

Documentación

Entre las librerías utilizadas para el programa se usaron fundamentalmente tres:

- **Matplotlib**, que es una librería de Python para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy. Esta librería fue utilizada en nuestro programa ya que el principal objetivo de la misma era permitirle al usuario graficar funciones, y Matplotlib cuenta con comandos como `pyplot` y `subplots`, fundamentales para trabajar con planos en python.

Comandos utilizados:

- `fig, ax = plt.subplots()`
- `ax.plot(x,y)`
- `plt.title ('grafico de {0}'.format(k))`
- `fig.canvas.set_window_title('grafico de {0}'.format(k))`
- `ax.grid(True, which='both')`
- `ax.spines['left'].set_position('zero')`
- `ax.spines['right'].set_position('zero')`
- `plt.show()`

- **NumPy**, por su parte es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una librería de funciones matemáticas de alto nivel para operar con esos vectores o matrices. Hicimos uso de esta librería ya que tiene capacidades de cálculo similares a las de otros softwares como MATLAB, y ya que contiene comandos que necesitamos como *linspace*, un comando muy útil a la hora de espaciar linealmente los puntos en la cantidad de subdivisiones de igual tamaño dadas.

Comandos utilizados:

- `linspace(x0,x1,num=1000)`

- **SymPy**, es una librería para Python cuyo objetivo es reunir todas las características de un sistema de álgebra computacional, ser fácilmente extensible y mantener el código todo lo simple que sea posible. SymPy no requiere ninguna biblioteca externa, salvo para soporte gráfico. Dicha biblioteca se usó para las funciones relacionadas a los números e y π , importarla fue necesario ya que se necesitaban comandos para el manejo de enteros de precisión arbitraria y de números racionales e irracionales.

Comandos utilizados:

- `sy.N(sy.E, n)`
- `sy.simplify(ecuacion)`
- `sy.symbols('{0}'.format(variable_a_resolver))`
- `sy.solve(ecuacion,var)`

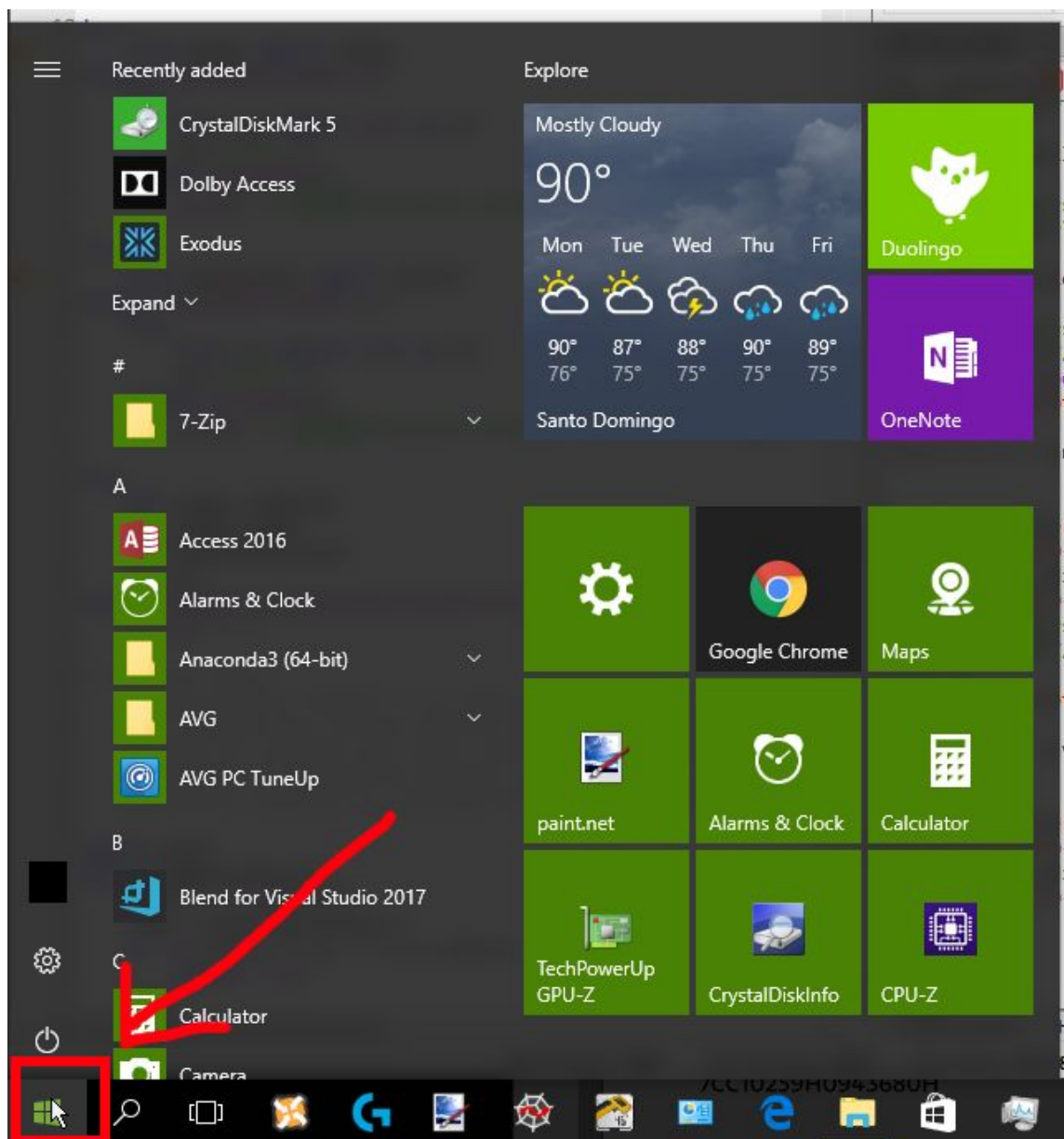
Dependencias:

- El programa requiere la instalación de Sympy, Numpy, Matplotlib, y Python 3 o superior.

****LUEGO DE INSTALAR PYTHON 3 O SUPERIOR****

A continuación instrucciones de como instalar las librerías requeridas

1- Hacemos click en el botón de windows:



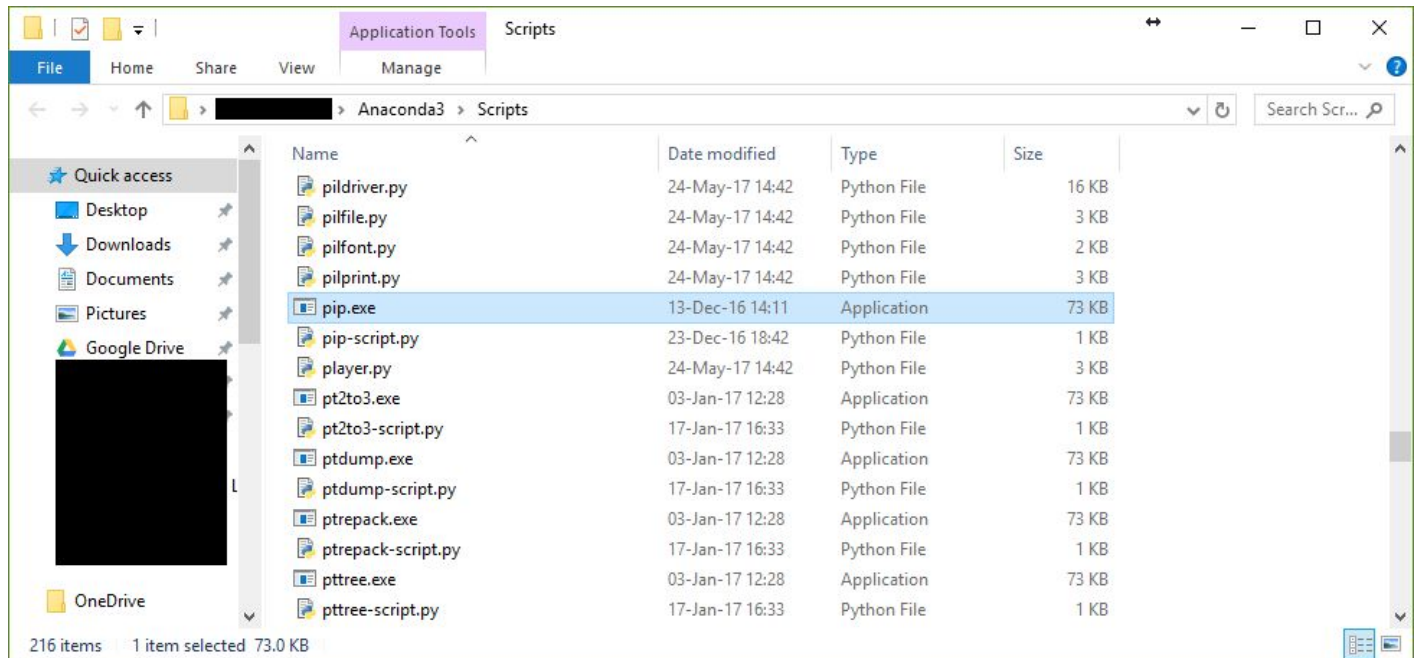
2- Escribimos “**pip.exe**” sin comillas



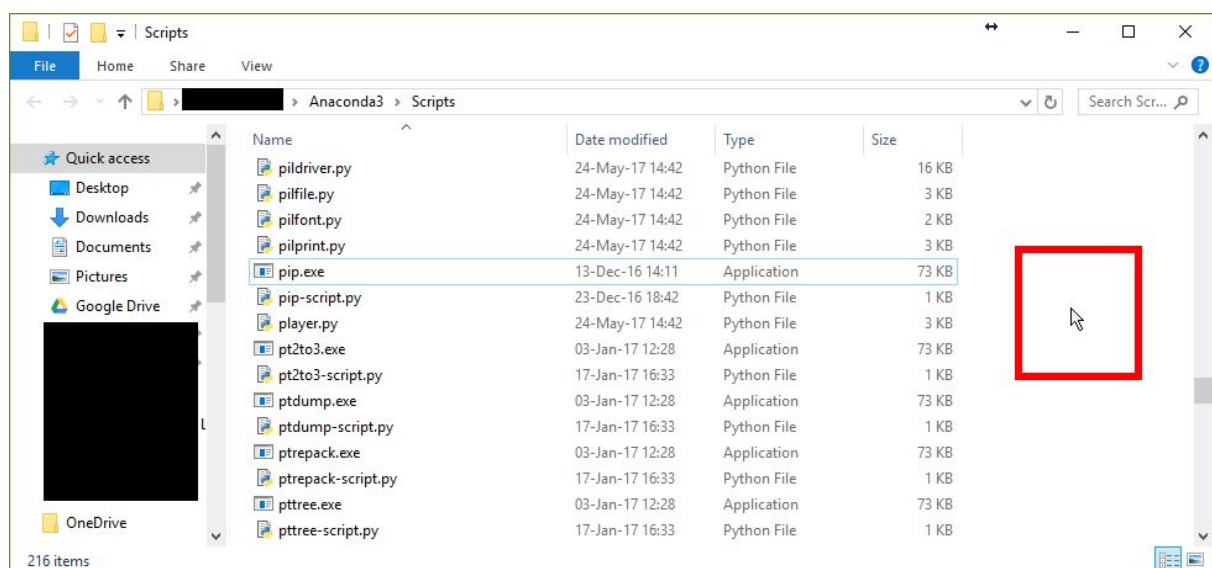
- Si no obtiene ningún resultado, probablemente no tenga python 3 instalado.

3- Hacemos click derecho en el resultado obtenido y clickeamos “Open file location” o “Abrir localización de archivo”, debe ser la única opción asumiendo que el usted tiene Windows 10.

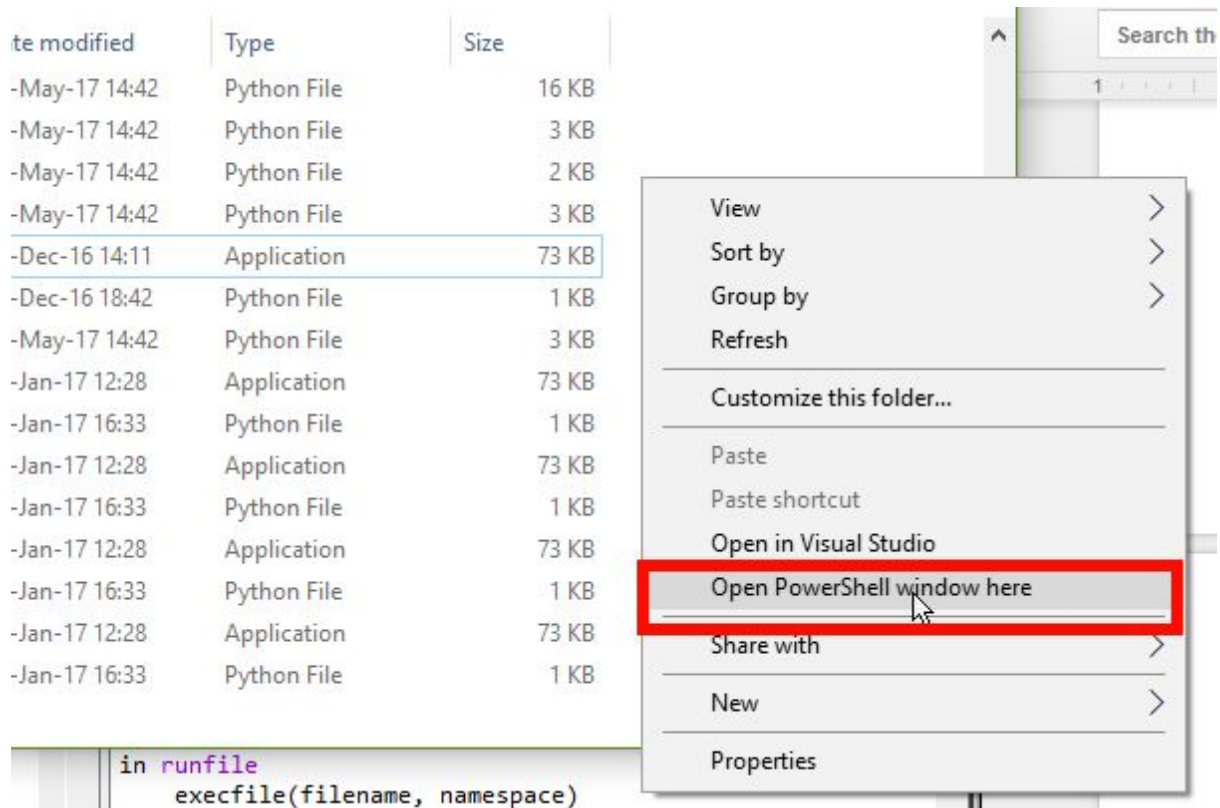
Se abrirá un folder con el archivo seleccionado.



