



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Diseño e implementación de un  
programa de escritorio para  
indexar patrones de difracción  
obtenidos en un microscopio  
electrónico de transmisión**

**TESIS**

Que para obtener el título de

**Ingeniero en Computación**

**P R E S E N T A**

Ulysses López López

**DIRECTOR DE TESIS**

Dr. Raúl Herrera Becerra



Ciudad Universitaria, Cd. Mx., 2024

## Índice

ÍNDICE.....	1
<i>Contenido</i> .....	2
CAPÍTULO I.....	4
<i>Introducción</i> .....	4
CAPÍTULO II.....	6
<i>Planteamiento del problema</i> .....	7
<i>Objetivo</i> .....	8
<i>Metodología</i> .....	9
<i>Contribución a la comunidad</i> .....	10
<i>Estructura de la Tesis</i> .....	11
CAPÍTULO III.....	12
<i>Antecedentes</i> .....	12
<i>Cristales</i> .....	13
<i>Cristalografía</i> .....	14
Grupos Cristalográficos y Redes de Bravais.....	14
Índices de Miller.....	20
<i>Difracción de Rayos X y HTEM</i> .....	23
CAPÍTULO IV.....	31
<i>Indexación de patrones de difracción</i> .....	31
Algoritmo.....	31
<i>Diagramas de flujo</i> .....	36
CAPÍTULO V.....	45
<i>Desarrollo e implementación del programa para indexar patrones de difracción obtenidos a partir de un microscopio electrónico de transmisión</i> .....	45
Inserción de campos e indexación.....	45
Cálculos y ordenamiento.....	46
Sistema de carga y manejo de errores.....	46
CAPÍTULO VI.....	47
<i>Resultados (Ejemplo)</i> .....	47
CAPÍTULO VII.....	56
<i>Conclusiones</i> .....	56
ANEXO I.....	59
<i>Manual de uso</i> .....	59
<i>Requerimientos</i> .....	59
<i>INSTALACIÓN</i> .....	59
Instalación de WinRAR.....	60
Instalación de Runtime de MATLAB.....	62
Descarga del programa "PIPD-II".....	66
<i>Contenido de los Archivos ".xls"</i> .....	68
Archivo de salida.....	74
ANEXO II.....	80
ANEXO III.....	81
Apéndice de código.....	81

Código "obtenciondatos.m" .....	81
Código "Compara.m" .....	83
Código "Calculateta13.m" .....	89
Código "Calculad2.m" .....	96
Código "Calculateta12.m" .....	100
Código "EjeZona.m" .....	106
Código "escribe.m" .....	108
Código "Extincion.m" .....	110
Código "main.m" .....	112
Código "PIPD2.m" .....	118
BIBLIOGRAFÍA.....	121

## CAPÍTULO I

### Introducción

A lo largo de la historia, la resolución de diversas incógnitas y actividades se ha logrado gracias a la invención y uso de nuevas tecnologías, lo que ha facilitado el estudio e investigación en diversas áreas del conocimiento. Un buen ejemplo de ello lo encontramos en la década de los años 30's del siglo XX, con los estudios de Turing y su propuesta de la primera máquina para hacer una infinidad de cálculos en unos pocos minutos. La máquina de Turing precursor de las ahora llamadas computadoras, ha venido a mejorar muchos procesos que anteriormente consumían grandes cantidades tiempo y por consiguiente eran sujetas a muchos posibles errores humanos y, con la constante evolución de las computadoras, el tiempo de procesamiento se ha reducido exponencialmente y con ello entre otros, al avance de la ciencia y la tecnología en la resolución de los diferentes problemas que enfrentan científicos y tecnólogos, con lo que ahora se puede pensar en problemas y soluciones más complejas y que tienen un mayor alcance.

En la actualidad es necesario procesar grandes cantidades de datos y realizar un gran número cálculos complejos. El avance en las Ciencias de la Computación y la computadora misma y sus componentes, han sido fundamentales para abordar estos desafíos. Además, las computadoras han permitido el desarrollo de modelos y nos han permitido ejecutar simulaciones que antes no eran posibles, contribuyendo así a comprender mejor algunos fenómenos complejos en áreas como la física, la química, la biología, la medicina y muchas otras.

Un ejemplo muy particular de estos fenómenos que atañen a las llamadas Ciencias exactas o duras es el apasionante mundo de la Materia Condensada donde se estudian los materiales sólidos y líquidos. Y esta es la razón por lo que, en este trabajo, nos concentraremos en aquellos materiales sólidos conocidos como, *cristales*.

Se conocen como materiales cristalinos aquellos materiales sólidos cuyos elementos constitutivos (átomos, iones o moléculas) se repiten de manera ordenada y paralela y cuya distribución en el espacio muestra ciertas relaciones de simetría. Es decir, la propiedad característica de un medio

cristalino es ser periódico, es decir, que a lo largo de cualquier dirección la materia que lo forma se halla a distancias específicas y paralelamente ordenada [2].

Una de las herramientas más comúnmente utilizadas para el estudio y caracterización de estos materiales, es el microscopio electrónico de alta resolución (HTEM por sus siglas en inglés). Este instrumento que nos permite en la actualidad observar imágenes con una resolución cercana a unos cuantos *Ángstroms*, también nos permite, como otros instrumentos ópticos, observar el patrón de difracción que genera nuestro cristal. Recordemos que el patrón de difracción es como una huella digital que nos deja la interacción de la luz con la estructura del objeto que lo genera, y que no existen dos materiales diferentes que generen exactamente el mismo patrón de difracción. De lo anterior, entonces una forma de estudiar, caracterizar o analizar un cristal, es formando su patrón de difracción para identificarlo y asociarlo a un cierto material.

Para identificar un patrón de difracción habría que entender la teoría física que nos explica el origen de estos patrones, la cual, dado que no es tema de este trabajo, dejaremos a un lado. Para este trabajo nos limitaremos a decir que un patrón de difracción está compuesto por una serie de puntos ordenadamente dispuestos en el espacio. Cada punto tiene su origen en alguna parte de la estructura de nuestro cristal, y dado que los cristales están ordenados en todas direcciones, entonces podemos pensar un cristal como un conjunto de planos paralelos entre ellos separados por una distancia característica de cada material. Por lo tanto, cada punto de nuestro patrón de difracción representa a algunos de esos planos que estamos observando [2].

La tarea entonces es identificar al menos tres de los planos que se presentan en nuestro patrón de difracción y asociarlos a un material en particular, comparándolo con la gran cantidad de información que se puede encontrar en libros, tablas de rayos x, etc. Existen varias formas de realizar esta tarea, sin embargo, el proceso generalmente es lento y tedioso, ya que no existe una solución única a este problema, por lo tanto, lo que se obtiene es una serie de buenas soluciones. Minimizar esta tarea es el objetivo, para lo cual se utiliza como herramienta las ciencias de la computación.

---

El desarrollo de las Ciencias en Computación y por ende de las computadoras, ha beneficiado enormemente a la microscopía electrónica. Hace 50 años era más caro obtener la infraestructura de cómputo necesaria para resolver estos problemas que cubrir el costo mismo de un microscopio electrónico. En la actualidad la situación se ha invertido y ahora el costo de una computadora potente que cubra las necesidades de cómputo para ejecutar los programas que se requieren, tiene un costo apenas del 1% del costo de un nuevo y moderno microscopio [12]. Hoy en día con el apoyo de las Ciencias de la Computación es posible llevar a cabo entre otros: simulación de imágenes, procesamiento de imágenes, diseño de instrumentos, el trabajo en tiempo real de un instrumento, almacenamiento masivo de información, etc. Todo ello en beneficio de la técnica de microscopía electrónica.

Un libro especializado en el tema es el de Andrews [1], aquí se puede encontrar toda la teoría al respecto, así como una amplia explicación y secuencia de los pasos a seguir para interpretar los patrones de difracción. Dado que matemáticamente no existe una solución única, a partir de 1975 existen varios trabajos publicados proponiendo formas de resolver el problema de indexar estos patrones de difracción [3-7], es decir, asociar cada punto en el patrón de difracción a un plano del cristal, la mayoría de ellos basado en el contenido del libro de Andrews, lo mismo que hacemos nosotros en este trabajo.

Por lo anterior, en este trabajo se presenta una propuesta para resolver el problema de indexar los patrones de difracción obtenidos directamente con el HTEM o a través de la FFT (transformada rápida de Fourier) de una zona selecta de una imagen obtenida con el mismo instrumento. El trabajo lo presentamos organizado partiendo en la introducción, puntualizaremos cuales son nuestro objetivo principal y cuáles son las metas que deberíamos cumplir, discutiremos brevemente la metodología para realizar este trabajo, haremos una rápida presentación de la Teoría, describiremos el programa que proponemos, presentaremos algunos resultados obtenidos con ejemplos reales, presentaremos nuestras conclusiones, enlistaremos la bibliografía utilizada y como anexo, propondremos un manual de operación del programa dirigido aquellos que quieran utilizar dicho programa, los códigos del programa mismo, y las ecuaciones que se utilizan en los cálculos.

## CAPÍTULO II

### Planteamiento del problema

El estudio de la materia condensada ha estado presente a lo largo de las investigaciones desde hace muchos años, en particular, las investigaciones realizadas por parte de las instituciones de la UNAM no son una excepción, y su estudio ha estado presente a lo largo de los años. Sin embargo, muchos de los cálculos y decisiones a tomar se han tenido que realizar con cálculos efectuados de manera manual e interpretados con base en el criterio de cada investigador, lo que ha originado ciertas diferencias y demoras en el avance al estudiar la materia condensada. En la actualidad, en el Instituto de Física existe un grupo que cuenta con un programa que se limita a solo tres de las siete posibles estructuras cristalinas que se presentan [7] y requiere mejoras en varios aspectos. Es por eso que, el programa que proponemos en este trabajo, tiene como objetivo disminuir el trabajo de cálculo y de toma de decisiones, presentando las mejores soluciones por cada patrón indexado de cada elemento que se presenta, además de contar con mejoras, como poder indexar resultados con alguna de las 7 estructuras cristalinas existentes, o como poderlo instalar en cualquier computadora (en la actualidad está fijo en una sola máquina), y en general, hacer que este programa sea universal y amigable para los usuarios, con una mayor eficiencia en la obtención de resultados en problemas cotidianos en materia condensada y en cristalografía.

## Objetivo

Diseñar e implementar una aplicación de escritorio, amigable, que, por medio del ciclo de desarrollo de software, proporcione posibles soluciones matemáticas basadas en la teoría, que ayuden, en una forma rápida y simple, a indexar patrones de difracción obtenidos mediante imágenes generadas en un microscopio electrónico de transmisión de alta resolución.

Para lograr el objetivo de este trabajo, entendemos que será necesario plantearnos las siguientes metas:

- i) Preguntarnos quienes serán los usuarios de un programa como éste y cuáles son sus necesidades.
- ii) Seleccionar el lenguaje de programación más adecuado considerando que el programa deberá poder ser actualizado por los usuarios en algún momento futuro.
- iii) Cuidar que el programa pueda ser ejecutado en cualquier máquina de escritorio.
- iv) Dejar un manual de apoyo al usuario, tomando en cuenta que el posible usuario no entiende de programación a pesar de que sea un experto en el estudio de cristales.

Por lo anterior, y después de comentar ampliamente el problema a resolver, entendimos que para lograr el alcance que debe tener el programa, tenemos que considerar entre otros aspectos:

- Que se está dando un apoyo a un grupo de académicos (estudiantes, técnicos e investigadores) de áreas diversas como la física, química e ingeniería, para llevar a cabo en forma rápida y segura, el procedimiento matemático para la indexación de los patrones de difracción, proponiendo las mejores soluciones para cada caso.
- Al ser un programa portable, no se limita a una máquina fija en particular, sino que cualquier usuario podrá tener el programa en su computadora de trabajo, y así tener la posibilidad de realizar con mayor rapidez sus trabajos, ya que, en la actualidad, el grupo con el que llevamos a cabo este trabajo cuenta con un primer programa, limitado en comparación con el que estamos desarrollando ahora, el cual, está instalado en una sola máquina.
- Ya que los avances en este campo no se detienen, se planea que en el futuro pueda ser enriquecido este programa. Es importante que todo usuario pueda contar con él y a su vez

enriquecer, en su propio beneficio, uno o varios de los archivos de apoyo que se requieren para que este programa funcione adecuadamente.

## **Metodología**

El camino óptimo para poder realizar el proyecto se definió en tres etapas importantes, a decir:

1.- Investigación de antecedentes: Debido al área tan compleja sobre el estudio de materiales y en específico sobre cristales y la difracción ya sea de rayos X, electrones o neutrones, se tendrá durante la realización del proyecto que investigar sobre la cristalografía y sobre la obtención de los datos que a los investigadores desean estudiar para así obtener las mejores soluciones mediante el programa.

2.- Desarrollo del programa: Con la debida investigación previa, se comienza a desarrollar el programa respetando las fases del ciclo de vida de software.

En primera instancia el análisis del problema y la toma de requerimientos fueron llevadas a cabo en cierta medida en la investigación de antecedentes, sin embargo, aún se requiere de mayor profundización para poder llevar a cabo el proyecto, es por eso por lo que se hicieron diversas preguntas referentes a cómo los investigadores podrían manejar el programa y cuáles son sus necesidades reales al tener un programa que les brinde un apoyo enorme para la indexación de patrones de difracción.

Tomando en cuenta el problema y sus requerimientos, se procede con el diseño de diversas partes del programa para realizar la indexación de los patrones de difracción, para lo cual se requiere de diversos pasos y de diversas variables a considerar, además de que deben de cumplir ciertas normas los datos que se deben de ingresar para que el programa proceda con su correcto funcionamiento.

Una vez teniendo el diseño de los diagramas de flujo y de su correcto orden, además de tener los bosquejos de las pantallas de la interfaz de usuario, se procede a codificar los diagramas de flujo en Matlab, ya que para los potenciales usuarios resulta ser el mejor lenguaje para poder realizar cálculos por su fácil manejo de matrices de manera eficiente, para obtener en primera instancia, una aplicación de consola en la cual se procede a verificar su correcto funcionamiento, además de tener

una mejor perspectiva sobre las entradas y salidas del programa, así como poder mejorar el código y ver los fallos en los diagramas de flujo.

Terminada la fase de codificación y de validar las entradas y salidas, así como el correcto funcionamiento del programa de consola, se procede al desarrollo de la interfaz gráfica de usuario con los bosquejos que previamente se tenían además de modificarlos con las pruebas realizadas en la aplicación de consola para hacer mucho más amigable y fácil de usar para los investigadores. Esta parte de la interfaz gráfica de usuario se realizará en Java ya que es la que brinda mejor soporte en temas de permisos y de realizar una interfaz de usuario más apegada al correcto funcionamiento y ensamblaje con Matlab.

3.- Pruebas y resultados: Para finalizar se realizarán pruebas con los siete distintos tipos de estructuras que existen en la cristalografía para tener una mayor cobertura en cuanto a los datos a procesar ya que en el estudio de la materia, existen distintos componentes en los cuáles, cada usuario se enfocará en los que se encuentra estudiando. Además de tener un programa estable, se tendrá el manual de usuario disponible para que los investigadores sepan de su correcto funcionamiento.

### **Contribución a la comunidad**

El proyecto contribuye significativamente debido al alcance que tiene el programa.

En primer lugar, al dar apoyo a los investigadores en el procedimiento matemático para la indexación y de obtener resultados en poco tiempo válidos matemáticamente. Además, al ser un programa portable, no se limita a una máquina en particular, sino que los investigadores podrán tener el programa en su computadora de trabajo, por lo que tendrán la posibilidad de realizar con mayor rapidez sus investigaciones, además de no esperar a la disponibilidad de una máquina que contenga el programa. Y finalmente, contar con archivos de Excel que funcionan para la recopilación de datos, ayuda a que los investigadores puedan compartir con mayor facilidad y eficacia los resultados obtenidos y las tablas de rayos X para futuras investigaciones.

Tiene a su vez una contribución que pocos usuarios del HTEM aceptan como válida. Como cualquier equipo científico, el microscopio electrónico cuenta con una estructura sumamente complicada la cual está sujeta a cambios, desperfectos e imprecisiones. Matemáticamente se supone que el

microscopio está perfectamente alineado y todas las medidas que, de acuerdo con la teoría de óptica geométrica, deben de hacerse respecto a su eje óptico, con correctas. Difícilmente podríamos asegurar que en la realidad dicha aseveración es totalmente cierta, por lo que en general, los resultados no siempre son los que el usuario esperaría y, por consiguiente, no son coincidentes con los resultados matemáticos que se pueden obtener de acuerdo con la teoría. Este programa tiene, por consiguiente, el valor de avisarle al usuario comprometido con la técnica de la microscopía electrónica, que su aparato requiere de ciertos ajustes, los cuales, en ocasiones ya son imposibles de realizar a la perfección, lo cual lo obliga a considerar otras formas de compensar algunos errores.

### **Estructura de la Tesis**

El presente proyecto cuenta con la siguiente estructura:

- El capítulo 1 presenta la introducción a la tesis.
- El capítulo 2 contiene el planteamiento del problema, los objetivos de la tesis, así como la metodología a seguir, la contribución a la comunidad y finalmente cómo está estructurada la tesis.
- El capítulo 3 presenta las bases teóricas que se deben de tener para el desarrollo del programa, desde los fundamentos de la cristalografía, los fundamentos de la difracción de electrones, las estructuras que presentan los materiales y finalmente la tabla para la determinación de espacios interplanares de las estructuras cristalinas.
- El capítulo 4 presenta el diseño del algoritmo para la indexación de los patrones de difracción, así como las consideraciones a tomar para que la indexación que se realizará en el programa sea la más eficiente y proporcione los mejores resultados.
- El capítulo 5 presenta tres partes fundamentales del programa, la primera parte, la inserción de campos e indexación, la segunda es la parte de los cálculos y el ordenamiento de éstos, la tercera parte es el sistema de carga y el manejo de errores
- El capítulo 6 presenta los resultados del proyecto.
- El capítulo 7 presenta la conclusión tras el desarrollo del proyecto.
- Anexos
- Bibliografía

## CAPÍTULO III

### Antecedentes

La teoría que aquí se aplica es conocida y se encuentra ampliamente discutida en muchas referencias, por lo que nosotros nos limitaremos a hacer en este escrito, un pequeño resumen de los aspectos que consideramos más importantes y que nos apoyaron en la realización de este trabajo. Lo que mostraremos a continuación lo resumimos básicamente de dos libros que son muy útiles en esta área, a saber [1] y [2].

En 1977 el premio Nobel en física Philip Anderson intentó definir un sólido como: “una red regular de átomos que tienen, en buena aproximación, y que tienen una simetría traslacional bajo algunos de los grupos espaciales”. Esta es sin duda una de las mejores definiciones de esos sólidos que conocemos como *sólidos cristalinos o cristales*. En términos muy generales se llama Física del Estado Sólido a la parte de la física que estudia los cristales y sus fases.

Las propiedades físicas de los sólidos se deben a sus estructuras a nivel atómico. Tomemos como ejemplo dos sólidos parecidos, el cuarzo (en su forma cristalina) y el conocido como vidrio o también, vidrio de ventana, cuyo elemento principal en su composición en ambos casos es el dióxido de silicio o sílice,  $\text{SiO}_2$ , donde los átomos de Si y de O están en proporción 1:2. Ahora bien, en el cristal los átomos se disponen ordenadamente de forma que, si se conoce la posición de un átomo de Si (o de O), se puede conocer la posición de todos los demás. Se dice entonces que existe invariancia traslacional, que dota al cristal de un tipo de orden llamado de *largo alcance*. En el vidrio, los átomos de Si y de O (también en proporción 1:2) no se disponen de forma regular, por lo tanto, dada la posición de un átomo (de Si o de O) sólo es posible indicar aproximadamente cuántos átomos y de qué tipo están próximos a él, pero la predicción no se puede extender a todo el volumen [2,13].

Es decir, existe una diferencia clara entre lo que se entiende por un cristal y por un vidrio, siendo el primero de estos, lo que representa el tema principal de nuestro trabajo.

Con este ejemplo podemos entonces decir que un cristal tiene una estructura tal, que si se tiene una red regular que se extiende ordenadamente en las tres dimensiones, veremos que en cada punto de la red se encuentra un átomo, los cuales pueden ser de un mismo elemento o de varios. A esta red vista en dos dimensiones se le llama, *base* [2].

Podemos decir entonces que un cristal tiene una estructura tal que si se tiene una red regular que se extiende en las tres dimensiones, veremos que a cada punto de la red se encuentra un átomo, los cuales pueden ser de un mismo elemento o de varios. A esta red vista en dos dimensiones se le llama *base*.

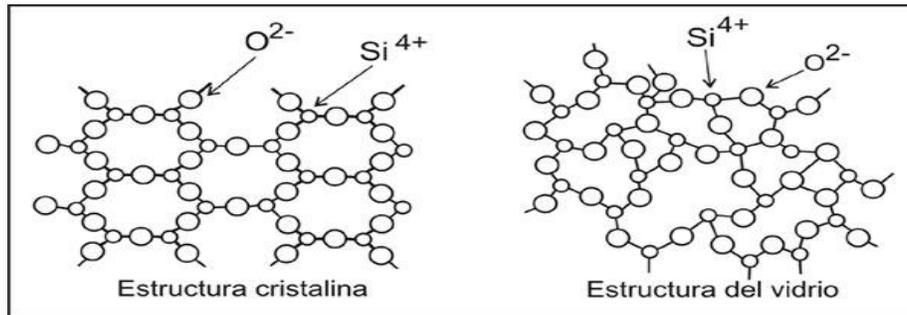


Figura 1. Diferencia de estructura entre un cuarzo y un vidrio.

## Cristales

Los cristales son sólidos con una estructura ordenada y repetitiva a nivel atómico lo que les confiere propiedades físicas y químicas particulares. Un cristal se define por su estructura tridimensional, en la cual los átomos, iones o moléculas se organizan en una red regular y periódica. Este arreglo ordenado da lugar a caras planas y ángulos precisos entre ellas, creando una forma definida y geométrica. “El carácter geométrico sumamente regular de los cristales y la forma en que éstos crecen sugiere que los átomos dentro de ellos están bien ordenados y arreglados simétricamente.” [1,2,12]

Esto quiere decir que los cristales ideales tienen algunas características únicas, que es necesario comprender para poder entender más adelante otros conceptos fundamentales, como las siguientes:

- 1.- Es infinito, se propaga sin fin en todas las direcciones del espacio, manteniendo un patrón de gran alcance.
- 2.- Es químicamente constante, cuando se refiere a un elemento en estado sólido, esto implica su pureza absoluta; en el caso de un compuesto, denota una estequiometría completamente precisa.
- 3.- Es estructuralmente uniforme, cada átomo ocupa las posiciones determinadas por las operaciones de simetría, sin presencia de átomos en ubicaciones no apropiadas ni ausencia de ninguno.

Los cristales se destacan por su transparencia, dureza y capacidad de reflejar la luz de manera característica. Su estructura interna determina las propiedades físicas y químicas del material, y a través de la cristalografía, se revela un mundo invisible de la estructura atómica que nos ayuda a comprender la naturaleza de la materia.

La simetría es una propiedad fundamental de los cristales ya que, como se mencionó anteriormente, para ser considerado cristal, una de sus características es que es una estructura ordenada y repetitiva a nivel atómico, por lo que, mediante operaciones de simetría, les permite alinearse con su propia estructura como resultado de ciertas transformaciones geométricas llamadas *operaciones de simetría*, de las cuáles, las más comunes son isométricas y se ven limitadas a la rotación, reflexión y a la traslación.

### **Cristalografía**

La cristalografía es una disciplina científica que se enfoca en el estudio de la estructura atómica y molecular de los cristales. La cristalografía nos brinda una visión detallada de la simetría, la disposición espacial y las interacciones en la estructura cristalina, lo que nos permite comprender mejor las propiedades físicas y químicas de los materiales. La cristalografía tiene una amplia gama de aplicaciones en diversos campos científicos y tecnológicos. En la química, la cristalografía permite comprender las estructuras moleculares y las interacciones entre los átomos, lo que es fundamental para el diseño de nuevos compuestos químicos y fármacos. En la física de materiales, la cristalografía es esencial para el desarrollo de materiales con propiedades específicas, como metales más resistentes o materiales semiconductores para dispositivos electrónicos.

### ***Grupos Cristalográficos y Redes de Bravais***

La mayoría de los sólidos son de naturaleza policristalina, lo que implica que están formados por numerosos cristales pequeños que se disponen de manera compacta y con orientaciones aleatorias. En condiciones específicas de temperatura y presión, es posible provocar una reorganización ordenada de los átomos de estos cristales pequeños, dando lugar a la formación de un único monocristal.

La diferencia entre los diversos tipos de sólidos radica en el orden casi perfecto de los átomos que los constituyen. Este orden proviene de una unidad estructural fundamental que se repite periódicamente en todo el volumen del monocristal.

En cada cristal, podemos identificar unidades estructurales equivalentes que se caracterizan por ciertos parámetros. Estas unidades, al repetirse sin superponerse, recrearían exactamente el cristal en su totalidad.

La clasificación de los sólidos cristalinos se basa en estas unidades, y junto con otros elementos, nos permite categorizarlos en uno de los 230 grupos espaciales cristalográficos existentes.

Estas unidades estructurales son denominadas celdas unitarias, y se les llama celdas unitarias primitivas si encierran el volumen más pequeño definible dentro de la red o arreglo espacial de puntos.

Los parámetros utilizados para definir una celda unitaria se conocen como vectores base. Estos vectores deben ser linealmente independientes y se relacionan con ángulos específicos que describen la asimetría de la red. La siguiente imagen muestra los vectores base con sus respectivos ángulos correspondientes:

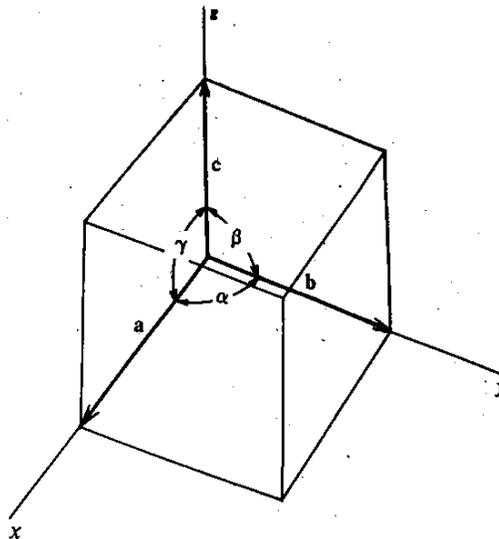


Figura 2. Parámetros reticulares.

Como se puede observar en la Figura 2 los vectores  $a$ ,  $b$  y  $c$  corresponden a las longitudes de las aristas y también corresponden a ejes de coordenada  $x$ ,  $y$ ,  $z$  respectivamente. Con respecto a los ángulos, se puede apreciar que el ángulo  $\alpha$  corresponde al ángulo entre las aristas  $a$  y  $b$ , el ángulo  $\beta$  entre las aristas  $b$  y  $c$  y finalmente el ángulo  $\gamma$  entre las aristas  $a$  y  $c$ .

La condición fundamental es que un vector  $G$  que apunta a cualquier punto de la red pueda expresarse en términos de múltiplos enteros de estos vectores, es decir:

$$G_{hkl} = h\vec{a} + k\vec{b} + l\vec{c}$$

donde  $h$ ,  $k$  y  $l$  son enteros positivos o negativos conocidos como índices de Miller que veremos más adelante. Si este sistema de ejes y sus vectores pueden ser desplazados y ajustados en todos los puntos de la red, estamos tratando con un monocristal. Si solo pueden ser trasladados y ajustados en puntos específicos, se trata de un sólido policristalino.

Matemáticamente, se ha demostrado que existen solo 14 combinaciones posibles para organizar los puntos en las redes espaciales, conocidas como las 14 redes de Bravais. Cada una de estas redes aseguran que cada punto en la red esté rodeado por un patrón de puntos idénticos al observado en cualquier otro punto de la misma red.

Tenemos que las secuencias de apilamientos de planos permiten describir redes cristalinas, la forma más elegante y útil de estudiarlas consiste en utilizar las herramientas algebraicas para especificar un grupo de traslación. Pensemos en una intersección en la red como la mostrada en la Figura 2, a partir de ahí, consideremos tres vectores  $x = \vec{a}$ ;  $y = \vec{b}$ ;  $z = \vec{c}$ , que van de un punto  $u$  origen hasta la siguiente intersección en cada una de las tres direcciones. Estos tres vectores pueden definir en el espacio un paralelepípedo al cual llamaremos *celda unidad*. Se conoce entonces como *red directa* a la repetición periódica de la celda unidad en las tres direcciones del espacio. Las celdas unidad se suelen describir geoméricamente mediante los llamados, *parámetros reticulares*, que son las longitudes de los vectores básicos  $\{\vec{a}, \vec{b}, \vec{c}\}$  y los ángulos que estos vectores forman entre los vértices, definidos por convenio como se muestran en la Figura 2.

La elección de los tres vectores no es única, y la terna puede generar un número infinito de formas, lo cual significa que la celda unidad para una red dada tampoco es única. Pero dado que nosotros

escogimos como nuestro sistema de simetría a la traslación, entonces podemos escoger aquellas estructuras que contengan entre sus simetrías a la traslación de la red directa. Esta condición nos lleva a que el número de posibles celdas unidad tridimensionales sea exactamente catorce, las cuales se agrupan en siete grupos llamados *sistemas cristalinos*.

Este grupo de sistemas cristalinos se les conoce como *redes de Bravais* y a continuación se muestran en la Tabla 1.

En tres dimensiones existen 14 estructuras posibles, las de mayor simetría son las *cúbicas*, las cuales se caracterizan porque sus tres aristas tienen igual longitud y son mutuamente perpendiculares. Contamos tres tipos de estructuras cúbicas, la *cúbica simple o primitiva (P)*, la *cúbica centrada en el interior o bcc (I)*, y la *cúbica centrada en las caras o fcc (F)*.

El segundo sistema más simétrico es el *tetragonal*, donde las aristas siguen siendo perpendiculares, pero una de ellas con diferente longitud, y existen dos tipos de estructuras, una primitiva (P) y una centrada en el interior (I).

El tercer sistema cristalino es el *ortorrómbico* que se caracteriza por tener vectores básicos mutuamente perpendiculares, pero de distinta longitud. Existen cuatro tipos, primitiva (P), centrada en el interior (I), centrada en las caras (F) y centrada en dos caras paralelas entre sí (C).

El cuarto sistema se llama *monoclínico*, y se puede ver como un sistema ortorrómbico distorsionado en uno de sus ángulos rectos del paralelepípedo, y existen dos tipos, primitiva (P) y centrado en dos caras (C).

