

Adquisición de datos

De los detectores a los datos

- Josefina Alconada (IFLP/UNLP/CONICET)
- Fernando Monticelli (IFLP/UNLP/CONICET)



Latin American alliance for
Capacity building in Advanced physics

LA-CoNGA physics



Cofinanciado por el
programa Erasmus+
de la Unión Europea



Introducción



TEMARIO:

- Repaso de detectores
 - Interacción de partículas con los subdetectores
- Front-end electronics:
 - Conversión de la interacción en pulsos eléctricos
 - Digitalización
 - DAQ
- Trigger
 - ¿Qué eventos se guardan y que no?
- Diseño del trigger
- Tratamiento de datos recolectados
- Procesamiento de datos
 - Simulación, reconstrucción, identificación, calibración
- Visualización y análisis

Detectando las partículas



Estructura de un setup HEP

- **OBJETIVO:** caracterizar un subconjunto específico de interacciones de partículas que ocurren en una región del espacio
- **EVENTO:** Instancias de las interacciones a estudiar.
 - Comienza con una interacción primaria e incluye todas las interacciones subsecuentes.
 - Diferentes eventos son independientes
- **CARACTERIZACIÓN:** Idealmente todas las partículas generadas son identificadas y sus trazas reconstruidas.
- **SETUP HEP:**
 - **REGIÓN DE INTERACCIÓN:** aquella donde los eventos van a ser detectados
 - Rodeada por detectores de partículas on readout y digitalización de partículas asociados (Front end)
 - Outputs del Front end se transmite al back end para procesamiento y storage
- **COLISIÓN DE PARTÍCULAS**
 - causa una reacción que genera nuevas partículas que deben ser identificadas
 - Detectores de partículas que interactúan con diferentes tipos de partículas y miden distintas propiedades físicas



Identificación de partículas

- Interacción con el detector → Identificación: [Ver clase 17](#)

Detector type	Interaction	Detector examples
Tracking	Ionization	Gaseous detectors: Multiwire Chamber Drift Chamber Time Projection Chamber Hodoscopes
	Pair production	Semiconductor detectors
Calorimetry	Bremsstrahlung	EM calorimeters Hadronic calorimeters
	Scintillation	Scintillation counters
Particle ID	Čerenkov radiation	Ring Imaging Čerenkov counters
	Transition radiation	TR detectors

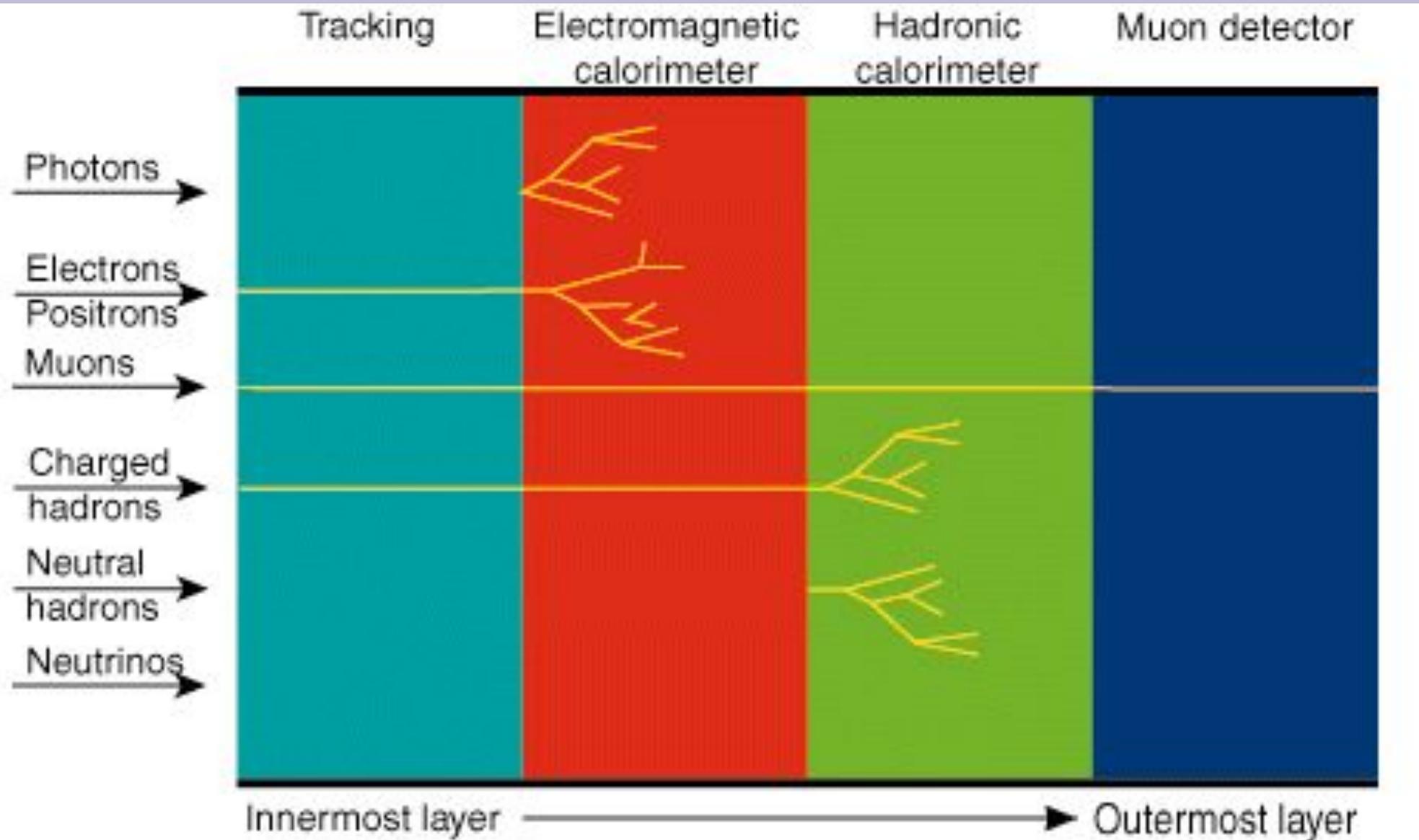
- Track detectors: Materials that react with charged particles
 - Cluster of hits
 - Fit
 → Trayectoria + momento
- Calorímetros: Bloques de material que absorbe la energía
 - EM: e y γ detectados por sus interacciones con partículas cargadas del material
 - HAD: hadrones son absorbidos por interacciones con los núcleos atómicos
- Muon chambers: interacción muy débil
- Únicas partículas que escapan: neutrinos

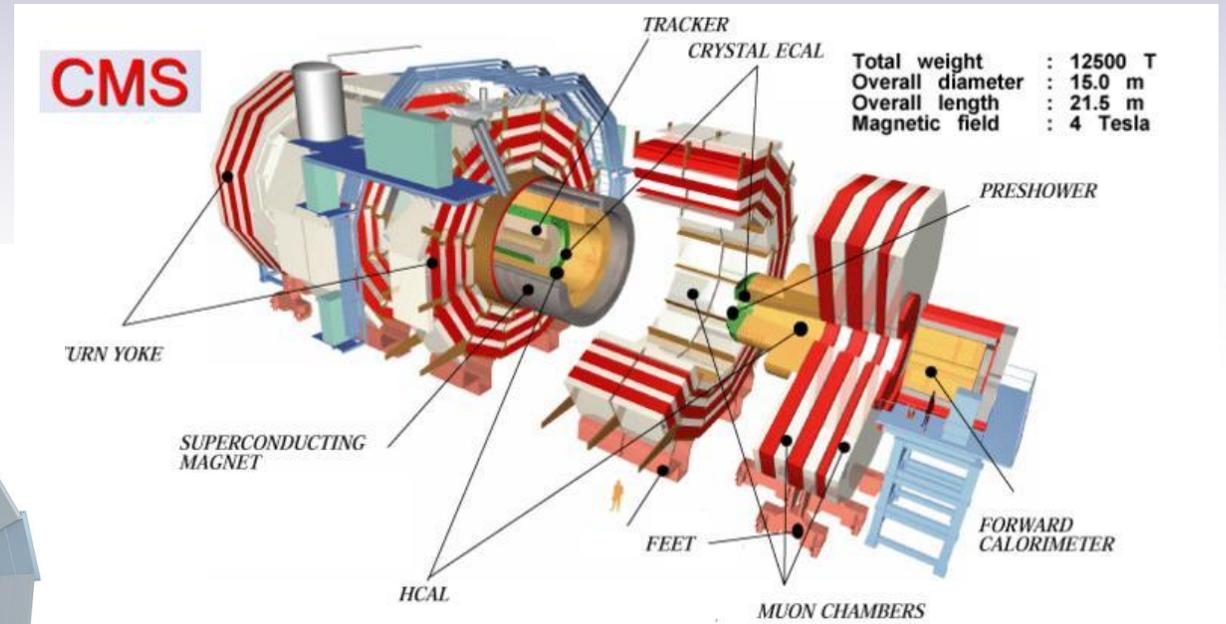
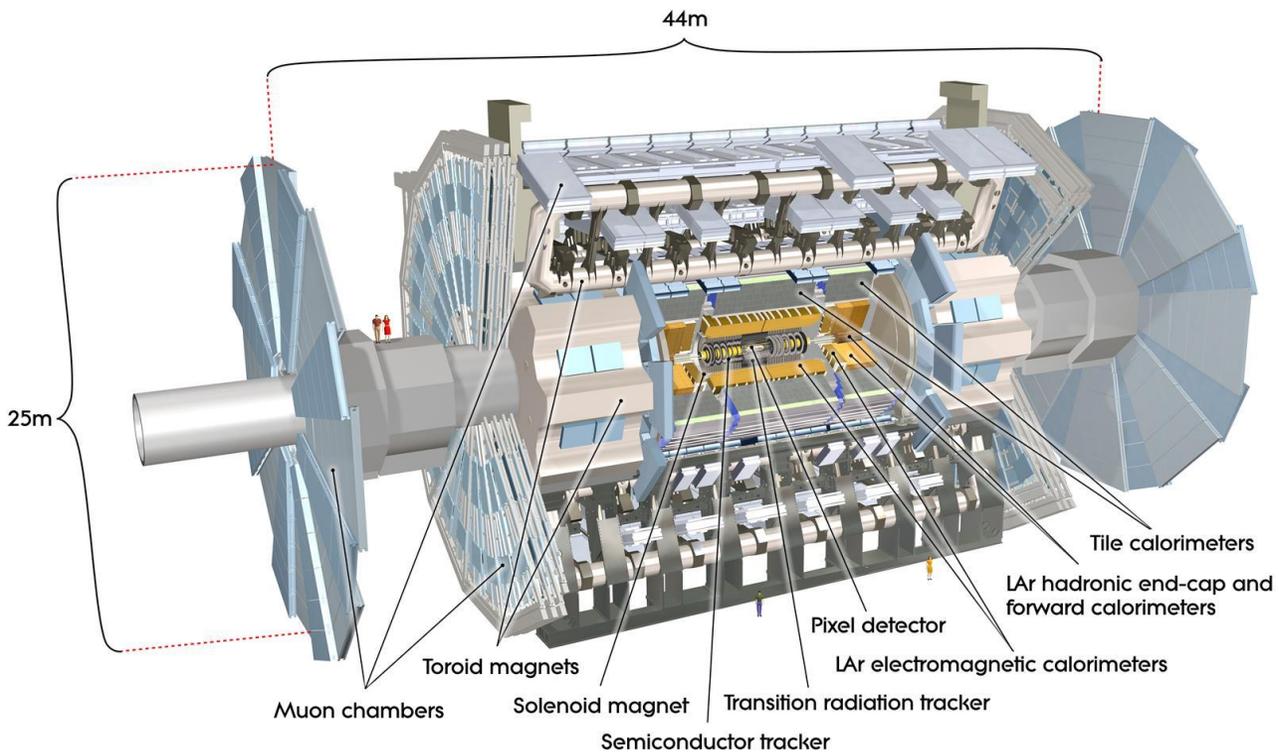
- Readout electrónico del detector:

- Resultado de la interacción de una partícula → aparición de una carga eléctrica Q
- Q es transportada a través de campos eléctricos y colectada por electrodos



Identificación de partículas





De las detecciones a las señales electrónicas



- El conocimiento y caracterización de materiales -> Que material interactúa de qué manera con qué partícula
- Se utilizan esa información para construir subdetectores
- Para poder hacer una medición necesitamos cuantificar esa interacción
 - ¿Por dónde pasó la partícula cargada?
 - ¿Cuánta energía tenía?

