

Materia:

Laboratorio de programación “ETN307”

Tema:

Algoritmos en C++ y PSeInt Vol. 1

Nombre:

Univ. Mijahel Alexander

Ruelas Machicado

Laboratorio #1

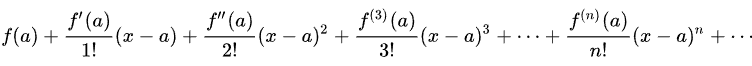
Algoritmos en C++ y PSeInt

Objetivos. –

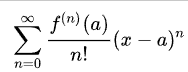
Implementar el uso de algoritmos en el lenguaje de Dev-C++ y de PSeInt para la resolución de los siguientes problemas matemáticos tipo:

* Series de Taylor y Maclaurin
* Ecuaciones de 2do grado
* Ecuaciones de 3er grado
* **SERIES DE TAYLOR Y MACLAURIN**

Marco teórico. –

La serie de Taylor de una [función](https://es.wikipedia.org/wiki/Funci%C3%B3n_(matem%C3%A1ticas)) [real](https://es.wikipedia.org/wiki/An%C3%A1lisis_real) o [compleja](https://es.wikipedia.org/wiki/An%C3%A1lisis_complejo) {\displaystyle f(x)}f(x) [infinitamente diferenciable](https://es.wikipedia.org/wiki/Funci%C3%B3n_continuamente_diferenciable) en el [entorno](https://es.wikipedia.org/wiki/Entorno_(matem%C3%A1tica)) de un número [real](https://es.wikipedia.org/wiki/N%C3%BAmero_real) o [complejo](https://es.wikipedia.org/wiki/N%C3%BAmero_complejo) *a* es la siguiente [serie de potencias](https://es.wikipedia.org/wiki/Serie_de_potencias):

{\displaystyle f(a)+{\frac {f'(a)}{1!}}(x-a)+{\frac {f''(a)}{2!}}(x-a)^{2}+{\frac {f^{(3)}(a)}{3!}}(x-a)^{3}+\cdots +{\frac {f^{(n)}(a)}{n!}}(x-a)^{n}+\cdots }

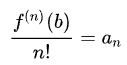
donde {\displaystyle n!} n! denota el [factorial](https://es.wikipedia.org/wiki/Factorial) de n{\displaystyle n}. Utilizando la notación sigma, lo anterior puede ser escrito de manera compacta como

donde {\displaystyle f^{(n)}(a)} denota la n{\displaystyle n}-ésima [derivada](https://es.wikipedia.org/wiki/Derivada) de {\displaystyle f} evaluada en el punto a{\displaystyle a}.

En particular, cuando a=0 {\displaystyle a=0}, la serie es denominada **Serie de Maclaurin**.

* **Función analítica.**

Si f(x){\displaystyle f(x)} está dada por una serie de potencias convergente en un disco abierto centrada en b{\displaystyle b} en el plano complejo entonces se dice que es analítica en el disco, por lo que para x{\displaystyle x} en este disco, f{\displaystyle f} está dada por la serie de potencia convergente

derivando con respecto a x{\displaystyle x} la fórmula anterior n{\displaystyle n} veces y evaluando {\displaystyle x=b}x=b obtenemos

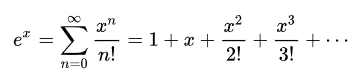
y en tal caso, la expansión en series de potencia coincide con la serie de Taylor. Por lo tanto, una función es analítica en un disco abierto centrado en b{\displaystyle b} si y sólo si su serie de Taylor converge al valor de la función en cada punto en el disco.

* **Series utilizadas en el algoritmo.**

Para este punto se dividirán en 2 tipos de series

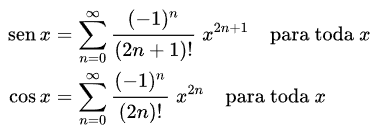
1. Convergente para toda x
2. Función exponencial

Sea la función exponencial e^x , que tiene como serie:



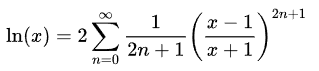
1. Funciones trigonométricas

Las funciones usadas serán las del seno y la del coseno, cuyas series son:



1. Convergente para un rango de x
2. Logaritmo natural

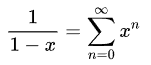
El logaritmo natural tiene cono serie de Taylor:



**Converge para 0<x<1.**

1. Función geométrica

La serie geométrica simple tiene como serie:



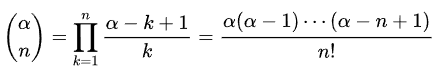
**Converge para -1<x<1.**

1. Función binomial

La serie binomial es la serie de potencias:



Cuyos coeficientes son los coeficientes binomiales generalizados



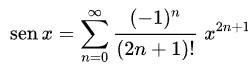
**Converge para -1<x<1 para cualquier valor de alpha.**

* Ejemplos prácticos. –
* Función exponencial

Sean los puntos de convergencia:

|  |
| --- |
|  |

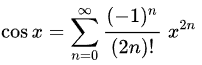
* Funciones trigonométricas
* Seno:



Sean los puntos de convergencia:

|  |
| --- |
|  |

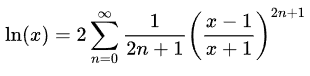
* Coseno:



Sean los puntos de convergencia:

|  |
| --- |
|  |

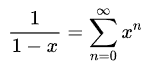
* Logaritmo natural



Sean los puntos de convergencia ( 0 < z\_0 < 1 )

|  |
| --- |
|  |

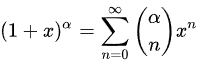
* Función geométrica



Sean los puntos de convergencia ( 0 < z\_0 < 1 )

|  |
| --- |
|  |

* Función binomial

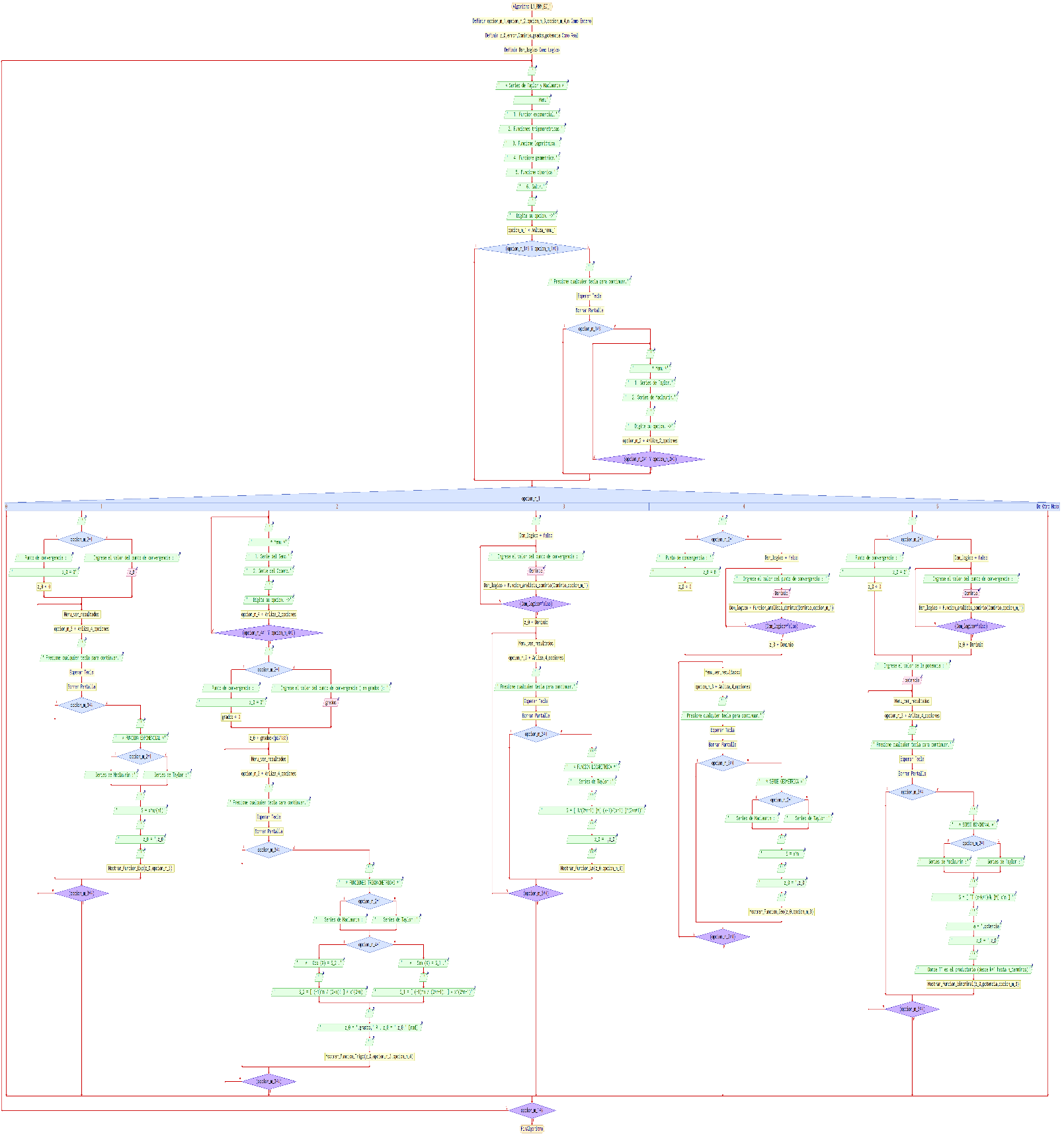


Sean los puntos de convergencia ( -1 < z\_0 < 1 )

|  |
| --- |
| α = {2 , 4 , 7} |

Diagramas de flujo. –

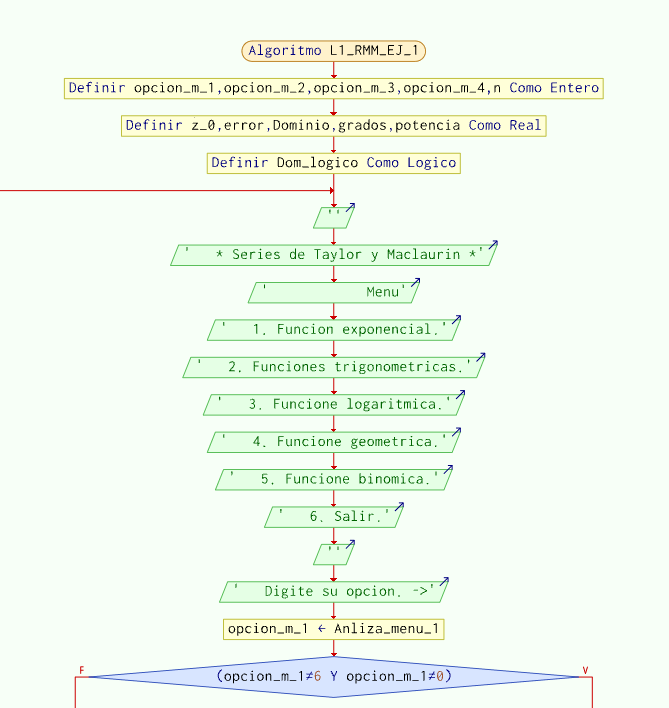
Para dibujar los diagramas de flujo de los algoritmos se empleó el programa “PSeInt”, y como estos están conformados por funciones se muestran de la siguiente manera; Además para poder visualizar de manera detallada se realizarán distintos acercamientos a cada diagrama de flujo.

* Series de Taylor y Maclaurin (L1\_RMM\_EJ\_1 es el nombre del archivo)
* Diagrama principal

Zoom programa principal

* Inicio del algoritmo: el algoritmo se llama L1\_RMM\_EJ\_1

* Definir variables: se observa las de tipo entero que servirán para el manejo de las opciones del menú y las del tipo real que constarían de valores decimales, estos se utilizaron para el desarrollo matemático del programa y finalmente una variable del tipo lógico que se usó para una sección en la que existe un radio del disco de convergencia, estos se verán en las series del logaritmo natural, serie geométrica y la serie binominal.
* Se inicia un ciclo do{}while() que termina si y solo si el usuario digita la opción 6
* Impresión en la pantalla: Muestra el menú de inicio y pide ingresar su opción.
* Analiza\_menu\_1: Es un subprograma que evita que el usuario ingrese una opción errónea

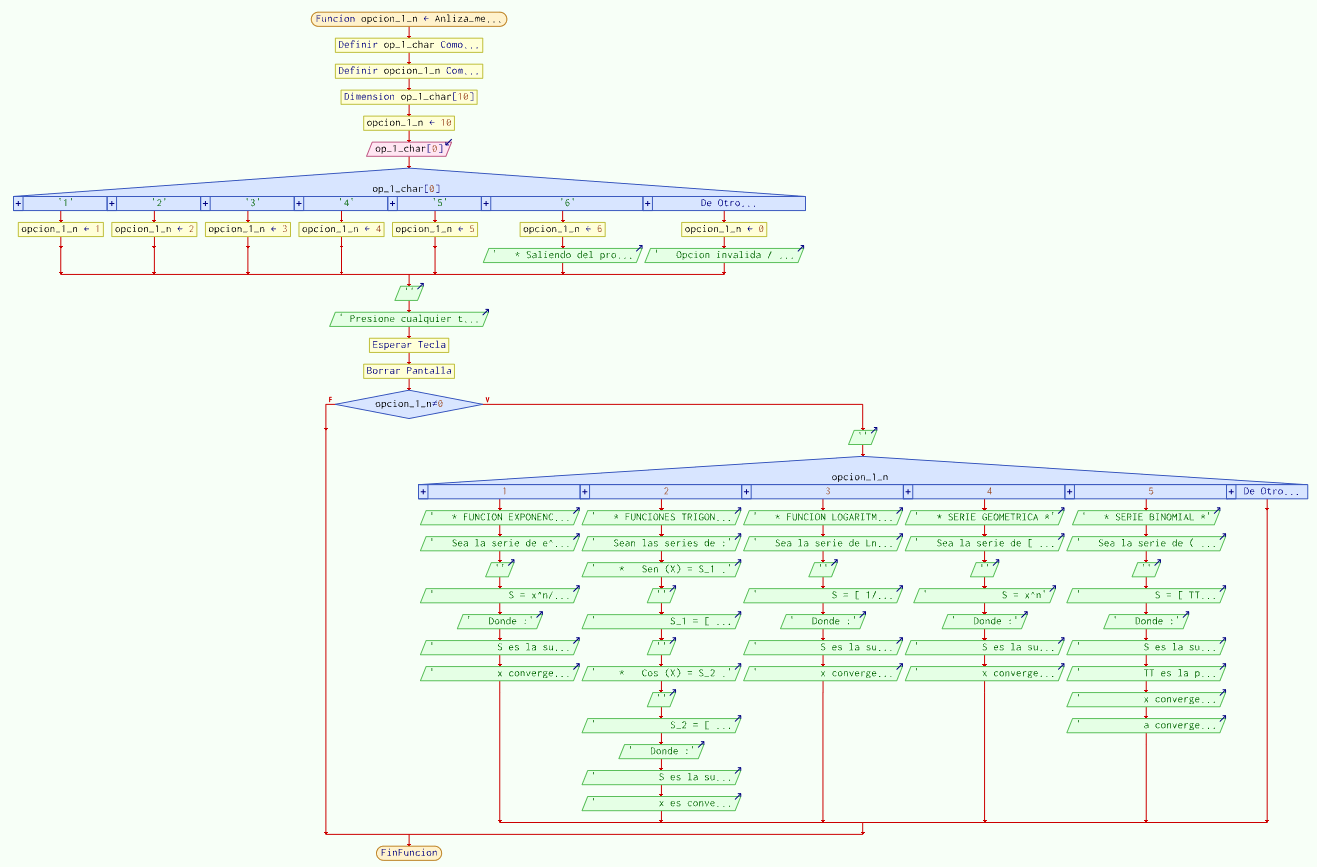


1. Subprograma Analiza\_menu\_1

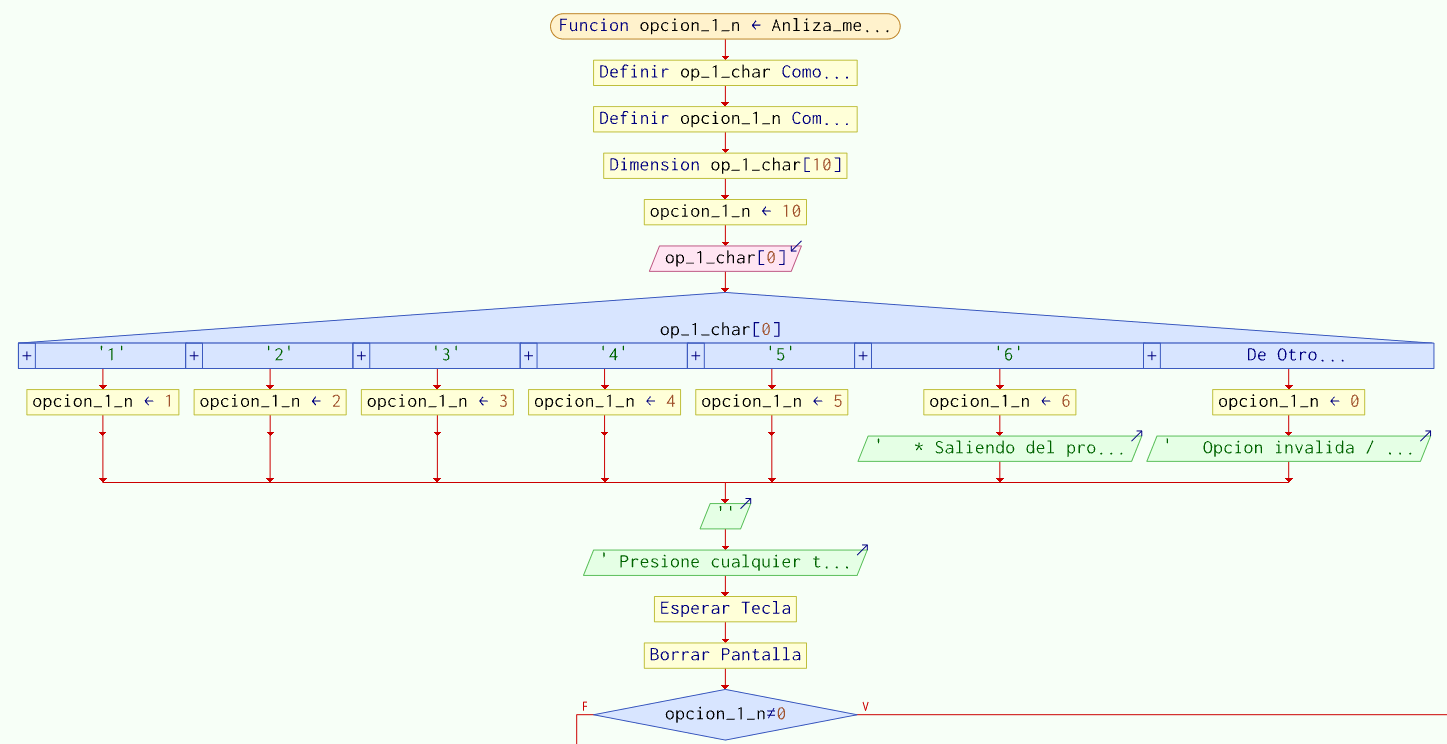
* **Para que sirve.** -El usuario digitará algún valor, y esta será almacenada en el carácter opción\_m\_1(se creó dicho carácter para que el usuario ingrese cualquier letra, numero o simplemente un espacio), esto con el propósito de evaluar el valor ingresado en una condicional switch, donde se verá que para los valores validos (existentes en el menú) tomen un valor numérico entero en la variable opción\_1\_n (esto se apreciara y detallara más adelante), sin embargo si la opción ingresada es invalida entrara en el bloque de DEFAULT, y por consiguiente opción\_n\_1 tomara el valor de cero además que en la pantalla se imprimirá Valor invalido/Inténtelo nuevamente; También se debe aclarar que para la opción de 6 (la opción de salir del programa esto en el menú), es imprime \* Saliendo del programa \*.

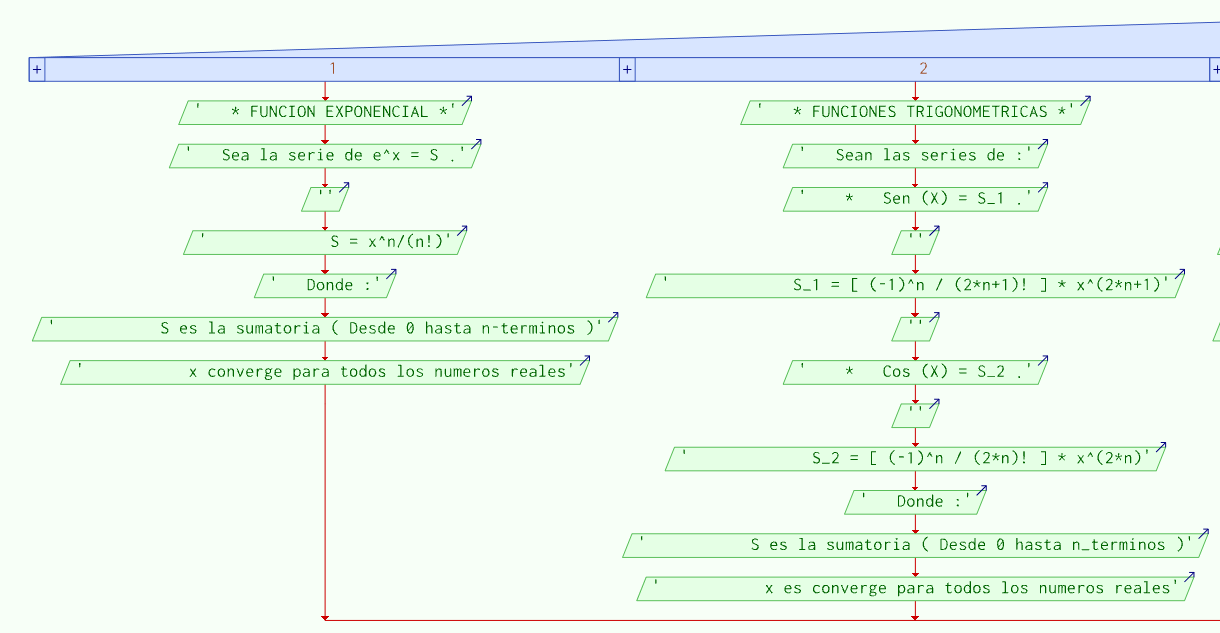
Luego de evaluar y asignar un valor ya sea erróneo o valido a la variable opción\_1\_n, es imprime el texto de Presione cualquier tecla para continuar, al efectuar dicha acción se procede a limpiar todo lo mostrado en pantalla anteriormente.

En la 2da mitad se ingresa el valor guardado de opción\_1\_n para ver si cumple o no con la condicional if() que evaluara dicho valor, si el valor fuera ‘0’ finalizaría el subprograma, caso contrario si este fuera diferente de ‘0’ ingresara a una condicional switch(esta dependiente del valor que tiene opción\_n\_1) que mostrara en la pantalla un breve resumen de la serie que el usuario digito, finalmente luego de realizar dicha acción el subprograma termina.



Zoom\_sub\_programa\_Analiza\_menu\_1

* El subprograma se llama Analiza\_menu\_1
* Es necesario definir nuevas variables que solo trabajan dentro de dicho subprograma
* Es necesario aclarar que esta función o subprograma, es del tipo int en lenguaje de C++, es decir que cuando finalice enviara un valor al programa principal (En este caso en particular el valor servirá para activar las tareas y subprogramas que el usuario solicito).

