



Universidad Autónoma del Estado de México
Centro Universitario UAEM Atlacomulco

ADQUISICIÓN DE ESPECTROS ÓPTICOS PARA LA ESTIMACIÓN
DE TEMPERATURA ELECTRÓNICA

T E S I S

QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A

Ing. Arturo Garduño Aparicio

TUTOR ACADÉMICO

Dr. Carlos Eduardo Torres Reyes

TUTORES ADJUNTOS

Dr. José Arturo Pérez Martínez

Dr. Allan Antonio Flores Fuentes

ATLACOMULCO, ESTADO DE MÉXICO, DICIEMBRE 2015.

Para la realización de esta tesis se contó con el apoyo de una beca del CONACYT.

AGRADECIMIENTOS

En primer lugar quiero darle gracias a Dios por la vida, por ser mi fortaleza en los momentos de debilidad y angustia y por permitirme culminar con esta meta tan llena de dificultades, de alegría y bendiciones.

Agradezco a mis padres, que con su infinito amor y cariño me han guiado a lo largo de mi vida, gracias por su ejemplo, por su apoyo y comprensión. Gracias Mariano y Lulú

A mis hermanos Martha, Mariano e Isabel, por ser los eternos compañeros que siempre han estado en los peores y mejores momentos de mi vida.

Quiero también agradecerte a ti mi Joss, por estar conmigo en todo momento brindándome tu amor, cariño y apoyo en estas largas jornadas, gracias por ser mi compañera. Gracias mi amor.

Agradezco el apoyo y dedicación que mis maestros y compañeros me han regalado para lograr mi formación, en especial al Dr. Carlos por la confianza que deposito en mí, por su tiempo, conocimientos y paciencia.

Del mismo modo, agradecer el apoyo otorgado que me brindó CONACYT.

ÍNDICE

Introducción	6
Estado del Arte	8
Planteamiento del Problema	10
Objetivos	11
Hipótesis.....	11
Alcances.....	11
Limitaciones	12
CAPITULO 1. MARCO TEÓRICO.....	13
1.1 Plasma	13
1.2 Espectroscopia	14
1.3 Ecuación de Boltzmann para la Estimación de Temperatura Electrónica.....	15
1.4 Metodologías para el desarrollo de software	17
1.4.1 Modelo en cascada	17
1.4.2 Modelo en V	18
1.4.3 Modelo de construcción de Prototipos.....	18
1.4.4 Modelo en Espiral.....	19
1.4.5 Modelo Incremental.....	20
1.5 Lenguaje de modelado de sistemas SysML.....	21
1.5.1 Casos de Uso	22
1.5.2 Diagrama de actividades	23
1.5.3 Diagrama de Interacción	24
1.5.4 Diagrama de Secuencia	24
1.5.5 Diagrama de requerimientos	24
1.5.6 Diagrama de paquetes	25
1.6 Programación Orientada a Objetos (POO).....	26
1.7Java.....	26
1.8 Librería JFreeChart	26
1.9 Librería JExcelApi (JXL)	27
1.10 Filtro media móvil	28
CAPITULO 2. MODELADO DEL SISTEMA.....	29
2.1 Diagrama de Requerimientos.....	30
2.2 DIAGRAMA DE PAQUETES.....	30

2.3 Diagrama de Casos de Uso	31
2.4 Diagrama de Actividades.....	33
2.5 Diagrama de Interacción	36
CAPÍTULO 3. DISEÑO DEL SISTEMA	38
3.1 Descripción General de la Aplicación	38
3.1 Formulario Principal de Actividades del Sistema	38
3.2 Módulo Traer Tabla de Elemento	38
3.3 Módulo de Análisis y Visualización de Espectro.....	40
3.5 Búsqueda de Espectros y Aplicación de Filtro.....	41
3.7 Pantalla de análisis y visualización de espectros	43
CAPITULO 4. PRUEBAS Y RESULTADOS	49
4.1 Pruebas de Estimación de Temperatura Electrónica sin Filtro Media Móvil	50
4.2 Pruebas de Estimación de Temperatura Electrónica con Filtro Media Móvil.....	55
4.3 Conclusiones.....	61
4.4 Trabajos a Futuro	61
BIBLIOGRAFIA Y REFERENCIAS	63
ANEXO A.....	66

ÍNDICE DE FIGURAS

Figura 1.1 Línea espectral asociada entre dos niveles de energía	14
Figura 1.2 Aproximación del área bajo la curva	16
Figura 1.3 Etapas del Modelo en Cascada.....	17
Figura 1.4 Modelo en V	18
Figura 1.5 Etapas del Modelo Basado en Prototipos	19
Figura 1.6 Etapas del Modelo en Espiral	20
Figura 1.7 Etapas del Modelo Incremental	20
Figura 1.8 Relación entre SysML y UML.....	21
Figura 1.9 Diagramas soportados por SysML.....	22
Figura 1.10 Ejemplo de Diagrama de Casos de Uso	23
Figura 1.11 Ejemplo de Diagrama de Actividades.....	23
Figura 1.12 Ejemplo de Diagrama de Secuencias.....	24
Figura 1.13 Ejemplo de Diagrama de Requerimientos.....	25
Figura 1.14 Ejemplo de Diagrama de Paquetes	25
Figura 2.1 Diagrama de flujo para el cálculo de temperatura electrónica de una longitud de onda observada	29
Figura 2.2 Diagrama de Requerimientos del Sistema de Estimación de Temperatura Electrónica..	30
Figura 2.3 Diagrama de Paquetes del Sistema de Estimación de Temperatura Electrónica	31
Figura 2.4 Diagrama de Caso de Uso Módulo Traer Tabla de Elemento	32
Figura 2.5 Diagrama de Caso de Uso Módulo de Análisis y Visualización de Espectro.....	33
Figura 2.6 Ejemplo de Cálculo de Área Bajo la Curva.....	33
Figura 2.7 Diagrama de Actividades Traer Tabla de Elemento	35
Figura 2.8 Diagrama de Clases de Tabla de Base de Datos	35
Figura 2.9 Diagrama de Interacción de la Operación del Sistema de Estimación de Temperatura Electrónica.....	37
Figura 3.2 Diagrama de clase para traer tabla de elemento.....	39
Figura 3.3 Ejemplo de presentación de tabla del NIST de Líneas Espectrales de HeI 200-900nm ...	40
Figura 3.4 Interfaz de Usuario del Módulo Traer Tabla de Elemento.....	40
Figura 3.6 Comparación de Espectros.....	42
Figura 3.8 Resultado de acercamiento a sección seleccionada del espectro	44
Figura 3.9 Elección de puntos en control visual.....	44
Figura 3.10 Diagrama de clase para graficar espectros	46

Figura 3.11 Pantalla de regresión lineal.....	46
Figura 3.12 Diagrama de clase para graficar regresiones	47
Figura 4.1 Espectro Continuo de Lámpara de Tungsteno	49
Figura 4.2 Barrido de Espectros desde el centro hacia una orilla	50
Figura 4.3 Pendiente de Boltzmann para Espectro 1	51
Figura 4.4 Pendiente de Boltzmann para Espectro 2	52
Figura 4.5 Pendiente de Boltzmann para Espectro 3	52
Figura 4.6 Pendiente de Boltzmann para Espectro 4	53
Figura 4.7 Pendiente de Boltzmann para Espectro 5	54
Figura 4.8 Pendiente de Boltzmann para Espectro 6	55
Figura 4.9 Pendiente de Boltzmann para Espectro 1 con Filtro	56
Figura 4.10 Pendiente de Boltzmann para Espectro 2 con Filtro.....	56
Figura 4.11 Pendiente de Boltzmann para Espectro 3 con Filtro.....	57
Figura 4.12 Pendiente de Boltzmann para Espectro 4 con Filtro.....	58
Figura 4.13 Pendiente de Boltzmann para Espectro 5 con Filtro.....	59
Figura 4.14 Pendiente de Boltzmann para Espectro 6 con Filtro.....	60
Figura 5.15 Comparación de temperaturas	61

ÍNDICE DE TABLAS

Tabla 2.1 Ejemplo de presentación de tabla de Líneas Espectrales de Ar I 400-700nm.....	31
Tabla 4.1 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1.....	50
Tabla 4.2 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 2.....	51
Tabla 4.3 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 3.....	52
Tabla 4.4 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 4.....	53
Tabla 4.5 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 5.....	54
Tabla 4.6 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 6.....	54
Tabla 4.7 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1 con Filtro	55
Tabla 4.8 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1 con Filtro	56
Tabla 4.9 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1 con Filtro	57
Tabla 4.10 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1 con Filtro	58
Tabla 4.11 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1 con Filtro	58
Tabla 4.12 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1 con Filtro	59
Tabla 4.13 Tabla de Pruebas de Temperaturas Electrónicas de barridos del lado izquierdo	60

Introducción

El análisis de espectros en los últimos años ha sido una herramienta indirecta empleada en múltiples campos [1] para obtener diversos parámetros en algunas aplicaciones como la estimación de la temperatura de cuerpos celestes y la contabilización de elementos y especies. Su uso para encontrar la temperatura de los electrones (temperatura electrónica) es de gran relevancia ya que es un método no intrusivo, por lo que no afecta al fenómeno físico o proceso que se pretenda analizar. Para llevar a cabo dicho proceso, se requiere adquirir un espectro óptico por medio de la separación de la luz emitida por la fuente en sus diferentes longitudes de onda, para lograr lo anterior, se hace uso de prismas que separan la luz [2] y un dispositivo de acoplamiento de cargas [3] como sensor para adquirir los espectros ópticos que serán tratados. Una vez adquiridos los espectros, se emplea entre otros métodos, la ecuación de Boltzmann, ésta requiere datos como el peso estadístico, la probabilidad de ionización, la longitud de onda, la energía del nivel superior, la intensidad y una constante de Boltzmann, al incluir todos éstos en dicha ecuación se logra estimar la temperatura electrónica.

Los procesos de análisis y estimación como el antes expuesto resultan extenuantes para realizarlos de manera manual, por lo que se hace uso de herramientas de cómputo que facilitan el proceso de interpretación, como lo es el caso del Lenguaje Java, el cual permite utilizar como interfaz de usuario una serie de herramientas, como por ejemplo, gráficos, tablas, formularios etiquetas, y librerías que permiten abstraer, diseñar y codificar problemáticas del mundo real.

Por otro lado, en este trabajo se desarrolló una aplicación que permite trasladar datos del NIST (National Institute of Standard System Technology) a una base de datos local, para consultar información específica de especies químicas reportadas. Todo esto con el fin de computar y estimar la temperatura electrónica de forma automática evitando así el cálculo matemático de forma manual.

En el capítulo 1 se muestran los conceptos usados para el desarrollo de este proyecto, esta información es importante, ya que representa la base y sustento teórico del tema de tesis que se presenta.

En el capítulo 2 se encuentra el análisis de requerimientos, las actividades y procesos que se deben llevar a cabo para calcular la temperatura electrónica de una fuente de plasma, haciendo uso de la especificación SysML.

El capítulo 3 describe el desarrollo del sistema explicando cada una de sus partes, haciendo uso del diagramado de clases para facilitar su comprensión.

Por último, en el capítulo 4, se dan a conocer las pruebas realizadas al sistema desarrollado, los resultados obtenidos, las conclusiones y trabajos a futuro.

Del mismo modo, se anexan la bibliografía y referencias que fueron consultadas para desarrollar el informe de tesis.

Estado del Arte

La espectroscopia óptica de emisión y de absorción ha emergido como una técnica útil de diagnóstico usada en diversos ramos de las ciencias como a continuación se menciona en las siguientes aportaciones:

En el artículo “aplicación del procesamiento digital de señales al estudio de espectros ópticos de emisión óptica” [2] se hace uso de la técnica de espectroscopia óptica de emisión –OES (por sus siglas en inglés Optical Emission Spectroscopy) para caracterizar las especies presentes y la densidad de las mismas en un plasma generado con la técnica de magnetron sputtering, empleando para el análisis automático de espectros un software desarrollado en builder C++ y un espectrómetro de alta resolución con rejillas de difracción, valiéndose a su vez de consultas SQL hacia la base de datos de NIST (National Institute of Standard System Technology), *NIST Atomic Spectra Database*, para la identificación de compuestos químicos a través de la longitud de onda de emisión encontrados en los picos máximos y mínimos arrojados por el espectro, en donde la temperatura electrónica y de excitación es calculada eligiendo una o varias líneas aisladas, de donde se pueda identificar totalmente un máximo y dos mínimos con el fin de obtener su intensidad calculando el área bajo la curva sustituyendo este y otros valores en la ecuación de Boltzmann, obteniendo resultados similares a los reportados en la literatura, reduciendo el tiempo de análisis espectral de forma manual.

En el artículo “Caracterización óptica de Diodos emisores de Luz mediante su espectro de emisión y patrones de radiación” [4] se obtuvieron espectros de emisión y modelos de radiación en diodos LED’s de distintos colores con el fin de obtener información relevante para uso posterior de los mismos en sistemas y circuitos optoelectrónicos usando un montaje óptico que permitía descomponer la radiación visible en sus diferentes longitudes de onda, así mismo, dicho ensamble sirvió para analizar los ángulos y los rangos de emisión de los diodos respectivamente,

encontrando que dichos análisis presentan un ligero y aleatorio margen de error debido al proceso de manufactura de los diodos emisores de luz.

Según “Sistema de Absorción Óptica Diferenciada del Espectro (DOAS) para evaluar la calidad del aire” [5]. Se usa un sistema basado en absorción de luz ultravioleta de especies químicas por la teoría de Beer-Lambert para evaluar la calidad del aire, donde las pruebas consideradas en este trabajo se compararon con las respuestas de sistemas de uso convencional aplicadas en tres entidades federativas de la república mexicana donde la réplica del equipo DOAS y el equipo convencional arroja datos similares, por lo que el sistema DOAS tiene un alto grado de confiabilidad.

En “Two-temperature modelling and optical emission spectroscopy of a constant current plasma arc welding process” [6]. Se usa la técnica de espectroscopia de emisión óptica (OES) para determinar cantidades físicas tales como la densidad de electrones y la temperatura de las partículas en plasmas térmicos manipulando la ley de Boltzmann, en donde se calcula la temperatura de excitación a través de la emisividad de dos o más líneas emitidas por la misma especie, sin embargo, las incertidumbres en la determinación de la temperatura dependerán en gran medida de la diferencia de los niveles de energía relacionados a las líneas seleccionadas, obteniendo como resultado que los métodos basados en la medición absoluta del coeficiente de la emisión volumétrica pueden ser más precisos, sin en cambio, el análisis por Boltzmann es relativamente fácil de usar y el análisis de datos es casi directo.

