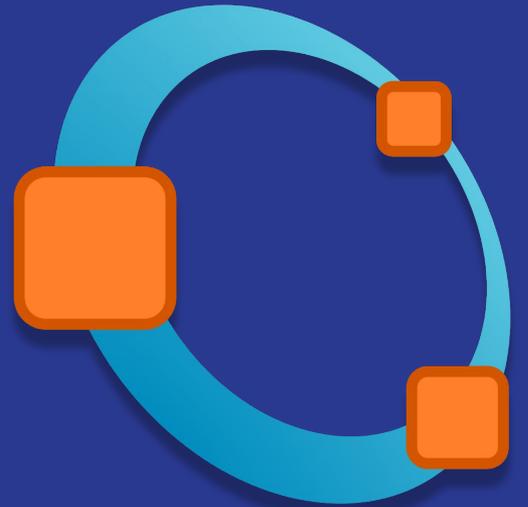


Introducción a Octave

- Instalación
- Interfaz
- Ingreso de Comandos
- Operaciones Básicas



Instalación

GNU Octave es un lenguaje de programación interpretado especializado en cálculos numéricos. Es libre, gratuito y puede descargarse desde:

<http://www.gnu.org/software/octave/download.html>

- Tutorial de instalación en windows: <https://youtu.be/s5h4-CrkHzc>
- Versión online: <https://octave-online.net>



Otros Softwares Similares (con lenguajes diferentes)