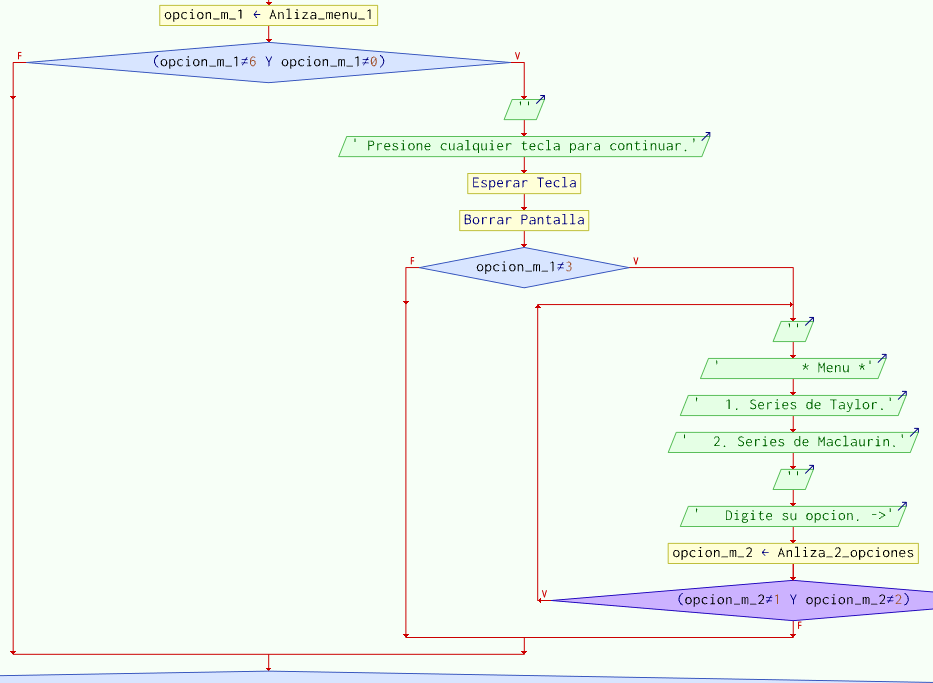
zoom a la 2ª condicional switch () si el valor de opción\_1 igual a la unidad o a 2.

Retornando a la función principal:

Se ve que al valor retornado del subprograma Analiza\_menu\_1 es guardado en la variable opción\_1\_m y a continuación esta es evaluada en una condicional if(), si el usuario digitara la opción de salir o cometiera el error de ingresar un valor no valido el algoritmo pasara directamente a la siguiente sección del programa, la cual se trata de una condicional switch(); Sin embargo si el valor digitado por el usuario es correcto, el algoritmo imprimirá Presione cualquier tecla para continuar a continuación se procede a limpiar la pantalla para luego ingresar a un if().

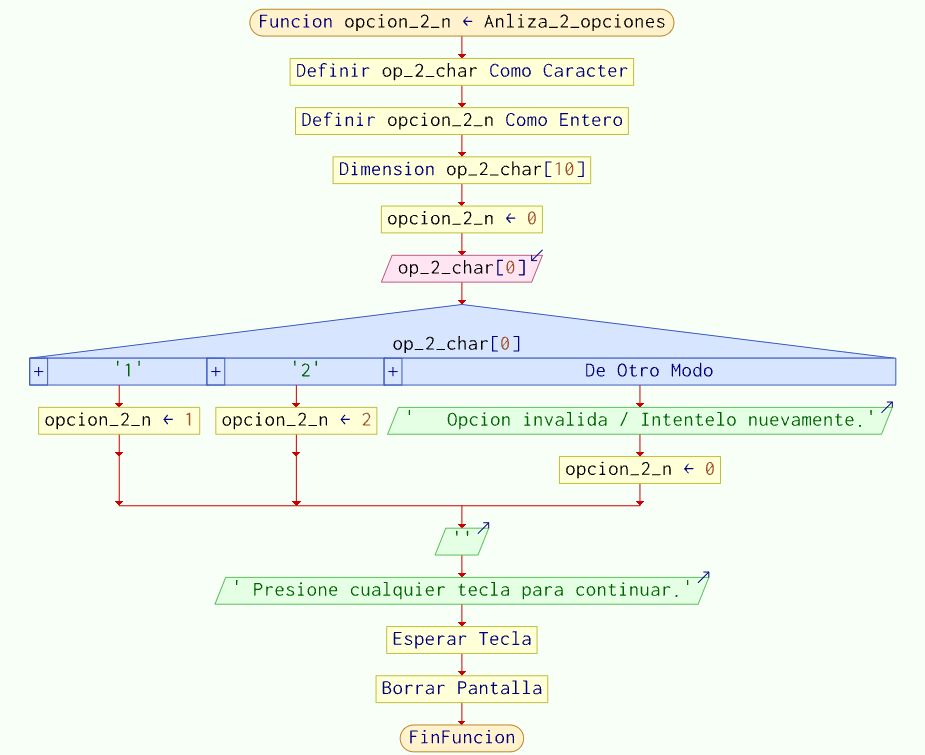
Se creó esta condicional if() para evitar mostrar el menú 2 que se ve en la imagen, el cual propone entre 2 opciones al usuario, elegir si desea preceder con una serie de Taylor o una de Maclaurin, sin embargo para la opción 3 es decir para la Función del Logaritmo natural realizar una serie de Maclaurin (punto de convergencia igual a cero) sería imposible. Continuando con el análisis de la 2da condicional de la imagen, si resulta opción\_m\_1 distinta de cero como verdadero, se procede a mostrar en la pantalla el 2do menú que dará a elegir al usuario entre 2 opciones para luego ingresar a un subprograma llamado Analiza\_2\_opciones.

Cuando el subprograma Analiza\_2\_opcion finaliza y entrega un valor, el cual es almacenado en la variable tipo int llamada opción\_m\_2, la cual será analizada por el ciclo do{}while() de la imagen, dicha condicional repetirá el proceso hasta que el usuario ingrese un valor correcto del menú.

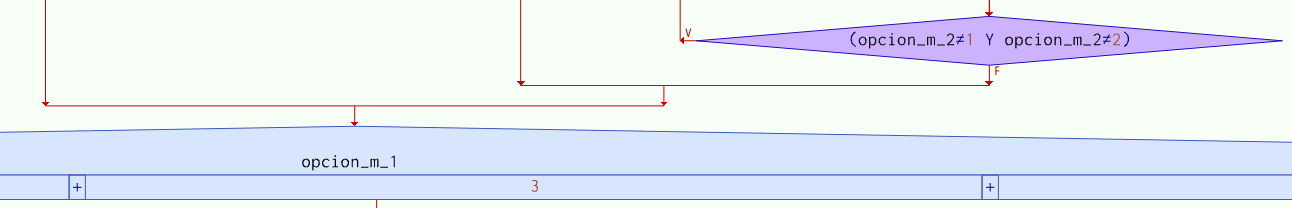


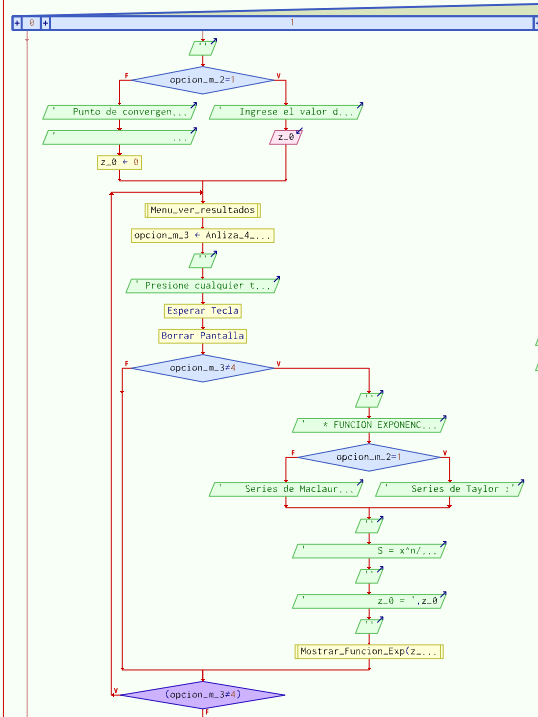
Subprograma Analiza\_2\_ociones

**Para que sirve.** –Similar al subprograma Analiza\_menu\_1, este realiza la tarea de evitar que el usuario ingrese valores erróneos al momento de escoger entre las opciones del menú 2.

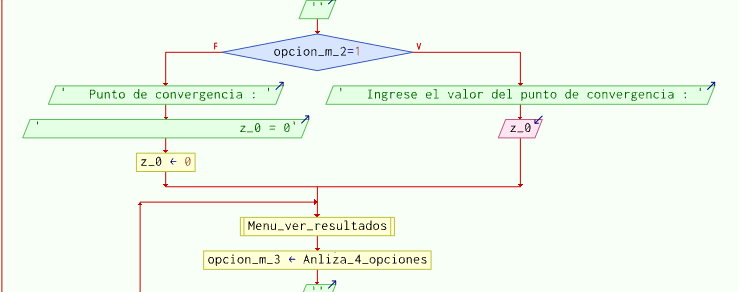
* Esta función es del tipo int, retornando un valor numérico entero cuando termina de realizar su tarea, este valor numérico que servirá para los procesos posteriores del algoritmo.

Retornando a la función principal:

 El algoritmo ingresa a una condicional switch que será visto por partes puesto que este realizará las tareas para cada serie ya sea exponencial, geométrica, etc.

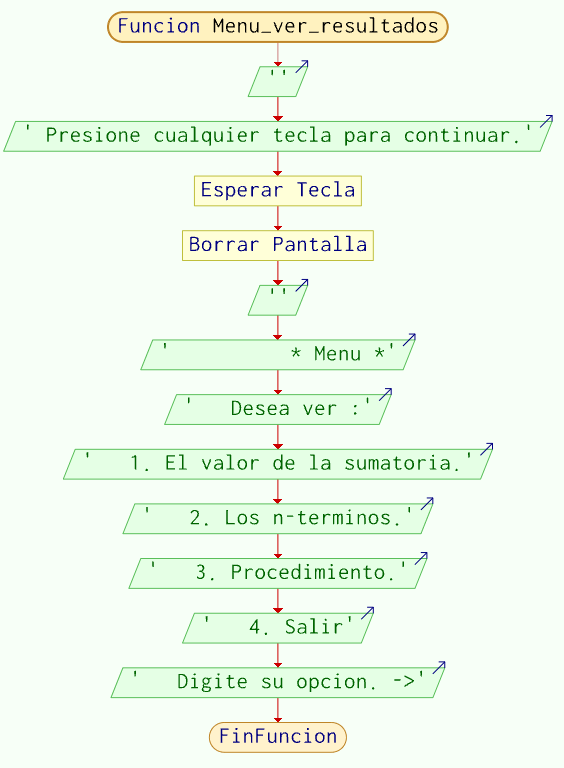
* Si opción\_m\_1 tiene los valores de 0 o 1.

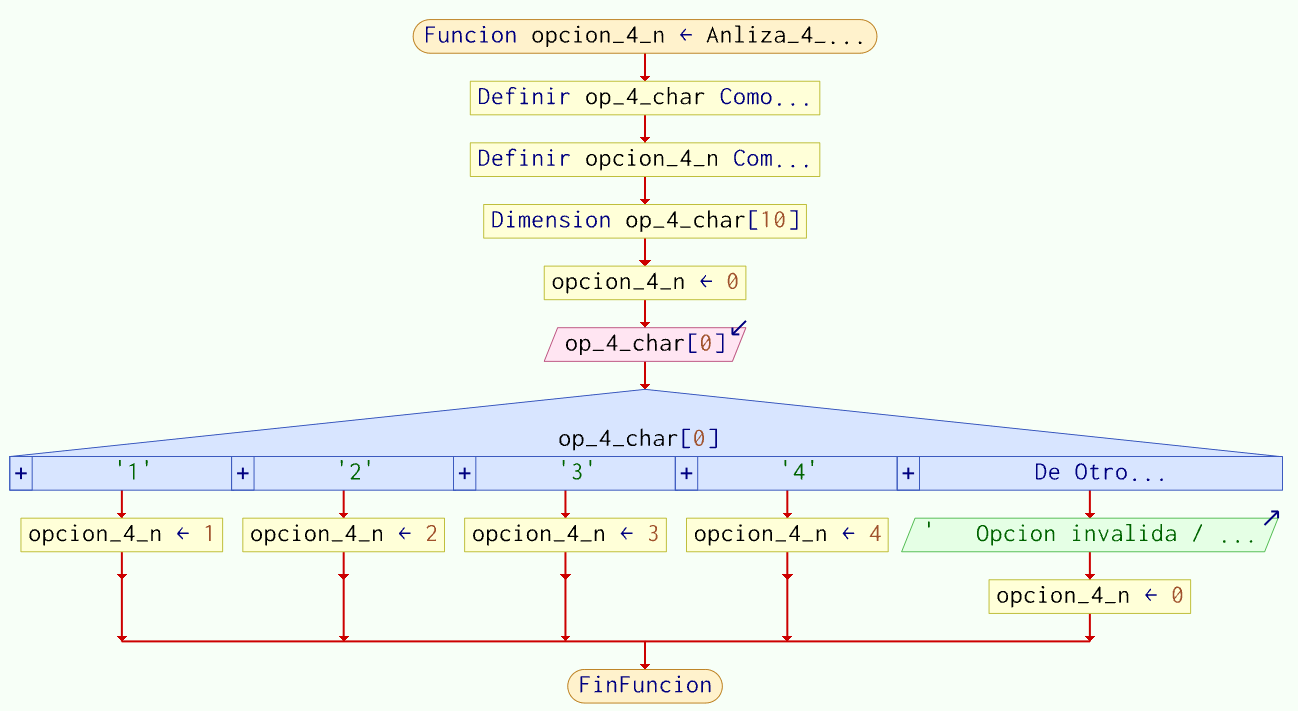
Para el caso de opción\_m\_1 fuera igual a cero llegaría a dar por terminado la condicional switch sin realizar ninguna acción; sin embargo, si el valor fuera igual a la unidad finalmente empecería a procesarse las tareas necesarias para realizar la serie exponencial, como se explicará a continuación.

Bloque de tareas para la Serie exponencial 1ra parte:

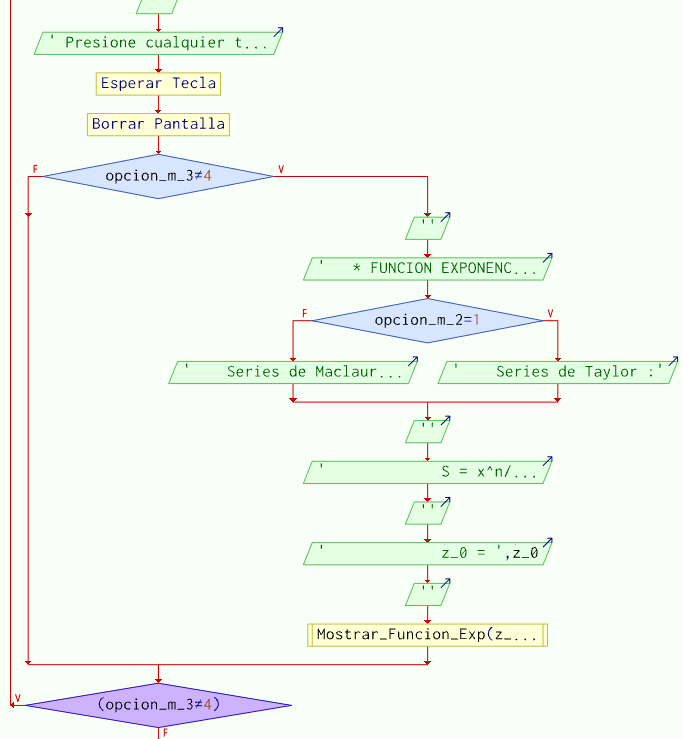
El bloque inicia con la con la condicional if() del valor de opción\_m\_2 el cual correspondía a si se deseaba una serie de Taylor o una de Maclaurin, para el caso de serie de Taylor mostrada en la pantalla que ingrese el valor del punto de convergencia para luego guardar el valor en la variable de tipo flotante z\_0; Pero si se deseara realizar una serie de Maclaurin es valor por defecto es de cero.

Luego se inicia un ciclo do{} while() que evitara que el menú que veremos continuación llegara acerrare sin que el usuario así lo deseara.

Subprograma Menu\_ver\_resultados :

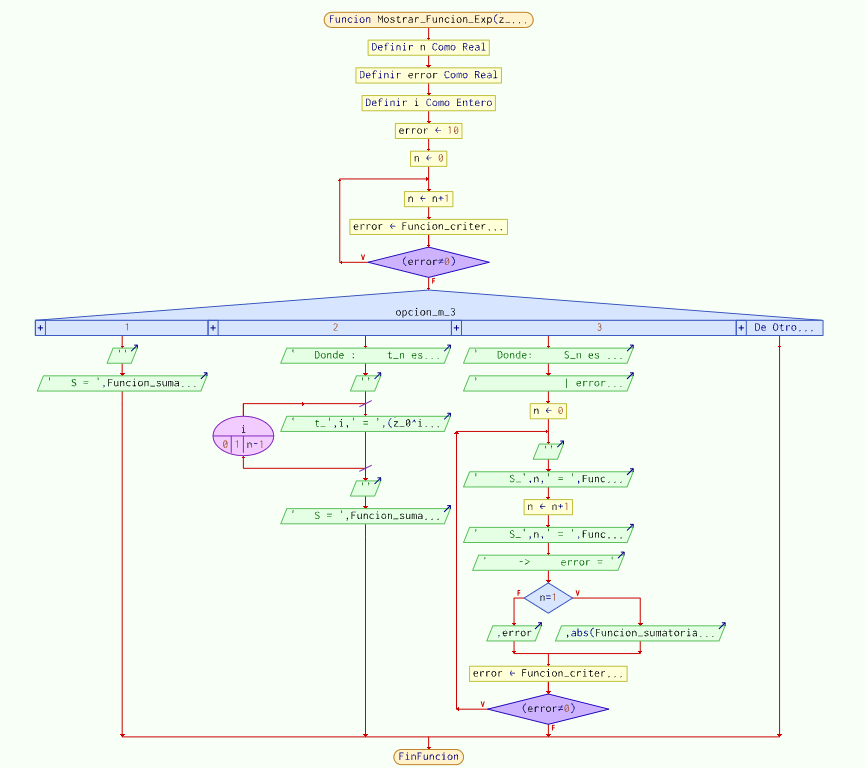
Subprograma Analiza\_4\_opciones:

* Como se aprecia en los diagramas de flujo el subprograma Menu\_ver\_resultados realizara la tarea simple de mostrar en la pantalla los mensajes del 3er menú (ver la página anterior).
* Para el caso del subprograma Analiza\_4\_opciones funciona al igual que los anteriores, es decir para que el usuario elija entre 4 opciones, si este valor ingresado es válido se convierte en un valor numerito para la variable tipo int opción\_4\_n; Pero si el valor fuera erróneo el valor de la variable opción\_4\_n será de 0 y gracias al ciclo do{}while() este proceso se repetirá hasta que se digite la opción de salir.

Bloque de tareas para la Serie exponencial 2da parte:

La segunda parte del bloque inicia borrando lo que antes se encontraba en la pantalla de la consola, luego inicia una condicional if() evaluando el valor de la opción ingresada si el valor fuera igual a 4 se saltaría a la parte final del condicional do{}while() realizado en la 1ra parte, esto visto desde la consola se traduciría en que el menú de mostrar resultados se repitiera hasta que el usuario digite el valor de la opción de salida el cual es opción\_m\_3=4.

Si la opción seleccionada fuera distinta de 4 se mostrara los mensajes como se puede ver en la imagen.

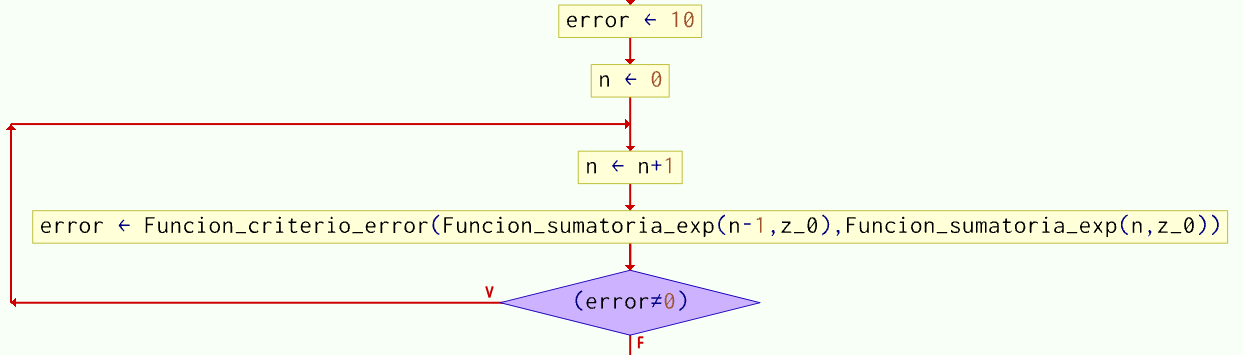


1. Subprograma Mostrar\_Funcion\_... (z\_0, opción\_m\_3)

**Para que sirve.** –Este subprograma o Función, realiza las tareas necesarias para generar los n-términos de la serie, esto gracias a un par de funciones (quizá las más importantes para este algoritmo), cabe aclarar que para las demás series es similar la aplicación de Funciones (de tipo matemáticas).

Además de realizar dichos procesos, como su nombre lo dice, la función mostrara los resultados finales que el usuario desee ver (esto entre 3 opciones).

Inicia definiendo las variables que necesitara en el transcurso de las diferentes tareas, es importante notar que 2 variables tendrán valores, la variable error tiene el valor de 10 puesto que se crea un ciclo do{}while() que generara los n-termino y este acaba cuando el error esa 0, también le otorga el valor de 0 a la variable n, esta variable nos dirá cuántos términos fueron necesarios para generar el resultado final de la sumatoria

Zoom a la parte del inicio ciclo do{}while()

Cabe aclarar que dentro del ciclo yace la función matemática Funcion\_criterio\_error, la cual es de tipo flotante (retornara un valor decimal el cual será almacenado en la variable error), sin embargo, de mayor importancia resulta la dependencia de esta variable, puesto que depende de la Funcion\_sumatoria\_exp la cual a su vez depende de la cantidad de términos “n”, y del punto de convergencia; Aunque no se vea por el momento, ambas funciones dependen de la función Funcion\_factorial.

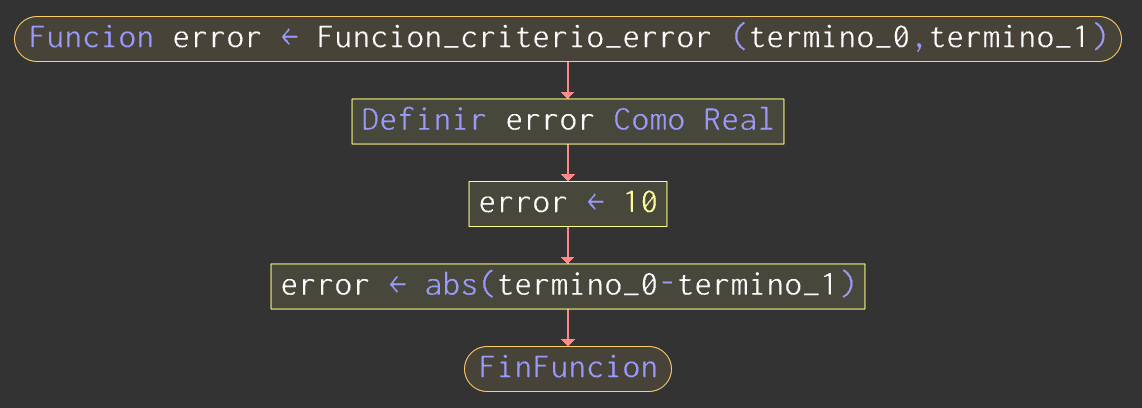
Antes de continuar será necesario ver a las funciones matemáticas dentro de este ciclo do{}while().

1. Funciones matemáticas.

Como ya se mencionó estas son las más importantes a la hora de entregar los valores finales de las series, estas son:

* **Funcion\_criterio\_error( , )**

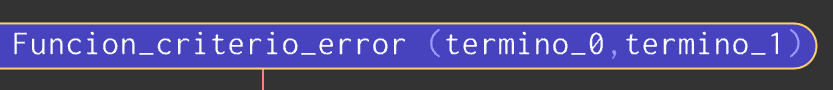
Esta evalúa de diferencia que hay entre la suma acumulada y la anterior, como también evalúa si el termino n-simo tiende a el valor nulo.