Electrónica de FrontEnd:

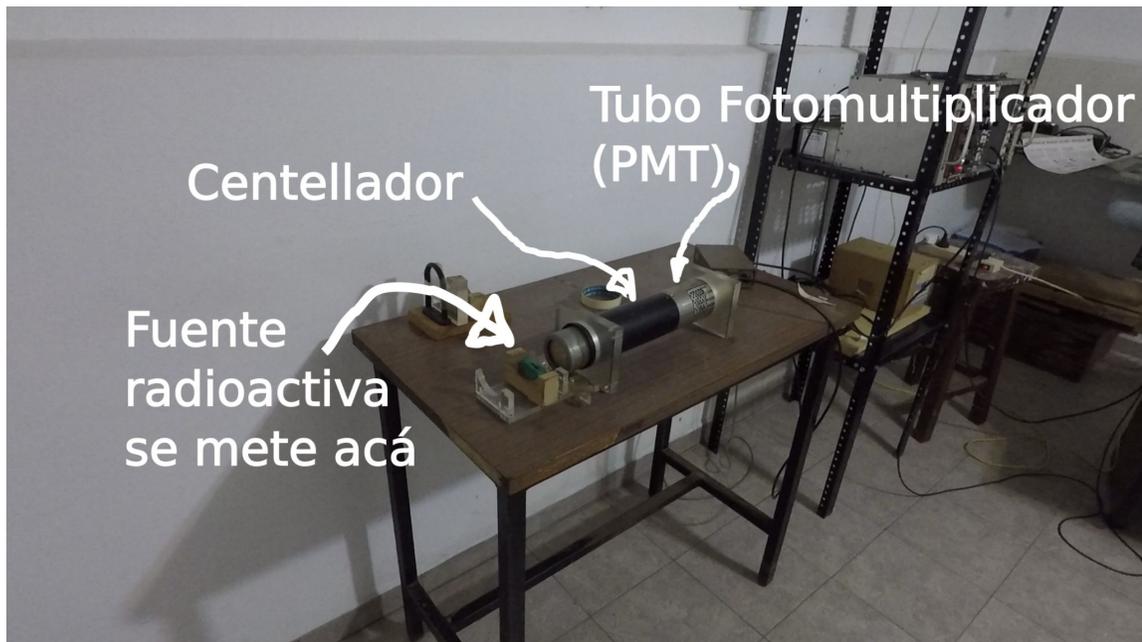
- La electrónica que está asociada a la pieza de subdetector
- Devuelve una señal relacionada con la interacción de la partícula con el subdetector
- La salida del FrontEnd se usa para adquisición y Trigger



Ejemplo con equipo de laboratorio: Detección de radiación gamma con un centellador y un fotomultiplicador

Subdetector: Centellador y fotomultiplicador

Frontend: Alimentación, amplificación, reshape/digitalización



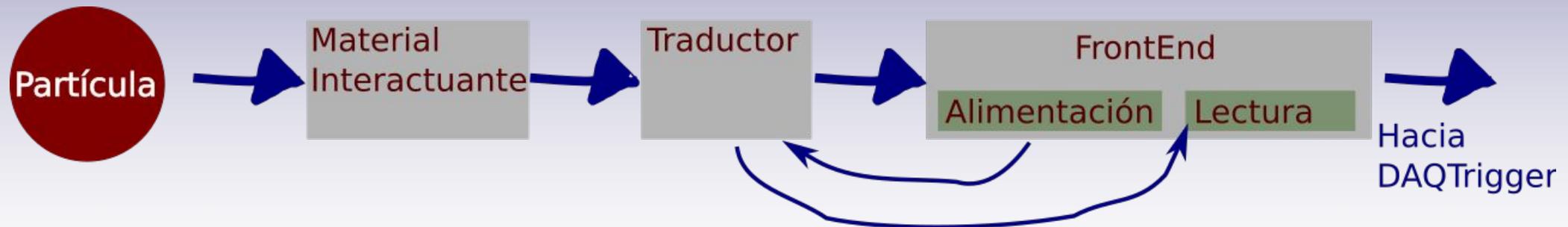


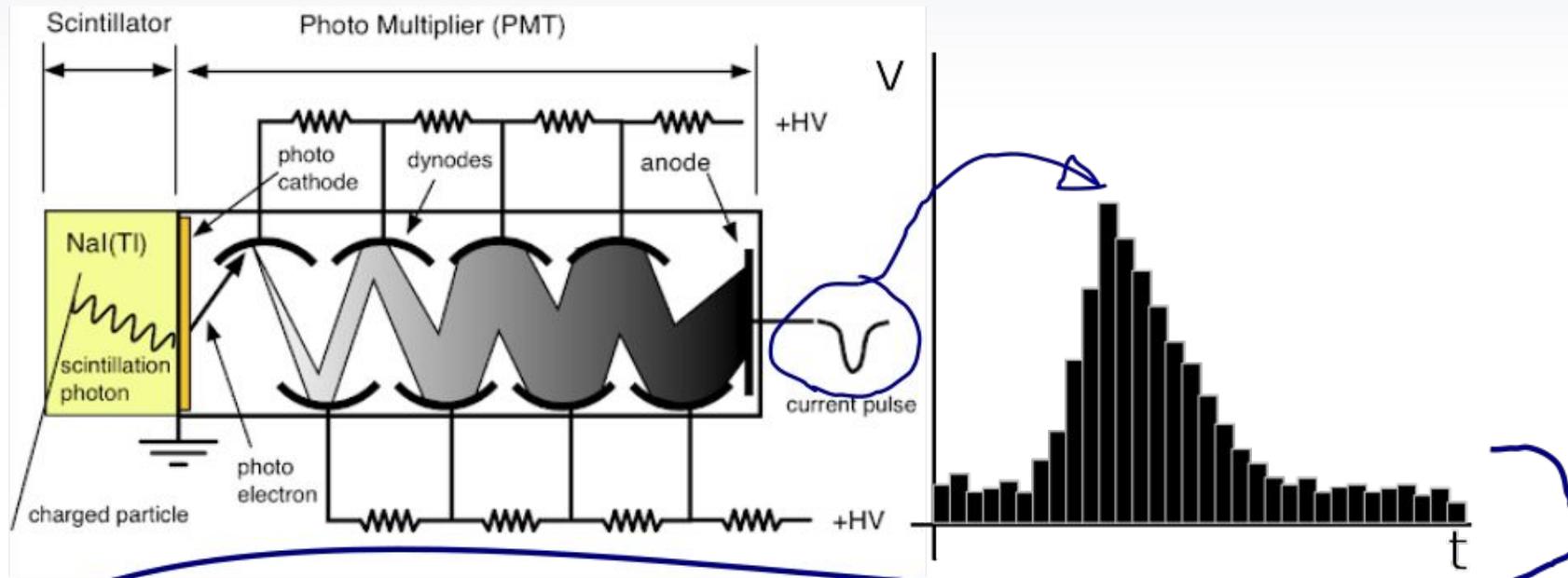
Diagrama esquemático sobresimplificado:

- Partícula interactúa con la pieza del subdetector (centellador, silicio, plomo, etc)
- Adosado tiene un traductor que transforma una interacción en magnitud eléctrica
 - El traductor puede ser una pieza adicional (calorímetros de muestreo)
 - Puede ser el mismo material (el silicio de los detectores de trazas)
- Ese traductor tiene que recibir alimentación eléctrica (polarización) y tiene que ser leído
- FrontEnd se encarga de operar el traductor. Puede ser analógico o digital
- Tanto la adquisición de los datos como el sistema de Trigger se hace via el FrontEnd



Representación lógica de estas señales

- Hoy en día los FrontEnds típicamente envían los datos en forma digital (si no, se digitalizan)
- De modo que tanto el Trigger como el DAQ reciben un tren de datos digitales (ya sea para disparar el evento, o almacenarlo)
- Cada tipo de subdetector tendrá un tipo distinto de tren de datos y codificado de forma distinta



El tren de datos enviado por el FrontEnd:

2, 5, 3, 6, 10, 32, 54, 100, 90, 95, 93,...



OK. Esa es la idea básica pero en la vida real...

... las cosas son más complicadas

Multiplicidad

- Los experimentos de HEP tienen que leer muchos de estos FrontEnds al mismo tiempo:
 - AUGER: 1660 x 3 PMTs (SD) + 24 x 440 PMTs (FD)
 - IceCube: 5160 módulos ópticos
 - ATLAS/CMS: ~100M canales (Calorímetros, trazas, muon detector)

Velocidad

- Los datos pueden venir todo el tiempo!
 - Los experimentos del LHC reciben datos a 40 MHz!!!
 - LIGO: constantemente leyendo datos -> few TB/day

Espacio y energía

- En general tenés el espacio limitado
 - Incluso en grandes facilidades, tenés que meter muchos FrontEnds en espacios reducidos
 - ¡Y tenés que alimentarlos y cablearlos todos! -> Servicios son una parte importante del diseño del experimento

Trigger

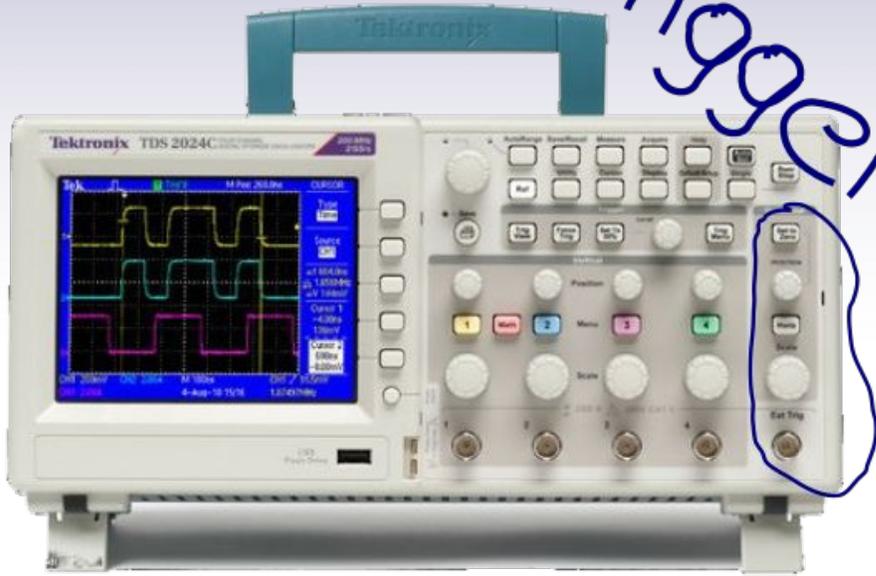


Discusión:

- ¿Conocen/vieron (en algún contexto (cualquiera sea)) alguna vez un sistema de Trigger?



¿Qué es un sistema de trigger?



Veamos el concepto a partir del que conocemos todos: EL OSCILOSCOPIO!

- El Trigger hace que veas una señal periódica "quieta" en la pantalla
- Define un criterio de cuándo empezar a dibujar

Trigger en el osciloscopio:

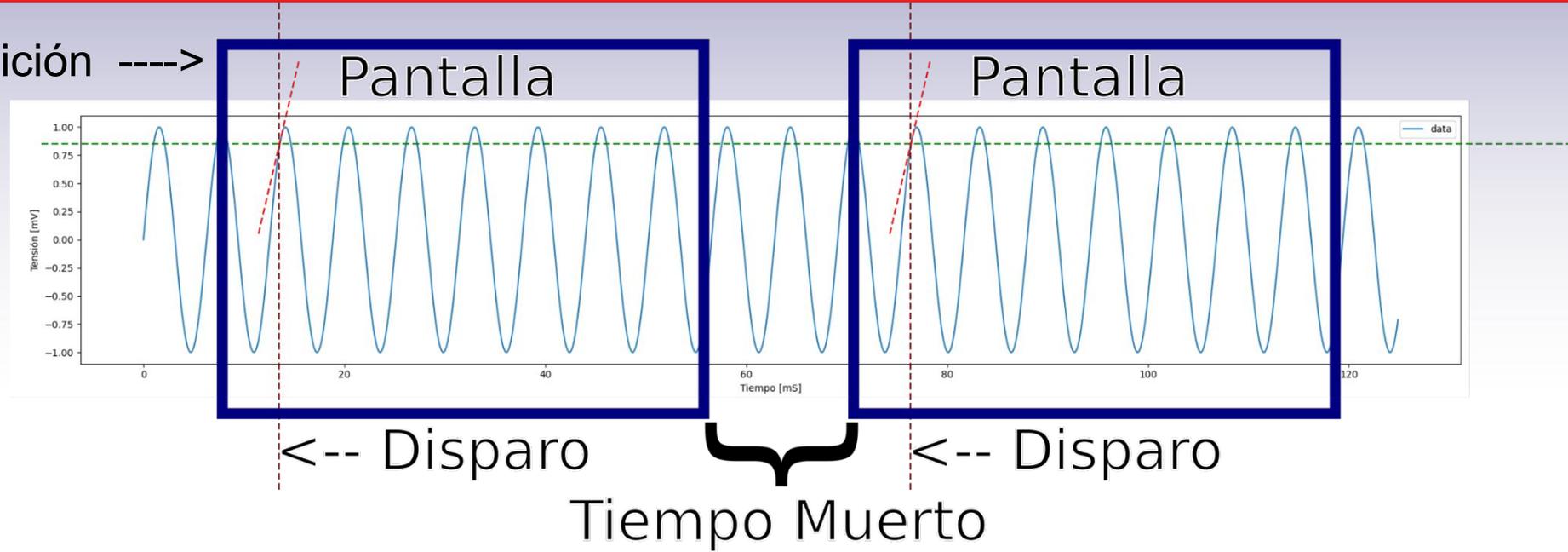
- Criterio: Umbral y Flanco:
- Si se cumple el criterio Dispara y adquiere los datos (dibuja en pantalla)
- Si no se cumple el criterio espera
- Mientras adquiere (dibuja) ignora la señal de entrada (dead time)



Trigger del osciloscopio como ejemplo

Mi sistema de adquisición ---->

Tren de datos ---->



Conceptos:

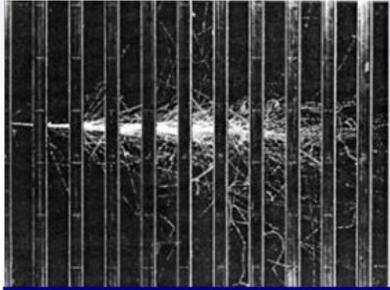
- Trigger: sistema que determina cuáles datos se adquieren
- Tiempo muerto: Tiempo que no se colectan datos porque DAQ está ocupado
- Si no se cumple el criterio espera
- Mientras adquiere (dibuja) ignora la señal de entrada (dead time)



¿Y en experimentos de escala que es un Trigger?

Define el criterio para que los datos sean colectados o almacenados!

- En HEP: Colisiones -> subdetectores -> FrontEnds -> Trigger

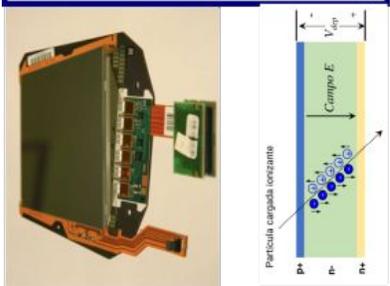


Frontend

→ 2, 5, 3, 6, 10, 32, 54, 100, 90, 95, 93, ... →

Frontend

→ 0, 0, 13, 32, 154, 87, 109, 59, 35, ... →



Tren de datos de los calorímetros

Tren de datos de los detectores de trazas

Trigger:

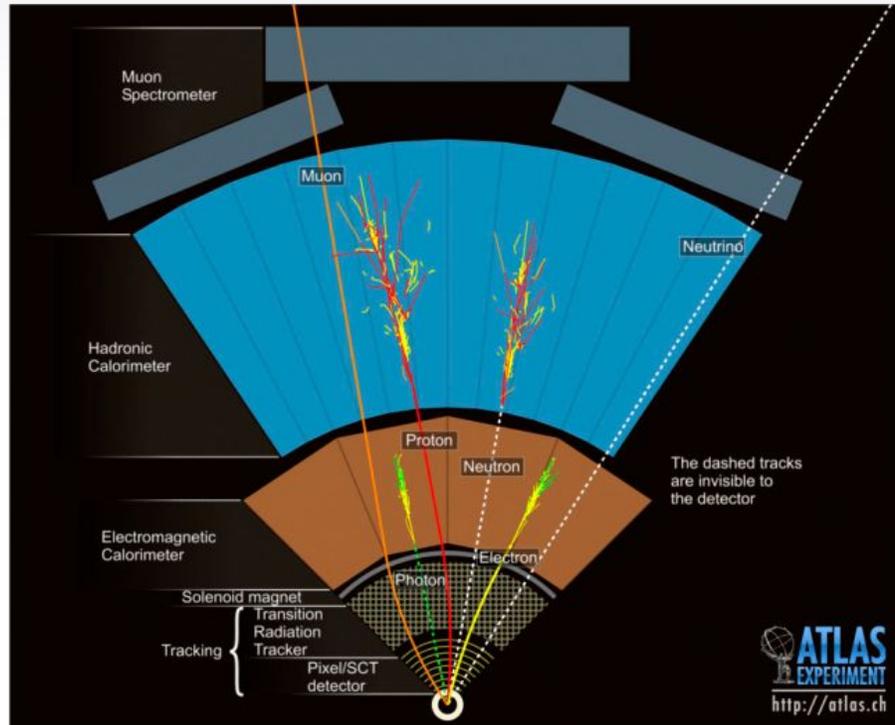
- Lee los datos
- Los interpreta
- Aplica un criterio
- Decide si guarda o no



¿Y en experimentos de escala que es un Trigger?