El quinto sistema y el de menor simetría es el *triclínico*, que sólo admite estructuras tipo (P).

El sexto sistema se llama *trigonal*, compatible solo con una celda primitiva que se puede visualizar como una cúbica estirada por una de sus diagonales, también llamada estructura *romboédrica* o tipo (R).

Finalmente, el séptimo sistema es el que se puede obtener de la cúbica mediante distorsiones y que se conoce como *hexagonal*, y solamente existe como primitiva (P).

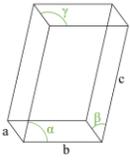
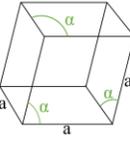
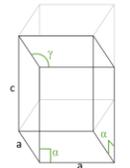
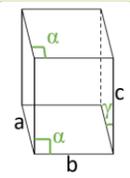
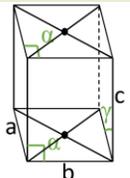
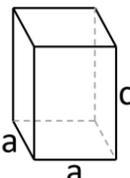
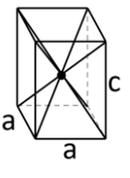
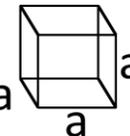
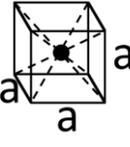
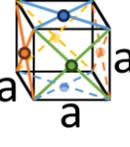
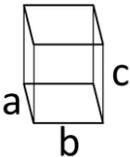
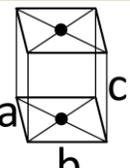
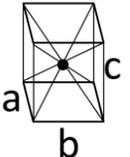
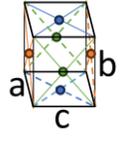
Sistema Cristalino	Características de las Celdas Unitarias	Redes de Bravais			
		Primitiva	Centrada en la base	Centrada en el cuerpo	Centrada en las caras
<b>Triclínico</b>	$\alpha \neq \beta \neq \gamma$  $a \neq b \neq c$				
<b>Trigonal</b>	$(\alpha = \beta = \gamma) \neq 90^\circ$  $a = b = c$				
<b>Hexagonal</b>	$\alpha = \beta = 90^\circ$ ; $\gamma = 120^\circ$  $a = b \neq c$				
<b>Monoclínico</b>	$\alpha = \beta = 90^\circ$ ; $\gamma \neq 120^\circ$  $a \neq b \neq c$				
<b>Tetragonal</b>	$(\alpha = \beta = \gamma) = 90^\circ$  $a = b \neq c$				
<b>Cúbico</b>	$\alpha = \beta = \gamma = 90^\circ$  $a = b = c$				
<b>Ortorrómbico</b>	$\alpha = \beta = \gamma = 90^\circ$  $a \neq b \neq c$				

Tabla 1. Sistemas cristalinos, Redes de Bravais.

Las redes de Bravais son una disposición periódica de puntos en el espacio, la base es el conjunto de átomos o moléculas que se colocan en cada punto de la red, y la estructura cristalina es la combinación de la red y la base que se repite en todo el cristal.

La presencia de ciertas operaciones de simetría en el grupo espacial de un cristal hace que los factores de estructura se anulen sistemáticamente para algunas reflexiones. En ese caso no se observa intensidad difractada a lo largo de direcciones del espacio recíproco predichas por las ecuaciones de Laue (2). Estas extinciones sistemáticas surgen cuando se utilizan celdas múltiples para describir la red directa del cristal; en ese caso, las extinciones sistemáticas afectan a vectores  $G_{hkl}$  distribuidos por todo el volumen del cristal. También surgen cuando existen operaciones de simetría no puntuales, aunque en ese caso las reflexiones ausentes están restringidas a un plano o a una dirección del espacio recíproco. Las extinciones sistemáticas permiten distinguir si una operación está o no presente en el grupo espacial del cristal y, por tanto, contribuyen a identificarlo.

Elemento de Simetría	Condición de Extinción
P	ninguna
I	$h+k+l = 2p+1$
A	$k+l = 2p+1$
B	$h+l = 2p+1$
C	$h+k = 2p+1$
F	h, k, l con distinta paridad
R	$h-k+l \neq 3p$

Tabla 2. Puntos de extinción en el patrón de difracción.

### Índices de Miller

Consideremos ahora un plano reticular que corta a los ejes definidos por los vectores básicos en tres puntos, A, B y C, con coordenadas  $(r,0,0)$ ,  $(0, m,0)$  y  $(0,0,s)$  respectivamente, dando lugar a la región indicada en la figura siguiente.

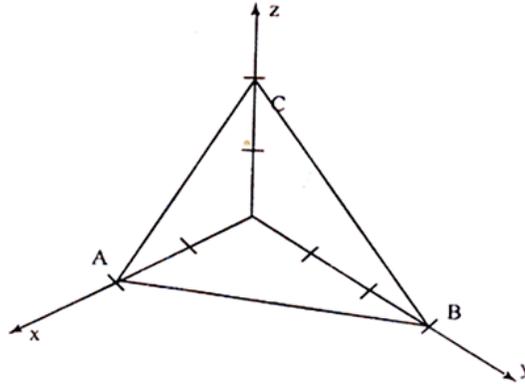


Figura 3. Plano reticular

Sean entonces:

$$h = \frac{1}{r}; \quad k = \frac{1}{m}; \quad l = \frac{1}{s} \quad (1)$$

de donde obtendremos números enteros sin divisor común. Entonces el plano reticular en cuestión se etiqueta como  $(h, k, l)$ , llamados los *índices de Miller*. La ventaja del uso de la notación de Miller reside en que, con ella, y sobre todo en los sistemas cúbicos, resulta muy fácil evidenciar las simetrías del cristal.

Hay que hacer notar que la región marcada en la Figura 3 por el triángulo ABC no representa un único plano, sino una familia de planos reticulares entre sí. Por extensión, se asignan los mismos índices de Miller  $(hkl)$  a todos los planos de esta familia. Dependiendo del valor de  $r$ ,  $m$  o  $s$ , los índices de Miller pueden ser positivos o negativos.

Entonces, si la construcción de un sólido en la red directa puede describirse por su simetría de traslación en el espacio real, con lo cual se obtiene una estructura cristalina, cuyas propiedades

físicas pueden tomar valores iguales en ciertos puntos para conformar dicha red, entonces, también podemos construir una segunda red llamada *red recíproca*, definida en el espacio de Fourier. Esta nueva red asociada a una red directa será entonces también una red de Bravais y sus simetrías guardan ciertas correspondencias.

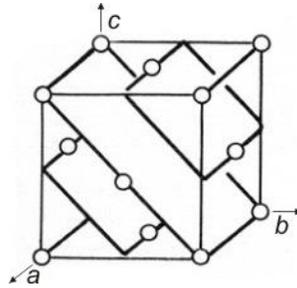


Figura 4. Sólido como una familia de planos.

Consideremos un sólido como el que se muestra en la Figura 4, este sólido podemos representarlo como una familia de planos paralelos entre sí e identificarlos por índices de Miller (h, k, l) separados una distancia  $d_{hkl}$  y orientados por un vector normal a ellos  $\mathbf{n}_{hkl}$ . De aquí que, para cada familia de planos se define un vector  $\vec{q}_{hkl}$  tal que:

$$|\vec{q}_{hkl}| = \frac{2\pi}{d_{hkl}} \quad (2)$$

Puesto que  $d_{hkl}$  es una distancia, estos vectores no están definidos en el espacio real, sino en el espacio de vectores de onda o espacio de Fourier, que en el contexto de la Física del estado Sólido se le llama *espacio recíproco*.

Hay que hacer notar que la región marcada en la Figura 3 por el triángulo ABC no representa un plano único, sino una familia de planos reticulares entre sí. Por extensión, se asignan los mismos índices de Miller (hkl) a todos los planos de esta familia. Dependiendo del valor de r, m o s, los índices de Miller pueden ser positivos o negativos. [2,11]

1.- Planos Cristalinos: Los índices de Miller de un plano cristalino se representan generalmente como (hkl), donde "h," "k" y "l" son números enteros que se obtienen a partir de las intersecciones

del plano con los ejes de coordenadas del sistema cristalino. Estos números se normalizan para asegurarse de que sean proporcionales y expresen la relación entre los planos de manera clara.

2.- Direcciones Cristalinas: Los índices de Miller para direcciones cristalinas se representan generalmente como  $[uvw]$ , donde "u," "v" y "w" son números enteros que describen la orientación de la dirección en el cristal. Estos números también se normalizan para garantizar una descripción clara y proporcional de la dirección.

Los índices de Miller son especialmente útiles para describir la estructura cristalina de un material de manera precisa y comprensible. Ayudan a los cristalógrafos a comunicar y entender la orientación de los planos y las direcciones en un cristal, lo que es esencial en la determinación de estructuras cristalinas y en la interpretación de datos de difracción de rayos X y otras técnicas de caracterización de materiales.

Para encontrar los índices de Miller de un plano cualquiera se procede en la siguiente forma:

- 1.- Elíjase un origen conveniente para el sistema de ejes cristalográficos asociados a los vectores base, y encuentrense las intersecciones  $(x, y, z)$  del plano en cuestión de estos ejes.
- 2.- Divídase estas intersecciones entre los parámetros  $a, b, c$  respectivos de la red, lo que dará tres números enteros  $x/a, y/b, z/c$ .
- 3.- Obténganse los recíprocos de estos números y redúzcanse tomando el mínimo común denominador de estas fracciones recíprocas y multiplicándolas, por el mismo, al más pequeño sistema de tres números  $h, k, l$ , que guarden las mismas razones entre sí, que las fracciones recíprocas anteriores.
- 4.- Estos tres números  $(h, k, l)$  son los índices de Miller.

## Difracción de Rayos X y HTEM

Una de las principales herramientas que existen para estudiar y analizar un cristal es la difracción y en particular, la difracción de electrones. De aquí, que uno de los instrumentos más utilizados para el estudio de los cristales es el Microscopio Electrónico de Transmisión [12].

Un esquema de uno de estos instrumentos se presenta en la Figura 5. Con este microscopio es posible obtener imágenes con resoluciones (separación punto a punto) de hasta  $1\text{\AA}$  lo cual permite darnos una buena idea de la estructura de un cristal. Sin embargo, también es posible observar con mucha precisión el patrón de difracción que forma al infinito un área selecta de la muestra. Debido a que el patrón de difracción de un objeto es único, es decir, no existen dos iguales [8], resulta ser una forma sumamente interesante de estudiar los cristales ya que nos permite identificar los planos cristalinos del material que estamos observando. Esta identificación es posible hacerla analizando con mucho cuidado algunos de los puntos que se observan en el patrón de difracción, por lo que, en una buena medida, este sería el objetivo principal de nuestro trabajo.

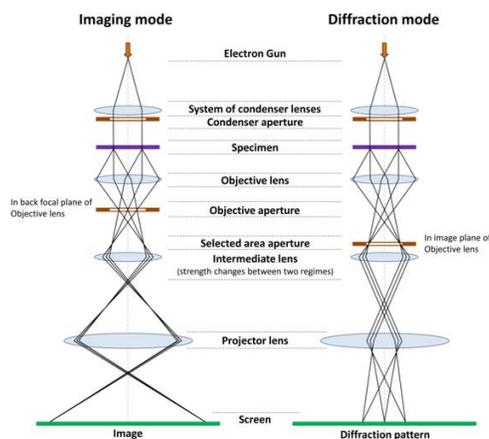


Figura 5. Microscopio electrónico de transmisión.

Para entender mejor nuestra tarea, recordemos que la luz se puede estudiar indistintamente en su forma ondulatoria, es decir, por ondas electromagnéticas como los rayos X, o en su forma corpúsculas, es decir por partículas como fotones, electrones, neutrones, etc. En nuestro caso particular, nos es de gran utilidad el hecho de que existen bases de datos que describen cristalográficamente (distancia del punto al centro del patrón, y plano  $\{hkl\}$  de origen) el origen de

los puntos que aparecen en un patrón de difracción obtenidos con Rayos X, por lo general, con una fuente de CuK $\alpha$ 1 y una longitud de onda de  $\lambda = 1.540598 \text{ \AA}$  de prácticamente todos los elementos químicos y de una gran cantidad de materiales derivados de los mismos. Mas adelante (página 27) indicamos cual es la importancia para nosotros tener acceso a estas bases datos, ya que representan un punto de partida para nuestros cálculos, sólo que con una  $\lambda = 2.51 \text{ \AA}$

¿Pero qué se entiende por patrón de difracción? Se entiende por difracción al fenómeno que ocurre cuando una onda o partícula se ve obstruida, desviada o deformada parcialmente por un objeto en su trayectoria [8]. Se puede decir que un patrón de difracción es un patrón de interferencia, es decir, cuando dos ondas que se desplazan en la misma dirección en el espacio se encuentran, ocurre que, por la naturaleza de sus componentes, se suman o se restan alterando la intensidad de la onda resultante en diferentes puntos del plano donde se observe.

Cuando la intensidad de la onda resultante crece decimos que tenemos interferencia constructiva, y cuando la intensidad de la onda resultante disminuye, decimos que tenemos interferencia destructiva.

Dado que estamos considerando que un cristal es una red reticular, nuestro patrón de difracción obtenido por este método estará compuesto por máximos y mínimos de intensidad luminosa, geoméricamente distribuidos acorde a la distribución de los átomos en el cristal. El patrón de difracción al igual que en un patrón de interferencia siempre tendrá un punto central que resalta por ser el de máxima intensidad, y que es el punto donde inciden todos los haces o partículas que no son desviadas durante su viaje a través del cristal.

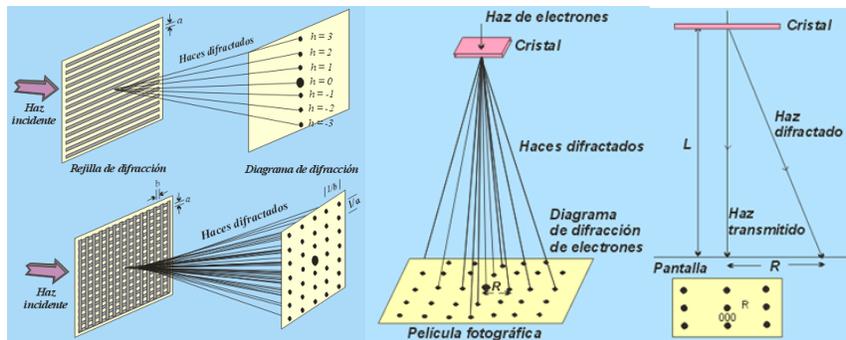


Figura 6. Formación de un patrón de difracción por una rejilla y por un cristal.

El resultado sobre la onda o partícula se puede observar desde dos puntos de vista diferentes [8]:

- 1.- Aquella donde tanto el plano de la fuente como el plano de la imagen se encuentran a una distancia infinita del objeto que provoca la difracción. Esta se conoce como difracción de Fraunhofer.
- 2.- Aquella donde el plano de la fuente o el plano de la imagen o ambos se encuentran a una distancia finita del objeto que provoca la difracción. Esta se conoce como difracción de Fresnel.

Dado que nos encontramos trabajando con las imágenes producidas con un microscopio electrónico de transmisión, los patrones de difracción se forman al infinito, es decir, difracción de Fraunhofer, pero con ayuda de los sistemas de lentes en el microscopio podemos transformarla en una difracción de Fresnel y tener una imagen de estos patrones en el plano imagen del microscopio.

Otra forma de hacerlo es trabajar con la imagen de alta resolución que forma el microscopio electrónico de una zona muy pequeña de la muestra. Afortunadamente hoy en día es muy fácil obtener estas imágenes en forma digital, la cual se encuentra representada como una matriz de  $(n \times m)$  elementos. Si a esta matriz le aplicamos una transformada rápida de Fourier (FFT) equivale a hacer lo que produce el microscopio al formar el patrón de difracción [8,12].

Evidentemente existe una gran diferencia entre el patrón generado directamente del objeto en tres dimensiones que el obtenido de una imagen en dos dimensiones, y esta diferencia se encuentra en el número de puntos que podemos observar en cada caso. Sin embargo, este último resulta ser de mucha más utilidad ya que el primero, es el resultado de una zona muy grande en la muestra comparado con el tamaño de la zona de la muestra que se toma en la segunda opción, es decir, es muy selectiva.

Es entonces que, a través de la difracción, se revela la arquitectura interna de los cristales y se abre una ventana al mundo de la *Cristalografía* (ciencia que se encarga de estudiar a los cristales) y que es tema fundamental de nuestro trabajo.

Observemos ahora la Figura 7, del círculo arriba a la derecha podemos observar una red de planos paralelos separados una distancia  $d$  (pensemos en un sólido como el de la Figura 3). Consideremos también un haz de electrones que incide sobre el plano superior de la red en un punto  $O'$  a un ángulo  $\theta$  con respecto a la normal  $N$  en ese punto. Como se observa, el haz deberá salir difractado en la dirección  $2\theta$  y si consideramos un círculo de radio  $1/\lambda$  y medimos la separación que habría entre la prolongación del haz incidente y el haz difractado y su intersección con el círculo,  $O''$  y  $P''$  respectivamente podemos deducir que [1]:

$$n\lambda = 2d \sin \theta \quad (3)$$

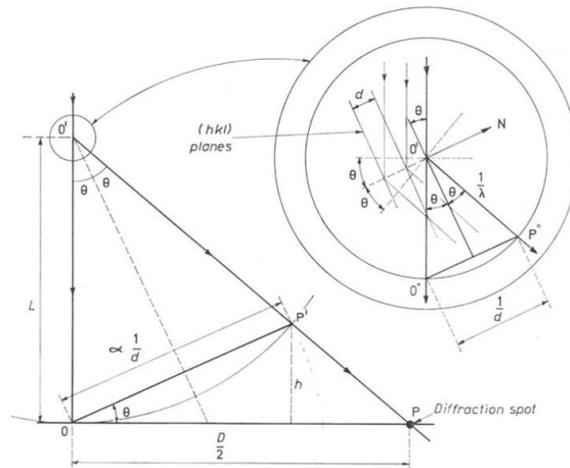


Figura 7. Esquema geométrico de la difracción en un cristal.

que es la conocida *Ley de Bragg*, donde  $n$  es un número entero. Si en la ecuación (3) hacemos  $n=1$  entonces:

$$d^* = \frac{1}{d} = 2 \left( \frac{1}{l} \right) \sin \theta \quad (4)$$

donde  $d^*$  se conoce como la distancia interplanar de la red en el espacio recíproco y se relaciona con la ecuación (2) que vimos antes. Entonces podemos decir que  $d_{hkl}^* = 1/d_{hkl}$  es un vector

perpendicular al plano (hkl) e inversamente proporcional a la distancia interplanar en espacio real  $d_{hkl}$ .

Si observamos la Figura 8, podemos entonces decir que los parámetros de la red los podemos ver como:  $d_{100} = a$ ;  $d_{010} = b$  y  $d_{001} = c$ .

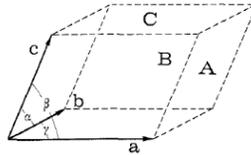


Figura 8. Parámetros de la red real.

La esfera que hemos construido alrededor del punto O' se le conoce como *esfera de Ewald*, la cual nos ayuda a interpretar mejor los patrones de difracción que se forman en un plano normal al haz incidente que viaja sobre OO' en la figura. El rayo difractado viaja a lo largo de la dirección O'P' y forma un spot sobre el plano imagen en P.

Finalmente, en microscopía electrónica la distancia  $L=OO'$  es conocida como la *longitud de cámara*, y la longitud de onda  $\lambda$  está dada por el voltaje que tiene el haz de acuerdo con la ecuación:

$$\lambda = \frac{12.236}{\sqrt{V}} \text{ \AA} \quad (5)$$

donde V es el voltaje en KV. Como podemos ver en la ecuación (3) la longitud de onda juega un papel sumamente importante en la formación de los patrones de difracción, por lo que el valor de esta variable tiene que estar bien definida para poder hacer nuestros cálculos.

Veamos ahora la Figura 9, en ella se hace un diagrama reticular con tamaños de retícula ( $a^* \times b^*$ ), de un patrón de difracción en un plano y donde seleccionamos el punto O como el spot central. Consideremos que tenemos tres spots, uno con coordenadas ( $a^*, 0$ ), otro en ( $0, b^*$ ) y un tercero en ( $a^*, b^*$ ).

La forma de analizar este patrón de difracción puede ser de la siguiente manera:

- Obtener las distancias desde O a cada spot, es decir obtener  $D_1/2$ ,  $D_2/2$  y  $D_3/3$  los cuales corresponden a las distancias  $d^*$  que queremos conocer.
- Obtener los ángulos  $\phi_1, \phi_2, \phi_3$  entre los tres vectores recíprocos lo cual equivale a tener los ángulos entre los planos que estamos tratando de indentificar.

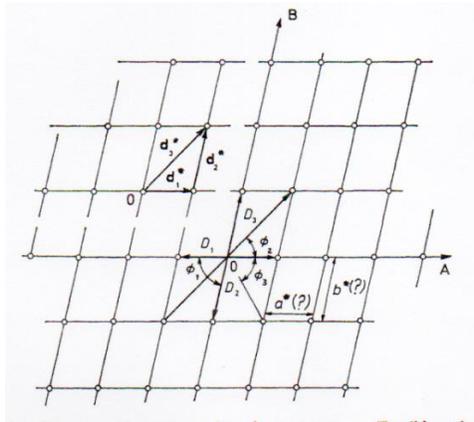


Figura 9. Diagrama en un espacio recíproco [1].

Suponiendo que tenemos una idea del material del cual es nuestro cristal, podemos entonces con la información hasta ahora obtenida, calcular las distancias y los ángulos procediendo de la siguiente manera:

- Comparar los valores de las distancias convertidas a espacio real y en las unidades adecuadas, con otros valores ya reportados para dicho material (por ejemplo, las tablas de Rayos X), lo cual nos da como punto de partida la familia de planos  $\{hkl\}$  de posibles soluciones.
- Con las tres familias de planos, calcular todos los posibles ángulos entre ellos hasta obtener los más cercanos a los medidos anteriormente.

Este proceso es un poco complicado por la multiplicidad de planos involucrados en este cálculo y los posibles errores al hacer las mediciones de las distancias o de los ángulos. Sin embargo, al

considerar los ángulos como indicador de una posible solución, el número de posibles planos que cumplan con el requisito de los ángulos el error disminuye considerablemente.

Para refinar más la selección de posibles soluciones, consideramos lo que en cristalografía se llama zona, es decir, a un conjunto de planos reticulares cuyas intersecciones son paralelas entre sí. La normal a estos planos se llama *eje de zona*. Teniendo en cuenta la orientación de los vectores de la red recíproca con respecto a los planos reticulares del espacio directo, se puede demostrar que la familia de planos  $\{hkl\}$  tienen como vector normal al vector  $[uvw]$  llamado *eje de zona* [1], de donde es claro que entonces, cada plano deberá cumplir que:

$$hu + kv + lw = 0. \quad (6)$$

Por su parte, el eje de zona está definido por dos familias de planos, digamos  $\{h_1k_1l_1\}$  y  $\{h_2k_2l_2\}$  por:

$$[uvw] = \{h_1k_1l_1\} \times \{h_2k_2l_2\}$$

$$\begin{aligned} u &= k_1l_2 - k_2l_1 \\ v &= l_1h_2 - l_2h_1. \\ w &= h_1k_2 - h_2k_1 \end{aligned} \quad (7)$$

Así entonces, si ya encontramos las condiciones necesarias para identificar a dos de los posibles planos, entonces podemos agregar una tercera familia de planos  $\{h_3k_3l_3\}$  ya que:

$$d_3^* = d_1^* + d_2^* \quad (8)$$

$$h_3 = h_1 + h_2, \quad k_3 = k_1 + k_2, \quad l_3 = l_1 + l_2$$

Lo cual nos lleva a que se tenga que cumplir con la siguiente condición [1]:

$$\begin{aligned}h_3 &= mh_1 + nh_2 \\k_3 &= mk_1 + nk_2 \\l_3 &= ml_1 + nl_2\end{aligned}\tag{9}$$

Donde m y n son dos enteros (con igual o diferente signo) y que se convierte en nuestra condición más importante para saber si tenemos o no una solución válida.

## CAPÍTULO IV

### Indexación de patrones de difracción

#### *Algoritmo*

Nos ocuparemos en indexar los patrones de difracción a partir de la posición espacial de tres puntos del patrón de difracción (spots) con respecto al spot central, y de los datos conocidos en alguna de las tablas de Rayos X que existen en la literatura, ya que es fundamental tener una idea del material del que proviene el patrón de difracción. El programa supone que los valores de (x, y) que se proporcionan como datos, ya están en la escala adecuada.

Los principales pasos que se siguieron para la elaboración de este programa se observan en el diagrama de bloques que se muestra en la Figura 9, el cual se basa en su totalidad en la teoría y las recomendaciones que resumimos en el capítulo de antecedentes en este trabajo.

Es importante indicar que este programa fue pensando, siguiendo una rutina ya establecida en aquellos usuarios que tienen la posibilidad de usar un microscopio electrónico de transmisión para obtener los patrones de difracción o en su defecto, lo que harían si lo que tienen es una imagen en alta resolución (HREM) en forma digital de una zona de la muestra que les interesa analizar, y de donde se puede obtener una FTT para obtener el patrón de difracción correspondiente.

Existen varias formas de obtener los datos que se requieren a partir del patrón de difracción; nos referimos a las coordenadas espaciales donde se encuentran los spots y el tamaño de los ángulos. Algunos lo hacen midiendo directamente sobre una imagen del patrón de difracción impresa en papel, pero teniendo cuidado de escalar sus medidas en base a la longitud de cámara (L) y la longitud de onda ( $\lambda$ ) con la que trabaja el microscopio (lo cual depende del voltaje de trabajo).

En nuestro caso, el grupo con el que colaboramos en este proyecto cuenta con un software (existen otros) que es una de las herramientas con la que cuenta su microscopio electrónico marca JEOL, se trata del paquete DigitalMicrograph 3.4 [9], el cual nos ayuda no solo a desplegar en pantalla nuestra imágenes digitales obtenidas en el microscopio electrónico ya escaladas, sino que permite hacer mediciones de distancias y ángulos con gran precisión, así como realizar algunos cálculos tales como obtener la FFT de un área (de 64x64, 128x128 o 256x256 píxeles) selecta de una imagen, aplicar filtros digitales, encontrar puntos de máxima intensidad, etc.

Aún con estos paquetes, las medidas de distancias y ángulos no están exentas de errores, ya que todo depende de la posición del cursor cuya selección depende sólo del usuario, y el cual puede que no corresponda exactamente a la posición del máximo de intensidad del spot, que es desde donde se debe de hacer la medición. Este posible error será tomado en cuenta en el bloque 3 para tratar de corregirlo.

De la Figura 9 siguiente, tenemos:

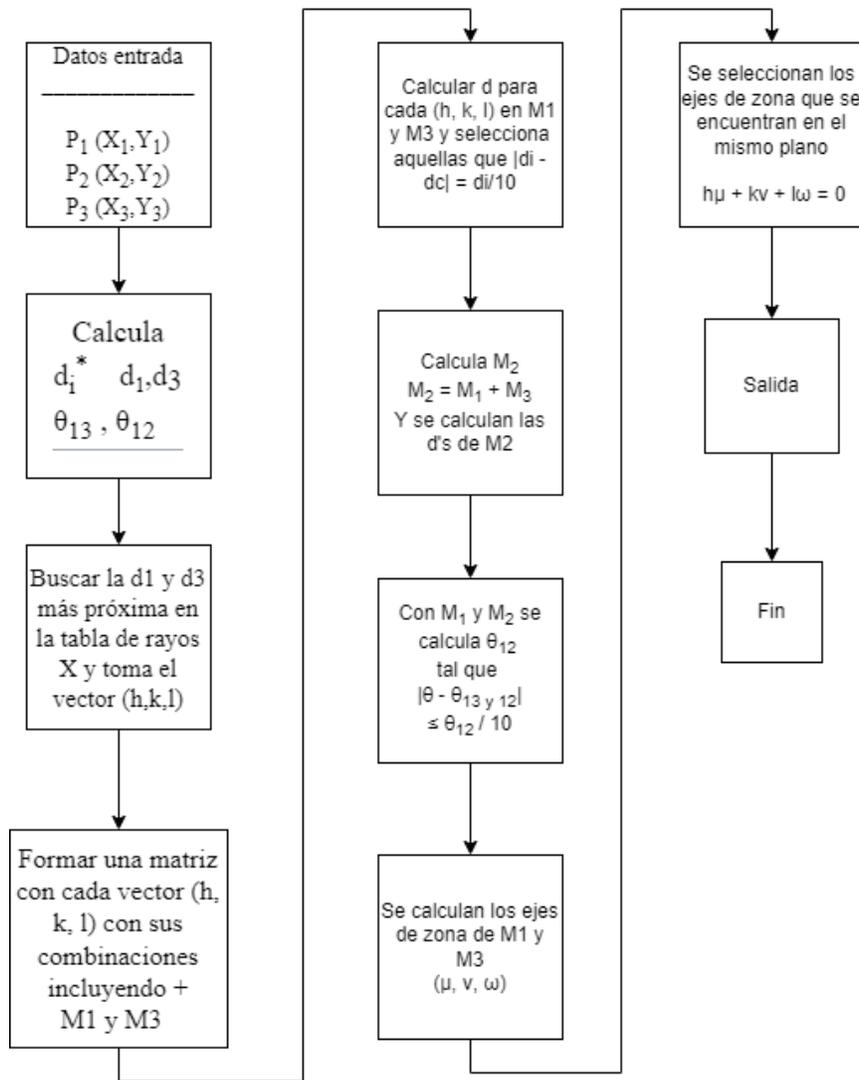


Figura 9. Diagrama de bloques.

1.- *Lectura de datos de entrada.* Los datos de entrada son importantes para el buen funcionamiento del programa, el cual solicita:

- La estructura cristalina con la que se quiere trabajar, eso quiere decir que el usuario debe saber de antemano cual es la estructura cristalina con la que quiere analizar sus datos, cuidando que coincida con la que le indica la tabla de Rayos X a usar;
- El nombre de la Tabla de rayos X con la que va a trabajar, el programa le dará la opción de trabajar con alguna de las ya existentes en la base de datos o crear una nueva. Cada vez que se introduce una tabla nueva, ésta se almacena en un archivo de Excel;
- Le pide también el valor máximo para cada elemento del vector (h, k, l) que se quiere como solución, existen valores de varias decenas, sin embargo, algunos especialistas recomiendan trabajar únicamente con valores menores a 10 así que el programa deja que ese valor máximo sea decisión del usuario;
- La posición espacial (x, y) de los spots con los que se va a trabajar; y
- Finalmente, el nombre del archivo de salida con extensión xls.

