

**Contenido**

[**Introducción** 3](#_Toc207638156)

[**Como puedo usar el programa** 4](#_Toc207638157)

[Paso 1: 4](#_Toc207638158)

[Paso 2: 5](#_Toc207638159)

[Paso 3: 6](#_Toc207638160)

[Paso 4: 7](#_Toc207638161)

[Paso 5: 8](#_Toc207638162)

[Recomendación 9](#_Toc207638163)

[Nota 10](#_Toc207638164)

[**Como funcionan las ventanas** 10](#_Toc207638165)

[**Inicio** 11](#_Toc207638166)

[Recomendación 18](#_Toc207638167)

[**Grafica** 19](#_Toc207638168)

[Explicación de cómo funciona esta ventana 27](#_Toc207638169)

[Nota 28](#_Toc207638170)

[Nota 2 31](#_Toc207638171)

[Ver la imagen 11.5: filtro 32](#_Toc207638172)

[Ver la imagen 11.6: tabla 34](#_Toc207638173)

[Ver imagen 11.7: edición del lienzo 35](#_Toc207638174)

[Ver la imagen 11.8 control de calidad 38](#_Toc207638175)

[**Análisis del archivo** 41](#_Toc207638176)

[**Archivo Toga** 44](#_Toc207638177)

[**Resumen General** 45](#_Toc207638178)

[Detalles 45](#_Toc207638179)

[Nota 49](#_Toc207638180)

[**Agregar nuevo filtro** 51](#_Toc207638181)

[**Mas información:** 59](#_Toc207638182)

[**Conclusión** 60](#_Toc207638183)

[**Referencias** 61](#_Toc207638184)

# **Introducción**

El día 3 de julio comencé estudiando los scripts y fui verificando que la mayoría de los archivos TOGA, son procesados para generar varias gráficas y realizar otros análisis, como, por ejemplo: estructura correcta de los archivos toga, que contemple el numero correcto de días y el número de líneas(filas), que el DataFrame no este vacío, entre otros. Algunos scripts llegaron a ser muy redundantes y como el objetivo era que cada script, tenía prioridades específicas, era muy independientes y no existía uno que pudiera englobarlos. Así nace la necesidad de tener un programa genérico en Python, un interfaz donde se puedan realizar análisis más autónomos (sin intervención constante del usuario), sin necesidad de ser tan manuales (requiere que el usuario ejecute cada paso o tome decisiones durante el uso).

Otras de las necesidades fueron, que el usuario no tenga que instalar nada para poder trabajar con dicho programa, que sea muy accesible. Esos fueron uno de los retos, al inicio fue un prototipo que se ejecutaba en un IDE (Visual Studio) pero esto no cumplía con los estándares propuestos, así que optamos por otra alternativa más completo y más accesible, donde cualquier persona puede acceder en internet y usarlo sin ningún problema y sin ningún conocimiento previo, esa solución fue el uso de notebooks, ya que es un entorno interactivo, donde se puede combinar código, texto, imágenes, gráficos y ecuaciones en un mismo documento. Ahora era encontrar donde ejecutar estos notebooks y la respuesta simple fue en Google Colab, ya que es un servicio gratuito de Google que permite crear y ejecutar Notebooks en la nube, existen otras alternativas, pero esta me resulto más familiar y además dicha plataforma es muy conocida por estudiantes y profesionales.

A continuación, presentare como usar el programa correctamente, además de realizar recomendaciones para futuros estudiantes que les interese seguir mejorando esta experiencia de estudio de los datos, ya que este programa esta echa con la intención de mejorar la comprensión sobre el comportamiento de las mareas, su uso permite analizar datos de manera interactiva, facilitando así la toma de decisiones en estudios costeros y proyectos marítimos.

# **Como puedo usar el programa**

Si esta leyendo este manual es porque ya lograste acceder al repositorio. <https://github.com/Octavio341/MareograficoGoogleColab.git>, ya que aquí es donde se encuentra el programa(notebook), así que ahora procedo a explicarte donde ubicar el programa correcto, una vez aquí:

|  |
| --- |
|  |
| Imagen 1: repositorio. |

## **Paso 1:**

Como paso 1 tendrás que ir a la carpeta versiones

|  |  |
| --- | --- |
|  |  |
| Imagen 2: aquí veras la carpeta. | **Imagen 3:** aquí veras todas las versiones disponibles. |

En la imagen 3 se ven las diferentes versiones del programa y la versión que ejecutaremos es la última, que es la “versión 4” ya que las demás no están completos como la versión 4, ya esta versión tiene todas las herramientas más completas y puede ser considerado más autónomo y genérico.

|  |  |
| --- | --- |
|  |  |
| Imagen 3.1: selección de version correcto. | **Imagen 3.2:** descargar el archivo. |

## **Paso 2:**

Una vez descargado, nos vamos al buscador y escribimos:

|  |
| --- |
|  |
| Imagen 4: busqueda. |

Una vez adentro lo que se tiene que hacer es lo siguiente:

|  |
| --- |
|  |

## **Paso 3:**

|  |  |
| --- | --- |
|  |  |
| Imagen 5: ir a Open Colab. | **Imagen 5.1**: ir a subir archivo, para seleccionar el notebook descargado. |

El siguiente paso es ir como se muestra en la imagen 5 y por último nos aparecerá el siguiente interfaz donde subiremos el notebook, nosotros seleccionamos subir archivo, para seleccionar nuestro notebook descargado.

|  |  |
| --- | --- |
|  |  |
| Imagen 5.2: selección del notebook. | **Imagen 6:** muy importante haber iniciado sesión. |

El siguiente paso es como se muestra en la imagen 5.2, pero antes de esto es necesario haber iniciado sesión con tu cuenta de Google, para pasar al siguiente paso.

## **Paso 4:**

Una vez que ya se hizo todo, veremos el contenido del notebook, donde se contemplaran distintas celdas, como objetivo, cada uno tiene una ejecución independiente, pero necesaria para un mejor funcionamiento.

|  |
| --- |
|  |
|  |
| Imagen 7: sobre las celdas |

Ahora bien, una vez dentro podemos ver como todas las celdas disponibles en este notebook. Explicación breve: “la celda numero 1; contiene información sobre el notebook”, “Celda numero 2; contiene el repositorio, aquí clonamos nuestro repositorio para acceder a los módulos y scripts (necesarios para el análisis del archivo TOGA)”, “Celda número 3; con tiene el código principal del interfaz”, “Celda numero 4; su ejecución no es necesaria, pero sirve para limpiar /content (es la carpeta principal del entorno de trabajo), aquí se borran todos los archivos seleccionados, es útil cuando quieres introducir archivos nuevos”.

