

# Calculadora General de Anchos Equivalentes

Sergio Montoya Ramirez\*

Universidad de los Andes, Bogotá, Colombia.

(Dated: 9 de diciembre de 2025)

Este proyecto desarrolló una calculadora versátil para el cálculo de anchos equivalentes en espectros astronómicos, con el objetivo de simplificar la caracterización de cuerpos celestes mediante espectroscopía. Se implementó un programa gráfico con un sistema modular de clases y herencia, que incluye dos métodos principales: uno basado en límites explícitos (utilizando sumas de Riemann) para análisis rápidos, y otro de simetría (empleando el método de los trapecios) para la detección automática de múltiples picos. La herramienta se aplicó exitosamente a datos del catálogo VizierR (J/A+A/587/A65) en la región del infrarrojo cercano ( $4348\text{--}5000\text{ cm}^{-1}$ ), filtrando líneas con absorción superior al 80 %. Los anchos equivalentes obtenidos se transformaron a unidades de longitud de onda (nm) y se reportaron en picómetros, demostrando la capacidad de la calculadora para convertir espectros complejos en valores numéricos comparables. La arquitectura modular y el flujo de usuario intuitivo permiten una extensión sencilla y un análisis iterativo. Se concluye que esta herramienta representa una contribución práctica para la comunidad astronómica, agilizando el procesamiento de espectros y facilitando comparaciones sistemáticas entre observaciones.

## I. INTRODUCCIÓN

Una de las mejores formas de conocer un cuerpo celeste es mediante la espectroscopía. Con este método se puede conocer multitud de variables esenciales para la caracterización del cuerpo, como su composición o su rotación. Sin embargo, los espectros crudos resultan particularmente difíciles de procesar, pues dependen, entre otros factores, de la resolución, la tasa de medida o la intensidad de la luz recibida. Uno de los métodos utilizados en la comunidad astronómica es el cálculo de áreas equivalentes para la caracterización de espectros. El área equivalente de un espectro es, esencialmente, la cantidad de energía absorbida (o emitida, dependiendo del tipo de espectroscopía) que una cierta longitud de onda experimenta. Se le denomina \*ancho equivalente\* porque, al calcularse en un espectro normalizado, corresponde con el ancho de un rectángulo de altura uno que ocupa la misma área en el espectro. De este modo, podemos pasar de representar espectros como una línea continua con picos a representarlos como una serie de números mucho más fáciles de comparar entre sí. Este proyecto tiene como objetivo desarrollar una calculadora versátil para calcular anchos equivalentes en múltiples espectros. Para ello, se creó un programa gráfico con un sistema de clases y herencia, así como un par de métodos que permiten determinar anchos equivalentes para espectros en VizierR.

## II. MARCO TEÓRICO

Según [1], un ancho equivalente se define como el ancho que tendría un rectángulo de altura uno (por estar normalizado) y que absorbe la misma cantidad de energía que la línea atómica correspondiente. En la figura 1 puede observarse un ejemplo de ancho equivalente para un espectro idealizado. En ella se aprecia cómo el espectro representa una curva y el ancho equivalente es un rectángulo de altura 1 y menor anchura, pero que en total ocupa la misma área en el espectro; es decir, absorbe la misma energía. Así, podemos entender una de las grandes ventajas de los anchos equivalentes: podemos representar líneas atómicas como un número que indica la misma cantidad de energía absorbida por parte del elemento en esa longitud de onda concreta, simplificando considerablemente su comparación.

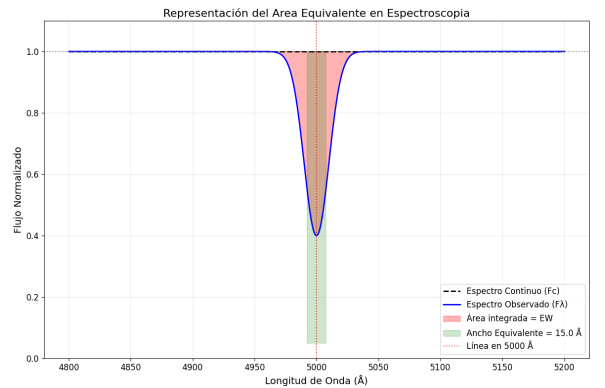


Figura 1. Representación de un ancho equivalente. Se observa un espectro idealizado con una única línea atómica, de modo que su ancho equivalente corresponde a un rectángulo que ocupa la misma área (aunque su anchura sea menor que la de la línea procesada, ya que su altura es 1).

\* Correo institucional: s.montoyar2@uniandes.edu.co

Ahora bien, los cálculos de anchos equivalentes se han simplificado considerablemente con la incorporación de computadoras en nuestra época, dada la gran cantidad de metodologías computacionalmente eficientes que existen para calcular el área bajo una curva. En nuestro caso, utilizaremos uno de los algoritmos más conocidos para el cálculo de áreas bajo una curva: el método de los trapecios. Este es una mejora respecto a un método previo conocido como sumas de Riemann. Vamos a desarrollar, entonces, las sumas de Riemann como contexto para luego explicar la mejora que representa el método de los trapecios.

