27-3-2023

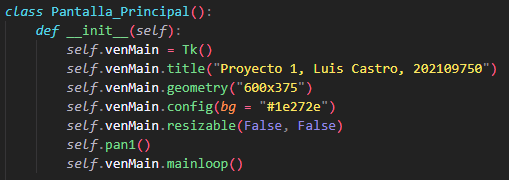
Luis Antonio Castro Padilla

CARNET: 202109750

Manual técnico

Proyecto 1 LFP B+

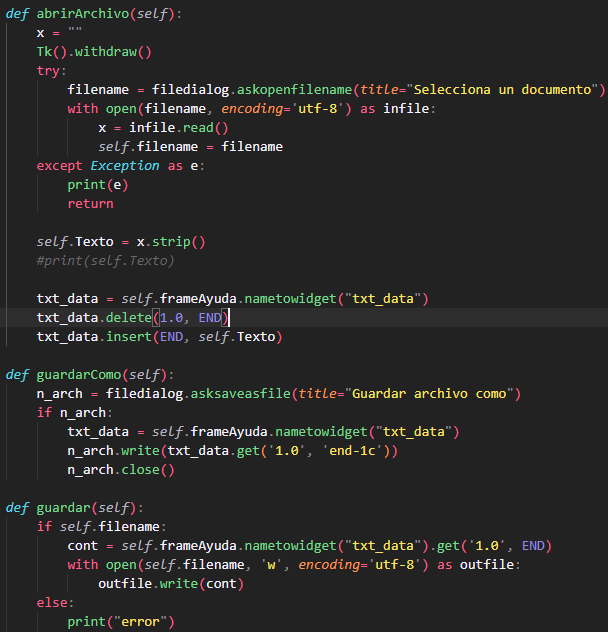
En la interfaz gráfica se creará la clase de pantalla principal, a la cual se le declarará que al momento de iniciar creará una ventana que tendrá un título específico, un tamaño de 600x375 px un fondo de pantalla verde menta, además tendrá la característica de que no se podrá ampliar, además llamará a la función pan1.



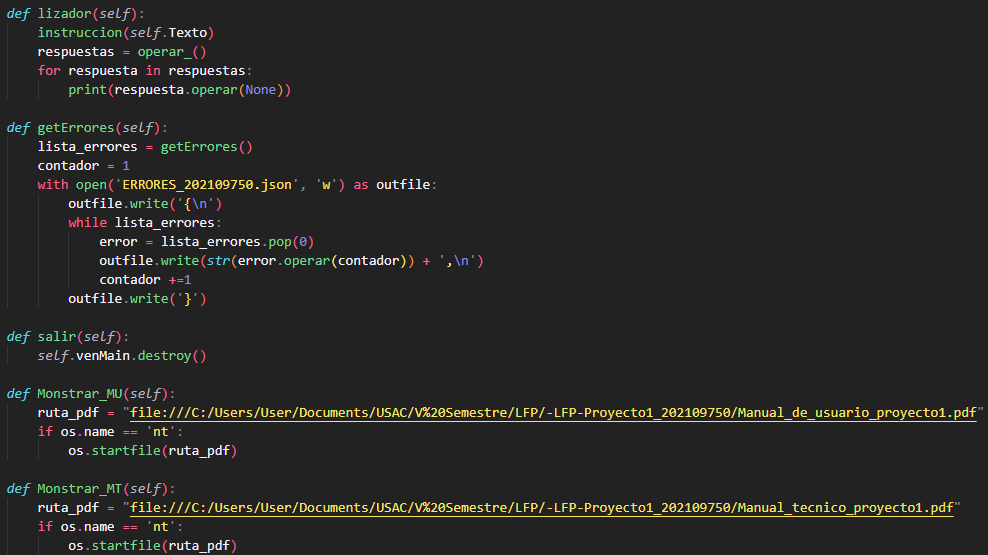
Pan1 creará dos frames distintos en los cuales irán los botones de archivo y ayuda, gracias a esta separación la interfaz se verá mejor la interfaz, arriba habrá un label en cada frame que indicará qué clase de acción mostrará, cada una tendrá sus botones requeridos en el proyecto, además habrá un Text en el cual se ingresará el texto de la carga inicial, se podrá modificar y ese será el texto a analizar.