Para que esta función realice la tarea solicitada dentro de ciclo do{}while() (sin provocar algún error de programación), el contador “n” inicia antes que esta tarea de criterio de efectué para que al momento de realizar la diferencia “ya exista un termino\_0”.

Para este punto es necesario aclarar la dependencia de las Funciones de ciertas variables o´ para este caso funciones que entregaran valores; Para esto tomaremos prestado a la función desde su llamado dentro del algoritmo principal, y se la comparara con la forma en la que esta se creó:

1. Forma en la que está creada la función



Básicamente la función F\_criterio\_error fue creada para que dependa de 2 valores numéricos de tipo flotante, estos son termino\_0 y termino\_1.

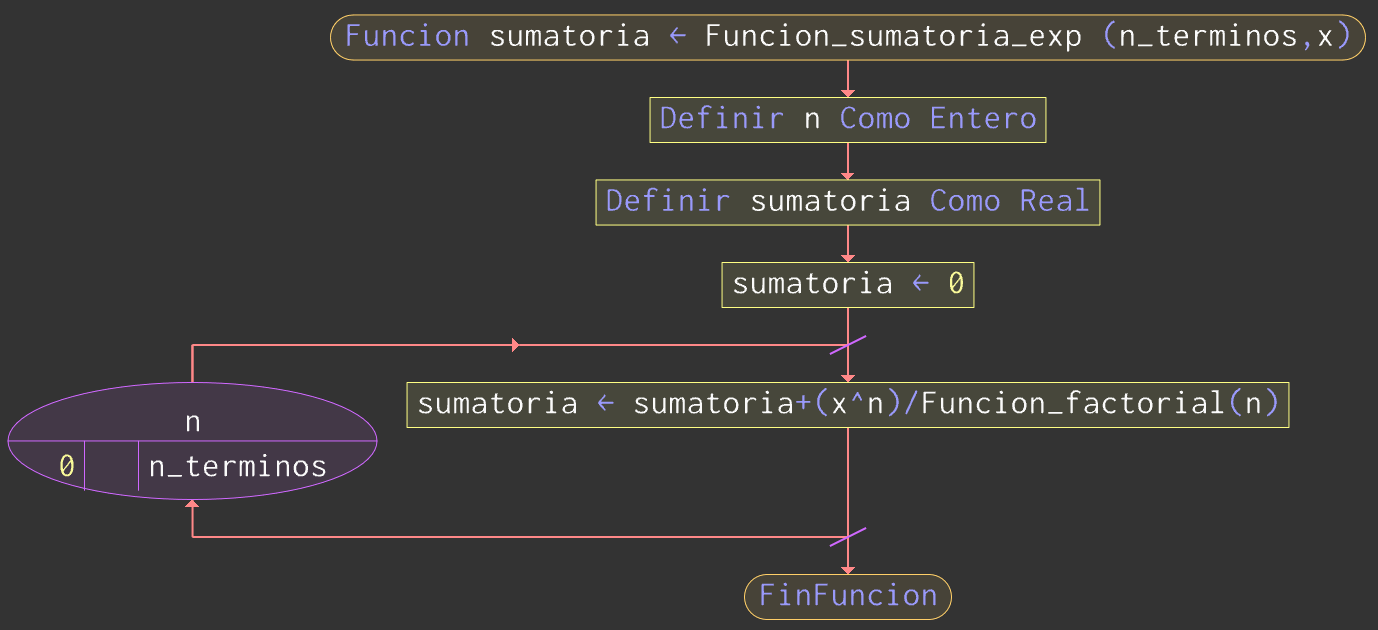
1. Llamando a la función dentro del algoritmo principal

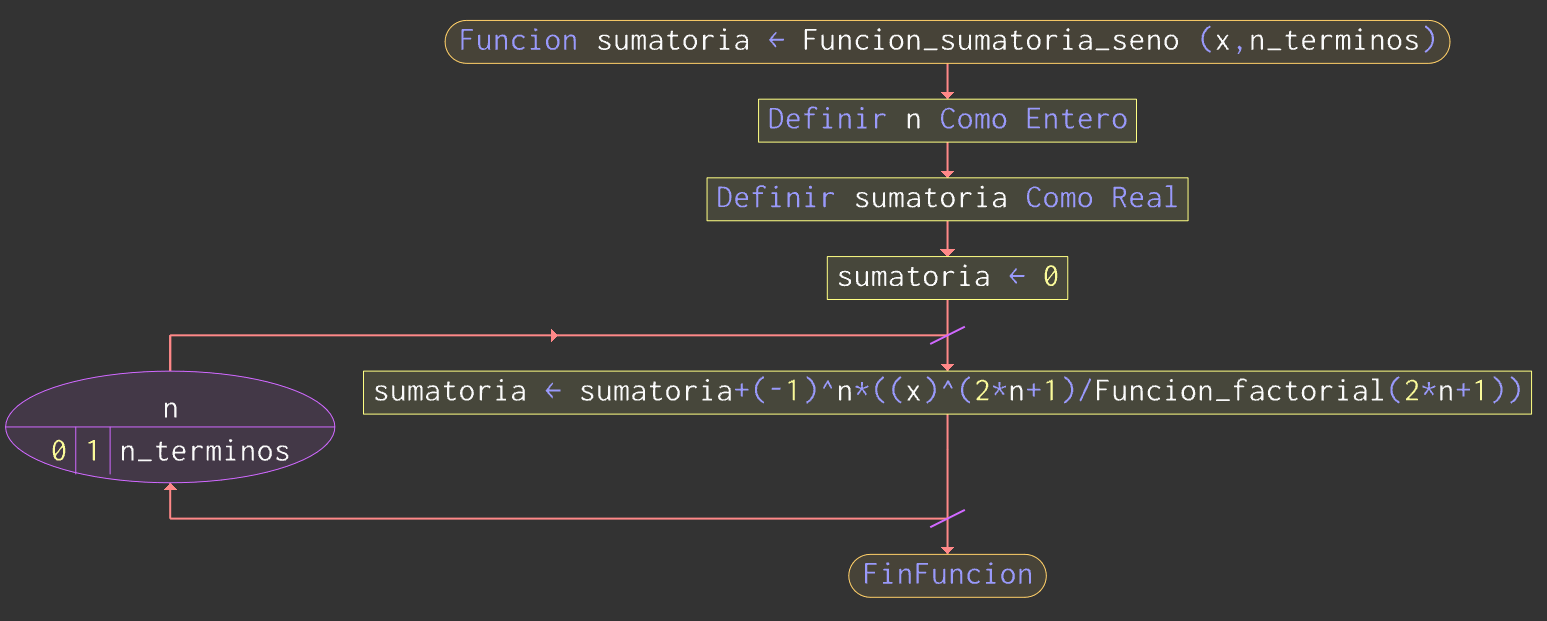
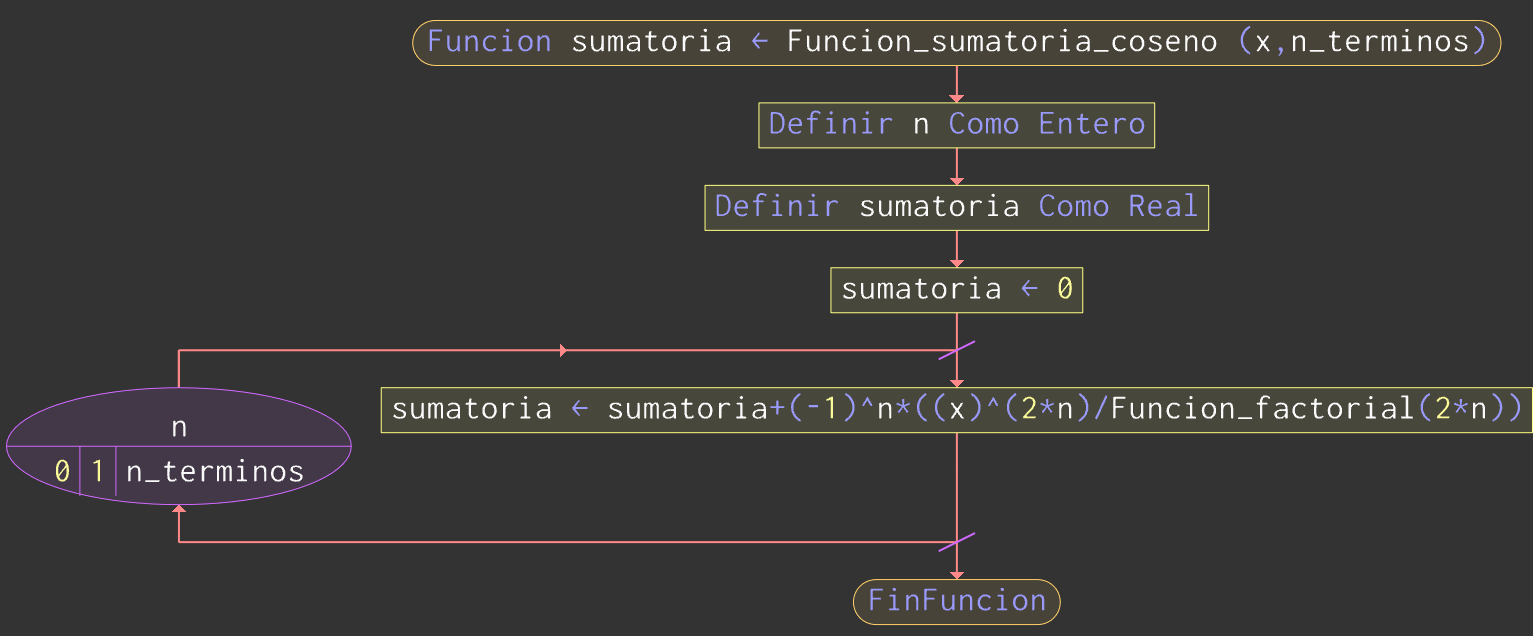


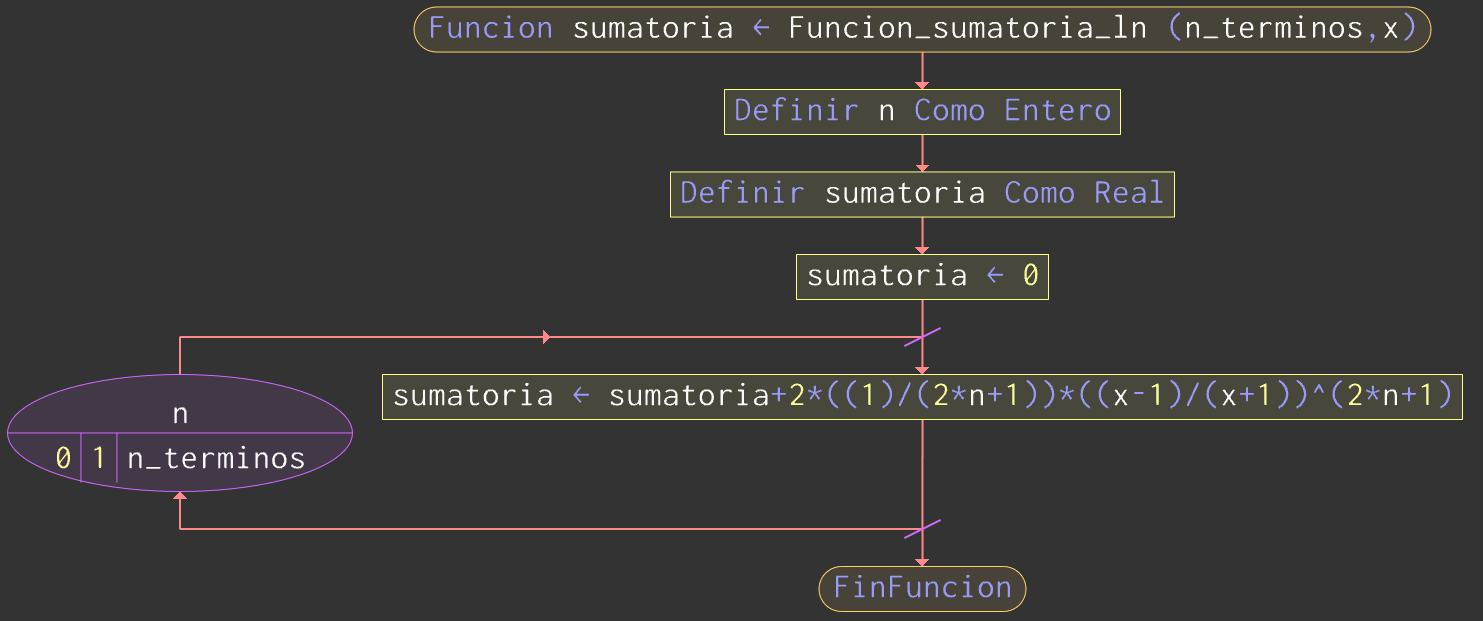
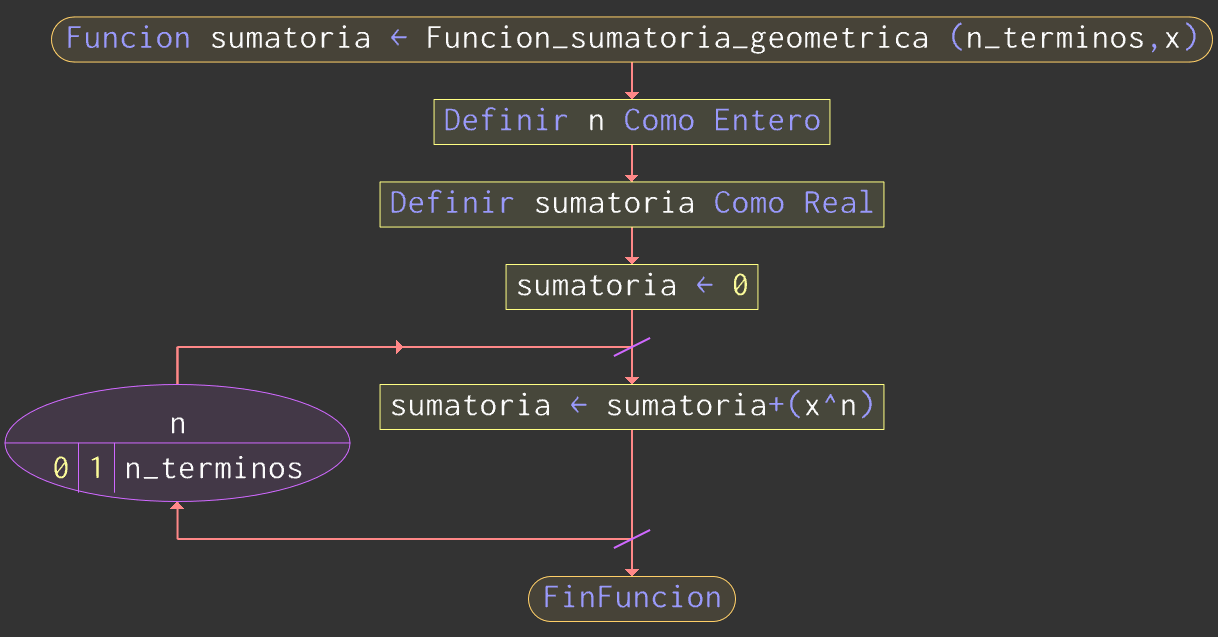
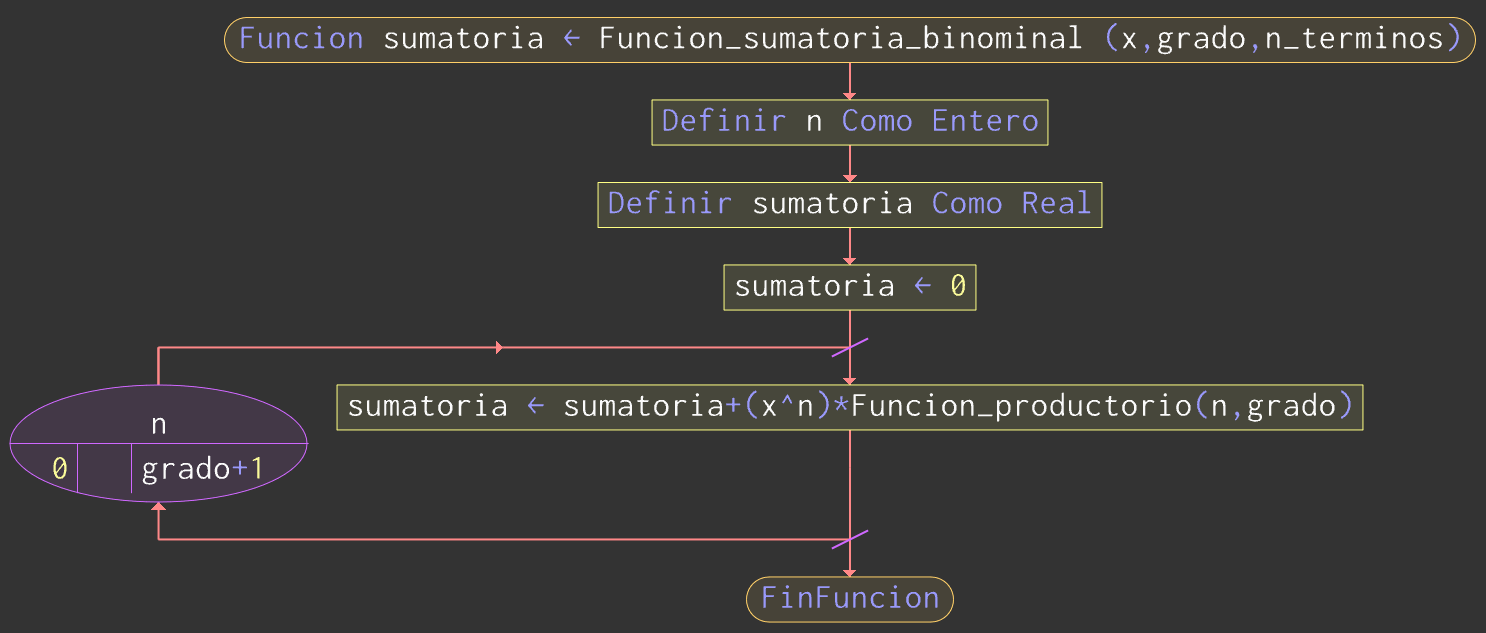
Al momento de ser convocada en el algoritmo principal, esta dependencia será de los valores que la Funcion\_sum\_exp() le entregue; Notando claramente en la imagen que a su vez esta función depende de las variables “n” y z\_0.

* **Funcion\_sumatoria\_... ( , )**
* Esta función acumula el valor de la fórmula de la serie que se desee realizar.
* Depende básicamente de la cantidad de termino y del punto de convergencia.
* Dentro de la función se definieron a n como entero ya que esta variable será utilizada en la fórmula de la serie, también se definió a una variable llamada “sumatoria” además que se le otorgo un valor de 0 para que al momento de ir acumulando el valor final de la sumatoria final este variable no tenga valores parásitos.
* Mediante un ciclo for(){}, iniciando desde 0 hasta el valor de n-términos, se irán generando los valores de los n-términos
* Dentro del ciclo for(){} ya mencionado se encuentra la fórmula matemática de la serie; Notando que en esta fórmula se encuentra la Funcion\_factorial(n).

Continuación se verán los diagramas de flujos de las diferentes series.

1. **Función serie exponencial**
2. **Funciones series trigonométricas**

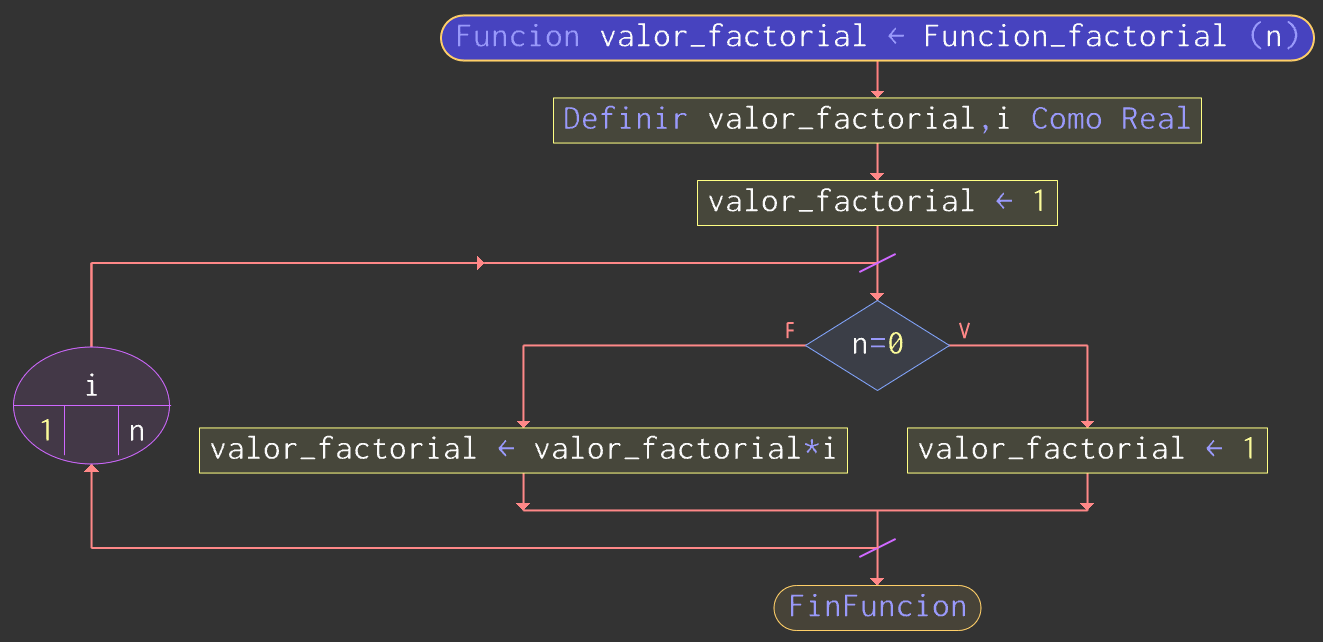
* **Función seno**
* **Función coseno**

1. **Función serie logaritmo natural**
2. **Función serie geométrica**
3. **Función serie Binominal**

Es necesario mencionar que esta última Funcion\_sumatoria\_binominal tiene dentro de su ciclo for(){} a la Funcion\_productorio(n,grado), esto debido a que la fórmula de la serie binominal requiere el uso de dicha función.

* **Funcion\_factorial ( , )**

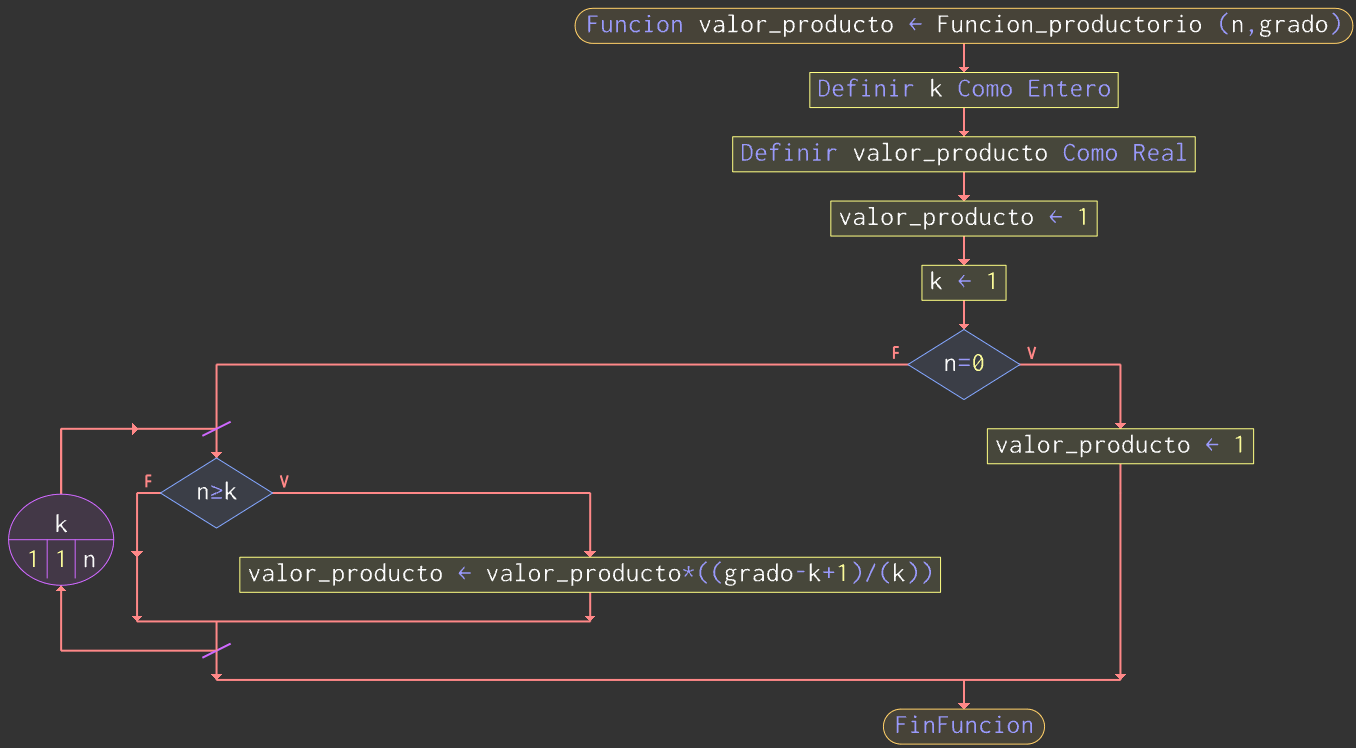
Talvez esta sea la función más recurrente al momento de ejecutar el programa final.



