossman⁵, D. A. Harris⁵, A. Higuera⁶, J. Hobbs¹², **K. Hurtado**²⁰, T. Kafka¹⁹, C.E Keppel⁷, J. Kilmer⁵, Kopp¹⁸, S.Kulagin⁸, G. Kumbartzki¹⁷, M. Kordosky²¹, L. Loiacono¹⁸, H. Lee¹⁶, A. Leister²¹, S. Manly¹⁶, E. Maher²³, W.A. Mann¹⁹, K. McFarland¹⁶, D. Meekins¹⁰, W. Melnitchouk¹⁰, A. Mislivec¹⁶, G. Moreno⁶, J.G. Morfin⁵, J. Moussau²², H. da Motta², D. Naples¹⁴, J.K. Nelson²¹, G. Niculescu⁹, I. Niculescu⁹, W. Oliver¹⁹, B. Osmanov²², J.Palomino², V. Paolone¹⁴, J. Park¹⁶, L. Patrick¹³, G. Perdue¹⁶, E. Paschos⁴, C. Perez¹⁵, A. Pla-Dalmau⁵, M. Proga¹⁸, R. Ransome¹⁷, M. A. Reyes⁶, **C. Romero**²⁰, C. Rude¹¹, P. Rubinov⁵, H. Ray²², P. Shanahan⁵, H. Schellman¹³, E. Schulte¹⁷, C. Simon³, **C. J. Solano**²⁰, P. Stamoulis¹, D. Schmitz⁵, N. Tagg²⁴, G. Tzanakos¹, Z. Urrutia⁶, M. Vaz², T. Walton⁷, S. Wood¹⁰, G. Zavala⁶, B. Ziemer³

University of Athens, Athens, Greece¹

Centro Brasileiro de Pesquisas Fisicas, Rio de Janeiro, Brazil²

University of California, Irvine, California³

University of Dortmund, Dortmund, Germany⁴

Fermi National Accelerator Laboratory, Batavia, Illinois⁵

Universidad de Guanajuato, Guanajuato, Mexico⁶

Hampton University, Hampton, Virginia⁷

Institute for Nuclear Research, Moscow, Russia⁸

James Madison University, Harrisonburg, Virginia⁹

Jefferson Lab, Newport News, Virginia¹⁰

University of Minnesota Duluth, Duluth, Minnesota¹¹

Northern Illinois University, DeKalb, Illinois¹²

Northwestern University, Evanston, IL¹³

University of Pittsburgh, Pittsburgh, Pennsylvania¹⁴

Pontificia Universidad Catolica del Peru, Lima, Peru¹⁵

University of Rochester, Rochester, New York¹⁶
Rutgers University, New Brunswick, New Jersey¹⁷
University of Texas-Austin, Texas¹⁸
Tufts University, Medford, Massachusetts¹⁹
Universidad Nacional de Ingenieria, Lima, Perú²⁰
William and Mary College, Williamsburg, Virginia²¹
University of Florida, Gainesville, Florida²²
Massachusetts College of Liberal Arts, North Adams, Massachusetts²³
Otterbein College, Westerville, Ohio²⁴

Agradecimientos	II
Resumen	III
Abstract	IV
Colaboración MINERνA	V
1. Teoría de Neutrinos	2
1.1. Introducción	2
1.1.1. Neutrinos Atmosféricos	4
1.1.2. Neutrinos Solares	4
1.1.3. Oscilaciones de Neutrinos	4
1.2. Neutrinos en el Modelo Estándar	5
1.3. Neutrinos masivos	7
1.4. Interacciones Neutrino-Nucleón	9
1.4.1. Dispersión Quasi-elástica	9
1.4.2. Dispersión Profundamente Inelástica	10
1.4.3. Producción Coherente, Resonante de Piones	11
2. Descripción del experimento MINERνA	12
2.1. Descripción General del Experimento	12
2.2. La fuente de neutrinos (El haz NuMI)	12
2.3. Descripción General del Detector	13
2.4. Los tubos fotomultiplicadores	14

2.5.	Tiras de Centelleador	15
2.6.	Calorímetros y Blancos	15
2.6.1.	Calorímetros Electromagnéticos	16
2.6.2.	Calorímetros Hadrónicos	17
2.6.3.	Blancos Nucleares	18
2.7.	Electrónica, DAQ (Adquisición de Datos)	19
2.7.1.	Tarjetas Electrónicas de Lectura (FEB)	20
2.7.2.	Descripción del Software para la Adquisición de Datos	21
2.8.	Pruebas Experimentales programadas en el Experimento	21
2.8.1.	Mapeo de Centelleadores	21
2.8.2.	Detector de Prueba (TestBeam Detector)	22
2.8.3.	Detector Prototipo (Tracking Prototype Detector)	22
2.9.	Pruebas al detector completo	23
3.	Calibración de los PMT y FEB	25
3.1.	Introducción	25
3.2.	Estudios de Crosstalk Eléctrico	26
3.2.1.	Procedimiento Experimental	26
3.2.2.	Coeficientes de Correlación y Crosstalk Eléctrico	27
3.3.	Estudios de linealidad en las mediciones	32
3.3.1.	Pruebas con medidas “Pedestales” (sin inyección de luz)	32
3.3.1.1.	Descripción	32
3.3.1.2.	Resultados	32
3.3.2.	Pruebas con medidas a MáxPE (con “máxima” inyección de luz)	34
3.3.2.1.	Descripción	34
3.3.2.2.	Resultados	35
3.4.	Estudio de filtraciones de luz en los tubos fotomultiplicadores	37
3.4.1.	Procedimiento Experimental	38
3.4.2.	Análisis de filtraciones de luz	38
3.5.	Implementación de Histogramas Online en GAUDI para los discriminadores en las tarjetas electrónicas de lectura en el paquete ForestTest del software MINER ν A	39
3.6.	Cálculo de Ganancia de los Tubos Fotomultiplicadores	42
3.6.1.	Procedimiento Experimental	42
4.	Búsqueda de Muones y software de visualización	47
4.1.	Conversión de los datos de RawDigits a IDDigits	47
4.2.	Búsqueda de Muones utilizando los IDDigits	49

4.2.1.	Resultados sin calibración electrónica	50
4.2.2.	Resultados con calibración electrónica	51
4.3.	Software de Visualización de Eventos	53
4.3.1.	IDDigits en Vista	54
4.3.2.	Implementación de Múltiples Vistas para la visualización de las proyecciones de los digits en simultáneo	54
4.3.3.	Cortes de Energía	56
4.3.3.1.	Manejo de Gaudi y Vista mediante Python	58
4.4.	Implementación del Objeto FilterTagList para la etiquetación de diferentes eventos físicos en los datos tomados en el paquete de definición de objetos del software MINERvA (MinervaEvent)	61
4.5.	FilterEvent Software	62
5.	Resultados y Conclusiones	64
A.	Introducción al Software MINERvA	66
A.1.	Repositorios CVS	67
A.2.	Utilizando el AFS	68
A.3.	Utilizando el software en CTIC-UNI	69
B.	Introducción a los paquetes usados del software MINERvA	72
C.	Descripción del modelo de objetos usados en el Software (Event Model)	79
D.	Programas Adicionales	83
D.1.	Búsqueda de Filtraciones de luz	83
D.2.	Cálculo de Ganancias e Niveles de Intensidad de Luz	84
D.2.1.	Script en bash para su utilización a nivel masivo	86
D.2.2.	Ejemplo de Archivo de Datos de Entrada	86
D.2.3.	Archivo principal de Algoritmo básico de reconocimiento de muones	87
D.2.3.1.	Declaración de las clases necesarias en Gaudi para el algoritmo	89
Bibliografía		91

Índice de figuras

1.1. Leon Lederman, Melvin Schwartz y Jack Steinberger, ganadores del Nobel de 1988 por demostrar la existencia del neutrino muónico	3
2.1. Diseño del NuMIBeam y el Hall NuMI	13
2.2. Detector Minerva	14
2.3. Tira de Centelleador trinangulos para el detector interior	15
2.4. Vista de un módulo activo del detector, junto con los 3 planos de centelleador X,U,V rotados $\pm 60^\circ$ respecto al otro	17
2.5. Calorímetros en Minerva	17
2.6. Blancos Nucleares: Rojo=Fe, Gris=Pb, Negro=C	19
2.7. Esquema electrónico de los FEB	20
2.8. Diseño del Mapeo de Centelleadores	22
2.9. Diseño Conceptual del TestBeam Detector	23
2.10. Detector Prototipo	24
3.1. Esquema de una Tarjeta Electrónica de Lectura utilizada en MINER ν A	26
3.2. Cable óptico manipulado para el estudio	27
3.3. Equipo utilizado	27
3.4. Ilustración gráfica de σ para una muestra de datos	28
3.5. Un ejemplo de coeficientes de correlación para una distribución x,y	29
3.6. Resultados PMT25/FEB331, píxel 46	30
3.7. Resultados PMT30/FEB331, píxel 46	30
3.8. Coeficientes de Crosstalk Eléctrico con píxel 46 iluminado	31

3.9. Estabilidad de la toma de datos de las medidas Pedestales $c/hora$ durante más de 1 día	33
3.10. Desviación estándar del promedio de las Medias de carga en cada píxel	34
3.11. Estabilidad de la toma de datos de las medidas Pedestales $c/hora$ durante casi 1 día	36
3.12. Desviación estándar del promedio de las Medias de carga en cada píxel	37
3.13. Ejemplo de filtraciones de luz en una lectura	39
3.14. Estructura ramificada del objeto (Tree) en Root con los histogramas implementados	40
3.15. Gráfico de Histogramas implementados para los Discriminadores en cada FEB .	41
3.16. Ejemplo de Ajuste Trilineal en una de las tarjetas electrónicas de lectura	43
3.17. Distribución de Carga después de la calibración	43
3.18. Esquema de Procedimientos para el Cálculo de Ganancia	44
3.19. Croc 1, Board 3, Chain 1	45
3.20. Croc 1, Board 6, Chain 5	46
4.1. Diagrama de algoritmos ejecutados para la obtención de IDDigits	48
4.2. Estructura de los tipos de archivo RawDigits, SupDigits y IDDigits, objetos contenidos en cada tipo de archivo	49
4.3. Esquema General para la búsqueda de muones desde la toma de datos	49
4.4. Energía Total por Evento (ADC Counts) vs Número de Hits	50
4.5. Líneas circulares encerrando a las dos distribuciones mostradas en la figura 4.4(b)	51
4.6. Energía Total del Evento (pC) vs Número de Hits	51
4.7. Trayectorias de un muon en el Plano X	52
4.8. Muon a través de 2 módulos	53
4.9. Esquema de objetos principales usados desde la toma de datos hasta su visualización en VISTA	53
4.10. Visualización de proyecciones en diferentes planos de manera simultánea	55
4.11. Archivo principal de implementación de cortes de energía en SoApp/Vista	57
4.12. Cortes de Energía en Vista	58
4.13. Manero de Vista a través de Python	60
4.14. Esquema de Objetos en MinervaEvent	61
4.15. Esquema de Objetos en MinervaEvent	62
4.16. Histograma mostrando el número de eventos con muones encontrados	63

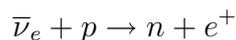
Índice de cuadros

3.1. Coeficientes de Correlación: Píxel 46, PMT25/FEB331	30
3.2. Coeficientes de Correlación: Píxel 46, PMT30/FEB331	31
3.3. Equivalencia entre los valores de Entrada en Voltios y ADC Counts	38
3.4. Ejemplo de tabla producida por el algoritmo PedestalFinder	42
C.1. IDs de algunas clases de Minerva	79

1.1. Introducción

A fines de 1930 el físico austriaco Wolfgang Ernst Pauli propuso la existencia de los neutrinos, en una carta a los participantes de una conferencia de Física en Tübingen (Alemania), a fin de explicar la aparente no conservación de energía producida en los decaimientos beta. Esta partícula debería de ser de carga neutra y extremadamente liviana, razón por la cual en 1933, Enrico Fermi propuso el nombre de *neutrino* para esta partícula, que es el equivalente en italiano a "neutronicito". Fermi desarrolló una teoría para el decaimiento beta, en la que el neutrino jugaba un rol importante.

En 1956 Frederick Reines y Clyde Cowan reportan la primera evidencia de neutrinos, utilizando un reactor de fisión como fuente de (anti)neutrinos. El anti-neutrino interactuó con protones del blanco usado (400 litros de agua con cloruro de cadmio), originando un positrón (e^+) y un neutrón¹.



En 1957, el italiano Bruno Pontecorvo formula la teoría de *oscilación de neutrinos*, mostrando que si existen diferentes especies de neutrinos, éstas podrían ser capaces de oscilar entre ellas

¹El e^+ se aniquila con un e^- del blanco originando dos fotones simultáneos. El neutrón desacelera antes de ser capturado eventualmente por un núcleo de cadmio, originando la emisión de fotones alrededor de 15 μseg después de los del positrón. Todos esos fotones son detectados, y los 15 μseg de diferencia identifican interacción de un neutrino



Figura 1.1: Leon Lederman, Melvin Schwartz y Jack Steinberger, ganadores del Nobel de 1988 por demostrar la existencia del neutrino muónico

(cambio de sabores).

El descubrimiento del neutrino muónico (ν_μ) se hizo en 1962, por un grupo de científicos del laboratorio Brookhaven y la Universidad de Columbia, utilizando un haz de protones energéticos del AGS² para producir una cascada de piones que viajó poco más de 21m hacia una pared de acero de cinco mil toneladas. En el camino, éstos decayeron en muones y neutrinos, pero solo éstos últimos pudieron atravesar la pared, llegando a un detector lleno de Neón llamado *spark chamber*.

Allí, el impacto de los neutrinos sobre placas de aluminio produjeron rastros de muones que fueron detectados y fotografiados, demostrando la existencia de los neutrinos muónicos (ν_μ). Leon Lederman, Melvin Schwartz y Jack Steinberger ganaron el premio Nobel por éste descubrimiento (ver figura 1.1).

En 1973, un grupo del CERN [5], utilizó una cámara de burbujas (Gargamelle) para la primera muestra de eventos de corriente neutra".

Dos años después el leptón τ es descubierto por un grupo dirigido por el físico Martin Perl, en SLAC³, lo cual más adelante dio evidencia de la existencia de una tercera especie, el neutrino tauónico ν_τ [6] que fue finalmente descubierto el 2000 en el experimento DONUT [7] del FER-MILAB [8].

²AGS: Brookhaven's **A**lternating **G**radient **S**ynchrotron[4]

³SLAC:Stanford Linear Accelerator Center

1.1.1. Neutrinos Atmosféricos

La Tierra es continuamente bombardeada por rayos cósmicos, constituídos en buena proporción por protones y electrones de altas energías. Los protones, en sus colisiones con núcleos en la atmósfera superior producen piones, los cuales decaen de la siguiente manera [9]:

$$\pi^+ \rightarrow \mu^+ + \nu_\mu \quad , \quad \pi^- \rightarrow \mu^- + \bar{\nu}_\mu \quad (1.1)$$

Los neutrinos y anti-neutrinos son producidos a una altura de ~ 20 km. A altas energías, los kaons también contribuyen en la producción de neutrinos y muones. Los muones que decaen producen electrones y neutrinos electrónicos y muónicos a través del proceso:

$$\mu^+ \rightarrow e^+ + \nu_e + \bar{\nu}_\mu \quad , \quad \mu^- \rightarrow e^- + \bar{\nu}_e + \nu_\mu \quad (1.2)$$

El experimento SuperKamiokande [10] encontró una diferencia entre el flujo de ν_μ proveniente desde la atmósfera, encima de nosotros, con respecto al flujo proveniente de las antípodas, debajo de la Tierra. El flujo en las antípodas fue la mitad de lo esperado. Este déficit puede ser explicado por el fenómeno de oscilación de neutrinos, donde algunos ν_μ podrían transformarse en ν_τ al atravesar los más de 12 mil kilómetros a través de la Tierra.

1.1.2. Neutrinos Solares

El sol es una fuente muy poderosa de neutrinos electrónicos, con energías del orden de 1MeV, producidos en reacciones de fusión termonuclear en el núcleo del Sol [11]. Debido a que las interacciones de neutrinos con la materia son extremadamente débiles, prácticamente todos los neutrinos producidos en el núcleo solar atraviesan el Sol hasta el espacio.

El flujo de neutrinos solares que llega a la Tierra es enorme pero su detección es bastante dificultosa y requiere grandes detectores debido a las pequeñas secciones eficaces de interacción de los neutrinos. Estos detectores son colocados bajo Tierra para estar protegidos por las rocas de los rayos cósmicos, cuyas interacciones superarían altamente en número y dominarían frente a las interacciones de los neutrinos solares.

En la década de los 80-90s diferentes experimentos reportaron flujos de ν_e solares incidentes en la Tierra inferiores, un tercio, a lo esperado por el modelo solar aceptado [12].

1.1.3. Oscilaciones de Neutrinos

El enigma debido a estas diferencias se mantuvo por casi tres décadas hasta que el experimento SNO, Sudbury Neutrino Observatory [13], dio una contribución importante en el 2001.

El SNO era capaz de detectar neutrinos de las tres familias, a diferencia de los experimentos anteriores. La explicación podría estar en que los neutrinos pueden oscilar. En el Sol solo se producen ν_e pero en su viaje a la Tierra una parte de ellos son transformados en ν_μ y ν_τ

1.2. Neutrinos en el Modelo Estándar

En los años sesenta, S. Weinberg y A. Salam, propusieron el modelo electrodébil, que enlaza las interacciones electromagnéticas y las débiles y postulando cuatro bosones mediadores sin masa, ordenados en un triplete isovectorial del grupo $SU(2)$ y un singlete isoescalar ("hipercarga débil" Y) perteneciente al grupo $U(1)$. El modelo es referido al grupo $SU(2)_L \otimes U(1)_Y$.

El proceso de quiebra espontánea de simetría $SU(2)_L \otimes U(1)_Y$ permite a los bosones adquirir masa al interactuar con un campo escalar (bosón de Higgs) que permeabiliza todo el espacio. Los bosones masivos son denotados por W_μ^\pm y Z_μ^0 , mientras que el fotón A_μ permanece sin masa [14].

En mecánica cuántica relativista, fermiones de spin $\frac{1}{2}$ de masa m , están descritas por la ecuación de Dirac (utilizando convenciones de Einstein y considerando $\hbar = c = 1$) [15]:

$$\left(i\gamma^u \frac{\partial}{\partial x^u} - m \right) \psi = 0 \quad (1.3)$$

Tal que ψ denota un espinor de cuatro componentes y γ^u son las matrices 4×4 denotadas por ⁴:

$$\gamma^0 = \begin{pmatrix} 0 & \sigma^0 \\ \sigma^0 & 0 \end{pmatrix} \quad \gamma^i = \begin{pmatrix} 0 & \sigma^i \\ -\sigma^i & 0 \end{pmatrix} \quad (1.4)$$

en donde σ_i , son las matrices 2×2 de Pauli [9]:

Las cuatro componentes de ψ , corresponden a partículas y antipartículas con dos posibles proyecciones $J_Z = \pm \frac{1}{2}$, equivalente a las dos helicidades $\mathcal{H} = \frac{\mathbf{s} \cdot \mathbf{p}}{|\mathbf{p}|} = \pm 1$ ⁵. Los neutrinos son leptones de spin $\frac{1}{2}$ como otros fermiones, sin embargo, es un hecho experimental de que sólo los neutrinos de quiralidad izquierda ($\mathcal{H} = -1$) y antineutrinos de quiralidad derecha ($\mathcal{H} = +1$) son observados.[15].

Por lo tanto, la descripción de un espinor de dos componentes (espinores de Weyl), deberían, en principio, ser suficientes. En una teoría con 4 componentes, esto se obtiene con la ayuda de los operadores $P_{L,R} = \frac{1}{2}(1 \mp \gamma^5)$ [9].

⁴Ésta es la representación quirál (o de Weyl) para γ

⁵Aquí, \mathbf{s} y \mathbf{p} son el spin y el momentum de la partícula

$$\psi = (P_L + P_R)\psi = \frac{1}{2}(1 - \gamma^5)\psi + \frac{1}{2}(1 + \gamma^5)\psi = \psi_L + \psi_R \quad (1.5)$$

$$\text{en donde } \gamma^5 = i\gamma^0\gamma^1\gamma^2\gamma^3 = \begin{pmatrix} -\sigma^0 & 0 \\ 0 & \sigma^0 \end{pmatrix}$$

Los neutrinos interactúan solo en forma débil, se presumen sin masa y los estados de neutrinos de mano derecha no son incluidos en el Modelo Estándar. Se conocen 12 partículas fundamentales fermiónicas (6 leptones y 6 quarks) que pueden escribirse como 6 dobletes para campos quirales de quiralidad izquierda y 9 singletes para los campos de quiralidad derecha en la forma:

$$\begin{pmatrix} e \\ \nu_e \end{pmatrix}_L \quad \begin{pmatrix} \mu \\ \nu_\mu \end{pmatrix}_L \quad \begin{pmatrix} \tau \\ \nu_\tau \end{pmatrix}_L \quad (1.6)$$

$e_R \quad \mu_R \quad \tau_R$

$$\begin{pmatrix} u \\ d' \end{pmatrix}_L \quad \begin{pmatrix} c \\ s' \end{pmatrix}_L \quad \begin{pmatrix} t \\ b' \end{pmatrix}_L \quad (1.7)$$

$u_R \quad d_R \quad s_R \quad c_R \quad b_R \quad t_R$

En las interacciones electromagnéticas el fotón se acopla tanto a estados de quiralidad izquierda como derecha. En interacciones débiles CC (corriente cargada - via W^\pm) de leptones se acopla con estados de quiralidad izquierda. Para interacciones débiles NC (corriente neutra - via Z^0) con neutrinos, solo los de quiralidad izquierda están involucrados. En interacciones débiles CC de quarks, adicionalmente se incluye la matriz de mezcla de Cabibbo-Kobayashi-Maskawa, U^{CKM} , que describe la relación matemática entre los estados de quarks de masa definida (d, s, b) y los de sabor definido (d', s', b'). En el Modelo Estándar no hay una matriz equivalente que relacione los estados observables (ν_e, ν_μ, ν_τ) a los estados de masa (ν_1, ν_2, ν_3).

La lagrangiana del modelo Glashow-Weinberg-Salam utilizando corrientes electromagnéticas, cargadas y neutrales es [15],[16]:

$$\mathcal{L} = -e\mathcal{J}_{EM}^\mu A_\mu - \frac{g}{\cos(\theta_W)}\mathcal{J}_{NC}^\mu Z_\mu - \frac{g}{\sqrt{2}}\left(J_{CC}^{\mu\dagger}W_\mu^+ + J_{CC}^\mu W_\mu^-\right) \quad (1.8)$$

De donde, \mathcal{J}_{EM}^μ es la corriente electromagnética, \mathcal{J}_{NC}^μ la corriente neutra débil, y $J^{\mu\dagger}, J^\mu$ la corriente cargada débil y sus acoplados: el campo fotónico A_μ , el campo del bosón W_μ^\pm y el campo del bosón neutrino Z_μ , g es la constante de acoplamiento de un leptón asociada al bosón W_μ^\pm , relacionada a la constante de Fermi G por la expresión:

$$\frac{g^2}{8M_w^2} = \frac{G}{\sqrt{2}}$$

Las corrientes leptónicas (en representación quiral) están dadas por [15],[16]:

$$\begin{aligned} \mathcal{J}_{EM}^\mu &= \bar{l}_L \gamma^\mu l_L + \bar{l}_R \gamma^\mu l_R = \bar{l} \gamma^\mu l \\ \mathcal{J}_{NC}^\mu &= \frac{1}{2} \bar{\nu}_L \gamma^\mu \nu_L - \frac{1}{2} \bar{l}_L \gamma^\mu l_L - \sin^2 \theta_W \mathcal{J}_{EM}^\mu \\ J_{CC}^{\mu\dagger} &= \bar{\nu}_L \gamma^\mu l_L \\ J_{CC}^\mu &= \bar{l}_L \gamma^\mu \nu_L \end{aligned} \quad (1.9)$$

O en representación de Dirac [17]:

$$\begin{aligned} \mathcal{J}_{EM}^\mu &= \bar{l}_L \gamma^\mu l_L + \bar{l}_R \gamma^\mu l_R = \bar{l} \gamma^\mu l \\ \mathcal{J}_{NC}^\mu &= \frac{1}{2} \bar{\nu}_l \gamma^\mu \left(\frac{1-\gamma^5}{2} \right) \nu_l - \frac{1}{2} (1 - 2 \sin^2 \theta_W) \bar{l} \gamma^\mu \left(\frac{1-\gamma^5}{2} \right) l + \sin^2 \theta_W \gamma^\mu \left(\frac{1-\gamma^5}{2} \right) l \\ J_{CC}^{\mu\dagger} &= \bar{\nu}_l \gamma^\mu \left(\frac{1-\gamma^5}{2} \right) l \\ J_{CC}^\mu &= \bar{l} \gamma^\mu \left(\frac{1-\gamma^5}{2} \right) \nu_l \end{aligned} \quad (1.10)$$

Siendo θ_W el ángulo de Weinberg, tal que: $\sin \theta_W = \frac{e}{g}$

1.3. Neutrinos masivos

En la formulación del Modelo Estándar[18] las partículas no tienen masa (esto garantiza la invarianza de gauge bajo transformaciones $SU(2)$ ó $U(1)$), sin embargo, es un hecho experimental que tanto las partículas como los bosones mediadores W^\pm , Z^0 tienen masa (lo cual hace que la fuerza débil sea de muy corto alcance⁶, en comparación al alcance infinito de la electromagnética regidas por el fotón, el cual no es masivo).

En el modelo estándar, la adición de masas se consigue mediante el quiebre de simetría via Mecanismo de Higgs. Para romper la simetría de $SU(2)$, se introduce un doblete de campos escalares complejos, uno cargado y uno neutro:

$$\phi = \begin{pmatrix} \phi^\dagger \\ \phi^0 \end{pmatrix} \quad (1.11)$$

lo cual nos lleva a adicionar el llamado Acoplamiento de Yukawa a la lagrangiana del Modelo

⁶El rango para la fuerza débil es de $10^{-18}m$

Standard para cada familia leptónica:

$$\mathcal{L}_{Yuk} = -c_l [\bar{\nu}_L \phi^\dagger l_R + \bar{l}_L \phi^0 l_R] + h.c \quad (1.12)$$

en donde c_l es una constante arbitraria de acoplamiento y $h.c$ la conjugada hermitiana.

Después del quiebre espontáneo de simetría, los valores para el campo ϕ vienen de una configuración particular seleccionada llamada *estado vacío*, motivados por el hecho de que dicho estado es electricamente neutro, en donde los valores esperados al vacío del campo de Higgs son $\langle \phi^\dagger \rangle = 0$ y $\langle \phi^0 \rangle = v/\sqrt{2}$, donde $v \simeq 246 GeV$, dejando por tanto a los neutrinos sin masa, y a los leptones cargados e, μ, τ un término de masa, proveniente de:

$$\mathcal{L}_D = -(m_D^l) \bar{l}_L l_R + h.c \quad (1.13)$$

siendo $m_D^l = c_l v/\sqrt{2}$, cuya constante c_l de acoplamiento se obtiene experimentalmente.

Sin embargo, resultados experimentales indican que los neutrinos tendrían masa, para lo cual se introduce el término de quiralidad derecha del neutrino, obteniendo una lagrangiana similar a la obtenida para los leptones cargados:

$$\mathcal{L}_D^{\nu_l}(x) = -\nu_{\alpha L}(x) m_{\alpha\beta} \nu_{\beta R} + h.c \quad (1.14)$$

en donde $m_{\alpha\beta}$ es una matriz compleja, que puede ser puesta en forma diagonal real con la ayuda de dos matrices unitarias:

$$m_{\alpha\beta} = U_{\alpha i}^{L*} m_i U_{\beta i}^R \quad (1.15)$$

aquí, m_i es una masa positiva real que puede tomar hasta tres valores, U^L, U^R son las matrices unitarias, y si definimos:

$$\nu_{iL}(x) = U_{i\alpha}^{L*} \nu_{\alpha L}(x) \quad (1.16)$$

$$\nu_{iR}(x) = U_{i\alpha}^{R*} \nu_{\alpha R}(x) \quad (1.17)$$

y reemplazamos en 1.14, obtenemos:

$$\mathcal{L}_D^{\nu_l}(x) = -m_i \left(\nu_{iL}^\dagger \nu_{iR} + \nu_{iR}^\dagger \nu_{iL} \right) \quad (1.18)$$

La cual se parece al término de masa de de la densidad lagrangiana estándar utilizando ψ_L y ψ_R .

