

Flujos Normalizantes Condicionales para la estimación del espectro de muones.

Cristian Vladimir Orduz Carvajal

Jhon Jairo Almanzar Quintero

Trabajo de Grado para Optar el Título de Ingeniero de Sistemas

Director

Luis A. Núñez de Villavicencio Martínez

Doctor en Física

Codirector

Christian Andres Sarmiento Cano

Doctor en Física

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2026

Dedicatoria

Dedico este trabajo de tesis a mi madre y a mis abuelos, por su apoyo emocional, su confianza y por creer siempre en mí a lo largo de este camino. A mis hermanas, Camila e Isabella, por sus consejos y su ayuda cuando más lo necesitaba. A mi perrito Pascal y a mis gatos, que no hubo un solo día en el cual no me sacaran una sonrisa.

Cristian Orduz Carvajal

Este proyecto está dedicado a mi madre, Claudia Quintero, por su amor, apoyo y compañía durante toda esta etapa de mi vida. A mis hermanos y familia, por estar presentes a lo largo del camino, y a mi pareja, por acompañarme en un proceso lleno de retos y dificultades. Sin su ayuda, no habría podido pensar en muones y computadoras todo el día.

Jhon Jairo Almanzar Quintero

Agradecimientos

Queremos expresar nuestro agradecimiento a nuestros directores, Crhistian Sarmiento y Luiz Núñez, por su tiempo, orientación y conocimiento invertido en nosotros durante el desarrollo de este proyecto. Además, agradecemos a los miembros del grupo LIDeRA por abrirnos las puertas y permitirnos conocer el mundo de la investigación.

Finalmente, expresamos un agradecimiento especial a nuestro tutor, Rafael Martínez, quien nos guió con paciencia y esfuerzo durante todo este proceso. ¡El mejor tutor!

Tabla de contenido

Introducción	14
1 Objetivos	17
1.1 Objetivo general	17
1.2 Objetivos específicos	17
2 Marco de referencia	18
2.1 Rayos cósmicos y muografía	18
2.1.1 Estimación del flujo de muones	19
2.2 Flujos normalizantes condicionales: fundamentos y aplicaciones	21
2.2.1 Modelos de inteligencia artificial.	21
2.2.2 Modelos discriminativos y generativos	22
2.2.3 Flujos Normalizantes y su extensión condicionada.	24
2.2.4 Implementación de CNFs en dominios análogos.	28
2.3 Generalidades del entorno de desarrollo web.	30
2.3.1 Arquitectura de software	30
2.3.1.1 Arquitectura cliente–servidor	30
2.3.1.2 Servicios <i>web</i> y API REST	31
2.3.2 Tecnologías y criterios de desarrollo web	31
2.3.2.1 Angular	32
2.3.2.2 FastAPI	32
2.3.2.3 Diseño web responsive	33

3 Metodología	33
3.1 Generación de un conjunto de datos balanceado a partir de ARTI/CORSIKA	34
3.1.1 Generación y exploración preliminar de datos	35
3.1.1.1 Selección de los sitios de estudio	35
3.1.1.2 Preparación de la simulación	36
3.1.1.3 Tiempos de una simulación	37
3.1.1.4 Filtrado de muones	38
3.1.1.5 Ajuste del flujo de muones vs altura.	39
3.2 Estimación y evaluación del espectro de muones para el CNF	41
3.2.1 Subconjunto preliminar de datos	41
3.2.2 Procesamiento de los datos para el CNF	42
3.2.2.1 Muestreo estratificado de eventos por ciudad	42
3.2.2.2 Transformación y normalización de variables	45
3.2.3 Modelado y evaluación del CNF	47
3.2.3.1 Arquitectura del modelado	47
3.2.3.2 Procedimiento de evaluación del modelo	49
3.2.3.3 Búsqueda aleatoria de hiperparámetros	51
3.2.3.4 Validación de robustez	52
3.2.3.5 Modelado final	53
3.3 Prototipo de la plataforma de consulta, generación y visualización	55
3.3.1 Diseño e implementación de la plataforma	55
3.3.1.1 Levantamiento funcional y alcance	55
3.3.1.2 Planeación técnica de la solución	55
3.3.1.3 Diseño de interfaz y especificación de interacción	56
3.3.1.4 Implementación de la arquitectura web	56
3.3.1.5 Integración funcional y verificación	57

3.3.2 Pruebas funcionales	59
3.3.3 Estrategia de despliegue web e integración	60
3.3.3.1 Configuración y publicación del cliente	60
3.3.3.2 Contenerización y ejecución del servidor.	60
4 Resultados	61
4.1 Conjunto de datos final	61
4.2 Estimación del espectro mediante CNF	62
4.2.1 Búsqueda de hiperparámetros	62
4.2.2 Validación de robustez	64
4.2.3 Modelado final	65
4.3 Costo computacional de la estimación	72
4.4 Prototipo web	75
4.4.1 Resultados del despliegue y validación en entorno real	82
5 Conclusión	83
Referencias bibliográficas	86
Apéndices	92

Lista de tablas

Tabla 1.	Rangos de B_i y h definidos para el muestreo estratificado.	35
Tabla 2.	Ejemplos de tiempos de ejecución para distintas simulaciones.	38
Tabla 3.	Ciudades incluidas en entrenamiento y validación para el subconjunto preliminar	42
Tabla 4.	Hiperparámetros de mayor incidencia en el ajuste del modelo	48
Tabla 5.	Hiperparámetros de menor incidencia en el ajuste del modelo	49
Tabla 6.	Configuración seleccionada del modelo a partir de la búsqueda aleatoria	51
Tabla 7.	Particiones empleadas en la validación de robustez	52
Tabla 8.	Conjunto final: ciudades incluidas en validación y prueba	53
Tabla 9.	Iteraciones de desarrollo del prototipo de la plataforma.	56
Tabla 10.	Casos de prueba funcional manual de la plataforma <i>web</i>	59
Tabla 11.	Cinco configuraciones con menor MARE global obtenidas en la búsqueda aleatoria	63
Tabla 12.	MARE de energía y ángulo por ciudad de validación para la configuración 2 . .	64
Tabla 13.	Resultados de las tres corridas de validación de robustez	65
Tabla 14.	Resultados globales del modelado final sobre los conjuntos de validación y prueba	66
Tabla 15.	MARE por intervalos de energía y ángulo en los conjuntos de validación y prueba	66
Tabla 16.	Precisión en el número de eventos generados por el CNF	68
Tabla 17.	Comparación agregada entre simulación de ARTI y generación en la plataforma <i>web</i>	73
Tabla 18.	Desempeño temporal del CNF por ciudad en la plataforma <i>web</i>	75

Tabla 19.	Servicios integrados en el flujo principal de la plataforma piloto.	76
Tabla 20.	Componentes visibles del prototipo y función dentro del flujo de uso.	77
Tabla C.1.	Ubicaciones simuladas del conjunto de datos final	99
Tabla D.1.	Eventos generados por ciudad en los conjuntos de validación y prueba	101

Lista de figuras

Figura 1.	Diagrama del flujo metodológico abordado.	34
Figura 2.	Salida generada por ARTI	38
Figura 3.	Ajuste mediante máxima verosimilitud de Poisson de la relación entre el flujo integrado de muones y la altura del sitio simulado.	40
Figura 4.	Espectros de energía transformada y ángulo cenital para Bucaramanga.	43
Figura 5.	Distribución de las variables objetivo transformadas para Bucaramanga.	46
Figura 6.	Ejemplo de espectros generados por el CNF para Jungfrauoch.	54
Figura 7.	Diagrama de casos de uso de la plataforma piloto.	57
Figura 8.	Diagrama de secuencia del flujo principal de la plataforma.	58
Figura 9.	Muestreo final con todas las ciudades simuladas.	62
Figura 10.	Reconstrucción del espectro energético en Zard-Kuh, Santos Cuchumatán y Shiraz	69
Figura 11.	Reconstrucción del espectro angular en Zard-Kuh, Santos Cuchumatán y Shiraz	69
Figura 12.	Evolución de la NLL en los conjuntos de entrenamiento y validación	71
Figura 13.	Relación entre el tiempo de la etapa runner y el número de partículas registradas.	72
Figura 14.	Tiempo de inferencia del CNF en función del número de eventos generados. . .	74
Figura 15.	Vista de inicio de la plataforma.	77
Figura 16.	Vista del simulador de la plataforma.	78
Figura 17.	Vista de descripción de la plataforma piloto.	78
Figura 18.	Adaptación responsive de la interfaz.	79

Figura 19.	Secuencia funcional del flujo principal de la plataforma.	80
Figura 20.	Resultados gráficos generados por la plataforma.	81
Figura 21.	Archivos generados por la plataforma.	82
Figura E.1.	Comparación del espectro energético proporcionado por ARTI y el generado por el CNF en ciudades de validación.	102
Figura E.2.	Comparación del espectro angular proporcionado por ARTI y el generado por el CNF en ciudades de validación.	103
Figura F.1.	Comparación del espectro energético proporcionado por ARTI y el generado por el CNF en ciudades de prueba.	104
Figura F.2.	Comparación del espectro angular proporcionado por ARTI y el generado por el CNF en ciudades de prueba.	105

Lista de apéndices

A	Ejemplo de estructura JSON para un sitio de simulación	93
B	Script runner.sh para la ejecución automatizada de simulaciones	94
C	Ubicaciones simuladas del conjunto de datos final	99
D	Precisión y tiempo en la generación de eventos por ciudad	101
E	Resultados para las ciudades del conjunto de validación	102
F	Resultados para las ciudades del conjunto de prueba	104

Resumen

Título: Flujos Normalizantes Condicionales para la estimación del espectro de muones.*

Autor: Cristian Vladimir Orduz Carvajal, Jhon Jairo Almanzar Quintero**

Palabras Clave: *Flujos normalizantes condicionales, Prototipo de plataforma web, Muografía, ARTI/CORSIKA, Muones, Costo computacional*

Descripción:

La muografía es una técnica de imagen no invasiva que infiere la estructura interna de grandes objetos mediante la atenuación del flujo de muones atmosféricos. Su viabilidad depende de estimar con precisión el espectro muónico del sitio de interés, el cual varía de forma no lineal con la altitud y el campo geomagnético. Los métodos tradicionales presentan limitaciones opuestas: simulaciones como ARTI/CORSIKA son precisas pero lentas, superando las once horas por sitio, mientras que las parametrizaciones analíticas son rápidas pero restringidas a condiciones geofísicas específicas. Esta brecha motivó la pregunta central: ¿pueden los flujos normalizantes condicionales replicar la precisión de ARTI con un costo computacional significativamente menor?

Para responderla, se generó un conjunto de datos balanceado mediante muestreo estratificado sobre 45 ubicaciones en 25 países, cubriendo altitudes entre 0 y 5230 m s.n.m. e intensidades geomagnéticas entre 22,5 y 56,75 μT . Con base en este conjunto, se entrenó un modelo bidimensional con transformaciones autoregresivas enmascaradas y splines racionales cuadráticas, capaz de estimar la distribución conjunta de energía y ángulo cenital condicionada por variables geofísicas. Adicionalmente, el modelo fue integrado en una plataforma web Angular–FastAPI para consultar, generar y visualizar espectros según su ubicación geográfica.

El modelo alcanzó un error relativo absoluto promedio de 0,1160 en validación y 0,0925 en prueba. Las mayores discrepancias se concentraron en el intervalo energético de 10^3 a 10^4 GeV y en los extremos angulares, regiones de baja densidad donde el propio ARTI presenta variabilidad estadística inherente a su naturaleza Monte Carlo. En eficiencia, el tiempo de generación se redujo de 8 horas 50 minutos por ciudad a un tiempo promedio de 60 segundos por ciudad, con aceleraciones superiores a $500\times$. Estos resultados respaldan la viabilidad de los flujos normalizantes condicionales como modelo sustituto para la estimación del espectro de muones en aplicaciones de muografía.

*Trabajo de Grado

**Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Luis A. Núñez de Villavicencio Martínez. Doctor en Física. Codirector: Christian Andres Sarmiento Cano. Doctor en Física.

Abstract

Title: Conditional Normalizing Flows for muon spectrum estimation.*

Author: Cristian Vladimir Orduz Carvajal, Jhon Jairo Almanzar Quintero**

Key Words: *Conditional normalizing flows, Web platform prototype, Muography, ARTI/CORSIKA, Muons, Computational cost*

Description:

Muography is a non-invasive imaging technique that infers the internal structure of large objects from the attenuation of atmospheric muon flux. Its feasibility depends on accurately estimating the muon spectrum at the site of interest, which varies nonlinearly with altitude and the geomagnetic field. Traditional methods have opposing limitations: simulations like ARTI/CORSIKA are accurate but take a long time, exceeding eleven hours per site, while analytical parameterizations are fast but restricted to specific geophysical conditions. This gap motivated the central question: can conditional normalizing flows replicate the accuracy of ARTI with a significantly lower computational cost?

To address this, a balanced dataset was generated through stratified sampling over 45 locations in 25 countries, covering altitudes between 0 and 5230 m a.s.l. and geomagnetic intensities between 22.5 and 56.75 μT . Based on this dataset, a bidimensional model was trained using masked autoregressive transformations and rational quadratic splines, capable of estimating the joint distribution of energy and zenith angle conditioned on geophysical variables. Additionally, the model was integrated into an Angular-FastAPI web platform to query, generate, and visualize spectra from a geographic location.

The model achieved a mean absolute relative error of 0.1160 on the validation set and 0.0925 on the test set. The largest discrepancies were concentrated in the energy range of 10^3 to 10^4 GeV and at angular extremes, low-density regions where ARTI itself shows statistical variability due to its Monte Carlo nature. In terms of efficiency, generation time was reduced from 8 hours 50 minutes per city to an average time of 60 seconds per city, with speedups exceeding $500\times$. The results support the viability of conditional normalizing flows as a surrogate model for estimating the muon spectrum in muography applications.

*Degree Work

**Faculty of Physicomechanical Engineering. School of Systems Engineering and Informatics. Director: Luis A. Núñez de Villavicencio Martínez. Doctor of Physics. Codirector: Christian Andres Sarmiento Cano. Doctor of Physics.

Introducción

La muografía es una técnica de imagenología no invasiva que aprovecha la atenuación del flujo de muones atmosféricos para inferir la distribución interna de densidad en estructuras de gran tamaño Bonechi et al., 2020; Kaiser, 2019. Puede entenderse como una radiografía de gran escala hecha con partículas que llegan de forma natural a la superficie terrestre. Uno de sus antecedentes más conocidos fue la búsqueda de cámaras ocultas en la pirámide de Kefrén mediante mediciones de muones Alvarez et al., 1970. Desde entonces, su uso se ha extendido a volcanes, donde se ha empleado para reconstruir estructuras internas como conductos y núcleos densos en domos de lava Oláh et al., 2018; Vesga-Ramírez et al., 2021; a espacios subterráneos, para estudiar cavidades y variaciones internas de densidad Saracino et al., 2019; y a instalaciones industriales, para detectar contrastes de densidad en reactores de hidrotratamiento Martínez-Rivero et al., 2025.

Para interpretar una imagen muográfica se necesita una referencia del flujo incidente esperado en ausencia de la estructura de interés Bonechi et al., 2020; Saracino et al., 2019. Ese flujo no permanece fijo entre sitios, pues depende de variables como la altitud y las condiciones geomagnéticas locales Hebbeker y Timmermans, 2002; Kaiser, 2019. Por esta razón, estimar el espectro de muones forma parte del problema operativo de la muografía: antes de comparar atenuaciones, se debe contar con una descripción confiable de cuántos muones llegan al sitio y cómo se distribuyen según su energía y ángulo de llegada.

En la literatura, este problema se ha abordado por dos vías. La primera agrupa los esquemas de simulación, como ARTI Sarmiento-Cano et al., 2022, que ofrecen alta fidelidad al modelar las interacciones de los rayos cósmicos en la atmósfera, pero implican un costo computacional elevado Baack, 2019. La segunda agrupa las parametrizaciones analíticas, como CMSCGEN Biallass y

Hebbeker, 2009, MUPAGE Carminati et al., 2008 y las versiones modificadas de la fórmula de Gaisser Guan et al., 2015, que permiten cálculos más rápidos pero operan dentro de dominios de validez restringidos.

Frente a esta brecha, los modelos de inteligencia artificial se han consolidado como herramientas útiles para aprender regularidades a partir de datos y construir modelos sustitutos de procesos numéricos costosos Jordan y Mitchell, 2015; Mehta et al., 2019. Dentro de los modelos generativos, los flujos normalizantes resultan especialmente pertinentes cuando el objetivo es representar distribuciones de forma explícita Bond-Taylor et al., 2022; Papamakarios et al., 2021: definen densidades probabilísticas expresivas a partir de una distribución base simple y una sucesión de transformaciones biyectivas, y permiten tanto evaluar la densidad como generar muestras nuevas Lu y Huang, 2020; Papamakarios et al., 2021. En su versión condicionada, un mismo modelo puede representar cómo cambia la distribución objetivo al variar las condiciones de entrada Lu y Huang, 2020; Winkler et al., 2019, formulación que ya ha sido aplicada en física para inferir distribuciones posteriores por evento, como en la reconstrucción de eventos de IceCube IceCube Collaboration, 2024.

A partir de estas consideraciones se plantea la pregunta de investigación: ¿es posible modelar el espectro de muones atmosféricos mediante flujos normalizantes condicionales, alcanzando un ajuste equivalente al de las simulaciones ARTI con un costo computacional inferior? Con esta pregunta como eje, el trabajo se organiza en tres frentes: la generación de un conjunto de datos balanceado a partir de simulaciones con ARTI, el modelado del espectro mediante flujos normalizantes condicionales y el desarrollo de una plataforma web para la consulta, generación y visualización de resultados.

El documento se estructura de la siguiente manera. El Capítulo 1 presenta los objetivos general y específicos del trabajo. El Capítulo 2 desarrolla el marco de referencia, abarcando los fundamentos de los rayos cósmicos y la muografía, los principios teóricos de los flujos normalizantes condicionales y las generalidades del entorno de desarrollo web empleado. El Capítulo 3 describe la metodología seguida, desde la generación del conjunto de datos hasta el modelado del

CNF y el desarrollo de la plataforma. El Capítulo 4 expone los resultados obtenidos en cada una de estas etapas. Finalmente, el Capítulo 5 presenta las conclusiones del trabajo y señala posibles líneas de trabajo futuro.

1. Objetivos

1.1. Objetivo general

- Desarrollar modelos predictivos basados en inteligencia artificial generativa para la estimación del espectro de muones integrados en una plataforma para visualizar y centralizar la consulta de datos.

1.2. Objetivos específicos

1. Generar un conjunto de datos a partir de ARTI/CORSIKA sujeto a un diseño de muestreo balanceado, como base de análisis y modelado.
2. Estimar el espectro de muones mediante flujos normalizantes condicionales, utilizando variables como la latitud, longitud, altitud y campo geomagnético.
3. Evaluar la precisión y el costo computacional de la estimación del modelo, en contraste con las simulaciones de referencia.
4. Desarrollar una plataforma piloto con servicios de consulta, generación y visualización; orientada a la centralización y exploración inicial de resultados.

2. Marco de referencia

2.1. Rayos cósmicos y muografía

Los rayos cósmicos son partículas de origen extraterrestre que se propagan a través del medio interestelar e intergaláctico. Poseen un rango de energía aproximado de entre 10^5 eV y 10^{20} eV, y están constituidos mayoritariamente por núcleos atómicos con número atómico entre $Z = 1$ y $Z = 26$, es decir, desde hidrógeno hasta hierro. Cuando un rayo cósmico penetra la atmósfera e interactúa con los núcleos de las moléculas del aire, se producen partículas como electrones, positrones, muones, fotones y hadrones.

Las partículas siguen interactuando durante su propagación atmosférica, generando nuevas partículas que repiten el proceso. De esta forma se desarrolla una lluvia atmosférica extensa, o EAS (*Extensive Air Shower*, por sus siglas en inglés), definida como la cascada de partículas producida a partir del rayo cósmico incidente. Dentro de las EAS se distinguen tres componentes principales: la hadrónica, la electromagnética y la muónica.

En el presente trabajo resulta de interés la componente muónica de las EAS, debido a su uso en aplicaciones de muografía. Los muones atmosféricos son partículas cargadas. Al nivel del mar presentan una tasa de llegada cercana a 100 Hz/m^2 Bonechi et al., 2020; Cecchini y Spurio, 2012. Debido a su capacidad de penetrar grandes espesores de materia, los muones pueden emplearse como sondas naturales para estudiar estructuras de gran tamaño. Su flujo depende de factores como el ángulo cenital, la altitud y los efectos geomagnéticos Hebbeker y Timmermans, 2002.

La muografía es una técnica de imagen no invasiva que aprovecha la atenuación del flujo de muones para inferir información sobre la distribución interna de densidad en objetos grandes

Kaiser, 2019; Saracino et al., 2019. Su principio consiste en medir el flujo de muones que atraviesa una estructura y compararlo con el flujo esperado en ausencia de esta Bonechi et al., 2020; Saracino et al., 2019. Cuando los muones atraviesan materia, pueden perder energía o desviarse de su trayectoria; por ello, la reducción del flujo registrado aporta información sobre la cantidad de materia atravesada y, en consecuencia, sobre la estructura interna del objeto Kaiser, 2019.

El desarrollo de la muografía se ha evidenciado en distintas aplicaciones. En arqueología uno de sus antecedentes más conocidos fue la búsqueda de cavidades ocultas en la pirámide de Kefrén mediante mediciones de muones Alvarez et al., 1970. En geociencias se ha empleado para estudiar volcanes y estimar contrastes internos de densidad en estructuras no accesibles mediante observación directa Oláh et al., 2018; Vesga-Ramírez et al., 2021. En ingeniería civil su uso se ha orientado a inspeccionar estructuras masivas, cavidades y espacios subterráneos Saracino et al., 2019.

En Colombia también se contempla el desarrollo de la muografía. En la Universidad de los Andes, se han realizado estudios orientados al análisis de estructuras geológicas del país, como el trabajo de Fajardo sobre muografía de estructuras geológicas en Colombia Fajardo, 2018. En la Universidad Industrial de Santander, el proyecto MuTe ha desarrollado un telescopio híbrido para estudiar volcanes colombianos, con énfasis en el volcán Cerro Machín de León-Barrios et al., 2021; Vesga-Ramírez et al., 2021. Se ha explorado la aplicación industrial de la muografía para el análisis de densidad en reactores de hidrotatamiento Martínez-Rivero et al., 2025. También se desarrolló ARTI, que permite estimar el flujo de partículas secundarias asociado a un sitio específico, incorporando condiciones locales relevantes para estudios muográficos Sarmiento-Cano et al., 2022.

2.1.1. Estimación del flujo de muones

La muografía requiere estimar el flujo de muones incidente en la ubicación de interés. Este flujo permite establecer una referencia contra la cual se compara la señal registrada después de

atravesar el objeto estudiado. Dicha estimación debe describir cuántos muones llegan a un sitio durante un intervalo de tiempo y cómo se distribuyen según variables como la energía y el ángulo de llegada.

La estimación del flujo de muones se aborda principalmente mediante dos enfoques, simulaciones por computadora y parametrizaciones analíticas. Las simulaciones modelan el desarrollo de la EAS desde el rayo cósmico incidente hasta las partículas producidas en las EAS que alcanzan la superficie. Las parametrizaciones, en cambio, usan expresiones cerradas para aproximar el flujo muónico bajo condiciones específicas Guan et al., 2015; Wang et al., 2024.

En el caso de las simulaciones, CORSIKA es uno de los programas más utilizados para modelar EAS. Fue desarrollado para el experimento KASCADE y ha sido ampliamente validado en estudios de astropartículas Antoni et al., 2003; Heck et al., 2012. A partir de los parámetros de entrada asociados al rayo cósmico, CORSIKA recrea la interacción con la atmósfera y genera como salida un registro de las partículas secundarias que alcanzan distintas alturas Heck et al., 2012.

También se encuentra ARTI, un *framework* desarrollado por la colaboración LAGO (*Latin American Giant Observatory*, por sus siglas en inglés). ARTI integra CORSIKA para estimar el flujo en un sitio específico e incorpora condiciones locales como el campo geomagnético y la atmósfera Sarmiento-Cano et al., 2022. Este tipo de herramientas permite obtener estimaciones detalladas del flujo muónico, aunque con un costo computacional elevado cuando se requiere explorar múltiples ubicaciones o condiciones físicas.

Por otro lado, las parametrizaciones analíticas consideran marcos simplificados dentro de los cuales una fórmula cerrada puede reproducir resultados de simulaciones o datos experimentales Biallass y Hebbeker, 2009; Carminati et al., 2008; Guan et al., 2015. Entre estas se encuentran CMSCGEN (*Compact Muon Solenoid Cosmic(-ray) GENerator*, por sus siglas en inglés), que ajusta resultados de CORSIKA mediante polinomios en función del momento y del ángulo cenital para parametrizar el flujo a nivel del suelo Biallass y Hebbeker, 2009; MUPAGE (*MUon GENerator from PArametric formulas*, por sus siglas en inglés), que emplea fórmulas paramétricas obtenidas

a partir de simulaciones Monte Carlo completas para generar haces de muones atmosféricos en telescopios de neutrinos submarinos o en hielo Carminati et al., 2008; y las versiones modificadas de la fórmula de Gaisser, orientadas a parametrizar el flujo de muones cósmicos a nivel del mar y mejorar su concordancia con resultados experimentales Guan et al., 2015.

En general, ambos enfoques presentan ventajas y limitaciones. Las simulaciones ofrecen mayor precisión y un dominio de aplicación más amplio, pero son computacionalmente costosas, ya que requieren modelar toda la EAS y su tiempo de ejecución puede variar desde horas hasta días, según las condiciones del cálculo Baack, 2019; LAGO Collaboration, 2024; Oláh et al., 2018; Sarmiento-Cano et al., 2022. Por su parte, las parametrizaciones estiman directamente el flujo muónico y resultan más rápidas al basarse en expresiones analíticas cerradas; sin embargo, presentan mayores restricciones en las condiciones bajo las cuales el cálculo es válido Biallass y Hebbeker, 2009; Carminati et al., 2008; Guan et al., 2015.

2.2. Flujos normalizantes condicionales: fundamentos y aplicaciones

2.2.1. Modelos de inteligencia artificial

La inteligencia artificial (IA) es un campo interdisciplinario de la informática orientado al diseño de sistemas capaces de ejecutar tareas asociadas con capacidades cognitivas humanas, como el reconocimiento de patrones, la toma de decisiones y el aprendizaje a partir de datos. Históricamente, el área evolucionó desde enfoques simbólicos basados en reglas explícitas hacia métodos guiados por datos, impulsados por el aumento de la capacidad computacional y la disponibilidad de grandes volúmenes de información Nilsson, 2011; Turing, 2009. Esta transición sentó las bases para el desarrollo de modelos parametrizados orientados a aprender patrones observados empíricamente.

