

DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DE UN MÓDULO COMPUTACIONAL
PARA EL ANÁLISIS DE MOMENTO ANGULAR FONÓNICO (PAM/PAM-DOS)
INTEGRADO EN LA BIBLIOTECA DE CÓDIGO ABIERTO PHONOPY.

BRAYAN RODOLFO BARAJAS OCHOA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA

2026

DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DE UN MÓDULO COMPUTACIONAL
PARA EL ANÁLISIS DE MOMENTO ANGULAR FONÓNICO (PAM/PAM-DOS)
INTEGRADO EN LA BIBLIOTECA DE CÓDIGO ABIERTO PHONOPY.

BRAYAN RODOLFO BARAJAS OCHOA

Trabajo de Grado para optar al título de Ingeniero de Sistemas

Director

Lola Xiomara Bautista Rozo

Doctora

Codirector

Andrés Camilo García Castro

Doctor

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA

2026

DEDICATORIA

A mi madre, Emma, pilar fundamental en mi vida. Su sacrificio incondicional y su fe inquebrantable en mis capacidades hicieron posible la culminación de este trabajo.

AGRADECIMIENTOS

El desarrollo de esta tesis ha sido un camino enriquecido por el apoyo y la guía de numerosas personas e instituciones, a quienes deseo expresar mi más profundo agradecimiento.

En primer lugar, mi más sincero agradecimiento a mi familia. A mis padres, Emma Ochoa y Rodolfo Barajas, a nuestra hermana, Lorent, por su amor, paciencia y apoyo incondicional, que han sido el pilar fundamental a lo largo de toda mi formación, y a mis gatos, Murphy y Nomi, cuya silenciosa compañía y travesuras alegraron las largas horas dedicadas a esta investigación. Extiendo un recuerdo especial y agradecido a mi abuela, Rogelia, que en paz descansa, por sus enseñanzas y el tiempo invaluable que me dedicó.

En el ámbito académico, mi reconocimiento y gratitud se dirigen de manera especial a mis directores, el profesor Andrés Camilo García Castro y la profesora Lola Xiomara Bautista. Su mentoría, paciencia y rigor científico guiaron el desarrollo técnico de esta investigación e inculcaron valores fundamentales para mi vida. Agradezco profundamente su disposición constante y la oportunidad de haber sido su estudiante.

Asimismo, extiendo mi agradecimiento a los profesores de la Escuela de Ingeniería de Sistemas y la Escuela de Física de la Universidad Industrial de Santander, cuyas enseñanzas sentaron las bases para este trabajo.

Finalmente, agradezco al grupo de investigación en Física de la Materia Condensada Computacional (FICOMACO) y a la Universidad Industrial de Santander por proporcionar la infraestructura y los recursos necesarios para la realización de esta investigación.

CONTENIDO

	pág.
INTRODUCCIÓN	15
1 MARCO DE REFERENCIA Y ESTADO DEL ARTE	18
1.1 Marco Teórico	18
1.1.1 Fundamentos de Ingeniería de Software para Cómputo Científico	18
1.1.2 Contexto Tecnológico: Ecosistema de Cómputo Científico en Python	21
1.1.3 Fundamentos Físicos para la Implementación.	21
1.2 Estado del Arte	25
2 ANÁLISIS DEL SISTEMA: ARQUITECTURA DE PHONOPY	27
2.1 Propósito y alcance	27
2.2 Flujo de alto nivel	28
2.3 Modelo de datos y transformaciones	29
2.3.1 Estructuras de datos internas:	29
2.3.2 Tres celdas: unitaria, supercelda y primitiva	30
2.3.3 Simetría y uso de <code>spglib</code>	30
2.3.4 Problema de autovalores fonónicos	31
2.4 Cálculos de bandas, malla y DOS	31
2.4.1 Estructura de bandas	31
2.4.2 Malla (<i>mesh</i>) y DOS	31
2.5 Rendimiento e implementación	31
2.6 Extensibilidad y puntos de integración	32
2.7 Patrones de uso práctico	32

2.7.1	Flujo orientado al CLI	32
2.7.2	Flujo orientado a la API.	33
3	Diseño e Implementación del Módulo PAM	35
3.0.1	Análisis de Requisitos:	35
3.0.2	Decisiones de Diseño:	38
3.0.3	Estructura de Módulos:	40
3.1	Implementación del Módulo <code>pam.py</code>	41
3.1.1	Función <code>phonon_angular_momentum</code>	42
3.1.2	Consideraciones Numéricas:	47
3.1.3	Tratamiento de la Temperatura:	49
3.2	Implementación de PAMDoS en <code>dos.py</code>	50
3.2.1	Arquitectura de la Clase <code>PAMDoS</code>	51
3.2.2	Algoritmo de Separación por Signo y Ponderación por Magnitud:	52
3.2.3	Adaptación de los Métodos de Cálculo de DOS:	55
3.2.4	Integración de la Densidad de Estados:	57
3.3	Interfaz de Usuario y Comandos	58
3.3.1	Nuevos Parámetros de Línea de Comandos:	59
3.3.2	Integración con <code>phonopy-load</code> :	60
3.3.3	Artefactos de salida	60
3.3.4	Compatibilidad y Versionado:	62
3.4	Herramientas Adicionales de Visualización	63
3.4.1	Arquitectura y Flujo de Procesamiento de Datos	64
3.4.2	Algoritmo de Cálculo de Pesos Elementales	65
3.4.3	Algoritmos de Coloración:	66
3.5	Documentación y Guías de Uso	68
3.5.1	Documentación para el Usuario y la Comunidad	69
3.5.2	Ejemplo Práctico y Cuaderno Interactivo	70

4 VERIFICACIÓN Y VALIDACIÓN	72
4.1 Propósito y alcance	72
4.2 Verificación del software	72
4.2.1 Estrategia de pruebas y criterios de aceptación	72
4.2.2 Implementación de Pruebas Unitarias	73
4.2.3 Pruebas de integración (CLI)	74
4.3 Validación física	75
5 Resultados: Caso de Estudio y Demostración (TaSi₂)	76
5.1 Detalles metodológicos de los cálculos computacionales.	76
5.1.1 Cálculos de Primeros Principios: Opt. Estructural y Prop. Electrónicas.	76
5.1.2 Cálculos Fonónicos: Constantes de Fuerza.	78
5.1.3 Post-procesamiento: Dispersión Fonónica y Momento Angular	78
5.2 Análisis Estadístico: Densidad de Estados Resuelta en PAM	81
5.3 Análisis Específico: Bandas Coloreadas por PAM	83
5.4 Interpretación de Origen: Interoperabilidad con Proyecciones Nativas	86
5.5 Validación Visual: Exportación de Datos para Animación	87
6 CONCLUSIONES Y TRABAJO FUTURO	89
6.1 Conclusiones	89
6.2 Trabajo Futuro	90
BIBLIOGRAFÍA	94

LISTA DE FIGURAS

	pág.	
Figura 1	Flujo de datos y procesos en un cálculo típico con Phonopy ¹ .	22
Figura 2	Ejemplo de dispersión de fonones: frecuencias por trayectoria en el espacio recíproco.	23
Figura 3	<i>Estructura cristalina del compuesto TaSi₂.</i>	25
Figura 4	<i>Diagrama de clases UML simplificado.</i>	40
Figura 5	<i>Diagrama de flujo que ilustra el proceso algorítmico de la función <code>phonon_angular_momentum</code>.</i>	45
Figura 6	<i>Visualización del proceso de integración del PAM-DOS.</i>	58
Figura 7	<i>Árbol de decisión procesal que ilustra la lógica de ejecución del <code>script phonopy-load</code> modificado.</i>	61
Figura 8	<i>Visualización de la metodología para obtener las gráficas haciendo uso de VASP, Phonopy y el código propio implementado.</i>	80
Figura 9	PAM-DOS para TaSi ₂ (SG 180). Las líneas continuas (PAM >0) y discontinuas (PAM <0) muestran una distribución no trivial de quiralidad a lo largo del espectro de frecuencias.	81
Figura 10	PAM-DOS para TaSi ₂ (SG 181). La estructura es casi idéntica a la de su enantiómero (Fig. 9), un primer indicio de su relación especular.	82

¹ Atsushi Togo. "First-principles Phonon Calculations with Phonopy and Phono3py". En: *J. Phys. Soc. Jpn.* 92.1 (2023), pág. 012001. DOI: 10.7566/JPSJ.92.012001.

- Figura 11 Bandas fonónicas y PAM-DOS para TaSi₂ (SG 180). El color de las bandas (rojo: PAM >0, azul: PAM <0) revela la localización del PAM en el espacio recíproco. 84
- Figura 12 Bandas fonónicas y PAM-DOS para TaSi₂ (SG 181). La inversión sistemática del color de las bandas es la prueba definitiva de la quiralidad opuesta y valida la correctitud física del software. 85
- Figura 13 Bandas fonónicas proyectadas por especie atómica (TaSi₂, SG 180). Las bajas frecuencias están dominadas por Ta (ocre) y las altas por Si (azul). Esta información permite inferir el origen atómico de los modos quirales. 86
- Figura 14 Secuencia de desplazamiento para un modo quiral en SG 180. Se observa un sentido de rotación horario, consistente con un PAM positivo calculado por el módulo. 87
- Figura 15 Secuencia de desplazamiento para el mismo modo en SG 181. El sentido de rotación es ahora antihorario, una confirmación visual directa de la inversión del signo del PAM. 88
- Figura 16 **Secuencia de desplazamiento del modo fonónico correspondiente a la banda 7 para el grupo de simetría 180, con frecuencia 157,866052 [cm⁻¹] y momento angular en z ($l_z/\hbar = -4,785034 e - 01$).** 100
- Figura 17 **Secuencia de desplazamiento del modo fonónico correspondiente a la banda 7 para el grupo de simetría 181, con frecuencia 157,853592 [cm⁻¹] y momento angular en z ($l_z/\hbar = 4,785024 e - 01$).** 100
- Figura 18 **Secuencia de desplazamiento del modo fonónico correspondiente a la banda 19 para el grupo de simetría 180, con frecuencia 323,838851 [cm⁻¹] y momento angular en z ($l_z/\hbar = 4,465218 e - 01$).** 101
- Figura 19 **Secuencia de desplazamiento del modo fonónico correspondiente a la banda 19 para el grupo de simetría 181, con frecuencia 323,483449 [cm⁻¹] y momento angular en z ($l_z/\hbar = -4,452149 e - 01$).** 101

LISTA DE TABLAS

	pág.
Tabla 1 <i>Análisis de la complejidad computacional del cálculo del PAM.</i>	49
Tabla 2 <i>Comparación de los métodos de cálculo de DOS implementados para PAM-DOS.</i>	56
Tabla 3 <i>Resumen de los nuevos parámetros de línea de comandos para el cálculo del PAM.</i>	59
Tabla 4 <i>Datos de frecuencia y momento angular en z para los fonones correspondientes al grupo de simetría 180 en el punto (0, 0, 1/4).</i>	98
Tabla 5 <i>Datos de frecuencia y momento angular en z para los fonones correspondientes al grupo de simetría 181 en el punto (0, 0, 1/4).</i>	99
Tabla 6 <i>Clasificación de materiales según la simetría de inversión espacial (P) y temporal (T), y su capacidad para albergar fonones con momento angular neto.</i>	115

LISTA DE ANEXOS

	pág.
Anexo A Datos suplementarios: tablas y secuencias de modos fonónicos	97
Anexo B Derivaciones matemáticas para la expresión del momento angular fonónico	102
Anexo C Análisis físico detallado de los fonones quirales en TaSi ₂	113

RESUMEN

TÍTULO: DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DE UN MÓDULO COMPUTACIONAL PARA EL ANÁLISIS DE MOMENTO ANGULAR FONÓNICO (PAM/PAM-DOS) INTEGRADO EN LA BIBLIOTECA DE CÓDIGO ABIERTO PHONOPY *

AUTORES: BRAYAN RODOLFO BARAJAS OCHOA **

PALABRAS CLAVE: Phonopy, Python, NumPy, Momento Angular Fonónico, PAM-DOS, CLI, Ingeniería de software científico.

DESCRIPCIÓN:

El análisis del Momento Angular Fonónico (PAM), una propiedad clave para entender fenómenos cuánticos en materiales, ha estado limitado por la falta de herramientas computacionales integradas, forzando a los investigadores a usar flujos de trabajo ad-hoc, poco reproducibles y difíciles de mantener. Este trabajo de grado resuelve este problema desde la ingeniería de sistemas, abordando el ciclo de vida completo para desarrollar un módulo de software de código abierto, robusto y validado para el análisis de PAM, integrado en la biblioteca Phonopy.

La solución se materializa en un artefacto de software cuyo diseño arquitectónico no intrusivo respeta la integridad del ecosistema Phonopy, garantizando la mantenibilidad y facilitando su futura adopción por la comunidad. Su núcleo de cálculo, implementado eficientemente en Python y NumPy, utiliza vectorización para permitir el análisis de sistemas a escala realista. La funcionalidad se expone a través de una interfaz de línea de comandos (CLI) diseñada para ser coherente e intuitiva, democratizando el acceso a un cálculo físico complejo.

El resultado es un instrumento computacional fiable, cuya correctitud fue rigurosamente comprobada mediante un proceso de Verificación y Validación (V&V). La validación física, utilizando el material quiral TaSi₂ como caso de estudio, demostró que el módulo reproduce con éxito las firmas físicas esperadas, como la inversión de signo del PAM entre enantiómeros. Esta evidencia confirma la validez del software como herramienta científica. Finalmente, este proyecto no solo entrega una extensión funcional para Phonopy, sino que establece un precedente para la integración de nuevas teorías en plataformas de software científico existentes, potenciando la investigación futura en la física de materiales cuánticos.

* Trabajo de grado

** Escuela de Ingeniería de Sistemas e Informática. Directora: Lola Xiomara Bautista Rozo, Doctora; Co-director: Andrés Camilo García Castro, Doctor.

ABSTRACT

TITLE: DESIGN, IMPLEMENTATION AND VALIDATION OF A COMPUTATIONAL MODULE FOR PHONON ANGULAR MOMENTUM ANALYSIS (PAM/PAM-DOS) INTEGRATED INTO THE OPEN-SOURCE PHONOPY LIBRARY *

AUTHORS: BRAYAN RODOLFO BARAJAS OCHOA **

KEYWORDS: Phonopy, Python, NumPy, Phonon Angular Momentum, PAM-DOS, CLI, Scientific software engineering.

DESCRIPTION:

The analysis of Phonon Angular Momentum (PAM), a key property for understanding quantum phenomena in materials, has been hindered by the lack of integrated computational tools, forcing researchers to rely on ad-hoc, poorly reproducible, and hard-to-maintain workflows. This thesis solves this problem from a systems engineering perspective, addressing the full lifecycle to develop a robust, validated, open-source software module for PAM analysis, integrated into the `Phonopy` library.

The solution materializes as a software artifact whose non-intrusive architectural design respects the integrity of the `Phonopy` ecosystem, ensuring maintainability and facilitating its future adoption by the community. Its computation core, efficiently implemented in Python and `NumPy`, uses vectorization to enable the analysis of realistic-scale systems. The functionality is exposed through a command-line interface (CLI) designed to be coherent and intuitive, democratizing access to a complex physical calculation.

The result is a reliable computational instrument, whose correctness was rigorously verified through a Verification and Validation (V&V) process. Physical validation, using the chiral material TaSi_2 as a case study, demonstrated that the module successfully reproduces expected physical signatures, such as the sign inversion of PAM between enantiomers. This evidence confirms the software's validity as a scientific tool. Ultimately, this project not only delivers a functional extension for `Phonopy` but also sets a precedent for integrating new theories into existing scientific software platforms, empowering future research in quantum materials physics.

* Bachelor Thesis

** School of Systems and Informatics Engineering. Advisor: Lola Xiomara Bautista Rozo, Doctor; Co-advisor: Andrés Camilo García Castro, Doctor.

INTRODUCCIÓN

El avance de la ciencia moderna depende intrínsecamente del desarrollo de software robusto, extensible y de código abierto. En el cómputo científico, la mantenibilidad y la capacidad de integrar nuevos modelos dentro de bibliotecas existentes son retos de ingeniería clave. Este trabajo se enmarca en ese contexto y aborda la extensión de *Phonopy*, una biblioteca en Python ampliamente adoptada para el cálculo de propiedades fonónicas¹. El objetivo central no es un descubrimiento físico per se, sino el diseño, la implementación y la validación de un módulo de software para el análisis del Momento Angular Fonónico (PAM) y su densidad de estados (PAM-DOS), integrado de forma coherente en el ecosistema de *Phonopy*.

El dominio de aplicación es el estudio de los fonones, excitaciones vibracionales cuantizadas que gobiernan propiedades macroscópicas como la conductividad térmica y la estabilidad estructural. En materiales quirales, estructuras cristalinas sin inversión espacial, emerge el interés por los llamados “fonones quirales”, modos vibracionales con momento angular intrínseco (PAM)²³. Aunque el formalismo físico para cuantificar el PAM está documentado en la literatura, *Phonopy* carecía de un módulo integrado que permitiera su cálculo y análisis reproducible dentro de su flujo de trabajo estándar. Esta carencia ha llevado a soluciones *ad-hoc* y fragmentadas, con riesgos de errores, costos de mantenimiento elevados y barreras para la reproducibilidad científica.

Desde la perspectiva de ingeniería de sistemas, este trabajo plantea una solución integral

-
- ¹ Atsushi Togo. “First-principles phonon calculations with phonopy and phono3py”. En: *Journal of the Physical Society of Japan* 92.1 (2023). <https://journals.jps.jp/doi/full/10.7566/JPSJ.92.012001>, pág. 012001.
 - ² Gerhard H Fecher, Jürgen Kübler y Claudia Felser. “Chirality in the solid state: Chiral crystal structures in chiral and achiral space groups”. En: *Materials* 15.17 (2022). <https://www.mdpi.com/1996-1944/15/17/5812>, pág. 5812.
 - ³ Lifa Zhang y Qian Niu. “Angular momentum of phonons and the Einstein–de Haas effect”. En: *Physical Review Letters* 112.8 (2014). <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.112.085503>, pág. 085503.

que recorre el ciclo de vida de desarrollo de software: análisis del sistema base (arquitectura de *Phonopy*), diseño modular y no intrusivo, implementación eficiente en Python con *NumPy* (enfazando la vectorización) e integración con la interfaz de línea de comandos (CLI) de la biblioteca. Se define una API clara para el núcleo de cálculo del PAM con contratos explícitos de entrada/salida en arreglos de *NumPy*, promoviendo bajo acoplamiento y alta cohesión, en línea con buenas prácticas de ingeniería para software científico ⁴.

El proceso se completa con una estrategia de Verificación y Validación (V&V): (i) verificación funcional mediante pruebas unitarias conceptuales y pruebas de integración para garantizar la correctitud del código respecto a las especificaciones; y (ii) validación física de resultados utilizando un caso de estudio representativo, TaSi_2 , donde se comprueba la coherencia de las predicciones, por ejemplo, la inversión de signo del PAM entre enantiómeros, como demostración de que el módulo implementado reproduce comportamientos físicos esperados. Enfatizamos que TaSi_2 actúa como banco de pruebas para la herramienta, no como fin último de investigación.

Las contribuciones de ingeniería de este trabajo son:

- Un diseño arquitectónico modular y no intrusivo para extender *Phonopy* con capacidades de PAM/PAM-DOS.
- Un núcleo de cálculo vectorizado en Python/*NumPy* para el PAM con API estable y documentada.
- La integración con la CLI de *Phonopy*, ampliando el flujo de trabajo del usuario sin romper compatibilidad.
- Un proceso de V&V que combina pruebas de software con validación física en un caso de estudio.
- Documentación técnica y de usuario orientada a reproducibilidad y mantenibilidad.

⁴ Greg Wilson et al. "Good enough practices in scientific computing". En: *PLoS Computational Biology* 13.6 (2017), e1005510. DOI: 10.1371/journal.pcbi.1005510.

Finalmente, la organización del documento es la siguiente. En el Capítulo 2 se presenta el marco de referencia y estado del arte desde la óptica de ingeniería de software y cómputo científico. En el Capítulo 3 se describe el análisis del sistema base: la arquitectura de Phonopy y su flujo de trabajo. El Capítulo 4 detalla el diseño e implementación del módulo PAM, incluyendo decisiones de arquitectura, algoritmo principal, vectorización e integración con la CLI. El Capítulo 5 formaliza la verificación y validación (V&V) del software. El Capítulo 6 muestra los resultados como una demostración de capacidades sobre TaSi₂. El Capítulo 7 resume conclusiones y plantea líneas de trabajo futuro orientadas a rendimiento (MPI/GPU), gestión de memoria y extensiones funcionales.

1. MARCO DE REFERENCIA Y ESTADO DEL ARTE

Este capítulo establece las bases teóricas y tecnológicas del proyecto desde la perspectiva de ingeniería de software aplicada al cómputo científico. Se divide en dos partes: (i) el Marco Teórico, que recoge los principios de ingeniería, el contexto tecnológico (Python/NumPy) y los fundamentos físicos necesarios para la implementación; y (ii) el Estado del Arte, que analiza el ecosistema de herramientas existente y la brecha específica que aborda este trabajo.

1.1. MARCO TEÓRICO

El marco teórico se organiza en tres pilares: fundamentos de ingeniería de software para cómputo científico; el ecosistema tecnológico sobre el cual se construye la solución (Python/NumPy); y los conceptos de física computacional que definen los datos y algoritmos a implementar.

1.1.1. Fundamentos de Ingeniería de Software para Cómputo Científico El desarrollo de software en el ámbito científico presenta un desafío dual: debe ser riguroso en su implementación de modelos físicos complejos y, al mismo tiempo, robusto, mantenible y extensible como cualquier otro sistema de software de alta calidad. Abordar la extensión de una biblioteca consolidada como `Phonopy` sin un marco de ingeniería de software sólido implicaría riesgos significativos, como la introducción de errores sutiles, la degradación de la mantenibilidad del código y la generación de resultados científicamente irreproducibles⁵. Por ello, este proyecto se fundamenta en los siguientes principios de ingeniería.

Arquitectura y Diseño de Software Modular. La base de un sistema de software sostenible es su arquitectura⁶. Para un proyecto que busca extender una herramienta existente,

⁵ Victoria Stodden et al. "Enhancing reproducibility for computational methods". En: *Science* 354.6317 (2016), págs. 1240-1241. DOI: 10.1126/science.aah6168.

⁶ Len Bass, Paul Clements y Rick Kazman. *Software Architecture in Practice*. 3.^a ed. Addison-Wesley

la **modularidad** es el principio arquitectónico rector. Un diseño modular divide el sistema en componentes con responsabilidades bien definidas, buscando **alta cohesión** (cada módulo se enfoca en una sola tarea) y **bajo acoplamiento** (los módulos son lo más independientes posible entre sí). En el contexto de este proyecto, en lugar de insertar la lógica para el cálculo del Momento Angular Fonónico (PAM) directamente en el código central de `Phonopy`, el diseño propone la creación de un módulo autocontenido. Este enfoque es fundamental para aislar la nueva funcionalidad, lo que simplifica su desarrollo, permite realizar pruebas de forma independiente y minimiza el impacto en la estabilidad del ecosistema de `Phonopy`.

Principios de Diseño para la Extensibilidad. Una vez decidido un enfoque modular, la siguiente pregunta es cómo integrar este nuevo módulo de forma limpia y sostenible. Los principios de diseño **SOLID**⁷ proporcionan un marco para ello. El **Principio Abierto/Cerrado** es particularmente relevante: establece que el software debe estar "abierto a la extensión, pero cerrado a la modificación". La arquitectura de `Phonopy` ya exhibe esta cualidad a través de su jerarquía de clases para el post-procesamiento. Por lo tanto, el diseño aprovecha esta característica: para la funcionalidad de la Densidad de Estados resuelta en PAM (PAM-DOS), se diseña una nueva clase que extiende la funcionalidad de la clase base `Dos` de `Phonopy` mediante herencia. Esta estrategia permite reutilizar toda la infraestructura existente y validada para el cálculo de DOS, añadiendo únicamente la nueva lógica de clasificación y ponderación por PAM, sin alterar el código original.

Diseño de APIs y Sistemas Modulares. Los módulos interactúan a través de sus Interfaces de Programación de Aplicaciones (APIs). Un diseño de API bien definido es crucial para que el sistema sea coherente y fácil de usar⁴. Para el núcleo de cálculo del PAM, se diseña una API funcional, donde una función principal recibe estructuras de datos están-

Professional, 2012.

⁷ Robert C. Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall, 2018.

dar del ecosistema científico (arreglos de NumPy) y devuelve sus resultados en el mismo formato. Este enfoque promueve el bajo acoplamiento, ya que el módulo de cálculo no necesita conocer los detalles internos de Phonopy, sino solo el contrato definido por su API.

Calidad y Pruebas de Software (Verificación y Validación). Para garantizar la confianza en los resultados científicos, el proyecto incluye un proceso riguroso de Verificación y Validación (V&V).

- **Verificación ("¿Se construyó el producto correctamente?"):** Se refiere a la confirmación de que el software se adhiere a sus especificaciones. Se contemplan pruebas de componentes para asegurar que la implementación numérica de las ecuaciones del PAM³ es correcta.
- **Validación ("¿Se construyó el producto correcto?"):** Asegura que el software resuelve el problema real. La validación implica someter el software a casos de prueba con resultados físicos conocidos. Por ejemplo, se verifica que el PAM calculado para cristales con simetría de inversión temporal sea nulo, como dicta la teoría.

Este doble enfoque de V&V es indispensable para establecer la credibilidad de la herramienta.

Metodologías de Desarrollo y Colaboración en Código Abierto. Este proyecto aspira a generar una contribución valiosa para una comunidad de código abierto. Por tanto, su proceso de desarrollo adopta las herramientas y prácticas estándar de este entorno, como el sistema de control de versiones **Git** y flujos de trabajo basados en *pull requests* en plataformas como GitHub⁸. Esto asegura una gestión de código organizada y alinea el proyecto con las convenciones de la comunidad de Phonopy, facilitando una eventual integración oficial.

⁸ Yasset Perez-Riverol et al. "Ten simple rules for taking advantage of Git and GitHub". En: *PLoS Computational Biology* 12.7 (2016), e1004947. DOI: 10.1371/journal.pcbi.1004947.

1.1.2. Contexto Tecnológico: Ecosistema de Cómputo Científico en Python

Stack científico de Python. La solución se apoya en NumPy como base para el cálculo numérico, adoptando vectorización para delegar operaciones intensivas a rutinas optimizadas de bajo nivel, lo que mejora rendimiento y escalabilidad ^{9,10}. Matplotlib apoya la visualización de resultados.

Arquitectura de Phonopy como sistema de post-procesamiento. Phonopy orquesta y analiza resultados de calculadores de DFT externos. Su diseño desacoplado lo hace agnóstico al calculador y naturalmente modular, lo que habilita una integración no intrusiva del módulo PAM en su etapa de post-procesamiento ¹¹.

Flujo de datos en Phonopy. La Figura 1 resume el pipeline típico: configuración; generación de superceldas desplazadas; cálculo externo de fuerzas; ajuste de constantes de fuerza; diagonalización de la matriz dinámica; y post-procesamiento (bandas, DOS, etc.). El módulo PAM se integra en este último paso, consumiendo eigenvectores complejos y frecuencias.

1.1.3. Fundamentos Físicos para la Implementación.

DFT como fuente de datos. El pipeline comienza con datos de DFT: desplazamientos y fuerzas interatómicas (por ejemplo, vía método de superceldas). Estos datos se transforman en constantes de fuerza, base para el modelo armónico ^{14,15}.

Fonones como problema de autovalores. A partir de la matriz dinámica $D(\mathbf{q})$, la re-

⁹ Travis E. Oliphant. “Python for Scientific Computing”. En: *Computing in Science & Engineering* 9.3 (2007), págs. 10-20. DOI: 10.1109/MCSE.2007.58.

