

Control del flujo

- Las estructuras de **control de flujo** en MATLAB son tres:

if, for, while

- La forma más sencilla de **if** es:

If expresión

sentencias

end

- Por ejemplo:

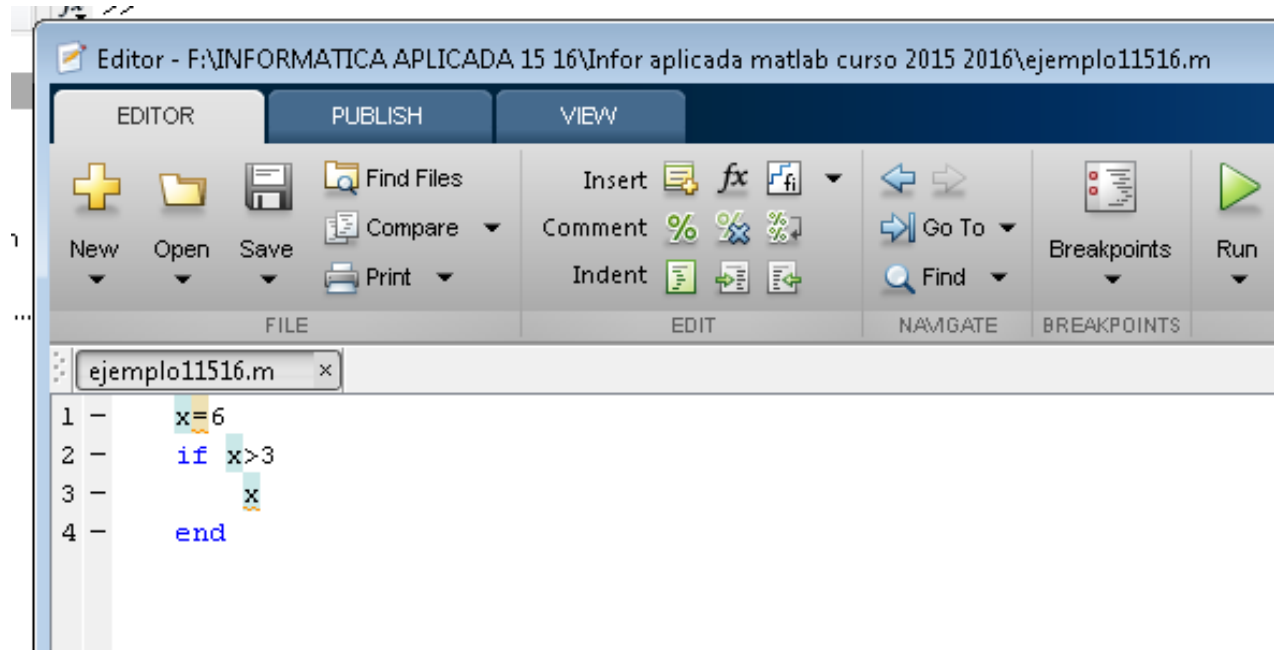
x=6

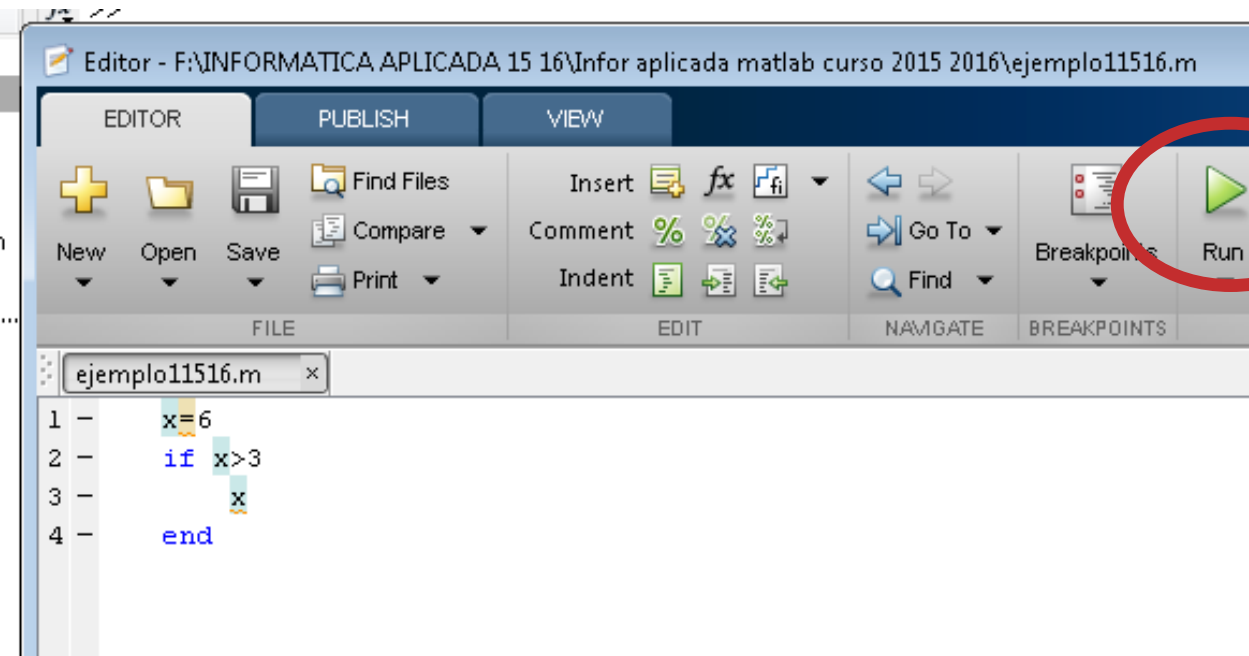
if x>3

x

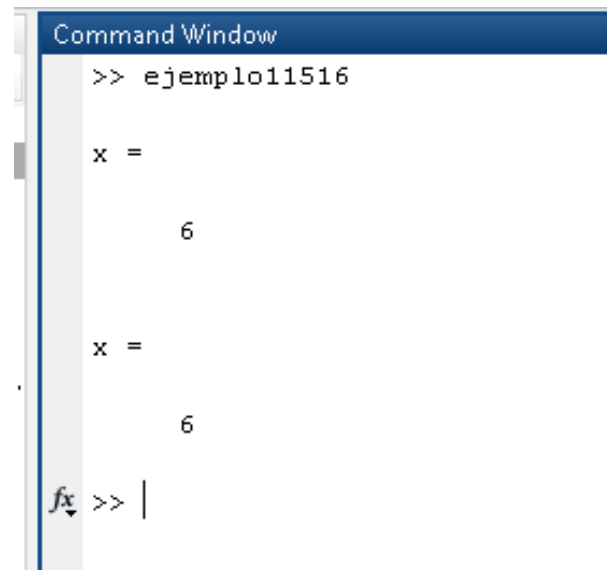
end

ABRIR EDITOR





IR A COMAND WINDOW



Editor - F:\INFORMATICA APLICADA 15 16\Infor apli

EDITOR PUBLISH

New Open Save Find Files Compare Print

ejemplo11516.m

```
1 - x=6
2 - b=1.5 * x
3 - if x>3 & b>4
4 -     x+b
5 - end
```