El aprendizaje automático se consolidó como una de las primeras líneas de desarrollo de la IA, al proponer modelos capaces de capturar regularidades del dominio sin codificar reglas

determinísticas de forma explícita Mehta et al., 2019. Dentro de este marco, el aprendizaje profundo constituye una subárea basada en redes neuronales de múltiples capas, capaces de aprender representaciones jerárquicas de los datos, lo que facilita el modelado de relaciones no lineales en espacios de alta dimensionalidad LeCun et al., 2015. Estas características resultan pertinentes cuando el objetivo no se limita a predecir valores puntuales, sino a aproximar comportamientos estadísticos y distribuciones asociadas a variables de interés.

De manera general, un problema de aprendizaje parte de un conjunto de datos $D = \{(x_i, y_i)\}_{i=1}^n$, supuesto como una muestra finita de una distribución subyacente $p(x, y)$. Sobre este conjunto se define un modelo parametrizado por ϕ , que representa el conjunto de parámetros del modelo. Su desempeño se cuantifica mediante una función de pérdida $\ell(\cdot)$, que mide la discrepancia entre las salidas del modelo y las observaciones. El ajuste de ϕ se formula como la minimización de una función de costo global $\mathcal{L}_D(\phi)$, resuelta mediante procedimientos de optimización numérica Mehta et al., 2019. Esta formulación permite extender el análisis hacia modelos probabilísticos, en los que el interés se centra en aproximar distribuciones condicionadas en lugar de relaciones deterministas.

En aplicaciones científicas, los modelos de IA se emplean para identificar patrones entre variables observadas en mediciones o simulaciones, así como para construir modelos sustitutos que reproduzcan, dentro de un dominio definido, la salida de simulaciones numéricas con menor costo de computación Mehta et al., 2019. En este marco, resulta pertinente distinguir entre enfoques discriminativos y generativos según la forma en que representan la información y formulan la inferencia.

2.2.2. Modelos discriminativos y generativos

Los modelos de aprendizaje pueden clasificarse según la función objetivo que optimizan y el tipo de representación probabilística que construyen. En este contexto, suele establecerse una diferencia entre los modelos discriminativos y los generativos. Ng y Jordan, 2001.

Los modelos discriminativos buscan aproximar directamente la probabilidad condicional $p(y | x)$, donde x representa una observación de entrada perteneciente al espacio $\mathcal{X}$, y y la salida o etiqueta asociada perteneciente al espacio $\mathcal{Y}$. Este enfoque puede formularse como el aprendizaje de una función parametrizada $f_\phi : \mathcal{X} \rightarrow \mathcal{Y}$, que permite predecir y a partir de x . Dado un conjunto de datos $D = \{(x_i, y_i)\}_{i=1}^n$, el ajuste del modelo se formula mediante

$$\mathcal{L}_D(\phi) \triangleq \frac{1}{n} \sum_{i=1}^n \ell(f_\phi(x_i), y_i), \quad (2.1)$$

y el estimador se obtiene como

$$\phi^* = \arg \min_{\phi} \mathcal{L}_D(\phi), \quad (2.2)$$

donde $\arg \min$ denota el operador de ϕ para el cual $\mathcal{L}_D(\phi)$ alcanza su mínimo.

A diferencia de los modelos discriminativos, los modelos generativos se orientan al aprendizaje de una distribución de probabilidad que describa la estructura estadística de los datos, ya sea la conjunta $p(x, y)$ o, en escenarios no supervisados, la marginal $p(x)$ Shlezinger y Routtenberg, 2023. Un criterio estándar de entrenamiento es la estimación por máxima verosimilitud, dada por:

$$\phi^* = \arg \max_{\phi} \sum_{i=1}^n \log p_\phi(x_i), \quad (2.3)$$

donde $p_\phi(x)$ representa una familia paramétrica de densidades, y $\arg \max$ denota el operador que devuelve el valor de ϕ que maximiza la suma de log-verosimilitudes. Cheng et al., 2017. Cuando el objetivo depende de un conjunto de variables de contexto c (por ejemplo, características del sitio), el problema se expresa como modelado condicional:

$$\phi^* = \arg \max_{\phi} \sum_{i=1}^n \log p_\phi(x_i | c_i), \quad (2.4)$$

lo que permite capturar cambios sistemáticos en la distribución de x inducidos por el contexto.

Los modelos generativos abarcan varias familias de modelado probabilístico. Entre estas familias se encuentran los autoencoders variacionales, que introducen variables latentes y recu-

ren a aproximaciones variacionales para el ajuste del modelo, y las redes generativas adversarias, orientadas a la generación de muestras mediante un esquema de entrenamiento adversarial. Estas diferencias muestran que no todas las familias generativas ofrecen el mismo tipo de acceso a la distribución objetivo ni responden de la misma manera a problemas de estimación de densidad Bond-Taylor et al., 2022.

En esta investigación, el objetivo no se limita solamente a generar muestras compatibles con los datos, sino a modelar distribuciones condicionadas del espectro energético y angular de la componente muónica a partir de descriptores del sitio. Esto requiere una formulación probabilística explícita que permita evaluar la densidad del modelo y entrenarla de manera directa por máxima verosimilitud. Bajo este criterio, en el siguiente apartado se introducen los flujos normalizantes condicionales (CNFs, por sus siglas en inglés), como una estrategia generativa para este propósito.

2.2.3. *Flujos Normalizantes y su extensión condicionada*

Los flujos normalizantes (NFs, por sus siglas en inglés) constituyen una familia de modelos generativos orientada al aprendizaje explícito de densidades Papamakarios et al., 2021. Su formulación parte de una transformación paramétrica f_ϕ , invertible y diferenciable, que conecta una distribución base conocida $p_Z(z)$, por ejemplo una gaussiana multivariada, con la distribución objetivo en el espacio de datos. Si z es una variable aleatoria en el espacio base y x una observación en el espacio de interés, el modelo establece la relación $x = f_\phi(z)$, con inversa $z = f_\phi^{-1}(x)$. La densidad inducida sobre x puede escribirse como

$$p_\phi(x) = p_Z(z) \left| \det J_{f_\phi^{-1}}(x) \right|. \quad (2.5)$$

En esta expresión, $p_Z(z)$ evalúa la densidad de la muestra una vez transformada al espacio base, mientras que $J_{f_\phi^{-1}}(x)$ denota la matriz jacobiana de la transformación inversa evaluada en x , cuyos elementos corresponden a las derivadas parciales de f_ϕ^{-1} . Su determinante corrige dicha densidad de acuerdo con el cambio local de volumen inducido por la transformación al pasar de x

a z Papamakarios et al., 2021.

La transformación total se define como la composición de varias transformaciones invertibles

$$f_\phi = f_K \circ f_{K-1} \circ \cdots \circ f_1. \quad (2.6)$$

Cada capa transforma sucesivamente la representación generada por la anterior, de modo que la composición permite construir transformaciones complejas a partir de bloques elementales. Para el ajuste del modelo se trabaja con la log-verosimilitud, ya que esta forma permite expresar la densidad inducida como la suma de la densidad en el espacio base y de los términos asociados al jacobiano de cada capa. En consecuencia,

$$\log p_\phi(x) = \log p_Z(z) + \sum_{k=1}^K \log \left| \det J_{f_k^{-1}}(h_k) \right|, \quad (2.7)$$

donde h_k denota las variables intermedias obtenidas a lo largo de la secuencia de transformaciones inversas Lu y Huang, 2020.

Esta construcción puede emplearse sobre una sola variable, caso en el cual el flujo modela una densidad univariada y permite estimar la distribución marginal. Sin embargo, no está restringida a ese escenario. También puede definirse sobre vectores $x \in \mathbb{R}^D$, donde D denota la dimensión del espacio de datos, de modo que el modelo represente una densidad conjunta sobre varias variables dentro de una única transformación Papamakarios et al., 2021.

Existen varias familias de transformaciones invertibles utilizadas en la construcción de los NFs, entre ellas las transformaciones de acoplamiento y las transformaciones autoregresivas. En esta investigación se emplean transformaciones autoregresivas enmascaradas con *splines* racionales-cuadráticas por tramos, siguiendo la formulación propuesta por Papamakarios et al. (2017) y Durkan et al. (2019). En una transformación autoregresiva, cada componente de salida se obtiene a partir de la componente correspondiente de entrada, mientras que los parámetros de dicha transformación se calculan según un orden prefijado a partir de las componentes anteriores. En el caso bidimensional considerado en este trabajo, esta dependencia puede presentarse como

$$y_1 = \tau_1(x_1), \quad y_2 = \tau_2(x_2; x_1), \quad (2.8)$$

donde la transformación de la primera componente y_1 depende únicamente de x_1 , mientras que la segunda y_2 se define sobre x_2 y puede estar parametrizada por la información contenida en x_1 .

El carácter enmascarado de la arquitectura indica que esta restricción de dependencia se impone mediante máscaras binarias en la red que produce los parámetros de cada transformación. En este caso, dichas máscaras eliminan las conexiones que introducirían dependencias fuera del orden autoregresivo. Dado que la primera componente y_1 no depende de x_2 , se cumple que

$$\frac{\partial y_1}{\partial x_2} = 0. \quad (2.9)$$

Como consecuencia, la matriz jacobiana de la transformación adopta una estructura triangular

$$J(x) = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & 0 \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{pmatrix}, \quad (2.10)$$

por lo que su determinante puede calcularse como el producto de sus elementos diagonales

$$\det J(x) = \frac{\partial y_1}{\partial x_1} \frac{\partial y_2}{\partial x_2}. \quad (2.11)$$

Además, entre capas sucesivas pueden intercalarse permutaciones para modificar el orden efectivo de dependencia entre variables y evitar que toda la composición opere siempre bajo la misma factorización Papamakarios et al., 2017.

Sobre esta estructura autoregresiva, la transformación de cada componente se define mediante una *spline* racional-cuadrática por tramos. Esto significa que el dominio de la variable se divide en intervalos y que, dentro de cada uno, la relación entre entrada y salida se describe mediante una función racional-cuadrática. La red interna produce los parámetros que determinan la posición de los nodos y la forma local de cada tramo Durkan et al., 2019. Frente a transformaciones afines, esta parametrización ofrece mayor flexibilidad local sin abandonar la invertibilidad

analítica ni el cálculo explícito de derivadas.

Debido a que el ajuste de los parámetros se realiza por máxima verosimilitud, el entrenamiento suele formularse como la minimización de la log-verosimilitud negativa (NLL)

$$\mathcal{L}_{\text{NLL}}(\phi) = -\frac{1}{N} \sum_{n=1}^N \log p_{\phi} \left(x^{(n)} \right), \quad (2.12)$$

donde N es el número de muestras del conjunto de entrenamiento. Minimizar esta función permite ajustar la transformación f_{ϕ} para que la distribución inducida por el flujo aproxime la distribución empírica de los datos Lu y Huang, 2020.

Sobre esta base, los CNFs extienden la formulación al caso en que la distribución objetivo dependa de un vector de contexto c . Si se define $z = f_{\phi}^{-1}(x; c)$, la densidad condicional puede escribirse como

$$\log p_{\phi}(x | c) = \log p_Z(z) + \sum_{k=1}^K \log \left| \det J_{f_k^{-1}}(h_k; c) \right|. \quad (2.13)$$

La diferencia con el caso no condicional es que el contexto modula la transformación. Se incorpora c como entrada adicional en las redes que generan los parámetros de cada capa, de modo que un solo modelo pueda representar una familia de distribuciones condicionadas. Esto permite estimar $p_{\theta}(x | c)$ y generar muestras con distintos valores del contexto sin entrenar una arquitectura independiente para cada condición Lu y Huang, 2020.

La misma formulación que hace tratables a los flujos impone también sus principales restricciones. Condiciones como la evaluación exacta de la densidad y el uso de transformaciones invertibles limitan el tipo de transformaciones que pueden emplearse y obligan a equilibrar capacidad de representación y costo computacional.

En *Wavelet Flow*, diseñado para disminuir el costo de entrenamiento de flujos de alta resolución, se reportan aceleraciones hasta 15 veces más rápidas frente a flujos previos en ciertos *benchmarks*, y se indica que algunas formulaciones podían requerir meses de entrenamiento en GPU Yu et al., 2020.

La revisión de Papamakarios et al. (2021) muestra que la evaluación de densidad y la generación de muestras no presentan el mismo costo computacional, ya que una de estas operaciones debe realizarse de forma secuencial sobre las dimensiones del vector de entrada. Como consecuencia, el costo de muestreo aumenta con la dimensionalidad del problema y puede volverse menos favorable que la evaluación de la densidad.

Una limitación adicional radica en que su desempeño puede deteriorarse en regiones de baja densidad, en particular en las colas de la distribución, donde los datos suelen ser más escasos. En estos casos, tienden a reproducir mejor las zonas de alta densidad que los eventos extremos Laszkiewicz et al., 2022.

En este marco, resulta pertinente revisar aplicaciones desarrolladas en problemas afines, en las que los CNFs han mostrado capacidad para representar distribuciones complejas y modelar dependencias condicionadas. Estos casos se presentan en la sección siguiente y sustentan la adopción de los CNFs en esta investigación.

2.2.4. Implementación de CNFs en dominios análogos

Las aplicaciones recientes de los CNFs muestran que esta familia de modelos puede emplearse en problemas científicos donde no solo interesa obtener una predicción puntual, sino también describir la incertidumbre asociada mediante distribuciones condicionales explícitas. La literatura reporta tres usos recurrentes: reconstrucción probabilística de variables físicas a partir de señales experimentales, generación condicional de campos o variables multiescala y aceleración de procedimientos de muestreo en sistemas donde los métodos convencionales presentan altos costos computacionales IceCube Collaboration, 2024; Singha et al., 2023; Winkler et al., 2023.

En reconstrucción de eventos físicos, destaca el trabajo de IceCube Collaboration (2024), donde se entrenaron CNFs para aproximar distribuciones posteriores por evento de la energía y la dirección de neutrinos a partir de resúmenes de señales del detector. En ese estudio se realizaron tres entrenamientos: dos sobre aproximadamente 1.2 millones de eventos de interacciones cargadas

de neutrinos electrónicos para reconstruir energía o dirección, y uno adicional sobre 1.4 millones de eventos de neutrinos muónicos para reconstrucción direccional. La evaluación se realizó con conjuntos de prueba independientes de alrededor de 100 000 eventos cada uno. En la reconstrucción direccional de cascadas, los autores reportan una resolución angular mediana equivalente a $5\text{--}10^\circ$, valor que describen como cercano al de reconstrucciones de referencia basadas en likelihood.

En modelación climática, los CNFs también se han explorado para reducción de escala estadística, con el objetivo de reconstruir campos de alta resolución a partir de entradas de baja resolución. En el trabajo de Winkler et al. (2023), el modelo se evaluó sobre datos ERA5 de contenido total de agua y se analizó en factores de ampliación $2\times$ y $4\times$. Frente a la interpolación bicúbica, el CNF redujo el error absoluto medio de 6.96 a 5.22 en $2\times$ y de 6.90 a 5.26 en $4\times$; asimismo, el error cuadrático medio descendió de 8.44 a 5.72 en $2\times$ y de 8.37 a 5.80 en $4\times$. Frente a una red generativa antagónica, el comportamiento fue similar en MAE y RMSE, pero el CNF obtuvo mejores valores de CRPS, con 0.0150 frente a 0.0373 en $2\times$ y 0.0174 frente a 0.0454 en $4\times$, lo que respalda su utilidad en evaluación probabilística. El estudio también muestra mapas de incertidumbre estimados a partir de 20 muestras generadas para una misma entrada y reporta un entrenamiento de 35 épocas.

Una tercera línea de trabajo utiliza CNFs como mecanismo de propuesta dentro de esquemas de Markov chain Monte Carlo para mitigar la ralentización crítica. En este enfoque, el modelo se entrena con muestras obtenidas en regiones no críticas y luego se emplea, por interpolación o extrapolación, para generar muestras en la región crítica. El entrenamiento del modelo toma aproximadamente 5–6 horas en una GPU Tesla P100 de Colab y, una vez ajustado, la generación de muestras requiere solo 5–7 minutos, con una tasa de aceptación cercana al 40%. Además, los autores muestran que el mismo modelo puede reutilizarse, lo que reduce la necesidad de entrenar un modelo distinto para cada condición Singha et al., 2023.

En conjunto, estos antecedentes muestran que los CNFs resultan útiles cuando el problema exige modelar distribuciones condicionales, generar múltiples realizaciones para un mismo contexto y mantener acceso explícito a densidades. Al mismo tiempo, la literatura coincide en va-

rias limitaciones: la necesidad de conjuntos de entrenamiento amplios y representativos, el costo computacional del ajuste y la sensibilidad del modelo al rango de condiciones cubierto durante el entrenamiento. Desde esta perspectiva, el interés de los CNFs para la presente investigación no radica solo en su capacidad generativa, sino en que ofrecen un marco adecuado para estimar espectros condicionados con incertidumbre explícita y muestreo eficiente bajo diferentes condiciones físicas IceCube Collaboration, 2024; Singha et al., 2023; Winkler et al., 2023.

2.3. Generalidades del entorno de desarrollo web

En esta sección se presentan algunas bases conceptuales del entorno de desarrollo *web*, necesarias para comprender la organización general de una aplicación.

2.3.1. *Arquitectura de software*

2.3.1.1. Arquitectura cliente–servidor. La arquitectura cliente–servidor es un modelo utilizado en el desarrollo de aplicaciones *web*. En este esquema, el cliente y el servidor cumplen funciones distintas dentro del sistema. El cliente se encarga de la interacción con el usuario, mientras que el servidor recibe solicitudes, ejecuta el procesamiento correspondiente y devuelve una respuesta. Desde el punto de vista funcional, el cliente puede entenderse como la capa de presentación. En ella se encuentran los elementos con los que interactúa el usuario, como formularios, controles de navegación, vistas de resultados y mecanismos de entrada de datos. El servidor, por su parte, constituye la capa de procesamiento. Allí se ubican la validación de datos, la lógica del sistema, el acceso a recursos internos y las operaciones que no dependen de la interfaz gráfica.

Bajo este esquema, la presentación y el procesamiento permanecen desacoplados, de modo que la interfaz puede modificarse sin intervenir directamente en la lógica interna, y la lógica del servidor puede ajustarse sin necesidad de rediseñar la capa visual.

2.3.1.2. Servicios *web* y API REST. Sobre esta arquitectura, la comunicación entre cliente y servidor suele establecerse mediante servicios *web*. Un servicio *web* puede entenderse como un mecanismo de intercambio de información entre aplicaciones a través de una red, bajo reglas de comunicación definidas. En el desarrollo *web* actual, esta interacción suele implementarse mediante API (*Application Programming Interface*, por sus siglas en inglés), es decir, interfaces que exponen funciones o recursos para ser consumidos por otros componentes del sistema Snodgrass y Milkov, 2020.

Uno de los estilos de diseño empleados con este fin es REST (*Representational State Transfer*, por sus siglas en inglés), un estilo arquitectónico para diseñar servicios web mediante recursos, representaciones e interfaces uniformes. Bajo este enfoque, la comunicación se apoya en el protocolo HTTP (*Hypertext Transfer Protocol*, por sus siglas en inglés), que define como clientes y servidores intercambian solicitudes y respuestas en la web, y en operaciones definidas sobre recursos. La API establece puntos de acceso mediante los cuales el cliente puede enviar solicitudes y recibir respuestas estructuradas, de manera que el cliente no interactúa directamente con la lógica interna del servidor Fielding et al., 2022; Lahkar, 2023.

En este esquema, cada solicitud contiene la información necesaria para ser atendida. Esto permite mantener una comunicación ordenada y facilita que un mismo servicio pueda ser consumido por distintos clientes siempre que respeten el formato definido. Además, el intercambio de datos suele realizarse mediante representaciones ligeras, como JSON, adecuadas para transportar parámetros, resultados y mensajes de respuesta de manera estandarizada.

2.3.2. Tecnologías y criterios de desarrollo *web*

El desarrollo de aplicaciones *web* suele apoyarse en *frameworks*, es decir, estructuras de software que proporcionan convenciones, componentes y herramientas para organizar el código de manera sistemática. Un *framework* no define por sí mismo la aplicación, pero sí ofrece una base de trabajo sobre la cual se construye, con una organización previa para distribuir responsabilidades

entre sus partes.

Los *frameworks* de *frontend* se enfocan en la construcción de la interfaz de usuario y en la gestión de la interacción en el cliente. Por otro lado, los *frameworks* de *backend* se orientan a la definición de servicios, el procesamiento de solicitudes y la administración de la lógica del servidor. Esta distinción corresponde a la separación introducida por la arquitectura cliente–servidor y permite mantener una organización más clara del desarrollo Mo y Li, 2025.

2.3.2.1. Angular. Angular es un *framework* orientado al desarrollo de aplicaciones *web* de una sola página. Su modelo de construcción se basa en componentes, entendidos como unidades que encapsulan la estructura, la lógica y los estilos asociados a una parte de la interfaz. Esta organización permite construir la capa cliente como una composición de secciones diferenciadas Angular, 2026.

Dentro de Angular, los componentes no operan de manera aislada. La navegación entre vistas se gestiona mediante un sistema de rutas, mientras que la lógica compartida y la comunicación con servicios externos suele concentrarse en servicios específicos. De este modo, la interfaz, mantiene una organización interna en la que presentación, navegación y acceso a datos se distribuyen de forma separada.

Otra característica del *framework* es su integración con TypeScript, lenguaje que incorpora tipado estático sobre JavaScript. Esto permite definir con mayor precisión las estructuras de datos manejadas por la aplicación y mantener mayor consistencia en proyectos con varias vistas, formularios y consumo de servicios HTTP.

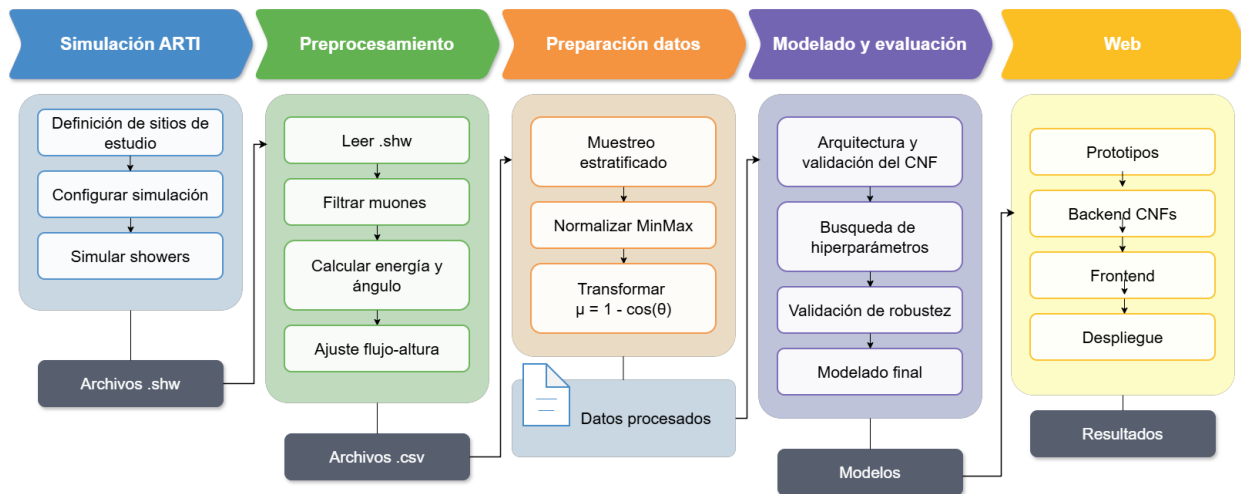
2.3.2.2. FastAPI. FastAPI es un *framework* para el desarrollo de servicios *web* en Python. Su propósito es facilitar la construcción de aplicaciones *backend* orientadas a exponer funcionalidades mediante solicitudes HTTP. Para ello, se apoya en anotaciones de tipos para definir con claridad los datos de entrada y salida, incorpora validación sobre la información recibida y genera documentación automática de la API a partir del esquema OpenAPI Lahkar, 2023.

Estas características favorecen una organización clara de los servicios del *backend*. Al definir de forma explícita qué datos recibe cada operación y qué estructura tendrá la respuesta, el servidor puede construirse a partir de puntos de acceso, cada uno asociado a una función específica. A ello se suma el sistema de dependencias de FastAPI, que permite incorporar componentes auxiliares dentro de los servicios sin mezclar su definición con la lógica principal de cada operación.

2.3.2.3. Diseño web responsive. Además del uso de *frameworks* para estructurar la aplicación, la construcción de la interfaz también puede apoyarse en criterios de diseño orientados a su adaptación visual. En este sentido, el diseño *web responsive* corresponde a un enfoque de desarrollo de interfaces que busca ajustar la presentación de una aplicación a distintos tamaños de pantalla y condiciones de visualización. Para ello, suele apoyarse en recursos como contenedores flexibles, sistemas de rejilla, unidades relativas y reglas condicionales de estilo mediante *media queries*, de modo que una misma interfaz pueda reorganizarse sin modificar su funcionalidad general y conserve una estructura coherente en diferentes dispositivos.

3. Metodología

El enfoque metodológico se dividió en cinco etapas detalladas en la Figura 1. El proceso inició con las fases de Simulación ARTI y Preprocesamiento, dedicadas a la generación y filtrado de datos. Posteriormente, se llevó a cabo la Preparación de datos para optimizar el entrenamiento del modelo. La etapa de Modelado y evaluación se centró en el desarrollo y la validación de la arquitectura CNF para la caracterización del espectro de muones. Finalmente, el flujo concluyó con la fase Web, donde se efectuó el desarrollo de una plataforma funcional para visualización de resultados del modelo.

Figura 1.*Diagrama del flujo metodológico abordado.*

Nota. La figura resume las cinco etapas del procedimiento metodológico desarrollado. En conjunto, estas etapas abarcan los cuatro objetivos específicos del proyecto y, por consiguiente, el objetivo general. Cada etapa concluye con un resultado que da inicio a la etapa siguiente, lo que define una secuencia de trabajo clara y organizada.

3.1. Generación de un conjunto de datos balanceado a partir de ARTI/CORSIKA

En esta etapa se llevó a cabo el desarrollo de las simulaciones correspondientes a las ciudades, de modo que se buscaba generar un conjunto de datos balanceado usando ARTI. Cabe resaltar que las simulaciones se ejecutaron en un clúster remoto denominado obatalá, al que se accedió mediante el protocolo de red para acceso remoto seguro SSH (*Secure Shell*, por sus siglas en inglés). Este entorno contaba con arquitectura x86_64, 8 CPU (*Central Processing Unit*, por sus siglas en inglés) lógicas visibles, un procesador Intel Xeon W3530 a 2.80 GHz, 15 GiB de memoria RAM (*Random Access Memory*, por sus siglas en inglés) y 7.5 GiB de *swap*. El sistema anfitrión operaba sobre Debian 13.