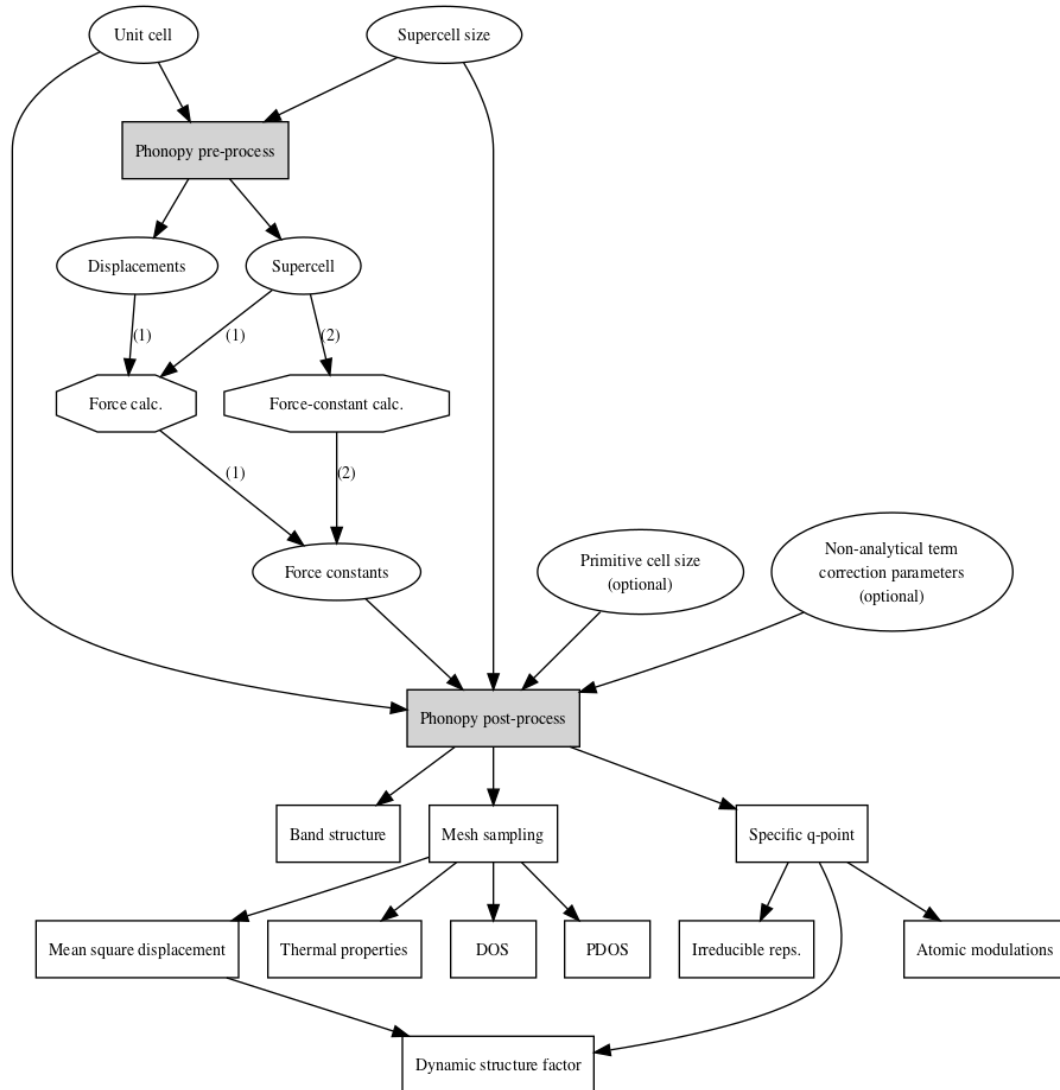
¹⁰ Charles R. Harris et al. “Array programming with NumPy”. En: *Nature* 585 (2020), págs. 357-362. DOI: 10.1038/s41586-020-2649-2.

¹¹ Atsushi Togo. “First-principles Phonon Calculations with Phonopy and Phono3py”. En: *J. Phys. Soc. Jpn.* 92.1 (2023), pág. 012001. DOI: 10.7566/JPSJ.92.012001.

¹⁴ Pierre Hohenberg y Walter Kohn. “Inhomogeneous electron gas”. En: *Physical review* 136.3B (1964), B864.

¹⁵ Walter Kohn y Lu Jeu Sham. “Self-consistent equations including exchange and correlation effects”. En: *Physical review* 140.4A (1965), A1133.

Figura 1. Flujo de datos y procesos en un cálculo típico con Phonopy¹².

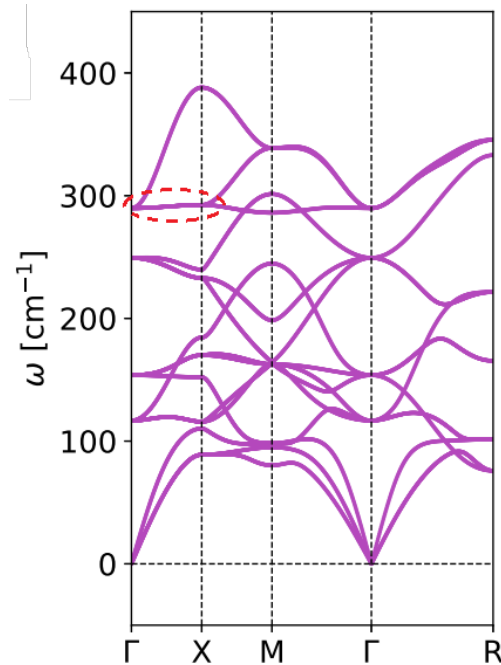


Fuente: "First-principles Phonon Calculations with Phonopy and Phono3py"¹³.

solución del problema de autovalores entrega frecuencias y autovectores complejos que definen los modos fonónicos. La relación $\omega(\mathbf{q})$ se visualiza comúnmente como dispersión de fonones (Figura 2)¹⁶.

¹⁶ Max Born y Kun Huang. *Dynamical Theory of Crystal Lattices*. Oxford University Press, 1954.

Figura 2. Ejemplo de dispersión de fonones: frecuencias por trayectoria en el espacio recíproco.



Fuente: "Interplay between the B-sites' valence states and phonon dynamics in the Ca_3BO antiperovskite" ¹⁷.

Formalismo del PAM y consecuencias algorítmicas. El valor esperado del PAM por modo se obtiene a partir de los autovectores de polarización complejos, con operaciones sobre números complejos y agregación por átomo; su estructura es naturalmente vectorizable, lo que motiva una implementación eficiente con NumPy ³. La derivación matemática detallada de este formalismo se presenta en el **Anexo B**.

El punto de partida es el momento angular de un modo fonónico individual (\mathbf{q}, ν) , donde \mathbf{q} es el vector de onda y ν es el índice de la banda. A partir de la derivación que conduce a la Ecuación 1, el valor esperado del PAM para un modo específico, $l_{\mathbf{q},\nu}$, se define en términos de su vector propio de polarización $\epsilon_{\mathbf{q},\nu}$. El vector $\epsilon_{\mathbf{q},\nu}$ es una entidad de $3N_p$ -dimensiones, donde N_p es el número de átomos en la celda primitiva, y puede ser descompuesto en los desplazamientos tridimensionales de cada átomo j como $\epsilon_{j,\mathbf{q},\nu}$.

La componente cartesiana α del PAM para este modo es:

$$l_{\mathbf{q},\nu}^{\alpha} = \hbar \sum_{j=1}^{N_p} \text{Im} \left(\epsilon_{j,\mathbf{q}\nu}^* \times \epsilon_{j,\mathbf{q}\nu} \right)_{\alpha} \quad (1)$$

donde la suma se realiza sobre todos los átomos j en la celda primitiva. Expandiendo el producto vectorial, obtenemos las tres componentes cartesianas:

$$l_{\mathbf{q},\nu}^x = \hbar \sum_j \left(\epsilon_{j,y,\mathbf{q}\nu}^* \epsilon_{j,z,\mathbf{q}\nu} - \epsilon_{j,z,\mathbf{q}\nu}^* \epsilon_{j,y,\mathbf{q}\nu} \right) = 2\hbar \sum_j \text{Im} \left(\epsilon_{j,y,\mathbf{q}\nu}^* \epsilon_{j,z,\mathbf{q}\nu} \right) \quad (2)$$

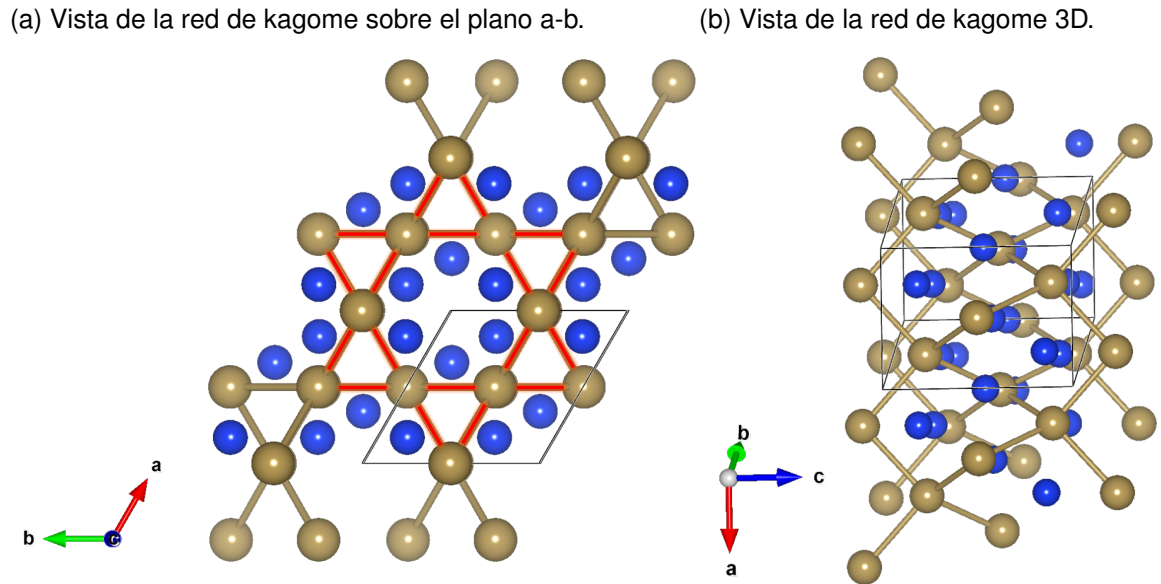
$$l_{\mathbf{q},\nu}^y = \hbar \sum_j \left(\epsilon_{j,z,\mathbf{q}\nu}^* \epsilon_{j,x,\mathbf{q}\nu} - \epsilon_{j,x,\mathbf{q}\nu}^* \epsilon_{j,z,\mathbf{q}\nu} \right) = 2\hbar \sum_j \text{Im} \left(\epsilon_{j,z,\mathbf{q}\nu}^* \epsilon_{j,x,\mathbf{q}\nu} \right) \quad (3)$$

$$l_{\mathbf{q},\nu}^z = \hbar \sum_j \left(\epsilon_{j,x,\mathbf{q}\nu}^* \epsilon_{j,y,\mathbf{q}\nu} - \epsilon_{j,y,\mathbf{q}\nu}^* \epsilon_{j,x,\mathbf{q}\nu} \right) = 2\hbar \sum_j \text{Im} \left(\epsilon_{j,x,\mathbf{q}\nu}^* \epsilon_{j,y,\mathbf{q}\nu} \right) \quad (4)$$

Estas tres ecuaciones son la base directa para la implementación computacional.

Caso de estudio para validación: TaSi₂. El disilicio de tantalio en el grupo espacial quiral P6₂22 (SG180) carece de inversión espacial y exhibe una red Kagome 3D de átomos de Ta, lo que lo convierte en un caso idóneo para validar la presencia de PAM no nulo (Figura 3) ².

Figura 3. **Estructura cristalina del compuesto** TaSi₂.



Fuente: Elaboración propia.

1.2. ESTADO DEL ARTE

El ecosistema de análisis fonónico cuenta con herramientas de alta madurez. Phonopy es el estándar de facto para dispersión de fonones, DOS y propiedades térmicas a nivel armónico, con arquitectura modular e interoperabilidad con múltiples códigos DFT ¹⁸. Paquetes complementarios amplían capacidades: phono3py aborda fenómenos anarmónicos ¹⁸, mientras que ALM se centra en la parametrización de constantes de fuerza de alto orden ¹⁹ y ShengBTE resuelve la ecuación de transporte de Boltzmann para fonones

¹⁸ Atsushi Togo. "First-principles phonon calculations with phonopy and phono3py". En: *Journal of the Physical Society of Japan* 92.1 (2023), pág. 012001.

¹⁹ Terumasa Tadano y Shinji Tsuneyuki. "ALM: A Python library for crystal lattice dynamics". En: *Journal of the Physical Society of Japan* 87.4 (2018), pág. 041015. DOI: 10.7566/JPSJ.87.041015.

²⁰. Estas herramientas ejemplifican buenas prácticas de ingeniería en software científico: control de versiones, validación y documentación abiertas.

No obstante, la propiedad de Momento Angular Fonónico (PAM) ha ganado relevancia solo recientemente, impulsada por predicciones teóricas y observaciones experimentales de fonones quirales ²¹. A la fecha, las bibliotecas principales (incluida `Phonopy`) no incluyen una funcionalidad nativa e integrada para calcular PAM/PAM-DOS. Si bien ofrecen los insumos (autovectores complejos), carecen de rutinas de post-procesamiento para cuantificar y analizar el PAM.

Para suplir esta carencia han surgido implementaciones aisladas, como scripts *ad-hoc* publicados en blogs personales ²². Aunque valiosas como pruebas de concepto, presentan limitaciones críticas desde ingeniería: (i) falta de integración con flujos establecidos; (ii) ausencia de verificación y validación sistemáticas; (iii) funcionalidad de análisis limitada (e.g., sin PAM-DOS ni proyección sobre bandas); y (iv) mantenibilidad frágil sin comunidad ni control de versiones robusto.

Se evidencia existe una brecha entre herramientas consolidadas para el análisis “clásico” de fonones y soluciones fragmentadas para el PAM. Este trabajo se posiciona para cerrar dicha brecha mediante una extensión modular, verificada y validada, integrada coherentemente en el flujo de trabajo de `Phonopy`.

²⁰ Wu Li et al. “ShengBTE: A solver of the Boltzmann transport equation for phonons”. En: *Computer Physics Communications* 185.6 (2014), págs. 1747-1758. DOI: 10.1016/j.cpc.2014.02.015.

²¹ H. Zhao et al. “Observation of chiral phonons”. En: *Nature Physics* 16.7 (2020), págs. 751-755. DOI: 10.1038/s41567-020-0903-7.

²² Qijing Zheng. *Phonon Angular Momentum*. Accedido el 2025-11-05. 2022. URL: <https://qijingzheng.github.io/posts/Angular-Momentum-in-Solids/>.

2. ANÁLISIS DEL SISTEMA: ARQUITECTURA DE PHONOPY

2.1. PROPÓSITO Y ALCANCE

PHONOPY es un software de post-procesamiento diseñado para calcular propiedades de fonones a partir de las fuerzas atómicas obtenidas de un calculador de primeros principios o de un potencial de campo de fuerzas. Su desarrollo, liderado por Atsushi Togo, comenzó como una evolución del código *fropho*, con el objetivo de proporcionar una herramienta más general, flexible y fácil de usar para la comunidad de la ciencia de materiales y la física del estado sólido.

Desde su concepción, PHONOPY se ha posicionado como un actor central en el ecosistema de software para la dinámica de redes. Su principal ventaja competitiva radica en su enfoque agnóstico respecto al calculador de fuerzas. Mientras que muchos paquetes de primeros principios como VASP, Quantum ESPRESSO o ABINIT incluyen sus propios módulos para el cálculo de fonones (principalmente mediante la Teoría de Perturbaciones de Densidad Funcional, DFPT), PHONOPY adopta el método de supercelda y desplazamientos finitos, lo que le permite interactuar con prácticamente cualquier código capaz de calcular las fuerzas sobre los átomos en una estructura periódica. Esta versatilidad se ve reflejada en la extensa lista de interfaces que mantiene, incluyendo VASP, Quantum ESPRESSO, ABINIT, CRYSTAL, SIESTA, entre otros.

Las ventajas de PHONOPY sobre otras implementaciones pueden resumirse en los siguientes puntos:

- **Modularidad y Extensibilidad:** Su base en Python y su API bien definida permiten la extensión de sus capacidades, como se demuestra en esta tesis con la implementación del PAM.
- **Generalidad:** Al desacoplar el cálculo de fuerzas de la construcción de la matriz dinámica, PHONOPY no está limitado a las aproximaciones de un único paquete de software (e.g., tipo de funcional, base de pseudopotenciales).

- **Soporte a la Comunidad y Validación:** Es un software ampliamente utilizado y validado, con una comunidad de usuarios activa y una documentación robusta. Sus resultados han sido contrastados en innumerables publicaciones científicas.
- **Manejo de Simetría:** Integra la librería *spglib* (**phonopy-phono3py-JPSJ**) para un manejo riguroso de la simetría cristalográfica, lo que es crucial para reducir el coste computacional y asegurar la correcta degeneración de los modos fonónicos.

Para el propósito de esta tesis, la característica más relevante es su extensibilidad. La implementación del PAM requirió un acceso profundo a las estructuras de datos internas que manejan los eigenvectores y eigenvalores fonónicos, algo que la arquitectura de PHONOPY facilita de manera natural.

2.2. FLUJO DE ALTO NIVEL

A volumen constante, un cálculo típico con *Phonopy* sigue un canal bien definido:

1. **Entrada del Usuario:** El proceso se inicia con una estructura de celda unitaria (e.g., 'POSCAR') y un archivo de configuración de PHONOPY (e.g., 'phonopy.conf' o parámetros de línea de comandos) que especifica la matriz de supercelda y otros parámetros de cálculo.
2. **Generación de Desplazamientos:** PHONOPY utiliza la simetría del cristal para generar un conjunto mínimo de desplazamientos atómicos en una supercelda. Estos desplazamientos se guardan en archivos de estructura (e.g., 'POSCAR-001', 'POSCAR-002', ...) y la información de los desplazamientos se almacena en el archivo 'phonopy_disp.yaml'.
3. **Cálculo de Fuerzas (Externo):** El usuario ejecuta un cálculo de fuerzas para cada una de las superceldas con desplazamientos utilizando un software externo (e.g., VASP). Este es el paso computacionalmente más costoso.

4. **Recolección de Fuerzas:** PHONOPY recolecta las fuerzas de los archivos de salida del calculador (e.g., 'vasprun.xml') y las asocia con sus desplazamientos correspondientes, generando un conjunto de datos de fuerzas-desplazamientos ('FORCE_SETS').
5. **Cálculo de Constantes de Fuerza:** A partir del conjunto de datos de fuerzas-desplazamientos, se calculan las constantes de fuerza de segundo orden ($\Phi_{\alpha\beta}(ij)$) mediante un ajuste por mínimos cuadrados (pseudo-inversa).
6. **Construcción de la Matriz Dinámica:** Con las constantes de fuerza, PHONOPY construye la matriz dinámica para cualquier vector de onda q del espacio recíproco.
7. **Post-procesamiento:** Finalmente, se diagonaliza la matriz dinámica para obtener las frecuencias y eigenvectores de los fonones, que se utilizan para calcular diversas propiedades como la estructura de bandas, la densidad de estados (DOS) o las propiedades térmicas.

Este flujo se refleja tanto en el CLI (comandos `phonopy` y `phonopy-load`) como en los métodos correspondientes de la API.

2.3. MODELO DE DATOS Y TRANSFORMACIONES

2.3.1. Estructuras de datos internas: Internamente, la información cristalográfica se maneja a través de la clase 'PhonopyAtoms', que es análoga a la clase 'Atoms' de la librería ASE. La orquestación del cálculo completo se gestiona mediante la clase 'Phonopy', que contiene como atributos instancias de otras clases especializadas como 'BandStructure' o 'TotalDos'.

Un aspecto fundamental para la implementación del PAM es la representación de los eigenvectores fonónicos. Dado que la matriz dinámica es Hermitiana, sus eigenvectores $e_{q,\nu}$ son, en general, complejos. Un eigenvector es un vector de dimensión $3N_p$, donde N_p es el número de átomos en la celda primitiva. Representa las amplitudes y fases relativas

de los desplazamientos atómicos para un modo fonónico específico. En los archivos de salida YAML, como 'band.yaml', cada componente del eigenvector se representa como un par de números de punto flotante: la parte real y la parte imaginaria. Por ejemplo, la componente x del desplazamiento del primer átomo se guardaría como '[real, imag]'. Esta representación explícita de las partes real e imaginaria es el punto de partida directo para el cálculo del PAM, como se describe en la Ecuación 2 y se implementa en el módulo 'pam.py' (sección 3.1). La preservación de esta información de fase es, por tanto, una capacidad esencial de PHONOPY que ha permitido el desarrollo de este trabajo.

2.3.2. Tres celdas: unitaria, supercelda y primitiva Phonopy mantiene internamente tres celdas relacionadas: la *celda unitaria* (entrada), una *supercelda* para cálculos de fuerzas por desplazamientos finitos y una *celda primitiva* que define la base de espacio recíproco para el cálculo fonónico.

Sea M_s la matriz entera de supercelda y M_p la matriz de ejes primitivos. Usando vectores columna para las bases reticulares, las transformaciones son

$$(\mathbf{a}_s \ \mathbf{b}_s \ \mathbf{c}_s) = (\mathbf{a}_u \ \mathbf{b}_u \ \mathbf{c}_u) M_s, \quad (5)$$

$$(\mathbf{a}_p \ \mathbf{b}_p \ \mathbf{c}_p) = (\mathbf{a}_s \ \mathbf{b}_s \ \mathbf{c}_s) M_s^{-1} M_p. \quad (6)$$

Tras construir la supercelda y la primitiva, la celda unitaria original normalmente no se utiliza en el *solve* fonónico.

2.3.3. Simetría y uso de spglib La detección de simetría y reducciones se apoyan en spglib (operaciones de grupo espacial)²³. La simetría informa la elección de celda primitiva (PRIMITIVE_AXES), la construcción de la supercelda (DIM), el muestreo de

²³ Atsushi Togo e Isao Tanaka. "Spglib: a software library for crystal symmetry search". En: *arXiv preprint arXiv:1808.01590* (2018). <https://arxiv.org/abs/1808.01590>.

puntos- \mathbf{q} irreducibles en mallas, la simetrización opcional de constantes de fuerza y las correcciones no analíticas cerca de Γ . La simetría puede desactivarse (por ejemplo, para depuración) vía CLI/archivo de configuración.

2.3.4. Problema de autovalores fonónicos En cada punto \mathbf{q} , los modos fonónicos $\{\omega_{\mathbf{q}\nu}, \mathbf{e}_{\mathbf{q}\nu}\}$ resuelven el problema de autovalores de la matriz dinámica

$$\mathbf{D}(\mathbf{q}) \mathbf{e}_{\mathbf{q}\nu} = \omega_{\mathbf{q}\nu}^2 \mathbf{e}_{\mathbf{q}\nu}, \quad (7)$$

construida a partir de constantes de fuerza reales de segundo orden Φ y masas atómicas m :

$$D_{\kappa\alpha,\kappa'\beta}(\mathbf{q}) = \frac{1}{\sqrt{m_\kappa m_{\kappa'}}} \sum_{\mathbf{R}} \Phi_{0\kappa\alpha, \mathbf{R}\kappa'\beta} e^{i\mathbf{q}\cdot(\mathbf{r}_{\mathbf{R}\kappa'} - \mathbf{r}_{0\kappa})}. \quad (8)$$

Aquí κ, κ' enumeran átomos en la *celda primitiva*, α, β son componentes cartesianas y \mathbf{R} recorre celdas reticulares.

2.4. CÁLCULOS DE BANDAS, MALLA Y DOS

2.4.1. Estructura de bandas Para una trayectoria a trozos en el espacio recíproco, Phonopy evalúa $\mathbf{D}(\mathbf{q})$ en cada punto y obtiene $\omega_{\mathbf{q}\nu}$ y $\mathbf{e}_{\mathbf{q}\nu}$. Las curvas pueden graficarse y exportarse (`band.yaml`), y admiten proyecciones parciales cuando hay autovectores.

2.4.2. Malla (*mesh*) y DOS Sobre mallas \mathbf{q} regulares, Phonopy integra DOS total y proyectada (PDOS) mediante el método de tetraedros o gaussianas. Los pesos de simetría reducen el costo si se usa la malla irreducible.

2.5. RENDIMIENTO E IMPLEMENTACIÓN

Los núcleos computacionalmente intensivos (p. ej., matrices dinámicas, eigenproblemas, integración por tetraedros) están acelerados por extensiones C/C++ y pueden paralelizar-

se con OpenMP cuando se compila con soporte. El CLI informa soporte OpenMP al inicio cuando está disponible. Las salidas HDF5 reducen E/S y mejoran robustez en conjuntos de datos grandes.

2.6. EXTENSIBILIDAD Y PUNTOS DE INTEGRACIÓN

La arquitectura de Phonopy expone ejes de integración claros:

- **Front-ends:** CLI (`phonopy`, `phonopy-load`) que parsean opciones en parámetros y orquestan el flujo.
- **Motor:** El objeto `Phonopy` encapsula celdas, conjuntos de desplazamientos/fuerzas, constantes de fuerza, NAC y los solucionadores fonónicos (`band/malla/qpoints`).
- **E/S:** YAML/HDF5 estructurado para entradas y resultados; interfaces de calculadores que desacoplan formatos específicos.
- **Análisis:** DOS/PDOS, termodinámica, desplazamientos térmicos, animaciones, modulaciones, representaciones irreducibles.

Nuevos análisis suelen consumir los datos en memoria existentes (frecuencias, autovectores, pesos) y pueden añadirse junto a los flujos existentes de graficación y exportación ¹.

2.7. PATRONES DE USO PRÁCTICO

2.7.1. Flujo orientado al CLI Este es un ejemplo de cómo sería un flujo usando los comandos CLI de `phonopy`.

```
# 1) Construir desplazamientos y snapshot  
phonopy -d --dim 2 2 2 --pa auto -c POSCAR
```

```
# 2) Calcular fuerzas externamente para cada supercelda desplazada
```

```

# (paso dependiente del calculador)

# 3) Ensamblar FORCE_SETS y persistir parametros
phonopy --sp -f vasprun.xml-00{1..N}

# 4) Ejecutar bandas o malla desde el YAML guardado
phonopy-load --band auto -p phonopy_params.yaml
phonopy-load --mesh 20 20 20 --dos -p phonopy_params.yaml

```

2.7.2. Flujo orientado a la API. Este es un ejemplo de cómo sería el flujo usando directamente la API para crear scripts personalizados.

```

from phonopy import Phonopy
from phonopy.interface.calculator import read_crystal_structure

cell, _ = read_crystal_structure("POSCAR", interface_mode='vasp')
phonon = Phonopy(cell, supercell_matrix=[[2,0,0],[0,2,0],[0,0,2]],
                 primitive_matrix='auto')

phonon.generate_displacements(distance=0.03)
supercells = phonon.supercells_with_displacements

# ... ejecutar fuerzas externamente ... luego
phonon.forces = sets_of_forces
phonon.produce_force_constants()

# Bandas
phonon.run_band_structure(bands, with_eigenvectors=True)
phonon.plot_band_structure().show()

# Malla y DOS

```

```
phonon.init_mesh([20,20,20], with_eigenvectors=True)
phonon.mesh.run()
phonon.run_total_dos(); phonon.plot_total_dos().show()
```

3. DISEÑO E IMPLEMENTACIÓN DEL MÓDULO PAM

Este capítulo documenta el diseño arquitectónico, la implementación y la integración en CLI del módulo para el *Momento Angular Fonónico* (PAM) y su densidad de estados asociada (PAM-DOS) dentro del ecosistema de *Phonopy*. El foco es de ingeniería de software: decisiones de arquitectura, modularidad, API, eficiencia (vectorización) y verificación/validación de la funcionalidad desarrollada. La arquitectura de base del sistema anfitrión se resume en el Capítulo 2, y la integración aquí descrita fue concebida para ser no intrusiva, coherente y mantenible ¹.

