

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE  
TELECOMUNICACIÓN



**UNIVERSITAS**  
*Miguel Hernández*

*”Diseño de Herramienta para el análisis de  
imágenes hiperespectrales”*

TRABAJO FIN DE GRADO

Septiembre - 2024

AUTOR: Antonio Sellar López

DIRECTOR/ES: Otoniel Mario López Granado

Miguel Onofre Martínez Rach





*Lo esencial es invisible a los ojos*

*~Antoine de Saint-Exupéry~*

## **AGRADECIMIENTOS**

---

Agradecer a la Escuela Politécnica Superior de Elche (EPSE) y a mis tutores del TFG Otoniel y Miguel por brindarme la oportunidad de trabajar junto a ellos en este tema y fomentar mi curiosidad y habilidad a la vez que descubría nuevas tecnologías.

Quiero agradecer a mi familia, a mis padres y a mi hermana por ser uno de los pilares fundamentales de mi vida, y por todo el apoyo que me dan cada día dando el 200% de lo que tienen. Sin ellos no habría llegado tan lejos.

Quiero agradecer a todos los compañeros y amigos que he hecho durante el grado, así como los que ya me acompañaban por todo el afecto y fuerzas que me dan día a día.

Y por último y no menos importante quiero agradecerle a mi novia Belén, por todas las veces que me ha dado alas nuevas cuando las necesitaba para seguir adelante.



## **RESUMEN**

---

El presente Trabajo Fin de Grado aborda el desarrollo de una herramienta avanzada para la gestión y análisis de datos hiperespectrales, con el objetivo e ilusión personal de investigar en la mejora de la eficiencia y precisión en la identificación y clasificación de materiales en diversas aplicaciones industriales y medioambientales. En un contexto donde la introducción de nuevas tecnologías se vuelve imperativa para optimizar la producción y cumplir con los objetivos globales de bienestar y cuidado del medio ambiente, esta herramienta se presenta como una solución integral y versátil.

La herramienta desarrollada permite la carga, visualización y análisis de archivos binarios generados por cámaras hiperespectrales. Entre sus funcionalidades más destacadas se incluyen la definición y gestión de regiones de interés (ROIs), la visualización de espectros específicos, y la capacidad de comparar diferentes regiones dentro de las imágenes capturadas. Además, se ha implementado una estructura de clases robusta que garantiza la modularidad y escalabilidad del software, facilitando futuras mejoras y ampliaciones.

En conclusión, el desarrollo de esta herramienta hiperespectral representa un avance significativo en la integración de tecnologías avanzadas en la industria, promoviendo la eficiencia operativa y apoyando los objetivos medioambientales. Su aplicación no solo mejora la productividad y precisión en la clasificación de materiales, sino que también refuerza el compromiso con el bienestar y la preservación del medio ambiente, alineándose con las metas globales de sostenibilidad.

# Índice de contenido

1	OBJETIVOS DEL PROYECTO .....	9
1.1	OBJETIVO PRINCIPAL: .....	9
1.2	HITOS DEL DESARROLLO:.....	9
2	INTRODUCCIÓN.....	12
2.1	INTRODUCCIÓN E IMPORTANCIA .....	12
2.2	MOTIVACIÓN DEL PROYECTO.....	12
2.3	TECNOLOGÍA HIPERESPECTRAL.....	13
2.3.1	FUNCIONAMIENTO DE LA TECNOLOGÍA HIPERESPECTRAL ....	13
2.4	CLASIFICACIÓN DE PLÁSTICOS Y ALGORITMOS (ESTADO DEL ARTE) 14	
3	HERRAMIENTAS UTILIZADAS .....	16
3.1	(CÁMARA) SPECIM FX17 (900-1700nm).....	16
3.2	VISUAL STUDIO .....	18
3.3	LENGUAJE DE PROGRAMACIÓN (C++) .....	18
3.4	OPEN CV .....	19
3.5	QT .....	20
3.6	OTRAS HERRAMIENTAS UTILIZADAS.....	20
3.6.1	GITHUB .....	20
3.6.2	DOXYGEN .....	21
3.6.3	ADOBE PHOTOSHOP CS5 .....	21
4	DESARROLLO DEL SOFTWARE .....	22
4.1	FUNCIONAMIENTO DE LA HERRAMIENTA .....	22
4.2	ESTRUCTURA DE LAS CLASES.....	28
4.2.1	CLASE HIPERCUBO (PRIMER NIVEL) .....	29
4.2.2	CLASE HCDATA (SEGUNDO NIVEL).....	33
4.2.3	CLASE ROI (SEGUNDO NIVEL).....	36
4.2.4	CLASE MATERIAL (SEGUNDO NIVEL).....	40

5	CONCLUSIONES.....	42
6	BIBLIOGRAFÍA CONSULTADA.....	43
	ANEXO I: HAT (MANUAL DE USUARIO).....	46
	PRIMEROS PASOS .....	47
	INTERFAZ DE USUARIO Y ESPACIO DE TRABAJO.....	47
	BARRA DE MENÚ .....	47
	BARRA DE HERRAMIENTAS .....	48
	INTERFAZ DE VISUALIZACIÓN DE ESPECTROS .....	50
	BARRA DE ESTILOS DE GRÁFICAS .....	50
	INTERFAZ DE VISUALIZACIÓN DE ROIS .....	53
	ROI VIEW MODO: ASOCIACIÓN .....	53



## Índice de Figuras:

Figura 1. Interfaz base de HAT.....	22
Figura 2. Visualización del hipercubo cargado.....	23
Figura 3. Asociación de ROIs y materiales .....	23
Figura 4. Adición de material .....	24
Figura 5. Vista de la gráfica en Raw .....	25
Figura 6. Vista de la gráfica de medias.....	26
Figura 7. Vista de la gráfica de medias en comparativa .....	26
Figura 8. Vista de la visualización de la curva del puntero .....	27
Figura 9. Diagrama UML de las clases del programa .....	28
Figura 10. Diagrama de jerarquía de la clase Hipercubo .....	29
Figura 11. Diagrama de jerarquía de la clase HcData .....	33
Figura 12. Diagrama de jerarquía de la clase ROI .....	36
Figura 13. Logotipo e icono de la herramienta HAT.....	46

## Índice de Tablas

Tabla 1. Especificaciones de la cámara Specim FX17	16
--	----

## Índice de Ilustraciones

1. Cámara Specim FX17.....	16
2. Logo Visual Studio .....	18
3. Logo C++.....	18
4. Logo OpenCV .....	19
5. Logo Qt.....	20
6. Logo GitHub.....	20
7. Logo doxygen.....	21
8. Logo Adobe Photoshop CS5 .....	21



# 1 OBJETIVOS DEL PROYECTO

---

## 1.1 OBJETIVO PRINCIPAL:

Diseño de una herramienta integral para la carga, análisis y visualización de archivos binarios de cámaras hiperspectrales, con aplicación industrial para la comparación y clasificación de espectros de materiales.

El objetivo principal de este trabajo es desarrollar una herramienta avanzada y versátil que permita la gestión completa de archivos hiperspectrales generados por cámaras especializadas. Esta herramienta está diseñada no solo para la investigación académica, sino también para su aplicación en entornos industriales, donde se requiere una precisa identificación y clasificación de materiales en base a sus firmas espectrales.

## 1.2 HITOS DEL DESARROLLO:

- **Carga del archivo binario del hipercubo:**

Desarrollar una funcionalidad que permita la carga de archivos binarios generados por cámaras hiperspectrales.

Garantizar que el archivo se cargue correctamente y se interprete como un hipercubo de datos.

- **Visualización de la imagen capturada en pantalla:**

Implementar una interfaz gráfica que permita visualizar la imagen capturada por la cámara hiperspectral en tiempo real.

- **Definición de ROIs (regiones de interés) para su análisis:**

Proporcionar herramientas que permitan al usuario seleccionar y definir regiones específicas de la imagen para un análisis detallado.

- **Creación de gráficas que muestren los espectros de las regiones seleccionadas:**

- **Gráfica de todas las curvas del ROI seleccionado:**

Desarrollar una gráfica que muestre todas las curvas espectrales de los píxeles dentro de una región de interés seleccionada.

- **Gráfica de un porcentaje de las curvas del roi seleccionado:**

Implementar la capacidad de mostrar un subconjunto representativo de las curvas espectrales del ROI.

- **Gráfica de una única curva que represente la media de todas las curvas:**

Crear una gráfica que muestre la curva espectral promedio de todas las curvas dentro del ROI seleccionado.

- **Visualización de las coordenadas de los píxeles incluidos en el roi seleccionado:**

Añadir una funcionalidad que permita visualizar las coordenadas exactas de los píxeles dentro del ROI.

- **Comparación de dos rois en cualquiera de las gráficas mencionadas:**

Desarrollar herramientas que permitan comparar dos regiones de interés (ROIs) utilizando cualquiera de las gráficas espectrales mencionadas anteriormente.

- **Herramienta “puntero” para ver la curva del píxel seleccionado:**

Implementar una herramienta de puntero que permita al usuario seleccionar un píxel específico y ver su curva espectral individual.

- **Añadir materiales a una lista con ID, nombre y color relacionado con el material:**

Crear una lista donde se puedan asociar diferentes materiales con un identificador único, un nombre y un color específico para cada material.

- **Asociación de material a un ROI:**

Permitir que los usuarios puedan asociar un material específico a una región de interés seleccionada.

- **Visualización de rois con el color del material asociado y una etiqueta con el ID del material:**

Asegurar que los ROIs definidos se visualicen con el color del material asociado y que muestren una etiqueta con el ID correspondiente del material.

- **Guardar los datos de la sesión en un archivo .json para su posterior carga:**

Implementar la opción de guardar todos los datos y configuraciones de la sesión actual en un archivo con formato "json".

Incluir la funcionalidad de cargar estos archivos con extensión "json" en futuras sesiones, permitiendo la continuación del análisis desde donde se dejó.



## 2 INTRODUCCIÓN

---

### 2.1 INTRODUCCIÓN E IMPORTANCIA

Actualmente, las empresas de producción de todas las índoles, así como las dedicadas a la clasificación de residuos para su reciclaje sostenible, están experimentando una transformación significativa impulsada por la implementación de tecnologías cada vez más avanzadas. Este avance tecnológico no solo optimiza los procesos operativos, sino que también juega un papel crucial en la mitigación del impacto ambiental.

En las empresas de producción, la adopción de tecnologías compatibles con la automatización basada en la interpretación de datos tal como la inteligencia artificial (IA) permite mejorar la eficiencia energética, reducir el desperdicio de material útil y optimizar la cadena de suministro.

Por otro lado, las empresas dedicadas a la clasificación y reciclaje de residuos, con la incorporación de tecnologías avanzadas como sensores ópticos, robótica, algoritmos basados en redes neuronales y **cámaras hiperespectrales**, logran alcanzar hitos en eficiencia y precisión, incrementando las tasas de recuperación y reduciendo la cantidad de residuos que terminan en vertederos acentuando la contaminación.

### 2.2 MOTIVACIÓN DEL PROYECTO

En el momento de la elección de mi Trabajo de fin de Grado, nos encontrábamos al comienzo del auge de la inteligencia artificial. Si bien mi intención era investigar a fondo la programación de redes neuronales, se me ofreció la oportunidad de crear una herramienta capaz de generar datos útiles para ser utilizados en entrenamientos de IA.

