



UNIVERSIDAD DE LA FRONTERA
FACULTAD DE INGENIERÍA Y CIENCIAS
DEPARTAMENTO DE CIENCIAS FÍSICAS

“DESARROLLO Y OPTIMIZACIÓN DE UN SOFTWARE
PARA LA OBTENCIÓN Y CONSTRUCCIÓN
TRIDIMENSIONAL DE INTENSIDADES DE
FLUORESCENCIA DE RAYOS X ASOCIADAS A
ELEMENTOS QUÍMICOS PRESENTES EN UN MATERIAL”

BRYAN ENRIQUE CASANELLI RIVEROS

2021



UNIVERSIDAD DE LA FRONTERA
FACULTAD DE INGENIERÍA Y CIENCIAS
DEPARTAMENTO DE CIENCIAS FÍSICAS

**“DESARROLLO Y OPTIMIZACIÓN DE UN SOFTWARE
PARA LA OBTENCIÓN Y CONSTRUCCIÓN
TRIDIMENSIONAL DE INTENSIDADES DE
FLUORESCENCIA DE RAYOS X ASOCIADAS A
ELEMENTOS QUÍMICOS PRESENTES EN UN MATERIAL”**

**TRABAJO PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL FÍSICO**

Profesor Guía: PROF. DR. RODOLFO GABRIEL FIGUEROA SAAVEDRA
Profesor Co-Guía: PROF. MSC. ING. JAIME ALEJANDRO GUARDA CHAMORRO

BRYAN ENRIQUE CASANELLI RIVEROS

2021

**DESARROLLO Y OPTIMIZACIÓN DE UN SOFTWARE PARA LA
OBTENCIÓN Y CONSTRUCCIÓN TRIDIMENSIONAL DE
INTENSIDADES DE FLUORESCENCIA DE RAYOS X ASOCIADAS A
ELEMENTOS QUÍMICOS PRESENTES EN UN MATERIAL**

Bryan Enrique Casanelli Riveros

COMISIÓN EXAMINADORA

**PROF. DR. RODOLFO
GABRIEL FIGUEROA
SAAVEDRA
Profesor Guía**

**PROF. MSC. ING. JAIME
ALEJANDRO GUARDA
CHAMORRO
Profesor Co-Guía**

**PROF. DR. PABLO IVÁN
DÍAZ RIQUELME
Académico Evaluador**

**PROF. DR. MAURO ANDRÉS
VALENTE
Académico Evaluador**

Calificación Plan de Trabajo :
Calificación Informe Escrito :
Calificación Presentación Oral :
Calificación Final :

*A mi familia y profesores,
quienes me apoyaron y guiaron
a lo largo de esta etapa.*

Agradecimientos

En primer lugar quisiera agradecer a mi familia, cuyo continuo apoyo ha sido fundamental a lo largo de estos 6 años, sobre todo en los momentos más difíciles de la carrera.

También agradezco al profesor Dr. Rodolfo Figueroa, quien dirige el proyecto FONDECYT OXIRIS, en el cual se enmarca este trabajo, y quien además ha sido mi profesor guía durante esta etapa, ayudándome a fijar el camino a seguir para cumplir con los objetivos fijados.

En este contexto, quiero agradecer muy especialmente al MSc. Ing. Jaime Guarda, quien ha sido mi profesor, compañero y colega, y cuya ayuda fue fundamental para introducirme al proyecto OXIRIS, entender la electrónica y física asociada, y realizar pruebas de concepto.

Adicionalmente, agradezco al Ing. Jorge Leiva y al Ing. Eduardo Quilagüilque, cuyo trabajo en OXIRIS ha servido de base para el desarrollo aquí presentado.

Agradezco también a la Universidad de La Frontera, particularmente al Departamento de Ciencias Físicas y sus profesores, lugar donde realicé los estudios que me han permitido desarrollar este trabajo.

Finalmente, mi más profundo respeto, admiración y agradecimiento a los profesores Dra. Fabiola Arévalo, Dr. Pablo Díaz, Dr. Fabián Torres, Mg. Julio Valdés y Dr. Mauro Valente, quienes contribuyeron no sólo a mi formación académica, sino que también a mi crecimiento personal.

Marzo, 2021.
Temuco, Chile.

Resumen

Se desarrolla el software principal del primer prototipo del proyecto OXIRIS (*Orthovoltage X-Ray Induced Radiation Integral System*), permitiendo el trabajo coordinado entre múltiples detectores de radiación, una fuente de rayos X giratoria y un sistema de posicionamiento.

El software desarrollado permite controlar el sistema OXIRIS para irradiar confocalmente una muestra material distribuida espacialmente y obtener espectros de fluorescencia de rayos X a partir de ésta, cuyas intensidades son posteriormente procesadas para generar mapas tridimensionales, que muestran la distribución espacial de los elementos químicos detectados dentro del volumen de estudio.

Además, se han implementado algoritmos para corregir los espectros tanto por eficiencia de detección así como también por picos de escape, obteniendo resultados similares a los de la literatura para el segundo caso.

Se realizaron múltiples pruebas con el software, demostrando que puede comunicarse correctamente con los diferentes componentes de OXIRIS, obtener espectros, procesarlos, y generar los mapas tridimensionales de la distribución espacial de las intensidades fluorescentes asociadas a uno o más elementos detectados.

Índice de Contenidos

Agradecimientos	VI
Resumen	VII
Índice de Tablas	XI
Índice de Figuras	XIII
Capítulo I Introducción	1
1.1 Objetivo General	3
1.2 Objetivos Específicos	3
Capítulo II Marco Teórico	4
2.1 Conceptos informáticos	4
2.1.1 Software y Hardware	4
2.1.2 Lenguaje de Programación	5
2.1.3 Periféricos	5
2.1.4 Protocolos de Comunicación	6
2.1.5 Interfaz Gráfica	6
2.2 Espectrometría de Fluorescencia de Rayos X	6
2.2.1 Emisión de Rayos X Característicos	6
2.2.2 Fuente de Radiación	8
2.2.3 Detección de Rayos X	9
2.2.4 Espectro de Fluorescencia de Rayos X	10
2.3 Características y Efectos Presentes en la Detección	11
2.3.1 Eficiencia de detección	11
2.3.2 Picos de Escape	13
2.4 Simulación Monte Carlo	13
Capítulo III Materiales y Método	15
3.1 Componentes de OXIRIS	15
3.1.1 Tubo de Rayos X	16
3.1.2 Detectores	17
3.1.3 Procesador Digital de Pulsos (DPP)	18
3.1.4 Sistema de Posicionamiento	19
3.1.5 Sistema de Rotación	20
3.1.6 Ordenador de Placa Reducida	21

3.2	Desarrollo del Software	22
3.2.1	Lenguaje de Programación y Librerías	22
3.2.2	Desarrollo Modular	23
3.2.3	Registro	25
3.2.4	Documentación del Código	26
3.2.5	Ejecución	27
3.3	Correcciones	27
3.3.1	Eficiencia del detector	28
3.3.2	Picos de Escape	29
Capítulo IV Resultados		33
4.1	Comunicación con los Periféricos	33
4.2	Interfaz gráfica	34
4.2.1	OXIRIS	34
4.2.2	Tubo de rayos X	35
4.2.3	DPP/Detectores	39
4.2.4	Posicionamiento	43
4.2.5	Rotación	45
4.2.6	Generador de rutinas	46
4.2.7	Ejecutor de rutinas	48
4.2.8	Analizador de espectros	50
4.2.9	Visor 3D	53
4.3	Correcciones de los Espectros EDXRF	56
4.3.1	Corrección por Eficiencia del Detector	56
4.3.2	Corrección por Picos de Escape	56
Capítulo V Conclusiones		58
5.1	Conclusiones Para el Primer Objetivo Específico	58
5.2	Conclusiones Para el Segundo Objetivo Específico	59
5.3	Conclusiones Para el Tercer Objetivo Específico	59
5.4	Conclusiones Para el Cuarto Objetivo Específico	60
5.5	Perspectivas Futuras	60
Financiamiento		61
Nomenclatura		62
Bibliografía		63
Anexo A	Tabla de Rayos X Característicos	65
Anexo B	Calibración de Espectros de Fluorescencia	69
Anexo C	Formato del Archivo de Calibración	70
Anexo D	Formato del Archivo de Parámetros	71
Anexo E	Formato del Archivo de Espectro	72

Anexo F	Protocolo de Comunicación del Sistema de Posicionamiento	74
Anexo G	Protocolo de Comunicación del Sistema de Rotación	77
Anexo H	Ejemplo de Rutina con Barrido Cúbico	80
Anexo I	Input para la Simulación de 50 keV	82
Anexo J	Geometría Utilizada en las Simulaciones	84

Índice de Tablas

2.1	Transiciones K y L del oro	7
A.1	Base de Datos de Rayos X Característicos	65

Índice de Figuras

1.1	Radiación convergente y divergente	2
2.1	Efecto fotoeléctrico y orbitales atómicos	7
2.2	Tubo de rayos X	8
2.3	Espectro de emisión de un tubo de rayos X	9
2.4	Detector de estado sólido	10
2.5	Espectro de fluorescencia de rayos X	11
2.6	Eficiencia de detección del detector XR-100T-CdTe	12
2.7	Picos de escape	13
3.1	Diagrama de OXIRIS	15
3.2	Tubo de rayos X portátil CP-120B y cono de radiación	16
3.3	Cono de radiación	17
3.4	Procesador de pulsos digitales PX5 y conjunto DP5/PC5	18
3.5	Sistema de posicionamiento con Arduino DUE	19
3.6	Sistema de rotación	20
3.7	Ordenador de placa reducida Raspberry Pi 4	21
3.8	Diagrama de dependencias del software	25
3.9	Página inicial de la documentación	26
3.10	Ventana de ingreso de contraseña	27
3.11	Eficiencia del detector XR-100T-CdTe	28
3.12	Geometría de la Simulación de Picos de Escape	30
3.13	Perfiles dosimétricos de las simulaciones Monte Carlo	30
3.14	Proceso y resultado de una simulación	31
3.15	Resultados de las simulaciones Monte Carlo	32
4.1	Pestaña OXIRIS	34
4.2	Ventanas para cambiar idioma y reiniciar	35
4.3	Pestaña del tubo de rayos X	35
4.4	Pestaña de DPP/detectores	40
4.5	Pestaña del sistema de posicionamiento	43
4.6	Pestaña del sistema de rotación	45
4.7	Pestaña del generador de rutinas	46
4.8	Ventana para generar barridos cúbicos	48
4.9	Pestaña del ejecutor de rutinas	49
4.10	Espectro de Co-57 en el Analizador de Espectros	50
4.11	Ventana de calibración	51
4.12	Construcciones 3D	54
4.13	Paletas de Color para los Gráficos 3D	55

4.14 Espectro de Co-57 corregido	56
4.15 Espectro Corregido por Picos de Escape	57

Capítulo I
INTRODUCCIÓN

Capítulo I

Introducción

Desde su descubrimiento en 1895 por Wilhelm Röntgen (Kaye, 1934), los rayos X han sido ampliamente utilizados en diversas áreas de la medicina, tanto para el diagnóstico como para el tratamiento de diferentes enfermedades. Y si bien la radiación ha demostrado tener efectos terapéuticos, también ha quedado en evidencia que causa muchos efectos adversos, tales como la aparición de hiperemia, destrucción escarótica, necrosis, gangrena, úlceras, tumores y cáncer (Mould, 1993).

Particularmente, para los pacientes de terapias de rayos X, existen dos fuentes de riesgo. Por un lado están los errores técnicos (Ford & Terezakis, 2010; Shafiq *et al.*, 2009), y por el otro el mecanismo mediante el cual se deposita la dosis requerida en el volumen de tratamiento. Por ejemplo, la radioterapia, que es aplicada al 52 % de los pacientes que tienen cáncer (Delaney *et al.*, 2005), puede conducir al origen de carcinomas (Suit *et al.*, 2007) debido a la irradiación no deseada de los tejidos sanos.

Una forma de disminuir el riesgo para los pacientes, ya sea en tratamiento o diagnóstico, consiste en dejar de utilizar haces de radiación divergentes, que irradian tanto el volumen de tratamiento como también los órganos de riesgo (figura 1.1(a)), y comenzar a utilizar conos convergentes compuestos de haces filiformes que operen con menor energía pero que depositen la misma dosis en el foco de convergencia, ubicado en el volumen de tratamiento, y una dosis inferior en los órganos de riesgo (figura 1.1(b)) (Figuroa & Valente, 2015; Figuroa *et al.*, 2016).

Esta idea se encuentra actualmente en desarrollo en el marco del proyecto FONDECYT 1171729 *Orthovoltage X-Ray Induced Radiation Integral System* (OXIRIS), dirigido por el Dr. Rodolfo Figuroa en el Centro de Física e Ingeniería en Medicina (CFIM) de la Universidad de La Frontera (UFRO).

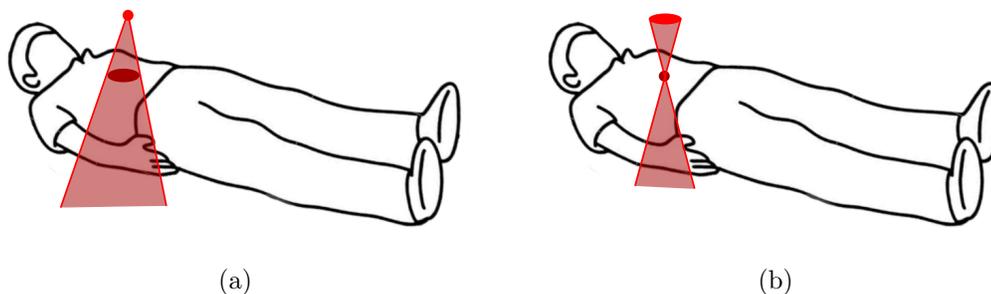


Figura 1.1: Distintas formas de irradiar un mismo objetivo. En a) se observa la típica configuración divergente, mientras que en b) se observa una configuración convergente que deposita menos dosis en los órganos de riesgo. (Elaboración propia).

Si bien el objetivo del proyecto OXIRIS es terapéutico, es decir, generar el diagnóstico del tumor y realizar el tratamiento de forma simultánea, en la etapa contemplada en este trabajo de título se apunta sólo al diagnóstico, que se realizará mediante la obtención de imágenes 3D a partir de espectros de fluorescencia de rayos X de tumores biomarcados con nanopartículas de oro (Ricketts *et al.*, 2013; Figueroa *et al.*, 2015; Guarda *et al.*, 2021).

Del proyecto previamente mencionado ya se ha construido el primer prototipo en el laboratorio búnker del CFIM, que reúne las características necesarias tanto para la generación del haz convergente así como también para la detección de la fluorescencia asociada.

El presente trabajo de título se abocó al desarrollo del software principal asociado al control operativo de este prototipo, logrando el objetivo de comunicarse con todo el equipo de OXIRIS, posibilitando así su funcionamiento sincronizado, controlando aspectos tales como el tiempo y energía de exposición, teniendo en cuenta los requerimientos de cada dispositivo. Además, dado que en el proceso de detección ocurren fenómenos que distorsionan el espectro fluorescente (Redus *et al.*, 2009), se programaron rutinas correctivas con el fin de realizar correcciones tanto por eficiencia del detector así como también por picos de escape.

Para desarrollar el software que se presenta en este trabajo se han requerido conocimientos de nivel avanzado de física de radiaciones, detectores, informática y electrónica, cuyas bases han sido adquiridas durante el transcurso de la carrera, y profundizadas gracias a las herramientas que entrega la misma.

Adicionalmente, este trabajo de título le ha dado continuidad a la práctica de estudios II, instancia en la que se concretaron múltiples pruebas de concepto relacionadas al desarrollo que así que presenta.

1.1 Objetivo General

Diseñar, desarrollar y optimizar un software capaz de trabajar coordinadamente con múltiples detectores de radiación, una fuente de rayos X convergente dinámica y un sistema de posicionamiento, lo que permitirá irradiar confocalmente un objetivo y obtener espectros de fluorescencia de rayos X distribuidos espacialmente con el fin de poder generar imágenes tridimensionales a partir de estos.

1.2 Objetivos Específicos

- Optimizar el desarrollo del software existente.
- Programar los módulos necesarios para establecer la comunicación con el hardware.
- Realizar correcciones con fundamento físico.
- Generar documentación del código.

Capítulo II
MARCO TEÓRICO

Capítulo II

Marco Teórico

Este trabajo se enmarca dentro de dos grandes áreas: la física y la informática; enfocándose principalmente en el desarrollo de una interfaz gráfica para controlar múltiples periféricos, pero además haciendo uso de simulaciones Monte Carlo, que, como se verá más adelante, fueron fundamentales para realizar la corrección por picos de escape.

La mezcla de estas dos áreas no es nueva, dado que en la actualidad la mayoría de los equipos utilizados en física operan de forma digital, por lo que necesitan un software para controlar adecuadamente su funcionamiento.

Desde el punto de vista físico, el concepto de espectrometría de fluorescencia de rayos X, así como también el funcionamiento asociado a los dispositivos para generar y detectar radiación, son fundamentales para este trabajo, ya que son la base del funcionamiento de OXIRIS. Esto, en conjunto con los conceptos informáticos de mayor relevancia utilizados, son expuestos a continuación.

2.1 Conceptos informáticos

2.1.1 Software y Hardware

Todo sistema electrónico se puede dividir en dos partes, software y hardware. El software hace referencia al conjunto de instrucciones que permite que el dispositivo funcione, mientras que el hardware se refiere a los componentes físicos del mismo.

El correcto funcionamiento de los dispositivos electrónicos se logra mediante una comunicación mutua entre ambos sistemas.

Por ejemplo, para escribir un documento en un computador, se necesita un teclado (hardware) que envía información a un programa informático (software), el que muestra lo que el usuario ha escrito por una pantalla (hardware).

2.1.2 Lenguaje de Programación

Un lenguaje de programación es un lenguaje que permite escribir una serie de órdenes que pueden ser leídas e interpretadas por un computador.

En función de la sintaxis del lenguaje éstos se clasifican en lenguajes de bajo nivel y de alto nivel. Los lenguajes de bajo nivel se caracterizan por ser muy complejos de programar pero sencillos de interpretar por las máquinas, mientras que los lenguajes de alto nivel son más amigables pero de interpretación más lenta.

Por otro lado, en función de cómo se procesan las instrucciones, los lenguajes se clasifican en compilados e interpretados. En los primeros la traducción a binario se realiza de una vez, mientras que en los segundos se lleva a cabo en tiempo de ejecución, instrucción por instrucción (Guttag, 2013).

La diferencia entre ambos es el tiempo que tarda el procesador en ejecutarlos. Un lenguaje compilado en general requiere de menos tiempo de procesamiento que un lenguaje interpretado.

Esto es crítico para softwares científicos intensivos como pueden ser simuladores, mientras que para interfaces gráficas en general no es relevante, ya que las demoras suelen ser inferiores a lo que el usuario tarda en interactuar con el programa.

2.1.3 Periféricos

Un periférico es cualquier pieza de hardware independiente que se puede conectar a un computador, ampliando así su funcionalidad original. Ejemplos de periféricos son teclados, mouses, impresoras, audífonos, etc.

Los periféricos tienen su propio procesador interno, que se encarga de realizar el trabajo mediante la ejecución de rutinas de bajo nivel.

La comunicación entre el computador y los periféricos se realiza mediante interfaces de comunicación, dentro de las que se encuentran, por ejemplo, la interfaz Bluetooth para comunicación inalámbrica, y la USB y serial para comunicación alámbrica.

2.1.4 Protocolos de Comunicación

Cada periférico cuenta con un protocolo de comunicación donde se especifican las distintas instrucciones y respuestas en bits que puede recibir y enviar desde y hacia un computador. Por ejemplo, si el periférico es una linterna, el protocolo de comunicación puede ser de la siguiente forma:

```
00000000 : Encender linterna .  
00000001 : Modo intermitente .  
00000011 : Apagar linterna .
```

Así, si el computador envía la secuencia de bits 00000000 al periférico, este mediante el uso de su procesador interno ejecuta la instrucción de encenderse.

2.1.5 Interfaz Gráfica

Una interfaz gráfica (GUI) es un software que actúa como interfaz de comunicación entre un usuario y un computador. Es la forma más sencilla e intuitiva de utilizar un computador, pues se vale de elementos gráficos, como botones, diálogos e imágenes, para comunicarse con el usuario.

2.2 Espectrometría de Fluorescencia de Rayos X

La espectrometría de fluorescencia de rayos X consiste en la medición de la intensidad de los rayos X emitidos desde un cuerpo como función de su energía o longitud de onda (Bounakhla & Tahri, 2007).

Se distinguen claramente dos etapas: emisión y detección, las que se detallan a continuación.

2.2.1 Emisión de Rayos X Característicos

Cuando un fotón de rayos X choca con un electrón de una capa interna de un átomo este puede ser eyectado de su orbital, fenómeno que se conoce como efecto fotoeléctrico (figura 2.1(a)), y que solo ocurrirá si la energía del fotón es superior a la energía de ligadura del estado atómico.

Tras la ocurrencia del efecto fotoeléctrico un electrón de un orbital superior decaerá para ocupar la vacancia generada, emitiendo un fotón de rayos X característico, proceso que es conocido como fluorescencia de rayos X (XRF).

La energía de este fotón es propia del elemento químico en el cual ha sido emitido, de ahí que sea llamado característico.

Estas vacancias se llenan siguiendo las reglas de selección cuánticas (Eisberg, 1978), por lo que no todas las transiciones están permitidas. Las que sí están permitidas se nombran anteponiendo la letra del orbital al que decaen, seguido de una letra griega y un subíndice.

Tal como se observa en la figura 2.1(b), los átomos tienen distintos orbitales, definidos mediante el número cuántico principal n . Particularmente, los orbitales con $n = 1, 2$ y 3 se llaman orbitales K, L y M, respectivamente.

La capa K suele ser mucho menos energética que la L y M, por lo que un salto hacia esta capa emite un fotón muy energético (tabla 2.1). En general, tanto los fotones de transiciones K como L son fácilmente detectables en un espectro de fluorescencia de rayos X.

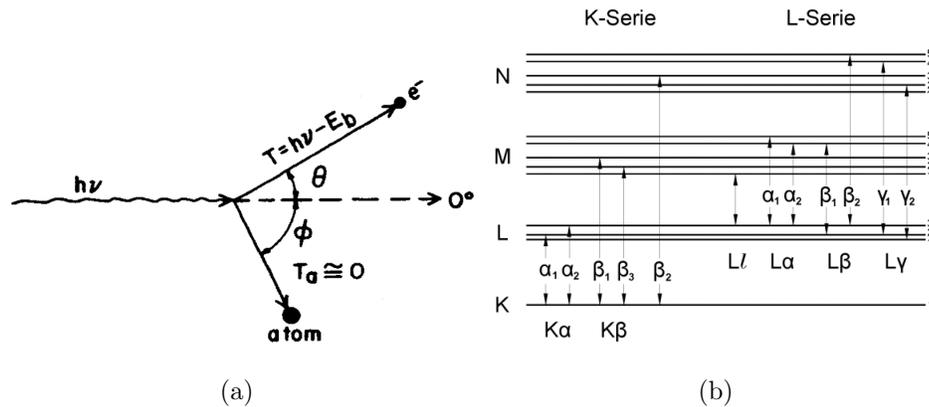


Figura 2.1: En a) se observa un esquema del efecto fotoeléctrico, donde se muestra que en la interacción interviene el fotón, el electrón y el átomo, aunque este último casi no se ve afectado debido a su masa (imagen extraída y modificada de Attix (2004)). En b) se observa un diagrama energético de los orbitales correspondientes a los primeros cuatro valores de n , junto con las transiciones permitidas (imagen extraída y modificada de www.ld-didactic.de/literatur/hb/e/p6/p6355_e.pdf).

