

Programa para el análisis de Espectros de Raman

Universidad de Guanajuato
División de Ciencias e Ingenierías
Laboratorio de Fotoacústica

Lara Arellano Edgar
Cárdenas Aguilar Uriel
laraae2018@licifug.ugto.mx
cardenasau2018@licifug.ugto.mx

Resumen—En este reporte presentamos los programas realizados las sesiones. Se trabajó en lenguaje C y Python, dichos programas podían tener como entrada uno o varios archivos que contenían datos de longitud de onda e intensidad, pero de todos los valores solo nos centramos en analizar dentro de un rango específico, dentro de este rango se obtenían máximos locales a los que denominamos picos para posteriormente obtener el área debajo de estos picos.

I. INTRODUCCIÓN

El análisis vibracional de los materiales poliméricos a través de las espectroscopias Raman es un método experimental apropiado para obtener información sobre parámetros estructurales de los mismos. Así, además de poderse analizar las especies químicas presentes en el compuesto, es posible obtener entre otros, datos sobre el estado de orden de los polímeros (orientación de cadenas, cristalinidad, fases cristalinas, etc.). La espectroscopia vibracional también permite seguir la evolución de los anteriores parámetros, al realizar diferentes procesos sobre los materiales, como por ejemplo, aplicación de tensiones, tratamientos térmicos, degradaciones, entre otros. [1]

II. OBJETIVOS

Programar un código que realice lo siguiente:

- Realice tratamientos matemáticos a espectros.
- Calcule los picos máximos de dicho espectros.
- Realice un ajuste polinomial
- Calcule el área asociada a dichos picos.
- Muestre con gráficas lo obtenido.
- En una tabla muestre la información anterior.

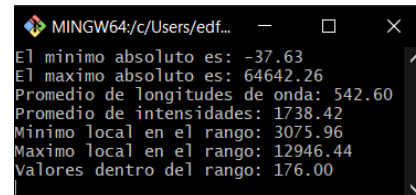
III. METODOLOGÍA

III-A. Código en C

III-A1. Programa 01: Datos estadísticos: En el primer código realizado se calcularon:

- Promedio de Lambdas e intensidades
- Mínimo y máximo absolutos de intensidades.
- Mínimo y máximo dentro de un rango.
- Cantidad de valores en ese rango.

Lo calculado se puede observar en la Figura 1



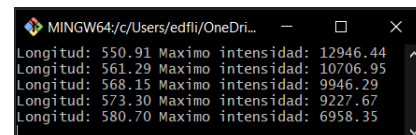
```

MINGW64/c/Users/edf...
El mínimo absoluto es: -37.63
El máximo absoluto es: 64642.26
Promedio de longitudes de onda: 542.60
Promedio de intensidades: 1738.42
Mínimo local en el rango: 3075.96
Máximo local en el rango: 12946.44
Valores dentro del rango: 176.00
  
```

Figura 1. Impresión a pantalla del Programa 01 en C

III-A2. Programa 02: Máximos dentro de un rango:

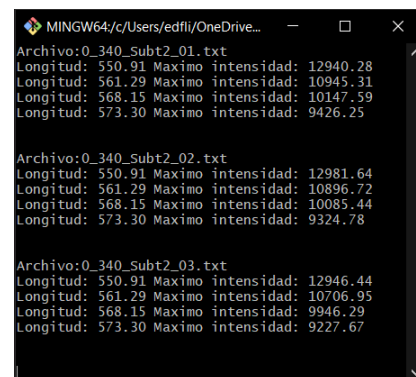
Para esta versión lo que se hizo fue proporcionaron 5 picos de interés, los cuales fueron: **[550.910, 561.289, 568.153, 573.301, 580.701]nm**, además de un $\delta\lambda = 1,0\text{nm}$. Con esto lo que se hizo fue posicionarnos en esas longitudes de onda proporcionadas y, analizar dentro de una vecindad $\delta\lambda$ cuales eran los picos máximos en intensidad. Esto se aprecia en la figura 2.



```

MINGW64/c/Users/edf/OneDrive...
Longitud: 550.91 Máximo intensidad: 12946.44
Longitud: 561.29 Máximo intensidad: 10706.95
Longitud: 568.15 Máximo intensidad: 9946.29
Longitud: 573.30 Máximo intensidad: 9227.67
Longitud: 580.70 Máximo intensidad: 6958.35
  
