

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. Tareas

9.1. Tarea 1 — Cálculos Numéricos y Funciones en Mathematic

9.2. Tarea 2 — Formato de Notebook

9.3. Tarea 3 — Aplicaciones de Funciones Trascendentes

9.4. Tarea 4 — Esferas 3D

9.5. Tarea 5 — Repaso general en Mathematica

10. Apéndice

10.1. Comandos comunes

Tareas

⚡ **Instrucciones:** En esta sección se agrupan las tareas asignadas.

Tarea 5 – Repaso general en Mathematica

📅 2025/10/06

1. Use Range para crear la lista {1, 2, 3, 4}

```
In[ ]:= Range[4]
| rango
```

```
Out[ ]:=
{1, 2, 3, 4}
```

2. Construya la lista de los números hasta el 100.

```
In[ ]:= Range[100]
| rango
```

```
Out[ ]:=
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100}
```

3. Use Range y Reverse para crear {4, 3, 2, 1}.

```
In[ ]:= Reverse[Range[4]]
| invierte ... | rango
```

```
Out[ ]:=
{4, 3, 2, 1}
```

4. Construya una lista de los números del 1 al 50 en orden inverso.

```
In[ ]:= Reverse[Range[50]]
| invierte ... | rango
```

```
Out[ ]:=
{50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27,
 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1}
```

5. Use Range, Reverse y Join para crear {1, 2, 3, 4, 4, 3, 2, 1}.

```
In[ ]:= Join[Range[4], Reverse[Range[4]]]
| junta | rango | invierte ... | rango
```

```
Out[ ]:=
{1, 2, 3, 4, 4, 3, 2, 1}
```

6. Use Range y RandomInteger para crear una lista de longitud aleatoria hasta 10.

```
In[ ]:= Range[RandomInteger[10]]
| rango | entero aleatorio
```

```
Out[ ]:=
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In[ ]:= Range[RandomInteger[10]]
| rango | entero aleatorio
```

```
Out[ ]:=
{1, 2, 3, 4, 5}
```

7. Encuentre una forma más simple para Reverse[Reverse[Range[10]]].

Ejemplo

```
In[ ]:= Reverse[Reverse[Range[10]]]
| invierte ... | invierte ... | rango
```

```
Out[ ]:=
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

Solución

```
In[ ]:= Range[10]
| rango
```

```
Out[ ]:=
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

8. Encuentre una forma más simple para Join[1, 2, Join[3, 4, 5]].

Ejemplo

```
In[ ]:= Join[1, 2, Join[3, 4, 5]]
| junta | junta
```

```
Out[ ]:=
Join[1, 2, 3, 4, 5]
```

Solución

```
In[ ]:= Range[5]
| rango
```

```
Out[ ]:=
{1, 2, 3, 4, 5}
```

9. Encuentre una forma más simple para Join[Range[10], Join[Range[10], Range[5]]].

Ejemplo

```
In[ ]:= Join[Range[10], Join[Range[10], Range[5]]]
```

Junta rango Junta rango rango

```
Out[ ]:= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5}
```

Solución

```
In[ ]:= Join[Range[10], Range[10], Range[5]]
```

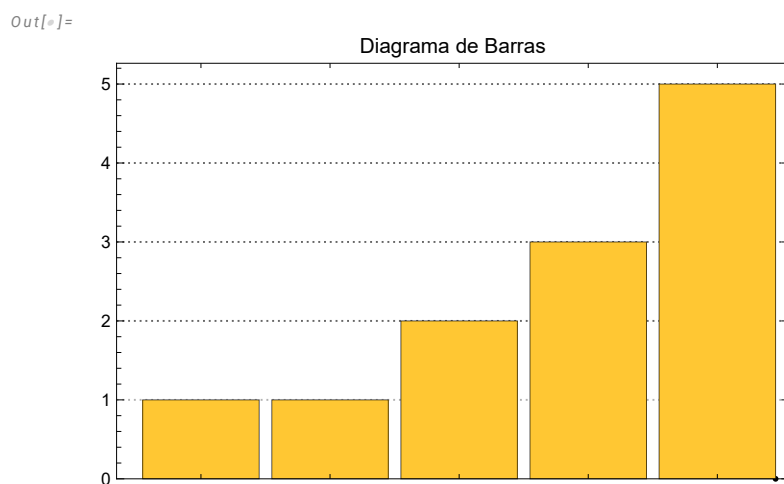
Junta rango rango rango

```
Out[ ]:= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5}
```