La extensión de un software científico consolidado como PHONOPY exige un diseño cuidadoso que respete su arquitectura existente, garantice la robustez de los nuevos cálculos y mantenga la coherencia con sus patrones de uso. El objetivo principal del diseño fue integrar el cálculo del Momento Angular Fonónico (PAM) y la Densidad de Estados asociada (PAM-DOS) de una manera que se percibiera como una capacidad nativa del programa, en lugar de un añadido inconexo. Para ello, se adoptó una estrategia de mínima intrusión, favoreciendo la creación de nuevos módulos y la herencia de clases existentes sobre la modificación extensiva del código base. Este enfoque no solo preserva la estabilidad del software original, sino que también facilita el mantenimiento y la futura expansión de las funcionalidades aquí implementadas. En esta sección, se detallan los requisitos que guiaron el desarrollo, las decisiones de diseño fundamentales que se tomaron y la estructura modular resultante.

3.0.1. Análisis de Requisitos: Antes de escribir una sola línea de código, se llevó a cabo un análisis exhaustivo de los requisitos funcionales y no funcionales que la nueva extensión debía cumplir. Estos requisitos se derivaron tanto de las necesidades teóricas del proyecto de investigación como de las mejores prácticas en el desarrollo de software científico.

Requisitos Funcionales: Los requisitos funcionales definen las capacidades específicas que la extensión debe proporcionar al usuario y al sistema:

- **Cálculo del Momento Angular Fonónico:** El sistema debe ser capaz de calcular el vector del PAM, $I(\mathbf{q}, \nu)$, para cada modo fonónico (\mathbf{q}, ν) a partir de los eigenvectores complejos y las frecuencias obtenidas de PHONOPY. Esto incluye la capacidad de incorporar efectos de la temperatura a través del factor de ocupación de Bose-Einstein.
- **Cálculo de la Densidad de Estados Resuelta en PAM (PAM-DOS):** Se debe implementar una nueva forma de Densidad de Estados que separe las contribuciones de los fonones con PAM positivo ($l_z > 0$) de aquellos con PAM negativo ($l_z < 0$) para cada eje cartesiano. Esta PAM-DOS debe poder ser calculada utilizando los métodos ya existentes en PHONOPY (ensanchamiento Gaussiano y método de los tetraedros) para asegurar la consistencia.
- **Integración con la Línea de Comandos (CLI):** Las nuevas funcionalidades deben ser accesibles a través de la interfaz de línea de comandos de PHONOPY (específicamente, 'phonopy-load'), con nuevos parámetros que permitan al usuario activar el cálculo del PAM-DOS, especificar la temperatura y controlar otros parámetros relevantes.
- **Visualización de Resultados:** Se debe proporcionar una herramienta para visualizar los resultados. Esto incluye:
 1. Un gráfico de la PAM-DOS que muestre las contribuciones positivas y negativas de forma clara, por ejemplo, en lados opuestos de un eje.
 2. Una visualización de la estructura de bandas de fonones donde el color de las bandas represente la magnitud y el signo del PAM (proyección del PAM).

- **Generación de Archivos de Salida:** Los resultados numéricos, tanto del PAM-DOS como de la proyección del PAM en las bandas, deben poder guardarse en archivos de texto para su posterior análisis y graficación con herramientas externas.

Requisitos No Funcionales: Estos requisitos definen las cualidades del sistema y las restricciones bajo las cuales debe operar:

- **Integración Arquitectónica:** La nueva funcionalidad debe integrarse de forma natural en la arquitectura de PHONOPY. Esto implica que la clase principal 'Phonopy' debe ser el punto de entrada para acceder a los nuevos cálculos, y las nuevas clases deben heredar de las clases base de PHONOPY siempre que sea posible para maximizar la reutilización de código.
- **Reutilización de Código:** El diseño debe evitar la duplicación de funcionalidades ya existentes. Por ejemplo, los métodos para el cálculo de DOS (tetraedros, smearing) y la infraestructura para el manejo de mallas de puntos q deben ser reutilizados, no reinventados.
- **Mantenibilidad y Modularidad:** El código de la extensión debe estar encapsulado en módulos bien definidos, separando la lógica del cálculo del PAM, el cálculo del PAM-DOS y la visualización. Esto facilita la depuración, el mantenimiento y futuras mejoras.
- **Compatibilidad hacia Atrás:** La adición de las nuevas funcionalidades no debe alterar el comportamiento existente de PHONOPY. Los usuarios que no utilicen los nuevos parámetros no deben experimentar ningún cambio en los resultados o en el flujo de trabajo.
- **Rendimiento:** Si bien la precisión es la prioridad, el diseño no debe impedir optimizaciones de rendimiento. El uso de operaciones vectorizadas de NumPy debe

ser priorizado sobre bucles explícitos en Python para los cálculos numéricamente intensivos.

3.0.2. Decisiones de Diseño: Basados en el análisis de requisitos, se tomaron varias decisiones estratégicas de diseño para guiar la implementación. Estas decisiones buscaron un equilibrio entre la eficiencia del desarrollo, la robustez del código y la elegancia de la integración.

Una de las decisiones más importantes fue cómo estructurar la nueva funcionalidad de PAM-DOS dentro del código existente. Se evaluaron dos patrones de diseño de software principales: la herencia y la composición.

- **Herencia para PAM-DOS:** PHONOPY ya posee una jerarquía de clases para el cálculo de la densidad de estados, con una clase base 'Dos' y una clase derivada 'TotalDos' (ver archivo 'phonopy/phonon/dos.py'). La funcionalidad requerida para la PAM-DOS es conceptualmente una especialización de la DOS total: en lugar de sumar todos los estados, los clasifica y los suma por separado según el signo del PAM. Por lo tanto, se optó por implementar 'PAMDos' como una nueva clase que hereda de la clase base 'Dos'. Esta decisión de diseño (ver figura 4) ofrece ventajas significativas:

1. **Reutilización de Código:** Se heredan directamente los métodos para configurar el rango de frecuencias, inicializar la malla de cálculo y aplicar los métodos de ensanchamiento o de tetraedros.
2. **Polimorfismo:** La nueva clase 'PAMDos' puede ser tratada como cualquier otro objeto de tipo 'Dos' en el resto del código de PHONOPY, asegurando una integración fluida.
3. **Consistencia:** La interfaz de la nueva clase es consistente con las clases de DOS existentes, lo que facilita su uso para desarrolladores familiarizados con el código de PHONOPY.

La implementación específica consistió en sobrecargar el método 'run' de la clase base para incluir el cálculo del PAM antes de proceder con el cálculo de la DOS, y adaptar los métodos de cálculo para manejar las contribuciones positivas y negativas por separado.

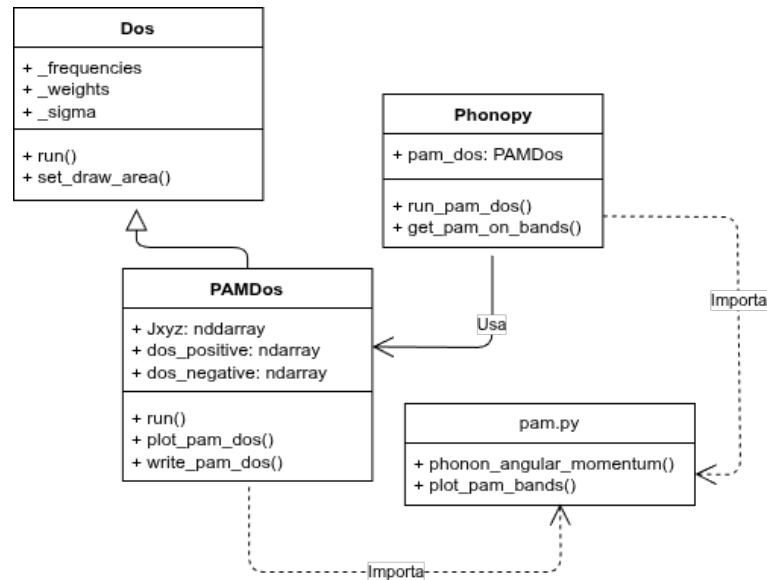
- **Módulo Independiente para el Cálculo del PAM:** A diferencia de la PAM-DOS, el cálculo del PAM en sí mismo no es una especialización de una funcionalidad existente en PHONOPY. Es un cálculo nuevo que toma frecuencias y eigenvectores como entrada. Por lo tanto, en lugar de integrarlo forzosamente en una clase existente, se decidió encapsular esta lógica en un nuevo módulo independiente: 'phonopy/phonon/pam.py'. Este módulo contiene funciones puras que realizan el cálculo matemático, siguiendo un patrón más funcional que orientado a objetos. Esta decisión promueve una alta cohesión (el módulo hace una sola cosa: calcular el PAM) y un bajo acoplamiento (no depende fuertemente de la estructura interna de otras clases de PHONOPY, solo de sus datos de entrada).

Para la integración a nivel de la API, se modificó la clase principal 'Phonopy' ('phonopy/api_phonopy.py') para que actúe como una "fábrica" de los nuevos objetos de cálculo. Se añadieron:

- Un método 'run_pam_dos()' para orquestrar la ejecución del cálculo de la PAM-DOS.
- Una propiedad '@property def pam_dos(self)' que devuelve una instancia de la clase 'PAMDos', inicializada con los datos necesarios de la malla de cálculo.

Este enfoque mantiene la clase 'Phonopy' como el punto de entrada central y coherente para todas las funcionalidades, tal como esperan los usuarios de la API.

Figura 4. *Diagrama de clases UML simplificado.*



Fuente: Elaboración propia.

3.0.3. Estructura de Módulos: La implementación se organizó en varios módulos, cada uno con una responsabilidad clara, para asegurar la modularidad y la mantenibilidad del código.

‘**pam.py**’ (Nuevo módulo) Contiene la lógica fundamental para el cálculo del vector del Momento Angular Fonónico. La función principal, ‘`phonon_angular_momentum`’, toma como entrada las frecuencias y los eigenvectores complejos y devuelve el PAM. Este módulo es autocontenido y no tiene dependencias con otras partes de PHONOPY, solo con NumPy.

‘**dos.py**’ (Módulo modificado) Este módulo existente de PHONOPY, que contiene las clases para el cálculo de la densidad de estados, fue extendido para incluir la nueva clase ‘**PAMDos**’. Esta clase hereda de ‘**Dos**’ y utiliza las funciones del módulo ‘**pam.py**’ para realizar sus cálculos. Contiene la lógica para separar los modos fonónicos por el signo de

su PAM y ponderar sus contribuciones a la DOS.

‘api_phonopy.py’ (Módulo modificado) Es el corazón de la API de PHONOPY. Fue modificado para exponer la nueva funcionalidad. Se añadieron métodos como `‘run_pam_dos()’` para el cálculo de PAM-DOS, `‘get_pam_on_bands()’` para el cálculo de PAM en bandas, y funciones de visualización como `‘plot_pam_bands()’` y `‘plot_band_structure_and_pam_dos()’`. Además, la propiedad `‘pam_dos’` da acceso a la instancia de la clase `‘PAMDos’` para cálculos y visualizaciones más detalladas. Este módulo actúa como el pegamento que conecta la nueva lógica de cálculo con el resto del sistema.

‘cui/settings.py’ (Módulo modificado) Este módulo gestiona la configuración de la línea de comandos. Se extendió para reconocer los nuevos parámetros como `‘-pamdos’`, `‘-pam-bands’`, `‘-pam-temperature’` y `‘-int-pamdos’`, y para configurar el objeto `‘Phonopy-Settings’` en consecuencia.

‘cui/phonopy_script.py’ (Módulo modificado) Este script es el punto de entrada principal para el comando `‘phonopy-load’`. Se modificó su lógica para invocar los nuevos métodos de la clase `‘Phonopy’` cuando los flags correspondientes son detectados en los argumentos de la línea de comandos.

Esta organización modular asegura que cada componente tenga un propósito bien definido, facilitando las pruebas, la depuración y futuras extensiones, al tiempo que se integra de manera coherente con la filosofía de diseño de PHONOPY.

3.1. IMPLEMENTACIÓN DEL MÓDULO PAM.PY

El módulo `pam.py` constituye el núcleo central de la presente contribución computacional. Su finalidad es la traducción del formalismo teórico del Momento Angular Fonónico (PAM) a un conjunto de rutinas numéricamente eficientes y estables, integradas dentro del ecosistema de PHONOPY. Este módulo encapsula la lógica fundamental para el cálculo del

PAM a partir de los eigenvectores y eigenvalores fonónicos, el manejo de su dependencia con la temperatura, y la estructuración de los datos para su posterior análisis y visualización. El diseño del módulo prioriza la claridad algorítmica, la eficiencia computacional mediante el uso de operaciones vectorizadas de NumPy, y la modularidad para facilitar futuras extensiones.

A continuación, se desglosa en detalle la arquitectura e implementación de cada componente funcional de este módulo.

3.1.1. Función `phonon_angular_momentum` La función `phonon_angular_momentum` es la implementación directa del cálculo del PAM para un conjunto de modos fonónicos. Esta función recibe como entrada las frecuencias de los fonones y sus correspondientes eigenvectores complejos, y retorna un tensor tridimensional que contiene las componentes cartesianas del PAM para cada modo.

Fundamento Teórico: El momento angular de un modo fonónico ν en el punto q del espacio recíproco, $I_{q\nu}$, se define como la suma de las contribuciones de momento angular de todos los átomos en la celda primitiva. Siguiendo el formalismo establecido por Zhang y Niu (2014), la componente cartesiana α del momento angular del modo se expresa como se vio en las ecuaciones 2, 3 y 4.

La implementación numérica de esta fórmula se diseñó para ser computacionalmente eficiente mediante la vectorización de operaciones sobre arreglos de NumPy, evitando bucles explícitos en Python.

El algoritmo implementado en la función sigue una serie de pasos lógicos diseñados para maximizar la eficiencia mediante la vectorización de operaciones, aprovechando las capacidades de la biblioteca NumPy. El diagrama de flujo de este proceso se ilustra en la Figura 5.

1. **Entrada de Datos:** La función acepta como argumentos principales un arreglo

NumPy de frecuencias fonónicas (`freq`) y un arreglo de eigenvectores de polarización complejos (`polar_vec`). Opcionalmente, se puede proporcionar la temperatura (`temp`) para incluir efectos térmicos.

2. **Manejo de la Forma del Eigenvector:** Se ha diseñado la función para aceptar dos formatos comunes para el arreglo de eigenvectores:

- Un arreglo de 4 dimensiones con forma $(N_q, N_b, N_a, 3)$, donde N_q es el número de puntos q , N_b es el número de bandas, N_a es el número de átomos, y 3 corresponde a las componentes cartesianas.
- Un arreglo de 3 dimensiones con forma $(N_q, N_b, 3 \times N_a)$, donde las componentes atómicas y cartesianas están aplanadas en el último eje.

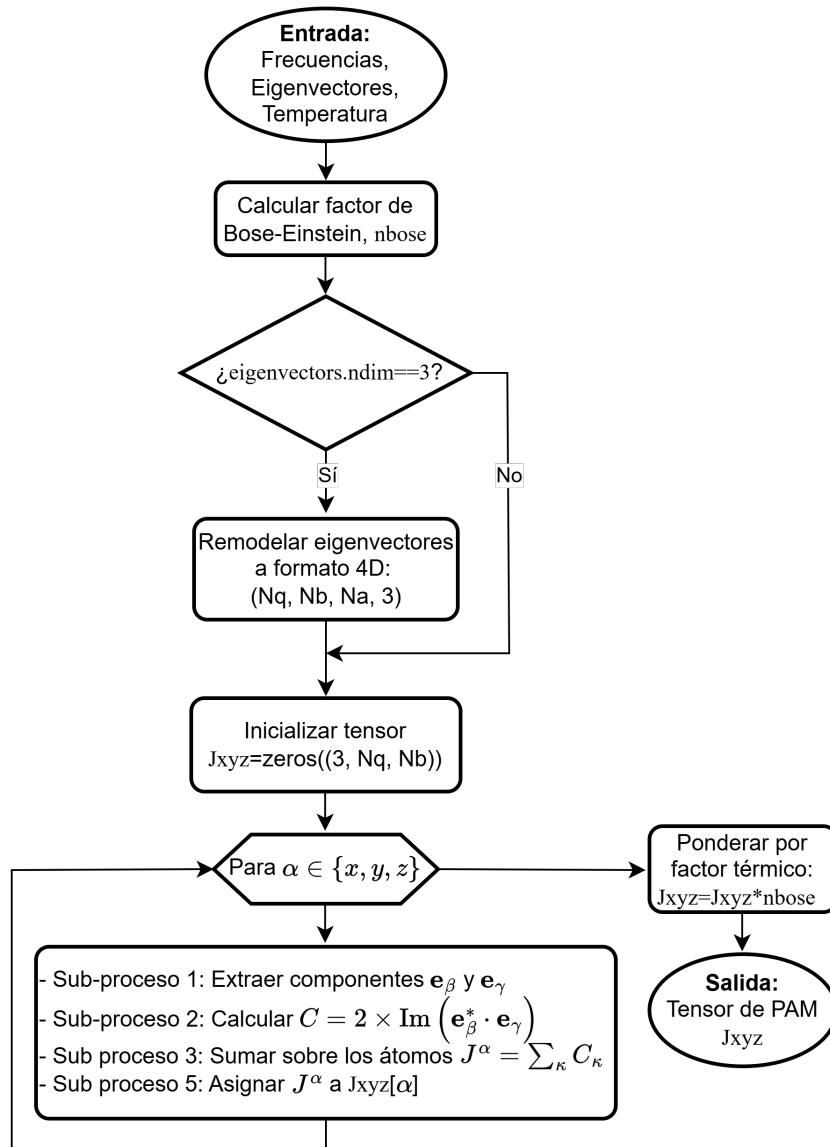
Un bloque condicional verifica la dimensionalidad del arreglo de entrada y, si es necesario, lo remodela a la forma de 4 dimensiones utilizando `numpy.reshape`. Esta flexibilidad mejora la interoperabilidad con diferentes módulos de PHONOPY.

3. **Cálculo Componente por Componente:** En lugar de calcular explícitamente el producto vectorial de la Ecuación 30, se utiliza la formulación de la Ecuación 2. Se define una lista de índices `ixyz = [,,]` que representa las permutaciones cíclicas para las componentes l_x, l_y, l_z . Se itera sobre estas componentes:

- Para cada componente α , se extraen los vectores de polarización correspondientes a las componentes β y γ (`e1` y `e2`) mediante rebanado (slicing) de arreglos de NumPy.
- Se calcula el producto $\epsilon_\beta(\mathbf{q}\nu) \cdot \epsilon_\gamma^*(\mathbf{q}\nu)$ de forma vectorizada para todos los puntos q , bandas y átomos simultáneamente.
- Se extrae la parte imaginaria del resultado con `numpy.imag`.
- Se realiza la suma sobre los índices de los átomos (`axis=2`) para obtener la contribución total de cada modo.

- El resultado se multiplica por 2, de acuerdo con la Ecuación 2.
4. **Aplicación del Factor Térmico:** El resultado intrínseco del PAM se multiplica por el factor de ocupación de Bose-Einstein (modificado para incluir la energía del punto cero), cuyo cálculo se detalla en la sección 3.1.3. Esta operación también se realiza de forma vectorizada.
 5. **Salida:** La función retorna un arreglo NumPy de forma $(3, N_g, N_b)$, donde la primera dimensión corresponde a las componentes cartesianas (l_x, l_y, l_z) del PAM.

Figura 5. **Diagrama de flujo que ilustra el proceso algorítmico de la función *phonon_angular_momentum*.**



Fuente: Elaboración propia.

El pseudocódigo del algoritmo se presenta a continuación para mayor claridad.

Listing 3.1. Cálculo del Momento Angular Fonónico (PAM)

```
function phonon_angular_momentum(frequencies, eigenvectors, temperature):  
    # 1. Calcular factor de Bose-Einstein (incluyendo punto cero)  
    nbose = calculate_bose_einstein_factor(frequencies, temperature)  
  
    # 2. Asegurar formato (N_q, N_b, N_a, 3) para eigenvectores  
    if eigenvectors.ndim == 3:  
        num_atoms = eigenvectors.shape / 3  
        E = reshape(eigenvectors, (N_q, N_b, num_atoms, 3))  
    else:  
        E = eigenvectors  
  
    # 3. Inicializar tensor de PAM  
    Jxyz = zeros((3, N_q, N_b))  
  
    # 4. Definir permutaciones ciclicas para componentes (y,z), (z,x), (x,y)  
    component_indices = [.,.]  
  
    # 5. Iterar sobre las componentes cartesianas del PAM  
    for alpha in: # Corresponde a x, y, z  
        # Extraer componentes de polarizacion de forma vectorizada  
        e_beta = E[:, :, :, component_indices[alpha]]  
        e_gamma = E[:, :, :, component_indices[alpha]]  
  
        # Calcular  $Im(e\_beta * conj(e\_gamma))$  y sumar sobre los atomos  
        pam_component = 2 * sum(imag(e_beta * conj(e_gamma)),  
→ axis=atom_axis)  
        Jxyz[alpha, :, :] = pam_component  
  
    # 6. Aplicar factor termico y retornar
```

```
return Jxyz * nbose
```

Este enfoque algorítmico garantiza que, a pesar de la aparente complejidad de la fórmula, el cálculo se realice de manera extraordinariamente rápida y eficiente en memoria, escalando linealmente con el número de modos fonónicos a calcular.

3.1.2. Consideraciones Numéricas: La implementación de un formalismo físico en un entorno computacional requiere una cuidadosa consideración de aspectos numéricos para garantizar la precisión, estabilidad y eficiencia del cálculo.

Precisión Numérica y Estabilidad: Para asegurar la máxima fidelidad con los resultados teóricos, todas las operaciones de punto flotante se realizan utilizando el tipo de dato `numpy.float64` para las cantidades reales y `numpy.complex128` para las cantidades complejas. Esta doble precisión es el estándar en la computación científica y es suficiente para mitigar errores de redondeo en la mayoría de los cálculos fonónicos, incluso para sistemas con simetrías complejas o degeneraciones cercanas.

Un punto crítico de estabilidad numérica surge en el tratamiento de modos de muy baja frecuencia (modos acústicos en $q = 0$ o modos blandos). Aunque la Ecuación 2 no depende explícitamente de la frecuencia, el factor de ocupación de Bose-Einstein sí lo hace (ver Sección 3.1.3). La implementación actual maneja estos casos en el cálculo del factor térmico, pero el cálculo intrínseco del PAM es robusto frente a frecuencias nulas.

Optimizaciones de Rendimiento: La principal optimización implementada es la **vectorización** completa del algoritmo. En lugar de utilizar bucles explícitos en Python para iterar sobre cada punto q , cada banda o cada átomo, se aprovecha la capacidad de NumPy para realizar operaciones elemento a elemento sobre arreglos multidimensionales. Como se observa en el pseudocódigo 3.1, las operaciones de extracción de componentes, multiplicación compleja, conjugación y suma se realizan en una sola línea de código para

todo el conjunto de datos.

Esta estrategia de diseño reduce drásticamente la sobrecarga del intérprete de Python, delegando los bucles intensivos a las rutinas subyacentes de NumPy, que están precompiladas en C o Fortran y altamente optimizadas. La complejidad computacional del algoritmo es lineal con respecto al tamaño del problema, lo que lo hace escalable a sistemas grandes y mallas de puntos q densas. Un análisis más detallado revela lo siguiente:

El bucle principal del algoritmo itera un número constante de veces (3, para cada componente cartesiana). Dentro de cada iteración, la operación más costosa es la multiplicación compleja elemento a elemento ('comp1.conj() * comp2'), que opera sobre arreglos de forma (N_q, N_b, N_a) . Esta operación tiene una complejidad de $O(N_q \cdot N_b \cdot N_a)$. La subsecuente suma sobre el eje de los átomos ('np.sum(..., axis=2)') debe recorrer la misma cantidad de elementos, por lo que comparte la misma clase de complejidad. En contraste, los pasos posteriores, como la aplicación del factor térmico, operan sobre arreglos de forma (N_q, N_b) , resultando en una complejidad menor de $O(N_q \cdot N_b)$. Por lo tanto, la complejidad total está dominada por los pasos que escalan con el número de átomos, como se resume en la Tabla 1.

Tabla 1. **Análisis de la complejidad computacional del cálculo del PAM.**

Parámetro	Símbolo
Número de puntos q	N_q
Número de bandas (modos)	N_b
Número de átomos en la celda	N_a
Operación	Complejidad
Cálculo del producto $e_{\beta}^* e_{\gamma}$	$O(N_q \cdot N_b \cdot N_a)$
Suma sobre los átomos	$O(N_q \cdot N_b \cdot N_a)$
Aplicación del factor térmico	$O(N_q \cdot N_b)$
Complejidad Total	$O(N_q \cdot N_b \cdot N_a)$

Fuente: Elaboración propia.

3.1.3. Tratamiento de la Temperatura: El momento angular fonónico, $I_{q\nu}$, es una propiedad intrínseca de un modo fonónico. Sin embargo, para obtener una cantidad físicamente observable o comparable con experimentos, es necesario considerar su valor esperado en un ensamble térmico. Este valor depende de la población promedio de fonones en el modo (q, ν) , la cual está determinada por la temperatura T .

El Factor de Ocupación de Bose-Einstein: La población de un modo fonónico a una temperatura T se describe por la distribución de Bose-Einstein. La implementación en `pam.py` no solo considera la población térmica, sino también la contribución de la energía del punto cero del oscilador armónico cuántico. El factor multiplicativo total, que representa el valor de expectación del hamiltoniano del modo, $\langle \hat{a}^{\dagger} \hat{a} + 1/2 \rangle$, es:

$$n'_B(\omega, T) = n_B(\omega, T) + \frac{1}{2} = \frac{1}{e^{\hbar\omega/k_B T} - 1} + \frac{1}{2} \quad (9)$$

donde \hbar es la constante de Planck reducida y k_B es la constante de Boltzmann. Este factor adimensional se multiplica por el PAM intrínseco para obtener el valor térmicamente ponderado. El código implementa esta fórmula directamente, como se muestra en el extracto de `phonon_angular_momentum`:

```
if temp == 0.0:
    nbose = 0.5
else:
    nbose = 0.5 + 1.0 / (np.exp(freq * THZ_TO_EV / (KB * temp)) - 1.0)
```

Se utilizan las constantes `THZ_TO_EV` y `KB` para asegurar la consistencia de unidades, convirtiendo la frecuencia de THz a eV y utilizando k_B en unidades de eV/K.

Casos Límite La implementación maneja correctamente los casos límite físicos:

- **Límite de $T \rightarrow 0$ K:** Cuando la temperatura es cero, la función evita la división por cero y la evaluación de $\exp(\infty)$ asignando directamente `nbose = 0.5`. Esto corresponde físicamente a la contribución del momento angular del punto cero, una consecuencia puramente cuántica.
- **Límite de Altas Temperaturas ($k_B T \gg \hbar\omega$):** En este régimen, la Ecuación 9 se aproxima a $n'_B(\omega, T) \approx k_B T / \hbar\omega$. El código maneja esto de forma natural a través de la evaluación de la función exponencial, sin necesidad de una aproximación explícita.

3.2. IMPLEMENTACIÓN DE PAMDOS EN `DOS.PY`

Mientras que el módulo `pam.py` se encarga del cálculo del Momento Angular Fonónico (PAM) para modos discretos, el análisis de las propiedades colectivas y la quiralidad glo-

bal del sistema requiere una herramienta estadística como la densidad de estados (DOS). Para este fin, se ha desarrollado una extensión significativa del módulo `dos.py` de PHONOPY, introduciendo una nueva clase, `PAMDos`, diseñada para calcular la DOS resuelta en PAM. Esta implementación no solo proporciona el espectro fonónico total, sino que lo descompone en contribuciones de modos con quiralidad positiva ($PAM > 0$) y negativa ($PAM < 0$) para cada eje cartesiano.

La decisión estratégica de extender el módulo `dos.py` existente, en lugar de crear un módulo completamente nuevo, se fundamenta en los principios de reutilización de código y mantenibilidad. Al heredar de la infraestructura de DOS de PHONOPY, la implementación de `PAMDos` aprovecha los métodos numéricos robustos y ya validados (método de *smearing* y método de tetraedros), así como la gestión de la malla de cálculo y el rango de frecuencias, permitiendo centrar el desarrollo en la nueva lógica de separación y ponderación por quiralidad.