En “Aplicación De La Espectroscopía Óptica De Emisión En La Caracterización De Plasmas De Descargas Eléctricas En Tubos De Lámparas HPS Y Plasmas Láser En Aire” [7] se analizan espectros de radiación emitida por plasmas de un tubo de lámpara de sodio de alta presión (HPS) generados por descargas eléctricas en operación normal, así como espectros de radiación emitidos por plasmas en mezcla de argón-aire generado por un láser pulsado, donde se pudo estimar la temperatura

y la densidad electrónica usando un espectrógrafo óptico, un detector de arreglo lineal CCD para la obtención de espectros y la relación de dos líneas aisladas aplicando el efecto de ensanchamiento Stark para la obtención de los parámetros antes mencionados, donde la respuesta de la lámpara HPS mostró una gran dinámica e información espectral desde su encendido hasta su estabilización, por otro lado, en los espectros de plasma-laser se observaron y midieron líneas pertenecientes a diversas cantidades de especies reportadas en el NIST estimando en ambos la temperatura y densidad electrónica en ambos tipos de plasma.

Planteamiento del Problema

Una de las problemáticas existentes encontrada en la revisión sobre el estado del arte es que no se cuenta con un programa que realice cálculos sobre el análisis de temperatura electrónica de forma automática, donde el usuario tiene que hacer en primera instancia el análisis del espectro óptico sobre las especies presentes en sus diferentes longitudes de onda empleando datos provenientes de la base de datos del NIST buscadas por el usuario sobre una página web recorriendo registros de forma visual hasta obtener específicamente referencias de interés. Como siguiente paso, el usuario realiza cálculos matemáticos de forma manual, como por ejemplo: Para obtener la temperatura electrónica, la densidad y presencia de especies y energías o probabilidades de ionización.

Tomando en cuenta todo lo anterior, se pretende resolver dicho problema a través de un programa de cómputo que facilite el análisis y la obtención de resultados posteriores de forma automática, en específico sobre la estimación de la temperatura electrónica.

Objetivos

General

Desarrollar un programa de cómputo que permita estimar la temperatura electrónica en una fuente de emisión continua.

Específicos

- Estimar el área bajo la curva de la línea deseada.
- Generar una base de datos local, basada en la información de especies proporcionada por el NIST.
- Estimar la temperatura electrónica de la línea deseada.
- Mostrar la interpolación de la temperatura de las líneas seleccionadas.

Hipótesis

Si se diseña y desarrolla un sistema codificado en lenguaje Java que permita analizar un espectro óptico se podrá estimar la temperatura electrónica de una fuente de emisión de plasma.

Alcances

- Desarrollar e implementar un sistema que permita graficar y analizar espectros ópticos.
- Desplegar información en un sistema de información visual que permita consultar datos específicos sobre especies químicas.

- Mostrar cálculos como áreas bajo la curva y temperaturas electrónicas de líneas elegidas durante el proceso de cálculo de temperatura electrónica local.
- La aplicación es multiplataforma, por lo que puede ser instalada y ejecutada también en computadoras personales que tengan sistemas operativos distintos a Windows.

Limitaciones

- La graficación de espectros no se realiza en tiempo real, se requiere almacenarlos para su análisis posterior.
- La elección correcta de líneas de interés depende del conocimiento del usuario del sistema.

CAPITULO 1. MARCO TEÓRICO

1.1 Plasma

El plasma es el cuarto estado de la materia, se encuentra en el 99.9% de la materia que existe en el universo. Es un gas parcial o totalmente ionizado constituido por una mezcla de electrones, iones, átomos, fotones, radicales libres y metaestables, donde las cargas eléctricas negativas como positivas se encuentran casi en equilibrio [8], de manera que se considera como un estado neutro. Los rayos, las auroras boreales, y el sol son manifestaciones naturales de un plasma.

Existen diversos métodos para crear plasma entre los más significativos, se encuentran, la inducción por efecto magnético, inducción láser, y por alto voltaje (inducción de campo eléctrico de alta intensidad). En el último, se emplean diversos circuitos capaces de elevar el voltaje hasta el nivel necesario para romper la condición aislante del gas y convertirlo en un conductor. Cuando la temperatura es similar entre los electrones, iones, y especies neutras se constituye un plasma térmico que puede alcanzar temperaturas de hasta 25000° K, si la temperatura de los electrones es mayor que la de las partículas pesadas, se considera un plasma frío, por lo que la temperatura electrónica oscila en un promedio de 4000° K, mientras que las partículas pesadas, presentan una temperatura de 300° K [9].

En base al modo en que se transmite la energía, los plasmas se dividen en plasmas transferidos y no transferidos, en los primeros, el arco se forma de cátodo a ánodo, de esta forma el gas es impulsado al exterior y la pieza a tratar se somete a la antorcha, por otro lado, en los no transferidos, la pieza funciona como un electrodo (generalmente el ánodo), por lo que la energía se transfiere totalmente a la pieza [10].

Los usos que se le dan al plasma son para corte de materiales, recubrimiento de superficies, esterilización de equipo médico, tratamiento de contaminantes, entre otros.

1.2 Espectroscopia

La espectroscopia es una de las herramientas de diagnóstico usadas en la astrofísica y la física del plasma [11], ofreciendo la posibilidad de observar líneas de emisión ópticas en tiempo real, este método se considera como no invasivo, por lo que el fenómeno de estudio no se ve contaminado por la medición.

En los compuestos o elementos, las moléculas tienen un conjunto de niveles de energía y las líneas que se observan en los espectros se deben a transiciones de estos niveles de energía como puede observarse en la Figura 1.1. Estas líneas son originadas por un fotón de luz cuya frecuencia ν , está relacionada con la brecha de energía ΔE entre dos niveles de energía de acuerdo con la ec. 1.1 [12]:

$$\Delta E = h\nu \quad (1.1)$$

Dónde:

h = Constante Universal de Planck

Por lo que si se consideran E_1 como la energía del nivel 1 y E_2 la energía del nivel 2, se obtiene la ec. 1.2.

$$\Delta E = E_2 - E_1 \quad (1.2)$$

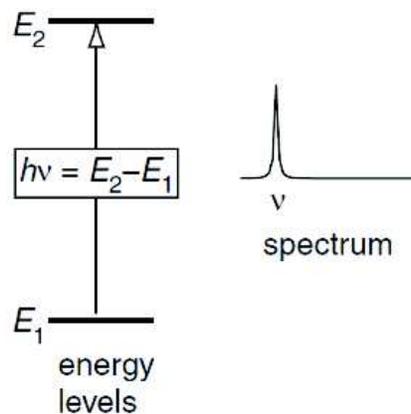


Figura 1.1 Línea espectral asociada entre dos niveles de energía

Para observar las líneas emitidas, se requiere de equipo especial que permita descomponer la luz emitida en sus componentes para después ser convertidas en señales eléctricas para su análisis posterior, en específico de la temperatura electrónica.

El equipo de espectroscopia, está formado generalmente por una lente convergente que proyecta la fuente de emisión del plasma por una fibra óptica posicionada por un sistema que permita fijarla en un lugar en específico. La fibra lleva la luz hasta un monocromador que por medio de prismas se encarga de separar la luz en sus diversas longitudes de onda, mismas que impactan en dispositivo acoplador de cargas (Charged Couple Device CCD) que convierte las señales luminosas en señales eléctricas, posteriormente con una tarjeta de adquisición de datos se convierte en una señal digital que es procesada en forma de líneas en la pantalla de una computadora.

1.3 Ecuación de Boltzmann para la Estimación de Temperatura Electrónica

Uno de los métodos para la estimación de temperatura electrónica basado en el equilibrio termodinámico local (LTE) es a través de la ec.1.3 [13].

$$\ln\left(\frac{\epsilon\lambda}{Au\ gu}\right) = -\frac{1}{kT}Eu + cte. \quad (1.3)$$

Dónde:

ϵ = Área bajo la curva

λ = Longitud de onda observada

Au = Probabilidad de transición

gu = Peso estadístico del electrón

Eu = Energía del nivel superior

k = Constante de Boltzmann (8.6173724×10^{-5} eV)

$T = \text{Temperatura en grados Kelvin}$

Los parámetros expuestos por la ec. 1.3 se obtienen de la base de datos Atomic Spectra Database del NIST, misma que tiene acceso público y que puede ser consultada a través de internet [14].

Conforme a la ecuación 1.3 es necesario proporcionar el área bajo la curva de la longitud de onda observada, misma que es calculada mediante la suma de varias áreas (Figura 1.14).

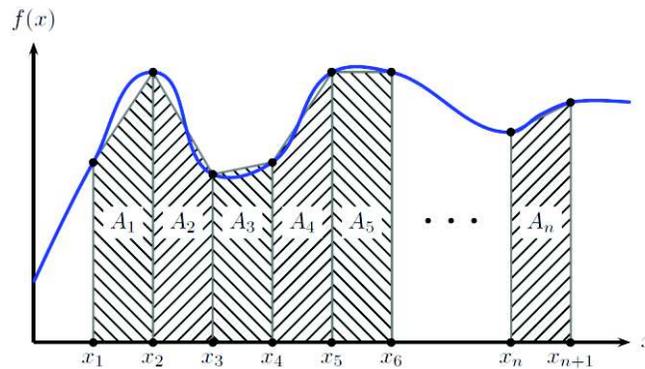


Figura 1.2 Aproximación del área bajo la curva

Para lograr lo anterior, se hace uso de la integración numérica por medio de trapecios [15,16] en donde el área del i ésimo cuerpo geométrico es:

$$A_i = \frac{f_i + f_{i+1}}{2} (x_{i+1} - x_i) \quad (1.4)$$

Asumiendo que el espaciamiento de los datos es diferente e igual a Δ tenemos que:

$$A_i = \Delta \frac{f_i + f_{i+1}}{2} \quad (1.5)$$

Por lo tanto, el área total es:

$$A = \sum_{i=1}^n A_i = \frac{\Delta}{2} \sum_{i=1}^n (f_i + f_{i+1}) \quad (1.6)$$

1.4 Metodologías para el desarrollo de software

Una metodología para el desarrollo de sistemas sugiere los procedimientos que se deben llevar a cabo para producir software [17]. En éstas, se analizan, se diseñan y se documentan los sistemas. A continuación se describen algunas de estas técnicas.

1.4.1 Modelo en cascada

En este modelo los procesos que se llevan a cabo sugieren un orden secuencial, para el desarrollo de un sistema. En la figura 1.3 se muestran las diferentes fases del modelo en cascada, en donde se comienza con el análisis de requerimientos, al terminar cada fase se continúa con la siguiente de manera sucesiva hasta llegar a la de mantenimiento. Este modelo presenta una desventaja ya que no se especifica como modificar alguna de sus etapas, del mismo modo, los resultados no pueden verse a corto plazo, ya que todas las fases deben ser completadas y con ello los errores de cada una de estas se acarrean en las siguientes [18].



Figura 1.3 Etapas del Modelo en Cascada

1.4.2 Modelo en V

El modelo en V (Figura 1.4) surge como variante al modelo en cascada, puede observarse como se relaciona la calidad, la comunicación y la codificación. El proceso inicia con el análisis de requerimientos siguiendo a detalle las especificaciones del mismo para llegar a la fase de codificación, en donde se generan pruebas unitarias y en conjunto para asegurar y validar la calidad desde un principio [19].

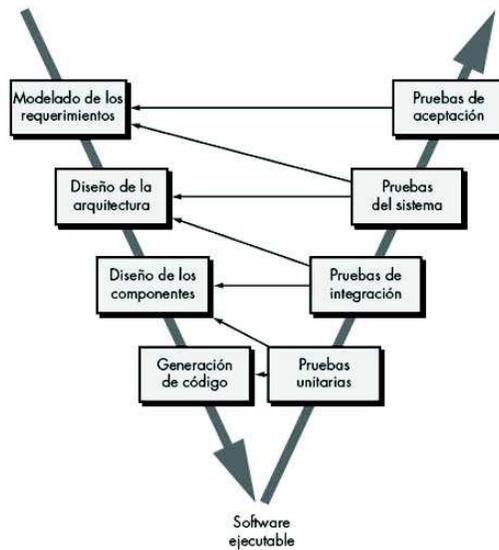


Figura 1.4 Modelo en V

1.4.3 Modelo de construcción de Prototipos

En este modelo el proceso inicia con la comunicación que tiene el cliente con el desarrollador (Figura 1.5), en donde se identifican los requisitos y objetivos globales proporcionando una vista general del sistema, con ello se permite que todo el sistema o algunas de sus partes se construyan rápidamente para comprender, aclarar o recabar nuevos requisitos [20], de esta forma el usuario y el programador revisan la calidad del software, lo que permite una mayor facilidad de mantenimiento en cada prototipo parcial o total. La construcción por prototipos se resume en las siguientes actividades [21]:

- Identificación de requerimientos: Se definen las necesidades básicas de información del usuario: Entradas y salidas del sistema
- Desarrollo de un prototipo: El diseñador crea un prototipo funcional.
- Uso del prototipo: El usuario prueba el sistema para determinar si cumple con sus necesidades, haciendo sugerencias u observaciones sobre el manejo del mismo.
- Mejora del prototipo: El desarrollador toma nota de todas las observaciones o mejoras que solicita el usuario, depura el prototipo y la penúltima y última etapa se repiten hasta que el usuario quede satisfecho.



Figura 1.5 Etapas del Modelo Basado en Prototipos

1.4.4 Modelo en Espiral

Este tipo de modelo representa el proceso de desarrollo de software como una sucesión de construcción de prototipos. En la figura 1.6 se puede observar que este modelo es representado por una espiral, en la cual, se describe cada una de sus etapas. Cada fase comienza con la fijación de objetivos, alternativas y restricciones, (primer cuadrante) se evalúan los riesgos antes de continuar con la siguiente (segundo cuadrante) y se concluye con la revisión (tercer cuadrante), en donde se discuten los logros y planes para realizar las siguientes etapas (cuarto cuadrante) [19].

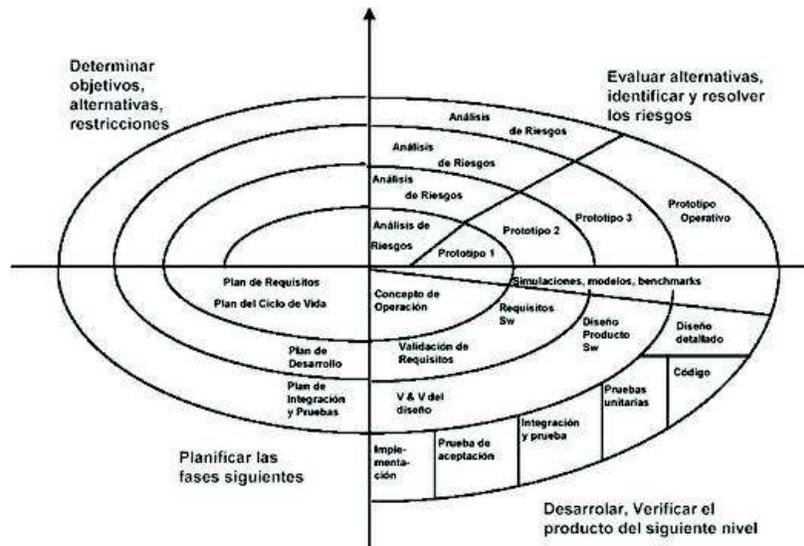


Figura 1.6 Etapas del Modelo en Espiral

1.4.5 Modelo Incremental

Éste, combina los elementos del modelo en cascada con los aspectos de la construcción de prototipos [19]. En la Figura 1.7 se muestran sus diversas etapas, en cada incremento se realiza el análisis, diseño, codificación y pruebas, para entregar nuevas mejoras del sistema.