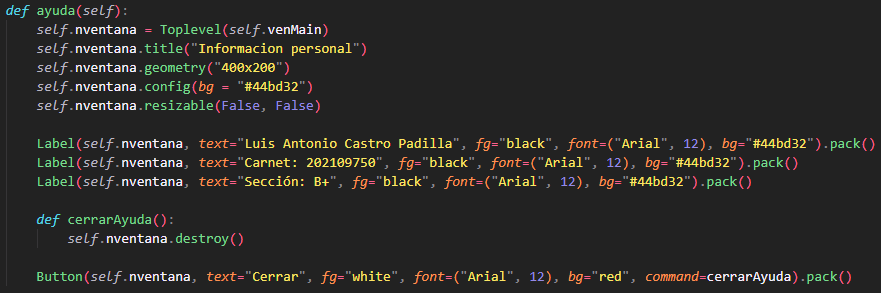
La función de abrirArchivo utilizará la opción askopenfilename la cual permite abrir el explorador de archivos para poder realizar la carga de archivos, lo lee y posteriormente lo guarda en la variable self.filename, a txt\_data (la caja de texto) se le insertará lo que se ingrese en el archivo de entrada; la función guardar como utiliza la opción asksaveasfile que abre el explorador de archivos y permite crear un nuevo doc con otro nombre; la función guardar recibe el mismo nombre pero sobrescribe la información que se cambió.



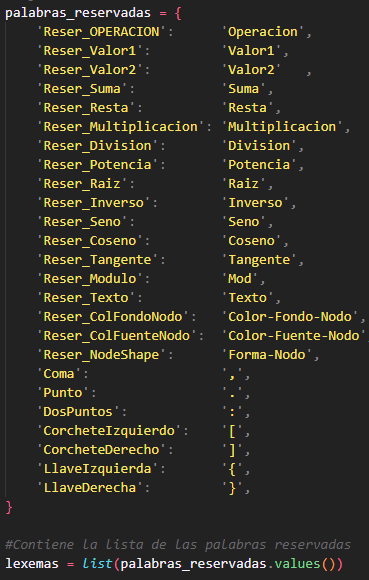
La función lizador llama a la función instrucción de la clase analizador y le ingresa la información que esté actualmente en la caja de texto, crea una variable que va a llamar a la función operar\_() y en un ciclo for imprimirá las respuestas de la función operar; getErrores llamará a la función getErrores del archivo de analizador y le declarará como lista\_errores, además usará un contador para poder incrementar la cantidad de errores que existen en el programa, abrirá un archivo llamado “ERRORES\_202109750.json” en formato de leer y modificar texto y lo mostrará como la variable outfile, por último dentro de outfile escribirá una llave de inicio y un salto de línea, se utilizará un ciclo while para recorrer todos los errores que el programa encontró en la lista de errores, por último escribirá con el formato definido en la clase operar (en la cual se usó abstracción) en función del contador, hará un salto de línea, sumará el contador y por último terminará escribiendo la llave de salida; salir cierra la venta principal; Mostrar\_MU y Mostrar\_MT abrirán la ruta especificada en ruta\_pdf, es distinta para cada manual pues son distintos archivos.



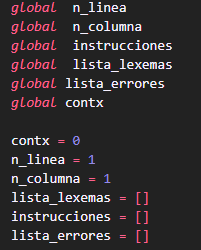
Por último, la función Ayuda creará una nueva ventana arriba de la ventana principal, al igual que la principal tendrá un título, un tamaño y fondo de pantalla específicos y además tendrá la misma característica de que no se podrá modificar su tamaño, esta solamente contiene 3 Labels y un botón, el cual tiene como comando una función llamada cerrarAyuda lo cual permite cerrar esta nueva ventana, el fondo de la nueva venta será verde, los datos serán el nombre del estudiante, el carnet del mismo y la sección del curso.