3.1.1. Generación y exploración preliminar de datos

3.1.1.1. Selección de los sitios de estudio. Se priorizó incorporar variabilidad en la altitud sobre el nivel del mar y en la intensidad del campo geomagnético. En consecuencia, se escogieron ubicaciones distribuidas en distintos intervalos de ambas variables, con el fin de conformar un conjunto de simulación representativo de condiciones físicas diversas.

Como variable geomagnética de agrupación se empleó B_i , entendida como la intensidad total del campo magnético en una ubicación. De este modo, se trabajó con una única magnitud de referencia para organizar los sitios según su condición geomagnética, acotada en el intervalo $B_i \in [23, 57] \mu\text{T}$. Para la altura h , se tomó como límite superior de referencia el valor correspondiente a Chacaltaya, $h = 5230 \text{ m s. n. m.}$, de modo que esta variable quedó acotada en el intervalo $h \in [0, 5230] \text{ m}$.

A partir de lo anterior, se estableció un esquema de muestreo estratificado en función de h y B_i . Este enfoque consiste en dividir los datos en estratos definidos sobre variables de interés y seleccionar subconjuntos representativos dentro de cada uno de ellos Nguyen et al., 2019. Además, para reducir la dependencia de cada estrato respecto a una única ciudad de referencia, se adoptó como criterio la simulación de dos ciudades por cada combinación de intervalos de h y B_i . Los rangos definidos para cada variable se presentan en la Tabla 1.

Tabla 1.

Rangos de B_i y h definidos para el muestreo estratificado.

Variable	Rangos
h (m)	0–1000, 1000–2000, 2000–3000, 3000–4000, 4000+
B_i (μT)	≤ 33 , 33–41, 41–49, > 49

Nota. La tabla presenta los rangos establecidos para el muestreo estratificado en función de las variables h y B_i . La variable h representa la altitud de la ciudad sobre el nivel del mar, mientras que B_i corresponde a la intensidad del campo geomagnético asociada a ese sitio.

Con base en este esquema de estratificación se realizó la selección de ciudades. En una primera etapa se priorizaron sitios donde se encuentra la colaboración LAGO LAGO Collaboration, 2022, debido a que constituían un conjunto inicial de referencia para la construcción del

conjunto de simulaciones. Sin embargo, esta selección no cubría todos los estratos definidos. Por esta razón, posteriormente se incorporaron otras ciudades hasta completar los intervalos de interés establecidos en el muestreo.

La incorporación de ciudades se desarrolló de manera progresiva. Inicialmente se seleccionaron sitios ubicados entre 0 y 1000 m, cubriendo los distintos rangos de B_i . Posteriormente, se añadieron ciudades del rango $B_i \leq 33 \mu\text{T}$ a lo largo de los diferentes intervalos de h . Esta configuración inicial se definió con el propósito de disponer, desde etapas tempranas, de datos representativos y, además, habilitar las primeras pruebas del modelo de inteligencia artificial. Por ello, se priorizó un orden de incorporación que permitiera cubrir la mayor cantidad posible de valores de h y B_i con el menor número de ciudades necesarias.

3.1.1.2. Preparación de la simulación. Una vez definidos los sitios de estudio, fue necesario organizar la información requerida para preparar cada caso de simulación en ARTI/CORSIKA¹. Para ello, se construyó un archivo JSON que reunió todas las ubicaciones a simular en una estructura homogénea. Cada entrada incluyó un identificador del sitio, su nombre, país, latitud, longitud, altura, así como los parámetros geomagnéticos B_x , B_z y B_i . Además, se añadió una etiqueta de tipo para distinguir entre los sitios pertenecientes a la colaboración LAGO y aquellos incorporados de forma adicional. Un ejemplo de la estructura empleada para representar un sitio dentro del archivo JSON se presenta en el Apéndice A. De este modo se unificaron los sitios disponibles en LAGO y los añadidos posteriormente. En el caso de las ciudades correspondientes a LAGO, la simulación es directa, ya que ARTI cuenta con esquemas predefinidos para estas ubicaciones. En cambio, para los sitios añadidos fue necesario los parámetros de las componentes geomagnéticas B_x y B_z , y de altura requeridos por el esquema de simulación.

Para las ubicaciones añadidas manualmente, los valores de B_x y B_z se obtuvieron a partir de las coordenadas geográficas de cada punto mediante la calculadora geomagnética del NCEI/NOAA². En este contexto, B_x corresponde a la componente horizontal del campo magnético terrestre y B_z a

¹El repositorio de ARTI se encuentra en el siguiente enlace: <https://github.com/lagoproject/arti>

²La calculadora se encuentra disponible en: <https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml?>

su componente vertical.

Con el fin de mantener condiciones comparables entre todos los casos, se fijó un tiempo de simulación de 3600s para cada ubicación. De esta manera, todas las corridas representaron el flujo de partículas correspondiente a una hora en el sitio considerado. Cabe resaltar que, aunque en el texto se hace referencia general a ciudades, no todas las ubicaciones correspondieron estrictamente a centros urbanos; en algunos casos fue necesario emplear pueblos o puntos de montaña para cubrir combinaciones específicas de altura y campo geomagnético dentro del esquema de muestreo.

3.1.1.3. Tiempos de una simulación. A continuación se presenta un resumen exploratorio de los pasos más relevantes, en cuanto a tiempo de simulación se refiere, para ejecutar una simulación en ARTI de una ciudad correspondiente a LAGO. Este procedimiento se desarrolló tomando como guía el workshop elaborado por Rafael Martínez sobre ARTI y MEIGA, disponible en ³.

Al lanzar la simulación, ARTI genera un conjunto de 15 *scripts* ejecutables asociados a distintos grupos de partículas: uno para helio, ocho para protones y seis para el resto de partículas. Su ejecución se automatizó por fases mediante el script runner .sh, cuyo contenido se presenta en el Apéndice B, de modo que se ejecutan lotes en paralelo de 3 en 3, incorporando pausas y verificaciones de menos de un minuto entre ejecuciones para evitar sobrecargar los recursos del sistema. Luego de esta ejecución se generan archivos binarios, de los cuales se extrae su información a formato de texto para luego generar un unico archivo de salida con los datos de la simulación.

A partir de lo anterior, el costo temporal asociado los tiempos de ejecución observados presentaron variación entre los distintos sitios. En la etapa identificada como runner, los valores registrados oscilaron entre 5 h y 8 h 57 m. Para ilustrar este comportamiento, en la Tabla 2 se presentan cinco casos, junto con una estimación del tiempo total por simulación al considerar adicionalmente la etapa de binarios, con una duración promedio de 140 minutos por cada simulación.

³El repositorio del workshop se encuentra disponible en el siguiente enlace: https://gitmilab.redclara.net/martinezr/lago_workshop_arti_meiga

Tabla 2.*Ejemplos de tiempos de ejecución para distintas simulaciones.*

Ciudad	$h(m)$	$B_i(\mu T)$	Runner	Total aprox.
Lima	150	24.751	5 h 00 m	7 h 20 m
San Jose	1172	33.631	5 h 33 m	7 h 53 m
Puebla	2143	39.314	6 h 02 m	8 h 22 m
Kerman	3150	47.069	7 h 41 m	10 h 01 m
Komic	4587	47.884	6 h 56 m	9 h 16 m
Jungfrauoch	3454	47.986	8 h 57 m	11 h 17 m

Nota. La tabla presenta ejemplos de tiempos de simulación para seis ciudades, junto con sus valores de altura h e intensidad del campo geomagnético B_i . La columna Runner corresponde al tiempo registrado durante la ejecución automatizada de los scripts, mientras que la columna Total aprox. incorpora además una estimación de 140 minutos promedio asociada a la etapa de procesamiento de binarios.

Como se observa en la tabla, la simulación que menos tiempo tardó fue aproximadamente siete horas y veinte minutos, para las simulaciones de más altura y campo geomagnético se superaron las once horas de tiempo de simulación. Además, se puede apreciar que a mayores alturas, la simulación parece tardar mas.

3.1.1.4. Filtrado de muones. Una vez finalizada la simulación, las salidas generadas por ARTI contenían, entre otras, las columnas CorsikaId, px, py, pz, x, y, z, shower_id, prm_id, prm_energy, prm_theta y prm_phi, como se muestra en la Figura 2. Para el presente trabajo, las variables de interés principal fueron CorsikaId, px, py y pz. La columna CorsikaId permitió filtrar únicamente las partículas correspondientes a muones, mientras que las componentes del momento se utilizaron para calcular tanto la energía como el ángulo cenital.

Figura 2.*Salida generada por ARTI*

```
# # # shw
# # CURVED mode is ENABLED and observation level is 956 m a.s.l.
# # This is the Secondaries file - ARTI      v1r9
# # 12 column format is:
# # CorsikaId px py pz x y z shower_id prm_id prm_energy prm_theta prm_phi
0013 -3.74963e-02 -2.80750e-02 +3.30317e-01 -2.16934e+04 -4.26298e+04 +7.76477e+02 00000001 0703 +6.01558e+01 +66.367 +062.581
0006 +7.62329e-02 -2.93719e-02 +1.60923e-01 +2.48316e+03 -8.80476e+02 +9.55455e+02 00000002 0703 +5.95206e+01 +40.978 -047.714
0006 -5.69543e-01 +1.92752e+00 +2.41325e+00 -3.18668e+02 -1.79414e+03 +9.55739e+02 00000003 0703 +3.16047e+02 +40.766 +104.774
```

Nota. En la figura se muestran las primeras filas de un archivo generado por ARTI, en las que se aprecian las columnas correspondientes a las diferentes características y variables asociadas a cada partícula simulada.

La energía total del muón se obtuvo a partir de la relación relativista energía–momento,

$$E^2 = p^2 + m_\mu^2, \quad (3.1)$$

Particle Data Group, 2013, donde p es el módulo del momento y m_μ corresponde a la masa del muón. Al expresar dicho módulo en términos de sus componentes cartesianas, $p^2 = p_x^2 + p_y^2 + p_z^2$, se obtiene

$$E = \sqrt{p_x^2 + p_y^2 + p_z^2 + m_\mu^2}. \quad (3.2)$$

Por su parte, el ángulo cenital se calculó como el ángulo polar respecto al eje z , a partir de la relación

$$\theta = \arccos \left(\frac{p_z}{p} \right), \quad (3.3)$$

donde p denota el módulo del momento, definido como

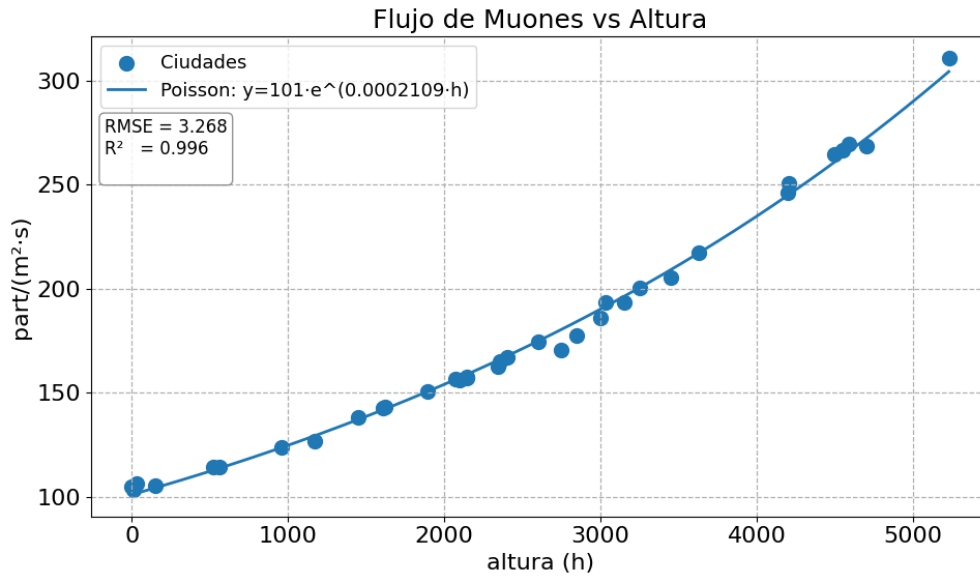
$$p = \sqrt{p_x^2 + p_y^2 + p_z^2}. \quad (3.4)$$

De este modo, fue posible generar los espectros energético y angular de cada ciudad simulada para su posterior análisis.

3.1.1.5. Ajuste del flujo de muones vs altura. Adicionalmente, se verificó la consistencia física de las simulaciones mediante el análisis del flujo de muones en función de la altura del sitio, como se muestra en la Figura 3. Se observa para 30 ciudades simuladas, el flujo de muones obtenido a partir del número de eventos registrados por unidad de tiempo, junto con un ajuste exponencial en función de la altura.

Figura 3.

Ajuste mediante máxima verosimilitud de Poisson de la relación entre el flujo integrado de muones y la altura del sitio simulado.



Nota. Los puntos muestran el flujo de muones por ciudad en función de la altura, que se ajustan bajo un supuesto de Poisson. Se aprecia además que, para alturas cercanas al nivel del mar, el flujo es aproximadamente tres veces menor que el observado para la ciudad situada a 5230 m s. n. m.

La Figura 3 muestra una relación creciente entre el flujo de muones y la altura del sitio. Esta tendencia se modeló mediante una función exponencial obtenida por máxima verosimilitud, cuya expresión es:

$$\Phi(h) = 101, e^{0.0002109, h}, \quad (3.5)$$

donde h representa la altura en metros sobre el nivel del mar y $\Phi(h)$ el flujo en $\text{part}/(\text{m}^2 \cdot \text{s})$. El coeficiente del exponente indica un crecimiento multiplicativo: por cada 1000 m de ascenso, el flujo se incrementa en aproximadamente un 23.5 % ($e^{0.2109} \approx 1.235$). La alta precisión de este modelo se refleja en un coeficiente de determinación $R^2 = 0.996$ y un $\text{RMSE} = 3.268$, lo que confirma que la función captura casi la totalidad de la variabilidad de los datos simulados. Gracias a este ajuste, la ecuación 3.5 permite realizar interpolación para estimar el flujo esperado en cualquier punto dentro del rango altitudinal estudiado. En la práctica, esta aproximación permitió parametrizar la cantidad de eventos a generar en etapas posteriores del análisis, garantizando que el flujo de muones sea consistente con la ubicación geográfica de cada ciudad simulada.

3.2. Estimación y evaluación del espectro de muones para el CNF

El modelado se realizó de forma condicional respecto a h , cuya relación exponencial con el flujo se estableció en la sección anterior, y sobre B_x y B_z que determinan la rigidez del corte geomagnético del sitio. Para ello, se entrenó un único CNF bidimensional que estima la densidad condicional conjunta de la energía y el ángulo cenital de la componente muonica.

3.2.1. Subconjunto preliminar de datos

Se inició con la búsqueda de un subconjunto preliminar de datos para las etapas de búsqueda de hiperparámetros y validación de robustez del CNF, con el fin de disminuir el costo computacional asociado al entrenamiento y evaluación del modelo.

La selección de las ubicaciones se realizó buscando una cobertura amplia en h , lo que dio lugar a distintas combinaciones de h , B_x y B_z . La Tabla 3 presenta la organización de estas ciudades en los subconjuntos de entrenamiento y validación, junto con el número de eventos asociado a cada una.

Tabla 3*Ciudades incluidas en entrenamiento y validación para el subconjunto preliminar*

No.	Ciudad	h (m)	B_x (μ T)	B_z (μ T)	Eventos
Entrenamiento					
1	Londres	24	19.542	45.044	374,028
2	Iqaluit	34	9.630	55.928	382,657
3	Chiapas	522	27.405	27.024	411,717
4	Bucaramanga	956	26.793	16.055	445,442
5	Brasilia	1172	20.148	-11.931	456,932
6	Malargüe	1450	18.987	-14.326	496,719
7	Creighton	2100	17.037	51.991	562,403
8	Pamplona	2342	26.743	15.955	584,908
9	Casposo	2600	19.473	-12.436	627,197
10	Riobamba	2750	26.577	8.499	633,342
11	Berlin	3450	26.751	16.029	739,526
12	La Paz	3630	22.362	-4.708	781,231
13	Andes	4200	19.658	-11.951	885,994
14	Sierra Negra	4550	27.358	28.038	959,411
15	Chacaltaya	5230	22.380	-4.620	1,118,185
Validación					
1	Lima	150	24.723	-0.476	378,691
2	Guatemala	1490	27.532	24.932	491,363
3	Quito	2850	26.717	9.971	638,657
4	Imata	4600	22.969	-3.790	955,307
5	Atacama	5105	20.638	-8.598	1,074,499

Nota. Se presentan 15 ciudades en el subconjunto de entrenamiento y 5 en el de validación. En conjunto, las ubicaciones seleccionadas cubren alturas entre 24 y 5230 metros (m), valores de B_x entre 9.630 y 27.532 μ T y valores de B_z entre -14.326 y 55.928 μ T, donde μ T corresponde a microteslas.

La partición en los subconjuntos de entrenamiento y validación se definió con una proporción aproximada del 75 % y 25 % de las ciudades, respectivamente, preservando la diversidad de condiciones entre ambos subconjuntos. A partir de esta partición, el subconjunto quedó conformado por 9 459 692 eventos en entrenamiento y 3 538 517 eventos en validación.

3.2.2. *Procesamiento de los datos para el CNF*

3.2.2.1. Muestreo estratificado de eventos por ciudad. En el conjunto de ciudades se observó un desbalance en la cantidad de datos por ciudad. Como se evidencia en la Figura 3, los sitios ubicados a mayor altura sobre el nivel del mar contenían un número más alto de registros,

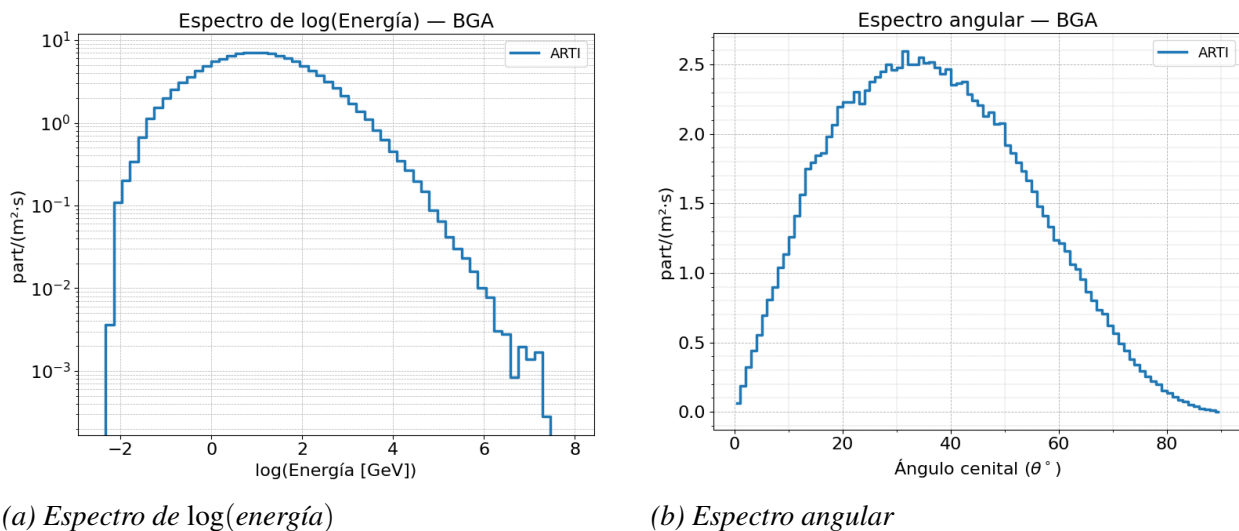
mientras que ciudades situadas a menor altura presentaban una cantidad menor; un ejemplo de este comportamiento es Chacaltaya, ubicada a una altura de 5230 metros, que presentó cerca de tres veces más partículas que Lima, ubicada a 24 metros sobre el nivel del mar. De haberse entrenado el modelo bajo estas condiciones, el ajuste habría quedado dominado por las ciudades con mayor número de observaciones.

Para mitigar este desbalance, se aplicó un muestreo estratificado, mencionado en la sección anterior, donde se dividen los datos en estratos definidos y se seleccionan subconjuntos representativos. En este caso la estratificación se empleó sobre los eventos de cada ciudad para reducir el desbalance entre sus tamaños muestrales.

En el caso de la energía, la variable presentaba valores en el rango de 10^{-1} a 10^4 GeV, por lo que se aplicó la transformación $z_E = \log(E)$ para reducir su dispersión. En el caso angular, se trabajó con el ángulo cenital $\theta \in [0^\circ, 90^\circ]$. La Figura 4 presenta los espectros de energía transformada y ángulo cenital para Bucaramanga, usados como referencia para visualizar la distribución de ambas variables antes del muestreo.

Figura 4.

Espectros de energía transformada y ángulo cenital para Bucaramanga.



Nota. Se presenta el espectro de energía de Bucaramanga (BGA), transformada mediante $\log(E)$ para reducir su dispersión, con E expresada en gigaelectronvoltios (GeV). También se muestra el espectro angular, donde θ corresponde al ángulo cenital expresado en grados. En ambos casos, el eje vertical indica el flujo estimado en $\text{part}/(\text{m}^2 \cdot \text{s})$ para los eventos generados con ARTI.

La Figura 4 muestra que la transformación logarítmica comprimió la energía en un intervalo aproximado de $[-2.5, 8]$, lo que redujo la dispersión asociada a su rango original. Por su parte, la variable angular presenta una mayor concentración de eventos en la zona central del intervalo, con una menor frecuencia hacia los extremos. A partir de estas representaciones, ambas variables se discretizaron en 100 intervalos (bins). Después, para cada evento se identificó el bin de energía y ángulo al que pertenecía. La pareja formada por estos dos intervalos se tomó como una etiqueta de estrato; por tanto, todos los eventos con la misma combinación quedaron agrupados en el mismo estrato.

El procedimiento se ejecutó fijando un objetivo de $N_{\text{obj}} = 370\,000$ eventos por ciudad. Este valor se definió a partir de la ciudad con menor cantidad de datos, Londres con 374\,028 eventos. De este modo, se garantizó un tamaño muestral común entre ciudades sin requerir sobremuestreo, al tiempo que se redujo el costo computacional de la fase exploratoria.

Para una ciudad s del conjunto de entrenamiento, y dado el número total de observaciones n_s , se definió un factor de retención proporcional dado por $\alpha_s = \frac{N_{\text{obj}}}{n_s}$.

Si $n_{s,b}$ representa el número de observaciones pertenecientes al bin b en la ciudad s , el número de registros retenidos en dicho estrato se calculó como

$$k_{s,b} = \text{round}(\alpha_s n_{s,b}) . \quad (3.6)$$

Una vez obtenidos los valores $k_{s,b}$, la selección se realizó aleatoriamente dentro de cada estrato. Cada ciudad aportó la misma fracción de datos en todos sus bins, en lugar de someterse a un recorte aleatorio global.

Este procedimiento se aplicó exclusivamente al subconjunto de entrenamiento. El subconjunto de validación se mantuvo en su distribución original, sin remuestreo, con el fin de conservar una referencia no intervenida para las etapas posteriores del modelado. Dado que el subconjunto preliminar quedó conformado por 9\,459\,692 de eventos en entrenamiento, el balanceo produjo un total de 5\,550\,000 eventos.

3.2.2.2. Transformación y normalización de variables. La energía, el ángulo y las variables condicionales no compartían una representación numérica homogénea. En CNFs, esta heterogeneidad ralentiza la convergencia del entrenamiento y provoca que las variables con mayor rango numérico dominen las actualizaciones de los parámetros, como los pesos y sesgos. En consecuencia, se definió un esquema de transformación y reescalamiento que permitió mapear todas las variables a un dominio común en el intervalo $[0, 1]$.

En cuanto a la energía, aunque la transformación logarítmica z_E redujo la dispersión de la variable al intervalo $[-2.5, 8]$, todavía fue necesario reescalarla a un dominio comparable con el resto de variables del modelo.

Por su parte, las variables condicionales del sitio presentaban diferencias en sus rangos de magnitud. Como se puede observar en la Tabla 3, la h variaba entre 24 y 5 230 m, B_x tomaba valores entre 9,630 y 27,532 μT , mientras que B_z se extendía entre -14,326 y 55,928 μT . Para resolver, el reescalamiento de z_E como el de las variables condicionales, se empleó una normalización Min-Max al intervalo $[0, 1]$. Este procedimiento transforma cada variable a un rango prefijado a partir de sus valores mínimo y máximo de Amorim et al., 2023. La transformación aplicada se expresó como

$$\tilde{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \quad x \in \{z_E, h, B_x, B_z\} \quad (3.7)$$

esta normalización conserva el orden de los valores y permite recuperar la escala original mediante la transformación inversa

$$x = \tilde{x}(x_{\max} - x_{\min}) + x_{\min}. \quad (3.8)$$

Los parámetros $x_{\min}$ y $x_{\max}$ se calcularon únicamente sobre el conjunto de entrenamiento y, posteriormente, se reutilizaron sin modificación sobre la validación.

El tratamiento del ángulo siguió una lógica distinta. Dado que $\theta \in [0^\circ, 90^\circ]$, se optó por representarlo mediante

$$\mu = 1 - \cos(\theta). \quad (3.9)$$

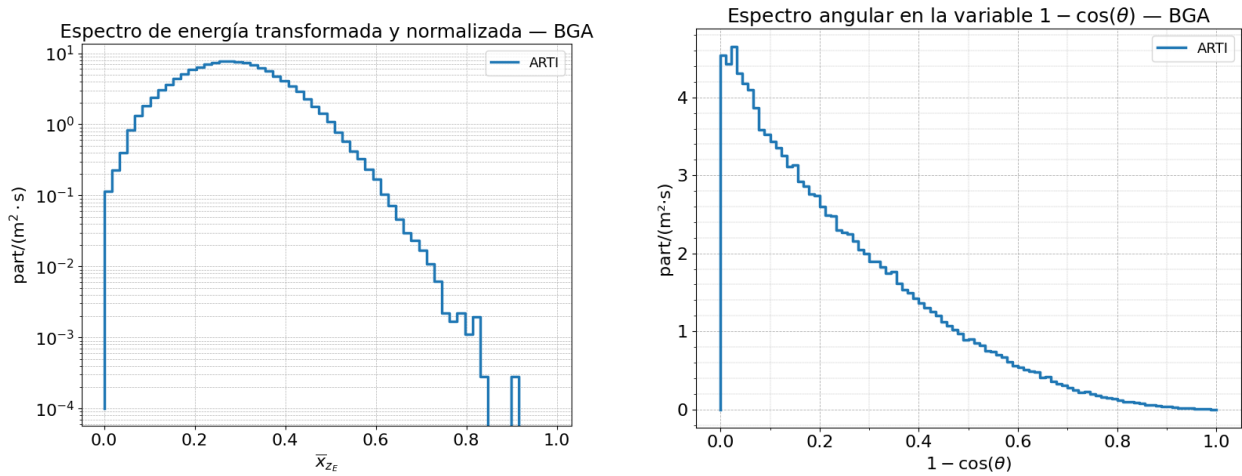
En lugar de aplicar una normalización Min-Max directa sobre θ . Esta transformación acota la variable al intervalo $[0, 1]$ y redistribuye los valores de manera que la mayor concentración de observaciones se ubica hacia un solo extremo del intervalo. La recuperación del ángulo en grados permanece disponible mediante la transformación

$$\theta = \arccos(1 - \mu). \quad (3.10)$$

Para complementar la descripción del preprocesamiento, la Figura 5 presenta la forma de las variables objetivo después de aplicar las transformaciones descritas.