Imagine que tiene una curva (dado que nosotros estamos trabajando con espectros, la tomaremos como una curva de absorción; sin embargo, este método es aplicable a cualquier curva). Esta curva puede dividirse en múltiples rectángulos que ocupan un área desde el punto de normalización hasta algún punto de la curva. Puede ver una simulación con pocos rectángulos (apenas 6) en la figura 7. Esta figura nos permite entender un poco la aproximación que estamos usando, pero también muestra una de sus mayores debilidades: la cantidad de área que no se correlaciona entre los rectángulos y la curva objetivo es alta. Puede verse en la gráfica 8 cómo esto mejora considerablemente al aumentar el número de rectángulos. Sin embargo, incluso en este caso, la correspondencia no es particularmente exacta.

Para estos casos es cuando se puede utilizar el método de los trapecios. Este consiste, en esencia, en los mismos rectángulos de antes, pero esta vez no los aproximaremos con un lado recto (particularmente el lado que limita con la curva), sino que asumiremos que la diferencia entre dos puntos consecutivos es una recta, lo que equivale a formar un pequeño triángulo sobre nuestro rectángulo, ajustándose mejor a la curva. La forma que adquiere entonces el elemento con el que se va a sumar es la de un trapecio. Puede observarse, bajo las mismas condiciones anteriores, la mejora que este método representa frente al anterior en las figuras 9 y 10. Comparar la exactitud de un método con el otro escapa por completo del alcance planeado para este documento. Sin embargo, confiamos en que el lector, con estos ejemplos, podrá darse una idea de las diferencias y ventajas de cada método.

### III. METODOLOGÍA

El objetivo de este trabajo era desarrollar una calculadora capaz de calcular áreas equivalentes para una gran diversidad de espectros. Con este fin, podemos dividir el sistema implementado en tres características principales. Primero, el flujo del usuario, es decir, la secuencia general que debe seguir un usuario para calcular los espectros. El segundo aspecto es el modelo de clases de la aplicación, que, aunque resulta increíblemente

simple, permite buena parte de su modularidad y mejora considerablemente las capacidades de la calculadora. La tercera y última categoría son los algoritmos específicos implementados para la detección de picos y para el cálculo de áreas equivalentes. Para cubrir estos tres puntos, dividiremos esta sección en tres subsecciones.

#### A. Flujo del Usuario

Para un usuario de nuestra aplicación, lo primero que deberá hacer es determinar el catálogo con el que trabajará. Inicialmente tendrá que definir el ID de VizieR para buscarlo en la plataforma. Nosotros le presentaremos las opciones posibles, de modo que pueda escoger el artículo que le interesa en una tabla. Luego de esto, deberá determinar cuál de todas las tablas es en la que le interesa trabajar, pudiendo escoger entre las distintas tablas del catálogo aquella que contenga la información relevante. Posteriormente, escogerá el modelo con el que va a trabajar y tendrá que completar la información necesaria para todos los modelos (que siempre coincide en ser la misma). Puede ver una lista de los inputs necesarios en el Apéndice VII. Con esto, aplica el método con la calculadora y verifica los resultados. Si está satisfecho, terminará y continuará con las áreas equivalentes para realizar su trabajo. En caso contrario, puede iterar volviendo a escoger el método y cambiando las variables seleccionadas. Podemos ver la representación de este flujo en la figura 2.

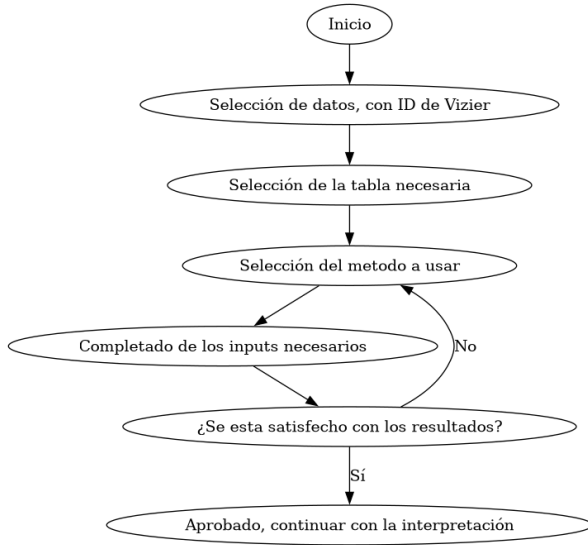


Figura 2. Representación del flujo que seguirá un usuario en nuestra aplicación. Los pasos son: selección de datos (con ID de Vizier), selección de la tabla necesaria, selección del método a usar, completado de los inputs necesarios, verificación de satisfacción con los resultados y, finalmente, aprobación y continuación con la interpretación. El flujo es esencialmente lineal, excepto en el paso de comprobación, donde, en caso de fallar, se vuelve a escoger un método y se prueba de nuevo.