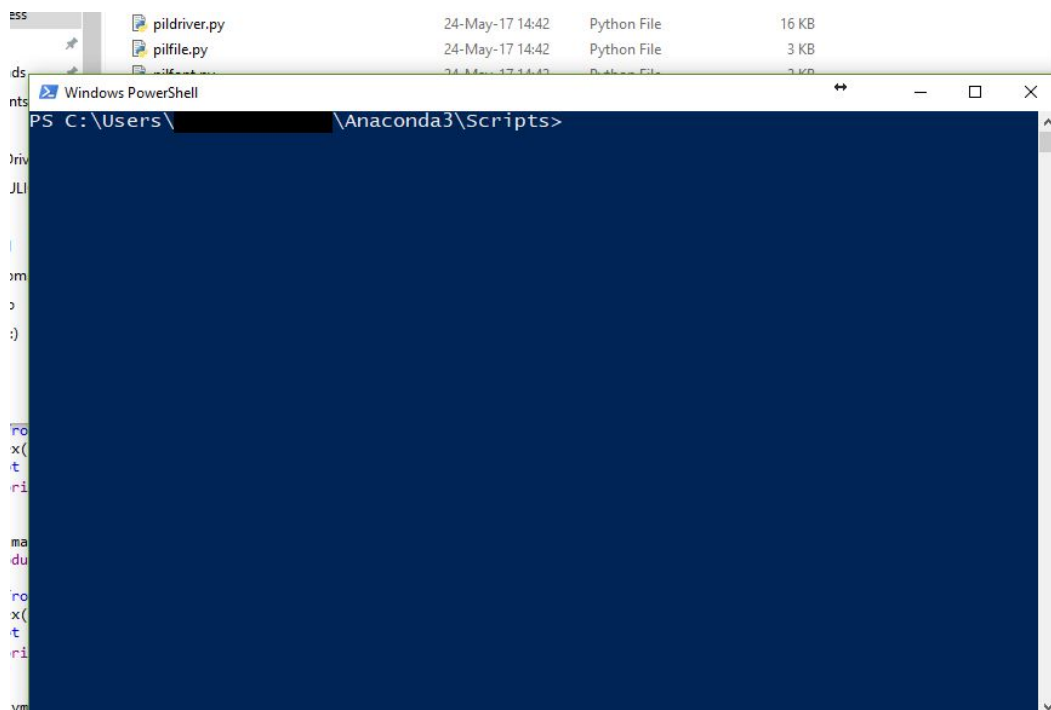
4- Presionamos en un espacio vacío de la ventana para dejar de seleccionar el archivo



5- Presionamos la tecla shift en el teclado mientras hacemos click derecho del mouse en el espacio vacío, ambos al mismo tiempo de manera que en el menú contextual nos aparezca la opción “Open Powershell Window Here” o “Abrir ventana de Powershell aquí” ó también “Open command prompt here” o “Abrir línea de comando aquí”.



6- Hacemos click en esta opción y debería aparecernos una ventana como la siguiente:



7- En la ventana de línea de comando o de powershell debemos escribir los siguientes comandos cuando la consola le presente con este caracter “>” :

“pip install sympy”

y presionar enter.

de la siguiente manera:

```
Scripts> pip install sympy
```

Luego, cuando dicho proceso termine y usted tenga nuevamente la opción de introducir otro comando, debe escribir:

“pip install numpy”

y presionar enter.

de la siguiente manera:

```
Scripts> pip install numpy
```

Esperar que el proceso anterior termine.

Y por último:

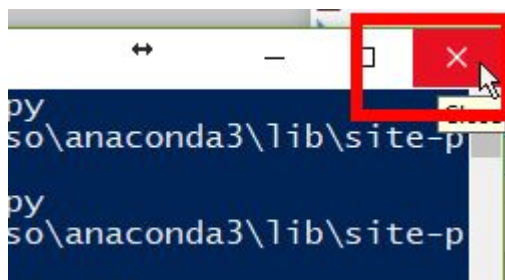
“pip install matplotlib”

y presionar enter.

de la siguiente manera:

```
Scripts> pip install matplotlib
```

Cuando el proceso anterior termine, ya estará listo para utilizar el programa, usted sabrá cuando terminan ya que se le presentará con el caracter “>” en donde usted escribe luego de que termine cada proceso. Una vez la última instalación esté lista, puede cerrar la ventana de powershell o de línea de comando.



Funciones:

- **Menú principal → Graficar**

El usuario es presentado con el siguiente menú principal:

```
In [28]: main()

Menú principal

¿Qué función desea utilizar?

1 - Graficar
2 - Resolver ecuaciones
3 - Calcular dígitos de un número irracional
4 - Cálculos simbólicos
c - Cerrar/cancelar

Introduzca la opción: |
```

Si introducimos la opción número **1**, el usuario es presentado con el menú de opciones de graficación:

Luego el usuario puede seleccionar si desea graficar alguna de las funciones presentadas anteriormente, el programa utiliza las funciones descritas individualmente más abajo en la documentación.

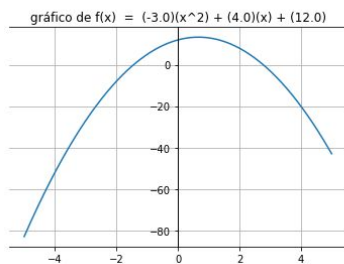
A continuación mostramos un ejemplo de cada una de las opciones:

```
Introduzca una opción: 1

Introduzca el límite inferior del intervalo a evaluar (x0): -5

Introduzca el límite superior del intervalo a evaluar (x1): 5
f(x) = a(x^2) + b(x) + c

a = -3
b = 4
c = 12
```



```
¿Desea continuar? (y,n)
```

```
(y,n) :
```

```
Introduzca la opción: 1

¿Qué función desea graficar?
```

```
1 - Función cuadrática
2 - Función cúbica
3 - Gráfica de movimiento armónico
4 - Función lineal
5 - Otra
m - volver al menú anterior
c - Cerrar/cancelar
```

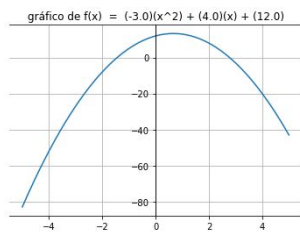
```
Introduzca una opción: |
```

La opción **1** pide al usuario la entrada de un límite inferior y un límite superior del intervalo en el que se desea graficar la función, luego muestra al usuario la forma de la ecuación que va a introducir y luego pide los coeficientes de dicha ecuación.