3.2.1. Arquitectura de la Clase `PAMDos` La clase `PAMDos` está diseñada como una especialización de la clase base `Dos` de PHONOPY. Esta relación de herencia es fundamental para la arquitectura, ya que permite que `PAMDos` herede toda la funcionalidad esencial para el cálculo de una DOS estándar, como la inicialización de la malla de frecuencias, la selección del método de cálculo y las rutinas de escritura de datos.

Herencia y Especialización La clase se define como:

```
class PAMDos(Dos):  
    # ...
```

Esta herencia directa proporciona acceso inmediato a los atributos y métodos de la clase `Dos`, tales como `_frequencies`, `_weights`, `_sigma`, y el método `set_draw_area`. La especialización se logra mediante la adición de nuevos atributos específicos para el PAM y la sobrecarga del método `run` para incorporar la lógica de separación por quiralidad.

Atributos y Métodos Específicos: La clase PAMDOS introduce los siguientes atributos clave:

- `self.Jxyz`: Un arreglo NumPy de forma $(3, N_q, N_b)$ que almacena los valores de PAM precalculados para cada componente cartesiana, punto q y banda. Este atributo es la entrada fundamental que conecta el cálculo de DOS con los resultados del módulo `pam.py`.
- `self.dos_positive`: Un arreglo de forma $(3, N_f)$ que almacena la DOS para los modos con PAM positivo, para cada eje (x, y, z). N_f es el número de puntos en la malla de frecuencias.
- `self.dos_negative`: Análogo a `dos_positive`, pero para los modos con PAM negativo.
- `self.threshold`: Un pequeño valor de umbral para evitar la ambigüedad numérica de los valores de PAM cercanos a cero.

El método principal que se sobrecarga es `run()`, que ahora orquesta la separación de modos y el cálculo ponderado de la DOS. Adicionalmente, se implementan nuevos métodos para la visualización (`plot_pam_dos()`) y la escritura de datos (`write_pam_dos()`), adaptados al formato de salida de tres componentes.

3.2.2. Algoritmo de Separación por Signo y Ponderación por Magnitud: El concepto central de la PAM-DOS es descomponer la densidad de estados total, $g(\omega)$, en contribuciones quirales. Para una componente cartesiana α , la DOS se separa en $g_\alpha^+(\omega)$ y $g_\alpha^-(\omega)$.

$$g(\omega) \approx g_x^+(\omega) + g_x^-(\omega) + g_y^+(\omega) + g_y^-(\omega) + g_z^+(\omega) + g_z^-(\omega) \quad (10)$$

Esta separación no es una simple clasificación binaria. Para reflejar la "intensidad" de la quiralidad, cada modo fonónico contribuye a su respectiva DOS (positiva o negativa) con

un peso que es proporcional no solo a su multiplicidad en la Zona de Brillouin, sino también al **valor absoluto** de su momento angular fonónico, $|J_{q\nu}^\alpha|$. Este esquema de ponderación asegura que los modos con un PAM más grande tengan un mayor impacto en la DOS quiral, proporcionando una medida más física de la quiralidad del espectro. El algoritmo implementado para esta separación y ponderación es el siguiente, y se resume en el Pseudocódigo 3.2.

1. **Iteración por Eje Cartesiano:** El proceso se realiza de forma independiente para cada componente del PAM (l_x, l_y, l_z) .
2. **Enmascaramiento por Signo:** Para un eje α , se crean dos máscaras booleanas utilizando el arreglo `self.Jxyz[α]`:
 - `positive_mask`: True para todos los modos donde $l_{q\nu}^\alpha > \text{threshold}$.
 - `negative_mask`: True para todos los modos donde $l_{q\nu}^\alpha < -\text{threshold}$.

Estas máscaras tienen la misma forma que el arreglo de frecuencias y el de pesos de la malla $((N_q, N_b))$.

3. **Filtrado de Frecuencias y Pesos:** Las máscaras se aplican para seleccionar los subconjuntos de frecuencias y pesos de la Zona de Brillouin correspondientes a los modos de PAM positivo y negativo.
4. **Cálculo de Pesos Ponderados por PAM:** Aquí reside la lógica clave. Los pesos para el cálculo de la DOS no son solo los pesos de la malla $(w_{q\nu})$, sino que se ponderan por la magnitud del PAM:
 - `positive_weights` = $w_{q\nu} \times |l_{q\nu}^\alpha|$ para los modos en `positive_mask`.
 - `negative_weights` = $w_{q\nu} \times |l_{q\nu}^\alpha|$ para los modos en `negative_mask`.

5. **Cálculo de la DOS Parcial:** Se invoca una rutina de cálculo de DOS (sea por *smearing* o tetraedros, ver sección 3.2.3) dos veces: una con las frecuencias y pesos positivos para obtener $g_{\alpha}^{+}(\omega)$, y otra con los negativos para obtener $g_{\alpha}^{-}(\omega)$.

Listing 3.2. Algoritmo de separación y ponderación para PAM-DOS

```
function calculate_pam_dos(frequencies, weights_BZ, Jxyz, threshold):
    dos_positive = zeros((3, num_freq_points))
    dos_negative = zeros((3, num_freq_points))

    for axis in [x, y, z]:
        # 1. Crear mascararas booleanas
        positive_mask = Jxyz[axis] > threshold
        negative_mask = Jxyz[axis] < -threshold

        # 2. Seleccionar frecuencias y pesos de la malla
        pos_freqs = frequencies[positive_mask]
        neg_freqs = frequencies[negative_mask]
        pos_mesh_weights = weights_BZ[positive_mask]
        neg_mesh_weights = weights_BZ[negative_mask]

        # 3. Ponderar por la magnitud del PAM
        pos_pam_values = abs(Jxyz[axis][positive_mask])
        neg_pam_values = abs(Jxyz[axis][negative_mask])

        final_pos_weights = pos_mesh_weights * pos_pam_values
        final_neg_weights = neg_mesh_weights * neg_pam_values

        # 4. Calcular las DOS separadas
        dos_positive[axis] = _calculate_dos(pos_freqs, final_pos_weights)
        dos_negative[axis] = _calculate_dos(neg_freqs, final_neg_weights)
```

```
return dos_positive, dos_negative
```

3.2.3. Adaptación de los Métodos de Cálculo de DOS: Una vez que los modos han sido separados y sus pesos correspondientes calculados, la construcción de la curva de DOS continua se realiza utilizando los métodos numéricos ya implementados en PHONOPY, con las adaptaciones necesarias.

Método de Ensanchamiento Gaussiano (*Smearing*): El método de *Smearing* es computacionalmente eficiente y consiste en reemplazar cada delta de Dirac en la definición de la DOS por una función de ensanchamiento, típicamente una Gaussiana. La PAM-DOS para la componente α y quiralidad \pm se calcula como:

$$g_{\alpha}^{\pm}(\omega) = \frac{1}{N} \sum_{i \in S_{\alpha}^{\pm}} w_i |l_i^{\alpha}| \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\omega - \omega_i)^2}{2\sigma^2}\right) \quad (11)$$

donde la suma se realiza sobre el conjunto de modos S_{α}^{\pm} con PAM positivo o negativo para el eje α , w_i es el peso del modo i en la Zona de Brillouin, $|l_i^{\alpha}|$ es la magnitud de su PAM, y $G(\omega - \omega_i; \sigma)$ es la función Gaussiana con una anchura σ . La implementación en PAMDOS reutiliza la clase `NormalDistribution` de PHONOPY, pasando el conjunto ponderado de pesos (`final_pos_weights` o `final_neg_weights`) a la rutina de cálculo de DOS.

Método de Tetraedros: El método de tetraedros es un enfoque más sofisticado que evita el uso de un parámetro de ensanchamiento artificial. La Zona de Brillouin se tesela en un conjunto de tetraedros, y dentro de cada tetraedro, las frecuencias de los fonones se interpolan linealmente. Para adaptar este método al cálculo de la PAM-DOS, se ha modificado la contribución de cada tetraedro a la DOS.

El peso de la DOS de cada tetraedro ahora se multiplica por el promedio de las magnitudes del PAM de sus cuatro vértices. Si $\{\omega_k\}$ y $\{|l_k|\}$ son las frecuencias y las magnitudes

del PAM en los vértices $k = 1, 2, 3, 4$ de un tetraedro, la contribución ponderada a la DOS se calcula utilizando los pesos estándar del método de tetraedros multiplicados por un factor de ponderación:

$$\bar{l}_{\text{tetra}} = \frac{1}{4} \sum_{k=1}^4 |l_k| \quad (12)$$

Esta adaptación se implementa modificando el arreglo de coeficientes que se pasa a la rutina de C del método de tetraedros en PHONOPY. En lugar de un arreglo de unos (para la DOS total), se pasa un arreglo que contiene las magnitudes de PAM para cada modo. La separación por signo se logra ejecutando el método dos veces, una vez con un arreglo de coeficientes donde los PAM negativos se han hecho cero, y otra vez donde los PAM positivos se han hecho cero.

Tabla 2. **Comparación de los métodos de cálculo de DOS implementados para PAM-DOS.**

Método	Precisión	Costo Compu- tacional	Parámetros Requeridos
Ensanchamiento Gaussiano (<i>Smearing</i>)	Dependiente de σ . Puede generar artefactos de ensanchamiento artificial.	Bajo. Escala linealmente con el número de modos.	Ancho de la Gaussiana (σ).
Método de Tetraedros	Alta. No introduce ensanchamiento artificial y reproduce con exactitud las singularidades de Van Hove.	Moderado. Requiere teselación de la ZB y un cálculo más complejo por tetraedro.	Ninguno, aparte de la densidad de la malla.

Fuente: Elaboración propia.

3.2.4. Integración de la Densidad de Estados: Una de las aplicaciones más importantes de la PAM-DOS es cuantificar la quiralidad neta en un rango de frecuencias específico. Esto se logra integrando la diferencia entre la DOS de PAM positivo y la de PAM negativo.

$$\Delta N_{\alpha}(\omega_1, \omega_2) = \int_{\omega_1}^{\omega_2} [g_{\alpha}^{+}(\omega) - g_{\alpha}^{-}(\omega)] d\omega \quad (13)$$

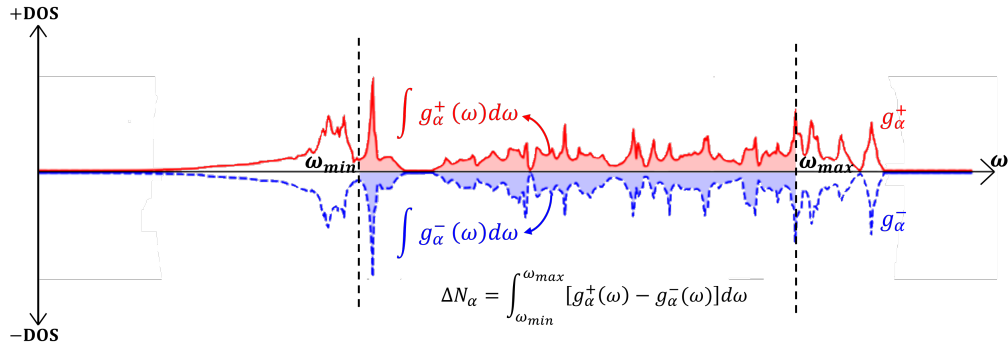
Un valor de ΔN_{α} significativamente diferente de cero indica un desequilibrio entre los modos de quiralidad derecha e izquierda para la componente α , lo que es una firma directa de la quiralidad fonónica del sistema en ese rango de energía.

Algoritmo de Integración: La integración se implementa numéricamente utilizando la regla del trapecio, que está disponible en NumPy a través de la función `numpy.trapz`. Este proceso se ilustra en la Figura 6 y el algoritmo es el siguiente:

1. Se definen los límites de integración, ω_{min} y ω_{max} .
2. Se seleccionan los segmentos de los arreglos `self.frequency_points`, `self.dos_positive`, y `self.dos_negative` que caen dentro de este rango.
3. Se aplica `numpy.trapz` a los segmentos de `dos_positive` y `dos_negative`, utilizando el segmento de `frequency_points` como el espaciado del eje x.
4. Se calcula la diferencia entre las dos integrales para obtener ΔN_{α} para cada eje.

Esta capacidad de integración es fundamental para el análisis cuantitativo y permite una comparación directa de la quiralidad entre diferentes materiales o bajo diferentes condiciones externas, como la presión o la deformación. Los resultados de esta sección serán clave para las conclusiones presentadas en capítulos posteriores de esta tesis.

Figura 6. **Visualización del proceso de integración del PAM-DOS.**



Fuente: Elaboración propia.

3.3. INTERFAZ DE USUARIO Y COMANDOS

La integración de una nueva capacidad computacional en un ecosistema de software científico consolidado como Phonopy exige un diseño cuidadoso de la interfaz de usuario. Una interfaz de línea de comandos (CLI, por sus siglas en inglés) bien diseñada no solo facilita el uso de las nuevas funcionalidades, sino que también garantiza la reproducibilidad de los cálculos y mantiene la coherencia con el paradigma operativo existente del software. El objetivo principal fue extender la CLI de Phonopy, específicamente el comando `phonopy-load`, para incorporar el cálculo del Momento Angular Fonónico (PAM) y la Densidad de Estados asociada (PAM-DOS) de una manera intuitiva y potente, sin perturbar el flujo de trabajo de los usuarios existentes.

La estrategia de implementación se centró en la adición de nuevos parámetros (flags) que activan y configuran los cálculos relacionados con el PAM. Esta aproximación, en lugar de la creación de un script independiente, permite aprovechar la robusta infraestructura de `phonopy-load` para la lectura de datos, la gestión de simetría y la preparación de cálculos. Se utilizó el módulo estándar de Python `argparse`, gestionado a través de los ficheros `phonopy/cui/phonopy_argparse.py` y `phonopy/cui/settings.py`, para definir y procesar los nuevos parámetros, asegurando una integración nativa y consistente.

3.3.1. Nuevos Parámetros de Línea de Comandos: Para habilitar los cálculos de PAM y PAM-DOS de forma intuitiva y reproducible, se extendió la interfaz de línea de comandos del script phonopy-load. Se introdujeron nuevos parámetros (flags) para activar los cálculos (`-pamdos`), proyectar el PAM sobre las bandas (`-pam-bands`), controlar la temperatura y gestionar la visualización. Este enfoque integra la nueva funcionalidad de manera coherente con el flujo de trabajo existente de PHONOPY. La Tabla 3 resume los nuevos parámetros, sus argumentos y su función principal.

Tabla 3. **Resumen de los nuevos parámetros de línea de comandos para el cálculo del PAM.**

Parámetro	Argumento	Descripción Detallada
<code>-pamdos</code>	Ninguno	Calcula PAM-DOS sobre la malla. Requiere autovectores; se desactiva la simetría de malla cuando es necesario para preservar la información de signo.
<code>-pam-bands</code>	Ninguno	Proyección de PAM en bandas. Opera en memoria y si faltan autovectores, se solicitan explícitamente.
<code>-int-pamdos</code>	Ninguno	Persiste las estadísticas de la integración de la PAM-DOS en la ventana de frecuencia definida por <code>-fmin</code> y <code>-fmax</code> .
<code>-pam-temperature</code>	FLOAT	Especifica la temperatura efectiva en Kelvin para ponderaciones dependientes de la ocupación de Bose-Einstein.
<code>-with-pam-bands</code>	Ninguno	Al graficar PAM-DOS, sobreimpone las bandas con proyección de PAM cuando se han calculado en la misma ejecución (modo <i>band_mesh</i>).
<code>-pam-cmap</code>	STRING	Permite la selección de un mapa de color (e.g., <i>seismic</i> , <i>viridis</i>) para la graficación de las proyecciones de PAM.
<code>-fmin, -fmax</code>	FLOAT	Definen el rango de frecuencias (mínimo y máximo) para la integración de la DOS cuando se utiliza <code>-int-pamdos</code> .

Fuente: Elaboración propia.

3.3.2. Integración con phonopy-load: La elección de integrar las nuevas funcionalidades dentro del script existente `phonopy-load` en lugar de crear un nuevo ejecutable fue una decisión de diseño estratégica. `phonopy-load` es el punto de entrada moderno para los cálculos de post-procesamiento en Phonopy, y su uso centraliza y simplifica el flujo de trabajo del usuario. Esta integración permite que los cálculos de PAM se beneficien directamente de la capacidad de `phonopy-load` para descubrir y cargar automáticamente ficheros esenciales como `phonopy_disp.yaml`, `FORCE_SETS`, `BORN`, etc.

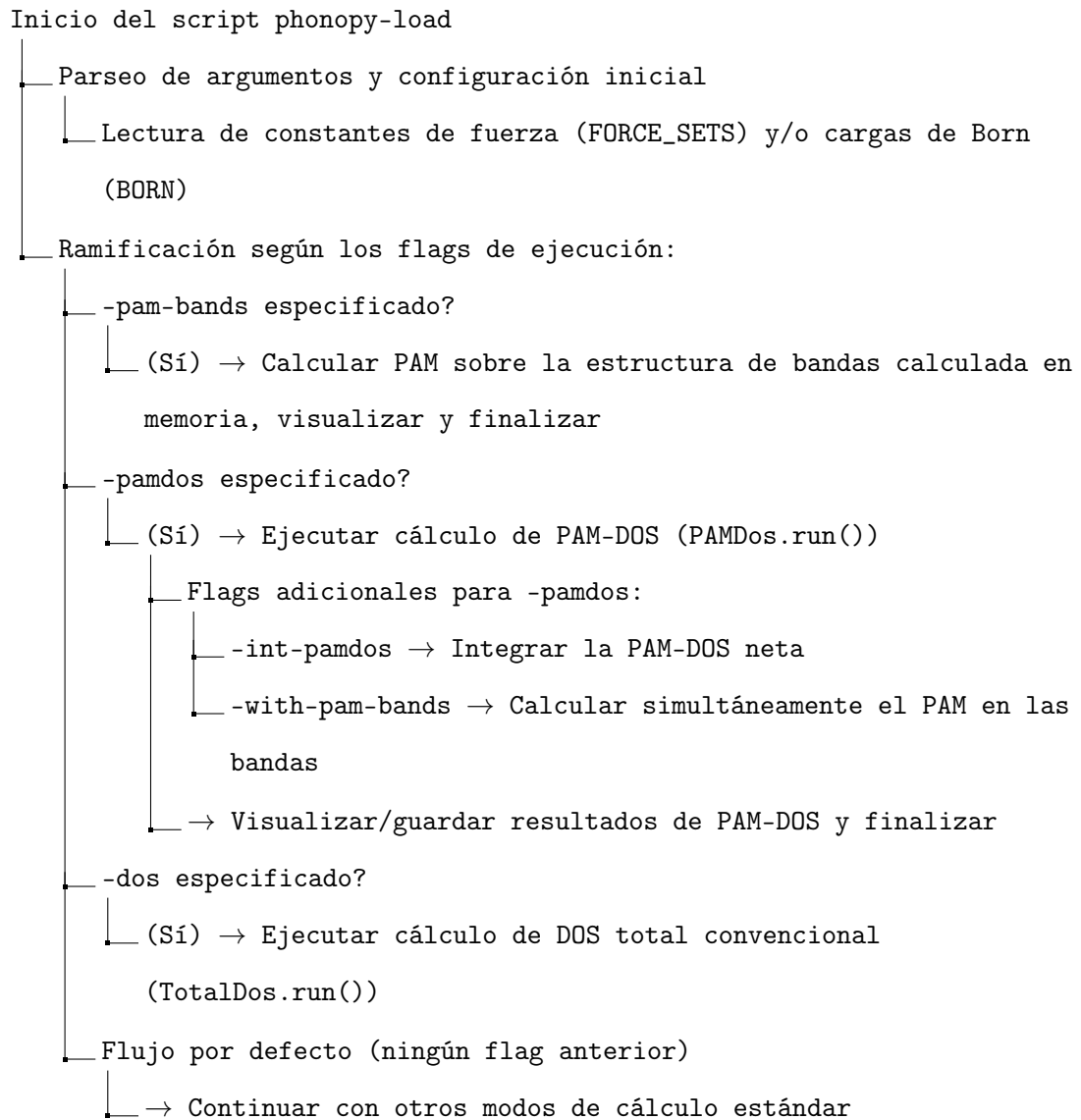
El flujo de ejecución dentro del módulo `phonopy/cui/phonopy_script.py` fue modificado para acomodar los nuevos modos de cálculo. Se implementó un árbol de decisión que prioriza los nuevos flags sobre los cálculos de DOS convencionales. El Diagrama 7 ilustra este flujo lógico.

Como se observa en el diagrama, el script primero verifica la presencia de los flags de mayor especificidad. Si se detecta `-pam-bands`, el programa calcula la proyección del PAM sobre la estructura de bandas calculada en la misma ejecución, operando sobre los datos en memoria; si faltan autovectores, se reejecutan las bandas con `with_eigenvectors`. Si se encuentra `-pamdos`, se inicia el cálculo sobre la malla de puntos-q, y flags adicionales como `-int-pamdos` o `-with-pam-bands` actúan como modificadores de este modo principal. La sobreimpresión de bandas con PAM (`-with-pam-bands`) requiere que en la misma ejecución se hayan calculado bandas y malla (modo *band_mesh*); en caso contrario, el flujo recae de forma segura en solo PAM-DOS. Solo si ninguno de los flags de PAM está presente, el programa procede a considerar los cálculos de DOS convencionales (total o proyectada). Este diseño jerárquico asegura que no haya ambigüedad en la intención del usuario y mantiene una estructura de control limpia y extensible.

3.3.3. Artefactos de salida Para trazabilidad y reproducibilidad, la ejecución genera archivos y figuras con nombres estandarizados:

- Datos: `pam_dos.dat` (PAM-DOS), `pam_bands.dat` (bandas con PAM), `int_pamdos`

Figura 7. **Árbol de decisión procesal que ilustra la lógica de ejecución del script *phonopy-load* modificado.**



Fuente: Elaboración propia.

`_results.dat` (integrales acumuladas).

- Figuras: `pam_dos.png`, `pam.png`, `pam_dos_with_bands.png`.
- Diagnóstico (PAM-DOS): `pam_raw_data.dat`, `pam_statistics.dat`.

3.3.4. Compatibilidad y Versionado: La introducción de nuevas funcionalidades en un software científico debe gestionarse con una atención especial a la compatibilidad hacia atrás y al control de versiones, para garantizar la reproducibilidad de los resultados científicos.

Estrategia de Versionado: La implementación de las funcionalidades de PAM y PAM-DOS se integró sobre la rama `develop` del proyecto base, correspondiente a la versión 2.43.2. Se mantuvo compatibilidad hacia atrás al no modificar formatos de entrada ni flujos estándar, y se añadieron nuevas capacidades de forma optativa vía API/CLI. El código fuente completo con las adiciones desarrolladas en este trabajo se encuentra alojado en un repositorio de control de versiones²⁴.

Formato de Ficheros: Los nuevos cálculos no modifican los formatos de ficheros de entrada existentes como `FORCE_SETS` o `BORN`. Los resultados se escriben en ficheros nuevos (`pam_dos.dat`, `pam_bands.dat`) o extienden los ficheros YAML existentes de manera compatible, asegurando que los scripts de análisis preexistentes que no dependen de la información de PAM sigan funcionando sin modificaciones. La estructura de los ficheros de salida de PAM está diseñada para ser fácilmente parseable y auto-contenida.

Esta aproximación disciplinada al diseño de la CLI y al versionado asegura que la extensión del software con las capacidades de PAM sea robusta, fácil de usar y se integre de

²⁴ El repositorio se encuentra en la siguiente URL: https://github.com/brayanb1701/phonopy_pam/tree/feature/pam. Actualmente, el acceso es privado con el objetivo de publicar los resultados de la investigación en un artículo científico en el futuro.

forma natural en el ecosistema de Phonopy, preservando al mismo tiempo la integridad y reproducibilidad de la investigación científica.

3.4. HERRAMIENTAS ADICIONALES DE VISUALIZACIÓN

La interpretación de las dispersiones de fonones en sistemas multicomponente, como es el caso del compuesto $TaSi_2$ investigado en esta tesis, presenta un desafío significativo. Una curva de dispersión estándar, que relaciona la frecuencia fonónica (ω) con el vector de onda (q), no ofrece información explícita sobre la naturaleza vibracional de los modos; es decir, no permite discernir qué subred atómica o qué elemento específico domina la dinámica de una banda fonónica particular. Esta limitación es especialmente crítica en el estudio de los fonones quirales, donde la quiralidad emerge de patrones de movimiento atómico complejos que involucran a distintas especies atómicas. Para superar esta deficiencia y facilitar un análisis más profundo del carácter atómico de los modos fonónicos, se ha desarrollado una herramienta de post-procesamiento computacional como parte integral de este trabajo de investigación.

Este conjunto de herramientas consiste en un script de Python, denominado 'plot_phonon_dispersion_color.py', diseñado para generar visualizaciones avanzadas de la estructura de bandas de fonones. A diferencia de las representaciones convencionales, este script colorea las bandas de dispersión de acuerdo con la contribución relativa de cada especie atómica a los eigenvectores fonónicos. Este enfoque de "dispersión proyectada por elementos" permite una identificación visual inmediata de qué átomos (e.g., Ta o Si) participan predominantemente en cada modo fonónico a lo largo de las trayectorias de alta simetría en la zona de Brillouin. La capacidad de discernir el carácter atómico de las bandas es fundamental para correlacionar la quiralidad fonónica con movimientos específicos de la subred Kagome o de los átomos de tántalo, proporcionando una capa de análisis cualitativo y cuantitativo que no es accesible a través de los métodos de visualización estándar.

El script fue concebido como una herramienta de post-procesamiento autónoma y modular, escrita en Python 3. Su diseño se fundamenta en el uso de bibliotecas científicas estándar como NumPy para el cálculo numérico eficiente, Matplotlib para la generación de gráficos de alta calidad para publicaciones, y PyYAML para el análisis sintáctico (parsing) de los ficheros de salida de PHONOPY. La arquitectura del script está orientada a la línea de comandos, utilizando el módulo 'argparse' para proporcionar una interfaz de usuario flexible y configurable.

3.4.1. Arquitectura y Flujo de Procesamiento de Datos El flujo de trabajo del script se puede descomponer en las siguientes etapas lógicas:

1. **Entrada de Datos:** El script toma como entrada principal un fichero 'band.yaml' generado por PHONOPY, que contiene la información completa de la estructura de bandas (frecuencias, eigenvectores, vectores de onda, etc.). Adicionalmente, requiere un fichero de configuración de colores (por defecto, 'elements-VESTA.ini.txt') que mapea cada símbolo químico a un color RGB específico, permitiendo una personalización completa de la paleta de visualización.
2. **Análisis Sintáctico (Parsing):** Se utiliza un analizador YAML optimizado ('CSafeLoader' de la biblioteca LibYAML, si está disponible) para leer eficientemente el fichero 'band.yaml'. El script extrae las frecuencias (ω), los eigenvectores complejos ($e_{q\nu}$), los vectores de onda (q) y la información estructural, como las posiciones y símbolos de los átomos en la celda primitiva.
3. **Cálculo de Contribuciones Elementales:** Para cada punto (q, ν) en la dispersión, se invoca un algoritmo para calcular el peso o contribución de cada especie atómica. Este es el núcleo computacional del script. El resultado es una matriz de pesos $W_E(q, \nu)$, donde E representa una especie atómica.
4. **Mapeo de Color:** Los pesos calculados $W_E(q, \nu)$ se utilizan para determinar el color

de cada punto de la banda de dispersión. Se han implementado dos algoritmos de coloración, un método de mezcla RGB y un método de dispersión (scatter).

5. **Generación del Gráfico:** Utilizando la biblioteca Matplotlib, el script renderiza la estructura de bandas. Para el método de mezcla RGB, se emplea la clase 'LineCollection', que es altamente eficiente para dibujar líneas con segmentos de colores variables. Para el método de dispersión, se superponen puntos coloreados sobre una curva base. El gráfico se decora con ejes, etiquetas, líneas de alta simetría y una leyenda de colores para los elementos.
6. **Salida:** El gráfico resultante se guarda en un fichero de imagen (e.g., PNG, PDF, SVG) con una resolución y tamaño personalizables, listo para ser incluido en publicaciones científicas.

