

Tarea 5: Capitulo 1, Tesis (Lista de reproduccion en Youtube)

William José, Sicán Hernández, 202002896^{1,*}

¹Facultad de Ingeniería, Universidad de San Carlos, Ciudad Universitaria, Zona 12, Guatemala.

I. DESCRIPCION DE LA TAREA

Realizar los ejercicios mostrados en el capitulo 1 del canal de la tesis https://www.youtube.com/watch?v=H-GCj6TmuUA&list=PLNS5_VEpRfRlyEH0BVm90NBxMjL_F0n4E

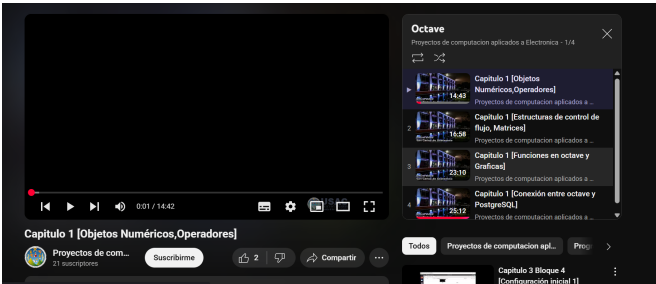


Figura 1: Lista de reproduccion del Capitulo 1

II. CODIGOS

Video 1

```
1 ##objetos numericos
2 ##
3 ##375 # numero escrito normalmente
4 ##3.75e2 # numero con notacion formato 1
5 ##3.75e2 # numero con notacion formato 2
6 ##0x177 #numero escrito en hexadecimal
7 ##
8 ##rangos numericos
9 ##$intaxis:
10 ##numero_inicial:salto:numero_final ##### si no se escribe el salto octa lo interpreta como salto 1
11 ##10:5:10 ### los numeros superiores al rango nunca se mostraran
12 ##
13 ##Matrices, se almacena en una variable
14 ##M = [1,2,3;4,5,6;7,8,9] #se declara por filas las comas separa a los datos de las filas y los ";" separa a las filas
15 ##
16 ##N = (1;3;4;6;7;9) #tambien se puede declarar con rangos numericos
17 ##
18 ##String o cadena de caracteres
19 ## se declara entre comillas simples o dobles
20 ##
21 ##"Hola Mundo"
22 ##"Hola Mundo \a" # el simbolo \a hace que la computadora produzca un sonido
23 ##
24 ##estructuras, siempre se asignan a una variable
25 ## las estructuras son "variables" que almacenan datos de distintos tipos por secciones
26 ##x = ()
27 ##x.secuencia = 1:5
28 ##x.matriz = [1,2;3,4]
29 ##x.string = "secuencia numerica"
30 ##x.estructura = ()
31 ##x.estructura.numero = 0x177
32 ##x.estructura.lectura = "A"
33 ##x
```

Figura 2: Video 1 P1

```
34 ----
35 ##tipos de operadores, devuelven un 1 o 0 dependiendo el resultado falso o verdadero
36 ##z=2
37 ##y=3
38 ##z + y
39 ##z - y
40 ##z * y
41 ##z / y
42 ##++z # z = z + 1
43 ##--z # z = z - 1
44 ##z < y
45 ##z <= y
46 ##z == y
47 ##z > y
48 ##z >= y
49 ##z != y #z diferente a y
50 ##z=1
51 ##y=0
52 ##z & y #and
53 ##z | y #or
54 ##not(z)
```