Luego de mostrar el gráfico, esta pregunta al usuario si desea continuar.

En caso del usuario decir que desea continuar con 'y' que debe introducir sin las comillas, el programa volverá

al menú principal como mostrado en el ejemplo siguiente:



¿Desea continuar? (y,n)

(y,n) : y

Menú principal

¿Qué función desea utilizar?

- 1 - Graficar
- 2 - Resolver ecuaciones
- 3 - Calcular dígitos de un número irracional
- 4 - Cálculos simbólicos
- c - Cerrar/cancelar

Introduzca la opción:

En el ejemplo siguiente lo mostramos:

Introduzca una opción: 2

Introduzca el límite inferior del intervalo a evaluar (x0): -5

Introduzca el límite superior del intervalo a evaluar (x1): 5

$f(x) = a(x^3) + b(x^2) + c(x) + d$

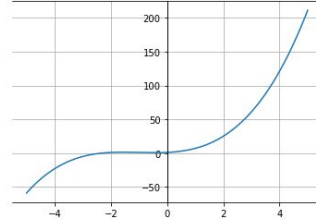
a = 1

b = 3

c = 2

d = 1

gráfico de $f(x) = (1.0)(x^3) + (3.0)(x^2) + (2.0)(x) + (1.0)$



¿Desea continuar? (y,n)

(y,n) : |

Ya que las funciones cúbicas poseen de hasta 4 coeficientes, el programa pide al usuario 4 coeficientes.

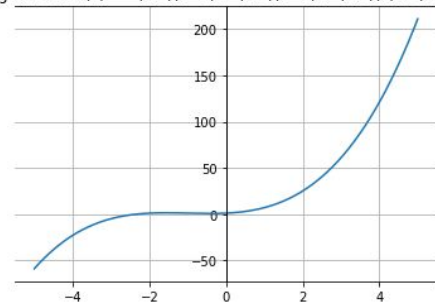
El usuario puede introducir cero en cualquiera de ellos para eliminar cualquiera de las variables.

Si el usuario introduce cero en el primer coeficiente ('a'), entonces el programa graficará una función cuadrática.

Luego de mostrar el gráfico, el usuario puede decidir si desea continuar, anteriormente mostramos el caso positivo, en el caso negativo, el programa cierra automáticamente y devuelve al usuario a la típica línea de comando. A la derecha mostramos un ejemplo de ello:

Se asume que el usuario sabe operar, al menos de una manera básica, una consola de comandos de python.

gráfico de $f(x) = (1.0)(x^3) + (3.0)(x^2) + (2.0)(x) + (1.0)$



¿Desea continuar? (y,n)

(y,n) : n

In [3]: |

La opción **3** le ofrece al usuario la posibilidad de graficar funciones de movimiento armónico, decidimos implementar movimiento armónico en vez de gráficos de funciones trigonométricas convencionales ya que ofrece mejor manipulación sobre estiramiento y cantidad de ciclos por longitud en x al graficar, esto también permite la aplicación de la función a materias como física 2, típicamente impartida en carreras de ingeniería y matemáticas.

El usuario es presentado con un menú similar a las funciones mostradas anteriormente, esta opción le pide al usuario primeramente el intervalo en el que desea graficar al función (en x) y luego el tipo de función que desea graficar, ya que podemos tener gráficos de movimiento armónico con la función seno o coseno.

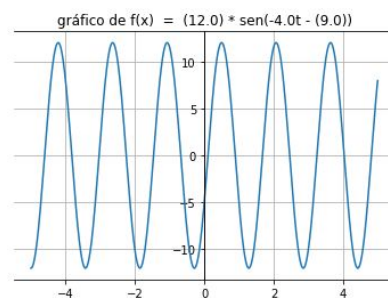
Este valor debe ser introducido como en el ejemplo de la derecha, **sin comillas**.

Luego pide al usuario la amplitud, frecuencia angular y constante de fase para realizar el gráfico y más abajo muestra el gráfico.

Introduzca una opción: 3
 Introduzca el límite inferior del intervalo a evaluar (t0): -5
 Introduzca el límite superior del intervalo a evaluar (t1): 5
 ¿Desea graficar con seno o coseno?
 "cos" para coseno y "sin"/"sen" para seno
 Introduzca la opción: sin

$$f(x) = (A) * \text{sen}(\omega t - \phi)$$

 Amplitud = 12
 Frecuencia angular = -4
 Constante de fase = 9



Típicamente, la función también le muestra al usuario la forma de la función que va a graficar para que sepa exactamente dónde va cada valor que pide en caso de que el usuario no lo sepa a primera vista. **Las constantes que se le piden al usuario están en orden de aparición en la función.**

Luego el programa pregunta al usuario si este desea continuar o no al igual que en las funciones anteriores. **Esto siempre ocurre con todas las funciones del programa.**

¿Desea continuar? (y,n)

(y,n) :

La opción **4** permite al usuario graficar funciones lineales, donde primero se le pide el intervalo de graficación al igual que en las funciones anteriores.

El la función luego procede pedirle los parámetros constantes de la función lineal, pendiente e intersección como podemos ver en el ejemplo siguiente:

Y la función luego es graficada, y, como siempre, el programa pregunta al usuario si este desea continuar.

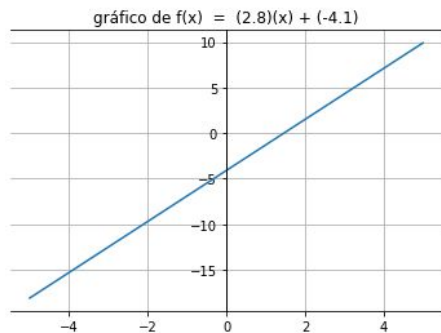
```
Introduzca una opción: 4
f(x) = m(x) + b
```

```
Introduzca el límite inferior del intervalo a evaluar (x0): -5
```

```
Introduzca el límite superior del intervalo a evaluar (x1): 5
```

```
Introduzca la pendiente de la recta (m): 2.8
```

```
Introduzca el intersección en y (b): -4.1
```



```
¿Desea continuar? (y,n)
```

```
(y,n) : |
```

La opción **5** permite al usuario graficar cualquier función que desee introducir, pero lleva una sintaxis especial, la lista de funciones que el usuario puede graficar en la forma en que debe graficarla se muestra a continuación:

Función	Descripción
p.sin(x)	seno de x
p.cos(x)	coseno de x
p.tan(x)	tangente de x
p.arcsin(x)	seno inverso de x
p.arccos(x)	coseno inverso de x
p.arctan(x)	tangente inversa de x
p.deg2rad(x)	convertir x de grado a radián
p.rad2deg(x)	convertir x de radián a grado
p.sinh(x)	seno hiperbólico de x
p.cosh(x)	coseno hiperbólico de x
p.tanh(x)	tangente hiperbólica de x
p.arcsinh(x)	seno hiperbólico inverso de x
p.arccosh(x)	coseno hiperbólico inverso de x
p.arctanh(x)	tangente hiperbólica inversa de x
p.exp(x)	función exponencial (e^x)
p.log(x)	logaritmo natural de x
p.log2(x)	logaritmo en base 2 de x
p.log10(x)	logaritmo en base 10 de x
p.power(x,y)	potencia x elevado a y (también puede escribirse $x^{**}y$)
p.divide(x,y)	cociente de x entre y (también puede escribirse x/y)
p.mod(x,y)	resto de dividir x con y ($x \bmod y$)
p.sqrt(x)	raíz cuadrada de x
p.cbrt(x)	raíz cúbica de x
p.max(x,y,z,...)	valor máximo de x,y,z,...
p.min(x,y,z,...)	valor mínimo de x,y,z,...
p.absolute(x)	valor absoluto de x

El usuario puede introducir la variable de graficación como parámetro de cualquiera de las funciones anteriores.