Ahora iniciemos con la ejecución de cada celda:

## **Paso 5:**

|  |
| --- |
|  |
| Imagen 8: da clic aquí y veras como se empieza a ejecutar. |

Si das clic en la carpetita, como se muestra en la imagen número 8.1, podrás ver la carpeta del repositorio:

|  |  |
| --- | --- |
|  |  |
| Imagen 8.1: el repositorio | |

Ahora pasemos a la tercera celda:

Ver imagen numero 8.2 al ejecutar la celda 3 se verá lo siguiente:

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |
| Imagen 8.2: montado drive y serie de permisos | |

Y ahora bien una vez logrado confirmar todo eso, ahora si puedes ver el interfaz, y también puedes ver la carpeta drive en el entorno de trabajo, todo esto se ilustra en la imagen número 8.2.1.

|  |
| --- |
|  |
| Imagen 8.2.1: interfaz principal |

Ahora bien, la celda numero 4 no es necesario ejecutarlo de inmediato, por que no tenemos archivos TOGA seleccionados, aún sigue vacío.

¡Ahora si estas listo para usar el programa!

### **Recomendación**

Para ver todo el interfaz completo primero tienes queir a las siguientes opciones, como se muestre en la imagen 9 y 9.1:

|  |  |
| --- | --- |
|  |  |
| Imagen 9 – Paso 1: Hacer clic aquí. | **Imagen 9.1** – Paso 2: Hacer clic aquí. |

## **Nota**

Si no activaste la primera celda correctamente puede que te salga este error, ya que necesita la carpeta del repositorio para usar el programa llamado “sincronizado” que es el script que se encarga de realizar resúmenes generales y también necesita los scripts número 6 y 11, más adelante con la ventana de “Análisis de archivo” (ver **imagen 9.2**).

|  |
| --- |
|  |
| Imagen 9.2: posible error al ejecutar la celda donde se encuentra el programa. |

# **Como funcionan las ventanas**

Una vez dentro, como podrás darte cuenta el interfaz cuenta, el interfaz cuenta con 5 ventanas, que son:

* **Inicio:** donde se suben los archivos, ya sea por drive o por medio de pc.
* **Grafica:** donde se procesan los archivos subidos y aquí es el punto principal, la cual los archivos pueden ser enviados a dos modos, grafica (análisis detallado) y resumen general (análisis general de muchos archivos toga), pero aquí se puede ver la gráfica interactiva y acceder a las herramientas.
* **Análisis del archivo:** aquí es donde se ejecutan 2 script que analizan a profundidad la estructura del archivo toga.
* **Archivo Toga:** aquí podemos ver la estructura del archivo TOGA, como un DataFrame (es una **estructura de datos en forma de tabla**).
* **Resumen General:** aquí se puede ver la gráfica, análisis simplificado y también existe la posibilidad de cambiar el formato del archivo Toga, pero la única diferencia con todas las demás es que aquí puedes ver muchas graficas con su análisis corto, puedes ver el contenido de más de 40 archivos Toga, es útil cuando no quieres ir analizando uno por uno y ver su gráfica, si no, cuando quieres ver todo y de ahí seleccionar cual quieres ver con más detalle y aplicar filtros a la gráfica.

Una vez que ya te resumí lo que hace cada ventana, procedemos a ver cómo funciona cada una.



## **Inicio**

Ahora puedes ver dos botones, pero al dar clic ocurre lo siguiente:

|  |
| --- |
|  |
| Imagen 10: interacción con el botón “Desde PC” |

Aquí te salen dos botones más, si das clic en “Elegir archivo” y “cancelar upload” si das clic a este ultimo puedes salir de esta opción, pero sí en cambio presionas “Elegir archivo”, lo que pasa es lo siguiente:

|  |
| --- |
|  |
| Imagen 10.1: selección de archivos |

**Imagen 10.1:** aquí podemos ver como se puede seleccionar los archivos toga y de esa forma podemos agregarlos al interfaz.

|  |
| --- |
|  |
| Imagen 10.2: vista de los archivos seleccionados. |

Y podemos seleccionar mas como se muestra en la **imagen número 10.3.**

|  |
| --- |
|  |
| Imagen 10.3: selección de varios archivos. |

Estos archivos nos servirán para poder seguir con el siguiente paso que seria el graficado y análisis del contenido del archivo TOGA. Ahora bien, pasemos con el funcionamiento del botón de drive; aquí su función es idéntica al del pc, por eso fue que al inicio nos pedía completar ciertos permisos, solo fueron esos detalles, que sirvieron para montar una carpeta drive, y solo será una vez ya que si lo volvemos a ejecutar ya no saldrá de nuevo.

|  |
| --- |
|  |
| Imagen 10.4: interfaz de selección del archivo desde el drive. |

**Imagen 10.4**, no muestra en el widget lo nombre de todos los archivos y carpetas y aquí podemos entrar y seleccionar un archivo, también vemos que para movernos dentro de este interfaz tenemos las siguientes opciones, que son “**ir atrás:** esto nos hace retroceder de carpeta”, luego tenemos el botón de “Entrar: esto nos permite seleccionar alguna carpeta, pero en archivos esto no funciona, al contrario, nos saldrá una advertencia que no dirá que no seleccionamos nada”.

|  |
| --- |
|  |
| Imagen 10.5: selección de archivos toga. |
|  |
| Imagen 10.6: selección de archivos toga. |

**Imagen 10.5:** aquí podemos ver que la dinámica es la misma podemos seleccionar varios archivos, y cuando los tengamos todos seleccionados con el botón “Seleccionar” procederá a cargar todos los archivos toga, como se muestra en la **Imagen 10.6.**

Ahora bien, una vez que tengamos todos los archivos seleccionados, si queremos volver a seleccionar un archivo que ya se encuentre cargado aquí en el interfaz, el programa podrá identificar que el archivo a sido seleccionado de nuevo y nos enviara un mensaje confirmando que el archivo a sido seleccionado de nuevo (Imagen 10.7).

De esta forma evitamos que los archivos puedan reescribirse y tenga que ser repetitivo procesar ciertos archivos de nuevo.

|  |
| --- |
|  |
| Imagen 10.7: identificador de archivos duplicados. |

Ahora bien, si queremos eliminar todos estos archivos para agregar nuevos, por que puede que tengamos que ejecutar de nuevo este programa y se pierdan todos estos archivos y tengas que seleccionar todos de nuevo. Lo que diré a continuación solo funciona para archivos cargados desde pc, porque estos archivos se mantienen en el entorno de trabajo de **Google Colab**, como se puede ver en la imagen 10.8, en drive no se guardan y al actualizar todo, se elimina los archivos cargados desde el drive y solo conservaremos los de pc. Así que es recomendable cargar los archivos toga desde Pc para poder mantenerlos.

|  |
| --- |
|  |
| Imagen 10.8: entorno de trabajo de Google colab, donde se puede ver la carpeta de Google colab, drive, repositorio y archivos cargados desde pc. |

Pero si no quieres mantener esos archivos cargados desde pc o cargar nuevo y quieres limpiarlos, la alternativa es ejecutar la celda numero 4, en otras palabras, es la que se encuentra abajo del interfaz. (**ver imagen 10.9**)

|  |  |
| --- | --- |
|  |  |
| Imagen 10.9: celda número 4, para eliminar archivos del entorno de trabajo de Google colab. | |

Si lo ejecutamos este procederá a eliminar todos los archivos que se encuentre aquí y ahora si podemos empezar a seleccionar nuevos archivos.