```

Figura 2. Impresión a pantalla del Programa 02 en C



```

MINGW64/c/Users/edf/OneDrive...
Archivo:0_340_Subt2_01.txt
Longitud: 550.91 Máximo intensidad: 12940.28
Longitud: 561.29 Máximo intensidad: 10945.31
Longitud: 568.15 Máximo intensidad: 10147.59
Longitud: 573.30 Máximo intensidad: 9426.25

Archivo:0_340_Subt2_02.txt
Longitud: 550.91 Máximo intensidad: 12981.64
Longitud: 561.29 Máximo intensidad: 10896.72
Longitud: 568.15 Máximo intensidad: 10085.44
Longitud: 573.30 Máximo intensidad: 9324.78

Archivo:0_340_Subt2_03.txt
Longitud: 550.91 Máximo intensidad: 12946.44
Longitud: 561.29 Máximo intensidad: 10706.95
Longitud: 568.15 Máximo intensidad: 9946.29
Longitud: 573.30 Máximo intensidad: 9227.67
  
```

Figura 3. Impresión a pantalla del Programa 03 en C

III-A3. Programa 03: Extendido para n-archivos: La siguiente versión del programa fue un cambio minúsculo en el

código, pero bastante importante la hora de analizar espectros. Lo que se hizo fue que ahora el código era capaz de ejecutar lo mismo que la versión anterior pero extendido para n-archivos. En este caso las pruebas solo se realizaron con 3 archivos, por esta razón solo aparece esa cantidad en la figura 3, pero está extendido hasta n-archivos.

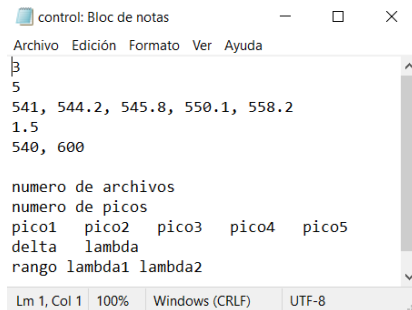


Figura 4. Archivo de control

III-A4. Programa 04: Archivo de control: Esta actualización fue pensada en el usuario, ya que él sólo tenía que crear un archivo .txt y escribir los siguientes datos:

- Cant. de archivos
- Cant. de picos
- Los picos a inspeccionar
- Vecindad de longitud de onda $\delta\lambda$
- rango de longitud $[\lambda_1, \lambda_2]$

Con estos, que también se observan en la figura 4, el programa se encargó de leerlos datos y automáticamente realizar los cálculos necesarios, para posteriormente desplegarlos en pantalla. Esto se puede apreciar en la figura 5

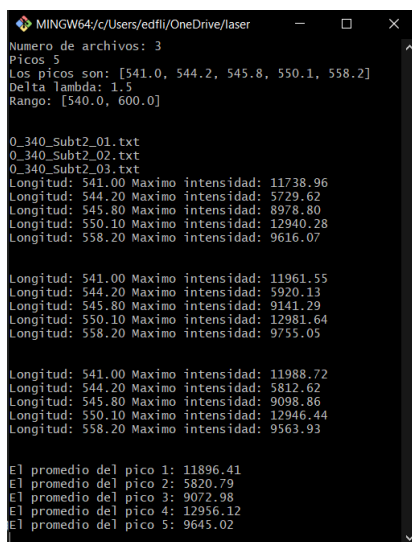


Figura 5. Impresión a pantalla del Programa 04 en C

Figura 6.

III-A5. Programa 05: Código ASCII (extra)*:

III-A6. Programa 06: Inclusión de GNU: Esta fue la última actualización antes de abandonar el código en C y cambiarnos a Python. También fue lo último que se realizó antes de que comenzara la cuarentena. Lo que se logró con esta actualización fue poder ver los espectros de manera gráfica.

