

## MANUAL TÉCNICO: DOCUMENTACIÓN Y EVALUACIÓN DEL SOFTWARE EDUCATIVO PROCESO2a2

Iván David Castañeda Balaguera<sup>1</sup>, Manuel Alejandro Segura Delgado<sup>2</sup>,  
Julián Andrés Salamanca Bernal<sup>3</sup>(jasalamanca@udistrital.edu.co)

<sup>1,2,3</sup>Universidad Distrital “Francisco José de Caldas”, Facultad de Ciencias y Educación  
Grupo de Física e Informática “FISINFOR”

Con el fin de generar un entorno asequible para el docente, en términos de dinamizar el proceso de aprendizaje de estudiantes, la creación de recursos educativos en tecnología para la enseñanza de la física es una apuesta en el propósito de establecer una alfabetización científico-tecnológica. El presente documento es un soporte del aporte de innovación en el desarrollo de software educativo, realizado desde el grupo de investigación en Física en Informática de la Universidad Distrital “Francisco José de Caldas” (FISINFOR), orientado a la enseñanza de la física de partículas, denominado **Proceso2a2**, el cual busca ser un instrumento para el estudio de las reacciones de partículas dos-a-dos ( $A+B \rightarrow C+D$ ), que modelan una cuantía de reacciones observadas en la naturaleza. Se siguen criterios de uso, manejo, desarrollo y evaluación (conforme a la norma ISO 9126 y el modelo de evaluación de software educativo del Instituto Latinoamericano de la Comunicación educativa en la Unidad de Investigación y Modelos educativos) para **Proceso2a2**, de acuerdo a los parámetros del Departamento Administrativo de Ciencia Tecnología e Innovación (Colciencias) para el registro de software según la resolución 00284-2004. De esta manera se busca establecer los indicadores de calidad del software.

*Palabras clave:* Reacciones de partículas 2 a 2, Energía umbral, Interfaz gráfica de usuario, alfabetización científica y tecnológica, Tecnología Rápida Orientada a Objetos (ROOT-C++)

Índice	Referencias	14
<b>INTRODUCCIÓN</b>	<b>3</b>	
<b>Validación de Proceso2a2</b>	<b>3</b>	
Reconocimiento y de los requisitos . . . . .	4	
Especificación y de los requisitos . . . . .	4	
<b>Proceso2a2: Diseño y desarrollo de software</b>	<b>4</b>	
Diseño de la interfaz gráfica de usuario . . . . .	4	
Desarrollo de software . . . . .	5	
<b>Implementación y prueba del software Proceso2a2</b>	<b>5</b>	
Sobre la simulación . . . . .	5	
Ícono . . . . .	11	
Gasto computacional . . . . .	11	
Declaración del idioma . . . . .	12	
Proceso de instalación . . . . .	12	
<b>Evaluación del software Proceso2a2</b>	<b>13</b>	
Norma ISO 9126 . . . . .	13	
Modelo del ILCE . . . . .	13	
Módulo del usuario . . . . .	13	
<b>CONCLUSIÓN</b>	<b>14</b>	

### Índice de figuras

1.	Matriz de trazabilidad del software <b>Proceso2a2</b>	3
2.	Ventana principal del software <b>Procesos2a2</b> . Se ilustra un inicio de la simulación con una energía del haz de fotones de 1.8 GeV, 100 eventos y un evento de energía. . . . .	6
3.	Lienzos para la visualización de los choques que generan una reacción de dos partículas, con sus respectivas etiquetas, en dos diferentes sistemas de referencia: laboratorio y centro de masas. . . . .	6
4.	Marco de agrupación, contiene los datos más relevantes de la simulación, incluyendo la opción de guardar archivos y nombrarlos. . .	6
5.	Lienzos para el dibujo de los histogramas. . .	6

6.	Botones de texto implementados en la interfaz. Cuando se deshabilita alguno, el texto se pone de color gris. “Limpiar”: ejecuta la función que limpia los lienzo y restablece los e inicializa todos los valores; “Simular”: ejecuta la función de la simulación; “Salir”: activa la función que cierra la ventana principal. . . . .	6	20.	Histograma de la distribución de la variable de Mandelstam $t$ en el laboratorio para la reacción $\Lambda_0 + K^+$ en el laboratorio con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía. . . . .	10
7.	Entradas numéricas, habilitadas (fondo blanco) para ingresar valores que son necesarios para los parámetros que requiere la simulación y deshabilitas (fondo gris) para solamente leer los datos arrojados por el programa. . . . .	7	21.	Histograma de la distribución de la variable de Mandelstam $t$ en el laboratorio para la reacción $\Lambda_0 + K^+$ en el centro de masa con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía. . . . .	10
8.	Selección de la reacción para variar las partículas que se producen después del choque. . . . .	7	22.	Histograma de la distribución de la variable de Mandelstam $u$ en el laboratorio para la reacción $\Lambda_0 + K^+$ en el laboratorio con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía. . . . .	10
9.	Botón de chequeo para habilitar o deshabilitar el dibujo de las flechas que representan los vectores del momento físico de las partículas. . . . .	7	23.	Histograma de la distribución de la variable de Mandelstam $u$ en el laboratorio para la reacción $\Lambda_0 + K^+$ en el centro de masa con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía. . . . .	10
10.	Botón de chequeo para guardar los histogramas y la entrada de texto para ingresar el nombre del archivo de salida. . . . .	7	24.	Distribución de la energía mínima para producir un $K^+$ con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía. . . . .	10
11.	Botón de ayuda, abre una ventana auxiliar que da instrucciones acerca del funcionamiento del software. . . . .	7	25.	Distribución de la energía mínima para producir un $\Lambda_0$ con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía. . . . .	10
12.	Lienzo que dibuja los allanadores de texto que muestran cómo funciona cada entrada numérica, de texto, y botones de la interfaz principal. . . . .	7	26.	Distribución del $\cos(\theta)$ del $\pi^0$ con una energía del haz de fotones de 1.8 GeV, 500 eventos y dos evento de energía en el centro de masa. . . . .	10
13.	Botón de texto que ejecuta la función para cerrar la interfaz auxiliar. . . . .	7	27.	Distribución del $\cos(\theta)$ del protón con una energía del haz de fotones de 1.8 GeV, 100 eventos y un evento de energía en el centro de masa. . . . .	10
14.	Diagrama de flujo del funcionamiento de la interfaz. . . . .	8	28.	Distribución de la energía mínima para producir un $\pi^0$ con una energía del haz de fotones de 1.8 GeV, 500 eventos y dos eventos de energía. . . . .	11
15.	Visualización con el dibujo de vectores habilitado. . . . .	8	29.	Distribución de la energía mínima para producir un protón con una energía del haz de fotones de 1.8 GeV, 500 eventos y dos eventos de energía. . . . .	11
16.	Visualización con el dibujo de vectores deshabilitado. . . . .	8	30.	Lista de directorios y código fuente para el funcionamiento del software <b>Proceso2a2</b> . . . . .	11
17.	Finalización de la simulación con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía. . . . .	9	31.	Ícono diseñado para el software “Proceso2a2”. . . . .	11
18.	Histograma de la distribución de la variable de Mandelstam $s$ en el laboratorio para la reacción $\Lambda_0 + K^+$ en el laboratorio con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía. . . . .	9	32.	Diagrama de barras de los tiempos de ejecución de los equipos para la reacción $\gamma + p \rightarrow p + \pi^0$ , como se muestra en la Tabla 2, con una energía de 2.3 GeV . . . . .	12
19.	Histograma de la distribución de la variable de Mandelstam $s$ en el laboratorio para la reacción $\Lambda_0 + K^+$ en el centro de masa con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía. . . . .	9			

33. Diagrama de barras de los tiempos de ejecución de los equipos para la reacción  $\gamma + p \rightarrow n + \pi^+$ , como se muestra en la Tabla 2, con una energía de 2.3 GeV . . . . . 12
34. Diagrama de barras de los tiempos de ejecución de los equipos para la reacción  $\gamma + p \rightarrow \Lambda_0 + K^+$ , como se muestra en la Tabla 2, para una energía de 2.3 GeV. . . . . 12

### Índice de Tablas

1. Especificación de los requisitos de software. . . . . 4
2. Tabla de tiempos de ejecución de software para diferentes reacciones, con una energía de haz de fotón de 2.3 GeV . . . . . 12

## INTRODUCCIÓN

En el contexto actual donde los avances científicos y tecnológicos se dan constantemente y tienen un influjo en la sociedad, en la cultura y por ende en la educación, los procesos de enseñanza-aprendizaje de las ciencias naturales y matemáticas se ven fuertemente afectados, tanto en su metodología de enseñanza-aprendizaje como en el uso y desarrollo de recursos educativos.