Sin embargo, debido a que los neutrinos son partículas neutras, es posible definirlos de manera diferente, considerando al neutrino como su propia antipartícula (lo cual es conocido como la *Condición de Majorana* [19]):

$$\bar{\nu} = \nu$$

La densidad lagrangiana del término de masa de Majorana está dado por [9]:

$$\mathcal{L}_{\mathcal{M}}(x) = -\frac{1}{2}\nu_{\alpha}^T(-i\sigma^2)\nu_{\beta}m_{\alpha\beta} + h.c \quad (1.19)$$

en donde α, β adquieren los valores de los tres tipos de neutrinos e, μ, τ , y además $\nu_{\alpha}, \nu_{\beta}$ son neutrinos de quiralidad izquierda (los subscript L han sido omitidos para mayor claridad) y $m_{\alpha\beta}$ es una matriz compleja arbitraria. Si consideramos $m_{\alpha\beta} = m_{\beta\alpha}$, podemos escribir:

$$m_{\alpha\beta} = U_{\alpha i}m_i U_{\beta i} \quad (1.20)$$

en donde m_i son tres masas positivas, y podemos definir:

$$\nu_i(x) = U_{\alpha i}\nu_{\alpha}(x) \quad (1.21)$$

de donde la ecuación 1.19 toma la forma:

$$\mathcal{L}_{\mathcal{M}}(x) = -\frac{1}{2}m_i\nu_i^T(-i\sigma^2)\nu_i + h.c \quad (1.22)$$

siendo:

$$\nu_{\alpha}(x) = U_{\alpha i}^*\nu_i(x) \quad (1.23)$$

1.4. Interacciones Neutrino-Nucleón

1.4.1. Dispersión Quasi-elástica

La dispersión quasi-elástica domina la tasa de interacciones $\nu - N$ para $E \leq 2\text{GeV}$. Mediciones de alta precisión de la sección eficaz para éstas reacciones, incluyendo su dependencia de energía y la variación con distintos blancos es esencial para los presentes y futuros experimentos de oscilaciones de neutrinos.

Reacciones quasi-elásticas de Corriente Cargada (CC)

Las interacciones quasi-elásticas de corriente cargada de neutrinos y anti-neutrinos con los nucleones son:

$$\nu_l + n \rightarrow p + l^- \quad (1.24)$$

$$\bar{\nu}_l + p \rightarrow n + l^+ \quad (1.25)$$

siendo $l = e, \mu, \tau$. En la práctica, solo haces de ν_e y ν_μ están disponibles en laboratorios, los haces de ν_τ son generados en ambientes físicos y por interacciones de altas energías de rayos cósmicos con la superficie terrestre.

La ecuación 1.25 en el que $l = e$ es a veces llamado también “proceso beta inverso”, y ha sido utilizado en experimentos históricos como el Cowan y Reines, donde los neutrinos fueron observados por primera vez. Considerando el caso de altas energías, en el que la diferencia de masas del neutrón y protón se puede despreciar, el proceso de dispersión $\nu_l - n$ en (1.24) puede ser obtenido, pasando desde el proceso de decaimiento del neutrón $n \rightarrow p + e^- + \bar{\nu}_e$.

Las muestras de eventos quasi-elásticos en MINER ν A brindarán una precisión sin precedentes sobre la respuesta Q^2 (momento transferido durante la colisión al cuadrado) de las *weak nucleon currents* (interacciones de neutrinos con el protón o neutrón).

1.4.2. Dispersión Profundamente Inelástica

La característica única de corrientes débiles para “probar” sabores de quark específicos ayudan en los estudios de PDFs (parton distribution functions). Los ν interactúan con d, s, \bar{u} y \bar{c} , mientras que los $\bar{\nu}$ interactúan con u, c, \bar{d} y \bar{s} .

Las principales interacciones son:

reacciones CC :

$$\nu_l + N \rightarrow l^- + X$$

$$\bar{\nu}_l + N \rightarrow l^+ + X$$

reacciones NC :

$$\nu_l + N \rightarrow \nu_l + X$$

$$\bar{\nu}_l + N \rightarrow \bar{\nu}_l + X$$

done, $N = p, n$ y X denota cualquier serie final de hadrones. Las mediciones de alta precisión de MINER ν A de la estructura partónica de los nucleones usando neutrinos, complementará los estudios de otros laboratorios utilizando sondas electromagnéticas.

1.4.3. Producción Coherente, Resonante de Piones

Los distintos blancos nucleares de MINER ν A, así como su granularidad fina, fuerte capacidad de reconocimiento de patrones y buena calorimetría electromagnética harán posible realizar estudios de producción coherente neutrino-núcleo de corriente neutra y cargada con una precisión nunca antes alcanzada.

La producción resonante en interacciones de neutrinos es una parte significativa de la sección eficaz total para el rango de algunos GeV, sin embargo ésta es la parte menos certera en el marco de secciones eficaces.

Las principales interacciones que generan producción resonante son:

$$\nu N \rightarrow \nu R \quad (1.26)$$

$$\nu N \rightarrow l^- R \quad (1.27)$$

Experimentos de estadísticas de desapariciones de neutrinos muónicos son particularmente sensibles a los estados finales hadrónicos, particularmente en número, carga y cinemática de los estados finales de piones. La falta de conocimiento de éstos estados finales contribuye a una incertidumbre en la medida de la energía total hadrónica y por tanto, en la estimación de la energía de los neutrinos incidentes y el parámetro Δm^2 [37], que relaciona las diferencias entre los estados de masa de neutrinos. Los estudios sistemáticos y detallados que MINER ν A puede brindar son importantes para mejorar éstos aspectos.

Descripción del experimento MINERvA

2.1. Descripción General del Experimento

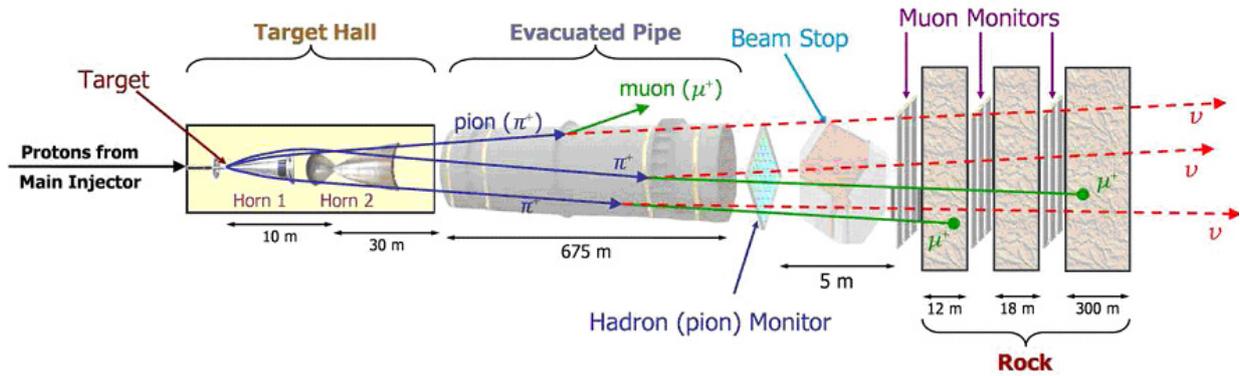
MINERvA¹ es un experimento de scattering de neutrinos [21], el cual utiliza el haz NuMI en Fermilab. Minerva apunta a medir bajas energías de interacciones de neutrinos, en soporte de los experimentos de oscilaciones de neutrinos como MINOS, además de estudiar la fuerte dinámica de los nucleones y núcleos que afectan éstas interacciones[28].

2.2. La fuente de neutrinos (El haz NuMI)

NuMI[22] (Neutrinos at the Main Injector), es un haz intenso de ν_μ ubicado en FERMILAB, destinado a servir a distintos experimentos de estudios de neutrinos. El experimento MINERvA, estará ubicado en el Hall NuMI (donde se localiza también el MINOS Near Detector [35]), a 1km de distancia del *NuMI target*, para la realización de medidas de secciones eficaces de neutrinos.

NuMI es un haz terciario que resulta del decaimiento de kaones y piones secundarios producidos en el *NuMI target* (ver figura 2.1(b)) por interacciones de protones de 120 GeV, extraídos del *Main Injector accelerator* y desviados en dirección al ambiente donde se encuentra, entre otros, el detector de MINERvA.[34]. Una descripción detallada del diseño del haz, puede ser encontrada en [33].

¹El nombre MINERvA viene de: Main INjector Experimenta: v-A



(a) Componentes principales del haz NuMI



(b) Instalación del detector MINERvA en el NuMI Hall

Figura 2.1: Diseño del NuMIBeam y el Hall NuMI

2.3. Descripción General del Detector

Exceptuando los Veto Walls, el detector MINERvA está segmentado transversalmente en un *detector interior* (ID) con planos de tiras sólidas y uno exterior (OD)². Para la construcción y por conveniencia, un plano de MINERvA incorpora ambos, el detector interior y el exterior OD (el cual sirve además como soporte de estructura).

Dos planos de centelleadores son montados en un marco, llamado *módulo*. Existen tres diferentes tipos de orientación para las tiras de centelleadores para el detector interior, cada uno con una diferencia de 60° y etiquetados como X, U y V. Las diferentes orientaciones en las tiras nos permiten hacer una reconstrucción tridimensional de las trayectorias. Las tiras de centelleadores se extienden a lo largo del exágono y tienen longitudes de entre 205 y 400cm.

²ID: Inner Detector, OD: Outer Detector

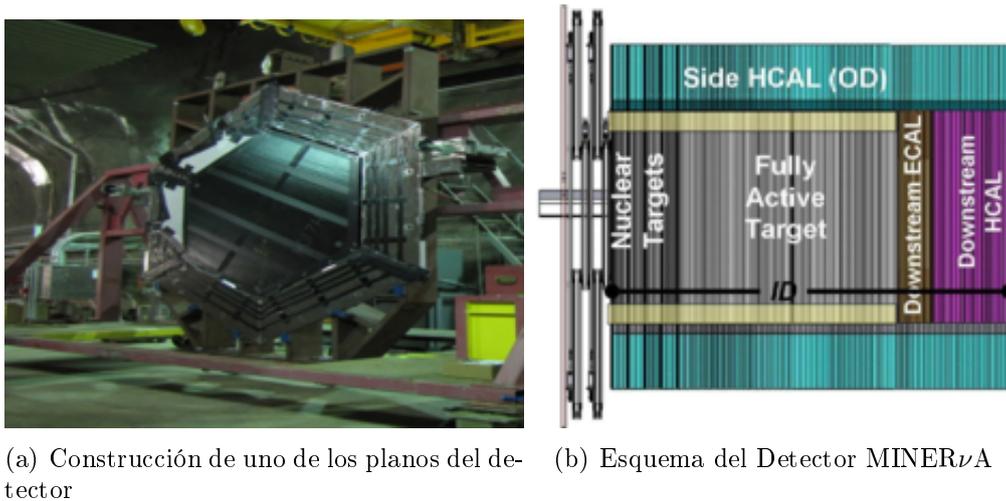


Figura 2.2: Detector Minerva

Después de los Veto Walls, tenemos los blancos nucleares, hechos de hojas de plomo, carbón y hierro, los cuales son intercalados con planos centelleadores para reconstrucción de vértices.

El centro del detector es el blanco activo del detector interior, el cual se usa para la mayoría de los análisis de MINERνA. Se construyen detectores calorimétricos en la region central del detector, insertando material absorbente entre los planos adyacentes. Placas de aleación de plomo-acero de 0.2 centímetros de espesor son colocadas entre las capas de centelleador y en la parte frontal de cada módulo para ser usados como calorímetros electromagnéticos³.

Después de dicho blanco activo, se encuentran los calorímetros electromagnéticos, los cuales cubren el área del blanco activo, pero no el detector exterior. La final granulidad de éstos calorímetros permiten excelentes resoluciones de energía de fotones y electrones.

El detector interior es rodeado por los marcos de material absorbente y tiras de centelleador que componen el detector exterior OD, el cual consiste de seis "torres" (un sexto de hexágono).

La parte final del detector está compuesta de calorímetros hadrónicos, con hierro absorbente de 2.54cm por plano centelleador.

2.4. Los tubos fotomultiplicadores

Ver sección 3.1

³Los calorímetros electromagnéticos son detectores que miden la energía de fotones y electrones

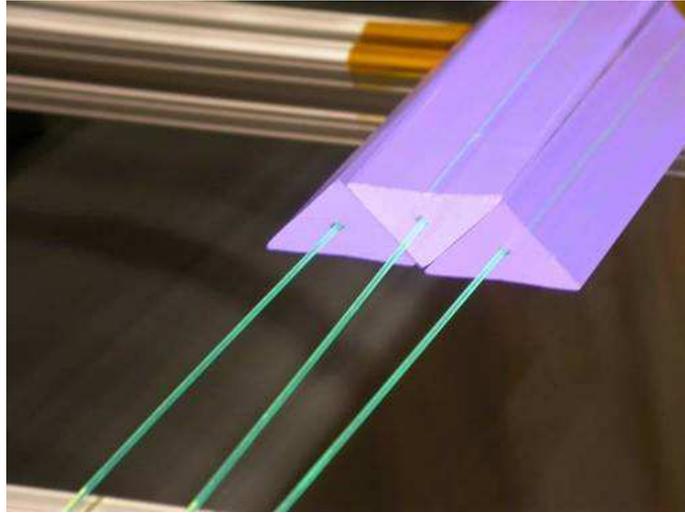


Figura 2.3: Tira de Centelleador trinangulos para el detector interior

2.5. Tiras de Centelleador

Las tiras de centelleador están hechas de poliestireno extruido dopado con compuestos fluorescentes de emisión azul, una capa exterior co-extruida de TiO_2 para reflexividad y un orificio en el centro para las fibras WLS (fibras de longitud de onda cambiante)⁴, Éstas tiras son preparadas con una sección eficaz triangular (3.3 cm de base y 1.7 cm de altura) para el detector interior (ID) (ver figura 2.5) y rectangular para el detector exterior (1.9cm de base y 1.5cm de altura).

2.6. Calorímetros y Blancos

El experimento MINER ν A , requiere hacer mediciones de partículas cargadas (p , $\pi^{+/-}$, $K^{+/-}$, $\mu^{+/-}$) y neutrales (π^0 , K^0 con energías de algunos GeV), para lo cual lo ideal sería crear un detector completamente activo conteniendo el 100 % de energía. Sin embargo los costos y problemas de localización, impiden hacer un detector de dichas características y con el tamaño apropiado, por lo cual se eligió un diseño MINER ν A compuesto por un detector totalmente activo en el centro, rodeado de calorímetros electromagnéticos componiendo el detector interior, y calorímetros hadrónicos en el exterior.

Otra necesidad de MINER ν A , es la de estudiar las interacciones de los neutrinos con la

⁴Las fibras WLS son de 1.2mm de diametro, proveídas por Kuraray [36] la cual ha sido usada por diferentes experimentos de altas energías como CDF, MINOS [20], CMS, etc.

materia ($\nu - A$). Debido a que la parte principal del detector es material centelleador, sirve como blanco de carbón, y además, existen blancos de hierro, plomo y carbon, que servirán como blancos nucleares y como calorímetros.

El rol de la calorimetría, es que las cascadas de energía electromagnética y hadrónica originadas en la región de trayectorias debe ser completamente contenida, siendo más crítica ésta función para los calorímetros de la parte final del detector, puesto que para la detección de neutrinos, la producción de las partículas de interés está fuertemente concentrada en ésa región, teniendo dichas partículas, las energías más altas.

2.6.1. Calorímetros Electromagnéticos

Los fotones de altas energías se detectan a través de los procesos de producción de pares y bremsstrahlung, conllevando a cascadas de $e^{+/-}$ y γ . Debido a que la sección eficaz de la producción de pares, en interacciones con la materia, es proporcional al Z^2 , generalmente se usan hojas de plomo para producir cascadas de longitudes razonables. La longitud característica de las cascadas varía con la energía, pero para los fotones, el rango es de hasta algunos GeV. El 99 % de la energía será contenida en 4cm de plomo (alrededor de 7 longitudes de radiación).

Los calorímetros laterales (que rodean a la parte central hecha de tiras triangulares de centelleador en el detector interior), están hechos de hojas trapezoidales de plomo de 2mm de espesor, intercalados con capas de centelleador (formado por las tiras triangulares comentadas en la sección 2.5). Las hojas se extienden 15cm en el área activa. Los fotones entrantes a éstos calorímetros serán completamente contenidos para ángulos menores a 25° con respecto al eje del haz de neutrinos. A mayores ángulos, la cascada no será completamente contenida, sino que penetrará en los calorímetros hadrónicos de la parte exterior del detector, en donde será completamente contenido.

Los calorímetros de la parte final del detector (*DownStream ECals* en la figura 2.2(b)) están formadas por 20 capas de plomo, de 2mm de espesor cada una, similar al de los calorímetros laterales, intercalados con capas de centelleador. Se espera una resolución de energía, $\frac{\Delta E}{E}$, de aproximadamente $6\%/\sqrt{E}$ (estando E en GeV).

Las hojas de plomo, hierro y carbón que sirven como blancos nucleares (ver figura 2.2(b)), tienen también una función como calorímetros, pero su eficiencia no es tan buena como los de la parte del final del detector, debido a que éstas hojas son más gruesas. El arreglo de blancos implica que el número de longitudes de radiación que la cascada ve antes de escapar de los

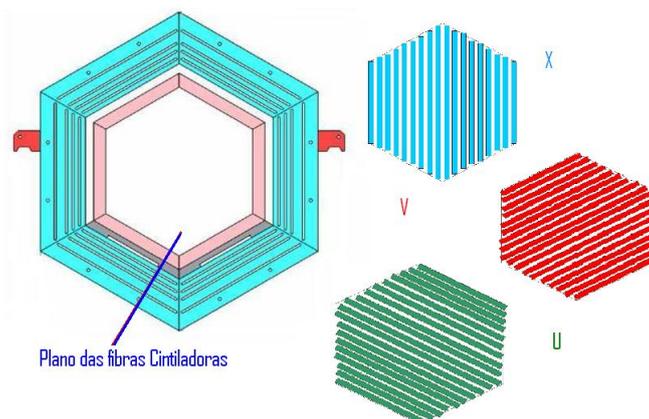


Figura 2.4: Vista de un módulo activo del detector, junto con los 3 planos de centelleador X,U,V rotados $\pm 60^\circ$ respecto al otro

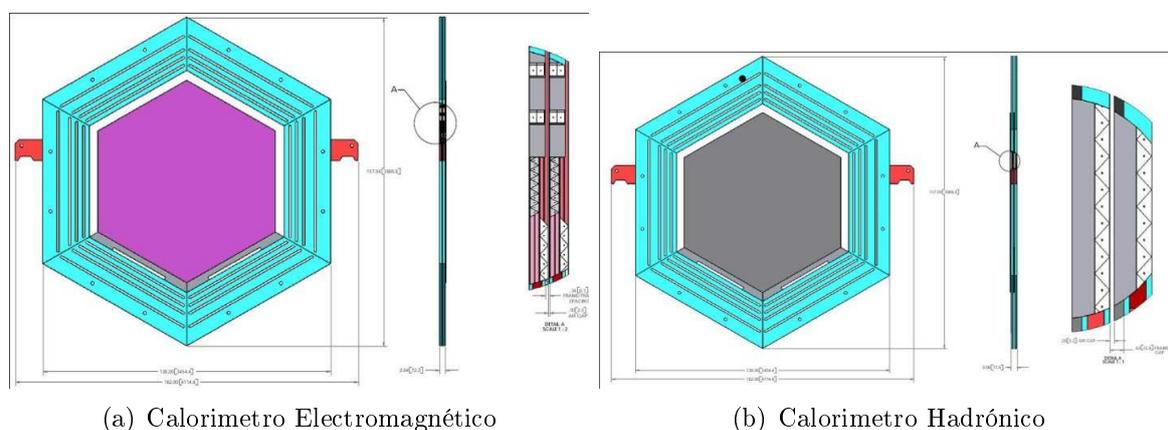


Figura 2.5: Calorímetros en Minerva

blancos varía entre 5 y 10. Sin embargo, ya que los rangos de energía de los fotones entrantes serán generalmente mucho menores, las cascadas que se inicien en la región central activa del detector, serán completamente contenidas.

2.6.2. Calorímetros Hadrónicos

Los calorímetros hadrónicos de la parte final del detector consisten de 20 capas de hierro, de 2.54 cm de espesor cada una, intercalados con capas de centelleador. El espesor combinado de 4cm de plomo y 50cm de fierro detendrán muones de hasta 600MeV y protones de hasta 800MeV. La longitud de una interacción nuclear es de 16cm para el hierro, por lo que energías más altas de protones (o piones) serán generalmente, también detenidos.

El calorímetro lateral está formado por láminas de hierro de 55.9cm de espesor, con 5 ranuras, de 2.5cm de ancho cada una, cubiertas con material centelleador. El espesor total de hierro es de 43.4cm, suficiente para detener la pérdidas por ionización de protones de hasta 750 MeV a 90° y protones de cerca de 1 GeV entrantes a un ángulo de 30°.

Basado en estudios de MINOS, la resolución $\frac{\Delta E}{E}$ de los calorímetros esperada es de alrededor de $50\%/\sqrt{E}$ para energías de hadrones de alrededor de 1 GeV, y para partículas de energía más bajas, dependiendo de la energía, de un 50 % o menos.

Respecto a la calorimetría de la parte del comienzo del detector, de igual manera que con los calorímetros electromagnéticos, éstos sirven como blancos nucleares, siendo su diseño menos eficiente que el de los calorímetros de la parte final. El grosor de la calorimetría en la parte inicial del detector es suficiente para detener protones de al menos 300MeV originándose en la región central activa.

2.6.3. Blancos Nucleares

Los blancos nucleares consistirán de carbón, hierro y plomo. El plomo es seleccionado por ser el material de mayor masa nuclear que es fácilmente obtenible, el hierro debido a que es relativamente barato y sirve de blanco de masa media y es un material absorbente usado en muchos detectores de neutrinos, como MINOS.

La resolución espacial intrínseca del detector es del orden de 1cm, por lo que blancos más delgados que esto serían ineficientes. MINOS usó láminas de hierro de 2.5cm, por lo que láminas más gruesas no permitirían una mejora significativa para el estudio de los espectros de partículas de bajas energías, el cual es uno de los objetivos de MINERVA .

La masa total de hierro es de 685kg y 855kg para el plomo. El número esperado de eventos para reacciones de corriente cargada (CC) esperado es de alrededor de 2 millones para el hierro, 2.5 millones para el plomo y 400K para el carbono.

El diseño decidido por el equipo MINERVA , considerando "FF como un marco XU o XV, y "FF como un arreglo XUXV, ha sido el siguiente:

FF Pb/Fe [1] FFFF Pb/Fe [2] FFFF Pb/Fe/C [3] FFFF Pb[4] FFFF Pb/Fe [5]

Los blancos [1] y [2] serán de 2.5cm de grosor para Plomo y Hierro en un plano, cubriendo el hierro un área de 60 %, lo que da alrededor de 230kg de plomo y hierro en cada blanco en un radio de 80cm. El blanco [3] tendrá un área de cobertura de 50 % para el Carbono, 30 % Hierro

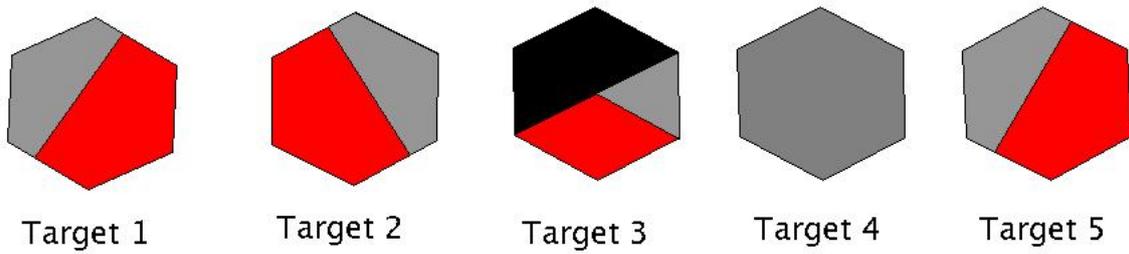


Figura 2.6: Blancos Nucleares: Rojo=Fe, Gris=Pb, Negro=C

y 20 % Plomo, dando 140kg de Carbono y 110kg de plomo y fierro cada uno. Los blancos de plomo y fierro serán de 2.5cm de grosor y el de Carbono de 7.5cm. El blanco [4] será plomo puro de 0.75cm de grosor, con una masa de 170kg. El blanco [5] será de 1.25cm de grosor de plomo y fierro, nuevamente 60 % del área de cobertura será de 60 % fierro y 40 % plomo, con una masa de 115kg cada una.

Los blancos [1] y [2] tendrán el plomo y fierro rotados con respecto al otro para permitir chequeos por diferencias de detección, el blanco [3] permitirá estudios detallados de la dependencia de A con las interacciones. El blanco [4] es de plomo puro, para asegurarse de que cualquier fotón producido en los blancos comience a hacer cascada, la longitud de radiación del plomo es de 1.5, lo suficiente para hacer que la cascada comience, pero no para contenerla completamente. El blanco [5] permitirá el estudio de multiplicidad y distribuciones de partículas de bajas energías con buena resolución de energía.

2.7. Electrónica, DAQ (Adquisición de Datos)

Los requerimientos de la electrónica de MINERVA está motivada por los siguientes objetivos:

- Resolución espacial fina, aprovechando la compartición de luz entre barras centelleadoras vecinas.
- Identificación de π^\pm , K^\pm y p , utilizando información de dE/dx .
- Un reconocimiento de patrones eficiente, utilizando cronometraje electrónico para identificar dirección de trayectorias y separar interacciones que ocurran durante un mismo ciclo (spill).
- Identificación de partículas *strange* y decaimiento de muones empleando técnicas de coincidencia

- Tiempo muerto de lectura insignificante entre un ciclo.

Los sistemas de electrónica y adquisición de datos (DAQ) son necesarios, no solamente para el detector final MINERVA, sino también para una serie de pruebas y subsistemas de medición previos. La tasa de datos promedio esperada por MINERVA es de 100kB/s, y se utilizarán un total de 473 tubos fotomultiplicadores y tarjetas de lectura electrónica en el detector.

2.7.1. Tarjetas Electrónicas de Lectura (FEB)

Éstas tarjetas se encargan de digitalizar información de *timing* (información del tiempo de lectura al final de cada ciclo o spill) y señales de pulso (carga), además de proveer de alto voltaje a los tubos fotomultiplicadores.

El diseño de éstas tarjetas está basado en los Trip-T ASIC (circuito integrado para aplicaciones específicas) diseñadas para el experimento D0 por Abder Mekkaoui del *Fermilab ASIC group*, y son controlados mediante una tarjeta FPGA. Cada uno de éstos chips contiene 32 canales para la digitalización y 16 para discriminación y *timing*. [39].

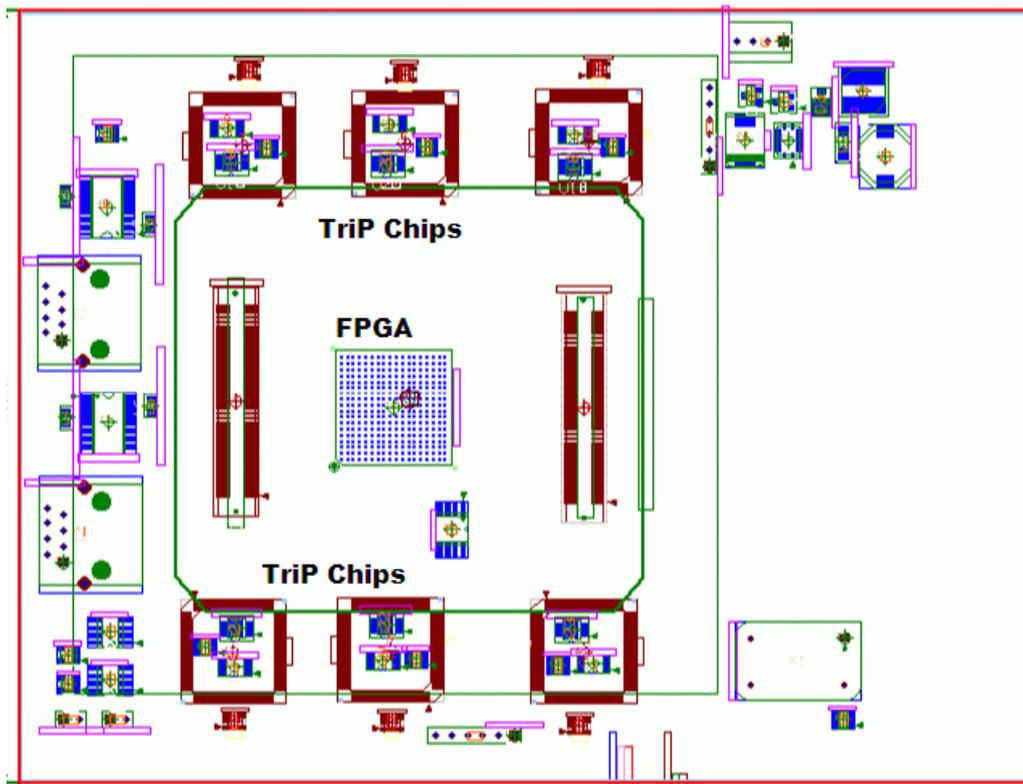


Figura 2.7: Esquema electrónico de los FEB

2.7.2. Descripción del Software para la Adquisición de Datos

El software elegido para la infraestructura cliente-servidor en la adquisición de datos está basado en el paquete GAUCHO a través del framework GAUDI, desarrollado por el *LHCb* y el *LHC Joint Controls Project*.