Tabla 2.1: Transiciones K y L del oro. Se nota que las transiciones K son más energéticas que las transiciones L (datos obtenidos del NIST (Hudson, 2003)).

Transición	$K_{\alpha_1} (L_3 \rightarrow K)$	$K_{\alpha_2} (L_2 \rightarrow K)$	$L_{\alpha_1} (M_5 \rightarrow L_3)$	$L_{\alpha_2} (M_4 \rightarrow L_3)$
Energía [keV]	68.8	66.9	9.7	9.6

2.2.2 Fuente de Radiación

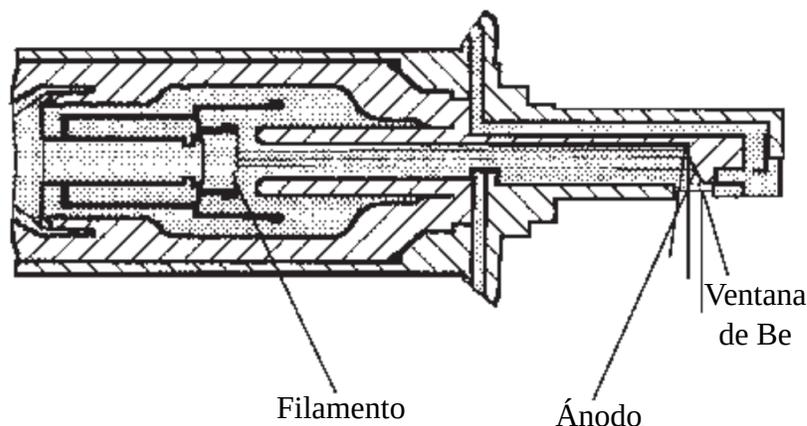


Figura 2.2: Esquema de una sección transversal de un tubo de rayos X, donde se aprecia el espacio en el que los electrones aceleran desde el filamento hasta el ánodo. (imagen extraída y traducida de Lindon *et al.* (2010)).

Los rayos X utilizados para generar la vacancia descrita en la sección anterior son comúnmente generados mediante un tubo de rayos X, que consiste en un cilindro vacío dentro del cual se encuentran un cátodo y un ánodo. El cátodo, mediante efecto termoiónico, libera electrones, los cuales son acelerados por una diferencia de potencial hasta el ánodo (figura 2.2). Estos electrones pueden interactuar con el campo eléctrico de los núcleos atómicos del ánodo, frenándose y emitiendo la pérdida de energía cinética como radiación electromagnética, llamada *bremsstrahlung* (Attix, 2004). Por otro lado, si la energía del electrón incidente supera a la energía de ligadura de los electrones de un orbital interno dado, entonces pueden interactuar liberando fotones característicos del ánodo. Así, el espectro de rayos X de un tubo de rayos X consiste en una superposición de ambas interacciones (Lindon, 2010) (figura 2.3). La energía máxima de un fotón proveniente del tubo (E) es el producto entre la carga del electrón (e) y la diferencia de potencial aplicada entre el cátodo y el ánodo (V):

$$E = eV \quad (2.1)$$

Esta unidad se mide en Joule, no obstante en física atómica y nuclear suele utilizarse el electronvoltio (eV), que corresponde a la energía cinética adquirida por un electrón al acelerarse con una diferencia de potencial de 1V. Por ejemplo, si un tubo opera a 100 kV, entonces la energía máxima de un fotón emitido por el tubo será de 100 keV.

Por otro lado, la intensidad de la radiación, es decir, la cantidad de fotones emitidos, depende de la corriente aplicada al tubo, ya que una corriente superior incrementará la tasa de incidencia del efecto termoiónico, y por el contrario una corriente inferior lo disminuirá.

La radiación que producen los tubos que funcionan entre 100 y 1000 kV se llama radiación de ortovoltaje, mientras que la radiación producida por tubos que funcionan por sobre 1 MV se llama radiación de megavoltaje.

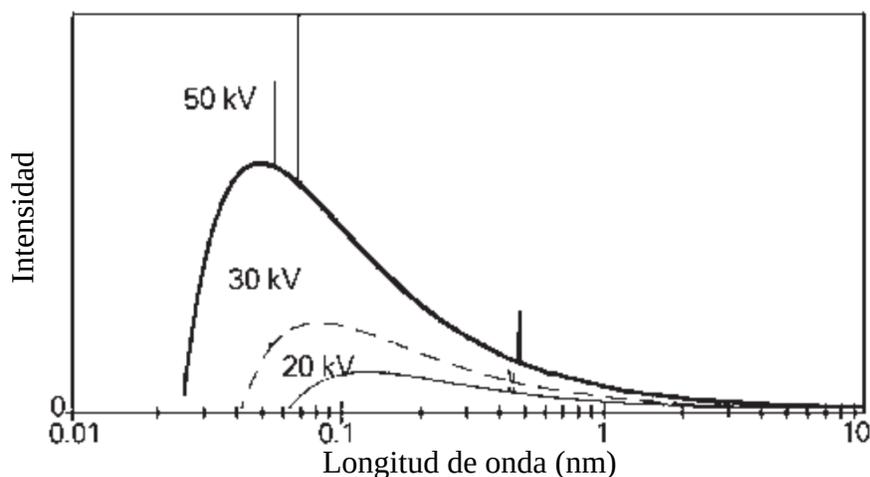


Figura 2.3: Espectro de emisión típico para tres voltajes distintos (50, 30 y 20 kV) de un tubo con ánodo de rodio, donde se distinguen claramente los picos característico del mismo (imagen extraída y traducida de Lindon *et al.* (2010)).

2.2.3 Detección de Rayos X

La fluorescencia de rayos X se divide en dos grupos en función de cómo se detecte: Fluorescencia de Rayos X Dispersiva por Energía (EDXRF), y Fluorescencia de Rayos X Dispersiva por Longitud de Onda (WDXRF).

En la primera el detector mide directamente la energía de los fotones emitidos, mientras que en la segunda se utiliza la ley de Bragg para medir la longitud de onda asociada.

Si bien el segundo método posee mayor resolución, es más caro y lento, pues necesita más tiempo debido a que realiza mediciones punto por punto en el espectro, a diferencia de la técnica de EDXRF que es capaz de detectar y procesar todo el espectro de forma simultánea. En el caso de EDXRF se cuenta con varios tipos de detectores, por ejemplo los de estado sólido y los de centelleo. De todos, los que presentan el mejor equilibrio entre resolución de energía y eficiencia de detección en los rangos energéticos de interés para OXIRIS (transiciones $K\alpha_1$ y $K\alpha_2$ del oro) son los de estado sólido, por lo que son los utilizados en el proyecto.

Estos detectores están compuestos por un cátodo, un ánodo, un semiconductor y un amplificador, tal como se observa en la figura 2.4.

Cuando la radiación ionizante pasa a través del semiconductor se generan pares electrón-hueco, generando una concentración de carga proporcional a la energía depositada en el medio (Knoll, 2010). Estos pares electrón-hueco son desviados hacia el cátodo y ánodo mediante la aplicación de una diferencia de potencial, generando así un pulso eléctrico que puede ser detectado, amplificado y procesado, cuya intensidad de corriente estará directamente relacionada a la energía incidente de la radiación.

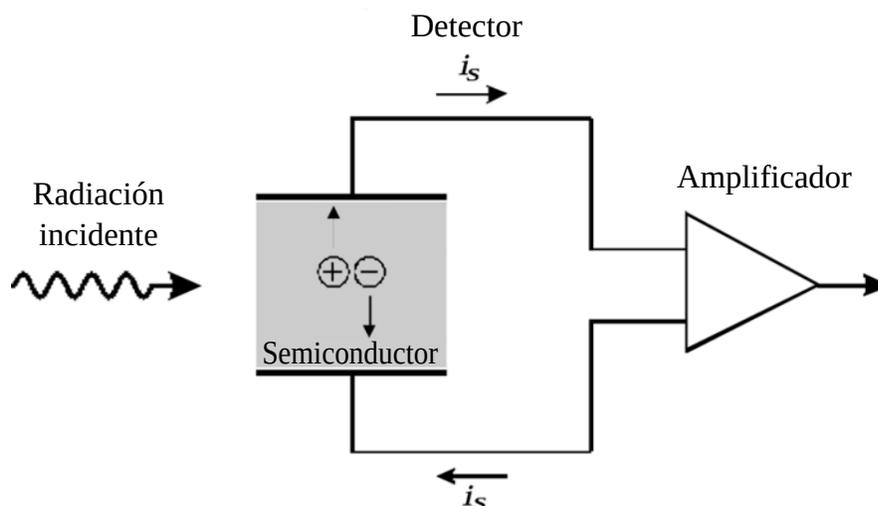


Figura 2.4: Esquema de un detector de estado sólido utilizado en EDXRF, donde se observa el semiconductor dentro del cual se generan los pares electrón-hueco, responsables del pulso eléctrico que posteriormente es amplificado y procesado (imagen extraída y modificada de www.desy.de/~niebuhr/Vorlesung/Detektor/Vorlesung_4.pdf).

Los detectores son utilizados junto con un procesador digital de pulsos (DPP), los que están optimizados para realizar algunas operaciones específicas a muy alta velocidad, tales como contar y clasificar todos los pulsos provenientes del circuito del detector.

En la práctica toda la interacción del usuario con el detector se realiza a través de un DPP.

2.2.4 Espectro de Fluorescencia de Rayos X

Al irradiar un material y detectar la fluorescencia emitida por éste se obtiene un espectro de fluorescencia de rayos X. Estos espectros contienen la cantidad de fotones detectados (cuentas) en el eje y y la energía asociada a cada uno en el eje x , tal como se observa en la figura 2.5.

Están compuestos tanto por los rayos X característicos así como también por otros tipos de radiaciones. Por ejemplo, en la figura 2.3 se observa un espectro compuesto por la radiación característica del rodio y por el bremsstrahlung del tubo de rayos X.

Estos espectros pueden ser analizados, proceso que consiste en identificar los picos y su energía, para luego buscar en una base de datos a qué transición y elemento químico corresponden, lo permite identificar los elementos químicos presentes en el material irradiado.

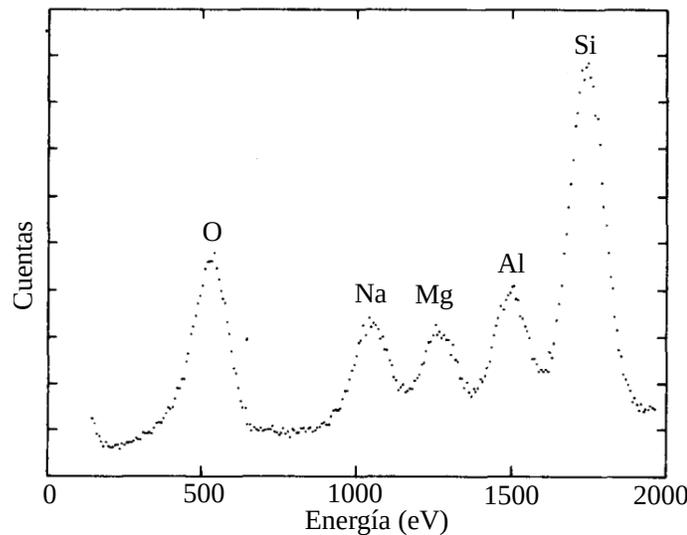


Figura 2.5: Espectro de fluorescencia de rayos X del mineral glaucofana ($\text{Na}_2(\text{Mg}_3\text{Al}_2)\text{Si}_8\text{O}_{22}(\text{OH})_2$). Se observan claramente todos los picos $K\alpha$ de los elementos químicos que componen el mineral (imagen extraída y modificada de Jaklevic y Goulding (1971)).

2.3 Características y Efectos Presentes en la Detección

En el proceso de detección ocurren fenómenos que distorsionan el espectro detectado. De ellos, tanto la eficiencia de detección como también los picos de escape son de interés para este trabajo, por lo que se describen a continuación.

2.3.1 Eficiencia de detección

Los rayos X se atenúan al atravesar un medio debido a diversos fenómenos, lo que se puede representar mediante el coeficiente de atenuación lineal $\mu(E)$.

Una forma empírica de modelar este comportamiento de forma aproximada corresponde a la ley de Beer-Lambert, llamada también ley de atenuación exponencial (Attix, 2004), dada por

$$P(x, E) = e^{-\mu(E)x} \quad (2.2)$$

donde $P(x)$ corresponde a la probabilidad de que el fotón no interactúe con el medio en función de la distancia recorrida x y de la energía inicial del fotón E .

Esta expresión también se puede escribir en función del coeficiente de atenuación másico μ/ρ , que corresponde al coeficiente de atenuación lineal dividido por la densidad del medio:

$$P(x, E) = e^{-\frac{\mu(E)}{\rho}\rho x} \quad (2.3)$$

Para que un fotón sea detectado éste debe interactuar con el detector, lo que se puede modelar mediante la ecuación 2.3, cuyo resultado recibe el nombre de eficiencia de detección (figura 2.6).

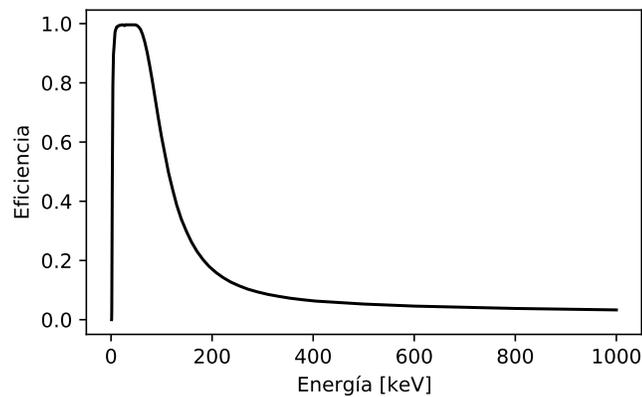


Figura 2.6: Eficiencia de detección del detector XR-100T-CdTe, utilizado en este trabajo, donde se observa la probabilidad de interacción total en función de la energía del fotón. Los datos fueron obtenidos desde la página web de AmpTek.

Los detectores de estado sólido mencionados en la sección 2.2.3 poseen una ventana (típicamente de berilio) antes del semiconductor. De esta forma, para que la radiación sea detectada, primero debe atravesar la ventana (v):

$$P_v = e^{-\left(\frac{\mu}{\rho}\right)_v \rho_v x_v}$$

y luego interactuar con el semiconductor (s):

$$P_s = 1 - e^{-\left(\frac{\mu}{\rho}\right)_s \rho_s x_s}$$

Por lo tanto, la probabilidad de interacción se puede modelar como

$$\begin{aligned}
 P &= P_v P_s \\
 P &= e^{-\left(\frac{\mu}{\rho}\right)_v \rho_v x_v} \left(1 - e^{-\left(\frac{\mu}{\rho}\right)_s \rho_s x_s}\right)
 \end{aligned}
 \tag{2.4}$$

2.3.2 Picos de Escape

Hay fotones que ingresan al detector con energía suficiente para generar rayos X característicos de los elementos que componen el semiconductor. Cuando estos fotones característicos interactúan y generan pares electrón-hueco dentro del volumen del detector no hay problema. No obstante si estos fotones fluorescentes escapan del detector se terminan generando picos de menor energía respecto a la energía real incidente debido a la menor carga recolectada, lo que se conoce como picos de escape (Knoll, 2010) (figura 2.7).

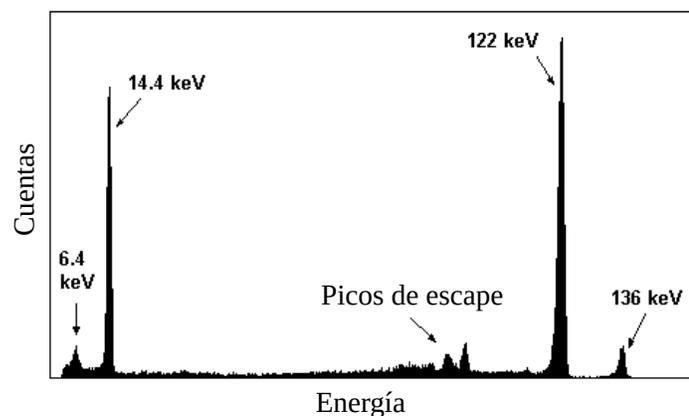


Figura 2.7: Picos de escape para el detector AmpTek XR-100T-CZT producidos por el decaimiento del cobalto-57 (imagen extraída y modificada de www.ndtnet.com/m/amptek/xr100czt.html).

2.4 Simulación Monte Carlo

Cuando no se puede modelar analíticamente un fenómeno físico mediante su formulación teórico-matemática se recurre a diversos métodos numéricos, dentro de los que se encuentran los métodos Monte Carlo, que simulan el proceso utilizando distribuciones de probabilidad, cuyo origen puede ser teórico o empírico.

Para la interacción de la radiación con la materia existen muchos simuladores Monte Carlo. Particularmente, debido al conocimiento previo del software, en las simulaciones realizadas en

este trabajo se utilizó PENELOPE (Baró *et al.*, 1995), el que trabaja simulando la interacción de electrones, positrones y fotones en materiales definidos por el usuario, en un rango de energía desde 50 eV hasta 1 GeV.

Capítulo III

MATERIALES Y MÉTODO

Capítulo III

Materiales y Método

3.1 Componentes de OXIRIS

El primer prototipo del proyecto OXIRIS está compuesto por un tubo de rayos X, 4 detectores de radiación, 4 procesadores digitales de pulso (DPP), múltiples colimadores, un sistema de posicionamiento, un sistema de rotación y un computador, tal como se puede observar parcialmente en el esquema de la figura 3.1.

Todas estas piezas son necesarias para generar la radiación confocal, barrer un objetivo, recolectar los datos de fluorescencia y luego hacer la construcción de la imagen 3D.

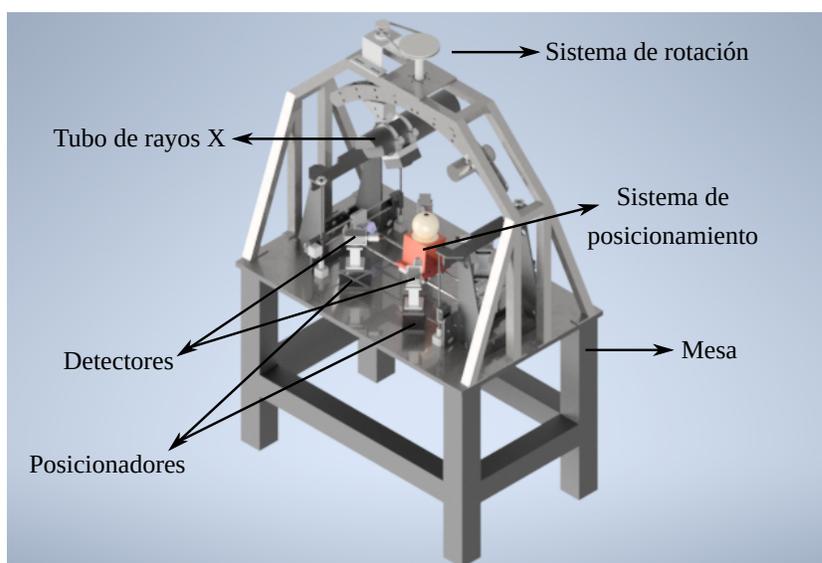


Figura 3.1: Diagrama referencial del primer prototipo de OXIRIS, donde se pueden ver algunos de sus componentes (imagen facilitada por Jorge Leiva, correspondiente a la solicitud de patente PCT/IB2020/05838).

A continuación se detallan los componentes de mayor importancia para este trabajo de título.

3.1.1 Tubo de Rayos X

La fuente de radiación utilizada en OXIRIS corresponde al tubo de rayos X portátil CP-120B producido por Teledyne ICM (figura 3.2(a)).

Sus principales características son:

- Voltaje de aceleración: 40 kV a 120 kV.
- Corriente de ánodo: 0.1 mA a 1.0 mA.
- Potencia máxima en el ánodo: 120 W.
- Objetivo de tungsteno.
- Disparo en modo continuo.
- Ángulo de emisión: cono de 50° de apertura.
- Punto de emisión focal: 0.8 x 0.5 mm².
- Operación remota mediante Bluetooth.

El tubo se encuentra unido a una plataforma giratoria (sección 3.1.5), de tal forma que haciendo uso de la comunicación Bluetooth el giro forma un cono de radiación cuyo foco corresponde a la zona objetivo (figura 3.2(b)).

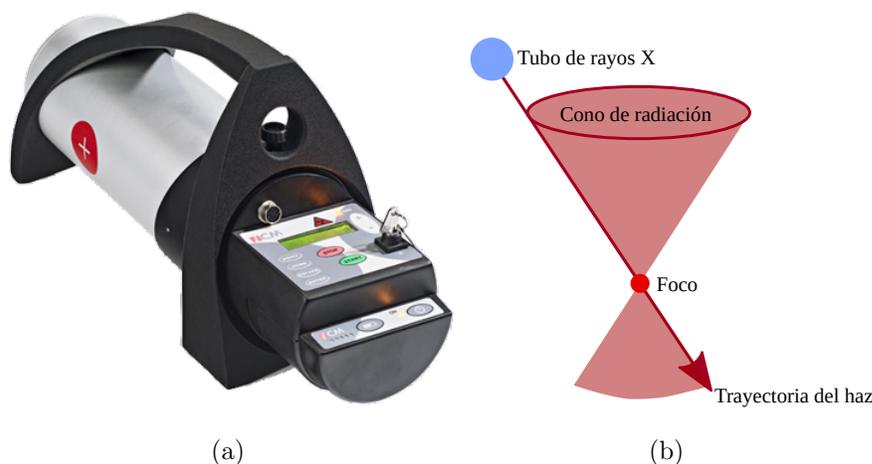


Figura 3.2: En a) se observa el Tubo de rayos X portátil CP-120B (imagen extraída de www.teledyneicm.com/products/ndt/cp120b) y en b) el cono de radiación producido cuando éste gira debido al sistema de rotación (elaboración propia en base a la figura 3.1.3 de la tesis de magíster de Jaime Guarda (2020)).

Protocolo de comunicación y software original

El tubo de rayos X es distribuido con un software propietario que sólo funciona en Windows y que permite una operación limitada del mismo, pues no tiene implementadas todas las funcionalidades que posee el equipo.

Por otro lado, el protocolo de comunicación no es distribuido con la compra del producto ni tampoco se encuentra disponible públicamente, no obstante se ha logrado obtener mediante comunicación directa con el equipo de Teledyne ICM.

Este protocolo es común para dos modelos de la misma marca, el CP120B y el CP160B, por lo que el software desarrollado es compatible con ambos.

Por último, dado que el protocolo es un archivo privado y que no se cuenta con autorización por escrito de Teledyne ICM, no se puede adjuntar ni hacer mayor referencia al mismo en el cuerpo de este documento.

3.1.2 Detectores

Se utilizan 4 detectores de estado sólido de telurio de cadmio (CdTe) modelo XR-100CdTe producidos por AmpTek (figura 3.3).

Sus principales características son:

- Área de detección: 5 mm x 5 mm .
- Espesor del detector: 1 mm.
- Máxima tasa de conteo por segundo: $\sim 50 \cdot 10^3$.
- Ventana de berilio de 100 μm de espesor.