De la información de la posición espacial  $S_i=(x_i,y_i)$  de tres spots seleccionados por el usuario del patrón de difracción que se va a analizar, hay que tener cuidado de: i) que no se consideren entre los tres spots seleccionados, el punto y su conjugado, recordar que por ejemplo el punto (1,0) es conjugado del punto (-1,0), es decir, corresponden al mismo spot, y ii) que los datos fueron obtenidos de una imagen calibrada previamente donde el punto origen ( $x=0,y=0$ ) corresponde al spot central, y el cual se reconoce por ser el spot de máxima intensidad.

De la tabla de Rayos X, cuando se crea una tabla nueva, ésta se construye en Excel donde se crea una matriz de n puntos por 4 columnas donde se encontrarán:

- (1,1) = (#) Estructura: (1) Cúbica; (2) Tetragonal;(3) Ortorrónica; (4) Hexagonal; (5) Romboédrica; (6) Monoclínica y (7) Triclínica.
- (1,2) = valor del parámetro de red a;
- (1,3) = valor del parámetro de red b;
- (1,4) = valor del parámetro de red c;
- (2,1) = valor del ángulo  $\alpha$ ;
- (2,2) = valor del ángulo  $\beta$ ;
- (2,3) = valor del ángulo  $\gamma$ ;

- (2,4) = (0) si la red es Primitiva; (1) si la red es centrada en las caras; (2) si la red es centrada en el cuerpo y; (3) si la red es centrada solo en dos caras paralelas entre sí.

A partir de la fila  $i=3$  y hasta  $i= \#$  de puntos en la tabla de rayos X, se encuentran:

- (i,1) = valor de la distancia en Ångstroms [ $\text{Å}$ ];
- (i,2) = valor de h;
- (i,3) = valor de k;
- (i,4) = valor de l;

2.- *Calcula distancias y ángulos.* Comenzamos trabajando en espacio recíproco, por lo que el primer paso es calcular las distancias  $d_{(x,y)}^*$  (distancia interplanar) que hay entre el origen (0,0) y la posición  $S_i=(x, y)$  de los spots seleccionados por el usuario. Cada valor es transformado a espacio real y es escalado a Ångstroms [ $\text{Å}$ ], entonces ya tenemos las distancias  $d_i$ .

Conocidas las distancias  $d_i$ , regresa a espacio recíproco para calcular los ángulos que se forman entre los vectores  $S_1S_3(\theta_{13})$  y  $S_2S_3(\theta_{23})$ , el correspondiente entre  $S_1S_2(\theta_{12})$  se obtiene de la diferencia entre los dos primeros.

3.- *Comparar datos con tablas de Rayos X.* El programa toma uno a uno cada valor de  $d_i$  y lo compara con las  $d_x$  de la tabla de rayos X hasta que encuentra la posición  $i$  ( $i=1$  hasta  $n$ ) donde la diferencia ( $d_x-d_i$ ) es la más cercana o igual a cero. En este caso lo que más importa es la posición  $i$ . Para cada  $d_i$  el programa tomará la familia  $\{hkl\}$  de la posición  $i$ , así como la familia de la posición  $i-1$  y  $i+1$ , por lo que, para cada  $d_i$  se tienen tres familias como posibles soluciones. Esto se hace al considerar un posible error al colocar el cursor para encontrar las coordenadas (x, y) en el bloque 1. Puede haber casos donde por la posición  $i$ , sólo se pueda agregar una posición más, lo cual ocurre en los extremos de la tabla de rayos X.

4.- *Forma matrices con posibles soluciones.* De cada familia de vectores  $\{h, k, l\}$  seleccionadas en el bloque anterior, se obtienen las permutaciones sin repetición considerando ambos signos. Lo cual, en principio, nos da una matriz de  $48 \times 3$  por cada familia, es decir, una matriz  $M_i$  de  $144 \times 3$ . Puede que la posición  $i$  de donde partimos se encuentre entre los primeros o dos últimos puntos de la tabla de rayos X, en estos casos sólo se podrán tomar dos familias de vectores  $\{h, k, l\}$  por lo que terminaremos con una matriz  $M_i$  de  $96 \times 3$  para cada  $d_i$ .

Es muy común que existan vectores (hkl) que se repitan durante el proceso, por lo que el programa irá eliminando los vectores repetidos, por lo que la dimensión de las matrices  $M_i$  podrían ser más pequeñas que lo indicado anteriormente.

5.- *Calcula las  $d_{hkl}$  y elimina aquellas con un error mayor al 10%.* En este bloque, para cada vector (h, k, l) en las matrices  $M_1$  y  $M_3$  se calculará la  $d_{hkl}$  correspondiente de acuerdo con las ecuaciones que se presentan en el Anexo II. El valor resultante se compara con la  $d_i$  obtenida en el segundo bloque, eliminando aquellos vectores donde la diferencia  $|d_i - d_{hkl}| > d_i/10$ . Con esto, las nuevas matrices se reducen y terminamos con dos nuevas matrices con  $N_1[n_1,3]$  y  $N_3[n_3,3]$ , respectivamente.

6.- *Calcula el ángulo ( $\theta_{13c}$ ).* Este bloque calcula el ángulo que se forma entre los vectores que se guardaron en las matrices  $N_1$  y  $N_3$ , cada vector (h, k, l) de  $N_1$  se combina con cada vector de  $N_3$  para calcular  $\theta_{13c}$  con ayuda de las ecuaciones que se enlistan en el Anexo II.

Cada  $\theta_{13c}$  calculado es comparado con el obtenido a partir de los datos de entrada en el segundo bloque  $\theta_{13}$ , si la diferencia es menor o igual al 10% del valor (en grados) de este último, se almacena en una nueva matriz junto con los vectores con los que se calculó.

Puede darse el caso que ningún  $\theta_{13c}$  calculado cumpla con la condición anterior, con lo cual se obtendría una matriz vacía. Para evitar eso, el programa automáticamente incrementa la tolerancia de la diferencia en 0.1 grados, y vuelve hacer los cálculos necesarios.

Al final de este bloque terminamos con una matriz de  $N_{13}[n_{13},3]$  donde se almacenaron los vectores que si cumplieron con la tolerancia.

7.- *Calcula el ángulo ( $\theta_{12c}$ ).* Se procede de la misma manera que en el bloque 5 pero ahora con la matriz  $M_2$  hasta obtener una nueva matriz  $N_2[n_2,3]$ . Se procede ahora como el bloque 6 con las matrices  $N_2$  y  $N_{13}$  de la cual utilizaremos únicamente la primera columna, la que corresponde a los vectores (h, k, l) del spot 1 y que cumplieron hasta ahora con todas las condiciones ya explicadas en los bloques anteriores. Al final terminaremos con una nueva matriz  $N_{123}[n_{123},3]$ , en la cual se encontrarán los vectores (h, k, l) que representan las posibles soluciones para los tres spots, los dos ángulos encontrados  $\theta_{13c}$  y  $\theta_{12c}$ , así como sus diferencias con los ángulos medidos.

8.- *Calcula el eje de zona ( $u, v, w$ ).* Con los datos contenidos en  $N_{123}$ , se calcula el eje de zona de acuerdo con las ecuaciones en (7). El resultado se anexa a la matriz para terminar ahora con la matriz  $N_{123e}[n_{123},3]$ .

9.- *Comprueba que las posibles soluciones estén en el mismo plano.* Como se indicó en la teoría, todos los valores que se encuentran en  $N_{123e}$  pueden considerarse matemáticamente posibles soluciones al problema planteado, sin embargo, pueden diferir entre ellas en el hecho de que no se encuentren en el mismo plano que define la esfera de Ewald. Para que los vectores de cada fila puedan ser tomados como posibles soluciones, los tres vectores propuestos deberán cumplir con la ecuación (6).

10.- *Elimina cuando sea el caso, reflexiones prohibidas.* Como vimos anteriormente en la teoría, puede haber soluciones que se extinguen o como se dice en cristalografía, no están permitidas. En este bloque el programa busca si alguna de las posibles soluciones corresponde a uno de estos casos de extinción mostrados en la Tabla 2, y por consiguiente las elimina.

De igual manera, en este bloque elimina aquellas posibles soluciones cuyos valores de  $h, k, l$  sean mayores a los que solicitó el usuario.

11.- *Escribe las posibles soluciones.*

### **Diagramas de flujo**

Los siguientes diagramas de flujo muestran el proceso del paso a paso de cada sección del programa PIPD-II, resaltando las operaciones realizadas y variables globales usadas para almacenar la información. Todos los diagramas están relacionados y en varios diagramas tenemos un valor de retorno.

1.- *Obtención de datos:* En este primer diagrama se especializa para poder obtener los datos necesarios para la indexación de patrones de difracción, los cuáles son las coordenadas/ángulo y distancia de los puntos (punto1, punto2 y punto3) así como el tipo de estructura que se manejará para la indexación.

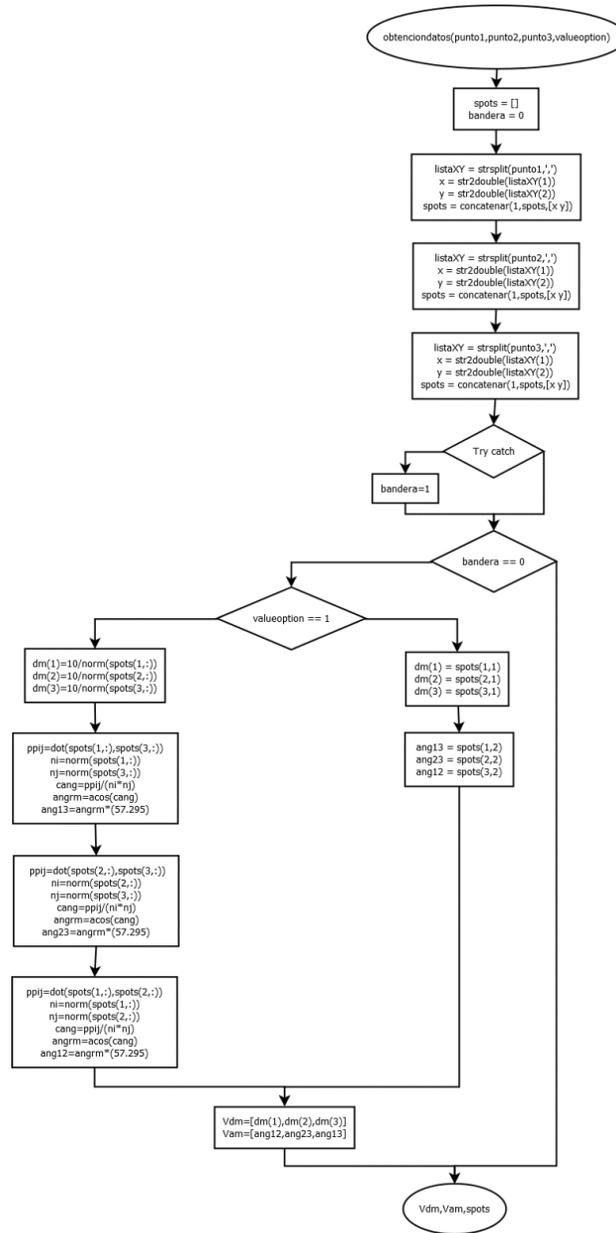


Diagrama de flujo 1 "Obtención de datos".

2.- Compara: Este diagrama obtiene las permutaciones de los puntos dando las diferentes combinaciones que existen entre los valores h k l.

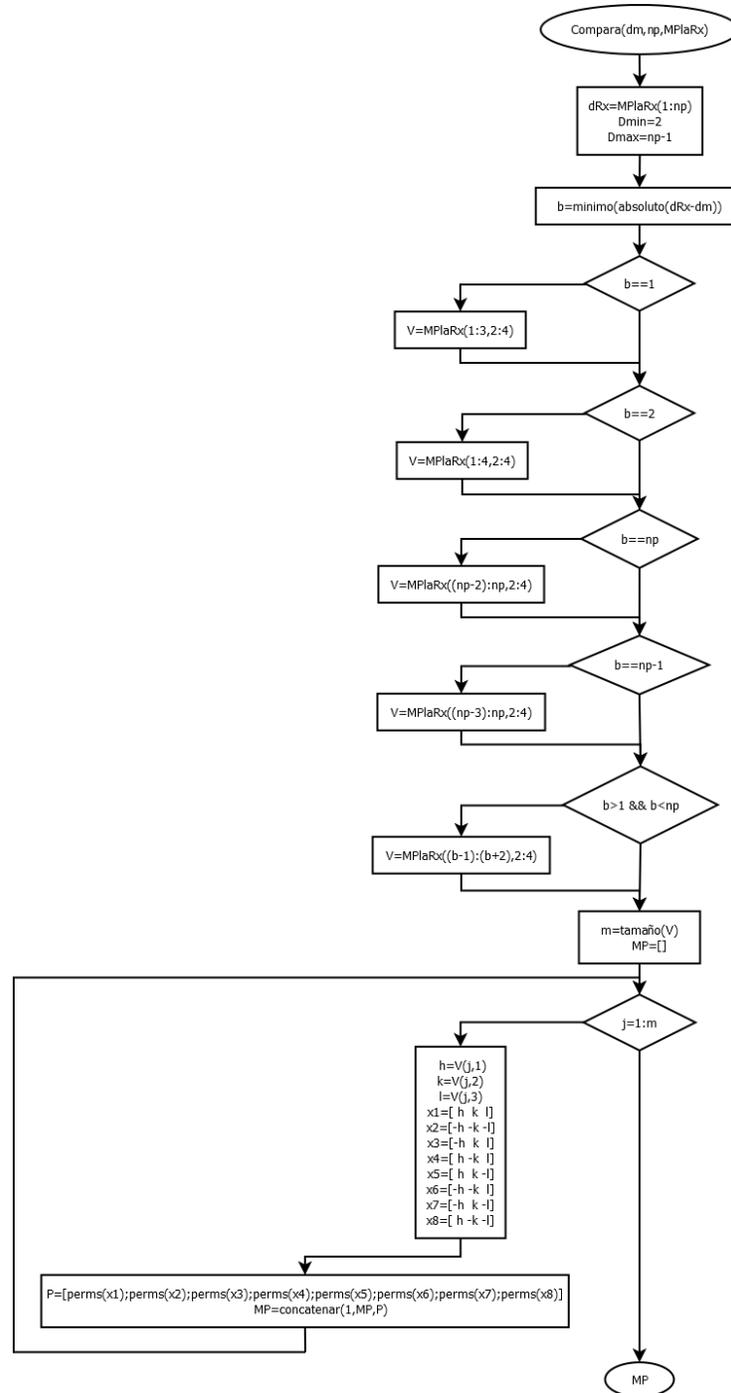


Diagrama de flujo 2 "Compara".

3.- Calculad13: Se calcula la distancia de un punto y el origen, y dependiendo del tipo de estructura con la que se está trabajando se obtendrá su distancia.

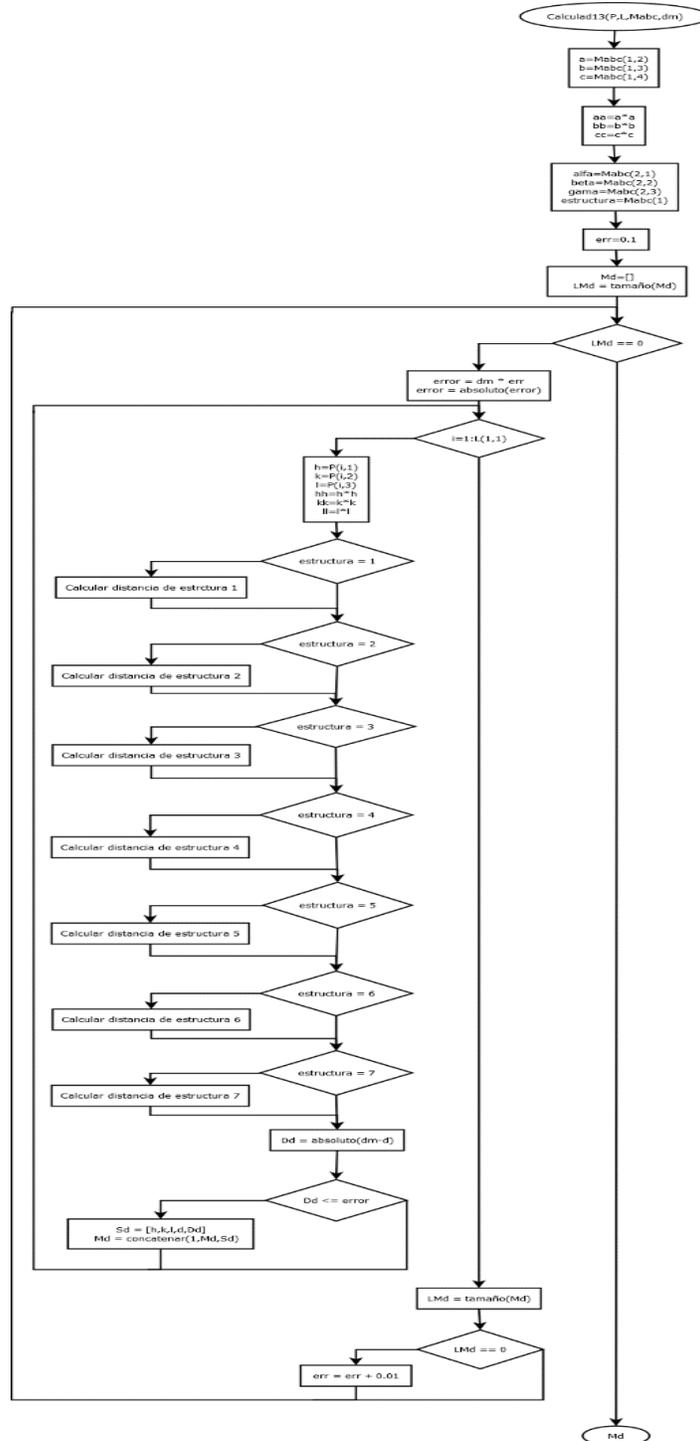


Diagrama de flujo 3 "Calculad13"

4.- Calculateta13: Se calcula el valor de teta entre los puntos 1 y 3, y dependiendo del tipo de estructura con la que se está trabajando se obtendrá el valor de teta.

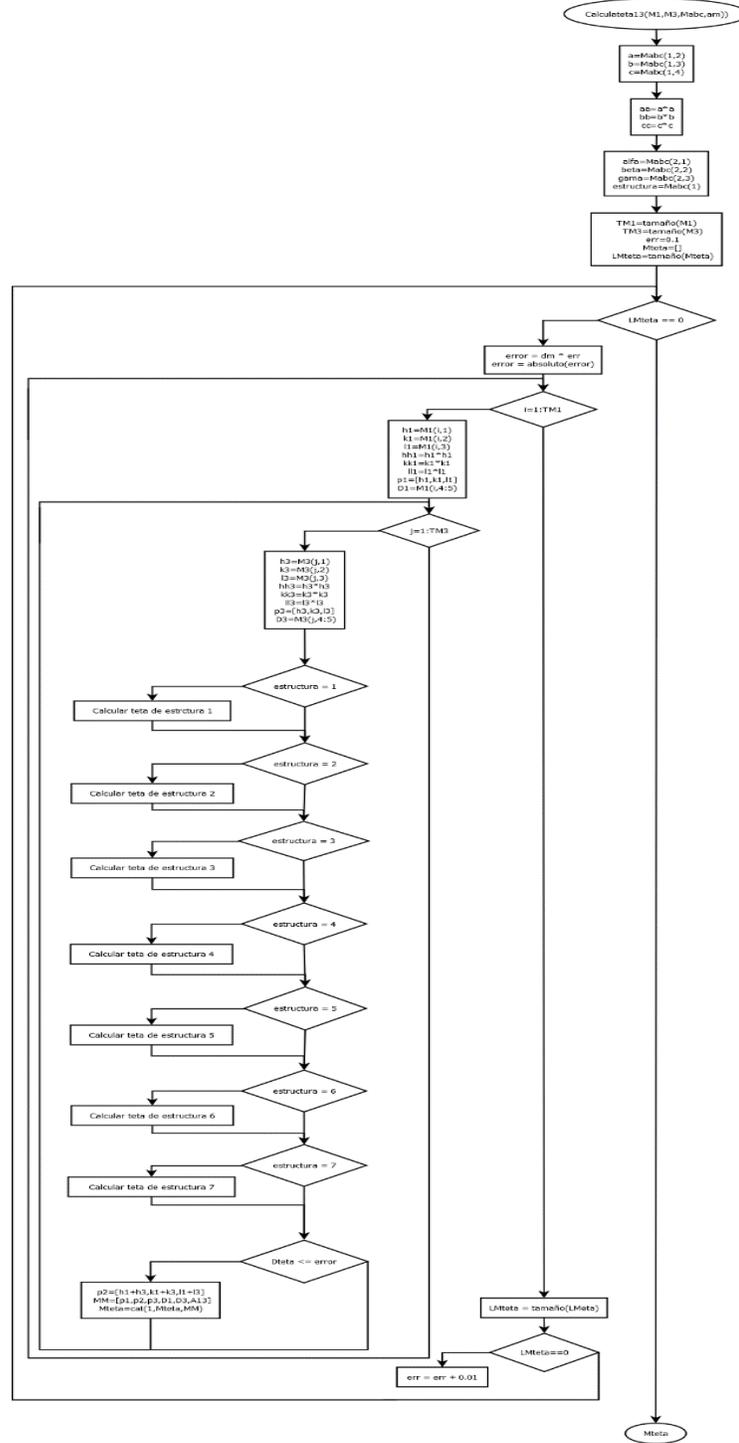


Diagrama de flujo 4 "Calculateta13".

5.- Calculad2: Se calcula la distancia entre dos puntos y dependiendo del tipo de estructura con la que se está trabajando se obtendrá su distancia.

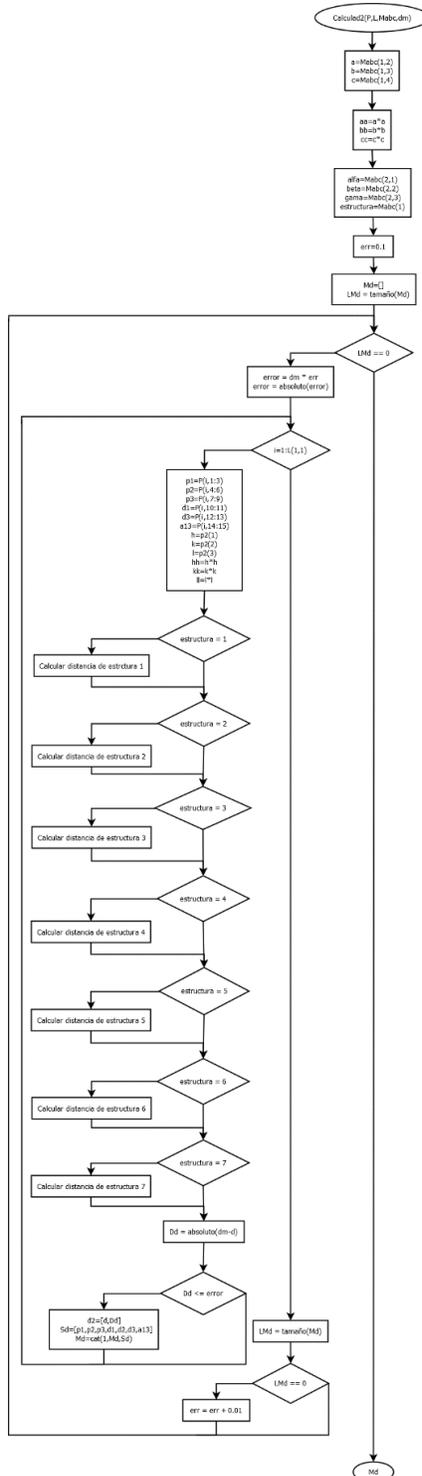


Diagrama de flujo 5 "Calculad2".

6.- Calculateta12: Se calcula el valor de teta entre los puntos 1 y 2, y dependiendo del tipo de estructura con la que se está trabajando se obtendrá el valor de teta.

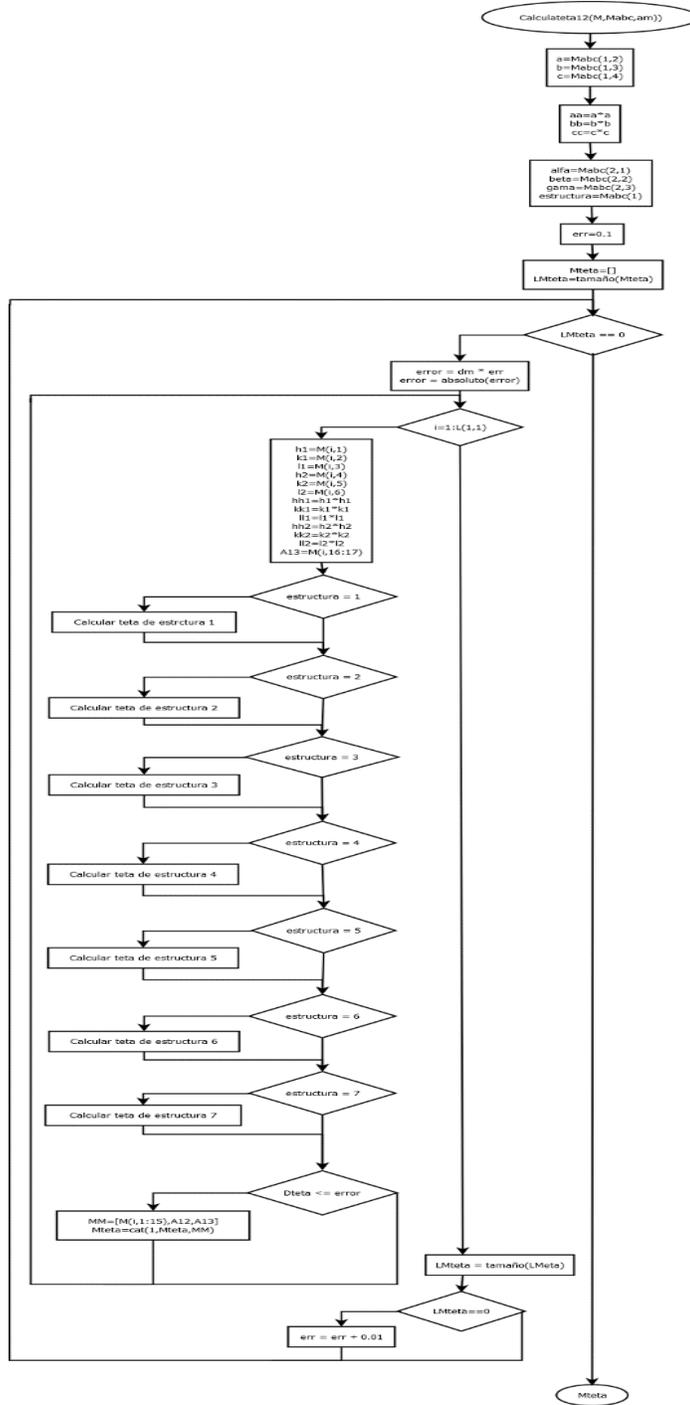


Diagrama de flujo 6 "Calculateta12".

7.- EjeZona: Se calcula las características del Eje de Zona con respecto a los puntos dados y se organizan en una matriz compuesta.

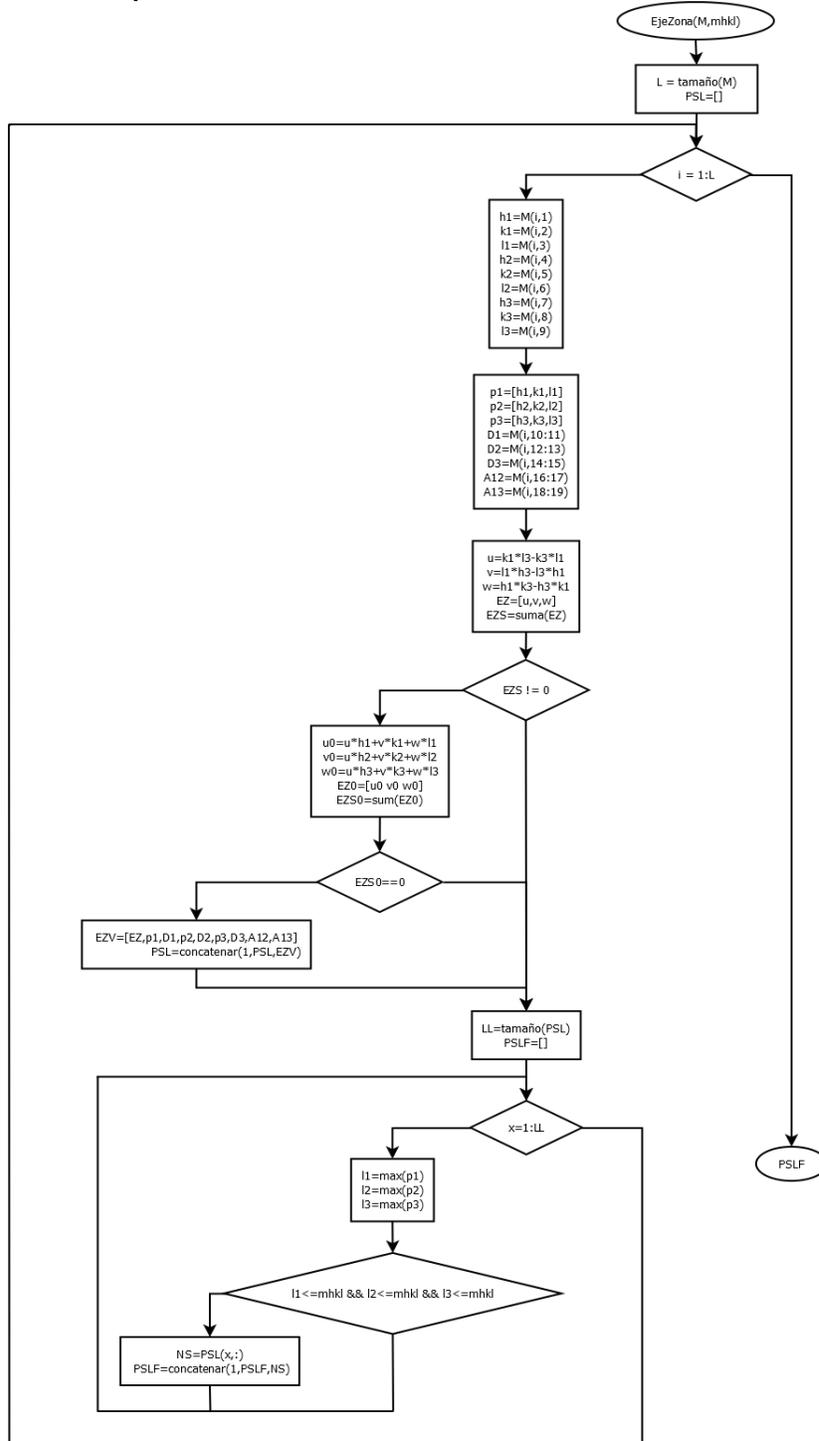


Diagrama de flujo 7 "EjeZona".