Figura 3: Video 1 P2

Video 2

```
1 ##estructuras de control
2 #cont
3 #y = 0
4 #z = 5
5 #if (x > y && z < 0) #condicion anidada
6 # # z es mayor a y'
7 #elseif (x == y || z < 0)
8 # # 'x y y son iguales'
9 #else
10 # # 'y es mayor a x'
11 #endif
12 ##
13 ## siempre se ejecutara la primera opcion que encuentre valida aunque mas adelante exista otra que tambien sea valida
14 ##
15 ##2da estructura while
16 ##
17 ##while(z < y) # mientras que z sea mayor a y, se ejecutara el codigo que este dentro de la estructura while
18 ## 'Valor z'
19 ## z
20 ## ++z
21 ##endwhile
22 ##
23 ##3ra estructura for
24 ##como ejemplo se calcula la secuencia fibonacci
25 ##fib = ones(1, 10); #ones funcion integrada, crea un vector de 10 posiciones donde todos sus valores son 1
26 ##for i = 3:10
27 ## fib(i) = fib(i-1) + fib(i-2);
28 ##endfor
29 ##fib
30 ##
31 ##manejo de errores/excepciones
32 #try #intenta ejecutar el siguiente codigo
33 # m = [1;5;10;15]
```

Figura 4: Video 2 P1

```
34 ##catch #codigo a ejecutar si el codigo anterior no se pudo o tuvo algun error al ejecutar
35 ## 'No se pudo ejecutar, se continua con la ejecucion'
36 ##end_try_catch
37 ##
38 ##operaciones basicas con matrices
39 ##M = [1,2,5;4,5,8;9,5,0]
40 ##N = [0,1,3;8,10,7;7,2,0]
41 ##M * N
42 ##M - N
43 ##M * N
44 ##cross(M,N) # M X N, solo se puede hacer entre matrices de mas de 2x2
45 ##dot(M,N)
46 ##M' #matriz traspuesta
```

Figura 5: Video 2 P2

* e-mail: 3058096360301@ingenieria.usac.edu.gt

Video 3

```

Video1.m V2.m video3.m function.m video4.m
1 #funciones
2 x = hipotenusa($sintaxis para pedir 1 valor
3 [x,b] = hipotenusa(1,2) $sintaxis para pedir 2 valores
4 #grafica
5 x = [-3:0.1:1]; # forma de declara de donde a donde se evalua la funciones
6 x = linspace(-3,1,50); # 2da forma
7 plot(x, function(x), 'color','red') $color cambia el color en el que se dibuja la grafica
8 # 'LineStyle','-' cambia el estilo en que se dibuja la grafica
9 title('Titulo')
10 ylabel('eje y')
11 xlabel('eje x')
12 stem(x, function(x)) #ploteo discreto
13
14
15 #2 funciones en la misma grafica
16 x = [0:0.1:4*pi]
17 y1 = sin(x);
18 y2 = cos(x);
19
20 hold on;
21 p1 = plot(x,y1);
22 p2 = plot(x,y2);
23 set(p1,'color','red','LineWidth',2);
24 set(p2,'color','red','LineWidth',1);
25 ylabel('eje y');
26 xlabel('eje x');
27 title('Seno y coseno');
28 legend('seno','coseno');
29 hold off; #hold on y hold off indica que todo lo que esta dentro es de una misma grafica

```

Figura 6: Video 3 P1

```

Video1.m V2.m v3.m hipotenusa.m
1 ## Copyright (C) 2025 willy
2
3 ## This program is free software: you can redistribute it and/or modify
4 ## it under the terms of the GNU General Public License as published by
5 ## the Free Software Foundation, either version 3 of the License, or
6 ## (at your option) any later version.
7
8 ## This program is distributed in the hope that it will be useful,
9 ## but WITHOUT ANY WARRANTY; without even the implied warranty of
10 ## MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
11 ## GNU General Public License for more details.
12
13 ## You should have received a copy of the GNU General Public License
14 ## along with this program. If not, see <https://www.gnu.org/licenses/>.
15
16 ## -- texinfo --
17 ## @defun {} (@var(retval) =) hipotenusa (@var(input1), @var(input2))
18 ##
19 ## @seealso{}
20 ## @end defun
21
22 ## Author: willy <willy@WILLYSIC>
23 ## Created: 2025-04-10
24
25 function retval = hipotenusa (input1, input2)
26
27 endfunction