Primero, que nada, para que funcione este código, el *Progra-*

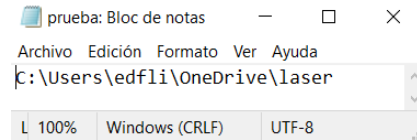


Figura 7. Archivo prueba.txt

ma *GNUplot* debe estar instalado en una carpeta específica, de lo contrario nuestro programa no podrá proceder. Lo que necesita el usuario es, en un archivo .txt colocar la dirección de la carpeta que contenga los datos de los espectros que se desea analizar, un ejemplo de esto se observa en la figura 7. Adicional a esto, creamos un archivo de extensión .gnu con

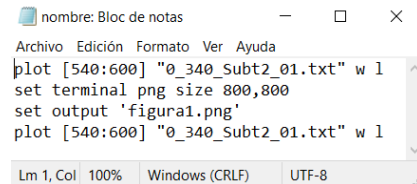


Figura 8. Archivo nombre.gnu

las especificaciones necesarias para graficar el espectro que deseamos, esas se pueden observar en la figura 8. Con estos 2 archivos el programa se encarga de concatenar un único comando que es el que va a imprimir en la terminal para que corra lo necesario, dicho comando concatenado es:

Ya que dicho comando fue impreso en la terminal automáti-

`c:/Gnuplot/gnuplot/bin/gnuplot.exe -p C:/Users/edfli/OneDrive/laser/nombre.gnu`

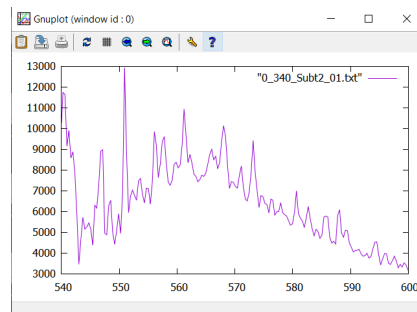


Figura 9. Grafico automático en gnuplot.

camente al correr el programa, este va a arrancar a *GNUplot*, ploteando el espectro esperado con las especificaciones proporcionados. Y además, se agregó la función de que ese gráfico se guarda automáticamente con extensión .png (ver figura 8) en la carpeta especificada por el usuario (ver figura 7).

III-B. Código en Python

III-B1. Versión 01: Ajustes polinómicos: Para este punto ya habíamos mudado el código anterior que teníamos en c para python. Lo que agregamos fue que el programa pudiera realizar un ajuste polinomial de grado 9 a un rango de datos que en este caso fue de **540 - 600**, la manera en la que se realizó el ajuste fue con una función de la biblioteca de numpy la cual se llama con el comando `poly1d` y que recibe como parámetros los valores en X, Y y el grado del ajuste. También es importante mencionar que de este punto en adelante se dejó de usar GNU plot para graficar y comenzamos a usar la biblioteca `matplotlib.pyplot` la cual sirve para poder graficar en python. Con lo anterior ya calculado, se procedió a buscar los

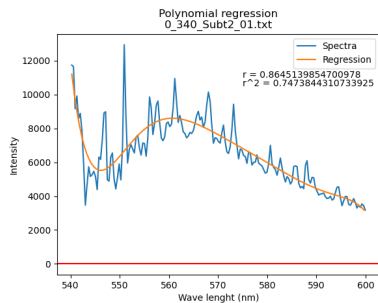


Figura 10. Gráfico del espectro con regresión polinomial.

valles del espectro, esto con el fin de ahora aplicar el ajuste polinomial a dichos puntos del espectro. Esto se observa en la figura 11

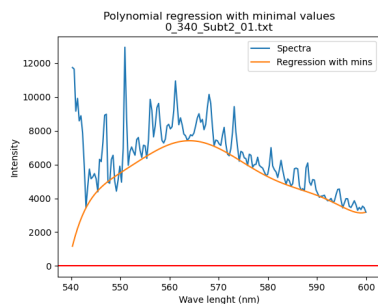


Figura 11. Gráfico del espectro con regresión polinomial en mínimos.