3.4.2. Algoritmo de Cálculo de Pesos Elementales El algoritmo para determinar la contribución de cada elemento a un modo fonónico específico (\mathbf{q} , ν) se basa en la proyección de los eigenvectores. El eigenvector $\mathbf{e}_{\mathbf{q}\nu}$ es un vector complejo de dimensión $3N_{atom}$, donde N_{atom} es el número de átomos en la celda primitiva. Este vector se puede descomponer en componentes atómicas tridimensionales $\mathbf{e}_{\mathbf{q}\nu,i}$ para cada átomo i .

1. **Cálculo de la Magnitud del Desplazamiento Atómico:** Para cada átomo i en la celda primitiva, se calcula la magnitud cuadrada de su vector de desplazamiento, sumando las contribuciones de las tres direcciones cartesianas:

$$A_i^2(\mathbf{q}, \nu) = |\mathbf{e}_{\mathbf{q}\nu,i}|^2 = |e_{\mathbf{q}\nu,i,x}|^2 + |e_{\mathbf{q}\nu,i,y}|^2 + |e_{\mathbf{q}\nu,i,z}|^2 \quad (14)$$

Dado que los eigenvectores de PHONOPY están normalizados, la suma de estas

magnitudes sobre todos los átomos es la unidad:

$$\sum_{i=1}^{N_{atom}} A_i^2(\mathbf{q}, \nu) = 1 \quad (15)$$

2. **Agrupación por Especie Atómica:** El peso o contribución de una especie atómica E (e.g., Ta o Si) se define como la suma de las magnitudes cuadradas de todos los átomos i que pertenecen a esa especie:

$$W_E(\mathbf{q}, \nu) = \sum_{i \in \text{átomos de especie } E} A_i^2(\mathbf{q}, \nu) \quad (16)$$

Estos pesos, por construcción, están normalizados a la unidad para cada modo fonónico, es decir, $\sum_E W_E(\mathbf{q}, \nu) = 1$. Esto permite interpretarlos como la fracción de la energía cinética vibracional total del modo que es aportada por cada especie atómica.

3. **Implementación Numérica:** Este cálculo se implementa de forma vectorizada utilizando NumPy. El eigenvector de $3N_{atom}$ se remodela a una matriz de $(N_{atom}, 3)$. El cuadrado del valor absoluto se calcula para obtener las magnitudes y, a continuación, se suman las componentes cartesianas. Finalmente, se utilizan máscaras booleanas o indexación avanzada para agrupar y sumar los pesos correspondientes a cada elemento, evitando bucles explícitos en Python y garantizando una alta eficiencia computacional.

3.4.3. Algoritmos de Coloración: Una vez calculados los pesos elementales W_E para cada punto de la dispersión, el siguiente paso es traducir esta información cuantitativa en una representación visual. Se implementaron dos métodos distintos de coloración para ofrecer diferentes perspectivas analíticas.

Método de Mezcla RGB Ponderada: Este es el método por defecto, activado con la opción ‘–method mixed’. La idea es crear un color compuesto para cada punto de la banda que refleje la mezcla de las contribuciones de todos los elementos.

Algoritmo: Mezcla de Colores Ponderada.

1. **Obtener Colores Base:** Para cada elemento E , se obtiene su color RGB base, $C_{E,RGB} = (R_E, G_E, B_E)$, a partir del fichero de colores proporcionado. Estos valores se normalizan en el rango $[0, 1]$.
2. **Calcular Color Compuesto:** El color final $C_{final,RGB}$ para un modo fonónico (\mathbf{q}, ν) se calcula como una combinación lineal de los colores base, ponderada por los pesos elementales $W_E(\mathbf{q}, \nu)$:

$$C_{final,RGB} = \sum_E W_E(\mathbf{q}, \nu) \cdot C_{E,RGB} \quad (17)$$

Componente a componente, esto se traduce en:

$$R_{final} = \sum_E W_E(\mathbf{q}, \nu) \cdot R_E$$

$$G_{final} = \sum_E W_E(\mathbf{q}, \nu) \cdot G_E$$

$$B_{final} = \sum_E W_E(\mathbf{q}, \nu) \cdot B_E$$

3. **Renderizado con LineCollection:** Para dibujar la banda, se crea una colección de segmentos de línea (‘LineCollection’ de Matplotlib) que conectan puntos de \mathbf{q} adyacentes. A cada segmento se le asigna un color que es el promedio de los colores compuestos de sus dos puntos extremos. Esta técnica es extremadamente eficiente, ya que permite dibujar una banda completa con colores variables en una sola llamada a la función de trazado.

Este método produce bandas con colores suaves y degradados que indican visualmente el cambio en el carácter atómico a lo largo de una banda. Es particularmente útil para identificar regiones de hibridación de modos, donde múltiples especies atómicas contribuyen de manera significativa.

Método de Puntos de Dispersión (Scatter): Una alternativa es el método de dispersión ('-method scatter'), que se enfoca en resaltar la contribución dominante en lugar de mezclarla.

1. **Trazar Bandas Base:** Primero, todas las bandas de dispersión se dibujan como líneas delgadas de color gris para servir como guía visual.
2. **Superponer Puntos Coloreados:** Para cada elemento E , se itera sobre todos los puntos (\mathbf{q}, ν) de la dispersión. Se dibuja un punto de dispersión (scatter point) en esa posición, utilizando el color base $C_{E,RGB}$.
3. **Ponderación Visual:** La prominencia de cada punto se modula en función del peso $W_E(\mathbf{q}, \nu)$. Esto se logra ajustando el tamaño del punto y/o su opacidad (canal alfa). Típicamente, el tamaño del punto es proporcional a W_E . Para evitar un exceso de puntos, se establece un umbral (e.g., $W_E > 0,01$), de modo que solo se dibujan las contribuciones significativas.

Este método es excelente para identificar de forma inequívoca qué elemento domina una banda en una región particular, ya que los colores no se mezclan. El resultado es un gráfico "punteado" donde el color y el tamaño de los puntos revelan la composición atómica de los modos.

3.5. DOCUMENTACIÓN Y GUÍAS DE USO

Un componente esencial para la adopción, reproducibilidad y sostenibilidad de cualquier software científico es una documentación clara y completa. Conscientes de ello, se ha

dedicado un esfuerzo significativo a la creación de materiales que no solo explican cómo usar las nuevas funcionalidades del Momento Angular Fonónico (PAM), sino que también guían al usuario a través de ejemplos prácticos y reproducibles. Este esfuerzo se materializó en dos entregables principales: una extensión de la documentación oficial de Phonopy y un ejemplo de aplicación autocontenido que incluye un cuaderno interactivo de Jupyter.

3.5.1. Documentación para el Usuario y la Comunidad La documentación oficial de Phonopy, construida con Sphinx a partir de ficheros Markdown, fue extendida para incorporar de manera integral las nuevas funcionalidades de PAM. El objetivo fue que cualquier usuario, desde uno nuevo hasta un desarrollador experimentado, pudiera encontrar la información necesaria de forma rápida y contextual. Las contribuciones se centraron en los siguientes ficheros:

- `doc/pam.md`: Se creó una página de documentación completamente nueva y dedicada exclusivamente al Momento Angular Fonónico. Este fichero centraliza la información y actúa como la fuente principal de verdad para la funcionalidad. Explica los conceptos teóricos clave, los flujos de trabajo desde la línea de comandos (CLI), la semántica de la temperatura en los cálculos, la normalización de los mapas de color para las bandas y los artefactos (ficheros de salida) que se generan. Además, incluye un apartado de "Inicio Rápido" para la API, mostrando cómo realizar los cálculos de PAM-DOS y proyección en bandas mediante la programación en Python.
- `doc/command-options.md`: Este fichero, que lista todas las opciones de la línea de comandos de Phonopy, fue actualizado para incluir una nueva sección titulada "PAM options". En ella se documentan de forma detallada todos los nuevos parámetros introducidos, como `-pamdos`, `-pam-bands`, `-with-pam-bands`, `-int-pamdos`, `-pam-temperature` y `-pam-cmap`, explicando su propósito y cómo interactúan entre sí.
- `doc/phonopy-load.md`: La documentación del comando `phonopy-load`, que es el

punto de entrada para los nuevos cálculos, se actualizó con una sección de "PAM workflows". Esta sección proporciona ejemplos canónicos y directos de cómo ejecutar los tres casos de uso principales: cálculo de PAM-DOS, proyección de PAM en bandas y la visualización combinada, reforzando el aprendizaje a través de ejemplos prácticos.

3.5.2. Ejemplo Práctico y Cuaderno Interactivo Para complementar la documentación formal y garantizar la reproducibilidad de los resultados, se creó un directorio de ejemplo autocontenido en `example/pam/TaSi2/`. Este directorio no solo incluye los ficheros de entrada necesarios para ejecutar un cálculo completo en el compuesto TaSi_2 , sino que también proporciona una guía interactiva que facilita enormemente la curva de aprendizaje.

- **Ficheros de Ejemplo:** El directorio contiene un fichero `phonopy.yaml` preconfigurado y el fichero `FORCE_CONSTANTS`, permitiendo al usuario ejecutar directamente los cálculos de post-procesamiento sin necesidad de realizar costosos cálculos de primeros principios.
- **Fichero `README.md`:** Un fichero `README.md` conciso proporciona un resumen de los comandos necesarios para ejecutar los diferentes flujos de trabajo de PAM, sirviendo como una referencia rápida.
- **Cuaderno Interactivo (`PAM_TaSi2_Tutorial.ipynb`):** Este cuaderno de Jupyter es el entregable más significativo de esta sección. Representa una guía interactiva y ejecutable que conduce al usuario a través de todo el proceso:
 1. Clonación del repositorio y configuración de un entorno virtual de Python aislado.
 2. Instalación de Phonopy desde el código fuente.

3. Ejecución paso a paso de cada uno de los flujos de trabajo de PAM (PAM-DOS, bandas y combinado).
4. Visualización de los gráficos generados directamente en el cuaderno.
5. Inspección de los ficheros de datos de salida.

Este enfoque no solo documenta el proceso, sino que lo convierte en un artefacto científico reproducible. Un usuario puede ejecutar el cuaderno celda por celda y verificar que obtiene exactamente los mismos resultados, lo que constituye el estándar de oro para la validación y la transferencia de conocimiento en la ciencia computacional moderna.

En conjunto, estos materiales de documentación y el ejemplo práctico aseguran que la funcionalidad desarrollada sea accesible, comprensible y verificable por toda la comunidad científica, fomentando su adopción y futuras contribuciones.

4. VERIFICACIÓN Y VALIDACIÓN

4.1. PROPÓSITO Y ALCANCE

Este capítulo formaliza el proceso de Verificación y Validación (V&V) del módulo de análisis de Momento Angular Fonónico (PAM) y su densidad de estados asociada (PAM-DOS), integrado en Phonopy ¹¹⁸. El objetivo es doble: (i) verificar que el software implementado cumple fielmente las especificaciones técnicas (correctitud funcional y numérica); y (ii) validar que los resultados que produce son físicamente coherentes con las simetrías y predicciones conocidas en la literatura.

El enfoque sigue la metodología definida en el plan del proyecto, con actividades de pruebas de componentes (unidad conceptual), pruebas de integración sobre el flujo CLI y validaciones físicas basadas en simetrías y en un caso de estudio representativo (TaSi₂). Para evitar redundancias, la validación detallada del caso de estudio se presenta en el Capítulo 5; aquí se establecen los criterios, los procedimientos de control de calidad y los indicadores de aceptación.

4.2. VERIFICACIÓN DEL SOFTWARE

La verificación responde a la pregunta “¿se construyó el producto correctamente?” e incluye pruebas de componentes e integración. Se establece un umbral de tolerancia numérica explícito para cada tipo de prueba, considerando redondeo en coma flotante y variaciones menores por parámetros de ensanchado en la DOS.

4.2.1. Estrategia de pruebas y criterios de aceptación

- **Pruebas de componentes (unidad conceptual):** verifican el núcleo vectorizado de PAM a partir de autovectores complejos normalizados por masa y masas atómicas.
- **Pruebas de integración (end-to-end):** verifican la interacción entre el núcleo de

PAM, el integrador de DOS y la CLI. Criterio: generación consistente de archivos de salida esperados, reproducibilidad de resultados bajo la misma configuración.

4.2.2. Implementación de Pruebas Unitarias Para garantizar la robustez y correctitud del código, se implementó una suite de pruebas unitarias utilizando el framework `pytest`. Estas pruebas, localizadas en los archivos `test/phonon/test_pam_math.py` y `test/phonon/test_dos.py`, automatizan la verificación de los componentes clave del nuevo módulo.

Pruebas del núcleo matemático (`test_pam_math.py`) Este conjunto de pruebas se enfoca en la función central `phonon_angular_momentum`, que calcula el PAM sobre un conjunto de fonones. Las verificaciones clave incluyen:

- **Correctitud numérica y física:** Se valida que el factor de Bose-Einstein se aplique correctamente cuando se introduce una temperatura finita, asegurando que el peso térmico de los fonones sea físicamente coherente.
- **Manejo de unidades:** Se comprueba que la función produce resultados consistentes independientemente de las unidades de frecuencia de entrada (e.g., THz o cm^{-1}), validando el factor de conversión interno.
- **Robustez ante formatos de entrada:** Las pruebas confirman que la función maneja correctamente diferentes formatos de autovectores (matricial o tensorial), una característica importante para la integración con las estructuras de datos de `Phonopy`.
- **Gestión de errores:** Se asegura que el sistema falle de manera controlada y emita un error informativo (`ValueError`) cuando los datos de entrada tienen dimensiones incorrectas, previniendo fallos inesperados.

Pruebas de la Densidad de Estados (`test_dos.py`) Estas pruebas se centran en la clase `PAMDOS`, responsable de calcular y procesar la densidad de estados resuelta en PAM. Los casos de prueba más relevantes son:

- **Contrato de la API y dimensiones de salida:** Se verifica que los cálculos de PAM-DOS produzcan arreglos numéricos con las dimensiones esperadas (e.g., componentes l_x, l_y, l_z para las contribuciones positivas y negativas), garantizando una interfaz predecible para otros componentes del sistema.
- **Generación de artefactos:** Se comprueba que funcionalidades como la integración de la PAM-DOS (`-int-pamdos`) y la exportación de datos crudos y estadísticas generen los archivos de salida correspondientes (`int_pamdos_results.dat`, `pam_raw_data.dat`, `pam_statistics.dat`), validando los flujos de trabajo completos.
- **Comportamiento de parámetros:** Se prueba la lógica de negocio, como el parámetro de umbral (*threshold*). Una prueba específica confirma que un umbral extremadamente alto anula correctamente todas las contribuciones a la PAM-DOS, asegurando que el filtrado de bajo momento angular funcione como se espera.

Esta suite de pruebas automatizadas es fundamental para el control de calidad continuo, permitiendo refactorizaciones y futuras extensiones del código con un alto grado de confianza en que la funcionalidad existente no se vea comprometida.

4.2.3. Pruebas de integración (CLI) Se verifican los flujos principales expuestos en la CLI `phonopy-load` y sus artefactos asociados:

- **PAM-DOS** (`-pamdos`): genera `pam_dos.dat` y `pam_dos.png`; la opción `-int-pamdos` persiste `int_pamdos_results.dat`. Cuando aplica para preservar el signo del observable, se deshabilita la reducción de malla por simetría.
- **Bandas con PAM** (`-pam-bands`): genera `pam.png` y `pam_bands.dat`. Si los autovectores no están presentes en memoria, se reejecuta la estructura de bandas con

`with_eigenvectors.`

- **Vista combinada** (`-with-pam-bands`): cuando en la misma ejecución se calculan bandas y malla (modo *band_mesh*), se genera `pam_dos_with_bands.png`; si no hay datos de bandas en memoria, el flujo recae en la generación de `pam_dos.png`.

4.3. VALIDACIÓN FÍSICA

La validación responde a la pregunta “¿se construyó el producto correcto?” y confirma que el software reproduce comportamientos físicos esperados:

Caso con simetría de Inversión temporal: para sistemas con simetría de inversión temporal, \mathcal{T} , la integral acumulada de PAM-DOS positiva y negativa coincide dentro de tolerancia.

Esto se detallará con más profundidad en el siguiente capítulo, donde se analizará con detalle para el caso de estudio seleccionado.

5. RESULTADOS: CASO DE ESTUDIO Y DEMOSTRACIÓN (TASi₂)

Este capítulo demuestra las capacidades y la validez del módulo de software implementado a través de un caso de estudio completo con el material quiral TaSi₂. El enfoque principal es presentar los resultados como una demostración de las funcionalidades de la herramienta. Para un análisis físico más profundo y detallado de los fonones quirales en TaSi₂, el lector puede consultar el **Anexo C**. Este capítulo sigue una narrativa de investigación, mostrando cómo cada funcionalidad de la herramienta permite responder preguntas científicas progresivamente más profundas. Cada sección presenta una capacidad del software, explica el resultado físico que genera y detalla cómo la implementación de ingeniería permite obtener dicha visión. El objetivo es ilustrar el ciclo completo: desde la pregunta científica hasta la respuesta habilitada por la herramienta de software desarrollada.

5.1. DETALLES METODOLÓGICOS DE LOS CÁLCULOS COMPUTACIONALES.

Los cálculos teóricos, asociados a la estructura electrónica y posteriormente, la obtención de las matrices de fuerza, presentados en este trabajo se realizaron en el marco de la Teoría de los Funcionales de la Densidad (DFT) utilizando con su implementación *Vienna Ab initio Simulation Package* (VASP). A continuación, se detalla el flujo de trabajo computacional completo, desde la optimización estructural hasta el cálculo de las propiedades fonónicas y el momento angular fonónico (PAM); esta información se presenta para permitir la reproducibilidad de los resultados obtenidos. Para consultar datos suplementarios, como tablas de parámetros de red o valores numéricos adicionales, véase el **Anexo A**. Es importante aclarar que los archivos de input para phonopy se incluyen como ejemplo directamente en la carpeta de ejemplos de la librería.

5.1.1. Cálculos de Primeros Principios: Opt. Estructural y Prop. Electrónicas. El punto de partida para todos los cálculos fue la definición de la estructura cristalina del

TaSi₂. El procedimiento para obtener una estructura de equilibrio precisa y para calcular sus propiedades electrónicas fundamentales se dividió en los siguientes pasos:

1. **Funcional y Pseudopotenciales:** La interacción entre los electrones de valencia y los núcleos iónicos se describió mediante el método del proyector de ondas aumentadas (PAW). Para el funcional de intercambio y correlación, se empleó la Aproximación de Gradiente Generalizado (GGA) en la parametrización de Perdew-Burke-Ernzerhof (PBE). Se utilizaron los siguientes pseudopotenciales de la librería de VASP:

- PAW_PBE Ta 17Jan2003, tratando 5 electrones de valencia ($6s^15d^4$).
- PAW_PBE Si 05Jan2001, tratando 4 electrones de valencia ($3s^23p^2$).

2. **Optimización Estructural:** Se realizó un proceso de relajación estructural en dos etapas para asegurar una alta convergencia tanto de los parámetros de red como de las posiciones iónicas. En ambos pasos, se permitió la relajación completa de la forma, el volumen y los iones de la celda (ISIF = 3).

- **Paso 1 (Relajación Gruesa):** Se partió de una estructura inicial y se relajó utilizando una energía de corte para las ondas planas de 600 eV (ENCUT = 600) y una malla de puntos-k de $5 \times 5 \times 3$ centrada en Gamma.
- **Paso 2 (Relajación Fina):** Tomando la estructura resultante del paso anterior (CONTCAR), se realizó una segunda optimización con una malla de puntos-k más densa de $9 \times 9 \times 7$ para garantizar una descripción precisa del espacio recíproco.

El criterio de convergencia para las fuerzas en los iones se estableció en 10^{-3} eV/Å (EDIFFG = -1E-03). Todos los cálculos se realizaron de forma espín-polarizada (ISPIN = 2) y se utilizó el método de Methfessel-Paxton de primer orden (ISMear = 1) con un ancho de $\sigma = 0,02$ eV, adecuado para sistemas metálicos.

3. **Cálculo de Propiedades Electrónicas:** Sobre la estructura final completamente relajada, se realizó un cálculo auto-consistente estático ($NSW = 0$) con la malla de $9 \times 9 \times 7$. La estructura de bandas electrónicas se calculó a lo largo de las trayectorias de alta simetría, y la Densidad de Estados (DOS) total y proyectada por orbitales se obtuvo activando la bandera `LORBIT = 11`.

5.1.2. Cálculos Fonónicos: Constantes de Fuerza. Las propiedades vibracionales de la red se calcularon mediante el método de las diferencias finitas, orquestado por el código `Phonopy`.

1. **Generación de Supercelda y Desplazamientos:** A partir de la celda unitaria primitiva optimizada, `Phonopy` construyó una supercelda de $2 \times 2 \times 2$, resultando en una estructura de 72 átomos. Se generó un conjunto mínimo de desplazamientos atómicos no equivalentes por simetría, con una amplitud de 0.01 Å.
2. **Cálculo de Fuerzas:** Para cada una de las superceldas desplazadas, se realizó un cálculo de fuerzas estático con VASP. En estos cálculos se utilizó una energía de corte de 500 eV (`ENCUT = 500`) y una malla de puntos-k de $4 \times 4 \times 3$ para la supercelda.
3. **Construcción de la Matriz de Constantes de Fuerza:** `Phonopy` recolectó el conjunto de pares de desplazamientos y fuerzas para construir la matriz de constantes de fuerza de segundo orden (armónicas), la cual contiene toda la información sobre la dinámica de la red en la aproximación armónica.

5.1.3. Post-procesamiento: Dispersión Fonónica y Momento Angular Con la matriz de constantes de fuerza como punto de partida, se realizaron los análisis finales para obtener las propiedades fonónicas quirales.

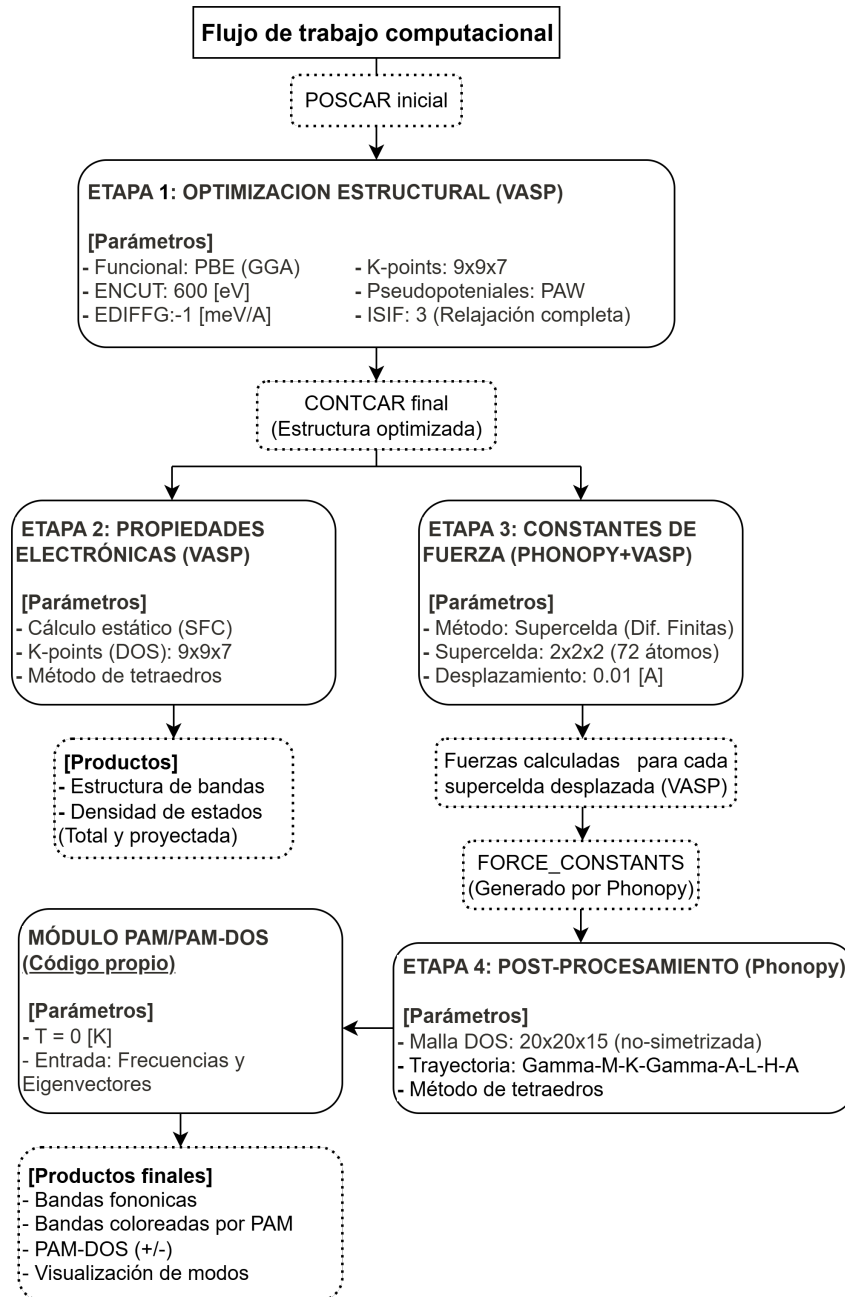
1. **Dispersión de Fonones:** La estructura de bandas fonónicas se obtuvo diagonali-

zando la matriz dinámica a lo largo de la trayectoria de alta simetría $\Gamma \rightarrow M \rightarrow K \rightarrow \Gamma \rightarrow A \rightarrow L \rightarrow H \rightarrow A$, según se definió en el archivo `band.conf`.

2. **Cálculo del Momento Angular Fonónico (PAM):** Para cada modo fonónico (\mathbf{q}, ν) a lo largo de la trayectoria de bandas, se calcularon las tres componentes cartesianas del PAM (l_x, l_y, l_z) utilizando el módulo en Python desarrollado en este trabajo. Todos los cálculos de PAM se realizaron en el límite de $T=0$ K.
3. **Cálculo de la PAM-DOS:** La Densidad de Estados Resuelta en Momento Angular (PAM-DOS) se calculó sobre una malla uniforme y densa de puntos- \mathbf{q} de $20 \times 20 \times 15$ en toda la Zona de Brillouin, sin aplicar reducciones de simetría para preservar los autovectores en todos los puntos \mathbf{q} . Posteriormente, se utilizó nuestro módulo para separar las contribuciones de los modos con PAM positivo ($l_z > 0$) y negativo ($l_z < 0$), ponderando cada estado por la magnitud de su momento angular, $|l_z|$, y utilizando el método de los tetraedros para una interpolación precisa.

La Figura 8 resume el flujo de trabajo completo para generar las salidas con VASP, Phonopy y el módulo implementado.

Figura 8. **Visualización de la metodología para obtener las gráficas haciendo uso de VASP, Phonopy y el código propio implementado.**



Fuente: Elaboración propia.

Las siguientes secciones se centran en el posprocesamiento, demostrando qué análisis permite realizar el software desarrollado a través de una narrativa de investigación guiada.

5.2. ANÁLISIS ESTADÍSTICO: DENSIDAD DE ESTADOS RESUELTA EN PAM

Para obtener una primera evidencia de la quiralidad en el sistema, la herramienta calcula la Densidad de Estados Resuelta en Momento Angular (PAM-DOS). Esta funcionalidad, invocada desde la CLI, descompone la DOS total en contribuciones de fonones con PAM positivo y negativo. Las Figuras 9 y 10 muestran este resultado para los dos enantiómeros de TaSi₂.

Figura 9. PAM-DOS para TaSi₂ (SG 180). Las líneas continuas (PAM >0) y discontinuas (PAM <0) muestran una distribución no trivial de quiralidad a lo largo del espectro de frecuencias.