## B. Modelo de Clases

A diferencia de la subsección anterior, aquí no nos interesa explicar todos los pasos ni todo el sistema de clases, sino solo los métodos y su interacción. En particular, nos interesa mostrar que todos los métodos heredan de una clase base (llamada *BaseMethod*). Esta es la clase que define el comportamiento básico. Pueden observarse los elementos básicos con los que cuenta en el Apéndice VII. De esta clase heredan todas las demás. Una característica principal de esta clase es un método abstracto llamado *\*calculate\**, que será sobrescrito por todas sus subclases para cambiar el comportamiento. La explicación detallada de los algoritmos utilizados para cada método se dará en la subsección III C. Puede verse un diagrama simplificado en la figura 3.

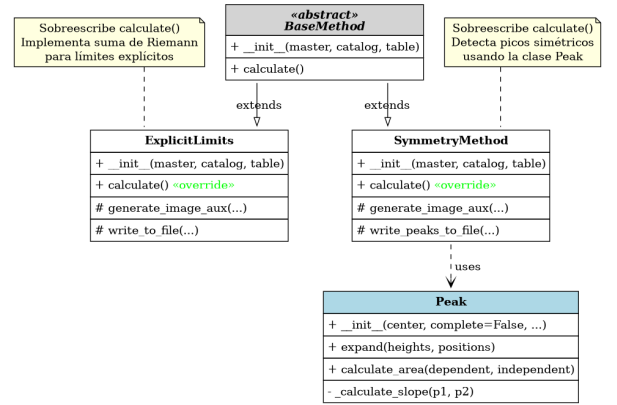


Figura 3. Diagrama que representa el sistema de clases de los métodos de cálculo de áreas equivalentes. Muestra cómo, heredando desde *BaseMethod*, se puede crear cualquier protocolo para calcular áreas equivalentes, resultando en un sistema modular y extensible.

## C. Algoritmos

En nuestro proyecto utilizamos dos algoritmos que definen el comportamiento. El primero es un algoritmo con límites explícitos, que confía plenamente en que el usuario le ha proporcionado los límites correctos para una sola línea atómica. El segundo método permite calcular, mediante un criterio de simetría, los límites de diferentes líneas atómicas y, por lo tanto, determinar distintos picos en una región mucho más extensa. A continuación, explicaremos cada uno de estos métodos.

### 1. Límites Explícitos

Como explicamos previamente, este algoritmo es relativamente sencillo y directo. No realiza preprocesamiento de los datos ni verificación. Funciona principalmente como una aproximación inicial para observar un dato rápidamente, pues es considerablemente más rápido que el otro método. Calcula el área mediante sumas de Riemann. Puede verse en la figura 4 una explicación de su flujo.

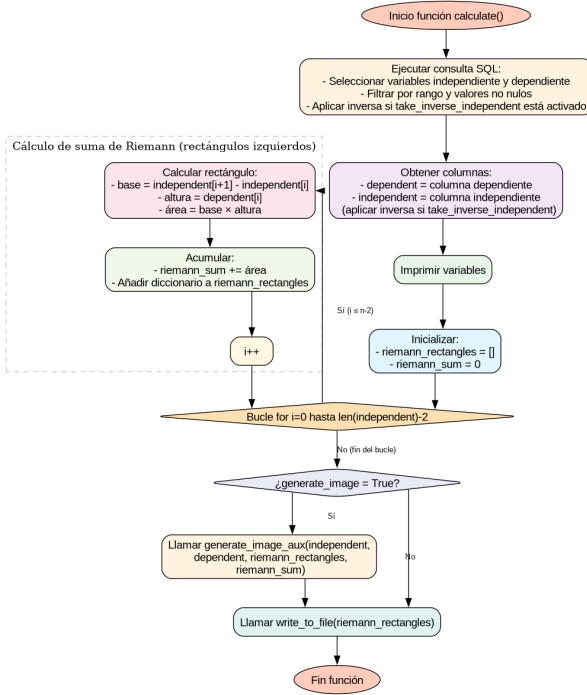


Figura 4. Diagrama de flujo de la función `*calculate()*` que implementa el cálculo numérico de integrales mediante sumas de Riemann. El proceso inicia con una consulta SQL parametrizada para extraer datos de una base de datos astronómica, aplicando transformaciones matemáticas condicionales según la configuración del usuario. El núcleo del algoritmo itera sobre los pares de datos independiente-dependiente para calcular rectángulos de aproximación usando el método del punto izquierdo, acumulando progresivamente la suma total. Finalmente, dependiendo de las opciones de configuración, genera visualizaciones gráficas y archivos de resultados.

## 2. Método de Simetría

A diferencia del caso anterior, este es un método mucho más complejo, con al menos dos fases. La primera fase consiste en descubrir aquellos picos en el rango indicado que cumplen con las condiciones. Para ello, recorre inicialmente el espectro identificando picos más altos que el umbral definido. Luego, expande los picos tomando uno a uno los puntos. Es simétrico porque se encarga de revisar que, si uno de los lados no es aproximadamente simétrico al otro, se queda con el más alto. Esto lo hace para evitar casos como líneas atómicas que se entrecruzan o ruido. Una vez verificados los picos, calcula el área bajo estos mediante el método de los trapecios. Dada la complejidad de este algoritmo, su diagrama de flujo se dividió en múltiples imágenes, que pueden verse en la sección VIII.

