

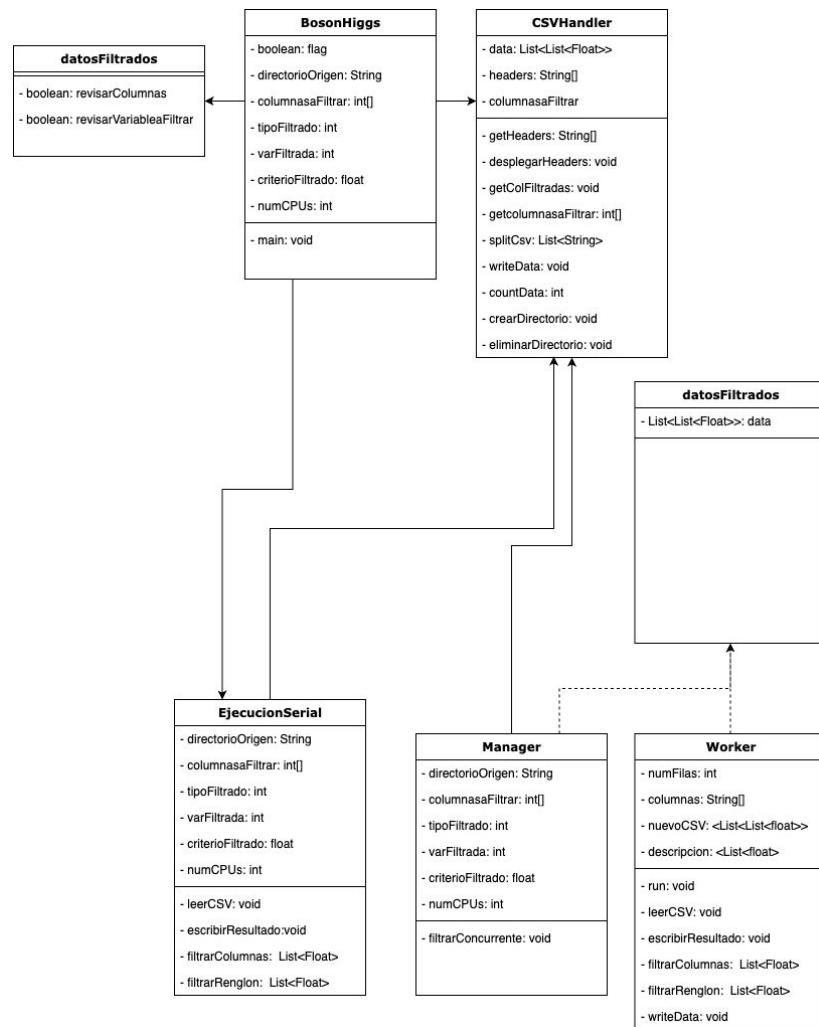
Bosón de Higgs

Maximiliano Galindo
Alejandro Mendoza Puig

Contexto

- En la búsqueda de nuevas partículas, los físicos emplean una herramienta conocida como acelerador de partículas.
- Estos aceleradores provocan la colisión entre partículas, generando así otras, como es el caso del bosón de Higgs
- Set de datos son variables simuladas de la colisión de partículas. Mostrando la diferencia cuando la colisión detectó un bosón de Higgs y cuando es otra partícula.

Diseño del programa



Clases

BosonHiggs.java

Clase principal que contiene el método main y la interfaz para que el usuario defina los parámetros de entrada.

CSVHandler.java

Clase encargada de hacer operaciones con los CSV y sus directorios. Crear/borrar directorios, contar número de líneas, dividir el archivo en n partes.

Manager.java

Clase encargada de crear el pool de workers, asignarles un CSV y esperar a que terminen para hacer join y unir los workers

Workers.java

Workers encargados de realizar la lectura de cada CSV y aplicar los filtros correspondientes. Corren de manera concurrente y utilizan candados para modificar la estructura de datos en la sección crítica.

DatosFiltrados.java

Clase con la estructura de datos que almacena toda la información que se ha encargado de filtrar cada uno de los workers

Funcionamiento general

1. Capturar información sobre el filtrado
2. División de CSV en N subarchivos
3. Manager crea pools de workers y asigna un CSV a cada uno
4. Workers (implementan Runnable) crean el filtrado de su CSV correspondiente
 - a. Escriben los resultados a la variable data.
5. Manager espera a que todos los workers hagan su tarea
6. Manager guarda la estructura de datos **data** en un nuevo CSV
7. Se elimina la carpeta con los CSV intermedios.