Depende de una variable “n” además dentro esta función se definen las variables de valor\_factorial e i ,la primera guarda el valor acumulado del factorial que además será retornada por la función y la variable i cumple como contador dentro del ciclo for.

Además dentro de dicho ciclo es encuentra una condicional if() la cual cumple con una de las propiedades del factorial, es decir 0! = 1

* **Funcion\_productorio ( , )**

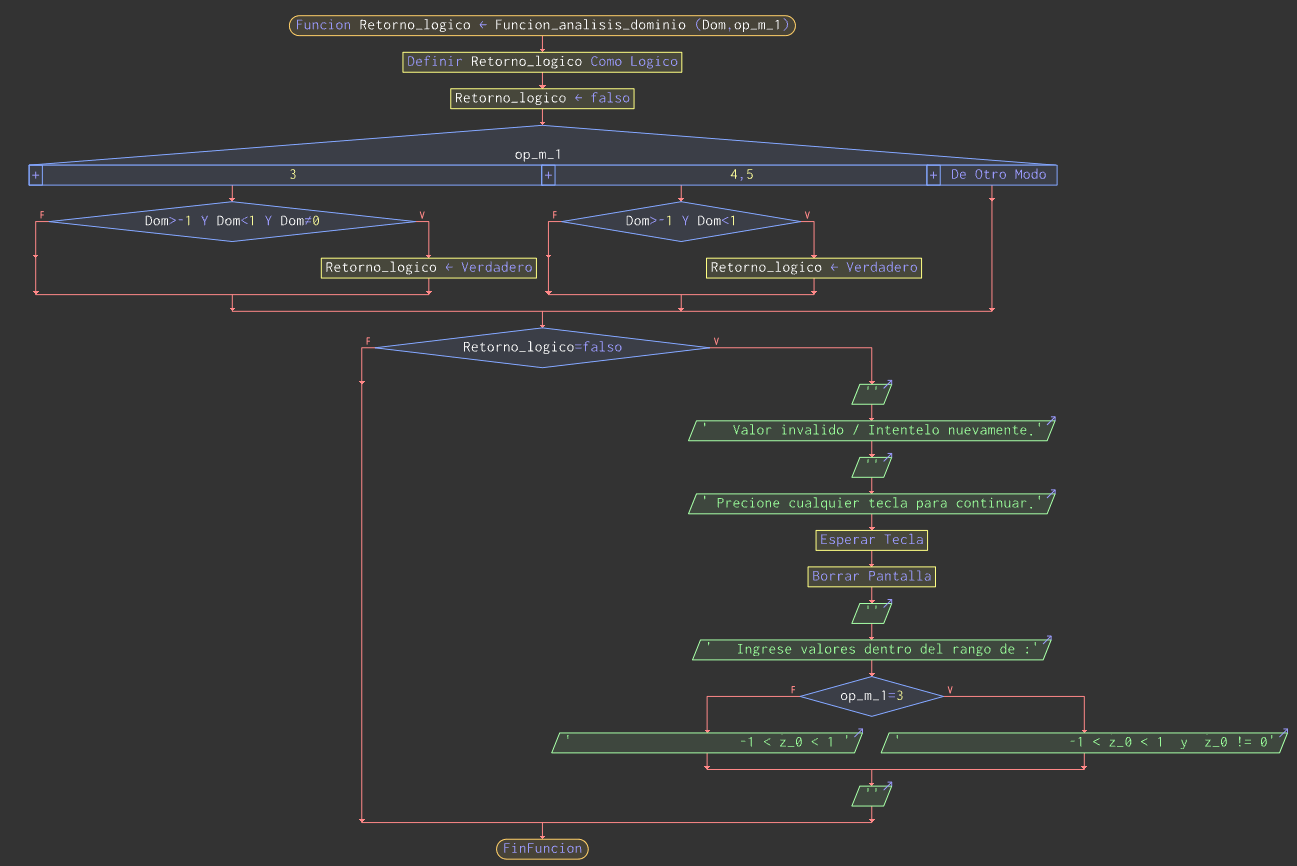
Esta función es usada para el momento en el que se desea obtener las operaciones de la serie binominal.

Cumple con la tarea matemática del productorio que la serie binominal necesita.

1. Funciones lógicas.

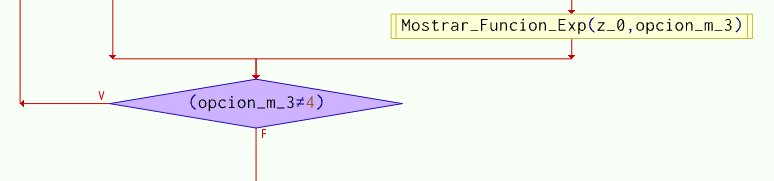
**Para que sirve.** –La única función lógica que se programó, fue la Funcion\_analisis\_dominio, esta función se activa al momento en el que el usuario necesite realizar las series de tipo logarítmica, serie geométrica y la serie binomial; Puesto que estas 3 tienen un radio en el disco de convergencia, el cual se vio en el marco teórico.

Básicamente esta función no dejara que el usuario ingrese un valor del punto de convergencia fuera de su respectivo radio de convergencia.

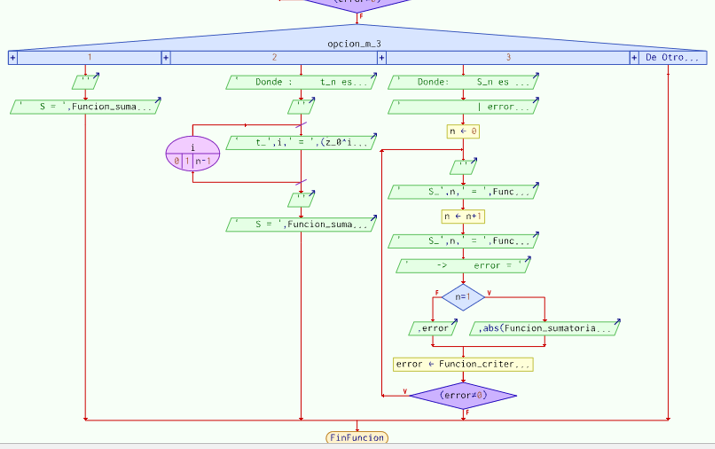
Dicha función depende de un valor de tipo flotante, el valor del punto ingresado, y de un valor entero el cual es el valor numérico del menu de opciones principal estos para este caso son los valores de { 3 (Serie logarítmica) , 4 (Serie geometrica) , 5 (Serie binomial) } es importante notar que la serie geométrica y la binomial comparten el mismo domino del punto de convergencia.

Para que el punto de convergencia ingresado por el usuario sea válido, este valor ingresa a una condicional switch() que depende de la opción seleccionada en el menú principal (Menú para elegir qué serie desea realizar) como esta función solo se activa para las series logarítmica, geométrica y binomial estas serán los posibles casos de la condicional switch; Sin embargo la serie geométrica y la serie binomial comparten el mismo radio de convergencia, por lo que para este caso se los agrupo en el mismo bloque de tareas, si el valor ingresado no cumple con la condicional if() encontrada dentro los posibles casos de switch() entonces la variable logia “Retorno\_logico” no cambia conserva el valor lógico de falso; Sin embargo si el valor ingresado cumple con dicha condicional, el retorno lógico cambiara a verdadero.

Para la segunda parte de esta función\_analisis\_dominio, se encuentra una condicional if() que evalúa si el “Retorno\_logico” cambio o no. Si fuera el caso de que “Retrono loico = True”, termina la función; Caso contrario se mostrara en la consola que dicho valor no se encuentra dentro del radio de convergencia y deberá ingresar nuevamente un valor, esta vez también le mostrara el radio de convergencia aceptado y luego de esto la función finaliza.

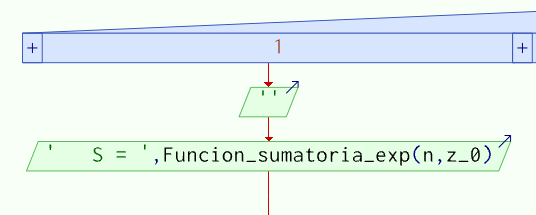
Y estos son las funciones que llegarían a activarse para las distintas series que tiene el algoritmo, y se procederá a explicar desde el punto en el cual el subprograma Mostar sumatoria … sale de un ciclo do{}while (El cual evaluaba el error )

Para ingresar en un switch() como se ve a continuación:



Ya generado la cantidad de términos que la serie contiene esto gracias al ciclo do{}while() ya cerrado finalmente el algoritmo ingresa a la condicional switch () la cual evalúa el valor de opción\_m\_3 la cual era el resultado del menú que pedía al usuario si quería ver el valor de la sumatoria final, los n-términos o el criterio que se utilizó para generar dicha sumatoria.

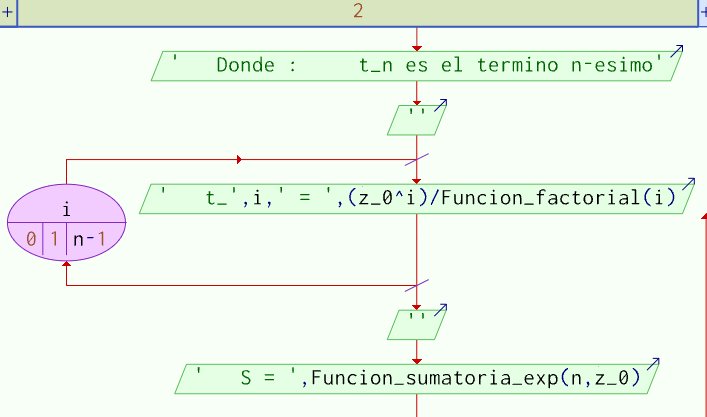
Para el 1er bloque, simplemente se llama a la Funcion\_sumatoria\_..., la cual para este punto tiene el valor final de dicha sumatoria



Para el 2do bloque, es utilizado un ciclo for(){} el cual inicia desde 0 hasta el valor que n tiene

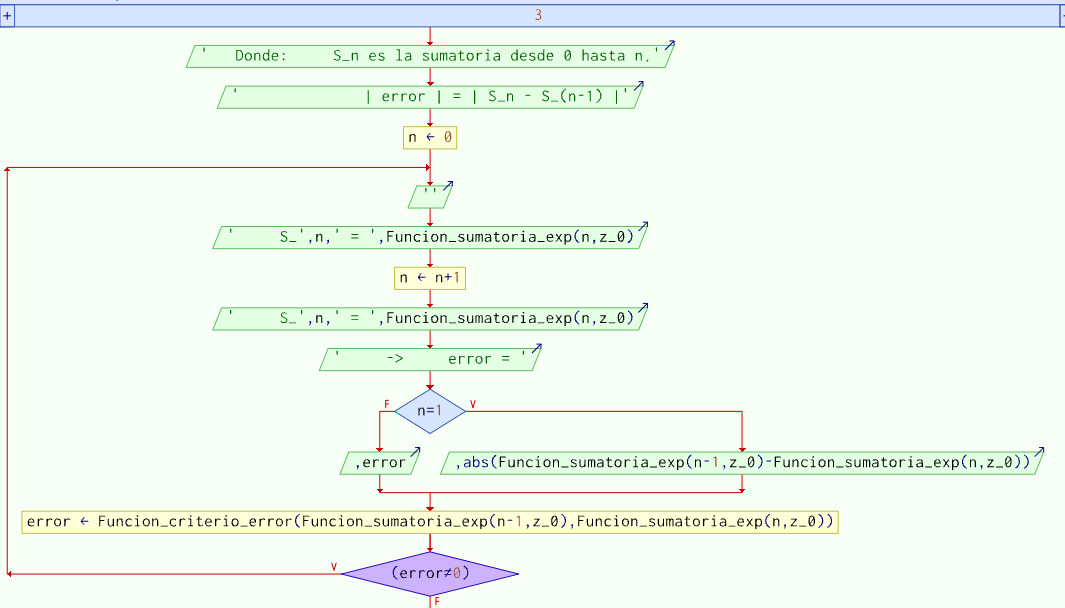
guardado, dentro del ciclo se realiza la tarea de generar y mostrar en la consola cada uno de los n termino esto con la fórmula de la serie y llamando a la Funcion\_factorial.

Saliendo del ciclo el bloque se imprime el valor de la sumatoria.



Para el 3er bloque, nuevamente se le da el valor de 0 a la variable “n”, esto con el fin de generar las n-sumatorias y poder aplicar el criterio mediante un ciclo do{}while() que finalizara si el criterio tiende a cero.

Además, se mostrará en la consola cada paso realizado.



Finalmente el subprograma Mostrar\_funcion\_...() termina pero este se repetirá hasta que el usuario digite la opción de salir y retornamos al menú principal.

Para terminar los procesos del programa finalmente el usuario deberá elegir la opción de salir del menú principal.

**Listado del código en el programa C++**

1. **Librerías**

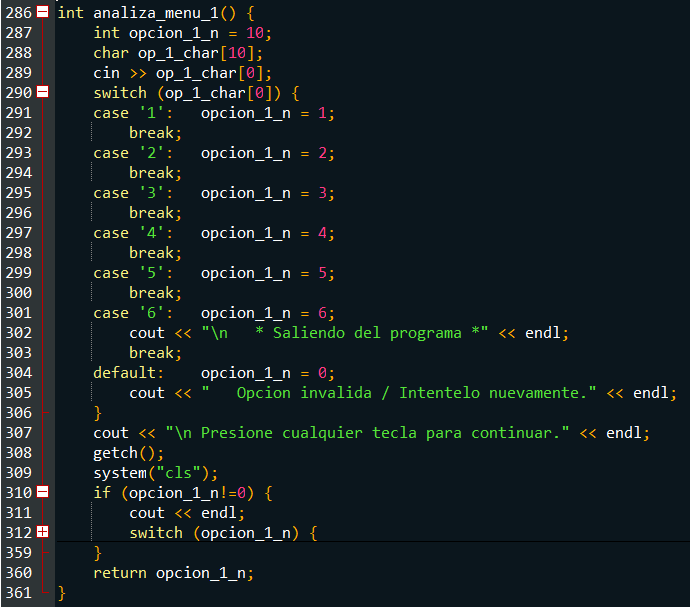
* **<iostream>**, para el ingreso y salida de datos.
* **<cmath>**, para realizar operaciones matemáticas de potencias o raíces, trigonométricas (seno, coseno, arco senos, etc.), etc.

* **<conio.h>**, para llamar a la función “getch(); “ que congela la consola hasta que el usuario digite una tecla.

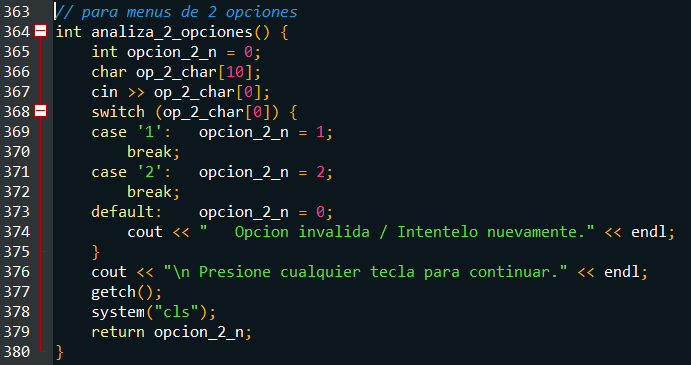
1. **Funciones**

**Nota. -** Para que las capturas de pantalla lograran entrar en las páginas, en algunos casos, la 2da condicional switch, la cual imprimia en pantalla la elección de usuario (teoría o la resolución de las series).

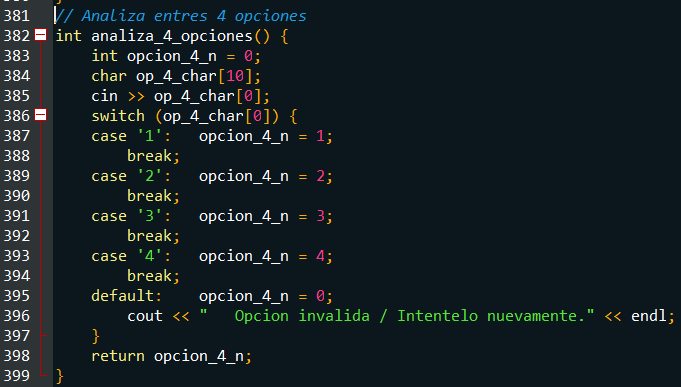
* **int analiza\_menu\_1()**

 Evita que el usuario ingrese valores erróneos para el momento en el que está escogiendo entre las opciones del menú principal.

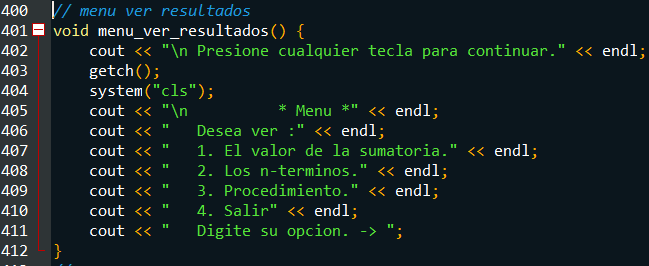
* **int analiza\_2\_opciones()**

Para los menús de 2 posibles opciones.

* **int analiza\_4\_opciones();**

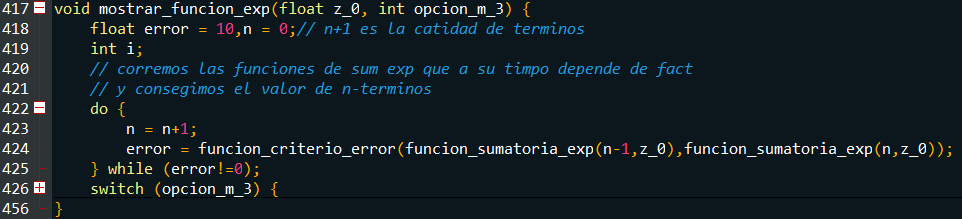
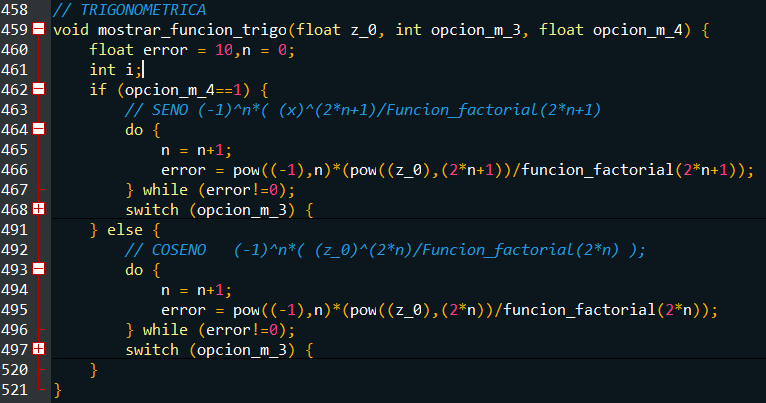
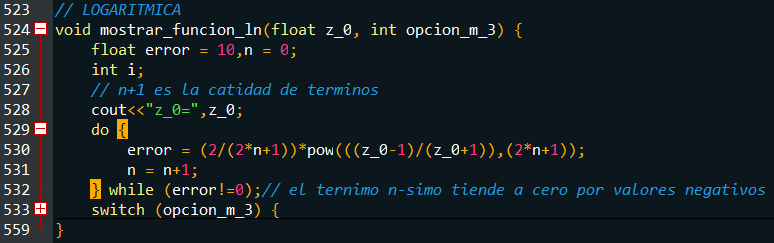
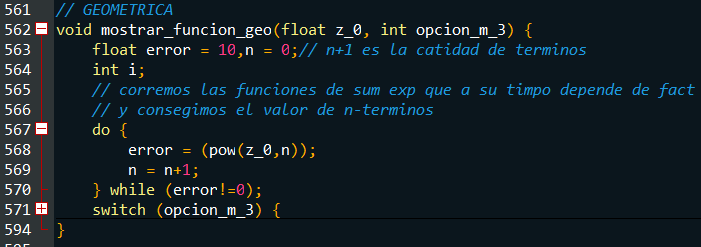
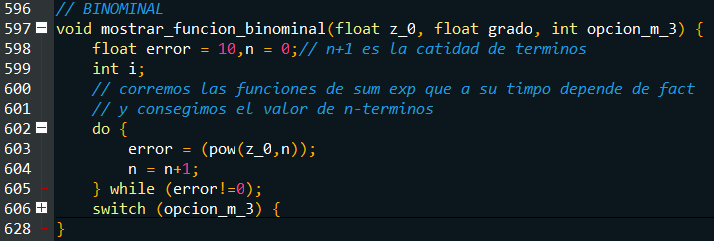
Para los menús de 4 posibles opciones.

* **void menu\_ver\_resultados();**

Mostrará en la pantalla el meno que se ve a continuación.