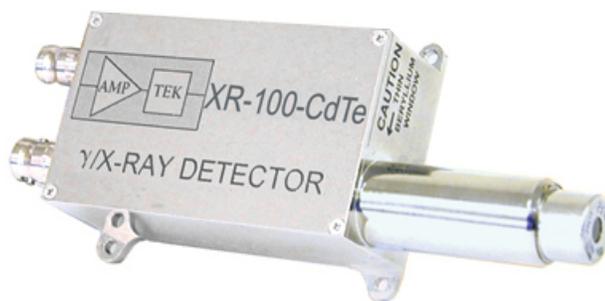
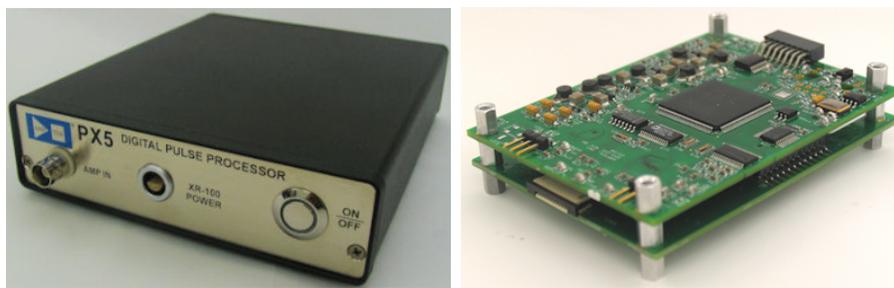


Figura 3.3: Detector XR-100CdTe de AmpTek (imagen extraída de www.amptek.com/products/x-ray-detectors/cdte-x-ray-and-gamma-ray-detectors/xr-100cdte-x-ray-and-gamma-ray-detector).

3.1.3 Procesador Digital de Pulsos (DPP)

Se utilizan dos tipos de DPP, el modelo PX5 (figura 3.4(a)), y el conjunto DP5/PC5 (figura 3.4(b)), ambos de la marca AmpTek.

El modelo PX5 puede ser utilizado con una amplia variedad de detectores, mientras que el DP5/PC5 ha sido configurado para trabajar solamente con los detectores XR-100CdTe.



(a)

(b)

Figura 3.4: En a) se observa el procesador de pulsos digitales PX5 (imagen extraída de www.amptek.com/products/digital-pulse-processors/px5-digital-pulse-processor) y en b) el conjunto DP5/PC5, en donde la placa superior corresponde a la DP5, y la inferior a la PC5 (imagen extraída de www.amptek.com/products/digital-pulse-processors/dp5-digital-pulse-processor-and-mca).

Sus principales características son:

- Amplificación digital de conformación de pulsos.
- Analizador multicanal integrado.
- Fuentes de alimentación para los detectores incorporadas.
- Rechazo de apilamiento de pulsos.

Protocolo de comunicación y software original

La comunicación con los detectores se hace mediante los DPP, por lo que sólo hace falta conocer el protocolo de comunicación de éstos últimos.

Ambos DPP utilizados en OXIRIS cuentan con un protocolo de comunicación casi totalmente compartido, que no se distribuye con el producto pero que se puede encontrar en la página web del laboratorio Donald A. Glaser de la Universidad de Berkeley (experimentationlab.berkeley.edu/sites/default/files/images/DP5_Programmers_Guide_A4.pdf).

Leer el protocolo de comunicación es necesario para entender el desarrollo del software, pero debido a su extensión (140 páginas) no se adjunta como anexo.

Además AmpTek pone a disposición del usuario un software propietario para la adquisición de espectros que funciona en Windows, así como también un kit de desarrollo multiplataforma desde donde se pueden examinar ejemplos básicos escritos en C (www.amptek.com/software/software-downloads/dp5-digital-pulse-processor-software).

3.1.4 Sistema de Posicionamiento

El sistema de posicionamiento fue desarrollando durante el trabajo de título de Eduardo Quilagüilque (2019), y consiste en un sistema similar al de una impresora 3D, con motores paso a paso de 200 pasos por vuelta, y codificadores que permiten mover una camilla en los ejes XYZ (figura 3.5(a)).

El procesador del posicionador corresponde a un microcontrolador Arduino DUE (figura 3.5(b)), que contiene un único núcleo, por lo que permite mover un motor a la vez.

El trabajo continuo y prolongado de la camilla genera una gran cantidad de calor en el circuito de los motores, por lo que el sistema tiene integrado un ventilador para mantener la temperatura estable.

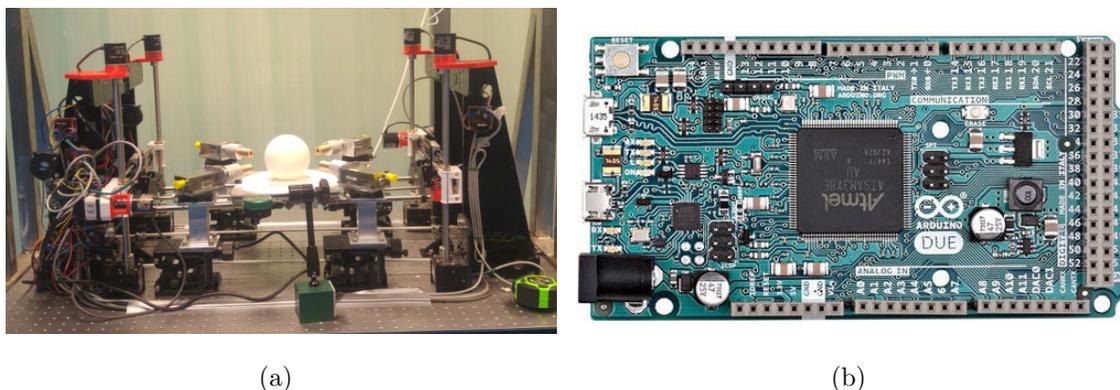


Figura 3.5: En a) se observa una fotografía del sistema de posicionamiento, con la camilla en el centro, conteniendo un volumen esférico sobre ella (fotografía hecha por Jaime Guarda). En b) se puede ver el microcontrolador Arduino DUE, con dimensiones de 10 x 5 cm aproximadamente (fotografía obtenida desde arduino.cl/producto/arduino-due).

Protocolo de comunicación

El protocolo de comunicación del sistema de posicionamiento se desarrolló junto con el hardware, no obstante es necesario hacerle modificaciones para asegurar la precisión de los movi-

mientos.

Debido a las condiciones actuales de pandemia esto no se ha podido llevar a cabo, por lo que se ha optado por escribir un nuevo protocolo de comunicación (Anexo F) que solucione los problemas previamente existentes, y que pueda ser implementado en el hardware una vez que se reanuden las actividades presenciales.

3.1.5 Sistema de Rotación

El sistema de rotación fue desarrollado en el CFIM y consiste en un motor que mediante una polea transmite la rotación a una pieza mecánica que sujeta el tubo de rayos X así como también un contrapeso, alcanzando velocidades de hasta 180 RPM (figura 3.6).

Esta pieza es libre de girar en los 360° del plano, y permite distintas configuraciones del tubo, para así poder generar el punto confocal a distintas alturas.

Actualmente el sistema de rotación no está totalmente instalado por lo que no es funcional.

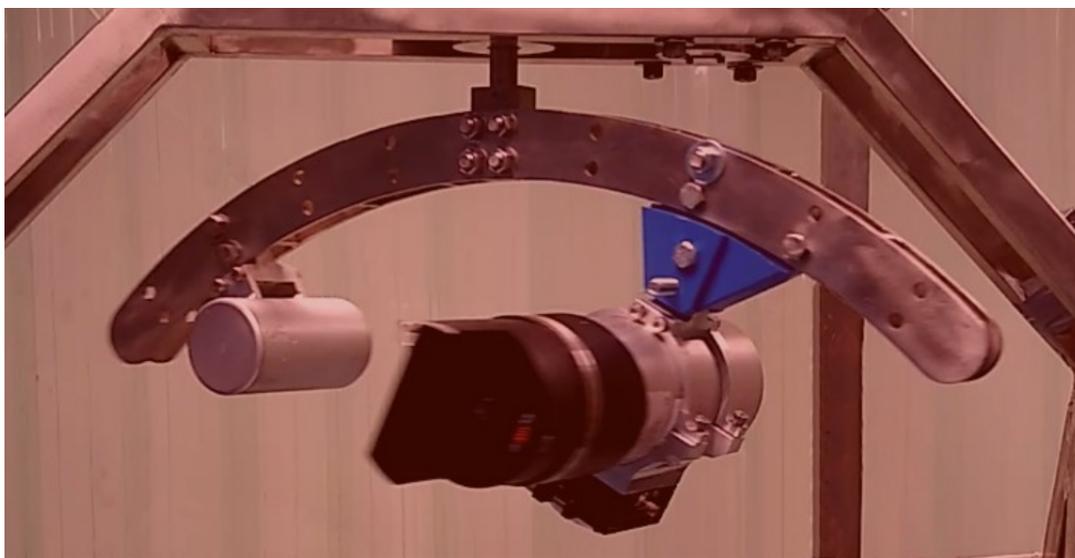


Figura 3.6: Fotografía del sistema de rotación funcionando durante una prueba, con el tubo de rayos X apagado (fotografía hecha por Jaime Guarda).

Protocolo de comunicación

El sistema de rotación no tenía ninguna especificación de software, por lo que en este trabajo se ha generado un protocolo de comunicación para el mismo (Anexo G). El sistema objetivo de este protocolo es un Arduino, al igual que para el sistema de posicionamiento.

3.1.6 Ordenador de Placa Reducida

El software desarrollado se debe ejecutar en un computador, que tenga los puertos y recursos necesarios para comunicarse con los periféricos previamente descritos.

Es por ello que se decidió utilizar el ordenador de placa reducida Raspberry Pi 4 (figura 3.7) fabricado por la Raspberry Pi Foundation, que contiene 4GB de RAM y 4 puertos USB, además de un procesador ARM de 4 núcleos modelo Cortex-A72 que opera a 1.5 GHz. Las características anteriores son suficientes para ejecutar el software de forma fluida, además por su tamaño (85.6 x 56.5 mm) es fácilmente integrable a la caja electrónica del sistema de posicionamiento, donde ya hay un ventilador instalado, por lo que no es necesario un sistema de refrigeración adicional.

Este ordenador funciona con sistemas operativos basados en Linux, admitiendo una gran variedad de distribuciones. Específicamente, se utilizó la distribución Raspberry Pi OS, que está basada en la distribución GNU/Linux Debian, y que se puede descargar gratuitamente desde su página web.

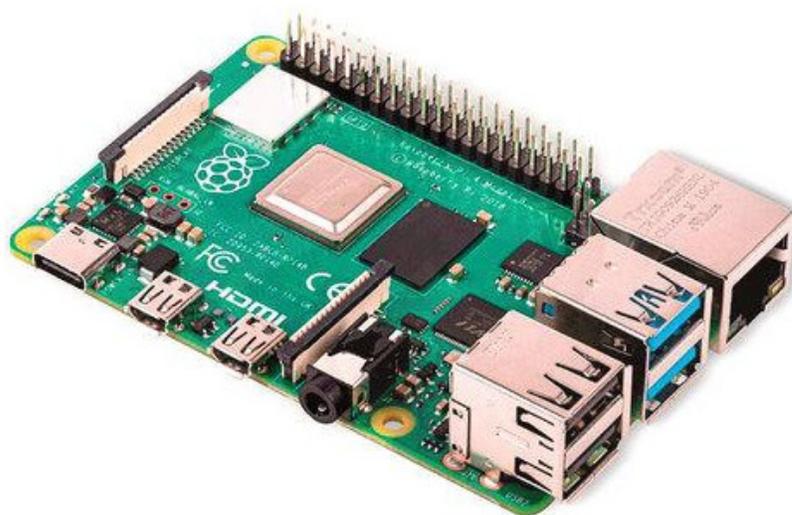


Figura 3.7: Fotografía del ordenador de placa reducida Raspberry Pi 4, donde se observan los puertos USB responsables de establecer la comunicación con los periféricos previamente expuestos (imagen obtenida desde www.pcfactory.cl/producto/34255-raspberry-pi-raspberry-pi-4-modelo-b-4gb-ram).

3.2 Desarrollo del Software

El desarrollo del software comenzó en la Práctica de Estudios II, realizando modificaciones a los ejemplos que provee AmpTek, así como también llevando a cabo pruebas de comunicación Bluetooth para establecer el vínculo con el tubo de rayos X.

Durante estas pruebas las comunicaciones se realizaban mediante escritura/lectura directa de los puertos USB y serial, lo que conducía a múltiples errores.

Tras incorporar las librerías `pyserial` y `pyusb` se pudo mejorar la comunicación con los periféricos mediante rutinas más sólidas, lo que permitió comenzar a desarrollar la interfaz gráfica.

Si bien durante esta etapa se consiguieron resultados importantes, el código era poco legible y no seguía una estructura modular, lo que hacía difícil su modificación.

De acuerdo a los objetivos planteados en este proyecto de título, se tomó el desarrollo previamente expuesto y se re-escribió completamente, revisando y mejorando todas las rutinas ya existentes, y añadiendo muchas más con el fin de generar un software completo y funcional, tal como se muestra a lo largo de esta sección.

3.2.1 Lenguaje de Programación y Librerías

Con el fin de dar cumplimiento a los objetivos de optimizar el software existente así como también generar los nuevos módulos necesarios, y considerando el tiempo reducido para el desarrollo de este proyecto, se optó por trabajar con Python 3.8, ya que permite escribir rutinas complejas de forma rápida y eficiente.

Python, al ser un lenguaje interpretado, es mucho más lento que otros disponibles, sin embargo esto no representa un problema, pues hay pocas rutinas de cálculo intensivo, las que en todo caso se ejecutan de forma optimizada en hilos separados, por lo que no hay latencias perceptibles para el usuario.

Las librerías utilizadas, además de las que vienen por defecto con Python 3.8, son:

- `numpy`: cálculo numérico.
- `numba`: optimización del cálculo numérico.
- `matplotlib`: definiciones de color para los gráficos 3D.
- `scipy`: interpolación de datos.
- `PyOpenGL`: renderizado por GPU para los gráficos 3D.
- `PyQt5`: librería gráfica.
- `PyQtGraph`: librería para incorporar gráficos 2D/3D a la interfaz gráfica.

- `pyserial`: comunicación serial.
- `pyusb`: comunicación por puerto USB.

Debido a que el sistema operativo utiliza el núcleo Linux, el software se desarrolló para ejecutarse principalmente sobre esta plataforma. No obstante algunas funciones están disponibles también para Windows, con el fin de permitir que el procesamiento de datos no esté limitado sólo a la Raspberry Pi, sino que el usuario pueda hacerlo donde estime conveniente.

En el caso de que el software se ejecute sobre una plataforma Linux todas las librerías listadas anteriormente se deben encontrar instaladas en el sistema, lo que se puede hacer mediante el gestor de paquetes `pip`. En el caso de Windows, las librerías `pyserial` y `pyusb` no son necesarias, pues el software no está diseñado para comunicarse con los periféricos desde este sistema operativo.

Debido a problemas de compatibilidad con el software la librería `PyQtGraph` ha sido modificada y se distribuye junto con el código fuente, por lo que no debe ser instalada por el usuario, ni en Linux ni en Windows.

Adicionalmente, para los sistemas Linux, se necesita instalar el paquete `bluez-tools`, disponible en los repositorios oficiales de las principales distribuciones. Su función es establecer la comunicación Bluetooth mediante un puerto serial virtual, lo que es expuesto en mayor detalle en la sección 4.2.2.

3.2.2 Desarrollo Modular

Con el fin de facilitar el desarrollo del software se optó por una metodología de programación modular, lo que hace más fácil aislar partes que no están relacionadas entre sí, y agrupar las que sí lo están, permitiendo realizar correcciones y ampliaciones de funcionalidad de manera sencilla.

Para lograr esta estructura modular se escribieron un total de 102 clases y 454 funciones, repartidas en 17 archivos. Estos archivos integran tanto las pruebas de concepto desarrolladas durante la Práctica de Estudios II, así como también las nuevas rutinas generadas para optimizar y complementar aquel desarrollo previo.

Los 17 archivos principales, relacionados entre sí según la figura 3.8, son los siguientes:

- `elements.py`: Contiene las principales transiciones de los elementos químicos, desde el neón hasta el fermio. Los datos fueron obtenidos desde la base de datos de energías de transiciones de rayos X del NIST (Hudson, 2003), y se adjuntan en el Anexo A.
- `calibration.py`: Contiene las rutinas para generar y leer archivos de calibración.

- `spectrum.py`: Contiene las rutinas para leer archivos de espectro, y guardar sus modificaciones.
- `qt_extended.py`: Contiene múltiples clases que añaden funcionalidades a las presentes en PyQt5.
- `general_functions_classes.py`: Contiene funciones de carácter general, tales como las encargadas de generar registros de todo lo que sucede en el software.
- `serial_comm.py`: Contiene rutinas para realizar y facilitar la comunicación serial proporcionada por `pyserial`.
- `password_window.py`: Contiene la interfaz gráfica para ingresar y validar la contraseña de usuario, que es necesaria en los sistemas Linux para realizar la comunicación USB.
- `OXIRIS.py`: Ejecuta el programa principal.
- `presentation_window.py`: Pestaña inicial, donde se muestra el nombre del software y las instituciones asociadas.
- `x_rays_tube_window.py`: Pestaña encargada de realizar la comunicación con el tubo de rayos X.
- `detectors_window.py`: Pestaña encargada de realizar la comunicación con los DPP.
- `positioning_window.py`: Pestaña encargada de realizar la comunicación con el sistema de posicionamiento.
- `rotation_window.py`: Pestaña encargada de realizar la comunicación con el sistema de rotación.
- `routine_generator_window.py`: Pestaña encargada de generar rutinas de operación.
- `routine_executor_window.py`: Pestaña encargada de leer las rutinas de operación y ejecutarlas.
- `analyzer_window.py`: Pestaña con un completo analizador de espectros, desde donde se pueden realizar calibraciones y correcciones por eficiencia del detector y por picos de escape.
- `viewer_window.py`: Pestaña encargada de realizar la construcción de las imágenes tridimensionales y mostrar los mapas de intensidad fluorescente de los elementos químicos.

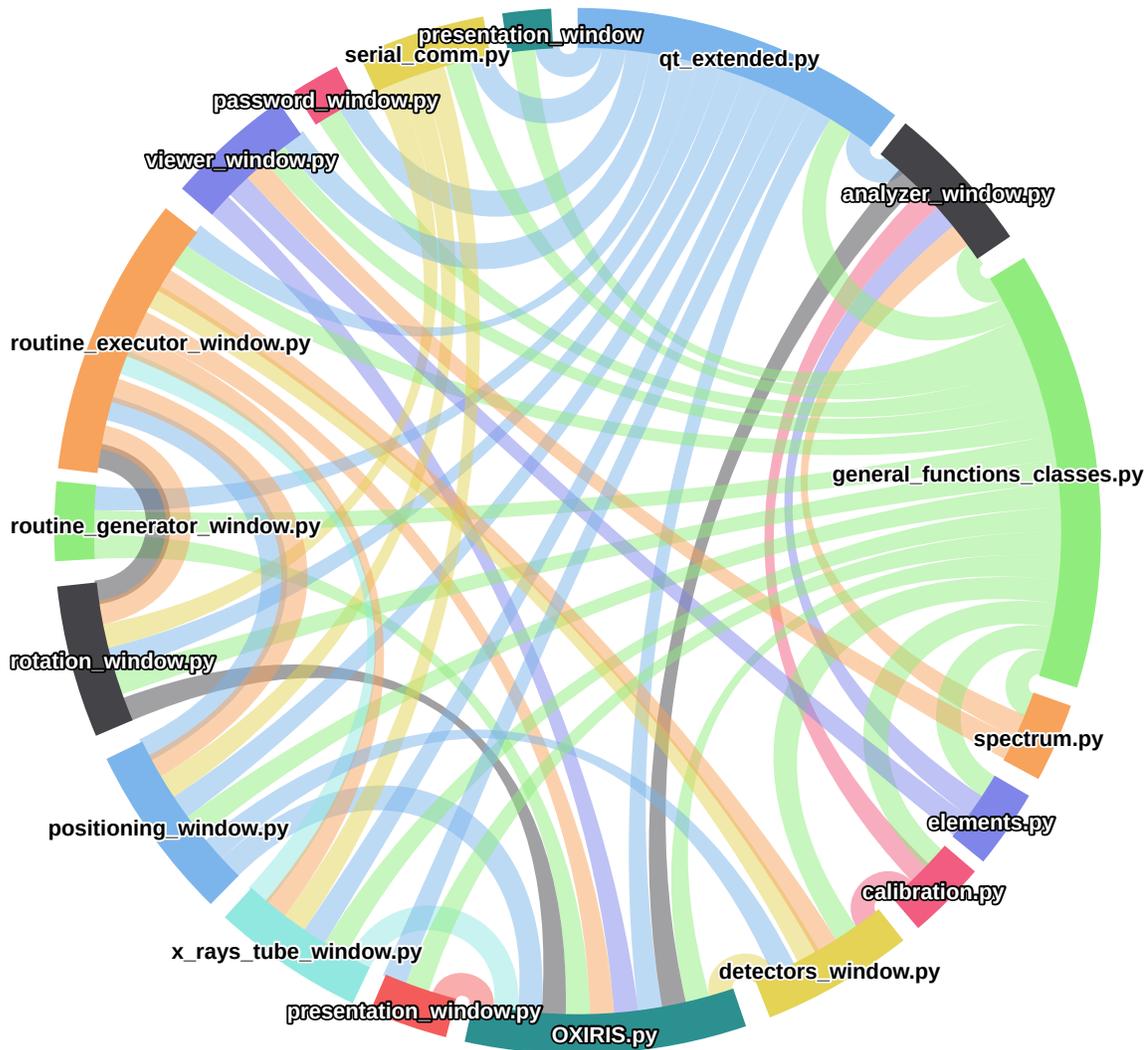


Figura 3.8: Diagrama de dependencias de los 17 archivos de código fuente. Cada archivo tiene un color y distintas ramas, que van desde el archivo original hasta algún otro donde éste sea utilizado. Se observa que los archivos más usados corresponden a `qt_extended.py` y `general_functions_classes.py`, lo que se fundamenta debido a que contienen funciones y clases esenciales y transversales a todo el software desarrollado.

3.2.3 Registro

Con el fin de generar un registro de los eventos que ocurren en el software se escribió una rutina especializada en esto, que se encuentra en el archivo `general_functions_classes.py`. En total se escriben 6 archivos de registro: `detectors_comm.log`, `general.log`, `positioning_comm.log`, `rotation_comm.log`, `routine_executor.log` y `x_rays_comm.log`.

Los archivos que terminan en `_comm` contienen todos los bytes enviados y recibidos hacia y desde los correspondientes periféricos, mientras que el `general` contiene información tal como la cantidad de dispositivos bluetooth encontrados o el idioma actual, y finalmente el `routine_executor` almacena información relacionada a la ejecución de las rutinas creadas por el usuario.

Cada uno de estos 6 archivos puede llegar a pesar hasta 10 mb, tras lo cual se comienza a escribir uno nuevo. Si existen más de 5 versiones del mismo archivo de registro se borrará el más antiguo, de forma tal que la suma de todos los registros nunca será superior a 300 mb. Estos registros se escribirán en español o inglés en función del idioma elegido por el usuario, y además también serán visibles desde la terminal en la que se ejecuta el software.

3.2.4 Documentación del Código

La documentación del código fuente es algo esencial en cualquier desarrollo de software, ya que permite realizar intervenciones más rápidas en el futuro, tanto para el creador así como también para los eventuales colaboradores. Debido a esto, y con el fin de dar cumplimiento al objetivo de documentar el código, todas las funciones del código están comentadas, así como también la mayoría de las variables. Además, todo el código ha sido procesado con la herramienta `doxygen`, lo que genera una documentación en formato `html` fácilmente accesible desde cualquier explorador web, tal como se observa en la figura 3.9.