Command Window

```
>> ejemplo11516

x =

     6

b =

     9

ans =

    15

fx >>
```

Editor - F:\INFORMATICA APLICADA 15 16\Infor apli

EDITOR PUBLISH VIEW

New Open Save Find Files Compare Print Insert Comment Indent

ejemplo11516.m

```
1 - x=4
2 - b=1.5 * x
3 - if x>3 & b>8
4 -     x+b
5 - end
```



Command Window

```
>> ejemplo11516

x =

     4

b =

     6

fx >> |
```

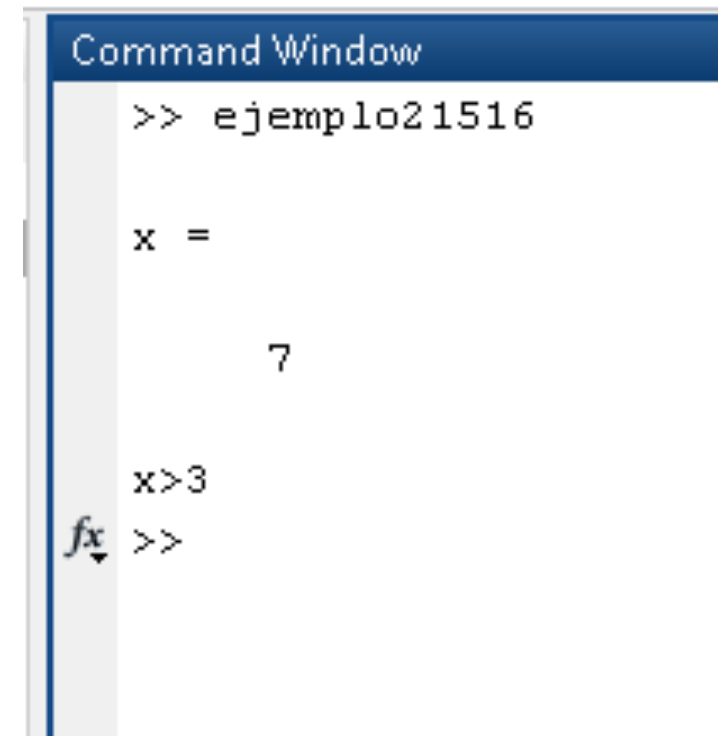
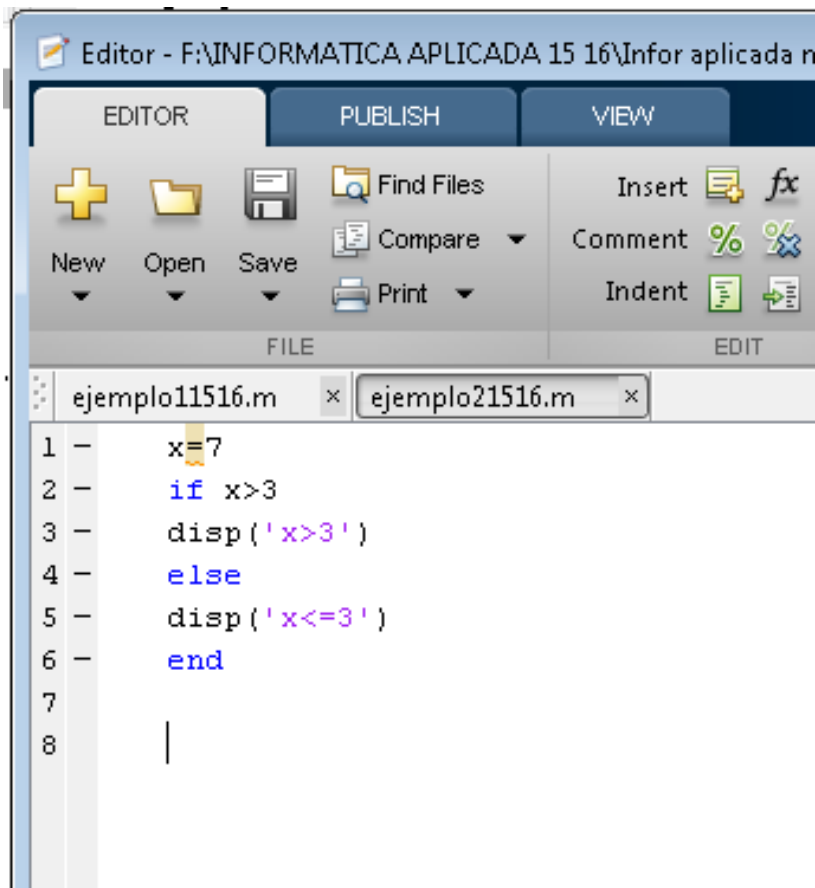


1.5 Operadores comparativos y lógicos. Control del flujo

Si se quiere que se ejecuten sentencias y si la expresión después de if es falsa, se emplea else

```
>> if x>3  
disp('x>3')  
else  
disp('x<=3')  
End
```

disp = escribe una sentencia



Control del flujo

Si se quiere comparar una condición determinada con más de una expresión en un mismo ciclo if, se emplea elseif