Figura 5.

Distribución de las variables objetivo transformadas para Bucaramanga.



(a) *Energía transformada y normalizada*

(b) *Variable angular $1 - \cos(\theta)$*

Nota. Se presentan los espectros obtenidos para Bucaramanga (BGA) después del preprocesamiento de las variables objetivo. En (a), la energía, expresada en gigaelectronvoltios (GeV), fue transformada mediante $z_E = \log(E)$ y posteriormente normalizada al intervalo $[0, 1]$ mediante Min-Max. En (b), el ángulo cenital θ fue representado mediante $1 - \cos(\theta)$, también acotado en $[0, 1]$. En ambos casos, el eje vertical indica el flujo estimado en $\text{part}/(\text{m}^2 \cdot \text{s})$ para los eventos generados con ARTI.

Como se observa, ambas variables quedan acotadas en el intervalo $[0, 1]$, lo que las hace compatibles con el dominio de entrada definido para el CNF. En la energía, este resultado se obtiene después de aplicar la transformación logarítmica y la normalización Min-Max, mientras que en

la variable angular se aprecia una disminución progresiva del flujo a medida que $1 - \cos(\theta)$ se aproxima a 1, tal como se indicó previamente.

En síntesis, la estrategia de muestreo estratificado, transformación y normalización de variables permitió definir un marco de preparación de datos consistente para el uso del CNF. Este esquema se aplicó tanto en el subconjunto preliminar como en el modelado final.

3.2.3. Modelado y evaluación del CNF

Una vez consolidada la etapa de preparación de los datos, se abordó la construcción del CNF empleado para representar las distribuciones energética y angular, así como la definición del marco experimental necesario para su ajuste y evaluación. A continuación, se presentan los elementos metodológicos que guiaron esta etapa del desarrollo.

3.2.3.1. Arquitectura del modelado. La implementación se realizó en Python, empleando PyTorch como entorno de entrenamiento y la librería *nflows* para el desarrollo del modelo. Estas herramientas proporcionan un marco computacional unificado para la construcción, entrenamiento e inferencia del CNF. El entrenamiento se llevó a cabo en la plataforma Kaggle, la cual ofrece acceso a recursos de cómputo acelerado, como la GPU NVIDIA Tesla T4 empleada en este trabajo.

De acuerdo con la formulación de los flujos normalizantes condicionados presentada en el marco teórico, el modelo se formuló como $p_\theta(x | c)$, donde $x = (x_E, \mu)$ representa el vector de variables objetivo y $c = (h, B_x, B_z)$ el vector de variables condicionales del sitio. En particular, x_E corresponde a z_E tras aplicar el reescalamiento Min-Max. Para modelar esta densidad se empleó un flujo construido por transformaciones autoregresivas condicionadas basadas en *splines* racionales cuadráticas monotónicas, con una variable latente definida sobre una distribución base normal estándar multivariada bidimensional

Sobre esta arquitectura se desarrolló el estudio de hiperparámetros del modelo. Esta etapa se llevó a cabo de manera iterativa, con el propósito de identificar las configuraciones que favore-

cían un ajuste estable y preciso. En este proceso se delimitó un espacio de búsqueda y se identificó que los hiperparámetros incluidos en la Tabla 4 ejercen una mayor influencia en la capacidad para modelar la estructura de la distribución objetivo.

Tabla 4

Hiperparámetros de mayor incidencia en el ajuste del modelo

Hiperparámetro	Descripción	Valores
hidden_features	Número de unidades ocultas empleadas en la red interna que parametriza cada transformación autoregresiva.	192, 256, 320
num_layers	Número de transformaciones apiladas que componen el flujo.	5, 6, 7
num_bins	Número de particiones empleadas en la <i>spline</i> racional cuadrática.	3, 4, 5
tail_bound	Límite del intervalo en el que la <i>spline</i> modela explícitamente la transformación antes del comportamiento lineal fuera de ese rango.	4, 6, 8

Nota. Para cada hiperparámetro se indica su función dentro de la arquitectura y el conjunto de valores considerado durante el estudio de ajuste. En el caso de `tail_bound`, se consideraron valores mayores a 1, dado que las transformaciones internas del flujo pueden operar fuera del intervalo de normalización $[0, 1]$, al deformar la distribución de los datos hacia el espacio de la base latente.

Aunque estos hiperparámetros inciden directamente en la capacidad de representación para modelar la distribución de la componente energética o angular, su efecto sobre el desempeño no resultó monótono dentro de los rangos evaluados. En particular, valores más altos no implicaron necesariamente un mejor ajuste, sino cambios en la estructura funcional del modelo y en su estabilidad de entrenamiento. Por esta razón, su selección no se basó en una regla creciente o decreciente, sino en la evaluación empírica de las configuraciones obtenidas durante la búsqueda.

De forma complementaria, se observó que los hiperparámetros presentados en la Tabla 5 tenían una incidencia menor dentro de los rangos examinados. En consecuencia, tras la fase iterativa inicial, se fijaron sus valores con el fin de mantener una base homogénea de comparación entre las distintas configuraciones.

Tabla 5*Hiperparámetros de menor incidencia en el ajuste del modelo*

Hiperparámetro	Descripción	Valor
optimizer	Optimizador utilizado para la actualización de los parámetros del modelo.	Adam
learning_rate	Tasa de aprendizaje que controla la magnitud de las actualizaciones de los parámetros durante el entrenamiento.	3×10^{-4}
batch_size_train	Número de muestras procesadas por iteración durante el entrenamiento.	4096
batch_size_val	Número de muestras procesadas por iteración durante la validación.	8192
gradient_clipping	Valor máximo aplicado al recorte de la norma del gradiente para estabilizar el entrenamiento.	5.0
epochs	Número de épocas utilizado en el entrenamiento.	20

Nota. Se presentan los hiperparámetros que se mantuvieron constantes en el modelo, junto con su descripción y el valor adoptado. Los valores consignados se definieron mediante pruebas preliminares iterativas y se conservaron tanto en la fase exploratoria como en el entrenamiento final.

En particular, se empleó la estimación adaptativa de momentos, (Adam, por sus siglas en inglés), como algoritmo de optimización para actualizar los parámetros del modelo durante el entrenamiento. Este optimizador adapta automáticamente la magnitud de actualización de cada parámetro a partir de estimaciones del primer y segundo momento de sus gradientes, lo que favorece una convergencia estable en esquemas de entrenamiento con minibatches Kingma y Ba, 2015. Por su parte, en validación se definió un número de muestras por iteración mayor que en entrenamiento, ya que en esta etapa solo se realiza la propagación hacia adelante para evaluar el modelo, sin cálculo ni actualización de gradientes. Los demás hiperparámetros se fijaron a partir de pruebas preliminares sucesivas, con el fin de mantener una configuración consistente tanto en la fase exploratoria como en el entrenamiento final.

3.2.3.2. Procedimiento de evaluación del modelo. Con el fin de establecer un procedimiento uniforme para el entrenamiento, seguimiento y evaluación del modelo, se definió una estrategia de validación aplicable a las distintas etapas de modelado consideradas en este trabajo.

Durante el entrenamiento se empleó la NLL como función de pérdida y como indicador del comportamiento del ajuste a lo largo de las épocas. Sin embargo, la validación del modelo no se restringió a este valor, debido a que el interés no radicaba únicamente en optimizar el ajuste global

de la densidad, sino también en preservar la forma del espectro reconstruido en todo su dominio.

Con ese propósito, la comparación entre los espectros se realizó mediante el error relativo absoluto promedio, o MARE (*Mean Absolute Relative Error*, por sus siglas en inglés), calculado de manera independiente para energía y ángulo. Esta métrica se obtuvo a partir de una comparación bin a bin entre los espectros proporcionados por ARTI y los generados por el CNF. Para realizar esta validación, las muestras generadas se llevaron a su escala original mediante las transformaciones inversas definidas previamente para la energía y el ángulo, correspondientes a las ecuaciones (3.8) y (3.10).

Si y_b denota el valor del histograma de referencia en el bin b , y $\hat{y}_b$ el valor correspondiente del histograma generado, el error relativo absoluto en cada bin se definió como

$$e_b = \left| \frac{\hat{y}_b - y_b}{y_b} \right|, \quad y_b > 0, \quad (3.11)$$

mientras que el valor global de comparación se calculó como

$$\bar{e}_{\text{rel}} = \frac{1}{B} \sum_{b=1}^B e_b, \quad (3.12)$$

donde B representa el número de bins empleados en la comparación. De esta forma, cada región del espectro contribuyó al criterio de validación, evitando que discrepancias de signo opuesto se compensaran entre sí al promediar. Con el fin de resumir el comportamiento conjunto del modelo en ambas variables, se definió además un MARE global como el promedio entre el MARE de energía y el MARE de ángulo.

El número de eventos generados con el CNF para cada ciudad se fijó en función del número de eventos producidos por ARTI. No obstante, dado el carácter estocástico del entrenamiento y de la generación de muestras, no se realizó una única ejecución. Para cada ciudad de validación se realizaron 5 corridas independientes, cada una con una semilla de inicialización distinta (valor inicial del generador pseudoaleatorio). Sobre esta base, el MARE se calculó a partir del promedio de los resultados obtenidos con el fin de reducir la variabilidad asociada a realizaciones particulares y evitar que la selección quedara condicionada por una sola corrida Henderson et al., 2018.

Con este procedimiento, la NLL quedó reservada al ajuste y seguimiento del entrenamiento, mientras que el MARE se adoptó como criterio principal evaluar el comportamiento final del modelo.

3.2.3.3. Búsqueda aleatoria de hiperparámetros. Una vez definida la arquitectura, el procedimiento de evaluación y los hiperparámetros considerados, el espacio de búsqueda delimitado dio lugar a 81 configuraciones candidatas. En consecuencia, fue necesario definir un procedimiento que permitiera identificar una combinación adecuada de hiperparámetros sin recurrir a una exploración de todas las posibilidades. Para ello, se implementó una búsqueda aleatoria de hiperparámetros, entendida como un esquema en el que las configuraciones se generan mediante muestreo aleatorio dentro de un dominio previamente acotado Bergstra y Bengio, 2012.

Bajo este esquema, se entrenaron 20 configuraciones generadas aleatoriamente. Los hiperparámetros sometidos a variación, así como aquellos que permanecieron fijos durante esta fase, fueron los ya consignados en las Tablas 4 y 5. Cada corrida se ejecutó durante 20 épocas, manteniendo un mismo marco experimental con el fin de asegurar la consistencia entre configuraciones. En este contexto, la selección del modelo se realizó a partir de la configuración que presentó el menor MARE. Los hiperparámetros correspondientes se presentan en la Tabla 6.

Tabla 6

Configuración seleccionada del modelo a partir de la búsqueda aleatoria

hidden_features	num_layers	num_bins	tail_bound
192	6	4	8

Nota. Se presenta la combinación de hiperparámetros que obtuvo el menor error relativo absoluto medio global (MARE) entre las 20 configuraciones evaluadas mediante la búsqueda aleatoria.

La configuración seleccionada indica que el desempeño no dependió de maximizar cada hiperparámetro por separado, sino del efecto conjunto que estos ejercieron sobre el ajuste. Esta etapa requirió un tiempo total de entrenamiento de 13 horas y 24 minutos.

3.2.3.4. Validación de robustez. La configuración seleccionada mediante la búsqueda aleatoria permitió fijar una arquitectura y un conjunto de hiperparámetros para el modelo. Sin embargo, en problemas de aprendizaje profundo, el desempeño observado en una configuración particular puede verse afectado por fuentes de variación inherentes al procedimiento experimental, entre ellas la inicialización aleatoria y la composición de los subconjuntos de entrenamiento y validación Henderson et al., 2018.

En este trabajo, la validación de robustez se entendió como la evaluación de la estabilidad del desempeño del modelo ante variaciones controladas en el procedimiento experimental. En particular, se analizaron cambios en la partición de ciudades entre los subconjuntos de entrenamiento y validación, así como en la semilla de inicialización, dado que estas modificaciones alteraban la cobertura del espacio de condiciones sobre el cual se ajustaba el modelo.

A partir de la configuración seleccionada en la búsqueda aleatoria, se realizaron tres corridas adicionales de robustez, con un tiempo total de 1 hora y 53 minutos. Las particiones realizadas en esta etapa se pueden observar en la Tabla 7.

Tabla 7

Particiones empleadas en la validación de robustez

Subc.	Entrenamiento	Validación	Semilla
1	Londres, Iqaluit, Chiapas, Bucaramanga, Brasilia, Malargüe, Sudbury, Pamplona, Quito, Riobamba, Berlin, La Paz, Imata, Sierra Negra, Chacaltaya	Lima, Guatemala, Casposo, Andes, Atacama	42
2	Londres, Iqaluit, Chiapas, Bucaramanga, Malargüe, Guatemala, Sudbury, Casposo, Pamplona, Berlin, La Paz, Andes, Sierra Negra, Atacama, Chacaltaya	Lima, Brasilia, Quito, Riobamba, Imata	43
3	Lima, Iqaluit, Chiapas, Brasilia, Malargüe, Guatemala, Sudbury, Pamplona, Riobamba, Casposo, Berlin, Andes, Imata, Atacama, Chacaltaya	London, Bucaramanga, Quito, Sierra Negra, La Paz	44

Nota. Se presentan las tres particiones de ciudades empleadas en la validación de robustez, definidas a partir del subconjunto preliminar. Cada una se construyó procurando conservar una cobertura diversa de altitud y campo geomagnético. En este conjunto, la altitud varió entre 24 y 5 230 m, B_x entre 9,630 y 27,532 μT , y B_z entre -14,326 y 55,928 μT . La columna Semilla indica el valor de inicialización empleado en cada ejecución.

En esta etapa no se buscaba reproducir exactamente los mismos resultados en todas las

corridas, sino verificar que el modelo mantuviera un desempeño consistente frente a variaciones en los subconjuntos y las semillas.

3.2.3.5. Modelado final. Una vez verificada, mediante la validación de robustez, la estabilidad de la configuración seleccionada, se procedió al modelado final. En esta etapa ya se disponía de la totalidad de las ciudades simuladas, cuya conformación se presenta en el Apéndice C, en la Tabla C.1. A partir de este conjunto total se definió una partición definitiva en entrenamiento, validación y prueba. Las ciudades asignadas a validación y prueba se presentan en la Tabla 8. Las restantes ciudades conformaron el subconjunto de entrenamiento.

Tabla 8

Conjunto final: ciudades incluidas en validación y prueba

No.	Ciudad	h (m)	B_x (μ T)	B_z (μ T)	Eventos
Validación					
1	Londres	24	19.542	45.044	374,028
2	Guatemala	1490	27.532	24.932	491,363
3	Riobama	2750	26.577	8.499	633,342
4	Imata	4600	22.969	-3.790	955,307
5	Atacama	5105	20.638	-8.598	1,074,499
Prueba					
1	San José	1172	27.216	19.757	467,858
2	Shiraz	1500	31.694	33.680	503,205
3	Santos Cuchumatan	3250	27.177	25.43	722,340
4	Jungfraujoeh	3454	22.038	42.626	758,580
5	Zard-kuh	4221	29.833	36.581	885,702

Nota. Se presentan las 5 ciudades asignadas al subconjunto de validación y las 5 asignadas al subconjunto de prueba en el modelado final. Para cada ciudad se reportan la altitud del sitio, h , expresada en metros (m); las componentes B_x y B_z del campo geomagnético, expresadas en microteslas (μ T); y el número de eventos de muones asociados. El subconjunto de prueba estuvo conformado por datos no utilizados durante el ajuste, la selección de hiperparámetros ni la validación de robustez.

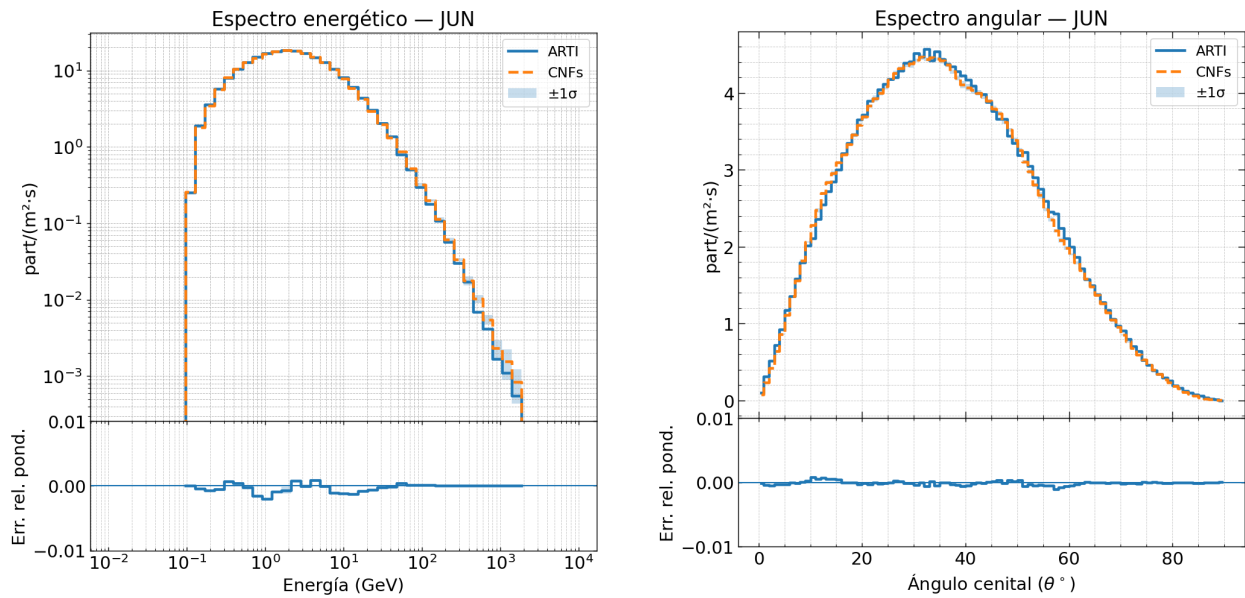
La partición definitiva se definió procurando conservar una distribución aproximada del 80 % de los datos en entrenamiento, 10 % en validación y 10 % en prueba, manteniendo al mismo tiempo diversidad en las condiciones geofísicas representadas en cada subconjunto. A partir de esta partición, el conjunto final quedó conformado por 33 804 246 eventos en entrenamiento, 3 689 844 en validación y 3 176 373 en prueba.

En fases anteriores, la cantidad de eventos generados por el CNF dependía de los datos suministrados por ARTI. En esta etapa, la cantidad de partículas a generar para cada sitio se determinó mediante la relación entre el flujo de muones y la altura, ya definida en la Ecuación 3.5. Sobre esta base, se calculó el error relativo mediante la comparación entre los espectros de referencia y los espectros generados por el modelo.

Seguidamente, el modelo entrenado se evaluó sobre los subconjuntos de validación y prueba, empleando los mismos criterios de seguimiento y comparación definidos previamente en esta metodología. La Figura 6 presenta un ejemplo de los espectros generados por el CNF para Jungfraujoch, ciudad perteneciente al subconjunto de prueba.

Figura 6.

Ejemplo de espectros generados por el CNF para Jungfraujoch.



(a) *Espectro energético*

(b) *Espectro angular*

Nota. Se presenta una salida representativa del modelo final para Jungfraujoch (JUN). Las curvas comparan los espectros obtenidos con ARTI y los generados por el CNF; la banda sombreada indica la variación de una desviación estándar ($\pm 1\sigma$) entre generaciones. El eje vertical expresa el flujo en $\text{part}/(\text{m}^2 \cdot \text{s})$, mientras que los ejes horizontales corresponden a la energía en gigaelectronvoltios (GeV) y al ángulo cenital θ en grados.

La Figura 6, ilustra el tipo de salida obtenida tras el entrenamiento del modelo, los espectros generados por el CNF reproducen de manera consistente la forma global de las distribuciones energéticas y angulares. El tiempo total de entrenamiento en esta etapa fue de 1 hora y 11 minutos.

Con ello, se dio por concluida la etapa de modelado del CNF y se estableció la base para su integración posterior en la plataforma web desarrollada en este trabajo.

3.3. Prototipo de la plataforma de consulta, generación y visualización

El desarrollo se fundamentó en un modelo de prototipado incremental, construyendo versiones funcionales refinadas mediante ciclos de retroalimentación técnica. Este enfoque permitió definir los requerimientos de interacción desde el inicio, mientras que la integración del servicio de inferencia se realizó de forma síncrona con el ajuste del modelo de inteligencia artificial. En consecuencia, el aplicativo evolucionó mediante iteraciones orientadas a estabilizar la experiencia de usuario y consolidar la arquitectura del sistema de manera progresiva.

3.3.1. Diseño e implementación de la plataforma

Se optó por una organización modular, diseñada para facilitar la trazabilidad de cada decisión tomada durante el ciclo de vida del desarrollo. Los puntos presentados a continuación exponen la evolución de la plataforma, partiendo de la base funcional y el diseño de interfaz.

3.3.1.1. Levantamiento funcional y alcance. Se definieron 3 capacidades principales del sistema. La geocodificación de ubicaciones, el procesamiento de variables geofísicas y la síntesis gráfica de datos. Esta arquitectura funcional facilitó el establecimiento de un flujo de información lineal y robusto, el cual garantizó la trazabilidad desde la captura inicial de coordenadas hasta la entrega final de los espectros, asegurando que cada componente respondiera a los requerimientos establecidos.

3.3.1.2. Planeación técnica de la solución. Primero se definió la interacción en la capa cliente y, posteriormente, se desarrolló la lógica del servidor. La Tabla 9 resume la evolución de

este proceso, en el que cada iteración permitió incorporar nuevas funciones y organizar mejor las responsabilidades entre la interfaz y los servicios de cálculo.

Tabla 9.

Iteraciones de desarrollo del prototipo de la plataforma.

Iteración	Componente principal	Descripción
1	Diseño visual del prototipo	Definición en Figma de la arquitectura de información, campos de entrada y flujo de navegación.
2	Implementación funcional inicial	Desarrollo del <i>frontend</i> en Angular y un <i>backend</i> preliminar para validar la comunicación básica entre componentes.
3	Ajuste de interacción e integración	Refinamiento de la interfaz y ajuste de las solicitudes y respuestas entre capas.
4	Ampliación funcional del sistema	Integración de servicios de descarga, obtención de coordenadas y procesamiento de parámetros de inferencia.
5	Consolidación para despliegue	Unificación de módulos y optimización de la arquitectura para la ejecución completa del flujo de simulación.

Nota. La tabla resume las cinco iteraciones seguidas en el desarrollo del prototipo, indicando el componente priorizado en cada fase y las principales actividades incorporadas en cada una.

3.3.1.3. Diseño de interfaz y especificación de interacción. A partir del alcance definido, se elaboró un prototipo visual en Figma para establecer la distribución de controles y la secuencia de navegación. Este proceso permitió validar la arquitectura de información antes de la codificación, fijando la lógica de uso y la ubicación de los elementos de visualización. Los ajustes iterativos aseguraron que la captura de parámetros y el despliegue de resultados se alinearan con las restricciones técnicas del motor de generación.

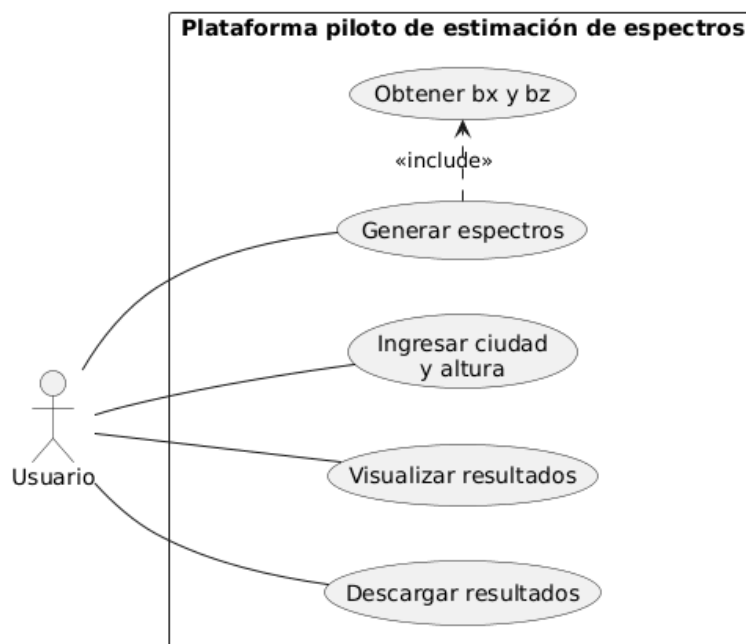
3.3.1.4. Implementación de la arquitectura web. Sobre la base del diseño validado, se implementó una arquitectura de dos capas. La capa cliente, desarrollada en Angular, gestiona la captura de parámetros y la representación de los espectros. La capa servidor, construida con FastA-

PI, centraliza la obtención de coordenadas, el cálculo de las componentes B_x y B_z y la invocación del servicio de inferencia. De modo que el sistema opera mediante un flujo de petición-respuesta basado en endpoints.

3.3.1.5. Integración funcional y verificación. La documentación del funcionamiento de la plataforma se apoyó en dos diagramas UML (*Unified Modeling Language*, por sus siglas en inglés), un lenguaje de modelado estandarizado utilizado para describir de forma visual la estructura y el comportamiento de un sistema. El primero, la figura 7, delimita las funciones disponibles para el usuario dentro del sistema, por su parte el segundo, la figura 8, representa la interacción temporal entre la interfaz, la API y los servicios de apoyo durante la ejecución del flujo principal.