Existen aún más funciones pero dudamos que el usuario las necesite, en el caso contrario, las ofrece la librería de numpy en el siguiente [enlace](#). Es importante mencionar que para utilizar cualquiera de las funciones mostradas en la documentación de python el usuario debe agregar 'p.' (sin comillas) delante de la función para poder utilizarla, las demás cosas puede hacerlas con notación típica de python.

A continuación podemos visualizar un ejemplo de varias de estas funciones aplicadas a un gráfico:

Primero el programa le explica al usuario la sintaxis de manera que no existan confusiones y para evitar errores de input ya que es fácil cometerlos.

Introduzca una opción: 5

Sintaxis:

- Al escribir una función como seno debe escribir `p.sin` (de la forma `p.nombre_de_función`)
- No debe poner signos de igual ("`=`") en el input
- Los exponentes deben escribirse con "`**`" en vez de "`^`"
- Si dos elementos se multiplican, debe de poner signo de multiplicación entre ellos "`*`"
ejemplo: en vez de $3x$ debe escribir $3*x$
- Debe usar la variable "`x`", otras variables no están permitidas
- La función está automáticamente igualada a "`y`"

Luego el programa presenta al usuario con varias opciones para introducir los datos mostradas a continuación junto con el output de la aplicación:

Introduzca el límite inferior del intervalo a evaluar (`x0`): -5

Primero se pide el intervalo de graficación.

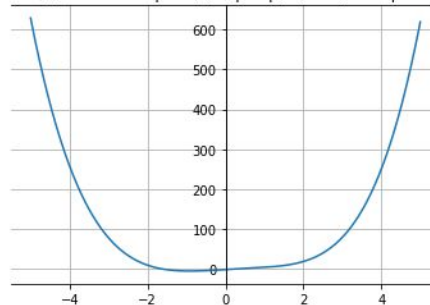
Introduzca el límite superior del intervalo a evaluar (`x1`): 5

Luego se pide la función a graficar con la sintaxis correcta.

Introduzca la función que desea graficar (despejada para `y`):
`x**4 + 5*p.sin(x) + p.exp(x-(x**2)) - 2**p.cos(p.sin(x))`

Se muestra el gráfico con su título en el intervalo especificado.

grafico de $f(x) = x^4 + 5p \cdot \sin(x) + p \cdot \exp(x - (x^2)) - 2^{p \cdot \cos(p \cdot \sin(x))}$



Luego el programa pregunta al usuario si este desea continuar.

¿Desea continuar? (`y,n`)

(`y,n`) :

Las siguientes opciones del menú de graficación son **m**, que devuelve al usuario al menú principal como podemos ver a continuación:

```
Introduzca una opción: m
Menú principal
¿Qué función desea utilizar?
1 - Graficar
2 - Resolver ecuaciones
3 - Calcular dígitos de un número irracional
4 - Cálculos simbólicos
c - Cerrar/cancelar
```

Introduzca la opción:

La última opción presenta al usuario la opción de cancelar la operación al introducir **c**.

Al cancelar la operación se le muestra al usuario la notificación de que acaba de cancelar la operación y luego lo devuelve a la línea de comando.

```
¿Qué función desea graficar?
1 - Función cuadrática
2 - Función cúbica
3 - Gráfica de movimiento armónico
4 - Función lineal
5 - Otra
m - volver al menú anterior
c - Cerrar/cancelar
```

Ilustrado en el ejemplo a la derecha:

```
Introduzca una opción: c
Operación cancelada
```

```
In [5]: |
```

- **Menú principal → Resolver ecuaciones:**

Podemos acceder la función de resolver ecuaciones introduciendo la opción **2** del menú principal:

```
Menú principal
¿Qué función desea utilizar?
1 - Graficar
2 - Resolver ecuaciones
3 - Calcular dígitos de un número irracional
4 - Cálculos simbólicos
c - Cerrar/cancelar

Introduzca la opción: 2
```


Luego de introducir la opción, el programa nos muestra la sintaxis para introducir ecuaciones a resolver, es importante recordar al usuario que **las ecuaciones están igualadas automáticamente a cero y el usuario no debe utilizar signos de igual o igualar la ecuación a nada**. Como toda ecuación lineal o no lineal permite al usuario mover todos los elementos de algún lado de la igualdad al otro, toda ecuación puede ser igualada a cero, por lo que el programa puede resolver **casi** cualquier ecuación no lineal que introduzca el usuario y absolutamente **todas** las ecuaciones lineales que el usuario introduzca.

Es importante tomar en cuenta que **demasiada** creatividad en introducción de ecuaciones no lineales puede hacer que el programa retorne un error.

A continuación muestra la sintaxis de introducción de datos:

Introduzca la opción: 2

Sintaxis:

- Debe introducir la ecuación sin "=" (signos de igual).
- La ecuación está automáticamente igualada a 0.
- Si introduce una ecuación no lineal muy compleja, el resultado será complejo y se expresará simbólicamente.
- Las soluciones se expresarán en una lista.
- El programa resuelve ecuaciones para múltiples variables, pero sólo resuelve una variable a la vez
- Al introducir la variable asegúrese de que no haya nada pegado a ella, es decir que sea sólo la variable.
- Si escribe m en el campo de ecuación sin escribir nada mas, el programa volverá al menú principal

Introduzca la ecuación que desea resolver: |

Es importante seguir esto al pie de la letra.

Luego el usuario debe introducir la ecuación con la sintaxis establecida por el programa y luego la variable cuya solución se desea obtener, como podemos ver a continuación:

El programa lista las soluciones con la palabra 'Solucion (#):' de manera individual para simplificar la salida de datos para el usuario.

