Universidad Autónoma de Yucatán

Facultad de Ingeniería

Selección de materiales óptimos para una estructura fotovoltaica basada en pirita mediante machine learning

Tesis

Presentado por:

Pedro Daniel Zapata Herrera

Para obtener el título de:

Ingeniero Físico

Asesor:

Dra. Inés Riech Méndez

Coasesor:

Dr. Enrique Camacho Pérez

Mérida, Yucatán, México
2024

"Aunque este trabajo hubiere servido para el Examen de Grado y hubiere sido aprobado por el Sínodo, solo el autor es responsable de las doctrinas emitidas en él"

Agradecimientos

Agradezco al *Dr. Camacho* y a la *Dra. Riech* por su guía, conocimientos y apoyo a lo largo de esta investigación, que han sido elementos clave para su realización. Su disposición para compartir su experiencia y su orientación en cada paso del camino me han ayudado a superar cada desafío que se presentó.

A mi *familia*, por su constante apoyo y aliento, que han sido esenciales para llegar hasta aquí, siempre motivándome a dar lo mejor de mí en todo lo que hago. Sin su acompañamiento y respaldo, nada de esto habría sido posible.

A mi *pareja*, quien ha estado presente de manera incondicional, brindándome siempre su comprensión y apoyo en todo momento.

A mis *amigos* y a todas las personas con quienes he compartido el mismo camino en algún momento de mi vida, gracias por ser parte de este recorrido.

Y finalmente, *a mi yo del futuro*, para que siempre recuerde dónde empezó y que **todo lo que se proponga lo puede lograr**.

- Pedro

Resumen

En este proyecto de tesis se utiliza el algoritmo K-Means para agrupar materiales basándose en tres propiedades: band gap, estructura cristalina y coeficiente de absorción óptica, logrando una selección preliminar de materiales para junturas PN en celdas solares de Pirita. A pesar de la restricción en el número de variables, los patrones obtenidos en las agrupaciones son nítidos y coherentes, lo que respalda la capacidad de K-Means para seleccionar materiales adecuados en celdas solares con capa absorbente de Pirita. La metodología se basa en la construcción de una base de datos, la optimización de los parámetros de K-Means y el uso de técnicas de validación como el método de la silueta y el codo, con el propósito de seleccionar materiales adecuados para la capa ventana en una celda solar.

El análisis ofrece resultados prometedores al identificar materiales con propiedades ópticas y estructurales compatibles con la Pirita, lo que facilita una integración eficiente en celdas solares. Estos materiales se caracterizan por un alto band gap y un bajo coeficiente de absorción óptica, optimizando la transmisión de luz hacia la capa absorbente. Su estructura cúbica permite una integración sin defectos, mientras que su composición en compuestos binarios simplifica la síntesis y reduce los costos. Además, la selección excluye elementos tóxicos o costosos, alineándose con un enfoque práctico y sostenible para aplicaciones fotovoltaicas.

Finalmente, esta investigación abre camino a estudios futuros, como la validación experimental de los resultados y la exploración de materiales adicionales que mejoren la eficiencia de las celdas solares. La metodología demuestra que el uso de machine learning facilita la identificación de materiales con propiedades específicas, impulsando avances no solo en celdas solares, sino también en otros dispositivos electrónicos. Este enfoque proporciona una herramienta replicable que permite optimizar constantemente la selección de materiales, integrando la ciencia de datos con la ciencia de materiales.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Perspectiva general

En los últimos años ha habido un aumento desorbitado en la temperatura de la atmósfera y de los océanos a nivel mundial. Este abrupto cambio climático es consecuencia del efecto invernadero generado por las actividades humanas que emiten grandes cantidades de gases a la atmósfera. Sin embargo, lo que todavía genera controversia es la fuente y razón de este aumento de la temperatura. Aun así, la mayor parte de la comunidad científica asegura que hay más de un 90 % de certeza que el aumento se debe a las concentraciones de gases de efecto invernadero por las actividades humanas, que incluyen deforestación y la quema de combustibles fósiles como el petróleo y el carbón (ONG, 2020).

Por estas alarmantes razones, el desarrollo e investigación de nuevas fuentes de energía es de vital importancia para prolongar la vida en nuestro planeta, ya que los combustibles fósiles generan una contaminación irreversible. Una de las alternativas más investigadas y utilizadas son las energías renovables ya que estas obtienen energía limpia a partir de fuentes naturales inagotables y no producen gases de efecto invernadero.

La energía solar es una clasificación de las energías renovables y se viene desarrollando desde la década de 1860. Este tipo de energía se obtiene a partir del aprovechamiento de la radiación solar. Hoy en día, la forma más común de aprovechar la radiación solar es mediante las celdas fotovoltaicas o celdas solares, las cuales transforman la radiación electromagnética del sol en energía eléctrica (Arencibia-Carballo, 2016).

Según Rodríguez (2022), la historia detrás de las celdas solares se remonta en 1839 cuando se reconoce por primera vez el efecto fotovoltaico, por parte del físico francés Alexadre-Edmond Becquerel y a partir de ese año, científicos empezaron a tratar de entender el efecto

fotovoltaico y realizar experimentos, pero no fue hasta más de un siglo cuando en 1954 se presentó la primera celda solar de silicio con un 4 % de eficiencia, dicha celda se desarrolló por los científicos Daryl Chapin, Calvin Souther Fullery y Gerald Pearson en los laboratorios Bell.

Desde la creación de la primera celda solar, el objetivo principal para los científicos e investigadores es aumentar la eficiencia de la celda y reducir el costo de producción, pues idealmente se desea una celda solar con alta eficiencia a bajo costo. Actualmente, existe una competencia a nivel mundial para obtener la celda solar con mayor eficiencia y que sea económicamente redituable.

Con el fin de crear celdas solares cada vez mejores, se han probado con diversos materiales, se han creado nuevos métodos y técnicas de crecimiento y han surgido nuevas tecnologías que han favorecido al aumento de la eficiencia. La Figura 1.1 muestra un resumen histórico de las mejores eficiencias de celdas solares de diversas tecnologías fotovoltaicas. También se puede apreciar que las eficiencias más altas se han conseguido con celdas solares de tipo multiunión (representadas de color morado).

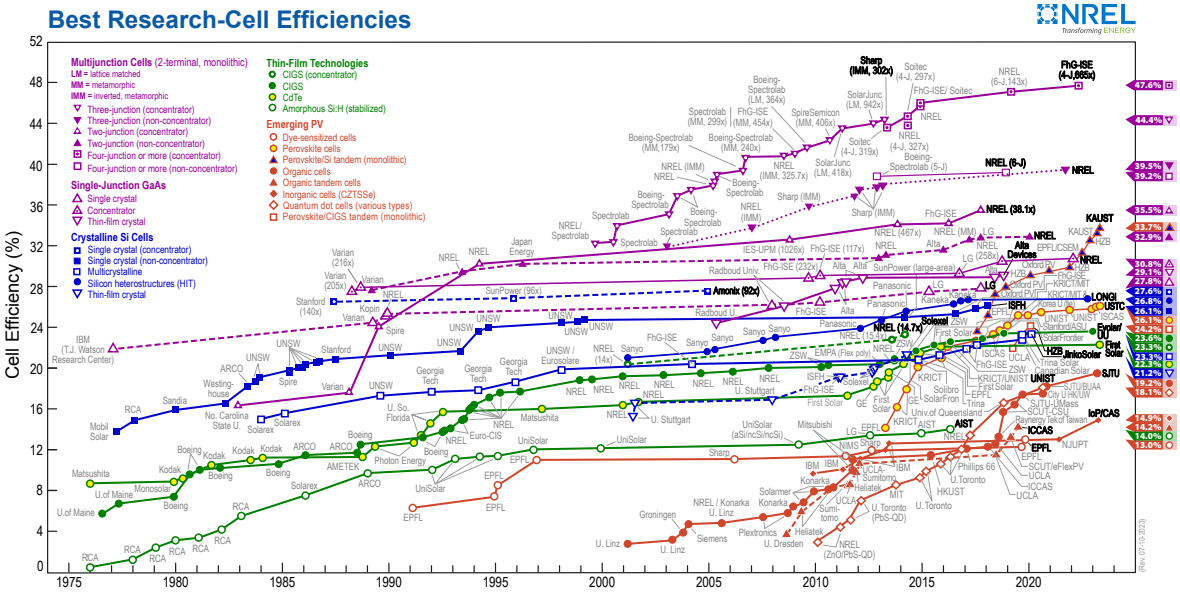


Figura 1.1: Mejores eficiencias de celdas solares obtenidas en laboratorios, (NREL, 2023).

La Figura 1.1 reporta una eficiencia máxima de 47.6 % perteneciente a una celda tipo multiunión. Sin embargo, ese valor se obtuvo bajo condiciones experimentales y en el ámbito comercial no se puede obtener ese valor máximo, pues resulta extremadamente caro fabricar celdas solares en masa con los materiales y procesos élite con los que

se realizan las celdas solares de investigación.

La eficiencia media de los paneles solares comerciales ha pasado del 15 % a valores por encima del 23 % en los últimos cinco años y se espera en un futuro no lejano que el desarrollo de las celdas solares de estructura tándem de silicio y perovskita proporcionen eficiencias cercanas al 30 % para celdas comerciales.

Desgraciadamente, el desarrollo de nuevas y mejores celdas solares no es tarea fácil y para comprender las limitaciones, primero es necesario conocer qué es una juntura PN; esta es una juntura de dos materiales [uno cargado con exceso de electrones (material N) y otro con déficit de electrones (material P)] y es un factor clave en el diseño de la celda gracias a que la compatibilidad de características de los materiales N y P proporciona una alta eficiencia en la celda (Kittel, 2012).

1.2. Problemática

1.2.1. Planteamiento del problema

El proceso de investigación para hallar la mejor combinación de materiales N y P es un aspecto que resulta tardado y complicado en el desarrollo de celdas, ya que se necesita un profundo conocimiento de las propiedades físicas, químicas y estructurales de los materiales a utilizar. Además, los resultados teóricos y las simulaciones de las junturas PN suelen diferir significativamente de los resultados experimentales, lo que hace que la mayoría de las celdas fotovoltaicas sean desarrolladas mediante ensayo y error.

Otro desafío crítico en este proceso es la dificultad de encontrar capas ventana y materiales absorbentes que sean compatibles entre sí. Esto es crucial, ya que la capa ventana debe permitir la entrada de luz mientras protege la capa absorbente, que es la responsable de captar los fotones y convertirlos en energía eléctrica. La incompatibilidad entre estos materiales puede llevar a problemas como la degradación de la eficiencia, tensiones mecánicas en la estructura y una vida útil reducida de la celda. Por lo tanto, se requiere una cuidadosa selección y optimización de los materiales para lograr un balance adecuado entre eficiencia y durabilidad.

1.2.2. Formulación del problema

De los argumentos expuestos anteriormente se nota que existe una problemática evidente: el proceso para encontrar los compuestos idóneos tanto para las junturas PN como para las capas ventana y absorbentes es tardado e ineficiente, ya que se desperdician grandes cantidades de dinero, tiempo y materiales en el testeo de combinaciones óptimas. La dificultad radica en que, además de buscar una combinación eficiente entre materiales N y P, es esencial asegurar que las capas ventana y absorbentes sean compatibles y efectivas en su función. Esta complejidad no solo incrementa los costos y el tiempo de desarrollo, sino que también limita la capacidad de innovar en la creación de celdas fotovoltaicas más eficientes y duraderas.

Es por esto que científicos, físicos e ingenieros esperan acortar el camino que los conduzca a correlacionar materiales N y P para formar junturas PN de alta eficiencia, recurriendo al empleo de algoritmos de inteligencia artificial y acceso a bancos de datos sobre las propiedades y estructuras de materiales (Pivetta, 2020).

Para abordar esta problemática, se han desarrollado diversas soluciones, pero la idea principal radica en crear y perfeccionar algoritmos capaces de prever con exactitud la correlación de materiales adecuados para crear celdas solares sin tener que recurrir al testeo de innumerables combinaciones.

En este proyecto de tesis, se propone utilizar la Pirita (FeS_2) como material tipo P (material absorbente) y se desea hallar el material tipo N (material ventana) adecuado para lograr una juntura PN eficiente. Para lograrlo, se debe recurrir a otras herramientas, ya que a la ciencia de materiales no le concierne la creación de algoritmos capaces de predecir; por eso se acude a la ciencia de datos, inteligencia artificial y la minería de datos pues han demostrado ser capaces de crear algoritmos capaces de hallar, analizar y predecir datos.

La unión de la ciencia de materiales y la inteligencia artificial (IA) es clave para erradicar la problemática planteada, pues la creación de algoritmos predictivos le corresponde al machine learning (disciplina del campo de la IA) y los parámetros de retroalimentación del algoritmo le corresponde a la ciencia de materiales. En conjunto, ambas ciencias han demostrado resultados fructíferos y replicables, por lo que se espera hallar el material adecuado para la Pirita mediante el uso de machine learning (ML).

1.3. Hipótesis

El uso de algoritmos de machine learning para analizar bases de datos de propiedades de materiales permitirá identificar materiales óptimos de tipo N que funcionen eficazmente como capas ventana en una juntura PN con la Pirita.

1.4. Objetivos

Objetivo General: Predecir materiales óptimos de tipo N que puedan funcionar como capas ventana para formar una juntura PN con la Pirita, utilizando análisis de bases de datos mediante técnicas de machine learning.

Objetivos Específicos:

- Búsqueda de base de datos de materiales.
- Manipulación y procesamiento de los datos.
- Evaluar el algoritmo de machine learning que mejor se adecue al problema.
- Optimización de hiperparámetros del algoritmo seleccionado.

1.5. Justificación

La implementación del machine learning en el proceso de investigación y desarrollo de dispositivos fotovoltaicos puede favorecer a la identificación de patrones y compatibilidad entre materiales semiconductores, lo cual resulta complicado con ciertos métodos convencionales de estudio y crecimiento de celdas. La relación entre materiales semiconductores también favorece a otros campos como la electrónica, ya que los mejores modelos de junturas PN propuestos por ML pueden tener un impacto significativo en el rendimiento y la eficiencia no solo de celdas solares, sino también de diodos o transistores.

En el presente trabajo de investigación se particulariza a la juntura con Pirita por motivo de sencillez, ya que el proceso de analizar junturas mediante ML sin proponer un material base tipo N o P puede resultar tardado y complicado para una tesis de nivel licenciatura. Sin embargo, esto no afecta la replicación del proceso para otros materiales diferentes a la Pirita, debido a que el análisis, el algoritmo y la base de datos serán similares. Por lo tanto, los modelos de machine learning

pueden ser entrenados para adaptarse a cambios en las propiedades de los materiales; lo cual permite una mejora continua en la precisión de las predicciones a medida que se obtiene más información.

Tradicionalmente, la búsqueda y prueba de nuevas junturas PN puede ser un proceso costoso y tardado. El uso de ML para predecir materiales podría acelerar este proceso al identificar candidatos prometedores antes de realizar experimentos en el laboratorio. Además, se minimiza el riesgo de desperdicio de material al garantizar una juntura eficiente y se consume menos energía durante el proceso de testeo. Todas estas ventajas favorecen económicamente al proceso de investigación, lo cual resulta útil si se tiene un presupuesto o equipo limitado. De esta manera, destaca la utilización de ML para la reducción de recursos durante el testeo de junturas PN.

Finalmente, el tiempo de investigación se ve recortado al identificar los mejores materiales sin recurrir al método convencional de ensayo y error. Esto es clave para alcanzar nuevas celdas solares con mejores eficiencias en el menor tiempo posible, garantizando una evolución en los dispositivos fotovoltaicos a mediano plazo.

Capítulo 2

Antecedentes

2.1. ML y la ciencia de materiales

El ML ha experimentado un enorme crecimiento en la última década y, más recientemente, se ha abierto camino en ámbitos científicos como la salud, la física, la astronomía y la ciencia de los materiales. Esta revolucionaria innovación de utilizar ML surge gracias a la presencia de las herramientas computacionales e IA en nuestra vida cotidiana, las cuales se fueron haciendo notorias a mediados del siglo XXI y hoy en día se utilizan en formas que afectan a nuestra vida cotidiana, a menudo de manera que ni siquiera nos damos cuenta.

Aterrizando en el sector de la ciencia de los materiales, la implantación del ML está en sus primeras fases y ya demuestra tener un gran potencial. Los científicos de materiales aplican técnicas de ML para extraer información de datos experimentales y computacionales previamente recopilados para la caracterización de materiales, la predicción de propiedades moleculares, la aceleración de simulaciones, el descubrimiento de materiales y el modelado generativo (Karande, Gallagher, y Han, 2022).

A este paso, no transcurrirá mucho tiempo antes de que el ML ayude a los científicos a producir nuevos materiales que cumplan con propiedades específicas de forma más rápida y eficiente de lo que permiten actualmente las herramientas o procesos tradicionales (Son, 2019).