Pero si no quieres eliminar ningún archivo, la siguiente posibilidad es que puedas usar ese entorno de conservar los archivos para poder cargarlos desde el interfaz sin que vuelvas a entrar con el botón de pc y seleccionar algún archivo desde ahí, si no que entrando con el botón de “Desde el drive” puedas cargar los archivos que anteriormente seleccionaste desde pc. (ver **imagen** **10.10**).

|  |
| --- |
|  |
| Imagen 10.10: archivos desde el entorno de trabajo, se pueden ejecutar más fácilmente. |

En la **imagen 10.11**, podemos observar como es el seleccionado y como sigue la misma dinámica al volver a seleccionar de nuevo esos archivos, aparecerá el mensaje de archivos duplicados.

|  |
| --- |
|  |
|  |
|  |
| Imagen 10.11: trancicion de seleccionado de archivos desde content (entorno de trabajo de google colab). |

### **Recomendación**

Para trabajar de forma más dinámica y limpia, es recomendable cargar archivos desde pc, porque eso evita perder los archivos seleccionados y su duración perdura hasta que decidas cerrar la sesión por completo del notebook, mas adelante explicare otro beneficio que tiene al seleccionar archivos desde pc, para saberlo ve en el siguiente apartado, donde hablo sobre como analizar muchos archivos de forma general en la ventana **número 5** **(Resumen general).**

## **Grafica**

Aquí podemos ver la gráfica de los archivos toga y aplicar filtros a la gráfica y de esa forma podemos conocer como es el comportamiento de los datos del nivel mar.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
| Imagen 11: varias graficas alicando distintos filtros, para ofrecer una grafica manipulable. |

**Imagen 11.1:** Este es punto de partida para proceder a analizar los archivos, aquí tenemos la opción de ver una gráfica con la posibilidad de manipularla y otra para realizar un diagnostico de su estructura y por último cargar muchos archivos para mandarlo a resumir cada uno, y ver su contenido.

Aquí se tenemos un panel desplegable, que sirve para mostrarla y ocultarla de esa forma podemos mandar a procesar y luego nos concentramos en la gráfica y sus herramientas.

|  |
| --- |
|  |
| Imagen 11.1: interfaz general de la ventana grafica. |

Ahora conozcamos como funciona este interfaz de forma más detallada.

|  |
| --- |
|  |
| Imagen 11.2: si presionamos este panel puede desplegar opciones para procesar archivos .dat. |
|  |
| Imagen 11.3: al presionar se muestra el panel para procesar archivos .dat. |

Como podemos observar en la **Imagen 11.2** y **11.3** sobre como funciona el panel, este panel tiene tres apartados, **la primera** estan los modos y la **segunda** es donde nos dice que archivos esta listo para aplicar un modo y que empiece a analizar(siempre y cuando se activen ), **la tercera** es donde esta el contenedor de archivos cargados y abajo se encuetra el boton de procesar ese archivo seleccionado.

|  |
| --- |
|  |
| Imagen 11.4: esta es la forma indicada para trabajar con el interfaz, ya que podemos ver la grafica y los filtros de igual forma se observan las ventanas. |

Vemos una forma indicada para empezar a utilizar el interfaz (ver **imagen 11.4**), esto puede ser posible cuando ocultamos el panel, es esencial cuando queremos concentrarnos completamente en la observación de la gráfica.

|  |  |
| --- | --- |
|  |  |
| Imagen 11.5: filtros esenciales ya que cada uno tiene un gran impacto en su demostración, aquí se pueden activar y desactivar los puntos, la línea, mostrar etiquetas en vez de puntos, cambiar el color a la línea de la gráfica. | **Imagen 11.6:** aquí se cuenta con un selector de tablas donde muestra el contenido del archivo, cada fila esta vinculada con la gráfica, si seleccionamos una fila en la grafica podemos ver donde se encuentra ese registro. |
|  |  |
| Imagen 11.7: filtros para cambiar el aspecto a la grafica. | **Imagen 11.8:** filtros para el control de calidad, donde se busca analizar la gráfica con estos filtros. |

|  |
| --- |
|  |
|  |
| Imagen 11.9.1: herramientas para hacer que la grafica sea mas interactiva. |
|  |
| Imagen 11.9.2: modos de movimiento de la gráfica, de forma horizontal y vertical. |
|  |
| Imagen 11.9.3: distintas formas de selección con forma de lazo. |
|  |
| Imagen 11.9.4: distintas formas para realizar zoom, con forma cuadrada. |
|  |
| Imagen 11.9.5: distintas formas de selección con forma de cuadro. |
|  |
| Imagen 11.9.6: distintas formas de selección de los puntos. |
|  |
| Imagen 11.9.7: Modos de captura de la gráfica, “opción 1: mostrar la gráfica en una ventana del navegador”, “opción 2: para copiar la gráfica (es como hacer control + C)”, “opción 3: descargar la gráfica y guardarla localmente”. |
|  |
| Imagen 11.9.8: Herramientas con una sola función, “opción 1: es una herramienta predetermina”, “opción 3: es para reiniciar la vista de la gráfica en su forma original”, “opción 4: es para realizar zoom para ampliar”, “opción 5: para minimizar la gráfica”, “opción 6: para ver más información acerca de la librería”, “opción 7: para ver una guía que acompaña al cursor”, “opción 8: es hover message, es donde muestra información acerca de la gráfica al pasar el cursor”. |

Desde la **imagen 11.9.1** y **11.9.8** todas estas herramientas nos sirven para interactuar con la gráfica, para acceder a ella solo tenemos que mantener presionado encima y nos aparecerán estas opciones o simplemente dar clic derecho y aparecerán esas opciones.

En las **imágenes 11.5** hasta la **11.8** vemos de forma general sobre los Tabs, estos nos permiten aplicar los filtros a la gráfica.

### **Explicación de cómo funciona esta ventana**

Ahora hablemos un poco de como funciona este interfaz con respecto a la ventana “Grafica”.