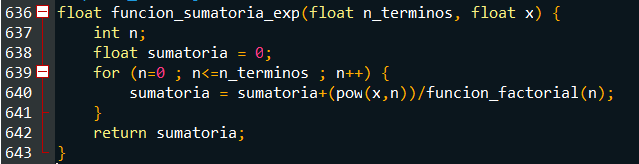
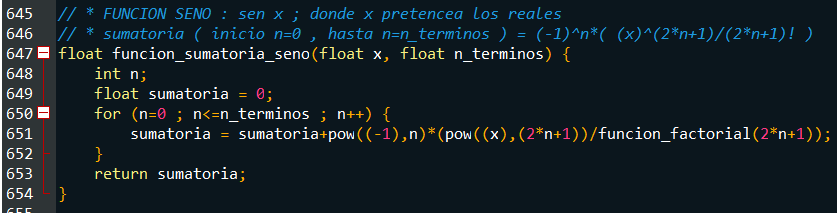
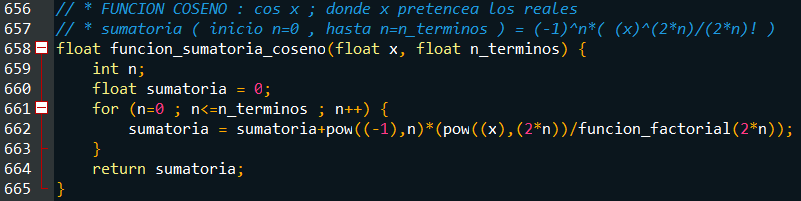
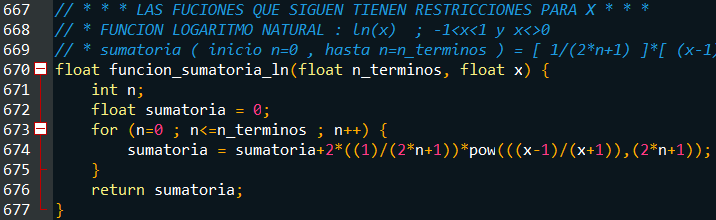
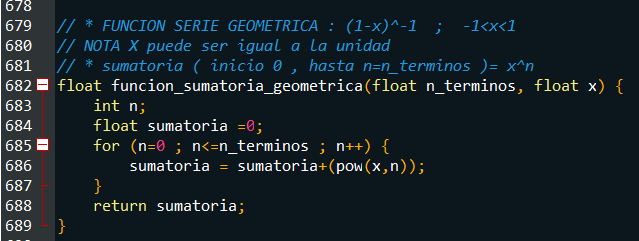
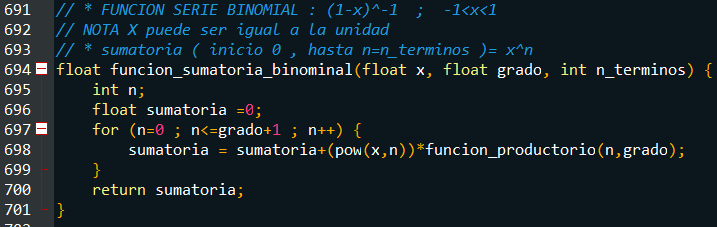
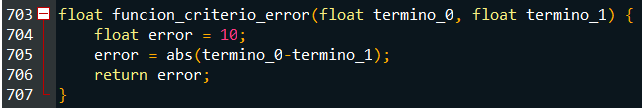
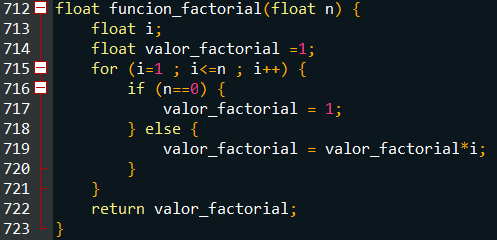
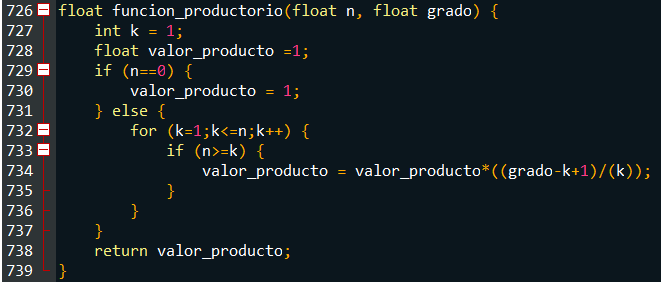
**Funciones tipo (mostrar\_funcion\_...). –**

En esta sección las funciones llamaran a los subprogramas que realizaran distinto tareas matemáticas que generaran el valor final de la sumatoria, el valor de los n- términos, y más; También mostraran esto en la consola dependiendo lo que el usuario desee ver.

* **void mostrar\_funcion\_exp(float z\_0, int opcion\_m\_3);**
* **void mostrar\_funcion\_trigo(float z\_0, int opcion\_m\_3, float opcion\_m\_4);**
* **void mostrar\_funcion\_ln(float z\_0, int opcion\_m\_3);**
* **void mostrar\_funcion\_geo(float z\_0, int opcion\_m\_3);**
* **void mostrar\_funcion\_binominal(float z\_0, float grado, int opcion\_m\_3);**

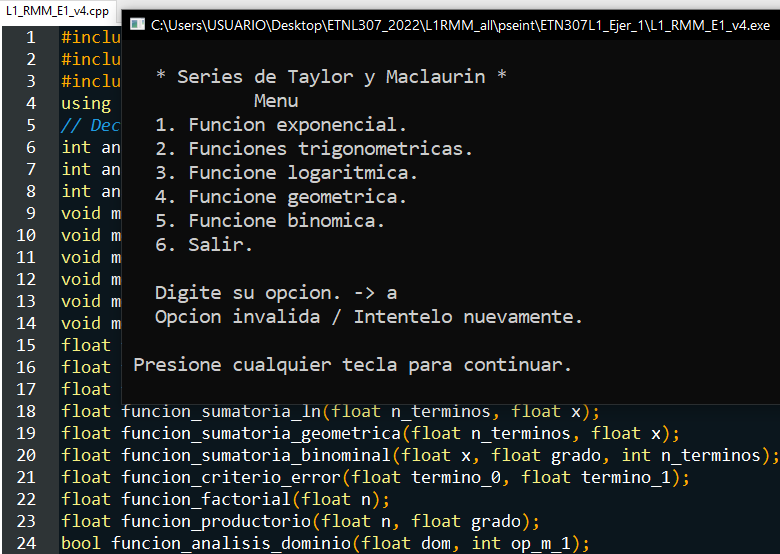
**Funciones tipo (función\_sumatoria\_...). –**

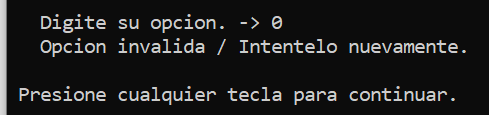
En esta sección las funciones llamaran a los subprogramas función\_factorial y función\_productorio, para generar el valor de la sumatoria.

* **float funcion\_sumatoria\_exp(float n\_terminos, float x);**
* **float funcion\_sumatoria\_seno(float x, float n\_terminos);**
* **float funcion\_sumatoria\_coseno(float x, float n\_terminos);**
* **float funcion\_sumatoria\_ln(float n\_terminos, float x);**
* **float funcion\_sumatoria\_geometrica(float n\_terminos, float x);**
* **float funcion\_sumatoria\_binominal(float x, float grado, int n\_terminos);**
* **float funcion\_criterio\_error(float termino\_0, float termino\_1);**
* **float funcion\_factorial(float n);**
* **float funcion\_productorio(float n, float grado);**
* **bool funcion\_analisis\_dominio(float dom, int op\_m\_1);**

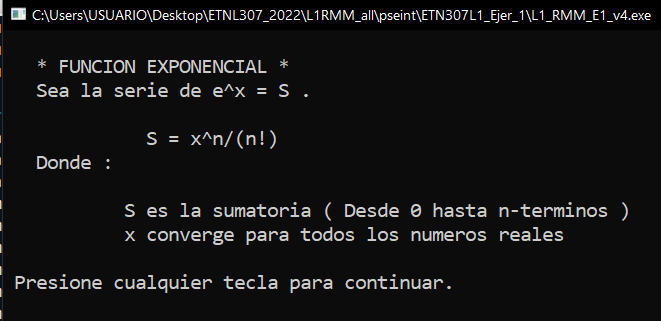
Capturas de pantalla de la ejecución del programa. –

* Probando el menú inicial:

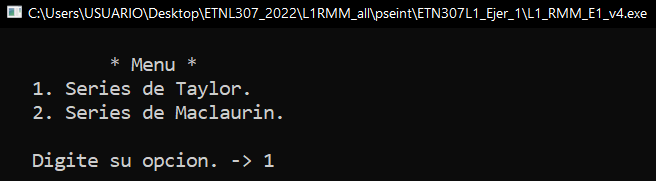
Ingresando valores erróneos al menú:

2da opción errónea.

Todo esto gracias a un ciclo do{}while.

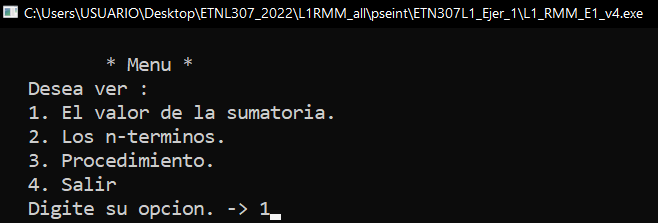
* Ingresando a la Serie exponencial.

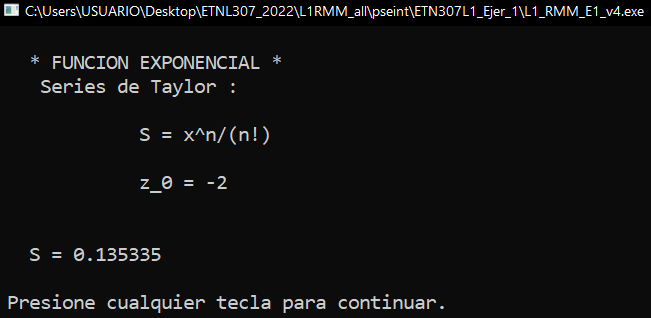
Presionando cualquier tecla:

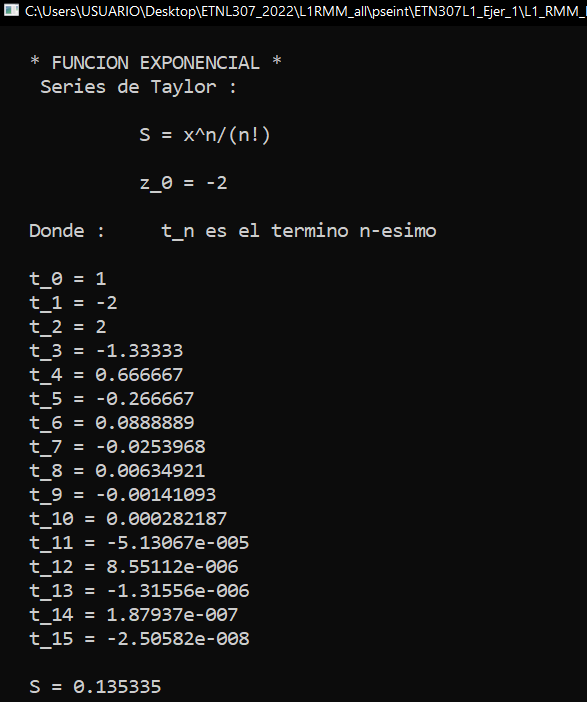
****

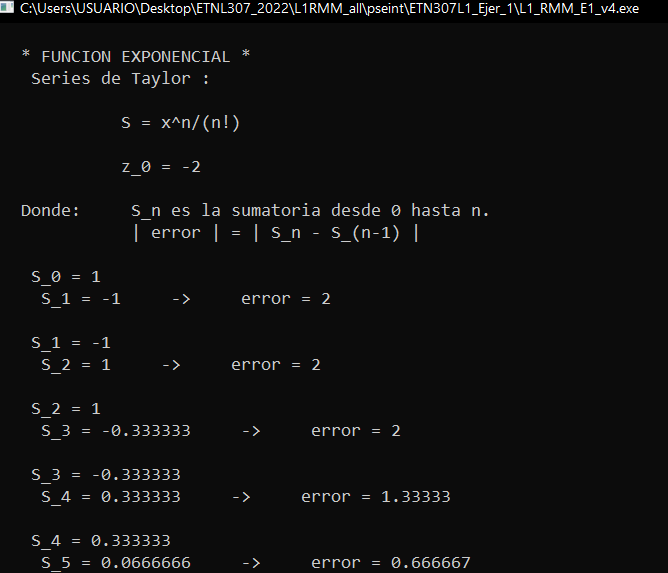
Luego de elegir la 1ra opción:

****

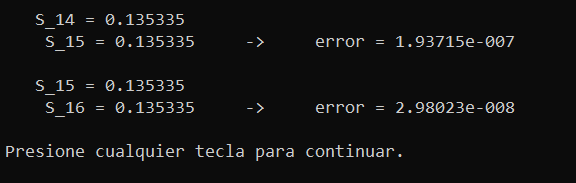
Ingresando el valor de convergencia = -2, y luego presionando enter.

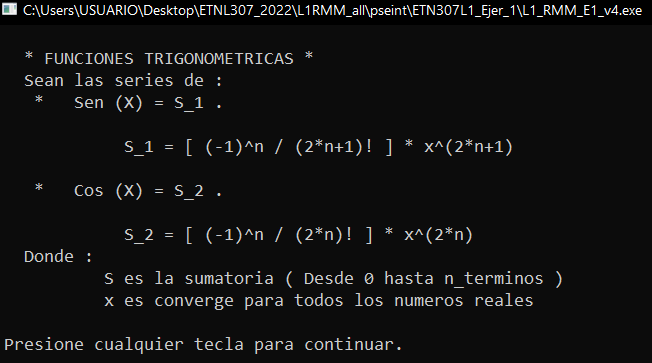
Para la 1ra opción.

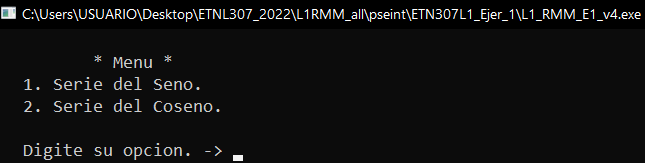
Para la 2da opción.

Para la 3ra opción.

Recortando la imagen.



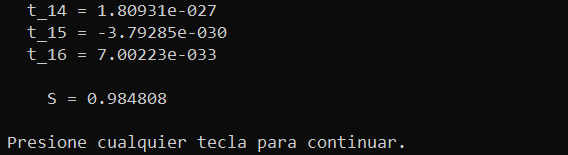
* Ingresando a la Serie trigonométrica.

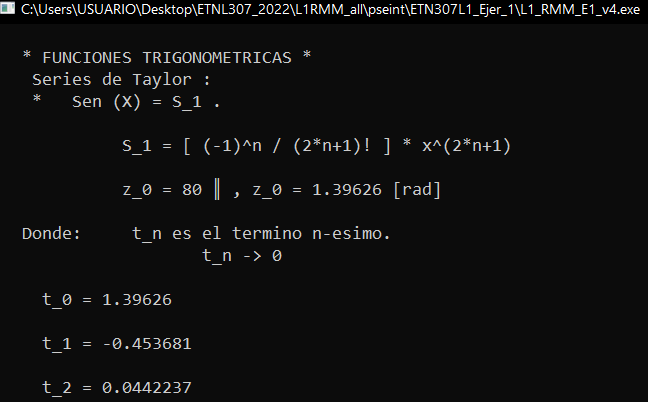
Luego de presionar cualquier tecla, ingresará al menú para seleccionar entre la serie de Taylor y Maclaurin para este caso se escogerá la serie de Taylor, luego aparecerá.

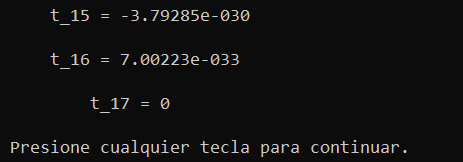
Escogiendo la serie del seno, luego preguntara que punto de convergencia desea y para este caso serán igual a 80 grado; luego ingresando en el menú de mostrar resultados.

Para la opción de ver el valor de la sumatoria.

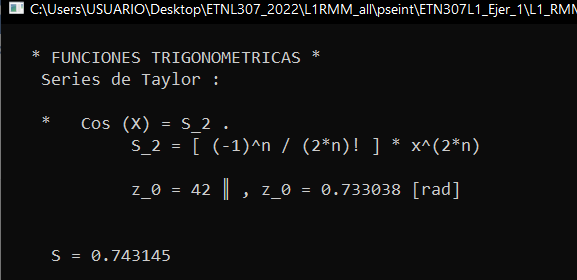
Para la opción de ver los n términos.

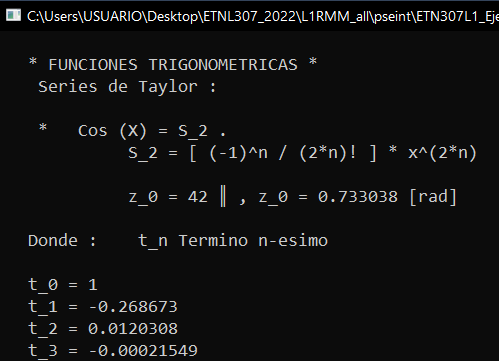
Cortando la imagen.

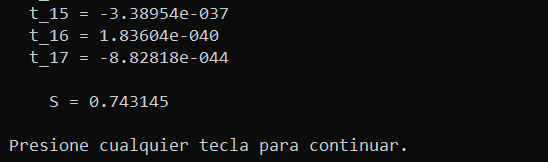
Para la opción de ver el procedimiento.

Cortando la imagen.

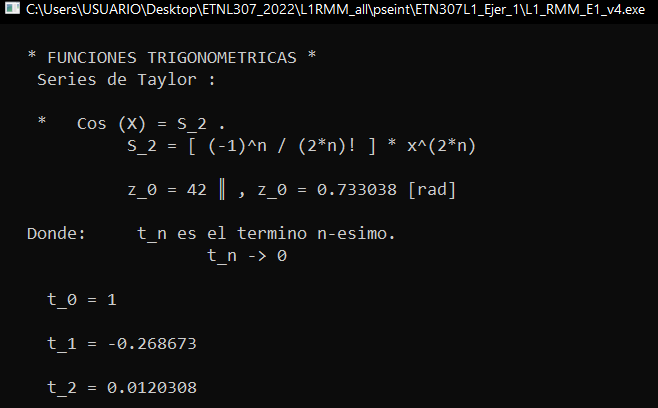
Escogiendo la serie del coseno, luego preguntara que punto de convergencia desea y para este caso serán igual a 42 grados; luego ingresando en el menú de mostrar resultados.

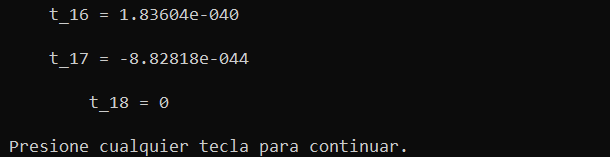
Para la opción de ver el valor de la sumatoria.

Para la opción de ver los n-términos.

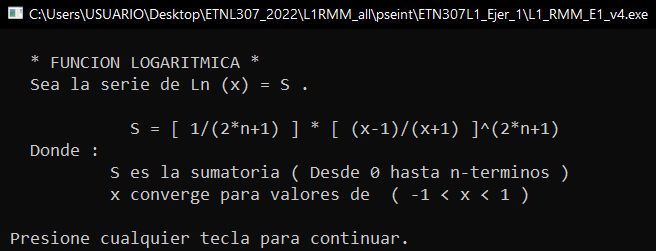
Recortando la imagen.

Para la opción de ver el procedimiento.

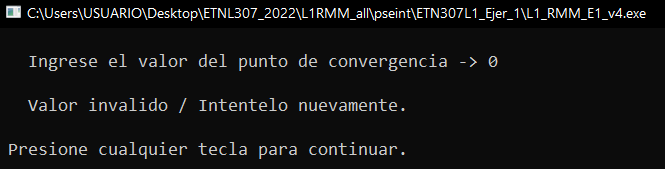


Recortando la imagen.

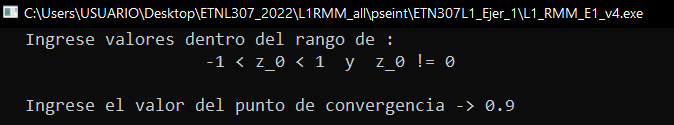
* Ingresando a la serie logarítmica.



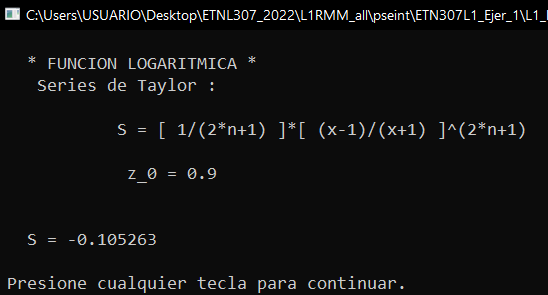
Ingresando valores erróneos para el punto de convergencia.



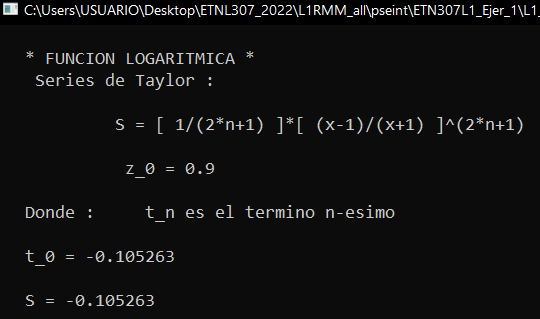
Presionando cualquier tecla, y luego ingresando un valor correcto.



Ingresando a la opción de ver el valor de la sumatoria.

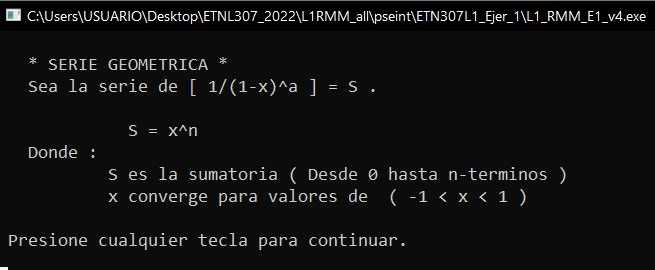


Ingresando a la opción de ver los n-términos.

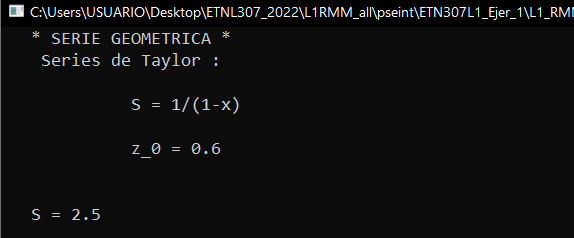


Ingresando a la opción de ver el procedimiento.

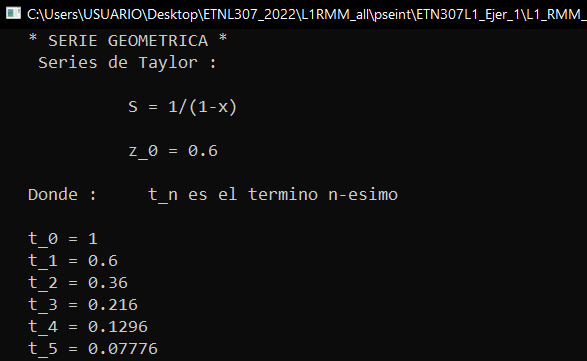
* Ingresando a la serie geométrica.

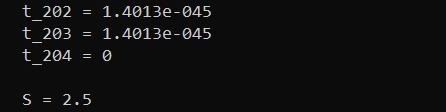


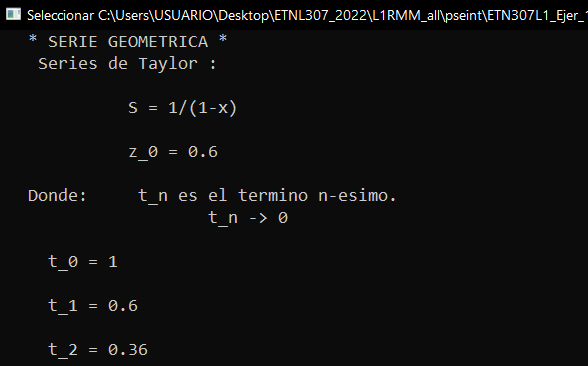
Escogiendo a la serie de Taylor para luego ingresar el valor del punto de convergencia igual a 0,6.

Ingresando a la opción de ver el valor de la sumatoria.

Ingresando a la opción de ver los n-términos.

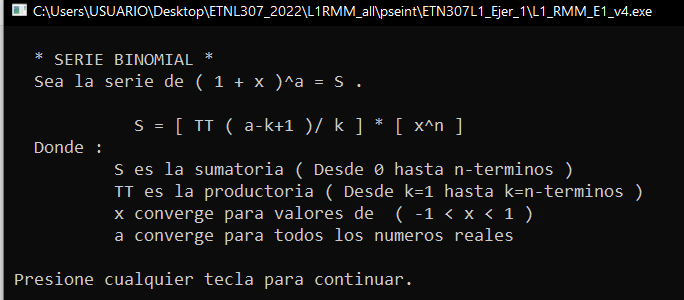


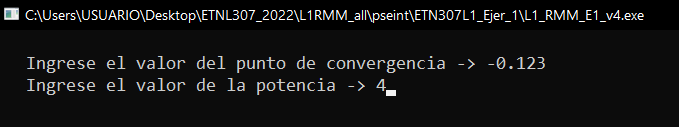
Recortando la imagen.

