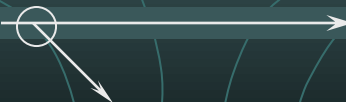




Introducción a

# Geant 4



E. Nácher, J. Balibrea

# Contenidos

- Características generales de Geant4
- Conceptos básicos en Geant4
- Estructura global de Geant4
- Ejemplo sencillo completo

# Características generales de Geant4



# ¿Qué es Geant4?

- Geant4 es un conjunto de clases de C++ para la simulación Montecarlo de detectores en HEP. Es el sucesor de GEANT3, las librerías de simulación en Fortran77 incluidas en *CERNLIB*.
- Geant4 es el rediseño de un gran paquete de software de HEP para adaptarlo a las nuevas necesidades experimentales usando un entorno *Orientado a Objetos*: se pasó del Fortran77 al C++
- Se aumentó el grado de funcionalidad y flexibilidad para cumplir muchos requisitos impuestos por la física nuclear, física médica, astrofísica...

# ¿Qué hace Geant4?

- Geant4, a través de los métodos definidos en sus clases:
  - Hace un seguimiento (*tracking*) de cada partícula primaria a través de los distintos volúmenes definidos por el usuario en pasos de longitud dinámica
  - En cada paso calcula: si se produce o no interacción, tipo de interacción, energía depositada en caso de que la haya, nueva dirección de la partícula, partículas secundarias generadas
  - Hace un seguimiento completo de todas las partículas secundarias generadas
  - Termina este *tracking* cuando la partícula tiene un rango en el material menor de un cierto valor: Cut

# ¿Qué hace Geant4?

¿Interacciona?



# ¿Qué hace Geant4?

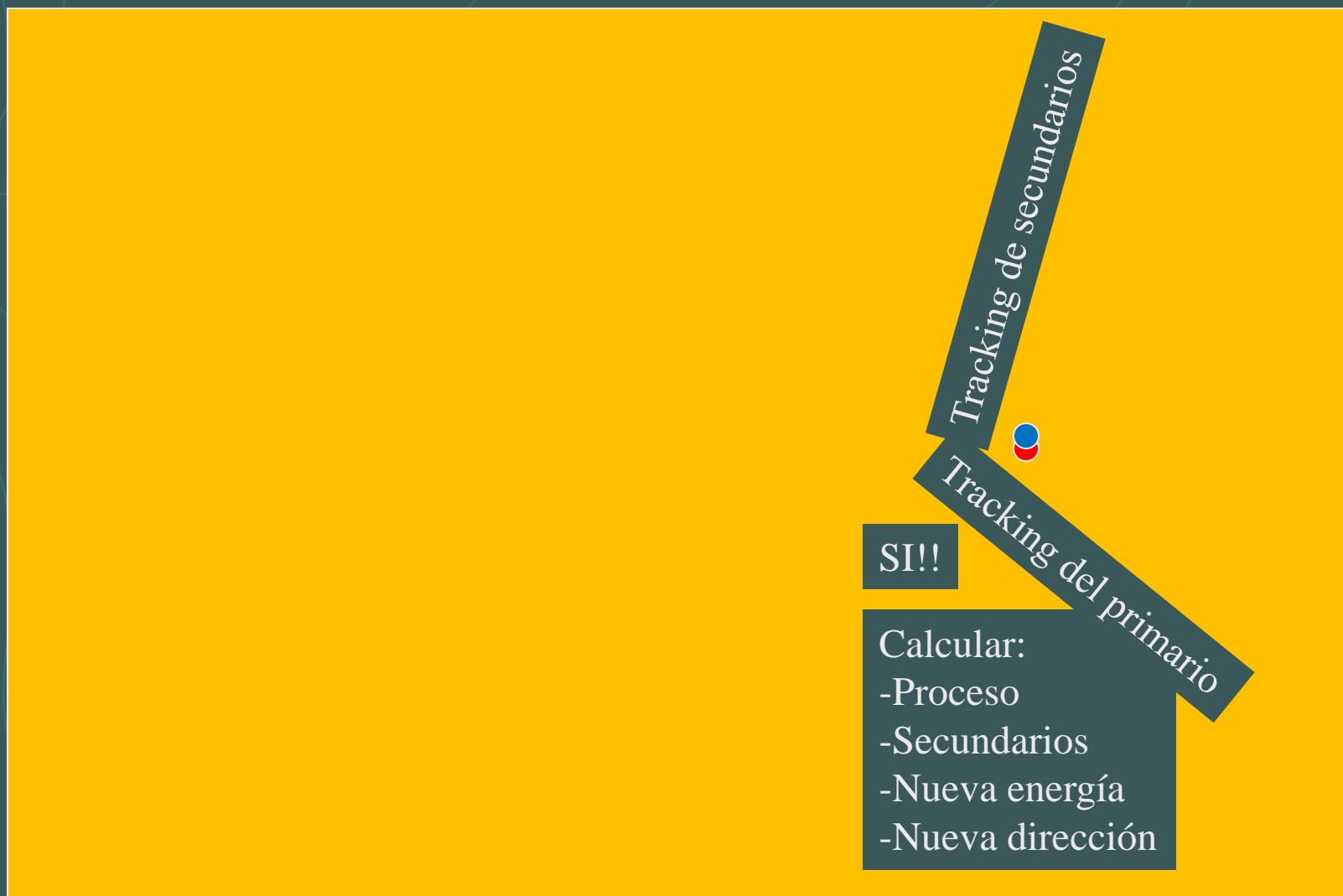
NO!!