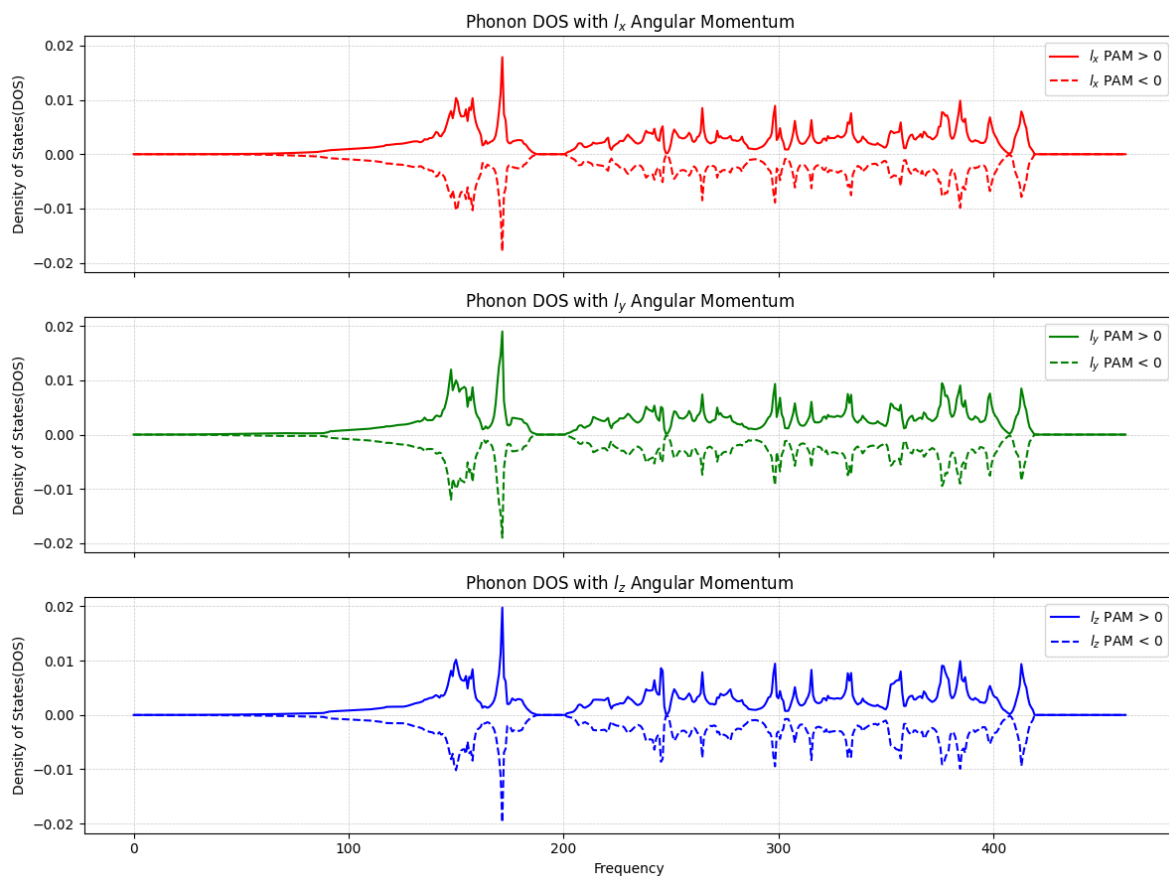
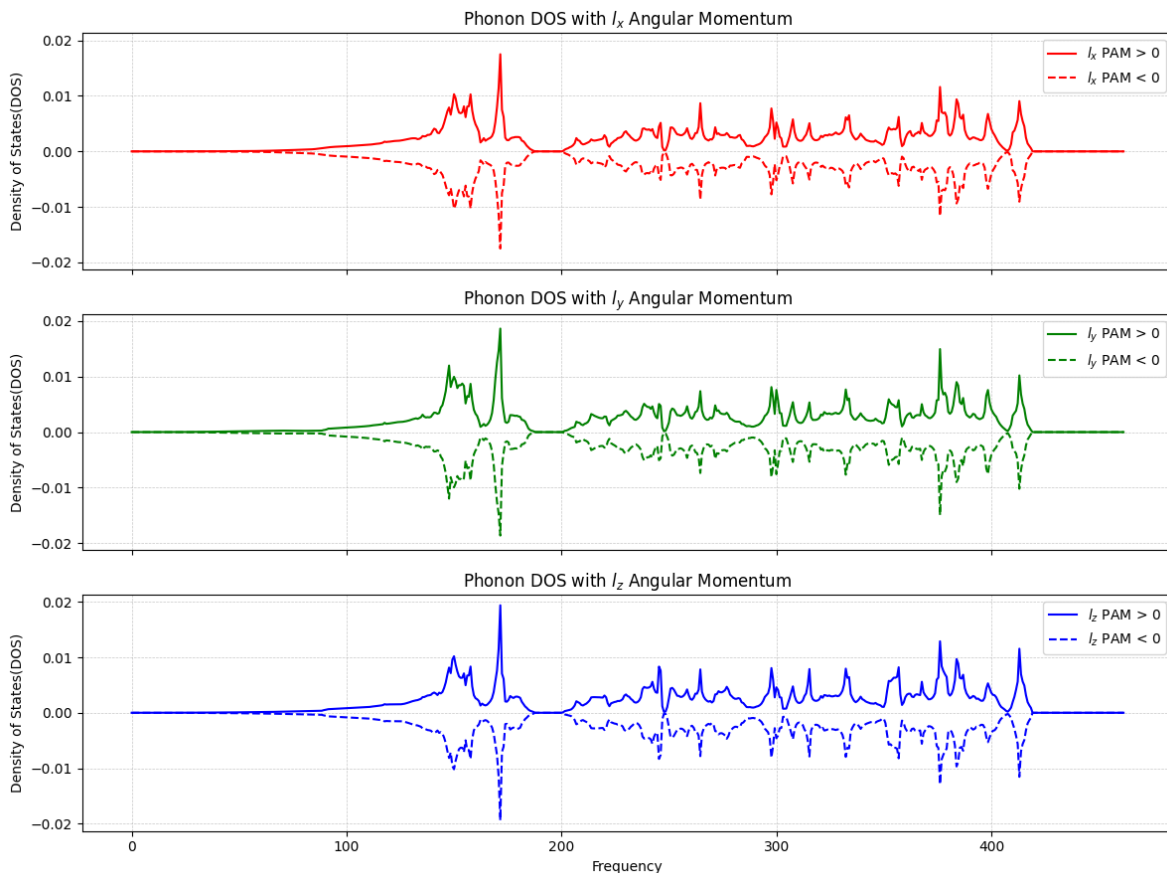


Figura 10. PAM-DOS para TaSi₂ (SG 181). La estructura es casi idéntica a la de su enantiómero (Fig. 9), un primer indicio de su relación especular.



Como se observa en las figuras, los resultados para ambos enantiómeros revelan que, aunque la integral acumulada de las contribuciones positiva y negativa de la PAM-DOS coincide dentro de la tolerancia (por simetría de inversión temporal), existen regiones de frecuencia con un claro desequilibrio entre modos de quiralidad positiva y negativa, lo que confirma la presencia de fonones quirales. Este análisis estadístico es posible gracias a la clase PAM_{DOS}, que fue diseñada para heredar de la infraestructura de Dos de Phonopy. El proceso implica calcular el PAM para cada modo en una malla densa, separar los modos por signo, ponderar su contribución por la magnitud del PAM y, finalmente, reutilizar los integradores de Phonopy para generar las curvas. Este diseño modular demuestra una

integración limpia y eficiente.

5.3. ANÁLISIS ESPECÍFICO: BANDAS COLOREADAS POR PAM

Una vez confirmada la presencia de quiralidad, el siguiente paso es localizarla en el espacio de momentos. El módulo logra esto mediante una de sus visualizaciones más potentes: la superposición del valor del PAM como un mapa de color sobre la estructura de bandas fonónicas. Como se aprecia en las Figuras 11 y 12, esta capacidad ofrece revelaciones cruciales.

Figura 11. Bandas fonónicas y PAM-DOS para TaSi₂ (SG 180). El color de las bandas (rojo: PAM >0, azul: PAM <0) revela la localización del PAM en el espacio recíproco.

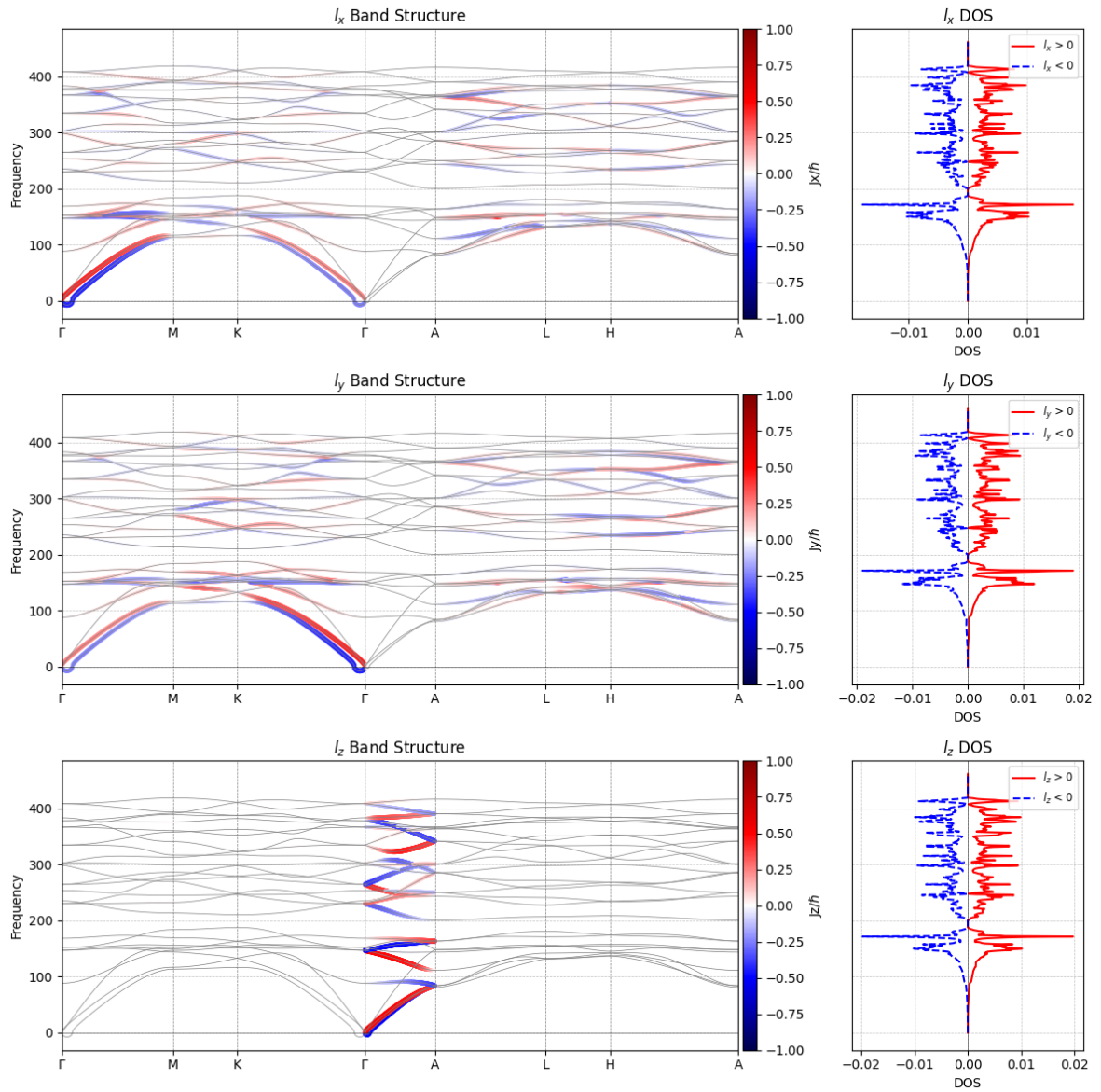
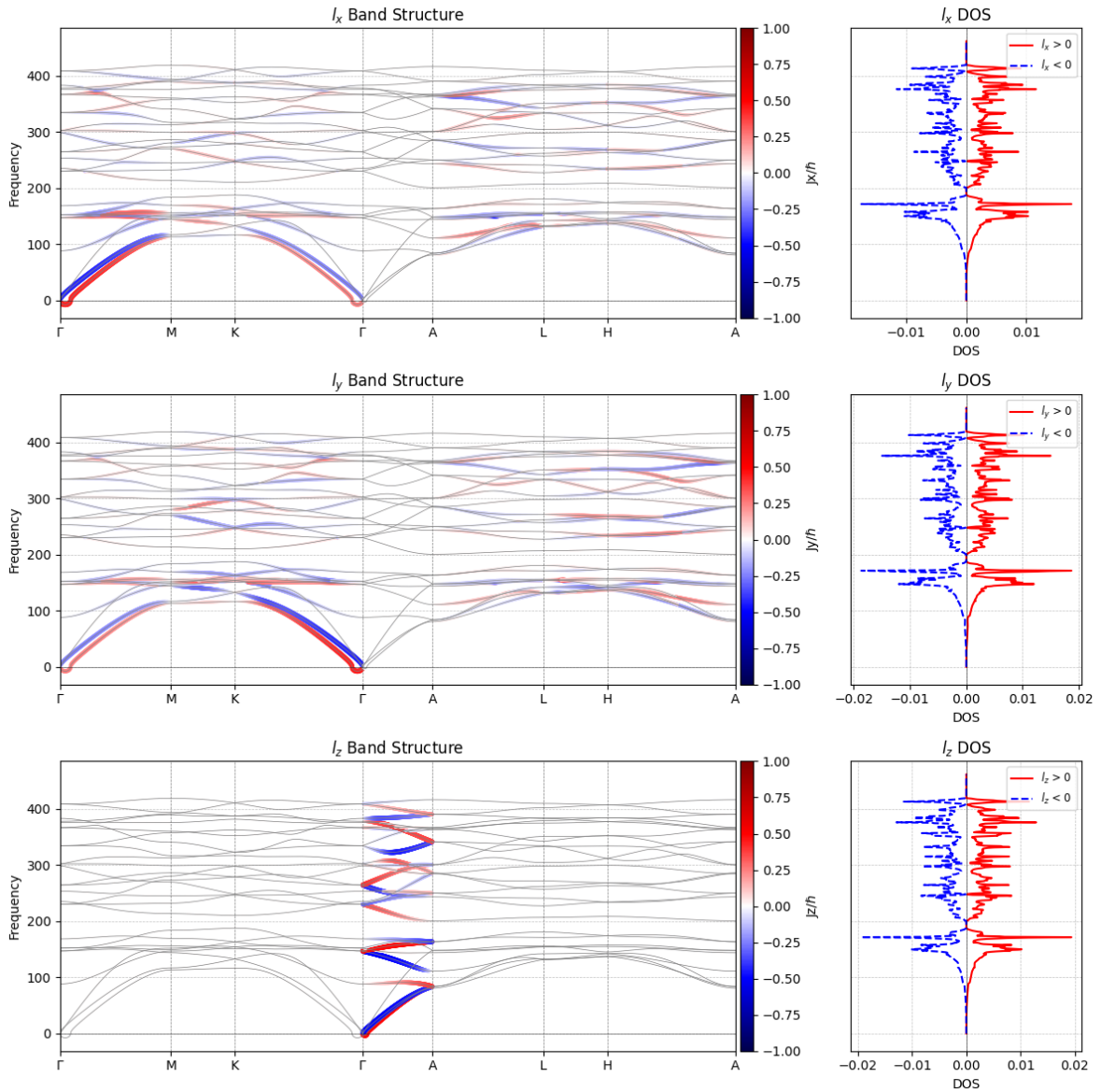


Figura 12. Bandas fonónicas y PAM-DOS para TaSi₂ (SG 181). La inversión sistemática del color de las bandas es la prueba definitiva de la quiralidad opuesta y valida la correctitud física del software.



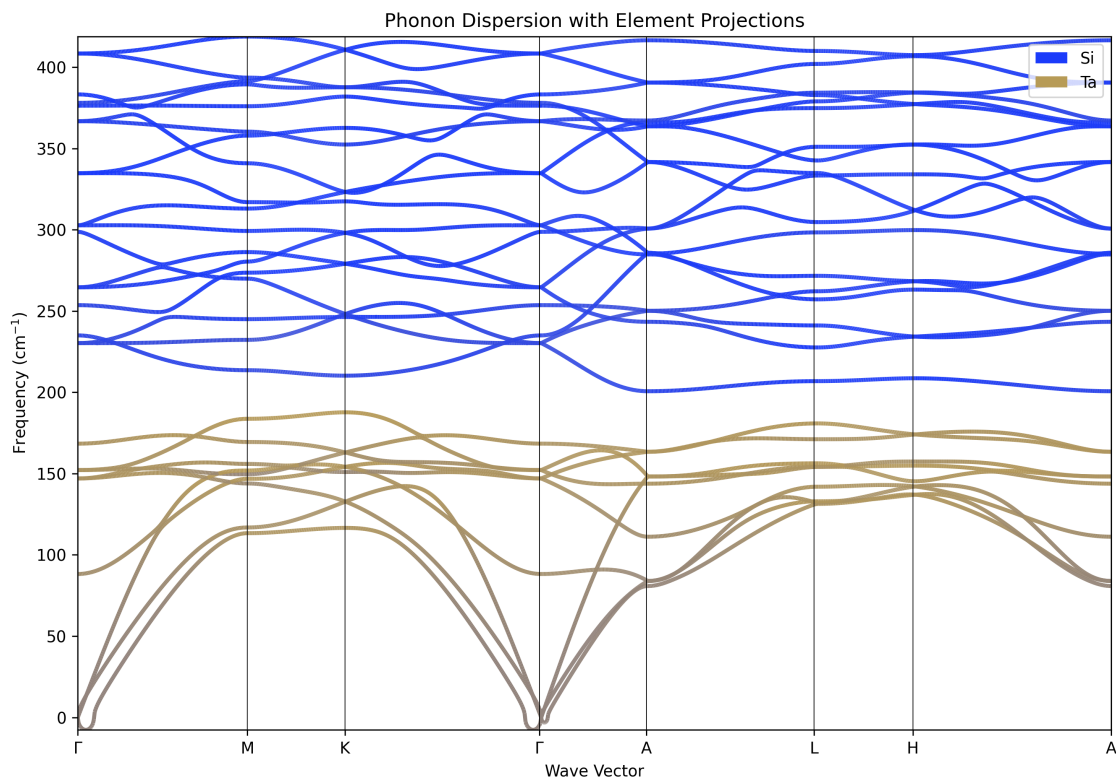
Primero, se observa que el PAM no es uniforme, sino que se concentra fuertemente a lo largo de la trayectoria $\Gamma \rightarrow A$, que corresponde al eje quiral del cristal. Segundo, y de vital importancia para la validación del software, la comparación entre ambas figuras muestra una **inversión sistemática del signo del PAM**. Este resultado, que confirma la correc-

ta implementación de la física, se obtiene aprovechando los autovectores complejos en memoria de la corrida de bandas; si no están presentes, la CLI reejecuta las bandas con `with_eigenvectors` y el módulo calcula el PAM por modo para enriquecer la visualización.

5.4. INTERPRETACIÓN DE ORIGEN: INTEROPERABILIDAD CON PROYECCIONES NATIVAS

Para interpretar el origen atómico de los modos quirales, el módulo demuestra su capacidad de interoperar con las funcionalidades nativas de `Phonopy`. En lugar de reinventar el análisis de proyecciones, el usuario puede emplear la herramienta estándar de proyección por especies atómicas (Figura 13) y cruzarla con los resultados del PAM.

Figura 13. Bandas fonónicas proyectadas por especie atómica (TaSi_2 , SG 180). Las bajas frecuencias están dominadas por Ta (ocre) y las altas por Si (azul). Esta información permite inferir el origen atómico de los modos quirales.



La figura revela una clara separación entre las vibraciones de los átomos pesados de Ta (baja frecuencia) y los ligeros de Si (alta frecuencia). Al combinar esta información con las figuras anteriores, se puede inferir el origen atómico de los fonones quirales. Desde la ingeniería, este flujo de trabajo resalta un principio clave del diseño: la reutilización de datos.

5.5. VALIDACIÓN VISUAL: EXPORTACIÓN DE DATOS PARA ANIMACIÓN

Como confirmación visual definitiva, se puede inspeccionar directamente el movimiento atómico de un modo quiral. Aunque el módulo no es una herramienta de animación, su diseño flexible permite exportar los datos brutos del PAM y los autovectores, que pueden ser utilizados por scripts externos para reconstruir la dinámica del modo. Las Figuras 14 y 15 muestran el resultado para el mismo modo en ambos enantiómeros.

Figura 14. Secuencia de desplazamiento para un modo quiral en SG 180. Se observa un sentido de rotación horario, consistente con un PAM positivo calculado por el módulo.

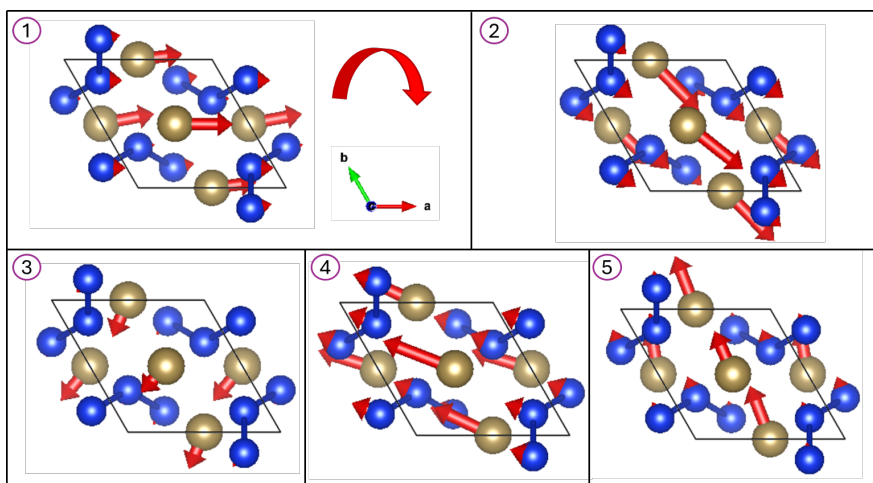
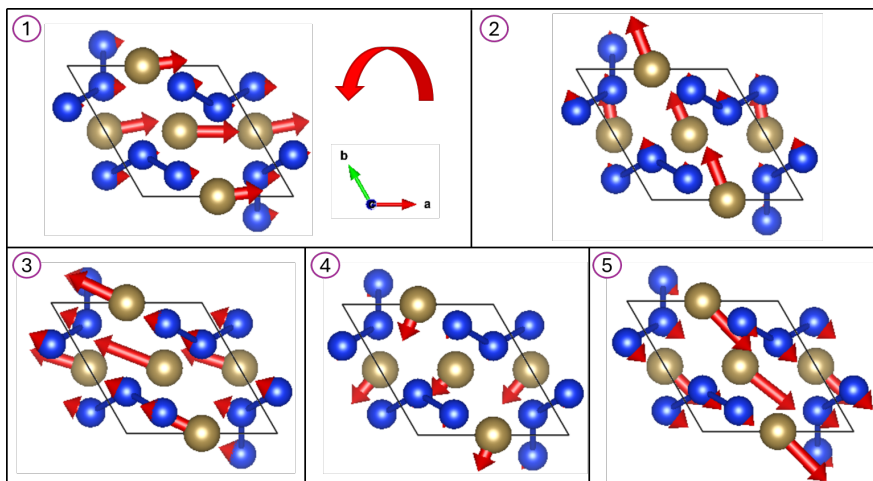


Figura 15. Secuencia de desplazamiento para el mismo modo en SG 181. El sentido de rotación es ahora antihorario, una confirmación visual directa de la inversión del signo del PAM.



El resultado es inequívoco: se observa un sentido de rotación horario (SG 180) y antihorario (SG 181), correspondiendo exactamente al signo del PAM calculado por el software.

6. CONCLUSIONES Y TRABAJO FUTURO

6.1. CONCLUSIONES

Este trabajo abordó el diseño, la implementación, la integración y la validación de un módulo de software para el análisis del Momento Angular Fonónico (PAM) y su densidad de estados (PAM-DOS) dentro del ecosistema de `Phonopy`¹. Desde una perspectiva de ingeniería de sistemas, se establecen las siguientes conclusiones clave:

1. **Se validó una estrategia de integración arquitectónica de bajo riesgo en un sistema complejo.** Más que una simple adición de código, se demostró un modelo de extensión no intrusivo en un ecosistema de software científico maduro. Al acoplarse a la capa de posprocesamiento y reutilizar las interfaces de datos existentes, se garantizó la compatibilidad y se minimizó el riesgo de regresión. Esta aproximación constituye una conclusión fundamental sobre cómo extender software heredado de manera sostenible, preservando su estabilidad y facilitando la mantenibilidad a largo plazo.
2. **Se tradujo un concepto físico complejo en un flujo de trabajo de usuario accesible y reproducible.** El diseño de la interfaz de línea de comandos (CLI) y la API no fue un mero añadido, sino un ejercicio de usabilidad centrado en el investigador. Se logró encapsular la complejidad del formalismo del PAM en comandos y parámetros intuitivos, integrados de forma natural en el flujo de trabajo de `Phonopy`. Esto demuestra el valor de la ingeniería de sistemas como un puente que transforma modelos teóricos en herramientas prácticas, reduciendo la barrera de entrada para la comunidad científica.
3. **La eficiencia computacional se consolidó como un habilitador directo de la investigación científica.** La implementación del núcleo de cálculo mediante vectorización con `NumPy` trascendió la mera optimización técnica. El rendimiento logrado

es un requisito habilitante que permite el análisis de sistemas con un gran número de átomos y el uso de mallas de muestreo densas en el espacio recíproco, condiciones indispensables para que los resultados de la PAM-DOS tengan resolución y significancia física.

- 4. Se estableció la confianza en el software como un instrumento científico fiable.** El proceso de Verificación y Validación (V&V) fue la etapa culminante que transformó el código de un prototipo a una herramienta robusta. La verificación de invariantes físicas (como el comportamiento del PAM en el límite acústico) y la validación de resultados (demostrada por la correcta inversión de signo del PAM entre los enantiómeros de TaSi₂) no solo confirman la correctitud del software, sino que construyen la confianza necesaria para su uso en investigación. Este paso es la conclusión del ciclo de vida del desarrollo, entregando un producto de software validado y listo para su contribución a la comunidad.

En suma, se completó un ciclo de vida de desarrollo de software completo: análisis del sistema base, diseño arquitectónico, implementación eficiente, integración de la CLI, y una fase de V&V alineada con mejores prácticas de ingeniería de software científico. El resultado es un módulo robusto, validado e integrado, listo para su adopción por parte de usuarios de Phonopy y para su evolución futura.

6.2. TRABAJO FUTURO

La presente versión establece una base sólida, pero ofrece oportunidades de mejora en rendimiento, cobertura funcional y calidad de ingeniería. A continuación se delinean líneas de trabajo priorizadas, inspiradas en las limitaciones y proyecciones identificadas durante el desarrollo del módulo PAM/PAM-DOS.

Rendimiento y escalabilidad

- **Paralelización por puntos- q** : El cálculo en la malla de puntos q es un problema inherentemente paralelizable. Se puede implementar una distribución de puntos q entre procesos usando `mpi4py` para clústeres o `multiprocessing` para nodos individuales. El reto de ingeniería consiste en implementar una operación de reducción (`MPI_Reduce` o similar) que agregue los histogramas de PAM-DOS de forma determinista, garantizando la equivalencia numérica con el cálculo secuencial.
- **Aceleración en GPU y compilación JIT**: Evaluar el reemplazo de `NumPy` por `CuPy` para delegar las operaciones de álgebra lineal (diagonalización, productos vectoriales) a GPUs NVIDIA. Adicionalmente, se puede usar `Numba` para compilar en tiempo de ejecución (JIT) los núcleos de cálculo más costosos, manteniendo la portabilidad del código Python.
- **Reintroducción de simetrías**: Investigar heurísticas para reintroducir un subconjunto de operaciones de simetría que no inviertan la quiralidad. Esto podría reducir el número de puntos q irreducibles sin comprometer el signo del PAM, optimizando el balance entre coste computacional y precisión física.