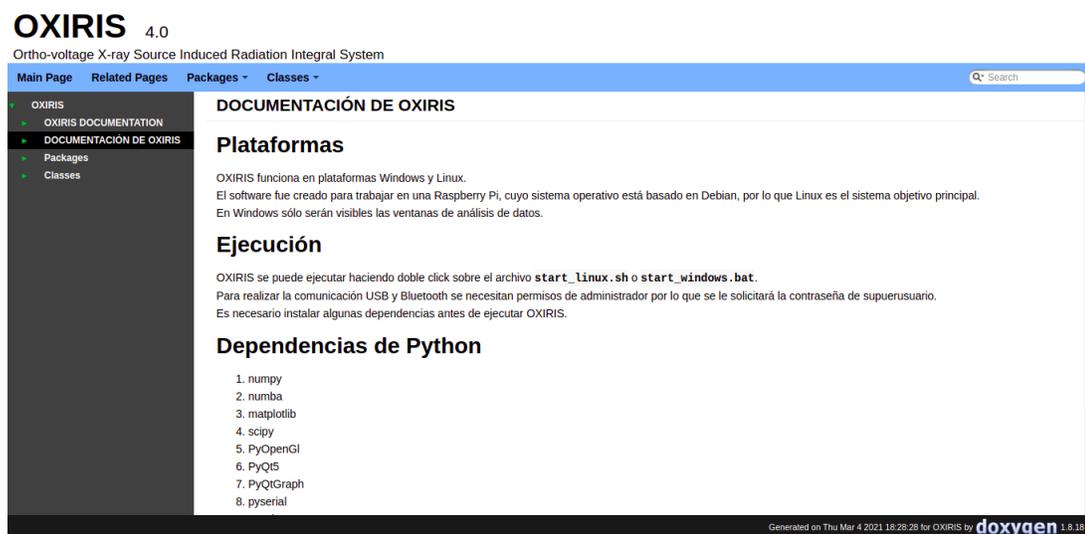


Figura 3.9: Página inicial en español de la documentación generada por `doxygen`. Desde aquí se pueden explorar todos los módulos, clases, funciones y variables. Además, posee un recuadro de búsqueda que facilita la tarea de encontrar elementos a partir de palabras claves.

3.2.5 Ejecución

Con el fin de ejecutar el programa de forma sencilla tanto desde Linux como desde Windows se escribieron dos ficheros, `start_linux.sh` y `start_windows.bat`. El primero se ejecuta desde una terminal de Linux escribiendo `bash start_linux.sh`, mientras que el segundo se ejecuta en Windows haciendo doble click.

En el caso de Linux aparece una ventana solicitando la contraseña del usuario, dado que es necesario tener permisos de superusuario para realizar la comunicación USB (figura 3.10).

Tras verificar que la contraseña es correcta se abre la interfaz gráfica de OXIRIS, y si se detecta una arquitectura ARM se bloquea el gráfico volumétrico del visor 3D, pues este tipo de gráfico no es compatible con dicha arquitectura.

En el caso de Windows se muestra la interfaz de forma inmediata, pero sin todas las funciones habilitadas. Sólo es accesible la pestaña de bienvenida, el generador de rutinas, el analizador de espectros y el visor 3D. Esto es debido a que la comunicación con los periféricos ha sido programada para llevarse a cabo en un sistema Linux, por lo que en Windows sólo se pueden analizar datos y generar rutinas.

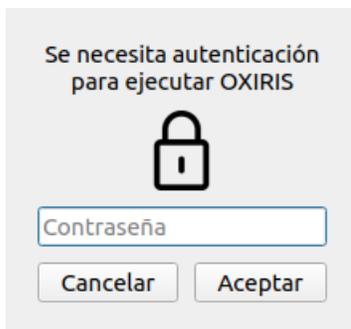


Figura 3.10: Ventana de ingreso de contraseña que se muestra en Linux. Si la contraseña es incorrecta se volverá a solicitar.

3.3 Correcciones

Dado que los espectros son distorsionados tanto por la eficiencia del detector así como también por los picos de escape, se llevan a cabo correcciones para ambos casos, cumpliendo así el objetivo de realizar correcciones con fundamento físico. A continuación se expone el detalle de cada una.

3.3.1 Eficiencia del detector

Los datos de eficiencia del detector, mostrados en la figura 3.11, han sido obtenidos desde la página web de AmpTek.

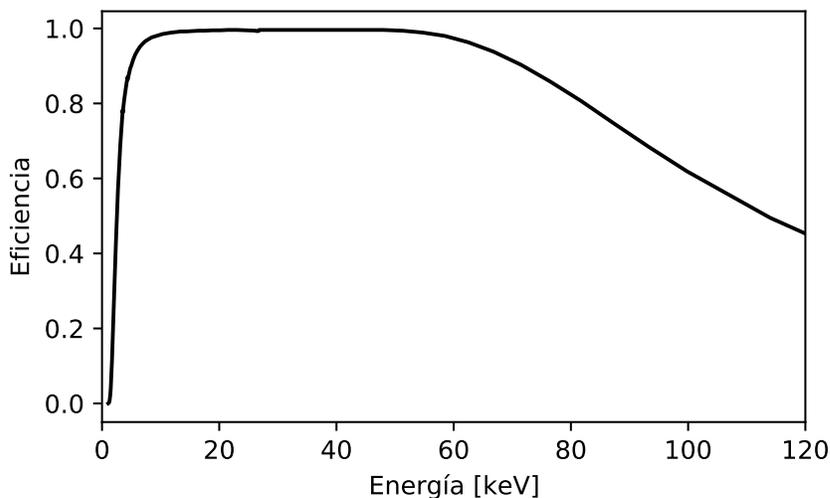


Figura 3.11: Gráfico de la eficiencia del detector XR-100T-CdTe entre 0 y 120 keV. Se observa que la eficiencia para energías muy bajas tiende a 0, mientras que en la zona ubicada entre 10 y 60 keV es muy cercana a 1.

El algoritmo correctivo recorre todos los puntos, de izquierda a derecha, y divide las cuentas detectadas por la eficiencia correspondiente a dicha energía, interpolando linealmente la base de datos si fuese necesario.

Por ejemplo, si en el canal equivalente a 60 keV hay 500 cuentas, y sabiendo que la eficiencia del detector para esa energía es de 0.97, entonces la cantidad corregida corresponde a $500/0.97 = 515$ cuentas.

El fundamento del algoritmo es el siguiente: sea A la cantidad de fotones detectados, B la cantidad de fotones emitidos, y e la eficiencia de detección, entonces:

$$A = Be \quad (3.1)$$

por lo que, despejando B :

$$B = A/e \quad (3.2)$$

3.3.2 Picos de Escape

Los picos de escape afectan el espectro llevando cuentas desde un canal de mayor energía a otro de menor energía.

Particularmente, en los detectores de CdTe el efecto de los picos de escape es más notorio pues las transiciones K del Cd y Te están entre 20 y 30 keV, lo que genera picos muy distanciados desde el canal en el que deberían estar.

Tal como indica Redus *et al.* (2009) se puede hacer una corrección por picos de escape sabiendo qué porcentaje de las cuentas totales para cierto canal han caído en un canal menos energético debido a la interacción con el volumen de CdTe.

Con el fin de obtener estos porcentajes se hicieron 441 simulaciones Monte Carlo con el software PENELOPE (Baró *et al.*, 1995), abarcando energías desde 20 hasta 130 keV.

Todas las simulaciones fueron ejecutadas en el clúster Lautaro de la Universidad de La Frontera, que cuenta con 60 GB de RAM y 40 CPUs que operan a una velocidad máxima de 3 GHz, lo que permitió completar la tarea en 5 días. El análisis posterior se realizó de forma local en un computador con procesador AMD Ryzen 5 que cuenta con 16 GB de RAM. Ambas tareas fueron intensivas en cuanto a CPU, pero no en cuanto a RAM.

En todos los casos se consideró un haz rectangular de 5 x 5 mm con $1e7$ fotones emitidos de forma homogénea en el eje $-\hat{z}$, impactando sobre la geometría de la figura 3.12(a), que consiste en una placa de CdTe de 5 mm x 5 mm x 1 mm, lo que corresponde a la especificación del detector XR-100T-CdTe utilizado en OXIRIS.

Envolviendo esta geometría se puso un detector esférico, visible en la figura 3.12(b), cuyo fin es detectar los fotones fluorescentes emitidos desde el volumen de CdTe.

A modo de ejemplo, en el Anexo I y Anexo J se adjuntan tanto el archivo de *input* como también de geometría de la simulación de 50 keV. Todos los demás archivos son iguales, excepto por la energía del haz primario.

Con el fin de verificar que la geometría y configuración del haz se han escrito correctamente se ejecutó una simulación de 50 keV para obtener los perfiles dosimétricos a lo largo de los 3 ejes cartesianos, lo que se observa en la figura 3.13.

El resultado es que la dosis en el plano xy para cierto z tiende a ser constante, mientras que a lo largo de z decae exponencialmente, teniendo su máximo poco después de la superficie donde impacta el haz y el mínimo en la superficie opuesta, lo que corresponde al perfil típico de un haz de rayos X. Lo anterior indica que el volumen está siendo irradiado homogéneamente, por lo que se verifica que las simulaciones han sido bien escritas.

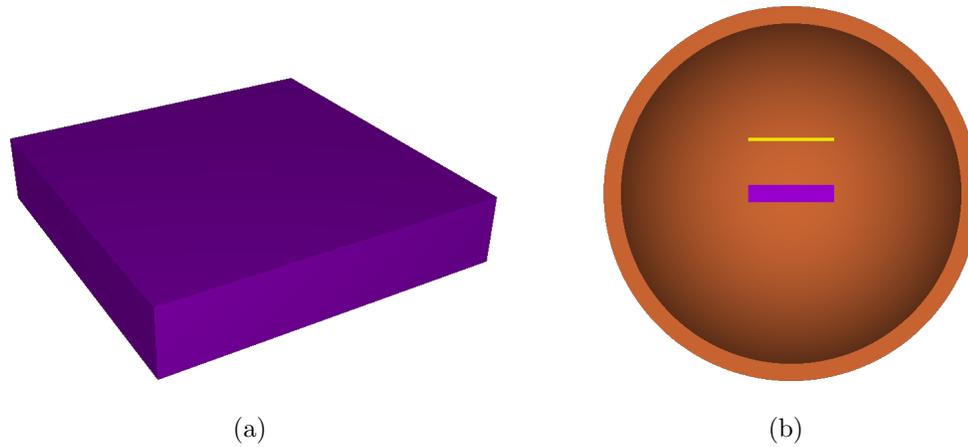


Figura 3.12: En a) se observa el volumen de CdTe, definido por $-0.25 \text{ cm} \leq x \leq 0.25 \text{ cm}$, $-0.25 \text{ cm} \leq y \leq 0.25 \text{ cm}$, $-0.05 \text{ cm} \leq z \leq 0.05 \text{ cm}$. Por otro lado, en b) se muestra un corte transversal donde se aprecia el volumen de CdTe junto con el detector esférico, de radio 1 cm, y la fuente de rayos X (en amarillo), ubicada en $z = 0.3 \text{ cm}$.

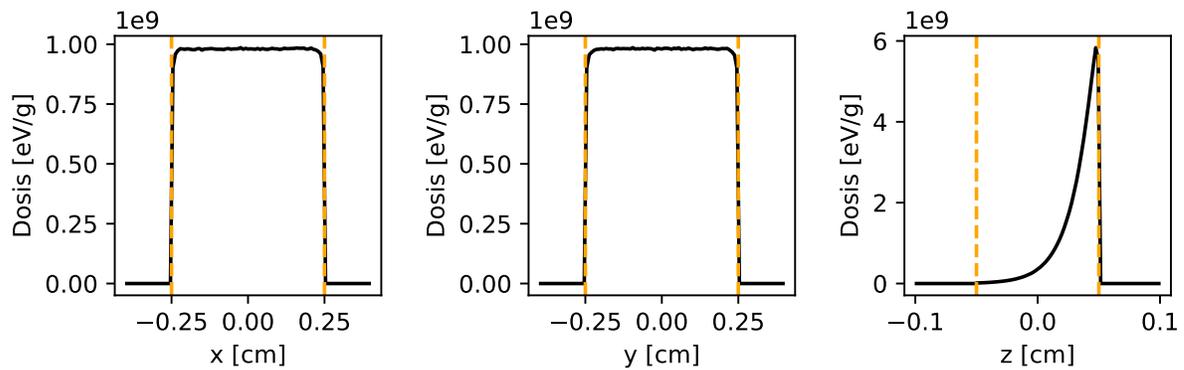


Figura 3.13: Perfiles dosimétricos de las simulaciones Monte Carlo. En línea segmentada naranja se muestran los límites del volumen de CdTe.

Tras realizar las simulaciones se obtienen resultados como el de la figura 3.14, donde se observan claramente las transiciones K del Cd ($K_{\alpha_1} = 23.173 \text{ keV}$, $K_{\alpha_2} = 22.984 \text{ keV}$, $K_{\beta_1} = 26.095 \text{ keV}$, $K_{\beta_2} = 31.698 \text{ keV}$) y del Te ($K_{\alpha_1} = 27.472 \text{ keV}$, $K_{\alpha_2} = 27.201 \text{ keV}$, $K_{\beta_1} = 30.995 \text{ keV}$ y $K_{\beta_2} = 31.698 \text{ keV}$).

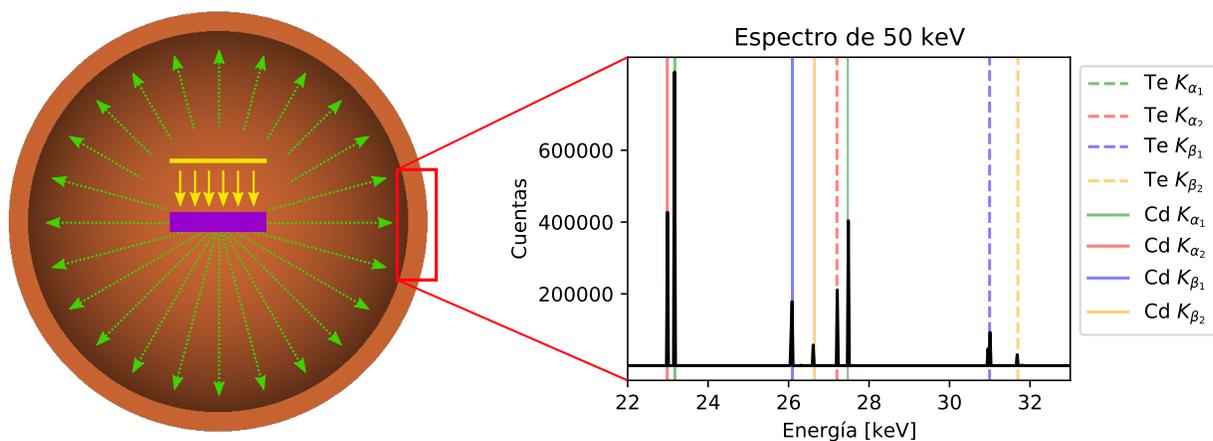


Figura 3.14: Proceso y resultado de una simulación. El haz incidente monoenergético (amarillo) interactúa con el volumen de CdTe (morado) generando fotones de escape (verde) que son recolectados por el detector esférico. Para un haz incidente de 50 keV se obtiene el espectro de la derecha, donde se observan claramente las transiciones K del CdTe.

Posteriormente, cada uno de los espectros obtenidos con el detector esférico se procesó de la siguiente forma:

1. Se identificaron los canales correspondientes a las energías de las transiciones K_{α_1} , K_{α_2} , K_{β_1} y K_{β_2} , tanto del Cd como del Te.
2. Se guardaron las cuentas presentes en cada uno de estos 8 canales.
3. Se dividieron estos valores por la intensidad del haz incidente, lo que permite obtener la probabilidad de generación de fotones de escape en el volumen de CdTe, variable que se denominó “probabilidad de escape”.
4. Se le restó al número 1 la suma de todas estas probabilidades, lo que permite obtener la “probabilidad de no escape”, es decir, la probabilidad de que no se generen fotones de escape en el volumen de CdTe.

Los datos obtenidos después de procesar todos los archivos se grafican en la figura 3.15.

Como se nota, las curvas no son suaves, por lo que no habrá una función que se ajuste bien. Debido a esto se ha decidido interpolar linealmente cuando sea necesario obtener los datos para una energía no simulada.

Una vez que se calculan todas las probabilidades se puede llevar a cabo la corrección, que consiste en:

- Procesar el espectro de mayor a menor energía.
- Obtener desde la base de datos las probabilidades de escape y de no escape para la energía actual.
- Dividir la cantidad de cuentas del canal seleccionado por la probabilidad de no es-

cape correspondiente (el fundamento es idéntico al de la corrección por eficiencia del detector).

- Multiplicar este valor corregido por cada una de las 8 probabilidades de escape.
- Para cada transición, buscar el canal más cercano al de la energía actual menos la energía de la transición, y restar la cantidad calculada en el paso anterior.

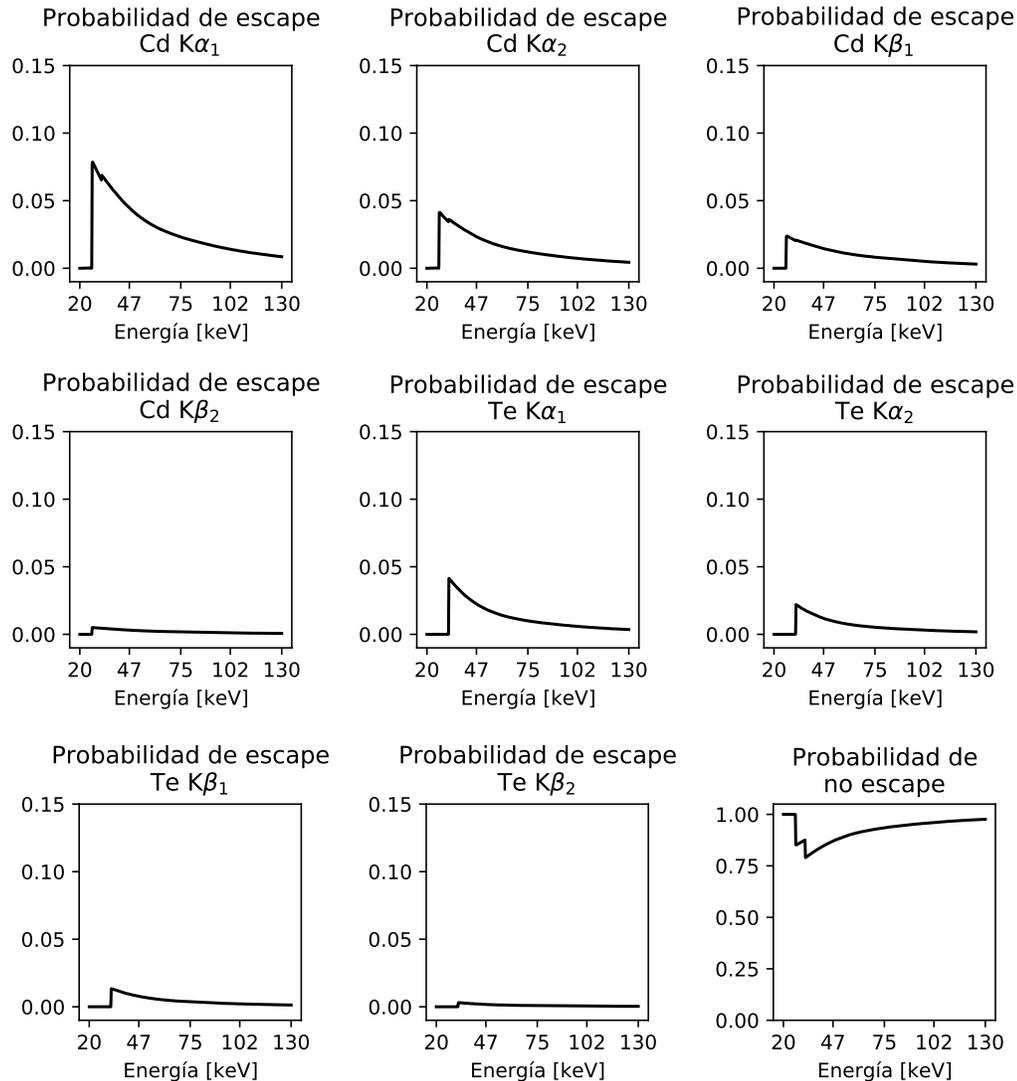


Figura 3.15: Se muestran las 8 probabilidades de escape así como también la probabilidad de no escape. Como era de esperar no existen fotones de escape para energías inferiores al borde de absorción K del Cd, por lo que en este intervalo la probabilidad de no escape es 1. Por otro lado, se observan dos mínimos de no escape, uno en 27 keV y el otro en 32 keV, que corresponden a los bordes de absorción K del Cd y Te, respectivamente. Posteriormente, aproximadamente desde los 100 keV, se observa un comportamiento tendiente a 1 para la probabilidad de no escape y a 0 para la probabilidad de escape.

Capítulo IV
RESULTADOS

Capítulo IV

Resultados

4.1 Comunicación con los Periféricos

Debido a la pandemia y a la falta de accesibilidad al laboratorio causada por ésta, no se pudo probar de forma adecuada la comunicación con los periféricos. Sin embargo, tanto el tubo de rayos X como los DPP/detectores se probaron preliminarmente de forma remota.

En el caso del tubo de rayos X se verificó que se pueden configurar correctamente todas sus variables, así como también iniciar y detener disparos. No obstante, se encontró un problema relacionado con el protocolo de comunicación Bluetooth mediante el cual opera el tubo, que es propenso a corromper los datos. Si bien el algoritmo es capaz de identificar la mayoría de estos problemas y solucionarlos mediante el re-envío o la re-solicitud de paquetes, en ciertas ocasiones la comunicación se corta o se retrasa. Este problema se puede solucionar utilizando un protocolo de comunicación no inalámbrico, sin embargo esto no es viable pues la rotación del tubo no permite establecer una conexión al computador mediante cables.

En el caso de los DPP/detectores se realizaron múltiples pruebas, comprobando que todos los parámetros incorporados en la interfaz gráfica se pueden modificar de forma correcta, y que se pueden obtener espectros y graficarlos en tiempo real.

Por otro lado, dado que el hardware del sistema de posicionamiento y de rotación no se puede probar pues no tienen implementado el protocolo de comunicación propuesto en este trabajo, se emuló el comportamiento de ambas piezas con un Arduino. El resultado es que en ambos casos se pueden configurar todas las variables así como también iniciar y detener movimientos.

Por último, se comprobó que se pueden escribir archivos de rutina tan largos como sea necesario, y que todas las instrucciones se interpretan correctamente.

4.2 Interfaz gráfica

La interfaz gráfica se ha separado en 9 pestañas, cada una de ellas con múltiples divisiones, las cuales son expuestas en detalle en esta sección.

4.2.1 OXIRIS



Figura 4.1: Pestaña principal de OXIRIS.

La pestaña principal del software lleva por nombre OXIRIS, y su función es abrir la documentación, cambiar el idioma, y mostrar las instituciones asociadas (figura 4.1).

Con el botón “documentación del código (HTML)” se abre la documentación generada por `doxygen` en el explorador web, mientras que con el botón “documentación del software (PDF)” se abrirá el manual de usuario cuando esté disponible.

La apertura de documentos en Linux es posible mediante el comando `gio`, mientras que en Windows es con la función `startfile` de la librería `os` de Python. De esta forma se abren todos los documentos dentro del software.

Por último, con el botón “cambiar idioma”, el usuario puede cambiar el idioma de la interfaz gráfica, pudiendo elegir entre español e inglés (figura 4.2(a)). Tras configurar un nuevo idioma el programa se debe reiniciar para ver los cambios (figura 4.2(b)).