Define el criterio para que los datos sean colectados o almacenados!

- En HEP: Colisiones -> subdetectores -> FrontEnds -> Trigger



Trigger:

- Lee los datos
- Los interpreta
- Aplica un criterio
- Decide si guarda o no



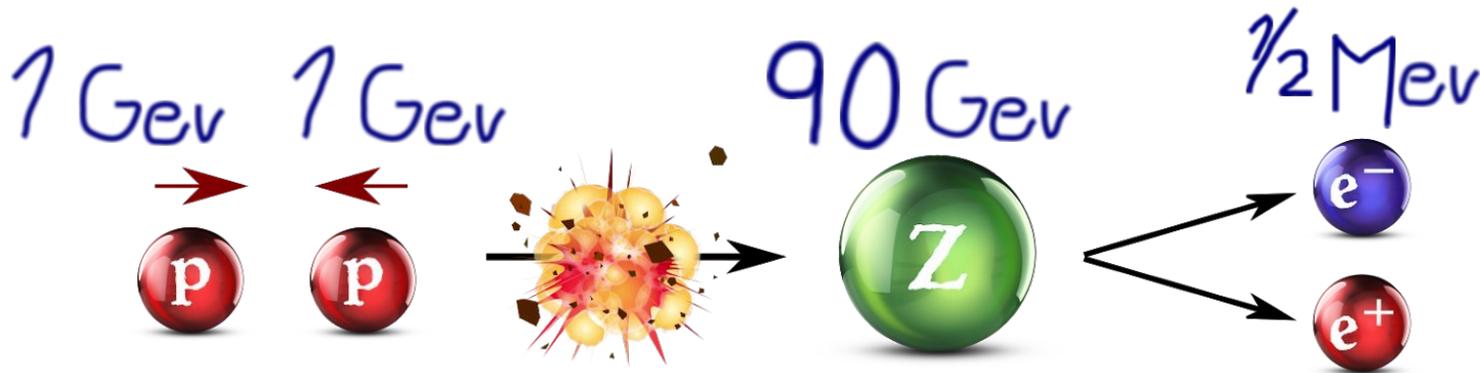
¿Y en experimentos de escala que es un Trigger?

Define el criterio para que los datos sean colectados o almacenados!

- En HEP: Colisiones \rightarrow subdetectores \rightarrow FrontEnds \rightarrow Trigger
- Con las señales de cada sub detector
- Evalúa la hipótesis (por ej $Z \rightarrow e^+e^-$)
- Si se cumple: colecta toda la información de cada FrontEnd y almacena los datos

Trigger:

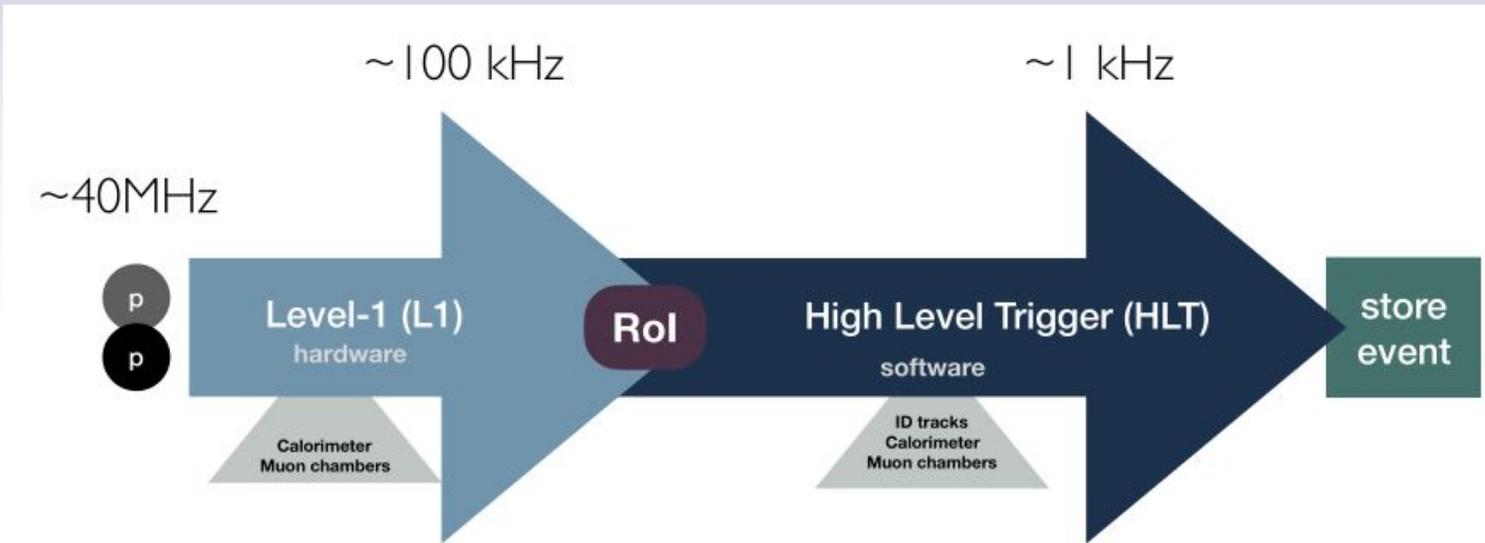
- Lee los datos
- Los interpreta
- Aplica un criterio
- Decide si guarda o no





Ejemplo de la vida real: Trigger de ATLAS

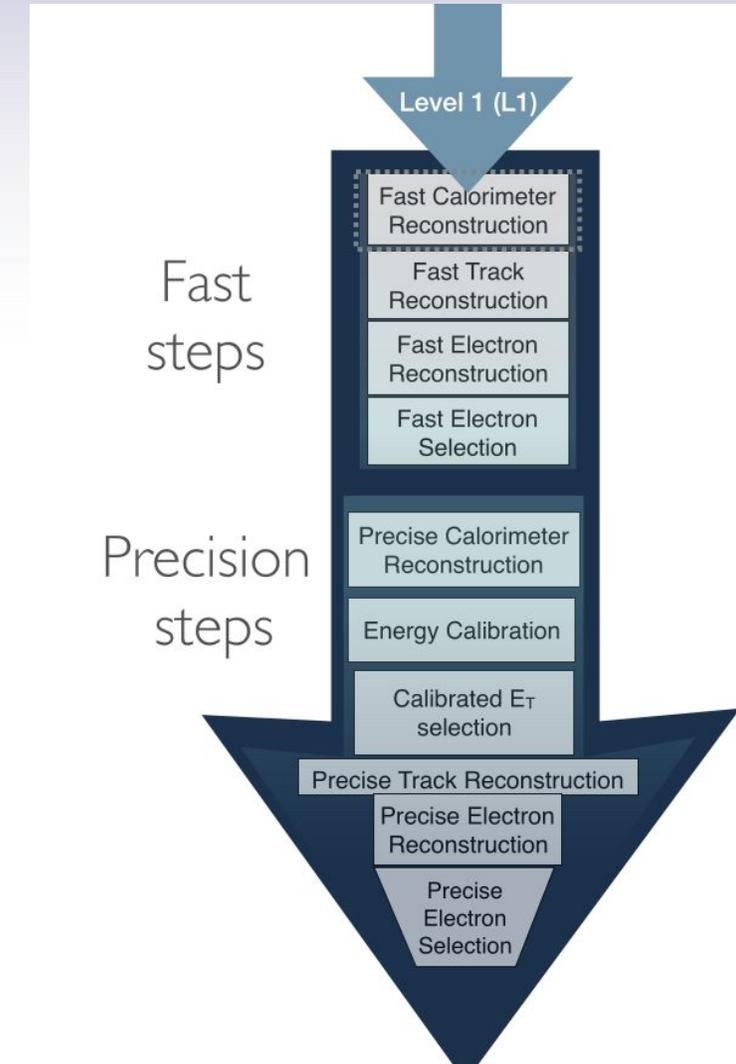
Misión: de los 40M/s colisiones -> quedarse con ~ 1000 de señal



Trigger en Niveles:

- Se analiza solo las regiones del detector que tienen señal
- Nivel 1: FPGAs -> granularidad reducida y sin trazas
- Alto nivel: Software (farms de PCs)

A su vez tienen una parte rápida y otra etapa tipo offline





¡¡¡¡¡IMPORTANTE!!!!

- En general, van a haber cosas que se parezcan a nuestra señal, pero que NO LO SEAN!
 - En colisionadores de hadrones esto es lo más común:
 - El Trigger, ante la sospecha de que sea señal lo tiene que guardar
 - Pero como no puede guardar todo, tiene que sospechar bien

Concepto:

- **Trigger RATE:** es la frecuencia de aceptación de eventos del trigger
- **Eficiencia del Trigger:** Fracción de eventos de señal son disparadas por el trigger
- **Dead Time:** Tiempo muerto -> que ignorás datos de entrada -> Sistema ocupado

Compromiso:

- La tecnología disponible -> Sistema de trigger -> Rate/Eficiencia al servicio de la física que estás buscando
 - LHC: Mucho fondo que no te interesa -> Trigger condicionado por el Rate
 - IceCube: tasa de eventos es muy baja -> Necesitas eficiencia ~100%

¿Recreito?

Criterios para diseñar un Trigger en HEP



Cómo diseñamos un trigger?

- Objetivo de los sistemas de trigger y DAQ: guardar la máxima cantidad de datos para un cierto proceso a un costo mínimo
- Aceptar todos los procesos de bajo rate (higgs, nueva física)
- Rechazar:
 - Eventos con candidatos que falseen nuestro objeto (fondo)
- Necesitamos definir un MENÚ de posibles triggers antes de tomar datos
- Diseño es un compromiso entre eficiencia y latencia:
- Incrementar la complejidad de los algoritmos del trigger
 - mayor fracción de eventos no deseados descartados
 - aumenta los requerimientos de procesamiento: mayor latencia y tamaños de buffer

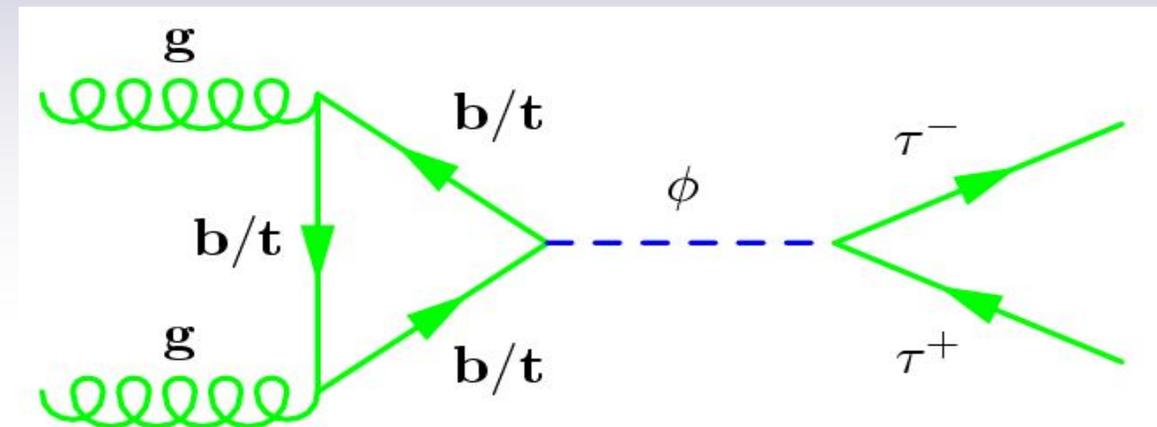


Cómo diseñamos un trigger?

Ejemplo: $H \rightarrow \tau\tau$

Estado final: 2 taus

- Cómo puedo disparar eventos que contribuyan a este proceso?
- Pedir 2 objetos taus
 - De qué energía?
- Monitorear $\text{RATE} = \sigma \cdot \mathcal{L}$
- Puedo usar el trigger de single taus?
- Agregar condiciones topológicas?
 - Distancia entre taus ($\Delta\Phi$)
- Pedir objetos extra (Jets)



PREGUNTA



PREGUNTA:

- Puedo perder nueva física por no tener los triggers necesarios?