Lenguaje muy similar a al de Octave

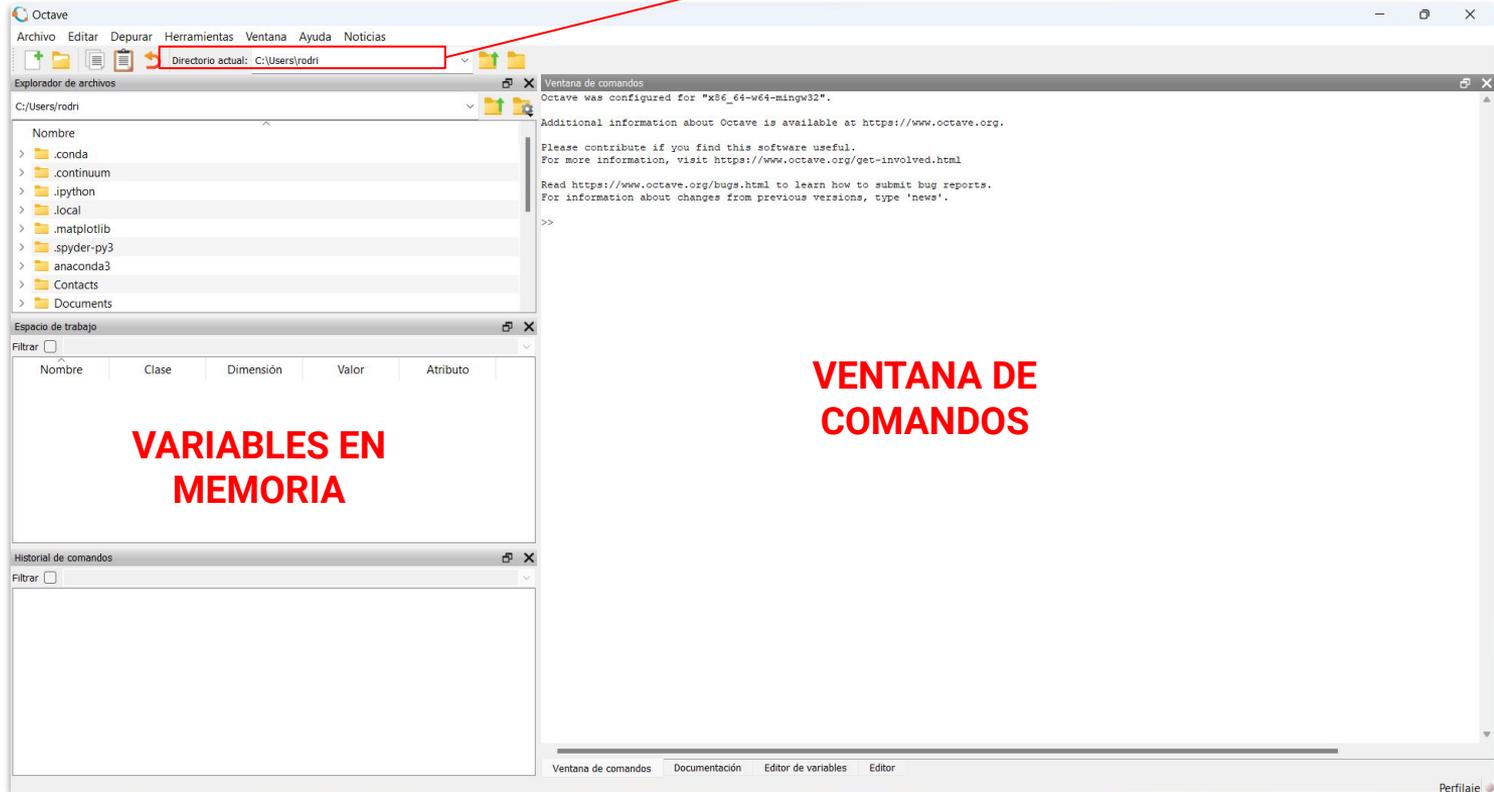


Lenguaje totalmente diferente



Interfaz

**DIRECTORIO
ACTUAL**



**VARIABLES EN
MEMORIA**

**VENTANA DE
COMANDOS**

Ingreso de Comandos

La forma más simple de interactuar con Octave es la ventana de comandos.

Permite ingresar instrucciones que la computadora ejecutará, estas instrucciones deben respetar el lenguaje que entiende Octave.

Ejemplos simples:

```
>> 2+3  
ans = 5  
>> ans + 3  
ans = 8
```

```
>> a = 5  
a = 5  
>> b = 3  
b = 3  
>> c = a + b  
c = 8
```

```
>> a = 3;  
>> b = 5;  
>> c = a + b  
c = 8
```

**; Silencia el
resultado de
la ejecución
del comando**

Ingreso de Comandos

Las instrucciones se ingresan de forma secuencial.

La instrucción más simple que permite almacenamiento de datos es la asignación de variables. Ej:

```
>> unNumero = 10;
```

El número 10 se guarda en la variable `unNumero`.

Las variables no pueden comenzar con caracteres numéricos, no pueden tener signos de acentuación, y no pueden ser *palabras reservadas*.

Formas correctas

```
>> a2 = 10;  
>> presion = 101325;  
>> TemperaturaMedia = 20;  
>> humedad_Rel = 80;  
>> c = sqrt(9);
```

Formas incorrectas

```
>> 2a = 10;  
>> presión = 101325;  
>> if = 27;  
>> sqrt = 34;
```

Tipos de datos

Existen diferentes tipos de datos con los que Octave puede trabajar:

- **Escalares**
 - `a1 = 10`
- **Vectores**
 - `vectorFila = [1 2 3 4 5 6]`
 - `vectorColumna = [1; 2; 3; 4; 5; 6]`
- **Matrices**
 - `matriz2x3 = [1 2 3; 4 5 6]`
 - `matrizB = [1 2 3; 4 5 6; 7 8 9]`
- **Texto**
 - `titulo = "Presión sobre superficie"`
 - `titulo2 = 'Temperatura media'`