Trabajar en este proyecto me ha permitido darme cuenta de lo que realmente es una inteligencia artificial, la importancia de generar un gran banco de datos para entrenarla y me ha propiciado más experiencia y conocimientos en programación más allá de lo estudiado en la carrera.

Conociendo el problema al que nos enfrentamos con la contaminación en el planeta, desarrollar la herramienta que he creado ha sido una experiencia de lo más interesante y gratificante. Me gustaría pensar que mi pequeña investigación puede ser uno de los pasos que demos hacia un futuro donde se desperdicie menos y supongamos una amenaza menor para el medio ambiente y el lugar donde vivimos.

## **2.3 TECNOLOGÍA HIPERESPECTRAL**

La tecnología hiperespectral es una técnica avanzada de teledetección que captura y procesa información a lo largo de un amplio rango del espectro electromagnético. Esta tecnología permite obtener un hipercubo, donde cada píxel de la imagen contiene un espectro detallado, proporcionando información sobre la composición y características del material observado. Esta capacidad permite identificar objetos y materiales al analizar sus firmas espectrales únicas, por lo que se puede aplicar a una amplia variedad de campos, desde la producción alimentaria hasta el monitoreo ambiental y la clasificación de residuos para su reciclaje.

Las imágenes recogidas por las cámaras hiperespectrales reciben el nombre de imágenes hiperespectrales, las cuales, son una forma de representación de objetos utilizando múltiples bandas estrechas a lo largo de todo el espectro, que van desde el ultravioleta hasta el infrarrojo. A diferencia de las imágenes multispectrales, que capturan bandas discretas y separadas, las imágenes hiperespectrales generalmente abarcan un mayor número de bandas de longitudes de onda. Esto nos permite obtener un 'cubo de datos', al cual llamaremos hipercubo, en el que cada píxel contiene información espectral detallada. Cabe destacar que, aunque las imágenes hiperespectrales suelen cubrir un rango continuo de longitudes de onda, no es obligatorio que este rango sea completamente continuo y contiguo. La resolución espectral de cada banda determina la calidad de la cámara hiperespectral. Tanto las imágenes multispectrales como las hiperespectrales pueden formar hipercubos, aunque la principal diferencia radica en la cantidad de bandas medidas: decenas en las multispectrales y centenas en las hiperespectrales.

### **2.3.1 FUNCIONAMIENTO DE LA TECNOLOGÍA HIPERESPECTRAL**

El proceso de captura de imágenes hiperespectrales implica el uso de una cámara hiperespectral, que incluye un espectrómetro como uno de sus componentes. Este dispositivo captura la luz de una escena y la descompone en sus longitudes de onda individuales, creando imágenes que contienen información espectral para cada píxel. Es importante señalar que, aunque generalmente hablamos de píxeles en el contexto de cámaras, también existen espectrómetros de punto (un solo píxel) y de línea, que proporcionan medidas espectrales para cada 'píxel' en una línea de la escena observada.

Esta información puede ser comparada con firmas espectrales conocidas para identificar y cuantificar materiales en la escena.

Este proceso puede categorizarse en los siguientes puntos:

1. Captura de la luz: La cámara hiperespectral recoge la luz que se refleja o emite desde el objeto de interés a través de una lente espacial.
2. Dispersión de la luz: La luz capturada pasa a través de un elemento dispersivo, como un prisma o una rejilla de difracción, que separa la luz en sus componentes de longitud de onda.
3. Selección de longitudes de onda: Mediante filtros o ranuras ajustables, el espectrómetro permite seleccionar las longitudes de onda específicas para el análisis. Esto permite enfocarse en bandas de interés particular dependiendo de la aplicación.
4. Detección de la luz: La luz dispersada llega a un detector, como un dispositivo de carga acoplada (CCD) o un fotodiodo, que mide la intensidad de la luz en cada longitud de onda específica.
5. Adquisición y análisis de datos: El detector convierte la intensidad de la luz en señales eléctricas, que son procesadas y registradas. Estos datos se representan como un “hipercubo”, una matriz tridimensional donde dos dimensiones representan el espacio y la tercera, representa las longitudes de onda. Este hipercubo contiene una cantidad enorme de datos que pueden ser analizados para identificar características específicas del objeto estudiado.
6. Calibración: Para asegurar la precisión, las cámaras hiperespectrales pasan por procedimientos de calibración comparando sus respuestas con estándares de referencia conocidos. Esto corrige cualquier sesgo o variación inherente del sensor.

## **2.4 CLASIFICACIÓN DE PLÁSTICOS Y ALGORITMOS (ESTADO DEL ARTE)**

En cuanto a la clasificación de plásticos se han desarrollado varios métodos y algoritmos sofisticados para mejorar la eficiencia y precisión en el ámbito.

Los algoritmos y tecnologías más recientes son:

- Redes Neuronales Convolucionales (CNNs):

Las CNNs se utilizan ampliamente para el reconocimiento y clasificación de imágenes debido a su capacidad para aprender características complejas. En el contexto de la clasificación de plásticos, se emplean para identificar y categorizar diferentes tipos, basándose en sus propiedades espectrales y visuales. Por ejemplo, un estudio reciente mostró que las CNNs pueden alcanzar una alta precisión en la clasificación de plásticos, superando a métodos tradicionales como *k-Nearest Neighbors (kNN)* y *Support Vector Machines (SVM)*. [14]

- Modelos Generativos Multimodales:

Los modelos generativos, como los Modelos Generativos Adversarios (GANs), se utilizan para mejorar la clasificación de plásticos al integrar datos de múltiples sensores. Estos modelos pueden combinar información visual y espectral para identificar con mayor precisión los diferentes tipos de plásticos, incluso en entornos complejos. [15]

- Detección y Rastreo con Transformadores:

La aplicación de modelos de detección basados en transformadores, como DETR, ha mostrado su eficacia en la clasificación de residuos plásticos. Estos modelos son capaces de procesar y analizar grandes volúmenes de datos de imágenes, mejorando la detección y clasificación de plásticos en tiempo real. [16]

- Algoritmos de aprendizaje profundo:

Los algoritmos de aprendizaje profundo, como YOLOv4 y YOLOv8, se utilizan para la detección y clasificación de residuos plásticos en tiempo real. YOLOv4 y YOLOv8 son modelos de detección de objetos que no utilizan información espectral, sino que se basan en imágenes RGB convencionales. Sin embargo, existen otros algoritmos de aprendizaje profundo diseñados específicamente para trabajar con datos hiperespectrales, como el *Convolutional Neural Network (CNN)* aplicado a imágenes hiperespectrales y el *Support Vector Machine (SVM)* para clasificación espectral. Estos algoritmos pueden identificar y categorizar diferentes tipos de plásticos con alta precisión y eficiencia, mejorando significativamente los procesos de reciclaje automatizado. [17]

### 3 HERRAMIENTAS UTILIZADAS

A continuación, se hace un repaso por los programas empleados, así como los datos de la cámara utilizada para la captura del hipercubo utilizado en la realización del proyecto, detallando las funcionalidades principales de cada uno y el motivo de su elección.

#### 3.1 (CÁMARA) SPECIM FX17 (900-1700nm)

Para la captura del hipercubo utilizado para el desarrollo del proyecto se ha utilizado la cámara hiperespectral Specim FX17, la cual opera en el rango de longitudes de onda de 900 – 1700nm. Destaca por su alta resolución espectral y velocidad de adquisición, lo que la hace ideal para entornos de producción a en tiempo real y clasificación de materiales.



1. Cámara Specim FX17

#### Especificaciones técnicas de la cámara hiperespectral Specim FX17:

Tabla 1. Especificaciones de la cámara Specim FX17

Cámara Hiperespectral		FX17
<i>Rango espectral</i>	Rango de longitudes de onda que maneja la cámara.	900-1700nm
<i>Bandas espectrales</i>	Las distintas longitudes de onda en las que la cámara puede capturar información.	224
<i>FWHM espectral</i>	<i>Full Width at Half Maximum</i> . Es una medida de la resolución espectral de la cámara. Capacidad para distinguir entre diferentes longitudes de onda cercanas.	8nm



<i>Muestreo espacial</i>	Resolución espacial de la cámara. Captura líneas de 640 píxeles de alto.	640 píxeles
<i>Cuadros por segundo</i>	Velocidad de captura de la cámara.	Fotograma completo de 670 FPS, 15000 FPS con 4 bandas seleccionadas
<i>Campo de visión</i>	Área visible a través de la lente en una instancia determinada.	38°
<i>Número F</i>	Cantidad de luz que entra en la lente. Un número F más bajo indica una mayor apertura, permitiendo más luz y mejor rendimiento en condiciones de poca iluminación.	F/1.7
<i>SNR de la cámara (pico)</i>	Relación señal a ruido.	1000:1
<i>Interfaz de cámara</i>	Medios de conexión y comunicación con otros dispositivos y sistemas.	CameraLink o GigE Vision
<i>Dimensiones</i>	Dimensiones físicas de la cámara.	150x85x75mm
<i>Peso</i>	Peso de la cámara.	1,56kg

## 3.2 VISUAL STUDIO

Visual Studio es un entorno de desarrollo integrado (IDE) creado por Microsoft. Proporciona una suite completa de herramientas de desarrollo para construir aplicaciones para diversas plataformas, como Windows, Android, iOS, web y servicios en la nube. Soporta múltiples lenguajes de programación, incluyendo C#, C++, Python, JavaScript, entre otros.



2. Logo Visual Studio

Se ha seleccionado Visual Studio por su robustez, versatilidad y la gran cantidad de herramientas que ofrece para el desarrollo de aplicaciones complejas, además de ser una de las aplicaciones utilizadas durante el grado, lo que evita la necesidad de aprender a utilizar el entorno desde cero.

También ha influido en su selección la facilidad de integración de librerías de código abierto útiles para el procesamiento de imágenes y desarrollo de interfaces gráficas, tales como OpenCV y Qt respectivamente.

## 3.3 LENGUAJE DE PROGRAMACIÓN (C++)

C++ es un lenguaje de programación de propósito general que se desarrolló como una extensión del lenguaje C en la década de los 80 por Bjarne Stroustrup. Combina características de programación de bajo nivel, típicas del lenguaje ensamblador, con características de programación de alto nivel, como la orientación a objetos. Esta combinación hace que C++ sea extremadamente eficiente, permitiendo tanto el control detallado de los recursos del hardware como la creación de programas complejos y estructurados.



3. Logo C++

La elección de C++ para el desarrollo de la aplicación HAT se fundamenta en dos razones clave:

- **Eficiencia y Rendimiento:** Dado que la aplicación requiere procesamiento intensivo de datos e imágenes, C++ es ideal por su capacidad de generar código altamente optimizado, esencial para el manejo de grandes volúmenes de datos en tiempo real.
- **Integración de Librerías:** C++ permite la fácil integración de librerías de código abierto como las utilizadas en este proyecto, las cuales nos ofrecen las herramientas necesarias para llevar a cabo el desarrollo de las funcionalidades de la aplicación.

### 3.4 OPEN CV

OpenCV (*Open Source Computer Vision Library*) es una biblioteca de software de visión por computadora y aprendizaje automático de código abierto. Desarrollada originalmente por Intel, ahora es mantenida por la comunidad y soportada por Willow Garage y Itseez.



4. Logo OpenCV

Proporciona más de 2500 algoritmos optimizados que abarcan una amplia gama de aplicaciones en el ámbito de la visión por computadora.

