

**ITESO**

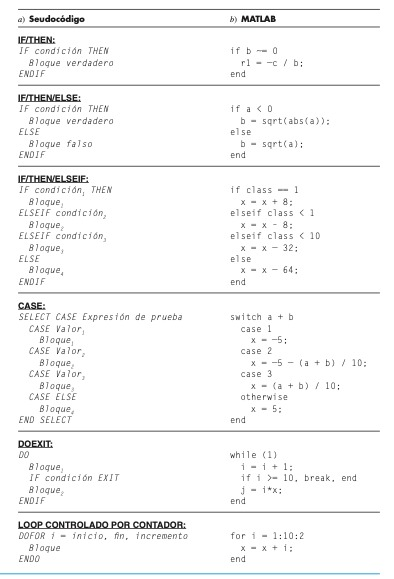
**MÉTODOS NUMÉRICOS**

**LÓPEZ LAZARENO DIEGO ALBERTO IF722100**

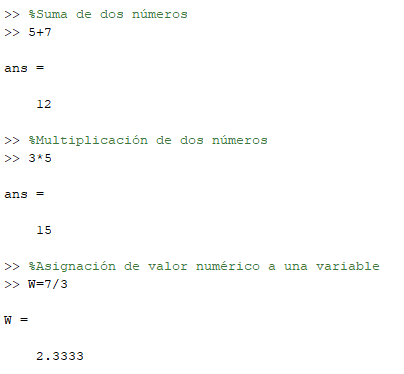
**PRÁCTICA 01**

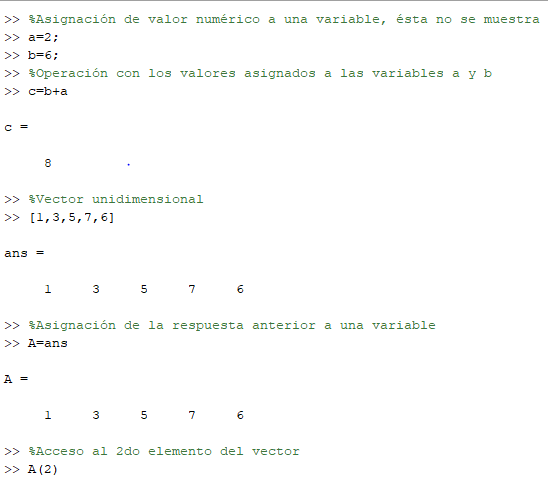
MATLAB es un software de programación en el que se pueden declarar variables que contengan valores numéricos como enteros y flotantes. Además, con estas variables se pueden hacer distintas operaciones aritméticas como sumas, restas, multiplicaciones, divisiones y potencias. También se pueden hacer arreglos unidimensionales y bidimensionales, con los que algunas operaciones vectoriales y matriciales están definidas.

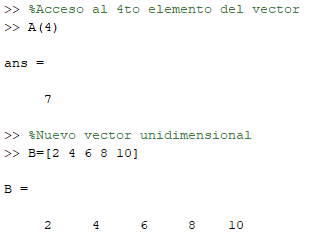
**Rutinas de Matlab**

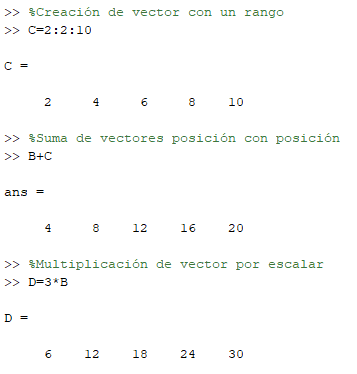


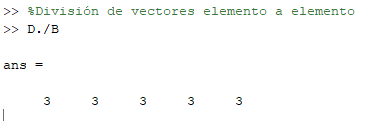
1. Describe qué hace cada operador que utilizaste en el ejercicio anterior