Al respecto, es de resaltar el papel que pueden desempeñar las tecnologías de la información y la comunicación (TIC) en el aula de clase, dado que co-adyudan, en muchos casos, a disminuir la curva de comprensión y presentación de contenidos curriculares, específicamente en ciencias naturales como la física. En este sentido se creó y desarrolló un recurso educativo computacional en la plataforma de análisis de datos ROOT, con una interfaz gráfica de usuario, denominado **Proceso2a2**, el cual busca ser un recurso educativo-computacional para el estudio de la cinemática relativista de un proceso dos-a-dos ( $A+B \rightarrow C+D$ ), y en general, un ejemplo de desarrollo de software para la alfabetización científica y tecnológica de profesores-investigadores en relación al desarrollo de aplicaciones para su quehacer.

### Validación de Proceso2a2

Alineados a la norma NTC/ISO IEC 17025, el desarrollo del software **Procesos2a2** se realizó haciendo uso de la “metodología cascada” como metodología de desarrollo, la cual se documenta a través de la matriz de trazabilidad ilustrada en la Figura 1.

Allí se definen siete elementos en la documentación de desarrollo a saber:

- El reconocimiento de requerimientos, el cual establece el objeto del software, donde se identifican y caracterizan los recursos computacionales necesarios para el correcto funcionamiento de **Procesos2a2**.

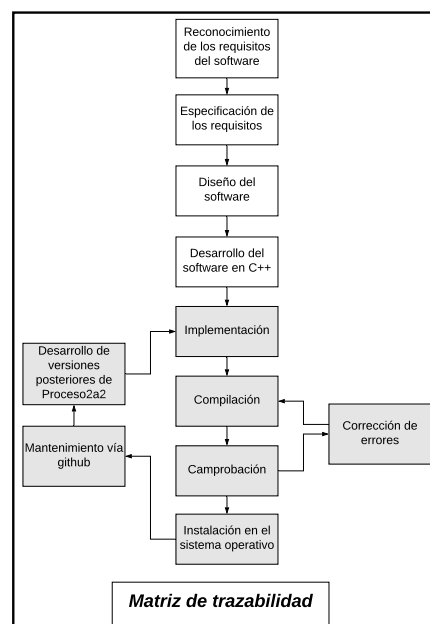


Figura 1. Matriz de trazabilidad del software **Proceso2a2**

- La especificación de los requerimientos, que ilustra el detalle de los requerimientos para el desarrollo, uso y manejo del software.
- El diseño de software, que elabora sobre la manera en la que se desarrolló el software.
- La etapa de implementación, que asegura el ejercicio de documentación en el proceso de compilación e instalación del software, la ejecución de pruebas, para auto-regular el ejercicio de corrección y recodificación de errores en sus funcionalidades.
- Fase de instalación y verificación, que detalla la instalación del software y verificación de su funcionamiento.
- El mantenimiento vía git, que permite el control y registro de versiones del software.
- Desarrollo de software de nuevas versiones, que como todo recurso de software, debe encontrarse en constante construcción, permitiendo así un retorno fresco a la etapa de implementación.

La evaluación de este software sigue la norma ISO 9126 y el modelo de evaluación de software educativo sigue los criterios del Instituto Latinoamericano de la Comunicación Educativa en la Unidad de Investigación y Modelos Educativos (Vanegas, 2016), obedeciendo los parámetros de Colciencias para el registro de software, conforme a la resolución 00285-2004.

## Reconocimiento y de los requisitos

Para el desarrollo de software fue necesario tener previamente instalada la plataforma de análisis de datos ROOT (*Rapid Oriented object technology*) en su versión 6.10/08 para equipos con una arquitectura de 64 bits. Con los paquetes y librerías de C++ versión 5.4.0 o posteriores que se usaron para crear las interfaces gráficas de usuario. La simulación cuenta con la declaración de diferentes objetos de diversas clases los cuales, que en su conjunto, generan una visualización y cálculos eficientes a través de constantes declaradas de alta precisión.

ROOT es una infraestructura de software utilizada por CERN (Centro Europeo de Investigaciones Nucleares) para el análisis de grandes cantidades de datos. Posee más de 3000 clases programadas en lenguaje C++ que permiten al usuario hacer visualización, diseño, control, simulación y análisis en tiempo real de experimentos en física de altas energías (Brun y Rademakers, 1997). Esta infraestructura tecnológica de software también se utiliza para el desarrollo de interfaces gráficas de usuario para el control remoto de experimentos y manipulación de software para usuario final.

## Especificación y de los requisitos

Dado que la exigencia de máquina pueden llegar a ser alta en la ejecución del software, se establece un requerimiento de sistema operativo Linux, con una distribución Ubuntu. La Tabla 1 consolida los requisitos de software para el desarrollo de software **Proceso2a2**.

Software	Versión (Arquitectura: 64 bits)
Ubuntu	16.04 LTS
ROOT	6.10.08
Build-essential	12.1
xutils-dev	1.77

Tabla 1

*Especificación de los requisitos de software.*

Con respecto al hardware, se requiere una máquina de procesador convencional para la fecha de esta documentación con más de 2G de memoria RAM.

## Proceso2a2: Diseño y desarrollo de software

### Diseño de la interfaz gráfica de usuario

La interfaz gráfica ha sido diseñada siguiendo los siguientes criterios:

#### 1. ¿Qué clase de interfaz es la más propicia para la simulación?

Una interfaz que permita visualizar los choques y la respectiva reacción en el centro de masa y en el laboratorio, con la posibilidad de dibujar los

vectores de los momentos lineales de cada partícula, ingresar los valores requeridos para la ejecución de la simulación; ver, en tiempo real, cómo varían los valores de las variables de Mandelstam y los histogramas. También que permita escoger guardar los histogramas si el usuario lo cree necesario, además de una ventana emergente que muestre indicaciones del uso de la interfaz y que se active mediante un botón de ayuda.

#### 2. ¿Cuántas ventanas se requieren para que el manejo de la interfaz sea intuitivo?

Se requieren dos ventanas: una principal en la que se visualice la simulación y se ingresen los parámetros y una ventana auxiliar, en la que se muestre el menú de ayuda para orientar el manejo del software desarrollado, logrando un manejo intuitivo del recurso educativo.

#### 3. ¿Qué cantidad de recursos utiliza la interfaz con la simulación incluida?

La interfaz está diseñada para gastar recursos computacionales propios de un computador convencional como ya se mencionó.

#### 4. ¿Cuántos botones debe tener la interfaz gráfica de usuario?

La interfaz debe contar con seis botones, uno botón para activar la simulación, otro botón para detener la simulación y limpiar los lienzos, y entradas numéricas y de texto, un botón para activar y desactivar la visualización de los vectores, otro botón que permita activar y desactivar la opción de guardado, el botón de ayuda que abra la ventana emergente para ver el menú de opciones de la interfaz y el botón que permita parar la simulación y cerrar la interfaz.

#### 5. ¿Qué parámetros se podrán establecer para la simulación, por medio de la interfaz?

Para la simulación se podrán establecer los mismos parámetros del programa base que son: energía del fotón, número de eventos, eventos de energía y nombre de los archivos de salida; además se podrá elegir la reacción, pues, a la que viene en el programa base se agregan dos más.