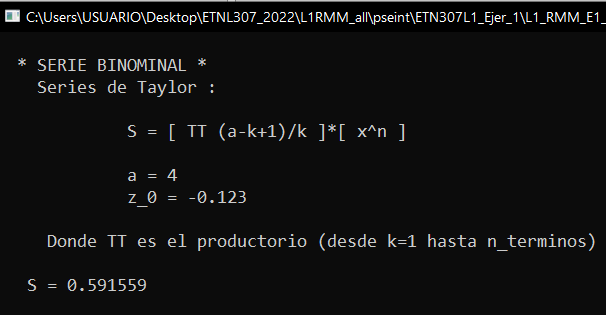
Ingresando a la opción de ver el procedimiento.

Recortando la imagen.

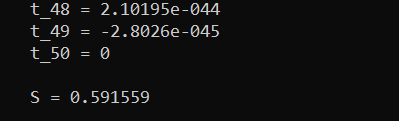
* Ingresando a la serie binomial



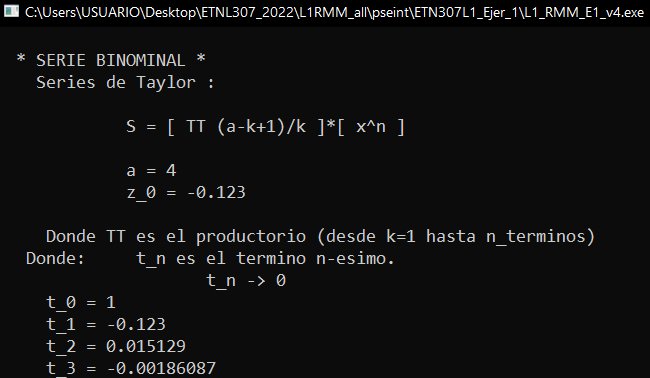
Escogiendo a la serie de Taylor para luego ingresar el valor del punto de convergencia igual a -0,123 además una potencia igual a 4.

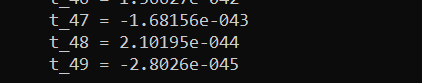
Ingresando a la opción de ver el valor de la sumatoria.

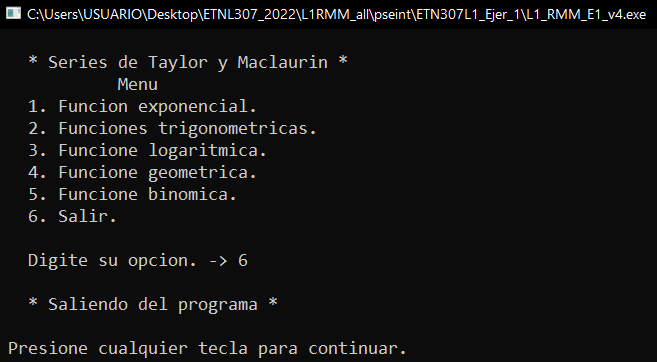
Ingresando a la opción de ver los n-términos.

Recortando la imagen.

Ingresando a la opción de ver el procedimiento.



Recortando la imagen.

* Finalmente escogiendo salir de menú principal.

Conclusiones. -

El algoritmo realizo bien la mayoría de las tareas, a excepción de la serie de la función logarítmica ya que en esta los valores que el programa entregaba fueron algo cercanos, pero en ningún punto de convergencia probado entrego un valor correcto.

Observaciones. -

Para el caso de la serie logarítmica talvez la razón por la que no funciona correctamente sea por las restricciones de su radio de convergencia ya que además de encontrarse entre -1<z\_0<1, también debía evitarse el punto con el valor igual a ‘0’.

Notando que el punto de convergencia era evaluado por la función\_análisis\_dominio, se realizaron distintos cambios para poder obtener el valor correcto de la sumatoria, sin embargo, ningún cambio fue exitoso.

Cabe aclarar que la función\_analisia\_dominio funciono con éxito para los subprogramas que realizaban las series geométrica y binomial (ambas con un radio del punto de convergencia).

* **ECUACION DE 2DO GRADO**

Marco teórico. –

Una ecuación de 2do grado también llamada ecuación cuadrática tiene la forma:



Dónde: x es la variable.

{\displaystyle a}a es el [coeficiente](https://es.wikipedia.org/wiki/Coeficiente_(matem%C3%A1ticas)) cuadrático (distinto de [cero](https://es.wikipedia.org/wiki/Cero)).

b{\displaystyle b}b el [coeficiente](https://es.wikipedia.org/wiki/Coeficiente_matematico) lineal.

c{\displaystyle c}cccccccccc es el término independiente.

a, b y c pertenecen a los números reales

* Demostración:

Lo primero es dividir la entre el coeficiente cuadrático.



Simplificando.



Despejamos el termino independiente



Sumamos en ambos miembros b^2/(4\*a^2)



Se puede observar un cuadrado perfecto la parte derecha de la ecuación



Despejando x se obtiene



Finalmente, la ecuación será:



* Cabe destacar el análisis de la discriminante la cual es:



1. Discriminante positiva resultara en dos soluciones reales distintas.
2. Discriminante igual a cero resultara en dos soluciones reales pero iguales.
3. Discriminante negativa resultara en dos soluciones imaginarias conjugadas.

* Ejemplos prácticos. –



1. Si

Sea la ecuación:

|  |
| --- |
| Resuelto por factorización tiene como raíces: |

Otro ejemplo:

|  |
| --- |
| Resuelto por la formula general cuadrática. |

1. Si

Sea la ecuación:

|  |
| --- |
| Resuelto por la formula general cuadrática. |

Otro ejemplo:

|  |
| --- |
| Resuelto por la formula general cuadrática. |



1. Si

Sea la ecuación:

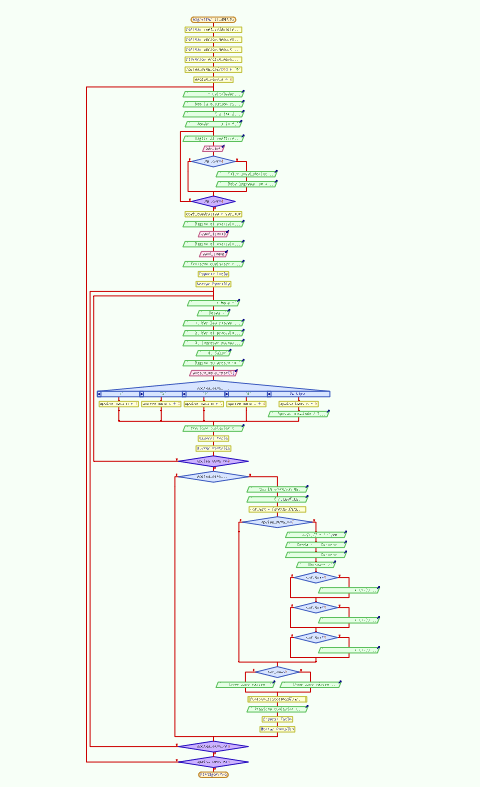
|  |
| --- |
| Resuelto por la formula general tiene como raíces imaginarias conjugadas: |

Otro ejemplo:

|  |
| --- |
| Resuelto por la formula general tiene como raíces imaginarias conjugadas: |

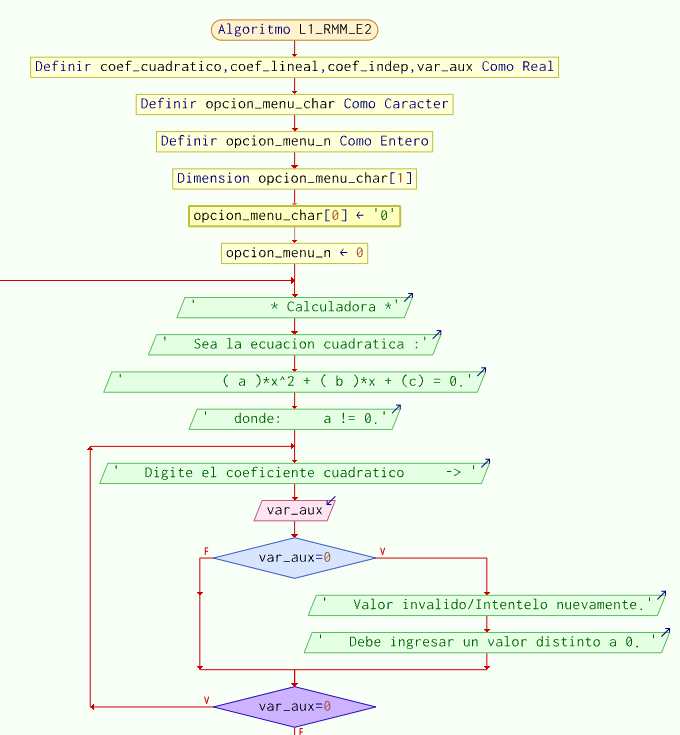
Diagramas de flujo. –

Para dibujar los diagramas de flujo de los algoritmos se empleó el programa “PSeInt”, y como estos están conformados por funciones se muestran de la siguiente manera; Además para poder visualizar de manera detallada se realizarán distintos acercamientos a cada diagrama de flujo.

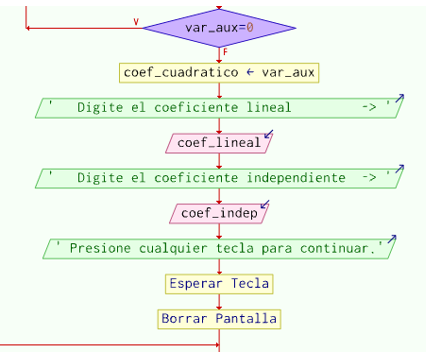
* Series de Taylor y Maclaurin (L1\_RMM\_EJ\_2 es el nombre del archivo)
* Diagrama principal

Zoom (superior del diagrama de flujo principal).

* Inicio del algoritmo: el algoritmo se llama L1\_RMM\_EJ\_2

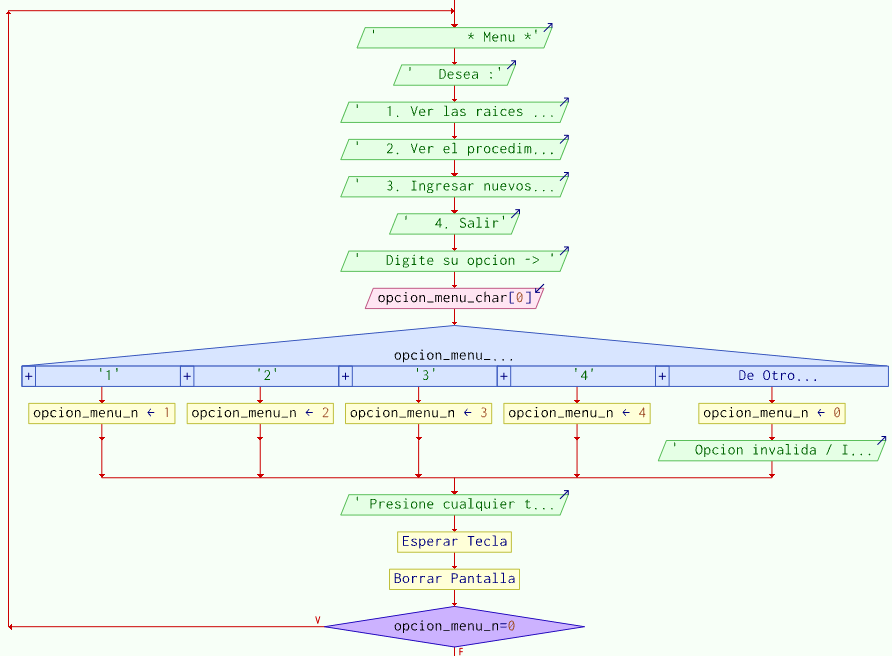
* Definir variables: Opcion\_menu\_char para que el usuario ingrese cualquier valor y el programa no tropiece con opciones letras u opciones que no se encuentran en le menú, opicon\_menu\_n para cuando si existe la opción ingresada tome un valor numérico entero.
* Se inicia un ciclo do{}while() que termina si y solo si el usuario digita la opción de salir.
* Impresión en la pantalla: Muestra el menú de inicio y pide ingresar su opción.
* Se inicia otro ciclo do{}while() para evaluar que el coeficiente cuadrático no tenga el valor de cero.
* Se ingrese el valor de una variable auxiliar, para la tarea ya mencionada.
* Condicional if(){}else{} , para mostrar en la consola un mensaje( si el valor del coef. Cuadrático fuera cero)
* Condicional While(), para terminar el proceso repetitivo si el valor del coef. Cuadrático fuera de 0

Zoom (fuera de la condicional while).

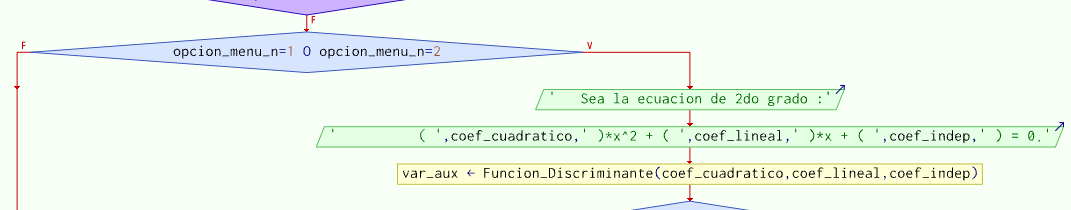
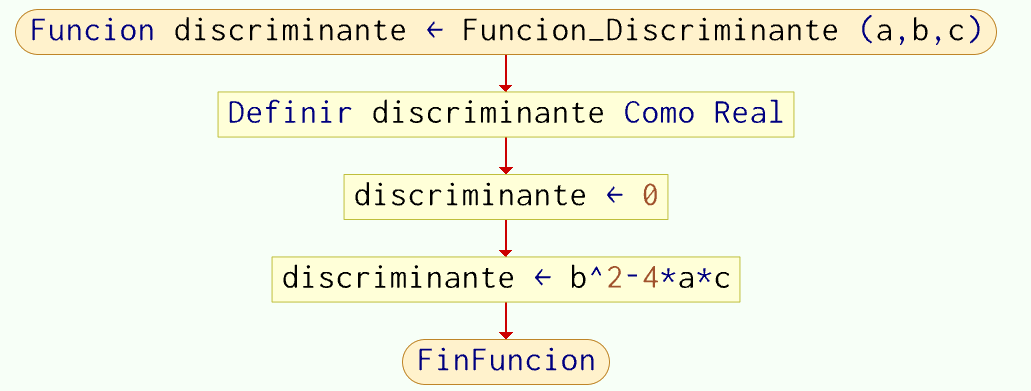


* Si el valor ingresado anteriormente fuera distinto de cero, se guardara en la variable coef\_cuadratico.
* Se piden los valores de los demás coeficientes, como también en cada variable se le guarda el valor que le corresponde
* Se mostrará un mensaje, luego se limpia la consola.
* Se inicia dos ciclos do{}while(), para que se retorne continuamente al menú.

Zoom (menú).

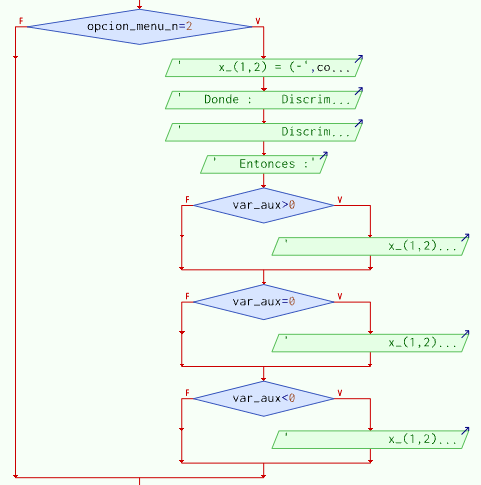
* Se mostrará el menú, para luego pedir la opción que se desea ejecutar(guardando la opción en una variable tipo char).
* Condicional switch, como se ve en el diagrama, si es ingresado una opción invalida (letras, etc.) la variable opción\_menu\_n tomara el valor de 0; Sin embargo, si el valor ingreso si está dentro de las opciones, la variable numérica tomara el valor de la opción requerida.
* Si fuera el caso de ‘opción\_menu\_n = 0’ la condicional while evitaría que el usuario pase del menu.

Zoom (condicional if).

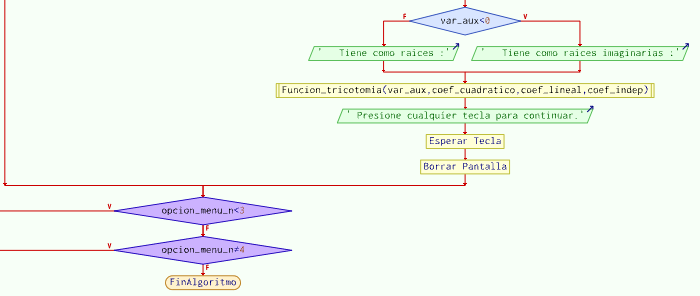
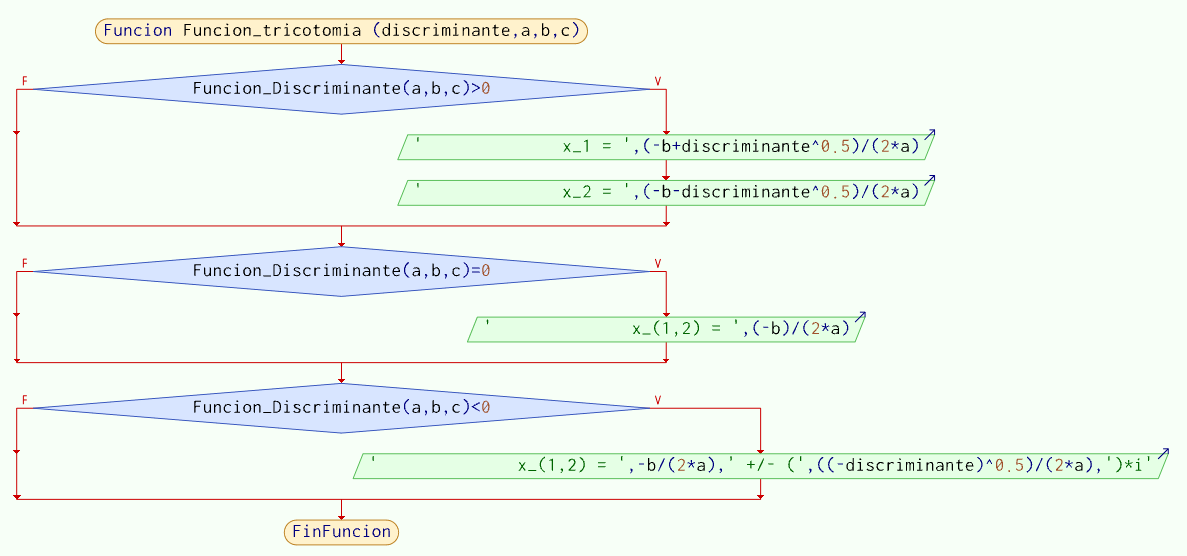
* La condicional if(){}else{}, que se ve en esta parte del diagrama mostrara o no el procedimiento.
* Cabe aclara que si fuera el caso (opción\_menu\_n = 2), la variable numérica var\_aux tomaría el valor retornado por la Función \_Discriminante.
* Antes de continuar será necesario notar que la función\_Discriminante depende de los 3 valores, ingresados ya que esta tendrá por formula:
* Funcion\_Discriminante, es llamada y por ello mostraremos si propio diagrama de flujo.
* Realiza la tarea matemática de la formula ya mencionada, para luego devolver el valor calculado.

Saliendo de la Función y retornando al diagrama principal en la parte que continua.

Zoom (condicional if).

* La condicional if(){}else{}, para mostrar o no el procedimiento para luego considerar si se trata de una discriminante positiva, negativa o nula(esto con sus respectivos condicionales if{}).
* Básicamente mostrar como de efectúan las operaciones de la formula cuadrática con los valores ingresados.

Zoom (condicional if).

* Se ingresa a otra condicional if{} para imprimir en la consola, si las raíces son reales o son imaginarias conjugadas; para luego llamar a la Funcion\_tricotomia.
* Antes de continuar será necesario notar que la función tricotomía depende de los 3 valores ingresados y del valor de la discriminante ya que para realizar las operaciones finales estos serán utilizados para los distintos casos de la tricotomía de la discriminante.
* Funcion\_Discriminante, es llamada y por ello mostraremos si propio diagrama de flujo.

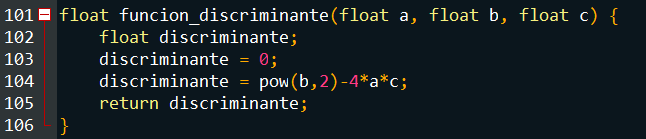
Para terminar los procesos del programa finalmente el usuario deberá elegir la opción de salir del menú.

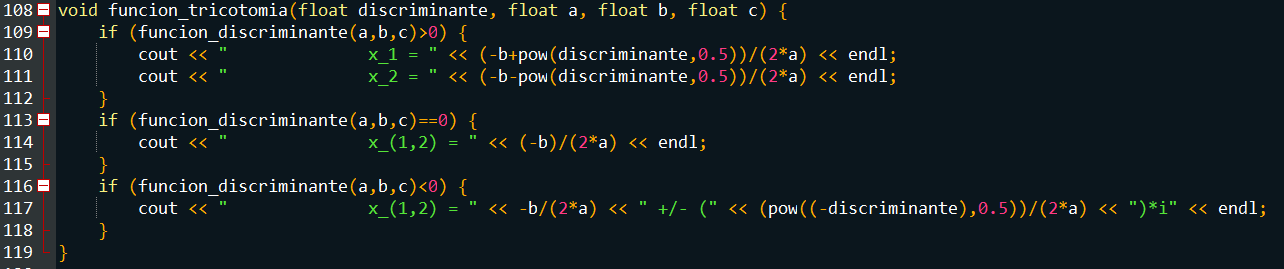
**Listado del código en el programa C++**

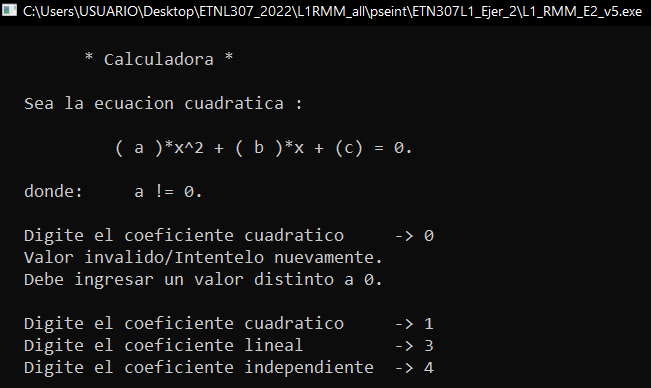
1. **Librerías**

* **<iostream>**, para el ingreso y salida de datos.
* **<cmath>**, para realizar operaciones matemáticas de potencias o raíces, trigonométricas (seno, coseno, arco senos, etc.), etc.
* **<conio.h>**, para llamar a la función “getch(); “ que congela la consola hasta que el usuario digite una tecla.

1. **Funciones**

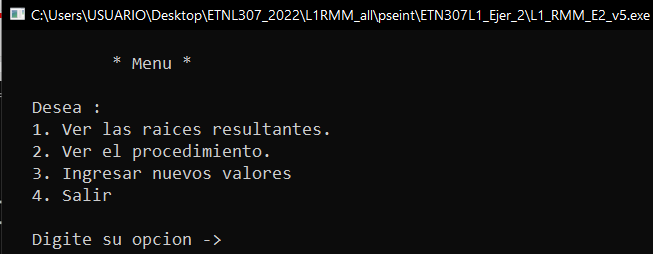
* **int funcion\_discriminante(),** depende de las 3 variables ingresadas y calcula el valor de la determinante.
* **Void funcion\_tricotomia(),** depende de las 3variables originales y también del valor de la determinante con el se aplican distintos condicionales para efectuar la terea matemática requerida.



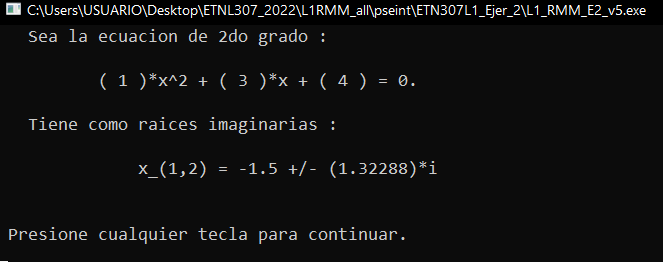
* Ingresando al programa.