## IV. ANÁLISIS DE DATOS Y RESULTADOS

Con el objetivo de utilizar este proceso, se calcularán los anchos equivalentes para el espectro de [2]. Estos son datos de espectros para luz visible y cercana al infrarrojo. Particularmente, nos interesa encontrar los anchos equivalentes para datos de esta última sección del espectro, pues no están reportados como en la primera parte. Para ello, tomaremos los datos con número de onda entre  $4348$  y  $5000 \text{ cm}^{-1}$ . Además, nos quedaremos con aquellos elementos que superen el 80% de absorción. Introducimos esos datos en la calculadora y recibimos los resultados que pueden verse en la figura 5. Los resultados que devuelve son preliminares; en particular, no corresponden en muchos casos con los picos que nos interesan y presentan errores. Afortunadamente, el programa asigna un área de 0 a aquellos picos que, aunque detectados, no son calculables (por ejemplo, si es solamente un punto en 0). Además, aquí ocurre el primer problema: dado que estábamos trabajando en  $\text{cm}^{-1}$ , debemos primero convertirlo a nm (longitud de onda) y hacer lo respectivo con el ancho equivalente para expresarlo en picómetros. Con este procesamiento, se obtienen los datos que pueden encontrarse en la tabla de la sección IX. Los resultados pueden observarse en la gráfica 6.



Figura 5. Resultados del cálculo de ancho equivalente para el espectro mostrado en la gráfica. Se muestran todos los picos que serán reportados en el archivo de resultados.

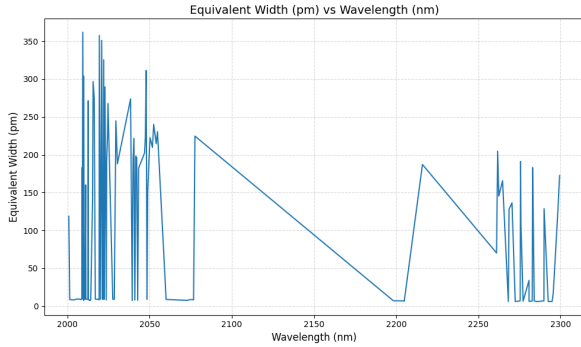


Figura 6. Gráfica de anchos equivalentes para diferentes longitudes de onda. Se representan los anchos equivalentes en picómetros respecto a la longitud de onda en nanómetros, siempre que superen una absorción del 80 %.

## V. CONCLUSIONES

Este proyecto ha desarrollado exitosamente una calculadora versátil para el cálculo de anchos equivalentes en espectros astronómicos, cumpliendo con el objetivo principal de crear una herramienta que simplifica considerablemente el procesamiento y análisis de datos espectroscópicos. La implementación de un sistema modular basado en clases con herencia ha demostrado ser particularmente efectiva, permitiendo la creación de múltiples métodos de cálculo que pueden extenderse fácilmente según las necesidades específicas del usuario.

La calculadora ha demostrado su utilidad práctica al aplicarse a datos reales del catálogo VizieR, específicamente para espectros en el infrarrojo cercano. Los resultados obtenidos, aunque requirieron un procesamiento adicional para la conversión de unidades (de  $\text{cm}^{-1}$  a nm y picómetros), muestran que el método de simetría implementado es capaz de identificar y caracterizar líneas atómicas incluso en regiones espectrales complejas. La capacidad de establecer un umbral de absorción (en este

caso del 80 %) permite filtrar los picos más relevantes, mejorando la calidad de los resultados finales.

La comparación entre los dos métodos implementados—límites explícitos y método de simetría—revela ventajas y desventajas de cada enfoque. Mientras que el método de límites explícitos resulta más rápido y adecuado para análisis preliminares, el método de simetría ofrece mayor robustez al detectar automáticamente múltiples picos y establecer sus límites mediante criterios de simetría. Esta complementariedad enriquece la herramienta, permitiendo al usuario seleccionar el enfoque más apropiado según el tipo de análisis requerido.

La arquitectura del sistema, particularmente el flujo de usuario implementado (selección de catálogo → elección de tabla → selección de método → cálculo → verificación), ha demostrado ser intuitiva y eficiente. La posibilidad de iterar en el proceso de cálculo, ajustando parámetros según los resultados obtenidos, proporciona flexibilidad que resulta valiosa en el trabajo científico real, donde la exploración de diferentes configuraciones es frecuentemente necesaria.

En cuanto a las limitaciones del proyecto, se identificó que el preprocesamiento de datos (especialmente la conversión de unidades) debe ser considerado cuidadosamente por el usuario, ya que afecta directamente la interpretación de los resultados. Futuras versiones de la calculadora podrían incorporar automatizaciones para estas conversiones, así como métodos adicionales de detección de picos y normalización de espectros.