#### 6. ¿Por qué esta interfaz es la que mejor se adapta como recurso educativo?

Esta interfaz es la que mejor se adapta como recurso educativo porque su estructura facilita su manejo, además el complemento de ayuda sirve como soporte para el buen uso del software.

## Desarrollo de software

El software que se desarrolló cuenta con una interfaz gráfica de usuario intuitiva que permite la interacción entre el programa y el usuario, ofreciendo un entorno amigable y de uso eficiente para el estudio del fenómeno físico, en este caso, de la cinemática relativista de un proceso dos a dos, como se ilustra en la Figura 2.

Se evidencian en la ventana principal cuatro secciones:

La primera sección se ha dispuesto para la visualización de la reacción dos-a-dos, dos partículas que chocan, y el resultado de su interacción.

Dado que para el estudio es relevante un marco de referencia físico, se establecen dos sistemas de referencia: uno en el laboratorio y otro en el centro de masa de la reacción, con lo que esta sección posee dos lienzos de visualización como se ilustra en la Figura 3.

La segunda sección contiene todas las entradas numéricas y alfanuméricas para el funcionamiento de la simulación, requeridas por el estudio físico de la reacciones, y pensando en una visualización didáctica de todos los parámetros de la simulación de una reacción dos-a-dos (ver Figura 4)

La tercera sección, en la parte superior derecha de la ventana principal, contiene los lienzos y contenedores para el dibujo de los histogramas generados en el programa tanto para la descripción cinemática y energética de la reacción (Figura 5)

El grupo para ingresar los valores contiene:

- Las entradas numéricas, en cuales se ingresan los valores de entrada de energía inicial de la reacción para la simulación (ver Figura 7).
- Combo para la selección de la reacción a simular (ver Figura. 8).
- Botón de chequeo (`Checkbutton`), sirve para habilitar o deshabilitar opciones o funciones (ver Figura 9).
- Entradas de texto, la cuales sirven para insertar cadenas de caracteres, en este caso, para poner el nombre de los archivos de salida si se habilita la función de guardado, mediante un botón de chequeo, como se ilustra en la Figura 10.
- El botón de ayuda para abrir un cuadro de diálogo en un ventana auxiliar (ver Figura 11).

La interfaz auxiliar cuenta con un lienzo para el texto y un botón para cerrar la ventana.

- Lienzo para la visualización de los allanadores de texto (`Pavetext`) (ver Figura 12).
- Botón de para cerrar la interfaz auxiliar (ver Figura 13).

## Implementación y prueba del software Proceso2a2

El software simula una reacción dos-a-dos ( $A+B \rightarrow C+D$ ), en el cual se estudian diferentes variables físicas como la energía umbral para producir partículas, las variables de Mandelstam y los ángulos de dispersión (Segura, Salamanca, y Munevar, 2017).

La simulación del programa **Proceso2a2** presenta un menú de opciones para la recreación de los eventos a partir de la energía del haz de fotones, el número de eventos, el número de eventos de energía, la reacción y el nombre de los archivos de salida, los cuales son en formato `.root`, para la creación de los árboles y el archivo en formato `.dat`, que almacena los datos de salida.

### Sobre la simulación

Como ya se dialogó, para desarrollar la simulación en la plataforma de análisis de datos ROOT, se siguió la estructura de su programación del lenguaje base C++. Esta plataforma cuenta con una variedad de clases y funciones que permiten graficar, recrear, simular fenómenos físicos, consintiendo el desarrollo de interfaces gráficas de usuario (GUI) (Vargas y Salamanca, 2015). Para la simulación, las clases propias de ROOT que se han utilizado son:

- **TLorentzVector**: Permite un manejo dinámico de cuadri-vectores para definir los estados inicial y final de las partículas de la reacción.
- **TGenPhaseSpace**: Genera una distribución isotrópica partículas del estado final de la reacción.
- **TLorentzRotation**: Permite hacer rotaciones entre sistemas de referencia.
- **TRandom**: Generador de variables aleatorias.
- **TH1F**: Clase que permite manejar histogramas en una dimensión.
- **TCanvas**: Clase que permite manejar lienzos para visualización.
- **TRootEbeddedCanvas**: Clase que permite manejar lienzos embebidos.
- **TEllipse**: Clase para dibujo de elipses.
- **TArrow**: Clase para dibujo de vectores.
- **TFile**: Clase para el manejo de archivos.
- **TDirectory**: Clase para el manejo de directorios.

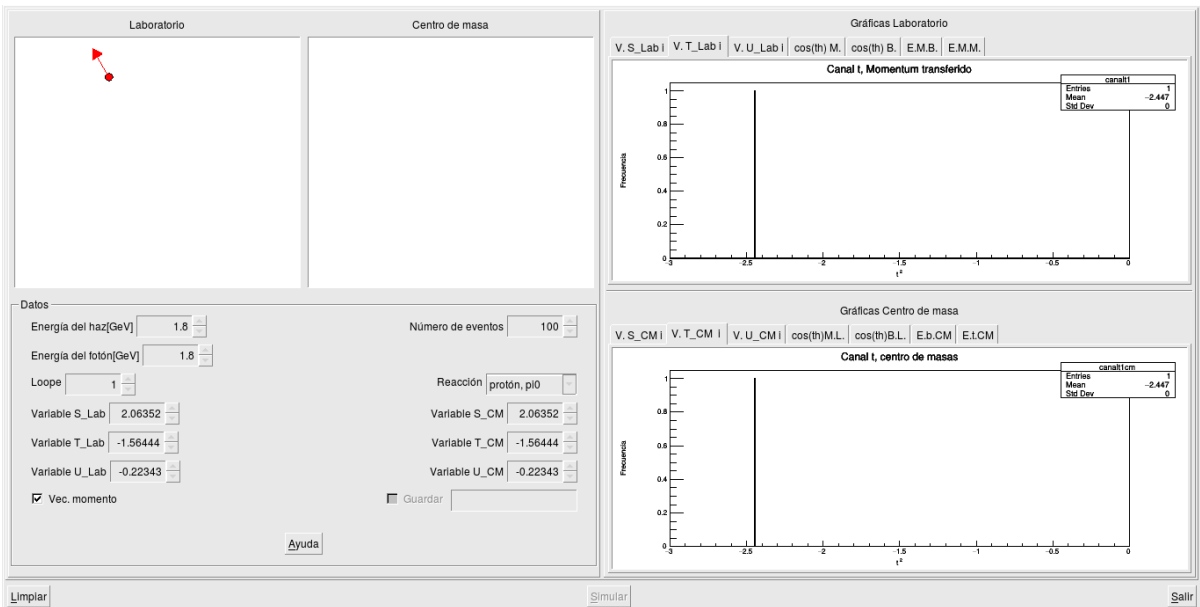


Figura 2. Ventana principal del software **Procesos2a2**. Se ilustra un inicio de la simulación con una energía del haz de fotones de 1.8 GeV, 100 eventos y un evento de energía.



Figura 3. Lienzos para la visualización de los choques que generan una reacción de dos partículas, con sus respectivas etiquetas, en dos diferentes sistemas de referencia: laboratorio y centro de masas.

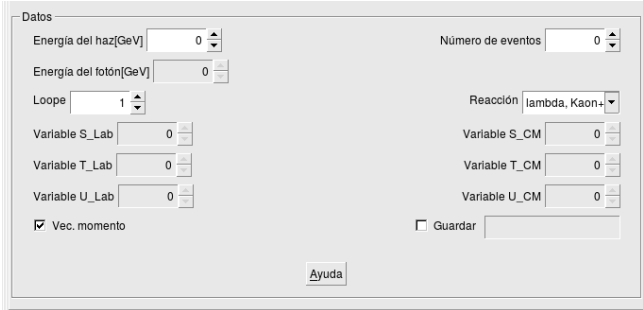


Figura 4. Marco de agrupación, contiene los datos más relevantes de la simulación, incluyendo la opción de guardar archivos y nombrarlos.