Cuando carguemos todos los archivos desde el inicio, automáticamente vemos esos archivos en el apartado de procesar archivos (ver imagen 9.9), donde veremos todos los archivos subidos y aquí elegimos que archivos queremos analizar, una vez seleccionado procedemos a ver los archivos seleccionados en el apartado de archivos procesados (ver imagen 9.9.2).

|  |
| --- |
|  |
|  |
|  |
| Imagen 9.9: contenedor donde podemos cargar archivos. |

Luego de eso procedemos a ver dos casillas (ver imagen 9.9.1), en el apartado de modos de análisis, si es la primera vez, esto es lo que pasara:

Cuando elijamos archivos a procesar y al mostrarse todos esos archivos en el apartado de “archivos a procesar”, automáticamente se activara la casilla “Analizar y graficar”, por que pasa esto, por que el programa detecta que es nuestra primera selección y procede a activar esta casilla indicando que de esa forma se puede ver la grafica al seleccionar esas casillas, es como un tipo ayuda al usuario, a partir de aquí si queremos seguir seleccionado mas archivos 1 o más nos aparecerá automáticamente la grafica sin hacer nada.

Al desactivar esta casilla y volvemos a mandar estos archivos a procesar la casilla nunca más se activará solo en esta sesión, solo ocurrirá cuando ejecutemos el programa de nuevo.

### **Nota**

Esta acción se hace solo para recargar este analizador, ya que ocurre el error de que el programa se congela si activamos la casilla “Resumen general”, lo que provoca esta casilla es que no puede mostrar la gráfica, solo cuando se activa primero la casilla “resumen general”. Esto ocurrirá solo 1 vez en todas las sesiones y más nunca esperemos que se active solita, y la única forma de evitarlo es que una vez ejecutado el programa entremos a esta ventana (Graficar), y activemos primero la casilla Analizar y graficar, de esta forma se evita el activado automático.

Aunque activemos Resumen general primero, es inevitable el activado de Analizar y graficar (ver imagen 9.9.0), que siempre mostrara la gráfica del primer archivo seleccionado.

|  |  |
| --- | --- |
|  |  |
| Imagen 9.9.0: activado inevitable de la casilla Analizar y graficar. |  |

|  |
| --- |
|  |
| Imagen 9.9.1: contenedor de modos, donde se cuenta con dos tipos de análisis el “resumido general” y “Analizar y graficar” |

|  |
| --- |
|  |
| Imagen 9.9.2: contenedor donde envía un mensaje de confirmación, sobre archivos seleccionados y listos para el análisis. |

|  |
| --- |
|  |
| Imagen 9.10: Activar la casilla “Analizar y graficar”, antes de mandar un archivo a procesar, nos muestra que no hay archivos seleccionados. |

Otros de los casos al activar primero la casilla “analizar y graficar” primero sin que se haya mandado a procesar algún archivo, aparecerá este error, pero tranquilo que esto no es nada, que cuando mandes a procesar algún archivo ese error desaparece (ver imagen 9.11).

|  |
| --- |
|  |
| Imagen 9.11: al seleccionar un archivo. |

Como se puede observar en la imagen 9.11, así se vería la gráfica, luego podemos seguir procesando mas archivos de esta forma. El programa esta diseñado para graficar un solo archivo y usar todos los filtros para un archivo, pero ahí ciertos filtros que pueden cambiar con cierto numero de archivos graficados, en la imagen 9.11.1, se puede observar como tres archivos son graficados y se pueden distinguir por los distintos colores de los puntos, también en el titulo de la grafica se puede ver esta diferencia.

|  |
| --- |
|  |
| Imagen 9.11.1: al seleccionar varios archivos |

### **Nota 2**

Si quieres graficar mas de 3 archivos y verlos en este lienzo ten cuidado la interacción puede verse afectada y los filtros puede ralentizarse.

|  |
| --- |
|  |
| Imagen 9.11.2: activación automática de la casilla “Analizar y graficar” |

Si es la primera vez que ejecutas el programa y mandas a procesar los archivos, puedes ver que como no has seleccionado ninguna casilla, automáticamente se activa la casilla de “Analizar y graficar”, esto solo funciona una vez y solo grafica el primer archivo, por si seleccionaste mas no se graficaran, como ya se explico esto anteriormente, es importante tenerlo en cuenta.

Ahora entenderemos cómo funcionan los filtros.

### **Ver la imagen 11.5: filtro**

Como podemos observar en las imágenes 11.12.1 y 11.12.4, como cada una funciona, cada filtro tiene su propio apartado.

|  |
| --- |
|  |
| Imagen 11.12.1: Activado de línea, se observa el nombre del archivo. |
|  |
| Imagen 11.12.2: Puntos desactivados |
|  |
| Imagen 11.12.3: Activado de etiquetas, estos son los registros del nivel del mar |
|  |
| Imagen 11.12.4: Cambiado de color de la línea del grafo. |

En la imagen 11.12.4 se observan distintos colores que se le puede aplicar a la

Al seleccionar varios archivos, procede un cambio en las casillas, donde se observarán mas casillas y estas corresponden a cada línea (ver imagen 11.12.5).

|  |
| --- |
|  |
| Imagen 11.12.5: la cantidad de filtros corresponden con la cantidad de archivos graficados. |

### **Ver la imagen 11.6: tabla**

El manejo de las tablas también se puede observar por medio de un selector, ese selector nos permite seleccionar alguno, para que a continuación en la gráfica se pueda ver reflejada que punto hemos seleccionado, esta es una mejor forma para ver cómo es la secuencia de los datos donde también, en la columna de incremento se observa cuanto aumento o disminuyo, ese registro con base al anterior (ver imagen 11.12.6).

|  |
| --- |
|  |
| Imagen 11.12.6: tabla con un selector, se observa el selector solo cuando existen más archivos graficados y de esa forma se puede ver su tabla. |

También podemos usar las herramientas (ver imagen 11.9.1) para poder seleccionar esos registros del nivel del mar y poder observar en que punto de la tabla se encuentra ubicados.

Esta tabla se puede sincronizar con la gráfica, así que cada acción se vera afectada en la tabla (ver imagen 11.12.7).

|  |
| --- |
|  |
| Imagen 11.12.7: donde se seleccionan varios puntos con la herramienta lazo. |

#### **Nota**

Esto también sirve para comprobar el numero de registros encontrados en la ventana de “Análisis de archivo”.

|  |  |
| --- | --- |
|  |  |
| Imagen 11.12.8: comparación de resultados, entre script y tabla. | |

### **Ver imagen 11.7: edición del lienzo**

Este filtro es ideal cuando queremos editar el lienzo de la gráfica, como cambiar el nombre a la gráfica, cambiar el tamaño de las escalas de los ejes, o mostrar ciertos ejes y cambiar la posición de las leyendas.