La alta predictibilidad del haz NuMI nos ahorra la necesidad del diseño de un trigger complicado, en vez de eso, simplemente se activa la electrónica del detector justo antes de la llegada del haz y se guarda toda la información al final de cada ciclo (spill).

El sistema de control (Slow-Control System) también es relativamente simple, ya que cada PMT tiene su propia fuente de alto tensión, definiendo el voltaje a través de las tarjetas electrónicas de lectura conectadas a cada tubo, permitiendo las lecturas de éste valor, así como temperatura y otros parámetros implementados para la monitorización (discriminadores, etc).

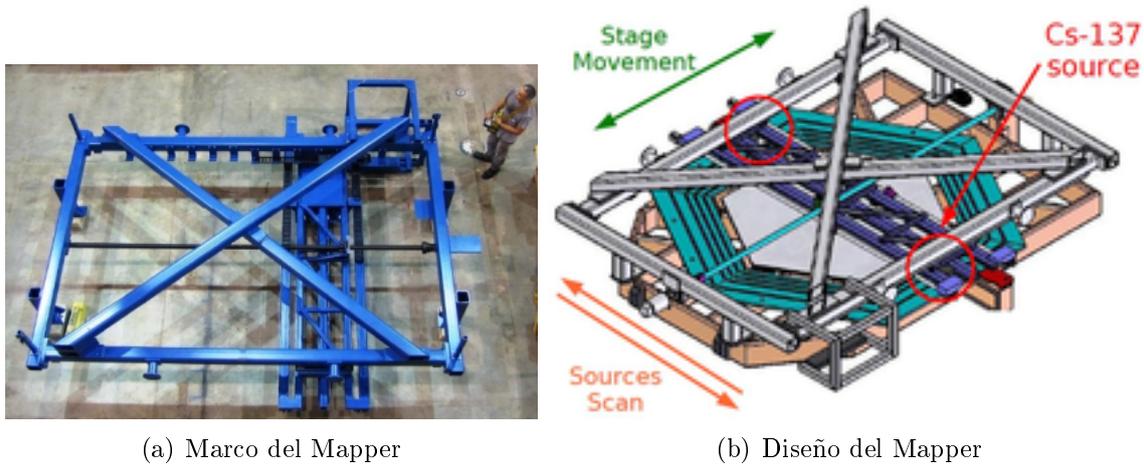
2.8. Pruebas Experimentales programadas en el Experimento

Existen varios sub-proyectos en el experimento MINER ν A que se realizan mientras se construye el detector completo, entre ellos el Mapeo de Centelleadores, el MTest Detector y el detector prototipo (Tracking Prototype)

2.8.1. Mapeo de Centelleadores

El propósito del mapeo es el de estudiar la respuesta local de los centelleadores. En el *Mapper* se barren los centelleadores de todos los módulos de MINER ν A en puntos pre-determinados con dos fuentes radioactivas Cs-137 y se lee su respuesta (ver figura 2.8.1). Dicha respuesta de los centelleadores como función de la posición se graba en un archivo.

El *Mapper* nos permite identificar irregularidades en los centelleadores que puedan afectar el rendimiento del detector. La respuesta de éstos centelleadores es leída mediante los M64 PMTs y las tarjetas de lectura electrónica mencionadas en las secciones 2.7 y 3.1.



(a) Marco del Mapper

(b) Diseño del Mapper

Figura 2.8: Diseño del Mapeo de Centelleadores

2.8.2. Detector de Prueba (TestBeam Detector)

El detector de prueba de MINER ν A (TestBeam Detector), será una versión pequeña de MINER ν A hecho de 40 planos centelleadores de $1.2 \times 1.2 \times 1.8 \text{ m}^2$ (ver figura 2.8.2). Al igual que con el detector completo, los planos estarán formados por barras de centelleadoras triangulares de $33 \text{ mm} \times 17 \text{ mm}$. Cada plano contendrá 76 barras de 1.2m. Utilizará los mismos PMTs y fibras que el detector MINER ν A completo.

El objetivo principal de éste proyecto, es el de medir la respuesta del detector MINER ν A a partículas individuales, en especial protones y piones. Ésto incluye información de la energía total de cascadas hadrónicas, dE/dx a lo largo de las trayectorias, resolución de energía para éstas cantidades, e identificación de partículas. La información obtenida por éste detector de prueba será útil para confirmar o afinar la simulación del detector (basado en Geant4 [40]) y para la calibración de los calorímetros hadrónicos. Será también importante para el desarrollo de patrones de reconocimiento y para medidas de eficiencias.

2.8.3. Detector Prototipo (Tracking Prototype Detector)

El detector prototipo MINER ν A para rastreo de trayectorias es el 20 % del detector completo MINER ν A , conteniendo calorimetría y parte del detector central.

El objetivo de éste prototipo es el de construir un número significativo de los módulos del detector MINER ν A y mostrar su rendimiento para el rastreo de partículas de rayos cósmicos, además permite verificar la operación y funcionamiento del detector antes del montaje final para el detector completo, entre ellos, pruebas de hardware, PMTs, y estabilidad del software

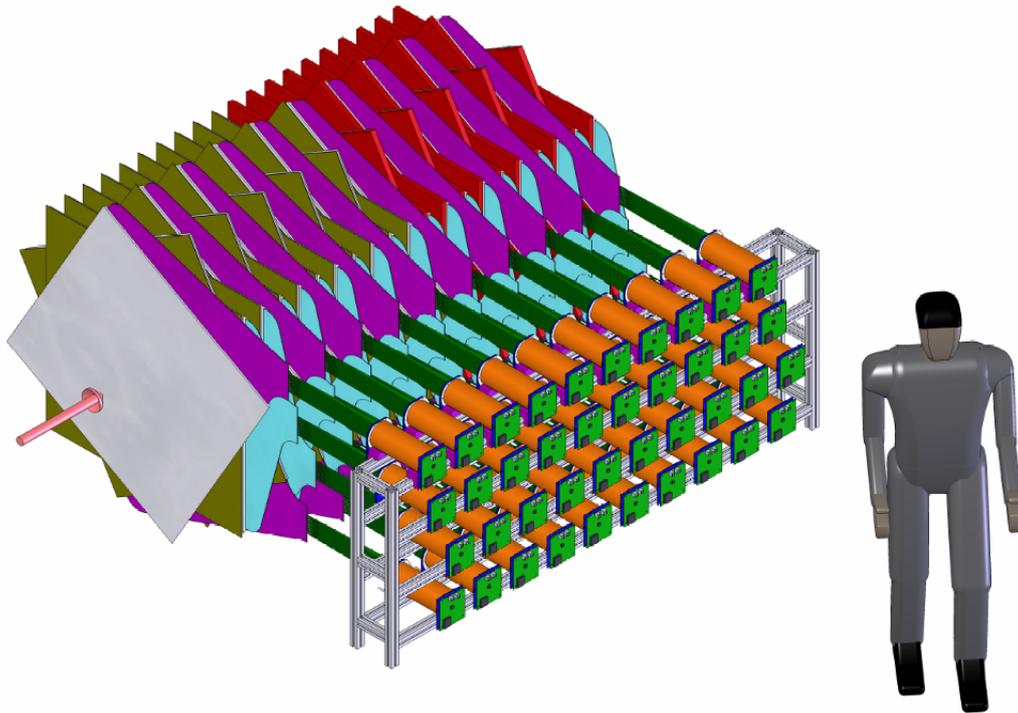


Figura 2.9: Diseño Conceptual del TestBeam Detector

de adquisición de datos.

2.9. Pruebas al detector completo

El detector completo empleará cerca de 500 PMTs, cada evento sin supresión de zeros será del orden de 1MB, sin embargo la mayoría de eventos contendrán supresión de zeros y serán del orden de 100kB por evento, ocurriendo cada 2 segundos. La tasa de eventos será de 5GB a 10GB por día, lo que equivale a 1,5TB a 3TB al año. En cuatro años corriendo, MINERVA tendrá una cantidad datos entre 6TB y 12TB.

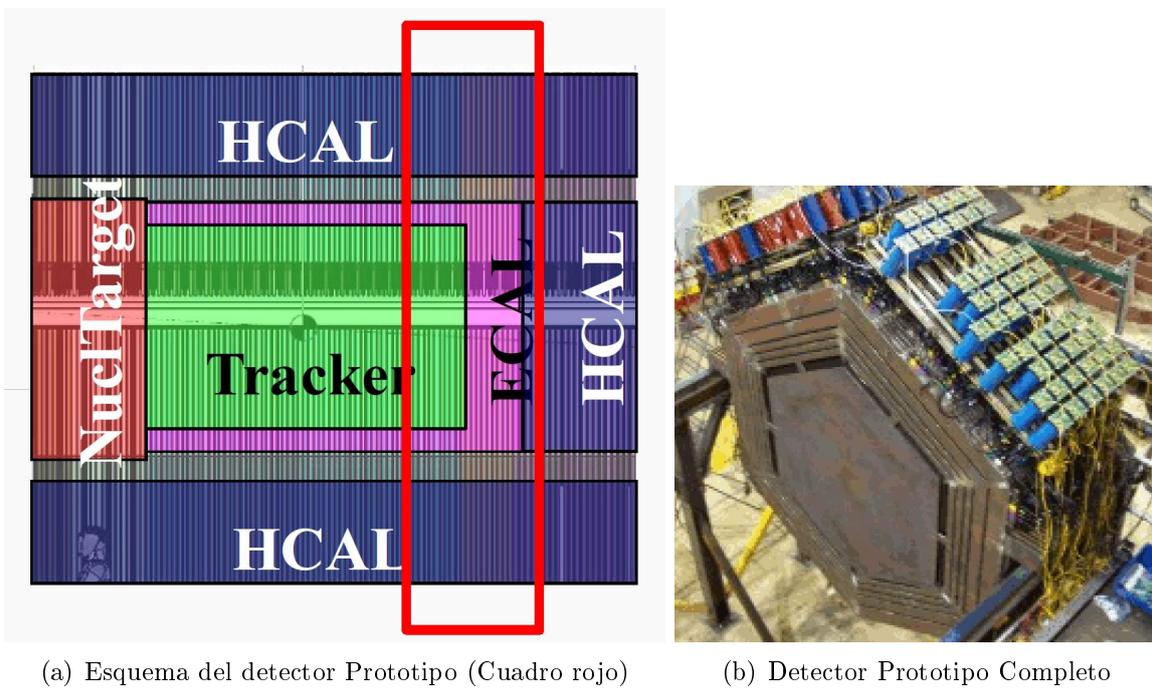


Figura 2.10: Detector Prototipo

3.1. Introducción

La luz adquirida por cerca de 30.000 centelleadores en el experimento MINER ν A debe ser convertida en pulsos eléctricos, la cual contiene información bastante precisa de tiempo y una amplitud proporcional a la energía depositada.

Para cumplir con dicho objetivo, el experimento MINER ν A utiliza tubos fotomultiplicadores R7600U-00-M64 (PMTs) multi-ánodo de 64 canales o píxeles, proporcionados por Hamamatsu Photonics[38], dividido en 19 PMTs por cada plano XU/XV en el detector como instrumento básico de detección. Cada PMT está cubierto por un cilindro de acero llamado “PMT box”, para aislarlos de la luz del entorno o campos magnéticos.

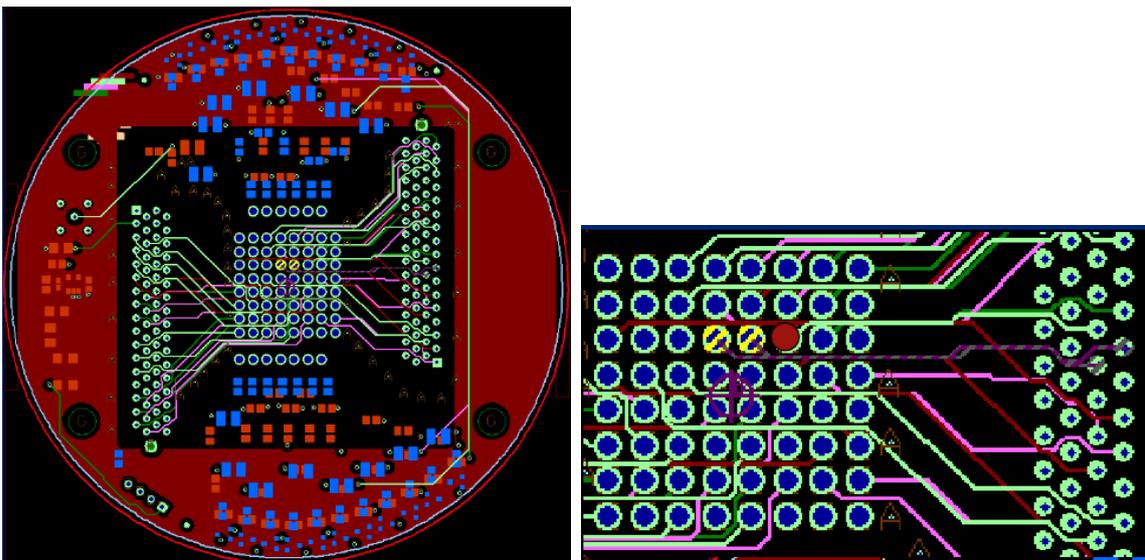
La señal de entrada de cada tubo fotomultiplicador son las adquiridas por los centelleadores a través de las llamadas “fibras de señal clara”, y por dos fibras ópticas conectadas a una caja de inyección de luz de prueba que servirán para el cálculo de ganancia en el experimento. Dichas señales son amplificadas y convertidas en señales digitales de tiempo corto mediante unas tarjetas electrónicas de lectura (Front End Boards:FEBs), que luego serán convertidas a medidas físicas. Con tubos multiánodo, la linealidad en las medidas y el crosstalk electrónico en los FEBs son temas a considerar.

El objetivo de ésta sección es mostrar los análisis hechos de linealidad de medida y crosstalk

electrónico, así como el empleo de ajustes tri-lineales para la calibración de las tarjetas electrónicas, pasando de cuentas digitales a medidas físicas, la implementación de histogramas online para los discriminadores en dichos FEBs y el cálculo de ganancia de los tubos fotomultiplicadores.

3.2. Estudios de Crosstalk Eléctrico

Dado que las señales analógicas leídas por los tubos fotomultiplicadores son luego convertidas a señales digitales, se considera importante saber cual es el factor en que puede verse afectada la lectura para un canal o píxel, debido a la señal óptica de un vecino en las fibras conectadas a los tubos, o debido a la corriente que pasa por un píxel contiguo a la leída en las tarjetas de lectura electrónica.



(a) Pixel 46 coloreado de rojo

Figura 3.1: Esquema de una Tarjeta Electrónica de Lectura utilizada en MINERνA

Se piensa que el crosstalk es uno de los responsables de desplazamientos y ensanchamientos de las medidas Pedestales, para estudiar el crosstalk es necesario datos en los que 1 solo píxel o canal sea iluminado.

3.2.1. Procedimiento Experimental

Para el análisis del comportamiento de la señal en los canales vecinos al canal medido, se utilizó el siguiente método:

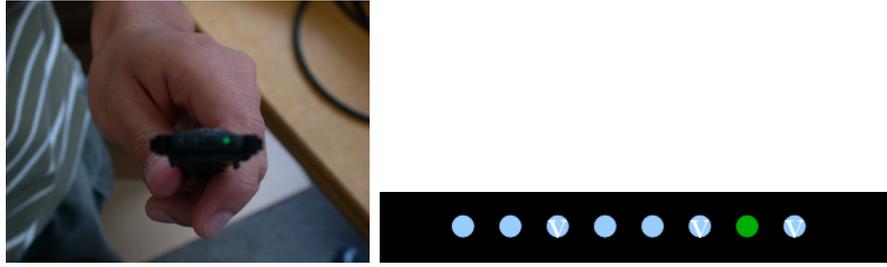
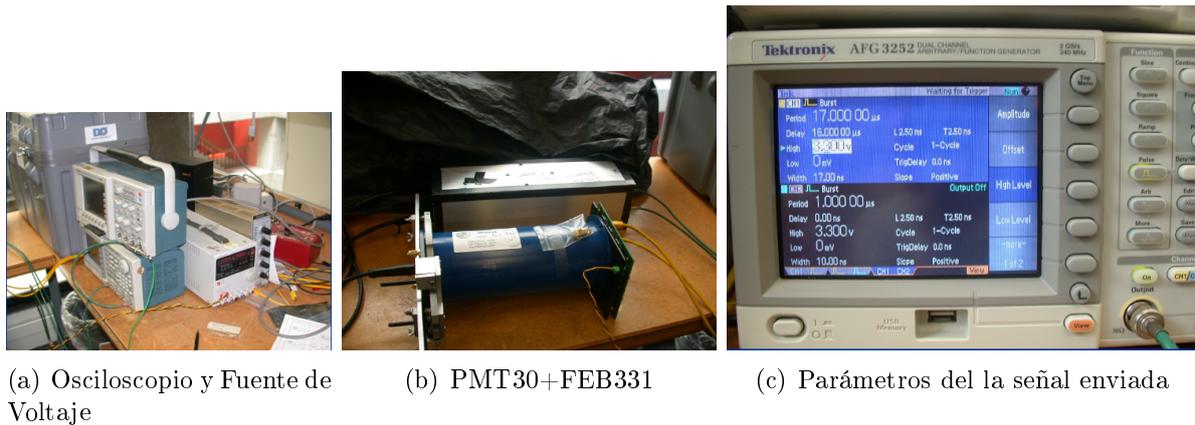


Figura 3.2: Cable óptico manipulado para el estudio



(a) Osciloscopio y Fuente de Voltaje

(b) PMT30+FEB331

(c) Parámetros de la señal enviada

Figura 3.3: Equipo utilizado

- Se obtuvo uno de los cables de fibra óptica de 8 conectores que se utilizan para leer la señal desde los centelleadores hacia los PMTs en el detector.
- Se cortaron los conectores hasta solo dejar uno sólo (Ver figura 3.2).
- Se conecta la fibra al PMT y a una caja de inyección de luz.
- Se definió una señal cuadrada de 3.3V (es la intensidad utilizada para pruebas de baja intensidad de luz, en el experimento llamada “1 PE”), así como un período de $17\mu\text{s}$, y un ancho de 17ns (ver figura 3.3(c)).
- Se alimentó el FEB con 5V.
- Se tomaron datos con 2000 eventos cada uno.

Se hizo lo mismo con 2 diferentes PMTs (PMT025 y PMT030) y FEBs (FEB331 y FEB320).

3.2.2. Coeficientes de Correlación y Crosstalk Eléctrico

El coeficiente de correlación (coeficiente de correlación momento-producto de Pearson) es una herramienta estadística que indica la relación lineal entre dos variables cuantitativas.

El coeficiente de correlación está definido como[49]:

$$\rho = \frac{cov(x, y)}{\sigma_x \sigma_y} \in [-1, +1] \quad (3.1)$$

En donde:

$$\begin{aligned} \bar{x} &= \frac{1}{N} \sum_{i=1}^N x_i \\ \sigma_x &= \sqrt{\frac{1}{N} \sum_i (x_i^2 - \bar{x}^2)} \\ cov(x, y) &= \frac{1}{N} \sum_i (x_i - \bar{x})(y_i - \bar{y}) \\ &= \bar{xy} - \bar{x}\bar{y} \end{aligned}$$

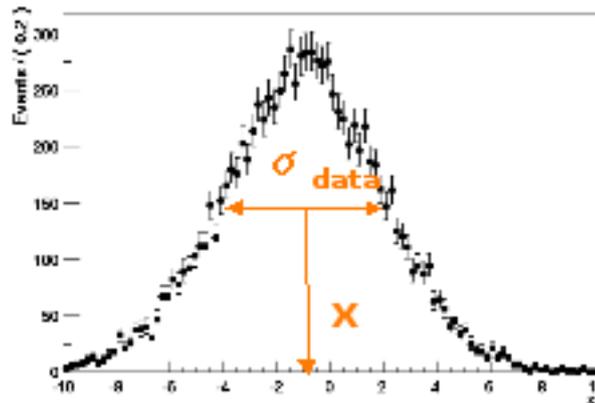


Figura 3.4: Ilustración gráfica de σ para una muestra de datos

Es decir, para una muestra de datos con dos variables x, y , \bar{x}, \bar{y} , son las medias o valores promedio de las variables de nuestra muestra, σ_x, σ_y las desviaciones estándar para cada variable de nuestra muestra y $cov(x, y)$ la covarianza entre dichas variables, que nos describe la dependencia entre éstas (nótese que la covarianza tendría dimensiones $D(x)D(y)$, mientras que el coeficiente de correlación es adimensional).

Cuando el coeficiente es positivo, esto indica una correlación directa, por ejemplo para un coeficiente de correlación perfecto $\rho = 1$, implica que si una de las variables aumenta, la otra lo hace en la misma proporción, para $\rho = -1$ implicaría la segunda variable disminuye en una idéntica proporción[50] [54], como se aprecia en figura 3.5.

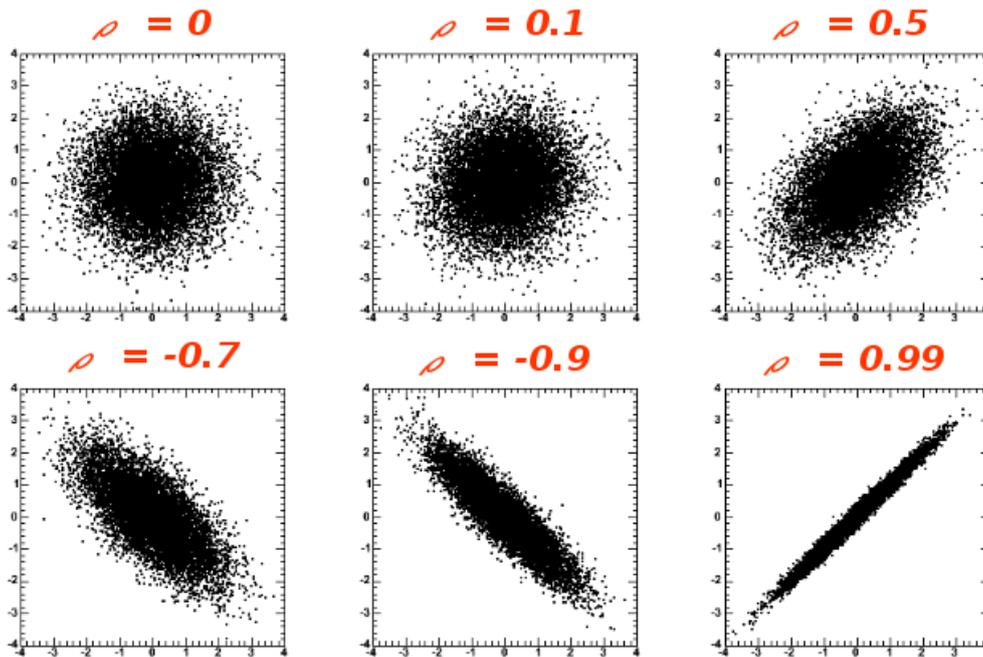


Figura 3.5: Un ejemplo de coeficientes de correlación para una distribución x,y

Además, definiremos también un “coeficiente de crosstalk eléctrico” como [51]:

$$C_{ij} = \frac{Q_{j,i} - Q_i}{M_j - Q_i} \quad (3.2)$$

En donde:

- $Q_{j,i}$ es la media del Pedestal del píxel i cuando el píxel j es iluminado.
- Q_i es la media del Pedestal del píxel i durante una corrida de pedestal
- Q_j es la media del Pedestal del píxel j durante una corrida de pedestal
- M_j es la media de la distribución del píxel j cuando es iluminado

Para el análisis, y cálculo de los coeficientes de correlación y de crosstalk eléctrico, se escribieron scripts en ROOT que nos proporcionan gráficas de la distribución de carga en escala logarítmica de la ganancia media de la señal medida y la ganancia alta de los canales vecinos, así como gráficas bidimensionales de dispersión de la ganancia media del píxel tomado, con la alta para cada píxel vecino [52] (ver figuras 3.6 y 3.7 y cuadros 3.1 y 3.2). El resultado de los coeficientes de crosstalk eléctrico pueden verse en la figura 3.8

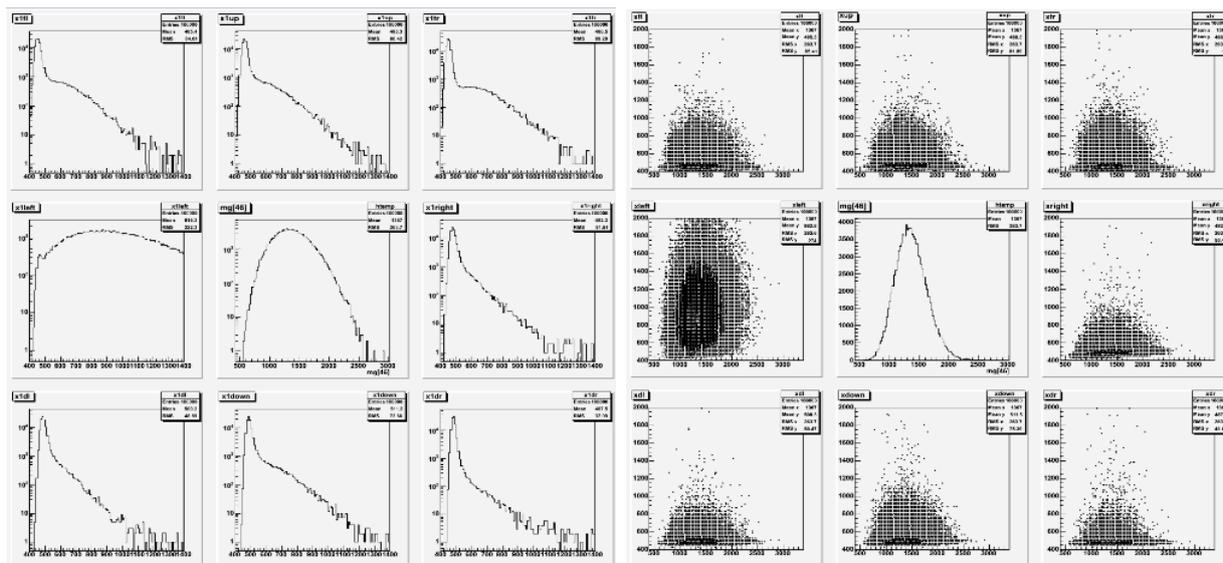


Figura 3.6: Resultados PMT25/FEB331, píxel 46

Coefficientes ($\times 10^{-2}$)

2.59	6.37	3.2
1.71	Pixel 46	15.6
3.29	5.85	1.96

Cuadro 3.1: Coeficientes de Correlación: Píxel 46, PMT25/FEB331

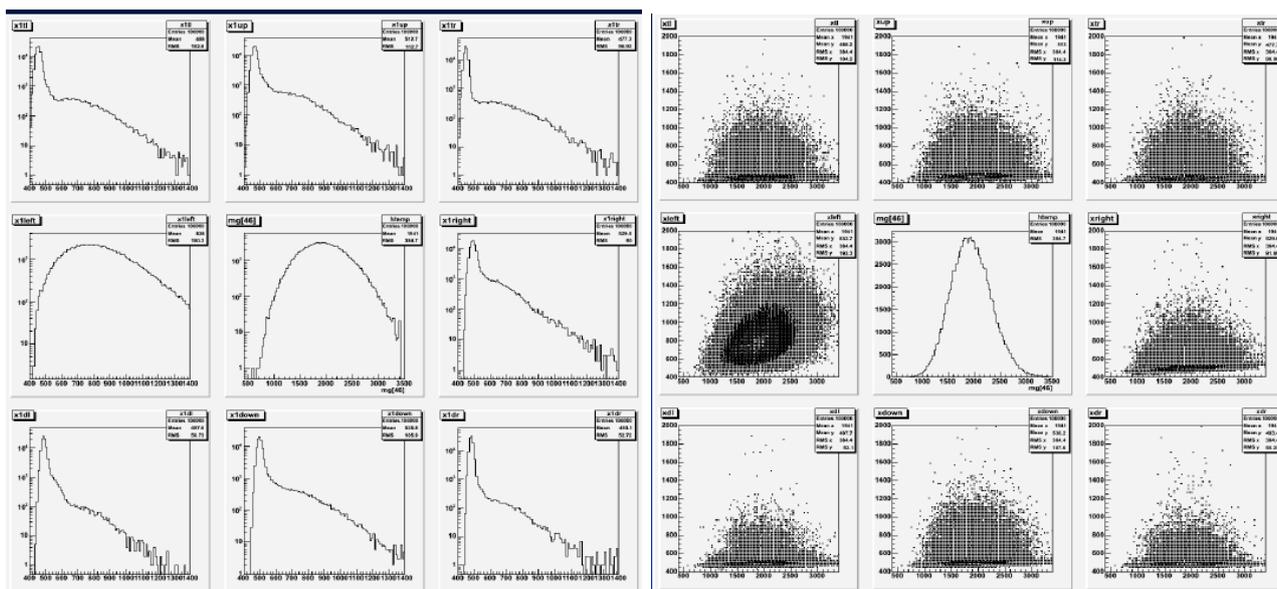


Figura 3.7: Resultados PMT30/FEB331, píxel 46

Coeficientes ($\times 10^{-2}$)		
11.18	12.01	5.92
7.66	Pixel 46	16.84
9.78	16.42	9.41

Cuadro 3.2: Coeficientes de Correlación: Píxel 46, PMT30/FEB331

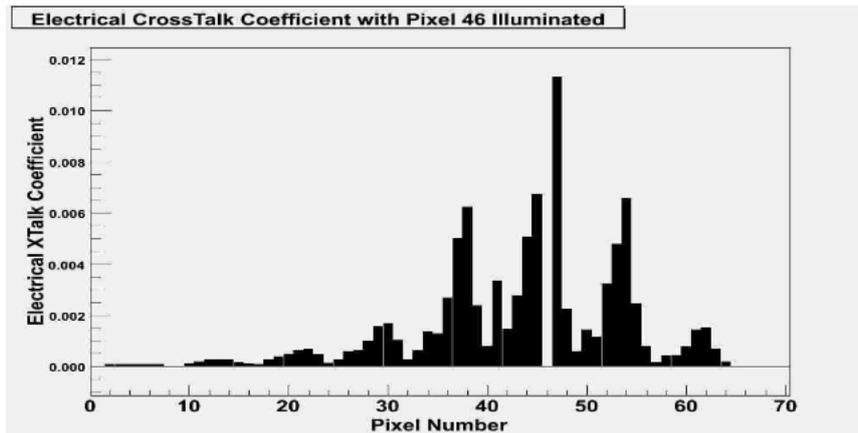


Figura 3.8: Coeficientes de Crosstalk Eléctrico con píxel 46 iluminado

- El canal vecino izquierdo muestra una gran cantidad de luz, al parecer principalmente por crosstalk óptico.
- El píxel vecino derecho es el que recibe mayor crosstalk eléctrico.
- Se aprecia de la figura 3.1(a), que el cable asociado al píxel 46 está bastante junto al cable que conecta al píxel 47, muy probablemente la razón por la cual se aprecia más crosstalk en éste píxel.
- Se espera una estructura periódica en la figura 3.8, además de mayor magnitud en los coeficientes para los píxeles vecinos.
- Del mismo gráfico 3.1(a), se puede apreciar que el píxel 41 no sigue el patrón.
- Sin embargo éste comportamiento es también esperado, observando la figura 3.2, se puede apreciar que el cable (rojo) asociado a dicho píxel (41) pasa también muy de cerca a los cables de los píxeles 46 y 47, por lo que, de iluminarse cualquiera de ellos, es de esperarse relativamente mayor coeficiente de crosstalk en los otros dos píxeles que en el resto de píxeles.
- Para la calibración, se necesitaría un análisis de coeficientes de crosstalk vs intensidad de

luz inyectada en un píxel y un modelo que nos permita contar la variación de forma de una distribución cuando hay crosstalk eléctrico.