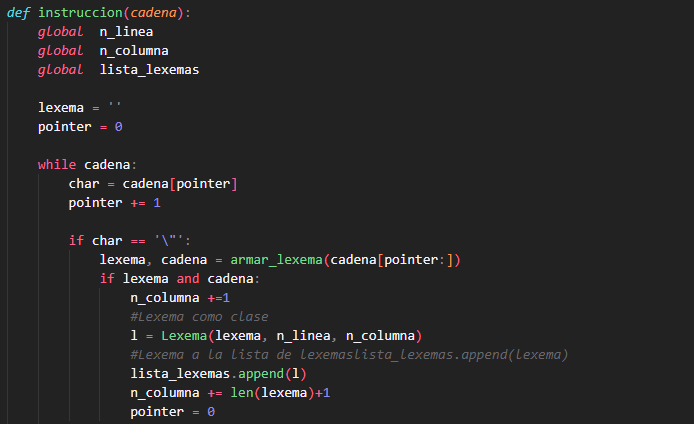
Dentro del archivo analizador.py se creará una lista de palabras reservadas, las cuales tomará el programa si es que las lee en el archivo de entrada, se agregan a la lista de lexemas.



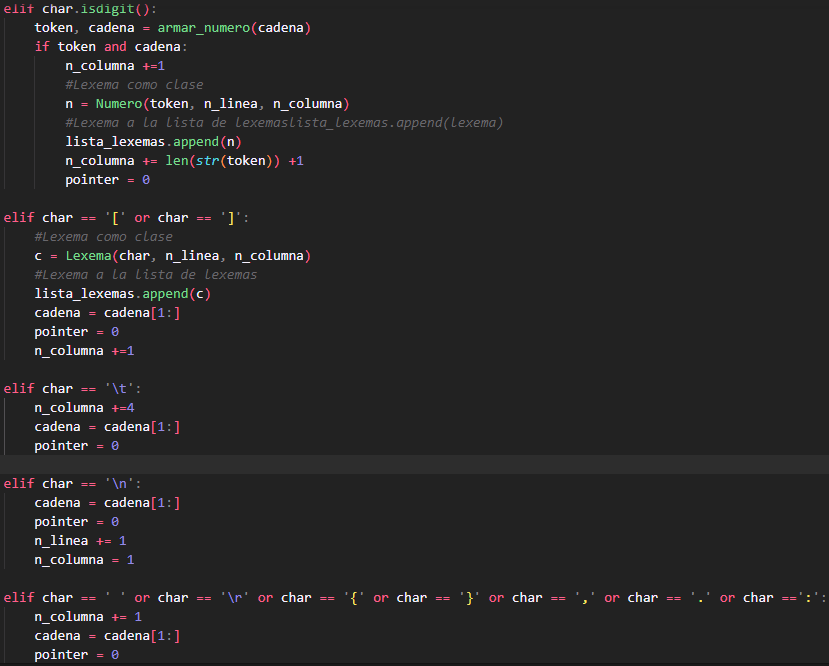
Se declaran las siguientes variables globales y se las define posteriormente.



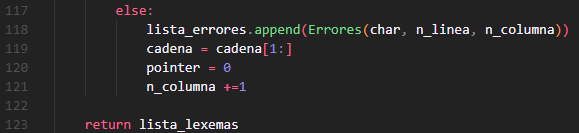
En la función instrucción se recibirá la cadena de texto de la caja de la interfaz gráfica, se declarará la variable pointer que irá recorriendo la cadena, si la cadena lee una comilla doble tomará en cuenta que es una palabra reservada por lo tanto creará un lexema como objeto y lo añadirá a la lista de lexemas, de tal manera irá registrando todas las operaciones que pida el programa.