- Ejemplo:

```
>>x=10*(rand-0.5), y=10*(rand-0.5)
```

```
>>if (x>0)&(y>0)
```

```
disp('x e y son positivos')
```

```
elseif (x<0)|(y<0)
```

```
disp('x o y, o ambos, son negativos')
```

```
else
```

```
disp('x e y son igual a cero')
```

```
end
```

```
if expresión 1
```

```
sentencias 1
```

```
elseif expresión 2
```

```
sentencias 2
```

```
elseif expresión 3
```

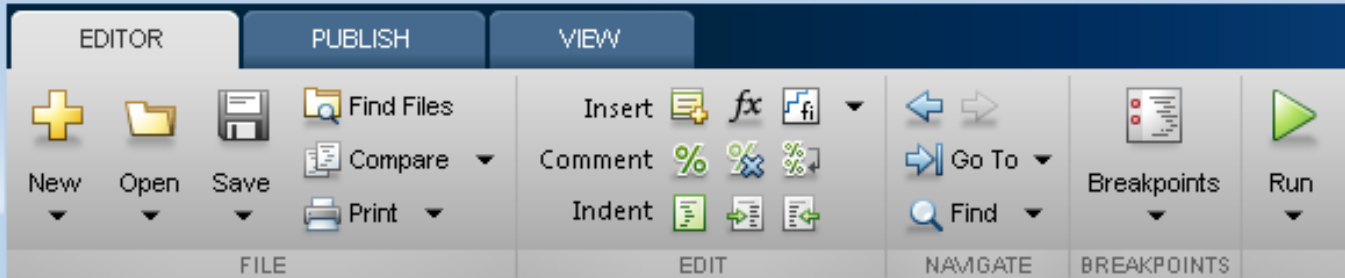
```
sentencias 3
```

```
else
```

```
sentencias 4
```

```
end
```





ejemplo11516.m x ejemplo21516.m x ejemplo31516.m x

```
1 - x=10*(rand-0.5), y=10*(rand-0.5)
2 - if (X>0) & (y>0)
3 -     disp('x e y son positivos')
4 - elseif (X<0) | (y<0)
5 -     disp('x o y, o ambos son negativos')
6 - else
7 -     disp('x e y son igual a cero')
8 - end
```

Command Window

```
>> ejemplo31516

x =

    4.6489

y =

   -3.4239

x o y, o ambos son negativos
fx >>
```



Hacer el siguiente programa:

Tres variable x, y, z en la forma $=10*(\text{rand}-0,5)$

Si $(x > y)$ y $(y > z)$, que ponga *x es el mayor*

Si $(x < y)$ y $(y < z)$, que ponga *z es el mayor*

Si $(y > x)$ y $(y > z)$, que ponga *y es el mayor*

Si y es igual a x ó y es igual a z, que ponga *igualdad*

Si no, que ponga *no hay relación*



Control del flujo FOR

La estructura 'for' se emplea para realizar ciclos en función de los valores consecutivos de una variable

for variable = expresión

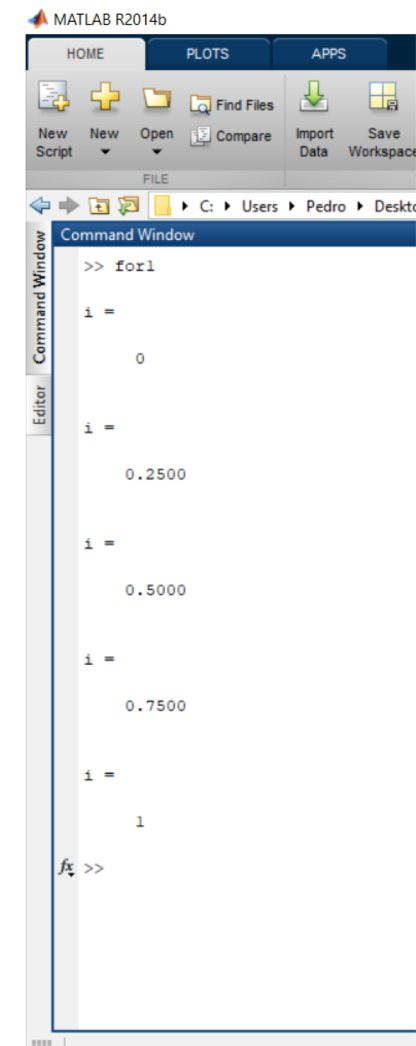
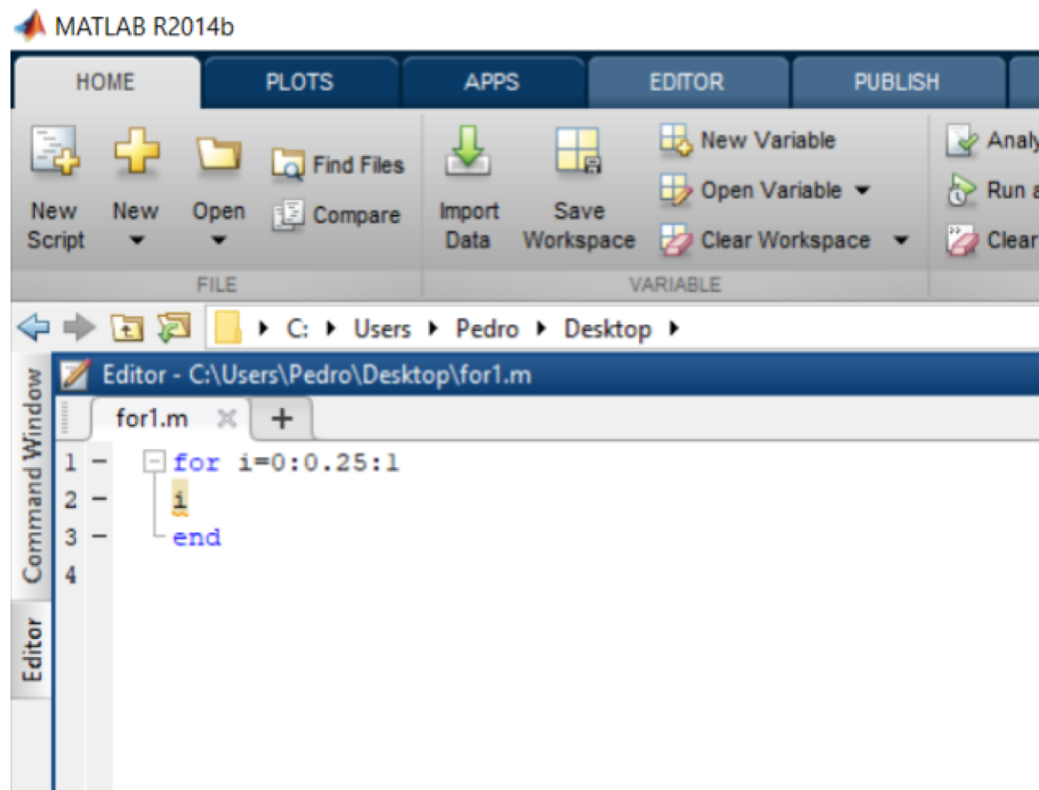
sentencias

