Universidad de Cantabria

Facultad de Ciencias

Estudio del Halo en Imágenes Tomadas con Skipper CCds
Study of the Halo in Images Acquired with Skipper CCDs

Trabajo de fin de grado
Grado en Física

Autor: Ismael Pardo Ortiz
Directora: Rocio Vilar Cortabitarte
Codirectora: Nuria Castelló Mor

Junio 2025

Resumen

Este Trabajo de Fin de Grado se dedica al estudio de uno de los fondos presentes en las imágenes recogidas por el prototipo Moskita, ubicado en el CERN a 80 metros bajo tierra cerca del anillo del LHC. En particular, nos enfocamos en analizar las deposiciones de muy baja energía, que aparecen en forma de pequeños grupos de píxeles activados (denominados clusters) alrededor de las señales producidas por el paso de partículas en los sensores de carga acoplada (CCDs), fenómeno al que denominaremos halo. Las CCDs son dispositivos de imagen de alta sensibilidad, empleadas principalmente en astronomía para la detección de señales extremadamente débiles en grandes telescopios. Desde 2012, estos sensores se han empezado a utilizar en experimentos de física de partículas orientados a la búsqueda de sucesos raros y de baja energía, como por ejemplo partículas de materia oscura, partículas con milicarga eléctrica u otras partículas exóticas no predichas por el Modelo Estándar de física de partículas.

Para obtener información útil de las imágenes registradas por los CCDs, es necesario aplicar un tratamiento de imágenes que incluye limpiar el pedestal, corregir los hot pixels(defectos) y las columnas calientes, calibrar el sistema, y a partir de ahí realizar una reconstrucción de las deposiciones de energía. En este proceso, se identifican y caracterizan clusters de baja y alta energía, que representan eventos generados por partículas que interactúan con el sensor. La búsqueda de nuevas partículas se centra en encontrar los clusters poco energéticos, al esperarse que tengan interacciones muy débiles con la materia, dejarán deposiciones de muy baja energía en las CCDs (por debajo de los 20–30 eV). Por tanto, caracterizar estas deposiciones tan bajas es crucial para la búsqueda de nuevas partículas. Un problema importante en esta búsqueda es que los clusters de alta energía pueden producir clusters secundarios de baja energía, que imitan las señales de interés, generando un fondo que dificulta la interpretación de los datos.

El objetivo de este trabajo es estudiar la naturaleza de estos clusters secundarios de baja energía, caracterizando cómo se distribuyen alrededor de los clusters de alta energía, analizando a qué distancia pueden extenderse y evaluando si el tamaño, la energía o la forma del cluster primario tienen alguna influencia en la producción de estos halos.

Este análisis permitirá desarrollar un modelo de limpieza de imágenes, cuyo objetivo es eliminar los eventos secundarios generados por los clusters de alta energía sin afectar a la detección de señales genuinas de baja energía. La implementación de este modelo ayudará a mejorar la precisión en la búsqueda de nuevas partículas y optimizará el rendimiento de los sensores utilizados en el experimento Moskita. Además, los resultados de este estudio pueden ser extrapolados a otros experimentos que empleen CCDs en la detección de partículas, proporcionando herramientas para la mejora en la calidad de los datos obtenidos en futuros proyectos de física experimental.

Palabras clave: materia oscura, Moskita, skipper CCDs, halo, CTI, clusters, detección, procesamiento y análisis de imágenes, ROOT, hot columns.

Abstract

This Bachelor's Thesis is dedicated to the study of one of the background sources present in the images collected by the Moskita prototype, located at CERN, 80 meters underground near the LHC ring. In particular, we focus on analyzing very low-energy depositions, which appear as small groups of activated pixels (referred to as clusters) surrounding the signals produced by the passage of particles in the charge-coupled device (CCD) sensors—a phenomenon we will refer to as the halo. CCDs are highly sensitive imaging devices, primarily used in astronomy to detect extremely faint signals in large telescopes. Since 2012, these sensors have been used in particle physics experiments aimed at searching for rare and low-energy events, such as dark matter particles, millicharged particles, or other exotic particles not predicted by the Standard Model of particle physics.

To extract useful information from the images recorded by the CCDs, a pre-processing must be applied, including pedestal subtraction, correction of hot pixels and hot columns, system calibration, and subsequently the reconstruction of energy depositions. In this process, both low- and high-energy clusters (groups of charge-associated pixels) are identified and characterized. The search for new particles focuses on finding low-energy clusters, as these are expected to result from very weak interactions with matter, leaving extremely low-energy depositions in the CCDs (below 20–30 eV). Therefore, accurately characterizing these faint depositions is crucial for the discovery of new particles. A major challenge in this search is that high-energy clusters can produce secondary low-energy clusters, which mimic the signals of interest and thus create a background that complicates data interpretation.

The objective of this work is to study the nature of these secondary low-energy clusters, characterizing how they are distributed around the high-energy clusters, analyzing how far they can extend, and assessing whether the size, energy, or shape of the primary cluster has any influence on the production of these halos.

This analysis will enable the development of an image-cleaning model aimed at removing secondary events generated by high-energy clusters without affecting the detection of genuine low-energy signals. The implementation of this model will help improve the precision in the search for new particles and optimize the performance of the sensors used in the Moskita experiment. Moreover, the results of this study may be extrapolated to other experiments using CCDs for particle detection, providing tools to improve data quality in future experimental physics projects.

Keywords: dark matter, Moskita, skipper CCDs, halo, CTI, clusters, detection, processing and analysis of images, ROOT, hot columns.

Agradecimientos

En estas líneas quiero agradecer a varias personas todo lo que me han ayudado en estos cuatro años de carrera. En primer lugar, quería agradecer a mis padres y a mi familia en general por todas las facilidades que me han dado para estudiar lo que he querido y siempre me han apoyado.
Por otro lado, quería agradecer a todos los profesores todo lo que me han enseñado a lo largo de la carrera. Tambien quiero mencionar a Rocío, la directora de este trabajo. Desde enero que empecé con este trabajo ha estado muy implicada en él. Me ha explicado con claridad todo lo que hemos ido haciendo. Además, siempre ha estado disponible para resolver cualquier tippo de duda/ atenderme (incluso los fines de semana).
Finalmente, quería hacer especial mención a Los chipis, el grupo de amigos que hemos formado a lo largo de la carrera. Han sido una compañia increible. Entre todos nos hemos ayudado unos a otros para llegar hasta aqui. Hemos pasado muchos momentos bonitos tanto en la universidad como fuera de ella.

Índice

1 Introducción teórica **1**
- 1.1 Materia oscura . . . 1
  - 1.1.1 Evidencias de materia oscura . . . 1
  - 1.1.2 Mecanismos de producción de materia oscura . . . 2
  - 1.1.3 Propiedades de la materia oscura . . . 3
  - 1.1.4 Candidatos a materia oscura . . . 4
  - 1.1.5 Detección de materia oscura . . . 6
- 1.2 El fenómeno del halo . . . 7
  - 1.2.1 Orígenes del halo . . . 8
  - 1.2.2 Características del halo . . . 9
  - 1.2.3 Impacto del halo en la detección de partículas . . . 10
- 1.3 Objetivos del trabajo . . . 10

2 Dispositivos de carga acoplada CCD **12**
- 2.1 Materiales semiconductores . . . 12
- 2.2 El diodo p-n y el capacitor MOS . . . 13
- 2.3 Estructura de un CCD y principios de funcionamiento . . . 13
- 2.4 Generación, difusión y colección de cargas . . . 14
- 2.5 Secuencia de transferencia de cargas . . . 15
- 2.6 Dispositivo de salida . . . 16
- 2.7 Skipper CCDs . . . 17

3 Procedimiento experimental **19**
- 3.1 Montaje experimental . . . 19
- 3.2 Adquisición de datos . . . 20
- 3.3 Procesamiento de datos . . . 21
- 3.4 Contenido de las imágenes a analizar . . . 22
- 3.5 Análisis de datos . . . 24
- 3.6 Función para detectar hot columns . . . 26

4 Resultados y análisis **29**
- 4.1 Estudio de la densidad de clusters de baja energía dentro de las cajas . . . 30
- 4.2 Cálculo de densidades de píxeles para anillos concéntricos . . . 31
- 4.3 Método alternativo para el cálculo de la densidad . . . 35
- 4.4 Caracterización de la forma de clusters energéticos . . . 39
- 4.5 Ajuste de clusters a una recta . . . 41

5 Conclusiones **46**

A Apéndices **51**

A.1 Variables utilizadas para el análisis de los clusters 51
A.2 Extensión de los estudios hecho a todos los datos analizados 52
A.3 Códigos desarrollados para este trabajo 58
A.3.1 Función para detectar hot columns (sección 3.6) 59
A.3.2 Código empleado en la sección 4.1 61
A.3.3 Código empleado en la sección 4.2 65
A.3.4 Código empleado en la sección 4.3 68
A.3.5 Código empleado en la sección 4.4 71
A.3.6 Código empleado en la sección 4.5 72

1 Introducción teórica

1.1. Materia oscura

La materia oscura es uno de los grandes enigmas de la física moderna. Sabemos que aproximadamente el 27 % del contenido total del universo está formado por este tipo de materia, ya que interactúa gravitacionalmente y su presencia es necesaria para explicar observaciones astronómicas y cosmológicas, como la rotación de las galaxias o la formación de estructuras a gran escala. Sin embargo, no sabemos cuál es su naturaleza. Sabemos que no puede estar formada (al menos en su mayoría) por partículas del Modelo Estándar, que no interactúa con la luz (por lo que no emite, absorbe ni refleja radiación electromagnética), que es neutra y que su vida media debe ser superior a la edad del universo. Aunque la hipótesis mayoritaria sostiene que existe una forma de materia aún no detectada, algunos investigadores (cada vez menos) plantean que quizá el problema esté en nuestra comprensión de la gravedad y que sería necesario modificar la teoría de la relatividad general propuesta por Einstein [3].

1.1.1. Evidencias de materia oscura

A pesar de que la materia oscura no se ha detectado de forma directa, hay evidencias de ella a partir de observaciones cosmológicas y astronómicas. La existencia de la materia oscura sirve para explicar el comportamiento de galaxias, cúmulos de galaxias, estructura a gran escala del universo, etc.

- Observaciones astromómicas: uno de los primeros indicios de la existencia de la materia oscura provino del astrónomo suizo Fritz Zwicky en la década de 1930. Zwicky estaba estudiando el Cúmulo de Coma, un grupo de galaxias situadas a aproximadamente 320 millones de años luz de la Tierra. Aplicando el teorema del virial, que establece una relación entre la velocidad de los objetos en un sistema gravitacionalmente ligado y la masa total del sistema, Zwicky descubrió que las galaxias del cúmulo se movían a velocidades mucho mayores de lo esperado si solo se tomaba en cuenta la masa visible. Este fenómeno implicaba que debía haber una gran cantidad de masa no visible ejerciendo una atracción gravitacional adicional sobre las galaxias. Zwicky llamó a esta masa restante 'materia oscura' [3].

- Curvas de rotación galáctica: en la década de 1970, la astrónoma Vera Rubin, junto con Kent Ford y otros colaboradores, realizó estudios detallados de la velocidad de rotación de las estrellas en galaxias espirales. Según la física clásica, en un sistema donde la masa está concentrada en el centro (como en el Sistema Solar), la velocidad de rotación de los objetos debe disminuir a medida que aumenta la distancia desde el centro, siguiendo la ley de Kepler: $v = (GM/R)^{0,5}$. Lo que descubrió Rubin fue que, en muchas galaxias espirales, la velocidad de las estrellas permanecía constante en las regiones más lejanas de la galaxia. En estas regiones, la cantidad de materia visible debería ser muy pequeña. Este fenómeno solo se puede explicar si existiera un halo de materia oscura que extendía su influencia gravitacional más allá de la materia visible [3].

- El Cúmulo Bala: uno de los ejemplos más impresionantes de la existencia de la materia oscura proviene del estudio del Cúmulo Bala, una colisión entre dos cúmulos de galaxias que ocurrió hace aproximadamente 150 millones de años. Este evento cósmico ha proporcionado evidencia directa de la existencia de la materia oscura. En primer lugar, el gas caliente en el cúmulo emite rayos X, lo que representa la mayor parte de la materia visible. Durante la colisión, el

gas de los dos cúmulos interactuó y se desaceleró debido a la fricción. Por otro lado, la masa total del sistema, determinada por el efecto de lentes gravitacionales (la distorsión de la luz de galaxias de fondo causada por la gravedad del cúmulo), mostró que la mayor parte de la masa no se encuentra en el gas caliente visible, sino en una componente invisible que se localiza por delante y por detrás del gas, en la dirección donde se produjo el choque. Esto indica que esta materia invisible apenas interactúa, ya que sigue su camino casi sin frenarse. Mucha gente lo considera una prueba directa de existencia de materia oscura [3].

- Radiación Cósmica de Fondo (CMB): la radiación cósmica de fondo de microondas (CMB) es la luz más antigua del universo, emitida aproximadamente 380 000 años después del Big Bang. Esta radiación proporciona una 'fotografía' del universo primitivo y revela fluctuaciones de densidad que dieron origen a las galaxias y cúmulos de galaxias actuales. Los datos obtenidos por los satélites COBE, WMAP y Planck han confirmado que la cantidad de materia oscura en el universo es mucho mayor que la de la materia ordinaria. Sin la influencia gravitacional de la materia oscura, las estructuras a gran escala del universo no podrían haberse formado en el tiempo observado [3].

1.1.2. Mecanismos de producción de materia oscura

La materia oscura se pudo haber generado en las primeras etapas del universo por diversos mecanismos físicos. Estos mecanismos pueden tratarse de interacciones con partículas del modelo estándar, desintegración de partículas masivas, transiciones de fase más allá del modelo estándar, etc.

- Freeze-out: las partículas de materia oscura se encontraban en equilibrio térmico con el plasma de partículas en los comienzos del universo. En esa etapa, la densidad de partículas era alta y las colisiones entre ellas eran muy frecuentes. Las WIMPs (Partículas Masivas Débilmente Interactivas), que interactúan mediante la fuerza nuclear débil, podían aniquilarse entre sí y producir partículas del Modelo Estándar, y viceversa, según la reacción

$$\chi + \overline{\chi} \leftrightarrow f + \overline{f},$$

donde χ representa (anti)partículas de materia oscura y f las (anti)partículas del Modelo Estándar.

Este equilibrio se mantenía mientras la tasa de aniquilación era mayor que la tasa de expansión del universo. Sin embargo, a medida que el universo se expandía y enfriaba, la densidad de WIMPs disminuía, y la tasa de colisiones caía por debajo de la tasa de expansión. En ese momento, las aniquilaciones dejaban de ser eficientes, y el número de partículas de materia oscura 'se congelaba', es decir, quedaba prácticamente constante. Este fenómeno se conoce como *freeze-out*.

El número final de partículas de materia oscura que quedó en el universo depende de cuán fácilmente podían aniquilarse entre sí, es decir, de la sección eficaz de aniquilación de las WIMPs. Si esta sección eficaz tiene un valor típico de aproximadamente $\langle \sigma v \rangle \approx 3 \times 10^{-26}$ cm^3/s, se obtiene una cantidad de materia oscura que coincide con la que observamos hoy en día en el universo. Este resultado se conoce como la 'coincidencia WIMP' y es uno de los motivos principales por los que se piensa que las WIMPs son buenas candidatas a formar la materia oscura [3].

- Freeze-in: existe la posibilidad de que nunca haya existido equilibrio térmico entre la materia oscura y el plasma primordial. En este escenario, la producción de materia oscura ocurre

a través de interacciones extremadamente débiles con las partículas del Modelo Estándar. Por ejemplo, partículas masivas del Modelo Estándar pueden decaer en otras partículas del modelo, generando materia oscura como subproducto. Sin embargo, debido a que la materia oscura no interactúa de forma apreciable con el plasma primordial, su abundancia nunca llegó a equilibrarse, sino que se fue acumulando lentamente en pequeñas cantidades a lo largo del tiempo. Este mecanismo contrasta con el freeze-out, donde la materia oscura sí alcanzó el equilibrio térmico antes de desacoplarse [3].

Como se muestra en la Figura 1.1, en el caso del freeze-out (curva azul), la materia oscura estuvo inicialmente en equilibrio térmico con el plasma primordial, y al bajar la temperatura y disminuir la tasa de colisiones, su abundancia se congeló en un valor constante. En cambio, en el mecanismo freeze-in (curva verde), la materia oscura nunca llegó a estar en equilibrio térmico: su producción se fue acumulando lentamente a través de interacciones muy débiles con las partículas del Modelo Estándar, sin alcanzar nunca el equilibrio. Esta diferencia fundamental se refleja claramente en la evolución de la abundancia a lo largo del tiempo.

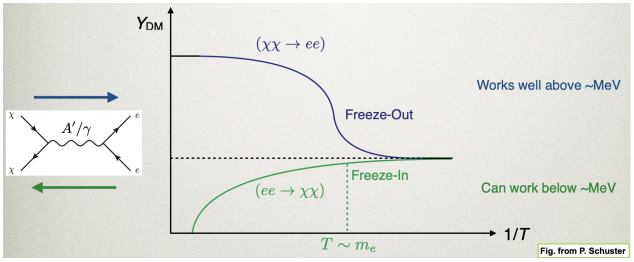

Figura 1.1: Representación comparativa de los mecanismos freeze-out y freeze-in para la producción de materia oscura. La curva azul muestra el caso freeze-out, mientras que la curva verde representa el freeze-in [4]

1.1.3. Propiedades de la materia oscura

Aunque existen muchos modelos que predicen diferentes candidatos adecuados para la materia oscura, la gran mayoría coincide en ciertas propiedades.
A partir de las observaciones astrofísicas y cosmológicas, sabemos que la materia oscura es una componente esencial del universo, pero sus propiedades son muy distintas de las de la materia ordinaria. La principal característica que la define es que no interactúa de forma significativa con la radiación electromagnética: no emite, no absorbe ni refleja luz en ninguna longitud de onda (visible, infrarroja, rayos X, etc.), lo que explica por qué no puede detectarse con telescopios u observatorios tradicionales.
Sin embargo, aunque no interactúe electromagnéticamente, la materia oscura sí ejerce influencia gravitacional, lo que permite inferir su existencia a través de efectos como la curvatura de la luz (lentes gravitacionales), la rotación de las galaxias o las fluctuaciones en la radiación cósmica de fondo.
Además, en muchos modelos se asume que, aunque sea neutra eléctricamente, la materia oscura podría interactuar débilmente con las partículas del Modelo Estándar o consigo misma, a través de nuevas fuerzas o mediadores, dando lugar a posibles señales detectables en experimentos directos, indirectos o en colisionadores. Estas posibles interacciones son la base de los esfuerzos experimentales actuales para tratar de identificar su naturaleza. Por otro lado, el rango de las masas de materia oscura es muy amplio. Los candidatos a materia oscura pueden tener desde masas de 10^{-22} eV hasta 10^{66} eV [3].

A diferencia de la materia ordinaria, la materia oscura se distribuye de forma difusa en el espacio, formando halos gravitacionales extendidos alrededor de las galaxias. Estos halos proporcionan la masa adicional necesaria para explicar la dinámica de rotación galáctica y no son estructuras compactas, sino regiones amplias donde la densidad de materia oscura disminuye progresivamente a medida que nos alejamos del centro de la galaxia [3].

1.1.4. Candidatos a materia oscura

Existen varios candidatos a partículas de materia oscura propuestos por modelos más allá del estándar. De manera muy breve repasamos los más populares.

- **WIMPs**: son uno de los candidatos a materia oscura más estudiados. Se trata de partículas hipotéticas que, en los modelos más comunes, tienen una masa del orden de entre 1 GeV y varios TeV. Son estables o extremadamente longevas, con un tiempo de desintegración mayor que la edad del universo, y presentan una interacción muy débil con la materia: interactúan con la materia ordinaria solo mediante la gravedad y, en muchos modelos, a través de la fuerza nuclear débil (aunque algunos escenarios teóricos exploran otras posibles interacciones). Estas características explican por qué no las detectamos fácilmente en los experimentos.

  Los WIMPs son la candidata más popular a materia oscura porque surgen de forma natural en muchos modelos de física más allá del Modelo Estándar, como la supersimetría (SUSY) o las teorías de dimensiones extra. Además, su densidad reliquia, predicha por el mecanismo de freeze-out como se ha explicado anteriormente, coincide notablemente con la cantidad de materia oscura observada en el universo, un resultado conocido como el milagro WIMP (WIMP miracle) [3].

- **Materia oscura liviana**: este término abarca una variedad de candidatos más ligeros que los WIMPs, desde fermiones estériles en el rango keV–GeV hasta partículas ultraligeras como los axiones (10^{-22} eV). La ausencia de señales de WIMPs ha motivado a explorar nuevos modelos teóricos, incluyendo interacciones mediadas por portales como el fotón oscuro (portal vectorial) o el bosón de Higgs oscuro (portal escalar), que permiten conexiones débiles entre el sector oscuro y el Modelo Estándar. Estos modelos plantean mecanismos de producción distintos al freeze-in y freeze-out, y algunos sugieren un exceso inicial de partículas sobre antipartículas (materia oscura asimétrica), lo que explicaría la supervivencia de una fracción relicta.

  A nivel experimental, la búsqueda de materia oscura liviana ha impulsado desarrollos como haloscopios (ej. ADMX) para detectar axiones, detectores ultrasensibles capaces de registrar interacciones electrónicas (como SENSEI y DAMIC), y sensores cuánticos que exploran señales extremadamente débiles. En estos escenarios, las partículas de materia oscura liviana podrían interaccionar preferentemente con los electrones del detector, generando señales similares a procesos conocidos como el efecto fotoeléctrico o el efecto Compton, lo que abre nuevas oportunidades para su detección [3].

  - **Axiones**: son partículas hipotéticas muy ligeras cuyas masas son del orden de $10^{-5}\ eV$. Fueron propuestos originalmente no para explicar la materia oscura, sino para resolver un problema específico en física de partículas: el problema CP fuerte en la cromodinámica cuántica (QCD), la teoría que describe la interacción fuerte. En principio, QCD permite una violación de la simetría CP (carga-paridad), lo que debería generar un momento dipolar eléctrico del neutrón; sin embargo, este efecto no se observa experimentalmente.

En 1977, Peccei y Quinn propusieron una solución introduciendo una nueva simetría (la simetría PQ), que al romperse espontáneamente genera una nueva partícula: el axión.

Los axiones cumplen con todas las características necesarias para ser materia oscura: son muy ligeros, tienen interacciones extremadamente débiles, son estables (o poseen vidas medias más largas que la edad del universo) y se producen de manera no térmica en el universo temprano. Por todo esto, son excelentes candidatos a materia oscura fría, aunque su mecanismo de producción es diferente al de los WIMPs [3].

- **Fotones oscuros**: son partículas hipotéticas que actuarían como análogos del fotón, pero en un sector oscuro: un conjunto de partículas e interacciones que no forman parte del Modelo Estándar, pero que podrían existir en la naturaleza e interactuar con la materia oscura. Este sector oscuro estaría prácticamente desacoplado del sector visible (las partículas que conocemos), pero podría comunicarse muy débilmente a través de un parámetro conocido como mezcla cinética ϵ_k.

  Teóricamente, los fotones oscuros aparecen en modelos que extienden el grupo gauge del Modelo Estándar incorporando una nueva simetría U(1)'. Esta simetría da lugar a una partícula mediadora —el fotón oscuro— que puede mezclarse cinéticamente con el fotón ordinario, permitiendo así una interacción extremadamente débil entre ambos sectores. Según los parámetros del modelo, los fotones oscuros podrían haberse producido de manera térmica o no térmica en el universo temprano, convirtiéndose en candidatos interesantes para explicar parte de la materia oscura [11].

- **Portales oscuros**: no son en sí mismos candidatos a materia oscura, sino mecanismos teóricos que permiten que las partículas del sector oscuro interactúen, aunque muy débilmente, con las del sector visible (las del Modelo Estándar). Estos portales proporcionan un marco para conectar ambos sectores y, lo más importante, abren la posibilidad de buscar señales experimentales de materia oscura a través de sus interacciones indirectas con partículas conocidas.

  Existen varios tipos de portales oscuros, cada uno asociado a posibles partículas mediadoras nuevas:

  - **Portal escalar**: conexión mediante el campo de Higgs.
  - **Portal vectorial**: asociado al fotón oscuro.
  - **Portal neutrino**: basado en neutrinos estériles que no interactúan con las fuerzas del Modelo Estándar pero que podrían mezclarse con los neutrinos activos y ser considerados candidatos a materia oscura, un tipo especial de materia oscura caliente.
  - **Portal axiónico**: relacionado con los axiones y partículas similares (axion-like particles).

  La ventaja de estos portales es que ofrecen estrategias teóricas y experimentales para buscar candidatos a materia oscura no solo por su existencia directa, sino a través de las interacciones sutiles que tendrían con el Modelo Estándar [12].

- **Materia oscura antisimétrica**: es un modelo que propone que, al igual que ocurrió con la materia ordinaria, existió una asimetría entre partículas y antipartículas de materia oscura en el universo temprano. Por algún mecanismo (como violación de CP, procesos fuera de equilibrio térmico, entre otros), se generó un exceso de partículas de materia oscura sobre antipartículas, evitando así la aniquilación completa y dejando como remanente una fracción que constituye la densidad actual de materia oscura.

Este enfoque es especialmente interesante porque explica de forma natural la coincidencia en las densidades entre la materia ordinaria y la materia oscura: si ambos sectores tienen un origen común, sus abundancias estarían relacionadas. Esta relación es difícil de justificar en ptros modelos de materia oscura como los de WIMPs, donde las densidades relictas se calculan de manera independiente. Además, este modelo sugiere que la materia oscura podría estar relacionada con la física de neutrinos y que su masa estaría en el rango de GeV, similar a la de los protones y neutrinos [15].

1.1.5. Detección de materia oscura

Dadas las características de la materia oscura descritas en la sección anterior —en particular, que no interactúa electromagnéticamente, no emite ni absorbe luz y apenas interacciona con la materia ordinaria— su detección resulta un enorme desafío experimental. Aun así, si queremos tener alguna esperanza de detectarla, debemos asumir que, de alguna forma, interactúa con las partículas del Modelo Estándar, aunque sea a través de interacciones muy débiles.
A partir de esto, los científicos han desarrollado varias estrategias de búsqueda, que dependen tanto de cómo pudo haberse producido la materia oscura en el universo temprano como de cómo podría interactuar hoy en día. Estas estrategias se dividen, de forma general, en tres enfoques complementarios: la detección directa, la detección indirecta y la producción en aceleradores. En la figura 1.2 se pueden observar los diferentes mecanismos de detección empleados para la búsqueda de materia oscura.

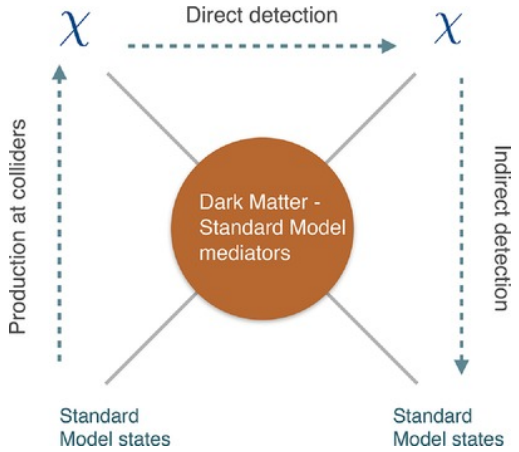

Figura 1.2: Esquema de los principales enfoques experimentales en la búsqueda de materia oscura [23]

- **Detección directa**: se basa en la búsqueda de interacciones entre partículas de materia oscura y el material blanco u objetivo. Estas interacciones pueden producirse a través de distintos mecanismos, dependiendo de si la materia oscura interactúa principalmente con los núcleos atómicos o con los electrones.

  En el caso de la interacción con núcleos, la partícula de materia oscura transfiere parte de su momento al núcleo mediante una colisión elástica, lo que genera un retroceso nuclear detectable. Alternativamente, si la interacción se produce con los electrones, puede generarse una excitación o ionización del átomo. Estas señales pueden manifestarse en el detector como ionización (si se liberan cargas eléctricas), centelleo (si se emiten fotones), o fonones/señales térmicas (si se generan vibraciones en la red cristalina del material) [3].

En la figura 1.3 se muestran los diferentes tipos de detectores empleados en la búsqueda directa de materia oscura, organizados según el tipo de señal que registran: fonones (calor), carga eléctrica o luz. Dependiendo del material utilizado y del diseño experimental, los detectores pueden estar optimizados para medir una o varias de estas señales simultáneamente. Por ejemplo, los bolómetros criogénicos detectan incrementos extremadamente pequeños de temperatura causados por los retrocesos nucleares (fonones), mientras que los detectores de germanio se centran en registrar la ionización generada por las interacciones. Los detectores de cristales centelleadores y de gases nobles líquidos son capaces de detectar fotones emitidos tras la excitación de los átomos, produciendo una señal luminosa (centelleo).