¿Interacciona?





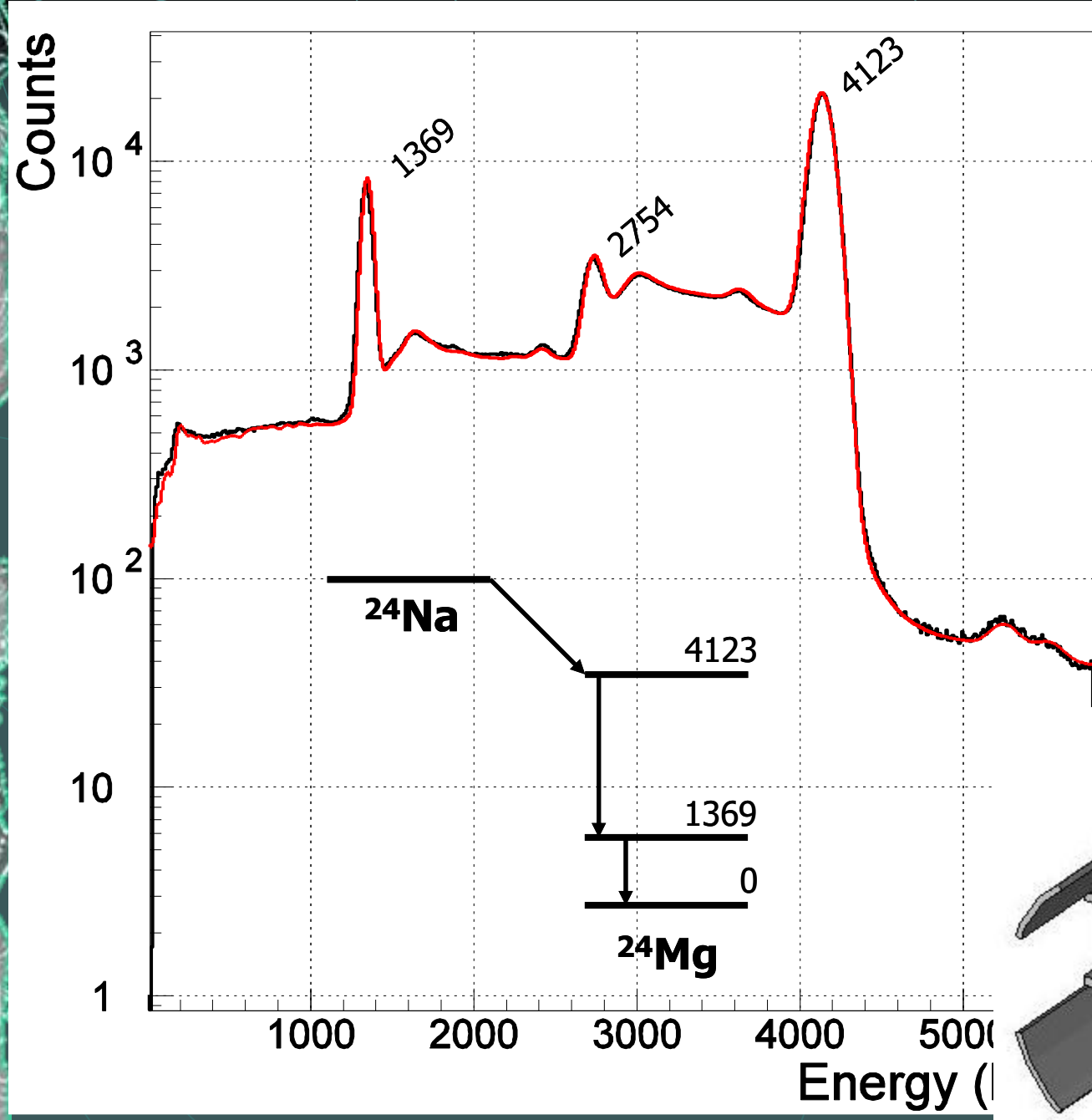
# ¿Qué hace Geant4?



