MANUAL DE USUARIO SBDART

#### alujan@unal.edu.co - flodonop@eafit.edu.co

- Alejandro Luján Rocha ([alujan@unal.edu.co](mailto:alujan@unal.edu.co)) está dispuesto a solucionar dudas sobre la instalación del programa en caso de que se presente algún problema siguiendo las instrucciones de instalación. (No posee conocimientos del funcionamiento del programa)

- Fernando Londoño Palacio ([flondonop@eafit.edu.co](mailto:flondonop@eafit.edu.co)) está dispuesto a solucionar dudas sobre la manipulación del programa y tiene experiencia trabajando en el modelo de transferencia radiativa.

### Instalación:

A continuación, se establece un paso a paso para la instalación del programa SBDART, todo esto es basado en los detalles expuestos en [foros pasados](https://www.researchgate.net/post/Does_anyone_have_a_working_SBDART_model#view=65b1373316a23509d70b6a6c) y detalles externos propuestos para poder emplear el programa.

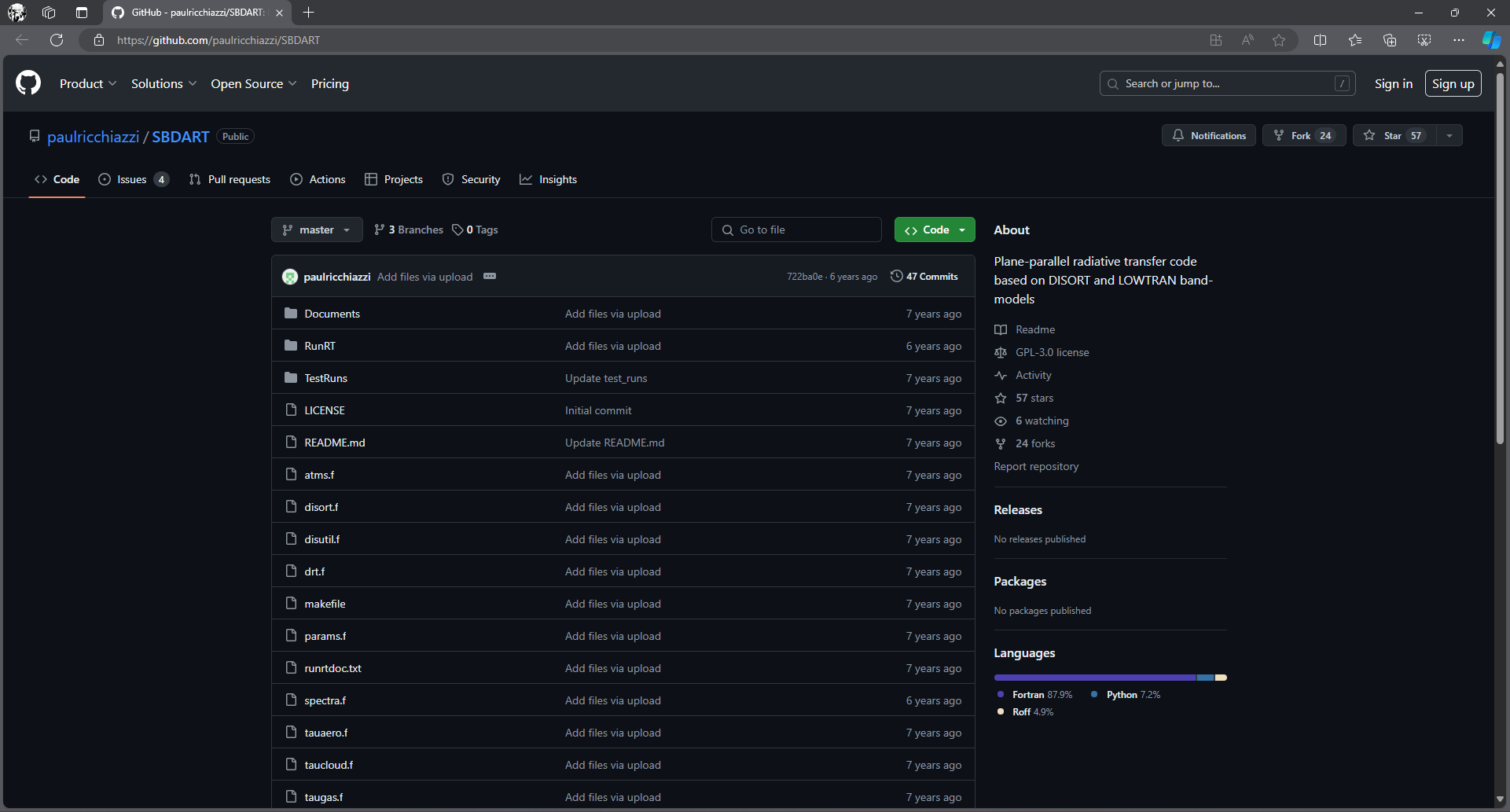
#### Paso 0: Prerrequisitos necesarios para este proceso

Para realizar la correcta instalación del SBDART, se requiere tener los siguientes componentes y programas ya instalados y en la última versión a menos que se indique lo contrario:

* GNU: Gzip, Gfortran, Tar
* Python2.7 o Python3.7 *(o más reciente estable)*
  + Tkinker == 0.1.0
  + Matplotlib == 3.9.0
  + Numpy == 1.26.4

#### Paso 1: Descarga/Clonado del repositorio

Primeramente, debemos descargar la versión que deseamos correr de la aplicación, en el [repositorio oficial](https://github.com/paulricchiazzi/SBDART/tree/python_3.7) en *main* se encuentra la versión original de la aplicación basada en Lowtran, disort y la interfaz creada en Python 2.7, esta versión corre en todos los computadores basados en LINUX y MACOSX, pero ya que algunas versiones de Windows por defecto no incorporan este interprete, se recomienda emplear la versión basada en Python 3.7, que se encuentra en la rama *Python\_3.7*, esta permite correr el programa empleando versiones estables actuales de Python.