- No todos los triggers del menú tienen que ser para un análisis:
 - calibración, tests, cosmits
- Para esos triggers, no es necesario guardar todos los eventos:
 - Prescale: Solo guardar 1 evento por cada X apariciones
 - Tener en cuenta a la hora de usar esta información
 - Depende de la toma de datos

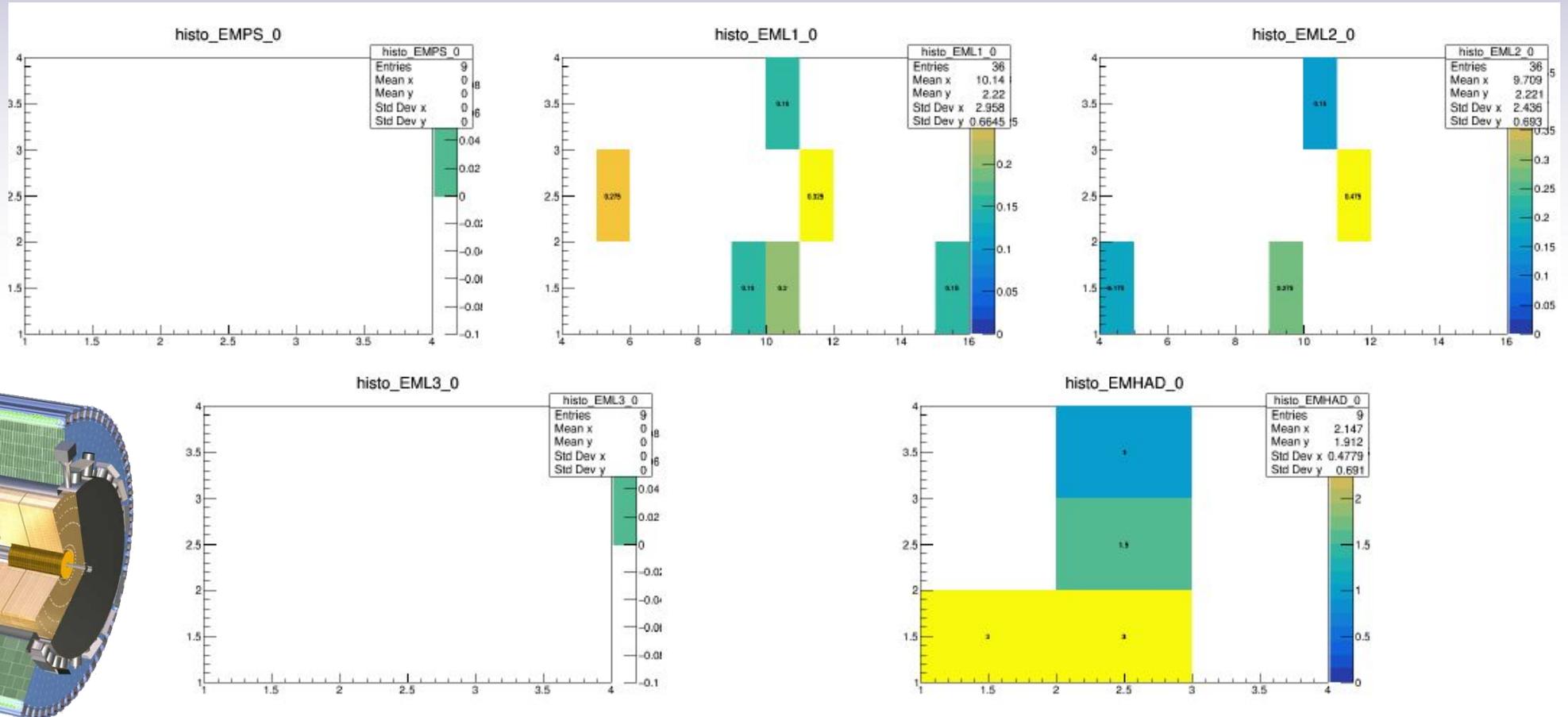


- Entender la electrónica disponible
- Componentes de un algoritmo de trigger:
 - **Seeding:** Elegir un criterio para mirar la actividad en el calor
 - En qué regiones del detector voy a aplicar un algoritmo para sumar la energía?
→ Ej. : Analizar eventos que tengan algún depósito de energía en EM2
 - **Clustering:** Como “sumo” la energía de un posible candidato
 - Sumo todos los depósitos en ventanas que rodéen el seed?
 - Elijo alguna forma característica?
 - Que limitaciones tengo a la hora de decidir cómo sumar?
 - **Isolation:** Usar información de la energía alrededor del candidato para rechazar el fondo aun más



Cómo diseñamos un trigger? Estrategia de L1 trigger

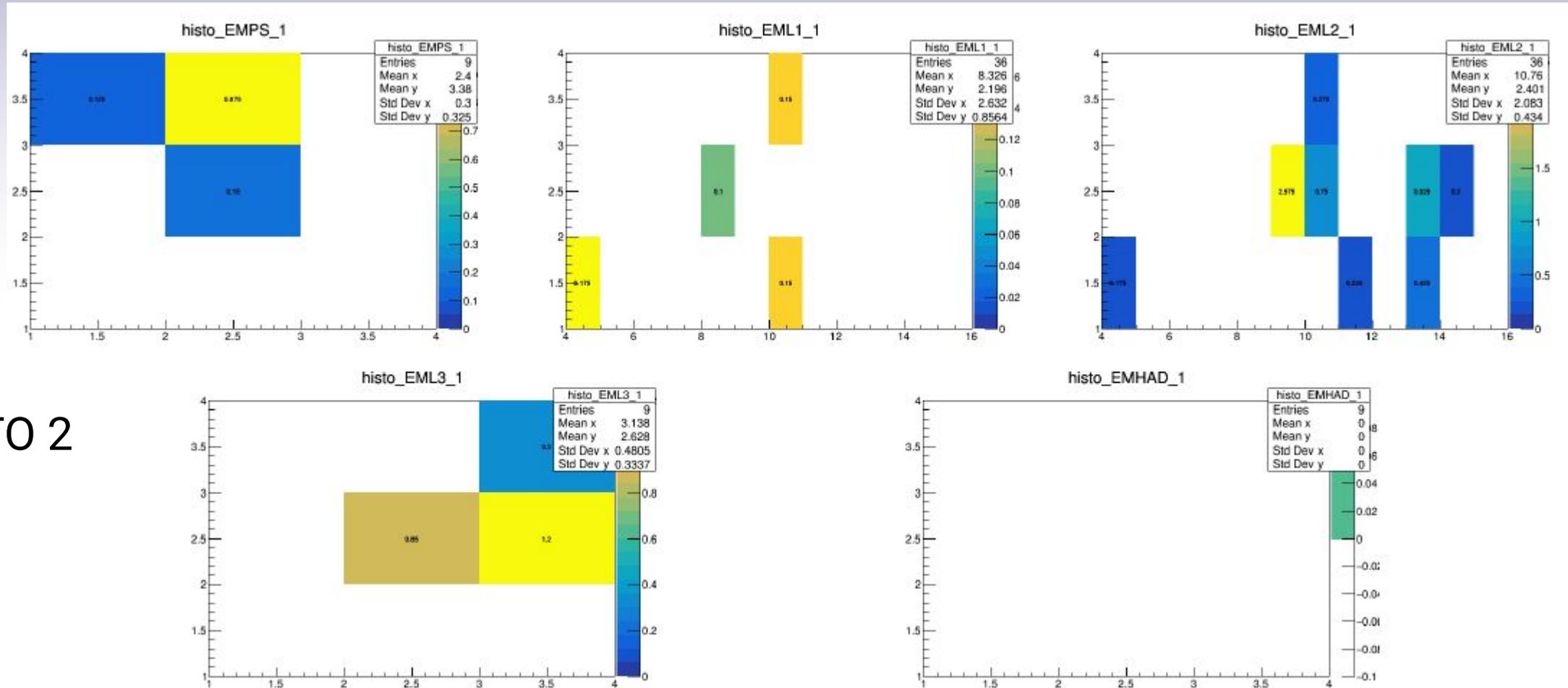
Ejemplo:
taus en L1
CANDIDATO 1



- Large activity in had calo
- EM3 empty



Cómo diseñamos un trigger? Estrategia de L1 trigger

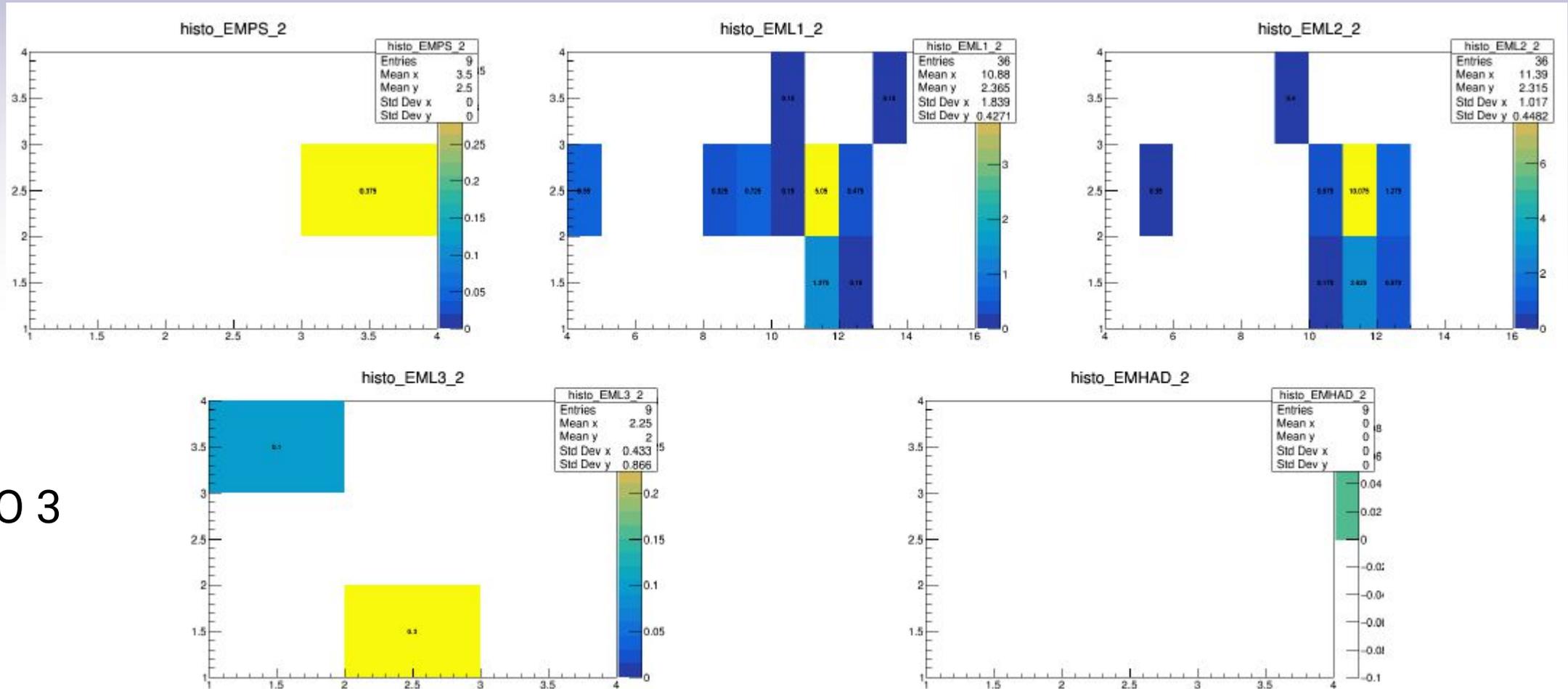


CANDIDATO 2

- EM1 y EM2 muestran actividad en regiones distintas
- Dos taus diferentes?