Resultados

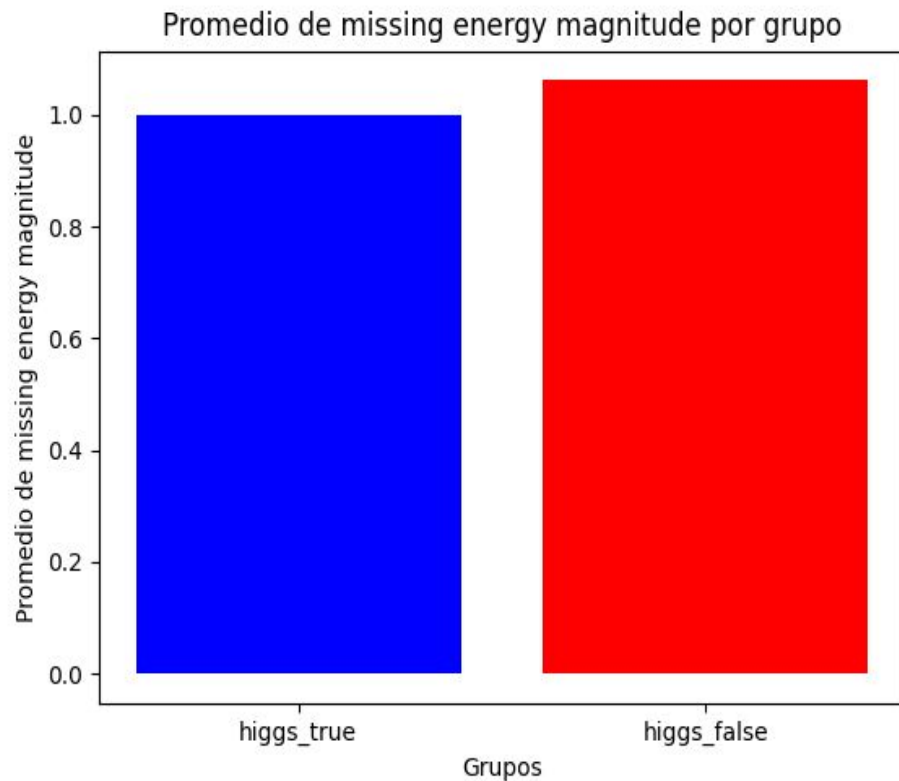
- El proceso evaluado fue únicamente considerando el filtrado del CSV sobre los criterios seleccionados.
- El correr de manera concurrente presentó recortes en los tiempos de ejecución de más de la mitad comparado con hacerlo serial.
- Se tuvo una aceleración promedio de 2.6

Secuencial/Concurrente	Tamaño original	Tamaño filtrado	Criterio de filtrado	Tiempo de ejecución (s)	Speedup
Concurrente	(25, 2,000,000)	(10: 324,342)	Col[1] <= 0.5	4.6	2.2
Serial	(25, 2,000,000)	(10: 324,342)	Col[1] <= 0.5	10.2	
Concurrente	(25, 2,000,000)	(20: 1,058,818)	Col[0] == 1	4.3	2.4
Serial	(25, 2,000,000)	(20: 1,058,818)	Col[0] == 1	10.5	
Concurrente	(25, 2,000,000)	(12: 18,941)	Col[10] <= 0.3	2.9	3.4
Serial	(25, 2,000,000)	(12: 18,941)	Col[10] <= 0.3	9.8	
Concurrente	(25, 2,000,000)	(6: 425,850)	Col[20] >= 1	3.4	2.9
Serial	(25, 2,000,000)	(6: 425,850)	Col[20] >= 1	10	
Concurrente	(25, 2,000,000)	(16: 924,453)	Col[11] > 0.1	5.1	2.6
Serial	(25, 2,000,000)	(16: 924,453)	Col[11] > 0.1	13.4	
Concurrente	(25, 2,000,000)	(21: 941,182)	Col[0] == 0	4.7	2.6
Serial	(25, 2,000,000)	(21: 941,182)	Col[0] == 0	12	
Concurrente	(25, 2,000,000)	(21: 2000,000)	Sin filtrado	5.8	2.6
Serial	(25, 2,000,000)	(21: 2000,000)	Sin filtrado	15.1	
Concurrente	(25, 2,000,000)	(17: 5,425)	Col[15] < 0.8	5.4	2.4
Serial	(25, 2,000,000)	(17: 5,425)	Col[15] < 0.8	12.9	
Concurrente	(25, 2,000,000)	(6: 1,213,483)	Col[2] <= 0.3	5	2.3
Serial	(25, 2,000,000)	(6: 1,213,483)	Col[2] <= 0.3	11.3	
Concurrente	(25, 2,000,000)	(24: 551,445)	Col[22] > 1	4.2	2.7
Serial	(25, 2,000,000)	(24: 551,445)	Col[22] > 1	11.4	