En conclusión, esta herramienta representa una contribución práctica a la comunidad astronómica, particularmente para investigadores que trabajan con espectroscopía. Al transformar espectros complejos en valores numéricos de ancho equivalente, facilita la comparación sistemática entre diferentes observaciones y cuerpos celestes. La modularidad del diseño asegura que pueda adaptarse y extenderse para abordar necesidades específicas futuras, manteniendo su relevancia a medida que evolucionan las técnicas espectroscópicas y las bases de datos astronómicas.

- 
- [1] F. LeBlanc, *An introduction to stellar astrophysics* (Standards Information Network, 2011).
  - [2] R. A., M. N., L. U., H. J., and R. K., «Visible and near-infrared solar spectra,» .

## VI. APÉNDICE: IMÁGENES AUXILIARES

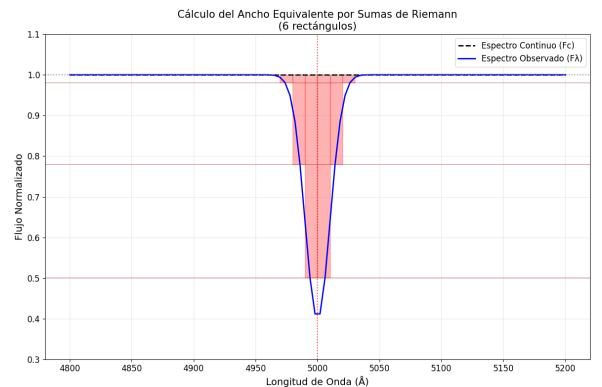


Figura 7. Representación de sumas de Riemann en el cálculo de integrales. Se observa un espectro idealizado equivalente al visto anteriormente y 6 rectángulos que aproximan el área bajo la curva. Sin embargo, la correlación no es perfecta.

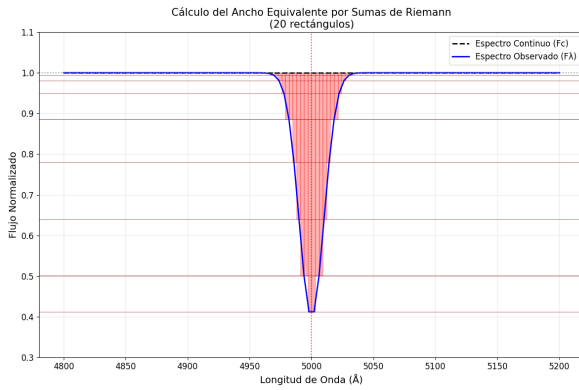


Figura 8. Representación de sumas de Riemann en el cálculo de integrales. Se observa un espectro idealizado equivalente al visto anteriormente y 20 rectángulos que aproximan el área bajo la curva. La correlación mejora respecto a la figura 7, pero aún no es perfecta.

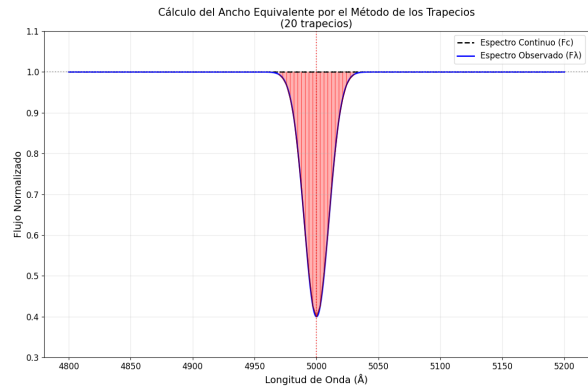


Figura 10. Representación del método de los trapecios en el cálculo de integrales. Se observa un espectro idealizado equivalente al visto anteriormente y 20 trapecios que aproximan el área bajo la curva. La aproximación mejora notablemente respecto a las sumas de Riemann.

## VII. APÉNDICE: INPUTS NECESARIOS DE PARTE DEL USUARIO

### Parámetros de Instanciación

- **master:** Ventana contenedora principal (marco padre).
- **catalog:** Objeto catálogo para acceso a la base de datos.
- **table:** Nombre de la tabla de la base de datos.

### Widgets de Entrada de Usuario

- **independent\_variables\_selector:**
  - *Tipo:* Menú desplegable (CTkOptionMenu).
  - *Función:* Selección de la variable independiente.
  - *Contenido:* Variables de la base de datos.
- **take\_inverse\_independent:**
  - *Tipo:* Casilla de verificación (CTkCheckBox).
  - *Función:* Indicar si se toma el inverso de la variable independiente. Es decir, transforma entre longitud de onda y numero de onda (con  $cm^{-1}$ )
- **dependent\_variables\_selector:**
  - *Tipo:* Menú desplegable (CTkOptionMenu).
  - *Función:* Selección de la variable dependiente.
  - *Contenido:* Variables de la base de datos.
- **begin\_entry:**