Cómo diseñamos un trigger? Estrategia de L1 trigger

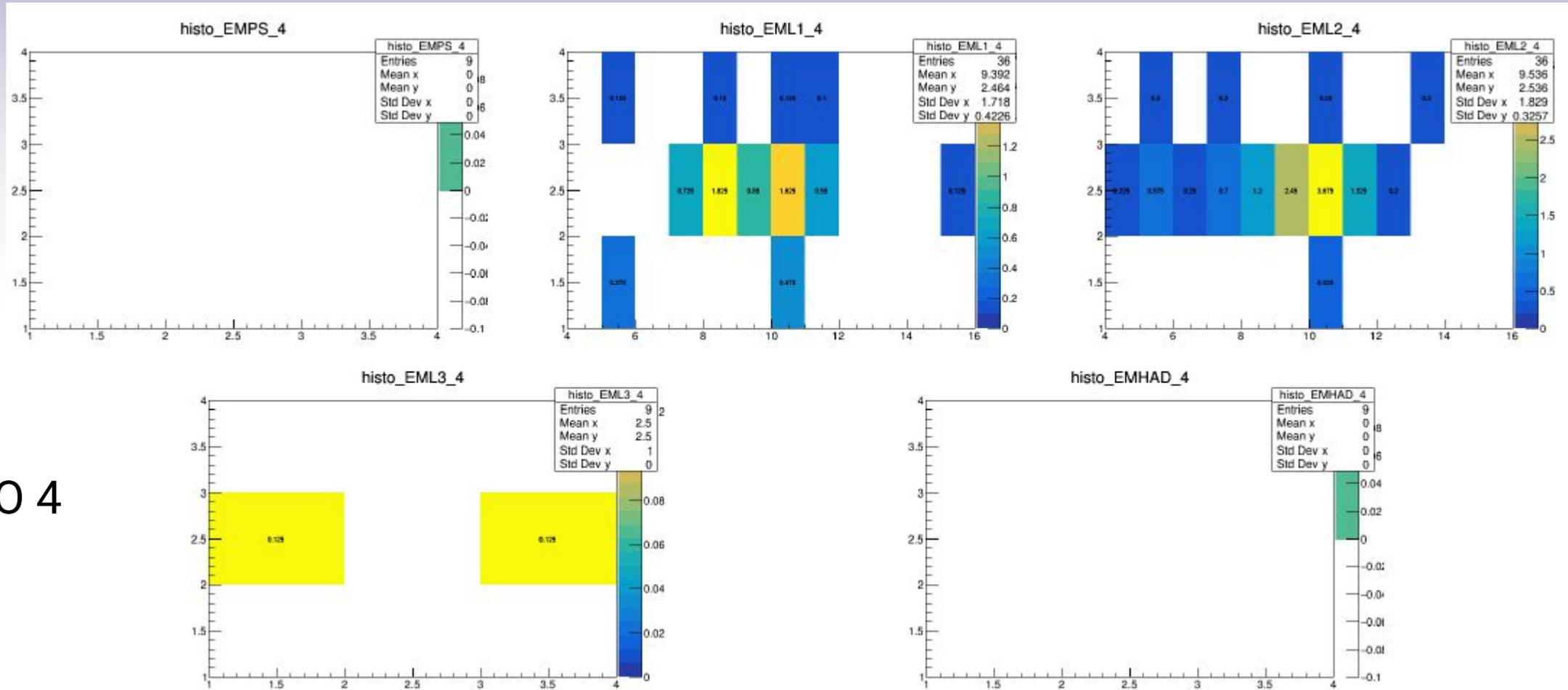


CANDIDATO 3

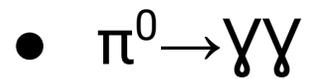
- Mucha actividad en EM2



Cómo diseñamos un trigger? Estrategia de L1 trigger



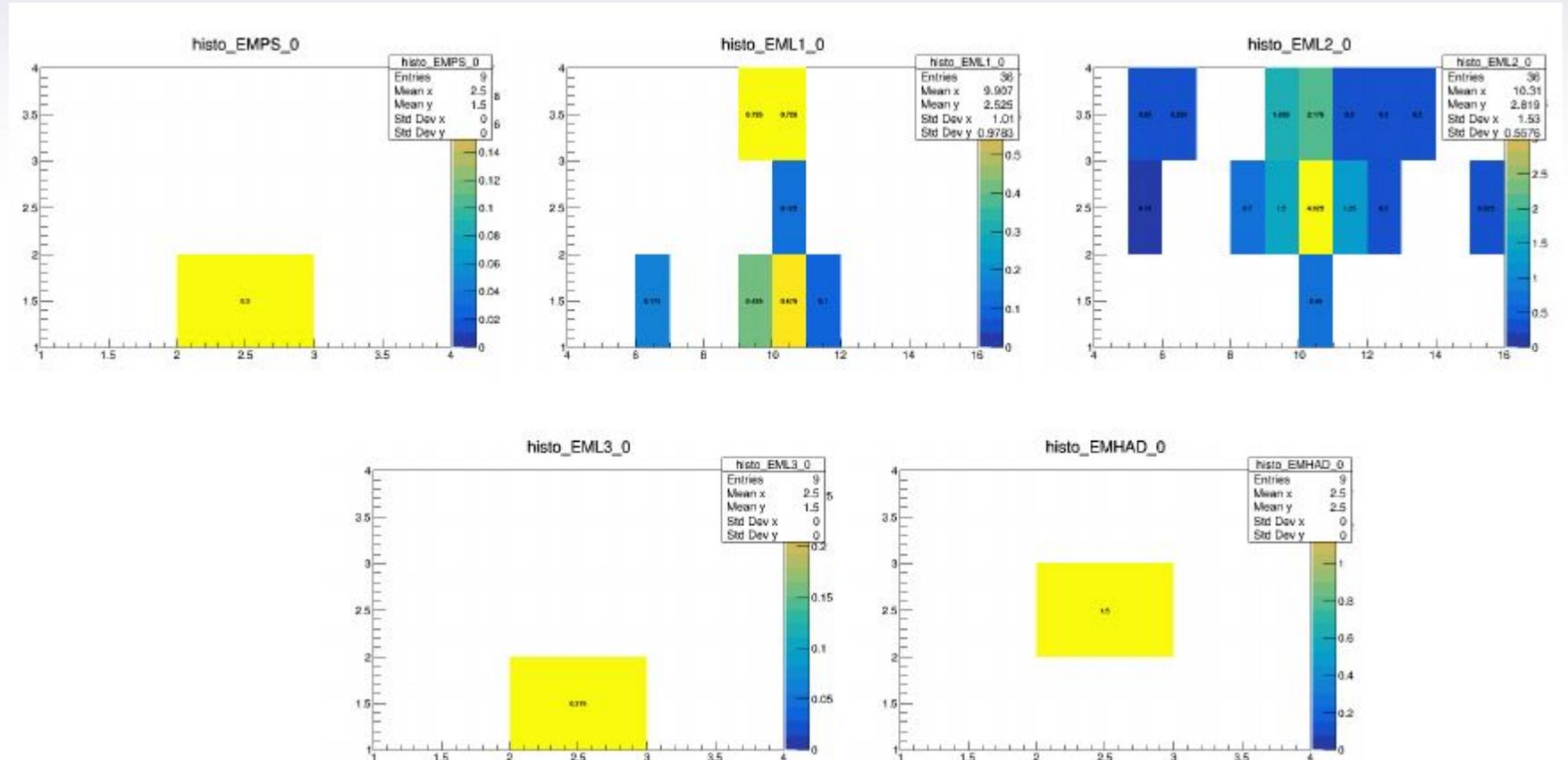
CANDIDATO 4





Cómo diseñamos un trigger? Estrategia de L1 trigger

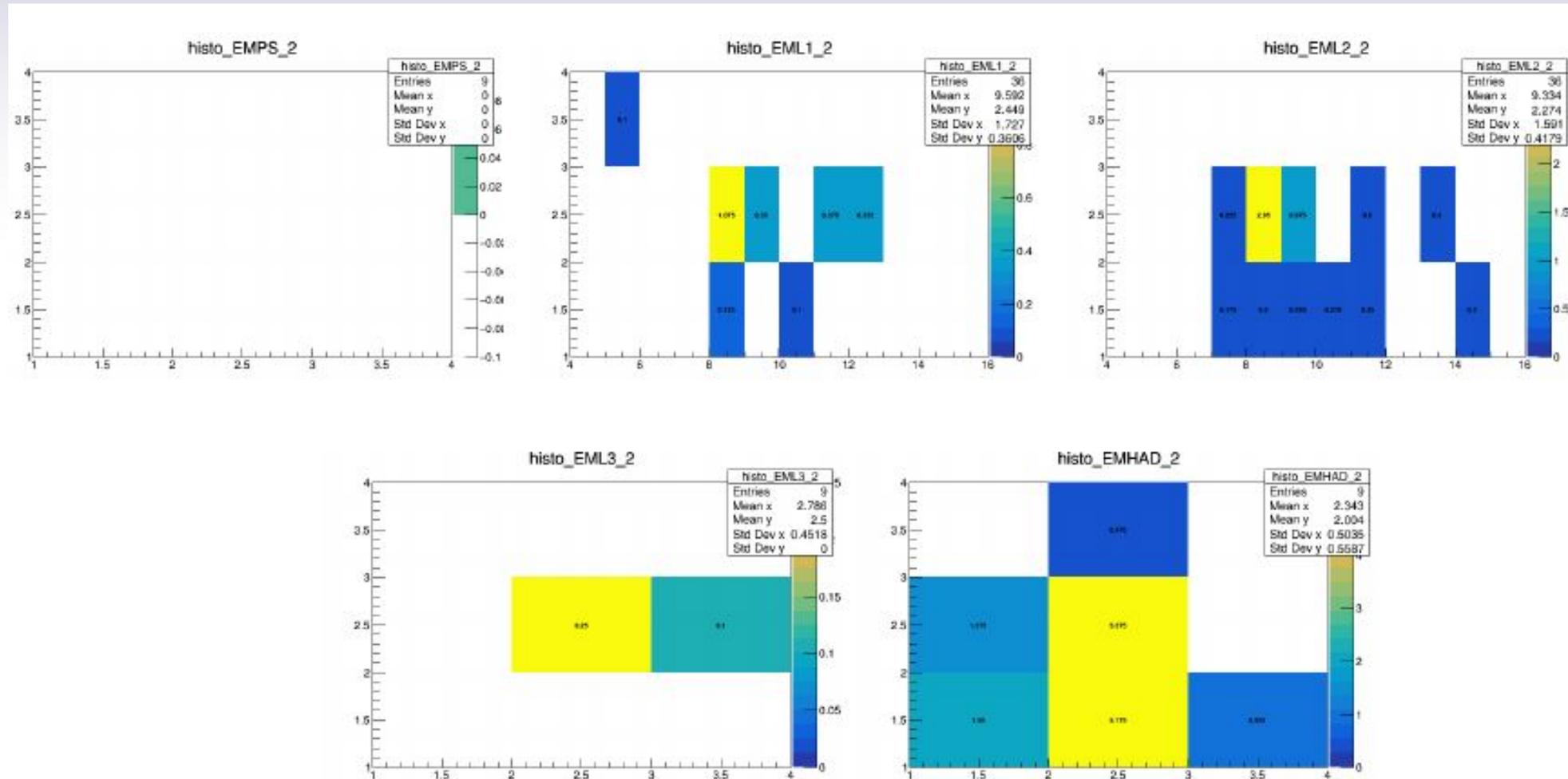
CANDIDATO 1 a jet (Principal fondo)





Cómo diseñamos un trigger? Estrategia de L1 trigger

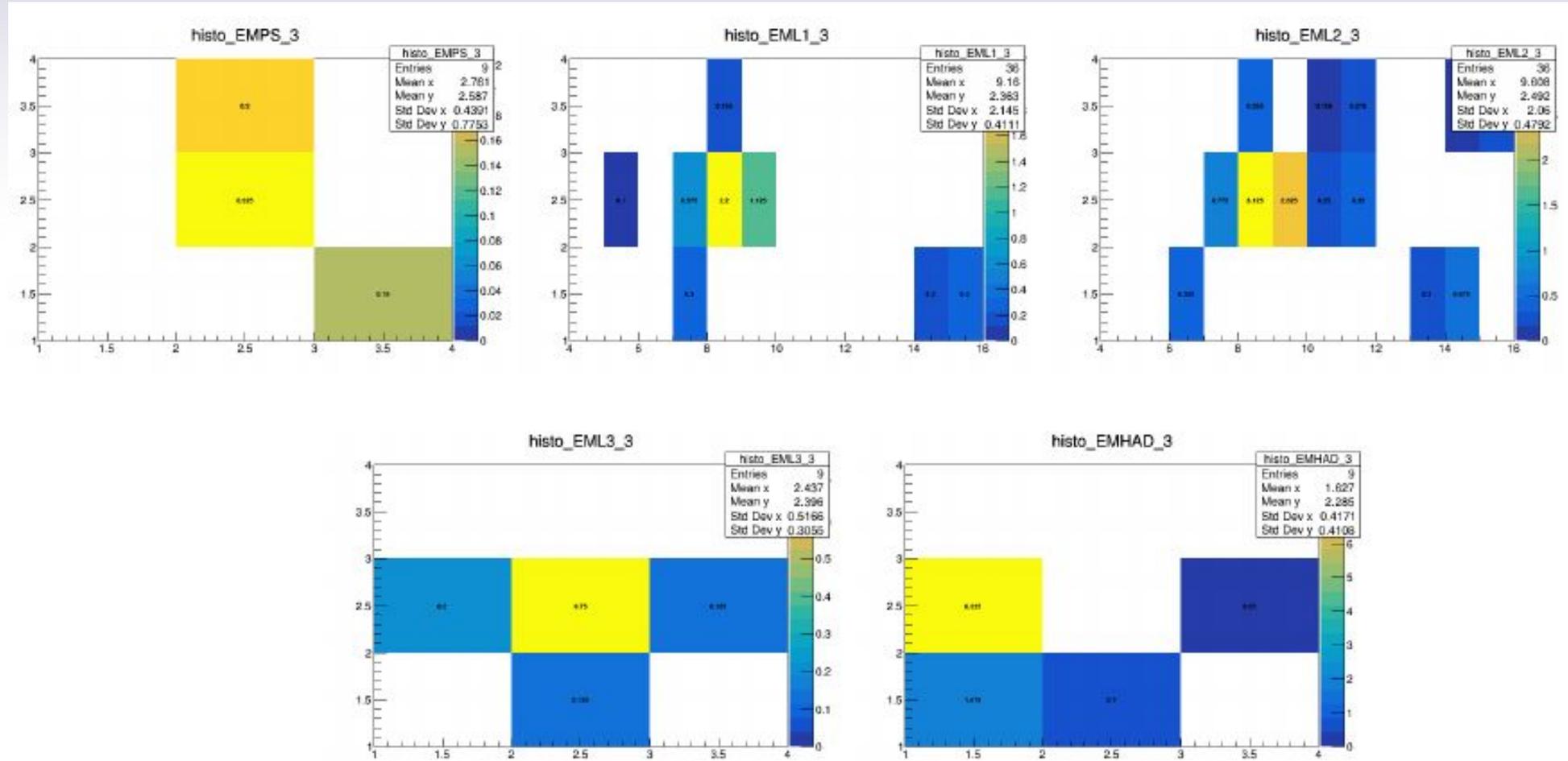
CANDIDATO 2 a jet





Cómo diseñamos un trigger? Estrategia de L1 trigger

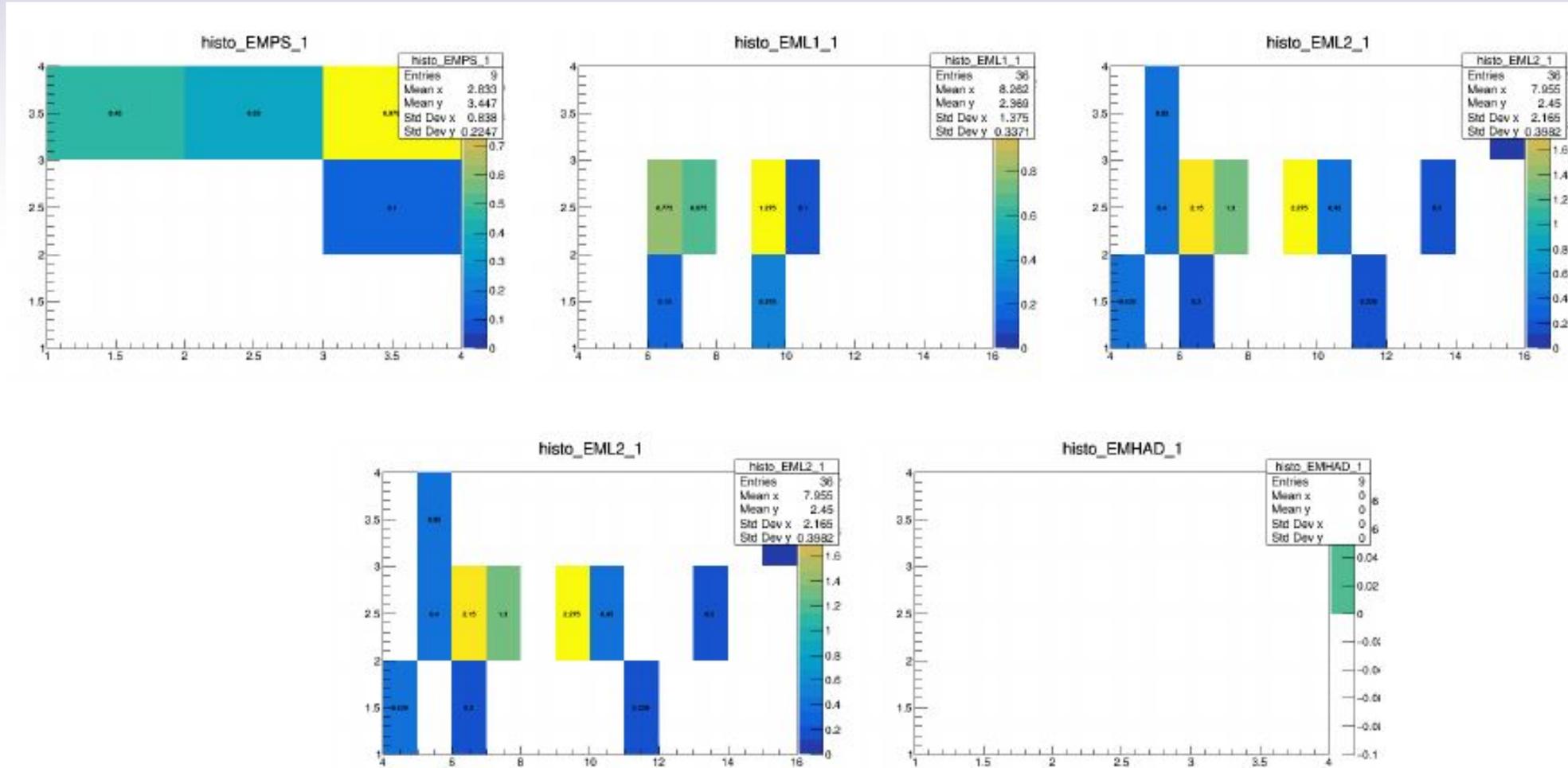
CANDIDATO 3 a jet





Cómo diseñamos un trigger? Estrategia de L1 trigger

CANDIDATO 4 a jet



QUIZ



Ejemplo: taus en L1

- Cuántos candidatos hay que mirar? 4? 100? 1000?
- Cuantos candidatos a rechazar (jets) hay que mirar?
- Podemos automatizar la forma de sumar energía (clustering)?
- Qué limitaciones podemos encontrar?