Figura 7.

Diagrama de casos de uso de la plataforma piloto.



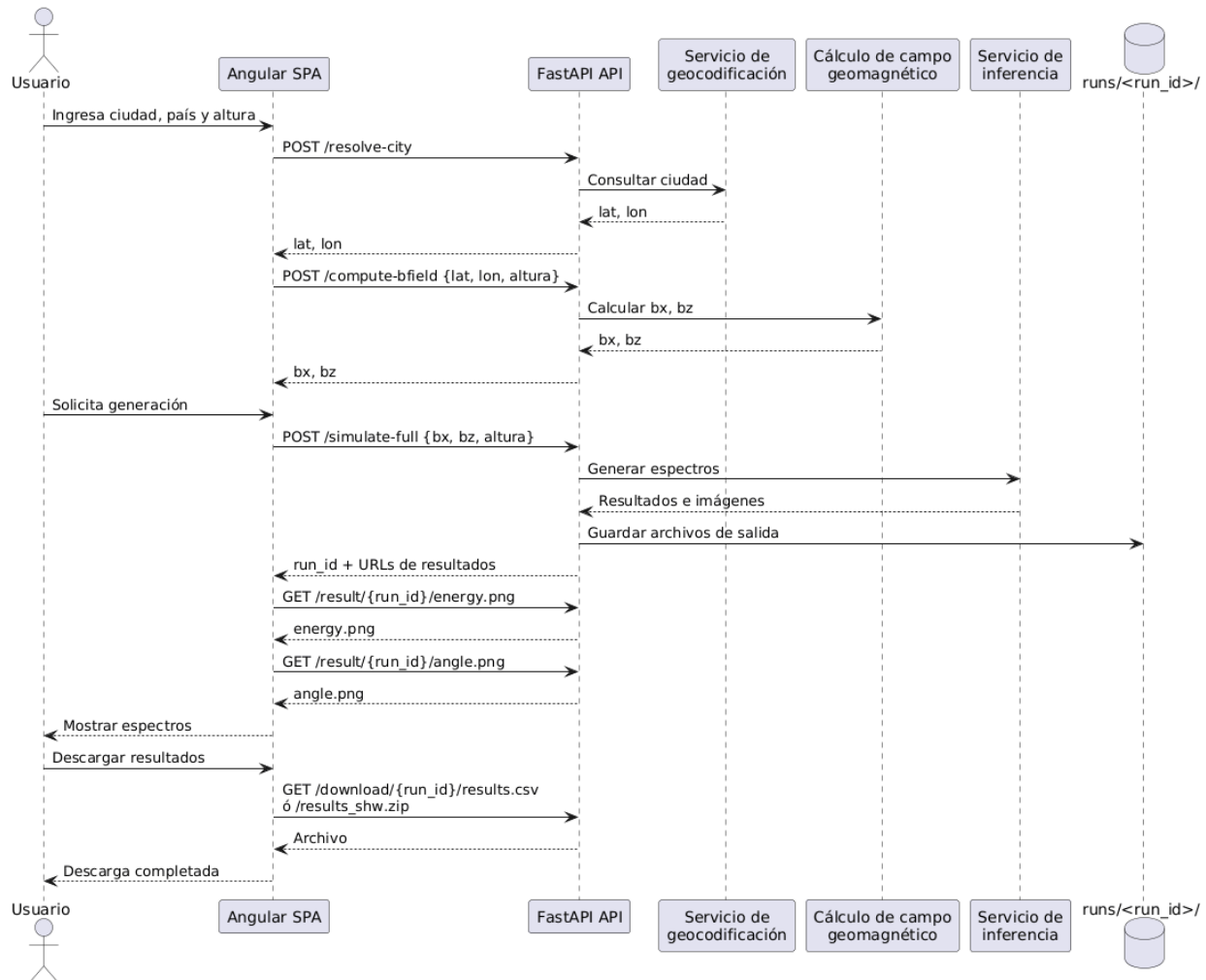
Nota. El diagrama se interpreta desde la perspectiva del usuario, representado por el actor ubicado a la izquierda. Los óvalos corresponden a las funciones que ofrece la plataforma, mientras que las líneas indican las acciones en las que participa el usuario. La relación *include* señala que, para generar espectros, primero es necesario obtener los valores de B_x y B_z .

La Figura 7 delimita el alcance funcional de la plataforma desde la perspectiva del usuario. En ella se representan las acciones principales habilitadas por el sistema, entre las que se incluyen

el ingreso de la ubicación y la altura, la obtención de los parámetros geomagnéticos, la visualización de resultados y la descarga de archivos. Su propósito es delimitar el alcance funcional de la plataforma y mostrar, desde la perspectiva del usuario, las operaciones que conforman el flujo principal de uso.

Figura 8.

Diagrama de secuencia del flujo principal de la plataforma.



Nota. El diagrama se lee de izquierda a derecha para identificar los componentes que intervienen en el proceso y de arriba hacia abajo para seguir el orden temporal de las acciones. Las flechas continuas representan solicitudes o envíos de información entre componentes, mientras que las flechas de retorno indican las respuestas generadas en cada etapa.

La Figura 8 representa la coordinación temporal entre la interfaz desarrollada en Angular, la API implementada en FastAPI y los servicios de apoyo encargados de la geocodificación, el cálculo geomagnético y la inferencia. A través de este diagrama se describe el intercambio de solicitudes

y respuestas que ocurre desde el ingreso de parámetros hasta la recuperación de resultados, lo que permite hacer explícita la lógica de interacción entre los componentes del sistema.

3.3.2. Pruebas funcionales

Para garantizar la estabilidad del sistema, se aplicaron pruebas funcionales manuales integradas en cada iteración del desarrollo. Estas revisiones comprobaron la consistencia de la navegación y la integridad del intercambio de datos vía HTTP. El plan de pruebas y la cobertura de los escenarios evaluados se presentan en la Tabla 10.

Tabla 10.

Casos de prueba funcional manual de la plataforma web.

Caso de prueba	Condición	Resultado esperado	Resultado observado
Navegación entre vistas	Acceso a secciones	Despliegue correcto de los módulos de inicio, simulador y descargas.	Exitoso
Resolución de ciudad	Ciudad y país válidos	Obtención de coordenadas geográficas para el flujo de cálculo.	Exitoso
Cálculo de B_x y B_z	Parámetros válidos	Devolución de valores geomagnéticos procesados por el <i>backend</i> .	Exitoso
Generación de espectros	Datos de sitio válidos	Ejecución de la inferencia y renderizado de la salida gráfica.	Exitoso
Visualización de resultados	Simulación finalizada	Actualización dinámica de la interfaz con los resultados generados.	Exitoso
Descarga de archivos	Selección de formato	Entrega de archivos CSV o SHW con la integridad de datos preservada.	Exitoso
Validación de entradas	Parámetros nulos	Notificación de error y bloqueo de la ejecución del flujo.	Exitoso
Recurso inexistente	Consulta de resultado nulo	Respuesta de estado HTTP coherente con la ausencia del recurso.	Exitoso

Nota. Cada fila corresponde a una prueba aplicada sobre una funcionalidad específica de la plataforma. La columna Condición indica el escenario evaluado, Resultado esperado describe el comportamiento previsto del sistema y Resultado observado registra el comportamiento efectivamente obtenido durante la verificación.

3.3.3. Estrategia de despliegue web e integración

El despliegue se planteó mediante una arquitectura desacoplada, en la que el cliente y el servidor se publicaron en servicios independientes. Esta decisión permitió separar la entrega de la interfaz, orientada a recursos estáticos, de la ejecución de la API, encargada del procesamiento geofísico y la inferencia del modelo. La selección de Vercel y Hugging Face Spaces respondió al alcance de prototipo de la plataforma, dado que ambos servicios permitían realizar el despliegue en modalidades gratuitas, sin incorporar infraestructura propia durante esta etapa del desarrollo.

3.3.3.1. Configuración y publicación del cliente. El *frontend* desarrollado en Angular se compiló en modo producción como una aplicación de página única. Para su publicación se seleccionó Vercel⁴, debido a su compatibilidad con aplicaciones estáticas y su integración directa con repositorios de código. Bajo este esquema, la versión publicada del cliente se obtiene a partir del proceso de construcción definido para la aplicación.

3.3.3.2. Contenerización y ejecución del servidor. El *backend* desarrollado en FastAPI se desplegó en Hugging Face Spaces⁵ mediante contenedores Docker. Esta estrategia se adoptó para disponer de un entorno reproducible para la API y para el servicio de inferencia, manteniendo separada la configuración del servidor respecto al cliente.

La integración entre ambas capas se definió mediante solicitudes HTTP con intercambio de datos en formato JSON. Además, se contempló el uso de variables de entorno para separar credenciales y parámetros de ejecución del código fuente, junto con una política CORS limitada al

⁴Vercel: <https://vercel.com/>

⁵Hugging Face: <https://huggingface.co/>

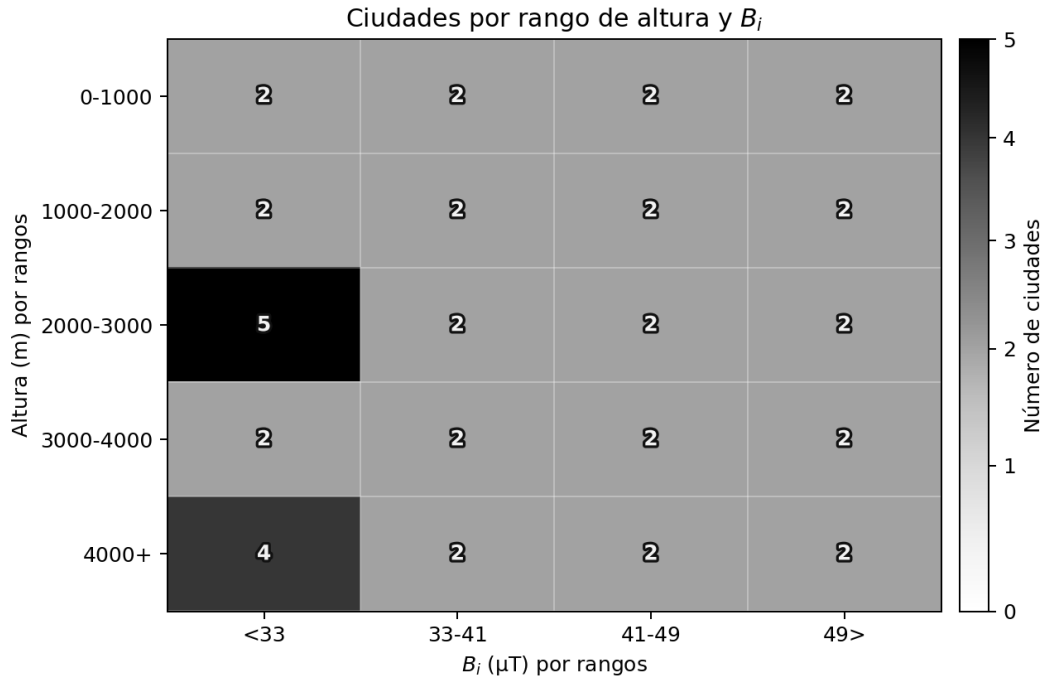
dominio del cliente desplegado.

4. Resultados

4.1. Conjunto de datos final

Se consolidó un conjunto de simulaciones balanceado correspondiente a 45 ubicaciones realizadas en ARTI, cuyos parámetros geográficos y geomagnéticos se presentan en el Apéndice C, en la Tabla C.1. En conjunto, los sitios seleccionados abarcaron 25 países distribuidos en cinco continentes, con alturas comprendidas entre 0 y 5230 m s. n. m. e intensidades geomagnéticas B_i entre 22.5 y 56.75 μT . En los archivos procesados, el número de muones registrados por sitio presentó valores que oscilaron entre 371 352 eventos y 1 118 185 eventos por ciudad. Además, debido al tamaño de los archivos, en términos de almacenamiento, el volumen total de las simulaciones descomprimidas fue de aproximadamente 41 GB.

La distribución final de ciudades sobre el esquema de muestreo estratificado quedó como se muestra en la Figura 9. Esta configuración representa el resultado acumulado de las simulaciones realizadas a lo largo del proyecto, una vez cubiertos los intervalos de altura y b_i definidos como objetivo. Además, en algunas casillas se incorporaron ciudades adicionales por curiosidades de experimentación surgidas durante el desarrollo del trabajo, lo que dio lugar a una cobertura no uniforme entre todos los estratos del conjunto, pero para el desarrollo del proyecto mas adelante se siguieron las reglas del muestreo.

Figura 9.*Muestreo final con todas las ciudades simuladas.*

Nota. La figura organiza las ciudades simuladas según el esquema de muestreo estratificado definido para la altura y la intensidad geomagnética B_i . Cada celda representa una combinación de rangos, y el valor mostrado en su interior corresponde al número de ciudades ubicadas en ese estrato.

4.2. Estimación del espectro mediante CNF**4.2.1. Búsqueda de hiperparámetros**

La Tabla 11 presenta las 5 configuraciones con menor MARE global obtenidas durante la búsqueda aleatoria, en la cual se evaluaron 20 combinaciones de hiperparámetros sobre el conjunto de validación de la fase exploratoria. Para cada configuración se reportan el MARE de energía, el MARE de ángulo y la NLL final, métricas definidas en la metodología para la evaluación del modelo, donde el MARE corresponde al error relativo absoluto medio y la NLL a la log-verosimilitud negativa.

Tabla 11*Cinco configuraciones con menor MARE global obtenidas en la búsqueda aleatoria*

Conf.	h_features	n_layers	n_bins	t_bound	NLL final	MARE energía	MARE ángulo	MARE global
2	192	6	4	8	-1,3317	0,1235	0,1096	0,1166
4	320	7	5	6	-1,3321	0,1714	0,1114	0,1414
15	320	5	3	4	-1,3360	0,1258	0,1963	0,1611
13	320	5	4	4	-1,3318	0,2141	0,1670	0,1905
17	256	5	3	6	-1,3281	0,1670	0,2187	0,1929

Nota. El error relativo absoluto medio (MARE) de energía y ángulo corresponden al promedio obtenido sobre las cinco ciudades que conforman el subconjunto de validación. El MARE global representa el promedio entre el MARE de energía y ángulo. h_features, n_layers, n_bins y t_bound representan hidden_features, num_layers, num_bins y tail_bound, respectivamente. La NLL (log-verosimilitud negativa) final corresponde a la última pérdida en validación. El MARE global se calculó como el promedio entre el MARE de energía y el MARE de ángulo.

La configuración 2 presentó el menor MARE global y fue seleccionada como configuración principal. Las cuatro configuraciones siguientes exhiben valores entre 0,1414 y 0,1929, lo que representa incrementos absolutos entre 0,0248 y 0,0763 respecto a la configuración 2. La NLL final, en contraste, exhibió una separación de apenas 0,008 unidades entre las cinco configuraciones, con valores comprendidos entre -1,3281 y -1,3360.

EL comportamiento de esta metrica se presenta debido a que promedia globalmente el ajuste de la densidad, de modo que las discrepancias en regiones con menor representación espectral tienden a atenuarse en el resultado final. Dichas características respaldan la decisión metodológica de adoptar el MARE como criterio principal de selección, dado que su cálculo bin a bin preserva la información local del espectro.

Asimismo, la configuración seleccionada no corresponde a la combinación con mayor capacidad de representación en términos absolutos, dado que hidden_features = 192 es el valor más reducido dentro del espacio de búsqueda explorado. Sin embargo, su desempeño superior frente a configuraciones con redes más anchas, como las configuraciones 4 y 13, es consistente con el carácter no monótono del espacio de hiperparámetros señalado en la metodología. En modelos de aprendizaje profundo, incrementar la capacidad de la red no garantiza una mejor reconstrucción espectral; por el contrario, una red más ancha puede requerir mayor tiempo de entrenamiento para converger sin asegurar un mejor comportamiento.

A continuación, la Tabla 12 reporta el MARE de energía y ángulo por ciudad para la configuración seleccionada, con el fin de examinar el comportamiento del modelo en cada sitio de validación de forma individual.

Tabla 12

MARE de energía y ángulo por ciudad de validación para la configuración 2

Ciudad	MARE energía	MARE ángulo
Lima	0,1763	0,1165
Guatemala	0,0902	0,0943
Quito	0,1230	0,1052
Imata	0,1068	0,0754
Atacama	0,1211	0,1567

Nota. Se reporta, para cada ciudad de validación, el error relativo absoluto medio (MARE) asociado a las componentes de energía y ángulo. El MARE de energía presentó valores entre 0,0902 y 0,1763, mientras que el MARE de ángulo osciló entre 0,0754 y 0,1567. En tres de las cinco ciudades, la componente angular registró un error menor que la energética.

La variabilidad del MARE entre ciudades de validación refleja, en parte, las diferencias en las condiciones de sitio representadas por cada ubicación. Lima y Atacama constituyen sitios cuyas condiciones se encuentran en los dos extremos del espacio de condicionamiento, lo que dificulta la interpolación del modelo en esa región.

En conjunto, la búsqueda aleatoria permitió identificar la configuración con el mejor desempeño. Este resultado se obtuvo sin necesidad de recorrer exhaustivamente las 81 combinaciones posibles del espacio de búsqueda, lo que respalda la utilidad del esquema aleatorio en esta fase exploratoria.

4.2.2. Validación de robustez

La Tabla 13 presenta los resultados de las tres corridas adicionales realizadas sobre la configuración 2, con el fin de verificar su estabilidad ante variaciones controladas en la partición de ciudades y en la semilla de inicialización.

Tabla 13*Resultados de las tres corridas de validación de robustez*

Corrida	NLL final	MARE energía	MARE ángulo	MARE global
00	-1,2809	0,1161	0,1792	0,1477
01	-1,2545	0,1693	0,2288	0,1991
02	-1,2850	0,1401	0,1667	0,1534

Nota. Se reportan tres corridas adicionales de la configuración 2, evaluadas con particiones alternativas de ciudades y semillas de inicialización distintas. La NLL final, entendida como la log-verosimilitud negativa al finalizar el entrenamiento, varió entre -1,2850 y -1,2545. Por su parte, el error relativo absoluto medio (MARE) de energía y ángulo corresponden al promedio obtenido sobre las cinco ciudades que conforman el subconjunto de validación. El MARE global representa el promedio entre el MARE de energía y ángulo. Esta validación no buscaba reproducir el mismo resultado en todas las corridas, sino verificar que el modelo conservara un desempeño consistente ante cambios controlados.

Las corridas 00 y 02 presentaron MARE globales similares entre sí, mientras que la corrida 01 registró el mayor MARE global, impulsado principalmente por la componente angular. En las tres corridas, el MARE de energía resultó inferior al MARE de ángulo.

Cada corrida implicó una redistribución completa de los conjuntos de entrenamiento y validación, con ciudades distintas en cada subconjunto. Bajo esta condición, el rango de MARE global entre 0,1477 y 0,1991 refleja la sensibilidad esperable ante cambios en la cobertura del espacio de condiciones representado durante el entrenamiento, y no una inestabilidad estructural del modelo. Las corridas 00 y 02, cuyas particiones distribuyeron de forma más equilibrada los sitios extremos entre ambos subconjuntos, exhibieron errores más reducidos, lo que indica que la configuración mantiene un ajuste estable siempre que el conjunto de entrenamiento preserve una cobertura adecuada del espacio condicional. Sobre esta base, la configuración se consideró suficientemente estable para proceder al modelado final.

4.2.3. Modelado final

Una vez seleccionada y validada la configuración con mejor desempeño, se procedió a su modelado final sobre la partición definitiva de entrenamiento, validación y prueba. La Tabla 14 presenta los resultados obtenidos en esta etapa.

Tabla 14*Resultados globales del modelado final sobre los conjuntos de validación y prueba*

Conjunto	MARE energía	MARE ángulo	MARE global
Validación	0,1378	0,0943	0,1160
Prueba	0,1109	0,0740	0,0925

Nota. Se presentan los resultados globales para los conjuntos de validación y prueba; en cada caso, el valor reportado corresponde al promedio obtenido sobre las cinco ciudades que conforman el subconjunto. El error relativo absoluto medio (MARE) se reporta para las componentes de energía y ángulo, y su valor global se obtuvo como el promedio entre ambas.

En la componente energética, el conjunto de prueba presentó un MARE muy similar al observado en validación. La diferencia principal se concentró en la componente angular, cuyo MARE disminuyó de 0,0943 a 0,0740. En consecuencia, el MARE global obtenido en prueba resultó inferior al de validación.

Este comportamiento puede atribuirse a que ambos subconjuntos fueron construidos a partir de sitios distintos, de modo que la variación observada responde a la composición específica de las ciudades incluidas en cada partición. En este sentido, los resultados indican que la configuración final mantiene un comportamiento estable frente a nuevas combinaciones de altura y condiciones geomagnéticas.

Con el fin de identificar con mayor precisión las regiones del espectro en las que se concentran las principales discrepancias, en la Tabla 15 se presenta el MARE para diferentes intervalos de energía y ángulo.

Tabla 15*MARE por intervalos de energía y ángulo en los conjuntos de validación y prueba*

Conjunto	$[10^{-1}, 10^2]$ GeV	$[10^2, 10^3]$ GeV	$[10^3, 10^4]$ GeV	$[0^\circ, 30^\circ]$ θ	$[30^\circ, 60^\circ]$ θ	$[60^\circ, 90^\circ]$ θ
Validación	0,0694	0,2465	0,5577	0,0765	0,0304	0,1759
Prueba	0,0387	0,2244	0,6283	0,0587	0,0214	0,1419

Nota. Se presenta el error relativo absoluto medio (MARE) calculado en tres intervalos de energía y tres intervalos de ángulo. En los encabezados, los valores entre corchetes indican el intervalo considerado en cada caso; las tres primeras columnas corresponden a energía, expresada en GeV (gigaelectronvoltios), y las tres últimas a ángulo, expresado en grados. Los intervalos se definieron para comparar el comportamiento del error en regiones del espectro con distinta densidad de eventos.

En la componente energética, el MARE más bajo se registró en el intervalo de 10^{-1} a 10^2

GeV, mientras que el valor más alto se observó entre 10^3 y 10^4 GeV. Este comportamiento indica que el modelo reproduce con mayor precisión la región central del espectro energético y concentra sus principales discrepancias en el intervalo de energías más altas, donde la cantidad de eventos es menor.

Este comportamiento es consistente con limitaciones reportadas para los CNFs, cuya capacidad de ajuste suele concentrarse en el cuerpo de la distribución, mientras que la representación de las regiones de baja densidad resulta más exigente Laszkiewicz et al., 2022. En este caso, diferencias absolutas pequeñas entre el espectro generado y el de referencia se traducen en errores relativos más altos. Por tanto, el aumento del MARE en 10^3 a 10^4 GeV no contradice el ajuste global del modelo, sino que delimita la zona del espectro cuya reconstrucción resulta más exigente para el CNF.

Para la componente angular, el MARE más bajo se ubicó entre 30° y 60° , mientras que los valores más altos se concentraron en los intervalos 0° a 30° y 60° a 90° . Esto muestra que el modelo describe con mayor precisión la región central de la distribución angular y presenta mayores discrepancias hacia los extremos del intervalo analizado. No obstante, el error en 0° a 30° se mantuvo por debajo del observado en 60° a 90° , lo que sugiere que la transformación μ , descrita en la metodología para la transformación y normalización de variables, favoreció una representación más estable en la zona cercana a 0° . Al redistribuir su comportamiento de forma decreciente, esta transformación evitó que la escasez de datos se reflejara con la misma intensidad en ambos extremos, como habría ocurrido con una transformación que preservara de forma más directa la forma original de θ .

Adicionalmente, se evaluó la precisión del modelo en términos del número total de partículas generadas. De acuerdo con lo descrito en la metodología para el modelado final, la cantidad de eventos a generar por el CNF se estimó mediante la relación entre el flujo de muones y la altura, definida en la Ecuación 3.5. La Tabla 16 presenta la comparación entre los eventos de ARTI y los eventos generados por el modelo.

Tabla 16*Precisión en el número de eventos generados por el CNF*

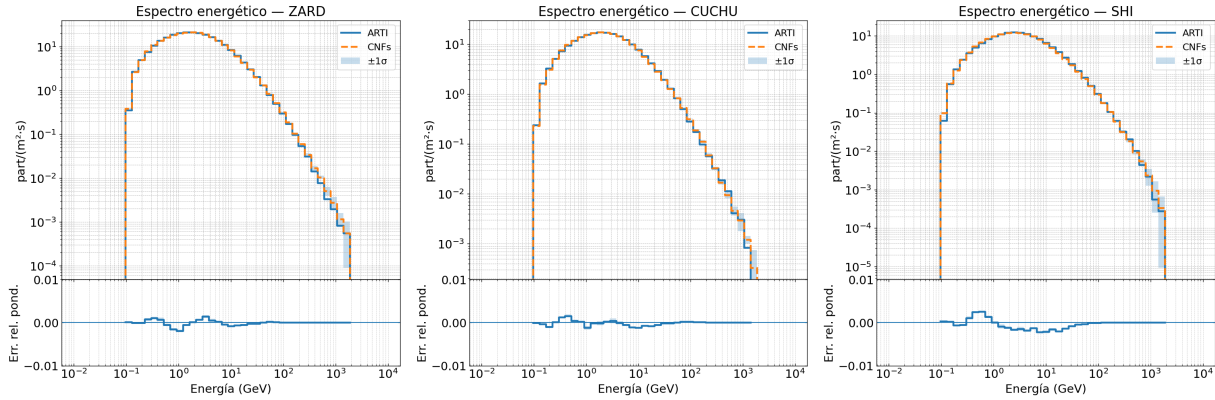
Conjunto	Eventos ARTI	Eventos CNF	Error eventos	Tiempo (s)
Validación	3 528 531	3 539 070	0,0145	124,65
Prueba	3 337 677	3 324 987	0,0043	117,52

Nota. Los eventos ARTI corresponden al número de partículas de referencia en cada subconjunto, mientras que los eventos CNF indican la cantidad de partículas generadas por el modelo a partir de la fórmula definida para el modelado final. El error de eventos expresa la discrepancia relativa entre ambas cantidades. El tiempo se reporta en segundos (s).