2.2. ML aplicado en celdas solares

Diversos autores que han utilizado ML enfocado en optimización de celdas solares recurrieron a k Nearest Neighbors (kNN), la cual es una herramienta basada en la idea de que la actividad de un compuesto dado puede predecirse promediando las de sus k vecinos más cercanos, es decir, los k compuestos más similares a él. Además de esta herramienta, también se ha reportado la utilización del algoritmo de

Boruta, Discriminant classifiers (DC), Support vector machine (SVM) y Decision tree (DT) para optimizar parámetros de crecimiento de las celdas solares (Dhimish, 2021).

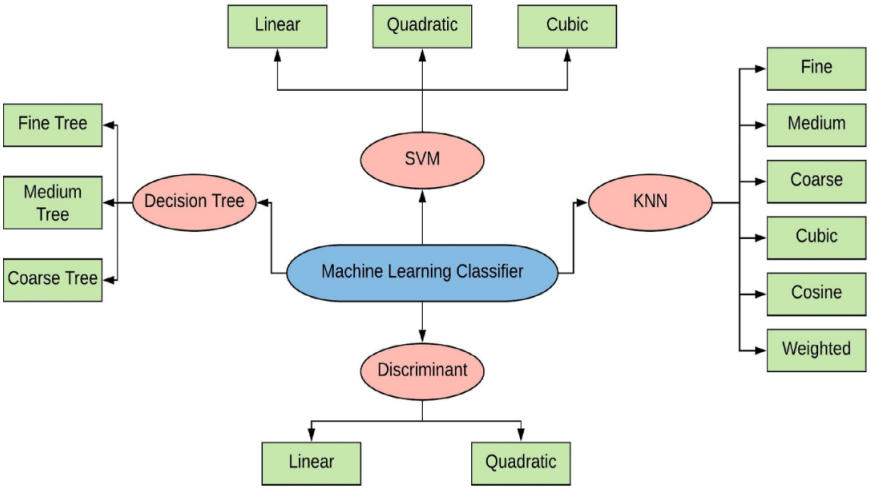


Figura 2.1: Herramientas de ML utilizadas en ciencia de materiales, (Dhimish, 2021).

El método de Boruta es utilizado por Ulaczyk, Morawiec, Zabierowski, Drobiazg, y Barreau (2017) para optimizar el crecimiento de celdas CIGSe, pues durante su crecimiento influyen diferentes parámetros como los tiempos de crecimiento, temperaturas de la fuente y del sustrato, el valor de la presión en la cámara de crecimiento, etc. El algoritmo de Boruta utiliza árboles de decisión aleatorios para resolver problemas de clasificación (discretos) como de regresión (continuos). Los valores de los parámetros de crecimiento de la celda fueron optimizados con este algoritmo para obtener crecimientos que no se pueden alcanzar con métodos tradicionales como la observación o simulaciones. Una vez optimizado el proceso de crecimiento, se esperan mejoras en las propiedades fotovoltaicas de la celda de CIGSe.

Por otro lado, Dhimish (2021) presenta el desarrollo de una herramienta basada en el aprendizaje automático para diagnosticar los "hot-spots" en su fase inicial. Los hot-spots representan un problema importante en las celdas fotovoltaicas, pues generan sobrecalentamiento en la celda y esto afecta su eficiencia. La detección temprana de los hot-spot mejoraría el tiempo de vida de la celda y su rendimiento energético. Dicha problemática fue abordada por Dhimish mediante la implementación de ML, específicamente con las herramientas de DT, DC y SVM para datos de celdas solares en uso.

Adicionalmente, Odabaşı y Yıldırım (2020) empleó ML para deter-

minar la estabilidad de celdas de perovskita. En este trabajo se utilizó DT y kNN para analizar los datos de celdas de perovskita en operación para determinar los factores que aumentan el deterioro de la celda. Con los resultados, se puede determinar las condiciones ideales para la operación de las celdas de perovskita, pues no en todos los ambientes o condiciones las celdas solares aportan su máxima eficiencia.

2.3. ML aplicado en la predicción de celdas solares

Para el caso de predicción de junturas PN en celdas solares, Ulaczyk y cols. (2017) afirma que existen muy pocas publicaciones que aborden un análisis de los datos distinto de kNN para la predicción de celdas fotovoltaicas; la primera aplicación prometedora fue presentada por Yosipof y cols. (2015) quienes utilizaron kNN para hallar los conjuntos de óxidos de metales compatibles para formar celdas solares.

Según el estudio realizado por Yosipof y cols. (2015), dada una base de datos de materiales bien caracterizada, se podrían utilizar herramientas de minería de datos y ML para obtener una buena perspectiva de los resultados prácticos de celdas solares. El diagrama de trabajo seguido por este autor y compañía se aprecia en la Figura 2.2 y a continuación, se enlistan los pasos desarrollados: caracterización de las bases de datos, visualización de datos, desarrollo de modelos, validación de modelos y diseño experimental.

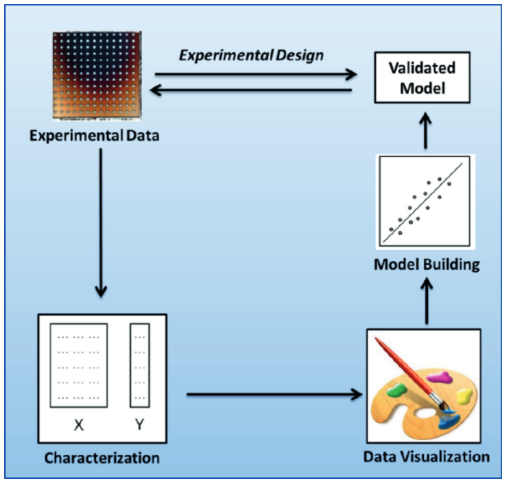


Figura 2.2: Diagrama de trabajo de (Yosipof y cols., 2015).

Para la caracterización de datos, el estudio analizó dos bibliotecas o base de datos de celdas solares compuestas por óxidos de titanio y cobre. Cada material en la base de datos se caracterizó mediante

siete descriptores medidos experimentalmente (variables independientes) y tres propiedades fotovoltaicas (variables dependientes). Posteriormente, para la visualización de datos se utilizó la técnica Principal Component Analysis (PCA), la cual ayuda en la descripción de conjuntos de datos en términos de nuevas variables (componentes) no correlacionadas. El PCA reduce la dimensionalidad de un conjunto de datos, conservando en la medida de lo posible su varianza original. Esta reducción se consigue transformando las variables originales en un nuevo conjunto de variables ortogonales denominadas variables principales (Hsieh y Tung, 2009).

Para el desarrollo del modelo predictivo, se emplearon dos herramientas de ML, k Nearest Neighbors (kNN) y programación genética (GP), respectivamente. La intención de aplicar kNN a las bases de datos de materiales es hallar grupos de compuestos similares que proporcionen celdas solares con las propiedades parecidas.

La herramienta kNN se empleó con el modelo Metropolis Monte Carlos/Simulated Annealing (MC/SA) como el motor de optimización. El objetivo de este algoritmo consiste en optimizar el proceso de dejar fuera un valor de validación cruzada Q^2_{LOO}" [Leave One Out (LOO) cross-validated value] en el espacio de k y sus descriptores. El diagrama del procedimiento kNN empleado en dicho estudio se muestra a continuación.

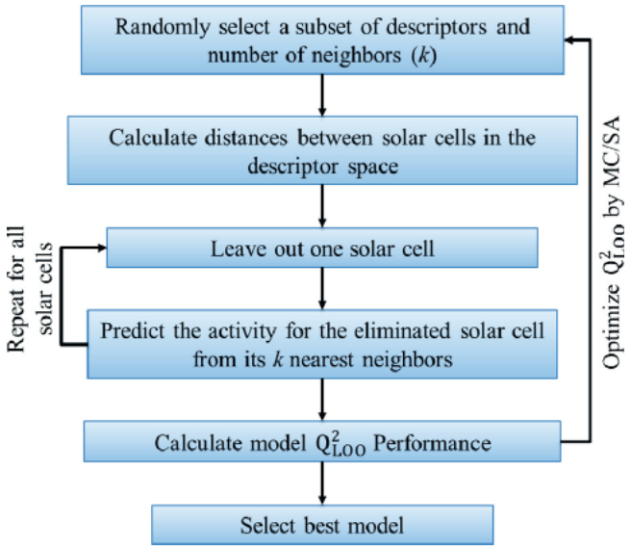
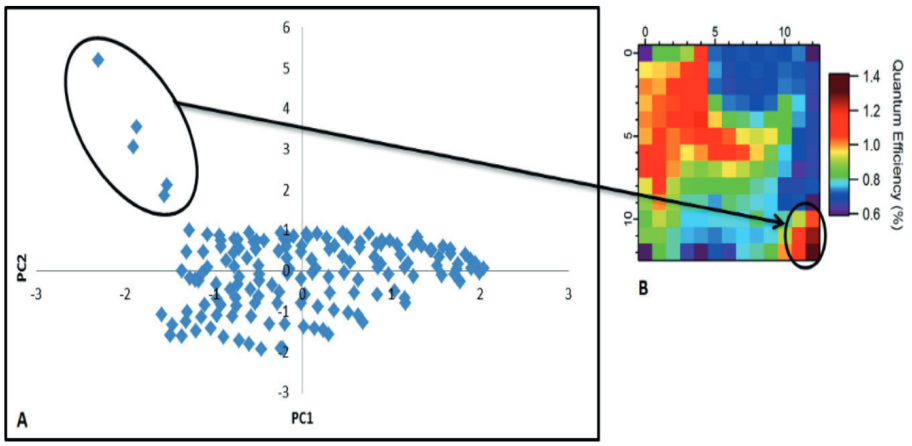


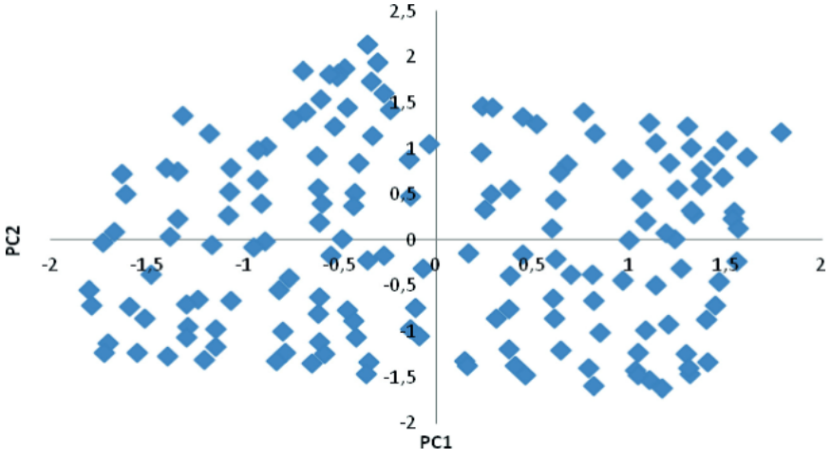
Figura 2.3: Representación esquemática del algoritmo de optimización kNN, (Yosipof y cols., 2015).

Posteriormente, los grupos de modelos seleccionados por kNN se introducen a la programación genética (GP), la cual produce iterati-

vamente una población de nuevos modelos cada vez mejores que los anteriores. Con esto Yosipof y cols. (2015) concluyen el modelo predictivo para la selección de materiales de óxido de titanio y cobre para celdas solares de MO. Los resultados de kNN se visualizaron mediante PCA, en la Figura 2.4a se aprecia una representación 2D de los dos primeros PC en términos de descriptores originales. Estos PC cubren el 70.1 % y el 16.2 % de la varianza original, respectivamente, para un total de 86.3 %. El PC1 son los descriptores de óxido de cobre y PC2 los descriptores de dióxido de titanio) y se encierran los modelos atípicos que no cumplen con las propiedades deseadas, en este caso los modelos atípicos sobrepasaban el porcentaje de eficiencia cuántica promedio.



(a) [A] Diagrama de dispersión de la biblioteca $TiO_2|Cu-O$ en el espacio definido por las dos primeras PC. [B] Gráfico de la eficiencia cuántica interna de las dos primeras PC, el área marcada con un círculo indica la ubicación de los valores atípicos.



(b) Diagrama de dispersión de la biblioteca $TiO_2|Cu-O$ libre de valores atípicos

Figura 2.4: Visualización mediante PCA de los mejores modelos de celdas, (Yosipof y cols., 2015).

En la Figura 2.4b se tiene nuevamente una visión por PCA, esta

vez los PC cubren el 60.7 % y el 36.7 % de la varianza original, respectivamente, para un total del 97.4 %. Las celdas solares se distribuyen uniformemente en el espacio PC sin valores atípicos evidentes, por lo que Yosipof y colaboradores concluyen que el modelado, filtrado y revisión de los grupos de celdas resultó exitoso. Además, los autores afirman que las características y propiedades de los modelos obtenidos concuerdan con las observaciones experimentales, y concluyen la posibilidad de derivar modelos con buenas estadísticas de predicción mediantes kNN y GP.

Capítulo 3

Marco teórico

3.1. Bases teóricas de sistemas fotovoltaicos

3.1.1. Juntura PN

La **unión PN** o **juntura PN** es una estructura en un semiconductor compuesta por dos regiones adyacentes, una de tipo N y otra de tipo P, y debido a que la región de tipo N tiene una concentración de electrones alta y la de tipo P tiene una concentración alta de huecos (déficit de electrones), los electrones se transportarán desde el lado de tipo N hasta el lado de tipo P debido a la fuerza de atracción entre cargas opuestas (ver Figura 3.1) (Ali, 1993).

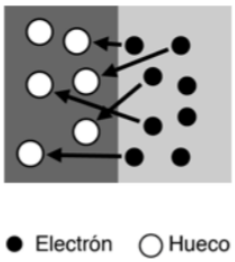


Figura 3.1: Representación de una juntura PN, (GSU, 2018).

Cuando un electrón se desplaza hacia el material tipo P, llena un hueco fijo en la red cristalina y se convierte en un **ión negativo**, mientras que en el material tipo N deja atrás un **ión positivo**. Este proceso genera una acumulación de iones negativos en el material P y de iones positivos en el material N, lo que da lugar a la formación de un campo eléctrico E (GSU, 2018). Esta interacción crea la **región de empobrecimiento**, una zona clave en las celdas solares donde se separan y dirigen los portadores de carga generados por la absorción de luz hacia los contactos eléctricos, lo que permite la generación de corriente eléctrica.

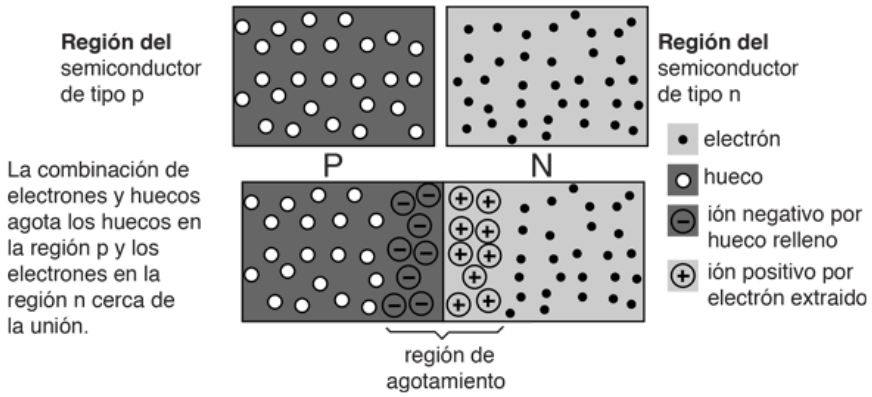


Figura 3.2: Representación de la región de agotamiento, (GSU, 2018).

3.1.2. Celda solar

Según Kittel (2012), una **celda solar** es un dispositivo diseñado para convertir la luz solar en electricidad mediante el uso de materiales semiconductores organizados en una **juntura PN**, que facilita la conversión de energía luminosa en **energía eléctrica**.

Cuando la luz incide sobre la celda solar se produce tanto un voltaje como una corriente que genera energía eléctrica y este proceso comienza con la absorción de luz por el material semiconductor donde los fotones excitan electrones llevándolos a un estado de energía más alto; a continuación, estos electrones de alta energía son transportados desde la celda hacia un circuito externo (ver Figura 3.3) donde liberan su energía al alimentar dispositivos eléctricos y, finalmente, los electrones regresan a la celda solar permitiendo que el ciclo se repita continuamente (PVEducation, 2019).

De acuerdo con Mertens (2018), aunque hay numerosos materiales y procesos que podrían satisfacer las necesidades de la conversión de energía fotovoltaica, en la práctica, esta conversión se realiza casi exclusivamente mediante materiales semiconductores en forma de junturas PN.