end

for i=0:0.25:1

- Ejemplo: i

end



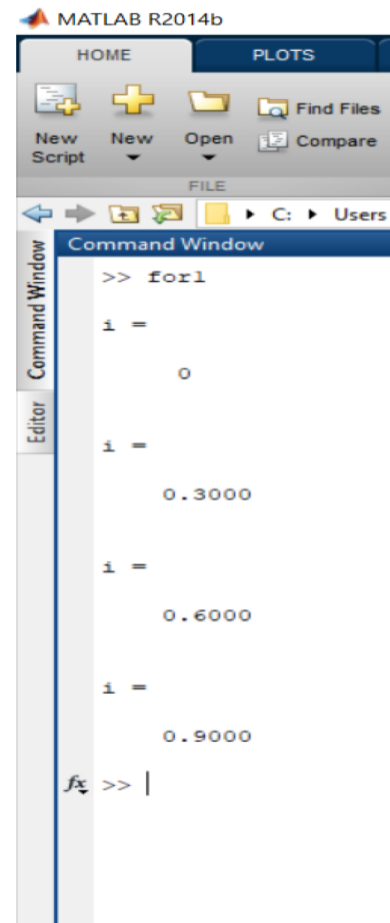
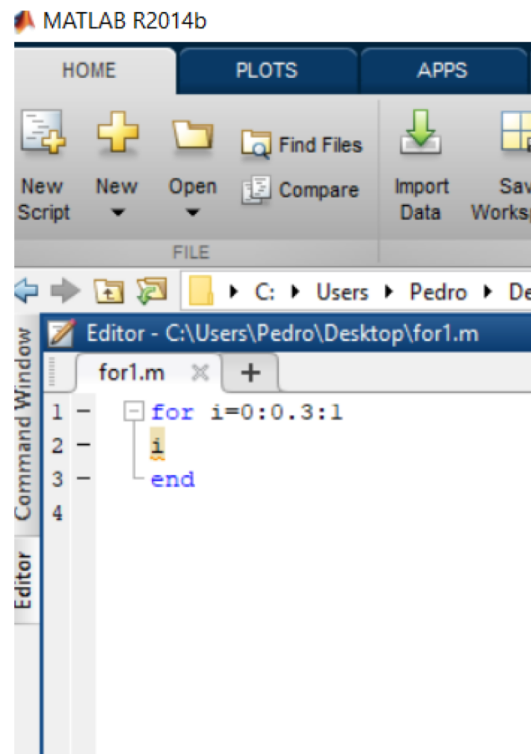
Control del flujo FOR

Si el intervalo entre el incremento de la variable no es un número entero, la variable no llega al límite final

```
for i=0:0.3:1
```

```
i
```

```
end
```



Control del flujo FOR

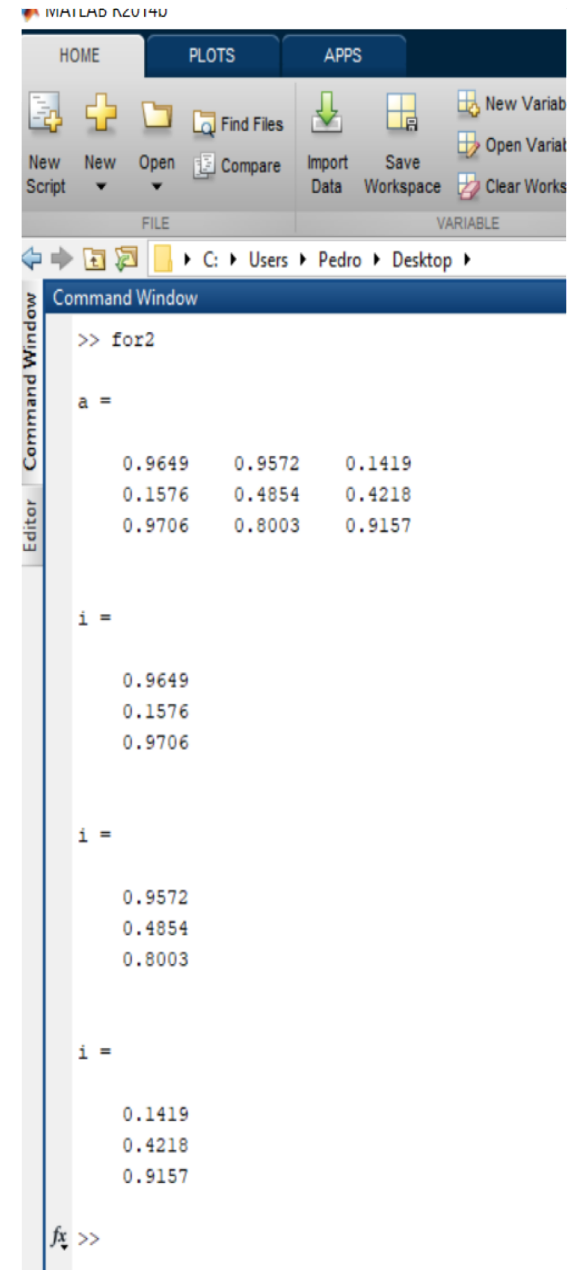
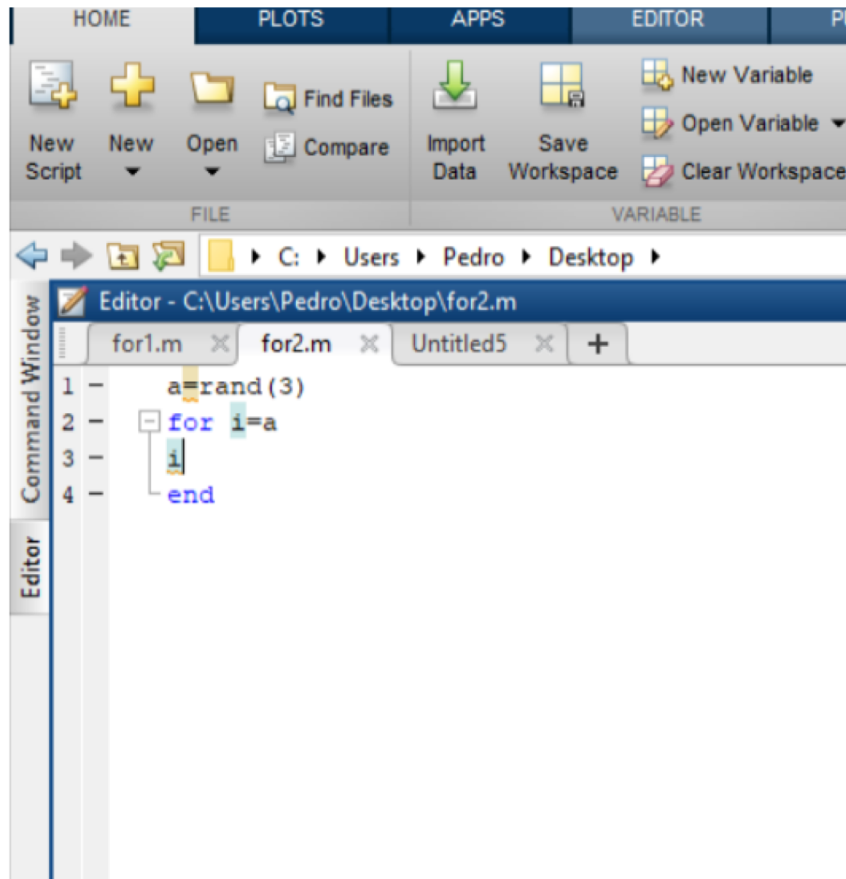
Si la expresión es una matriz, la variable se iguala a cada columna de la matriz

```
a=rand(3)
```

```
for i=a
```

```
i
```

```
end
```