10. Construya un diagrama de barras con 1, 1, 2, 3, 5.

```
In[ ]:= BarChart[{1, 1, 2, 3, 5}, PlotTheme -> "Detailed", PlotLabel -> "Diagrama de Barras"]
```

diagrama de barras tema de representación etiqueta de representación

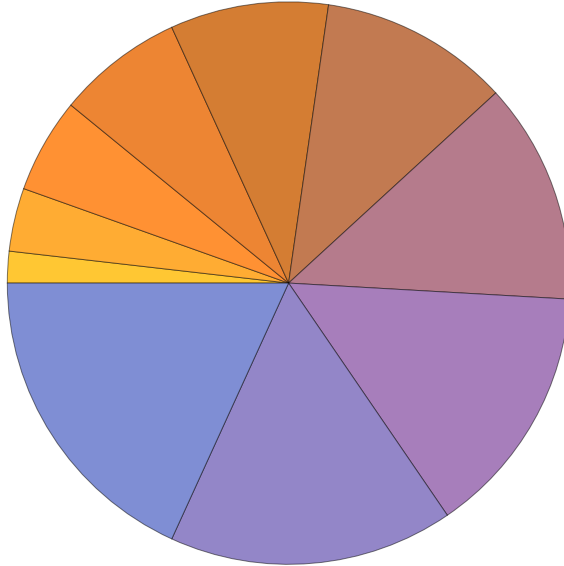


11. Produzca un diagrama circular con los números del 1 al 10.

```
In[ ]:= PieChart[Range[10], PlotTheme → "Detailed", PlotLabel → "Diagrama Circular"]
[diagrama ... [rango] [tema de representación] [etiqueta de representación]
```

Out[]:=

Diagrama Circular

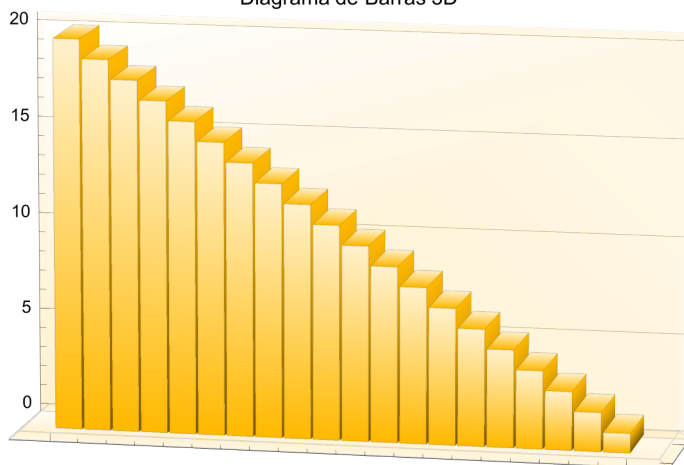


12. Forme un diagrama de barras de los números consecutivos del 20 al 1.

```
In[ ]:= BarChart3D[Reverse[Range[20]], PlotLabel → "Diagrama de Barras 3D"]
[diagrama de ... [invierte ... [rango] [etiqueta de representación] [deriv]
```

Out[]:=

Diagrama de Barras 3D



13. Muestre en una columna los números del 1 al 5.


```
In[ ]:= Column[Range[5]]
columna rango
```

```
Out[ ]:=
```

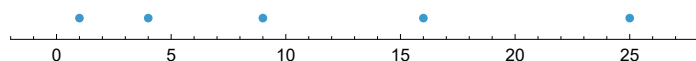
```
1
2
3
4
5
```

14. Presente los cuadrados 1, 4, 9, 16, 25 sobre una recta numérica.

Forma 1

```
In[ ]:= NumberLinePlot[Table[n^2, {n, 1, 5}]]
representación de línea numérica
```

```
Out[ ]:=
```