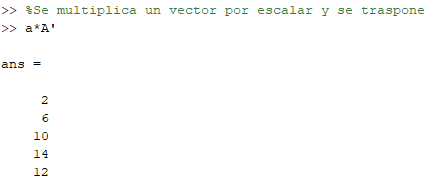


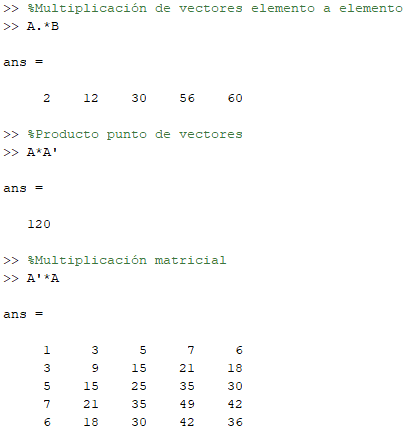


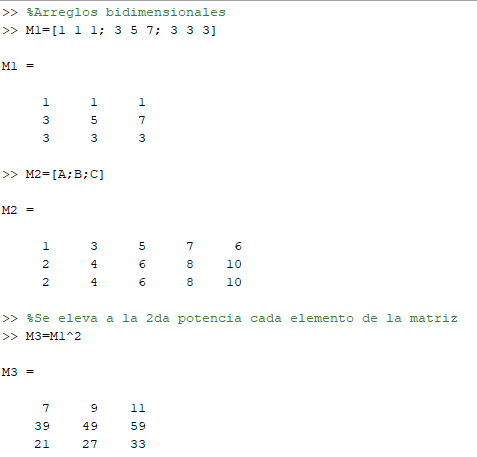


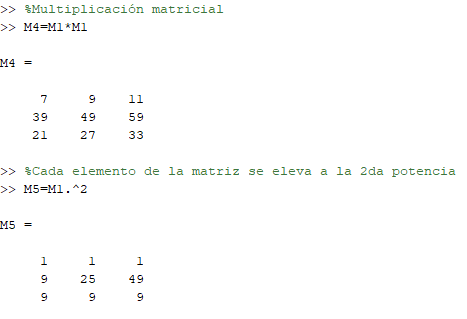


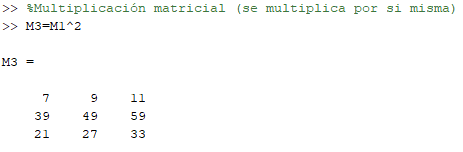




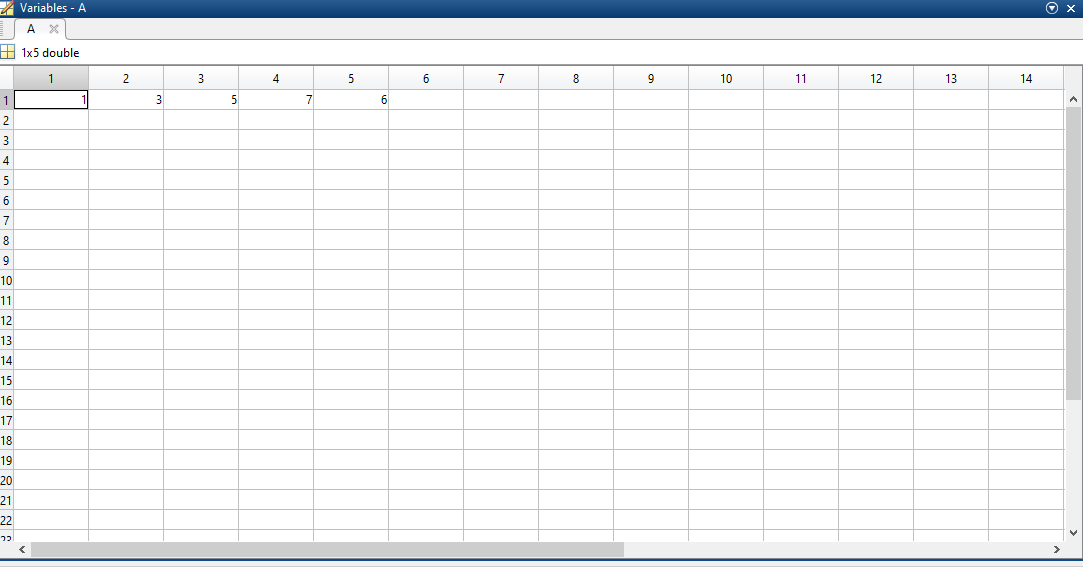








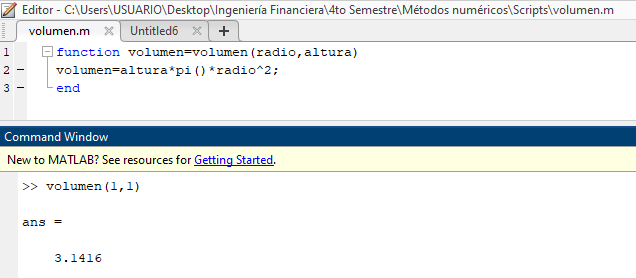
1. Activa la pestaña área de trabajo y haz doble click en cada una de las variables que se muestran, describe que muestra esta ventana (editor de arreglos).