Es posible cambiar el idioma gracias a que cada `string` ha sido tratado como un vector, en el cual el índice 0 es ocupado con la palabra en inglés y el índice 1 con la palabra en español. El idioma actual se guarda en el fichero `conf`. Por ejemplo, si corresponde a español, en este archivo la primera línea será `language=es`.

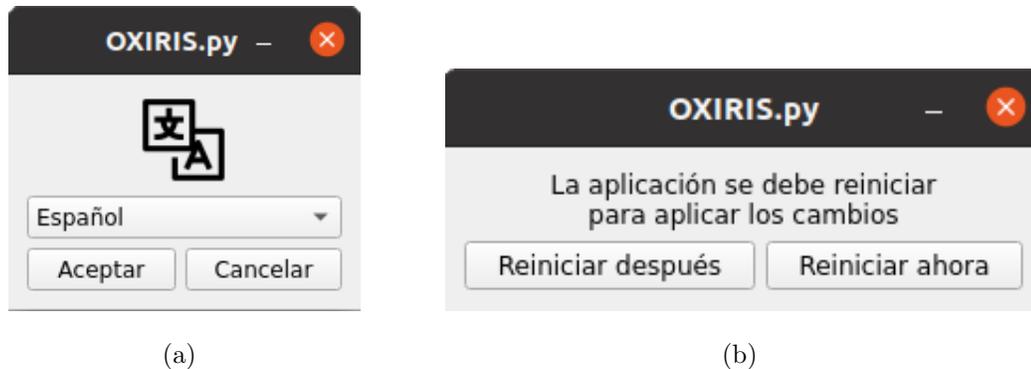


Figura 4.2: En a) se muestra la ventana para cambiar idioma, y en b) la ventana de reinicio que aparece tras seleccionar un idioma distinto al actual.

4.2.2 Tubo de rayos X

Tal como se observa en la figura 4.3, esta pestaña está compuesta por 8 secciones, que se detallan a continuación:

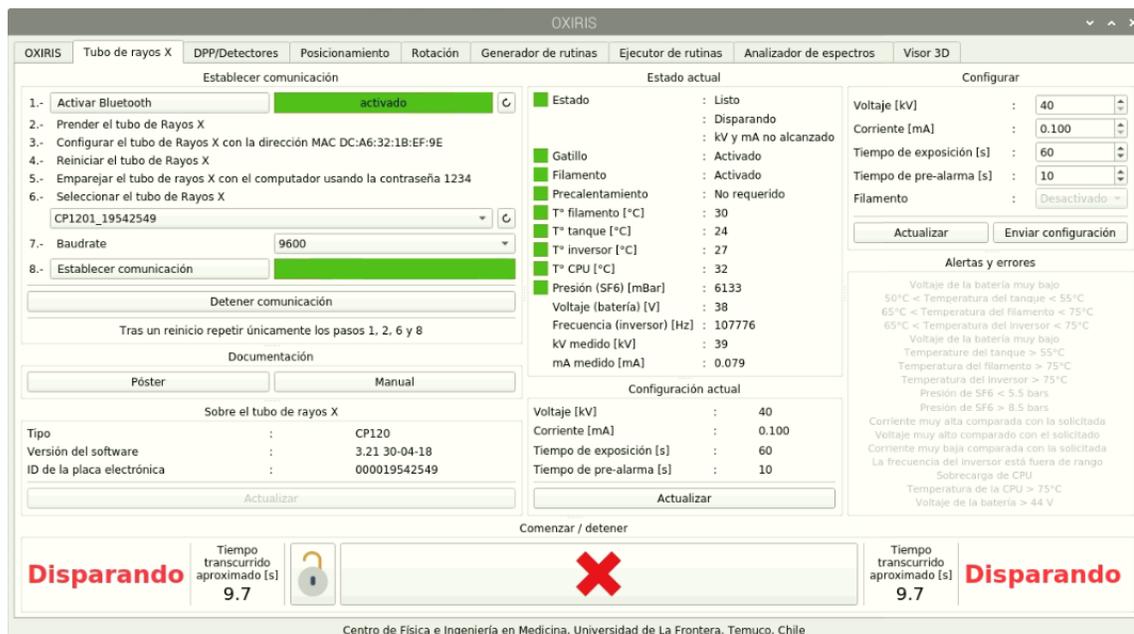


Figura 4.3: Pestaña encargada de la comunicación con el tubo de rayos X. En la captura se observa el tubo de rayos X disparando, mientras se muestra su estado en tiempo real.

Ventana: Establecer comunicación

La comunicación con el tubo de rayos X se realiza a través de un puerto serial utilizando la interfaz bluetooth del computador, para lo cual es necesario crear un puerto serial virtual.

Esto en Linux se logra mediante el comando `rfcomm`. Así, por ejemplo, para crear el puerto serial virtual `rfcomm5`, cuyo punto de acceso se ubica en `/dev`, se utiliza el comando `sudo rfcomm bind 5 MAC`, donde `MAC` corresponde al MAC del dispositivo bluetooth que se quiere utilizar.

Para que este comando se ejecute exitosamente el computador y el tubo de rayos X deben estar previamente emparejados, para lo cual primero el tubo se debe configurar con el MAC del computador, que se muestra en el paso 3, y luego desde el computador se debe realizar el emparejamiento.

En total son 8 pasos para realizar la primera conexión:

1. Activar bluetooth. El indicador de la derecha se pondrá verde cuando esté activo, y rojo cuando esté apagado.
2. Prender el tubo de rayos X.
3. Configurar el tubo de rayos X con la dirección MAC mostrada. Esta dirección cambia en función del computador.
4. Reiniciar el tubo de rayos X.
5. Emparejar el tubo de rayos X con el computador. Esto se hace a través del gestor bluetooth del sistema operativo, utilizando la contraseña 1234, que es una de las contraseñas por defecto que utilizan los dispositivos bluetooth.
6. Seleccionar el tubo de rayos X. En este paso se muestran todos los dispositivos detectados por el computador, entre los cuales debería estar el tubo. Este paso es importante pues el elemento seleccionado se utiliza para extraer el MAC que es usado para crear el puerto virtual.
7. Baudrate. Se debe elegir la velocidad de transmisión de datos, cuyo valor por defecto es 9600, y en la mayoría de los casos estará bien.
8. Establecer comunicación. Este botón generará el puerto serial virtual, lo abrirá y comenzará a escribir instrucciones de forma periódica cuyo objetivo es mantener actualizada la pestaña y mostrar el estado actual del tubo. Mientras la comunicación esté activa, la barra de la derecha estará verde, en caso contrario se pondrá roja.

Tras un reinicio no es necesario repetir todos los pasos, pues el tubo de rayos X ya está configurado con el MAC del computador, y ambos dispositivos ya se encuentran emparejados, por lo que basta con ejecutar los pasos 1, 2, 6 y 8.

Una vez que el puerto está creado, éste se puede manejar con la librería `pyserial` como si fuera cualquier otro puerto normal.

Finalmente, si el usuario quiere cerrar la comunicación, lo puede hacer de 3 formas: apagando el tubo de rayos X, lo que será detectado tras 10 segundos; cerrando el programa, lo que será detectado por el tubo de rayos X tras 5 segundos, y provocará que cualquier disparo activo se detenga; y por último, apretando el botón “detener comunicación”, que es el método recomendado y más seguro.

Utilizando puertos virtuales es imposible saber de forma certera cuándo el dispositivo bluetooth está activo o inactivo. Para lograrlo se ha escrito una rutina con múltiples validadores en el archivo `serial_comm.py`, que envía las instrucciones de forma persistente durante 5 intentos de 2 segundos cada uno. Si tras todos los intentos no se recibe ninguna respuesta, se asume que el tubo está inactivo y la comunicación se corta automáticamente. Es debido a esto que si el tubo se apaga el software lo detecta 10 segundos después.

Ventana: Documentación

En el apartado de documentación se pueden abrir 2 documentos PDF: un póster creado por el fabricante donde se muestran los detalles del tubo, y el manual oficial del mismo.

Ventana: Sobre el tubo de rayos X

En esta sección se muestra la siguiente información relativa al tubo de rayos X: tipo (CP120B o CP160B), versión del software interno, e ID de la placa electrónica.

Estos datos se reciben automáticamente una vez que se inicia la comunicación, aunque también pueden ser solicitados por el usuario haciendo click en “actualizar”.

Tanto este botón como los demás presentes en la pestaña, envían un comando al tubo mediante el puerto serial virtual, cuya especificación se encuentra en el protocolo de comunicación. Las respuestas son procesadas en una función dedicada a ello, que interpreta cada sucesión de bytes, o bien cada bit, tal como se indica en el protocolo.

Ventana: Estado actual

Al iniciarse la comunicación comienzan a enviarse múltiples instrucciones al tubo de rayos X, cuyas respuestas contienen: el estado actual, estado del gatillo, estado del filamento, tiempo necesario de precalentamiento, temperatura del filamento, temperatura del tanque, temperatura del inversor, temperatura de la CPU, presión, voltaje de la batería, frecuencia del inversor, kV medido y mA medido.

Cada ítem se actualiza aproximadamente cada 4 segundos, lo que le permite al usuario conocer el estado del tubo casi en tiempo real.

Algunos de estos ítems pueden generar alertas y alarmas, por ejemplo si la temperatura del tanque es superior a 55°C. En tal caso el recuadro de la izquierda adquirirá distintos colores: gris si no hay ningún aviso o el ítem está desactivado, verde si los valores están correctos o si el ítem está activo, naranja si los valores están sobrepasando los límites, y rojo si los valores son críticos. Si ocurre esto último el tubo automáticamente se detiene y/o apaga.

Ventana: Configuración actual

En esta sección se muestra el valor actual de 4 de los 5 elementos configurables por el usuario: voltaje, corriente, tiempo de exposición, y tiempo de pre-alarma.

La pre-alarma consiste en un estado previo al disparo en el cual el tubo hace sonar una alarma, con el fin de avisar que posteriormente estará activo.

Los valores de esta sección se actualizan automáticamente según se vayan configurando, aunque el usuario también lo puede hacer mediante el botón “actualizar”.

Ventana: Configurar

En la sección configurar se muestran los 5 elementos configurables del tubo, junto con los *widgets* para elegir sus valores dentro de los rangos permitidos:

- Voltaje: Entre 40 y 80 kV si la corriente está entre 1.0 y 1.5 mA, y entre 40 y 120 kV si la corriente está entre 0.1 y 1.0 mA.
- Corriente: Entre 0.1 y 1.0 mA si el voltaje está entre 20 y 120 kV, y entre 1.0 y 1.5 mA si el voltaje está entre 40 y 80 kV.
- Tiempo de exposición: Entre 1 y 999 segundos.
- Tiempo de pre-alarma: Entre 3 y 99 segundos.
- Filamento: Activado o desactivado. Cuando está activado el tubo tarda menos en llegar al voltaje y corriente especificadas, pero la batería se agota más rápido.

Esta configuración se envía al tubo de rayos X con el botón “enviar configuración”.

Tras establecer la conexión con el tubo los *widgets* son actualizados con los valores actuales de los 5 parámetros. El usuario también puede ejecutar esta acción mediante el botón “actualizar”.

Ventana: Alertas y errores

El tubo puede presentar 17 alertas y/o errores, ya sea de forma individual o combinada, tal como se muestra en la sección “alertas y errores” de la figura 4.3.

Cuando es una alerta, el texto en cuestión se pondrá naranja, y el tubo seguirá operando normalmente. Por otro lado, si es un error, el texto se pondrá rojo y se detendrá cualquier tarea que el tubo esté ejecutando en ese momento.

Cuando la alerta o error deje de existir el texto volverá a estar en negro.

Ventana: Comenzar/detener

Desde esta sección se puede iniciar y detener un disparo.

Con el fin de generar mayor seguridad se ha puesto un candado al lado izquierdo del botón principal, de tal forma que cuando está cerrado el botón principal se deshabilita, y cuando está abierto el botón principal se habilita. Así, y dado que inicialmente está cerrado, se evita que el usuario comience un disparo por error.

Una vez que el disparo haya comenzado se mostrará el tiempo transcurrido aproximado, así como también una animación en función del estado actual.

Dado que el tiempo no puede ser medido con precisión pues el protocolo de comunicación no tiene ninguna especificación para ello, éste se mide de forma aproximada utilizando un cronómetro.

La animación, que consiste en un texto con color fijo o parpadeante, puede tener 4 estados:

- Desconectado: Color gris.
- Conectado: Parpadeante entre color verde y gris.
- Pre-alarma: Parpadeante entre color naranja y gris.
- Disparando: Parpadeante entre color rojo y gris.

Por último, cuando el tubo esté en tiempo de pre-alarma o de exposición, emitirá alarmas sonoras fácilmente identificables.

4.2.3 DPP/Detectores

Tal como se observa en la figura 4.4, esta pestaña está compuesta por 5 secciones, que se detallan a continuación:

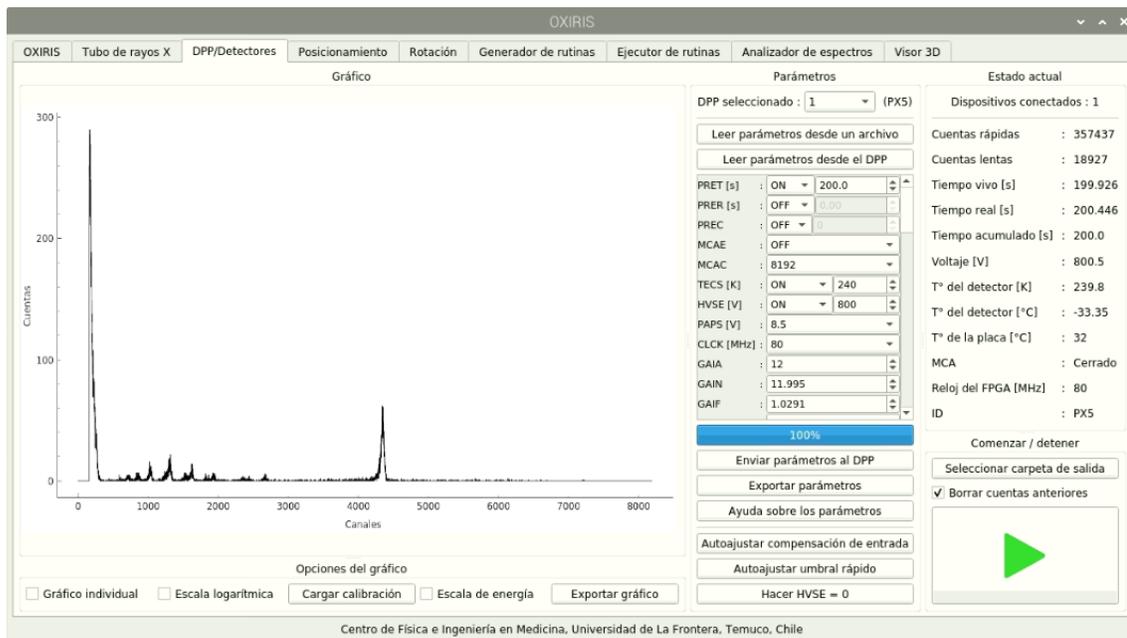


Figura 4.4: Pestaña encargada de la comunicación con los DPP y detectores. En la captura se observa un par DPP/detector conectado y capturando un espectro de Am-241, mientras se muestra su estado en tiempo real. El peak marcado que aparece al inicio corresponde a ruido electrónico del dispositivo.

Ventana: Gráfico

Es la sección que más espacio visual utiliza, y su función es mostrar en vivo el espectro que se está detectando.

Para mantener los gráficos actualizados se envía la solicitud de paquete de espectro a todos los DPP conectados cada 1 segundo, desde que comienza la detección hasta que finaliza la misma.

Si es que hay más de 1 DPP conectado el gráfico se subdivide de tal forma que se puedan ver todos los espectros detectados.

Ventana: Opciones del gráfico

Desde este apartado se pueden seleccionar 5 opciones:

- **Gráfico individual:** Por defecto el gráfico se subdivide cuando hay más de 1 DPP conectado, pero si se marca esta casilla se evita este comportamiento y todos los espectros se muestran juntos.
- **Escala logarítmica:** Habilita o deshabilita la escala logarítmica en el eje y . Los valores menores o iguales a 0 no se muestran.

- Cargar calibración: Permite cargar uno o varios archivos de calibración, los que se crean en la pestaña “anализador de espectros”. Los archivos de calibración permiten generar la equivalencia entre canales y energía (ver Anexo B y Anexo C).
- Escala de energía: Si es que se han cargado archivos de calibración para los DPP conectados, entonces se habilita la escala de energía en el eje x .
- Exportar gráfico: Exporta la vista actual del gráfico en formato png.

Ventana: Parámetros

Desde aquí se controlan todos los parámetros de los DPP conectados, mediante las opciones que se indican a continuación:

- DPP seleccionado: Se muestran todos los DPP conectados al computador, y se da la opción de seleccionar uno de ellos para configurarlo.
- Parámetros: El protocolo de comunicación documenta 66 parámetros distintos, que modifican cosas tales como: tiempo de captura, cantidad de canales, temperatura del detector, ganancia, etc. De éstos, sólo se implementaron 61 parámetros, dejando fuera: ACKE, SCAH, SCAI, SCAL y SCAO; pues no tienen relevancia para el uso actual. Cada uno de estos 61 parámetros tiene sus límites, muchos de los cuales no son fijos y dependen de varios otros parámetros.

En la interfaz se presentan las siguientes opciones:

- Leer parámetros desde un archivo: Abre una ventana para que el usuario seleccione un archivo desde donde cargar el valor de los parámetros (ver Anexo D).
- Leer parámetros desde el DPP: Carga la configuración actual que posee el DPP.
- Parámetros: Se muestran los 61 parámetros junto con las opciones para elegir sus valores. Se han contemplado todas las interconexiones que se especifican en el protocolo de comunicación, por lo que es imposible que algún parámetro se configure mal. Además, algunos parámetros existen sólo para uno de los dos DPP, por lo que, en función del DPP elegido, estos parámetros se habilitan o deshabilitan. Finalmente, dado que cada parámetro se especifica únicamente por una variable de 4 letras, se muestra la definición de cada uno si el mouse se deja encima durante 2 o más segundos.
- Enviar parámetros al DPP: Se envían todos los parámetros al DPP seleccionado.
- Exportar parámetros: Se exportan los parámetros a un archivo de texto que más tarde puede ser cargado con el botón “leer parámetros desde un archivo” (ver Anexo D).

- Ayuda sobre los parámetros: Muestra la sección del protocolo de comunicación donde se define cada uno de todos los parámetros.
- Autoajustar compensación de entrada y autoajustar umbral rápido: Corresponden a dos parámetros que pueden ser autoajustados. Para ello se debe quitar la fuente de radiación y apretar el botón correspondiente.
- Hacer HVSE=0: El parámetro HVSE controla el voltaje al cual opera el detector. Antes de apagar o desconectar los DPP el valor de HVSE debe ser configurado como 0, lo que se logra mediante este botón.

Dado que idealmente las subidas y bajadas de voltaje deben ser graduales, se configura el parámetro HVSE para que cada vez que sea modificado se alcance el nuevo valor en hasta 10 pasos con valores de voltaje intermedio, separados entre sí por 1 segundo.

Ventana: Estado actual

En esta sección se muestra la cantidad total de DPPs conectados, junto con el estado actual de cada uno de ellos y del detector asociado, lo que incluye: cuentas rápidas, cuentas lentas, tiempo vivo, tiempo real, tiempo acumulado, voltaje, temperatura del detector, temperatura de la placa, estado del MCA, frecuencia del reloj, e ID del DPP.

Para ello se envía el paquete de solicitud de estado cada 1 segundo a todos los DPP.

Por otro lado, y de forma paralela, los puertos USB son escaneados cada 2 segundos con el fin de mantener actualizada la cantidad de dispositivos conectados. Cada vez que se detecta un nuevo dispositivo, se inicializa y se comienza a controlar mediante la librería `pyusb`.

Ventana: Comenzar/detener

Desde aquí se puede elegir la carpeta donde se guardarán los espectros.

También se puede elegir si se quiere borrar el espectro detectado la última vez, o bien si el nuevo espectro se debe sumar al anterior.

Finalmente, se puede iniciar la detección, que durará hasta que se cumpla el tiempo fijado o la cantidad de cuentas configurada.

Durante la detección y cada aproximadamente 1 segundo se escribirán en el disco duro los archivos de espectro asociados a cada DPP/detector conectado, según se especifica en el Anexo E. Se decidió escribir continuamente y no sólo tras finalizar con el fin de no perder el tiempo de detección frente a posibles cortes de electricidad o problemas externos que cierren la interfaz gráfica.

4.2.4 Posicionamiento

Tal como se observa en la figura 4.5, esta pestaña está compuesta por 4 secciones, que se detallan a continuación:

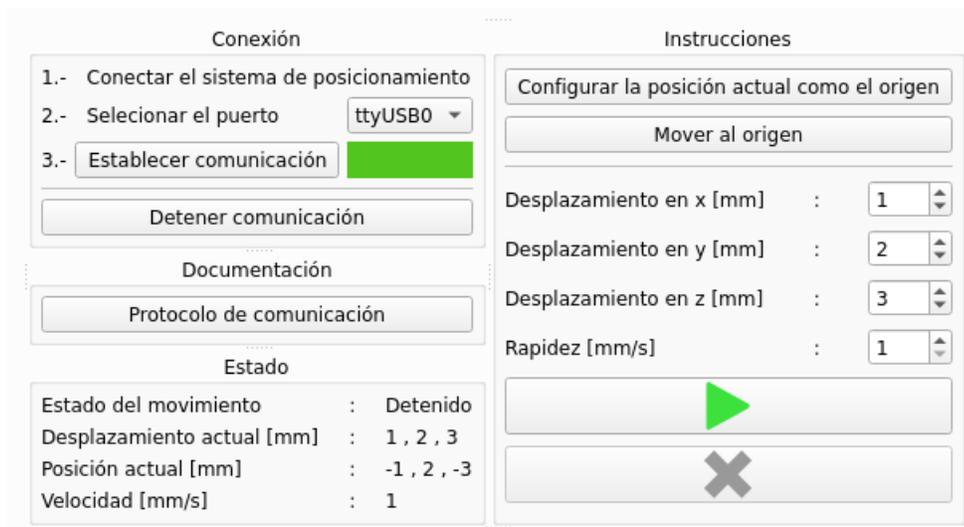


Figura 4.5: Pestaña encargada de la comunicación con el sistema de posicionamiento. Esta pestaña es más pequeña que las anteriores, por lo que a la captura se le ha recortado el borde para evitar espacios vacíos. En el puerto serial se encuentra conectado el sistema de posicionamiento emulado por un Arduino, y como se observa ya ha realizado un desplazamiento en (1, 2, 3) mm a 1 mm/s, quedando en la posición (-1, 2, -3) mm.

Ventana: Conexión

Los pasos para establecer la conexión son 3: primero se debe conectar el sistema de posicionamiento, luego se debe seleccionar el puerto en el que ha sido conectado, y finalmente se debe hacer click en “establecer comunicación”.

Cada 2 segundos se escanean automáticamente los puertos USB con el fin de detectar si el dispositivo está conectado o no. No obstante, el puerto en el cual está conectado no se puede deducir, es por ello que el usuario lo debe seleccionar de entre los que están disponibles.

Cuando la comunicación es establecida, se inicializa y se abre el puerto elegido mediante la librería `pyserial`, y se comienza a enviar de forma continua, cada 2 segundos, los comandos para actualizar el estado del desplazamiento y la posición actual.