Forma 2

```
In[ ]:= NumberLinePlot[{1, 4, 9, 16, 25}]
representación de línea numérica
```

```
Out[ ]:=
```

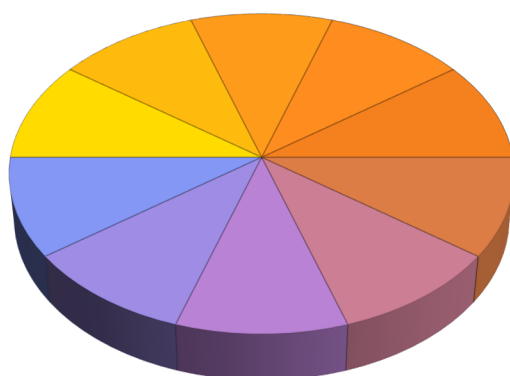


15. Forme un gráfico circular con 10 sectores idénticos, cada uno de tamaño 1.

```
In[ ]:= PieChart3D[Table[1, {n, 1, 10}], PlotLabel -> "Diagrama con Sectores Iguales"]
diagrama circular tabla etiqueta de representación
```

```
Out[ ]:=
```

Diagrama con Sectores Iguales



16. Presente una columna de los gráficos circulares con 1, 2 y 3 sectores idénticos.

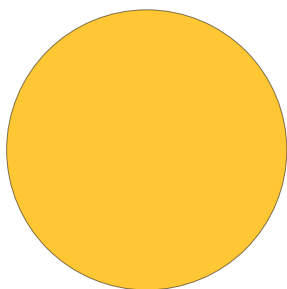
```

In[ ]:= Column[ {PieChart[Range[1], PlotLabel → "Sector 1"],
  [columna  [diagrama ... [rango  [etiqueta de representación
    PieChart[Table[1, {n, 1, 2}], PlotLabel → "Sector 2"],
    [diagrama ... [tabla  [etiqueta de representación
    PieChart[Table[1, {n, 1, 3}], PlotLabel → "Sector 3"]]}]]
  [diagrama ... [tabla  [etiqueta de representación

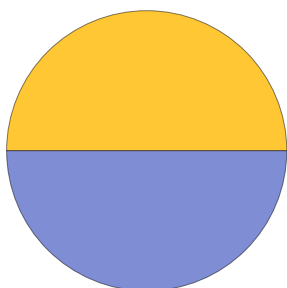
```