8.- Main: Programa principal del sistema PIPD-II al ejecutarse el proceso de "Calcular".

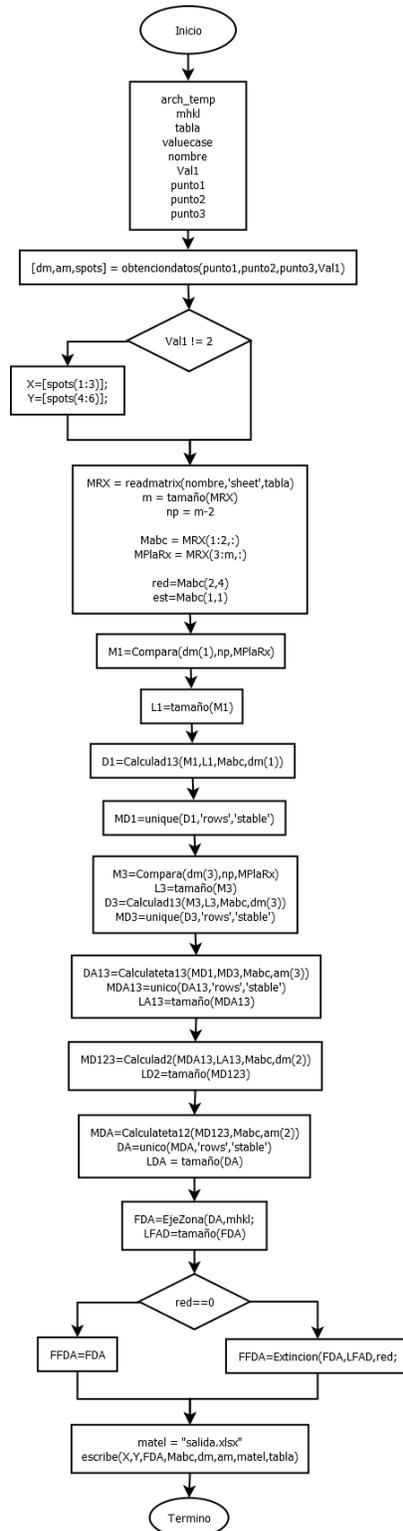


Diagrama de flujo 8 "Main".

## CAPÍTULO V

### **Desarrollo e implementación del programa para indexar patrones de difracción obtenidos a partir de un microscopio electrónico de transmisión**

Terminados los diagramas de flujo y después de comprobar que con esos procedimientos se llega a los resultados esperados respaldados con el material teórico, se codifican esos diagramas de flujo, en este caso, el lenguaje de programación elegido para realizar el programa fue Matlab, porque es el mejor lenguaje para realizar cálculos de precisión y ya que el uso de arreglos y matrices para indexar patrones de difracción es muy requerido, Matlab al ser un lenguaje de alto nivel y orientado a objetos, proporciona funciones matemáticas, de análisis y de investigación [11].

#### ***Inserción de campos e indexación***

Antes de iniciar el sistema se requiere tener información de material a trabajar como lo son los parámetros esenciales como puntos de trabajo, ángulos, distancias, y un valor máximo para hkl. Además, se contempla la selección del tipo de estructura y la inclusión de un porcentaje de error en los cálculos.

Esta pantalla permite al usuario ingresar los campos requeridos para comenzar el proceso, lo que incluye la dirección de la ubicación de la plantilla y el tipo de estructura con el que se va a trabajar el material. Una vez ingresados estos campos, se presiona el botón de indexar, lo que inicia el proceso de extracción de información del archivo Excel proporcionado como plantilla. Este proceso es crucial para obtener los datos necesarios para el cálculo y análisis de la estructura. Durante este proceso, el sistema extrae los puntos marcados, el tipo de estructura, su nombre y el porcentaje de error aceptado. Luego, extrae la información de las tablas proporcionadas en el Excel del material elegido y la almacena en memoria.

### ***Cálculos y ordenamiento***

Con esta información, calcula los ángulos y distancias entre los puntos, después, se hacen los cálculos correspondientes al tipo de estructura obtenido, con datos de la tabla de rayos X. Estos cálculos permiten obtener resultados precisos y confiables. Puede que se obtengan datos repetidos, por lo que se realiza un barrido para eliminar quienes incumplan el sesgo de error o estén duplicados. Una vez obtenida la información definitiva, se ordena según el sesgo de error, colocando en primer lugar aquel con el menor porcentaje de error.

Una vez completada la ejecución, el sistema creará automáticamente un archivo de Excel donde se presentarán los resultados. De esta manera, se facilitará al usuario una visualización clara y directa de los resultados obtenidos. Finalmente, se genera un archivo Excel de salida, que se descarga automáticamente en el computador y se abre para que el usuario pueda acceder a la información de inmediato. Este archivo contiene los resultados del cálculo y análisis.

### ***Sistema de carga y manejo de errores***

Durante el proceso, la pantalla muestra un sistema de carga que informa al usuario del estatus del proceso. En caso de error, se muestra un mensaje indicando que se debe revisar el archivo error.txt, que contiene una breve descripción del error causado para su análisis. Esto permite al usuario identificar y solucionar problemas de manera rápida y eficiente.

El usuario tiene la opción de cerrar el programa en cualquier momento mediante un botón, finalizando el proceso. El funcionamiento de la pantalla es un proceso complejo que requiere precisión y atención al detalle.

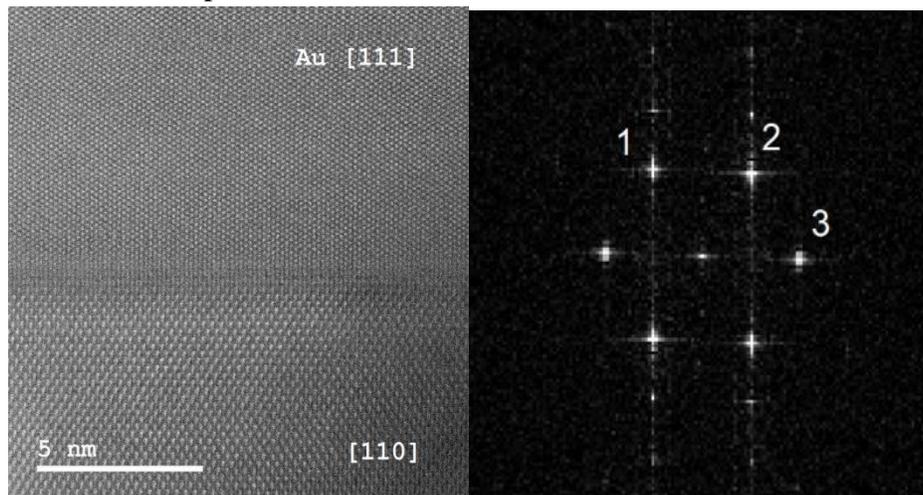
## CAPÍTULO VI

### Resultados (Ejemplo)

Hay que recordar que el Programa para Indexar Patrones de Difracción, PIPD-II nos va a proporcionar soluciones matemáticas a las ecuaciones utilizadas, y dado que el sistema de ecuaciones a resolver es no lineal, entonces nos encontraremos al final con una serie de soluciones, cada una de las matemáticamente válidas. Aquí es donde la experiencia y criterio del usuario lo lleva a seleccionar cuál es la mejor solución para el problema que está tratando de resolver. Todas las posibles soluciones se podrán ver en el archivo "salida.xls".

Veamos algunos ejemplos con resultados reales obtenidos con un microscopio electrónico de transmisión con aberración esférica corregida marca JEOL. En todos los casos trabajaremos con una imagen en alta resolución (puede verse la escala en el extremo izquierdo inferior de cada imagen), sobre la cual se ha seleccionado una pequeña zona cuadrada (cuadro rojo) de la cual se obtiene la FFT. El difractograma obtenido se muestra a la derecha de cada imagen y se indican los tres spots con los que se realizará la indexación.

1. Partícula de Au, correspondiente a un sistema cúbico FCC.



*Foto Caso 1. Nanopartícula de Au.*

Datos de los spots:

Spot	x	y
1	-3.47	-5.92
2	3.47	-5.72
3	7.15	0.2

Posibles resultados:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
1																								
2																								
3																								
4																								
5																								
6																								
7																								
8																								
9																								
10	Solución	u	v	w	h1	k1	l1	h2	k2	l2	h3	k3	l3	d1	Δd1	d2	Δd2	d3	Δd3	θ1	Δθ1	θ3	Δθ3	
11		1	-4	-4	4	0	2	2	-2	2	0	-2	0	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
12		2	4	-4	4	0	2	2	-2	0	2	-2	-2	0	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
13		3	-4	4	-4	0	2	2	2	2	0	2	0	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
14		4	4	4	-4	0	2	2	2	2	0	2	-2	0	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
15		5	4	4	-4	2	0	2	2	2	-2	0	0	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
16		6	4	-4	-4	2	0	2	0	-2	2	-2	-2	0	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
17		7	-4	4	4	2	0	2	2	2	0	0	2	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
18		8	-4	-4	4	2	0	2	0	2	2	-2	2	0	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
19		9	-4	4	-4	2	2	0	2	0	-2	0	-2	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
20		10	-4	4	4	2	2	0	0	2	-2	-2	0	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
21		11	4	-4	-4	2	2	0	2	0	2	0	-2	2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
22		12	4	-4	4	2	2	0	0	2	2	-2	0	2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
23		13	-4	-4	4	0	-2	-2	2	-2	0	2	0	2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
24		14	4	-4	4	0	-2	-2	2	0	-2	2	2	0	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
25		15	-4	4	-4	0	-2	-2	-2	-2	0	-2	0	2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
26		16	4	4	-4	0	-2	-2	-2	0	-2	-2	2	0	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
27		17	4	4	-4	-2	0	-2	-2	2	0	0	2	2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
28		18	4	-4	-4	-2	0	-2	0	2	-2	2	2	0	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789

Archivo "salida.xlsx" del caso 1.

2. Partícula de Au, correspondiente a un sistema cúbico FCC.

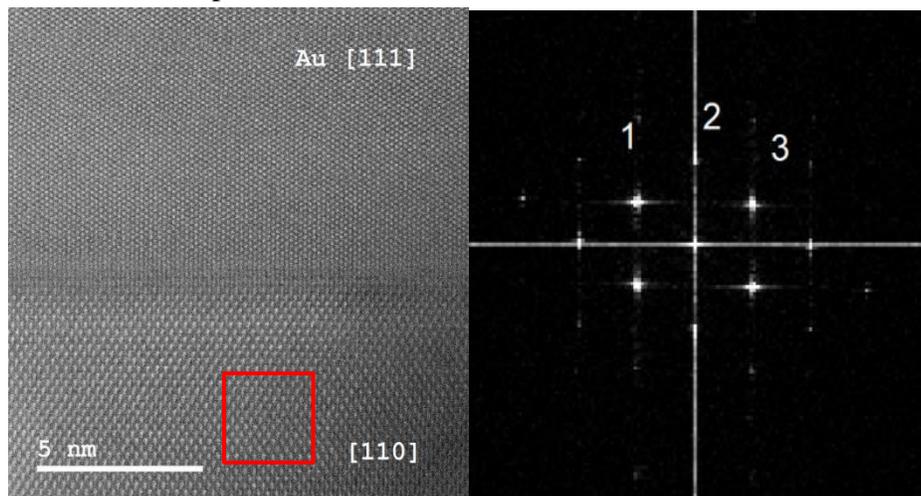


Foto Caso 2. Nanopartícula de Au.

Datos de los spots:

Spot	x	y
1	-3.47	-2.45
2	0	-4.9
3	3.47	-2.45

Posibles resultados:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
1																								
2																								
3																x	y	d medi	$\theta$ medido		a,b,c	$\alpha,\beta,\gamma$		
4																Punto 1	-3.47	-2.45	2.3542	54.775		4.068	89.99878	
5																Punto 2	0	-4.9	2.0408	54.775		4.068	89.99878	
6																Punto 3	3.47	-2.45	2.3542	109.55		4.068	89.99878	
7																								
8																								
9																								
10	Solución	u	v	w	h1	k1	l1	h2	k2	l2	h3	k3	l3	d1	$\Delta d1$	d2	$\Delta d2$	d3	$\Delta d3$	$\theta 1$	$\Delta \theta 1$	$\theta 3$	$\Delta \theta 3$	
11		1	0	-2	-2	1	1	-1	2	0	0	1	-1	1	2.3487	0.0055	2.034	0.0055	2.3487	0.0055	54.7349	0.0403	109.4697	0.0806
12		2	2	0	2	1	1	-1	0	2	0	-1	1	1	2.3487	0.0055	2.034	0.0055	2.3487	0.0055	54.7349	0.0403	109.4697	0.0806
13		3	0	2	2	1	-1	1	2	0	0	1	1	-1	2.3487	0.0055	2.034	0.0055	2.3487	0.0055	54.7349	0.0403	109.4697	0.0806
14		4	-2	-2	0	1	-1	1	0	0	2	-1	1	1	2.3487	0.0055	2.034	0.0055	2.3487	0.0055	54.7349	0.0403	109.4697	0.0806
15		5	-2	0	-2	-1	1	1	0	2	0	1	1	-1	2.3487	0.0055	2.034	0.0055	2.3487	0.0055	54.7349	0.0403	109.4697	0.0806
16		6	2	2	0	-1	1	1	0	0	2	1	-1	1	2.3487	0.0055	2.034	0.0055	2.3487	0.0055	54.7349	0.0403	109.4697	0.0806
17		7	2	2	0	1	-1	-1	0	0	-2	-1	1	-1	2.3487	0.0055	2.034	0.0055	2.3487	0.0055	54.7349	0.0403	109.4697	0.0806
18		8	-2	0	-2	1	-1	-1	0	-2	0	-1	-1	1	2.3487	0.0055	2.034	0.0055	2.3487	0.0055	54.7349	0.0403	109.4697	0.0806
19		9	-2	-2	0	-1	1	-1	0	0	-2	1	-1	-1	2.3487	0.0055	2.034	0.0055	2.3487	0.0055	54.7349	0.0403	109.4697	0.0806
20		10	0	2	2	-1	1	-1	-2	0	0	-1	-1	1	2.3487	0.0055	2.034	0.0055	2.3487	0.0055	54.7349	0.0403	109.4697	0.0806
21		11	2	0	2	-1	-1	1	0	-2	0	1	-1	-1	2.3487	0.0055	2.034	0.0055	2.3487	0.0055	54.7349	0.0403	109.4697	0.0806
22		12	0	-2	-2	-1	-1	1	-2	0	0	-1	1	-1	2.3487	0.0055	2.034	0.0055	2.3487	0.0055	54.7349	0.0403	109.4697	0.0806
23		13	-4	-4	4	0	-2	-2	2	-2	0	2	0	2	1.4383	0.019	1.438	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
24		14	4	-4	4	0	-2	-2	2	0	-2	2	2	0	1.4383	0.019	1.438	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
25		15	-4	4	-4	0	-2	-2	-2	0	-2	0	-2	0	1.4383	0.019	1.438	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
26		16	4	4	-4	0	-2	-2	-2	0	-2	-2	2	0	1.4383	0.019	1.438	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
27		17	4	4	-4	-2	0	-2	-2	2	0	0	2	2	1.4383	0.019	1.438	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789
28		18	4	-4	-4	-2	0	-2	0	2	-2	2	2	0	1.4383	0.019	1.438	0.019	1.4383	0.0402	59.9992	0.3594	119.9984	1.9789

Archivo "salida.xlsx" del caso 2.

3. Partícula de Au, correspondiente a un sistema cúbico FCC.

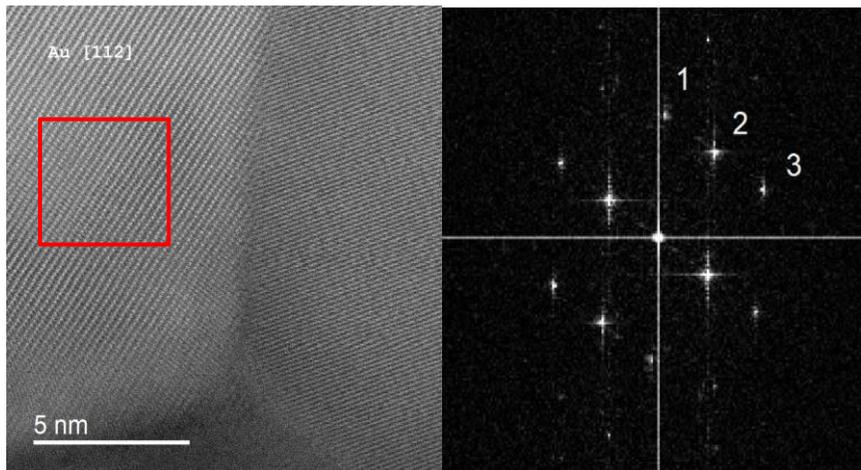


Foto Caso 3. Nanopartícula de Au.

Datos de los spots:

Spot	x	y
1	0.47	-8.31
2	4.08	-5.8
3	7.69	-3.29

Posibles resultados:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
1																								
2																								
3																								
4																								
5																								
6																								
7																								
8																								
9																								
10	Solución	u	v	w	h1	k1	l1	h2	k2	l2	h3	k3	l3	d1	Δd1	d2	Δd2	d3	Δd3	θ1	Δθ1	θ3	Δθ3	
11		1	-8	-4	4	1	1	3	0	4	4	-1	3	1	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
12		2	4	8	-4	1	1	3	4	0	4	3	-1	1	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
13		3	8	-4	4	1	3	1	0	4	4	-1	1	3	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
14		4	-4	4	-8	1	3	1	4	4	0	3	1	-1	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
15		5	4	-8	-4	3	1	1	4	0	4	1	-1	3	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
16		6	-4	4	8	3	1	1	4	4	0	1	3	-1	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
17		7	-8	-4	4	-1	-1	-3	0	-4	-4	1	-3	-1	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
18		8	4	8	-4	-1	-1	-3	-4	0	-4	-3	1	-1	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
19		9	8	-4	4	-1	-3	-1	0	-4	-4	1	-1	-3	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
20		10	-4	4	-8	-1	-3	-1	-4	-4	0	-3	-1	1	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
21		11	4	-8	-4	-3	-1	-1	-4	0	-4	-1	1	-3	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
22		12	-4	4	8	-3	-1	-1	-4	-4	0	-1	-3	1	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
23		13	8	4	4	1	1	-3	0	4	-4	-1	3	-1	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
24		14	-4	-8	-4	1	1	-3	4	0	-4	3	-1	-1	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
25		15	-8	-4	-4	1	-3	1	0	-4	4	-1	-1	3	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
26		16	4	4	8	1	-3	1	4	-4	0	3	-1	-1	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
27		17	4	8	4	-3	1	1	-4	0	4	-1	-1	3	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636
28		18	-4	-4	-8	-3	1	1	-4	4	0	-1	3	-1	1.2265	0.0251	0.7191	0.0251	1.2265	0.031	31.4817	0.2308	62.9635	0.636

Archivo "salida.xlsx" del caso 3.

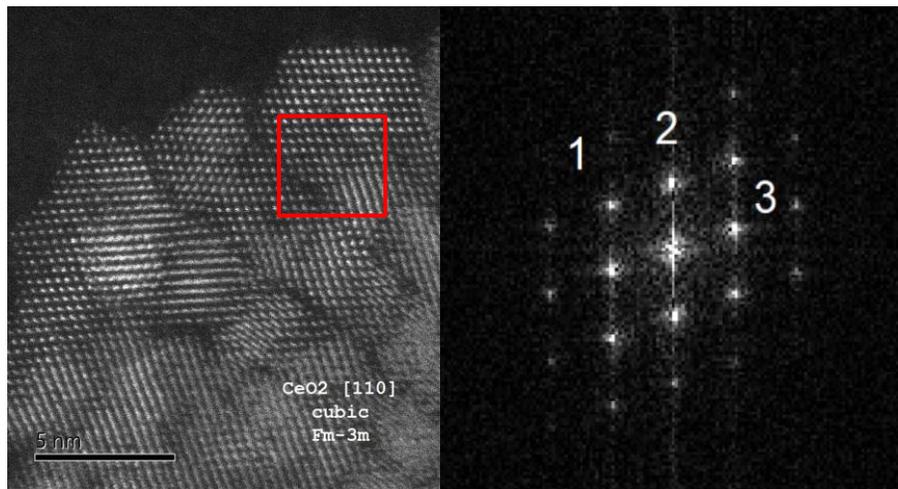
4. Partícula de Óxido de Cerio, CeO<sub>2</sub>, sistema cúbico FCC.

Foto Caso 4. Nanopartícula de CeO<sub>2</sub>.

Datos de los spots:

Spot	x	y
1	-3.03	-2.12
2	-0.15	-3.18
3	3.03	-1.06

Posibles resultados:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1																							
2																							
3																							
4																	x	y	d med	$\theta$ medido	a,b,c	$\alpha,\beta,\gamma$	
5																	Punto 1	-3.03	-2.12	2.7042	52.319	5.40972	89.99878
6																	Punto 2	-0.15	-3.18	3.1412	73.418	5.40972	89.99878
7																	Punto 3	3.03	-1.06	3.1152	125.74	5.40972	89.99878
8																							
9																							
10	Solución	u	v	w	h1	k1	l1	h2	k2	l2	h3	k3	l3	d1	$\Delta d1$	d2	$\Delta d2$	d3	$\Delta d3$	$\theta 1$	$\Delta \theta 1$	$\theta 3$	$\Delta \theta 3$
11		1	2	2	0	0	2	1	-1	1	1	-1	-1	2.7049	0.0007	3.1233	0.0007	3.1233	0.0081	54.7349	18.6831	125.2627	0.4747
12		2	-2	-2	0	0	2	-1	1	1	-1	1	-1	2.7049	0.0007	3.1233	0.0007	3.1233	0.0081	54.7349	18.6831	125.2627	0.4747
13		3	-2	0	-2	0	2	0	1	1	-1	1	-1	2.7049	0.0007	3.1233	0.0007	3.1233	0.0081	54.7349	18.6831	125.2627	0.4747
14		4	2	0	2	0	2	0	-1	1	1	-1	-1	2.7049	0.0007	3.1233	0.0007	3.1233	0.0081	54.7349	18.6831	125.2627	0.4747
15		5	0	2	2	0	0	1	1	-1	-1	1	-1	2.7049	0.0007	3.1233	0.0007	3.1233	0.0081	54.7349	18.6831	125.2627	0.4747
16		6	0	-2	-2	0	0	1	-1	1	-1	-1	1	2.7049	0.0007	3.1233	0.0007	3.1233	0.0081	54.7349	18.6831	125.2627	0.4747
17		7	-2	-2	0	0	0	-2	1	-1	-1	1	-1	2.7049	0.0007	3.1233	0.0007	3.1233	0.0081	54.7349	18.6831	125.2627	0.4747
18		8	2	2	0	0	0	-2	-1	1	-1	-1	1	2.7049	0.0007	3.1233	0.0007	3.1233	0.0081	54.7349	18.6831	125.2627	0.4747
19		9	2	0	2	0	-2	0	1	-1	-1	1	1	2.7049	0.0007	3.1233	0.0007	3.1233	0.0081	54.7349	18.6831	125.2627	0.4747
20		10	-2	0	-2	0	-2	0	-1	-1	1	-1	1	2.7049	0.0007	3.1233	0.0007	3.1233	0.0081	54.7349	18.6831	125.2627	0.4747
21		11	0	-2	-2	0	0	-1	1	-1	1	1	-1	2.7049	0.0007	3.1233	0.0007	3.1233	0.0081	54.7349	18.6831	125.2627	0.4747
22		12	0	2	2	-2	0	0	-1	-1	1	1	-1	2.7049	0.0007	3.1233	0.0007	3.1233	0.0081	54.7349	18.6831	125.2627	0.4747
23																							
24																							
25																							
26																							
27																							
28																							

Archivo "salida.xlsx" del caso 4.

5. Película de MoS2 correspondiente a un sistema hexagonal.

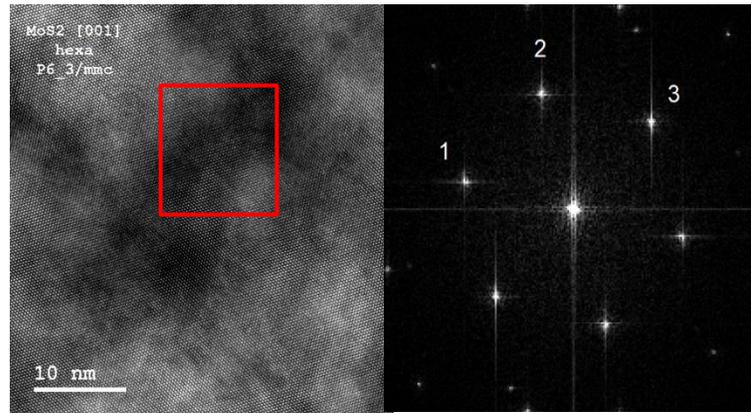


Foto Caso 5. Película de MoS2.

Datos de los spots:

Spot	x	y
1	-3.563	-0.862
2	-1.034	-3.505
3	2.528	-2.643

Posibles resultados:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W		
1																									
2																									
3																									
4																									
5																									
6																									
7																									
8																									
9																									
10	Solución	u	v	w	h1	k1	l1	h2	k2	l2	h3	k3	l3	d1	Δd1	d2	Δd2	d3	Δd3	θ1	Δθ1	θ3	Δθ3		
11		1	0	0	1	0	1	0	-1	1	0	-1	0	0	2.7377	0.0098	2.7377	0.0098	2.7377	0.0035	59.9992	0.1624	119.9984	0.1257	
12		2	0	0	-1	0	1	0	0	1	0	0	-1	-1	0	2.7377	0.0098	2.7377	0.0098	2.7377	0.0035	59.9992	0.1624	119.9984	0.1257
13		3	1	0	1	0	1	0	-1	1	1	-1	0	1	2.7377	0.0098	2.6723	0.0098	2.6723	0.0619	60.7864	0.6248	119.2111	0.9129	
14		4	-2	0	1	0	1	0	-1	-2	-1	0	-2	0	2.7377	0.0098	2.501	0.0098	2.501	0.2332	58.2108	1.9508	121.7868	1.6627	
15		5	2	0	1	0	1	0	-1	1	2	-1	0	2	2.7377	0.0098	2.501	0.0098	2.501	0.2332	61.5144	1.3528	118.4832	1.6409	
16		6	0	0	-1	1	0	0	-1	1	0	0	-1	0	2.7377	0.0098	2.7377	0.0098	2.7377	0.0035	59.9992	0.1624	119.9984	0.1257	
17		7	0	-1	-1	1	0	0	1	-1	1	0	-1	1	2.7377	0.0098	2.6723	0.0098	2.6723	0.0619	60.7864	0.6248	119.2111	0.9129	
18		8	0	0	1	1	0	0	0	1	0	-1	1	0	2.7377	0.0098	2.7377	0.0098	2.7377	0.0035	59.9992	0.1624	119.9984	0.1257	
19		9	0	-2	-1	1	0	0	1	-2	0	-1	-2	0	-1	2.7377	0.0098	2.501	0.0098	2.501	0.2332	58.2108	1.9508	121.7868	1.6627
20		10	0	-2	-1	1	0	0	1	-2	0	-1	2	0	-1	2.7377	0.0098	2.501	0.0098	2.501	0.2332	61.5144	1.3528	118.4832	1.6409
21		11	0	0	1	0	-1	0	-1	1	0	1	0	0	2.7377	0.0098	2.7377	0.0098	2.7377	0.0035	59.9992	0.1624	119.9984	0.1257	
22		12	1	0	1	0	-1	0	1	-1	-1	1	0	-1	2.7377	0.0098	2.6723	0.0098	2.6723	0.0619	59.1441	1.0175	120.8534	0.7294	
23		13	0	0	-1	0	-1	0	-1	0	-1	1	0	2.7377	0.0098	2.7377	0.0098	2.7377	0.0035	59.9992	0.1624	119.9984	0.1257		
24		14	-2	0	1	0	-1	0	-1	-2	1	0	2	2.7377	0.0098	2.501	0.0098	2.501	0.2332	61.5144	1.3528	118.4832	1.6409		
25		15	2	0	1	0	-1	0	1	-2	1	0	-2	0	2.7377	0.0098	2.501	0.0098	2.501	0.2332	58.2108	1.9508	121.7868	1.6627	
26		16	0	0	-1	0	0	-1	1	0	0	1	0	2.7377	0.0098	2.7377	0.0098	2.7377	0.0035	59.9992	0.1624	119.9984	0.1257		
27		17	0	0	1	-1	0	0	-1	0	1	-1	0	2.7377	0.0098	2.7377	0.0098	2.7377	0.0035	59.9992	0.1624	119.9984	0.1257		
28		18	0	-1	-1	0	0	-1	1	-1	0	1	-1	0	-1	2.7377	0.0098	2.6723	0.0098	2.6723	0.0619	59.1441	1.0175	120.8534	0.7294
	<	>	ZnO	MoS2	CeO2	Au	+																		

Archivo "salida.xlsx" del caso 5.

6. Nanopartícula de ZnO de sistema hexagonal

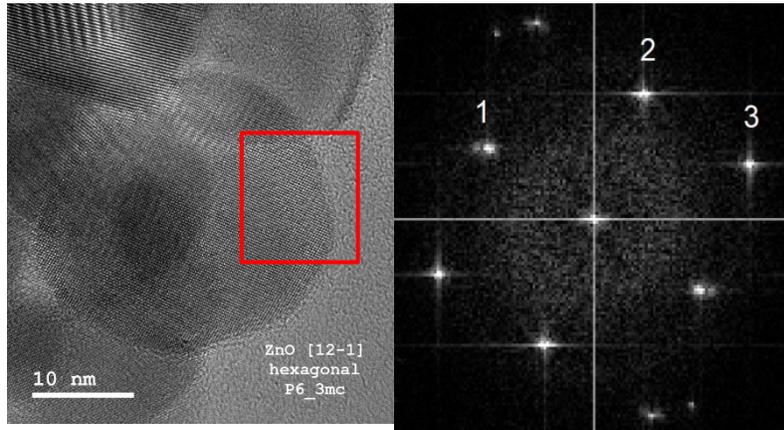


Foto Caso 6. Nanopartícula de ZnO.

