

# **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.** Exponentiales
  - 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 Numéricicas

## 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. Clase 12 — Análisis de Datos

### 14.1. Import

### 14.2. Export

### 14.3. Do

### 14.4. Fit

### 14.5. FindFit

## 15. Tareas

### 15.1. Tarea 1 — Cálculos Numéricos y Funciones en Mathematica

### 15.2. Tarea 2 — Formato de Notebook

- 15.3.** Tarea 3 — Aplicaciones de Funciones Trascendentes
- 15.4.** Tarea 4 — Esferas 3D
- 15.5.** Tarea 5 — Repaso general en Mathematica
- 15.6.** Tarea 6 — Solución de ecuaciones
- 15.7.** Tarea 7 — Variable Compleja
- 15.8.** Tarea 8 — Reto Matrices
- 15.9.** Tarea 9 — Aplicación del condicional if
- 15.10.** Tarea 10 — Aplicación del bucle for
- 15.11.** Tarea 11 — Toma de decisiones, ciclos y depuración
- 15.12.** Tarea12 — Análisis de Datos con Ajustes Polinómicos y Trigonométricos

## Tarea 12 – Análisis de Datos con Ajustes Polinómicos y Trigonometrías

 2025/10/22

- En la siguiente tabla se muestran las masas de los mesones pseudoescalares. La Masa es obtenida usando un modelo teórico y  $m_{SP}^{\text{exp}}$  es la masa que se obtiene experimentalmente.

- Pase esta tabla a Mathematica.

	Mass [Gev]	$m_{SP}^{\text{exp}}$	error[%]	Massa en[MeV]
ūd	0.139	0.139	□	□
ūs	0.499	0.493	□	□
c̄ū	1.855	1.864	□	□
c̄s	1.945	1.986	□	□
ūb	5.082	5.279	□	□
̄sb	5.281	5.366	□	□
̄cb	6.138	6.274	□	□
c̄c	2.952	2.983	□	□
̄bb	9.280	9.398	□	□

```
# Creamos lista de cada columna
```

```
In[1]:= (*Nombres de los mesones*)
meson = {"ūd", "us", "cū", "cs", "ūb", "b̄", "b̄", "cc", "b̄"};

(*Masa teórica en GeV*)
massGev = {0.139, 0.499, 1.855, 1.945, 5.082, 5.281, 6.138, 2.952, 9.280};

(*Masa experimental en GeV*)
massExp = {0.139, 0.493, 1.864, 1.986, 5.279, 5.366, 6.274, 2.983, 9.398};
```

- Calcule el error entre el valor obtenido con el modelo y el valor experimental.

$$\text{ErrorPorcentual} = \frac{\text{Exp}-\text{Teorico}}{\text{Exp}} * 100$$

```
# Creamos una lista vacía para agregar las listas
# Agregamos la tabla final
```

```
In[1]:= errorPorcentual = {};

Do[AppendTo[errorPorcentual, (massExp[[i]] - massGev[[i]]) / massExp[[i]] * 100],  
  |<> |añade al final  
  {i, 1, Length[massGev]}];  
  |longitud

errorPorcentual

Out[1]= {0., -1.21704, 0.482833, 2.06445, 3.73177, 1.58405, 2.16768, 1.03922, 1.25559}
```

```
In[2]:= TablaFinal = Transpose[{meson, massGev, massExp, errorPorcentual}];  
  |transposición

Out[2]= {{ūd, 0.139, 0.139, 0.}, {us, 0.499, 0.493, -1.21704}, {cū, 1.855, 1.864, 0.482833},  
  {cs, 1.945, 1.986, 2.06445}, {üb, 5.082, 5.279, 3.73177}, {b̄, 5.281, 5.366, 1.58405},  
  {b̄, 6.138, 6.274, 2.16768}, {cc, 2.952, 2.983, 1.03922}, {b̄, 9.28, 9.398, 1.25559}}
```

**■ Obtenga el valor de la masa en MeV. Una vez completada la tabla, exportela a un archivo llamado “TablaC.dat”.(Esta tabla también se envia a la plataforma)**

```
# Creamos los encabezados y una nueva lista para exportar datos
```

```
In[3]:= masaMeV = {};  
Do[AppendTo[masaMeV, massGev[[i]] * 1000], {i, 1, Length[massGev]}];  
  |<> |añade al final  
  |longitud

TablaC = Transpose[{meson, massGev, massExp, errorPorcentual, masaMeV}];  
  |transposición

encabezado = {"Mesón", "Masa [GeV]", "m_SP_exp [GeV]", "Error [%]", "Masa [MeV]"};  
TablaCCompleta = Prepend[TablaC, encabezado];  
  |añade al principio

Export["TablaC.dat", TablaCCompleta, "Table"]  
  |exporta  
  |tabla
```

```
Out[3]= TablaC.dat
```