3.3. Estudios de linealidad en las mediciones

Éste estudio sirve para ver la estabilidad de mediciones relativamente fijas, utilizando inyección de luz de prueba en los tubos fotomultiplicadores, calculando la media y el RMS (*root mean square* o raíz cuadrada media) de los espectros de distribución de carga para cada píxel.

3.3.1. Pruebas con medidas “Pedestales” (sin inyección de luz)

3.3.1.1. Descripción

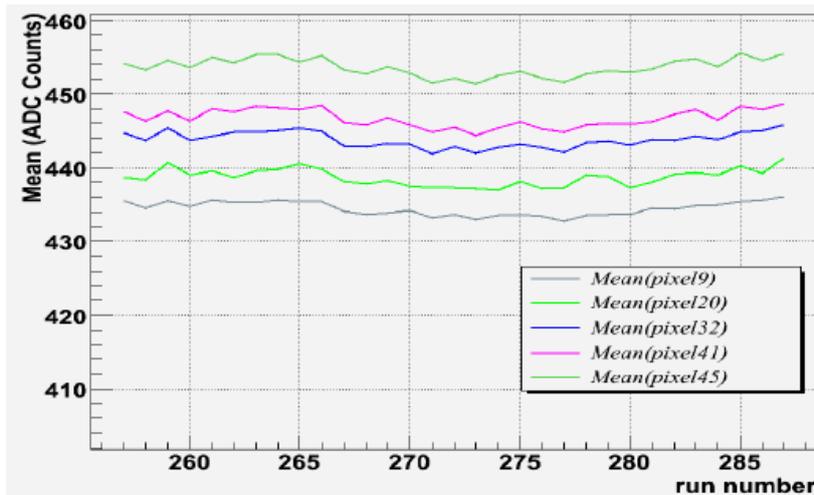
Primero se realizaron pruebas sin inyectar luz, para medir la linealidad de las medidas de background (datos pedestales) en cada píxel. Ésto sirve para analizar si hay cambios drásticos en las medidas debido a factores externos en el laboratorio, como el encendido o apagado de las luces en la construcción, o posibles cambios de temperatura.

Para dicho propósito, se tomaron datos cada hora durante un día entero o más varias veces, utilizando múltiples tubos fotomultiplicadores y tarjetas FEB.

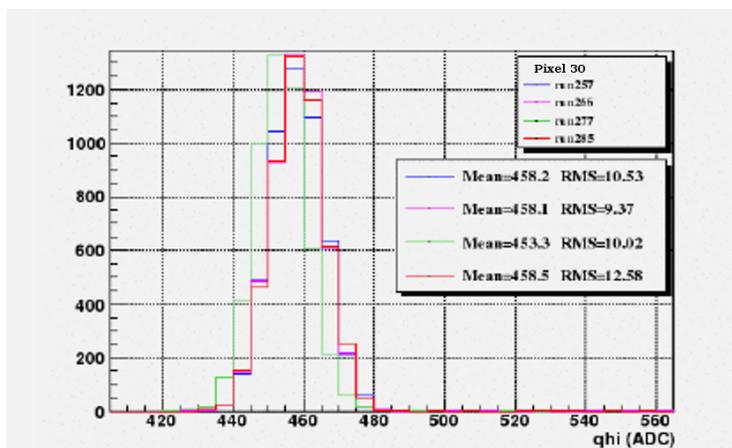
3.3.1.2. Resultados

Los resultados de estabilidad, que se pueden ver en las figuras 3.9 y 3.10, son los siguientes:

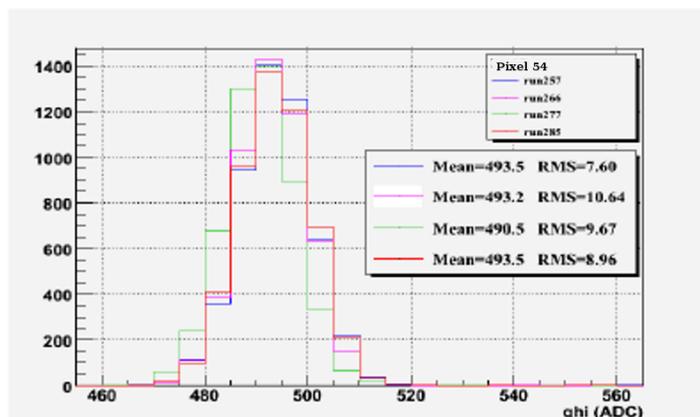
- En la figura 3.9(a), tenemos una gráfica que representa la evolución en el tiempo de la Media (en ADC Counts) de 5 pixels distintos de un mismo PMT/FEB. En éste caso la gráfica va desde el Run_number o *número de corrida* 257 hasta el 287, que representan 30 horas en total (ya que se tomaron datos con 1 hora de diferencia entre cada corrida). Las variaciones no son muy significativas (se verá un poco más en la figura 3.10)
- Las figuras 3.9(b) y 3.9(c), nos muestran los datos tomados de los espectros de distribución de carga de 2 pixels distintos, aproximadamente cada 9 horas. Se aprecia de las figuras, que las distribuciones no se mueven significativamente.



(a) Gráfica la Media (Mean) de varios pixels vs Run_Number (tiempo)



(b) Espectros de distribución de carga (qhi) de 5 datos diferentes para un mismo pixel (30)



(c) Espectros de distribución de carga (qhi) de 5 datos diferentes para un mismo pixel (54)

Figura 3.9: Estabilidad de la toma de datos de las medidas Pedestales c/hora durante más de 1 día

En la figura 3.10, tenemos la desviación estándar respecto al promedio de todos los datos (30 corridas), donde cada dato corresponde a la Media del espectro de distribución de carga en cada píxel es:

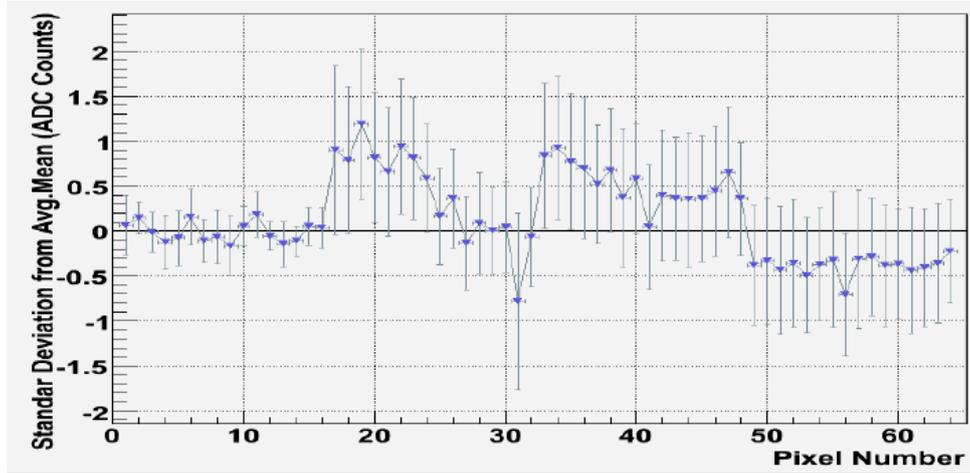


Figura 3.10: Desviación estándar del promedio de las Medias de carga en cada píxel

- En ésta figura podemos apreciar la desviación estándar respecto al promedio de cada píxel tomado en un mismo PMT/FEB (64 pixels en total). La desviación estándar en algunos pixels es de hasta 1 ADC Count, lo cual podría representar una variación de hasta 0.7% respecto a la media promedio, implicando que dicha variación no es muy significativa.
- Se hicieron las mismas pruebas durante varios días y comenzando a diferentes horas obteniendo buenos resultados en todos los PMT/FEBs [55] [56]

3.3.2. Pruebas con medidas a MáxPE (con “máxima” inyección de luz)

3.3.2.1. Descripción

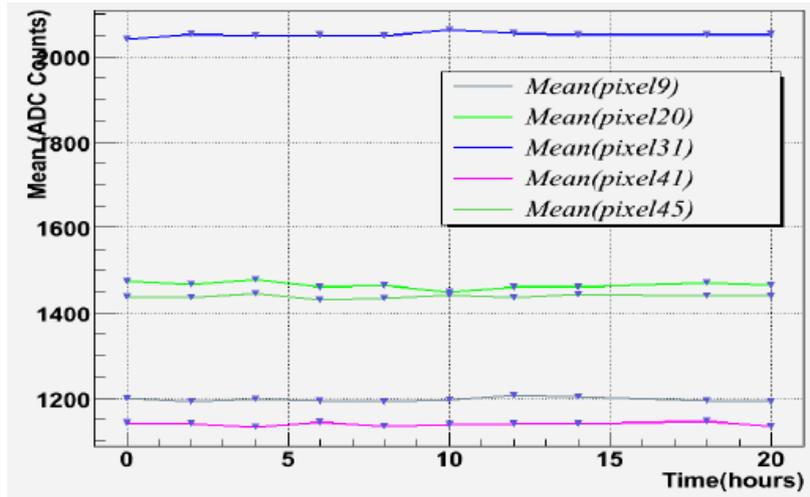
Éstas pruebas, tienen como propósito medir la linealidad de las medidas tomadas a una intensidad de luz constante, permitiéndonos a su vez, probar el sistema de inyección de luz, el cual servirá más adelante para calcular las medidas de ganancia en los PMTs y tarjetas electrónicas de lectura.

El método para la toma de datos es idéntico al de las pruebas sin inyección de luz.

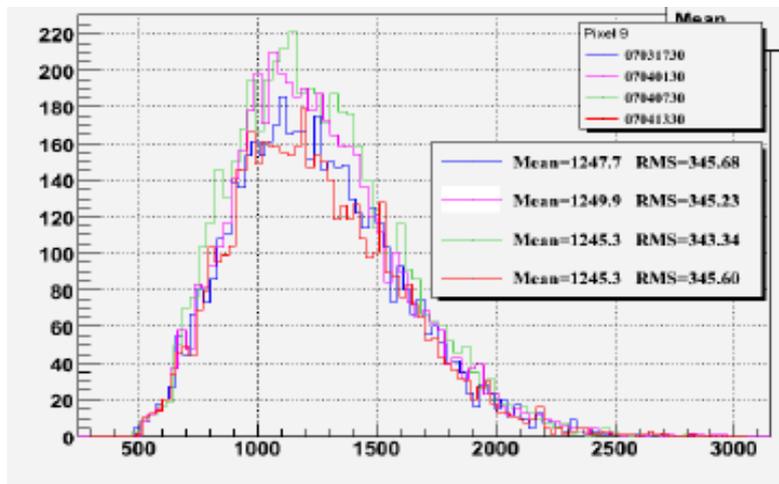
3.3.2.2. Resultados

Los resultados se aprecian en las figuras 3.11 y 3.12, y son los siguientes:

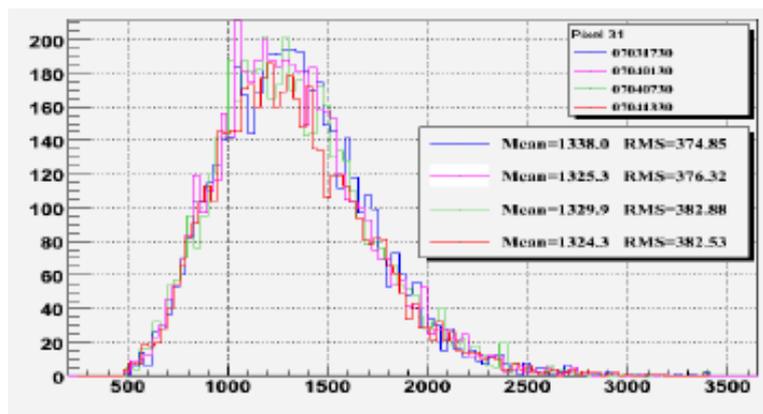
- En la figura 3.11(a), de manera similar a las pruebas con medidas pedestales, podemos ver la evolución en el tiempo de la Media (en ADC Counts) de 5 pixels distintos de un mismo PMT/FEB durante 20 horas.
- Las figuras 3.11(b) y 3.11(c), nos muestran los datos tomados de los espectros de distribución de carga de 2 pixels distintos, aproximadamente cada 6 horas. Se aprecia de las figuras, que las distribuciones no se mueven significativamente.



(a) Gráfica la Media (Mean) de varios pixels vs Tiempo (horas)



(b) Espectros de distribución de carga (qhi) de 4 datos diferentes para un mismo pixel (9)



(c) Espectros de distribución de carga (qhi) de 4 datos diferentes para un mismo pixel (31)

Figura 3.11: Estabilidad de la toma de datos de las medidas Pedestales c/hora durante casi 1 día

La figura 3.12, nos muestra la desviación estándar con respecto al promedio de las Medias de los espectros de carga en cada píxel:

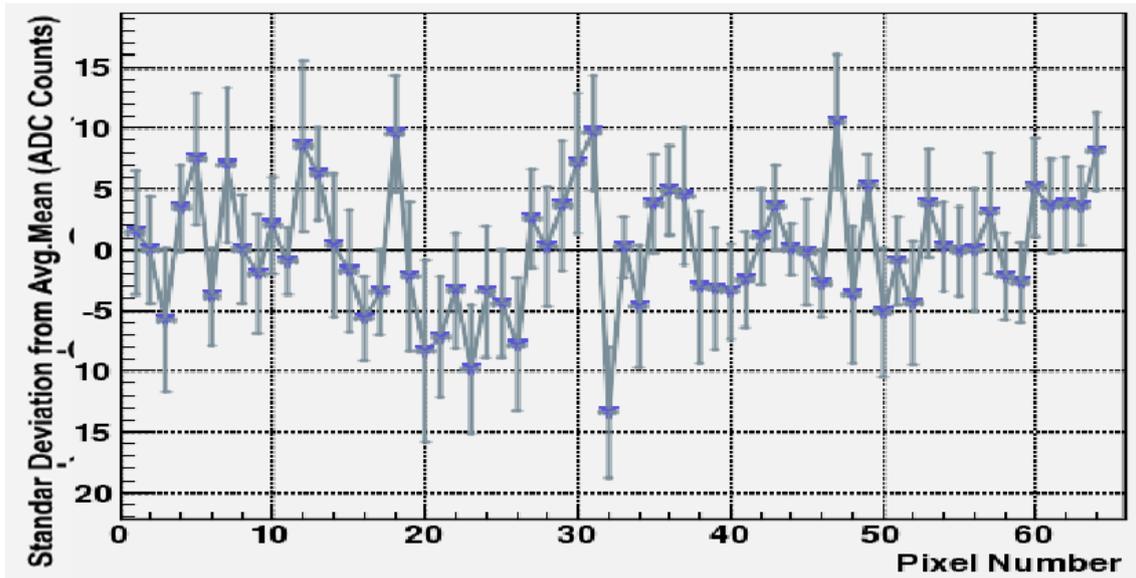


Figura 3.12: Desviación estándar del promedio de las Medias de carga en cada píxel

- En ésta figura, se puede apreciar en algunos pixeles que la desviación estándar es de hasta 6 ADC Counts con respecto a la media promedio, lo cual representa una variación de hasta 1.5 %.
- Al igual que con los pedestales, se hicieron las mismas pruebas durante varios días y comenzando a diferentes horas obteniendo resultados similares.

3.4. Estudio de filtraciones de luz en los tubos fotomultiplicadores

Los estudios de filtración de luz son realizados cada vez que nuevos grupos de 4 módulos son instalados y completamente cableados en el detector, en el futuro se necesitarán hacer dichas pruebas con el detector completo cada cierto tiempo para detectar y resolver posibles problemas de filtraciones que puedan generarse con el paso de los días.

Cuando se toman datos con los tubos fotomultiplicadores en un entorno *aislado* (digamos, de luz o campos electromagnéticos), se deberían registrar puras señales Pedestales. Para dicho propósito, los tubos se encuentran debidamente encerrados en contenedores cilíndricos metálicos, a prueba de luz, campos electromagnéticos e incluso muones provenientes de rayos cósmicos.

Sin embargo, es posible que la luz externa se filtre -por ejemplo- a través de los alrededores de los conectores ópticos que van a los PMTs, o a través de centelleadores que no hayan sido debidamente cubiertos en el detector. Por ésta razón, éstos estudios son necesarios.

3.4.1. Procedimiento Experimental

- Los datos para éstos estudios se llaman *Pedestal Runs*, las cuales son tomadas sin inyectar nada de luz. Para esto, el software de toma de datos (OnlineControl), se coloca en modo *One Shot Mode*[32]¹, pero cuando el detector esté completo y funcionando constantemente, se tomarán datos pedestales cada cierto tiempo junto con los eventos con neutrinos, caso para el cual solo se tomarán los eventos con pedestales para éstos estudios.
- Debido a que, si la filtración de luz es muy fuerte, podría saturar los tubos fotomultiplicadores, es necesario empezar haciendo el estudio con voltajes seguros. El procedimiento se empieza con 600V, y se va aumentando de 50 en 50 hasta los 900V.

En las tarjetas electrónicas de lectura utilizadas en el experimento, 1V equivale a 58.79333 ADC counts, por lo que los valores colocados son:

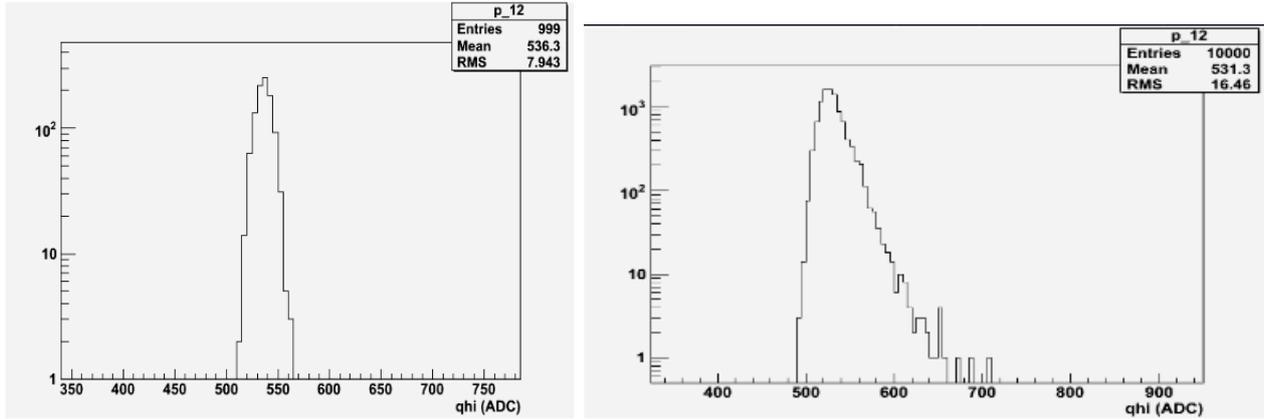
Voltios	ADC Counts
600	35276
650	38216
700	41156
750	44095
800	47035
850	49975
900	52914

Cuadro 3.3: Equivalencia entre los valores de Entrada en Voltios y ADC Counts

3.4.2. Análisis de filtraciones de luz

Cuando buscamos filtraciones de luz en el espectro de una distribución ADC de Pedestales, ésta luce como si le añadiera una cola pasado el pico del pedestal a la derecha, esto hará que el RMS de la distribución y del ajuste gaussiano se incrementen.

¹Existen 3 modos base para tomas de datos: OneShot (utilizado principalmente para pruebas con inyección de luz, utiliza un generador de pulsos como trigger), cosmicrays (para tomar datos de rayos cósmicos, utiliza unos Veto Walls como trigger) y MixMode



(a) Lectura normal del Pedestal de un pixel

(b) Lectura con filtraciones de luz

Figura 3.13: Ejemplo de filtraciones de luz en una lectura

Para el análisis, se escribió un script en ROOT D que busca en todas las tarjetas y todos los píxeles la ganancia alta de cada señal, luego hace un ajuste gaussiano y obtiene la desviación estándar σ de dicho ajuste ($h \rightarrow GetFunction("gaus") \rightarrow GetParameter(2)$).

Si el parámetro obtenido es mayor a un corte definido en 10 ADC Counts (se estima que el RMS de la distribución debería encontrarse entre 7 y 8 ADC Counts), se imprime el pixel en pantalla.

```
BasicHistos/elec_0_0_1_0_5/qhi/p_47 : 537.741  11.8956
BasicHistos/elec_0_0_1_0_5/qhi/p_48 : 524.629  24.7312
BasicHistos/elec_0_0_1_0_5/qhi/p_49 : 518.547  86.7867
BasicHistos/elec_0_0_1_0_5/qhi/p_50 : 523.658  26.1579
BasicHistos/elec_0_0_1_0_5/qhi/p_51 : 527.637  67.3227
BasicHistos/elec_0_0_1_0_5/qhi/p_52 : 509.789  46.0077
```

3.5. Implementación de Histogramas Online en GAUDI para los discriminadores en las tarjetas electrónicas de lectura en el paquete ForestTest del software MINER ν A

Se disponen de discriminadores adjuntos a los canales de ganancia alta de las tarjetas de lectura electrónica. Existen 2 discriminadores por FEB, uno por cada 32 píxeles o canales. Cada vez que un canal recibe una señal de carga de más del threshold definido, el discriminador

registra un timestamp ², después el FEB continúa registrando datos por 150ns más, antes de resetearse y grabar dicha carga como un “hit”, para tomar más datos. Dichos discriminadores se utilizan principalmente para -tal y como su nombre lo sugiere- discriminar buena parte de las señales de baja intensidad, consideradas como “ruido” (después de la supresión de background en las señales, quedan aún algunos hits que, se pueden desaparecer en gran medida con el uso de los discriminadores).

El trabajo hecho en ésta sección, consiste en la implementación de histogramas para los discriminadores en “tiempo-real” al momento de tomar los datos, a través de GAUDI (ver figura 3.5).

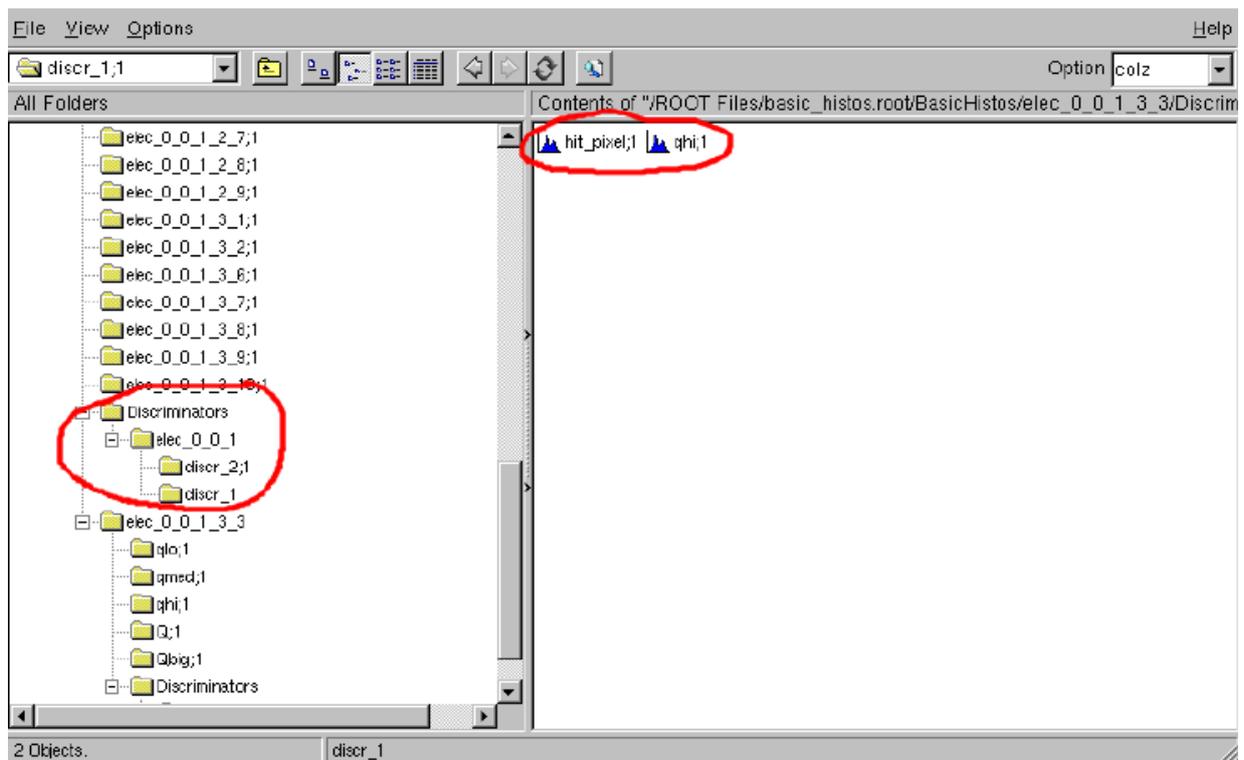


Figura 3.14: Estructura ramificada del objeto (Tree) en Root con los histogramas implementados

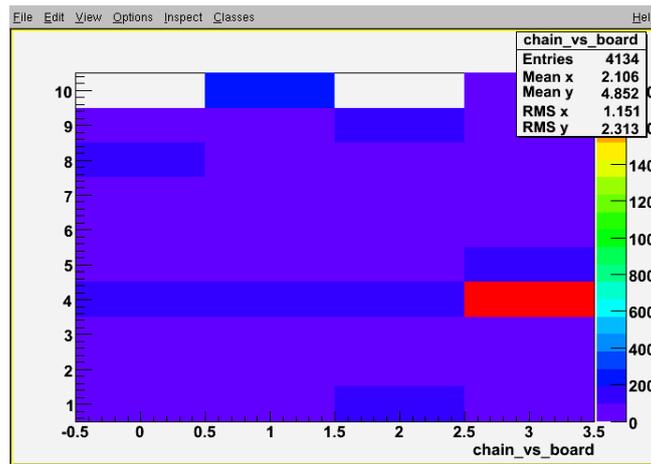
La figura 3.5, muestra la estructura de los objetos en ROOT para la parte de discriminadores. En el primer círculo rojo a la izquierda, podemos elegir el FEB deseado y los histogramas implementados para cualquiera de sus 2 discriminadores.