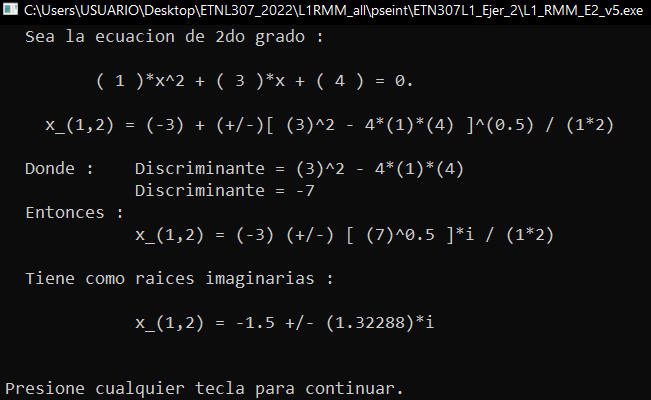
El programa inicia mostrando la forma de una ecuación de 2do grado, para luego aclarar que el valor de ‘a’ debe ser distinto de 0 y finalmente pide al usuario ingresar un valor para a; Si el usuario ingresara el valor de ‘0’ para el coef. Cuadrático, se mostrará el mensaje “valor invalido/Inténtelo nuevamente.” Y aclarara que el valor ingresado para el coeficiente ‘a’ debe ser dittinto de ‘0’

Luego de que se ingrese algún valor valido para ‘a’, se precede a pedir los coeficientes restantes (b y c).

Presionando cualquier tecla para continuar:

Se muestra un menú, es cual será recurrente hasta que el usuario elija la opción 3 o la 4.

Luego de elegir la 1ra opción:

Para la 2da opción:

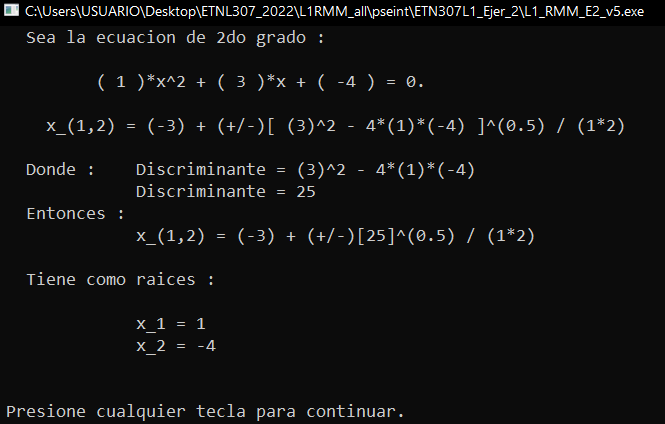
Ahora se probará al programa para que resuelva los ejemplos vistos anteriormente, con las distintas posibles soluciones y en la opción de ver desarollo.



1. Si

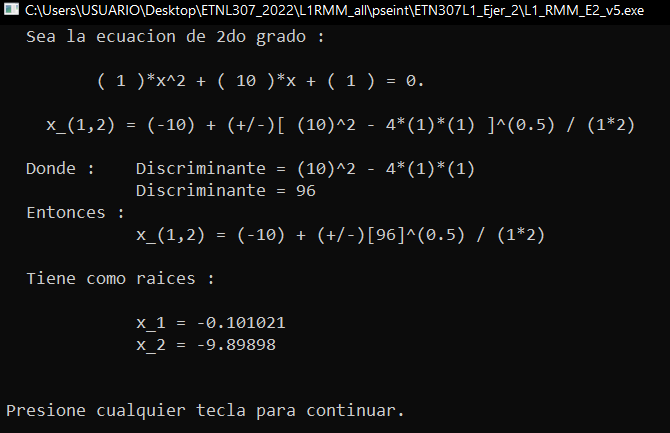
Sea la ecuación:

|  |
| --- |
| Resuelto por factorización tiene como raíces: |

En el programa.

Otro ejemplo:

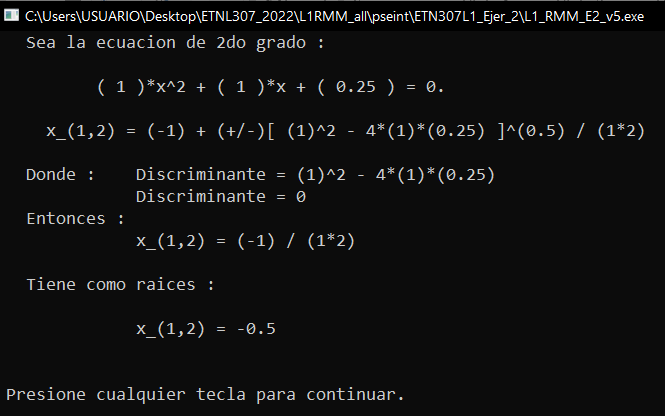
|  |
| --- |
| Resuelto por la formula general cuadrática. |

En el programa.

1. Si

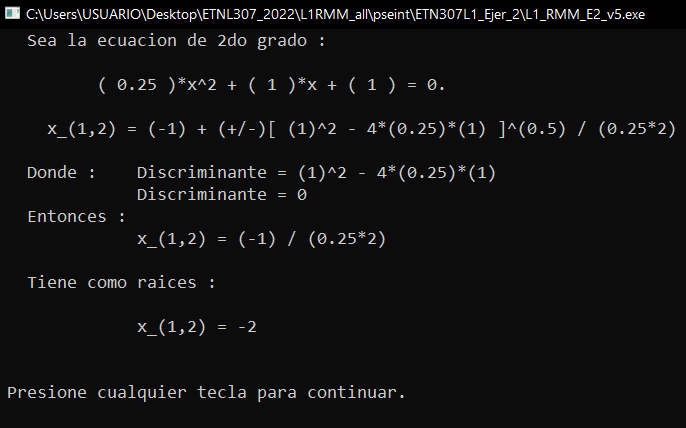
Sea la ecuación:

|  |
| --- |
| Resuelto por la formula general cuadrática. |

En el programa.

Otro ejemplo:

|  |
| --- |
| Resuelto por la formula general cuadrática. |

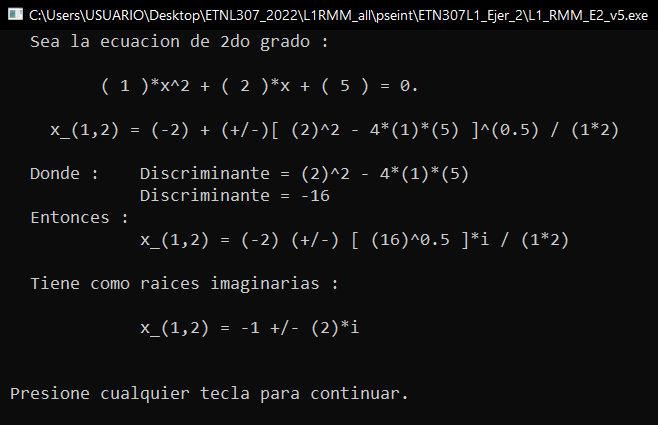
En el programa.

1. Si

Sea la ecuación:

|  |
| --- |
| Resuelto por la formula general tiene como raíces imaginarias conjugadas: |

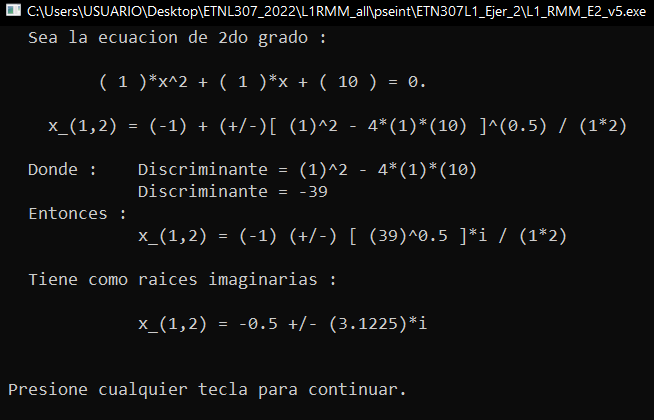
En el programa.



Otro ejemplo:

|  |
| --- |
| Resuelto por la formula general tiene como raíces imaginarias conjugadas: |

En el programa.



Conclusiones. -

El algoritmo realizo bien todos los casos que se le presento.

Observaciones. -

Los decimales que el programa maneja varía entre 5 a 6 decimales, se deberá tomar esto en cuenta para futuros proyectos si fueran necesario.

* **ECUACION DE 3ER GRADO**

Marco teórico. –

La ecuación general de tercer grado es:



Dónde: x es la variable.

a es el coeficiente cubico (distinto de cero)

{\displaystyle a}b es el [coeficiente](https://es.wikipedia.org/wiki/Coeficiente_(matem%C3%A1ticas)) cuadrático (distinto de [cero](https://es.wikipedia.org/wiki/Cero)).

c{\displaystyle b}b el [coeficiente](https://es.wikipedia.org/wiki/Coeficiente_matematico) lineal.

d{\displaystyle c}cccccccccc es el término independiente.

a, b, c y d pertenecen a los números reales.

Para la resolución de ecuaciones cubicas se empleará el método de Cardano-Tartaglia.

* Demostración:

Dividiendo a de la ecuación general



Para mayor facilidad de operaciones se partirá de la ecuación

 . . . 

Realizando el cambio de variable

 . . . 

Se elimina el termino cuadrático y se obtiene la forma reducida

 . . . 

Donde:



Realizando otro cambio de variable

 . . . 

Elevando este último al cubo se obtiene

. . . 

Que nos da la identidad

 . . . 

Por lo tanto, si encontramos los calores de u, v tales que

 . . . 

Despejando v de la primera ecuación



Reemplazando está en la ec. (4) obtenemos

 . . . 

También podemos reemplazar v en la 2da ecuación y obtenemos



Multiplicando por u^3

 . . . 

* Nótese que esta ecuación tiene una forma similar a una ecuación cuadrática, entonces podemos verla de la siguiente manera



(esto solo para tomar prestado el análisis de la discriminante)

Reemplazando la ec. (6) en la ec. (5) obtenemos:

 . . . 

Así pues, concluimos que resolver la ecuación (3) es equivalente a resolver la ecuación (8), en el sentido de que las soluciones de la ecuación (3) son las que se obtienen a partir de las soluciones de la ecuación (8) a través de la ecuación (7). Ahora bien, en la ecuación (8) u es cuadrática con en u^3, luego sus soluciones son:



O visto de una manera más sencilla:

 . . . 

Donde la discriminante la podemos escribir de la siguiente manera



Combinando las 2 raíces cuadradas de la discriminante con las 3 raíces cubicas, nos salen las 6 raíces de la ecuación (8), pero la ecuación (3) solo puede tener 3 raíces. Ello se debe a que, cada valor de t, hay 2 valores de u que cumplen con la ecuación (7). Dado uno de ellos, el otro está determinado por la relación

 . . . 

Que se deduce de que u u’ son raíces de la ecuación



Fijando un u positivo que cumpla con la ecuación (10), con signo positivo y una raíz cuadrada prefijada, y sea u negativo una raíz de la ecuación (10) con signo negativo (y de la misma raíz cuadrada). Entonces



Luego



Donde omega es la solución de la raíz cubica de la unidad. Por consiguiente



Cambiando u negativo por (omega)^2\*u\_, concluimos que, para cada u positivo elegido arbitrariamente, existe un u negativo elegido adecuadamente tal que u positivo y u negativo determinan la misma raíz t de la ecuación (3). En definitiva, vemos que las soluciones de (3) son de la forma de la ecuación (7), con



donde en esta expresión hay que entender que la raíz cuadrada del determinante es fija (elegida arbitrariamente de entre las dos posibilidades) y que al variar la elección de la raíz cubica recorremos las distintas raíces de la ecuación (3). Finalmente observamos que, de la segunda ecuación de (6), se sigue que



Despejando el cubo obtenemos



* Análisis de los casos posibles

1. **Caso de una raíz triple**

En este caso en particular la forma de la ecuación cubica será de la forma:



Notando resulta una raíz triple de alpha.

Donde en la fórmula de Cardano se reduce a



* Cubicas en el caso clásico

Esta depende de la discriminante y sus casos son:

1. **Si**

Todas sus raíces son reales, y almenos dos de ellas son iguales.

P y q son distintos de cero, entonces la ecuación tiene como raíces





1. **Si**

Se tiene una raíz real y dos imaginarias.

Una raíz real viene dada por



donde las raíces cubicas u y v son reales. Las otras dos raíces son imaginarias, y vienen dadas por





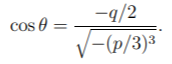
1. **Si**

La ecuación tiene tres raíces reales y simples.

Estas raíces vienen dadas por



Donde k=0,1,2 y el ángulo 0 < theta < pi está determinado por



* Ejemplos prácticos. –



1. Caso de una raíz triple

Sea la ecuación:

|  |
| --- |
| Resuelto por factorización tiene como raíces |

1. Si

Sea la ecuación:

|  |
| --- |
| * Valores calculados:   Resuelto por factorización tiene como raíces: |

Otro ejemplo ecuación:

|  |
| --- |
| Valores calculados:  Resuelto por factorización tiene como raíces: |

1. Si

Sea la ecuación:

|  |
| --- |
| * Valores calculados:   Resuelto por factorización tiene como raíces: |



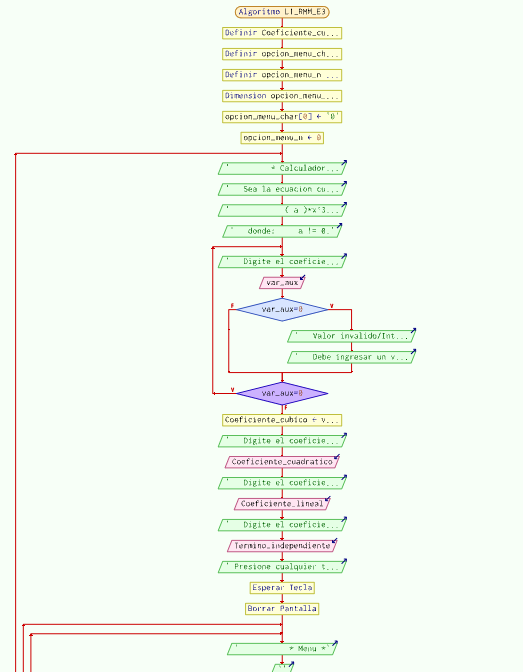
1. Si

Sea la ecuación:

|  |
| --- |
| * Valores calculados:   Resuelto por factorización tiene como raíces: |

Diagrama de flujo. –

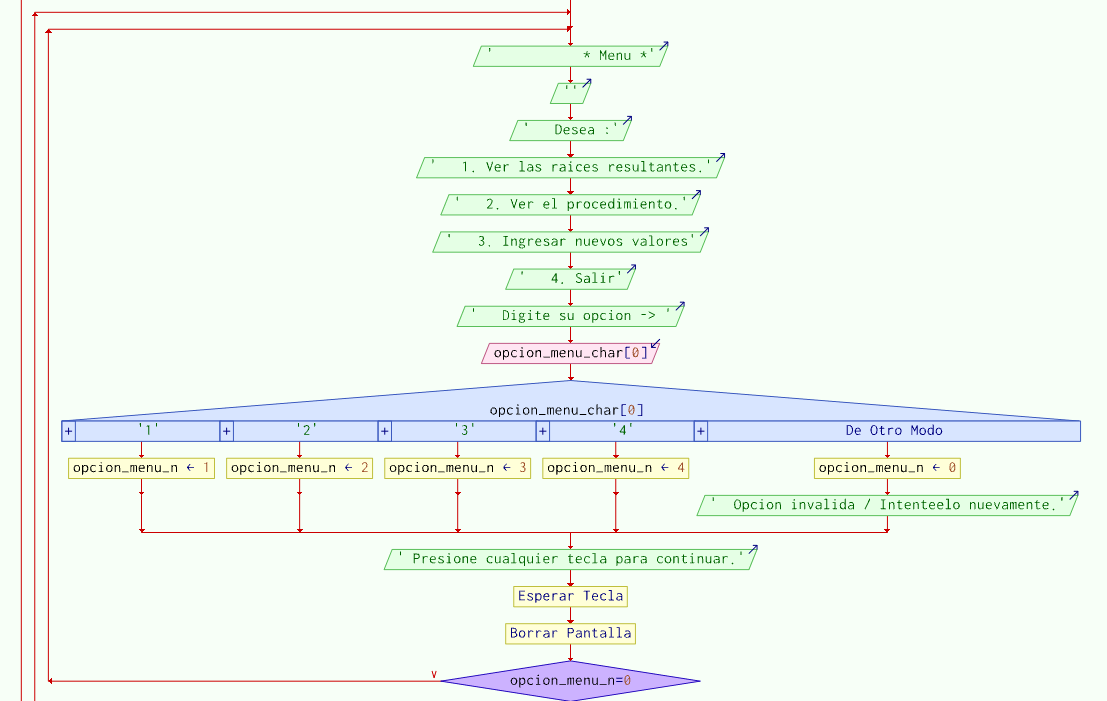
Para dibujar los diagramas de flujo de los algoritmos se empleó el programa “PSeInt”, y como estos están conformados por funciones se muestran de la siguiente manera; Además para poder visualizar de manera detallada se realizarán distintos acercamientos a cada diagrama de flujo.

* Series de Taylor y Maclaurin (L1\_RMM\_EJ\_3 es el nombre del archivo)
* Diagrama principal (parte superior)
* ****Zoom a la parte superior del diagrama de flujo

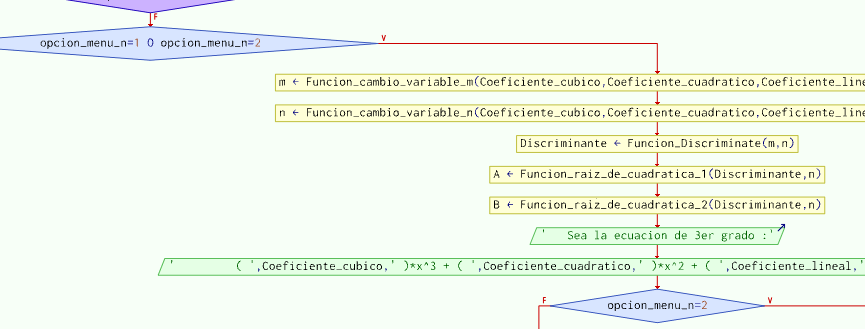
Similar al algoritmo anterior, se declaran variables que se usuran en determinados puntos del programa.

También consta de un ciclo do{}while() que impedirá que el programa llegara a cerrarse, sí que el usuario no lo deseara; Otro ciclo do{}while() que impide que el valor para el coeficiente cubico sea cero.

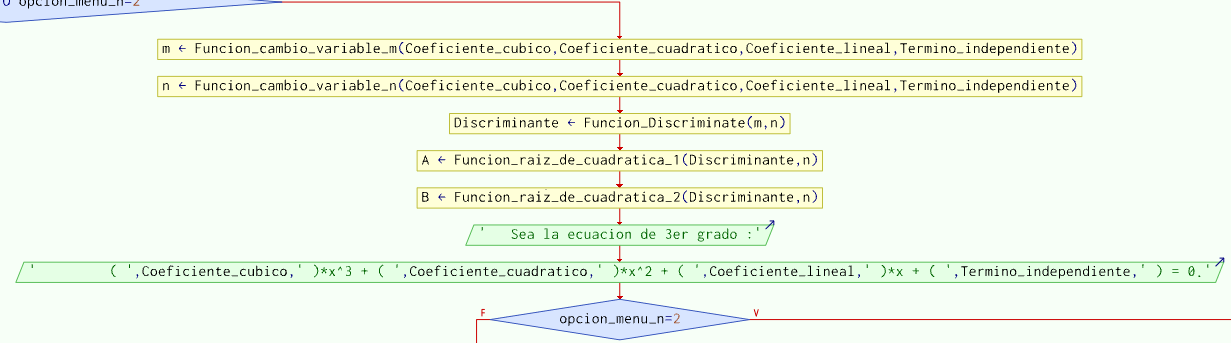
Se ingresan los valores de los demás coeficientes, se borra la pantalla y finalmente ingresamos al menú.

****Zoom a la parte del menú

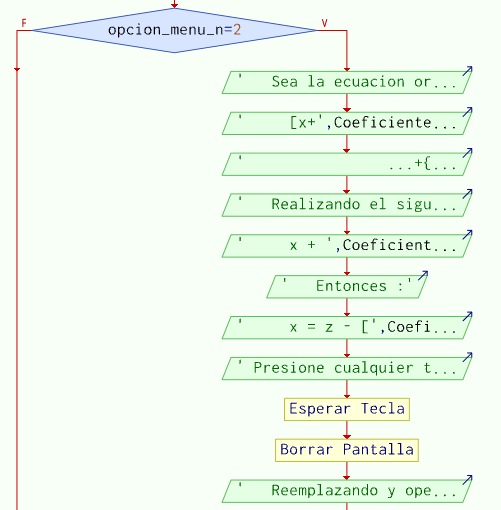
En esta parte del se elija que el usuario ingrese un valor erróneo para la selección de opciones, guardando dicho valor para luego pasar por un chico do{}while() para continuar o repetir hasta que el valor ingresado sea correcto.

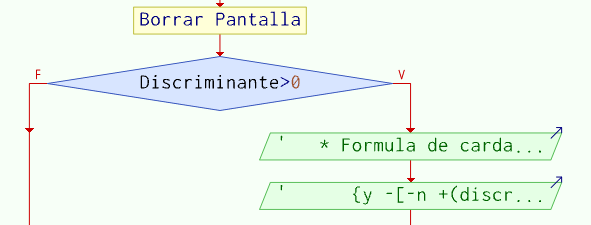
Zoom a la parte del menú

El valor ingresado finalmente sele del ciclo, para ser evaluado en una condiciones if(){}else{} para iniciar con los cálculos para resolver la ec. De 3er grado, o saltarlos y reiniciar el programa o simplemente cerrar dicho programa.

Si fuera el caso de iniciar con los cálculos:

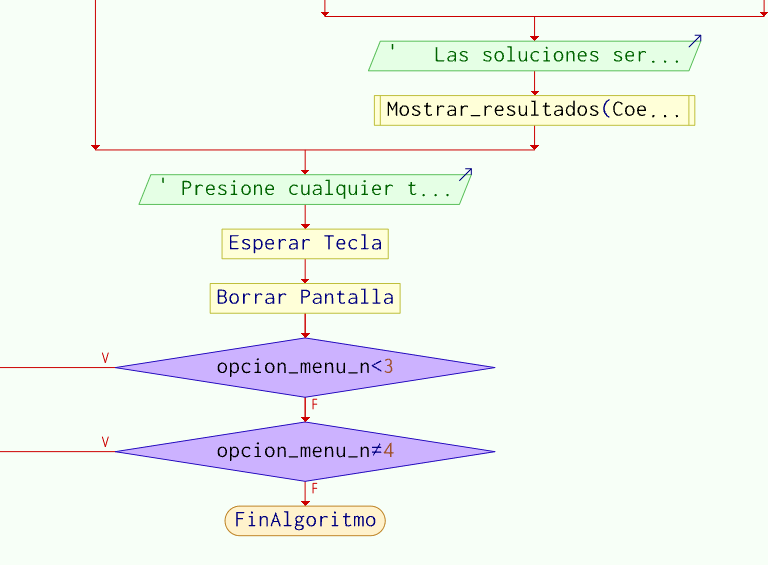
Realiza los cálculos que ya se vieron en la parte de marco teórico para este caso, se ingresa en una segunda condicional para mostrar en pantalla los procedimientos o simplemente mostrar los resultados; luego de terminar con el diagrama principal se mostrar con más detalle las funciones y operaciones se realizaron en esta parte.

****Zoom a la parte de mostrar procedimiento

Recortando la imagen

El algoritmo ingresa en distintas condicionales if{}else{}, esto simplemente para mostrar el caso correspondiente de la discriminante, para finalmente mostrar los resultados de las raíces.