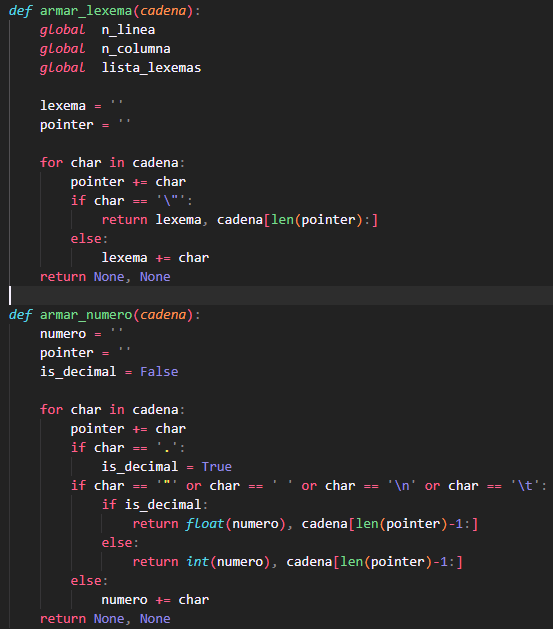
Siguiendo instrucción se verificará si la cadena es un número, en caso de ser así se creará un número con los datos que vaya ingresando la cadena, creará el objeto del número y lo añadirá a la lista de lexemas; si lee un inicio o final de corchete tomará en cuenta que hay una operación dentro de un valor de la primera operación, por lo cual también creará un lexema distinto para cada operación; si lee una tabulación incrementa el tamaño de la columna en 4 y si reconoce un salto de línea reiniciará el tamaño de la columna y añadirá en uno el valor de la fila; si lee un espacio vacío, llaves, punto, coma o dos puntos solamente añadirá un dato a la columna.



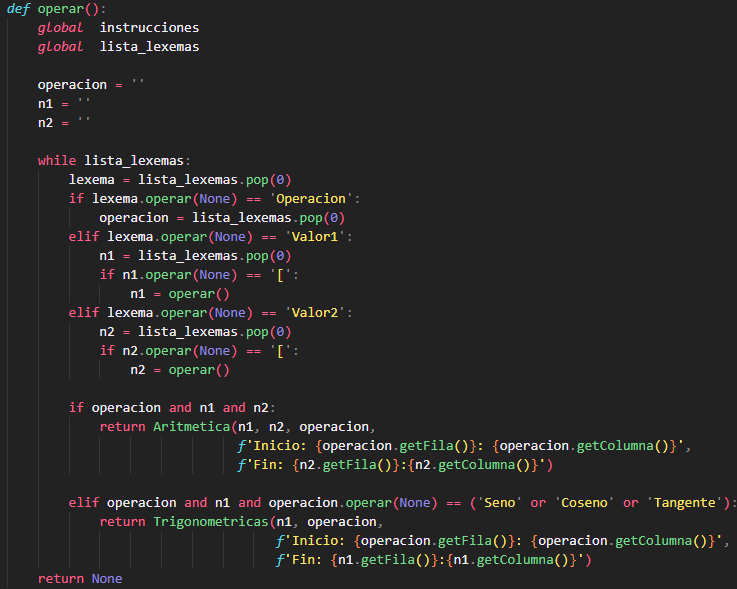
Finalmente, si no es ninguna de las anteriores lo tomará como error, en la lista de errores lo añadirá como objeto, tomando el carácter, su fila y columna.



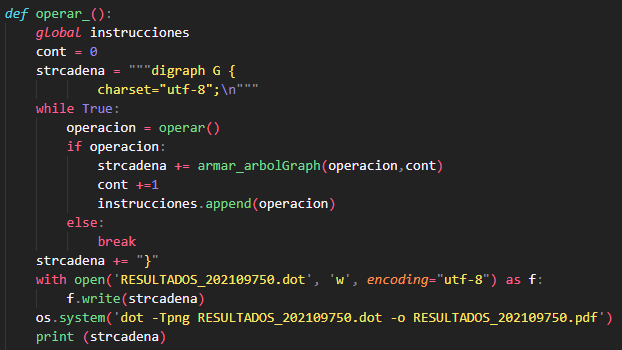
Con el siguiente código es que se crean los lexemas y números de la función instrucción.