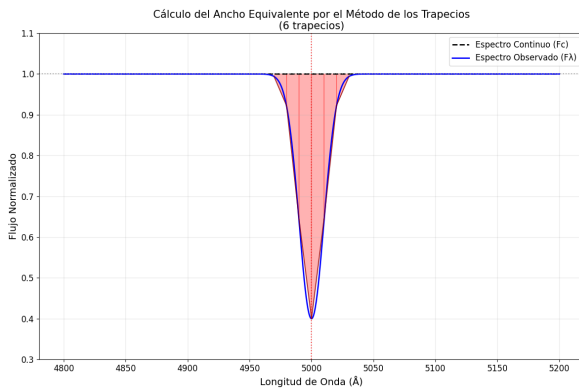


Figura 9. Representación del método de los trapecios en el cálculo de integrales. Se observa un espectro idealizado equivalente al visto anteriormente y 6 trapecios que aproximan el área bajo la curva. Comparando con las figuras 7 y 8, se aprecia una mejora considerable.

- *Tipo*: Campo de texto (CTkEntry).
- *Función*: Definir el valor inicial del rango de datos.

#### ■ end\_entry:

- *Tipo*: Campo de texto (CTkEntry).
- *Función*: Definir el valor final del rango de datos.

#### ■ threshold\_entry:

- *Tipo*: Campo de texto (CTkEntry).
- *Función*: Establecer el valor de umbral para cálculos.

#### ■ generate\_image:

- *Tipo*: Casilla de verificación (CTkCheckBox).
- *Función*: Activar/desactivar generación de imagen gráfica.

#### Botón de Acción

#### ■ calculate\_buttons:

- *Tipo*: Botón (CTkButton).
- *Función*: Ejecutar el método `calculate()`.

## VIII. APÉNDICE: DIAGRAMAS DEL MÉTODO DE SIMETRÍA

### A. Diagrama Principal del Método `calculate()`

Diagrama de Flujo - Método `calculate()` de `SymmetryMethod`

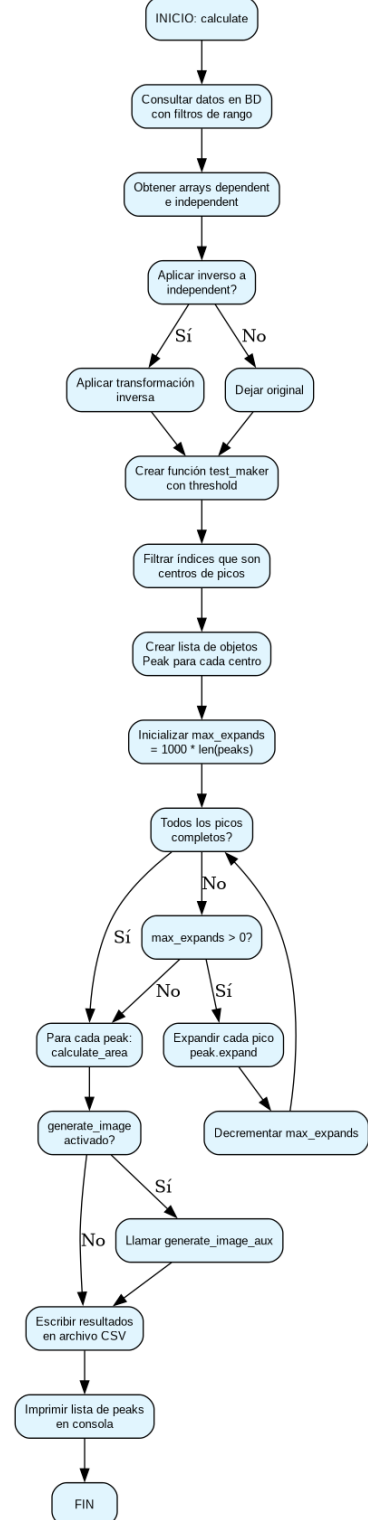


Figura 11. Diagrama de flujo principal del método `calculate()` de la clase `SymmetryMethod`. Muestra el flujo completo desde la consulta de datos hasta la generación de resultados.



La Figura 11 muestra el flujo principal del método `calculate()`, que incluye:

- Consulta de datos desde la base de datos con filtros aplicados.
- Transformación opcional de variables independientes.
- Detección de centros de picos usando la función `test_maker`.
- Expansión iterativa de picos hasta completarlos.
- Cálculo de áreas de cada pico.
- Generación de imágenes y archivos de resultados.