MATLAB cuenta con una ventana de variables que despliega los arreglos en celdas de Excel para facilitar su visualización.

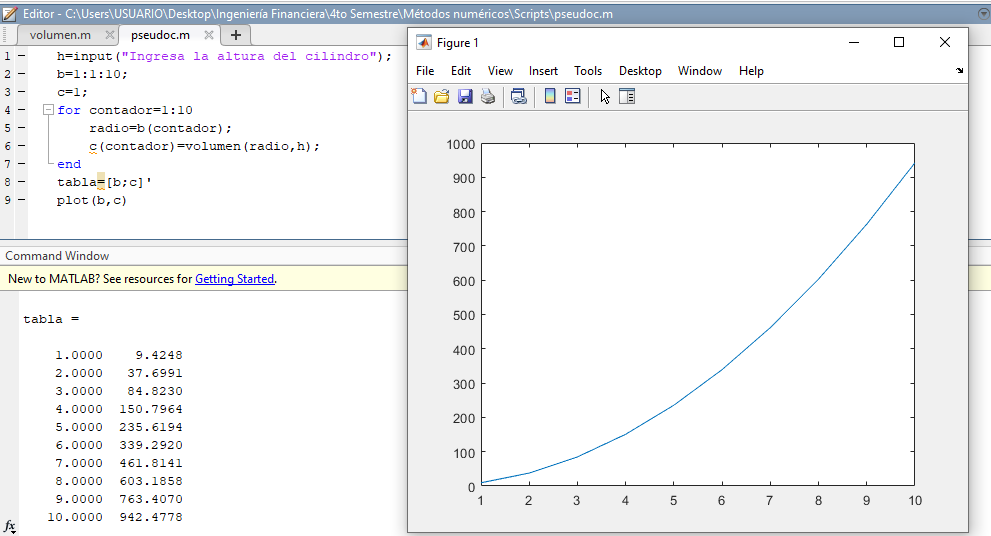
1. Identifica las estructuras de programación en los pseudocódigos y da una explicación a qué hacen.

Función que calcula el volumen de un cilindro



El pseudocódigo anterior se caracteriza por emplear una estructura de programación bastante conocida: la función. Ésta trabaja como una caja a la que damos atributos y nos entrega un resultado. La función que hemos creado nos arroja el volumen de un cilindro dependiendo de los parámetros que ingresemos desde la ventana de comandos para el radio y la altura del mismo.

Ciclo que calcula el volumen de un cilindro con distintos radios y altura constante

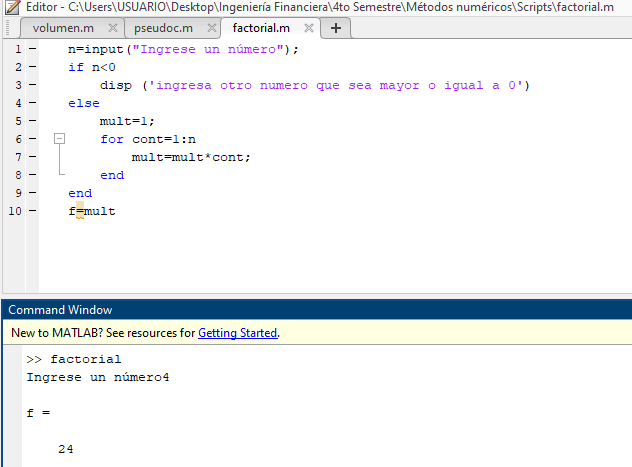


La estructura de programación que utiliza el anterior pseudocódigo es el ciclo for. Dicha estructura iterará líneas de código tantas veces como lo deseemos para realizar una labor. En este caso hemos programado un ciclo que calcula el volumen de un cilindro con un radio distinto para cada iteración. He aquí la ventaja de los ciclos, pues no necesitamos repetir líneas y líneas de código para cada cálculo. Al final se elabora una gráfica que muestra cómo cambia el volumen del cilindro a lo largo de las iteraciones.