Out[]:=

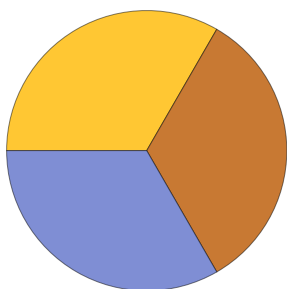
Sector 1



Sector 2



Sector 3

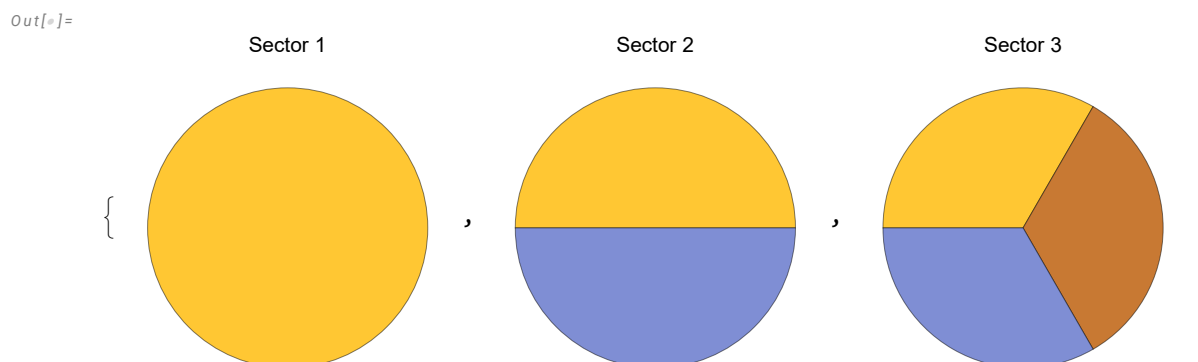


17. Forme una lista de gráficos circulares con 1, 2 y 3 sectores idénticos.

```

In[*]:= {PieChart[Range[1], PlotLabel → "Sector 1"],
          PieChart[Table[1, {n, 1, 2}], PlotLabel → "Sector 2"],
          PieChart[Table[1, {n, 1, 3}], PlotLabel → "Sector 3"]}

```

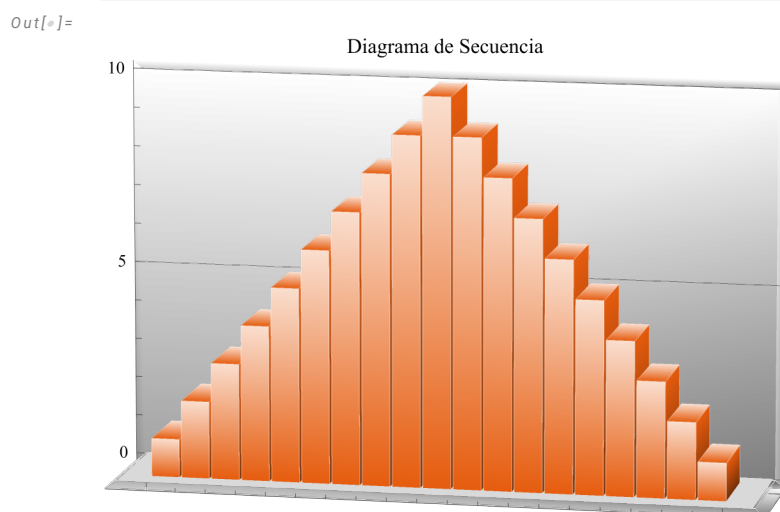


18. Presente un diagrama de barras con la secuencia 1, 2, 3, ..., 9, 10, 9, 8, 7, ..., 1.

```

In[*]:= BarChart3D[Join[Range[10], Reverse[Range[9]]],
                  PlotTheme → "Scientific", PlotLabel → "Diagrama de Secuencia"]

```



19. Cree una lista de los 10 primeros cuadrados en orden inverso.

```

In[*]:= Reverse[Table[n^2, {n, 1, 10}]]

```

Out[*]=

{100, 81, 64, 49, 36, 25, 16, 9, 4, 1}

20. Calcule el total de los 10 primeros cuadrados.

```
In[ ]:= Total[Table[n^2, {n, 1, 10}]]
      |total |tabla
```

```
Out[ ]:=
```