## B. Método `expand()` de la Clase `Peak`

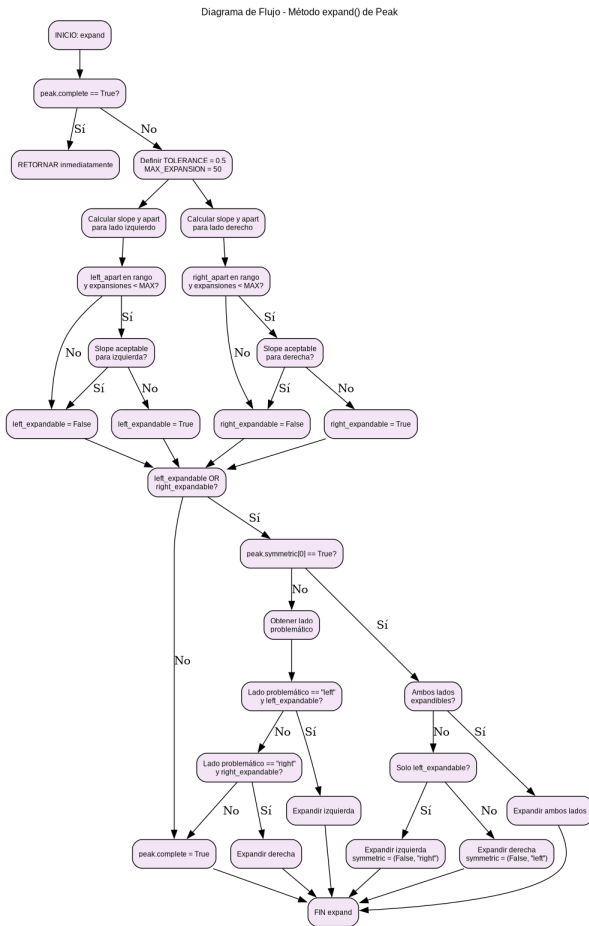


Figura 12. Diagrama de flujo del método `expand()` de la clase `Peak`. Muestra la lógica de expansión de picos basada en pendientes y simetría.

La Figura 12 ilustra el algoritmo de expansión de picos, que:

- Verifica si un pico ya está completo.
- Calcula pendientes en ambos lados del pico.
- Determina si las pendientes son aceptables según la tolerancia.
- Expande el pico en la dirección apropiada.
- Maneja casos simétricos y asimétricos.
- Establece límites de expansión máxima.

## C. Método `calculate_area()` de la Clase `Peak`

Diagrama de Flujo - Método `calculate_area()` de `Peak`



Figura 13. Diagrama de flujo del método `calculate_area()` de la clase `Peak`. Muestra cómo se calcula el área bajo la curva para picos simétricos y asimétricos.

La Figura 13 detalla el cálculo de áreas, que incluye:

- Verificación de simetría del pico.
- Cálculo de áreas para picos simétricos (suma de ambos lados).
- Cálculo de áreas para picos asimétricos (duplicación del lado válido).
- Manejo de índices de puntos adyacentes.



#### D. Función calculate\_segment\_area()

Diagrama de Flujo - Función calculate\_segment\_area()

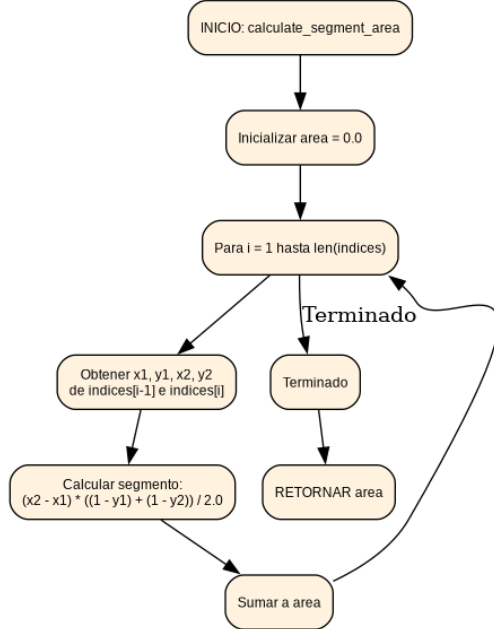


Figura 14. Diagrama de flujo de la función interna calculate\_segment\_area(). Muestra el cálculo trapezoidal del área entre puntos consecutivos.

La Figura 14 muestra el cálculo trapezoidal:

- Inicialización del área acumulada.
- Iteración sobre pares consecutivos de puntos.
- Cálculo del área trapezoidal para cada segmento.
- Acumulación del área total.

#### E. Flujo General de Detección de Picos

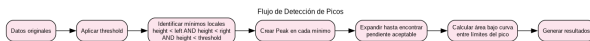


Figura 15. Flujo general de detección de picos. Resume el proceso completo desde datos originales hasta la generación de resultados.

La Figura 15 resume el proceso completo:

- Aplicación del umbral a los datos originales.
- Identificación de mínimos locales.
- Creación de objetos Peak.

- Expansión de límites de picos.
- Cálculo de áreas.
- Generación de resultados finales.