Datos de los spots:

Spot	x	y
1	-3.393	-2.106
2	1.521	-3.696
3	4.915	-1.58

Posibles resultados:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W		
1																									
2																									
3																									
4																x	y	d medido	θ medido		a,b,c	α,β,γ			
5																Punto	-3.39	-2.11	2.5041	80.54		3.2489	89.9988		
6																Punto	1.521	-3.7	2.502	49.81		3.2489	89.9988		
7																Punto	4.915	-1.58	1.937	130.35		5.2049	119.976		
8																									
9																									
10	Solución	u	v	w	h1	k1	l1	h2	k2	l2	h3	k3	l3	d1	Δd1	d2	Δd2	d3	Δd3	θ1	Δθ1	θ3	Δθ3		
11		1	0	-3	0	-1	0	-1	0	0	-3	1	0	-2	2.4751	0.029	1.735	0.029	1.9105	0.0265	89.9988	68.0754	140.016	9.666	
12		2	3	0	0	0	-1	-1	0	0	-3	0	1	-2	2.4751	0.029	1.735	0.029	1.9105	0.0265	89.9988	68.0754	140.016	9.666	
13		3	0	3	0	1	0	-1	0	0	-3	-1	0	-2	2.4751	0.029	1.735	0.029	1.9105	0.0265	89.9988	68.0754	140.016	9.666	
14		4	-3	0	0	0	1	-1	0	0	-3	0	-1	-2	2.4751	0.029	1.735	0.029	1.9105	0.0265	89.9988	68.0754	140.016	9.666	
15																									
16																									
17																									
18																									
19																									
20																									
21																									
22																									
23																									
24																									
25																									
26																									
27																									
28																									
	<	>														ZnO	MoS2	CeO2	Au	+					

Archivo "salida.xlsx" del caso 6.

## CAPÍTULO VII

### Conclusiones

Una de las herramientas más poderosas con la que cuenta la microscopía electrónica, es la difracción de electrones. En el campo de la Ciencia de Materiales la difracción es usada como una herramienta standard y su uso ha proporcionado una enorme cantidad de información acerca de la estructura de los materiales.

A cada valor de la distancia interplanar  $d$  se le puede asignar un juego  $(hkl)$  de valores enteros llamados índices de Miller, dependiendo del tipo de estructura cristalina y su orientación  $(uvw)$  (eje de zona) respecto al eje óptico del instrumento. Identificar los valores del vector que corresponde a cada punto en un patrón de difracción se conoce como indexación de un patrón de difracción [13]. Científicos y técnicos trabajando en el ramo de la cristalografía han hecho tablas de las distancias interplanares  $d$ 's para cada elemento y compuesto conocido hasta la fecha. Estas tablas se han elaborado generalmente usando Tablas de Rayos X y se encuentran publicadas en varias partes, por ejemplo, las que publica la Unión Internacional de Cristalografía.

Un usuario puede consultar estas tablas y ver si los valores de  $d$  calculados corresponden a los que él está analizando, y bien podría dar por bueno los valores  $(hkl)$  que le proponen como solución. De no ser el caso, habrá que hacer cálculos, pero para la gran mayoría de usuarios, el procedimiento resulta ser muy tedioso, lento y siempre está sujeto a errores en las mediciones y los cálculos que hay que hacer, y sólo se simplifican un poco si sabe o se tiene una idea del material del que se trata, de lo contrario es muy complicado hacer estos cálculos.

La bondad del método que nosotros usamos en este trabajo radica principalmente en la confianza en que siempre tenemos una idea de qué material estamos analizando, de que nuestros resultados fueron obtenidos en un HTEM con buena calibración y la resolución suficiente para poder hacer nuestras mediciones. En nuestro caso y con nuestro microscopio, partimos del hecho de que parámetros como la longitud de onda, definida por el voltaje, es exactamente de 200 KeV lo cual corresponde a una  $\lambda=0.00251\text{nm}$ , y a una longitud de cámara (véase Figura 6) de  $L=80\text{mm}$ , y que, los valores de desenfoque, amplificación, aberración esférica (muy importante en los cálculos), son todos

conocidos y fijos; que la platina del porta muestras está perfectamente perpendicular al haz de incidencia de la fuente y que la superficie de la muestra es totalmente plana.

Obviamente al hacer estas suposiciones, estamos conscientes que estamos aceptando pequeños errores, lo que posteriormente deberemos tomar en cuenta en la interpretación de nuestros resultados. Y es aquí donde la experiencia del usuario es de gran ayuda, por ejemplo, un experto al observar el patrón de difracción lo primero que le vendrá a la mente es, si se tiene:

- un arreglo cuadrado de puntos, con ángulos aparentes entre puntos de  $90^\circ$ , indica que se trata de un material con estructura cúbica u ortorrómbica o tetragonal.
- un arreglo hexagonal de puntos, con ángulos aparentes entre lados de  $60^\circ$ , indica que se trata de un material con estructura hexagonal o cúbica.
- un arreglo asimétrico de puntos, indican que se trata de un material con estructura monoclinica o triclinica.

Sin embargo, la vista puede engañarnos, los patrones de difracción se forman en el espacio, dicho de otra manera, en tres dimensiones, nosotros al sólo analizar una imagen de dicho patrón, lo hacemos en un plano, es decir, en dos dimensiones.

Por tal razón, para hacer los cálculos y el análisis de nuestra imagen, el desarrollo y la implantación del programa propuesto, representa un avance significativo para aquellos que basan su trabajo en el campo de la cristalografía y el estudio de la materia condensada.

Por todo lo anterior, podemos concluir que:

- El objetivo principal de diseñar una aplicación de escritorio amigable, que proporcione soluciones matemáticas basadas en la teoría para la indexación de patrones de difracción obtenidos mediante microscopios electrónicos de transmisión de alta resolución [1], ha sido abordado con éxito.
- El programa se centra en proporcionar una herramienta versátil y accesible que supera las limitaciones del que está en uso actualmente [7].
- La portabilidad del software asegura que pueda ser instalado en cualquier máquina de escritorio, eliminando la dependencia de equipos específicos y promoviendo un acceso más amplio entre los investigadores

- El desarrollo en Matlab para la codificación y para la interfaz gráfica ha sido clave para ofrecer una herramienta eficiente y amigable para los potenciales usuarios ya que es un lenguaje que se utiliza mucho en problemas que involucran una gran cantidad de matrices y vectores.
- Se proporciona un software que agiliza el procedimiento matemático y permite una rápida indexación de patrones de difracción, el programa no solo mejora la eficiencia en la investigación, sino que también facilita la compartición de datos y resultados al tener plantillas en Excel ya que es un programa universal y que todo usuario de un HTEM podría utilizar.
- Se incluye un manual que facilita la instalación y manejo del programa, así como la interpretación de los resultados, el cual se puede ver en Anexo I de este trabajo.

Sabemos que, con el uso, sobre todo por los estudiantes (que son los que lo usarán con mayor frecuencia), se hará necesario hacerle correcciones, adecuaciones o ampliaciones a este programa. Una recomendación a futuro es incluir en el sistema una alerta sobre posibles desajustes en el microscopio electrónico, ya que con los cálculos adecuados [1], el programa puede ayudar a los investigadores a identificar y corregir errores de calibración en el HTEM, lo que aumenta su utilidad. En resumen, la implantación de este programa no solo optimiza el análisis de datos en cristalografía, sino que también ofrece una herramienta flexible y accesible que puede adaptarse y evolucionar con los avances futuros en la materia. La combinación de una interfaz intuitiva, la capacidad de operar en diferentes equipos y el soporte para una amplia gama de estructuras cristalinas representa un avance significativo para la comunidad científica en la Cristalografía, facilitando el trabajo de investigadores y mejorando la calidad de los resultados obtenidos.

## **Anexo I**

### **PIPD – II**

#### **Manual de uso**

En este manual se muestran los detalles sobre los pasos a seguir como usuario del programa "PIPD – II". El usuario encontrará las características de cada sección del programa y se mostrarán los archivos a usar, imágenes ilustrativas del uso del programa, etc.

Este programa representa una propuesta para resolver el problema de indexar los patrones de difracción obtenidos directamente con el microscopio electrónico de transmisión o a través de la FFT (transformada rápida de Fourier) de una zona selecta de una imagen obtenida con un HTEM. Parte de la teoría para la interpretación de los patrones de difracción [1] y se apoya en las tablas de Rayos X, otorgando al usuario una pequeña lista de posibles soluciones matemáticas a partir de los índices de Miller de los planos analizados y el eje de zona correspondiente, dejando al usuario y a su experiencia, la selección de la mejor o más apropiada solución.

#### **Requerimientos**

- Computadora con Sistema Operativo Windows 10 o superior / MacOS Monterey.
- Procesador Core i3/AMD FX 2.4 [GHz] o superior.
- Memoria RAM de 4 [GB].
- 10 GB de espacio disponible de almacenamiento.
- Runtime R2022b (9.13) o superior de MATLAB.
- Descompresor de archivos RAR/ZIP.
- Tablas de Rayos X (archivo descargable).

#### **INSTALACIÓN**

Si al tratar de cargar los archivos de este programa no le aparecen en la pantalla en forma de libros apilados, sino aparecen como un archivo que no permite su lectura, entonces, deberá usar WinRAR para poder descomprimir el archivo.

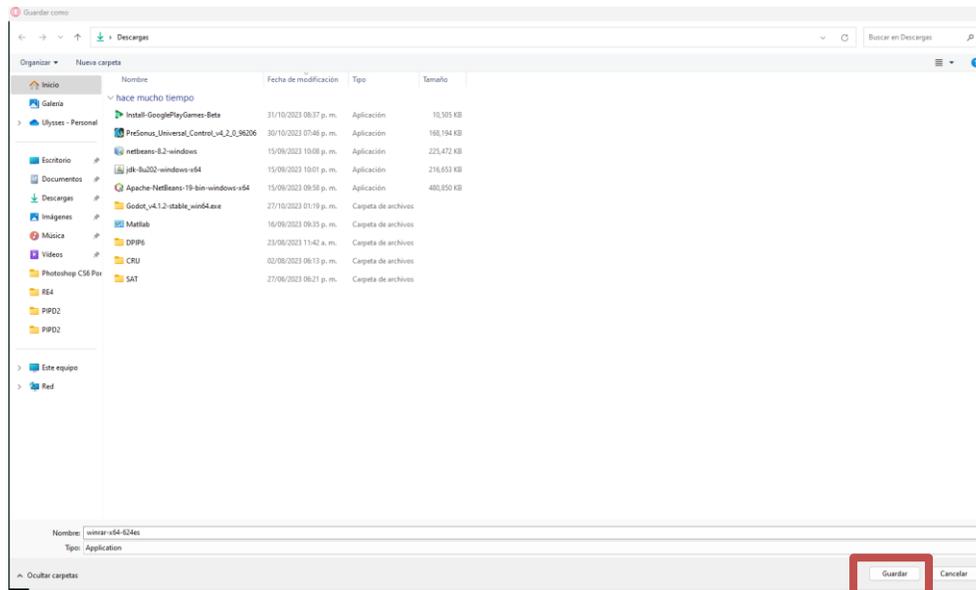
## Instalación de WinRAR

Para poder instalar WinRAR es necesario ir a la página: <https://www.winrar.es/descargas>, para posteriormente dar clic en el recuadro verde que dice "Descargue WinRAR ahora".

The screenshot shows the WinRAR website's download page. The navigation bar includes 'Inicio', 'Descargas', 'Comprar', 'Soporte', and 'Noticias'. A search bar is present on the right. The main content area is titled 'Descarga gratuita recomendada' and features a prominent green button labeled 'Descargue WinRAR ahora' with a dropdown arrow. Below this, there is a section for 'Aviso de nuevas versiones' and a list of 'Ultimas descargas' with various WinRAR versions and languages. A red arrow points to the green button.

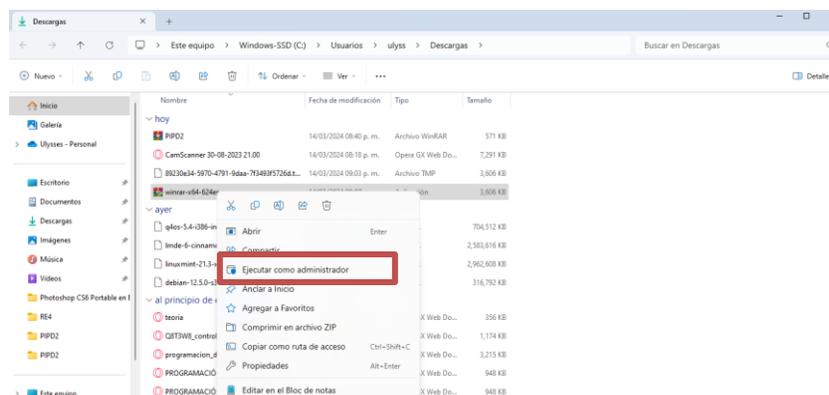
*Página principal de descarga de WinRAR.*

Se abrirá una ventana para seleccionar en dónde descargar el archivo, escriba entonces el nombre de la carpeta donde quiera bajarlo o acepte la acción general donde automáticamente lo abrirá en la carpeta de *Descargas*, posteriormente no hay más que dar clic en "guardar" o sólo dar ENTER.



*Opción de guardar ejecutable de WinRAR.*

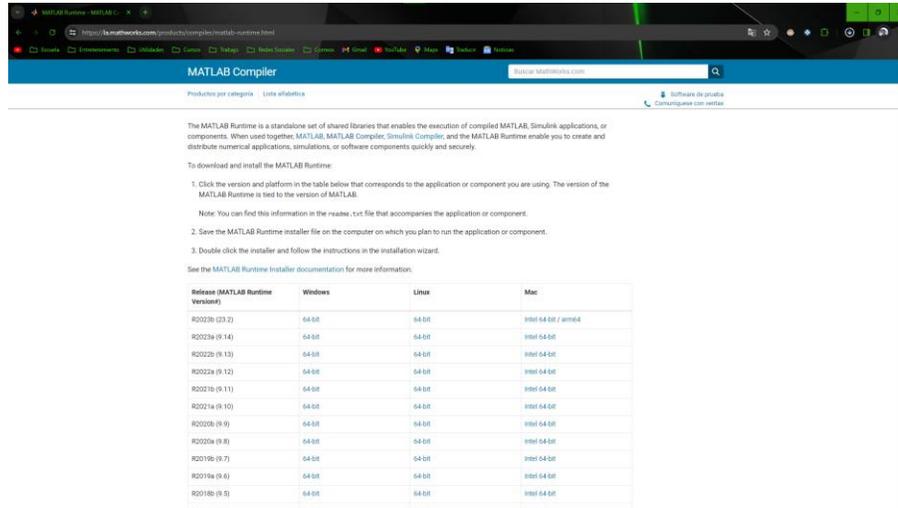
Al entrar a WinRAR, observará que se cuenta con un archivo ejecutable, ejecútelo como administrador dando clic derecho en el archivo ejecutable de WinRAR, opte por la opción de *aceptar*, e inicie la instalación. Si tiene dudas, WinRAR cuenta con su soporte oficial en: <https://www.winrar.es/descargas/103/descargar-winrar-para-windows-x64-en-espanol>.



*Opción de ejecución como administrador.*

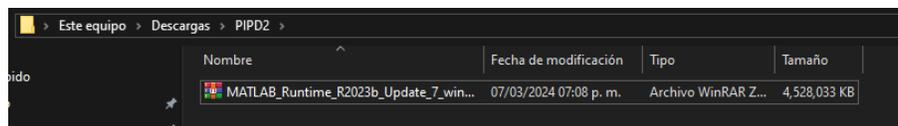
## Instalación de Runtime de MATLAB

Inicie la instalación ingresando a <https://la.mathworks.com/products/compiler/matlab-runtime.html>



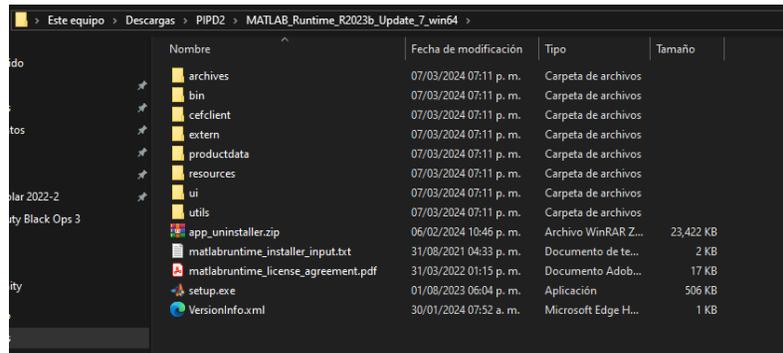
*Página principal de descarga de Runtime de Matlab.*

Seleccione la última versión en el sistema operativo con el que esté trabajando (Windows o Mac) y espere la descarga.



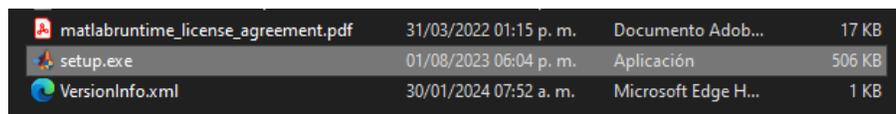
*Archivo comprimido del Runtime de Matlab.*

Una vez descargado, descomprima en la carpeta de su elección el contenido del archivo.



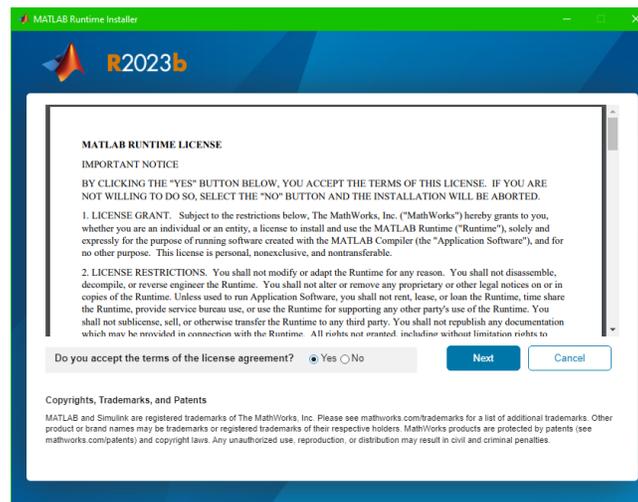
*Contenido del archivo descomprimido del Runtime de Matlab.*

Dentro de la carpeta, ejecute en modo *administrador* dando clic derecho sobre el programa Setup.exe y "Ejecutar como administrador".



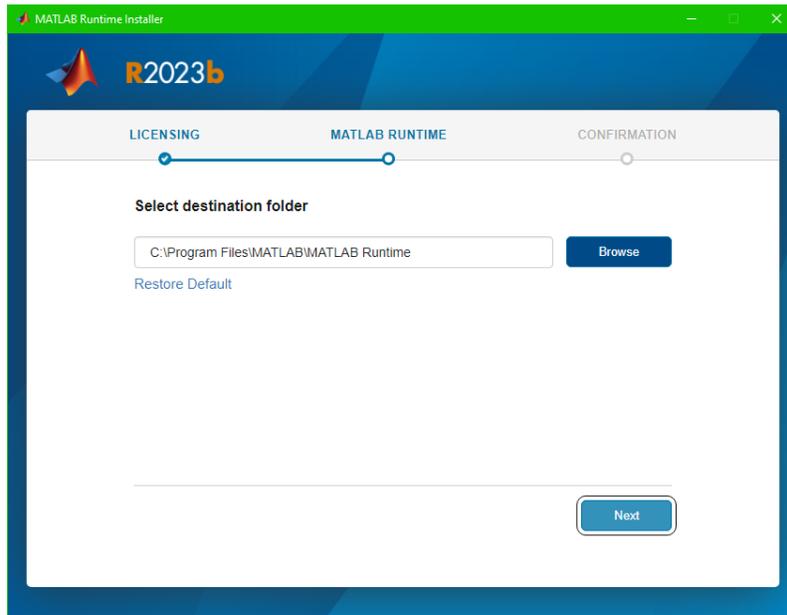
*Instalador de Runtime de Matlab.*

El proceso de instalación iniciará una vez que Usted acepte la licencia con los términos y condiciones que ahí se indican.



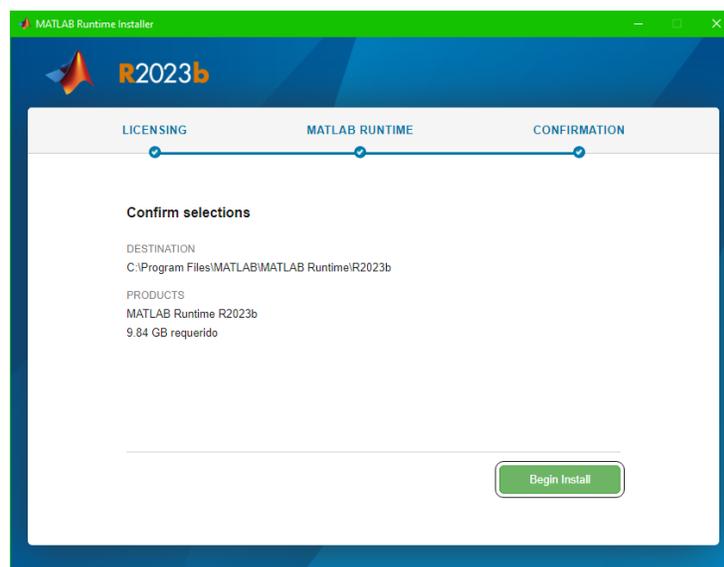
*Página inicial de "términos y condiciones" del Runtime de Matlab.*

Seleccione la ubicación de instalación del programa (se recomienda el que viene por defecto).



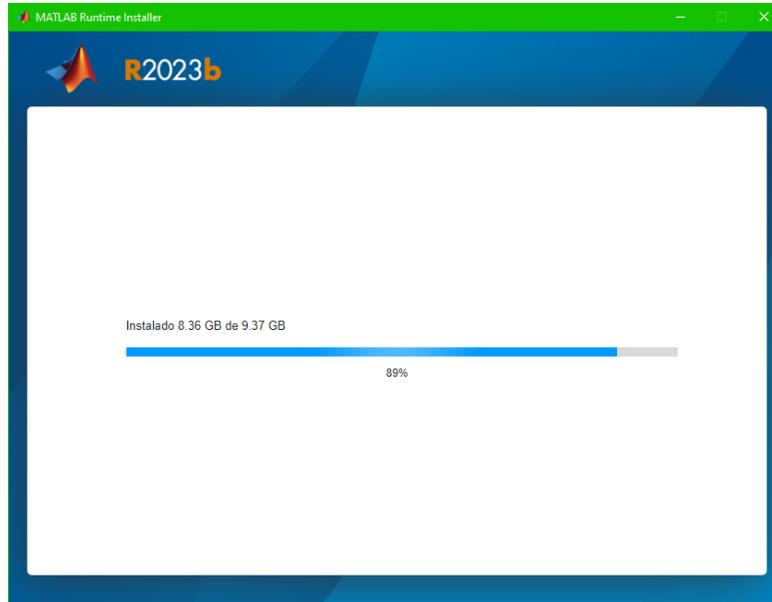
*Ubicación donde se instalará el Runtime de Matlab.*

Y presione el botón verde *Begin Install* para comenzar la instalación.



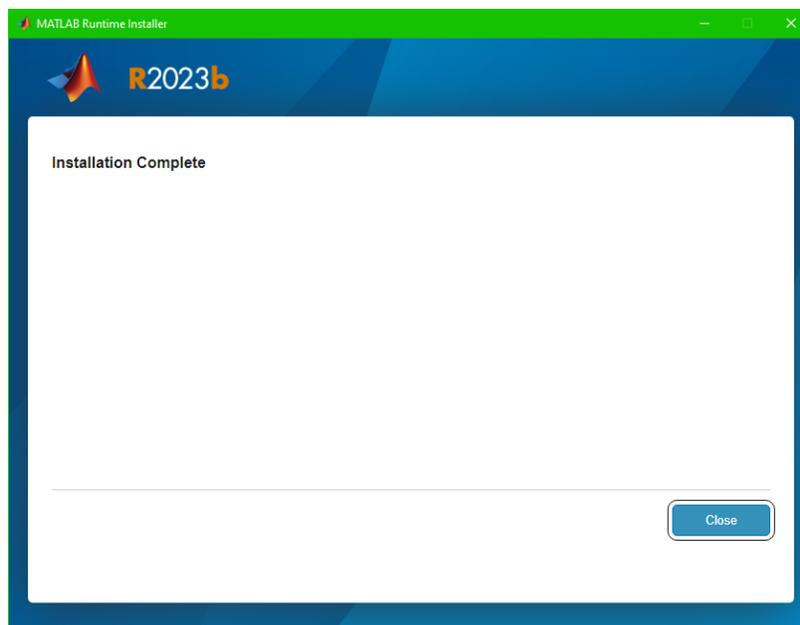
*Inicialización de la instalación del Runtime de Matlab.*

Espere a que termine el proceso.



*Proceso de instalación del Runtime de Matlab.*

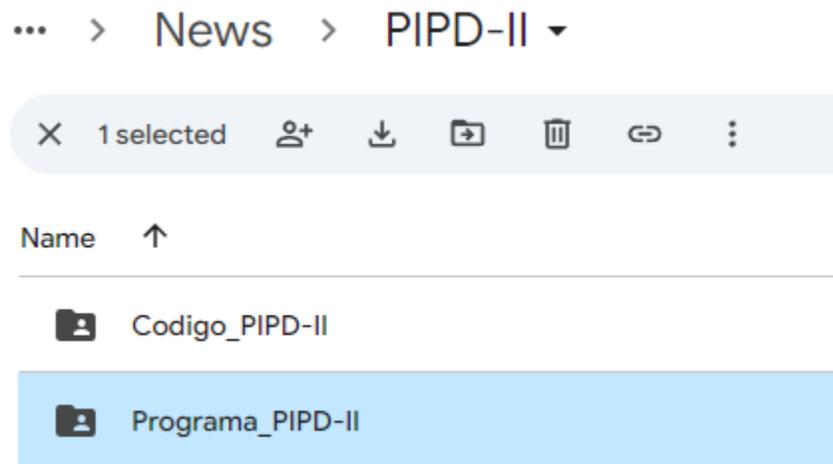
Una vez terminada la instalación, presione el botón de *close* con lo cual se termina la instalación del Runtime de MATLAB.



*Runtime instalado exitosamente.*

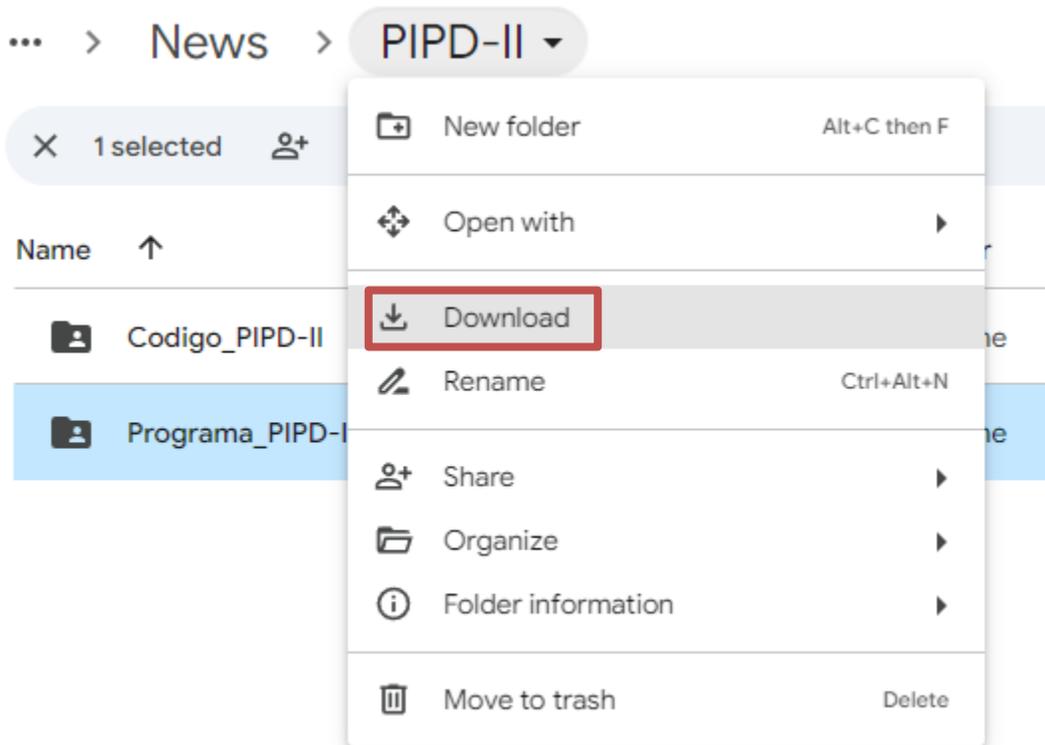
### **Descarga del programa "PIPD-II"**

Descargar el programa "PIPD-II" se logra ingresando al enlace: [https://drive.google.com/drive/folders/1QEAsISbEKdO83I7ByPAHRoAUsKDrGzgm?usp=drive\\_link](https://drive.google.com/drive/folders/1QEAsISbEKdO83I7ByPAHRoAUsKDrGzgm?usp=drive_link)



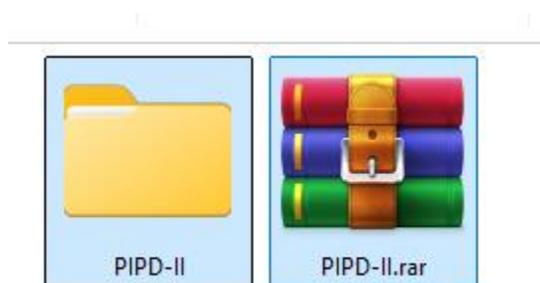
*Contenido de la carpeta "PIPD-II" visto en Drive*

Descargue la carpeta dando clic en la flecha en el encabezado donde aparece el nombre de la carpeta "PIP-II" y proceda a seleccionar descargar o "download".



*Ícono de descargar en Drive*

Al dar clic se empezará a descargar el archivo, el cuál es de 571 KB como podrá verse en el explorador de archivos en la careta de descargas:



*Archivo "PIPD-II.rar" descargado de Drive*

Ahora se procede a descomprimir el archivo descargado de la carpeta de Drive. Este tiene dos carpetas, una con el código fuente y la otra con el programa "PIPD-II", para el usuario son importantes los archivos de "plantilla.xlsx" y el programa principal que es "PIPD-II" que aparece como aplicación de MATLAB.