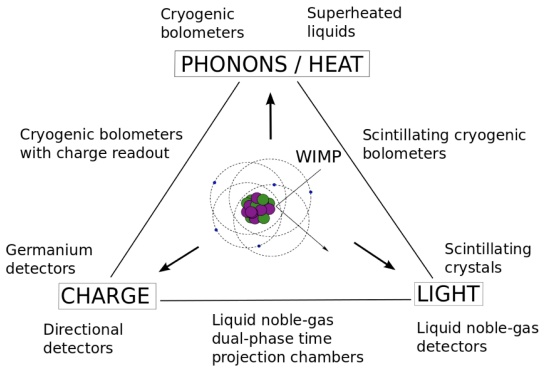

Figura 1.3: Esquema de los diferentes tipos de detectores utilizados en la búsqueda directa de materia oscura, clasificados según el tipo de señal que registran: fonones o calor, carga eléctrica o luz [22].

En este trabajo, nos centraremos específicamente en la detección directa a través de la señal de ionización, medida mediante dispositivos de carga acoplada (CCDs). Estos sensores permiten detectar con gran precisión las cargas liberadas por las interacciones entre partículas de materia oscura y electrones del material, y representan una tecnología muy prometedora para la exploración de materia oscura de baja masa.

- **Detección Indirecta**: busca señales de aniquilación o desintegración de materia oscura en partículas del modelo estándar (rayos gamma, positrones, neutrinos) provenientes de regiones con alta concentración de materia oscura, como el centro de la galaxia. Los experimentos con telescopios tanto en la Tierra como en el espacio son los más importantes en este ámbito [3].

- **Detección en Colisionadores**: en experimentos como los del Gran Colisionador de Hadrones (LHC), se busca producir partículas de materia oscura mediante colisiones de partículas del Modelo Estándar, en un proceso similar al que pudo ocurrir en el origen del universo a través de mecanismos como freeze-in o freeze-out. Dado que la materia oscura es estable, de larga vida media y con una probabilidad de interacción extremadamente baja, no deja señales directas en los detectores. Su presencia se deduce a partir de un desbalance en el momento y la energía total tras la colisión, lo que constituye una posible señal de su producción [3].

1.2. El fenómeno del halo

Antes de describir el fenómeno del halo, es importante mencionar que este efecto se manifiesta en dispositivos conocidos como Charge-Coupled Devices (CCDs), ampliamente utilizados en física de partículas y astrofísica por su alta sensibilidad y resolución espacial. En el capítulo 2 se presentan

en detalle las características físicas y operativas de estos detectores.

Las imágenes registradas por los CCDs contienen no solo las señales de interés (como los eventos generados por el paso de partículas), sino también diversos fondos (backgrounds) que pueden interferir en la interpretación de los datos. Entre los principales fondos que se encuentran en las imágenes destacan el ruido electrónico, los hot pixels, las columnas defectuosas y los patrones de reseteo, entre otros (además de la interacción de partículas ionizantes con la CCD).

De particular interés para este trabajo es el llamado halo, un fondo asociado a la aparición de una distribución extendida de carga alrededor de eventos de alta energía. Este fenómeno afecta directamente a la reconstrucción y análisis de señales en los experimentos, y resulta especialmente relevante cuando se busca detectar eventos de muy baja energía, donde la precisión es fundamental.

1.2.1. Orígenes del halo

El halo se genera principalmente debido a la dispersión y difusión de carga dentro del material semiconductor del detector. Cuando una partícula de alta energía interactúa con el CCD, deposita energía en forma de pares electrón-hueco. En una situación ideal, estos pares quedarían confinados en un área reducida del detector, registrándose como un único evento. Sin embargo, diversos procesos físicos pueden hacer que la carga se extienda a píxeles cercanos, formando un patrón difuso alrededor del evento central [15].

Las principales causas de la formación del halo incluyen:

1. **Generación de múltiples clústers**: cuando partículas de alta energía interactúan con el detector, pueden producir varios clústeres debido a procesos como la dispersión Compton o el efecto Cherenkov, que generan múltiples depósitos de energía en distintas zonas del sensor. Los clústeres de alta energía son fáciles de identificar debido a la gran cantidad de señal que generan, pero los de baja energía, que son clave en la búsqueda de partículas exóticas como la materia oscura, son más difíciles de detectar. Esta dificultad se debe a la presencia del halo, que introduce señales residuales que pueden enmascarar o distorsionar los clústeres de baja energía [3].

2. **Dispersión Compton:** en interacciones de alta energía, los fotones incidentes pueden transferir parte de su energía a electrones del material mediante dispersión inelástica (efecto Compton), expulsándolos con suficiente energía como para recorrer cierta distancia dentro del detector antes de ser detenidos. Este proceso contribuye a la formación de estructuras extendidas en la imagen [15].

3. **Efectos de saturación o desbordamiento de carga:** cuando se acumula una cantidad excesiva de carga en ciertos píxeles del CCD, por ejemplo, debido a eventos de muy alta energía o una alta densidad local de interacciones, se pueden superar los límites de almacenamiento de los pozos de potencial. Como resultado, parte de esa carga se "derrama" hacia píxeles adyacentes, generando una dispersión artificial de señal en la imagen (fenómeno conocido como blooming). Este efecto es especialmente problemático en entornos de detección de partículas con alta tasa de eventos, ya que puede dificultar la correcta reconstrucción de la señal original [3].

1.2.2. Características del halo

El fenómeno del halo en CCDs presenta una serie de características dependiendo de su origen y del tipo de interacción que lo genera. En primer lugar, cabe destacar que el halo no se distribuye de forma uniforme en el detector, sino que la forma de su distribución depende de la energía de las partículas incidentes, la profundidad de la interacción en el silicio y las propiedades de los CCDs. Normalmente, el halo se extiende de forma radial alrededor del evento principal. La intensidad decrece cuanto mayor sea la distancia al punto de interacción. Además, eventos que ocurren en la parte superficial del detector generan halos más compactos, a diferencia de los eventos que ocurren a mayor profundidad cuyos halos son más extensos [3].

Por otro lado, la forma del halo puede variar en función del tipo de interacción que lo origina. Se pueden clasificar en tres categorías principales. Una de ellas es el halo difuso. Se observa como una nube uniforme de carga alrededor del evento principal. Es el resultado de la difusión de electrones dentro del material y está influenciado por factores como la temperatura y las propiedades del silicio. Otro de ellos es el halo estructurado. Presenta patrones específicos, como trazas o extensiones asimétricas. Este tipo de halo se produce cuando los electrones secundarios tienen suficiente energía para desplazarse direccionalmente dentro del detector antes de ser recolectados. Por último, se encuentra el halo anisotrópico. En ciertos casos, el halo no es completamente simétrico, sino que adopta una forma dependiente de la geometría interna del CCD y la orientación de los campos eléctricos dentro del detector [3].

Además, la intensidad y el tamaño del halo están directamente relacionados con la energía depositada por la partícula incidente. En eventos de alta energía, el halo es más prominente y extenso debido a la mayor cantidad de carga generada y a la posibilidad de que electrones secundarios viajen mayores distancias antes de ser capturados. En eventos de baja energía, el halo es más débil y difícil de distinguir del ruido de fondo del detector, lo que puede afectar la identificación de señales de interés. En experimentos de búsqueda de materia oscura y partículas exóticas, los eventos de baja energía pueden quedar enmascarados por la señal residual del halo, lo que complica su análisis [3].

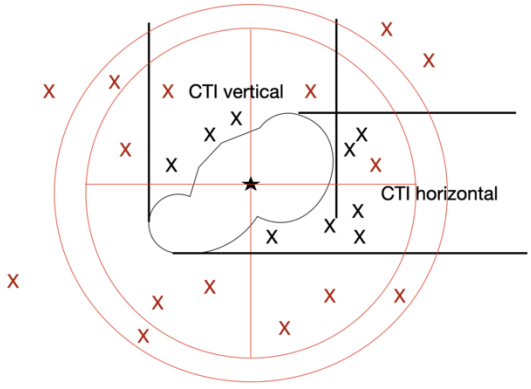

Figura 1.4: Esquema que detalla cómo se distribuye el halo alrededor del cluster

En la figura 1.4 se puede observar cómo el halo se distribuye alrededor del evento principal. El halo se corresponde con las 'X' rojas. Las 'X' negras se corresponde con el denominado CTI,

una inefiencia en la trasmisión de carga, ver mas detalles en el cápitulo 2. Lo que se quiere analizar en este trabajo es la region en la que aparece el halo que no está contaminado por CTI.

1.2.3. Impacto del halo en la detección de partículas

El halo es una fuente de ruido y contaminación en la detección de eventos en experimentos de física de partículas y astrofísica que usan CCDs. Su presencia afecta la interpretación de los datos, ya que distorsiona la información de eventos reales y puede generar falsas detecciones. A continuación, se detallan los principales efectos negativos del halo en la detección de partículas y las estrategias utilizadas para mitigar su impacto.

Uno de los problemas más importantes del halo es que puede ocultar eventos de baja energía, como aquellos que se buscan en experimentos de detección de materia oscura o neutrinos.

Por otro lado, el halo puede inducir señales espurias que podrían ser interpretadas erróneamente como eventos físicos reales. En algunas ocasiones, la carga dispersa del halo puede crear pequeñas fluctuaciones en la señal del detector, que podrían parecer eventos de baja energía. Esto es especialmente problemático en la búsqueda de partículas raras, donde la tasa de eventos es extremadamente baja, y cada detección es crítica. Los falsos positivos pueden llevar a resultados incorrectos y afectar la validez estadística de los datos obtenidos en un experimento.

El halo también puede alterar la cantidad de carga medida en los píxeles del detector, afectando la reconstrucción precisa de la energía depositada en cada evento. La carga que se dispersa fuera del cluster original modifica su perfil energético, lo que puede llevar a mediciones incorrectas. En experimentos donde se requieren mediciones extremadamente precisas —como en la búsqueda de nuevas partículas o en la detección de rayos cósmicos—, el halo introduce un fondo no deseado que limita la capacidad del experimento para identificar eventos genuinos. Esto puede provocar que un evento de alta energía parezca menos energético de lo que realmente es, o que se confunda con otro tipo de interacción.

En astrofísica, el halo puede afectar la observación de débiles fuentes de luz provenientes de objetos distantes, reduciendo la resolución de la imagen obtenida. En la detección directa de materia oscura, donde los eventos son extremadamente escasos, la contaminación del halo puede hacer que la señal de interés sea indistinguible del ruido del detector [3].

1.3. Objetivos del trabajo

Este trabajo tiene como objetivo principal estudiar el fenómeno del halo que aparece en las imágenes tomadas con sensores Skipper-CCDs en el experimento Moskita del CERN. Estos sensores se usan para detectar eventos de muy baja energía, que podrían estar relacionados con partículas nuevas como las de materia oscura. Sin embargo, cuando una partícula muy energética interactúa con el detector, puede dejar una señal muy intensa que se extiende alrededor de su traza principal. Esta señal adicional, conocida como halo, puede generar

clusters secundarios de baja energía que se parecen a los eventos que realmente nos interesa estudiar.

Es decir, el halo complica el análisis de las imágenes porque puede hacernos confundir eventos reales con señales falsas. Esto es un problema especialmente importante en la búsqueda de materia oscura, donde se esperan señales muy débiles y poco frecuentes.

Por eso, este trabajo busca entender cómo se forma el halo, a qué distancia pueden aparecer respecto al evento principal, y cuánta energía tienen (no nos hemos centrado demasiado en este aspecto). Para ello, se han aplicado diferentes métodos de análisis, que se describen en la sección de resultados, con el fin de estimar, sobre todo, la extensión del halo (no hemos tenido tiempo para mucho más).

La idea final es poder definir una región alrededor de los clusters energéticos que se pueda 'enmascarar', es decir, excluir del análisis para evitar errores. De esta forma, se limpiarán las imágenes y se mejorará la capacidad del experimento para detectar señales reales de baja energía. Además, los resultados de este trabajo podrían ser útiles en otros experimentos que también usen CCDs para detectar partículas.

2 Dispositivos de carga acoplada CCD

Los Dispositivos de Carga Acoplada (CCDs, Charge-Coupled Devices) son detectores de luz ampliamente utilizados en astronomía, microscopía, espectroscopía y otras aplicaciones de detección de imágenes. Los CCDs fueron muy comunes en la vida diaria, ya que a partir de los años 2000 fueron los sensores que usaban muchas cámaras digitales y los primeros teléfonos móviles con cámara. Gracias a ellos, se pudieron fabricar las primeras cámaras de vídeo digitales portátiles. Con el tiempo, sobre todo a finales de esa década, los sensores CCD empezaron a ser reemplazados por sensores CMOS (Complementary Metal Oxide Semiconductors), que son menos costosos a la hora de su fabricación, consumen menos energía y permiten tomar imágenes más rápido [14]. En los últimos años, se han empleado como detectores de particulas en la búsqueda de materia oscura. Esto se debe a su alta sensibilidad y bajo nivel de ruido. Funcionan mediante la captura y transferencia de carga eléctrica en una matriz de píxeles, permitiendo la conversión de señales ópticas en señales electrónicas [15].

Los CCDs están compuestos por un sustrato de silicio sobre el cual se fabrican estructuras de electrodos que controlan el almacenamiento y la transferencia de carga. Cuando los fotones inciden sobre la superficie del detector, generan pares electrón-hueco en el material semiconductor, donde los electrones quedan atrapados en pozos de potencial creados por los electrodos [15].

2.1. Materiales semiconductores

Los CCDs están basados en el uso de materiales semiconductores, principalmente el silicio, debido a sus propiedades electrónicas particulares. Un semiconductor es un material cuya conductividad eléctrica se sitúa entre la de los conductores (como los metales) y la de los aislantes (como el vidrio). Esta conductividad no es fija, sino que varía dependiendo de varios factores, especialmente la temperatura y las impurezas presentes en el material. Por ejemplo, al aumentar la temperatura, más electrones adquieren la energía suficiente para pasar a la banda de conducción, lo que incrementa la conductividad. Además, al introducir ciertas impurezas de forma controlada (un proceso llamado dopado), se pueden modificar las propiedades eléctricas del semiconductor, facilitando la creación de pares electrón-hueco cuando una partícula interactúa con el material. Esta capacidad para generar portadores de carga es clave para el funcionamiento de los CCDs como detectores de partículas o de luz [3].

- Banda prohibida: en los semiconductores, los electrones ocupan bandas de energía definidas, separadas por una banda prohibida (gap de energía). En el silicio, esta banda tiene un valor de aproximadamente 1.1 eV, lo que permite la generación de pares electrón-hueco cuando la energía de los fotones incidentes supera este valor [15].

- Dopaje: para mejorar la eficiencia en la captura y manipulación de carga, los CCDs cuentan con regiones dopadas con elementos como el fósforo (tipo n) o el boro (tipo p). Este dopado controlado permite la creación de uniones p-n, que son esenciales para definir los pozos de potencial donde se almacenan los portadores de carga (electrones o huecos) generados por la interacción con la radiación o partículas. Estas uniones permiten controlar el movimiento de la carga dentro del dispositivo mediante campos eléctricos, lo que hace posible su lectura y transporte secuencial a lo largo del CCD [3].

- Estructura cristalina: el silicio utilizado en los CCDs es monocristalino, lo que garantiza una distribución homogénea de los portadores de carga y reduce la presencia de defectos cristalinos que podrían atrapar electrones y dificultar la transferencia de carga. Además, este silicio está dopado de forma muy ligera, lo que le confiere una resistividad eléctrica muy alta (en el

caso de CCDs como los del experimento SENSEI, superior a $10,000\ \Omega cm$). Esta alta resistividad permite generar una región deplecionada amplia dentro del detector, lo que mejora la eficiencia en la recolección de carga, reduce el ruido electrónico y, en consecuencia, aumenta la sensibilidad del dispositivo para detectar señales extremadamente débiles. [3].

2.2. El diodo p-n y el capacitor MOS

El funcionamiento de los CCDs se basa en la física de los semiconductores, en particular en las propiedades del diodo p-n y del capacitor MOS (Metal-Oxide-Semiconductor) [15].
En un CCD, la región activa está formada por una unión p-n que permite la generación y recolección de cargas fotoinducidas. Cuando la luz (o una partícula que pueda producir ionización de alguna forma) incide sobre la superficie del semiconductor, los fotones generan pares electrón-hueco mediante efecto fotoeléctrico. Por un lado, los electrones se desplazan hacia la región n y los huecos hacia la región p, permitiendo el almacenamiento de carga en los pozos de potencial formados por los electrodos de control [15].
El almacenamiento y manipulación de la carga en un CCD se realiza mediante estructuras MOS, donde una capa de óxido de silicio actúa como aislante entre el electrodo de polisilicio y el sustrato semiconductor. Al aplicar diferentes voltajes a los electrodos, se regula el campo eléctrico en el semiconductor, permitiendo el confinamiento y transferencia de carga a lo largo del dispositivo [15].
En la Figura 2.1 se puede ver un esquema básico de un capacitor MOS (Metal-Oxide-Semiconductor), que es una parte esencial en el funcionamiento de los CCDs. Este tipo de estructura está formada por una capa metálica en la parte superior, una capa de óxido de silicio (que actúa como aislante) en el medio, y el silicio en la parte inferior. Cuando se aplica un voltaje al metal, se genera un campo eléctrico que afecta a las cargas eléctricas en el silicio. Esto permite crear zonas donde se pueden acumular, repeler o mover los electrones. Gracias a este efecto, los CCDs pueden controlar dónde se almacena la carga y cómo se va transfiriendo dentro del detector.

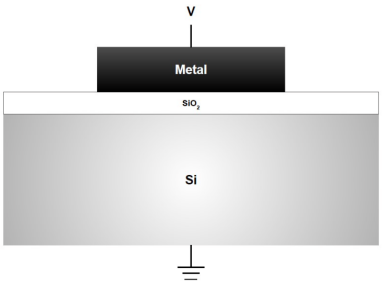

Figura 2.1: Esquema del capacitados MOS [15]

2.3. Estructura de un CCD y principios de funcionamiento

Cada píxel en un CCD está compuesto por una celda fotográfica de silicio, cubierta por una serie de electrodos dispuestos en capas de óxido de silicio y polisilicio. La función principal de estos electrodos es modular el potencial eléctrico para controlar la acumulación y el movimiento de carga dentro del dispositivo. Cuando una partícula o fotón interactúa con el silicio, genera una carga proporcional a la energía depositada, la cual se acumula en el píxel correspondiente. Posteriormente, mediante la acción secuencial de los electrodos, esta carga se desplaza de forma controlada a lo largo de la

matriz de píxeles hasta un amplificador de salida, donde se convierte en una señal eléctrica. Esta señal puede ser digitalizada y procesada para formar la imagen final. En los CCDs de tipo Full Frame (los empleados en este trabajo), toda la superficie del sensor está expuesta a la luz y la transferencia de carga se realiza después de la captura, mientras que en los CCDs de tipo Interline Transfer, cada fila de píxeles activos se intercala con filas de píxeles de almacenamiento que permiten transferir la carga sin perder datos por la exposición continua a la luz [3].
El tamaño de los píxeles, generalmente en el rango de micrómetros (entre 5 y 20 μm), afecta la capacidad de captación de luz y la resolución del sensor. Píxeles más grandes pueden capturar más fotones, mejorando la sensibilidad, mientras que píxeles más pequeños permiten una mayor resolución espacial. Los que se han usado en este caso tienen dimensiones 15 x 15 μm [3].

2.4. Generación, difusión y colección de cargas

Cuando la luz incide sobre el sensor, se generan pares electrón-hueco en el silicio. Los electrones libres se desplazan bajo la influencia del campo eléctrico interno del dispositivo. Este campo se establece gracias a un voltaje de polarización (bias) aplicado entre las regiones dopadas tipo p y tipo n, lo que crea una zona denominada región de depleción. En esta región, los portadores de carga móviles (electrones y huecos) son barridos, dejando una zona libre de carga donde las partículas generadas por la luz pueden moverse con mayor libertad. Cuanto mayor es el voltaje de polarización, más profunda es la región de depleción, lo que permite recolectar carga desde zonas más internas del silicio y mejorar así la sensibilidad del detector. Esta zona permite que las cargas se separen eficientemente: los huecos se recombinan o se desplazan fuera de la región activa, mientras que los electrones son dirigidos hacia los pozos de potencial formados en la interfaz de la estructura MOS. Es en estos pozos donde las cargas quedan temporalmente confinadas, a la espera de ser leídas. En la figura 2.2 se puede observar cómo las cargas generadas en el volumen del detector se difunden hacia la superficie [3].
Sin embargo, antes de ser completamente recogidos, los electrones pueden desplazarse lateralmente dentro del material. Este fenómeno, conocido como difusión de carga, ocurre porque los pares electrón-hueco generados en el volumen del silicio no son recolectados instantáneamente por los pozos de potencial de los píxeles. La cantidad de difusión depende de la profundidad en la que se generó la carga: cuanto más profunda sea la interacción dentro del silicio, mayor será la dispersión lateral de los electrones antes de ser capturados. Esto puede afectar a la precisión espacial con la que se localiza la señal, ya que la carga puede extenderse sobre varios píxeles, difuminando el evento original [3].
La cantidad de difusión de carga depende de varios factores, especialmente de las condiciones internas del CCD y del punto donde se genera la carga dentro del silicio. En general, cuanto más profundo sea el punto de interacción de la partícula o fotón en el sustrato, mayor será la distancia que los electrones deben recorrer hasta llegar al pozo de potencial, y por tanto, mayor será la dispersión lateral. Este efecto es responsable de patrones característicos en los eventos, como la forma de bate de béisbol que se observa en trayectorias de muones que atraviesan el detector.
Por otro lado, existen factores que pueden reducir la difusión. Uno de ellos es la profundidad del pozo de potencial: si el pozo es más profundo, los electrones quedan mejor confinados y se dispersan menos. También influye la intensidad del campo eléctrico en el sustrato: un campo más fuerte acelera a los electrones hacia el pozo, reduciendo el tiempo que tienen para difundirse y, por tanto, limitando su dispersión.
En cambio, otros factores favorecen una mayor difusión. Por ejemplo, los fotones más energéticos pueden generar electrones cerca de la superficie del silicio, donde el campo eléctrico es más débil,

lo que facilita su dispersión lateral antes de ser atrapados por el pozo de potencial [15].
La Figura 2.2 muestra un esquema del movimiento de la carga generada en el volumen del CCD. A medida que los electrones se desplazan hacia la superficie, pueden desviarse lateralmente debido a la difusión, lo que hace que la señal se reparta entre varios píxeles. Este fenómeno es especialmente relevante cuando la generación de carga ocurre en zonas profundas del silicio, ya que aumenta la probabilidad de dispersión antes de llegar al pozo de potencial.

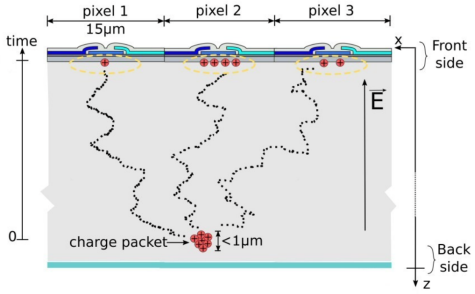

Figura 2.2: Imagen de cómo la carga generada en el volumen del detector se difunde hacia la superficie, dejando huella en píxeles más allá de la traza inicial [3].

2.5. Secuencia de transferencia de cargas

El proceso de lectura de un CCD se basa en la transferencia de carga a través de la matriz de píxeles mediante un sistema de potenciales eléctricos aplicado a los electrodos de control [3]. Este proceso se lleva a cabo en varias etapas:

1. Acumulación de carga: durante la exposición, la carga generada en cada píxel se almacena.

2. Transferencia de carga: en un CCD, la carga acumulada en cada píxel se desplaza aplicando voltajes oscilantes a los electrodos. Este desplazamiento se realiza en dos etapas: primero, la carga se mueve verticalmente a lo largo de las columnas hacia el registro de transferencia horizontal, un proceso relativamente lento. Luego, desde este registro, la carga se transfiere horizontalmente mucho más rápido hasta llegar al nodo de lectura (sense node), donde se convierte en una señal eléctrica [3]. Este proceso se repite para todos los píxeles, permitiendo leer toda la imagen secuencialmente. Esto se puede apreciar en la Figura 2.3 en la que se observa como se desplaza la carga hasta llegar al amplificador.

3. Conversión y amplificación: Una vez que la carga llega al registro de salida, es convertida en un voltaje proporcional a la cantidad de carga acumulada en el píxel. Este voltaje es leído por un amplificador y digitalizado para su procesamiento posterior [3].

El diseño del CCD debe minimizar la CTI (Charge Transfer Inefficiency), que representa la pérdida de eficiencia en la transferencia de carga entre píxeles. Este fenómeno ocurre cuando parte de la carga generada no logra llegar completamente al siguiente píxel durante el proceso de lectura. Una causa común de la CTI son las trampas electrónicas presentes en el sustrato de silicio, formadas por defectos cristalinos o impurezas, que capturan temporalmente los electrones y los liberan más tarde, generando una distorsión en la señal.
Sin embargo, la CTI no se debe exclusivamente a defectos del material. También puede aparecer si los voltajes de control o los tiempos de transferencia no están bien ajustados. Si el campo eléctrico

aplicado no es suficiente o si el tiempo asignado a cada paso de la transferencia es demasiado corto, parte de la carga puede quedarse atrás, provocando pérdidas o deformaciones en la imagen final [3].

2.6. Dispositivo de salida

El dispositivo de salida de un CCD es responsable de convertir la carga acumulada en cada píxel en una señal eléctrica legible. El componente clave de esta etapa es el Amplificador de Detección de Carga (CSA, Charge-Sensing Amplifier), que convierte la carga en un voltaje analógico [3],[15]. El proceso de salida de un CCD incluye:

1. Registro de salida: La carga es transferida secuencialmente desde los píxeles hasta un registro de desplazamiento [3].
2. Conversión de carga a voltaje: Un amplificador de conversión convierte la carga en un voltaje proporcional a la intensidad de la señal [3].
3. Digitalización: El voltaje es finalmente digitalizado mediante un Convertidor Analógico-Digital (ADC) para su almacenamiento y procesamiento [15].

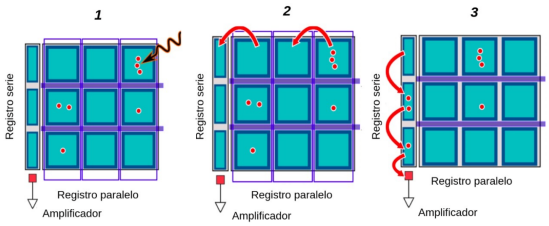

Figura 2.3: Imagen explicativa de como se realiza la lectura en un ccd. En primer lugar, la carga llega al detector. Luego, la carga se transfiere verticalmente hasta llegar a los registros en serie. Finalmente, la carga se transfiere horizontalmente en el registro en serie hasta llegar al amplificador [15].

Durante el proceso de lectura de un CCD, es habitual que se incluyan también píxeles adicionales que no forman parte del área activa de detección. Estos píxeles conforman las llamadas regiones de overscan, que se utilizan para estimar el valor del bias o pedestal, es decir, un valor fijo añadido por la electrónica para evitar que las señales sean negativas. Este valor se resta posteriormente a todos los píxeles para obtener una medida física real de la carga depositada. Por otro lado, el ruido electrónico corresponde a la dispersión estadística de los valores de estos píxeles en ausencia de señal, y representa la anchura del nivel cero tras corregir el bias [17]. Existen dos tipos principales: el overscan horizontal (x-overscan) y el overscan vertical (y-overscan), aunque su origen y comportamiento son distintos.
La región de overscan horizontal se obtiene al leer más columnas por fila de las que existen físicamente en el CCD. Estas columnas adicionales se generan de forma electrónica, por lo que no están asociadas a píxeles reales ni han estado expuestas a radiación. Debido a que solo registran el nivel de señal del sistema de lectura en ese momento, se espera que su valor represente únicamente el pedestal, salvo en casos excepcionales como eventos coincidentes con la lectura (por ejemplo, partículas que atraviesan el detector en ese instante). Esta información es fundamental para corregir el bias y mejorar la precisión de la medida de carga real en los píxeles activos.

En cambio, la región de overscan vertical se forma al realizar más ciclos de lectura vertical (filas) de los que realmente contiene el CCD. A diferencia del overscan horizontal, estas filas sí corresponden a píxeles físicos del sensor que se han desplazado desde la zona activa durante el proceso de lectura. Por ello, pueden haber estado expuestas a luz o a radiación a lo largo de su recorrido, especialmente en dispositivos con lecturas lentas, como los Skipper CCDs. En estos casos, los píxeles del y-overscan pueden contener cantidades de carga similares a los de la imagen, y se deben tener en cuenta al interpretar los datos [20].
En la Figura 2.4 se muestra un ejemplo de imagen CCD donde se visualizan claramente las regiones de *overscan* horizontal y vertical. Estas zonas se emplean habitualmente para corregir las imágenes antes del análisis físico, y son especialmente importantes en experimentos donde se requiere una resolución de carga a nivel de electrón individual.