Sus características principales son las que siguen:

1. **Procesamiento de imágenes:** Ofrece funciones avanzadas para la manipulación de imágenes, como filtrado, transformación y análisis de características.
2. **Detección y reconocimiento de objetos:** Incluye herramientas para detección de caras, reconocimiento de objetos, y seguimiento de movimiento.
3. **Interfaz de usuario gráfica:** Proporciona capacidades para la creación de interfaces de usuario, facilitando la visualización de resultados y la interacción con los datos.
4. **Compatibilidad multiplataforma:** Disponible en múltiples sistemas operativos, incluyendo Windows, Linux, macOS, Android e iOS.
5. **Integración con otros lenguajes:** Ofrece bindings para lenguajes como Python, Java y MATLAB, además de C++, lo que permite una mayor flexibilidad en su uso.

Se ha seleccionado para el análisis de los hipercubos de la aplicación por su potencia para el procesamiento de imágenes, y la gran documentación y comunidad que ofrece apoyo ante cualquier duda o problema que surja durante el desarrollo de esta.

### 3.5 QT

Qt es un framework multiplataforma para el desarrollo de aplicaciones con interfaces gráficas de usuario (GUI) y aplicaciones sin interfaz gráfica (como herramientas en línea de comandos y servidores).

Desarrollado por The Qt Company, este marco de trabajo permite a los desarrolladores crear aplicaciones de alto rendimiento visualmente atractivas que pueden ejecutarse en diversas plataformas con mínima o ninguna modificación del código fuente.



5. Logo Qt

Para el desarrollo de la interfaz gráfica de la aplicación de análisis de hipercubos, Qt se ha integrado con Visual Studio por la interfaz intuitiva y atractiva que permite crear ventanas de manera sencilla, agilizando el proceso, y por la capacidad de integración existente entre este y OpenCV, que nos proporciona herramientas para que ambas librerías puedan comunicarse entre ellas para mostrar imágenes en la aplicación.

### 3.6 OTRAS HERRAMIENTAS UTILIZADAS

Además, se ha hecho uso de otras herramientas útiles para trabajar con el código, documentarlo, e incluso para crear iconos para la interfaz gráfica.

#### 3.6.1 GITHUB

GitHub es una plataforma de alojamiento de código fuente y control de versiones basada en Git. Fundada en 2008, permite a los desarrolladores colaborar en proyectos de software, gestionar versiones de código, y realizar un seguimiento de cambios de manera eficiente. GitHub proporciona herramientas para la gestión de proyectos, revisión de código, e integración continua, además de una interfaz web que facilita la visualización y edición del código.



6. Logo GitHub

Su utilización en el desarrollo de HAT, y de cualquier otra aplicación, es esencial debido a su capacidad para administrar el control de versiones, y la gestión de proyectos, asegurando desarrollo organizado y de alta calidad para la aplicación.

### 3.6.2 DOXYGEN

Doxygen es una herramienta de documentación para varios lenguajes de programación como C++, C, Java, Python y otros.



Permite a los desarrolladores generar documentación del código fuente en varios formatos, incluyendo HTML, PDF y LaTeX, a partir de comentarios estructurados dentro del código.

*7. Logo doxygen*

Doxygen cuenta con la capacidad de Generación automática de documentos, de forma que extrae información del código fuente y los comentarios para generar documentación completa y detallada. Esto asegura que la documentación del proyecto sea clara, completa y accesible para cualquier lector.

### 3.6.3 ADOBE PHOTOSHOP CS5

Adobe Photoshop CS5 es una versión popular del software de edición de imágenes desarrollado por Adobe Systems. Lanzado en abril de 2010, Photoshop CS5 consolidó su posición como una herramienta esencial para diseño de imágenes.



*8. Logo Adobe Photoshop CS5*

En el contexto del proyecto, se ha utilizado para la creación de los iconos, y ha sido seleccionada puesto que conocía su funcionamiento debido a años utilizándola para uso personal.

## 4 DESARROLLO DEL SOFTWARE

### 4.1 FUNCIONAMIENTO DE LA HERRAMIENTA

En este apartado se presenta una descripción detallada del funcionamiento de la herramienta desarrollada para el manejo de hipercubos generados por cámaras hiperespectrales. Ha sido diseñada para ofrecer un conjunto de funcionalidades avanzadas que permiten la carga, visualización y análisis de datos hiperespectrales de manera eficiente y precisa.



Figura 1. Interfaz base de HAT

Al abrir la interfaz del programa, como se muestra en la Figura 1, se presenta una botonera con varias funciones. Inicialmente, solo el botón *"Open Hypercube"* está activado, permitiendo la carga del archivo hiperespectral. Las otras funciones, incluyendo *"Añadir Material"*, *"ROI Creator"*, *"ROI View"*, *"ROI Analysis"*, *"ROI Comparator"*, *"Pointer"*, *"Save Data"* y *"Json Loader"*, se muestran desactivadas hasta que se complete la carga del hipercubo. Esta disposición asegura un flujo de trabajo estructurado y facilita la comprensión de los pasos necesarios para utilizar la herramienta de manera efectiva.

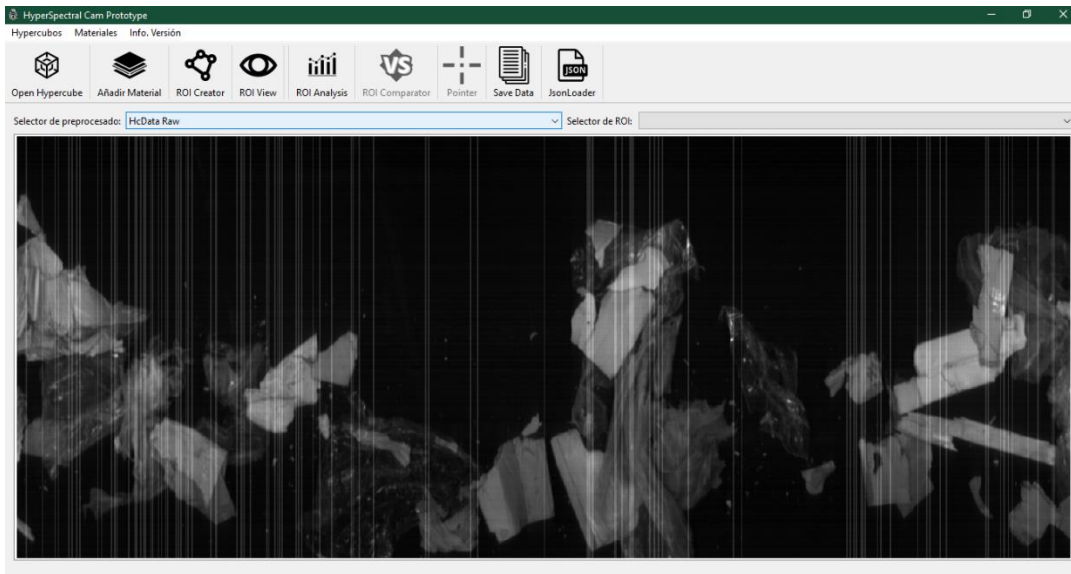


Figura 2. Visualización del hipercubo cargado

Una vez cargados los datos, la herramienta permite la visualización de la imagen capturada en pantalla (véase Figura 2). Esta funcionalidad es crucial para ofrecer una representación visual clara y comprensible de los datos hiperespectrales, permitiendo a los usuarios identificar rápidamente áreas de interés dentro de la imagen.

Además, se incluye la capacidad de definir regiones de interés (ROIs) para su análisis. Mediante la herramienta “ROI Creator”, los usuarios pueden seleccionar y delimitar áreas particulares de la imagen para un examen más detallado, tal y como se ve en la Figura 3. Esto es particularmente útil para focalizar el análisis en zonas específicas, optimizando el uso de recursos y mejorando la precisión del análisis.

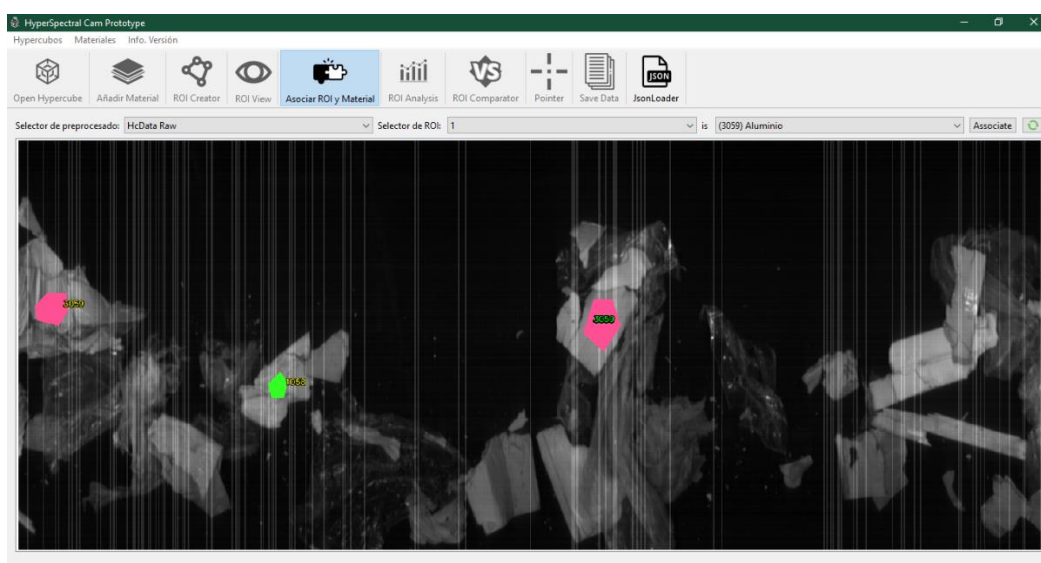


Figura 3. Asociación de ROIs y materiales

La visualización de las ROIs se realiza mediante la herramienta “ROI Viewer”, que, al ser seleccionada, despliega en la botonera la herramienta "Asociar ROI y Material". Esta función permite asociar las regiones delimitadas a materiales específicos, los cuales se visualizan con una etiqueta de su ID y un color correspondiente. Esto facilita la identificación rápida y clara de los materiales dentro de las imágenes hiperespectrales, mejorando así la interpretación y el análisis.

