

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

7. Clase 5 — Visualización matemática interactiva

31 2025/10/01

⚡ Introducción:

En esta clase se profundiza en la visualización matemática interactiva utilizando Wolfram Mathematica, una herramienta poderosa para representar y explorar datos y funciones de manera dinámica. Se abordarán técnicas para crear gráficas bidimensionales (2D) y tridimensionales (3D), que permiten visualizar funciones matemáticas, datos estadísticos y otros objetos gráficos con gran claridad y precisión.

Además, se presentarán los manipuladores interactivos que facilitan la exploración visual a través de controles dinámicos, como deslizadores y menús desplegables. El uso de estos manipuladores permite modificar parámetros en tiempo real y observar cómo cambian las gráficas, potenciando el aprendizaje y la comprensión intuitiva de conceptos matemáticos y estadísticos.

Durante la clase, se trabajará con comandos clave como *Graphics*, *Graphics3D* y *Manipulate*, aplicándolos a diferentes tipos de gráficos, como plots, gráficos de barras, gráficos circulares, entre otros.

📋 Objetivos de la clase:

- Comprender y aplicar las técnicas básicas para la creación de gráficas bidimensionales en Mathematica.
- Construir y manipular gráficos tridimensionales para una mejor representación espacial de funciones y datos.
- Utilizar el comando *Manipulate* para crear visualizaciones interactivas que permitan la modificación dinámica de parámetros.
- Explorar diferentes tipos de gráficos: plots, gráficos de barras, gráficos circulares, etc.
- Desarrollar habilidades para combinar gráficos y controles interactivos, facilitando la interpretación visual y el análisis de información matemática.

7.1 Gráficas Bidimensionales (2D)

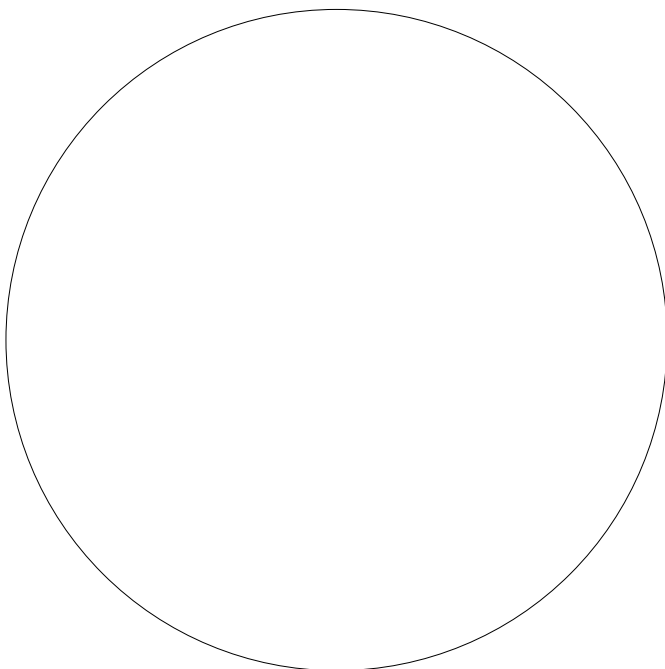
✎ **Explicación:** Las gráficas bidimensionales son representaciones visuales de funciones o datos en dos dimensiones, generalmente sobre un plano XY. En Mathematica, se pueden crear fácilmente utilizando funciones como Plot, ListPlot o Graphics. Estas gráficas permiten analizar comportamientos, tendencias y relaciones entre variables de manera clara y sencilla. Se pueden personalizar con estilos, colores y etiquetas para mejorar su comprensión.

Dibuja un círculo con centro en (0, 0) y radio 1 por defecto

```
In[*]:= Graphics[{Circle[]}]
```

gráfico círculo

Out[*]=

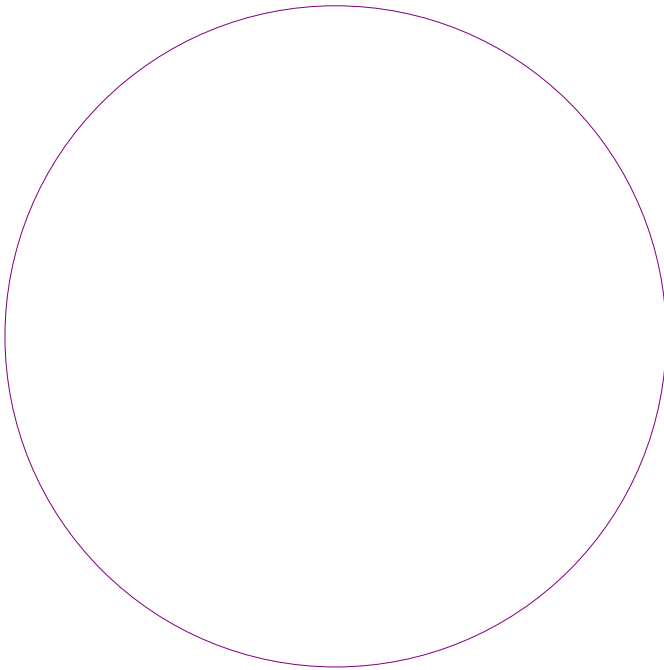


Dibuja un círculo en (0, 0) con radio 1 y lo colorea de púrpura

```
In[*]:= Graphics[Style[Circle[], Purple]]
```

gráfico estilo círculo púrpura

Out[*]=

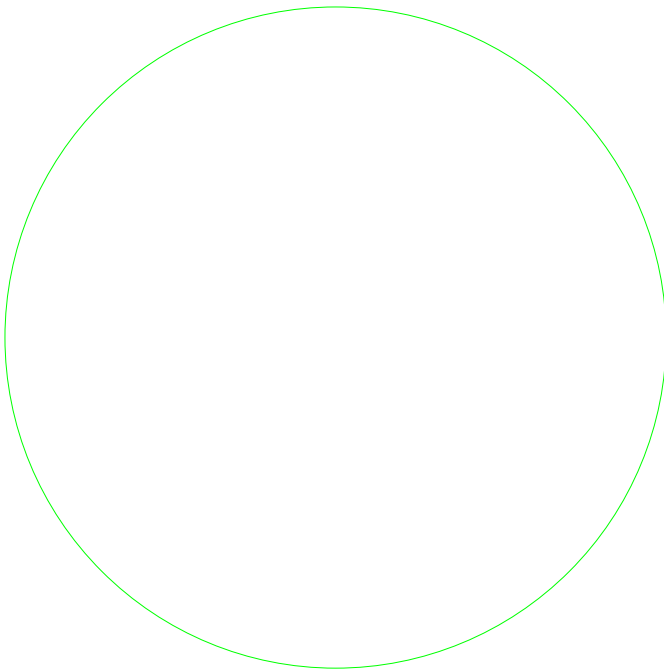


Dibuja un círculo en (0, 0) con radio 1 y lo colorea de verde

```
In[*]:= Graphics[Style[Circle[], Green]]
```

gráfico estilo círculo verde

Out[*]=

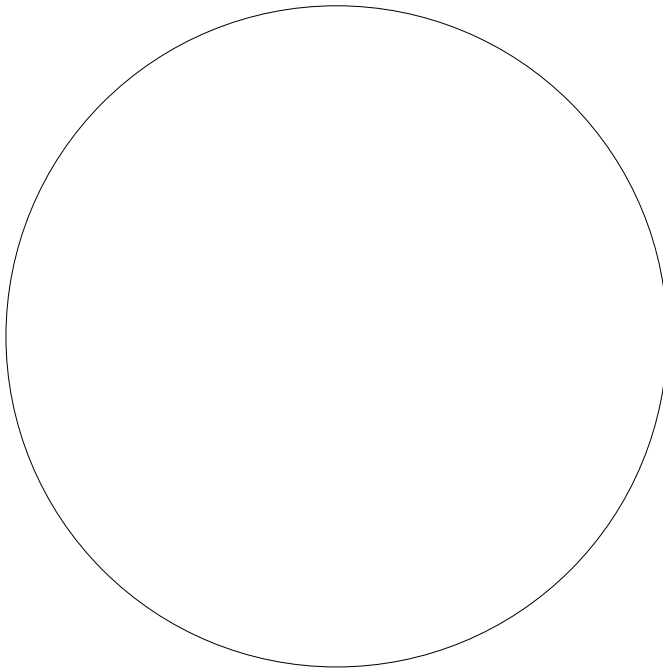


Dibuja un círculo con centro en (1, 1) y radio 0.5

```
In[ ]:= Graphics[Circle[{1, 1}, 0.5]]
```

gráfico círculo

Out[]=

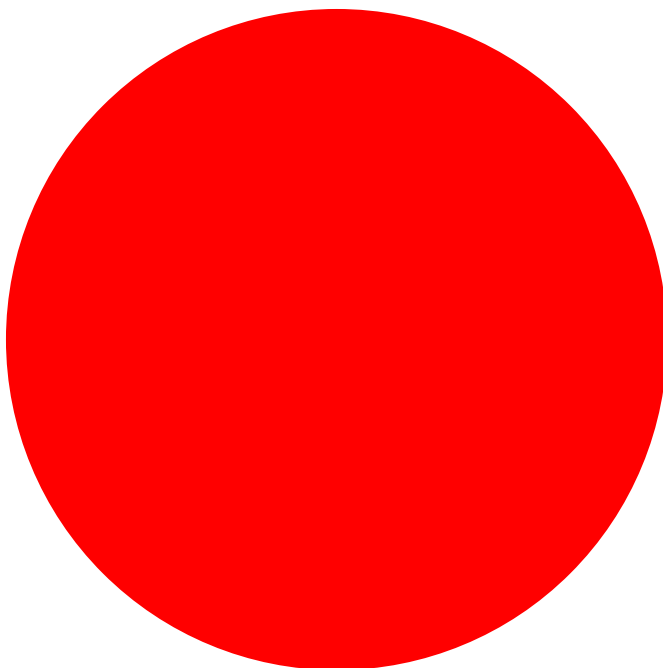


Dibuja un disco sólido con centro en (0, 0) y radio 1, de color rojo

```
In[ ]:= Graphics[Style[Disk[], Red]]
```

gráfico estilo disco rojo

Out[]=

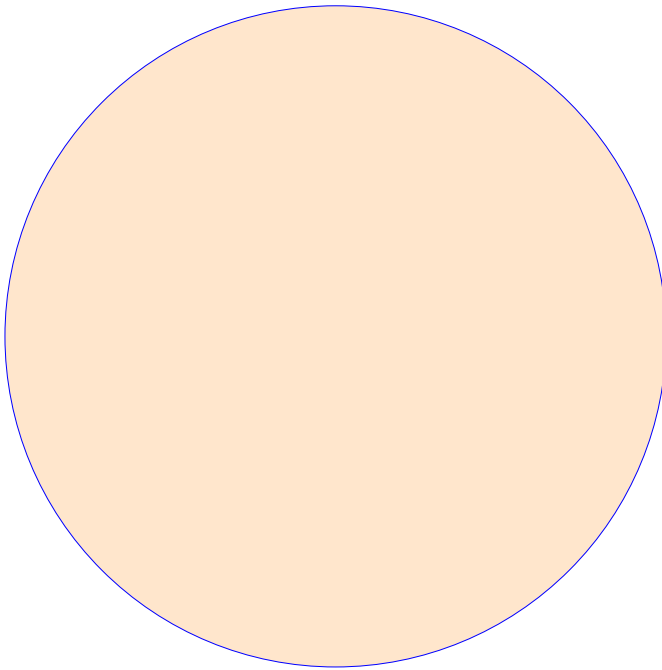


Dibuja un disco naranja claro con centro en (0, 0) y radio 1, y encima un círculo azul en el mismo lugar


```
In[*]:= Graphics[{LightOrange, Disk[], Blue, Circle[]}]
```

gráfico naranja claro disco azul círculo

Out[*]=

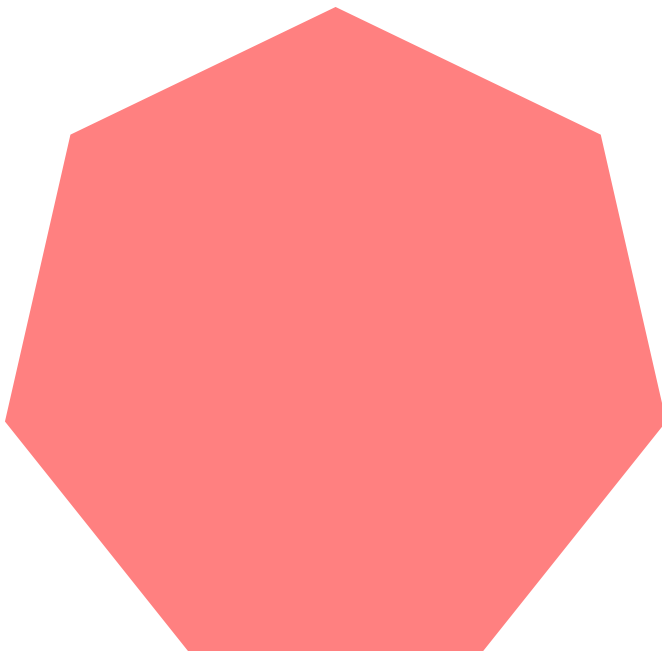


Dibuja un polígono regular de 7 lados centrado en (0, 0) y lo colorea de rosa

```
In[*]:= Graphics[Style[RegularPolygon[7], Pink]]
```