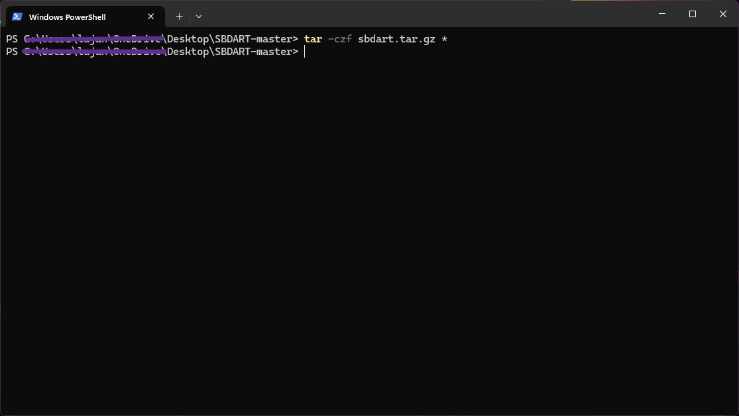
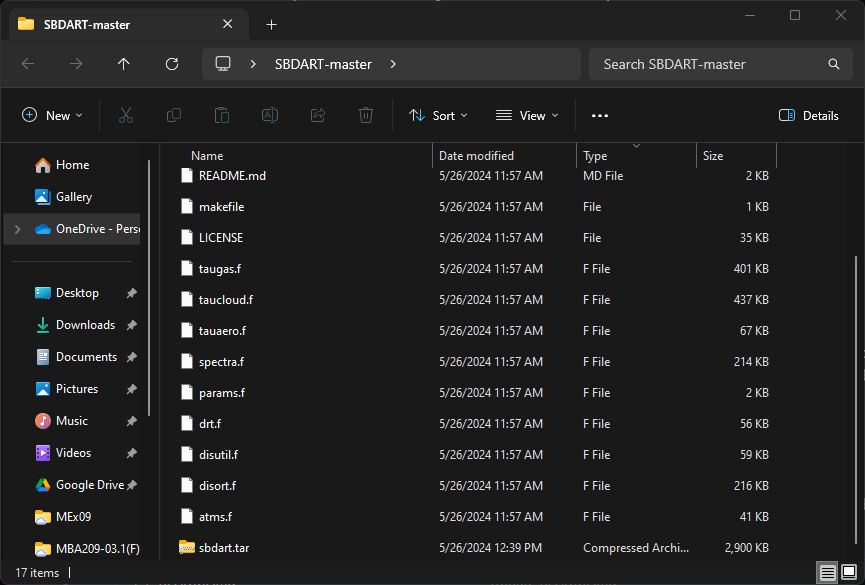
**Figura 1.** Repositorio oficial de SBDART

#### Paso 2: Creación de Tarball .gz

Con el repositorio a la mano, podemos crear el archivo portable Tarball, con este es que vamos a crear nuestro programa en Tar, para esto dentro de la carpeta del SBDART se debe correr el comando:

tar -czf sbdart.tar.gz \*

Esto genera un archivo .gz que posteriormente vamos a descomprimir para emplearlo.

A) B)

**Figura 2.** A)Comando y Feedback tras ejecución B) Archivo generado

#### Paso 3: Descompresión en Tar .tar

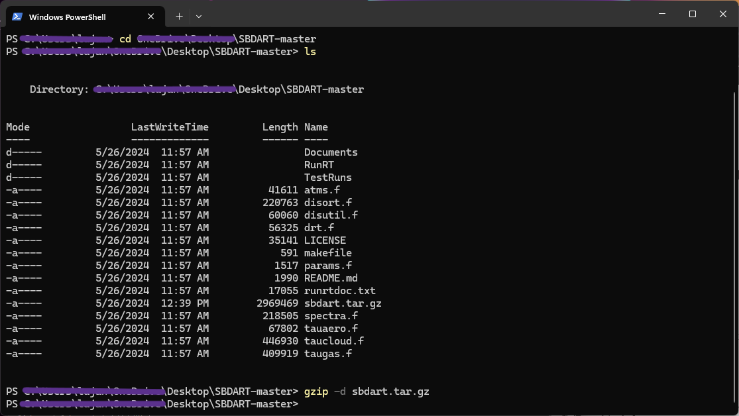
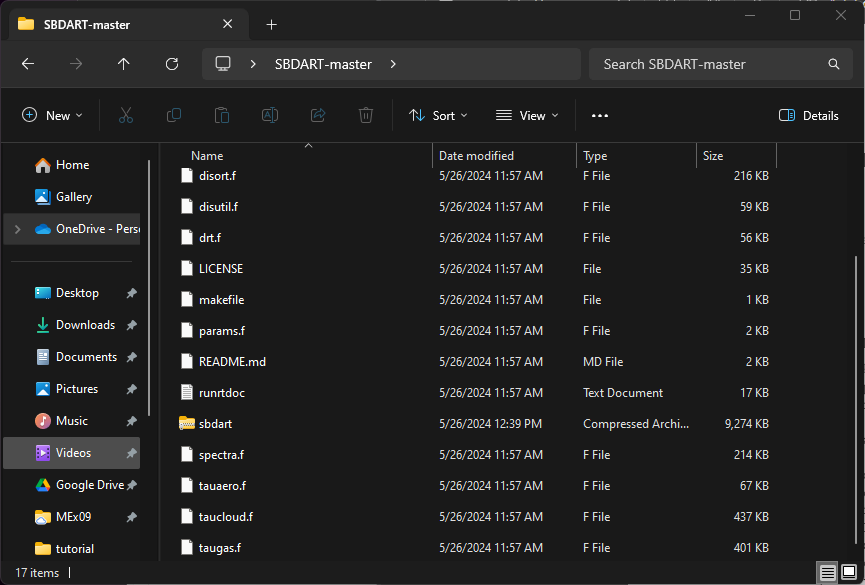
De una vez con el comprimido creado, se corre el siguiente comando para generar la descompresión:

gzip -d sbdart.tar.gz

En caso de que la instalación se realice en un sistema operativo basado en Linux no se requiere especificar funciones internas, por lo que el comando toma la siguiente forma:

gunzip sbdart.tar.gz

Esto genera un archivo tar descomprimido que emplearmos para generar las rutas necesarias

A) B)

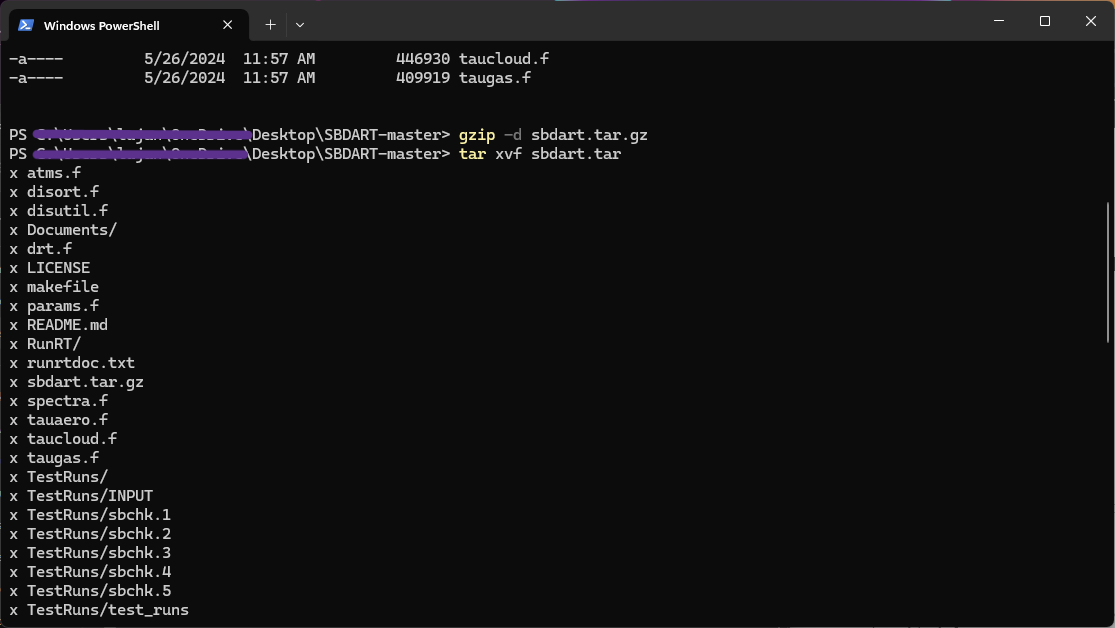
**Figura 3.** A)Comando y Feedback tras ejecución B) Archivo generado

#### Paso 4: Establecimiento de aplicación y rutas con Tar

Este paso crítico, ya que se va a generar rutas a partir de los archivos comprimidos creados, en caso de que todo el proceso anterior se haya realizado exitosamente, únicamente se requiere correr el comando:

tar xvf sbdart.tar

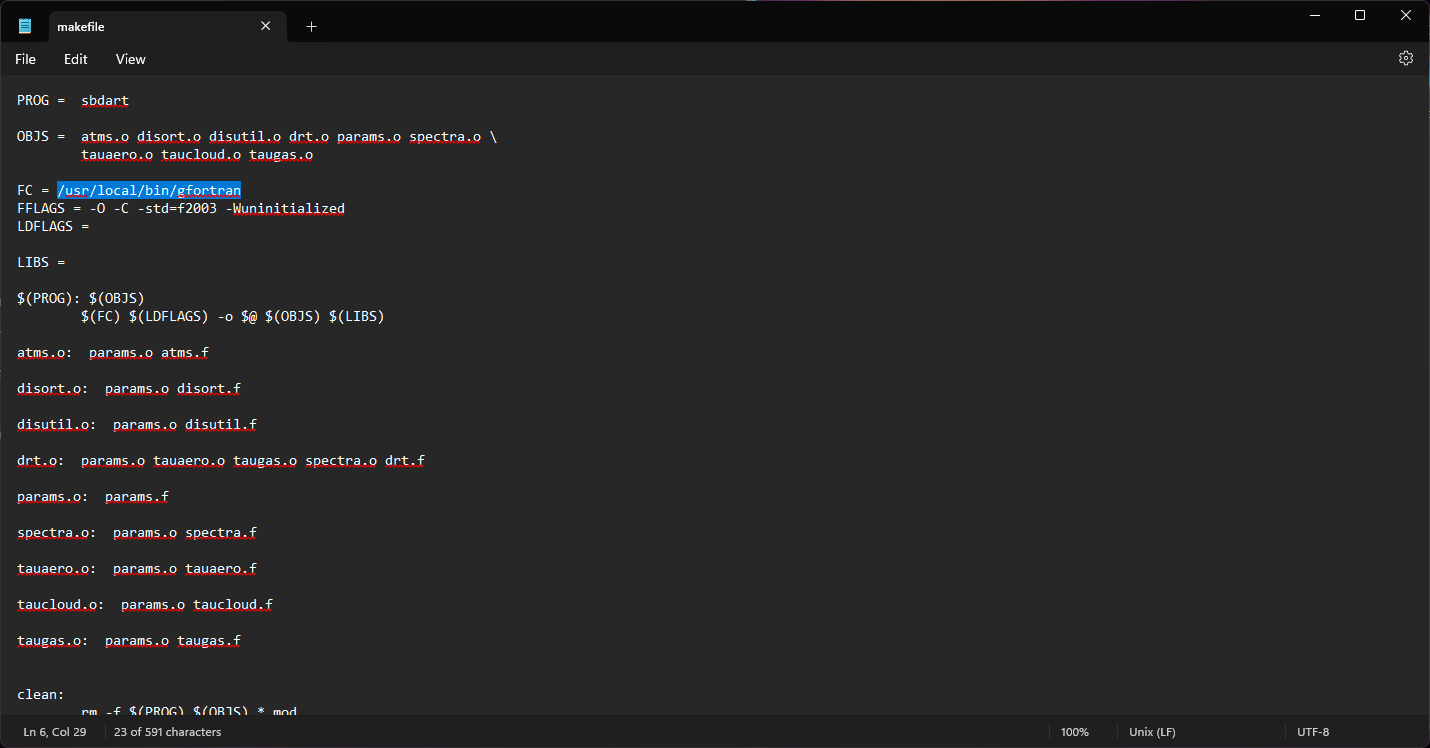
Todo se ejecuta correctamente si podemos visualizar en consola las diferentes rutas que se generar en el archivo para la creación de la aplicación:



**Figura 4.** Ejecución del comando y feedback esperado

#### Paso 5: Creación de la app MAKE

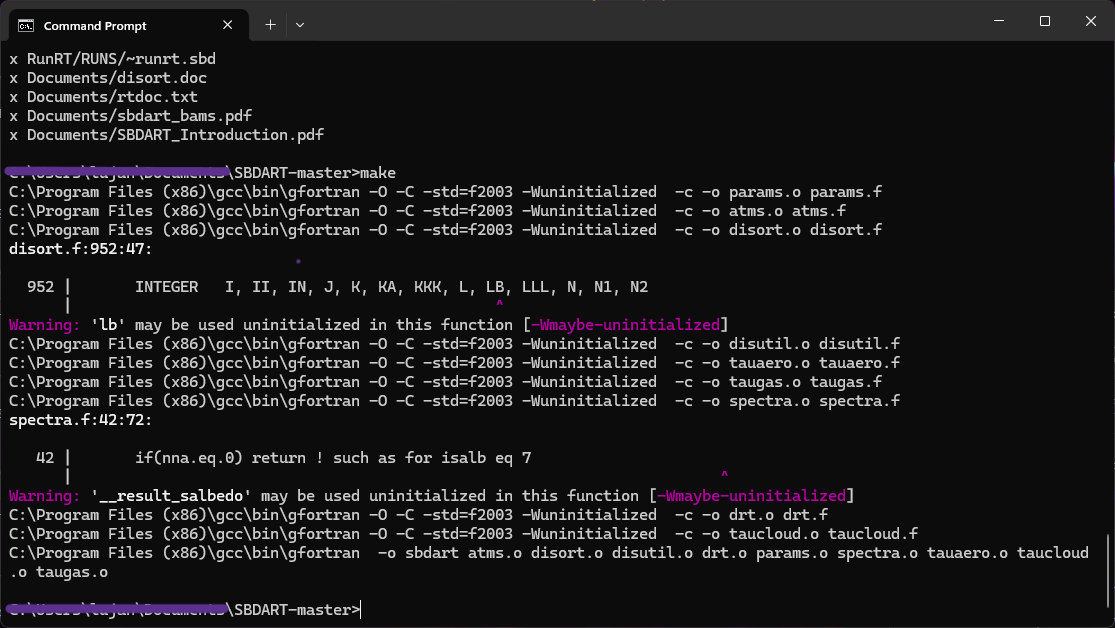
Antes de crear la aplicación, debemos comenzar a enrutar la aplicación a cada una de las dependencias que va a tener, para esto es necesario modificar el archivo **makefile**, y cambiar la línea que hace referencia a la ruta de *gfortran*, aquí es necesario usar una ruta absoluta del sistema como: *C:\Program Files (x86)\gcc\bin\gfortran* **(buscar la ruta de gfortran)**

**

**Figura 5.** Línea para modificar en el archivo makefile

Una vez esta ruta actualizada debemos ser capaces de correr la línea de comando:

make



**Figura 6.** Ejecución del comando y feedback esperado

#### Paso 6: Inicio y correcciones de la aplicación Python

Con los pasos anteriores los archivos basados en LOWTRAN y DISORT ya quedaron compilados y listos para ejecutar. Pero este repositorio nos adiciona una interfaz gráfica con la cual podemos emplear estos códigos de bajo nivel, sin necesidad de manipulación directa en fortran o llamado de módulos u objetos. Para esto debemos emplear el archivo *RunRT.py* que se encuentra en la carpeta con el mismo nombre **RunRT**.

Es común para algunos usuarios experimentar un problema al correr este archivo, gracias a ambigüedades que quedaron del código original.