Nombre	Fecha de modificación	Tipo	Tamaño
E_Cubica.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	28 KB
E_Hexagonal.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	31 KB
E_Monoclinica.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	7 KB
E_Ortorombica.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	32 KB
E_Rombohedral.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	7 KB
E_Tetragonal.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	7 KB
E_Triclinica.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	7 KB
plantilla.xlsx	24/08/2024 05:11 p. m.	Hoja de cálculo d...	11 KB
PIPDII.exe	24/08/2024 05:18 p. m.	Aplicación	1,246 KB

*Plantilla de Excel y la aplicación PIPD-II en el explorador de archivos.*

### Contenido de los Archivos ".xls"

Los archivos ".xls" son hojas de cálculo pertenecientes al programa Microsoft Excel en sus versiones 1997 al 2003 o posterior. Estos permitirán leer y editar múltiples herramientas que se nos proporcionan en la aplicación. Por ejemplo, si examinamos "plantilla.xlsx" nos encontramos lo siguiente:

	x	y	Porcentaje de error (%)	Valor maximo de h,k,l	Material	Tipo de E	Estructuras
Punto 1	-3.47	-5.92	5	15	Au	1	1 Cúbica
Punto 2	3.47	-5.72					2 Tetragonal
Punto 3	7.15	0.2					3 Ortorrómbica
							4 Hexagonal
							5 Romboédrica
							6 Monoclínica
							7 Triclínica

*Imagen de plantilla de Excel.*

Esta plantilla que es donde los datos de entrada van a estar almacenados, está dividido en 3 tablas:

**Tabla A:** La tabla A, primera a la izquierda, abarca de la celda B3 hasta la celda D5 y es la que almacena los datos de las coordenadas (x, y) de cada uno de los tres puntos del patrón de difracción que se quiere indizar.

**Tabla B:** La tabla B, al centro, guardará los siguientes datos: *porcentaje de error (%)*, que es el error máximo que el usuario aceptará entre el valor de ángulos medidos y calculados; *Valor máximo de los índices de Miller (h, k, l)*, para algunos especialistas no vale la pena trabajar con valores muy grandes de los índices de Miller, por ejemplo, Andrews [1] en su libro sugiere tomar un valor máximo de 8, en algunos otros casos se limitan a valores máximos de 5, etc. Por esa razón, dejamos que el usuario defina con base a gusto o experiencia este valor; *Material*, será más rápido y seguro que se llegue a una buena solución si se conoce el material que se pretende estudiar o se tiene una idea de él. En caso contrario, tendrá que correr el programa con las opciones que él se consideren necesarias y compararlas al final para poder llegar a una posible solución; *Tipo de estructura cristalina*, este dato está ligado a la **Tabla C**, también puede ahorrar tiempo si se conoce la estructura que se está analizando, recordemos que existen materiales que se pueden presentar con más de una estructura cristalina.

### Archivo ".exe"

El archivo ".exe" es el ejecutable para sistemas operativos Windows. Esta extensión hace referencia a "executable" (o ejecutable en español) y al abrirlos, el sistema operativo leerá y ejecutará las instrucciones que éste contenga dando lugar a múltiples acciones como mostrar ventanas nuevas, operaciones matemáticas, uso y manejo de archivos, entre otros.

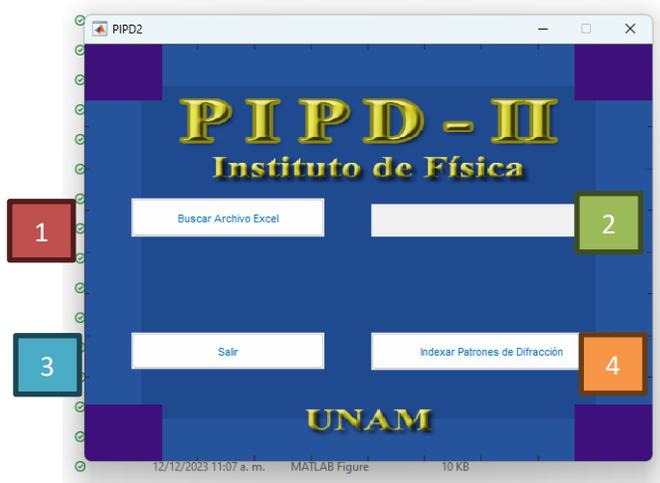
El primer paso será que se ubique en el explorador de archivos, es decir, en la carpeta donde encontramos nuestro programa.

Nombre	Fecha de modificación	Tipo	Tamaño
E_Cubica.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	28 KB
E_Hexagonal.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	31 KB
E_Monoclinica.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	7 KB
E_Ortorombica.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	32 KB
E_Rombohedral.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	7 KB
E_Tetragonal.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	7 KB
E_Triclinica.xls	24/08/2024 05:11 p. m.	Hoja de cálculo d...	7 KB
plantilla.xlsx	24/08/2024 05:11 p. m.	Hoja de cálculo d...	11 KB
PIPDII.exe	24/08/2024 05:18 p. m.	Aplicación	1,246 KB

*PIPD-II en el explorador de archivos de Windows*

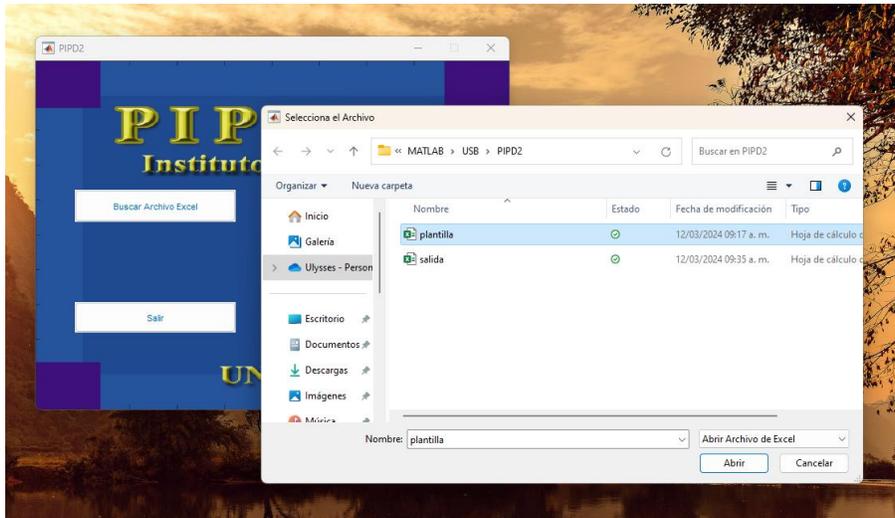
Al ejecutar el programa "PIPD-II.exe" se abrirá la ventana inicial del programa, tal como se muestra en la siguiente imagen.

Como se observa, la ventana cuenta con 3 botones y un label de texto, los cuales están enumerados, y tienen las siguientes funciones.



*Ventana del programa PIPD-II*

**Botón "1"**: el primer botón (botón "1") que se encuentra, es el de buscar el archivo de la *platilla*. De clic sobre dicho botón y se abrirá una ventana con el explorador de archivos del sistema operativo, en dicha ventana deberá ubicar la plantilla de Excel con nombre "*plantilla.xlsx*" como se muestra a continuación.



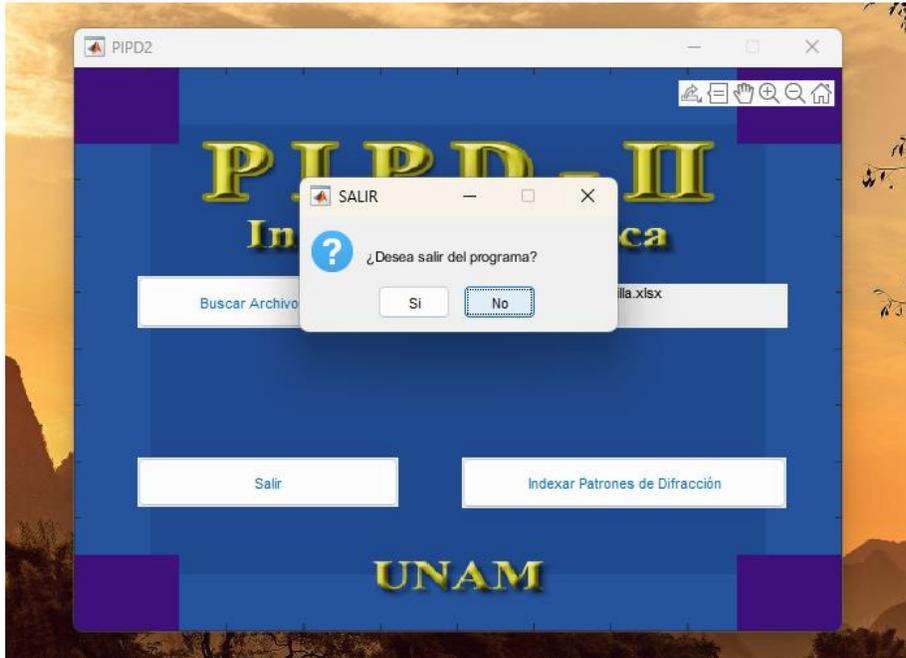
*Función del botón "1" (Buscar Archivo de Excel)*

Para corroborar que no cometió algún error al seleccionar la plantilla, dicho nombre aparecerá en la posición del **Label "2"**.



*Visualización del label con el nombre del archivo a usar (plantilla.xlsx)*

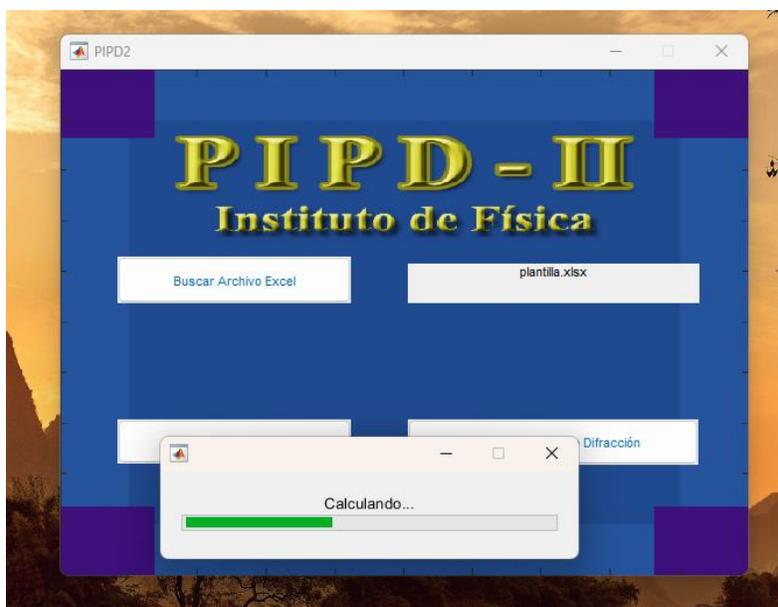
**Botón "3"**: El botón "3" sirve para poder salir de la aplicación, al dar clic, abrirá una ventana emergente para preguntar si se está seguro de que desea salir de la aplicación como se muestra a continuación.



**Botón "4"**: El botón "4", es para ejecutar el programa a partir de los datos que se encuentran en la plantilla de Excel. Al dar clic, se mostrará una ventana emergente, la cual indica el proceso que se está efectuando con una barra de porcentaje que irá llenándose conforme avance el proceso, al final, se cerrará la ventana emergente con la barra de porcentaje y abrirá otra emergente que indicará "proceso terminado", además creará un archivo de Excel con nombre "salida.xlsx" donde se mostrarán los resultados de la indexación realizada. El archivo de salida se almacenará en la misma carpeta donde se ubica el programa.



*Inicio del proceso de indexado (recopilar la información de "plantilla.xlsx").*



*Proceso de la realización de los cálculos para indexar.*

**Archivo de salida**

El archivo de salida tendrá la siguiente información:

**Sección A:** En esta sección se tendrán algunos de los datos que se ingresaron en el archivo de "plantilla.xlsx", para que se pueda identificar rápidamente.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
1																								
2																								
3																								
4																								
5																								
6																								
7																								
8																								
9																								
10	Solución	u	v	w	h1	k1	l1	h2	k2	l2	h3	k3	l3	d1	Δd1	d2	Δd2	d3	Δd3	θ1	Δθ1	θ3	Δθ3	
11	1	-4	-4	4	0	2	2	-2	2	0	-2	0	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
12	2	4	-4	4	0	2	2	-2	0	2	-2	-2	0	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
13	3	-4	4	-4	0	2	2	2	2	0	2	0	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
14	4	4	4	-4	0	2	2	2	2	0	2	2	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
15	5	4	4	-4	2	0	2	2	2	-2	0	0	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
16	6	4	-4	-4	2	0	2	0	-2	2	-2	-2	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
17	7	-4	4	4	2	0	2	2	2	0	2	0	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
18	8	-4	-4	4	2	0	2	0	2	2	-2	2	0	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
19	9	-4	4	-4	2	2	0	2	0	-2	0	-2	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
20	10	-4	4	4	2	2	0	0	2	-2	-2	0	-2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
21	11	4	-4	-4	2	2	0	2	0	2	0	-2	2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
22	12	4	-4	4	2	2	0	0	2	2	-2	0	2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
23	13	-4	-4	4	0	-2	-2	2	-2	0	2	0	2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
24	14	4	-4	4	0	-2	-2	2	0	-2	2	2	0	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
25	15	-4	4	-4	0	-2	-2	-2	-2	0	-2	0	2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
26	16	4	4	-4	0	-2	-2	-2	0	-2	-2	2	0	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	
27	17	4	4	-4	-2	0	-2	-2	2	0	0	2	2	1.4383	0.019	1.4383	0.019	1.4383	0.0402	59.9992	57.4526	119.9984	1.9789	

Proceso de indexado terminado con el archivo de "salida.xlsx".

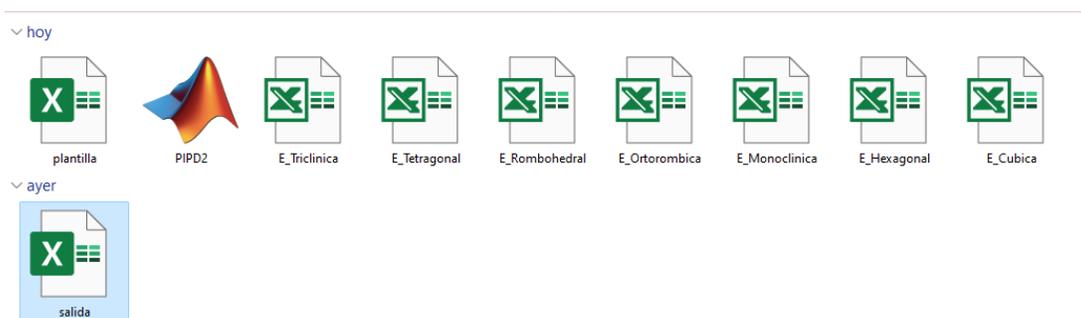
**Columna A:** En esta columna se incluye la numeración progresiva de las posibles soluciones que propone el programa para la indexación.

**Columnas B-D:** En estas columnas encontrará el vector (u, v, w) correspondiente al eje de zona de cada posibles solución.

**Columnas E-M:** En estas columnas encontrará los índices de Miller que el programa propone para cada uno de los tres planos para esa posible solución.

**Columnas N-S:** En estas columnas encontrará las distancias y las diferencias entre las distancias obtenidas a partir de los datos de partida, es decir, las que se obtienen a partir de las posiciones (x, y) que usted proporcionó como datos y en centro del patrón de difracción, y las distancias que resultarán al considerar los índices de Miller propuesto como solución para cada caso.

**Columnas T-W:** En estas columnas encontrará los ángulos formados entre el centro del patrón de difracción y dos de los spots cuyas coordenadas (x, y) usted proporcionó como datos de entrada. Además, encontrará las diferencias entre dichos ángulos y los que resultan a partir de los índices de Miller propuestos como solución para cada caso.

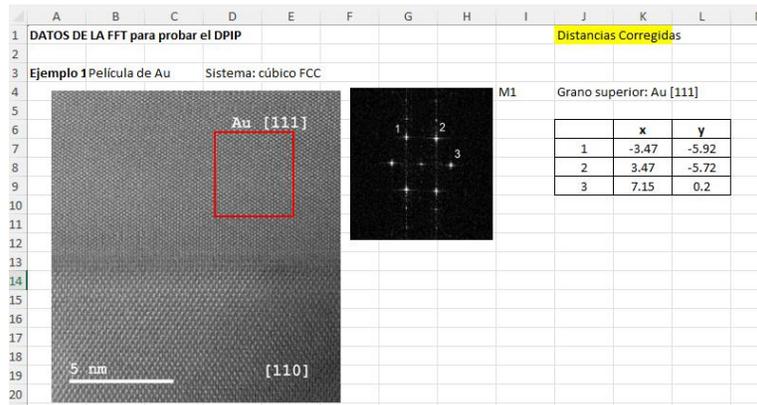


*Archivo "salida.xlsx" en la carpeta donde se encuentra el programa.*

Para mejor seguimiento se realizarán dos indexaciones con el elemento Au [111] que se muestra a continuación:

- Ejemplo Au cúbico:

Se tienen los siguientes datos para poder indexar:



*Datos proporcionados de Au.*

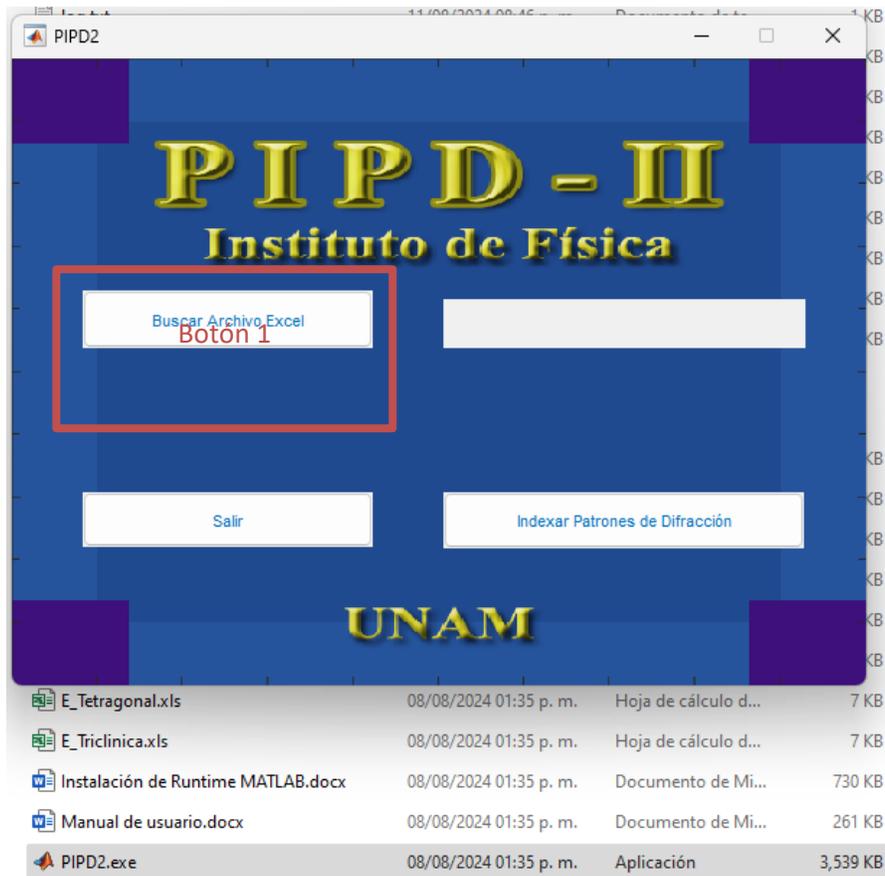
Por lo que se procede a buscar el archivo "plantilla.xlsx" del programa descargado y después se ingresan los datos proporcionados:

	B	C	D	F	G	H	I	K	L
2		x	y	Porcentaje de error (%)	Valor maximo de h,k,l	Material	Tipo de Estructura	Estructuras	
3	Punto 1	-3.47	-5.92	1	10	Au	1	1	Cúbica
4	Punto 2	3.47	-5.72					2	Tetragonal
5	Punto 3	7.15	0.2					3	Ortorrónica
								4	Hexagonal
								5	Romboédrica
								6	Monoclínica
								7	Triclínica

*Imagen de la plantilla del ejemplo Au.*

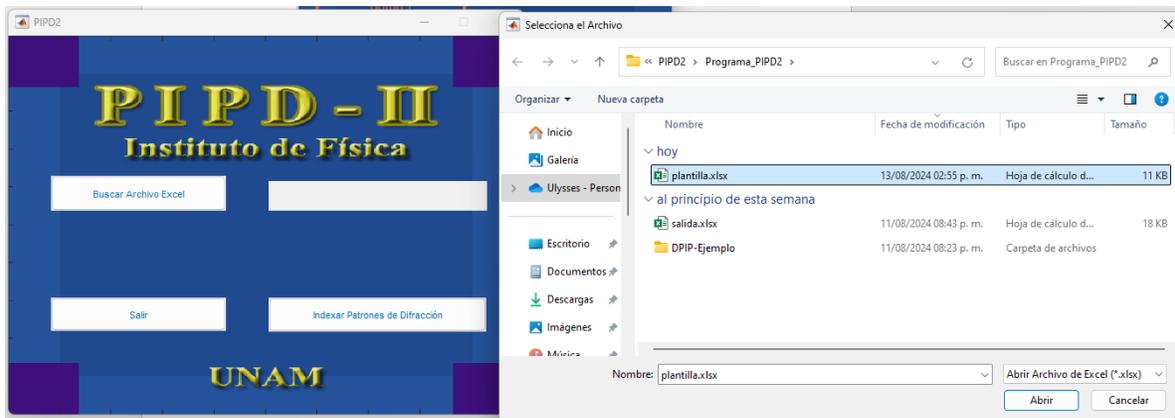
Como se observa, los datos de los spots, que son las coordenadas, están en distintas celdas, es decir, el spot 1 o punto 1, sus coordenadas son (-3.47, -5.92) esto lo vamos a colocar de manera ordenada, en la fila que dice "Punto 1" (celda B3 en Excel) en la columna que dice "x" (celda C3 en Excel) colocaremos el valor de "-3.47" y en la columna de las "y" (celda D3 en Excel) de la misma fila irá "-5.92" tal como se muestra en la imagen de arriba. También se procede a colocar en la celda F3 el porcentaje de error que uno requiera, en este caso es de 1%, los valores máximos de los índices de Miller que van en la celda G3, en este caso 10 de valores máximo, el nombre del material que es "Au" irá en la celda H3 y el tipo de estructura que es cúbica, se revisa la tabla que está en las celdas K2 a L9, podemos revisar que el tipo de estructura cúbica es la número 1, por lo que en la celda I3 que corresponde al tipo de estructura se colocará 1.

Se guarda el archivo y se cierra para poder trabajar de manera correcta en el programa.



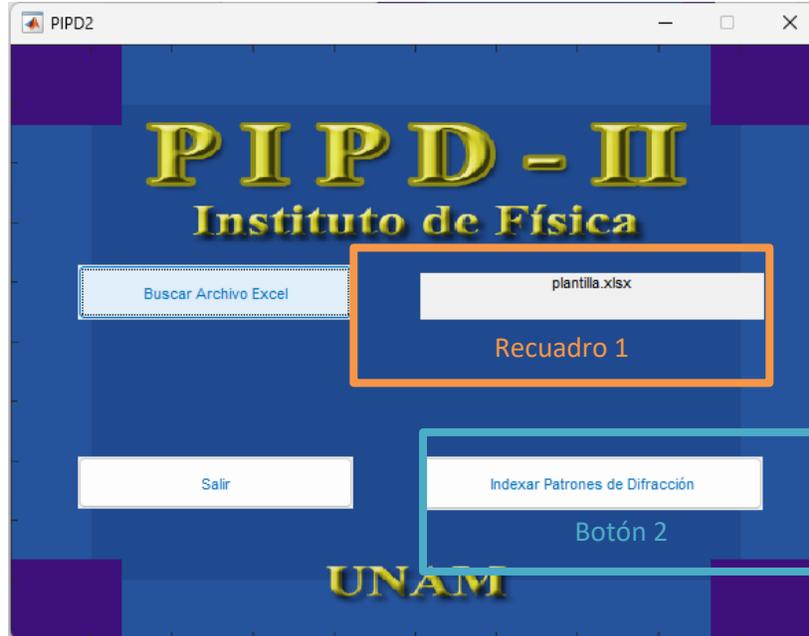
*Ejecución del programa PIPD-II.*

Una vez que abrimos el programa, se procede a buscar la plantilla para poder hacer la indexación, dando clic en el **Botón 1** y se abrirá el buscador de archivos de Windows:



*Explorador de archivo de Windows y selección de plantilla.*

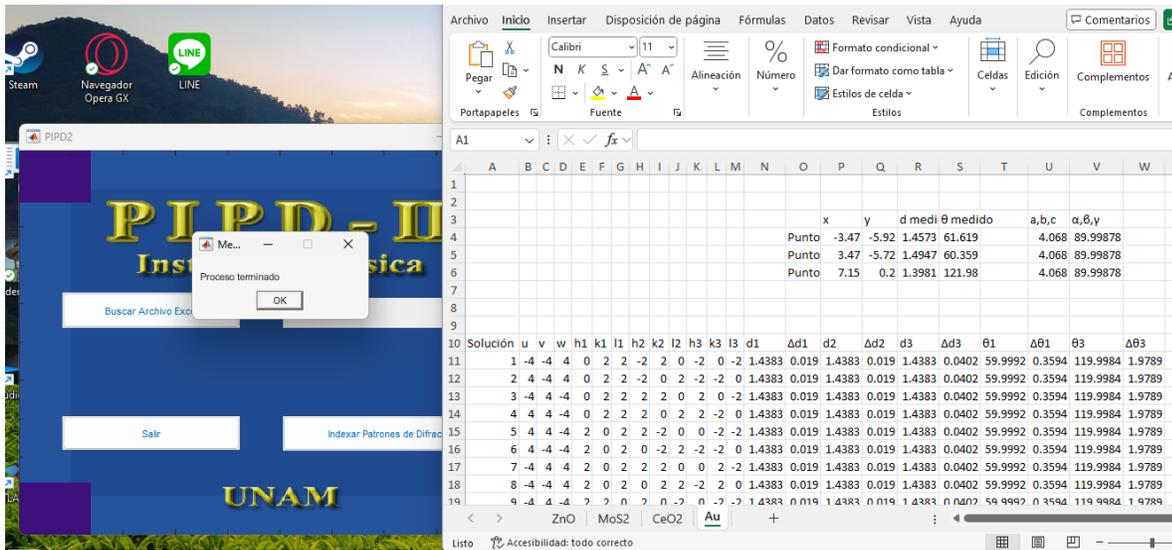
Como se observa, en el explorador de archivos de Windows se encuentra de manera rápida el archivo de "plantilla.xlsx" por lo que se procede a seleccionar y abrirlo para que el programa lo cargue:



*Programa PIPD-II con el archivo cargado.*

Se observa que, para este punto, el archivo de plantilla se cargó correctamente, dando la indicación de su nombre en el **Recuadro 1**. Por lo que para indexar, se procede a dar clic en el **Botón 2**, cuya etiqueta expresa "Indexar Patrones de Difracción".

Si todo marchó de manera correcta, deberá aparecer el mensaje "Proceso terminado" en una ventana emergente y se abrirá el archivo de Excel llamado "salida.xlsx" como se muestra a continuación.



*Indexación terminada por el programa PIPD-II.*

Y finalmente obtenemos los posibles resultados de la indexación del elemento Au [111] con los datos proporcionados anteriormente, como se observa, el archivo de "salida.xlsx" facilita al investigador con los símbolos para identificar ya sea distancias y ángulos, así como contiene también los datos proporcionados para visualizar mejor los resultados.

Cualquier duda, sugerencia o problema se podrá contactar directamente al correo [ulyssesbass@hotmail.com](mailto:ulyssesbass@hotmail.com).