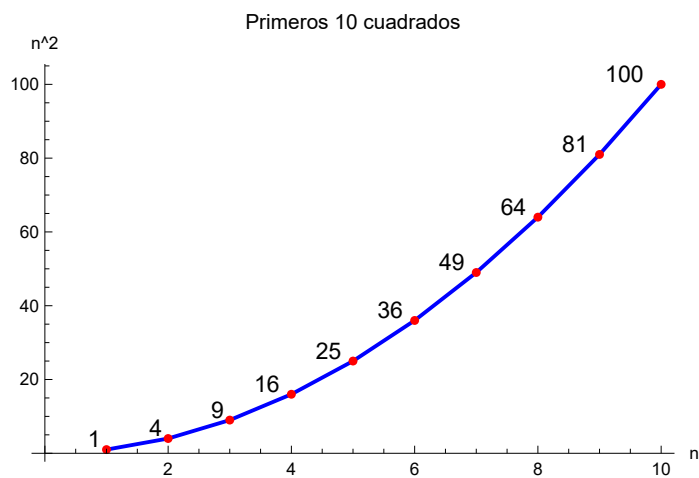
385

21. Muestre gráficamente los 10 primeros cuadrados, comenzando por el 1.

```
# Gráfica a partir de una tabla
# ListPlot[Table[n^2, {n, 1, 10}]] # Generar la lista de los cuadrados del 1 al 10
# Joined -> True # Conectar los puntos con una línea
# PlotStyle -> Blue # Estilo de la línea: azul
# PlotMarkers -> {Graphics[{Red, Disk[]}], 0.03} # Marcadores rojos en forma de círculo
# PlotLabel -> "Primeros 10 cuadrados" # Título del gráfico
# AxesLabel -> {"n", "n^2"} # Etiquetas de los ejes
# Epilog -> Table[... ] # Añadir etiquetas sobre los puntos
# Text[Style[n^2, 12, Black], {n, n^2}, {1.9, -0.8}] # Estilo y posición de cada etiqueta
# Table[... , {n, 1, 10}] # Repetir la etiqueta para cada punto n = 1 a 10
```

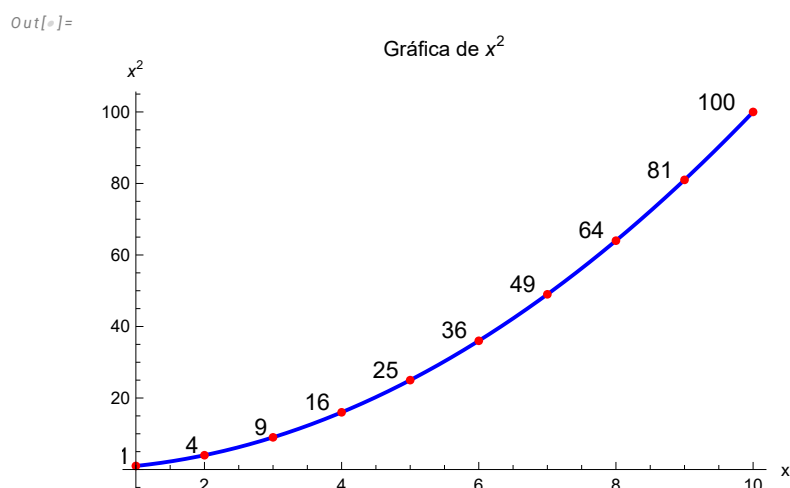
```
ListPlot[Table[n^2, {n, 1, 10}], Joined -> True,
  |represent... |tabla |unido |verdadero
  PlotStyle -> Blue, PlotMarkers -> {Graphics[{Red, Disk[]}], 0.03},
  |estilo de repr... |azul |marcadores de re... |gráfico |rojo |disco
  PlotLabel -> "Primeros 10 cuadrados", AxesLabel -> {"n", "n^2"},
  |etiqueta de representación |etiqueta de ejes
  Epilog -> Table[Text[Style[n^2, 12, Black], {n, n^2}, {1.9, -0.8}], {n, 1, 10}]
  |epílogo |tabla |texto |estilo |negro
```

```
Out[ ]:=
```



```
# Gráfica a partir de una función
# Show[...] # Combina varias gráficas en una sola visualización
# Plot[x^2, {x, 1, 10}, PlotStyle -> Blue, PlotLabel -> "Gráfica de x^2", AxesLabel -> {"x", "x^2"}] # Gráfica de la función x^2 de 1 a 10 con línea azul y etiquetas
# Graphics[{...}] # Agrega elementos gráficos personalizados sobre la gráfica principal
# {Red, PointSize[Medium], Point[Table[{n, n^2}, {n, 1, 10}]]} # Dibuja puntos rojos de tamaño medio en las coordenadas (n, n^2) para n de 1 a 10
# Black # Cambia el color a negro para los textos que siguen
# Table[Text[Style[n^2, 12], {n, n^2}, {1.9, -0.8}], {n, 1, 10}] # Añade etiquetas negras con el valor n^2 encima de cada punto, desplazadas para mejor visualización
```

```
In[ ]:= Show[Plot[x^2, {x, 1, 10}, PlotStyle -> Blue,
  PlotLabel -> "Gráfica de x^2", AxesLabel -> {"x", "x^2"}],
Graphics[{Red, PointSize[Medium], Point[Table[{n, n^2}, {n, 1, 10}]],
  Black, Table[Text[Style[n^2, 12], {n, n^2}, {1.9, -0.8}], {n, 1, 10}]}]]
```