gráfico estilo polígono regular rosa

Out[*]=



Dibuja un polígono regular de 4 lados (cuadrado) centrado en (0, 0) y lo colorea de verde claro

```
In[*]:= Graphics[Style[RegularPolygon[4], LightGreen]]
```

gráfico estilo polígono regular verde claro

Out[*]=

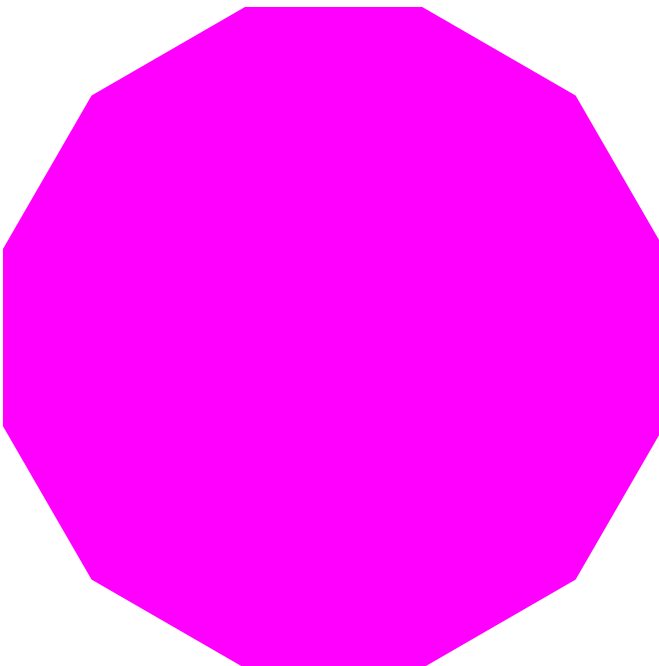


Dibuja un polígono regular de 12 lados centrado en (0, 0) y lo colorea de magenta

```
In[*]:= Graphics[Style[RegularPolygon[12], Magenta]]
```

gráfico estilo polígono regular magenta

Out[*]=

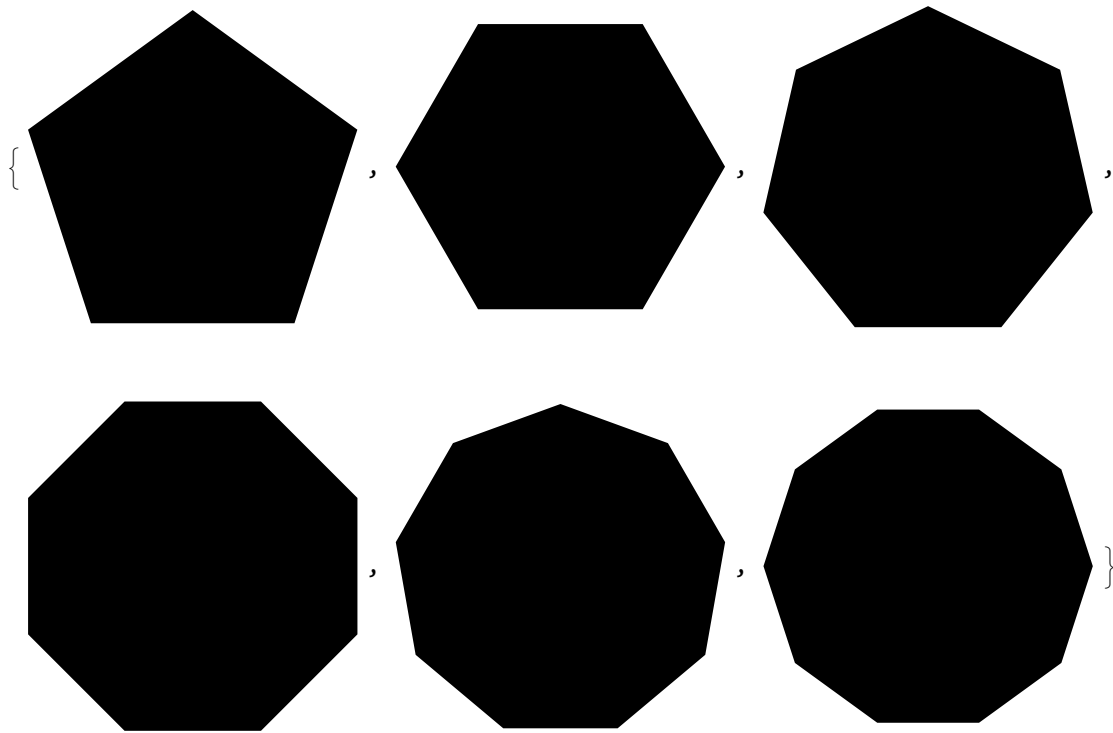


Genera gráficos de polígonos regulares con lados desde 5 hasta 10, uno para cada valor de n

```
In[*]:= Table[Graphics[RegularPolygon[n]], {n, 5, 10}]
```

tabla gráfico polígono regular

Out[*]=

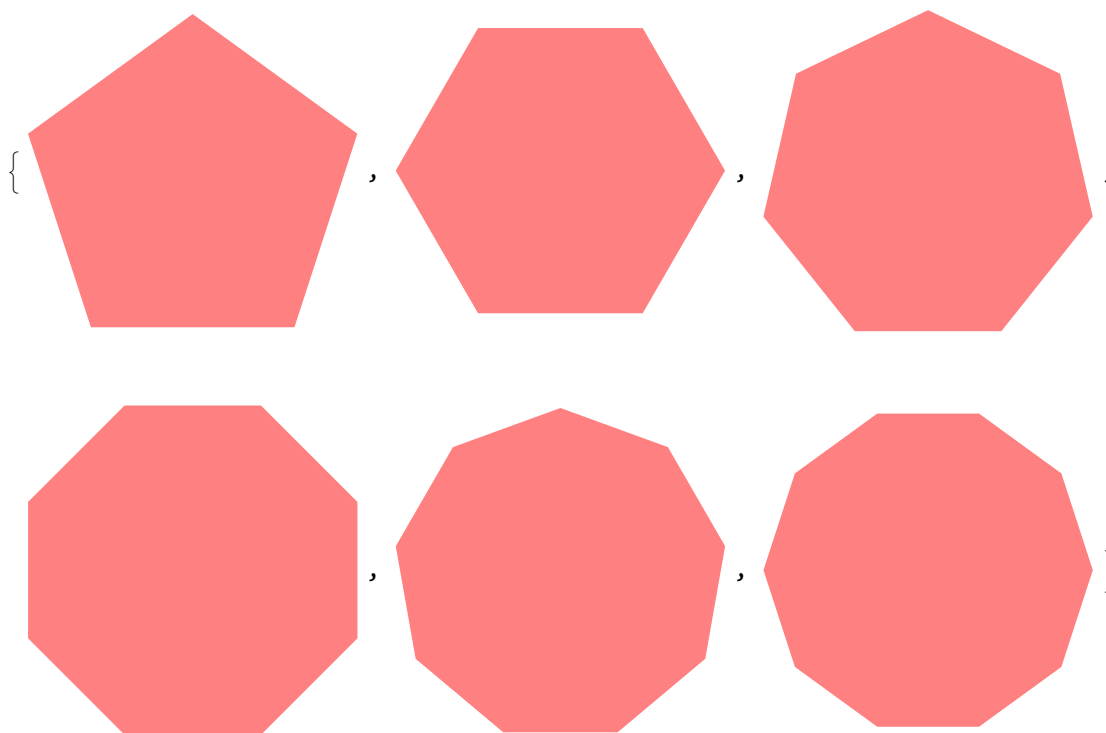


```
# Genera gráficos de polígonos regulares con lados desde 5 hasta 10, uno para cada valor de n en color rosado
```

```
In[*]:= Table[Graphics[Style[RegularPolygon[n], Pink]], {n, 5, 10}]
```

tabla gráfico estilo polígono regular rosa

Out[*]=



7.2 Gráficas Tridimensionales (3D)

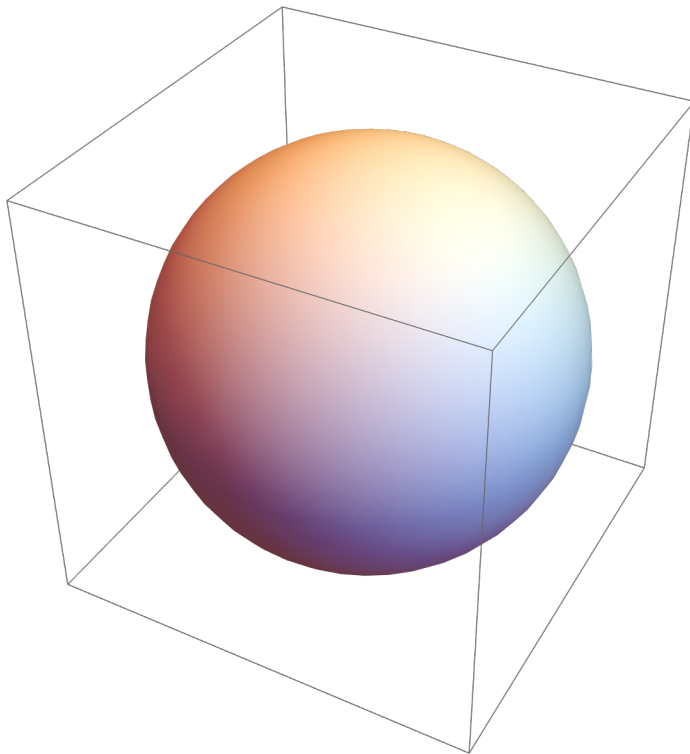
✎ **Explicación:** Las gráficas tridimensionales amplían la visualización al agregar una tercera dimensión, lo que permite representar funciones de dos variables o datos con tres componentes. Mathematica ofrece herramientas como `Plot3D`, `Graphics3D` y `ListPointPlot3D` para crear estos gráficos. Las visualizaciones 3D ayudan a comprender superficies, formas y estructuras espaciales, y permiten rotar y explorar los objetos desde diferentes ángulos para un análisis más completo.