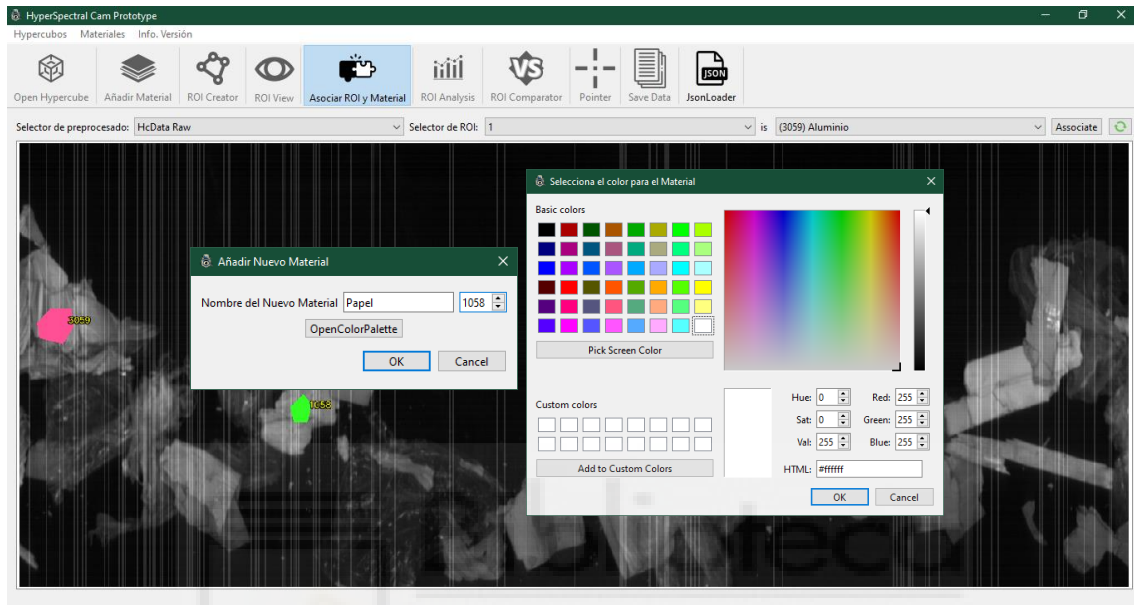


Figura 4. Adición de material

Los materiales pueden ser agregados a la sesión de trabajo utilizando la herramienta **Añadir Material**, la cual dispone de un espacio editable para introducir el nombre del material, un selector numérico que permite elegir un ID desde el 1000 hasta el 9999, garantizando un ID único, ya que la herramienta no permite utilizar un ID existente, y un botón que abre un selector de color para elegir el color asociado al material. Esta funcionalidad asegura que cada material esté claramente identificado y diferenciado visualmente en la interfaz.



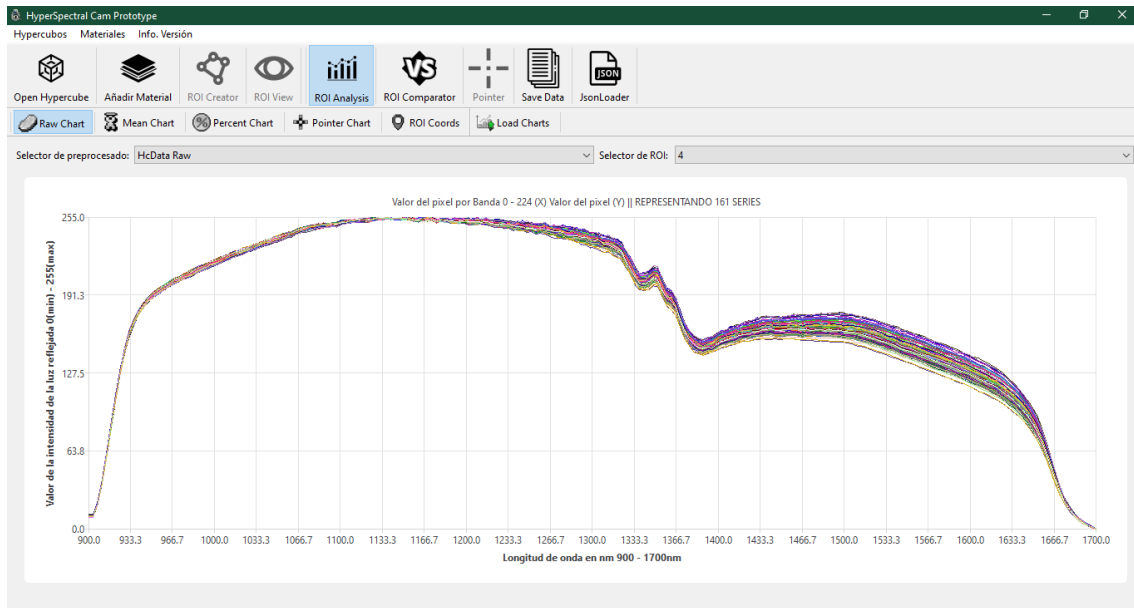


Figura 5. Vista de la gráfica en Raw

La herramienta también está equipada para la creación de gráficas que muestran los espectros de las regiones seleccionadas. Estas gráficas incluyen:

- La visualización de todas las curvas espectrales de los píxeles dentro de un ROI seleccionado.
- La opción de mostrar un porcentaje representativo de las curvas espectrales del ROI.
- Una gráfica que representa la curva espectral promedio de todas las curvas dentro del ROI.
- La capacidad de visualizar las coordenadas exactas de los píxeles incluidos en el ROI.

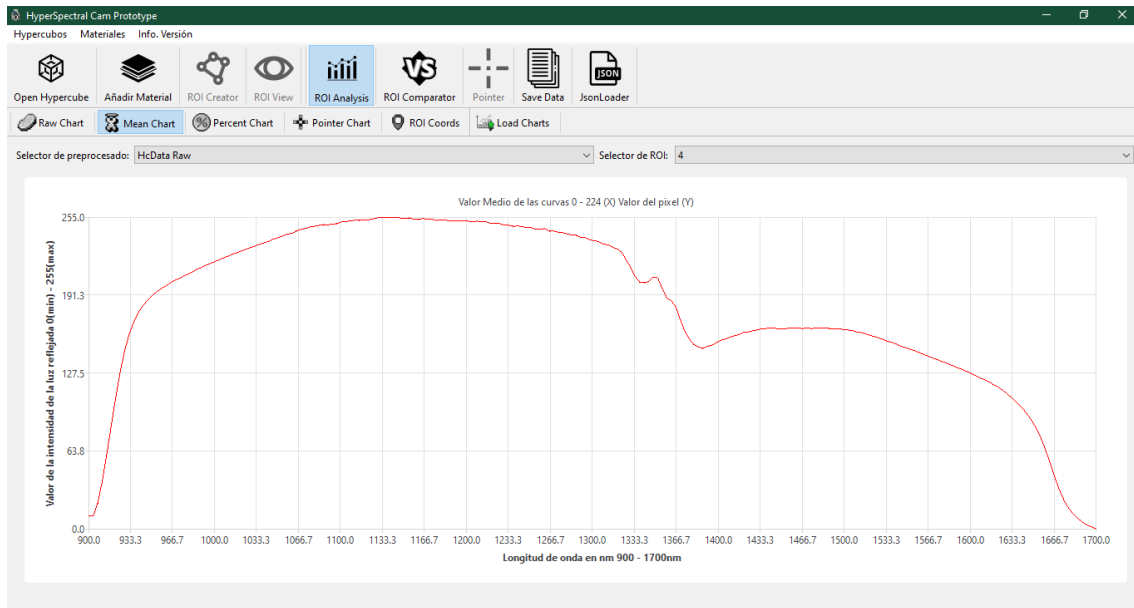


Figura 6. Vista de la gráfica de medias

Asimismo, se ha implementado la opción de comparar dos ROIs en cualquiera de las gráficas mencionadas, mediante la herramienta “*ROI Comparator*”. Esta funcionalidad permite realizar análisis comparativos detallados entre diferentes áreas de la imagen, facilitando la identificación de diferencias y similitudes espectrales (véase Figura 7).

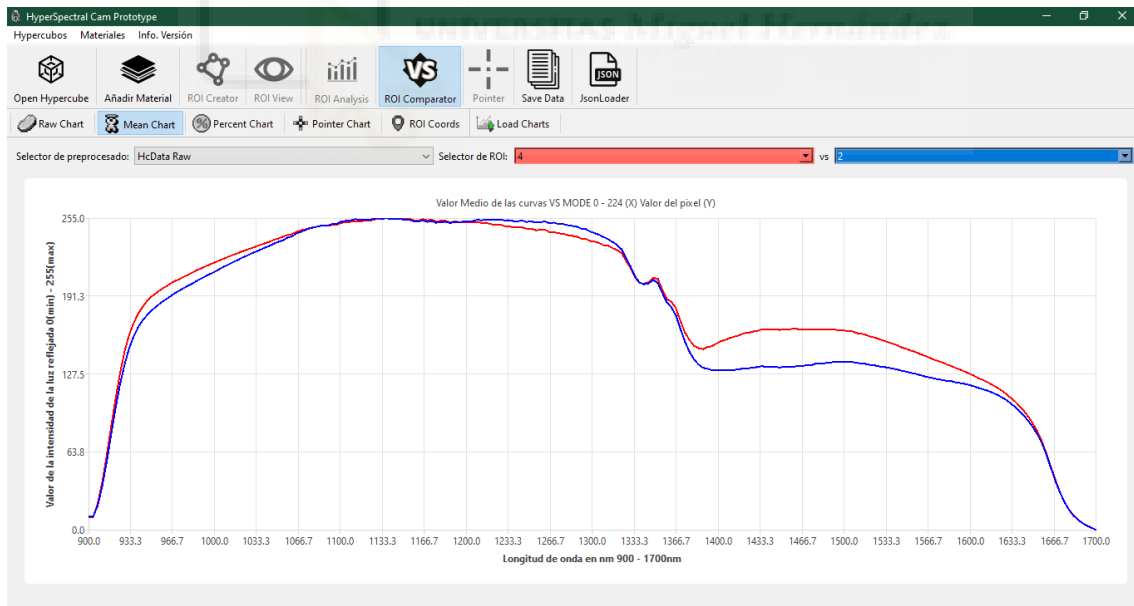


Figura 7. Vista de la gráfica de medias en comparativa

Otra característica importante es la herramienta “*Pointer*”, o puntero, para ver la curva del píxel seleccionado. Esta herramienta permite a los usuarios seleccionar cualquier píxel específico y visualizar su curva espectral individual, proporcionando un análisis detallado

a nivel de píxel, además de incluir la opción de añadir la curva como ROI para poder analizarla más adelante junto con otras curvas.

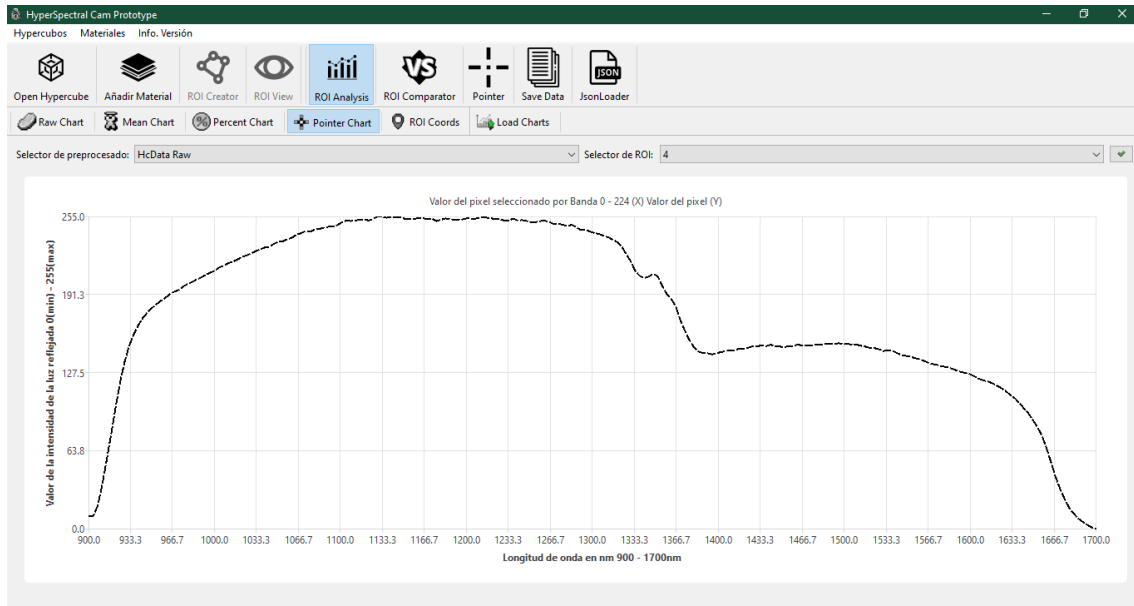


Figura 8. Vista de la visualización de la curva del puntero

Finalmente, la herramienta ofrece la opción de guardar los datos de la sesión en un archivo con formato “json” para su posterior carga. Esta funcionalidad permite la continuidad del análisis en futuras sesiones, asegurando que los datos y configuraciones se mantengan intactos y puedan ser reutilizados.