Figura 5. Lienzos para el dibujo de los histogramas.

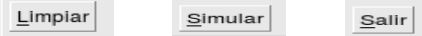


Figura 6. Botones de texto implementados en la interfaz. Cuando se deshabilita alguno, el texto se pone de color gris. “Limpiar”: ejecuta la función que limpia los lienzos y restablece los e inicializa todos los valores; “Simular”: ejecuta la función de la simulación; “Salir”: activa la función que cierra la ventana principal.

A text input field with the label 'Energía del haz[GeV]' and a numeric value '0'. The input field is active, indicated by a white background and a small cursor.

(a) Entrada numérica habilitada

 A text input field with the label 'Energía del fotón[GeV]' and a numeric value '0'. The input field is disabled, indicated by a gray background.

(b) Entrada numérica deshabilitada

Figura 7. Entradas numéricas, habilitadas (fondo blanco) para ingresar valores que son necesarios para los parámetros que requiere la simulación y deshabilita (fondo gris) para solamente leer los datos arrojados por el programa.

 A dropdown menu with the label 'Reacción' and the selected option 'lambda, Kaon+'.

Figura 8. Selección de la reacción para variar las partículas que se producen después del choque.

 A checkbox labeled 'Vec. momento' which is currently checked.

Figura 9. Botón de chequeo para habilitar o deshabilitar el dibujo de las flechas que representan los vectores del momento físico de las partículas.

 A checkbox labeled 'Guardar' which is currently unchecked, followed by an empty text input field.

(a) Inhabilitado cuando no corre la simulación

 A checkbox labeled 'Guardar' which is currently checked, followed by an empty text input field.

(b) Habilitado cuando no corre la simulación

 A checkbox labeled 'Guardar' which is currently checked, followed by a text input field containing the word 'Archivo'.

(c) Inhabilitado cuando corre la simulación

Figura 10. Botón de chequeo para guardar los histogramas y la entrada de texto para ingresar el nombre del archivo de salida.

El algoritmo de la simulación primeramente permite el ingreso de valores de entrada: energía del haz de fotones, número de eventos, eventos de energía por eventos generados, tipo de reacción, y si el usuario lo desea, guardar los archivos generados con un nombre de archivo de salida. Posteriormente, las partículas se representan mediante elipses (**TEllipse**), los vectores mediante la clase **TARrow** (que visualizan la dirección y magnitud del momento lineal de cada una de las partículas en su respectivo marco inercial de referencia), donde se pueden dibujar o no, si el usuario lo desea. Para una precisión estadística se puede introducir un número de eventos de energía mayor que 1 GeV.

 A button with the label 'Ayuda'.

Figura 11. Botón de ayuda, abre una ventana auxiliar que da instrucciones acerca del funcionamiento del software.

 A screenshot of a help window titled 'Ayuda'. It contains two columns of text explaining the various input fields and buttons of the software. The text is in Spanish and describes the purpose of each input field, such as 'Energía del haz', 'Energía del fotón', 'Número de eventos', 'Reacción', 'Vec. momento', and 'Guardar'.

Figura 12. Lienzo que dibuja los allanadores de texto que muestran cómo funciona cada entrada numérica, de texto, y botones de la interfaz principal.

 A button with the label 'Cerrar'.

Figura 13. Botón de texto que ejecuta la función para cerrar la interfaz auxiliar.

La Figura 14 ilustra el diagrama de flujo del software **Proceso2a2**.

Primero se crea la interfaz gráfica para poder dar la entrada numérica **Energía del haz** que asigna un valor de energía al haz de fotones, que corresponde al mismo valor de la energía del fotón (entrada numérica **Energía del fotón**), si el valor del número de eventos de energía **Loop** es uno; si es mayor, genera valores aleatorios de energía para los fotones mediante el método de Montecarlo, con un valor que oscila entre 0 GeV y el valor de energía que se ha ingresado para el haz; si el valor es cero, no genera eventos de energía y, por lo tanto, no hay choques y no se ejecuta la simulación.

Así mismo, se ingresa el **Número de eventos** en su respectiva entrada numérica, y se selecciona la **Reacción** en el menú de reacciones. Mientras no se ejecuta la simulación se encuentra habilitado el botón **Simular**, dando click a este botón, y habiendo ingresado los datos correspondientes, se ejecuta la simulación mostrando el primer choque en ambos marcos de referencia y se sigue al llenando los histogramas. En la interfaz es opcional guardar los histogramas, función que se activa o desactiva con el botón de chequeo **Guardar**. Aún si se habilita esa opción, pero no se ingresan caracteres a la entrada de texto, el programa no guarda.

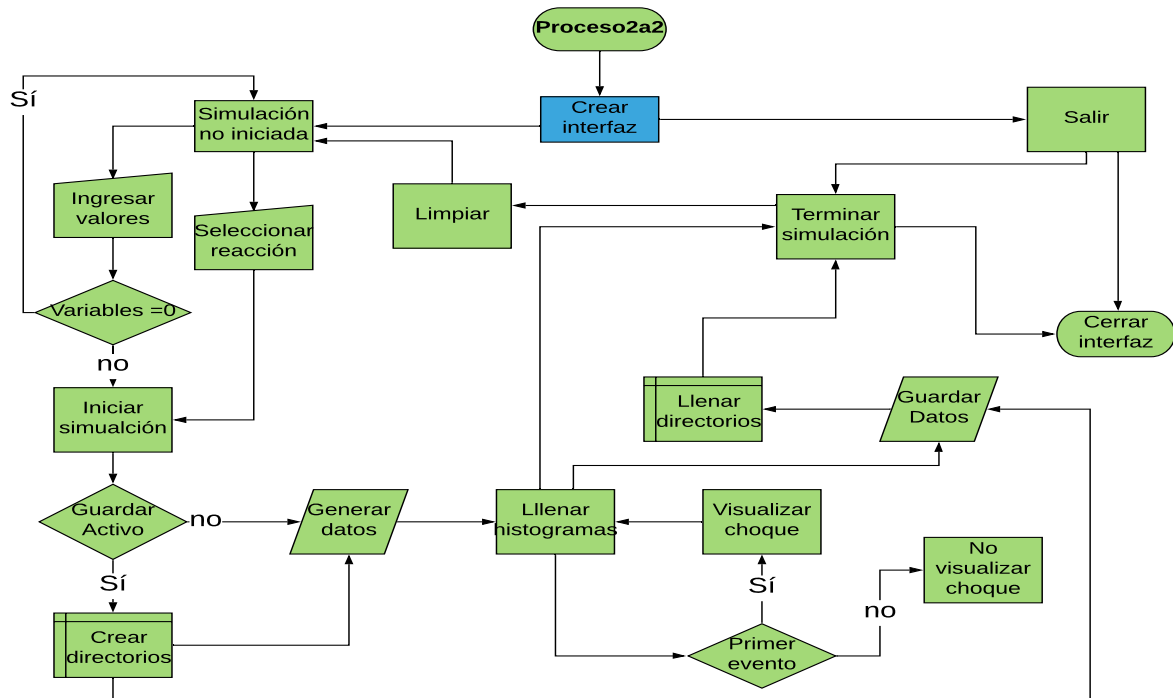
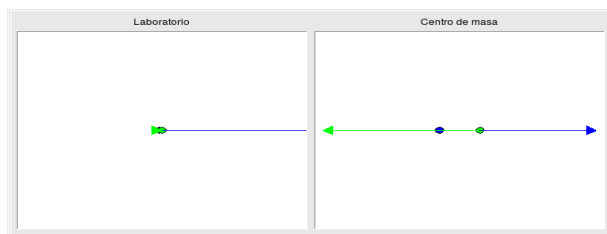
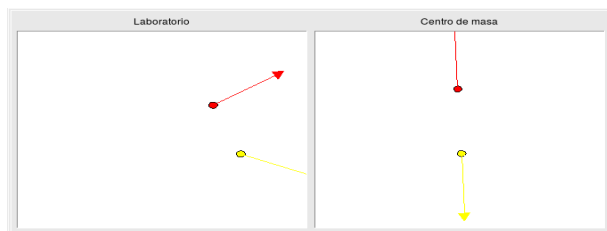


Figura 14. Diagrama de flujo del funcionamiento de la interfaz.