III-B2. Versión 02: Espectro aplanado: Para obtener un espectro aplanado (sin fluorescencia) lo que se hizo fue tomar el espectro original y restarle el ajuste en mínimos, claramente se ve en la figura 11 lo que sucedería si estos dos se sustraen. El resultado se aprecia en la figura 12. Pero podemos notar que hay un pequeño pico cercano a los 550 nm que ahora sobresale por debajo del cero. Para arreglar este desperfecto lo que se hizo fue calcular el mínimo absoluto de este espectro aplanado. Con esto calculado, lo que se hizo fue sumar este mínimo a este mismo espectro para obtener uno nuevo, de igual forma pero ahora sin ningún punto debajo del cero. Este nuevo espectro corregido ya es el óptimo, y se aprecia en la figura 13.

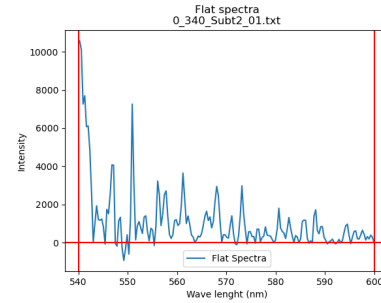


Figura 12. Espectro Raman aplanado

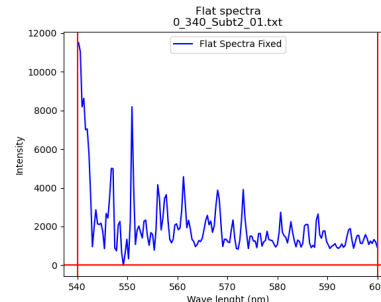


Figura 13. Espectro Raman procesado

III-B3. Versión 03: Intensidades máximas: Para realizar este cálculo se migró el código en C que se utilizó en la sección III-A2, con el cual dadas ciertas longitudes de onda $\lambda[nm]$, se buscaron los máximos de intensidad dentro de una vecindad $\delta\lambda[nm]$, dichas intensidades máximas se señalaron en el espectro y se colocó en su cresta el valor de cada una. Esto se puede apreciar en la figura 14.

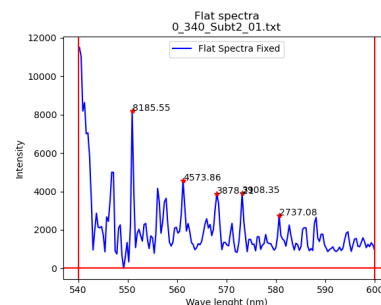


Figura 14. Espectro Raman con intensidades máximas.

III-B4. Versión 04: Cálculo de áreas: Lo último que pudimos realizar durante todo el tiempo que estuvimos trabajando en este proyecto fue calcular el área que había debajo de los picos que anteriormente ya habíamos analizado, para calcular esta área utilizamos un método de integración numérica conocido como trapecio, el cual consistía en sumar todos los valores en Y que formaban el pico, a esa suma dividirlo entre el número de valores que fueron sumados y por último multiplicarlo por la resta de la longitud en la cual termina e inicia el pico. El área nos servía al momento de comparar si un pico es mayor que otro pero tiene menor área

eso nos dice que el pico que es menor tiene una mayor longitud de onda. La manera de implementar esta lógica en el programa era que nos teníamos que posicionar en el pico (máximo) y de ahí explorar los valores hacia la izquierda y hacia la derecha y ver si se cumplía que el siguiente valor era menor a en el que estábamos si era así que fuera avanzando hasta que se topara con que otra vez la gráfica comenzaba a subir, esto nos daba los valores en y que teníamos que sumar y también las posiciones iniciales y finales de la longitud.