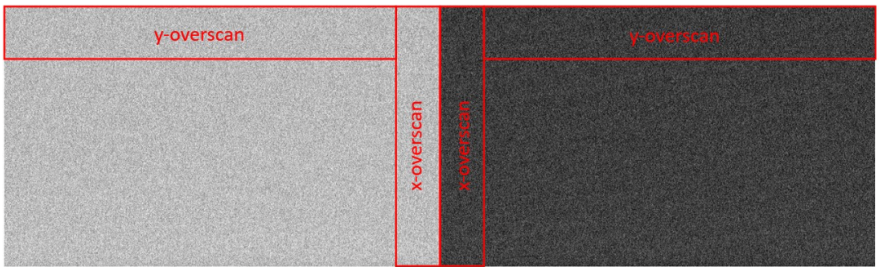

Figura 2.4: Esquema de una imagen de una ccd dividida en 2 porque cada una se ha tomado con un amplificador distinto. Se indica la región de overscan [17]

2.7. Skipper CCDs

Los Skipper CCDs representan una evolución de los CCDs tradicionales al permitir la lectura de carga con muy alta precisión, posibilitando el conteo de electrones de manera individual. Esta capacidad es crucial en experimentos de baja señal, como la detección de materia oscura, astronomía de alta precisión y espectroscopía.
La diferencia fundamental de los Skipper CCDs respecto a los CCDs convencionales es el método de lectura. En un CCD estándar, la carga acumulada en cada píxel se lee una única vez, lo que introduce un límite en la precisión debido al ruido de lectura del amplificador. En cambio, en un Skipper CCD, la carga en cada píxel puede leerse múltiples veces de manera no destructiva, promediando cada una de las mediciones tratándolas como si fueran independientes, reduciendo drásticamente el ruido. El ruido de lectura en una medición individual de un píxel se denomina σ_{pix}, y suele tener un valor entre 1.5 y 4 electrones, dependiendo del sistema. En los CCDs tipo Skipper, donde se realizan múltiples lecturas del mismo píxel, este ruido se reduce al promediar las lecturas. El valor resultante del ruido total es:$\sigma_{\text{lectura}} = \frac{\sigma_{\text{pix}}}{\sqrt{N}}$ donde N es el número de lecturas realizadas [3]. Este proceso permite medir cargas con una precisión de sub-electrón, mejorando la sensibilidad del detector. Para poder hacer esto, la estructura del nodo de salida del skipper CCD es distinta de la convencional. Los CCD tradicionales emplean una puerta de difusión flotante conectada mediante contacto óhmico al amplificador, los skipper emplean una puerta flotante aislada, permite devolver la carga al pozo de suma (summing well, SW) tras cada lectura. Esta propiedad evita la destrucción de la señal y repetir las medidas hasta alcanzar la resolución deseada (por debajo de un electrón) [3].
En la Figura 2.5 se muestra un esquema simplificado del proceso de lectura en un *Skipper CCD*, un tipo especial de CCD que permite realizar múltiples mediciones no destructivas de la misma carga. El sistema cuenta con varios elementos clave: **H** (Horizontal) son los electrodos que transfieren la

carga a lo largo del eje horizontal del CCD; **SW** (*Summing Well*) es un pozo temporal donde se acumula la carga antes de ser leída; **OG** (*Output Gate*) es la compuerta de salida que regula el paso de la carga desde el SW al nodo de lectura; el **Sense Node** es el nodo de lectura donde se mide la señal inducida por la carga. Se comporta como una puerta flotante, es decir, aislada eléctricamente, lo que permite conservar la carga tras la lectura; y **VR** (*Reset Gate* o *Voltage Reference*) es el nodo que restablece el nivel base de voltaje del sistema tras cada lectura.
El proceso de lectura ocurre en varias etapas. Primero, la carga se transfiere horizontalmente mediante los electrodos H hasta el pozo SW. Luego, la compuerta OG se activa para permitir que la carga pase al *sense node*, donde induce una variación de potencial que es registrada por el amplificador [17].

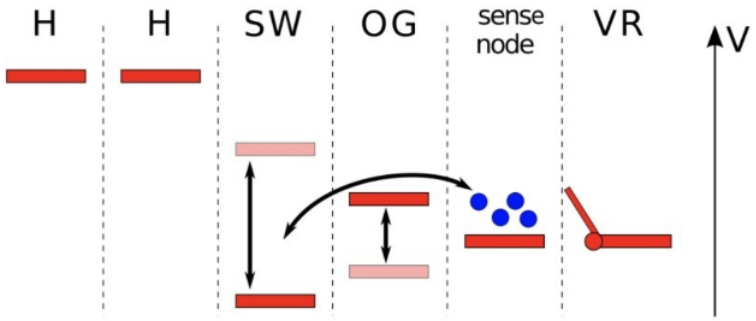

Figura 2.5: Esquema de como se realizan las medidas en el skipper CCD [17].

Los Skipper CCDs han sido empleados exitosamente en experimentos como SENSEI (Sub-Electron-Noise Skipper-CCD Experimental Instrument), donde su capacidad para medir eventos de un solo electrón ha permitido establecer límites sin precedentes en la búsqueda de interacciones débiles en el régimen de materia oscura [13] [19] [9].
Actualmente, esta tecnología también se está utilizando en el experimento DAMIC-M (Dark Matter in CCDs at Modane), en el que participamos activamente. DAMIC-M está diseñado para la búsqueda de materia oscura de baja masa, y utilizará cerca de 1 kg de Skipper CCDs, lo que lo convertirá en el experimento con la mayor masa activa basada en esta tecnología hasta la fecha. Su objetivo es alcanzar un nivel de fondo radiactivo inferior a 0.5 dru (eventos/kg/día/keV), lo que le permitirá explorar regiones del espacio de parámetros inaccesibles para otros detectores. Aunque DAMIC-M comenzará operaciones científicas hacia finales de este año, su prototipo LBC (Low Background Chamber) ya ha logrado establecer los límites más restrictivos hasta la fecha para interacciones de materia oscura con electrones, demostrando la viabilidad del diseño y el excelente control del fondo instrumental [8, 16].

3 Procedimiento experimental

El experimento Moskita está diseñado para la búsqueda de partículas con carga fraccionaria (millicharged particles, mCPs) mediante el uso de sensores Skipper-CCDs. Estas partículas no están contempladas en el Modelo Estándar de física de partículas, pero sí aparecen en algunas extensiones teóricas que introducen sectores ocultos o nuevas simetrías. En ciertos modelos, las mCPs también se proponen como posibles candidatas a materia oscura. El rango de masas de las particulas milicharge seria entre el MeV y el GeV en el haz del LHC. La existeencia de partículas milicharge podría ampliar el modelo estándar de física de partículas. Se han realizado experimentos previos como SENSEI, que han establecido límites de detección de mCPs en el rango de 30 MeV-380 MV. Sin embargo, la alta energía con la que ese experimenta en el LHC permite explorar masas superiores al GeV [5].

3.1. Montaje experimental

El experimento Moskita se encuentra instalado en la galería de drenaje de CMS en el LHC. Se encuentra a 70 metros bajo tierra y a 33 metros del punto de interacción en el LHC. Además, está protegido por 17 metros de roca desde el punto de interacción que está en el detector CMS. El hecho de que esté a 70 m bajo tierra permite una reducción del flujo de rayos cósmicos en un factor de 10 a 100. Por otro lado, los 17 metros de roca que rodean la zona del detector permiten atenuar la mayoría de las partículas del Modelo Estándar provenientes del punto de interacción, con excepción de partículas muy penetrantes como muones, neutrinos o posibles nuevas partículas. Aunque los muones que atraviesan la roca pueden ser identificados y vetados de forma eficiente, su interacción con el material circundante, como la roca de la galería de drenaje, puede generar partículas secundarias como fotones, neutrones y electrones, que contribuyen al fondo y deben ser tenidas en cuenta en el análisis. Además, la propia roca emite radiación, lo que genera un fondo radiativo no deseado que se puede evitar con un blindaje de unos pocos cm de plomo [5]. En la figura 3.1 se puede observar mejor dónde se sitúa el experimento en relación a CMS, lo que ayuda a visualizar su localización respecto al punto de interacción y el entorno geológico. En la Figura 3.2 se muestra el montaje experimental utilizado para la toma de las imágenes. En la parte inferior izquierda se observa la cámara de vacío, que contiene el detector y está rodeada de blindaje de plomo para reducir el fondo radiactivo ambiental. Justo por encima se encuentra el criocooler, encargado de mantener la temperatura adecuada para el funcionamiento de los sensores CCD. La cámara de vacío está conectada, en la parte inferior derecha, a una bomba de vacío, que permite alcanzar las condiciones necesarias para el correcto aislamiento del detector. En la parte superior del sistema se encuentra la unidad de lectura, que incluye el controlador de temperatura, el controlador de voltaje de polarización (bias voltage) y el controlador de lectura de las CCDs, donde se configuran los parámetros de adquisición.

El detector empleado consiste en un recipiente de acero inoxidable. Este acero está rodeado de un blindaje de plomo de 2 pulgadas. Este recipiente contiene un Skipper-CCD de 6,29 MPix (matriz de 1024 filas $\times$ 6144 columnas de pixeles de $15 \times 15 \mu m^2$) y un espesor de $675 \mu m$. Las condiciones en las que opera con las siguientes: temperatura de $140K$ (minimizando la carga generada térmicamente) y una presión de vacío de $10^{-4} torr$ [5].

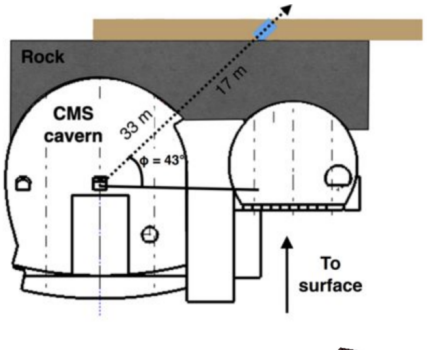

Figura 3.1: Esquema de dónde se encuentra el experimento moskita en relación con CMS. El rectángulo azul indica dónde se encuentra el experimento Moskita.

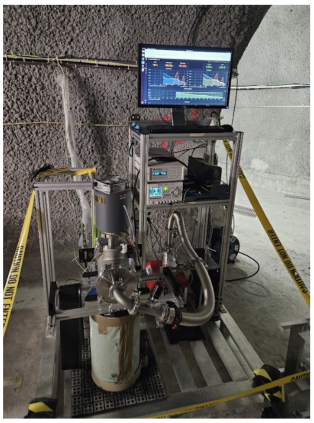

Figura 3.2: Imagen del detector empleado en el experimento Moskita [6]

Figura 3.3: Dibujo explicativo de la estructura del Skipper CCD empleado. Los cuadros rojos numerados se corresponden con los amplificadores. Las lineas azules se corresponden con los registros en serie. En gris se delimitan los cuadrantes [5].

El sensor consta de 4 amplificadores de lectura, uno en cada esquina. Cada amplificador se encarga de la lectura de un cuadrante. Por otro lado, los registros en serie son las últimas filas de píxeles en el sensor y su función es transferir la carga de los píxeles a los amplificadores, como se ha explicado en la sección 2.8. En la figura 3.3 se observa que los amplificadores 1 y 4 comparten un registro en serie, al igual que los amplificadores 2 y 3. A lo largo del experimento, no se utilizó el amplificador 1, ya que presentaba problemas desde las pruebas iniciales del sensor. Por otro lado, el registro en serie asociado al amplificador 2 mostraba una transferencia de carga ineficiente, lo que provocaba que parte de la señal quedara retenida en píxeles posteriores. Esto daba lugar a deposiciones de energía alargadas artificialmente, distorsionando así las trazas registradas por el CCD, lo que se conoce como CTI (charge transfer inefficiency) [5]. Por lo tanto, solo se utilizaron los datos correspondientes a los cuadrantes 1 y 2 de la figura 3.3.

3.2. Adquisición de datos

El proceso de adquisición de datos consta de dos fases. En primer lugar, antes de cada lectura científica, se realiza una fase de limpieza del sensor, en la que se transfiere y descarta la carga residual acumulada en el CCD. Esta operación, que dura aproximadamente 10 minutos, se realiza a través de los amplificadores 1 y 2. Su objetivo es eliminar cualquier carga remanente que pudiera haberse generado en exposiciones anteriores, tanto en los píxeles activos como en los registros en

serie. Esto garantiza que la imagen adquirida refleje únicamente los eventos nuevos, sin contaminación por señales previas. A continuación, el sensor se expone durante 1 segundo para capturar la señal, y luego comienza el proceso de lectura píxel a píxel utilizando los amplificadores mediante el procedimiento skippers explicado en el capitulo 2. Cada amplificador procesa 515 filas y 3200 columnas de píxeles. Este proceso de lectura, que es un proceso intensivo, tarda aproximadamente 3.5 horas en completarse [5].

Esta adquisición de datos se almacena en un archivo de formato FITS bidimensional [18]. Este archivo cuenta con 512 filas $\times$ 3200 columnas $\times$ N_{smp} donde N_{smp} representa el número de mediciones tomadas por píxel con el objetivo de lograr un ruido de lectura por debajo de un electrón (0.18). Las unidades de carga de los píxeles son ADUs (unidades analógico-digitales). El valor de N_{smp} ha variado a lo largo del tiempo: inicialmente, se tomaban 169 'samples' por medición, pero posteriormente se aumentó a 250 para mantener la misma resolución (por debajo de un electrón). Esto se debe a que, por alguna razón no identificada, el ruido de lectura ha ido aumentado. Aunque el tiempo total de exposición programado para el sensor es de 1 segundo, no todas las regiones de la CCD permanecen expuestas durante ese tiempo exacto. Debido al proceso de lectura secuencial, las filas de la CCD se van transfiriendo una a una hacia el registro de lectura. Esto implica que las primeras filas se leen antes que las últimas y, por tanto, dejan de estar expuestas antes, mientras que las últimas filas permanecen expuestas durante más tiempo. Como resultado, el tiempo efectivo de exposición puede variar ligeramente a lo largo del sensor. Cuando se aumenta el valor de `Nsmp`, el tiempo necesario para leer cada fila aumenta (aumenta el tiempo para cada pixel). Dado que la lectura del CCD se realiza fila por fila, desde la parte superior hacia la inferior, las regiones que se encuentran más abajo en el sensor permanecen expuestas durante más tiempo tras el final de la exposición principal. Esto ocurre porque esas filas aún no han sido leídas y siguen acumulando carga mientras se procesan las superiores. Como consecuencia, es posible que se registren más clústeres en las últimas filas que en las primeras, debido a esa exposición adicional no intencionada. Este efecto será visible en las imágenes analizadas en el capítulo de resultados [5].

3.3. Procesamiento de datos

Durante el procesamiento de datos, cada imagen pasa por una etapa inicial de preprocesado que tiene como objetivo limpiar la señal de cargas espurias y defectos instrumentales. En primer lugar, se sustrae el pedestal de la señal utilizando las regiones de overscan, lo que permite corregir el nivel de fondo electrónico. A continuación, se enmascaran los hot pixels y hot columns mediante el uso de mapas de píxeles defectuosos previamente identificados. También se excluyen zonas con alta densidad de carga y se eliminan los eventos que coinciden con trazas en el registro serial, ya que no corresponden a posiciones reales del sensor. Todo este proceso mejora significativamente la relación señal-ruido y proporciona una base robusta y fiable para la detección de eventos de baja energía.

En el experimento se generan dos tipos de archivos FITS. En algunos casos, se guarda una imagen que contiene todas las lecturas individuales realizadas para cada píxel, lo que da lugar a un archivo de dimensiones $N_{\text{col}} \times N_{\text{smp}} \times N_{\text{row}}$, donde N_{col} y N_{row} son el número de columnas y filas del sensor, y N_{smp} es el número de muestras por píxel. Estas imágenes más grandes permiten verificar la estabilidad de cada lectura individual, detectar posibles pérdidas o aumentos de carga, y comprobar que el sistema funciona correctamente.
A partir de estas, se genera una segunda versión más compacta, en la que se guarda únicamente el promedio de todas las lecturas por píxel. Esta imagen final tiene dimensiones $N_{\text{col}} \times N_{\text{row}}$, co-

rresponde al valor total de carga por píxel y se utiliza para el análisis principal. Al ocupar menos espacio y ser más fácil de manejar, esta forma es la que se emplea habitualmente para el tratamiento de los datos [5].
Además, se corrige el valor del pedestal introducido por la electrónica del sistema de lectura, de modo que el nivel base de los píxeles quede centrado en cero. Para corregir el pedestal introducido por la electrónica de lectura, se utiliza la región de overscan inverso, compuesta por píxeles que no pertenecen al área activa del detector y que, por tanto, no registran carga real. Estos píxeles solo contienen el nivel base del sistema y permiten estimar el valor del pedestal para cada fila. En concreto, se calcula la mediana de los valores en la región de overscan inverso de cada fila. Esta mediana representa el nivel de referencia de esa fila y se resta a todos los píxeles de la misma, eliminando así variaciones sistemáticas entre filas debidas a la lectura electrónica. Tras esta corrección, cualquier señal remanente en la imagen procesada puede atribuirse con mayor fiabilidad a eventos físicos reales [5].

3.4. Contenido de las imágenes a analizar

En las imágenes tomadas por las CCDs, se pueden observar lo que denominamos clusters. Un cluster se corresponde con un conjunto de píxeles contiguos que registran una señal significativa en el detector como resultado de la interacción de una partícula ionizante (generalmente) con el silicio del detector. Cuando una partícula atraviesa el detector, va depositando energía en el material generando pares electrón hueco (como se ha explicado previamente). La deposición de energía dentro del silicio sufre un proceso de difusión lateral, lo que provoca que la carga generada no quede confinada en un solo píxel, sino que se reparta entre varios. Este efecto da lugar a la formación de clústeres de señal en lugar de eventos puntuales [3].

Las características de los clústeres (forma, tamaño, energía, etc.) aportan información sobre la partícula que los ha producido, lo que permite identificar el tipo de interacción ocurrida en el detector. Cada tipo de partícula (o interacción) deja un tipo de señal característico. Por ejemplo, los muones (suelen ser de alta energía) atraviesan la CCD generando trazas rectas y alargadas. Los muones son partículas mínimamente ionizantes, con una pérdida de energía típica de aproximadamente $dE/dx \approx 3.9\,\mathrm{MeV/cm}$, lo que equivale a unas 80 cargas (electrones) por micra dentro del silicio. A lo largo de su trayectoria, apenas se desvían ni pierden energía de forma significativa, por lo que generan trazas rectas que atraviesan gran parte del detector. Estas trazas se extienden sobre muchos píxeles y presentan una carga distribuida de manera aproximadamente uniforme. Sin embargo, debido al efecto de difusión lateral, la forma de la traza no es completamente homogénea: suele observarse una morfología característica en forma de 'bate de béisbol', con secciones más estrechas cerca de la superficie del CCD y más anchas en profundidad, ya que la dispersión de carga aumenta con la distancia al pozo de potencial.
Por otro lado, las partículas alfa son más masivas y tienen mayor carga que los muones, lo que hace que depositen mucha energía en un trayecto corto. Esto genera trazas compactas, generalmente circulares, que ocupan pocos píxeles. Si la partícula alfa se origina en el volumen del silicio o desde la parte trasera del CCD, puede producir estructuras redondeadas de mayor tamaño debido al efecto plasma, el cual provoca una expansión lateral de la señal por la alta densidad de carga generada [10].
Las trazas generadas por partículas beta (electrones) presentan una gran variabilidad en forma y extensión. Su geometría suele ser irregular y la carga registrada en los píxeles puede estar distribuida de manera no simétrica. En electrones de alta energía, es común observar trazas alargadas con forma

de "gusano", donde la mayor parte de la energía se deposita hacia el final del recorrido, debido al pico de Bragg. En estos casos, el electrón pierde poca energía al inicio y la libera bruscamente al final, generando una concentración de carga en un punto [10].
En cambio, los electrones de baja energía (o los retrocesos nucleares producidos por partículas como neutrones o materia oscura) tienen una trayectoria física muy corta, inferior a 15 μm, y su señal se encuentra dominada por la difusión de carga. A estos se les conoce como clústeres limitados por difusión, ya que su forma está determinada más por el transporte lateral de la carga que por la trayectoria original de la partícula. En ocasiones, estos eventos pueden parecerse a los de partículas alfa de baja energía.
Por último, los fotones no ionizan directamente el material como las partículas cargadas, pero pueden generar electrones secundarios a través de procesos como el efecto fotoeléctrico, dispersión Compton o producción de pares. Los clusters resultantes tienden a ser más difusos, dependiendo del tipo y energía del fotón y del mecanismo de interacción. En la figura 3.4 se representan las diferentes trazas que se pueden generar en función de la partícula que interacciona con el detector [7].

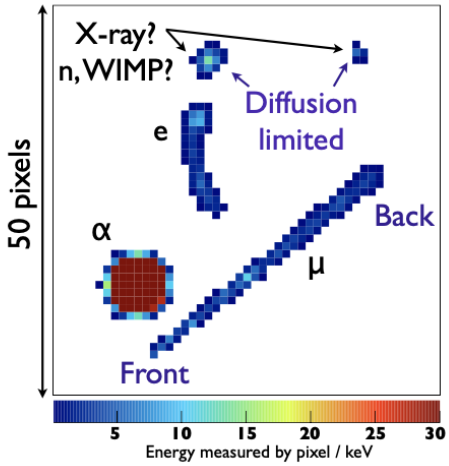

Figura 3.4: Distintas trazas que se pueden observar en la imagen de la CCD en función del tipo de partícula [7].

En el análisis de las imágenes obtenidas por los CCDs, pueden aparecer eventos no asociados a partículas, sino a efectos internos del detector, como defectos en la red cristalina o la corriente oscura. Esta última se debe principalmente a la generación térmica de pares electrón-hueco, aunque en algunos casos puede incluir contribuciones no térmicas relacionadas con defectos u otros procesos aún no completamente comprendidos. Un ejemplo de esto son los hot pixels. Un hot pixel es un pixel individual que presenta una señal muy destacada de forma persistente incluso en condiciones de nula luminosidad (ausencia de cualquier interacción física externa). Otros de los causantes pueden ser impurezas en el silicio que facilitan la generación de carga o que el silicio haya estado tanto tiempo expuesto que su estructura puede verse dañada debido a los distintos impactos recibidos.

Otro de los eventos más comunes en las imágenes de CCD son las hot columns. Estas columnas muestran señales persistentemente elevadas que no se corresponden con interacciones reales, y aparecen como agrupaciones verticales de píxeles con valores anómalos. Su origen puede estar en defectos estructurales repetidos a lo largo de la columna, en problemas del canal de transferencia vertical, o en hot pixels muy activos cuya carga se derrama hacia píxeles adyacentes. Al igual que los hot

pixels, las hot columns suelen deberse a defectos en la red cristalina del silicio, que generan carga espuria de forma continua. Este fenómeno se analizará en mayor detalle en la sección de resultados.

Otro fenómeno característico es el de los eventos en el serial register. Durante la lectura de un CCD, la señal se transfiere primero verticalmente, fila a fila, y luego horizontalmente a través del registro serial, que conduce la carga al nodo de lectura. Si una partícula (como un electrón o fotón) interactúa con la zona del serial register durante esta etapa final, puede generar carga directamente en esa región. Como resultado, aparecen clusters alineados horizontalmente, que se registran como señales reales pero no suceden dentro del área activa de la imagen. Este tipo de eventos puede confundirse con trazas físicas si no se identifica correctamente su origen. También pueden deberse a defectos locales en el propio registro serial.
Tanto las hot columns como los eventos en el serial register representan señales espurias que no corresponden a interacciones físicas reales en la región activa del detector. Estos artefactos pueden confundirse con eventos genuinos, afectando la interpretación de los datos y la búsqueda de señales raras, como las producidas por materia oscura. Por ello, es fundamental identificarlos correctamente y excluirlos del análisis para garantizar la calidad y fiabilidad de los resultados.

3.5. Análisis de datos

Los datos empleados para este análisis han sido tomados desde Abril 2024, hasta Diciembre 2024 de manera casi continua con paradas para reestablecer los parametros adecuados o por cambios del acelerador, cada uno de estos cambios se llamará *run*. En las tablas 3.1 y 3.2 se describen lo distintos runes y las características de cada uno. En este trabajo se van a analizar archivos de tipo ROOT [2]. ROOT es un paquete de análisis de datos desarrollado por el CERN para física de partículas. Está diseñado para manejar eficientemente grandes volúmenes de información, es de software libre y es compatible con Python. Las imágenes FITS contienen la información bruta leída del detector, donde cada píxel almacena un valor en ADU (Analog to Digital Units), que representa la señal digitalizada obtenida durante la lectura. A partir de los archivos FITS se construyen los archivos tipo ROOT.

En primer lugar, se seleccionan únicamente las imágenes adquiridas en modo Skipper, es decir, aquellas en las que se han realizado múltiples lecturas por píxel. A continuación, se realiza la conversión de los valores en ADU a número de electrones, dividiendo por la ganancia del sistema de lectura, la cual se determina previamente mediante un proceso de calibración.
Una vez calibradas las imágenes, se procede a la identificación y selección de clústeres. Para detectarlos, se utiliza un algoritmo que busca agrupaciones de píxeles contiguos cuya señal supere un cierto umbral. En este caso, se establece un umbral que varía de run a run, pero que en promedio está en torno a 0.75 electrones por píxel, lo cual permite contar electrones con alta eficiencia sin incluir demasiadas fluctuaciones del ruido. Este valor es inferior a 1 electrón no porque se midan fracciones de carga, sino porque el bajo ruido de lectura de los Skipper CCDs (por debajo de 0.2 e^- RMS) permite establecer umbrales con resolución mucho menor que la carga de un solo electrón. Así, por ejemplo, cuando se registra un valor cercano a 0.8 e^-, estadísticamente es probable que represente un único electrón, aunque en un pequeño porcentaje de los casos aún podría tratarse de ruido. Una vez identificados los píxeles que superan el umbral, se agrupan en clústeres. Para considerar un clúster como válido, se requiere que la suma total de la carga de todos sus píxeles sea superior a un valor mínimo definido, que depende del análisis (por ejemplo, 1.5 o 2 electrones), con el fin de evitar falsas detecciones debidas al ruido.

Tras aplicar los distintos pasos de preprocesamiento y selección con el software WADERS, se genera una nueva imagen en la que solo se conservan los píxeles pertenecientes a clústeres válidos. La información de estos clústeres se guarda en un archivo ROOT en formato TTree, una estructura de datos jerárquica optimizada para el análisis de grandes volúmenes de información. Esta información se almacena en los parámetros que se detallan en el apéndice A.1.

En el apéndice A.1 se detallan los diferentes parámetros, pero voy a hacer una breve explicación de los más importantes. En primer lugar, a cada cluster se le asigna un identificativo (cluster id). Para cada cluster, se almacenan su valor de NPix (el número de píxeles que componen el cluster); PosX y PosY (posición del centro de masa del cluster); QmaxX y QmaxY (posición del pixel con mayor carga del cluster); pixels-x y pixels-y (coordenadas de los píxeles que componen el cluster en los ejes x e y); minX, minY, maxX y maxY (coordenadas minima y máxima de los clusters en ambos ejes); pixels-E (energía de cada píxel del cluster); energy (energía total del cluster); has-seed (1 si tiene seed y 0 si no lo tiene, hace refeerencia a la carga necesaria para que se considere que tiene seed) y hotreg (1 si se encuentra cerca de una hot column y 0 en otro caso).

En cuanto a los archivos que se van a analizar, hay dos tipos. Uno de ellos son los archivos hadd. Estos archivos tienen varias entradas donde cada entrada se corresponde con una imagen tomada de los run. En concreto, los archivos hadd que vamos a analizar contienen todas las imágenes del run1 (el resto de los runes se muestran en el apendice A.2). Está el archivo hadd-ext2 y el hadd-ext3, correspondientes a cada uno de los cuadrantes que tienen las CCDs respectivamente. El otro tipo de archivos son los denominados run. Estos archivos tienen una única entrada correspondiente a la conjunción de todas las imágenes colocadas una detrás de otra. En estos run no se ha analizado el total de las imágenes ya que era imposible aplicar los códigos al ordenador, por lo que solo se han analizado 10 imágenes. Estos run almacenan imágenes tomadas en las mismas condiciones cada uno (voltaje, N_{smp}, fecha de comienzo y final de exposición...). Toda esta información se detalla en las tablas 3.1 y 3.2.

Run	**Inicio**	**Final**	**No. de imágenes**	**Condiciones**
1	Mar 15	May 22	462	Volt A; $N_{\text{smp}} = 169$
2	May 22	Jul 31	500	Volt B; $N_{\text{smp}} = 169$
3	Ago 04	Sep 14	293	Volt B; $N_{\text{smp}} = 169$
4	Sep 14	Oct 17	159	Volt B; $N_{\text{smp}} = 256$
5	Oct 19	Dic 09	236	Volt B; $N_{\text{smp}} = 256$
6	Dic 21	Dic 31	50	Volt B; $N_{\text{smp}} = 256$

Tabla 3.1: Resumen temporal de las imágenes adquiridas en cada run, junto con las condiciones de voltaje y número de muestras por lectura [5].

