

Diplomado En Programación Básica

Universidad Autónoma de Chiapas
Centro Mesoamericano de Física Teórica

Michael Steven Paucar Rojas

MATHEMATICA



WOLFRAM

1. Introducción

El presente cuaderno constituye un recurso de apoyo para el aprendizaje de Mathematica orientado a la programación y al uso de sus principales funciones en contextos académicos y prácticos. El contenido se organiza de manera progresiva iniciando con operaciones básicas sobre listas, expresiones matemáticas y representaciones gráficas para avanzar hacia temas más complejos como manejo de entidades, conversiones de unidades, generación de visualizaciones interactivas y aplicaciones en análisis de datos.

El enfoque seguido combina teoría con ejemplos prácticos que buscan ilustrar no solo la sintaxis del lenguaje sino también la lógica detrás de cada comando. Se ha procurado mantener una estructura clara donde cada sección incluye subtítulos, descripciones y comentarios en el código para facilitar la comprensión. Esto permite que el material pueda ser utilizado tanto por estudiantes en formación como por interesados en explorar las capacidades del software en distintos escenarios.

Cabe señalar que el documento reúne apuntes propios sistematizados a partir del estudio y la práctica personal. Estos apuntes no reemplazan la documentación oficial de Mathematica pero sí constituyen un complemento útil para guiar el aprendizaje y servir como referencia en la resolución de ejercicios y proyectos futuros.