En el segundo círculo a la derecha, se encuentran disponibles para seleccionar los histogramas *hit_pixel*, que nos muestra el número de hits por pixel en un discriminador y *ghi*, que nos

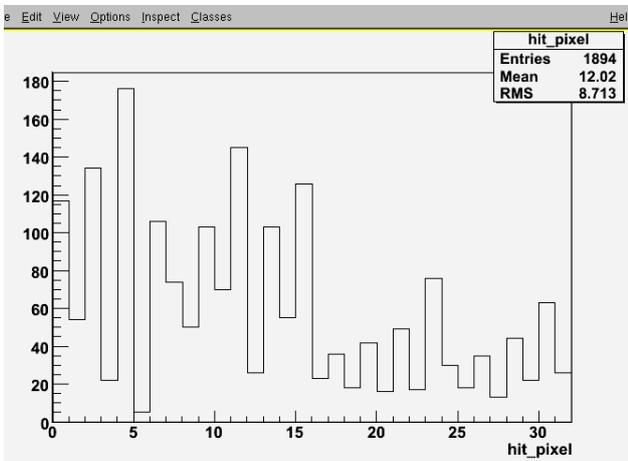
²Timestamp es una secuencia de caracteres, que denotan la hora y fecha (o alguna de ellas) en la cual ocurrió determinado evento

da el espectro de distribución de carga de un píxel para hits que pasan un threshold especificado en el discriminador (los histogramas producidos se pueden ver en las figuras 3.15(b) y 3.15(c)). La figura 3.15(a) nos muestra una gráfica con el número de Hits por PMT (chain y board son las direcciones electrónicas que nos permiten diferenciar entre cada PMT).

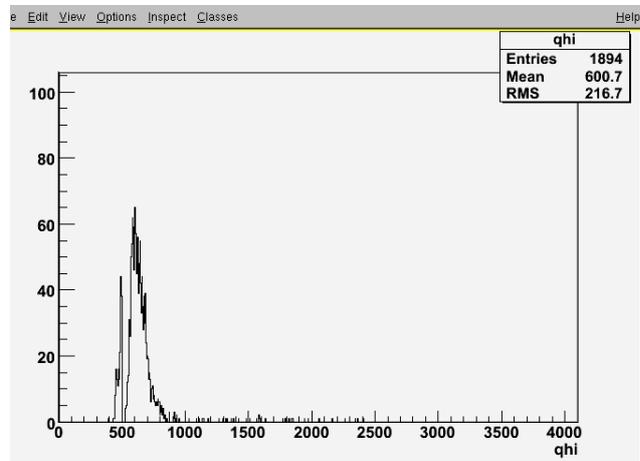
En el detector MINER ν A se utilizan 4 Chains por módulo, 2 con 9 PMT/FEBS o “Boards” (chains 0 y 2) y otros 2 con 10 (chains 1 y 3).



(a) Chain vs Board:Número de Hits por PMT



(b) Número de Hits por píxel en un discriminador



(c) Distribución de carga de un píxel, para hits que pasaron el threshold

Figura 3.15: Gráfico de Histogramas implementados para los Discriminadores en cada FEB

Las aplicaciones inmediatas a dichos discriminadores, son:

- La verificación inmediata de filtraciones de luz en un tubo (cuando el número de hits en un tubo es demasiado alta en comparación a las demás, ver figura 3.15(a)).

crate	croc	chain	board	pixel	hg_mean	hg_rms	hg_entries
0	1	0	1	0	546.026	6.35762	1988
0	1	0	1	1	541.333	5.18876	1990
0	1	0	1	2	535.591	5.88884	1985

Cuadro 3.4: Ejemplo de tabla producida por el algoritmo PedestalFinder

- La búsqueda de píxels que registren muy pocos hits en un tubo (lo cual puede ser indicio de un píxel “muerto”, ver figura 3.15(b)) cuando se utilizan los discriminadores.
- La visualización de la distribución de carga que pasa el threshold definido por el discriminador en cada píxel (ver figura 3.15(c)).
- Verificar la respuesta en general de los datos registrados con los discriminadores activados.

3.6. Cálculo de Ganancia de los Tubos Fotomultiplicadores

3.6.1. Procedimiento Experimental

- EL cálculo de ganancia se realiza utilizando el *sistema de inyección de luz* para diferentes voltajes, tomando pedestales de 2000 eventos e inmediatamente datos a máxima intensidad de luz (MaxPE) de 5000 eventos cada uno en todos los tubos fotomultiplicadores [57] (ver figura 3.18(a)).
- Una vez hecho esto, se utiliza el paquete MinervaEvent, para pasar los datos de Pedestales a través de un algoritmo llamado PedestalFinder, el cual produce una tabla con información de los valores de las *medias* y *rms* de todos los píxels para las ganancias altas, bajas y medias en cada tubo fotomultiplicador, así como el número de entradas registrado en éstos.

En la tabla 3.4, puede apreciarse un ejemplo de lo que es producido por éste algoritmo, en donde los valores de *crate*, *croc*, *chain*, *board*, son la dirección que describe la ubicación de cada tubo en la configuración electrónica, *hg_mean* es la media promedio de dicho píxel, *hg_rms* el RMS, y *hg_entries* el número de hits, (valores similares son calculados para la ganancia media y alta, pero se han omitido en la tabla).

- Los datos tomados empleando inyección de luz, se pasan por el algoritmo llamado OfflinePedSup, el cual, utilizando la información de las tablas producidas en el algoritmo previo, suprime los pedestales de dichas señales (lo que significa una señal más limpia), y luego graba los datos procesados a un nuevo archivo.

- Los archivos procesados por los algoritmos mencionados, son luego pasados a través del Paquete ForestTest, el cual hace uso de un *tool* ubicado en el paquete RawToDigit y un algoritmo dentro del mismo paquete: IGetCalCharge, y BasicHistosAlg.

IGetCalCharge accede a la base de datos de los ajustes tri-lineales para convertir la carga de unidades digitales (ADC Counts) a Coulombs (ver figuras 3.6.1 y 3.6.1).

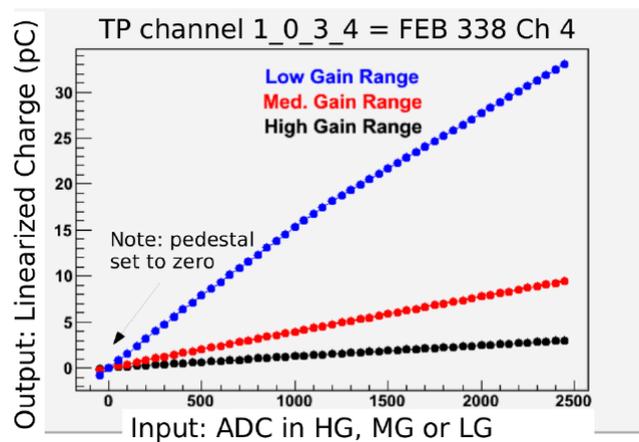


Figura 3.16: Ejemplo de Ajuste Trilineal en una de las tarjetas electrónicas de lectura

Una vez pasado por el ajuste, los datos calibrados son almacenados en archivos generados por el algoritmo *BasicHistosAlg*, y se utilizan para el cálculo de ganancias.

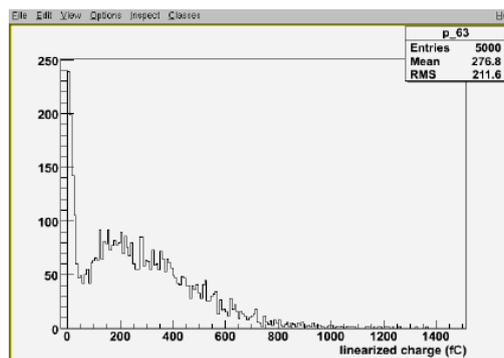


Figura 3.17: Distribución de Carga después de la calibración

- Para el cálculo de ganancia, primero se calcula el número de fotoelectrones[pe], el cual representa la intensidad de luz inyectada en el píxel del tubo [59]:

$$pe = K \left(\frac{Media_{dato} - Media_{pedestal}}{\sqrt{RMS_{dato}^2 - RMS_{pedestal}^2}} \right)^2$$

En donde K , es una constante de corrección", elegida en base a experiencias en el experimento MINOS.

- La ganancia es definida como:

$$gain = \frac{Media_{dato} - Media_{pedestal}}{pe} \cdot \frac{1e - 15 [C]}{1,622e - 19 [C/e]}$$

Y debido a la supresión de pedestal hecha, $Media_{pedestal} = 0$.

Se hacen éstos cálculos en todos los píxeles, a través de un script hecho en ROOT (ver apéndice, sección D.1), obteniendo los resultados de las figuras 3.19 y 3.20.

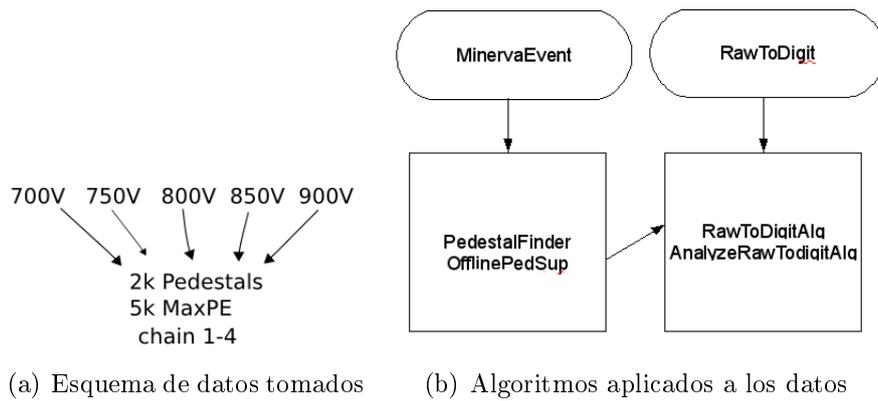


Figura 3.18: Esquema de Procedimientos para el Cálculo de Ganancia

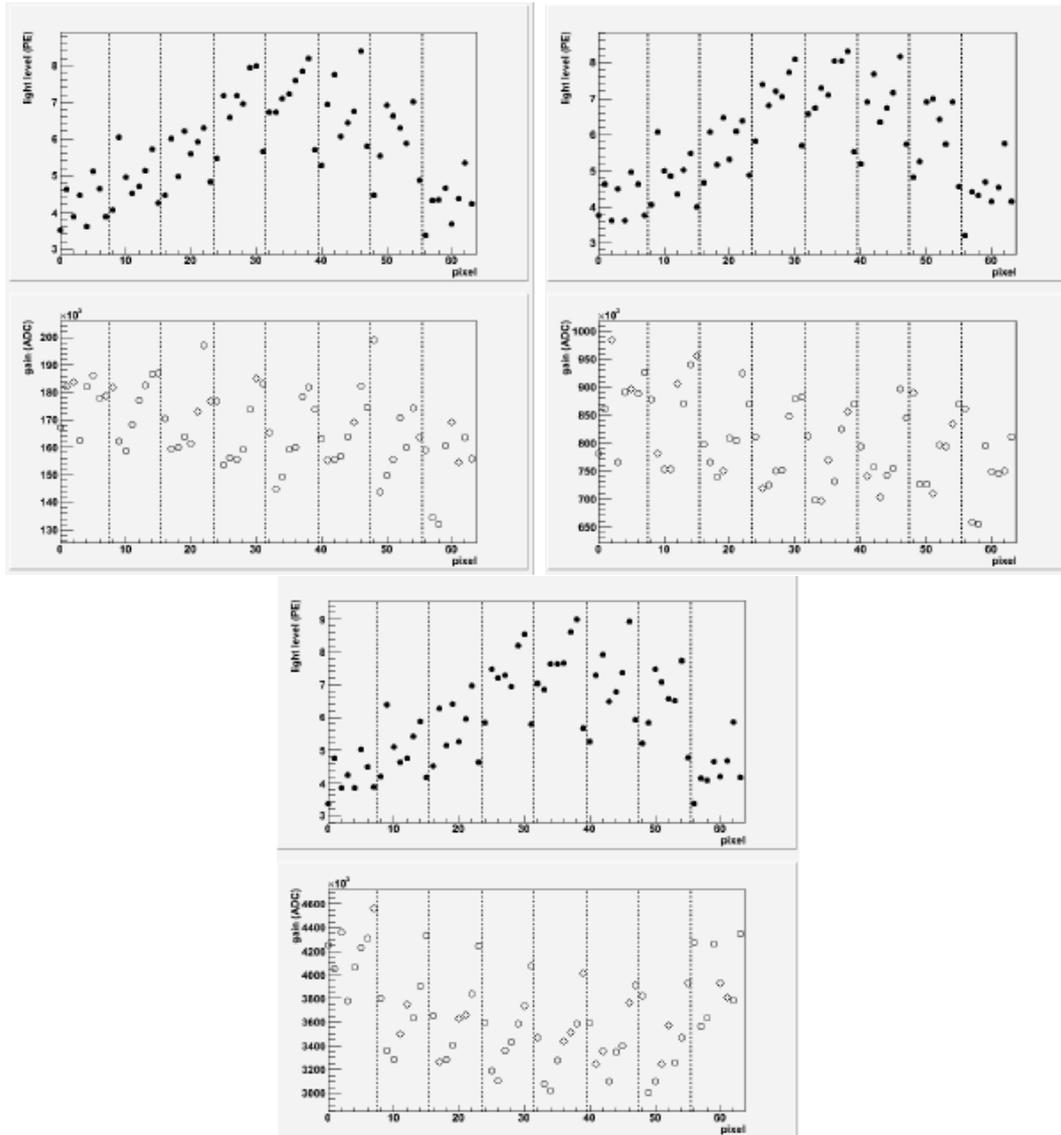


Figura 3.19: Croc 1, Board 3, Chain 1

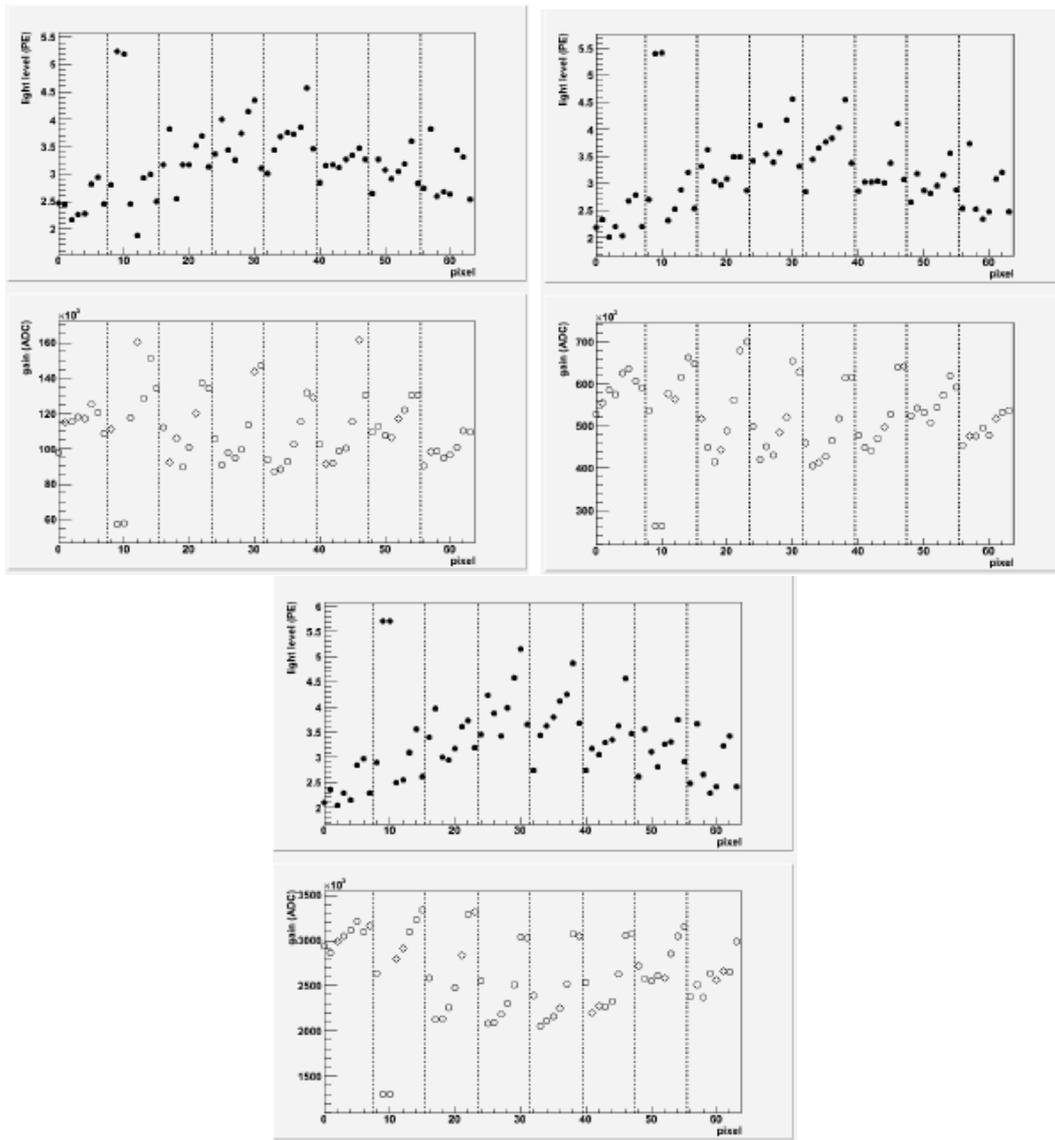


Figura 3.20: Croc 1, Board 6, Chain 5

- El script en ROOT produce, junto con los gráficos, un archivo de texto con la información de dichas ganancias, las cuales son subidas a la base de datos para la calibración de los Digits en el experimento mediante un script en python [58]:

Listing 3.1: Comando para subir las constantes a la base de datos

```
./db_read_write.py -s 1220278860 -e 0 -f GAINCAL -l 0 -r 0
1 -c 0 -b 3 -p 0 -d CALIB -a insert gainfile.dat
```

4.1. Conversión de los datos de RawDigits a IDDigits

Cada tubo fotomultiplicador contiene 64 canales o píxels, los cuales registran una distribución de carga en el detector. En cada plano XU/XV, tenemos 19 tubos conectados en cada centelleador contenido en dicho plano. Los tipos de archivo que contienen los objetos de ésta manera, son llamados *RawDigits* [60].

Así por ejemplo, podemos obtener la distribución de carga del tubo fotomultiplicador 10, píxel 46 del segundo plano, etc. Sin embargo, para realizar el rastreo de trayectorias de partículas, es necesario *convertir* dichos datos, de tal manera que no hablemos de tubos fotomultiplicadores y píxels como dato de referencia para la posición de una partícula, sino de planos, y franjas de centelleador en dichos planos y número de barra de triángulo centelleador (para la descripción de los centelleadores, ver sección 2.5).

Debemos tener en consideración que, cada señal registrada en un PMT, contiene la señal detectada más un pedestal, razón por la cual se toman dos tipos de datos (usualmente a 800V):

- Pedestal Runs: De 5000 eventos, utilizando un generador de pulsos como trigger (cada evento dura $10\mu s$)
- Cosmic Runs: Tomas de datos de 1 hora, usando un Veto Wall como trigger.

¹Todo lo escrito en éste capítulo se refiere al trabajo hecho en el detector Prototipo (Tracking Prototype)

Ambos son almacenados en contenedores RawDigits. Los *Pedestal Runs* pasan por un algoritmo llamado “Pedestal Table(*pedestal_table*)”, el cual extrae información de las Medias de dichos histogramas, así como el RMS de los datos Pedestales, con esto luego, se pasan los *Cosmics* tomados con el detector a través de otro algoritmo llamado “Offline Pedestal Suppression (*OfflinePedSup*)”, el cual suprime las señales pedestales en cada distribución de carga por píxel. Dichos datos ahora contienen la señal pura deseada, y están contenidos en archivos llamados *SuppDigits* (que en el fondo, siguen siendo contenedores RawDigits, pues almacenan la información por píxel).

El tercer paso es pasarlo a través de un último algoritmo, el cual hace uso de Plex² para re-escribir los datos pero por información de número de plano (*PlaneID*), franja (*stripID*), barra de centelleador (*barID*).

Éste archivo se puede visualizar en el software de visualización del experimento (VISTA), o puede ser pasado a través del paquete DSTWriter, para ser convertido en una ntupla, que puede ser vista a través de ROOT. En la figura 4.1, se aprecian los pasos a seguir para la obtención de los IDDigits, y en la figura 4.2, los objetos contenido en cada tipo de archivo.

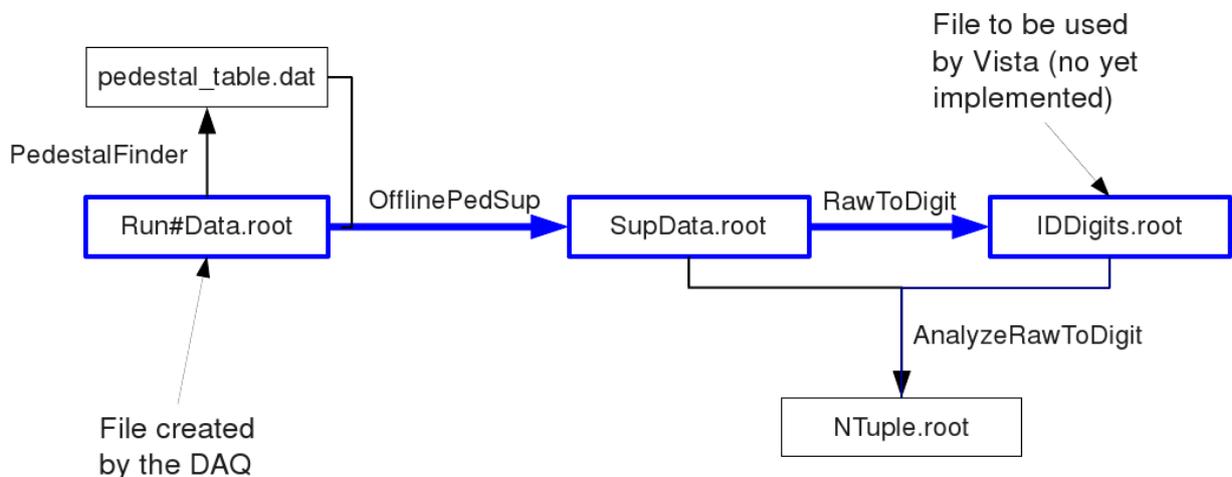


Figura 4.1: Diagrama de algoritmos ejecutados para la obtención de IDDigits

²Plex es un paquete en Gaudi que contiene la información geométrica necesaria para localizar en que lugar ha sido detectado el dato en el detector.

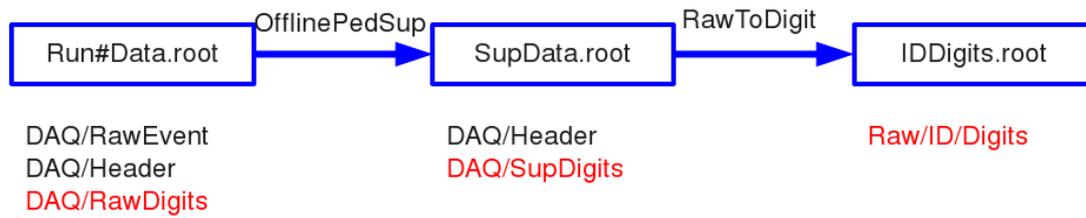


Figura 4.2: Estructura de los tipos de archivo RawDigits, SupDigits y IDDigits, objetos contenidos en cada tipo de archivo

4.2. Búsqueda de Muones utilizando los IDDigits

Una vez creado los IDDigits, es posible -como se mencionó anteriormente- crear ntuplas utilizando el paquete DSTWriter, el cual nos permite analizar los datos mediante ROOT, sin necesidad de utilizar el framework GAUDI. Durante la primera fase del Detector Prototipo, se buscaron los muones haciendo uso de dichas ntuplas.

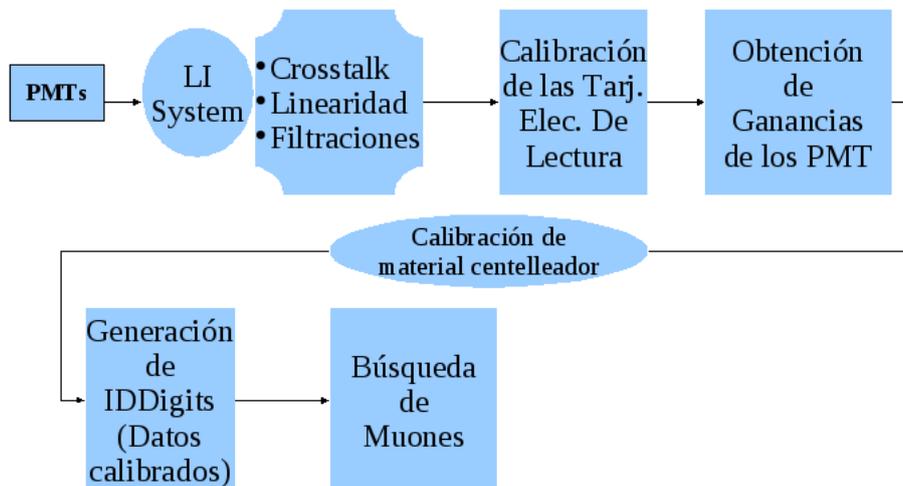
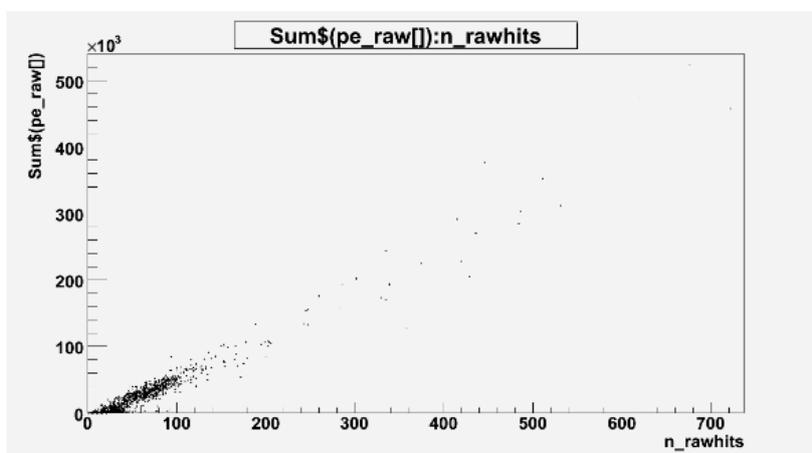


Figura 4.3: Esquema General para la búsqueda de muones desde la toma de datos

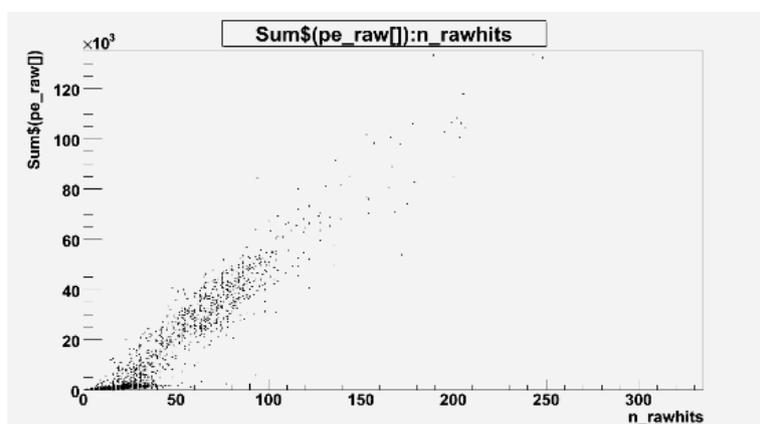
4.2.1. Resultados sin calibración electrónica

El primer paso es filtrar los eventos, basados en el número de hits ($n_rawhits$ en la ntupla) y la energía acumulada ($Sum_pe_raw[]$ ó $Sum_pe_cal[]$, según sean datos calibrados o no³) por cada evento (ver figura 4.4(a)).

El segundo paso es cortar eventos menos energéticos, para lo cual se hace una ampliación del histograma (ver figura 4.4(b)), donde podemos apreciar dos distribuciones, siendo la primera la menos energética (ver figura 4.5).