#### **Editar las escalas:**

|  |
| --- |
|  |
|  |
| Imagen 11.13: aquí podemos ver mas escalas con la configuración de las barras y en las casillas se puede cambiar que fecha se quiere observar y, por último, forma para mostrar esos ejes, ya sean verticalmente o horizontalmente. |

#### **Cambiar la posición de las leyendas:**

|  |
| --- |
|  |
| Imagen 11.13.1: apartado de casillas, donde vemos el selector de posiciones de las leyendas y las casillas representan distintas formas mostradas en la gráfica. |

|  |
| --- |
|  |
| Imagen 11.13.2: he aquí un ejemplo de como manipulando las casillas y el selector tenemos este resultado. |

#### **Cambio de nombre**

En las imágenes 11.13.3, 11.13.4 y 11.13.5 se pueden observar como se usan el espacio para editar el nombre a la gráfica.

|  |  |
| --- | --- |
|  |  |
| Imagen 11.13.3: grafica original. | |

|  |  |
| --- | --- |
|  |  |
| Imagen 11.13.4: cambio de nombre | |

Cuando se grafican 4 archivos el nombre se mostrará de la siguiente manera (ver imagen 11.13.5).

|  |
| --- |
|  |
| Imagen 11.13.5: se muestra 4 archivos graficados y se puede ver como el nombre de la grafica tiene un buen formato que de igual forma se puede editar. |

### **Ver la imagen 11.8 control de calidad**

Se implementaron métodos para mostrar filtros que permiten analizar los registros de las listas de datos utilizadas en las gráficas de control de calidad.

En este caso solo se abordó 3 filtros:

1. Picos y valles (ver imagen 11.13.6) aquí se recomienda que se desactive la casilla de puntos para ver este filtro aplicado.
2. StuckData (ver imagen 11.13.7).
3. Valor promedio de todos los registros, donde se sugiere que al activar se ajuste los rangos 1,2 y 3, para abordar todos los datos (ver imagen 11.13.8).

|  |  |
| --- | --- |
|  |  |
| Imagen 11.13.6: resultado al aplicar el filtro de picos y valles, aquí pues como su nombre lo dice representamos con triángulos los puntos mas altos y bajos de la gráfica. | |

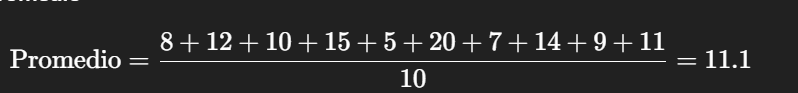
|  |
| --- |
|  |
| Imagen 11.13.7: resultado al aplicar el filtro de stuckData. |

|  |
| --- |
|  |
| Imagen 11.13.8: resultado al aplicar el filtro de los valores promedio. |

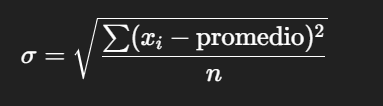
|  |  |
| --- | --- |
|  |  |
| Imagen 11.13.9: campos editables, solo cuando se activan sus respectivas casillas, pero cuando se desactivan regresa el promedio original. | **Imagen 11.13.10:** estos campos son editables, son botones con rangos y en la parte de abajo se pueden observar los resultados al aplicar esta variación con los rangos. |

En este apartado (ver imagen 11.13.8) de los valores promedio se calculo con la siguiente ecuación:

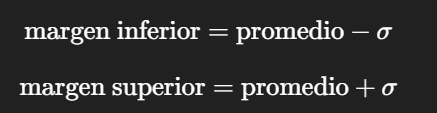
Calculamos el promedio con:



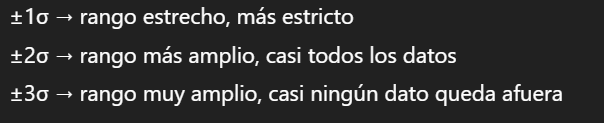
Luego calculamos la desviación estándar con:



usamos el promedio y sigma para definir un “margen fijo”:



Ahora por último aplicamos algún rango:



Estos rangos te permiten **decidir qué datos son “normales” y cuáles son atípicos**:

En el control de calidad se puede observar un Tab vacío, aquí es donde se espera que se puedan agregar mas filtros para el control de calidad, modificando el código original, pero esto lo discutiremos mas adelante.

|  |
| --- |
|  |
| Imagen 11.13.11: apartado para nuevos filtros. |

## **Análisis del archivo**

Aquí se realiza el análisis interno del archivo Toga, comprobamos que toda su estructura este de forma correcta, que no cuente con datos nulos y que se puedan identificar picos y Stuckdata.

En la imagen 12.1 podemos ver el momento exacto cuando se procesa un archivo y se activa la primera casilla (analizar y graficar), aquí es donde se muestra la grafica y luego procede a actualizar la ventana “Análisis del archivo”, donde podemos ver el resultado que se muestra en la **imagen 12.3**, se ejecutan dos scripts (ver **imagen 12.2**), donde mostraran un análisis mas profundo de los archivos TOGA.

|  |
| --- |
|  |
| Imagen 12.1: como se empieza el procesamiento del algún archivo. |

|  |
| --- |
|  |
| Imagen 12.2: Los scripts que se encuentran en el repositorio y se mandan a llamar, para realizar esa tarea en la ventana de “Análisis del archivo” |

|  |
| --- |
|  |
| Imagen 12.3: ubicación del script. El script numero 6 se ejecuta primero y luego el script número 11. |

Esta es la salida del script que identifican que archivo cuenta con el 100% de registros, sin datos nulos:

|  |
| --- |
| Se va a cargar el archivo: /content/h316a58.dat  Archivo h316a58.dat cargado, se han encontrado datos del anio 1958  El archivo debe contener 730 l'ineas de datos y se han encontrado 730  Fin del analisis, el archivo esta completo, se han encontrado las 730 lineas de datos esperadas.  Hay 8760 datos validos (100.00%) y 0 datos nulos (0.00%)  EN TOTAL EXISTE 8760 datos  Archivo: /content/h316a58.dat  No es bisiesto  La cantidad de datos esta completa  Se ha cargado el archivo /content/h316a58.dat y se han encontrado datos del anio 1958.  \*\*\*\*\* Inicio de la revision del archivo \*\*\*\*\*  ESTE ARCHIVO CONTIENE REGISTROS DE 365 dias  Comprobando que el archivo contenga el numero correcto de lineas de datos [OK]  Comprobando que el archivo no este conformado unicamente de datos nulos [OK]  Comprobando que el tiempo se incremente cronologicamente en los datos [OK]  Detectando si las columnas de datos contienen caracteres invalidos [OK]  \*\*\*\*\* analisis de los datos \*\*\*\*\*  Realizando prueba de estabilidad (linea recta)  Realizando la prueba de deteccion de picos.  Resumen del contol de calidad. Se han encontrado 0 picos y 28 lineas (stuckdata).  Se va a proceder a escribir el archivo de salida con los datos etiquetados. |