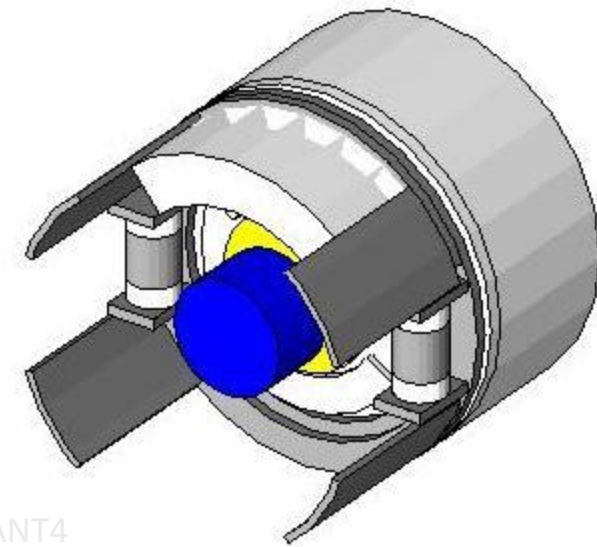


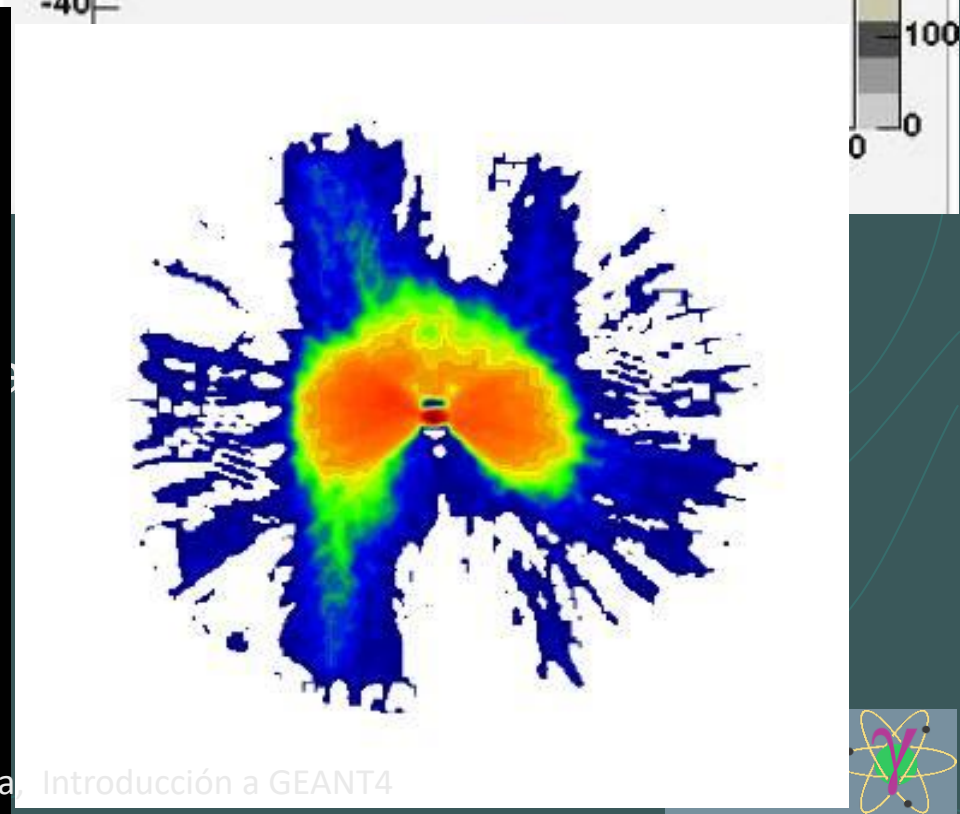