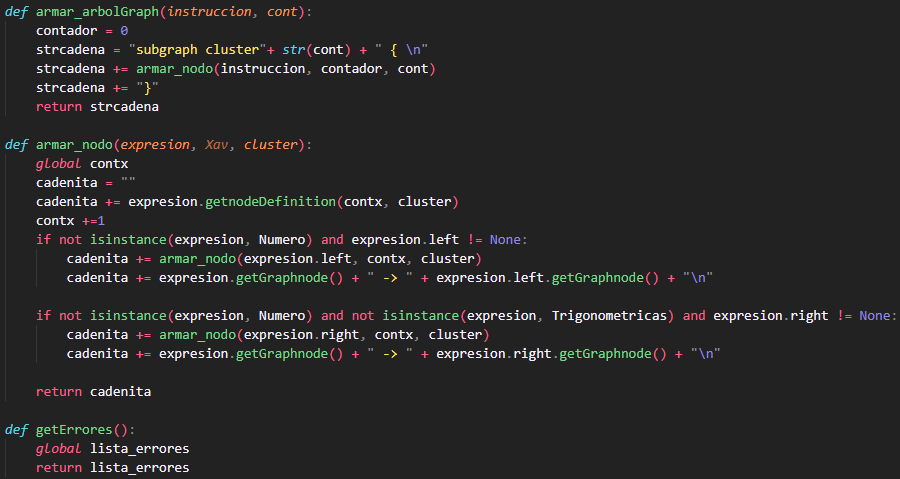
La función operar tendrá la característica que es recursiva, es decir que si hay se requiere de usar una operación en uno de los valores volverá a llamar a la función operar y ahí volverá a crear el árbol con los nuevos datos que se crearon en el lexema anterior.



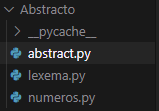
La función operar\_ creará el grafo y además usará la función operar para crear las instrucciones con las cuales trabajará el programa su análisis.



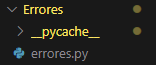
Con armar árbol se crearán los subgrafos en donde se almacenarán los nodos del graphviz, con armar nodo se revisa la definición del nodo que tiene como atributos el contador y el cluster, si es el valor 1 y no es una instancia entonces creará un nodo con el valor 1, el contador del cluster y el contador normal, entonces creará la relación con la cual se podrá crear el grafo, si es el valor 2 (o derecho) debe verificar que tampoco sea trigonométrica (pues no tiene valor 2), la función getErrores es sólo un trampolín que simplifica los errores para que sea más sencillo usarlo en el menú.



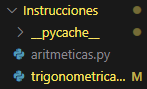
Dentro de la carpeta Abstracto se guardarán los archivos que usan abstracción para funcionar.



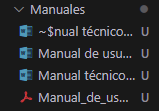
Dentro de la carpeta Errores se guardará el archivo donde se manejarán los errores.



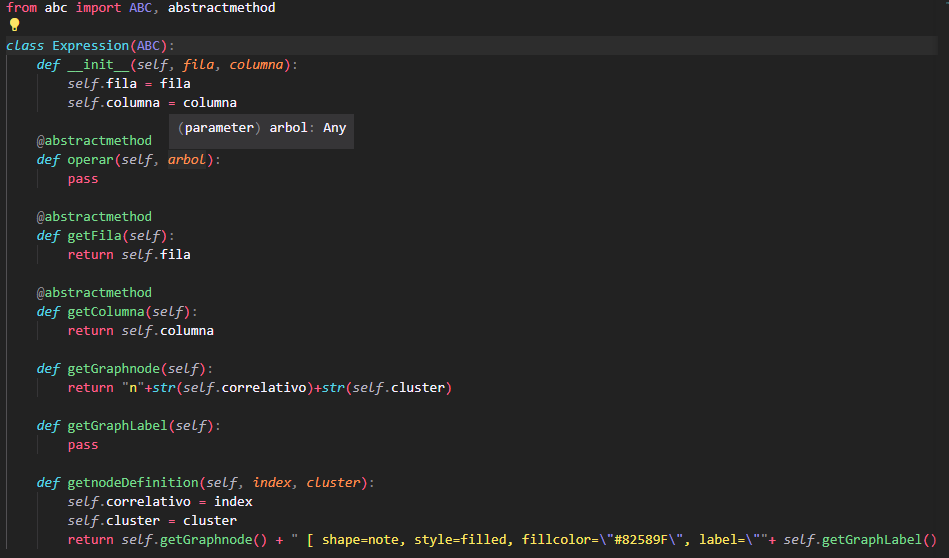
Dentro de la carpeta de Instrucciones se encontrarán las operaciones aritméticas y trigonométricas.