Ejemplos con algunas operaciones

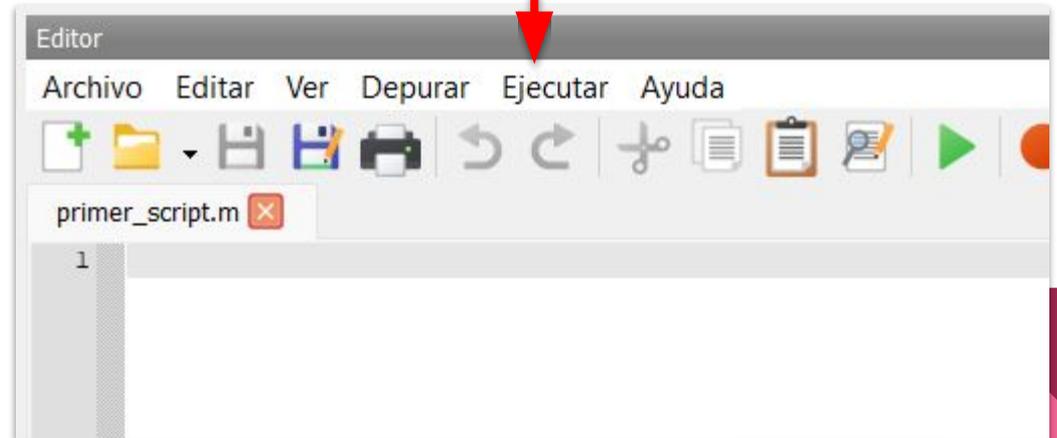
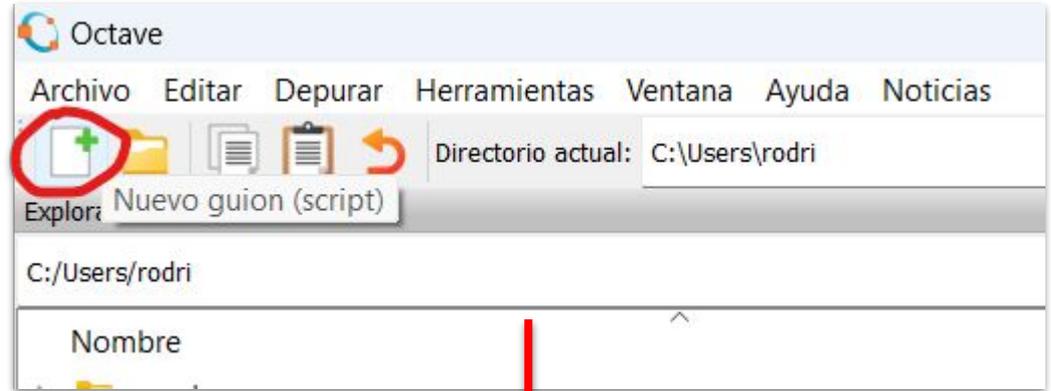
```
>> TempACarr06hs = [23 22 21 20];  
>> TemACarrColumna = TempACarr06hs';  
>> TempACarr12hs = [25 20 22 23]  
  
>> TempACarr06hs12hs = [TempACarr06hs;  
TempACarr12hs]
```