(a) Antes del choque, fotón azul y protón verde



(b) Después del choque, barión amarillo y mesón rojo

☒ Vec. momento

(c) Botón de chequeo para  
dibujar vectores: opción  
activada



(a) Antes del choque, fotón azul y protón verde



(b) Después del choque, barión amarillo y mesón rojo

☐ Vec. momento

(c) Botón de chequeo para  
dibujar vectores: opción  
desactivada

Figura 15. Visualización con el dibujo de vectores habilitado.

Figura 16. Visualización con el dibujo de vectores deshabilitado.



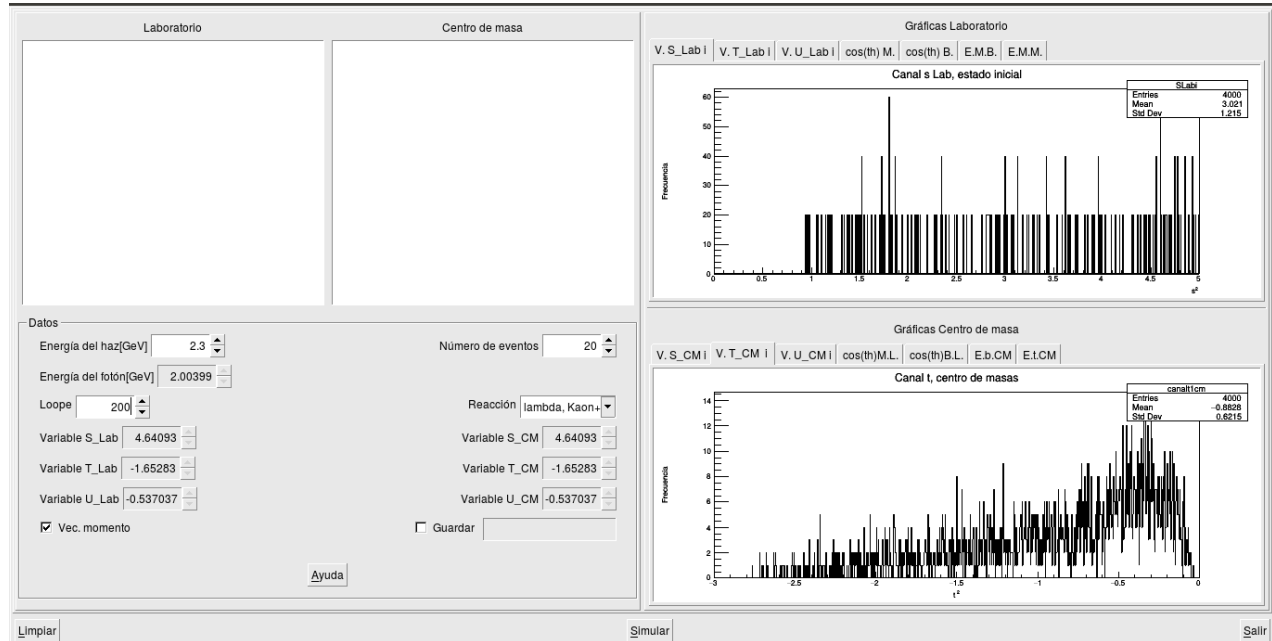


Figura 17. Finalización de la simulación con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía.

Mientras se visualiza el choque se puede activar o desactivar la opción para dibujar los vectores, con el botón de chequeo **Vec. momento**. Mientras la simulación está corriendo, se habilita el botón **Limpiar** y se deshabilitan las demás entradas y botones, al dar click se detiene la simulación y se restauran todas las entradas, quedando la interfaz en su estado inicial (ver Figuras 15 y 16).

En cualquier momento se encuentra habilitado el botón **Ayuda**, el cual abre la interfaz auxiliar, también siempre se encuentra habilitado el botón **Salir**, que, mediante un click, cierra la interfaz.

La interfaz auxiliar contiene el botón **Cerrar**, el cual emite la señal para ejecutar la función que cierra la ventana.

De esta manera, teniendo en cuenta el funcionamiento descrito, se puede hacer un buen uso de la interfaz; por ejemplo, ingresando un valor de energía para el haz de fotones de 2.3 GeV, con 20 eventos y 200 eventos de energía, se obtienen los resultados ilustrados en la Figuras 17 a 29.

Se han utilizado condicionales para guardar, dibujar vectores y dibujar los histogramas, dependiendo del valor del número de eventos, esto último, con el fin de optimizar el tiempo de simulación y recreación de eventos del software.

A mayor cantidad de eventos, mejor es la definición de las distribuciones; sin embargo, una cantidad de eventos grande implica tiempos largos en el proceso debido al consumo de memoria debido a que la interfaz, como entorno gráfico, sumado a la simulación, exige en buena medida al equipo.

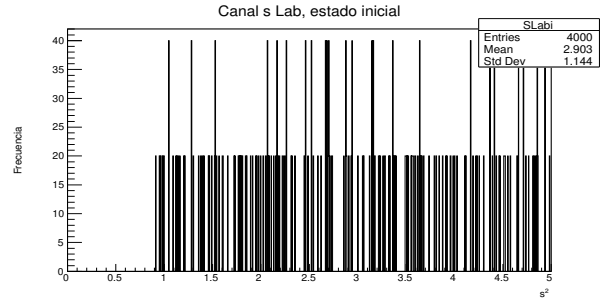


Figura 18. Histograma de la distribución de la variable de Mandelstam  $s$  en el laboratorio para la reacción  $\Lambda_0 + K^+$  en el laboratorio con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía.

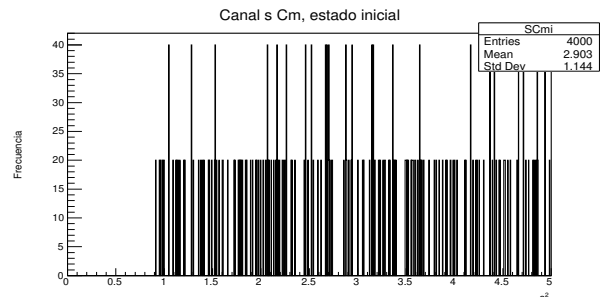


Figura 19. Histograma de la distribución de la variable de Mandelstam  $s$  en el laboratorio para la reacción  $\Lambda_0 + K^+$  en el centro de masa con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía.

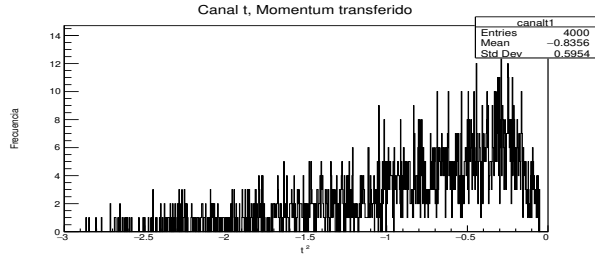


Figura 20. Histograma de la distribución de la variable de Mandelstam  $t$  en el laboratorio para la reacción  $\Lambda_0 + K^+$  en el laboratorio con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía.

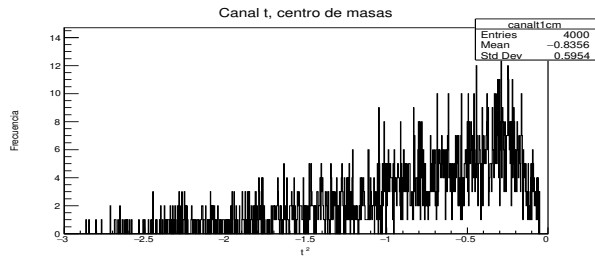


Figura 21. Histograma de la distribución de la variable de Mandelstam  $t$  en el laboratorio para la reacción  $\Lambda_0 + K^+$  en el centro de masa con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía.