2. Tabla de contenidos

1. Introducción

2. Tabla de contenidos

3. Clase 1 — Introducción a Wolfram Mathematica

3.1. Captura y análisis de imagen

4. Clase 2 — Comandos básicos, listas y entidades

4.1. Comandos del sistema

4.2. Comandos interactivos

4.3. Entidades: países y banderas

4.4. Exploración planetaria

4.5. Conversiones de unidades y monedas

4.6. Listas: creación y operaciones básicas

4.7. Funciones para secuencias y combinación de listas

4.8. Manipulación avanzada de listas

4.9. Funciones adicionales sobre listas

5. Clase 3 — Gráficos, colores y funciones trigonométricas

5.1. Gráficas estadísticas (barras y pastel)

5.2. Selección y manipulación de datos para visualización

5.3. Colores y estilos gráficos (paletas y transformaciones)

5.4. Funciones matemáticas básicas y plots elementales

6. Clase 4 — Funciones Trascendentes

6.1. Expansión de expresiones trigonométricas

6.2. Números complejos

6.3. Logaritmos

6.4. Exponenciales

6.5. Series

6.6. Límites

6.7. Funciones

6.8. Derivadas

6.9. Integrales

6.10. Notación de Lagrange

6.11. Integración Numérica

6.12. Tablas

6.13. Gráfica de Tablas

7. Clase 5 — Visualización Matemática Interactiva

7.1. Gráficas Bidimensionales (2D)

7.2. Gráficas Tridimensionales (3D)

7.3. Manipuladores Interactivos

8. Clase 6 — Álgebra Simbólica y Series Numéricas

8.1. Solución de ecuaciones

8.2. Manipulación algebraica

8.3. Series Numéricas

9. Clase 7 — Variable Compleja

9.1. Números Complejos

9.2. Conversión de la forma Polar a Rectangular

9.3. Conversión de la forma Rectangular a Polar

9.4. Gráficas de Números Complejos

10. Clase 8 — Álgebra Lineal

10.1. Definición y creación de matrices

10.2. Operaciones básicas con matrices

10.3. Acceso a elementos

10.4. Operaciones avanzadas con matrices y vectores

10.5. Programación básica en Mathematica

11. Clase 9 — Álgebra Lineal

11.1. Operadores condicionales

11.2. Condicional if

12. Clase 10 — Bucle for

12.1. Operadores de incremento y decremento

12.2. Bucle for

12.3. Armar listas

13. Clase 11 — Control de flujo y depuración

13.1. Which

13.2. Echo

13.3. Switch

13.4. While

14. Tareas

14.1. Tarea 1 — Cálculos Numéricos y Funciones en Mathematic

14.2. Tarea 2 — Formato de Notebook

14.3. Tarea 3 — Aplicaciones de Funciones Trascendentes

14.4. Tarea 4 — Esferas 3D

14.5. Tarea 5 — Repaso general en Mathematica

14.6. Tarea 6 — Solución de ecuaciones

14.7. Tarea 7 — Variable Compleja

14.8. Tarea 8 — Reto Matrices

14.9. Tarea 9 — Aplicación del condicional if

14.10. Tarea 10 — Aplicación del bucle for

14.11. Tarea 11 — Toma de decisiones, ciclos y depuración

15. Apéndice

15.1. Comandos comunes

Tarea 11 – Toma de decisiones, ciclos y depuración

📅 2025/11/03

1. Pide al usuario un número y clasificalo como par o impar.

Usando Which: más directo y legible para condiciones lógicas

```
In[*]:= n = Input["Ingresa un número: "];
          |entra

Which[IntegerQ[n] && EvenQ[n], "Par",
      |cuál   |¿entero?   |¿par?
      IntegerQ[n] && OddQ[n], "Impar", True, "Valor no aceptado"]
      |¿entero?   |¿impar?   |verdadero
```

Out[*]=
Par

Usando Switch: se puede usar, pero es más forzado; útil si ya tienes un valor booleano o casos concretos

```
n = Input["Ingresa un número: "];
    |entra

Switch[EvenQ[n], True, "Par", False, "Impar", _, "Valor no aceptado"]
|evalúa... |¿par?   |verdadero |falso
```

Out[*]=
Impar

2. Solicita una edad y clasificala: niño, adolescente, adulto, adulto mayor.

Más directo y legible para condiciones lógicas

```
n = Input["Ingresa tu edad: "];
    |entra

Which[n ≥ 0 && n ≤ 12, "niño", n > 12 && n < 18, "adolescente",
      |cuál
      n ≥ 18 && n < 65, "adulto", n ≥ 65, "adulto mayor", True, "Valor no aceptado"]
                                          |verdadero
```

Out[*]=
adulto

3. Clasifica una temperatura como congelante, fría, agradable o calurosa.

*# Desplegamos opciones para seleccionar
Imprimimos opción seleccionada
Usamos el which para comparar rangos*

```

In[*]:= Print["1. Centígrado °C\n", "2. Fahrenheit °F\n", "3. Kelvin K"]
|describe

(*Opción seleccionar la temperatura*)
op = Input["Seleccione la Unidad de Temperatura: "];
|entra

Echo[op, "Seleccionó opción: "];
|eco

(*Centígrados*)
Which[op == 1, x = Input["Ingrese la temperatura en °C: "];
|cuál |entra
Which[x ≤ 0, Echo[ToString[x] <> " °C es Congelante"],
|cuál |eco |convierte a cadena de caracteres
x > 0 && x ≤ 15, Echo[ToString[x] <> " °C es Fría"],
|eco |convierte a cadena de caracteres
x > 15 && x ≤ 25, Echo[ToString[x] <> " °C es Agradable"], x > 25,
|eco |convierte a cadena de caracteres
Echo[ToString[x] <> " °C es Calurosa"], True, "Valor no aceptado"],
|eco |convierte a cadena de caracteres |verdadero

(*Fahrenheit*)
op == 2, y = Input["Ingrese la temperatura en °F: "];
|entra
Which[y ≤ 32, Echo[ToString[y] <> " °F es Congelante"],
|cuál |eco |convierte a cadena de caracteres
y > 32 && y ≤ 59, Echo[ToString[y] <> " °F es Fría"],
|eco |convierte a cadena de caracteres
y > 59 && y ≤ 77, Echo[ToString[y] <> " °F es Agradable"], y > 77,
|eco |convierte a cadena de caracteres
Echo[ToString[y] <> " °F es Calurosa"], True, "Valor no aceptado"],
|eco |convierte a cadena de caracteres |verdadero

(*Kelvin*)
op == 3, z = Input["Ingrese la temperatura en K: "];
|entra
Which[z ≤ 273.15, Echo[ToString[z] <> " K es Congelante"],
|cuál |eco |convierte a cadena de caracteres
z > 273.15 && z ≤ 288.15, Echo[ToString[z] <> " K es Fría"],
|eco |convierte a cadena de caracteres
z > 288.15 && z ≤ 298.15, Echo[ToString[z] <> " K es Agradable"],
|eco |convierte a cadena de caracteres
z > 298.15, Echo[ToString[z] <> " K es Calurosa"], True,
|eco |convierte a cadena de caracteres |verdadero
"Valor no aceptado"], True, "Valor no aceptado"];
|verdadero