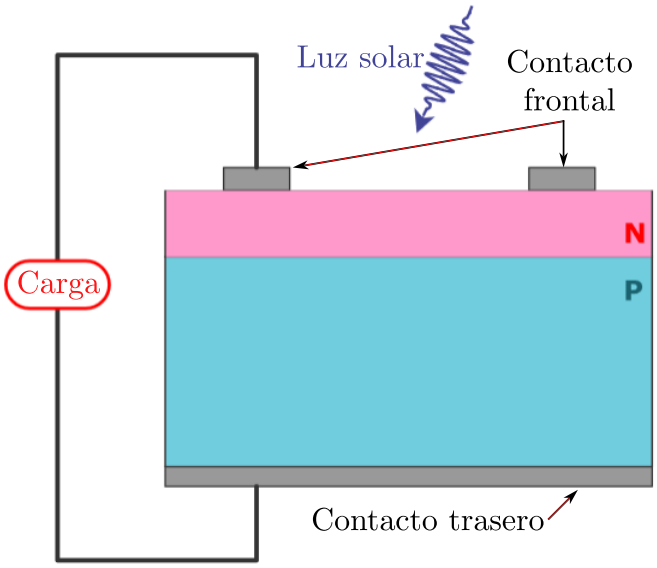


Figura 3.3: Juntura PN y circuito externo de la celda solar, (PVEducation, 2019).

3.1.3. Materiales ventana y absorbentes

En sistemas de celdas solares, los materiales ventana y absorbentes juegan roles fundamentales en la conversión de la luz solar en electricidad. A continuación se proporcionan las definiciones según Mertens (2018).

- **Material ventana:** Es un material transparente que se coloca en la capa superior de la celda solar para permitir que los fotones lleguen al material absorbente. Para cumplir su función, debe presentar una alta transmitancia, un band gap adecuado, un bajo coeficiente de absorción óptica y buena conductividad eléctrica para facilitar el flujo de carga. Además, el tipo de conductividad, ya sea N o P, es esencial, ya que influye en la formación de la juntura con el material absorbente, lo que optimiza la eficiencia de la celda.
- **Material absorbente:** Este es el material principal en la celda solar encargado de absorber los fotones y convertir la energía de la luz en energía eléctrica. Cuando la luz es absorbida, los electrones en el material absorbente son excitados, generando pares electrón-hueco (portadores de carga) que se movilizan para generar una corriente eléctrica. Los materiales absorbentes más comunes incluyen el silicio, perovskitas, arseniuro de galio (GaAs), o diseleniuro de cobre e indio (CIS) en tecnologías de capa delgada.

Ambos materiales son esenciales para la eficiencia de la celda solar: el material ventana optimiza la entrada de luz, mientras que el material absorbente maximiza la generación de carga eléctrica. Para lograr un buen funcionamiento de la celda, es fundamental que ambos materiales sean compatibles. Esta compatibilidad depende de parámetros como el **band gap**, que debe ser adecuado para permitir la absorción selectiva de la luz; el **coeficiente de absorción óptica**, que determina la eficiencia con la que el material puede absorber fotones; y la **estructura cristalina**, que influye en la estabilidad y en la transferencia eficiente de carga.

Debido a que estos tres parámetros serán fundamentales para el agrupamiento de materiales a lo largo de este proyecto, es esencial que cada uno de ellos quede claramente definido:

- **Band gap**: El band gap o banda prohibida es la diferencia de energía entre la banda de valencia, donde se encuentran los electrones ligados, y la banda de conducción, donde los electrones pueden moverse libremente (ver Figura 3.4). Este parámetro determina las propiedades eléctricas y ópticas de un material. Por ejemplo, los *semiconductores*, como el silicio, tienen un band gap moderado, lo que los hace ideales para aplicaciones electrónicas y fotovoltaicas. En contraste, los materiales con un band gap muy grande son *aislantes*, mientras que los *metales*, que carecen de esta brecha, conducen electricidad con facilidad (PVEducation, 2019).

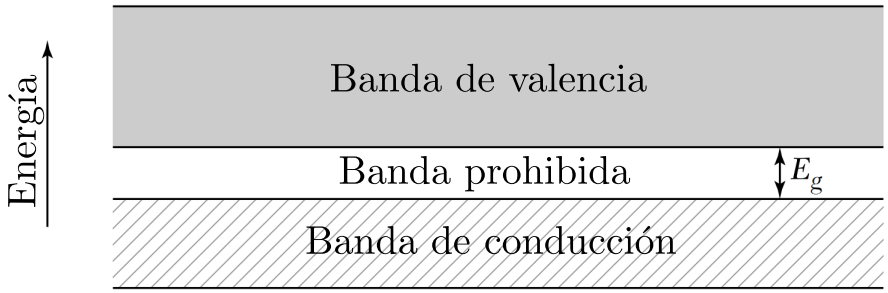


Figura 3.4: Diferencia de energía E_g (Kittel, 2012).

- **Coeficiente de absorción óptica**: El coeficiente de absorción óptica mide la cantidad de luz de una longitud de onda específica que un material puede absorber. Este parámetro es crucial en dispositivos como celdas solares y fotodetectores, donde maximizar

la captación de luz es esencial. La capacidad de absorción depende de la energía del fotón incidente y del band gap del material. Los materiales con un alto coeficiente de absorción son ideales para aplicaciones ópticas que requieren la interacción eficiente con la luz (Ali, 1993).

- **Estructura cristalina**: La estructura cristalina se refiere a la disposición ordenada y repetitiva de átomos, iones o moléculas en un material sólido. Se define mediante una celda unitaria, el bloque básico que se repite en las tres dimensiones del espacio, caracterizada por los parámetros de red: las longitudes de los ejes a, b y c, y los ángulos entre ellos (α, β, γ). Existen siete sistemas cristalinos principales (ver Figura 3.5): *cúbico*, *tetragonal*, *ortorrómbico*, *hexagonal*, *trigonal* (o *romboédrico*), *monoclínico* y *triclínico*, los cuales determinan la simetría y propiedades físicas del material. Esta organización atómica influye en propiedades como la dureza, la conductividad térmica y eléctrica, y las interacciones ópticas, siendo fundamental para el diseño y análisis de materiales (Ali, 1993).

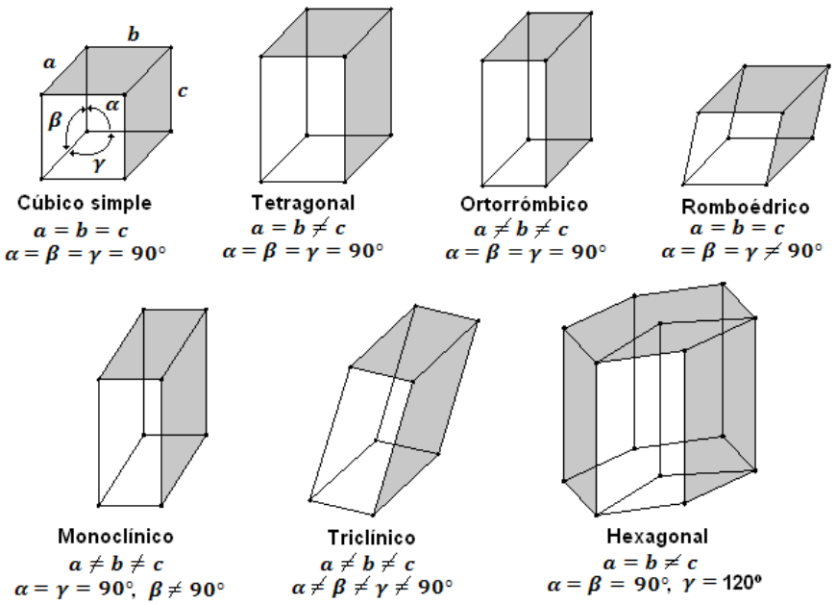


Figura 3.5: Tipos de estructuras cristalinas, (Rosas, 2013).

3.1.4. Celdas solares de Pirita

La Pirita (FeS_2) es un mineral compuesto de hierro y azufre que presenta características interesantes para su uso en celdas solares. Posee una estructura cristalina cúbica, similar a la del NaCl, en la cual

cada átomo de hierro está rodeado de seis átomos de azufre y viceversa. Esta estructura le confiere propiedades electrónicas y ópticas que la hacen atractiva para aplicaciones fotovoltaicas (Schmidt y McIntyre, 2008).

Uno de los aspectos más notables de la pirita es su band gap de aproximadamente 0.95 eV, y aunque este valor es ligeramente menor al ideal para la máxima eficiencia en celdas solares, aún permite una absorción significativa de la luz solar, especialmente en la región visible del espectro; además, la pirita tiene un coeficiente de absorción extremadamente alto, lo que significa que puede capturar la energía de la luz solar de manera eficiente en capas muy delgadas, lo cual es particularmente útil para celdas solares de película delgada ya que permite reducir el grosor del material activo y, por ende, los costos de material y fabricación (Lu y cols., 2021).

Otro punto a favor de la pirita es su abundancia y bajo costo. A diferencia de otros materiales utilizados en celdas solares, como el telurio o el indio, la pirita es uno de los minerales más comunes en la corteza terrestre. Esto no solo reduce los costos de producción, sino que también la convierte en una opción ambientalmente sostenible. En un contexto donde la economía y la sostenibilidad son factores clave en el desarrollo de tecnologías solares, la pirita emerge como una opción prometedora para el futuro de las celdas solares (Department of Earth Sciences, University of Minnesota, 2021).

3.2. Bases teóricas de ciencia de datos y ML

3.2.1. Inteligencia artificial

La inteligencia artificial (IA) es la capacidad de las máquinas para emplear algoritmos, aprender a partir de datos y aplicar ese aprendizaje en la toma de decisiones de manera similar a un ser humano. A diferencia de los seres humanos, los sistemas de IA pueden operar sin descanso, procesando grandes volúmenes de información al mismo tiempo y con una tasa de errores mucho menor (Rouhiainen, 2018).

Por otro lado, la ciencia de datos es una herramienta esencial para el análisis y aprovechamiento de la información, permitiendo la generación de conocimiento. Su propósito incluye desarrollar modelos que identifiquen patrones y comportamientos en los datos, con el objetivo de apoyar la toma de decisiones y realizar predicciones (García y cols.,

2018).

Mitchell y cols. (2019) afirman que a ciencia de datos y la IA tienen una relación estrecha y complementaria, pues la ciencia de datos recopila y analiza grandes volúmenes de información, mientras que la IA emplea estos datos para entrenar modelos capaces de replicar capacidades humanas. Juntas, optimizan procesos, mejoran la precisión de predicciones y apoyan la toma de decisiones en campos como la medicina, el marketing, las finanzas y la tecnología.

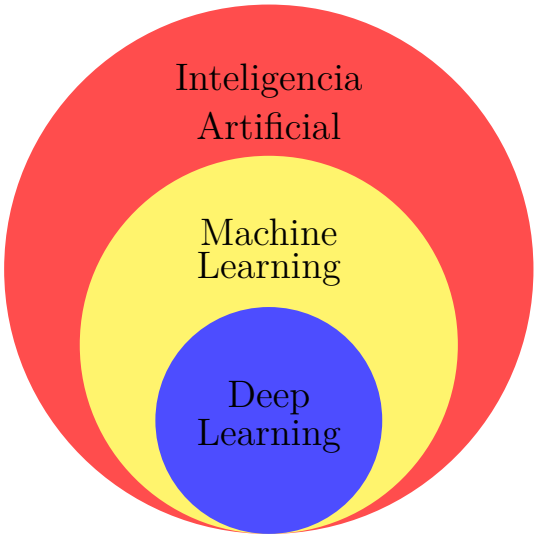


Figura 3.6: Relación entre IA, ML y DL. Elaboración propia.

La IA abarca diversas técnicas, entre ellas el aprendizaje automático (Machine Learning, ML), que permite a las máquinas aprender de los datos sin una programación específica para cada tarea. Dentro de ML, el aprendizaje profundo (Deep Learning, DL) utiliza redes neuronales complejas para analizar grandes volúmenes de datos y resolver problemas avanzados, como el reconocimiento de imágenes y el procesamiento de lenguaje. De este modo, IA, ML y DL se relacionan en una estructura jerárquica (ver Figura 3.6), donde cada nivel se vuelve más especializado y capaz de abordar tareas más complejas con mayor autonomía (Russell y Norvig, 2016).

3.2.2. Machine Learning

El Machine Learning, es una disciplina dentro de la IA y utiliza algoritmos para que las computadoras identifiquen patrones en grandes volúmenes de datos y generen predicciones de forma autónoma, sin necesidad de programación específica para cada tarea (Alpaydin, 2020).

A diferencia del ML, el cual utiliza algoritmos estadísticos y modelos más simples para identificar patrones en los datos, el Deep Learning se basa en redes neuronales que imitan el funcionamiento del cerebro humano para realizar tareas aún más complejas. Sin embargo, en este proyecto de tesis no se aborda esta disciplina.

Tipos de ML

Según Mirjalili y Raschka (2020) existen tres tipos de ML:

- **Aprendizaje supervisado:** es un tipo de algoritmo que utiliza datos etiquetados para aprender y crear un modelo capaz de hacer predicciones sobre nuevos datos. El objetivo es entrenar el modelo con ejemplos en los que se conocen las etiquetas o resultados esperados, permitiéndole así realizar predicciones en datos futuros o desconocidos.
- **Aprendizaje no supervisado:** es un tipo de algoritmo que trabaja con datos sin etiquetar, lo que significa que no tiene ejemplos con resultados conocidos para aprender. El objetivo principal es identificar patrones o estructuras ocultas en los datos, como agrupaciones o relaciones, sin tener una respuesta correcta predefinida. Este enfoque es útil para segmentar datos, reducir su dimensionalidad o encontrar asociaciones.
- **Aprendizaje reforzado:** es una técnica en la que un agente aprende a tomar decisiones a través de prueba y error, interactuando con un entorno y recibiendo recompensas o castigos según las acciones que realiza. El objetivo es maximizar las recompensas acumuladas a largo plazo, ajustando las estrategias basadas en los resultados obtenidos, lo que es ideal para resolver problemas secuenciales y optimizar comportamientos en entornos complejos.

Aprendizaje supervisado	-Datos etiquetados -Feedback directo -Predicción de resultados/futuro
Aprendizaje no supervisado	-Datos sin etiquedas -Sin Feedback -Encontrar estructuras ocultas en los datos
Aprendizaje reforzado	-Proceso de decisión -Sistema de recompensa -Aprender series de acciones

Figura 3.7: Tipos de ML. Elaboración propia.

3.2.3. *K*-Means

El ML abarca una amplia variedad de tareas y aplicaciones prácticas que permiten automatizar procesos complejos mediante el análisis de datos. Algunas de las tareas comunes incluyen la clasificación de datos para facilitar la identificación de patrones y estructuras que se encuentran implícitas.

El método K-Means es un tipo de **aprendizaje no supervisado**, ya que se utiliza para agrupar datos sin etiquetar en un número predefinido de grupos o *clusters* . Este algoritmo busca patrones dentro de los datos para dividirlos en k grupos, de modo que los datos dentro de cada grupo sean más similares entre sí que con los de otros grupos. Dado que no se cuenta con etiquetas o resultados conocidos para cada dato, el algoritmo debe encontrar estas agrupaciones por sí mismo, identificando similitudes entre los datos. Esto lo convierte en una técnica clásica de *clustering* dentro del aprendizaje no supervisado (Alpaydin, 2020).

La diferencia principal entre K-Means y kNN es que kNN es un algoritmo supervisado que clasifica nuevos datos basándose en las etiquetas de sus vecinos más cercanos en un conjunto previamente etiquetado. Para facilitar este proyecto de tesis, se utiliza K-Means , ya que los datos obtenidos no están etiquetados y este método es más sencillo de implementar en tales condiciones.

Hiperparámetros

Los parámetros de K-Means controlan aspectos clave del algoritmo, como el número de *clusters* , la inicialización de centroides, el número máximo de iteraciones, el umbral de convergencia, el método de cálculo y la semilla de aleatoriedad. Ajustarlos adecuadamente mejora la precisión y eficiencia del modelo de agrupamiento; además, el algoritmo ofrece diversos parámetros modificables, aunque, según Scikit-learn Developers (2024b), los parámetros fundamentales que se deben declarar son:

- `n_clusters`: define cuántos grupos o *clusters* se quieren formar en los datos. Modificar este parámetro afecta directamente la estructura de los grupos; un valor demasiado bajo puede agrupar elementos dispares, mientras que un valor demasiado alto podría dividir grupos naturales.