Dibuja una esfera en 3D con centro en (0, 0, 0) y radio 1 por defecto

```
In[ ]:= Graphics3D[Sphere[ ]]
```

gráfico 3D esfera

Out[]=

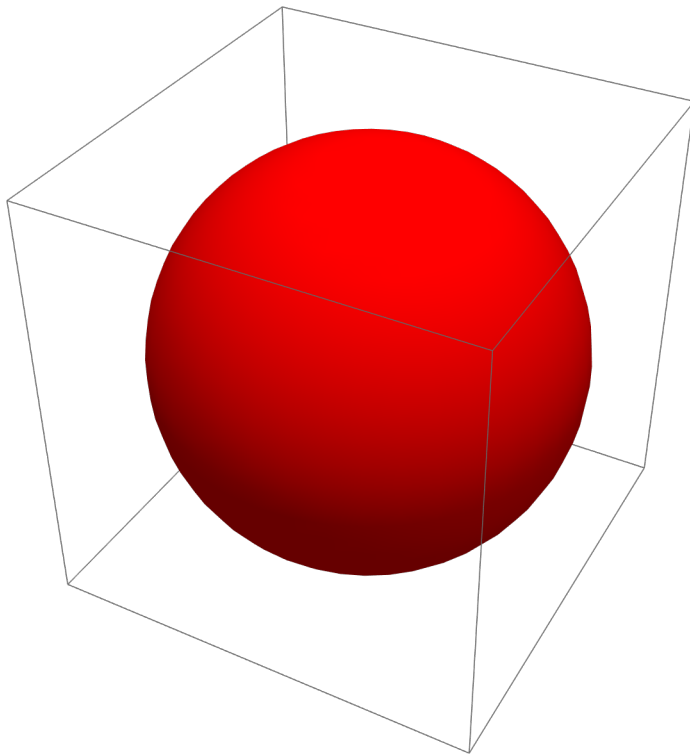


Dibuja una esfera 3D roja con centro en $(0, 0, 0)$ y radio 1 por defecto

```
In[*]:= Graphics3D[Style[Sphere[], Red]]
```

gráfico 3D estilo esfera rojo

Out[*]=

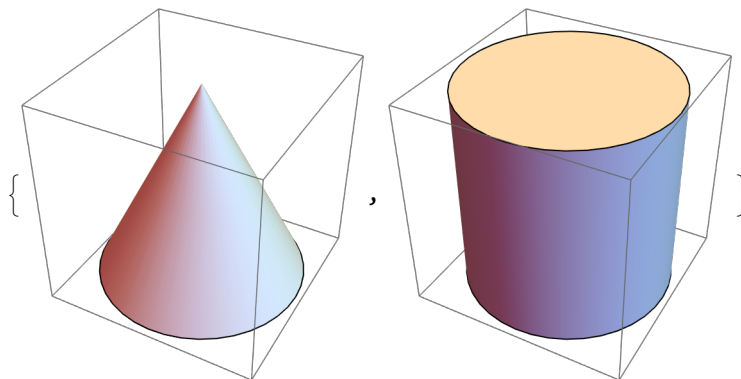


Dibuja un cono y un cilindro 3D, ambos con dimensiones y posiciones por defecto

```
In[*]:= {Graphics3D[Cone[]], Graphics3D[Cylinder[]]}
```

gráfico 3D cono gráfico 3D cilindro

Out[*]=

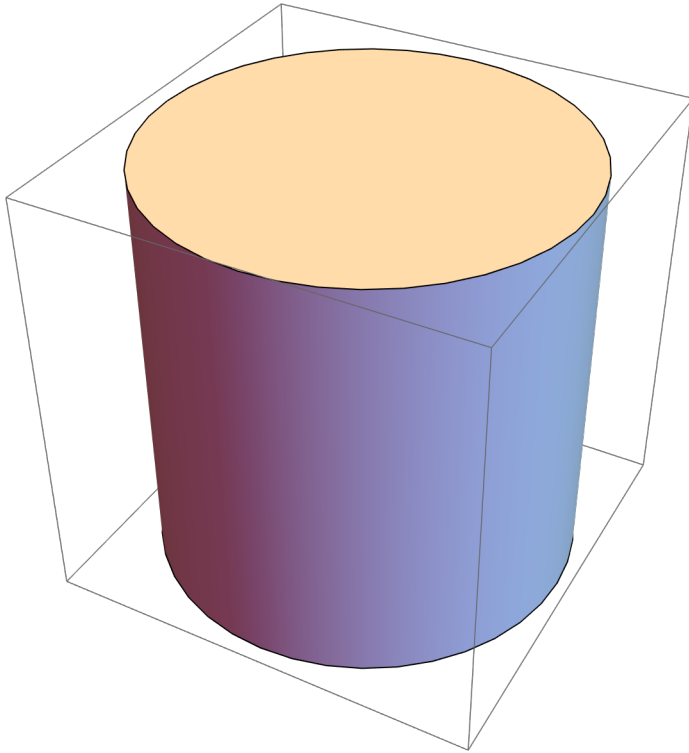


Dibuja un cilindro 3D con base en $(0, 0, 0)$, altura 1 y radio 1 por defecto

```
In[ ]:= Graphics3D[Cylinder[ ]]
```

gráfico 3D cilindro

Out[]=

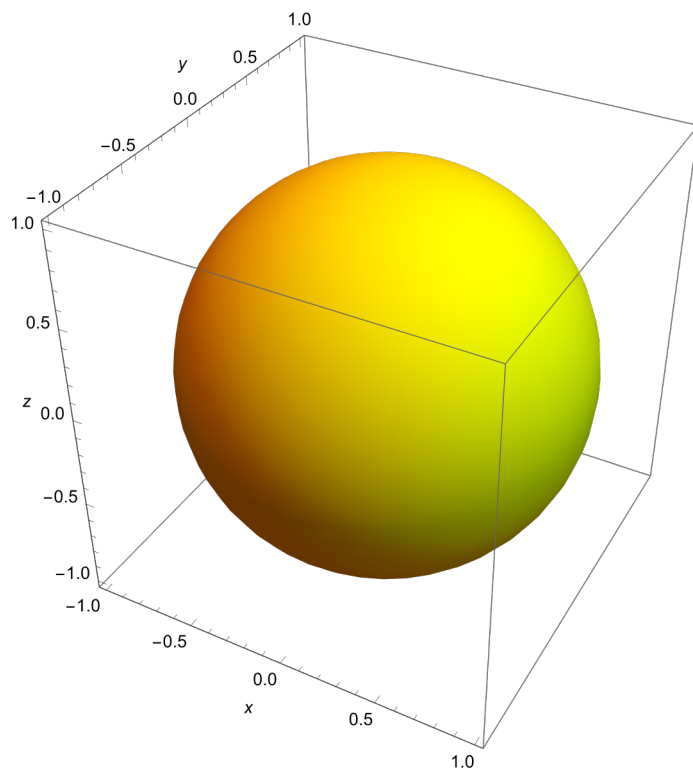


```
# Dibuja una esfera 3D amarilla con centro en (0,0,0) y radio 1, mostrando ejes con etiquetas x, y, z
```

```
In[ ]:= Graphics3D[Style[Sphere[], Yellow], Axes → True, AxesLabel → {x, y, z}]
```

gráfico 3D estilo esfera amarillo ejes verd... etiqueta de ejes

Out[]:=

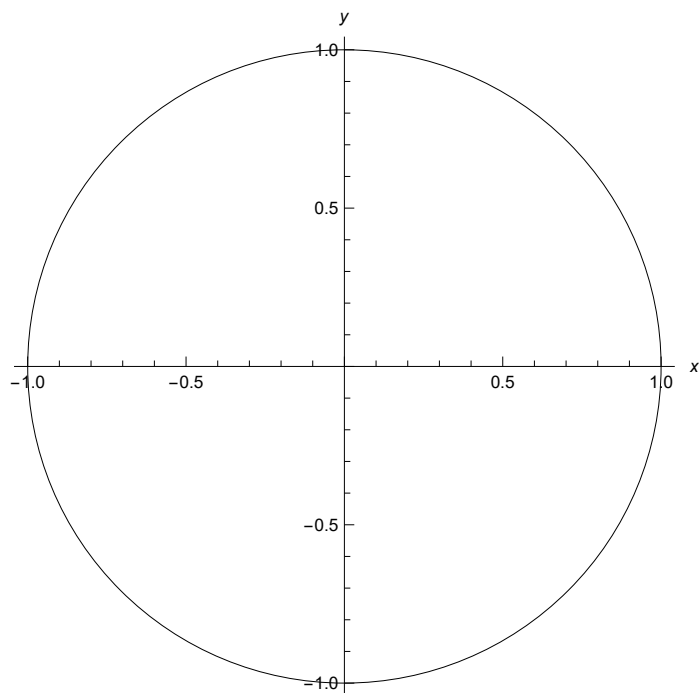


Dibuja un círculo con centro en (0, 0) y radio 1, mostrando ejes con etiquetas x y y

```
In[ ]:= Graphics[Circle[], Axes → True, AxesLabel → {x, y}]
```

gráfico círculo ejes verd... etiqueta de ejes

Out[]:=



7.3 Manipuladores Interactivos

✎ **Explicación:** Los manipuladores interactivos son controles dinámicos, como deslizadores, botones y menús, que permiten cambiar parámetros de una gráfica o cálculo en tiempo real. En Mathematica, la función `Manipulate` integra estos controles con las visualizaciones para crear experiencias interactivas. Esto facilita la exploración de cómo distintos valores afectan una función o gráfica, promoviendo un aprendizaje activo y una comprensión más profunda de conceptos matemáticos.

Crea un manipulador interactivo para evaluar x^n con n desde 1 hasta 10 en pasos de 2

```
In[ ]:= Manipulate[x^n, {n, 1, 10, 2}]
|manipula
```

Out[]:=



Binomio de Newton

```
In[ ]:= Manipulate[Expand[(a + b)^n], {n, 1, 10, 1}]
|manipula |expande factores
```

- **Forma Binomial:** $(a + b)^n = \sum_{k=0}^n \binom{n}{k} a^{n-k} b^k$

`Sum[Binomial[n, k] a^(n - k) b^k, {k, 0, n}]` genera directamente el desarrollo del binomio sin recurrir a `Expand`

Out[]:=



💎 Nota:

- `Sum[Binomial[n, k] a^(n - k) b^k, {k, 0, n}]` genera directamente el desarrollo del binomio sin recurrir a `Expand`.
- `Manipulate` permite variar n dinámicamente de 1 a 10.

```
In[ ]:= Manipulate[Sum[Binomial[n, k] a^(n - k) b^k, {k, 0, n}], {n, 1, 10, 1}]
|manipula |S... |número binomial
```

Out[]:=



Muestra una tabla con $\{i^n, (2i)^n\}$ para i de 1 a 10 y n de 1 a 10 en pasos de 2

In[]:= Manipulate[Table[$\{i^n, (2i)^n\}$, {i, 1, 10}], {n, 1, 10, 2}]

Out[]:=

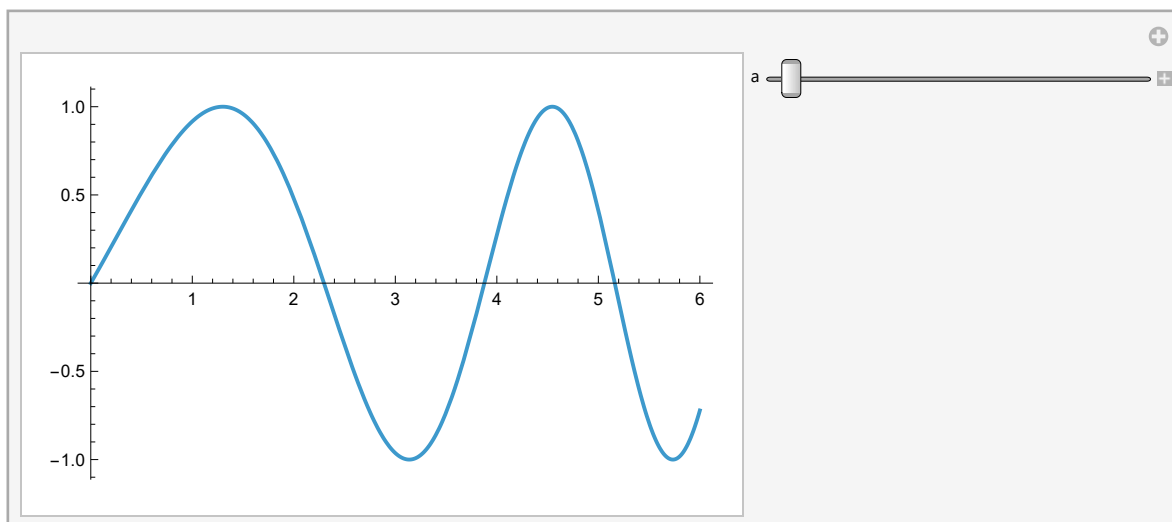
$\{\{1, 2\}, \{2, 4\}, \{3, 6\},$
 $\{4, 8\}, \{5, 10\}, \{6, 12\}, \{7, 14\},$
 $\{8, 16\}, \{9, 18\}, \{10, 20\}\}$

ServiceFramework`Private`n

Grafica $\sin[x(1+a)]$ en x de 0 a 6, con a entre 0 y 10

In[]:= Manipulate[Plot[Sin[x (1 + a)], {x, 0, 6}], {a, 0, 10}]

Out[]:=



Crea un manipulador interactivo para factorizar $x^n + 1$ con n de 10 a 100 en pasos de 1

In[]:= Manipulate[Factor[$x^n + 1$], {n, 10, 100, 1}]

Out[]:=

$(1 + x^2) (1 - x^2 + x^4 - x^6 + x^8)$

ServiceFramework`Private`n

Calcular el factorial de n con n de 1 a 50 en pasos de 1

In[]:= Manipulate[n!, {n, 1, 50, 1}]
 |manipula

Out[]:=



Crea un manipulador interactivo que genera una lista con n veces el color Orange, con n de 1 a 5