## 4.2 ESTRUCTURA DE LAS CLASES

Para el desarrollo de la herramienta mostrada en la sección anterior, se ha desarrollado una serie de clases en c++ que facilitan la modularidad del desarrollo.

El análisis de la estructura de clases es fundamental para comprender la organización y funcionamiento de la herramienta desarrollada. A continuación, se presenta el diagrama de flujo UML, el cual ilustra la interrelación entre las distintas clases, destacando sus responsabilidades y métodos principales.

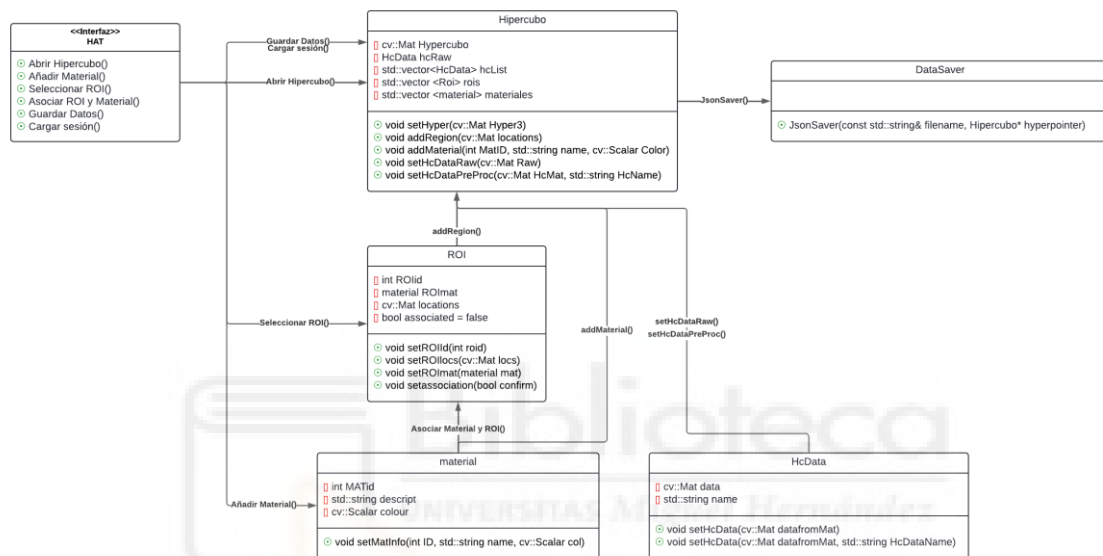


Figura 9. Diagrama UML de las clases del programa

Basándose en este diagrama, se describen las clases principales que componen la herramienta:

La primera de las clases, la más grande, se llama **HiperCubo**. En esta clase se guardan los datos de la carga de la imagen, así como los datos en crudo para analizarlos y datos preprocesados en caso de haber efectuado alguna interacción numérica con estos, en objetos de la clase **HcData**. En el caso de los datos originales, estos se almacenaran en el objeto hcRaw de tipo HcData, mientras que los preprocesados se almacenaran en un vector de tipo HcData llamado hcList.

Además, esta clase contiene un vector llamado Rois formado por elementos de la clase **ROI**, y otro vector de la clase **material**, llamado materiales.

#### 4.2.1 CLASE HIPERCUBO (PRIMER NIVEL)

La clase Hipercubo es la clase principal de gestión de los datos. Engloba a las demás y es donde se guardan los datos de la carga del hipercubo, así como los datos preprocesados para su análisis.

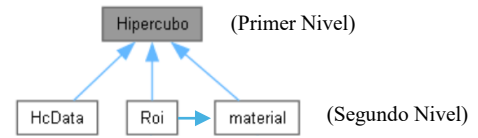


Figura 10. Diagrama de jerarquía de la clase Hipercubo

Además, esta clase contiene un vector de ROIs formado por elementos de la clase ROI, y otro vector de la clase materiales.

- **ATRIBUTOS PRIVADOS DE LA CLASE:**

---

<code>cv::Mat</code>	<b>Hypercubo</b>
	Matriz de OpenCV donde vamos a guardar el hipercubo cargado.

---

<code>std::vector&lt; Roi &gt;</code>	<b>rois</b>
	Vector de ROIs (Region of Interest) que contendrá este hipercubo.

---

<code>std::vector&lt;material&gt;</code>	<b>materiales</b>
	Convierte los datos del hipercubo en un formato que pueda ser leído por los elementos de Qt.

---

<b>HcData</b>	<b>hcRaw</b>
	Volumen da datos del cubo en raw.

---

<code>std::vector&lt;HcData&gt;</code>	<b>hcList</b>
	Vector de volúmenes de datos de los diferentes preprocesados que se han hecho sobre el hipercubo.

---

- **MÉTODOS PÚBLICOS PRINCIPALES DE LA CLASE:**

---

**Hipercubo ()**

**~Hipercubo ()**

---

void            **setHyper** (cv::Mat Hyper3)

Guarda el hipercubo cargado en la clase hipercubo.

---

cv::Mat        **getHyper** ()

Devuelve el hipercubo para extraer los datos.

---

void            **addRegion** (cv::Mat **locations**)

Crea un nuevo ROI y guarda las posiciones que se han seleccionado en el ROI Creator para almacenarlo en la lista de ROIs.

---

void            **addMaterial** ()

Añade un material a la lista de los materiales contenidos en la clase Hipercubo.

---

void            **setHcDataRaw** (cv::Mat Raw)

Guarda los datos del hipercubo en crudo.

---

void            **setHcDataPreProc** (cv::Mat HcMat, std::string HcName)

Guarda los datos del hipercubo preprocesado con un nombre.

---

- **FUNCIONES MIEMBRO DE LA CLASE:**

#### ◆ addMaterial()

```
void Hipercubo::addMaterial ( int      MatID,  
                             std::string name,  
                             cv::Scalar Color )
```

inline

Añade un material a la lista de los materiales contenidos en la clase **Hipercubo**.

Definición en la línea **159** del archivo **Hypercube.h**.

#### ◆ addRegion()

```
void Hipercubo::addRegion ( cv::Mat locations )
```

inline

Crea un nuevo ROI y guarda las posiciones que se han seleccionado en el ROI creator para almacenarlo en la lista de ROIs.

##### Parámetros

**locations** Matriz de coordenadas que componen el ROI

Definición en la línea **133** del archivo **Hypercube.h**.

#### ◆ getHyper()

```
cv::Mat Hipercubo::getHyper ( )
```

inline

Devuelve el hipercubo para extraer los datos.

##### Devuelve

cv::Mat

Definición en la línea **113** del archivo **Hypercube.h**.

### ◆ setHcDataPreProc()

```
void Hipercubo::setHcDataPreProc ( cv::Mat HcMat,  
                                  std::string HcName )
```

inline

Guarda los datos del hipercubo preprocesado con un nombre.

#### Parámetros

**HcMat** **Hipercubo** cargado con preprocesados realizados

**HcName** Nombre con el que queremos guardar los datos preprocesados

Definición en la línea **191** del archivo **Hypercube.h**.

### ◆ setHcDataRaw()

```
void Hipercubo::setHcDataRaw ( cv::Mat Raw )
```

inline

Guarda los datos del hipercubo en crudo.

#### Parámetros

**Raw** **Hipercubo** cargado sin preprocesados realizados

Definición en la línea **181** del archivo **Hypercube.h**.

### ◆ setHyper()

```
void Hipercubo::setHyper ( cv::Mat Hyper3 )
```

inline

Guarda el hipercubo cargado en la clase hipercubo.

#### Parámetros

**Hyper3** **Hipercubo** cargado

Definición en la línea **104** del archivo **Hypercube.h**.



#### 4.2.2 CLASE HCDATA (SEGUNDO NIVEL)

La clase HcData contiene los datos sobre los píxeles del hipercubo ya sean en crudo (raw) o preprocesados (tras aplicar alguna operación sobre los datos crudos). Por eso mismo, cada hipercubo contiene un vector de objetos de tipo HcData y un objeto llamado hcRaw también del tipo HcData.

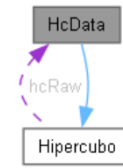


Figura 11. Diagrama de jerarquía de la clase HcData

Al acceder al hipercubo, los datos a analizar se leen de esta clase. Al

existir en la clase Hipercubo un vector que contiene varias versiones de esta clase, desde la aplicación existe la posibilidad de elegir con que versión del volumen de datos trabajaremos en la aplicación HAT. De esta forma no es necesario aplicar procesados cada vez, mejorando así la eficiencia en tiempo de ejecución, puesto que podemos acceder a ellos y saltar de unos a otros con rapidez y eficacia.

- **ATRIBUTOS PRIVADOS DE LA CLASE:**

cv::Mat

**data**

Matriz donde se guardan el volumen de datos precargados del cubo. Es decir que no se guardan los datos en las posiciones de la matriz, si no, en una variable que contiene todos los datos dispuestos en fila de donde nos será mucho más rápido acceder con el algoritmo adecuado agilizando la lectura de los datos.

std::string

**name**

Nombre que recibe el volumen da datos del hipercubo. Raw o un nombre en concreto si es un preprocesado.

bool

**cropped**

Su finalidad es indicar si en este preprocesado de datos tenemos un rango de longitudes de onda recortado. Es decir, si hemos seleccionado un rango de longitudes específico y no trabajamos con todas.

int

**iniBand, endBand**

Estos dos enteros marcan la banda inicial y la final del rango de longitudes de onda con el que trabajamos.

- **MÉTODOS PÚBLICOS DE LA CLASE:**

---

void                    **setHcData** (cv::Mat datafromMat)

Primera sobrecarga de la función setHcData(). Carga los datos en raw del hipercubo con el nombre "HcData Raw".

---

void                    **setHcData** (cv::Mat datafromMat, std::string HcDataName)

Segunda sobrecarga de la función setHcData(). Carga los datos del preprocesado realizado del hipercubo con un nombre elegido por el usuario.

---

cv::Mat                **getDatafromHc** ()

Devuelve los datos guardados de este volumen de datos.

---

- **FUNCIONES MIEMBRO DE LA CLASE:**

◆ **getDataformHc()**

cv::Mat HcData::getDataformHc ( ) inline

---

Devuelve los datos guardados de este volumen de datos.

**Devuelve**  
cv::Mat

Definición en la línea **67** del archivo **HcData.h**.

◆ **setHcData()** [1/2]

void HcData::setHcData ( cv::Mat datafromMat ) inline

---

Primera sobrecarga de la función setHcData. Carga los datos en raw del hipercubo con el nombre "HcData Raw".

**Parámetros**  
**datafromMat** Matriz de OpenCV del volumen datos del hipercubo

Definición en la línea **45** del archivo **HcData.h**.

## ◆ setHcData() [2/2]

```
void HcData::setHcData ( cv::Mat  datafromMat,  
                        std::string HcDataName )
```

inline

Segunda sobrecarga de la función setHcData. Carga los datos de el preprocesado realizado del hipercubo con el nombre que elijamos.