1.5 Operadores comparativos y lógicos. Control del flujo

- Se pueden construir ciclos for anidados

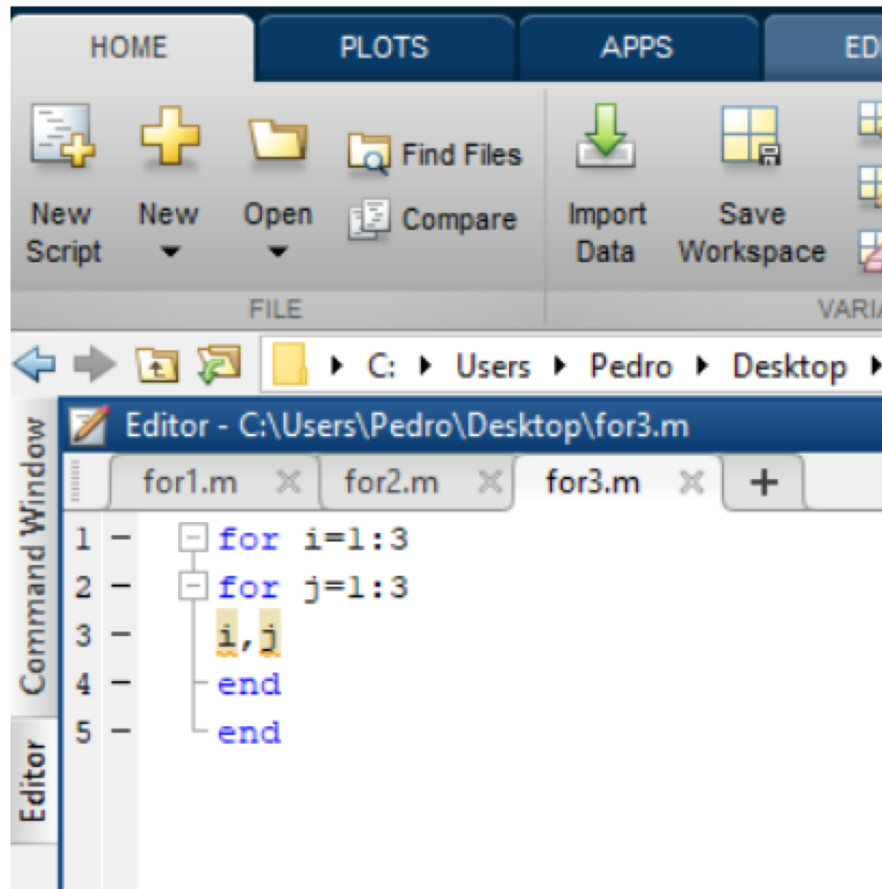
```
for i=1:3
```

```
for j=1:3
```

```
i,j
```

```
end
```

```
end
```



```
>> for3
```

```
i = 1
```

```
j = 1
```

```
i = 1
```

```
j = 2
```

```
i = 1
```

```
j = 3
```

```
i = 2
```

```
j = 1
```

```
i = 2
```

```
j = 2
```

```
i = 2
```

```
j = 3
```

```
i = 3
```

```
j = 1
```

```
i = 3
```

```
j = 2
```

```
i = 3
```