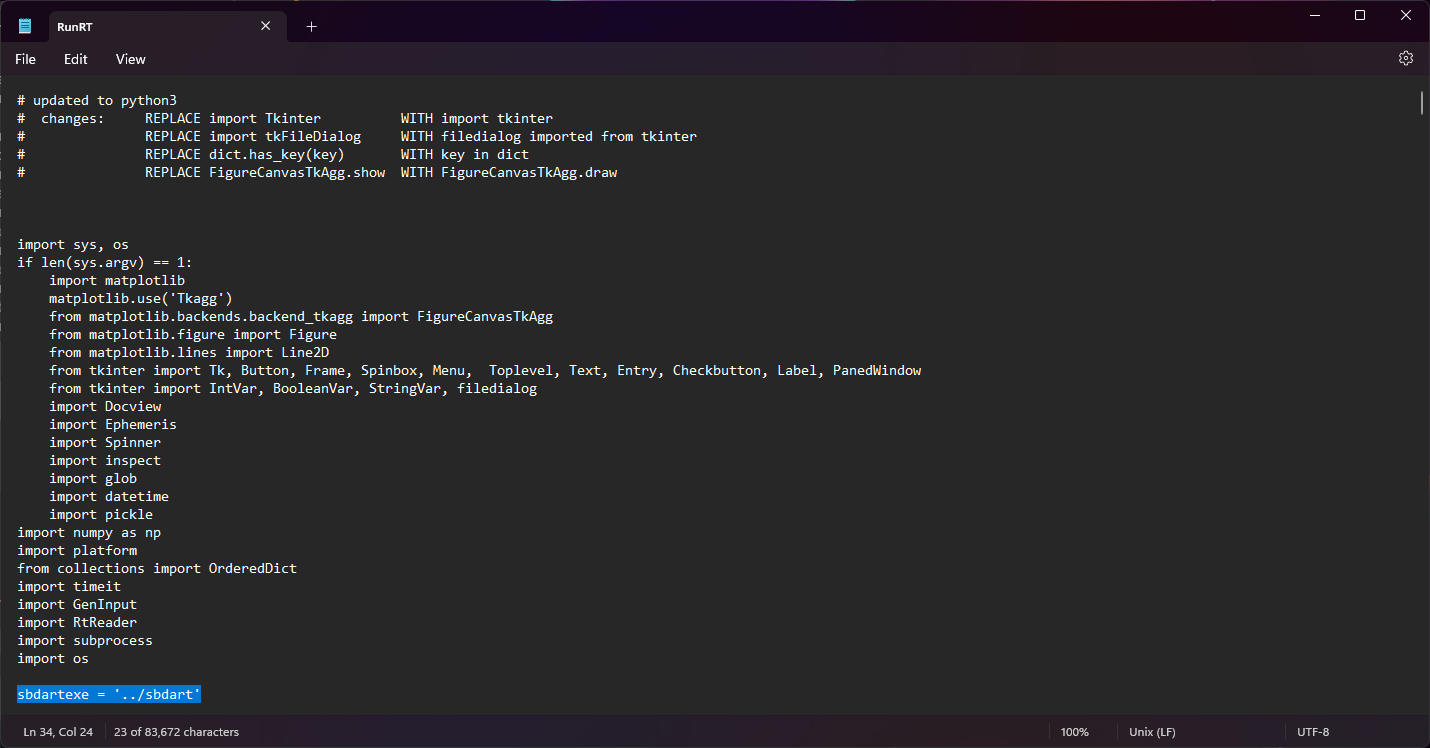
**A screenshot of a computer

Description automatically generated**

**Figura 7.** Ejecución del comando y feedback con fallas

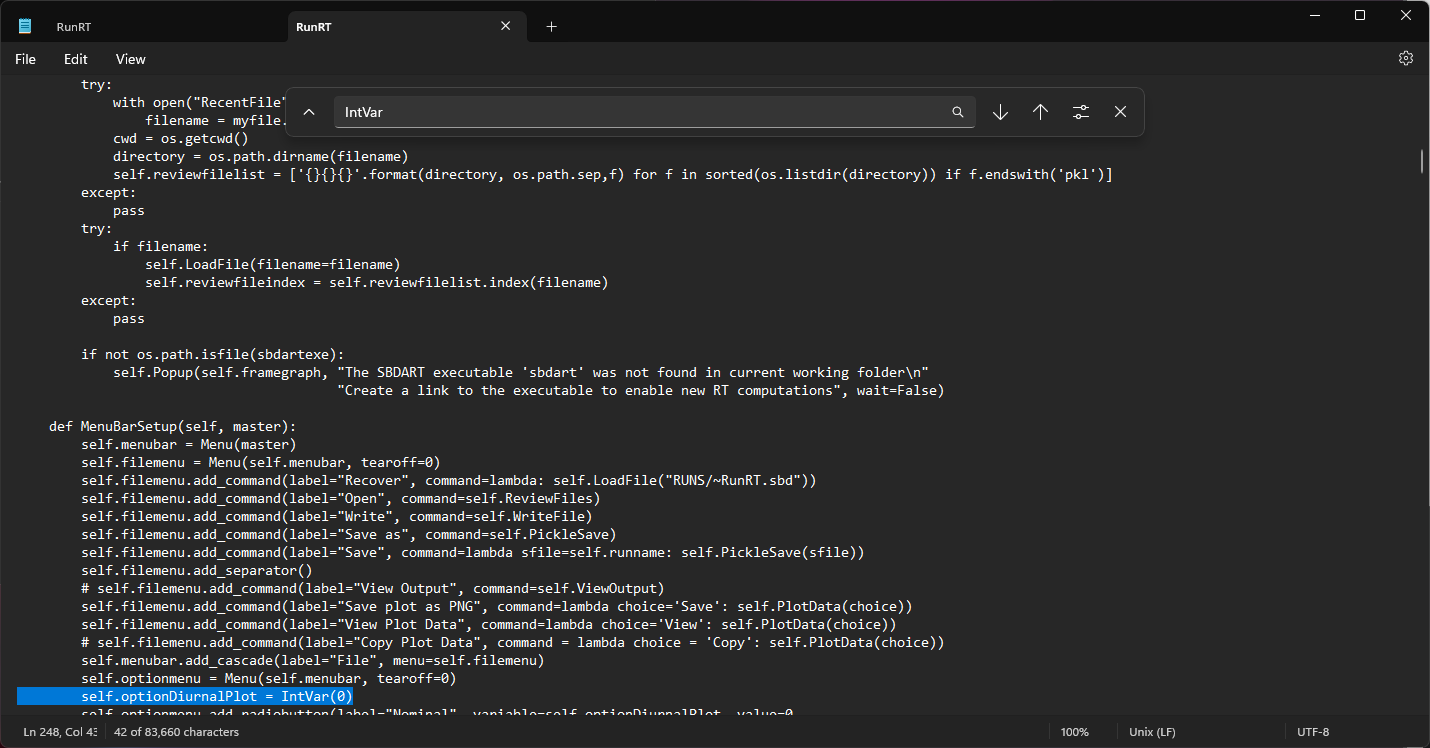
Para recuperar el programa se requiere modificar el código fuente de la aplicación para esto se requieren 2 cambios:

* **Cambio 1:** Se debe modificar la ruta relativa al ejecutable de SBDART, ya que la carpeta en donde se encuentra la interfaz no es la misma en la que está el ejecutable para esto modificamos la declaración de la ruta de la variable **sbdartexe**, por la ruta donde queda el ejecutable.



**Figura 8.** Primer cambio del archivo RunRT.py : Ruta relativa del ejectable

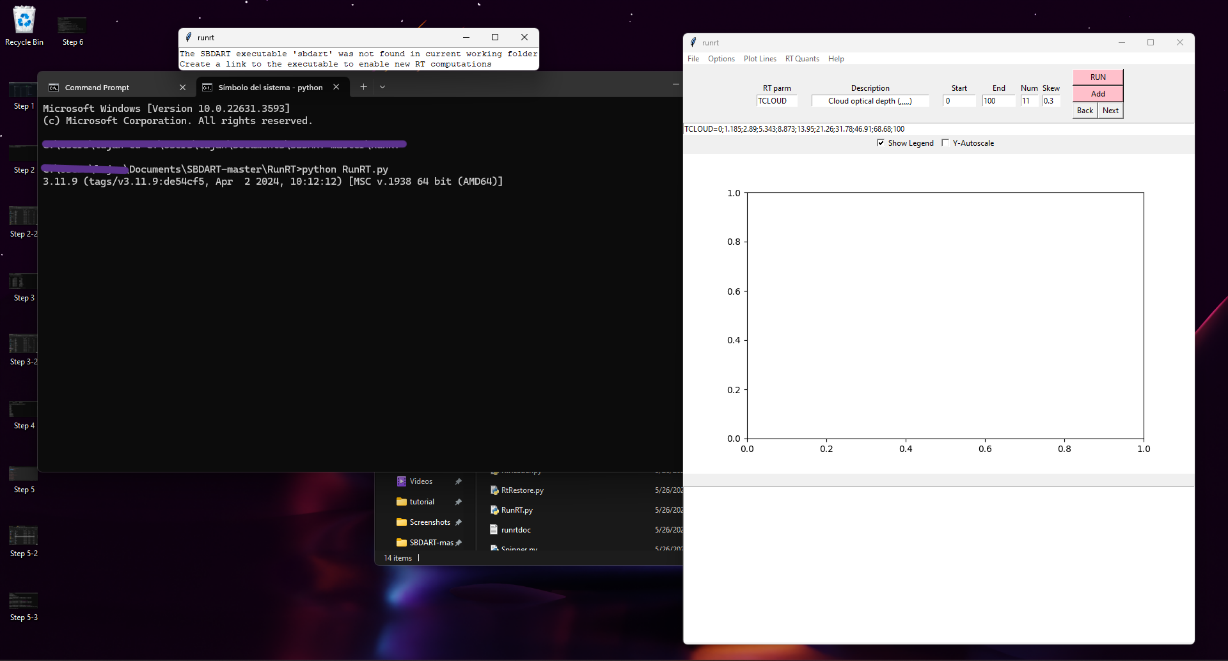
* **Cambio 2:** Se debe hacer una especificación en las funciones del modulo tkinter, y es necesario especificar en todos los llamados de la función **IntVar**, que el valor dado, hace referencia a la variable **value** y no una prioridad para maestro (default), esto lo logramos especificando en el valor que tiene cada llamado como *value=x.*