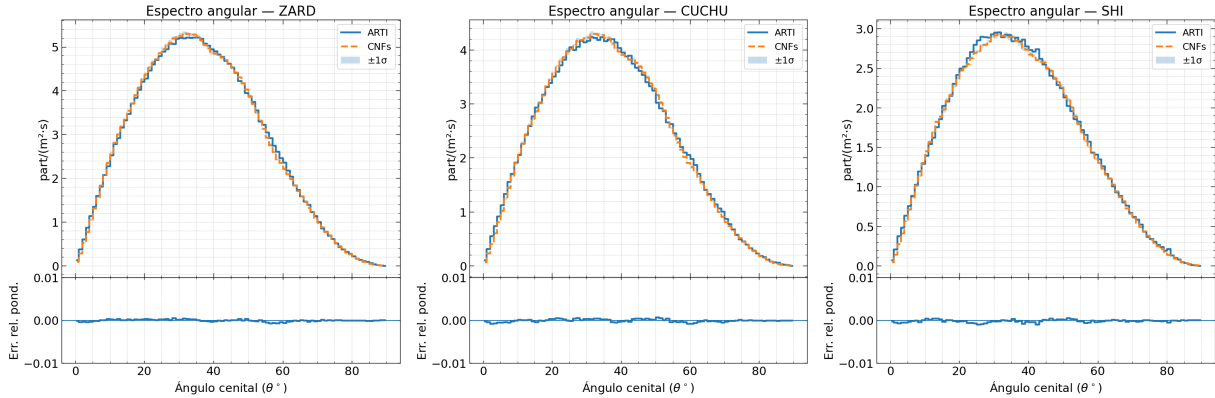
El número de eventos generado por el CNF fue cercano al número de eventos de referencia obtenido con ARTI. Para cada subconjunto se calculó el error relativo por ciudad y posteriormente se promedió, obteniendo un error global de 0,0145 en validación y de 0,0043 en prueba. Estos valores corresponden a precisiones aproximadas de 98,55 % y 99,57 %, respectivamente.

Con esta precisión, el modelo generó los subconjuntos evaluados en 124,65 s para validación y 117,52 s para prueba, empleando una GPU NVIDIA Tesla T4 en Kaggle. En comparación con el proceso de simulación de ARTI, este resultado evidencia una reducción en el tiempo de generación, al pasar de escalas de horas a segundos. A nivel de ciudad, los errores mantienen una escala similar, los resultados se presenta en el Apéndice D.

Las Figuras 10 y 11 presentan, para un grupo de ciudades del conjunto de prueba, la comparación entre los espectros de referencia obtenidos con ARTI y los espectros generados por el CNF en las componentes energética y angular. Estas figuras complementan la información detallada en la Tabla 15 y permiten examinar localmente el ajuste del modelo a lo largo del dominio. Los resultados completos para las ciudades de validación y prueba se presentan en los Apéndices E y F, respectivamente.

Figura 10.*Reconstrucción del espectro energético en Zard-Kuh, Santos Cuchumatán y Shiraz*

Nota. Se compara el espectro de energía de referencia proporcionado por ARTI con el espectro generado por el CNF en tres ciudades. En el panel superior de cada gráfica, la línea azul continua representa el espectro de referencia y la línea naranja discontinua el espectro medio generado por el modelo; la banda sombreada indica la desviación estándar ($\pm 1\sigma$) entre corridas. En el panel inferior se presenta el aporte del error relativo ponderado por intervalo de energía. La energía se expresa en GeV (gigaelectronvoltios) y el eje vertical del panel superior en partículas por metro cuadrado y por segundo.

Figura 11.*Reconstrucción del espectro angular en Zard-Kuh, Santos Cuchumatán y Shiraz*

Nota. Se compara el espectro angular de referencia proporcionado por ARTI con el espectro generado por el CNF en tres ciudades. En el panel superior de cada gráfica, la línea azul continua representa el espectro de referencia y la línea naranja discontinua el espectro medio generado por el modelo; la banda sombreada indica la desviación estándar ($\pm 1\sigma$) entre corridas. En el panel inferior se presenta el aporte del error relativo ponderado por intervalo angular (aporte err.). El ángulo cenital θ se expresa en grados y el eje vertical del panel superior en partículas por metro cuadrado y por segundo.

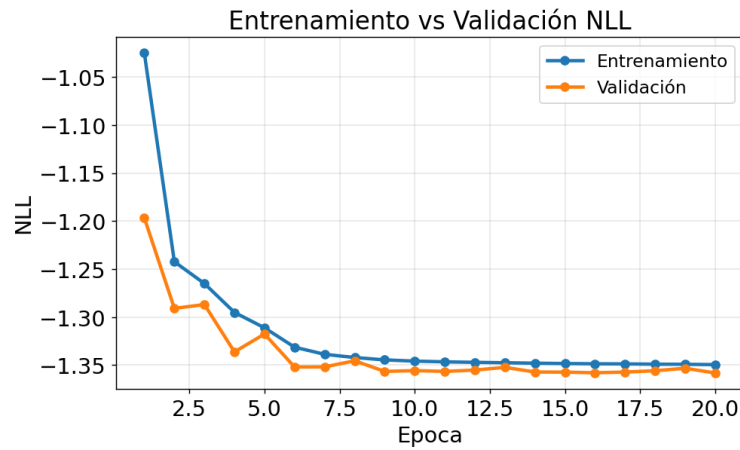
En cada figura, el panel superior superpone los espectros de ARTI y del CNF, mientras que el panel inferior muestra la curva del error relativo ponderado, acompañada de una banda de variación $\pm 1\sigma$. La curva del error relativo ponderado se empleó como complemento del MARE

bin a bin, ya que reduce la sobredimensión visual de las regiones con pocos eventos y resalta los desajustes con mayor influencia sobre la forma global del espectro. En este trabajo, dicho indicador puede tomar valores aproximados en la escala $[-1, 1]$, donde cero representa coincidencia entre el espectro generado por el CNF y la referencia de ARTI. Los resultados se mantuvieron principalmente en el orden de 10^{-3} , lo que indica una baja discrepancia entre ambos espectros.

La banda $\pm 1\sigma$ permite interpretar las diferencias observadas considerando la naturaleza estocástica de las simulaciones Monte Carlo, donde ejecuciones independientes bajo las mismas condiciones pueden producir variaciones ligeras en los espectros Heck et al., 2025.

En este contexto, las discrepancias localizadas en regiones de baja densidad no debe interpretarse de forma aislada como una falla del modelo, ya que en esas regiones la propia referencia de ARTI puede presentar variaciones apreciables entre realizaciones. Desde esta perspectiva, el hecho de que los espectros generados por el CNF permanezcan dentro la banda $\pm 1\sigma$ en buena parte del dominio respalda que el modelo captura la estructura global de la distribución y reproduce una variabilidad compatible con la incertidumbre estadística inherente al problema.

Para examinar la evolución del ajuste durante el entrenamiento del modelo final, en la Figura 12 se presenta el comportamiento de la NLL en los conjuntos de entrenamiento y validación a lo largo de las 20 épocas consideradas.

Figura 12.*Evolución de la NLL en los conjuntos de entrenamiento y validación*

Nota. Se presenta la evolución de la log-verosimilitud negativa (NLL) durante las 20 épocas de entrenamiento. La curva azul corresponde al conjunto de entrenamiento y la curva naranja al conjunto de validación. En ambas se observa un descenso pronunciado en las primeras épocas, seguido de una estabilización progresiva.

Se observa que ambas curvas descienden de forma pronunciada en las primeras épocas y, posteriormente, tienden a estabilizarse. A partir de la época 8, los valores de entrenamiento y validación permanecen muy próximos entre sí, con oscilaciones pequeñas y sin separaciones sostenidas entre ambas trayectorias. Este comportamiento sugiere que el proceso de optimización alcanzó un régimen estable y que, dentro del horizonte de entrenamiento considerado, no se evidencian señales claras de sobreajuste.

Con base en los resultados obtenidos, puede afirmarse que los CNFs permiten estimar el espectro de muones con una precisión elevada frente a las simulaciones de ARTI, al reproducir de manera consistente la estructura global de las distribuciones en los sitios evaluados.

A partir de este punto, resulta pertinente complementar el análisis de precisión con el costo computacional asociado al uso del modelo. En particular, se examina la relación entre el número de partículas a generar y el tiempo de inferencia requerido, con el fin de contrastar la eficiencia práctica del CNF frente al costo de las simulaciones de ARTI.

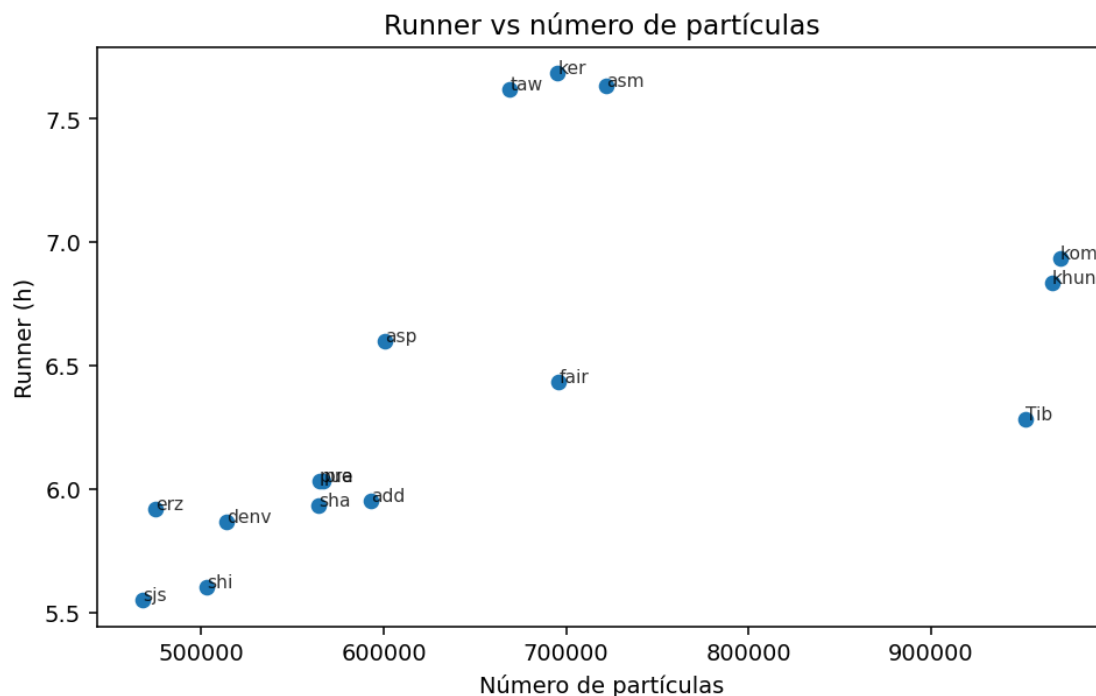
4.3. Costo computacional de la estimación

En base a los registros de tiempo de las simulaciones, la etapa *runner* presentó duraciones entre 5 h y 8 h 57 m. Al añadir el tiempo de conversión de binarios, con una duración promedio de 140 min por simulación, el tiempo total por ciudad se ubicó entre 7 h 20 m y 11 h 17 m. A partir de un promedio estadístico, el costo medio se estimó en 8 h 50 m por sitio. Extendido a las 45 ubicaciones del conjunto, esto representa cerca de 396 h de tiempo simulado, equivalentes a 16.5 días.

Con el fin de examinar la relación entre la duración de la etapa *runner* y el volumen de partículas registradas en cada simulación, se construyó un diagrama de dispersión para las 16 ciudades con información temporal disponible, como se ve en la figura 13.

Figura 13.

*Relación entre el tiempo de la etapa *runner* y el número de partículas registradas.*



Nota. Cada punto representa una ciudad y su etiqueta corresponde a la sigla del sitio simulado. El eje horizontal indica el número de partículas registradas, mientras que el eje vertical muestra la duración de la etapa *runner*, expresada en horas.

La Figura 13 examina la relación entre la duración de la etapa *runner* y el número de

partículas registradas en la simulación de referencia. Se aprecia una tendencia creciente, aunque con dispersión entre ciudades: las corridas que producen más partículas tienden a requerir más tiempo, pero el tamaño de la salida no explica por sí solo toda la variabilidad observada. Este comportamiento es consistente con el hecho de que una simulación con mayor número de eventos exige más procesamiento y más operaciones. La figura permite considerar el volumen de partículas como un factor directamente asociado al costo de la simulación.

Tomando el tiempo de generación observado en la plataforma web como indicador del costo temporal de consulta, la inferencia del modelo final operó en una escala de segundos. En las cuatro ciudades del conjunto de prueba consideradas, el tiempo total acumulado en ARTI fue de 35 h 48 min, mientras que el tiempo acumulado promedio en la página web fue de 224.60 s, como se aprecia en la tabla 17.

Tabla 17.

Comparación entre simulación de ARTI y generación en la plataforma web.

Conjunto	Ciudades	Tiempo total	ARTI (s)	Tiempo web total	Aceleración estimada*	Reducción temporal
Prueba	4	35 h 48 min	224.60		573.8×	99.83 %

Nota. La tabla compara, para cuatro ciudades del conjunto de prueba, el tiempo total de simulación requerido por ARTI frente al tiempo total promedio observado en la plataforma web. El tiempo web corresponde al intervalo desde la solicitud de generación hasta la visualización de los resultados en la interfaz. La aceleración estimada corresponde a la razón entre el tiempo total de ARTI, convertido a segundos, y el tiempo web total.

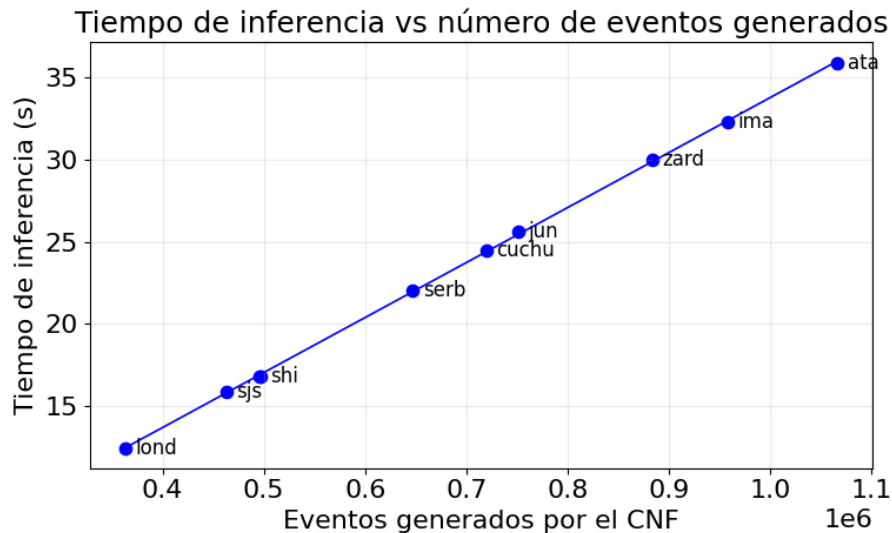
La Tabla 17 resume la comparación entre la simulación de referencia con ARTI y la generación realizada mediante el CNF en la plataforma web. La aceleración estimada fue de 573.8× en el conjunto de prueba, calculada con respecto al tiempo acumulado de simulación de ARTI para las cuatro ciudades evaluadas. Estos resultados evidencian una reducción del costo computacional superior a dos órdenes de magnitud a favor del modelo generativo. En términos prácticos, el tiempo requerido para producir el conjunto completo de ciudades, que mediante simulación directa tardó casi 17 días de cómputo, puede reproducirse con el modelo en menos de una hora.

Con el fin de caracterizar el costo de consulta del modelo una vez finalizado el entrenamiento, se analizó la relación entre el número de eventos generados por el CNF y el tiempo requerido

para producirlos. Para ello, se construyó un diagrama sobre diez sitios de evaluación, donde se aplica un ajuste lineal que permite resumir la tendencia observada entre ambas variables, como se observa en la figura 14.

Figura 14.

Tiempo de inferencia del CNF en función del número de eventos generados.



Nota. Cada punto representa un sitio evaluado y su etiqueta corresponde a la sigla de la ciudad. El eje horizontal, con escala de 10^6 , indica el número de eventos generados por el CNF, mientras que el eje vertical muestra el tiempo de inferencia, expresado en segundos. La línea azul corresponde a un ajuste lineal.

La Figura 14 muestra que el tiempo de inferencia crece con el número de eventos generados. Un ajuste lineal sobre los diez sitios evaluados produce la relación $t \approx 3.35 \times 10^{-5}N + 0.27$, con $R^2 \approx 0.9999$, donde t corresponde al tiempo en segundos y N al número de eventos generados. Esto indica que, una vez entrenado, el costo de consulta queda determinado principalmente por el tamaño de la muestra solicitada y no por cambios abruptos de condiciones entre sitios. Esta regularidad resulta útil porque permite anticipar el tiempo de generación a partir de una cantidad físicamente interpretable. En particular, si el número de eventos por sitio se fija a partir del flujo esperado y este se aproxima mediante la fórmula del flujo esperado a una altura dada 3.5. Entonces la altura puede emplearse para estimar indirectamente el tiempo de consulta dentro del intervalo cubierto por las simulaciones.

La ecuación no predice el tiempo de forma directa, pero sí permite encadenar altura, flujo

esperado, número de eventos y costo de inferencia. En los sitios evaluados del conjunto de prueba, el tiempo de generación en la plataforma web se mantuvo en una escala de segundos, como se muestra en la tabla 18.

Tabla 18.

Desempeño temporal del CNF por ciudad en la plataforma web.

Ciudad	Tiempo ARTI	Web1 (s)	Web2 (s)	Web3 (s)	Promedio (s)	Factor de mejora	Reducción
San Jose	7 h 41 min	40.60	46.08	42.03	42.90	644.7×	99.84 %
Shiraz	8 h 26 min	48.00	52.00	47.40	48.00	632.5×	99.84 %
Santos Cuchumatan	8 h 22 min	68.00	65.00	62.70	65.23	461.7×	99.78 %
Jungfraujoeh	11 h 19 min	70.50	67.00	67.90	68.47	595.0×	99.83 %

Nota. Cada fila corresponde a una ciudad del conjunto de prueba y resume la comparación entre el tiempo de simulación requerido por ARTI y el tiempo observado en la plataforma web. La columna Web prom. corresponde al promedio de las ejecuciones realizadas para cada ciudad. En Shiraz se reportó una única ejecución. La aceleración representa la razón entre el tiempo de simulación con ARTI, convertido a segundos, y el tiempo promedio registrado en la plataforma web.

La Tabla 18 muestra los datos de las 4 ciudades evaluadas del conjunto de prueba. El tiempo observado en la plataforma web osciló entre 42.90 s y 68.47 s, mientras que el tiempo de simulación en ARTI se ubicó entre 7 h 41 min y 11 h 19 min. La aceleración estimada frente al tiempo de ARTI se mantuvo entre 461.7× y 644.7×, incluso al considerar el tiempo completo percibido por el usuario durante la generación y visualización de los resultados.

Finalmente se aprecia en las tablas 17 y 18 que la plataforma permite consultar espectros generados por el modelo en una escala de segundos, mientras que la simulación directa con ARTI requiere varias horas por ciudad. De manera que el despliegue web permite materializar la reducción del costo computacional en una experiencia de consulta interactiva.

4.4. Prototipo web

Se obtuvo un prototipo web capaz de ejecutar el flujo principal de consulta de un sitio, cálculo de parámetros de entrada y generación de resultados asociados al modelo. El resultado de esta etapa se refleja en la operación del flujo completo dentro de la interfaz, desde el ingreso de la

ubicación hasta la recuperación de salidas gráficas y archivos exportables, como se ve en la tabla 19.

Tabla 19.

Servicios integrados en el flujo principal de la plataforma piloto.

Etapas del flujo	Servicio o recurso	Entrada principal	Salida devuelta	Evidencia en interfaz
Busqueda de ubicación	Servicio de geocodificación	Ciudad y país	Latitud, longitud y nombre resuelto	Autocompletado de coordenadas y confirmación de la ubicación
Procesamiento geomagnético	Cálculo de B_x y B_z	Latitud, longitud y altura	Componentes geomagnéticas del sitio	Actualización de los campos B_x y B_z antes de simular
Generación de resultados	Servicio de inferencia	B_x , B_z y altura	Ejecución del modelo y creación de la corrida	Cambio de estado del simulador y activación del resultado
Recuperación de imágenes	Recursos de salida de la corrida	Identificador de ejecución	Gráficas de energía y ángulo	Despliegue de resultados en pantalla
Exportación de archivos	Descarga de resultados	Identificador de ejecución y formato	Archivo CSV o SHW comprimido	Descarga del producto generado

Nota. Cada fila resume una etapa del flujo principal, el servicio o recurso involucrado, la información de entrada, la salida generada y su evidencia correspondiente en la interfaz.

La Tabla 19 resume el flujo desde la lógica de operación del sistema. La consulta del usuario pasa primero por una etapa de resolución y cálculo, y posteriormente se convierte en una solicitud de inferencia con salidas visuales y descargables. De modo que se muestra qué funciones quedaron integradas en el prototipo y cómo se conectan dentro de una misma ejecución.

Con el fin de sintetizar la estructura visible del prototipo, la Tabla 20 presenta los módulos principales de la interfaz y la función que cumple cada uno dentro del flujo de uso.

Tabla 20.*Componentes visibles del prototipo y función dentro del flujo de uso.*

Módulo o vista	Función dentro del flujo	Resultado observable
Inicio	Presentar el propósito general del aplicativo y el acceso al simulador	Entrada al flujo principal de consulta
Simulador	Concentrar la operación principal de resolución, cálculo, generación y descarga	Ejecución completa del proceso y visualización de resultados
Descripción	Presentar los parámetros de entrada y las salidas generadas	Apoyo a la interpretación de la consulta

Nota. Cada fila corresponde a un módulo visible del prototipo y resume su función dentro del flujo de uso, junto con el resultado observable asociado en la interfaz.

La Tabla 20 permite distinguir el papel de cada vista dentro del aplicativo. El núcleo funcional del prototipo se concentra en el simulador, mientras las demás vistas organizan la navegación y el acceso a información de apoyo. Esta distribución ayuda a delimitar el alcance real del desarrollo, se muestran las vistas principales del prototipo en las Figuras 15, 16 y 17.

Figura 15.*Vista de inicio de la plataforma.*

Nota. La figura muestra la vista de la pestaña de Inicio de la plataforma piloto, donde se da una introducción al concepto del proyecto.

Figura 16.
Vista del simulador de la plataforma.

Generador global del flujo de muones

Inicio **Simulador** Descripción

Generador global del flujo de muones

Busque una ciudad para autocompletar latitud y longitud. La altura, Bx y Bz pueden ajustarse manualmente antes de ejecutar la simulación.

Ciudad
ej., Bucaramanga

País (opcional)
ej., Colombia

Buscar ciudad

Lugar encontrado: Buenos Aires, Comuna 1, Ciudad Autónoma de Buenos Aires, Argentina

Bx y Bz actualizados. Puede ajustarlos manualmente si necesita un valor más específico.

Latitud
ej., 7.12539

Simulación terminada.

Simulation result

1 / 2

Descarga los resultados
Exporta la simulación en diferentes formatos

Formato de archivo
CSV

Selecciona el tipo de archivo a descargar.

Descargar

Nota. La figura muestra la vista a la pestaña de Simulador donde se realizan los procesos funcionales de la plataforma y se obtienen resultados.

Figura 17.
Vista de descripción de la plataforma piloto.

Generador global del flujo de muones

Inicio Simulador **Descripción**

Parámetros de entrada

B_x y B_z
Componentes del campo magnético terrestre en el sitio de observación. bx es la componente horizontal y bz la componente vertical.

Altura sobre el nivel del mar
La altitud afecta significativamente la distribución de muones debido a la menor cantidad de atmósfera que las partículas deben atravesar a mayores alturas.

Resultados generados

Distribución de energía
Representa el espectro de energía de los muones en escala logarítmica (log₁₀(E)). Permite analizar la contribución de diferentes rangos de energía.

Distribución del ángulo
Muestra cómo varía la intensidad de muones en función del ángulo cenital (θ).

© 2025 Todos los derechos reservados

Nota. La figura muestra la vista a la pestaña de Descripción donde se da detalles y explicaciones de los conceptos asociados al simulador.

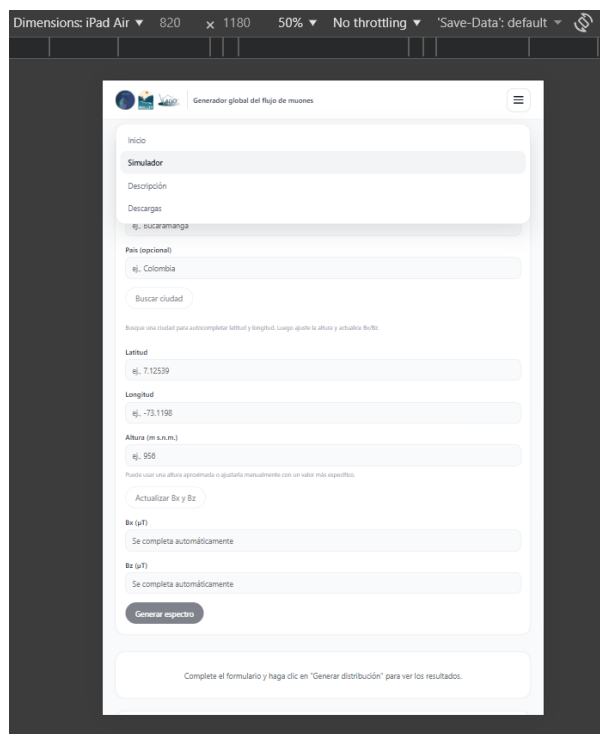
Las Figuras 15, 16 y 17 muestran las vistas principales del prototipo. Esta organización hace visible la separación entre el módulo de operación principal y las secciones de apoyo. A partir de esta distribución, la navegación del aplicativo mantiene una estructura clara y concentra

la generación de resultados en una vista específica, sin dispersar el flujo principal entre múltiples pantallas.

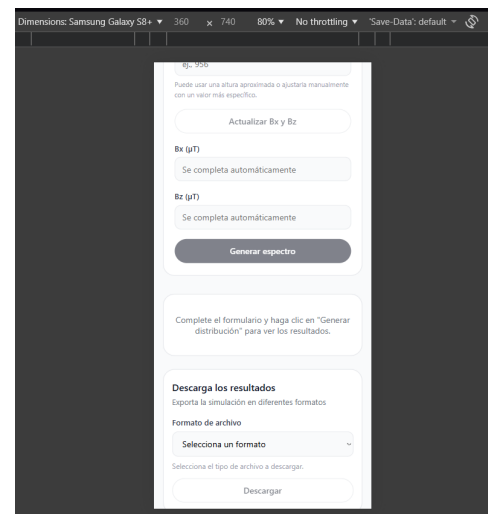
Con el fin de verificar la adaptabilidad visual de la interfaz en distintos dispositivos, se evaluó su comportamiento en resoluciones correspondientes a una tableta y a un teléfono móvil. Esta comparación permitió comprobar si la distribución de los controles, los campos de entrada y los elementos de salida se conserva de forma funcional al variar el tamaño de pantalla, como se ve en la figura 18.