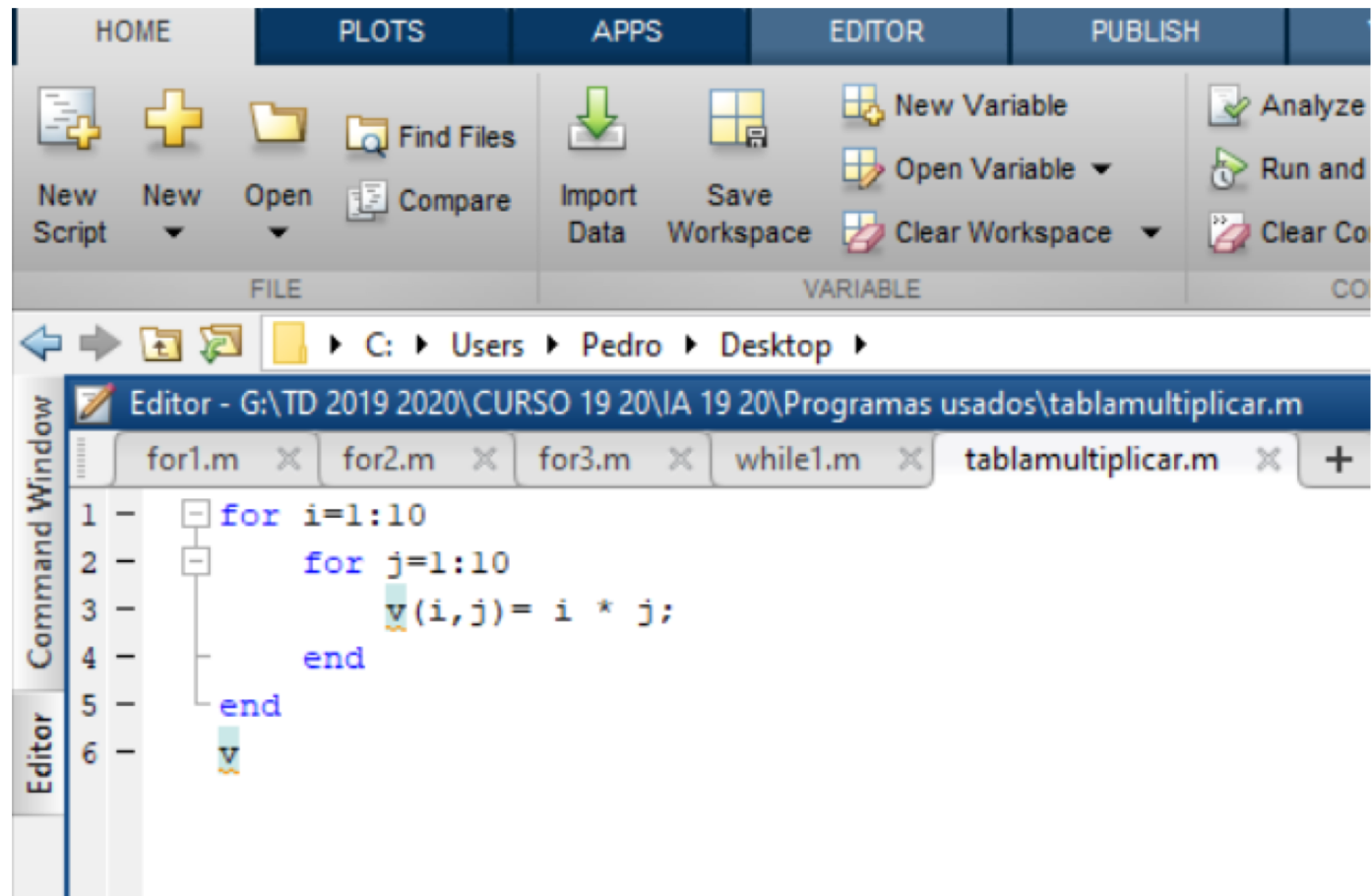
```
j = 3
```

```
>>
```



EJEMPLO for: TABLA DE MULTIPLICAR

```
for i=1:10
    for j=1:10
        v(i,j)= i * j;
    end
end
v
```



HOME PLOTS APPS

New Script New Open Find Files Compare Import Data Save Workspace New Variable Open Variable Clear Workspace Analyze Code Run and Time Clear Commands