- **`init`**: controla cómo se inicializan los centroides. El método "`k-means++`" suele ser preferido porque mejora la velocidad de convergencia y reduce la probabilidad de caer en mínimos locales, produciendo mejores resultados en comparación con una inicialización completamente aleatoria. Cambiar este parámetro afecta la estabilidad y precisión del modelo.
- **`max_iter`**: fija el número máximo de iteraciones que el algoritmo realizará antes de detenerse. Un valor bajo podría interrumpir el proceso antes de alcanzar un óptimo, mientras que un valor demasiado alto puede consumir más tiempo sin mejoras significativas. Ajustarlo implica balancear entre velocidad y precisión.
- **`tol`**: establece la tolerancia o umbral de convergencia, es decir, la mínima mejora en la ubicación de los centroides que el algoritmo acepta como progreso. Un valor bajo puede hacer que el algoritmo busque con mayor precisión, pero aumentando el tiempo de ejecución. Un valor alto permite converger más rápido, aunque con menor exactitud en el agrupamiento.
- **`algorithm`**: selecciona el método de cálculo, siendo "`auto`" y "`full`" opciones comunes. Este parámetro puede afectar la eficiencia del modelo en conjuntos de datos grandes y es útil ajustar el algoritmo según la naturaleza y tamaño de los datos.
- **`random_state`**: es la "semilla" para la aleatoriedad en la inicialización, lo que asegura que los resultados sean reproducibles en cada ejecución. Cambiar este parámetro permite experimentar con distintas inicializaciones y, al usar una semilla fija, se garantiza que los resultados se puedan replicar, facilitando la validación del modelo.

Para optimizar estos parámetros, se suele combinar el análisis gráfico (como el método del codo o silueta para hallar `n_clusters`) con pruebas iterativas o técnicas de búsqueda de hiperparámetros, como `grid_search` o `random_search`. Estas técnicas ayudan a probar varias combinaciones y encontrar los valores que maximicen la calidad del agrupamiento sin comprometer la eficiencia del modelo (Scikit-learn Developers, 2024b).

3.2.4. Método del codo y la silueta

El **método del codo** (*Elbow Method*) es una técnica gráfica para encontrar el valor óptimo de k en el agrupamiento de K-Means . Se basa en calcular la WCSS (Suma de cuadrados dentro del grupo, o Within-Cluster Sums of Squares), que mide la suma de las distancias al cuadrado entre cada punto de un grupo y su centroide. Al aplicar K-Means a diferentes números de *clusters* , se obtienen valores de inercia, representados en un gráfico, que reflejan esta WCSS (Cui y cols., 2020).

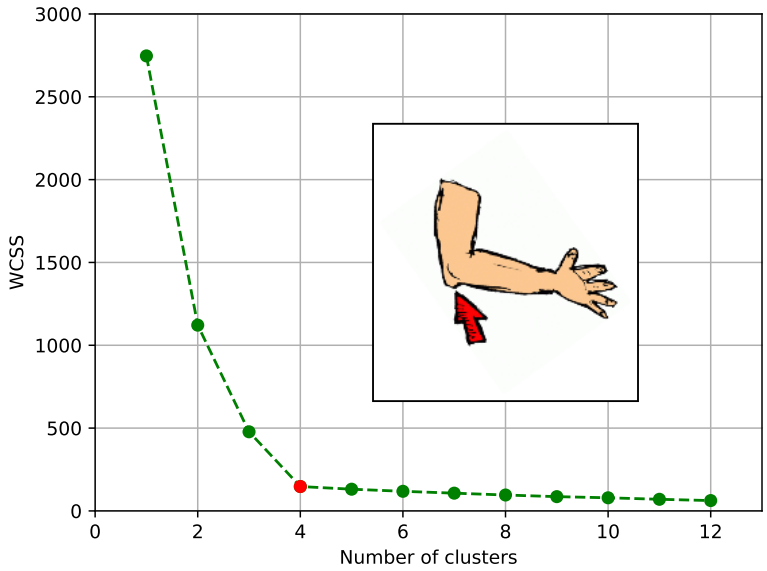


Figura 3.8: Ejemplo del Método del Codo. Elaboración propia.

En la Figura 3.8 se muestra los valores de WCSS en el eje y para distintos valores de k en el eje x. Al observar el gráfico, se identifica un "codo" o punto de inflexión en $k = 4$, donde aumentar el valor de k deja de reducir significativamente la WCSS. Este punto, llamado "punto del codo", se considera el valor óptimo de k para el modelo.

El **método de la silueta** (*Silhouette Method*) se usa para evaluar la distancia de separación entre *clusters* en un agrupamiento. Un gráfico de silueta muestra qué tan cerca están los puntos de un *cluster* respecto a los *clusters* vecinos, permitiendo evaluar visualmente parámetros como el número de *clusters* . Los coeficientes de silueta (*Silhouette coefficients*) van de -1 a +1: valores cercanos a +1 indican puntos bien separados de otros *clusters* , 0 significa que están cerca de los límites de dos *clusters* y valores negativos sugieren que podrían estar en el *cluster* equivocado (Scikit-learn Developers, 2024a).

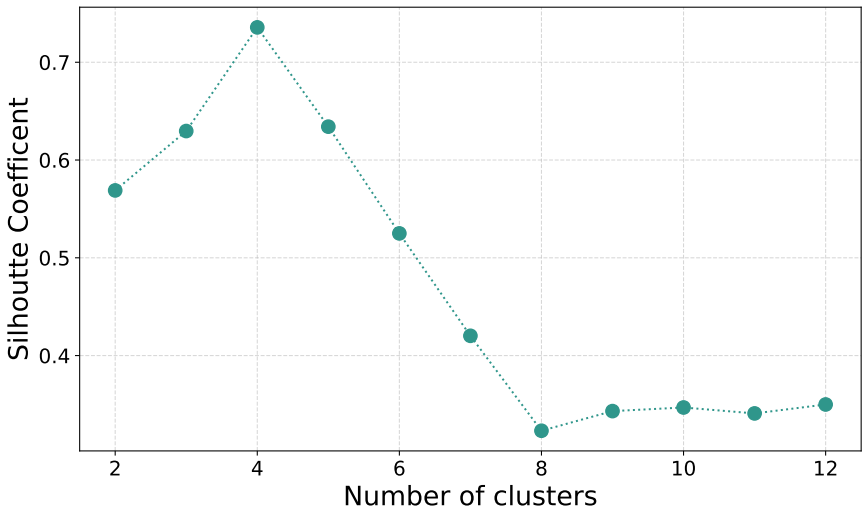


Figura 3.9: Ejemplo del Método de la Silueta. Elaboración propia.

En la Figura 3.9, se muestran los valores de los coeficientes de silueta en el eje y para diferentes valores de k en el eje x. Al analizar el gráfico, se observa un valor máximo en $k = 4$, donde se obtuvo el mayor coeficiente. Comparado con el método del codo, para este ejemplo ambos métodos coinciden en que el número ideal de *clusters* es 4. Sin embargo, el método de silueta es más visual y claro, mientras que el punto de inflexión en el método del codo no siempre será evidente en todos los casos.

3.3. Herramientas de programación

3.3.1. Lenguaje de programación: Python

Python es un popular lenguaje de programación creado por Guido van Rossum en 1991, que se utiliza en desarrollo web, desarrollo de software, matemáticas y scripting de sistemas; permite crear aplicaciones web, integrarse con software para generar flujos de trabajo, conectarse a sistemas de bases de datos y manejar grandes volúmenes de datos o realizar cálculos complejos, con una sintaxis simple y similar al inglés que facilita escribir programas con menos líneas de código, además de ser multiplataforma y poder utilizarse de manera procedimental, orientada a objetos o funcional (W3Schools, 2018).

Una de las grandes ventajas de Python es que es un lenguaje de código abierto, lo cual permite a la comunidad global de desarrolladores contribuir continuamente a su mejora y evolución. Además, Python

es completamente multiplataforma, compatible con sistemas operativos como Windows, Linux y Mac, lo que facilita su uso en distintos entornos y aplicaciones (Saabith, Vinothraj, y Fareez, 2020).

Destaca también su extensa colección de bibliotecas y módulos, lo cual cubre una variedad inmensa de necesidades, desde procesamiento de datos hasta desarrollo web. En particular, Python ofrece poderosas herramientas en el ámbito del aprendizaje automático y la inteligencia artificial, con librerías como TensorFlow, Scikit-learn y PyTorch, que proporcionan funcionalidades avanzadas y simplifican el desarrollo de proyectos complejos en estas áreas (Unpingco, 2021).

Bibliotecas utilizadas

En este proyecto, se requirieron bibliotecas especializadas en manipulación y visualización de datos, cálculo numérico y aprendizaje automático. A continuación, se presenta una breve descripción de las principales bibliotecas utilizadas, según las definiciones de Unpingco (2021).

- **Cálculo numérico**
  - `NumPy`: Es una biblioteca fundamental para el cálculo científico en Python. Proporciona soporte para arreglos multidimensionales (`arrays`) y una gran colección de funciones matemáticas para operaciones rápidas y eficientes. Es especialmente útil para cálculos numéricos y álgebra lineal, y se usa mucho en ciencia de datos y aprendizaje automático.
- **Manipulación de datos**
  - `Pandas`: Es una biblioteca diseñada para la manipulación y análisis de datos. Ofrece estructuras de datos como `DataFrame` (tablas) y `Series` (columnas), que facilitan el manejo, limpieza y transformación de datos. Es muy usada en ciencia de datos y análisis, permitiendo cargar, procesar y visualizar datos de manera eficiente.
- **Visualización de datos**
  - `Matplotlib`: Es una biblioteca de visualización que permite crear gráficos estáticos, animados e interactivos en Python.

Con `Matplotlib`, se pueden generar gráficos de líneas, barras, dispersión, histogramas, entre otros, lo que es ideal para representar datos y patrones de manera visual.

- **Machine Learning**
  - **`Sckit-Learn`**: Es una biblioteca para aprendizaje automático en Python. Incluye herramientas para clasificación, regresión, agrupación (*clustering*), reducción de dimensionalidad y más. `Scikit-Learn` está diseñada para ser fácil de usar, y se utiliza mucho en proyectos de machine learning y análisis predictivo.

* **Biblioteca especial de materiales**
  - **`mp_api.client`**: Es una herramienta en Python que permite interactuar de manera programática con la API de **Materials Project**, una plataforma que proporciona datos sobre las propiedades de materiales. Con esta biblioteca, los usuarios pueden acceder de forma gratuita a una extensa base de datos de propiedades de materiales (Materials Project, 2019).

3.3.2. Uso de API

Una API (Interfaz de Programación de Aplicaciones, por sus siglas en inglés: *Application Programming Interface*) es un conjunto de reglas y definiciones que permite que diferentes programas o aplicaciones se comuniquen entre sí. Las API actúan como intermediarios que facilitan la interacción entre un sistema y otro, permitiendo a los desarrolladores acceder a ciertas funcionalidades o datos de una aplicación sin necesidad de entender su funcionamiento interno completo (Amazon Web Services, 2021).

En términos simples, una API proporciona una manera estructurada para que los programas realicen solicitudes y reciban respuestas. Esto es común en aplicaciones que necesitan acceder a información o servicios de terceros, como base de datos. En este proyecto, el uso de una API fue fundamental para acceder a datos de miles de materiales de estado sólido, permitiendo así recopilar y organizar la información necesaria para alimentar el algoritmo de agrupación K-Means .

Capítulo 4

Metodología

4.1. Tipo y enfoque de la investigación

Esta investigación es de tipo exploratoria, con un enfoque cuantitativo e interpretativo. Su desarrollo se realizó mediante la creación de una base de datos y la implementación del algoritmo K-Means con un propósito aplicativo: proponer una solución a un problema específico en el área de sistemas fotovoltaicos.

El carácter exploratorio de esta investigación se debe a que aborda un área poco estudiada que fusiona la ciencia de datos con los sistemas fotovoltaicos. Esto abre la posibilidad de futuras investigaciones que profundicen en aspectos tanto teóricos como experimentales, contribuyendo a la validación y aplicación práctica de los resultados obtenidos.

Para cumplir con los objetivos de este proyecto de tesis, se utilizó una metodología basada en los estudios presentados en el Capítulo 2, con la diferencia de que se implementó el algoritmo K-Means en lugar de kNN . Esta elección buscó determinar si el método de agrupación K-Means podría ofrecer resultados similares. El desarrollo del proceso se estructuró en las siguientes fases:

a) Obtención de datos
b) Filtrado de datos
c) Preparación de datos
d) Implementación del algoritmo K-Means
e) Análisis de resultados

La implementación del K-Means requiere una base de datos sólida y representativa, que proporcione al algoritmo los datos necesarios para realizar predicciones precisas. No obstante, la base de datos (BD) no se pudo obtener a partir de las fuentes citadas en el Capítulo 2, por lo que fue necesario crear una.

4.2. Obtención de datos

Se realizó una búsqueda exhaustiva de repositorios que ofrecieran información detallada y confiable sobre materiales. Además, era fundamental que el repositorio seleccionado permitiera la descarga masiva de dicha información y que su acceso fuera libre y gratuito. Tras evaluar diversas alternativas, se concluyó que el portal **The Materials Project**[1] era el más adecuado, ya que proporciona datos sobre materiales conocidos de estado sólido, además de información sobre miles de materiales aún no catalogados.

4.2.1. The Materials Project

The Materials Project , fundado en 2011 a partir de la investigación postdoctoral de Kristin Persson, es uno de los repositorios de materiales más grandes disponibles en internet, con alrededor de 154,718 registros. Este portal permite a cualquier usuario acceder a su información mediante una API gratuita compatible con Python (Jain y Ong, 2013).

Para descargar los datos es necesario registrarse en la API[2] y generar una "API key" personal que otorga acceso a la BD de **The Materials Project**. A continuación, es necesario instalar la siguiente librería de Python:

```
pip install mp-api
```

Una vez obtenida la "API key" e instalada la librería `mp-api`, es posible acceder a la información de cualquier material registrado en **The Materials Project**. Cada material en la BD está identificado por un código o *id*, que comienza con las letras "`mp`" seguidas de un número entero. Este identificador no es secuencial, ya que depende del número de cálculos realizados sobre el material, de descubrimientos o de otras propiedades; por lo tanto, no es correcto asumir que existen *ids* consecutivos desde `mp-0` hasta `mp-154717`.

4.2.2. Descarga de datos

Para obtener los *ids* disponibles, se desarrolló el código mostrado en el Código 7.1, con el cual se extrajeron 143,666 *ids* de un total de

[1] Sitio web oficial: https://next-gen.materialsproject.org/
[2] Sitio web oficial: https://next-gen.materialsproject.org/api

154,718. La cantidad de materiales resultó ser suficiente para el análisis requerido, ya que se descargaron casi todos los materiales disponibles en la base de datos. Posteriormente, se creó un archivo `.csv` con los 143,666 *ids* disponibles, y mediante el Código 7.2, se realizaron las siguientes solicitudes (*requests*) para cada *id* :

a) Fórmula química
b) Número total de elementos en la fórmula química
c) Elementos presentes en la fórmula química
d) Band Gap
e) Si es un material metálico o no
f) Si el Band Gap es directo o indirecto
g) Tipo de estructura cristalina
h) Parámetros de red a, b y c
i) Ángulos de red α, β y γ
j) Coeficiente de absorción óptica

Con la ayuda de la librería `Pandas` de Python, toda la información descargada se almacenó en un `DataFrame` , que luego fue exportado en formato `.csv` para su posterior procesamiento.

4.3. Filtrado de datos

El `DataFrame` que almacena los datos crudos de los 143,666 materiales contiene las siguientes columnas:

Columna	Descripción	Unidad	Tipo de dato
`id`	*id* del material	-	object
`Formule`	Fórmula química	-	object
`No Ele`	Número total de elementos contenidos en la fórmula química	-	int
`Ele`	Elementos de la fórmula química	-	object
`BG`	Band Gap	eV	float
`IM`	Si es metal o no	-	bool
`Is BG Dir`	Si su Band Gap es directo o no	-	bool
`Structure`	Tipo de estructura cristalina	-	object
`a`	Parámetro de red a	Å	float
`b`	Parámetro de red b	Å	float
`c`	Parámetro de red c	Å	float
`alfa`	Parámetro de red α	°	float
`beta`	Parámetro de red β	°	float
`gamma`	Parámetro de red γ	°	float
`CA`	Coeficiente de absorción óptica para $\lambda = 500$ nm	1/cm	float

Tabla 4.1: Descripción del `DataFrame` correspondiente al Código 7.2

Las columnas `Formule` , `No Ele` y `Ele` contienen información química que sirve para el análisis de resultados. De la misma forma, las columnas `Is BG Dir` , `a` , `b` , `c` , `alfa` , `beta` y `gamma` poseen información relevante sobre la estructura cristalina del compuesto y sirven para un análisis posterior. Únicamente las columnas `BG` , `IM` , `Structure` y `CA` son fundamentales para el filtrado de datos.

El primer criterio de filtrado consistió en eliminar los materiales metálicos, dado que no poseen Band Gap y su alta conductividad impide la formación de junturas PN. El segundo criterio fue excluir los materiales no metálicos que no tenían un coeficiente de absorción óptica registrado, ya que **The Materials Project** no cuenta con datos completos sobre este coeficiente para todos los materiales. Finalmente, se filtraron los materiales no metálicos que presentaban un coeficiente de absorción óptica y un Band Gap en el rango de 0.95 eV a 3 eV, ya que este rango es óptimo para su aplicación como capa ventana.