Esta es la salida del script que identifican archivos que cuentan con registros nulos:

|  |  |  |
| --- | --- | --- |
| Archivo: /content/h316a92.dat  ejecutando: 06\_comprobar\_datos\_toga.py  Se va a cargar el archivo: /content/h316a92.dat  Archivo h316a92.dat cargado, se han encontrado datos del anio 1992  El archivo debe contener 732 l'ineas de datos y se han encontrado 732  Fin del analisis, el archivo esta completo, se han encontrado las 732 lineas de datos esperadas.  Hay 0 datos validos (0.00%) y 8784 datos nulos (100.00%)  EN TOTAL EXISTE 8784 datos  Archivo: /content/h316a92.dat  Es bisiesto  La cantidad de datos esta incompleta ... Faltan 8784  Se ha cargado el archivo /content/h316a92.dat y se han encontrado datos del anio 1992.  \*\*\*\*\* Inicio de la revision del archivo \*\*\*\*\*  ESTE ARCHIVO CONTIENE REGISTROS DE 366 dias  Comprobando que el archivo contenga el numero correcto de lineas de datos [OK]  Comprobando que el archivo no este conformado unicamente de datos nulos [FAIL]  ERROR: El archivo de entrada no contiene datos, unicamente contiene valores nulos. No se puede procesar.  \*\*\*\*\* Fin de la revision del archivo \*\*\*\*\*  ====== FIN DEL ANALISIS ===============================   |  | | --- | |  | | Imagen 12.4: cuando se grafican 2 archivos, se analizan dos archivos por separado y si existieran más, habrá más divisiones. | |

Como podemos observar en estos ejemplos, como existen diferentes resultados, este apartado se ejecuta automáticamente al procesar algún archivo y si seleccionamos mas archivos veremos distintas salidas, por que cada una es independiente y se dividen por archivos procesados (ver **imagen 12.4).**

## **Archivo Toga**

Para ver este resultado no tienes que hacer nada, solo esperas que el programa ejecute el archivo completo, si vez funcionado los scripts en la ventana “Análisis de archivo” ya podrás ver el Datafreme del archivo toga seleccionado:

|  |
| --- |
|  |
|  |
|  |
| Imagen 13.1: como es la interacción del archivo toga. |

En la imagen 13.1 podemos observar como se puede ir deslizando y ver el contenido completo del archivo TOGA, de esta forma comprobamos que todos los análisis son correctos, también sirve para identificar como realmente es la estructura de los datos.

Cuando se grafican varias graficas aquí veremos varias divisiones, cada uno con responder al archivo seleccionado, se muestra todo en un contenedor para evitar scrollear demasiado y así ver el archivo de nuestro interés (ver imagen 13.2).

|  |
| --- |
|  |
| Imagen 13.2: división de Dataframe representando distintos archivos Toga. |

## **Resumen General**

Esto solo funciona al activar la segunda casilla (ver imagen 14) que se encuentra en el panel de selección de archivos (en la ventana de grafica), al activarla dirá que ya se están procesando estos archivos.

|  |
| --- |
|  |
| Imagen 14: panel en la ventana “Grafica” |

### **Detalles**

La casilla no se activa automáticamente como Analizar y grafiar, aquí si queremos ver esta función tenemos que activarla:

Iniciamos el programa y como hemos mandado a procesar todos estos registros entonces vemos que la casilla “Analizar y graficar” se activa:

|  |
| --- |
|  |
| Imagen 14.1 |

Entonces ahora procedemos a ver en la ventana Resumen general, observamos que esta vacío y no se encuentra ningún resultado.

|  |
| --- |
|  |
| Imagen 14.1.1 |

Entonces procedemos activar la casilla Resumen general, donde vemos que nos sale un mensaje de <<realizando resumen>>, ahora si vemos en la ventana de “Resumen general” que ya se pueden apreciar esos resultados.

|  |
| --- |
|  |
| Imagen 14.1.2: |
|  |
| Imagen 14.2 |

Ahora vemos que esta ventana nos muestra distintos apartados (ver imagen 14.2.1 hasta 14.2.4).

|  |
| --- |
|  |
| Imagen 14.2.1: La casilla sirve para seleccionar todos los archivos seleccionados, para resumir y el boton “Descargar Zip”, sirve para convertir cada archivos en un nuevo formato y luego procede a agruparlos en una archivo con extencion zip. |
|  |
|  |
| Imagen 14.2.2: Resumen general de cada uno de los archivos, con una vista previa de la grafica y un poco de informacion sobre el archivos, cada archivo esta en un contenedor. |
|  |
| Imagen 14.2.3: En el contenedor podemos ver el nombre del archivos, entre otros mensajes importantes como la cantidad de datos validos y nulos. Cada contenedor cuenta con su propia casilla. |
|  |
| Imagen 14.2.4: Aquí se puede apreciar todos los archivos a transformar y descargar en un nuevo formato. |

Luego de todo esto al presionar el botón Descargar zip, nos saldrá los siguientes mensajes indicándonos que los archivos están siendo procesados correctamente (ver la imagen 14.2.5 hasta 14.2.7).

|  |
| --- |
|  |
| Imagen 14.2.5: guardar el archivo |

|  |
| --- |
|  |
| Imagen 14.2.6: dentro del archivo zip descargado. |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Imagen 14.2.7: resultados de todos archivos transformados. | | |

### **Nota**

Esto solo funciona para archivos que fueron cargados desde pc ya que si son archivos que fueron cargados desde el drive, descargarlos con un nuevo formato no es posible ya que no encontrará esos archivos y por lo tanto el archivo zip estará vacío, prueba de ello se muestra en la imagen 14.3

Uno de los benéficos de usar el resumen general es que de esta forma puede realizar una búsqueda de alguna grafica en especifica, ver que grafico no cuenta con los registros completos y detectar archivos completamente vacíos.