En la tabla 3.1, los voltajes A y B hacen reeferenica a los voltajes de operación empleados en cada run. En la tabla 3.2, la ganancia representa la proporción entre el valor digital leído (en ADUs) y la carga en electrones. Se obtiene ajustando el histograma correspondiente a la carga de los píxeles a una suma de 3 funciones gaussianas convolucionadas con una distribución de poisson para tener en cuenta la corriente oscura. Cada pico describe una gaussiana con 3 parámetros: a_n, σ_n y μ_n. La λ se obtiene del ajuste de la poissoniana y nos da el valor de la corriente oscura por imagen y por bin. Estas variables son la amplitud (número de píxeles que caen en cada pico), la desviación estándar (anchura del pico) y la media (posición en la que está centrada el pico. La ganancia se define como $G = \frac{\mu_2 - \mu_0}{2}$. Es la separación entre los picos de 0 y 1 e^- (es decir, cuántos ADUs se

Run	Ganancia [ADU/e$^-$]		SER [10^{-3} e$^-$/pix]		Ruido [e$^-$]	
	Cuadrante 1	Cuadrante 2	Cuadrante 1	Cuadrante 2	Cuadrante 1	Cuadrante 2
1	230.4 ± 1.5	223.1 ± 1.4	1.4 ± 0.2	1.2 ± 0.2	0.204 ± 0.001	0.212 ± 0.001
2	230.0 ± 1.9	222.9 ± 1.7	1.5 ± 0.3	1.2 ± 0.2	0.214 ± 0.006	0.227 ± 0.008
3	230.4 ± 1.5	222.7 ± 1.7	1.4 ± 0.2	1.2 ± 0.2	0.208 ± 0.001	0.229 ± 0.001
4	229.4 ± 0.8	221.9 ± 0.9	2.0 ± 0.3	1.7 ± 0.2	0.166 ± 0.001	0.183 ± 0.003
5	229.6 ± 0.8	222.0 ± 1.0	2.1 ± 0.3	1.8 ± 0.2	0.169 ± 0.001	0.200 ± 0.006
6	230.2 ± 0.8	222.4 ± 0.8	2.4 ± 0.3	2.1 ± 0.3	0.167 ± 0.001	0.190 ± 0.001

Tabla 3.2: Resumen de la ganancia, tasa de eventos de un solo electrón (SER) y ruido para cada run y cada cuadrante [5].

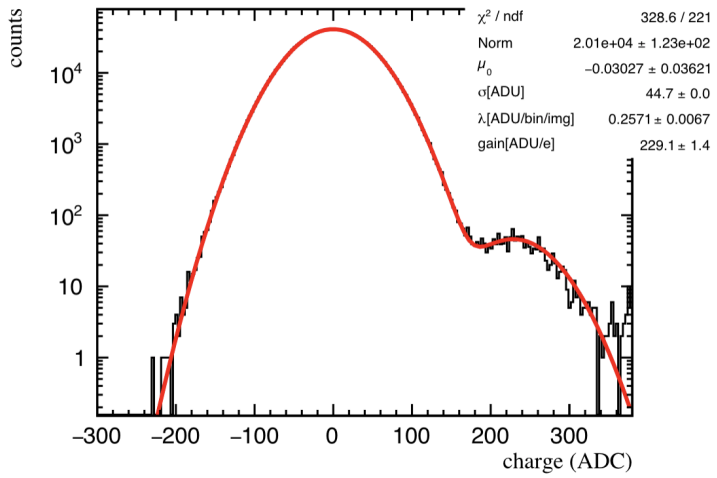

Figura 3.5: Distribución de la carga medida en los píxeles, ajustada a una suma de dos funciones gaussianas mas poissonianas (rojo).

corresponden con $1e^-$). El SER se define como $SER = \frac{a_1}{a_0}$. Es el cociente entre el número de pixeles con un electrón de carga entre los que tienen 0 electrones. Indica la tasa de aparición de eventos de un electrón sin señal externa (fondo electrónico del sensor). Finalmente, el ruido es igual a σ_0, es decir, la desviación del pico de 0 electrones. Representa la variación debida al ruido electrónico, que se observa en un pixel cuando no hay carga[5]. En la figura 3.5 se muestra el ajuste descrito a la distribución de carga encontrada para una imagen a modo de ejemple. Para esta imagen se encuentra $\sigma_0 = \tilde{4}4$ ADUs, una ganancia de $\tilde{2}29$ ADU/e^- y una $\lambda = 0.257$ ADU/img/bin.

3.6. Función para detectar hot columns

Aunque el programa WADERs ya implementa un método automático para detectar hot columns — pero que esta basado en imágenes con baja contaminación por rayos cósmicos, típicas de entornos de laboratorio subterráneo— en este trabajo se ha desarrollado un enfoque alternativo adaptado a imágenes con mayor nivel de ruido, como las utilizadas en el experimento Moskita. Esta nueva función permite identificar de forma eficaz las columnas defectuosas persistentes, incluso en presencia de eventos físicos, mejorando así la limpieza previa al análisis del halo. La detección se basa en analizar únicamente los píxeles que no pertenecen a clústeres con al menos una semilla, tal como se

definió anteriormente.
Para estos píxeles, se calcula la energía total registrada en cada columna, sumando la energía de cada uno de ellos. Esta distribución muestra picos claramente visibles sobre una base media. Para localizar estos picos de forma automática, se aplica un algoritmo de búsqueda de máximos de ROOT. La función creada se denomina 'get hot column pixels' que se detalla en el apéndice A.3.1. Esto permite identificar con precisión las columnas problemáticas y excluirlas del análisis posterior, evitando así distorsiones en los resultados.

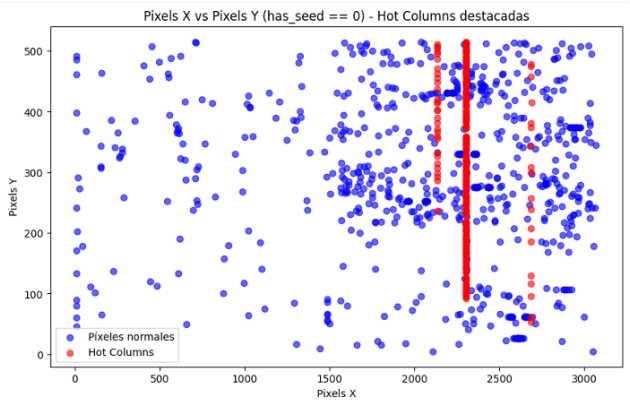

Figura 3.6: Misma imagen que la figura 4.3 pero solo aparecen pixeles sin seed. En rojo se destacan las hot columns.

En la figura 3.6 se puede observar una imagen en las que solo se han dibujado los pixeles sin seed. En rojo se han marcado las hot columns detectadas.

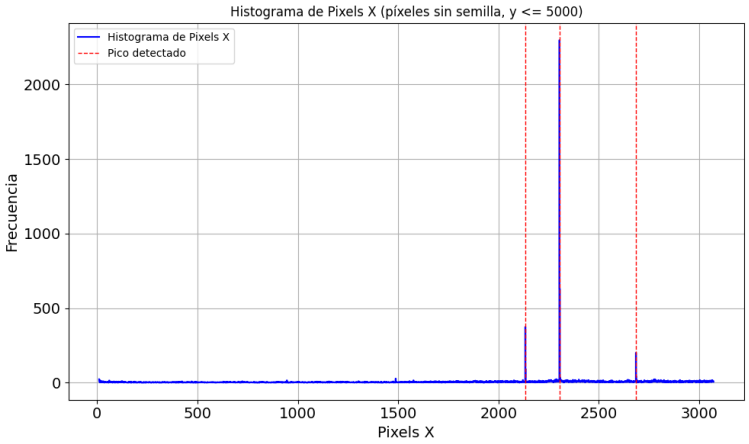

Figura 3.7: Distribución de la variable pixelsX con las hot columns detectadas por la función en rojo para el run3-ext2.

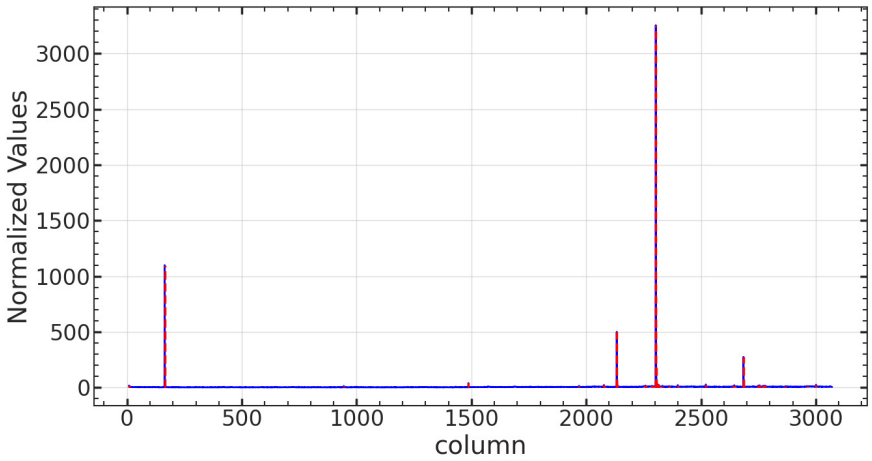

Figura 3.8: Hot columns detectadas en rojo por el otro método para el run3-ext2.

En las figuras 3.7 y 3.8 se puede observar las hot columns detectadas por dos métodos distintos. La figura 3.7 corresponde con las hot columns detectadas por la función que he creado. La figura 3.8 se ha obtenido utilizando un método diferente para detectar las hot columns. Primero, se cuentan los píxeles con cargas bajas (entre 1 y 4 electrones) en cada columna, después de aplicar una máscara para eliminar los clústeres. Luego, se ordenan estas columnas de mayor a menor según el número de píxeles y se busca el salto más grande entre columnas consecutivas. Las columnas por encima de ese salto se marcan como hot y se repite el proceso hasta que ya no se observan cambios importantes, solo variaciones debidas al ruido. Por último, se comprueba si las columnas vecinas también tienen más píxeles de lo normal, lo que ayuda a confirmar si realmente pertenecen a una hot column. Como se observa en las gráficas, el método que he implementado es capaz de detectar las principales hot columns, coincidiendo con las identificadas por el método oficial. No obstante, no logra identificar columnas muy pequeñas, es decir, aquellas formadas por pocos píxeles con una energía ligeramente por encima de la media. Esta limitación debe tenerse en cuenta a la hora de interpretar los resultados. Hay que matizar que el número de hot columns detectadas por la función creada puede aumentar si se disminuye el umbral de que una columna sea considerada pico en un histograma. El objetivo de esta sección era comprobar si existía una forma automática de detectar las hot columns sin depender directamente de la base media ni aplicar cortes explícitos. La función desarrollada demuestra ser útil, ya que permite identificar las columnas principales, lo que valida su enfoque. Además, existe margen de mejora ajustando los parámetros, aunque no se ha abordado en profundidad dentro del alcance de este trabajo. Esta línea podría explorarse en desarrollos futuros. Otro fenómeno a destacar es que las principales hot columns detectadas son prácticamente las mismas en los distintos runs. Esto indica que se trata de defectos propios de la CCD presentes desde el inicio de la toma de datos. Cabe señalar que los defectos en las CCDs pueden deberse a múltiples causas y, aunque algunos se mantienen estables, otros pueden desaparecer tras la aplicación de ciclos térmicos. Esto es importante ya que nos permite eliminar dichas columnas para nuestro análisis en todos los archivos. Las gráficas para el resto de runs se muestran en el apéndice.

4 Resultados y análisis

En este capítulo se presentan los distintos resultados obtenidos del análisis de las imágenes de los Skipper CCDs. En este estudio se han caracterizado los clusters de baja energía que podrían ser producidos por clusters más energéticos (como se ha explicado anteriormente y que vamos a denominar halo [21]), principalmente en base a su energía y la distancia entre los clusteres. A partir de estos estudios se puede determinar un enmascaramiento del halo para llevar a cabo la limpieza de las imágenes. En la figura 4.1 se puede ver un ejemplo de una imagen a la que le hemos aplicado el procesamiento explicado en el capítulo anterior para extraer los clusters. En la imagen se observan los clusters encontrados en cada fila y columna. Se pueden distinguir clusters energéticos (suelen ser varios píxeles juntos con varios electrones cada uno) y clusters de baja energía (suelen ser pixeles aislados con 1-2 electrones cada uno). También se pueden observar hot columns, hot pixels y serial registers. Se aprecia un mayor número de eventos en la segunda mitad de la imagen (mirando por columnas a partir de la columna 1500) que en la primera. Se esta investigando a que es debido este fenomeno pero no se ha encontrado justificación de momento.

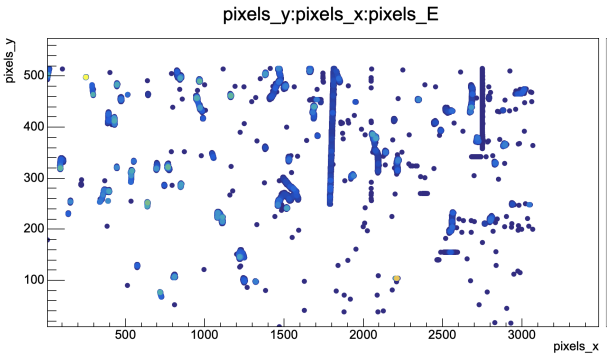

Figura 4.1: Ejemplo de imagen procesada que se va a analizar. En esta imagen aparecen los distintos clusters detectados dibujados

Com dijimos, a partir de todas las imágenes, se seleccionan todos los clústeres cuya energía total sea superior a 50 keV. Para ello, se calcula la energía sumando la carga de los píxeles y convirtiéndola a energía mediante la relación $1\,e^- \approx 3.7\,eV$. También se requiere que el clúster contenga al menos un píxel con una energía superior a 13 eV (es decir, con más de 4 electrones), lo que se considera como una *semilla* o *seed pixel*. Se excluyen los clústeres que estén en contacto con *hot columns* o que contengan *hot pixels*. Para estos clusters selecionados se calcula su centro geométrico haciendo la media entre la posición máxima y mínima de los clusters en los ejes x e y. Una vez encontrados los clusters energéticos con sus centros, se construyen las cajas en base a estos criterios:

Límites de la Caja:

- Izquierda: $\text{minX} - Xmargin$
- Derecha: $\text{maxX} + Xmargin$
- Inferior: $\text{minY} - \text{Y}margin$
- Superior: $\text{maxY} + \text{Y}margin$

Condiciones de la Caja:

- **X-margin:** $\max(100, (\text{maxX} - \text{minX})) \times (1.2 \text{ si } N_{\text{pix}} > 100 \text{ else } 1.0)$
- **Y-margin:** $20 \times (1.1 \text{ si } N_{\text{pix}} > 100 \text{ else } 1.0)$

En la figura 4.2 se puede observar el tipo de cajas que se están generando alrededor de los clústeres energéticos. Estas cajas nos permitirán estudiar la densidad de energía, la forma y otros parámetros

en función de la distancia al centro del clúster. Este análisis será desarrollado en las distintas secciones posteriores.

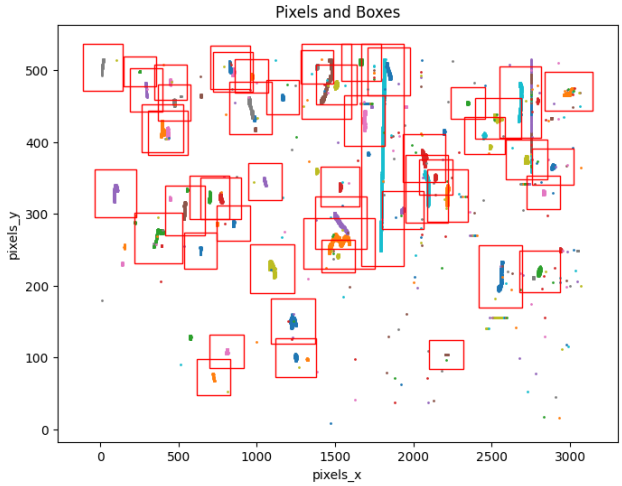

Figura 4.2: Imagen en la que aparecen las cajas dibujadas para los clusters energéticos de la figura 4.1

Para estas imágenes se han desarrollado diferentes códigos en Python para estudiar las distancias previamente mencionadas. Todos estos códigos se expondrán en el apéndice.

4.1. Estudio de la densidad de clusters de baja energía dentro de las cajas

En esta sección hemos estudiado cómo se comporta la densidad de píxeles (especialmente aquellos sin seed) en las regiones cuadradas definidas anteriormente alrededor del clúster. Las cajas permiten una caracterización más restrictiva del entorno, ya que abarcan un área mayor y ofrecen una segmentación más clara en filas y columnas. Se calculan las distancias desde los píxeles sin seed en el cuadrante abajo-izquierda respecto al centro del cluster. Además de las distancias, se calculan las densidades de los píxeles dividiendo el numero de ellos en el cuadrante inferior-izquierda con respecto al centro del cluster entre el área de la caja que estamos analizando (aproximadamente un cuarto del área total). Para cada pixel, se representa su distancia al cluster y la densidad encontrada en la caja, ver figura 4.3.

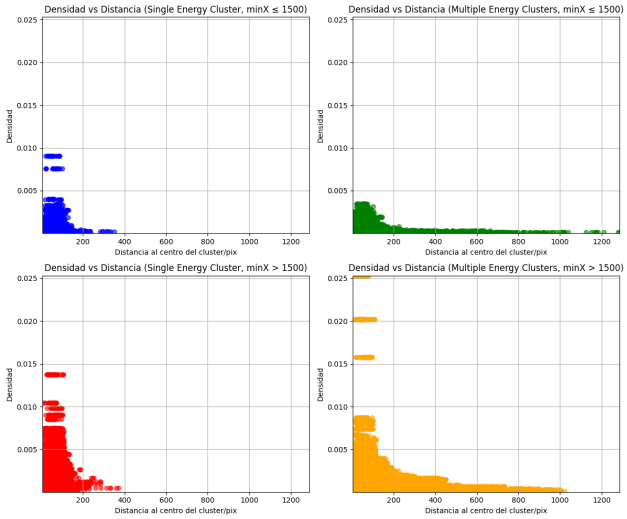

Figura 4.3: Representación de las densidades encontradas en las cajas frente a la distancia al centro del cluster correspondiente para el archivo hadd-ext2. A la izquierda se muestran para las cajas con un solo cluster y a la izquierda para las que tienen múltiples clusters. Las gráficas de arriba son para clusters en columnas inferiores a 1500 y abajo para clusters que se encuentran en columnas mayores.

En esta figura se muestra la densidad para varios casos que reflejan la particularidad de estas imagenes que tienen muchos clusters. En primer lugar, se ha separado el estudio entre la primera mitad de la imagen ($minX \leq 1500$) y la segunda mitad de la imagen ($minX > 1500$). Además, se ha distinguido entre cajas que contienen un único cluster energético (Single Energy Cluster) y aquellas que contienen varios clusters energéticos dentro de la misma caja (Multiple Energy Clusters).
Al observar las gráficas, se aprecia que en las cajas con un solo cluster energético, la gran mayoría de los píxeles sin seed se encuentran a distancias inferiores a 100 píxeles. La densidad de píxeles es más alta en la segunda mitad de la imagen, algo esperado por el aumento de sucesos en la segunda mitad de las imagenes, efecto todavia sin entender.
Por otro lado, en las cajas que incluyen varios clusters energéticos, aunque la mayoría de los píxeles también se concentran a distancias cortas (menores a 100 píxeles), en este caso las distancias tienden a ser mayores. De hecho, la distribución se extiende hasta casi 200 píxeles con densidades cercanas a 0,005, lo que no ocurre en los casos con un único cluster. También se observa la presencia de píxeles que llegan a distancias de hasta 800–1200 píxeles. Este efecto puede explicarse por dos razones principales: en primer lugar, algunos clusters tienen trazas alargadas que extienden su influencia espacial; en segundo lugar, al calcular las distancias desde el centro de un cluster, pueden incluirse píxeles que realmente están más próximos a otro cluster cercano dentro de la misma caja, generando distancias más grandes de las debidas. En otras palabras, esos píxeles pueden provenir del halo de otro cluster energético presente en la caja.
Finalmente, es importante destacar que en todas las gráficas el decrecimiento inicial de la densidad es muy marcado, pero comienza a estabilizarse alrededor de los 100 píxeles.

4.2. Cálculo de densidades de píxeles para anillos concéntricos

En esta sección estudiaremos la densidad de clusters sin semilla que se encuentran en las regiones definidas alrededor de los clústeres energéticos con semilla. Para ello, se definen anillos concéntricos

centrados en la posición del cluster energético, tal como se describió al comienzo de este capítulo. En estos anillos, solo se analiza la densidad para píxeles que se encuentren en el sector abajo-izquierda respecto al centro del cluster energético. Restringimos el análisis a este cuadrante ya que es la zona más limpia para analizar el halo. Debería ser muy similar en la parte superior-izquierda y en la parte inferior-derecha. Cuando hablo de abajo izquierda, me refiero a píxeles cuya coordenada en el eje x y en el eje y son menores que las del centro del cluster energético (cuadrante inferior-izquierda en la figura 1.4).

La forma de construir los anillos concéntricos es la siguiente. Se crea un primer círculo cuyo radio se corresponde con la distancia entre las posiciones del centro del cluster energético y la posición mínima en el eje X de la traza del cluster. A continuación, se van construyendo anillos concéntricos aumentando el radio en 10 píxeles hasta construir 15 anillos concéntricos. Para calcular la densidad, se hace el cociente del número de píxeles sin seed que hay en el área comprendido entre los anillos concéntricos dividido entre todo el área de esa región. En el caso del primer anillo, el área total será el area del circulo dividido entre 4: densidad $= \frac{N_{\text{pix}}}{0{,}25 \cdot \pi \cdot \left(r_{\text{out}}^2 - r_{\text{inner}}^2\right)}$. Para este primer anillo no se tienen en cuenta los pixeles del propio cluster energético en lo que sería el numerador. En la figura 4.4 se puede observar un ejemplo de un cluster en rojo, los anillos dibujados alrededor en verde (solo he dibujado los 3 primeros) y en morado los pixeles sin seed que encontramos en el segundo anillo. Tambien se observa como el primer anillo tiene como radio la distancia entre el centro y las coordenadas de minX.

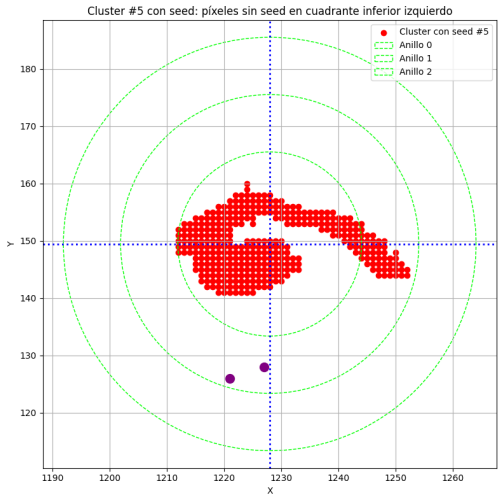

Figura 4.4: Imagen de un cluster, con los primeros anillos dibujados y los pixeles sin seed analizados en la región de interés.

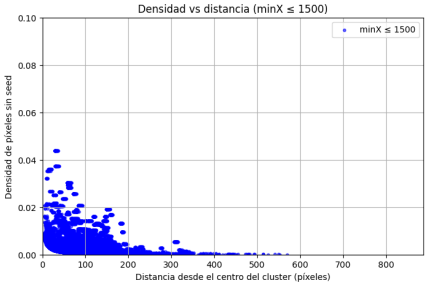

Figura 4.5: Representación de las densidades frente a la distancia al centro del cluster correspondiente para el run1 (primera mitad de la imagen).

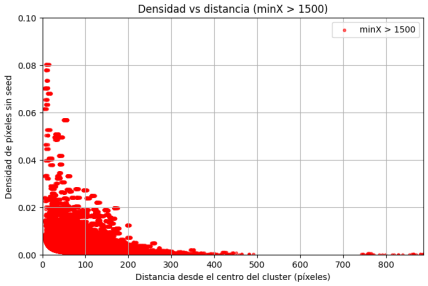

Figura 4.6: Representación de las densidades frente a la distancia al centro del cluster correspondiente para el run1 (segunda mitad de la imagen).

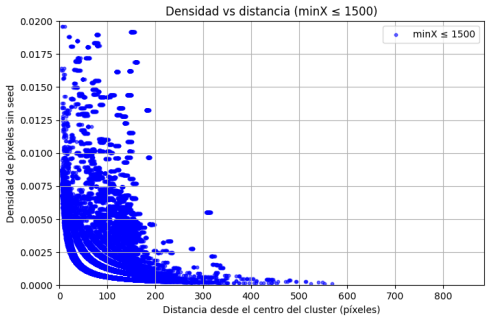

Figura 4.7: Representación de las densidades frente a la distancia al centro del cluster correspondiente para el run1 (primera mitad de la imagen) haciendo zoom en el eje hasta 0,02.

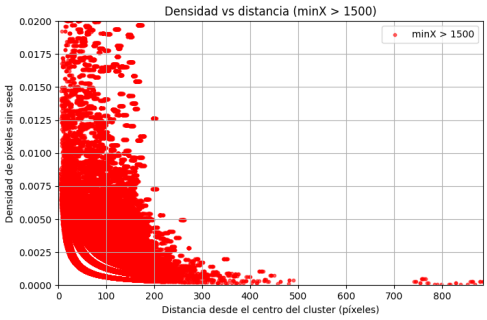

Figura 4.8: Representación de las densidades frente al orden de anillo correspondiente para el run1 (segunda mitad de la imagen) haciendo zoom en el eje hasta 0,02.

En las figuras [4.5, 4.8] se ha representado el modulo de la distancia de los píxeles al centro del cluster energético frente a la densidad de pixeles en cada anillo. A los pixeles correspondientes al mismo anillo se les ha asignado la misma densidad para realizar la gráfica.

En estas gráficas se puede observar un comportamiento similar para todos los archivos analizados. En primer lugar, cabe destacar que la densidad de píxeles decrece exponencialmente y comienza a mantenerse constante a unos 50 píxeles de distancia del centro del cluster energético. Esto esta de acuerdo con lo encontrado en el estudio de las cajas. Donde mejor se observa esto es en la figuras 4.6, en la que hay un máximo claro a distancias pequeñas y el decrecimiento deja de ser tan pronunciado a partir de 50 píxeles aproximadamente. Esto concuerda con la hipótesis de que los clusters energéticos generan un halo de energía que tiene mayor concentración cerca del propio cluster, y que se atenúa con la distancia. También se observa que, en la segunda mitad de la CCD (como en la sección 4.1), los valores de las densidades de pixeles es mucho mayor que en la primera mitad ya que el número de eventos producidos es más alto.
En las figuras 4.7 y 4.8 se ha hecho un zoom en la densidad a valores menores que 0,02 donde se puede observar la aparición de diversas curvas. Esto se puede deber a la forma de los clusters. Los más compactos parece que concentran el halo más cercano a los píxeles y por tanto sus valores correspondientes a la densidad en distancias más cortas, mientras que los clusters más alargados concentran sus valores de la densidad a mayores distancias. Otra de las cosas que puede pasar es que, en este caso, no hemos distinguido entre anillos con un solo cluster y los que tienen varios. De ahí

que el halo detectado para un cluster pueda provenir de otro. Esto puede causar que encontremos distancias más altas de las debidas.

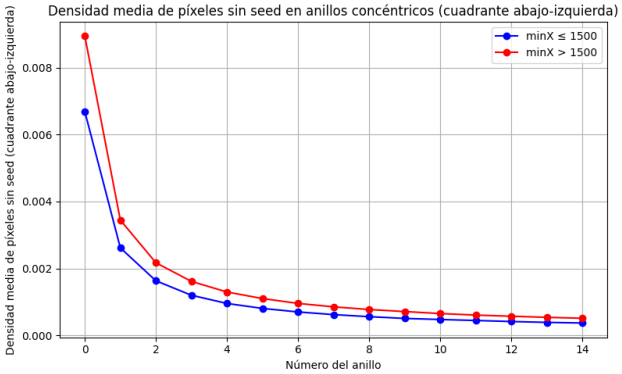

Figura 4.9: Representación de las densidades promedio frente al orden de anillo correspondiente para el run1.

Para determinar mejor la distancia en la que halo empieza a decrecer, se ha representado el orden del anillo (primero, segundo, tercero, etc.) frente al valor medio de la densidad calculado para todos los anillos del mismo orden pertenecientes a diferentes clusters energéticos. Es decir, para cada cluster energético, se calcula la densidad de píxeles sin seed dentro de cada uno de sus anillos concéntricos. A continuación, se utilizan los datos de todos los primeros anillos de todos los clusters y se calcula su densidad media; luego se hace lo mismo con los segundos anillos, los terceros, y así sucesivamente. El resultado es una curva que muestra cómo evoluciona la densidad promedio en función de la distancia (representada aquí por el número de anillo) al centro de los clusters energéticos. Se observa que la densidad disminuye exponencialmente de forma pronunciada para los primeros anillos, lo que refleja la influencia del halo creado por los clusters energéticos. El decrecimiento se estabiliza a partir del cuarto anillo, correspondiente a unos 46 pixeles más o menos.
En la tabla 4.1 se detalla cuál es primer radio que se dibuja alrededor del cluster energético. En otras palabras, la distancia entre el centro del cluster a las coordenadas de la posición minX. Esto quiere decir que el segundo círculo tendrá un radio de 22-23 píxeles, y el tercero de 32-33, y así sucesivamente. La gráfica se estabiliza a partir del cuarto anillo, este anillo se encuentra comprendido entre los radios de 42 y 52 píxeles aproximadamente. Esta distancia está de acuerdo con lo estudiado hasta ahora. Pese a que el valor de la densidad es más alto en la segunda mitad de la CCDs, el comportamiento de la curva es el mismo en ambas mitades.