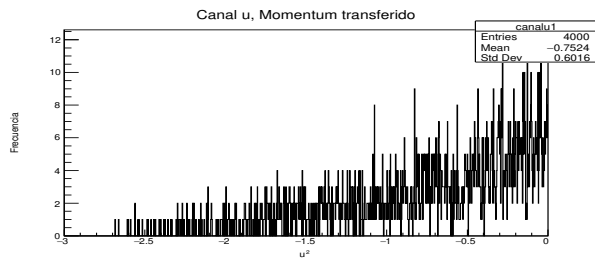


Figura 22. Histograma de la distribución de la variable de Mandelstam  $u$  en el laboratorio para la reacción  $\Lambda_0 + K^+$  en el laboratorio con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía.

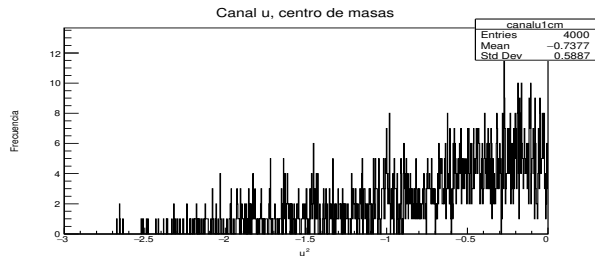


Figura 23. Histograma de la distribución de la variable de Mandelstam  $u$  en el laboratorio para la reacción  $\Lambda_0 + K^+$  en el centro de masa con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía.

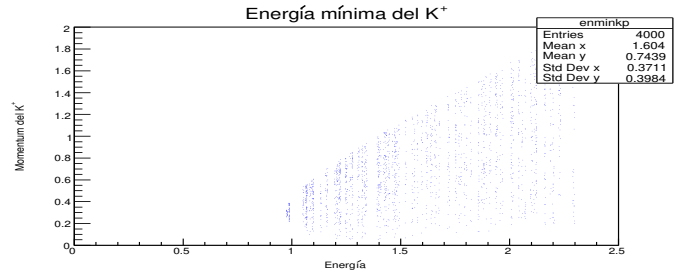


Figura 24. Distribución de la energía mínima para producir un  $K^+$  con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía.

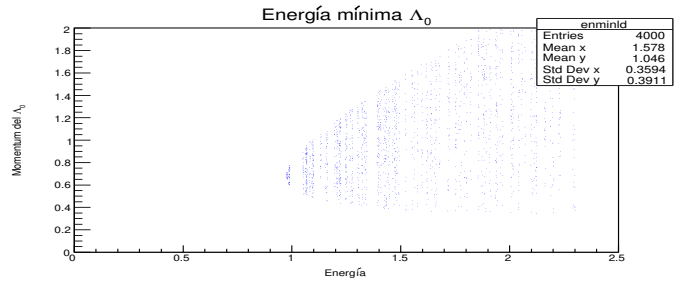


Figura 25. Distribución de la energía mínima para producir un  $\Lambda_0$  con una energía del haz de fotones de 2.3 GeV, 20 eventos y 200 eventos de energía.

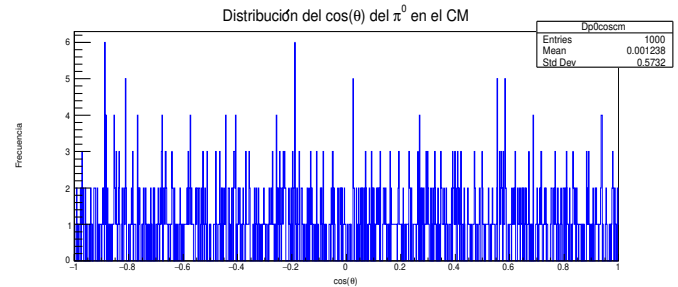


Figura 26. Distribución del  $\cos(\theta)$  del  $\pi^0$  con una energía del haz de fotones de 1.8 GeV, 500 eventos y dos evento de energía en el centro de masa.

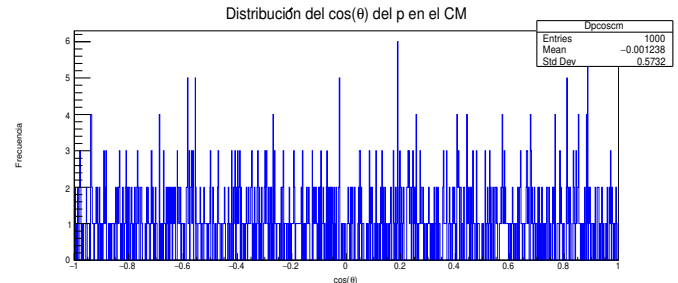


Figura 27. Distribución del  $\cos(\theta)$  del protón con una energía del haz de fotones de 1.8 GeV, 100 eventos y un evento de energía en el centro de masa.

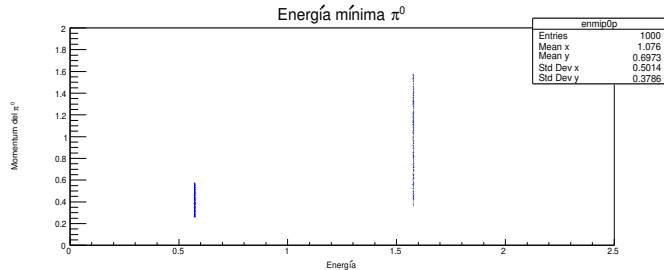


Figura 28. Distribución de la energía mínima para producir un  $\pi^0$  con una energía del haz de fotones de 1.8 GeV, 500 eventos y dos eventos de energía.

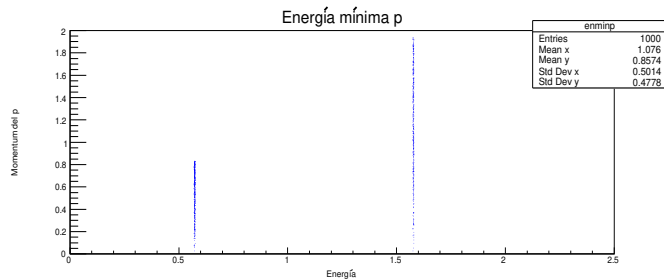


Figura 29. Distribución de la energía mínima para producir un protón con una energía del haz de fotones de 1.8 GeV, 500 eventos y dos eventos de energía.

De lo codificado anteriormente se obtiene un directorio que contiene los archivos necesarios para el funcionamiento de **Procesos2a2** como lo describe la Figura 30.

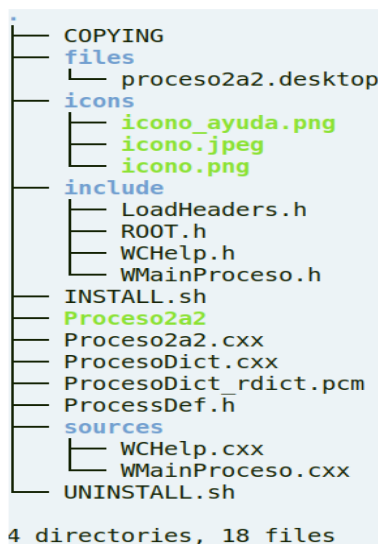


Figura 30. Lista de directorios y código fuente para el funcionamiento del software **Proceso2a2**.

Esta primera versión de **Proceso2a2** puede ser encontrada en la plataforma de github:

<https://github.com/fisinforgh/Proceso2a2.git>

## Ícono

El ícono que se ha diseñado para el software está basado en el diagrama de Feynmann del proceso y hace alusión al lenguaje de programación en el cual se hizo, que es C++, teniendo en cuenta que la esencia de ROOT es este lenguaje (Figura 31).

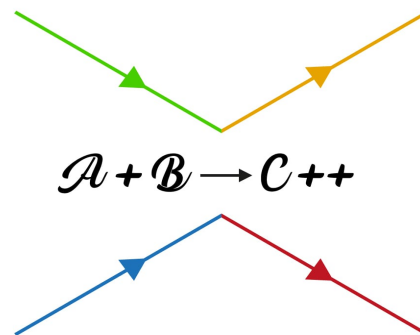


Figura 31. Ícono diseñado para el software “Proceso2a2”.