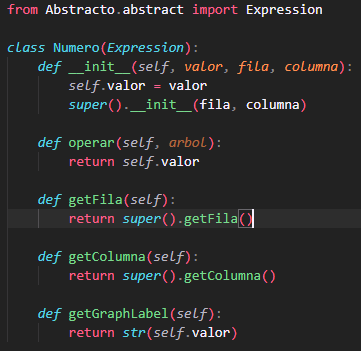
Dentro de la carpeta de los manuales estarán los manuales de usuario y el técnico.



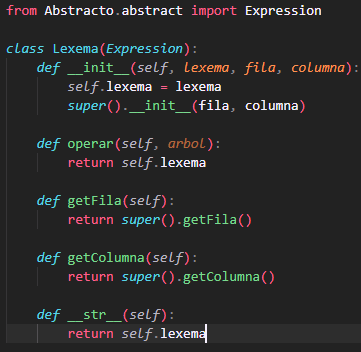
La clase Expression será abstracta, pues su función operar se podrá manejar de forma distinta para cada uno de los demás archivos que componen su carpeta y errores, pues cada una la manejará de forma distinta al igual que el getFila y getColumna, las otras 3 funciones se utilizarán para la creación del grafo, la definición del nodo creará los nodos según su correlativo y su número de cluster, el Label será abstracto también y el nodo como tal tomará el número del contador correlativo y cluster.



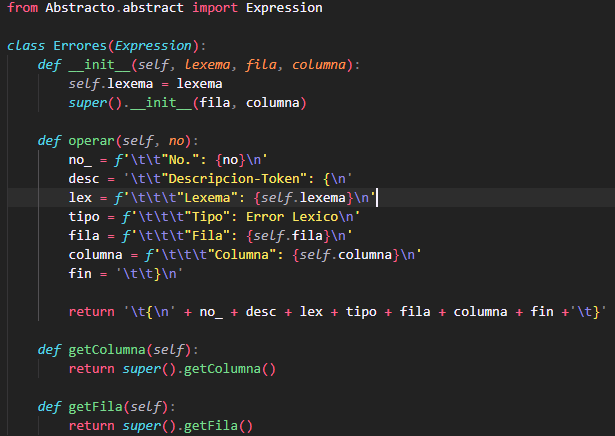
El número tomará la expresión de la clase abstracta, usará la misma fila y columna y añadirá el valor, el valor se usa en la función operar.



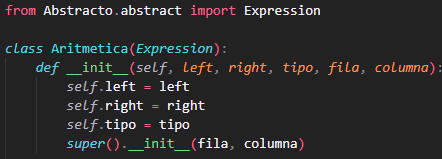
El Lexema tomará la expresión de la clase abstracta, usará la misma fila y columna y añadirá el lexema, el lexema se usa en la función operar



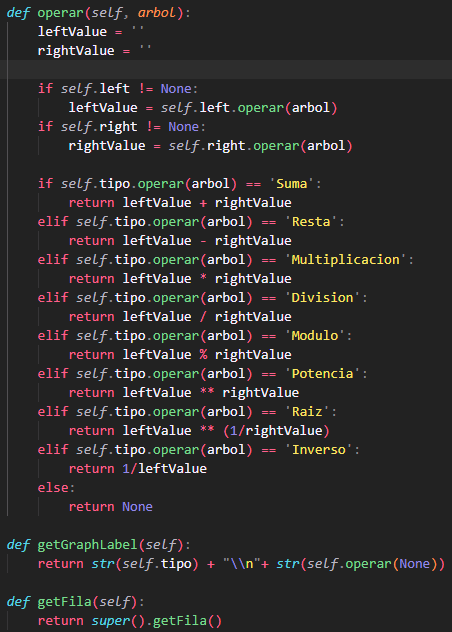
En la clase Errores se usará además de fila y columna el lexema, para la abstracción de su función operar creará una serie de cadenas de texto en donde se le da un formato json al texto de errores, tomará un contador inicial, usará el lexema que acaba de encontrar, y lo ubicará en una fila y columna.