Ventana: Documentación

Permite abrir el protocolo de comunicación (Anexo F) por si el usuario necesitara conocer detalles sobre la comunicación del sistema.

Ventana: Estado

Desde aquí se puede ver el estado actual del sistema de posicionamiento:

- Estado del movimiento: Puede tomar dos valores, “moviendo” o “detenido”.
- Desplazamiento actual: Muestra el desplazamiento que se está ejecutando actualmente (Δx , Δy , Δz).
- Posición actual: Muestra la posición actual de la camilla (x , y , z), en referencia al origen especificado por el usuario.
- Velocidad: Muestra la velocidad que está configurada (no la velocidad actual).

Ventana: Instrucciones

Desde este apartado se pueden enviar las siguientes instrucciones al sistema de posicionamiento:

- Configurar la posición actual como el origen: Este botón permite configurar cualquier posición en la que se encuentre el sistema como el origen (0, 0, 0).
- Mover al origen: El sistema de posicionamiento se mueve al origen utilizando la última rapidez configurada. Si no se ha configurado ninguna, se elige automáticamente la rapidez de 1 mm/s.
- Desplazamientos: Se especifican los desplazamientos que debe realizar el sistema de posicionamiento a partir del punto actual. El valor debe ser un número entero entre -999 y +999 mm.
- Rapidez: Se configura la rapidez a la que el sistema realizará todos los movimientos. El valor mínimo es 1 mm/s y el máximo es 99 mm/s.
- Iniciar/detener: Desde aquí se puede iniciar el último desplazamiento configurado, o bien detener la ejecución del mismo.

4.2.5 Rotación

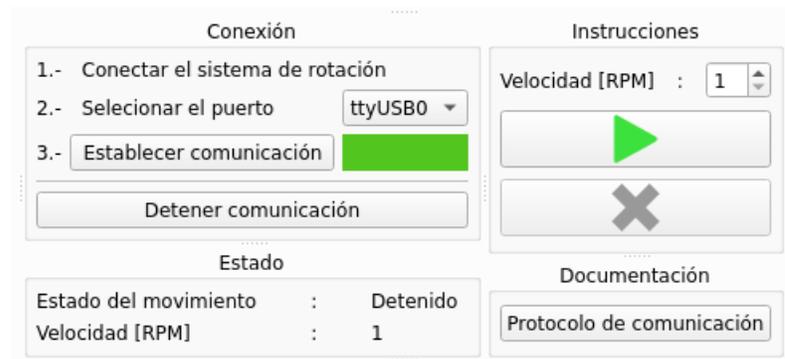


Figura 4.6: Pestaña encargada de la comunicación con el sistema de rotación. Esta pestaña es más pequeña que el resto, al igual que la anterior, por lo que nuevamente se han recortado los bordes para evitar espacios vacíos. En el puerto serial se encuentra conectado el sistema de rotación emulado por un Arduino, y como se observa ha sido configurado un giro de 1 RPM.

La pestaña del sistema de rotación es bastante similar a la del sistema de posicionamiento, dado que ambos poseen pocas instrucciones y además se idearon para funcionar sobre un mismo tipo de plataforma. Tal como se observa en la figura 4.6, esta pestaña está compuesta por 4 secciones, que se detallan a continuación.

Ventana: Conexión

Los pasos para establecer la conexión son 3: primero se debe conectar el sistema de rotación, luego se debe seleccionar el puerto en el que ha sido conectado, y finalmente se debe hacer click en “establecer comunicación”.

Cada 2 segundos se escanean automáticamente los puertos USB con el fin de detectar si el dispositivo está conectado o no. No obstante, el puerto en el cual está conectado no se puede deducir, es por ello que el usuario lo debe seleccionar de entre los que están disponibles.

Cuando la comunicación es establecida, se inicializa y se abre el puerto elegido mediante la librería `pyserial`, y se comienza a enviar de forma continua, cada 2 segundos, los comandos para actualizar el estado de la rotación y la velocidad actual.

Ventana: Documentación

Permite abrir el protocolo de comunicación (Anexo G) por si el usuario necesitara conocer detalles sobre la comunicación del sistema.

Ventana: Estado

Desde aquí se puede ver el estado actual del sistema de rotación:

- Estado del movimiento: Puede tomar dos valores, “moviendo” o “detenido”.
- Velocidad: Muestra la velocidad actual del sistema de rotación en RPM.

Ventana: Instrucciones

Desde este apartado se puede configurar la velocidad deseada del sistema de rotación, así como también comenzar y detener el giro.

El valor de la velocidad debe estar entre 1 y 99 RPM.

4.2.6 Generador de rutinas

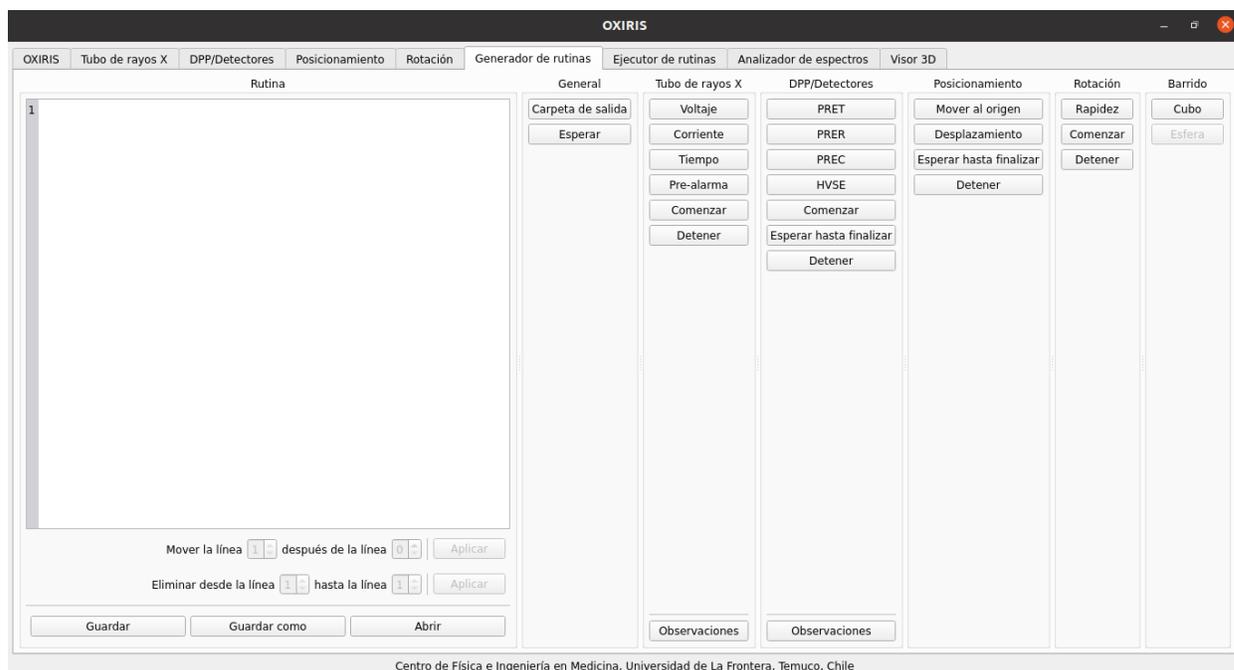


Figura 4.7: Pestaña encargada de generar las rutinas.

El generador de rutinas de la figura 4.7 permite generar un archivo de texto plano con extensión `.rout` que contiene una secuencia de instrucciones que ejecutará el software de forma automática.

Este archivo puede ser generado, editado, guardado y cargado desde la interfaz gráfica, que cuenta con una serie de validadores para evitar que existan errores que lleven a comportamientos inesperados.

Las instrucciones están compuestas por:

- Instrucciones de carácter general:
 - Carpeta de salida: Especifica la carpeta donde se guardarán todos los espectros.
 - Esperar: Permite que la ejecución de la rutina se pause por una determinada cantidad de segundos.
- Instrucciones para el tubo de rayos X:
 - Voltaje: Configura el voltaje del tubo.
 - Corriente: Configura la corriente del tubo.
 - Tiempo: Configura el tiempo de exposición.
 - Pre-alarma: Configura el tiempo de pre-alarma.
 - Comenzar: Comienza un disparo. Notar que la ejecución de la rutina continúa tras recibir esta instrucción.
 - Detener: Detiene el disparo actual.
- Instrucciones para los DPP/detectores: Las siguientes instrucciones se aplican a todos los DPP conectados al computador.
 - PRET: Configura el tiempo de adquisición.
 - PRER: Configura el tiempo real.
 - PREC: Configura la cantidad de cuentas a detectar.
 - HVSE: Configura el voltaje del detector.
 - Comenzar: Comienza la detección.
 - Esperar hasta finalizar: Pausa la rutina hasta que haya finalizado la detección.
 - Detener: Detiene la detección.
- Instrucciones para el sistema de posicionamiento:
 - Mover al origen: Mueve el sistema de posicionamiento al origen.
 - Desplazamiento: Configura y ejecuta un desplazamiento tridimensional.
 - Esperar hasta finalizar: Pausa la ejecución de la rutina hasta que se haya concretado el desplazamiento.
 - Detener: Detiene el desplazamiento actual.

- Instrucciones para el sistema de rotación:
 - Rapidez: Configura la rapidez del giro.
 - Comenzar: Comienza el giro.
 - Detener: Detiene el giro.
- Instrucciones para generar barridos: Con las instrucciones definidas previamente el usuario puede generar una gran cantidad de barridos. Con el fin de facilitar esta tarea, se desarrolla un algoritmo que genera barridos cúbicos, cuya ventana se puede ver en la figura 4.8. En el Anexo H se adjunta un ejemplo.



The screenshot shows a window titled "OXIRIS.py" with a standard Windows-style title bar (minimize, maximize, close). The window contains several input fields with up and down arrows, indicating they are spin boxes. The parameters are:

- Largo en x [mm] : 5
- Largo en y [mm] : 5
- Largo en z [mm] : 5
- Puntos en el eje x : 10
- Puntos en el eje y : 2
- Puntos en el eje z : 2
- Velocidad [mm/s] : 2

Below these fields, there are two lines of text:

- Máxima cantidad de puntos : 40
- Puntos totales : 24

At the bottom of the window is a button labeled "Aceptar".

Figura 4.8: Ventana para generar barridos cúbicos. Se deben especificar las dimensiones del cubo así como también la cantidad de puntos por eje y la velocidad a la que se moverá el sistema de posicionamiento. Con toda esta información, y considerando que el protocolo de comunicación del sistema de posicionamiento admite desplazamientos milimétricos enteros, se calcula la máxima cantidad de puntos así como también la cantidad de puntos totales o efectiva. Por ejemplo, en la imagen, la dimensión en x es de 5 mm y se pide que existan 10 puntos en este eje, pero como el máximo posible es 6 entonces el total de puntos es 24 y no 40.

4.2.7 Ejecutor de rutinas

El ejecutor de rutinas de la figura 4.9 tiene por objetivo leer un archivo de rutinas y ejecutar todas las instrucciones presentes, lo que se logra mediante la emulación de la interacción usuario-interfaz.

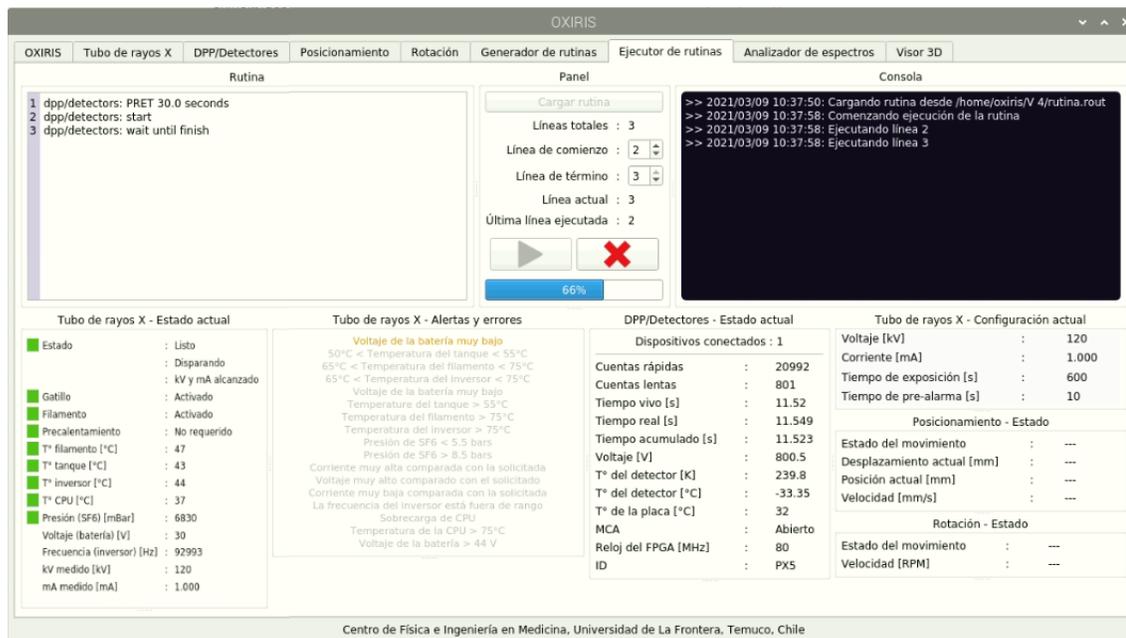


Figura 4.9: Pestaña encargada de ejecutar las rutinas. En la captura se está ejecutando una rutina simple de 3 líneas: configurar el tiempo de detección en 30 segundos, comenzar a detectar, y esperar hasta que la detección finalice. En paralelo el tubo de rayos X está disparando. Tanto el estado del tubo como del DPP/detector se muestran en tiempo real en la parte inferior. Se observa además que el tubo presenta la alerta de “voltaje de la batería muy bajo”.

Por ejemplo, si la instrucción es comenzar el disparo del tubo de rayos X, el software va hasta el botón ubicado en la pestaña correspondiente y lo presiona. Si el botón está inhabilitado, ya sea porque el tubo no está conectado o bien porque el seguro está activo, entonces el click fallará, lo que se mostrará en la consola ubicada al lado derecho.

Sucede lo mismo con el resto de instrucciones, lo que reduce el comportamiento del ejecutor de rutinas a sólo modificar valores, hacer clicks en botones y detectar cuando alguno de estos eventos falle.

Respecto a la ejecución, se puede elegir el intervalo de líneas a procesar, y además el software recuerda la última línea ejecutada correctamente, lo que permite retomar el trabajo si es que se interrumpe por factores externos.

Adicionalmente, en la zona inferior se muestra el estado en tiempo real del tubo de rayos X, los DPP/detectores, el sistema de posicionamiento y el sistema de rotación. Así, el usuario puede supervisar el funcionamiento de todo el sistema desde un mismo lugar.

4.2.8 Analizador de espectros

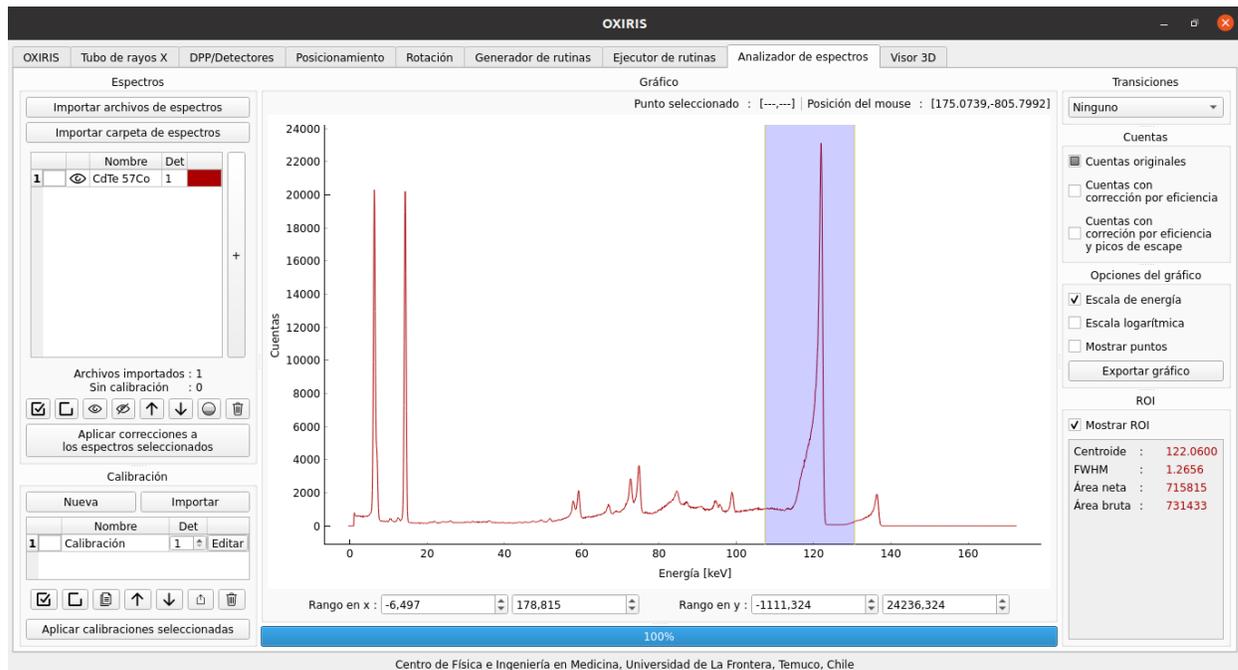


Figura 4.10: Espectro de Co-57 facilitado por AmpTek cargado en el analizador de espectros. En azul transparente se marca una región de interés. El espectro ha sido calibrando con los puntos (174, 14.4 keV) y (1450, 122.06 keV), obteniendo la función de ajuste $E = 0.0844C - 0.2809$.

El analizador de espectros de la figura 4.10 tiene por objetivo analizar, calibrar y corregir los espectros detectados. Está compuesto por 7 secciones, que se detallan a continuación.

Ventana: Espectros

Desde aquí se pueden cargar múltiples espectros en el formato desarrollado para este software (Anexo E).

Dado que importar una gran cantidad de archivos puede generar tiempos de espera muy elevados, se cuenta con una barra de progreso bajo el gráfico, que muestra el progreso tanto del trabajo de importación así como también de otras tareas.

Cada uno de los espectros cargados puede ser analizado en detalle utilizando el botón “+”, teniendo acceso a toda la información almacenada en el archivo.

Además, si los archivos se encuentran calibrados, pueden ser procesados para ser corregidos tanto por eficiencia del detector así como también por picos de escape (para más detalles ver

la sección 3.3 y 4.3).

Ventana: Calibración

En esta sección se pueden generar e importar archivos de calibración (Anexo C).

Para crear un archivo de calibración se debe hacer click en “nueva”, lo que añadirá un archivo de calibración estándar con función de ajuste

$$E = 1 \cdot C + 0 \quad (4.1)$$

donde E es la energía en keV y C es el canal.

Con el fin de personalizar esta calibración se dispone del botón “editar”, que abrirá la ventana de la figura 4.11, desde donde el usuario puede agregar, quitar y modificar todos los puntos que sean necesarios.

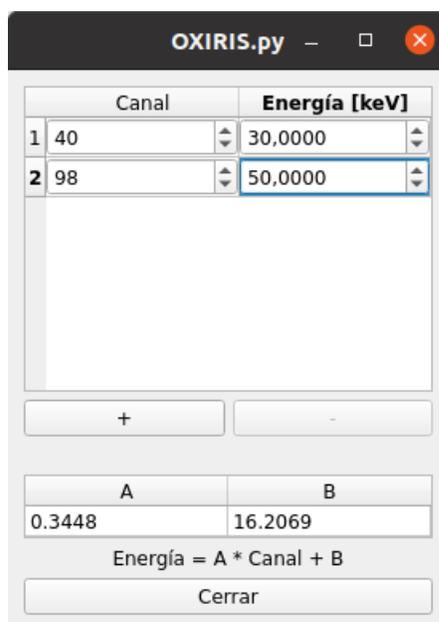


Figura 4.11: Ventana para editar calibraciones. El usuario puede agregar, quitar y modificar puntos. También puede ingresar manualmente el valor de A y B .

A medida que el usuario va modificando los datos se hace una regresión lineal con el fin de obtener la función que mejor se ajuste.

Finalmente, el usuario debe seleccionar todas las calibraciones que quiera aplicar, y hacer click en “aplicar calibraciones seleccionadas”, lo que calibrará todos los espectros cargados.

Ventana: Gráfico

Aquí el usuario puede visualizar el gráfico de todos los espectros, lo que permite realizar análisis visuales. Además, se muestran las coordenadas del mouse así como también las coordenadas de cualquier punto seleccionado, lo que hace más sencillo obtener los datos para generar calibraciones.

Ventana: Transiciones

Desde aquí el usuario puede visualizar en el gráfico las transiciones de los elementos químicos, desde el neón hasta el fermio, lo que sirve para identificar picos desconocidos.

Cada transición aparece como una línea vertical roja junto con el nombre de la transición en cuestión.

Ventana: Cuentas

Si es que el espectro ha sido corregido, desde esta sección el usuario puede elegir qué datos ver en el eje y : cuentas originales, cuentas con corrección por eficiencia, o cuentas por corrección por eficiencia más corrección por picos de escape.

Ventana: Opciones del gráfico

En esta sección se pueden realizar 4 acciones relacionadas al gráfico:

- Si es que los datos están calibrados, se puede elegir mostrar la energía en el eje x .
- Se puede activar la escala logarítmica para el eje y (los valores iguales o inferiores a 0 no se muestran).
- Se pueden mostrar los puntos del gráfico, lo que permite seleccionarlos y obtener sus coordenadas.
- Se puede exportar la vista actual del gráfico en formato `.png`.

Ventana: ROI (Región de Interés)

Desde esta sección se puede activar una región de interés (ROI) cuyos límites se pueden elegir interactivamente en el gráfico.

Para los datos que hay dentro del área se calcula el centroide, el ancho a media altura, el área neta y el área bruta:

- Centroide: Corresponde al punto medio en el eje x .
- Ancho a media altura:

Para calcularlo se asume que dentro del ROI existe un único pico y que el fondo corresponde a las cuentas que se ubican entre $y = 0$ y el valor mínimo en y dentro del ROI.

Si la primera condición no se cumple y existen varios picos, se calculará el ancho a media altura del primer pico de la izquierda. Si por el contrario no existen picos bien definidos, el valor calculado no tendrá ninguna interpretación.

Para realizar el cálculo primero se resta el fondo y luego se comienzan a analizar los puntos de izquierda a derecha, y se guarda la posición en x para los dos puntos cuyo valor en y sea más cercano al de la mitad del valor máximo en y .

Luego, se calcula el ancho a media altura como la diferencia entre estos dos valores.
- Área neta: Se calcula como la suma de todas las cuentas dentro del ROI, menos el fondo. El fondo se aproxima de igual forma que para el cálculo del ancho a media altura.
- Área bruta: Se calcula como la suma de todas las cuentas dentro del ROI.

4.2.9 Visor 3D

Desde esta pestaña se pueden cargar y procesar datos para obtener mapas tridimensionales de intensidad fluorescente.

El visor 3D ha sido probado usando espectros generados artificialmente, tal como se observa en la figura 4.12.

Las pruebas incluyeron datos unidimensionales, bidimensionales y tridimensionales. De todos, sólo los datos unidimensionales no se han podido visualizar de forma correcta, por lo que en este caso se desactivan todos los gráficos para evitar que el software presente errores. Esto es debido a que la librería utilizada está hecha para la visualización tridimensional, y no considera el caso unidimensional como un caso particular del tridimensional.