DAQ



- En tu experimento, lo que necesitas es una representación lógica del fenómeno que querés estudiar
- Hasta ahora contamos:
 - Cómo usamos detectores para interactuar con ese fenómeno
 - Cómo traducimos las señales para obtener de eso magnitudes eléctricas
 - Cómo las interpreta el sistema de trigger para decidir si las guarda
- Generar una descripción completa del evento para almacenar:
- DAQ: (data acquisition) es ese proceso
 - Si el sistema de trigger le da el OK, colecta la información de cada subdetector, la formatea y la almacena en disco para ser analizada



DAQ y Trigger son sistemas fuertemente interrelacionados

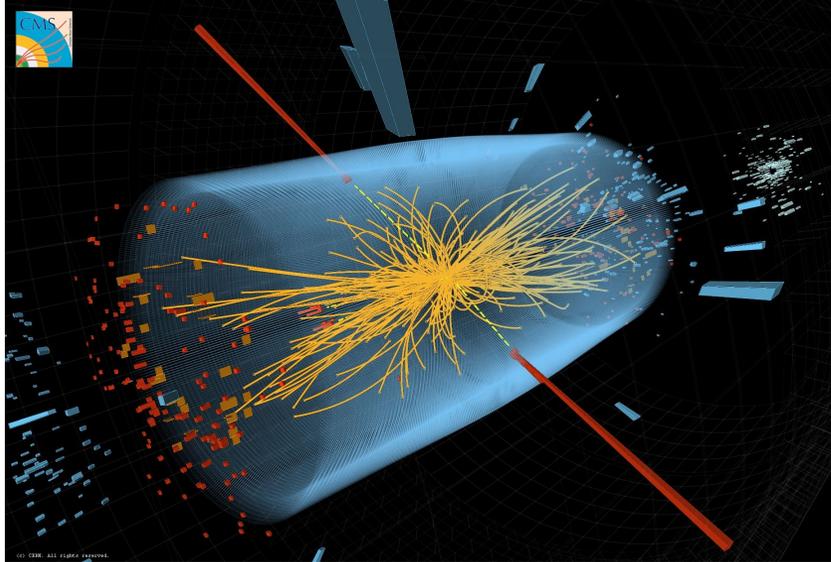
- Las limitaciones de uno son condiciones de trabajo del otro
- FrontEnds: Máxima velocidad de lectura / latencia de cada subdetector -> Limita el Rate
 - Por encima de eso: deadtimes
- DAQ: ancho de banda máximo de escritura -> Limita el Rate
 - Por encima de eso -> dead times por "Presión de retorno"
- Almacenamiento/CPU disponible: -> Implica un Rate máximo
 - Por encima de eso -> Llenás tu almacenamiento muy pronto o no te dan los recursos de cómputo para analizar tus datos



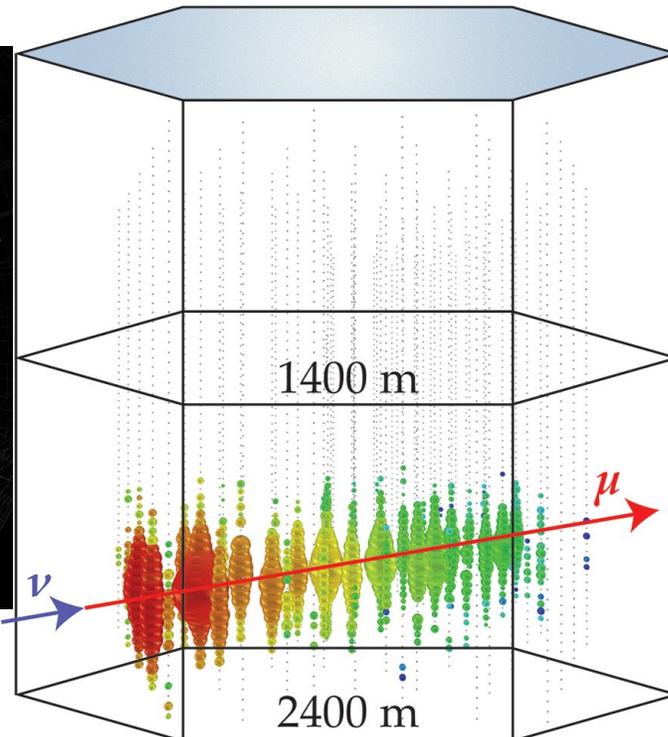
Definición de Evento

- Evento: es la instancia en el detector del fenómeno que querés estudiar
 - CMS: el resultado de la colisión de un paquete de protones
 - LIGO: un ripple en la lectura de los datos del interferómetro
 - IceCube: Radiación Cherenkov de una partícula cargada detectada en los PMTs

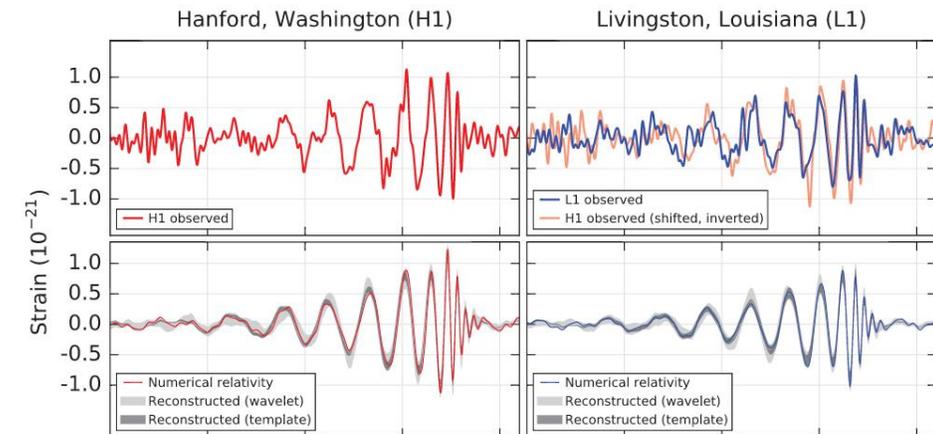
CMS



IceCube



LIGO





Los datos son la naturaleza hablándonos

- El DAQ termina en el almacenamiento
- El almacenamiento es Toda una disciplina en sí misma
 - ¿Qué tecnologías usás?
 - ¿Dónde guardás los datos?
 - ¿Cómo los guardás?
 - **¡¿Cómo hacés para que no se pierdan?!**
 - ¿Cómo los distribuís?
 - ¿Cómo los accedés?
- Experimentos de LHC:
 - Worldwide LHC Computing Grid [WLCG](#)



- Una infraestructura distribuida de almacenamiento y cómputo
- Mantenido por ~170 centros
- Transmisión de datos: ~50GB/s

Procesamiento de datos, simulaciones y calibración



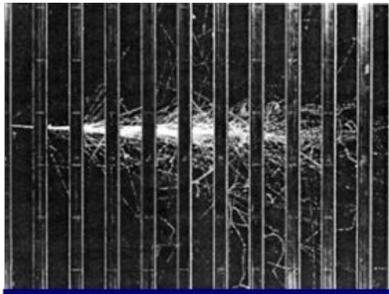
Formateo de datos:

- Cuando almacenás tus datos, elegís una convención : Formato
- Datos Crudos (los bytes de cada subdetector) los almacenás con algún formato:
 - [Encabezado de evento][Encabezado SD1][Datos SD1...][Fin SD1][Encabezado SD2][Datos SD2...][Fin SD2][.....][Fin Evento]
- SOFTWARE:
 - La lectura de esos datos crudos
 - Interpretación en términos de señales en cada subdetector (Reconstrucción)
 - Calibración de tus datos
 - Interpretación en términos de los observables físicos que buscas (Identificación)
 - Análisis de esos datos



Reconstrucción 1

- Una vez que tenés los datos a mano -> Interpretarlos como representación lógica de observables físicos

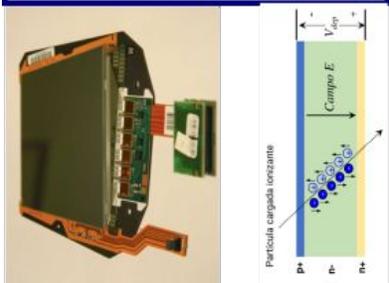


Frontend

→ 2, 5, 3, 6, 10, 32, 54, 100, 90, 95, 93, ...

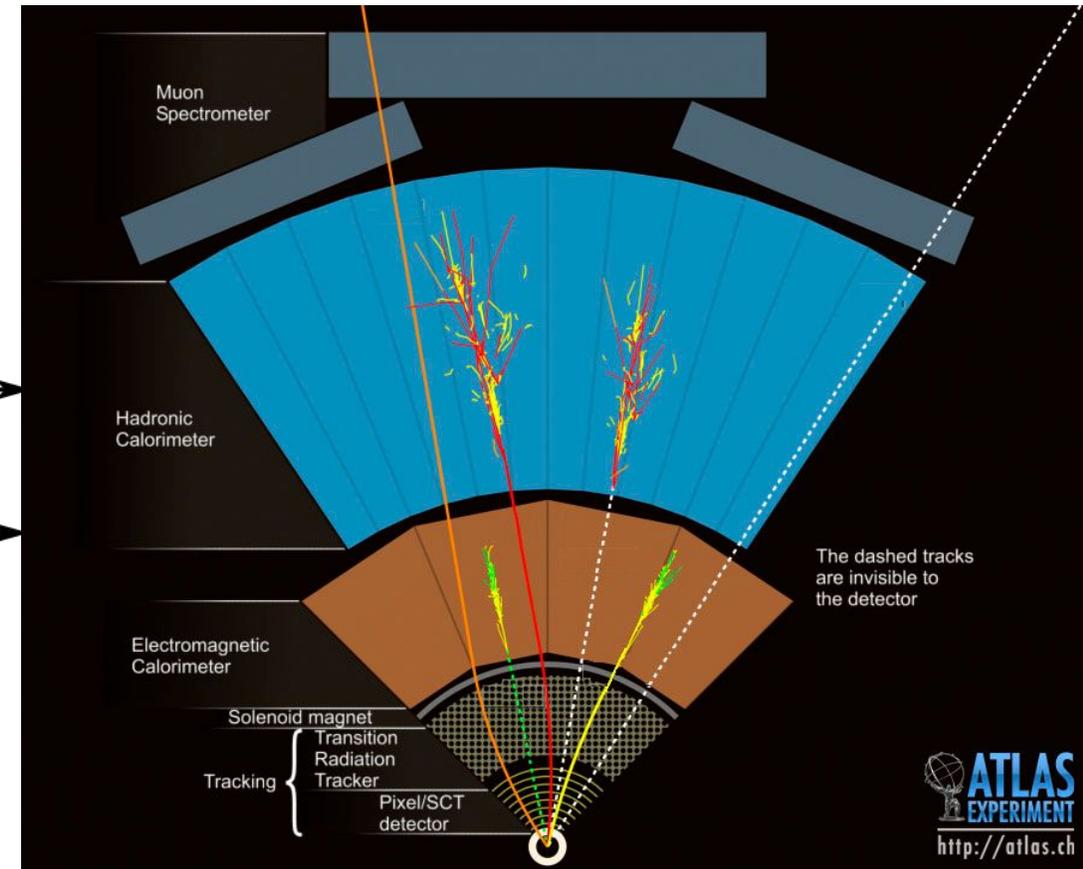
Frontend

→ 0, 0, 13, 32, 154, 87, 109, 59, 35, ...