FILE VARIABLE CODE SIMUL

← → ↗ ↘ G: TD 2019 2020 CURSO 19 20 IA 19 20 Programas usados

Command Window

```
>> tablamultiplicar
```

v =

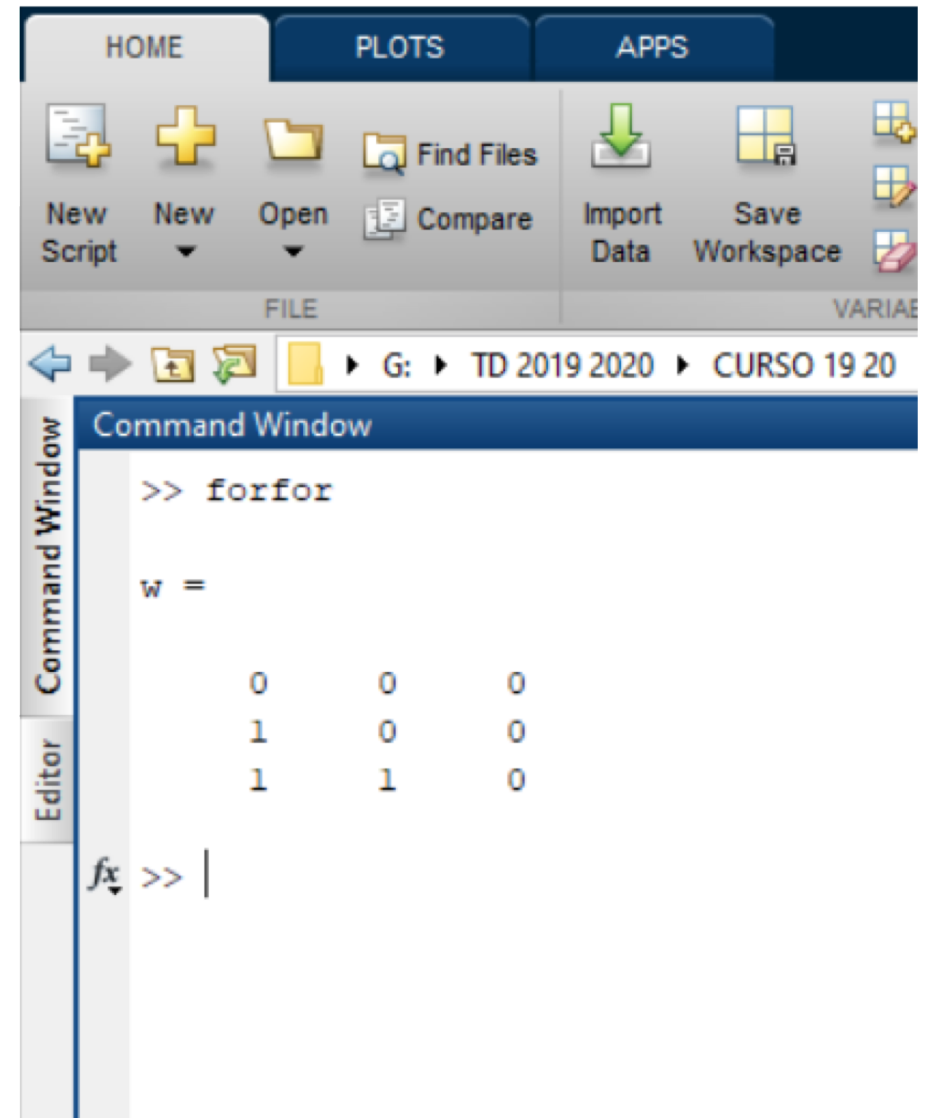
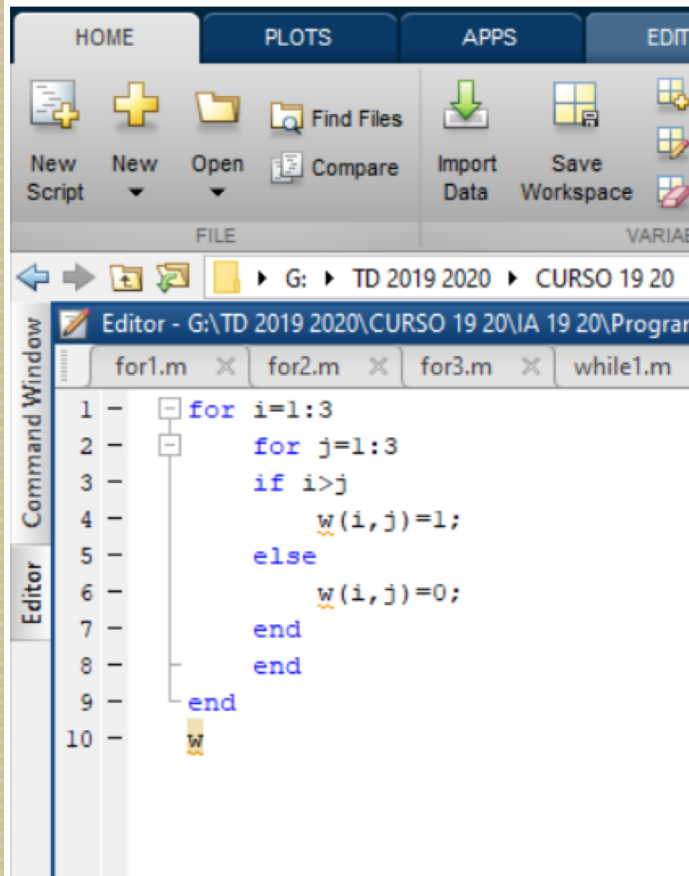
1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

fx >>



EJEMPLO for:

```
for i=1:3
    for j=1:3
        if i>j
            w(i,j)=1;
        else
            w(i,j)=0;
        end
    end
end
w
```

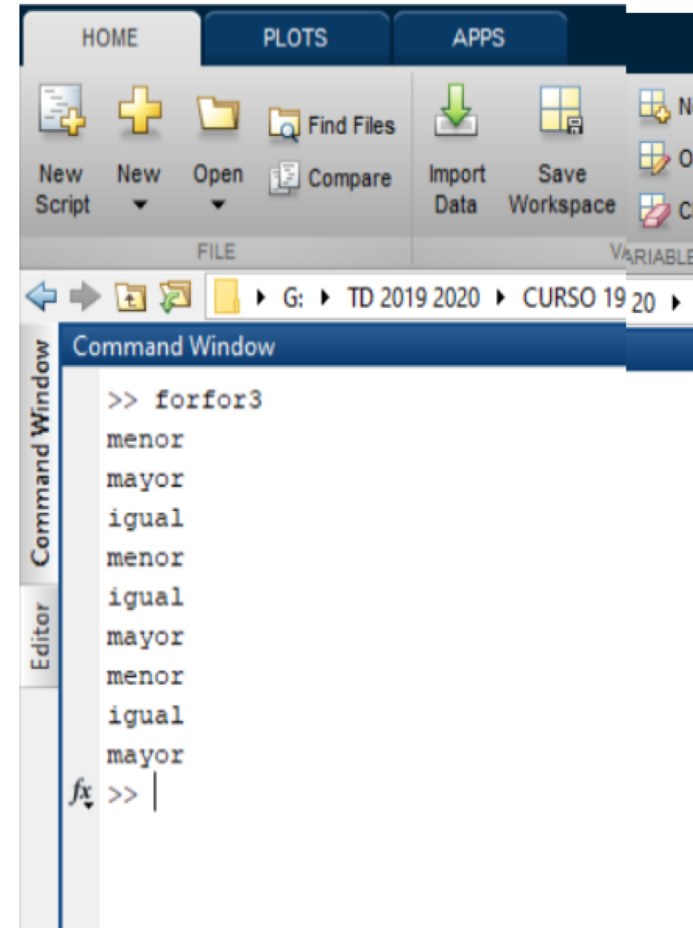
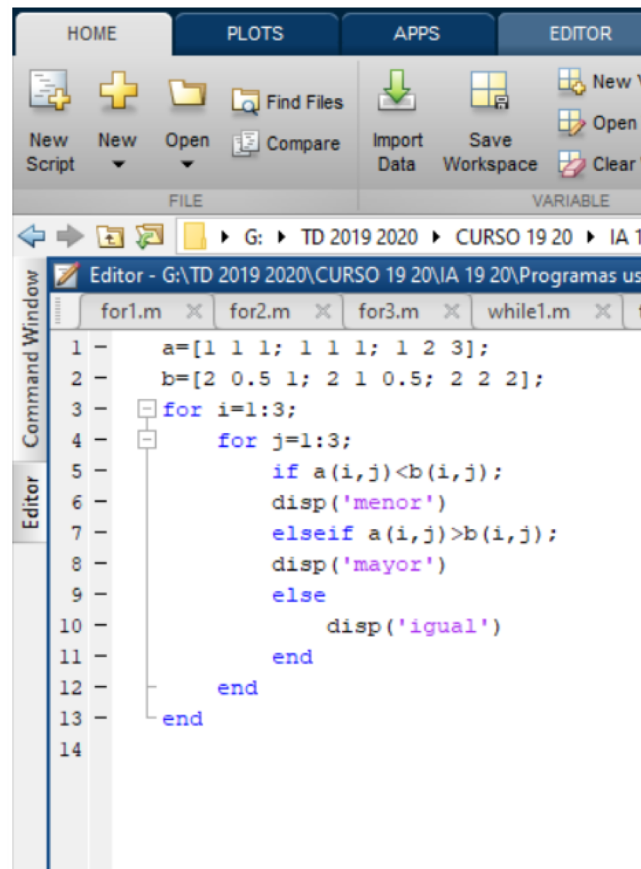