#### IX. APÉNDICE: RESULTADOS

Wave	Area	Wavelength_nm	Area_transformed
4348.944	0.091	2299.409	172.555
4351.05	0.064	2298.296	122.032
4356.219	0.009	2295.569	17.329
4356.227	0.009	2295.565	17.194
4357.515	0.003	2294.886	6.166
4362.072	0.003	2292.488	6.095
4366.833	0.068	2289.989	128.753
4367.005	0.004	2289.899	6.919
4375.621	0.003	2285.39	5.956
4375.715	0.003	2285.341	6.008
4378.299	0.004	2283.992	6.981
4380.148	0.095	2283.028	183.093
4380.329	0.068	2282.934	131.291
4380.657	0.003	2282.763	6.563
4384.35	0.003	2280.84	6.295
4384.355	0.003	2280.837	6.294
4384.374	0.004	2280.827	6.737
4384.385	0.018	2280.821	34.011
4391.316	0.003	2277.222	6.511
4393.855	0.056	2275.906	108.532
4394.314	0.099	2275.668	191.034
4394.749	0.004	2275.443	6.951
4400.576	0.003	2272.43	6.033
4404.408	0.07	2270.453	136.519
4407.887	0.066	2268.661	127.887
4408.723	0.003	2268.231	6.083
4415.55	0.085	2264.724	165.616
4420.015	0.074	2262.436	145.446
4421.403	0.105	2261.726	204.785
4422.684	0.036	2261.071	70.151
4512.685	0.092	2215.976	187.04
4535.344	0.003	2204.904	6.746
4535.363	0.003	2204.895	6.985
4535.373	0.003	2204.89	6.844
4549.032	0.003	2198.27	7.024
4813.396	0.097	2077.535	224.594
4815.343	0.004	2076.695	8.261
4817.27	0.004	2075.865	8.532
4819.178	0.004	2075.043	8.7
4824.679	0.003	2072.677	7.571
4854.406	0.004	2059.984	8.794
4866.796	0.097	2054.74	230.494
4868.66	0.091	2053.953	214.607
4872.31	0.101	2052.414	240.059
4874.094	0.088	2051.663	209.699
4877.58	0.093	2050.197	222.412
4879.284	0.08	2049.481	190.196
4880.958	0.063	2048.778	150.041

Continúa en la siguiente página

Wave	Area	Wavelength_nm	Area_transformed
4881.97	0.004	2048.354	8.926
4883.152	0.131	2047.858	311.345
4884.314	0.094	2047.37	224.033
4885.455	0.085	2046.892	202.49
4894.111	0.076	2043.272	182.312
4895.529	0.003	2042.68	7.807
4896.921	0.082	2042.1	195.826
4898.284	0.083	2041.531	198.191
4899.621	0.003	2040.974	7.998
4900.93	0.092	2040.429	221.61
4902.211	0.081	2039.896	194.626
4903.465	0.003	2039.374	7.378
4905.887	0.114	2038.367	273.695
4925.23	0.078	2030.362	188.112
4927.554	0.101	2029.404	244.725
4929.852	0.004	2028.459	8.912
4932.119	0.004	2027.526	9.107
4939.202	0.11	2024.619	267.598
4939.807	0.094	2024.371	229.901
4941.121	0.079	2023.832	192.289
4941.763	0.003	2023.569	8.215
4943.694	0.118	2022.779	289.542
4944.886	0.004	2022.291	8.904
4945.594	0.133	2022.002	325.479
4946.731	0.004	2021.537	8.847
4947.466	0.004	2021.237	9.013
4948.552	0.143	2020.793	350.961
4950.349	0.004	2020.06	9.023
4952.119	0.146	2019.337	357.551
4952.436	0.003	2019.208	8.188
4953.867	0.004	2018.625	9.111
4954.669	0.004	2018.298	9.118
4956.398	0.004	2017.594	9.022
4958.099	0.004	2016.902	9.112
4959.771	0.112	2016.222	275.327
4961.413	0.12	2015.555	296.613

4962.225	0.058	2015.225	142.115
4964.92	0.003	2014.131	8.402
4965.366	0.003	2013.95	7.811
4965.385	0.003	2013.942	7.724
4966.875	0.003	2013.338	8.077
4966.933	0.003	2013.315	7.631
4968.444	0.004	2012.703	8.778
4968.801	0.11	2012.558	271.33
4969.21	0.004	2012.392	8.849
4970.694	0.004	2011.791	9.126
4972.152	0.004	2011.202	9.054
4972.574	0.065	2011.031	160.088
4973.583	0.004	2010.623	9.15
4974.201	0.004	2010.373	9.187
4974.992	0.004	2010.054	9.235
4975.566	0.123	2009.822	304.034
4975.658	0.003	2009.784	7.465
4976.903	0.146	2009.282	361.86
4977.73	0.004	2008.948	9.151
4978.029	0.074	2008.827	183.06
4978.21	0.004	2008.754	9.162
4979.061	0.004	2008.411	9.158

Continúa en la siguiente página

Wave	Area	Wavelength_nm	Area_transformed
4979.489	0.004	2008.238	9.138
4979.794	0.003	2008.115	8.48
4980.739	0.004	2007.734	9.135
4981.96	0.004	2007.242	9.233
4986.521	0.004	2005.406	9.193
4988.195	0.004	2004.733	8.729
4990.672	0.003	2003.738	8.171
4996.625	0.003	2001.351	8.626
4998.065	0.047	2000.774	118.645