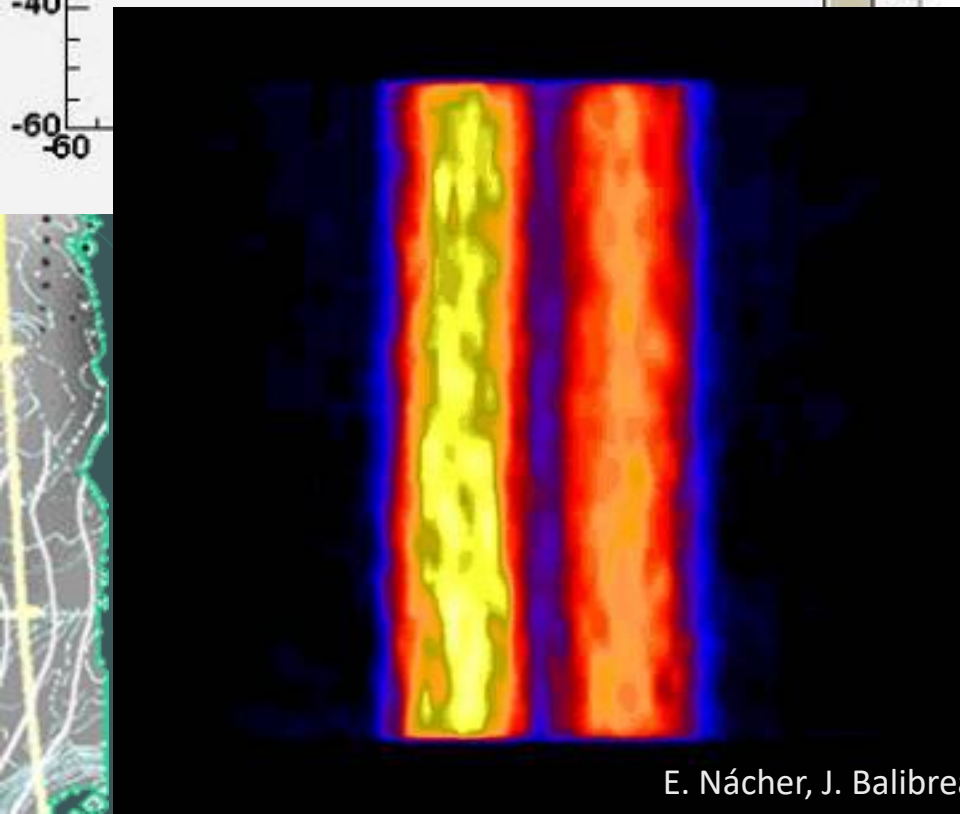
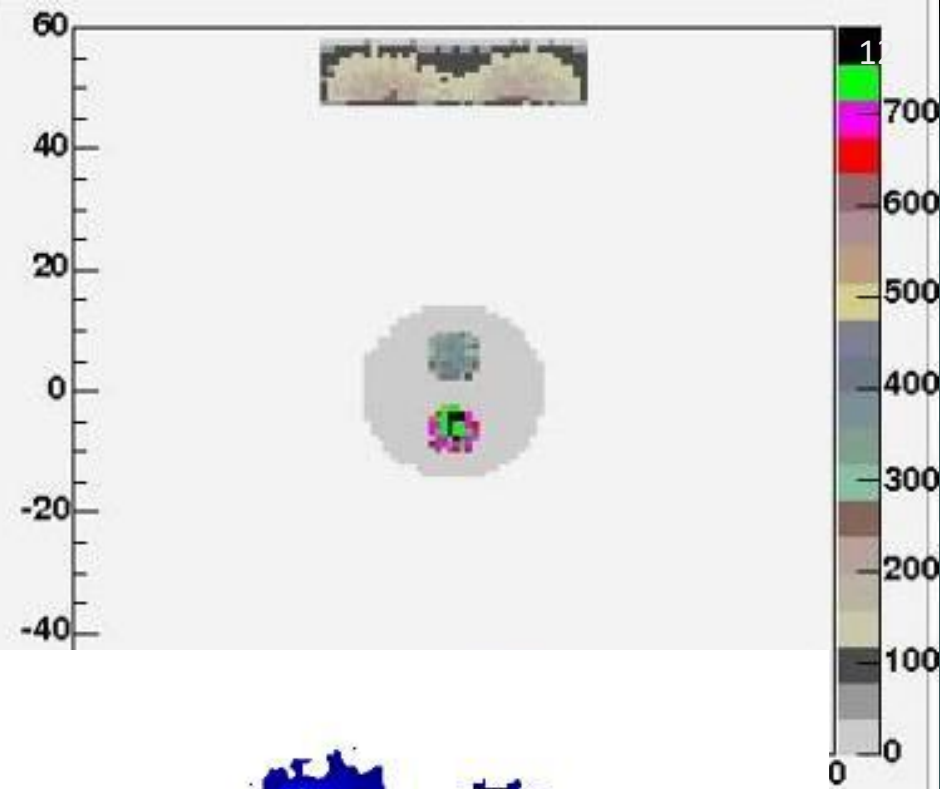