El Código 7.3 aplica filtros mediante máscaras en el `DataFrame` , resultando en un total de 535 no metales con coeficiente de absorción y un Band Gap en el rango de deseado. Los demás resultados son:

Total de metales:	70,108
Total de no metales:	73,558
No metales con CA:	940
No Metales con CA y $0.95\,\text{eV} < \text{BG} < 3\,\text{eV}$:	535
Total de materiales:	143,666

4.4. Preparación de datos

Los datos filtrados requieren un procesamiento adicional antes de ser utilizados por el algoritmo kNN para asegurar su correcto funcionamiento. Este procesamiento es necesario debido a la naturaleza original de los datos, que pueden incluir tanto valores numéricos como categóricos. El algoritmo kNN , al operar principalmente con datos normalizados, necesita que todos los valores sean convertidos a un formato numérico. Esta normalización garantiza que todas las características tengan la misma escala, lo que permite al algoritmo calcular distancias de manera precisa y consistente.

Los datos fundamentales para el agrupamiento de los materiales son: `BG` , `Structure` y `CA` . Estos tres parámetros proporcionan una descripción general de las características del material, ya que abarcan tanto las propiedades electrónicas como estructurales. Debido a

su relevancia, permiten realizar una comparación a priori entre los diferentes materiales.

En la Tabla 4.1 se muestra que los datos almacenados en `Structure` no son numéricos, por lo que se cambiaron los valores cualitativos a cuantitativos asignando un valor numérico a cada tipo de estructura:

Estructura	Valor
Cúbica	1
Tetragonal	2
Ortorrómbica	3
Hexagonal	4
Trigonal	5
Monoclínica	6

Tabla 4.2: Asignación numérica al tipo de estructura cristalina

Posteriormente, se redujo el `DataFrame` que contenía los datos filtrados, conservando únicamente las columnas `Formula`, `Ele`, `BG`, `Structure` y `CA`. Luego, se estandarizaron las variables `BG`, `Structure` y `CA`, ya que estas servirán para el proceso de agrupamiento. En el Código 7.4 se encuentra dicha estandarización, reducción del `DataFrame` y el cambio de variable para `Structure` .

4.5. Implementación del algoritmo *K*-Means

Para implementar este algoritmo de agrupamiento, se realizó una selección cuidadosa de los hiperparámetros adecuados para garantizar su correcto funcionamiento. Este paso es crucial, ya que los hiperparámetros determinan el comportamiento y la eficiencia del algoritmo. Una vez definidos los hiperparámetros, se determinó el número óptimo de *clusters* o grupos en los que se dividirán los datos. Este proceso de clasificación es esencial para asegurar que los datos se agrupen de manera coherente y que los resultados obtenidos reflejen con precisión los patrones o características presentes en cada conjunto de datos.

4.5.1. Optimización de hiperparámetros

Esta tarea se llevó a cabo utilizando `grid search` implementado en `GridSearchCV` de la biblioteca `Scikit-learn`. Se utilizó el Código 7.5 para optimizar los hiperparámetros de *K*-Means mediante `GridSearchCV` , evaluando diferentes combinaciones de los paráme-

tros, para obtener como resultado la combinación óptima que minimiza el error, maximizando la eficiencia y precisión del algoritmo.

4.5.2. Método del codo y la silueta

Existen varios métodos para determinar el número adecuado de *clusters* , y aunque `GridSearchCV` puede ayudar con esta tarea, se optó por implementar los métodos del codo (*Elbow Method*) y la silueta (*Silhouette Method*). De estos, se eligió el método de la silueta como el principal, utilizando el método del codo como apoyo adicional.

En el Código 7.6 se definió un rango de 30 *clusters* (del 2 al 32) para realizar el barrido. Posteriormente, utilizando la métrica `inertia`, se graficó la suma de las distancias cuadradas dentro de cada cluster, lo que permite evaluar la compactación de los grupos formados. Adicionalmente, se empleó `silhouette_score` para graficar el *Silhouette Coefficient* correspondiente a cada número de *clusters* , proporcionando una medida de la separación y coherencia de los puntos dentro de los *clusters* . Esto permitió evaluar el número óptimo de *clusters* a partir de ambas métricas.

4.5.3. Predicción de *clusters*

Finalmente, con el número de *clusters* y los parámetros óptimos ya definidos, se realizó la predicción y agrupamiento de los datos filtrados. En el Código 7.7, se implementó el algoritmo K-Means configurado con los parámetros mediante `GridSearchCV` y por el método de la silueta. Los resultados de las bases de datos y del algoritmo se muestran en el siguiente capítulo.

Capítulo 5

Resultados y discusión

5.1. Análisis exploratorio de datos

Las dos primeras secciones del capítulo anterior tuvieron como objetivo principal la creación de una base de datos. Como resultado, se obtuvieron datos filtrados de 535 materiales, con valores de `CA` y `BG` entre 0.95 y 3 eV. La Figura 5.1 muestra un fragmento del `DataFrame` donde se almacenan los datos filtrados, para consultar la base de datos completa, ver la Sección 7.2 Apéndice B. El algoritmo K-Means utilizó dichos datos para la agrupación de los materiales y los resultados se describen en las siguientes secciones de este capítulo.

	id	Formule	No Ele	Ele	BG	IM	Is BG Dir	Structure	a	b	c	alfa	beta	gamma	CA
0	mp-239	BaS3	2	Ba-S	1.3913	False	False	Tetragonal	4.216011	6.951955	6.951955	90.000000	90.000000	90.000000	62546.518850
1	mp-241	CdF2	2	Cd-F	2.8977	False	False	Cubic	3.819109	3.819111	3.819110	60.000012	60.000000	60.000017	1582.356544
2	mp-252	BeTe	2	Be-Te	2.0173	False	False	Cubic	4.004287	4.004288	4.004287	60.000014	60.000010	60.000006	10109.303511
3	mp-375	UO3	2	O-U	1.6350	False	False	Cubic	4.135416	4.135416	4.135416	90.000000	90.000000	90.000000	228128.774945
4	mp-441	Rb2Te	2	Rb-Te	1.8766	False	False	Cubic	6.054396	6.054396	6.054397	60.000001	60.000001	59.999998	65916.018982
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
530	mp-999472	NaLaSe2	3	La-Na-Se	2.2767	False	False	Trigonal	4.373467	4.373468	7.322635	72.624865	72.624866	59.999991	31385.701456
531	mp-999474	NaHoSe2	3	Ho-Na-Se	1.8867	False	False	Trigonal	4.098759	4.098760	7.319147	73.739649	73.739643	60.000008	26900.545035
532	mp-999488	NaDySe2	3	Dy-Na-Se	1.8635	False	False	Trigonal	4.116628	4.116628	7.313150	73.653005	73.653010	59.999996	30314.566053
533	mp-999489	NaGdSe2	3	Gd-Na-Se	1.3585	False	False	Trigonal	4.171647	4.171665	7.318768	73.464234	73.463315	59.969785	121914.219728
534	mp-999490	NaDyS2	3	Dy-Na-S	2.2856	False	False	Trigonal	3.967093	3.967093	6.979570	73.489369	73.489364	60.000001	11759.128473

Figura 5.1: Fragmento del `DataFrame` con los datos filtrados

Por otra parte, el método del codo y el análisis de silueta, aplicados para determinar el número óptimo de *clusters* , arrojaron la distribución visualizada en la Figura 5.2. En esta figura, es posible identificar el valor máximo del *Silhouette Coefficient*, así como el valor de WCSS (*Within-Cluster Sum of Squares*) que corresponde al punto de inflexión. De esta forma, se logró una segmentación efectiva y optimizada con un total de **siete** *clusters* .

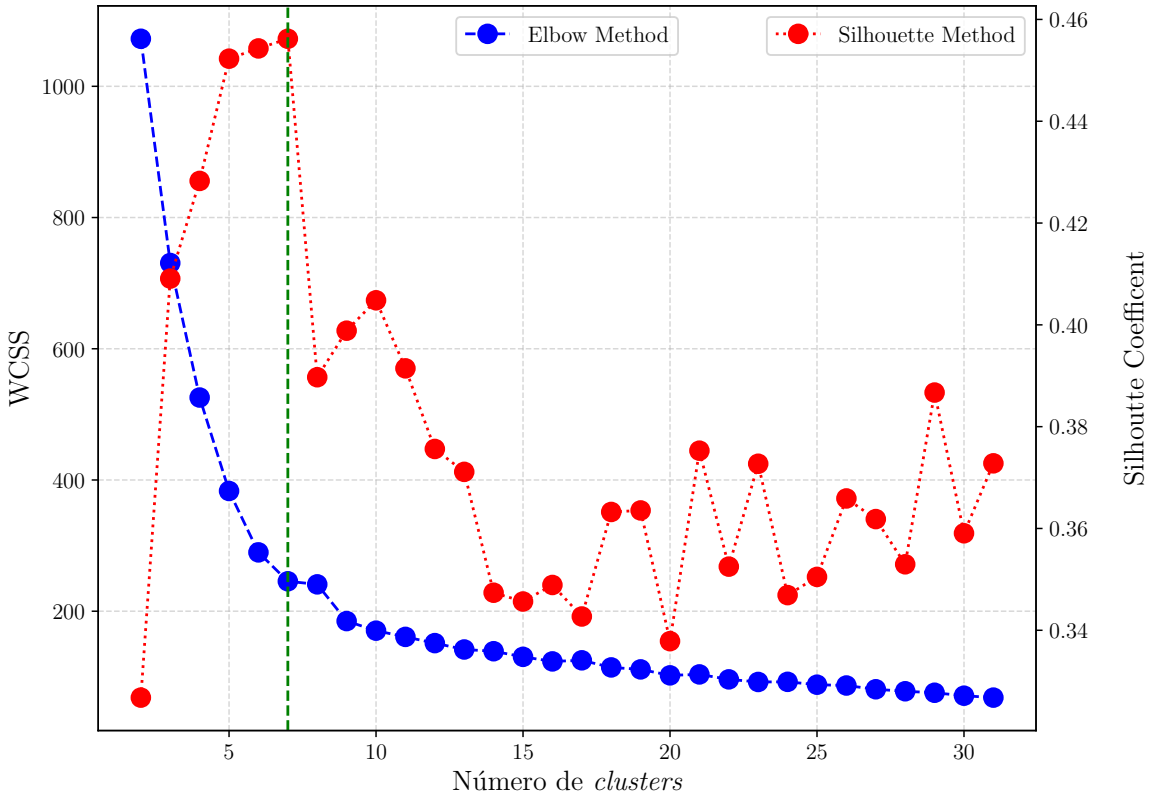


Figura 5.2: Resultados del método del codo y la silueta

El algoritmo *K*-Means fue configurado utilizando los parámetros óptimos obtenidos a partir de `grid_search.best_params_`, permitiendo agrupar los 535 datos analizados en siete grupos. La Figura 5.3 muestra un fragmento del `DataFrame` donde se almacenan los datos filtrados con su respetiva asignación de *clusters* , para consultar la base de datos completa, ver la Sección 7.2 Apéndice B.

	Formule	Ele	BG	Structure	CA	cluster
0	BaS3	Ba-S	1.3913	2	62546.518850	3
1	CdF2	Cd-F	2.8977	1	1582.356544	4
2	BeTe	Be-Te	2.0173	1	10109.303511	4
3	UO3	O-U	1.6350	1	228128.774945	1
4	Rb2Te	Rb-Te	1.8766	1	65916.018982	3
...	...	...	...	...	...	...
530	NaLaSe2	La-Na-Se	2.2767	5	31385.701456	7
531	NaHoSe2	Ho-Na-Se	1.8867	5	26900.545035	2
532	NaDySe2	Dy-Na-Se	1.8635	5	30314.566053	2
533	NaGdSe2	Gd-Na-Se	1.3585	5	121914.219728	2
534	NaDyS2	Dy-Na-S	2.2856	5	11759.128473	7

Figura 5.3: Fragmento del `DataFrame` com los datos agrupados

5.2. Agrupamiento de materiales

Como primer paso en el análisis del agrupamiento, se presenta la Figura 5.4, en la cual se graficó la distribución de materiales dentro de cada *cluster* . Esta gráfica facilita una visión general del tamaño de cada grupo y permite identificar diferencias en la densidad de materiales por *cluster*.

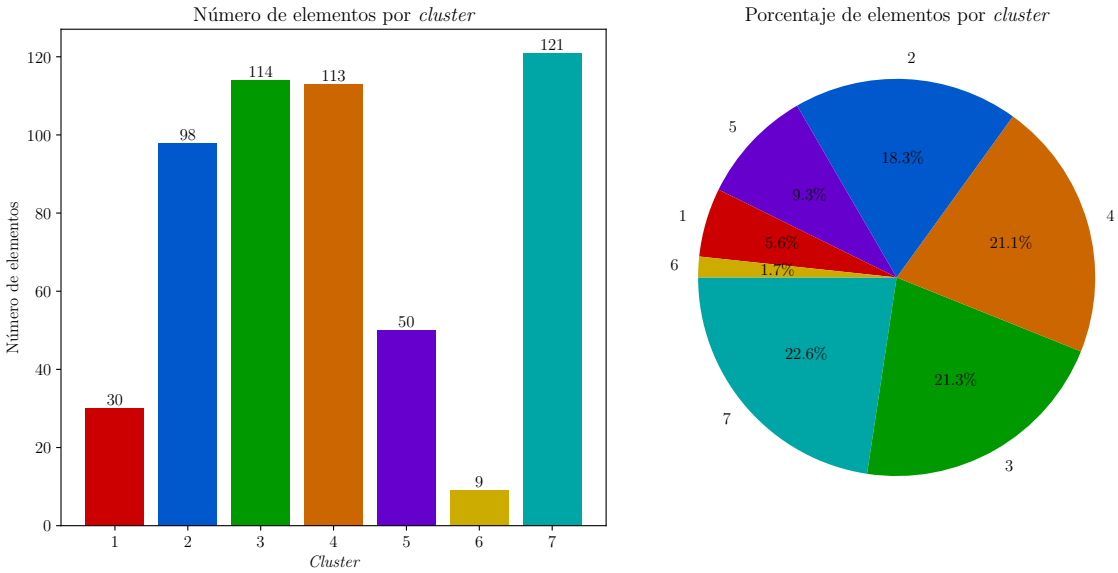


Figura 5.4: Número de materiales por *cluster*

Se observó que los *clusters*: 3, 4 y 7 contienen la mayor cantidad de materiales, con 114, 113 y 121 materiales, respectivamente. En contraste, los *clusters*: 1, 5 y 6 presentan el menor número de materiales, siendo el *cluster*: 6 el de menor cantidad, con solo 9 materiales en total. En términos porcentuales, el *cluster*: 7 representa casi el 23 % del total de materiales, mientras que los *clusters*: 3 y 4 aportan cada uno el 21 % del total.

5.2.1. `BG` y `CA`

Por otro lado, como determinantes para la selección del *cluster* idóneo se requiere que el `BG` de los materiales sea mucho mayor al de la Pirita y que posean un bajo `CA` para que funjan como capa ventana. Siguiendo estas determinantes se requirió de la Figura 5.5 para analizar el `BG` y `CA` por *cluster* .

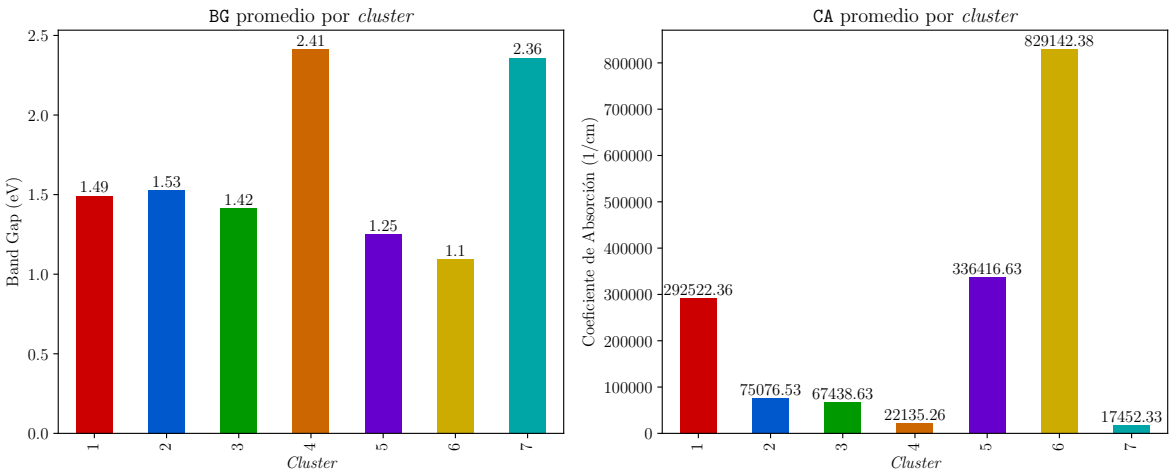


Figura 5.5: BG y CA promedio por *cluster*

La primera gráfica de la Figura 5.5 analiza el BG promedio por *cluster* , destacando a los *clusters*: 4 y 7, ya que presentan un BG superior a 2.3 eV, mientras que los valores en los demás *clusters* se encuentran entre 1.1 y 1.5 eV. En contraste, la segunda gráfica muestra que los *clusters*: 4 y 7 tienen el menor CA promedio, sugiriendo que estos materiales tienen una baja capacidad de absorción de luz y son más transparentes en comparación con los otros *clusters* .