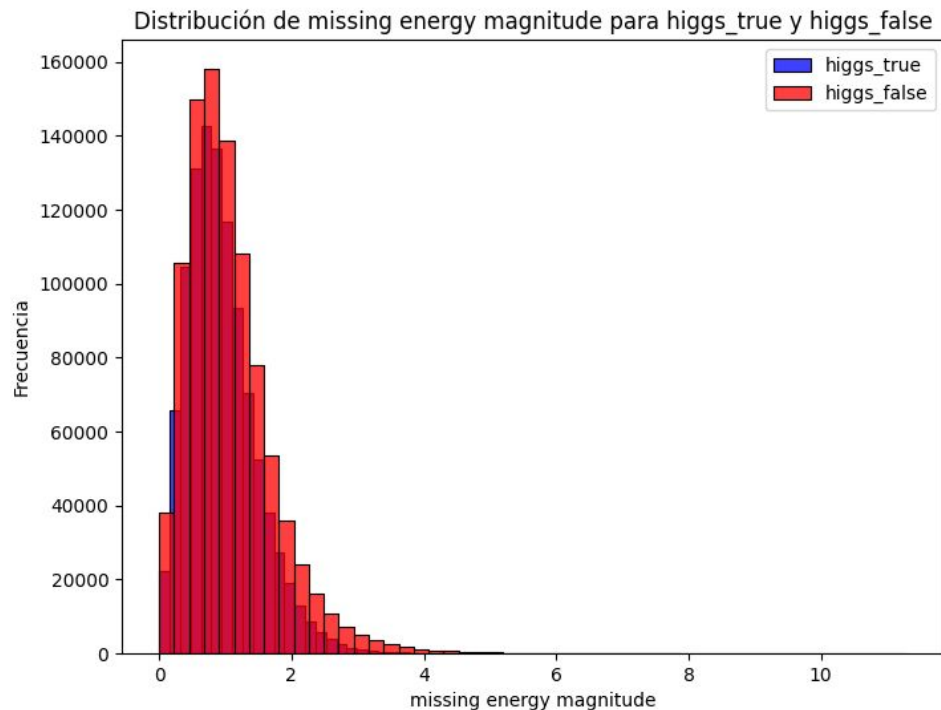
Análisis de los datos

- Utilizamos Python para el análisis.
- Mayor comodidad y herramientas efectivas (Pandas, Seaborn).
- Enfoque: Obtener distribuciones, promedios y comparar procesos con y sin bosón de Higgs.

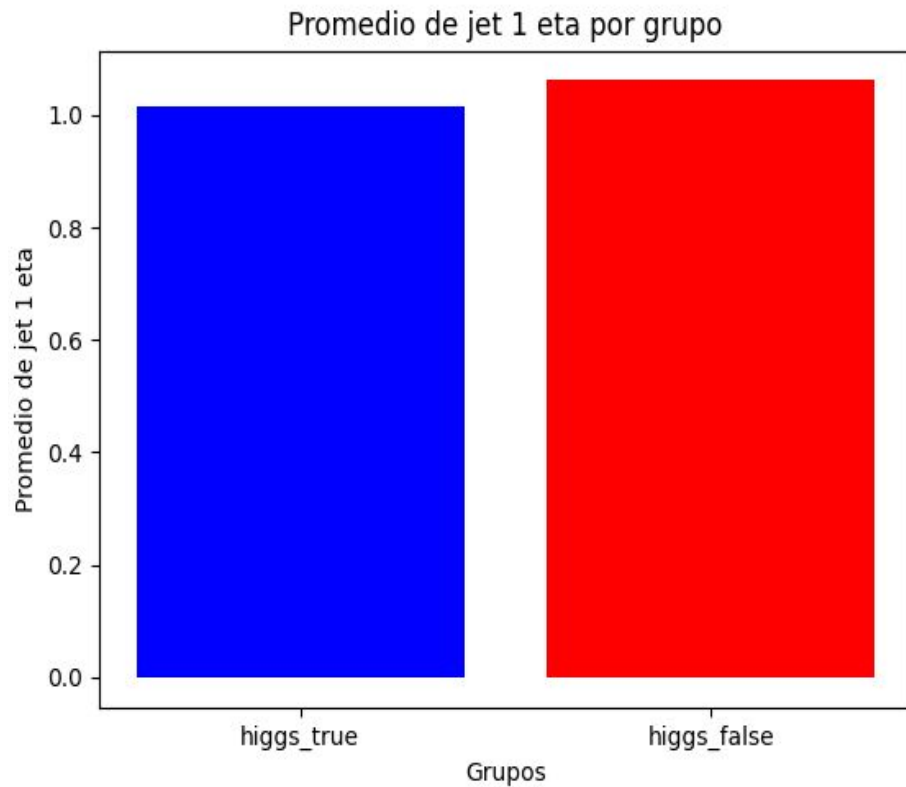
Promedio de la magnitud de la energía perdida