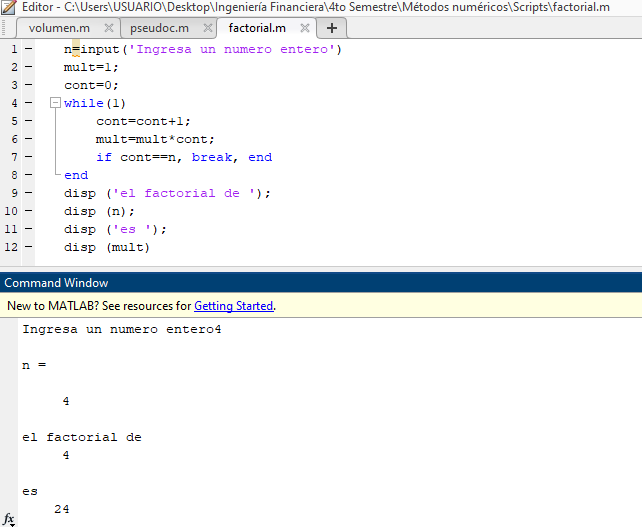
1. Identifica las diferencias entre los programas.

La diferencia entre los siguientes programas es que estos calculan el factorial de un número ayudándose con distintos tipos de ciclos: for y while. El primero itera las veces preestablecidas para ejecutar su labor, mientras que el segundo necesita que se cumpla una condición para romperse y dar por finalizada la tarea.

Cálculo del factorial de un número con ayuda del ciclo for



Cálculo del factorial de un número con ayuda del ciclo while

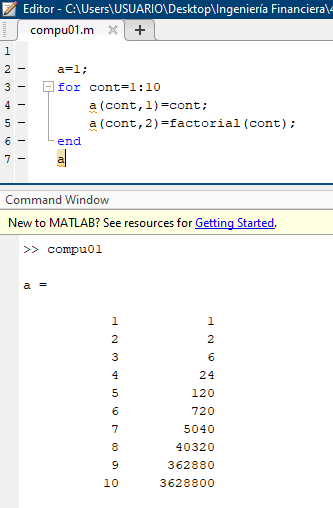


1. ¿Cuáles son los comandos y estructuras de programación que identificas?

Los comandos que identifico son los ciclos (while y for), los condicionales (if y else) y la palabra reservada break. Por una parte, el while y el for nos ayudan en programación para resolver una tarea ejecutando repetidas veces el código. A su vez, el if y else son instrucciones a través de las cuales se ejecutarán líneas de código si su condición se cumple. Por último, el break sirve para terminar un ciclo cuando se ha cumplido cierta sentencia condicional.

1. Modifica el programa de manera que muestre la tabla que realizaste en la clase anterior para calcular una aproximación para el seno de un ángulo.

Factorial de los números del 1 al 10



%Aproximación del seno de un ángulo mediante series de Taylor

a=1;

angulo=input("Ingrese el ángulo que desea aproximar")

valor\_verdadero=sin(angulo)

angulo\_anterior=0;

for n=1:10

a(n,1)=n;

n=n-1;

angulo\_actual=angulo\_anterior+(((-1)^n)/(factorial(2\*n+1)))\*(angulo^(2\*n+1));

n=n+1;

a(n,2)=angulo\_actual;

angulo\_anterior=angulo\_actual;

end

for n=1:10

a(n,3)=abs(valor\_verdadero-a(n,2));

a(n,4)=abs(((valor\_verdadero-a(n,2))/valor\_verdadero)\*100);

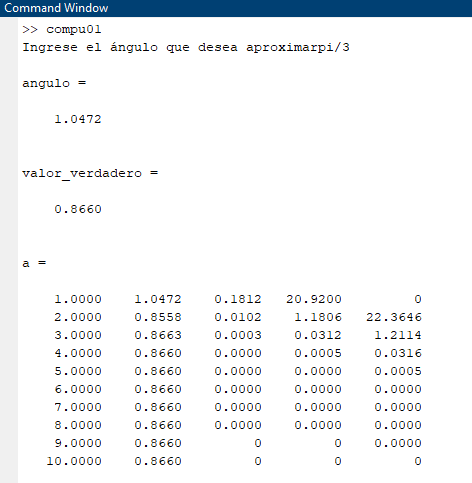
if n>1

a(n,5)=abs((a(n,2)-a(n-1,2))/a(n,2)\*100);

end

end

a



**Conclusión**

En esta práctica de laboratorio se conocieron los fundamentos básicos de la programación en MATLAB. Tales habilidades nos permiten realizar rutinas y pseudocódigos que resuelven tareas que, en veces, se vuelven repetitivas y tediosas. En concreto, el alumno realizó por cuenta propia un pseudocódigo que aproxima de forma satisfactoria el seno de un ángulo mediante polinomios de Taylor.