De esta forma solo veras que grafica quieres analizar en la ventana Grafica, sin que tengas que ir graficando una por una, un ejemplo de su uso seria la que se muestra en la imagen 14.2.8.

|  |
| --- |
|  |
|  |
|  |
| Imagen 14.2.8: pasos para usar la ventana “Resumen general”. |

|  |
| --- |
|  |
|  |
|  |
| Imagen 14.3: Proceso de selección de un archivo con ruta desde drive, pero si fuera desde el entorno de trabajo de archivos cargados desde pc, su transformacion es posible. |

# **Agregar nuevo filtro**

Aquí te comparto como puedes agregar un nuevo filtro con algún nuevo método, en este caso vemos como se agregó el filtro StuckData, donde la lógica de programación se tomó del script número 11, donde mostraba la grafica original, la grafica sin picos y el stuckData.

|  |
| --- |
|  |
| Imagen 15.1: grafica aplicando el filtro del stuckData. |

|  |
| --- |
|  |
| Imagen 15.2: esta es la lógica para procesar el archivo como en Datafreme. |

Podemos ver la **imagen 15.2**, donde verificamos un poco sobre la lógica para procesar los datos del archivo toga, de esta forma reconstruimos la lista del registro (ver **imagen 15.3**), en la imagen 15.4 es donde programamos la lógica del StuckData.

|  |  |
| --- | --- |
|  |  |
| Imagen 15.3: lista de registros | |

|  |
| --- |
|  |
| Imagen 15.4: lista de registros |

En la **imagen 15.4**: es donde podemos ver un poco sobre la inserción de este nuevo método, se puede ubicar como se muestra en la **imagen 15.5**, de esta forma el proceso se realizará una vez finalizando se construyen un DataFrame de pandas a partir de estas dos listas (ver imagen 15.3), esto se puede observar en la **imagen 15.6**, donde se observa la primera línea de código y en la segunda agregamos nuevas columnas.

|  |
| --- |
|  |
| Imagen 15.5: fuera del bucle for, es donde se puede crear un nuevo método para procesar la lista de datos. |

|  |
| --- |
|  |
| Imagen 15.6: fragmento de código donde se crea el DataFrame. |

|  |
| --- |
|  |
| Imagen 15.7: ejemplo de cómo se vería las nuevas columnas (ver imagen 15.6). |

En la **imagen 15.6**, se observa la creación del Datraframe, esto permite que pandas alinee los nuevos datos con el índice existente, garantizando la coherencia estructural de la tabla aun cuando las longitudes de las listas sean diferentes.

En cada iteración, el DataFrame generando a partir de un archivo se almacena en la lista dfs, el nombre del archivo procesado se guarda en la lista archivos\_cargados.

Y de esa forma pasamos al siguiente apartado donde explicamos como integramos este nuevo filtro al gráfico.

|  |
| --- |
|  |
| Imagen 15.8: ese bucle for se dedica en procesar cada Datafreme. |

En la **imagen 15.8**, es donde se ven distintos ColumnDataSource, que sirve como contenedor de datos tabular, sirve como fuente de datos para gráficos, cada columna es un vector de datos, cada fila representa un registro o punto en la gráfica, permite interactividad, como mover puntos, actualizar datos dinámicamente o usar herramientas como hover, selección y callbacks. También esta en un bucle for donde se recorre simultáneamente la lista de DataFrames (dfs) y la lista de nombres de archivos (archivos\_cargados) utilizando zip (), mientras enumerate() proporciona un índice de interacción. Esto permite procesar cada DataFrame junto con su archivo correspondiente de manera ordenada y numerada.

|  |
| --- |
|  |
| Imagen 15.9: representación de los puntos stuck mediante una línea roja(linea3) y, adicionalmente, se marcaron con estrellas doradas(linea4) usando un glyph independiente. |

En la **imagen 15.9** vemos el siguiente paso para representar esos datos en la gráfica, además se puede observar que también se realiza la construcción de leyendas, donde se ajustan para identificar los puntos destacados, y la línea se almacena en la lista línea 33 para permitir futuras modificaciones o referencias.

Por último, si queremos ya observar el nuevo filtro, comprobamos si realimente está procesando correctamente los datos, ejecutamos el programa y veremos el siguiente resultado como la **imagen 15.1**.

Nota: tenemos que tener declarado las variables línea33 y línea44 como globales (ver **imagen 15.10**).

|  |
| --- |
|  |
| Imagen 15.10: creamos estas dos listas como globales para guardar todos los registros. |

Como se observa en la **imagen 15.11**: se recorrieron las listas linea33 y línea44 asignando la propiedad visible = False a cada elemento, con el fin de ocultar temporalmente esas líneas en la grafica sin eliminarlas. Luego tenemos la siguiente línea:

|  |
| --- |
| lineas33\_44 = linea33 + linea44 |

Lo que hace es unir las listas línea33 y línea44 en una sola (líneas33\_44) para recorrerlas en conjunto y asignar visible = False, ocultando todas las líneas relacionadas en una sola operación.

La siguiente es:

|  |
| --- |
| checkbox\_StuckData = CheckboxGroup(labels=["Mostrar StuckData"]) |

Se implemento un CheckboxGroup con la etiqueta “Mostrar StuckData”, el cual permite al usuario controlar de manera interactiva la visibilidad de las líneas correspondientes a los datos StuckData.

Por último, en la **imagen 15.11** se observa el siguiente fragmento de código:

|  |
| --- |
| checkbox\_StuckData.js\_on\_change("active",CustomJS(args=dict(lineas=lineas33\_44), code="""            const mostrar = this.active.includes(0); // solo una casilla            for (let i = 0; i < lineas.length; i++) {                lineas[i].visible = mostrar;      }    """)) |

Aquí se asoció un CustomJs al evento active del CheckboxGroup. Lo que vemos es un código en JavaScript que recorre la lista lineas33\_44 y cambia la propiedad visible de cada línea según el estado de la casilla, permitiendo al usuario mostrar u ocultar los datos StuckData de forma interactiva.

|  |
| --- |
|  |
| Imagen 15.11: checkbox de la línea33 y línea44 |

En la **imagen 15.12**, nos disponemos a agregar los controles checkbox\_pico\_valle y checkbox\_stuckData en un mismo contenedor tipo row, con el fin de presentarlos de manera alineada en la interfaz de la gráfica.

Luego vemos la flecha roja que señala la siguiente línea:

Tab\_16 = TabPanel(child=Tab\_14, title="Parte 1")

Se crea un TabPanel con el contenido de Tab\_14 y el título “Parte 1”, permitiendo organizar la visualización en pestañas dentro de la aplicación interactiva. Ya por último vemos la siguiente línea de código

Tab\_13 = Tabs(tabs=[Tab\_16, Tab\_15])