22. Cree una lista de los 10 primeros múltiplos de 3.

```
In[ ]:= Table[3 n, {n, 1, 10}]
```

Out[]:=

```
{3, 6, 9, 12, 15, 18, 21, 24, 27, 30}
```

23. Use Sort, Join y Range para crear {1, 1, 2, 2, 3, 3, 4, 4}

```
In[ ]:= Sort[Join[Range[4], Range[4]]]
```

Out[]:=

```
{1, 1, 2, 2, 3, 3, 4, 4}
```

24. Use Range y + para formar la lista de los números del 10 al 20, inclusive.

Solución correcta usando Range y +

```
In[ ]:= Range[11] + 9
```

Out[]:=

```
{10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
```

Sin usar +

```
In[ ]:= Range[10, 20]
| rango
```

```
Out[ ]:=
{10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
```

25. Forme la lista de los 10 primeros cuadrados, usando únicamente Range y Times.

```
In[ ]:= Times[Range[10], Range[10]]
| multip· | rango | rango
```

```
Out[ ]:=
{1, 4, 9, 16, 25, 36, 49, 64, 81, 100}
```

26. Encuentre el número de dígitos en 2^{218} .

```
In[ ]:= Length[IntegerDigits[2^218]]
| longitud | dígitos de entero
```

```
Out[ ]:=
66
```

27. Encuentre el primer dígito de 2^{32} .

Salida guía

```
In[ ]:= 2^32
```

```
Out[ ]:=
4 294 967 296
```

Solución

```
In[ ]:= First[IntegerDigits[2^32]]
| primero | dígitos de entero
```

```
Out[ ]:=
4
```

28. Encuentre los 10 primeros dígitos en 2^{100}

Salida guía

```
In[ ]:= 2^100
```

```
Out[ ]:=
1 267 650 600 228 229 401 496 703 205 376
```

```
# Toma los 10 primeros números
# Take[list, {-3, -1}]
```

Sintaxis	Significado
Take[list, n]	Primeros n elementos
Take[list, -n]	Últimos n elementos
Take[list, {i, j}]	Elementos desde la posición i a j

Take[list, {i, j, s}]

Desde i a j, con paso s

```
In[ ]:= Take[IntegerDigits[2^100], 10]
```

toma dígitos de entero

```
Out[ ]:= {1, 2, 6, 7, 6, 5, 0, 6, 0, 0}
```

29. Encuentre el último dígito de 2³⁷.

Salida guía

```
In[ ]:= 2^37
```

```
Out[ ]:= 137 438 953 472
```

Solución

```
In[ ]:= Last[IntegerDigits[2^37]]
```

último dígitos de entero

```
Out[ ]:= 2
```

30. Encuentre el penúltimo dígito de 2³².

Salida guía

```
In[ ]:= 2^32
```

```
Out[ ]:= 4 294 967 296
```

Solución

Acción	Número de caracteres	Ejemplo
Extraer un elemento	Doble [[]]	IntegerDigits[2^37][[-3]]
Acceder al último 3 dígitos	[[-3, -1]]	IntegerDigits[2^37][[-1]]
Obtener toda la lista	Solo uno []	IntegerDigits[2^37]
Tomar varias posiciones	[[1, 3, 5]]	list[{{1, 3, 5}}] -> extrae esos 3

```
In[*]:= IntegerDigits[2^32][[-2]]  
[dígitos de entero]
```

```
Out[*]=
```

9