La leyenda  $A + B \rightarrow C ++$ , como se ha mencionado anteriormente hace alusión al lenguaje de programación en el cual se ha escrito el programa, así también se recuerda el choque entre las partículas, además se buscó que fuera un diseño sencillo y que fuera acorde al software elaborado.

## Gasto computacional

La implementación del software está hecha en los recursos computacionales que utiliza el software, además usa una capacidad de procesamiento y funcionamiento de la máquina, cuya prueba se ha hecho simulando con una valor de energía para el haz de fotones de 2.3 GeV, 100, 200, 500 y 1000 números de eventos y un evento de energía por cada valor de energía.

Como se puede ver en la Tabla 2, el software corre sin mayor dificultad en un equipo de características básicas, sin afectar de manera significativa sus funciones.

Las figuras 32, 33 y 34 comparan el desempeño de tres equipos para tres reacciones con procesadores de diferente gama, donde se obtiene mejor desempeño del software en el equipo que cuenta con un procesador de mayor capacidad; no obstante, no presenta mayores diferencias en desempeño con respecto a los otros dos equipos.

Gasto computacional								
S. Operativo	RAM (Gb)	Procesador	Reacción	No de eventos →	2.3 GeV			
					100	200	500	1000
Ubuntu 16.04 LTS 64bits	5.5	Intel core i5-3317U 1.70GHz		$\Lambda, K^+$	64.34s	20.97s	29.91s	67.58s
				$p, \pi^0$	35.76s	16.27s	32.25s	81.58s
				$n, \pi^+$	28.65s	12.19s	36.32s	75.00s
Ubuntu 18.04 LTS 64bits	3.8	Intel Celeron 1.60GHz		$\Lambda, K^+$	50.28s	25.78s	28.95s	50.30s
				$p, \pi^0$	26.84s	19.87s	27.19s	49.26s
				$n, \pi^+$	27.90s	12.87s	28.94s	52.75s
Ubuntu 16.04 LTS 64bits	2.7	Intel Pentium(R) CPU P6000 1.87GHz × 2		$\Lambda, K^+$	61.76s	30.56s	59.47s	111.75s
				$p, \pi^0$	55.19s	34.26s	58.96s	107.36s
				$n, \pi^+$	66.34s	30.89s	60.31s	115.23s

Tabla 2

Tabla de tiempos de ejecución de software para diferentes reacciones, con una energía de haz de fotón de 2.3 GeV

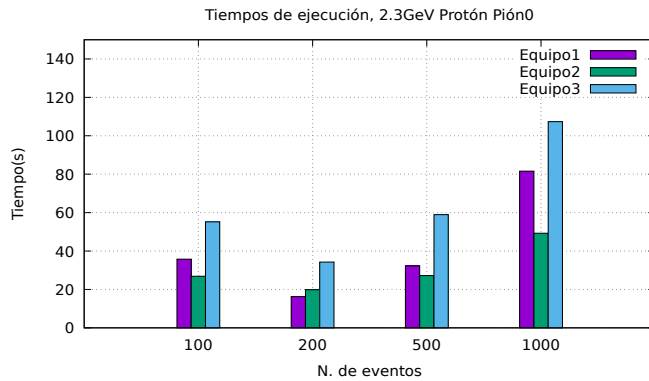


Figura 32. Diagrama de barras de los tiempos de ejecución de los equipos para la reacción  $\gamma + p \rightarrow p + \pi^0$ , como se muestra en la Tabla 2, con una energía de 2.3 GeV

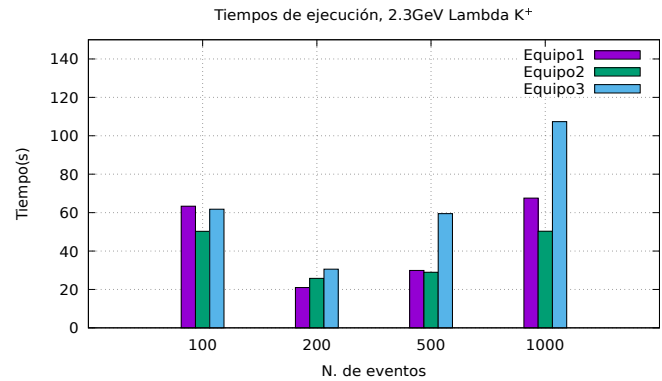


Figura 34. Diagrama de barras de los tiempos de ejecución de los equipos para la reacción  $\gamma + p \rightarrow \Lambda + K^+$ , como se muestra en la Tabla 2, para una energía de 2.3 GeV.

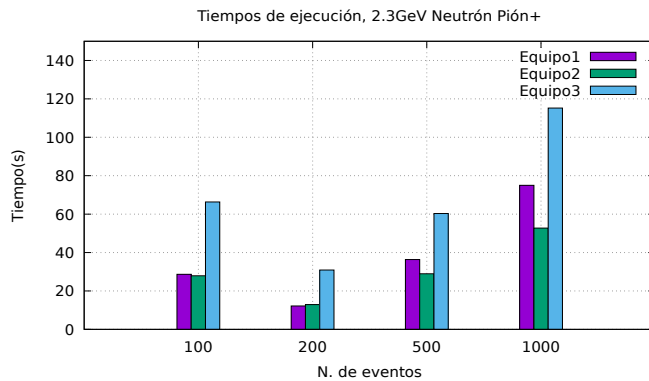


Figura 33. Diagrama de barras de los tiempos de ejecución de los equipos para la reacción  $\gamma + p \rightarrow n + \pi^+$ , como se muestra en la Tabla 2, con una energía de 2.3 GeV

## Declaración del idioma

El idioma en el cual se desarrolló el software es el español, debido a que es un producto hecho en Latinoamérica que pretende acercar la producción académica de este tipo a la población de habla hispana.

## Proceso de instalación

**Licencia:** El código fuente del software **Proceso2a2** está liberado bajo los términos de la licencia GPLv3<sup>1</sup> que permite realizar modificaciones, crear copias y distribuirlas (esta no tiene restricción comercial). Además está permitido, o no, cobrar por la distribución. Esta primera versión de **Proceso2a2** puede ser encontrada en la plataforma de github: <https://github.com/fisinforgh/Proceso2a2.git>

Con github, una excelente herramienta de desarrollo colaborativo de software, se puede continuar el proceso de desarrollo de versiones posteriores.

Adicionalmente, de acuerdo a la norma NTC/ISO IEC 17025 y haciendo uso de la metodología para un modelo de cascada orientado al desarrollo de software, como se esquematiza en la matriz de trazabilidad (Figura 1), se consolida un proceso de instalación software, que en primera instancia, exige la correcta instalación de los requisitos de

<sup>1</sup>Para mayor información consultar: <https://www.gnu.org/licenses/gpl-3.0.html>.

software citados en la Tabla 1.

Este software se instala mediante un proceso básico, ejecutando pocas instrucciones en la terminal de Linux. Ubicándose en la carpeta de archivos del software, se le otorgan permisos a los archivos `INSTALL.sh` y `UNINSTALL.sh` mediante el comando:

```
chmod +x INSTALL.sh UNINSTALL.sh.
```

Posteriormente, se ejecuta el archivo `INSTALL.sh` mediante la instrucción:

```
./INSTALL.sh
```

donde se visualiza la respuesta de ejecución:

```
Instalando...
```

```
¡Instalacion completada exitosamente!
```

```
Para correr Proceso2a2 desde la terminal ejecute:
Proceso2a2
```

Para el proceso desinstalación, se ejecuta el archivo `UNINSTALL.sh` mediante la instrucción

```
./INSTALL.sh
```

obteniéndose en la terminal:

```
Eliminando archivos...
```

```
¡Ha sido exitoso el proceso de desinstalacion!
```

**Troubleshooting:** En caso de que la desinstalación no sea satisfactoria, se debe leer el archivo `UNINSTALL.sh` y realizar por terminal cada una de sus líneas de comando. En total se corresponde a un ejercicio de realizar 5 líneas de comando.