In[]:= Manipulate[Table[Orange, n], {n, 1, 5, 1}]
 |manipula |tabla |naranja

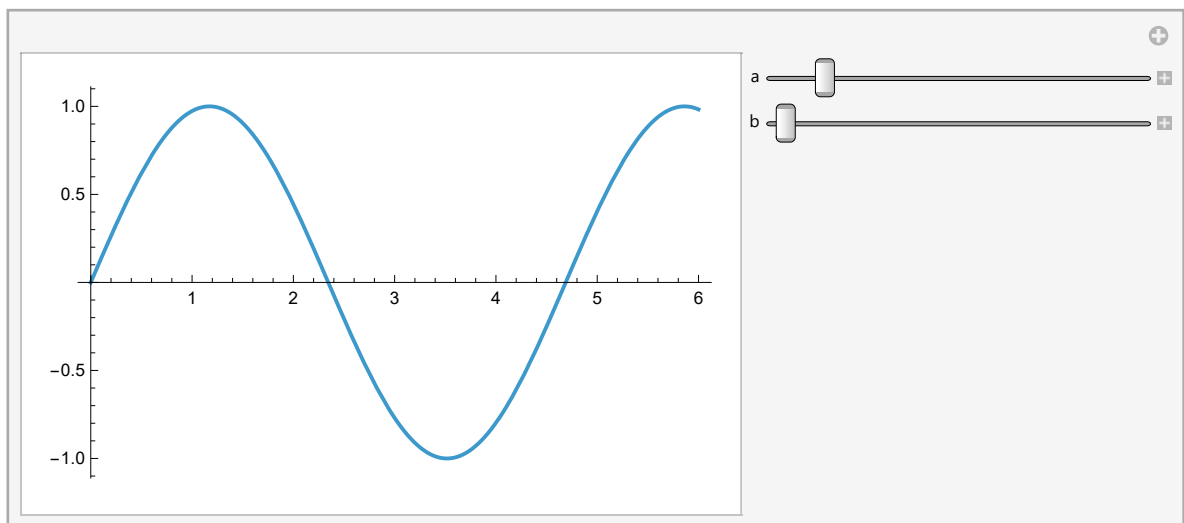
Out[]:=



Grafica Sin[a x + b] en x de 0 a 6, con a entre 1 y 4, y b entre 0 y 10

In[]:= Manipulate[Plot[Sin[a x + b], {x, 0, 6}], {a, 1, 4}, {b, 0, 10}]
 |manipula |repre... |seno

Out[]:=



Crea un manipulador interactivo que muestra en columna los valores n, n^2 y n^3 para n de 1 a 10

In[]:= Manipulate[Column[{n, n^2, n^3}], {n, 1, 10, 1}]
 |manipula |columna

Out[]:=

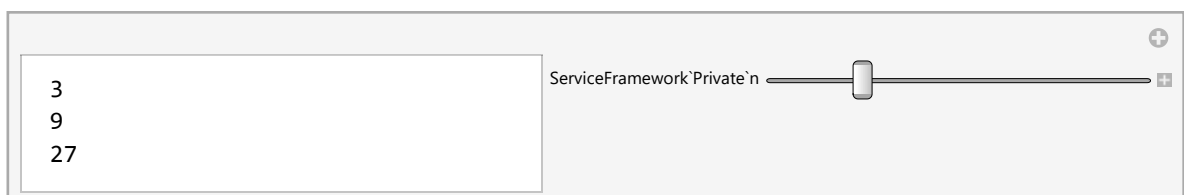


Gráfico de barras 3D con alturas que dependen de a , variando a de 0 a 5

```
In[ ]:= Manipulate[BarChart3D[{1, a, 4, 2 * a, 4, 3 * a, 1}], {a, 0, 5}]
```

manipula diagrama de barras 3D

Out[]=

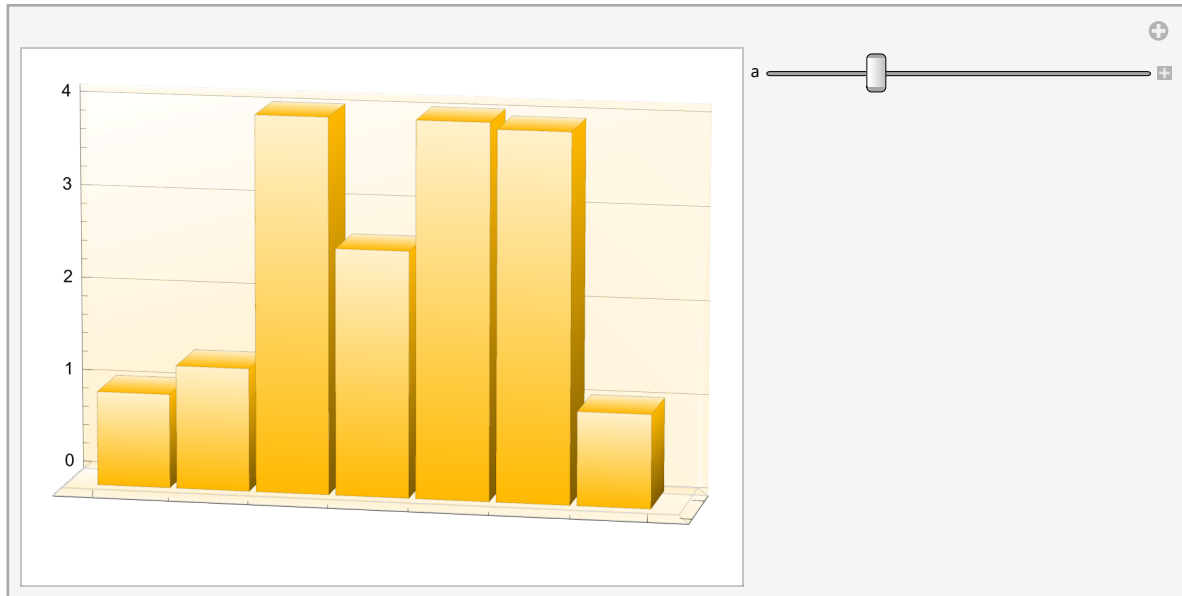
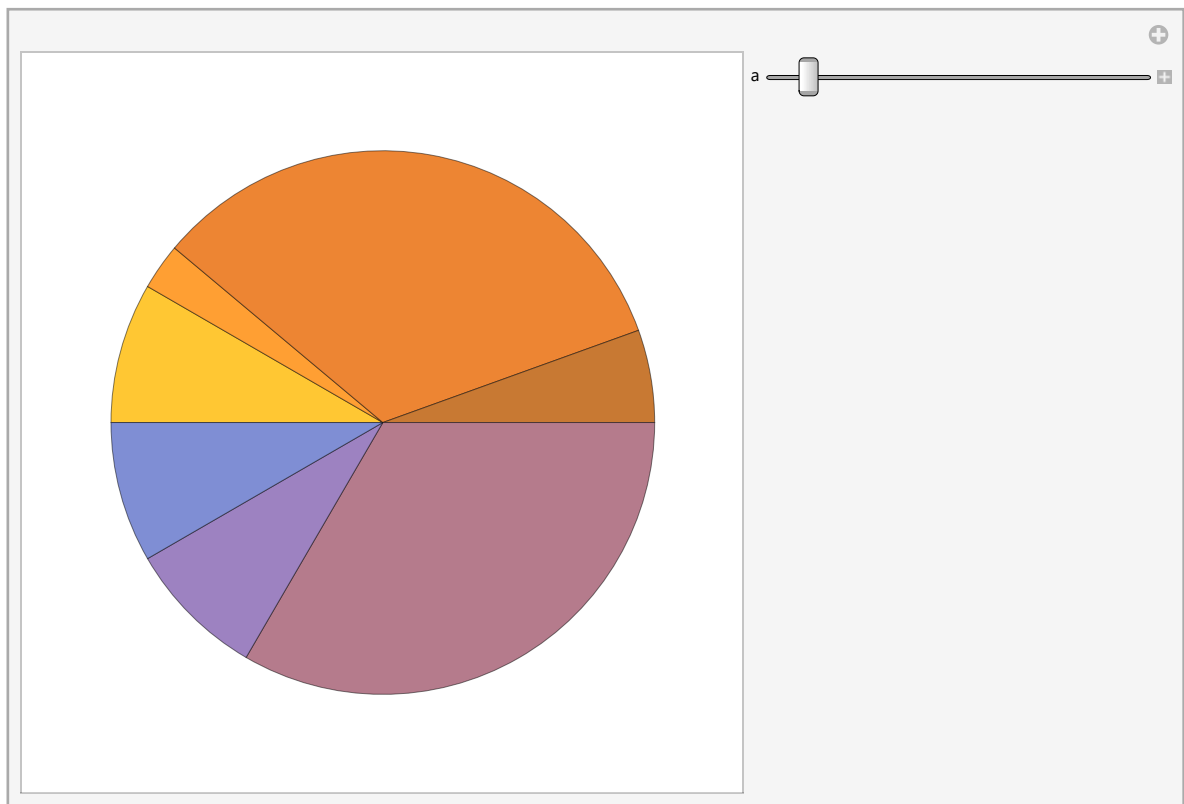


Gráfico circular con sectores que dependen de a , variando a de 0 a 5

In[*]:= Manipulate[PieChart[{1, a, 4, 2 * a, 4, 3 * a, 1}], {a, 0, 5}]

manipula diagrama circular

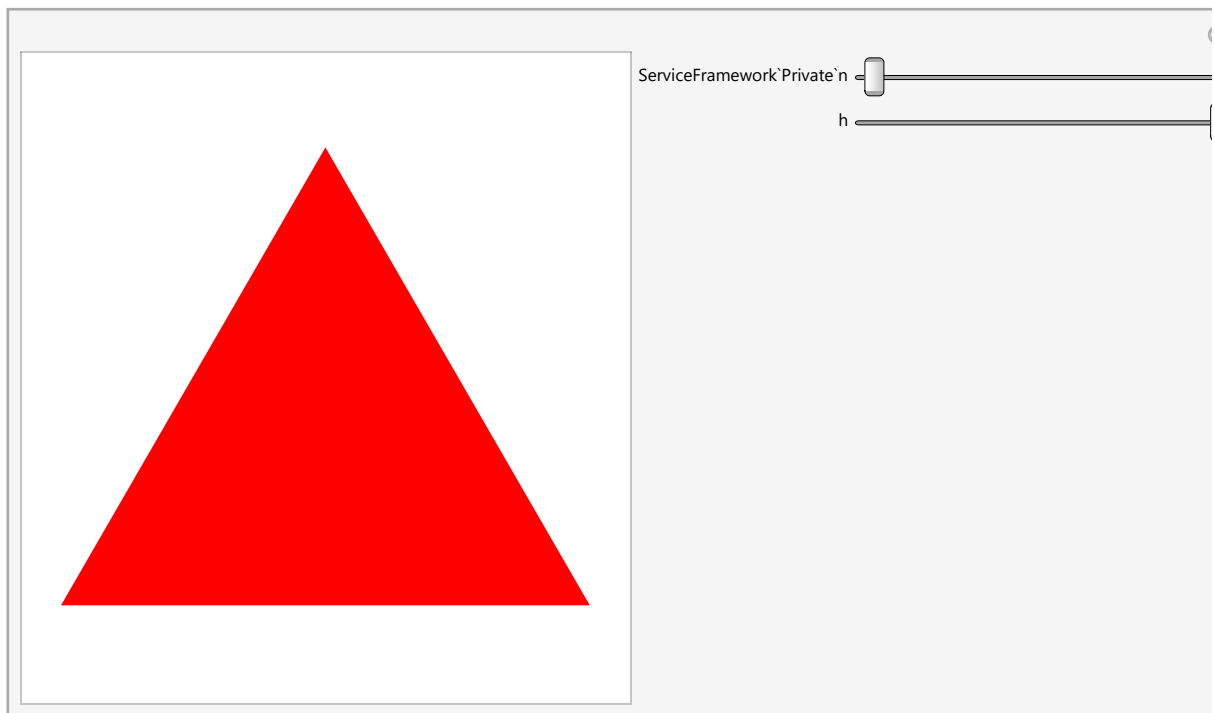
Out[*]=



Crea un manipulador interactivo que dibuja un polígono regular con n lados y color según Hue[h]; n varía de 3 a 20 y h de 0 a 1

```
In[*]:= Manipulate[Graphics[Style[RegularPolygon[n], Hue[h]]], {n, 3, 20, 1}, {h, 0, 1}]
```