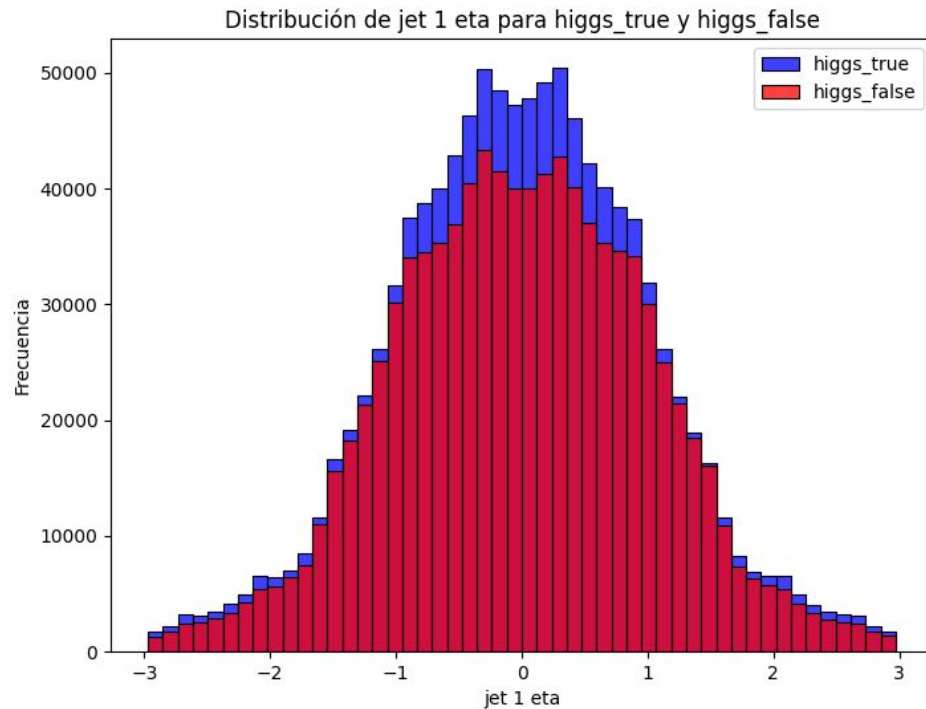
Distribución de la magnitud de la energía perdida



Promedio del momento del jet-1



Distribución del momento del jet-1



Secuencial/Concurrente	Tamaño original	Tamaño filtrado	Criterio de filtrado	Tiempo de ejecución (s)	Speedup
Concurrente	(25, 2,000,000)	(10: 324,342)	Col[1] <= 0.5	4.6	2.2
Serial	(25, 2,000,000)	(10: 324,342)	Col[1] <= 0.5	10.2	
Concurrente	(25, 2,000,000)	(20: 1,058,818)	Col[0] == 1	4.3	2.4
Serial	(25, 2,000,000)	(20: 1,058,818)	Col[0] == 1	10.5	
Concurrente	(25, 2,000,000)	(12: 18,941)	Col[10] <= 0.3	2.9	3.4
Serial	(25, 2,000,000)	(12: 18,941)	Col[10] <= 0.3	9.8	
Concurrente	(25, 2,000,000)	(6: 425,850)	Col[20] >=1	3.4	2.9
Serial	(25, 2,000,000)	(6: 425,850)	Col[20] >=1	10	
Concurrente	(25, 2,000,000)	(16: 924,453)	Col[11]>0.1	5.1	2.6
Serial	(25, 2,000,000)	(16: 924,453)	Col[11]>0.1	13.4	
Concurrente	(25, 2,000,000)	(21: 941,182)	Col[0] == 0	4.7	2.6
Serial	(25, 2,000,000)	(21: 941,182)	Col[0] == 0	12	
Concurrente	(25, 2,000,000)	(21: 2000,000)	Sin filtrado	5.8	2.6
Serial	(25, 2,000,000)	(21: 2000,000)	Sin filtrado	15.1	
Concurrente	(25, 2,000,000)	(17: 5,425)	Col[15] < 0.8	5.4	2.4
Serial	(25, 2,000,000)	(17: 5,425)	Col[15] < 0.8	12.9	
Concurrente	(25, 2,000,000)	(6: 1,213,483)	Col[2] <= 0.3	5	2.3
Serial	(25, 2,000,000)	(6: 1,213,483)	Col[2] <= 0.3	11.3	
Concurrente	(25, 2,000,000)	(24: 551,445)	Col[22] > 1	4.2	2.7
Serial	(25, 2,000,000)	(24: 551,445)	Col[22] > 1	11.4	