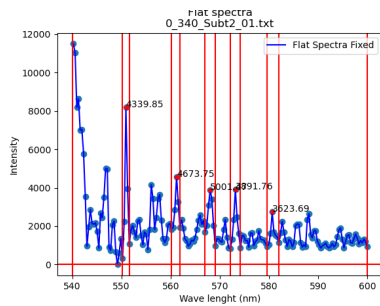


Figura 15. Espectro Raman con delimitaciones de área en picos máximos.

III-B5. Versión 05: Comparación de espectros: En esta sección se agregó una gráfica con todo el proceso y los cambios realizados en el espectro con los tratamientos matemáticos realizados. Esto es únicamente para tener una ayuda visual de todo con lo que se fue trabajando, dicho proceso se observa en la figura 16.

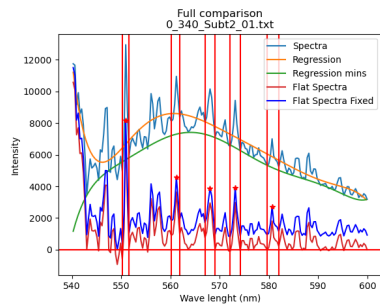


Figura 16. Empalmación gráfica de espectro Raman.

IV. RESULTADOS

IV-A. Interfaz Gráfica

Implementar la interfaz gráfica se podría decir que es como hacer la evolución del archivo de control, pues notamos que python ofrece mucha facilidad para poder trabajar con esta herramienta. Lo que hace es ofrecer mayor comodidad al usuario al momento de poder ingresar los datos para que el programa los analice. Los valores a ingresar son casi los mismos que en el archivo de control:

- Cantidad de archivos
- Vecindad de longitud de onda $\delta\lambda$
- Los picos a inspeccionar
- Rango de longitud $[\lambda_1, \lambda_2]$

Con un solo pequeño cambio el cual era que ya no era necesario decir cuantos picos se iban a inspeccionar, pues con los picos que ingresaba el usuario entraban en un vector que con solo usar la función len nos daba la cantidad de picos. Algo a destacar al momento de ingresar tanto el rango de longitud como los picos a inspeccionar era que es necesario que estén separados por una coma y un espacio pues el programa detecta los valores solo si se usa esa sintaxis. Esto se aprecia en la figura 16.

A pesar de que se trabajó en conjunto y con las especificacio-

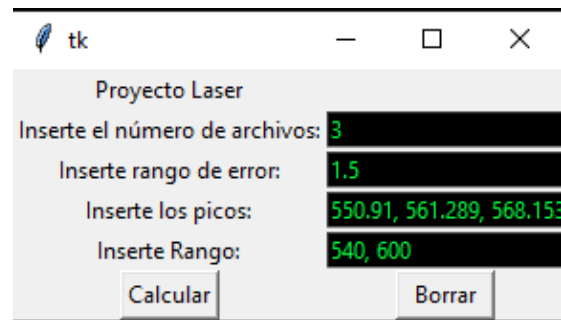


Figura 17. Interfaz para modificar valores de entrada.

nes requeridas, proporcionadas por los Dres. del Laboratorio de fotoacústica, cada quien realizó su propio código. Por esta razón podemos ver que hay una segunda interfaz gráfica propia de otro integrante del equipo. Esta se puede apreciar en la figura 18 Como se puede observar ambas

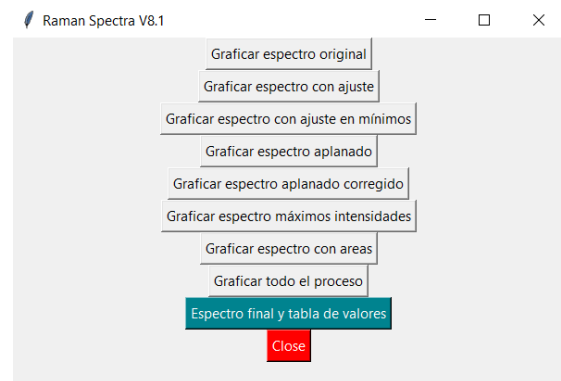


Figura 18. Interfaz para mostrar los procesos gráficos.

interfaces son distintas, mientras que una permite modificar los parámetros de entrada, la otra gráfica los procesos intermedios y la tabla de resultados. Pero lo importante es que ambas muestran los resultados finales que son lo más importante. Cabe mencionar que estas se podrían fusionar para ser mejores entre sí y más optimizadas para el usuario.