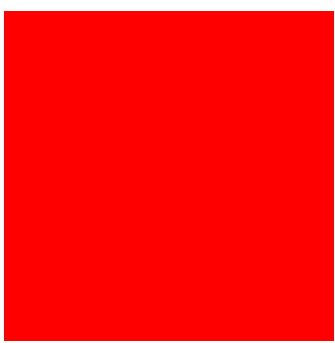
Out[*]=

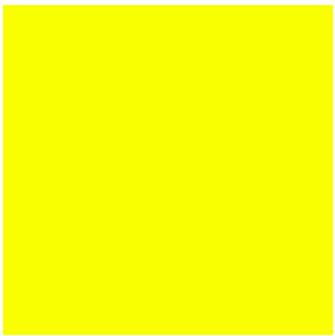
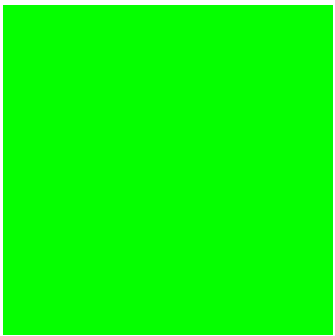
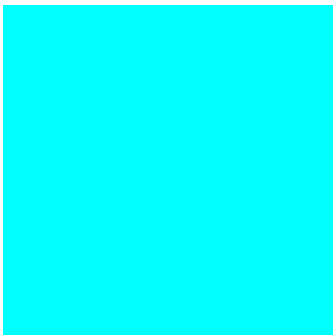
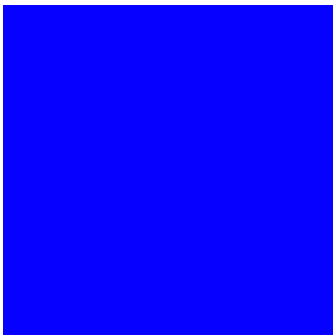
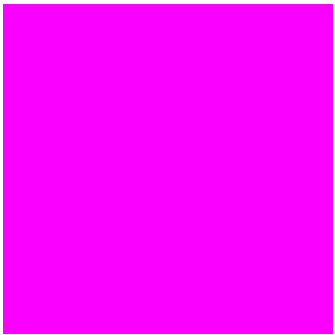



💎 **Nota:** Espectro de Hue[h] (Hue, Saturation, Brightness), para $h \in [0,1]$; El cambio es suave y continuo. A medida que se mueve el deslizador de h, el color del polígono cambia por todo el arcoíris.

```
In[*]:= Grid[Prepend[Table[{Style[NumberForm[h, {2, 2}], Style[Graphics[{Hue[h], Rectangle[]}]],
Which[h == 0.0, "Red", h ≤ 0.17, "Yellow", h ≤ 0.33, "Green", h ≤ 0.5,
"Cyan", h ≤ 0.67, "Blue", h ≤ 0.83, "Magenta", h ≤ 1.0, "Red"]],
{h, {0.0, 0.17, 0.33, 0.5, 0.67, 0.83, 1.0}}, {"h", "Color", "Color Aproximado"}], Frame → All]
```

Out[*]=

h	Color	Color Aproximado
0.00		Red

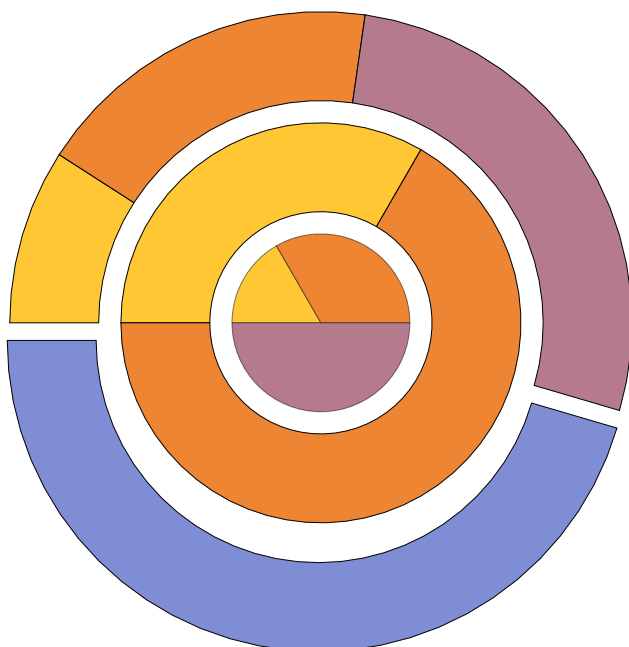
0.17		Yellow
0.33		Green
0.50		Cyan
0.67		Blue
0.83		Magenta

1.00		Red
------	---	-----

Intenta crear un gráfico circular con tres conjuntos de datos: {1,2,3}, {1,2}, y {1,2,3,5}, pero PieChart espera una lista plana de valores numéricos

```
In[ ]:= PieChart[ {Range[3], Range[2], {1, 2, 3, 5}} ]
          |diagrama ci... |rango      |rango
```

Out[]:=



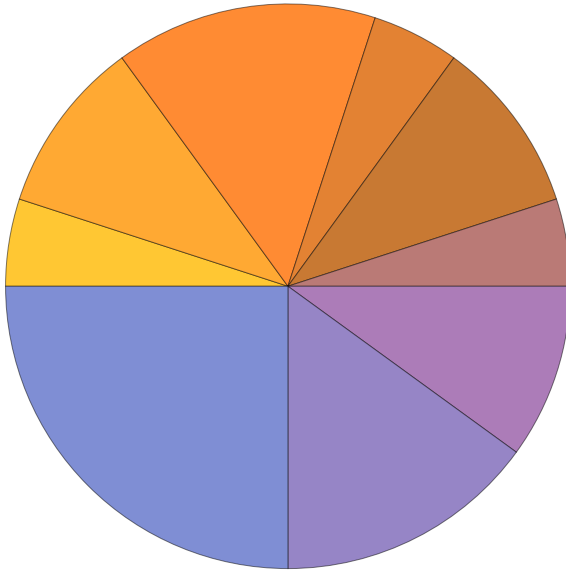
💎 Nota:

- PieChart no acepta listas anidadas; se debe pasar una sola lista con números.
- PieChart espera una lista plana de valores.
- Pasar una lista anidada genera un error o un gráfico que no tiene sentido, porque intenta interpretar cada sublista como un sector (lo cual no está definido).
- No hace un gráfico de dona ni agrupa sectores de forma esperada.

Crea un gráfico circular con los valores {1, 2, 3, 1, 2, 1, 2, 3, 5} combinando todas las listas


```
In[ ]:= PieChart[Flatten[{Range[3], Range[2], {1, 2, 3, 5}}]]
```

```
Out[ ]:=
```



💎 **Nota:** La función **Flatten** toma una lista anidada y la convierte en una lista plana. Es decir, si tienes una lista que contiene otras listas dentro de ella, Flatten las “aplana” y las combina en una sola lista.

```
# Dibuja un gráfico circular con 3 sectores y estilos personalizados
# Colorea sectores en rojo, verde y azul
# Agrega leyendas para cada sector
# Título del gráfico
# Ángulo inicial del primer sector
# Etiquetas radiales para cada sector
# Bordes negros para los sectores
```

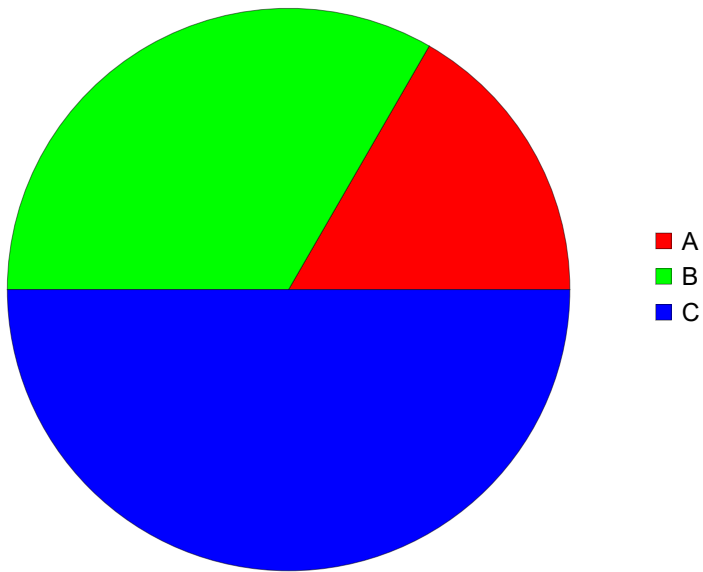
```

In[ ]:= PieChart[ {1, 2, 3}, ChartStyle → {Red, Green, Blue},
|diagrama circular |estilo de diagrama |rojo |verde |azul
ChartLegends → {"A", "B", "C"}, PlotLabel → "Gráfico de dona",
|leyendas de diagrama |con... |etiqueta de representación
SectorOrigin → 0, ChartLabels → Placed[Automatic, "RadialCallout"],
|origen de sector |etiquetas de dia... |colocado |automático
ChartBaseStyle → Directive[EdgeForm[Black] ] ]
|estilo base de diag... |directiva |forma de ... |negro

```

Out[]:=

Gráfico de dona



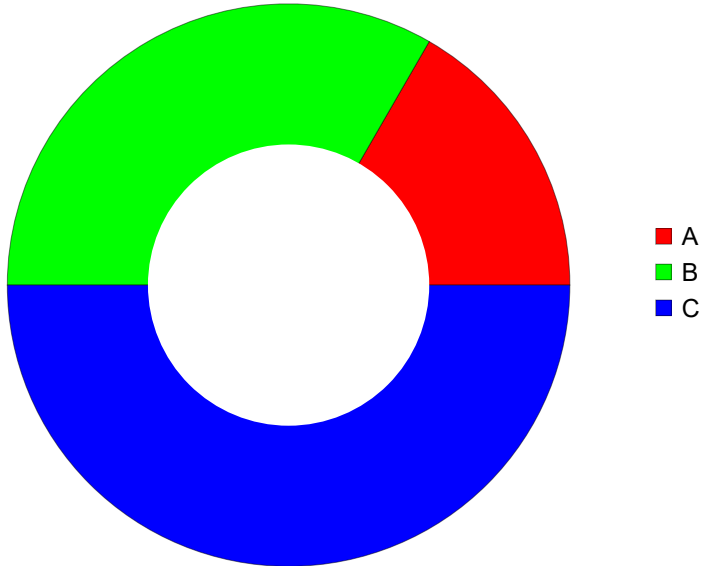
Superpone un disco blanco en el centro para simular el "agujero" de la dona

```
In[ ]:= Show[%, Graphics[{White, Disk[{0, 0}, 0.5]}]]
```

muestra gráfico blanco disco

Out[]:=

Gráfico de dona



■ **🔗 Reto en clase — Gráfica de círculo interactivo**

Graficar un círculo con radio interactivo, es decir, que crezca la figura en el plano.

```
In[ ]:= ?PlotRange
```

rango de representación

Out[]:=

Symbol ⓘ

PlotRange is an option for graphics functions that specifies what range of coordinates to include in a plot.

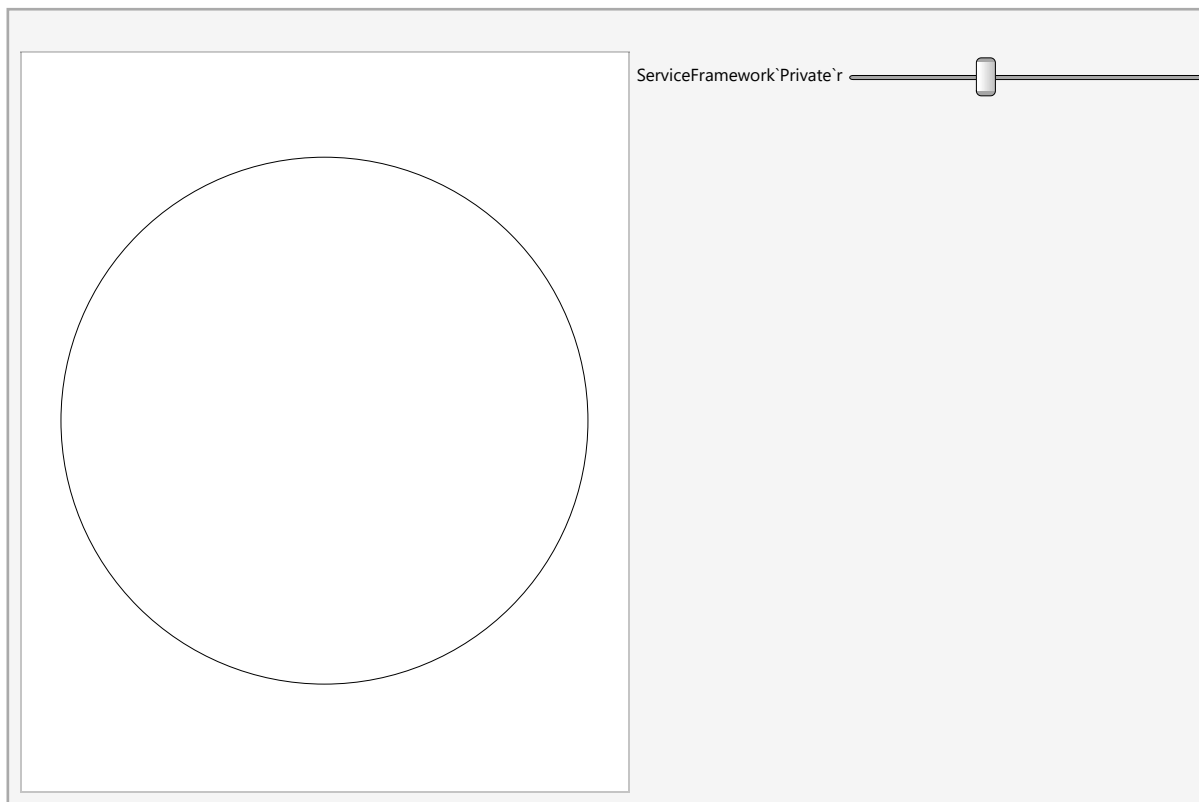
▼

```
# Crea un manipulador interactivo para dibujar un círculo con centro en (0, 0) y radio r
# El valor de r varía entre 1 y 10
```

```
In[ ]:= Manipulate[Graphics[Circle[{0, 0}, r]], {r, 1, 10}]
```