**' Matriz
transpuesta**

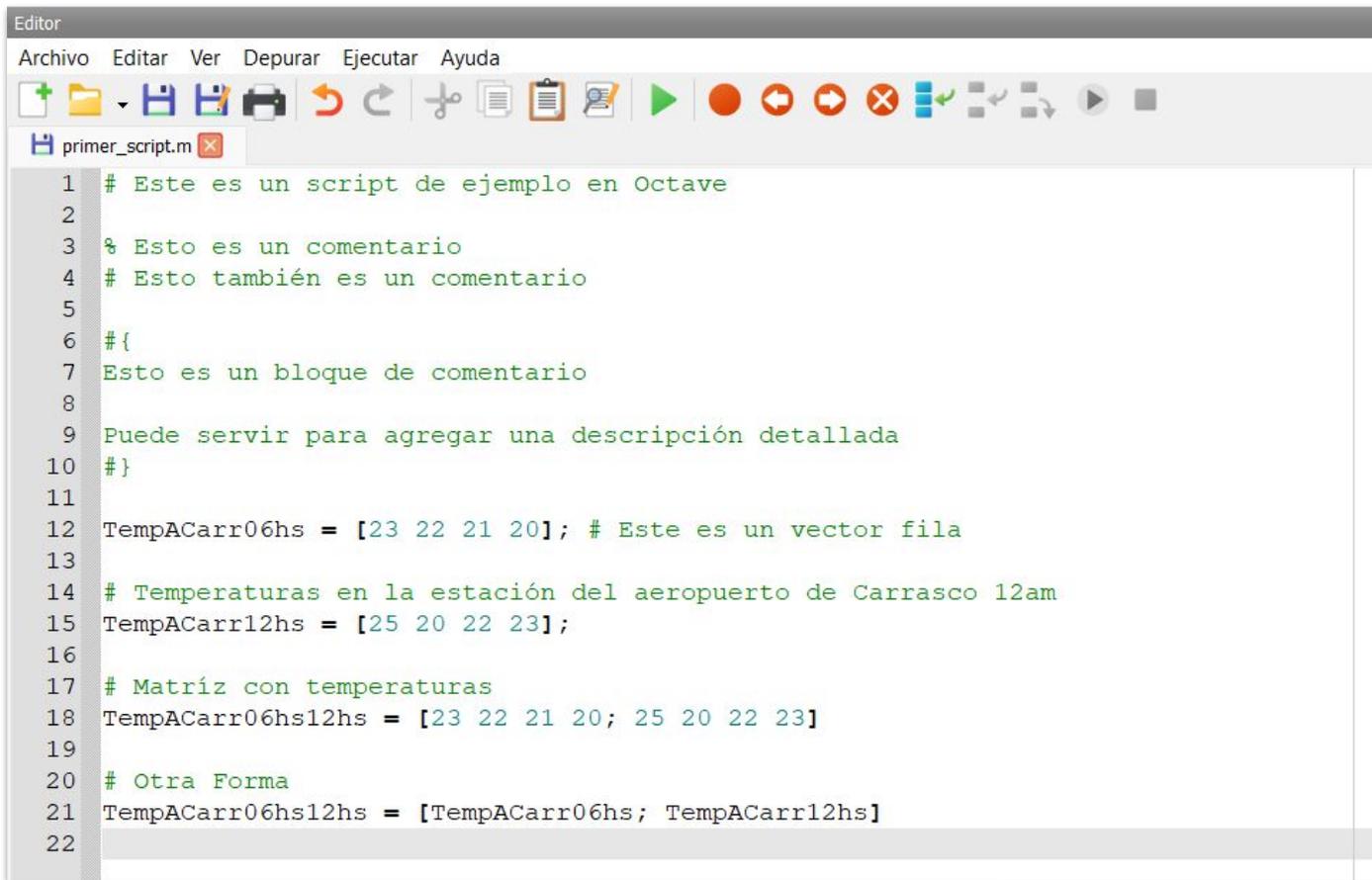
Scripts

Cuando se necesitan realizar operaciones un poco más complejas es conveniente utilizar *scripts*.

Los *scripts* permiten guardar las instrucciones de forma ordenada en un archivo.



Scripts



```
Editor
Archivo  Editar  Ver  Depurar  Ejecutar  Ayuda
[Icons]
primer_script.m
1  # Este es un script de ejemplo en Octave
2
3  % Esto es un comentario
4  # Esto también es un comentario
5
6  #{
7  Esto es un bloque de comentario
8
9  Puede servir para agregar una descripción detallada
10 #}
11
12 TempACarr06hs = [23 22 21 20]; # Este es un vector fila
13
14 # Temperaturas en la estación del aeropuerto de Carrasco 12am
15 TempACarr12hs = [25 20 22 23];
16
17 # Matriz con temperaturas
18 TempACarr06hs12hs = [23 22 21 20; 25 20 22 23]
19
20 # Otra Forma
21 TempACarr06hs12hs = [TempACarr06hs; TempACarr12hs]
22
```

Guardar matriz, comando **save**

```
>> TempACarr06hs = [23 22 21 20];  
  
>> TempACarr12hs = [25 20 22 23]  
  
>> TempACarr06hs12hs = [TempACarr06hs;  
TempACarr12hs]  
  
>> %Guardar la matriz en un archivo llamado  
"matriz.mat"  
  
>> save("matriz.mat","TempACarr06hs12hs ");
```

Resumen de funciones importantes:

```
>> v = [x1:n:x2] %Devuelve un vector v que va desde x1 a x2 con un paso de n
```

```
>> v = linspace(x1,x2,n) %Devuelve un vector v de n elementos que va desde x1 a x2
```

```
>> max(v) %Devuelve el valor máximo en el vector v
```

```
>> min(v) %Devuelve el valor mínimo en el vector v
```

```
>> mean(v) %Devuelve el valor promedio en el vector v
```

```
>> sum(v) %Devuelve la suma de los elementos en el vector v
```

```
>> size(v) %Devuelve el tamaño de la matriz v
```

Para obtener más información de un comando se puede utilizar la función help en la ventana de comandos

```
>> help sum
```