EJEMPLO for:

```
a=[1 1 1; 1 1 1; 1 2 3];
b=[2 0.5 1; 2 1 0.5; 2 2 2];
for i=1:3;
    for j=1:3;
        if a(i,j)<b(i,j);
            disp('menor')
        elseif a(i,j)>b(i,j);
            disp('mayor')
        else
            disp('igual')
        end
    end
end
end
```

$a =$	1	1	1
	1	1	1
	1	2	3

$b =$	2	0,5	1
	2	1	0,5
	2	2	2



1.5 Operadores comparativos y lógicos. Control del flujo

- El ciclo while tiene la sintaxis

while expresión

sentencias

end

x=1,

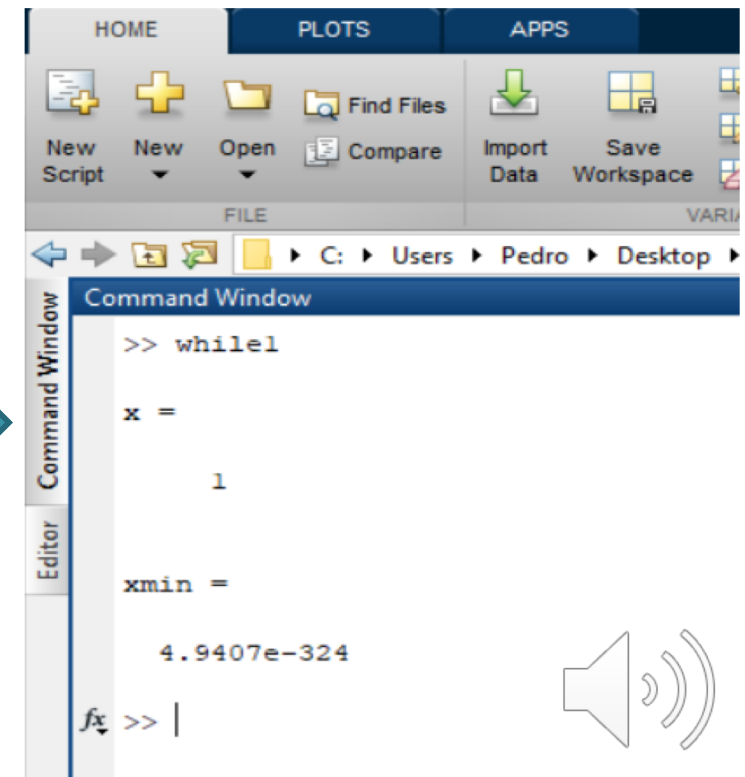
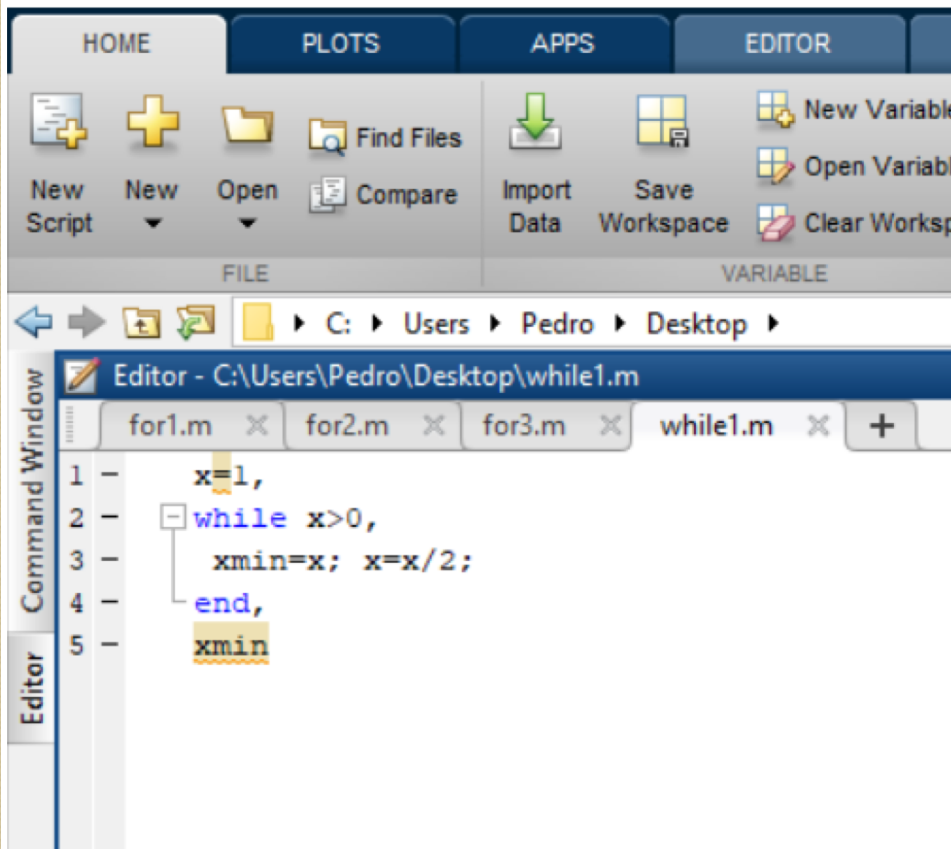
while x>0,

xmin=x;

x=x/2;

end,

xmin



Determinar cuánto tiempo se tardará en acumular al menos 10.000€ en una cuenta bancaria si se deposita 500€ inicialmente y 500€ al final de cada año, si la cuenta produce un interés anual del 5%.

```
cantidad = 500; k = 0;  
while cantidad < 10000  
    k = k+1;  
    cantidad = (cantidad + 500)*1.05 ;  
end  
cantidad  
k
```