Para realizar las construcciones 3D se debe hacer lo siguiente:

- Cargar datos calibrados.
- Seleccionar el elemento químico cuya intensidad fluorescente se quiere observar. Se puede elegir desde el neón hasta el fermio.
- Indicar la transición del elemento químico seleccionado en la que se debe centrar el procesamiento.
- Especificar un intervalo de energía alrededor de la transición seleccionada.
- Elegir cómo se procesarán las cuentas dentro del intervalo seleccionado cuando haya

más de un espectro con las mismas coordenadas espaciales: utilizar aquel que tenga más cuentas, sumar las cuentas, o calcular el promedio.

- Para construir los gráficos tridimensionales se necesita una matriz cúbica de puntos equispaciados. En muchos casos los datos por sí solos no serán suficientes para formar esta matriz, por lo que se debe interpolar. El usuario puede elegir entre dos modos de interpolación: punto más cercano o lineal.
- Seleccionar si se utilizarán las cuentas originales o las corregidas.
- Generar los gráficos.

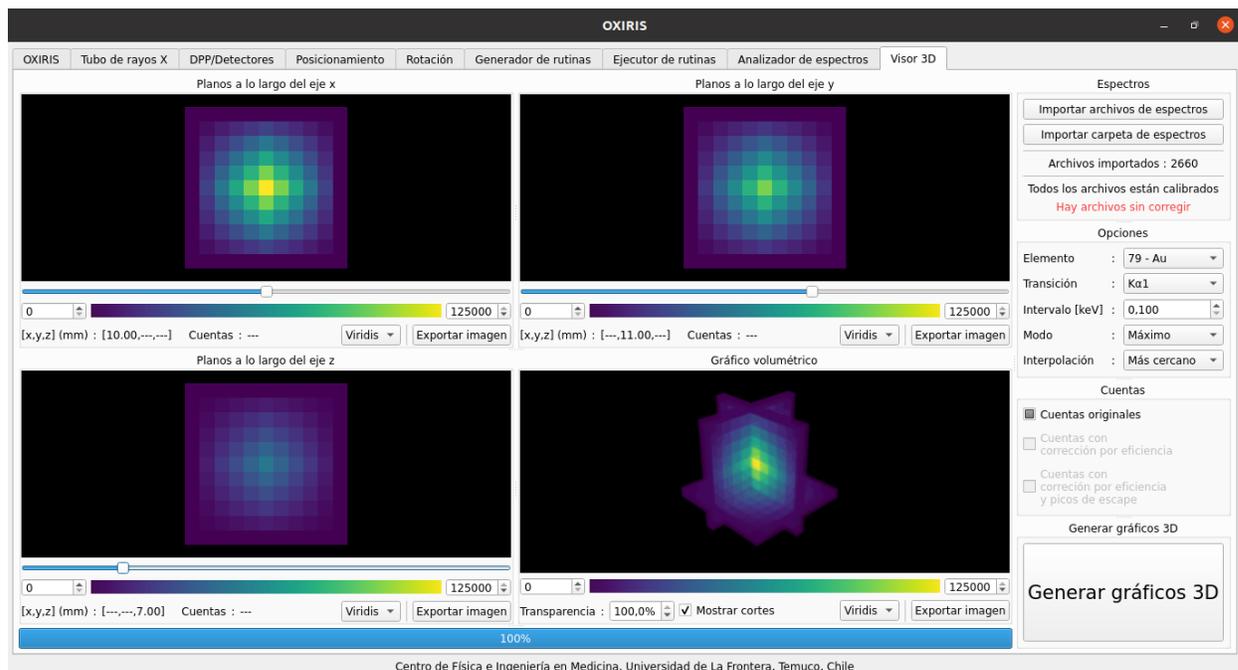


Figura 4.12: Construcciones tridimensionales hechas con el visor 3D, donde se muestra la intensidad de la fluorescencia de rayos X correspondiente a la transición K_{α_1} del oro. Los datos fueron generados artificialmente dentro de un cubo de $10 \times 10 \times 10$ mm, con origen en $(5,5,5)$ mm. La intensidad detectada en cada punto se ha especificado de forma manual, asignando un máximo de 125000 cuentas en el centro, que disminuyen radialmente hasta llegar a 0 en el borde. En la captura se observan distintos planos a lo largo de los 3 ejes cartesianos, y también el gráfico volumétrico, que se sincroniza con los planos que está visualizando el usuario.

El proceso de generación del gráfico consiste en identificar el intervalo seleccionado en cada espectro importado, integrar las cuentas, y asignar este valor al punto tridimensional extraído desde las coordenadas espaciales del espectro.

En el caso de existir dos o más espectros con la misma posición se procederá según haya especificado el usuario.

Finalmente, si se detecta que los datos no son suficientes para llenar una matriz cúbica equispaciada, se interpolan los datos las veces que sean necesarias para completar esta matriz.

El gráfico generado, que corresponde a la matriz de intensidades fluorescentes, se muestra de 4 formas distintas: planos a lo largo de cada eje y un gráfico volumétrico.

En los primeros 3 el usuario puede ver la cantidad de cuentas asociadas, así como también la ubicación espacial, de cada pixel sobre el cual pase el cursor.

Por otro lado en el gráfico volumétrico se puede cambiar el nivel de transparencia con el fin de ver los puntos interiores, y también se puede activar la superposición de los planos de las otras 3 vistas.

Con el fin de optimizar la visualización de las imágenes el usuario puede elegir entre 4 paletas de colores: *viridis*, *hot*, *jet* y *gray*, visibles en la figura 4.13.

Finalmente, todos los gráficos pueden ser exportados en formato *.png*.

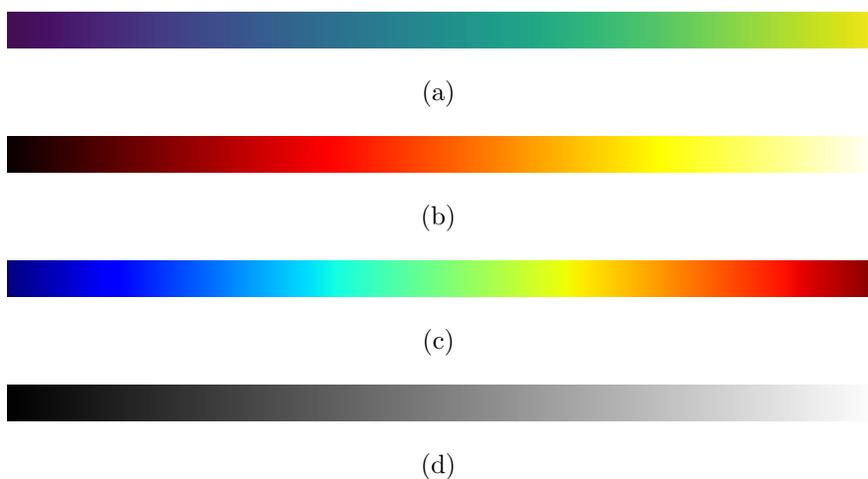


Figura 4.13: Paletas de colores utilizadas en el visor 3D: a) *viridis*, b) *hot*, c) *jet* y d) *gray*. Las intensidades menores toman los colores de la izquierda, mientras que las mayores los de la derecha. La paleta recomendada es *viridis*, pues su interpretación no cambia para las personas con daltonismo.

4.3 Correcciones de los Espectros EDXRF

4.3.1 Corrección por Eficiencia del Detector

Tal como se observa en la figura 4.14 las correcciones por eficiencia de detección incrementan la cantidad de cuentas para valores muy bajos de energía así como también para aquellos que están por sobre los 60 keV, lo que era esperable debido a la curva de la figura 3.11.

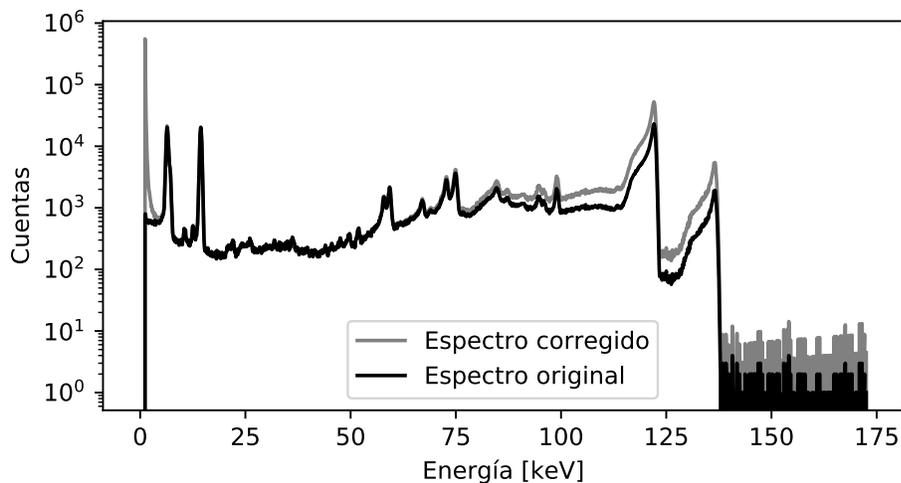
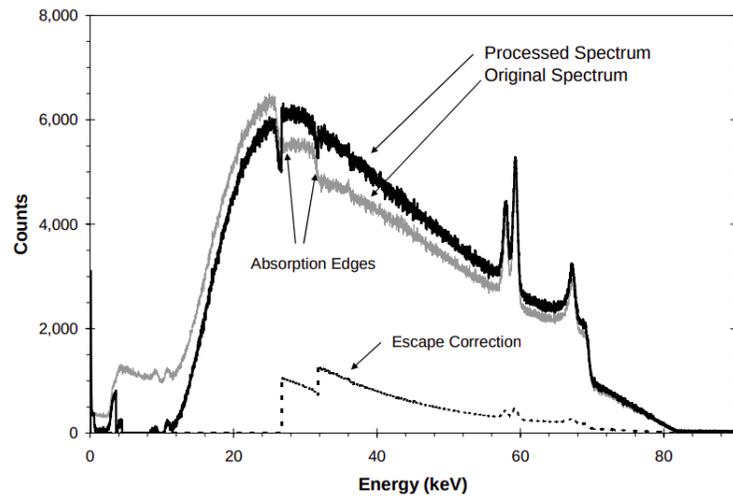


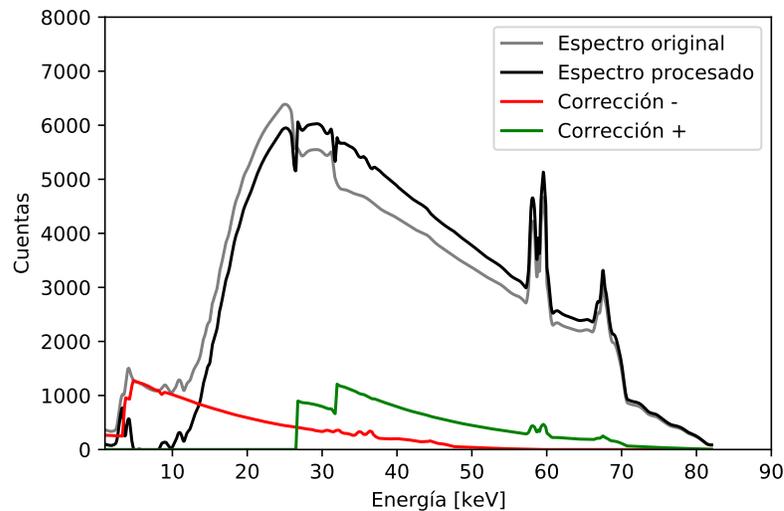
Figura 4.14: Corrección por eficiencia de detección del espectro mostrado en la figura 4.10.

4.3.2 Corrección por Picos de Escape

Con el fin de verificar que la rutina para corrección por picos de escape funciona correctamente, se utilizó el espectro de la figura 4 de Redus *et al.* (Redus *et al.*, 2009), al cual los autores también le aplicaron una corrección por picos de escape. La comparación se muestra en la figura 4.15, y se observa que los resultados son similares.



(a)



(b)

Figura 4.15: En a) se observa la figura 4 de Redus *et al* (Redus *et al.*, 2009), que corresponde al espectro de un tubo con ánodo de tungsteno medido con el mismo detector utilizado por OXIRIS. En negro se aprecia el espectro corregido con el algoritmo desarrollado por los autores. En b) se muestran los mismos datos, pero utilizando el algoritmo desarrollado en este trabajo. Como se observa, ambos resultados son similares, lo que valida el algoritmo aquí presentado. Respecto al espectro corregido, se identifica una zona donde la cantidad de cuentas disminuye, seguida de una zona donde la cantidad de cuentas aumenta, siendo el punto de separación aquel que corresponde a la energía del borde de absorción K del Cd.

Capítulo V
CONCLUSIONES

Capítulo V

Conclusiones

En este trabajo se ha demostrado que se puede escribir una interfaz gráfica compleja y funcional utilizando un lenguaje de programación de alto nivel como lo es Python, ya que cuenta con todas las características para comunicarse con los distintos periféricos así como también para procesar los datos obtenidos.

Esto ha permitido cumplir el objetivo general planteado en este trabajo, logrando desarrollar y optimizar el software requerido por OXIRIS, permitiendo el trabajo coordinado entre todos sus componentes, con el fin de irradiar confocalmente un objetivo y generar imágenes tridimensionales de la intensidad fluorescente asociada.

Además, este trabajo ha servido para poner en práctica gran parte de los conocimientos adquiridos en la carrera, demostrando la utilidad de un ingeniero físico en proyectos que involucren la física aplicada a diversas áreas de la ingeniería y la medicina.

5.1 Conclusiones Para el Primer Objetivo Específico

El software existente hasta antes de este trabajo de título consistía en los programas distribuidos por los fabricantes y en las pruebas de comunicación e interfaz gráfica realizadas durante la Práctica de Estudios II.

El presente desarrollo le dio continuidad a esto último, revisando todo el código y re escribiéndolo de forma más eficiente y organizada, lo que sentó las bases para poder continuar desarrollando el software.

Como resultado, se ha optimizado el desarrollo previo, cumpliendo así con este objetivo específico.

5.2 Conclusiones Para el Segundo Objetivo Específico

Los 17 archivos de código fuente escritos demostraron ser eficientes consiguiendo la comunicación con los periféricos, tal como se observa en las capturas de pantalla presentadas en la sección 4.2.

Esta comunicación se ha conseguido gracias a las librerías `pyusb` y `pyserial`, que se tomaron como base para escribir las rutinas específicas encargadas de interactuar con las piezas de OXIRIS.

5.3 Conclusiones Para el Tercer Objetivo Específico

Se escribieron dos algoritmos correctivos, uno por eficiencia del detector y otro por picos de escape.

En el primer caso se observó que el espectro no sufre modificaciones apreciables para energías ubicadas entre 10 y 75 keV. Sin embargo, para energías inferiores, la corrección da como resultado un aumento de la cantidad de cuentas en varios ordenes de magnitud, dado que la eficiencia del detector tiende a 0, mientras que para energías superiores la cantidad de cuentas aumenta en función de la energía. Dado que las energías de las transiciones $K\alpha_1$ y $K\alpha_2$ del oro son 68.8 keV y 66.9 keV, entonces se justifica la utilización del detector XR-100CdTe para el proyecto OXIRIS, pues según los resultados en este intervalo el espectro no sufre correcciones perceptibles, alcanzando una gran eficiencia, lo que permite omitir esta corrección para este caso en particular.

En cuando a la corrección por picos de escape, se observó que para energías superiores a 27 keV hay un aumento en la cantidad de cuentas, mientras que para energías inferiores la cantidad de cuentas disminuye. Esto tiene sentido, pues para energías menores a 27 keV, que corresponde al borde de absorción K para el Cd, no se generan fotones fluorescentes en el volumen del semiconductor, por lo tanto en esta zona el espectro se distorsiona únicamente debido a las cuentas extra que debieron ser contabilizadas en un canal superior. Por último, se ha comparado el resultado obtenido en esta sección con uno encontrado en la literatura, verificando que el algoritmo desarrollado funciona correctamente.

5.4 Conclusiones Para el Cuarto Objetivo Específico

La correcta documentación del código se ha logrado mediante la aplicación de buenas técnicas de programación, tales como realizar comentarios en todas las funciones y clases, así como también mediante la implementación de un desarrollo modular. Esto ha permitido que, utilizando la herramienta `doxygen`, se haya podido generar una documentación muy completa en formato `html`, que sin lugar a dudas será de utilidad en el futuro para añadir, ajustar o quitar características sin mayor esfuerzo, lo que puede servir, por ejemplo, frente a cambios de hardware.

5.5 Perspectivas Futuras

En el futuro cercano es necesario realizar pruebas presenciales para terminar de validar el software, así como también realizar la corrección necesaria para visualizar geometrías unidimensionales.

Por otro lado, el desarrollo aquí presentado da pie a una nueva versión, ya sea para el mismo prototipo de OXIRIS o para uno posterior, en el que el software esté totalmente optimizado para la detección de oro u otros elementos pesados, y centrado en identificar altas concentraciones en tiempo real.

Finalmente, la metodología y algoritmo desarrollado para la corrección por picos de escape es aplicable a muchos otros detectores en rangos de energía más amplios, por lo que se abre la posibilidad de trabajar en un software completamente dedicado a la corrección por picos de escape, con una base de datos dinámica, que vaya creciendo a medida que se realicen las simulaciones correspondientes.

Financiamiento

El desarrollo del software llevado a cabo en este proyecto de titulación no necesitó financiamiento adicional, pues lo esencial ya fue financiado por el proyecto FONDECYT 1171729.

Nomenclatura

3D	: Tridimensional.
CFIM	: Centro de Física e Ingeniería en Medicina.
CPU	: Unidad Central de Procesamiento (del inglés Central Processing Unit).
DPP	: Procesador Digital de Pulsos (del inglés Digital Pulse Processor).
EDXRF	: Fluorescencia de Rayos X Dispersiva por Energía (del inglés Energy-Dispersive X-Ray Fluorescence).
FONDECYT	: Fondo Nacional de Desarrollo Científico y Tecnológico.
GNU	: Acrónimo recursivo para GNU's Not Unix.
GUI	: Interfaz Gráfica (del inglés Graphical User Interface).
NIST	: Instituto Nacional de Estándares y Tecnología (de Estados Unidos) (del inglés National Institute of Standards and Technology).
OXIRIS	: Del inglés Orthovoltage X-Ray Induced Radiation Integral System.
PENELOPE	: Del inglés Penetration and Energy Loss of Positrons and Electrons.
RAM	: Memoria de Acceso Aleatorio (del inglés Random Access Memory).
RPM	: Revoluciones Por Minuto.
UFRO	: Universidad de La Frontera.
USB	: Bus Universal en Serie (del inglés Universal Serial Bus).
WDXRF	: Fluorescencia de Rayos X Dispersiva por Longitud de Onda (del inglés Wavelength-Dispersive X-Ray Fluorescence).
XRF	: Fluorescencia de Rayos X (del inglés X-Ray Fluorescence).

Bibliografía

- Attix, Frank. 2004. *Introduction to radiological physics and radiation dosimetry*. New York: Wiley-VCH.
- Baró, J., Sempau, J., Fernández-Varea, J.M., & Salvat, F. 1995. PENELOPE: An algorithm for Monte Carlo simulation of the penetration and energy loss of electrons and positrons in matter. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, **100**(1), 31–46.
- Bounakhla, M., & Tahri, Mounia. 2007. X-Ray Fluorescence Analytical Techniques.
- Delaney, Geoff, Jacob, Susannah, Featherstone, Carolyn, & Barton, Michael. 2005. The role of radiotherapy in cancer treatment. *Cancer*, **104**(6), 1129–1137.
- Eisberg, Robert. 1978. *Física cuántica : átomos, moléculas, sólidos, núcleos y partículas*. México: Limusa.
- Figuroa, R G, & Valente, M. 2015. Physical characterization of single convergent beam device for teletherapy: theoretical and Monte Carlo approach. *Physics in Medicine and Biology*, **60**(18), 7191–7206.
- Figuroa, R.G., Santibañez, M., Malano, F., & Valente, M. 2015. Optimal configuration for detection of gold nanoparticles in tumors using K β X-ray fluorescence line. *Radiation Physics and Chemistry*, **117**(Dec.), 198–202.
- Figuroa, R.G., Santibañez, M., & Valente, M. 2016. Dosimetric and bremsstrahlung performance of a single convergent beam for teletherapy device. *Physica Medica*, **32**(12), 1489–1494.
- Ford, Eric C., & Terezakis, Stephanie. 2010. How Safe Is Safe? Risk in Radiotherapy. *International Journal of Radiation Oncology Biology Physics*, **78**(2), 321–322.
- Guarda, Jaime. 2020. Desarrollo y caracterización de un sistema de detección de fluorescencia por rayos X en modalidad confocal en una matriz de tejido biológico equivalente de gran volumen. *Tesis de Magíster en Física Médica, Facultad de Ingeniería y Ciencias, Universidad de La Frontera*.
- Guarda, Jaime, Valente, Mauro, & Figuroa, Rodolfo. 2021. Development and characterisation of a confocal detection array for K-lines of heavy metals in big light matrix. *Radiation Physics and Chemistry*, **179**, 109116.

-
- Guttag, John. 2013. *Introduction to computation and programming using Python*. Cambridge, MA: The MIT Press.
- Hudson, Lawrence. 2003. *X-ray Transition Energies, NIST Standard Reference Database 128*.
- Jaklevic, J. M., & Goulding, F. S. 1971. Detection of Low Energy X Rays with Si(Li) Detectors. *IEEE Transactions on Nuclear Science*, **18**(1), 187–191.
- Kaye, G. W. C. 1934. Wilhelm Conrad Röntgen: and the Early History of the Roentgen Rays. *Nature*, **133**(3362), 511–513.
- Knoll, Glenn. 2010. *Radiation detection and measurement*. Hoboken, N.J.: John Wiley.
- Lindon, John. 2010. *Encyclopedia of spectroscopy and spectrometry*. Amsterdam: Elsevier.
- Mould, Richard. 1993. *A century of x-rays and radioactivity in medicine : with emphasis on photographic records of the early years*. Bristol Philadelphia: Institute of Physics Pub.
- Quilagüilque, Eduardo. 2019. Desarrollo de un sistema de barrido con control de posición y velocidad 3D asociadas a un sistema de detección y excitación para la obtención de imágenes biomédicas. *Facultad de Ingeniería y Ciencias, Universidad de La Frontera*.
- Redus, Robert H., Pantazis, John A., Pantazis, Thanos J., Huber, Alan C., & Cross, Brian J. 2009. Characterization of CdTe Detectors for Quantitative X-ray Spectroscopy. *IEEE Transactions on Nuclear Science*, **56**(4), 2524–2532.
- Ricketts, K, Guazzoni, C, Castoldi, A, Gibson, A P, & Royle, G J. 2013. An x-ray fluorescence imaging system for gold nanoparticle detection. *Physics in Medicine and Biology*, **58**(21), 7841–7855.
- Shafiq, Jesmin, Barton, Michael, Noble, Douglas, Lemer, Claire, & Donaldson, Liam J. 2009. An international review of patient safety measures in radiotherapy practice. *Radiotherapy and Oncology*, **92**(1), 15–21.
- Suit, Herman, Goldberg, Saveli, Niemierko, Andrzej, Ancukiewicz, Marek, Hall, Eric, Goitein, Michael, Wong, Winifred, & Paganetti, Harald. 2007. Secondary Carcinogenesis in Patients Treated with Radiation: A Review of Data on Radiation-Induced Cancers in Human, Non-human Primate, Canine and Rodent Subjects. *Radiation Research*, **167**(1), 12–42.

ANEXO A

TABLA DE RAYOS X CARACTERÍSTICOS

Anexo A

Tabla de Rayos X Característicos

Tabla A.1: Base de datos de transiciones y rayos X característicos (en keV) utilizada en este trabajo (Hudson, 2003).