manipula gráfico círculo

Out[]=



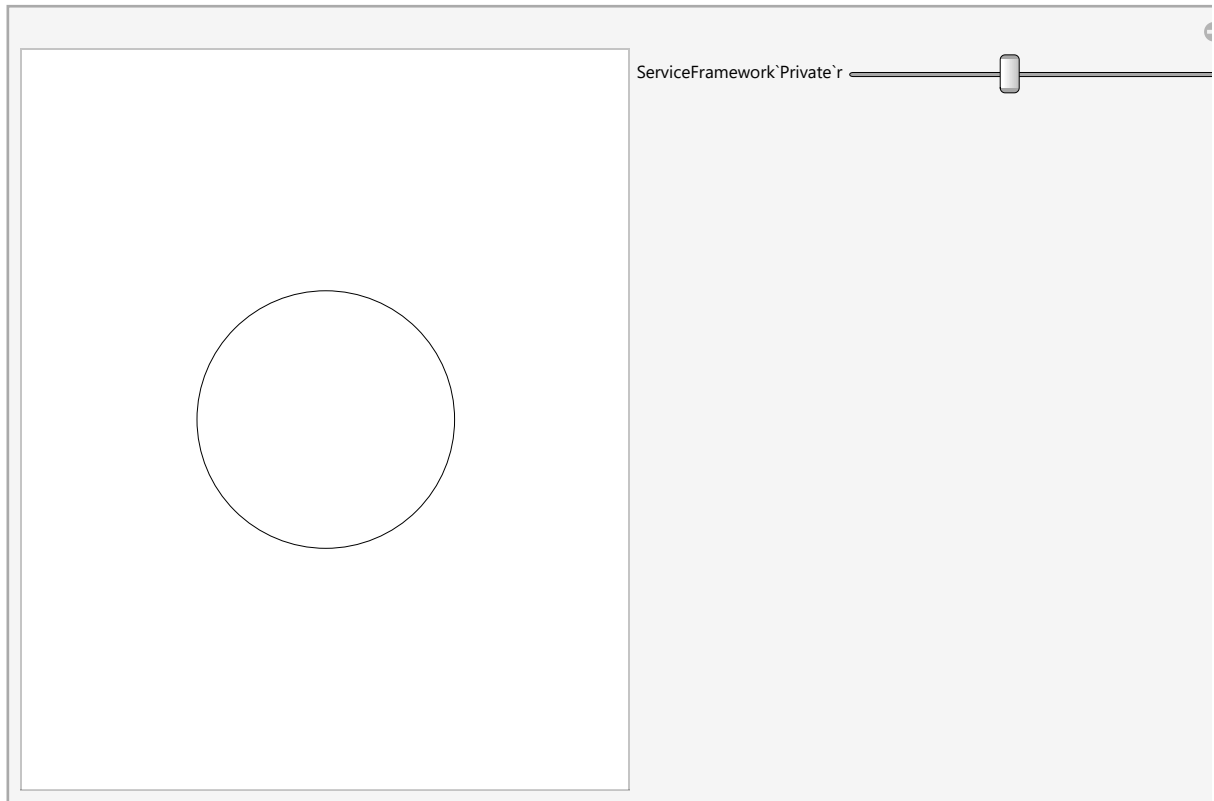
💎 **Nota:** El valor de **r** debe tener un rango definido, ya que de lo contrario no se puede modificar dinámicamente el radio del círculo.

```
# Crea un manipulador interactivo para dibujar un círculo con centro en (0, 0) y radio r
# El valor de r varía entre 1 y 10
# Se define el rango de los ejes con PlotRange para asegurar que el círculo siempre se muestre completamente en el
área de gráficos de tamaño fijo
```

```
In[ ]:= Manipulate[Graphics[Circle[{0, 0}, r], PlotRange → {{-10, 10}, {-10, 10}}], {r, 1, 10}]
```

manipula gráfico círculo rango de representación

Out[]=

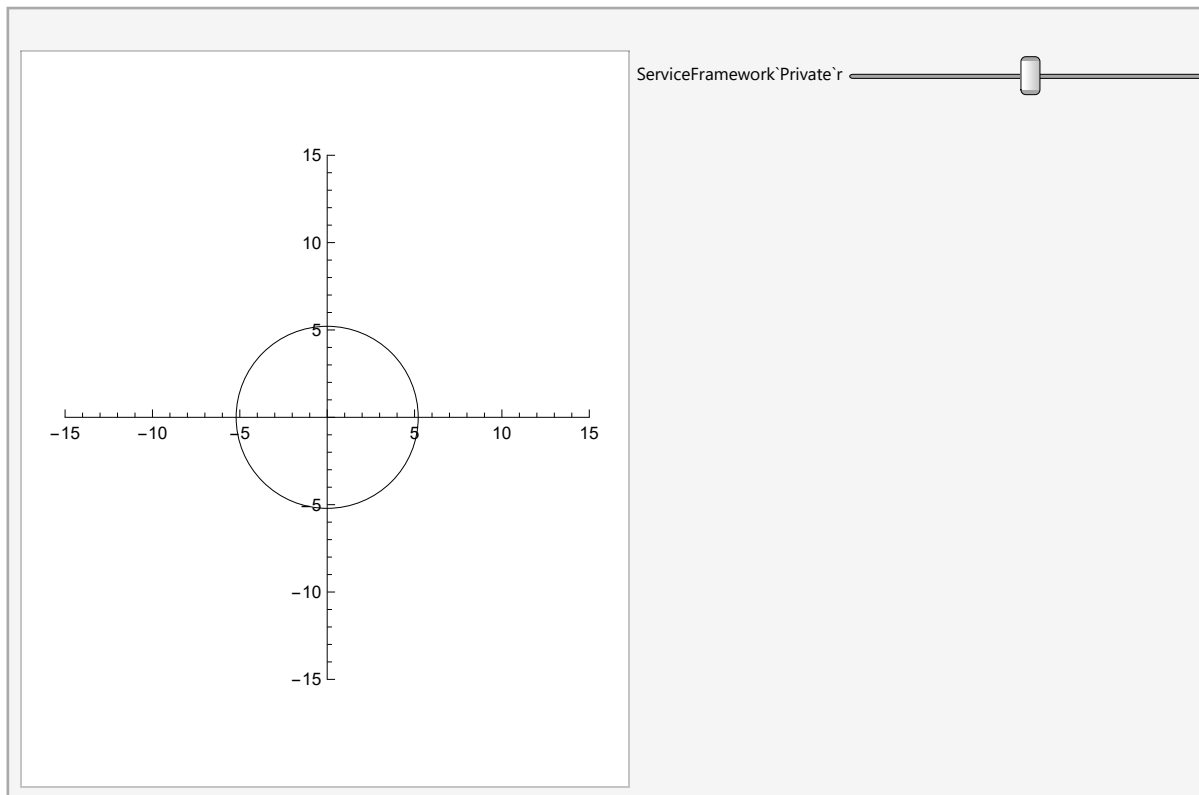


💎 **Nota:** El rango de los ejes está fijo en $\{-10, 10\}$, por lo que el círculo siempre se ajustará dentro de este espacio, sin importar el valor de r

```
# Crea un manipulador interactivo para dibujar un círculo con centro en (0, 0) y radio r
# El valor de r varía entre 1 y 10
# Se define el rango de los ejes con PlotRange en {-15, 15} para que el círculo siempre se dibuje dentro de esta área
# Los ejes están activados con Axes -> True para mostrar las coordenadas en el gráfico
```

```
In[ ]:= Manipulate[Graphics[Circle[{0, 0}, r],
  |manipula |gráfico |círculo
  PlotRange -> {{-15, 15}, {-15, 15}}, Axes -> True], {r, 1, 10}]
  |rango de representación |ejes |verdadero
```

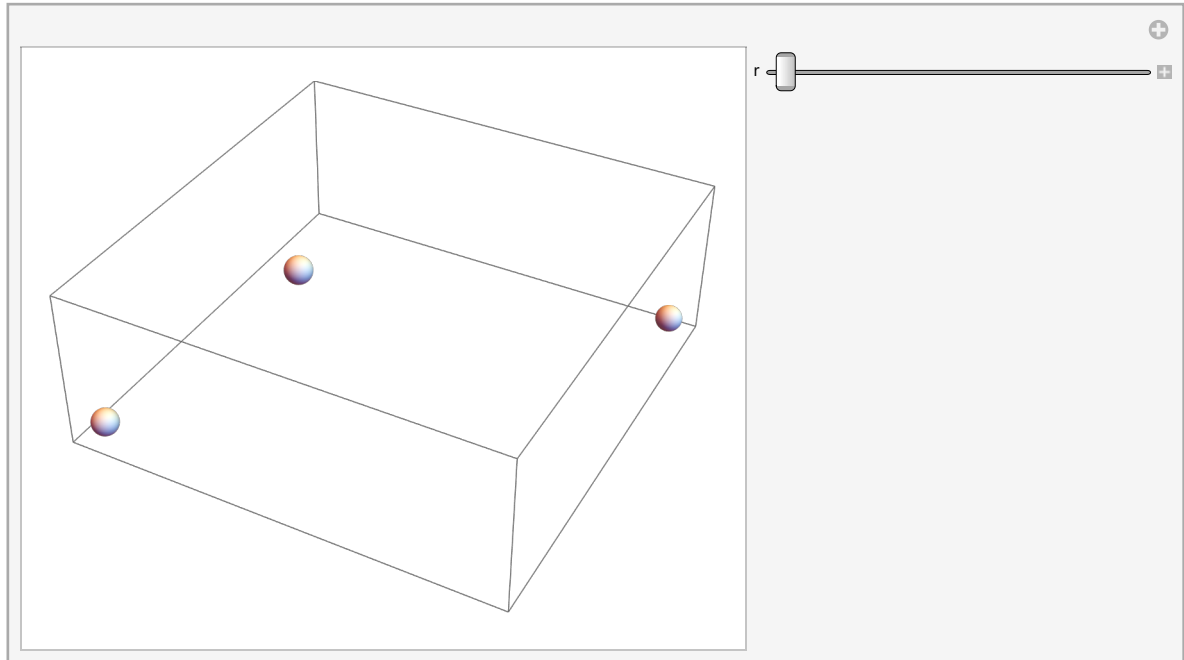
Out[]=



Esferas 3D

```
In[1]:= Manipulate[
  Graphics3D[{Sphere[{0, 0, 0}, r], Sphere[{3, 3, 0}, r], Sphere[{1, 1, 1}, r]}],
  {r, 0.1, 5, 0.1}]
```