## Anexo II

Formulae for determination of interplanar spacings, interplanar angles and interzonal angles

Crystal system	Interplanar spacing of the (hkl) plane	Angle $\phi$ between $(h_1k_1l_1)$ and $(h_2k_2l_2)$
Cubic $a = b = c$ $\alpha = \beta = \gamma = 90^\circ$	$\frac{1}{d^2} = \frac{1}{a^2}(h^2 + k^2 + l^2)$	$\cos \phi = \frac{h_1h_2 + k_1k_2 + l_1l_2}{\sqrt{[(h_1^2 + k_1^2 + l_1^2)(h_2^2 + k_2^2 + l_2^2)']}}$
Tetragonal $a = b \neq c$ $\alpha = \beta = \gamma = 90^\circ$	$\frac{1}{d^2} = \frac{1}{a^2}(h^2 + k^2) + \frac{1}{c^2}(l^2)$	$\cos \phi = \frac{\frac{1}{a^2}(h_1h_2 + k_1k_2) + \frac{1}{c^2}(l_1l_2)}{\sqrt{\left\{ \left[ \frac{1}{a^2}(h_1^2 + k_1^2) + \frac{1}{c^2}l_1^2 \right] \left[ \frac{1}{a^2}(h_2^2 + k_2^2) + \frac{1}{c^2}l_2^2 \right] \right\}}}$
Orthorhombic $a \neq b \neq c$ $\alpha = \beta = \gamma = 90^\circ$	$\frac{1}{d^2} = \frac{1}{a^2}(h^2) + \frac{1}{b^2}(k^2) + \frac{1}{c^2}(l^2)$	$\cos \phi = \frac{\frac{1}{a^2}h_1h_2 + \frac{1}{b^2}k_1k_2 + \frac{1}{c^2}l_1l_2}{\sqrt{\left[ \left( \frac{1}{a^2}h_1^2 + \frac{1}{b^2}k_1^2 + \frac{1}{c^2}l_1^2 \right) \left( \frac{1}{a^2}h_2^2 + \frac{1}{b^2}k_2^2 + \frac{1}{c^2}l_2^2 \right) \right]}}$
Hexagonal $a = b \neq c$ $\alpha = \beta = 90^\circ; \gamma = 120^\circ$	$\frac{1}{d^2} = \frac{4}{3a^2}(h^2 + hk + k^2) + \frac{1}{c^2}(l^2)$	$\cos \phi = \frac{h_1h_2 + k_1k_2 + \frac{1}{2}(h_1k_2 + k_1h_2) + \frac{2}{3}l_1l_2}{\sqrt{\left[ \left( h_1^2 + k_1^2 + h_1k_1 + \frac{2}{3}l_1^2 \right) \left( h_2^2 + k_2^2 + h_2k_2 + \frac{2}{3}l_2^2 \right) \right]}}$
Rhombohedral $a = b = c$ $\alpha = \beta = \gamma < 120^\circ \neq 90^\circ$	$\frac{1}{d^2} = \frac{1}{a^2} \left\{ \frac{(1 + \cos \alpha)(h^2 + k^2 + l^2) - (1 - \tan^2 \frac{1}{2}\alpha)(hk + hl + lh)}{(1 + \cos \alpha - 2 \cos^2 \alpha)} \right\}$	Convert to corresponding hexagonal indices (see p. 78) and use the above two formulae.
Monoclinic $a \neq b \neq c$ $\alpha = \gamma = 90^\circ \neq \beta$	$\frac{1}{d^2} = \frac{1}{a^2} \left( \frac{h^2}{\sin^2 \beta} \right) + \frac{1}{b^2}(k^2) + \frac{1}{c^2} \left( \frac{l^2}{\sin^2 \beta} \right) - \frac{2hl \cos \beta}{ac \sin^2 \beta}$	$\cos \phi = \frac{\frac{1}{a^2}h_1h_2 + \frac{1}{b^2}k_1k_2 \sin^2 \beta + \frac{1}{c^2}l_1l_2 - \frac{1}{ac}(l_1h_2 + l_2h_1) \cos \beta}{\sqrt{\left[ \left( \frac{1}{a^2}h_1^2 + \frac{1}{b^2}k_1^2 \sin^2 \beta + \frac{1}{c^2}l_1^2 - \frac{2h_1l_1}{ac} \cos \beta \right) \left( \frac{1}{a^2}h_2^2 + \frac{1}{b^2}k_2^2 \sin^2 \beta + \frac{1}{c^2}l_2^2 - \frac{2h_2l_2}{ac} \cos \beta \right) \right]}}$
Triclinic $a \neq b \neq c$ $\alpha \neq \beta \neq \gamma$	$\frac{1}{d^2} = \frac{1}{V^2}(s_{11}h^2 + s_{22}k^2 + s_{33}l^2 + 2s_{12}hk + 2s_{23}kl + 2s_{31}lh)$ <p>where  <math>V^2 = a^2b^2c^2(1 - \cos^2 \alpha - \cos^2 \beta - \cos^2 \gamma + 2 \cos \alpha \cos \beta \cos \gamma)</math>                      and  <math>s_{11} = b^2c^2 \sin^2 \alpha</math>  <math>s_{22} = a^2c^2 \sin^2 \beta</math>  <math>s_{33} = a^2b^2 \sin^2 \gamma</math>  <math>s_{12} = abc^2(\cos \alpha \cos \beta - \cos \gamma)</math>  <math>s_{23} = a^2bc(\cos \beta \cos \gamma - \cos \alpha)</math>  <math>s_{31} = ab^2c(\cos \gamma \cos \alpha - \cos \beta)</math></p>	$\cos \phi = \frac{F}{A_{h_1k_1l_1} \cdot A_{h_2k_2l_2}}$ <p>where <math>F = h_1h_2b^2c^2 \sin^2 \alpha + k_1k_2a^2c^2 \sin^2 \beta + l_1l_2a^2b^2 \sin^2 \gamma + abc^2(\cos \alpha \cos \beta - \cos \gamma)(k_1h_2 + h_1k_2) + ab^2c(\cos \gamma \cos \alpha - \cos \beta)(h_1l_2 + l_1h_2) + a^2bc(\cos \beta \cos \gamma - \cos \alpha)(k_1l_2 + l_1k_2)</math>                      and  <math display="block">A_{hkl} = \sqrt{\left[ \begin{aligned} &amp;h^2b^2c^2 \sin^2 \alpha + k^2a^2c^2 \sin^2 \beta + l^2a^2b^2 \sin^2 \gamma \\ &amp;+ 2hkb^2c^2(\cos \alpha \cos \beta - \cos \gamma) \\ &amp;+ 2hlab^2c(\cos \gamma \cos \alpha - \cos \beta) \\ &amp;+ 2kla^2bc(\cos \beta \cos \gamma - \cos \alpha) \end{aligned} \right]}</math></p>

(Andrews et al., Electron Diffraction Patterns: facing page 72)

## Anexo III

### Apéndice de código

#### Código "obtenciondatos.m"

```
function [Vdm,Vam,spots] = obtenciondatos(punto1,punto2,punto3,valueoption)
spots = [];
bandera = 0;
try
    listaXY = punto1; %strsplit(punto1,',');
    x = str2double(listaXY(1));
    y = str2double(listaXY(2));
    spots = cat(1,spots,[x y]);

    listaXY = punto2; %strsplit(punto2,',');
    x = str2double(listaXY(1));
    y = str2double(listaXY(2));
    spots = cat(1,spots,[x y]);

    listaXY = punto3; %strsplit(punto3,',');
    x = str2double(listaXY(1));
    y = str2double(listaXY(2));
    spots = cat(1,spots,[x y]);
catch
    bandera = 1;
end

if bandera == 0
    if valueoption == 1
        dm(1)=10/norm(spots(1,:));
        dm(2)=10/norm(spots(2,:));
        dm(3)=10/norm(spots(3,:));

        ppij=dot(spots(1,:),spots(3,:));
        ni=norm(spots(1,:));
        nj=norm(spots(3,:));
        cang=ppij/(ni*nj);
        angrm=acos(cang);
        ang13=angrm*(57.295);

        ppij=dot(spots(2,:),spots(3,:));
        ni=norm(spots(2,:));
        nj=norm(spots(3,:));
        cang=ppij/(ni*nj);
        angrm=acos(cang);
        ang23=angrm*(57.295);

        ppij=dot(spots(1,:),spots(2,:));
        ni=norm(spots(1,:));
        nj=norm(spots(2,:));
        cang=ppij/(ni*nj);
```

```
    angrm=acos(cang);
    ang12=angrm*(57.295);
else
    dm(1) = spots(1,1);
    dm(2) = spots(2,1);
    dm(3) = spots(3,1);

    ang13 = spots(1,2);
    ang23 = spots(2,2);
    ang12 = spots(3,2);
end

Vdm=[dm(1),dm(2),dm(3)];
Vam=[ang12,ang23,ang13];
end
end
```

## Código "Compara.m"

```

function [MP]= Compara(dm,np,MPlARx)
%
% Esta función comparada el valor de una distancia medida (dm)en el patrón
% de difracción con las d's de la Tabla de Rayos X (dRx,h,k,l) a usar.
% Primero busca el valor más cercano entre ellas en la columna 1 de dRx,
% y guarda esa posición en b. Localizado el dRx más cercano a dm, a partir
% de ese dRx, toma un dRx más hacia arriba(b+1) y un dRx más hacia
% abajo (b-1) para tener un total de 3 posibles soluciones. Esto se hace
% considerando que se comete un error de un pixel al escoger el pixel
% de máxima intensidad del spot a estudiar.
% El número de dRx's que se consideran, disminuye dependiendo del valor
% de b.
% En Celd unicamente se guardan los (h,k,l) seleccionados.

dRx=MPlARx(1:np);           % vector con d's de los ns spots a tratar.
Dmin=2;
Dmax=np-1;

[~,b]=min(abs(dRx-dm));

if b==1
    V=MPlARx(1:3,2:4);
end

if b==2
    V=MPlARx(1:4,2:4);
end

if b==np
    V=MPlARx((np-2):np,2:4);
end

if b==np-1
    V=MPlARx((np-3):np,2:4);
end

if b>1 && b<np
    V=MPlARx((b-1):(b+2),2:4);
end

[m,~]=size(V);
MP=[];
%
% Ahora realiza todas las permutaciones sin repetición de cada
% vector (h,k,l) de Mcorta considerando los signos.
% Mcorta puede ser de [3x3] entonces MP será [144x3] y puede ser de
% [2x3] por lo que MP será de [96x3].

```

```

for j=1:m
    h=V(j,1);
    k=V(j,2);
    l=V(j,3);
    x1=[ h k 1];
    x2=[-h -k -1];
    x3=[-h k 1];
    x4=[ h -k 1];
    x5=[ h k -1];
    x6=[-h -k 1];
    x7=[-h k -1];
    x8=[ h -k -1];

    P=[perms(x1);perms(x2);perms(x3);perms(x4);
        perms(x5);perms(x6);perms(x7);perms(x8)];
    MP=cat(1,MP,P);
end

end

```

### ***Código "Calculad13.m"***

```

function [Md]= Calculad13(P,L,Mabc,dm)
    FILE = fopen("Calculad13.txt","w+");

    [z,x] = size(P);
    fprintf(FILE,"\nP = \n");
    for q = 1:z
        fprintf(FILE,"[");
        for e = 1:x
            if e ~= 1
                fprintf(FILE,",");
            end
            fprintf(FILE,"% .2f",P(q,e));
        end
        fprintf(FILE,"]\n");
    end
    [z,x] = size(L);
    fprintf(FILE,"L = \n");
    for q = 1:z
        fprintf(FILE,"[");
        for e = 1:x
            if e ~= 1
                fprintf(FILE,",");
            end
            fprintf(FILE,"% .2f",L(q,e));
        end
        fprintf(FILE,"]\n");
    end
    [z,x] = size(Mabc);

```

```

fprintf(FILE,"Mabc = \n");
for q = 1:z
    fprintf(FILE,"[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE,",");
        end
        fprintf(FILE,"%2f",Mabc(q,e));
    end
    fprintf(FILE,"]\n");
end
fprintf(FILE,"dm = %2f\n",dm);

% Esta función calcula las distancias d de acuerdo a la estructura.

a=Mabc(1,2);
b=Mabc(1,3);
c=Mabc(1,4);

fprintf(FILE,"a = %2f\n",a);
fprintf(FILE,"b = %2f\n",b);
fprintf(FILE,"c = %2f\n\n",c);

aa=a*a;
bb=b*b;
cc=c*c;

fprintf(FILE,"\naa = %2f\n",aa);
fprintf(FILE,"bb = %2f\n",bb);
fprintf(FILE,"cc = %2f\n\n",cc);

alfa=Mabc(2,1);
beta=Mabc(2,2);
gama=Mabc(2,3);
estructura=Mabc(1);

fprintf(FILE,"\nalfa = %2f\n",alfa);
fprintf(FILE,"beta = %2f\n",beta);
fprintf(FILE,"gama = %2f\n",gama);
fprintf(FILE,"estructura = %d\n\n",estructura);

err=0.1;

fprintf(FILE,"err = %2f\n\n",err);

Md=[];
[LMD,~] = size(Md);

while LMD == 0
    error = dm * err;
    fprintf(FILE,"\nerr = %2f\n",err);
    error = abs(error);
    fprintf(FILE,"err = %2f\n",err);
end

```

```

for i=1:L(1,1)

    h=P(i,1);
    k=P(i,2);
    l=P(i,3);
    hh=h*h;
    kk=k*k;
    ll=l*l;

    fprintf(FILE, "\nh = %.2f\n",h);
    fprintf(FILE, "k = %.2f\n",k);
    fprintf(FILE, "l = %.2f\n",l);
    fprintf(FILE, "hh = %.2f\n",hh);
    fprintf(FILE, "kk = %.2f\n",kk);
    fprintf(FILE, "ll = %.2f\n",ll);

switch estructura

    case 1

        d=sqrt((aa/(hh+kk+ll)));

    case 2

        d=sqrt(aa*cc/(cc*(hh+kk)+aa*ll));

    case 3

        d=sqrt(aa*bb*cc/(bb*cc*hh+aa*cc*kk+aa*bb*ll));

    case 4

        d=sqrt(3*aa*cc/(4*cc*(hh+h*k+kk)+3*aa*ll));

    case 5

        ca=cos(alfa);
        ca2=ca*ca;
        ta=tan(alfa/2)^2;
        x1=aa*(1+ca-2*ca2);
        x2=(1+ca)*(hh+kk+ll)-(1-ta)*(h*k+k*l+l*h);
        d=sqrt(x1/x2);

    case 6

        sb2=sin(beta)*sin(beta);
        cb=cos(beta);
        x1=hh/aa*sb2;
        x2=kk/bb;
        x3=ll/cc*sb2;
        x4=2*h*l*cb/a*c*sb2;
        x=x1*x2+x3-x4;
        d=sqrt(1/x);

```

```

    case 7

        sa=sin(alfa);
        sb=sin(beta);
        sg=sin(gama);
        sa2=sa*sa;
        sb2=sb*sb;
        sg2=sg*sg;
        ca=cos(alfa);
        cb=cos(beta);
        cg=cos(gama);
        ca2=ca*ca;
        cb2=cb*cb;
        cg2=cg*cg;
        s11=bb*cc*sa2;
        s22=aa*cc*sb2;
        s33=aa*bb*sg2;
        s12=a*b*cc*(ca*cb-cg);
        s23=aa*b*c*(cb*cg-ca);
        s31=a*bb*c*(cg*ca-cb);
        V2=(aa*bb*cc)*(1-ca2-cb2-cg2+2*ca*cb*cg);
        x=V2i*(s11*hh+s22*kk+s33*ll+2*s12*h*k+2*s23*k*l+2*s31*l*h);
        d=sqrt(V2/x);
    end

    Dd = abs(dm-d);

    fprintf(FILE,"d = %.2f\n",d);
    fprintf(FILE,"Dd = %.2f\n",Dd);

    if Dd <= error
        Sd = [h,k,l,d,Dd];
        Md = cat(1,Md,Sd);
    end
end
LMd = size(Md);
if LMd == 0
    err = err + 0.01;
end
end
[z,x] = size(Md);
fprintf(FILE,"\nMd = \n");
for q = 1:z
    fprintf(FILE,"[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE,",");
        end
        fprintf(FILE,"%.2f",Md(q,e));
    end
    fprintf(FILE,"]\n");
end
end

```

```
    fclose(FILE);  
end
```

**Código "Calculateta13.m"**

```

function [Mteta]= Calculateta13(M1,M3,Mabc,am)

    FILE = fopen("Calculateta13.txt","w+");

    [z,x] = size(M1);
    fprintf(FILE, "\nM1 = \n");
    for q = 1:z
        fprintf(FILE, "[");
        for e = 1:x
            if e ~= 1
                fprintf(FILE, ",");
            end
            fprintf(FILE, "%.2f", M1(q,e));
        end
        fprintf(FILE, "]\n");
    end

    [z,x] = size(M3);
    fprintf(FILE, "M3 = \n");
    for q = 1:z
        fprintf(FILE, "[");
        for e = 1:x
            if e ~= 1
                fprintf(FILE, ",");
            end
            fprintf(FILE, "%.2f", M3(q,e));
        end
        fprintf(FILE, "]\n");
    end

    [z,x] = size(Mabc);
    fprintf(FILE, "Mabc = \n");
    for q = 1:z
        fprintf(FILE, "[");
        for e = 1:x
            if e ~= 1
                fprintf(FILE, ",");
            end
            fprintf(FILE, "%.2f", Mabc(q,e));
        end
        fprintf(FILE, "]\n");
    end
    fprintf(FILE, "am = %.2f\n", am);

    % Esta función calcula el ángulo entre los dos vectores
    % de acuerdo a la estructura explorada.

    % Se trabaja con:
    % caso: se refiere a los spots que se están analizando, 12 significa
    % que se están trabajando los spots 1 y 2; 23 los spots 2 y 3.

```

```

% Pla1= Matriz de indices de Miler spot 1(caso 12) ó 2 (caso 23);
% pla2=iMatriz de indices de Miler spot 2(caso 12) ó 3 (caso 23);

% Lo que se encuentra en la Matrix Mabc:
% (1,1)= Estructura;
% (1,2)= valor del parámetro de red a;
% (1,3)= valor del parámetro de red b;
% (1,4)= valor del parámetro de red c;
% (2,1)= valor del ángulo alfa;
% (2,2)= valor del ángulo Beta;
% (2,3)= valor del ángulo gamma;
% (2,4)= 1 si la estructura es centrada en las caras 2 si es centrada
%          en el cuerpo.r=ángulo calculado en radianes;

% tetag=ángulo calculado en grados;
% dc= distancia calculada;
% EZ=eje de zona;
% Solu=matriz de salida

% Pla1: es la matriz que contiene todas las combinaciones (incluyendo
% signos) de (h k l) para el spot 1

a = Mabc(1,2);
b=Mabc(1,3);
c=Mabc(1,4);

fprintf(FILE,"a = %.2f\n",a);
fprintf(FILE,"b = %.2f\n",b);
fprintf(FILE,"c = %.2f\n\n",c);

aa=a*a;
bb=b*b;
cc=c*c;

fprintf(FILE,"\naa = %.2f\n",aa);
fprintf(FILE,"bb = %.2f\n",bb);
fprintf(FILE,"cc = %.2f\n\n",cc);

alfa=Mabc(2,1);
beta=Mabc(2,2);
gama=Mabc(2,3);
estructura=Mabc(1,1);

fprintf(FILE,"\nalfa = %.2f\n",alfa);
fprintf(FILE,"beta = %.2f\n",beta);
fprintf(FILE,"gama = %.2f\n",gama);
fprintf(FILE,"estructura = %d\n\n",estructura);

TM1=size(M1);
TM3=size(M3);
err=0.1;

```

```

Mteta=[];
LMteta=size(Mteta);

while LMteta == 0
    error = am * err;
    fprintf(FILE, "\nerr = %.2f\n",err);
    error = abs(error);
    fprintf(FILE, "err = %.2f\n",err);
    for i=1:TM1
        h1=M1(i,1);
        k1=M1(i,2);
        l1=M1(i,3);
        hh1=h1*h1;
        kk1=k1*k1;
        ll1=l1*l1;
        p1=[h1,k1,l1];
        D1=M1(i,4:5);

        fprintf(FILE, "\nh1 = %.2f\n",h1);
        fprintf(FILE, "k1 = %.2f\n",k1);
        fprintf(FILE, "l1 = %.2f\n",l1);
        fprintf(FILE, "hh1 = %.2f\n",hh1);
        fprintf(FILE, "kk1 = %.2f\n",kk1);
        fprintf(FILE, "ll1 = %.2f\n",ll1);
    [z,x] = size(p1);
    fprintf(FILE, "p1 = \n");
    for q = 1:z
        fprintf(FILE, "[");
        for e = 1:x
            if e ~= 1
                fprintf(FILE, ",");
            end
            fprintf(FILE, "%.2f", p1(q,e));
        end
        fprintf(FILE, "]\n");
    end
    [z,x] = size(D1);
    fprintf(FILE, "D1 = \n");
    for q = 1:z
        fprintf(FILE, "[");
        for e = 1:x
            if e ~= 1
                fprintf(FILE, ",");
            end
            fprintf(FILE, "%.2f", D1(q,e));
        end
        fprintf(FILE, "]\n");
    end

    for j=1:TM3
        h3=M3(j,1);
        k3=M3(j,2);
        l3=M3(j,3);

```

```

hh3=h3*h3;
kk3=k3*k3;
l13=l3*l3;
p3=[h3,k3,l3];
D3=M3(j,4:5);

fprintf(FILE,"h3 = %.2f\n",h3);
fprintf(FILE,"k3 = %.2f\n",k3);
fprintf(FILE,"l3 = %.2f\n",l3);
fprintf(FILE,"hh3 = %.2f\n",hh3);
fprintf(FILE,"kk3 = %.2f\n",kk3);
fprintf(FILE,"l13 = %.2f\n",l13);
[z,x] = size(p3);
fprintf(FILE,"p3 = \n");
for q = 1:z
    fprintf(FILE,"[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE,",");
        end
        fprintf(FILE,"%.2f",p3(q,e));
    end
    fprintf(FILE,"]\n");
end

[z,x] = size(D3);
fprintf(FILE,"D3 = \n");
for q = 1:z
    fprintf(FILE,"[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE,",");
        end
        fprintf(FILE,"%.2f",D3(q,e));
    end
    fprintf(FILE,"]\n");
end

switch estructura
case 1
    costeta=(h1*h3+k1*k3+l1*l3)/sqrt((hh1+kk1+l11)*(hh3+kk3+l13));
    teta=acos(costeta)*57.295;
    Dteta=abs(am-teta);
    A13=[teta,Dteta];
    if Dteta<=error
        p2=[h1+h3,k1+k3,l1+l3];
        MM=[p1,p2,p3,D1,D3,A13];
        Mteta=cat(1,Mteta,MM);
    end
case 2
    t1=((h1*h3+k1*k3)/aa)+(l1*l3/cc);
    t2=((hh1+kk1)/aa)+(l11/cc);

```

```

t3=((hh3*kk3)/aa)+(l13/cc);
costeta=t1/sqrt(t2*t3);
teta=acos(costeta)*57.295;
Dteta=abs(am-teta);
A13=[teta,Dteta];
if Dteta<=error
    p2=[h1+h3,k1+k3,l1+l3];
    MM=[p1,p2,p3,D1,D3,A13];
    Mteta=cat(1,Mteta,MM);
end

case 3
t1=(h1*h3/aa)+(k1*k3/bb)+(l1*l3/cc);
t2=(hh1/aa)+(kk1/bb)+(l11/cc);
t3=(hh3/aa)+(kk3/bb)+(l13/cc);
costeta=t1/sqrt(t2*t3);
teta=acos(costeta)*57.295;
Dteta=abs(am-teta);
A13=[teta,Dteta];
if Dteta<=error
    p2=[h1+h3,k1+k3,l1+l3];
    MM=[p1,p2,p3,D1,D3,A13];
    Mteta=cat(1,Mteta,MM);
end

case 4
t1=h1*h3+k1*k3+0.5*(h1*k3+k1*h3)+(0.75*aa*l1*l3)/cc;
t2=hh1+kk1+h1*k1+(0.75*aa*l1)/cc;
t3=hh3+kk3+h3*k3+(0.75*aa*l3)/cc;
costeta=t1/sqrt(t2*t3);
teta=acos(costeta)*57.295;
Dteta=abs(am-teta);
A13=[teta,Dteta];
if Dteta<=error
    p2=[h1+h3,k1+k3,l1+l3];
    MM=[p1,p2,p3,D1,D3,A13];
    Mteta=cat(1,Mteta,MM);
end

case 5
h1r=(2*h1+k1+l1)/3;
k1r=(-h1+k1+l1)/3;
l1r=(-h1-2*k1+l1)/3;
h3r=(2*h3+k3+l3)/3;
k3r=(-h3+k3+l3)/3;
l3r=(-h3-2*k3+l3)/3;
hh1r=h1r*h1r;
kk1r=k1r*k1r;
ll1r=l1r*l1r;

t1r=h1r*h3r+k1r*k3r+0.5*(h1r*k3r+h3r*k1r)+(0.75*aa/cc)*l1r*l3r;
t2r=hh1r+kk1r+h1r*k1r+(0.75*aa/cc)*ll1r;
t3r=hh3r+kk3r+h3r*k3r+(0.75*aa/cc)*ll3r;

```

```

costeta=t1r/sqrt(t2r*t3r);
teta=acos(costeta)*57.295;
Dteta=abs(am-teta);
A13=[teta,Dteta];
if Dteta<=error
    p2=[h1+h3,k1+k3,l1+l3];
    MM=[p1,p2,p3,D1,D3,A13];
    Mteta=cat(1,Mteta,MM);
end

case 6
sb=sin(beta);
sb2=sb*sb;
cb=cos(beta);
s1=(h1*h3/aa)+(k1*k3*sb2/bb)+(l1*l3/cc)-
((l1*h3+l3*h1)*cb/a*c);
s2=(hh1/aa)+(kk1*sb2/bb)+(ll1/cc)-(2*h1*l1*cb/a*c);
s3=(hh3/aa)+(kk3*sb2/bb)+(ll3/cc)-(2*h3*l3*cb/a*c);
costeta=s1/sqrt(s2*s3);
teta=acos(costeta)*57.295;
Dteta=abs(am-teta);
A13=[teta,Dteta];
if Dteta<=error
    p2=[h1+h3,k1+k3,l1+l3];
    MM=[p1,p2,p3,D1,D3,A13];
    Mteta=cat(1,Mteta,MM);
end

case 7
sa=sin(alfa);
sb=sin(beta);
sg=sin(gama);
sa2=sa*sa;
sb2=sb*sb;
sg2=sg*sg;
ca=cos(alfa);
cb=cos(beta);
cg=cos(gama);
f1=h1*h3*bb*cc*sa2+k1*k3*aa*cc*sb2+l1*l3*aa*bb*sg2;
f2=a*b*cc*(ca*cb-cg)*(k1*h3+h1*k3);
f3=a*bb*c*(cg*ca-cb)*(h1*l3+l1*h3);
f4=aa*b*c*(cb*cg-ca)*(k1*l3+l1*k3);
F=f1+f2+f3+f4;
a11=hh1*bb*cc*sa2+kk1*aa*cc*sb2+ll1*aa*bb*sg2;
a12=2*h1*k1*a*b*cc*(ca*cb-cg);
a13=2*h1*l1*a*bb*c*(cg*ca-cb);
a14=2*k1*l1*aa*b*c*(cb*cg-ca);
A1=a11+a12+a13+a14;

a21=hh3*bb*cc*sa2+kk3*aa*cc*sb2+l13*aa*bb*sg2;
a22=2*h3*k3*a*b*cc*(ca*cb-cg);
a23=2*h3*l3*a*bb*c*(cg*ca-cb);
a24=2*k3*l3*aa*b*c*(cb*cg-ca);

```

```

        A2=a21+a22+a23+a24;
        costeta=F/(sqrt(A1)*sqrt(A2));
        teta=acos(costeta)*57.295;
        Dteta=(am-teta);
        A13=[teta,Dteta];
        if Dteta<=error
            p2=[h1+h3,k1+k3,l1+l3];
            MM=[p1,p2,p3,D1,D3,A13];
            Mteta=cat(1,Mteta,MM);
        end
    end % termina switch
end % termina for j
end % termina for i
LMteta = size(Mteta);
if LMteta == 0
    err=err+0.01;
end
end % termina while
E=error;
[z,x] = size(Mteta);
fprintf(FILE,"Mteta = \n");
for q = 1:z
    fprintf(FILE,"[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE,",");
        end
        fprintf(FILE,"%0.2f",Mteta(q,e));
    end
    fprintf(FILE,"]\n");
end
fclose(FILE);
end % termina function

```

## Código "Calculad2.m"

```
function [Md]= Calculad2(P,L,Mabc,dm)

FILE = fopen("Calculad2.txt","w+");

[z,x] = size(P);
fprintf(FILE, "\nP = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", P(q,e));
    end
    fprintf(FILE, "]\n");
end

[z,x] = size(L);
fprintf(FILE, "L = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", L(q,e));
    end
    fprintf(FILE, "]\n");
end

[z,x] = size(Mabc);
fprintf(FILE, "Mabc = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", Mabc(q,e));
    end
    fprintf(FILE, "]\n");
end
fprintf(FILE, "dm = %.2f\n", dm);

% Esta función calcula las distancias d de acuerdo a la estructura.

a=Mabc(1,2);
b=Mabc(1,3);
c=Mabc(1,4);
```

```
fprintf(FILE, "a = %.2f\n", a);
fprintf(FILE, "b = %.2f\n", b);
fprintf(FILE, "c = %.2f\n\n", c);

aa=a*a;
bb=b*b;
cc=c*c;

fprintf(FILE, "\naa = %.2f\n", aa);
fprintf(FILE, "bb = %.2f\n", bb);
fprintf(FILE, "cc = %.2f\n\n", cc);

aai=1/aa;
bbi=1/bb;
cci=1/cc;

alfa=Mabc(2,1);
beta=Mabc(2,2);
gama=Mabc(2,3);
estructura=Mabc(1);

fprintf(FILE, "\nalfa = %.2f\n", alfa);
fprintf(FILE, "beta = %.2f\n", beta);
fprintf(FILE, "gama = %.2f\n", gama);
fprintf(FILE, "estructura = %d\n\n", estructura);

err=0.1;

fprintf(FILE, "err = %.2f\n\n", err);

Md=[];
LMd=size(Md);

while LMd==0
    error = dm * err;
    fprintf(FILE, "\nerr = %.2f\n", err);
    error = abs(error);
    fprintf(FILE, "err = %.2f\n", err);
for i=1:L
    p1=P(i,1:3);
    p2=P(i,4:6);
    p3=P(i,7:9);
    d1=P(i,10:11);
    d3=P(i,12:13);
    a13=P(i,14:15);
    h=p2(1);
    k=p2(2);
    l=p2(3);
    hh=h*h;
    kk=k*k;
    ll=l*l;

    fprintf(FILE, "\nh = %.2f\n", h);
```

```

fprintf(FILE, "k = %.2f\n", k);
fprintf(FILE, "l = %.2f\n", l);
fprintf(FILE, "hh = %.2f\n", hh);
fprintf(FILE, "kk = %.2f\n", kk);
fprintf(FILE, "ll = %.2f\n", ll);

switch estructura
    case 1
        d=sqrt((aa/(hh+kk+ll)));

    case 2
        d=sqrt(aa*cc/(cc*(hh+kk)+aa*ll));

    case 3
        d=sqrt(aa*bb*cc/(bb*cc*hh+aa*cc*kk+aa*bb*ll));

    case 4
        d=sqrt(3*aa*cc/(4*cc*(hh+h*k+kk)+3*aa*ll));

    case 5
        ca=cos(alfa);
        ca2=ca*ca;
        ta=tan(alfa/2)^2;
        x1=aa*(1+ca-2*ca2);
        x2=(1+ca)*(hh+kk+ll)-(1-ta)*(h*k+k*l+l*h);
        d=sqrt(x1/x2);

    case 6
        sb2=sin(beta)*sin(beta);
        cb=cos(beta);
        x1=hh/aa*sb2;
        x2=kk/bb;
        x3=ll/cc*sb2;
        x4=2*h*l*cb/a*c*sb2;
        x=x1*x2+x3-x4;
        d=sqrt(1/x);

    case 7
        sa=sin(alfa);
        sb=sin(beta);
        sg=sin(gama);
        sa2=sa*sa;
        sb2=sb*sb;
        sg2=sg*sg;
        ca=cos(alfa);

```

```

    cb=cos(beta);
    cg=cos(gama);
    ca2=ca*ca;
    cb2=cb*cb;
    cg2=cg*cg;
    s11=bb*cc*sa2;
    s22=aa*cc*sb2;
    s33=aa*bb*sg2;
    s12=a*b*cc*(ca*cb-cg);
    s23=aa*b*c*(cb*cg-ca);
    s31=a*bb*c*(cg*ca-cb);
    V2=(aa*bb*cc)*(1-ca2-cb2-cg2+2*ca*cb*cg);
    x=V2i*(s11*hh+s22*kk+s33*ll+2*s12*h*k+2*s23*k*l+2*s31*l*h);
    d=sqrt(V2/x);

end

Dd=abs(dm-d);
fprintf(FILE,"d = %.2f\n",d);
    fprintf(FILE,"Dd = %.2f\n",Dd);

if Dd<=error
    d2=[d,Dd];
    Sd=[p1,p2,p3,d1,d2,d3,a13];
    Md=cat(1,Md,Sd);
end
end
LMd=size(Md);
if LMd==0
    err=err+0.01;
end

end
[z,x] = size(Md);
fprintf(FILE,"\nMd = \n");
for q = 1:z
    fprintf(FILE,"[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE,",");
        end
        fprintf(FILE,"%.2f",Md(q,e));
    end
    fprintf(FILE,"]\n");
end
end
end