### Parámetros

**datafromMat** Matriz de OpenCV del volumen datos del hipercubo

**HcDataName** Nombre que se le va a asignar al volumen de datos.

Definición en la línea 57 del archivo **HcData.h**.



### 4.2.3 CLASE ROI (SEGUNDO NIVEL)

La clase ROI (*Region of Interest*) guarda los datos de localización de los Rois creados por el usuario en la aplicación. Tiene como función diferenciar cada ROI de los demás almacenando un ID único para cada región y acumular las localizaciones de todos los píxeles que lo conforman.

De esta manera, al analizar los datos, HAT analizará solo los píxeles archivados en el ROI del ID seleccionado, creando la posibilidad de comparar varias partes de la imagen para analizar espectros de materiales distintos.

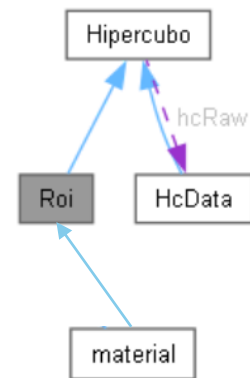


Figura 12. Diagrama de jerarquía de la clase ROI

- **ATRIBUTOS PRIVADOS DE LA CLASE:**

int	<b>ROIid</b>	Número identificador del ROI.
material	<b>ROImat</b>	Material asociado al ROI.
cv::Mat	<b>locations</b>	Coordenadas de los píxeles de la imagen que conforman el ROI.
bool	<b>associated</b>	Booleano que permite dar a conocer al programa si el ROI tiene un material asociado o no.

- **MÉTODOS PÚBLICOS PRINCIPALES DE LA CLASE:**

---

**Roi ()**

---

**~Roi ()**

---

std::string      **getROIIdtoString** (cv::Mat locs)

Transforma el ID del ROI en una string o cadena para que pueda ser utilizada para mostrar como texto en los gadgets de Qt.

---

void      **setROIlocs** (cv::Mat locs)

Guarda las coordenadas de los píxeles recogidas por la herramienta ROISelector.

---

void      **setROIId** (int roid)

Guarda el ID único del ROI.

---

cv::Mat      **retlocs** ()

Devuelve las coordenadas de los píxeles del ROI para que sean utilizadas por otras funciones.

---

void      **setROI**mat (material mat)

Establece el material asociado al ROI seleccionado.

---

**material**      **getROI**mat ()

Devuelve el material asociado al ROI.

---

void      **setassociation** (bool confirm)

Establece si el ROI está asociado o no a un material.

---

- **FUNCIONES MIEMBRO DE LA CLASE:**

◆ **getROIIdtoString()**

std::string Roi::getROIIdtoString ( ) inline

Transforma el ID del ROI en una string o cadena para que pueda ser utilizada para mostrar como texto en los gadgets de Qt.

**Devuelve**

std::string

Definición en la línea **89** del archivo **roi.h**.

### ◆ retlocs()

cv::Mat Roi::retlocs ( )

inline

Devuelve las coordenadas de los píxeles del ROI para que sean utilizadas por otras funciones.

#### Devuelve

cv::Mat

Definición en la línea 117 del archivo [roi.h](#).

### ◆ setROIId()

void Roi::setROIId ( int roid )

inline

Guarda el ID único del ROI.

#### Parámetros

**roid** Número Identificativo del ROI

Definición en la línea 108 del archivo [roi.h](#).

### ◆ setROIlocs()

void Roi::setROIlocs ( cv::Mat locs )

inline

Guarda las coordenadas de los píxeles recogidas por la herramienta [ROISelector](#).

#### Parámetros

**locs** Coordenadas de los píxeles de la imagen que conforman el ROI

Definición en la línea 99 del archivo [roi.h](#).

### ◆ setROIImat()

void Roi::setROIImat ( **material** mat )

inline

Establece el material asociado al ROI seleccionado.

#### Parámetros

**mat** material de la lista que será pasado desde la clase [Hipercubo](#)

Definición en la línea 126 del archivo [roi.h](#).

### ◆ getROImat()

**material** Roi::getROImat ( ) inline

Devuelve el material asociado al ROI.

#### **Devuelve**

material

Definición en la línea **135** del archivo **roi.h**.

### ◆ setassociation()

void Roi::setassociation ( bool confirm ) inline

Establece si el ROI esta asociado o no a un material.

#### **Parámetros**

**confirm** booleano que marca true si está asociado o false si no lo está

Definición en la línea **144** del archivo **roi.h**.



#### 4.2.4 CLASE MATERIAL (SEGUNDO NIVEL)

La clase Material almacena atributos para asociar los ROIs a un material creado por el usuario en la aplicación. La finalidad principal de la clase `material` es permitir al usuario clasificar y organizar los ROIs de manera efectiva. Al asociar cada ROI con un material específico, utilizando el ID, el nombre y el color, se mejora significativamente la visualización y comprensión de los datos. Esta clasificación facilita el análisis comparativo y la detección de patrones dentro de las imágenes hiperespectrales, contribuyendo a una mejor toma de decisiones basada en datos precisos y bien organizados.

- **ATRIBUTOS PRIVADOS DE LA CLASE:**

int	<b>MATid</b>	Número identificador del material.
std::string	<b>descript</b>	Descripción breve o nombre del material.
cv::Scalar	<b>colour</b>	Variable escalar que guarda los valores RGB del color designado al material.

- **MÉTODOS PÚBLICOS PRINCIPALES DE LA CLASE:**

	<b>material ()</b>	
	<b>~material ()</b>	
void	<b>setMatInfo</b> (int ID, std::string name, cv::Scalar col)	Guarda la información de los valores de entrada en las variables de la clase.
int	<b>getMatID ()</b>	Devuelve el ID del material.
std::string	<b>getMatName ()</b>	Devuelve el nombre del material.
cv::Scalar	<b>getMatColour ()</b>	Devuelve el escalar del color del material.



- **FUNCIONES MIEMBRO DE LA CLASE**

### ◆ setMatInfo()

```
void material::setMatInfo ( int      ID,  
                           std::string name,  
                           cv::Scalar col )
```

inline

Guarda la información de los valores de entrada en las variables de la clase.

#### Parámetros

**ID** ID del Material  
**name** Nombre del Material  
**col** Escalar del color del material

### ◆ getMatID()

```
int material::getMatID ( )
```

inline

Devuelve el ID del material.

**Devuelve**  
int

### ◆ getMatName()

```
std::string material::getMatName ( )
```

inline

Devuelve el nombre del material.

**Devuelve**  
std::string

### ◆ getMatColour()

```
cv::Scalar material::getMatColour ( )
```

inline

Devuelve el escalar del color del material.

**Devuelve**  
cv::Scalar

## 5 CONCLUSIONES

---

El desarrollo de este proyecto ha representado un desafío significativo, especialmente considerando que la programación no es uno de los temas más profundamente tratados en mi carrera. Sin embargo, agradezco la oportunidad de haberme enfrentado a este reto, ya que ha sido una experiencia enriquecedora que ha ampliado mis horizontes académicos y profesionales.

A lo largo del proyecto, he tenido la oportunidad de profundizar en el tratamiento de datos hiperspectrales y de descubrir tecnologías que, hasta ahora, no había explorado. Esta experiencia me ha permitido mejorar mi comprensión de la gestión y análisis de grandes volúmenes de datos, así como de la aplicación práctica de herramientas avanzadas de software. Cada obstáculo superado y cada nueva habilidad adquirida han contribuido a mi crecimiento personal y profesional, proporcionando una sólida base para futuros proyectos.

Además, el trabajo en este proyecto ha estimulado mi capacidad creativa y me ha permitido valorar la importancia del trabajo duro y la dedicación. He generado nuevas ideas para mejorar y expandir la herramienta ya desarrollada, lo que demuestra que este proyecto ha sido solo el comienzo de un camino lleno de posibilidades. La emoción y el entusiasmo que siento al pensar en los futuros proyectos que podré llevar a cabo son inmensos, y estoy deseoso de continuar explorando y aprendiendo.

Concluyo este trabajo con la satisfacción de haber cumplido con los objetivos propuestos y con la convicción de que este será el primero de muchos proyectos a lo largo de mi carrera profesional. Esta experiencia no solo ha fortalecido mis habilidades técnicas, sino que también ha reafirmado mi pasión por la innovación y la resolución de problemas complejos. Estoy emocionado por las oportunidades futuras y agradecido por el aprendizaje obtenido durante este proceso.

## 6 BIBLIOGRAFÍA CONSULTADA

---

### *Introducción a la tecnología hiperespectral.*

- [1]. Wikipedia. "Hyperspectral imaging." Disponible en:  
[https://en.wikipedia.org/wiki/Hyperspectral\\_imaging](https://en.wikipedia.org/wiki/Hyperspectral_imaging)
- [2]. Specim. "What is hyperspectral imaging?" Disponible en:  
<https://www.specim.com/technology/what-is-hyperspectral-imaging/>
- [3]. Springer. "Hyperspectral Imaging for Sustainable Waste Recycling." Disponible en:  
<https://link.springer.com/article/10.1007/s00216-022-03959-y>
- [4]. Cambridge University Press. "Introduction to Hyperspectral Imaging and Remote Sensing." Disponible en: <https://www.cambridge.org/core/books/abs/hyperspectral-imaging-remote-sensing/introduction/C3B81BE172ACAC562BA8EC479D6B3B3B>

### *Proceso de captura de una cámara hiperespectral.*

- [5]. Advian. "What is hyperspectral imaging?" Disponible en:  
<https://www.advian.fi/en/what-is-hyperspectral-imaging>
- [6]. Spectral Imaging Hub. "How do spectral sensors work?" Disponible en:  
[https://spectralimaginghub.com/hyperspectral\\_imaging\\_how\\_do\\_spectral\\_sensors\\_work/](https://spectralimaginghub.com/hyperspectral_imaging_how_do_spectral_sensors_work/)
- [7]. IMEC. "Hyperspectral Imaging Technology." Disponible en:  
<https://www.imechyperspectral.com/en/hyperspectral-imaging-technology>
- [8]. Photonic Science. "Hyperspectral Imaging: Principles & Applications." Disponible en: <https://photonicscience.com/hyperspectral-imaging-principles-applications/>
- [9]. Middleton Spectral Vision. "What is hyperspectral imaging?" Disponible en:  
<https://www.middletonspectral.com/resources/what-is-hyperspectral-imaging/>

### *Estado del arte de la clasificación de plásticos.*

- [10]. Springer. "Review on Automatic Plastic Classification Methods." Disponible en:  
<https://link.springer.com/article/10.1007/s12008-023-01320-w>
- [11]. IEEE. "Plastic Waste Classification Using Hyperspectral Imaging." Disponible en:  
<https://ieeexplore.ieee.org/document/9506074>
- [12]. MDPI. "Sustainable Plastic Waste Management." Disponible en:  
<https://www.mdpi.com/2071-1050/15/10/7752>

- [13]. MDPI. "Hyperspectral Imaging for Plastic Detection." Disponible en: <https://www.mdpi.com/2313-433X/5/5/52>
- [14]. Agarap, A.F. "An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification." Disponible en: <https://arxiv.org/abs/1712.03541>
- [15]. ScienceDirect. "Generative Adversarial Networks for Multimodal Data Integration." Disponible en: <https://www.sciencedirect.com/science/article/pii/S095741742030346X>
- [16]. Carion, N., et al. "End-to-End Object Detection with Transformers." Disponible en: <https://arxiv.org/abs/2005.12872>
- [17]. Redmon, J., et al. "You Only Look Once: Unified, Real-Time Object Detection." Disponible en: <https://arxiv.org/abs/1506.02640>