(a) Energía Total por Evento (sin calibrar) vs Número de Hits



(b) Ampliación de figura 4.4(a)

Figura 4.4: Energía Total por Evento (ADC Counts) vs Número de Hits

³Calibrados se refiere a la calibración de las tarjetas electrónicas de lectura después de aplicar el ajuste tri-lineal

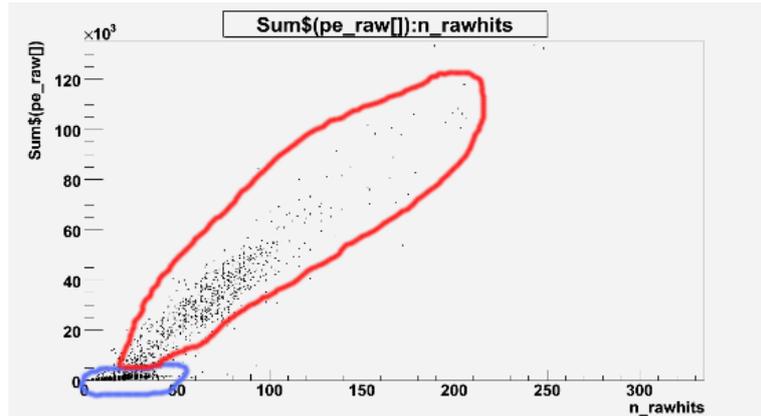


Figura 4.5: Líneas circulares encerrando a las dos distribuciones mostradas en la figura 4.4(b)

4.2.2. Resultados con calibración electrónica

Si se procede exactamente de la misma manera que con los datos sin calibrar, se observa de que el rango utilizado para el filtro mediante energías por evento, se hace constante. Dicho rango varía entre 25 y 130 pC.

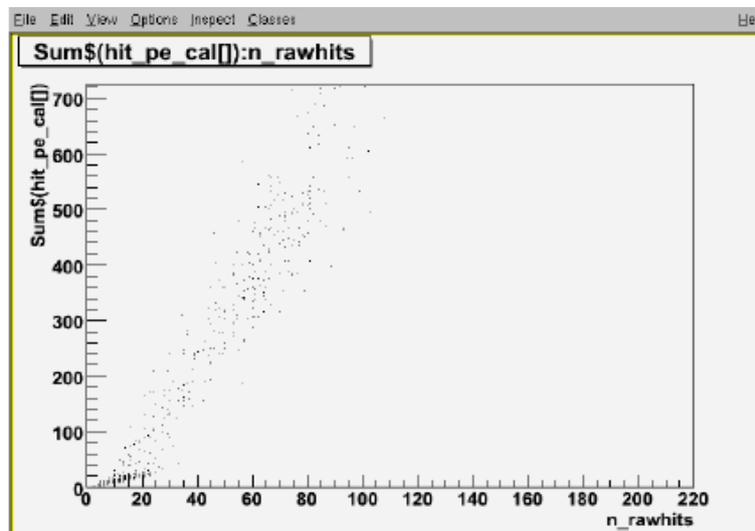
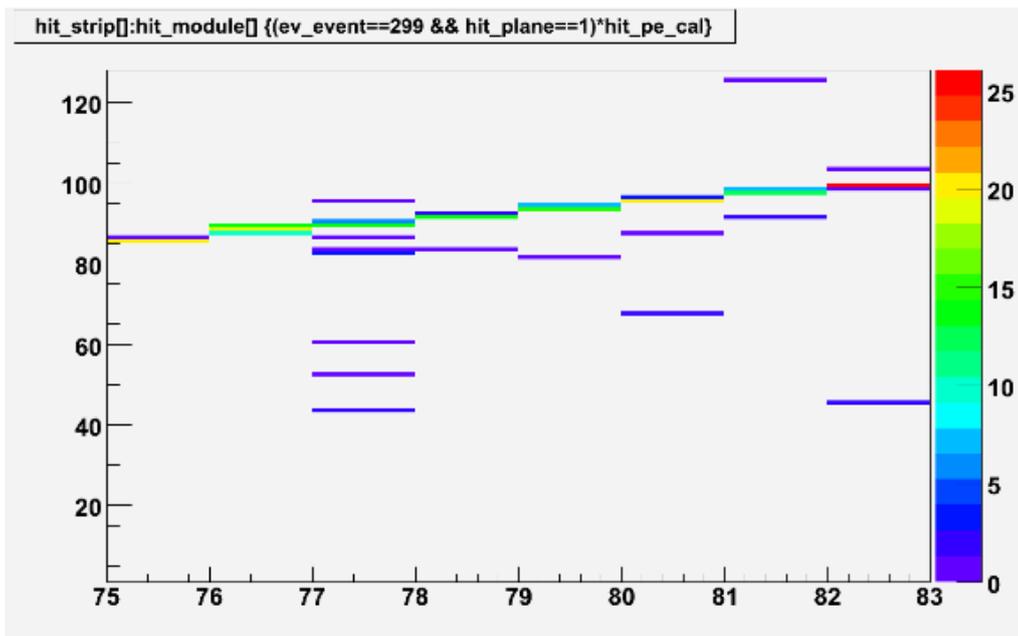
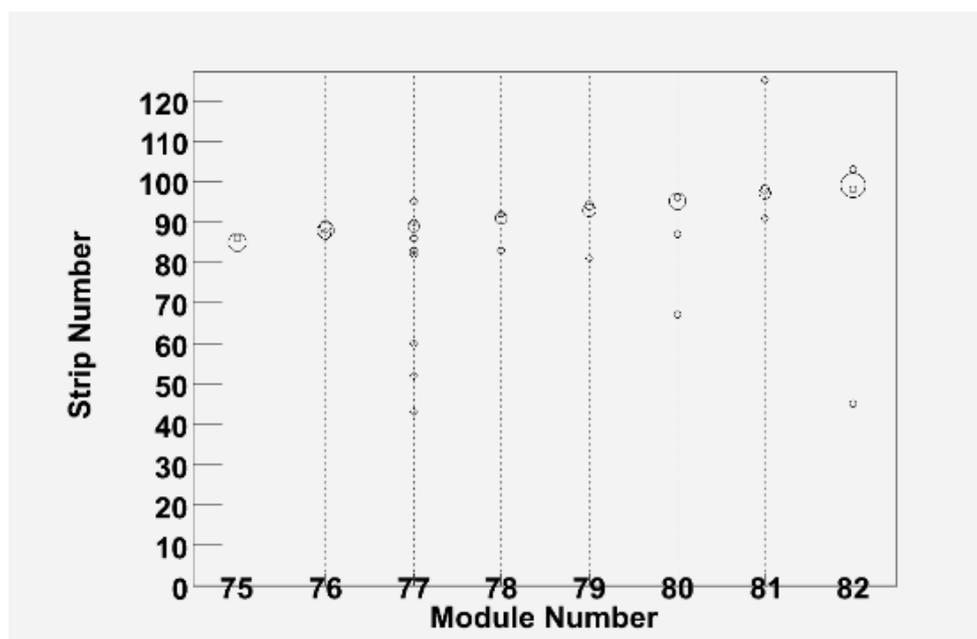


Figura 4.6: Energía Total del Evento (pC) vs Número de Hits

Se muestran algunos resultados de eventos con muones, con datos calibrados [61]:



(a) Muon en evento 299, Run40_Subrun2



(b) Otro método de Graficación

Figura 4.7: Trayectorias de un muon en el Plano X

La razón por la que se pueden ver partículas alrededor de la trayectoria de muones, es debido a distancia respecto respecto vértice, como se muestra en la figura 4.2.2.

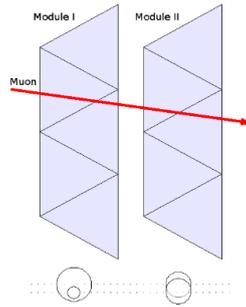


Figura 4.8: Muon a través de 2 módulos

4.3. Software de Visualización de Eventos

Vista es un paquete, parte del Software MINERVA , basado en el paquete Panoramix del software LHC, que a su vez es basado en OpenScientist [29] [30], el cual depende directamente de otros dos paquetes más para su funcionamiento(ver figura 4.3). SoWright, contiene los objetos necesarios que se visualizan en el software, mientras que SoApp define los menús con los objetos a visualizar, así como parámetros de filtrado o métodos de visualización del propio software. Vista es capaz de visualizar IDDigits generados por simulación montecarlo, así como por tomas reales de datos.

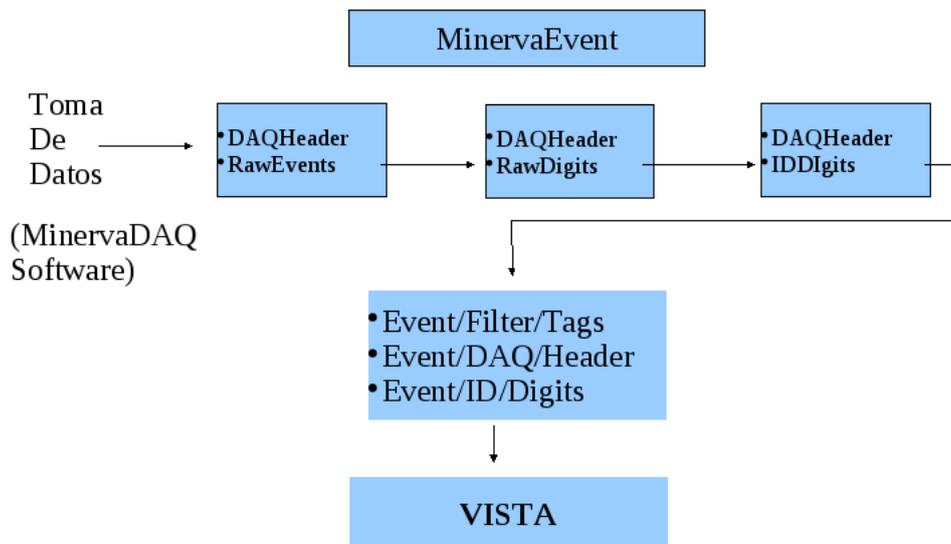


Figura 4.9: Esquema de objetos principales usados desde la toma de datos hasta su visualización en VISTA

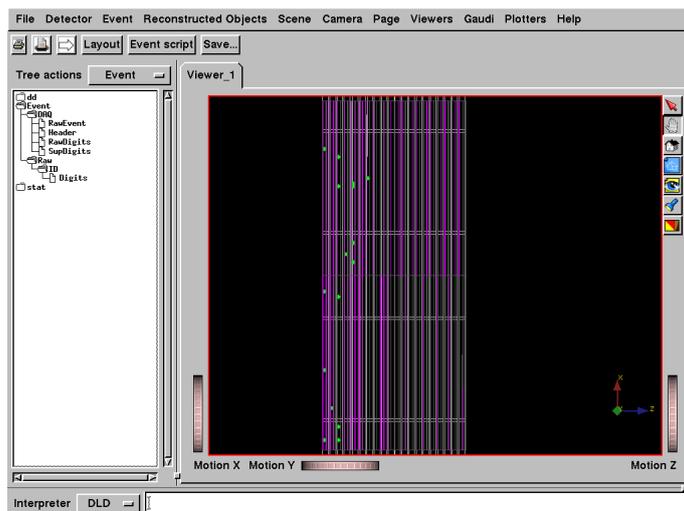
4.3.1. IDDigits en Vista

Para que Vista pueda visualizar correctamente los IDDigits, depende del paquete *Plex*, el cual contiene la información geométrica del detector.

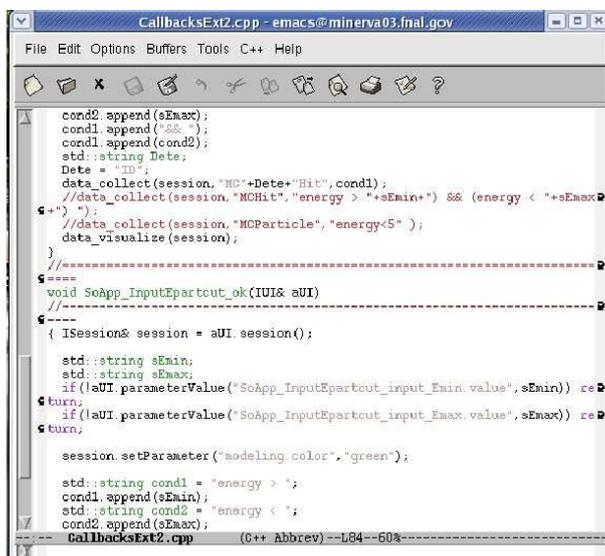
4.3.2. Implementación de Múltiples Vistas para la visualización de las proyecciones de los digits en simultáneo

Actualmente, SoWright reconstruye los IDDigits en 3 objetos, los cuales son las proyecciones del mismo en los planos X,U,V. Se implementó un sistema de visualización múltiple para los tres planos (ver figura 4.10(c)), así como una vista para el objeto tridimensional final (objeto aún en desarrollo).

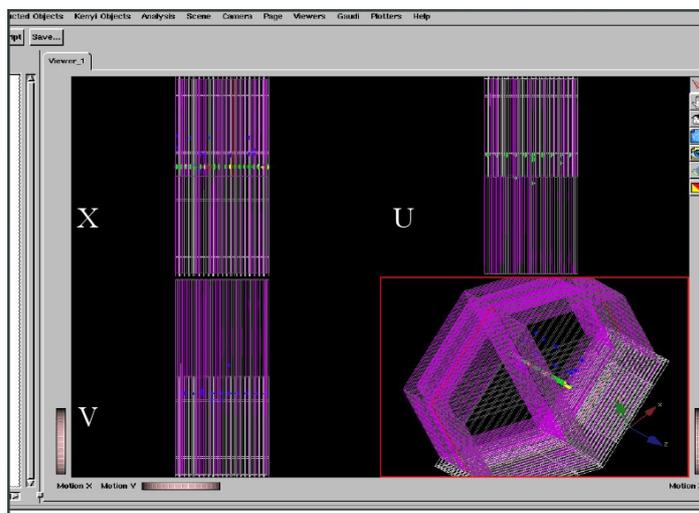
La implementación se hace mediante OnX [31], el cual permite crear interfaces gráficas (GUIs), mediante descripciones en XML depositadas en archivos “.onx” (ver figura 4.10(b)) definiendo los ángulos y posición del detector en el visualizador, archivos que luego son accedidos por el programa mediante los objetos para OnX en Gaudi al momento de correr Vista.



(a) Vista del Plano X del Detector en Vista



(b) Archivo principal de implementación de vistas múltiples en SoApp/Vista



(c) Proyección de los IDDigits en los planos X,U,V y vista tridimensional(no terminado)

Figura 4.10: Visualización de proyecciones en diferentes planos de manera simultánea

4.3.3. Cortes de Energía

Se implementaron cortes por energía depositada y por partícula. Para ello se definieron las funciones de corte a los objetos necesarios (MCHits, MCParticles, IDDigits) en el paquete SoApp, haciendo uso de creación de clases en C++ con objetos específicos de Gaudi y del software MINERνA .

Los cortes fueron implementados para el Inner y Outer Detector, enlazando luego dichas funciones a través de scripts basados en OnX.

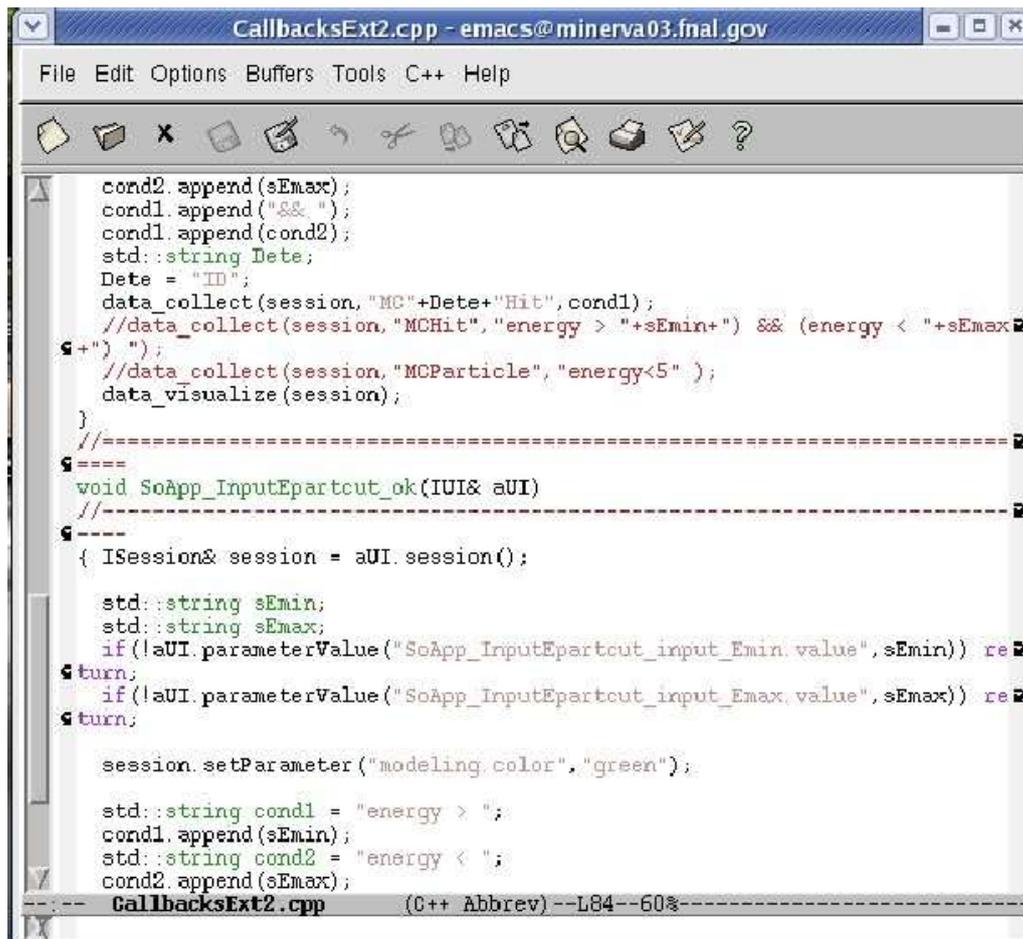
Ejemplo de definición de cortes de Energía por partícula utilizando las propiedades de los objetos de MCParticle:

```
//=====
void SoApp_InputEpartcut_ok(IUI& aUI)
//-----
{ ISession& session = aUI.session();

  std::string sEmin;
  std::string sEmax;
  if(!aUI.parameterValue("SoApp_InputEpartcut_input_Emin.value",sEmin)) return;
  if(!aUI.parameterValue("SoApp_InputEpartcut_input_Emax.value",sEmax)) return;

  session.setParameter("modeling.color","green");

  std::string cond1 = "energy > ";
  cond1.append(sEmin);
  std::string cond2 = "energy < ";
  cond2.append(sEmax);
  cond1.append("&& ");
  cond1.append(cond2);
  data_collect(session,"MCParticle",cond1);
  //data_collect(session,"MCParticle","energy<5" );
  data_visualize(session);
}
```



```
CallbacksExt2.cpp - emacs@minerva03.fnal.gov
File Edit Options Buffers Tools C++ Help

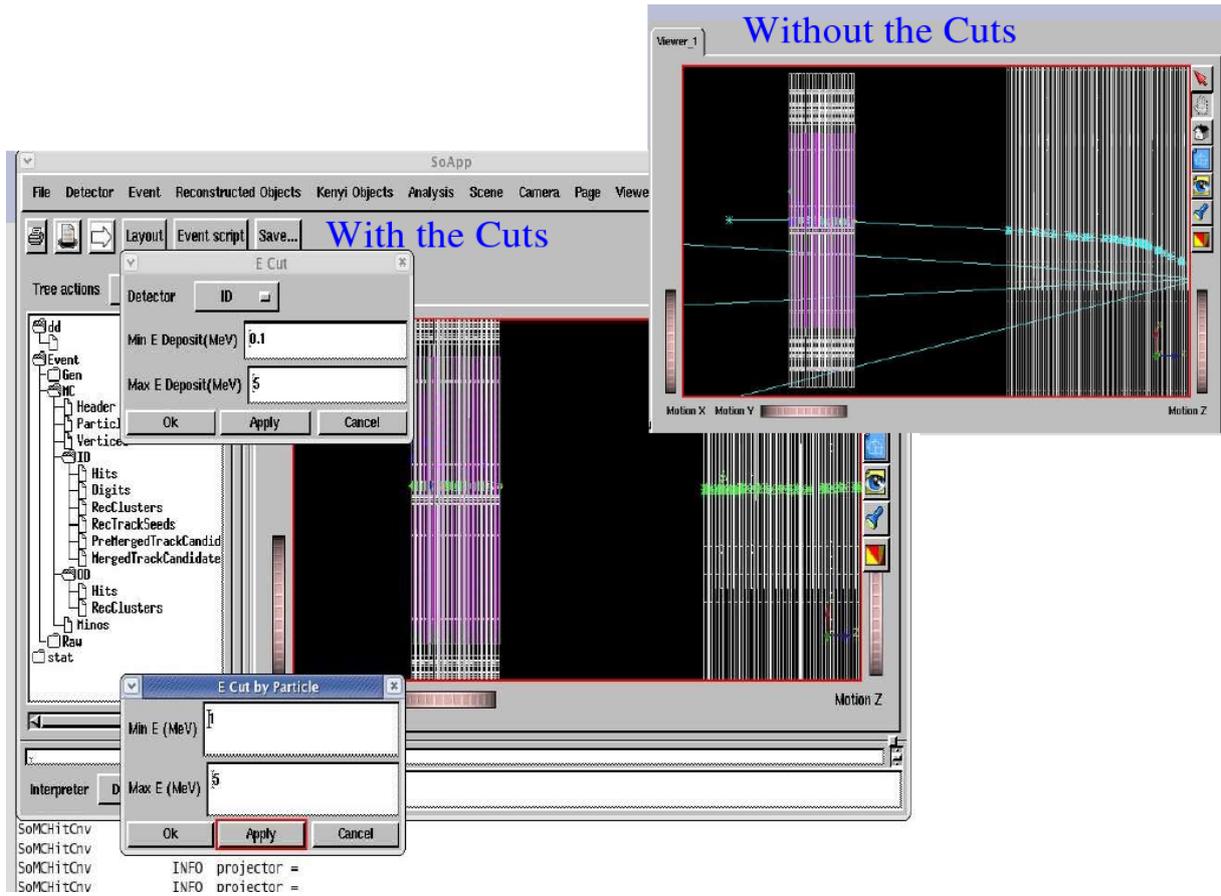
cond2.append(sEMax);
cond1.append("&& ");
cond1.append(cond2);
std::string Dete;
Dete = "ID";
data_collect(session, "MC"+Dete+"Hit", cond1);
//data_collect(session, "MCHit", "energy > "+sEmin+" ) && (energy < "+sEMax+
s+") ");
//data_collect(session, "MCParticle", "energy<5" );
data_visualize(session);
}
//-----
void SoApp_InputEpartcut_ok(IUI& aUI)
//-----
{
  ISession& session = aUI.session();

  std::string sEmin;
  std::string sEmax;
  if(!aUI.parameterValue("SoApp_InputEpartcut_input_Emin.value", sEmin)) re
s turn;
  if(!aUI.parameterValue("SoApp_InputEpartcut_input_Emax.value", sEmax)) re
s turn;

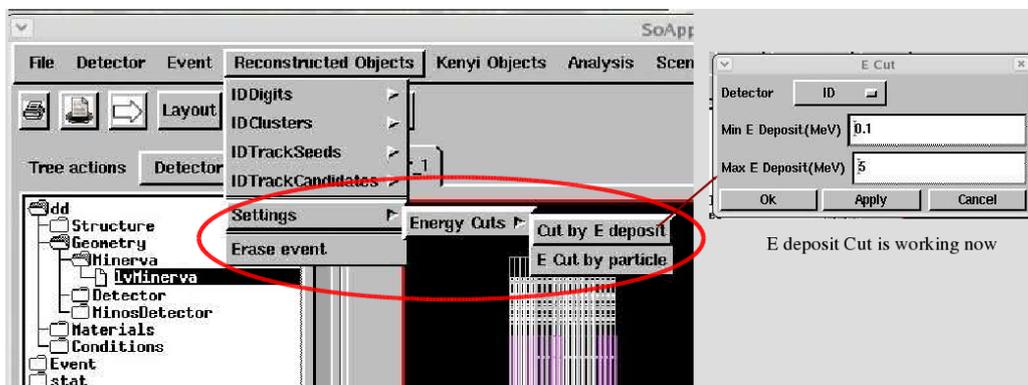
  session.setParameter("modeling.color", "green");

  std::string cond1 = "energy > ";
  cond1.append(sEmin);
  std::string cond2 = "energy < ";
  cond2.append(sEmax);
}
-- CallbacksExt2.cpp (C++ Abbrev) --L84--60%
```

Figura 4.11: Archivo principal de implementación de cortes de energía en SoApp/Vista



(a) Cortes de Energía depositada o por partícula



(b) Menú de Cortes de Energía

Figura 4.12: Cortes de Energía en Vista

4.3.3.1. Manejo de Gaudi y Vista mediante Python

También es posible realizar tareas similares de manera *dinámica*, mediante python, ya que Gaudi dispone de un paquete que permite llamar a los objetos de éste a través del lenguaje mencionado, un ejemplo para adquirir los datos de energía almacenados en el objeto IDDigits e imprimirlos en pantalla o para adquirir datos de energía de partículas Montecarlo y generar

un histograma, serían [62] [63]:

```
appMgr = gaudimodule.AppMgr(outputlevel=3,joboptions="$VISTAROOT/options/MyVista.opts")

evt=appMgr.evtSvc()

appMgr.run(4)
#Adquirimos la energia almacenada en los IDDigits

for part in evt['Raw/ID/Digits']:

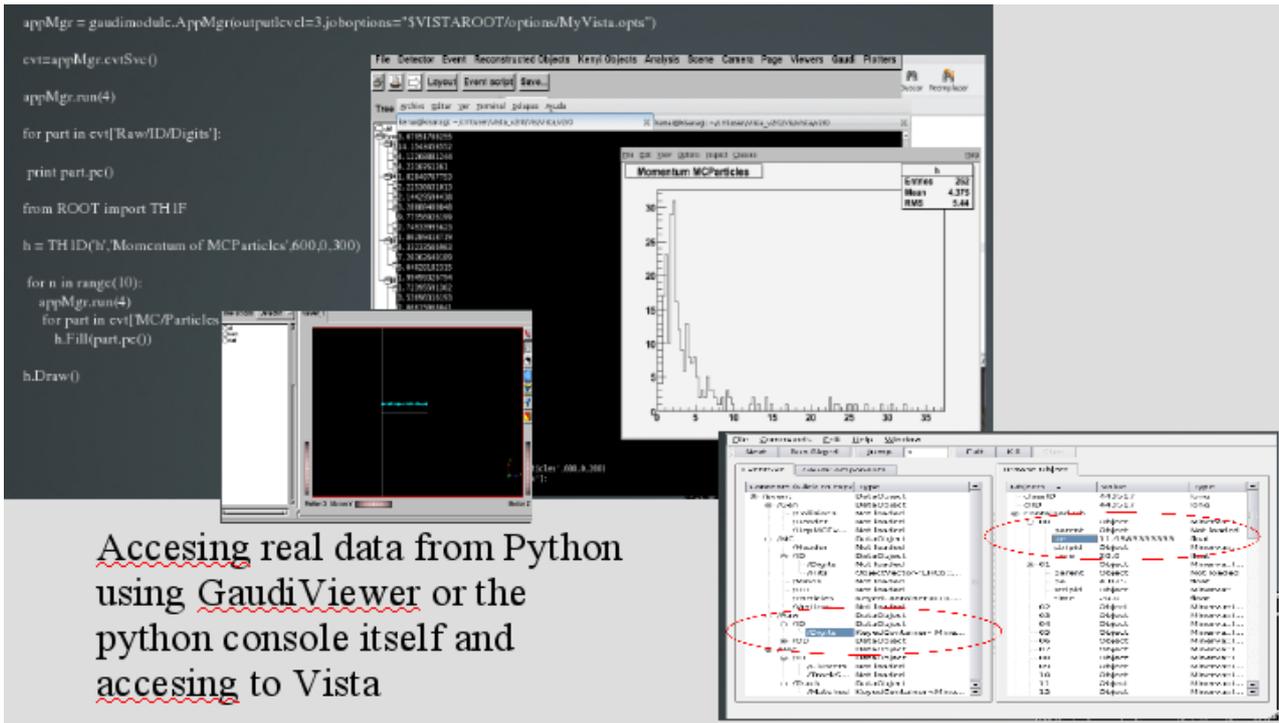
#Imprimimos en pantalla los datos
    print part.pe()

-----
#Creamos un histograma en ROOT
from ROOT import TH1F

h = TH1D('h','Momentum of MCParticles',600,0,300)

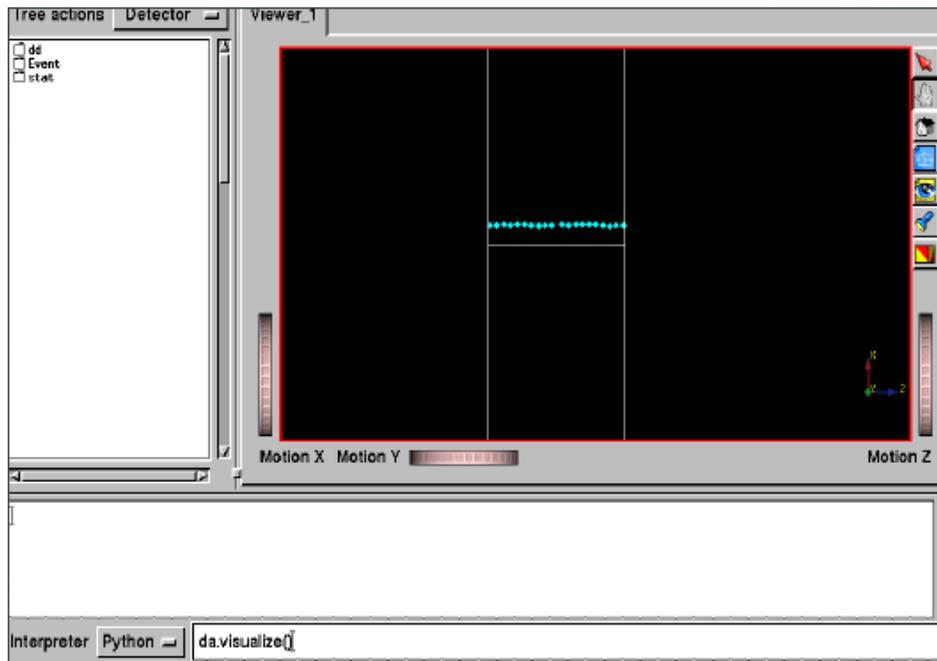
for n in range(10):

    appMgr.run(4)
#Adquirimos datos de particulas Montecarlo
    for part in evt['MC/Particles']:
#Lo llenamos en un histograma
        h.Fill(part.pe())
h.Draw()
```



Accesing real data from Python using GaudiViewer or the python console itself and accesing to Vista

(a) Consola de Python Controlando Vista



(b) Vista de partículas en Plano X usando Python

Figura 4.13: Manero de Vista a través de Python

4.4. Implementación del Objeto FilterTagList para la etiquetación de diferentes eventos físicos en los datos tomados en el paquete de definición de objetos del software MINERνA (MinervaEvent)

MinervaEvent, es el paquete en donde se encuentran las definiciones de los objetos, clases y Tools necesarios para la toma de datos en el detector MINERνA . La definición de objetos se realiza mediante archivos xml, que luego son convertidos mediante *gccxml*⁴ a los objetos usados por Gaudi.