```

1. Centígrado °C
2. Fahrenheit °F
3. Kelvin K

» Selección opción: 2

» 12.78 °F es Congelante

■ Ejercicios con Switch

4. Pide un número (1–3) y muestra un color correspondiente (Rojo, Azul, Verde).

```
# Desplegamos opciones para el usuario
# Mostramos la opción que seleccione
# Comparamos el valor ingresado con las opciones
# Imprime el color seleccionado
```

```
In[*]:= Print["1. Rojo\n", "2. Azul\n", "3. Verde"];
         |escribe

n = Input["Ingresa una opción: "];
         |entra
Echo[n, "Seleccionó opción: "];
         |eco
Switch[n, 1, RGBColor[1, 0, 0],
         |evalúa depend... |color RGB
        2, RGBColor[0, 0, 1],
         |color RGB
        3, RGBColor[0, 1, 0], _, "Opción no aceptada"]
         |color RGB
```

1. Rojo
2. Azul
3. Verde

» Selección opción: 2

Out[*]=



5. Pide un número (1–7) y muestra el día de la semana.

```
# Desplegamos opciones para el usuario
# Mostramos la opción que seleccione
# Comparamos el valor ingresado con las opciones
```

```
In[*]:= Print["DÍAS DE LA SEMANA"];
         |escribe

Print["1. Lunes\n", "2. Martes\n", "3. Miércoles\n",
         |escribe
        "4. Jueves\n", "5. Viernes\n", "6. Sábado\n", "7. Domingo\n"];

n = Input["Ingresa una opción: "];
         |entra
Echo[n, "Día seleccionado: "];
         |eco

Switch[n, 1, "Lunes", 2, "Martes", 3, "Miércoles", 4, "Jueves",
         |evalúa dependiendo de
        5, "Viernes", 6, "Sábado", 7, "Domingo", _, "Opción no aceptada"]
```


DÍAS DE LA SEMANA

1. Lunes
2. Martes
3. Miércoles
4. Jueves
5. Viernes
6. Sábado
7. Domingo

» Día seleccionado: 4

Out[]:=

Jueves

6. Solicita dos números y una opción (1-4) para realizar suma, resta, multiplicación o división.

```
# Pedimos la usuario ingrese dos valores
# Desplegamos opciones para el usuario
# Mostramos la opción que seleccione
# Comparamos el valor ingresado con las opciones
```

```
In[ ]:= n = Input["Ingrese un número: "];
        |entra
m = Input["Ingrese un número: "];
    |entra

Print["CALCULADORA"];
|escribe
Print["1. Suma\n", "2. Resta\n", "3. Multiplicación\n", "4. División\n"];
|escribe

op = Input["Ingresa una opción: "];
    |entra
Echo[op, "Operación seleccionada: "];
|eco

Print["Números ingresados: ", n, " y ", m]
|escribe

Switch[op, 1, Echo[n + m, "La suma es: "], 2, Echo[n - m, "La resta es: "], 3,
|evalúa dependi... |eco
    Echo[n * m, "El producto es: "], 4, If[m ≠ 0, Echo[n / m, "La división es: "],
|eco
    Echo["No se puede dividir entre cero"]], _, "Opción no aceptada"];
|eco
```

CALCULADORA

1. Suma
2. Resta
3. Multiplicación
4. División

» Operación seleccionada: 4

Números ingresados: 5 y 0

» No se puede dividir entre cero

7. Pide un número (1-3) e imprime el nombre de una figura (cuadrado, triángulo, círculo).

```
# Validamos entrada
# Graficamos las figuras con su nombre
```

```
In[ ]:= Print["FIGURAS GEOMÉTRICAS"];
         |escribe

Print["1. Cuadrado\n", "2. Triángulo\n", "3. Círculo\n"];
         |escribe

n = Input["Ingresa una opción: "];
      |entra

Echo[n, "Figura seleccionada: "];
     |eco

Switch[n, 1, Labeled[Graphics[{Blue, Rectangle[{0, 0}, {2, 2}]}], "Cuadrado", Top], 2,
         |evalúa depend... |etiquetado |gráfico |azul |rectángulo |arriba
      2, Labeled[Graphics[{Green, Polygon[{0, 0}, {2, 0}, {1, 2}]}], "Triángulo", Top],
         |etiquetado |gráfico |verde |polígono |arriba
      3, Labeled[Graphics[{Red, Disk[{1, 1}, 1]}], "Círculo", Top],
         |etiquetado |gráfico |rojo |disco |arriba
      _, Print["Opción no aceptada"]
         |escribe
```