*Cámara Hiperespectral utilizada.*

- [18]. Channel Systems. "Specim FX17 900-1700 nm." Disponible en: <https://channelsystems.ca/products/near-infrared-nir/specim-fx17-900-1700-nm>

*Visual Studio.*

- [19]. Microsoft. "Visual Studio." Disponible en: <https://visualstudio.microsoft.com/es/>

*C++:*

- [20]. Microsoft. "C++ Documentation." Disponible en: <https://learn.microsoft.com/en-us/cpp/cpp/?view=msvc-170>
- [21]. Wikipedia. "C++." Disponible en: <https://en.wikipedia.org/wiki/C%2B%2B>

*OpenCV:*

- [22]. OpenCV. "OpenCV Library." Disponible en: <https://opencv.org/>
- [23]. GitHub. "OpenCV Repository." Disponible en: <https://github.com/opencv/opencv>

*Qt:*

- [24]. Qt Company. "Qt." Disponible en: <https://www.qt.io/>
- [25]. Wikipedia. "Qt (software)." Disponible en: [https://en.wikipedia.org/wiki/Qt\\_\(software\)](https://en.wikipedia.org/wiki/Qt_(software))

*GitHub:*

[26]. GitHub. "GitHub." Disponible en: <https://github.com/>

[27]. Wikipedia. "GitHub." Disponible en: <https://en.wikipedia.org/wiki/GitHub>

*Doxygen:*

[28]. Doxygen. "Doxygen." Disponible en: <https://www.doxygen.nl/>

[29]. Wikipedia. "Doxygen." Disponible en: <https://en.wikipedia.org/wiki/Doxygen>



## ANEXO I: HAT (MANUAL DE USUARIO)

---

Este manual de usuario está diseñado para guiar a los usuarios en el uso de la herramienta de análisis hiperespectral desarrollada en mi Trabajo de Fin de Grado (TFG). La herramienta HAT (*Hyperspectral Analysis Tool*) tiene como objetivo facilitar la visualización de espectros y la clasificación de materiales mediante el procesamiento de datos hiperespectrales.

El análisis hiperespectral es una técnica avanzada que captura y procesa información a lo largo de un amplio rango del espectro electromagnético. Esto permite identificar y clasificar diferentes materiales en función de sus firmas espectrales únicas. Nuestra herramienta ofrece una interfaz intuitiva y una serie de funcionalidades que permiten a los usuarios cargar datos hiperespectrales, visualizar espectros y realizar clasificaciones de materiales de manera eficiente.

A lo largo de este manual, se proporcionarán instrucciones detalladas sobre cómo utilizar cada una de las características del programa, asegurando que incluso aquellos usuarios sin experiencia previa en análisis hiperespectral puedan aprovechar al máximo las capacidades de la herramienta.

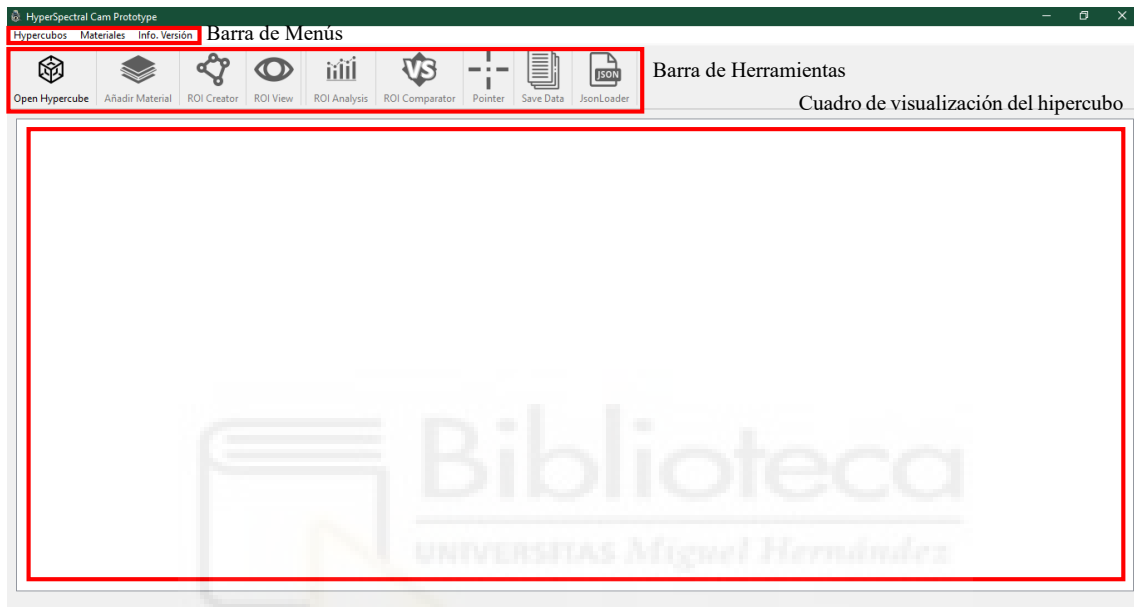


Figura 13. Logotipo e icono de la herramienta HAT

## PRIMEROS PASOS

### INTERFAZ DE USUARIO Y ESPACIO DE TRABAJO

La interfaz de usuario de la herramienta de análisis hiperespectral ha sido diseñada para ser intuitiva y fácil de usar, permitiendo a los usuarios acceder rápidamente a todas las funciones necesarias para la visualización y clasificación de materiales. A continuación, se presenta una visión general de los componentes principales de la interfaz, basada en la imagen proporcionada.



Como se puede comprobar en la figura disponemos de una botonera extensa con diversas funciones.

### BARRA DE MENÚ




La barra de menú, ubicada en la parte superior de la ventana, contiene las siguientes opciones:


- **Hipercubos:** Permite abrir los archivos binarios que contienen los datos capturados por la cámara hiperespectral.
- **Materiales:** Contiene la opción de añadir materiales al espacio de trabajo.
- **Info. Versión:** Muestra un cuadro de diálogo con el número de versión, fecha de revisión y nombre del creador de la herramienta.


## BARRA DE HERRAMIENTAS


Justo debajo de la barra de menú se encuentra la barra de herramienta, equipada con una amplia gama de funciones. A continuación, se muestra la botonera desplegada totalmente y cada una de las funciones que desempeña:





 **Open Hypercube:** Al igual que “Abrir Hipercono” de la barra de menú, permite abrir los archivos binarios que contienen los datos capturados por la cámara hiperespectral.


 **Añadir Material:** También ejerce la misma función que su versión en la barra de menú “Materiales”. Proporciona la opción de añadir materiales al espacio de trabajo. Esta función sólo puede ser activada una vez se ha realizado la carga del Hipercono.

 **ROI Creator:** Abre el modo de creador de ROIs. Esta función sólo puede ser activada una vez se ha realizado la carga del Hipercono.


 **ROI View:** Cambia la vista previa de la imagen visible del Hipercono, a una vista donde podemos ver superpuestas las regiones dibujadas con ROI Creator. Esta función sólo puede ser activada una vez se ha realizado la carga del Hipercono.


 **Asociar ROI y Material:** Permite al usuario asignar materiales creados a ROIs definidos para que sean clasificados en la vista de ROI View. Esta función sólo puede ser activada una vez se haya activado el “ROI View” y exista al menos un material y un ROI.


 **ROI Analysis:** Abre la interfaz de gráficas que le permiten al usuario realizar el análisis de las huellas espectrales de los materiales seleccionados por los ROIs. Esta función sólo puede ser activada una vez se ha realizado la carga del Hipercono. También será necesario la existencia de al menos un ROI para poder visualizar datos.

 **ROI Comparator:** Una vez dentro de la interfaz de gráficas, o de visualización de espectros, de “ROI Analysis” se puede activar esta función para dibujar 2 resultados de ROI a la vez, uno en azul y otro en rojo, para compararlos al mismo tiempo.



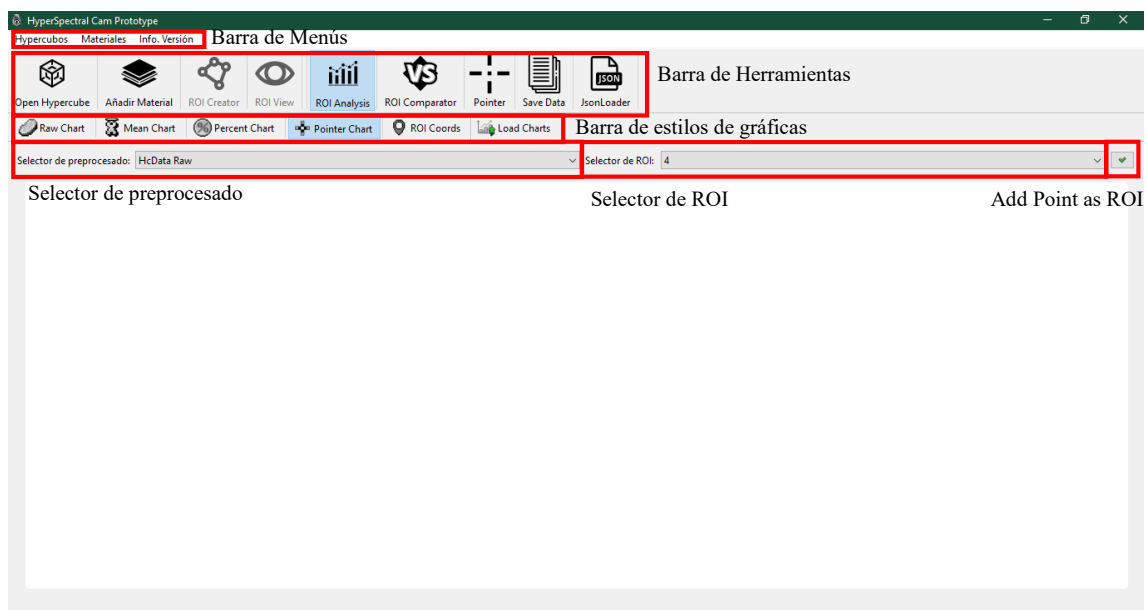
 **Pointer:** Desde la función de “*ROI Analysis*” el usuario puede elegir la gráfica “pointer” para que se desbloquee esta función, la que nos permite ver la curva de un único pixel de la imagen hiperespectral.

 **Save Data:** Función de guardado. Guarda un archivo binario del cubo que se estaba leyendo en la sesión, así como un .json donde se almacenan todos los datos de guardado sobre ROIs y Materiales para poder mantenerlos de forma indefinida como archivos locales, permitiendo al usuario cargar sus sesiones a deseo. Esta función sólo puede ser activada una vez se ha realizado la carga del Hipercono.

 **JsonLoader:** Función de carga de archivos de guardado guardados en un .json. Esta función sólo puede ser activada una vez se ha realizado la carga del Hipercono.



# INTERFAZ DE VISUALIZACIÓN DE ESPECTROS



En esta sección específica de la interfaz, se ha añadido una barra de herramientas adicional para mejorar la funcionalidad y el análisis de los datos espectrales.