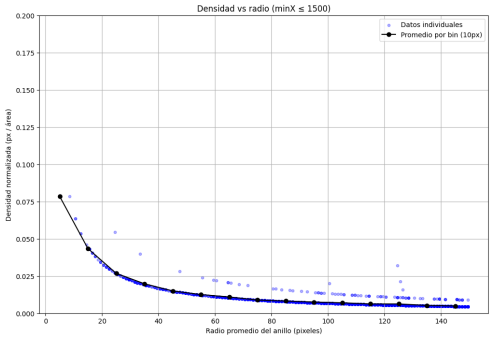

Figura 4.10: Densidades frente a las distancias para todas las regiones con halo y sin CTI dibujando 150 anillos para la primera mitad de la imagen.

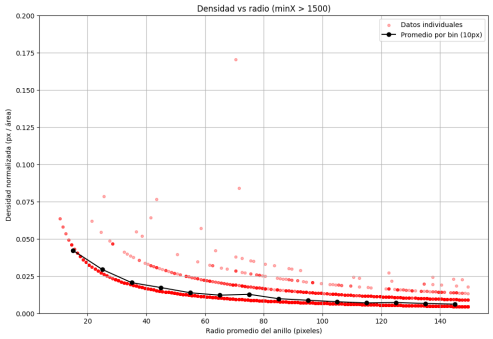

Figura 4.11: Densidades frente a las distancias para todas las regiones con halo y sin CTI dibujando 150 anillos para la segunda mitad de la imagen.

Para asegurarnos que esta distancia es consistente y robusta a variaciones, analizamos la región entera del halo sin CTI (ahora es la región izquierda entera y la parte derecha por debajo del cluster), como se indicaba en la figura 1.4. Las figuras 4.10 y 4.11 muestran este estudio donde para estas gráficas se han modificado 2 cosas: la zona de análisis y la forma de construir los anillos. Para este caso se han construido anillos con radios de tamaño 1 píxel hasta 150. Es decir, primero se ha dibujado un anillo cuyo radio interno es de 1 píxel y el externo de 2, y así sucesivamente hasta llegar a 150 píxeles. En las gráficas se observa cómo la densidad decrece exponencialmente hasta que se estabiliza a unos 40-50 píxeles. Se vuelven a observar distintas curvas, sobre todo en la segunda mitad de la imagen donde hay más clusters. Cabe destacar que estas figuras son el análisis de 10 imágenes únicamente ya que era imposible analizar muchas más imagenes (Python no era capaz de procesar ese volumen de datos).

4.3. Método alternativo para el cálculo de la densidad

Hemos estudiado un método alternativo al ya presentado para calcular la densidad de los píxeles alrededor del cluster energético. Hay dos diferencias principales: la construcción de los anillos y la forma de calcular la densidad de píxeles por anillo.

En la sección 4.2, lo que se hacía era que el primer radio tenía como valor la distancia entre el centro del cluster y las coordenadas de la posición minX. En este caso, se ha optado por dibujar anillos idénticos para todos los clusters, cuyos radios son de 10, 20, 30 hasta 150 píxeles. Por otro lado, la forma de calcular la densidad de píxeles no tiene en cuenta el área del anillo. Cabe destacar que, como en la sección 4.2, el análisis se ha centrado en el cuadrante inferior izquierdo respecto al centro del cluster, donde se espera que aparezca el halo.

En este método el centro de los anillos se corresponde con las coordenadas del valor Qmax (el píxel con mayor carga del cluster). La razón es que hay algunos clusters cuyo centro no se correspondía con ningún píxel del cluster. Esto ocurre en aquellos casos en los que la traza del cluster energético es muy pequeña o bien la forma es muy extraña. En la figura 4.12 se muestra la distribución de distancias entre el centro geométrico del cluster y su posición de máxima carga. Se puede observar que, en general, la distancia entre ambos puntos es muy pequeña (1-5 pixeles), sin embargo, hay algunos casos en los que esa distancia puede llegar incluso a los 50 pixeles. Esto puede ocurrir

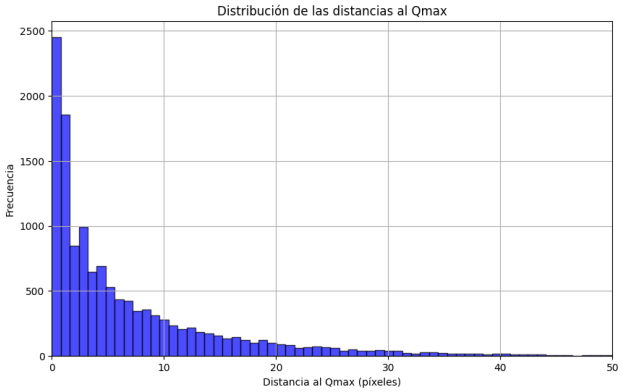

Figura 4.12: Distribución de las distancias entre el centro geométrico del cluster y la posición de Qmax de los clusters.

para muones, cuya traza es muy alargada y abarca bastantes píxeles. Tambien puede ocurrir que determinados clusters depositan su carga al final de la traza (caso electrones energéticos).
Para calcular la densidad, el numerador sigue siendo el número de píxeles que encontramos en el anillo como en el método anterior. En cambio, el denominador es la suma de los píxeles encontrados en el anillo y los píxeles del cluster que se encuentran en el cuadrante inferior-izquierda respecto a la posición Qmax. En la figura 4.13 se expone lo que se quiere calcular en esta sección. Se puede observar los anillos concéntricos cuyo centro se corresponde con el centro del cluster. Por ejemplo, abajo a la izquierda de la imagen se pueden observar 2 píxeles en el mismo anillo (anillo 13). Suponiendo que el número de píxeles en verde por ejemplo es 18, la densidad calculada sería $2/(2+18) = 0.1$.

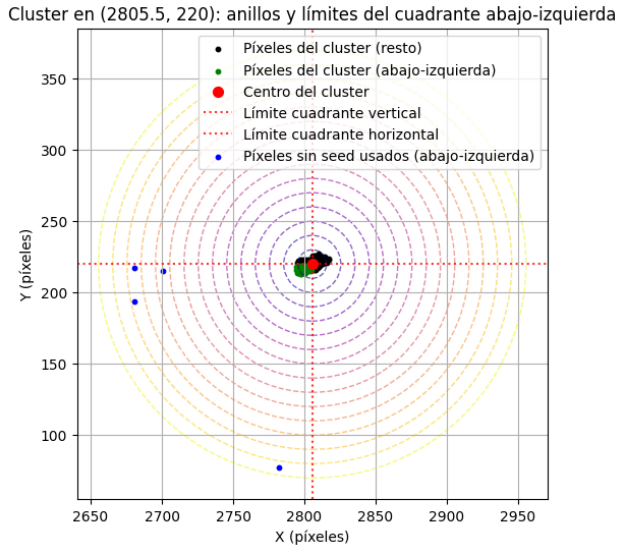

Figura 4.13: Imagen explicativa de lo que se quiere hacer en esta sección, donde se muesran los anillos dibujados, los píxeles que se encuentran dentro de los anillos (azul) y los píxeles del cluster energético que se usan para calcular la densidad (verde).

Esta densidad por anillo se ha representado gráficamente frente a la distancia promedio de pixeles del anillo a la posición Qmax. Siguiendo con el ejemplo anterior, si los dos píxeles de antes se

encuentran a 50 y 54 píxeles de distancia, se representraria el 0,1 anterior frente a 52.

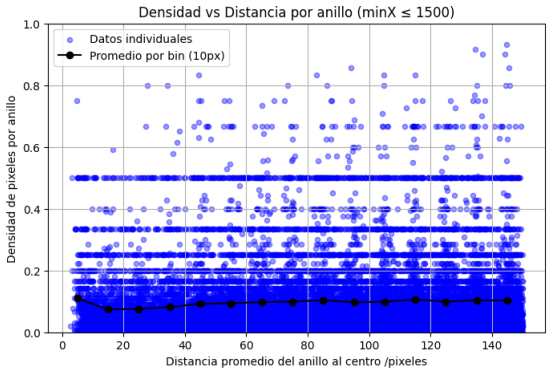

Figura 4.14: Representación de las densidades frente a las distancias promedio para la primera mitad de la imagen para el run1.

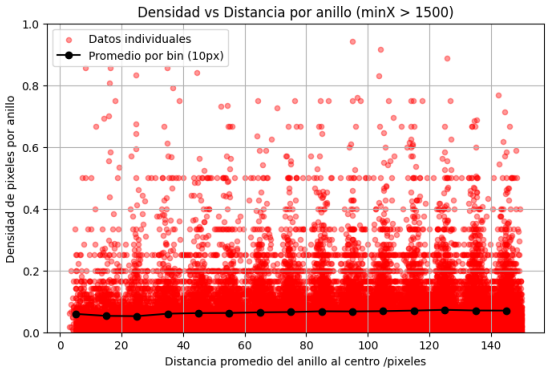

Figura 4.15: Representación de las densidades frente a las distancias promedio para la segunda mitad de la imagen en el run1.

En las figuras 4.14 y 4.15, se muestran las densidades representadas frente a la distancia promedio de los píxeles al centro de los anillos para el run1 separando los datos por cada mitad de las imágenes. La curva negra representa el promedio de los valores con calculado en intervalos de 10 píxeles. En estas gráficas la mayoría de las densidades tienen valores bajos (por debajo de 0,2). Sin embargo, también alcanza valores altos alrededor de 0.8 y 0.9. Esto se debe a cómo se ha definido la densidad de píxeles. En el denominador, al incluir los píxeles en el cuadrante inferior-izquierda respecto del centro de los anillos, se puede dar el caso en el que ese número sea muy bajo (por ejemplo, al estar el pixel con mas carga cerca del final del cluster). Algo a destacar es que la curva promedio se mantiene bastante constante a lo largo de todas las distancias. Esto lo que da a entender es que no hay mucha variación de la distribución de píxeles en el cuadrante que estamos analizando. Uno de los fenómenos que se sigue apreciando es que en la segunda mitad de la imagen (figuras en rojo), la dispersión de puntos es mayor para valores altos de la densidad.

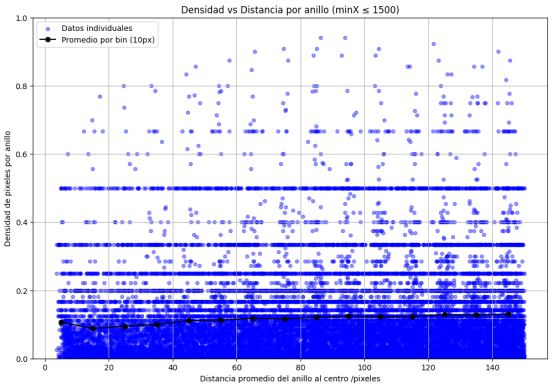

Figura 4.16: Representación de las densidades frente a las distancias promedio para la primera mitad de la imagen en el run1 en el cuadrante inferior-derecha.

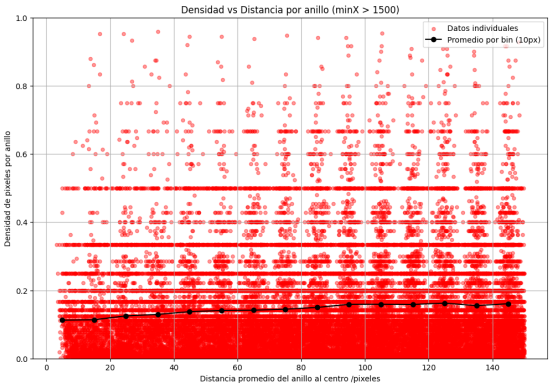

Figura 4.17: Representación de las densidades frente a las distancias promedio para la segunda mitad de la imagen en el run1 en el cuadrante inferior-derecha.

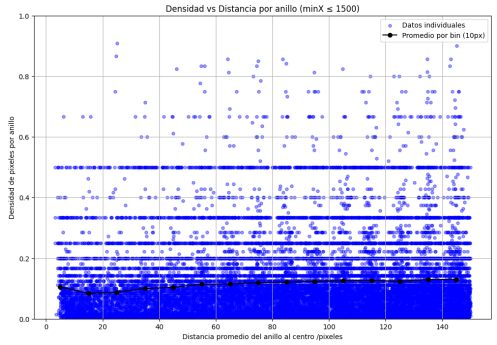

Figura 4.18: Representación de las densidades frente a las distancias promedio para la primera mitad de la imagen en el run1 en el cuadrante superior-derecha.

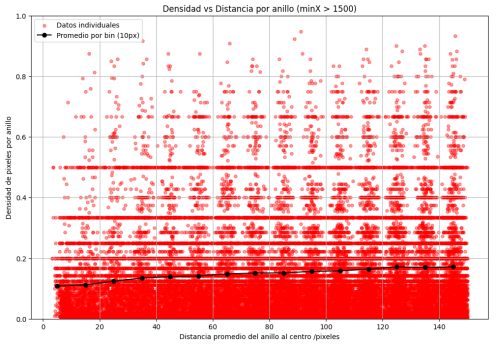

Figura 4.19: Representación de las densidades frente a las distancias promedio para la segunda mitad de la imagen en el run1 en el cuadrante superior-derecha.

En las figuras [4.16-4.19], se han realizado las mismas figuras que las anteriores pero para los cuadrantes de la derecha: el superior y el inferior. Esto se ha hecho a modo de comparación con el cuadrante analizado. En estas regiones esperamos encontrar el CTI (sobre todo en el cuadrante superior-derecha), aunque se incluyen regiones que no tienen CTI. Las gráficas no muestran nada distinto a lo observado en las gráficas anteriores. La tendencia de los puntos en negro es muy similar.

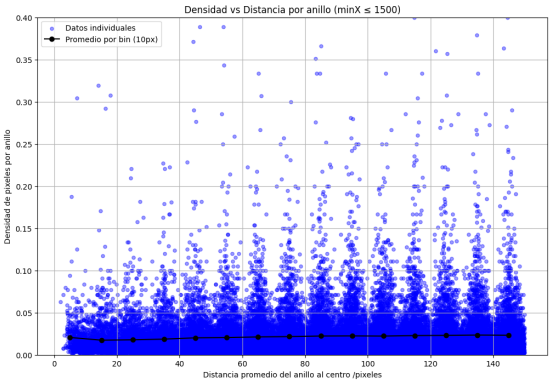

Figura 4.20: Representación de las densidades frente a las distancias en las zonas de halo sin CTI para la primera mitad.

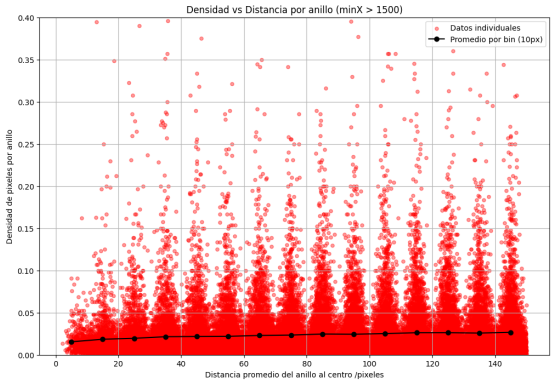

Figura 4.21: Representación de las densidades frente a las distancias en las zonas de halo sin CTI para la primera mitad.

En las figuras 4.20 y 4.21 se ha hecho el mismo análisis pero en las regiones en las que hay halo pero sin CTI de acuerdo a la figura 1.4. No se encuentra nada que no se haya discutido previamente.

El objetivo principal de este método era evaluar si la densidad de píxeles disminuye progresivamente a medida que aumenta la distancia al cluster, lo que permitiría caracterizar mejor el alcance espacial del halo. Sin embargo, los resultados obtenidos no han mostrado el comportamiento esperado: las densidades calculadas tienden a mantenerse bajas y relativamente constantes en todas las distancias, sin evidenciar un descenso sistemático en los anillos exteriores.
Además, el análisis por cuadrantes no revela diferencias destacables entre las zonas donde se esperaba observar efectos de halo (como el cuadrante inferior izquierdo) y las regiones dominadas por efectos de CTI (en los cuadrantes derechos).

Estos resultados podrían deberse a dos causas principales: por un lado, es posible que el método aplicado no sea el más adecuado para detectar el efecto buscado puesto que es muy dependiente de la forma del cluster; por otro lado, los archivos analizados podrían no ser los mejores candidatos, ya que podrían corresponder a imágenes relativamente limpias, sin suficientes señales contaminadas por halos. Dentro de los problemas del método se puede destacar lo que se observa en la figura 4.22. En esta imagen se observa que, para el cluster señalado con la flecha, al dibujar los anillos, cuente como halo el CTI de la traza alargada (circulo). Este problema tambien puede manifestarse en la sección 4.2. Como se ha explicado en la secciñon anterior, la densidad no decrece a mayores distancias ya que podemos estar asignando píxeles de halo cercanos a un cluster a otro cluster, alterando los datos.

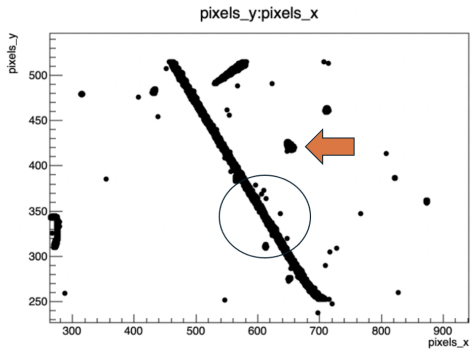

Figura 4.22: Imagen que representa uno de los problemas del método aplicado en la sección 4.3.

4.4. Caracterización de la forma de clusters energéticos

Se ha llevado a cabo un estudio para caracterizar la forma de los clusters más energéticos, con el objetivo de determinar la posible partícula que los origina y analizar si existe una correlación con el halo. Los clusters generados en sensores CCD pueden mostrar morfologías muy distintas en función del tipo de partícula incidente. Por ejemplo, los muones tienden a dejar trazas largas y lineales; los electrones, partículas beta generan trazas más cortas y curvas, y las partículas alfa o los depósitos de baja energía suelen formar estructuras más compactas o redondeadas. Para ello, se introduce una variable que determina la ocupación de los píxeles del cluster dentro del área mínima que podría ocupar si tuviera forma rectangular. Esta variable, que denominaremos fracción de píxeles, por simplicidad $fpix$, se define como el cociente entre el número de píxeles cargados que conforman el cluster y el área del rectángulo mínimo que lo contiene [1]. Esta área se calcula como:

$$\text{Área} = (\text{maxX} - \text{minX} + 1) \times (\text{maxY} - \text{minY} + 1)$$

Se suma una unidad en cada dimensión para contar correctamente todos los píxeles incluidos en los extremos del rectángulo. Sin esta corrección, la fracción podría superar la unidad en ciertos casos, lo cual carece de sentido físico. Este análisis se ha realizado exclusivamente sobre los clusters que presentan seed.

La variable $fpix$ resulta útil para caracterizar cuantitativamente estas diferencias morfológicas. Un cluster con forma alargada, como el dejado por un muón, ocupará una gran área en el plano del detector con relativamente pocos píxeles cargados, dando lugar a un valor bajo de $fpix$. En cambio, un cluster más compacto y denso, como el generado por una partícula alfa, tendrá una fracción de píxeles más elevada, al ocupar un área menor para un número similar de píxeles activos. Las partículas alfa suelen tener energías altas (por encima de 2 MeV). Los electrones suelen tener

energías bastante más bajas ya que son originados principalmente por actividad radiactiva. Estas diferencias se notan en la figura 4.23.
Los muones suelen llegar a la superficie terrestre con energias alrededor de $2-4\ GeV$. Sin embargo, los datos se han tomado con skipper ccds situados a 70 m bajo tierra. Teniendo en cuenta que la atenuación de los muones es de alrededor de $2-3\ MeV/cm$, se puede entender que en las figuras que se han realizado los muones lleguen al detector con energías de unos pocos MeV. Se concluye que los puntos correspondientes a altas energías y bajos valores de $fpix$ puedan corresponderse con trazas de muones. En la figura 3.4 se observa que los muones tienen una traza alargada en diagonal, por lo que sus valores de $fpix$ serán muy pequeños.
Por tanto, esta variable permite discriminar entre distintos tipos de eventos en función de su morfología, ofreciendo una herramienta sencilla pero eficaz para el análisis de imágenes CCD.

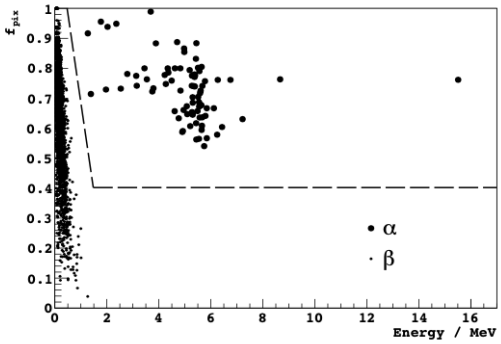

Figura 4.23: Grafica tomada de un artículo que se ha usado como referencia [1].

Además, este estudio se complementa con el análisis de la distribución espacial del halo de carga que rodea al cluster principal. Esta componente difusa también puede aportar información valiosa sobre la naturaleza de la partícula incidente, así como sobre los mecanismos de difusión y compartición de carga en el dispositivo.

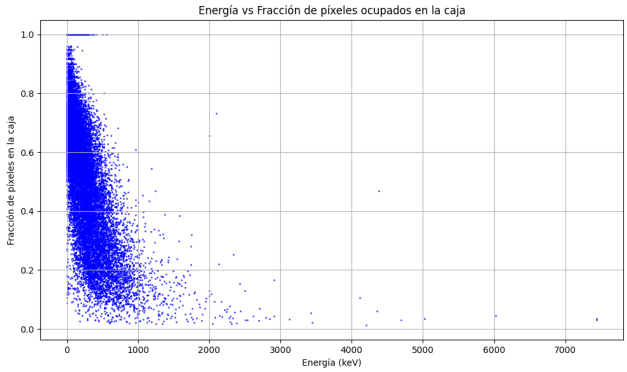

Figura 4.24: Representacion de $fpix$ en funcion de la energía del cluster para el run1.

En la figura 4.24 se puede observar que la mayoría de clusters con seed tienen energías que no superan los 1000 keV. Por otro lado, la mayoria de ellos tienen valores de $fpix$ por encima de 0,2. Los valores de $fpix$ más bajos se corresponden con trazas de clusters alargadas, mientras que los valores más altos se corresponden con trazas que se asemejan a un círculo/cuadrado o que son alargadas bien en el eje x o en el eje y.

Comparando la figura 4.23 con la 4.24, en las imagenes analizadas no se observan partículas alfa salvo algún caso en las que aparecen clusters con energías cercanas a 4 MeV y cuyo valor de $fpix$ es cercano a $0,6$. En la figura 4.24, observamos clusters con valores de fracción de píxeles cercanos a 1, pero no son muy energéticos. Por lo tanto, se espera que todos esos clusters se correspondan con electrones/partículas beta de baja energía. En la figura 3.4 se observa que la traza de estas partículas es curva y llega un momento en el que depositan toda su energía. Sin embargo, hay casos en que la traza pueda ser similar a la de las partículas alfa debido a que depositan toda su energía rápidamente.

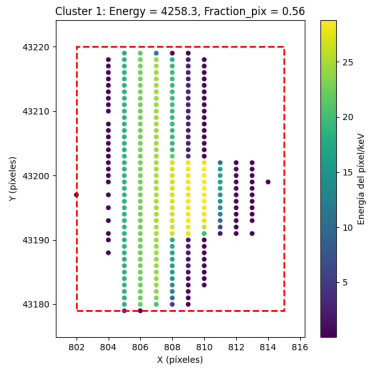

Figura 4.25: Traza de uno de los clusters con $fpix \approx 0,6$ y energía > 4000 keV . Se puede identificar como una partícula alfa.

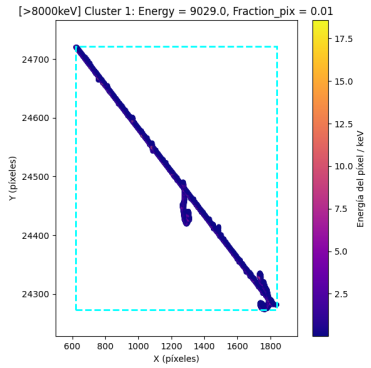

Figura 4.26: Traza de uno de los clusters con $fpix \approx 0$ y energía > 8000 keV. Se puede identificar como un muón.

En la figura 4.25 se puede observar la traza de un cluster que se podria identificar como una partícula alfa. Su traza es bastante compacta y ocupa bastánte hueco del rectángulo del borde rojo. Este rectángulo delimita el área que se ultiliza para calcular el valor de $fpix$. Por otro lado, en la figura 4.26 se puede observar una traza más alargada y en diagonal, lo que se puede identificar como un muón.

4.5. Ajuste de clusters a una recta

En esta sección se presentan los resultados obtenidos a partir del ajuste de los clusters energéticos ($E > 50$ keV) a una recta. Este procedimiento se ha utilizado para facilitar la identificación del inicio y el final de cada cluster, ya que su forma y tamaño pueden variar. Además, el ajuste lineal permite definir con mayor precisión las regiones del cluster que se desean estudiar. En esta sección solo se ha analizado el archivo correspondiente al run3 (únicamente 10 imágenes).

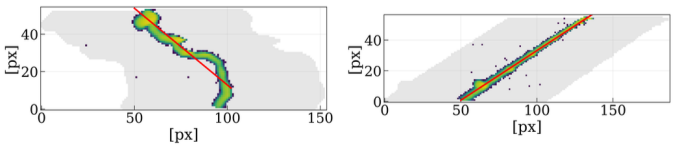

Figura 4.27: Ejemplo de ajuste de clusters a una recta tomado como referencia .

En la figura 4.27 se puede observar el ajuste de dos clusters individuales a una recta [15] como ejemplo de lo que se ha hecho en este estudio. En nuestro caso, a partir del ajuste de los clusters

energéticos a una recta se ha intentado obtener la distribución de distancias de píxeles sin seed cercanos al cluster energético. Para ello, se crean unas cajas más reducidas que las de la figura 4.2 de la siguiente forma:

- Borde izquierdo = minX - 50 // Borde derecho = minX + 50
- Borde superior = maxY + 20 // Borde inferior = minY - 20

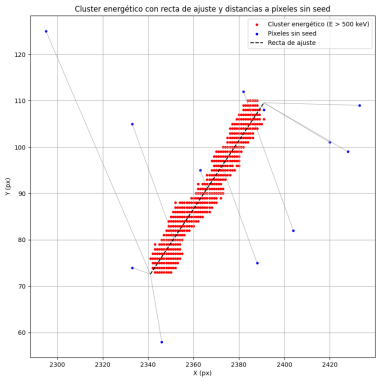

Figura 4.28: Ejemplo de ajuste de un cluster a una recta y destacados también los pixeles para los que se calcula las distancias

Se ha creado una función que calcula las distancias desde los píxeles sin seed al punto más cercano de la recta de ajuste del cluster correspondiente. Es importante matizar que dicha recta está delimitada por el tamaño del cluster (es decir, sus posiciones mínima y máxima). Para calcular la distancia, se obtiene la proyección perpendicular del píxel sobre la recta, y se mide la distancia correspondiente. Si no existe tal proyección (por estar fuera de los límites del cluster), se toma la distancia al extremo más cercano de la recta. Cabe destacar que un mismo píxel puede encontrarse dentro de varias cajas y, por tanto, se calcularían varias distancias a distintas rectas. En ese caso, se conserva únicamente el valor de distancia más pequeño en módulo. Este método permite eliminar el problema observado en secciones anteriores, donde los halos o el efecto del CTI de otros clusters podían interferir en el análisis del cluster principal, al asociar a cada píxel únicamente la distancia más representativa.
En la figura 4.28 se puede observar un ejemplo de un cluster que ha sido ajustado a una recta. Además, se han dibujado los píxeles que se encuentran dentro de la caja y las diferentes rectas representan la distancia que se está calculando, uniendo el pixel sin seed al punto más cercano de la recta de ajuste. Las rectas no son perpendiculares en la figura debido a que las cuadrículas no tienen el mismo rango en el eje x que en el y (es decir, es un problema de escala).