(i mayúscula es la unidad imaginaria)

Sintaxis:

- Debe introducir la ecuación sin "=" (signos de igual).
- La ecuación está automáticamente igualada a 0.
- Si introduce una ecuación no lineal muy compleja, el resultado será complejo y se expresará simbólicamente.
- Las soluciones se expresarán en una lista.
- El programa resuelve ecuaciones para múltiples variables, pero sólo resuelve una variable a la vez
- Al introducir la variable asegúrese de que no haya nada pegado a ella, es decir que sea sólo la variable.
- Si escribe m en el campo de ecuación sin escribir nada mas, el programa volverá al menú principal

Introduzca la ecuación que desea resolver: $4x^{**2} - 9x + \text{sqrt}(x)$

Introduzca la variable para la cual desea resolver la ecuación: x

☞ Solucion 1: 0

☞ Solucion 2: $\frac{3}{2} - \frac{27}{(16*(-\frac{1}{2} - \text{sqrt}(3)*\frac{I}{2})*(675/64 + 27*\text{sqrt}(26)*\frac{I}{32})^{**}(\frac{1}{3}))} - (-\frac{1}{2} - \text{sqrt}(3)*\frac{I}{2})*(675/64 + 27*\text{sqrt}(26)*\frac{I}{32})^{**}(\frac{1}{3})/3$

☞ Solucion 3: $\frac{3}{2} - (675/64 + 27*\text{sqrt}(26)*\frac{I}{32})^{**}(\frac{1}{3})/3 - \frac{27}{(16*(675/64 + 27*\text{sqrt}(26)*\frac{I}{32})^{**}(\frac{1}{3}))}$

Las funciones que se pueden utilizar son muy similares a las mostradas en la función del programa que grafica cualquier función, excepto que no se utiliza 'p.' antes de escribir el nombre de la función, así que si no conoce la notación para trabajar con funciones específicas al escribir ecuaciones no lineales, puede referirse al cuadro presentado en dicha sección.

• Menú principal → Calcular dígitos de un número irracional

El programa ofrece la opción de calcular dígitos de un número irracional determinado, los números en particular que ofrece son 2 números trascendentales y la posibilidad de calcular dígitos de raíces cuadradas de números que no son cuadrados perfectos (como raíz cuadrada de 2).

En la imagen de la derecha podemos ver específicamente como se ve este menú luego de seleccionarlo.

```

._____._____.
◇ Menú principal ◇
¿Qué función desea utilizar?
1 - Graficar
2 - Resolver ecuaciones
3 - Calcular dígitos de un número irracional
4 - Cálculos simbólicos
c - Cerrar/cancelar

Introduzca la opción: 3
._____._____.

◇ Cálculo de dígitos especiales
1 - Raíz de números que no son cuadrados perfectos
2 - e (base del logaritmo natural)
3 -  $\pi$  (razón de la circunferencia y el diámetro de un círculo)
m - Volver al menú principal
c - Cerrar/cancelar

Introduzca la opción que desea:

```

La opción 1 permite calcular cifras decimales de raíces de números que no son cuadrados perfectos, en el ejemplo siguiente se ilustra el cálculo de 250 cifras decimales de raíz cuadrada de 2:

Aquí podemos ver los parámetros que la función le pide al usuario, específicamente, le pide la raíz de qué número desea calcular y le pide la cantidad de cifras decimales que desea calcular, luego los imprime y pregunta al usuario si este desea continuar.

```

Introduzca la opción: 3
._____._____.

◇ Cálculo de dígitos especiales
1 - Raíz de números que no son cuadrados perfectos
2 - e (base del logaritmo natural)
3 -  $\pi$  (razón de la circunferencia y el diámetro de un círculo)
m - Volver al menú principal
c - Cerrar/cancelar

Introduzca la opción que desea: 1

Introduzca el número cuya raíz desea calcular: 2

Introduzca la cantidad de dígitos que desea calcular
(más de 70 mil puede tomar mucho tiempo): 250

Dígitos calculados: 250 de raíz de 2

1.4142135623730950488016887242096980785696718753769480731
766797379907324784621070388503875343276415727350138462309
122970249248360558507372126441214970999358314132226659275
055927557999505011527820605714701095599716059702745345968
620147285174186408891986

¿Desea continuar? (y,n)

(y,n) : |

```


La opción **2** permite calcular cifras decimales del número irracional **e**, similar a la opción anterior, este recibe la cantidad de cifras decimales que el usuario desee calcular y este las presenta en consola para el usuario.

En el ejemplo siguiente lo ilustramos:

La función recibe solamente un parámetro que será la cantidad de cifras a calcular.

Luego, el programa pregunta al usuario si este desea continuar.

Introduzca la opción: 3

_____.

◆ Cálculo de dígitos especiales

- 1 - Raíz de números que no son cuadrados perfectos
- 2 - e (base del logaritmo natural)
- 3 - π (razón de la circunferencia y el diámetro de un círculo)
- m - Volver al menú principal
- c - Cerrar/cancelar

Introduzca la opción que desea: 2

Introduzca cuántos dígitos de e desea calcular: 135

Dígitos calculados: 135 de e

2.7182818284590452353602874713526624977572470936999595749
669676277240766303535475945713821785251664274274663919320
03059921817413596629044

¿Desea continuar? (y,n)

(y,n) : |

Introduzca la opción: 3

_____.

◆ Cálculo de dígitos especiales

- 1 - Raíz de números que no son cuadrados perfectos
- 2 - e (base del logaritmo natural)
- 3 - π (razón de la circunferencia y el diámetro de un círculo)
- m - Volver al menú principal
- c - Cerrar/cancelar

Introduzca la opción que desea: 3

Introduzca cuántos dígitos de π desea calcular: 425

Dígitos calculados: 425 de π

3.1415926535897932384626433832795028841971693993751058209749445923
078164062862089986280348253421170679821480865132823066470938446095
505822317253594081284811174502841027019385211055596446229489549303
819644288109756659334461284756482337867831652712019091456485669234
603486104543266482133936072602491412737245870066063155881748815209
209628292540917153643678925903600113305305488204665213841469519415
1160943305727036575959195309219

¿Desea continuar? (y,n)

(y,n) :

La opción **3** permite al usuario calcular cifras decimales del número irracional **π** , similar a las funciones anteriores, este recibe, solamente, la cantidad de cifras decimales que el usuario desee calcular, podemos verlo ilustrado en el ejemplo a la izquierda.

Luego de calcular las cifras que el usuario desee calcular, este pregunta al mismo si le gustaría continuar.

- **Menú principal → Cálculos simbólicos**

La opción 4 del menú principal permite al usuario realizar cálculos simbólicos, esencialmente utilizando el programa como un sistema de álgebra computarizado, donde se le retorna al usuario un cálculo algebraico de la manera que resultaría si se hiciera a mano, para facilitar la comprobación de resultados manuales o para resolver problemas dados.

Las dos opciones que se ofrecen son el cálculo de derivadas indefinidas (es decir, sin evaluar en un punto determinado), y de integrales indefinidas (sin evaluar la integral en un intervalo).

La primera opción es el cálculo de derivadas indefinidas, la función pregunta al usuario 2 parámetros: la función a derivar y la variable con respecto a la cual se deriva la función. Si la función es multivariable, esta devuelve la derivada parcial con respecto a la variable especificada.

El input de la función debe hacerse de manera similar al de la función de resolución de ecuaciones, y a diferencia de la función que grafica cualquier función dada, esta no necesita llevar 'p.' delante de utilidades como $\sin(x)$, sin embargo, esta requiere que el input sea con los mismos nombres utilizados en dicha tabla, así que para cualquier función que necesite, particularmente funciones trigonométricas, puede referirse a dicha tabla.