Out[1]=

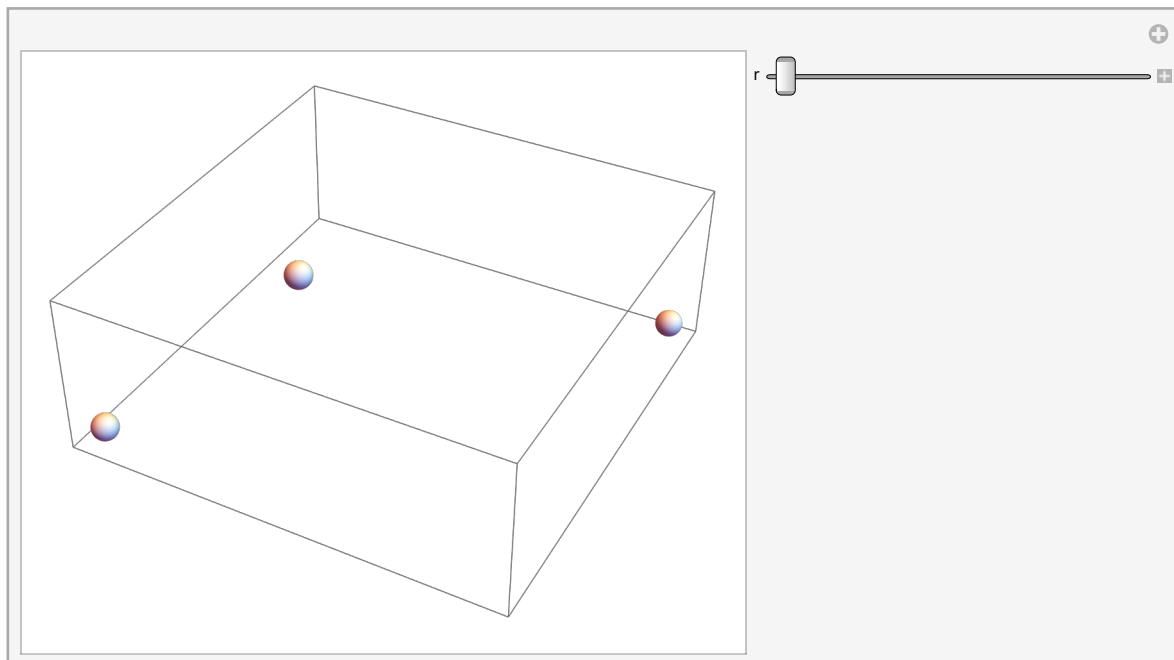


- **🌀 Reto en clase — Gráfica esfera interactiva**
Graficar tres esferas con radio interactivo, es decir, que crezca la figura en el plano; adicionalmente no deben solaparse y mantener distancia entre sí.
- **Ejemplo en clase mejorado (sin solaparse)**

```
# Esferas 3D
# Graphics3D[{Sphere[{0,0,0}, r] -> Esfera con coordenadas (0, 0, 0)}
# {r, 0.1, 0.8, 0.1} -> radio con rango de 0.1 a 0.8 con pasos de 0.1
# se modifica el radio para que las esferas no se solapen entre ellas
```

```
In[*]:= Manipulate[
  Graphics3D[{Sphere[{0, 0, 0}, r], Sphere[{3, 3, 0}, r], Sphere[{1, 1, 1}, r]}],
  {r, 0.1, 0.8, 0.1}]
```

Out[*]=



■ Mejoras personales

```
# Crea un manipulador interactivo para dibujar tres esferas 3D de diferentes colores y tamaños, con centro en
diferentes posiciones
# Las esferas están coloreadas: roja en (-7, 0, 0), verde en (0, 0, 0), y azul en (7, 0, 0)
# El radio de las esferas varía entre 0.1 y 3
# Se activan los ejes 3D con Axes -> True y se etiquetan como "x", "y", y "z"
# Boxed -> False elimina el cuadro de la escena 3D
# PlotRange está configurado para que los ejes varíen en el rango x: {-10, 10}, y: {-5, 5}, z: {-5, 5}
```

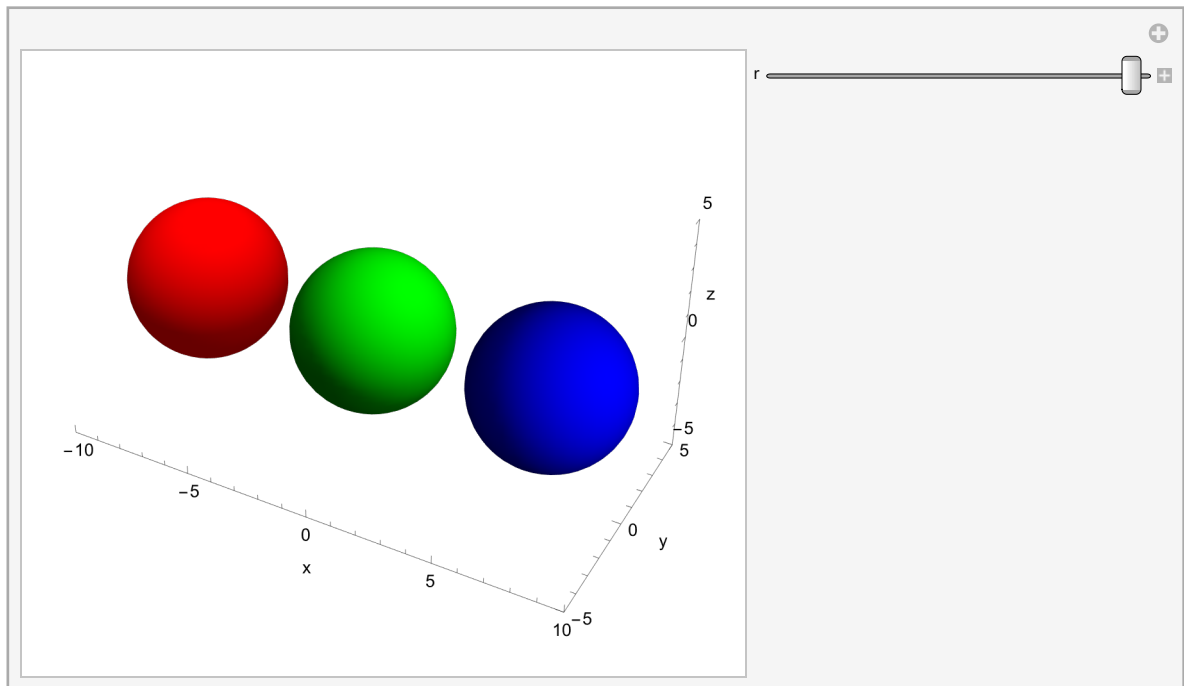


```

In[ ]:= Esfera3d = Manipulate[
  Graphics3D[ {Style[Sphere[{-7, 0, 0}, r], Red], Style[Sphere[{0, 0, 0}, r], Green],
    Style[Sphere[{7, 0, 0}, r], Blue] }, Axes -> True, AxesLabel -> {"x", "y", "z"},
    Boxed -> False, PlotRange -> {{-10, 10}, {-5, 5}, {-5, 5}}, {r, 0.1, 3} ]

```

Out[]:=



♦ **Nota:** El tamaño de las esferas cambia dinámicamente, pero el rango de visualización de los ejes se mantiene fijo, se puede manipular las coordenadas para que las esferas no choquen.

```

In[ ]:= Export["esfera.mp4", Esfera3d]

```

Out[]:=

esfera.mp4

```

In[ ]:= SystemOpen[DirectoryName[AbsoluteFileName["esfera.mp4"]]]

```

■ Mejoras personales

```

# Crea un manipulador interactivo para dibujar cuatro esferas 3D de diferentes colores y tamaños
# Las esferas están distribuidas en 4 cuadrantes del plano XY, con la distancia entre ellas dependiente del radio r
# Esfera roja: en el primer cuadrante con coordenada {r*4, r*4, 0}
# Esfera verde: en el segundo cuadrante con coordenada {-r*4, r*4, 0}
# Esfera azul: en el tercer cuadrante con coordenada {r*4, -r*4, 0}
# Esfera amarilla: en el cuarto cuadrante con coordenada {-r*4, -r*4, 0}
# Los ejes están activados con Axes -> True y etiquetados como "x", "y", "z"
# Boxed -> True cuadro de la escena 3D
# PlotRange está configurado para mostrar un espacio de visualización más amplio: x, y, z entre -20 y 20

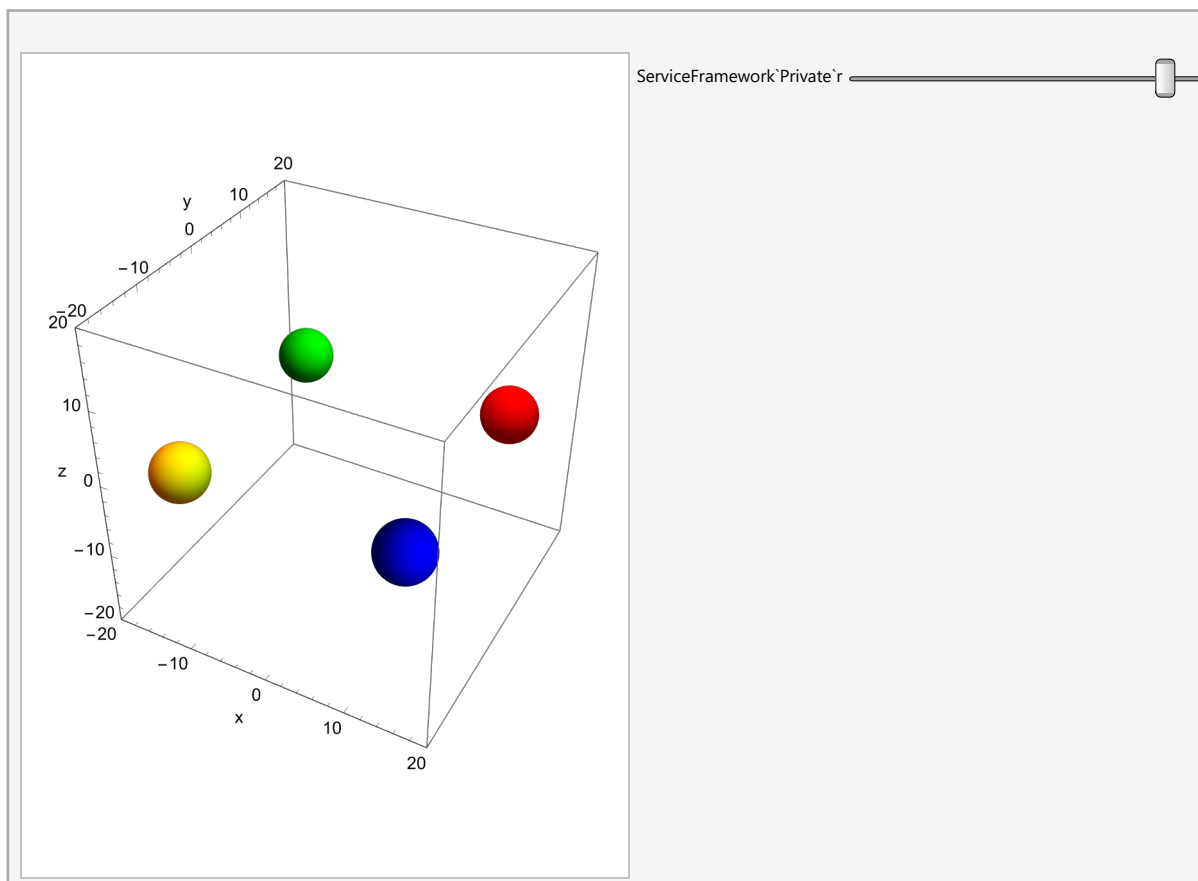
```

```

In[ ]:= Manipulate[Graphics3D[{Style[Sphere[{r*4, r*4, 0}, r], Red],
  Style[Sphere[{-r*4, r*4, 0}, r], Green], Style[Sphere[{r*4, -r*4, 0}, r], Blue],
  Style[Sphere[{-r*4, -r*4, 0}, r], Yellow]}],
  Axes → True, AxesLabel → {"x", "y", "z"}, Boxed → True,
  PlotRange → {{-20, 20}, {-20, 20}, {-20, 20}}, {r, 0.2, 4}]

```

Out[]:=



Nota:

- Manipula el parámetro r que representa el radio de las esferas.
- El valor de r puede variar entre 1 y 4 (gracias al manipulador interactivo).
- Cada esfera se coloca en un cuadrante diferente del espacio 3D:
 - **Esfera roja:** en el primer cuadrante con coordenadas $\{r*4, r*4, 0\}$.
 - **Esfera verde:** en el segundo cuadrante con coordenadas $\{-r*4, r*4, 0\}$.

- **Esfera azul:** en el tercer cuadrante con coordenadas $\{r*4, -r*4, 0\}$.
- **Esfera amarilla:** en el cuarto cuadrante con coordenadas $\{-r*4, -r*4, 0\}$.
- El parámetro **$r*4$** se usa para garantizar que las esferas no se solapen.
 - **r** es el radio de cada esfera, y **$r*4$** controla la separación entre ellas.
 - Si **r** es el radio de una esfera, la distancia mínima entre los centros de las esferas es de **$r*4$** .
 - Esto asegura que las esferas se mantengan separadas, incluso cuando el valor de **r** cambie.
- Al usar **$r*4$** , se mantiene suficiente distancia entre las esferas para que no se toquen.
 - La distancia entre las esferas es siempre al menos **$2r$** (el doble del radio).
 - La separación real entre los centros de las esferas es mayor o igual a **$2r$** para evitar el solapamiento.
- La visualización de las esferas se realiza en el rango de coordenadas -20 a 20 en los ejes x, y, y z (*PlotRange* -> $\{\{-20, 20\}, \{-20, 20\}, \{-20, 20\}\}$).
 - Esto proporciona suficiente espacio para observar las esferas y sus posiciones.
- **Axes** -> True' y 'AxesLabel' etiquetan los ejes para mayor claridad en la visualización.
- **Boxed** -> True' mantiene el marco alrededor de la visualización 3D.