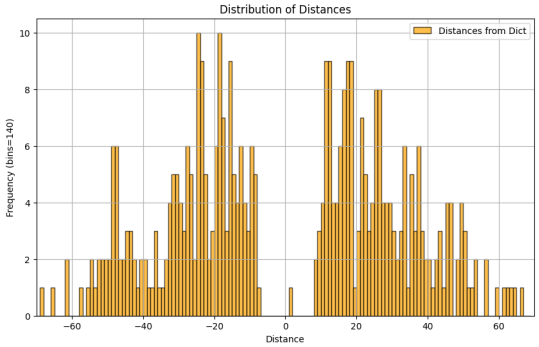

Figura 4.29: Distribución de las distacias calculadas desde los pixeles sin seed a la recta de ajuste de los clusters energéticos dentro de cada caja.

En la figura 4.29 se calcula la distacia entre los pixeles y la recta como hemos explicado. Se puede observar que los valores más frecuentes se encuentran en torno a ± 20 píxeles de la recta de ajuste. Lo que se puede apreciar también de la gráfica es que la frecuencia de valores de la distancia positivos decrece de forma más lineal que las distancias negativas. Para las distancias negativas decrecen de forma más rápida después de alcanzar el máximo y la gráfica parece más constante. Para decidir si una distancia es positiva o negativa, se compara la posición de la coordenada x del pixel sin seed con la del centro del cluster energético. Esto puede tener un inconveniente, ya que puede haber píxeles que se encuentren a la izquierda del centro del cluster pero se encuentran en la zona que está contaminada por CTI de acuerdo con la figura 1.4. Esto ocurre para los píxeles que se encuentren cerca del centro del cluster energético (se puede observar en la figura 4.28). No he encontrado una forma de separar las distancias como positivas y negativas en función de si se trata de halo o CTI. El centro del cluster se obtiene calculando la media entre las posiciones máximas y mínimas del cluster en ambos ejes.

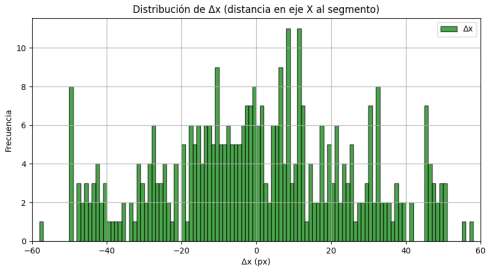

Figura 4.30: Distancias en el eje x desde los pixeles sin seed a la recta de ajuste de los clusters energéticos.

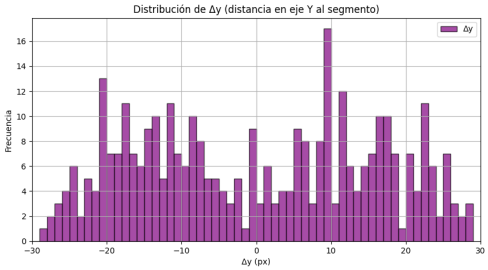

Figura 4.31: Distancias en el eje y desde los pixeles sin seed a la recta de ajuste de los clusters energéticos.

En las figuras 4.30 y 4.31, a diferencia de la anterior, se ha representado por separado las distancias en el eje x y las del eje y. En la figura 4.30 se puede observar que el máximo de las distancias positivas se encuentra en torno a 10 píxeles aproximadamente. Hay otros dos máximos en 30 y 45 más o menos. El máximo en 10 podría deberse al CTI. Sin embargo, los otros dos máximos también tienen una explicación: corresponden a píxeles situados a la derecha del cluster. Creemos que no se deben a CTI pero si pueden deberse a efectos del halo, de hecho, también se observa un comportamiento simétrico a la izquierda, con picos similares alrededor de las posiciones 30 y 45, como se aprecia en la imagen proporcionada. Para las distancias negativas la gráfica es bastante

constante entre 0 y -20 píxeles de distancia. Por otro lado, en la figura 4.31 hay un máximo claro en la distancia de 9 pixeles. Para las distancias negativas, la región con valores más frecuentes se encuentra en el intervalo de $[-10, -20]$ píxeles.
Cabe destacar que, para la generación de estas gráficas, si un mismo píxel se encuentra dentro de varias cajas (es decir, está cerca de varios clusters), se calculan las distancias desde ese píxel a las distintas rectas de ajuste asociadas a cada cluster. En ese caso, se comparan los módulos de todas las distancias obtenidas, y se conserva únicamente la menor de ellas, que se considera la más representativa.A partir de esa distancia mínima, se extraen las componentes en los ejes x e y (es decir, cuánto se aleja el píxel de la recta en cada dirección). Este procedimiento garantiza que ambas componentes (x e y) correspondan a la misma recta de ajuste. De este modo, se evita un problema importante: que para un mismo píxel se represente, por ejemplo, una distancia en el eje x calculada respecto a una recta, y una distancia en el eje y calculada respecto a otra distinta, lo cual distorsionaría la interpretación espacial de los datos.

Otra de las representaciones útiles es la de las distancias en el eje x frente a las distancias en el eje y, como se muestra en las figuras 4.32 y 4.33. En este caso, en lugar de ajustar el cluster a una recta, se ha calculado directamente la distancia de cada píxel al centro del cluster. En la figura 4.32 se representa la distancia en el eje x frente a su correspondiente distancia en el eje y, ambas en valor absoluto. Se observa que la mayoría de los puntos se concentran dentro del rectángulo $[10, 50] \times [10, 30]$, lo que indica que las distancias más comunes se sitúan entre los 10 y 50 píxeles en x, y entre los 10 y 30 en y. Un patrón similar se aprecia en la figura 4.33, incluso teniendo en cuenta las distancias negativas: la distribución es bastante simétrica respecto al centro, lo que refuerza la idea de una estructura regular del halo. Esta simetría y concentración permiten interpretar que el halo comienza a partir de los 10 píxeles de distancia al centro del cluster, se extiende aproximadamente hasta los 40 píxeles, y presenta colas largas más allá de ese rango. Esta observación está en línea con los estudios realizados hasta ahora sobre la distribución espacial del halo.
Con estas gráficas lo que se puede hacer es relacionarlas con el cluster energético al que se calcula la distancia. Para ello, se puede relacionar cada punto con la energía del cluster y su valor de $fpix$.

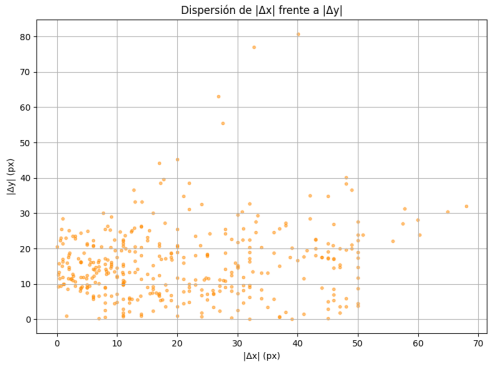

Figura 4.32: Representación de las distancias en X frente a las distancias en Y en valor absoluto

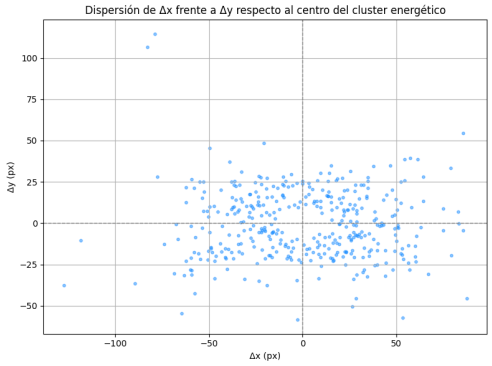

Figura 4.33: Representación de las distancias en X frente a las distancias en Y en positivo y negativo.

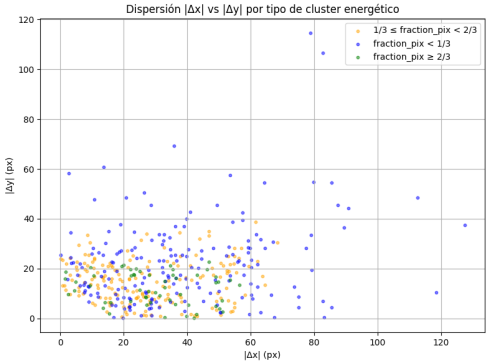

Figura 4.34: Representación de distancias en X vs distancias en Y en valor absoluto en función del valor de $fpix$ del cluster energético.

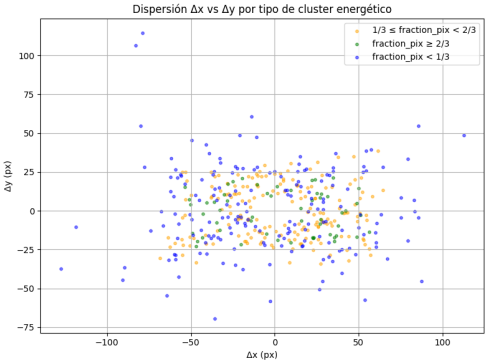

Figura 4.35: Representación de distancias en X vs distancias en Y en positivo y negativo en función del valor de $fpix$ del cluster energético.

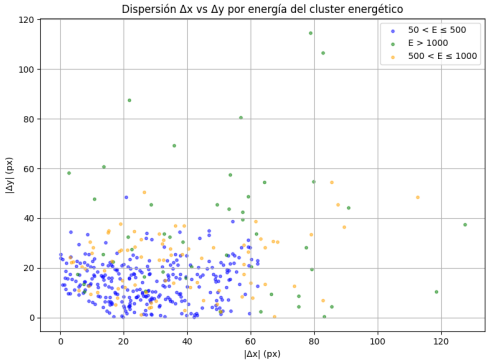

Figura 4.36: Representación de distancias en X vs distancias en Y en valor absoluto en función de la energía del cluster energético.

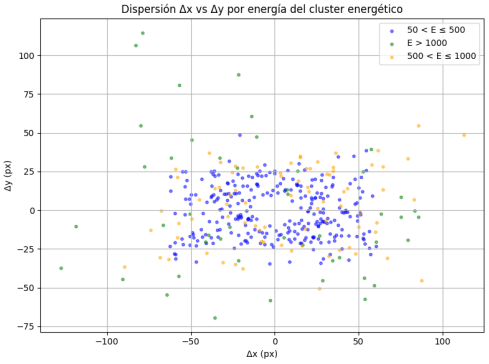

Figura 4.37: Representación de distancias en X vs distancias en Y en positivo y negativo en función de la energía del cluster energético.

En las figuras 4.34 y 4.35 se ha representado las mismas gráficas que la 4.32 y la 4.33 pero con distintos colores en función del valor de $fpix$. En azul se destacan los puntos para clusters con $fpix < 1/3$; en naranja, los de $1/3 \leq fpix < 2/3$ y en verde los valores de fracción de pixeles $> 2/3$. Se puede observar en ambas gráficas que los puntos verdes se concentran en distancias más bajas. Los puntos naranjas se expanden un poco más. Las máximas distancias se corresponden con los puntos azules. Esto tiene sentido ya que las trazas de clusters más alargadas (sin ser del todo horizontales o verticales) les corresponde un menor valor de $fpix$. Las cajas correspondientes a estos clusters serán más grandes que el resto. Además, al recorrer mayor distancia en el detector, la energía que va perdiendose se distribuye por más píxeles. Algo parecido se ha hecho para las figuras 4.36 y 4.37, pero en vez de diferenciar en rangos de fracción de píxeles se ha hecho por rangos de energías. Ocurre algo similar a las figuras 4.34 y 4.35. Los valores más grandes de las distancias se corresponden con los clusters más energéticos, y como habiamos visto en la figura 4.24, tiene $fpix$ mas pequeño correspondiendo a muones y electrones energéticos.

5 Conclusiones

Este trabajo se ha centrado en el estudio del fenómeno del halo en sensores Skipper CCDs del experimento Moskita, con el objetivo de caracterizar su alcance espacial, su comportamiento y su impacto en la identificación de señales de baja energía. El halo, generado por eventos de alta energía que dejan trazas alargadas en el detector, constituye una fuente relevante de fondo, especialmente crítica en la búsqueda de partículas exóticas como la materia oscura.

En primer lugar, para mejorar la limpieza de las imagenes de este experimento, en la sección 3.6 se desarrolló una función automatizada para detectar hot columns, es decir, columnas del CCD que presentan una actividad anómalamente alta y constante a lo largo de distintos runs. Esta herramienta resultó eficaz al compararse con otros métodos independientes y ya establecidos por la colaboración, permitiendo identificar las principales columnas defectuosas de forma sistemática y limpiar las imágenes antes del análisis físico. Puede ser un método prometedor de limpieza de imágenes optimizando los parámetros de ajuste. Este paso es fundamental para reducir los falsos positivos y obtener una caracterización más fiable del halo.

A continuación, estudié la distribución del halo alreadodor del cluster energético de varias maneras. En la sección 4.1 se propuso un método basado en cajas rectangulares centradas en cada cluster energético. Este enfoque permitió estudiar de forma sencilla y directa la distribución de píxeles sin seed dentro de una región ampliada alrededor del evento. Los resultados muestran que, en cajas con un solo cluster, la mayoría de los píxeles sin seed se concentran en los primeros 100 píxeles desde el centro. En cambio, cuando hay múltiples clusters dentro de la misma caja, se detectan píxeles a distancias mucho mayores (hasta 800–1200 píxeles), efecto atribuido a la superposición de halos y no a un solo evento extendido.

En la sección 4.2 se refinó este análisis utilizando anillos concéntricos centrados en el cluster energético, limitando el estudio al cuadrante inferior izquierdo para evitar la contaminación causada por el CTI. Se observó que la densidad de píxeles decrece rápidamente con la distancia al cluster, estabilizándose a partir de los 42–52 píxeles (figura 4.9). Este comportamiento se mantuvo de forma consistente en todos los runs y archivos analizados, indicando que el efecto del halo pierde influencia significativa más allá de esa distancia. Este resultado es coherente con el criterio adoptado habitualmente en la colaboración, que aplica una máscara circular de 40 píxeles alrededor del cluster tras haber enmascarado previamente los primeros 100 píxeles afectados por CTI.

La sección 4.3 introdujo una alternativa metodológica al análisis con anillos concéntricos: se emplearon anillos de radio fijo centrados en la posición de máxima carga (Qmax) del cluster. El objetivo era comprobar si la densidad de píxeles sin seed disminuía con la distancia. Sin embargo, los resultados no mostraron una tendencia clara; las densidades se mantuvieron bajas y relativamente constantes en todos los anillos. Esta falta de tendencia puede deberse a varios factores. Por un lado, el método podría estar fuertemente influenciado por la forma del propio cluster, que no siempre es simétrica ni regular respecto a la Qmax. Por otro lado, también puede haber interferencia de efectos externos, como el CTI o los halos de otros clusters cercanos, que podrían entrar dentro de los anillos definidos y distorsionar los resultados.

Por último realicé dos estudios novedosos para entender el halo, relacionado con la forma y la energía de los clusters. En la sección 4.4 se estudió la forma de los clusters energéticos mediante la definición de una nueva variable: la fracción de píxeles ocupados dentro del rectángulo mínimo que los contiene. Esta métrica permitió distinguir entre trazas alargadas (como las de muones, con fracción

baja) y trazas compactas (como las de partículas alfa o beta, con fracción alta). Esta caracterización morfológica aporta información relevante sobre la naturaleza del evento y sobre cómo se distribuye la carga en el detector.

Por último en la sección 4.5 se llevó a cabo un ajuste lineal de los clusters energéticos, con el fin de calcular las distancias perpendiculares desde los píxeles sin seed hasta la traza principal del evento. Se observó que estos píxeles se concentran típicamente a ± 20 píxeles de la recta ajustada, y que tanto su distribución como su número están relacionados con la energía y morfología del cluster. En particular, los halos más extendidos se asociaron a eventos más energéticos o morfológicamente más alargados. Este comportamiento abre la posibilidad de diseñar máscaras adaptativas que dependan de las características específicas del cluster—como su energía o forma—, lo cual podría resultar especialmente útil en imágenes con alta ocupación, permitiendo un tratamiento más preciso y eficiente del halo.

En base a los distintos enfoques desarrollados, se concluye que el halo producido por un cluster energético pierde su influencia directa, en general, a partir de una distancia de entre 50 y 60 píxeles. Esta distancia puede utilizarse como criterio para definir un radio de exclusión que permita enmascarar regiones contaminadas, reduciendo los falsos positivos sin comprometer la detección de señales genuinas de baja energía. No obstante, se ha observado que en el caso de partículas más energéticas o con una morfología más alargada (por ejemplo, cuando la energía se concentra en menos píxeles), el halo puede extenderse a distancias mayores. Por tanto, este criterio podría beneficiarse de ser ajustado dinámicamente según la energía o forma del cluster. En conjunto, estos resultados son extrapolables a otros experimentos que utilicen sensores Skipper CCDs, y pueden contribuir a mejorar la sensibilidad en la búsqueda de eventos raros mediante un tratamiento más preciso del halo.

Este trabajo abre una línea de investigación orientada al diseño de máscaras más precisas para el tratamiento del halo en imágenes obtenidas con sensores Skipper CCD. A partir de los resultados, se plantean varias direcciones futuras que podrían mejorar significativamente la limpieza y el análisis de los datos.
En primer lugar, sería conveniente desarrollar un criterio más robusto para separar las contribuciones del halo y del CTI, así como mejorar la asignación de píxeles a los clusters energéticos, evitando interferencias de clusters cercanos. También resulta interesante estudiar cómo varía la extensión del halo con la energía y morfología del cluster, especialmente en el régimen de baja energía (eV), y analizar cuántos clusters secundarios se generan a partir del halo.
Además, aplicar estos métodos a imágenes más limpias —como las obtenidas en laboratorios subterráneos (Modane, Canfranc) o con fuentes radiactivas— podría ayudar a identificar patrones característicos del halo y entender mejor su origen.
Todo ello permitiría desarrollar modelos más precisos de limpieza de imagen, aplicables a futuras campañas de adquisición de datos, e incluso generar simulaciones del fondo asociado al halo cuando no pueda eliminarse completamente.

En cuanto al aprendizaje personal, este trabajo me ha permitido aplicar numerosos conocimientos adquiridos a lo largo de la carrera, tanto teóricos como prácticos. Sin embargo, también me ha enfrentado a múltiples retos nuevos que han supuesto una ampliación significativa de mi formación. En primer lugar, me he introducido en el fenómeno de la materia oscura, un tema del que apenas tenía conocimiento previo, así como en los distintos tipos de experimentos que buscan detectarla, especialmente aquellos basados en búsquedas directas. He aprendido el funcionamiento detallado de los sensores Skipper CCDs y el tipo de señales que registran, así como aspectos técnicos del

experimento Moskita, que no formaban parte del temario habitual de la carrera.
También he adquirido experiencia en el procesamiento de imágenes científicas en formato FITS y en el manejo de herramientas específicas como ROOT, muy utilizadas en física experimental para el análisis de grandes volúmenes de datos.
Además, una parte especialmente enriquecedora ha sido el contacto con un grupo de investigación real —el grupo DAMIC en el IFCA—, lo que me ha permitido conocer de primera mano cómo se organiza y trabaja en un entorno científico colaborativo. Esta experiencia me ha resultado muy valiosa y podría ser determinante si en el futuro decido orientar mi carrera hacia la investigación.

Bibliografía

[1] A. Aguilar-Arevalo et al. Measurement of radioactive contamination in the high-resistivity silicon ccds of the damic experiment. *Journal of Instrumentation*, 10:P08014, 2015.

[2] R. Brun and F. Rademakers. Root — an object oriented data analysis framework. *Nuclear Instruments and Methods in Physics Research Section A*, 389(1–2):81–86, 1997.

[3] Mariano Rubén Cababie. *Origen y caracterización de eventos de un electrón en Skipper-CCDs para la búsqueda de materia oscura liviana.* Tesis doctoral, Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales, Buenos Aires, Argentina, 2023.

[4] Alberto Chavarria. Damic: Dark matter in ccds. `https://indico.in2p3.fr/event/35893/attachments/92148/140694/DAMIC_Chavarria.pdf`, 2024.

[5] Colaboración MOSKITA. Moskita-week. `https://moskita.web.cern.ch`, marzo 2025. Informe interno que describe el montaje experimental y los datos del prototipo MOSKITA con Skipper-CCDs en el LHC.

[6] Colaboración MOSKITA. Presentación sobre el experimento moskita. `https://docs.google.com/presentation/d/1BKzUX5_D8Y3AZvmaECbnuGbed9AZdMR0-rh0b3I_ed8/edit?pli=1&slide=id.g32065412158_0_0`, 2025.

[7] DAMIC-M Collaboration. Damic-m: Dark matter in ccds at modane. `https://damic.uchicago.edu/detector.php`.

[8] DAMIC-M Collaboration. First results from the damic-m low-background chamber: Dark matter–electron scattering with a skipper ccd. *arXiv preprint*, 2024.

[9] SENSEI Collaboration, O. Abramoff, et al. Improved constraints on sub-gev dark matter–electron scattering from the sensei experiment at snolab. *Phys. Rev. Lett.*, 134(1):011804, 2025.

[10] Claudia De Dominicis. Development of a prototype for the moskita experiment. Master's thesis, Università degli Studi di Milano, 2022.

[11] Marco Fabbrichesi, Emiliano Gabrielli, and Gaia Lanfranchi. The dark photon. *arXiv preprint*, 2020.

[12] Anastasiia Filimonova. Theory overview on dark matter portals. *ResearchGate*, 2023. Accedido en junio de 2025.

[13] A. Drlica-Wagner for the SENSEI Collaboration. First constraints from the full-scale sensei detector at snolab. In *Proceedings of Science (ICHEP2022)*, volume 444, page 1398, 2023.

[14] DGP Fotografía. Evolución de los sensores digitales en fotografía: de ccd a cmos, 2015.

[15] Manuel Estanislao Gaido. Fondo de fotones cherenkov para detectores de silicio de ultra bajo ruido en el espacio, 2024.

[16] D. Gallardo et al. The damic-m low-background chamber: a prototype skipper-ccd dark matter detector. *Journal of Instrumentation*, 19(11):T11010, 2024.

[17] María Pérez Martínez. Caracterización de aparatos de carga acoplada (ccds) con sistema de lectura skipper, 2023.

[18] W. D. Pence, L. Chiappetti, C. G. Page, R. A. Shaw, and E. Stobie. Definition of the flexible image transport system (fits), version 3.0. *Astronomy & Astrophysics*, 524:A42, 2010.

[19] Kartik Ramanathan. The sensei experiment at fermilab: Results and future prospects. Technical Report 2477324, OSTI (Office of Scientific and Technical Information), 2021.

[20] Kartik Ramanathan. *Charge-Coupled Devices for Low-Energy Particle Detection*. Ph.d. dissertation, University of Chicago, 2022.

[21] Joshua A. Sedlacek, Andrew Schwettmann, and Georg Raithel. Efficient microwave-to-optical conversion using rydberg atoms. *Physical Review Applied*, 17(1):014022, 2022.

[22] Teresa Marrodán Undagoitia. Dark matter detection. `https://indico.ifca.es/event/1199/contributions/9558/attachments/1326/1816/Marrodan_DM_2021_Santander.pdf`, 2021.

[23] Teresa Marrodán Undagoitia and Ludwig Rauch. Dark matter direct-detection experiments. *Annalen der Physik*, 528(5-6):397–417, 2016.

A Apéndices

A.1. Variables utilizadas para el análisis de los clusters

El fichero que hemos creado usando el programa ROOT y del cual hemos partido para nuestro análisis contiene información acerca de los clusteres y de los pixeles que contienen. A continuación se presenta una tabla con la descripción breve de cada variable empleada durante el análisis de clústeres:

Variable	Descripción
`Nclusters`	Número total de clústeres reconstruidos.
`DX`, `DY`	Elongación del clúster en los ejes X e Y, i.e., máx(x) − mín(x) y máx(y) − mín(y).
`Energy`	Energía total del clúster (keV).
`Energy_AVG`	Energía del clúster antes de la substracción del pedestal (ADU).
`Energy_PS`	Energía del clúster después de la substracción del pedestal (ADU).
`Npix`	Número total de píxeles que forman el clúster.
`PosX`, `PosY`	Posición promedio ponderada por energía en X e Y: `PosX` = `avg`$(x_i, w = pixels\_E)$.
`Qmax`	Carga máxima (keV) en un píxel del clúster.
`QmaxX`, `QmaxY`	Coordenadas del píxel con carga máxima. Índice empieza en 1.
`RMSX`, `RMSY`	Media del valor cuadrático de las posiciones X e Y, i.e., `mean`(x^2).
`STD_X`, `STD_Y`	Desviación estándar en las posiciones X e Y.
`STD_XY`	Desviación estándar combinada en XY.
`closest_cls_dist`	Distancia al clúster más cercano con seed.
`closest_cls_dist_X`, `closest_cls_dist_Y`	Componentes X e Y de la closest_cls_dist.
`closest_cls_id`	ID del clúster más cercano con seed.
`cluster_id`	Identificador único del clúster.
`dwSTD_X`, `dwSTD_Y`	Desviación estándar "normalizada ponderada".
`wSTD_X`, `wSTD_Y`, `wSTD_XY`	Desviaciones estándar ponderadas por energía.
`fSTD_X`, `fSTD_Y`	Sigma obtenida de un ajuste gaussiano a la proyección 1D del perfil del clúster.
`fPosX`, `fPosY`	Posición final del clúster tras ajustes o filtros.
`has_seed`	Indicador de si el clúster tiene un píxel semilla (`1` si lo tiene).
`hotreg`	El clúster contiene al menos un píxel en una región caliente".
`maxX`, `maxY`	Máximas coordenadas X e Y del clúster.
`meanX`, `meanY`	Promedio simple de las posiciones X e Y de los píxeles del clúster.
`minX`, `minY`	Mínimas coordenadas X e Y del clúster.
`nfile`	ID del archivo FITS del que se extrajo el clúster.
`pixels_E`	Energía (keV) de cada píxel que compone el clúster.
`pixels_E_AVG`, `pixels_E_PS`	Energía promedio y post-substracción de pedestal por píxel.
`pixels_x`, `pixels_y`, `pixels_z`	Coordenadas de los píxeles del clúster (x: columnas, y: filas, z: capas si aplica).
`ppcol`	Categoría o tipo del clúster tras procesamiento.

Variable	Descripción (continuación)
`projectionX`, `projectionY`	Proyección del clúster en los ejes X e Y.
`projectionX_cols`, `projectionY_rows`	Dimensiones de las proyecciones.
`qmin_cluster_seed`	Carga mínima requerida para que un píxel sea semilla de clúster.
`readout_start`, `readout_end`	Tiempos de inicio y fin de lectura del clúster.
`seed_nsig_delta`	Diferencia entre los niveles de sigma máximo y mínimo del píxel semilla.
`seed_nsig_max`, `seed_nsig_min`	Valores máximo y mínimo de sigma del píxel semilla.
`seed_nsig_pixels`	Número de píxeles del clúster dentro del umbral de sigma semilla.
`sigma_seed_eV`	Sigma (ruido) en unidades de eV para el píxel semilla.

A.2. Extensión de los estudios hecho a todos los datos analizados

En este apendice mostramos los estudios presentados en este trabajo pero para todos los datos recogidos con Moskita durante el tiempo que ha estado activamente tomando datos. Los detalles de cada conjunto de datos se pueden ver en la Tabla **??**. En primer lugar, se muestran las gráficas adicionales de la sección 3.6 dedicada a las hot columns.

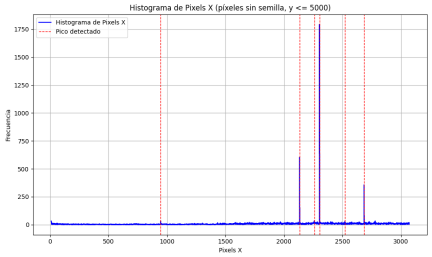

Figura A.1: Distribución de la variable pixelsX con las hot columns detectadas por la función en rojo para el run4.

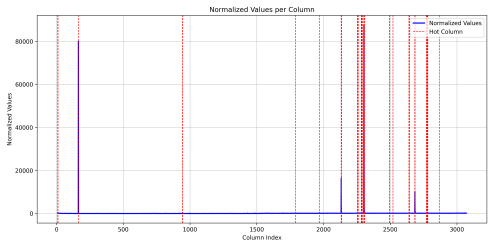

Figura A.2: Hot columns detectadas en rojo por el otro método para el run4.

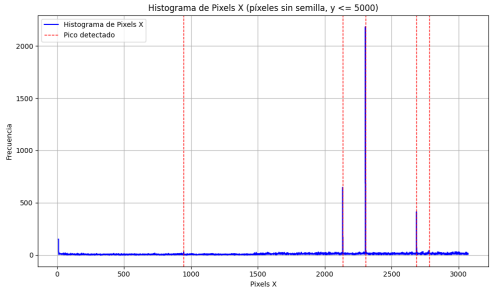

Figura A.3: Distribución de la variable pixelsX con las hot columns detectadas por la función en rojo para el run5.

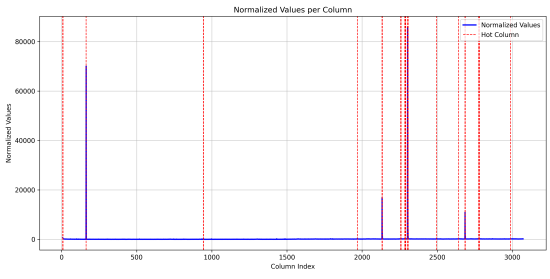

Figura A.4: Hot columns detectadas en rojo por el otro método para el run5.