**■ Calcule la diferencia de masas entre todos los mesones.**

```
# Calculamos la diferencia entre masas mediante indices
```

```
In[4]:= diferenciaMasas = Table[massExp[[i]] - massGev[[i]], {i, 1, Length[massGev]}];  
  |tabla  
  |longitud

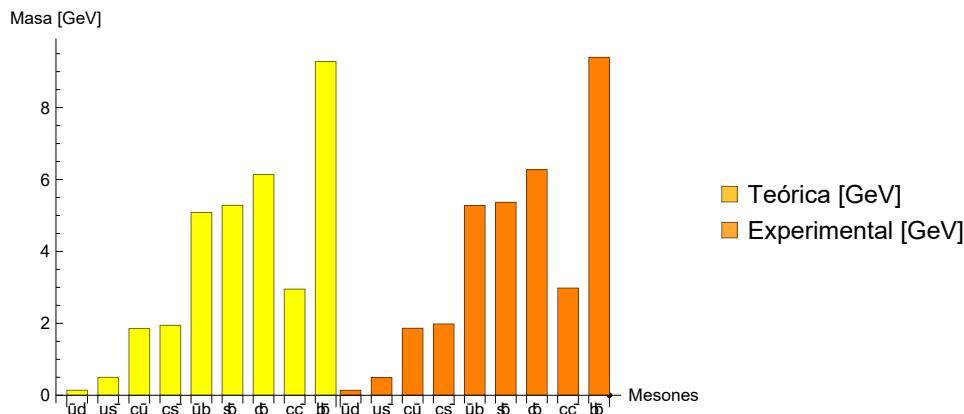
Out[4]= {0., -0.006, 0.009, 0.041, 0.197, 0.085, 0.136, 0.031, 0.118}
```

**■ Realice una gráfica donde se muestre una comparación entre las masas experimentales y las teóricas. (Sugerencia: Use el comando BarChart).**

```
# Creamos las gráficas entre las dos tablas de masas
```

```
In[8]:= BarChart[{Style[massGev, Yellow], Style[massExp, Orange]},  
|diagrama d...|estilo |amarillo |estilo |naranja  
ChartLabels → meson, BarSpacing → 0.5, AxesLabel → {"Mesones", "Masa [GeV]"},  
|etiquetas de diagrama |espaciado de barras |etiqueta de ejes  
ChartLegends → {"Teórica [GeV]", "Experimental [GeV"]}]  
|leyendas de diagrama
```

Out[8]=



## 2. Implemente esta tabla en Mathematica. Primero escribela en un editor de textos e importala a Mathematica.

0.	0.
1.	1.36395
2.	-1.1352
3.	-0.419123
4.	1.48404
5.	-0.816032
6.	-0.804859
7.	1.48591
8.	-0.431855
9.	-1.12648
10.	1.36942

```
# Importamos la tabla creada con los datos del enunciado
```

```
In[9]:= datos = Import["Table.dat", "Table"]  
|importa |tabla |tabla
```

Out[9]=

```
{ {0., 0.}, {1., 1.36395}, {2., -1.1352}, {3., -0.419123}, {4., 1.48404}, {5., -0.816032},  
{6., -0.804859}, {7., 1.48591}, {8., -0.431855}, {9., -1.12648}, {10., 1.36942} }
```

## ■ Usa el comando “fit” para encontrar la mejor ecuación que se ajuste a los datos.

```
# Ajustamos un modelo polinómico para que se ajuste a los valores
```