## BARRA DE ESTILOS DE GRÁFICAS



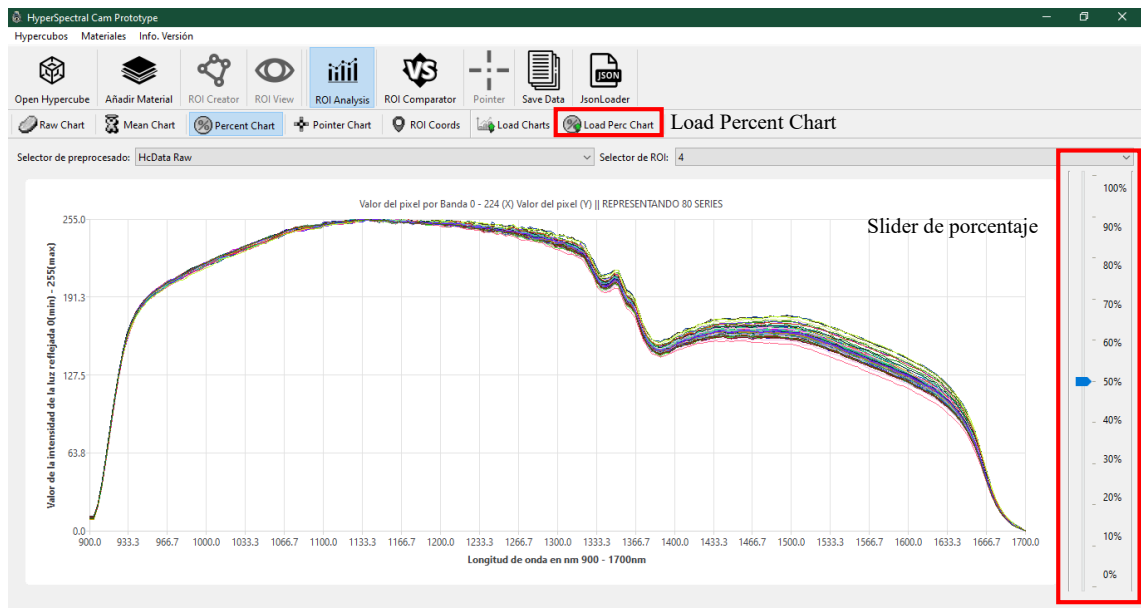
**Raw Chart:** Muestra al usuario la ventana donde se creará el gráfico con las curvas de todos los píxeles al completo del ROI seleccionado. La gráfica se dibuja pulsando *Load Chart* con el ROI y el preprocesado elegidos.




**Mean Chart:** En la ventana que se muestra cuando el usuario activa esta función se crea una gráfica de una sola curva, media de todas las curvas del ROI seleccionado. La gráfica se dibuja pulsando *Load Chart* con el ROI y el preprocesado elegidos.




**Percent Chart:** Hace aparecer un “*slider*” en la parte derecha de la aplicación con la que nos permite seleccionar un porcentaje de curvas del total que van a ser representadas aleatoriamente de entre todas las que pueden dibujarse en el ROI seleccionado.





La gráfica se dibuja pulsando *Load Percent Chart* con el ROI y el preprocesado elegidos.

 **Pointer Chart:** Desbloquea la herramienta “Pointer” al usuario, la cual permite elegir un pixel a mano de la imagen y representar la curva inmediatamente tras hacerlo. Se puede guardar como dato de ROI mediante el botón que aparece también con esta opción “*Add Point*”.

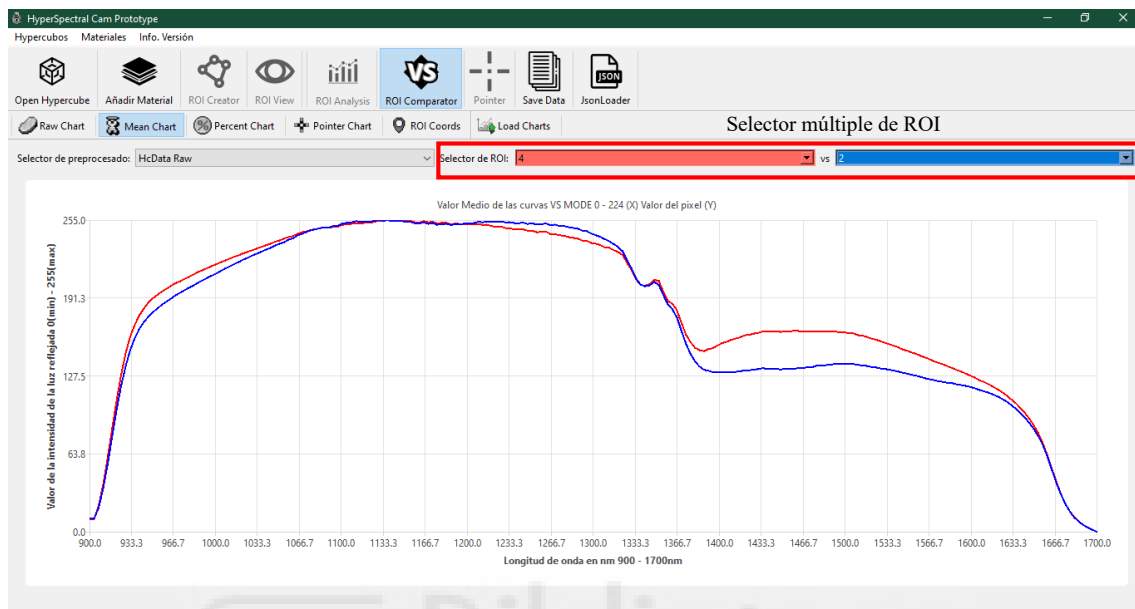
 **Add Point:** Permite añadir el punto visualizado como ROI para su análisis.

 **ROI Coords:** Muestra las coordenadas del ROI seleccionado. Los puntos se muestran pulsando *Load Chart* con el ROI y el preprocesado elegidos.

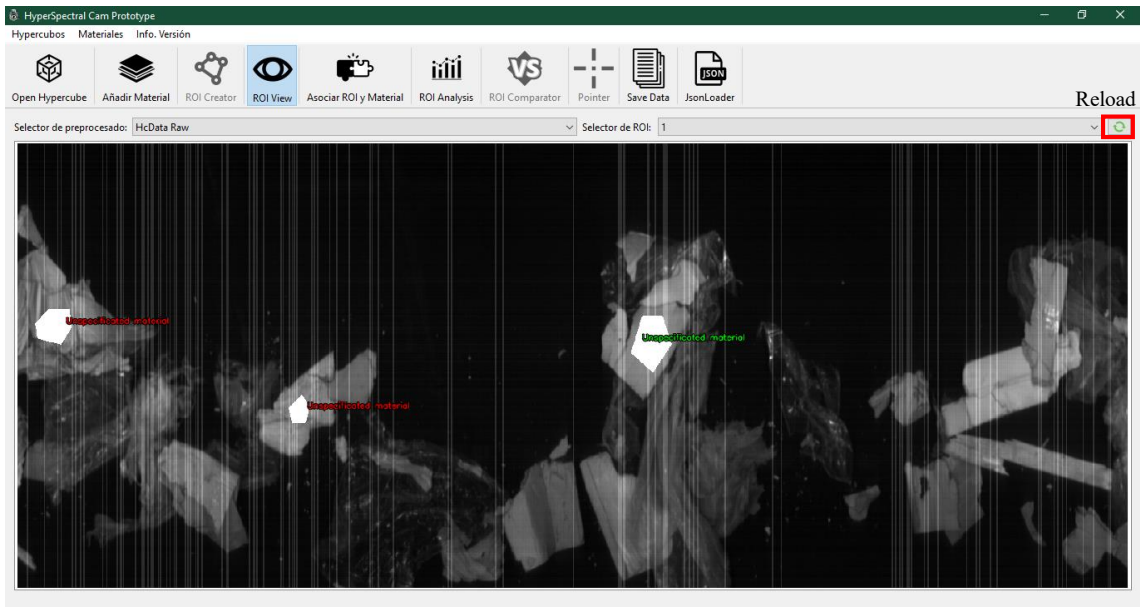
 **Load Charts:** Carga con el ROI y el HcData seleccionados las gráficas “*Raw Chart*”, “*Mean Chart*” y “*ROI Coords*”.

 **Load Percent Chart:** Carga con el ROI, el HcData y el porcentaje seleccionado la gráfica “*Percent Chart*”.


Si activamos la opción “ROI Comparator” el selector de ROIs cambiará por 2 selectores de color rojo y azul, donde podremos elegir los ROIs a comparar y utilizar las mismas gráficas.



## INTERFAZ DE VISUALIZACIÓN DE ROIS

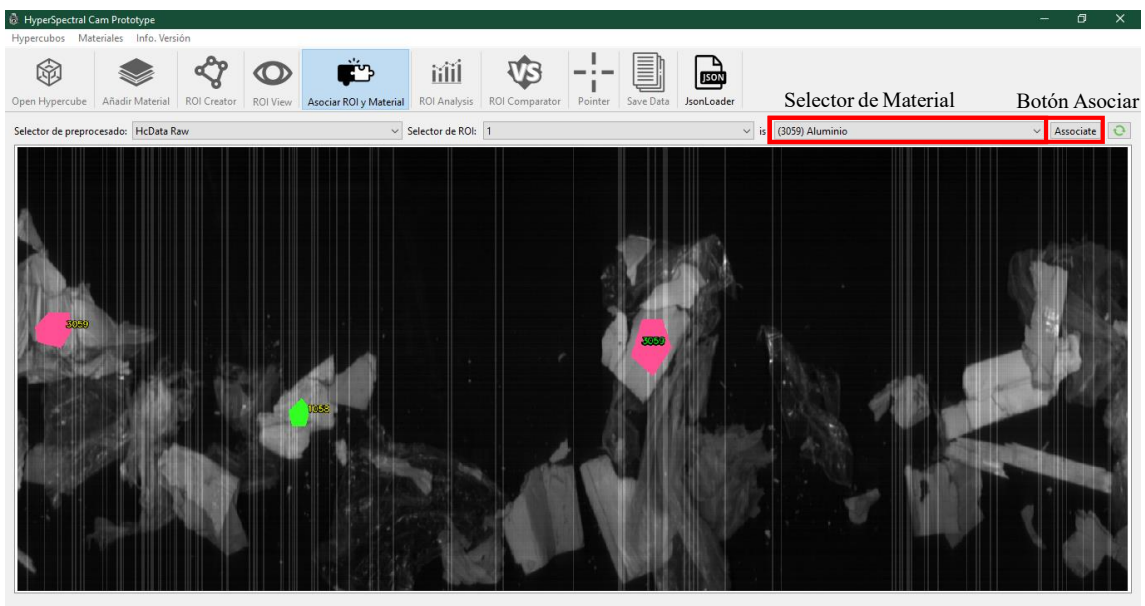


Aquí los usuarios pueden interactuar con la visualización, viendo con que ROI se está trabajando, que material tiene asociado etc.

 **Reload Button:** Este botón permite al usuario refrescar la vista del “ROI View” de forma que:

- Pueda ver con letras verdes el ROI Seleccionado por el “ROI Selector”
- Recargue la vista de los ROIs actualizando los materiales asociados.

### ROI VIEW MODO: ASOCIACIÓN



En este modo el usuario puede seleccionar un ROI mediante el Selector de ROIs, un material mediante el Selector de materiales y utilizar el “Botón *Associate*” para asignarle el material al ROI seleccionado.

Mediante el botón Reload podemos ver en un momento como se actualiza la vista para comprobar que, efectivamente, el material ha sido asociado y se representa correctamente.