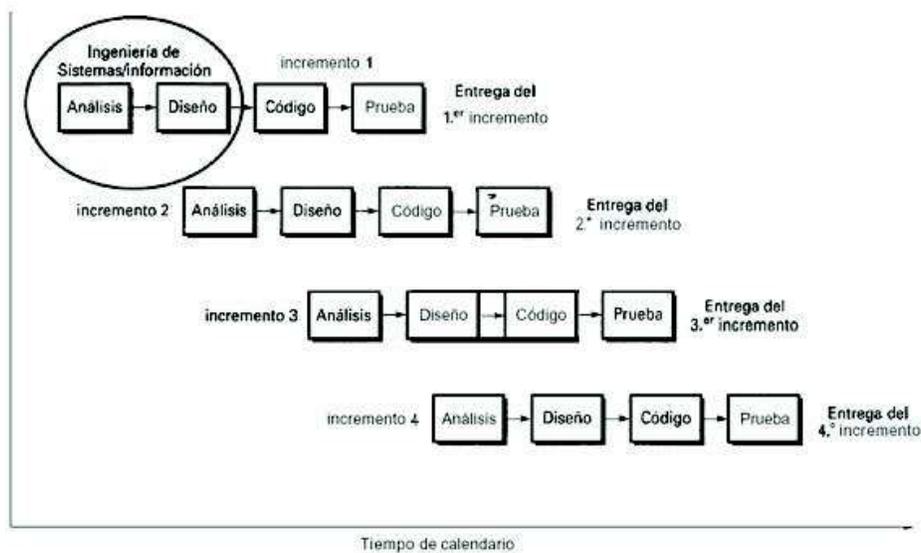


Figura 1.7 Etapas del Modelo Incremental

Además del uso de metodologías para el desarrollo de software también suelen explotarse herramientas que facilitan la generación de modelos que permiten dar a entender a los clientes mediante un panorama no técnico el sistema que se está desarrollando y a los desarrolladores del sistema una extensión técnica para el desarrollo del mismo.

1.5 Lenguaje de modelado de sistemas SysML

SysML (por sus siglas en inglés Systems Modeling Language) es un estándar publicado en septiembre de 2007 por OMG (Object Management Group) que soporta la especificación, análisis, diseño, verificación y validación de sistemas complejos. Dichos sistemas pueden incluir hardware, software, información personal, procesos, entre otros, en donde la práctica común de esta corriente se usa en lenguajes de modelado, herramientas y técnicas sobre proyectos de sistemas [22].

El SysML utiliza algunas extensiones de UML y provee extensiones adicionales (Figura 1.8). Esto puede constituirse como una mejora a la comunicación entre los diferentes participantes en el proceso de desarrollo de los sistemas y promueve la interoperabilidad entre herramientas de modelado. Todo esto teoriza que con este lenguaje de modelado pueden fabricarse modelos en diferentes áreas, tales como las automotrices, las aeronáuticas, de comunicación y de sistemas de información.

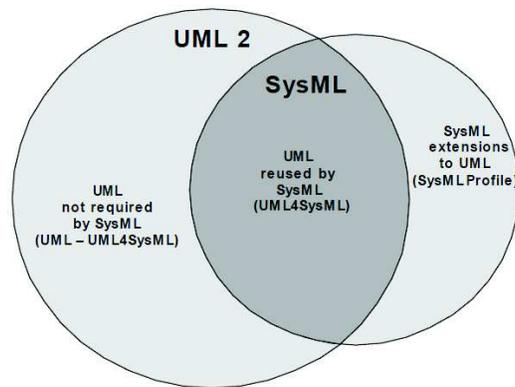


Figura 1.8 Relación entre SysML y UML

En la Figura 1.9 se muestran los diagramas que conforman el estándar SysML, cada etapa se describen en los apartados 1.4.1 al 1.4.

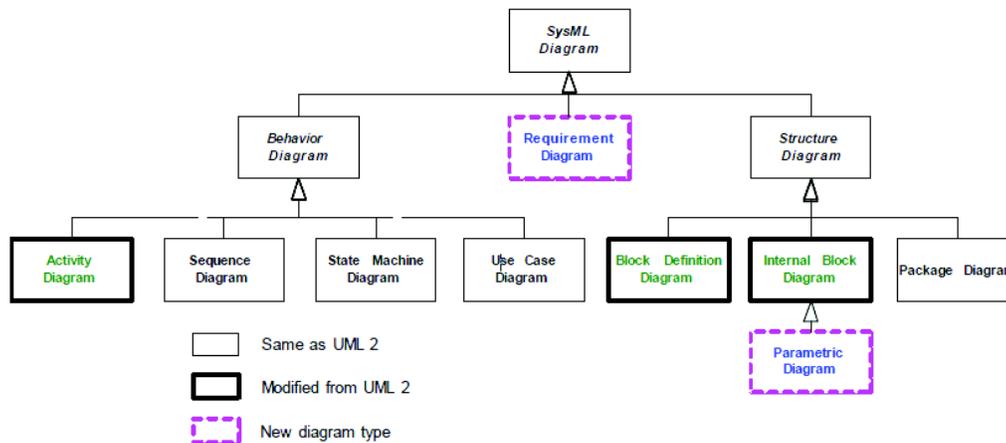


Figura 1.9 Diagramas soportados por SysML

1.5.1 Casos de Uso

El diagrama de casos de uso es un método usado para describir los diferentes escenarios del sistema, en los cuales se observan las funcionalidades o capacidades realizadas por la interacción entre el sistema y sus actores. Este tipo de diagramas tiene como finalidad describir la manera en que se usará el sistema y captar de forma visual el propósito del mismo, se deben incluir los casos de uso, los actores y las comunicaciones asociadas entre ellos [23]. En la figura 1.10 se muestra un ejemplo del diagrama de casos de uso, se puede observar que se encuentra limitado por un rectángulo (pertenece o se encuentra dentro del sistema), posteriormente se halla a un usuario del sistema, mejor conocido como actor y enseguida se encuentran diversas líneas (conectores) que comunican al actor con los casos de uso (ovalos).

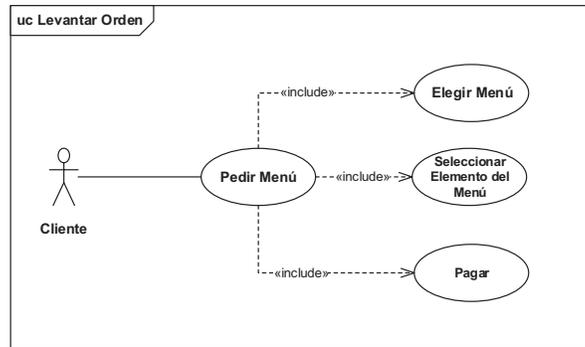


Figura 1.10 Ejemplo de Diagrama de Casos de Uso

1.5.2 Diagrama de actividades

El modelado de actividad acentúa las entradas, las secuencias y las condiciones necesarias para coordinar otros comportamientos, mostrando la forma en que se conduce un sistema entre las diversas actividades que se componen de cada una de sus partes [24]. Este tipo de diagrama es parecido a un diagrama de flujo y tiene por objetivo mostrar una visión simplificada de lo que ocurre en una transacción o proceso. En la figura 1.11, se aprecia que se cuenta con un punto inicial de actividad (denotado por un círculo relleno), así como un punto final (representado por una diana), las actividades se representan con un rectángulo con esquinas redondeadas, las transacciones (flujo de control) son representadas por flechas, en el flujo de control pueden existir operaciones paralelas representadas por una línea horizontal seguida de flechas.

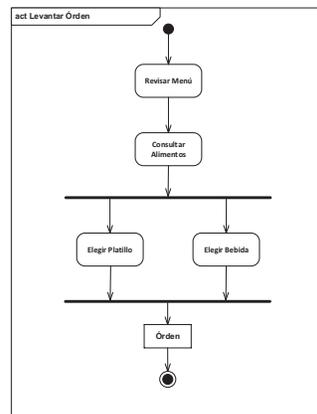


Figura 1.11 Ejemplo de Diagrama de Actividades

1.5.3 Diagrama de Interacción

En este tipo de diagramas los objetos que conforman el sistema interactúan entre sí, la interacción puede darse en base a objetos individuales o bien, en una colección de objetos cooperantes. Este diagrama proporciona una vista del comportamiento de un conjunto de objetos que transmiten mensajes entre sí durante el proceso de una transacción [25]. En la especificación SysML para el diagrama de interacción se usa el diagrama de secuencia [22].

1.5.4 Diagrama de Secuencia

El diagrama de secuencia describe el flujo de control entre las diversas partes/bloques de un sistema. Este diagrama representa el envío y la recepción de mensajes entre las entidades que obran recíprocamente teniendo una línea de vida, en donde el tiempo es simbolizado a lo largo del eje vertical del mismo. En la Figura 1.12 puede verse que este tipo de modelado consta de entidades que se representan mediante rectángulos con el nombre de una acción, mensajes que son representados por líneas de punta de flecha que van de un objeto a otro o en ocasiones hacia el mismo objeto y una línea de vida única para cada objeto.

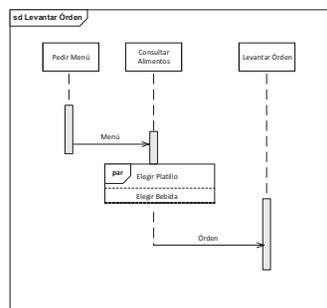


Figura 1.12 Ejemplo de Diagrama de Secuencias

1.5.5 Diagrama de requerimientos

Un requisito especifica una capacidad o condición que debe ser satisfecha. El diagrama de requerimientos especifica la realización de una función en el sistema

o una condición del funcionamiento a satisfacer de un sistema. En la figura 1.13 puede verse que los requerimientos se simbolizan mediante el uso de tablas de dos filas por una columna, en la primera fila se especifican sus nombres y en la segunda fila se especifica una descripción del mismo, las líneas punteadas con punta de flecha indican que un requerimiento es necesario para cubrir en su totalidad otro requerimiento, o en su defecto, que a partir de uno ya existente se genere uno nuevo.

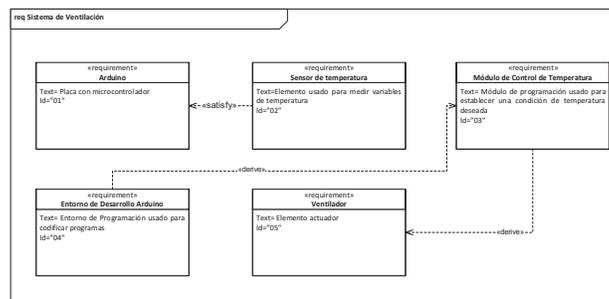


Figura 1.13 Ejemplo de Diagrama de Requerimientos

1.5.6 Diagrama de paquetes

En un diagrama de paquetes, se muestra la descomposición física y lógica de un sistema, indicando la relación que existe entre sus partes. En la figura 1.14 puede observarse que un paquete está definido por una imagen similar a la una carpeta, dentro de éstos existen más paquetes.

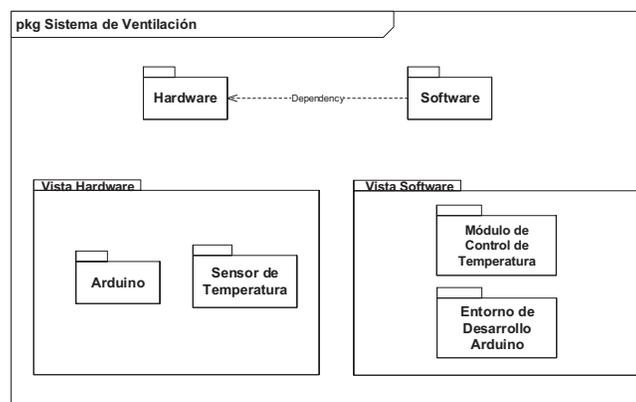


Figura 1.14 Ejemplo de Diagrama de Paquetes

1.6 Programación Orientada a Objetos (POO)

Una debilidad de la programación estructurada es la separación conceptual de datos y código, esto se complica a medida que el programa crece [26].

La programación orientada a objetos, se basa en construir programas a través de entidades más simples llamadas objetos, instancias o clases, lo que permite heredar datos y código de aplicaciones existentes, mejorando la productividad, facilitando la extensión y la reutilización de clases en otras aplicaciones.

1.7 Java

Java, es un lenguaje de programación orientado a objetos presenta ciertas características mencionadas a continuación [27]:

- Simple: Su diseño es muy similar a los lenguajes de programación C y C++ con la intención de facilitar su aprendizaje.
- Orientado a Objetos: Soporta las tres características de la POO: Encapsulación, herencia y polimorfismo.
- Compilado e Interpretado: El compilador Java genera un ByteCode (Código intermedio entre el lenguaje máquina del procesador y Java).
- Portable/Multiplataforma: Sus interfaces pueden ser implementadas en entornos Unix, PC y Mac, gracias al mecanismo que permite ejecutar el ByteCode. Este mecanismo es la denominada máquina Virtual de Java (JVM por sus siglas en inglés).
- Multihilo: Permite realizar actividades simultáneas en un solo programa.
- Dinámico: El manejo de librerías no paraliza la ejecución de aplicaciones en versiones actuales siempre que se mantenga la API anterior.

1.8 Librería JFreeChart

El proyecto JFreechart fundado en Febrero de 2000 por David Gilbert es una librería Java 100% gratuita gestionada y modificada por las contribuciones de una diversa

comunidad de desarrolladores. Es utilizada como apoyo en la creación de diversos tipos de gráficos, sus características incluyen [28]:

- Diseño flexible que es fácil de extender en aplicaciones por parte del servidor y del cliente.
- Soporta diversos tipos de salida, incluyendo archivos de imagen (PNG y JPEG) y formatos de archivo gráficos vectoriales (PDF, EPS y SVG).
- Es de código abierto. Se distribuye bajo los términos de la Licencia Publica General Reducida de GNU (LGPL por sus siglas en inglés), que permite el uso de aplicaciones propietarias.