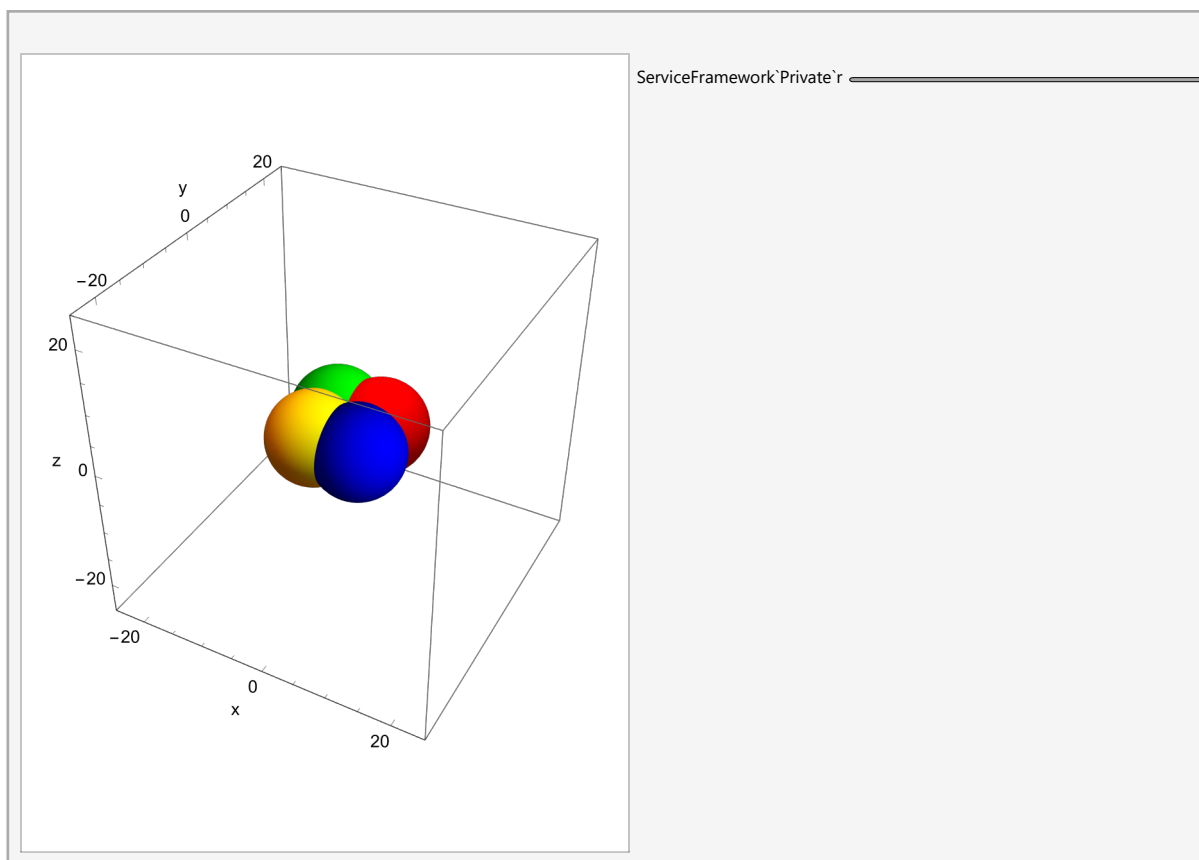
Variación con distancia de radio $r=0.5$

```

In[ ]:= Manipulate[Graphics3D[ {Style[Sphere[{r*0.5, r*0.5, 0}, r], Red],
  Style[Sphere[{-r*0.5, r*0.5, 0}, r], Green],
  Style[Sphere[{r*0.5, -r*0.5, 0}, r], Blue],
  Style[Sphere[{-r*0.5, -r*0.5, 0}, r], Yellow] },
  Axes → True, AxesLabel → {"x", "y", "z"}, Boxed → True,
  PlotRange → {{-25, 25}, {-25, 25}, {-25, 25}}, {r, 0.2, 7}

```

Out[]:=



■ Mejoras personales

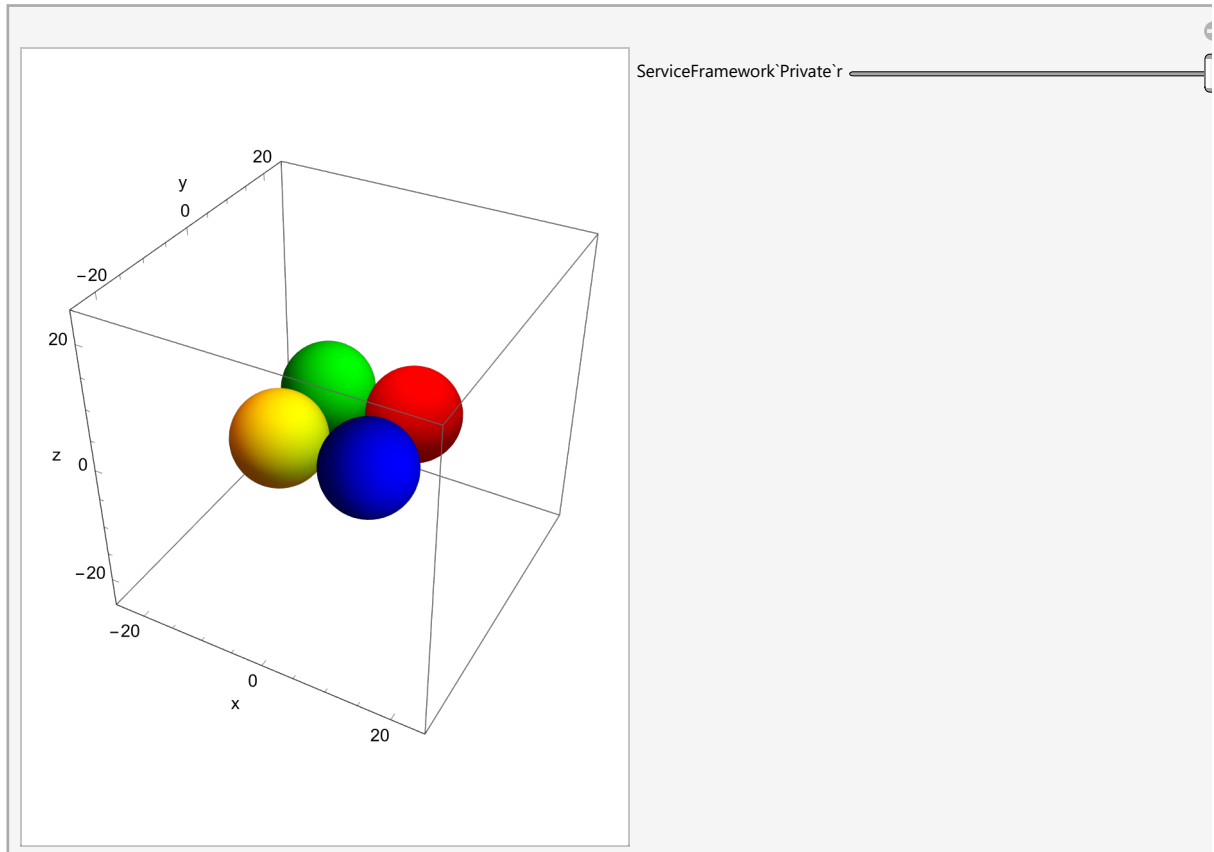
Variación con distancia de radio = r

```

In[ ]:= Manipulate[
  Graphics3D[ {Style[Sphere[{r, r, 0}, r], Red], Style[Sphere[{-r, r, 0}, r], Green],
    Style[Sphere[{r, -r, 0}, r], Blue], Style[Sphere[{-r, -r, 0}, r], Yellow] },
  Axes → True, AxesLabel → {"x", "y", "z"}, Boxed → True,
  PlotRange → {{-25, 25}, {-25, 25}, {-25, 25}}, {r, 0.2, 7} ]

```

Out[]=



■ Mejoras personales

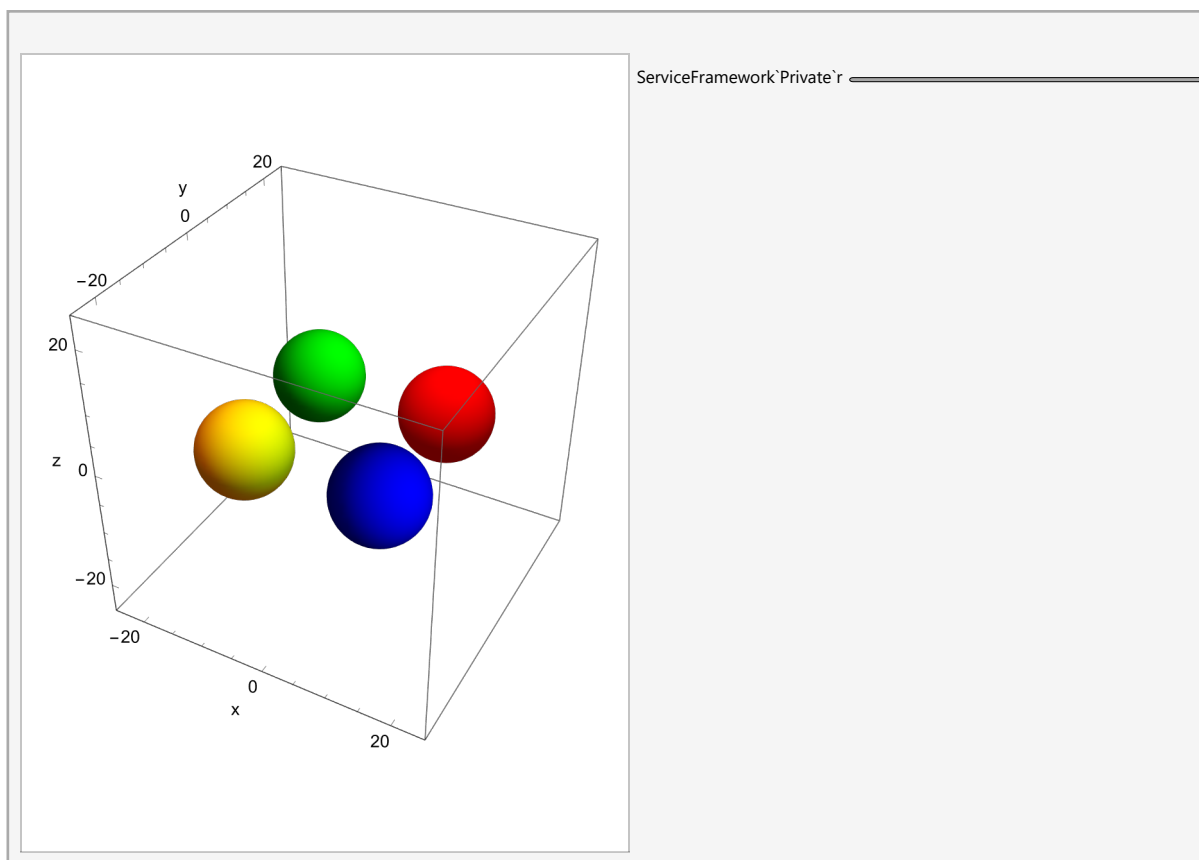
Variación con distancia de radio $r=1.5$

```

In[ ]:= Manipulate[Graphics3D[{Style[Sphere[{r*1.5, r*1.5, 0}, r], Red],
  Style[Sphere[{-r*1.5, r*1.5, 0}, r], Green],
  Style[Sphere[{r*1.5, -r*1.5, 0}, r], Blue],
  Style[Sphere[{-r*1.5, -r*1.5, 0}, r], Yellow] },
  Axes → True, AxesLabel → {"x", "y", "z"}, Boxed → True,
  PlotRange → {{-25, 25}, {-25, 25}, {-25, 25}}, {r, 0.2, 7}]

```

Out[]:=



Tareas

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

Tarea 4 – Esferas 3D

📅 2025/10/01

Se solicita subir los ejercicios de clase, asegurándose de que cumplan con el formato indicado, incluyendo color de fondo, comentarios y títulos. (Gráficas con animación)

Ver Clase 5 (Retos en clase)