N	Elemento	K_{α_1}	K_{α_2}	K_{β_1}	K_{β_2}	L_{α_1}	L_{α_2}	L_{β_1}	L_{β_2}
10	Ne	0.849	0.849	0.858	—	—	—	—	—
11	Na	1.041	1.041	1.071	—	—	—	—	—
12	Mg	1.254	1.253	1.302	—	—	—	—	—
13	Al	1.487	1.486	1.558	—	—	—	—	—
14	Si	1.740	1.739	1.836	—	—	—	—	—
15	P	2.014	2.013	2.139	—	—	—	—	—
16	S	2.308	2.307	2.464	—	—	—	—	—
17	Cl	2.622	2.621	2.816	—	—	—	—	—
18	Ar	2.958	2.956	3.190	—	—	—	—	—
19	K	3.314	3.311	3.590	—	—	—	—	—
20	Ca	3.692	3.688	4.013	—	0.341	0.341	0.345	—
21	Sc	4.091	4.086	4.460	—	0.395	0.395	0.400	—
22	Ti	4.511	4.505	4.932	—	0.452	0.452	0.458	—
23	V	4.952	4.945	5.427	—	0.511	0.511	0.519	—
24	Cr	5.415	5.406	5.947	—	0.573	0.573	0.583	—
25	Mn	5.899	5.888	6.491	—	0.637	0.637	0.649	—
26	Fe	6.404	6.391	7.058	—	0.705	0.705	0.718	—
27	Co	6.930	6.916	7.649	—	0.776	0.776	0.791	—
28	Ni	7.478	7.461	8.265	—	0.851	0.851	0.869	—
29	Cu	8.048	8.028	8.905	—	0.930	0.930	0.950	—

30	Zn	8.639	8.616	9.572	9.658	1.012	1.012	1.035	—
31	Ga	9.252	9.225	10.264	10.366	1.098	1.098	1.125	—
32	Ge	9.887	9.855	10.982	11.101	1.188	1.188	1.218	—
33	As	10.543	10.508	11.726	11.864	1.282	1.282	1.317	—
34	Se	11.223	11.182	12.496	12.652	1.379	1.379	1.419	—
35	Br	11.924	11.878	13.292	13.470	1.480	1.480	1.526	—
36	Kr	12.648	12.595	14.113	14.315	1.585	1.585	1.637	—
37	Rb	13.395	13.336	14.961	15.186	1.694	1.693	1.752	—
38	Sr	14.165	14.098	15.836	16.085	1.807	1.805	1.872	—
39	Y	14.959	14.883	16.738	17.016	1.923	1.920	1.996	—
40	Zr	15.775	15.691	17.667	17.970	2.042	2.040	2.124	2.219
41	Nb	16.615	16.521	18.623	18.953	2.166	2.163	2.257	2.367
42	Mo	17.479	17.374	19.608	19.965	2.293	2.290	2.395	2.518
43	Tc	18.367	18.251	20.619	21.005	2.424	—	2.537	—
44	Ru	19.279	19.150	21.657	22.074	2.559	2.554	2.683	2.836
45	Rh	20.216	20.074	22.724	23.173	2.697	2.692	2.834	3.001
46	Pd	21.177	21.020	23.819	24.299	2.839	2.833	2.990	3.172
47	Ag	22.163	21.990	24.942	25.457	2.984	2.978	3.151	3.348
48	Cd	23.174	22.984	26.095	26.644	3.134	3.127	3.317	3.528
49	In	24.210	24.002	27.276	27.861	3.287	3.279	3.487	3.714
50	Sn	25.271	25.044	28.486	29.110	3.444	3.437	3.663	3.905
51	Sb	26.359	26.111	29.726	30.390	3.605	3.595	3.844	4.101
52	Te	27.473	27.202	30.996	31.701	3.769	3.759	4.030	4.302
53	I	28.612	28.318	32.295	33.042	3.938	3.926	4.221	4.508
54	Xe	29.779	29.458	33.624	34.415	4.110	4.097	4.418	—
55	Cs	30.973	30.625	34.987	35.822	4.286	4.272	4.620	4.936
56	Ba	32.193	31.817	36.377	37.258	4.466	4.451	4.828	5.157
57	La	33.442	33.034	37.801	38.730	4.651	4.634	5.042	5.383
58	Ce	34.720	34.279	39.258	40.233	4.840	4.823	5.263	5.613
59	Pr	36.027	35.551	40.749	41.774	5.034	5.014	5.489	5.850
60	Nd	37.361	36.848	42.271	43.335	5.230	5.208	5.721	6.091
61	Pm	38.725	38.172	43.825	44.937	5.433	5.408	5.962	6.339
62	Sm	40.118	39.523	45.413	46.575	5.636	5.609	6.204	6.587

63	Eu	41.543	40.902	47.038	48.257	5.846	5.817	6.456	6.842
64	Gd	42.997	42.309	48.697	49.961	6.057	6.025	6.713	7.103
65	Tb	44.483	43.745	50.383	51.724	6.273	6.238	6.978	7.367
66	Dy	45.999	45.208	52.120	53.510	6.495	6.458	7.248	7.636
67	Ho	47.547	46.700	53.877	55.325	6.720	6.678	7.526	7.911
68	Er	49.127	48.222	55.674	57.214	6.948	6.904	7.810	8.189
69	Tm	50.741	49.773	57.509	59.095	7.180	7.134	8.102	8.469
70	Yb	52.389	51.355	59.367	60.985	7.416	7.367	8.402	8.759
71	Lu	54.070	52.966	61.283	62.967	7.656	7.605	8.709	9.049
72	Hf	55.791	54.612	63.234	64.980	7.899	7.845	9.023	9.347
73	Ta	57.533	56.278	65.223	67.013	8.146	8.088	9.343	9.652
74	W	59.319	57.982	67.245	69.101	8.398	8.335	9.673	9.964
75	Re	61.141	59.719	69.310	71.232	8.653	8.586	10.010	10.275
76	Os	63.001	61.487	71.414	73.403	8.912	8.841	10.355	10.599
77	Ir	64.896	63.287	73.562	75.619	9.175	9.100	10.708	10.920
78	Pt	66.833	65.123	75.749	77.878	9.442	9.362	11.071	11.251
79	Au	68.805	66.991	77.980	80.186	9.713	9.628	11.442	11.585
80	Hg	70.820	68.895	80.254	82.545	9.989	9.898	11.823	11.924
81	Tl	72.873	70.832	82.577	84.948	10.269	10.173	12.213	12.272
82	Pb	74.970	72.805	84.939	87.367	10.552	10.450	12.614	12.623
83	Bi 209	77.109	74.816	87.344	89.862	10.839	10.731	13.024	12.980
84	Po 209	79.293	76.864	89.797	92.400	11.131	11.016	13.447	13.341
85	At 210	81.514	78.944	92.304	94.991	11.427	11.305	13.876	—
86	Rn 222	83.783	81.066	94.867	97.639	11.727	11.598	14.316	—
87	Fr 223	86.105	83.232	97.478	100.326	12.031	11.895	14.770	14.450
88	Ra 226	88.476	85.436	100.136	103.045	12.340	12.196	15.236	14.842
89	Ac 227	90.885	87.676	102.847	105.868	12.652	12.501	15.713	—
90	Th 232	93.347	89.957	105.602	108.718	12.968	12.809	16.202	15.624
91	Pa 231	95.866	92.283	108.417	111.625	13.291	13.122	16.702	16.025
92	U 233	98.434	94.653	—	—	—	—	—	—
92	U 238	98.432	94.651	111.295	114.607	13.615	13.439	17.220	16.428
93	Np 237	101.056	97.068	114.243	117.569	13.944	13.760	17.750	16.840
94	Pu 239	103.734	99.523	117.244	—	14.279	14.084	18.294	17.255

94	Pu 244	103.740	99.529	117.232	120.674	—	—	—	—
95	Am 241	106.471	102.030	120.281	123.817	14.617	14.412	18.852	17.677
95	Am 243	106.473	102.031	120.279	123.816	—	—	—	—
96	Cm 245	109.273	104.589	123.407	126.982	—	—	—	—
96	Cm 248	109.272	104.590	123.403	127.039	—	—	—	—
97	Bk 249	112.127	107.194	126.577	—	—	—	—	—
97	Bk 250	112.111	107.164	126.582	—	—	—	—	—
98	Cf 249	—	—	—	—	—	—	—	—
98	Cf 250	115.035	109.837	129.845	133.675	—	—	—	—
98	Cf 251	115.067	109.860	—	—	15.654	15.434	20.656	—
99	Es 251	118.018	112.501	133.188	—	—	—	—	—
100	Fm 254	121.095	115.319	136.555	140.592	16.384	16.122	21.898	19.879

ANEXO B

**CALIBRACIÓN DE ESPECTROS
DE FLUORESCENCIA**

Anexo B

Calibración de Espectros de Fluorescencia

Los DPP separan los fotones detectados, que reciben el nombre de **cuentas**, en función de su energía. Típicamente la cantidad de intervalos es 256, 512, 1024, 2048, 4096 u 8192, lo que recibe el nombre de **canales**.

De esta forma, el espectro en bruto que resulta de cualquier detector corresponde a una serie de datos de cuentas v/s canales.

Con el fin de establecer una equivalencia entre los canales y la energía asociada se debe hacer una **calibración**, que consiste en identificar 2 o más picos cuya energía es conocida, pudiendo de esta forma generar una función lineal que relacione los canales con la energía.

Por ejemplo, si un pico en el canal 40 corresponde a 30 keV, y otro en el canal 98 corresponde a 50 keV, entonces la función de calibración corresponde a la función que pasa por ambos puntos:

$$E = 0.34 \cdot C + 16.2 \tag{B.1}$$

donde E es la energía en keV y C es el canal en cuestión.

Para calibrar todo el espectro se debe aplicar la función de calibración a cada uno de los puntos.

ANEXO C

**FORMATO DEL ARCHIVO DE
CALIBRACIÓN**

Anexo C

Formato del Archivo de Calibración

Los archivos de calibración se pueden generar desde la pestaña “anализador de espectros”, que permite ver en detalle cada punto con el fin de generar una calibración más fina.

La función de calibración tiene la forma

$$E = AC + B \tag{C.1}$$

donde E es la energía en keV, C es el canal, A es la pendiente y B es la intersección con el eje y .

El archivo de calibración se ha definido como un texto plano con extensión `.cal` que posee el siguiente formato:

- Línea 1: Letra A , seguida de un espacio y el valor del parámetro A .
- Línea 2: Letra B , seguida de un espacio y el valor del parámetro B .
- Línea 3: Palabra *det*, seguida de un espacio y el número del detector asociado.
- Resto de las líneas: En cada línea, separados por un espacio, se encuentran los valores de los canales y su energía asociada, que han sido utilizados para calcular los parámetros A y B .

ANEXO D

**FORMATO DEL ARCHIVO DE
PARÁMETROS**

Anexo D

Formato del Archivo de Parámetros

Con el fin de guardar los valores de los parámetros de los DPP para ser utilizados nuevamente en el futuro, se ha definido un archivo de parámetros.

Consiste en un texto plano con extensión `.dat` que posee el nombre de cada parámetro, seguido del símbolo *es igual*, el valor, y finalmente un *punto y coma*.

El orden en el que aparecen los parámetros en este archivo de texto ha sido elegido cuidadosamente para cumplir con los requisitos presentados en el protocolo de comunicación.

Un ejemplo de archivo de parámetros, separado en 4 columnas, es el siguiente:

PRET=0.0;	MCSH=8191.0;	CUSP=OFF;	RTDE=OFF;
PREC=OFF;	MCST=0.0;	DACF=0.0;	RTDD=1.0;
PRER=OFF;	SOFF=OFF;	DACO=OFF;	RTDT=0.0;
MCAC=1024;	THSL=0.0;	GATE=OFF;	RTDW=1.0;
TECS=OFF;	THFA=0.0;	GPED=FALLING;	SCAW=100;
HVSE=OFF;	RTDS=0.0;	GPGA=ON;	SCOE=RISING;
GAIA=1.0;	PAPZ=OFF;	GPIN=AUX1;	SCOG=1;
CLCK=AUTO;	AINP=NEG;	GPMC=ON;	SCOT=87;
TPEA=0.8;	AUO1=ICR;	GPME=ON;	TLLD=OFF;
GAIN=0.75;	AUO2=ICR;	INOG=LOW;	TFLA=0.013;
GAIF=0.5;	BLRD=0;	INOF=DEF;	TPFA=50;
VOLU=OFF;	BLRM=OFF;	MCAE=OFF;	TPMO=OFF;
MCAS=NORM;	BLRU=0;	PAPS=8.5;	
PRCL=0.0;	BOOT=OFF;	PDMD=NORM;	
PRCH=8191.0;	CON1=DAC;	PURE=OFF;	
MCSL=0.0;	CON2=AUXOUT2;	RESL=OFF;	

ANEXO E

**FORMATO DEL ARCHIVO DE
ESPECTRO**

Anexo E

Formato del Archivo de Espectro

AmpTek posee un archivo de espectro con extensión `.mca`, el cual ha sido utilizado de base para generar un nuevo archivo de espectro con extensión `.mcax`.

Se muestra el siguiente ejemplo (separado en 3 columnas) con el fin de explicar su formato:

Fast counts: 5635	FPGA clock [MHz]: 80	...
Slow counts: 230	Device ID: PX5	«ENERGY»
Accumulated time [s]: 3.356	Detector: 1	...
Live time [s]: 3.355	Position: 2 , 3 , 4	«END»
Real time [s]: 3.361	«DATA»	«DP5 CONFIGURATION»
Firmware version: 6.8	...	PRET=OFF;
FPGA version: 6.11	«END»	PREC=OFF;
Serial number: 2543	«DATA EFFICIENCY»	PRER=OFF;
High voltage [V]: 800.5
Detector temperature [K]: 239.6	«END»	...
Detector temperature [°C]: -	«DATA ESCAPE PEAKS»	«DP5 CONFIGURATION
33.549	...	END»
Board temperature [°C]: 30	«END»	

Como se observa, en la primera parte se especifican algunos detalles, tanto del proceso de detección como del estado del DPP/detector, así como también la posición del sistema de posicionamiento.

Entre «DATA» y «END» se encuentran todas las cuentas detectadas, ordenadas verticalmente y separadas mediante un salto de línea.

Entre «DATA EFFICIENCY» y «END» se encuentran todas las cuentas detectadas corregidas por eficiencia del detector, ordenadas verticalmente y separadas mediante un salto de línea. Esta sección estará vacía si es que aún no se ha realizado la corrección.

Entre «DATA ESCAPE PEAKS» y «END» se encuentran todas las cuentas detectadas corregidas por eficiencia del detector y por picos de escape, ordenadas verticalmente y separadas mediante un salto de línea. Esta sección estará vacía si es que aún no se han realizado las

correcciones.

Entre «ENERGY» y «END» se encuentran las energías correspondientes a cada canal, ordenadas verticalmente y separadas mediante un salto de línea. Esta sección estará vacía si es que el espectro no ha sido calibrado.

Entre «DP5 CONFIGURATION» y «DP5 CONFIGURATION END» se encuentran todos los parámetros con sus respectivos valores, según se explica en el Anexo D.

ANEXO F

PROTOCOLO DE COMUNICACIÓN DEL SISTEMA DE POSICIONAMIENTO

Anexo F

Protocolo de Comunicación del Sistema de Posicionamiento

A partir de la siguiente página se adjunta el protocolo de comunicación creado para el sistema de posicionamiento.

Communication protocol - Positioning system

1 General specifications

- The microcontroller must be an Arduino.
- Serial communication of 9600 bauds.
- The delay between a request packet and a response packet must be lower than 5 seconds.
- The positioning system must be able to move in the x , y , z axes at a speed specified by the user.

2 Commands

DIS \pm xxx \pm yyy \pm zzz\n

Displace \pm xxx mm in the x direction, \pm yyy mm in the y direction, and \pm zzz mm in the z direction.

DIS \pm xxx \pm yyy \pm zzz\n

Sended when the **DIS \pm xxx \pm yyy \pm zzz\n** packet is correctly received.

DIS?\n

To know the current state of the positioning system.

DIS X \n

Sends the status of the positioning system. X must be **MOVING** or **STOPEED**.

POS?\n

To know the current position with respect to the origin.

POS \pm xxx \pm yyy \pm zzz\n

Sends back the current position in mm with respect to the origin.

SET_ORIGIN\n

Set the current position as the origin.

ORIGIN!\n

Sended when the **SET_ORIGIN\n** packet is correctly received.

SPE ss \n

Configures the speed as ss mm/s.

SPE ss \n

Sended when the **SPE ss \n** is correctly received and setted.

STOP\n

To stop a displacement

STOP!\n

Sended when the movement finish by the **STOP\n** command.

3 Proposed Arduino Code

The following code is a proposed one to read the commands sended from the computer to the Arduino:

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  read_packet();
}

// Delays are required.
// Do not use readString, is a slow function.
void read_packet(){
  String data = "";
  char data_byte;
  int n1, n2, n3;
  while (Serial.available() > 0){
    data_byte = Serial.read();
    data += data_byte;
    delay(1);
  }
  if (data.startsWith("DIS_")){
    n1 = data.substring(4, 8).toInt();
    n2 = data.substring(9, 13).toInt();
    n3 = data.substring(14, 18).toInt();
    Serial.print(data + "\n");
    // Process
  }
  else if (data.startsWith("DIS?")){
    // Process
    Serial.print("DIS_STOPPED\n");
    //Serial.print("DIS_MOVING\n");
  }
  else if (data.startsWith("POS?")){
    // Process
    Serial.print("POS_-001_+002_-003\n");
  }
  else if (data.startsWith("SET")){
    // Process
    Serial.print("ORIGIN!\n");
  }
  else if (data.startsWith("SPE_")){
    n1 = data.substring(4, 6).toInt();
    Serial.print(data + "\n");
    // Process
  }
  else if (data.startsWith("STOP")){
    Serial.print("STOP!\n");
    // Process
  }
  delay(3);
}
```

ANEXO G

PROTOCOLO DE COMUNICACIÓN DEL SISTEMA DE ROTACIÓN

Anexo G

Protocolo de Comunicación del Sistema de Rotación

A partir de la siguiente página se adjunta el protocolo de comunicación creado para el sistema de rotación.

Communication protocol - Rotation system

1 General specifications

- The microcontroller must be an Arduino.
- Serial communication of 9600 bauds.
- The delay between a request packet and a response packet must be lower than 5 seconds.
- The rotation system must be able to rotate at a speed specified by the user.

2 Commands

RPM_xx\n

Set the rotation speed as **xx** RPM (revolutions per minute).

RPM_xx\n

Sended when the **RPM_xx\n** packet is correctly received.

START\n

To start the rotation.

START!\n

Sended when the rotation begins by the **START\n** command.

STOP\n

To stop the rotation.

STOP!\n

Sended when the rotation finish by the **STOP\n** command.

ROT?\n

To know the rotation status.

ROT_X\n

Sends the status of the current rotation. **X** must be **ROTATING** or **STOPPED**.

3 Proposed Arduino Code

The following code is a proposed one to read the commands sented from the computer to the Arduino:

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  read_packet();
}

// Delays are required.
// Do not use readString, is a slow function.
void read_packet(){
  String data = "";
  char  data_byte;
  int   n;
  while (Serial.available() > 0){
    data_byte = Serial.read();
    data += data_byte;
    delay(1);
  }
  if (data.startsWith("RPM_")){
    n = data.substring(4, 6).toInt();
    Serial.print(data + "\n");
    // Process
  }
  else if (data.startsWith("START")){
    Serial.print("START!\n");
    // Process
  }
  else if (data.startsWith("STOP")){
    Serial.print("STOP!\n");
    // Process
  }
  else if (data.startsWith("ROT?")){
    // Process
    //Serial.print("ROT_ROTATING\n");
    Serial.print("ROT_STOPPED\n");
  }
  delay(3);
}
```

ANEXO H

EJEMPLO DE RUTINA CON BARRIDO CÚBICO

Anexo H

Ejemplo de Rutina con Barrido Cúbico

A continuación se muestra la rutina generada para un barrido cúbico de dimensiones 5x5x5 mm, con 3x2x2 puntos:

```
positioning system: speed 2 mm/s
positioning system: move to origin
positioning system: displacement 0 0 5 mm
positioning system: wait until finish
dpp/detectors: start
dpp/detectors: wait until finish
positioning system: displacement 0 5 -5 mm
positioning system: wait until finish
dpp/detectors: start
dpp/detectors: wait until finish
positioning system: displacement 0 0 5 mm
positioning system: wait until finish
dpp/detectors: start
dpp/detectors: wait until finish
positioning system: displacement 2 -5 -5 mm
positioning system: wait until finish
dpp/detectors: start
dpp/detectors: wait until finish
positioning system: displacement 0 0 5 mm
positioning system: wait until finish
dpp/detectors: start
dpp/detectors: wait until finish
```

positioning system: displacement 0 5 -5 mm

positioning system: wait until finish

dpp/detectors: start

dpp/detectors: wait until finish

positioning system: displacement 0 0 5 mm

positioning system: wait until finish

dpp/detectors: start

dpp/detectors: wait until finish

positioning system: displacement 3 -5 -5 mm

positioning system: wait until finish

dpp/detectors: start

dpp/detectors: wait until finish

positioning system: displacement 0 0 5 mm

positioning system: wait until finish

dpp/detectors: start

dpp/detectors: wait until finish

positioning system: displacement 0 5 -5 mm

positioning system: wait until finish

dpp/detectors: start

dpp/detectors: wait until finish

positioning system: displacement 0 0 5 mm

positioning system: wait until finish

dpp/detectors: start

dpp/detectors: wait until finish

ANEXO I

**INPUT PARA LA SIMULACIÓN DE
50 KEV**

Anexo I

Input para la Simulación de 50 keV

```
TITLE 50.0e3 keV
.
>>>>>>> Source definition.
SKPAR 2 [Primary particles: 1=electron, 2=photon, 3=positron]
SENERG 50.0e3 [Initial energy (monoenergetic sources only)]
SPOSIT 0 0 0.3 [Coordinates of the source]
SCONE 180 0 0 [Conical beam; angles in deg]
.
>>>>>>> Material data and simulation parameters.
MFNAME ../CdTe.mat [Material file, up to 20 chars]
MFNAME ../air.mat [Material file, up to 20 chars]
.
>>>>>>> Geometry and local simulation parameters.
GEOMFN ../geometry.geo [Geometry file, up to 20 chars]
DSMAX 1 1e-4 [KB, maximum step length in body KB]
DSMAX 1 1e-4 [KB, maximum step length in body KB]
.
>>>>>>> Emerging particles. Energy and angular distributions.
NBE 0 50.0e3 2000 [Energy window and no. of bins]
NBANGL 1 1 [No. of bins for the angles THETA and PHI]
.
>>>>>>> Impact detectors (up to 25 different detectors).
IPSF=0; no psf is created.
IPSF=1; a psf is created (for only one detector).
```

IDCUT=0; tracking is discontinued at the detector entrance.

IDCUT=1; the detector does not affect the tracking.

IDCUT=2; the detector does not affect tracking, the energy distribution of particle fluence (integrated over the volume of the detector) is calculated.

IMPDET 0 50.0e3 2000 0 0 [E-window, no. of bins, IPSF, IDCUT]

IDBODY 2 [Active body]

.

>>>>>>> Job properties

RESUME dump.dat [Resume from this dump file, 20 chars]

DUMPTO dump.dat [Generate this dump file, 20 chars]

DUMPP 30 [Dumping period, in sec]

.

NSIMSH 1e7 [Desired number of simulated showers]

TIME 2e9 [Allotted simulation time, in sec]

.

END

ANEXO J

GEOMETRÍA UTILIZADA EN LAS SIMULACIONES