Tren de datos de los calorímetros

Tren de datos de los detectores de trazas

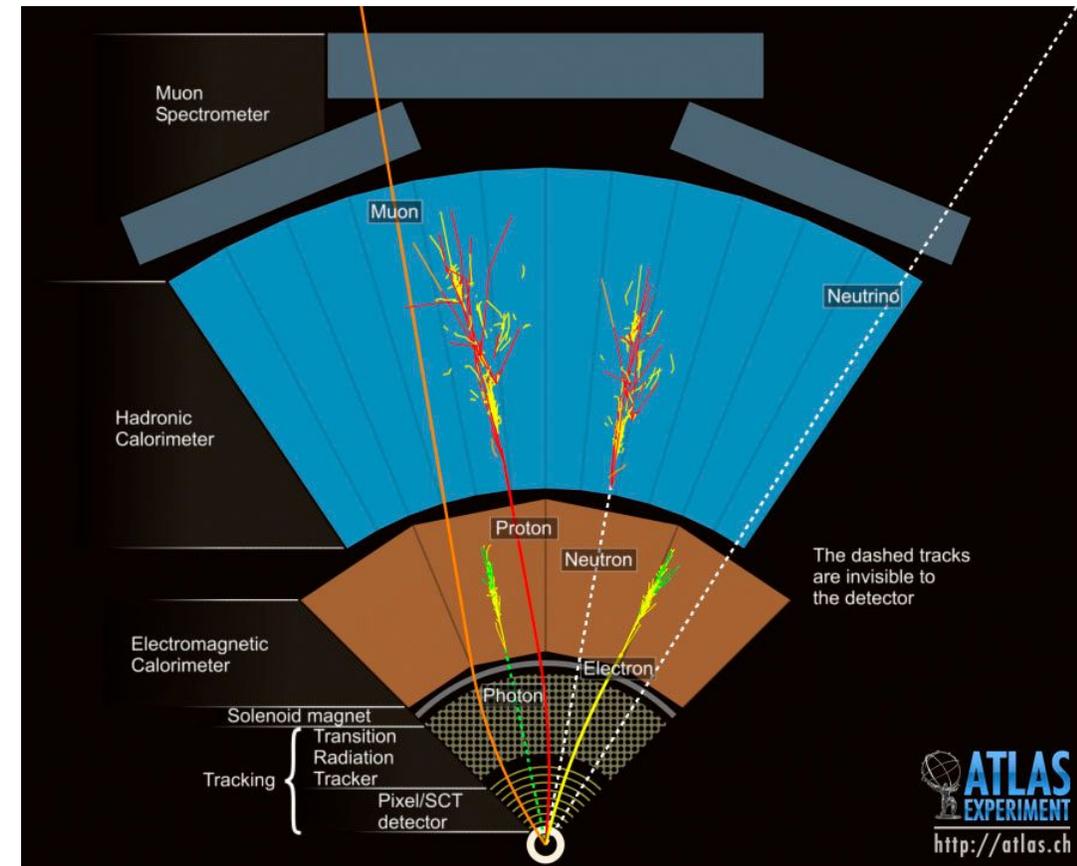




- Con la lista de los objetos que reconstruiste y calibraste:
 - Los combinás para **reconstruir** candidatos de interés

En ATLAS/CMS

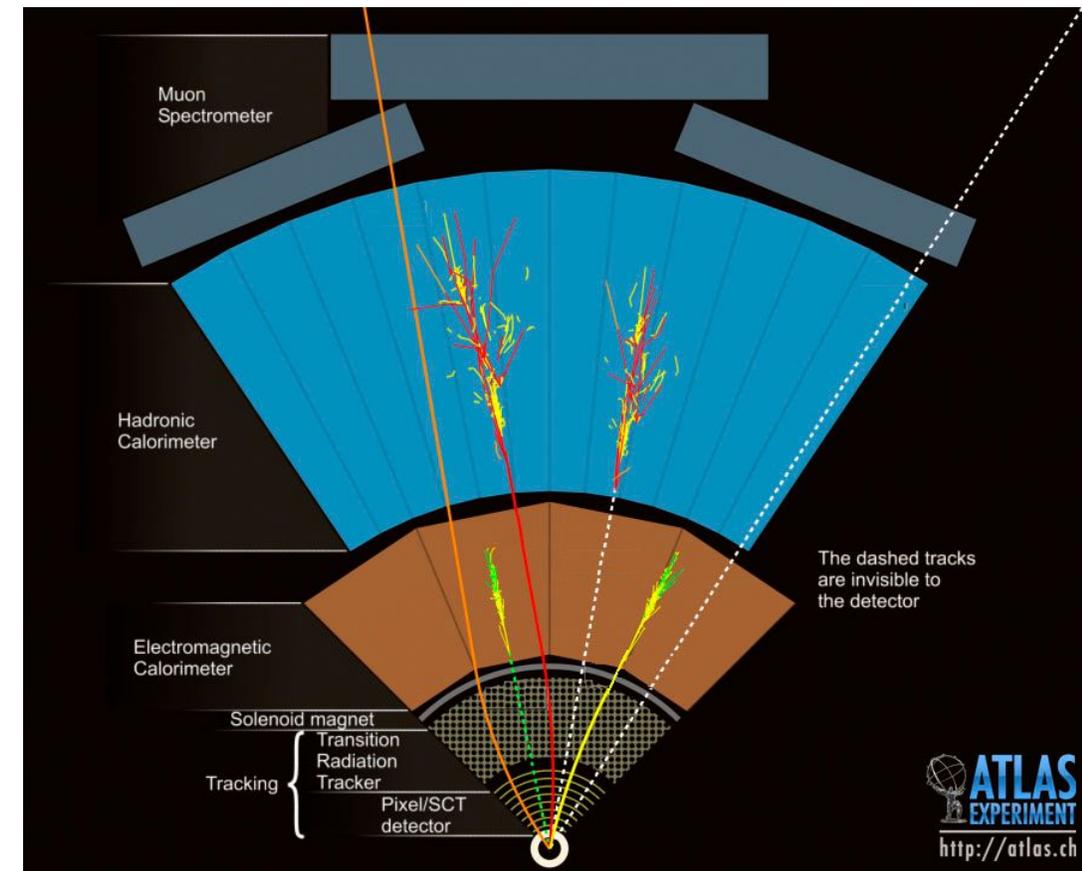
- Combinás trazas e información de los calorímetros para reconstruir partículas cargadas (electrones)
- Señal de las cámaras de muones para identificar muones
- Señal en calorímetro EM sin trazas -> Fotones





Tus observables físicos **TIENEN** que estar bien calibrados

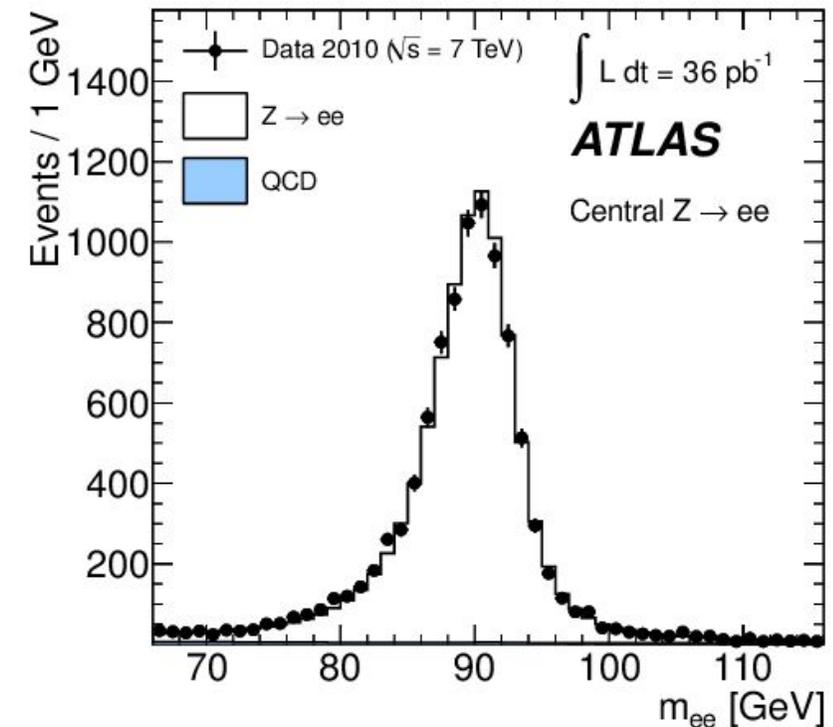
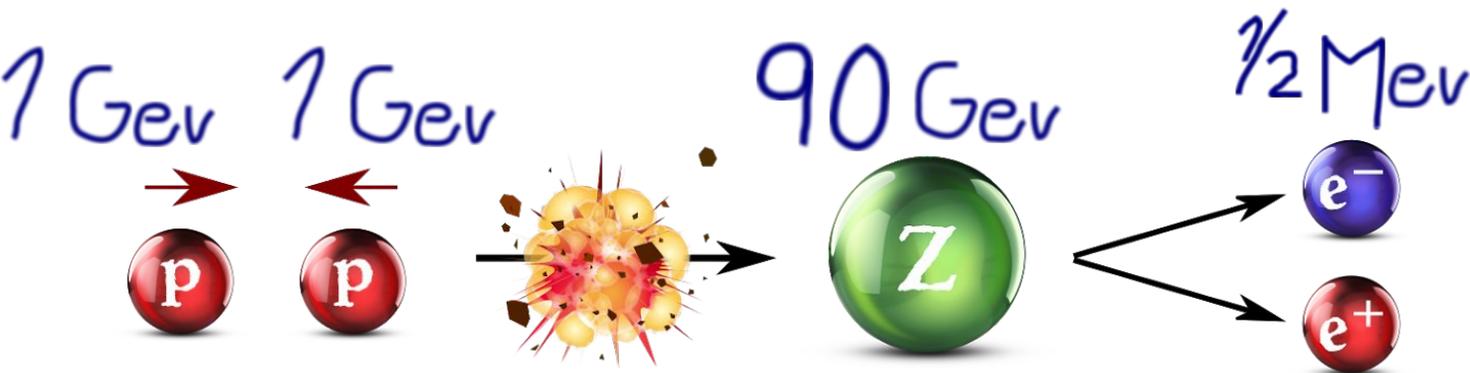
- Las trazas se curvan por el campo magnético
 - Descripción PRECISA del valor del campo en todo el espacio y la posición de cada pieza del detector
 - ¿A que pT corresponde el electrón reconstruido?
- Depósito en el calorímetro
 - ¿Cuánta energía tenía el candidato fotón que se frenó ahí?
- Identificación





Calibración

- Es el proceso por el cual ajustas tu instrumento de medición para que obtengas las mediciones de tu observable en los valores y unidades "correctos"
- En los experimentos de HEP esta calibración está a veces basada en datos y otras veces en simulaciones
- Ejemplo de calibración en datos: $Z \rightarrow e^+e^-$:
 - la masa invariante de e^+e^- tiene que ser ~ 91.19 GeV:





Otro ejemplo: EHT

Ejemplo de Event Horizon Telescope

<https://arxiv.org/abs/1906.11240>

- Es una colaboración de grandes facilidades de RadioTelescopios enlazados
- Un solo radiotelescopio virtual del diámetro terrestre

- Datos crudos
- Reconstrucción
- Calibración
- Identificación



meta-data

data reduction

schedule

Antenna voltages

[PB]

*Earth model
clock model*

Correlated data

[GB]

*sensitivity estimate
gain variations
network constraints*

Calibrated data

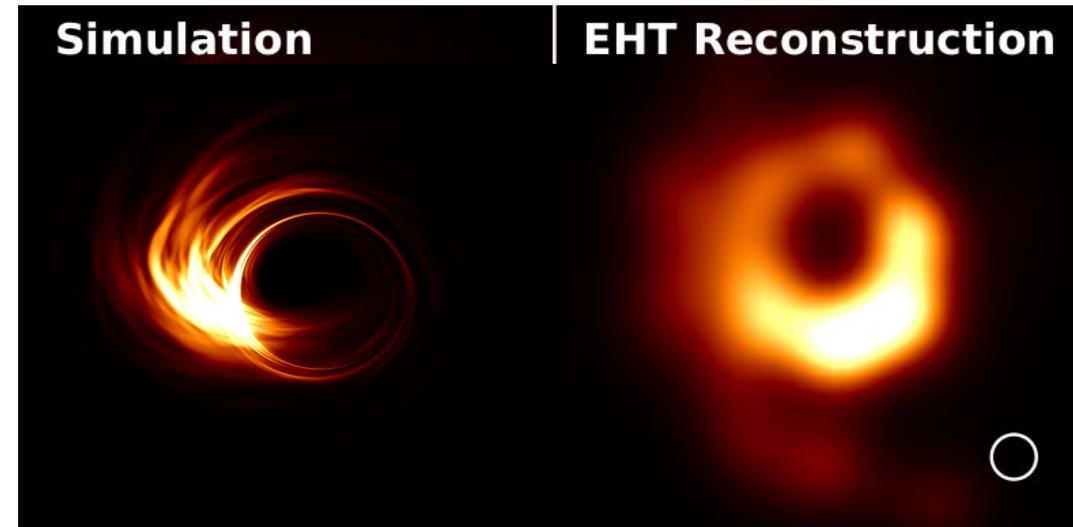
[MB]

additional priors

analysis



- Si bien los experimentos buscan tomar datos de la naturaleza, necesitan de Simulaciones
 - Entender las respuestas de tu detector ante tu señal
 - Desarrollar técnicas de limpieza de fondo
 - Desarrollar técnicas de calibración
 - Calcular sistemáticos
- Tesis de diploma de Jose -> el efecto de los servicios (cables, soportes, material muerto)
-> Descripción precisa del detector
- En el EHT, durante muchísimo tiempo se estudió como sería la observación a partir de simulaciones
- El algoritmo de reconstrucción se probó exhaustivamente mediante simulaciones



Análisis físico de los datos procesados



¿Cómo veo los datos que medí?