El objeto FilterTagList, sirve para etiquetar eventos físicos, aunque para esto necesita de otros algoritmos que diferencien dichos eventos. Actualmente el objeto solo dispone de las etiquetas para eventos con muones [64]

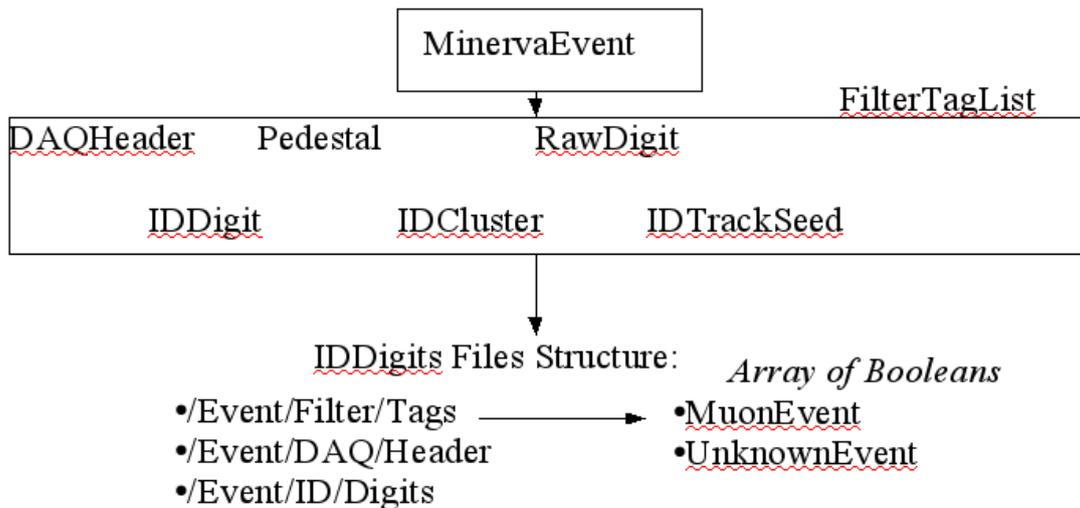


Figura 4.14: Esquema de Objetos en MinervaEvent

⁴GCCXML es un analizador sintáctico de C++, que nos permite crear cabeceras de objetos desde archivos XML

4.5. FilterEvent Software

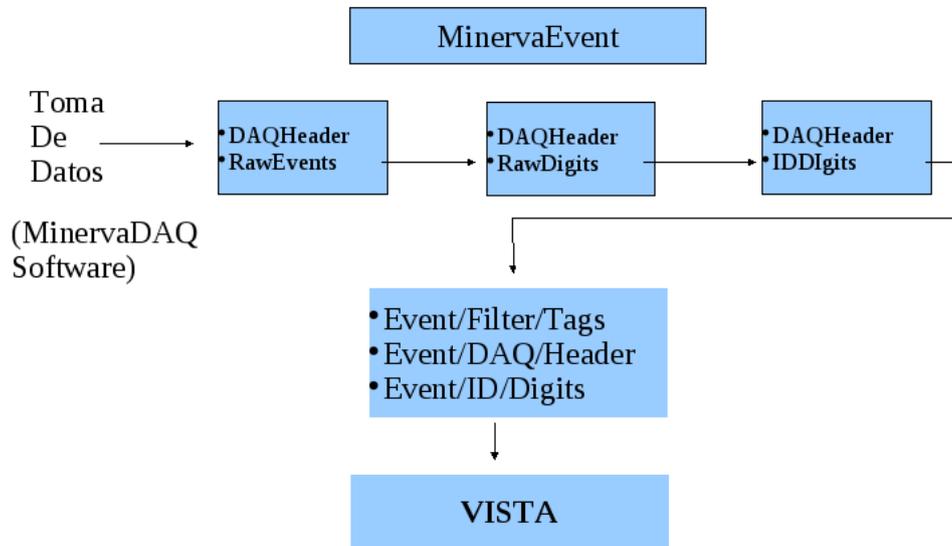


Figura 4.15: Esquema de Objetos en MinervaEvent

Éste paquete fue creado para contener los algoritmos de reconocimiento de eventos físicos al momento de tomar los datos.

Por el momento solo contiene un algoritmo simple de reconocimiento de muones, basado en el mismo criterio usado en la sección 4.1, sin embargo el software apunta al desarrollo de otros tipos de algoritmo para filtrar, por ejemplo, eventos que han tenido problemas al momento de la adquisición de datos (por errores reportados por el DAQ, o por si alguno de los PMT/FEBS registra un voltaje de entrada de más de 10V de diferencia con respecto al valor nominal), así como diferentes tipos de eventos más asociados con el haz de neutrinos (eventos quasi-elásticos, profundamente inelásticos, producción por resonancia, etc) una vez se cuente con el haz de neutrinos para la toma de datos.

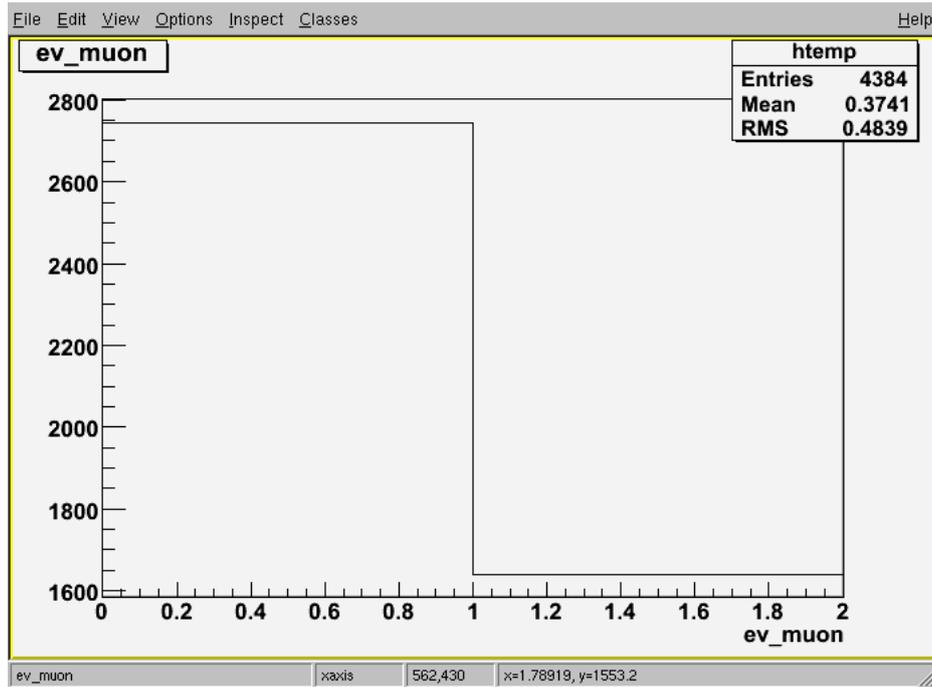


Figura 4.16: Histograma mostrando el número de eventos con muones encontrados

Resultados y Conclusiones

Se consiguió finalizar el ensamblaje del Detector Prototipo de MINER ν A , y hacer análisis principalmente en las partes de hardware y software asociados a la reconstrucción de trayectorias en el *Tracking Prototype*.

En cuanto a hardware, se hicieron estudios que garantizaron el correcto funcionamiento de los tubos fotomultiplicadores junto con sus tarjetas electrónicas de lectura, analizando las variaciones que pueden ocurrir ya sea por condiciones climáticas (variación de temperatura), filtraciones de luz, crosstalk en las tarjetas electrónicas de lectura, así como por estabilidad o cambio de parámetros en el software de adquisición de datos.

Realizamos el cableado de las fibras ópticas que van desde los PMTs hacia los centelleadores en los módulos del detector, verificando que funcionen adecuadamente, permitiendo relacionar las direcciones electrónicas (CRIM, CROC, FEB, Board) de cada tubo fotomultiplicador con la posición geometría de la zona sensible en el detector.

Desarrollamos un algoritmo en el sistema de adquisición de datos, que permiten hacer uso de los discriminadores de los FEBs para localizar más rápidamente tubos fotomultiplicadores que puedan tener filtraciones de luz, así como pixels muertos

Calibramos y medimos las ganancias de los PMTs, almacenando los datos en una base de datos, proceso necesario para la reconstrucción de energía depositada en las zonas sensibles del detector posteriormente.

Se crearon scripts de ejecución de los algoritmos necesarios para la conversión de los objetos *RawDigits* a *IDDigits*, necesario para tener la información, no solo de energía y tiempo, sino también de la posición geométrica en el detector de los datos que se toman.

Se implementó diferentes vistas por proyección en los planos (X,U,V) para la visualización de los *IDDigits* en el software de visualización de eventos VISTA.

Se añadieron opciones de cortes de energía por partículas o energía depositada en el detector.

Se desarrolló el paquete *FilterEvent*, con un algoritmo básico de detección de eventos con muones, utilizado para obtener las trayectorias de dichas partículas en el detector, permitiendo depurar la parte de mapeado del software (la parte que se encarga de relacionar las medidas tomadas en los PMTs con su correspondiente posición geométrica en el detector), dejando camino para el desarrollo de otros algoritmos más adelante, como para la detección y filtrado de eventos en donde el hardware ha tenido problemas, algoritmos de reconocimiento de eventos quasi-elásticos, profundamente inelásticos, muones que se generen por la interacción de los neutrinos provenientes del haz NuMI con las rocas en las instalaciones donde se encuentre el detector (rock muons), etc.

Se trabajó en Python con GAUDI, para trabajar en VISTA de manera más dinámica, permitiéndonos acceder a los objetos de Gaudi y volcar su información (energía, tiempo y posición geométrica de los datos) en pantalla o incluso utilizarlo con ROOT para crear histogramas al vuelo, tanto de los datos simulados como los reales, sin necesidad utilizar algoritmos especiales para generar Ntuplas compatibles con ROOT.

Adicionalmente, se publicará un artículo relacionado con éste trabajo en el REVCUNI de la Facultad de Ciencia de la Uni, así como un proceeding para el *XI HADRON PHYSICS*, en Brasil, marzo de 2010.

Todo este trabajo ha servido para construir un número significativo de los módulos del detector MINER ν A, calibrar y mostrar su rendimiento para el rastreo de partículas de rayos cósmicos, además permitió verificar la operación y funcionamiento del detector antes del montaje final para el detector completo, entre ellos, pruebas de hardware, PMTs, y estabilidad del software de adquisición de datos.

Introducción al Software MINERvA

El software MINERvA , recopila todos los paquetes desarrollados internamente para operar, simular, reconstruir, analizar y visualizar los datos de MINERvA . El proyecto incluye muchos paquetes externos, tales como ROOT, Python, AIDA, gccxml, GEANT4, MySQL, etc, que proveen servicios que no se desean reinventar [41].

El Software MINERvA utiliza el framework GAUDI [43], el cual provee un punto común de referencia para el desarrollo de aplicaciones, además de permitir la reutilización e intercambio de componentes de software individuales.

El Framework utilizado depende principalmente de otros paquetes como ROOT [44], AIDA [45], SEAL [46], POOL, y otros, así como algunas clases heredadas del LHCb Software. El manejo del desarrollo de paquetes y control de dependencias de los mismos se hace a través de CMT (Configuration Management Tools), el cual soporta ambientes de trabajo tanto para Linux, como Windows.

Adquiriendo el Software

El código se mantiene via CVS (Concurrent Versioning System) [47] a través de repositorios que pueden ser accesados como lectura incluso anónimamente y pueden ser compilados. Adicionalmente, se liberan versiones ya compiladas del software cada cierto tiempo, disponibles a través del sistema de ficheros distribuido AFS [42], para lo cual es necesario tener una cuenta

kerberos [48] con los permisos necesarios para el acceso al Software.

A.1. Repositorios CVS

El acceso a los repositorios se puede hacer de 2 maneras:

- Utilizando Kerberos:

```
kinit
export CVSROOT="minervacvs@cdcvs.fnal.gov:/cvs/minervasw"
export CVS_RSH=ssh
```

- Accesando anónimamente:

```
export CVSROOT=":pserver:anonymous@cdcvs.fnal.gov:/cvs/minervasw"
```

Para la compilación e instalación del Software, se deben descargar y ejecutar primero los scripts necesarios, por ejemplo para la versión v6r2:

```
cd $HOME/<directorio local de instalacion deseado>
cvs -d minervacvs@cdcvs.fnal.gov:/cvs/minervasw co -d v6r2 -r v6r2 Bootstrap
cd v6r2
source install/install.sh > install.log
```

Una vez hecho ésto, el script de instalación realizará una serie de preguntas para la instalación, como confirmar la versión deseada, ya sea estable o de desarrollo, la ruta de instalación, si se desea instalar el framework GAUDI, etc. Todos éstos tienen ya un valor por defecto.

```
* Which version of the Minerva Software do you wish to install? [v6r2]:
* Do you wish to install a particular development tag? Default is main release. :
* Full path of the location to do the installation [$PWD]:
* Do you wish to do a full install of LHCb Software and the GAUDI framework? [y]:
* Do you wish to install the MINERVA Software components [y]:
* Installation platform (CMTCONFIG) ...? [slc4_ia32_gcc34]:
```

En caso seleccionar que no se desea instalar GAUDI y LHCb software, el script asume que éstos ya se encuentran instalados en algún lado, y solicita ingresar las rutas de dichos paquetes:

- CMT
- LCG

- LHCb
- LHCb scripts

```
* Please specify the location where CMT software is installed [/afs/fnal.gov/files...] :
* Please specify the location where LCG external software is installed [/afs/...] :
* Please specify the location where LHCb software is installed [/afs/...] :
* Please specify the location where LHCb scripts are located [/afs/fn...] :
```

Luego de esto, el script inicia el proceso de descarga, compilación e instalación del software. Para comprobar que el proceso ha terminado sin errores, la forma más rápida es mirando en los archivos `mkproject.log_*`:

```
cd minerva
grep finished mkproject.log_*

mkproject.log_Endavor_v5r1:...compilation is finished
mkproject.log_EXTERNAL_v5r1:...compilation is finished
mkproject.log_Mdbase_v5r1:...compilation is finished
mkproject.log_Mega_v5r1:...compilation is finished
mkproject.log_Minerva_v5r1:...compilation is finished
mkproject.log_Owl_v5r1:...compilation is finished
mkproject.log_SoVis_v5r1:...compilation is finished
mkproject.log_Vista_v5r1:...compilation is finished
mkproject.log_Weber_v5r1:...compilation is finished

grep fail mkproject.log_*
```

Una vez instalado, sólo hace falta el script `setup.sh`, para configurar las variables de entorno del software:

```
cd ~/<directorio de instalación>/v6r2
source setup.sh
```

A.2. Utilizando el AFS

Es posible acceder al software sin necesidad de instalarlo localmente, pero para esto es necesario una cuenta kerberos con los privilegios necesarios y el acceso remoto a una máquina del proyecto MINERVA. Éste es el proceso más utilizado y práctico dentro del propio Fermilab:

```
kinit usuario
ssh minerva01.fnal.gov
cd /afs/fnal.gov/files/code/minerva/release/v6r2
source setup.sh
```

A.3. Utilizando el software en CTIC-UNI

Actualmente, mediante NFS, se ha montado una ruta AFS, en algunas de las máquinas disponibles en una de las oficinas del CTIC, utilizadas por el grupo de estudiantes de Altas Energías de la facultad de ciencias, por lo cual es posible interactuar con el software desde la UNI sin privilegios de kerberos ni instalaciones locales del software entero.

El servidor utiliza Scientific Linux 4.x, S.O utilizado en Fermilab. Para utilizar el software se puede acceder via SSH a la máquina, o es posible acceder también localmente montando el directorio de instalación en la máquina local deseada. Para esto, se necesita tener NFS-client instalado.

Scientific Linux utiliza NFS-client por default, en CTIC, las máquinas disponibles vienen mayoritariamente con Ubuntu-Desktop, por lo que allí hay que instalar el software:

```
sudo apt-get install nfs-common
```

Después creamos un directorio de montaje y procedemos a montar la partición del servidor:

```
sudo mkdir /afs
sudo mount -t nfs voms.grid.ctic.uni.edu.pe:/afs /afs
```

Nota: voms.grid.ctic.uni.edu.pe tiene la IP: 192.168.10.149

Otro método es utilizando el fstab del sistema para montar la partición automáticamente cada vez que se carga el Sistema: Editamos nuestro fichero `/etc/fstab`, agregando lo siguiente:

```
192.168.10.149:/afs /afs nfs defaults 0 0
```

Ejecutamos como root en SL4.x, o usando SUDO en Ubuntu:

```
sudo mount -a
```

Y para configurar las variables de entorno del software:

```
cd /afs/fnal.gov/files/code/minerva/release/v6r2
source setup.sh
```

```
=====
* *
* Welcome to the MINERVA Software Framework *
* *
=====

=====
Configuration Summary
-----
* Release : v6r2
* Install area : /afs/fnal.gov/files/code/minerva/release/v6r2/
* Platform : slc4_ia32_gcc34
=====

*****
* WELCOME to the LHCb software *
*****
--- $SITEROOT is set to /afs/fnal.gov/files/code/minerva/release/v6r2/
--- $CMTSITE is set to LOCAL
--- $CMTROOT is set to /afs/fnal.gov/files/./CMT/v1r20p20070208
--- $CMTCONFIG is set to slc4_ia32_gcc34
--- to compile and link in debug mode : export CMTCONFIG=$CMTDEB ; gmake
--- $LHCBPROJECTPATH is set to $LHCb_release_area:$LCG_release_area
--- projects will be searched in $CMTPROJECTPATH set by the user
-----
```

Utilizando los Paquetes

Para utilizar los paquetes, se recomienda crear un entorno local CMTUSER, para lo cual hacemos:

```
cd $HOME
mkdir cmtuser #(Si es que no existe, pero probablemente el setup.sh lo cree)
export User_release_area="$HOME/cmtuser"
```

Si deseamos el paquete VISTA por ejemplo:

```
cd $User_release_area
setenvVista
```

you must give a project version:

Please enter your choice (v6r2 q[uit] [v6r2]):

```
-----
--- Vista environment is set for version v6r2
--- packages will be searched for in CMTPROJECTPATH:
/afs/fnal.gov/files/code/minerva/release/v6r2//minerva:../lcg/external
-----
```

```
-----
Configuring environment for project Vista_v6r2
CMT version v1r20p20070208.
```

Si estamos usando ubuntu, necesitaremos las bibliotecas libstdc++5:

```
sudo apt-get install libstdc++5
```

Después, copiamos el código, entramos al directorio local de configuración y configuramos el Paquete para trabajar localmente:

```
cp -r $SITEROOT/minerva/VISTA/VISTA_v6r2/* .
cd Vis/Vista/v6r2/cmt
cmt config
```

```
Removing all previous make fragments from cmt
Removing all previous make fragments from slc4_ia32_gcc34
Creating setup scripts.
Creating cleanup scripts.
applications directory already installed
cmt directory already installed
doc directory already installed
examples directory already installed
options directory already installed
scripts directory already installed
```

Después, definimos las variables de entorno del paquete:

```
source setup.sh
```

Introducción a los paquetes usados del software MINERvA

MinervaDDDB y MinervaSQLDDDB

Éstos paquetes contienen la descripción geométrica del detector, dicha descripción se realiza de dos formas diferentes, mediante una base de dato mySQL (usando MinervaSQLDDDB), o mediante archivos XML (MinervaDDDB) que encapsulan la misma información. Debido a que la tecnología usada para la descripción del detector está escondida para el usuario final, la elección del método no implica ningún cambio en el código cliente. Basta con especificar la opción y la geometría deseada en un archivo de configuración, de la siguiente manera:

```
//Para utilizar la geometría del detector
// Tracking Prototype-Cosmic Rays desde MinervaSQLDDDB
//DetectorDataSvc.DetDbLocation = "conddb:/minerva-TP-CR-200708.xml";

// Para usar la misma geometría desde MinervaDDDB
DetectorDataSvc.DetDbLocation = "$MINERVADDDDBROOT/DDDB/minerva-TP-CR-200708.xml";
```

En donde *minerva-TP-CR-200708.xml* es la geometría deseada a usar. Actualmente, las geometrías disponibles son:

- Para el detector Completo:

```
minerva-200708.xml
minerva-200811.xml
```

minerva-200903.xml

- Para el detector en su fase Tracking Prototype:

minerva-TP-200708.xml

minerva-TP-CR-200708.xml

minerva-TP-CR-200903.xml

Plex y MinervaXMLConditions

Plex, se encarga de la interface para el mapeo de los elementos del detector MINERVA a los canales de la electrónica y viceversa. El mapeo se lee desde el paquete MinervaXMLConditions, la cual organiza las definiciones del mapeo en formato XML, para las diferentes fases en las que se encuentre el detector. Existen condiciones para el detector completo (módulo 1 al 114), así como para el detector en su fase llamada Tracking Prototype, que contiene los módulos del 75 al 98.

Éstos paquetes son usados cada vez que se quiere asociar las señales de lectura electrónica en los PMTs a hits en los módulos del detector.

El paquete no requiere configuraciones, pero es una dependencia importante para otros paquetes.

PedestalSuppresion

Éste paquete sirve para encontrar los pedestales de señales tomadas en los PMTs y luego suprimirlos, dejando solo la señal en sí. Su utilización básica se basa en la edición de dos archivos de configuración, para definir los archivos de entrada y salida de los algoritmos:

PedestalFinder.opts:

```
//-----  
// Input File and Event Selection  
//-----
```

```
EventSelector.Input    = {"DATAFILE='PFN:Run317Data.root'
TYP='POOL_ROOTTREE' OPT='READ'" };
EventSelector.PrintFreq = 100;
//PedestalFinder.OutFileName = "pedestal_table.dat";
```

La línea *EventSelector.Input* define el archivo de datos a analizar. Adicionalmente, se puede especificar el archivo de salida con *PedestalFinder.OutFileName* si se desea. Un ejemplo del archivo que produce éste algoritmo se puede ver en el cuadro 3.4.

OfflinePedSup.opts:

```
// Input File and Event Selection
//-----

EventSelector.Input = {"DATA='PFN:Run317Data.root'
TYP='POOL_ROOTTREE' OPT='READ'"};

EventSelector.PrintFreq = 100;

EventDataSvc.ForceLeaves = true;
EventDataSvc.RootCLID    = 1;

ApplicationMgr.EvtMax    = 2000;
ToolSvc.Minerva::PedSupTool.InputFile="pedestal_table.dat";

//-----
// Output File
//-----

ApplicationMgr.OutStream = {"SuppressedData"};
SuppressedData.ItemList=
    {"Event/DAQ/SupDigits#999", "Event/DAQ/Header#999"};
SuppressedData.Output =
    "DATAFILE='PFN:SuppressedData.root' TYP='POOL_ROOTTREE' OPT='RECREATE'";
```

Como en el caso anterior, *EventSelector.Input* define el archivo de entrada de datos, si cambiamos el nombre del archivo de salida del algoritmo anterior, necesitamos definirlo aquí, con *ToolSvc.Minerva::PedSupTool.InputFile*. *ApplicationMgr.EvtMax* define el número máximo

de eventos a procesar, se puede usar **-1** para infinito.

Nuestro algoritmo creará un `POOL_ROOTTREE`, que sería como una tupla en forma de árbol en un archivo `Gaudi-ROOT`, por lo que `SuppressedData.ItemList` define el tipo de objetos que se desean almacenar en el archivo de salida, en éste caso la cabecera de los datos, y los datos en sí.

El nombre de dicho archivo se especifica con `SuppressedData.Output`.

RawToDigit

Éste paquete convierte `RawDigits` en `IDDigits 4`, es decir, utiliza el paquete `Plex` para transformar las lecturas en forma de canales electrónicos al mapeo en el detector, aplicando algoritmos de calibración, permitiendonos pasar de carga en unidades digitales, a Coulombs.

En el experimento, primero se filtran los objetos `RawDigits` a otros llamados `SupDigits`, que son los los mismos objetos `RawDigits` con las lecturas de pedestales suprimidas mediante el paquete `PedestalSuppresion`, y luego dichos `SupDigits` se convierten a `IDDigits`.

Para convertir objetos `RawDigits/SupDigits` a `IDDigits`, las opciones principales a configurar en `RawToDigit.opts` son:

```
// Definimos la geometria a usar
DetectorDataSvc.DetDbLocation =
"$MINERVADDDDBROOT/DDDB/minerva-TP-CR-200708.xml";

//Archivo de Entrada
EventSelector.Input    = {"DATAFILE='PFN:Run148Data.root'
TYP='POOL_ROOTTREE' OPT='READ'" };

//Numero de Eventos a procesar
// usar -1 para procesar todos.
ApplicationMgr.EvtMax    = 1000;

//-----
// Archivo de Salida
//-----
//Definimos algoritmo que escribira el archivo
```

```

ApplicationMgr.OutputStream = { "RawToDigit" };
//Seleccionamos los objetos y las rutas que queremos que se creen
//En este caso, la cabecera de la toma de datos
//y los IDDigits.
RawToDigit.ItemList = {/Event/DAQ#1"
, /Event/DAQ/Header#1"
, "/Event/Raw#1"
, "/Event/Raw/ID#1"
, "/Event/Raw/ID/Digits#1"};

//-----
// Nombre de archivo de salida
//-----
RawToDigit.Output = "DATAFILE='RawToDigit.root'
TYP='POOL_ROOTTREE' OPT='RECREATE'";

```

DSTWriter

DSTWriter contiene un algoritmo que crea una ntupla de datos MonteCarlo y/o Datos Reconstruidos desde el Gaudi TES (Transient Event Store) para análisis. El uso es similar el de RawToDigits, editamos el archivo DSTWriter.opts

```

//Definimos archivo de Entrada
EventSelector.Input =
{"DATAFILE='PFN:TP_0810301314_RawDigits.root' TYP='POOL_ROOTTREE' OPT='READ' " };
// Definimos archivo de Salida
//-----
NTupleSvc.Output = {"FILE1 DATAFILE='TP__cosmc_v01_DST.root' TYP='ROOT' OPT='NEW'"};
//-----
// Establecemos los objetos
// que desamos que se escriban
//-----

DSTWriterAlg.WriteDAQHeader = true;
DSTWriterAlg.WriteRawDigits = true;
DSTWriterAlg.WriteIDDigits = false;
DSTWriterAlg.WriteODDigits = false;
DSTWriterAlg.WriteClusters = false;

```

```
DSTWriterAlg.WriteTracks    = false;
//Indicamos el tipo de objeto que vamos a leer, para los datos tomados
//pueden ser "DAQ/RawDigits" o "DAQ/SupDigits"
DSTWriterAlg.RawDigitLocation = "DAQ/SupDigits";
```

ForestTest

Al igual que DSTWriter, puede de crear una ntupla de los datos, pero también un histograma en ROOT para el análisis de datos. Consta de 2 algoritmos: BasicHistosAlg y BasicNtuplesAlg.