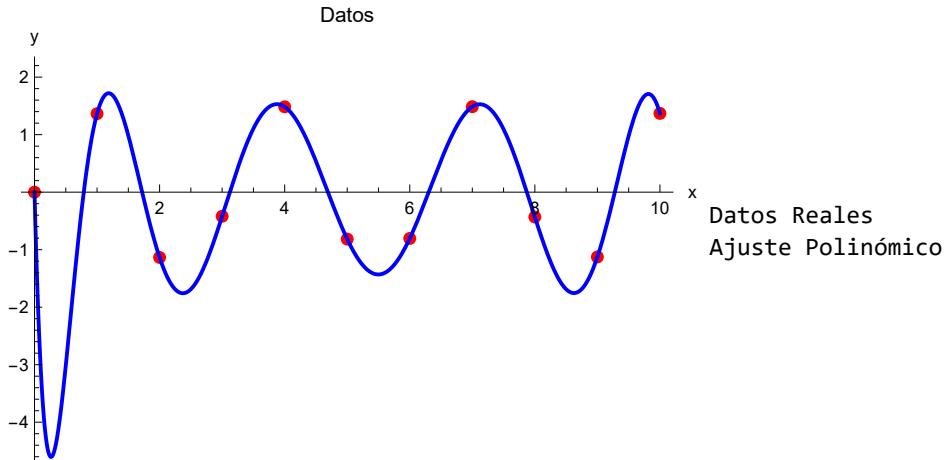
```
In[8]:= ajuste = Fit[datos, {1, x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^10}, x]
          |ajusta
```

```
Out[8]= -2.7055 × 10-10 - 41.9401 x + 124.445 x2 - 139.126 x3 + 79.7093 x4 - 26.4894 x5 +
5.39953 x6 - 0.684921 x7 + 0.0527102 x8 - 0.00225226 x9 + 0.0000409862 x10
```

■ **Gráfica la tabla junto con tu ajuste para ver que tan bueno es.**

```
In[9]:= Show[ListPlot[datos, PlotStyle -> {Red, PointSize[0.015]}, PlotMarkers -> Automatic,
          |mue...|representación de li...|estilo de repre...|rojo|tamaño de punto|marcadores de ...|automático
          PlotLabel -> "Datos", AxesLabel -> {"x", "y"}, PlotLegends -> "Datos Reales",
          |etiqueta de representación|etiqueta de ejes|leyendas de representación
          Plot[ajuste, {x, 0, 10}, PlotStyle -> {Blue, Thick}, PlotLegends -> "Ajuste Polinómico"],
          |representación gráfica|estilo de repre...|azul|grueso|leyendas de representación
          PlotRange -> {{0, 10}, {-4.3, 2}}]
          |rango de representación
```

```
Out[9]=
```



**3. Haz una tabla con los primeros 25 números pares, graficalos y haz un ajuste de la curva. Exporta la tabla a “Pares.dat”(se sube a la plataforma.)**

```
# Creamos la tabla de pares con sus indices identificadores
```

```
In[10]:= numerosPares = Table[2 * i, {i, 1, 25}];
          |tabla
tabla = Table[{i, 2 * i}, {i, 1, 25}]
          |tabla
```

```
Out[10]= {{1, 2}, {2, 4}, {3, 6}, {4, 8}, {5, 10}, {6, 12}, {7, 14}, {8, 16}, {9, 18},
{10, 20}, {11, 22}, {12, 24}, {13, 26}, {14, 28}, {15, 30}, {16, 32}, {17, 34},
{18, 36}, {19, 38}, {20, 40}, {21, 42}, {22, 44}, {23, 46}, {24, 48}, {25, 50}}
```

```
# Exportamos la tabla de datos
```

```
In[]:= Export["Pares.dat", tabla, "Table"]
          |exporta
          |tabla
```

```
Out[=]= Pares.dat
```

# Limpiamos variables y creamos un modelo lineal para mostrar en la gráfica

```
In[]:= Clear[x, a, b];
          |borra
ajuste = FindFit[tabla, a + b x, {a, b}, x]
          |ajusta una función parametrizada
```

```
Out[=]= {a → 2.84217 × 10^-15, b → 2.}
```

```
In[]:= Show[ListPlot[tabla, PlotStyle → {Red, PointSize[0.015]}, PlotMarkers → Automatic,
          |muestra...|representación de lí...|estilo de representación|rojo|tamaño de punto|marcadores de ...|automático
          PlotLabel → "Números Pares", AxesLabel → {"Índice", "Número Par"}, 
          |etiqueta de representación|etiqueta de ejes
          PlotLegends → "Datos Reales"], Plot[a + b x /. ajuste, {x, 1, 25},
          |leyendas de representación|representación gráfica
          PlotStyle → {Blue, Thick}, PlotLegends → "Ajuste Lineal"]
          |estilo de representación|azul|grueso|leyendas de representación]
```

```
Out[=]=
```