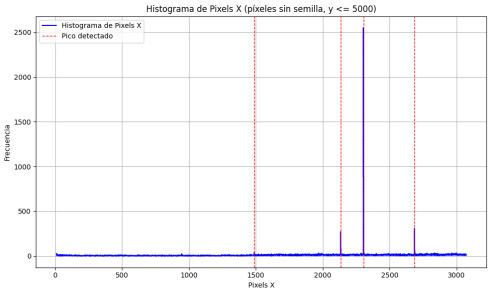

Figura A.5: Distribución de la variable pixelsX con las hot columns detectadas por la función en rojo para el run6.

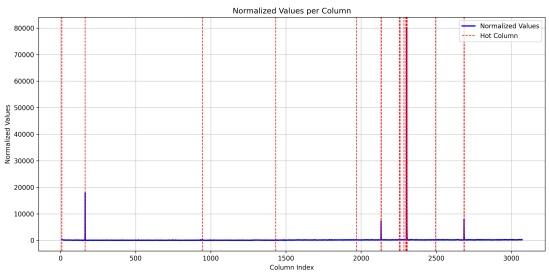

Figura A.6: Hot columns detectadas en rojo por el otro método para el run6.

A continuación, se muestran las gráficas obtenidas en la sección para densidades dentro de las cajas (4.1):

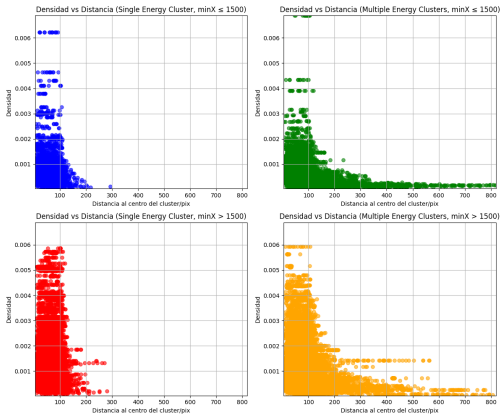

Figura A.7: Representación de las densidades frente a la distancia al centro del cluster correspondiente para el otro archivo del run1.

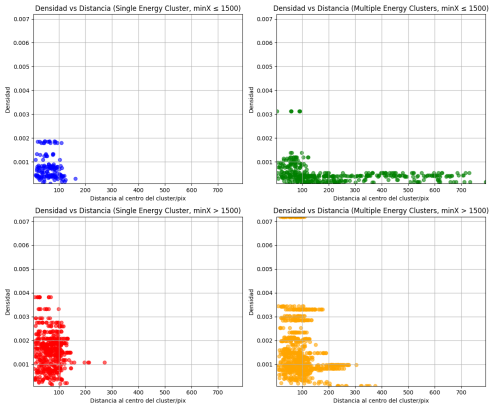

Figura A.8: Representación de las densidades frente a la distancia al centro del cluster correspondiente para el archivo run3.

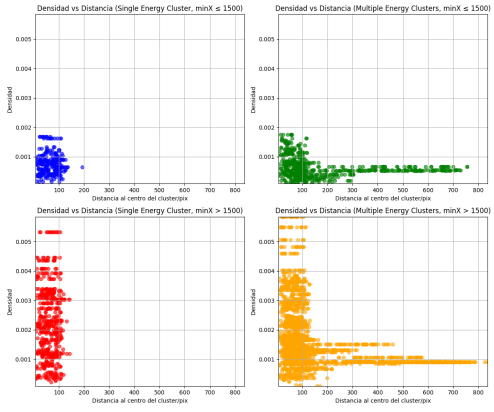

Figura A.9: Representación de las densidades frente a la distancia al centro del cluster correspondiente para el archivo run4.

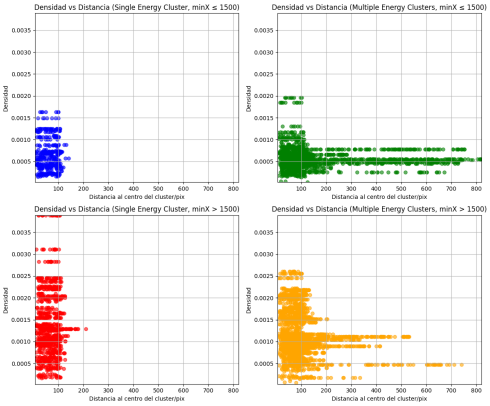

Figura A.10: Representación de las densidades frente a la distancia al centro del cluster correspondiente para el archivo run6.

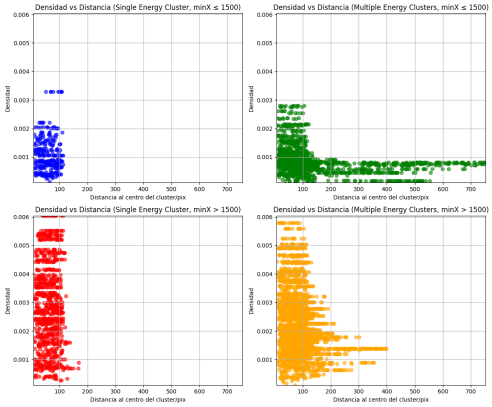

Figura A.11: Representación de las densidades frente a la distancia al centro del cluster correspondiente para el archivo run6.

Para la sección 4.2 dedicaca al cálculo de densidades de píxeles por anillos, se han realizado estas gráficas para otros archivos:

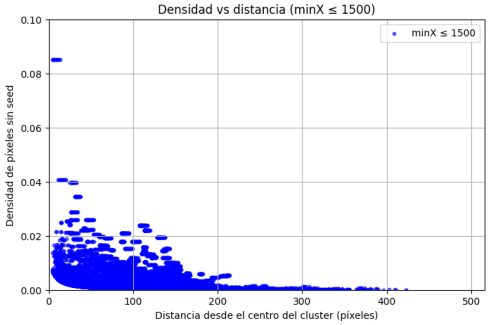

Figura A.12: Representación de las densidades medias frente al orden de anillo correspondiente para el otro archivo del run1 para la primera mitad de la imagen.

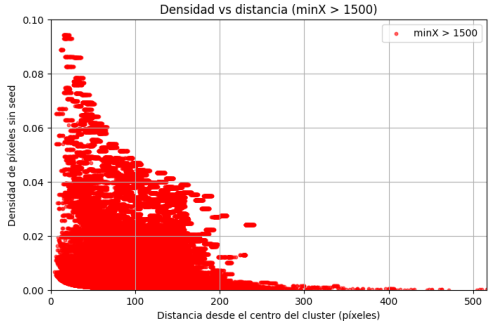

Figura A.13: Representación de las densidades medias frente al orden de anillo correspondiente para el otro archivo del run1 para la segunda mitad de la imagen.

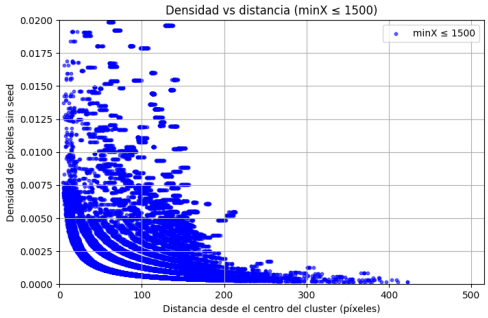

Figura A.14: Representación de las densidades medias frente al orden de anillo correspondiente para el otro archivo del run1 para la primera mitad de la imagen haciendo zoom en el eje y hasta 0,02.

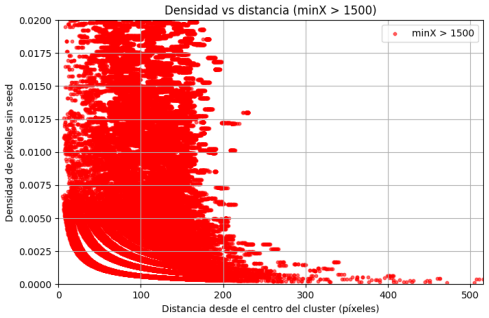

Figura A.15: Representación de las densidades medias frente al orden de anillo correspondiente para el otro archivo del run1 para la segunda mitad de la imagen haciendo zoom en el eje y hasta 0,02.

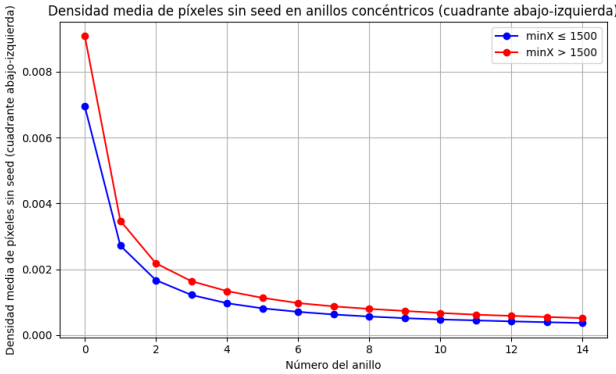

Figura A.16: Representación de las densidades medias frente al orden de anillo correspondiente para el otro archivo del run1.

Para la seccion 4.3 que trata de un cáculo alternativo de la densidad de píxeles por anillos, se han realizado estas gráficas:

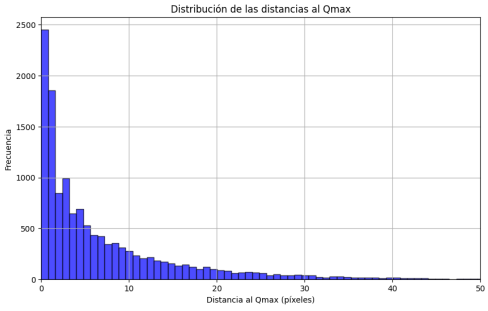

Figura A.17: Distribución de distancias entre el centro geométrico del cluster y la posición de Qmax para los clusters energéticos en el otro archivo del run1.

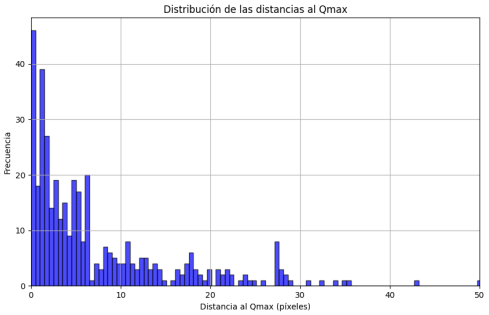

Figura A.18: Distribución de distancias entre el centro geométrico del cluster y la posición de Qmax para los clusters energéticos en el archivo run3.

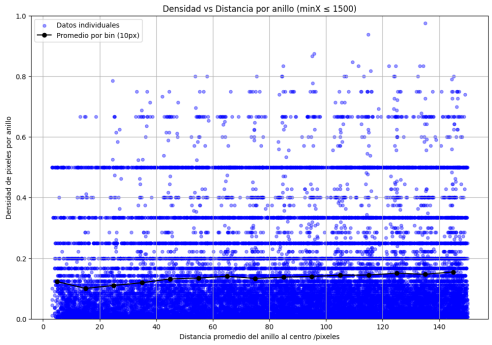

Figura A.19: Representación de las densidades frente a las distancias promedio para la segunda mitad de la imagen en el otro archivo del run1.

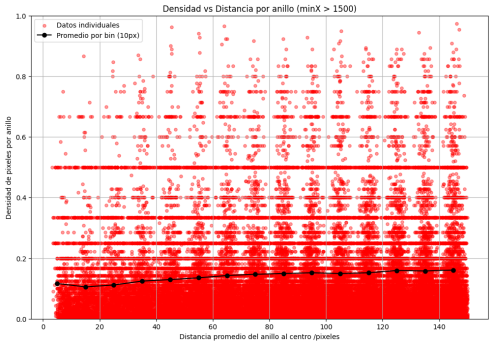

Figura A.20: Representación de las densidades frente a las distancias promedio para la segunda mitad de la imagen en el otro archivo del run1.

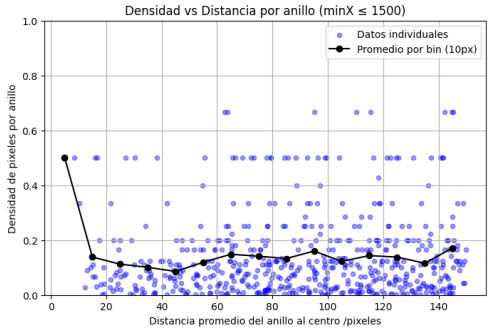

Figura A.21: Representación de las densidades frente a las distancias promedio para la primera mitad de la imagen en el archivo run3.

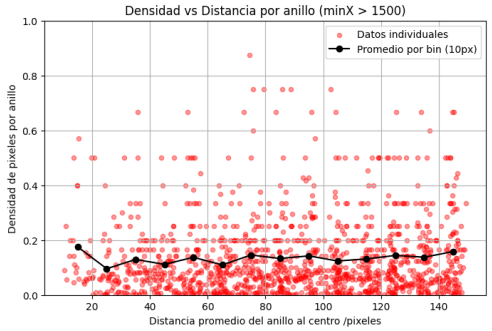

Figura A.22: Representación de las densidades frente a las distancias promedio para la segunda mitad de la imagen en el archivo run3.

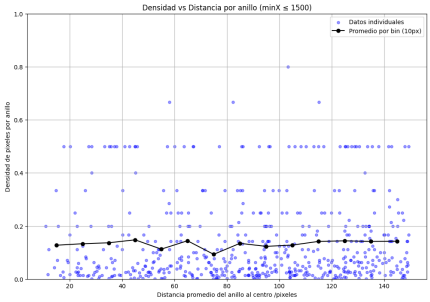

Figura A.23: Representación de las densidades frente a las distancias promedio para la primera mitad de la imagen en el archivo run3 en el cuadrante inferior-derecha.

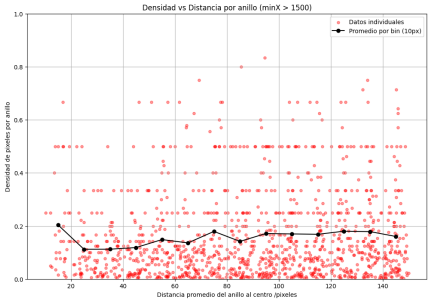

Figura A.24: Representación de las densidades frente a las distancias promedio para la segunda mitad de la imagen en el archivo run3 en el cuadrante inferior-derecha.

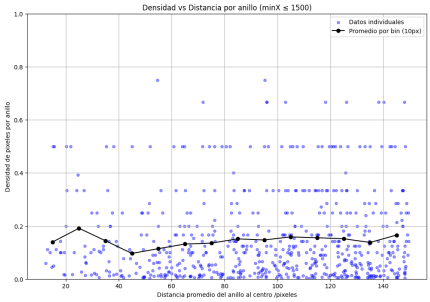

Figura A.25: Representación de las densidades frente a las distancias promedio para la primera mitad de la imagen en el archivo run3 en el cuadrante superior-derecha.

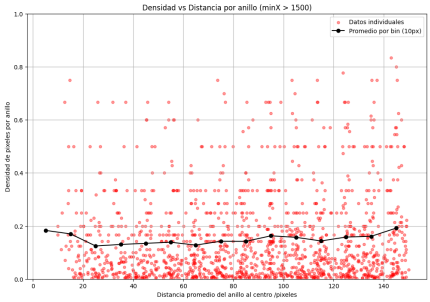

Figura A.26: Representación de las densidades frente a las distancias promedio para la segunda mitad de la imagen en el archivo run3 en el cuadrante superior-derecha.

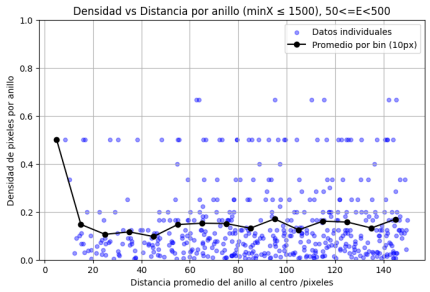

Figura A.27: Representación de las densidades frente a las distancias promedio para la primera mitad de la imagen en el archivo run3 para clusters con energías entre 50 y 500 keV.

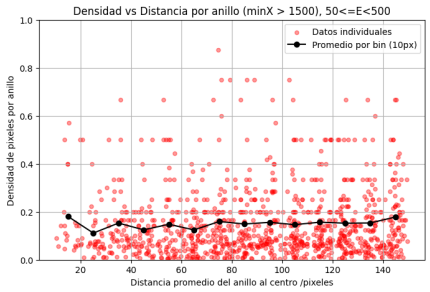

Figura A.28: Representación de las densidades frente a las distancias promedio para la segunda mitad de la imagen en el archivo run3 para clusters con energías entre 50 y 500 keV.

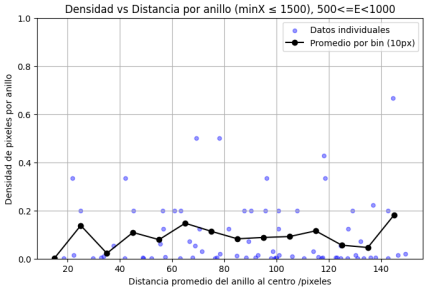

Figura A.29: Representación de las densidades frente a las distancias promedio para la primera mitad de la imagen en el archivo run3 para clusters con energías entre 500 y 1000 keV.

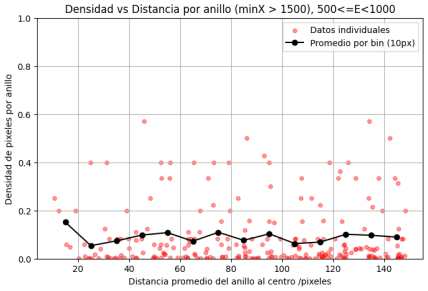

Figura A.30: Representación de las densidades frente a las distancias promedio para la segunda mitad de la imagen en el archivo run3 para clusters con energías entre 500 y 1000 keV.

Finalmente, para la sección 4.4 dedicada a relacionar la forma del cluster con su energía se han realizado estas gráficas:

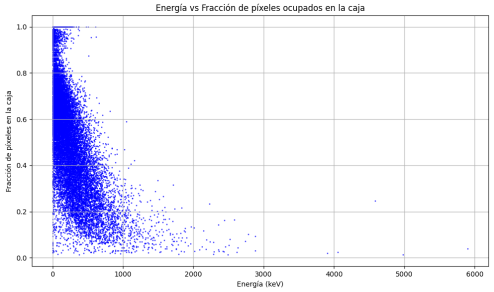

Figura A.31: Representacion de la fracción de píxeles en funcion de la energía del cluster para el archivo run3.

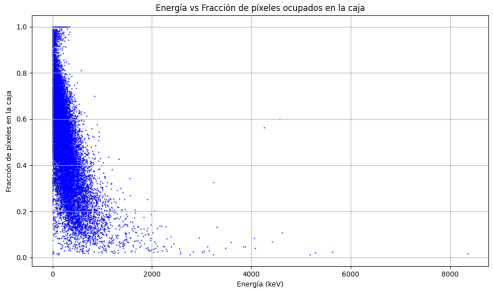

Figura A.32: Representacion de la fracción de píxeles en funcion de la energía del cluster para el archivo run4.

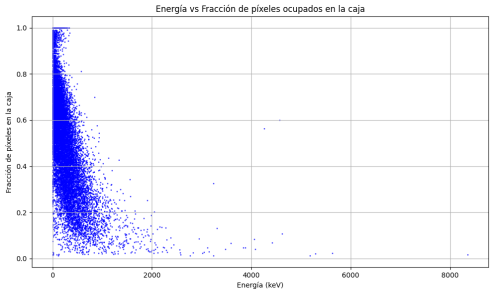

Figura A.33: Representacion de la fracción de píxeles en funcion de la energía del cluster para el archivo run5.

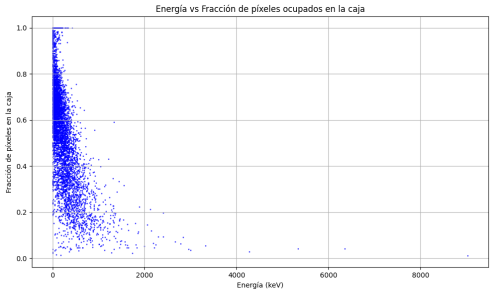

Figura A.34: Representacion de la fracción de píxeles en funcion de la energía del cluster para el archivo run6.

A.3. Códigos desarrollados para este trabajo

A continuación se muestran los códigos creados en PYTHON dentro del entorno de ROOT para analizar los datos usados en este trabajo, y para conseguir los resultados mostrados.

A.3.1. Función para detectar hot columns (sección 3.6)

Esta es la función 'get hot column pixels' creada para detectar hot columns en las imágenes analizadas (como se ha explicado en la sección 3.6).

```
import ROOT
import numpy as np

def get_hot_column_pixels(clustersRec, range_around_hot_column=1,
    peak_threshold=0.05):
    """
    Detecta hot columns (columnas con exceso de p xeles) en un objeto
       clustersRec,
    y devuelve los p xeles pertenecientes a esas columnas, restringidos a
        y <= 5000.
    Tambi n devuelve el histograma de las posiciones X y los picos
       detectados con TSpectrum.
    """
    nentries = clustersRec.GetEntries()  # N mero total de entradas (
       eventos) en el  rbol  ROOT
    all_pixels_x = []  # Lista para almacenar todas las coordenadas X de
       p xeles sin seed (de todos los eventos)

    # Primer bucle: recolecta todas las coordenadas X de p xeles sin seed
       , limitados a y <= 5000
    for entry in range(nentries):
        clustersRec.GetEntry(entry)
        has_seed = np.array([clustersRec.has_seed[i] for i in range(
           clustersRec.has_seed.size())], dtype=int)
        pixels_x = [np.array(clustersRec.pixels_x[i], dtype=int) for i in
           range(has_seed.size)]
        pixels_y = [np.array(clustersRec.pixels_y[i], dtype=int) for i in
           range(has_seed.size)]

        for i in range(has_seed.size):
            if has_seed[i] == 0:  # Solo considera p xeles sin seed
                valid_pixels = pixels_x[i][pixels_y[i] <= 5000]  # Filtra
                   los que cumplen y <= 5000
                all_pixels_x.extend(valid_pixels)  # A ade a la lista
                   global

    if not all_pixels_x:
        return {}  # Si no se encontraron p xeles, devuelve diccionario
           vac o

    # Crear un histograma ROOT para contar cu ntos p xeles caen en cada
       coordenada X
    x_min, x_max = min(all_pixels_x), max(all_pixels_x)
    hist_x = ROOT.TH1D("hist_x", "Distribuci n de p xeles X", 2000,
       x_min, x_max)

    for x in all_pixels_x:
        hist_x.Fill(x)  # Llenar el histograma con las posiciones X
           recolectadas

```

```
    # Usar TSpectrum para buscar picos en el histograma (posibles hot columns)
    spectrum = ROOT.TSpectrum()
    nPeaks = spectrum.Search(hist_x, 1, "", peak_threshold)  # Detecta picos con un umbral dado
    xPeaks = np.frombuffer(spectrum.GetPositionX(), dtype=np.float64, count=nPeaks) if nPeaks > 0 else np.array([])
    xPeaks = np.round(xPeaks).astype(int)  # Redondear las posiciones X de los picos a enteros

    # Expandir los picos detectados para incluir las columnas vecinas (seg n rango definido)
    hot_columns = set()
    for peak in xPeaks:
        hot_columns.update([peak - range_around_hot_column, peak, peak + range_around_hot_column])

    # Inicializar listas para guardar los p xeles identificados como pertenecientes a hot columns
    hot_pixels_x, hot_pixels_y, hot_pixels_E = [], [], []

    # Segundo bucle: recopilar coordenadas y energ as de los p xeles que est n en hot columns
    for entry in range(nentries):
        clustersRec.GetEntry(entry)
        has_seed = np.array([clustersRec.has_seed[i] for i in range(clustersRec.has_seed.size())], dtype=int)
        pixels_x = [np.array(clustersRec.pixels_x[i], dtype=int) for i in range(has_seed.size)]
        pixels_y = [np.array(clustersRec.pixels_y[i], dtype=int) for i in range(has_seed.size)]
        pixels_E = [np.array(clustersRec.pixels_E[i], dtype=float) for i in range(has_seed.size)]

        for i in range(has_seed.size):
            if has_seed[i] == 0:  # Solo considera p xeles sin seed
                mask_hot = np.isin(pixels_x[i], list(hot_columns)) & (pixels_y[i] <= 5000)  # Selecciona p xeles en hot columns y y <= 5000
                if np.any(mask_hot):
                    hot_pixels_x.append(pixels_x[i][mask_hot])  # Guarda coordenadas X
                    hot_pixels_y.append(pixels_y[i][mask_hot])  # Guarda coordenadas Y
                    hot_pixels_E.append(pixels_E[i][mask_hot])  # Guarda energ as

    # Devuelve un diccionario con los resultados recolectados
    return {
        "pixels_x": hot_pixels_x,
        "pixels_y": hot_pixels_y,
        "pixels_E": hot_pixels_E,
        "histogram": hist_x,
        "x_peaks": xPeaks
```

```
72      }
```