En el ejemplo de la izquierda podemos visualizar una utilización común de la aplicación para derivar simbólicamente.

```

Introduzca la opción: 4
*-----*
♦ Menú de cálculos simbólicos

1 - Derivar función
2 - Integrar función
m - Menú principal
c - Cancelar/cerrar

Introduzca la opción que desea: 1

Sintaxis:
- Debe introducir el valor como un string (entre comillas)
- Debe utilizar signos de multiplicacion entre cada elemento que se multiplique por otro
- NO agregue signos de igual, la funcion está automaticamente igualada a la variable dependiente
- Las funciones deben introducirse en inglés ("sin" en vez de "sen")

Introduzca la funcion que desea derivar: 3*x**4 + sin(exp(x))

Introduzca la variable con respecto a la que desea derivar: x

La derivada de 3*x**4 + sin(exp(x)) con respecto a x es:
12*x**3 + exp(x)*cos(exp(x))

¿Desea continuar? (y,n)

(y,n) :

```

La aplicación permite, también, en esta sección, integrar funciones con respecto a una variable, y, de la misma manera, si se introduce una función multivariable, esta retorna al usuario la integral parcial indefinida de la función con respecto a la variable asignada por el usuario.

Recibe los mismos parámetros que la función anterior y a la derecha podemos ver un ejemplo de la aplicación en funcionamiento.

```
Introduzca la opción: 4
'_____'.
```

◆ Menú de cálculos simbólicos

```
1 - Derivar función
2 - Integrar función
m - Menú principal
c - Cancelar/cerrar
```

```
Introduzca la opción que desea: 2
```

Sintaxis:

- Debe introducir el valor como un string (entre comillas)
- Debe utilizar signos de multiplicación entre cada elemento que se multiplique por otro
- NO agregue signos de igual, la función está automáticamente igualada a la variable dependiente
- Las funciones deben introducirse en inglés ("sin" en vez de "sen")

```
Introduzca la función que desea integrar: 7*x**3 - exp(x)
```

```
Introduzca la variable con respecto a la que desea integrar: x
```

```
La integral de 7*x**3 - exp(x) con respecto a x es:
```

```
7*x**4/4 - exp(x)
```

```
¿Desea continuar? (y,n)
```

```
(y,n) :
```

Versión anterior de la documentación, pero aún la consideramos relevante ya que explica cómo usar algunas de las funciones de manera individual si el usuario desea accionarlas individualmente:

- **Graficar Funciones Cuadráticas:**

La función recibe 2 parámetros, x0 y x1:

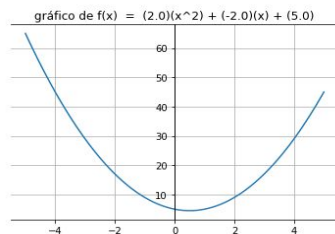
x0: es el límite inferior del intervalo [a,b] en el que se desea graficar la función.

x1: es el límite superior del intervalo [a,b] en el que se desea graficar la función.

Luego la función pide al usuario los coeficientes de la función cuadrática que se desea graficar, ejemplo:

```
In [3]: graf_funcion_cuadratica(-5,5)
f(x) = a(x^2) + b(x) + c

a = 2
b = -2
c = 5
```



Como se puede ver, la función luego permite al usuario introducir los valores de los coeficientes de la función y luego muestra el gráfico.

La función acepta cualquier valor del tipo **float**. Matemáticamente, cualquier número real. Los coeficientes también pueden ser cualquier número real.

- **Graficar Funciones Cúbicas:**

La función recibe 2 parámetros, x0 y x1:

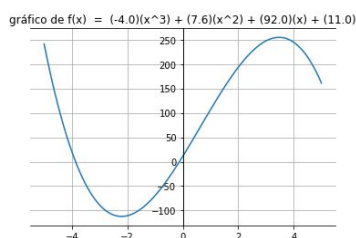
x0: es el límite inferior del intervalo [a,b] en el que se desea graficar la función

x1: es el límite superior del intervalo [a,b] en el que se desea graficar la función

Luego la función pide al usuario los coeficientes de la función cúbica que se desea graficar, ejemplo:

```
In [4]: graf_funcion_cubica(-5,5)
f(x) = a(x^3) + b(x^2) + c(x) + d

a = -4
b = 7.6
c = 92
d = 11
```



Como se puede ver, la función luego permite al usuario introducir los valores de los coeficientes de la función y luego muestra el gráfico.

La función acepta cualquier valor del tipo **float**. Matemáticamente, cualquier número real.

Los coeficientes también pueden ser cualquier número real.

- **Graficar Función Lineal:**

La función recibe 4 parámetros, x_0 , x_1 , m y b :

x_0 : es el límite inferior del intervalo $[a,b]$ en el que se desea graficar la función

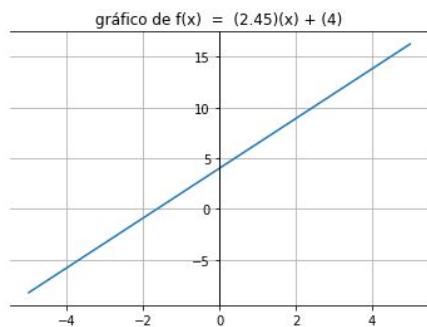
x_1 : es el límite superior del intervalo $[a,b]$ en el que se desea graficar la función

m : es la pendiente de la recta ($y = m \cdot x + b$)

b : es el intersepto de la recta ($y = m \cdot x + b$)

Ejemplo:

```
In [8]: graf_funcion_lineal(-5,5,2.45,4)
f(x) = m(x) + b
```



La función acepta cualquier valor del tipo **float**. es decir (matemáticamente), cualquier número real.

Los coeficientes también pueden ser cualquier número real

- **Graficar Movimiento Armónico:**

La función recibe 3 parámetros, x_0 , x_1 , \cos_o_sen :

x_0 : es el límite inferior del intervalo $[a,b]$ en el que se desea graficar la función

x_1 : es el límite superior del intervalo $[a,b]$ en el que se desea graficar la función

\cos_o_sen : es la función de movimiento armónico que se desea graficar

El parámetro \cos_o_sen sólo recibe los siguientes valores:

Para seno:

“sen”, “sin”, “Sen”, “Sin”.

Para coseno:

“cos”, “Cos”.

El que sea que se introduzca debe introducirse **con** las comillas.

Ejemplo:

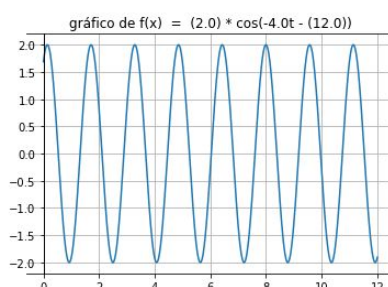
```
In [12]: graf_mov_armonico(0,12,'cos')
```

$f(x) = (A) * \cos(\omega t - \phi)$

Amplitud = 2

Frecuencia angular = -4

Constante de fase = 12



En este ejemplo podemos ver cómo se introducen los 3 parámetros principales y el output de la función.

- **Calcular dígitos decimales de e:**

La función recibe un parámetro (n):

n: Este parámetro indica la cantidad de valores decimales que se quieren conseguir de e. Este valor debe ser de tipo entero, o de lo contrario se producirá un error.