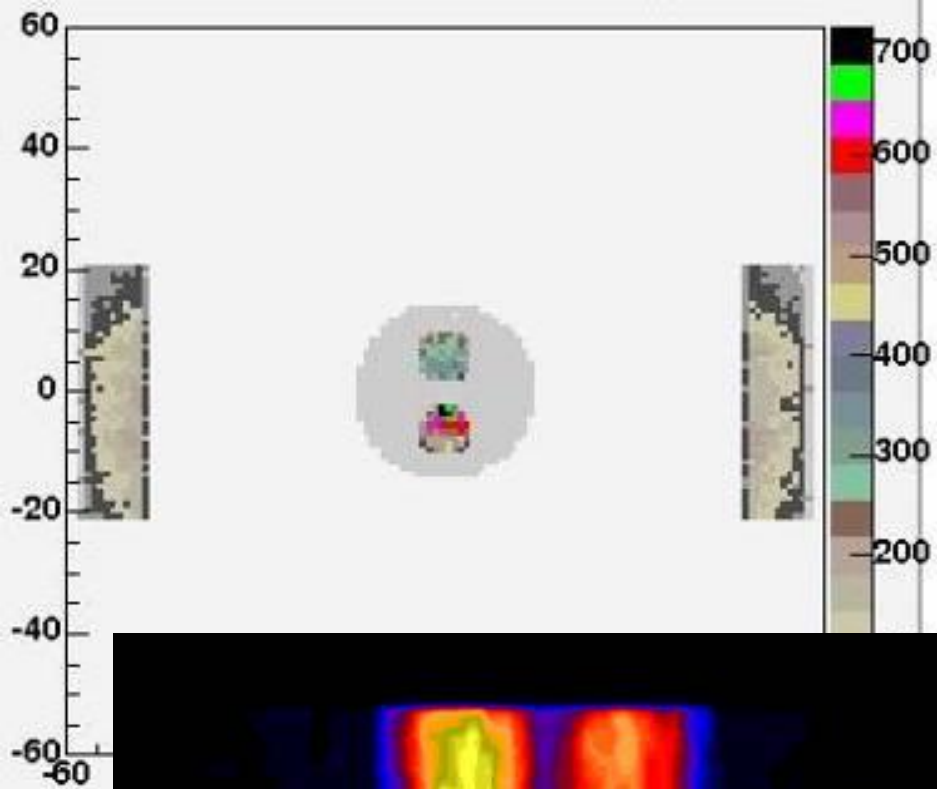
# Física en Geant4