Gestión de memoria

- **Cómputo por lotes (*on-the-fly*)**: Para mitigar el alto consumo de RAM, se puede adaptar la estrategia de la clase `IterMesh` de `Phonopy`. Se implementaría un iterador que calcule fonones y PAM para un bloque de puntos- q , acumule su contribución a la PAM-DOS y libere la memoria antes de procesar el siguiente bloque. Esto reduce el requisito de memoria a sublineal a costa de un posible aumento en el tiempo de E/S.
- **Modo de baja memoria**: Ofrecer una ruta de ejecución explícita que priorice la conservación de memoria sobre el tiempo de ejecución, documentando claramente los compromisos de rendimiento y precisión para el usuario.

Extensiones funcionales y científicas

- **Integración con efectos anarmónicos:** Extender la compatibilidad para integrar el cálculo de PAM con `phono3py`. Esto permitiría investigar cómo las interacciones fonón-fonón (anarmonicidad) afectan la vida media y las frecuencias de los fonones quirales, proporcionando una descripción más precisa de la quiralidad a temperaturas finitas.
- **Cálculo de propiedades de transporte quiral:** Implementar el formalismo para calcular la curvatura de Berry de las bandas de fonones a partir de los autovectores. La integración de esta curvatura sobre la Zona de Brillouin permitiría el cálculo de propiedades de transporte como la conductividad Hall térmica de fonones, conectando directamente el PAM con una observable macroscópica.
- **Acoplamiento con otros grados de libertad:** Diseñar interfaces para acoplar los resultados del PAM con otros tipos de excitaciones. Por ejemplo, con cálculos de acoplamiento electrón-fonón para investigar la dispersión de electrones por fonones quirales, o con cálculos de ondas de espín para estudiar la hibridación entre magnones y fonones quirales en materiales magnéticos.

Calidad, V&V y automatización

- **Pruebas de regresión y datos “golden”:** Conjuntos mínimos reproducibles (YAML/HDF5) para validar PAM por modo y PAM-DOS; invariantes (límite acústico, inversión de signo entre enantiómeros) y tolerancias documentadas.
- **Integración continua (CI):** Pipelines con pruebas unitarias y de integración en matrices de versiones de Python/NumPy y sistemas operativos.

Usabilidad y adopción

- **CLI y mensajes de diagnóstico:** Refinar ayuda contextual y errores accionables (por ejemplo, guía cuando se requiere desactivar simetría para preservar el signo del PAM).
- **Contribución oficial:** Preparar una propuesta/*pull request* a Phonopy con API estable, pruebas, documentación y ejemplos, facilitando la revisión y la mantenibilidad a largo plazo.

Estas líneas de evolución consolidan el valor de ingeniería del módulo: mejoran eficiencia y escalabilidad, amplían su alcance analítico y refuerzan su calidad y usabilidad, manteniendo la coherencia con la arquitectura de Phonopy y con las decisiones de diseño que guiaron esta integración.

BIBLIOGRAFÍA

- Bass, Len, Paul Clements y Rick Kazman. *Software Architecture in Practice*. 3.^a ed. Addison-Wesley Professional, 2012 (vid. pág. 18).
- Born, Max y Kun Huang. *Dynamical Theory of Crystal Lattices*. Oxford University Press, 1954 (vid. pág. 22).
- Coh, Sinisa. "Classification of materials with phonon angular momentum and microscopic origin of angular momentum". En: *Physical Review B* 108.13 (2023), pág. 134307 (vid. págs. 114, 115).
- Fecher, Gerhard H, Jürgen Kübler y Claudia Felser. "Chirality in the solid state: Chiral crystal structures in chiral and achiral space groups". En: *Materials* 15.17 (2022). <https://www.mdpi.com/1996-1944/15/17/5812>, pág. 5812 (vid. págs. 15, 24).
- García Castro, Andrés Camilo. "Interplay between the B-sites' valence states and phonon dynamics in the Ca₃BO (B= Si, Ge, Sn, and Pb) antiperovskites". En: *Available at SSRN 5030298* () (vid. pág. 23).
- Harris, Charles R. et al. "Array programming with NumPy". En: *Nature* 585 (2020), págs. 357-362. DOI: 10.1038/s41586-020-2649-2 (vid. pág. 21).
- Hohenberg, Pierre y Walter Kohn. "Inhomogeneous electron gas". En: *Physical review* 136.3B (1964), B864 (vid. pág. 21).
- Kohn, Walter y Lu Jeu Sham. "Self-consistent equations including exchange and correlation effects". En: *Physical review* 140.4A (1965), A1133 (vid. pág. 21).

- Li, Wu et al. "ShengBTE: A solver of the Boltzmann transport equation for phonons". En: *Computer Physics Communications* 185.6 (2014), págs. 1747-1758. DOI: 10.1016/j.cpc.2014.02.015 (vid. pág. 26).
- Martin, Robert C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall, 2018 (vid. pág. 19).
- Oliphant, Travis E. "Python for Scientific Computing". En: *Computing in Science & Engineering* 9.3 (2007), págs. 10-20. DOI: 10.1109/MCSE.2007.58 (vid. pág. 21).
- Perez-Riverol, Yasset et al. "Ten simple rules for taking advantage of Git and GitHub". En: *PLOS Computational Biology* 12.7 (2016), e1004947. DOI: 10.1371/journal.pcbi.1004947 (vid. pág. 20).
- Stodden, Victoria et al. "Enhancing reproducibility for computational methods". En: *Science* 354.6317 (2016), págs. 1240-1241. DOI: 10.1126/science.aah6168 (vid. pág. 18).
- Stokes, Harold T. y Dorian M. Hatch. "FINDSYM: program for identifying the space-group symmetry of a crystal". En: *Journal of Applied Crystallography* 38.1 (2005), págs. 237-238 (vid. pág. 116).
- Tadano, Terumasa y Shinji Tsuneyuki. "ALM: A Python library for crystal lattice dynamics". En: *Journal of the Physical Society of Japan* 87.4 (2018), pág. 041015. DOI: 10.7566/JPSJ.87.041015 (vid. pág. 25).
- Togo, Atsushi. "First-principles phonon calculations with phonopy and phono3py". En: *Journal of the Physical Society of Japan* 92.1 (2023). <https://journals.jps.jp/doi/full/10.7566/JPSJ.92.012001>, pág. 012001 (vid. págs. 15, 32, 35, 72, 89).

- Togo, Atsushi. “First-principles Phonon Calculations with Phonopy and Phono3py”. En: *J. Phys. Soc. Jpn.* 92.1 (2023), pág. 012001. DOI: 10.7566/JPSJ.92.012001 (vid. págs. 21, 22).
- “First-principles phonon calculations with phonopy and phono3py”. En: *Journal of the Physical Society of Japan* 92.1 (2023), pág. 012001 (vid. págs. 22, 25, 72).
- Togo, Atsushi e Isao Tanaka. “Spglib: a software library for crystal symmetry search”. En: *arXiv preprint arXiv:1808.01590* (2018). <https://arxiv.org/abs/1808.01590> (vid. pág. 30).
- Wilson, Greg et al. “Good enough practices in scientific computing”. En: *PLoS Computational Biology* 13.6 (2017), e1005510. DOI: 10.1371/journal.pcbi.1005510 (vid. págs. 16, 19).
- Zhang, Lifa y Qian Niu. “Angular momentum of phonons and the Einstein–de Haas effect”. En: *Physical Review Letters* 112.8 (2014). <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.112.085503>, pág. 085503 (vid. págs. 15, 20, 23, 103, 109, 112).
- Zhao, H. et al. “Observation of chiral phonons”. En: *Nature Physics* 16.7 (2020), págs. 751-755. DOI: 10.1038/s41567-020-0903-7 (vid. pág. 26).
- Zheng, Qijing. *Phonon Angular Momentum*. Accedido el 2025-11-05. 2022. URL: <https://qijingzheng.github.io/posts/Angular-Momentum-in-Solids/> (vid. pág. 26).

ANEXO A

Anexo A. Datos suplementarios: tablas y secuencias de modos fonónicos

Este anexo reúne, en un único lugar, los datos suplementarios usados para la validación y demostración de la herramienta desarrollada: (i) tablas de frecuencias y momento angular proyectado en z (l_z/\hbar) por modo y (ii) secuencias de desplazamiento de modos fonónicos seleccionados. Las figuras referenciadas fueron generadas con el flujo de trabajo descrito en el cuerpo principal del documento. En las Tablas 4 y 5 se listan valores de referencia para los grupos espaciales 180 y 181 en $\mathbf{q} = (0, 0, 1/4)$. Las Figuras 16, 17, 18 y 19 muestran secuencias de desplazamiento para modos representativos.

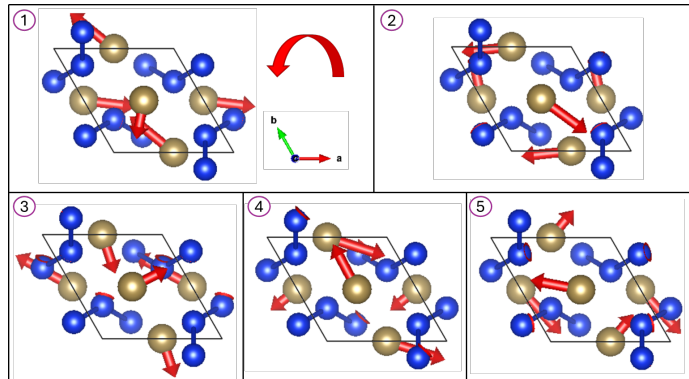
Tabla 4. *Datos de frecuencia y momento angular en z para los fonones correspondientes al grupo de simetría 180 en el punto $(0, 0, 1/4)$.*

Banda	Fonón	Frecuencia [cm^{-1}]	l_z/\hbar
1	A_2	48.439847	-4.783804 e-01
2	E_2	52.337894	4.826908 e-01
3	E_2	88.384909	1.016700 e-04
4	B_2	90.820730	-1.883001 e-01
5	E_1	129.550981	4.716688 e-01
6	E_1	143.751843	-1.773560 e-02
7	E_2	157.866052	-4.785034 e-01
8	E_2	164.110815	1.451419 e-02
9	B_1	166.955594	1.955967 e-01
10	E_1	210.893834	-2.168775 e-01
11	E_1	243.011596	1.929086 e-01
12	A_2	248.012281	2.839108 e-01
13	B_2	253.032044	-9.636329 e-02
14	E_1	253.542046	6.019579 e-02
15	E_1	288.890615	-2.684800 e-01
16	B_1	290.652522	-5.841817 e-02
17	E_2	300.179576	8.792314 e-02
18	E_2	306.556078	-3.086243 e-01
19	E_2	323.838851	4.465218 e-01
20	E_2	356.964879	4.374780 e-02
21	E_1	362.190106	7.138393 e-02
22	E_1	366.248165	-3.747741 e-01
23	B_1	367.977541	-6.032168 e-02
24	A_2	371.947156	1.143470 e-03
25	A_1	385.316523	3.144767 e-01
26	E_2	399.509086	-1.524117 e-01
27	E_2	413.812436	3.240597 e-02

Tabla 5. **Datos de frecuencia y momento angular en z para los fonones correspondientes al grupo de simetría 181 en el punto $(0, 0, 1/4)$.**

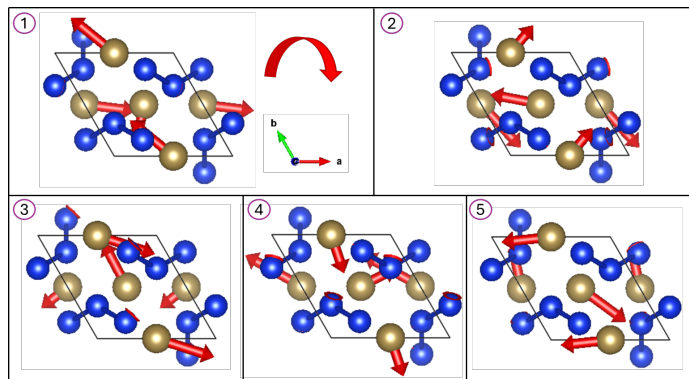
Banda	Fonón	Frecuencia [cm^{-1}]	l_z/\hbar
1	A_2	48.406837	4.784524e-01
2	E_2	52.295809	-4.826663e-01
3	E_2	88.290392	-1.103900e-04
4	B_2	90.827392	1.885017e-01
5	E_1	129.504110	-4.717726e-01
6	E_1	143.750792	1.774417e-02
7	E_2	157.853592	4.785024e-01
8	E_2	164.068294	-1.458772e-02
9	B_1	166.947382	-1.958124e-01
10	E_1	210.842561	2.169579e-01
11	E_1	242.890711	-1.927628e-01
12	A_2	247.892745	-2.838876e-01
13	B_2	252.856830	9.640745e-02
14	E_1	253.578838	-5.978913e-02
15	E_1	288.768929	2.681833e-01
16	B_1	290.219926	5.962477e-02
17	E_2	300.076572	-8.796731e-02
18	E_2	306.484239	3.091444e-01
19	E_2	323.483449	-4.452149e-01
20	E_2	356.892802	-4.471168e-02
21	E_1	362.036526	-7.125663e-02
22	E_1	366.240473	3.732416e-01
23	B_1	367.851216	6.024995e-02
24	A_2	371.698543	-1.129420e-03
25	A_1	384.933323	-3.133421e-01
26	E_2	399.543475	1.506581e-01
27	E_2	413.776411	-3.265726e-02

Figura 16. **Secuencia de desplazamiento del modo fonónico correspondiente a la banda 7 para el grupo de simetría 180, con frecuencia 157,866052 [cm⁻¹] y momento angular en z ($l_z/\hbar = -4,785034 \text{ e} - 01$).**



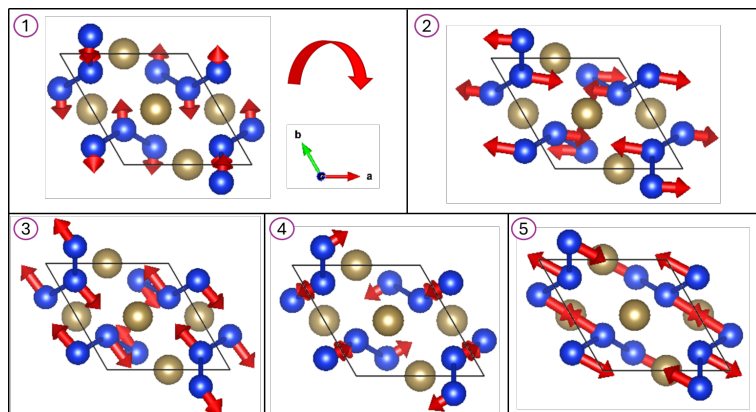
Fuente: Elaboración propia.

Figura 17. **Secuencia de desplazamiento del modo fonónico correspondiente a la banda 7 para el grupo de simetría 181, con frecuencia 157,853592 [cm⁻¹] y momento angular en z ($l_z/\hbar = 4,785024 \text{ e} - 01$).**



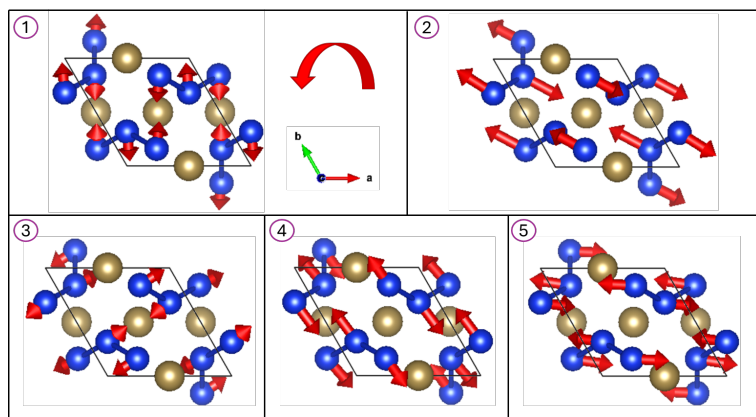
Fuente: Elaboración propia.

Figura 18. **Secuencia de desplazamiento del modo fonónico correspondiente a la banda 19 para el grupo de simetría 180, con frecuencia 323,838851 [cm⁻¹] y momento angular en z ($l_z/\hbar = 4,465218 \text{ e} - 01$).**



Fuente: Elaboración propia.

Figura 19. **Secuencia de desplazamiento del modo fonónico correspondiente a la banda 19 para el grupo de simetría 181, con frecuencia 323,483449 [cm⁻¹] y momento angular en z ($l_z/\hbar = -4,452149 \text{ e} - 01$).**



Fuente: Elaboración propia.

ANEXO B

Anexo B. Derivaciones matemáticas para la expresión del momento angular fonónico

El momento angular de los fonones, \mathbf{J}^{ph} , se define matemáticamente como:

$$\mathbf{J}^{ph} = \sum_{lj} \mathbf{u}_{lj} \times \dot{\mathbf{u}}_{lj}. \quad (18)$$

En esta expresión $\mathbf{u}_{l\alpha}$ y $\dot{\mathbf{u}}_{lj}$ son los vectores de desplazamiento de átomo j en la celda unitaria l , multiplicado por la raíz cuadrada de la masa y su derivada temporal, respectivamente.

Simplificado por componentes, a lo largo de la dirección z , obtenemos la expresión:

$$\begin{aligned} J_z^{ph} &= \sum_{lj} (u_{lj}^x \dot{u}_{lj}^y - u_{lj}^y \dot{u}_{lj}^x) \\ &= \sum_{lj} \begin{pmatrix} u_{lj}^x \\ u_{lj}^y \end{pmatrix}^T \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \dot{u}_{lj}^x \\ \dot{u}_{lj}^y \end{pmatrix}. \end{aligned} \quad (19)$$

Específicamente, para una celda unitaria con $n = 2$ átomos, el momento angular de los fonones se escribe de la siguiente forma:

$$J_z^{ph} = \sum_l \begin{pmatrix} u_{l1}^x \\ u_{l1}^y \\ u_{l2}^x \\ u_{l2}^y \end{pmatrix}^T \begin{pmatrix} 0 & 1 & & \\ & -1 & 0 & \\ & & 0 & 1 \\ & & -1 & 0 \end{pmatrix} \begin{pmatrix} \dot{u}_{l1}^x \\ \dot{u}_{l1}^y \\ \dot{u}_{l2}^x \\ \dot{u}_{l2}^y \end{pmatrix}. \quad (20)$$

En el caso del disilicato de tantalio, TaSi_2 , la anterior expresión se expande para 9 átomos en la celda primitiva (3 sitios de Ta y 6 sitios de Si). Usando $u_l = (u_{l1}^x u_{l1}^y u_{l2}^x u_{l2}^y)^T$, obtenemos:

$$J_z^{\text{ph}} = \sum_l u_l^T i M \dot{u}_l, \quad (21)$$

donde $M = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \otimes I_{n \times n}$. La expresión 21 representa el momento angular fonónico desde una perspectiva clásica, en función de las coordenadas de desplazamiento y velocidad de los átomos. Para obtener una descripción cuántica de este observable, es necesario aplicar el formalismo de segunda cuantización a las vibraciones de la red.

Para el desarrollo de este formalismo cuántico, se adoptará temporalmente la notación (k, σ) para describir un modo fonónico, donde k es el vector de onda y σ es el índice de la rama. Esta elección se alinea con la convención estándar utilizada en muchos textos teóricos de segunda cuantización y mecánica de muchos cuerpos. Es importante destacar que esta notación es conceptualmente idéntica a la notación (q, ν) utilizada en el resto del documento: k es equivalente al vector de onda q , y el índice de modo σ es equivalente al índice de banda ν .

La necesidad de este formalismo surge porque el Hamiltoniano de una red cristalina en la aproximación armónica describe un sistema de osciladores acoplados. La cuantización directa de este sistema es compleja. El procedimiento estándar, por lo tanto, consiste en transformar las coordenadas de desplazamiento atómico $u_{l\alpha}$ a un conjunto de coordenadas normales (o modos normales) $Q_{\mathbf{k},\sigma}$, donde cada modo (\mathbf{k}, σ) se comporta como un oscilador armónico cuántico independiente.

La cuantización de estos osciladores independientes introduce los operadores de creación ($a_{\mathbf{k},\sigma}^\dagger$) y aniquilación ($a_{\mathbf{k},\sigma}$). En este formalismo, el desplazamiento atómico clásico u_l es promovido a un operador cuántico \hat{u}_l , que se expresa entonces como una superposición de todos los modos normales de la red, ponderados por estos operadores cuánticos. Como se detalla en (Zhang y Niu, 2014)³, la forma explícita del operador de desplazamiento es:

$$u_l = \sum_k \epsilon_k e^{i(\mathbf{R}_l \cdot \mathbf{k} - \omega_k t)} \sqrt{\frac{\hbar}{2\omega_k N}} a_k + \epsilon_k^* e^{-i(\mathbf{R}_l \cdot \mathbf{k} - \omega_k t)} \sqrt{\frac{\hbar}{2\omega_k N}} a_k^\dagger, \quad (22)$$

donde \mathbf{R}_l es el vector de posición de la celda unitaria l , ϵ_k es el vector de polarización (eigenvector) del modo, y a_k y a_k^\dagger son los operadores de aniquilación y creación de fonones, respectivamente. A partir de esta expresión cuántica para el desplazamiento, podemos proceder con la derivación del valor de expectación del momento angular.

Derivando con respecto al tiempo, queda

$$\dot{J}_l = \sum_{k'} \left[-i\epsilon_k \omega_k e^{i(\mathbf{R}_l \cdot \mathbf{k} - \omega_k t)} a_k + i\epsilon_k^* \omega_k e^{-i(\mathbf{R}_l \cdot \mathbf{k} - \omega_k t)} a_k^\dagger \right] \sqrt{\frac{\hbar}{2\omega_k N}}. \quad (23)$$

Reemplazando en la ecuación 21, queda

$$\begin{aligned} J_z^{ph} = & \sum_l \sum_{k,k'} \frac{\hbar}{2N} \left[\epsilon_k^T M \epsilon_{k'} \sqrt{\frac{\omega_{k'}}{\omega_k}} e^{i(\mathbf{k}' - \mathbf{k}) \mathbf{R}_l} e^{i(\omega_k - \omega_{k'}) t} a_k a_{k'} \right. \\ & + \epsilon_k^T (-M) \epsilon_{k'}^* \sqrt{\frac{\omega_{k'}}{\omega_k}} e^{i(\mathbf{k}' - \mathbf{k}) \mathbf{R}_l} e^{i(\omega_k - \omega_{k'}) t} a_k a_{k'}^\dagger + \epsilon_k^\dagger M \epsilon_{k'} \sqrt{\frac{\omega_{k'}}{\omega_k}} e^{i(\mathbf{k}' - \mathbf{k}) \mathbf{R}_l} e^{i(\omega_k - \omega_{k'}) t} a_k^\dagger a_{k'} \\ & \left. + \epsilon_k^\dagger (-M) \epsilon_{k'}^* \sqrt{\frac{\omega_{k'}}{\omega_k}} e^{-i(\mathbf{k}' - \mathbf{k}) \mathbf{R}_l} e^{-i(\omega_k - \omega_{k'}) t} a_k^\dagger a_{k'}^\dagger \right]. \quad (24) \end{aligned}$$

Para simplificar la Ecuación 24, consideramos el valor de expectación en un estado de equilibrio térmico. En este contexto, los términos que contienen productos de operadores del tipo $a_k a_{k'}$ y $a_k^\dagger a_{k'}^\dagger$ no contribuyen al promedio y pueden ser descartados. Esta conclusión se fundamenta en principios básicos de la mecánica cuántica y estadística. A continuación, se detallan dos justificaciones complementarias y formalmente correctas.

1. Perspectiva del Dominio del Tiempo (Evolución de Heisenberg):

Este enfoque, que corresponde a la justificación física mencionada en la literatura de referencia, analiza la evolución temporal de los operadores. La dependencia temporal completa de cada término en la Ecuación 24 es la combinación de la del producto de operadores y la del factor exponencial explícito $e^{i(\omega_k - \omega_{k'}) t}$.

- Los términos que se descartan, $a_k a_{k'}$ y $a_k^\dagger a_{k'}^\dagger$, tienen una dependencia tempo-

ral intrínseca de $e^{-i(\omega_k+\omega_{k'})t}$ y $e^{i(\omega_k+\omega_{k'})t}$, respectivamente. Al combinarse con el factor externo, las dependencias temporales totales son $e^{-2i\omega_{k'}t}$ y $e^{2i\omega_{k'}t}$. Ambos términos oscilan rápidamente con frecuencias del orden del doble de las frecuencias fonónicas. En un estado de equilibrio, cualquier observable físico corresponde a una medición promediada en el tiempo. El promedio temporal de una función que oscila rápidamente es cero.

- Por el contrario, los términos que se conservan, $a_k^\dagger a_{k'}$ y $a_k a_{k'}^\dagger$, tienen una dependencia temporal intrínseca de $e^{i(\omega_k-\omega_{k'})t}$ y $e^{-i(\omega_k-\omega_{k'})t}$. Al combinarse con el factor exponencial explícito, la dependencia temporal total se cancela ($e^0 = 1$), resultando en términos estacionarios que no oscilan. Son estos términos estacionarios los únicos que contribuyen al promedio en el tiempo.

2. Perspectiva de la Mecánica Estadística (Formalismo de la Matriz de Densidad):

Este enfoque es más formal y no depende del tiempo. Se basa en las propiedades del estado cuántico del sistema en equilibrio térmico.

- Un sistema en equilibrio térmico se describe mediante la matriz de densidad $\rho = e^{-\beta H}/Z$, donde H es el Hamiltoniano del sistema. El Hamiltoniano de los fonones, $H = \sum_k \hbar\omega_k (a_k^\dagger a_k + 1/2)$, es diagonal en la base de estados de número de fonones (estados de Fock), denotada por $|\{n_k\}\rangle$. En consecuencia, la matriz de densidad ρ también es diagonal en esta base.
- El valor de expectación de un operador O se calcula como $\langle O \rangle = \text{Tr}(\rho O)$. Dado que ρ es diagonal, esta traza se simplifica a:

$$\langle O \rangle = \sum_{\{n\}} \langle \{n\} | \rho O | \{n\} \rangle = \sum_{\{n\}} \rho_{\{n\},\{n\}} \langle \{n\} | O | \{n\} \rangle$$

Esta expresión muestra que solo los elementos de matriz diagonales del operador O , $\langle \{n\} | O | \{n\} \rangle$, contribuyen al valor de expectación en equilibrio.

■ Analicemos los operadores de la Ecuación 24:

- Los operadores $a_k a_{k'}$ y $a_k^\dagger a_{k'}^\dagger$ son operadores que no conservan el número de fonones. Por ejemplo, $a_k a_{k'}$ actúa sobre un estado $|\{n_k\}\rangle$ y lo transforma en un estado con un número diferente de fonones. Por la ortogonalidad de la base de Fock, el producto escalar entre el estado original y el estado transformado es siempre cero. Por lo tanto, todos los elementos diagonales de estos operadores son nulos:

$$\langle \{n_k\} | a_k a_{k'} | \{n_k\} \rangle = 0 \quad \text{y} \quad \langle \{n_k\} | a_k^\dagger a_{k'}^\dagger | \{n_k\} \rangle = 0, \quad \forall \{n_k\}$$

Al no tener elementos diagonales, su valor de expectación en equilibrio es necesariamente cero.

- Por el contrario, los operadores $a_k^\dagger a_{k'}$ y $a_k a_{k'}^\dagger$ sí conservan el número de fonones. Estos operadores pueden tener elementos de matriz diagonales no nulos (por ejemplo, cuando $k = k'$, $a_k^\dagger a_k$ es el operador número, que es diagonal por definición) y, por lo tanto, su valor de expectación en equilibrio no es cero en general.