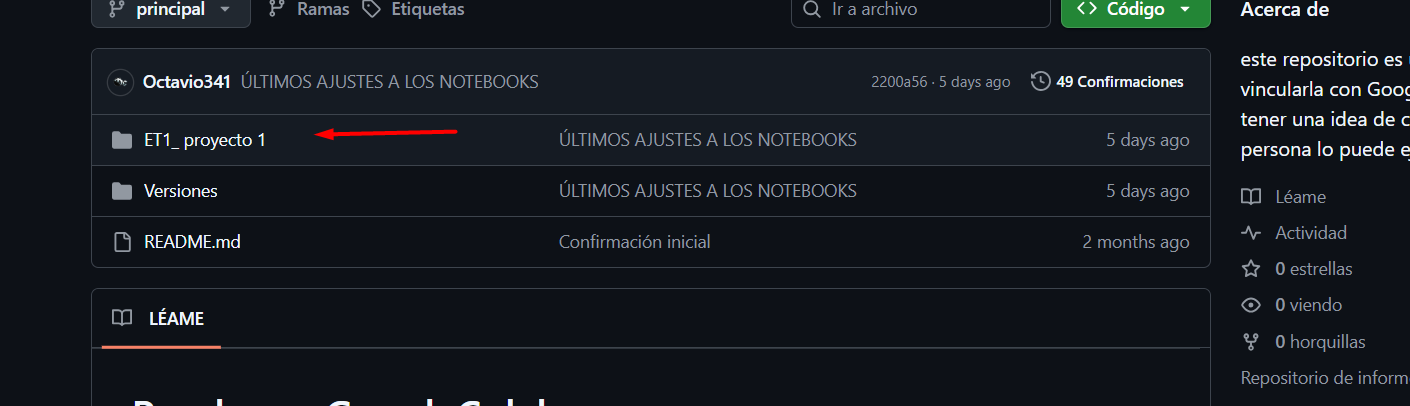
Aquí se implementa un contenedor Tabs para agrupar los paneles Tab\_16 y Tab\_15, permitiendo organizar el contenido en pestañas de navegación dentro de la aplicación.

|  |
| --- |
|  |
| Imagen 15.12: organizador de chekboxes en un mismo reglón (layout horizontal) dentro del interfaz Bokeh. |

|  |
| --- |
|  |
| Imagen 15.13: Grafica con filtro StuckData y su casilla para mostrarla y ocultarla. |
| |  | | --- | |  | | Imagen 15.14: resultados referente a los stuckData. | |

Echo todo esto, si ahora al ejecutar el programa podremos ver que el interfaz ya nos aparece una nueva casilla tal y como la configuramos (ver **imagen 15.13**), además podemos hacer utilidad de la ventana Análisis del archivo (donde se muestra la salida de los scripts) comprobamos que datos realimente aplico ese filtro, puede que a veces este filtro no funcione y solo se pueda activar la casilla sin mostrar ninguna línea, siempre que usemos este filtro verifiquemos la salida de los scripts para ver cuando se les aplica a los datos su stuckData(ver **imagen 15.14**).

# **Mas información:**

Te invito a seguir explorando más de esta área, si quieres saber más información sobre las actividades realizadas en mi estancia, en el siguiente repositorio encontraras un poco de contenido que puede ser de utilidad: <https://github.com/Octavio341/MareograficoGoogleColab.git>

Entrando a la carpeta ET1\_ proyecto 1 encontraras todos los scrips que hicieron posible que este proyecto funcionara, también encontraras algunos programas, que había propuesto mucho antes de comenzar a desarrollar este proyecto, entre otros recursos utilizados en el proceso.

# **Conclusión**

Hemos llegado al final de este manual, esto fue una explicación breve de cómo usar el este programa, la verdad, aunque muestre simpleza, tuvo un largo proceso, paso por pruebas y estudio profundo, y aunque faltaron aún más cosas por agregar, se puede decir que se presenta un trabajo que cumple con herramientas básicas, todo esto es gracias a los scripts que se me proporciono, para analizar y así proponer una herramienta que combine funciones útiles en ciertos análisis.

La verdad fue muy gratificante haber trabajado en este interfaz me dejo buenas experiencias, y cada función me hacía comprender mejor como se analizaban los datos, si alguien llegara a agregar nuevas funciones recomendaría hacer pruebas pequeñas, comprender paso a paso cada instrucción y cada avance se va creando nuevas necesidades, comenzar por cosas básicas nos abre paso a crear algo grande.

Durante el desarrollo del proyecto, reforcé mis conocimientos en programación en Python y el uso de librerías como PySide 6, Bokeh e ipywidgets para crear interfaces interactivas y graficas dinámicas, cada filtro y herramienta solo incentivaban mas ideas para seguir implementado, existieron mas ideas, pero por el momento solo estos fueron los que pude implementar, tengo que reconocer que existe ciertos detalles al usar archivos TOGA desde el drive, aunque no lo pode resolver, esto pasa al usar la herramienta resumen general, al crear nuevos formatos, este no funciona aquí, solo con archivos subidos desde pc, puede que en el futuro este detalle alguien lo podrá resolver o puede que inspire en la creación de un software más completo y rápido, ya que estando en un notebook, los recursos de la plataforma Google Colab son muy limitados, solo contamos con 1.90 GB de memoria RAM, no podemos realizar un análisis con mas datos ya que luego se congela el programa y la interacción se pierde, también estas fueron algunas de las razones las cuales no agregue filtros, porque comprometía mucho el rendimiento.

Llegar a este punto no fue nada fácil, aunque me demore en mostrar avances, la verdad es que perfeccionaba cada parte del programa, verificaba cada detalle y eso me consumió mucho tiempo, pero agradezco al maestro Octavio Gómez Ramos, jefe del Servicio Mareográfico Nacional y a la maestra Miriam Arianna Zarza Alvarado, por haberme guiado en esta trayectoria, ya que ellos me ayudaron a moldear el proyecto porque antes de esto fue un script .bat y luego fue **una aplicación de escritorio** (o **GUI applications**) con Qt/PySide6 y finalmente en un notebook.

# **Referencias**

* Bokeh Development Team. (2024). Bokeh (versión más reciente) [Software]. <https://bokeh.org/>
* Jupyter Widgets Community. (2024). ipywidgets (versión más reciente) [Software]. <https://ipywidgets.readthedocs.io/>
* IPython Development Team. (2024). IPython (versión más reciente) [Software]. <https://ipython.org/>
* OpenAI. (2025). ChatGPT (2025 versión) [Modelo de lenguaje]. <https://chat.openai.com/>
* Google. (2024). \*Google Colaboratory\* [Software]. <https://colab.research.google.com/>
* Instituto de Ciencias de la Atmósfera y Cambio Climático & Servicio Mareográfico Nacional. (2023). Actualización de las tendencias del nivel del mar en las costas de México (Informe\_tecnico-PINCC) [PDF]. Universidad Nacional Autónoma de México.
* Instituto de Ciencias de la Atmósfera y Cambio Climático & Servicio Mareográfico Nacional. (2023). Actualización de las tendencias del nivel del mar en las costas de México: Reporte de avances, primer semestre (Informe técnico) [PDF]. Universidad Nacional Autónoma de México.
* Instituto de Geofísica servicio mareografico nacional. (2023) Reporte de control de calidad a la estación(2023\_01\_25-12AYT-RCC)[PDF]. Universidad Nacional Autónoma de México.
* Servicio Mareográfico Nacional, Instituto de Geofísica, UNAM. (s. f.). Servicio Mareográfico Nacional [Sitio web]. Universidad Nacional Autónoma de México. <https://www.mareografico.unam.mx/>