- ❖ No se puede desarrollar un modelo válido para todo tipo de partículas y rango de energía
- ❖ Parte de la física implementada viene de teoría, otra parte de tablas de datos experimentales (secciones eficaces...), y otra de modelos fenomenológicos
- ❖ Cada tabla de datos o modelo físico tiene su propio rango energético de aplicación. Combinando varios se puede dar cobertura a un amplio rango de energías y partículas para distintas simulaciones
- ❖ El usuario elige el modelo más adecuado a la simulación particular (rango energético), así como los distintos procesos físicos relevantes (¿demasiada flexibilidad?)



de datos  
respuesta





# Conceptos básicos en Geant4



# Run

- Como analogía con los experimentos en HEP, un *run* de Geant4 comienza con: “Beam On”.
- Conceptualmente, el *run* es una colección de *eventos* que comparten el mismo detector y condiciones físicas.
- Durante el *run*, el usuario no puede cambiar:
  - La geometría del sistema de detección
  - La física de la simulación



# Event

- Al principio de su procesado, un *event* contiene partículas primarias (y su posición, momento inicial y energía). Estos primarios se ‘lanzan’, y se crea un ‘stack’ que contiene toda la información del evento.
- Cuando termina el procesado del evento se vacía el ‘stack’.
- La clase G4Event representa un evento. Contiene los siguientes objetos al final de su procesado:
  - Lista de vértices y partículas primarias
  - Colección de Trayectorias (opcional)
  - Colección de Hits y partículas secundarias ...
  - La información que el usuario quiera añadir (p.ej. acumulada en cada *step*)

# Step

- Cada *step* tiene dos puntos (inicio y final), además de información tipo: “delta” de la partícula (energía depositada en el paso, tiempo de vuelo del paso, etc.).
- La longitud del paso la calcula el programa de manera dinámica en función del rango de la partícula en el material (el usuario puede limitar este cálculo).
- En cada punto se conoce el volumen en el que se encuentra la partícula. Si un paso cruza la superficie de