1.9 Librería JExcelApi (JXL)

JXL es un API de Java de código abierto que permite a los desarrolladores leer, escribir y actualizar dinámicamente hojas de cálculo de Excel. Permite escribir los cambios en cualquier flujo de salida (disco, servidor, o cualquier socket). Cualquier sistema operativo que pueda ejecutar una Máquina Virtual Java puede hacer uso de esta librería. Algunas de sus características son las siguientes [29]:

- Lee datos desde libros de Excel 95,97,2003
- Lee y escribe formulas en Excel 97 y posteriores
- Genera hojas de cálculo en Formato Excel 97-2003
- Soporta formato de fuente, de número y de fecha
- Soporta sombreados, bordes y color de celdas
- Modifica hojas de cálculo
- Es internacionalizada, permite el procesamiento en casi cualquier localidad, país, idioma, o codificación de caracteres
- Soporta la inserción y copia de imágenes en las hojas de calculo

1.10 Filtro media móvil

Esta técnica usa información pasada, en la que se construye una nueva serie a partir de la media de un número determinado de datos, en ella se va añadiendo de forma sucesiva un nuevo dato y quitando otro al mismo tiempo (el más antiguo de los datos incluidos en la media anterior) [30].

La expresión que define a una media móvil de orden s consiste en calcular una serie para cada momento t computando el siguiente valor:

$$M_t = \frac{Y_t + Y_{t-1} + \dots + Y_{t-s+1}}{s} \quad (1.7)$$

CAPITULO 2. MODELADO DEL SISTEMA

MODELADO SYSML

El presente capítulo, muestra el modelado del sistema a desarrollar, mediante el uso de diagramas especificados en el estándar SysML, el proceso que detalla las necesidades del sistema de estimación de temperatura electrónica se muestra en el siguiente diagrama de flujo (Figura 2.1).

- Se requiere graficar un espectro de naturaleza óptica en un control gráfico
- El control gráfico debe de ser capaz de seleccionar en él mismo tres puntos: Longitud de onda, y dos puntos base para aproximar el área bajo la curva (Revisar Anexo A), parámetro requerido por la ecuación 1.3.
- Se necesita diseñar una base de datos local con la finalidad de realizar consultas sobre elementos ionizados como el gas a utilizar para generar la descarga, todos los parámetros como: **Longitud de onda observada**, **Probabilidad de Transición**, **Peso Estadístico del Electrón y Energía del Nivel Superior** se obtienen de la base de datos Spectra Atomic Database del NIST [14], todos éstos datos son necesarios para ser sustituidos en la ecuación de Boltzmann (ec. 1.3) con la finalidad de mostrar al usuario el cálculo de temperatura electrónica.

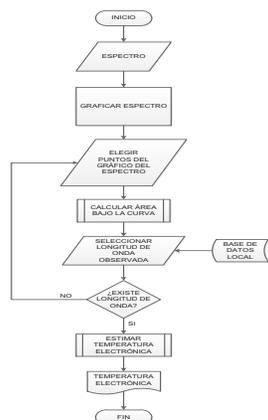


Figura 2.1 Diagrama de flujo para el cálculo de temperatura electrónica de una longitud de onda observada

2.1 Diagrama de Requerimientos

La Figura 2.2 muestra los requerimientos indispensables para el desarrollo del Sistema de Estimación de Temperatura Electrónica. Puede notarse que la **Fibra Óptica** es un elemento indispensable para satisfacer por completo la adquisición realizada por el **Monocromador (USB 2000+)**, el **Módulo de Análisis y Visualización del Espectro** se apoyará de la **Librería JXL** usada para leer el espectro almacenado en un archivo Excel 97-2003, la **Librería JFreeChart** ayuda a visualizar el espectro leído sobre un componente gráfico. El gestor **MySQL** es explotado para generar una base de datos que almacene y consulte información como la de la Tabla 2.1.

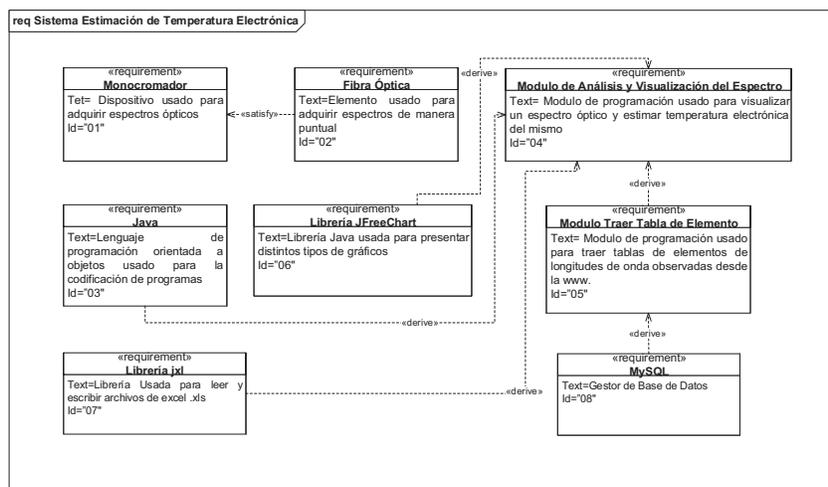


Figura 2.2 Diagrama de Requerimientos del Sistema de Estimación de Temperatura Electrónica

2.2 DIAGRAMA DE PAQUETES

En la Figura 2.3 se muestra la organización completa del sistema en sus diversas partes y/o elementos. Puede verse que la parte que corresponde al **Software** depende del **Hardware**, ya que es la encargada de adquirir espectros de naturaleza óptica, **El Módulo de Análisis y Visualización del Espectro** así como el **Módulo de Traer Tabla de Elemento** son codificados en lenguaje **Java**, ambos se comunican con una base de datos local almacenada en **MySQL**, el primer módulo

hará consultas de los parámetros requeridos para el cálculo de la temperatura electrónica y el segundo se diseña de manera que pueda traer y recopilar datos del Atomic Spectra Database del NIST [14].

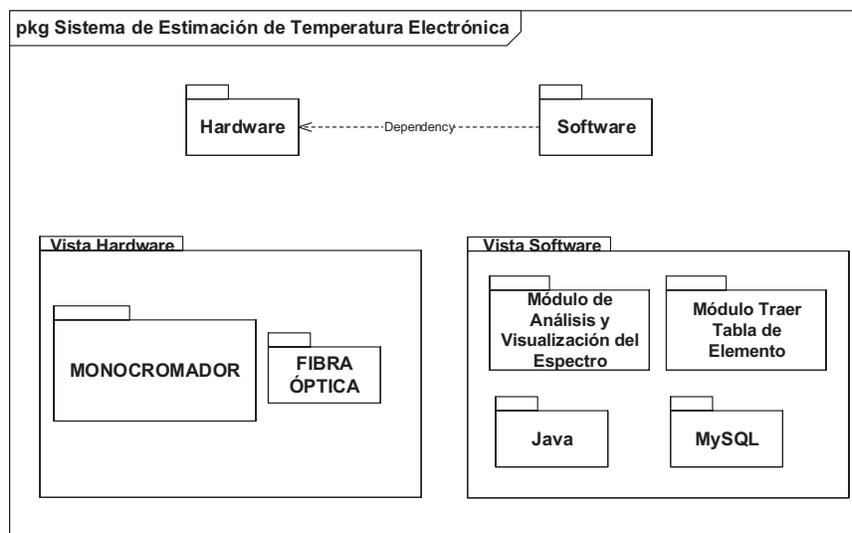


Figura 2.3 Diagrama de Paquetes del Sistema de Estimación de Temperatura Electrónica

2.3 Diagrama de Casos de Uso

Los parámetros expuestos por la ecuación 1.1 pueden obtenerse de varias tablas almacenadas en la base de datos Atomic Spectra Database del NIST misma que es de acceso público y que puede ser consultada por internet [14]. La tabla 2.1 muestra un ejemplo de líneas espectrales observadas de Ar I en un rango de 400nm-700nm.

Tabla 2.1 Ejemplo de presentación de tabla de Líneas Espectrales de Ar I 400-700nm

λ (nm)	Intensidad Relativa	Prob. Trans. (Au)	Energía (Eu)	Peso Estadist (gu)
404.4418	50	3.33e+05	14.6882	5
415.8590	400	1.40e+06	14.5289	5
416.4180	50	2.88e+05	14.5249	3
.
.
.
695.1478	7	2.2e+05	14.9548	5
696.0250	7	2.4e+05	14.9526	5
696.5431	10000	6.39e+06	13.3278	3

El **Caso de Uso Módulo Traer Tabla de Elemento** (Figura 2.4) ilustra la manera en la cual el usuario hace la consulta de una tabla de la base de datos del NIST para conseguir almacenarla en una base de datos local, misma que es utilizada para realizar el análisis del espectro óptico. Para conseguir lo anterior, se requiere acceder mediante una conexión a internet a una tabla de interés en la base de datos Spectra Atomic Database [14], ésta es almacenada en un Disco Duro o Memoria USB como página web en formato HTML misma que se importa a una tabla local generada de forma automática al leer las etiquetas HTML de dicha página. Dicho proceso será explicado a más detalle en la sección 2.4 de este capítulo.

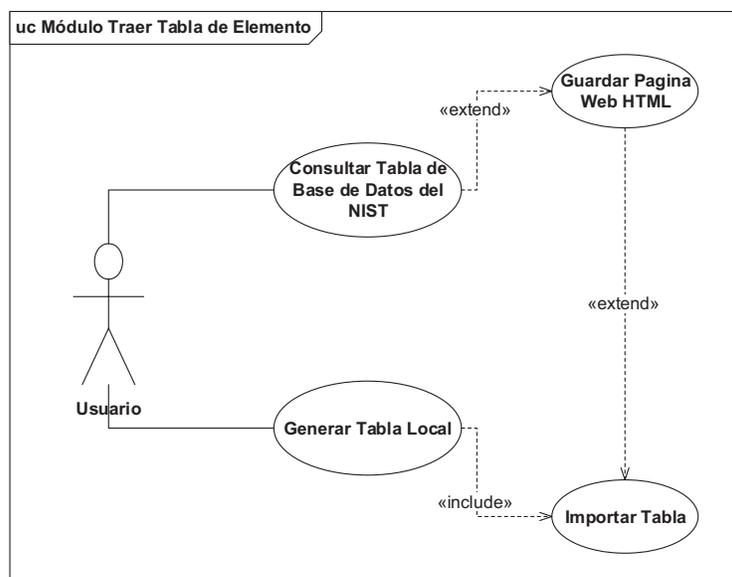


Figura 2.4 Diagrama de Caso de Uso Módulo Traer Tabla de Elemento

Como se puede observar en el **Caso de Uso Módulo de Análisis y Visualización de Espectro** (Figura 2.5) el usuario tiene que elegir tres puntos en el gráfico donde se visualiza el espectro: La longitud de onda, y dos puntos base para realizar el correspondiente cálculo del área bajo la curva (Figura 2.6 y explicada en el apartado 1.3), éstos son requeridos por la ecuación de Boltzmann (ec. 1.3), con este dato se procede a realizar la estimación de la temperatura electrónica sobre una longitud de onda observada, este paso se repite como mínimo dos veces para cumplir con la condición de interpolación lineal y con ella obtener la temperatura electrónica.

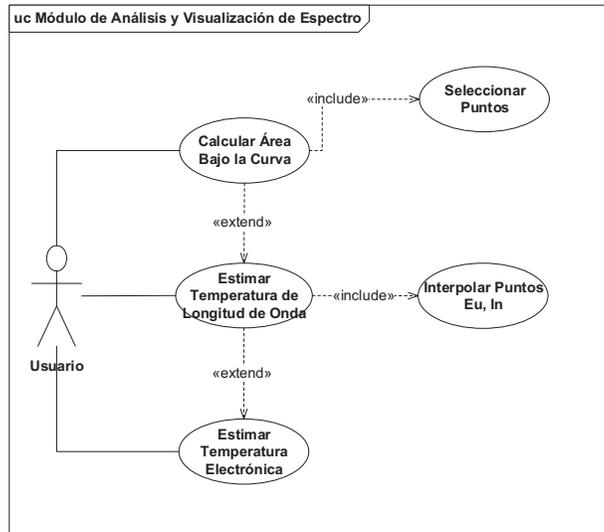


Figura 2.5 Diagrama de Caso de Uso Módulo de Análisis y Visualización de Espectro

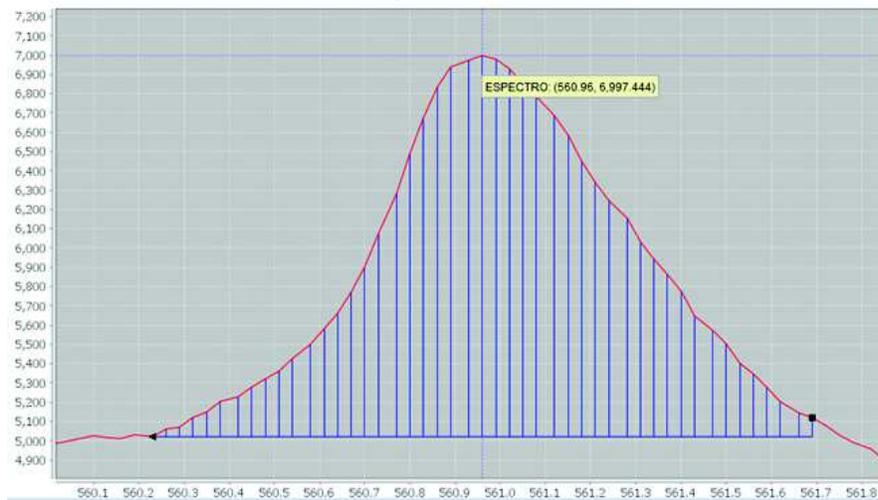


Figura 2.6 Ejemplo de Cálculo de Área Bajo la Curva

2.4 Diagrama de Actividades

La figura 2.7 muestra las entradas y salidas de las actividades que dan seguimiento al comportamiento del sistema para traer una tabla del NIST a una tabla de una base de datos local. Cabe destacar, que consultar las tablas de diversas especies contenidas en el NIST de forma manual es un proceso que conlleva a la transcripción a papel de diversos datos, ya que es necesario identificar en ellas las

longitudes de onda que se desean estudiar y con ello, traer todos los parámetros requeridos por la ecuación de Boltzmann, por lo que el algoritmo descrito por el siguiente diagrama de actividades (Figura 2.7), es el encargado de traducir el lenguaje de marcado de hipertexto (HTML) para importar las mismas a una base de datos local diseñada en MySQL. El proceso a seguir es el siguiente:

- El usuario accede al NIST para consultar mediante una conexión de internet una tabla alojada en la base de datos Atomic Spectra Database.
- El usuario guarda la consulta anterior como una página web en formato HTML.
- El usuario carga en el programa la página anteriormente almacenada, dando nombre a la tabla que se ha de generar de forma local.
- El programa hace la lectura de las etiquetas de todo el archivo HTML quedándose con los datos relevantes al formato de tabla que presenta la página web del NIST, descartando aquellas etiquetas que no pertenecen a la tabla, es decir, se extraen los datos que corresponden a los parámetros de la ecuación de Boltzmann. Cabe mencionar que los datos deben de ajustarse al formato de los campos requeridos por la tabla local, por lo que el **Módulo de Traer Tabla de Elemento** también realiza el ajuste y validación de los mismos.
- Finalmente se resguardan dichos parámetros en una tabla que se genera de forma automática en una base de datos local diseñada en MySQL.