La Figura 5.5 también muestra una tendencia inversa entre el valor de BG y CA en algunos *clusters*. Por ejemplo, los *clusters* con valores altos de BG (como el 4 y 7) tienden a tener valores bajos de CA , mientras que aquellos con valores bajos de BG (como el *cluster* 6) presentan valores altos de CA . Esta relación es coherente con lo que se esperaría en ciertos materiales, donde un band gap más bajo facilita la absorción de luz debido a la menor energía requerida para excitar electrones. *A priori*, las características de los *clusters*: 4 y 7 los hacen candidatos adecuados para formar una capa ventana idónea con la Pirita.

Cluster	BG [eV]		CA [1/cm]	
	mean	std	mean	std
1	1.49	0.29	292,522.36	90,975.79
2	1.53	0.25	75,076.53	48,696.55
3	1.42	0.26	67,438.63	43,722.31
4	2.41	0.31	22,135.26	29,213.36
5	1.25	0.19	336,416.63	78,406.12
6	1.10	0.21	829,142.38	143,134.20
7	2.36	0.28	17,452.33	21,845.79

Tabla 5.1: Promedio y desviación estándar por *cluster*

La Tabla 5.1 complementa la información presentada en la figura anterior al incluir las desviaciones estándar (std) para `BG` y `CA` . Las desviaciones estándar de `BG` son relativamente bajas, lo que indica una variación moderada dentro de cada *cluster* y sugiere que los materiales dentro de cada grupo tienen características similares en cuanto a su band gap. Por otro lado, la desviación estándar de `CA` es considerablemente alta en algunos *clusters* , lo cual sugiere una variabilidad significativa en la capacidad de absorción de los materiales dentro de los *clusters* .

5.2.2. `Structure`

Un factor crucial a considerar en el crecimiento de una juntura PN es la presencia de defectos en las uniones, ya que estas imperfecciones reducen la eficiencia de la celda solar. Para maximizar el rendimiento en una juntura con Pirita, la cual presenta una estructura cúbica, es necesario emplear materiales con redes cristalinas similares para minimizar defectos en la interfaz de las superficies. La compatibilidad estructural entre materiales reduce la cantidad de irregularidades en la unión, favoreciendo una transferencia eficiente de cargas.

Para este estudio, se realizó una selección de *clusters* basada exclusivamente en la estructura cristalina, sin considerar otros parámetros como las constantes de red o propiedades adicionales de los materiales. Esto responde a un criterio práctico, dado que incorporar más variables en la selección de *clusters* podría aumentar considerablemente la complejidad del proceso, dificultando la identificación de combinaciones óptimas para la Pirita.

La Figura 5.6 agrupa a los materiales según su estructura cristalina. En los *clusters*: 1 y 4, la estructura cúbica destaca con altas frecuencias; para los *clusters*: 2 y 7 destaca la estructura trigonal y los *clusters*: 3 y 4 engloban la mayoría de los materiales con estructuras ortorrómbicas. Las estructuras hexagonales y monoclínicas son poco frecuentes en la mayoría de los *clusters* . La Tabla 5.2 muestra la predominancia de la estructura trigonal entre los 535 materiales, siendo el *cluster* 7 el que engloba más estructuras asimétricas, mientras que en el *cluster* 4 posee principalmente estructuras simétricas.

Estructura	Frecuencia	Predominancia
Cúbica	121	*cluster* 4
Tetragonal	116	*cluster* 3
Ortorrómbica	30	*cluster* 3
Hexagonal	45	*cluster* 7
Trigonal	219	*cluster* 7
Monoclínica	4	*cluster* 7

Tabla 5.2: Distribución de estructuras cristalinas

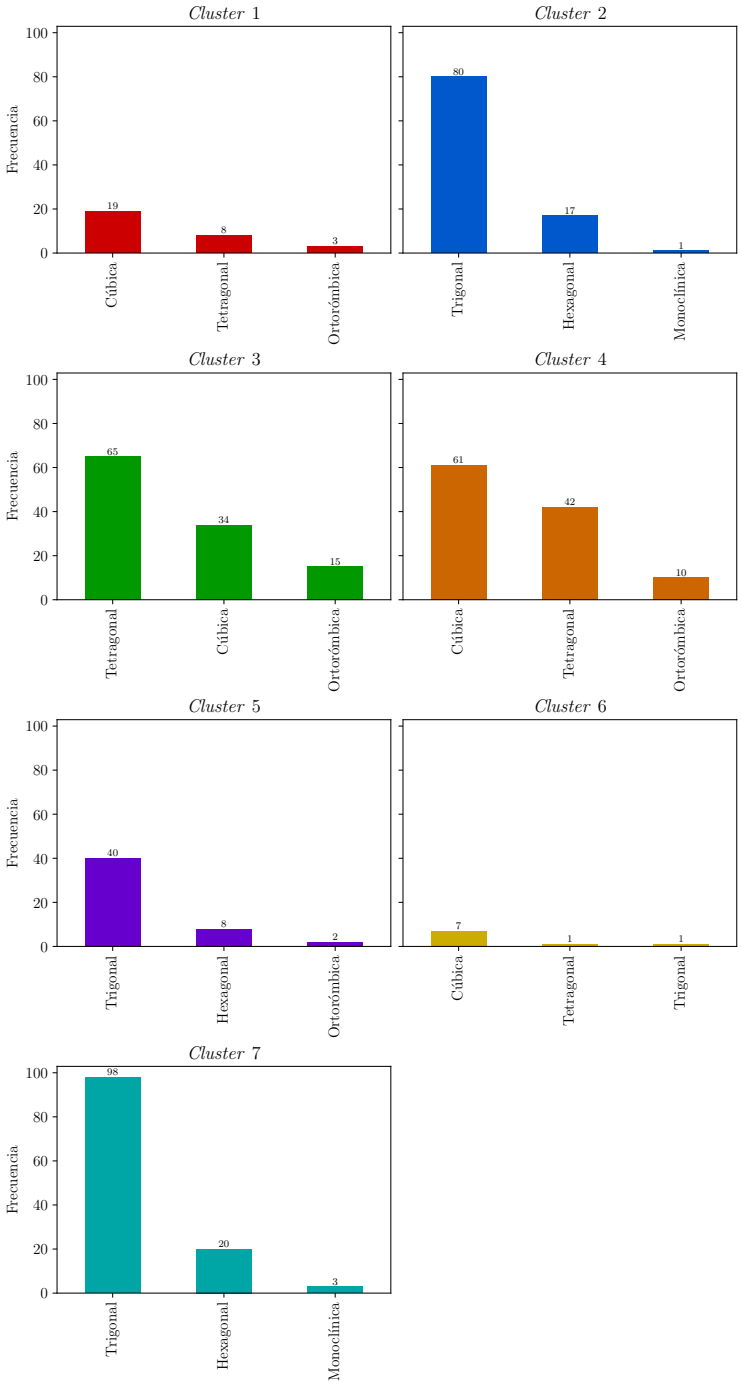


Figura 5.6: Estructuras cristalinas por *clusters*

Hasta ahora las variables `BG`, `CA` y `Structure` se han analizado por separado. Sin embargo, para relacionarlas en un espacio tridimensional se realizó la siguiente gráfica:

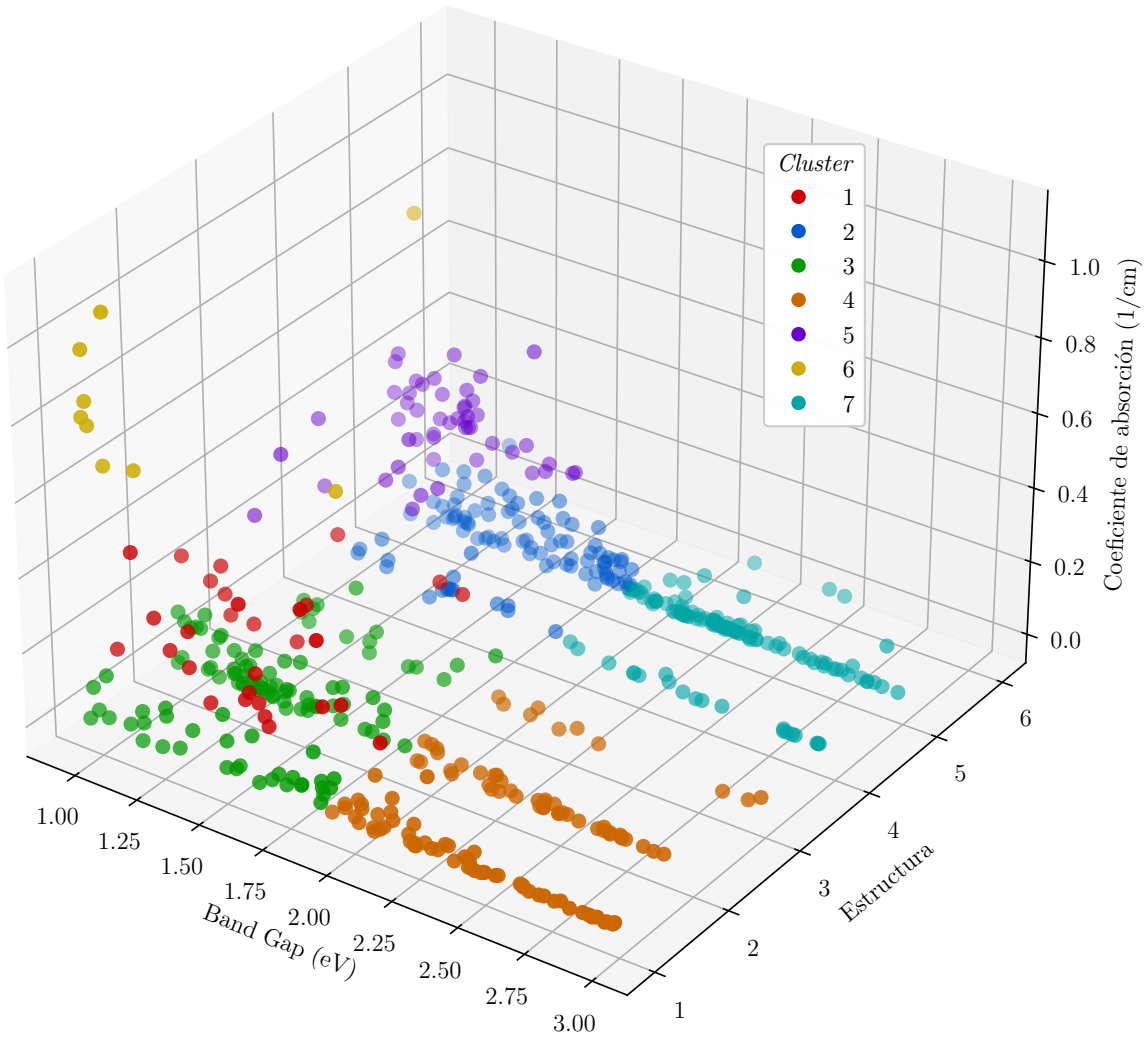


Figura 5.7: Visualización 3D de las variables `BG`, `CA` y `Structure`

La Figura 5.7 representa en el eje x el band gap, en el eje y el tipo de estructura cristalina y en el eje z el coeficiente de absorción. Los siete *clusters* están distribuidos en el espacio tridimensional y se aprecia claramente la delimitación de cada grupo por colores.

Esta gráfica reafirma la idea de proponer el *cluster* 4 como el adecuado para formar una capa ventana, pues el agrupamiento se caracteriza por tener un `BG` en el rango de 2.0 a 2.5 eV, lo cual es ideal para este rol, ya que permite que la mayor parte de la luz visible pase a través de él sin ser absorbida, dejando que la Pirita actúe como la capa absorbente.

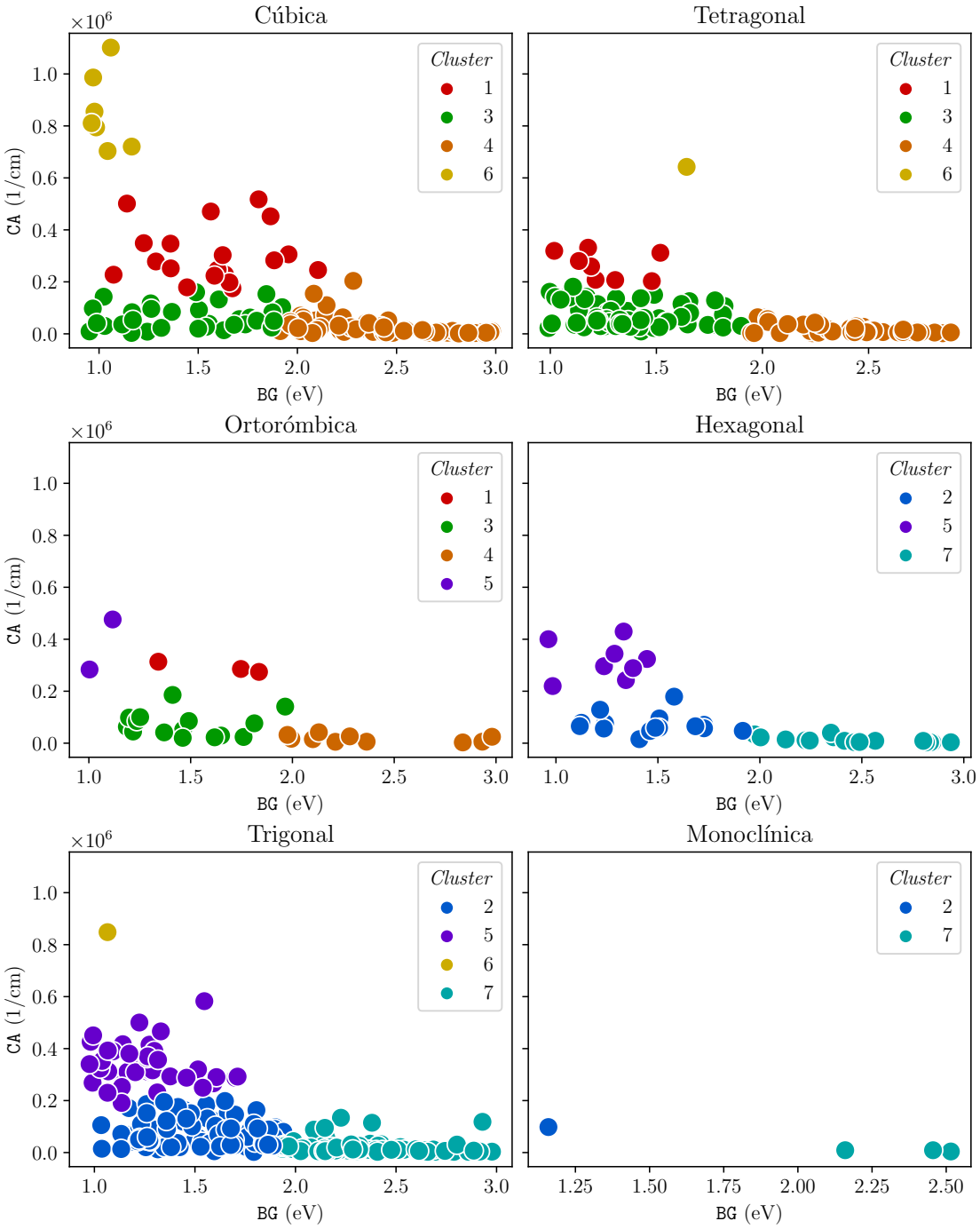


Figura 5.8: `BG` vs. `CA` por tipo de estructura

La Figura 5.8 proporciona cortes transversales de la gráfica 3D anterior, mostrando la relación entre el `BG` y el `CA` para los distintos tipos de estructuras cristalinas. En este análisis, la mayoría de los *clusters* muestran limitaciones significativas para actuar como capa ventana.

El *cluster* 1, identificado en rojo, presenta un band gap bajo (entre 1.0 y 1.75 eV) y un coeficiente de absorción bajo. Sin embargo, el band gap restringido de este grupo hace que absorba una porción de la luz

en lugar de dejarla pasar hacia la Pirita, lo cual perjudica la eficiencia esperada de la capa ventana. De manera similar, los *clusters* 2 y 3, aunque cuentan con un band gap de bajo a intermedio (1.25 a 2.0 eV) y un coeficiente de absorción bajo, no ofrecen un BG lo suficientemente alto para permitir una transmisión eficiente de la luz visible hacia la capa absorbente.