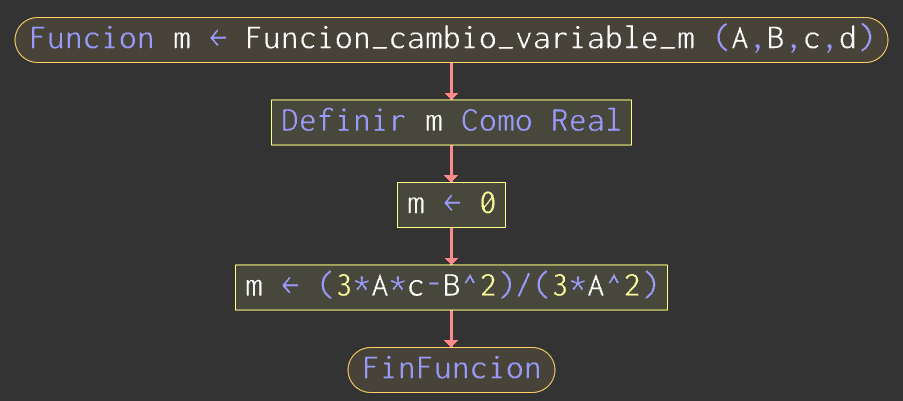
Zoom a la parte de mostrar resultados



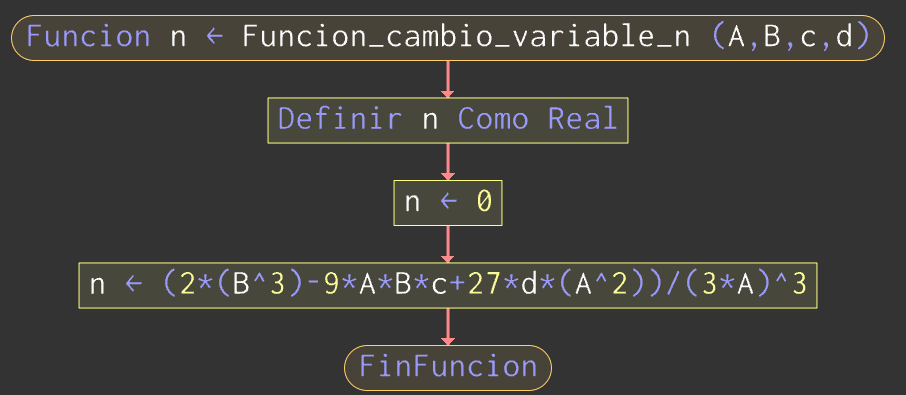
Finalmente, el algoritmo ingresa a 2 condicionales while, estos impedirán que el programa se cierre si el usuario no lo deseara.

El primer While retornara a la sección de menú ver resultados, mientras que el 2do retorna la a sección de ingresar datos.

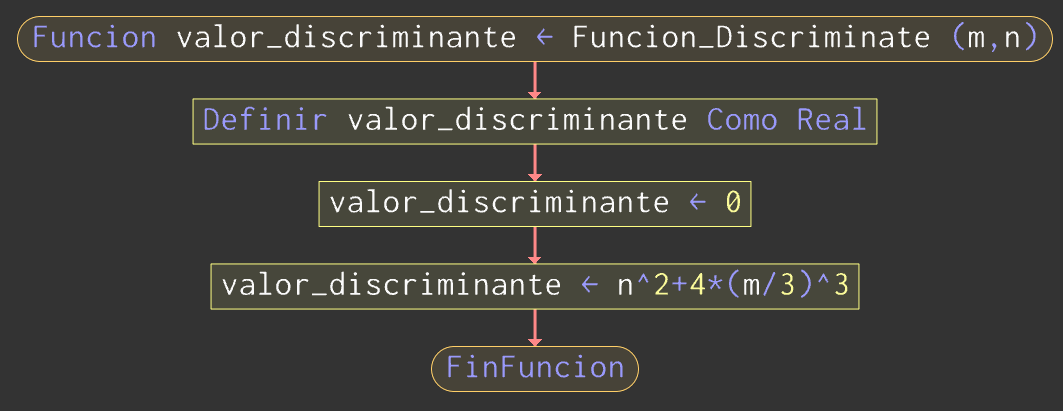
* **Funcion\_cambio\_variable\_m( , )**

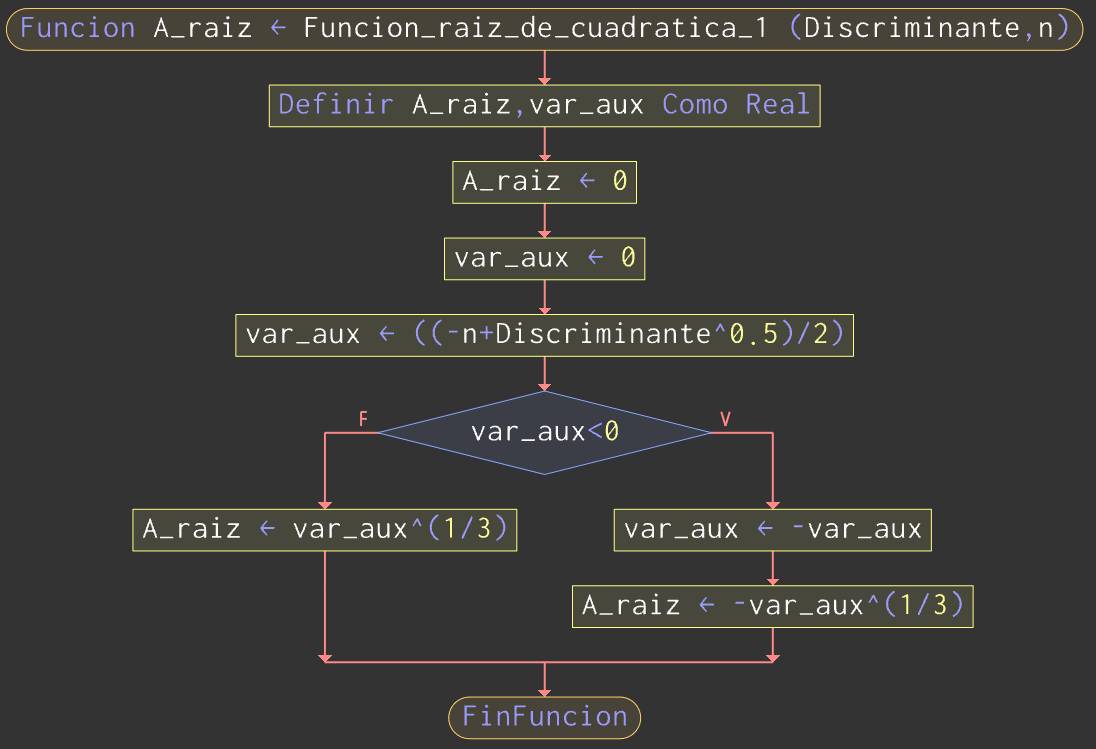


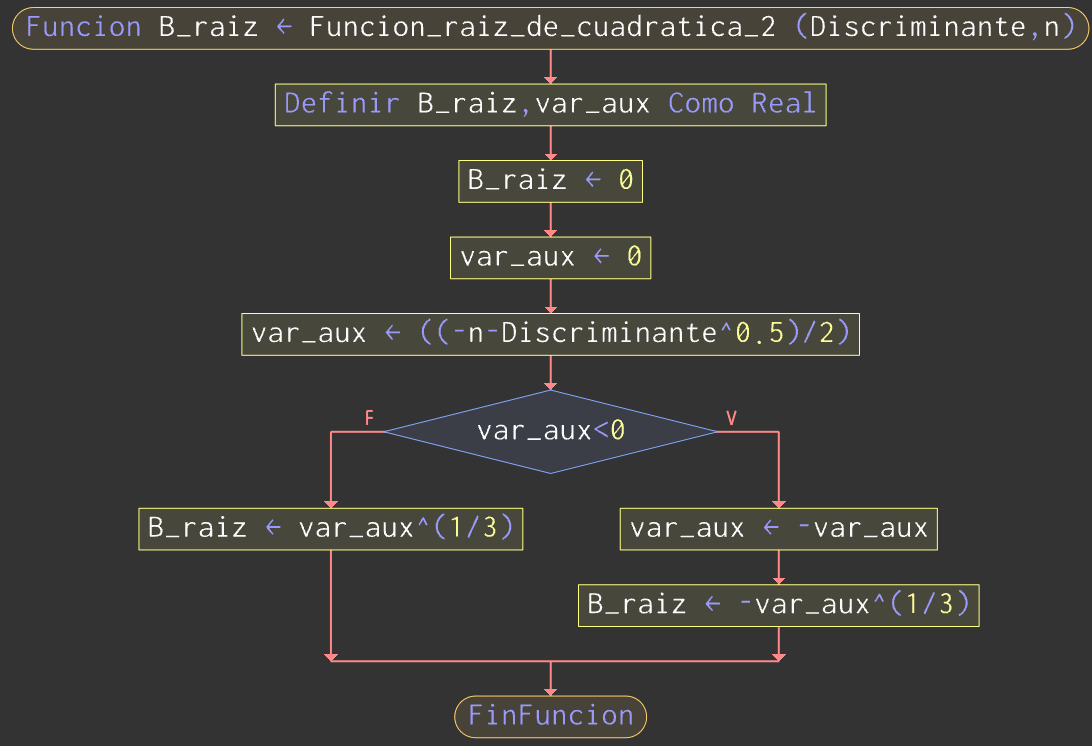
* **Funcion\_cambio\_variable\_n( , )**



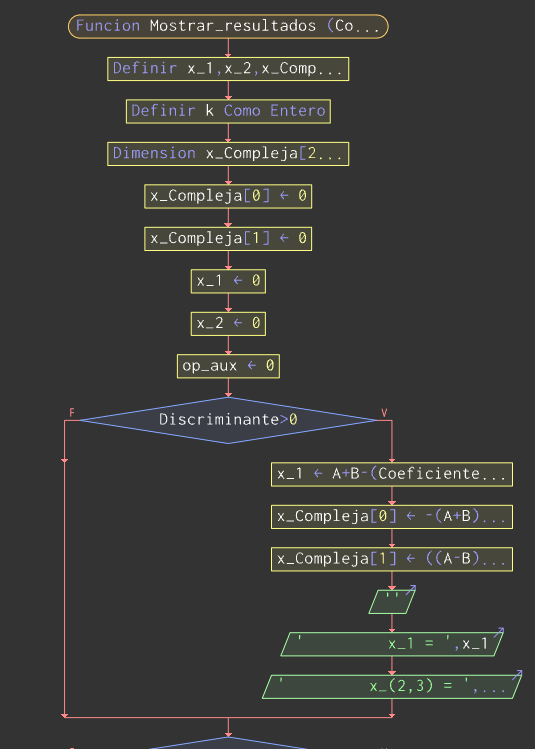
* **Funcion\_valor\_discriminante( , )**



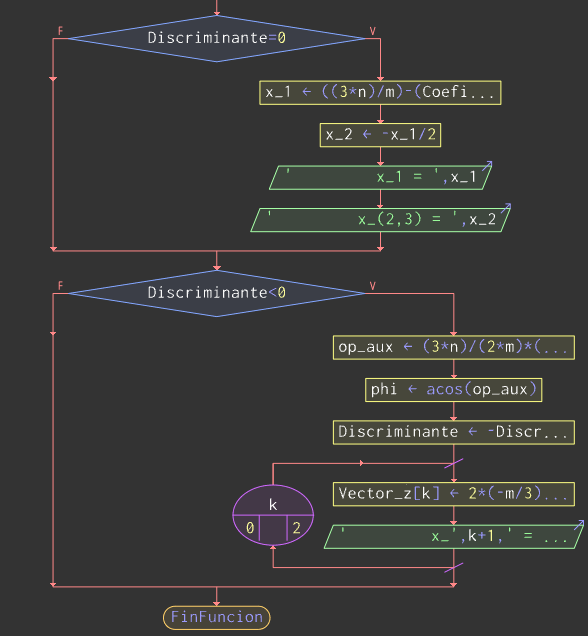
* **Funcion\_raiz\_1( , )**
* **Funcion\_raiz\_2( , )**



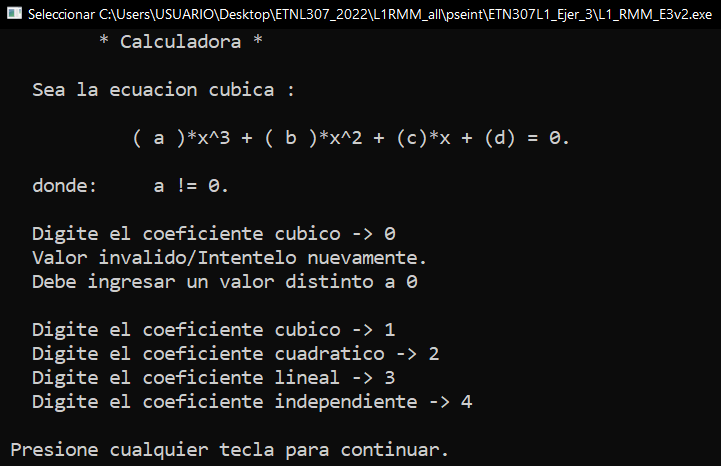
* Las funciones que se acaban de ver realizan los cambios de variable y demás operaciones que son necesarios para hallar las raíces de la ec de 3er grado.
* **Funcion\_mostar\_resultados( , )**

**Zoom a la parte superior del diagrama de flujo.**

**Zoom a la parte inferior del diagrama de flujo.**



Como su nombre lo indica, mostrara lis resultados finales de las raíces.

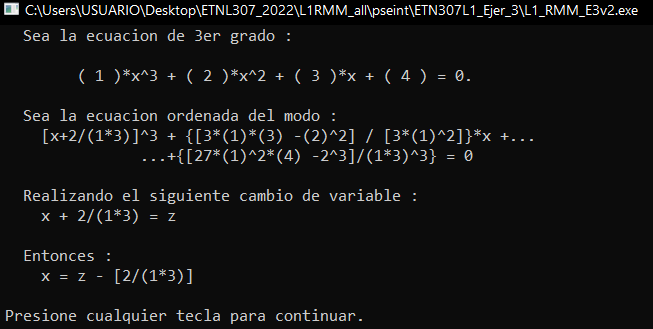
* Ingresando al programa.

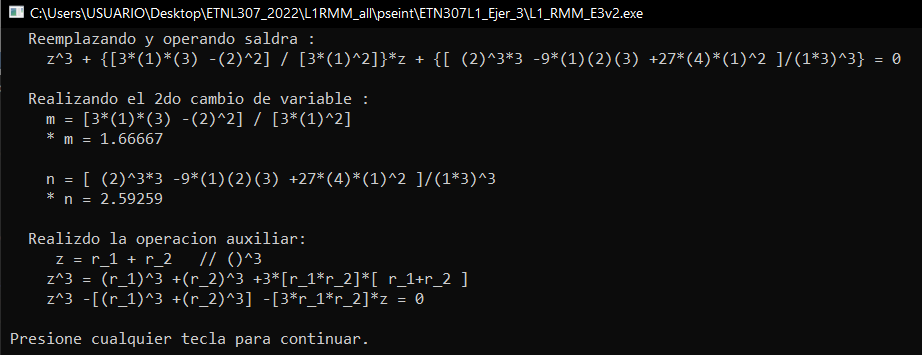
El programa inicia mostrando la forma de una ecuación de 3er grado, para luego aclarar que el valor de ‘a’ debe ser distinto de 0 y finalmente pide al usuario ingresar un valor para a; Si el usuario ingresara el valor de ‘0’ para el coef. Cubico, se mostrará el mensaje “valor invalido/Inténtelo nuevamente.” Y aclarara que el valor ingresado para el coeficiente ‘a’ debe ser distinto de ‘0’

Luego de que se ingrese algún valor valido para ‘a’, se procede a pedir los coeficientes restantes.

Presionando cualquier tecla para continuar

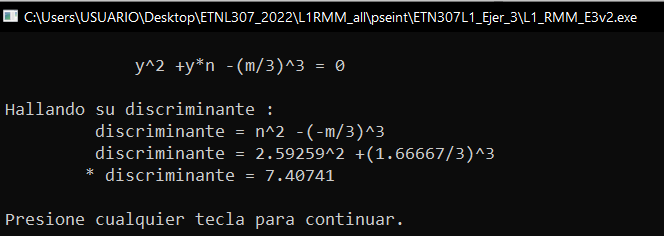
Se muestra un menú, es cual será recurrente hasta que el usuario elija la opción 3 o la 4.

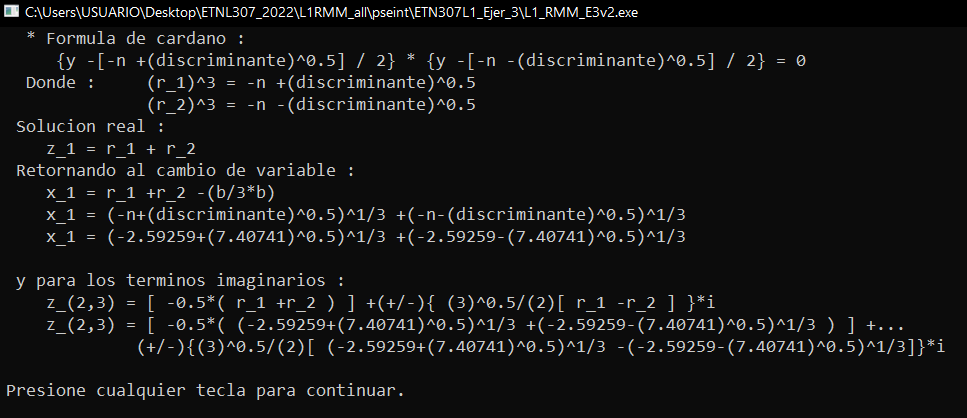
Eligiendo la 1 opción de ver procedimiento

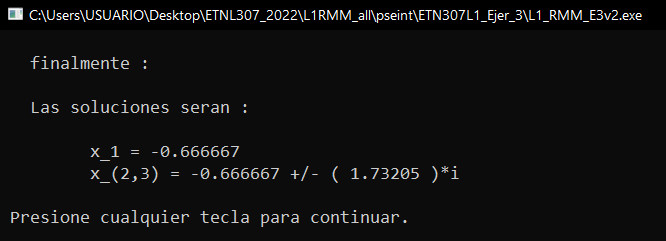
Presionando cualquier tecla para continuar

Presionando cualquier tecla para continuar

Presionando cualquier tecla para continuar

Presionando cualquier tecla para continuar

Presionando cualquier tecla para continuar

Presionando cualquier tecla para continuar

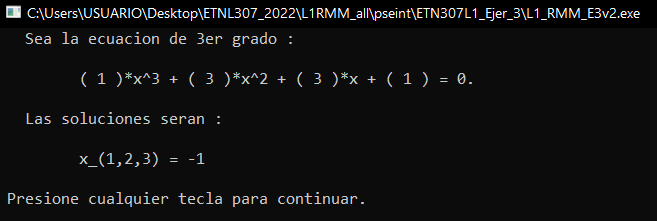
**Para las siguientes pruebas solo se tomarán capturas de pantalla a las partes importantes del procedimiento.**

* Ejemplos prácticos. –

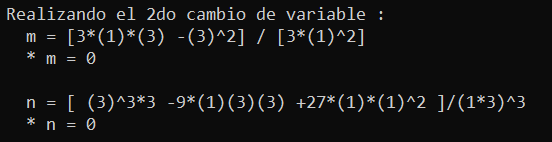
1. Caso de una raíz triple

Sea la ecuación:

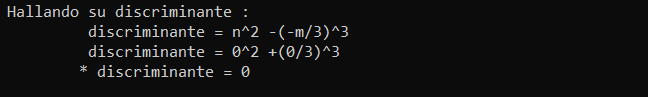
|  |
| --- |
| Resuelto por factorización tiene como raíces |



Para la opción ver procedimiento:

Donde p = m y q = n.

Luego

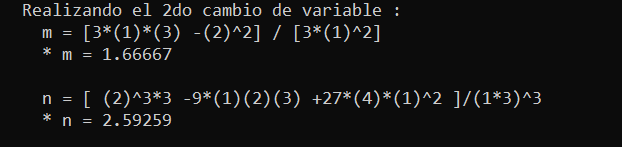


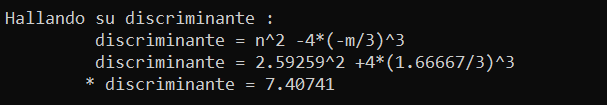
1. Si

Sea la ecuación:

|  |
| --- |
| * Valores calculados:   Resuelto por factorización tiene como raíces: |

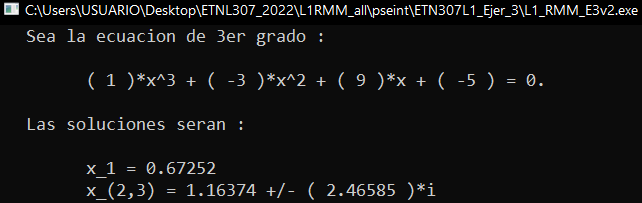


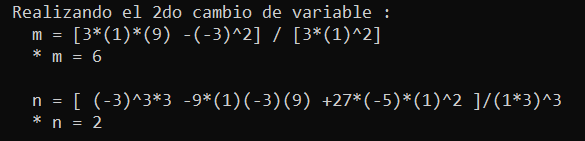
Para m y n

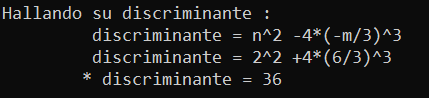
La discriminante.

Otro ejemplo ecuación:

|  |
| --- |
| Valores calculados:  Resuelto por factorización tiene como raíces: |



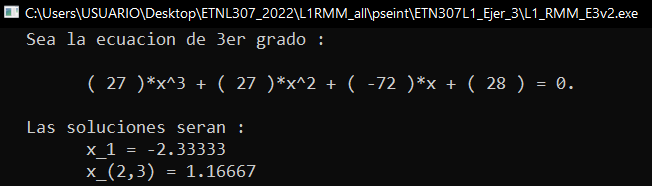
Para m y n

La discrimínate.

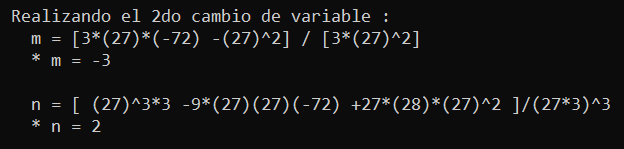
1. Si

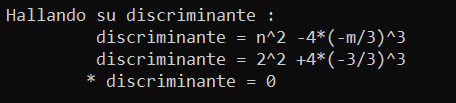
Sea la ecuación:

|  |
| --- |
| * Valores calculados:   Resuelto por factorización tiene como raíces: |



Para m y n



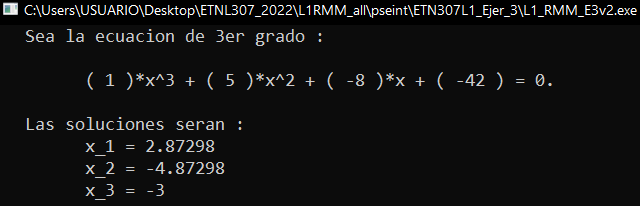
La discrimínate.

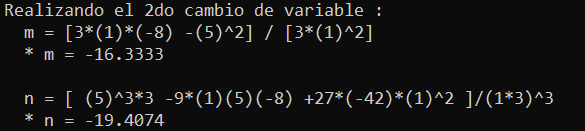


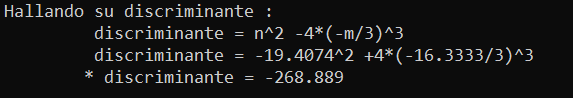
1. Si

Sea la ecuación:

|  |
| --- |
| * Valores calculados:   Resuelto por factorización tiene como raíces: |



Para m y n

La discrimínate.

Conclusiones. -

El algoritmo realizo bien todos los casos que se le presento.

Observaciones. -

 Se presentó un problema para el siguiente caso.

Esta operación se utilizó para cuando la discriminante tiene un valor positivo mayor a 0, sin embargo, como se ve en la imagen hubo que cambiar la forma de expresar la raíz cubica puesto que si se lo escribía como una división el resultado final no correspondía al valor de la raíz.