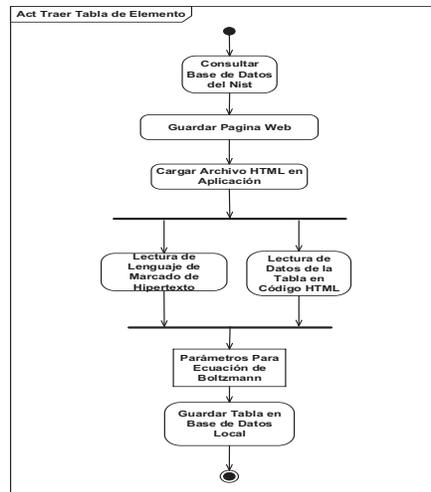


Figura 2.7 Diagrama de Actividades Traer Tabla de Elemento

El diseño de la base de datos en MySQL por el carácter consultivo de los parámetros de la ecuación de Boltzmann no requiere ningún tipo de relación, por lo que la base de datos no es relacional, cada una de sus tablas contiene los mismos campos teniendo por nombre: El elemento químico, su estado de ionización y la longitud de onda observada, por ejemplo para guardar los datos de la tabla 2.1, la tabla local deberá llamarse Arl_400_700nm. La Figura 2.8 muestra el diseño de las tablas contenidas en la base de datos ELEMENTOS.sql nombrada así por gusto propio.

Nombre_de_Tabla
id: integer
Wavelength: double
Intensity: double
Au: double
Eu: double
Gu: double

Figura 2.8 Diagrama de Clases de Tabla de Base de Datos

2.5 Diagrama de Interacción

La Figura 2.9 muestra el envío y recepción de mensajes entre las diversas entidades que trabajan de manera recíproca para conjuntar resultados. El proceso que se sigue se relata a continuación:

1. El usuario realiza la adquisición del Espectro: Dicho espectro puede ser un espectro de una fuente de emisión como lo es un plasma o un espectro continuo.
2. El usuario detiene la adquisición, por lo tanto deberá almacenar el tipo de espectro adquirido.
3. Al realizar la carga del espectro, paralelamente se está graficando el espectro de emisión y el espectro continuo, ya que del espectro de emisión se despreja su continuo, por lo que el usuario tendrá que buscar en una unidad de almacenamiento los dos espectros para proceder a visualizar el resultado de dicha operación.
4. Al resultado del punto anterior se le aplica un filtro de tipo media móvil, con la intención de suavizar la curva. La forma en la cual se aplica dicho filtro será explicada en el siguiente capítulo.
5. Una vez que el espectro se encuentra listo para su análisis el usuario tendrá que elegir tres puntos: Un punto máximo (longitud de onda observada), y dos puntos como base, con la intención de que el programa calcule el área bajo la curva de la longitud de onda observada.
6. El sistema hará la consulta de los parámetros de la ecuación de Boltzmann de la longitud de onda observada en la base de datos local ELEMENTOS.sql, para que junto con el área bajo la curva, se logre estimar la temperatura electrónica de esa longitud de onda.
7. Para lograr interpolar linealmente información de las líneas analizadas es necesario repetir dos veces como mínimo los procedimientos 5 y 6. El resultado final será la estimación de temperatura electrónica.

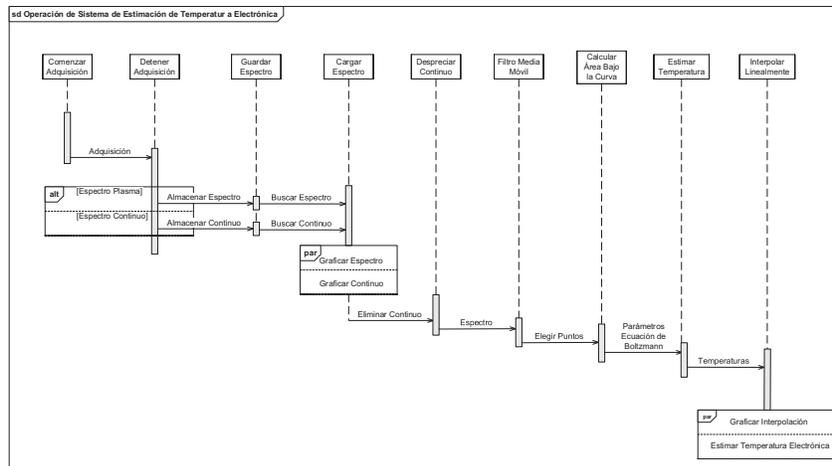


Figura 2.9 Diagrama de Interacción de la Operación del Sistema de Estimación de Temperatura Electrónica

CAPÍTULO 3. DISEÑO DEL SISTEMA

En este apartado se muestran y explican los formularios y diagramas de clase usados para construir la aplicación en código Java siguiendo el modelado presentado en el capítulo anterior.

3.1 Descripción General de la Aplicación

La aplicación completa codificada en Java consta de dos operaciones fundamentales: Traer Tabla del NIST y Analizar Espectro, por lo que dichas actividades se encuentran contenidas en un menú de opciones dentro de un formulario padre. La Figura 3.1 muestra las operaciones generales que realiza el sistema.



3.1 Formulario Principal de Actividades del Sistema

3.2 Módulo Traer Tabla de Elemento

El módulo correspondiente a traer tabla de elemento permite trasladar a nivel local diversas tablas de especies químicas consultadas en la base de datos Atomic Spectra Database [14]. Dicho módulo se ejecuta al dar clic en el botón **Traer Tabla del Nist** del formulario principal.

Como se mencionó en la sección 2.4 del capítulo anterior, el usuario tiene que consultar la tabla del elemento químico deseado haciendo uso de una conexión a internet, posteriormente esta debe ser guardada en formato HTML para que al ser leída se puedan extraer y almacenar los parámetros de la ecuación de Boltzmann (ec. 1.3).

La clase **TraerDeHTML** (Figura 3.2) muestra dos atributos de clase y un método que no devuelve/retorna ningún tipo de dato:

- **Dato** es utilizado para almacenar la información numérica que se recolecta al identificar las etiquetas HTML correspondientes a cada columna “<td>” resultado de ir leyendo línea a línea el código HTML, cabe mencionar que es necesario validar dicha información, dado que la consulta arrojada por el NIST puede llegar a mostrar caracteres no numéricos en la lectura del mismo, puede verse en la Figura 3.3 que la columna **Rel. Int.** llega a tener caracteres de tipo “*” y las columnas E_i y E_k presentan caracteres de tipo “[]”.
- **aBase[5]**, constituye un arreglo de 5 posiciones, usadas para almacenar **Dato**, es decir, aBase[0] almacena la **Longitud de Onda Observada**, aBase[1] la **Intensidad**, aBase[2] la **Probabilidad de Transición**, aBase[3] la **Energía del Nivel Superior** y aBase[4] el **Peso Estadístico del Electrón**.
- Una vez que se tienen validados todos los parámetros de la lectura de cada fila “<tr>”, se procede a almacenarlos en una tabla local diseñada en MySQL, la sección 2.4 del capítulo anterior hace mención al diseño de cada una de las tablas que alojará ELEMENTOS.sql, por lo que el método **Guardar()** es el encargado de generar automáticamente la tabla de la especie química de interés.

TraerDeHTML
dato : doublé
aBase[5] : doublé
Guardar() : void

Figura 3.2 Diagrama de clase para traer tabla de elemento

Observed Wavelength Air (nm)	Rel. Int. (%)	A_{ki} (s ⁻¹)	Acc.	E_i (eV)	E_k (eV)	$g_i - g_k$	Type
402.3973	1	1.1281e+06	AAA	[21.2180220941]	- [24.2982851667]	3 - 1	
402.61914	50*	3.2224e+05	AAA	[20.96408629014]	- [24.042662765801]	5 - 3	
402.61914	50*	2.8999e+06	AAA	[20.96408629014]	- [24.042661593090]	5 - 5	
402.61914	50*	1.1601e+07	AAA	[20.96408629014]	- [24.042661513599]	5 - 7	
402.61914	50*	4.8336e+06	AAA	[20.96409576568]	- [24.042662765801]	3 - 3	
402.61914	50*	8.6997e+06	AAA	[20.96409576568]	- [24.042661593090]	3 - 5	
412.08154	12*	2.4738e+06	AAA	[20.96408629014]	- [23.9719706931]	5 - 3	
412.08154	12*	1.4843e+06	AAA	[20.96409576568]	- [23.9719706931]	3 - 3	
414.3761	3	4.8812e+06	AAA	[21.2180220941]	- [24.209249058389]	3 - 5	
416.8967	1	1.8298e+06	AAA	[21.2180220941]	- [24.1911595405]	3 - 1	

Figura 3.3 Ejemplo de presentación de tabla del NIST de Líneas Espectrales de Hel 200-900nm

La interfaz de usuario usada para traer tabla de elemento se muestra en la Figura 3.4. Puede verse que se debe buscar el archivo HTML guardado por la consulta realizada a través de internet y definir su nombre (esto siguiendo la sugerencia antes mencionada en la sección 2.4 del capítulo 2).

Figura 3.4 Interfaz de Usuario del Módulo Traer Tabla de Elemento

3.3 Módulo de Análisis y Visualización de Espectro

El módulo de análisis y visualización del espectro permite realizar el cálculo de la temperatura electrónica. Para realizar dicha actividad el usuario debe de dar clic en el segundo botón del menú de opciones del formulario principal Figura 3.1, esta

acción permite elegir el espectro de emisión y el espectro continuo que previamente han sido adquiridos y almacenados en formato “xls”, en la Figura 3.5 puede notarse que es necesario buscar la ubicación del espectro de emisión (1) y del espectro continuo (2). La parte ilustrada por (3) permite seleccionar un ventaneo de filtro de media móvil (5) con un intervalo cerrado [3-15] con el fin de suavizar la curva del espectro que resulta de la diferencia del espectro de emisión (1) y el espectro continuo (2). Cabe mencionar que el usuario, si así lo desea, puede analizar el espectro “con” o “sin” la aplicación del filtro (4).

1 Archivo de Espectro

BUSCAR ARCHIVO DE ESPECTRO

Nombre:

Path:

2 Archivo de Continuo

BUSCAR ARCHIVO DE CONTINUO

Nombre:

Path:

3 Opciones de Filtro

4 Aplicar Filtro **5** Ventaneo del Filtro:

3.5 Búsqueda de Espectros y Aplicación de Filtro

La Figura 3.6 muestra la comparación de dos espectros, en donde en el primero “no” se aplica un filtro media móvil y en donde en el segundo “si” se aplica un filtro de media móvil.

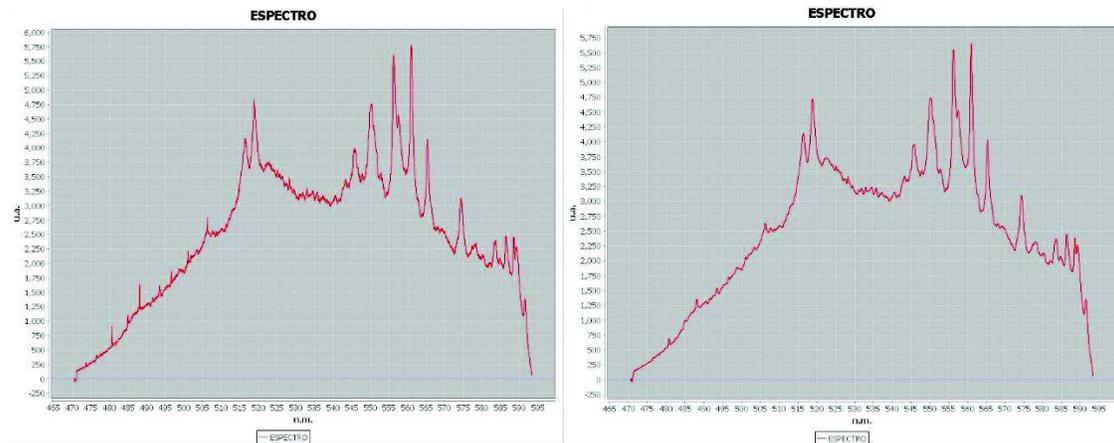


Figura 3.6 Comparación de Espectros

a) Sin Filtro Media Móvil

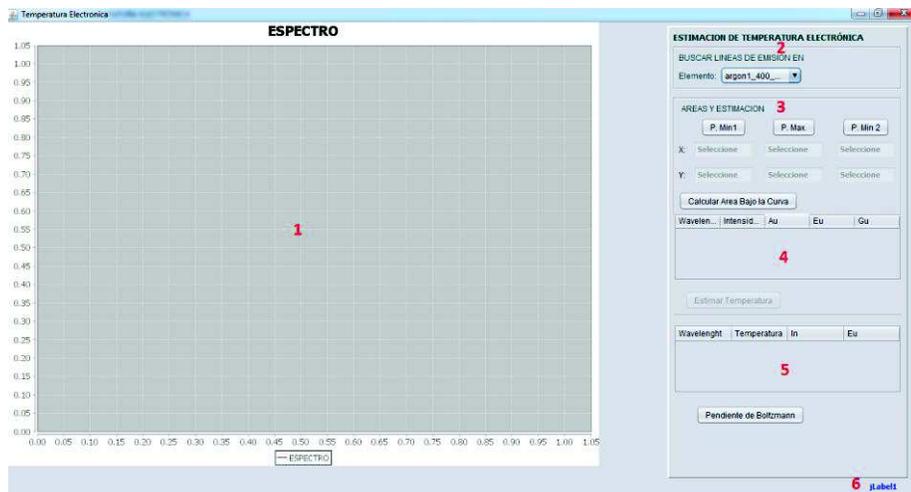
b) Con Filtro Media Móvil

Para realizar el análisis y el cálculo de temperatura electrónica se propone la siguiente interfaz gráfica (Figura 3.7). A continuación se describen los elementos que integran esta pantalla.