**Figura 9.** Segundo cambio del archivo RunRT.py: Especificación de valor

#### Paso 7: Inicialización SBDART

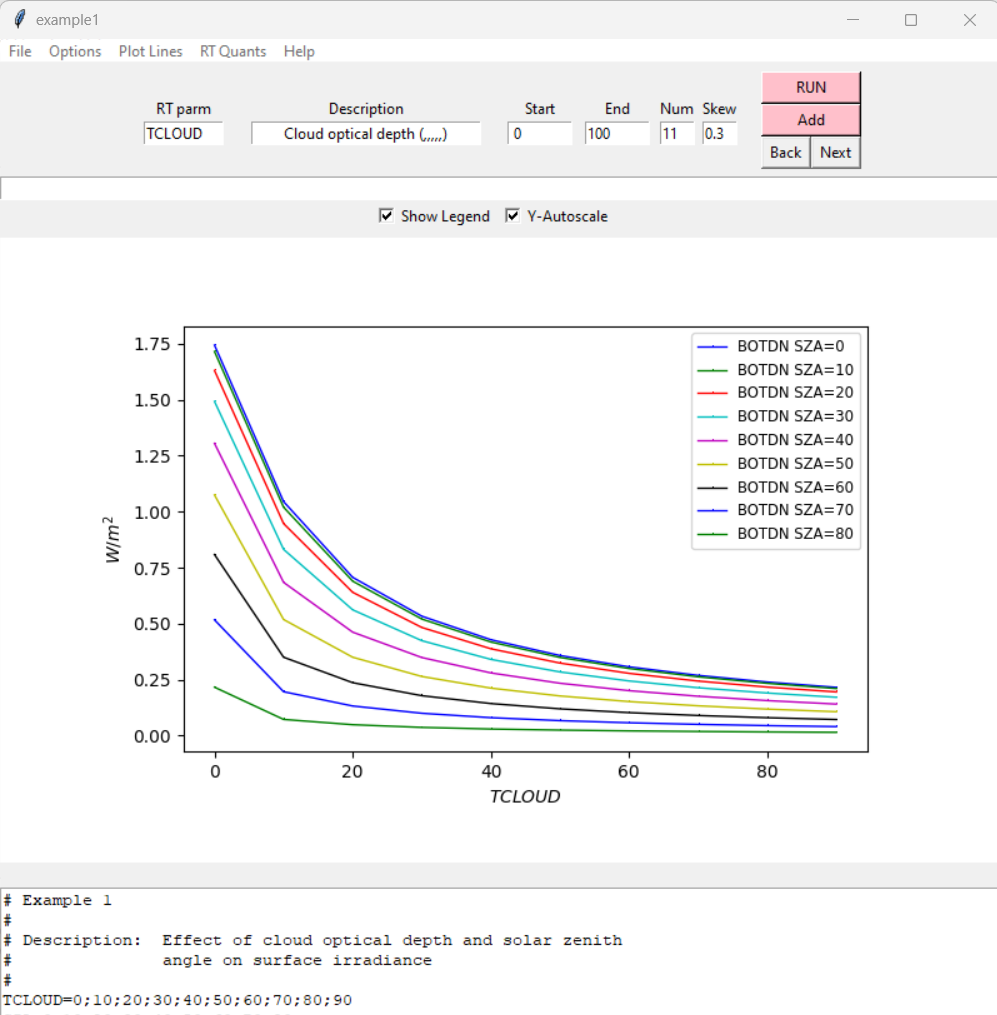
Con el desarrollo anterior realizado de manera correcta, solo basta con hacer un llamado al programa desde la carpeta de RunRT, para que la interfaz inicie y podamos comenzar a emplear el programa.



**Figura 10.** Llamada a ejecución tras correcciones realizadas

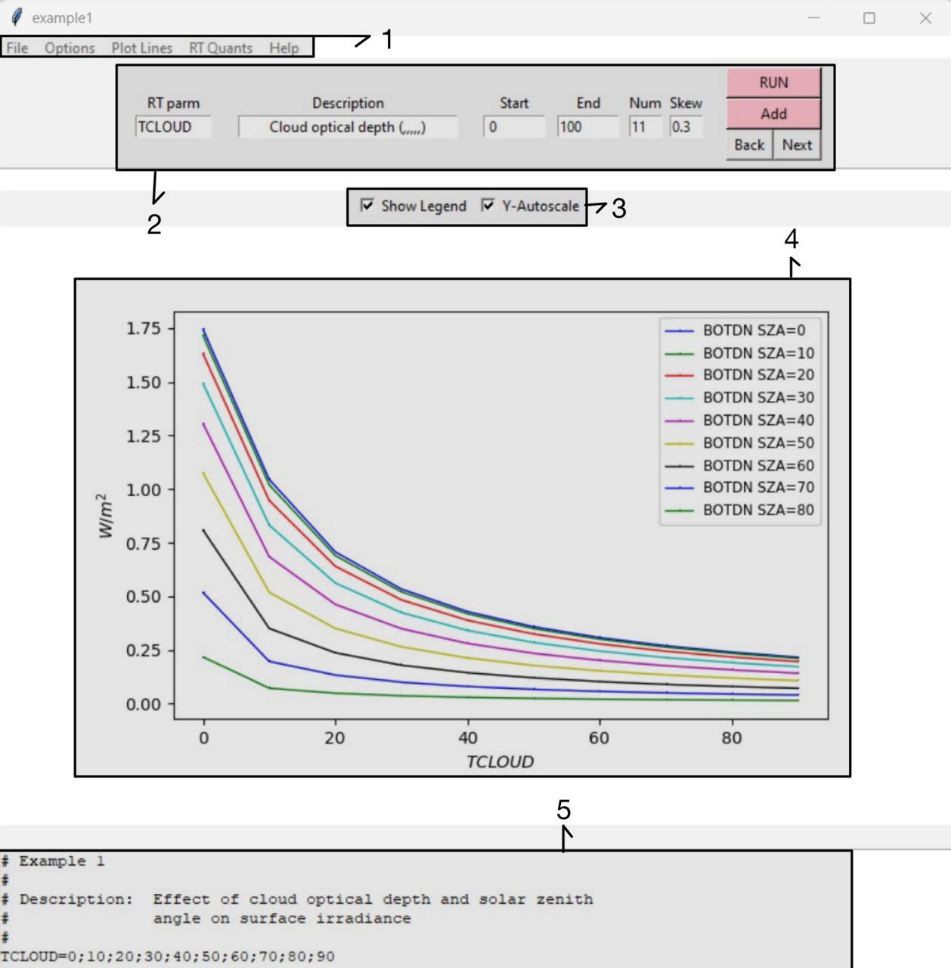
### ¿Cómo trabajar con SBDART?