Ambas justificaciones demuestran que los términos que no conservan el número de fonones no contribuyen al momento angular promedio en equilibrio. Por lo tanto, es rigurosamente correcto descartarlos y proceder con la expresión simplificada.

$$J_z^{\text{ph}} = \frac{\hbar}{2N} \sum_l \sum_{k,k'} \left(\sqrt{\frac{\omega_k}{\omega_{k'}}} \epsilon_k^\dagger M \epsilon_{k'} a_k^\dagger a_{k'} + \sqrt{\frac{\omega_{k'}}{\omega_k}} \epsilon_{k'}^T (-M) \epsilon_k^* a_{k'} a_k^\dagger \right) e^{i(\mathbf{k}' - \mathbf{k}) \mathbf{R}_l} e^{i(\omega_k - \omega_{k'})t}. \quad (25)$$

Recordando que $\frac{1}{N} \sum_l e^{i(\mathbf{k}' - \mathbf{k}) \mathbf{R}_l} = \delta_{\mathbf{k}, \mathbf{k}'}$. Por otra parte, dado que M es antisimétrica ($M^T = -M$), se tiene $\epsilon_{k'}^T (-M) \epsilon_k^* = \epsilon_k^\dagger M \epsilon_{k'}$, entonces

$$J_z^{\text{ph}} = \frac{\hbar}{2} \sum_{k,k'} \epsilon_k^\dagger M \epsilon_{k'} \left(\sqrt{\frac{\omega_k}{\omega_{k'}}} a_k^\dagger a_{k'} + \sqrt{\frac{\omega_{k'}}{\omega_k}} a_{k'} a_k^\dagger \right) \delta_{\mathbf{k},\mathbf{k}'} e^{i(\omega_k - \omega_{k'})t}. \quad (26)$$

Tomando en cuenta la relación $[a_{\mathbf{k},\sigma'}, a_{\mathbf{k},\sigma}^\dagger] = \delta_{\sigma,\sigma'} = a_{\mathbf{k}\sigma'} a_{\mathbf{k}\sigma}^\dagger - a_{\mathbf{k}\sigma}^\dagger a_{\mathbf{k}\sigma'}$, se obtiene el momento angular de los fonones

$$J_z^{\text{ph}} = \frac{\hbar}{2} \sum_{k,k'} \epsilon_k^\dagger M \epsilon_{k'} \left(\sqrt{\frac{\omega_k}{\omega_{k'}}} + \sqrt{\frac{\omega_{k'}}{\omega_k}} \right) a_k^\dagger a_{k'} \delta_{\mathbf{k},\mathbf{k}'} e^{i(\omega_k - \omega_{k'})t} + \frac{\hbar}{2} \sum_k \epsilon_k^\dagger M \epsilon_k. \quad (27)$$

La ecuación 27 representa el operador del momento angular. Para obtener una cantidad físicamente observable, calculamos su valor de expectación en un estado de equilibrio térmico, $\langle J_z^{\text{ph}} \rangle$. En esta condición, el delta de Kronecker $\delta_{\mathbf{k},\mathbf{k}'}$ simplifica la suma sobre k' y el término exponencial se vuelve unitario, ya que $\omega_k = \omega_{k'}$. La expresión para el valor esperado se convierte en:

$$\langle J_z^{\text{ph}} \rangle = \hbar \sum_k \epsilon_k^\dagger M \epsilon_k \langle a_{\mathbf{k},\sigma'}^\dagger a_{\mathbf{k},\sigma} \rangle + \frac{\hbar}{2} \sum_k \epsilon_k^\dagger M \epsilon_k, \quad (28)$$

Aquí, $\langle a_{\mathbf{k},\sigma'}^\dagger a_{\mathbf{k},\sigma} \rangle$ es el valor esperado del operador número de fonones para el modo k . En equilibrio térmico a una temperatura T , este valor corresponde al número de ocupación promedio, dado por la distribución de Bose-Einstein, que denotaremos como $f(\omega_k)$. En equilibrio se tiene $\langle a_{\mathbf{k},\sigma'}^\dagger a_{\mathbf{k},\sigma} \rangle = f(\omega_k) \delta_{\sigma,\sigma'}$, por lo que queda

$$J_z^{\text{ph}} = \hbar \sum_k \epsilon_k^\dagger M \epsilon_k f(\omega_k) + \frac{\hbar}{2} \sum_k \epsilon_k^\dagger M \epsilon_k, \quad (29)$$

donde $l_{\mathbf{k},\sigma}^z = \left(\epsilon_{\mathbf{k},\sigma}^\dagger M \epsilon_{\mathbf{k},\sigma} \right) \hbar$, que corresponde al momento angular de un modo fonónico en particular. Al sustituir se obtiene:

$$J_z^{\text{ph}} = \sum_{\sigma,\mathbf{k}} l_{\mathbf{k},\sigma}^z \left[f(\omega_{\mathbf{k},\sigma}) + \frac{1}{2} \right]. \quad (30)$$

COMPROBACIÓN DEL MOMENTO ANGULAR CERO DE UN SISTEMA DE FONONES TRIVIAL SIN INTERACCIÓN ESPÍN-FONÓN

Para el sistema trivial de fonones $H = \frac{1}{2}p^T p + \frac{1}{2}u^T K u$, se tiene un problema de valores propios de la siguiente forma

$$D(\mathbf{k})\epsilon(\mathbf{k}, \sigma) = \omega_{\mathbf{k},\sigma}^2 \epsilon(\mathbf{k}, \sigma). \quad (31)$$

Es importante notar que $D(\mathbf{k})$ es una matriz hermítica, es decir, satisface $D^\dagger = D$. Entonces

$$\epsilon^\dagger(\mathbf{k}, \sigma)D(\mathbf{k}) = \omega_{\mathbf{k},\sigma}^2 \epsilon^\dagger(\mathbf{k}, \sigma). \quad (32)$$

De la ecuación 31, para el vector de onda $-\mathbf{k}$, se tiene

$$D(-\mathbf{k})\epsilon(-\mathbf{k}, \sigma) = \omega_{-\mathbf{k},\sigma}^2 \epsilon(-\mathbf{k}, \sigma),$$

entonces

$$\epsilon^T(-\mathbf{k}, \sigma)D^T(-\mathbf{k}) = \omega_{-\mathbf{k},\sigma}^2 \epsilon^T(-\mathbf{k}, \sigma).$$

Ahora, se utiliza la propiedad de simetría inversa en la matriz dinámica $D(\mathbf{k})$. En la mayoría de los sistemas cristalinos, existe una simetría en el espacio recíproco que conecta \mathbf{k} con $-\mathbf{k}$, de modo que $D^T(-\mathbf{k}) = D^T(\mathbf{k})$, entonces

$$\epsilon^T(-\mathbf{k}, \sigma)D(\mathbf{k}) = \omega_{-\mathbf{k},\sigma}^2 \epsilon^T(\mathbf{k}, \sigma). \quad (33)$$

De las ecuaciones 32 y 33, tenemos

$$\omega_{-\mathbf{k},\sigma} = \omega_{\mathbf{k},\sigma}, \quad (34)$$

$$\epsilon(-\mathbf{k}, \sigma) = \epsilon^*(\mathbf{k}, \sigma). \quad (35)$$

Entonces,

$$l_{-\mathbf{k},\sigma}^z = \left(\epsilon_{-\mathbf{k},\sigma}^\dagger M \epsilon_{-\mathbf{k},\sigma} \right) \hbar = \left(\epsilon_{\mathbf{k},\sigma}^T M \epsilon_{\mathbf{k},\sigma}^* \right) \hbar = \left(\epsilon_{\mathbf{k},\sigma}^\dagger M^T \epsilon_{\mathbf{k},\sigma} \right) \hbar, \quad (36)$$

dado que la matriz M , que en este caso describe el momento angular, es antisimétrica, se cumple que $M^T = -M$, entonces

$$l_{-\mathbf{k},\sigma}^z = -l_{\mathbf{k},\sigma}^z. \quad (37)$$

Esto indica que el momento angular de los fonones cambia de signo cuando se invierte el vector de onda \mathbf{k} . Finalmente, $f(\omega_{\mathbf{k},\sigma})$ es la función de ocupación para un modo de fonón a una frecuencia $\omega_{\mathbf{k},\sigma}$, y satisface la relación $f(\omega_{-\mathbf{k},\sigma}) = f(\omega_{\mathbf{k},\sigma})$. Recordando la expresión general para el momento angular fonónico total de la Ecuación 1, dado que $l_{-\mathbf{k},\sigma}^z = -l_{\mathbf{k},\sigma}^z$, los términos se cancelan entre sí al sumar sobre todos los \mathbf{k} , resultando en:

$$J_z^{\text{ph}} = 0.$$

Este resultado indica que no hay flujo neto de momento angular en el sistema de fonones sin interacción espín-fonon ³.

PSEUDO-MOMENTO ANGULAR DEL FONÓN EN EL LÍMITE CLÁSICO

La distribución de Bose-Einstein para fonones está dada por:

$$f(x) = \frac{1}{e^x - 1},$$

donde $x = \frac{\hbar\omega}{k_B T}$.

En el límite de altas temperaturas ($T \rightarrow \infty$), x es muy pequeño ($x \ll 1$), por lo que podemos expandir la función $f(x)$ en una serie de Taylor alrededor de $x = 0$:

$$f(x) = \frac{1}{x} - \frac{1}{2} + \frac{x}{12} + O(x^2), \quad (38)$$

donde $\frac{1}{x}$ es el término dominante en altas temperaturas, ya que x es pequeño.

Insertándolo en la ecuación del momento angular de los fonones en equilibrio $J_z^{\text{ph}} = \sum_{\sigma, \mathbf{k}} l_{\mathbf{k}, \sigma}^z [f(\omega_{\mathbf{k}, \sigma}) + \frac{1}{2}]$, tenemos

$$J_z^{\text{ph}}(T \rightarrow \infty) = \sum_{\sigma > 0, \mathbf{k}} \left[\left(\frac{k_B T}{\hbar\omega} + \frac{\hbar\omega}{12k_B T} \right) l_{\mathbf{k}, \sigma}^z \right]. \quad (39)$$

El término lineal a T ($\frac{k_B T}{\hbar\omega_{\mathbf{k}, \sigma}}$) es cero, esto es, $\sum_{\sigma > 0, \mathbf{k}} \frac{\epsilon_{\mathbf{k}, \sigma}^\dagger M \epsilon_{\mathbf{k}, \sigma}}{\omega_{\mathbf{k}, \sigma}} = 0$, y lo demostramos de la siguiente forma.

La relación de completitud para los modos normales de vibración en un sistema discreto se expresa como:

$$\sum_{\sigma} x_{\mathbf{k}, \sigma} \otimes \tilde{x}_{\mathbf{k}, \sigma}^T = I_{4n \times 4n}, \quad (40)$$

con $x_{\mathbf{k}} = \begin{pmatrix} \mu_{\mathbf{k}} \\ \epsilon_{\mathbf{k}} \end{pmatrix}$, $\tilde{x}_{\mathbf{k}}^T = \frac{1}{-2i\omega_{\mathbf{k}}} \left(\epsilon_{\mathbf{k}}^\dagger - \mu_{\mathbf{k}}^\dagger \right)$, donde $\mu_{\mathbf{k}, \sigma}$ representa las amplitudes asociadas a los momentos (o velocidades) de los átomos en el modo (\mathbf{k}, σ) y $\epsilon_{\mathbf{k}, \sigma}$ representa las amplitudes de desplazamiento (vectores de polarización) en el mismo modo. Así, el bloque fuera de la es:

$$\sum_{\sigma} \epsilon_{\mathbf{k}, \sigma} \otimes \tilde{\epsilon}_{\mathbf{k}, \sigma}^\dagger / (-2i\omega_{\mathbf{k}, \sigma}) = O_{2n \times 2n}, \quad (41)$$

donde $\tilde{\epsilon}_{\mathbf{k}, \sigma}^\dagger = (\tilde{x}_{\mathbf{k}, \sigma}^T)^\dagger$ es el conjugado hermítico de $\tilde{x}_{\mathbf{k}, \sigma}^T$ y $O_{2n \times 2n}$ es una matriz de ceros.

El hecho de que este bloque sea una matriz de ceros implica que las amplitudes de desplazamiento y sus duales están ortogonalmente relacionadas de tal manera que su suma sobre todos los modos resulta en cero. Entonces considerando un i y j arbitrarios, tenemos

$$\sum_{\sigma} \frac{\epsilon_i(\mathbf{k}, \sigma) \epsilon_j^*(\mathbf{k}, \sigma)}{\omega_{\mathbf{k}, \sigma}} = 0. \quad (42)$$

Comenzamos con:

$$\sum_{\mathbf{k}, \sigma < 0} \frac{\epsilon_{\mathbf{k}, \sigma}^{\dagger} M \epsilon_{\mathbf{k}, \sigma}}{\omega_{\mathbf{k}, \sigma}}. \quad (43)$$

Notamos que para cada \mathbf{k} y $\sigma < 0$, podemos escribir $\mathbf{k}' = -\mathbf{k}$ y $\sigma' = -\sigma$, donde $\sigma' > 0$. Entonces, la suma sobre $\sigma < 0$ se puede reescribir en términos de $\sigma' > 0$ de la siguiente forma

$$\sum_{\mathbf{k}, \sigma < 0} \frac{\epsilon_{\mathbf{k}, \sigma}^{\dagger} M \epsilon_{\mathbf{k}, \sigma}}{\omega_{\mathbf{k}, \sigma}} = \sum_{\mathbf{k}', \sigma' > 0} \frac{\epsilon_{-\mathbf{k}', -\sigma'}^{\dagger} M \epsilon_{-\mathbf{k}', -\sigma'}}{\omega_{-\mathbf{k}', -\sigma'}}. \quad (44)$$

Las frecuencias cumplen $\omega_{-\mathbf{k}', -\sigma'} = -\omega_{\mathbf{k}', \sigma'}$ y los vectores de polarización satisfacen $\epsilon_{-\mathbf{k}', -\sigma'} = \epsilon_{\mathbf{k}', \sigma'}^*$. Por lo tanto:

$$\sum_{\mathbf{k}', \sigma' > 0} \frac{\epsilon_{-\mathbf{k}', -\sigma'}^{\dagger} M \epsilon_{-\mathbf{k}', -\sigma'}}{\omega_{-\mathbf{k}', -\sigma'}} = \sum_{\mathbf{k}', \sigma' > 0} \frac{\left(\epsilon_{\mathbf{k}', \sigma'}^*\right)^{\dagger} M \epsilon_{\mathbf{k}', \sigma'}}{-\omega_{\mathbf{k}', \sigma'}} = \sum_{\mathbf{k}', \sigma' > 0} \frac{\epsilon_{\mathbf{k}', \sigma'}^T M \epsilon_{\mathbf{k}', \sigma'}^*}{-\omega_{\mathbf{k}', \sigma'}} = \sum_{\mathbf{k}, \sigma > 0} \frac{\epsilon_{\mathbf{k}, \sigma}^{\dagger} M \epsilon_{\mathbf{k}, \sigma}}{\omega_{\mathbf{k}, \sigma}}. \quad (45)$$

Entonces

$$\sum_{\mathbf{k}, \sigma > 0} \frac{\epsilon_{\mathbf{k}, \sigma}^{\dagger} M \epsilon_{\mathbf{k}, \sigma}}{\omega_{\mathbf{k}, \sigma}} = \frac{1}{2} \sum_{\mathbf{k}, \sigma} \frac{\epsilon_{\mathbf{k}, \sigma}^{\dagger} M \epsilon_{\mathbf{k}, \sigma}}{\omega_{\mathbf{k}, \sigma}} = \frac{1}{2} \sum_{\substack{\mathbf{k}, \sigma \\ j, i}} \frac{\epsilon_j^*(\mathbf{k}, \sigma) M_{ji} \epsilon_i(\mathbf{k}, \sigma)}{\omega_{\mathbf{k}, \sigma}} = \frac{1}{2} \sum_{\substack{\mathbf{k}, i \\ j, i}} M_{ji} \sum_{\sigma} \frac{\epsilon_j^*(\mathbf{k}, \sigma) \epsilon_i(\mathbf{k}, \sigma)}{\omega_{\mathbf{k}, \sigma}} = 0. \quad (46)$$

Por lo tanto, a altas temperaturas, el momento angular total fonónico es proporcional a $1/T$ y tiende a cero

$$J_z^{ph}(T \rightarrow \infty) = \sum_{\sigma > 0, \mathbf{k}} \frac{\hbar \omega_{\mathbf{k}, \sigma}}{12k_B T} l_{\mathbf{k}, \sigma}^z \rightarrow 0 \quad (47)$$

De modo que es posible concluir que el momento angular del fonón sólo es relevante en sistemas cuánticos de baja temperatura ³.

La validez de esta aproximación depende de la frecuencia del fonón. Se puede definir una temperatura característica para cada modo, $\Theta(\omega) = \hbar\omega/k_B$. La condición $T \gg \Theta(\omega)$ debe cumplirse. Para fonones ópticos en $TaSi_2$, las frecuencias pueden alcanzar hasta $\approx 400 \text{ cm}^{-1}$ ($\approx 12 \text{ THz}$). Esto corresponde a una energía de $\approx 50 \text{ meV}$ y una temperatura característica de $\Theta \approx 580 \text{ K}$. Por lo tanto, en este contexto, altas temperaturas se refieren a temperaturas significativamente por encima de los 580 K para poder considerar que todo el espectro fonónico se encuentra en el límite clásico. Para los modos acústicos de baja frecuencia, este límite se alcanza a temperaturas mucho más bajas.

ANEXO C

Anexo C. Análisis físico detallado de los fonones quirales en TaSi₂

Este anexo compila el análisis físico que complementa la validación del software: la clasificación por simetría de los materiales con fonones de pseudo-momento angular (PAM), y el estudio específico de las operaciones de simetría relevantes en TaSi₂ (grupos espaciales 180 y 181).

OPERACIONES DE SIMETRÍA Y CLASIFICACIÓN POR PAM

En las siguientes subsecciones vamos a discutir sobre las operaciones de simetría responsables de la aparición de fonones quirales y, adicionalmente, discutiremos las operaciones inherentes al cristal quiral SG. 180 y 181 mostrando cómo estas inducen y explican los fonones con PAM no nulo en sistemas con esta simetría, por ejemplo, TaSi₂.

Momento angular de acuerdo al rompimiento de simetría El momento angular de un modo fonónico (PAM) es una propiedad intrínseca que emerge de la dinámica de la red. Su origen se encuentra en la relación de fase entre las componentes ortogonales del desplazamiento vibracional de los átomos alrededor de sus posiciones de equilibrio. Un PAM no nulo se manifiesta cuando los átomos ejecutan movimientos colectivos de tipo circular o elíptico. La existencia de estos modos quirales está directamente ligada a la simetría del cristal; la simetría no impone una trayectoria de movimiento a los átomos, sino que determina las propiedades de transformación de los modos normales de vibración (eigenvectores fonónicos obtenidos de la diagonalización de la ecuación dinámica.), permitiendo la existencia de modos degenerados que pueden combinarse para formar estados con momento angular definido.

De acuerdo a esto, se puede realizar un análisis de simetría del momento angular del

fonón en un material ²⁵, donde se va a realizar una clasificación de materiales según las posibles combinaciones de si cumplen o no la simetría de inversión espacial (\mathcal{P}) y temporal (\mathcal{T}). La operación de inversión espacial (\mathcal{P}) transforma un fonón con momento angular l y momento lineal q en un fonón con el mismo momento angular, pero momento lineal opuesto.

$$\mathcal{P} : (l, q) \longrightarrow (l, -q). \quad (48)$$

De manera similar, la operación de simetría de inversión temporal (\mathcal{T}) transforma un fonón con momento angular l y momento lineal q en un fonón con momento angular y momento lineal opuestos.

$$\mathcal{T} : (l, q) \longrightarrow (-l, -q). \quad (49)$$

Finalmente, hay una operación de simetría de inversión temporal con inversión espacial (\mathcal{PT}). En este caso, sí y solo sí \mathcal{PT} es una simetría, este fonón tendrá un momento angular igual a cero, ya que q no cambia.

$$\mathcal{PT} : (l, q) \longrightarrow (-l, q) \quad (50)$$

A continuación (6) se agrupan materiales en una combinación de estas cinco clases de simetría y se determinan en cuál de estas clases de fonones transportan momento angular en la zona de Brillouin, lejos de un punto, línea o plano de alta simetría.

De acuerdo a esto, se permite momento angular en las clases III, IV y V. Teniendo en cuenta que el compuesto $TaSi_2$ rompe la simetría de inversión espacial, pero no la temporal, entraría en el grupo III.

²⁵ Sinisa Coh. "Classification of materials with phonon angular momentum and microscopic origin of angular momentum". En: *Physical Review B* 108.13 (2023), pág. 134307.

Tabla 6. **Clasificación de materiales según la simetría de inversión espacial (P) y temporal (T), y su capacidad para albergar fonones con momento angular neto.**

Clase	Simetrías presentes			¿Momento angular?
	\mathcal{P}	\mathcal{T}	\mathcal{PT}	
I	✓	✓	✓	No
II	×	×	✓	No
III	×	✓	×	Sí
IV	✓	×	×	Sí
V	×	×	×	Sí

Fuente: Adaptación de Sinisa Coh. “Classification of materials with phonon angular momentum and microscopic origin of angular momentum”. En: *Physical Review B* 108.13 (2023), pág. 134307.

Análisis de las operaciones de simetría del compuesto TaSi_2 A continuación se presenta un análisis detallado de las operaciones de simetría que caracterizan a los grupos espaciales 180 ($P6_222$) y 181 ($P6_422$). Es fundamental distinguir entre distintos niveles de simetría en un cristal. El grupo espacial describe el conjunto completo de operaciones que dejan invariante a toda la red periódica. El grupo puntual del cristal (en este caso, D_6) se obtiene al eliminar la parte traslacional de estas operaciones y describe la simetría macroscópica. Finalmente, la simetría de sitio (site symmetry) es el subgrupo de operaciones del grupo puntual que dejan una posición atómica específica (un sitio de Wyckoff) sin mover. La quiralidad del cristal está definida por la ausencia de operaciones impropias (inversión, planos especulares) en el grupo espacial, una condición que cumplen los grupos $P6_222$ y $P6_422$. El análisis siguiente se centrará en las operaciones del grupo espacial, ya que son estas las que definen la quiralidad global y la naturaleza de los ejes helicoidales.

Grupo Espacial 180 El grupo espacial 180 se designa $P6_222$. La notación “6₂” indica la presencia de un eje de tornillo (screw axis) de orden 6 a lo largo del eje c . La operación generadora de este eje consiste en una rotación de $\frac{360^\circ}{6} = 60^\circ$ alrededor del eje, seguida

de una traslación de $\frac{2}{6}c = \frac{1}{3}c$ a lo largo del mismo.

A continuación, se presentan las operaciones de simetría en su representación matricial-vectorial, tal como son utilizadas en las tablas cristalográficas internacionales y en software como Phonopy. Estas matrices de simetría fueron obtenidas a partir del archivo .cif generado mediante el apoyo del código FINDSYM Harold T. Stokes y Dorian M. Hatch. "FINDSYM: program for identifying the space-group symmetry of a crystal". En: *Journal of Applied Crystallography* 38.1 (2005), págs. 237-238. Es fundamental notar que estas matrices no actúan sobre un sistema de coordenadas cartesiano, sino sobre las coordenadas fraccionarias (x, y, z) expresadas en la base de los vectores de la red hexagonal. Por esta razón, la parte rotacional de la matriz puede contener componentes enteras. Por ejemplo, la operación de rotación de 60° en esta base hexagonal se representa por la transformación de coordenadas $(x, y) \rightarrow (x - y, x)$.

Las operaciones de simetría obtenidas a partir del análisis del cristal son descritas a continuación:

$$1. \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = [x, y, z]$$

Esta corresponde a la operación identidad.

$$2. \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{3} \end{bmatrix} = [x - y, x, z + \frac{1}{3}]$$

Esta se puede interpretar como una rotación de 60° alrededor del eje *c*, combinada con una traslación fraccionaria de $\frac{1}{3}$ a lo largo del eje *z*. El mapeo de (x, y) a $(x - y, x)$ es precisamente la rotación de 60° en coordenadas hexagonales

$$3. \begin{bmatrix} 0 & -1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix} = [-y, x - y, z + \frac{2}{3}]$$

Esta operación corresponde a "dos pasos" de la operación anterior, por lo que se da un corrimiento de $\frac{2}{3}$ al momento de realizar la rotación alrededor del eje c .

$$4. \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = [-x, -y, z]$$

Aquí se puede observar que no hay un corrimiento en el eje z , ya que en este caso la componente z no cambia la posición relativa dentro de la celda, sino que se traslada a la celda adyacente la cual se repite periódicamente.

$$5. \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{3} \end{bmatrix} = [-x + y, -x, z + \frac{1}{3}]$$

$$6. \begin{bmatrix} 0 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix} = [y, -x + y, z + \frac{2}{3}]$$

$$7. \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = [x - y, -y, -z]$$

$$8. \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{3} \end{bmatrix} = [x, x - y, -z + \frac{1}{3}]$$

$$9. \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix} = [y, x, -z + \frac{2}{3}]$$

$$10. \begin{bmatrix} -1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = [-x + y, y, -z]$$

$$11. \begin{bmatrix} -1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{3} \end{bmatrix} = [-x, -x + y, -z + \frac{1}{3}]$$

$$12. \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix} = [-y, -x, -z + \frac{2}{3}]$$

Estas 6 últimas operaciones (de la 7 a la 12), son combinaciones de las rotaciones anteriores con ejes binarios perpendiculares que producen un cambio de signo en z y a la vez alteran (x, y) de manera característica.

En general, es posible observar cómo las rotaciones se reflejan en los mapeos de (x, y) y las traslaciones $\frac{1}{3}$ o $\frac{2}{3}$ se añaden a z .

Grupo Espacial 181 El grupo 181, $P6_422$, es también hexagonal y quiral, con la misma clase puntual. Sin embargo, la notación “6₄” implica que la rotación de 60° se combina ahora con una traslación de $\frac{2}{3}$ a lo largo del eje c , en lugar de $\frac{1}{3}$. Por lo tanto, cuando se examinan las operaciones en el CIF, se observa que por ejemplo, La operación que antes se describía como $(x, y, z) \mapsto (x - y, x, z + \frac{1}{3})$ en el grupo 180, ahora aparece con $\frac{2}{3}$ en el desplazamiento de z .

Todas las operaciones de simetría obtenidas a partir Phonopy son descritas a continuación:

$$1. \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = [x, y, z]$$

$$2. \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix} = \left[x - y, \quad x, \quad z + \frac{2}{3} \right]$$

$$3. \begin{bmatrix} 0 & -1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{3} \end{bmatrix} = \left[-y, \quad x - y, \quad z + \frac{1}{3} \right]$$

$$4. \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \left[-x, \quad -y, \quad z \right]$$

$$5. \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix} = \left[-x + y, \quad -x, \quad z + \frac{2}{3} \right]$$

$$6. \begin{bmatrix} 0 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{3} \end{bmatrix} = \left[y, \quad -x + y, \quad z + \frac{1}{3} \right]$$

$$7. \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \left[x - y, \quad -y, \quad -z \right]$$

$$8. \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix} = \left[x, \quad x - y, \quad -z + \frac{2}{3} \right]$$

$$9. \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{3} \end{bmatrix} = \left[y, \quad x, \quad -z + \frac{1}{3} \right]$$

$$\begin{aligned}
10. \quad & \begin{bmatrix} -1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = [-x + y, \quad y, \quad -z] \\
11. \quad & \begin{bmatrix} -1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix} = [-x, \quad -x + y, \quad -z + \frac{2}{3}] \\
12. \quad & \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{3} \end{bmatrix} = [-y, \quad -x, \quad -z + \frac{1}{3}]
\end{aligned}$$

Se puede observar que, tal como se mencionó anteriormente, las 12 operaciones en 181 son análogas en estructura (seis con $z \mapsto z + \dots$, seis con $z \mapsto -z + \dots$) pero cambian las fracciones asociadas a cada rotación.

La diferencia fundamental entre las operaciones de simetría de los grupos 180 y 181 reside en la naturaleza de sus ejes de tornillo principales: 6_2 y 6_4 , respectivamente. Un eje 6_2 implica una traslación de $c/3$ por cada rotación de 60° , mientras que un eje 6_4 implica una traslación de $2c/3$ por cada rotación de 60° . Estas dos operaciones generan hélices de quiralidad (handedness) opuesta. La hélice 6_2 es levógira (left-handed), mientras que la 6_4 es dextrógira (right-handed). Esta diferencia en la quiralidad intrínseca de la estructura cristalina es la responsable directa de la inversión del signo del momento angular fonónico entre los dos enantiómeros.