1. **Panel de visualización de espectros:** Una vez que se desprecia el espectro continuo del espectro de línea, se procede a cargarlo en un control gráfico. Permite realizar acercamientos Figura 3.8 y elección de puntos coordenados tal y como lo muestra la Figura 3.9.
2. **Listado de tablas:** En esta lista se realiza una consulta de todas las tablas de especies químicas que con anterioridad han sido importadas desde el NIST, para de este modo buscar líneas de emisión en el elemento de elección.
3. **Área bajo la curva:** En este apartado se realiza la selección de puntos coordenados: Un punto máximo (longitud de onda observada), y dos puntos como base, para que con éstos se realice el cálculo del área bajo la curva de la longitud de onda observada, cabe mencionar, que para elegir estos puntos el usuario se ayuda del panel de visualización de espectros.
4. **Panel de consulta de parámetros de ecuación de Boltzmann:** En este elemento de tipo tabla se muestran los parámetros: **Longitud de Onda Observada**, la **Intensidad**, la **Probabilidad de Transición**, la **Energía del**

Nivel Superior y el **Peso Estadístico del Electrón**. El usuario debe hacer la elección correcta de los mismos siguiendo el criterio de longitud de onda observada y de intensidad de la línea, por lo que en la tabla se cargan los parámetros correspondientes a la consulta de las 5 líneas de arriba y las 5 líneas de abajo de la longitud de onda observada que proporciona el usuario en la sección de Área Bajo la Curva. Con todos estos datos se estima la temperatura electrónica de la línea elegida.

- 5. Panel de temperaturas electrónicas estimadas:** En este elemento de tipo tabla se muestran todas las temperaturas electrónicas que el usuario ha calculado, para que con ellas se realice un promedio y se estime así la temperatura electrónica.
- 6. Etiqueta de control:** En esta etiqueta se va mostrando el estado y/o resultados de las actividades que el usuario va realizando a lo largo del análisis espectral: Área bajo la curva, temperatura electrónica, pendiente, etc.



3.7 Pantalla de análisis y visualización de espectros

Uno de los elementos fundamentales del Módulo descrito en esta sección es el elemento en donde se visualizan los espectros, ya que es necesario realizar acercamientos de partes seccionadas del espectro y con ello contribuir a un mejor análisis por parte del usuario. La Figura 3.8 muestra que al dibujar un área rectangular con el cursor del ratón es posible realizar la acción descrita.

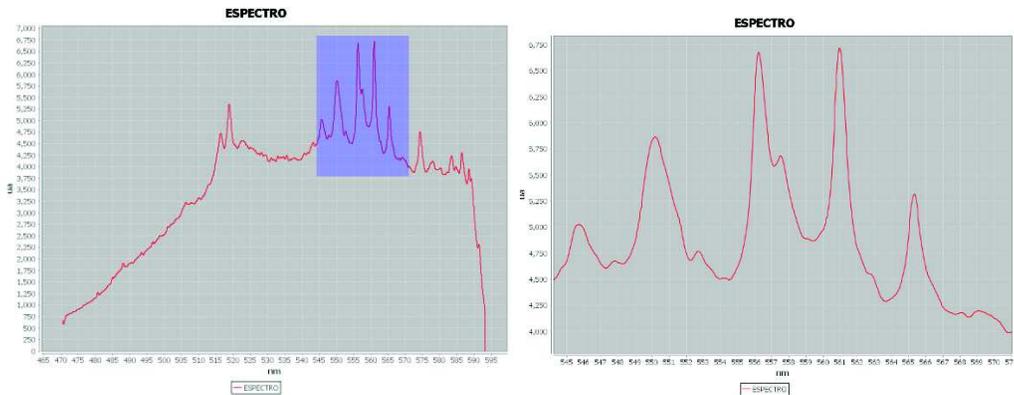


Figura 3.8 Resultado de acercamiento a sección seleccionada del espectro

Para realizar la elección de puntos en el mismo control visual, se genera una etiqueta que muestre los puntos coordenadas (x,y) en donde se encuentra posicionado el cursor, con el fin de visualizar fácilmente en qué lugar del espectro se encuentra el usuario. Del mismo modo al dar clic en dicho elemento se dibujan dos líneas punteadas (vertical y horizontal) con la finalidad de que la intersección sea un punto base o un punto longitud de onda observada elegido por el usuario (Figura 3.9).

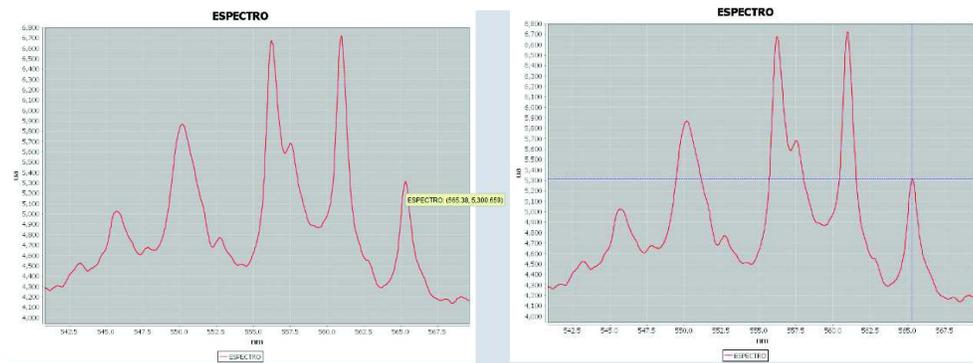


Figura 3.9 Elección de puntos en control visual

Para graficar el espectro sobre un componente gráfico, es necesario programar una clase java. La clase **ParaGraficarEspectros** (Figura 3.10) muestra siete atributos de clase y 5 métodos:

- **Gráfica:** Es usado para construir diversos tipos de gráficos: Lineal, Polar, Dispersión, Área, Serie de tiempo.
- **Datos:** Es una colección de datos, contiene la información referente a los puntos coordinados del espectro que se han de graficar
- **Título:** Es un dato de tipo String el cual se pasará como parámetro al método constructor, mismo que lleva el nombre del gráfico.
- **Tx:** Es un dato de tipo String el cual se pasará como parámetro al método constructor, mismo que lleva el nombre del eje X.
- **Ty:** Es un dato de tipo String el cual se pasará como parámetro al método constructor, mismo que lleva el nombre del eje Y.
- **tipoGráfica:** Ya que el tipo de gráfico requerido para visualizar el espectro es el lineal, se recibe siempre un uno como parámetro ("1 entero") para que en un elemento JPanel se visualice el tipo de gráfico requerido.
- **agregarGráfica:** Este método se encarga de dibujar sobre un JPanel todos los elementos necesarios para construir el gráfico: El nombre de la serie, el dominio (datos leídos para el eje X) y el rango (datos leídos para el eje Y), por lo que los parámetros recibidos: **double [] x** y **double [] y**, son dos arreglos que deben contener los datos previamente procesados de la diferencia del espectro de emisión y del espectro continuo, previamente adquiridos. Actividad previamente ilustrada en la Figura 3.5.
- **obtienePanel:** Es el encargado de traer el eje coordinado (x,y) que elija el usuario tal y como se mencionó anteriormente. La Figura 3.9 muestra un ejemplo de esta acción.
- **ConfigurarRendered:** Este método, es usado para mostrar el nombre y el color de la serie, este último se eligió de un tono rojo por gusto propio. En la Figura 3.7 se puede ver claramente que en la parte inferior se muestra la serie que representa e identifica al espectro.

ParaGraficarEspectros
Gráfica: JFreechart
Datos: XYSeriesCollection
Título: String
Tx: String
Ty: String
x: double
y: double
ParaGraficarEspectros (int tipo, String título, String tx1, String ty1): Constructor
tipoGráfica (int tipo): void
agregarGrafica (String id, double [] x, double [] y):void
obtienePanel: ChartPanel
configurarRendered (XYLineAndShapeRenderer renderer): void

Figura 3.10 Diagrama de clase para graficar espectros

Para estimar la temperatura electrónica del espectro visualizado sobre el componente visual, es necesario interpolar linealmente los puntos E_u , $In\left(\frac{\epsilon\lambda}{Au\ gu}\right)$ de dos o más líneas observadas conforme a la ec. 1.3 longitudes de onda con el fin de reducir el error de estimación. La Figura 3.11 muestra la aproximación lineal (1) de puntos de dos líneas graficados sobre E_u e In .

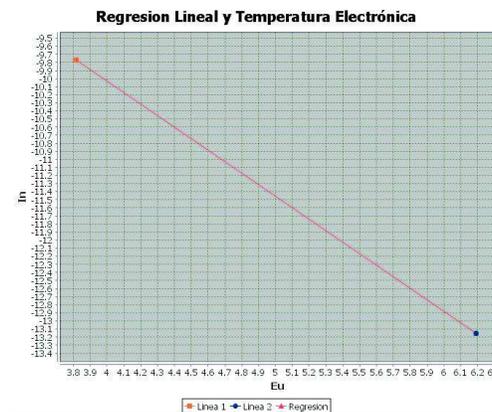


Figura 3.11 Pantalla de regresión lineal

Para lograr lo anterior, es necesario programar un submódulo de programación que permita realizar dicha actividad. Por lo que el formulario en donde se presenta el gráfico de aproximación lineal hace uso de la clase **ParaGraficarRegresiones** (Figura 3.12) descrita a continuación:

ParaGraficarRegresiones
chart: JFreechart COLOR_RECUADROS_GRAFICA: Color COLOR_FONDO_GRAFICA: Color
configurarPlot (XYPlot plot):void configurarDomainAxis (NumberAxis domainAxis):void configurarRangeAxis (NumberAxis rangeAxis): void configurarRendered (XYLineAndShapeRenderer renderer): void crearGrafica(XYSeriesCollection dataset): JFreeChart

Figura 3.12 Diagrama de clase para graficar regresiones

- **Chart:** Usado para generar un gráfico de tipo lineal, así como para asignar su nombre y los títulos de los ejes.
- **COLOR_RECUADROS_GRAFICA:** Es un dato de tipo Color (R, G, B) para dibujar las líneas de cuadrícula del gráfico, con el fin de identificar más fácilmente la información presentada.
- **COLOR_FONDO_GRAFICA:** Es un dato de tipo Color (R, G, B) usado para asignar un color de fondo en el gráfico
- **configurarPlot:** En este método se hace uso de los atributos de clase: COLOR_RECUADROS_GRAFICA y COLOR_FONDO_GRAFICA para que en el grafico se dibujen las rejillas o líneas de cuadrícula para el eje X (**Eu**) y el eje Y (**In**).
- **configurarDomainAxis:** Este método es usado para que los avances hacia la derecha del eje X (**Eu**) vayan de 0.1 unidades, a partir del origen (primer dato encontrado por el gráfico). Esto puede verse claramente en la Figura 3.11.

- **configurarRangeAxis:** Este método es usado para que los avances hacia la derecha del eje Y (*In*) vayan de 0.1 unidades, a partir del origen (primer dato encontrado por el gráfico). Esto puede verse claramente en la Figura 3.11.
- **configurarRendered:** Usado para mostrar los colores, formas y nombres de leyenda del gráfico.
- **crearGráfica:** Este método devuelve un objeto de tipo JFreeChart, usado para dibujar sobre un JPanel el gráfico resultante de la colección recibida como parámetro, que contiene la información que ha de ser graficada.

CAPITULO 4. PRUEBAS Y RESULTADOS

En este capítulo se muestran las pruebas realizadas para la determinación de la temperatura electrónica para comprobar el correcto funcionamiento del sistema.

Al adquirir espectros de línea de una fuente como el plasma, es común que se encuentre montado sobre un espectro continuo, lo que ocasiona que exista un error mayor en el cálculo de la temperatura electrónica. Para reducir este efecto, se usa una lámpara de calibración como una fuente de tungsteno, mercurio, entre otras y se extrae del espectro adquirido, este proceso ayuda además a reducir algunos efectos que puede provocar el propio hardware, como desplazamientos en la respuesta. Un espectro de una lámpara de tungsteno puede apreciarse en la Figura 4.1.

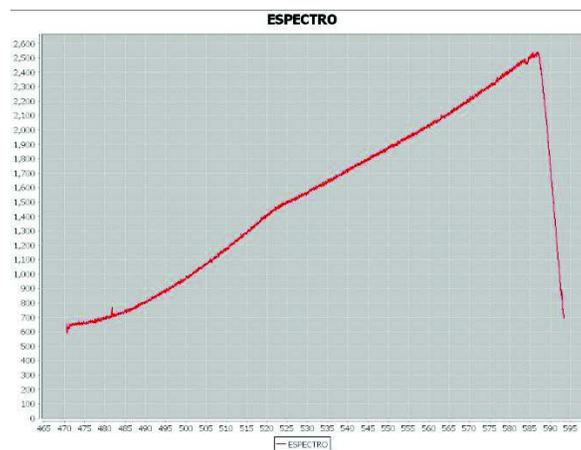


Figura 4.1 Espectro Continuo de Lámpara de Tungsteno

El dispositivo de donde se adquiere el espectro es una antorcha de plasma no transferido de CD con una corriente eléctrica de 100A, un voltaje entre electrodos de 30V y un flujo de gas de Argón a 7.5Lpm a presión atmosférica., se realizó un barrido axial Figura 4.2 del centro hacia una de las orillas de la fuente antes mencionada. En los espectros adquiridos se identificaron dos líneas de cobre emitiendo en 510.55 n.m. y 515.32 n.m.

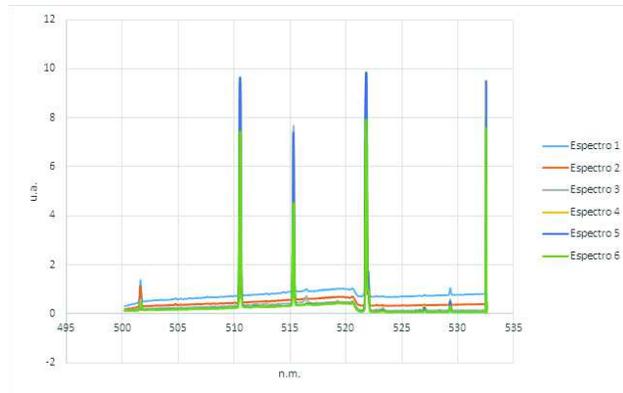


Figura 4.2 Barrido de Espectros desde el centro hacia una orilla

Una vez adquirido el espectro de una antorcha de plasma y despreciando el continuo, se procede a cargarlo en un elemento visual para ser analizado. La temperatura electrónica puede calcularse con analizar una longitud de onda, pero esto puede no representar el valor real, por lo que se requieren más para poder obtener un promedio y así proporcionar un valor más cercano a la realidad [31, 32, 33]. Por lo anterior, se recomienda calcularla mediante la interpolación lineal de diversas temperaturas tomando dos o más líneas espectrales.

4.1 Pruebas de Estimación de Temperatura Electrónica sin Filtro Media Móvil

Las pruebas iniciales se realizaron en el orden de mayor a menor intensidad sobre dos longitudes de onda. La tabla 4.1 muestra los parámetros de estimación de 2 diferentes longitudes de onda del espectro “Espectro 1” de la Figura 4.2.