El algoritmo de BasicNTuples se ha reemplazado por las generadas por DSTWriter, pero la creación de histogramas aún se utiliza, debido a que la creación de éstas consume menos recursos y espacio en el buffer, lo cual constituye una ventaja de BasicHistosAlg, ya que crear archivos para análisis *online* con los datos, de manera automática, además tiene opciones para realizar la calibración de FEBs y generar histogramas de los datos después de haber pasado por los Discriminadores incorporados en los FEBs.

Para activar o desactivar dichas opciones, modifica el archivo BasicHistos.opts

```
//Archivo de Entrada
EventSelector.Input    = {"DATAFILE='PFN:Run1235Data.root'   TYP='POOL_ROOTTREE'  OPT='REALTIME'}

//Numero de Eventos a Procesar
ApplicationMgr.EvtMax    = 1000;

//Ruta de los Objetos a procesar
BasicHistos.DigitLocation="DAQ/SupDigits";
BasicNtuples.DigitLocation="DAQ/SupDigits";

//Especifica si se desea producir Ntuplas o no
BasicNtuples.NTupleProduce = false;
BasicNtuples.PreScale = 10;

//Definimos si deseamos hacer
//calibracion de FEBs y/o
//crear histogramas de las señales
//despues de pasar por los discriminadores
BasicHistos.Do_FEB_Cal = true;
```

```
BasicHistos.Do_Disc_Plots = true;
```

```
//-----  
// Archivo de Salida  
//-----  
NTupleSvc.Output = { "FILE1 DATAFILE='basic_ntuples.root' OPT='NEW' TYP='ROOT'" };  
HistogramPersistencySvc.OutputFile = "basic_histos.root";
```

VISTA

VISTA es un paquete que depende de otros dos: SoWright y SoApp 4.2.2. Utilizamos VISTA para visualizar los IDDigits generados por DSTWriter en el detector. Para el uso básico de este paquete solo hace falta especificar la geometría a utilizar en Vista.opts:

```
//-----  
// Geometry specific options  
//-----  
  
//Especificamos la geometría usando MinervaSQLDDDB:  
//DetectorDataSvc.DetDbLocation = "conddb:/minerva-200708.xml";  
DetectorDataSvc.DetDbLocation = "conddb:/minerva-TP-CR-200708.xml";  
  
// 0 Usando la geometria en MinervaDDDB  
//DetectorDataSvc.DetDbLocation =  
"$MINERVADDDDBROOT/DDDB/minerva-TP-CR-200708.xml";
```

Y el archivo de datos a usar en MyVista.opts:

```
//Incluimos las opciones de Vista.opts  
#include "$VISTAROOT/options/Vista.opts"  
//Archivo de Entrada:  
EventSelector.Input = { "DATAFILE='PFN:TP-CR.trk.root'  
TYP='POOL_ROOTTREE' OPT='OLD'" };
```

Descripción del modelo de objetos usados en el Software (Event Model)

Existen diferentes clases y objetos utilizados por MINERVA , necesarios para la toma y almacenamiento de datos, así como calibración y reconstrucción. Aquí se describen algunos de los objetos con los que se trabajó para la toma de datos.

Clase	ID	Ruta en Event Store	Etiqueta de Ruta
DAQHeader	50300	DAQ/Header	Minerva::DAQHeaderLocation::Default
RawDigit	50303	DAQ/RawDigits	Minerva::RawDigitLocation::Default
IDDigit	50301	Raw/ID/Digits	Minerva::IDDigitLocation::Default
ODDigit	50302	Raw/OD/Digits	Minerva::ODDigitLocation::Default
Pedestal	50311	Cal/Pedestals	
MCHit	50104	MC/ID/Hits	

Cuadro C.1: IDs de algunas clases de Minerva

RawEvent

RawEvent es el objeto que almacena los datos en bruto que se toman por el MinervaDAQ Software. Al momento de hacer mediciones, dichos datos se envían a bancos de memoria (RawBanks) que luego son almacenados en RawEvents.

Un objeto es una unidad que contiene datos y las funciones que operan sobre esos datos. Los datos se denominan miembros dato y las funciones métodos o funciones miembro. Los datos y

las funciones se encapsulan en una única entidad. Los datos están ocultos y sólo mediante las funciones miembro es posible acceder a ellos.

RawDigit

Una vez que se tienen los RawEvents, éstos se pasan por un algoritmo llamado *DecodeRawEvent*, que desempaqueta los bancos de memoria almacenados en los RawEvents luego utilizar esos datos en crear un nuevo objeto, llamado *RawDigit*, con información relevante de los hits para cada pixel (número de canal, ADC, clock ticks, información de si el discriminador se disparó o no, información de dicha carga con los 3 tipos de ganancia alta, media y baja dados por el chip Trip-T para dicho pixel), es decir contiene información de la carga y tiempo medido asociado a cada píxel de cada tubo fotomultiplicador así como información electrónica.

Los RawDigits se guardan en:

- Event/DAQ/RawDigits
- Event/DAQ/SupDigits

Los SupDigits se utilizan para los mismos RawDigits después de haber pasado por una supresión de pedestales (ver paquete *PedestalSuppression* y objeto *Pedestal*)

DAQHeader

Es otro objeto, que contiene información básica sobre los parámetros principales que describen al tipo de datos que se está tomando. Por ejemplo si la lectura es para detectar rayos cósmicos o es una medida de Pedestales, o mediante inyección de luz, o son datos generados por MonteCarlo, o si es proveniente del NuMI. También contiene información sobre el número y sub-número de corrida del dato (actualmente los nombres de los datos tienen la forma:

```
TP_0810041202_00000002_0001_pdst1_v01.root
```

para datos de pedestales, tomada a las 12:02 el 4 de octubre de 2008, corrida 2, subcorrida 2, o:

```
TP_0810040344_00000004_0009_cosmc_v01.root
```

para rayos cósmicos por ejemplo, tomada a las 3:44pm un 4 de octubre de 2008, corrida 4, subcorrida 9.

El objeto DAQHeader se guarda en la siguiente ruta:

- Event/DAQ/Header

MCHit

Son datos generados por MonteCarlo, representa básicamente energía depositada por una sola partícula en una región sensitiva del detector.

Los objetos MCHits son guardados en las siguientes localizaciones:

- Event/MC/ID/Hits
- Event/MC/OD/Hits

MCDigit

Un objeto MCDigit, es la colección de todos los posibles signos de luz provenientes a un píxel particular, incluyendo corriente oscura y crosstalk.

Los objetos de Minerva::MCDigit residen en las siguiente localizaciones:

- Event/MC/ID/Digits
- Event/MC/OD/Digits

Pedestal

Este objeto está ligado al paquete PedestalSuppresion. Establece ciertos umbrales de selección para los datos. Si el RMS de una medida, después de habersele pasado por la supresión de pedestal, es menor a un threshold especificado, dicha medida no se toma en cuenta.

Es decir, si en un tiempo específico, rayos cósmicos atraviesan el detector, se toman los datos, luego se le suprimen los pedestales a los datos y por tanto, solo las barras centelleadoras que hayan sido atravesadas por alguna partícula registraran un RMS significativo (RMS menor a 2 se considera no significativo), las demás son omitidas.

Los eventos suprimidos se guardan en objetos RawDigits en:

- Event/DAQ/SupDigits

Minerva::Digit

Ésta clase asocia a 2 objetos: IDDigits y ODDigits, que son los datos de entrada iniciales que se necesitan para la reconstrucción de datos. Tanto RawDigits como MCDigits son convertidos a IDDigits/ODDigits para la reconstrucción de eventos reales como generados por Montecarlo.

Los Digits son guardados en:

- Event/Raw/ID/Digits
- Event/Raw/OD/Digits

D.1. Búsqueda de Filtraciones de luz

```
void mean_and_rms(){
const char* filename="c:\\home\\data\\Run137\\basic_histos137.root";
TFile f(filename);
f->cd();
const char* format ="BasicHistos/elec_%i_%i_%i_%i/%s/p_%i";
char buf[256];
float cut=200.0;
int ncut=0;
for(int iboard=1; iboard<=10; iboard++){
// gDirectory->ls();
for(int ipix=0; ipix<64; ipix++){
sprintf(buf,format,0,0,7,0,iboard,"qhi",ipix);
TH1* h = (TH1*) f->Get(buf);
if(h) {
h->Fit("gaus","Q");
float rms = h->GetFunction("gaus")->GetParameter(2);
if(rms > cut) {
cout<<buf<<" : "<<h->GetMean()<<" "<<rms<<endl;
ncut++;
}
}
// else { cout<<buf<<endl;}
}
}
cout<<"number passing cut: "<<ncut<<endl;
}
```

D.2. Cálculo de Ganancias e Niveles de Intensidad de Luz

```

void mean_over_sigma2(const char* flash, const char* dir, const char* name="estimate_pes");
void mean_over_sigma2(const char* flash, const char* dir, const char* name){

// gROOT->ProcessLine(".L estimate_zero_pe.C+");
int method=2; // 0=use whole peak, 1=use lower side of peak
TFile f(flash);
TGraph* gpe = new TGraph();
TGraph* ggain = new TGraph();
int cnt1=0; int cnt2=0;
for(int i=0; i<=63; i++){
    TH1* h = (TH1*) f.Get(Form("BasicHistos/elec_0_0_%s/Qbig/p_%i", dir, i));
    if(!h) {cout<<"Trouble h: "<<i<<endl; continue;}
    float peak, rms, frac;

    float mean=h->GetMean();
    float rms=h->GetRMS();
    float ped=0;
    float rmsped=7.5*1.25;

    // cout<<i<<" "<<mean<<" "<<ped<<" "<<rms<<" "<<rmsped<<endl;
// cout<<"mean= "<<mean<<endl;
// cout<<"ped= "<<ped<<endl;
// cout<<"rms= "<<rms<<endl;
// cout<<"rmsped= "<<rmsped<<endl;
    float pnpe=pow(mean-ped,2.0)/(TMath::Abs(rms*rms-rmsped*rmsped));
    float pe=pnpe*1.2;
    cout<<"pe= "<<pe<<endl;

    gpe->SetPoint(cnt1,i,pe); cnt1++;

    if(pe>0.0){
        float gain=(mean-ped)/pe;
        gain*=1e-15/(1.622e-19);
        ggain->SetPoint(cnt2,i,gain); cnt2++;
    }
}

gpe->SetMarkerStyle(kFullCircle);
ggain->SetMarkerStyle(kOpenCircle);

gpe->SetTitle("; pixel; light level (PE)");
ggain->SetTitle("; pixel; gain (ADC)");

TCanvas* c = new TCanvas("c","c",700,800);
c->Divide(1,2);
c->cd(1);
gpe->Draw("ap");
gpe->GetXaxis()->SetRangeUser(-0.5,63.5);
float max=gpe->GetHistogram()->GetMaximum();
float min=gpe->GetHistogram()->GetMinimum();

for(int i=0; i<7; i++){

```

```

float x=7.5 + i*8.0;
TLine* l = new TLine(x,min,x,max);
l->SetLineStyle(kDashed);
l->SetLineWidth(2);
l->Draw();
}

c->cd(2);
ggain->Draw("ap");
ggain->GetXaxis()->SetRangeUser(-0.5,63.5);
max=ggain->GetHistogram()->GetMaximum();
min=ggain->GetHistogram()->GetMinimum();

for(int i=0; i<7; i++){
float x=7.5 + i*8.0;
TLine* l = new TLine(x,min,x,max);
l->SetLineStyle(kDashed);
l->SetLineWidth(2);
l->Draw();
}

c->Print(Form("%s.eps",name));
c->Print(Form("%s.gif",name));

double avpe=TMath::Mean(gpe->GetN(),gpe->GetY());
cout<<"avpe= "<<avpe<<endl;
double unpe=TMath::RMS(gpe->GetN(),gpe->GetY());
cout<<"unpe= "<<unpe<<endl;
double avgain=TMath::Mean(ggain->GetN(),ggain->GetY());
cout<<"avgain= "<<avgain<<endl;
double ungain=TMath::RMS(ggain->GetN(),ggain->GetY());
cout<<"ungain= "<<ungain<<endl;

int nlowp=0;
int nhighp=0;
for(int i=0; i<=63; i++){
if(gpe->GetY()[i]<0.1) nlowp++;
if(gpe->GetY()[i]>10.0) nhighp++;
}

int nlowg=0;
int nhighg=0;
for(int i=0; i<=63; i++){
if(ggain->GetY()[i]<0.2e6) nlowg++;
if(ggain->GetY()[i]>2e6) nhighg++;
}

cout<<"AVERAGE PE: "<<avpe<<" PE"<<endl;
cout<<"PE UNIFORMITY: "<<unpe/avpe<<endl;
cout<<"#LOW PE PIXELS: "<<nlowp<<endl;
cout<<"#HIGH PE PIXELS: "<<nhighp<<endl;
cout<<"AVERAGE GAIN: "<<avgain/1e6<<" * 10^6"<<endl;
cout<<"GAIN UNIFORMITY: "<<ungain/avgain<<endl;
cout<<"#LOW GAIN PIXELS: "<<nlowg<<endl;
cout<<"#HIGH GAIN PIXELS: "<<nhighg<<endl;

```

```

ofstream fout(Form("%s.dat",name));

fout<<"AVERAGE PE: "<<avpe<<" PE"<<endl;
fout<<"PE UNIFORMITY: "<<unpe/avpe<<endl;
fout<<"#LOW PE PIXELS: "<<nlowp<<endl;
fout<<"#HIGH PE PIXELS: "<<nhighp<<endl;
fout<<"AVERAGE GAIN: "<<avgain/1e6<<" * 10^6"<<endl;
fout<<"GAIN UNIFORMITY: "<<ungain/avgain<<endl;
fout<<"#LOW GAIN PIXELS: "<<nlowg<<endl;
fout<<"#HIGH GAIN PIXELS: "<<nhighg<<endl;
fout<<"\n"<<endl;

fout<<"pixel   pe   gain"<<endl;
for(int i=0; i<=63; i++){
    fout<<i<<"   "<<gpe->GetY()[i]<<"   "<<ggain->GetY()[i]<<endl;
}
fout.close();

}

```

D.2.1. Script en bash para su utilización a nivel masivo

```

#!/bin/bash

export RUNFILES="/data_disk"
export list="datosTP.txt" #put all the 1PE runs here
for var in $(cat $list) ; do
    for j in 2 3 ; do
        if [ $j == "3" ] ; then
            for (( i=1;i<=10;i++ )) ; do
                root -q -l \
"mean_over_sigma2.C(\"${RUNFILES}/${var}\",\"1_${j}_${i}\",\"gainpe_1_${j}_${i}_${var%.*.root}\")"
            done
        else
            for (( i=1;i<10;i++ )); do
                root -q -l \
"mean_over_sigma2.C(\"${RUNFILES}/${var}\",\"1_${j}_${i}\",\"gainpe_1_${j}_${i}_${var%.*.root}\")"
            done
        fi #end if condition
    done #end for j condition
done #end for var

```

D.2.2. Ejemplo de Archivo de Datos de Entrada

```

cat datosTP.txt
TP_0810160241_00000020_0002_linjc_v01.root_basic_histos.root
TP_0810160430_00000020_0006_linjc_v01.root_basic_histos.root
TP_0810161414_00000020_0010_linjc_v01.root_basic_histos.root
TP_0810161630_00000020_0015_linjc_v01.root_basic_histos.root

```

TP_0810161858_00000020_0020_linjc_v01.root_basic_histos.root

D.2.3. Archivo principal de Algoritmo básico de reconocimiento de muones

```
//#####
//
// BasicCosmicAlg.cpp
// Simple Algorithm to tag the Muon Events in the
// IDDigits Files.
//
// author: Kenyi Hurtado.
// 10/19/08
//#####

// Gaudi
#include "GaudiKernel/DeclareFactoryEntries.h"
#include "GaudiKernel/AlgFactory.h"
#include "GaudiKernel/MsgStream.h"
#include "GaudiKernel/INTupleSvc.h"
#include "GaudiKernel/SmartDataPtr.h"

// Minerva Event
#include "Event/DAQHeader.h"
#include "Event/RawDigit.h"
#include "Event/IDDigit.h"
#include "Event/ODDigit.h"
#include "Kernel/ChannelID.h"
#include "Kernel/StripID.h"
#include "Event/FilterTagList.h"

// Local
#include "FilterEvent/BasicCosmicAlg.h"

DECLARE_ALGORITHM_FACTORY( BasicCosmicAlg );

using namespace Minerva;

//Constructor

BasicCosmicAlg::BasicCosmicAlg( const std::string& name,
                               ISvcLocator* pSvcLocator)
:GaudiAlgorithm(name, pSvcLocator){

    declareProperty("Cut", m_cut = 1000);
    declareProperty("Maxhits", m_maxhits = 1000);
    declareProperty("Minhits", m_minhits = 40);
// declareProperty("DigitLocation", m_digitLocation = "DAQ/SupDigits");
// declareProperty("DigitLocation", m_digitLocation = Minerva::RawDigitLocation::Default);

    declareProperty ("DAQHeaderLocation", m_daqHeaderLocation = Minerva::DAQHeaderLocation::Default);
    declareProperty ("IDDigitLocation", m_idDigitLocation = Minerva::IDDigitLocation::Default);
```

```

declareProperty ("ODDDigitLocation", m_odDigitLocation = Minerva::ODDDigitLocation::Default);

}

//Initialization
StatusCode BasicCosmicAlg::initialize()
{

    StatusCode sc = GaudiAlgorithm::initialize();
    return StatusCode::SUCCESS;

}

//Execution

StatusCode BasicCosmicAlg::execute()
{
    StatusCode sc;
    // RawDigits* rawDigits = get<RawDigits>(m_digitLocation);
    // RawDigits* rawDigits = NULL;

    /// Create output containers for Digits and register it with the event store
    // put(new IDDDigits(), Minerva::IDDDigitLocation::Default);
    // IDDDigits* ids = getOrCreate<IDDDigits,IDDDigits>(Minerva::IDDDigitLocation::Default);
    //
    // put(new ODDDigits(), Minerva::ODDDigitLocation::Default);
    // ODDDigits* ods = getOrCreate<ODDDigits,ODDDigits>(Minerva::ODDDigitLocation::Default);

    IDDDigits* idDigits = get<IDDDigits>(m_idDigitLocation);
    IDDDigits::iterator idDigit = idDigits->begin();
    // put(new IDDDigits(), "Raw/ID/FilteredDigits");

    double m_sumpe=0;
    for(; idDigit!=idDigits->end(); idDigit++)
    {
        IDDDigit* dig = *idDigit;
        m_counter_hits=idDigits->size();
        double pe = dig->rawPE();
        m_sumpe+=pe;
    }
    // info() << "sumpe=" << m_sumpe <<"\t"<< m_counter_hits << endmsg;
    DAQHeader* daqHeader = get<DAQHeader>(m_daqHeaderLocation);
    // FilterTagList *filterTags = new FilterTagList();

    // put(new FilterTagList(), Minerva::FilterTagListLocation::Default);
    // FilterTagList* filterTags = getOrCreate<FilterTagList,FilterTagList>(Minerva::FilterTagListLocation::Default);

    SmartDataPtr<Minerva::FilterTagList> filterTags( eventSvc(), Minerva::FilterTagListLocation::Default );
    if( !filterTags ){
        debug() << "No filterTags currently in store, creating a new one." << endreq;
        filterTags = new Minerva::FilterTagList();
    }
}

```

```

    m_event = daqHeader->eventNumber();
//    bool statc=false;
    if (m_sumpe>m_cut) {
        if (m_counter_hits>m_minhits && m_counter_hits<m_maxhits){
info() << "event = " << m_event << "\t"<<"sumpe = "<<m_sumpe<<"\t"<<"hits = "<<m_counter_hits<<endmsg;
//        statc=true;
        filterTags->setMuonEvent(true);
        if (filterTags->muonEvent()) info() << "muonEvent()=true" <<endmsg;
        filterTags->setUnknownEvent(false);
        }
        else{
            filterTags->setUnknownEvent(true);
            filterTags->setMuonEvent(false);
        }
    }

    debug() << "Storing Tags" << endreq;

    sc = eventSvc()->registerObject( Minerva::FilterTagListLocation::Default, filterTags );
    if ( sc.isFailure() ){
        error() << "Unable to register Tags in the store" << endmsg;
        return sc;
    }

    //    setFilterPassed(statc); // If it's true, the event is accepted!
    return StatusCode::SUCCESS;
}

// Finalize

StatusCode BasicCosmicAlg::finalize() {

    return GaudiAlgorithm::finalize();
}

```

D.2.3.1. Declaración de las clases necesarias en Gaudi para el algoritmo

```

#####
//
// BasicCosmicAlg.h
// Simple Algorithm to tag the Muon Events in the
// IDDigits Files
//
// author: Kenyi Hurtado.
// 10/19/08
#####

#ifndef BASICCOSMICALG_H
#define BASICCOSMICALG_H 1

#include <string>

```

```
#include "GaudiAlg/GaudiAlgorithm.h"
using namespace Minerva;

class BasicCosmicAlg : public GaudiAlgorithm
{
public:
    // Standard Constructor
    BasicCosmicAlg(const std::string& name, ISvcLocator* pSvcLocator);

    // Destructor
    virtual ~BasicCosmicAlg(){};

    //Algorithm initialization, execution and finalization.
    StatusCode initialize();
    StatusCode execute();
    StatusCode finalize();

private:
    int m_cut;
    int m_minhits;
    int m_maxhits;
    int m_counter_hits;
    std::string m_digitLocation;
    int m_event;
    std::string m_daqHeaderLocation;
    std::string m_idDigitLocation;
    std::string m_odDigitLocation;

    bool m_writeDAQHeader;
    bool m_writeRawDigits;
    bool m_writeIDDigits;
    bool m_writeODDigits;
    // bool statc;

};

#endif // BASICCOSMICALG_H
```

Bibliografía

- [1] <http://minerva.fnal.gov/>
- [2] Proposal to perform a High Statistics Neutrino Scattering Experiment using a fine-grained detector in the NUMI beam , <http://minerva.fnal.gov/proposal.pdf>
- [3] Neutrino Beam data in the MINOS (Main Injector Neutrino Oscillation Search) experiment ,Maury C Goodman for the MINOS Collaboration, 29th International Cosmic Ray Conference Pune (2005) 9, 311–314, <http://icrc2005.tifr.res.in/htm/Vol-Web/Vol-19/19311-usa-diwan-M-abs1-he24-oral.pdf>
- [4] Brookhaven’s Alternating Gradient Synchrotron, <http://www.bnl.gov/bnlweb/facilities/AGS.asp>
- [5] Neutral currents are revealed, <http://public.web.cern.ch/public/en/About/History73-en.html>
- [6] Tau lepton Discovery, Perl, Martin L, U.S.A, Stanford Linear Accelerator Center, Stanford University, Stanford, C, <http://www2.slac.stanford.edu/vvc/nobel/1995nobel.html>
- [7] <http://www-donut.fnal.gov/>
- [8] <http://www.fnal.gov/>
- [9] W.N. Cottingham and D.A Greenwood, “An Introduction to the Standard Model of Particle Physics”, Second Editions, Cambridge University, xv,42 (2001).
- [10] <http://www-sk.icrr.u-tokyo.ac.jp/sk/index-e.html>
- [11] E. K. Akhmedov, 1997, hep-ph/9705451, Fourth International Solar Neutrino Conference, Heidelberg, Germany, 8–11 Apr 1997.

- [12] J Konijn, "The solar neutrino problem", *European Journal of Physics* 20, 399-407 (1999).
- [13] <http://www.sno.phy.queensu.ca/>
- [14] D. Perkins, "Introduction to High Energy Physics", 4th edition, Cambridge University Press, 246 (2000).
- [15] K. Zuber, "Neutrino Physics", Institute of Physics Publishing Bristol and Philadelphia, Oxford, 22,32,57(2004).
- [16] A. Thomas and W. Weise, "The structure of the nucleon", Wiley-VCH, 8 (2001)
- [17] D. Griffiths, "Introduction to Elementary Particles", John Wiley and Sons, Inc, (1987).
- [18] "Introduction to Particle Physics", C.J. Solano Salinas, K. Hurtado, C. Romero. AIP Conf.Proc.1123:13-29,2009.
- [19] C. Giunti and C. Kim, "Fundamentals of Neutrino Physics and Astrophysics",193 (2007)
- [20] MINOS Collaboration:D.G. Michael et al., Observation of Muon Neutrino Disappearance with the MINOS Detector in the NuMI Neutrino Beam, *Phys. Rev. Lett.* 97 (2006)191801.
- [21] "MINERvA, a neutrino nucleus interaction experiment", C.J. Solano Salinas *et al.* AIP Conf.Proc.947:239-244,2007.
- [22] J. Hylen et al., NuMI Technical Design Handbook, Internal NuMI report (2003).
- [23] R. M. Zwaska, Accelerator Systems and Instrumentation for the NuMI Neutrino Beam, PhD thesis University of Texas at Austin, 2005.
- [24] V. Bernard, L. Elouadrhiri, and U. G. Meissner, *J. Phys.*, G28, R1–R35, 2002, hep-ph/0107088.
- [25] D. Drakoulakos et al., 2004, hep-ex/0405002
- [26] Particle Data Group, *J. Phys. G: Nucl. Part. Phys.* (2006).
- [27] C. H. Llewellyn Smith, *Phys. Rep.*, 3, 261, 1972.
- [28] <http://minerva.fnal.gov/news/mri.html>
- [29] <http://openscientist.lal.in2p3.fr/>
- [30] An integration to do scientific visualization and data analysis,G.Barrand, permanent debugger of CERN software at LAL,http://openscientist.lal.in2p3.fr/download/presentations/CHEP04_OpenScientist.ppt

- [31] http://ppewww.physics.gla.ac.uk/~pickford/pacman3/LHCb/dev-source/openscientist-v14r0/OpenScientist/v14r0/OnX/v16r0/doc/h/onx_introduction.h
- [32] GUI User's Manual (Windows DAQ)" (DocDB-MINERvA Document 3027-v3 <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=3027>
- [33] NuMI Technical Design Handbook, retrieved 5 October 2007
- [34] Sacha E. Kopp, "The NuMI Neutrino Beam at Fermilab, Department of Physics", Austin, TX, 2004
- [35] <http://www-numi.fnal.gov/Minos/>
- [36] <http://www.kuraray.co.jp/en/>
- [37] The MINERvA Technical Design Report, Colaboración MINERvA , FERMILAB-DESIGN-2006-01, MINERVA-DOCUMENT-700, 2006, 252pp. (p49).
- [38] http://sales.hamamatsu.com/assets/applications/ETD/pmt_handbook_complete.pdf
- [39] J. Estrada, C. Garcia, B. Hoeneisen and P. Rubinov, "MCM II and the Trip Chip", August 2002, FERMILAB-TM-2226.
- [40] "A Student Project to use Geant4 Simulations for a TMS-PET combination", A. Altamirano, A. Chamorro, K. Hurtado, C. Romero, A. Rueda, C. J. Solano Salinas, D. Wahl, A. Zamudio. AIP Conf.Proc.947:499-500 (2007).
- [41] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=390>
- [42] <http://www.openafs.org/>
- [43] <http://cern.ch/gaudi>
- [44] <http://root.cern.ch/root>
- [45] <http://wwwinfo.cern.ch/asd/lhc++/AIDA/>
- [46] <http://seal.cern.ch>
- [47] <http://www.nongnu.org/cvs/>
- [48] <http://web.mit.edu/Kerberos/>
- [49] Wouter Verkerke, "Data Analysis", University of California Santa Barbara / NIKHEF), 2004,p10-11

- [50] Spiegel, M. R. Correlation Theory. Ch. 14 in Theory and Problems of Probability and Statistics, 2nd ed. New York: McGraw-Hill, pp. 294-323, 1992.
- [51] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=2575>
- [52] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=2409>
- [53] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=2443>
- [54] Edwards, A. L. "The Correlation Coefficient." Ch. 4 in An Introduction to Linear Regression and Correlation. San Francisco, CA: W. H. Freeman, pp. 33-46, 1976.
- [55] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=2682>
- [56] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=2544>
- [57] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=2969>
- [58] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=3033>
- [59] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=2948>
- [60] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=2879>
- [61] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=3076>
- [62] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=2811>
- [63] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=2686>
- [64] <http://minerva-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=3104>