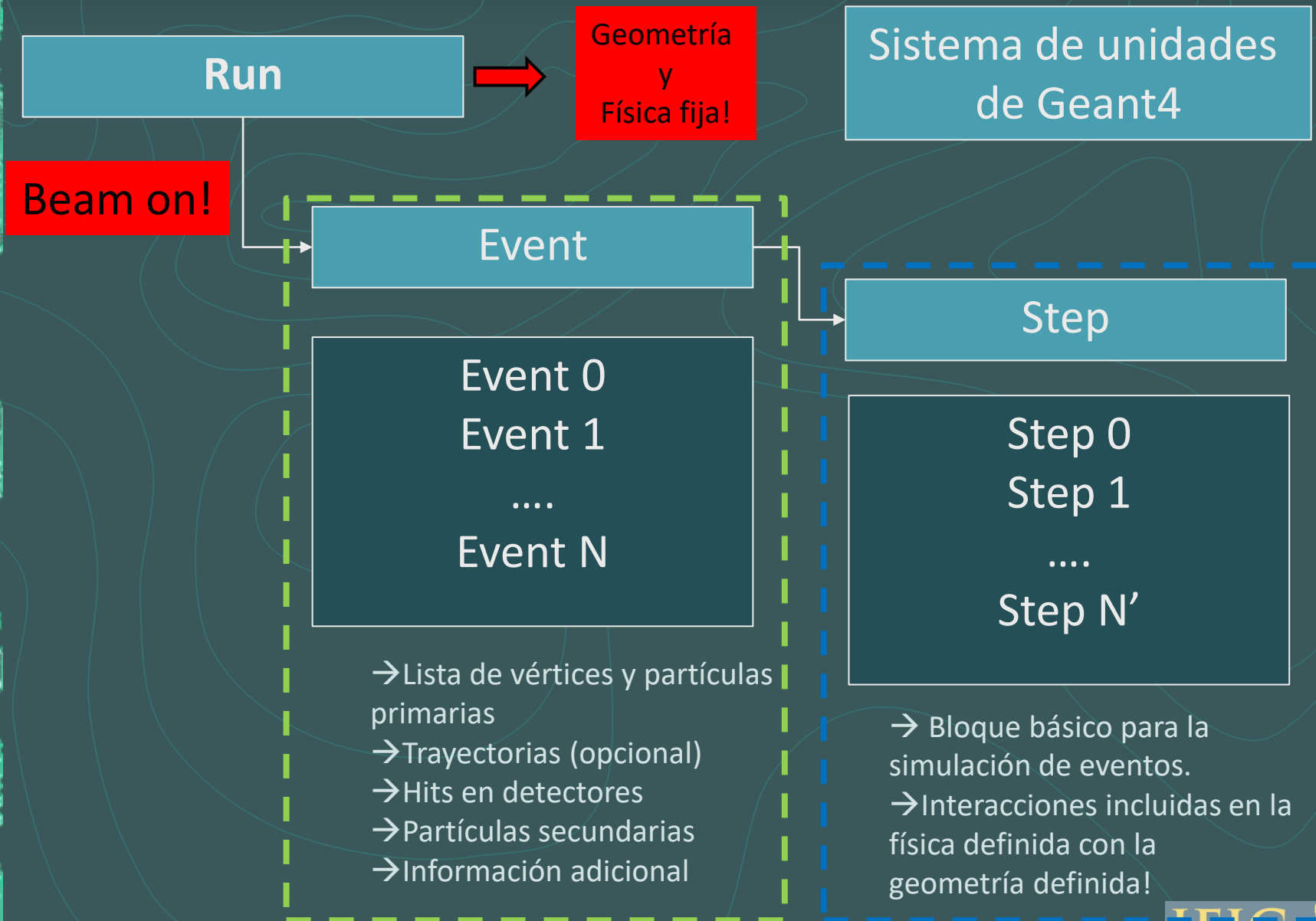


# Sistema de Unidades en Geant4

- Geant4 no tiene una unidad por defecto. Para introducir el valor de una magnitud física, una unidad debe estar multiplicando a un número .
  - Ejemplos:
    - double width = 12.5\*m;
    - double density = 2.7\*g/cm3;
  - Todas las unidades más comunes están incluidas.
  - El usuario puede definir unidades nuevas.
- Para obtener un valor en una cierta unidad debe dividirse la variable por esa unidad:  

```
G4cout << deltaE / MeV << " MeV" << G4endl;
```

# Resumen





# Estructura global de Geant4



# Funcionamiento de la aplicación G4

- Inicialización
  - Construcción de los materiales y geometría
  - Construcción de partículas, procesos físicos y cálculo de tablas de secciones eficaces
- “Beam-On” → Inicia el *run*
  - Event Loop
    - Se pueden lanzar más de un *run* con diferentes geometrías o tipo de partículas primarias...

# User classes

- Clases que debe definir el usuario
- Initialization classes
  - Se invocan en la inicialización
    - G4VUserDetectorConstruction
    - G4VUserPhysicsList
- Action classes
  - Se invocan durante el *event loop* (o el *run*)
    - G4VUserPrimaryGeneratorAction
    - G4UserRunAction
    - G4UserEventAction
    - G4UserSteppingAction

*User Mandatory Classes*

*User Optional Classes*

# Más información:

- Geant4 está continuamente desarrollandose (v11.0)
  - Nuevas funcionalidades, modelos, física... 😊
  - Versiones anteriores dejan de ser compatibles ☹
    - Nuevos estándares para C++: C++14/C++17/C++21
    - El código evoluciona! → Mejoras en rendimiento
- <https://geant4.web.cern.ch/>
  - <https://geant4-userdoc.web.cern.ch/UsersGuides/InstallationGuide/html/index.html>
  - <https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/html/index.html>
- Consejo → Si vais a usar GEANT4, usad linux:
  - <https://ubuntu.com/download>