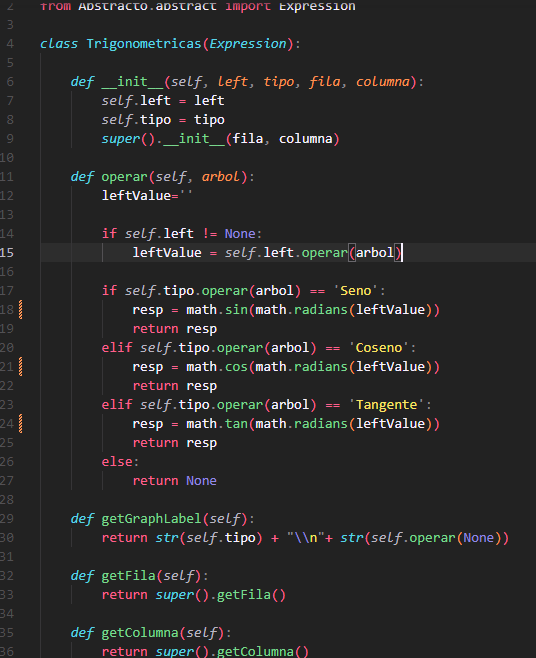
Para la clase Aritmética se usará la expresión y usará el valor izquierdo, el valor derecho, el tipo, la fila y la columna como atributos.



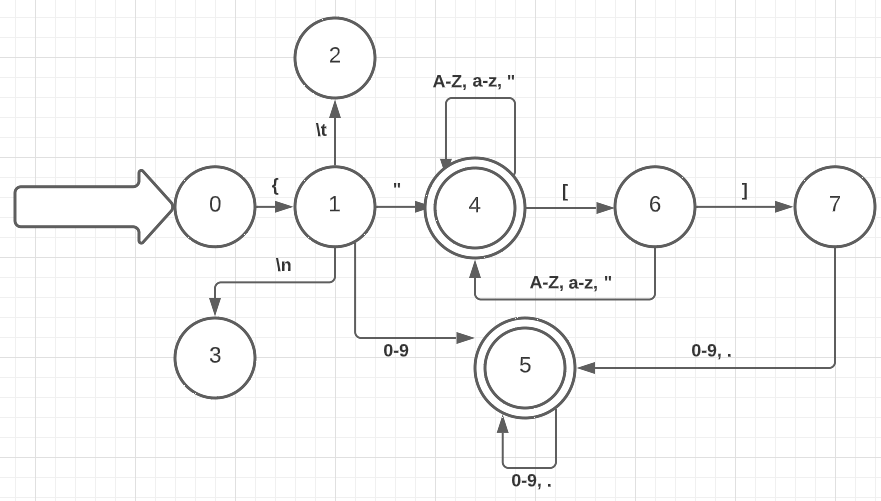
Como en la abstracción cada función puede tomarse de distinta forma lo que hace la función operar en las operaciones aritméticas y definir el valor izquierdo y derecho, con ello verifica si el tipo de operación es una de las solicitadas en el proyecto y por último retorna el resultado de la operación de ambos números. Además, llama a getFila y getColumna de la clase abstracta y usa el getGraphLabel para que en el grafo arriba aparezca cuál es el tipo de operación que se realizó.



Para las trigonométricas se toma la abstracción y se añade el importe de la librería math, toma como atributos izquierda, tipo, fila y columna, fila y columna se toman de la clase abstracta, su operación será verificar si el dato izquierdo es distinto de vacío le dará un valor y luego verificará si el tipo de operación es una de las solicitadas en el proyecto, de ser así, convertirá el dato a grados y luego hará la operación que se le solicita retornando el resultado, también se usará la abstracción del getGraphLabel para identificar qué tipo de operación es y se usará el getFila y getColumna de la clase abstracta.



AFD



Método del arbol