```

**Código "Calculateta12.m"**

```

function [Mteta]= Calculateta12(M,Mabc,am)

FILE = fopen("Calculateta12.txt","w+");
[z,x] = size(M);
fprintf(FILE,"M = \n");
for q = 1:z
    fprintf(FILE,"[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE,",");
        end
        fprintf(FILE,"% .2f",M(q,e));
    end
    fprintf(FILE,"]\n");
end

[z,x] = size(Mabc);
fprintf(FILE,"Mabc = \n");
for q = 1:z
    fprintf(FILE,"[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE,",");
        end
        fprintf(FILE,"% .2f",Mabc(q,e));
    end
    fprintf(FILE,"]\n");
end
fprintf(FILE,"am = %.2f\n",am);

% Esta función calcula el ángulo entre los dos vectores
% de acuerdo a la estructura explorada.

% Se trabaja con:
% caso: se refiere a los spots que se están analizando, 12 significa
% que se están trabajando los spots 1 y 2; 23 los spots 2 y 3.
% Pla1= Matriz de índices de Miler spot 1(caso 12) ó 2 (caso 23);
% pla2=iMatriz de índices de Miler spot 2(caso 12) ó 3 (caso 23);

% Lo que se encuentra en la Matrix Mabc:
% (1,1)= Estructura;
% (1,2)= valor del parámetro de red a;
% (1,3)= valor del parámetro de red b;
% (1,4)= valor del parámetro de red c;
% (2,1)= valor del ángulo alfa;
% (2,2)= valor del ángulo Beta;
% (2,3)= valor del ángulo gamma;
% (2,4)= 1 si la estructura es centrada en las caras 2 si es centrada
% en el cuerpo.r=ángulo calculado en radianes;

```

```

% tetag=ángulo calculado en grados;
% dc= distancia calculada;
% EZ=eje de zona;
% Solu=matriz de salida

% Pla1: es la matriz que contiene todas las combinaciones (incluyendo
% signos) de (h k l) para el spot 1

a = Mabc(1,2);
  b=Mabc(1,3);
  c=Mabc(1,4);

  fprintf(FILE,"a = %.2f\n",a);
  fprintf(FILE,"b = %.2f\n",b);
  fprintf(FILE,"c = %.2f\n\n",c);

  aa=a*a;
  bb=b*b;
  cc=c*c;

  fprintf(FILE,"\naa = %.2f\n",aa);
  fprintf(FILE,"bb = %.2f\n",bb);
  fprintf(FILE,"cc = %.2f\n\n",cc);

  alfa=Mabc(2,1);
  beta=Mabc(2,2);
  gama=Mabc(2,3);
  estructura=Mabc(1,1);

  fprintf(FILE,"\nalfa = %.2f\n",alfa);
  fprintf(FILE,"beta = %.2f\n",beta);
  fprintf(FILE,"gama = %.2f\n",gama);
  fprintf(FILE,"estructura = %d\n\n",estructura);

L=size(M);
err=0.1;
Mteta=[];
LMteta=size(Mteta);

while LMteta==0
  error = am * err;
  fprintf(FILE,"\nerr = %.2f\n",err);
  error = abs(error);
  fprintf(FILE,"err = %.2f\n",err);
  for i=1:L
    h1=M(i,1);
    k1=M(i,2);
    l1=M(i,3);
    h2=M(i,4);
    k2=M(i,5);
    l2=M(i,6);

```

```

hh1=h1*h1;
kk1=k1*k1;
ll1=l1*l1;
hh2=h2*h2;
kk2=k2*k2;
ll2=l2*l2;

A13=M(i,16:17);

fprintf(FILE, "\nh1 = %.2f\n", h1);
fprintf(FILE, "k1 = %.2f\n", k1);
fprintf(FILE, "l1 = %.2f\n", l1);
fprintf(FILE, "hh1 = %.2f\n", hh1);
fprintf(FILE, "kk1 = %.2f\n", kk1);
fprintf(FILE, "ll1 = %.2f\n", ll1);

fprintf(FILE, "h2 = %.2f\n", h2);
fprintf(FILE, "k2 = %.2f\n", k2);
fprintf(FILE, "l2 = %.2f\n", l2);
fprintf(FILE, "hh2 = %.2f\n", hh2);
fprintf(FILE, "kk2 = %.2f\n", kk2);
fprintf(FILE, "ll2 = %.2f\n", ll2);

[z,x] = size(A13);
fprintf(FILE, "A13 = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", A13(q,e));
    end
    fprintf(FILE, "]\n");
end

switch estructura

case 1

    costeta=(h1*h2+k1*k2+l1*l2)/sqrt((hh1+kk1+ll1)*(hh2+kk2+ll2));
    teta=acos(costeta)*57.295;
    Dteta=abs(am-teta);
    A12=[teta,Dteta];
    if Dteta<=error
        MM=[M(i,1:15),A12,A13];
        Mteta=cat(1,Mteta,MM);
    end

case 2

t1=((h1*h2+k1*k2)/aa)+(l1*l2/cc);

```

```

t2=((hh1+kk1)/aa)+(l11/cc);
t3=((hh2*kk2)/aa)+(l12/cc);
costeta=t1/sqrt(t2*t3);
teta=acos(costeta)*57.295;
Dteta=abs(am-teta);
A12=[teta,Dteta];
    if Dteta<=error
        MM=[M(i,1:15),A12,A13];
        Mteta=cat(1,Mteta,MM);
    end

case 3

t1=(h1*h2/aa)+(k1*k2/bb)+(l1*l2/cc);
t2=(hh1/aa)+(kk1/bb)+(l11/cc);
t3=(hh2/aa)+(kk2/bb)+(l12/cc);
costeta=t1/sqrt(t2*t3);
teta=acos(costeta)*57.295;
Dteta=abs(am-teta);
A12=[teta,Dteta];
    if Dteta<=error
        MM=[M(i,1:15),A12,A13];
        Mteta=cat(1,Mteta,MM);
    end

case 4

t1=h1*h2+k1*k2+0.5*(h1*k2+k1*h2)+(0.75*aa*l1*l2)/cc;
t2=hh1+kk1+h1*k1+(0.75*aa*l1)/cc;
t3=hh2+kk2+h2*k2+(0.75*aa*l2)/cc;
costeta=t1/sqrt(t2*t3);
teta=acos(costeta)*57.295;
Dteta=abs(am-teta);
A12=[teta,Dteta];
    if Dteta<=error
        MM=[M(i,1:15),A12,A13];
        Mteta=cat(1,Mteta,MM);
    end

case 5

h1r=(2*h1+k1+l1)/3;
k1r=(-h1+k1+l1)/3;
l1r=(-h1-2*k1+l1)/3;
h2r=(2*h2+k2+l2)/3;
k2r=(-h2+k2+l2)/3;
l2r=(-h2-2*k2+l2)/3;
hh1r=h1r*h1r;
kk1r=k1r*k1r;
ll1r=l1r*l1r;
t1r=h1r*h2r+k1r*k2r+0.5*(h1r*k2r+h2r*k1r)+(0.75*aa/cc)*l1r*l2r;
t2r=hh1r+kk1r+h1r*k1r+(0.75*aa/cc)*ll1r;
t3r=hh2r+kk2r+h2r*k2r+(0.75*aa/cc)*ll2r;

```

```

costeta=t1r/sqrt(t2r*t3r);
teta=acos(costeta)*57.295;
Dteta=abs(am-teta);
A12=[teta,Dteta];
    if Dteta<=error
        MM=[M(i,1:15),A12,A13];
        Mteta=cat(1,Mteta,MM);
    end

case 6

sb=sin(beta);
sb2=sb*sb;
cb=cos(beta);
s1=(h1*h2/aa)+(k1*k2*sb2/bb)+(l1*l3/cc)-((l1*h2+l2*h1)*cb/a*c);
s2=(hh1/aa)+(kk1*sb2/bb)+(ll1/cc)-(2*h1*l1*cb/a*c);
s3=(hh2/aa)+(kk2*sb2/bb)+(ll2/cc)-(2*h2*l2*cb/a*c);
costeta=s1/sqrt(s2*s3);
teta=acos(costeta)*57.295;
Dteta=abs(am-teta);
A12=[teta,Dteta];
    if Dteta<=error
        MM=[M(i,1:15),A12,A13];
        Mteta=cat(1,Mteta,MM);
    end

case 7
sa=sin(alfa);
sb=sin(beta);
sg=sin(gama);
sa2=sa*sa;
sb2=sb*sb;
sg2=sg*sg;
ca=cos(alfa);
cb=cos(beta);
cg=cos(gama);
f1=h1*h2*bb*cc*sa2+k1*k2*aa*cc*sb2+l1*l2*aa*bb*sg2;
f2=a*b*cc*(ca*cb-cg)*(k1*h2+h1*k2);
f3=a*bb*c*(cg*ca-cb)*(h1*l2+l1*h2);
f4=aa*b*c*(cb*cg-ca)*(k1*l2+l1*k2);
F=f1+f2+f3+f4;
a11=hh1*bb*cc*sa2+kk1*aa*cc*sb2+l11*aa*bb*sg2;
a12=2*h1*k1*a*b*cc*(ca*cb-cg);
a13=2*h1*l1*a*bb*c*(cg*ca-cb);
a14=2*k1*l1*aa*b*c*(cb*cg-ca);
A1=a11+a12+a13+a14;

a21=hh2*bb*cc*sa2+kk2*aa*cc*sb2+l12*aa*bb*sg2;
a22=2*h2*k2*a*b*cc*(ca*cb-cg);
a23=2*h2*l2*a*bb*c*(cg*ca-cb);
a24=2*k2*l2*aa*b*c*(cb*cg-ca);
A2=a21+a22+a23+a24;
costeta=F/(sqrt(A1)*sqrt(A2));

```

```
teta=acos(costeta)*57.295;
Dteta=(am-teta);
A12=[teta,Dteta];
    if Dteta<=error
        MM=[M(i,1:15),A12,A13];
        Mteta=cat(1,Mteta,MM);
    end

end % termina switch

end % Termina for i
LMteta=size(Mteta);
if LMteta==0
    err=err+0.01;
end
end % termina while
[z,x] = size(Mteta);
fprintf(FILE,"Mteta = \n");
for q = 1:z
    fprintf(FILE,"[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE,",");
        end
        fprintf(FILE,"%0.2f",Mteta(q,e));
    end
    fprintf(FILE,"]\n");
end
end % termina function
```

### Código "EjeZona.m"

```

function [PSLF]=EjeZona(M,mhk1)
    [L,~] = size(M);
    PSL=[];
    for i = 1:L
        h1=M(i,1);
        k1=M(i,2);
        l1=M(i,3);
        h2=M(i,4);
        k2=M(i,5);
        l2=M(i,6);
        h3=M(i,7);
        k3=M(i,8);
        l3=M(i,9);

        p1=[h1,k1,l1];
        p2=[h2,k2,l2];
        p3=[h3,k3,l3];
        D1=M(i,10:11);
        D2=M(i,12:13);
        D3=M(i,14:15);
        A12=M(i,16:17);
        A13=M(i,18:19);

        u=k1*l3-k3*l1;
        v=l1*h3-l3*h1;
        w=h1*k3-h3*k1;
        EZ=[u,v,w];
        EZS=sum(EZ);

        if EZS ~= 0
            u0=u*h1+v*k1+w*l1;
            v0=u*h2+v*k2+w*l2;
            w0=u*h3+v*k3+w*l3;
            EZ0=[u0 v0 w0];
            EZS0=sum(EZ0);
            if EZS0==0
                EZV=[EZ,p1,D1,p2,D2,p3,D3,A12,A13];
                PSL=cat(1,PSL,EZV);
            end
        end
    end
    LL=size(PSL);
    PSLF=[];
    for x=1:LL
        l1=max(p1);
        l2=max(p2);
        l3=max(p3);
        if l1<=mhk1 && l2<=mhk1 && l3<=mhk1
            NS=PSL(x,:);
            PSLF=cat(1,PSLF,NS);
        end
    end

```

```
end
end
end
```

**Código "escribe.m"**

```

function escribe(X,Y,ME,Mabc,dm,am,matel,elemento)

LE=size(ME);
Nsol=[1:LE]';
u=ME(:,1);
v=ME(:,2);
w=ME(:,3);
h1=ME(:,4);
k1=ME(:,5);
l1=ME(:,6);
d1=ME(:,7);
Dd1=ME(:,8);
h2=ME(:,9);
k2=ME(:,10);
l2=ME(:,11);
d2=ME(:,12);
Dd2=ME(:,13);
h3=ME(:,14);
k3=ME(:,15);
l3=ME(:,16);
d3=ME(:,17);
Dd3=ME(:,18);
a12=ME(:,19);
Da12=ME(:,20);
a13=ME(:,21);
Da13=ME(:,22);

format short
a=Mabc(1,2);
b=Mabc(1,3);
c=Mabc(1,4);
Vabc=[a;b;c];
alfa=Mabc(2,1)*57.295;
beta=Mabc(2,2)*57.295;
gama=Mabc(2,3)*57.295;
Vang=[alfa;beta;gama];

d1 = round(d1,4);
Dd1 = round(Dd1,4);
d2 = round(d2,4);
Dd2 = round(Dd1,4);
d3 = round(d3,4);
Dd3 = round(Dd3,4);
a12 = round(a12,4);
Da12 = round(Da12,4);
a13 = round(a13,4);
Da13 = round(Da13,4);

format short
d_medida=[dm];

```

```

Ang_medido=[am];

nomXYdA= {' ', 'x', 'y', 'd medida', 'Ang medido'};
names={'Punto 1'; 'Punto 2'; 'Punto 3'};
TXYdA=table(names,X',Y',d_medida',Ang_medido');
CXYdA=table2cell(TXYdA);
XYdA=[nomXYdA;CXYdA];
writecell(XYdA,matel, "Sheet", elemento, "Range", "O3");

nomabc={'a,b,c', 'alfa,beta,gama'};
Tabc=table(Vabc,Vang);
Cabc=table2cell(Tabc);
abc=[nomabc;Cabc];
writecell(abc,matel, "Sheet", elemento, "Range", "U3");

nomP={'Solución', 'u', 'v', 'w', 'h1', 'k1', 'l1', 'h2', 'k2', 'l2', 'h3', 'k3', ...
      'l3', 'd1', 'Dd1', 'd2', 'Dd2', 'd3', 'Dd3', 'ang1', 'Dang1', 'ang3', 'Dang3'};
TP=table(Nsol,u,v,w,h1,k1,l1,h2,k2,l2,h3,k3,l3,d1,Dd1,d2,Dd2,d3,Dd3, ...
        a12,Da12,a13,Da13);
CP=table2cell(TP);
P=[nomP;CP];
writecell(P,matel, 'sheet', elemento, 'Range', 'A10');

end

```

### Código "Extincion.m"

```

function [MF]=Extincion(M,L,red)

% Si la estructura es:
% Primitiva, entonces red=0 y no pasa por esta función.
% Centrada en las caras (fcc), entonces es el caso 1.
% Centrada en el cuerpo (bcc), entonces es el caso 2.
% Centrada en 2 caras paralelas, entonces es el caso 3.

MF=[];

for i=1:L

    p1=[M(i,4)^2,M(i,5)^2,M(i,6)^2];
    p2=[M(i,9)^2,M(i,10)^2,M(i,11)^2];
    p3=[M(i,14)^2,M(i,15)^2,M(i,16)^2];
    s1=sum(p1);
    s2=sum(p2);
    s3=sum(p3);
    r1=rem(s1,2);
    r2=rem(s2,2);
    r3=rem(s3,2);

    switch red

        case 1 % Red tipo F (en el caso cúbico: fcc)

            if r1==0 || r1==3
                if r2==0 || r2==3
                    if r3==0 || r3==3
                        MF=cat(1,MF,M(i,:));
                    end
                end
            end

        case 2 % Red tipo F (en el caso cúbico: bcc)

            if r1==0 || r2==0 || r3==0
                MF=cat(1,MF,M(i,:));
            end

        case 3 % Red tipo A

            a1=rem(M(i,5)^2+M(i,6)^2); % (k+1)
            a2=rem(M(i,10)^2+M(i,11)^2);
            a3=rem(M(i,15)^2+M(i,16)^2);
            A1=rem(a1,2);
            A2=rem(a2,2);
            A3=rem(a3,2);
    end
end

```

```
b1=rem(M(i,4)^2+M(i,6)^2); % (h+1)
b2=rem(M(i,9)^2+M(i,11)^2);
b3=rem(M(i,14)^2+M(i,16)^2);
B1=rem(b1,2);
B2=rem(b2,2);
B3=rem(b3,2);

c1=rem(M(i,4)^2+M(i,5)^2); % (h+k)
c2=rem(M(i,9)^2+M(i,10)^2);
c3=rem(M(i,14)^2+M(i,15)^2);
C1=rem(c1,2);
C2=rem(c2,2);
C3=rem(c3,2);

if A1==1 && A2==1 && A3==1
    MF=cat(1,MF,M(i,:));
end

if B1==1 && B2==1 && B3==1
    MF=cat(1,MF,M(i,:));
end

if C1==1 && C2==1 && C3==1
    MF=cat(1,MF,M(i,:));
end
end
end
end
```

## Código "main.m"

```

function main(h)
try
waitbar(0.1,h,"Recuperando información");
FILE = fopen("log.txt","w+");

FILE2 = fopen("tmp_plantilla.txt","r");

FILE3 = fopen("error.txt","w+");

arch_temp = fscanf(FILE2,"%s");

punto1 = readmatrix(arch_temp,"OutputType","string","Range","C3:D3");
punto2 = readmatrix(arch_temp,"OutputType","string","Range","C4:D4");
punto3 = readmatrix(arch_temp,"OutputType","string","Range","C5:D5");
Val1 = 1;
mhk1 = readmatrix(arch_temp,"Range","G3:G3");
tabla = readmatrix(arch_temp,"OutputType","string","Range","H3:H3");
valuecase = readmatrix(arch_temp,"Range","I3:I3");

if Val1 ~= 1
    Val1 = 0;
end

if valuecase == 1
    nombre = "E_Cubica.xls";
elseif valuecase == 2
    nombre = "E_Tetragonal.xls";
elseif valuecase == 3
    nombre = "E_Ortorombica.xls";
elseif valuecase == 4
    nombre = "E_Hexagonal.xls";
elseif valuecase == 5
    nombre = "E_Rombohedral.xls";
elseif valuecase == 6
    nombre = "E_Monoclinica.xls";
elseif valuecase == 7
    nombre = "E_Triclinica.xls";
end

fprintf(FILE,"arch_temp = %s\n",arch_temp);
fprintf(FILE,"mhk1 = %d\n",mhk1);
fprintf(FILE,"tabla = %s\n",tabla);
fprintf(FILE,"valuecase = %d\n",valuecase);
fprintf(FILE,"nombre = %s\n",nombre);
fprintf(FILE,"Val1 = %d\n",Val1);
fprintf(FILE,"punto1 = [%f,%f]\n",punto1(1),punto1(2));
fprintf(FILE,"punto2 = [%f,%f]\n",punto2(1),punto2(2));
fprintf(FILE,"punto3 = [%f,%f]\n\n",punto3(1),punto3(2));

```

```

waitbar(0.2,h,"Calculando distancias...");
[dm,am,spots] = obtenciondatos(punto1,punto2,punto3,Val1);

fprintf(FILE,"am = [%f,%f,%f]\n",am(1),am(2),am(3));
fprintf(FILE,"dm = [%f,%f,%f]\n",dm(1),dm(2),dm(3));
fprintf(FILE,"spots = [%f,%f]\n",spots(1,1),spots(1,2));
fprintf(FILE,"spots = [%f,%f]\n",spots(2,1),spots(2,2));
fprintf(FILE,"spots = [%f,%f]\n\n",spots(3,1),spots(3,2));

if Val1 ~= 2
    X=[spots(1:3)];
    Y=[spots(4:6)];
end

MRX = readmatrix(nombre,'sheet',tabla);
[m,z] = size(MRX);
np = m-2;

fprintf(FILE,"MRX = \n");
for q = 1:m
    fprintf(FILE,"[");
    for e = 1:z
        if e ~= 1
            fprintf(FILE,",");
        end
        fprintf(FILE,"% .2f",MRX(q,e));
    end
    fprintf(FILE,"]\n");
end

fprintf(FILE,"Tamaño = %d\n",m);
fprintf(FILE,"np = %d\n\n",np);

Mabc = MRX(1:2,:);
MPlaRx = MRX(3:m,:);

[z,x] = size(Mabc);
fprintf(FILE,"Mabc = \n");
for q = 1:z
    fprintf(FILE,"[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE,",");
        end
        fprintf(FILE,"% .2f",Mabc(q,e));
    end
    fprintf(FILE,"]\n");
end
[z,x] = size(MPlaRx);
fprintf(FILE,"MPlaRx = \n");
for q = 1:z
    fprintf(FILE,"[");

```

```

    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", MPlaRx(q,e));
    end
    fprintf(FILE, "]\n");
end

red=Mabc(2,4);
est=Mabc(1,1);

fprintf(FILE, "\nred = %d\n", red);
fprintf(FILE, "est = %d\n", est);

waitbar(0.4,h, "Calculando...");

M1=Compara(dm(1), np, MPlaRx);
L1=size(M1);
D1=Calculad13(M1, L1, Mabc, dm(1));
MD1=unique(D1, 'rows', 'stable');

[z,x] = size(M1);
fprintf(FILE, "\nM1 = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", M1(q,e));
    end
    fprintf(FILE, "]\n");
end
[z,x] = size(D1);
fprintf(FILE, "D1 = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", D1(q,e));
    end
    fprintf(FILE, "]\n");
end
[z,x] = size(MD1);
fprintf(FILE, "MD1 = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");

```

```

        end
        fprintf(FILE, "%.2f", MD1(q,e));
    end
    fprintf(FILE, "]\n");
end

M3=Compara(dm(3),np,MPlarX);
L3=size(M3);
D3=Calculad13(M3,L3,Mabc,dm(3));
MD3=unique(D3,'rows','stable');

[z,x] = size(M3);
fprintf(FILE, "\nM3 = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", M3(q,e));
    end
    fprintf(FILE, "]\n");
end
[z,x] = size(D3);
fprintf(FILE, "D3 = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", D3(q,e));
    end
    fprintf(FILE, "]\n");
end
[z,x] = size(MD3);
fprintf(FILE, "MD3 = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", MD3(q,e));
    end
    fprintf(FILE, "]\n");
end

DA13=Calculateta13(MD1,MD3,Mabc,am(3));
MDA13=unique(DA13,'rows','stable');
LA13=size(MDA13);

[z,x] = size(DA13);

```

```

fprintf(FILE, "\nDA13 = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", DA13(q,e));
    end
    fprintf(FILE, "]\n");
end
[z,x] = size(MDA13);
fprintf(FILE, "MDA13 = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", MDA13(q,e));
    end
    fprintf(FILE, "]\n");
end

MD123=Calculad2(MDA13, LA13, Mabc, dm(2));
LD2=size(MD123);

[z,x] = size(MD123);
fprintf(FILE, "\nMD123 = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", MD123(q,e));
    end
    fprintf(FILE, "]\n");
end

MDA=Calculateta12(MD123, Mabc, am(2));
DA=unique(MDA, 'rows', 'stable');
LDA = size(DA);

waitbar(0.8,h, "Guardando...");

[z,x] = size(MDA);
fprintf(FILE, "\nMDA = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", MDA(q,e));
    end
    fprintf(FILE, "]\n");
end

```

```
        end
        fprintf(FILE, "%.2f", MDA(q,e));
    end
    fprintf(FILE, "]\n");
end

FDA=EjeZona(DA,mhk1);
LFAD=size(FDA);

[z,x] = size(FDA);
fprintf(FILE, "\nFDA = \n");
for q = 1:z
    fprintf(FILE, "[");
    for e = 1:x
        if e ~= 1
            fprintf(FILE, ",");
        end
        fprintf(FILE, "%.2f", FDA(q,e));
    end
    fprintf(FILE, "]\n");
end

if red==0
    FFDA=FDA;
else
    FFDA=Extincion(FDA,LFAD,red);
end
matel = "salida.xlsx";
escribe(X,Y,FDA,Mabc,dm,am,matel,tabla);
fclose(FILE);
fclose(FILE3);
catch ME
    fclose(FILE);
    causa = string(ME.cause);
    fecha = datetime("now", "InputFormat", "uuuu-MM-dd'T'HH:mm:ss.SSS");
    fprintf(FILE3, "Error en proceso\n");
    fprintf(FILE3, "%s\n", fecha);
    fprintf(FILE3, "%s\n", ME.identifier);
    fprintf(FILE3, "%s\n", ME.message);
    fprintf(FILE3, "%s\n", causa);
    fprintf(FILE3, "%s\n", ME.getReport);
    fclose(FILE3);
end
end
```

## Código "PIPD2.m"

```

function varargout = PIPD2(varargin)
% PIPD2 MATLAB code for PIPD2.fig
%   PIPD2, by itself, creates a new PIPD2 or raises the existing
%   singleton*.
%
%   H = PIPD2 returns the handle to a new PIPD2 or the handle to
%   the existing singleton*.
%
%   PIPD2('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in PIPD2.M with the given input arguments.
%
%   PIPD2('Property','Value',...) creates a new PIPD2 or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before PIPD2_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to PIPD2_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help PIPD2

% Last Modified by GUIDE v2.5 26-Nov-2023 15:14:28

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @PIP2_OpeningFcn, ...
                  'gui_OutputFcn',  @PIP2_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before PIPD2 is made visible.
function PIP2_OpeningFcn(hObject, eventdata, handles, varargin)
axes('Units', 'normalized', 'Position',[0 0 1 1]);
[x,map] = imread('PantallaGUIDE.png','png');

```

```

image(x)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to PIPD2 (see VARARGIN)

% Choose default command line output for PIPD2
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes PIPD2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = PIPD2_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
[filename, pathname] = uigetfile({'*.xlsx','Abrir Archivo de Excel'}, 'Selecciona el
Archivo');
FILE = fopen("tmp_plantilla.txt","w");
fprintf(FILE,"%s%s",pathname,filename);
fclose(FILE);
set(handles.text3,'String',filename);

% warndlg('Mensaje de aviso','Mensaje de Usuario');
% helpdlg({'Mensaje'} ,{'de'}, {'ayuda'},'Mensaje de usuario');
% msgbox('Mensaje','Mensaje de Usuario','modal');
%
% [x,map] = imread('img.jpg');
% msgbox('Mensaje de aviso','Mensaje de Usuario','costum',x,map,'modal');
%[M, string, full] = xlsread([pathname, filename]);

% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)

```

---

```

opc=questdlg('¿Desea salir del programa?','SALIR','Si','No','No');
if strcmp(opc, 'No')
    return
end
clear, clc, close all;
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
plantilla = get(handles.text3,'String');
if isempty(plantilla)
    warndlg('No ha selecciona plantilla','Mensaje de Usuario');
else
    h = waitbar(0,'Ejecutando, por espere');
    pause(.5);
    main(h);
    waitbar(1,h,"Terminando...");
    pause(1);
    close(h);
    FILE2 = fopen("error.txt","r");
        txt_error = fscanf(FILE2,"%s");
    fclose(FILE2);
    if isempty(txt_error)
        msgbox('Proceso terminado','Mensaje de Usuario','modal');
        set(handles.text3,'String','');
        !salida.xlsx;
    else
        warndlg({'El procesos no pudo terminar','Revisar archivo de
error.txt'},'Mensaje de Usuario');
        set(handles.text3,'String','Error, revisar error.txt');
    end
end
end
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

function text3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

---

## Bibliografía

- [1] *Interpretation of electron diffraction patterns*, K.W. Andrews, D.J. Dyson, S.R. Keown, Hilger & Watts LTD, Plenum US, (1967), 5 – 129.
- [2] *Estructura de los sólidos, cristalografía, difracción y defectos*, Meléndez Martínez Juan José, Badajoz, España: Universidad de Extremadura, (2021), 19 – 153.
- [3] *Indexing of electron diffraction spot patterns by computer*, B. L. Rhoades, *Microm*, 6, (1975), 123-127.
- [4] *Computer-Aided indexing of transmission electron diffraction patterns*, Raymond P. Goehner and Prakash Rao, *Metallography*, 10, (1977), 415-424.
- [5] *A computer program for unique indexing of electron diffraction patterns*, W. Prantj, Erich-Schid, *J. Appl. Cryst.* 17, (1984), 39-42.
- [6] *JECP/PCED—a computer program for simulation of polycrystalline electron diffraction pattern and phase identification*, X.Z. Li, *Ultramicroscopy*, 99, (2004), 257-261.
- [7] *A program for phase identification using diffractograms obtained from TEM structure images*, R. Galicia, R. Herrera, J. L. Rius, C. Zorrilla, and A. Gómez, *Revista Mexicana de Física*, 59, (2013), 102-106.
- [8] Francis A. Jenkins, Harvey E. White. *Fundamentals of Optics*. MacGraw-Hill Primls, (2001).
- [9] DigitalMicrograph 3.4 User's Guide, Gatan, Inc. (1999).
- [10] Waseda Yoshio, Matsubara Eiichiro, Shinoda Kozo. *X-Ray Diffraction Crystallography*. Tokio, Japón: Universidad de Tohoku, (2011), 1 – 80.
- [11] The MathWorks, Inc. (2022). *MATLAB versión: 9.13.0 (R2023b)*. Consultado: Octubre 17, (2023). Disponible: <https://www.mathworks.com>
- [12] J. Kirkland Earl. *Advanced Computing in Electron Microscopy*. SpringerLink, (2010).
- [13] Notas de Física, M. J. Yacamán y Alfredo Gómez, (1981), Instituto de Física, vol. 4 n°. 4.
- [14] *Nano phase characterization by transmission electron microscopy: experimental and simulation*, Francisco Miguel Ascencio Aguirre, Lourdes Bazán-Díaz, Rubén Mendoza-Cruz, Alfredo Gómez Rodríguez, Cristina Zorrilla-Cangas, Raúl Herrera-Becerra, (2015), *Materials Sciences and Applications*, 6(11), pp. 935-942.