### Evaluación del software Proceso2a2

Los recursos digitales deben ser evaluados, bajo indicadores de valoración que den cuenta de la calidad sistémica y se adecúen a las necesidades siendo una estrategia de verificación conceptual y procedimental antes de ser llevados al aula, es decir, que tengan una pertinencia en los aspectos técnicos y tecnológicos de la herramienta y los pedagógicos (Vanegas, 2016).

Con relación a este recurso educativo computacional se siguieron los criterios que establece la norma ISO 9126 para la medición de la calidad en la industria del software, añadiendo la evaluación de software educativo que plantea el Instituto Latinoamericano para la Comunicación Educativa en la Unidad de Investigación y Modelos Educativos.

### Norma ISO 9126

La norma ISO 9126 (también conocida como ISO/IEC 9126) fue publicada en 1992 y es usada para la evaluación de la calidad del software (Largo y Marin, 2005). Establece siete cualidades para tener en cuenta que son:

- *Funcionabilidad*
- *Confiabilidad*
- *Usabilidad*
- *Eficiencia*
- *Mantenibilidad*
- *Portabilidad*
- *Calidad de uso*

Estos indicadores propician un seguimiento correcto del desarrollo de los productos computacionales. El ILCE acoge las categorías estipuladas y las clasifica en módulos para estructurar la evaluación de software.

### Modelo del ILCE

Este modelo en su aplicación propone cuatro módulos:

- Módulo del diseñador
- Módulo del usuario
- Módulo del administrador
- Módulo del evaluador

Esta evaluación considera los criterios del módulo del usuario, en vista de que será él quien hará uso del software. La calificación cualitativa se propone cuantitativamente con una métrica de 0 a 10 para cada uno de los criterios. Entre 0 y 2 la calificación es muy baja, entre 2 y 4 la calificación es baja, entre 4 y 6 es media, entre 6 y 8 es alta y entre 8 y 10 es muy alta. Finalmente, el promedio ponderado arroja la calificación final a la evaluación del software.

### Módulo del usuario

**Caracterización o perfil del usuario:** En este módulo se busca medir la efectividad y pertinencia del software, contempla al profesor-usuario del software como destinatario y receptor inicial. En primer lugar, el docente debe tener la capacidad de interpretar el manual técnico, teniendo en cuenta los conocimientos básicos y habilidades en computación, propios del contexto actual, además de una formación en ciencias básicas y/o ingeniería. Los conocimientos previos que debe tener corresponden a conocer y manejar otros software de tipo científico-didáctico, también en el área de la física de partículas y la cinemática

relativista, por tal motivo debe tener la habilidad y destreza en la resolución de problemas numéricos, sumado al buen criterio para la selección de software, de acuerdo con sus necesidades de enseñanza. Algunos criterios que se deben considerar son con el fin de contextualizar el software, de acuerdo con los requisitos de las máquinas que posea.

Los criterios que se han de tener en cuenta para evaluar el software son:

- **Calidad:** Este criterio evalúa la presentación y estructura del software, uso y recursos técnicos que ofrece, la interfaz con el usuario y las potencialidades generales que lo constituyen como una herramienta educativa (Morales, González, Carmona, y Espíritu, 1998). El software cuenta con una interfaz gráfica de usuario cuya estructura y presentación permiten al usuario interactuar con el programa mediante el ingreso manual de números, opciones y caracteres. Además de la manipulación de sus gráficos, el software da una experiencia interactiva entre el usuario y la máquina, la cual es la que procesa la información. **En este ítem se le asigna un valor de 10.**
- **Eficacia:** Este criterio evalúa el potencial que tiene el software para cumplir sus objetivos educativos (Morales et al., 1998). Teniendo en cuenta que para el uso del software hay que tener un dominio conceptual respecto de la física de partículas y la cinemática relativista, la eficacia no será 100 % garantizada. No obstante, es tarea del profesor asumir el reto de “propio aprendizaje” para lograr una base conceptual sobre el tema del software. Por lo tanto, a este criterio se le asigna una **evaluación de 8**
- **Eficiencia:** Este criterio evalúa qué tanto puede el software ayudar a ahorrar esfuerzos a los actores implicados en el proceso de enseñanza y aprendizaje (Morales et al., 1998). En este sentido, el software, como recurso educativo, es eficiente porque ayuda a reducir los esfuerzos para explicar la cinemática relativista, el hecho de poder visualizar los choques vuelve mayormente dinámico al programa. En este criterio, **la calificación es de 9**
- **Pertinencia:** Este criterio evalúa al software como recurso educativo que apoya procesos en el aula (Morales et al., 1998). El software **Proceso2a2** está especialmente orientado a docentes, para que usen este recurso como herramienta tecnológica que busca lograr el fin de comprender los conceptos relacionados

con un proceso dos-a-dos, lo cual permite adecuar los contenidos y, de esta manera, llegar a generar interés por parte del alumno. Por lo tanto la **calificación para este criterio es 9**

- **Impacto:** Este criterio evalúa el efecto que causa el software sobre el usuario, especialmente el del aprendizaje. En este sentido, el software puede generar un impacto de largo alcance (Morales et al., 1998) en sus entornos más próximos, debido a que sus recursos gráficos son un atractivo y un buen ejemplo de lo que pasa en este fenómeno. Adicionalmente, y por la liberación del código fuente bajo la Licencia GPLv3, asiente un ejercicio de desarrollo de software que puede llevar al profesor a desarrollar sus propias aplicaciones, con el objeto de hacer alfabetización científica y tecnológica orientado a profesores-investigadores. **La calificación para este criterio es 9**

## CONCLUSIÓN

Basado en los criterios de evaluación usados y sus respectivas calificaciones, se obtuvo un promedio de **9.0/10**, lo cual ubica al software en la calificación de **MUY ALTO**; esto evidencia que el software se corresponde a un recurso educativo computacional que puede facilitar la labor docente y allanar dudas acerca del comportamiento de la naturaleza en experimentos como el que se simula, reacciones de partículas como ejemplos de la interacción radiación-materia. Así, su posible impacto en la población colombiana, a través del docente y el uso continuado del software por los estudiantes, aunará en la formación científica y tecnológica de profesores-investigadores y en la inmersión científica de estudiantes de ciencias e ingeniería.

## Referencias

- Brun, R., y Rademakers, F. (1997). ROOT an object oriented data analysis framework. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 389(1-2), 81–86.
- Largo, C., y Marin, E. (2005). *Guía técnica para evaluación de software*. Punto exe. Descargado de [https://jrvargas.files.wordpress.com/2009/03/guia\\_tecnica\\_para\\_evaluacion\\_de\\_software.pdf](https://jrvargas.files.wordpress.com/2009/03/guia_tecnica_para_evaluacion_de_software.pdf)
- Morales, C., González, Y., Carmona, V., y Espíritu, S. (1998). *Evaluación de software educativo*. ILCE. Descargado de [http://investigacion.ilce.edu.mx/panel\\_control/doc/c36,evaluacsoft.pdf](http://investigacion.ilce.edu.mx/panel_control/doc/c36,evaluacsoft.pdf)
- Segura, M. A., Salamanca, J., y Munevar, E. (2017). Monte carlo threshold energy estimation for a+b c+d

- processes: An educational resource in experimental high energy physics research. *Revista Brasileira de Ensino de Física*, 39(1). doi: <http://dx.doi.org/10.1590/1806-9126-RBEF-2016-0114>
- Vanegas, G. (2016). *Modelo de evaluación para recursos digitales en la enseñanza de la física en el ced cafam bellavista*. Universidad Distrital Francisco José de Caldas. Descargado de <http://repository.udistrital.edu.co/handle/11349/4238>
- Vargas, A. M., y Salamanca, J. A. (2015). *Rootlets, interfaces graficas en root*. Editorial académica española.