Figura 18.

Adaptación responsive de la interfaz.



(a) Vista responsive en iPad Air



(b) Vista responsive en Galaxy S8

Nota. La figura compara la disposición de la interfaz en dos tamaños de pantalla. A la izquierda se presenta la visualización en una tableta y, a la derecha, en un teléfono móvil. En ambos casos se conservan los elementos principales del flujo de consulta, generación y descarga de resultados, pero se crea un menú de hamburguesa para navegar entre las pestañas.

La Figura 18 permite mostrar la adaptación de la interfaz en dos resoluciones diferentes, la de un iPad Air y la de un Samsung Galaxy S8. Este ajuste conserva la organización general del contenido y mantiene accesibles los elementos centrales del flujo de consulta. Como la plataforma

fue planteada como un aplicativo web de uso interactivo, la legibilidad de la interfaz forma parte del alcance funcional alcanzado.

Con el fin de ilustrar de manera directa el funcionamiento observado en la interfaz, se organizaron capturas de pantalla correspondientes a las etapas principales del flujo de uso del simulador. Esta secuencia permite seguir el proceso desde la vista inicial y el ingreso de la ubicación hasta la generación de los espectros y la descarga de los resultados, como se ve en las imágenes de la figura 19.

Figura 19.

Secuencia funcional del flujo principal de la plataforma.



Nota. Cada imagen corresponde a una etapa visible del flujo principal de uso, desde la interacción inicial con el simulador hasta la obtención de los resultados y su descarga.

La Figura 19 resume la secuencia principal implementada en la plataforma. El flujo inicia con la consulta de la ubicación, continúa con la obtención de las variables requeridas para la inferencia y concluye con la generación de los espectros y la descarga de los datos. Esta evidencia permite mostrar de forma directa el funcionamiento del prototipo y ubicar, dentro de una sola secuencia, las etapas que conectan la interacción del usuario con la recuperación de resultados.

El comportamiento observado de la plataforma es consistente con el alcance de un prototipo piloto orientado a exploración inicial. La base obtenida permite centralizar el uso del modelo en un entorno más accesible y deja definida una estructura sobre la cual pueden desarrollarse etapas

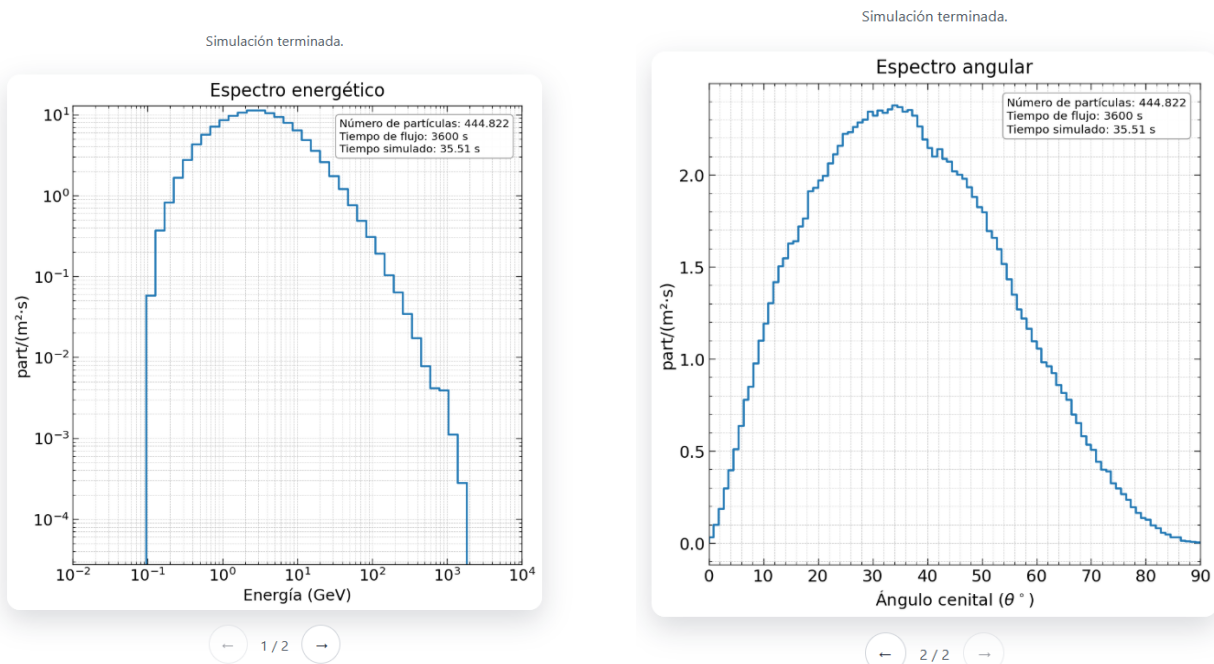
posteriores de ampliación y validación.

Resultados generados. A partir de la información ingresada y de las variables obtenidas para la inferencia, la plataforma genera salidas orientadas tanto a la visualización gráfica de los espectros como a la descarga de los datos producidos. A continuación se presentan estos resultados en dos grupos: resultados gráficos y resultados en forma de archivos.

Resultados gráficos.

Figura 20.

Resultados gráficos generados por la plataforma.



(a) Generación del espectro energético

(b) Generación del espectro angular

Resultados en forma de archivos.

Figura 21.

Archivos generados por la plataforma.

```

1 # simulation_time_s=35.51191346900305
2 event_id,energy_GeV,theta_deg,phi_deg,p_GeV_c,px_GeV_c,py_GeV_c,pz_GeV_c
3 0,1.4940970087e-01,45.99507141,280.92692777,1.0563884892e-01,1.4403277147e-02,-7.4606307164e-02,7.3389447066e-02
4 1,1.6207003545e+01,69.35636902,55.28123223,1.6206659132e+01,8.6378061639e+00,1.2465849888e+01,5.7137283326e+00
5 2,8.2692216527e+00,61.14918900,337.33966961,8.2685466105e+00,6.6831823551e+00,-2.7902004283e+00,3.9898268419e+00
6 3,2.7105093548e+00,44.70172882,0.47354062,2.7084492371e+00,1.9051018662e+00,1.5745726362e-02,1.9251068079e+00
7 4,2.4552357367e+00,32.08834839,297.56729339,2.4529612370e+00,6.0305135035e-01,-1.1551363940e+00,2.0782222717e+00
8 5,4.6243271650e-01,26.03405380,212.32742118,4.5020031650e-01,-1.6696920644e-01,-1.0566542971e-01,4.0451999461e-01

```

(a) [Archivo generado en formato CSV]

```

1 # simulation_time_s=35.51191346900305
2 # columns: event_id energy_GeV theta_deg phi_deg p_GeV_c px_GeV_c py_GeV_c pz_GeV_c
3 0 1.4940970087e-01 45.99507141 280.92692777 1.0563884892e-01 1.4403277147e-02 -7.4606307164e-02 7.3389447066e-02
4 1 1.6207003545e+01 69.35636902 55.28123223 1.6206659132e+01 8.6378061639e+00 1.2465849888e+01 5.7137283326e+00
5 2 8.2692216527e+00 61.14918900 337.33966961 8.2685466105e+00 6.6831823551e+00 -2.7902004283e+00 3.9898268419e+00
6 3 2.7105093548e+00 44.70172882 0.47354062 2.7084492371e+00 1.9051018662e+00 1.5745726362e-02 1.9251068079e+00
7 4 2.4552357367e+00 32.08834839 297.56729339 2.4529612370e+00 6.0305135035e-01 -1.1551363940e+00 2.0782222717e+00
8 5 4.6243271650e-01 26.03405380 212.32742118 4.5020031650e-01 -1.6696920644e-01 -1.0566542971e-01 4.0451999461e-01

```

(b) [Archivo generado en formato SHW]

La estructura de los archivos generados, como se muestra en la Figura 21, sigue el formato de un archivo de simulación de ARTI, como se muestra en la Figura 2. Para construirlos, se utilizaron la energía y el ángulo generados por el modelo y, a partir de las ecuaciones 3.1–3.4, se calcularon las componentes del momento p_x , p_y y p_z .

4.4.1. Resultados del despliegue y validación en entorno real

El despliegue permitió verificar la ejecución del prototipo fuera del entorno local, manteniendo separadas la interfaz publicada en Vercel y la API alojada en Hugging Face Spaces. Bajo esta configuración, el usuario puede ingresar los parámetros del sitio, solicitar la generación de espectros, visualizar los resultados y descargar los archivos producidos desde el navegador.

Durante la validación del despliegue se identificó que el plan gratuito de Render no ofrecía memoria suficiente para sostener la ejecución del servicio de inferencia, lo que generaba interrupciones durante la generación de espectros. Por esta razón, el *backend* se trasladó a Hugging Face Spaces, cuya modalidad gratuita permitió ejecutar el contenedor con mayor disponibilidad de memoria y mejor manejo de los archivos temporales requeridos por el flujo.

Con esta integración se comprobó el funcionamiento de los servicios principales de la plataforma: resolución de coordenadas, cálculo de las componentes B_x y B_z , generación de espectros y descarga de resultados en formatos CSV y SHW. La plataforma se encuentra disponible para consulta en el siguiente enlace: <https://muon-generator.vercel.app>.

5. Conclusión

Este trabajo demostró que es posible modelar el espectro de muones atmosféricos mediante flujos normalizantes condicionales, alcanzando un ajuste comparable al de las simulaciones ARTI con un costo computacional más de tres órdenes de magnitud inferior. Con ello, se da respuesta afirmativa a la pregunta de investigación planteada.

El modelo desarrollado opera como un sustituto eficaz de la simulación de referencia dentro del dominio cubierto por el entrenamiento. Su precisión, medida mediante el error relativo absoluto promedio, fue de 0,1160 en validación y 0,0925 en prueba, con un comportamiento estable ante variaciones en la partición de datos y la inicialización. La reducción del tiempo de generación de 8 horas 50 minutos por ciudad a 60 segundos aproximados representa una aceleración superior a $500\times$, lo que hace viable su uso en aplicaciones de consulta rápida donde la simulación directa resulta impráctica. Adicionalmente, el modelo es capaz de estimar el espectro en sitios donde no existen simulaciones previas de ARTI, lo que amplía su utilidad a ubicaciones arbitrarias dentro del dominio geofísico cubierto por el entrenamiento.

La integración del modelo en una plataforma web representa un aporte de accesibilidad relevante: cualquier usuario, independientemente de su capacidad computacional o conocimiento técnico, puede obtener una estimación del flujo de muones correspondiente a una hora de simulación para cualquier ubicación del mundo, de forma rápida y sin ningún cálculo adicional.

Sin embargo, el trabajo también expuso limitaciones que deben considerarse. El modelo

mostró mayores dificultades en regiones de baja densidad, particularmente en la cola energética de 10^3 a 10^4 GeV y en los extremos angulares, lo cual es consistente con limitaciones documentadas en la literatura para esta familia de modelos. Su capacidad de generalización está además acotada al dominio geofísico cubierto por las 45 ubicaciones de entrenamiento.

A partir de estos hallazgos, se identifican varias líneas de trabajo futuro. Ampliar el conjunto de entrenamiento con simulaciones de mayor duración y cobertura geográfica extendería el dominio de validez del modelo. Explorar arquitecturas alternativas podría mejorar la representación en regiones de baja densidad. Finalmente, la metodología empleada es transferible a otros problemas de física donde se requieran modelos sustitutos de simulaciones Monte Carlo costosas.

Divulgación de los resultados

Durante la realización de este trabajo de grado se generaron espacios de divulgación científica asociados a los resultados obtenidos en el desarrollo de la investigación.

- ***Muon spectrum generation with generative AI.*** J. Almanzar Quintero, C. Orduz Carvajal, R. A. Martinez Rivero, L. Nunez y C. Sarmiento-Cano. Resumen aceptado en modalidad póster en *Muographers 2026*. ID del resumen: 40. **Aceptada.**

Además, los resultados del trabajo se presentaran en el siguiente evento académico:

- Ponencia, ***Estimación del espectro de muones usando algoritmos de inteligencia artificial.*** C. Orduz Carvajal y J. Almanzar Quintero R. A. Martinez Rivero, L. Nunez y C. Sarmiento-Cano.. En el *Encuentro CyTED LAGO INDICA*. Universidad Industrial de Santander, Bucaramanga, Colombia, diciembre de 2025.

Estos espacios permitieron divulgar los avances del trabajo ante comunidades académicas vinculadas con el estudio de muones, la simulación de rayos cósmicos y el uso de técnicas computacionales para la estimación de espectros.

Referencias bibliográficas

- Alvarez, L. W., Anderson, J. A., El Bedwei, F., Burkhard, J., Fakhry, A., Girgis, A., Goneid, A., Hassan, F., Iverson, D., Lynch, G., Miligy, Z., Moussa, A. H., Sharkawi, M., y Yazolino, L. (1970). Search for Hidden Chambers in the Pyramids. *Science*, 167(3919), 832-839. <https://doi.org/10.1126/science.167.3919.832>
- Angular. (2026). What is Angular? [Accessed: 2026-04-10]. <https://angular.dev/overview>
- Antoni, T., Apel, W. D., Badea, F., Bekk, K., Bercuci, A., Blümer, H., Bozdog, H., Brancus, I. M., Büttner, C., Chilingarian, A., Daumiller, K., Doll, P., Engler, J., Feßler, F., Gils, H. J., Glastetter, R., Haeusler, R., Haungs, A., Heck, D., . . . y Zagromski, S. (2003). The cosmic-ray experiment KASCADE. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 513(3), 490-510. [https://doi.org/10.1016/S0168-9002\(03\)02076-X](https://doi.org/10.1016/S0168-9002(03)02076-X)
- Baack, D. (2019). Performance optimization of air shower simulations with CORSIKA. *Proceedings of the 36th International Cosmic Ray Conference (ICRC2019)*, 181.
- Bergstra, J., y Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(10), 281-305. <https://jmlr.org/papers/v13/bergstra12a.html>
- Biallass, P., y Hebbeker, T. (2009). Parametrization of the Cosmic Muon Flux for the Generator CMSCGEN.
- Bond-Taylor, S., Leach, A., Long, Y., y Willcocks, C. G. (2022). Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autore-

- gressive Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11), 7327-7347. <https://doi.org/10.1109/TPAMI.2021.3116668>
- Bonechi, L., D'Alessandro, R., y Giammanco, A. (2020). Atmospheric muons as an imaging tool. *Reviews in Physics*, 5, 100038. <https://doi.org/10.1016/j.revip.2020.100038>
- Carminati, G., Bazzotti, M., Margiotta, A., y Spurio, M. (2008). Atmospheric MUons from PArametric formulas: a fast GEnerator for neutrino telescopes (MUPAGE). *Computer Physics Communications*, 179(12), 915-923. <https://doi.org/10.1016/j.cpc.2008.07.014>
- Cecchini, S., y Spurio, M. (2012). Atmospheric muons: experimental aspects. *Geoscientific Instrumentation, Methods and Data Systems*, 1(2), 185-196. <https://doi.org/10.5194/gi-1-185-2012>
- Cheng, Y., Liu, Y., y Xu, W. (2017). Maximum Reconstruction Estimation for Generative Latent-Variable Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1). <https://doi.org/10.1609/aaai.v31i1.10971>
- de Amorim, L. B. V., Cavalcanti, G. D. C., y Cruz, R. M. O. (2023). The choice of scaling technique matters for classification performance. *Applied Soft Computing*, 133, 109924. <https://doi.org/10.1016/j.asoc.2022.109924>
- de León-Barrios, R., Peña-Rodríguez, J., Sanabria-Gómez, J., Vásquez-Ramírez, A., Calderón-Ardila, R., Sarmiento-Cano, C., Vesga-Ramírez, A., Sierra-Porta, D., Suárez-Durán, M., Asorey, H., y Núñez, L. (2021). Muography for the Colombian Volcanoes. *Proceedings of the 37th International Cosmic Ray Conference (ICRC2021)*, 280. <https://doi.org/10.22323/1.395.0280>
- Durkan, C., Bekasov, A., Murray, I., y Papamakarios, G. (2019). Neural Spline Flows. *Advances in Neural Information Processing Systems*, 32, 7509-7520.
- Fajardo, J. A. (2018). *Muografía de estructuras geológicas en Colombia* [Tesis de maestría, Universidad de los Andes]. <https://doi.org/10.57784/1992/34812>
- Fielding, R. T., Nottingham, M., y Reschke, J. (2022). *HTTP Semantics* (inf. téc. N.º RFC 9110). RFC Editor. <https://doi.org/10.17487/RFC9110>

- Guan, M., Chu, M.-C., Cao, J., Luk, K.-B., y Yang, C. (2015). A parametrization of the cosmic-ray muon flux at sea-level.
- Hebbeker, T., y Timmermans, C. (2002). A compilation of high energy atmospheric muon data at sea level. *Astroparticle Physics*, 18(1), 107-127. [https://doi.org/10.1016/S0927-6505\(01\)00180-3](https://doi.org/10.1016/S0927-6505(01)00180-3)
- Heck, D., Hüge, T., y Pierog, T. (2025). Extensive Air Shower Simulation with CORSIKA: A User's Guide [Version 7.8050, December 18, 2025, Karlsruhe Institute of Technology (KIT)]. https://www.iap.kit.edu/corsika/downloads/CORSIKA_GUIDE7.8050.pdf
- Heck, D., Pierog, T., y Knapp, J. (2012). CORSIKA: An Air Shower Simulation Program.
- Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., y Meger, D. (2018). Deep Reinforcement Learning That Matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 3207-3214. <https://doi.org/10.1609/aaai.v32i1.11694>
- IceCube Collaboration. (2024). Conditional normalizing flows for IceCube event reconstruction [PoS(ICRC2023)1003]. *Proceedings of Science*, 444, 1003. <https://pos.sissa.it/444/1003/>
- Jordan, M. I., y Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260. <https://doi.org/10.1126/science.aaa8415>
- Kaiser, R. (2019). Muography: overview and future directions. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 377(2137), 20180049. <https://doi.org/10.1098/rsta.2018.0049>
- Kingma, D. P., y Ba, J. (2015). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/1412.6980>
- LAGO Collaboration. (2022). LAGO sites 1.1 - LAGO Data Management Plan [Version 1.1, updated 2022-09-15, accessed 2026-04-11]. <https://lagoproject.github.io/DMP/defs/sitesLago.1.1/>
- LAGO Collaboration. (2024). Applications of Low Energy Astroparticle Simulations on HPC and Cloud Infrastructures. *Proceedings of Science (ICRC2023)*, 242.

- Lahkar, M. L. (2023). *High-Performance Web Apps with FastAPI: The Asynchronous Web Framework Based on Modern Python*. BPB Publications.
- Laszkiewicz, M., Lederer, J., y Fischer, A. (2022). Marginal Tail-Adaptive Normalizing Flows. *Proceedings of the 39th International Conference on Machine Learning*, 162, 12020-12048. <https://proceedings.mlr.press/v162/laszlkiewicz22a.html>
- LeCun, Y., Bengio, Y., y Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444. <https://doi.org/10.1038/nature14539>
- Lu, Y., y Huang, B. (2020). Structured Output Learning with Conditional Generative Flows. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04), 5005-5012. <https://doi.org/10.1609/aaai.v34i04.5940>
- Martínez-Rivero, R. A., Sarmiento-Cano, C., Castillo-Morales, D., Perea-Pérez, J., Baldovino-Medrano, V. G., Sanabria-Gómez, J. D., y Núñez, L. A. (2025). Muon imaging of hydro-treatment reactors. *Journal of Applied Physics*, 138(24), 244901. <https://doi.org/10.1063/5.0281667>
- Mehta, P., Bukov, M., Wang, C.-H., Day, A. G. R., Richardson, C., Fisher, C. K., y Schwab, D. J. (2019). A high-bias, low-variance introduction to Machine Learning for physicists. *Physics Reports*, 810, 1-124. <https://doi.org/10.1016/j.physrep.2019.03.001>
- Mo, Y., y Li, B. (2025). ArchiWeb: A web platform for AI-driven early-stage architectural design. *Frontiers of Architectural Research*, 14(6), 1551-1566. <https://doi.org/10.1016/j.foar.2025.06.002>
- Ng, A. Y., y Jordan, M. I. (2001). On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. *Advances in Neural Information Processing Systems (NIPS)*, 14, 841-848.
- Nguyen, T. D., Shih, M.-H., Srivastava, D., Tirthapura, S., y Xu, B. (2019). Stratified Random Sampling from Streaming and Stored Data. *Proceedings of the 22nd International Conference on Extending Database Technology (EDBT)*, 25-36. <https://doi.org/10.5441/002/edbt.2019.04>

- Nilsson, N. (2011). The Quest for Artificial Intelligence: A History of Ideas and Achievements. *Isis*, 102, 588-589. <https://doi.org/10.1086/663065>
- Oláh, L., Tanaka, H., Hamar, G., y Varga, D. (2018). Investigation of the limits of high-definition muography for observation of Mt. Sakurajima. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 377(2137), 20180135. <https://doi.org/10.1098/rsta.2018.0135>
- Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., y Lakshminarayanan, B. (2021). Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research*, 22(57), 1-64.
- Papamakarios, G., Pavlakou, T., y Murray, I. (2017). Masked Autoregressive Flow for Density Estimation. *Advances in Neural Information Processing Systems*, 30, 2338-2347.
- Particle Data Group. (2013). Kinematics [Revised January 2000 by J.D. Jackson and June 2008 by D.R. Tovey]. <https://pdg.lbl.gov/2013/reviews/rpp2013-rev-kinematics.pdf>
- Saracino, G., Ambrosino, F., Bonechi, L., Cimmino, L., D'Alessandro, R., D'Errico, M., Noli, P., Scognamiglio, L., y Strolin, P. (2019). Applications of muon absorption radiography to the fields of archaeology and civil engineering. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 377(2137), 20180057. <https://doi.org/10.1098/rsta.2018.0057>
- Sarmiento-Cano, C., Suárez-Durán, M., Calderón-Ardila, R., Vásquez-Ramírez, A., Jaimes-Motta, A., Núñez, L. A., Dasso, S., Sidelnik, I., Asorey, H., y LAGO Collaboration. (2022). The ARTI framework: cosmic rays atmospheric background simulations. *The European Physical Journal C*, 82(11), 1019. <https://doi.org/10.1140/epjc/s10052-022-10883-z>
- Shlezinger, N., y Routtenberg, T. (2023). Discriminative and Generative Learning for the Linear Estimation of Random Signals [Lecture Notes]. *IEEE Signal Processing Magazine*, 40(6), 75-82. <https://doi.org/10.1109/MSP.2023.3271431>

- Singha, A., Chakrabarti, D., y Arora, V. (2023). Conditional normalizing flow for Markov chain Monte Carlo sampling in the critical region of lattice field theory. *Physical Review D*, 107(1), 014512. <https://doi.org/10.1103/PhysRevD.107.014512>
- Snodgrass, J. E., y Milkov, A. V. (2020). Web-based machine learning tool that determines the origin of natural gases. *Computers & Geosciences*, 145, 104595. <https://doi.org/10.1016/j.cageo.2020.104595>
- Turing, A. M. (2009). Computing Machinery and Intelligence. En R. Epstein, G. Roberts, y G. Beber (Eds.), *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer* (pp. 23-65). Springer Netherlands. https://doi.org/10.1007/978-1-4020-6710-5_3
- Vesga-Ramírez, A., Sanabria-Gómez, J. D., Sierra-Porta, D., Arana-Salinas, L., Asorey, H., Kudryavtsev, V. A., Calderón-Ardila, R., y Núñez, L. A. (2021). Simulated Annealing for volcano muography. *Journal of South American Earth Sciences*, 109, 103248. <https://doi.org/10.1016/j.jsames.2021.103248>
- Wang, A.-B., Pan, C.-C., Dong, X., Sun, Y.-C., Hu, Y.-X., Cheng, A.-Y., Cai, H., y Fan, X.-L. (2024). DeepMuon: Accelerating Cosmic-Ray Muon Simulation Based on Optimal Transport. <https://arxiv.org/abs/2410.06539>
- Winkler, C., Harder, P., y Rolnick, D. (2023). Climate Variable Downscaling with Conditional Normalizing Flows [Available online]. *NeurIPS 2023 Workshop on Tackling Climate Change with Machine Learning (Climate Change AI Workshop)*. <https://arxiv.org/abs/2405.20719>
- Winkler, C., Worrall, D., Hoogeboom, E., y Welling, M. (2019). Learning Likelihoods with Conditional Normalizing Flows [arXiv preprint].
- Yu, J. J., Derpanis, K. G., y Brubaker, M. A. (2020). Wavelet Flow: Fast Training of High Resolution Normalizing Flows. *Advances in Neural Information Processing Systems (NeurIPS 2020)*, 33, 6184-6196.