```

Figura 9: Video 3 Funcion creada

Video 4

```

Video1.m V2.m video3.m function.m video4.m
1 #conexion de octave con postgres
2 pkg load database $ocupa el paquete de base de datos
3 conn = pg_connect ($redshift ('dbname','postgres','host','localhost','port','5432','user','postgres','password','shwilly')); # conexion a la ba
4 w = pg_exec_params (conn, 'SELECT FROM test'); # estructura, hace la conexion y luego ejecuta la instruccion de postgres que esta en las comil
5 w_data; #muestra los datos que hay en la estructura w
6 w = pg_exec_params (conn, 'INSERT INTO test (num,texto,num2) VALUES (22, 'texto de octa',365)'); #para insertar datos

```

Figura 10: Video 4

```

Video1.m V2.m video3.m function.m video4.m
1 ## Copyright (C) 2025 willy
2
3 ## This program is free software: you can redistribute it and/or modify
4 ## it under the terms of the GNU General Public License as published by
5 ## the Free Software Foundation, either version 3 of the License, or
6 ## (at your option) any later version.
7
8 ## This program is distributed in the hope that it will be useful,
9 ## but WITHOUT ANY WARRANTY; without even the implied warranty of
10 ## MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
11 ## GNU General Public License for more details.
12
13 ## You should have received a copy of the GNU General Public License
14 ## along with this program. If not, see <https://www.gnu.org/licenses/>.
15
16 ## -- texinfo --
17 ## @defun {} (@var(retval) =) funcion (@var(input1), @var(input2))
18 ##
19 ## @seealso{}
20 ## @end defun
21
22 ## Author: willy <willy@WILLYSIC>
23 ## Created: 2025-04-10
24
25 function y = f(x)
26 y = 4*x.^3 + 10* x.^2 +6
27
28 endfunction

```

Figura 7: Video 3 P2

```

Query Query History
1 create table test(
2   num INTEGER,
3   texto varchar(50), -- se usa texto porque text es una palabra reservada
4   num2 INTEGER
5 )

Data Output Messages Notifications
CREATE TABLE

Query returned successfully in 55 msec.

```

Figura 11: Video 4 Creacion de tabla

```

Query Query History
1 select * from test

Data Output Messages Notifications
num integer texto character varying (50) num2 integer

```

Figura 12: Video 4 Tabla creada

III. RESULTADOS

Video 1

```

95 62 71
ans =
1 4 9
2 5 5
5 8 0
>> edit hipotenusa.m

```

Figura 8: Video 3 Creacion de funcion

```

ans = 375
ans = 375
ans = 375
ans = 375
ans =
Columns 1 through 12:
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000    6.5000
Columns 13 through 19:
    7.0000    7.5000    8.0000    8.5000    9.0000    9.5000    10.0000
>> Video1
M =
     1     2     3
     4     5     6
     7     8     9
N =
     1     2     3
     4     5     6
     7     8     9

```

Figura 13: Resultados Video 1 P1

```

>> Video1

ans = Hola Mundo
ans = Hola Mundo
\\ |

```

Figura 14: Resultados Video 1 P2

```

string = Secuencia numerica
estructua = {} (0x0)
estructura =

    scalar structure containing the fields:

        numero = 375
        letra = A

x =

```

```

    scalar structure containing the fields:

        secuencia =

            1     2     3     4     5

        matriz =

            1     2     3     4

    string = Secuencia numerica
    estructua = {} (0x0)
    estructura =

        scalar structure containing the fields:

            numero = 375
            letra = A

```

Figura 15: Resultados Video 1 P3

```
>> Video1
```

```

z = 2
y = 3
ans = 5
ans = -1
ans = 6
ans = 0.6667
ans = 3
ans = 2
ans = 1
ans = 1
ans = 0
ans = 0
ans = 0
ans = 1
z = 1
y = 0
ans = 0
ans = 1
ans = 0
\\ |