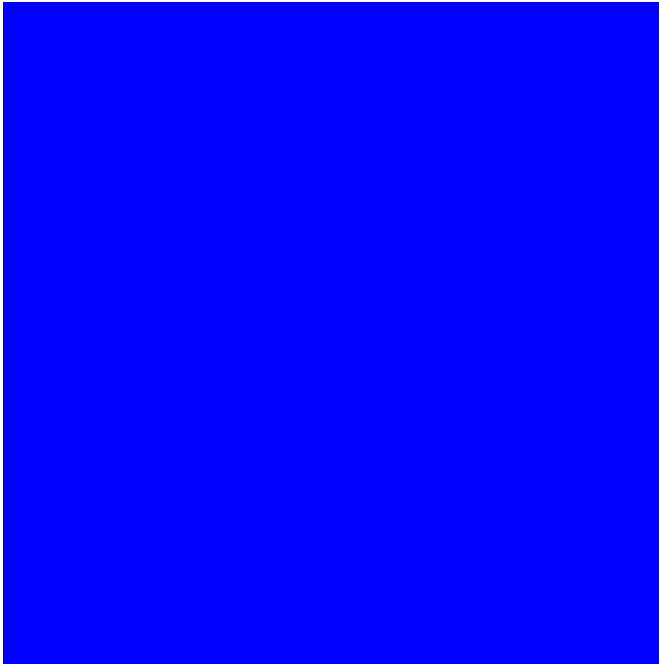
FIGURAS GEOMÉTRICAS

1. Cuadrado
2. Triángulo
3. Círculo

» Figura seleccionada: 1

Out[]=

Cuadrado



8. Cuenta del 1 al 10 mostrando cada número.

Usamos el While inicia desde 1 y terminando el bucle cuando se menor o igual a 10

```
In[ ]:= n = 1;
While [n ≤ 10, Print [n]; n++]
|_mientras_ |_escribe_
```

1
2
3
4
5
6
7
8
9
10

9. Pide un número y cuenta hacia atrás hasta llegar a 0.

Validamos la entrada del usuario sea un entero

```
# Realizamos el bucle de cuenta regresiva
```

```
In[*]:= n = Input["Ingrese un número: "];
         |entra
Which[IntegerQ[n],
      |cuál  |¿entero?
      While[n ≥ 0, Print[n];
            |mientras |escribe
            n--], True, "Valor no aceptado"];
            |verdadero
```

5

4

3

2

1

0

```
In[*]:= n = Input["Ingrese un número: "];
         |entra
Which[IntegerQ[n],
      |cuál  |¿entero?
      While[n ≥ 0, Print[n];
            |mientras |escribe
            n--], True, "Valor no aceptado"]
            |verdadero
```

```
Out[*]=
```

```
Valor no aceptado
```

10. Pide números hasta que el usuario escriba 0 y cuenta cuántos fueron positivos y negativos.

```
# Inicializamos contadores para positivos y negativos
# Validamos si no es entero, no entre al bucle
# Si ingresamos de entrada 0 da un mensaje...
# Validamos en el while entradas no validas y no cuenten en los contadores
# Termina el programa solo ingresando 0
```

```

In[*]:= auxp = 0;
auxn = 0;

n = Input["Ingrese un número: "];
      |entra

While[! IntegerQ[n], Echo["Valor no aceptado"];
      |mientras |¿entero? |eco
      n = Input["Ingrese un número: "];];
      |entra

(*Si el primer número es 0*)
If[n == 0, Echo["Ingreso 0 valores positivos y negativos"],
   |si |eco

   (*Si no es 0, iniciamos el ciclo *) While[n != 0,
      |mientras

      Which[! IntegerQ[n], Echo["Valor no aceptado"], n > 0, auxp++, n < 0, auxn++];
      |cuál |¿entero? |eco

      (*Pedimos el número y validamos*)
      n = Input["Ingrese otro número (0 para terminar): "];
      |entra

      While[! IntegerQ[n], Echo["Valor no aceptado"];
      |mientras |¿entero? |eco
      n = Input["Ingrese otro número (0 para terminar): "];];];

Echo[auxp, "Cantidad de números positivos: "];
|eco
Echo[auxn, "Cantidad de números negativos: "];];
|eco

```

- » Valor no aceptado
- » Valor no aceptado
- » Cantidad de números positivos: 2
- » Cantidad de números negativos: 3