Apéndices

A. Ejemplo de estructura JSON para un sitio de simulación

```
{
  "@id": "lim",
  "@type": "lago:DetectorSite",
  "name": "Lima",
  "country": "Peru",
  "geometry": {
    "@id": "lim#geometry",
    "@type": "geo:Point",
    "geo:latitude": "-12.1",
    "geo:longitude": "-77.02",
    "geo:altitude": "150"
  },
  "lago:magnet": {
    "@default": {
      "lago:bx": "24.723",
      "lago:bz": "-0.476",
      "lago:bi": "24.751"
    }
  },
  "type": "lagoCollaboration"
}
```

B. Script `runner.sh` para la ejecución automatizada de simulaciones

```
#!/usr/bin/env bash

# run_go_bga_full_workflow.sh

# 1) Ejecuta los 'he' uno a uno (30 s entre cada uno)

# 2) Ejecuta los 'all' en lotes de 3 en paralelo (30 s entre lanzamientos, espera a
    screens)

# 3) Ejecuta los 'pr' en lotes de 3 en paralelo
#     - 30 s entre lanzamientos
#     - 5 min despues del tercer lanzamiento antes de comprobar screens
# Genera log con hora de inicio y fin de cada script.

set -euo pipefail
IFS=$'\n\t'
shopt -s nullglob extglob

# Parametros
DELAY=30          # Delay entre lanzamientos (s)
PR_PRECHECK=120 # Espera antes de chequear screens tras lote 'pr' (s)
LOG="run_go_bue_log_$(date +%Y%m%d_%H%M%S).txt"

# [NTFY] Topic para notificaciones
TOPIC_URL="https://ntfy.sh/cris-obatala"

echo "Ejecucion iniciada el $(date +%Y-%m-%d %H:%M:%S)" > "$LOG"
```

```
# [NTFY] Funcion de notificacion al salir (exito o fallo)
notify_exit() {
    local status=$?
    if [[ $status -eq 0 ]]; then
        curl -d " runner.sh TERMINADO en Obatala (exit 0). Log: $LOG" \
            "$TOPIC_URL" \
            || echo " No se pudo enviar la notificacion ntfy (exito)"
    else
        curl -d " runner.sh FALLO en Obatala (exit $status). Revisa el log: $LOG" \
            "$TOPIC_URL" \
            || echo " No se pudo enviar la notificacion ntfy (error)"
    fi
}

trap notify_exit EXIT

wait_screens_clear() {
    echo " Esperando a que no queden sesiones de screen activas..."
    while screen -ls | grep -q "[0-9]\+\."; do
        sleep 5
    done
}

run_and_log() {
    local script=$1
    local start=$(date +%Y-%m-%d %H:%M:%S)
    echo "> $start Iniciando: $script" | tee -a "$LOG"
    [[ -x $script ]] || chmod +x "$script"
```

```
bash "$script"

local status=$?

local end=$(date +%Y-%m-%d %H:%M:%S')

echo "< $end Finalizado: $script (exit $status)" | tee -a "$LOG"
}

# 1) Fase 'he'
he_list=( $(printf '%s\n' go-*he.sh 2>/dev/null | sort -V) )
if (( ${#he_list[@]} )); then

    echo -e "\n=== Fase 1: Ejecutando ${#he_list[@]} scripts 'he' ==="

    for s in "${he_list[@]"; do

        wait_screens_clear

        run_and_log "$s"

        echo "    Durmiendo ${DELAY}s..."

        sleep "$DELAY"

    done
else
    echo " No se encontraron scripts 'he'"
fi

# 2) Fase 'all' en lotes de 3
all_list=( $(printf '%s\n' go-*all-*.sh 2>/dev/null | sort -V) )
if (( ${#all_list[@]} )); then

    echo -e "\n=== Fase 2: Ejecutando ${#all_list[@]} scripts 'all' en lotes de 3
    ==="

    for (( i=0; i<${#all_list[@]}; i+=3 )); do

        batch=( "${all_list[@]:i:3}" )

        echo -e "\n-- Lote ALL de ${#batch[@]} scripts --"

        wait_screens_clear
```

```
    for s in "${batch[@]}"; do
        run_and_log "$s" &
        echo "    Durmiendo ${DELAY}s..."
        sleep "$DELAY"
    done
    wait_screens_clear
done
else
    echo " No se encontraron scripts 'all'"
fi

# 3) Fase 'pr' en lotes de 3
pr_list=( $(printf '%s\n' go-*pr-*.sh 2>/dev/null | sort -V) )
if (( ${#pr_list[@]} )); then
    echo -e "\n=== Fase 3: Ejecutando ${#pr_list[@]} scripts 'pr' en lotes de 3 ==="
    for (( i=0; i<${#pr_list[@]}; i+=3 )); do
        batch=( "${pr_list[@]:i:3}" )
        echo -e "\n-- Lote PR de ${#batch[@]} scripts --"
        wait_screens_clear

        for s in "${batch[@]}"; do
            run_and_log "$s" &
            echo "    Durmiendo ${DELAY}s antes del siguiente lanzamiento..."
            sleep "$DELAY"
        done

        echo "    Esperando ${PR_PRECHECK}s antes de chequear sesiones de screen..."
        sleep "$PR_PRECHECK"
        wait_screens_clear
    done
fi
```

```
done
else
    echo " No se encontraron scripts 'pr'"
fi

echo -e "\nWorkflow completo. Revisa el log: $LOG"
```

C. Ubicaciones simuladas del conjunto de datos final

Tabla C.1

Ubicaciones simuladas del conjunto de datos final

Ciudad	Sigla	País	Latitud	Longitud	$h(m)$	$B_x(\mu T)$	$B_z(\mu T)$	$B_t(\mu T)$
Lima	lim	Perú	-12.1	-77.02	150	24.723	-0.476	24.751
Chiapas	chia	México	16.75	-93.12	522	27.405	27.024	38.497
Bucaramanga	bga	Colombia	7.14	-73.12	956	26.793	16.055	31.506
Malargue	mge	Argentina	-35.45	-69.57	1450	18.987	-14.326	23.79
Guatemala	gua	Guatemala	14.63	-90.59	1490	27.532	24.932	37.145
Creighton	sudb	Canadá	46.46	-81.17	2100	17.037	51.991	54.79
Casposo	casp	Argentina	-31.22	-69.63	2600	19.473	-12.436	23.109
Quito	quie	Ecuador	-0.2	-78.5	2850	26.717	9.971	28.579
Berlin	ber	Colombia	7.19	-72.88	3450	26.751	16.029	31.467
La paz	lpb	Bolivia	-16.49	-68.15	3630	22.362	-4.708	23.109
Andes	and	Argentina	-30.19	-69.82	4200	19.658	-11.951	23.011
Sierra Negra	sng	México	18.16	-97.95	4550	27.358	28.038	39.217
Imata	ima	Perú	-15.841	-71.091	4600	22.969	-3.79	23.422
Atacama	ata	Chile	-23.0058	-67.7592	5105	20.638	-8.598	22.5
Chacaltaya	cha	Bolivia	-16.35	-68.13	5230	22.38	-4.62	23.112
Brasilia	bra	Brasil	-15.464	-47.553	1172	20.148	-11.931	23.4158
Trujillo	truj	España	39.47	-5.88	560	26.223	35.844	44.413
Londres	lond	Reino Unido	51.302	-0.739	24	19.542	45.044	49.102
Iqalit	iqa	Canadá	63.75	-68.55	34	9.63	55.928	56.75
Pamplona	pamp	Colombia	7.22	-72.39	2342	26.743	15.955	31.443
Riobama	serb	Ecuador	-1.81	-78.74	2750	26.577	8.499	27.952
Serro machin	srm	Colombia	4.36	-75.23	2750	26.815	13.63	30.084
Tibet	tib	China	31.42	86.56	4500	32.868	39.403	51.312
Lisboa	lisb	Portugal	38.42	-9.88	0	27	34.984	44.02
Santo domingo	std	República Dominicana	18.28	-69.54	14	26.89	25.28	36.909

Ciudad	Sigla	País	Latitud	Longitud	$h(m)$	$B_x(\mu T)$	$B_z(\mu T)$	$B_i(\mu T)$
Komic	kom	Croacia	44.27	15.44	4587	23.028	41.983	47.884
Albuquerque	alb	Estados Unidos	35.084	-106.65	1619	22.817	42.54	48.273
Denver	denv	Estados Unidos	39.739	-104.99	1609	20.863	46.587	51.045
Fairplay	fair	Estados Unidos	39.133	-105.595	3034	21.139	45.998	50.623
Santos Cuchumatan	cuchu	Guatemala	15.31	-91.36	3250	27.177	25.43	37.219
Puebla	pue	México	19.216	-98.1212	2143	26.997	28.579	39.314
Addis Ababa	add	Etiopía	9.159	38.45	2355	35.707	2.106	35.769
Pradollano	pra	España	37.54	-3.236	2143	27.546	33.868	43.656
Shahr-e Kord	sha	Irán	32.19	50.51	2070	30.067	36.472	47.267
Kerman	ker	Irán	30.1659	57.443	3150	31.5848	34.898	47.069
Asmara	asm	Eritrea	15.202	38.561	3250	36.118	12.016	38.064
Tawang	taw	India	27.588	91.865	3000	35.631	34.127	49.338
Jungfrauoch	jun	Suiza	46.325	7.584	3454	22.038	42.626	47.986
Aspen	asp	Estados Unidos	39.11	106.5	2405	21.219	45.844	50.517
Mauna Kea summit	mks	Estados Unidos	19.35	-155.58	4205	27.464	20.251	34.123
Khunjerab pass	khun	Pakistán	36.51	75.25	4700	28.896	44.161	52.726
Zard-kuh	zard	Irán	32.354	49.475	4221	29.833	36.581	47.203
Erzurum	erz	Turquía	39.54	41.15	1890	25.357	42.016	49.074
San Jose	sjs	Costa Rica	9.556	-84.45	1172	27.216	19.757	33.631
Shiraz	shi	Irán	29.364	52.315	1500	31.694	33.68	46.247

Nota. La tabla reúne las 45 ubicaciones incluidas en el conjunto final de simulaciones. La columna Sigla corresponde al identificador abreviado utilizado para nombrar los archivos y referirse a cada sitio a lo largo del estudio; h indica la altitud sobre el nivel del mar, B_x y B_z representan las componentes geomagnéticas empleadas en el modelado, y B_i corresponde a la intensidad total del campo magnético en cada ubicación. Las coordenadas geográficas se reportan en

grados decimales.

D. Precisión y tiempo en la generación de eventos por ciudad

Tabla D.1

Eventos generados por ciudad en los conjuntos de validación y prueba

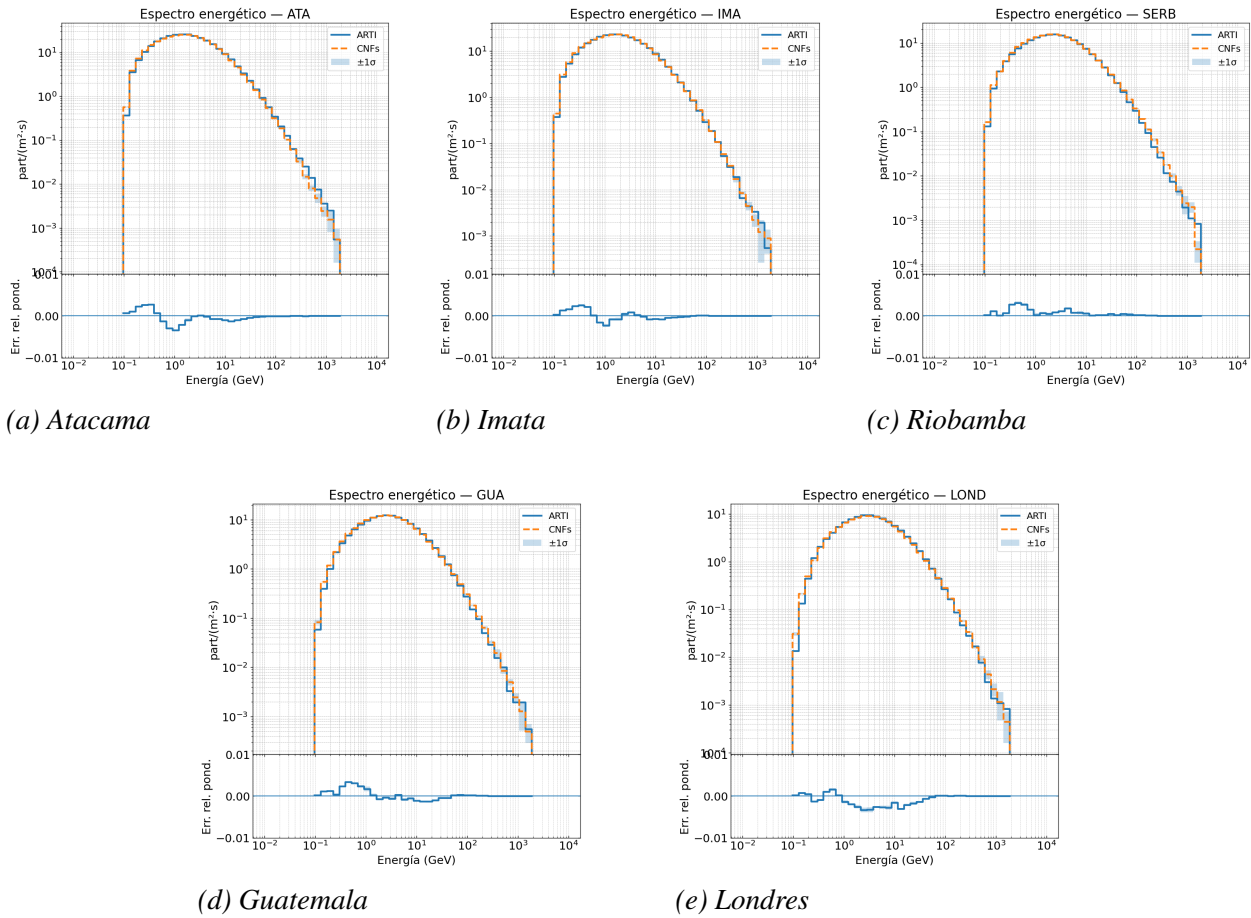
Ciudad	Eventos ARTI	Eventos CNF	Error eventos	Tiempo (s)
Validación				
Londres	374027	365445	0,0229	13,04
Guatemala	491361	497848	0,0132	17,53
Riobamba	633340	649387	0,0253	22,92
Imata	955306	959291	0,0042	33,72
Atacama	1074497	1067099	0,0069	37,44
Prueba				
San José	467856	465554	0,0049	16,38
Shiraz	503205	498899	0,0086	17,78
Santos Cuchumatán	722337	721606	0,0010	25,46
Jungfraujoeh	758579	753330	0,0069	26,68
Zard-Kuh	885700	885598	0,0001	31,22

Nota. Se presenta el número de eventos de referencia obtenidos con ARTI y el número de eventos generados por el CNF para cada ciudad evaluada. El error de eventos corresponde a la discrepancia relativa entre ambas cantidades. El tiempo se reporta en segundos (s).

E. Resultados para las ciudades del conjunto de validación

Figura E.1.

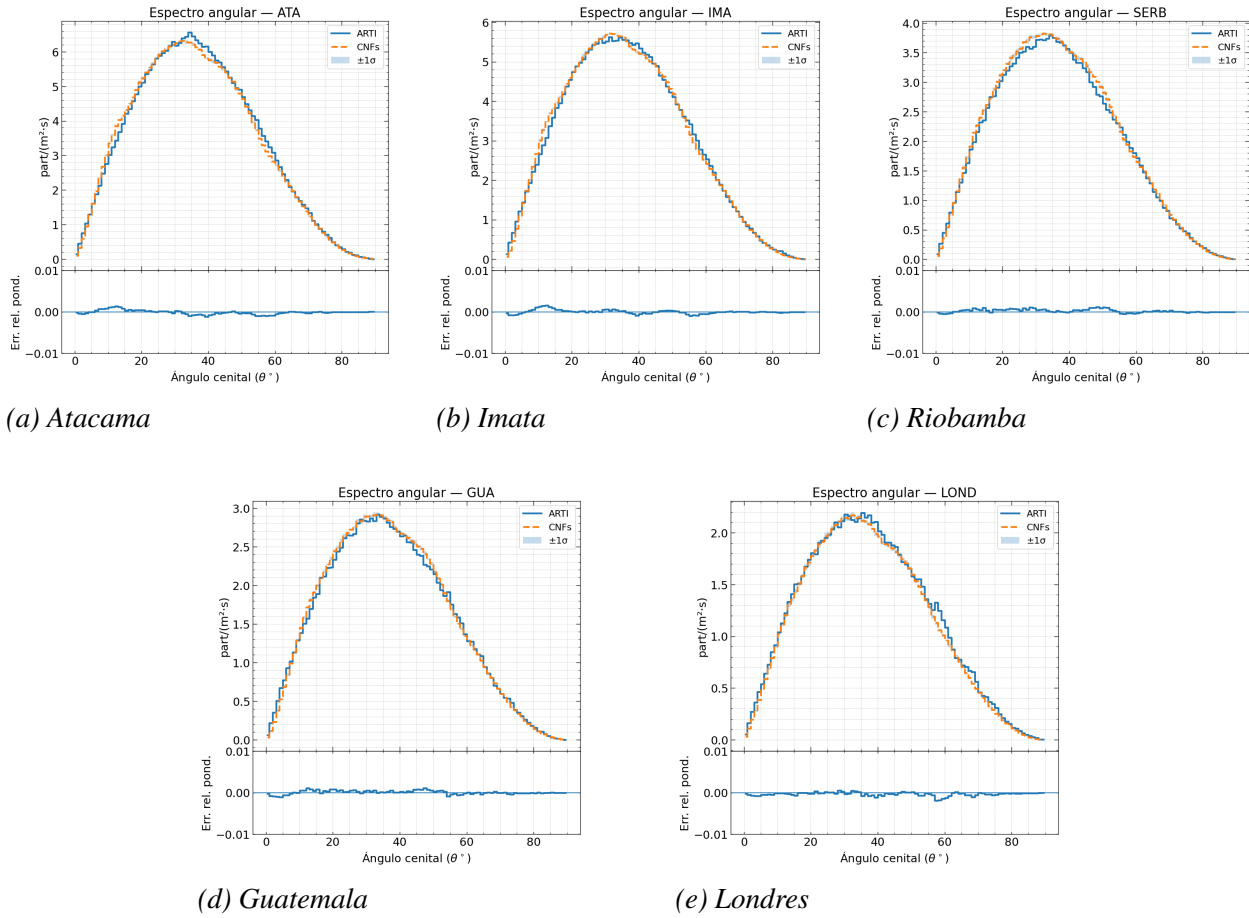
Comparación del espectro energético proporcionado por ARTI y el generado por el CNF en ciudades de validación.



Nota. Se compara el espectro de energía de referencia proporcionado por ARTI con el espectro generado por el CNF en cinco ciudades. En el panel superior de cada gráfica, la línea azul continua representa el espectro de referencia y la línea naranja discontinua el espectro medio generado por el modelo; la banda sombreada indica la desviación estándar ($\pm 1\sigma$) entre corridas. En el panel inferior se presenta el aporte del error relativo ponderado por intervalo de energía. La energía se expresa en GeV (gigaelectronvoltios) y el eje vertical del panel superior en partículas por metro cuadrado y por segundo.

Figura E.2.

Comparación del espectro angular proporcionado por ARTI y el generado por el CNF en ciudades de validación.

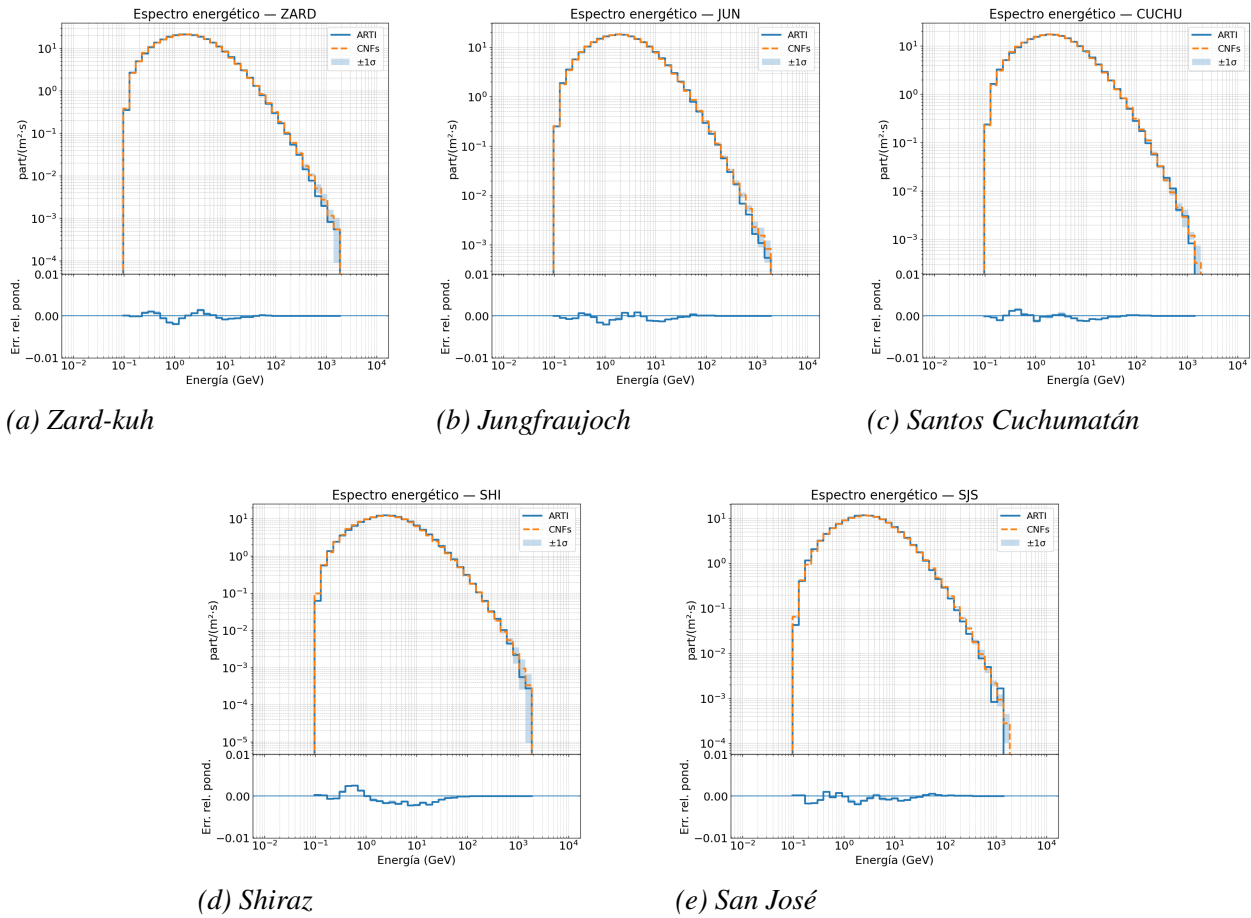


Nota. Se compara el espectro angular de referencia proporcionado por ARTI con el espectro generado por el CNF en cinco ciudades. En el panel superior de cada gráfica, la línea azul continua representa el espectro de referencia y la línea naranja discontinua el espectro medio generado por el modelo; la banda sombreada indica la desviación estándar ($\pm 1\sigma$) entre corridas. En el panel inferior se presenta el aporte del error relativo ponderado por intervalo angular. El ángulo cenital se expresa en grados y el eje vertical del panel superior en partículas por metro cuadrado y por segundo.

F. Resultados para las ciudades del conjunto de prueba

Figura F.1.

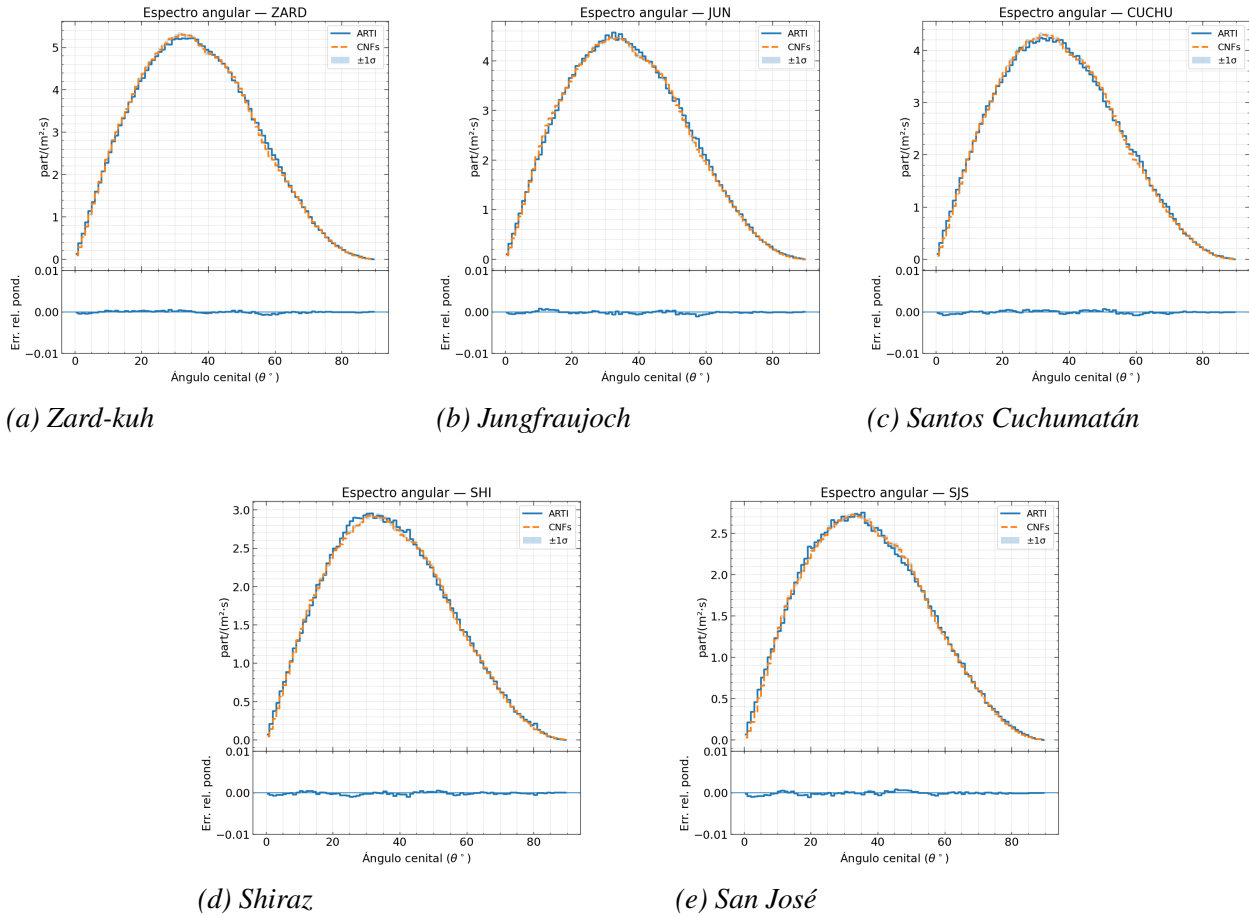
Comparación del espectro energético proporcionado por ARTI y el generado por el CNF en ciudades de prueba.



Nota. Se compara el espectro de energía de referencia proporcionado por ARTI con el espectro generado por el CNF en cinco ciudades del conjunto de prueba. En el panel superior de cada gráfica, la línea azul continua representa el espectro de referencia y la línea naranja discontinua el espectro medio generado por el modelo; la banda sombreada indica la desviación estándar ($\pm 1\sigma$) entre corridas. En el panel inferior se presenta el aporte del error relativo ponderado por intervalo de energía. La energía se expresa en GeV (gigaelectronvoltios) y el eje vertical del panel superior en partículas por metro cuadrado y por segundo.

Figura F.2.

Comparación del espectro angular proporcionado por ARTI y el generado por el CNF en ciudades de prueba.



Nota. Se compara el espectro angular de referencia proporcionado por ARTI con el espectro generado por el CNF en cinco ciudades del conjunto de prueba. En el panel superior de cada gráfica, la línea azul continua representa el espectro de referencia y la línea naranja discontinua el espectro medio generado por el modelo; la banda sombreada indica la desviación estándar ($\pm 1\sigma$) entre corridas. En el panel inferior se presenta el aporte del error relativo ponderado por intervalo angular. El ángulo cenital se expresa en grados y el eje vertical del panel superior en partículas por metro cuadrado y por segundo.