Ya que la consola puede tomar un tiempo para calcular dichos valores, este parámetro tiene una advertencia al intentar calcular 70,000 dígitos decimales. Es posible hacerlo, pero no es recomendable si no posee de un procesador con la suficiente capacidad.

Ejemplo:

```
In [9]: digitos_e(300)
Out[9]:
2.7182818284590452353602874713526624977572470936999595749669676277240
766303535475945713821785251664274274663919320030599218174135966290435
729003342952605956307381323286279434907632338298807531952510190115738
341879307021540891499348841675092447614606680822648001684774118537423
4544243710753907774499207
```

En el ejemplo mostrado se puede visualizar el cálculo de 300 cifras decimales de e.

Ejemplo de la advertencia al usuario al calcular demasiados dígitos:

```
In [17]: digitos_e(1e5)

¿El proceso puede tardar mucho, seguro que desea continuar?

¿Desea continuar? (y,n) : y
Out[17]:
2.7182818284590452353602874713526624977572470936999595749669676277240
766303535475945713821785251664274274663919320030599218174135966290435
729003342952605956307381323286279434907632338298807531952510190115738
341879307021540891499348841675092447614606680822648001684774118537423
454424371075390777449920695517027618386062613313845830007520449338265
602976067371132007093287091274437470472306969772093101416928368190255
151086574637721112523897844250569536967707854499699679468644549059879
316368892300987931277361782154249992295763514822082698951936680331825
```

- **Calcular dígitos decimales de π :**

Al igual que en la función anterior, ésta función recibe un solo parámetro (n):

n: Este parámetro indica la cantidad de valores decimales que se quieren conseguir del valor de π . Debe ser de tipo entero.

Ya que la consola puede tomar un tiempo para calcular dichos valores, este parámetro tiene una advertencia al intentar calcular 70,000 dígitos decimales. Es posible hacerlo, pero no es recomendable si no posee de un procesador con la suficiente capacidad.

```
In [19]: digitos_pi(220)
Out[19]:
3.1415926535897932384626433832795028841971693993751058209749445923078
164062862089986280348253421170679821480865132823066470938446095505822
317253594081284811174502841027019385211055596446229489549303819644288
10975665933446
```

En el ejemplo mostrado, podemos ver el cálculo de 220 cifras decimales de pi.

La advertencia al intentar calcular más de 70,000 o 70,000 dígitos sería igual que en el ejemplo anterior.

- **Calcular dígitos decimales de Cuadrados no perfectos:**

Acepta dos parámetros:

a: Es el valor dentro de la raíz, que se desea evaluar, puede ser de tipo flotante, o entero.

n: Es la cantidad de dígitos decimales que desean obtenerse. Este dato sólo puede ser entero.

Ya que la consola puede tomar un tiempo para calcular dichos valores, este parámetro tiene una advertencia al intentar calcular 70,000 dígitos decimales. Es posible hacerlo, pero no es recomendable si no posee de un procesador con la suficiente capacidad.

Ejemplo

```
In [26]: digitos_sqrt_irr(2,250)
Out[26]:
1.4142135623730950488016887242096980785696718753769480731766797379907
324784621070388503875343276415727350138462309122970249248360558507372
126441214970999358314132226659275055927557999505011527820605714701095
59971605970274534596862014728517418640889199
```

En el ejemplo anterior podemos ver el cálculo de raíz de 2, un número irracional con infinitos dígitos después del punto decimal. En el ejemplo anterior se calculan 250 dígitos luego del punto decimal.

Si el usuario introduce en el parámetro a un número que sea un cuadrado perfecto, entonces el programa va a imprimir solamente una cifra decimal, ya que si imprimiese 250 como en el ejemplo anterior, imprimiría 250 ceros luego del punto, lo que sería molesto en consola.

Ejemplo:

```
In [27]: digitos_sqrt_irr(4,250)
Out[27]: 2.0
```

- **Calcular Derivadas simbólicamente:**

Acepta dos parámetros:

Función: Se refiere a la función que el usuario desea derivar, y debe estar escrita entre comillas simples o dobles ya que recibe el tipo de dato como string. No puede igualarse la función a un número/constante.

Variable: Éste parámetro se refiere a la variable respecto a la que se va a derivar la función. También es un tipo de dato string, y no puede tener signo de igual ya que la función introducida está automáticamente igualada a la variable dependiente.

Dependiendo de la cantidad de variables en la función introducida, esto podría ser y, z, etc... sin embargo, la función devuelve el resultado expresado en términos de x, por lo que la variable dependiente sería irrelevante en el cálculo asumiendo que el usuario sabe lo que está calculando. En el ejemplo más abajo lo podemos visualizar.

```
In [28]: derivar_funcion('sin(x) + exp(cos(x))', 'x')
Out[28]: -exp(cos(x))*sin(x) + cos(x)
```

En el ejemplo anterior podemos ver el cálculo de **sen(x) + e^{cos(x)}**

Ejemplo 2:

```
In [41]: derivar_funcion('9*x**(-4) + sin(7*x**5)', 'x')
Out[41]: 35*x**4*cos(7*x**5) - 36/x**5
```

Al introducir valores como $9x^{-4}$, el usuario debe poner signo de multiplicación entre cada variable que se multiplique, de manera que el sa

Debe usarse notación de funciones en inglés, por ejemplo 'sin' en vez de 'sen'.

- **Calcular Integrales simbólicamente:**

Acepta dos parámetros:

Función: Se refiere a la función que el usuario desea integrar, y debe estar escrita entre comillas simples o dobles ya que recibe el tipo de dato como string. No debe utilizarse el signo de igual ya que la función introducida está igualada a la variable dependiente automáticamente.

Variable: Éste parámetro se refiere a la variable respecto a la que se va a integrar la función. También debe estar escrito con comillas ya que es dato tipo string, en este campo debe sólo introducirse un elemento, si se introduce más de un elemento, la función devuelve un error.

Dependiendo de la cantidad de variables en la función introducida, esto podría ser y, z, etc... sin embargo, la función devuelve el resultado expresado en términos de x, por lo que la variable dependiente sería irrelevante en el cálculo asumiendo que el usuario sabe lo que está calculando. En el ejemplo más abajo lo podemos visualizar.

```
In [4]: integrar_funcion('3*x**2 + 4*x', 'x')
Out[4]: x**3 + 2*x**2
```

Si el usuario es **demasiado creativo** puede ser que la función no pueda calcular la integral introducida, por ejemplo:

```
In [29]: integrar_funcion('sqrt(tan(x))','x')
Out[29]: Integral(sqrt(tan(x)), x)

In [30]: integrar_funcion('exp((tan(x)))','x')
Out[30]: Integral(exp(tan(x)), x)

In [31]: integrar_funcion('x**exp(x)','x')
Out[31]: Integral(exp(exp(x)*log(x)), x)

In [32]: integrar_funcion('x**log(exp(x))','x')
Out[32]: Integral(x**log(exp(x)), x)

In [33]: integrar_funcion('x**sqrt(x)','x')
Out[33]: Integral(x**(sqrt(x)), x)
```

Como podemos ver en el ejemplo anterior, la función no llega a calcular las **creativas** integrales introducidas, entonces devuelve un valor simbólico para dicha integral que no representa la respuesta.