Listing A.1: Código para detectar hot columns

A.3.2. Código empleado en la sección 4.1

A partir de este código se ha llevado a cabo el cálculo de las densidades de píxeles dentro de cajas en relación a los clusters energéticos.

```
1  # Importar librer as necesarias
2  import ROOT
3  from ROOT import TFile
4  import numpy as np
5  import matplotlib.pyplot as plt
6  from hot_columns import get_hot_column_pixels  # Funci n externa para
       detectar columnas "calientes"
7
8  # Abrir el archivo ROOT que contiene los datos de clusters
9  file = TFile("/Users/isma/Desktop/universidad/Cuarto/Tfg/
       hadd_01APR2024_panaSKImg_clustersRec_pp_proc_skp_ext2.root")
10 clustersRec = file.Get("clustersRec")  # Obtener  rbol  de clusters
11 nentries = clustersRec.GetEntries()    # Total de eventos en el  rbol
12
13 # Obtener datos de columnas calientes
14 hot_column_data = get_hot_column_pixels(clustersRec) if
       get_hot_column_pixels(clustersRec) else {}
15 hot_columns = set(map(tuple, hot_column_data.get("pixels_x", []))) if
       hot_column_data else set()
16
17 # Listas para guardar distancias y densidades clasificadas por regi n y
       tipo de vecindad energ tica
18 distances_low_single_energy, densities_low_single_energy = [], []
19 distances_low_multiple_energy, densities_low_multiple_energy = [], []
20 distances_high_single_energy, densities_high_single_energy = [], []
21 distances_high_multiple_energy, densities_high_multiple_energy = [], []
22
23 # Funci n para verificar si un cluster contiene alg n p xel dentro de
       una columna caliente
24 def contains_hot_column_pixels(pixels_x):
25     return any(x in hot_columns for x in pixels_x)
26
27 # Funci n para determinar si hay m s de un cluster energ tico (>50)
       cerca del actual
28 def is_multiple_energy_clusters(cluster_index, minX, maxX, minY, maxY,
       energies, margin_x, margin_y):
29     # Definir m rgenes ampliados alrededor del cluster
30     left_limit, right_limit = minX[cluster_index] - margin_x, maxX[
           cluster_index] + margin_x
31     bottom_limit, top_limit = minY[cluster_index] - margin_y, maxY[
           cluster_index] + margin_y
32
33     # Contar cu ntos clusters energ ticos hay en esa vecindad
34     count = sum(
35         1 for i in range(len(energies))
```

```
        if energies[i] > 50 and left_limit <= minX[i] <= right_limit and
            bottom_limit <= minY[i] <= top_limit
    )
    return count > 1

# Iterar sobre todos los eventos
for entry in range(nentries):
    clustersRec.GetEntry(entry)

    # Extraer datos de cada cluster en el evento
    cluster_ids = np.array(clustersRec.cluster_id)
    has_seed = np.array(clustersRec.has_seed)
    pixels_x = [np.array(clustersRec.pixels_x[i]) for i in range(len(
        cluster_ids))]
    pixels_y = [np.array(clustersRec.pixels_y[i]) for i in range(len(
        cluster_ids))]
    minX = np.array(clustersRec.minX)
    maxX = np.array(clustersRec.maxX)
    minY = np.array(clustersRec.minY)
    maxY = np.array(clustersRec.maxY)
    NPix = np.array(clustersRec.Npix)
    energies = np.array(clustersRec.Energy)

    # Excluir clusters con p xeles en columnas calientes
    valid_clusters = [i for i in range(len(cluster_ids)) if not
        contains_hot_column_pixels(pixels_x[i])]

    # Clasificar clusters energ ticos por regi n
    high_energy_low = [
        i for i in valid_clusters if (energies[i] > 50) and (has_seed[i]
            == 1) and
        (minX[i] <= 1500) and np.any(pixels_y[i] <= 5000)
    ]
    high_energy_high = [
        i for i in valid_clusters if (energies[i] > 50) and (has_seed[i]
            == 1) and
        (minX[i] > 1500) and np.any(pixels_y[i] <= 5000)
    ]

    # Clasificar clusters sin seed (potenciales vecinos)
    no_seed_low = [
        i for i in valid_clusters if (has_seed[i] == 0) and (minX[i] <=
            1500) and np.any(pixels_y[i] <= 5000)
    ]
    no_seed_high = [
        i for i in valid_clusters if (has_seed[i] == 0) and (minX[i] >
            1500) and np.any(pixels_y[i] <= 5000)
    ]

    # Funci n para procesar clusters energ ticos y calcular densidades
        de vecinos sin seed a su izquierda
    def process_clusters(high_energy_clusters, no_seed_clusters,
                         distances_list_single, densities_list_single,
                         distances_list_multiple, densities_list_multiple)
```

```
                                :
81
82          for j in high_energy_clusters:
83              # Definir m rgenes de b squeda ampliados en X e Y
84              margin_x = max(100, (maxX[j] - minX[j]) * 1.2)
85              margin_y = max(20, (maxY[j] - minY[j]) * 1.2)
86              left_limit, right_limit = minX[j] - margin_x, maxX[j] +
                    margin_x
87              bottom_limit, top_limit = minY[j] - margin_y, maxY[j] +
                    margin_y
88
89              # Centroide del cluster energ tico
90              center_x = np.mean(pixels_x[j])
91              center_y = np.mean(pixels_y[j])
92
93              #  rea  total donde se buscan los p xeles vecinos (a la
                    izquierda del centro)
94              area_left = (center_x - left_limit) * (center_y - bottom_limit
                    )
95
96              # Verificar si hay m ltiples clusters energ ticos en la
                    vecindad
97              multiple_energy_clusters = is_multiple_energy_clusters(j, minX
                    , maxX, minY, maxY, energies, margin_x, margin_y)
98
99              pixels_left, distances = [], []
100
101             # Buscar p xeles sin seed en la regi n a la izquierda del
                    cluster
102             for i in no_seed_clusters:
103                 for x, y in zip(pixels_x[i], pixels_y[i]):
104                     if (left_limit <= x < minX[j]) and (bottom_limit <= y
                            <= top_limit) and \
105                        (x < center_x and y < center_y) and (NPix[i] <= 2):
106                         pixels_left.append((x, y))
107                         distances.append(np.sqrt((x - center_x) ** 2 + (y
                                - center_y) ** 2))
108
109             # Calcular densidad como #pixeles vecinos /  rea  de b squeda
110             density = len(pixels_left) / area_left if area_left > 0 else 0
111
112             # Guardar resultados en listas correspondientes (dependiendo
                    de si hay m ltiples clusters o no)
113             if multiple_energy_clusters:
114                 densities_list_multiple.extend([density] * len(distances))
115                 distances_list_multiple.extend(distances)
116             else:
117                 densities_list_single.extend([density] * len(distances))
118                 distances_list_single.extend(distances)
119
120     # Procesar regiones izquierda (low) y derecha (high)
121     process_clusters(high_energy_low, no_seed_low,
122                      distances_low_single_energy,
                            densities_low_single_energy,
```

```
                    distances_low_multiple_energy,
                        densities_low_multiple_energy)

    process_clusters(high_energy_high, no_seed_high,
                    distances_high_single_energy,
                        densities_high_single_energy,
                    distances_high_multiple_energy,
                        densities_high_multiple_energy)

# === Graficar los resultados ===
plt.figure(figsize=(12, 10))

# Determinar los l mites comunes de los ejes
all_distances = distances_low_single_energy +
    distances_low_multiple_energy + distances_high_single_energy +
    distances_high_multiple_energy
all_densities = densities_low_single_energy +
    densities_low_multiple_energy + densities_high_single_energy +
    densities_high_multiple_energy

x_min, x_max = min(all_distances), max(all_distances)
y_min, y_max = min(all_densities), max(all_densities)

# Subgr fico 1: regi n izquierda, un solo cluster energ tico
plt.subplot(2, 2, 1)
plt.scatter(distances_low_single_energy, densities_low_single_energy,
    alpha=0.6, color='blue')
plt.xlabel("Distancia al centro del cluster/pix")
plt.ylabel("Densidad")
plt.title("Densidad vs Distancia (Single Energy Cluster, minX     1500)")
plt.grid(True)
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)

# Subgr fico 2: regi n izquierda, m ltiples clusters energ ticos
plt.subplot(2, 2, 2)
plt.scatter(distances_low_multiple_energy, densities_low_multiple_energy,
    alpha=0.6, color='green')
plt.xlabel("Distancia al centro del cluster/pix")
plt.ylabel("Densidad")
plt.title("Densidad vs Distancia (Multiple Energy Clusters, minX     1500)
    ")
plt.grid(True)
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)

# Subgr fico 3: regi n derecha, un solo cluster energ tico
plt.subplot(2, 2, 3)
plt.scatter(distances_high_single_energy, densities_high_single_energy,
    alpha=0.6, color='red')
plt.xlabel("Distancia al centro del cluster/pix")
plt.ylabel("Densidad")
plt.title("Densidad vs Distancia (Single Energy Cluster, minX > 1500)")
plt.grid(True)
```

```
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)

# Subgr fico 4: regi n derecha, m ltiples clusters energ ticos
plt.subplot(2, 2, 4)
plt.scatter(distances_high_multiple_energy, densities_high_multiple_energy
    , alpha=0.6, color='orange')
plt.xlabel("Distancia al centro del cluster/pix")
plt.ylabel("Densidad")
plt.title("Densidad vs Distancia (Multiple Energy Clusters, minX > 1500)")
plt.grid(True)
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)

plt.tight_layout()
plt.show()
```

Listing A.2: Código empleado para calcular las densidades de pixeles por cajas en la sección 4.1

A.3.3. Código empleado en la sección 4.2

Con este código se ha calculado la densidad de píxeles dentro de anillos concéntricos centrados en los clusters energéticos teniendo en cuenta el área de los anillos.

```
import ROOT  # Importa ROOT para manejar archivos .root (com n en f sica
    de part culas)
from ROOT import TFile
import numpy as np  # Importa numpy para trabajar con arrays
import matplotlib.pyplot as plt  # Importa matplotlib para generar
    gr ficas

# Abrir archivo ROOT
file = TFile("/Users/isma/Desktop/universidad/Cuarto/Tfg/
    hadd_01APR2024_panaSKImg_clustersRec_pp_proc_skp_ext3.root")
clustersRec = file.Get("clustersRec")  # Accede al  rbol  llamado '
    clustersRec'
nentries = clustersRec.GetEntries()  # Obtiene el n mero de entradas en
    el  rbol

# Listas para almacenar los resultados de distancias y densidades
distances_low, densities_low = [], []
distances_high, densities_high = [], []

# Iterar sobre todas las entradas del  rbol
for entry in range(nentries):
    clustersRec.GetEntry(entry)  # Cargar la entrada actual

    # Extraer informaci n del  rbol  como arrays numpy
    cluster_ids = np.array(clustersRec.cluster_id)
    has_seed = np.array(clustersRec.has_seed)
    pixels_x = [np.array(clustersRec.pixels_x[i]) for i in range(len(
        cluster_ids))]
    pixels_y = [np.array(clustersRec.pixels_y[i]) for i in range(len(
        cluster_ids))]
```

```
    minX = np.array(clustersRec.minX)
    NPix = np.array(clustersRec.Npix)
    energies = np.array(clustersRec.Energy)
    hotregions = np.array(clustersRec.hotreg)

    # Seleccionar  ndices  de clusters con alta energ a y clusters sin
       seed
    high_energy_low = np.where((energies > 50) & (has_seed == 1) & (minX
       <= 1500))[0]
    high_energy_high = np.where((energies > 50) & (has_seed == 1) & (minX
       > 1500))[0]
    no_seed_low = np.where((has_seed == 0) & (minX <= 1500))[0]
    no_seed_high = np.where((has_seed == 0) & (minX > 1500))[0]

    # Funci n para procesar clusters y calcular densidades en anillos
    def process_clusters(high_energy_clusters, no_seed_clusters,
       distances_list, densities_list):
        for j in high_energy_clusters:
            center_x = np.mean(pixels_x[j])  # Calcular centro en X
            center_y = np.mean(pixels_y[j])  # Calcular centro en Y
            pixE = set(zip(pixels_x[j], pixels_y[j]))  # Conjunto de
               p xeles del cluster energ tico

            # Encontrar el  ndice  del p xel con el valor m nimo en X (
               minX)
            min_index = np.where(pixels_x[j] == minX[j])[0][0]
            x_min = pixels_x[j][min_index]
            y_min = pixels_y[j][min_index]

            # Calcular radio interno (r_inner) como la distancia del
               centro al p xel con minX
            r_inner = np.sqrt((x_min - center_x) ** 2 + (y_min - center_y)
                ** 2)

            # Iterar sobre 15 anillos conc ntricos alrededor del centro
            for i in range(15):
                r_outer = r_inner + 10  # Radio externo del anillo
                area_ring = (np.pi / 4) * (r_outer**2 - r_inner**2)  #
                    rea  del anillo
                count_pixels = 0  # Contador de p xeles sin seed dentro
                   del anillo
                pixel_distances = []  # Lista para almacenar las
                   distancias

                # Recorrer clusters sin seed
                for k in no_seed_clusters:
                    if hotregions[k] != 0 or k == j:  # Saltar si est  en
                        hotregion o es el mismo cluster
                        continue
                    for x, y in zip(pixels_x[k], pixels_y[k]):
                        if (x, y) in pixE:  # Saltar si ya pertenece al
                           cluster energ tico
                            continue
                        distance = np.sqrt((x - center_x)**2 + (y -
```

```
                                center_y)**2)
65                          if r_inner <= distance < r_outer and x < center_x
                                and y < center_y:
66                              count_pixels += 1
67                              pixel_distances.append(distance)
68
69                  # Calcular densidad como n mero de p xeles dividido
                        entre el  rea
70                  density = count_pixels / area_ring if area_ring > 0 else 0
71
72                  # Asignar densidad a todos los p xeles encontrados en
                        este anillo
73                  for d in pixel_distances:
74                      distances_list.append(d)
75                      densities_list.append(density)
76
77                  r_inner = r_outer  # Actualizar radio interno para el
                        siguiente anillo
78
79      # Procesar ambos grupos (minX bajo y alto)
80      process_clusters(high_energy_low, no_seed_low, distances_low,
            densities_low)
81      process_clusters(high_energy_high, no_seed_high, distances_high,
            densities_high)
82
83  # Determinar l mites para las gr ficas
84  x_max = max(max(distances_low), max(distances_high))
85  x_min = min(min(distances_low), min(distances_high))
86  print(f"Rango de distancias: {x_min:.2f} - {x_max:.2f} p xeles")
87
88  # Gr fica para clusters con minX     1500
89  plt.figure(figsize=(8, 5))
90  plt.scatter(distances_low, densities_low, alpha=0.6, s=10, color="blue",
        label="minX     1500")
91  plt.xlabel("Distancia desde el centro del cluster (p xeles)")
92  plt.ylabel("Densidad de p xeles sin seed")
93  plt.title("Densidad vs distancia (minX     1500)")
94  plt.legend()
95  plt.grid()
96  plt.xlim(0, x_max)
97  plt.ylim(0, 0.1)
98  plt.show()
99
100 # Gr fica para clusters con minX > 1500
101 plt.figure(figsize=(8, 5))
102 plt.scatter(distances_high, densities_high, alpha=0.6, s=10, color="red",
        label="minX > 1500")
103 plt.xlabel("Distancia desde el centro del cluster (p xeles)")
104 plt.ylabel("Densidad de p xeles sin seed")
105 plt.title("Densidad vs distancia (minX > 1500)")
106 plt.legend()
107 plt.grid()
108 plt.xlim(0, x_max)
109 plt.ylim(0, 0.1)
```

```
plt.show()
```

Listing A.3: Código empleado para calcular las densidades de pixeles por anillos en la sección 4.2

A.3.4. Código empleado en la sección 4.3

Se adjunta el código que me ha permitido realiza un cálculo alternativo al anterior para calcular la densidad de píxeles por anillos sin tener en cuenta el área de los anillos.

```
# Importaci n de librer as necesarias
import ROOT
from ROOT import TFile
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import binned_statistic

# Abrir archivo ROOT con datos de clusters reconstruidos
file = TFile("/Users/isma/Desktop/universidad/Cuarto/Tfg/
    hadd_01APR2024_panaSKImg_clustersRec_pp_proc_skp_ext3.root")
clustersRec = file.Get("clustersRec")  # Obtener el  rbol  con los
    clusters
nentries = clustersRec.GetEntries()    # N mero total de eventos en el
     rbol

# Inicializaci n de listas para almacenar resultados por regi n (low =
    minX     1500, high = minX > 1500)
ring_distances_low, ring_densities_low = [], []
ring_distances_high, ring_densities_high = [], []
clusters_with_density1 = []  # Guardar  clusters donde la densidad en un
    anillo es 1

# Mapa global de p xeles para evitar contar el mismo pixel sin seed
    varias veces en diferentes anillos
pixel_distance_map = {}

# Definici n de anillos conc ntricos de radio creciente (10, 20, ...,
    150 p xeles)
r_step = 10
num_rings = 15
ring_radii = [r_step * (i + 1) for i in range(num_rings)]

# Funci n principal que calcula distancias y densidades en anillos para
    clusters con seed y energ a alta
def process_clusters(high_energy_clusters, no_seed_clusters,
    distances_list, densities_list):
    global pixel_distance_map

    for j in high_energy_clusters:
        # Obtener coordenadas del centro del cluster energ tico
        center_x = QmaxX[j]
        center_y = QmaxY[j]
        pixE = set(zip(pixels_x[j], pixels_y[j]))  # Pixeles del cluster
            energ tico

```

```
        # Contar cu ntos de sus p xeles est n en el cuadrante superior derecho respecto al centro
        count_cluster_quad = sum((x > center_x and y > center_y) for x, y in zip(pixels_x[j], pixels_y[j]))

        # Recorremos todos los anillos definidos
        for i in range(len(ring_radii)):
            r_inner = 0 if i == 0 else ring_radii[i - 1]
            r_outer = ring_radii[i]
            count_no_seed = 0
            distances_in_ring = []

            # Buscar clusters sin seed (pixeles "flotantes") que est n en este anillo
            for k in no_seed_clusters:
                if hotregions[k] != 0 or k == j:
                    continue  # Saltar si est  en zona caliente o es el mismo cluster

                for x, y in zip(pixels_x[k], pixels_y[k]):
                    if (x, y) in pixE or x <= center_x or y <= center_y:
                        continue  # Saltar si el pixel ya pertenece al cluster energ tico o no est  en el cuadrante

                    # Calcular distancia al centro del cluster energ tico
                    distance = np.sqrt((x - center_x) ** 2 + (y - center_y) ** 2)
                    if r_inner <= distance < r_outer:
                        # Verificar que no se haya contado ya este p xel para un anillo m s interno
                        if ((x, y) not in pixel_distance_map) or (pixel_distance_map[(x, y)] > r_inner):
                            pixel_distance_map[(x, y)] = r_inner
                            count_no_seed += 1
                            distances_in_ring.append(distance)

            # Calcular densidad como proporci n de pixeles sin seed sobre total relevante
            denom = count_no_seed + count_cluster_quad
            density = count_no_seed / denom if denom > 0 else 0

            # Si la densidad es pr cticamente 1, reportar el cluster
            if np.isclose(density, 1, atol=1e-3):
                print(f"DENSIDAD = 1     cluster_id: {cluster_ids[j]}, posici n: ({center_x:.1f}, {center_y:.1f})")
                clusters_with_density1.append((cluster_ids[j], center_x, center_y))

            # Si hay datos v lidos , guardar para graficar
            if density > 0 and distances_in_ring:
                if density == 1:
                    continue
                avg_distance_ring = np.mean(distances_in_ring)
                distances_list.append(avg_distance_ring)
```

```
                densities_list.append(density)

# Recorremos todas las entradas del  rbol
for entry in range(nentries):
    clustersRec.GetEntry(entry)

    # Extraer datos de cada cluster en el evento actual
    cluster_ids = np.array(clustersRec.cluster_id)
    has_seed = np.array(clustersRec.has_seed)
    pixels_x = [np.array(clustersRec.pixels_x[i]) for i in range(len(
        cluster_ids))]
    pixels_y = [np.array(clustersRec.pixels_y[i]) for i in range(len(
        cluster_ids))]
    minX = np.array(clustersRec.minX)
    maxX = np.array(clustersRec.maxX)
    minY = np.array(clustersRec.minY)
    maxY = np.array(clustersRec.maxY)
    QmaxX = np.array(clustersRec.QmaxX)
    QmaxY = np.array(clustersRec.QmaxY)
    energies = np.array(clustersRec.Energy)
    hotregions = np.array(clustersRec.hotreg)

    # Clasificaci n por regiones y tipo de cluster
    high_energy_low = np.where((energies > 50) & (has_seed == 1) & (minX
        <= 1500) & (hotregions == 0) & (maxY < 5000))[0]
    high_energy_high = np.where((energies > 50) & (has_seed == 1) & (minX
        > 1500) & (hotregions == 0) & (maxY < 5000))[0]

    no_seed_low = np.where((has_seed == 0) & (minX <= 1500) & (maxY <
        5000))[0]
    no_seed_high = np.where((has_seed == 0) & (minX > 1500) & (maxY <
        5000))[0]

    # Procesar ambas regiones por separado
    process_clusters(high_energy_low, no_seed_low, ring_distances_low,
        ring_densities_low)
    process_clusters(high_energy_high, no_seed_high, ring_distances_high,
        ring_densities_high)

# Funci n para graficar resultados con perfil promedio por bin
def plot_with_profile(x_data, y_data, color, title, y_limits=(0, 1)):
    plt.figure(figsize=(12, 8))
    plt.scatter(x_data, y_data, alpha=0.4, s=20, color=color, label="Datos
         individuales")

    # Calcular promedio por bin de 10 p xeles
    bins = np.arange(0, max(x_data) + 10, 10)
    mean_stat, bin_edges, _ = binned_statistic(x_data, y_data, statistic='
        mean', bins=bins)
    bin_centers = (bin_edges[:-1] + bin_edges[1:]) / 2

    # Dibujar perfil promedio como l nea
    plt.plot(bin_centers, mean_stat, color='black', marker='o', linestyle=
        '-', label="Promedio por bin (10px)")
```

```
122
123     plt.xlabel("Distancia promedio del anillo al centro /pixeles")
124     plt.ylabel("Densidad de pixeles por anillo")
125     plt.title(title)
126     plt.ylim(y_limits)
127     plt.grid()
128     plt.legend()
129     plt.show()
130
131 # Mostrar gr fica para la regi n izquierda (minX     1500)
132 if ring_distances_low:
133     plot_with_profile(ring_distances_low, ring_densities_low, color="blue"
            , title="Densidad vs Distancia por anillo (minX     1500)")
134 else:
135     print("No hay datos para minX     1500")
136
137 # Mostrar gr fica para la regi n derecha (minX > 1500)
138 if ring_distances_high:
139     plot_with_profile(ring_distances_high, ring_densities_high, color="red
            ", title="Densidad vs Distancia por anillo (minX > 1500)")
140 else:
141     print("No hay datos para minX > 1500")
```

Listing A.4: Código empleado para calcular las densidades de pixeles por anillos en la sección 4.3

A.3.5. Código empleado en la sección 4.4

Este es el código empleado para relacionar la energía del cluster con su forma a partir de la variable *fpix*.

```
1  # Importar librer as necesarias
2  import ROOT                      # Para trabajar con archivos ROOT (de
       f sica de part culas)
3  from ROOT import TFile           # Clase para abrir archivos .root
4  import numpy as np              # Para manejar arrays num ricos
5  import matplotlib.pyplot as plt # Para graficar
6  import matplotlib.patches as patches  # Para dibujar rect ngulos
7
8  # Abrir el archivo ROOT que contiene los datos de clusters reconstruidos
9  root_file = TFile("/Users/isma/Desktop/universidad/Cuarto/Tfg/
       panaSKImg_clustersRec_composeimage_pp_proc_Run4_ext2_ext0_waders_ext0.
       root")
10
11 # Acceder al  rbol /clase 'clustersRec' dentro del archivo
12 clustersRec = root_file.Get("clustersRec")
13
14 # N mero total de entradas (eventos) en clustersRec
15 nentries = clustersRec.GetEntries()
16
17 # Inicializar listas para guardar datos extra dos
18 energy_values = []
19 fraction_pix_values = []
20
21
```

```
# Recorrer cada evento (entrada) del  rbol
for entry in range(nentries):
    clustersRec.GetEntry(entry)  # Cargar datos de ese evento
    cluster_ids = np.array(clustersRec.cluster_id)  # Obtener IDs de clusters

    # Recorrer cada cluster individual
    for i in range(len(cluster_ids)):
        # Solo considerar clusters que tienen una semilla
        if clustersRec.has_seed[i] == 1:
            # Obtener propiedades del cluster
            energy = float(clustersRec.Energy[i])
            maxX = int(clustersRec.maxX[i])
            minX = int(clustersRec.minX[i])
            maxY = int(clustersRec.maxY[i])
            minY = int(clustersRec.minY[i])
            Npix = int(clustersRec.Npix[i])  # N mero de p xeles

            # Obtener coordenadas y energ as de los p xeles del cluster
            pixels_x = np.array(clustersRec.pixels_x[i])
            pixels_y = np.array(clustersRec.pixels_y[i])
            pixels_E = np.array(clustersRec.pixels_E[i])

            # Calcular la fracci n de p xeles ocupados respecto a la caja que los contiene
            area_around_cluster = (maxX - minX + 1) * (maxY - minY + 1)
            fraction_pix = Npix / area_around_cluster if area_around_cluster > 0 else 0

            # Guardar energ a y fracci n para graficar luego
            energy_values.append(energy)
            fraction_pix_values.append(fraction_pix)

# ----------------------------------------------------
# Graficar Energ a vs Fracci n de p xeles ocupados
plt.figure(figsize=(10, 6))
plt.scatter(energy_values, fraction_pix_values, s=1, color='blue', alpha=0.7)
plt.xlabel("Energ a (keV)")
plt.ylabel("Fracci n de p xeles en la caja")
plt.title("Energ a vs Fracci n de p xeles ocupados en la caja")
plt.grid(True)
plt.tight_layout()
plt.show()
```

Listing A.5: Código empleado para calcular el valor de $fpix$ y relacionarlo con la energía de los clusters en la sección 4.4

A.3.6. Código empleado en la sección 4.5

Finalmente, se muestra el código empleado en la última seccióin de los resultados dedicdada a estudiar la distribución de los píxeles sin seed alrededor del cluster que fue ajustado a una recta.

```
# Importar librer as necesarias
```

```
import ROOT                                  # Para manejar archivos ROOT (formato t pico en f sica de part culas)
import numpy as np                           # Para operaciones num ricas y arrays
import matplotlib.pyplot as plt              # Para visualizaci n gr fica

# ------------------------------------------------------------
# Funci n que calcula:
# - la distancia firmada de un punto (px, py) a un segmento [x1,y1]-[x2,y2]
# - y el punto de proyecci n m s cercano sobre el segmento
def signed_distance_and_projection(px, py, x1, y1, x2, y2):
    dx = x2 - x1
    dy = y2 - y1
    len_sq = dx * dx + dy * dy

    if len_sq == 0:
        return 0, x1, y1  # Si el segmento es un punto

    # Proyecci n escalar del punto sobre la recta del segmento
    t = ((px - x1) * dx + (py - y1) * dy) / len_sq

    # Forzar la proyecci n a estar dentro del segmento
    if t < 0.0:
        closest_x, closest_y = x1, y1
    elif t > 1.0:
        closest_x, closest_y = x2, y2
    else:
        closest_x = x1 + t * dx
        closest_y = y1 + t * dy

    # Calcular distancia eucl dea y firmarla seg n la orientaci n
    distance = np.hypot(px - closest_x, py - closest_y)
    sign = np.sign(dy * (px - x1) - dx * (py - y1))  # Determina el lado del punto respecto al segmento

    return sign * distance, closest_x, closest_y

# ------------------------------------------------------------
# Abrir el archivo ROOT y acceder al  rbol  'clustersRec'
root_file = ROOT.TFile("/Users/isma/Desktop/universidad/Cuarto/Tfg/panaSKImg_clustersRec_pp_proc_skp_modS89_T140_Vv3_bkgd_CLEAR600_NSAMP169_NROW515_NC.root")
clustersRec = root_file.Get("clustersRec")
nentries = clustersRec.GetEntries()

found_cluster = False  # Flag para saber si se encuentra un cluster adecuado

# ------------------------------------------------------------
# Recorrer todas las entradas del archivo ROOT
for entry in range(nentries):
    clustersRec.GetEntry(entry)

```

```
    # Extraer datos de los clusters
    cluster_ids = np.array(clustersRec.cluster_id)
    energies = np.array(clustersRec.Energy)
    minX = np.array(clustersRec.minX)
    maxX = np.array(clustersRec.maxX)
    minY = np.array(clustersRec.minY)
    maxY = np.array(clustersRec.maxY)

    # Obtener las coordenadas de los p xeles por cluster
    pixels_x = [list(clustersRec.pixels_x[i]) for i in range(len(
        cluster_ids))]
    pixels_y = [list(clustersRec.pixels_y[i]) for i in range(len(
        cluster_ids))]
    has_seed = np.array(clustersRec.has_seed)  # Si el cluster tiene
        semilla
    hotreg = np.array(clustersRec.hotreg)      # Si est  en regi n
        caliente

    # Filtrar clusters que:
    # - tengan energ a > 500 keV
    # - tengan semilla
    # - no est n en regi n caliente
    high_energy_clusters = np.where((energies > 500) & (has_seed == 1) & (
        hotreg == 0))[0]

    # Si no se encuentran clusters v lidos , pasar al siguiente evento
    if len(high_energy_clusters) == 0:
        continue

    # Seleccionar el primer cluster v lido
    i = high_energy_clusters[0]
    cluster_pixels_x = pixels_x[i]
    cluster_pixels_y = pixels_y[i]

    if len(cluster_pixels_x) < 2:
        continue  # Necesitamos al menos dos puntos para ajustar una recta

    # ------------------------------------------------------------
    # Ajuste lineal del cluster energ tico
    A, B = np.polyfit(cluster_pixels_x, cluster_pixels_y, 1)  # y = Ax + B
    x1 = min(cluster_pixels_x)
    x2 = max(cluster_pixels_x)
    y1 = A * x1 + B
    y2 = A * x2 + B

    # Definir una "caja de an lisis" alrededor del cluster
    left_limit = minX[i] - 50
    right_limit = maxX[i] + 50
    bottom_limit = minY[i] - 20
    top_limit = maxY[i] + 20

    # Inicializar listas para p xeles cercanos y sus proyecciones
    surrounding_pixels_x = []
    surrounding_pixels_y = []
```

```
98      projection_points_x = []
99      projection_points_y = []
100
101     # Recorrer otros clusters para buscar p xeles dentro de la caja
102     for k in range(len(cluster_ids)):
103         if k == i or has_seed[k] != 0 or hotreg[k] != 0:
104             continue  # Saltar el cluster energ tico y otros con semilla
                o en zona caliente
105
106         for x, y in zip(pixels_x[k], pixels_y[k]):
107             if left_limit <= x <= right_limit and bottom_limit <= y <=
                top_limit:
108                 # Calcular distancia firmada y punto proyectado sobre la
                    recta del cluster energ tico
109                 dist, proj_x, proj_y = signed_distance_and_projection(x, y
                    , x1, y1, x2, y2)
110                 surrounding_pixels_x.append(x)
111                 surrounding_pixels_y.append(y)
112                 projection_points_x.append(proj_x)
113                 projection_points_y.append(proj_y)
114
115     # ------------------------------------------------------------
116     # Visualizaci n del cluster y sus alrededores
117     plt.figure(figsize=(10, 10))
118
119     # Cluster energ tico (rojo)
120     plt.scatter(cluster_pixels_x, cluster_pixels_y, color='red', s=10,
            label="Cluster energ tico (E > 500 keV)")
121
122     # P xeles sin semilla cercanos (azul)
123     plt.scatter(surrounding_pixels_x, surrounding_pixels_y, color='blue',
            s=10, label="P xeles sin seed")
124
125     # L nea de ajuste del cluster
126     plt.plot([x1, x2], [y1, y2], 'k--', label="Recta de ajuste")
127
128     # Dibujar l neas desde cada p xel al punto de proyecci n sobre la
            recta
129     for (x, y, px, py) in zip(surrounding_pixels_x, surrounding_pixels_y,
            projection_points_x, projection_points_y):
130         plt.plot([x, px], [y, py], color='gray', linewidth=0.8, alpha=0.6)
131
132     plt.xlabel("X (px)")
133     plt.ylabel("Y (px)")
134     plt.title("Cluster energ tico con recta de ajuste y distancias a
            p xeles sin seed")
135     plt.legend()
136     plt.grid(True)
137     plt.show()
138
139     found_cluster = True  # Ya se encontr  y grafic  un cluster, salimos
            del bucle
140     break
141
```

```
142 # ------------------------------------------------------------
143 # Mensaje si no se encontr  ning n cluster v lido
144 if not found_cluster:
145     print("No se encontr  ning n cluster con energ a > 500 keV.")
```

Listing A.6: Código empleado para representar la distribución de distancias a la recta ajustada en la sección 4.5