```

Figura 16: Resultados Video 1 P4

Video 2

```
>> V2
```

```

x = 1
y = 0
z = 5
ans = y es mayor a x

```

Figura 17: Resultados Video 2 P1

```
>> V2

x = 1
y = 0
z = 5
>> |
```

Figura 18: Resultado Video 2 P2

```
>> V2

x = 1
y = 0
z = 5
fib =

    1    1    2    3    5    8   13   21   34   55
```

Figura 19: Resultado Video 2 P3

```
>> V2

x = 1
y = 0
z = 5
ans = No se pudo ejecutar, se continua con la ejecucion
```

Figura 20: Resultado Video 2 P4

```
    1    3    8
   12   15   15
   16    7    0

ans =

    1    1    2
   -4   -5    1
    2    3    0

ans =

   51   31   17
   96   70   47
   40   59   62

ans =

  -44  -40    0
   -7    1    0
    8   15   11

ans =

   95   62   71

ans =

    1    4    9
    2    5    5
    5    8    0
```

Figura 21: Resultado Video 2 P5

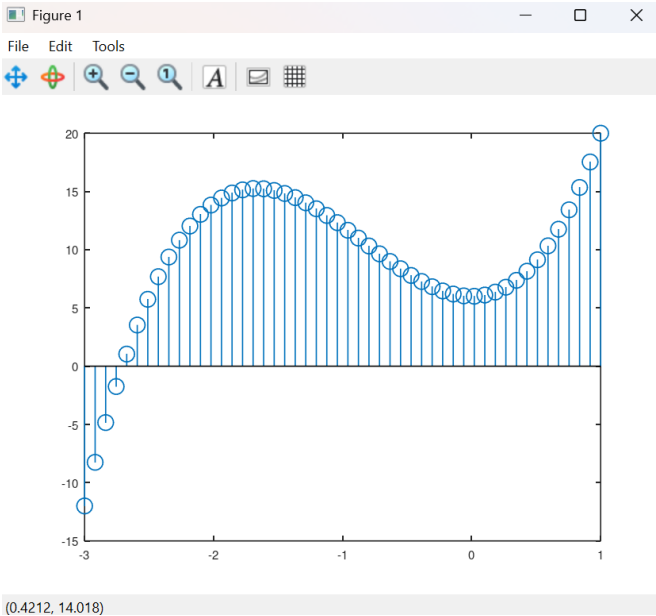


Figura 22: Resultado Video 3 P1

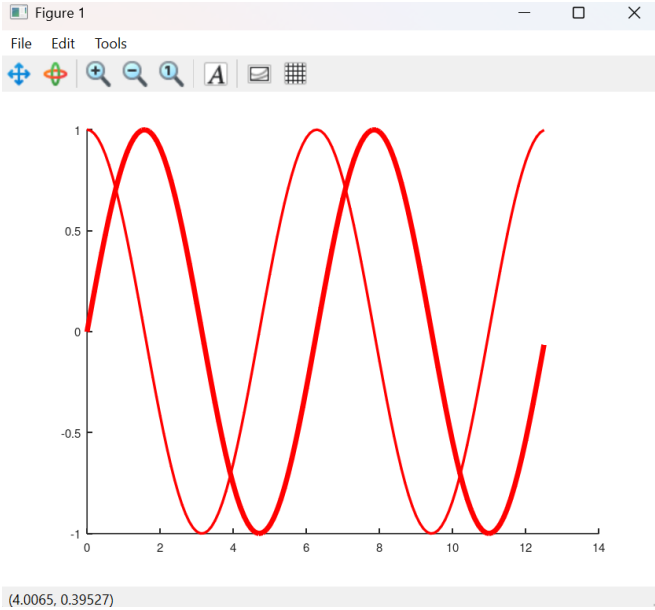


Figura 23: Resultado Video 3 P2

Video 4

Query Query History

```
1 select * from test;
```

Data Output Messages Notifications

	num integer	texto character varying (50)	num2 integer
1	12	texto de prueba	1005
2	22	Texto desde OCTA	365

Figura 24: Resultado Video 4

IV. REPOSITORIO

<https://github.com/WilliamSican/Tareas-Proyectos.git>