- Al final de la adquisición de datos, queremos ver las cantidades que medimos y usarlas
 - Los analistas recibimos una muestra en algún formato
 - El más común en física de altas energías es [ROOT](#)
 - Ese archivo va a contener para cada evento, información de cada una de las partículas
 - En general, aplicamos una serie de requisitos para reducir el tamaño de las muestras y que sea mas fácil/rápido trabajar



Ejemplo de cómo se ven los datos para analizar

- La información de un archivo en crudo para un evento se ve así:

EventInfo__NominalAuxDyn.CorrectedAndScaledAvgMu = 54.5
EventInfo__NominalAuxDyn.CorrectedAndScaledMu = 54.5
EventInfo__NominalAuxDyn.CorrectedAvgMu = 54.5
EventInfo__NominalAuxDyn.CorrectedMu = 54.5
EventInfo__NominalAuxDyn.NVtx2Trks = 22
EventInfo__NominalAuxDyn.PileupReweight = 6.14558
EventInfo__NominalAuxDyn.RandomRunNumber = 338263
EventInfo__NominalAuxDyn.RhoEventShape = 10442
EventInfo__NominalAuxDyn.ZPV = -19.7683
EventInfo__NominalAuxDyn.actualInteractionsPerCrossing = 54.5
EventInfo__NominalAuxDyn.averageInteractionsPerCrossing = 54.5
EventInfo__NominalAuxDyn.eventNumber = 102546
EventInfo__NominalAuxDyn.hasPV = 1
EventInfo__NominalAuxDyn.isMC = 1
EventInfo__NominalAuxDyn.lumiBlock = 55



Ejemplo de cómo se ven los datos para analizar

```
EventInfo__NominalAuxDyn.mcChannelNumber = 345919
EventInfo__NominalAuxDyn.mcEventNumber = 102546
EventInfo__NominalAuxDyn.passHLT_2e12_lhloose_L12EM10VH = 0
EventInfo__NominalAuxDyn.passHLT_2e17_lhvloose_nod0 = 0
EventInfo__NominalAuxDyn.passHLT_e120_lhloose = 0
EventInfo__NominalAuxDyn.passHLT_e140_lhloose = 0
EventInfo__NominalAuxDyn.passHLT_e140_lhloose_nod0 = 0
EventInfo__NominalAuxDyn.passHLT_e17_lhloose_mu14 = 0
EventInfo__NominalAuxDyn.passHLT_e17_lhloose_nod0_mu14 = 0
EventInfo__NominalAuxDyn.passHLT_e17_lhmedium_iloose_tau25_medium1_tracktwo = 0
EventInfo__NominalAuxDyn.passHLT_e17_lhmedium_nod0_ivarloose_tau25_medium1_tracktwo = 0
EventInfo__NominalAuxDyn.passHLT_e17_lhmedium_nod0_ivarloose_tau25_medium1_tracktwoEF = 0
```

- Además tenemos los valores de cada variable para cada partícula

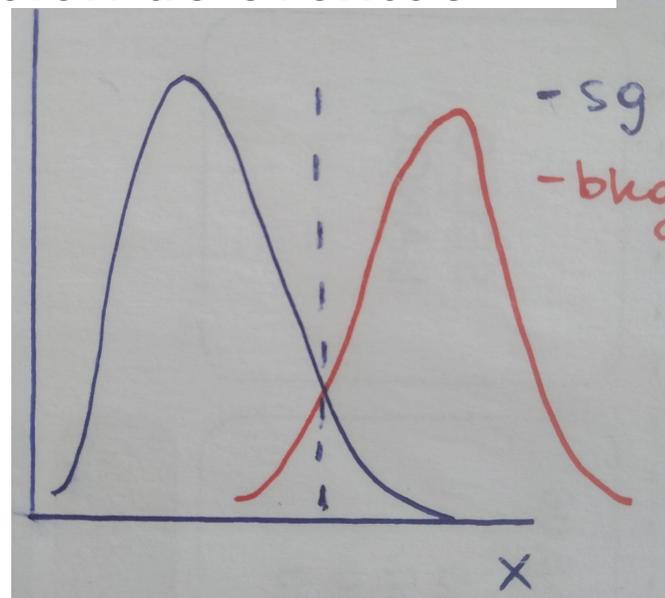
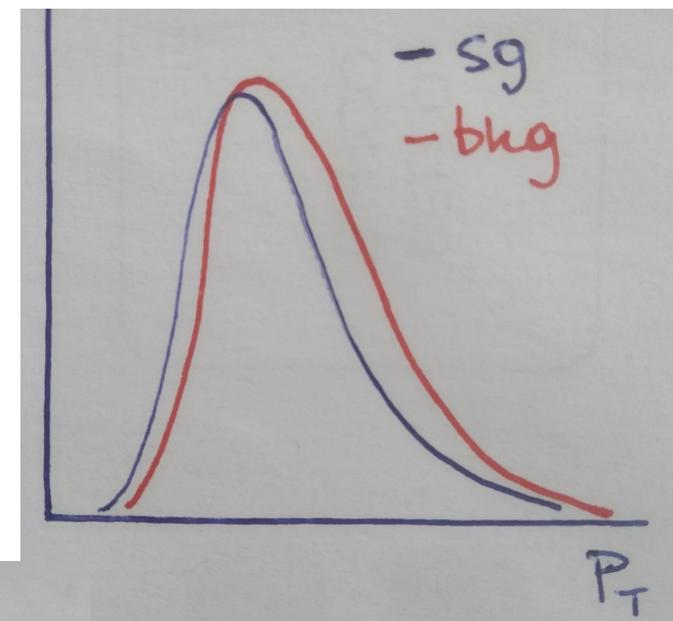


Ejemplo de cómo se ven los datos para analizar

Photons__NominalAuxDyn.convType = (vector<int>*)0x55d1c17fd930
Photons__NominalAuxDyn.eta = (vector<float>*)0x55d1c182f250
Photons__NominalAuxDyn.isAmbiguous = (vector<int>*)0x55d1c183dc60
Photons__NominalAuxDyn.isEM_Loose = (vector<int>*)0x55d1c183e000
Photons__NominalAuxDyn.isEM_Medium = (vector<int>*)0x55d1c183e300
Photons__NominalAuxDyn.isEM_Tight = (vector<int>*)0x55d1c1869e80
Photons__NominalAuxDyn.isFixedCutTightCaloOnlyIso = (vector<int>*)0x55d1c1878920
Photons__NominalAuxDyn.isLoose = (vector<int>*)0x55d1c1878d60
Photons__NominalAuxDyn.isMedium = (vector<int>*)0x55d1c18960b0
Photons__NominalAuxDyn.isTight = (vector<int>*)0x55d1c1896490
Photons__NominalAuxDyn.looseEffSF = (vector<double>*)0x55d1c18966d0
Photons__NominalAuxDyn.m = (vector<float>*)0x55d1c18b3aa0
Photons__NominalAuxDyn.mediumEffSF = (vector<double>*)0x55d1c18d5de0
Photons__NominalAuxDyn.partIndex = (vector<int>*)0x55d1c18d6180
Photons__NominalAuxDyn.phi = (vector<float>*)0x55d1c18f8580
Photons__NominalAuxDyn.photonFixedCutTightEffSF = (vector<double>*)0x55d1c18f8980
Photons__NominalAuxDyn.pt = (vector<float>*)0x55d1c191ac40



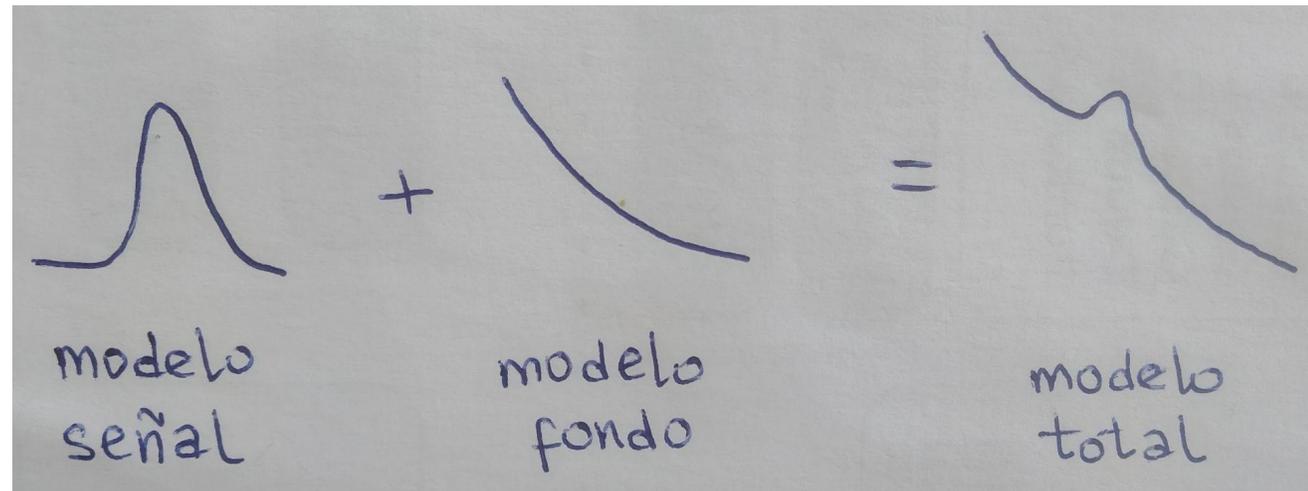
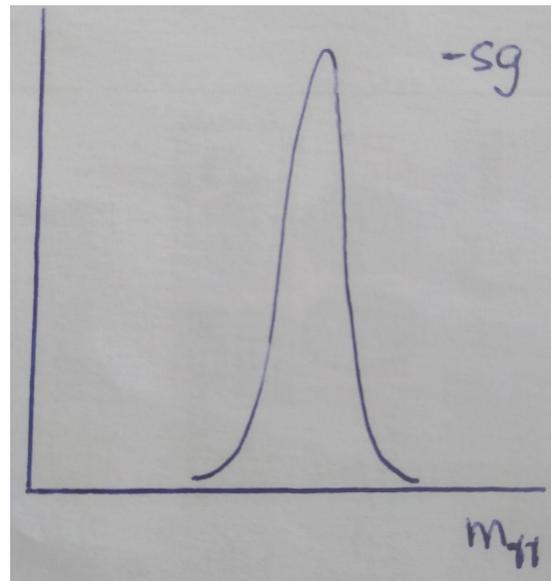
- No podemos mirar uno a uno cada evento
- Visualizamos en **histogramas**
- Caracterizar cada partícula relevante
 - Variables **cinemáticas**
 - Variables **discriminatorias** (Señal vs fondo)
 - Nos permiten hacer la selección de eventos
- Intentamos maximizar S/\sqrt{B}
- Usualmente : “cut-based”
- Transicionando a ML (BDT)





Identificar variable para buscar una señal: ejemplo típico, masa invariante

- Como esperamos ver la distribución para fondo y para señal?
- Modelar señal y fondo

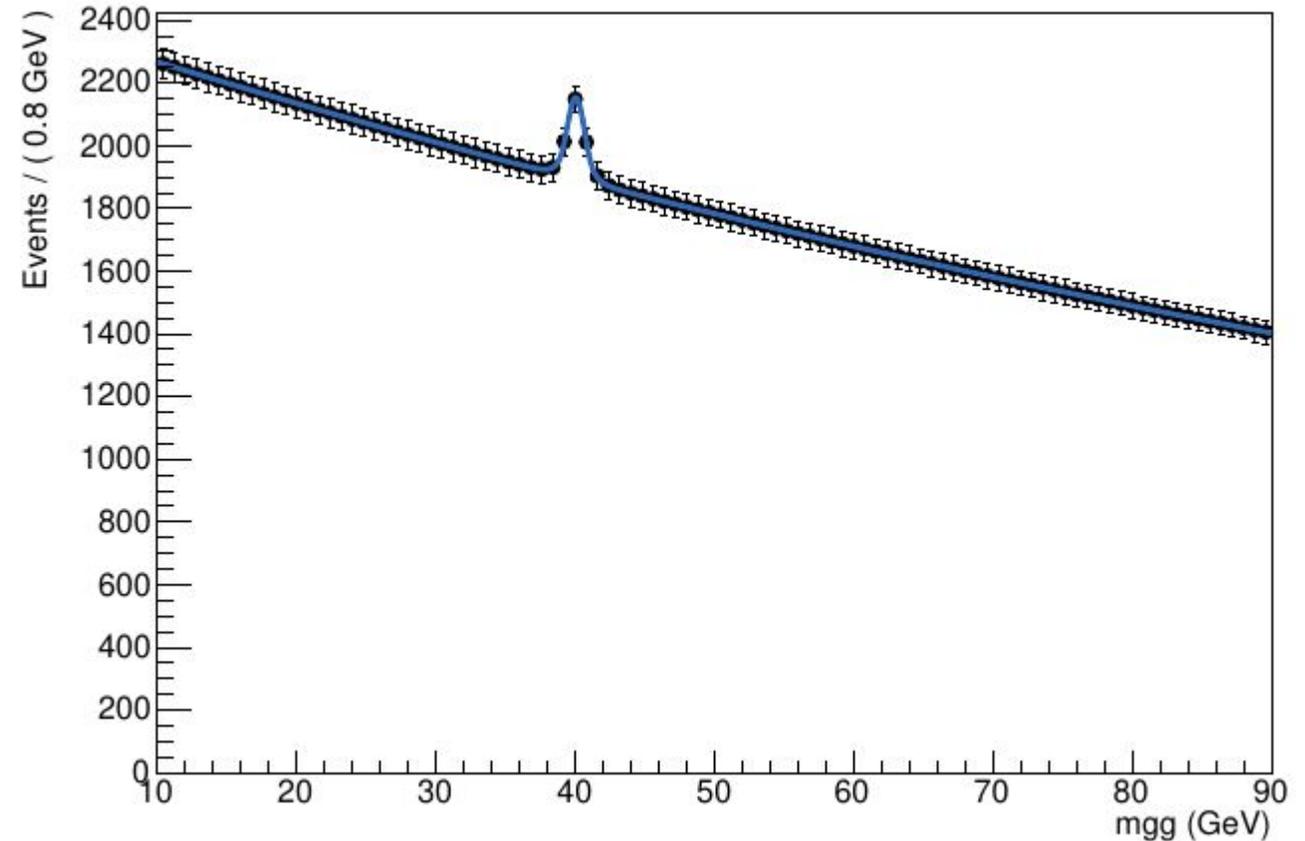




Ajuste a datos con el modelo:

$$n_{\text{signal}} \text{PDF}_{\text{signal}} + n_{\text{bkg}} \text{PDF}_{\text{bkg}}$$

- Análisis estadístico usando la información del ajuste
→ Descubrimiento de nueva partícula!





TEMARIO:

- Repaso de detectores ✓
 - Interacción de partículas con los subdetectores
- Front-end electronics: ✓
 - Conversión de la interacción en pulsos eléctricos
 - Digitalización
 - DAQ
- Trigger ✓
 - ¿Qué eventos se guardan y que no?
- Diseño del trigger ✓
- Tratamiento de datos recolectados ✓
- Procesamiento de datos ✓
 - Simulación, reconstrucción, identificación, calibración
- Visualización y análisis ✓

Referencias



ATLAS Fact sheet: [ATLAS fact sheet.pdf](#)

CMS Fact sheet: [CMS Fact Sheet](#)

Pierre Auger Observatory: [Nuclear Instruments and Methods in Physics Research A 798 \(2015\) 172–213](#)

[PHD thesis, R. J. Aliaga. 2015, Uiversidad de Valencia](#)

Event Horizon Telescope: [https://arxiv.org/abs/1906.11240](#)



<http://laconga.redclara.net>



contacto@laconga.redclara.net



lacongaphysics



Latin American alliance for
Capacity buildiNG in Advanced physics

LA-CoNGA physics



Cofinanciado por el
programa Erasmus+
de la Unión Europea

El apoyo de la Comisión Europea para la producción de esta publicación no constituye una aprobación del contenido, el cual refleja únicamente las opiniones de los autores, y la Comisión no se hace responsable del uso que pueda hacerse de la información contenida en la misma.



Latin American alliance for
Capacity building in Advanced physics
LA-CoNGA physics



Cofinanciado por el
programa Erasmus+
de la Unión Europea