Tabla 4.1 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1

λ (nm)	Temperatura (T)	$\ln\left(\frac{\epsilon\lambda}{Au\ gu}\right)$	Energía E_u
510.55	4533.9299	-9.7687	3.8166
512.32	5461.2722	-13.1555	6.1911

Al obtener los valores del lado izquierdo y derecho de la ecuación de Boltzmann (ecuación 1.3) se realiza un ajuste lineal Figura 4.3. La temperatura se obtiene a partir de la ecuación 4.1 al despejar T [34], obteniendo una temperatura de 8136.1344 °K (0.7014 eV).

$$-\frac{1}{kT} = m \quad (\text{ecuacion 4.1})$$

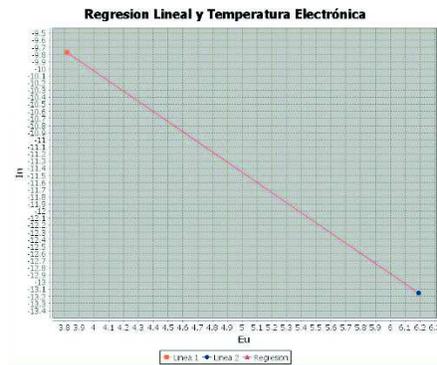


Figura 4.3 Pendiente de Boltzmann para Espectro 1

Del mismo modo, se procede a calcular la temperatura del “Espectro 2” de la Figura 4.2. La tabla 4.2 muestra los pares coordenados Eu , $ln\left(\frac{\epsilon\lambda}{A u g u}\right)$ de dos líneas, al realizar la interpolación lineal (Figura 4.4), se obtiene una temperatura de 7718.7547 °K (0.6654 eV).

Tabla 4.2 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 2

λ (nm)	Temperatura (T)	$ln\left(\frac{\epsilon\lambda}{A u g u}\right)$	Energía Eu
510.55	4363.0931	-10.1512	3.8166
512.32	5236.1415	-13.7211	6.1911

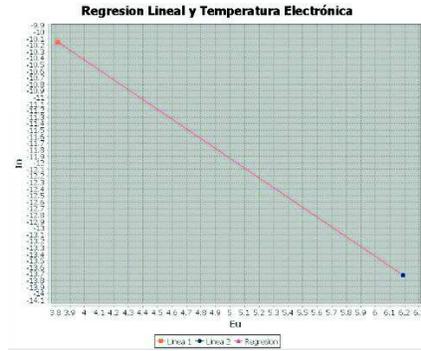


Figura 4.4 Pendiente de Boltzmann para Espectro 2

Por otro lado, se procede a calcular la temperatura del “Espectro 3” de la Figura 4.2. La tabla 4.3 muestra los pares coordenados E_u , $\ln\left(\frac{\epsilon\lambda}{A_{u g u}}\right)$ de dos líneas, al realizar el ajuste lineal (Figura 4.5), se obtiene una temperatura de 7415.8426 °K (0.6393 eV).

Tabla 4.3 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 3

λ (nm)	Temperatura (T)	$\ln\left(\frac{\epsilon\lambda}{A_{u g u}}\right)$	Energía E_u
510.55	4818.7744	-9.1913	3.8166
512.32	5566.4196	-12.9070	6.1911

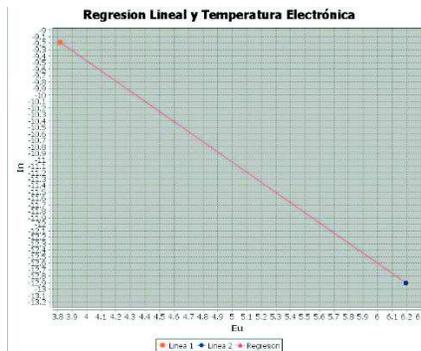


Figura 4.5 Pendiente de Boltzmann para Espectro 3

A continuación, se procede a calcular la temperatura del “Espectro 4” de la Figura 4.2. La tabla 4.4 muestra los pares coordenados E_u , $\ln\left(\frac{\epsilon\lambda}{A_{u g u}}\right)$ de dos líneas, al realizar el ajuste lineal (Figura 4.6), se obtiene una temperatura de 7342.8464°K (0.6330 eV).

Tabla 4.4 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 4

λ (nm)	Temperatura (T)	$\ln\left(\frac{\epsilon\lambda}{A_{u g u}}\right)$	Energía E_u
510.55	4824.0949	-9.1812	3.8166
512.32	5554.8836	-12.9338	6.1911

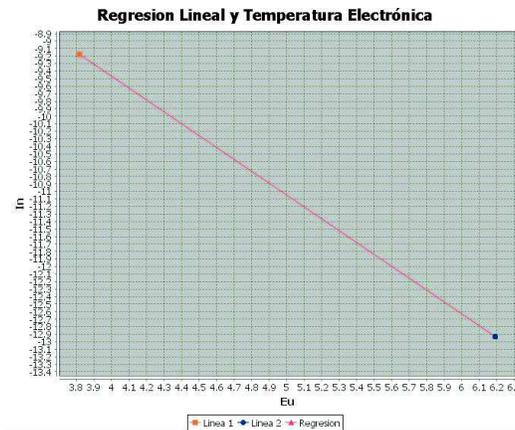


Figura 4.6 Pendiente de Boltzmann para Espectro 4

Del mismo modo, se procede a calcular la temperatura del “Espectro 5” de la Figura 4.2. La tabla 4.5 muestra los pares coordenados E_u , $\ln\left(\frac{\epsilon\lambda}{A_{u g u}}\right)$ de dos líneas, al realizar la interpolación lineal (Figura 4.7), se obtiene una temperatura de 7249.0795°K (0.6249 eV).

Tabla 4.5 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 5

λ (nm)	Temperatura (T)	$\ln\left(\frac{\epsilon\lambda}{Au\ gu}\right)$	Energía E_u
510.55	4849.3262	-9.1334	3.8166
512.32	5554.5530	-12.9345	6.1911

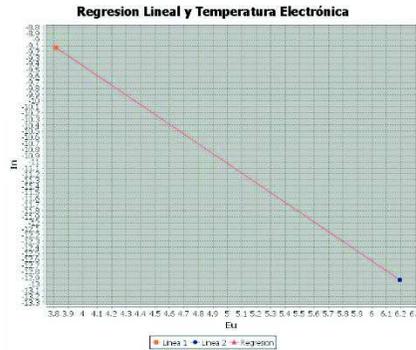


Figura 4.7 Pendiente de Boltzmann para Espectro 5

Por otro lado, se procede a calcular la temperatura del “Espectro 6” de la Figura 4.2. La tabla 4.6 muestra los pares coordenados E_u , $\ln\left(\frac{\epsilon\lambda}{Au\ gu}\right)$ de dos líneas, al realizar el ajuste lineal (Figura 4.8), se obtiene una temperatura de 7134.9906 °K (0.6151 eV).

Tabla 4.6 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 6

λ (nm)	Temperatura (T)	$\ln\left(\frac{\epsilon\lambda}{Au\ gu}\right)$	Energía E_u
510.55	4642.0513	-9.5412	3.8166
512.32	5360.3546	-13.4031	6.1911

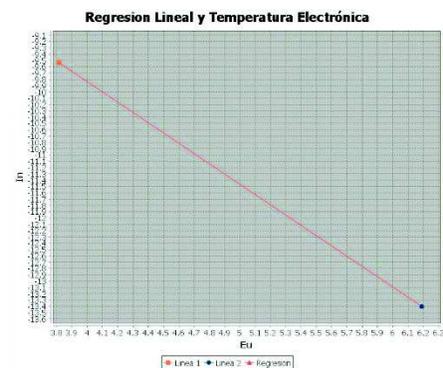


Figura 4.8 Pendiente de Boltzmann para Espectro 6

4.2 Pruebas de Estimación de Temperatura Electrónica con Filtro Media Móvil

Con el fin de reducir aún más el error de los cálculos, se agregó un filtro de media móvil, con este se reduce el ruido (se suaviza la curva del espectro). Se realizaron estimaciones con ventaneos al azar para verificar el efecto del mismo.

Se procede a calcular la temperatura del “Espectro 1” de la Figura 4.2 con un ventaneo de 5 unidades. La tabla 4.7 muestra los pares coordenados Eu , $ln\left(\frac{\epsilon\lambda}{A_{\text{u gu}}}\right)$ de dos líneas, al realizar el ajuste lineal (Figura 4.9), se obtiene una temperatura de 8137.5113°K (0.7015 eV).

Tabla 4.7 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1 con Filtro

λ (nm)	Temperatura (T)	$ln\left(\frac{\epsilon\lambda}{A_{\text{u gu}}}\right)$	Energía Eu
510.55	4533.9410	-9.7687	3.8166
512.32	5461.5200	-13.1549	6.1911

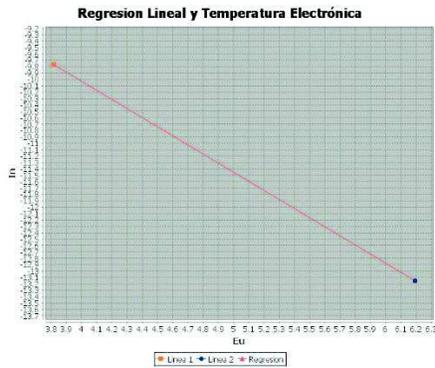


Figura 4.9 Pendiente de Boltzmann para Espectro 1 con Filtro

Por otro lado, se procede a calcular la temperatura del “Espectro 2” de la Figura 4.2 con un ventaneo de 10 unidades. La tabla 4.8 muestra los pares coordenados E_u , $ln\left(\frac{\epsilon\lambda}{A_{u g u}}\right)$ de dos líneas, al realizar su interpolación lineal (Figura 4.10), se obtiene una temperatura de 7797.6613°K (0.6722 eV).

Tabla 4.8 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1 con Filtro

λ (nm)	Temperatura (T)	$ln\left(\frac{\epsilon\lambda}{A_{u g u}}\right)$	Energía E_u
510.55	4363.0920	-10.1512	3.8166
512.32	5249.9623	-13.6850	6.1911

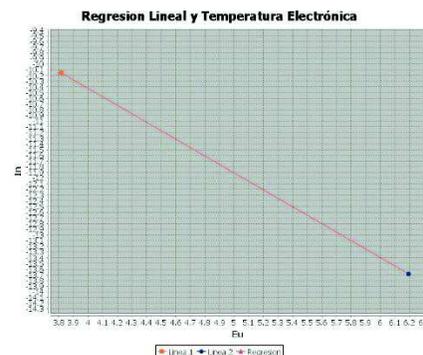


Figura 4.10 Pendiente de Boltzmann para Espectro 2 con Filtro

Del mismo modo, se procede a calcular la temperatura del “Espectro 3” de la Figura 4.2 con un ventaneo de 4 unidades. La tabla 4.9 muestra los pares coordenados E_u , $\ln\left(\frac{\epsilon\lambda}{A_{u\ g u}}\right)$ de dos líneas, al realizar un ajuste lineal (Figura 4.11), se obtiene una temperatura de 7413.9250°K (0.6392 eV).

Tabla 4.9 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1 con Filtro

λ (nm)	Temperatura (T)	$\ln\left(\frac{\epsilon\lambda}{A_{u\ g u}}\right)$	Energía E_u
510.55	4818.7028	-9.1914	3.8166
512.32	5565.9447	-12.9081	6.1911

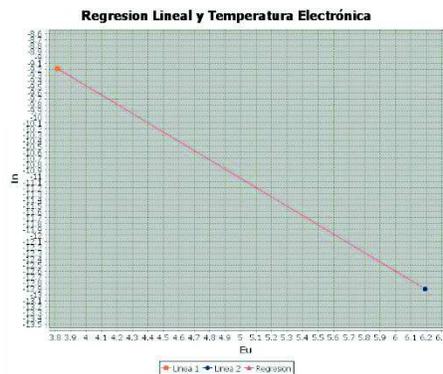


Figura 4.11 Pendiente de Boltzmann para Espectro 3 con Filtro

Se procede a calcular la temperatura del “Espectro 4” de la Figura 4.2 con un ventaneo de 9 unidades. La tabla 4.10 muestra los pares coordenados E_u , $\ln\left(\frac{\epsilon\lambda}{A_{u\ g u}}\right)$ de dos líneas, al realizar su interpolación lineal (Figura 4.12), se obtiene una temperatura de 7340.3344°K (0.6328 eV).

Tabla 4.10 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1 con Filtro

λ (nm)	Temperatura (T)	$ln\left(\frac{\epsilon\lambda}{Au\ gu}\right)$	Energía Eu
510.55	4824.0533	-9.1812	3.8166
512.32	5554.2982	-12.9351	6.1911

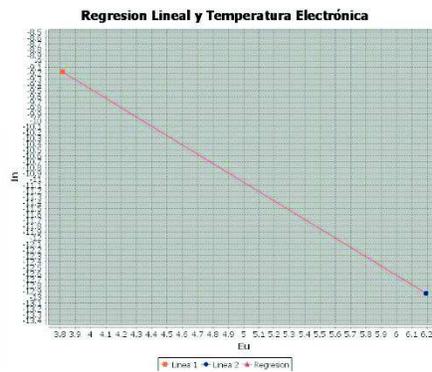


Figura 4.12 Pendiente de Boltzmann para Espectro 4 con Filtro

Por otro lado, se procede a calcular la temperatura del “Espectro 5” de la Figura 4.2 con un ventaneo de 7 unidades. La tabla 4.11 muestra los pares coordenados Eu , $ln\left(\frac{\epsilon\lambda}{Au\ gu}\right)$ de dos líneas, al realizar su interpolación lineal (Figura 4.13), se obtiene una temperatura de 7258.0562°K (0.6257 eV).

Tabla 4.11 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1 con Filtro

λ (nm)	Temperatura (T)	$ln\left(\frac{\epsilon\lambda}{Au\ gu}\right)$	Energía Eu
510.55	4845.8973	-9.1399	3.8166
512.32	5553.7967	-12.9363	6.1911

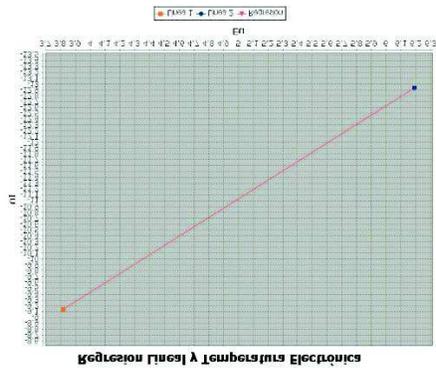


Figura 4.13 Pendiente de Boltzmann para Espectro 5 con Filtro

Del mismo modo, se procede a calcular la temperatura del “Espectro 6” de la Figura 4.2 con un ventaneo de 6 unidades. La tabla 4.12 muestra los pares coordenados E_u , $\ln\left(\frac{\epsilon\lambda}{A_{u g u}}\right)$ de dos líneas, al realizar su ajuste lineal (Figura 4.14), se obtiene una temperatura de 7132.7160°K (0.6149 eV).