El *cluster* 5, en morado, presenta un rango de BG intermedio (1.5 a 2.5 eV) con una absorción algo variable, lo que sugiere una posible interferencia en el paso de luz debido a su mayor absorción, afectando también la eficiencia de la capa.

El *cluster* 6, en amarillo, con su rango de band gap bajo (1.0 a 1.75 eV) y baja absorción, resulta aún menos adecuado para esta función. Por último, el *cluster* 7, en azul claro, presenta un BG alto y un coeficiente de absorción bajo, lo cual favorece el funcionamiento de la capa ventana; lamentablemente las estructuras cristalinas que predominan en este *cluster* no son compatibles con la Pirita.

En resumen, aunque algunos *clusters* tienen propiedades que permiten cierta transparencia, la mayoría no alcanzan el BG necesario para maximizar la transmisión de luz hacia la capa absorbente, limitando así su idoneidad como capas ventana en una celda solar eficiente. Por otro lado, el *cluster* 4 (naranja) destaca como un candidato ideal para actuar como capa ventana debido a sus propiedades de BG intermedio y bajo coeficiente de absorción. Su capacidad para minimizar la absorción de fotones es crucial, ya que permite que la mayor parte de la energía lumínica pase directamente hacia la capa absorbente, optimizando así la eficiencia de la celda. Además, el *cluster* 4 se compone de estructuras cristalinas simétricas, lo cual facilita su integración con la Pirita sin producir defectos entre capas.

5.2.3. No Ele y Ele

Para asegurar la viabilidad de los materiales propuestos y confirmar los resultados expuestos, es fundamental que estos materiales sean de fácil crecimiento en laboratorio, no tóxicos y económicos. Para llevar a cabo este análisis, se utilizó la Figura 5.9, que muestra la cantidad de compuestos simples, binarios, ternarios y cuaternarios presentes en cada *cluster* .

Dado que los compuestos binarios facilitan la producción y el crecimiento en celdas solares, se decidió darles prioridad en el análisis. De este modo, el *cluster* 4 resalta nuevamente, ya que contiene la ma-

yor cantidad de materiales binarios, lo que lo convierte en un grupo prometedor. En contraste, el *cluster* 7 contiene principalmente materiales ternarios y representa un criterio de exclusión adicional para este grupo, dado su menor viabilidad en términos de simplicidad de producción y costo.

Finalmente, en la Figura 5.10 se presenta un gráfico que muestra la frecuencia de los 20 elementos más abundantes en el *cluster* 4. Este análisis se llevó a cabo con el propósito de evaluar la viabilidad del grupo, considerando evitar elementos tóxicos, de alto costo o clasificados como tierras raras. La gráfica permite observar la abundancia del oxígeno, lo cual supone la predominancia de óxidos no metálicos en el *cluster* .

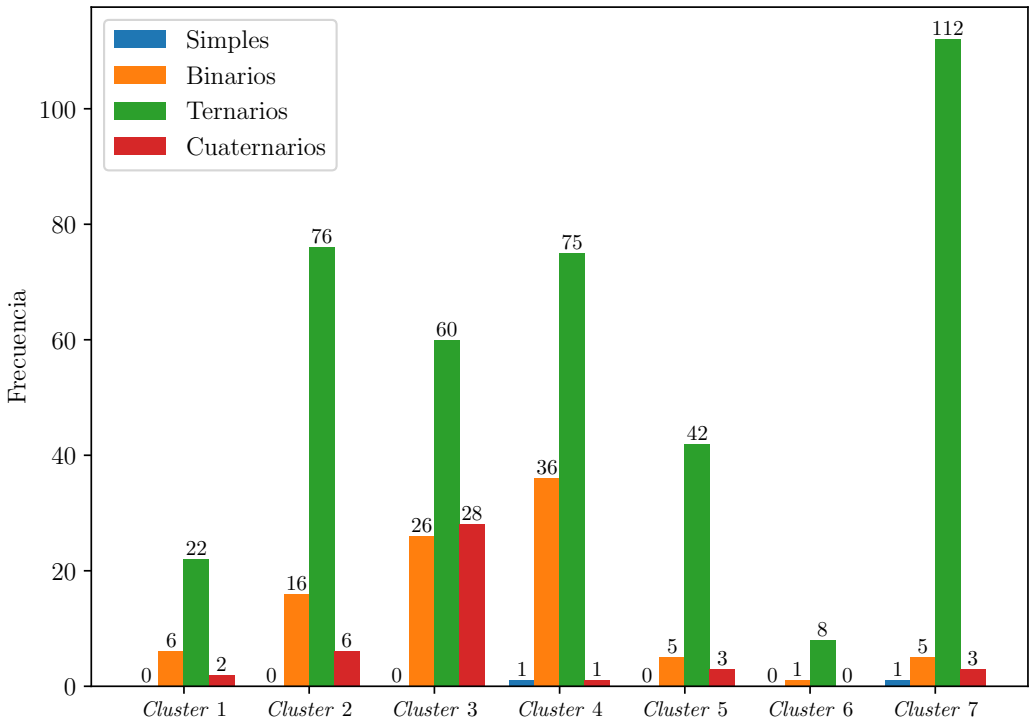


Figura 5.9: Tipos de compuestos por *cluster*

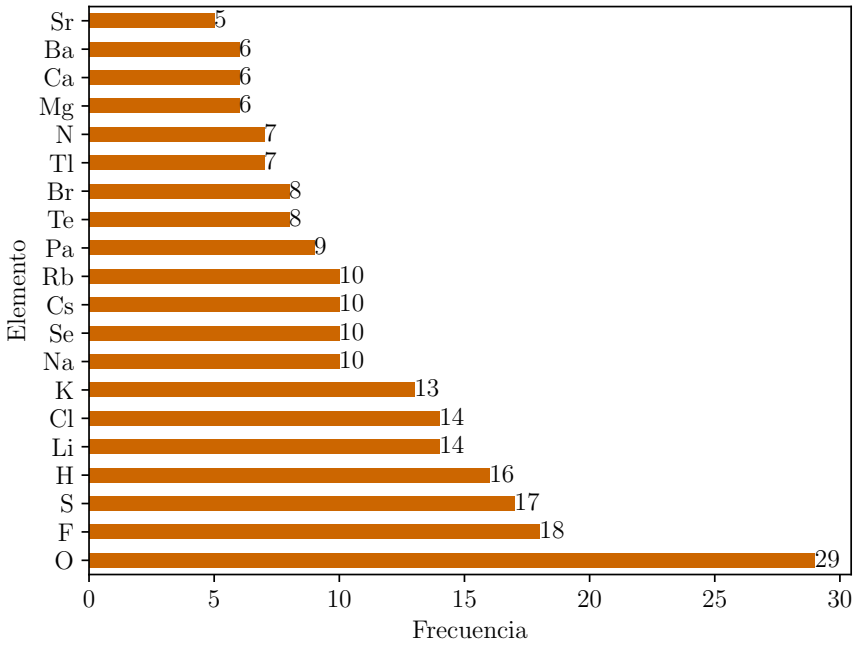


Figura 5.10: Frecuencia de elementos en el *cluster* 4

5.3. Materiales propuestos

A continuación, se presentan en la Tabla 5.3 los materiales del *cluster* 4 que ha sido filtrado para no contener elementos tóxicos o de tierras raras y se rescatan únicamente los de estructura cristalina cúbica y compuestos binarios. Estos materiales han sido seleccionados por sus propiedades idóneas para conformar una juntura efectiva con la Pirita, desempeñándose como capa ventana debido a su bajo coeficiente de absorción óptica y su elevado band gap.

La incorporación de estos materiales en las celdas solares busca optimizar la eficiencia en la captación de energía, mejorando el flujo de electrones y la estabilidad de la estructura frente a variaciones en las condiciones ambientales. Sin embargo, la evaluación detallada del rendimiento energético de estas celdas, al utilizar dichos materiales, permanece como un área de interés para futuras investigaciones, donde podrán explorarse tanto aspectos teóricos como experimentales para validar su potencial en aplicaciones prácticas.

Formule	Ele	BG	Structure	CA	cluster	Compuesto
LiH	H-Li	2.981	1	10195.55	4	Binario
Li2Se	Li-Se	2.9746	1	4373.009	4	Binario
RbH	H-Rb	2.9163	1	6317.384	4	Binario
MgS	Mg-S	2.7574	1	4272.297	4	Binario
SnF3	F-Sn	2.7195	1	7433.082	4	Binario
BeSe	Be-Se	2.6637	1	4356.541	4	Binario
MgSe	Mg-Se	2.5476	1	14204.63	4	Binario
TlCl	Cl-Tl	2.5369	1	11392.24	4	Binario
SrS	S-Sr	2.497	1	8017.256	4	Binario
Li2Te	Li-Te	2.4914	1	7401.305	4	Binario
Na2S	Na-S	2.4399	1	20758.96	4	Binario
CaS	Ca-S	2.3819	1	7292.231	4	Binario
CsH	Cs-H	2.3475	1	42008.82	4	Binario
K2S	K-S	2.3201	1	20112.42	4	Binario
TlBr	Br-Tl	2.2962	1	17298.44	4	Binario
SrSe	Se-Sr	2.2313	1	13825.00	4	Binario
TlI	I-Tl	2.2086	1	31587.38	4	Binario
BaS	Ba-S	2.1488	1	51310.03	4	Binario
K2Te	K-Te	2.141	1	72126.33	4	Binario
BaO	Ba-O	2.0906	1	51618.28	4	Binario
CaSe	Ca-Se	2.0728	1	11530.26	4	Binario
Na2Te	Na-Te	2.0282	1	62561.62	4	Binario
BeTe	Be-Te	2.0173	1	10109.30	4	Binario
Na2Se	Na-Se	2.0153	1	70067.02	4	Binario
Rb2S	Rb-S	1.9632	1	45433.19	4	Binario

Tabla 5.3: Materiales ventana propuestos

Capítulo 6

Conclusiones

6.1. Conclusiones generales

Las predicciones de agrupamiento obtenidas con el algoritmo de machine learning K-Means se realizaron considerando solo tres variables clave: el band gap, la estructura cristalina y el coeficiente de absorción óptica. Aunque estos parámetros ofrecen información relevante sobre el material, la selección precisa de materiales para formar junturas PN requiere un análisis más amplio que incluya otras propiedades físicas y químicas, como la conductividad, la movilidad de carga y la estabilidad térmica, entre otros.

A pesar de la limitación a solo tres variables, los agrupamientos generados en este proyecto mostraron características bien definidas. Esto facilitó una interpretación clara de los resultados y respaldó la selección preliminar de materiales prometedores. La coherencia observada en los agrupamientos indica que el uso de K-Means fue eficaz para identificar patrones significativos en los datos, incluso bajo un conjunto restringido de criterios.

Los resultados obtenidos en este estudio permiten afirmar la hipótesis planteada, ya que el uso del algoritmo de machine learning K-Means para analizar propiedades de materiales facilitó la identificación de opciones adecuadas para junturas PN. Aunque el análisis se basó en un conjunto limitado de criterios, el algoritmo logró identificar patrones esenciales que guiaron de manera efectiva el proceso de selección de materiales.

A pesar de los desafíos enfrentados, como la falta de una base de datos específica y la necesidad de crear una adaptada a los requisitos del proyecto, los objetivos planteados fueron alcanzados. Esto respalda la validez del enfoque de machine learning en este contexto, mostrando que incluso con un número limitado de variables de agrupamiento, se puede obtener una selección certera.

La metodología empleada en esta investigación demostró ser ade-

cuada para alcanzar los objetivos planteados, facilitando el análisis y selección de materiales. La implementación del algoritmo K-Means resultó efectiva para el agrupamiento de materiales, permitiendo la identificación de patrones y propiedades en la base de datos. Al utilizar una metodología cuantitativa e interpretativa, se pudo establecer un enfoque sólido que combina ciencia de datos y sistemas fotovoltaicos en una investigación exploratoria. Además, la elección del algoritmo K-Means en lugar del kNN se justificó adecuadamente, ya que logró clasificar los materiales en grupos sin la necesidad de etiquetas previas, abriendo posibilidades de investigación futura en la aplicación de algoritmos de agrupamiento en el área de sistemas fotovoltaicos.

6.2. Trabajos a futuro

El enfoque desarrollado en este estudio no solo facilita la selección de materiales con propiedades ópticas y estructurales óptimas, sino que también aporta una base sólida para mejorar la eficiencia de las celdas solares mediante criterios de selección basados en aspectos prácticos. A futuro, la validación experimental de estos materiales permitirá evaluar su rendimiento energético y su comportamiento en climas específicos, como el de la península de Yucatán, donde la alta demanda de energía ha impulsado el uso de sistemas solares más eficientes y accesibles. Esta demanda podría atenderse con la creación de celdas solares a base de Pirita, un material económico y abundante, en combinación con materiales ventana elegidos según los resultados de este estudio.

Además, el desarrollo de futuras investigaciones en esta línea podría profundizar en la optimización de los parámetros de búsqueda y los criterios de agrupamiento, explorando materiales adicionales que mejoren aún más la eficiencia y sostenibilidad de las celdas solares. A medida que esta metodología evolucione, se podrán identificar materiales ventana con propiedades cada vez más específicas, adaptados para maximizar el aprovechamiento de la intensa irradiación solar y resistir las altas temperaturas características de la península. Esto contribuirá a soluciones energéticas más eficientes, accesibles y ecológicas para la región.

Finalmente, es importante mencionar que la implementación de machine learning en el desarrollo de dispositivos fotovoltaicos ofrece un valor significativo, al facilitar la identificación de patrones y com-

patibilidades entre materiales semiconductores de una manera más eficiente que los métodos convencionales. Este enfoque no solo impulsa la innovación en celdas solares, sino que también es aplicable a otros dispositivos electrónicos, como diodos y transistores, donde los modelos de junturas PN optimizados mediante ML pueden mejorar tanto el rendimiento como la eficiencia.

En este estudio, el enfoque en junturas con Pirita ha permitido simplificar el análisis a nivel licenciatura. No obstante, este modelo es replicable y adaptable a otros materiales semiconductores, ya que el algoritmo y la base de datos pueden ajustarse a distintas combinaciones de materiales. Al entrenarse continuamente con datos adicionales, estos modelos de ML permiten mejorar progresivamente la precisión de las predicciones, adaptándose a las propiedades específicas de nuevos materiales a medida que se dispone de más información experimental.

Capítulo 7

Apéndices

7.1. Apéndice A: Códigos

En este apéndice se presentan los códigos resumidos empleados en el desarrollo del presente proyecto. No obstante, los códigos completos están disponibles para su descarga en el repositorio de GitHub[1].