IV-B. Área de picos de interés

Esta última gráfica muestra lo que estábamos buscando durante todo este proceso:

1. Espectro aplanado
2. Área debajo de picos de interés.

Con estos datos obtenidos, se obtuvo una gráfica con un espectro limpio y obtenido con los tratamientos matemáticos mencionados, en el cual, en la cresta de dichos picos, se colocaron etiquetas con el asociada a cada uno de estos.

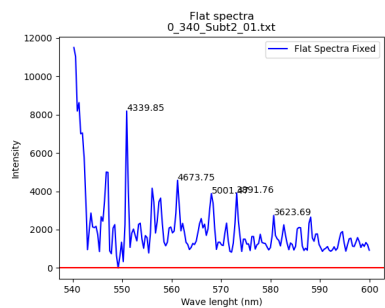


Figura 19. Espectro Raman: Etiquetas de área bajo picos máximos.

IV-C. Tabla de resultados.

Esta tabla de resultados fue generada automáticamente al dar clic sobre el botón de la interfaz, esto fue algo que se agregó de último momento al tener conocimiento del módulo de Python que nos permitía realizar dicha tarea. Si bien el código ya estaba programa para que mostrara dichos datos en la terminal, el despliegue de estos datos en una verdadera tabla supone una ventaja visual a la hora de la interpretación.

Longitud de onda	Intensidad	Area
550.91	8185.55125	4339.849774547025
561.194	4573.860546874999	4673.7501669596695
568.03	3878.214765625	5001.47353960374
573.146	3908.351328125	3891.7583504742524
580.632	2737.0803906249994	3623.6920449463173

Figura 20. Espectro Raman: Tabla de resultados.

REFERENCIAS

[1] PASTOR, J. M., JAWHARI, T., & MERINO, J. C. (2003). CARACTERIZACIÓN DE MATERIALES POLÍMEROS MEDIANTE LA UTILIZACIÓN DE LA MICRO ESPECTROSCOPIA VIBRACIONAL (RAMAN, FTIR Y FT-RAMAN). REVISTA IBEROAMERICANA DE POLÍMEROS, 4(3), 1-43.

V. ANEXOS

En esta sección una serie de gráficos provenientes del **Programa 04: Archivo de control** que se encuentra en la sección III-A4. Donde estos gráficos se obtienen al graficar cada uno de los 3 archivos de datos con los que estuvimos trabajando.

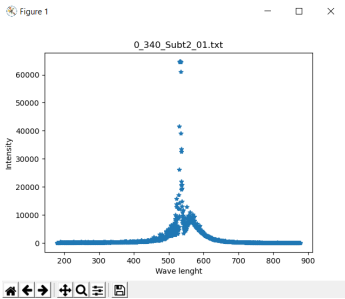


Figura 21. Archivo 0_340_Subt2_01.txt graficado.

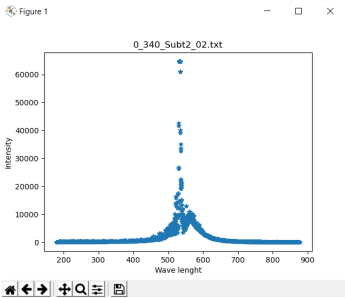


Figura 22. Archivo 0_340_Subt2_02.txt graficado.

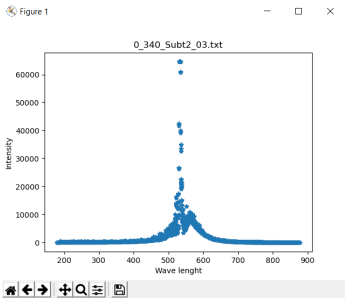


Figura 23. Archivo 0_340_Subt2_03.txt graficado.