Tabla 4.12 Temperaturas Electrónicas de diferentes longitudes de onda Espectro 1 con Filtro

λ (nm)	Temperatura (T)	$\ln\left(\frac{\epsilon\lambda}{A_{u g u}}\right)$	Energía E_u
510.55	4642.1552	-9.5410	3.8166
512.32	5359.9474	-13.4042	6.1911

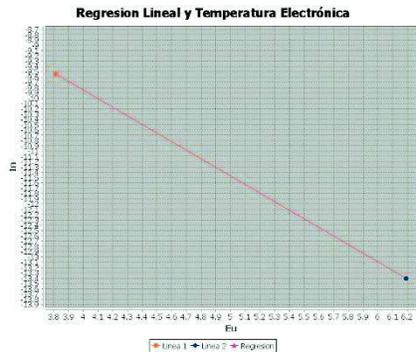


Figura 4.14 Pendiente de Boltzmann para Espectro 6 con Filtro

A continuación se muestra una tabla comparativa de las estimaciones de los barridos sin filtro contra las estimaciones de los barridos con filtro. En tabla 4.13 puede verse que las temperaturas son similares, las discrepancias se deben a que el filtro suaviza al espectro, por lo tanto las áreas bajo la curva de una longitud de onda observada son siempre distintas.

Tabla 4.13 Tabla de Pruebas de Temperaturas Electrónicas de barridos del lado izquierdo

BARRIDOS DEL LADO IZQUIERDO DE LA FUENTE DE PLASMA					
ESPECTRO	SIN FILTRO		CON FILTRO		VENTANEO
	T. EN ° K	T. EN eV.	T. EN ° K	T. EN eV.	
Espectro 1	8136.1344	0.7014	8137.5113	0.7015	5
Espectro 2	7718.7547	0.6654	7797.6613	0.6722	10
Espectro 3	7415.8426	0.6393	7413.9250	0.6392	4
Espectro 4	7342.8464	0.6330	7340.3344	0.6328	9
Espectro 5	7249.0795	0.6249	7258.0562	0.6257	7
Espectro 6	7134.9906	0.6151	7132.7160	0.6149	6

Al realizar una gráfica comparativa de las temperaturas obtenidas “Con Filtro” y “Sin Filtro” (Figura 5.15) puede verse que las temperaturas electrónicas son similares, el hecho por el cual son diferentes es que al aplicar el filtro media móvil, se obtiene un área bajo la curva distinta sobre la línea de estudio. Del mismo modo, se aprecia

que los espectros con mayor intensidad (Figura 4.2) presentan mayores temperaturas.

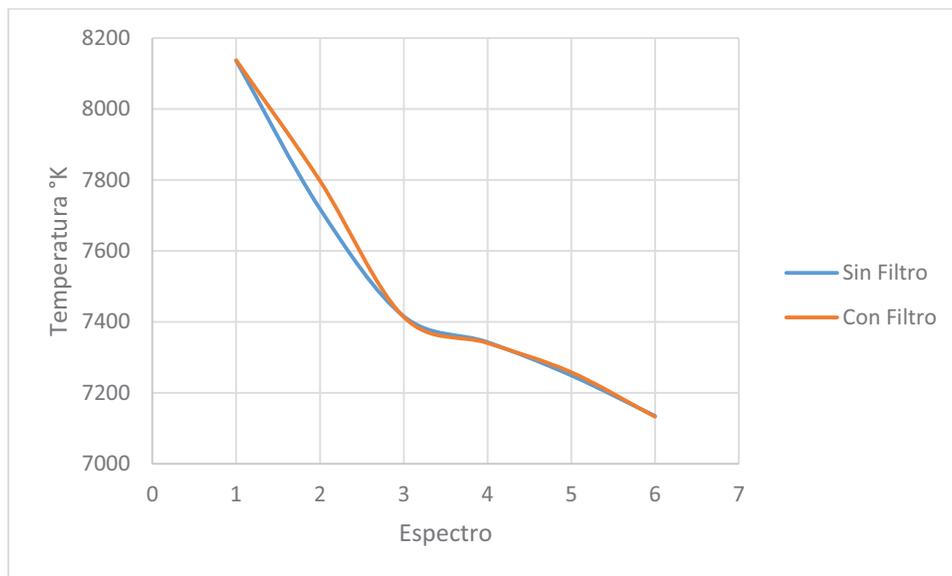


Figura 5.15 Comparación de temperaturas

4.3 Conclusiones

El programa codificado en lenguaje java permite estimar la temperatura electrónica de una fuente de emisión de plasma.

Las respuestas de los análisis con filtro media móvil respondieron favorablemente en comparación con los que no se aplica el filtrado.

La aplicación descrita representa una herramienta que disminuye los tiempos de análisis, consulta y cómputo de la estimación de temperatura electrónica en el estudio de espectros ópticos

La base de datos local generada por el sistema puede ser usada para trabajos futuros.

4.4 Trabajos a Futuro

La implementación de una red neuronal podría ayudar a clasificar líneas de emisión de especies presentes en el espectro, haciendo uso de la base de datos local, por lo que se requeriría traer el mayor número de tablas de elementos en amplios rangos de observación y niveles de ionización.

Sería deseable desarrollar una versión en donde se puedan obtener los espectros en tiempo real.

En relación a las pruebas del sistema sobre los barridos realizados, se sugiere desarrollar un prototipo de software en donde se puedan sobreponer varios espectros de la misma fuente de emisión y con ello calcular la temperatura electrónica de cada uno de ellos como si se tratase de uno solo.

BIBLIOGRAFIA Y REFERENCIAS

- [1] García B., Conde O., “Espectroscopia óptica de imagen para el control de calidad en la industria alimentaria, monitorización en línea de procesos de soldadura y discriminación de patologías tumorales en tejidos cancerígenos”, *Óptica Pura y Aplicada*. 2012.
- [2] Parra E., Riaño J., Serna A., Cárdenas D., Prieto F., “Aplicación del Procesamiento Digital de Señales al Estudio de Espectros de Emisión Óptica”, *Redalyc Dyna*, vol. 76, Septiembre.2009, pp. 205-215.
- [3] “Full-Frame CCD Operation”, Abril 2015, Disponible: <http://www.olympusmicro.com/primer/java/digitalimaging/ccd/fullframe/index.html>
- [4] Rengifo M., Vargas C., “Caracterización óptica de Diodos emisores de Luz mediante su espectros de emisión y patrones de radiación”, *Scientia Et Technica*, vol. 17, Agosto 2012, pp. 66-70.
- [5] López J., Mandujano C., “Sistema de Absorción Óptica Diferenciada del Espectro (DOAS) para evaluar la calidad del aire”, *Revista CENIC, Ciencias Químicas*, vol. 36, 2015.
- [6] Boselli M., Colombo V., Ghedini E., Gherardi M., Sanibondi P., “Two-temperature modelling and optical emission spectroscopy of a constant current plasma arc welding process”, *Journal of Physics*, vol. 46, Mayo 2013.
- [7] Sarmiento R., “Aplicación De La Espectroscopía Óptica De Emisión En La Caracterización De Plasmas De Descargas Eléctricas En Tubos De Lámparas HPS Y Plasmas Láser En Aire”, *Bistua: Revista de la Facultad de Ciencias Básicas*, Vol. 6, 2008, pp. 1-6
- [8] F. Francis Chen, “Introduction to Plasma Physics and Crontrrolled Fusion: Plasma Physics”, 2ª. Ed., New Cork, EEUU, Plenum Press, 1984, pp. 40-65.
- [9] Cruz A., “Control y Monitoreo por Software de Experimentos en el Tratamiento de Gases por Plasma, desde una Perspectiva Sistemática”, *Centro Universitario UAEM Ecatepec, México*, 2013.
- [10] Torres C. “Campo magnético transverso y sincronizado para rotación de plasmas”, *Toluca, México*, 2010.
- [11] Fantz U., “Basics of Plasma Spectroscopy”, *Plasma Sources Science and Technology*, vol. 15, October 2006, pp. 137-138.

[12] Keeler J., “Understanding NMR Spectroscopy”, University of Cambridge, Department of Chemistry, 2002, pp. 6-8

[13] Salcedo K., Rodríguez A., Román S., Riascos H., “Análisis Espectroscópico De Un Plasma De Aluminio En Atmósfera De Nitrógeno Producido Por Láser Pulsado”, Revista de la Facultad de Ciencias Básicas de la Universidad de Pamplona Colombia, Vol. 7, pp. 1-5, Dic.2009.

[14] “NIST Atomic Spectra Database Line Form”, Septiembre 2015. Disponible: http://physics.nist.gov/PhysRefData/ASD/lines_form.html

[15] García C., “Apuntes de Métodos Numéricos”, Universidad de Malaga, pp. 143-205.

[16] Delores M., “Solución de Problemas de Ingeniería con Matlab”, Pearson Prentice Hall, 2ª Ed., 1998, pp. 184-201.

[17] Sommerville I., “Ingeniería del Software”, Pearson Educación, 7ª Ed., 2005, pp. 5-12.

[18] Instituto Nacional de Tecnologías de Comunicación, “Curso de Introducción a la Ingeniería del Software”, Laboratorio de la Calidad del Software, España, 2009, pp. 23-32.

[19] Presmman R., “Ingeniería del Software un Enfoque Práctico”, McGrawHill, 5ª ed., 2009, pp. 16-39.

[20] Weitzenfeld A., “Ingeniería del Software Orientada a Objetos con UML, Java e Internet”, Thomson, 2007, pp. 15- 72.

[21] Laudon K., Laudon J., “Sistemas de Información Gerencial”, Pearson, 12ª. Ed., 2012, pp. 506-511.

[22] Objet Management Group, “OMG Systems Modeling Language”, Septiembre 2015.

[23] Kimmel P., “Manual de UML”, McGrawHill, 1ª. Ed., 2088, pp 17-43.

[24] “Introducción a SysML”, Noviembre 2015, Disponible: http://www.atego.com/downloads/webinars/Introduccion-a-SysML_120627.pdf

[25] Rumbaugh J, Jacobson I., Booch G., “El Lenguaje Unificado de Modelado Manual de Referencia”, Addison Wesley, 1ª Ed., 2000, pp. 75-80.

[26] Velarde O., Murillo M., Gómez L., Castillo F., “Introducción a la Programación Orientada a Objetos”, Pearson Educación, 2006, pp. 8-10.

[27] Froute A., “Java 2 Manual de Usuario y Tutorial”, Alfaomega, 2006, pp. 7-12.

- [28] “Introduction to JFreeChart”, Octubre 2015, Disponible: <http://www.oracle.com/technetwork/articles/marx-jchart-085298.html>
- [29] “Java Excel API - A Java API to read, write and modify Excel spreadsheets”, Octubre 2015. Disponible: <http://jexcelapi.sourceforge.net/>
- [30] “Curso de Predicción Económica y Empresarial”, Octubre 2015, Disponible: https://www.uam.es/docencia/predysim/predysim/2_3_ficha.htm
- [31] Valdivia R., Pacheco J., Mercado A., “Temperature Evaluation of the Nonthermal Equilibrium of Plasma Discharges by OES Analysis”, IEEE, Laser Physics, Vol. 18, pp. 1-5
- [32] Pacheco M., Pacheco J., Moreno H., Valdivia R., Santana A., “OES Analysis in a Nonthermal Plasma Used for Toxic Gas Removal: Rotational and Excitation Temperature Estimation”, IEEE, Lases Physics, Vol. 18.
- [33] Harilal S., Bindhu C., Issac C., “Electron density and temperature measurements in a laser produced carbon plasma”, Journal of applied physics, No. 82, Vol. 5, 1997.
- [34] Torres C., Pacheco J., “Magnetically Assisted Transferred Plasma Torch Applied to Vitrification of Ceramic Clay”, IEEE, Transactions on Plasma Science, Vol. 35, pp. 1758-1765.

ANEXO A

El presente anexo pretende ilustrar mediante código fuente java las actividades relacionadas al cálculo del área bajo la curva.

Área bajo la curva

A continuación se describen las acciones necesarias para obtener el área bajo la curva (sección 1.3):

- Dado que los puntos coordenados del espectro a analizar están almacenados en archivos de tipo “.xls” de Excel, se requiere leerlo para tomar puntos x,y que sirvan como referencia para descomponer el área bajo la curva. “NombreEspectroFinal” almacena la diferencia del espectro de emisión y del espectro continuo.
- Se identifican dos puntos base que fueron seleccionados por el usuario en el espectro en el archivo Excel, con esto se encuentran los puntos correspondientes al eje x desde donde debe comenzar y finalizar el cálculo del área.
- Se requiere definir una base de la curva (*Variable nivel*).
- *deltax* tomará los valores correspondientes a la Altura de cada trapecio.
- Se conoce la Base Mayor y la Base Menor de cada trapecio restando el punto y *leído* menos *nivel*.
- Se calculan y se integran todas las áreas de los trapecios.

```
public double CalcularArea(){
    double deltax;
    double areas[]=new double[numFilas-1]; sirva
    area=0;
    int fila_ini_area=0;
    int fila_fin_area=0;

    try {
```

```

//archivo excel donde se almacena el espectro a analizar
Workbook archivoExcel= Workbook.getWorkbook(new File(NombreEspectroFinal));

Sheet hoja = archivoExcel.getSheet(0); //Hoja1 del libro de Excel
numFilas = hoja.getRows();

String data;
for (int fila = 0; fila < numFilas; fila++) {
// Recorre toda la primer columna de la hoja de calculo, por eso la iteración es FILA
data = hoja.getCell(0, fila).getContents();
//Encontramos la coordenada x en el archivo excel que corresponde al primer punto base
if(Double.parseDouble(data)!=Double.parseDouble(txtMin1_x.getText())){
    fila_ini_area=fila_ini_area+1;
}
else{
    fila_ini=fila_ini_area;
}
//Encontramos la coordenada x en el archivo excel que corresponde al segundo punto base
if(Double.parseDouble(data)!=Double.parseDouble(txtMin2_x.getText())){
    fila_fin_area=fila_fin_area+1;
}
else{
    fila_fin=fila_fin_area;
}
}

System.out.println("ini: "+fila_ini+" fin: "+fila_fin);
}

catch (Exception ioe) {
    ioe.printStackTrace();
}

//Se conocen la base desde donde se calcula el area
double nivel=0;

if(Double.parseDouble(txtMin1_y.getText())<Double.parseDouble(txtMin2_y.getText())){
    nivel=Double.parseDouble(txtMin1_y.getText());
}
else{
    nivel=Double.parseDouble(txtMin2_y.getText());
}
System.out.println("nivel: "+nivel);

//Se calcula las areas de cada trapezio
// ((B+b)*h)/2
for(int i=fila_ini;i<fila_fin;i++){
    if(i<numFilas-1){
        deltax=x[i+1]-x[i];
        areas[i]=deltax*(((y[i+1]-nivel)+(y[i]-nivel))/2);
    }
}

```

```
        area=area+areas[i];
    }
}
return area;
}
```