Una vez se tiene el programa instalado y abierto, se pueden apreciar unos cuantos ejemplos detallados por el creador que se mostrarán por pantalla así:



**Figura 11.** Ejemplo 1 - SBDART

Ahora bien, entender la forma en la que funciona la interfaz gráfica resulta importante, por ello, acá dejamos explícita la forma de interactuar con el programa.



**Figura 11.** Interfaz Gráfica - SBDART

Primero, es necesario familiarizarse con las secciones principales del programa, en la Figura 11 se puede apreciar cada una de estas enumeradas.

#### Opciones del programa:

En este apartado se pueden encontrar diferentes opciones, entre ellas se encuentran:

“File”: Donde se podrán abrir archivos, escribir archivos, opciones de guardado, y también se encuentra una opción que muestra por pantalla los datos con los que fue generada la gráfica. (Darle especial atención a esta opción ya que la interfaz gráfica muestra las salidas del modelo únicamente con gráficas y si se desea realizar un análisis de datos se debe recurrir a esa opción para analizarlos con Python, Excel, Matlab o el programa de su preferencia.

“Options”: En este apartado se selecciona si se desea realizar promedios diarios, flujos horarios o gráficas de comparación, de razones o sin comparación.

“Plot Lines”: En este apartado puedes seleccionar específicamente qué líneas deseas mostrar por pantalla dependiendo de la salida que hayas elegido.

“RT Quants”: En este apartado puedes elegir qué tipo de salidas quieres mostrar, la intensidad, intensidad transmitida, flujos Top of Atmosphere o Bottom of Atmosphere.

“Help”: En este apartado se muestra RunRT, que es un archivo de texto en donde se explica brevemente el programa SBDART y se detallan los ejemplos predefinidos; también se muestra “RTdoc”, que es un archivo de texto que detalla cada uno de los parámetros de entrada del modelo SBDART, los parámetros de entrada y las salidas; también se muestra cada una de las advertencias que podría arrojar el modelo y cómo solucionarlas o tenerlas en cuenta.

Es necesario aclarar que ante cualquier duda específica de los parámetros se debe recurrir a este archivo o al documento en Word “Funcionamiento – SBDART” creado por mí, donde paso esta información e intento recopilarla brevemente y organizarla corrigiendo algunos errores de redacción y formato del archivo original para que sea más comprensible.

#### Parámetros del programa:

En este apartado, se encuentran varios recuadros de especial importancia y es acá en donde se realizan las primeras interacciones directas con el programa.

“RT parm”: Es la primera forma y más intuitiva de seleccionar parámetros de entrada del modelo, una vez hayas dado click en el recuadro blanco, podrás seleccionar entre los principales parámetros de entrada empleando la rueda del mouse.

“Description”: Este te otorga una leve descripción del parámetro a utilizar seguido del formato de entrada de este parámetro. Es importante recalcar que si no se tiene conocimiento de qué es el parámetro y se desea comprender a detalle la entrada del programa se debe recurrir al archivo “RTdoc” o “Funcionamiento – SBDART”.

“Start”, “End”, “Num” y “Skew”: Estos 4 parámetros te permiten, nuevamente elegir el inicio del parámetro de entrada, el final, la cantidad de datos que se desea tomar entre cada uno de ellos y el ajuste que utilizará para el llenado.

“Add”: Después de haber elegido el parámetro y haber realizado el ajuste interactuando con las 4 opciones anteriores, al dar click sobre este botón se añadirán a las entradas del programa, es importante recalcar que si no se da click a este botón, el programa no guardará los cambios si cambias de parámetro.

“Back”, “Next”: Estos dos permiten cambiar entre los diferentes ejemplos predefinidos o guardar simulaciones diferentes al mismo tiempo.

“RUN”: Es la forma en la que la interfaz gráfica comienza a realizar la simulación deseada.

#### Opciones de visualización:

Muestra dos opciones seleccionables en donde se muestran las leyendas de las líneas mostradas y te permite realizar un autoescalado de las salidas en el eje Y. Dependiendo de los parámetros, también podrá aparecer una nueva opción que te permite mostrar la misma salida del programa para diferentes valores de algunos parámetros.

#### Visualización de las salidas:

Este apartado es únicamente la forma en la que la interfaz gráfica muestra las salidas obtenidas con los códigos predefinidos del archivo RunRT.py (Si deseas realizar un ajuste en esta visualización debes modificar este archivo directamente.

#### Entradas del programa:

Este recuadro muestra las entradas del programa y va directamente a ser leído por las diferentes subrutinas en Fortran. Se recomienda que no sea manipulado sin tener un conocimiento adecuado del formato de entrada de cada uno de los parámetros seleccionados o el programa va a presentar errores en caso de que la sintaxis que hayas utilizado no sea la que está definida para Fortran90 o no sea explícitamente el formato de entrada de cada uno de los parámetros seleccionados.