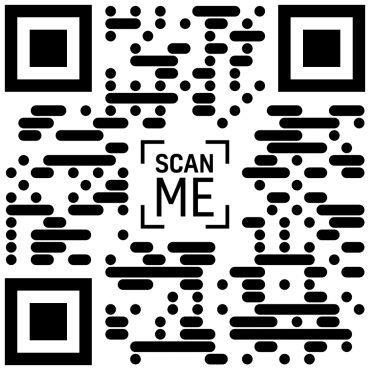


Figura 7.1: URL al GitHub

Obtención de *ids*

Código 7.1: Iterar sobre los *ids* desde cero hasta dos millones, accediendo a la base de datos de **The Materials Project** para realizar un *request* y verificar si el *id* iterado existe. Si el *id* se encuentra en la base de datos, se registra en un archivo .csv para su posterior tratamiento; de lo contrario, se omite.

```
### Se carga libreria m-api ###
from mp_api.client import MPRester

### Se declaran variables ###
key="API key"
mpr = MPRester(key)

### Se realiza iteracion ###
```

[1]Sitio web: https://github.com/PeterZH20/-Undergraduate-Thesis.git

```
with open('datos/ids.csv','w') as file:
  for i in range(2000001):
    docs = mpr.summary.search(material_ids=['mp-'+str(i)])
    if len(docs)!=0:
      file.write('mp-'+str(i)+'\n')
    i+=1
print('FINISHED')
```

Código 7.1: Barrido de ids existentes

Descarga de datos

Código 7.2: Itera sobre el archivo "`ids.csv`" generado por Código 7.1. Durante este proceso, se realizan los *requests* para cada *id* y, posteriormente, se almacena toda la información obtenida en un `DataFrame` .

```
### Se cargan librerias ###
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from mp_api.client import MPRester
key="API key"
mpr = MPRester(key)

### Se crea DataFrame ###
df=pd.DataFrame()

### Se crean listas ###
ids=[] #Lista para el ids
formule=[] #Lista para la formula quimica
NELE=[] #Lista para el numero total de elementos del compuesto
ELE=[] #Lista para los elementos del compuesto
BG=[] #Lista para el BandGap
IBGD=[] #Lista para el tipo de BandGap
IM=[] #Lista para saber si es metal
SC=[] #Lista para el tipo de estructura cristalina
a=[] #Lista para el parametro de red "a"
b=[] #Lista para el parametro de red "b"
c=[] #Lista para el parametro de red "c"
alfa=[] #Lista para el parametro de red "alpha"
beta=[] #Lista para el parametro de red "beta"
gamma=[] #Lista para el parametro de red "gamma"
CA=[] #Lista para el coeficiente de absorcion optica

### Se realiza requests ###
with open('datos/ids.csv','r') as file:
    for material in file:
        material=str(material.strip())
        ids.append(material)
```

```
34
35          docs = mpr.summary.search(material_ids=[material])
36          IM.append(docs[0].is_metal)
37          formule.append(docs[0].formula_pretty)
38          SC.append(str(list(docs[0].symmetry)[0][-1]))
39          IBGD.append(docs[0].is_gap_direct)
40          NELE.append(docs[0].nelements)
41          ELE.append(docs[0].chemsys)
42          BG.append(docs[0].band_gap)
43          a.append(docs[0].structure.lattice.abc[0])
44          b.append(docs[0].structure.lattice.abc[1])
45          c.append(docs[0].structure.lattice.abc[2])
46          alfa.append(docs[0].structure.lattice.angles[0])
47          beta.append(docs[0].structure.lattice.angles[1])
48          gamma.append(docs[0].structure.lattice.angles[2])
49
50          absorption = mpr.absorption.search(material_ids=[
               material], fields=["energies", "
               absorption_coefficient"])
51          if len(absorption)==0:
52              CA.append(np.nan)
53          else:
54              absorption=absorption
55              energia=absorption[0].energies
56              absorcion=absorption[0].absorption_coefficient
57              x=np.array(absorption[0].energies)
58              y=np.array(absorption[0].absorption_coefficient)
59              CA.append(np.interp(2.479647809,x,y))
60
61  ### Se estructura columanàs del DataFrame ###
62  df['id']=ids
63  df['Formule']=formule
64  df['No Ele']=NELE
65  df['Ele']=ELE
66  df['BG']=BG
67  df['IM']=IM
68  df['Is BG Dir']=IBGD
69  df['Structure']=SC
70  df['a']=a
71  df['b']=b
72  df['c']=c
73  df['alfa']=alfa
74  df['beta']=beta
75  df['gamma']=gamma
76  df['CA']=CA
77
78  ### Se exporta el DataFrame ###
79  df.to_csv('datos/datos_crudos.csv',index=False)
```

Código 7.2: Descarga y almacenamiento de datos

Filtro de datos

Código 7.3: Aplica filtros utilizando máscaras en el `DataFrame` que contiene los datos crudos. Este proceso permite reducir el conjunto de datos a aquellos materiales que cumplen con las siguientes condiciones: su valor de `IM` es igual a cero, el coeficiente de absorción óptica `CA` no es nulo y el valor de `BG` se encuentra entre 0.95 eV y 3 eV.

```
### Se carga libreria y se define###
import pandas as pd
file='datos/datos_crudos.csv'

### Se realizan filtros individuales para testeo###
df=pd.read_csv(file)
df=df.set_index('id')
print('Metales:',df[df['IM']==True].shape[0])
print('Metales con CA:',df[df['IM']==True].dropna().shape[0])
print('No Metales:',df[df['IM']==False].shape[0])
print('No Metales con CA:',df[df['IM']==False].dropna().shape[0
    ])
print('No Metales con CA y 0.95eV< BG <3eV:',df[(df['IM']==False
    )&((df['BG']>=0.95)&(df['BG']<=3))].dropna().shape[0])
print('Total de materiales:',
      df[df['IM']==True].shape[0]+
      df[df['IM']==False].shape[0])

### Se realiza filtrado en conjunto y se guardan los datos ###
df_clean=df[(df['IM']==False)&((df['BG']>=0.95)&(df['BG']<=3))].
    dropna()
df_clean.to_csv('datos/datos_clean.csv', index=False)
```

Código 7.3: Filtro de datos

Preparación de datos

Código 7.3: Realiza cambio de variable `str` a `int` para los valores `Structure` ; elimina las columnas innecesarias del `DataFrame` y se estandarizan los datos de `BG` , `Structure` y `CA` .

```
### Se cargan librerias ###
import pandas as pd
from sklearn import preprocessing
file='datos/datos_clean.csv'

### Se realiza el cambio de variable en Structure ###
df = pd.read_csv(file)
cambios={
```

```
    'Trigonal':5,
    'Cubic':1,
    'Tetragonal':2,
    'Hexagonal':4,
    'Orthorhombic':3,
    'Monoclinic':6
}
df_replace=df.replace(cambios)

### Se elimna columnas que no son de interes ###
df_replace=df_replace.drop(columns=['id','No Ele','IM','Is BG
   Dir','a','b','c','alfa','beta','gamma'])
df_replace

### Se seleccionan caracteristicas para llevar a cabo el
   agrupamiento ###
features = ['BG','Structure','CA']

### Se estandarizan los features ###
df_standardized = preprocessing.scale(df_replace[features])
X = df_standardized
```

Código 7.4: Preparación de datos

Optimización de hiperparámetros

Código 7.5: Optimiza los hiperparámetros para mediante `grid search` , evaluando diferentes combinaciones de parámetros. La declaración de `param_grid` incluye un rango de valores para cada hiperparámetro, y el proceso prueba todas las combinaciones posibles usando validación cruzada (5 particiones) para seleccionar la mejor configuración.

```
### Se cargan librerias ###
from sklearn.cluster import KMeans
from sklearn.model_selection import GridSearchCV
import numpy as np

### Se proponen valores de los parametros a iterar ###
param_grid = {
    'n_clusters': range(2, 31),
    'init': ['k-means++', 'random'],
    'n_init': [10,30,50],
    'max_iter': [100, 200, 300],
    'tol': [0.0001, 0.001, 0.01],
    'algorithm': ['lloyd', 'elkan'],
    'random_state': [0, 42, 84, 126]
}
```

```
16
17 ### Se crea el objeto K-Means ###
18 kmeans = KMeans(random_state=42)
19
20 ### Se declara grid search ###
21 grid_search = GridSearchCV(kmeans, param_grid=param_grid, cv=5,
       n_jobs=-1)
22
23 ### Se ajusta grid search con los datos estandarizados###
24 grid_search.fit(X)
25
26 ### Se imprimen resultados ###
27 print("Best hyperparameters: ", grid_search.best_params_)
```

Código 7.5: Optimización de parámetros para el K-Means

Método del codo y la silueta

Código 7.6: Calcula las distancias cuadradas dentro de cada *cluster* y el coeficiente de silueta (*Silhouette Coefficient*) para un rango de 30 posibles agrupamientos, con el objetivo de determinar el número óptimo de *clusters* en los datos filtrados.

```
1  ### Se cargan librerias ###
2  import numpy as np
3  from sklearn.cluster import KMeans
4  from sklearn.metrics import silhouette_score
5
6  ### Se declaran listas ###
7  wcss = []
8  sil_values = []
9
10 ### Se define rango dle numero de clusters ###
11 range_n_clusters = range(2, 32)
12
13 ### Se realiza la iteración de clusters con los parametros
       hallados ###
14 for i in range_n_clusters:
15     kmeans = KMeans(n_clusters = i,
16                     init=grid_search.best_params_['init'],
17                     max_iter=grid_search.best_params_['max_iter'],
18                     tol=grid_search.best_params_['tol'],
19                     algorithm=grid_search.best_params_['algorithm'],
20                     random_state=grid_search.best_params_['
                           random_state']
21                     )
22     kmeans.fit(X)
23     ### Se guardan las distancias cuadradas de cada cluster ###
24     wcss.append(kmeans.inertia_)
```

```
25      ### Se guardan el Silhoutte Coefficent de cada cluster ###
26      cluster_labels = kmeans.labels_
27      silhouette_avg = silhouette_score(X, cluster_labels)
28      sil_values.append(silhouette_avg)
29
30  ### Se halla a cual cluster le corresponde el Silhoutte
        Coefficent mas grande ###
31  sil_values = np.array( sil_values )
32  n_clusters_g = np.argmax(sil_values) + 2
33
34  ### Se imprimen resultados ###
35  print( 'El valor Silhoutte Coefficent máximo es:', sil_values[
        np.argmax(sil_values) ] )
36  print( 'El número óptimo de clusters es:', np.argmax(sil_values)
        + 2 )
```

Código 7.6: Elbow y Silhouette Method

Predicción *K*-Means con los datos filtrados

Código 7.7: Aplica el algoritmo *K*-Means para agrupar datos en el número total de *clusters* hallado anteriormente y utiliza los parámetros obtenidos por `grid_search.best_params_`. Luego, asigna el número de *clusters* a cada elemento en el `DataFrame` original para facilitar su identificación.

```
1   ### Se cargan librerias ###
2   from sklearn.cluster import KMeans
3
4   ### Se realiza prediccion de etiquetas con los parametros
        hallados ###
5   kmeans = KMeans(n_clusters = n_clusters_g,
6               init=grid_search.best_params_['init'],
7               max_iter=grid_search.best_params_['max_iter'],
8               tol=grid_search.best_params_['tol'],
9               algorithm=grid_search.best_params_['algorithm'],
10              random_state=grid_search.best_params_['random_state'
                    ])
11  y_kmeans = kmeans.fit_predict(X)
12
13  ### Se asigna el numero de cluster a cada elemento del DataFrame
        original ###
14  df_replace['cluster'] = y_kmeans + 1
15  df_replace.head()
```

Código 7.7: Algoritmo K-means

7.2. Apéndice B: Base de datos

Debido al volumen de datos, no es posible incluir toda la información en este apéndice. No obstante, las bases de datos están disponibles para su descarga o visualización en el repositorio de GitHub[2], en la carpeta `datos`. El código QR que enlaza al URL es el mismo que se muestra en la Figura 7.1.

En el GitHub se puede hallar los siguientes archivos:

a) `ids.csv`: *ids* de los materiales disponibles en **The Materials Project** .

b) `datos_crudos.csv`: Base de datos con los 143,666 materiales descargados.

c) `datos_clean.csv`: Base de datos con los 535 materiales filtrados que poseen `CA` y `BG` entre 0.95 y 3 eV.

d) `datos_replace.csv`: Base de datos `datos_clean.csv` modificada con la presencia únicamente de las columnas `features`.

e) `datos_cluster.csv`: Base de datos `datos_clean.csv` modificada con el anexo de los *clusters* .

f) `datos_final.csv`: Base de datos con los materiales finales propuestos.

[2]Sitio web: https://github.com/PeterZH20/-Undergraduate-Thesis/tree/main/datos

Capítulo 8

Referencias

Ali, M. O. (1993). *Elementary solid state physics: principles and applications*. Addison-Wesley publishing company.

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Amazon Web Services. (2021). *¿qué es una api?* https://aws.amazon.com/es/what-is/api/. (Recuperado el 2024-10-06)

Arencibia-Carballo, G. (2016). La importancia del uso de paneles solares en la generación de energía eléctrica. *REDVET. Revista Electrónica de Veterinaria*, *17*(9), 1–4.

Cui, M., y cols. (2020). Introduction to the k-means clustering algorithm based on the elbow method. *Accounting, Auditing and Finance*, *1*(1), 5–8.

Department of Earth Sciences, University of Minnesota. (2021). *Pyrite*. https://commonminerals.esci.umn.edu/minerals-o-s/pyrite. (Recuperado el 2024-11-07)

Dhimish, M. (2021). Defining the best-fit machine learning classifier to early diagnose photovoltaic solar cells hot-spots. *Case Studies in Thermal Engineering*, *25*, 100980.

García, J., Molina, J., Berlanga, A., Patricio, M., Bustamante, A., y Padilla, W. (2018). Ciencia de datos. *Técnicas Analíticas y Aprendizaje Estadístico. Bogotá, Colombia. Publicaciones Altaria, SL*.

GSU. (2018). *El dopado de semiconductores*. http://hyperphysics.phy-astr.gsu.edu/hbasees/Solids/dope.html#c3. (Recuperado el 2023-10-18)

Hsieh, P.-C., y Tung, P.-C. (2009). A novel hybrid approach based on sub-pattern technique and whitened pca for face recognition. *Pattern Recognition*, *42*(5), 978–984.

Jain, A., y Ong, S. (2013). Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, *1*(1).

Karande, P., Gallagher, B., y Han, T. Y.-J. (2022). A strategic approach to machine learning for material science: How to tackle real-world challenges and avoid pitfalls. *Chemistry of Materials*, *34*(17), 7650–7665.

Kittel, C. (2012). *Introducción a la física del estado sólido*. Reverté.

Lu, Z., Zhou, H., Ye, C., Chen, S., Ning, J., Halim, M. A., … Wang, S. (2021). Fabrication of iron pyrite thin films and photovoltaic devices by sulfurization in electrodeposition method. *Nanomaterials*, *11*(11), 2844.

Materials Project. (2019). *Next-generation materials data platform*. https://next-gen.materialsproject.org/. (Recuperado el 2024-10-06)

Mertens, K. (2018). *Photovoltaics: fundamentals, technology, and practice*. John Wiley & Sons.

Mirjalili, V., y Raschka, S. (2020). *Python machine learning*. Marcombo.

Mitchell, M., y cols. (2019). Artificial intelligence: A guide for thinking humans.

NREL. (2023). *Best research-cell efficiency chart*. https://www.nrel.gov/pv/cell-efficiency.html. (Recuperado el 2023-09-27)

Odabaşı, Ç., y Yıldırım, R. (2020). Machine learning analysis on stability of perovskite solar cells. *Solar Energy Materials and Solar Cells*, *205*, 110284.

ONG. (2020). *Calentamiento global*. https://www.manosunidas.org/observatorio/cambio-climatico/calentamiento-global. (Recuperado el 2023-09-27)

Pivetta, M. (2020). *Big data de materiales*. https://revistapesquisa.fapesp.br/es/big-data-de-materiales/. (Recuperado el 2023-09-28)

PVEducation. (2019). *A collection of resources for the photovoltaic educator or student*. https://www.pveducation.org/es. (Recuperado el 2023-10-18)

Rodríguez, S. (2022). *Historia del panel solar: ¿cómo nació y cuál ha sido su evolución?* https://solfy.net/placas-solares/historia-del-panel-solar/. (Recuperado el 2023-09-27)

Rosas, G. (2013). *Estados inestables de los materiales bganp/gap y bganas/gaas*. Tesis de Licenciatura, Instituto Politecnico Nacional. Ciudad de Mexico, Mexico. Descargado de https://tesis.ipn.mx/jspui/bitstream/123456789/15047/1/I.C.%2009-13.pdf (Recuperado el 2024-11-08)

Rouhiainen, L. (2018). Inteligencia artificial. *Madrid: Alienta Editorial*, 20-21.

Russell, S. J., y Norvig, P. (2016). *Artificial intelligence: a modern approach*. Pearson.

Saabith, A. S., Vinothraj, T., y Fareez, M. (2020). Popular python libraries and their application domains. *International Journal of Advance Engineering and Research Development*, *7*(11).

Schmidt, M., y McIntyre, P. C. (2008). Pyrite fes for thin-film photovoltaics: A re-assessment. *Journal of Renewable and Sustainable Energy*, *2*(3), 1–7.

Scikit-learn Developers. (2024a). K-means silhouette analysis example — scikit-learn 1.5 documentation [Manual de software informático]. Descargado de https://scikit-learn.org/1.5/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Scikit-learn Developers. (2024b). sklearn.cluster.kmeans — scikit-learn 1.5 documentation [Manual de software informático]. Descargado de https://scikit-learn.org/1.5/modules/generated/sklearn.cluster.KMeans.html

Son, P. (2019). *Acelerar la innovación en la ciencia de los materiales con el aprendizaje automático*. https://www.cas.org/es-es/resources/blog/acelerar-la-innovacion-en-la-ciencia-de-los-materiales-con-el-aprendizaje-automatico. (Recuperado el 2023-10-11)

Ulaczyk, J., Morawiec, K., Zabierowski, P., Drobiazg, T., y Barreau, N. (2017). Finding relevant parameters for the thin-film photovoltaic cells production process with the application of data mining methods. *Molecular informatics*, *36*(9), 1600161.

Unpingco, J. (2021). *Python programming for data analysis*. Springer.

W3Schools. (2018). *Python tutorial*. https://www.w3schools.com/python/python_intro.asp. (Recuperado el 2024-10-06)

Yosipof, A., Nahum, O. E., Anderson, A. Y., Barad, H.-N., Zaban, A., y Senderowitz, H. (2015). Data mining and machine learning tools for combinatorial material science of all-oxide photovoltaic cells. *Molecular informatics*, *34*(6-7), 367–379.